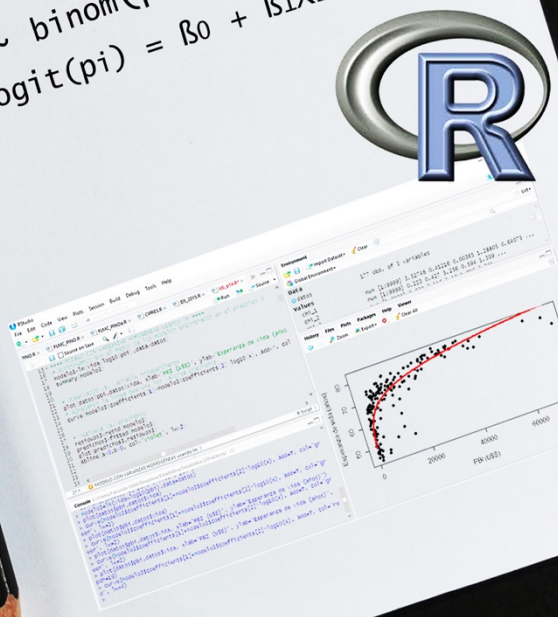
Modelos generalizados aplicados a la Economía en lenguaje R

$$Y_i \sim binom(p_i; N)$$
$$logit(p_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$$



Lucas A. Garibaldi
Facundo J. Oddi
Francisco J. Aristimuño
Aliosha N. Behnisch

Garibaldi, Lucas Alejandro
Modelos generalizados aplicados a la economía en lenguaje R / Lucas Alejandro Garibaldi. - 1a ed . - San Carlos de Bariloche : Lucas Alejandro Garibaldi, 2016.
Libro digital, PDF

Archivo Digital: descarga
ISBN 978-987-42-3009-6

1. Análisis Estadístico. 2. Modelado. 3. Programación. I. Título.
CDD 310

Cita de la obra
Garibaldi L.A., Oddi F., Aristimuño F.J., Behnisch A.N. 2016. Modelos generalizados aplicados a la Economía en el lenguaje R. 1a ed. - Bariloche: el autor. pp. 197.
ISBN: 978-987-42-3009-6

Modelos generalizados aplicados a la Economía en lenguaje R

Lucas A. Garibaldi, Facundo Oddi, Francisco J. Aristimuño, Aliosha N. Behnisch

Lucas Alejandro Garibaldi

Doctor en Ciencias Agropecuarias – UBA
Profesor regular – UNRN
Investigador – CONICET
lgaribaldi@unrn.edu.ar

Facundo José Oddi

Doctor en Biología – UNCo
Jefe de Trabajos Prácticos – UNRN
Becario Posdoctoral Fundación Bunge y Born
foddi@unrn.edu.ar

Francisco Javier Aristimuño

Licenciado en Ciencias Económicas – UBA
Doctorando en Ciencias Económicas – UBA
Jefe de Trabajos Prácticos regular – UNRN
Becario Doctoral – CONICET
faristimuno@unrn.edu.ar

Aliosha Nicolás Behnisch

Estudiante Lic. en Economía – UNRN
Ayudante Alumno – UNRN
aliosha.n.behnisch@gmail.com

Prefacio

Los profesionales de las ciencias económicas y ambientales deben resolver problemas a partir de la colección y el análisis de datos. En general, estos datos son tomados de una muestra procedente de relevamientos o experimentos, es decir que la información con la que trabajan es parcial. Por lo tanto, estos deben contar con herramientas que los ayuden a tomar la mejor decisión ante preguntas que tienen respuestas inciertas. La estadística aporta las herramientas necesarias para colectar los datos adecuadamente, a la vez que permite resumir, presentar y extrapolar la información contenida en la muestra. Luego, posibilita cuantificar la incertidumbre asociada a nuestras respuestas o, dicho de otra manera, la probabilidad de equivocarnos en la decisión tomada. Por lo tanto, al utilizar la estadística, las conclusiones estarán sustentadas por un sólido marco de análisis.

Con el objetivo de brindar a estudiantes y profesionales las técnicas estadísticas necesarias para un adecuado análisis de datos, el Dr. Lucas A. Garibaldi ha diseñado el curso de postgrado "Modelos generalizados aplicados a la Economía en el lenguaje R", el cual también forma parte de la currícula de grado de la Licenciatura en Economía de la Universidad Nacional de Rio Negro (como Estadística II). El mismo cubre los temas de un segundo curso de grado en Estadística ampliando sus contenidos a muchas de las herramientas utilizadas actualmente para la resolución de problemas económicos y ambientales, pero de las que la oferta académica es reducida. Creemos que ello, junto con el enfoque de taller del curso, flexibiliza el aprendizaje del estudiante ayudándolo a lograr independencia para resolver los problemas a los cuales se enfrentará durante su actividad profesional.

A lo largo de los capítulos (suponemos que el lector está familiarizado con los conceptos básicos de estadística y probabilidad) ponemos a disposición ejercicios prácticos para adquirir los conocimientos básicos sobre cómo colectar datos (diseño de estudios), modelarlos y analizarlos utilizando el programa R. Recomendamos seguir los capítulos frente a una computadora analizando las sentencias y los datos reales que se encuentran disponibles en el siguiente enlace https://www.mediafire.com/folder/zlgobjqlknybx/Modelos_generalizados_aplicados_a_la_Economía_en_lenguaje_R_(2016) . Los ejercicios son presentados con la intención de proporcionar al estudiante un marco similar al que un profesional se enfrentaría comúnmente en su ámbito de trabajo. Esto es, con un marco conceptual del que deriva

un problema relacionado, y para el cual el alumno es guiado hacia su resolución (analizar cómo ha sido la recolección de los datos, explorarlos, plantear modelos interesantes, determinar si el modelo planteado es adecuado, plantear modelos alternativos, realizar las inferencias y predicciones) y arribar a una conclusión. Un aspecto relevante es que los ejercicios están basados en datos reales (datos publicados en sitios web o cedidos por colegas).

Limitándonos al estudio de modelos con una sola variable de respuesta (dependiente), la obra se organiza en ocho Unidades. Las primeras siete abarcan el modelado de datos con distribución normal. En la Unidad I se estudia el modelo de regresión lineal entre dos variables cuantitativas (Regresión Lineal Simple) y se introducen los conceptos de Criterio de Mínimos Cuadrados Ordinarios para la estimación de parámetros, bondad de ajuste y validez de los modelos a partir de sus supuestos. Las Unidades II y III tratan con variables independientes categóricas, el Análisis de la Varianza (ANOVA) y los test a posteriori (comparaciones múltiples). En particular, se aborda el Diseño Completamente Aleatorizado (DCA), y se introduce el modelado con más de una variable independiente, todas categóricas en este caso, a partir del Diseño en Bloques Completamente Aleatorizados (DBCA) y el Diseño Multifactorial. La Unidad IV trata con más de una variable independiente, pero en este caso cuantitativas (Regresión Lineal Múltiple) e introduce un aspecto fundamental del modelado estadístico: la multicolinearidad. Para ello, en esta unidad se estudian las sumas de cuadrados parciales (ANOVA Tipo III) y secuenciales (ANOVA Tipo I). En las Unidad V se formaliza el concepto de Modelo Lineal General y se plantean problemas que tratan con variables independientes cuantitativas y categóricas de manera conjunta. Se introducen además el concepto de Verosimilitud y los distintos Criterios de Información (AIC, BIC etc.) como medidas de bondad de ajuste, y el Criterio de Máxima Verosimilitud como método de estimación de parámetros. Al llegar a la Unidad VI, se incorpora el modelado de la varianza flexibilizando la homocedasticidad, uno de los supuestos del modelo lineal. Por su lado, la Unidad VII cubre conceptos detrás del modelado de relaciones no lineales. Finalmente, en la Unidad VIII se flexibiliza el supuesto de normalidad para modelar datos no normales. Este es el campo de los Modelos Lineales Generalizados que permiten tratar con distribuciones de la familia exponencial: Binomial, Poisson, Normal y Gamma, y también se abarca la distribución Binomial Negativa. En forma general, a través de la obra presentamos el marco de inferencia frecuentista y abordamos la evaluación de relaciones de verosimilitud e

inferencias multimodelo. Los capítulos no desarrollan los conceptos teóricos. Para ello, sugerimos la lectura de diversos libros de textos que abordan los conceptos presentes en esta obra de manera exhaustiva (Pinheiro y Bates 2000; Webster, 2000; Anderson et al., 2008; Gelman y Hill 2007; Fox y Weisberg, 2010, 2011; entre otros).

Esta obra es fruto de varios años llevando adelante este curso en la Universidad Nacional de Río Negro. Esperemos que la misma les resulte útil y de ayuda para el abordaje de sus propios análisis.

Lucas Alejandro Garibaldi
Facundo José Oddi
Francisco Javier Aristimuño
Aliosha Behnisch

Contenidos

¿Cómo usar este libro?

Este libro está dirigido hacia aquellas personas que ya se encuentran con datos para analizar. No es un primer libro de Estadística. Obtendrán un mejor uso del libro leyendolo desde una computadora en el programa R. Para ello pueden descargar los archivos de código (en la jerga conocidos como "scripts") y de datos directamente a su computadora desde el siguiente enlace: https://www.mediafire.com/folder/zlgobjqlknybx/Modelos_generalizados_aplicados_a_la_Economía_en_lenguaje_R_(2016). Recomendamos que ejecuten los scripts también con sus propios datos, además de los ejemplos con datos reales provistos. El script (archivo .R) que se descarga de la versión en línea sólo tiene códigos y texto (antecedido por un numeral). La versión pdf del libro además presenta algunos resultados de aplicar los códigos, como gráficos y tablas. A lo largo de los prácticos explicamos cómo utilizar el programa R. Este programa tiene varias ventajas, como ser gratuito y de código abierto. Tal vez por ello, este programa es muy completo en sus capacidades para resolver distintos modelos estadísticos y está siendo utilizado habitualmente a nivel mundial. Existen muchos "paquetes" orientados específicamente a resolver problemas de múltiples profesiones. Algunos de ellos que consideramos claves también serán introducidos a lo largo de los prácticos. En caso que el lector no tenga experiencia en R, recomendamos una lectura progresiva. El libro va sumando información capítulo a capítulo y varios aspectos utilizados brevemente en capítulos avanzados son explicados con gran detalle en los primeros capítulos. Aquellos lectores con cierta experiencia en R, y un sólido conocimiento estadístico, podrán consultar sólo aquellos capítulos que necesiten para resolver un problema particular y utilizarlo para recordar algunos aspectos claves de códigos y conceptuales que hayan olvidado. El sistema de aprendizaje propuesto se basa en el conocimiento situado. En cada práctico partimos de una problemática real a resolver que nos lleva a discutir y presentar marcos conceptuales relevantes de la estadística para una mejor resolución de ese problema. Es decir que si bien el libro está organizado en "prácticos", esto nos lleva a discutir conceptos "teóricos". Esperamos que la obra sea de utilidad para estudiantes y profesionales de ciencias exactas (básicas y aplicadas), ambientales, económicas y sociales.

Volver al Índice principal

Regresión lineal de dos variables

Trabajo Práctico Nº 1: Introducción al lenguaje R y sus funciones básicas

```
# Este archivo se puede leer con cualquier editor de texto
# Pueden descargar gratuitamente R Studio de:
# http://www.rstudio.org/download/desktop

# R es un lenguaje y un ambiente para análisis de datos y gráficos. Puede ser
# considerado una implementación de S, un lenguaje de programación desarollado
# inicialmente en los laboratorios Bell durante la década de los 70s. El proyecto R
# fue iniciado por Robert Gentleman y Ross Ihaka en la Universidad de Auckland,
# Nueva Zelanda, en la década de los 90s, y continuó siendo desarrollado por un
```

equipo internacional desde mediados de 1997.

Instrucciones para instalar R y R Studio:

Sitio de descarga para R:
https://cran.r-project.org/Una vez descargado R, se procede a instalarlo y, una vez iniciado, instalar demás plug-ins que el programa pida instalar. Completado este paso, se descargará R Studio del siguiente sitio: https://www.rstudio.com/products/rstudio/download/. Este programa, como el anterior, se instala y se instalan también los demás paquetes accesorios que pudiera pedir el programa una vez iniciado. Es en este último programa, R, que abriremos nuestros códigos ("scripts" en la jerga) con los cuales trabajaremos a lo largo del libro.

1.1 Codificación del texto

```
# Para leer correctamente este script (y los siguientes) deben ir a:
# 1) "Tools"
# 2) "Options"
# 3) cambiar "Default texting encoding" a "UTF-8"
# Esto garantiza que podamos abrir los scripts tanto en LINUX como en WINDOWS
 sin errores.
```

1.2 Directorio de trabajo

```
# Indicar "working directory" (directorio de trabajo) en la ventana inferior derecha:
# 1) Situarse en el directorio donde se encuentran los archivos a trabajar
# 2) Apretar "More" y luego "Set As Working Directory"
```

1.3 Importar datos

```
# cargar tabla de datos, en este caso con extensión txt
# a esta tabla la llamaremos "datos"
datos=read.table("datos_p_1.txt", header=TRUE)
# lo que acabamos de hacer es crear un OBJETO llamado "datos"
# "read.table" es una FUNCIÓN
# "header" es un ARGUMENTO (opción) de la FUNCIÓN


# datos provenientes de la Organización de las Naciones Unidas
# para la Agricultura y la Alimentación (FAO):
# http://www.fao.org/index_es.htm
```

1.4 Explorar datos

```
# miramos las primeras filas de la tabla


head(datos)
```

```
      pais     crec_prod  cv_prod     riqueza    area
```

```
1   Argentina    1.028730   14.45836   62.79167  19739246.604
2   Afghanistan  1.007682   16.27734   32.54167  3251411.229
3   Albania      1.026333   21.87418   32.10417  455077.875
4   Algeria      1.043270   24.72541   50.04167  3588182.542
5   A_Samoa      1.014967   25.91996   10.93750  4091.083
6   Angola       1.027138   43.46338   28.00000  2038843.104
```

```
# si queremos saber que hace la función "head" le preguntamos a R:
?head

# ¿Cuántos países se relevaron?
length(datos$pais)
```

```
[1] 170
```

```
# Variables relevadas en cada país:

# crec_prod = Es el crecimiento anual promedio de la producción agrícola
# total de un país entre 1961 y 2008 (toneladas de materia prima producidas).
# Esta variable no tiene unidades ya que fue estimada como el promedio entre la
# producción del año t sobre el año t-1.

# cv_prod = representa la variabilidad interanual desde 1961 al 2008 en el
# volumen total de producción agrícola de un país. Expresada en porcentaje de
# la media.

# riqueza = número promedio de cultivos del país desde 1961 al 2008.

# area = área agríola (hectáreas) promedio del país desde 1961 al 2008.


# ¿En cuántos países decreció la producción agrícola?
length(subset(datos$crec_prod, datos$crec_prod<1))
```

```
[1] 170
```

```
# vemos la tabla de datos entera
View(datos)

# pedimos algunas medidas resumen para cada variable
summary(datos)
```

```
   pais              crec_prod        cv_prod           riqueza          area
 A_Samoa     :  1   Min.   :0.929   Min.   :  4.235   Min.   : 2.50   Min.   :      300
 Afghanistan :  1   1st Qu.:1.014   1st Qu.: 10.514   1st Qu.:21.30   1st Qu.:    82982
 Albania     :  1   Median :1.024   Median : 13.403   Median :31.75   Median :   911022
 Algeria     :  1   Mean   :1.026   Mean   : 16.888   Mean   :36.70   Mean   :  5594636
 Angola      :  1   3rd Qu.:1.037   3rd Qu.: 19.198   3rd Qu.:50.64   3rd Qu.:  3516285
 Antigua_Barb:  1   Max.   :1.161   Max.   :132.683   Max.   :97.73   Max.   :169689502
 (Other)     :164
```

```
# en la mayoría de los países el volumen de producción agrícola ha crecido.
# Este crecimiento fue, en promedio, de un 2.6% anual
```

```
# (este porcentaje surge de observar el valor "Mean: 1.026").
```

1.5 Problema y gráfico de dispersión

```
# Nos interesa evaluar cómo cambia el cv_prod en función de la riqueza
# de cultivos.
# Nuestro MARCO CONCEPTUAL PREDICE que países con más cultivos tendrán
 una
# producción más estable en el tiempo.

# Primero hacemos un DIAGRAMA DE DISPERSIÓN exploratorio
with(datos,plot(riqueza, cv_prod))
```

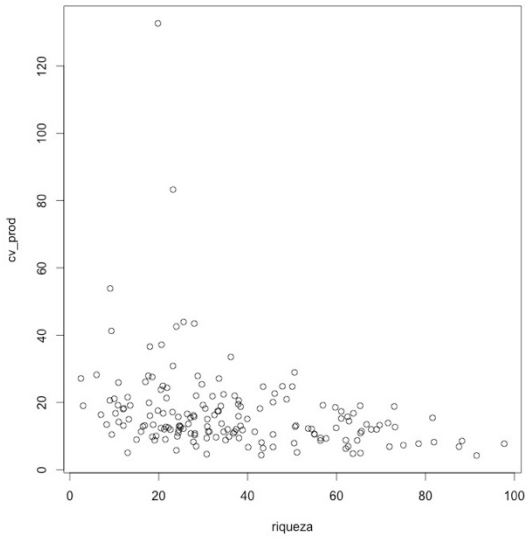


Figura 1.1: Diagrama de Dispersión del CV de la Producción en función de la Riqueza.

```
# Vemos que hay dos (de un total de 170) países que tienen CV
# extremadamente altos y esos países dominan la visual del gráfico
# mientras que en realidad nosotros queremos estudiar la tendencia habitual
# o más común entre países sin estar tan influenciada por valores extremos.
# Entonces una opción es usar log10(cv_prod).
with(datos,plot(riqueza, log10(cv_prod)))
```

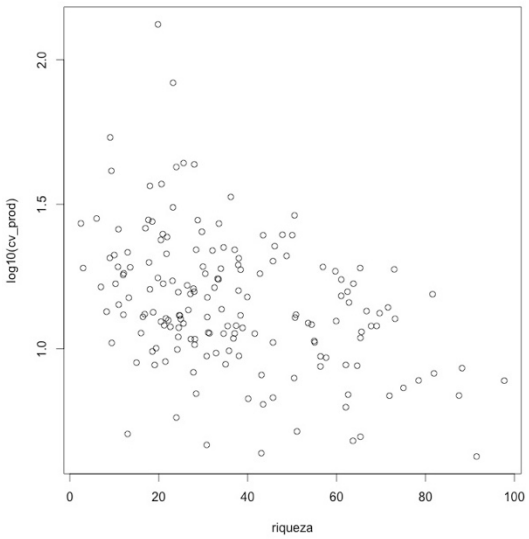


Figura 1.2: Diagrama de Dispersión del log10 del CV de la Producción en función de la Riqueza.

1.6 Modelo de regresión lineal simple

```
# agregamos la recta estimada promedio según un modelo de regresión lineal simple
with(datos,abline(lm(log10(cv_prod)~riqueza),lwd=3, col="blue"))
```

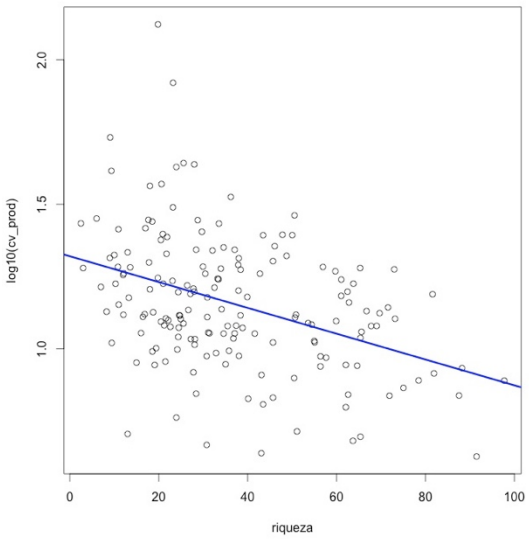


Figura 1.3: Diagrama de Dispersión del log10 del CV de la Producción en función de Riqueza, con la correspondiente recta de regresión lineal.

```
# lwd (por "line width") es un ARGUMENTO de la FUNCIÓN abline que controla el
  grosor de la línea
# col (por "color") es un ARGUMENTO de la FUNCIÓN abline que controla el color
  de la línea


# En la jerga estadísta definimos a cv_prod como la "variable dependiente" y
```

```
# a riqueza como la "variable independiente"

# Creamos un OBJETO llamado modelo con los resultados de estimar
# un modelo de regresión lineal simple sobre nuestros datos.
modelo=lm(log10(cv_prod)~riqueza,data=datos)

# "lm" utiliza el método de estimación por mínimos cuadrados ordinarios
# "lm" por "linear model"

# vemos los valores estimados de los parámetros.
modelo
```

```
Call:
lm(formula = log10(cv_prod) ~ riqueza, data = datos)

Coefficients:
(Intercept)      riqueza
   1.320205    -0.004469
```

```
# MODELO DE REGRESIÓN LINEAL SIMPLE          yi  = B0     +      B1   *  xi +
 Ei      i=1,....,N
# MODELO ESTIMADO DE REGRESIÓN LINEAL SIMPLE    yi  = 1.32    -   0.0045
 *  xi + ei      i=1,....,170
# ECUACIÓN DE REGRESIÓN LINEAL SIMPLE          yi  = 1.32    -   0.0045   *  xi
# ¿Qué CV_prod promedio tendrán los países con una riqueza de 25 cultivos?
1.320205-0.004469*25 # da valores en log10
# alternativamente
modelo$coefficients[1]+modelo$coefficients[2]*25  # da valores en log10
# retransformando de log10(cv_prod) a cv_prod:
10^(modelo$coefficients[1]+modelo$coefficients[2]*25)
# continuará.....
```

1.7 Objetos

```
# El principio fundamental detrás de R (y S) es "todo es un
# objeto". Entonces, no sólo vectores y matrices son objetos que pueden ser
# procesados por funciones, sino que las funciones mismas y sus características
# también son objetos. Esto permite computación en el lenguaje y puede simplificar
# tareas de programación.

objects()
# Esta función indica solamente los objetos "nuevos" que hemos creado.
# Hay más de mil objetos que están en el paquete base que no se muestran aquí
# incluyendo la función objects... :-) .... ver:
objects("package:base")
```

1.8 Limpiar espacio de trabajo

```
# Para eliminar todos los objetos del espacio de trabajo ("workspace")
# apretamos "Clear All". También se puede usar la función rm()
```

Trabajo Práctico Nº 2: Primeros pasos con modelo de regresión lineal simple

2.1 Problema

```
# Desde un órgano de un gobierno provincial patagónico se está
# evaluando la posibilidad de imponer un impuesto a los hogares que
# posean mayor número de ambientes (habitaciones). Existe el supuesto de que
# hogares más grandes se condicen con ingresos mayores.
# Uds sospecha de la efectividad de la medida y decide tomar una
# muestra de 30 viviendas al azar relevando el número de ambientes,
# el ingreso total familiar (ITF) y la cantidad de miembros en el hogar.

# Los datos pertencen al 4to trimestre de 2010 y fueron obtenidos de
# http://www.indec.mecon.ar/principal.asp?id_tema=9556
```

2.2 Cargar datos manualmente “concatenando”

```
# Cargamos los datos, el orden es importante:
ambientes = c(4, 4, 3, 2, 3, 3, 4, 5, 3, 2, 2, 2, 3, 4, 4, 2, 1,
          4, 2, 2, 5, 1, 1, 1, 4, 3, 5, 3, 3, 1)
ITF = c(2500, 2600, 430, 1757, 720, 1830, 900, 480, 2800, 700, 1800,
       60, 1950, 1890, 500, 1500, 1654, 2800, 800, 200, 1000, 600,
       1340, 160, 2000, 1800, 2350, 550, 480, 600)
personas = c(7, 5, 2, 3, 2, 4, 2, 5, 10, 2, 5, 6, 4, 3, 5, 5, 3, 6,
           2, 1, 2, 1, 5, 2, 4, 1, 5, 1, 1, 2)
# "c" significa concatenar (unir)
```

2.3 Diseño

```
# Identifique la unidad experimental, la muestra y la población.


# ¿De que tipo de variables se trata?
```

2.4 Gráfico de dispersión

```
# Realice un diagrama de dispersión que sea de utilidad para cumplir con su objetivo
plot(ambientes, ITF)
```

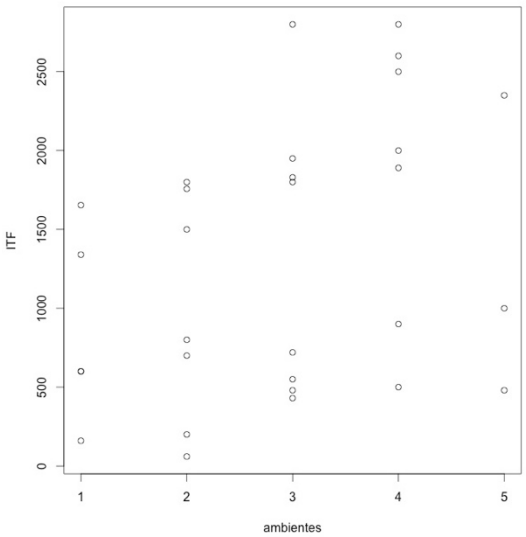


Figura 2.1 Ingreso Total Familiar en función de cantidad de miembros del hogar

```
# ¿Existe asociación lineal entre ambas variables?
```

2.5 Modelo: estimación y predicción

```
# Plantee y estime un modelo adecuado para cumplir con su objetivo
modelo=lm(ITF~ambientes)
# lm es la función que ajusta modelos lineales por el método de mínimos
# cuadrados ordinarios
?lm

modelo
Call:
lm(formula = ITF ~ ambientes)

Coefficients:
(Intercept)    ambientes
      630.2        230.8

# ¿Qué le dice este modelo sobre la relación entre el número de ambientes e ITF?
# Interprete el valor de la pendiente en términos del problema e indique sus unidades
# ¿Qué ITF pronosticaría para personas que viven en hogares con 4 ambientes?
modelo$coefficients[1]+modelo$coefficients[2]*4

(Intercept)
   1553.226

# otra manera es utilizando la función "predict.lm"
nuevos_datos=data.frame(ambientes=4)
nuevos_datos

   ambientes
1          4
```

```
predict.lm(modelo,newdata=nuevos_datos)
```

```
       1
1553.226
```

```
# "data.frame" sirve para crear una tabla de datos.
# Ver más información al respecto abajo


# ¿Qué ITF promedio pronosticaría para personas que viven en hogares de 8 ambientes?


# ¿El análisis estadístico asegura que el ITF es causa del número de ambientes?


# ¿Por debajo de cuantos ambientes se encuentra el 75% de los hogares?
summary(ambientes)
```

```
  Min.    1st Qu.  Median    Mean    3rd Qu.   Max.
  1.000   2.000    3.000     2.867   4.000    5.000
```

```
boxplot(ambientes)
```

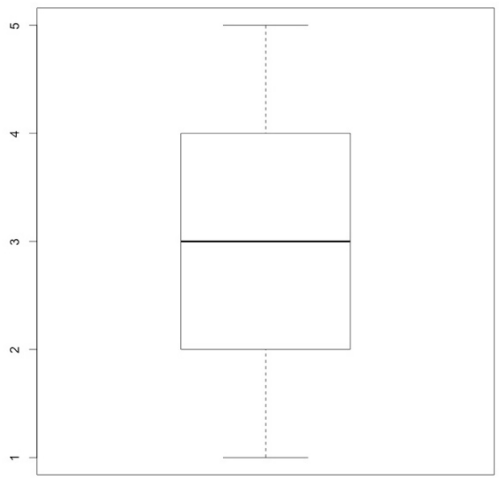


Figura 2.2: Diagrama de caja y bigotes para "ambientes"

```
# En promedio, ¿cuántos miembros tiene un hogar patagónico?


# Confeccione otro gráfico de dispersión
# pero que relacione la cantidad de ambientes con la cantidad de miembros en cada
hogar.
# ¿Existe  relación lineal entre ambas variables?
```

```
# Concluya respecto del objetivo del trabajo

# GRÁFICO: modelo y datos ####
plot(ambientes, ITF,
     xlab="Ambientes en el hogar (número)",
     ylab="Igreso total familiar (pesos/familia)")
abline(lm(ITF~ambientes), col="blue", lw=3)
```

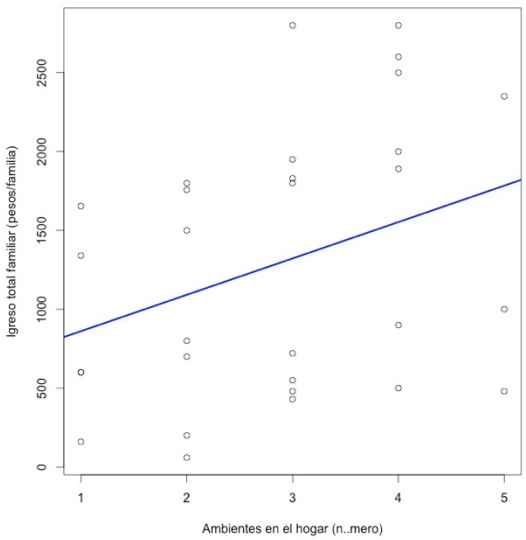


Figura 2.3: Ingreso Total Familiar en función de Cantidad de Ambientes en el Hogar, con respectiva linea de regresión estimada

2.6 Clases y funciones genéricas

```
# Todo objeto tiene una clase que puede ser consultada usando class().
# Por ejemplo:
# "data.frame" es una clase de objeto (una lista o tabla
#         con una cierta estructura, es el formato preferido para contener los datos),
# "lm" es otra clase de objeto (representa un modelo de
#         regresión lineal ajustado por el método de mínimos cuadrados ordinarios) y
# "matrix" es otro ejemplo (que es como lo indica el nombre, una matriz).

# Para cada clase de objeto, existen ciertos métodos,
# llamados funciones genéricas, que están disponibles;
# ejemplos típicos son summary() y plot().
# El resultado de estas funciones depende en la clase de objeto sobre la cual se aplique.
# Cuando se la aplica sobre un vector numérico, summary() devuelve resumenes básicos
de
# de la distribución empírica de los datos (medidas descriptivas de la muestra), como la
# media y la mediana. Cuando se aplica sobre un vector de datos categóricos
(cualitativos)
# en cambio, devuelve una tabla de frecuencia. Y en el caso de que se aplique sobre
# un modelo lineal, el resultado es el resúmen estandar para una regresión (valor-p, etc).
# En forma similar, la función plot() devuelve un diagrama de dispersión cuando se
aplica
```

```
# sobre un conjunto de datos y devuelve un diagnóstico básico de gráficos cuando se la
# aplica sobre un objeto que sea un modelo lineal.
```

2.7 Ejercicio para el hogar

```
# Se seleccionaron 12 consumidores al azar para estudiar si su consumo depende de su
ingreso.
# Los datos fueron relevados en Florida (USA) y las unidades son miles de dólares
estadonunidenses.

# Datos provenientes de
# Webster, A. L. (2000) Estadística aplicada a los negocios y la economía.
# 3era edición. Ed. Irwin McGraw-Hill. (página 335)

# Cargamos los datos, el orden es importante:
ingreso=c(24.3, 12.5, 31.2, 28.0, 35.1, 10.5, 23.2, 10.0, 8.5, 15.9, 14.7, 15)
consumo=c(16.2, 8.5, 15, 17, 24.2, 11.2, 15, 7.1, 3.5, 11.5, 10.7, 9.2)
# "c" significa concatenar (unir)

# Realice un diagrama de dispersión para los datos
plot(ingreso, consumo)

# Plantee y estime un modelo adecuado para cumplir con su objetivo
modelo=lm(consumo~ingreso)
modelo

Call:
lm(formula = consumo ~ ingreso)

Coefficients:
(Intercept)      ingreso
     1.7779       0.5582

# ¿Qué le dice este modelo sobre la relación entre el consumo y el ingreso?
# ¿Qué proporción de cada dólar adicional que se gana se invierte en consumo?

# ¿Qué consumo pronosticaría el modelo para personas que ganan 27.5 mil dólares
estadounidenses?
modelo$coefficients[1]+modelo$coefficients[2]*27.5

(Intercept)
   17.12759

# otra manera es utilizando la función "predict.lm"
nuevos_datos=data.frame(ingreso=27.5)
nuevos_datos

ingreso
1    27.5

predict.lm(modelo,newdata=nuevos_datos)
```

```
1
17.12759
```

```
# "data.frame" sirve para crear una tabla de datos.

# ¿Qué consumo promedio pronosticaría para personas que ganan 2.5 mil dólares estadounidenses?

# ¿El análisis estadístico asegura que el ingreso causa el consumo?
```

Trabajo Práctico Nº 3: Pruebas, Intervalos y Supuestos del modelo de regresión lineal simple

3.1 Problema y datos

```
# Recordar indicar el directorio de trabajo.

# Seguimos trabajando con los datos del TP 1,
# los cuáles provienen de la Organización de las Naciones Unidas
# para la Agricultura y la Alimentación:
# http://www.fao.org/index_es.htm

# cargamos la tabla
datos=read.table("datos_p_1.txt", header=TRUE)
```

3.2 Modelo, Prueba “T” e intervalos

```
# Recordamos que queremos estudiar cómo varía el coeficiente de variación
# de la producción agrícola de un país (cv_prod) en función
# de su cantidad de cultivos (riqueza)
modelo=lm(log10(cv_prod)~riqueza,data=datos)
modelo

Call:
lm(formula = log10(cv_prod) ~ riqueza, data = datos)

Coefficients:
(Intercept)      riqueza
   1.320205    -0.004469

# MODELO DE REGRESIÓN LINEAL SIMPLE          yi  = B0    +   B1   *  xi + Ei
i=1,....,N
# MODELO ESTIMADO DE REGRESIÓN LINEAL SIMPLE    yi  = 1.32   - 0.0045 *
xi + ei      i=1,....,170
# ECUACIÓN DE REGRESIÓN LINEAL SIMPLE         yi  = 1.32   - 0.0045 *  xi

summary(modelo)

Call:
lm(formula = log10(cv_prod) ~ riqueza, data = datos)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.5572 -0.1287 -0.0047  0.1416  0.8912

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.3202047  0.0337274  39.143  < 2e-16 ***
riqueza     -0.0044694  0.0008017  -5.575  9.7e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.215 on 168 degrees of freedom
Multiple R-squared:  0.1561,    Adjusted R-squared:  0.1511
F-statistic: 31.08 on 1 and 168 DF,  p-value: 9.704e-08
```

```
# ¿Qué significa el valor p para la fila riqueza?
# ¿Cuál es la hipótesis nula y la hipótesis alternativa?


# Para los que quieren el valor de t "crítico":
qt(p=0.025, df=170-2)
```

```
[1] -1.974185
```

```
# o
qt(p=0.025, df=170-2, lower.tail=FALSE)
```

```
[1] 1.974185
```

```
# como prefieran

# Intervalos de confianza para la ordenada y la pendiente
confint(modelo, level=0.95)
```

```
                2.5 %       97.5 %
(Intercept)  1.25362066  1.386788827
riqueza     -0.00605218  -0.002886701
```

```
# ¿Qué significa el intervalo para "(Intercept)"? ¿Y para riqueza?
# Relacionar con los parámetros del modelo.


# Intervalo de confianza para la media de log10(cv)
# para países con 29 cultivos
nuevos_datos=data.frame(riqueza=29)
predict.lm(modelo,newdata=nuevos_datos, interval="confidence")
```

```
   fit           lwr          upr
1 1.190591     1.155834    1.225348
```

```
# Interpretar
```

```
# Ahora el intervalo de predicción
predict.lm(modelo,newdata=nuevos_datos, interval="prediction")
```

```
   fit           lwr          upr
1 1.190591     0.764751     1.616431
```

```
# Interpretar

# Incorporamos los intervalos para cada valor de riqueza observado a la tabla de datos
datos[,6:8]=predict.lm(modelo,interval="confidence")
datos[,6:8]
datos[,9:11]=predict.lm(modelo,interval="prediction")
datos[,9:11]
colnames(datos)=c("pais","crec_prod","cv_prod","riqueza","area",
                  "media","iclow","icup","media2","iplow","ipup")

# diagrama de dispersión
with(datos,plot(riqueza,log10(cv_prod),ylim=c(0.5,2.3)))

# agregamos los valores predichos (promedios)
datos=datos[order(datos$media),] # ordenamos la matriz según media en orden
creciente
lines(datos$riqueza,datos$media,lwd=2, col="red")

# agregamos al gráfico el límite superior e inferior de intervalo de confianza
# para los valores predichos
datos=datos[order(datos$icup),] # ordenamos la matriz según icup en orden creciente
lines(datos$riqueza,datos$icup,lwd=2, lty=5, col="red")
# "lty" es el tipo de línea (por "line type")
datos=datos[order(datos$iclow),] # ordenamos la matriz según iclow en orden creciente
lines(datos$riqueza,datos$iclow,lwd=2, lty=5, col="red")

# agregamos el limite superior e inferior del intervalo de predicción
datos=datos[order(datos$ipup),]
lines(datos$riqueza,datos$ipup,lwd=2, lty=5, col="blue")
datos=datos[order(datos$iplow),]
lines(datos$riqueza,datos$iplow,lwd=2, lty=5, col="blue")
```

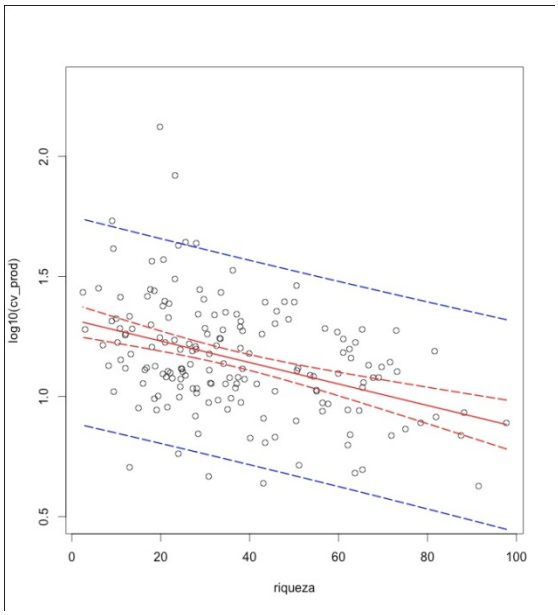


Figura 3.1: Intervalos de confianza y predicción para el log10(CV_Producción) en función de la Riqueza

Interpretar el gráfico prestando especial atención a los intervalos.

3.3 Bondad de ajuste

```
# CME como medida de bondad de ajuste
residuos=resid(modelo)
CME = (sum(residuos^2))/(length(residuos)-2)
CME
[1] 0.04621833

# el siguiente es el desvío estándar residual que está expresado
# en las mismas unidades que la variable dependiente
DS = sqrt(CME)
DS

[1] 0.2149845

# recuerden que por regla empírica, si los datos están distribuidos normalmente,
# el rango comprendido por
# la media +/- 1 desvío estándar contendrá al 68.3% de las observaciones
# media +/- 2 DS ----> 95.5% de las observaciones
# media +/- 3 DS ----> 99.7% de las observaciones


# línea de regresión junto con el intervalo que contiene a casi el 70% de los países
with(datos,plot(riqueza,log10(cv_prod)))
with(datos,abline(lm(log10(cv_prod)~riqueza),lwd=2, col="blue"))
with(datos,abline(lm(log10(cv_prod)+DS~riqueza),lwd=2,lty=5, col="blue"))
with(datos,abline(lm(log10(cv_prod)-DS~riqueza),lwd=2,lty=5, col="blue"))
```

```
# también es interesante ver:
predichos=fitted(modelo)
with(datos,plot(predichos,log10(cv_prod)))
abline(a=0,b=1,lwd=3, col="brown")
# ¿Cuál es la utilidad de la línea con ordenada al origen igual a cero
# y pendiente igual a 1?

# COEFICIENTE DE DETERMINACIÓN
# lo podemos ver haciendo
summary(modelo)
# mirar el "Multiple R-squared" el otro es interesante cuando trabajemos con
# regresión múltiple

# también podemos estimarlo "nosotros" y comparar con la salida anterior
SCE = sum(residuos^2)
SCTotal = with(datos, sum( ( log10(cv_prod)-(mean(log10(cv_prod))) ) ^2  )  )
r2 = (SCTotal - SCE) / SCTotal
r2

[1] 0.1561128

round(r2, 4)

[1] 0.1561
```

3.4 Coeficiente de correlación de Pearson

```
# Este índice NO es parte del modelo de regresión lineal simple con el que venimos
# trabajando. Se encuentra aquí para recordar este índice y su relación con el
# coeficiente de determinación (r2)

# coeficiente de correlación de Pearson (r):
r = with(datos,cor(log10(cv_prod),riqueza))
r

[1] -0.3951111

# NO es el coeficiente de "asimetría" de Pearson.

# Prueba de significancia sobre el coeficiente de correlación
with(datos,cor.test(log10(cv_prod),riqueza))

Pearson's product-moment correlation

data:  log10(cv_prod) and riqueza
t = -5.5748, df = 168, p-value = 9.704e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5149985 -0.2600627
sample estimates:
       cor
-0.3951111
```

```
# Interpretar el intervalo de confianza para r

# También se puede obtener como raíz cuadrada de r2 pero uno debe agregar el signo
# (el signo de r es el mismo que el de la pendiente)
sqrt(r2)
```

```
[1] 0.3951111
```

```
# Aplicando la ecuación de Pearson:
r_trabajoso =
  with(datos,
       (sum((log10(cv_prod)-mean(log10(cv_prod)))*(riqueza-mean(riqueza))))
       /
       (sqrt(      sum((log10(cv_prod)-mean(log10(cv_prod)))^2)
             *  sum((riqueza-mean(riqueza))^2)
       )))
r_trabajoso
```

3.5 Supuestos

```
# Volvemos al modelo de regresión lineal simple.
# Evaluamos los supuestos del modelo a través de los residuos
residuos=resid(modelo)
```

3.5.1 Independencia, Homogeneidad de varianzas y Linealidad

```
# 4.1) INDEPENDENCIA, HOMOGENEIDAD, LINEALIDAD ####
# Para evaluar el supuesto de HOMOGENEIDAD DE VARIANZAS,
# así como el de INDEPENDENCIA
predichos=fitted(modelo)
plot(predichos,residuos)

# Gráfico que complemente el anterior para evaluar INDEPENDENCIA,
# HOMOGENEIDAD DE VARIANZAS,
# y el supuesto de LINEALIDAD del modelo:
with(datos,plot(riqueza,residuos))

# El paquete car tiene una función interesante en este sentido
library(car)
residualPlots(modelo)
```

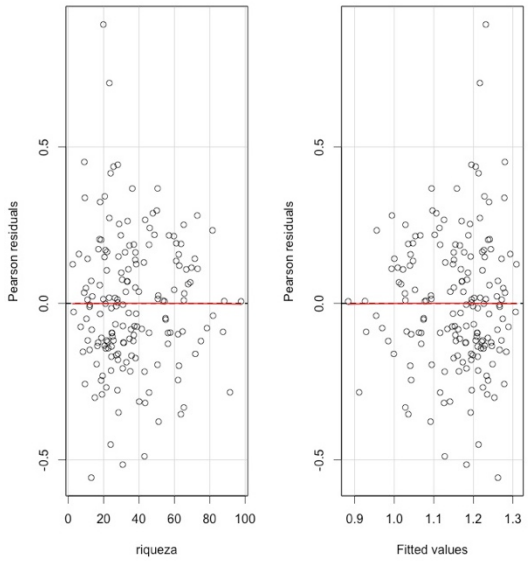


Figura 3.2: Gráficos de Residuos de Pearson en función de Riqueza y de valores predichos

```
# más adelante veremos que son los "Pearson" residuals,
# por ahora pensemos solamente en que son residuales.
?residualPlots
```

3.5.2 Normalidad

```
# 4.2) NORMALIDAD ####
par(mfrow=c(2,1))
hist(residuos, col="yellow")
boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# El punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior.
# "bty" por "box type"

# Comparamos con una distribución normal de igual desvío estándar residual a nuestros
datos
par(mfrow=c(2,2))
hist(residuos, col="yellow", freq=F)
lines(density(residuos), col="blue", lw=3)

normal=rnorm(mean=0, sd=summary(modelo)$sigma,
n=length(summary(modelo)$residuals))
hist(normal, col="green", freq=F)
lines(density(residuos), col="blue", lw=3)

boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")
```

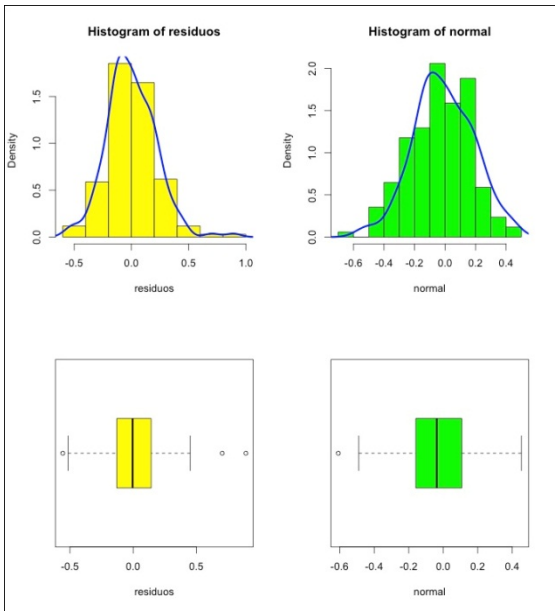


Figura 3.3: Histograma y Caja y bigotes de residuos y distribución normal de nuestros datos

```
# ¿A que conclusión llegamos?

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos,"pnorm", mean(residuos), sd(residuos))

One-sample Kolmogorov-Smirnov test

data:  residuos
D = 0.05573, p-value = 0.6666
alternative hypothesis: two-sided

# Alternativamente conviene usar la modificación de Lilliefors a este test
# esta corrección considera que los parámetros son estimados, a diferencia del ks "a secas"
library(nortest) # antes hay que instalar el paquete con install.packages("nortest")
lillie.test(residuos)

Lilliefors (Kolmogorov-Smirnov) normality test

data:  residuos
D = 0.05573, p-value = 0.2213

# Test de Shapiro-Wilk para evaluar normalidad
# n debe ser menor a 2000 y no es muy útil cuando tenemos datos repetidos...
shapiro.test(residuos)

Shapiro-Wilk normality test

data:  residuos
W = 0.9787, p-value = 0.01026

# ¿Cuáles son las hipótesis que se están evaluando en los tres casos anteriores?
```

```
# El famoso "Q-Q plot"
par(mfrow=c(1,1))
qqnorm(residuos)
qqline(residuos)
# también se puede hacer para los residuos estandarizados según transformación z.
plot(modelo, which = c(2))

# Q-Q plot paso a paso:
n=length(datos[,2])
pval=(1:n)/(n+1)
pval # pval representa la probabilidad acumulada desde 1 (=0.0058) país
# hasta la suma de 170 países (= 0.994)
# si dividimos por n en lugar de (n+1) el último valor de nuestra muestra tendrá una
# probabilidad = 1, pero en una distribución normal este valor sería +infinito

# ¿Cuáles son los cuantiles de una normal para las mismas probabilidades?
cuantiles.normal=qnorm(pval)
cuantiles.normal

# representamos los cuantiles observados en función de lo esperado bajo una
# distribución normal
plot(cuantiles.normal,sort(residuos))
# es fundamental ordenar los residuos de menor a mayor.
# Este gráfico es practicamente igual al qqnorm(residuos) que vimos más arriba
# comparemos:
layout(matrix(1:2,2,1))
qqnorm(residuos)
plot(cuantiles.normal,sort(residuos))

# otra opción muy común es hacer lo mismo con los residuos luego de realizar
# una transformación z. Se los llama residuos estandarizados,
# pero "estandarizado" se usa como nombre genérico para otras transformaciones,
# mejor entonces siempre hacer explícito cual estandarización realizamos
observados.z =  (residuos - mean(residuos)) /  sd(residuos)
plot(cuantiles.normal,sort(observados.z)) # Los cuantiles de los residuos
# observados están en una escala "z" y los estamos comparando con los
# cuantiles de una distribución normal en una escala "z". Entonces, si
# los residuos se distribuyen normalmente, deberían seguir una recta
# con ordenada al origen = 0  y pendiente = 1.
abline(a=0,b=1, col="red", lw=2)

# este gráfico es practicamente igual que habíamos hecho anteriormente:
plot(modelo, which = c(2))

#### SÓLO PARA FANÁTICOS DEL Q-Q PLOT
# para algunas personas es poco preciso utilizar simplemente los residuos observados
ordenados
# y prefieren realizar una corrección a los mismos a partir de "pval"
cuantiles.corregidos=quantile(residuos,probs=pval)
cuantiles.corregidos # son los cuantiles de la distribución acumulada.
```

```
# Son los valores de los residuos expresados en log10(cv) para los cuáles
# se acumula una determinada probabilidad.
# Es decir que en lugar de usar los residuos "reales",
# usamos una función "quantile" que estima los residuos para cada "pval".
# Se pueden elegir entre 9 métodos (algoritmos) para estimar estos "residuos según
pval"
# a partir de los "residuos reales". Si no se elige un método (algoritmo), R utiliza el
# método 7 (ver "?quantile"). No es importante para nuestro curso saber las diferencias
entre
# los 9 métodos, sino solamente comprender que se usan los residuos reales para
interpolar
# con algún método valores que correspondan a los de pval.
# Entonces comparemos:
layout(matrix(1:3,3,1))
plot(cuantiles.normal,sort(residuos), main="Simplemente residuos observados
ordenados")
plot(cuantiles.normal,cuantiles.corregidos, main="Residuos corrregidos por la función
quantile")
qqnorm(residuos)
# Se puede hacer lo mismo con los estandarizados:
cuantiles.corr.z=quantile(observados.z,probs=pval)
# Queda de ejercicio para ustedes la comparación.
# AQUÍ TERMINA LA SECCIÓN PARA FANÁTICOS DEL Q-Q PLOT
# Y LO QUE SIGUE ES IMPORTANTE PARA TODOS LOS ESTUDIANTES

# ¿Cómo afecta la curtosis y el muestreo?
par(mfcol=c(2,4))

hist(residuos, main="Residuos REALES")
qqnorm(residuos, main="")
qqline(residuos)

normal=rnorm(n=170, mean=0, sd=1)
hist(normal,main="Distribución normal")
qqnorm(normal, main="")
qqline(normal)

t.student=rt(n=170, df=2)
hist(t.student,main="Mayor curtosis")
qqnorm(t.student, main="")
qqline(t.student)

uniforme=runif(n=170, min=0, max=1)
hist(uniforme,main="Menor curtosis")
qqnorm(uniforme, main="")
qqline(uniforme)
```

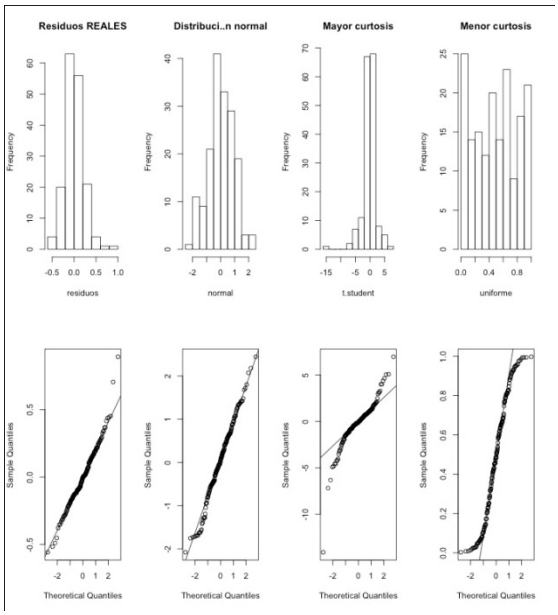


Figura 3.4: Q-Q Plot según residuos, distribución normal y diferentes niveles de curtosis

```
# install.packages("moments") # Si no instalaron el paquete
library(moments)
skewness(residuos) # debería dar cero
```

```
[1] 0.5017823
```

```
kurtosis(residuos) # debería dar tres
```

```
[1] 4.529746
```

```
# el mundo no es perfecto...
# probemos
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.
```

3.5.3 Observaciones atípicas, gran leverage e influyentes

```
# 4.3) OBSERVACIONES ATÍPICAS, GRAN LEVERAGE, INFLUYENTES ####
# Leer sección 14.9 del libro de Anderson et al.
# Muchas veces la violación a los supuestos anteriormente mencionados
# puede darse por la presencia de este tipo de observaciones.
# Por lo tanto, el análisis de los supuestos incluye,
# de modo importante, la evaluación de observaciones
# atípicas y (o), con gran leverage y (o) influeyentes.
# En principio estas observaciones pueden detectarse
# a partir de todos los gráficos de puntos (cada punto una observación)
# realizados en las secciones 4.1) y 4.2).
# A continuación definimos brevemente cada término y explicamos
# algunos estadísticos comunes:
# A)  ATÍPICAS (en inglés "OUTLIERS"):
# aquellas alejadas, raras y extremas en el eje y
```

```
# B) con gran "LEVERAGE":
# aquellas alejadas, raras y extremas en el eje x.
# OJO, Anderson et al traducen este término incorrectamente como "influencia"

# Para medir Leverage generalmente se usa el estadístico "hi".
# Hay un "hi" para cada observación y su valor mínimo es 0
# (en realidad 1/n en los modelos con ordenada al origen) y
# el valor máximo es 1.
# La suma de todos los hi es igual al número de coeficientes en
# el modelo, incluyendo la ordenada al origen.
# Entonces en la regresión lineal simple este valor es 2.

# hi =   1 / n   +   (xi -xmedia)^2 / sum((xi -xmedia)^2)

# C) y por lo tanto con gran "INFLUENCIA" en el análisis.
# Una observación que es ambas, atípica y con gran leverage,
# genera una gran influencia en el análisis.
# Para evaluar esto, un método básico es hacer el análisis con y sin la observación
# que se quiere evaluar. Esto es simplemente para dectectar si
# esa observación modifica de manera importante el análisis.
# Si los coeficientes cambian mucho se dice que la observación
# tiene gran "influencia".
# Una medida común de influencia es la "Cook's distance".
# Al identificar que existe una observación influyente la misma
# NO DEBE SER REMOVIDA DEL ANÁLISIS de manera AUTOMÁTICA.
# Discutir las posibles causas para este tipo de
# observaciones y las posibles soluciones,
# así como sus relaciones de compromiso.

# La función "plot" de R es útil porque da 4 gráficos que sirven
# para evaluar varios supuestos incluyendo estos aspectos que acabamos de discutir.
par(mfcol=c(1,1))
plot(modelo)
```

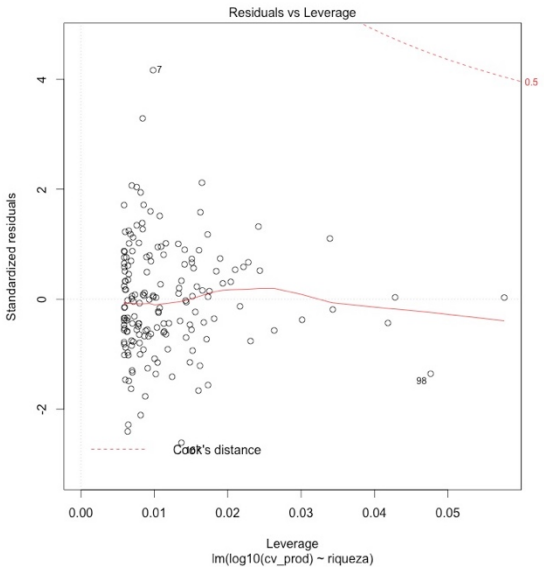


Figura 3.5: Residuos estandarizados en función del "leverage" de los datos observados del modelo log10 del CV de producción en función de Riqueza

```
# La observación 7 podría tener una alta influencia. Veamos
# que sucede cuando la removemos del análisis
?update
modelo2=update(modelo, subset=-c(7))
library(car)
compareCoefs(modelo,modelo2)
```

```
Call:
1: lm(formula = log10(cv_prod) ~ riqueza, data = datos)
2: lm(formula = log10(cv_prod) ~ riqueza, data = datos, subset = -c(7))
               Est. 1      SE 1    Est. 2      SE 2
(Intercept)  1.320205  0.033727  1.318944  0.033982
riqueza     -0.004469  0.000802 -0.004422  0.000814
```

```
# Concluir.
# Recordar que sacamos la observación 7 sólamente para evaluar su
# influencia y no esperamos sacarla del análisis.

# saquemos ahora además la observación 167
modelo3=update(modelo, subset=-c(7,167))
compareCoefs(modelo,modelo2, modelo3)
```

```
Call:
1: lm(formula = log10(cv_prod) ~ riqueza, data = datos)
2: lm(formula = log10(cv_prod) ~ riqueza, data = datos, subset = -c(7))
3: lm(formula = log10(cv_prod) ~ riqueza, data = datos, subset = -c(7, 167))
               Est. 1      SE 1    Est. 2      SE 2    Est. 3      SE 3
(Intercept)  1.320205  0.033727  1.318944  0.033982  1.320548  0.034385
riqueza     -0.004469  0.000802 -0.004422  0.000814 -0.004454  0.000821
```

```
# Los aspectos discutidos en el punto 4.3) se aplican tanto
# a modelos lineales generalES como a modelos lineales generalIZADOS.
# Al final del curso veremos los modelos lineales generalizados.
```

3.5.4 En el examen

```
# En resumen, hemos visto varias formas de evaluar los supuestos.
# Esto es importante para comprender cada uno de los supuestos.
# En el examen no es necesario que repitan absolutamente todas las formas
# que vimos. Se espera en el examen que listen cada uno de los supuestos y
# que utilicen al menos un gráfico relevante para cada supuesto (esto ya lo pueden
hacer)
# y además algún test inferencial para cada supuesto (por ahora vimos pruebas
sólamente para
# evaluar normalidad pero más adelante veremos otras).
# En algunos exámenes nosotros daremos algún gráfico en particular
# y ustedes tendrán que interpretarlo. De este modo, no es
# necesario usar todos los gráficos en un examen pero si es
# necesario comprenderlos todos.
```

3.6 ANOVA

```
# Filosofía "Frecuentista"
summary(modelo)
```

```
Call:
lm(formula = log10(cv_prod) ~ riqueza, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5572 -0.1287 -0.0047  0.1416  0.8912

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.3202047  0.0337274  39.143  < 2e-16 ***
riqueza     -0.0044694  0.0008017  -5.575  9.7e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.215 on 168 degrees of freedom
Multiple R-squared:  0.1561,    Adjusted R-squared:  0.1511
F-statistic: 31.08 on 1 and 168 DF,  p-value: 9.704e-08
```

```
# Interpretar el "F-statistic: 31.08" en términos del problema

# para obtener la famosa tabla de ANOVA...
anova(modelo)
```

```
Analysis of Variance Table

Response: log10(cv_prod)
           Df Sum Sq Mean Sq F value    Pr(>F)
riqueza     1 1.4364 1.43641  31.079 9.704e-08 ***
Residuals 168 7.7647 0.04622
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.7 Selección de modelos por AIC

```
# Teoría de la información
# Esta sección es a modo de discución para aquellos que ya han leído sobre el tema.
# No hemos explicado en el curso el fundamento de este análisis.

# Para que aquellos que aún no saben del tema, lean esto recién
# luego de que hayamos explicado AIC en el curso (faltan varias semanas todavía)
# Por ahora entonces pasen a la siguiente sección y realicen el EJERCICIO propuesto
# Lean también el EJEMPLO ADICIONAL

# Comparar con resultados ANOVA
library("MuMIn") # Recordar instalar el paquete antes con install.packages("MuMIn")

modelo=update(modelo, na.action="na.fail")
selec<-dredge(modelo)
# IMPORTANTE: la función dredge genera todos los modelos posibles
# (incluido el modelo nulo que solo tiene a la ordenada al origen) y los compara
ajustados por ML.
nrow(selec) # Contamos la cantidad de filas que tiene la tabla "selec"
```

```
[1] 2
```

```
# La función dredge generó 2 modelos
selec
```

```
Global model call: lm(formula = log10(cv_prod) ~ riqueza, data = datos,
na.action = "na.fail")
---
Model selection table
  (Intrc)     riquz df logLik  AICc delta weight
2   1.320 -0.004469  3 21.109 -36.1  0.00      1
1   1.156            2  6.681  -9.3 26.78      0
Models ranked by AICc(x)
```

```
# La salida es una tabla ordenada de los modelos del mejor al peor ajuste según AIC.
# La tabla contiene: la estimación de los predictores continuos
# y un signo "+" para los predictores categóricos incluidos en el modelo (en este caso
ninguno).
# Además, muestra la cantidad de parámetros (df), el logLik, el AICc,
# el delta (la diferencia respecto del mejor modelo) y el weight (peso relativo).
```

3.8 Ejercicio

```
# Evaluar los supuestos de un modelo de regresión lineal simple
# estimado para cv_prod (SIN log10) en función de la riqueza.
# Concluya.
```

3.9 Ejemplo adicional

```
# comparamos la asociación de la riqueza con cv_prod vs. con crec_prod
layout(matrix(1:2,2,1))

with(datos, plot(riqueza, crec_prod, bty="l",
            xlab="Riqueza (no. cultivos)",ylab="Crecimiento producción"))
with(datos,abline(lm(crec_prod~riqueza),lwd=2))


with(datos,plot(riqueza,log10(cv_prod),bty="l", axes="F",
           xlab="Riqueza (no. cultivos)",ylab="CV producción (%)"))
with(datos,abline(lm(log10(cv_prod)~riqueza),lwd=2))
axis(1,at=c(0,20,40,60,80,100),lab=c(0,20,40,60,80,100))
axis(2,at=c(0.6,1.1,1.6,2.1),lab=c(4,13,40,126))


# vemos los valores estimados de los parámetros
# y su significancia para el
# modelo de crecimiento de producción
mod2=lm(log10(crec_prod)~riqueza,data=datos)
summary(mod2)

Call:
lm(formula = log10(crec_prod) ~ riqueza, data = datos)

Residuals:
      Min        1Q    Median        3Q       Max
-0.042684 -0.005065 -0.000813  0.004321  0.054401

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 9.527e-03  1.694e-03   5.626 7.58e-08 ***
riqueza     4.622e-05  4.026e-05   1.148    0.253
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0108 on 168 degrees of freedom
Multiple R-squared:  0.007783,   Adjusted R-squared:  0.001877
F-statistic: 1.318 on 1 and 168 DF,  p-value: 0.2526

summary(modelo) # este es el modelo de log10(cv_prod),

Call:
lm(formula = log10(cv_prod) ~ riqueza, data = datos, na.action = "na.fail")

Residuals:
    Min      1Q  Median      3Q     Max
-0.5572 -0.1287 -0.0047  0.1416  0.8912

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.3202047  0.0337274  39.143  < 2e-16 ***
riqueza     -0.0044694  0.0008017  -5.575  9.7e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.215 on 168 degrees of freedom
Multiple R-squared:  0.1561,     Adjusted R-squared:  0.1511
F-statistic: 31.08 on 1 and 168 DF,  p-value: 9.704e-08

# comparar con el anterior y
# discutir las diferencias entre los dos modelos evaluados.
```

Volver al Índice principal

Análisis de la varianza (ANOVA) y comparaciones múltiples

Trabajo Práctico Nº 4: Análisis de la varianza (ANOVA) y Comparaciones Múltiples

4.1 Problema y datos

```
# A continuación se observa el número de empleados públicos cada 1000 habitantes en
# las provincias Argentinas.
# Datos del ministerio de economía:
# http://www.mecon.gov.ar/peconomica/basehome/fichas_provinciales.htm
empleados=c(105, 103, 63, 67, 92, 54, 42, 51, 45, 74, 62, 37, 43, 88, 56, 30, 35, 33, 55,
 46, 45, 47,44, 82)
region=c("Patagonia", "Patagonia", "Patagonia","Patagonia", "Patagonia", "Patagonia",
 "NEA", "NEA", "NEA", "NEA", "NOA", "NOA", "NOA", "NOA", "NOA", "Centro y
 Bs.As","Centro y Bs.As", "Centro y Bs.As","Centro y Bs.As","Centro y Bs.As",
 "Nuevo Cuyo","Nuevo Cuyo","Nuevo Cuyo","Nuevo Cuyo")
provincia=c("Tierra del Fuego", "Santa Cruz", "Chubut", "Río Negro", "Neuquén", "La
 Pampa", "Misiones", "Corrientes", "Chaco", "Formosa", "Jujuy", "Salta", "Tucuman",
 "Catamarca", "Santiago del Estero", "Córdoba", "Buenos Aires", "Santa Fe", "Entre
 Ríos", "CABA", "Mendoza", "San Luis", "San Juan", "La Rioja")
datos = data.frame(empleados,region,provincia)

# Queremos estudiar si la cantidad de empleados públicos por 1000 habitantes varía
 según regiones

# Región es una variable categórica.
```

4.2 Sobre factores en R

```
# Cuando las variables categóricas son predictoras (independientes) en estadística
# se las llama "factores".
# A las categorías dentro de un factor se las llama "niveles"
# R también usa esta terminología... ver...
str(datos)
# R nos indica que región es un factor con 5 niveles
datos$region
# o también interesante
class(datos$region)
levels(datos$region) # R ordena los niveles alfabéticamente.
```

4.2.1 Cambiar orden

```
# Algo que muchas veces queremos hacer es cambiar el orden de los niveles,
# ya que este mismo orden es tomado luego por la función "lm" y ubica
# al primer nivel como ordenada (ver sección MODELO).
# Entonces, si queremos cambiar el orden, supongamos de norte a sur:
datos$region=with(datos, factor(region, levels=c("NOA", "NEA", "Nuevo Cuyo",
 "Centro y Bs.As", "Patagonia")))
# Vemos como cambia el orden:
levels(datos$region)
# Volvemos a ordenar alfabéticamente
datos$region=with(datos, factor(region, levels=c("Centro y Bs.As", "NEA",
 "NOA","Nuevo Cuyo", "Patagonia")))
levels(datos$region)
```

4.2.2 Cambiar nombres

```
# Finalmente puede resultarnos interesante cambiar los nombres de las clases.
# Por ejemplo por nombres más cortos, para ello usamos el argumento "labels"
# dentro de la función "factor":
datos$region=with(datos, factor(region, labels=c("Centro", "NEA", "NOA","Cuyo",
 "Pat")))
levels(datos$region)
# En la sección MODELO explicamos como hacer algo similar con la función "relevel"
```

4.2.3 Caracteres vs. factores

```
# R reconoce a datos$región como un factor porque está dentro de una tabla de datos.
# En cambio al vector "región" que cargamos al principio del script lo reconoce
# como un vector de caracteres.
str(region)
# una de las diferencias es que R no reconoce los niveles cuando la clase es vector de
 caracteres
levels(region)
# en cambio si lo transformamos en factor si los reconoce
region = as.factor(region)
levels(region)
# Solo un detalle para tener en cuenta. Casi siempre guardamos los datos
```

```
# en tablas de datos ("data.frames") entonces R siempre va a reconocer a nuestras
 variables
# categóricas como factores con sus niveles.
```

4.3 Modelo y ANOVA

```
# Volvamos al problema.
# Recordemos que queremos estudiar si la cantidad de empleados públicos por 1000
 habitantes
# varía según regiones.
# Plantear el modelo estadístico y evaluarlo:
modelo=lm(empleados~region, data=datos)
modelo
summary(modelo)
# Interpretar el "F-statistic: 3.813" en términos del problema

summary(modelo)$fstatistic
```

```
   value     numdf     dendf
3.813321  4.000000 19.000000
```

```
# ¿Que son numdf y dendf? ¿Cómo se obtienen?

anova(modelo)
```

```
Analysis of Variance Table

Response: empleados
          Df Sum Sq Mean Sq F value  Pr(>F)
region     4 4889.0 1222.26  3.8133 0.01937 *
Residuals 19 6089.9  320.52
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ¿Qué unidades tiene el CME?
# Las secciones 13.1 y 13.2 del libro de Anderson et al. explica de manera
# clara los fundamentos del ANOVA para un ejemplo similar.

# Otra opción si queremos cambiar el nivel que esta como base en R es
# usando la función relevel. Por ejemplo si queremos usar como base "Patagonia".
datos$region=relevel(datos$region, ref="Pat")
modelo2=update(modelo)
summary(modelo2)
```

4.4 Comparaciones múltiples

```
install.packages("agricolae")
library(agricolae)
# vamos a ver 3 tests, los ordenamos crecientemente en su potencia
# pero también crecientemente en la probabilidad de cometer un error de tipo 1.
```

```
# ¿Por qué realizamos comparaciones múltiples?
```

4.4.1 Bonferroni

```
LSD.test(modelo, trt="region", p.adj="bonferroni", alpha = 0.05, console=TRUE)
# Console = TRUE para que nos muestre el resultado en la pantalla de de R.

Study: modelo ~ "region"

LSD t Test for empleados
P value adjustment method: bonferroni

Mean Square Error:  320.5228

region,  means and individual ( 95 %) CI

       empleados      std r      LCL      UCL Min Max
Centro  39.80000 10.42593 5 23.04214 56.55786  30  55
Cuyo    54.50000 18.37571 4 35.76414 73.23586  44  82
NEA     53.00000 14.49138 4 34.26414 71.73586  42  74
NOA     57.20000 19.89221 5 40.44214 73.95786  37  88
Pat     80.66667 22.04238 6 65.36890 95.96443  54 105

alpha: 0.05 ; Df Error: 19
Critical Value of t: 3.173725

Minimum difference changes for each comparison

Means with the same letter are not significantly different.

Groups, Treatments and means
a     Pat       80.67
ab    NOA       57.2
ab    Cuyo      54.5
ab    NEA       53
b     Centro    39.8

# ¿Que es el alpha?

# ¿Cuál es el alfa que utiliza el test de Bonferroni en cada
# contraste de pares de medias?

# ¿Cuál es el estadístico utilizado en estos contrastes?
```

4.4.2 Tukey

```
# HSD = Honestly significant difference
tukey=HSD.test(modelo,"region", console=TRUE)

Study: modelo ~ "region"

HSD Test for empleados

Mean Square Error:  320.5228
```

```
region,  means

       empleados      std r Min Max
Centro  39.80000 10.42593 5  30  55
Cuyo    54.50000 18.37571 4  44  82
NEA     53.00000 14.49138 4  42  74
NOA     57.20000 19.89221 5  37  88
Pat     80.66667 22.04238 6  54 105
alpha: 0.05 ; Df Error: 19
Critical Value of Studentized Range: 4.252831

Harmonic Mean of Cell Sizes  4.6875
Honestly Significant Difference: 35.16713

Means with the same letter are not significantly different.

Groups, Treatments and means
a     Pat       80.67
ab    NOA       57.2
ab    Cuyo      54.5
ab    NEA       53
b     Centro    39.8
```

```
# "Critical Value of Studentized Range" es el valor "q" de la fórmula 10.10
# del libro de Webster
# "q" sigue una distribución de rangos estudentizada
# HSD = q * sqrt((CME /repeticiones))
HSD = 4.252831 * sqrt((320.5228 / 4.6875) )
HSD

# "q" puede obtenerse también como:
qtukey(0.95,nmeans=5,df=19)

# "Harmonic Mean of Cell Sizes" es el número de repeticiones que se usa en
# la fórmula 10.10 de HSD, en este caso es 4.6875 porque los distintos "tratamientos"
# tienen distinto número de repeticiones.
HarmonicMean = 5 / (1/5 + 1/4 + 1/5 + 1/4 + 1/6)
HarmonicMean
```

```
[1] 4.6875
```

```
# El valor de la media armónica generalmente es menor al de la media aritmética.
```

4.4.3 LSD

```
LSD.test(modelo,"region", p.adj="none", alpha = 0.05, console=TRUE)
```

```
Study: modelo ~ "region"

LSD t Test for empleados

Mean Square Error:  320.5228

region,  means and individual ( 95 %) CI
```

```
       empleados      std r      LCL      UCL Min Max
Centro  39.80000 10.42593 5 23.04214 56.55786  30  55
Cuyo    54.50000 18.37571 4 35.76414 73.23586  44  82
NEA     53.00000 14.49138 4 34.26414 71.73586  42  74
NOA     57.20000 19.89221 5 40.44214 73.95786  37  88
Pat     80.66667 22.04238 6 65.36890 95.96443  54 105

alpha: 0.05 ; Df Error: 19
Critical Value of t: 2.093024

Minimum difference changes for each comparison

Means with the same letter are not significantly different.

Groups, Treatments and means
a     Pat       80.67
b     NOA       57.2
b     Cuyo      54.5
b     NEA       53
b     Centro    39.8
```

```
# fórmula 10.11 del libro de Webster
# LSD = sqrt (2*CME*F/repeticiones)
LSD = sqrt(2*320.5228*(2.093024^2) / 4.6875 )
LSD
# Cuando los grados de libertad del numerador de la F son = 1, la F = t^2
# por eso es que aquí se usa t^2 pero la ecuación es la misma que la fórmula 10.11 del
 Webster
# la única diferencia es que en la fórmula 10.11 usamos F y aquí usamos t^2.

# si utilizamos la ecuación 13.17 del libro de Anderson
# LSD = t * sqrt (CME*((1/repeticiones)^2))
LSD2= qt(0.025, df=19,lower.tail=F)*sqrt(320.5228*(1/4.6875+1/4.6875))
LSD2

# vemos que dan lo mismo
LSD; LSD2

# En la nueva versión del paquete "agricolae" en lugar de utilizar
# una diferencia mínima significativa para todas las comparaciones,
# utiliza comparaciones t de a pares, empleando las réplicas y los
# datos sólo de los dos tratamientos en cuestión. Por eso el
# análisis indica "Minimum difference changes for each comparison".
# Esta es una pequeña corrección a lo que plantean
# los libros de Webster y también Anderson. No necesariamente
# aceptada.

# En este caso indicamos en la función LSD.test el argumento
# p.adj="none" , es decir, que son comparaciones de t apareadas
# sin corregir por comparaciones múltiples sobre un mismo
# set de datos. De las tres pruebas por lo tanto esta sería
# la de mayor alfa global (pero menor error de tipo 2).
```

```
# Previamente hemos visto el método de ajuste por bonferroni
# con el argumento p.adj="bonferroni" pero hay varios
# otros métodos de ajuste que se podrían haber empleado. Ver:
?LSD.test
```

4.4.4 Aspectos claves sobre comparaciones múltiples

```
# A) cuando se realizan múltiples tests sobre los mismos datos
# el alfa global (es decir, la probabilidad de cometer al menos
# un error de tipo 1 en alguna de las pruebas) aumenta drásticamente.
# Esto es especialmente grave para estudios con muchos tratamientos
# y pocas repeticiones.

# B) reducir el alfa "particular" (de cada prueba)
# para mantener el alfa "global" en niveles aceptables
# aumenta el error de tipo 2.

# C) por lo tanto, debido a lo planteado en B) existen
# distintos métodos para corregir el alfa "particular".
# No hay un único método que sea el mejor para todas las situaciones.

# D) las comparaciones múltiples también se conocen como
# comparaciones "a posteriori" porque se realizan
# SOLAMENTE luego de encontrar efectos significativos en el ANOVA
```

4.5 Gráfico

```
medias=with(datos,tapply(empleados,region,mean))
medias=sort(medias) # ordenamos las medias para que luego el gráfico nos quede de
 menor a mayor
medias
desvios=with(datos,tapply(empleados,region,sd))
num=with(datos,tapply(empleados,region,length))
error=desvios / sqrt(num)

fig= barplot(medias,ylim=c(0,100),ylab= "Número de empleados cada 1000 hab")
arrows(fig,medias+error,fig,medias-error, angle=90,code=3)

# agregamos al gráfico los resultados del test de tukey
text(0.7,55,"a")
text(1.9,66,"ab")
text(3.1,69,"ab")
text(4.3,72,"ab")
text(5.5,95,"b")
```

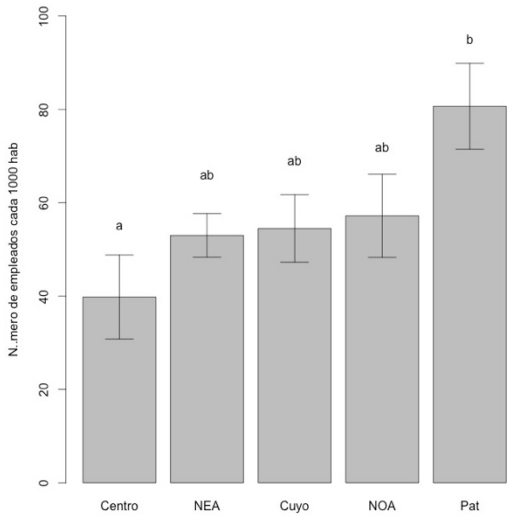


Figura 4.1: Histograma de promedio de empleados cada 1000 habitantes según provincia, con correspondientes desvíos estándar y resultados del test de Tukey

```
# similar y algo más fácil....
bar.group(tukey$groups,ylim=c(0,100),density=4,border="blue")
# nos gusta más el gráfico anterior porque presenta el error estándar para la media
# de cada tratamiento.
```

4.6 Supuestos

```
# evaluamos los supuestos del modelo a través de los residuos
residuos=resid(modelo)

# INDEPENDENCIA, HOMOGENEIDAD, LINEALIDAD
# para evaluar el supuesto de homogeneidad de varianzas,
# así como el de independencia
predichos=fitted(modelo)
plot(predichos,residuos)
# Indicar las unidades de los residuos y los valores predichos.
# ¿Que son los valores predichos?

library(car)
residualPlots(modelo)
# más adelante veremos que son los "Pearson" residuals,
# por ahora pensémoslos solamente como residuales.

# NORMALIDAD
par(mfrow=c(2,1))
hist(residuos, col="yellow")
boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# El punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior.
# "bty" por "box type"
```

```
# Comparamos con una distribución normal de igual
# desvío estándar residual a nuestros datos
par(mfrow=c(2,2))
hist(residuos, col="yellow", freq=F)
lines(density(residuos), col="blue", lw=3)

normal=rnorm(mean=0, sd=summary(modelo)$sigma,
 n=length(summary(modelo)$residuals))
hist(normal, col="green", freq=F)
lines(density(residuos), col="blue", lw=3)

boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")
# ¿A que conclusión llegamos?


par(mfrow=c(2,1))
qqnorm(residuos)
qqline(residuos)
# comparemos con una normal
qqnorm(normal)
qqline(normal)

par(mfrow=c(1,1))
# también se puede hacer para los residuos estandarizados según transformación z.
plot(modelo, which = c(2))


# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos,"pnorm",mean(residuos),sd(residuos))

# alternativamente conviene usar la modificación de Lilliefors a este test
# esta corrección considera que los parámetros son estimados, a diferencia del ks "a
 secas"
library(nortest) #antes hay que instalar el paquete
lillie.test(residuos)

# Test de Shapiro-Wilk para evaluar normalidad
# n debe ser menor a 2000 y no es muy útil cuando tenemos datos repetidos...
shapiro.test(residuos)


# si hay tiempo y quieren obtener un 10 en su examen también hacen lo siguiente:
# install.packages("moments") # Si no instalaron el paquete
library(moments)
skewness(residuos) # debería dar cero
kurtosis(residuos) # debería dar tres
# el mundo no es perfecto...
# ¿Cómo afecta la curtosis y el muestreo?
par(mfcol=c(2,4))
```

```
hist(residuos, main="Residuos REALES")
qqnorm(residuos, main="")
qqline(residuos)

normal=rnorm(n=24, mean=0, sd=1)
hist(normal,main="Distribución normal")
qqnorm(normal, main="")
qqline(normal)

t.student=rt(n=24, df=2)
hist(t.student,main="Mayor curtosis")
qqnorm(t.student, main="")
qqline(t.student)

uniforme=runif(n=24, min=0, max=1)
hist(uniforme,main="Menor curtosis")
qqnorm(uniforme, main="")
qqline(uniforme)
par(mfcol=c(1,1))
plot(modelo)
# Recordar que la parte de leverage está explicada
# en las páginas 616-618 del libro de Anderson.
# Recordar también que leverage está incorrectamente traducido como "influencia"
```

4.7 ¿Cuántas repeticiones necesito?

```
# La ecuación de HSD contiene el número de repeticiones por tratamiento:
#       HSD = q * sqrt((CME /repeticiones))

# Despejando "repeticiones", obtenemos:
#       repeticiones = CME * (q / HSD)^2
# Reemplazando en esta ecuación podemos tener una idea del número de repeticiones
# que necesitamos previo al experimento.

# para el ejemplo que venimos discutiendo:
CME = ( sum(residuos^2) ) / (length(residuos) - 5)
CME # las unidades son (empleados cada mil hab)^2

# En clases anteriores habíamos usado "qnorm" para obtener el cuantil de
# la distribución normal para una determinada probabilidad acumulada.
# Ahora, para obtener "q", hacemos lo mismo pero con la distribución de
# rangos estudentizados (o distribución tukey):
q = qtukey(0.95,nmeans=5,df=19)
q
# el primer valor es la confianza,
# el segundo el número de tratamientos
# y el tercero son los grados de libertad del error.

# el HSD lo definimos nosotros:
```

```
HSD=20

# ahora sí:
repeticiones = CME * (q / HSD)^2
repeticiones
# De manera importante, si bien es otra ecuación,
# aparecen los mismos conceptos que discutimos en Estadística 1
# respecto de cuáles aspectos son importantes a considerar
# para obtener un número de repeticiones adecuado:
# .- Variabilidad (en este caso residual: CME)
# .- Confianza (influye sobre "q")
# .- Precisión (HSD)
# Indicar dónde se encuentran estos tres aspectos en la
# ecuación de t de student que usábamos en Estadística 1 para
# obtener el número de repeticiones.
# Comparar, indicar similitudes y diferencias.
```

4.8 Sobre probabilidades e inferencia

```
# Todo lo ejercitado es de gran utilidad para
# inferir y modelar aspectos de una población a partir de
# de una muestra.
# Pero en el caso particular del ejemplo tratado:
# ¿Cuál es la población? ¿Cuál es la muestra?
# ¿Existe incertidumbre? ¿A qué se refieren las
# probabilidades obtenidas?
```

4.9 Función de verosimilitud y matrices

```
# Vuelvan sobre este punto más adelante cuando veamos las funciones
# de verosimilitud y la notación matricial (segunda mitad del curso).
# Las consignas son:
# 1) plantear la función de verosimilitud de acuerdo al modelo estimado arriba.
# 2) escribir matricialmente el modelo estimado arriba.
# 3) indicar y explicar el método por el cual se estiman los parámetros.
```

Volver al Índice principal

Diseño de experimentos (muestreos)

Trabajo Práctico Nº 5: Diseño en bloques completos aleatorizados (DBCA)

5.1 Problema y datos

```
# Se desea estudiar los beneficios obtenidos por las empresas más importantes del
 globo. Se piensa que el beneficio
# promedio depende del sector en el que se desempeña cada empresa, dado que sectores
 distintos poseen diferencias
# en sus estructuras de costos y en las posibilidades de retener rentas extraordinarias.
 Para llevar adelante
# el estudio se tomaron empresas de hasta 100.000 empleados, relevando datos de
 empresas de 5 de los sectores más
# significativos de toda economía, cuantificando el beneficio generado por ellas en 1986
 en millones de USD de 1986.
# Como el tamaño de la empresa es sin duda un factor determinante a la hora de
 considerar los beneficios que genera,
# para cada sector se ha elegido una empresa entre 0 y 5.000 empleados,otra entre 5 y
 20 mil empleados y otra entre
# de 20 a 100 mil. De este modo el estudio también consideró el efecto del tamaño de la
 empresa sobre el beneficio,
# pero el principal objetivo fue evaluar el efecto del sector sobre el beneficio.

# Los datos provienen de "Forbes 500 list", pertenecen al año 1986 y pueden ser
 encontrados en:
# http://lib.stat.cmu.edu/DASL/Datafiles/Companies.html

datos = read.table("datos_p_5.txt", header =T, dec=",")
str(datos)
```

5.2 Consignas a resolver

```
# 1) En papel, indique el modelo conceptual y el modelo estadístico acorde (en
 cualquiera de las dos versiones
# presentadas en el curso). Identifique la unidad experimental y los tratamientos.

# 2) ¿Por qué se incluyó el factor "tamaño de empresa" en el estudio? ¿Cuáles son las
 ventajas de esta inclusión?
# ¿Podría tener desventajas haber incluido este factor? ¿Cuáles? Indique las
 consecuencias de dicha elección sobre
# el cociente F.

# 3) Interprete algún valor del cociente F en términos del problema. Indique su
 relevancia y la hipótesis asociada.
# ¿Cómo varía el valor-p a medida que aumenta el valor del cociente F?

# 4) Indique V o F y justifique:
# a- Que las empresas grandes tengan beneficios mayores en el sector de energía refleja
 la
# existencia de interacción entre los factores y por ende invalida el modelo previamente
 planteado.

# b- Un diseño en bloques completos aleatorizados implica una "aleatorización
 restringida".

# c- Es un experimento mensurativo porque se asignan aleatoriamente los tratamientos a
 las unidades experimentales.

# d- La asignación (o selección) aleatoria de las unidades experimentales es
 fundamental para (pero no garantiza) lograr
# independencia en los residuos y evitar efectos confundidos.
```

5.3 Supuesto de ausencia de interacción entre los efectos de bloques y tratamientos

```
# 5) Evalue los supuestos del modelo y análisis desarrollados.
# ¡IMPORTANTE! NUEVO SUPUESTO: ausencia de interacción entre los bloques y
 los tratamientos
with(datos, interaction.plot(sector,nro.emp,Profits))
```

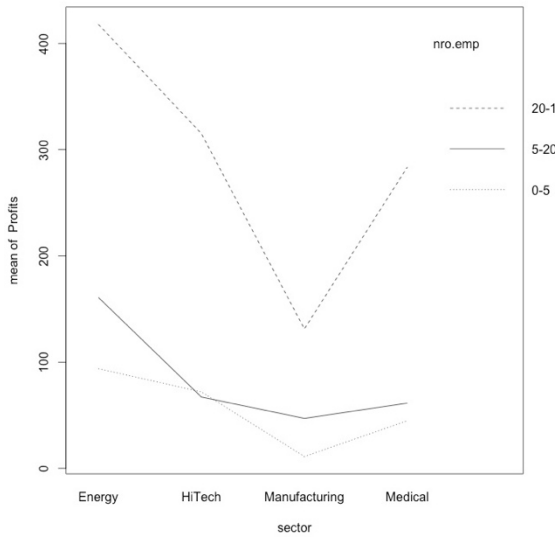


Figura 5.1: Media de beneficios de las empresas según el sector en el que se desempeñan y cantidad de empleados con los que operan, como "bloque".

```
# Interpretar el gráfico.
# Evaluar los otros supuestos por su cuenta y consultar con los docentes.
```

5.4 Más consignas

```
# 6) Interprete el coeficiente de determinación.
# ¿Puede haber modelos adecuados pero con bajo coeficiente de determinación?
 Justifique

# 7) Plantear dos modelos interesantes en el contexto del problema
# y compararlos por AIC
```

5.5 Otro ejemplo DBCA

```
# Datos ejercicio 10 página 297 del libro de Webster
# Una empresa carbonífera en West Virginia analizó la producción promedio de tres
 minas.
# Cuatro grupos de empleados trabajaron en cada mina y se registró en toneladas la
 producccción de
# carbón resultante por día. Se utilizó un modelo
# con dos factores considerando a cada "grupo" como un bloque. Como nuevo
 supervisor administrativo, usted
# debe determinar si existe alguna diferencia en la productividad promedio de las minas.
carbon=c(42.7,47.1,32.1,29.2,
        54.1, 59.2, 53.1, 41.1,
        56.9, 59.2, 58.7, 49.2)
mina=c(1,1,1,1,2,2,2,2,3,3,3,3)
mina=as.factor(mina)
grupo=c(1,2,3,4,1,2,3,4,1,2,3,4)
grupo=as.factor(grupo)
datos=data.frame(carbon,grupo,mina)
```

```
# 1) escribir el modelo y estimar sus parámetros
modelo=with(datos, lm(carbon~mina + grupo))
modelo
summary(modelo)

# 2) evaluar las hipótesis de interés
anova(modelo)

# install.packages("agricolae")
library(agricolae)

HSD.test(modelo,"mina", console=TRUE)

# 3) Interpretar los resultados de la siguiente función y relacionar con los
# objetivos del estudio
confint(modelo, level = 0.99)
# o más lindo y completo
cbind(Estimado=coef(modelo),confint(modelo, level = 0.99))

# ¿Cuáles son la diferencias con el siguiente resultado?
# ¿Cuáles son los parámetros estimados en cada caso?
predict.lm(modelo, interval="confidence")

# 4) Si la unidad experimental es cada día, y se elijen días al azar para cuantificar la
# producción. ¿Cómo cambiaría el diseño si se realiza según DCA en lugar de DBCA
 (bajo
# el mismo objetivo: evaluar diferencias entre las minas)? Utilizar el término
# "aleatorización restringida" en su respuesta.

# 5) ¿En que consiste el experimento? ¿En que consiste la muestra?

# 6) Dado lo indicado en 3), este es un experimento mensurativo o manipulativo?
 Justifique

# 7) El análisis continúa con la evaluación de los supuestos
# ¡IMPORTANTE! NUEVO SUPUESTO: ausencia de interacción entre los bloques y
 los tratamientos
with(datos, interaction.plot(mina,grupo,carbon))
```

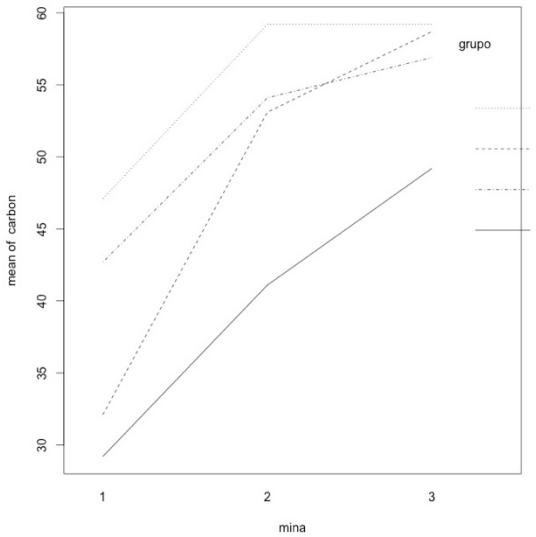


Figura 5.2: Producción de carbón (en toneladas diarias) en función de 3 minas diferentes, con 4 grupos de trabajadores como "bloque".

Interpretar el gráfico.

Evaluar los otros supuestos por su cuenta y consultar con los docentes.

Trabajo Práctico Nº 6: Diseño multi-factorial

6.1 Problema y datos

Los siguientes datos fueron obtenidos de CEPAL: Comisión Económica para América
Latina y el Caribe, División de Estadística y Proyecciones Económicas,
Unidad de Estadísticas Sociales, sobre la base de tabulaciones especiales de las
encuestas de hogares de los respectivos países.
ver http://www.eclac.org/

La variable dependiente es el Porcentaje del salario medio de las mujeres asalariadas urbanas,
con respecto al salario de los hombres de iguales características.
Se utilizó en el cálculo hombres y mujeres de 20 a 49 años de edad, que trabajan 35 horas y más por semana,

```
datos=read.csv("datos_p_6.csv", sep=",")
str(datos)
```

Para los que trabajan con Windows:
Es posible que a tu Windows no le gustén los acentos y te haya cambiado
de nombre los títulos de las columnas que tienen acentos. Eso se
corrige relativamente fácil renombrando las columnas de las tablas de datos

```
colnames(datos)=c("año", "escolaridad", "remuneración")
```

Dentro del factor escolaridad hay un nivel que se llama "13 y más".

```
# Tal vez tu Windows le cambió el nombre a este nivel también.
# Revisar los códigos que dimos en scripts anteriores respecto de cómo
# renombrar niveles de factores.

# Se desea conocer cómo varía el porcentaje del salario de las mujeres en función de la
 escolaridad y
# el año en el cual se realizó la encuesta. Ambas variables expresadas como variables
 categóricas.
```

6.2 Gráficos

```
levels(datos$escolaridad)
# Los ordenamos de menor a mayor para que quede más lindo el gráfico
datos$escolaridad=with(datos, factor(escolaridad, levels=c("0_5", "6_9", "10_12",
                                    "13 y más")))
# Un gráfico interesante en el contexto del problema
with(datos, interaction.plot(escolaridad,año,remuneración,
                        xlab="Escolaridad (años)",
                        ylab="Remuneración mujeres (%)",
                       ylim=range(remuneración),
                        type="b", pch=c(1,16),
                        cex=2, leg.bty="o", bty="l"))
# el type="b" nos da un símbolo en el centro
# mientras que el pch=c(1,16) nos dice que esos símbolos son
# círculos vacíos y negros según el año.
# cex=2 nos da el tamaño de los símbolos en el gráfico.
# leg.bty="o" le agrega el cuadro a la leyenda.
# bty="l" es el box type para el gráfico.

# Si además de las medias queremos ver los valores
# para cada unidad experimental. Entonces al gráfico anterior
# le agregamos los puntos con la función "points"   :-)
with(datos, points(jitter(as.numeric(escolaridad),factor=0.5),remuneración,
    pch=ifelse(año == "98-99", 16, 1), cex=0.8))
```

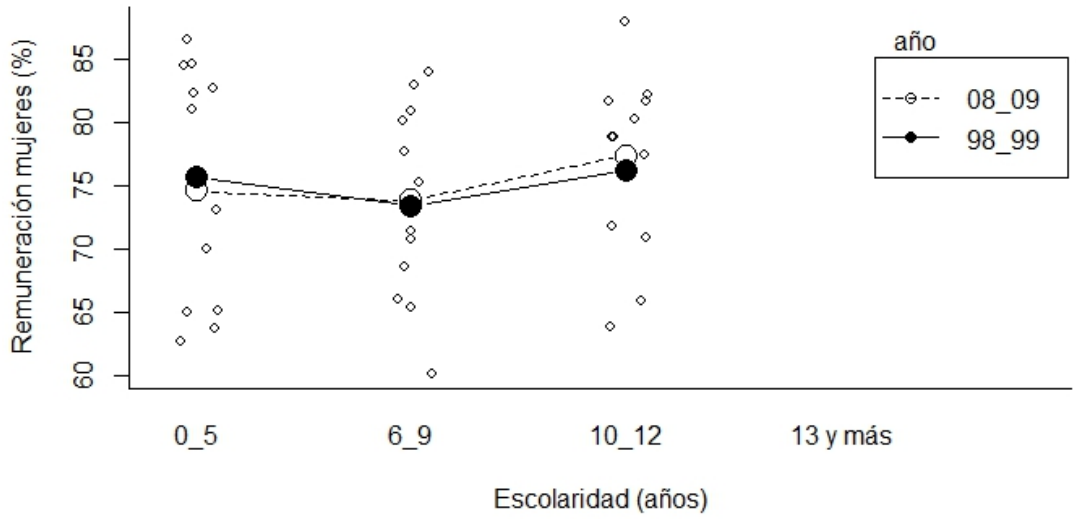


Figura 6.1: Remuneración de las mujeres (medida como % del salario de un hombre en igual puesto) en función de la escolaridad (medida en años de estudios). Se observan medias, como así todas las unidades experimentales.

```
# ¡Quedó lindo!
# Interpretar el gráfico en términos del problema.

# Otro gráfico interesante
with(datos,boxplot(remuneración~escolaridad*año,,
                   xlab="Escolaridad (años)",
                   ylab="Remuneración mujeres (%)"))
abline(v=4.5, lty="dashed", col ="red")
```

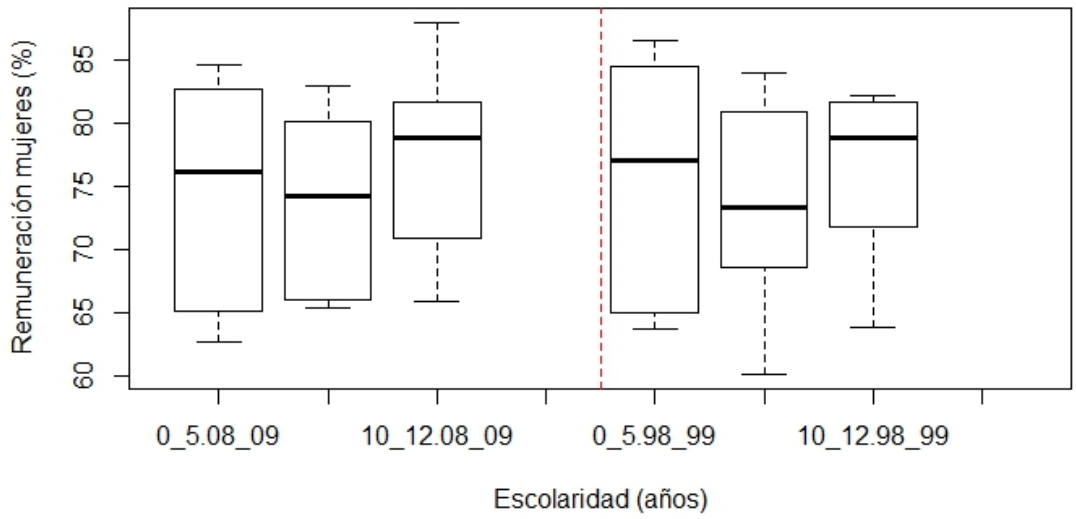


Figura 6.2: Diagrama de caja y bigotes reflejando la remuneración de las mujeres (medida como % del salario de un hombre en igual puesto) en función de la escolaridad (medida en años de estudios) según el modelo estimado.

6.3 Modelo y ANOVA

```
modelo=lm(remuneración~escolaridad*año, data=datos)
# Notar el uso del "*" en el modelo para indicar:
# Efecto principal de Escolaridad
# Efecto principal de año
# Efecto de interacción escolaridad y año
summary(modelo)
```

```
Call:
lm(formula = remuneración ~ escolaridad * año, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-13.233  -6.763   1.483   6.617  10.917

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 74.500      3.478  21.422   <2e-16 ***
escolaridad6_9              -0.700      4.918  -0.142    0.888
escolaridad10_12             2.817      4.918   0.573    0.571
año98_99                     1.083      4.918   0.220    0.827
escolaridad6_9:año98_99     -1.550      6.955  -0.223    0.825
escolaridad10_12:año98_99   -2.250      6.955  -0.323    0.749
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.518 on 30 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.03051,     Adjusted R-squared:  -0.1311
F-statistic: 0.1888 on 5 and 30 DF,  p-value: 0.9645
```

```
# A continuación mostramos la matriz de las variables predictoras para entender
# cómo es la parametrización del modelo.
model.matrix(~escolaridad*año, data=datos)
# tal vez es más práctico ver solamente las primeras filas
head(model.matrix(~escolaridad*año, data=datos))
```

```
(Intercept) escolaridad6_9 escolaridad10_12 escolaridad13 y más año98_99
1           1              0                0                    0        1
2           1              1                0                    0        1
3           1              0                1                    0        1
5           1              0                0                    0        0
6           1              1                0                    0        0
7           1              0                1                    0        0
escolaridad6_9:año98_99 escolaridad10_12:año98_99 escolaridad13 y más:año98_99
1                       0                         0                          0
2                       1                         0                          0
3                       0                         1                          0
5                       0                         0                          0
6                       0                         0                          0
7                       0                         0                          0
```

```
# Interpretar el ANOVA
anova(modelo)
```

```
Analysis of Variance Table
Response: remuneración
               Df  Sum Sq Mean Sq F value Pr(>F)
escolaridad     2   60.26  30.130  0.4152 0.6639
año             1    0.30   0.303  0.0042 0.9489
escolaridad:año 2    7.95   3.977  0.0548 0.9468
Residuals      30 2176.94  72.565
```

```
# ¿Por qué siempre evaluamos primero la hipótesis de no interacción
# entre los factores?

# En este caso, ¿Continúa el análisis con un test de comparaciones múltiples?
# ¿Por qué?
```

6.4 Potencia del Test

```
# ¿Cuántas repeticiones hay por tratamiento?
xtabs(~escolaridad+año, data=datos)

# install.packages("agricolae")
library(agricolae)
HSD.test(modelo,"escolaridad", console=TRUE)
HSD.test(modelo,"año", console=TRUE)
# si bien no es exactamente la potencia (probabilidad),
# el valor de HSD nos dice la magnitud de diferencias que podía detectar nuestro
 ensayo/análisis
# Estas diferencias hay que contextualizarlas según la magnitud de
# variación en nuestra variable de interés:
summary(datos$remuneración)
# Interpretar entonces los valores de HSD en términos del problema
# y discutir la utilidad el estudio realizado.

# si hacemos el test para los 8 tratamientos logicamente hay menos potencia
tratamientos=datos$escolaridad:datos$año
CME = deviance(modelo)/df.residual(modelo)
HSD.test(datos$remuneración,tratamientos,df.residual(modelo),CME, alpha=0.05,
 console=TRUE)
```

6.5 Supuestos

```
# Evaluar los supuestos del modelo por su cuenta y consultar a los docentes

# Una ayuda:
plot(modelo, which = c(2))
# Cuando los puntos se "cruzan" en la línea del qq plot como en este caso es un típico
 indicio de
# distinta curtosis a la distribución normal. Generalmente cuando "empiezan" por arriba
 de la línea
# y "terminan" por debajo de la línea, como en este caso, indica menor curtosis
```

```
# ¿Cómo afecta la curtosis y el muestreo?
par(mfcol=c(2,4))
residuos=resid(modelo)
hist(residuos, main="Datos REALES")
qqnorm(residuos, main="")
qqline(residuos)

normal=rnorm(n=48, mean=0, sd=1)
hist(normal,main="Distribución normal")
qqnorm(normal, main="")
qqline(normal)

t.student=rt(n=48, df=2)
hist(t.student,main="Mayor curtosis")
qqnorm(t.student, main="")
qqline(t.student)

uniforme=runif(n=48, min=0, max=1)
hist(uniforme,main="Menor curtosis")
qqnorm(uniforme, main="")
qqline(uniforme)
# Aquí se ve bien que nuestros datos se parecen más a una distribución uniforme,
# la cual tiene menor curtosis que la distribución normal

library(moments)
skewness(residuos) # debería dar cero
kurtosis(residuos) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# nuevamente vemos que nuestra muestra tiene menor curtosis que lo esperado según
 una normal
# ¿Cómo será en la población?
# Para ello tenemos los tests inferenciales:
# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos,"pnorm",mean(residuos),sd(residuos))

# alternativamente conviene usar la modificación de Lilliefors a este test
# esta corrección considera que los parámetros son estimados, a diferencia del ks "a
 secas"
library(nortest) #antes hay que instalar el paquete
lillie.test(residuos)

# Test de Shapiro-Wilk para evaluar normalidad
# n debe ser menor a 2000 y no es muy útil cuando tenemos datos repetidos...
shapiro.test(residuos)
# probemos con la normal
shapiro.test(normal)
```

```
# ¡Auch!

# Recordar evaluar los otros supuestos también.
```

6.6 Una alternativa: transformaciones

```
modelo=lm(log10(datos$remuneración)~datos$escolaridad*datos$año)
anova(modelo)
# el logaritmo no sirve para aumentar la curtosis!!!

# luego se evalúan los supuestos nuevamente
# se puede probar con otras transformaciones o asumiendo otro tipo de distribución tal
 como
# veremos más adelante en el curso.
```

6.7 Segundo problema

```
# El flujo de fondos es una herramienta central para la administración de una empresa. ·
# Este representa la acumulación neta de activos líquidos de una compañía en
# un período determinado y por lo tanto es utilizado como indicador de su liquidez.
# Concretamente, el flujo de fondos permite optimizar el uso diario de los fondos. Sin
# embargo, uno de los problemas financieros más comunes de una empresa es
#desatender su flujo de fondos lo que las puede llevar a tomar decisiones equivocadas.
#Por ejemplo, una mala gestión de cobros genera problemas para financiar las
#operaciones corrientes. Esto tiene un efecto en cadena ya que si una empresa se
#demora en el pago de deudas, perjudica la capacidad de pago de sus proveedores,
#quienes a su vez se retrasan con sus acreedores. Es así que es común que muchas
#empresas quiebren no por tener pérdidas sino por no poder resolver de forma eficiente
#los problemas asociados al flujo de fondos.

# Una empresa dedicada a la venta de bienes finales en la Patagonia ha perdido
# en los últimos meses la posibilidad de que sus proveedores le otorgen importantes
# descuentos por no poder afrontar el pago de contado al momento de realizar las
# operaciones de compra. Por lo tanto, la empresa desea mejorar la logística de compra
# de mercaderías y para ello es fundamental predecir el flujo de ingresos corrientes
# semanales, ya que estos ingresos indican la capacidad de pago inmediato. Al conocer
# en que momento del mes es más probable tener liquidez, la empresa podría planificar
# la compra de mercaderías y negociar con sus proveedores las bonificaciones por pago
# de contado. De esta manera, el área financiera hace tres meses que comenzó a
# desarrollar un sistema de previsión de ingresos por ventas. Específicamente, han
# registrado los ingresos (en miles de pesos) correspondientes a cada una de las semanas
# del mes en sus dos casas de ventas (Bariloche y Comodoro Rivadavia).

# ============================================================

# OBSERVACIÓN: se trata de un problema real acercado por un alumno y modificado
con fines didácticos
#
# TÉRMINOS TÉCNICOS UTILIZADOS:
#
```

```
#   - "Activos de una empresa": conjunto de bienes económicos.
#
#   - "Activos liquidos": dinero o activos que pueden convertirse en dinero
inmediatamente.
#
#   - "Liquidez": capacidad de una empresa de atender sus obligaciones de pago.
Relacionado con la cantidad de activos liquidos que posee la empresa.
#
#   - "Operaciones corrientes": operaciones de pago de personal, bienes y servicios,
gastos financieros, y trasferencias corrientes.
#
#   - "Bienes finales": bienes que ya están aptos para el consumo de la sociedad.
#
================================================================
#      Desde la gerencia de la empresa se ha solicitado al área de finanzas un informe
# evaluando si el flujo de ingresos varía de acuerdo a la semana del mes, y si esa
# variación es consistente entre ciudades.


INGRESO = c(292, 293, 261, 229, 135, 129, 108, 92,
            260, 325, 215, 210, 152, 145, 125, 86,
            334, 295, 248, 248, 176, 167, 96, 51)

SEMANA = c("S1", "S2", "S3", "S4", "S1", "S2", "S3", "S4",
           "S1", "S2", "S3", "S4", "S1", "S2", "S3", "S4",
           "S1", "S2", "S3", "S4", "S1", "S2", "S3", "S4")

CIUDAD = c("BCH", "BCH", "BCH", "BCH", "CR", "CR", "CR", "CR",
           "BCH", "BCH", "BCH", "BCH", "CR", "CR", "CR", "CR",
           "BCH", "BCH", "BCH", "BCH", "CR", "CR", "CR", "CR")

datos = data.frame(INGRESO, SEMANA, CIUDAD)


# El siguiente es un grafico gráfico que debería mostrarse en el informe. Interpretelo.

with(datos, interaction.plot(SEMANA, CIUDAD, INGRESO,
                             xlab="Semana",
                             ylab="Ingreso (miles de pesos/semana)",
                             ylim=c(50, 350),
                             type="b", pch=c(19, 19), cex=1.25, col=c("darkred", "darkgreen"),
                             leg.bty="o", bty="l"))
with(datos, points(jitter(as.numeric(SEMANA),factor=0.5), INGRESO,
                   pch=ifelse(CIUDAD == "CR", 1, 19), cex=0.8))
```

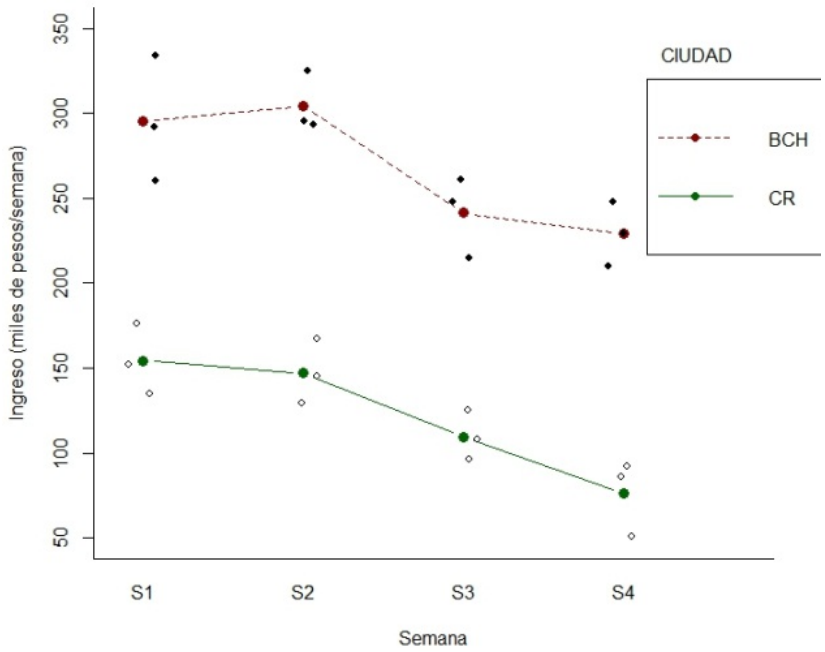


Figura 6.3.: Gráfico de efectos de interacción entre tratamientos para ingreso promedio según ciudad.

```
# Plantee un modelo estadístico que considere adecuado para evaluar lo que requiere
# la gerencia. Estime e informe los parámetros del modelo con sus respectivas unidades.

# Podemos obtener una matriz con:

#     i)   los datos observados de la variable dependiente (Y).
#     ii)  los datos observados en las variabels independientes (X).
#     iii) los Residuos.
#     iv)  los valores Predichos por el modelo para cada unidad experimental.

modelo = lm(INGRESO ~ SEMANA * CIUDAD, data = datos)
y = INGRESO
residuos = resid(modelo)
x = model.matrix(~ SEMANA * CIUDAD, data=datos)
predichos =  fitted(modelo)

matriz = data.frame(Y, X, residuos, predichos)
matriz

# ¿Cuáles son las unidades de cada término de la matriz obtenida?


# ¿Cuál es la diferencia entre multicolinearidad e interacción? Explique en relación
# al problema.


# ¿El informe debería recomendar alguna/s semana/s en la cual realizar el pago a
```

```
# los proveedores? ¿La recomendación debería ser la misma en ambas casas de venta?
# Apoye su respuesta con el análisis estadístico que considere adecuado.
```

Volver al Índice principal

Regresión múltiple

Trabajo Práctico Nº 7: Diseño multi-factorial

7.1 Problema y datos

```
# datos obtenidos de www.gapminder.org

# Desde un organismo internacional de desarrollo se desea estudiar cómo varía el
 número de hijos por
# madre en promedio. Se cree que la cantidad de hijos promedio por madre está en parte
 determinada
# por el grado de desarrollo del país el cual es medido mediante dos indicadores
 clásicos: la
# esperanza de vida y el PBI per cápita

datos=read.table("datos_p_7.txt")
colnames(datos)= c("pais","hijos","vida","PBI")
```

```
datos
str(datos)

# Variables:
# país
# hijos = hijos por madre en promedio
# vida = esperanza de vida
# PBI = producto bruto interno per cápita corregido por inflación
```

7.2 Consignas a resolver

```
# 1) ¿Cuál es la unidad experimental? ¿Qué tipo de diseño se empleó en este caso?

# 2) ¿Es un experimento mensurativo o manipulativo? Justifique

# 3) Responda a la pregunta principal del trabajo utilizando los análisis que considere
# adecuados.

# 4) Elija algún valor p relevante y explique en términos del problema
# qué significa ese valor p. ¿Cuál es la distribución del estadístico bajo la hipótesis
# nula en este caso?

# 5) Evalúe los supuestos del análisis.

# 6) ¿Valdría la pena aumentar la potencia de este ensayo? En caso afirmativo, ¿qué
 medidas tomaría?
```

7.3 Otro ejemplo

```
# Ejercicio 32 de la página 527 de Anderson et al. 2008
# La "U.S. Bureau of Labor Statistics", recoge información sobre sueldos de hombres y
# mujeres en diversas ocupaciones. Se desea conocer si hay diferencia entre los
# sueldos semanales de hombres y mujeres que trabajan como administradores
 financieros,
# programadores y farmaceúticos. De cada una de estas ocupaciones se encuestan
# cinco hombres y cinco mujeres y se registra el sueldo semanal de cada uno de ellos.
# Los datos obtenidos son los que se presentan a continuación:
ocupación=factor(rep(c("adm_financiero","programador","farmacéutico"),c(10,10,10)))
ocupación
sueldo=c(872,859,1028,1117,1019,519,702,805,558,591,747,766,901,690,881,884,765,
685,700,671,
        1105,1144,1085,903,998,813,985,1006,1034,817)
sueldo
género=factor(rep(rep(c("Hombre","Mujer"),c(5,5)),  3))
género
datos=data.frame(ocupación,sueldo,género)
View(datos)


# 1) ¿Cuál es el tamaño de la muestra?
```

```
# ¿Cuál es la muestra? ¿Cuál es la unidad experimental?
# 2) Plantear un modelo estadístico que permita
# llevar adelante el objetivo planteado.


# 3) Cumpla con el objetivo planteado a partir de los datos recolectados.

interaction.plot(ocupación,género,sueldo)
```

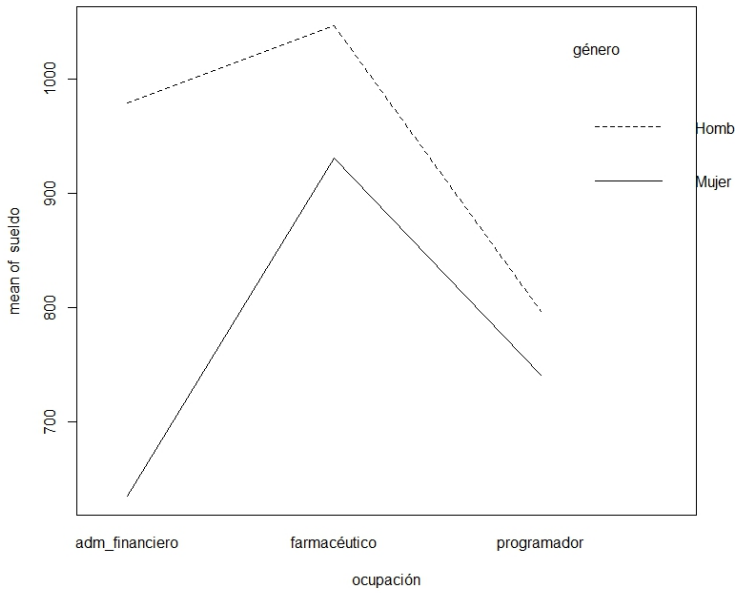


Figura 7.1: Gráfico de efectos de interacción entre tratamientos para remuneración promedio de mujeres y hombres según sector administrativo/financiero, farmacéutico y programador.

```
# este gráfico también sirve para tener una impresión respecto de si se cumple el
 supuesto
# de no interacción en los diseños en bloques que vimos en clases anteriores
# Sería mucho más útil si le agregáramos +- 1 error estándar a cada media

modelo = with(datos, lm(sueldo~ocupación*género))
summary(modelo)
anova(modelo)
```

```
Analysis of Variance Table

Response: sueldo
                Df Sum Sq Mean Sq F value    Pr(>F)
ocupación        2 276560  138280 13.2456 0.0001330 ***
género           1 221880  221880 21.2536 0.0001119 ***
ocupación:género 2 115440   57720  5.5289 0.0105954 *
Residuals       24 250552   10440
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Como hay interacción significativa no podemos hablar del efecto de un factor
# sin considerar en que nivel del otro factor nos encontramos.
```

```
# Por lo tanto el análisis de tukey se hace para todos los tratamientos. No es
# válido comparar mediante análisis de tukey los niveles de cada factor por separado.
# Necesitamos crear una variable que sea "tratamiento"

# install.packages("agricolae")
library(agricolae)
tratamiento=factor(rep(c("finan_hombre","finan_mujer","program_hombre",
                "program_mujer","farma_hombre","farma_mujer"),
                c(5,5,5,5,5,5)))
# otra forma mucho más fácil
datos$tratamiento=paste(datos$ocupación, datos$género, sep="_")

CME = deviance(modelo)/df.residual(modelo)
tukey=with(datos, HSD.test(y=sueldo,
                    trt=tratamiento,
                    DFerror=df.residual(modelo),
                    MSerror=CME,
                    alpha=0.05,
                    console=TRUE))
```

```
Study: sueldo ~ tratamiento

HSD Test for sueldo

Mean Square Error:  10439.67

tratamiento,  means

                       sueldo       std r Min  Max
adm_financiero_Hombre     979 110.55994 5 859 1117
adm_financiero_Mujer      635 116.95084 5 519  805
farmacéutico_Hombre      1047  96.63591 5 903 1144
farmacéutico_Mujer        931 107.31962 5 813 1034
programador_Hombre        797  90.52900 5 690  901
programador_Mujer         741  87.66698 5 671  884

alpha: 0.05 ; Df Error: 24
Critical Value of Studentized Range: 4.372651

Honestly Significant Difference: 199.8035

Means with the same letter are not significantly different.

Groups, Treatments and means
a         farmacéutico_Hombre       1047
ab        adm_financiero_Hombre     979
abc       farmacéutico_Mujer        931
bcd       programador_Hombre        797
cd        programador_Mujer         741
d         adm_financiero_Mujer      635
```

```
bar.group(tukey$groups,ylim=c(0,1200),density=4,border="blue")
```

```
# 4) ¿Cuál es el valor predicho para hombres que son programadores?
summary(modelo)
modelo$coefficients[1]+modelo$coefficients[3]
# interpretar dicho valor en términos del problema.
# Indicar sus unidades

# 5) ¿Cuál es la capacidad predictiva (bondad de ajuste) del modelo?
# Interpretar.

# 6) Interprete el resultado del siguiente código
cbind(Estimado=coef(modelo), confint(modelo))
```

```
                                Estimado      2.5 %     97.5 %
(Intercept)                          979  884.69241 1073.30759
ocupaciónfarmacéutico                 68  -65.37108  201.37108
ocupaciónprogramador                -182 -315.37108  -48.62892
géneroMujer                         -344 -477.37108 -210.62892
ocupaciónfarmacéutico:géneroMujer    228   39.38481  416.61519
ocupaciónprogramador:géneroMujer     288   99.38481  476.61519
```

```
# 7) Otro gráfico interesante
library(effects)
plot(effect("ocupación:género",modelo))
```

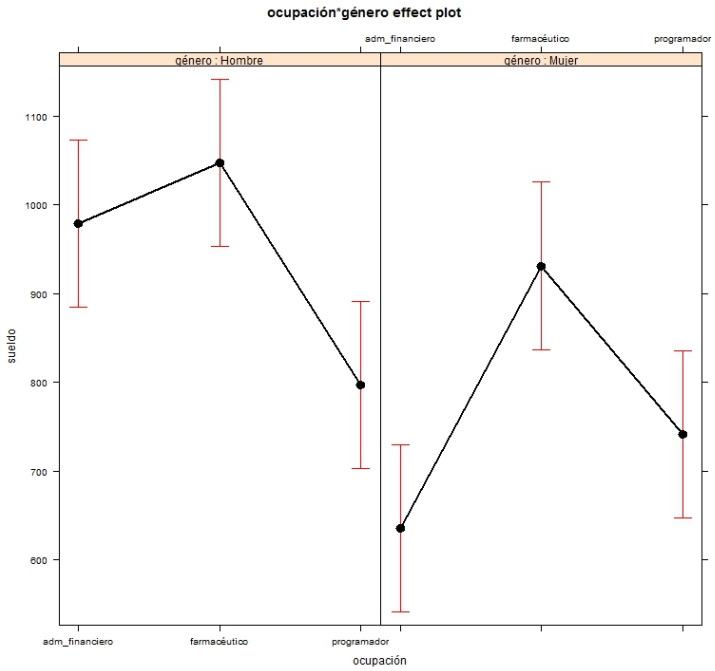


Figura 7.2: Nivel de remuneración de hombres y mujeres según ocupación con intervalos de confianza del 0.95 incluidos.

```
# Interpretar el gráfico y los intervalos de confianza en el contexto del
# problema

# 8) Continúen con la evaluación de los supuestos.
```

Trabajo Práctico Nº 8: Multicolinealidad y potencia de los coeficientes de regresión lineal

8.1 Problema y Datos

```
# La inflación y sus determinantes es uno de los principales temas de discución entre
 economistas.
# Por un lado autores de la corriente "monetaristas" sostienen que esta se ve
 directamente determinada
# por las variaciones en la masa de dinero circulante, mientras que otros autores más
 partidarios del
# análisis "real" de la economía sostienen que esta es en realidad producto de la presión
 de la demanda
# sobre una oferta rezagada. Un investigador pretende conciliar ambas posturas en una
 única al proponer
# que la inflación es en realidad producto de las variaciones en la base monetaria, y por
 otro, de las
# variaciones en la demanda agregada de una nación.

# Los datos son indices correspondientes al 2010 tomando al 2005 como base
 (2005=100).
# IPC= indice de precios al consumidor. Las diferencias en la elaboración de los indices
 son mínimas
# siendo que son todos elaborados en acuerdo con reglas estrictas de la OECD
# DD: Demanda doméstica. Total de bienes y servicios demandados por los residentes
 de una nación (quitando
# exportaciones y sumando importaciones del PBI).
# M1: Base monetaria. Valor del total de billetes impresos por el Banco central de una
 nación.

# La tabla es de elaboración propia en base a datos de la OECD.
 www.oecd.org/statistics/
```

8.2 Consignas a resolver

```
# 1) En papel, indique el modelo conceptual y el modelo estadístico acorde.  Indique
 unidad experimental
# , muestra y población. Dada la naturaleza del experimento llevado adelante, ¿Que
 problema considera que
# pueda acarrear el hecho de tomar un gran número de paises desarrollados en su
 muestra?

# 2) Elabore un gráfico que permita visualizar la relación entre las variables que plantea
 el modelo conceptual
# del investigador.

# 3) Estime un modelo para cada variable independiente por separado. ¿Su muestra le
 permite corroborar la significancia
# de la idea de los monetaristas? y ¿de la del grupo de economistas volcado al análisis
```

```
 "real" de la economía? De alguna
# medida de presición de la estimación de la pendiente y de la bondad de ajuste de cada
 modelo.

# 4) Estime el modelo de regresión múltiple propuesto por el investigador. Discuta su
 validez y relevancia.
# ¿Cómo explicaría que en el summary() la variable DD no es significativa mientras que
 en el anova() siempre
# que se coloca primera resulta significativa? Puede concluirse que la variabilidad de
 DD no es significativa
# para explicar las variaciones en IPC? Desde la economía ¿que crítica haría al modelo
 conceptual del investigador?

# 5) Evalue los supuestos de todos los modelos elaborados. ¿Con cual se quedaría?
#¿Podemos extrapolar las conclusiones de dicho modelo al caso argentino?
```

8.3 Otro ejemplo

```
# Queremos saber si existe una relación entre el ingreso bruto en función
# de los gastos de publicidad en televisión y periódicos.
# Unidad experimental = cada empresa

ingreso=c(96,90,95,92,95,94,94,94)

tele=c(5.0,2.0,4.0,2.5,3.0,3.5,2.5,3.0)

diarios=c(1.5,2.0,1.5,2.5,3.3,2.3,4.2,2.5)

datos=data.frame(ingreso,tele,diarios)
# las tres variables están expresadas en miles de pesos

#### MARCO CONCEPTUAL ####
# La publicidad en diarios y en televisión genera una necesidad por un determinado
 producto
# en el consumidor, por lo tanto estimula la compra de ese producto en la empresa que
 publicita.

# La publicidad en diarios y en televisión influye mayoritariamente sobre distintos
 consumidores,
# por lo tanto tienen efectos independientes.

# Los niveles de publicidad habitualmente empleados por las empresas medianas de
 interés
# para el estudio no logran alcanzar a todos los consumidores, por lo tanto la relación
# entre gasto en publicidad e ingreso bruto es lineal.

# Se supone que invertir en publicidad en televisión y diarios es rentable, por lo tanto
# los coeficientes de regresión parcial deberían ser mayor a 1.
```

8.4 Gráfico y modelo

```
# ¿Qué tipo de variables son ingreso, tele y diarios?

# ¿La tabla de datos es de tipo "panel",
# "serie de tiempo" o "corte transversal"?
# ¿Depende el ingreso de los gastos en publicidad en televisión y diarios?
# install.packages("scatterplot3d")
library(scatterplot3d) # Recuerden tener instalado el paquete antes

s3d=scatterplot3d(x=datos$tele, y=datos$diarios, z=datos$ingreso,
          xlab="Gastos en televisión (miles de $)", ylab="Gastos en diarios (miles de
 $)",
             zlab="Ingreso bruto (miles de $)",
             xlim=c(2,5.1),
          pch=16, highlight.3d=TRUE, angle=50,type="h")

modelo=lm(ingreso~tele + diarios, data=datos)
s3d$plane3d(modelo, lty.box="solid",lwd=2, col="blue")
```

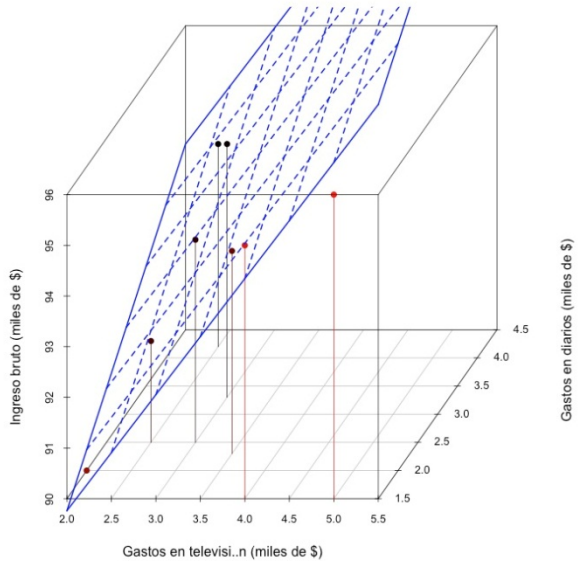


Figura 8.1: Ingresos brutos (en miles de $) en función de gastos en propaganda de televisión y diarios (ambos en miles de $).

```
# Estimamos los coeficientes parciales de regresión
# ¿Cuáles son sus unidades?
summary(modelo)

Call:
lm(formula = ingreso ~ tele + diarios, data = datos)

Residuals:
      1       2       3       4       5       6       7       8
-0.6325 -0.4124  0.6577 -0.2080  0.6061 -0.2380 -0.4197  0.6469
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.2301     1.5739  52.882 4.57e-08 ***
tele          2.2902     0.3041   7.532 0.000653 ***
diarios       1.3010     0.3207   4.057 0.009761 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6426 on 5 degrees of freedom
Multiple R-squared:  0.919,       Adjusted R-squared:  0.8866
F-statistic: 28.38 on 2 and 5 DF,  p-value: 0.001865
```

```
# el summary también contiene el anova global del modelo

# ANOVA secuencial para cada una de las variables independientes
anova(modelo)
```

```
Analysis of Variance Table

Response: ingreso
          Df  Sum Sq Mean Sq F value   Pr(>F)
tele       1 16.6401 16.6401  40.299 0.001432 **
diarios    1  6.7953  6.7953  16.457 0.009761 **
Residuals  5  2.0646  0.4129
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# para entender el ANOVA secuencial
modelo2=lm(ingreso~tele, data=datos)
anova(modelo2)
```

```
Analysis of Variance Table

Response: ingreso
          Df  Sum Sq Mean Sq F value  Pr(>F)
tele       1 16.6401 16.6401  11.269 0.01529 *
Residuals  6  8.8599  1.4767
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Comparar la SC de "tele" en este modelo con respecto
# al anova(modelo). Interpretar.

# Comparar la SC de "Residuals" en este modelo con
# respecto a la SC de "diarios" sumada a la de "Residuals" del anova(modelo).
# Interpretar.

rm2=resid(modelo2)
plot(datos$diarios,rm2, xlab="Gasto en diarios (miles de $)",
    ylab="Ingreso bruto (miles de $)")
abline(lm(rm2~datos$diarios ))
```

```
anova(lm(rm2~datos$diarios))

# Comparar la SC de "datos$diarios" + "Residuals"
# con la SC de "Residuals" en anova(modelo2). Interpretar.
```

8.5 Multicolinealidad

```
# la multicolinealidad es un problema que sucede cuando las variables
# independientes están muy correlacionadas entre sí.
with(datos, cor.test(tele,diarios))
with(datos, plot(tele,diarios,
                 xlab="gasto en televisión (miles $)",
                 ylab="gasto en diarios (miles $)"))
abline(lm(datos$diarios~datos$tele))
# es un problema de grado, cuanto mayor es la correlación entre las variables
# independientes mayor es este problema

# corremos nuevamente el anova secuencial
anova(modelo)
```

```
Analysis of Variance Table

Response: ingreso
          Df  Sum Sq Mean Sq F value   Pr(>F)
tele       1 16.6401 16.6401  40.299 0.001432 **
diarios    1  6.7953  6.7953  16.457 0.009761 **
Residuals  5  2.0646  0.4129
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Los gastos de publicidad en diarios explican significativamente parte de la variación
# en el ingreso bruto que no ha sido explicada por una relación lineal entre ingreso
# bruto y gastos de publicidad en televisión.
# NO HAY UN PROBLEMA DE MULTICOLINEALIDAD AQUÍ.
```

8.6 Factor de inflación de la varianza

```
summary(lm(datos$tele~datos$diarios))

FIV= 1 / (1-0.3096)
FIV
# es un valor relativamente bajo. Algunos autores dicen que uno debería preocuparse a
 partir de un
# valor de 5 - 10 . Pero recordar que este es un valor totalmente arbitrario.
# Recordar que hay un FIV por cada variable independiente. En este caso como son sólo
 dos variables
# independientes hay dos FIV con el mismo valor.
```

```
# Consideremos ahora la variable gasto de publicidad en el diario más importante
# recordemos que la variable anterior "diarios" era el gasto de publicidad en todos los
# diarios, mientras que ahora nos concentramos solamente en el diario más importante
undiario=c(1.1,1.4,0.9,2.1,2.8,1.7,3.6,2.2)
datos=data.frame(datos,undiario)
plot(datos$undiario,datos$diarios)
abline(lm(datos$diarios~datos$undiario))

# ajustemos ahora un modelo de regresión múltiple con tres variables independientes
modelo2=with(datos,lm(ingreso~tele+diarios+undiario))
anova(modelo2)

modelo3=with(datos,lm(ingreso~tele+undiario+diarios))
anova(modelo3)
# ¿A qué conclusión llegamos al comparar el modelo 2 con el modelo 3?


# IMPORTANTE, comparar:
summary(modelo)

Call:
lm(formula = ingreso ~ tele + diarios, data = datos)

Residuals:
      1       2       3       4       5       6       7       8
-0.6325 -0.4124  0.6577 -0.2080  0.6061 -0.2380 -0.4197  0.6469

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.2301     1.5739  52.882 4.57e-08 ***
tele          2.2902     0.3041   7.532 0.000653 ***
diarios       1.3010     0.3207   4.057 0.009761 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6426 on 5 degrees of freedom
Multiple R-squared:  0.919,       Adjusted R-squared:  0.8866
F-statistic: 28.38 on 2 and 5 DF,  p-value: 0.001865

summary(modelo2)

Call:
lm(formula = ingreso ~ tele + diarios + undiario)

Residuals:
      1       2       3       4       5       6       7       8
-0.6845 -0.3422  0.7669 -0.3106  0.5988 -0.1417 -0.3506  0.4637

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.7260     2.1334  39.246 2.52e-06 ***
tele          2.2718     0.3367   6.747  0.00252 **
diarios       0.4077     2.2852   0.178  0.86706
undiario      0.8979     2.2698   0.396  0.71259
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7048 on 4 degrees of freedom
Multiple R-squared:  0.9221,      Adjusted R-squared:  0.8636
F-statistic: 15.78 on 3 and 4 DF,  p-value: 0.01108
```

```
# comparar el valor estimado para el efecto de diarios entre modelos.
# ¿Por qué cambia tan fuertemente?

# Comparar el error estándar asociado al efecto de diarios entre modelos.
# ¿Por qué cambia tan fuertemente?

# FIV para las tres variables independientes
summary(lm(tele~diarios+undiario,data=datos))
FIVtele= 1 / (1-0.3227)

summary(lm(diarios~tele+undiario,data=datos))
FIVdiarios= 1 / (1-0.9836)

summary(lm(undiario~tele+diarios,data=datos))
FIVundiario= 1 / (1-0.9831)

FIVs=c(FIVtele,FIVdiarios,FIVundiario)
FIVs=as.data.frame(FIVs)
rownames(FIVs)=c("FIVtele","FIVdiarios","FIVundiario")
View(FIVs)

summary(modelo2)
# ¿Por qué diarios no es significativo en el "modelo2" pero sí lo es en el "modelo"?
summary(modelo)

# Presentamos los gls y el ANOVA tipo 3
install.packages("nlme")
library(nlme)
modelo4=gls(ingreso~tele+diarios+undiario, data=datos)
summary(modelo4)
```

```
Generalized least squares fit by REML
  Model: ingreso ~ tele + diarios + undiario
  Data: datos
       AIC      BIC    logLik
  21.54984 18.48132 -5.774922

Coefficients:
               Value Std.Error  t-value p-value
(Intercept) 83.72602 2.1333795 39.24572  0.0000
tele         2.27182 0.3367069  6.74719  0.0025
diarios      0.40774 2.2852201  0.17842  0.8671
undiario     0.89792 2.2697867  0.39560  0.7126

 Correlation:
         (Intr) tele   diaris
tele     -0.799
```

```
diarios  -0.686  0.221
undiario  0.588 -0.138 -0.988

Standardized residuals:
       Min         Q1        Med         Q3        Max
-0.9711670 -0.4885479 -0.3208217  0.7059034  1.0882119

Residual standard error: 0.7047798
Degrees of freedom: 8 total; 4 residual
```

```
# Corroboramos que el "summary(modelo4)" da lo mismo que el "summary(modelo2)"
# con "lm" se puede estimar un subconjunto de modelos de los que podremos estimar
 con "gls"
summary(modelo2)
# ahora verificamos lo mismo para anova
anova(modelo2)
anova(modelo4) #si uno no indica que anova quiere, presenta el anova secuencial o tipo
 1

# ver
anova(modelo4, type="sequential")
```

```
Denom. DF: 4
            numDF   F-value p-value
(Intercept)     1 141555.15  <.0001
tele            1     33.50  0.0044
diarios         1     13.68  0.0209
undiario        1      0.16  0.7126
```

```
# ¿Por qué con el ANOVA secuencial "diarios" tiene un efecto significativo
# pero NO con el test de t dado con el summary?

# presentamos el anova marginal o tipo 3
anova(modelo4, type="marginal")

# los valores p son iguales a los del summary
summary(modelo4)
# Explicar
```

8.7 Intervalos de confianza y predicción

```
# Intervalos de confianza para la ordenada y los coeficientes de regresión parciales
confint(modelo, level=0.95)
# Interpretar

# pedimos el intervalo de confianza para la media de ingreso bruto
# dado cierto valor de gasto en publicidad en tele y en diarios.
predict.lm(modelo,interval="confidence")
```

```
fit      lwr      upr
1 96.63249 95.31835 97.94664
```

```
2 90.41244 89.08565 91.73923
3 94.34231 93.43544 95.24917
4 92.20802 91.42194 92.99411
5 94.39391 93.54881 95.23900
6 94.23801 93.61968 94.85633
7 94.41970 93.07741 95.76200
8 93.35312 92.75344 93.95279
```

```
# ¿Cuál es el intervalo de confianza para la media de ingreso bruto si se invierte
# 3 mil pesos en  publicidad en televisión y 4 mil pesos en diarios?
newdata=data.frame(tele=3.0,diarios=4.0)
predict(modelo,newdata,interval="confidence")

# ¿Cuál es el intervalo de confianza para la media de ingreso bruto si se invierte
# 10 mil pesos en  publicidad en televisión y 4 mil pesos en diarios?
# ahora el intervalo de predicción
predict.lm(modelo,interval="prediction")


# Si el año que viene mi empresa quiere invertir 2 mil pesos en televisión
# y 3 mil pesos en diarios:
# ¿Qué valor esperaría obtener de ingreso bruto? ¿Entre qué valores podría estar el
# ingreso bruto del año que viene? ¿Con qué confianza?
newdata=data.frame(tele=3.0,diarios=4.0)
predict(modelo,newdata,interval="prediction")
```

8.8 Bondad de ajuste

```
# CME como medida de bondad de ajuste
residuos=resid(modelo)
CME = (sum(residuos^2))/(length(residuos)-3)
CME
```

```
[1] 0.4129184
```

```
# este es el desvío estándar residual que está expresado en las mismas
# unidades que la variable dependiente
DSR = sqrt(CME)
DSR
```

```
[1] 0.6425873
```

```
# recuerden que por regla empírica, si los datos están distribuidos normalmente,
# el rango comprendido por
# la media +/- 1 desvío estándar contendrá al 68.3% de las observaciones
# media +/- 2 DE ----> 95.5% de las observaciones
# media +/- 3 DE ----> 99.7% de las observaciones


# plano de regresión junto con el intervalo que contiene a casi el 70% de las empresas
s3d=scatterplot3d(x=datos$tele, y=datos$diarios, z=datos$ingreso,
```

```
        xlab="Gastos en televisión (miles de $)", ylab="Gastos en diarios (miles de
$)",
          zlab="Ingreso bruto (miles de $)",
          pch=16, highlight.3d=TRUE, angle=35,type="h")


s3d$plane3d(modelo, lty.box="dashed",lwd=2, col="blue")

modelo5=lm(datos$ingreso+DSR~datos$tele + datos$diarios)
s3d$plane3d(modelo5)

modelo6=lm(datos$ingreso-DSR~datos$tele + datos$diarios)
s3d$plane3d(modelo6)

# también es interesante ver:
predichos=fitted(modelo)
with(datos,plot(predichos,ingreso))
abline(a=0,b=1,lwd=2)
```

8.9 Coeficiente de determinación

```
# lo podemos ver haciendo
summary(modelo)

# o más directamente
summary(modelo)$r.squared

# también podemos estimarlo "nosotros" y comparar con la salida anterior
SCE = sum(residuos^2)
SCTotal = with(datos, sum (  (ingreso-(mean(ingreso)))  ^2  )  )
r2 = (SCTotal - SCE) / SCTotal
r2


###### COEFICIENTE DE DETERMINACIÓN CORREGIDO ######
r2corregido = 1  - ( (SCE/5) / (SCTotal/7)  )# usando fórmula Webster
r2corregido

[1] 0.8866498

r2_corr_ande = 1 - (1-r2) * (8-1)/(8-2-1) # usando fórmula Anderson (y Webster)
r2_corr_ande

[1] 0.8866498

# comparar con
summary(modelo)$adj.r.squared
```

8.10 Supuestos

```
# evaluamos los supuestos del modelo a través de los residuos
residuos=resid(modelo)

# para evaluar el supuesto de homogeneidad de varianzas,
# así como el de independencia
predichos=fitted(modelo)
plot(predichos,residuos)
abline(a=0,b=0, col="violet", lw=2)

# para evaluar el supuesto de normalidad:
par(mfrow=c(2,1))
hist(residuos, col="yellow")
boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
 datos
par(mfrow=c(2,2))
hist(residuos, col="yellow", freq=F)
lines(density(residuos))

normal=rnorm(mean=0, sd=summary(modelo)$sigma,
 n=length(summary(modelo)$residuals))
hist(normal, col="green", freq=F)
lines(density(normal))

boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")
```

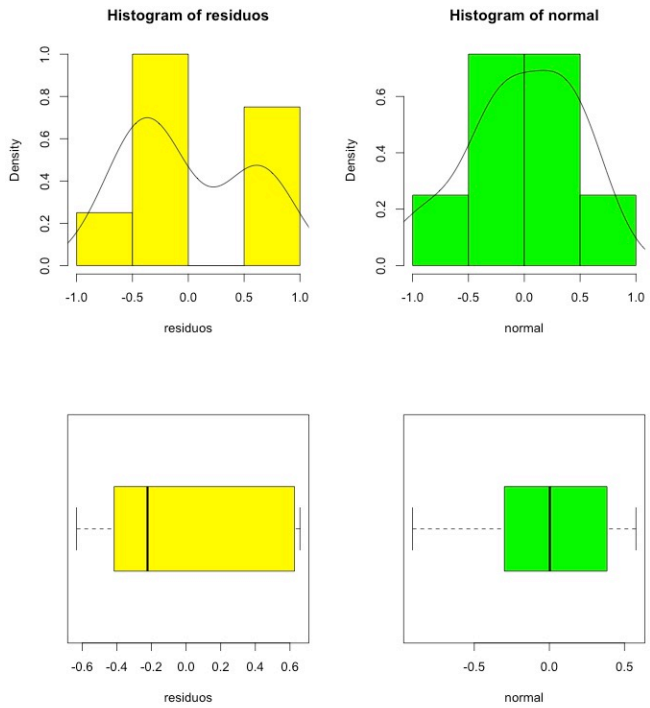


Figura 8.2: Distribución de los residuos para la muestra en estudio y una distribución normal de igual desvío estándar residual que la muestra.

```
qqnorm(residuos)
qqline(residuos)

# también se puede hacer para los residuos estandarizados
plot(modelo, which = c(2))

library(moments)
skewness(residuos) # debería dar cero
kurtosis(residuos) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos,"pnorm",mean(residuos),sd(residuos))

# alternativamente conviene usar la modificación de Lillefors a este test
# install.packages("nortest")
library(nortest) # primero hay que instalar el paquete
lillie.test(residuos)

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos)


# Evalúen si alguna observación/nes son muy
```

```
# influyentes en el análisis.
```

8.11 Potencia

```
summary(modelo)

# vamos a calcular la potencia para las pruebas t del "summary"
# lo haremos en particular para el coeficiente de regresión parcial de "tele"
# Recordemos la estandarización t:

# t = (b1 - B1)  /  sb1

# empezamos por "B1"
# debemos elegir la hipótesis alternativa "más chica" relevante
# en este caso podría ser 1, que es aquella en la que las empresas no pierden
# dinero ya que por cada peso extra gastado en publicidad hay un ingreso extra en
# televisión.
# Es importante justificar conceptualmente la hipótesis alternativa
# elegida para estimar la potencia.
B1=1

# seguimos por "b1"
# queremos el valor del coeficiente de regresión parcial a partir del cual
# se rechazaría la hipótesis nula.
# Primero lo buscamos en escala t suponiendo un alfa = 0.05
t1=qt(0.025,df=5,lower.tail=FALSE)
t1

[1] 2.570582

# con t1 buscamos el valor de b1 a partir del cual hubiésemos
# rechazado la hipótesis nula
# t1 = (b1 - 0)  /  sb1
b1 = t1 * 0.3041
b1

[1] 0.7817139

# es decir que hubiésemos rechazado la hipótesis nula siempre y cuando nuestro
# valor de medio del coeficiente de regresión parcial asociado a tele fuese mayor
# a 0.78


# entonces ahora podemos calcular el t asociado a la hipótesis alternativa.
# Valores menores a este t podrían implicar un error de tipo 2.
t = (b1 - B1) / 0.3041
t

[1] -0.7178101
```

```
# ahora buscamos la probabilidad asociada a este valor t
beta= pt(t,df=5,lower.tail=TRUE)
beta

[1] 0.252502

# por lo tanto la potencia
potencia= 1 - beta
potencia

[1] 0.747498
```

8.12 Selección de modelos por AIC

```
# Teoría de la información
# Esta sección es a modo de discución para aquellos que ya han leído sobre el tema.
# No hemos explicado en el curso el fundamento de este análisis.
# Comparar con resultados ANOVA
library("MuMIn") # Recordar instalar el paquete antes con install.packages("MuMIn")

modelo=lm(ingreso~tele + diarios, data=datos, na.action="na.fail")
# volvemos a correr el modelo aclarando que si hay NAs
# el modelo no debería estimarse. En nuestro caso no hay NAs
# por lo tanto se estima sin problemas. Esta acción para NAs
# es requisito para utilizar la función "dredge" que sigue.

selec<-dredge(modelo)
# IMPORTANTE: la función dredge genera todos los modelos posibles
# (incluido el modelo nulo que solo tiene a la ordenada al origen) y los compara
 ajustados por ML.
nrow(selec) # Contamos la cantidad de filas que tiene la tabla "selec"
# La función dredge generó 4 modelos
selec
Global model call: lm(formula = ingreso ~ tele + diarios, data = datos, na.act
ion = "na.fail")
---
Model selection table
  (Intrc)    diars  tele df  logLik AICc delta weight
4   83.23  1.30100 2.290  4  -5.933 33.2  0.00  0.718
3   88.64          1.604  3 -11.760 35.5  2.32  0.225
1   93.75                 2 -15.988 38.4  5.18  0.054
2   93.86 -0.04299        3 -15.987 44.0 10.77  0.003
Models ranked by AICc(x)

# La salida es una tabla ordenada de los modelos del mejor al peor ajuste según AIC.
# La tabla contiene: la estimación de los predictores continuos
# y un signo "+" para los predictores categóricos incluidos en el modelo (en este caso
 ninguno).
# Además, muestra la cantidad de parámetros (df), el logLik, el AICc,
# el delta (la diferencia respecto del mejor modelo) y el weight (peso relativo).
```

8.13 Mínimos cuadrados

```
# Recordar estudiar las diapositivas del teórico de clase
# para comprender esta sección al detalle.

# Escribir el modelo estadístico matricialmente incluyendo
# los parámetros y cada una de los valores observados para las
# variables.


# Estimación puntual de los coeficientes de regresión parcial
one=rep(1, each=8)
one
X=as.matrix(data.frame(one,tele,diarios))
X
Y=ingreso
Y
beta <- solve(t(X)%*%X)    %*%       t(X)%*%Y
beta
# da lo mismo que:
modelo$coefficients
summary(modelo)


# Varianza residual
residuos=resid(modelo)
ver=(sum(residuos^2))/(length(residuos)-3)
ver
# mismo resultado que:
summary(modelo)$sigma^2


# Error estándar de los estimadores
var=ver*solve(t(X)%*%X)
var
# comparar con
vcov(modelo)
# En las diagonales están las varianzas, si obtenemos la raíz cuadrada
Std._Error_Intercept=sqrt(var[1,1])
Std._Error_tele=sqrt(var[2,2])
Std._Error_diarios=sqrt(var[3,3])

Std._Error_Intercept
Std._Error_tele
Std._Error_diarios

# O más fácil
sqrt(diag(var))
# comparar con Std. Error en:
summary(modelo)
```

```
##### DEMO GRÁFICOS ####
demo("graphics")
# algunas capacidades gráficas de R
```

8.14 Información útil para objetos con clase "lm"

```
# Generic functions for fitted (linear) model objects.

# print()                    simple printed display
# summary()                  standard regression output
# coef() (or coefficients())  extracting the regression coefficients
# residuals() (or resid())    extracting residuals
# fitted() (or fitted.values())  extracting fitted values
# anova()                    comparison of nested models
# predict()                  predictions for new data
# plot()                     diagnostic plots
# confint()                  confidence intervals for the regression coefficients
# deviance()                 residual sum of squares
# vcov() (estimated)         variance-covariance matrix
# logLik()                   log-likelihood (assuming normally distributed errors)
# AIC()                      information criteria including AIC, BIC/SBC (assuming
#                                        normally distributed errors)
```

Trabajo Práctico Nº 9: Modelos polinómicos y logarítmicos

9.1 Problema y Datos

```
# datos obtenidos de www.gapminder.org
datos=read.table("datos_p_9.txt")
colnames(datos)=c("pais","vida","pbi")
```

9.2 Primer modelo

```
# Estudiamos cómo varía la experanza de vida (años) en función del
# pbi per cápita (dólares por habitante por año)
modelo=lm(vida~pbi,data=datos)
summary(modelo)
```

```
Call:
lm(formula = vida ~ pbi, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4663 -0.2978  0.2251  0.3037  1.2403

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.6399     0.9366   4.954  0.00257 **
pbi          -0.5868     0.3578  -1.640  0.15207
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8628 on 6 degrees of freedom
Multiple R-squared:  0.3096,      Adjusted R-squared:  0.1945
F-statistic:  2.69 on 1 and 6 DF,  p-value: 0.1521
```

```
# ¿Qué conclusión obtiene de los gráficos siguientes?
plot(datos$pbi,datos$vida)
# Agregamos la línea que une los valores predichos
curve(modelo$coefficients[1]+modelo$coefficients[2]*x, add=T, col="green", lw=2)

# residuos vs. predichos
residuos=resid(modelo)
predichos=fitted(modelo)
par(mfrow=c(1,1))
plot(predichos,residuos)
abline(a=0,b=0, col="violet", lw=2)

# observados vs. predichos
plot(predichos,datos$vida)
abline(a=0,b=1, col="blue", lw=2)

# supuesto de normalidad
par(mfrow=c(2,1))
hist(residuos, col="yellow")
boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
 datos
par(mfrow=c(2,2))
hist(residuos, col="yellow")

normal=rnorm(mean=0, sd=summary(modelo)$sigma,
 n=length(summary(modelo)$residuals))
hist(normal, col="green")

boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")

par(mfrow=c(1,1))
qqnorm(residuos)
qqline(residuos)

# también se puede hacer para los residuos estandarizados
plot(modelo, which = c(2))


library(moments)
skewness(residuos) # debería dar cero
```

```
kurtosis(residuos) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos,"pnorm",mean(residuos),sd(residuos))
```

```
One-sample Kolmogorov-Smirnov test

data:  residuos
D = 0.22859, p-value = 0.7177
alternative hypothesis: two-sided
```

```
#alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) #primero hay que instalar el paquete
lillie.test(residuos)

#Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos)
```

9.3 Polinomio de segundo grado

```
datos$pbi2=datos$pbi^2

modelo2=lm(vida~pbi+pbi2,data=datos)
summary(modelo2)
anova(modelo2)
```

```
Analysis of Variance Table

Response: vida
          Df Sum Sq Mean Sq F value Pr(>F)
pbi        1 2.0026 2.00261  2.9846 0.1446
pbi2       1 1.1113 1.11129  1.6562 0.2545
Residuals  5 3.3549 0.67097
```

```
# observados vs. variable independiente.
# Además se agrega la línea que une los valores predichos
plot(datos$pbi,datos$vida)
curve(modelo2$coefficients[1]+modelo2$coefficients[2]*x+modelo2$coefficients[3]*x
 ^2
    , add=T, col="green", lw=2)

## residuos vs. predichos
residuos2=resid(modelo2)
predichos2=fitted(modelo2)
plot(predichos2,residuos2)
abline(a=0,b=0, col="violet", lw=2)

## observados vs. predichos
```

```
plot(predichos2,datos$vida)
abline(a=0,b=1, col="red", lw=2)

# normalidad
par(mfrow=c(2,1))
hist(residuos2, col="yellow")
boxplot(residuos2, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
 datos
par(mfrow=c(2,2))
hist(residuos2, col="yellow")

normal=rnorm(mean=0, sd=summary(modelo2)$sigma,
 n=length(summary(modelo2)$residuals))
hist(normal, col="green")

boxplot(residuos2, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")

par(mfrow=c(1,1))
qqnorm(residuos2)
qqline(residuos2)

# también se puede hacer para los residuos estandarizados
plot(modelo2, which = c(2))


library(moments)
skewness(residuos2) # debería dar cero
kurtosis(residuos2) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos2,"pnorm",mean(residuos2),sd(residuos2))

One-sample Kolmogorov-Smirnov test

data:  residuos2
D = 0.20404, p-value = 0.8316
alternative hypothesis: two-sided

# alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) # primero hay que instalar el paquete
lillie.test(residuos2)
```

```
Lilliefors (Kolmogorov-Smirnov) normality test

data:  residuos2
D = 0.20404, p-value = 0.421

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos2)

######### LOGARITMO ##############
# Si la relación es del siguiente modo
plot(datos$pbi,datos$vida)
# con el logaritmo podemos transformarla en lineal
plot(log10(datos$pbi),datos$vida)
# y utilizar entonces un modelo de regresión lineal simple
modelo3=lm(vida~log10(pbi),data=datos)
summary(modelo3)

# observados vs. variable independiente.
# Además se agrega la línea que une los valores predichos
plot(datos$pbi,datos$vida)
curve(modelo3$coefficients[1]+modelo3$coefficients[2]*log10(x)
      , add=T, col="green", lw=2)

## residuos vs. predichos
residuos3=resid(modelo3)
predichos3=fitted(modelo3)
plot(predichos3,residuos3)
abline(a=0,b=0, col="violet", lw=2)

## observados vs. predichos
plot(predichos3,datos$vida)
abline(a=0,b=1, col="red", lw=2)

# normalidad
par(mfrow=c(2,1))
hist(residuos3, col="yellow")
boxplot(residuos3, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
 datos
par(mfrow=c(2,2))
hist(residuos3, col="yellow")

normal=rnorm(mean=0, sd=summary(modelo3)$sigma,
 n=length(summary(modelo3)$residuals))
hist(normal, col="green")

boxplot(residuos3, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")
```

```
par(mfrow=c(1,1))
qqnorm(residuos3)
qqline(residuos3)

# también se puede hacer para los residuos estandarizados
plot(modelo3, which = c(2))


library(moments)
skewness(residuos3) # debería dar cero
kurtosis(residuos3) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos3,"pnorm",mean(residuos3),sd(residuos3))

# alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) #primero hay que instalar el paquete
lillie.test(residuos3)

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos3)
```

9.4 Polinomio de tercer grado

```
datos$pbi2=datos$pbi^2
datos$pbi3=datos$pbi^3
modelo4=lm(vida~pbi+pbi2+pbi3,data=datos)
summary(modelo4)

# observados vs. variable independiente.
# Además se agrega la línea que une los valores predichos
plot(datos$pbi,datos$vida)
curve(modelo4$coefficients[1]+modelo4$coefficients[2]*x+modelo4$coefficients[3]*x
 ^2
    +modelo4$coefficients[4]*x^3, add=T, col="green", lw=2)


# residuos vs. predichos
residuos4=resid(modelo4)
predichos4=fitted(modelo4)
plot(predichos4,residuos4)
abline(a=0,b=0, col="violet", lw=2)

## observados vs. predichos
plot(predichos4,datos$vida)
```

```
abline(a=0,b=1, col="red", lw=2)

# normalidad
par(mfrow=c(2,1))
hist(residuos4, col="yellow")
boxplot(residuos4, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
 datos
par(mfrow=c(2,2))
hist(residuos4, col="yellow")

normal=rnorm(mean=0, sd=summary(modelo4)$sigma,
 n=length(summary(modelo4)$residuals))
hist(normal, col="green")

boxplot(residuos4, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")

par(mfrow=c(1,1))
qqnorm(residuos4)
qqline(residuos4)

# también se puede hacer para los residuos estandarizados
plot(modelo4, which = c(2))


library(moments)
skewness(residuos4) # debería dar cero
kurtosis(residuos4) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos4,"pnorm",mean(residuos4),sd(residuos4))

One-sample Kolmogorov-Smirnov test

data:  residuos4
D = 0.13519, p-value = 0.9935
alternative hypothesis: two-sided

#alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) #primero hay que instalar el paquete
lillie.test(residuos4)

#Test de Shapiro-Wilk para evaluar normalidad
```

```
shapiro.test(residuos4)
```

9.5 Polinomio de grado 10

```
modelo5=lm(vida~pbi+I(pbi^2)+I(pbi^3)
        +I(pbi^4)+I(pbi^5)+I(pbi^6)+I(pbi^7)+
          I(pbi^8)+I(pbi^9)+I(pbi^10),data=datos)
summary(modelo5)
```

```
Call:
lm(formula = vida ~ pbi + I(pbi^2) + I(pbi^3) + I(pbi^4) + I(pbi^5) +
    I(pbi^6) + I(pbi^7) + I(pbi^8) + I(pbi^9) + I(pbi^10), data = datos)

Residuals:
         1          2          3          4          5          6          7
         8
 5.000e-01 -2.739e-13 -5.000e-01 -2.500e-01  2.928e-14  6.330e-13 -2.359e-15
2.500e-01

Coefficients: (5 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1430.641    762.849   1.875    0.202
pbi         -2950.745   1603.104  -1.841    0.207
I(pbi^2)     2366.932   1304.979   1.814    0.211
I(pbi^3)     -921.561    514.570  -1.791    0.215
I(pbi^4)      174.167     98.290   1.772    0.218
I(pbi^5)      -12.783      7.278  -1.757    0.221
I(pbi^6)           NA         NA      NA       NA
I(pbi^7)           NA         NA      NA       NA
I(pbi^8)           NA         NA      NA       NA
I(pbi^9)           NA         NA      NA       NA
I(pbi^10)          NA         NA      NA       NA

Residual standard error: 0.559 on 2 degrees of freedom
Multiple R-squared:  0.9034,      Adjusted R-squared:  0.6618
F-statistic:  3.74 on 5 and 2 DF,  p-value: 0.2243
```

```
# observados vs. variable independiente.
# Además se agrega la línea que une los valores predichos
plot(datos$pbi,datos$vida)
curve(modelo5$coefficients[1]+modelo5$coefficients[2]*x+
      modelo5$coefficients[3]*x^2
     +modelo5$coefficients[4]*x^3
     +modelo5$coefficients[5]*x^4
     +modelo5$coefficients[6]*x^5
     +modelo5$coefficients[7]*x^6
     +modelo5$coefficients[8]*x^7
     +modelo5$coefficients[9]*x^8
     +modelo5$coefficients[10]*x^9
     +modelo5$coefficients[11]*x^10, add=T, col="green", lw=2)
```

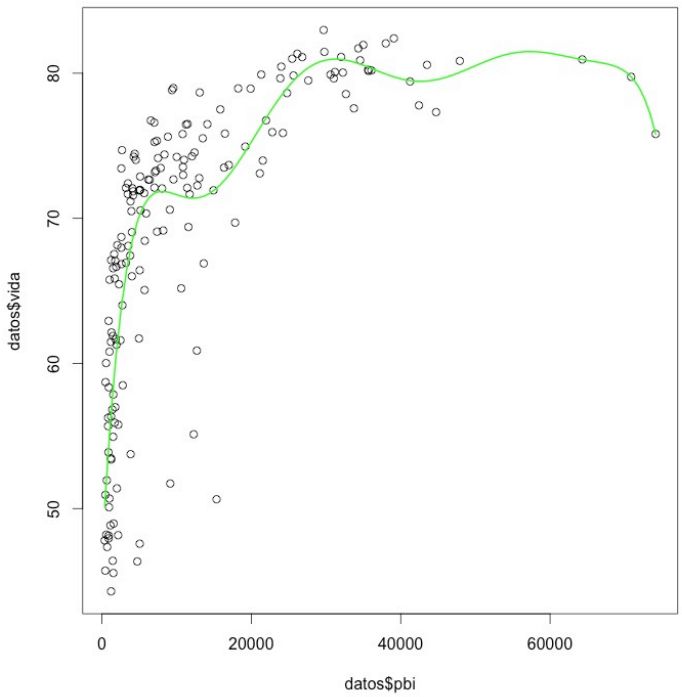


Figura 9.1: Expectativa de vida en años en función de PBI per cápita en miles de dólares. Línea de regresión del modelo polinómico de grado 10.

```
# ¿Cómo se interpretan los parámetros del modelo?
# ¿Los valores estimados de estos parámetros son generalizables? ¿Por qué?
# ¿Cuál es el aporte conceptual del modelo?


# residuos vs. predichos
residuos5=resid(modelo5)
predichos5=fitted(modelo5)
plot(predichos5,residuos5)
abline(a=0,b=0, col="violet", lw=2)

## observados vs. predichos
plot(predichos5,datos$vida)
abline(a=0,b=1, col="red", lw=2)

# normalidad
par(mfrow=c(2,1))
hist(residuos5, col="yellow")
boxplot(residuos5, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
 datos
par(mfrow=c(2,2))
hist(residuos5, col="yellow")
```

```
normal=rnorm(mean=0, sd=summary(modelo5)$sigma,
 n=length(summary(modelo5)$residuals))
hist(normal, col="green")

boxplot(residuos5, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")

par(mfrow=c(1,1))
qqnorm(residuos5)
qqline(residuos5)

# también se puede hacer para los residuos estandarizados
plot(modelo5, which = c(2))


library(moments)
skewness(residuos5) # debería dar cero
kurtosis(residuos5) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos5,"pnorm", mean(residuos5), sd(residuos5))

One-sample Kolmogorov-Smirnov test

data:  residuos5
D = 0.25, p-value = 0.6134
alternative hypothesis: two-sided

# alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) # primero hay que instalar el paquete
lillie.test(residuos5)

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos5)

##### BONDAD AJUSTE 4 MODELOS ####
bondad=data.frame(matrix(NA,ncol=6,nrow=5))
rownames(bondad)=c("lineal","grado_2","log","grado_3", "grado_10")
colnames(bondad)=c("r2","r2ajustado","DesvResid","AIC","BIC","logVEROSIMILIT
 UD")

#comparemos los r2 de los cinco modelos
bondad[1,1]=summary(modelo)$r.squared
bondad[2,1]=summary(modelo2)$r.squared
bondad[3,1]=summary(modelo3)$r.squared
bondad[4,1]=summary(modelo4)$r.squared
```

```
bondad[5,1]=summary(modelo5)$r.squared

bondad[1,2]=summary(modelo)$adj.r.squared
bondad[2,2]=summary(modelo2)$adj.r.squared
bondad[3,2]=summary(modelo3)$adj.r.squared
bondad[4,2]=summary(modelo4)$adj.r.squared
bondad[5,2]=summary(modelo5)$adj.r.squared

# comparemos sqrt(CME)
bondad[1,3]=summary(modelo)$sigma
bondad[2,3]=summary(modelo2)$sigma
bondad[3,3]=summary(modelo3)$sigma
bondad[4,3]=summary(modelo4)$sigma
bondad[5,3]=summary(modelo5)$sigma


library(nlme)
modelo=gls(vida~pbi,data=datos)
modelo2=gls(vida~pbi+pbi2,data=datos)
modelo3=gls(vida~log10(pbi),data=datos)
modelo4=gls(vida~pbi+pbi2+pbi3,data=datos)
modelo5=gls(vida~pbi+I(pbi^2)+I(pbi^3)
          +I(pbi^4)+I(pbi^5)+I(pbi^6)+I(pbi^7)+
           I(pbi^8)+I(pbi^9)+I(pbi^10),data=datos)


bondad[1,4]=summary(modelo)$AIC
bondad[2,4]=summary(modelo2)$AIC
bondad[3,4]=summary(modelo3)$AIC
bondad[4,4]=summary(modelo4)$AIC
bondad[5,4]=summary(modelo5)$AIC

bondad[1,5]=summary(modelo)$BIC
bondad[2,5]=summary(modelo2)$BIC
bondad[3,5]=summary(modelo3)$BIC
bondad[4,5]=summary(modelo4)$BIC
bondad[5,5]=summary(modelo5)$BIC

# los valores de log(verosimilitud) habitualmente son negativos ya que
# son logaritmos de valores entre cero y uno
bondad[1,6]=summary(modelo)$logLik
bondad[2,6]=summary(modelo2)$logLik
bondad[3,6]=summary(modelo3)$logLik
bondad[4,6]=summary(modelo4)$logLik
bondad[5,6]=summary(modelo5)$logLik

bondad
```

```
r2 r2ajustado DesvResid      AIC      BIC logVEROSIMILITUD
lineal   0.3095817  0.1945119 0.8627614 25.09574 24.47102        -9.547870
grado_2  0.4813747  0.2739246 0.8191285 25.55138 23.98913        -8.775690
```

```
log       0.3966378  0.2960775 0.8065357       NA       NA               NA
grado_3   0.6506383  0.3886171 0.7516537 24.94034 21.87181        -7.470171
grado_10  0.9033816  0.6618357 0.5590170       NA       NA               NA
```

```
with(bondad, plot(DesvResid,r2))
abline(lm(r2~DesvResid, data=bondad), col="blue", lw=3)

with(bondad, plot(logVEROSIMILITUD,r2))
# Bajo el supuesto de normalidad los resultados por mínimos
# cuadrados coinciden con los de máxima verosimilitud.
# Pero como vimos en la evaluación de supuestos, este no es el caso!
```

Volver al Índice principal

Modelos lineales generales

Trabajo Práctico Nº 10: Modelo de regresión con variables categóricas

10.1 Problema y Datos

```
# datos obtenidos de www.gapminder.org
datos=read.table("datos_p_10.txt",dec=",")
colnames(datos)=c("pais","año","vida","fertilidad")
View(datos)
str(datos)

datos=na.omit(datos)

modelo=lm(vida~fertilidad*factor(año),data=datos)

# 1) Observados vs. predichos
plot(fitted(modelo),datos$vida)
abline(a=0,b=1, col="red", lw=3)
# Interpretar gráfico y definir bondad de ajuste.


# 2) Observados vs. fecundidad según año con ajuste del modelo
a1950=subset(datos,año==1950)
a2009=subset(datos,año==2009)
plot(a1950$fertilidad,a1950$vida, col="brown", ylim=c(25,90),xlim=c(1,9), pch=19,
cex=0.8,
     ylab="Esperanza de vida (años)",xlab="Fecundidad (no. hijos por mujer)")
points(a2009$fertilidad,a2009$vida, col="blue", pch=19, cex=0.8)
lines(a1950$fertilidad,predict(modelo,a1950), col="brown", lwd=4)
lines(a2009$fertilidad,predict(modelo,a2009),col="blue",lwd=4)
legend("topright",legend=c("1950","2009"),col=c("brown","blue"),lty=1,lwd=4)
```

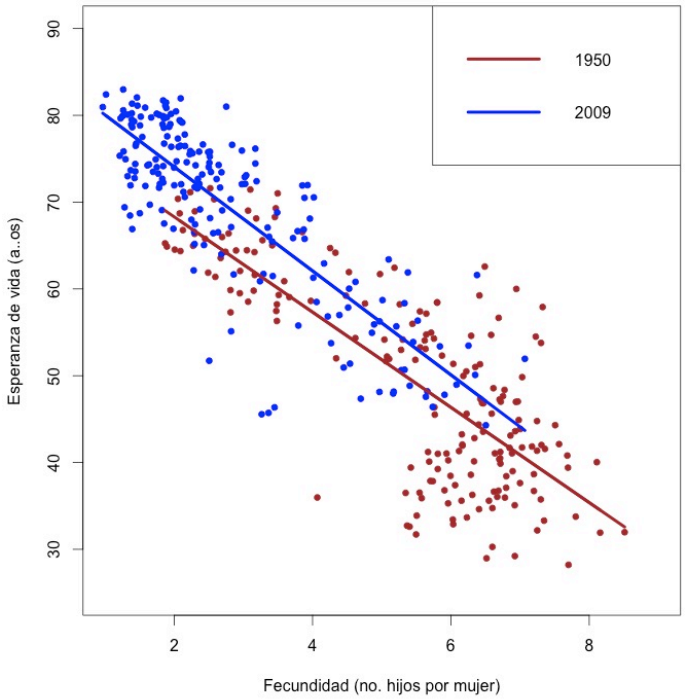


Figura 10.1: Esperanza de vida (en años) en función de fecundidad (en números de hijos por mujer) según los años 1950 y 2009.

# 3) Interpretar el summary y el anova en el contexto de la figura anterior
summary(modelo)

```
Call:
lm(formula = vida ~ fertilidad * año, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-20.9840  -3.7259   0.6663   4.4355  18.9091

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         79.2710     1.6352  48.476  < 2e-16 ***
fertilidad          -5.4860     0.2876 -19.072  < 2e-16 ***
año1970              2.7429     2.1153   1.297 0.195274
año2009              6.7576     1.9484   3.468 0.000564 ***
fertilidad:año1970   0.8221     0.3785   2.172 0.030258 *
fertilidad:año2009  -0.5021     0.4418  -1.137 0.256184
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.532 on 563 degrees of freedom
Multiple R-squared:  0.7712,      Adjusted R-squared:  0.7692
F-statistic: 379.6 on 5 and 563 DF,  p-value: < 2.2e-16
```

anova(modelo)

```
Analysis of Variance Table
```

```
Response: vida
               Df Sum Sq Mean Sq   F value    Pr(>F)
fertilidad      1  75511   75511 1769.8074 < 2.2e-16 ***
año             2   4986    2493   58.4339 < 2.2e-16 ***
fertilidad:año  2    478     239    5.5979  0.003915 **
Residuals     563  24021      43
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

10.2 Consignas a resolver

4) Escribir el modelo estadístico e interpretar cada uno de sus parámetros
en el contexto del problema. Indicar las unidades de los parámetros.
Indicar el rango del modelo

5) Escribir la función de verosimilitud para el modelo planteado.

6) Evalúe los supuestos del modelo

7) Obtenga el residual de la observación número 27. Indique sus unidades e interprete en términos del problema.

Seguimos con este ejemplo más adelante####

Trabajo Práctico Nº 11: Un ejemplo de utilización de variables dummies

11.1 Problema y Datos

Los datos fueron obtenidos de Roberto Kozulj (elaboración propia)

```
datos = read.table("datos_p_11.txt", header=T, dec=",")
head(datos)
str(datos)
```

Se desea conocer como el Saldo de Balanza Comercial (SBC) ("serie
explicada", "variable respuesta") varía en función de la tasa de cambio
real (TCR) (nivel de apreciación-depreciación de la moneda). Se propone
que cuanto menor sea la tasa de cambio real (mayor apreciación) menor
será el SBC (Exportación - Importación). A su vez, se plantea que la
implementación del Plan Real por parte de Brasil en 1995 implicó cambios
estructurales en la economía Argentina afectando el SBC y la relación de
éste con el TCR. Otro cambio estructural para la economía Argentina fue
el ingreso de China a la OMC, lo cual implico un significativo aumento en
la demanda internacional de productos básicos con sus consiguientes
efectos sobre el comportamiento del SBC de Argentina.

Se proponen 3 cálculos distintos de la TRC en función de las distintas
estimaciones de la inflación surgidas en el último decenio. La TCR

```
# incopora la diferencia de precios entre algun país de referencia
# (e.g. EEUU) y la Argentina. Relaciona el costo de una canasta de
# productos básicos en ambos países.
# TCR = tipo de cambio nominal * precio internacional (e.g. dólar) / precio local
 (pesos)
# Se eligió base 1986 porque hubo estudio de CEPAL para fijar en ese año
# el tipo de cambio en un nivel de equilibrio.

# A continuación se describen con mayor detalle cada una de las variables analizadas:

# año

# TCR_38_Indec --- Tasa de Cambio real ajustada por IPC-2002 (IPC es Índice de
 Precios al consumidor)
# (oversht.con 38.7% de inflación). Inflación según INDEC

# TCR_38_Consultoras --- Tasa de Cambio real ajustada por IPC-2002
# (oversht.con 38.7% de inflación). Inflación según Consultoras

# TCR_80_Consultoras --- Tasa de Cambio real ajustada por IPC-2002
# (overshot. con 80% de inflación). Inflación según Consultoras

# SBC --- SBC mercancias = Exportación - Importación FOB (Valor en u$s millones)

# Plan_Real --- Plan Real en Brasil

# Ingreso_China --- Ingreso China en OMC

# Las series fueron elaboradas por Roberto Kozulj a partir de varios números
# del Boletin Informativo Techint entre otras fuentes.

### Estructuralismo Económico ###
# La corriente "estructuralista" en economía propone que la relación entre
# algunas variables macroeconómicas depende de condiciones estructurales de
# la economía que se analiza. Así, por ejemplo, en los 90´ la política
# económica generaba un marco estrutural que implicaba una relación inversa
# entre el Consumo y el SBC Argentina. Al estar las importaciones libres de
# aranceles y el TC sobreevaluado, el aumento en el consumo Argentino
# implicaba un aumento en las importaciones y por ende una caída en el SBC.
# Bajo el actual modelo económico se busca generar las condiciones
# estructurales a nivel macro para invertir esa relación. Al cerrar las
# fronteras a la importación indiscriminada se busca que aumentos en la
# capacidad de consumo de los argentinos se vuelquen a productos nacionales
# y no tengan un efecto directo sobre el SBC.

# Éste tipo de análisis puede ser integrado a la estadística mediante la
# incorporación de variables categóricas a un modelo de regresión lineal.
# En muchos textos de estadística aplicados a la economía a estas variables
# se las conoce como variables "dummy" (categóricas), y a estos modelos como
# modelos de cambio estructural.
```

11.2 Consignas a resolver

```
# 1) Plantee el modelo estadístico de acuerdo al Marco conceptual propuesto
# más arriba (tome sólo una de las estimaciones del TCR).

modelo=lm(SBC~TCR_38_Indec*Plan_Real*Ingreso_China,data=datos)
summary(modelo)
# Conviene reescribir el modelo del siguiente modo
modelo=lm(SBC~TCR_38_Indec+Plan_Real+Ingreso_China+TCR_38_Indec:Plan_Re
al+TCR_38_Indec:Ingreso_China,data=datos)
# ya que no hay datos para considerar la interacción entre el ingreso de China y el plan
 real
summary(modelo)
```

```
Call:
lm(formula = SBC ~ TCR_38_Indec + Plan_Real + Ingreso_China +
    TCR_38_Indec:Plan_Real + TCR_38_Indec:Ingreso_China, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-189.79  -66.71  -14.31   46.30  348.81

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               -130.897     61.061  -2.144  0.03789 *
TCR_38_Indec                 2.128      0.665   3.200  0.00262 **
Plan_Real                 -445.753    129.674  -3.438  0.00134 **
Ingreso_China             1009.916    349.597   2.889  0.00609 **
TCR_38_Indec:Plan_Real      10.915      2.054   5.315 3.81e-06 ***
TCR_38_Indec:Ingreso_China  -9.920      4.390  -2.260  0.02910 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 107.7 on 42 degrees of freedom
Multiple R-squared:  0.885,    Adjusted R-squared:  0.8713
F-statistic: 64.67 on 5 and 42 DF,  p-value: < 2.2e-16
```

```
# 2) Reflexione ¿que cambios estructurales propone el modelo conceptual?
# ¿Como los modelamos en nuestro modelo estadístico?

# 3) Constuya un gráfico que permita ver los distintos escenarios
# contemplados por su modelo estadístico.

# Primero separamos los datos de acuerdo a las distintas situaciones
# estructurales
chinaybrasil=subset(datos,Ingreso_China==1)
# Situación en la que China entró a la OMC y Brasil ya implementó el Plan Real

brasil=subset(datos,Plan_Real==1)
brasilnochina=subset(brasil,Ingreso_China==0)
# Situación en la que Brasil implemento el Plan Real pero China aún no
```

```
# ingresa a la OMC

nobrasil=subset(datos,Plan_Real==0)
# Situación en la que ni China ingresó a la OMC ni Brasil implementó el Plan Real

plot(chinaybrasil$TCR_38_Indec,chinaybrasil$SBC, col="brown", pch=19,
    ylim=c(-200,900),xlim=c(30,180), ylab="SBC",xlab="TCR")
# Primero graficamos la Situación en la que China entro a la OMC y Brasil
# ya implementó el Plan Real.
# Establecemos los límites "ylim" y "xlim" mirando:
range(datos$SBC)

[1] -194.5  870.7

range(datos$TCR_38_Indec)

[1]  37.1 170.7

points(brasilnochina$TCR_38_Indec,brasilnochina$SBC, col="blue", pch=19)
# Luego agregamos los puntos que corresponden a la segunda situación

points(nobrasil$TCR_38_Indec,nobrasil$SBC, col="red", pch=19)
# Y finalmente los correspondientes a la tercera.

# Cada una de estas situaciones las modelaremos mediante una recta
# diferente pero sólo con una ecuación.
lines(chinaybrasil$TCR_38_Indec,predict(modelo,chinaybrasil), col="brown", lwd=2)
# Recta pertenenciente a la primer situación
lines(brasilnochina$TCR_38_Indec,predict(modelo,brasilnochina),col="blue", lwd=2)
# Recta pertenenciente a la segunda situación
lines(nobrasil$TCR_38_Indec,predict(modelo,nobrasil),col="red",lwd=2)
# Recta pertenenciente a la última situación
legend("topright",legend=c("China-OMC","Brasil-
Real","NiChina_niBrasil"),col=c("brown","blue","red"),lty=1,lwd=2)
```

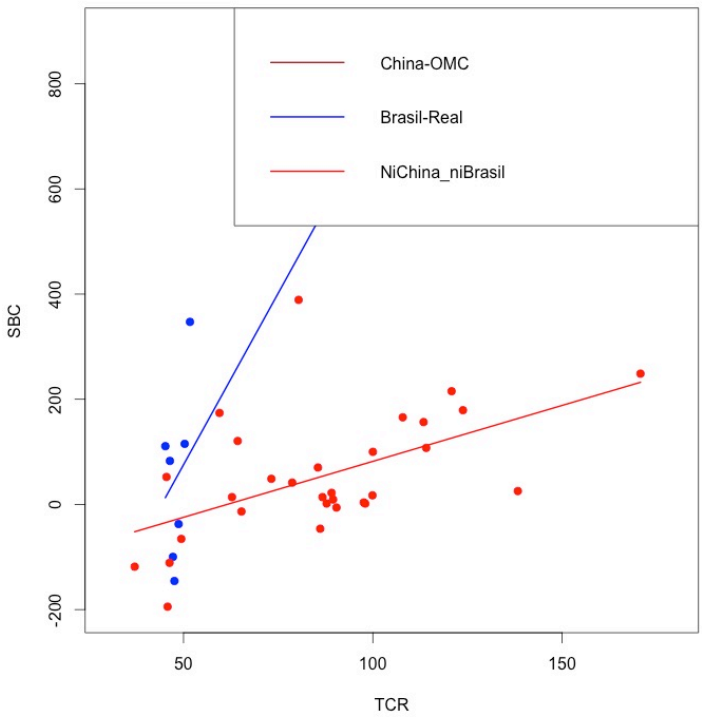


Figura 11.1: Saldo de la Balanza Comercial según Tipo de Cambio de Real en diferentes contextos (años donde China perteneció a la OMC, años donde Brasil tenía el Plan Real, años donde ambos factores anteriores estaban ausentes)

```
# Observando el gráfico responda:
# - ¿El ingreso de China a la OMC implicó un cambio en el SBC Argentina?
# - ¿Que parámetro modela dicho salto?
# - ¿La implementación del plan real por parte de Brasil implico un cambio
# significativo en la relación entre el SBC y el TCR?
# - ¿Que parámetro modela dicho cambio en la relación entre SBC y TCR
# fruto del Plan Real?

# 4) Elabore 3 modelos: uno con cada estimación del TCR y luego evalue la
# bondad de ajuste de cada modelo. Concluya

# 5) Indague sobre la presencia de valores extremos en su modelo. En caso de
# haberlos re-escriba el modelo quitando ese valor y vuelva a realizar el
# gráfico de arriba. Concluya sobre el efecto de esa observación sobre sus
# estimaciones. ¿Que haría al respecto?

# 6) Evalúe los supuestos del modelo. ¿Se trata de datos en corte
# transversal o de una serie de tiempo? ¿Que características presentan
# comunmente las series de tiempo? ¿Nuestro modelo es adecuado?

# 7) Escriba matricialmente el modelo.
```

Trabajo Práctico Nº 12: Análisis de corte trasversal con diferentes años como factor

12.1 Problema y Datos

```
# Se desea conocer cómo es la esperanza de vida en países que presentan
# distintos niveles de fecundidad, y cómo cambió esta relación a lo largo
# del tiempo.
datos=read.delim("datos_p_12.txt",dec=",",header=F)
colnames(datos)=c("pais","año","vida","fertilidad")
datos=na.omit(datos)
str(datos)
summary(datos)

datos$año=as.factor(datos$año)
summary(datos)

modelo=lm(vida~fertilidad*año,data=datos)
summary(modelo)
```

```
Call:
lm(formula = vida ~ fertilidad * año, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-20.9840  -3.7259   0.6663   4.4355  18.9091

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         79.2710     1.6352  48.476  < 2e-16 ***
fertilidad          -5.4860     0.2876 -19.072  < 2e-16 ***
año1970              2.7429     2.1153   1.297 0.195274
año2009              6.7576     1.9484   3.468 0.000564 ***
fertilidad:año1970   0.8221     0.3785   2.172 0.030258 *
fertilidad:año2009  -0.5021     0.4418  -1.137 0.256184
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.532 on 563 degrees of freedom
Multiple R-squared:  0.7712,      Adjusted R-squared:  0.7692
F-statistic: 379.6 on 5 and 563 DF,  p-value: < 2.2e-16
```

12.2 Consignas a resolver

```
# 1) Escribir modelo e interpretarlo con la ayuda del gráfico abajo

# 2) Escribir el modelo matricialmente, incluyendo los 5 primeros valores
# observados.

# observados vs. fertilidad según año con ajuste del modelo
a1950=subset(datos,año==1950)
```

```
a1970=subset(datos,año==1970)
a2009=subset(datos,año==2009)
plot(a1950$fertilidad,a1950$vida, col="brown", ylim=c(25,90),xlim=c(1,9),
    ylab="Esperanza de vida (años)",xlab="Fecundidad (no. hijos por mujer)")
points(a1970$fertilidad,a1970$vida,col="green")
points(a2009$fertilidad,a2009$vida, col="blue")
lines(a1950$fertilidad,predict(modelo,a1950), col="brown", lwd=2)
lines(a1970$fertilidad,predict(modelo,a1970), col="green", lwd=2)
lines(a2009$fertilidad,predict(modelo,a2009),col="blue",lwd=2)
legend("topright",legend=c("1950","1970", "2009"),col=c("brown","green",
"blue"),lty=1,lwd=2)
```

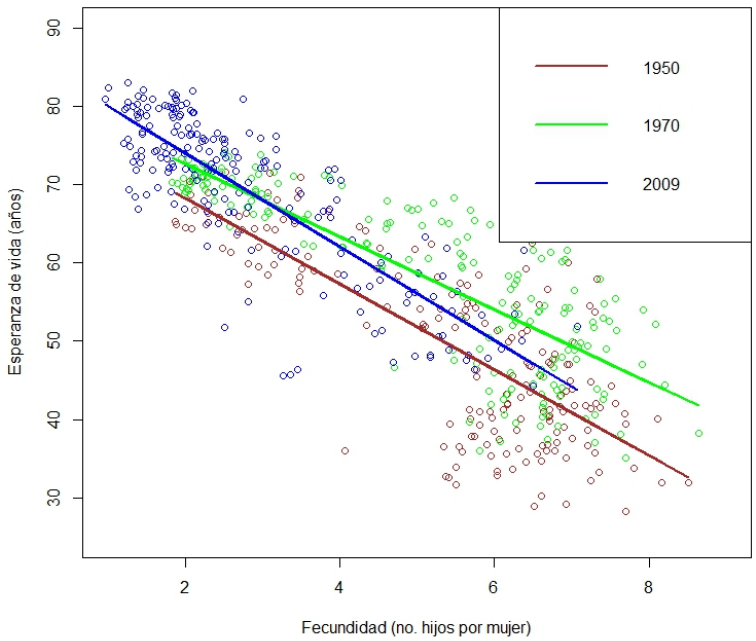


Figura 12.1: Esperanza de vida (en años) en función de Fecundidad (en cantidad de hijos por mujer) según los años 1950, 1970 y 2009.

```
# 3) Interpretar ANOVA. Indique las unidades del estadístico F.
anova(modelo)
```

```
Analysis of Variance Table

Response: vida
               Df Sum Sq Mean Sq   F value    Pr(>F)
fertilidad      1  75511   75511 1769.8074 < 2.2e-16 ***
año             2   4986    2493   58.4339 < 2.2e-16 ***
fertilidad:año  2    478     239    5.5979  0.003915 **
Residuals     563  24021      43
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# 4) ¿Cuál es el valor esperado de vida en un país durante 1970 y cuya
# fecundidad promedio es de 7 hijos?
```

```
# 5) ¿Cuál es el valor esperado de vida en un país durante 2009 y cuya
# fecundidad promedio es de 8 hijos?

# 6) ¿Cuál es el valor esperado de vida en un país durante 2009 y cuya
# fecundidad promedio es de 4 hijos?

# 7) ¿Cuál es el valor esperado de vida en un país durante 1950 y cuya
# fecundidad promedio es de 3 hijos?
```

12.3 Intervalos de predicción y confianza

```
# 8) INTERVALOS DE PREDICCIÓN Y DE CONFIANZA #
# Intervalos de confianza para la ordenada y los coeficientes de regresión
# parciales
confint(modelo, level=0.95)
# Interpretar

# pedimos el intervalo de confianza para la media de esperanza de vida
# dado cierto año y valor de hijos por mujer.
predict.lm(modelo,interval="confidence")
# Interpretar.

# ¿Cuál es el intervalo de confianza para la media de esperanza de vida en
# el 2009 para un país con 3 hijos por mujer en promedio?
newdata=data.frame(año=2009,fertilidad=3)
newdata$año=as.factor(newdata$año)
predict(modelo,newdata,interval="confidence")
```

```
fit      lwr    upr
1 68.06427 67.13153 68.997
```

```
# Indicar unidades. Interpretar.

# Ahora el intervalo de predicción para la misma situación que arriba:
predict.lm(modelo,newdata,interval="prediction")
```

```
  fit      lwr     upr
1 68.06427 55.20043 80.9281
```

```
# Explicar las diferencias entre el intervalo de predicción y de confianza.

# 9) Evaluar el cumplimiento de los supuestos

# 10) Estime el r2 y el AIC del modelo y explique sus diferencias.
AIC(modelo)
```

```
[1] 3758.412
```

Volver al Índice principal

Modelos lineales generales con heterogeneidad de varianzas

Trabajo Práctico Nº 13: Varianzas en función de variable independiente categórica

13.1 Problema y Datos

```
# Los datos fueron obtenidos de Banco Mundial (http://www.worldbank.org/)

datos=read.table("datos_p_13.txt", dec=",")
colnames(datos)=c("pais","ocupacion","region")
str(datos)
# Se desea evaluar si los niveles de ocupación varían según
# regiones en el mundo.

# Ocupación = % de personas mayores de 15 años que han sido empleada durante el
2007

# ¿Cuál es la Unidad Experimental, la muestra y la población?

library(nlme)
```

13.2 Modelo de varianzas homogéneas

```
modelo=gls(ocupacion~region, data=datos)
# Plantear matricialmente este modelo
```

```
# e indicar los primeros 5 valores de cada matriz o vector.

anova(modelo)

par(mfrow=c(1,1))
boxplot(datos$ocupacion~datos$region)


datos$residuos=resid(modelo)
plot(datos$region,datos$residuos)

# otra opción para el gráfico de caja y bigotes ("boxplot" en inglés)
boxplot(datos$residuos~datos$region)
# el largo de los bigotes está definido por 1,5 veces la amplitud intercuartilar.
# Sin embargo, los bigotes pueden ser asimétricos (como en Asia) porque el final del
# bigote se ubica en el punto más extremo dentro de este rango. Todos aquellos valores
# que estén por fuera de 1,5 * Rango Intercuartilar se grafican como puntos.

# El tamaño de los bigotes puede modificarse con el código "range" como vemos a
continuación
boxplot(datos$residuos~datos$region, range =0.5)
# largo bigote = 0.5 * Rango intercuartilar. ¿Cuáles cambios se observan en el gráfico?

# Sucede lo opuesto si damos un valor más alto a range
boxplot(datos$residuos~datos$region, range = 2)

# Si queremos que el límite inferior y superior de los bigotes sean los valores mínimos
# y máximos en cada región ponemos range = 0
boxplot(datos$residuos~datos$region, range = 0)


tapply(datos$residuos,datos$region,sd)


plot(fitted(modelo),resid(modelo))
abline(a=0,b=0, col="violet", lw=3)


# otro gráfico interesante (sin acentos en las secuencias)
medias=with(datos,tapply(ocupacion,region,mean))
medias
desvios=with(datos,tapply(ocupacion,region,sd))
desvios
# ojo que estamos graficando el desvío estándar y no el error estándar.
# ¿Cuál es la diferencia entre ambos?

fig = barplot(medias,ylim=c(0,80),ylab= "personas ocupadas (% población mayor a
15)")
arrows(fig,medias+desvios,fig,medias-desvios, angle=90,code=3)
# ¿Por qué ambas secuencias dan los mismos valores?
```

```
with(datos,tapply(ocupacion,region,sd))
tapply(datos$residuos,datos$region,sd)
```

13.3 Modelo de varianzas heterogéneas según región

```
# Var[i] = (DesvíoEstándar * Beta[i])^2

# DesvíoEstándar = de una región que se toma como base.
# En el summary se presenta como "Residual standard error"

# Beta[i] = hay tantos Betas como grupos menos 1.
# Es el desvío estándar del grupo de interés dividido el
# desvío estándar del grupo que se toma como base.

modident = gls(ocupacion~region, weights=varIdent(form=~ 1|region),data=datos)
# Escribir la función de verosimilitud para este modelo y el anterior.
# Explicar las diferencias.

summary(modident)
```

```
Generalized least squares fit by REML
  Model: ocupacion ~ region
  Data: datos
       AIC      BIC    logLik
  384.4923 403.8105 -182.2461

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | region
 Parameter estimates:
        Asia        Europa        Africa LatinoAmerica    NorAmerica
   1.0000000     0.8911821     1.5186985     0.7260788     0.2920424

Coefficients:
                      Value Std.Error   t-value p-value
(Intercept)          64.76316  2.615139 24.764706  0.0000
regionAsia           -0.49316  3.531676 -0.139638  0.8895
regionEuropa        -12.67030  3.167798 -3.999719  0.0002
regionLatinoAmerica  -5.71770  3.088531 -1.851269  0.0699
regionNorAmerica      0.28684  3.039975  0.094357  0.9252

 Correlation:
                    (Intr) regnAs rgnErp rgnLtA
regionAsia          -0.740
regionEuropa        -0.826  0.611
regionLatinoAmerica -0.847  0.627  0.699
regionNorAmerica    -0.860  0.637  0.710  0.728

Standardized residuals:
        Min          Q1         Med          Q3         Max
-1.96183039 -0.59588840 -0.07217778  0.67091902  2.08346365

Residual standard error: 7.505854
Degrees of freedom: 56 total; 51 residual
```

```
# interpretar el summary en conjunto con el gráfico realizado
# arriba ("barplot")


# Según este modelo, ¿cuál es la varianza en la ocupación
# entre países en Latino América?


# comparamos con el modelo que supone homogeneidad de varianzas
anova(modelo,modident)
```

```
Model df      AIC      BIC    logLik   Test  L.Ratio p-value
modelo       1  6 386.6105 398.2015 -187.3053
modident     2 10 384.4923 403.8105 -182.2461 1 vs 2 10.11826  0.0385
```

```
# el modelo con homogeneidad de varianzas está
# anidado dentro del "modident". ¿Por qué?


# verificamos que se solucionó el problema de heterogeneidad de varianzas.
datos$res_id=resid(modident,type="pearson")

plot(datos$region,datos$res_id)

tapply(datos$res_id,datos$region,sd)
# ¡Copado!

plot(fitted(modident),datos$res_id)
abline(a=0, b=0, col="violet", lw=3)


# comparamos la normalidad para los residuos entre los modelos
# con y sin heterogeneidad de varianzas
layout(matrix(1:2,1,2))
qqnorm(datos$residuos, main="SinModVar")
qqline(datos$residuos)
qqnorm(datos$res_id, main="ConModVar")
qqline(datos$res_id)
```

13.4 Prueba del cociente de verosimilitudes

```
# volvamos al anova
anova(modelo,modident)
# Plantear la H_0) y la H_alt)

# L.Ratio:
L.Ratio = - 2 * (-187.3053 - (-182.2461))
L.Ratio
```

```
# en este caso los grados de libertad son 4,
# que es la diferencia en el número de parámetros.
# ¿Cuáles son los parámetros que difieren entre ambos modelos?

# Podemos entonces calcular la probabilidad de encontrar un valor de L.Ratio
# de 10.12 o mayor si la hipótesis nula que indica que no hay diferencias entre
# la verosimilitud de los modelos sin y con heterogeneidad de varianzas fuese cierta.
1 - pchisq(L.Ratio,4)
# Vemos que da lo mismo que el valor obtenido a partir de la función
anova(modelo,modident)

# El LRT sólo sirve para comparar modelos anidados.

# Ver
?pchisq
# Esta distribución la vamos a ver con más detalle luego en el curso.

# AIC
# modelo sin heterogeneidad de varianzas
-2 * -187.3053 + 2 * 6
# Son 6 parámetros porque hay 5 regiones más una varianza.
# Recordemos que en la función de verosimilitud utilizando
# la distribución normal utilizamos tanto la media como la varianza.

# modelo con heterogeneidad de varianzas
-2 * -182.2461 + 2 * 10
# Son 10 parámetros porque estimamos una media y una varianza
# para cada una de las 5 regiones


# BIC
# modelo sin heterogeneidad de varianzas
-2 * -187.3053 + 6 * log(51)
# log(51) porque con el método REML se usa n - p (parámetros del componente fijo) =
51,
# osea los grados de libertad residuales.
# Recordar que en R "log" es ln, a menos que se especifique lo contrario,
# por ejemplo "log10"

# modelo con heterogeneidad de varianzas
-2 * -182.2461 + 10 * log(51)

# como es de esperar el BIC favorece modelos
# más "simples" (con menos parámetros) que el AIC



# Interesante saber el tipo de funciones disponibles para modelar la varianza
?varClasses
```

```
# Recuerden lo que dice al final respecto que pueden incoporar cualquier función que
# ustedes deseen simplemente con un poco más de notación.
```

Trabajo Práctico Nº 14: Varianzas en función de variable independiente cuantitativa

14.1 Problema y Datos

```
# datos obtenidos de www.gapminder.org
datos=read.table("datos_p_9.txt")
colnames(datos)=c("pais","vida","pbi")
# Recordemos que estudiamos cómo varía la experanza de vida (años)
# en función del pbi per cápita (dólares por habitante por año)

# ¿Cuál es la unidad experimental?
```

14.2 Modelo con varianzas homogéneas usando “lm”

```
# continuamos con el mejor de los modelos encontrados en el práctico 9
modelo3=lm(vida~log10(pbi),data=datos)
summary(modelo3)


# observados vs. variable independiente
plot(datos$pbi,datos$vida)
# Agregamos la línea que une los valores predichos
curve(modelo3$coefficients[1]+modelo3$coefficients[2]*log10(x), add=T,
col="green", lw=2)


# residuos vs. predichos
residuos3=resid(modelo3)
predichos3=fitted(modelo3)
plot(predichos3,residuos3)
abline(a=0,b=0, col="violet", lw=2)


# observados vs. predichos
plot(predichos3,datos$vida)
abline(a=0,b=1, col="blue", lw=2)
```

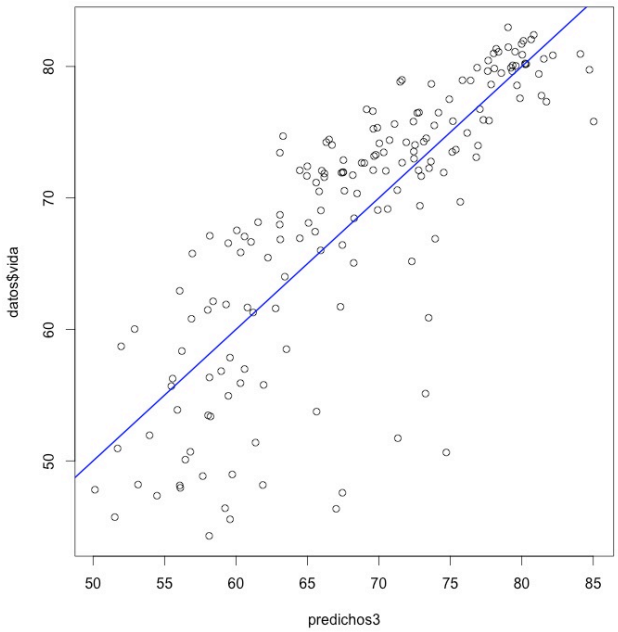


Figura 14.1:Esperanza de vida en años en función de los valores predichos para la esperanza de vida en años según el modelo estimado

```
# Aparentemente los gráficos indican que:
# a mayor pbi mayor media para la esperanza de vida
# a mayor pbi menor varianza para la esperanza de vida
# Es interesante entonces modelar ambas, la media y la varianza.


# Sin embargo también se ve asimetría en los siguientes gráficos y análisis,
# aspecto que no podría solucionarse modelando sólamente varianzas distintas
# para distribuciones simétricas, en este caso distribuciones normales.
par(mfrow=c(2,1))
hist(residuos3, col="yellow")
boxplot(residuos3, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
datos
par(mfrow=c(2,2))
hist(residuos3, col="yellow")

normal=rnorm(mean=0, sd=summary(modelo3)$sigma,
n=length(summary(modelo3)$residuals))
hist(normal, col="green")

boxplot(residuos3, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")

par(mfrow=c(1,1))
qqnorm(residuos3)
```

```
qqline(residuos3)

# también se puede hacer para los residuos estandarizados
plot(modelo3, which = c(2))


library(moments)
skewness(residuos3) # debería dar cero
kurtosis(residuos3) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos3,"pnorm",mean(residuos3),sd(residuos3))

# alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) #primero hay que instalar el paquete
lillie.test(residuos3)

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos3)
```

14.3 Modelo con varianzas homogéneas utilizando "gls"

```
# Como lo hemos hecho en clases anteriores, recurrimos a los gls, primero ajustando
# un modelo igual al modelo3 anterior, es decir un modelo que supone varianzas
homogéneas
library(nlme)
modgls= gls(vida~log10(pbi),data=datos)

# ¿Que hace la función gls? Ver
?gls
# gls por "generalized least squares" estimados mediante máxima verosimilitud
restringida (REML)
# Utiliza el algoritmo de Newton-Raphson.
# gls entonces permite estimar modelos lineales generales
# felixibilizando la matriz de varianzas y covarianzas,
# permitiendo estimar modelos de heterogeneidad de varianzas (sí, los veremos en este
curso)
# así como modelos de auto-correlación temporal y espacial entre los residuos (no los
veremos en este curso).
# La función "lm" estima modelos lineales generales los cuáles suponen
# homogeneidad de varianzas y residuos independientes (correlación = 0).


# corroboramos que da lo mismo que el modelo3
summary(modgls)
summary(modelo3)
```

```
anova(modgls,modelo3)
# aprovechamos para practicar la fórmula de AIC y BIC
# AIC:
-2*summary(modgls)$logLik + 2 * 3
# en R log es el logaritmo natural
# el modelo consta de tres parámetros:
# - ordenada al origen
# - pendiente
# - varianza

# BIC:
-2*summary(modgls)$logLik + log(length(datos[,1])-2)*3
```

14.4 Modelo de varianza fijada

```
# Incorporamos un modelo que supone que la varianza varía proporcionalmente con el
pbi:
# Var[i] = Var * PBI[i]

# 1) Un aspecto interesante de este modelo es que no incorpora nuevos parámetros

# 2) PBI debería ser positivo ya que las varianzas son siempre positivas

# 3) Asume una relación lineal positiva entre Var[i]
# y PBI[i] con ordenada al origen igual a cero y
# pendiente igual a Var


# Escribir el modelo estadístico completo para:
# .- El caso con varianzas homogéneas presentado arriba
# .- El caso con varianzas heterogéneas (fijadas) presentado ahora
# Explicar las diferencias entre los modelos.

modvarfija= gls(vida~log10(pbi), weights=varFixed(~pbi), data=datos)

summary(modvarfija)
```

```
Generalized least squares fit by REML
  Model: vida ~ log10(pbi)
  Data: datos
      AIC      BIC    logLik
  1288.99 1298.485 -641.4951

Variance function:
 Structure: fixed weights
 Formula: ~pbi

Coefficients:
                Value Std.Error   t-value p-value
(Intercept)  5.786515  3.477200  1.664131  0.0979
log10(pbi)  16.740421  1.088318 15.381917  0.0000
```

```
 Correlation:
           (Intr)
log10(pbi) -0.991

Standardized residuals:
        Min          Q1         Med          Q3         Max
-3.18576450 -0.25024408  0.01341118  0.42033523  3.15053491

Residual standard error: 0.118227
Degrees of freedom: 177 total; 175 residual
```

Aparece un nuevo término en el modelo que hace referencia al modelo de varianza ajustado

El summary presenta los residuos estandarizados.
En este caso cada residuo (valor observado - esperado)
se divide por el desvío estándar que le corresponde según su PBI.

¿Cuál es la varianza en la esperanza de vida para países con PBI = 5000?
modvarfija$sigma^2*5000
¿Cuáles son las unidades? Interpretar

¿Cuál es la varianza en la esperanza de vida para países con PBI = 14000?

anova(modgls, modvarfija)

```
Model         df      AIC      BIC    logLik
modgls         1  3 1144.698 1154.192 -569.3490
modvarfija     2  3 1288.990 1298.485 -641.4951
```

comparamos este modelo con el anterior que consideraba la
misma varianza para todas las observaciones y vemos que el ajuste empeora sensiblemente

Esto es lógico ya que en los gráficos anteriores observábamos MENOR variabilidad a
MAYOR PBI. Ello es lo opuesto a lo que pusimos en el modelo, en el que indicamos
que la variabilidad es MAYOR a MAYOR PBI.

# Entonces, para jugar un poco podemos hacer lo siguiente:
pbi2 = 1 / datos$pbi
modvarfija2= gls(vida~log10(pbi), weights=varFixed(~pbi2), data=datos)
anova(modgls, modvarfija, modvarfija2)

```
Model          df      AIC      BIC    logLik
modgls          1  3 1144.698 1154.192 -569.3490
modvarfija      2  3 1288.990 1298.485 -641.4951
modvarfija2     3  3 1177.261 1186.755 -585.6303
```

¿Que pasó?

14.5 Modelo de varianza como potencia de la variable independiente

```
# Var[i] = var * |PBI[i]| ^ (2*Gamma)
# 1) Si Gamma = 0 estamos ante un caso de homogeneidad de varianzas
# Por eso se dice que el "modgls" está anidado dentro del "modvarpotencia"
# y se pueden comparar ambos modelos con el test de cociente de verosimilitudes.

# Si Gamma = 0,5 como está multiplicado por 2 se obtiene la misma relación que
# con el modelo de varianza fijada presentado anteriormente. Sin embargo, el modelo
# de varianza fijada no está anidado dentro del "modvarpotencia" porque el
# parámetro Gamma sigue siendo parte del modelo (Gamma = 0,5).

# 2) Es un modelo muy flexible ya que representa un polinomio

# 3) No debería ser utilizado si PBI = 0, por ejemplo, dado que las varianzas son
siempre
# positivas.

modvarpotencia= gls(vida~log10(pbi), weights=varPower(form=~pbi), data=datos)
anova(modgls, modvarpotencia)
```

```
Model             df      AIC      BIC    logLik   Test  L.Ratio p-value
modgls             1  3 1144.698 1154.192 -569.3490
modvarpotencia     2  4 1137.682 1150.341 -564.8412 1 vs 2 9.015644  0.0027
```

```
# Vemos en cambio que este modelo con varpotencia tiene mejor ajuste que aquel que
# considera varianzas homogéneas.

# Como modgls está anidado dentro del modvarpotencia ahora el comando anova
# nos da un contraste de hipòtesis utilizando el estadístico que surge
# del cociente de verosimilitudes entre ambos modelos (Lratio).

# Plantear las hipótes que se están contrastando.

# Obtenemos el cociente de verosimilitudes
CocVer= -2 *(summary(modgls)$logLik - summary(modvarpotencia)$logLik)
CocVer
# Vemos que da lo mismo que en:
anova(modgls, modvarpotencia)
# Podemos pedir el L.Ratio como:
anova(modgls, modvarpotencia)$L.Ratio

# Luego el valor de p utilizando la distribución de Chi^2 como
pchisq(CocVer, df=1, lower.tail=FALSE)
# comparar con
anova(modgls, modvarpotencia)$"p-value"
# Más adelante veremos con más detalle de que se trata la distribución Chi^2

# Esta comparación también nos sirve como un análisis inferencial
# para evaluar el supuesto de homogeneidad de varianzas.
# En este caso concluimos que hay heterogeneidad de varianzas
# y que es necesario modelarlas.
```

```
summary(modvarpotencia)
# vemos que la potencia es "negativa" (Gamma estimado = -0.157),
# es decir que a mayor pbi menor varianza.
# Similar a lo que observamos en el gráfico.
```

14.6 Residuos de Pearson

```
# e[i] = ( y[i] - E(y[i]) ) / sqrt(Var(y[i]))
# la "E" es por "valor esperado"
# Recordando estadística 1... ¿Qué es el valor esperado?

residuos_p=resid(modvarpotencia, type="pearson")
# le pedimos los residuos estandarizados
# según pearson que tienen en cuenta la varianza correspondiente para cada residual

# Los siguientes son los residuos sin estandarizar con los que veníamos trabajando
previamente.
# Eran útiles porque suponíamos homogeneidad de varianzas. En cambio ahora
# estamos modelando la heterogeneidad de varianzas
residuos_c=resid(modvarpotencia, type="response")

# los comparamos
View(data.frame(residuos_p,residuos_c))

# Veamos cómo pasamos de los residuos ordinarios
# a los resiudos estandarizados según el modelo propuesto
# Primero necesitamos el "Gamma", lamentablemente lo guardaron como
# atributo dentro del objeto del modelo y entonces el código para obtenerlo
# es algo largo, ver:
Gamma = attr(modvarpotencia$apVar, which="Pars" )[1]
Gamma
# Vemos que es el mismo valor que nos daba el summary(modvarpotencia)
# Si resulta muy complejo entender el código de arriba simplemente
# copien y peguen el valor del Gamma del summary.
# Seguimos ahora obteniendo el desvío estándar correspondiente
# a cada unidad experimental según su valor de pbi con nuestro modelo:
DS = sqrt( (modvarpotencia$sigma^2) * (datos$pbi ^ (2*Gamma)) )
DS
# Ahora dividimos los residuos comunes por su desvío estándar
res_pearson=residuos_c/DS
# Comparamos los residuos de Pearson que obtuvimos "paso a paso" con
# aquellos que obtuvimos con la función "resid"
plot(residuos_p,res_pearson)
abline(a=0, b=1, col="red", lw=3)
# ¡Genial!


# Vemos algunos supuestos
predichos=fitted(modvarpotencia)
```

```
plot(predichos,residuos_p)
abline(a=0,b=0, col="violet", lw=2)
### mmmm.... la distribución parece asimétrica. ¿Por qué?

par(mfrow=c(2,1))
hist(residuos_p, col="yellow")
boxplot(residuos_p, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
datos
par(mfrow=c(2,2))
hist(residuos_p, col="yellow")

normal=rnorm(mean=0, sd=summary(modvarpotencia)$sigma, n=length(residuos_p))
hist(normal, col="green")

boxplot(residuos_p, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos
estandarizados")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")

par(mfrow=c(1,1))
qqnorm(residuos_p)
qqline(residuos_p)
# "Cara triste" indica asimetría negativa

library(moments)
skewness(residuos_p) # debería dar cero
# Asimetría negativa
kurtosis(residuos_p) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos_p,"pnorm",mean(residuos3),sd(residuos3))

# alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) #primero hay que instalar el paquete
lillie.test(residuos_p)

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos_p)
# ¡¡Auch!!!
```

14.7 Modelo constante más potencia de la variable independiente

```
# Var[i] = var * (Gamma1 +  |PBI[i]| ^ Gamma2   )^2
```

```
modvarconst= gls(vida~log10(pbi), weights=varConstPower(form=~pbi), data=datos)
anova(modgls, modvarpotencia,modvarconst)
```

```
Model             df      AIC      BIC    logLik   Test  L.Ratio p-value
modgls             1  3 1144.698 1154.192 -569.3490
modvarpotencia     2  4 1137.682 1150.341 -564.8412 1 vs 2 9.015644  0.0027
modvarconst        3  5 1139.682 1155.506 -564.8412 2 vs 3 0.000000  0.9997
```

```
# Hasta ahora el mejor modelo es el que tiene en cuenta la potencia del pbi para
# modelar la varianza (modvarpotencia).
# Sin embargo aún no encontramos un modelo "adecuado" dado que no se cumplen los
supuestos
# de este análisis.

# Este ejemplo sirve para comprender que significa el supuesto de homogeneidad
# de varianzas y cómo se puede modelar la heterogeneidad de varianzas en caso que
# este supuesto no se cumpla. Recordemos que además en caso de existir
heterogeneidad
# de varianzas este es un aspecto en si mismo interesante, tanto como modelar las
tendencias
# promedio. Este ejemplo también sirve para recordar el concepto de asimetría y
# que nosotros por ahora estamos modelando variables asumiendo que tienen una
distribución
# normal, la cual es simétrica.

# En otros scripts veremos ejemplos con modelos adecuados y heterogeneidad de
varianzas,
# que al combinarse con la experiencia lograda con este ejemplo, lograrán una
comprensión
# abarcativa del tema.
```

Trabajo Práctico Nº 15: La propensión marginal a consumir decreciente según el keynesianismo

15.1 Ejercicio 1

```
# Generalmente en los modelos de análisis macroeconómico, herencia de J.M Keynes,
# se supone que el consumo del individuo promedio de un país depende positivamente
del
# ingreso que dispone, pero a una tasa menos que proporcional. Es decir, que
# incrementos de una unidad de ingreso generan aumentos del consumo
# mayores a 0 pero menores a 1. Un economista se propone evaluar dicho modelo
# conceptual con datos de 34 países de la OECD ("Organization for Economic
# Co-operation and Developtment"). A su vez, reconociendo las diferencias entre
# economías con distinto nivel de desarrollo, plantea la posibilidad de que dicha
# propensión marginal a consumir (el incremento en el consumo
# por unidad de incremento del ingreso) sea diferente entre ellas.
```

```
# Los datos pertenecen a la OECD y al año 2006.
(http://stats.oecd.org/Index.aspx?DataSetCode=MON20123_2)
# consumo = Es el consumo per capita valuado en USD a precios constantes.
# yd = Es el ingreso per capita también valuado en USD a precios constantes.
# desarr = Es el nivel de desarrollo del país.

datos = read.table("datos_p_15.txt", header = T, dec = ",")
```

15.1.1 Consignas a resolver

```
# 1) Indique unidad experimental, muestra y población. ¿De qué tipo son las variables
# de interés? Presente una medida de la correlación lineal entre el consumo y
# el ingreso per capita. Indique si la siguiente afirmación es verdadera o falsa
# (justifique): "El coeficiente de correlación de Pearson brinda una medida adecuada
# de la asociación lineal así como no lineal entre dos variables cuantitativas siempre
# y cuando ambas se distribuyan de acuerdo a una misma distribución de probabilidad."

# 2) Plantee (y además estime) un modelo estadístico relevante de acuerdo a los
# objetivos del economista. Verifique los supuestos del modelo.

# 3) Escriba en papel la recta estimada para los países de desarrollo bajo. ¿Que
# interpretación tiene el intercepto en términos del problema? ¿Y el coeficiente de
# regresión parcial? Indique las unidades de ambos coeficientes.

# 4) El economista sospecha que podría mejorar la bondad de ajuste de su modelo si
# considerara varianzas diferentes entre economías de dispar desarrollo. Indague
# mediante un gráfico si las sospechas del economista son fundadas. Estime un modelo
# que contemple una varianza distinta para cada nivel de la variable categórica.

# 5) ¿Cuál de los dos modelos es preferible? Plantee las hipótesis adecuadas y
# concluya. ¿Cuál nivel de desarrollo posee mayor varianza? ¿Cuáles son las ventajas
# (o desventajas) del modelo elegido  sobre el coeficiente de correlación de Pearson
# planteado en el punto 1?
```

15.2 Ejercicio 2

```
# Datos ejercicio 10 página 297 del libro de Webster

# Una empresa carbonífera en West Virginia analizó la producción promedio de tres
minas.
# Cuatro grupos de empleados trabajaron en cada mina y se registró en toneladas la
# produccción de carbón resultante por día. Se utilizó un modelo con dos factores
# considerando a cada "grupo" como un bloque. Como nuevo supervisor administrativo,
# usted debe determinar si existen diferencias en la productividad promedio de las
minas.
carbon=c(42.7,47.1,32.1,29.2,
         54.1, 59.2, 53.1, 41.1,
         56.9, 59.2, 58.7, 49.2)

mina=c(1,1,1,1,2,2,2,2,3,3,3,3)
```

```
mina=as.factor(mina)

grupo=c(1,2,3,4,1,2,3,4,1,2,3,4)
grupo=as.factor(grupo)

datos=data.frame(carbon,grupo,mina)
```

15.2.1 Modelo con varianzas homogéneas entre minas

```
# 1) escribir el modelo y estimar sus parámetros
modelo=with(datos, lm(carbon ~ mina + grupo))
modelo
summary(modelo)

# 2) evaluar las hipótesis de interés
anova(modelo)

# install.packages("agricolae")
library(agricolae)

HSD.test(modelo,"mina", console=TRUE)

# El análisis continúa con la evaluación de los supuestos

# Miremos entonces la heterogeneidad de varianzas primero...
datos$residuos=resid(modelo)
plot(datos$mina,datos$residuos)
tapply(datos$residuos,datos$mina,sd)

# otro gráfico interesante (sin acentos en las secuencias)
medias=with(datos,tapply(carbon,mina,mean))
medias
desvios=with(datos,tapply(residuos,mina,sd))
desvios
# ojo que estamos graficando el desvío estándar y no el error estándar.
fig= barplot(medias,ylim=c(0,63),ylab= "carbón (toneladas por día)")
arrows(fig,medias+desvios,fig,medias-desvios, angle=90,code=3)

# agregamos al gráfico los resultados del test de tukey
text(0.7,44,"a")
text(1.9,57,"b")
text(3.1,62,"b")
```

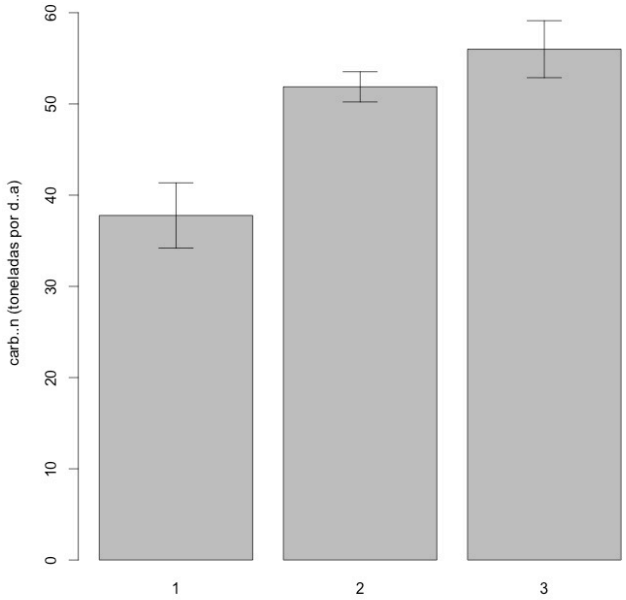


Figura 15.1: Media de producción de carbon en toneladas por día según las minas 1,2 y 3 con el correspondiente desvío estándar.

Aparentemente la mina 2 tiene menor varianza.
Vamos a evaluarlo utilizando un test inferencial:

15.2.2 Modelo varianzas distintas para cada mina

```
library(nlme)
modgls=gls(carbon ~ mina + grupo, data=datos)

# ahora proponemos que las distintas minas tienen también distintas varianzas.
modident=gls(carbon~mina + grupo, varIdent(form= ~ 1 | mina),data=datos)

summary(modident)
```

```
Generalized least squares fit by REML
  Model: carbon ~ mina + grupo
  Data: datos
       AIC      BIC    logLik
  54.48482 52.61066 -18.24241

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | mina
 Parameter estimates:
           1           2           3
1.000000e+00 4.667644e-05 7.592935e-01

Coefficients:
              Value Std.Error   t-value p-value
(Intercept)  40.000 2.3047006     17.36   0e+00
mina2        14.100 2.3047006      6.12   9e-04
```

```
mina3        18.225 2.8937776      6.30   7e-04
grupo2        5.100 0.0003043  16761.50   0e+00
grupo3       -1.000 0.0003043  -3286.57   0e+00
grupo4      -13.000 0.0003043 -42725.40   0e+00

 Correlation:
       (Intr) mina2  mina3  grupo2 grupo3
mina2  -1.000
mina3  -0.796  0.796
grupo2  0.000  0.000  0.000
grupo3  0.000  0.000  0.000  0.500
grupo4  0.000  0.000  0.000  0.500  0.500

Standardized residuals:
          Min            Q1           Med            Q3           Max
-1.496941e+00 -7.186400e-02  2.244322e-09  4.447432e-01  8.623680e-01

Residual standard error: 4.609401
Degrees of freedom: 12 total; 6 residual
```

anova(modgls,modident)

```
Model       df      AIC      BIC    logLik   Test  L.Ratio p-value
modgls       1  7 52.31623 50.85854 -19.15811
modident     2  9 54.48482 52.61066 -18.24241 1 vs 2 1.831402  0.4002
```

Cuando evaluamos si los residuos cuentan con
distribución normal lo hacemos a partir de métodos
gráficos y métodos inferenciales.
En los primeros scripts del curso evaluamos el supuesto
de homogeneidad de varianzas sólamente a partir
de métodos gráficos. La secuencia aquí descripta
es una manera adecuada de evaluar este supuesto
utilizando métodos inferenciales.

Interesante saber el tipo de funciones disponibles para modelar la varianza
?varClasses
Recuerden lo que dice al final respecto que pueden incoporar cualquier función que
ustedes deseen simplemente con un poco más de notación.

Volver al Índice principal

Modelos no lineales generales

Trabajo Práctico Nº 16: Primeros casos prácticos dentro del modelo no lineal general

16.1 Problema y Datos

```
# datos obtenidos de www.gapminder.org
datos=read.table("datos_p_9.txt")
colnames(datos)=c("pais","vida","pbi")

# Nos movemos del modelo lineal al modelo no lineal general

# Volvemos al ejemplo en el que modelamos la esperanza de vida en función del PBI.
# Previamente habíamos detectado que esta relación no era lineal. Países con mayor pbi
# tenían mayor esperanza de vida a una tasa decreciente.

# En una primera etapa usamos modelos "lineales en los parámetros" para estimar esta
relación
# no lineal y comparamos la bondad de ajuste de distintos modelos.

# El modelo cuyo componente determinista fue log10(pbi) tuvo mejor bondad de ajuste
pero
# no era un modelo adecuado, utilizamos distintos modelos de heterogeneidad de
varianzas,
# que era uno de los problemas que habíamos encontrado.

# Ahora vamos a comparar el modelo log10(pbi) con un modelo "no lineal en los
```

```
parámetros", cuyos
# parámetros tengan un significado económico importante, es decir que respondan aun
modelo
# económico interesante. Ya vimos que los modelos polinómicos son lineales en los
parámetros
# y permiten ajustar relaciones no lineales, y que a medida que aumentamos el grado del
polinomio
# el modelo estadístico ajusta mejor a los datos. Sin embargo estos parámetros son
difíciles
# (a veces imposible) de interpretar en términos económicos y por lo tanto el modelo es
cuestionable.
```

16.2 Modelo lineal

```
modelo3=lm(vida~log10(pbi),data=datos)
summary(modelo3)

# observados vs. variable independiente
plot(datos$pbi,datos$vida)
# Agregamos la línea que une los valores predichos
curve(modelo3$coefficients[1]+modelo3$coefficients[2]*log10(x),
    add=T, col="green", lw=3)


# residuos vs. predichos
residuos3=resid(modelo3)
predichos3=fitted(modelo3)
plot(predichos3,residuos3)
abline(a=0,b=0, col="violet", lw=3)


# observados vs. predichos
plot(predichos3,datos$vida)
abline(a=0,b=1, col="blue", lw=3)
```

16.3 Modelo no lineal

```
# usamos la función "nls"
?nls

# primero ajustamos el modelo lineal que previamente ajustáramos por lm.
lineal_nls=nls(vida~ a + b * log10(pbi), data=datos)
# nos avisa que conviene dar valores iniciales pero no es obligatorio.
summary(lineal_nls)
```

```
Formula: vida ~ a + b * log10(pbi)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a   11.613      3.160   3.675 0.000316 ***
b   15.075      0.827  18.229  < 2e-16 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.1 on 175 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.327e-08
```

summary(modelo3)

```
Call:
lm(formula = vida ~ log10(pbi), data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-24.074  -2.134   1.058   3.738  11.411

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   11.613      3.160   3.675 0.000316 ***
log10(pbi)    15.075      0.827  18.229  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.1 on 175 degrees of freedom
Multiple R-squared:  0.655,       Adjusted R-squared:  0.6531
F-statistic: 332.3 on 1 and 175 DF,  p-value: < 2.2e-16
```

Naturalmente ambas funciones dan los mismos resultados ya que usan el método de
mínimos cuadrados para estimar parámetros. La diferencia es que nls emplea
optimización numérica mientras que lm emplea optimización analítica. Como nls
emplea optimización numérica aparecen términos asociados a este tipo de estimación
en el summary como "Number of iterations to convergence" y "Achieved convergence
tolerance" que no estaban en el summary de objetos resultantes de aplicar la función
"lm". En el caso de nls, utiliza el algoritmo de Gauss-Newton. El método de mínimos
cuadrados podemos utilizarlo para los modelos no lineales porque suponemos en
nuestros modelos que los residuos son aditivos. Es decir: y = f(x) + e

# Un modelo no lineal en los parámetros interesante
# conocido como el modelo de Michaelis-Menten
mod_nls1=nls(vida ~ (a * pbi) / (b + pbi), data=datos)
summary(mod_nls1)

```
Formula: vida ~ (a * pbi)/(b + pbi)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  76.4236     0.7362  103.81   <2e-16 ***
b 383.4443    31.2166   12.28   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.608 on 175 degrees of freedom
```

```
Number of iterations to convergence: 9
Achieved convergence tolerance: 3.37e-06
# probamos dando valores iniciales y obtenemos los mismos resultados
# (aunque naturalmente si los valores iniciales que indicamos son cercanos
# a los que minimizan la SCE el algoritmo convergerá en menos iteraciones)
mod_nls1b=nls(vida ~ (a * pbi) / (b + pbi), data=datos, start=list(a=76, b=383))
summary(mod_nls1b)
# Comparar "Number of iterations to convergence" de este modelo con respecto al
anterior.


# En papel escribir el modelo estadístico y discutir con el compañero.

# observados vs. variable independiente
plot(datos$pbi,datos$vida)
curve((76.4236 * x) / (383.4443 + x), add=T, col="red", lwd=3)

# a = asíntota de esperanza de vida
# Probemos reemplazar por un valor de pbi muy grande
(76.4236 * 85000000) / (383.4443 + 85000000)
# Interpretar ese valor y dar sus unidades

# b = pbi al que ocurre a / 2
# Comprobemos
(76.4236 * 383.4443) / (383.4443 + 383.4443)
# es igual a
76.4236 / 2

# Ambos parámetros de la función de Michaelis-Mentel son interesantes y directamente
# interpretables en términos económicos.

# También podemos darles los nombres que querramos a los parámetros
mod_nls1b=nls(vida ~ (E_max * pbi) / (pbi_mitad + pbi), data=datos,
        start=list(E_max=76, pbi_mitad=383))
summary(mod_nls1b)


# Intervalos de confianza para a y b
confint(mod_nls1, level=0.95)
# Interpretar

# ¿Cuál es el valor esperado para la esperanza de vida de los paises con 6500 de pbi?
newdata=data.frame(pbi=6500)
predict(mod_nls1,newdata)
# Interpretar e indicar unidades.
```

16.4 Prueba F para comparar modelos no lineales anidados

```
# La función anova sólo puede utilizarse para modelos anidados.
# por lo que NO podemos hacer:
```

```
anova(mod_nls1,lineal_nls)
# ya que ninguno de estos modelos esta anidado dentro de otro.
# ¿Qué significa que dos modelos estén anidados?
# En cambio sí podemos hacer:
mod_sin_b=nls(vida ~ (a * pbi) / (0 + pbi), data=datos)
anova(mod_nls1,mod_sin_b)

# El estadístico F se obtiene de la siguiente manera:
((18876.7-7641.4)/7641.4) / ((176-175)/175)
round(((18876.7-7641.4)/7641.4) / ((176-175)/175), 2)
# Al eliminar el parámetro "b" aumentamos la variación no explicada en un
# 147 % = ((18876.7-7641.4)/7641.4)*100,
# mientras que sólo agregamos un parámetro al modelo, perdiendo así
# sólamente un 0,57% = ((176-175)/175)*100 de grados de libertad residduales.
# De este modo el parámetro "b" contribuye importantemente.
# La prueba F obtenida de este modo para modelos anidados es interesante
# porque tiene en cuenta el principio de parsimonia al dividir
# los cambios en la variabilidad no explicada por los cambios en el número
# de grados del libertad residuales (inversamente proporcionales al
# número de parámetros, es decir complejidad del modelo).

# Plantear la H0) y la H1)
# ¿Qué significa el valor p?
# ¿Cuál es el valor esperado de F bajo H0?

# Probemos con otro modelo
# A este se lo llama modelo de "Hill"
# El modelo de Michaelis-Menten es un caso particualr del de Hill
# cuando n=1
mod_nls_hill=nls(vida ~ (a * pbi^n) / (b + pbi^n), data=datos,
          start=list(a=76, b=383, n=1))
summary(mod_nls_hill)

Formula: vida ~ (a * pbi^n)/(b + pbi^n)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a 91.76954    7.13031  12.870  < 2e-16 ***
b  9.24841    4.21919   2.192   0.0297 *
n  0.38701    0.09055   4.274 3.15e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.035 on 174 degrees of freedom

Number of iterations to convergence: 19
Achieved convergence tolerance: 5.739e-08

confint(mod_nls_hill)
# El n es bien distinto de 1.
# Comparemos con el modelo que no tiene el parámetro "n"
anova(mod_nls1,mod_nls_hill)
```

```
# ¡Copado!

# Veamos
par(mfrow=c(2,1))
plot(datos$pbi,datos$vida, main="Michaelis-Menten")
curve((76.4236 * x) / (383.4443 + x),
      add=T, col="red", lwd=3)
plot(datos$pbi,datos$vida, main="Hill")
curve((coef(mod_nls_hill)[1] * x^coef(mod_nls_hill)[3]) /
        (coef(mod_nls_hill)[2] + x^coef(mod_nls_hill)[3]),
      add=T, col="blue", lwd=3)
```

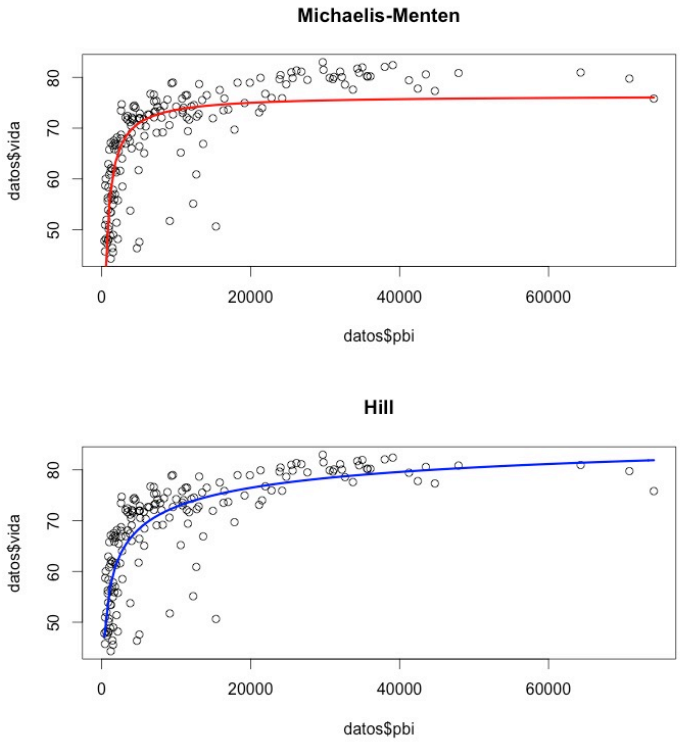


Figura 16.1: Esperanza de vida en años en función de PBI per cápita en miles de dólares según modelos Michaelis-Menten y Hill.

```
# ¿Que hace el parámetro n?
par(mfrow=c(3,1))
curve((coef(mod_nls_hill)[1] * x^coef(mod_nls_hill)[3]) /
        (coef(mod_nls_hill)[2] + x^coef(mod_nls_hill)[3]),
      add=F, col="blue", lwd=3, main ="n = 0.39")
curve((coef(mod_nls_hill)[1] * x^1) /
        (coef(mod_nls_hill)[2] + x^1),
      add=F, col="blue", lwd=3, main = "n = 1")
curve((coef(mod_nls_hill)[1] * x^2) /
        (coef(mod_nls_hill)[2] + x^2),
      add=F, col="blue", lwd=3, main = "n = 2")
```

16.5 Bondad de ajuste en modelos no lineales

```
# A diferencia de lm, el summary de un objeto resultante de la función nls NO provee
```

```
r2.
# Recordemos que r2 = (SCTOT - SCE) / SCTOT,
# siendo SCTOT = Sum(y[i] - ymedia)^2.
# Por ejemplo, si tenemos un modelo lineal como:
# ymedia = B0 + B1 * xi
# y suponemos que B1 = 0, nos queda:
# ymedia = B0
# Entonces la SCTOT refleja la dispersión de los datos con respecto a lo
# esperado bajo un modelo simple que sólo contempla ymedia(=B0) y que está
# ANIDADO dentro del modelo que emplea la SCE. Este último refleja la dispersión
# de los datos con respecto a lo esperado bajo un modelo más complejo, supongamos
# ymedia dado xi = B0 + B1 * xi.

# En cambio, para el modelo de Michaelis-Menten que venimos discutiendo
# ymedia = (a * pbi) / (b + pbi)
# si suponemos que los parámetros a  y (o) b son iguales a cero
# NO llegamos al modelo simple de ymedia constante que usa la SCTOT.
# Por lo tanto este modelo no se encuentra anidado y no tiene sentido utilizar el r2
# como índice de bondad de ajuste.
# Esto no siempre pasa para todos los modelos no lineales,
# y depende de cada modelo particular empleado.

# En general, para modelos no lineales, podemos usar como criterio de bondad de
ajuste:
# .- Desvío estándar residual
# .- AIC, BIC, etc.
# .- Gráfico de observados vs. predichos (y eventualmente un r2 de esta relación)
```

16.6 Comparación modelos lineales y no lineales

```
AIC(modelo3,lineal_nls, mod_nls1, mod_nls_hill)
```

```
              df      AIC
modelo3       3 1146.423
lineal_nls    3 1146.423
mod_nls1      3 1174.741
mod_nls_hill  4 1143.636
```

```
BIC(modelo3,lineal_nls, mod_nls1, mod_nls_hill)
```

```
              df      BIC
modelo3       3 1155.951
lineal_nls    3 1155.951
mod_nls1      3 1184.270
mod_nls_hill  4 1156.341
```

```
# Interpretar

# Observados vs. predichos
plot(fitted(mod_nls_hill),datos$vida)
abline(a=0, b=1, lw=3, col="red")
cor(fitted(mod_nls_hill),datos$vida)^2
```

```
# Interpretar


# Evalúen los supuestos del mejor modelo.... :D
# y concluyan
```

16.7 Modelo no lineal con heterogeneidad de varianzas

```
install.packages("nlreg")
library(nlreg)
?nlreg
# Para jugar en sus casas....
# antes que alguno pregunte.... modelos no lineales con heterogeneidad de varianzas NO
# entra en el examen...

## A esta altura del curso tienen que haberse dado cuenta que en R podemos ajustar
# cualquier modelo estadístico. Sólo nos limita nuestra imaginación para plantear
# modelos y tener buenas ideas.....
```

Trabajo Práctico Nº 17 y 18: Diferentes modelos de crecimiento demográfico

17.1 Problema y Datos

```
# datos obtenidos de www.gapminder.org

datos=read.table("datos_p_17.txt")
colnames(datos)=c("habitantes","anio")
str(datos)
# Datos del número de habitantes en el mundo para cada año

plot(datos$anio,datos$habitantes)
# ¿Son datos en panel? ¿En corte transversal? ¿Serie de tiempo?
```

17.2 Modelo exponencial de crecimiento demográfico

```
modelo=nls(habitantes~a*exp(b*anio), start=list(a=min(datos$habitantes), b=0.002),
        nls.control(maxiter=2000), data=datos, trace=t)
# cuando ponemos trace=t nos indica la secuencia de iteraciones hasta encontrar los
valores
# estimados de los parámetros que hagan mínima la suma de los residuos elevados al
cuadrado
# (la primera columna que disminuye en su valor a través de las iteraciones).

summary(modelo)

Formula: habitantes ~ a * exp(b * anio)
```

```
Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 1.970e-06  4.836e-07   4.075 0.000168 ***
b 1.785e-02  1.238e-04 144.167  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 51580000 on 49 degrees of freedom

Number of iterations to convergence: 1084
Achieved convergence tolerance: 1.946e-07
```

```
# a = representa el valor de la población cuando nació Cristo
# ¿Qué pueden decir acerca de este valor?

# b representa la tasa de crecimiento de la población
# Interpretar e indicar sus unidades.

# Gráfico interesante
plot(datos$anio,datos$habitantes)
curve(1.970e-06*exp(0.01785*x), add=T, col="red", lwd=2)

# Interpretar
confint(modelo)

# ¿Cuántos habitantes seremos en la tierra según este modelo en 2050?
newdata=data.frame(anio=2050)
predict(modelo,newdata)
# auchh.... 15 mil millones de personas! ¡Hoy somos 7 mil millones!
# Si consideramos que la tasa de consumo per cápita aumenta más rápido que el
crecimiento
# poblacional. ¿Seremos capaces de satisfacer esa demanda en este planeta?
# ¿O conquistamos marte? Justifique su respuesta... :D
```

17.3 Modelo logístico de crecimiento demográfico

```
# agregemos ahora a la base anterior los datos de población hasta el 2011 y las
# proyecciones de las naciones unidas hasta el año 2050
datos2=read.table("datos_p_18.txt")
colnames(datos2)=c("habitantes","anio")
str(datos2)
plot(datos2$anio,datos2$habitantes)


modelo2=nls(habitantes~Asym/(1+exp((xmid-anio)/scal)),
        start=list(Asym=max(datos2$habitantes),xmid=1990,scal=1),
        nls.control(maxiter=2000), data=datos2, trace=t)

summary(modelo2)
```

```
Formula: habitantes ~ Asym/(1 + exp((xmid - anio)/scal))
```

```
Parameters:
      Estimate Std. Error t value Pr(>|t|)
Asym 1.079e+10  4.475e+07   241.1   <2e-16 ***
xmid 1.991e+03  3.230e-01  6164.6   <2e-16 ***
scal 3.280e+01  2.192e-01   149.7   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43980000 on 98 degrees of freedom

Number of iterations to convergence: 8
Achieved convergence tolerance: 3.564e-06
```

```
# Interpretar el valor t asociado a Asym en términos del problema.

# Gráfico interesante
plot(datos2$anio,datos2$habitantes)
curve(1.079e+10/(1+exp((1991-x)/3.280e+01)), add=T, col="blue", lwd=3)

# Asym = valor máximo de habitantes (asíntota)

# xmid = año en el cual se produce el punto de inflección.
# Es el año al cual se produce la mitad de la población máxima.
# Si Asym es 1.079e+10, entonces
1.079e+10/2
# es igual a:
1.079e+10/(1+exp((1991-1991)/3.280e+01))

# scal = parámetro de escala. Si la escala es mayor que cero el
# término exp((xmid-anio)/scal se reduce cuando aumenta x, por lo tanto
# el resultado de lafunción se acerca al valor máximo de habitantes (Asym).
# Ver:
exp((1991-1950:2050)/3.280e+01)
# en cambio:
exp((1991-1950:2050)/-3.280e+01)
# Para valores positivos, si la escala disminuye, la tasa de crecimiento aumenta.
# Veamos:
plot(datos2$anio,datos2$habitantes)
curve(1.079e+10/(1+exp((1991-x)/2)), add=T, col="red", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/20)), add=T, col="red", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/3.280e+01)), add=T, col="blue", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/40)), add=T, col="orange", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/400)), add=T, col="orange", lwd=3)
# Sucede lo opuesto para valores negativos
plot(datos2$anio,datos2$habitantes)
curve(1.079e+10/(1+exp((1991-x)/-2)), add=T, col="red", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/-20)), add=T, col="red", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/-3.280e+01)), add=T, col="blue", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/-40)), add=T, col="orange", lwd=3)
curve(1.079e+10/(1+exp((1991-x)/-400)), add=T, col="orange", lwd=3)
```

```
# ¿Cuántos habitantes seremos en la tierra según este modelo en 2050?
newdata=data.frame(anio=2050)
predict(modelo2,newdata)

[1] 9252938747
# Este nuevo modelo predice que para el 2050 seremos 9 mil millones de habitantes en
# vez de los 15 mil millones planteados por el modelo exponencial.
# Aproximadamente 9 mil millones es el número aceptado actualmente para esta
predicción.
# ¿Por qué los dos modelos discutidos dan predicciones distintas?
# ¿Qué es el dominio de un modelo? ¿Cuál es la importancia de este concepto en esta
discusión?

# ¿Cuál es el tamaño máximo de habitantes que tendrá la tierra según este modelo?

# si uno no sabe cuáles valores iniciales podrían ser lógicos para los parámetros
# se pueden usar algunas funciones predeterminadas en R. Por ejemplo para el caso que
# acabamos de discutir sería:
modelo3=nls(habitantes~SSlogis(anio,Asym,xmid,scal),data=datos2)
# SS por "Self-Starting model"
summary(modelo3)

Formula: habitantes ~ SSlogis(anio, Asym, xmid, scal)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Asym 1.079e+10  4.475e+07   241.1   <2e-16 ***
xmid 1.991e+03  3.230e-01  6164.6   <2e-16 ***
scal 3.280e+01  2.192e-01   149.7   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43980000 on 98 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 6.399e-07

# comparar con el modelo anterior
summary(modelo2)

# Otros modelos no lineales de tipo "Self-starting" habitualmente utilizados son:
?SSasymp
?SSasympOff
?SSasympOrig
?SSbiexp
?SSfol
?SSfpl
?SSgompertz
?SSlogis
?SSmicmen
?SSweibull
```

17.4 Supuestos

```
# Evaluemos los supuesto del primer modelo, el exponencial
# residuos vs. predichos
residuos=resid(modelo)
predichos=fitted(modelo)
plot(predichos,residuos)
abline(a=0,b=0, col="violet", lw=2)
```

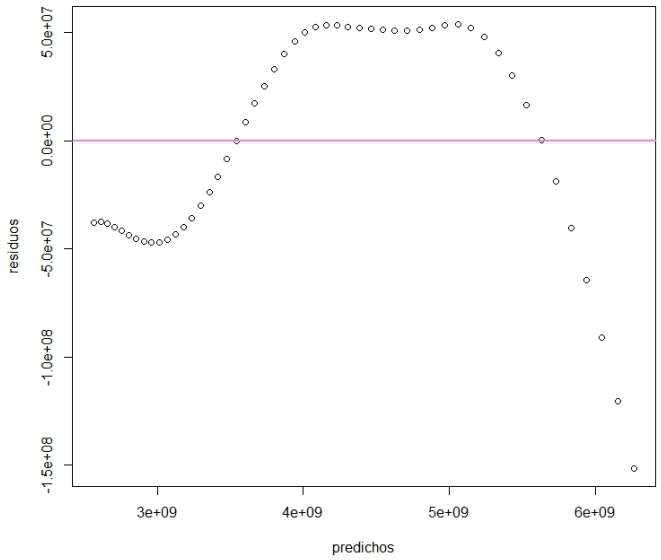


Figura 17/18.1: Residuos en función de predichos para el modelo exponencial

```
# Interpretar.

# residuos vs. variable independiente
plot(datos$anio,residuos, xlab="año")
abline(a=0,b=0, col="orange", lw=3)


# observados vs. predichos
plot(predichos,datos$habitantes)
abline(a=0,b=1, col="blue", lw=2)


par(mfrow=c(2,1))
hist(residuos, col="yellow")
boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
# el punto más cercano (pero sin superar)
# 1.5 * rango intercuartil es el bigote superior o inferior

# comparamos con una distribución normal de igual desvío estándar residual a nuestros
datos
par(mfrow=c(2,2))
hist(residuos, col="yellow")
```

```
normal=rnorm(mean=0, sd=summary(modelo)$sigma,
n=length(summary(modelo)$residuals))
hist(normal, col="green")

boxplot(residuos, bty="l", range=1.5, col="yellow", horizontal=T, xlab="residuos")
boxplot(normal, bty="l", range=1.5, col="green", horizontal=T, xlab="normal")
```

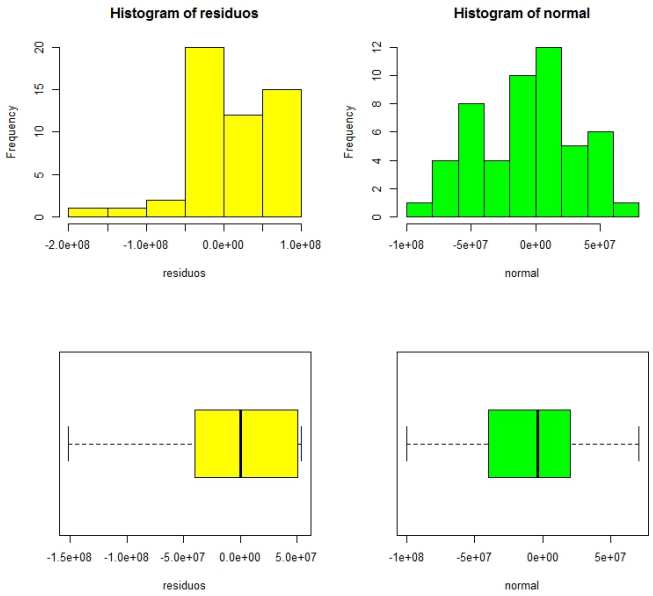


Figura 17/18.2: Distribución de los residuos para la muestra en estudio y una distribución normal de igual desvío estándar residual que la muestra.

```
par(mfrow=c(1,1))
qqnorm(residuos)
qqline(residuos)

library(moments)
skewness(residuos) # debería dar cero
kurtosis(residuos) # debería dar tres
# probemos con la normal
skewness(normal)
kurtosis(normal)
# están estimados como el tercer y cuarto momento estandarizados, respectivamente.

# Test de Kolmogorov-Smirnov para evaluar normalidad
ks.test(residuos,"pnorm",mean(residuos),sd(residuos))

# alternativamente conviene usar la modificación de Lillefors a este test
library(nortest) #primero hay que instalar el paquete
lillie.test(residuos)

# Test de Shapiro-Wilk para evaluar normalidad
shapiro.test(residuos)

# Esta evaluación de los supuestos muestra claramente falta de independencia,
# en particular, hay un patrón de autocorrelación temporal en los residuos. Modelos
```

```
# de autocorrelación temporal serán discutidos en econometría. El mismo problema
# seguramente presentará el modelo de regresión logística, pueden practicarlo en
# sus casas.

# En el ejemplo de la clase anterior los residuos seguían una distribución asimétrica
# (i.e. no normal)
# En cada tema vemos ejemplos en los cuáles se cumplen los supuestos y otros en los
# que no se cumplen para lograr una comprensión adecuada de los modelos. El próximo
# ejemplo será un modelo no lineal en el que se cumplen los supuestos.
```

17.5 Fórmulas en “nls”

```
#Como en “lm”, la parte izquierda de la fórmula especifica la variable de respuesta, y es
#seguida por (~) como separador que se lee habitualmente como “es regresada sobre” o
#“es modelada por”. La parte derecha de la fórmula para los modelos no lineales es muy
#diferente de los modelos lm. En su forma más simple, para los nls, la parte derecha es
#una expresión matemática que consiste de constantes como ser el número 1;
#predictores, en este caso solamente “año”; parámetros nombrados a priori como theta
#1, theta 2 y theta 3; y operadores matemáticos como exp para la exponenciación, / para
#la división y + para la suma. Factores, interacciones y en general la notación
#Wilkinson-Rogers utilizada para modelos lineales no es utilizada para nls. Paréntesis
#son usados con las reglas de precedencia matemática, pero los corchetes “[]” y “{}” no
#pueden ser usados. Si los valores de los parámetros y predictores han sido
#especificados, entonces la parte derecha de la fórmula sería evaluada como un número,
#ver Fox y Weisberg (2011, Sec. 1.15). No podemos nombrar los parámetros con un
#nombre legal de R, como ser theta 1, alpha, t1 o “Asymptote”.

# ver
?nls
```

Trabajo Práctico Nº 19: La cinética Michaelis-Menten y la función “self-start”

19.1 Problema y Datos

```
# Datos obtenidos de "The Demand for New Automobiles in the United States" p. 279,
de Daniel B. Suits, en
# The Review Economics and Statistics, Vol. XL (Agosto 1958).
datos=read.csv("datos_p_19.csv")

# precio = Índice del Precio Real de Automóviles Nuevos
# ingreso = Ingreso Disponible Real (en miles de millones de dólares)
# autos = Automóviles en Circulación al principio de cada año (millones de unidades)
# ventas = Ventas de Automóviles Nuevos (millones de unidades).


# Queremos estudiar cómo varían las ventas de automóviles en función del ingreso
```

```
modelo_lineal=nls(ventas~ a + b *ingreso, data=datos,
                  start=list(a=1, b=1))
summary(modelo_lineal)
```

```
Formula: ventas ~ a + b * ingreso

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a 0.899249   0.483118   1.861 0.083823 .
b 0.015373   0.003142   4.892 0.000238 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6328 on 14 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 3.969e-07
```

```
with(datos,plot(ingreso,ventas))
curve(0.899249 + 0.015373 * x, add=T, col="red", lwd=2)
```

```
modelo_pot=nls(ventas ~ a*ingreso^b, data=datos,
               start=list(a=1, b=1), nls.control(maxiter=2000))
summary(modelo_pot)
```

```
Formula: ventas ~ a * ingreso^b

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  0.10323    0.07602   1.358 0.195995
b  0.68857    0.14467   4.760 0.000305 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6119 on 14 degrees of freedom

Number of iterations to convergence: 8
Achieved convergence tolerance: 3.079e-06
```

```
with(datos,plot(ingreso,ventas))
curve(0.10323*x^0.68857, add=T, col="red", lwd=2)
```

```
# Función de Michaelis Menten
mod_MM = nls(ventas ~ (a * ingreso) / (b + ingreso), data=datos,
             start=list(a=1, b=1), nls.control(maxiter=2000))
summary(mod_MM)
```

```
Formula: ventas ~ (a * ingreso)/(b + ingreso)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
```

```
a    9.866      4.122   2.394   0.0313 *
b  298.287    191.533   1.557   0.1417
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5918 on 14 degrees of freedom

Number of iterations to convergence: 7
Achieved convergence tolerance: 2.745e-06
```

```
# Interpretar el valor p en términos del problema.
with(datos,plot(ingreso,ventas))
curve((9.866*x) / (298.287+x), add=T, col="red", lwd=2)


# Para Michaelis-Menten podemos usar la función "self starting"
?SSmicmen
mod_SS_MM=nls(ventas~SSmicmen(ingreso,a,b),data=datos)
summary(mod_SS_MM)
# obtenemos los mismos resultados que en mod_MM
summary(mod_MM)
AIC(mod_SS_MM,mod_MM)
BIC(mod_SS_MM,mod_MM)


mod_poli = nls(ventas ~ a + b*ingreso+c*ingreso^2 +d*ingreso^3, data=datos,
        start=list(a=1, b=1, c=1,d=1), nls.control(maxiter=2000))
summary(mod_poli)
```

```
Formula: ventas ~ a + b * ingreso + c * ingreso^2 + d * ingreso^3

Parameters:
    Estimate Std. Error t value Pr(>|t|)
a -6.713e+00  4.995e+00  -1.344    0.204
b  1.379e-01  1.103e-01   1.251    0.235
c -5.201e-04  7.765e-04  -0.670    0.516
d  4.985e-07  1.743e-06   0.286    0.780

Residual standard error: 0.4318 on 12 degrees of freedom

Number of iterations to convergence: 2
Achieved convergence tolerance: 7.873e-08
```

```
with(datos,plot(ingreso,ventas))
curve(-6.713e+00+ 1.379e-01*x+-5.201e-04*x^2+4.985e-07*x^3, add=T, col="red",
lwd=2)
```

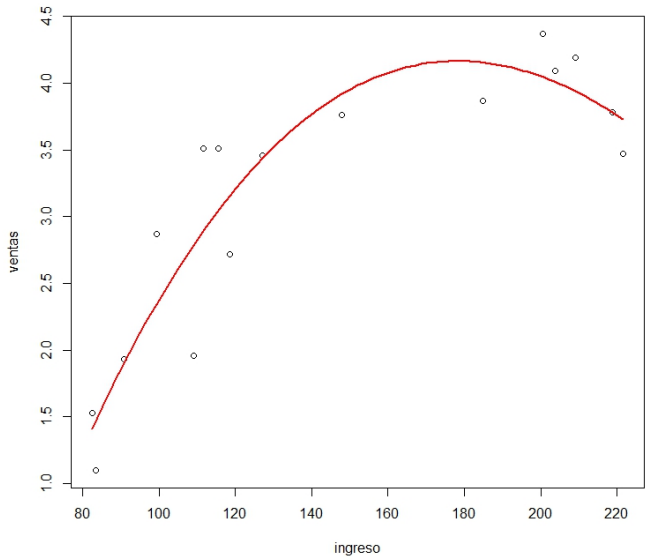


Figura 19.1: Ventas (en millones de unidades) en función de ingreso disponible real (en miles de millones de dólares). Curva del modelo no lineal estimado.

AIC(modelo_lineal,modelo_pot,mod_MM, mod_poli)

```
                df      AIC
modelo_lineal   3 34.62445
modelo_pot      3 33.55006
mod_MM          3 32.48294
mod_poli        5 23.93124
```

BIC(modelo_lineal,modelo_pot,mod_MM, mod_poli)

```
                df      BIC
modelo_lineal   3 36.94222
modelo_pot      3 35.86782
mod_MM          3 34.80070
mod_poli        5 27.79418
```

no se pueden comparar mediante un test de verosimilitud ya que no son modelos anidados

¿Por qué el mod_poli tiene menor AIC pero ninguno de sus parámetros son
significativamente distintos de cero? Relacione dicho aspecto con el
concepto de multi-colinealidad. Estime un polinomio de segundo grado
y compare su bondad de ajuste con el mod_poli. Concluya.

Evaluar los supuestos del modelo con mejor bondad de ajuste.

¿Cuál es la diferencia entre presentar un intervalo de confianza
y presentar el intervalo que va desde la media menos un error estándar hasta la
media más un error estándar?

19.2 Paquetes para modelos no lineales

```
# Si el algoritmo de Gauss-Newton de nls no converge se puede usar

# 1) el paquete nlmrt que utiliza otro algoritmo.
install.packages("nlmrt")
library(nlmrt)

# 2) otra opción:
```

A aquellos que trabajan mucho con regresiones no lineales les va a gustar mucho la función “nls” de R. En la mayoría de los casos funciona muy bien, pero hay algunos inconvenientes que pueden ocurrir cuando se utilizan valores iniciales malos para los parámetros. Uno de los más temidos es el “singular gradient matrix at initial parameter estimates” que frena la función porque el chequeo del gradiente en stats:::nlsModel se terminará si la decomposición QR no tiene la columna completa.
Casi todos los programas existentes para ajustes no lineales utilizan el algoritmo Levenberg-Marquardt para la regresión no lineal. Esto es así porque cambiar entre Gauss-Newton y disminución del gradiente es muy robusto en comparación con valores de inicio lejos de los óptimos. Lamentablemente, la función estándar de nls no tiene lm implementado, y, en cambio, utiliza el tipo Gauss-Newton, las rutinas PORT y un filtro lineal secuencial. El fabuloso paquete minipack.lm de Katherine M. Mullen ofrece un interfaz de R para una implementación de tipo Fortra LM del paquete minipack. La función nls.lm debe ser ofrecida con una función objetiva que devuelve un vector con los residuos a ser minimizados. Junto con Kate yo desarrollé una función nlsLM que tiene el interfaz de la función nls, pero responde con LM en vez de Gauss-Newton. Esto tiene algunas ventajas. La función devuelve el resultado usual de clase “nls”, y, dado algunas modificaciones, todos los genéricos estándar funcionan. Las modificaciones fueron hechas de manera que la fórmula es transformada en una función que devuelve un vector de residuos ponderados cuya suma cuadrada es minimizada por nls.lm. Los parámetros optimizados son luego transferidos a stats:::nlsModel para obtener un objeto de clase “nlsModel”. Las funciones internas de C C_nls_iter y stats:::nls_port_fit fueron eliminadas para evitar una subsiquiente optimización del modelo nlsModel por la via “Gauss-Newton”, “port” o “plinear”.

```
# 3) Otra:
```

nls2 es un paquete de R que agrega el “brute-force” y algoritmnos relacionados como también múltiples valores de inicio a las funciones nls de R.
nls2 es un software libre con licencia GPL y disponible desde CRAN.
Ofrece una función, nls2, que es un “superset” de la función nls de R a la cual responde.

```
install.packages("nls2")
library(nls2)
?nls2
```

Volver al Índice principal

Modelos lineales generalizados

Trabajo Práctico Nº 20: Distribución binomial

20.1 Problema y Datos

```
# Datos obtenidos de Gelman A y Hill J (2007) Data analysis using regression and
multilevel/hierarchical models.
# Ed. Cambridge University Press.

# A partir de este script comenzamos con modelos lineales generalizados:
# expandimos los modelos lineales generales para abarcar otras
# distribuciones además de la normal que pertenencen a la familia de
# distribuciones exponencial (e.g. Poisson, binomial, gamma).

# El marco conceptual de modelo lineal generalizado tuvo sus comienzos
# alrededor de 1970 con el siguiente artículo fundacional:

# Nelder JA, Wedderburn RWM (1972). Generalized Linear Models. Journal
# of the Royal Statistical Society, Series A, 135, 370–384.
```

20.2 Repaso distribución binomial

```
# Unidad experimental = cada pueblo de Argentina
# Muestra = los pueblos evaluados de Argentina
# Población = todos los pueblos de Argentina

# En cada pueblo se encuestan 29 personas y se les pregunta si votarán al
# candidato "liberal" o no (no votarlo incluye voto en blanco)

# Proponemos que
# Yi~Binom(p, N')
# donde
#     Yi  es el número de personas que votan al candidato "liberal" en el pueblo i
#     p   es la probabilidad individual de votar al candidato "liberal"
#     N'  es 29 en este caso, utilizamos N' para diferenciarlo de N el tamaño poblacional

espacio_muestral=seq(from=0, to=29, by=1)
espacio_muestral
# El espacio muestral son los resultados posibles para cada unidad
# experimental luego de cada experimento aleatorio.

# Supongamos que la probabilidad individual de votar al candidato "liberal"
# es 0.9. Entonces la distribución de datos será:
probabilidad = dbinom(espacio_muestral, size=29, prob=0.9)
probabilidad
```

```
[1] 1.000000e-29 2.610000e-27 3.288600e-25 2.663766e-23 1.558303e-21 7.012364e
    -20 2.524451e-18 7.465162e-17
[9] 1.847628e-15 3.880018e-14 6.984033e-13 1.085700e-11 1.465694e-10 1.725010e
    -09 1.774296e-08 1.596866e-07
[17] 1.257532e-06 8.654779e-06 5.192868e-05 2.705757e-04 1.217591e-03 4.696422
     e-03 1.537011e-02 4.210073e-02
[25] 9.472664e-02 1.705080e-01 2.360879e-01 2.360879e-01 1.517708e-01 4.710129
     e-02
```

```
# corroboramos que la suma de las probabilidades sea igual a 1
sum(probabilidad)
# vamos bien, veamos el histograma
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal")

# Dada esta distribución, ¿Cuál es la probabilidad de que en un pueblo
# elegido al azar se observen hasta 2 personas que voten al candidato
# liberal?
pbinom(2,size=29,prob=0.9)
# Si lo hacemos paso por paso:
# (revisen sus carpetas de Est 1 en cuanto a distribución binomial)
P_X_0 = (factorial(29)/(factorial(29-0)*factorial(0))) * (0.9^0) * (0.1^29)
P_X_1 = (factorial(29)/(factorial(29-1)*factorial(1))) * (0.9^1) * (0.1^28)
P_X_2 = (factorial(29)/(factorial(29-2)*factorial(2))) * (0.9^2) * (0.1^27)
paso_paso=P_X_0 + P_X_1 + P_X_2
# comparemos
c(pbinom(2,size=29,prob=0.9), paso_paso)

# ¿Y hasta 26 personas?
pbinom(26,size=29,prob=0.9)

# ¿Exactamente 26 personas?
a = pbinom(25,size=29,prob=0.9)
b = pbinom(26,size=29,prob=0.9)
b-a

# ¿Más de 26 personas?
1-b

# la media de la distribución
# N' * p
29*0.9

# otra manera
sum(probabilidad*espacio_muestral)


# Veamos cómo cambia la distribución para valores crecientes de
# probabilidad individual
par(mfrow=c(3,3))
probabilidad = dbinom(espacio_muestral, size=29, prob=0.01)
```

```
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.01")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.05)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.05")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.10)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.10")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.25)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.25")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.50)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.50")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.75)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.75")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.90)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.90")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.95)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.95")
probabilidad = dbinom(espacio_muestral, size=29, prob=0.99)
barplot(probabilidad, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.99")
```

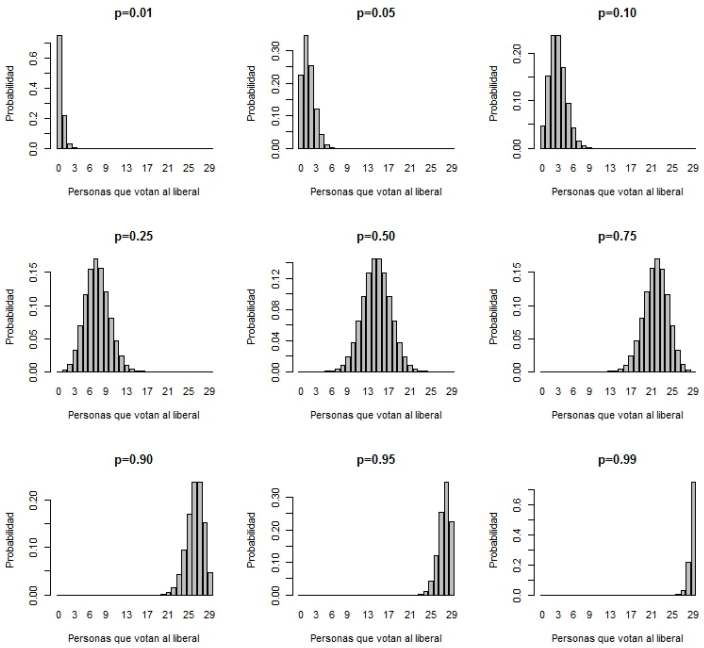


Figura 20.1: Distribución de probabilidad binomial para la cantdad de personas que votan al candidato liberal según varía la probabilidad individual de voto "liberal".

```
# Es evidente que cambios en p implican cambios en la media y en la
# varianza de la distribución.
# Recuerden:
```

```
# media = N' * p
# varianza = N' * p * q

# En sus casas realicen gráficos similares modificando N'

# Antes de continuar discutamos cómo expandir este modelo simple para
# reflejar que "p" varía entre pueblos (unidades experimentales) y que esa
# variación se debe en parte al ingreso promedio que presentan las
# personas en ese pueblo.

# El número de personas que votan al candidato liberal es una variable:
# .- cuantitativa discreta. Recordemos que esta variable surge de
# otra variable observada que es categórica con dos niveles (votar vs.
# no votar al liberal) que representan eventos complementarios: mutuamente
# excluyentes y colectivamente exhaustivos.
# .- que presenta un límite inferior (0) y superior (29)

# por lo que suele (no siempre) presentar:
# -- residuos con distribución distinta a la normal
# -- heterogeneidad de varianzas

# La distribución binomial considera ambos aspectos ya que tiene implícito
# cambios en la varianza a medida que cambia la media:
# varianza = N' * p * q = media * q
# Entonces uno de los supuestos claves de la binomial es que la varianza
# es MENOR que la media ya que 0 < q < 1. Recordemos que q = 1 - p.
```

20.3 Modelo con distribución de error binomial

```
# Datos de Gelman A. & Hill J. 2007 Data analysis using regression
# multilevel/hierarchical models. Cambridge University Press.
datos=read.table("datos_p_20.txt")

# En Estados Unidos de América se postula que las personas con mayores
# ingresos son más propensos a votar al partido conservador. Para evaluar
# esta idea en 1992 se encuestó a 861 personas y se les preguntó si
# votarían a George Bush (voto = 1) o a Bill Clinton (voto = 0)

# Unidad experimental = cada persona que vota a Bush o Clinton en Estados Unidos
# Muestra = 861 personas que votan a Bush o Clinton en Estados Unidos
# Población = todas las personas que votan a Bush o Clinton en Estados Unidos

# Con respecto al ejemplo anterior, N' pasa de 29 a 1.
# Además, ahora    Yi~Binom(pi, N')
# de modo que "p" lo modelamos y por eso lleva el subíndice "i"
modelo= glm (voto ~ ingreso, family=binomial(link="logit"),data=datos)
```

20.4 Estimación y análisis de la devianza

```
# La estimación de los parámetros se realiza por el método de máxima
```

```
# verosimilitud. El algoritmo que usa glm se llama "iterative weighted
# least squares (IWLS)" y es una implementación del "Fisher scoring
# algorithm" para glm

summary(modelo)

Call:
glm(formula = voto ~ ingreso, family = binomial(link = "logit"),
    data = datos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3543  -1.1119  -0.8911   1.2443   1.4938

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.00010    0.20128  -4.969 6.74e-07 ***
ingreso      0.28138    0.06303   4.464 8.03e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1188.1  on 860  degrees of freedom
Residual deviance: 1167.6  on 859  degrees of freedom
AIC: 1171.6

Number of Fisher Scoring iterations: 4

# una diferencia que vemos entre este summary y el de nls, gls o lm es
# que se provee el estadistico z en vez del estadístico t. Esto se debe a
# que los estimadores de los parámetros no siguen una distribución t,
# incluso en condiciones ideales, y entonces se suele utilizar
# z (aproximación asintótica de t).

# Además este summary provee el número de iteraciones que fueron necesarias
# para la convergencia ("Number of Fisher Scoring iterations")

# Null deviance es conceptualmente parecido a la SCTot del ANOVA
# (pero cuidado, recordar que el ANOVA supone distribución normal en los
# residuos, no es el caso que venimos discutiendo y por lo tanto no podemos
# usar ANOVA).

# Null Deviance = -2 (LogLik(modelo nulo) - LogLik(modelo saturado o completo))
# Null Deviance = -2 (LogLik(modelo nulo) - 0)

# El modelo saturado tiene un parámetro por cada observación y por lo tanto
# la bondad de ajuste (sin corregir por el número de parámetros) es máxima.
# Como el Lik del modelo saturado es = 1, el ln(1) = 0

# Residual deviance es conceptualmente parecido a la SCE del ANOVA
# Residual deviance = -2 (LogLik(modelo propuesto) - LogLik(modelo saturado o
completo))
```

```
# Residual deviance = -2 (LogLik(modelo propuesto) - 0)
# Es decir que la "residual deviance" aquí en glm es lo mismo que en
# nls llamabamos como "deviance" a secas. Comprobemos:
c(modelo$deviance, -2*logLik(modelo))


# Ambos son cocientes de verosimilitudes. En general a los cocientes de
# verosimilitudes también se los conoce como "deviance". En terminos
# generales, las clases anteriores, habíamos discutido:
# Lik Ratio = -2 (LogLik(modelo reducido) - LogLik(modelo grande))


# Análisis de deviance secuencial
anova(modelo,test="Chisq")
# Plantear las hipótesis de interés
# ¿Cuáles modelos se están comparando? ¿Son anidados?
```

20.5 Escalas de expresión del modelo

20.5.1 Escala variable respuesta

```
par(mfrow=c(1,1))
plot(jitter(datos$ingreso, .8), jitter(datos$voto, .8), xlim=c(-15,20),
    pch=20, cex=1, ylab="Probabilidad de votar a Bush", xlab="Ingreso")
curve((exp(-1.00010+0.28138*x))/(1+exp(-1.00010+0.28138*x)), add=T)
# En todos los casos la probabilidad es lo que describe la línea pero los
# puntos son los valores de la variable respuesta. En este caso la
# variable respuesta sólo puede tomar dos valores, cero (voto a clinton) o
# uno (voto a Bush). Cada punto refleja el voto de una persona.

# El mismo gráfico pero solo en el rango de estudio de la variable x
plot(jitter(datos$ingreso, .8),jitter(datos$voto, .8), pch=20, cex=1,
    ylab="Probabilidad de votar a Bush", xlab="Ingreso")
curve((exp(-1.00010+0.28138*x))/(1+exp(-1.00010+0.28138*x)),
      add=T)

# Aquí también podemos usar la función predict
plot(jitter(datos$ingreso, .8),jitter(datos$voto, .8), pch=20, cex=1,
    ylab="Probabilidad de votar a Bush", xlab="Ingreso")
pred_response=predict(modelo, type="response")
# El type es importante aquí para que nos de la
# escala de la variable respuesta.
points(datos$ingreso,pred_response,type="l")


# Interpretar el siguiente valor predicho
newdata=data.frame(ingreso=3)
predict(modelo,newdata, type="response")
```

20.5.2 Escala logit

```
# El glm estimó los parámetros de un modelo lineal en escala logit por el
# método de máxima verosimilitud, es decir:
# logit(p)= ln(p / q) = -1.00010 + 0.28138 * ingreso
curve(-1.0010 + 0.28138 * x, xlim=c(-15,20),
     ylab="ln(prob votar Bush / 1 - Prob votar Bush)", xlab="Ingreso",
     col="blue")

# Podemos agregar los valores predichos en la escala logit
pred_logit=predict(modelo,type="link")
# el "link" significa en la escala de la función de enlace, en este caso logit
points(datos$ingreso,pred_logit,type="l", col="red", lw=4)
# Los valores son de logit(probabilidad), si queremos los datos de
# "probabilidad" a secas, tenemos que usar la función "logit-1"
# (inversa de la función de enlace logit), es decir:
inv = 1 / (1 + exp(-1*pred_logit))
# Entonces obtuvimos con la función de arriba los valores de probabilidad,
# es decir, en la escala de interés o "response". Por lo tanto es lo mismo
# que hicimos arriba cuando  pusimos type="response" en lugar de type="link".
# Veamos:
plot(pred_response, inv)
abline(a=0, b=1, col="magenta", lw=3)
## voilà!!!


# Interpretar el intervalo de confianza para cada parámetro
confint(modelo)
```

20.5.3 Escala Odd

```
# Entonces p/q = exp(-1.00010 + 0.28138 * ingreso)
# odd = exp(-1.00010 + 0.28138 * datos$ingreso)
curve(exp(-1.00010 + 0.28138 * x), xlim=c(-15,20),
     ylab="prob votar Bush / 1 - Prob votar Bush", xlab="Ingreso")
# odd = 1  es equivalente a p = 0.5 y q = 0.5
# OJO recordar que este es el odd, no el COCIENTE de odds (= Odds ratio)
# que discute Anderson et al. 2008 en el texto que deben leer
```

20.6 Bondad de ajuste

```
mod_nulo = glm (voto ~ 1, family=binomial(link="logit"), data=datos)

AIC(modelo, mod_nulo)
```

```
          df      AIC
modelo     2 1171.593
mod_nulo   1 1190.064
```

```
BIC(modelo, mod_nulo)
```

```
df      BIC
modelo    2 1181.109
mod_nulo  1 1194.822
```

```
c(logLik(modelo), logLik(mod_nulo))

# PseudoR2 de McFadden
PseudoR2 = 1 - as.vector((logLik(modelo) / logLik(mod_nulo)))
PseudoR2

# Alternativamente D2
D2 = (mod_nulo$deviance - modelo$deviance) / mod_nulo$deviance
D2
# Esta última manera de expresarlo es todavía más parecida a la del r2 que
# conocemos. Recordar
# r2 = (SCTot - SCE) / SCTot
# Recordar también que null deviance es conceptualmente similar a SCTot
# y que residual deviance es conceptualmente similar a SCE.

# PseudoR2 y D2 dan el mismo valor en una binomial con N'= 1 pero dan
# valores "parecidos" en una binomial con N' mayor a 1
```

Trabajo Práctico Nº 21: ANDEVA y otros componentes de modelos "GLM"

21.1 Componentes de los modelos "GLM"

```
# Recordamos entonces que los GLM tienen tres componentes:

# .- COMPONENTE ALEATORIO: especifica la distribución condicional de la
variable
# respuesta en función de las variables independentientes. La función glm de R
# considera distribuciones de la "familia exponencial": Binomial, Poisson,
# Normal, Gamma, Normal inversa
# Las últimas dos distribuciones son para variables cuantitativas asimétricas
# positivas, lo veremos más adelante. Excepto por la normal (= gaussiana), las
# otras distribuciones NO aceptan valores negativos.

# .- COMPONENTE DETERMINISTA: sólo se permiten funciones lineales en glm

# .-FUNCIÓN DE ENLACE
```

21.2 Problema y Datos

```
GRÁFICO Y MODELO
# Datos del paquete AER: Applied Econometrics with R
datos=read.table("datos_p_21.txt")

str(datos)
```

```
# Este es un ejemplo de Economía del Trabajo que se obtuvo del libro
# Kleiber & Zeileis 2008 Applied Econometrics with R.
# Se encuestaron 872 mujeres en Suiza. Las variables son:
# participation: si la mujer encuestada trabaja.
# income: ingreso
# age: edad
# education: educación
# youngkids: el número de hijos jóvenes
# oldkids: el número de hijos grandes
# foreign: si es extranjero o ciudadano suizo

# gráfico interesante
par(mfcol = c(1, 2))
plot(density(datos$income[datos$participation == "no"]),
main = "", xlab = "Ingreso", ylim=c(0,1.6), lwd = 2)
lines(density(datos$income[datos$participation == "yes"]),
main = "", xlab = "", ylab = "", lty = 3, col="red",lwd = 2)

plot(density(datos$education[datos$participation == "no"]),
    main = "", xlab = "Educación (años)", lwd = 2, ylim=c(0,0.18))
lines(density(datos$education[datos$participation == "yes"]),
main = "", xlab = "", ylab = "", lty = 3, col="red", lwd = 2)

legend("topright", legend=c("desocupada","ocupada"), col=c("black","red"),
lty=c(1,3),lwd=2)
```

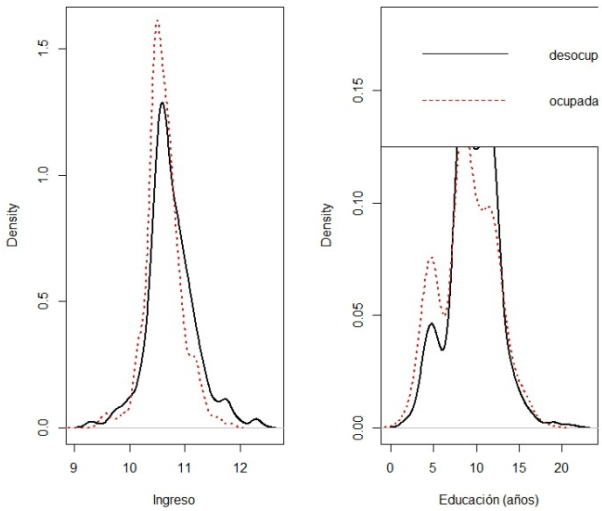


Figura 21.1: Densidad según ingreso y educación dependiendo de si la mujer se encuentra ocupada o desocupada.

```
# otro gráfico interesante es:
plot(participation~age, data=datos) # lleva el nombre de "spinogram"
# se observa una tendencia no lineal  (en particular, similar a una parábola)
# entre la proporción de mujeres empleadas y la edad. Por lo tanto, en el modelo
# convendría tener en cuenta este aspecto.
```

```
# El "ancho" de cada barra es proporcional a la cantidad de observaciones en ese
# intervalo.
# Modelo
age2=datos$age^2
modelo=glm(participation~income+age+age2+education+youngkids+oldkids+foreign,
        data=datos, family = binomial(link = "logit"))
```

```
Call:
glm(formula = participation ~ income + age + age2 + education +
    youngkids + oldkids + foreign, family = binomial(link = "logit"),
    data = datos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9061  -0.9627  -0.4924   1.0171   2.3915

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  6.19639    2.38309   2.600  0.00932 **
income      -1.10409    0.22571  -4.892 1.00e-06 ***
age          3.43661    0.68789   4.996 5.86e-07 ***
age2        -0.48764    0.08519  -5.724 1.04e-08 ***
education    0.03266    0.02999   1.089  0.27611
youngkids   -1.18575    0.17202  -6.893 5.46e-12 ***
oldkids     -0.24094    0.08446  -2.853  0.00433 **
foreignyes   1.16834    0.20384   5.732 9.94e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1203.2  on 871  degrees of freedom
Residual deviance: 1017.6  on 864  degrees of freedom
AIC: 1033.6

Number of Fisher Scoring iterations: 4
```

```
# el análisis considera yes=1 y no=0
summary(modelo)

# interpretar el siguiente intervalo de confianza para cada parámetro
confint(modelo)
```

21.3 ANDEVA

```
# Análisis de deviance (ANDEVA) secuencial
anova(modelo,test="Chisq")
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: participation

Terms added sequentially (first to last)
```

```
          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                        871     1203.2
income     1   27.251       870     1176.0 1.786e-07 ***
age        1    7.491       869     1168.5  0.006199 **
age2       1   64.958       868     1103.5 7.652e-16 ***
education  1    1.217       867     1102.3  0.269899
youngkids  1   39.907       866     1062.4 2.663e-10 ***
oldkids    1   10.111       865     1052.3  0.001474 **
foreign    1   34.717       864     1017.6 3.813e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Son pruebas de cociente de verosimilitud secuenciales
# Recordemos que genéricamente:
# Deviance = Lratio= -2(logLik(mod_simple) - logLik(mod_completo))

# Estimemos entonces el primer término de la tabla
mod_nulo=glm(participation~1,data=datos, family = binomial(link = "logit"))
mod_income=glm(participation~income,data=datos, family = binomial(link = "logit"))
Deviance_1=-2*(logLik(mod_nulo)-logLik(mod_income))
Deviance_1
pchisq(Deviance_1,df=1,lower.tail=F)
# vemos que da igual a la fila "income" del análisis de Deviance (ANDEVA)
anova(modelo,test="Chisq")

# Segundo término de la tabla de ANDEVA
mod_incomeyage=glm(participation~income+age,data=datos,
                   family = binomial(link = "logit"))
Deviance_2=-2*(logLik(mod_income)-logLik(mod_incomeyage))
Deviance_2
pchisq(Deviance_2,df=1,lower.tail=F)
#comparamos con
anova(modelo,test="Chisq")


# Recordemos que:
# Genéricamente Deviance es igual a Lratio pero siempre en relación al modelo
# "full" por eso se dice que es cuánto nos alejamos ("deviance") del modelo full
# Null deviance = -2(logLik(mod_nulo)-logLik(mod_full))
# Residual deviance = -2(logLik(mod_interés)-logLik(mod_full))
# Es decir que Null y Residual Deviance hacen referencia a Lratio específicos
# de interés
```

21.4 Bondad de ajuste

```
# PseudoR2 de McFadden
PseudoR2 = 1 - as.vector(logLik(modelo) / logLik(mod_nulo))
PseudoR2
```

```
[1] 0.1542966
```

```
# Alternativamente D2
D2=(mod_nulo$deviance-modelo$deviance) / mod_nulo$deviance
D2
# Esta última manera de expresarlo es todavía más parecida a la del r2 que
# conocemos. No hay aún una versión aceptada respecto de la mejor forma de
# obtener (pseudo) r2 (o D2) para modelos lineales generalizados


# interpretar el siguiente valor predicho
newdata=data.frame(income=11,age=4,age2=16,education=10, youngkids=2,
oldkids=2,
                   foreign="yes")
predict(modelo,newdata, type="response")
```

```
1
0.2037744
```

21.4 Función de verosimilitud

```
# En papel, plantee la función de verosimilitud para este ejemplo.
```

Trabajo Práctico Nº 22: Función binomial y su expresión a través de diferentes escalas

22.1 Problema y Datos

```
# Datos obtenidos de Gelman A y Hill J (2007) Data analysis using regression and
multilevel/hierarchical models.
# Ed. Cambridge University Press.

# Se postula que en aquellos pueblos con mayores ingresos las personas son más
# propensas a votar al partido conservador en los Estados Unidos de América.
# Para ello en cada uno de 57 pueblos se encuestaron 15 personas a las que se
# les preguntó su ingreso y si votarían a Bush (favor) o Clinton (contra).

datos=read.table("datos_p_22.txt")
str(datos)
# Proporción: sabemos el límite inferior (0 votos) y el límite superior
# (15 votos).

# Tradicionalmente estos datos podrían haberse analizado utilizando una
# distribución normal y como variable respuesta el % de votos a Bush.
# Habitualmente existen tres problemas con esto:
# 1.- Los errores no están normalmente distribuidos (en parte porque la
# respuesta está acotada entre un límite inferior y uno superior)
# 2.- La varianza no es constante.
```

```
# 3.- Calculando el porcentaje, perdemos información sobre el tamaño de la
# sub-muestra, N', a partir de la cual se estima la proporción. En este caso es
# 15, y es igual para todos los pueblos pero podría no serlo. La función glm
# realiza estimaciones de los parámetros ponderando por el tamaño de la
# submuestra de cada unidad experimental.
plot(datos$ingreso, (datos$favor/(datos$favor+datos$contra)))

hist(datos$favor)

modelo=glm(cbind(favor,contra)~ingreso,family=binomial(link="logit"), data=datos)
summary(modelo)

Call:
glm(formula = cbind(favor, contra) ~ ingreso, family = binomial(link = "logit"
),
    data = datos)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.95852  -0.64708   0.03597   0.61489   1.88819

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.5968     0.6948  -2.298   0.0216 *
ingreso       0.4815     0.2326   2.070   0.0384 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 64.572  on 56  degrees of freedom
Residual deviance: 60.258  on 55  degrees of freedom
AIC: 241.07

Number of Fisher Scoring iterations: 4

# ¿Que unidades tiene el AIC?

anova(modelo,test="Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(favor, contra)

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                       56     64.572
ingreso  1   4.3136        55     60.258  0.03781 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Escribir el modelo.
# ¿Cuál es la función de enlace?

# ATENCIÓN: la función de enlace (g) se aplica sólo sobre el componente
# determinista, no se aplica sobre el error (E):
# y = g^-1(X*b) + E
# En cambio una típica transformación de la variable respuesta afecta ambos
# componentes del modelo:
# g(y) = X*b + E
# Es decir que función de enlace y transformación de la variable respuesta NO
# son sinónimos

# Este modelo estimado lo expresaremos de tres maneras:
######## ESCALA VARIABLE RESPUESTA ###########
par(mfcol=c(1,1))
plot(datos$ingreso, (datos$favor/(datos$favor+datos$contra)),
    xlim=c(-10,15), ylim=c(0,1), pch=20, cex=1,
    ylab="Proporción de votos a Bush", xlab="Ingreso")
curve((exp(-1.5968+0.4815*x))/(1+exp(-1.5968+0.4815*x)),
      add=T)
```

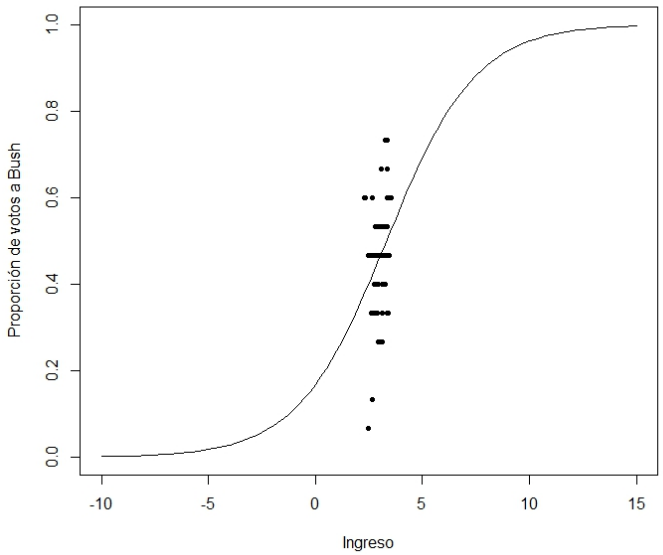


Figura 22.1.: Proporción de votos a Bush en función del Ingreso, con la correspondiente curva estimada, en escala de variable de respuesta.

```
# considerando sólamente el rango de estudio (dominio del modelo)
plot(datos$ingreso, (datos$favor/(datos$favor+datos$contra)),
    pch=20, cex=1,
    ylab="Proporción de votos a Bush", xlab="Ingreso")
curve((exp(-1.5968+0.4815*x))/(1+exp(-1.5968+0.4815*x)),
      add=T)


# También podemos usar la función predict
```

```
plot(datos$ingreso, (datos$favor/(datos$favor+datos$contra)),
    pch=20, cex=1,
    ylab="Proporción de votos a Bush", xlab="Ingreso")
pred_response=predict(modelo, type="response") # el type es importante para obtener
# los valores predichos en la escala de la variable respuesta.
points(datos$ingreso,pred_response,type="l")

# finalmente el gráfico para el número de votos a favor en función del ingreso
plot(datos$ingres,datos$favor)
# como en todos los pueblos N'=15
curve(((exp(-1.5968+0.4815*x))/(1+exp(-1.5968+0.4815*x)))*15,
     add=T, col="brown", lw=3)


# interpretar el siguiente valor predicho
newdata=data.frame(ingreso=3)
predict(modelo,newdata, type="response")

        1
0.4620359

####### ESCALA LOGIT ###############
# La función glm estimó los parámetros de un modelo lineal en escala logit por
# el método de máxima verosimilitud, es decir que
# logit(p)= ln(p / q) = -1.5968 + 0.4815 * ingreso

curve(-1.5968 + 0.4815 * x, xlim=c(-5,5),
     ylab="ln(prob votar Bush / 1 - Prob votar Bush)", xlab="Ingreso")

# podemos agregar los valores predichos en la escala logit
pred_logit=predict(modelo,type="link")
points(datos$ingreso,pred_logit,type="l", col="red", lwd=7)


# interpretar el siguiente intervalo de confianza para los parámetros
confint(modelo)


####### ESCALA ODD #########
# entonces p/q= exp(-1.5968 + 0.4815 * ingreso)
# odd=exp(-1.5968 + 0.4815 * datos$ingreso)
curve(exp(-1.5968 + 0.4815 * x), xlim=c(-15,20),
     ylab="prob votar Bush / 1 - Prob votar Bush", xlab="Ingreso")
# odd = 1  es equivalente a p = 0.5 y q = 0.5
# ojo recordar que este es el odd, no el COCIENTE de odds que discute Anderson
# et al. 2008 en el texto que deben leer

######## Bondad de ajuste ############
mod_nulo=glm (cbind(favor,contra) ~ 1, family=binomial(link="logit"),data=datos)

AIC(modelo,mod_nulo)
```

```
df      AIC
modelo    2 241.0685
mod_nulo  1 243.3822
```

```
BIC(modelo,mod_nulo)
```

```
  df      BIC
modelo    2 245.1546
mod_nulo  1 245.4252
```

```
logLik(modelo)
```

```
'log Lik.' -118.5343 (df=2)
```

```
logLik(mod_nulo)
```

```
'log Lik.' -120.6911 (df=1)
```

```
# PseudoR2 de McFadden
PseudoR2 = 1 - as.vector(logLik(modelo) / logLik(mod_nulo))
PseudoR2
```

```
[1] 0.01787057
```

```
# Alternativamente D2
D2=(mod_nulo$deviance-modelo$deviance) / mod_nulo$deviance
D2
```

```
[1] 0.06680395
```

```
# Esta última manera de expresarlo es todavía más parecida a la del r2 que
# conocemos. D2 y Pseudo R2 dan el mismo valor en una binomial con N=1 sin
# sobredispersión, pero no en una binomial con N'>1


# observados proporción vs. predichos
plot(pred_response,(datos$favor/(datos$favor+datos$contra)))
abline(a=0,b=1, col="red", lwd=5)
# observados votos a favor vs. predichos
# como N' = 15 para todos los pueblos:
plot(pred_response*15,datos$favor)
abline(a=0,b=1, col="red", lwd=5)
# igual a:
plot(pred_response*15,(datos$favor/(datos$favor+datos$contra))*15)
abline(a=0,b=1, col="red", lwd=5)

# ¿Que información obtenemos de estos gráfico?

# ¿Cuál es la diferencia con el siguiente gráfico?
```

```
# observados vs. variable independiente
plot(datos$ingres,datos$favor)
curve(((exp(-1.5968+0.4815*x))/(1+exp(-1.5968+0.4815*x)))*15,
     add=T, col="brown", lw=3)

##### SUPUESTOS #####
residuos=resid(modelo, type="deviance")
# IMPORTANTE: estos son los residuos "deviance".
# Ver más abajo explicación sobre residuos.

# residuos vs. predichos
plot(pred_response,residuos)
abline(a=0, b=0, col="violet", lw=5)
```

22.2 Sobredispersión y Chi^2

```
# La ausencia de sobredispersión es un supuesto clave.

# Antes, veamos algunos aspectos de la distribución de Chi^2 que usaremos para
# evaluar sobredispersión. La misma surge de sumas de normales estándar (z) al
# cuadrado y tiene un único parámetro = grados de libertad

# Chi^2 con 1 gl = z^2
# Chi^2 con 2 gl = z^2 + z^2
# Chi^2 con 3 gl = z^2 + z^2 + z^2
# Chi^2 con 4 gl = z^2 + z^2 + z^2 + z^2

chi_1 = rnorm(n=9999, mean=0, sd=1)^2
chi_2 = rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2
chi_3 = rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2 +
rnorm(n=9999, mean=0, sd=1)^2
chi_4 = rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2 +
rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2


# Comparemos gráficamente con la distribución chi
par(mfcol=c(2,2))
hist(chi_1, freq=F)
lines(density(chi_1), col="blue")
# es igual a la distribución chi con grados de libertad=1
lines(density(rchisq(n=9999,df=1)), col="red")

hist(chi_2, freq=F)
lines(density(chi_2), col="blue")
# es igual a la distribución chi con grados de libertad=2
lines(density(rchisq(n=9999,df=2)), col="red")

hist(chi_3, freq=F)
lines(density(chi_3), col="blue")
# es igual a la distribución chi con grados de libertad=3
```

```
lines(density(rchisq(n=9999,df=3)), col="red")

hist(chi_4, freq=F)
lines(density(chi_4), col="blue")
# es igual a la distribución chi con grados de libertad=4
lines(density(rchisq(n=9999,df=4)), col="red")
```

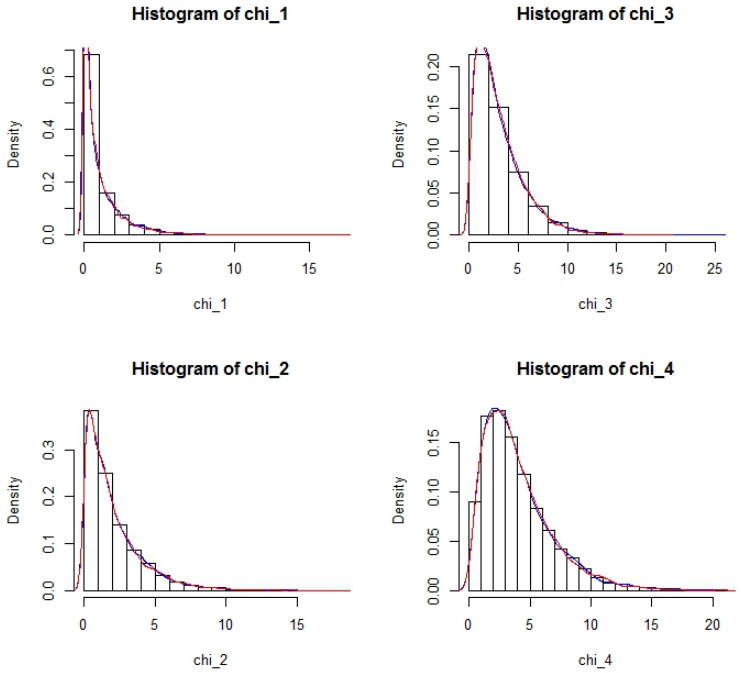


Figura 22.2.: Distribución de los datos muestrales ajustados según distribución chi con grados de libertad 1 a 4 vs. los de varias distribuciones chi estándar con los mismos grados de libertad (parámetro).

```
# La media de la Chi^2 es igual a los grados de libertad (el único parámetro de
# la distribución).
# Veamos, para un grado de libertad
mean(rchisq(n=999999, df=1))
# para dos grados de libertad
mean(rchisq(n=999999, df=2))
# para tres grados de libertad
mean(rchisq(n=999999, df=3))
# y ya que R hace el trabajo, veamos para cuatro grados de libertad también.
mean(rchisq(n=999999, df=4))


# Luego de esta pequeña descripción de algunas características de la
# distribución de Chi^2, volvamos al ejemplo de la distribución binomial y la
# sobredisperción.

# Si la especificación del modelo binomial es correcta, la Devianza residual
# debería seguir una distribución de Chi cuadrado con los grados de libertad
# residuales (cuando N' es grande). Recordemos que la media de la Chi^2 es igual
# a los grados de libertad residuales, por lo tanto, se "espera" que la
# "Residual deviance" sea igual a los grados de libertad residuales.
```

```
# Si la "Residual deviance" es mayor que los "residual degrees of freedom"
# estamos en presencia de sobre-dispersión. En este caso la s2 no es igual a
# N' * p * q.


# La falta de cumplimiento en los supuestos se puede deber, por ejemplo, a que
# no hemos incluido los predictores adecuados en el modelo o a que los datos
# no tienen distribución binomial.

summary(modelo)

# En este caso la residual deviance es ~60 y los grados de libertad residuales
# son 55
60.258/55
# Cercano a 1... vamos bien :D

# También podemos evaluar la hipótesis nula que dice que "Residual deviance"
# proviene de una distribución Chi^2 con df = grados de libertad residuales
pchisq(q = deviance(modelo),df = df.residual(modelo),lower=FALSE)
# No rechazamos la H0)
# Es razonable pensar entonces que el modelo es adecuado y que no hay
# sobredispersión..... ay... que alivio!!!
```

22.3 Residuos dentro del modelo GLM

```
# Dentro del contexto de GLM hay dos tipos de residuales que son comúnmente
# utilizados:
```

22.3.1 Residuos de Pearson

```
res_p=resid(modelo, type="pearson")
res_p
# Estos residuos los hemos discutido en detalle previamente cuando vimos modelos
# con heterogeneidad de varianzas. Se expresan como
# (yi - E(yi)) / sqrt(Var(yi))
# Los residuos de tipo Pearson tienen en cuenta las predicciones del modelo
# tanto por cambios en la media como en la varianza ante cambios en la(s)
# variable(s) independiente(s). Por ejemplo, modelar "p" en una distribución
# binomial ante cambios en el ingreso, implica que la media (p*N') y la varianza
# (p*q*N') de la variable respuesta cambian con el ingreso.
```

22.3.2 Residuos Deviance

```
# Estos residuos también tienen en cuenta cambios en la media y en la varianza
# según una distribución binomial en este caso. La función utilizada para
# obtener los "deviance residuals" depende de la distribución estocástica elegida.
res_dev=resid(modelo, type="deviance")
res_dev
# ¿Cuántos "deviance residuals" = "residuos deviance" hay en este ejemplo?
```

```
# ATENCIÓN: "residual deviance" está relacionado pero no es lo mismo que
# "deviance residuals". Así como la SCE es la suma de los residuos elevados al
# cuadrado, la "Residual deviance" es la suma de los "deviance residuals" al
# cuadrado.
sum(res_dev^2)
```

```
[1] 60.25796
```

```
# es igual a
deviance(modelo)
```

```
[1] 60.25796
```

```
# también reportada al final del summary
summary(modelo)

# En cambio la suma de los cuadrados de los residuos de pearson NO es igual a
# la devianza
sum(res_p^2)
# distinto a
deviance(modelo)


#### Los residuales ordinarios (o en escala z) no tienen sentido aquí ya que
# suponen homogeneidad de varianzas.
# ei = Yi-E(yi)
res_resp=resid(modelo, type="response")
res_resp
# para el primer pueblo
res_resp[1]
# valores predichos para el primer pueblo
newdata=data.frame(ingreso=datos$ingreso[1])
p1=predict(modelo,newdata, type="response")

# residuales
(datos$favor[1]/(datos$favor[1]+datos$contra[1])) - p1
# equivalente a
res_resp[1]
# si queremos expresarlo en términos de número de votantes a favor lo
# multiplicamos por 15

# Es decir que en este curso vimos tres tipos de residuos:
  # .- Ordinarios
  # .- De Pearson
  # .- Deviance

##### FUNCIÓN VEROSIMILITUD ####
# En papel, plantee la función de verosimilitud para este ejemplo.
```

Trabajo Práctico Nº 23: Distribución Gamma y Chi^2

23.1 Problema y Datos

```
# primero agradecemos a las personas que han trabajado y que llevan adelante el
projecto R
contributors()

# datos obtenidos de www.gapminder.org
datos=read.table("datos_p_9.txt")
colnames(datos)=c("pais","vida","pbi")

# Estamos interesados en modelar la esperanza de vida en función del pbi
# entre distintos países.
# Previamente habíamos observado que la relación entre ambas variables no era lineal
plot(datos$pbi,datos$vida)
plot(log10(datos$pbi),datos$vida) # con log 10 pbi

# También observamos luego de ajustar un número amplio de modelos
# que los residuos indicaban que la distribución era asimétrica.

# La distribución Gamma es interesante para modelar un amplio rango de procesos
# en los que:
# 1) la variable respuesta es cuantitativa contínua.
# 2) la variable respuesta toma valores mayores a cero.
# 3) la distribución de la variable respuesta es asimétrica positiva
# (con una cola larga hacia la derecha, es decir hacia valores grandes)
# 4) la varianza aumenta más que linealmente con la media
# La gamma presenta CV (coeficiente de variación) constante e igual a
sqrt("shape")...sigan leyendo para entender.
# 5) la distribución de la variable respuesta puede tomar distintos valores de curtosis
# sin embargo, según lo que vimos anteriormente la variable de respuesta (esperanza
# de vida) era asimétrica negativa en vez de positiva:
mod_normal=lm(datos$vida~log10(datos$pbi))
plot(log10(datos$pbi),datos$vida)
abline(mod_normal,col="green",lwd=5)
```

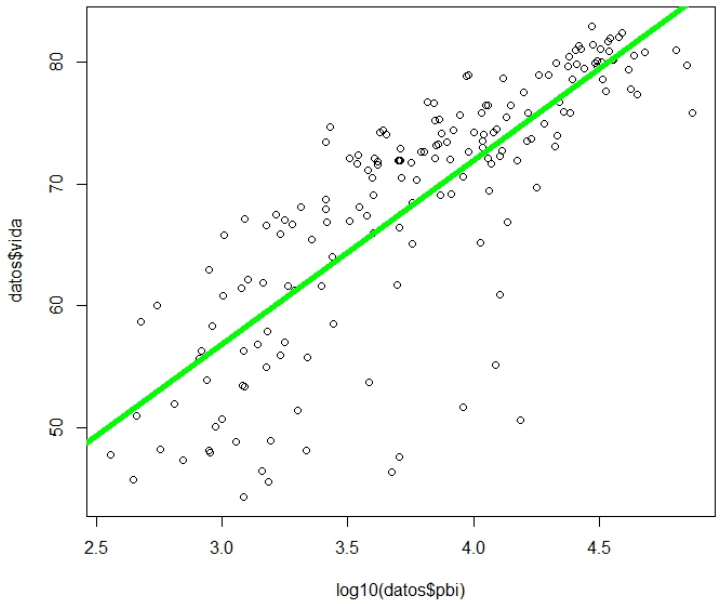


Figura 23.1.: Esperanza de vida en función del logaritmo base 10 del PBI con recta estimada de regresión lineal simple.

```
plot(predict(mod_normal),resid(mod_normal))
abline(a=0,b=0, col="green", lwd=5)

hist(resid(mod_normal))
# Entonces sabemos que posiblemente no se cumplan los supuestos del modelo
# que asume distribución Gamma.
# Más abajo ajustaremos un modelo con distribución Gamma y evaluaremos estos
aspectos.
```

23.2 Gamma

```
# Aquí seguimos la parametrización de Gamma según Faraway 2006, que es la utilizada
en
# la función glm (ojo que dgamma usa una parametrización ligeramente distinta).
# Entonces tenemos dos parámetros, la media y la "forma (shape)".
# Varianza = media^2 / forma
# Si asumimos un valor de "forma" constante. La relación entre varianza y media es:
curve (x^2 / 7)

# con otro valor de forma:
curve (x^2 / 3)

# Algunas distribuciones gamma con distinta forma:
# para valores grandes de forma (= shape) la distribución gamma se vuelve simétrica
# y con menor curtosis.
# En estos casos una distribución normal podría ser utilizada
# para modelar nuestros datos.
x<-seq(from=0.01,to=28,by=.01)
y=matrix(ncol=9, nrow=length(x))
```

```
par(mfrow=c(3,3))
y[,1]<-dgamma(x,shape=0.5)
plot(x,y[,1],type="l")
y[,2]<-dgamma(x,shape=1)
plot(x,y[,2],type="l")
y[,3]<-dgamma(x,shape=2)
plot(x,y[,3],type="l")
y[,4]<-dgamma(x,shape=3)
plot(x,y[,4],type="l")
y[,5]<-dgamma(x,shape=4)
plot(x,y[,5],type="l")
y[,6]<-dgamma(x,shape=5)
plot(x,y[,6],type="l")
y[,7]<-dgamma(x,shape=6)
plot(x,y[,7],type="l")
y[,8]<-dgamma(x,shape=7)
plot(x,y[,8],type="l")
y[,9]<-dgamma(x,shape=15)
plot(x,y[,9],type="l")
# la distribución gamma no acepta y=0.
```

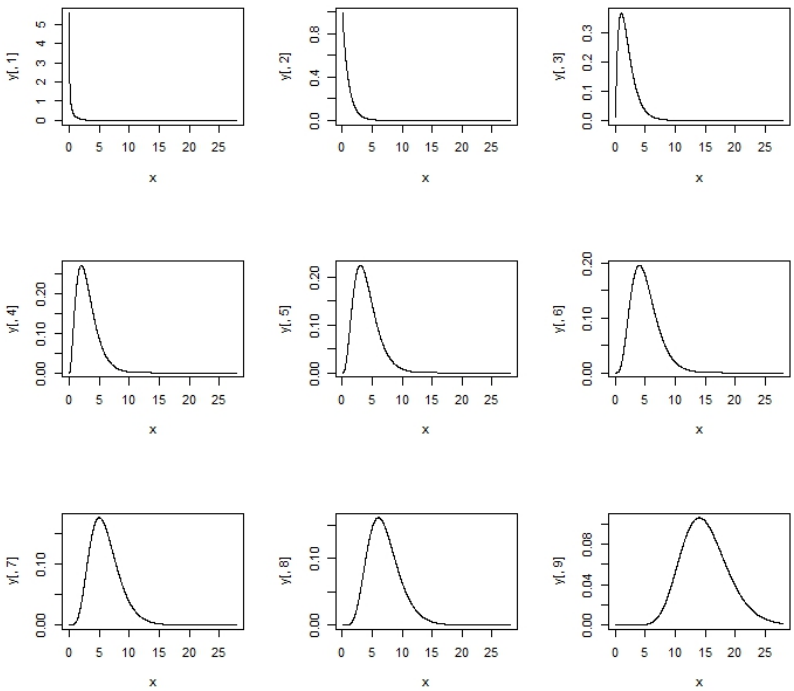


Figura 23.2.: Función de densidad Gamma para diferentes "shapes".

```
# ¿Que unidades presenta el eje y?

library(moments)
kurtosis(y[,1:9])
# a medida que aumenta el "shape" (forma) también disminuye la curtosis.
# Esta es una propiedad interesante que usaremos en el próximo script.
```

```
# Recordemos que la curtosis de una normal es = 3

skewness(y[,1:9])
# siempre asimetría positiva a medida que aumenta el "shape" disminuye la asimetría

# ¿Cuál es el valor de asimetría para una distribución normal?
```

23.3 Chi^2

```
# Revisar script 22 con una presentación de la distribución de Chi 2

# La distribución Chi2 es un caso particular de la Gamma. Mientras que la Gamma
# tiene dos parámetros, la Chi2 tiene un solo parámetro (i.e. los grados de libertad).

# Según la parametrización de dgamma, la Chi2 tiene un rate constante igual a 0.5,
# mientras que el shape son los grados de libertad dividido 2
# (es decir, la media dividido 2).
# Si
# Media = shape / rate
# Media = shape / 0.5
# Media = shape * 2
# Media / 2 = shape

# Veamos cómo cambia la distribución de Chi2 para distintos grados de libertad:
x<-seq(from=0.01,to=80,by=.01)
y=matrix(ncol=9, nrow=length(x))
par(mfrow=c(3,3))
y[,1]<-dgamma(x,shape=1/2,rate=0.5)
plot(x,y[,1],type="l")
y[,2]<-dgamma(x,shape=5/2,rate=0.5)
plot(x,y[,2],type="l")
y[,3]<-dgamma(x,shape=10/2,rate=0.5)
plot(x,y[,3],type="l")
y[,4]<-dgamma(x,shape=15/2,rate=0.5)
plot(x,y[,4],type="l")
y[,5]<-dgamma(x,shape=20/2,rate=0.5)
plot(x,y[,5],type="l")
y[,6]<-dgamma(x,shape=25/2,rate=0.5)
plot(x,y[,6],type="l")
y[,7]<-dgamma(x,shape=30/2,rate=0.5)
plot(x,y[,7],type="l")
y[,8]<-dgamma(x,shape=35/2,rate=0.5)
plot(x,y[,8],type="l")
y[,9]<-dgamma(x,shape=40/2,rate=0.5)
plot(x,y[,9],type="l")

# Por ejemplo, obtenemos 10000 valores aleatorios de una distribución
# Chi2 con df= 20
a=rchisq(10000,df =20)
```

```
a
# Nos da lo mismo que obtener 10000 valores de una distribución
# gamma con shape = 20/2 y rate=0.5
b=rgamma(10000,shape = 20/2, rate=0.5)
b
c(mean(a),mean(b))

par(mfrow=c(2,1))
hist(a, main="chisq")
hist(b, main="gamma")
```

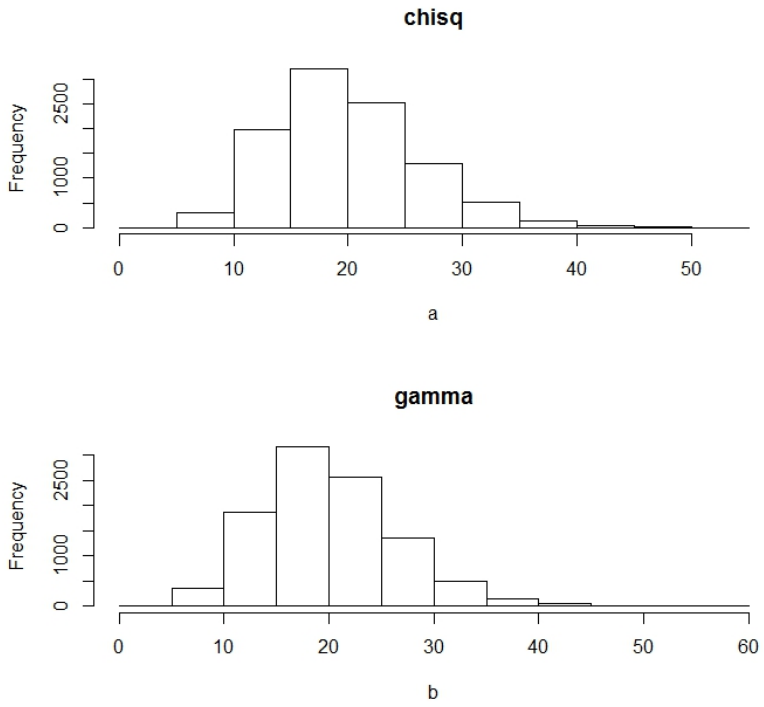


Figura 23.3.: Histogramas para una distribución chi cuadrado de 10.000 valores aleatorios con 20 g.l. y de una distribución gamma con shape 20/2 y rate 0,5.

```
# Finalmente mostramos que en la distribución Chi2 la media es igual a los grados de
# libertad
round(mean(rchisq(1000000,df=1)))
round(mean(rchisq(1000000,df=10)))
round(mean(rchisq(1000000,df=15)))
round(mean(rchisq(1000000,df=30)))
round(mean(rchisq(1000000,df=60)))
round(mean(rchisq(1000000,df=1000)))


####### MODELO ##########
mod_gamma=glm(vida~log10(pbi),family=Gamma(link=identity),data=datos)
summary(mod_gamma)

Call:
glm(formula = vida ~ log10(pbi), family = Gamma(link = identity),
```

```
    data = datos)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.36754  -0.03217   0.01325   0.05376   0.17282

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.9634     3.0826   3.232  0.00147 **
log10(pbi)   15.5153     0.8381  18.513  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.008722055)

    Null deviance: 4.4501  on 176  degrees of freedom
Residual deviance: 1.6672  on 175  degrees of freedom
AIC: 1175.3

Number of Fisher Scoring iterations: 4
```

```
# el parámetro de dispersión (lo llamaremos "pd") es la inversa de la forma (shape)
# Habíamos dicho que      Varianza (y) = media^2 / forma
# Entonces                Varianza (y) = media^2 * pd

# Es decir que la varianza es función de la media. A diferencia de algunos modelos
# con distribución normal en los que suponemos que hay una sola varianza residual

# veamos cómo varía la varianza con la media según el modelo gamma estimado
par(mfrow=c(1,1))
plot(datos$vida,datos$vida^2*0.008722055, xlab="Media", ylab="Varianza")

# Plantear el modelo en papel
# Plantear la función de verosimilitud


anova(mod_gamma,test="Chisq")
```

```
Analysis of Deviance Table

Model: Gamma, link: identity

Response: vida

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                         176     4.4501
log10(pbi)  1    2.783       175     1.6672 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# ¿Que indica la columna "Deviance"?
```

```
# comparamos los coeficientes con un modelo que asume distribución normal y vemos
que
# son similares
coef(mod_gamma)

(Intercept)  log10(pbi)
   9.963426   15.515256

coef(mod_normal)

(Intercept) log10(datos$pbi)
   11.61284         15.07488

# supuestos
# residuos vs. predichos
par(mfrow=c(1,1))
pred_response=predict(mod_gamma,type="response")
plot(pred_response,resid(mod_gamma, type="deviance"))
abline(a=0,b=0, col="red",lwd=4)
```

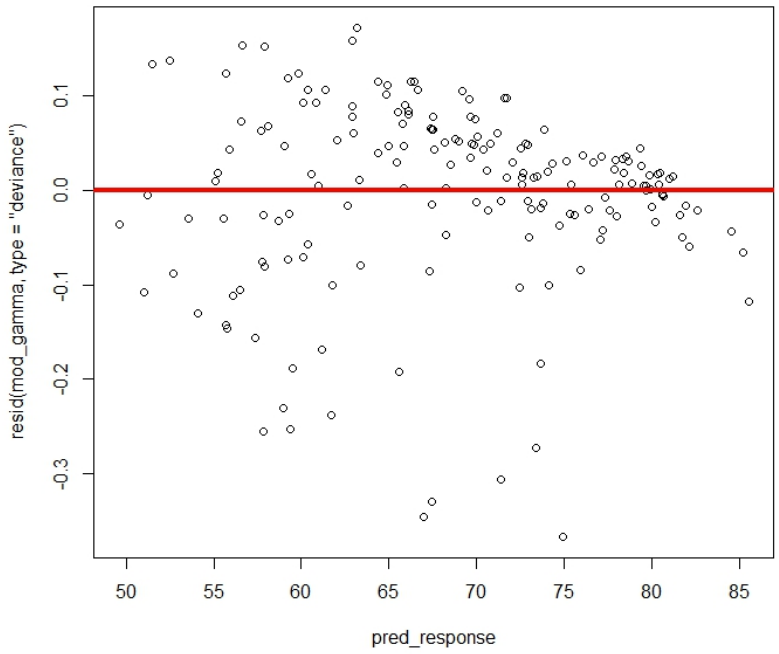


Figura 23.4.: Residuos vs. predichos para el modelo estimado según distribución Gamma.

```
# observados vs. variable independiente
plot(log10(datos$pbi),datos$vida)
abline(mod_gamma,col="red",lwd=4)

# observados vs. predichos
plot(pred_response,datos$vida)
```

```
abline(a=0, b=1,col="red",lwd=4)

hist(resid(mod_gamma, type="deviance"))


# Bondad de ajuste: D2 similar a R2
mod_nulo=glm(vida~1,family=Gamma(link=identity),data=datos)

D2=(mod_nulo$deviance-mod_gamma$deviance) / mod_nulo$deviance
D2
```

```
[1] 0.6253636
```

Trabajo Práctico Nº 24: Funciones Gamma vs. Normal

```
### Para próximo curso retomar discusión sobre multicolinealidad

#### Ver excelente blog en
# http://www.statisticalhorizons.com/multicollinearity
```

24.1 Problema y Datos

```
# Datos obtenidos de FAO (http://www.fao.org/index_es.htm) y Banco Mundial
(http://www.worldbank.org/)

datos=read.table("datos_p_24.txt")
str(datos)
colnames(datos)=c("pais","capital","PEA","fertilizante","cereal")
str(datos)

# cereal = producción de cereales (toneladas) en el año 2007
# capital = stock de capital en desarrollo de la tierra (millones de dólares)
# PEA = población económicamente activa (miles de personas)
# fertilizante = consumo de fertilizante nitrogenado (toneladas de nutrientes)

# Se desea estimar una función de producción para cereales a partir de los factores de
# producción mencionados anteriormente

# Dado el siguiente modelo
mod_normal=glm(cereal~capital+PEA+fertilizante,
family=gaussian(link="identity"),data=datos)

summary(mod_normal)
```

```
Call:
glm(formula = cereal ~ capital + PEA + fertilizante, family = gaussian(link =
"identity"),
    data = datos)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
```

```
-83006898   -2317292    -716837    2255558   91557589

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.109e+05  1.534e+06   0.463    0.644
capital       6.458e+02  6.158e+01  10.487  < 2e-16 ***
PEA          -3.181e+02  7.203e+01  -4.416 2.11e-05 ***
fertilizante  2.226e+01  1.863e+00  11.950  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 2.409032e+14)

    Null deviance: 4.5183e+17  on 132  degrees of freedom
Residual deviance: 3.1077e+16  on 129  degrees of freedom
AIC: 4787.7

Number of Fisher Scoring iterations: 2
```

```
# Como la residual deviance es la SCE, la varianza muestral residual es:
mod_normal$deviance / mod_normal$df.residual
```

```
[1] 2.409032e+14
```

```
# Vemos que es igual al mensaje del summary:
# "Dispersion parameter for gaussian family taken to be 2.409032e+14"

res_normal=resid(mod_normal, type="deviance")

hist(res_normal) # muy empinado.... "colas" gruesas... seguir leyendo...
qqnorm(res_normal)
qqline(res_normal)
# Que feo....
# Este es un típico patrón de una distribución con mayor curtosis que la normal
# ya que hay valores más chicos (a la izquierda del gráfico) y más grandes (a la derecha
# del gráfico) que lo esperado según una distribución normal.
# Veamos:
library(moments)
kurtosis(res_normal) # debería dar tres
```

```
[1] 18.74351
```

```
# ¡¡¡Que grande!!!!

# Probemos con una distribución Gamma
# en la misma cuanto menor es el "shape" (forma) mayor es la curtosis
mod_gamma=glm(cereal~capital+PEA+fertilizante,family=Gamma
(link="identity"),data=datos)

summary(mod_gamma)
```

```
Call:
```

```
glm(formula = cereal ~ capital + PEA + fertilizante, family = Gamma(link = "id
entity"),
    data = datos)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17491  -0.62380  -0.08872   0.30924   1.87582

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -9110.25    1277.48  -7.131 6.34e-11 ***
capital         96.39      39.13   2.463   0.0151 *
PEA            236.30      36.03   6.557 1.20e-09 ***
fertilizante    12.79       2.03   6.299 4.33e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.5280171)

    Null deviance: 565.228  on 132  degrees of freedom
Residual deviance:  82.058  on 129  degrees of freedom
AIC: 4207.5

Number of Fisher Scoring iterations: 10
```

```
# A diferencia de la normal,
# aquí el parámetro de dispersión (lo llamaremos "pd") es la inversa de la forma (shape)
# Habíamos dicho que      Varianza (y) = media^2 / forma
# Entonces                Varianza (y) = media^2 * pd
# No podemos dar un único valor de varianza residual (i.e. varianza constante) como
indicamos
# arriba en el modelo de distribución normal ya que la varianza aumenta con la media.

res_gamma=resid(mod_gamma, type="deviance")

hist(res_gamma)
qqnorm(res_gamma)
qqline(res_gamma)
kurtosis(res_gamma) # debería dar tres
```

```
[1] 3.176855
```

```
# ¡¡Que lindo!!!


# Motivo por el cual no comparamos con AIC o BIC ambos modelos según Faraway
2006
# (aquí se está discutiendo otro ejemplo pero se aplica también al nuestro)

# "These two models are not nested and have different distributions for the response,
# which makes direct comparison problematic. The AIC criterion, which is minus twice
the
# maximized likelihood plus twice the number of parameters, has often been used as a
way
```

```
# to choose between models. Smaller values are preferred. However, when computing a
# likelihood, it is common practice to discard parts that are not functions of the
parameters.
# This has no consequence when models with same distribution for the response are
# compared since the parts discarded will be equal. For responses with different
# distributions, it is essential that all parts of the likelihood be retained. (...)
# Note that purely numerical comparisons such as
# this are risky and that some attention to residual diagnostics, scientific context and
# interpretation is necessary."


# observados vs. predichos Normal vs. Gamma
par(mfrow=c(1,2))
pred_response=predict(mod_gamma,type="response")
plot(pred_response,datos$cereal, main="Gamma")
abline(a=0, b=1,col="red",lwd=4)

pred_normal=predict(mod_normal,type="response")
plot(pred_normal,datos$cereal, main="Normal")
abline(a=0, b=1,col="red",lwd=4)
```

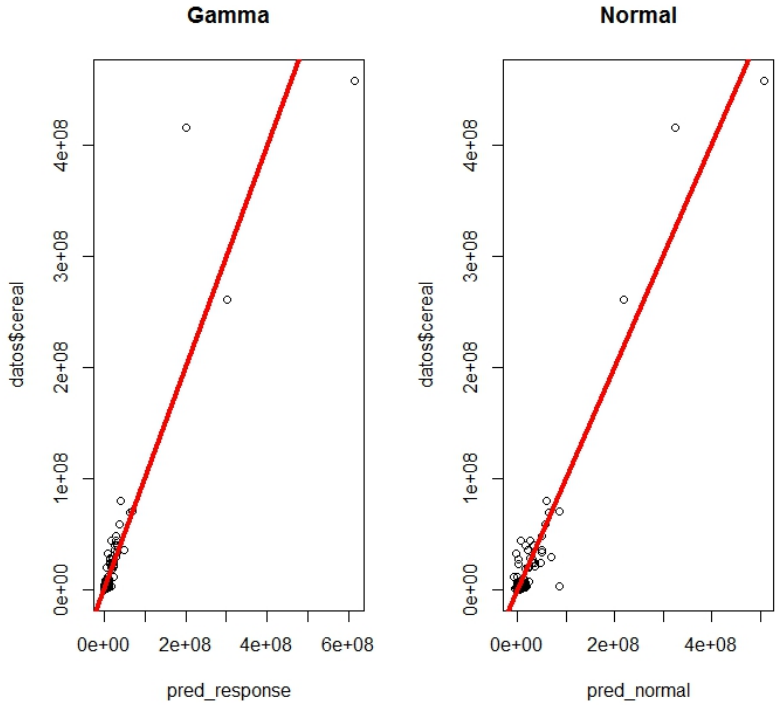


Figura 24.1.: Observados vs. predichos para el modelo ajustado a distribución normal y gamma.

```
# hay tres países que dominan la figura, sin embargo
# la muestra contiene 133 países.


# Ahora nos concentramos en los valores menores a 100000000
par(mfrow=c(1,2))
```

```
pred_response=predict(mod_gamma,type="response")
plot(pred_response,datos$cereal, main="Gamma",
    xlim=c(0,1e+08), ylim=c(0,1e+08))
abline(a=0, b=1,col="red",lwd=4)

pred_normal=predict(mod_normal,type="response")
plot(pred_normal,datos$cereal, main="Normal",
    xlim=c(0,1e+08), ylim=c(0,1e+08))
abline(a=0, b=1,col="red",lwd=4)
# se puede observar que en general los puntos están más alejados de la recta
# en el modelo con distribución normal.
```

24.2 Consignas a resolver

```
# Para el modelo con distribución Gamma:
# 1) Plantear el modelo estimado e interpretarlo.
# 2) Indique cuáles son los los coeficientes de regresión parcial

# 3) Interprete el valor de residual deviance en términos del problema.

# 4) Concluya sobre el problema de interés
# e indique la probabilidad de error asociada a sus conclusiones.

# 5) Estime la potencia del test estadístico t
# utilizado en el summary para algún efecto de interés.

# 6) ¿Existe multicolinealidad?

# 7) ¿Qué medidas tomaría desde el punto de vista del diseño del experimento
#  (es decir previo a tomar los datos) para lograr una mayor bondad de ajuste?

# 8) Plantear la función de verosimilitud

# 9) Interpretar el siguiente modelo y evaluar los supuestos
mod_gamma2=glm(cereal~capital+PEA+fertilizante + capital:PEA,
           family=Gamma (link="identity"),data=datos)
summary(mod_gamma2)
```

```
Call:
glm(formula = cereal ~ capital + PEA + fertilizante + capital:PEA,
    family = Gamma(link = "identity"), data = datos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1556  -0.6452  -0.1879   0.3582   1.7809

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -8.481e+03  1.246e+03  -6.806 3.48e-10 ***
capital       4.420e+01  3.210e+01   1.377   0.1710
PEA           2.256e+02  3.469e+01   6.504 1.60e-09 ***
```

```
fertilizante  1.208e+01  2.042e+00   5.913 2.87e-08 ***
capital:PEA   7.744e-03  3.600e-03   2.151   0.0333 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.5352344)

    Null deviance: 565.228  on 132  degrees of freedom
Residual deviance:  79.204  on 128  degrees of freedom
AIC: 4204.3

Number of Fisher Scoring iterations: 12
```

Interpretar el valor de interacción.

AIC(mod_gamma,mod_gamma2)

```
df      AIC
mod_gamma   5 4207.464
mod_gamma2  6 4204.296
```

BIC(mod_gamma,mod_gamma2)

```
  df      BIC
mod_gamma   5 4221.916
mod_gamma2  6 4221.638
```

anova(mod_gamma, mod_gamma2, test="Chisq")

```
Analysis of Deviance Table

Model 1: cereal ~ capital + PEA + fertilizante
Model 2: cereal ~ capital + PEA + fertilizante + capital:PEA
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       129     82.058
2       128     79.204  1   2.8538  0.02094 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

mod_nulo=glm(cereal~1,family=Gamma(link=identity),data=datos)
D2_gamma=(mod_nulo$deviance-mod_gamma$deviance) / mod_nulo$deviance
D2_gamma2=(mod_nulo$deviance-mod_gamma2$deviance) / mod_nulo$deviance
D2_gamma; D2_gamma2

```
[1] 0.8548226
[1] 0.8598716
```

Trabajo Práctico Nº 25: Distribución de Poisson y Binomial Negativa

25.1 Problema y Datos

```
# Datos obtenidos del libro "Kleiber & Zeileis. 2008. Applied Econometrics with R"
datos=read.table("datos_p_25.txt")
names(datos)

# Se quiere comprender cómo varía la demanda de viajes en lancha (trips)
# a partir de una serie de predictores relevantes para el mercado recreacional


# trips = número de viajes en lancha solicitados a una determinada agencia
# quality = un orden (ranking) subjetivo de la calidad de la lancha
# ski = si se contrató ski acuático o no
# income = el ingreso familiar
# userfee = indica si la persona tuvo que pagar o no un derecho de uso en el lago destino
# costC, costS, costH = costos de oportunidad
```

25.2 Distribución Poisson

```
# Como estamos queriendo modelar el número de viajes solicitados a cada agencia,
# variable cuantitativa discreta que resulta de un conteo, podríamos modelar estos
# datos utilizando una distribución poisson.

# Una ventaja en usar una distribución poisson es que la probabilidad
# para valores negativos es cero. Además al suponer que la media es igual a la varianza
# permite modelar heterogeneidad de varianzas. Sin embargo, en economía muchas
# veces se observan datos en los que la varianza es mayor que la media y este fenómeno
# se llama sobredispersión

mod_poisson=glm(trips~quality+ski+income+userfee+costC+costS+costH,
          data=datos,family=poisson(link="log"))
summary(mod_poisson)
```

```
Call:
glm(formula = trips ~ quality + ski + income + userfee + costC +
    costS + costH, family = poisson(link = "log"), data = datos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9471  -1.0149  -0.2532   0.8884   2.4638

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.8997103  0.1705453   5.275 1.32e-07 ***
quality      0.2368516  0.0356710   6.640 3.14e-11 ***
skiyes       0.3964794  0.0820709   4.831 1.36e-06 ***
income      -0.0338797  0.0246441  -1.375 0.169206
userfeeyes   0.4868726  0.1318410   3.693 0.000222 ***
costC       -0.0005908  0.0087377  -0.068 0.946090
costS       -0.0290212  0.0044563  -6.512 7.39e-11 ***
costH        0.0252301  0.0067742   3.724 0.000196 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 311.58  on 126  degrees of freedom
Residual deviance: 168.34  on 119  degrees of freedom
AIC: 611.55

Number of Fisher Scoring iterations: 4
```

```
# Estimar la potencia de alguno de los tests elegidos.
# Justificar el valor de la hipótesis alternativa elegida para estimar la potencia
# y explicar su importancia en el contexto del problema.
```

```
anova(mod_poisson, test="Chisq")
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: trips

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                      126     311.58
quality  1   21.956       125     289.62 2.790e-06 ***
ski      1    3.036       124     286.58 0.0814292 .
income   1   13.279       123     273.31 0.0002684 ***
userfee  1   20.690       122     252.62 5.399e-06 ***
costC    1   13.296       121     239.32 0.0002659 ***
costS    1   57.202       120     182.12 3.932e-14 ***
costH    1   13.781       119     168.34 0.0002054 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Evaluemos la presencia de sobredispersión
deviance(mod_poisson)/df.residual(mod_poisson)
```

```
[1] 1.414592
```

```
# Recordamos que si la especificación del modelo Poisson es correcta,
# la Devianza residual debería seguir una
# distribución de Chi^2 con los grados de libertad residuales.
# La media de la Chi^2 es igual a los grados de libertad residuales.
# Por lo tanto se "espera" que la "Residual deviance" sea igual a los grados de libertad
# residuales.

# Si la "Residual deviance" es mayor que los "residual degrees of freedom" estamos en
# presencia de sobre-dispersión. En este caso la varianza no es igual la media como
asume
# una distribución Poisson.

# La falta de cumplimiento en los supuestos se puede deber, por ejemplo, a que no
```

```
hemos
# incluido los predictores adecuados en el modelo o a que los datos no tienen
# distribución Poisson.


# También podemos evaluar la hipótesis nula que dice que
# el "Residual deviance" proviene de una distribución Chi^2 con df = grados de libertad
# residuales
pchisq(q = deviance(mod_poisson),df = df.residual(mod_poisson),lower=FALSE)
```

```
[1] 0.00198
```

```
# Plantear la hipótesis nula y la hipótesis alternativa.
# Concluir
```

25.3 Distribución binomial negativa

```
# La distribución binomial negativa es útil para describir variables cuantitativas
# discretas, por ejemplo conteos. A diferencia de la distribución binomial, ahora
sabemos
# el límite inferior (generalmente cero) pero no conocemos el límite superior.
# Es decir que no estamos trabajando con proporciones.

# La distribución binomial negativa NO es parte de la
# familia exponencial (normal, binomial, gamma, poisson, etc.)

# La distribución binomial negativa tiene dos parámetros, la media y el parámetro de
# agregación (que llamaremos "Theta").

# Theta =  media^2 / (varianza - media)
# Reordenando
# Varianza = ((media^2) / Theta) + media
# Por lo tanto cuanto mayor es Theta menor es la varianza (mayor es la agregación)

# La distribución Poisson es un caso particular de la distribución binomial,
# en el que Theta = infinito, dado que la media es igual a la varianza

# Como a Theta no se le permite tomar valores negativos, la distribución binomial
# negativa no es adecuada para situaciones en las que la varianza es menor que la
media.
# Esto es bien distinto a la distribución binomial en la que suponíamos que la varianza
# era menor que la media.
install.packages("MASS")
library(MASS)
mod_negbin<-glm.nb(trips~quality+ski+income+userfee+costC+costS+costH,
                   data=datos, link=log)
# Notar que usamos el mismo "log" link que en una Poisson

summary(mod_negbin)
```

```
Call:
glm.nb(formula = trips ~ quality + ski + income + userfee + costC +
    costS + costH, data = datos, link = log, init.theta = 14.83718619)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7072  -0.9276  -0.2249   0.7533   1.9500

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.932892   0.198322   4.704 2.55e-06 ***
quality      0.224604   0.042035   5.343 9.13e-08 ***
skiyes       0.370340   0.097950   3.781 0.000156 ***
income      -0.030309   0.028859  -1.050 0.293609
userfeeyes   0.458847   0.168930   2.716 0.006604 **
costC        0.001522   0.010513   0.145 0.884874
costS       -0.028385   0.005305  -5.350 8.78e-08 ***
costH        0.022798   0.008009   2.846 0.004421 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(14.8372) family taken to be 1)

    Null deviance: 221.14  on 126  degrees of freedom
Residual deviance: 122.23  on 119  degrees of freedom
AIC: 605.87
Number of Fisher Scoring iterations: 1


              Theta:  14.84
          Std. Err.:  6.75

 2 x log-likelihood:  -587.868
```

```
# El "Std. Err.:  6.75" es el error estándar asociado a Theta.
# Explicar qué significa este error estándar y cuál es su utilidad
# en el contexto del problema.


# Comparar e interpretar:
deviance(mod_poisson)/df.residual(mod_poisson)
```

```
[1] 1.414592
```

```
deviance(mod_negbin)/df.residual(mod_negbin)
```

```
[1] 1.027125
```

```
# ¿Cuántos viajes realizarán en promedio familias con un ingreso de
# 5, que califica a su lancha con un 3, que no le interesa el ski acuático,
# que no tuvo que pagar derecho de uso en el lago, dados los costos de
# oportunidad promedios observados en la base de datos?

# ¿Cuánto es la variabilidad entre esas familias?
```

```
# ¿Cuáles son las unidades del estadístico elegido?
```

25.4 Varianza en función de la media

```
# grafiquemos cómo cambia la varianza con la media según el modelo estimado
par(mfrow=c(1,2))
plot(datos$trips,datos$trips,xlab="Media", ylab="Varianza",main="Poisson")
plot(datos$trips,((datos$trips^2/14.84)+datos$trips),
    xlab="Media", ylab="Varianza",main="Binomial negativa")
```

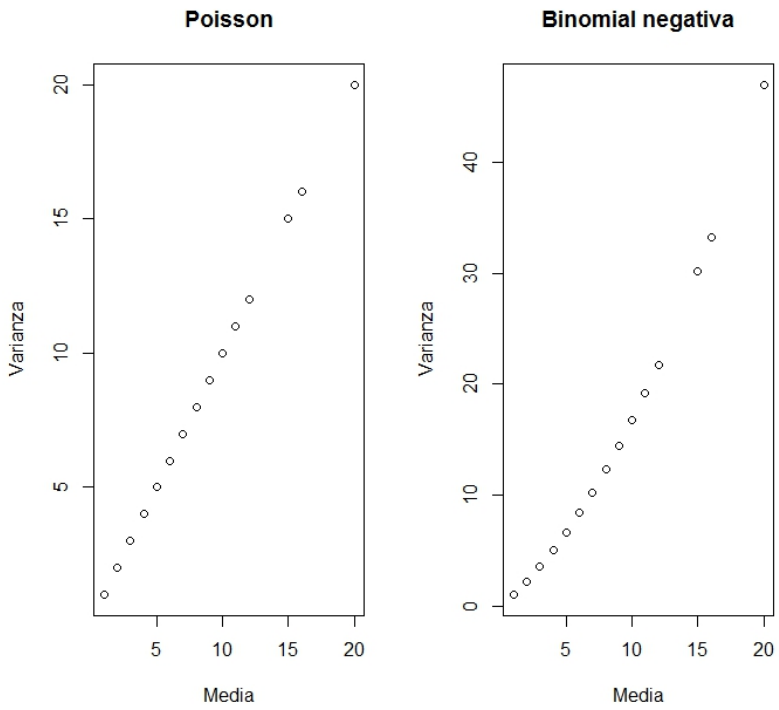


Figura 25.1.: Evolución de la varianza en función de la media para distribuciones Poisson y Binomial negativa.

```
anova(mod_negbin,test="Chisq")
```

```
Analysis of Deviance Table

Model: Negative Binomial(14.8372), link: log

Response: trips

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                      126     221.14
quality  1   16.128       125     205.01 5.920e-05 ***
ski      1    2.352       124     202.66 0.1251293
income   1    9.802       123     192.85 0.0017430 **
userfee  1   14.233       122     178.62 0.0001615 ***
costC    1    8.615       121     170.01 0.0033347 **
costS    1   39.621       120     130.39 3.084e-10 ***
costH    1    8.158       119     122.23 0.0042864 **
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Plantear hipótesis e interpretar


# ¿Que significa el valor esperado de una variable aleatoria?


# para los coeficientes de regresión parcial significativamente distintos de cero:
# los valores estimados por el modelo poisson y por el modelo negbinom son parecidos
coef(mod_poisson); coef(mod_negbin)
```

25.5 GLM y Binomial negativa

```
# La función glm puede ajustar una binomial negativa...
# pero... probar...
mod_negbin_B<-glm(trips~quality+ski+income+userfee+costC+costS+costH,
                  data=datos, family=negative.binomial(link="log"))
# ...le tenemos que indicar el valor estimado de "theta".
# Demos el mismo valor que observamos arriba cuando estimamos el modelo
# con la función glm.nb...
mod_negbin_B<-glm(trips~quality+ski+income+userfee+costC+costS+costH,
                  data=datos, family=negative.binomial(theta=14.84, link="log"))
# ...y ahora corre... comparemos:
round(coef(mod_poisson), digits =3); round(coef(mod_negbin), digits = 3);
round(coef(mod_negbin_B), digits = 3)

# ¿Que observan?

# Si ponemos otro theta...
mod_negbin_C<-glm(trips~quality+ski+income+userfee+costC+costS+costH,
                  data=datos, family=negative.binomial(theta=2, link="log"))
# ... naturalmente da valores distintos para los coeficientes parciales:
round(coef(mod_poisson), digits =3); round(coef(mod_negbin), digits = 3);
round(coef(mod_negbin_B), digits = 3); round(coef(mod_negbin_C), digits = 3)


# Concluir en función de:
AIC(mod_negbin,mod_negbin_B, mod_negbin_C)
```

```
             df      AIC
mod_negbin    9 605.8684
mod_negbin_B  8 603.8684
mod_negbin_C  8 651.2831
```

```
# ¿Cuál es el modelo con mejor bondad de ajuste?

# ACLARACIÓN: "mod_negbin" da un valor ligeramente distinto a "mod_negbin_B"
# porque la función glm.nb estima un parámetro más (theta),
# mientras que a glm le damos el valor estimado de theta.
# Entonces, la verosimilitud es la misma en ambos modelos:
c(logLik(mod_negbin), logLik(mod_negbin_B))
```

```
[1] -293.9342 -293.9342

# lo que cambia es el número de parámetros y por lo tanto el AIC
c(AIC(mod_negbin), -2*logLik(mod_negbin)+2*9)

[1] 605.8684 605.8684

# distinto a:
c(AIC(mod_negbin_B), -2*logLik(mod_negbin)+2*8)

[1] 603.8684 603.8684
# Por lo tanto no es muy útil utilizar la función glm para estimar
# modelos con distribución binomial negativa ya que tenemos que darle
# nosotros el valor estimado de "theta".

# OJO que glm estima binomial negativa siempre y cuando
# tengan el paquete MASS instalado el cual incopora la función binomial negativa
# Probar:
detach("package:MASS")
mod_negbin_B<-glm(trips~quality+ski+income+userfee+costC+costS+costH,
                  data=datos, family=negative.binomial(theta=14.84, link="log"))
# En cambio:
library(MASS)
mod_negbin_B<-glm(trips~quality+ski+income+userfee+costC+costS+costH,
                  data=datos, family=negative.binomial(theta=14.84, link="log"))


######## ESCALA VARIABLE RESPUESTA ########
# mu(i)=exp (B0 + B1*quality(i) + B2*skiyes(i) + B3*income(i) + B4*userfeeyes(i) +
#           + B5*costC(i) + B6*costS(i) + B7*costH(i) )

# POISSON
par(mfrow=c(1,2))
plot(predict(mod_poisson, type="response"),datos$trips,
    ylab="viajes observados (núm)", xlab="viajes predichos (núm)",
    main="Poisson")
abline(a=0,b=1, col="green", lwd=3)

plot(predict(mod_negbin, type="response"),datos$trips,
    ylab="viajes observados (núm)", xlab="viajes predichos (núm)",
    main="Binomial negativa")
abline(a=0,b=1, col="green", lwd=3)

# ESCALA LOG
# el glm estimó los parámetros de un modelo lineal en escala log por el método de
# máxima verosimilitud
# log(mu(i))= B0 + B1*quality(i) + B2*skiyes(i) + B3*income(i) + B4*userfeeyes(i) +
#           + B5*costC(i) + B6*costS(i) + B7*costH(i)
par(mfrow=c(1,2))
plot(predict(mod_poisson, type="link"),log(datos$trips),
```

```
    ylab="viajes observados (ln núm)", xlab="viajes predichos (ln núm)",
    main="Poisson")
abline(a=0,b=1, col="green", lwd=3)

plot(predict(mod_negbin, type="link"),log(datos$trips),
    ylab="viajes observados (ln núm)", xlab="viajes predichos (ln núm)",
    main="Binomial negativa")
abline(a=0,b=1, col="green", lwd=3)
```

25.6 Supuestos

```
# residuos vs. predichos
par(mfrow=c(1,2))
plot(predict(mod_poisson, type="link"),resid(mod_poisson, type="deviance"),
    ylab="residuos (ln núm viajes)", xlab="viajes predichos (ln núm)",
    main="Poisson")
abline(a=0,b=0, col="green", lwd=3)

plot(predict(mod_negbin, type="link"),resid(mod_negbin, type="deviance"),
    ylab="residuos (ln núm viajes)", xlab="viajes predichos (ln núm)",
    main="Binomial negativa")
abline(a=0,b=0, col="green", lwd=3)


res_poisson=resid(mod_poisson, type="deviance")
hist(res_poisson)
qqnorm(res_poisson)
qqline(res_poisson)
library(moments)
kurtosis(res_poisson)
skewness(res_poisson)


res_negbin=resid(mod_negbin, type="deviance")
hist(res_negbin)
qqnorm(res_negbin)
qqline(res_negbin)
library(moments)
kurtosis(res_negbin)
skewness(res_negbin)

# Plantear la función de verosimilitud para los dos modelos evaluados
```

25.6.1 Poisson

```
# A medida que aumenta lambda (= media = varianza) la distribución poisson se
# asemeja a una normal
x<-seq(from=1,to=40,by=1)
par(mfrow=c(3,3))
y<-dpois(x,lambda=0.5)
plot(x,y,type="h")
```

```
y<-dpois(x,lambda=1)
plot(x,y,type="h")
y<-dpois(x,lambda=2)
plot(x,y,type="h")
y<-dpois(x,lambda=4)
plot(x,y,type="h")
y<-dpois(x,lambda=6)
plot(x,y,type="h")
y<-dpois(x,lambda=8)
plot(x,y,type="h")
y<-dpois(x,lambda=10)
plot(x,y,type="h")
y<-dpois(x,lambda=15)
plot(x,y,type="h")
y<-dpois(x,lambda=20)
plot(x,y,type="h")
```

25.6.1 Binomial Negativa

```
# size = theta
x<-seq(from=1,to=150,by=1)
par(mfrow=c(3,3))
y<-dnbinom(x,mu=1,size=0.1)
plot(x,y,type="h", main="media=1 theta=0.1")
y<-dnbinom(x,mu=1, size=1)
plot(x,y,type="h", main="media=1 theta=1")
y<-dnbinom(x,mu=1, size=100000)
plot(x,y,type="h", main="media=1 theta=100000")
y<-dnbinom(x,mu=10, size=0.1)
plot(x,y,type="h", main="media=10 theta=0.1")
y<-dnbinom(x,mu=10, size=1)
plot(x,y,type="h", main="media=10 theta=1")
y<-dnbinom(x,mu=10, size=100000)
plot(x,y,type="h", main="media=10 theta=100000")
y<-dnbinom(x,mu=100, size=0.1)
plot(x,y,type="h", main="media=100 theta=0.1")
y<-dnbinom(x,mu=100, size=1)
plot(x,y,type="h", main="media=100 theta=1")
y<-dnbinom(x,mu=100, size=100000)
plot(x,y,type="h", main="media=100 theta=100000")
```

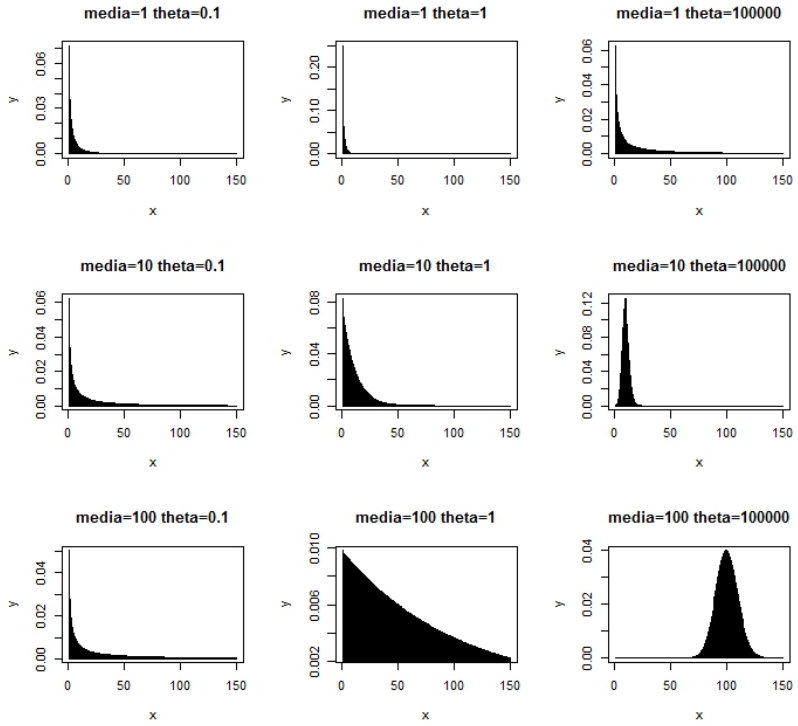


Figura 25.2.:Función de densidad de la distribución binomial negativa para diferentes valores de la media y theta.

cuando theta es = 1 (los tres paneles del medio) la binomial negativa
también es conocida como la distribución geométrica.

Cuanto mayor es la media Y el theta, la binomial negativa se parece a una normal

Cuanto mayor es theta la binomial negativa se parece a una distribución Poisson.

Trabajo Práctico Nº 26: Ejercicios varios

26.1 Problema y Datos

Datos del INDEC
http://www.indec.gov.ar/

```
datos=read.table("datos_p_26.txt", header=T, dec=",")
str(datos)
```

26.2 Primer ejercicio

Se desea comprender cómo varía la cantidad de hijos menores de un hogar en función de:

#---prom_niveled: el nivel educativo promedio de los habitantes mayores de edad de la casa.

#---Empleo: la situación laboral de la casa
.-empleado

```
#                 .-desempleado (no tiene trabajo pero esta buscando)
#                 .-inactivo (no tiene trabajo pero tampoco esta buscando ya sea porque
tiene capitales invertidos u otros motivos)

#---ITF: ingreso total familiar

#---Region


summary(datos)

#   Para los objetivos particulares de este problema nos interesa juntar a los
# desempleados y a los inactivos
for (i in 1:length(datos$Cant_menores)) {
  datos$Empleo_2[i]=if(datos$Empleo[i]=="emp") {"si"}  else {"no"}
}
datos$Empleo_2 =  factor(datos$Empleo_2)


View(datos)
str(datos)
summary(datos)

# ¿Cuál es el tamaño de la muestra?
# ¿Cuál es la unidad experimental, la muestra y la población?

# ¿Es un DCA o un DBCA?

# Si un experimento es mensurativo, ¿podría ser un DBCA?

# ¿Cuántos hogares de los encuestados no tuvieron hijos?
length(subset(datos$Cant_menores, datos$Cant_menores==0))

[1] 270

# ¿Qué porcentaje representan del total?
length(subset(datos$Cant_menores, datos$Cant_menores==0)) /
length(datos$Cant_menores) * 100

[1] 47.61905

# Casi la mitad de los hogares en Argentina no tienen hijos... interesante...

# Alternativamente
par(mfrow=c(1,1))
hist(datos$Cant_menores)
hist(datos$Cant_menores, right= FALSE)
# ¿Cuál es la diferencia entre ambos histogramas?

summary(datos)
```

```
# Antes de lanzarse a estimar un modelo:
# 1) ideas claras (marco conceptual) y 2) explorar datos
# por ejemplo:
par(mfrow=c(2,2))
plot(datos$prom_niveled,datos$Cant_menores, xlab="Educación (años)", ylab="Hijos
(número)")
plot(datos$Empleo_2,datos$Cant_menores, xlab="Empleo", ylab="Hijos (número)")
plot(datos$ITF,datos$Cant_menores, xlab="Ingreso total familiar ($)", ylab="Hijos
(número)")
plot(datos$Region,datos$Cant_menores, xlab="Región", ylab="Hijos (número)")
# Interpretar cada uno de los gráficos en términos del problema
```

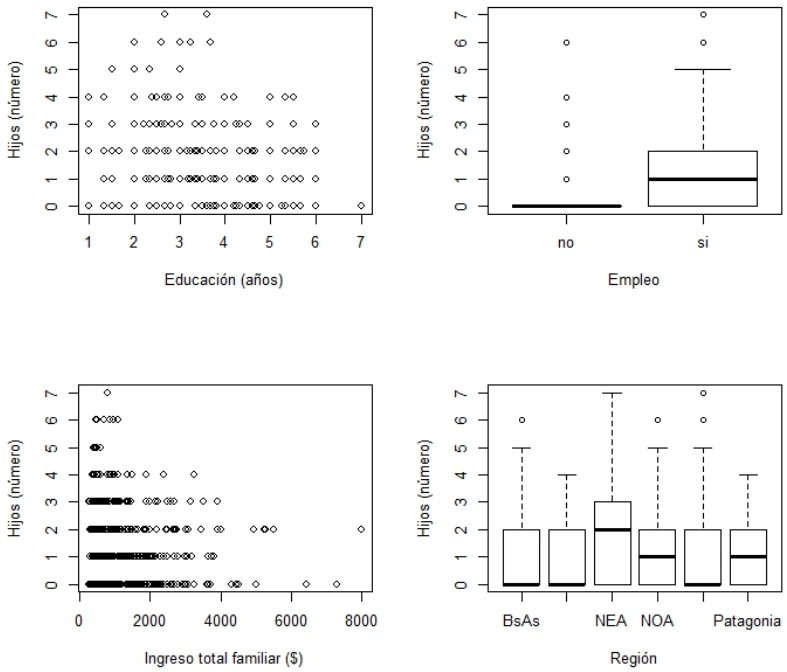


Figura 26.1.: Exploración general de los datos. Cantidad de hijos en función de años de educación, cantidad de hijos en función de empleado o no, cantidad de hijos en función de ingreso total familiar y cantidad de hijos según región geográfica.

```
# Poisson
mod_poisson=glm(Cant_menores~prom_niveled+Empleo_2+ITF+Region,
          data=datos,family=poisson(link="log"))
summary(mod_poisson)

Call:
glm(formula = Cant_menores ~ prom_niveled + Empleo_2 + ITF +
    Region, family = poisson(link = "log"), data = datos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3312  -1.4086  -0.5288   0.6337   4.9737

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -9.390e-01  2.226e-01  -4.219 2.46e-05 ***
```

```
prom_niveled    -1.324e-01  3.268e-02  -4.051 5.10e-05 ***
Empleo_2si       1.678e+00  1.990e-01   8.436  < 2e-16 ***
ITF             -3.362e-05  4.523e-05  -0.743  0.45735
RegionCuyo       3.014e-02  1.712e-01   0.176  0.86022
RegionNEA        4.043e-01  1.349e-01   2.998  0.00272 **
RegionNOA        1.598e-01  1.219e-01   1.310  0.19007
RegionPampeana  -9.167e-02  1.170e-01  -0.784  0.43329
RegionPatagonia  1.440e-01  1.662e-01   0.867  0.38606
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1030.5  on 566  degrees of freedom
Residual deviance:  870.1  on 558  degrees of freedom
AIC: 1655.8

Number of Fisher Scoring iterations: 6
```

```
anova(mod_poisson, test="Chisq")
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: Cant_menores

Terms added sequentially (first to last)


             Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                          566    1030.53
prom_niveled  1   11.303       565    1019.22 0.0007737 ***
Empleo_2      1  130.624       564     888.60 < 2.2e-16 ***
ITF           1    1.592       563     887.01 0.2069761
Region        5   16.906       558     870.10 0.0046818 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# ¿La varianza es más grande que la media?
deviance(mod_poisson)/df.residual(mod_poisson)
pchisq(q = deviance(mod_poisson),df = df.residual(mod_poisson),lower=FALSE)
# Plantear H0) y H1)

# Entonces...
#install.packages("MASS")
library(MASS)
mod_negbin<-glm.nb(Cant_menores~prom_niveled+Empleo+ITF+Region,
                   data=datos, link=log)
summary(mod_negbin)
# Obtener el "valor-p" que se observa en el summary para el efecto del nivel
# promedio de educación utilizando la función pnorm


# Comparar e interpretar:
```

```
deviance(mod_poisson)/df.residual(mod_poisson)

[1] 1.559319
deviance(mod_negbin)/df.residual(mod_negbin)

[1] 1.027125


# Interpretar los siguientes intervalos de confianza
confint(mod_negbin, level=0.99)

# Interpretar
predict.lm(mod_negbin,interval="confidence")
predict.lm(mod_negbin,interval="prediction")

# Interpretar el siguiente análisis de devianza
anova(mod_negbin,test="Chisq")

Analysis of Deviance Table

Model: Negative Binomial(14.8372), link: log

Response: trips

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                      126     221.14
quality  1   16.128       125     205.01 5.920e-05 ***
ski      1    2.352       124     202.66 0.1251293
income   1    9.802       123     192.85 0.0017430 **
userfee  1   14.233       122     178.62 0.0001615 ***
costC    1    8.615       121     170.01 0.0033347 **
costS    1   39.621       120     130.39 3.084e-10 ***
costH    1    8.158       119     122.23 0.0042864 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Explicar la diferencia entre este análisis de devianza y un análisis de
# varianza con SC-Secuencial y un análisis de varianza con SC-Parcial.
# ¿Cuáles son las diferencias entre el estadístico F y el
# cociente de verosimilitudes?

# Plantear la función de verosimilitud para los dos modelos evaluados

# Se propone que el efecto del nivel de educación varía según el empleo.
# En particular, se propone que en hogares con desempleo el efecto de la educación
sobre
# la cantidad de hijos es mayor.
# Realice las modificaciones que considere pertinentes en el modelo para evaluar este
# aspecto y concluya.
# ¿Que es la multicolinealidad y cómo afecta la estimación de los parámetros?
```

```
# ¿Sucede en este caso?

# ¿Cuál es la diferencia entre una pendiente y un coeficiente de regresión parcial?
# Concluya respecto las ideas que motivaron este análisis.
# ¿Cuál es la probabilidad de error asociada a su conclusión?


# ¿Cuántos hijos menores "esperaría" para un hogar en el NEA con empleo,
# con 6 años de educación promedio y 4870$ de ITF? ¿Cuál es la varianza en este caso?
# ¿Qué utilidad tiene conocer la varianza?

# ¿El AIC es una medida de bondad de ajuste? ¿Por qué?
# Desde el punto de vista del diseño experimental:
# ¿Qué medidas tomaría para disminuir el AIC?

# Evalúe los supuestos del modelo elegido.

# Aquí observamos datos para los cuáles la binomial negativa (o Poisson) no es un
modelo
# adecuado.

# Del gráfico cuantil-cuantil y de los histogramas se observa que los datos tienen
# "más ceros" de lo que se espera según una Poisson o una binomial negativa.
hist(datos$Cant_menores)
# Las distribuciones que habitualmente se utilizan en estos datos se
# llaman distribuciones "infladas en cero"
# Para otro curso... porque este se acaba... :-).... ver
install.packages("pscl")
library("pscl")
?zeroinfl
# zero-inflated Poisson (ZIP)
# zero-inflated negative binomial (ZINB)
# zero-altered Poisson (ZAP)
# zero-altered negative binomial (ZANB)

# ZIP - ZAP.... estudien para el examen... :-)
```

26.3 Segundo ejercicio

```
### Cambiando de tema pero con los mismos datos.
### Evalúen si el empleo difiere significativamente según regiones.

# Interpretar el siguiente gráfico, plantear un modelo acorde y evaluar sus supuestos
par(mfrow=c(1,1))
plot(datos$Region,datos$Empleo_2, ylab="Empleo", xlab="Región")

# Ayuda
for (i in 1:length(datos$Cant_menores)) {
  datos$Empleo_num[i]=if(datos$Empleo[i]=="emp") {"1"}  else {"0"}
}
```

```
datos$Empleo_num=as.numeric(datos$Empleo_num)


str(datos)
View(datos)
```

26.4 Armar propio GLM

```
# "Fox y Weisberg 2011 An R Companion to Applied Regression"

# To supplement the flexibility provided by the standard and quasi
# family generators, it is also possible to add family generators,
# link functions, and variance functions to R (assuming, of course,
# that you have the necessary statistial knowledge and
# programming prowess). In the previous section, for example,
# we used the family generator negative.binomial, added
# by Venables and Ripley's MASS package, to fit a
# negative binomial GLM.
```

Trabajo Práctico Nº 27: Ejemplos de diferentes distribuciones y sus relaciones varianza vs. media

27.1 Distribuciones

27.1.1 Binomial

```
# Unidad experimental = cada pueblo de Argentina
# Muestra = los pueblos evaluados de Argentina
# Población = todos los pueblos de Argentina

# En cada pueblo se encuestan 29 personas y se les pregunta si votarán al candidato
"liberal" o no
# (no votarlo incluye voto en blanco)
# Proponemos que    Yi~Binom(p, N')
# donde Yi es el número de personas que votan al candidato "liberal" en el pueblo i
#       p es la probabilidad individual de votar al candidato "liberal"
#       N' es 29 en este caso, utilizamos N' para diferenciarlo de N el tamaño poblacional

espacio_muestral=seq(from=0, to=29, by=1)
espacio_muestral
# el espacio muestral son los resultados posibles para cada unidad experimental luego
de
# cada experimento aleatorio.

# supongamos que la probabilidad individual de votar al candidato "liberal" es 0.9
# Entonces la distribución de datos será:
resultado = dbinom(espacio_muestral, size=29, prob=0.9)
resultado
```

```
# corroboramos que la suma de las probabilidades sea igual a 1
sum(resultado)
# vamos bien, veamos el histograma
barplot(resultado, names.arg=espacio_muestral,
        ylab="Probabilidad", xlab="Personas que votan al liberal")
```

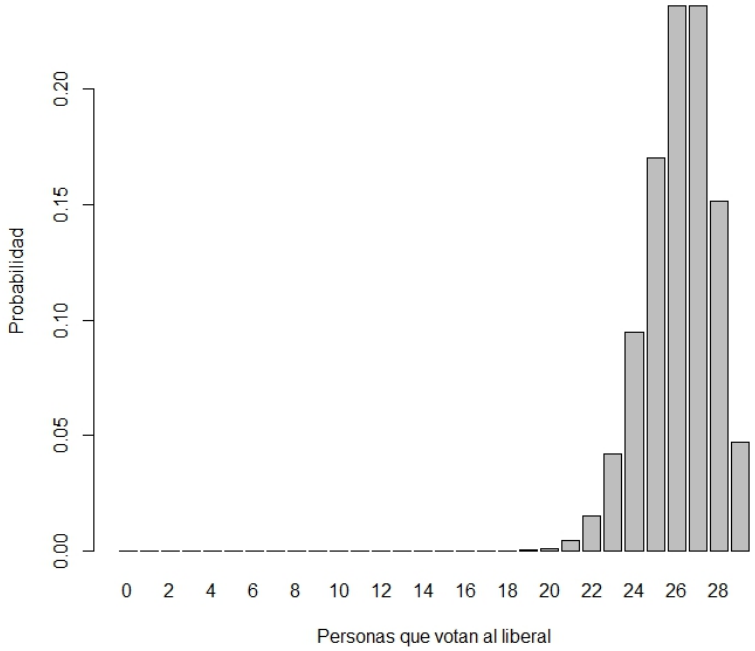


Figura 27.1.: Distribución de probabilidad de la cantidad de personas que votan al candidato liberal, de una muestra de 29 personas y una probabilidad individual de 0,9 de votar al candidato liberal.

```
# dada esta distribución. ¿Cuál es la probabilidad de que en un pueblo elegido al azar
# se observen hasta 2 personas que voten al candidato liberal?
pbinom(2,size=29,prob=0.9)
```

```
[1] 3.3148e-25
```

```
# ¿Y hasta 26 personas?
pbinom(26,size=29,prob=0.9)
```

```
[1] 0.56504
```

```
# ¿Exactamente 26 personas?
a = pbinom(25,size=29,prob=0.9)
b = pbinom(26,size=29,prob=0.9)
b-a
```

```
[1] 0.2360879
```

```
# ¿Más de 26 personas?
```

1-b

```
[1] 0.43496
```

```
# la media de la distribución
# N' * p
29*0.9
```

```
[1] 26.1
```

```
# otra manera
sum(resultado*espacio_muestral)


# Veamos cómo cambia la distribución para valores crecientes de probabilidad
individual
par(mfrow=c(3,3))
resultado = dbinom(espacio_muestral, size=29, prob=0.01)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.01")
resultado = dbinom(espacio_muestral, size=29, prob=0.05)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.05")
resultado = dbinom(espacio_muestral, size=29, prob=0.10)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.10")
resultado = dbinom(espacio_muestral, size=29, prob=0.25)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.25")
resultado = dbinom(espacio_muestral, size=29, prob=0.50)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.50")
resultado = dbinom(espacio_muestral, size=29, prob=0.75)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.75")
resultado = dbinom(espacio_muestral, size=29, prob=0.90)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.90")
resultado = dbinom(espacio_muestral, size=29, prob=0.95)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.95")
resultado = dbinom(espacio_muestral, size=29, prob=0.99)
barplot(resultado, names.arg=espacio_muestral,
      ylab="Probabilidad", xlab="Personas que votan al liberal", main="p=0.99")
# Es evidente que cambios en p implican cambios en la media y en la varianza de la
distribución.
# Recuerden:
# media = N' * p
# varianza = N' * p * q

# En sus casas realicen gráficos similares modificando N'
```

```
# Antes de continuar discutamos cómo expandir este modelo simple para reflejar que
"p"
# varía entre pueblos y que esa variación se debe en parte al ingreso promedio que
presentan las personas
# en ese pueblo.
```

27.1.2 Gamma

```
# La distribución Gamma es interesante para modelar un amplio rango de procesos
# en los que:
# 1) la variable respuesta es cuantitativa contínua.
# 2) la variable respuesta toma valores mayores a cero.
# 3) la distribución de la variable respuesta es asimétrica positiva
# (con una cola larga hacia la derecha, es decir hacia valores grandes)
# 4) la varianza aumenta más que linealmente con la media
# La gamma presenta CV (coeficiente de variación) constante e igual a
sqrt("shape")...sigan leyendo para entender.
# 5) la distribución de la variable respuesta puede tomar distintos valores de curtosis

# Aquí seguimos la parametrización de Gamma según Faraway 2006, que es la utilizada
en
# la función glm (ojo que dgamma usa una parametrización ligeramente distinta).
# Entonces tenemos dos parámetros, la media y la "forma (shape)".
# Varianza = media^2 / forma
# Si asumimos un valor de "forma" constante. La relación entre varianza y media es:
curve (x^2 / 7)

# con otro valor de forma:
curve (x^2 / 3)


# Algunas distribuciones gamma con distinta forma:
# para valores grandes de "forma(shape)" la distribución gamma se vuelve simétrica
# y con forma de campana. En estos casos una distribución normal podría ser utilizada
# para modelar nuestros datos.
x<-seq(from=0.01,to=120,by=.01)
y=matrix(ncol=27, nrow=length(x))
par(mfrow=c(3,3))
y[,1]<-dgamma(x,shape=0.5)
plot(x,y[,1],type="l")
y[,2]<-dgamma(x,shape=1)
plot(x,y[,2],type="l")
y[,3]<-dgamma(x,shape=2)
plot(x,y[,3],type="l")
y[,4]<-dgamma(x,shape=3)
plot(x,y[,4],type="l")
y[,5]<-dgamma(x,shape=4)
plot(x,y[,5],type="l")
y[,6]<-dgamma(x,shape=5)
```

```
plot(x,y[,6],type="l")
y[,7]<-dgamma(x,shape=6)
plot(x,y[,7],type="l")
y[,8]<-dgamma(x,shape=7)
plot(x,y[,8],type="l")
y[,9]<-dgamma(x,shape=15)
plot(x,y[,9],type="l")
# la distribución gamma no acepta y=0.

# ¿Que unidades presenta el eje y?

library(moments)
kurtosis(y[,1:9])
# a medida que aumenta el shape también disminuye la curtosis.

skewness(y[,1:9])
# siempre asimetría positiva que disminuye a mayor shape

# La distribución Gamma no puede ser asimétrica negativa.
# probemos aumentando aún más el shape
par(mfrow=c(3,3))
y[,10]<-dgamma(x,shape=10)
plot(x,y[,10],type="l")
y[,11]<-dgamma(x,shape=20)
plot(x,y[,11],type="l")
y[,12]<-dgamma(x,shape=30)
plot(x,y[,12],type="l")
y[,13]<-dgamma(x,shape=40)
plot(x,y[,13],type="l")
y[,14]<-dgamma(x,shape=50)
plot(x,y[,14],type="l")
y[,15]<-dgamma(x,shape=60)
plot(x,y[,15],type="l")
y[,16]<-dgamma(x,shape=70)
plot(x,y[,16],type="l")
y[,17]<-dgamma(x,shape=80)
plot(x,y[,17],type="l")
y[,18]<-dgamma(x,shape=90)
plot(x,y[,18],type="l")

kurtosis(y[,1:18])
# a medida que aumenta el shape también disminuye la curtosis.

skewness(y[,1:18])
# siempre asimetría positiva
```

27.1.3 Chi^2

```
# La CHI 2 surge de sumas de normales estándar (z) al cuadrado
# y tiene un único parámetro = grados de libertad
```

```
# Chi^2 con 1 gl = z^2
# Chi^2 con 2 gl = z^2 + z^2
# Chi^2 con 3 gl = z^2 + z^2 + z^2
# Chi^2 con 4 gl = z^2 + z^2 + z^2 + z^2

chi_1 = rnorm(n=9999, mean=0, sd=1)^2
chi_2 = rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2
chi_3 = rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2 +
rnorm(n=9999, mean=0, sd=1)^2
chi_4 = rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2 +
rnorm(n=9999, mean=0, sd=1)^2 + rnorm(n=9999, mean=0, sd=1)^2


# Comparemos gráficamente con la distribución chi
par(mfcol=c(2,2))
hist(chi_1, freq=F)
lines(density(chi_1), col="blue")
# es igual a la distribución chi con grados de libertad=1
lines(density(rchisq(n=9999,df=1)), col="red")

hist(chi_2, freq=F)
lines(density(chi_2), col="blue")
# es igual a la distribución chi con grados de libertad=2
lines(density(rchisq(n=9999,df=2)), col="red")

hist(chi_3, freq=F)
lines(density(chi_3), col="blue")
# es igual a la distribución chi con grados de libertad=3
lines(density(rchisq(n=9999,df=3)), col="red")

hist(chi_4, freq=F)
lines(density(chi_4), col="blue")
# es igual a la distribución chi con grados de libertad=4
lines(density(rchisq(n=9999,df=4)), col="red")


# La media de la Chi^2 es igual a los grados de libertad (el único parámetro de la
distribución)
# Veamos, para un grado de libertad
mean(rchisq(n=999999, df=1))
# para dos grados de libertad
mean(rchisq(n=999999, df=2))
# para tres grados de libertad
mean(rchisq(n=999999, df=3))
# y ya que R hace el trabajo, veamos para cuatro grados de libertad también
mean(rchisq(n=999999, df=4))
# Para más de 10 gl y redondeando:
round(mean(rchisq(999999,df=10)))
round(mean(rchisq(999999,df=15)))
```

```
round(mean(rchisq(999999,df=30)))
round(mean(rchisq(999999,df=60)))
round(mean(rchisq(999999,df=1000)))
# La distribución Chi2 es un caso particular de la Gamma. Mientras que la Gamma
# tiene dos parámetros, la Chi2 tiene un solo parámetros (i.e. los grados de libertad).

# Según la parametrización de dgamma, la Chi2 tiene un rate constante igual a 0.5,
mientras
# que el shape son los grados de libertad dividido 2
# (es decir, la media dividido 2).
# Si
# Media = shape / rate
# Media = shape / 0.5
# Media = shape * 2
# Media / 2 = shape

# Según la parametrización de dgamma, la Chi2 tiene un rate constante igual a 0.5,
mientras
# que el shape son los grados de libertad dividido 2.
# Veamos cómo cambia la distribución de Chi2 para distintos grados de libertad:
x<-seq(from=0.01,to=80,by=.01)
y=matrix(ncol=9, nrow=length(x))
par(mfrow=c(3,3))
y[,1]<-dgamma(x,shape=1/2,rate=0.5)
plot(x,y[,1],type="l")
y[,2]<-dgamma(x,shape=5/2,rate=0.5)
plot(x,y[,2],type="l")
y[,3]<-dgamma(x,shape=10/2,rate=0.5)
plot(x,y[,3],type="l")
y[,4]<-dgamma(x,shape=15/2,rate=0.5)
plot(x,y[,4],type="l")
y[,5]<-dgamma(x,shape=20/2,rate=0.5)
plot(x,y[,5],type="l")
y[,6]<-dgamma(x,shape=25/2,rate=0.5)
plot(x,y[,6],type="l")
y[,7]<-dgamma(x,shape=30/2,rate=0.5)
plot(x,y[,7],type="l")
y[,8]<-dgamma(x,shape=35/2,rate=0.5)
plot(x,y[,8],type="l")
y[,9]<-dgamma(x,shape=40/2,rate=0.5)
plot(x,y[,9],type="l")


# Por ejemplo, obtenemos 10000 valores aleatorios de una distribución
# Chi2 con df= 20
a=rchisq(10000,df =20)
a
# Nos da lo mismo que obtener 10000 valores de una distribución
# gamma con shape = 20/2 y rate=0.5
b=rgamma(10000,shape = 20/2, rate=0.5)
```

```
b
c(mean(a),mean(b))

par(mfrow=c(2,1))
hist(a, main="chisq")
hist(b, main="gamma")
```

27.1.4 Distribución Poisson

```
# A medida que aumenta lambda (= media = varianza) la distribución poisson se
# asemeja a una normal
x<-seq(from=1,to=40,by=1)
par(mfrow=c(3,3))
y<-dpois(x,lambda=0.5)
plot(x,y,type="h")
y<-dpois(x,lambda=1)
plot(x,y,type="h")
y<-dpois(x,lambda=2)
plot(x,y,type="h")
y<-dpois(x,lambda=4)
plot(x,y,type="h")
y<-dpois(x,lambda=6)
plot(x,y,type="h")
y<-dpois(x,lambda=8)
plot(x,y,type="h")
y<-dpois(x,lambda=10)
plot(x,y,type="h")
y<-dpois(x,lambda=15)
plot(x,y,type="h")
y<-dpois(x,lambda=20)
plot(x,y,type="h")
```

27.1.5 Distribución binomial negativa

```
# size = theta
x<-seq(from=1,to=150,by=1)
par(mfrow=c(3,3))
y<-dnbinom(x,mu=1,size=0.1)
plot(x,y,type="h", main="media=1 theta=0.1")
y<-dnbinom(x,mu=1, size=1)
plot(x,y,type="h", main="media=1 theta=1")
y<-dnbinom(x,mu=1, size=100000)
plot(x,y,type="h", main="media=1 theta=100000")
y<-dnbinom(x,mu=10, size=0.1)
plot(x,y,type="h", main="media=10 theta=0.1")
y<-dnbinom(x,mu=10, size=1)
plot(x,y,type="h", main="media=10 theta=1")
y<-dnbinom(x,mu=10, size=100000)
plot(x,y,type="h", main="media=10 theta=100000")
y<-dnbinom(x,mu=100, size=0.1)
```

```
plot(x,y,type="h", main="media=100 theta=0.1")
y<-dnbinom(x,mu=100, size=1)
plot(x,y,type="h", main="media=100 theta=1")
y<-dnbinom(x,mu=100, size=100000)
plot(x,y,type="h", main="media=100 theta=100000")
```

27.2 Relación varianza vs. media para las distribuciones

```
x=c(1:100)
par(mfrow=c(2,3))

# Binomial
# Varianza = n . p . q
p=seq(from = 0, to = 1, by = 0.01)
q= 1 - p
plot(100*p,100*p*q, type = "l", col = "orange", lwd = 5,
    xlab="Media", ylab="Varianza",main="Binomial")
text(x=17, y=24, "Var < Media", cex=1.2)

# Poisson
# Varianza = media
plot(x,x, type = "l", col = "orange", lwd = 5,
    xlab="Media", ylab="Varianza",main="Poisson")
text(x=18, y=95, "Var = Media", cex=1.2)

# Binomial negativa
# Varianza = ((media^2) / Theta) + media
plot(x,((x^2/10)+x), type = "l", col = "orange", lwd = 5,
    xlab="Media", ylab="Varianza",main="Binomial negativa")
text(x=19, y=1000, "Var >  Media", cex=1.2)

# Normal
# Homogeneidad de varianzas = la varianza no cambia con la media
plot(x,(x-x+50), type = "l", col = "orange", lwd = 5,
    xlab="Media", ylab="Varianza",main="Normal")

# Gamma
# Varianza = media^2 / forma
plot(x,(x^2/100), type = "l", col = "orange", lwd = 5,
    xlab="Media", ylab="Varianza",main="Gamma")
```

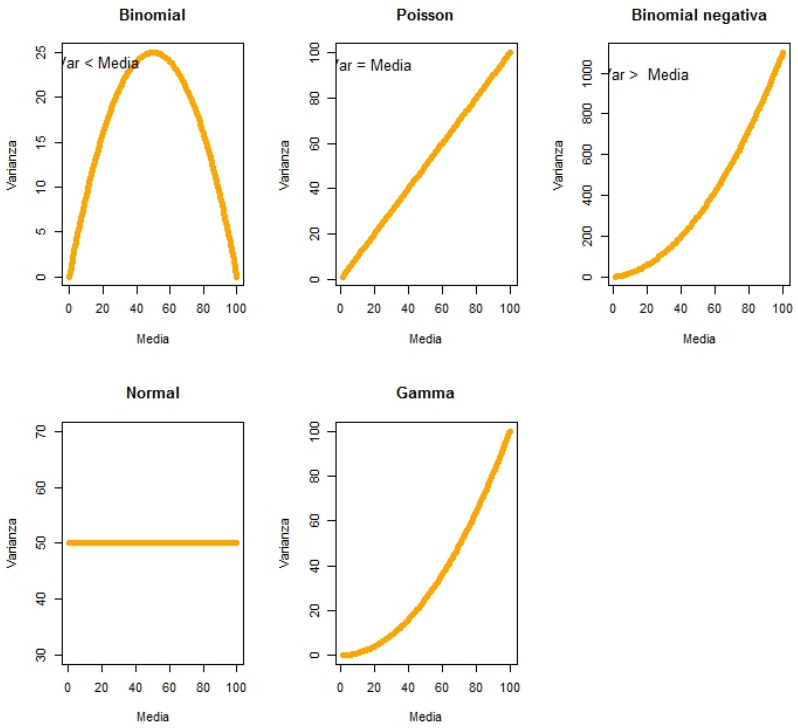


Figura 27.2.: Varianza en función de media para las distribuciones binomial, poisson, binomial negativa, normal y gamma.

Referencias bibliográficas

Anderson D.R., Sweeney D. J. y Williams T. A. (2008). Estadística para administración y economía. 10ma edición. Ed. Cengage Learning.

Fox J., Weisberg S. (2010) Nonlinear Regression and Nonlinear Least Squares in R. An Appendix to An R Companion to Applied Regression, second edition.

Gelman, A., Hill J. (2007). Data analysis using regression and multilevel/hierarchical models. Ed. Cambridge University Press.

Kleiber, C. y Zeileis, A.. (2008). Applied Econometrics with R. Springer-Verlag: New York.

Pinheiro J. C. y Bates D. M. (2000). Mixed-effects models in S and S-plus. Ed.Springer.

Webster, A. L. (2000). Estadística aplicada a los negocios y la economía. 3era edición. Ed. Irwin McGraw- Hill.