

## CAPÍTULO 2

---

### Repaso del Lenguaje R

---

#### 2.1. Características de R

1. Es un software libre, de código abierto, para programación orientada a objetos tipo S3 y S4. Una ejemplo de la capacidad de la programación con objetos S4 está en las librerías `distr`, `distrEx` (mirar el documento de ayuda de éstas).
2. La información sobre R está en el sitio web <http://www.cran.r-project.org/>, incluyendo
3. Task View
4. R Journal
5. Manuals
6. Search (Google)
7. Packages
8. R Binaries

### 2.1.1. Instalación de R

1. Entrar a `http://www.cran.r-project.org/`.
2. Seleccionar: “R Binaries”
3. Seleccionar: “Windows”
4. Seleccionar: “base”
5. Seleccionar: “Download R 4.1.1 for Windows (86 megabytes, 32/64 bit)”.
6. Ejecutar el archivo .exe desde el directorio donde se descargó.

### 2.1.2. Ayudas de R

La ayuda (help) de R esta es una direccion URL interna que se abre con un navegador. Por ejemplo, para utilizar Chrome hay que colocar la dirección interna del ejecutable del navegador en la consola de R

```
options(browser="C:/Program Files (x86)/Google/Chrome/Application/chrome
```

luego, en la ventana de comandos se escribe, por ejemplo, `help("mean")`, ó el signo de interrogación y la palabra sobre la que se busca ayuda `?mean`.

En la ventana Windows de R en el vínculo “Ayuda”, abre una página html con un menú de ayudas. Es útil para explorar el contenido de las librerías.

### 2.1.3. Instalación librerías de R

R instala varias librerías por defecto, por ejemplo, la librería `stats`. El resto de las librerías se deben instalar desde la consola ó mediante comandos. Los comandos siguientes muestran el número de librerías disponibles a la fecha. Nota: entre 2020 y 2021 este número aumento un 10 %, aproximadamente.

```
nrow(available.packages())
--- Please select a CRAN mirror for use in this session
Sys.Date()
"2020-10-01"
```

```
16321
"2021-10-07"
17842
```

Instalación desde la consola de R.

- Dar Enter en Paquetes
- Luego Instalar paquete(s)
- Luego Secure Cran mirror: escoge una localización del cran, p.ej. Ecuador(Quito)
- Luego se escoge el **repositorio**. Por defecto es CRAN, pero existen alternos como R-Forge, BioC, Omegahat.
- A continuación aparece la lista de paquetes que se pueden instalar.

Algunas librerías para series de tiempo:

- `forecast`: ajuste, diagnósticos, pronósticos.
- `mFilter`: varios filtros para tendencia.
- `tseries`: varias funciones.
- `TSA`: funciones del texto
- `urroot`: pruebas de hipótesis
- `arima`: modelos arima-sarima.
- `nnet`: redes neuronales.

Instalación con comandos. En la librería `utils` (instalada por defecto) está el comando

```
install.packages("forecast", dependencies=TRUE)
```

En algunos casos una librería ya no está activa en el CRAN (deprecated), pero todavía se proporcionan los archivos de instalación en un archivo comprimido con extensión `.tar.gz`. En Google hay instrucciones para instalar una librería desde el archivo comprimido.

Algunas librerías pueden estar localizadas fuera del CRAN, por ejemplo en `github.com`.

Cuáles librerías utilizar para algún procedimiento se pueden encontrar ingresando palabras claves seguidas de la frase “with R”, en, por ejemplo, Google.

#### 2.1.4. Algunos manuales de R

- Entrar a `http://www.cran.r-project.org/`.
- Seleccionar: “Contributed”,
- Seleccionar: “Manuals”

En Contributed:

- Seleccionar: “Non-English Documents”. Hay varias opciones.
- “R para Principiantes”, the Spanish version of “R for Beginners”, translated by Jorge A. Ahumada (PDF).
- A Spanish translation of “An Introduction to R” by Andrés González and Silvia González (PDF, Texinfo sources).
- “Metodos Estadisticos con R y R Commander” by Antonio Jose Saez Castillo (PDF, ZIP, 2010-07-08).

#### 2.1.5. Algunos manuales Internet

dirección url: `https://rc2e.com/`: “Welcome to the R Cookbook 2nd Edition”, por James (JD) Long, and Paul Teetor, 2019-09-26.

dirección url: `https://stats.idre.ucla.edu/`: UCLA: Institute for digital research and education. Statistical consulting. (ver el vínculo para la página de R: “Class notes: Introduction to R”).

## 2.2. Estructuras de datos

### 2.2.1. Objeto vectores

Hay varios tipos de vectores: lógico, entero, doble (numérico), complejos y carácter. Todos se generan con el comando “c()”.

```
# Ejemplo de tipos de vectores
e = c(34, 35, 56, 78)
g = c("h", "m", "m", "m")
s = c(3.4, 4.5, 2.4, 12.3)
L = c(TRUE, FALSE, FALSE, NA)
r = complex(3)
r[1] = -0.8 - 1.3i
r[2] = Conj(r[1])
r[3] = 3
#-----
str(e)
num [1:4] 34 35 56 78
str(g)
chr [1:4] "h" "m" "m" "m"
str(s)
num [1:4] 3.4 4.5 2.4 12.3
str(L)
logi [1:4] TRUE FALSE FALSE NA
str(r)
cplx [1:3] -0.8-1.3i -0.8+1.3i 3+0i
#-----
e = as.integer(e)
str(e)
int [1:4] 34 35 56 78
#-----
a = as.numeric(L)
a
[1] 1 0 0 NA
```

```
str(a)
num [1:4] 1 0 0 NA
```

### 2.2.2. Objeto data.frame

Un objeto `data.frame` es una clase de objeto en R que es una colección de columnas que contienen datos, que no tienen que ser del mismo tipo, pero deben tener el mismo número de filas. Las columnas pueden tener nombres. Un ejemplo simple es el siguiente.

Se define `D` como `data.frame`, y contiene las columnas `edad`, `genero`, `salario`. Entonces

```
e = c(34, 35, 56, 78)
g = c("h", "m", "m", "m")
s = c(3.4, 4.5, 2.4, 12.3)
D = data.frame(edad=e, genero=g, salario=s)
str(D)
'data.frame':  4 obs. of  3 variables:
 $ edad      : num  34 35 56 78
 $ genero    : Factor w/ 2 levels "h","m": 1 2 2 2
 $ salario   : num  3.4 4.5 2.4 12.3
es = D$edad/D$salario
es
[1] 10.000000  7.777778 23.333333  6.341463
```

### 2.2.3. Objetos tibble y Tidyverse

La librería `tidyverse` carga otras librerías como `dplyr`, `tibble`, `ggplot2`. Habilita el uso de varios comandos para modificar los objetos `tibble`, que son similares a un `data.frame`. Ver <sup>(1)</sup> versión en español de “R for Data Science”, de Hadley Wickham y Garrett Grolemund. El objetivo es poder realizar varias operaciones en secuencia para modificar un `tibble`, encadenadas mediante el operador `%>%`.

---

<sup>1</sup><https://es.r4ds.hadley.nz/tibbles.html>

- Un objeto tibble se puede generar directamente o se puede convertir un `data.frame` previo en un tibble con la instrucción

```
library(tidyverse)
G = tibble(
  x = 1:5,
  y = 1,
  z = x^2 + y
)
G = as_tibble(G)
```

- Los comandos para modificar ó añadir columnas en un tibble son `summarize`, `group_by`, `ungroup`, `mutate`, `filter`, `as_data.frame`.

- Ejemplo con `mutate`

```
#-----Ejemplo
library(tidyverse)
G = tibble(
  x = rnorm(5, 2, 3),
  y = 1,
  z = x^2 + y)
G = as_tibble(G %>% mutate(t = time(x), u = x/t))
View(G)
```

- Retomando los datos en el `data.frame` D del ejemplo anterior de `data.frame`, se convierte en un tibble y se añade columna edad. Luego se calculan estadísticos por grupos de Nombre en un nuevo tibble.

```
# uso de mutate
Dt <- as_tibble(
D %>% mutate(edad = c(rep(23, 3), rep(45, 3), rep(32, 3)))
)

# uso de group_by + summarize + mutate

Mt = as_tibble(Dt %>%
```

```

group_by(Nombre) %>%
  summarize(m = mean(Rate1),
            s = sd(Rate1),
            n = n()) %>%
  # n() calcula total observaciones por grupo
  ungroup() %>%
  mutate(cv = s/m)
View(Mt)

```

■ La función `filter` extrae subgrupos

```

Bt = as_tibble(Dt %>%
  filter(Nombre == "Ben" | Nombre == "Cat", Rate2 >= 30))
View(Bt)

```

## 2.2.4. Objeto list

Una lista (`list`) es otro tipo de objeto en R, más general que un `data.frame`. En una lista se coloca un grupo de otros objetos que pueden ser: escalares, `data.frames`, cadenas de símbolos. Las funciones en R generalmente retornan un objeto que puede ser un escalar, o también vector, o matriz, o un `data.frame`, o un `list` nuevamente. El siguiente es un ejemplo simple de objeto `list`

```

# -----ejemplo simple de objeto list
#         usando el data.frame D del ejemplo:
#         ''ejemplo.media.por.grupos.r''
a = matrix(c(2,3,4,5),2,2)
b = c("complex","real")
d = runif(120) # 120 números aleatorios entre 0 y 1
L = list(a=a,b=b,d=d,D=D)
#-----mirar la estructura de L
str(L)
$ a: num [1:2, 1:2] 2 3 4 5
$ b: chr [1:2] "complex" "real"
$ d: num [1:120] 0.04572 0.00316 0.82478 0.83271 0.99105 ...

```



```

$ D:'data.frame':      9 obs. of  4 variables:
 ..$ Nombre: chr [1:9] "Aira" "Aira" "Aira" "Ben" ...
 ..$ Mes    : int [1:9] 1 2 3 1 2 3 1 2 3
 ..$ Rate1  : int [1:9] 12 18 19 53 22 19 22 67 45
 ..$ Rate2  : int [1:9] 23 73 45 19 87 45 87 43 32
#-----extraer e Nodo 'Nombre'
L$D$Nombre
[1] "Aira" "Aira" "Aira" "Ben"  "Ben"  "Ben"  "Cat"  "Cat"  "Cat"

```

## 2.3. Lectura de datos en archivos

R posee varias funciones para lectura de datos. Incluyendo leer archivos que se encuentran en un sitio Web determinado, dado que exista una conexión de internet.

### 2.3.1. La función `read.table`

La función `G = read.table("nombre")` se utiliza para leer datos en R de tal forma que retorna un objeto `data.frame`, en este ejemplo indicado por la letra `G`, y donde “nombre” es el nombre del archivo en donde están los datos, por ejemplo, `nombre = base.dat`. Las extensiones pueden ser “.dat”, “.txt”, “.prn”.

En particular, “.prn” es una alternativa cuando se quieren leer datos de Excel. En este caso se guarda el archivo excel con el formato “.prn”, que corresponde a columnas separadas por espacios. No es conveniente salvar un archivo excel con extensión “.txt” porque este formato guarda controles de tabuladores invisibles que impiden la lectura de R.

Si cada columna en el archivo de lectura de datos tiene un nombre entonces el comando se modifica colocando `G = read.table("nombre", header = TRUE)`. Entonces añadir la instrucción `attach(G)`, lo hace que los nombres de las columnas pasen a ser los nombres de las variables activas en R. Cuando una de las columnas es la fecha de la observación, con el formato, por ejemplo, día-mes-año: “23/10/2010”, es conveniente colocar la instrucción de lectura en la forma

```
G = read.table("nombre", header = TRUE, stringsAsFactors=FALSE)
```

El efecto es que toma los datos de las fechas como caracteres y no como factores, que es un tipo de datos para utilizar en diseño de experimentos.

### 2.3.2. La función `read.table` datos en una URL

En este ejemplo se muestra cómo utilizar la función `read.table` para leer datos de un archivo en una página web. El archivo tiene dos columnas con los nombres: `fecha` y `x`. Los datos de la variable `fecha` son alfanuméricos.

Código R 2.1: Ejemplo de uso de `read.table` con datos en una dirección URL

```
#-----
#---Ejemplo A.6.4 lectura de una URL

# Ver https://a-little-book-of-r-for-time-series.readthedocs.io/
# en/latest/src/timeseries.html
# An example is a data set of the number of births
# per month in New York city ,
# from January 1946 to December 1959

archivo = "http://robjhyndman.com/tsdldata/data/nybirths.dat"
G = read.table(archivo)
str(G)
x = ts(G,frequency=12,start=c(1946,1))

#-----grafica
np = length(x)
#-----convierte fecha a formato de R
fechas = seq(as.Date("1946/01/01"), as.Date("1959/12/31"), by="months")
ejex.mes = seq(fechas[1],fechas[np], "months")
ejex.año = seq(fechas[1],fechas[np],"years")

plot(fechas,x, xaxt="n", panel.first = grid(),
type='b',ylab='numero recién nacidos')
axis.Date(1, at=ejex.mes, format="%m/%y")
```

```

axis.Date(1, at=ejex.año, labels = FALSE, tcl = -0.2)

#-----lectura archivos .csv
archivo = "https://stats.idre.ucla.edu/stat/data/poisson_sim.csv"
H = read.csv(archivo)
str(H)

#----grafica trayectorias
N = H$num_awards
plot(time(N),N,type='s')

#----grafica datos discretos en tabla

I = H$math
plot(table(I))

```

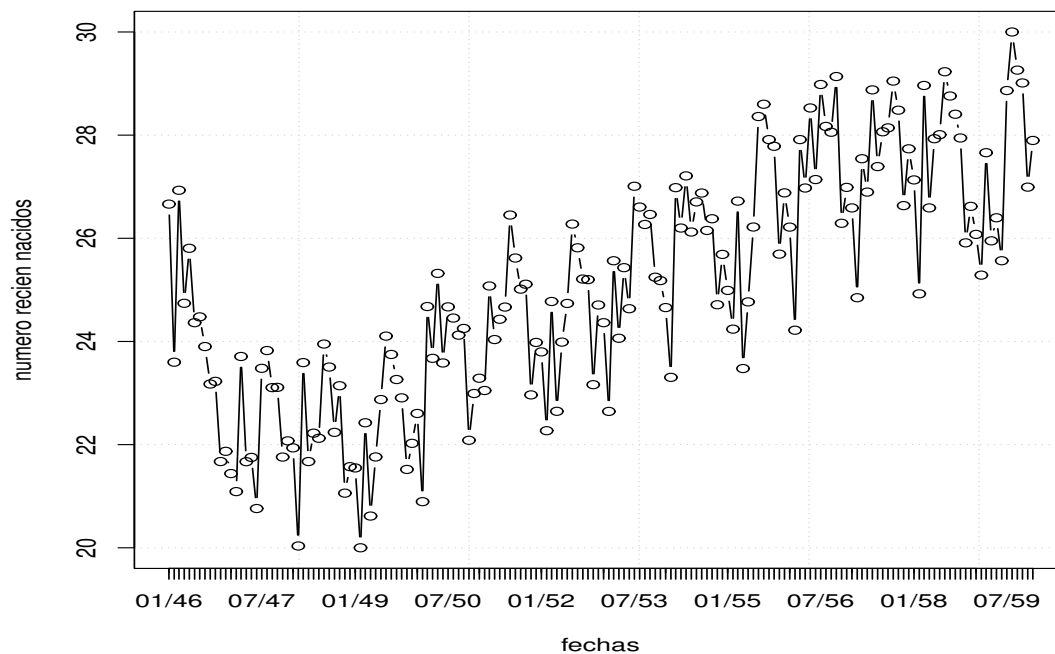


Figura 2.1: No nacimientos diarios

### 2.3.3. La función `ts`

Una serie de tiempo  $T_t$ ,  $t = 1, 2, \dots, T$ , se ingresa inicialmente como un objeto tipo vector numérico columna.

Utilizando el comando `ts()` se puede añadir información acerca de las fechas y la frecuencia de muestreo, es decir, cuántas observaciones anuales se tomaron, así como la fecha inicial. En este ejemplo se asigna una frecuencia de 12, es decir, datos mensuales. Además, se especifica que la serie inicia en Enero (01) de 1990.

Se puede ingresar los datos de la serie de tiempo directamente en el programa mediante la instrucción `structure()`.

Código R 2.2: Ejemplo de uso de la función `structure` para lectura de datos

```
# leer la serie de tiempo como un vector numerico
y=structure(c(1574,1368,1387,1109,1257,1376,2143,1208,
2007,1876,1702,1819,1802,1205,1684,1682,1991,2394,1914,
2499,2130,2529,2328,2076,2496,1647,2518,2205,2395,2891,
2712,2427,2477,2860,2505,3355,1760,2318,3111,2570,2868,
3042,2749,2839,3140,2909,2982,3667,2814,2732,3265,3166,
2792,3742,3099,3278,4120,3553,3675,3799,3427,3234,3733,
3642,3553,3647,3624,2973,3597,3731,4092,4100,2762,3953,
4152,4229,4419,4774,4313,4060,4664,4374,4419,4908,4321,
4772,4361,4969,5111,5014,4858,5159,5086,5379,5605,5269))

# convertir a objeto ts
y=ts(y,frequency=12,start=c(1990,1))

# generar un vector de fechas, clase 'Date'
fechas = seq(as.Date("1990/3/1"), length.out = length(y), by =
"months")

# grafica con fechas
ts.plot(y,main="serie F")

# graficar con fechas con mas detalle: mes-año
```

```

np = length(y)
ejex.mes = seq(fechas[1],fechas[np], "months")
ejex.año = seq(fechas[1],fechas[np],"years")

plot(fechas,y, xaxt="n", panel.first = grid(),type='b',
ylab='produccion.mes.')
axis.Date(1, at=ejex.mes, format="%m/%y")
axis.Date(1, at=ejex.año, labels = FALSE, tcl = -0.2)

# añadir una horizontal en el nivel 4000
abline(h = 4000, col='red')

# cual es la fecha en la cual se supera el nivel
# 4000 por primera vez?

j=1
while( y[j] < 4000 ){
j = j + 1}
# respuesta
fechas[j]
[1] "1994-11-01"
# añadir una vertical en esta fecha
points(fechas[j],y[j],type='h',col='blue')

```

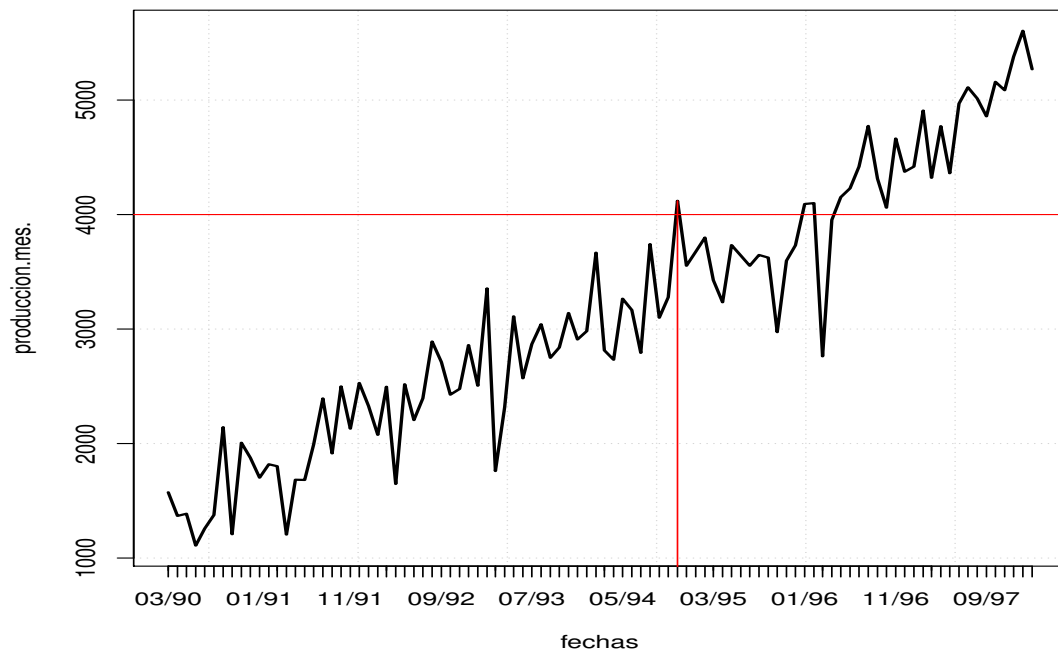


Figura 2.2: Gráfica de una serie mensual

### 2.3.4. Funciones de lectura de archivos Excel

Las funciones de lectura de archivos Excel son: `G = read.csv("nombre", header = TRUE)`, para archivos excel “delimitados por comas”, de extensión .csv.

Los archivos con extensión “.xls”, “.xlsx” se pueden leer con la función `read_excel()` de la librería `readxl`.

Código R 2.3: Ejemplo de lectura de datos desde archivos Excel

```
#-----
install.packages("readxl")
library(readxl)
res <- read_excel("international-petroleum-world-cr.xlsx")
attach(res)

# El comando plot.ts(D) grafica varias series
```

```

#(hasta 10 máximo), que deben estar en una matriz ó
# data.frame, por ejemplo, D.

str(res)
Classes 'tbl_df', 'tbl' and 'data.frame':      486 obs. of
5 variables:
 $ fecha : POSIXct, format: "1973-01-01" ...
 $ mexico: num  453 458 458 463 468 468 463 463 463 473 ...
 $ golfo : num  20051 20335 20332 20241 21369 ...
 $ noopec: num  25378 25627 25654 25812 25913 ...
 $ total : num  54389 54930 54995 55049 56323 ...
plot.ts(res[,c(2,3,4,5)])

# Pero se pueden comparar las series utilizando el
# comando lines.
#-----grafica
np = nrow(res)
#-----convierte fecha a formato de R
fechas = as.Date(fecha, format="%Y/%m/%d")
ejex.mes = seq(fechas[1], fechas[np], "months")
ejex.año = seq(fechas[1], fechas[np], "years")

plot(fechas, res$golfo, xaxt="n", panel.first = grid(),
type='l', lwd=2, ylab='produccion.mes.', ylim=c(9000,45000))
axis.Date(1, at=ejex.mes, format="%m/%y")
axis.Date(1, at=ejex.año, labels = FALSE, tcl = -0.2)
lines(fechas, res$noopec, col='red', lwd=2)

```

### 2.3.5. Series con datos faltantes

Varias librerías permiten imputar datos faltantes en series de tiempo. Por ejemplo `pastecs`.

Código R 2.4: Ejemplo de imputacion de datos faltantes en una serie de tiempo

```

#-----

```

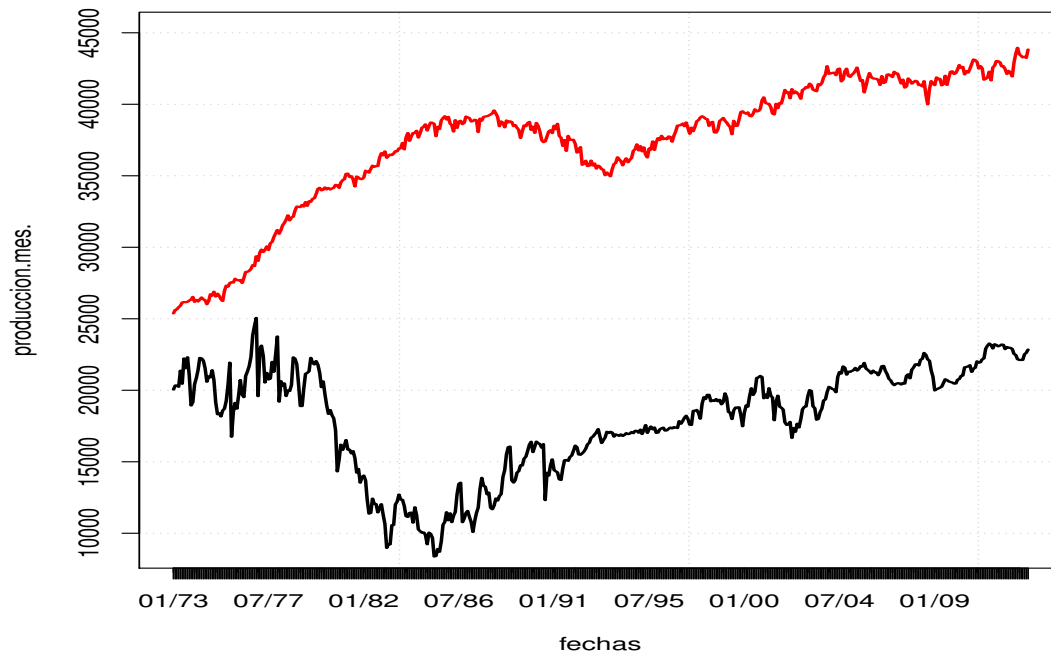


Figura 2.3: Gráfica producción mensual de crudo para países del Golfo y para los no pertenecientes a la Opec (en rojo)

```
install.packages("pastecs")
library(pastecs)
library(help="datasets")
attach(airquality)
help(airquality)
# Daily readings of the following air quality values for May 1,
# 1973 (a Tuesday) to September 30, 1973.
# Ozone: Mean ozone in parts per billion from 1300 to 1500 hours at
# Roosevelt Island

# 'data.frame':  153 obs. of  6 variables:
# $ Ozone   : int  41 36 12 18 NA 28 23 19 8 NA ...
# $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
# $ Wind    : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
```



```

# $ Temp    : int  67 72 74 62 56 66 65 59 61 69 ...
# $ Month   : int   5 5 5 5 5 5 5 5 5 5 ...
# $ Day      : int   1 2 3 4 5 6 7 8 9 10 ...

# Nótese que la serie Ozone tiene datos faltantes , NA.
# El procedimiento de imputación es con base en
# interpolación mediante splines.
n = length(Ozone)
x = seq(1,n)
ozone.lleno = regul(x, y=Ozone,
xmin=min(x), n=length(x),
units="days", deltat=1,
methods="spline",
rule=2, f=0.5, periodic=FALSE,
window=(max(x) - min(x))/(n - 1),
split=100, specs=NULL)

str(ozone.lleno)
layout(1:2)
ts.plot(Ozone)
ts.plot(ozone.lleno$y$Series, ylab="imputada")

```

### 2.3.6. Funciones para escritura de datos

Hay funciones para escribir datos en un archivo, como la función `write.matrix` de la librería MASS

```

#-----
D = data.frame(fechas=fecha, x=x)
require(MASS)
write.matrix(D, "h.dat")

```

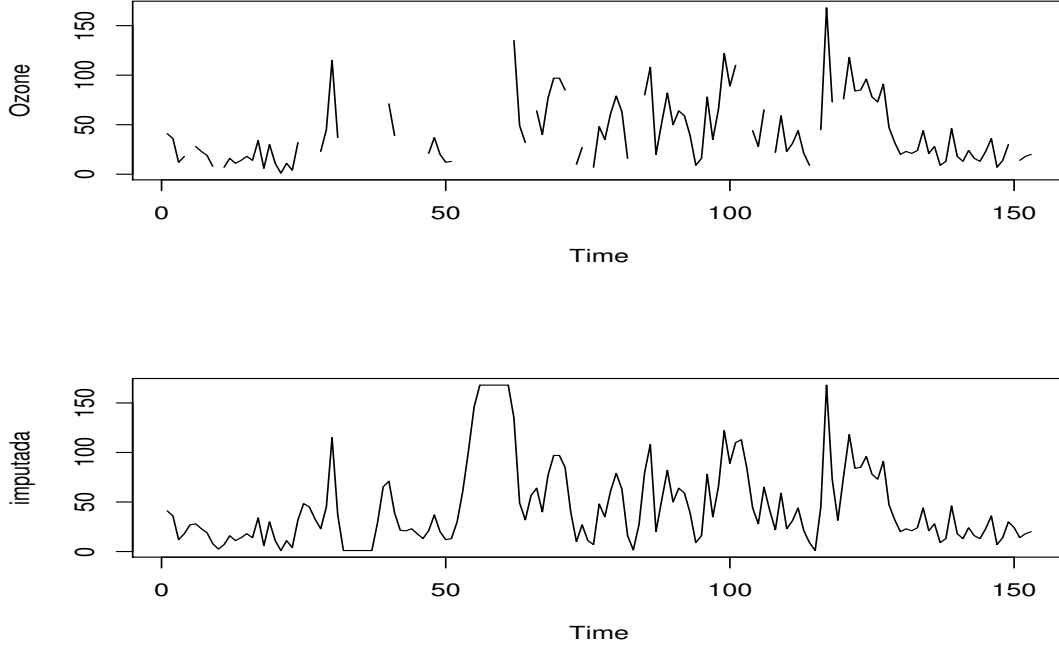


Figura 2.4: Serie con datos faltantes versus serie con imputación mediante splines

## 2.4. Repaso del Modelo de Regresión Lineal

El modelo de regresión lineal múltiple es básico para los modelos de componentes de series de tiempo. Una exposición más completa de este modelo está en, por ejemplo, en Kleinbaum et al. [1988] y Sheather [2009, caps. 5,6].

El modelo siguiente es lo suficientemente general para introducir los conceptos básicos. Suponga que se tiene una muestra de tamaño  $T$  de la variable aleatoria  $Y$ , y  $T$  mediciones de las variables  $X_1, X_2$ , que no se consideran aleatorias, que es una manera de expresar que estos datos están prefijados

$$Y_t, X_{1,t}, X_{2,t}, \quad t = 1, \dots, T.$$

Además suponga que  $\varepsilon_t \sim iid(0, \sigma^2)$  es una muestra de una variable aleatoria continua de media cero y varianza  $\sigma^2$ . Por ejemplo, una  $N(0, \sigma^2)$  ó una t-Student  $t_\nu$ . Si se cumple que

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \varepsilon_t, \quad (2.1)$$

entonces se dice que las variables  $Y, X_1, X_2$  satisfacen un modelo de regresión lineal. Donde  $Y_t$  es la variable dependiente y  $X_{1,t}, X_{2,t}$  son las variables explicativas ó predictoras y el término  $\varepsilon_t$  es el error ó residuo. Las variables  $\beta_0, \beta_1, \beta_2, \sigma^2$  se denominan los parámetros.

Si se define la matriz de diseño  $X \in \mathbb{R}^{T \times 3}$ , y los vectores  $\underline{Y}, \underline{\varepsilon}$

$$X = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} \\ 1 & X_{1,2} & X_{2,2} \\ \vdots & \vdots & \vdots \\ 1 & X_{1,T} & X_{2,T} \end{bmatrix}, \underline{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_T \end{bmatrix}, \underline{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{bmatrix}, \underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix},$$

entonces el modelo se puede expresar en forma matricial como

$$\underline{Y} = X\underline{\beta} + \underline{\varepsilon}. \quad (2.2)$$

Los supuestos del modelo lineal son

1. Las  $\varepsilon_t$  no están correlacionadas.
2. Las  $\varepsilon_t$  tienen varianzas constante.
3. El rango de la matriz  $X$  es 3, es decir, no existen constantes  $a, b$  tal que

$$aX_{1,t} + bX_{2,t} = 1.$$

De manera equivalente, las columnas de  $X$  son linealmente independientes.

El modelo se dice que es lineal porque  $Y_t$  se relaciona linealmente con  $X_{1,t}, X_{2,t}$ , es decir,  $Y_t$  es una combinación lineal de  $X_{1,t}, X_{2,t}$ . Relaciones de la forma siguiente son ejemplos de modelos no lineales:

$$Y_t = \alpha_0 + \frac{\alpha_1}{1 + e^{\beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t}}} + \varepsilon_t, \quad (2.3)$$

$$Y_t = \begin{cases} \alpha Y_{t-1} + \varepsilon_t & \text{if } Y_{t-1} \leq r \\ \beta Y_{t-1} + \gamma \varepsilon_t & \text{if } Y_{t-1} > r \end{cases} \quad (2.4)$$

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-1} \varepsilon_{t-1} + \varepsilon_t \quad (2.5)$$

El modelo (2.3) es un ejemplo simple de una red neuronal. La forma general de una red neuronal con  $p = 1, 2, \dots$  nodos de entrada y  $q = 1, 2, \dots$  nodos ocultos es

$$Y_t = \alpha_0 + \sum_{j=1}^q \alpha_j f \left( \beta_{0,j} + \sum_{k=1}^p \beta_{j,k} X_{k,t} \right) + \varepsilon_t, \quad (2.6)$$

donde se elige  $f(x) = (1 + e^x)^{-1}$ , entre otras opciones. Para el caso de (2.3) se tiene  $p = 2, q = 1$ . En la sección §5.6 se retoman las redes neuronales con más detalle.

Es conveniente señalar que, además de los conceptos de modelos lineal y no lineal, también existen los de serie lineal y no lineal, ver por ejemplo, Priestley [1988, Section 11.5.1, pag. 869]. Pero son diferentes.

En la sección §5.6 se definirá con mayor precisión serie no lineal. Muchas series conocidas se sabe que son de naturaleza no lineal, por ejemplo, las que muestran ciclos como las poblaciones de animales e insectos, la serie del número de manchas solares en la superficie del sol, entre otras.

Pero para decidir a priori si una serie de tiempo es no lineal se recurre a ciertas pruebas de hipótesis.

Una vez que se tiene evidencia de la no linealidad de la serie, se recomienda utilizar modelos no lineales, como las redes neuronales. En Venables and Ripley [2002], pag. 243:

“Las redes neuronales proveen una manera flexible de generalización de modelos lineales. Son modelos de regresión no lineal pero con muchos parámetros, lo que las hace muy flexibles, capaces de aproximar cualquier función suave.”

El pronóstico de  $Y_t$  se define en general como una esperanza condicional

$$\begin{aligned} \hat{Y}_t &= \mathbb{E}(Y_t | Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}) \\ &= f(Y_{t-1}, \dots, Y_{t-p}), \end{aligned}$$

para cierta función  $f(\cdot)$ . Si la serie es lineal y cumple otras condiciones adicionales, entonces esta esperanza es una función lineal de las variables predictoras, por lo que un modelo lineal es adecuado. Aplicar un modelo no lineal sería innecesario. En

cambio, si la serie es no lineal la esperanza condicional se podría aproximar por un modelo no lineal.

En el blog de F. Diebold, Diebold [2018], hay discusiones y observaciones útiles sobre el uso de los métodos de Aprendizaje de Máquina aplicados a series de tiempo.

### 2.4.1. Estimación de los Parámetros

Los parámetros de los modelos (2.1) y (2.3), son:  $\beta_0, \beta_1, \beta_2, \sigma^2$  y se busca estimarlos a partir de la muestra  $Y_t, X_{1,t}, X_{2,t}, t = 1, 2, \dots, T$ . Algunos métodos de estimación son:

1. Mínimos cuadrados ordinarios. (MCO)
2. Mínimos cuadrados no lineales. (MCNL)
3. Mínimos cuadrados robustos. (MCR)
4. Máxima verosimilitud. (MLE)
5. Momentos.

### 2.4.2. Mínimos cuadrados ordinarios

Los estimadores MCO de los coeficientes  $\beta_i$  en (2.1), se definen como el vector  $\underline{\hat{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$  que minimiza la función objetivo

$$G(\underline{\beta}) = \sum_{t=1}^T [Y_t - (\beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t})]^2, \quad (2.7)$$

es decir,  $\underline{\hat{\beta}}$  cumple

$$\forall \underline{\beta}, G(\underline{\hat{\beta}}) \leq G(\underline{\beta}).$$

Y se escribe

$$\underline{\hat{\beta}} = \underset{\underline{\beta} \in \mathbb{R}^3}{\operatorname{argmin}} G(\underline{\beta}).$$

Minimizar la función objetivo  $G(\underline{\beta})$  es resolver el problema de estimación por MCO. Sin embargo,  $\hat{\underline{\beta}}$  satisface la fórmula matricial,

$$\hat{\underline{\beta}} = (X'X)^{-1}X'Y, \quad (2.8)$$

ver la comprobación en la sección al final. Hay dos propiedades básicas del estimador MCO  $\hat{\underline{\beta}}$ . Es un estimador lineal insesgado, es decir,

$$\mathbb{E}(\hat{\underline{\beta}}) = \underline{\beta}, \quad (2.9)$$

ver la comprobación en la sección al final. Se puede comprobar que el estimador MCO coincide, en este caso del modelo lineal, con el estimador de máxima verosimilitud. Lo cual implica que comparte la propiedad de mínima varianza, que significa,

$$Var(\hat{\beta}_i) \leq Var(\hat{\beta}_i^*), \quad i = 0, 1, 2,$$

para  $\hat{\beta}_i^*$  cualquier estimador diferente de MCO.

Se comprueba que de (2.8) se obtiene la expresión para la matriz de varianzas y covarianzas del vector  $\hat{\underline{\beta}}$  como

$$Var(\hat{\underline{\beta}}) = \sigma^2(X'X)^{-1}, \quad (2.10)$$

ver la comprobación en la sección al final. La diagonal de la matriz  $\sigma^2(X'X)^{-1}$  contiene las varianzas de los  $\hat{\underline{\beta}}$ , indicadas por

$$Var(\hat{\beta}_i) = \sigma_{\hat{\beta}_i}^2 = diag(\sigma^2(X'X)^{-1})_i. \quad (2.11)$$

### 2.4.3. Pronósticos

Una vez definidos los  $\hat{\beta}_i$ , se definen los pronósticos dentro de la muestra ó valores ajustados, como

$$\hat{Y}_t = \hat{\beta}_0 + \hat{\beta}_1 X_{1,t} + \hat{\beta}_2 X_{2,t}, \quad t = 1, \dots, T. \quad (2.12)$$

Nótese que se cumple  $\hat{Y} = X\hat{\underline{\beta}}$ . Reemplazando (2.8) en esta expresión se obtiene

$$\hat{Y} = X(X'X)^{-1}X'Y = HY, \quad (2.13)$$

donde la matriz simétrica

$$H = X(X'X)^{-1}X' \quad (2.14)$$

se denomina la matriz “hat” ó “de proyección”. Los pronósticos fuera de la muestra (predicciones) se definen como

$$\hat{Y}_{T+h} = \hat{\beta}_0 + \hat{\beta}_1 X_{1,T+h} + \hat{\beta}_2 X_{2,T+h}, \quad h = 1, \dots, m, \quad (2.15)$$

donde  $X_{1,T+h}$ ,  $X_{2,T+h}$  hay que proveerlos.

#### 2.4.4. Error cuadrático medio, MSE

El error cuadrático medio, MSE (*mean square error*), se define como

$$MSE = \frac{1}{T-k} \sum_{t=1}^T (Y_t - \hat{Y}_t)^2 = \frac{1}{T-k} \sum_{t=1}^T \hat{\varepsilon}_t^2, \quad (2.16)$$

donde  $k$  es el número de coeficientes en el modelo, es decir, la dimensión de  $\beta$ . El MSE se define también como el estimador de la varianza del error

$$\hat{\sigma}^2 = MSE. \quad (2.17)$$

Y se define el estimador

$$\widehat{Var}(\hat{\beta}) = \hat{\sigma}^2 (X'X)^{-1}. \quad (2.18)$$

MSE se calcula directamente con instrucción

```
MSE = anova(m2) [ 'Residuals' , 'Mean Sq' ]
```

#### 2.4.5. Residuos

Se define los residuos estimados como

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t, \quad t = 1, \dots, T. \quad (2.19)$$

En forma vectorial es

$$\underline{\hat{\varepsilon}} = \underline{Y} - X\underline{\hat{\beta}} = \underline{Y} - H\underline{Y} = (I - H)\underline{Y}. \quad (2.20)$$

La esperanza y la varianza de  $\underline{\hat{\varepsilon}}$  son

$$\mathbb{E}(\underline{\hat{\varepsilon}}) = \mathbb{E}(\underline{Y} - X\underline{\hat{\beta}})$$

$$\begin{aligned}
&= \mathbb{E}(\underline{Y}) - \mathbb{E}(X\underline{\hat{\beta}}) \\
&= \mathbb{E}(X\underline{\beta}) - X\mathbb{E}(\underline{\hat{\beta}}) \\
&= X\underline{\beta} - X\underline{\beta} = \underline{0}.
\end{aligned}$$

$$\text{Var}(\underline{\hat{\varepsilon}}) = \sigma^2(I - H), \quad (2.21)$$

ver la comprobación de esta identidad en la sección al final.

Los elementos de la diagonal de  $\sigma^2(I - H)$  son las varianzas  $\text{Var}(\hat{\varepsilon}_t) = \sigma^2(1 - H_{t,t})$ .

Los residuos estandarizados se definen como

$$r_t = \frac{\hat{\varepsilon}_t}{\hat{\sigma}\sqrt{1 - H_{t,t}}}. \quad (2.22)$$

**Ejemplo 2.4.1.** *Los datos de tres variables  $Y, X_1, X_2$ , para ajustar el modelo de regresión (2.1) están en la página web*

<http://www.stat.umn.edu/geyer/5102/data/ex5-3.txt>

#### Código R 2.5: Ejemplo de regresión lineal con R

```

#-----
#----- leer datos en una url
archivo = "http://www.stat.umn.edu/geyer/5102/data/ex5-3.txt"
# archivo = "ex5-3.txt"
D = read.table(archivo, header=T)
str(D)
attach(D)
#----- estimar el modelo y=x1+x2+e, por MCO
m2 = lm(y ~ x1 + x2)
summary(m2)
#----- calcular residuos
ehat = residuals(m2)
#----- estimar mse y sigma
sigma=summary(m2)$sigma
MSE = sigma^2
#----- directamente
MSE = anova(m2)[ 'Residuals ', 'Mean Sq' ]
#----- calcular residuos estandarizados r

```



```

require(MASS)
r = stdres(m2)
#-----valores ajustados
yhat = fitted(m2)
#-----graficas
par(mfrow=c(2,2))
plot(x1,y,type='p',main="A")
plot(r ,main="B")
abline(h=0)
plot(y,yhat,type='p',main="C")
plot(ehat ,type='p',main="D")
abline(h=0)

```

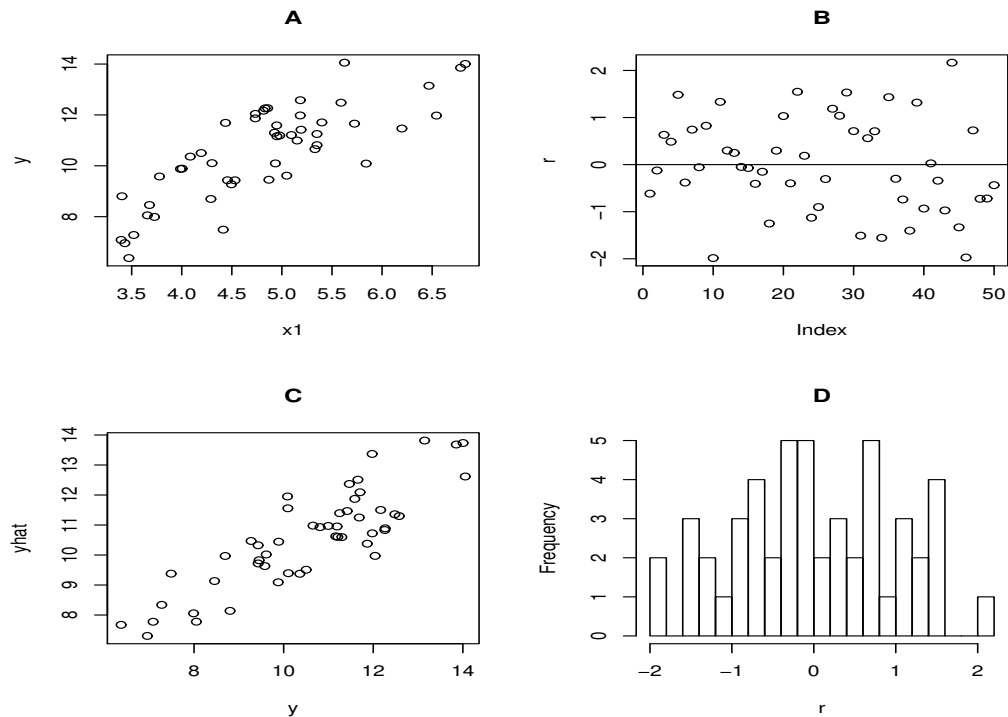


Figura 2.5: gráficas A:  $y$  versus  $x_1$ , B: residuos  $r$ , C:  $y$  versus  $\hat{y}$ , D: histograma residuos  $r$

## Demostraciones de identidades

- Demostración de la fórmula (2.8).

*Demostración.* Expresando  $G(\underline{\beta})$  por

$$G(\underline{\beta}) = \|\underline{Y} - X\underline{\beta}\|^2 = (\underline{Y} - X\underline{\beta})'(\underline{Y} - X\underline{\beta}).$$

donde  $\underline{x}'$  es el transpuesto de  $\underline{x}$ , también indicado por  $\underline{x}^\top$ . Utilizando fórmulas para derivadas matriciales se plantea obtener el mínimo como el valor  $\underline{\beta}$  que resuelve la ecuación

$$\frac{\partial}{\partial \underline{\beta}} G(\underline{\beta}) = \underline{0}.$$

Se puede comprobar que (ver, por ejemplo, <sup>(2)</sup>, ó Wikipedia <sup>(3)</sup>)

$$\frac{\partial}{\partial \underline{\beta}} G(\underline{\beta}) = 2X'(X\underline{\beta} - \underline{Y})$$

por tanto, la solución es

$$\hat{\underline{\beta}} = (X'X)^{-1}X'\underline{Y}.$$

□

- Comprobación de la propiedad de insesgamiento (2.9)

*Demostración.*

$$\begin{aligned} \mathbb{E}(\hat{\underline{\beta}}) &= \mathbb{E}((X'X)^{-1}X'\underline{Y}) = (X'X)^{-1}X'\mathbb{E}(\underline{Y}) \\ &= (X'X)^{-1}X'\mathbb{E}(X\underline{\beta} + \underline{\varepsilon}) = (X'X)^{-1}X'(X\underline{\beta} + \mathbb{E}(\underline{\varepsilon})) \\ &= (X'X)^{-1}X'X\underline{\beta} = \underline{\beta}. \end{aligned} \tag{2.23}$$

□

- Demostración de la identidad (2.10)

---

<sup>2</sup><http://www.matrixcalculus.org/>

<sup>3</sup>[https://en.wikipedia.org/wiki/Matrix\\_calculus](https://en.wikipedia.org/wiki/Matrix_calculus)

*Demostración.* Con base en la expresión para  $\hat{\underline{\beta}}$

$$\begin{aligned}\hat{\underline{\beta}} - \underline{\beta} &= (X'X)^{-1}X'Y - \underline{\beta} \\ &= (X'X)^{-1}X'(X\underline{\beta} + \underline{\varepsilon}) - \underline{\beta} \\ &= (X'X)^{-1}X'\underline{\varepsilon}\end{aligned}$$

de donde

$$\begin{aligned}Var(\hat{\underline{\beta}}) &= \mathbb{E}((\hat{\underline{\beta}} - \underline{\beta})(\hat{\underline{\beta}} - \underline{\beta})') = (X'X)^{-1}X'\mathbb{E}(\underline{\varepsilon}\underline{\varepsilon}')X(X'X)^{-1} \\ &= \sigma^2(X'X)^{-1}\end{aligned}$$

□

- Demostración de la identidad (2.21).

*Demostración.* Por definición de Varianza de un vector aleatorio de media cero,

$$\begin{aligned}Var(\hat{\underline{\varepsilon}}) &= \mathbb{E}(\hat{\underline{\varepsilon}}\hat{\underline{\varepsilon}}') \in \mathbb{R}^{T \times T} \\ &= \mathbb{E}((I - H)\underline{Y}[(I - H)\underline{Y}]') \\ &= (I - H)\mathbb{E}(\underline{Y}\underline{Y}')(I - H)'\end{aligned}$$

A partir de  $\mathbb{E}(\underline{Y}\underline{Y}') = Var(\underline{Y}) = Var(X\underline{\beta} + \underline{\varepsilon}) = \mathbb{E}(\underline{\varepsilon}\underline{\varepsilon}')$ , la hipótesis  $\varepsilon_t \sim iid(0, \sigma^2)$  implica  $\mathbb{E}(\underline{\varepsilon}\underline{\varepsilon}') = \sigma^2 I$ , con  $I \in \mathbb{R}^{T \times T}$  la matriz identidad. Luego  $\mathbb{E}(\underline{Y}\underline{Y}') = \sigma^2 I$ . Se obtiene entonces  $Var(\hat{\underline{\varepsilon}}) = \sigma^2(I - H)(I - H)'$ . Se puede comprobar que  $H^2 = H$  y por tanto  $(I - H)(I - H)' = I - H$ , luego finalmente se obtiene  $Var(\hat{\underline{\varepsilon}}) = \sigma^2(I - H)$ . □

## 2.4.6. Mínimos Cuadrados Nolineales

En algunos casos es necesario ajustar un modelo no lineal. R posee funciones que permiten ajustar por mínimos cuadrados no lineales modelos, por ejemplo, de la forma

$$Y_t = g(X_{1,t}, X_{2,t}; \underline{\beta}) + \varepsilon_t, t = 1, 2, \dots, T. \quad (2.24)$$

donde  $g(x_1, x_2; \underline{\beta}) : \mathbb{R}^2 \rightarrow \mathbb{R}$  es una función dada que depende de los parámetros  $\underline{\beta}$ . Nótese que debe ser un modelo con errores aditivos. La función en R para mínimos

cuadrados no lineales es `nls()` y se aplica de la misma forma que `lm()`. Por ejemplo, para el modelo no lineal (2.3), pag. 31

$$Y_t = \alpha_0 + \frac{\alpha_1}{1 + e^{\beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t}}} + \varepsilon_t,$$

Se puede estimar con la función `nls()`, con la instrucción

```
m = nls(Y ~ a0+a1/(1+exp(b0+b1*X1+b2*X2)),
start=list(a0=0.5, a1=1, b0=0.1, b1=1, b2=1))
```

Nótese que la función requiere unos valores iniciales para los parámetros. En el capítulo siguiente hay un ejemplo concreto de estimación de este modelo.

### 2.4.7. Regresión Local Polinómica

Es una alternativa a la regresión lineal, que puede aproximar casos no lineales. Se puede adaptar al caso de series de tiempo, en cuyo caso produce estimadores de la tendencia de una serie.

Suponga los datos:  $(X_{1,i}, X_{2,i}, Y_i), i = 1, \dots, N$ . El modelo de regresión local polinómica asume que existe una función  $\mu(x_1, x_2)$  que cumple

$$Y_i = \mu(X_{1,i}, X_{2,i}) + \varepsilon_i \quad (2.25)$$

La función  $\mu(x_1, x_2)$  se puede aproximar en una vecindad del punto  $(X_{1,i}, X_{2,i})$  mediante un desarrollo de Taylor bivariado de orden 2

$$\begin{aligned} \mu(x_1, x_2) &\approx a_0 + a_1(x_1 - X_{1,i}) + a_2(x_2 - X_{2,i}) + \frac{1}{2}a_3(x_1 - X_{1,i})^2 + \\ &a_4(x_1 - X_{1,i})(x_2 - X_{2,i}) + \frac{1}{2}a_5(x_2 - X_{2,i})^2 \\ &= G(X_{1,i}, X_{2,i}; \underline{a}) \end{aligned}$$

con  $\underline{a} = (a_1, \dots, a_5)'$ .

Defina la función tricúbica

$$W(x) = (1 - |x|^3)^3, \quad -1 \leq x \leq 1. \quad (2.26)$$

Y defina para un punto  $(x_1, x_2)$  los pesos

$$W_i(x_1, x_2) = W\left(\frac{\|(X_{1,i}, X_{2,i}) - (x_1, x_2)\|}{h}\right) \quad (2.27)$$

donde  $h > 0$  es un parámetro que se fija arbitrariamente, y los índices  $i$  se escogen de tal forma que

$$\frac{\|(X_{1,i}, X_{2,i}) - (x_1, x_2)\|}{h} \leq 1$$

Para cada punto  $(x_1, x_2)$  se calcula

$$\hat{\underline{a}}(x_1, x_2) = \underset{\underline{a} \in \mathbb{R}^5}{\operatorname{argmin}} \left\{ \sum_{i=1}^N W_i(x_1, x_2) (Y_i - G(X_{1,i}, X_{2,i}; \underline{a}))^2 \right\} \quad (2.28)$$

Es estimador de regresión múltiple local en el punto  $(x_1, x_2)$  se define como el primer componente del vector  $\hat{\underline{a}}(x_1, x_2)$

$$\hat{\mu}(x_1, x_2) = \hat{a}_0(x_1, x_2) \quad (2.29)$$

## 2.5. Pruebas de Ajuste del Modelo

El problema de estimación por MCO no requiere el supuesto de Normalidad en los residuos:  $\varepsilon_t \sim iid N(0, \sigma^2)$ . Asumir este supuesto adicional implica que los estimadores MCO  $\hat{\underline{\beta}}$  se distribuyen Normal multivariado,  $\hat{\underline{\beta}} \sim N(\underline{\beta}, \sigma^2(X'X)^{-1})$ . Por tanto, las distribuciones de las marginales  $\hat{\beta}_i$  son Normales univariadas

$$\hat{\beta}_i \sim N(\beta_i, \sigma_{\hat{\beta}_i}^2), \quad (2.30)$$

donde las varianzas  $\sigma_{\hat{\beta}_i}^2$  son los elementos de la diagonal de la matriz  $\sigma^2(X'X)^{-1}$ . Además, con  $\hat{\sigma}_{\hat{\beta}_i} = \sqrt{\hat{\sigma}^2(X'X)^{-1}_{i,i}}$ , denominado el error estándar, se definen los estadísticos t-Student,  $t_i$

$$t_i = \hat{\beta}_i / \hat{\sigma}_{\hat{\beta}_i}. \quad (2.31)$$

## Terminología básica

Un repaso de la terminología básica de las pruebas de hipótesis. Para realizar un prueba se consideran cuatro pasos:

1. Hipótesis de la Prueba.
2. Estadístico de la Prueba.
3. Distribución del Estadístico bajo  $H_0$ .
4. Decisión.

		$H_0$ real	
		V	F
$H_0$ según prueba	V	OK	error II
	F	error I	OK

Tabla 2.1: Tabla de decisiones en una prueba de hipótesis

La prueba se define colocando expresiones para las hipótesis nula  $H_0$  y la alterna  $H_1$ .

El Error Tipo I es rechazar la Hipótesis nula  $H_0$  siendo ésta cierta. Se trata de un falso positivo porque en ocasiones la alterna  $H_1$  es el resultado de interés. El nivel de significación  $\alpha$  se define como

$$\mathbb{P}(\text{Error Tipo I}) = \alpha.$$

El Error Tipo II es no rechazar la nula  $H_0$  siendo la alterna  $H_1$  la cierta. Se trata de un falso negativo. La probabilidad de cometer este error es

$$\mathbb{P}(\text{Error Tipo I}) = \beta.$$

Además, la probabilidad  $1 - \beta$ , denominada la potencia de la prueba, se interpreta como la probabilidad de rechazar la nula correctamente, es decir, la probabilidad de rechazar  $H_0$  cuando  $H_1$  es cierta. Fijado el nivel de significación  $\alpha$  y el tamaño de la muestra, la potencia puede ser baja ó alta.

Las pruebas que tienen una baja potencia tienden a no rechazar la nula  $H_0$ . No tienen capacidad para detectar  $H_1$ . En ocasiones, la alterna  $H_1$  puede estar muy cerca de la nula  $H_0$ , y se requiere una prueba con alta potencia para detectarla.

## Pruebas de Significación de los parámetros

La prueba se describe así. Para  $i = 1, \dots, k$ ,

1. Hipótesis nula versus alterna:

$$H_0 : \beta_i = 0 \text{ versus } H_1 : \beta_i \neq 0.$$

Rechazar la nula se expresa diciendo: “el parámetro es significativamente diferente de cero”, ó simplemente “es significativo”.

2. Estadístico de prueba  $t_i = \hat{\beta}_i / \hat{\sigma}_{\hat{\beta}_i}$ .
3. Si  $H_0$  es cierta entonces  $t_i \sim t_{T-k}$ , t-Student con  $T - k$  grados de libertad, donde  $k$  es la dimensión del vector  $\underline{\beta}$ .
4. Decisión. Se plantean 3 maneras equivalentes para decidir acerca de la hipótesis nula  $H_0$ .

- 1 Si  $|t_i| > 1.96$  entonces se rechaza  $H_0 : \beta_i = 0$  con un nivel de significancia de 5 %, es decir, hay una probabilidad de 5 % de rechazar  $H_0$  siendo cierta.

- 2 Defina el valor p como

$$\text{Valorp} := \mathbb{P}(|t_{T-k}| > |t_{obs}|) = \mathbb{P}(t_{T-k} > |t_{obs}|) + \mathbb{P}(t_{T-k} < -|t_{obs}|).$$

Si  $\text{Valorp} < 0.05$  se rechaza  $H_0$  a un nivel de significancia del 5 %, de lo contrario no se rechaza  $H_0$ .

- 3 Con base en intervalos de confianza para el parámetro. Se rechaza la nula a un nivel de significancia de 5 % si

$$0 \notin [\hat{\beta}_i - 1.96 \hat{\sigma}_{\hat{\beta}_i}, \hat{\beta}_i + 1.96 \hat{\sigma}_{\hat{\beta}_i}]$$

Note que, con R se pueden obtener los intervalos anteriores con la instrucción `confint(m2, level=...)`. Por ejemplo,

**Ejemplo 2.5.1.** *Continuando con el Ejemplo 2.4.1, las pruebas t-Student se pueden programar, como se muestra a continuación. Para obtener los coeficientes estimados, los estadísticos  $t$  y los valores  $p$  se puede utilizar la función `coeftest` de la librería `lmtest`. También el comando `summary(modelo)`.*

```
#-----pruebas de significacion
require(lmtest)
coeftest(m2)
t test of coefficients:

              Estimate Std. Error t value  Pr(>|t|)
(Intercept)  0.43421     0.90937   0.4775 0.6352321
x1           1.41790     0.17192   8.2473 1.091e-10 ***
x2           0.67427     0.16884   3.9934 0.0002275 ***
---
Signif. codes:
0 *** 0.001 ** 0.01 * 0.05 . 0.1
```

## Prueba F

Se define la hipótesis

$$H_0 : \forall j = 1, 2, \dots, k, \beta_j = 0$$

$$H_1 : \exists j, \ni b_j \neq 0$$

Se define la suma de cuadrados total  $SST$  como la suma de cuadrados de la regresión  $SSR$  más suma de cuadrados de errores  $SSE$ , dados por

$$\begin{aligned} SST &= \sum_{t=1}^T (Y_t - \bar{Y})^2 \\ &= \sum_{t=1}^T (\hat{Y}_t - \bar{Y})^2 + \sum_{t=1}^T (Y_t - \hat{Y}_t)^2 \\ &= SSR + SSE. \end{aligned} \tag{2.32}$$

El estadístico para esta prueba se define como

$$F = \frac{SSR/(k-1)}{SSE/(T-k)} = \frac{SSR/(k-1)}{MSE} \tag{2.33}$$

el cual se distribuye, bajo  $H_0$ , con una distribución tipo F-Fisher

$$F \stackrel{H_0}{\sim} F_{k-1, T-k}. \tag{2.34}$$



Si se rechaza la nula  $H_0$ , las variables  $X_{1,t}, \dots, X_{k-1,t}$  tienen capacidad para predecir los valores de  $Y_t$ . Si el valor observado de  $F$  es  $F_{obs} > F_{k-1,T-k,\alpha}$ , se rechaza  $H_0$ . De manera equivalente, si el Valor  $p$ , definido como  $\text{Valor } p = \mathbb{P}(F_{obs} > F_{k-1,T-k})$ , es menor que el nivel de significación  $\alpha$ , se rechaza la nula.

**Ejemplo 2.5.2.** Continuando con el Ejemplo 2.4.1, con  $k = 3$ ,  $T = 50$ ,  $\alpha = 0,05$ , se tiene que  $F_{k-1,T-k,\alpha} = F_{2,47,0.95} = 3.195056$ , utilizando el comando: `qf(0.95, 2, 47)`. Luego, dado  $F_{obs} = 66.48$ , como  $F_{obs} > 3.2$ , se rechaza la nula  $H_0$ , es decir,  $X_{1,t}$  y  $X_{2,t}$  tienen capacidad predictiva sobre  $Y_t$ .

```
summary(m2)
#-----
F-statistic: 66.48 on 2 and 47 DF,  p-value: 1.987e-14
```

### 2.5.1. Estadísticos de Ajuste y de Selección de Modelos

Se define el estadístico de ajuste  $R$  cuadrado,  $R^2$ , como

$$R^2 = 1 - \frac{\sum_{t=1}^T \hat{\varepsilon}_t^2}{\sum_{t=1}^T (Y_t - \bar{Y})^2} = 1 - \frac{SSE}{SST}. \quad (2.35)$$

como el porcentaje de la varianza de  $Y_t$  que es explicado (atribuible) por las variables explicativas. Observe que  $k$  no aparece.

$R^2 \geq 0.7$  buena

$R^2 = 0.6$  buena pero no mucho

$R^2 = 0.4$  regular-mala

$R^2 = 0.2$  desechar, seguramente la prueba  $F$  y las pruebas  $t$  no rechazan la nula

Hay que anotar que en los textos sobre regresión se recomienda no usar el  $R^2$  si el modelo no tiene intercepto. Carece de sentido en este caso.

Se define el  $R$  cuadrado ajustado,  $\bar{R}^2$  como

$$\bar{R}^2 = 1 - \frac{(T-1)SSE}{(T-k)SST}. \quad (2.36)$$

Como  $R^2 = 1 - SSE/SST$ , entonces  $\bar{R}^2$  se puede escribir en terminos de  $R^2$  de la siguiente manera

$$\bar{R}^2 = 1 - \frac{T-1}{T-k}(1 - R^2). \quad (2.37)$$

En las expresiones siguientes el símbolo  $\nearrow$  indica aumento en una variable, y  $\searrow$  disminución. En (2.37), si  $k \nearrow$ , entonces  $T - k \searrow$  y  $\frac{T-1}{T-k} \nearrow$ , luego  $\bar{R}^2 \searrow$ . Lo que se interpreta como que  $\bar{R}^2$  penaliza aumentar el número de parámetros.

Se define el criterio de información de Akaike, AIC, como

$$AIC = e^{2k/T} \frac{\sum_{t=1}^T \hat{\varepsilon}_t^2}{T} = \frac{(T-k)\hat{\sigma}^2}{T} e^{2k/T}. \quad (2.38)$$

El AIC también se define como el logaritmo de (2.38), y además se puede relacionar con la suma de cuadrados  $G(\hat{\beta})$  en (2.7), usando las identidades en (2.16) y (2.17), para obtener

$$\begin{aligned} AIC &= \log \left( \frac{(T-k)\hat{\sigma}^2}{T} \right) + \frac{2k}{T} \\ &= \log \left( \frac{G(\hat{\beta})}{T} \right) + \frac{2k}{T} \end{aligned}$$

Se observa que si  $k \nearrow$ , entonces  $AIC \nearrow$ . Si  $G(\hat{\beta}) \searrow$ , entonces  $AIC \searrow$ .

Por lo que el AIC disminuye mientras menor sea  $G(\hat{\beta})$  ó  $\hat{\sigma}$ , es decir, mientras mejor sea el ajuste, pero aumenta si el número de parámetros  $k$  aumenta. El modelo con menor AIC debe ser el modelo que mejor ajusta con el menor número de parámetros, lo que se relaciona con el principio de parsimonia (navaja de Ockam, ver <sup>(4)</sup>)

En Diebold [1999, pág. 73]

“Como en el caso de  $S^2$ , muchos de los criterios más importantes para la selección de modelos de pronóstico tienen la forma de factor de penalización multiplicado por  $MSE$ ”.

Se define el criterio de información de Schwarz BIC

$$BIC = T^{\frac{k}{T}} \frac{\sum_{t=1}^T \hat{\varepsilon}_t^2}{T} = T^{\frac{k}{T}} \frac{(T-k)\hat{\sigma}^2}{T} \quad (2.39)$$

El BIC también se define como el logaritmo de (2.39).

$$BIC = \log \left( \frac{(T-k)\hat{\sigma}^2}{T} \right) + \frac{k}{T} \log(T). \quad (2.40)$$

---

<sup>4</sup>[https://es.wikipedia.org/wiki/Navaja\\_de\\_Ockham](https://es.wikipedia.org/wiki/Navaja_de_Ockham)

De la definición del criterio  $BIC$  se observa:

Si  $k \nearrow$ , entonces  $BIC \nearrow$ . Si  $G(\hat{\beta}) \searrow$ , entonces  $BIC \searrow$ .

La regla para utilizar AIC y BIC es que para escoger entre varios modelos de regresión con respecto a la misma variable dependiente, lineales ó no lineales, se escoge el de menor AIC ó menor BIC.

Diebold [1999, pág. 74–75], introduce dos criterios para elegir entre AIC y BIC: Consistencia y Eficiencia asintótica.

**Definición 2.5.1 (Consistencia).** *Si se examinan varios modelos, y el modelo generador de los datos (MGD) está incluido, el criterio se dice consistente si a medida que  $T$  aumenta, la probabilidad de que el criterio seleccione el MGD tiende a 1.*

**Resultado:**  $\bar{R}^2$ , AIC no son consistentes pero BIC sí es.

**Definición 2.5.2 (Eficiencia Asintótica).** *Un criterio es eficiente asintóticamente si a medida que  $T$  aumenta, elige un modelo cuya varianza de error de pronóstico a un paso*

$$\text{Var}(Y_{t+1} - \hat{Y}_{t+1} | Y_s, s \leq t)$$

*tiende a la que se obtendría con el MGD.*

**Resultado:** AIC es eficiente asintóticamente pero BIC no. En Diebold [1999, pág. 75]:

“Muchos autores recomiendan usar el modelo más parsimonioso que selecciona el  $BIC$  en igualdad de circunstancias.”

**Ejemplo 2.5.3.** *Continuando con el Ejemplo 2.4.1, se examinan 3 modelos lineales:*

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \varepsilon_t, \quad (2.41a)$$

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \varepsilon_t, \quad (2.41b)$$

$$Y_t = \beta_1 X_{1,t} + \beta_2 X_{2,t} + \beta_3 X_{2,t}^2 + \varepsilon_t \quad (2.41c)$$

*Las instrucciones en R se dan a continuación. Los resultados aparecen en la tablas siguientes.*

```
#----estimar 4 modelos
m1 = lm(y ~ x1)
summary(m1)
m2 = lm(y ~ x1 + x2)
summary(m2)
x3 = x1^2
m3 = lm(y ~ x1 + x2 + x3)
summary(m3)
```

Tabla 2.2: Resultados estimación MCO modelo m1

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.378	0.880	2.703	0.009
x1	1.698	0.180	9.446	0.000

Tabla 2.3: Resultados estimación MCO modelo m2

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.434	0.909	0.477	0.635
x1	1.418	0.172	8.247	0.000
x2	0.674	0.169	3.993	0.000

Tabla 2.4: Resultados estimación MCO modelo m3

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-5.650	3.375	-1.674	0.101
x1	4.044	1.416	2.857	0.006
x2	0.617	0.167	3.683	0.001
x3	-0.262	0.140	-1.869	0.068

El comando en R con resultados se muestra a continuación.

Escoger uno de los modelos comparando los estadísticos de ajuste y de selección. Para esto se diseñó la función (de autor) `medidas`, que se debe cargar con la función `source`, como aparece en el código siguiente.

```
medidas = function(m, y, k) {
```

```

T = length(y)
yest = fitted(m)
sse = sum((yest-y)^2)
ssr = sum((y-mean(y))^2)
mse = sse/(T-k)
R2 = 1 - sse/ssr
Ra2 = 1 - (T-1)*(1-R2)/(T-k)
aic = log((T-k)*exp(2*k/T)*mse/T)
bic = log(T^(k/T)*(T-k)*mse/T)
M = c(mse,Ra2,aic,bic)
return(M)
}

```

Y su aplicación con los resultados, permite escoger uno de los modelos mediante los estadísticos de ajuste y selección, como se muestra en el código siguiente.

```

source("medidas.r")
k1 = 2; k2 = 3; k3 = 4;
N1 = medidas(m1,y,k1)
N2 = medidas(m2,y,k2)
N3 = medidas(m3,y,k3)
N = cbind(N1,N2,N3)
rownames(N) = c("MSE", "R2-ajus", "AIC", "BIC")
colnames(N) = c("m1", "m2", "m3")
(N)

```

Los resultados están en la tabla 2.5 siguiente.

Tabla 2.5: Resultados los estadísticos de ajuste y selección

	m1	m2	m3
MSE	1.240	0.945	0.898
R2-ajus	0.643	0.728	0.741
AIC	0.254	0.002	-0.031
BIC	0.330	0.116	0.122

Cuál modelo se escoge?. La elección estaría entre m2 y m3. El AIC dá el m3. Pero el BIC dá el m2.

### 2.5.2. Pruebas F parciales múltiples

En casos en los que los modelos a comparar consisten de un modelo inicial con  $k$  variables predictoras y un modelo ampliado con  $k + p$  variables predictoras, es posible medir el efecto de la inclusión de las nuevas  $p$  variables predictoras, mediante una prueba F parcial múltiple. Tomando como ejemplo  $p = 2$ , el modelo completo se puede escribir

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \beta_1^* X_{1,t}^* + \beta_2^* X_{2,t}^* + \varepsilon_t, \quad (2.42)$$

La hipótesis nula es que las variables predictoras adicionales no contribuyen de manera significativa en la predicción de  $Y_t$

$$H_0 : \beta_1^* = \beta_2^* = 0. \quad (2.43)$$

El estadístico de la prueba F parcial múltiple es

$$F = \frac{(SSR(ampliado) - SSR(reducido))/(k - 1)}{MSE(ampliado)} \quad (2.44)$$

donde ampliado corresponde al modelo en (2.42) y reducido es el modelo original. El estadístico se distribuye, bajo  $H_0$ , según una distribución tipo F-Fisher

$$F \stackrel{H_0}{\sim} F_{k-1, T-k-p}. \quad (2.45)$$

De la ayuda de la función `anova`: `help('`anova`')`, cuando se le da una secuencia de objetos, `anova` prueba los modelos entre sí en el orden especificado, mediante pruebas F parciales múltiples.

**Ejemplo 2.5.4.** *Continuando con los 3 modelos del Ejemplo 2.5.3, se aplicó la función `anova` `anova(m1,m2,m3)`.*

```
Analysis of Variance Table
```

```
Model 1: y ~ x1
```

Model 2:  $y \sim x_1 + x_2$

Model 3:  $y \sim x_1 + x_2 + x_3$

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	48	59.498				
2	47	44.424	1	15.074	16.7929	0.0001675 ***
3	46	41.290	1	3.134	3.4914	0.0680637 .

La conclusión es que al comparar  $m_1$  y  $m_2$  se rechaza la nula. Luego, añadir  $X_{2,t}$  aumenta la capacidad predictiva. Sin embargo, incorporar  $X_{2,t}^2$  resulta en no rechazo de la nula, aunque con un valor  $p$  cercano a 0.05. La conclusión es escoger el modelo  $m_2$ , como lo hizo el estadístico BIC en el análisis de las medidas de selección.