Politica Economica II

Máximo Sangiácomo

Índice

1 Introduccion a R **5**

1.1 Primeros pasos . . . 5
1.2 Busacar ayuda . . . 6
1.3 Tipos de datos . . . 6
1.4 Limpieza de memoria . . . 6
1.5 Asignación de valores . . . 7
1.6 Operadores aritméticos . . . 7
1.7 Operadores relacionales . . . 8
1.8 Operadores lógicos . . . 9

2 Base de datos **11**

2.1 Directorio de trabajo . . . 11
2.2 Cargar datos . . . 12
2.2.1 Ingrasar datos con `tidyverse` . . . 12
2.3 Exportar datos . . . 13
2.4 Variables . . . 13
2.5 Merge . . . 14
2.6 Variables: group_by, mutate . . . 14
2.7 Guardar datos . . . 15
2.8 Valores missing . . . 16
2.9 Análisis de datos . . . 17
2.9.1 Tablas . . . 18
2.10 group_by, summarise . . . 19
2.11 Gráficos . . . 19
2.12 GGPlot . . . 20
2.13 Guardar un gráfico . . . 22

3 Conceptos generales **23**

3.1 Estimacion . . . 23
3.2 Prediccion . . . 23
3.3 Metodos parametricos . . . 24
3.4 Metodos no parametricos . . . 25
3.5 Evaluacion de la precision del modelo . . . 25
3.5.1 Calidad del ajuste . . . 25
3.5.2 Trade-off Sesgo-Varianza . . . 27
3.5.3 Clasificacion . . . 29
3.5.4 Matriz de confusion . . . 30

4 Arboles de decision **31**

4.1 *Classification and Regression Tree* (CART) . . . 31
4.2 Bagging . . . 33
4.3 Random Forest . . . 34

5 Trabajo Practico **35**

5.1 Reglas del Trabajo practico . . . 35
5.2 Enunciado del Trabajo Practico . . . 35
5.3 Aplicacion practica . . . 36

Descripcion del curso

El objetivo del curso es abordar distintas metodologías de análisis de datos desde el punto de vista teórico y práctico utilizando el programa `R`. Si bien, dada la extensión de las clases, la cobertura de cada tema no busca ser exhaustiva intenta capturar las principales intuiciones de cada método.[1]

En los Capítulos 1 y 2 se hace una breve introducción al manejo de bases de datos en `R`. El Capitulo 3 se ocupa de los conceptos teóricos[2] a ser utilizados en la práctica. Luego, el Capítulo 4 presenta la metodología arboles. Finalmente, el Capítulo 5 presenta una guía de ejercicios para la elaboración del **Trabajo Práctico**.

[1] Alguno de los temas no será cubierto completamente durante las clases pero la idea es dejar el material disponible para que pueda ser revisado de manera individual.

[2] Se muestran las principales formulaciones matemáticas aunque la derivación formal de los resultados excede los objetivos de este curso.

Capítulo 1

Introduccion a R

- La programación de rutinas en *software* específico para la manipulación y análisis de bases de datos permite asegurar procesos homogéneos, documentados, fácilmente auditables, modificables y que pueden ser compartidos entre diferentes usuarios. Además, resulta sumamente útil para realizar tareas repetitivas generando ganancias de eficiencia.
- Es muy importante realizar anotaciones, tanto para compartir código con otro usuario como para uno mismo en el futuro (por ejemplo, cuáles son los insumos/*output*, los ¿por qué?). La combinación de teclas **Ctrl/Cmd + Shift + R** permite crear secciones en *scripts* que luego sirven para navegar y ser ordenados.

Importante. *A la gloria no se llega por un camino de rosas.* **Trabajar, trabajar y trabajar.** Osvaldo Zubeldia.

1.1 Primeros pasos

`R` es un lenguaje orientado a **objetos** (vectores, listas, matrices). Si bien al principio puede parecer demasiado complejo, no es así. De hecho, una característica destacada de `R` es su flexibilidad.

Mientras que un software clásico muestra inmediatamente los resultados de un comando, `R` los almacena en un objeto, por lo que se puede realizar un análisis sin el resultado desplegado.

`R` (el motor) puede complementarse con **RStudio** (tablero de instrumentos) que es una IDE (*integrated development environment*) para operar de manera mas amigable (editor con sintaxis y distintos espacios de trabajo). Cada uno se encuentra disponible en:[1]

1. R

[1] Buscar las versiones adecuadas para el sistema operativo utilizado.

2. RStudio

Una vez instalados los programas se deben descargar "paquetes" que agregan funcionalidades al paquete que viene incorporado (base).

```
#Descarga de programa
install.packages('tidyverse')

#Carga de programa antes de utilizarlo en un script
library(tidyverse)
```

Eventualmente se puede utilizar una función específica de un paquete previamente instalado sin cargar el paquete completo. Por ejemplo, si se quiere utilizar la función `read_excel()` del paquete `readxl`.

```
readxl::read_excel()
```

1.2 Busacar ayuda

```
# Por comando
?rm # Para poder ver la ayuda el paquete debe estar instalado
help(lm)
```

- Buscar el tab Help en la ventana de abajo a la derecha de RStudio
- Google

1.3 Tipos de datos

- character/string
- numeric (integer, double)
- factor (variables categóricas, importante para clasificación)
- logical
- date

1.4 Limpieza de memoria

```
rm(list = ls()) # Elimina todos los objetos en memoria
gc() # Garbage Collection
```

1.5 Asignación de valores

```
# nombre_objeto <- valor
x <- 1
x = 5
x
```

```
## [1] 5
```

```
y = x
y = 4
```

1.6 Operadores aritméticos

```
y + x
```

```
## [1] 9
```

```
y - x
```

```
## [1] -1
```

```
y * x
```

```
## [1] 20
```

```
4 / 8
```

```
## [1] 0.5
```

```
8 %% 4
```

```
## [1] 0
```

```
2**5
```

```
## [1] 32
```

```
2^5
```

```
## [1] 32
```

```
sqrt(9)
```

```
## [1] 3
```

```
log(1)
```

```
## [1] 0
```

1.7 Operadores relacionales

```
# < <= > >= == !=
x == 1
```

```
## [1] FALSE
```

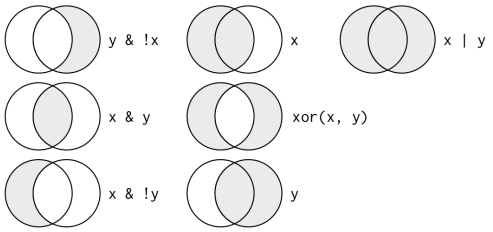
```
x == y
```

```
## [1] FALSE
```

```
y < x
```

```
## [1] TRUE
```

1.8 Operadores lógicos



Nota: Se puede usar || (o) y && (y) para combinar múltiples expresiones lógicas. Estos operadores se llaman "cortocircuito": tan pronto como || ve el primer VERDADERO devuelve VERDADERO sin calcular nada más. Tan pronto como && ve el primer FALSO, devuelve FALSO.

```
# ! & | && ||
TRUE
```

```
## [1] TRUE
```

```
!FALSE
```

```
## [1] TRUE
```

```
!F
```

```
## [1] TRUE
```

```
F & T
```

```
## [1] FALSE
```

```
F | T
```

```
## [1] TRUE
```

```
x == 1 & y==4
```

```
## [1] FALSE
```

```
x == 1 | y==4
```

```
## [1] TRUE
```

```
!(x > y); x <= y
```

```
## [1] FALSE
```

```
## [1] FALSE
```

Capítulo 2

Base de datos

En esta clase nos vamos a centrar en el uso de `tidyverse`.

En `R` existen dos tipos de bases de datos `data.frame()` y `tibble()` que son las bases de datos de `tidyverse` el mejor paquete para manipulación y transformación de datos (ver Wickham and Grolemund (2017)). Un `data.frame` (objeto df) se convierte fácilmente a tibble (y viceversa).

```
# Un data.frame (objeto df) se convierte fácilmente a tibble
tib = as_tibble(df)
```

Las tibbles tienen algunas funciones especiales como poder usar nombres de variables con espacio (se deben utilizar *back ticks*).

```
library(tidyverse)
tb <- tibble(
  `Plazo Fijo` = "espacio",
  `2000` = "numero"
)
tb
```

```
## # A tibble: 1 x 2
##   `Plazo Fijo` `2000`
##   <chr>        <chr>
## 1 espacio      numero
```

2.1 Directorio de trabajo

```
# Para ver en que directorio estamos trabajando
getwd()
# Definir directorio. Notar barras invertidas en la ruta
setwd('C:/Documentos/CianciaDatos')
```

2.2 Cargar datos

Tip. La función `fread()` del paquete `data.table` es la más eficiente para grandes volúmenes de datos porque permite paralelizar con *multithread*.

```
# CSV
bd = read.csv("b_datos.csv", header=TRUE, stringsAsFactors=TRUE, sep=",")

bd = data.table::fread('b_datos.txt', header=TRUE, stringsAsFactors=F, sep='\t', nThre

bd = read.delim('datos/b_datos.txt', header=TRUE, stringsAsFactors=F, sep='\t')
```

```
# Excel
# También puede suministrarse la ruta de acceso completa
bd = readxl::read_excel('./data/datos_wb.xlsx', sheet='1')
datos = readxl::read_excel('./data/datos_ts.xlsx', sheet='datos')
str(bd)
```

```
## tibble [60 x 11] (S3: tbl_df/tbl/data.frame)
##  $ year      : num [1:60] 2011 2011 2011 2011 2011 ...
##  $ cname     : chr [1:60] "Argentina" "Brazil" "Chile" "France" ...
##  $ ccode     : chr [1:60] "ARG" "BRA" "CHL" "FRA" ...
##  $ gdp_pc2010: num [1:60] 10883 11628 13456 41369 36228 ...
##  $ gdp_pc2017: num [1:60] 24648 15323 22338 42864 42892 ...
##  $ gdp_2010  : num [1:60] 4.49e+11 2.30e+12 2.32e+11 2.70e+12 2.15e+12 ...
##  $ credit_ps : num [1:60] 14 58.1 101.3 96.8 94.1 ...
##  $ inv       : num [1:60] 17.2 20.6 23.1 22.4 19.7 ...
##  $ exports   : num [1:60] 18.4 11.6 37.8 28.4 26.9 ...
##  $ imports   : num [1:60] 16.8 12.4 34.4 30.4 28.3 ...
##  $ popu      : num [1:60] 4.13e+07 1.98e+08 1.72e+07 6.53e+07 5.94e+07 ...
```

2.2.1 Ingrasar datos con `tidyverse`

Cuadro 2.1: Vista de la base de datos (World Bank)

year	cname	ccode	gdp_pc2010	gdp_pc2017
2,011	Argentina	ARG	10,883	24,648
2,011	Brazil	BRA	11,628	15,323
2,011	Chile	CHL	13,456	22,338
2,011	France	FRA	41,369	42,864
2,011	Italy	ITA	36,228	42,892
2,011	United Kingdom	GBR	39,729	42,294

Comando	Separador
`read_csv()`	coma
`read_csv2()`	punto y coma
`read_tsv()`	tab
`read_delim()`	otros

2.3 Exportar datos

```
# CSV
write.csv(bd,"b_datos.csv")
write_csv()
write_excel_csv()
# TXT
write_delim()
write_tsv()

# Excel
library("xlsx")
# Primera base de datos
write.xlsx(USArrests, file = "b_datos.xlsx", sheetName = "IRIS", append = FALSE)
# Segunda base de datos
write.xlsx(mtcars, file = "b_datos.xlsx", sheetName="MTCARS", append=TRUE)
```

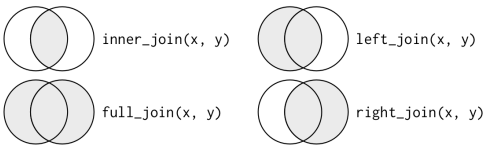
2.4 Variables

```
library(tidyverse)
bd1 = bd %>%
  mutate(gdp_pc2010bis = gdp_2010 / popu, # crear
```

```
        logGDP_pc2010 = log(gdp_pc2010),
        open = exports + imports,
        inv_demean = inv - mean(inv)) %>%
  rename(poblacion = popu) %>%  # rename (newname = oldname)
  mutate(gdp_2010 = NULL)       # drop (también con select(-gdp_2010))
```

year	cname	gdp_pc2010	gdp_pc2010bis	open	inv_demean
2,011	Argentina	10,883	10,883	35.2	-2
2,011	Brazil	11,628	11,628	23.9	2
2,011	Chile	13,456	13,456	72.2	4
2,011	France	41,369	41,369	58.8	4
2,011	Italy	36,228	36,228	55.1	1
2,011	United Kingdom	39,729	39,729	62.4	-3

2.5 Merge




```
meta = readxl::read_excel('./data/datos_wb.xlsx', sheet='2')
bd = left_join(bd, meta, by=c('ccode'))
```

year	cname	region
2011	Argentina	Latin America & Caribbean
2011	Brazil	Latin America & Caribbean
2011	Chile	Latin America & Caribbean
2011	France	Europe & Central Asia
2011	Italy	Europe & Central Asia
2011	United Kingdom	Europe & Central Asia

2.6 Variables: group_by, mutate

```
# Si quiero usar una función propia
demean = function(x) {x - mean(x, na.rm = TRUE)}
bd = bd %>%
  mutate(open = exports + imports) %>%
  dplyr::select(ccode, year, region, gdp_pc2017, credit_ps, inv, open) %>%
    arrange(ccode, year) %>%
```

```
  group_by(ccode) %>%
  mutate(obs = seq(1:length(ccode)),     # igual con row_number()
         gdp_gr = 100 * (gdp_pc2017 / dplyr::lag(gdp_pc2017, 1) - 1),
         credit_ps_mean = mean(credit_ps, na.rm = TRUE),
         dev = ifelse(region=='Latin America & Caribbean', 0, 1),
         gdp_dem = demean(gdp_pc2017)) %>%
  ungroup()
head(bd[c('ccode', 'dev', 'year', 'gdp_pc2017', 'gdp_gr', 'gdp_dem')],10)
```

```
## # A tibble: 10 x 6
##    ccode   dev  year gdp_pc2017 gdp_gr  gdp_dem
##    <chr> <dbl> <dbl>      <dbl>  <dbl>    <dbl>
##  1 ARG       0  2011     24648.  NA     1451.
##  2 ARG       0  2012     24119.  -2.15   922.
##  3 ARG       0  2013     24424.   1.27  1227.
##  4 ARG       0  2014     23550.  -3.58   353.
##  5 ARG       0  2015     23934.   1.63   737.
##  6 ARG       0  2016     23190.  -3.11    -7.58
##  7 ARG       0  2017     23597.   1.76   400.
##  8 ARG       0  2018     22759.  -3.55  -438.
##  9 ARG       0  2019     22064.  -3.06 -1133.
## 10 ARG       0  2020     19687. -10.8  -3511.
```

2.7 Guardar datos

```
bd = bd %>% select(ccode, year, region, gdp_gr, credit_ps, inv, open)
save(bd, file="datos_wb.rda")
```

ccode	year	region	gdp_pc2017	credit_ps	inv	open	obs	gdp_gr	cre
ARG	2018	Latin America & Caribbean	22759.4	NA	14.7	31.2	8	-3.6	
ARG	2019	Latin America & Caribbean	22063.9	NA	13.5	32.6	9	-3.1	
ARG	2020	Latin America & Caribbean	19686.5	NA	13.4	30.5	10	-10.8	
BRA	2018	Latin America & Caribbean	14668.3	60.2	15.1	28.9	8	1.0	
BRA	2019	Latin America & Caribbean	14763.9	62.6	15.3	28.5	9	0.7	
BRA	2020	Latin America & Caribbean	14064.0	70.2	16.4	32.4	10	-4.7	
GBR	2018	Europe & Central Asia	46037.9	134.6	17.8	63.0	8	0.6	
GBR	2019	Europe & Central Asia	46406.5	133.5	18.0	63.4	9	0.8	
GBR	2020	Europe & Central Asia	41627.1	146.4	17.6	55.1	10	-10.3	
ITA	2018	Europe & Central Asia	42052.6	76.7	17.8	60.3	8	1.1	
ITA	2019	Europe & Central Asia	42662.5	74.3	18.0	60.1	9	1.5	
ITA	2020	Europe & Central Asia	38992.1	83.6	17.8	55.3	10	-8.6	

2.8 Valores missing

Se debe tener presente que se elimina la fila completa.

```
#  Volvemos a la base de WB. Recordamos la estructura
str(bd)
```

```
## tibble [60 x 12] (S3: tbl_df/tbl/data.frame)
##  $ ccode         : chr [1:60] "ARG" "ARG" "ARG" "ARG" ...
##  $ year          : num [1:60] 2011 2012 2013 2014 2015 ...
##  $ region        : chr [1:60] "Latin America & Caribbean" "Latin America & Caribbea
##  $ gdp_pc2017    : num [1:60] 24648 24119 24424 23550 23934 ...
##  $ credit_ps     : num [1:60] 14 15.2 15.7 13.8 14.4 ...
##  $ inv           : num [1:60] 17.2 15.9 16.3 16 15.6 ...
##  $ open          : num [1:60] 35.2 30.5 29.3 28.4 22.5 ...
##  $ obs           : int [1:60] 1 2 3 4 5 6 7 8 9 10 ...
##  $ gdp_gr        : num [1:60] NA -2.15 1.27 -3.58 1.63 ...
##  $ credit_ps_mean: num [1:60] 14.7 14.7 14.7 14.7 14.7 ...
##  $ dev           : num [1:60] 0 0 0 0 0 0 0 0 0 0 ...
##  $ gdp_dem       : num [1:60] 1451 922 1227 353 737 ...
```

```
summary(bd[,1:4])
```

```
##     ccode                year         region            gdp_pc2017   
##  Length:60          Min.   :2011   Length:60          Min.   :14064  
##  Class :character   1st Qu.:2013   Class :character   1st Qu.:23273  
##  Mode  :character   Median :2016   Mode  :character   Median :32011  
##                     Mean   :2016                      Mean   :31864  
##                     3rd Qu.:2018                      3rd Qu.:42828  
##                     Max.   :2020                      Max.   :46406
```

```
# cuenta valores missing de CreditoSPriv
sum(ifelse(is.na(bd$gdp_gr),1,0))
```

```
## [1] 6
```

```
sum(ifelse(is.na(bd$credit_ps),1,0))
```

```
## [1] 4
```

```
sum(ifelse(is.na(bd$gdp_gr&bd$credit_ps),1,0))
```

```
## [1] 10
```

```
bd1 = na.omit(bd)
nrow(bd1)
```

```
## [1] 50
```

```
rm('bd1')
```

2.9 Análisis de datos

```
str(bd)
```

```
## tibble [60 x 12] (S3: tbl_df/tbl/data.frame)
##  $ ccode         : chr [1:60] "ARG" "ARG" "ARG" "ARG" ...
##  $ year          : num [1:60] 2011 2012 2013 2014 2015 ...
##  $ region        : chr [1:60] "Latin America & Caribbean" "Latin America & Caribbean"
##  $ gdp_pc2017    : num [1:60] 24648 24119 24424 23550 23934 ...
##  $ credit_ps     : num [1:60] 14 15.2 15.7 13.8 14.4 ...
##  $ inv           : num [1:60] 17.2 15.9 16.3 16 15.6 ...
##  $ open          : num [1:60] 35.2 30.5 29.3 28.4 22.5 ...
##  $ obs           : int [1:60] 1 2 3 4 5 6 7 8 9 10 ...
##  $ gdp_gr        : num [1:60] NA -2.15 1.27 -3.58 1.63 ...
##  $ credit_ps_mean: num [1:60] 14.7 14.7 14.7 14.7 14.7 ...
##  $ dev           : num [1:60] 0 0 0 0 0 0 0 0 0 0 ...
##  $ gdp_dem       : num [1:60] 1451 922 1227 353 737 ...
```

```
dim(bd)
```

```
## [1] 60 12
```

```
names(bd)
```

```
##  [1] "ccode"          "year"           "region"         "gdp_pc2017"
##  [5] "credit_ps"      "inv"            "open"           "obs"
##  [9] "gdp_gr"         "credit_ps_mean" "dev"            "gdp_dem"
```

```
glimpse(bd)
```

```
## Rows: 60
## Columns: 12
## $ ccode          <chr> "ARG", "ARG", "ARG", "ARG", "ARG", "ARG", "ARG", "ARG",~
## $ year           <dbl> 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2~
## $ region         <chr> "Latin America & Caribbean", "Latin America & Caribbean~
## $ gdp_pc2017     <dbl> 24647.63, 24118.87, 24424.14, 23550.10, 23933.89, 23189~
## $ credit_ps      <dbl> 14.00872, 15.21282, 15.72909, 13.82377, 14.41423, 13.66~
## $ inv            <dbl> 17.24828, 15.85753, 16.28951, 15.97995, 15.56475, 14.27~
## $ open           <dbl> 35.20615, 30.52654, 29.33393, 28.40679, 22.48623, 26.09~
## $ obs            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, ~
## $ gdp_gr         <dbl> NA, -2.1452844, 1.2656852, -3.5785805, 1.6296643, -3.11~
## $ credit_ps_mean <dbl> 14.68782, 14.68782, 14.68782, 14.68782, 14.68782, 14.68~
## $ dev            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ gdp_dem        <dbl> 1450.520204, 921.758446, 1227.027377, 352.989990, 736.7~
```

```
head(bd)
```

```
## # A tibble: 6 x 12
##   ccode  year region               gdp_pc2017 credit_ps   inv  open   obs gdp_gr
##   <chr> <dbl> <chr>                     <dbl>     <dbl> <dbl> <dbl> <int>  <dbl>
## 1 ARG    2011 Latin America & Car~     24648.      14.0  17.2  35.2     1  NA
## 2 ARG    2012 Latin America & Car~     24119.      15.2  15.9  30.5     2  -2.15
## 3 ARG    2013 Latin America & Car~     24424.      15.7  16.3  29.3     3   1.27
## 4 ARG    2014 Latin America & Car~     23550.      13.8  16.0  28.4     4  -3.58
## 5 ARG    2015 Latin America & Car~     23934.      14.4  15.6  22.5     5   1.63
## 6 ARG    2016 Latin America & Car~     23190.      13.7  14.3  26.1     6  -3.11
## # ... with 3 more variables: credit_ps_mean <dbl>, dev <dbl>, gdp_dem <dbl>
```

2.9.1 Tablas

Los valores `NA` afectan a todas las estadísticas. Opción na.rm = `F` / `T`.

```
bd %>% summarise(
  credit_ps_media = mean(credit_ps),
  inv_max = max(inv),
  open_min = min(open)
)
```

```
## # A tibble: 1 x 3
##   credit_ps_media inv_max open_min
##             <dbl>   <dbl>    <dbl>
## 1              NA    24.9     22.5
```

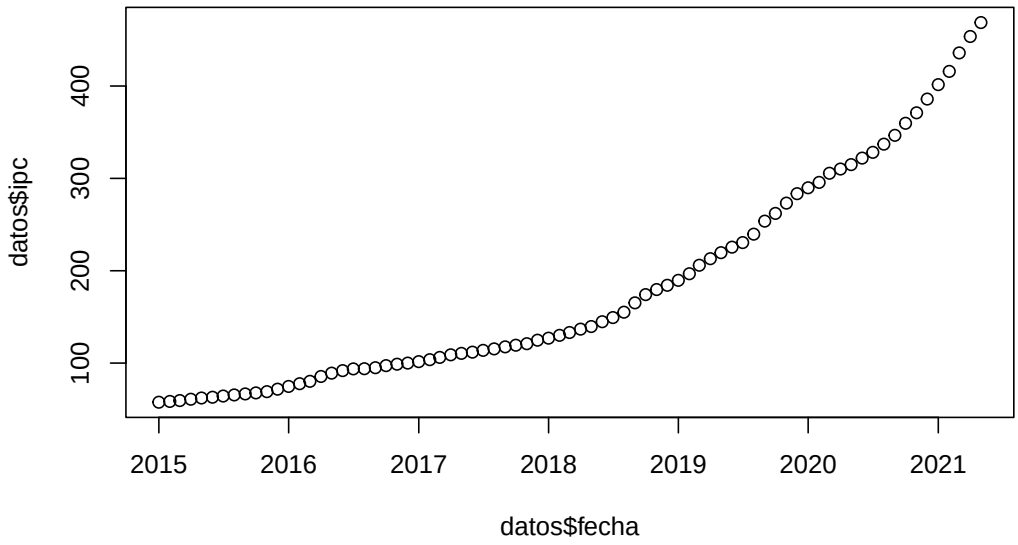
2.10 group_by, summarise

```
tab = bd %>%
  dplyr::select_if(is.numeric) %>%
  mutate(id = 1) %>%  # esta variable es solo para usar pivot
  pivot_longer(cols = -id, names_to = 'Variable', values_to = 'Value') %>%
  mutate(id = NULL) %>%
  group_by(Variable) %>%
  summarise(
    Obs = n(),
    Media = mean(Value, na.rm = T),
    Mediana = median(Value, na.rm = T),
    SD = sd(Value, na.rm = T),
    Min = min(Value, na.rm = T),
    Max = max(Value, na.rm = T)) %>%
  ungroup()
tab
```

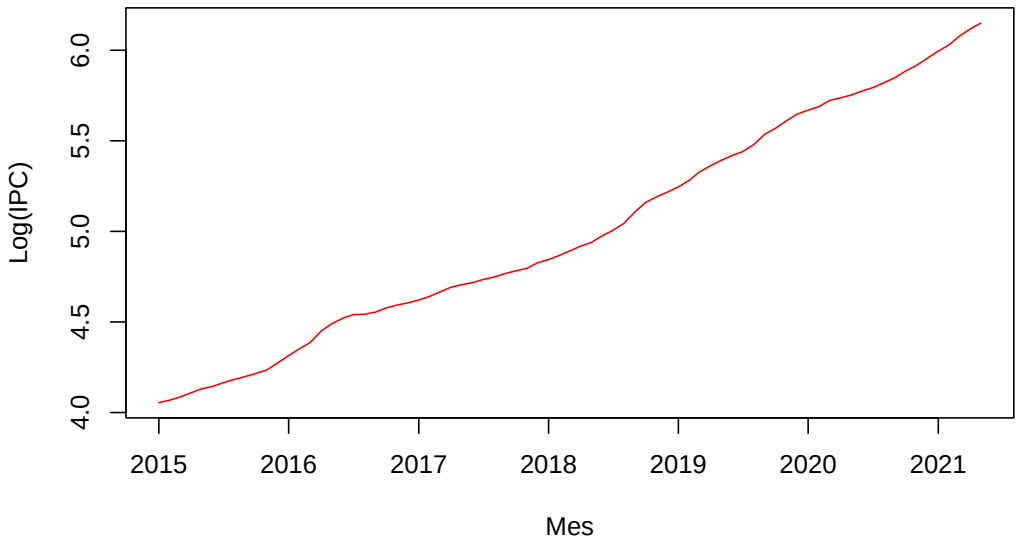
```
## # A tibble: 10 x 7
##    Variable         Obs     Media   Mediana        SD     Min      Max
##    <chr>          <int>     <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 credit_ps         60  8.97e+ 1    94.6      38.9      13.7   171.
##  2 credit_ps_mean    60  8.63e+ 1    93.4      40.5      14.7   143.
##  3 dev               60  5   e- 1     0.5       0.504     0       1
##  4 gdp_dem           60 -7.88e-13    78.4    1180.    -3511.   2211.
##  5 gdp_gr            60 -6.78e- 1     0.710     3.34    -10.8     4.31
##  6 gdp_pc2017        60  3.19e+ 4 32011.    11682.    14064.  46406.
##  7 inv               60  1.88e+ 1    17.8       3.19     13.4    24.9
##  8 obs               60  5.5 e+ 0     5.5       2.90      1      10
##  9 open              60  4.91e+ 1    56.2      15.7      22.5    72.2
## 10 year              60  2.02e+ 3  2016.        2.90   2011    2020
```

2.11 Gráficos

```
plot(datos$fecha, datos$ipc )
```



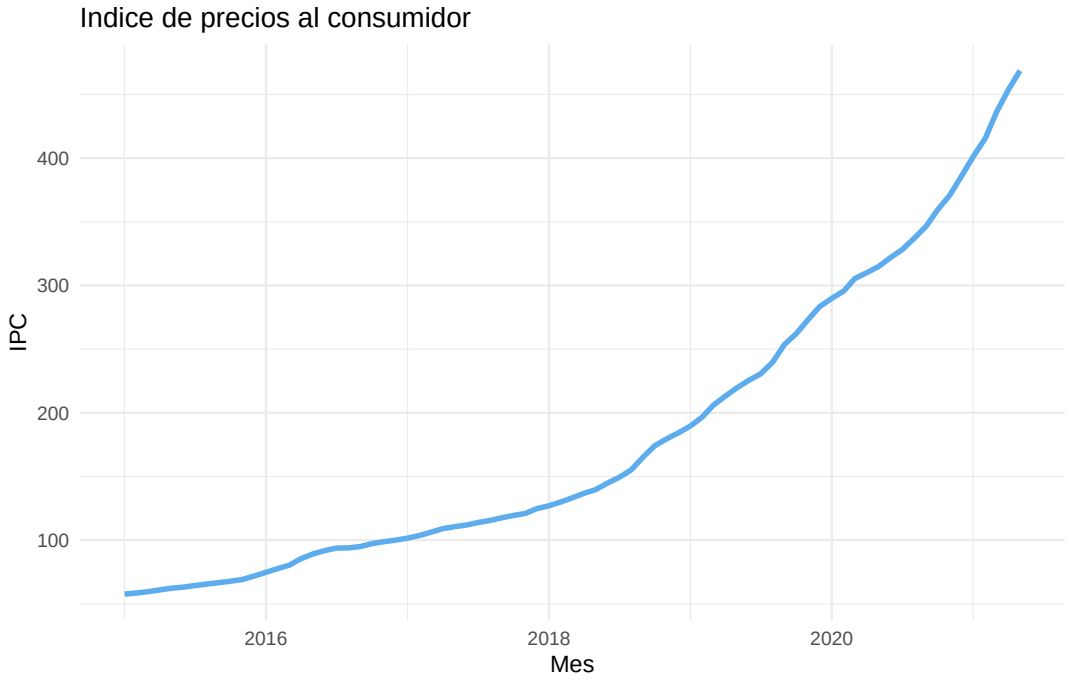

```
plot(datos$fecha, log(datos$ipc), type= 'l', col = 'red', xlab ='Mes', ylab ='Log(IPC)
```



2.12 GGPlot

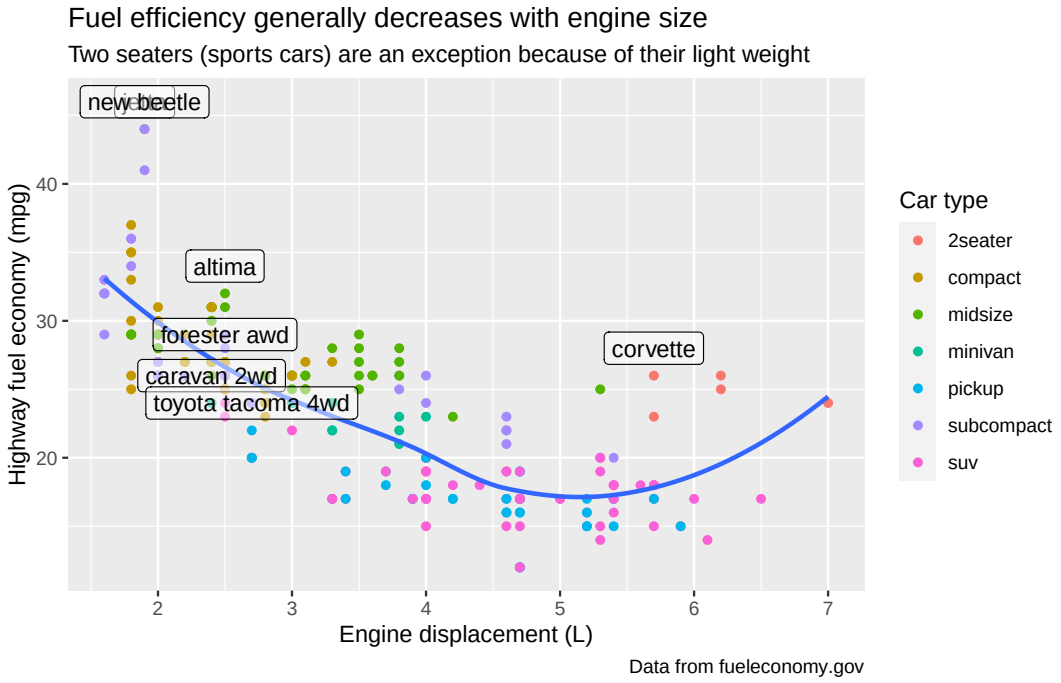
Grammar of Graphics. Ver más detalles en (Wickham et al., 2016).

```
#SINTAXIS
# ggplot(data = <DATA>) +
#   <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
# se agregan layers (piont, line, etc.)
# aes() "aesthetic" define la estética del gráfico
library(ggplot2)
datos1 = datos %>%
     select(fecha, ipc)

ggplot(datos1, aes(x=fecha, y=ipc)) +
  geom_line(color = 'steelblue2', size = 1.2) +
  theme_minimal() +
  labs(title="Indice de precios al consumidor", x="Mes", y="IPC") +
  theme(legend.position="none") +
  NULL
```




```
# Selecciona el mejor de cada clase de acuerdo al consumo en highway
best_in_class <- mpg %>%
  group_by(class) %>%
  filter(row_number(desc(hwy)) == 1)

g = ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class)) +
  geom_smooth(se = FALSE) +
  geom_label(aes(label = model), data = best_in_class, nudge_y = 2, alpha = 0.5) +
```

```
  labs(title = "Fuel efficiency generally decreases with engine size",
       subtitle = "Two seaters (sports cars) are an exception because of their light w
       x = "Engine displacement (L)",
       y = "Highway fuel economy (mpg)",
       colour = "Car type",
       caption = "Data from fueleconomy.gov") +
  NULL
g
```



2.13 Guardar un gráfico

```
work = "C:/Documentos/PoliticaII/work"
filesave = paste0(work,'ts.png')
ggsave(filesave, g, width=10, height=8)
```

Capítulo 3

Conceptos generales

3.1 Estimacion

Supongamos que se quiere estudiar la relación entre el gasto en publicidad a través de diversos canales como televisión, radio, diarios (*inputs*) y las ventas en distintos mercados (*output*).

$$Y = f(X) + \epsilon \tag{3.1}$$

donde f es una función desconocida de (X_1, X_2, X_3) y ϵ es un término de error aleatorio independiente de X con media igual a 0.

En esencia, el aprendizaje estadístico se refiere a un conjunto de enfoques para estimar f.

3.2 Prediccion

Supongamos que se dispone de datos de variables independientes por no de la dependiente, en ese caso, dado que el error en promedio es 0 podríamos predecir Y utilizando:

$$\hat{Y} = \hat{f}(X) \tag{3.2}$$

donde $\hat{f}$ representa nuestra estimación de f y $\hat{Y}$ representa la predicción de Y. En este contexto, $\hat{f}$ a menudo se trata como una **caja negra**, en el sentido que no importa la forma exacta de $\hat{f}$, siempre que produzca predicciones precisas de Y.

La precisión con la que $\hat{Y}$ se acerca a Y depende de dos cantidades, el error *reducible* y el *irreducible*. En general, $\hat{f}$ no será una estimación perfecta de f, y esta inexactitud introducirá un error que es reducible porque potencialmente podemos mejorar la precisión de $\hat{f}$ usando la técnica de aprendizaje estadístico más apropiada para estimar f. Sin embargo, si fuera posible estimar f exactamente de manera que la respuesta estimada

$\hat{Y} = f(X)$, nuestra predicción todavía tendría algún error dado que Y también es función de ϵ, que por definición, no se puede predecir usando X. Por lo tanto, la variabilidad asociada con ϵ también afecta la precisión de nuestras predicciones. Esto se conoce como el error irreducible, porque no importa qué tan bien estimemos f, no puede reducir el error introducido por ϵ.

El término de error ϵ puede contener variables no observables que son útiles para predecir Y y, por lo tanto, f no puede usarlos para su predicción.

$$E(Y - \hat{Y})^2 = E[f(X) + \epsilon - \hat{f}(X)]^2 \tag{3.3}$$

$$= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{Var(\epsilon)}_{\text{Irreducible}} \tag{3.4}$$

donde de $E(Y - \hat{Y})^2$ representa el promedio, o valor esperado, de la diferencia entre el valor predicho y el valor real de Y elevado al cuadrado (diferencia por exceso y defecto ponderan igual), y $Var(\epsilon)$ representa la varianza asociada al término de error ϵ.

El **foco** está en las técnicas para estimar f con el objetivo de minimizar el error reducible. Es importante tener en cuenta que la error irreducible siempre proporcionará un límite superior en la precisión de nuestra predicción para Y que en la práctica casi siempre es desconocido.

3.3 Metodos parametricos

Se realiza en dos etapas:

- Asumir una forma funcional (**modelo**, por ejemplo lineal)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p \tag{3.5}$$

- Estimar los parámetros (**método**, por ejemplo Mínimos Cuadrados Ordinarios - MCO-)

Si bien el problema se reduce a estimar $p + 1$ parámetros, la desventaja es que la forma funcional elegida puede diferir de la verdadera f.

3.4 Metodos no parametricos

No realizan supuestos sobre la forma funcional de f sino que tratan de buscar una estimación que se acerque lo más posible a los datos sin ser ni demasiado tosco ni demasiado ondulado.

Este enfoque puede tener una gran ventaja sobre los métodos paramétricos: al evitar el supuesto de una forma funcional particular para f, tiene el potencial para adaptarse con precisión a una gama más amplia de posibles formas para f. Cualquier enfoque paramétrico tiene la posibilidad de que la forma funcional utilizada para estimar f sea muy diferente de la verdadera f, en cuyo caso el resultado modelo no se ajustará bien a los datos. El costo es que se necesitan más datos para estimar.

3.5 Evaluacion de la precision del modelo

Ningún método domina al resto sobre todas las bases de datos posibles. En un *set* de datos en particular, un método específico puede funcionar mejor, pero algún otro método lo puede superar con otra base de datos. Por lo tanto, en cada caso se debe decidir qué método produce los mejores resultados.

3.5.1 Calidad del ajuste

Para evaluar el desempeño de un método de aprendizaje estadístico en una base de datos dada, se necesita alguna forma de medir qué tan bien sus predicciones coinciden con los datos observados. Es decir, se necesita cuantificar el grado en el cual el valor pronosticado para una observación dada está cerca de el verdadero valor de respuesta para esa observación. En el escenario de regresión, la medida más utilizada es el error medio cuadrático (EMC):

$$EMC = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \tag{3.6}$$

donde $\hat{f}(x_i)$ es la predicción que hace $\hat{f}$ sobre la observación i. El EMC será pequeño si las respuestas predichas están muy cerca de las respuestas verdaderas y será grande si para algunas observaciones difieren demasiado.

El EMC en (3.6) se calcula usando los **datos de entrenamiento** (*training*) que se usaron para estimar el modelo, por lo que debería denominarse con mayor precisión EMC de entrenamiento. Pero en general, no nos interesa realmente qué tan bien funciona el método sobre los datos de entrenamiento. Más bien, estamos interesados en la precisión de las predicciones que obtenemos cuando aplicamos nuestro método **datos de *test*** que no fueron visto antes (datos no utilizados para entrenar el modelo). Es decir, se busca elegir el método que produzca el menor EMC de *test*.

$$Prom(y_0 - \hat{f}(x_0))^2 \tag{3.7}$$

el error de predicción cuadrático promedio para estas observaciones de *test* (y_0, x_0).

¿Qué sucede si se elige en base al EMC de *training* (3.6)? No hay garantía de que el método con el EMC de entrenamiento más bajo también tenga el EMC de *test* más bajo.

El panel de la izquierda de la Figura 3.1 muestra la verdadera f dada por la curva negra. Las curvas naranja, azul y verde ilustran tres posibles estimaciones de f obtenidas utilizando métodos con distintos niveles de flexibilidad. La línea naranja es el ajuste de regresión lineal, que es relativamente inflexible. Las curvas azul y verde se produjeron usando *splines* con diferentes niveles de suavidad. Es claro que a medida que aumenta el nivel de flexibilidad, las curvas se ajustan mejor a los datos observados. La curva verde es la más flexible y coincide muy bien con los datos; sin embargo, se observa que se ajusta mal a la verdadera f (en negro) porque es demasiado ondulada. Cambiando el nivel de flexibilidad del *spline* se pueden producir ajustes diferentes para estos datos.

En el panel de la derecha de la Figura 3.1 la curva **gris** muestra el EMC de **entrenamiento** promedio en función de la flexibilidad, o más formalmente los grados de libertad (resume la flexibilidad de una curva), para una serie de *splines*. Los cuadrados naranja, azul y verde indican los EMC asociados con las curvas correspondientes en el panel izquierdo. El EMC de entrenamiento disminuye monótonamente a medida que aumenta la flexibilidad. Dado que la verdadera f es no lineal, el ajuste lineal naranja no es lo suficientemente flexible para estimar bien f. La curva verde tiene el EMC de entrenamiento más bajo de los tres métodos, ya que corresponde a la más flexible de las tres curvas.

En este ejemplo, se conoce la verdadera función f, por lo que también se puede calcular el EMC de *test* (en general f es desconocida, por lo que esto no es posible). El EMC de ***test*** se muestra usando la curva **roja** en el panel derecho de la Figura 3.1. Como con el EMC de entrenamiento, el EMC de *test* disminuye inicialmente a medida que el nivel de flexibilidad aumenta. Sin embargo, en algún momento el EMC de *test* se nivela y luego empieza a aumentar. En consecuencia, las curvas naranja y verde tienen un EMC de *test* alto. La curva azul minimiza el EMC de *test*, dado que visualmente parece estimar mejor f en el panel izquierdo. La línea discontinua horizontal indica $Var(\epsilon)$, el error irreducible en la ecuación de $E(Y - \hat{Y})^2$, que corresponde al menor alcanzable por el EMC de *test* entre todos los métodos posibles. Por lo tanto, el suavizado de *spline* representado por la curva azul está cerca del óptimo.

Importante. En el panel de la derecha de la Figura 3.1, a medida que la flexibilidad del método de aprendizaje aumenta, se observa una disminución monótona en el EMC de entrenamiento y una forma de U en el EMC de *test*. Esta es una propiedad fundamental de aprendizaje estadístico que se mantiene independientemente de la base de datos particular en cuestión e independientemente del método estadístico que se utilice.

Cuando un método dado produce un EMC de entrenamiento pequeño pero un EMC de *test* grande, se dice que está haciendo *overfitting*/sobreajustando los datos. Esto sucede porque nuestro aprendizaje estadístico está trabajando demasiado para encontrar patrones

en los datos de entrenamiento, y puede estar detectando algunos patrones que son causados por casualidad en lugar de por las verdaderas propiedades de la función desconocida f. ***Overfitting*** se refiere específicamente al caso en el que un modelo menos flexible podría haber producido un menor error de predicción en *test*.

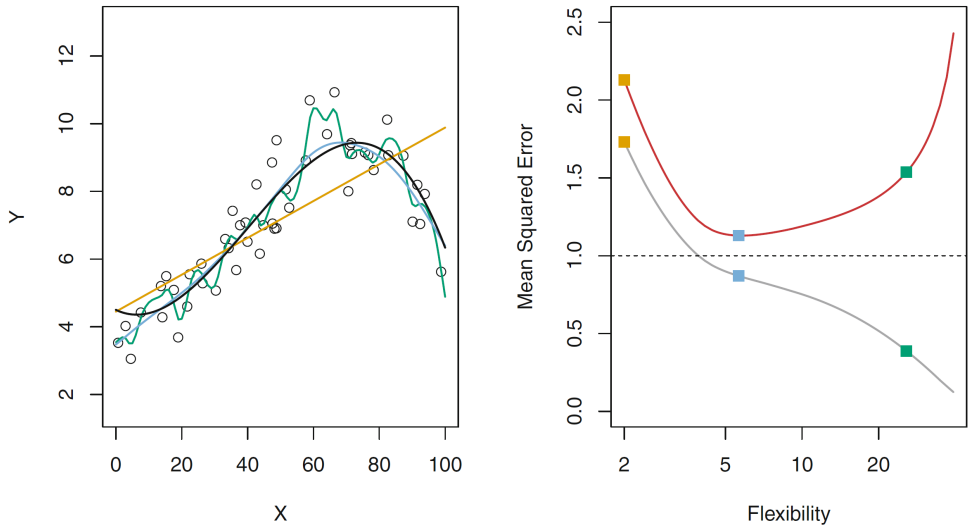


Figura 3.1: Datos en curva y ECM

La Figura 3.2 proporciona otro ejemplo en el que la verdadera f es aproximadamente lineal por lo que este tipo de modelos obtienen el menor EMC en *test* (curva roja en el panel derecho de la Figura 3.2).

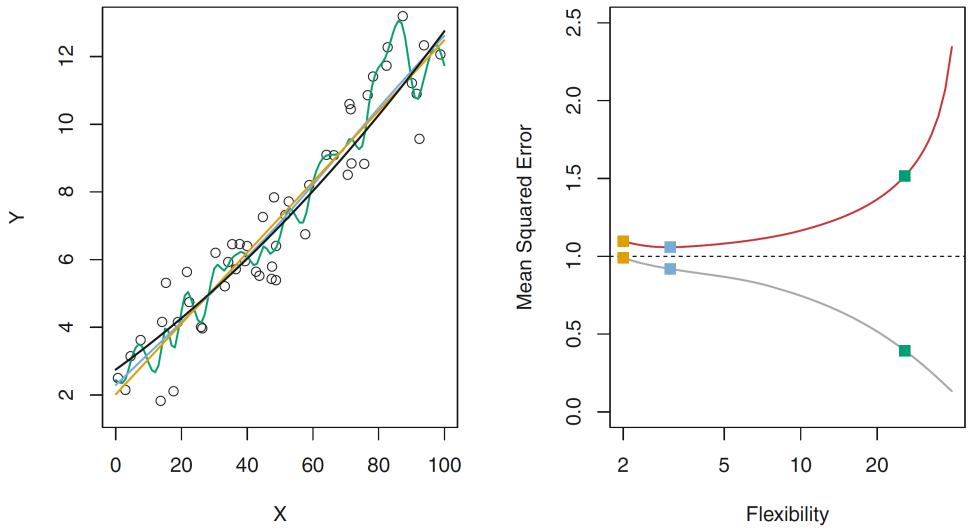


Figura 3.2: Datos lineales y EMC

3.5.2 Trade-off Sesgo-Varianza

La Figura 3.3 muestra el *trade-off* **Sesgo - Varianza** intuitivamente.

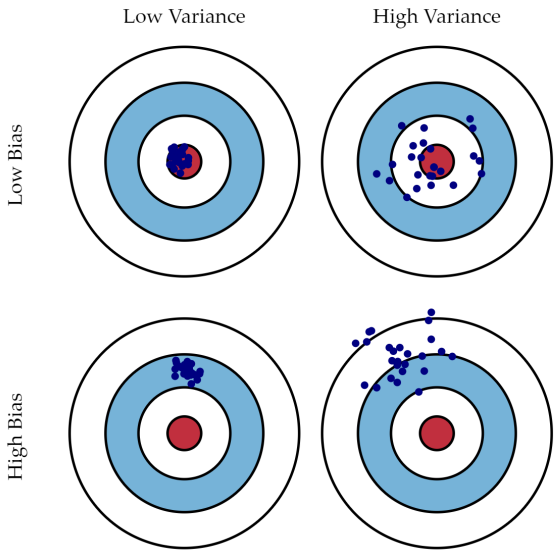


Figura 3.3: Estimacion y EMC

Fuente: Scott Fortmann-Roe

La forma de U observada en las curvas EMC de *test* es el resultado de dos propiedades que compiten en los métodos de aprendizaje estadístico. El EMC de *test* esperado, para un valor dado x_0, puede descomponerse en la suma de tres cantidades fundamentales: la varianza de $\hat{f}(x_0)$, el sesgo al cuadrado de $\hat{f}(x_0)$ y la varianza del error ϵ.

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Sesgo(\hat{f}(x_0))]^2 + Var(\epsilon) \tag{3.8}$$

donde $E(y_0 - \hat{f}(x_0))^2$ el valor esperado de EMC de *test* en x_0. Para minimizar el error de *test* esperado, se necesita seleccionar un método de aprendizaje estadístico que logre simultáneamente **baja varianza** y **bajo sesgo**.

La **varianza** se refiere al valor en que f cambiaría si se estimara utilizando una base de datos de entrenamiento diferente. **Sesgo** se refiere al error que se introduce al aproximar un problema de la vida real, que puede ser extremadamente complicado, por mucho modelo más simple. Como regla general, a medida que se utilizan métodos más flexibles, la varianza aumenta y el sesgo disminuye. La tasa relativa de cambio de estas dos cantidades determina si el EMC de *test* aumenta o disminuye.

Los dos paneles de la Figura 3.4 ilustran la Ecuación (3.8) para los ejemplos en Figuras 3.1 y 3.2. En cada caso, la curva sólida azul representa el cuadrado del sesgo, para diferentes niveles de flexibilidad, mientras que la curva naranja corresponde a la varianza. La línea discontinua horizontal representa $Var(\epsilon)$, el error irreducible. Finalmente, la curva roja, corresponde al EMC de *test*, es la suma de estas tres cantidades.

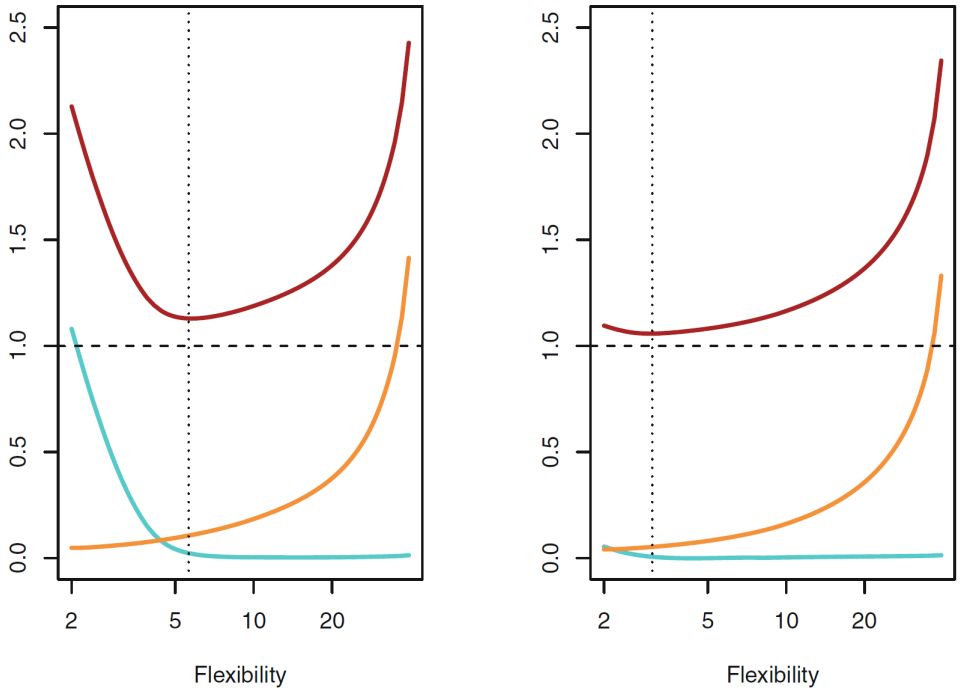


Figura 3.4: Estimacion y EMC

3.5.3 Clasificacion

Muchos de los conceptos del contexto de regresión, como el *trade-off* sesgo-varianza, se transfieren al entorno de clasificación donde ahora y_i es cualitativa. El enfoque más común para cuantificar la precisión de la estimación $\hat{f}$ es la **tasa de error** de entrenamiento, es decir, la proporción de errores que se cometen si aplicamos nuestra estimación $\hat{f}$ a las observaciones de entrenamiento.

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \tag{3.9}$$

Aquí $\hat{y}_i$ es la etiqueta de clase predicha para la i-ésima observación usando $\hat{f}$. Por lo tanto, $I(y_i \neq \hat{y}_i)$ es una variable indicadora que es igual a 0 si $y_i = \hat{y}_i$ ó 1 si $y_i \neq \hat{y}_i$, es decir, si la i-ésima observación fue clasificada correctamente o no por el método de clasificación.

La tasa de error de *test* asociada con un conjunto de observaciones de *test* de la forma (x_0, y_0) está dada por:

$$Prom(I(y_0 \neq \hat{y}_0)) \tag{3.10}$$

donde $\hat{y}_0$ es la etiqueta de clase predicha que resulta de aplicar el clasificador a la observación de *test* con predictor x_0. Un buen clasificador es aquel para el cual el error de *test* (3.10) es el más pequeño.

3.5.3.1 Clasificador de Bayes

Es posible mostrar que bajo penalidad simétrica[1] la tasa de error de *test* postulada en (3.10) se minimiza, en promedio, por un clasificador muy simple que asigna cada observación a la clase más probable, dados sus valores predictores. En otras palabras, se debería asignar una observación de *test* con vector predictor x_0 a la clase j para la cual (3.11) es mayor.

$$Pr(Y = j \mid X = x_0) \tag{3.11}$$

Es decir, en un problema donde sólo hay dos categorías el clasificador de Bayes predice la clase 1 si $Pr(Y = 1 \mid X = x_0) > 0.5$ y la clase 0 en caso contrario.

3.5.4 Matriz de confusion

		Observado	
		0	1
Predicción	0	VN	FN
(decision)	1	FP	VP

VN: Verdadero Negativo; FN: Falso Negativo; FP: Falso Positivo; VP: Verdadero Positivo

Métricas para comparar modelos. La **precisión** (*accuracy*) es la cantidad de predicciones correctas.

$$\text{Precision} = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.12}$$

[1] ¿Útil para probabilidad de *default*?

Capítulo 4

Arboles de decision

Los métodos basados en árboles para regresión y clasificación estratifican o segmentan el espacio predictor en varias regiones. Para hacer una predicción para una observación dada normalmente utiliza el valor de respuesta promedio de las observaciones de la base de entrenamiento en la región a la que pertenece. En el caso de clasificación se asigna a la categoría mayoritaria dentro del nodo terminal.

4.1 *Classification and Regression Tree* (CART)

En el caso de árboles de regresión, si Y es la respuesta y X_1 y X_2 los *inputs* se parte el espacio (X_1, X_2) en dos regiones, en base a una sola variable (partición horizontal o vertical). Dentro de cada región proponemos como predicción la media muestral de Y en cada región.

Se busca elegir la variable y el punto de partición de manera óptima (mejor ajuste global). Es computacionalmente inviable considerar cada posible partición del espacio de atributos en J regiones. Por lo tanto, toma un enfoque *top-down greedy* que se conoce como división binaria recursiva. El enfoque es *top-down* porque comienza en la parte superior del árbol (en cuyo punto todas las observaciones pertenecen a una sola región) y luego divide sucesivamente el espacio predictor; cada división se indica a través de dos nuevas ramas más abajo en el árbol. Es *greedy* porque en cada paso del proceso de construcción del árbol, la mejor división se hace en ese paso en particular, en lugar de mirar hacia adelante y elegir una división que conducirá a un mejor árbol en algún paso futuro.

El panel izquierdo de la Figura 4.1 muestra un árbol de regresión para predecir el logaritmo del salario (en miles de dólares) de un jugador de béisbol, basado en la cantidad de años que ha jugado en las ligas mayores y la cantidad de *hits* que hizo en el año anterior. En un un nodo interno dado, la etiqueta (de la forma $X_j < t_k$) indica la rama izquierda que sale de esa división, y la rama de la derecha corresponde a $X_j \geq t_k$. Por ejemplo, la división en la parte superior del árbol da como resultado dos ramas grandes. El la rama izquierda corresponde a `Years < 4,5`, y la rama derecha corresponde a `Years >= 4,5`.[1]

[1] Al estar arriba, `Years` es la variable más importante para explicar el salario.

El árbol tiene dos nodos internos y tres nodos terminales u hojas. El número en cada hoja es la media de la variable de respuesta de las observaciones que caen allí. Por ejemplo, la predicción para el nodo terminal de la izquierda es $e^{5,107} \times 1.000 = \$165.174$. El panel derecho la Figura 4.1 muestra las regiones en función de `Years` y `Hits`.

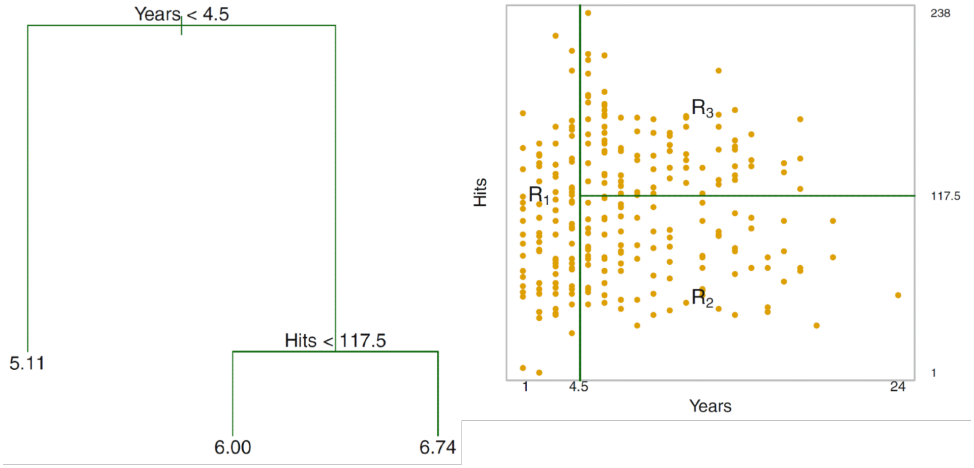


Figura 4.1: Arbol de regresión

Notar:

- Cada región tiene su propio modelo.
- Ciertas variables importan en determinadas regiones y no en otras (*Hits*).

Dado Y y X un vector de p variables con n observaciones el algoritmo busca determinar cuál variable usar para la partición y que punto de esa variable usar para la partición. Si j es la variable de partición y el punto de partición es s, se definen los siguientes semiplanos:

$$R_1(j,s) = X \mid X_j < s$$
$$R_2(j,s) = X \mid X_j \geq s$$

Se trata de buscar la variable de partición X_j y el punto de partición s que resuelvan (minimizar el EMC en cada región):

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{4.1}$$

Donde $\hat{y}_{R_1}$ y $\hat{y}_{R_2}$ es el promedio de la respuesta en las regiones 1 y 2, respectivamente. Para cada variable y punto de partición, la minimización interna se corresponde con la **media** dentro de cada región.[2]

[2]Recordar que si se quiere predecir una variable aleatoria Y con una constante m el mejor predictor es su esperanza, es decir, $m = E(Y)$.

¿Cuándo parar de realizar divisiones?

Un árbol demasiado extenso sobreajusta (*overfit*) los datos. Pero dado que el proceso es secuencial y cada corte no mira lo que puede suceder después, si se detiene el proceso demasiado pronto se puede perder un "gran" corte más abajo. *Prunning*: ajustar un árbol grande y luego podarlo (*prune*) usando un criterio de *cost-complexity*.

Classification tree

Un árbol de clasificación es muy similar a un árbol de regresión, excepto que se utiliza para predecir una respuesta cualitativa en lugar de una cuantitativa. Recordar que para un árbol de regresión, la respuesta predicha para una observación esta dada por la respuesta media de las observaciones de entrenamiento que pertenecen al mismo nodo terminal. En contraste, para un árbol de clasificación, predice que cada observación pertenece a la clase que ocurre más comúnmente en las observaciones de entrenamiento en la región a la que pertenece. Se basa en el error de clasificación o índice de Gini (pureza), análogo a EMC en un árbol de regresión.

4.2 Bagging

Ventajas y desventajas de $CART$:

- Forma inteligente de representar no linealidades.
- Arriba quedan las variables más relevantes entonces es fácil de comunicar. Reproduce proceso decisorio humano.
- Si la estructura es lineal, $CART$ no anda bien.
- Poco robusto, variaciones en los datos modifican el resultado.

Un método de *ensemble* es un enfoque que combina muchos modelos simples en uno único y potencialmente muy poderoso. Los modelos simples se conocen como modelos de aprendizaje débil, ya que por sí mismos pueden generar predicciones mediocres.

Una posible solución es el *bootstrap aggregation* que consiste en tomar como predicción el promedio de las predicciones *bootstrap*.[3]

$$\hat{f}_{bag} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \tag{4.2}$$

Esta idea se basa en que la varianza del promedio es menor que la de una predicción sola. Bajo independencia si $V(x) = \sigma^2$ entonces $V(\overline{x}) = \frac{\sigma^2}{n}$. Pero existe el **problema** que si hay un predictor fuerte (siempre va a ser seleccionado primero), los distintos árboles son muy similares entre sí por lo que habrá alta correlación.

[3]Es decir, muestreo con reemplazo.

4.3 Random Forest

Busca bajar la correlación entre los árboles del *bootstrap*. Al igual que en *bagging*, construye una serie de árboles de decisión en muestras de entrenamiento *bootstrap*. Pero al construir estos árboles de decisión, cada vez que se considera una división en un árbol, se elige como candidatos de división una muestra aleatoria de m predictores del conjunto completo de p predictores ($m < p$).

Capítulo 5

Trabajo Practico

5.1 Reglas del Trabajo practico

1. **Integrantes:** máximo 3 por grupo.
2. **Extensión:** máximo 8 carillas (hoja A4, 12pts, etc.). La página 9 **no se corrige.**
3. **Copia o plagio:** trabajo desaprobado, grupo fuera del régimen de promoción.
4. **Redacción:** Formal.
5. **Presentación:** tablas/cuadros bien descriptas y ordenadas.
6. **Bases de datos:** a definir diferenciando por grupos.
7. Sugerencias bibliográficas:

    - Fortalezas y debilidades de la evaluación de créditos con técnicas de *machine learning* (Bazarbash, 2019).
    - *Performance* predictiva (Frost et al., 2019).
    - *Performance* predictiva (Petropoulos et al., 2018).

5.2 Enunciado del Trabajo Practico

En base a lo desarrollado en las clases teóricas se busca que elaboren un modelo de *scoring* que permita discriminar entre buenos y malos deudores.

1. Realizar una revisión de la literatura teórica y empírica para elaborar una sección que describa:

- ¿Por qué es importante el problema a analizar?

- Ventajas y desventajas de enfoques tradicionales vs. *machine learning*.
- Los principales resultados encontrados.

2. Presentar, describir y analizar los datos utilizados.
3. Presentar e interpretar los principales resultados de un modelo *logit* en comparación con técnicas de árboles.
4. Elaborar conclusiones de política.

5.3 Aplicacion practica

La irrupción de las firmas *BigTech* en la provisión de crédito está modificando la estructura del sistema financiero. Si bien la actividad principal de estas compañías es la provisión de servicios digitales como el *e-commerce* y servicios de pago paulatinamente han ido incorporando otros productos como la provisión de crédito, seguros, inversiones y ahorro.

El modelo de negocios de las *BigTech* difiere del modelo de las entidades financieras tradicionales principalmente por dos factores distintivos: efectos de red (generados por las plataformas de *e-commerce*, aplicaciones de mensajería y redes sociales); el uso de la tecnología (inteligencia artificial utilizando *big data*).

La utilización de nuevas técnicas de análisis y fuentes de datos alternativos brindan a las empresas tecnológicas una ventaja informativa para la evaluación de deudores respecto de las entidades financieras, que utilizan métodos econométricos convencionales (ej. estimaciones *logit*) menos flexibles para capturar la información contenida en grandes volúmenes de datos.

En esta sección se utiliza una base de datos bancaria para predecir la probabilidad de *default* con distintas metodologías, un modelo *logit*, un árbol simple y un *random forest*, para comparar las capacidades predictivas.

Primero se limpia la memoria y se cargan las librerías que vamos a utilizar.

```
# Limpiar memoria
rm(list=ls())
gc()
```

```
##           used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells 2345879 125.3    3988661 213.1  3988661 213.1
## Vcells 4070464  31.1    8388608  64.0  6451855  49.3
```

```
# Librerias
library(tidyverse)
library(rsample)
```

```
library(yardstick)
library(rpart)
library(rpart.plot)
library(ranger)
library(caret)
```

Se cargan los datos desde un archivo separado por comas.

```
# Cargar datos
score_data_raw <- read_csv('./data/CreditScore1.csv')

score_data_tbl <- score_data_raw %>%
  dplyr::select(-id) %>% drop_na()
```

Se realiza una inspección inicial de la base de datos.

```
head(score_data_tbl)
```

```
## # A tibble: 6 x 10
##   default personal_total  edad nro_atraso3059 gastos_ingreso ingreso
##     <dbl>          <dbl> <dbl>          <dbl>          <dbl>   <dbl>
## 1       1          0.766    45              2         0.803     9120
## 2       0          0.957    40              0         0.122     2600
## 3       0          0.658    38              1         0.0851    3042
## 4       0          0.234    30              0         0.0360    3300
## 5       0          0.907    49              1         0.0249   63588
## 6       0          0.213    74              0         0.376     3500
## # ... with 4 more variables: lineas_credito <dbl>, nro_atraso90 <dbl>,
## #   nro_hipoteca <dbl>, familia <dbl>
```

```
glimpse(score_data_tbl)
```

```
## Rows: 120,269
## Columns: 10
## $ default        <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1~
## $ personal_total <dbl> 0.76612663, 0.95715100, 0.65818012, 0.23380977, 0.90723~
## $ edad           <dbl> 45, 40, 38, 30, 49, 74, 39, 57, 30, 51, 46, 40, 76, 64,~
## $ nro_atraso3059 <dbl> 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0~
## $ gastos_ingreso <dbl> 0.80298215, 0.12187620, 0.08511338, 0.03604968, 0.02492~
## $ ingreso        <dbl> 9120, 2600, 3042, 3300, 63588, 3500, 3500, 23684, 2500,~
## $ lineas_credito <dbl> 13, 4, 2, 5, 7, 3, 8, 9, 5, 7, 13, 9, 6, 7, 7, 7, 2, 10~
## $ nro_atraso90   <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0~
## $ nro_hipoteca   <dbl> 6, 0, 0, 0, 1, 1, 0, 4, 0, 2, 2, 1, 1, 1, 0, 1, 0, 2, 1~
## $ familia        <dbl> 2, 1, 0, 0, 0, 1, 0, 2, 0, 2, 2, 2, 0, 2, 0, 2, 0, 0, 2~
```

```
summary(score_data_tbl)
```

```
##     default        personal_total          edad        nro_atraso3059
##  Min.   :0.00000   Min.   :    0.00   Min.   :  0.00   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.:    0.04   1st Qu.: 40.00   1st Qu.: 0.0000
##  Median :0.00000   Median :    0.18   Median : 51.00   Median : 0.0000
##  Mean   :0.06949   Mean   :    5.90   Mean   : 51.29   Mean   : 0.3818
##  3rd Qu.:0.00000   3rd Qu.:    0.58   3rd Qu.: 61.00   3rd Qu.: 0.0000
##  Max.   :1.00000   Max.   :50708.00   Max.   :103.00   Max.   :98.0000
##  gastos_ingreso        ingreso        lineas_credito    nro_atraso90
##  Min.   :    0.00   Min.   :      0   Min.   : 0.000   Min.   : 0.0000
##  1st Qu.:    0.14   1st Qu.:   3400   1st Qu.: 5.000   1st Qu.: 0.0000
##  Median :    0.30   Median :   5400   Median : 8.000   Median : 0.0000
##  Mean   :   26.60   Mean   :   6670   Mean   : 8.758   Mean   : 0.2119
##  3rd Qu.:    0.48   3rd Qu.:   8249   3rd Qu.:11.000   3rd Qu.: 0.0000
##  Max.   :61106.50   Max.   :3008750   Max.   :58.000   Max.   :98.0000
##   nro_hipoteca       familia
##  Min.   : 0.000   Min.   : 0.0000
##  1st Qu.: 0.000   1st Qu.: 0.0000
##  Median : 1.000   Median : 0.0000
##  Mean   : 1.055   Mean   : 0.8518
##  3rd Qu.: 2.000   3rd Qu.: 2.0000
##  Max.   :54.000   Max.   :20.0000
```

```
unique(score_data_tbl$familia)
```

```
##  [1]  2  1  0  3  4  5  6  8  7 20 10  9 13
```

```
table(score_data_tbl$default)
```

```
##
##      0      1
## 111912   8357
```

```
# Porcentaje de positivos
8357 / (8357 + 111912)
```

```
## [1] 0.0694859
```

Luego, se calculan algunas estadíticas descriptivas.

```
stat = score_data_tbl %>%
  dplyr::select_if(is.numeric) %>%
  pivot_longer(everything(), names_to = 'Variable', values_to = 'Value') %>%
  group_by(Variable) %>%
  summarise(
    Obs = n(),
    Media = mean(Value, na.rm = T),
    Mediana = median(Value, na.rm = T),
    SD = sd(Value, na.rm = T),
    Min = min(Value, na.rm = T),
    Max = max(Value, na.rm = T)) %>%
  ungroup()
stat
```

```
## # A tibble: 10 x 7
##    Variable          Obs     Media  Mediana        SD   Min      Max
##    <chr>           <int>     <dbl>    <dbl>     <dbl> <dbl>    <dbl>
##  1 default        120269    0.0695    0         0.254     0       1
##  2 edad           120269   51.3      51        14.4       0     103
##  3 familia        120269    0.852     0         1.15      0      20
##  4 gastos_ingreso 120269   26.6       0.296   424.        0   61106.
##  5 ingreso        120269 6670.     5400     14385.        0 3008750
##  6 lineas_credito 120269    8.76      8         5.17      0      58
##  7 nro_atraso3059 120269    0.382     0         3.50      0      98
##  8 nro_atraso90   120269    0.212     0         3.47      0      98
##  9 nro_hipoteca   120269    1.05      1         1.15      0      54
## 10 personal_total 120269    5.90      0.177   257.        0   50708
```

Se procede a realizar el *feature engineering* o creación de variables...notar que la capacidad de clasificación depende de los atributos disponibles y los valores de los hiperpárametros.[1]

```
# Transformacion
score_data_tbl = score_data_tbl %>%
                 mutate(
                    default = factor(default),
                    ingreso = log(1+ingreso),
                    familia_bin = case_when(familia == 0 ~ 1,
                                            familia == 1 ~ 2,
                                            familia >= 2&familia < 5 ~ 3,
                                            familia >= 5 ~ 4))
score_data_tbl = score_data_tbl %>% dplyr::select(-familia)
```

Se divide la muestra en 80% para entrenamiento y 20% para *test*.

[1] Es importante señalar que la estadística descriptiva sugiere realizar más modificaciones.

```
# Train / Test split
set.seed(1234)
train_test_split <- initial_split(score_data_tbl, prop = 0.8)
train_test_split
```

```
## <Analysis/Assess/Total>
## <96215/24054/120269>
```

```
train_tbl <- training(train_test_split)
test_tbl  <- testing(train_test_split)
```

Se definen dos objetos para utilizar más abajo.

```
# Formula
formula  <-  formula(default ~ .)

# Y observado a 0/1 para confusionMatrix
obs =  factor(test_tbl$default)
```

Se estima el modelo lineal (*default* vs. resto de variables).

```
lm.mod = lm(as.numeric(default)~., data = train_tbl)
summary(lm.mod)
```

```
##
## Call:
## lm(formula = as.numeric(default) ~ ., data = train_tbl)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90421 -0.08651 -0.06467 -0.03990  1.18753
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.157e+00  6.296e-03 183.764  < 2e-16 ***
## personal_total -3.760e-06  3.623e-06  -1.038 0.299402
## edad           -1.585e-03  5.894e-05 -26.887  < 2e-16 ***
## nro_atraso3059  2.144e-02  1.059e-03  20.242  < 2e-16 ***
## gastos_ingreso -5.645e-06  1.962e-06  -2.877 0.004010 **
## ingreso        -2.315e-03  6.831e-04  -3.389 0.000701 ***
## lineas_credito -6.870e-04  1.776e-04  -3.868 0.000110 ***
## nro_atraso90   -1.354e-02  1.071e-03 -12.651  < 2e-16 ***
```

```
## nro_hipoteca    1.685e-03  7.908e-04   2.131 0.033066 *
## familia_bin     7.086e-03  9.798e-04   7.232 4.78e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2504 on 96205 degrees of freedom
## Multiple R-squared:  0.02673,    Adjusted R-squared:  0.02664
## F-statistic: 293.6 on 9 and 96205 DF,  p-value: < 2.2e-16
```

```
t(broom::glance(lm.mod))
```

```
##                         [,1]
## r.squared       2.673354e-02
## adj.r.squared   2.664249e-02
## sigma           2.504424e-01
## statistic       2.936161e+02
## p.value         0.000000e+00
## df              9.000000e+00
## logLik         -3.305975e+03
## AIC             6.633949e+03
## BIC             6.738167e+03
## deviance        6.034112e+03
## df.residual     9.620500e+04
## nobs            9.621500e+04
```

Se estima el modelo logit.[2]

```
glm.mod <- glm(formula, data = train_tbl, family = binomial)
summary(glm.mod)
```

```
##
## Call:
## glm(formula = formula, family = binomial, data = train_tbl)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3117  -0.4195  -0.3503  -0.2850   3.6748
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.142e+00  1.007e-01 -11.341  < 2e-16 ***
## personal_total -8.713e-05  9.148e-05  -0.953 0.340843
```

[2] Qué sucede si para definir la clase se modifica el umbral $p > 0.5$?

```
## edad           -2.770e-02  1.027e-03 -26.979  < 2e-16 ***
## nro_atraso3059  2.548e-01  1.393e-02  18.288  < 2e-16 ***
## gastos_ingreso -2.669e-04  7.694e-05  -3.469 0.000523 ***
## ingreso        -4.414e-02  1.181e-02  -3.738 0.000186 ***
## lineas_credito -9.410e-03  3.021e-03  -3.114 0.001843 **
## nro_atraso90   -2.120e-01  1.415e-02 -14.982  < 2e-16 ***
## nro_hipoteca    4.254e-02  1.204e-02   3.533 0.000410 ***
## familia_bin     1.265e-01  1.478e-02   8.555  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 48424  on 96214  degrees of freedom
## Residual deviance: 46551  on 96205  degrees of freedom
## AIC: 46571
##
## Number of Fisher Scoring iterations: 6
```

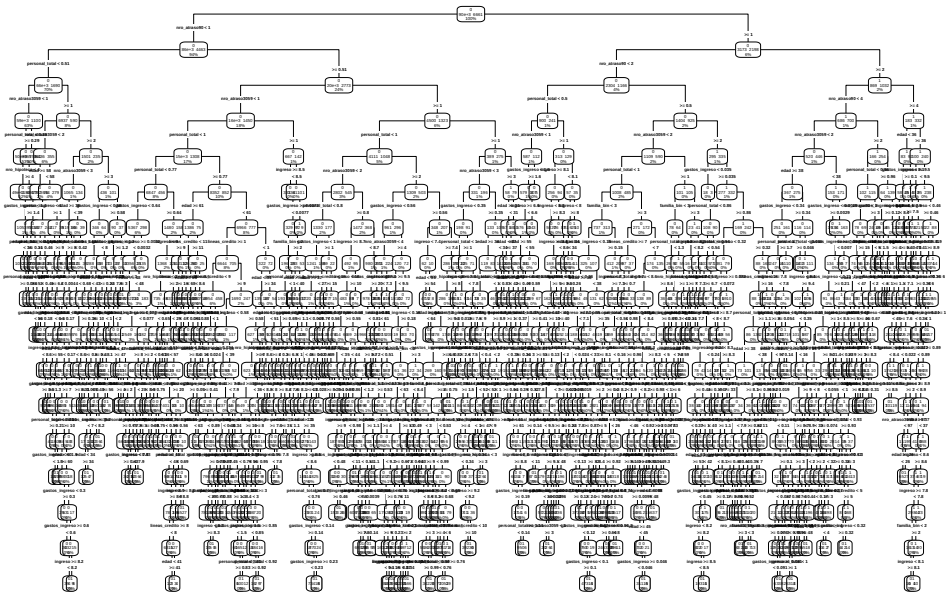
```
# Efectos marginales ver:
#library(mfx)
#logitmfx(formula, data)

glm.probs <- predict(glm.mod, test_tbl,  type = 'response')
glm.class <- factor(ifelse(glm.probs > 0.5, 1, 0))

cm_logit = confusionMatrix(glm.class, obs, positive = '1')
```

Se estima un árbol simple.

```
set.seed(4321)
rpart.mod = rpart(formula,
                  data = train_tbl,
                  control = rpart.control(minsplit = 20,
                                          minbucket = 6,
                                          cp = 0,
                                          xval = 0,
                                          maxdepth = 16))
names(rpart.mod)
```

```
##  [1] "frame"              "where"              "call"
##  [4] "terms"              "cptable"            "method"
##  [7] "parms"              "control"            "functions"
## [10] "numresp"            "splits"             "variable.importance"
## [13] "y"                  "ordered"
```

```
rpart.prob = predict(rpart.mod, test_tbl)
rpart.class = factor(ifelse(rpart.prob[, '1']>0.5, 1, 0))

cm_rpart = confusionMatrix(rpart.class, obs, positive = '1')

prp(rpart.mod, extra=101, digits=2, branch=1, type=4, varlen=0, faclen=0)
```



```
rpartVarImp = as_tibble_row(rpart.mod$variable.importance) %>%
 mutate(id = 1) %>%
 pivot_longer(cols = -id, names_to = 'Variable', values_to = 'Value') %>%
 mutate(id = NULL) %>% arrange(desc(Value))
rpartVarImp
```

```
## # A tibble: 9 x 2
##   Variable        Value
##   <chr>           <dbl>
## 1 nro_atraso90   1451.
## 2 personal_total  748.
## 3 nro_atraso3059  436.
## 4 gastos_ingreso  428.
## 5 ingreso         395.
## 6 edad            265.
## 7 lineas_credito  226.
## 8 nro_hipoteca    112.
## 9 familia_bin      69.5
```

Se estima un *random forest*.

```
set.seed(1234)
ranger.mod  = ranger(formula,
                     data = train_tbl,
                     probability = TRUE,
                     num.trees = 300,
                     min.node.size = 15,
                     mtry = 3,
                     splitrule ='gini',
                     importance ='impurity')
names(rpart.mod)
```

```
##  [1] "frame"              "where"              "call"
##  [4] "terms"              "cptable"            "method"
##  [7] "parms"              "control"            "functions"
## [10] "numresp"            "splits"             "variable.importance"
## [13] "y"                  "ordered"
```

```
ranger.prob = predict(ranger.mod, test_tbl)
ranger.class = factor(ifelse(ranger.prob$predictions[, '1']>0.5, 1, 0))

cm_ranger = confusionMatrix(ranger.class, obs, positive = '1')

rangerVarImp = as_tibble_row(ranger.mod$variable.importance) %>%
               mutate(id = 1) %>%
               pivot_longer(cols = -id, names_to = 'Variable', values_to = 'Value') %>
               mutate(id = NULL) %>%
               arrange(desc(Value))
rangerVarImp
```

```
## # A tibble: 9 x 2
##   Variable       Value
##   <chr>          <dbl>
## 1 personal_total 1741.
## 2 gastos_ingreso 1330.
## 3 ingreso        1186.
## 4 nro_atraso90   1149.
## 5 edad            839.
## 6 nro_atraso3059  611.
## 7 lineas_credito  607.
## 8 nro_hipoteca    237.
## 9 familia_bin     204.
```

Se presentan los resultados (no se analizan...) en tabla resumen.

```
tab_acc = tibble(logit  = cm_logit$overall[['Accuracy']],
                 rpart  = cm_rpart$overall[['Accuracy']],
                 ranger = cm_ranger$overall[['Accuracy']])

tab_acc = tab_acc %>% pivot_longer(everything(), names_to='Modelo', values_to='Accuracy'

tab_acc %>% arrange(desc(Accuracy))
```

```
## # A tibble: 3 x 2
##   Modelo Accuracy
##   <chr>     <dbl>
## 1 ranger    0.933
## 2 logit     0.930
## 3 rpart     0.926
```

Bibliografía

Bazarbash, M. (2019). Fintech in financial inclusion machine learning applications in assessing credit risk. *IMF Working Paper*, (109).

Frost, J., Gambacorta, L., Huang, Y., Shin, H. S., and Zbinden, P. (2019). Bigtech and the changing structure of financial intermediation. *BIS Working Papers*, (779).

Petropoulos, A., Siakoulis, V., Stavroulakis, E., and Klamargias, A. (2018). A robust machine learning approach for credit risk analysis of large loan-level datasets using deep learning and extreme gradient boosting. *Irving Fisher Committee*.

Wickham, H. and Grolemund, G. (2017). *R for Data Science*.

Wickham, H., Navarro, D., and Pedersen, T. L. (2016). *ggplot2: Elegant Graphics for Data Analysis*.