

Análisis de Datos

4º GITI

Especialidad Matemáticas y Organización

Primera parte: Descriptiva Univariante

- 1 Datos
- 2 Frecuencias
- 3 Histogramas
- 4 Media y desviación típica
- 5 Boxplot

Ejemplo: salario profesores de universidad

```
dat = read.table("data/salary.txt", header=TRUE)
print(dat[1:20,]) # muestro los 20 primeros
```

```
##      degree rank   sex year ysdeg salary
## 1    Masters Prof Male  25    35 36350
## 2    Masters Prof Male  13    22 35350
## 3    Masters Prof Male  10    23 28200
## 4    Masters Prof Female 7    27 26775
## 5      PhD Prof Male  19    30 33696
## 6    Masters Prof Male  16    21 28516
## 7      PhD Prof Female 0    32 24900
## 8    Masters Prof Male  16    18 31909
## 9      PhD Prof Male  13    30 31850
## 10     PhD Prof Male  13    31 32850
## 11   Masters Prof Male  12    22 27025
## 12   Masters Assoc Male  15    19 24750
## 13   Masters Prof Male  9    17 28200
## 14      PhD Assoc Male  9    27 23712
## 15   Masters Prof Male  9    24 25748
## 16   Masters Prof Male  7    15 29342
## 17   Masters Prof Male  13    20 31114
## 18      PhD Assoc Male  11    14 24742
## 19      PhD Assoc Male  10    15 22906
## 20      PhD Prof Male  6    21 24450
```

Cuantitativos

- Continuos: Magnitudes que pueden medirse (estatura de una persona, salario)
- Discretos: Cantidades que se pueden contar (número de hijos)

Cualitativos

- Ordinales: Nominales pero tienen orden (nivel de estudio, categoría de profesor)
- No ordinales: Nominales sin orden (género)

5

Descriptiva de 'rank'

```
t1 = table(dat$rank)

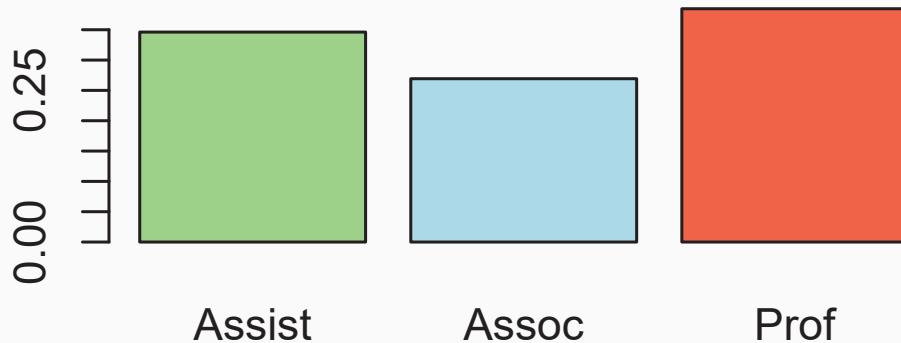
addmargins(
  cbind( Fr_Abs = t1, Fr_Rel = prop.table(t1)*100,
        1 ) )

##           Fr_Abs Fr_Rel
## Assist      18   34.6
## Assoc       14   26.9
## Prof        20   38.5
## Sum         52 100.0
```

6

Diagrama de barras

```
barplot( prop.table(t1),  
        col=c("lightgreen","lightblue","tomato"))
```



7

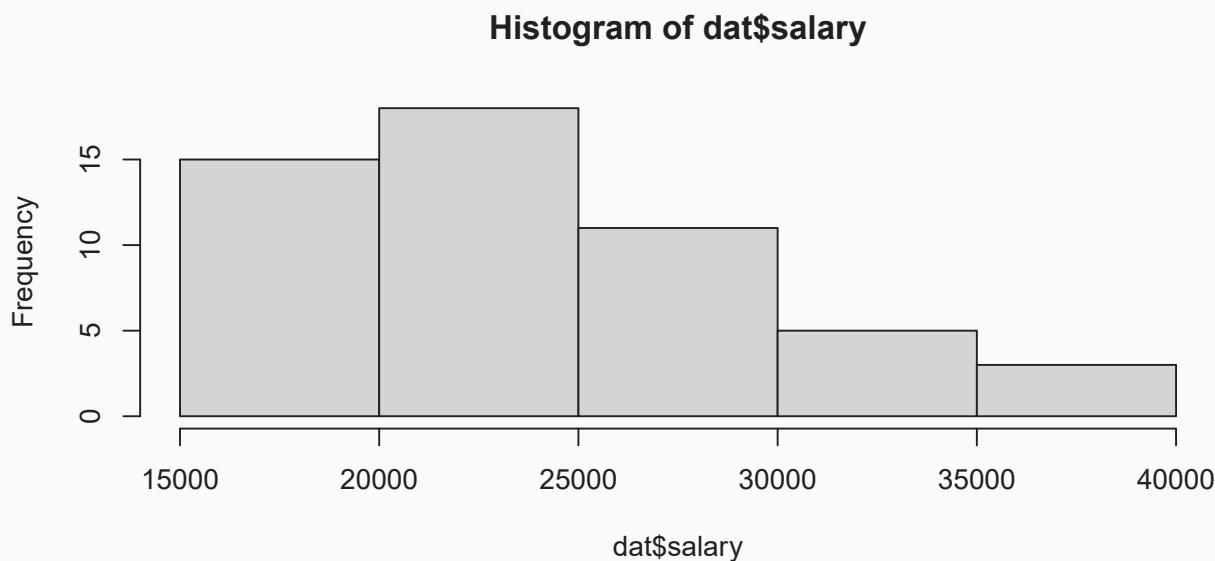
Distribución de frecuencias variable continua (salary)

Clase	Infer	Super	FrAbs	FrRel
1	15000	20000	15	28.85
2	20000	25000	18	34.62
3	25000	30000	11	21.15
4	30000	35000	5	9.62
5	35000	40000	3	5.77

8

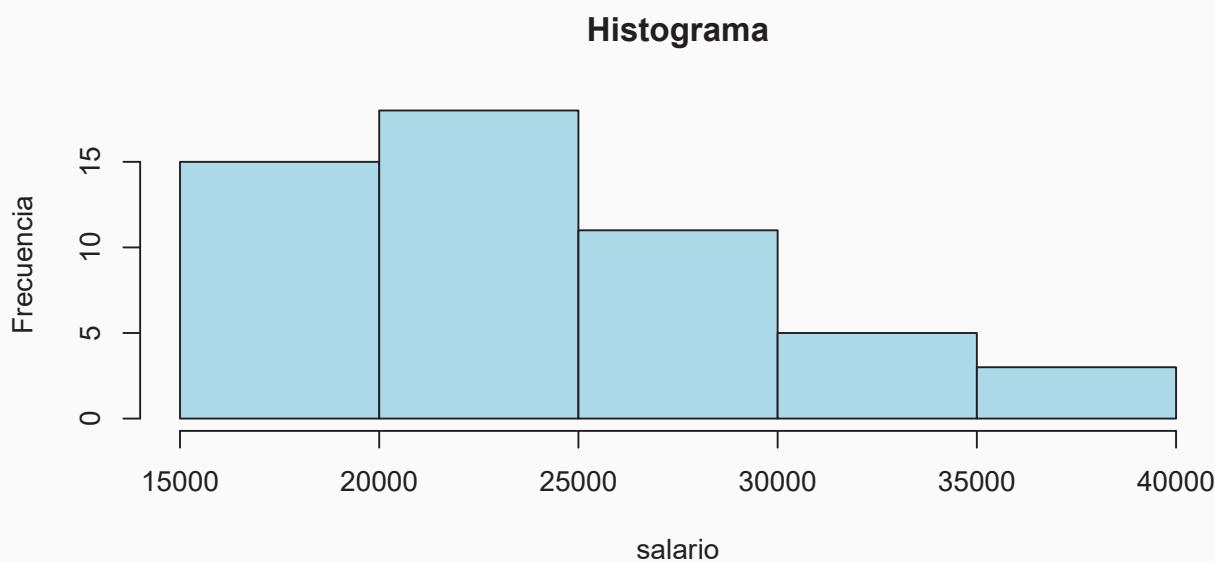
Histograma básico

```
hist(dat$salary)
```

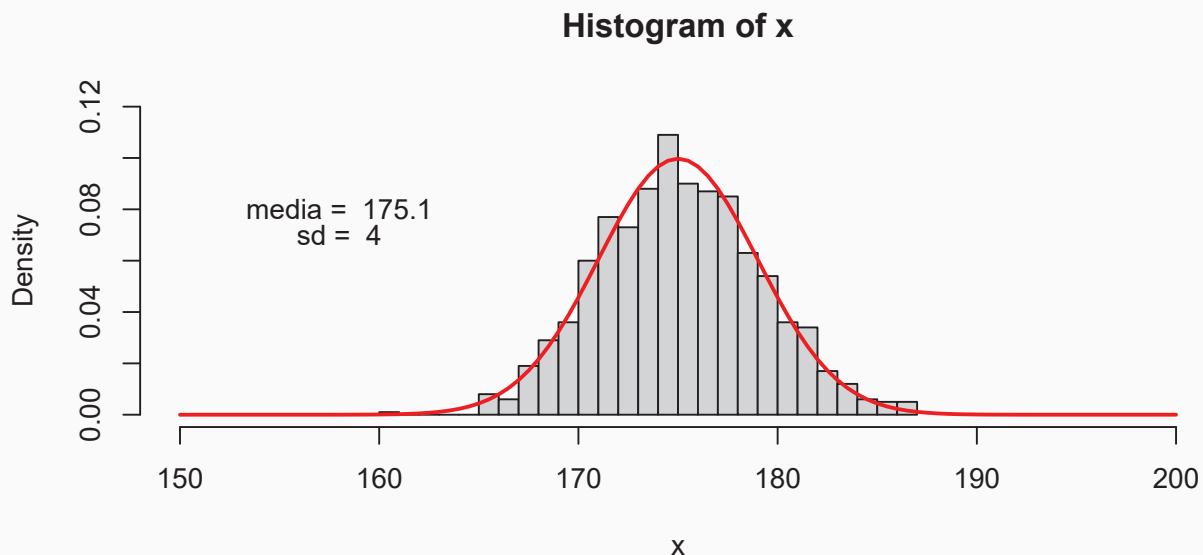


Histograma elaborado

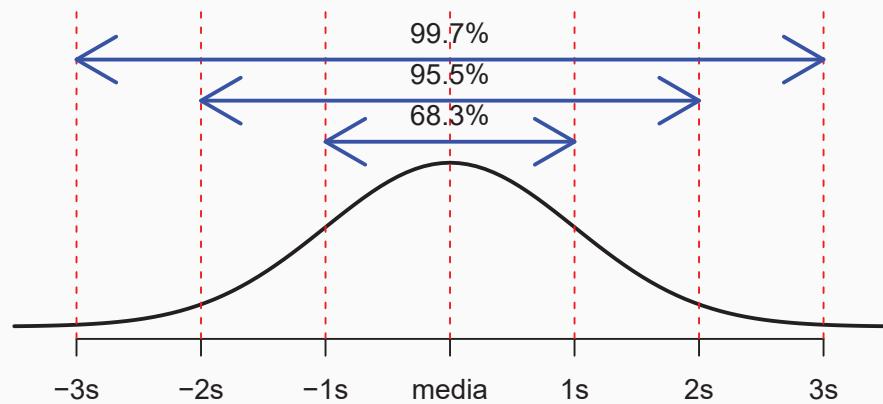
```
hist(dat$salary, col ="lightblue",
     main = "Histograma", xlab = "salario", ylab = "Frecuencia")
```



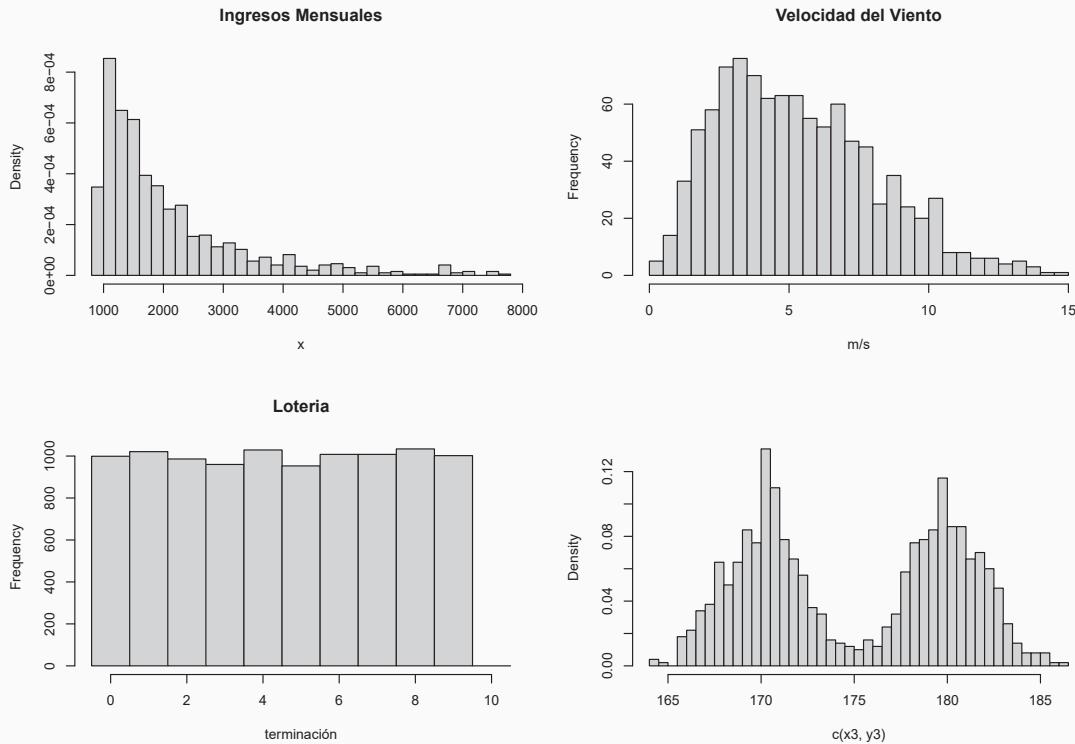
Distribución normal



11



12

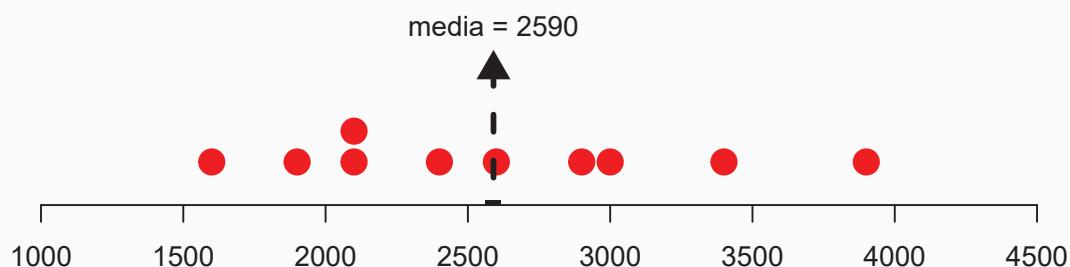


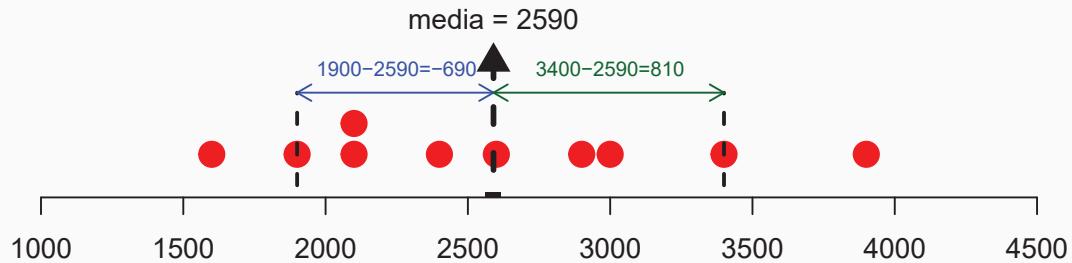
Media aritmética

$$\bar{X} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

Ejemplo: Salario Hombres

```
## 3400 2600 1600 2100 1900 3900 2100 2900 2400 3000
```





$$d_i = x_i - \bar{x}$$

$$d_1 + d_2 + \dots + d_n = 0 \quad \dots \Rightarrow \quad \sum_{i=1}^n (x_i - \bar{x}) = 0$$

15

Desviación típica

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n}} = 679 \text{ euros}$$

Salario	desviacion
3400	810
2600	10
1600	-990
2100	-490
1900	-690
3900	1310
2100	-490
2900	310
2400	-190
3000	410
mean	2590
	0

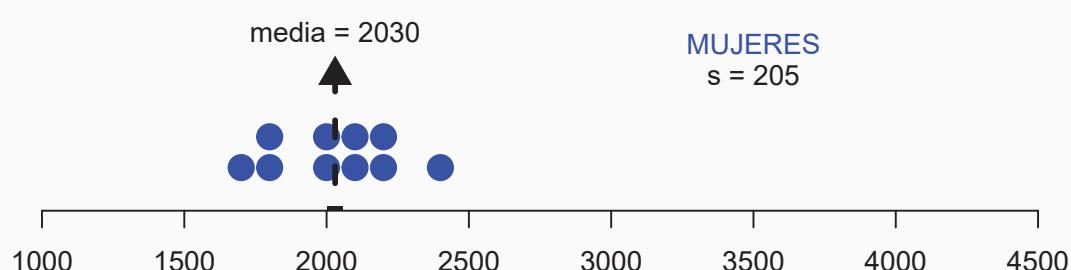
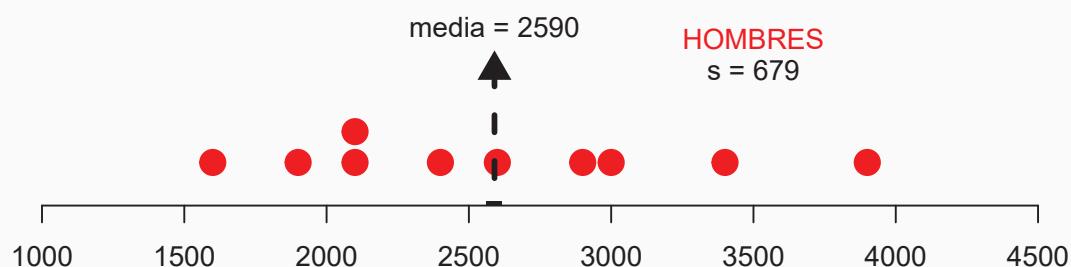
16

Desviación típica

hombres	mujeres
3400	2100
2600	2000
1600	2200
2100	1700
1900	2000
3900	1800
2100	2400
2900	2100
2400	2200
3000	1800

17

Desviación típica

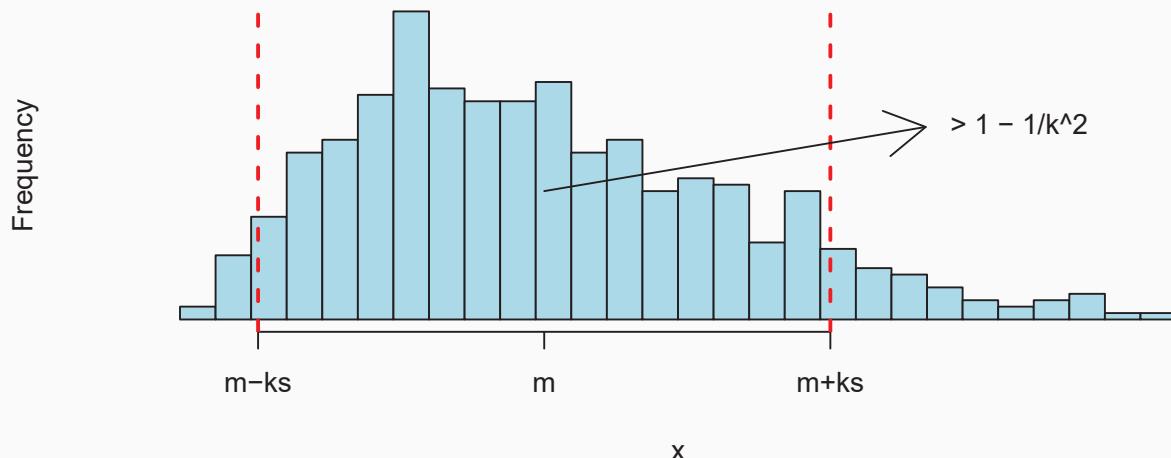


18

Desigualdad de Tchebychev

$$fr(|x_i - \bar{x}| < ks) \geq 1 - \frac{1}{k^2}$$

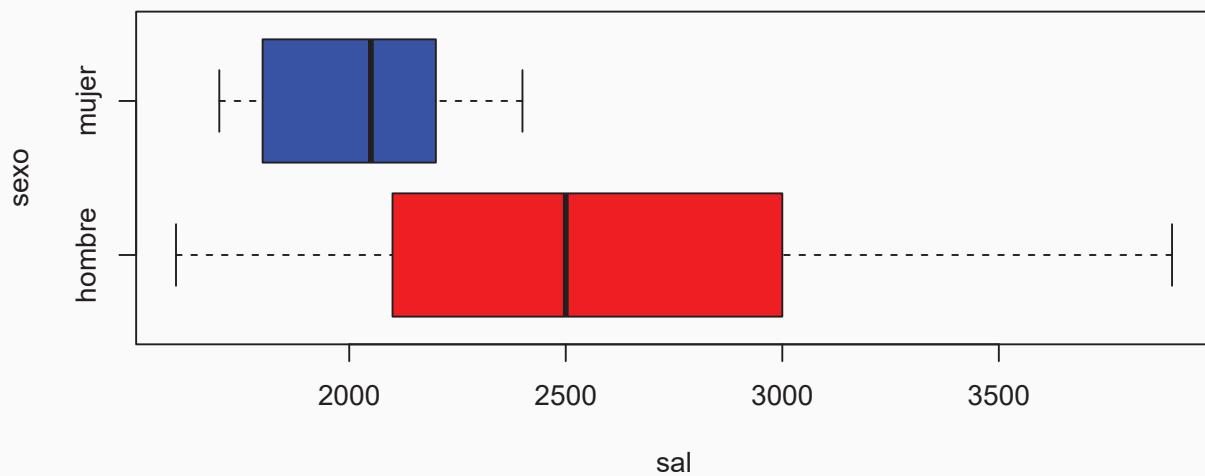
Histogram of x



19

Boxplot

```
x1 = c(3400, 2600, 1600, 2100, 1900, 3900, 2100, 2900, 2400, 3000)
x2 = c(2100, 2000, 2200, 1700, 2000, 1800, 2400, 2100, 2200, 1800)
salario = data.frame(sal = c(x1,x2), sexo = c(rep(1,10),rep(2,10)))
salario$sexo = factor(salario$sexo, labels=c("hombre","mujer"))
boxplot(sal~sexo,data=salario, horizontal = T, col=c("red","blue"))
```



20

1 Hombres

```
summary(x1)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	1600	2100	2500	2590	2975	3900

2 Mujeres

```
summary(x2)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	1700	1850	2050	2030	2175	2400

Segunda parte: Descriptiva Multivariante

Tres posibilidades

- 1 Cualitativa - Cualitativa: *sex - rank*
- 2 Cuantitativa - Cualitativa : *salary - rank*
- 3 Cuantitativa - Cuantitativa: *salary - year (antigüedad)*

23

Cualitativa - Cualitativa (Sex- Rank)

```
(t2 = table(dat$rank,dat$sex))
```

```
##  
##          Female Male  
##  Assist      8   10  
##  Assoc       2   12  
##  Prof        4   16
```

```
(t3 = prop.table(t2,1)*100 ) # Porcentaje por género de cada rango
```

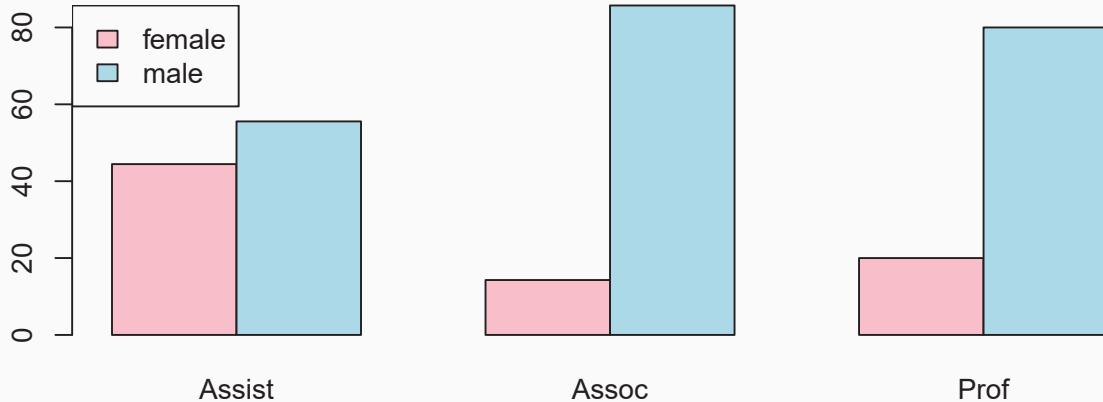
```
##  
##          Female Male  
##  Assist     44.4 55.6  
##  Assoc      14.3 85.7  
##  Prof       20.0 80.0
```

24

Gráfico de tabla doble

```
barplot(t(t3),beside = TRUE,           # t(t3) significa: transponer/girar t3
        col=c("pink","lightblue"))

legend("topleft",
      legend = c("female","male"),
      fill = c("pink","lightblue"),          # Color de los rectángulos
      border = "black") # Color del borde de los rectángulos
```



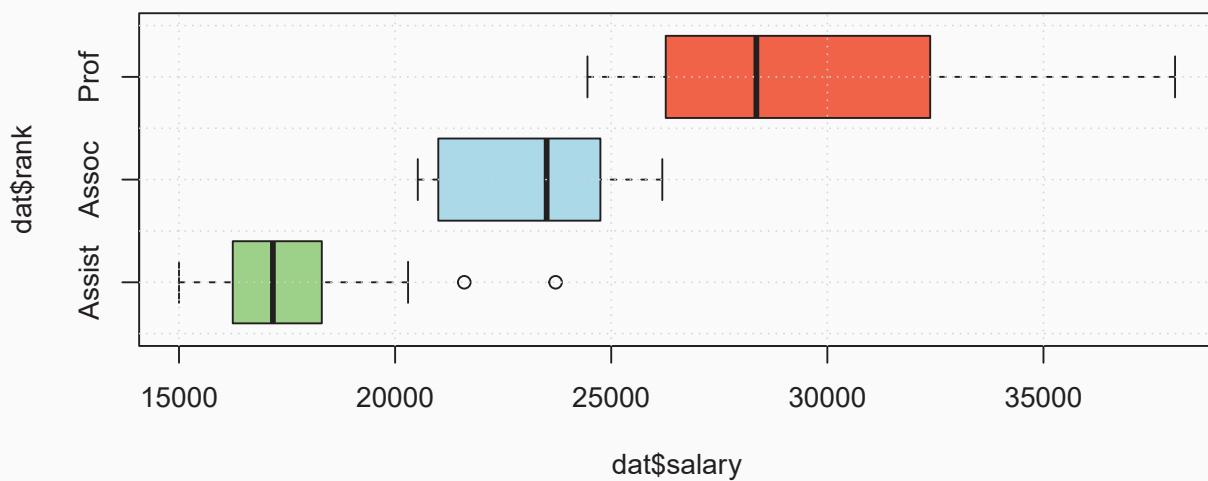
25

Cuantitativa - Cualitativa: salary - rank

```
tapply(dat$salary,dat$rank,mean)    # mean, o sd, var, length, median, ...

## Assist  Assoc  Prof
## 17769  23176  29659

boxplot(dat$salary ~ dat$rank, horizontal = TRUE,
        col=c("lightgreen","lightblue","tomato")); grid()
```



26

Cuantitativa - cualitativa: (salary ~ sex * rank)

```
tapply(dat$salary, list(dat$sex, dat$rank), mean)
```

```
##          Assist Assoc Prof
## Female    17580 21570 28805
## Male      17920 23444 29872
```

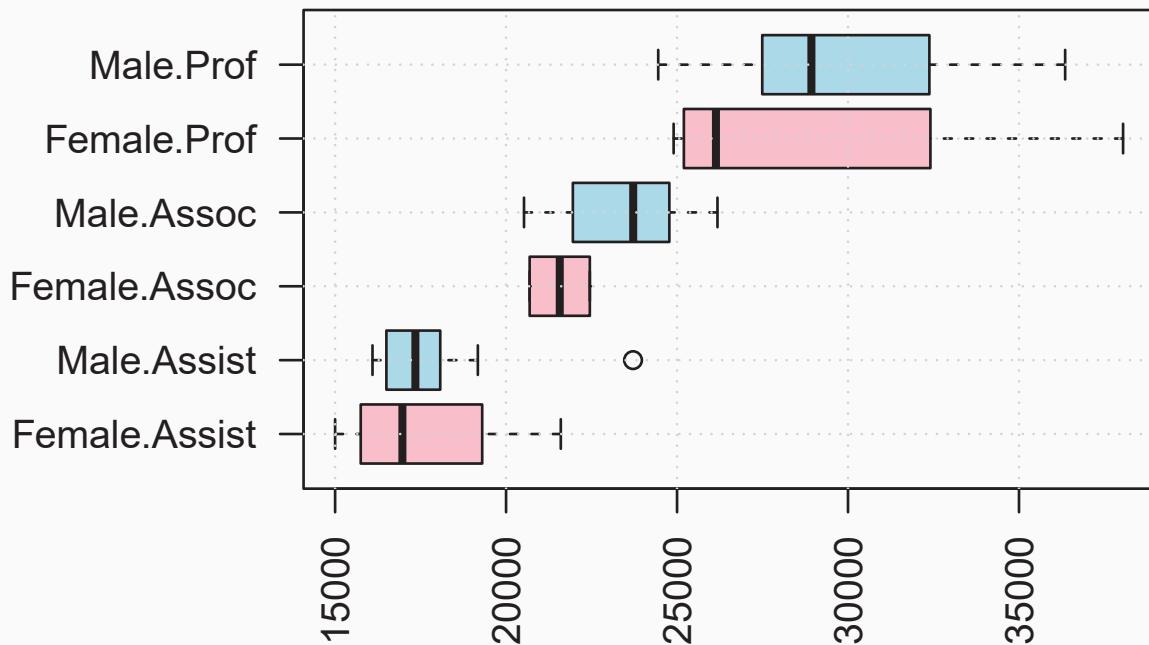
```
tapply(dat$salary, list(dat$sex, dat$rank), length)
```

```
##          Assist Assoc Prof
## Female      8     2     4
## Male       10    12    16
```

27

Boxplot Múltiple

```
par(mar=c(5.1,7.1,1.1,2.1))
boxplot(dat$salary~dat$sex*dat$rank,
        ylab="", xlab="", las=2,
        horizontal=TRUE,
        col=c("pink","lightblue"))
grid()
```

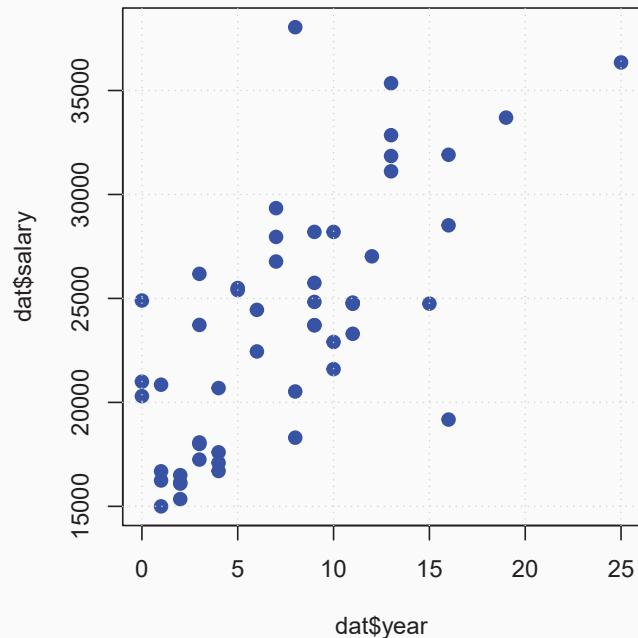


28

```

Código R:
par(pty = "s")
plot(dat$year,dat$salary, col = "blue", pch= 19, cex=1.2)
grid()

```



29

Modelo de regresión simple

```

par(pty = "s")
plot(dat$year,dat$salary, col = "blue", pch= 19, cex=1.2)
mod = lm(salary ~ year, data = dat); summary(mod)
abline(mod, col="red", lty=2, lwd=2)
grid()

```

$$salary_i = 18166 + 753 \times year_i + e_i, \quad \hat{s}_R = 4.260\$, \quad R^2 = 49.1\%$$

```

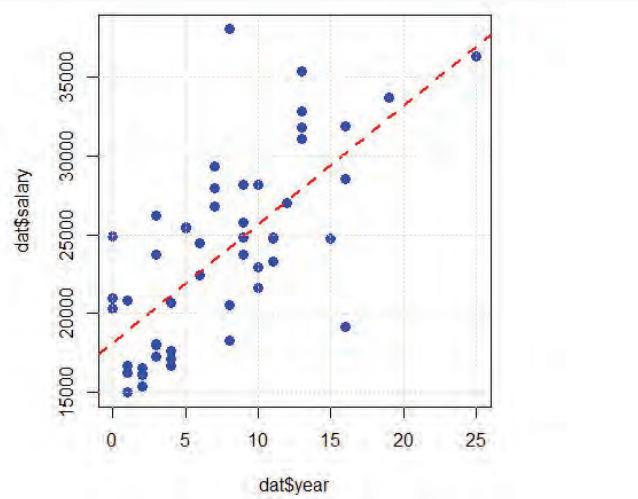
Call:
lm(formula = salary ~ year, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-11036  -3172   -562   3186  13856 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 18166      1004  18.10 < 2e-16 ***
year         753       108   6.94 7.3e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' 

Residual standard error: 4260 on 50 degrees of freedom
Multiple R-squared:  0.491, Adjusted R-squared:  0.481 
F-statistic: 48.2 on 1 and 50 DF,  p-value: 7.34e-09

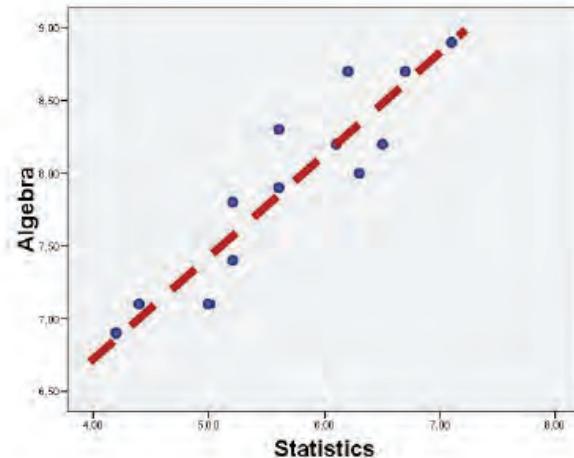
```



30

Gráfico dispersión

Statistics	Algebra
7,10	8,90
6,10	8,20
5,60	8,30
5,20	7,40
4,40	7,10
6,20	8,70
6,30	8,00
5,00	7,10
4,20	6,90
6,70	8,70
6,50	8,20
5,60	7,90
5,20	7,80



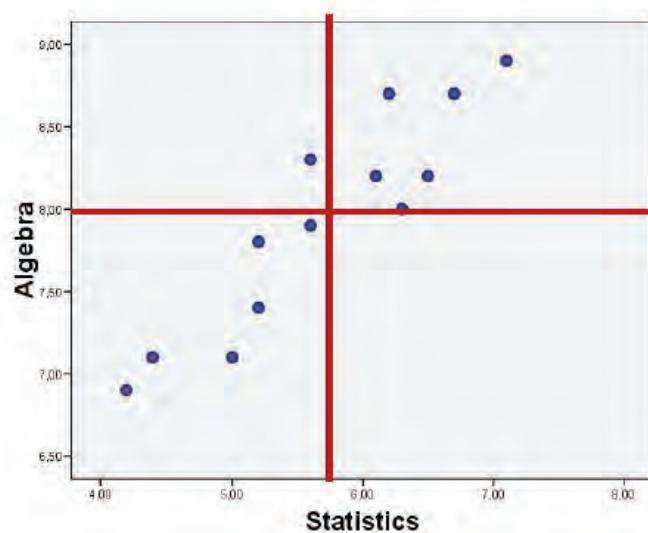
31

Gráfico dispersión

Statistics	Algebra
7,10	8,90
6,10	8,20
5,60	8,30
5,20	7,40
4,40	7,10
6,20	8,70
6,30	8,00
5,00	7,10
4,20	6,90
6,70	8,70
6,50	8,20
5,60	7,90
5,20	7,80

5,7 7,9

Medias



32

$$S_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n}$$

COVARIANZA

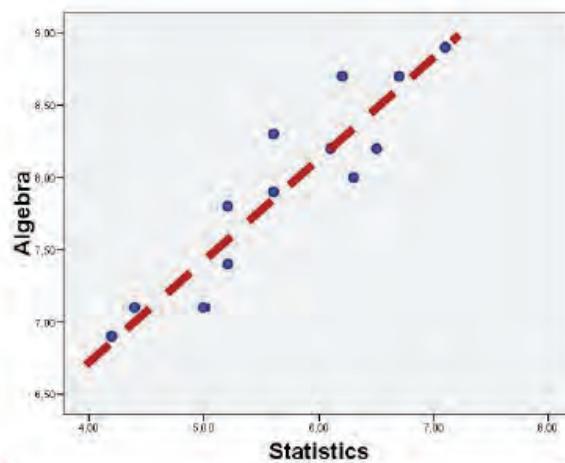
Statistics	Algebra	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$
7,1	8,9	1,40	0,96	1,35
6,1	8,2	0,40	0,26	0,10
5,6	8,3	-0,10	0,36	-0,04
5,2	7,4	-0,50	-0,54	0,27
4,4	7,1	-1,30	-0,84	1,09
6,2	8,7	0,50	0,76	0,38
6,3	8,0	0,60	0,06	0,04
5,0	7,1	-0,70	-0,84	0,59
4,2	6,9	-1,50	-1,04	1,56
6,7	8,7	1,00	0,76	0,76
6,5	8,2	0,80	0,26	0,21
5,6	7,9	-0,10	-0,04	0,00
5,2	7,8	-0,50	-0,14	0,07
5,70	7,94	0,00	0,00	0,49

$$S_{xy} = 0.49$$

Correlacion

Statistics	Algebra
7,10	8,90
6,10	8,20
5,60	8,30
5,20	7,40
4,40	7,10
6,20	8,70
6,30	8,00
5,00	7,10
4,20	6,90
6,70	8,70
6,50	8,20
5,60	7,90
5,20	7,80

0.85 0.63
S.D.



$$r_{xy} = \frac{0.49}{0.85 \times 0.63} = +0.92$$

Covarianza y Correlación

Medidas de relación lineal

covarianza

$$s_{x,y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n}$$

```
cov(dat$salary,dat$year)
```

```
## [1] 22835
```

correlación

$$r_{x,y} = \frac{s_{x,y}}{s_x s_y}$$

```
cor(dat$salary,dat$year)
```

```
## [1] 0.701
```

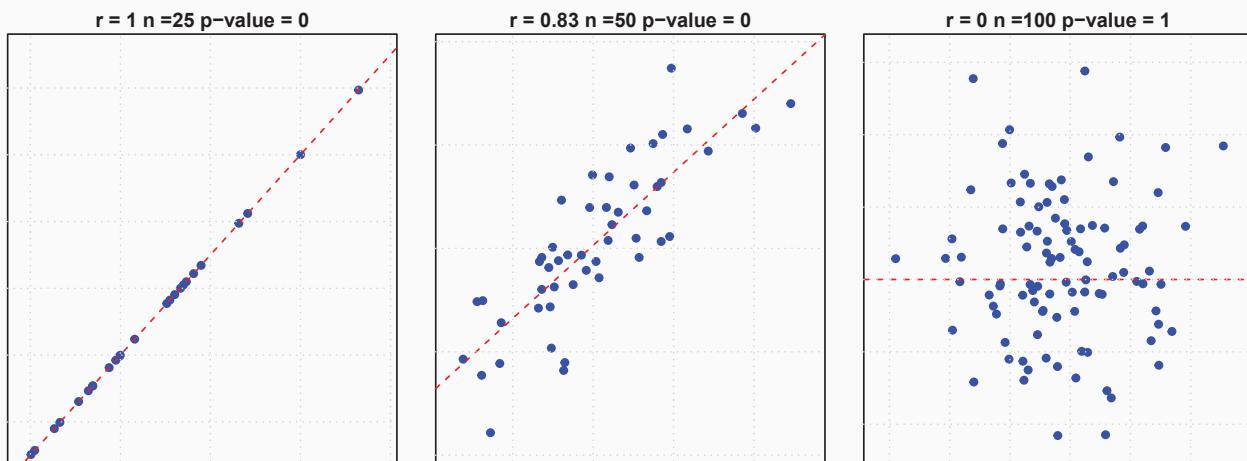
35

Correlación

Propiedades

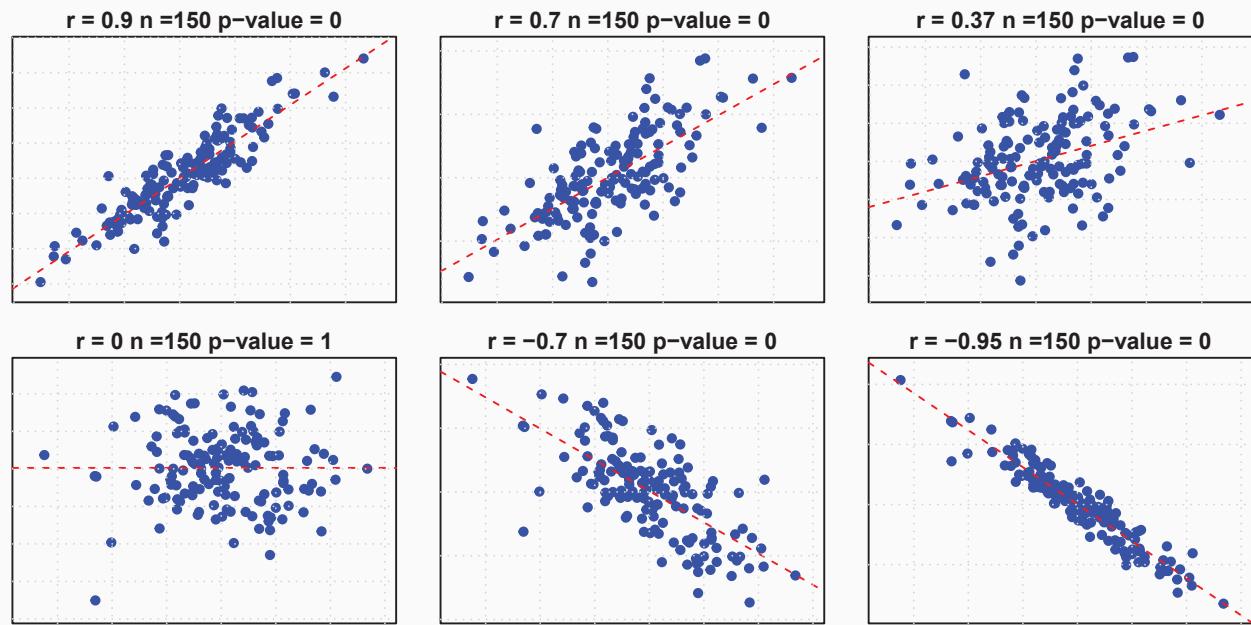
$$-1 \leq r_{x,y} \leq 1$$

- Simétrica: $r_{x,y} = r_{y,x}$
- No tiene unidades (No cambia si cambiamos (linealmente) las unidades de las variables)
- Fácil de interpretar



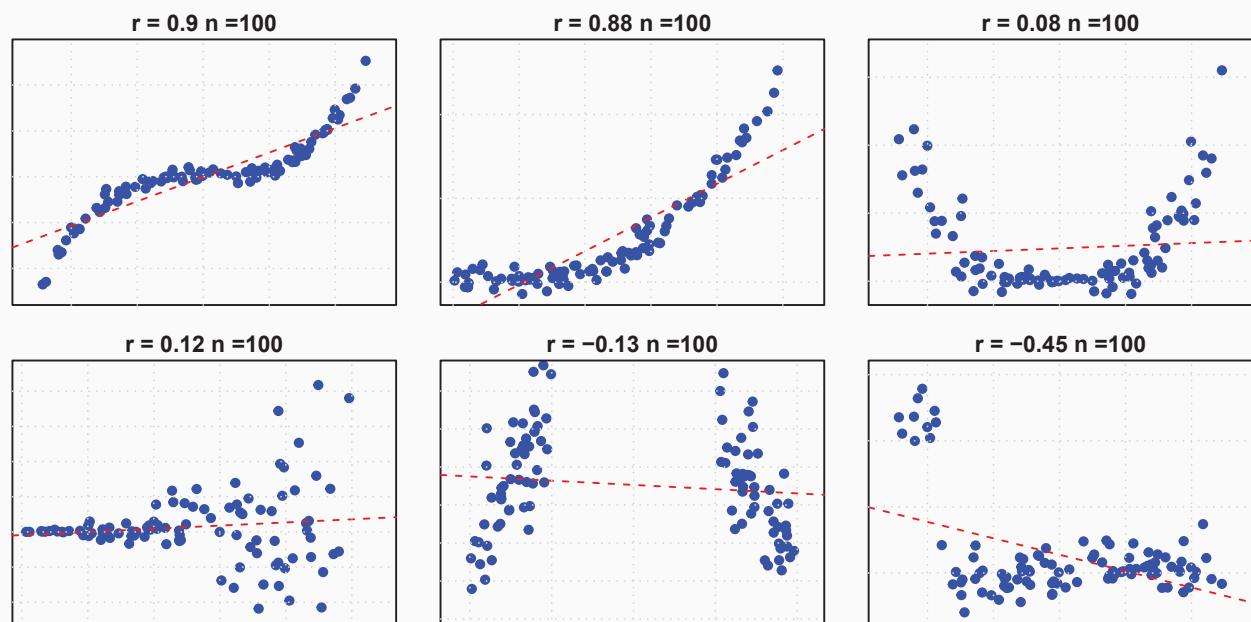
36

Ejemplos de correlaciones



37

Relaciones no lineales (correlación no es útil)



38

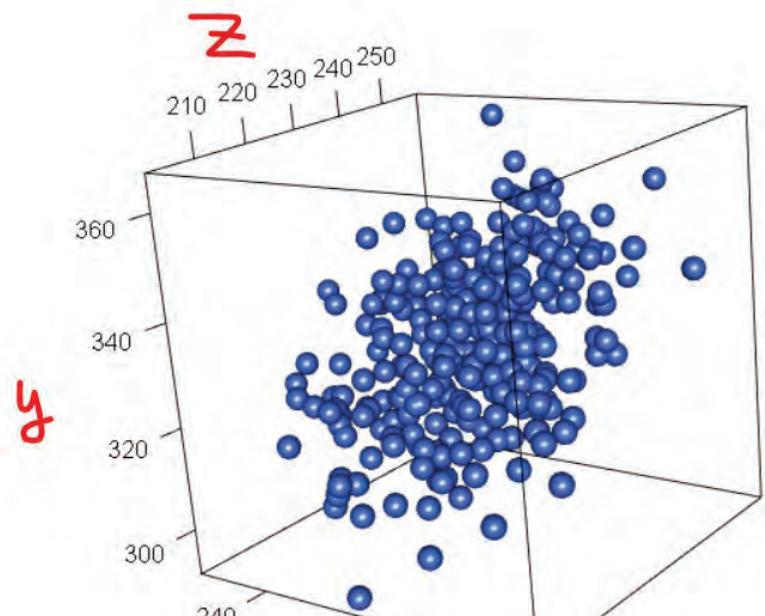
Tercera parte: Varias variables continuas ($k>2$)

Ejemplo 1 (artificial)

```
dat1 = read.table("data/students.txt", header=TRUE)
head(dat1[, 2:6], 12)
```

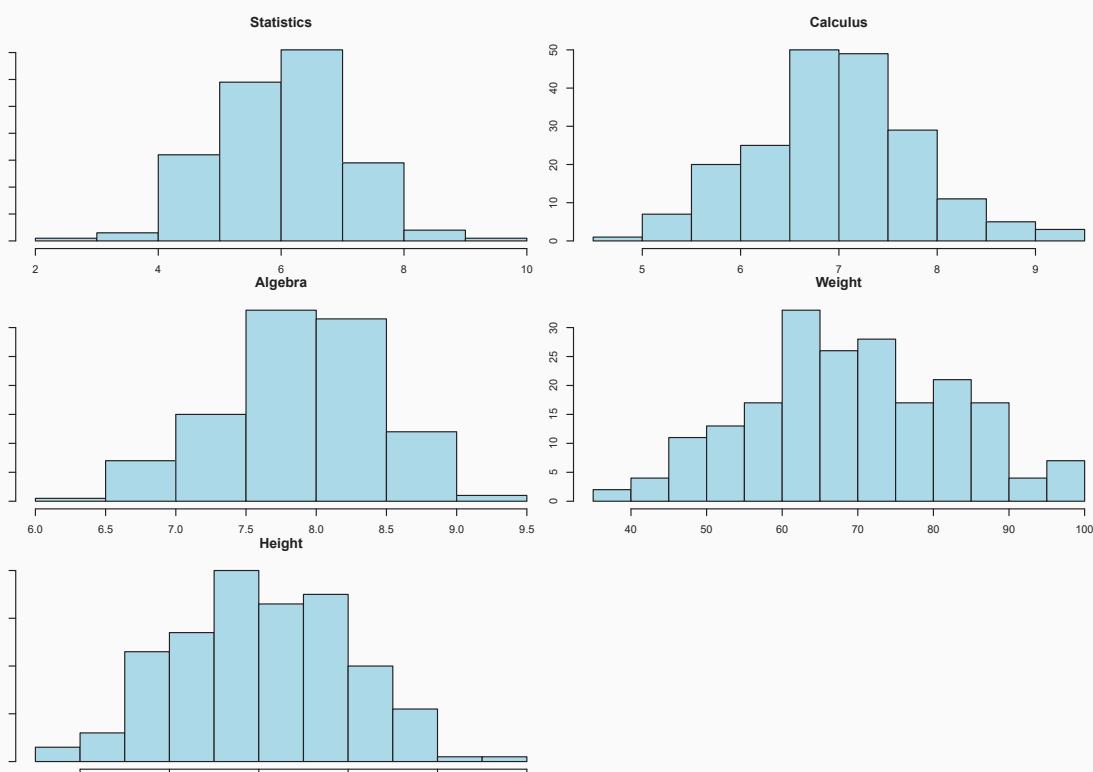
	Statistics	Calculus	Algebra	Weight	Height
## 1	7.1	7.4	8.9	81.4	178
## 2	6.1	7.1	8.2	84.1	185
## 3	5.6	7.1	8.3	60.3	172
## 4	5.2	6.6	7.4	65.9	175
## 5	4.4	6.0	7.1	72.8	176
## 6	6.2	7.0	8.7	79.5	181
## 7	5.7	6.7	8.0	91.7	183
## 8	6.3	7.2	8.1	74.7	194
## 9	4.4	5.1	7.9	93.3	176
## 10	7.4	8.1	8.4	83.9	168
## 11	6.5	7.7	8.0	92.3	175

- Estudiamos k variables continuas
- Los datos están formando una única nube elíptica en el espacio de dimensión k
- Cada variable tiene un histograma con forma de campana
- La relación entre cualquier par de variables es lineal



41

Histogramas (univariantes)



42

```
( m=sapply(dat1[,2:6],mean) )
```

```
## Statistics      Calculus      Algebra      Weight      Height
##           6.09       7.02        7.95      69.28     170.96
```

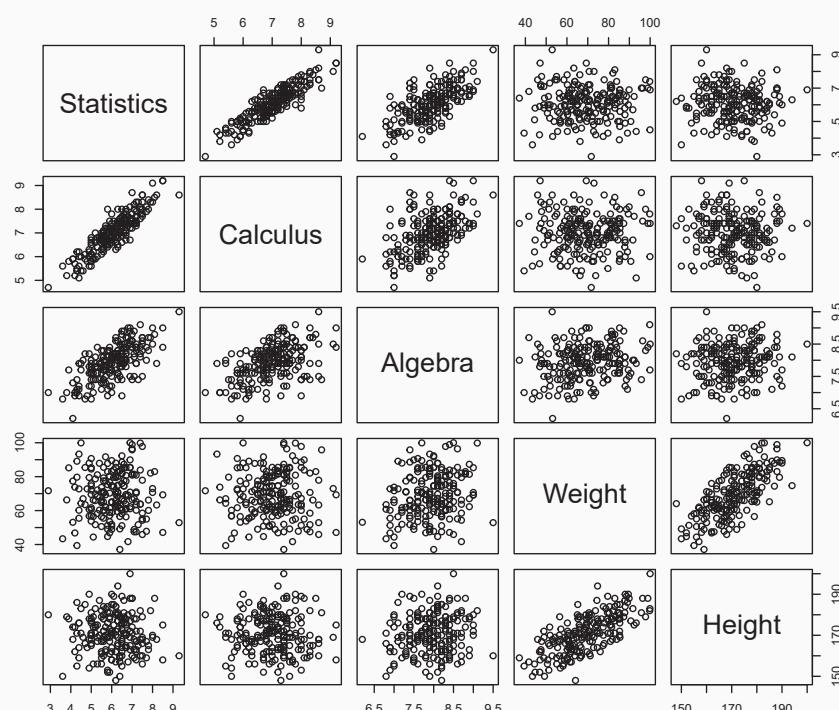
```
( s=sapply(dat1[,2:6],sd) )
```

```
## Statistics      Calculus      Algebra      Weight      Height
##           1.036      0.834       0.552      13.537     9.604
```

43

Gráficos Múltiples

```
pairs(dat1[,2:6])
```



44

Matriz de Covarianzas

```
cov(dat1[,2:6])
```

```
##          Statistics Calculus Algebra  Weight Height
## Statistics     1.0742    0.775   0.386  0.0156 -0.574
## Calculus       0.7751    0.695   0.228  0.1315 -0.213
## Algebra        0.3859    0.228   0.304  1.6105  0.688
## Weight         0.0156    0.131   1.610 183.2422 94.577
## Height        -0.5742   -0.213   0.688  94.5772 92.234
```

Propiedades

- Cuadrada
- Simétrica
- Semidefinida positiva

45

Matriz de correlaciones

```
(r = cor(dat1[,2:6]))
```

```
##          Statistics Calculus Algebra  Weight Height
## Statistics     1.00000   0.8971   0.675  0.00111 -0.0577
## Calculus       0.89712   1.0000   0.495  0.01165 -0.0266
## Algebra        0.67498   0.4949   1.000  0.21571  0.1299
## Weight         0.00111   0.0117   0.216  1.00000  0.7275
## Height        -0.05768  -0.0266   0.130  0.72749  1.0000
```

Propiedades

- Cuadrada
- Simétrica
- Semidefinida positiva

Significativo Depende del número de datos. Un coeficiente r_{ij} es significativo (al 95%) si:

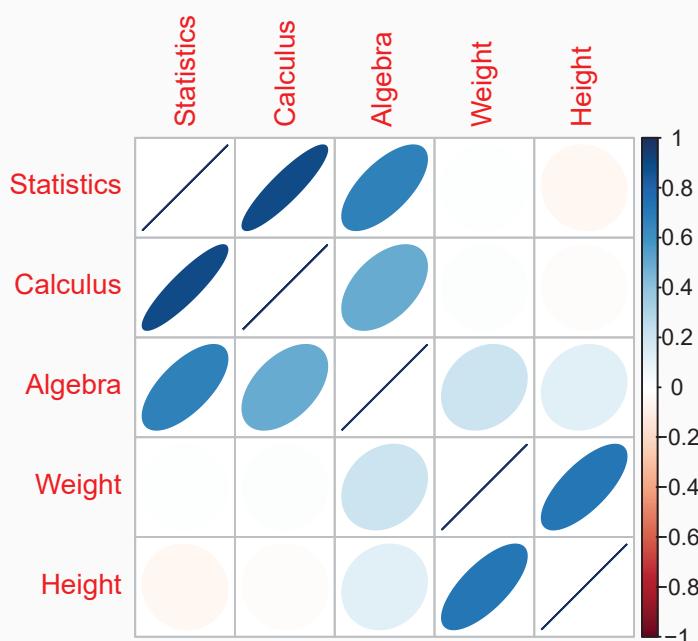
46

```
library(Hmisc)
rcorr(as.matrix(dat1[,2:6])) # , type = c("pearson", "spearman")
```

```
##          Statistics Calculus Algebra Weight Height
## Statistics      1.00     0.90    0.67   0.00 -0.06
## Calculus        0.90     1.00    0.49   0.01 -0.03
## Algebra         0.67     0.49    1.00   0.22  0.13
## Weight          0.00     0.01    0.22   1.00  0.73
## Height         -0.06    -0.03   0.13   0.73  1.00
##
## n= 200
##
## P
##          Statistics Calculus Algebra Weight Height
## Statistics      0.0000    0.0000  0.9875 0.4172
## Calculus        0.0000    0.0000  0.8699 0.7080
## Algebra         0.0000    0.0000  0.0022 0.0668
## Weight          0.9875    0.8699  0.0022 0.0000
## Height         0.4172    0.7080  0.0668 0.0000
```

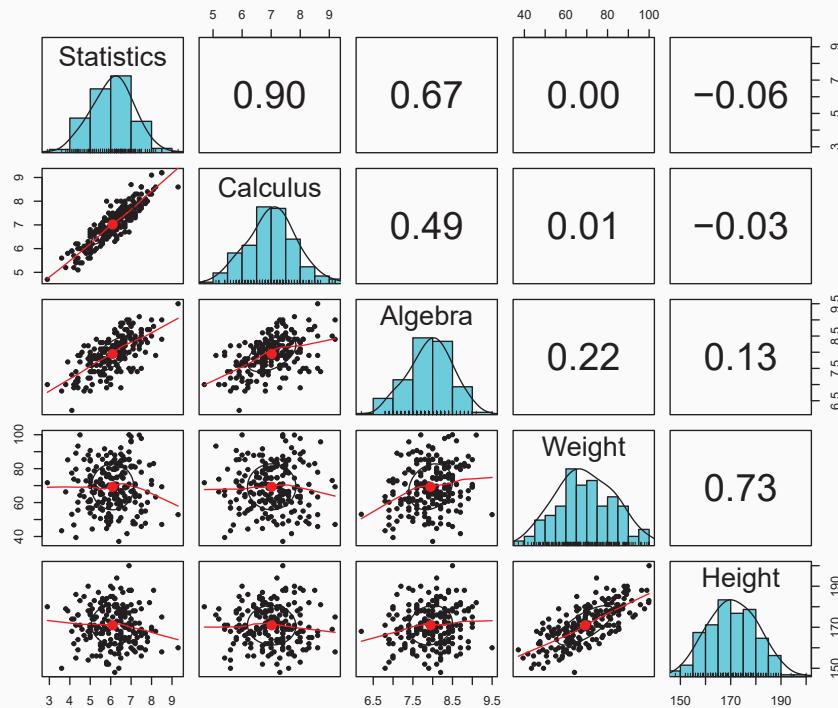
Paquete corrplot

```
library(corrplot)
corrplot(r, method = "ellipse")
```



Paquete psych

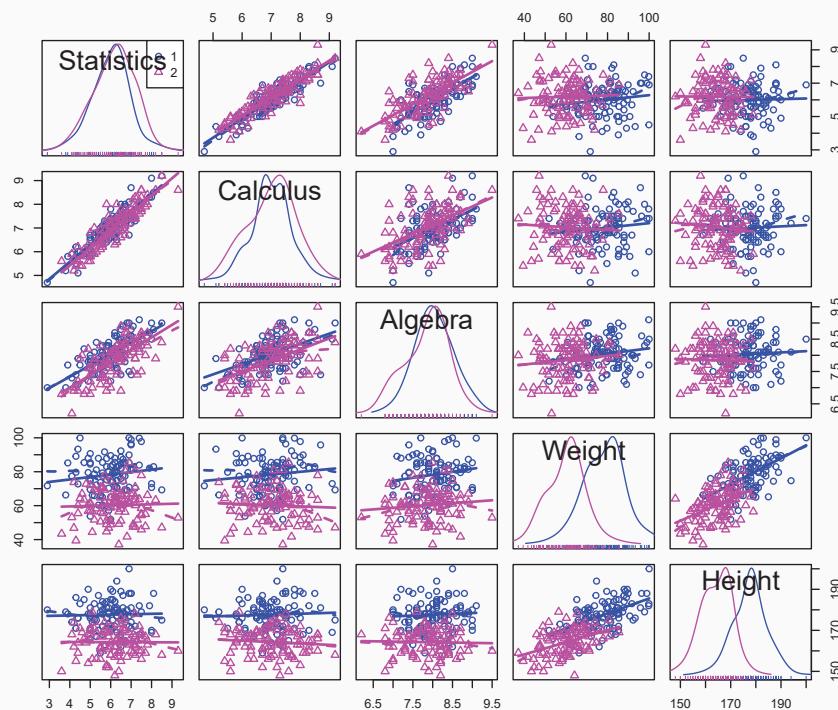
```
library(psych)
pairs.panels(dat1[,2:6])
```



49

Paquete car

```
library(car)
scatterplotMatrix(dat1[,2:6], groups = dat1$Gender)
```



50

```
dat2 = read.table("data/cars.txt", header=TRUE)
head(dat2)
```

```
##   mpg engine horse weight accel origin cylinders
## 1 14     340    160   3609     8.0      1          8
## 2 14     440    215   4312     8.5      1          8
## 3 15     390    190   3850     8.5      1          8
## 4 14     454    220   4354     9.0      1          8
## 5 15     400    150   3761     9.5      1          8
## 6 16     400    230   4278     9.5      1          8
```

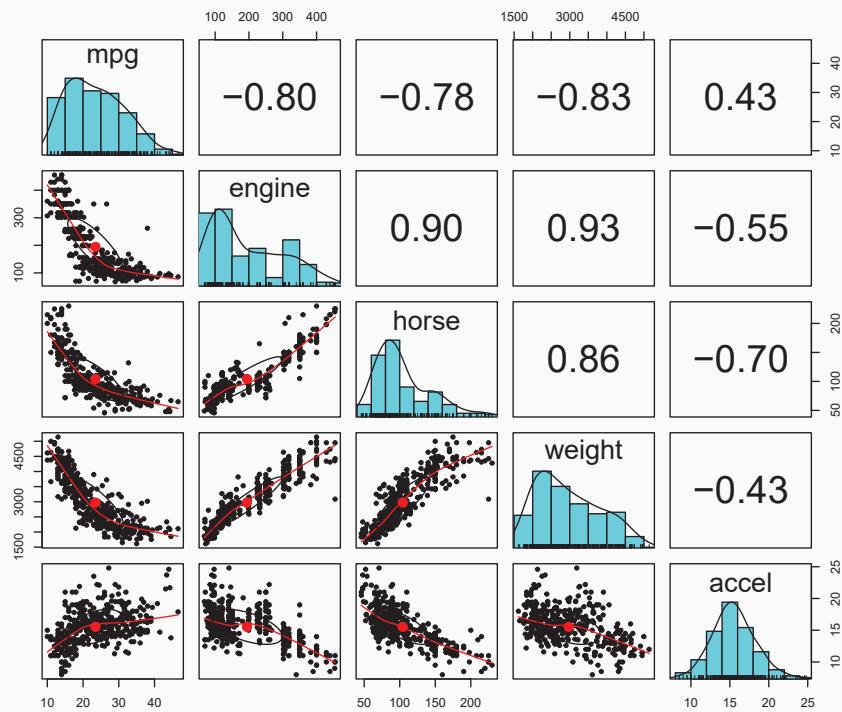
Medidas univariantes

```
psych::describe(dat2[,1:5])
```

	vars	n	mean	sd	median	trimmed	mad	min
## mpg	1	391	23.5	7.78	23.0	23.0	8.90	10
## engine	2	391	194.1	104.63	151.0	183.4	90.44	68
## horse	3	391	104.2	38.28	93.0	99.6	28.17	46
## weight	4	391	2973.1	845.83	2800.0	2912.6	942.93	1613
## accel	5	391	15.5	2.76	15.5	15.5	2.52	8
		max	range	skew	kurtosis	se		
## mpg		46.6	36.6	0.46	-0.54	0.39		
## engine		455.0	387.0	0.70	-0.78	5.29		
## horse		230.0	184.0	1.09	0.70	1.94		
## weight		5140.0	3527.0	0.52	-0.82	42.78		
## accel		24.8	16.8	0.30	0.42	0.14		

Correlaciones

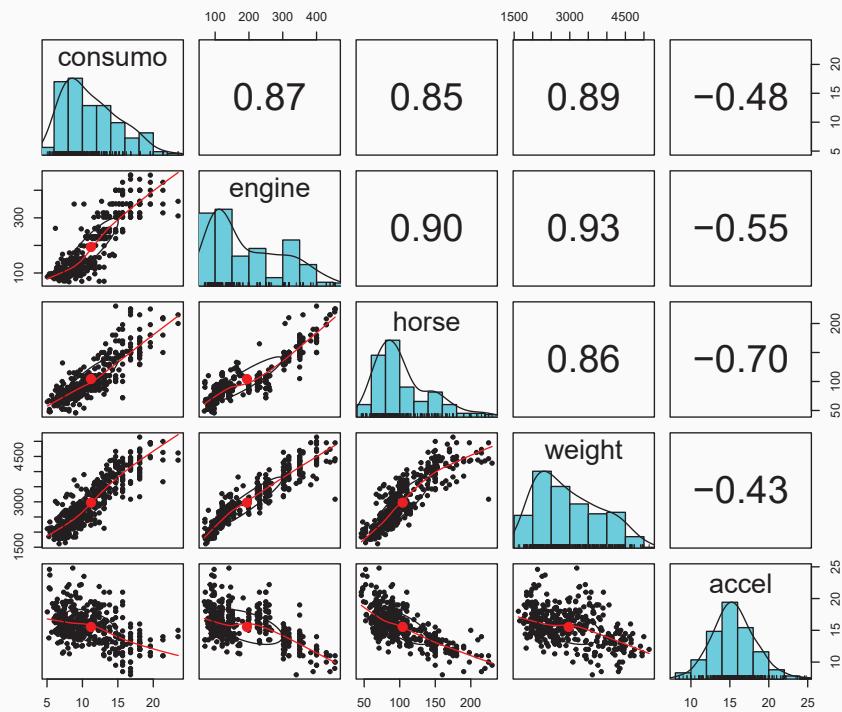
```
pairs.panels(dat2[,1:5])
```



53

Transformaciones

```
dat2$consumo = 235.1/dat2$mpg  
pairs.panels(dat2[,c(8,2:5)])
```



54

Ejemplo 3 Cuerpo

```
dat3 = read.table("data/cuerpo.txt", header=TRUE)
names(dat3)
```

```
## [1] "A_Hombros"  "A_Pelvis"    "A_Cade"      "AP_Pecho"
## [5] "AD_Pecho"    "A_Codo"      "A_Muneca"    "A_Rodilla"
## [9] "A_Tobillo"   "C_hombros"   "C_Pecho"     "C_Cintura"
## [13] "C_abdomen"   "C_Cadera"    "C_Muslo"     "C_Biceps"
## [17] "C_Brazo"     "C_Rodilla"   "C_Gemelo"    "C_Tobillo"
## [21] "C_Muneca"    "Edad"        "Peso"        "Altura"
## [25] "Sexo"
```

55

Medias

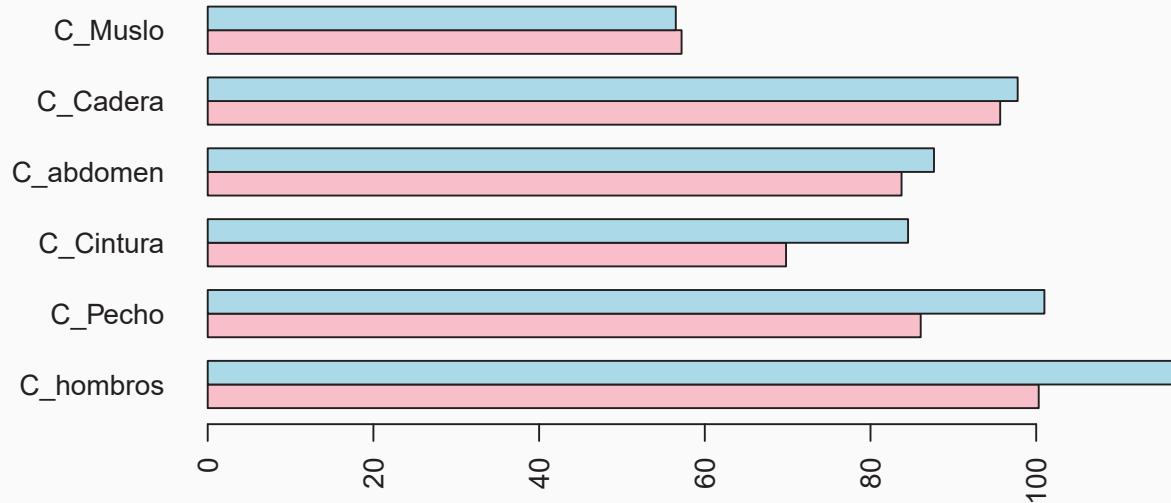
```
sel = c(10,11,12,13,14,15)
m1 = sapply(dat3[dat3$Sexo==0,sel],mean)
m2 = sapply(dat3[dat3$Sexo==1,sel],mean)
m = rbind(mujer=m1,hombre=m2,dif=m2-m1)
print(m,digits=2)
```

```
##          C_hombros C_Pecho C_Cintura C_abdomen C_Cadera
## mujer       100      86       70      83.7      95.7
## hombre      117      101      85      87.7      97.8
## dif         16       15       15      3.9       2.1
##          C_Muslo
## mujer      57.2
## hombre     56.5
## dif        -0.7
```

56

Diagrama de barras

```
par(mar=c(5.1,7.1,1.1,2.1))
barplot(m[1:2,],beside = TRUE, horiz=TRUE, las=2,col=c("pink","lightblue"))
```



57

Matriz de varianzas

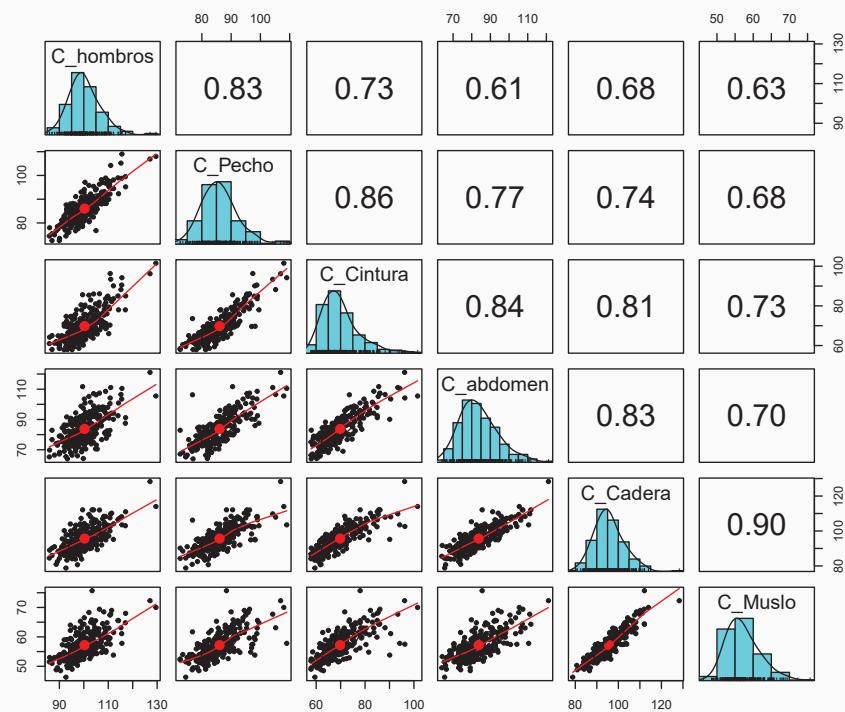
```
(s1 = cor(dat3[dat3$Sexo ==0,sel]))
```

```
##          C_hombros C_Pecho C_Cintura C_abdomen C_Cadera
## C_hombros    1.000   0.827   0.726    0.613   0.679
## C_Pecho      0.827   1.000   0.859    0.766   0.744
## C_Cintura    0.726   0.859   1.000    0.835   0.812
## C_abdomen    0.613   0.766   0.835    1.000   0.830
## C_Cadera     0.679   0.744   0.812    0.830   1.000
## C_Muslo       0.630   0.675   0.728    0.701   0.904
##          C_Muslo
## C_hombros    0.630
## C_Pecho      0.675
## C_Cintura    0.728
## C_abdomen    0.701
## C_Cadera     0.904
## C_Muslo      1.000
```

58

Gráficos para Mujeres

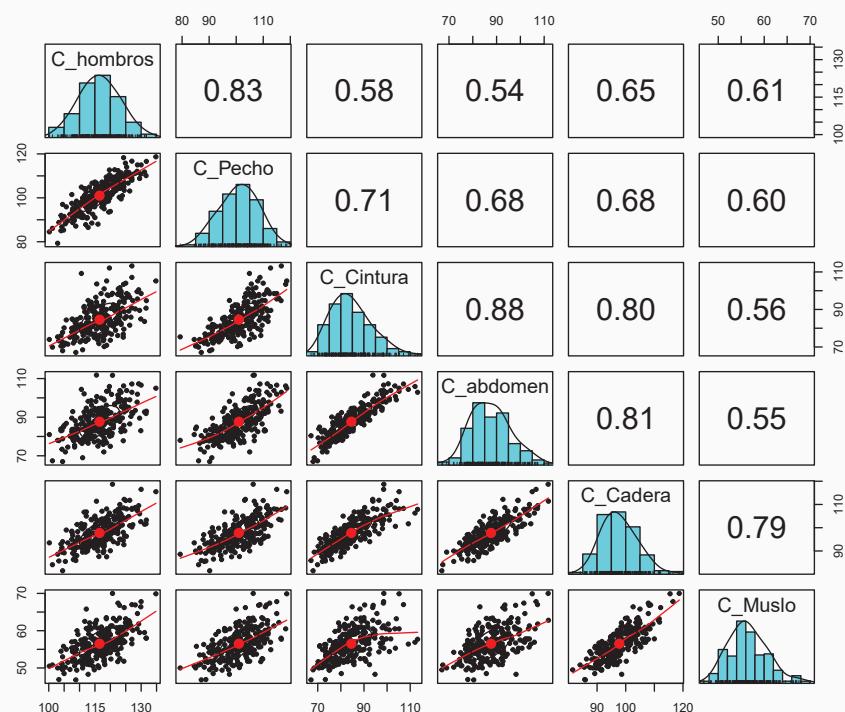
```
pairs.panels(dat3[dat3$Sexo ==0,sel])
```



59

Gráficos para Hombres

```
pairs.panels(dat3[dat3$Sexo ==1,sel])
```



60

- La descriptiva multivariante es una parte fundamental en el análisis de datos y tiene como objetivo el descubrimiento, la visualización y la comunicación de **patrones** significativos en los datos.
- Se han proporcionado un conjunto de herramientas básicas útiles para realizar análisis multivariante (análisis de datos)
- Hacer una estadística descriptiva previa a cualquier otro análisis es imprescindible y útil
- Muchas de las técnicas que vamos a estudiar en este curso son útiles para completar la “descripción” multivariante de un conjunto de datos (componentes principales, análisis cluster, ...)

Intrucciones más importantes 1

```
dat = read.table("data/salary.txt",header=TRUE)

(t1 = table(dat$rank))

barplot(t1,col=c("lightgreen","lightblue","tomato"))

hist(dat$salary,col="lightblue")

(t2 = table(dat$rank,dat$sex))

(t3 = prop.table(t2,1)*100 )

tapply(dat$salary,dat$rank,mean)
```

Intrucciones más importantes 2

```
boxplot(dat$salary ~ dat$rank, horizontal = TRUE,  
        col=c("lightgreen","lightblue","tomato"))  
  
tapply(dat$salary,list(dat$sex,dat$rank),mean)  
  
plot(dat$year,dat$salary, col = "blue", pch= 19, cex=1.2)  
  
cov(dat$salary,dat$year)  
  
cor(dat$salary,dat$year)
```

63

Intrucciones más importantes 3

```
dat1 = read.table("data/students.txt",header=TRUE)  
( m=sapply(dat1[,2:6],mean) )  
  
pairs(dat1[,2:6])  
  
cov(dat1[,2:6])  
(r = cor(dat1[,2:6]))  
  
Hmisc::rcorr(as.matrix(dat1[,2:6]))  
  
corrplot::corrplot(r,method = "ellipse")  
  
psych::pairs.panels(dat1[,2:6])  
  
car::scatterplot(dat3$C_Muslo,dat3$Peso,groups = dat3$Sexo)  
car::scatterplotMatrix(dat1[,2:6],groups=dat1$Gender)
```

64

Datos en R

Primeros Pasos

Lectura y manejo de datos

R es muy flexible y puede leer casi cualquier archivo de datos (texto, csv, excel, ...). Lo habitual es utilizar archivos de texto.

En el archivo “salary.txt” se encuentra la información del salario de distintos profesores de universidad norteamericanos.

```
dat = read.table("salary.txt", header=T) # (1)
head(dat) # (2)

##   degree rank   sex year ysdeg salary
## 1 Masters Prof Male   25   35  36350
## 2 Masters Prof Male   13   22  35350
## 3 Masters Prof Male   10   23  28200
## 4 Masters Prof Female 7   27  26775
## 5 PhD Prof   Male   19   30  33696
## 6 Masters Prof Male   16   21  28516
```

En la instrucción 1 se lee el archivo de datos y se guarda en la memoria del programa con el nombre *dat*, elegido por el usuario. Si la primera fila del archivo tiene el nombre de las variables (como en este caso), se indica con la opción *header=T*.

La instrucción (2) permite mostrar las seis primeras observaciones del archivo, para hacernos una idea de los datos que contiene. Si queremos visualizar todos los datos, se teclea el nombre *dat*.

La tabla de datos “*dat*”, se denomina *data.frame*.

El *data frame* tiene 6 variables (columnas) y 52 observaciones (filas).

```
names(dat)
## [1] "degree" "rank"    "sex"     "year"    "ysdeg"   "salary"
dim(dat)
## [1] 52  6
```

Cada variable del *data frame* se identifica utilizando el nombre del mismo seguido del nombre de la variable, separado por el símbolo \$.

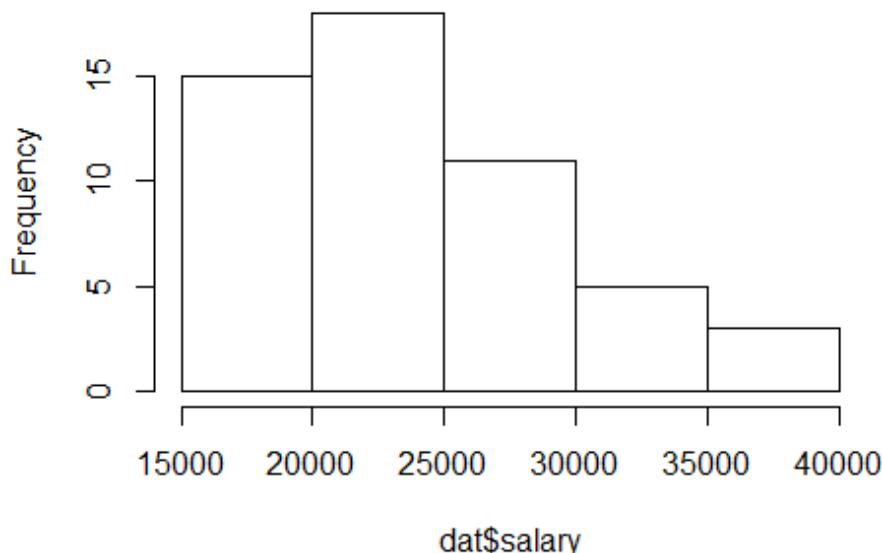
```
dat$salary
```

```
## [1] 36350 35350 28200 26775 33696 28516 24900 31909 31850 32850 27025
## [12] 24750 28200 23712 25748 29342 31114 24742 22906 24450 19175 20525
## [23] 27959 38045 24832 25400 24800 25500 26182 23725 21600 23300 23713
## [34] 20690 22450 20850 18304 17095 16700 17600 18075 18000 20999 17250
## [45] 16500 16094 16150 15350 16244 16686 15000 20300
```

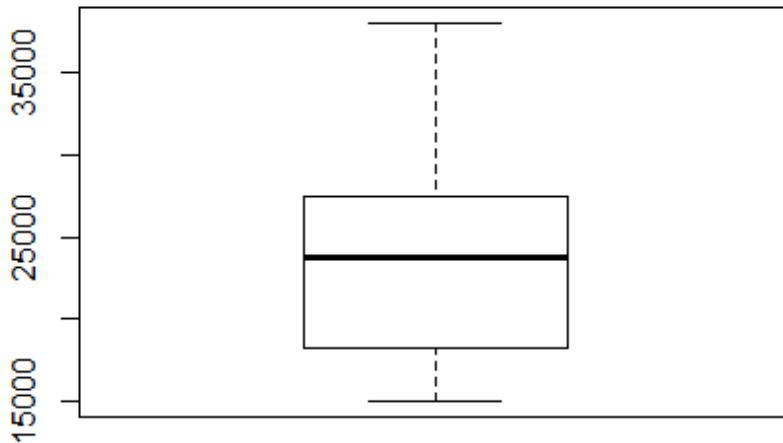
Ahora podemos calcular la media, desviación típica, longitud, cuartiles de estos datos, el histograma y el box-plot con las siguientes instrucciones

```
mean(dat$salary)
## [1] 23797.65
sd(dat$salary)
## [1] 5917.289
length(dat$salary)
## [1] 52
hist(dat$salary)
```

Histogram of dat\$salary



```
boxplot(dat$salary)
```



Otra forma de identificar una variable es por la posición que ocupa en el **data frame**, como *salary* es la sexta columna, se obtiene con las instrucción *dat[6]*.

Si queremos visualizar los datos de la fila 25, se utiliza la instrucción

```
dat[25,]
##      degree rank sex year ysdeg salary
## 25 Masters Assoc Male    9    12 24832
```

Si solo queremos acceder al salario del individuo 25 tenemos dos opciones

```
dat[25,6]
## [1] 24832
dat$salary[25]
## [1] 24832
```

Si queremos extraer los datos de los profesores 1,3,5 y 7, utilizamos

```
dat[c(1,3,5,6),]
##      degree rank sex year ysdeg salary
## 1 Masters Prof Male   25    35 36350
## 3 Masters Prof Male   10    23 28200
## 5    PhD Prof Male   19    30 33696
## 6 Masters Prof Male   16    21 28516
```

Si son contiguos, por ejemplo del 3 al 6, otra alternativa es `dat[3:6,]`.

Si queremos los salarios de las mujeres o hacer cálculos para esos datos se sigue el siguiente procedimiento.

```
dat$salary[dat$sex=="Female"]
## [1] 26775 24900 38045 25500 21600 20690 22450 18304 17250 16150 15350
## [12] 16686 15000 20300
mean(dat$salary[dat$sex=="Female"])
## [1] 21357.14
```

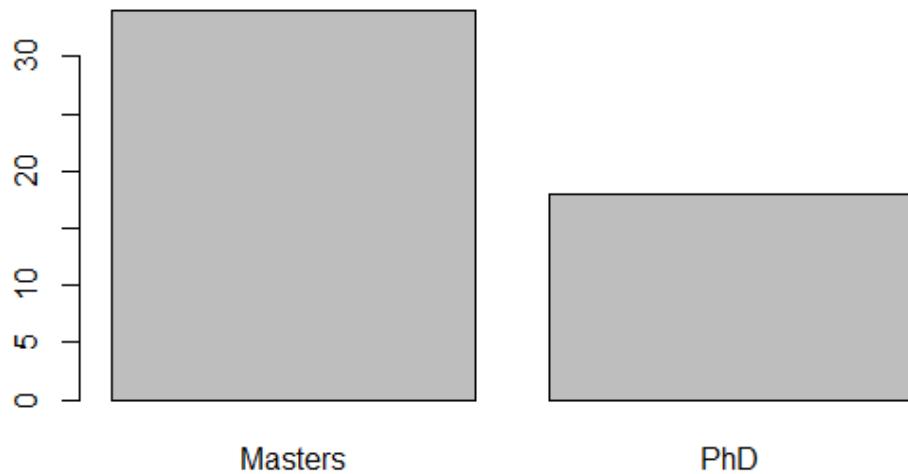
A continuación se muestran un resumen de comandos útiles. Son auto-explicativos. Si se requiere ayuda para alguno de ellos, utilizar el signo de interrogación, por ejemplo la instrucción `?prop.table` nos proporciona ayuda sobre la instrucción `prop.table`.

Variables Cualitativas (Tablas)

```
table(dat$degree)
##
## Masters      PhD
##      34        18
table(dat$rank)
##
## Assist  Assoc   Prof
##     18     14     20
```

Variables Cualitativas (diagrama de barras)

```
barplot(table(dat$degree))
```



Variables Cualitativas (diagrama de barras)

```
barplot(table(dat$rank), col=rainbow(3))
```



Tablas dobles

```
(t1 = table(dat$degree, dat$rank))
```

```
##  
##          Assist Assoc Prof
```

```

##   Masters     14      5    15
##   PhD          4      9    5

(t2 = table(dat$rank,dat$sex))

##
##           Female  Male
##   Assist      8    10
##   Assoc       2    12
##   Prof        4    16

```

Tablas dobles

```

prop.table(t1,1) # suma 1 por columnas

##
##           Assist     Assoc     Prof
##   Masters 0.4117647 0.1470588 0.4411765
##   PhD      0.2222222 0.5000000 0.2777778

prop.table(t1,2) # suma 1 por filas

##
##           Assist     Assoc     Prof
##   Masters 0.7777778 0.3571429 0.7500000
##   PhD      0.2222222 0.6428571 0.2500000

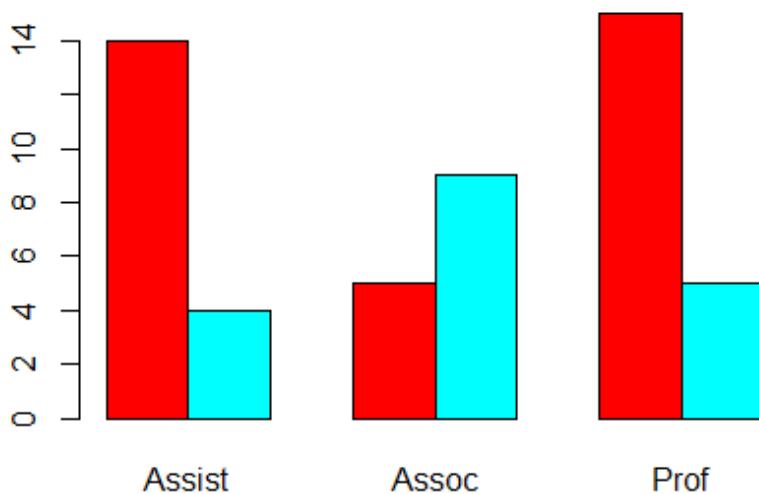
print(prop.table(t1,2),digits=3)

##
##           Assist Assoc  Prof
##   Masters  0.778 0.357 0.750
##   PhD      0.222 0.643 0.250

```

Gráficos de barras apiladas

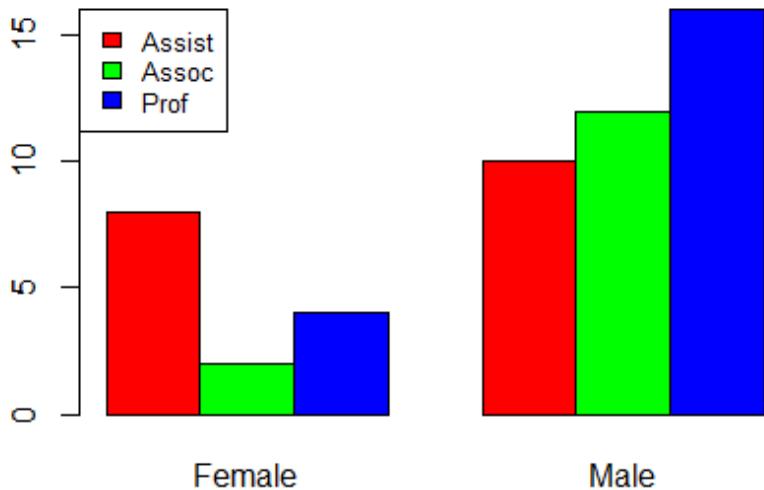
```
barplot(t1,beside=TRUE,col=rainbow(2))
```



```
print(prop.table(t2,1),digits=3)

##
##          Female  Male
##  Assist  0.444 0.556
##  Assoc   0.143 0.857
##  Prof    0.200 0.800

barplot(t2,beside=TRUE,col=rainbow(3))
legend("topleft", legend = c("Assist","Assoc","Prof"),
       fill = rainbow(3),cex=.8)
```



Variables Continuas

```
sapply(dat[,4:6],mean)

##      year      ysdeg      salary
##    7.480769   16.115385 23797.653846

print(sapply(dat[,4:6],mean),digits=3)

##      year      ysdeg      salary
##    7.48     16.12  23797.65
```

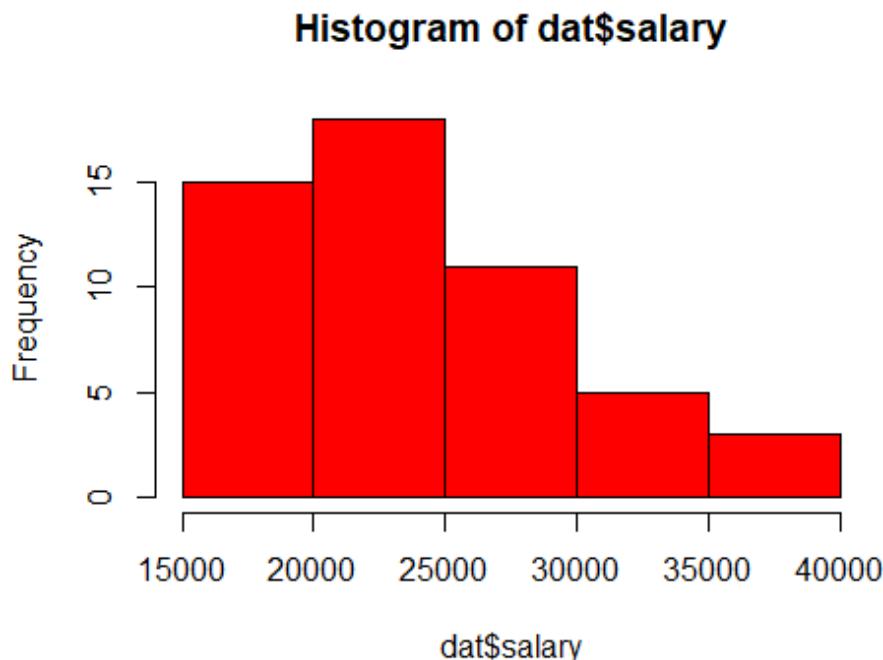
Tablas dobles para Variables Continuas

```
tapply(dat$salary,list(dat$degree,dat$sex),mean)

##          Female     Male
## Masters  20936 24568.83
## PhD      22410 24916.14
```

Histograma

```
h=hist(dat$salary,col="red")
```



```
100*(h$counts/52)
## [1] 28.846154 34.615385 21.153846  9.615385  5.769231
```

Variables Continuas

```
tapply(dat$salary,dat$degree,mean)
##  Masters      PhD
## 23500.35 24359.22

tapply(dat$salary,dat$degree,sd)
##  Masters      PhD
## 6621.102 4408.291
```

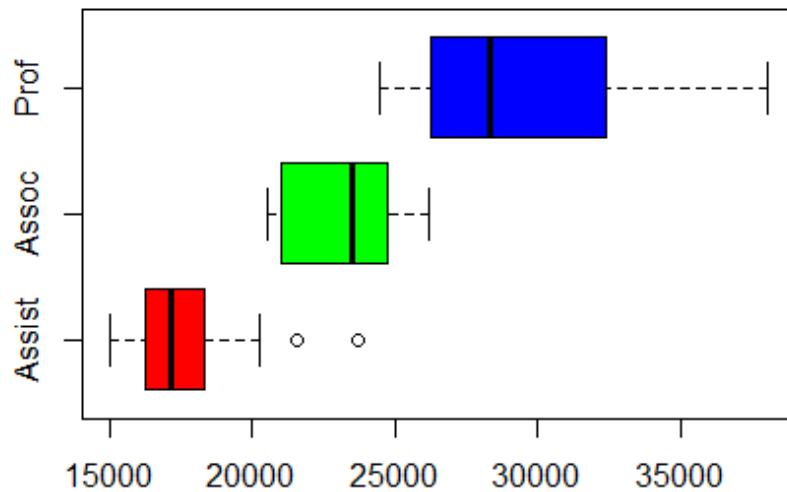
Variables Continuas

```
tapply(dat$salary,dat$rank,mean)
##  Assist    Assoc     Prof
## 17768.67 23175.93 29658.95

tapply(dat$salary,dat$rank,sd)
##  Assist    Assoc     Prof
## 2233.084 1837.792 4041.011
```

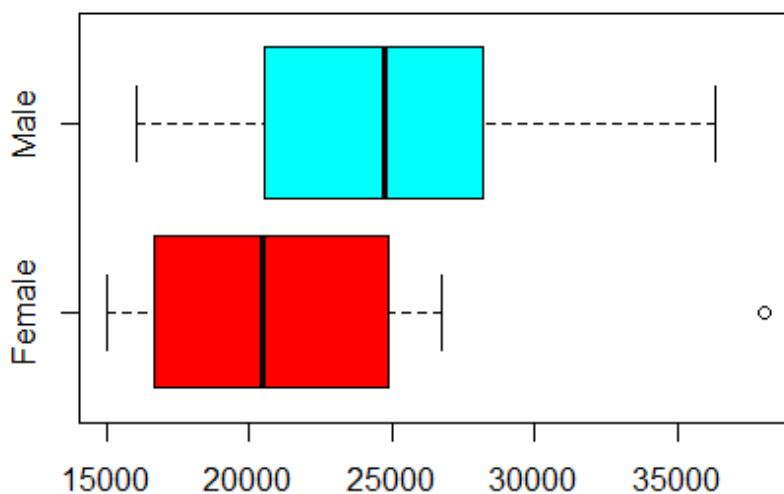
Variables Continuas

```
boxplot(dat$salary~dat$rank,col=rainbow(3),horizontal=T)
```



Variables Continuas

```
boxplot(dat$salary~dat$sex,col=rainbow(2),horizontal=T)
```



summary(dat)

```

##      degree      rank      sex
##  Masters:34  Assist:18  Female:14
##  PhD     :18   Assoc :14   Male  :38
##                  Prof  :20

##      year      ysdeg      salary
##  Min.   : 0.000  Min.   : 1.00  Min.   :15000
##  1st Qu.: 3.000  1st Qu.: 6.75  1st Qu.:18247
##  Median : 7.000  Median :15.50  Median :23719
##  Mean   : 7.481  Mean   :16.12  Mean   :23798
##  3rd Qu.:11.000  3rd Qu.:23.25  3rd Qu.:27259
##  Max.   :25.000  Max.   :35.00  Max.   :38045

```

1

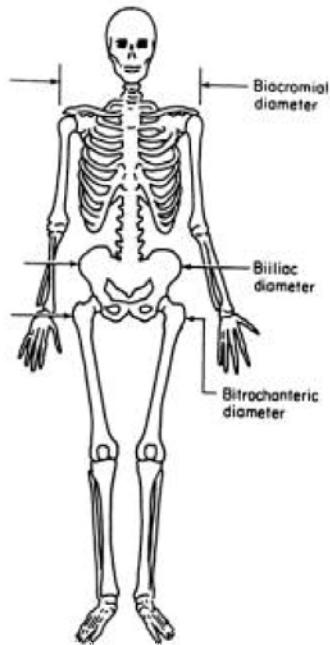
Ejercicio Medidas del Cuerpo Humano

ANÁLISIS DE DATOS

Introducción

Vamos a analizar los datos correspondientes a diferentes medidas del cuerpo humano (perímetros, medidas del esqueleto, edad, peso, estatura y sexo) de 507 personas (247 hombres y 260 mujeres) con el objetivo de estudiar las relaciones entre las diferentes medidas por un lado y encontrar las variables que mejor discriminan a hombres y mujeres, por otro.

Los datos corresponden a personas que acuden periódicamente a un gimnasio y se han obtenido de la revista electrónica *Journal of Statistics Educations*, Vol. 11, Nº 2, "Exploring Relationships in Body Dimensions". Grete Heinz, Louis J. Peterson, Roger W. Johnson & Carter J. Kerk (2003) 11:2, DOI: 10.1080/10691898.2003.11910711



Se disponen de tres grupos de variables:

1. Medidas Esqueléticas:

1. Diámetro de Biacromial (véase figura)
2. Diámetro de Biiliac, o anchura pélvica (figura)
3. Diámetro de Bitrochanteric (Figura)

2

4. Profundidad del pecho entre la espina dorsal y el esternón en nivel del pezón, media inspiración
5. Diámetro del pecho en el nivel del pezón (media expiración)
6. Diámetro del codo, suma de los dos codos
7. Diámetro de la muñeca, suma de las dos muñecas
8. Diámetro de la rodilla, suma de las dos rodillas
9. Diámetro del tobillo, suma de los dos tobillos

2. Medidas musculares:

1. Perímetro de los hombros sobre los músculos deltoides
2. Circunferencia del pecho, línea del pezón en varones y apenas sobre el pecho tejido fino en hembras, media expiración
3. Circunferencia de la cintura, la parte más estrecha del torso debajo de la caja torácica (costillas) promedio de contraído y posición relajada
4. Circunferencia del ombligo (o "abdominal") en ombligo y cresta ilíaca, cresta ilíaca como señal
5. Circunferencia de la cadera en el nivel del diámetro bitrochanteric
6. Circunferencia debajo del doblez glúteo, promedio del perímetro del muslo de la derecha e izquierda
7. Circunferencia de Biceps, doblada, promedio de la derecha e izquierda
8. Circunferencia del antebrazo, extendido, palma para arriba, media de la derecha e izquierda.
9. Circunferencia de la rodilla sobre la pantorrilla, doblada levemente, promedio
10. Circunferencia máxima en los gemelos, promedio
11. Circunferencia mínima del tobillo, promedio
12. Circunferencia mínima de la muñeca, promedio

3. Otras Medidas:

1. Edad (años)
2. Peso (kilogramo)
3. Altura (centímetro)
4. Sexo (1 - varón, 0 - hembra)

Todas las medidas están en centímetros excepto el peso que está en kilogramos y la edad en años. El primer grupo de medidas corresponden al esqueleto (empiezan por A) y junto con la altura proporcionan información de la estructura corporal de cada individuo. El resto de las variables (empiezan por C) están afectadas por la masa corporal y muscular del individuo.

3

Preguntas

1. Realiza el histograma de la variable **Altura**, proporciona la media y la desviación típica. ¿Cuántas personas miden más de 180?
2. Realiza un gráfico con el boxplot de la **Altura** para hombres y mujeres. Describe las diferencias que se observan.
3. Realiza un gráfico con dos histogramas, en la parte superior coloca el histograma de la **Altura** de hombres y en la inferior el de mujeres. Utiliza la misma escala en los dos histogramas. Calcula la media y la desviación típica asociados a cada uno de los histogramas. Interpreta los resultados.
4. Compara las medias de hombres y mujeres para las medidas musculares (variables que empiezan por C, van de la 10 a la 21 ambas inclusive).
 - Realiza un boxplot (múltiple en función de la variable Sexo) para la variable donde haya una mayor diferencia entre la media de hombres y mujeres.
 - Realiza un boxplot (múltiple en función de la variable Sexo) para la variable donde la media de las medidas de las mujeres sea superior a la media de las medidas de los hombres.
5. Contesta a las siguientes cuestiones
 - a. Realiza un gráfico de dispersión del peso de una persona en función de la altura, utiliza un color diferente para hombres y mujeres. Calcula la correlación entre las dos variables con todos los datos, y para hombres y para mujeres. Comenta los resultados.
 - b. Repite el apartado 5 (a), pero utiliza ahora las variables *Peso* y *C_Muslo*. Explica las diferencias entre los resultados de (a) y (b).
 - c. Repite el apartado 5(a) y 5(b) pero utiliza ahora las variables *Altura* y *C_Muslo*. Explica las diferencias entre los resultados de los tres apartados (a), (b) y (c).

Cuerpo Humano

Descriptiva

31 de enero de 2018

0. Lectura de datos

Lecturas de datos.

```
datos = read.table("cuerpo.txt", header=T)
names(datos)

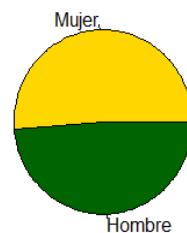
## [1] "A_Hombros"  "A_Pelvis"    "A_Cade"      "AP_Pecho"    "AD_Pecho"
## [6] "A_Codo"     "A_Muneca"   "A_Rodilla"   "A_Tobillo"   "C_hombros"
## [11] "C_Pecho"    "C_Cintura"  "C_abdomen"  "C_Cadera"   "C_Muslo"
## [16] "C_Bicep"   "C_Brazo"    "C_Rodilla"  "C_Gemelo"   "C_Tobillo"
## [21] "C_Muneca"  "Edad"       "Peso"        "Altura"     "Sexo"

datos$Sexo = factor(datos$Sexo, labels=c("Mujer", "Hombre")) # A la variable Sexo le pongo las etiquetas "Mujer", "Hombre"

table(datos$Sexo)

##
## Mujer Hombre
##    260     247

pie(table(datos$Sexo), col=c("gold", "darkgreen"))
```



1. Descriptiva de Altura y Peso

```
tapply(datos$Altura, list(datos$Sexo), mean)
```

```

##      Mujer     Hombre
## 164.8723 177.7453

tapply(datos$Peso,list(datos$Sexo),mean)

##      Mujer     Hombre
## 60.60038 78.14453

tapply(datos$Altura,list(datos$Sexo),sd)

##      Mujer     Hombre
## 6.544602 7.183629

tapply(datos$Peso,list(datos$Sexo),sd)

##      Mujer     Hombre
## 9.615699 10.512890

```

Boxplot Altura

```

# grafico boxplot
summary(datos$Altura[datos$Sexo=="Mujer"])

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    147.2   160.0   164.5   164.9   169.5   182.9

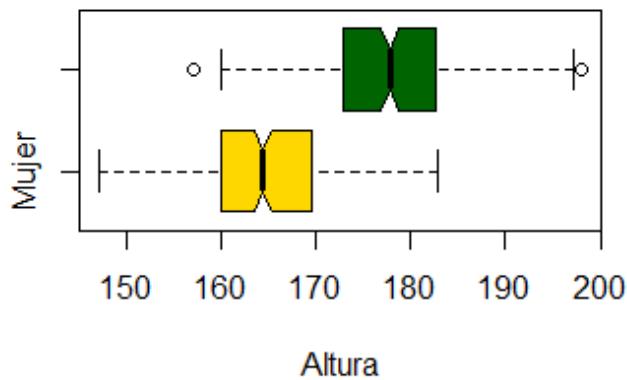
summary(datos$Altura[datos$Sexo=="Hombre"])

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    157.2   172.9   177.8   177.7   182.7   198.1

boxplot(Altura~Sexo, data=datos, notch=TRUE,
        col=(c("gold","darkgreen")),
        main="Altura por Sexo",
        horizontal=TRUE, xlab="Altura")

```

Altura por Sexo



```
# grafico boxplot
```

Boxplot Peso

```
summary(datos$Peso[datos$Sexo=="Mujer"])

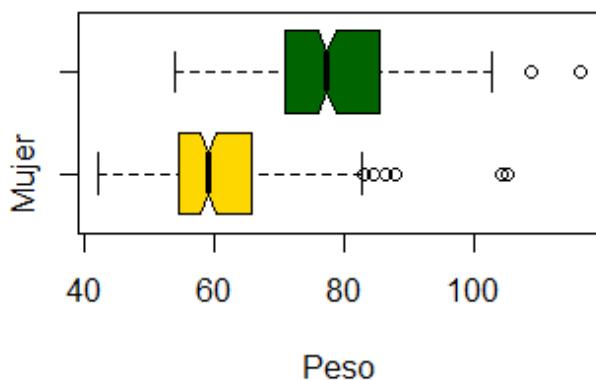
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##     42.0    54.5   59.0     60.6   65.6   105.2

summary(datos$Peso[datos$Sexo=="Hombre"])

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##     53.90   70.95  77.30    78.14   85.50  116.40

boxplot(Peso~Sexo, data=datos, notch=TRUE,
        col=(c("gold","darkgreen")),
        main="Peso por Sexo",
        horizontal=TRUE, xlab="Peso")
```

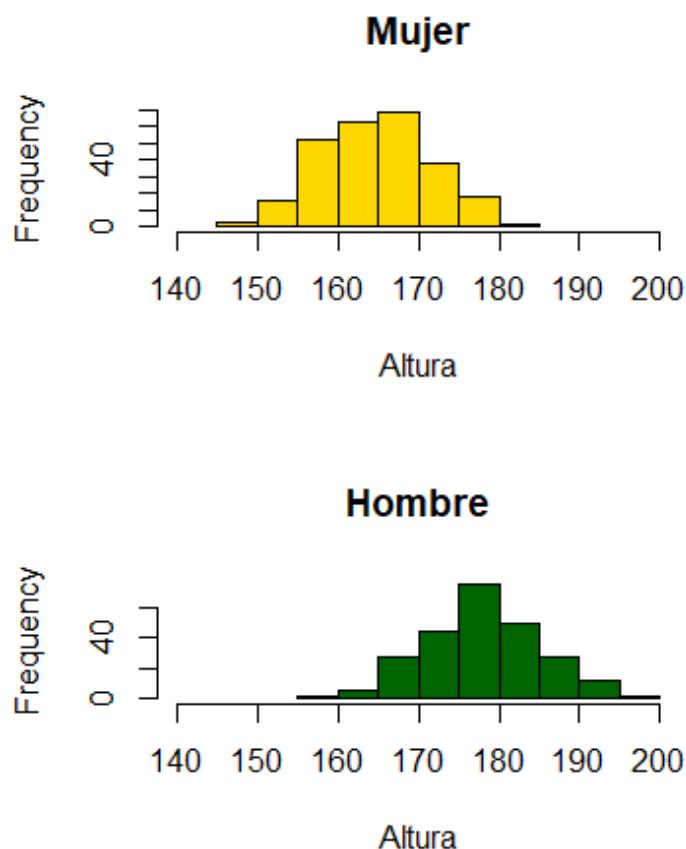
Peso por Sexo



Histogramas Altura

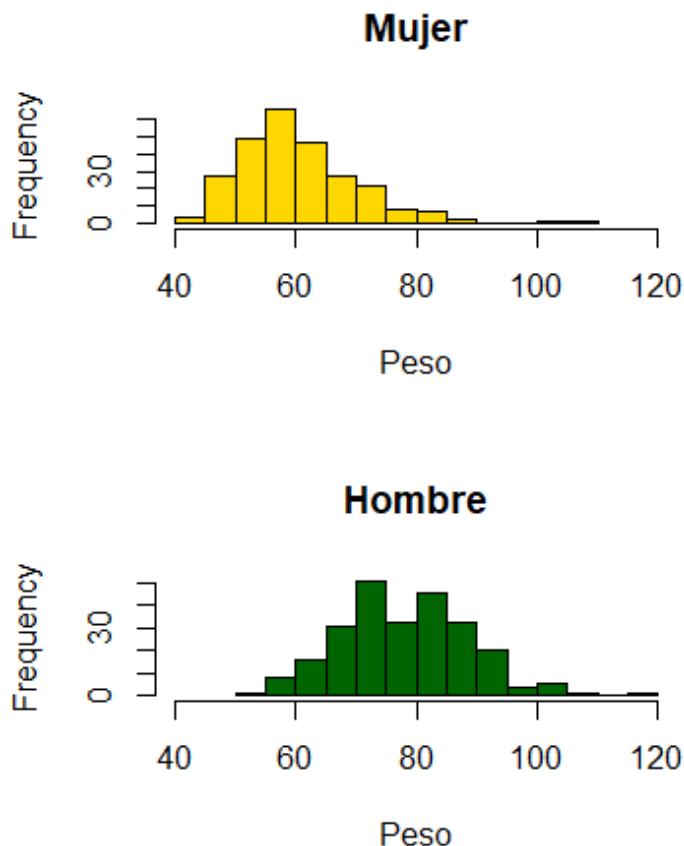
```
# histogramas
par(mfrow=c(2,1))
hist(datos$Altura[datos$Sexo=='Mujer'], nclass=12,
      col='gold', xlim = c(140,200), main='Mujer',
      xlab = "Altura")

hist(datos$Altura[datos$Sexo=='Hombre'], nclass=12,
      col='darkgreen', xlim = c(140,200), main='Hombre',
      xlab = "Altura")
```



```
# histogramas Peso
par(mfrow=c(2,1))
hist(datos$Peso[datos$Sexo=='Mujer'], nclass=12,
      col='gold', xlim = c(40,120), main='Mujer',
      xlab = "Peso")

hist(datos$Peso[datos$Sexo=='Hombre'], nclass=12,
      col='darkgreen', xlim = c(40,120), main='Hombre',
      xlab = "Peso")
```



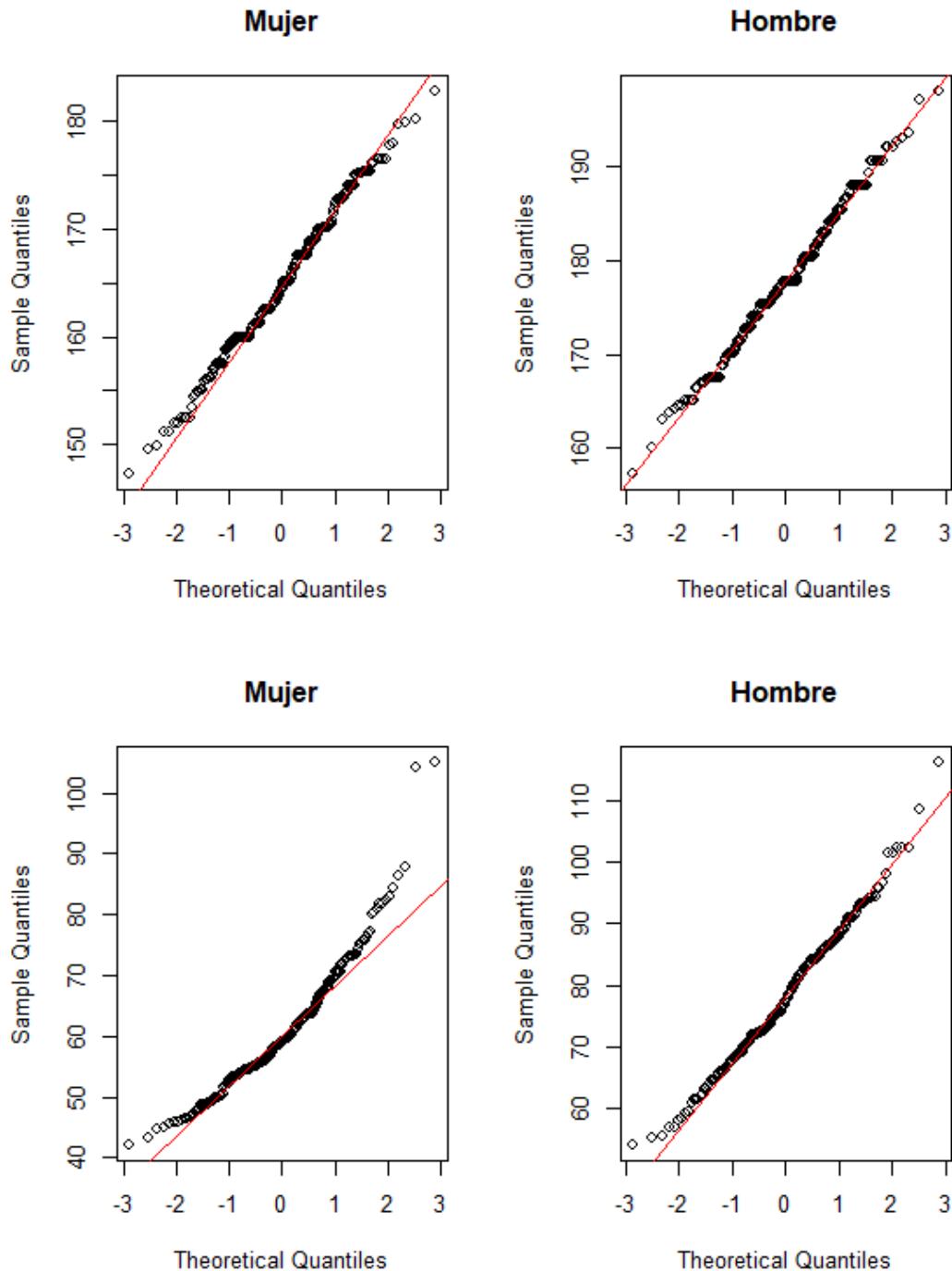
2. Gráficos qq

```
# qqplots Altura
par(mfrow=c(2,2))
qqnorm(datos$Altura[datos$Sexo=='Mujer'],main='Mujer')
qqline(datos$Altura[datos$Sexo=='Mujer'],col='red')

qqnorm(datos$Altura[datos$Sexo=='Hombre'],main='Hombre')
qqline(datos$Altura[datos$Sexo=='Hombre'],col='red')

# qqplots Peso
# par(mfrow=c(2,1))
qqnorm(datos$Peso[datos$Sexo=='Mujer'],main='Mujer')
qqline(datos$Peso[datos$Sexo=='Mujer'],col='red')

qqnorm(datos$Peso[datos$Sexo=='Hombre'],main='Hombre')
qqline(datos$Peso[datos$Sexo=='Hombre'],col='red')
```



3. Otras Variables

```
mHom=sapply(datos[datos$Sexo=="Hombre",1:24],mean)
mMuj=sapply(datos[datos$Sexo=="Mujer",1:24],mean)
sHom=sapply(datos[datos$Sexo=="Hombre",1:24],sd)
sMuj=sapply(datos[datos$Sexo=="Mujer",1:24],sd)
```

```

mHom1= mHom[1:9]
mMuj1= mMuj[1:9]
mHom2= mHom[10:24]
mMuj2= mMuj[10:24]
o1 = order(mHom1)
o2 = order(mHom2)
t1=rbind(mMuj1[o1],mHom1[o1])
t2=rbind(mMuj2[o2],mHom2[o2])
d1=as.data.frame(t(t1))
d2=as.data.frame(t(t2))
names(d1)=c("Mujer","Hombre")
names(d2)=c("Mujer","Hombre")
print(d1,digits=4)

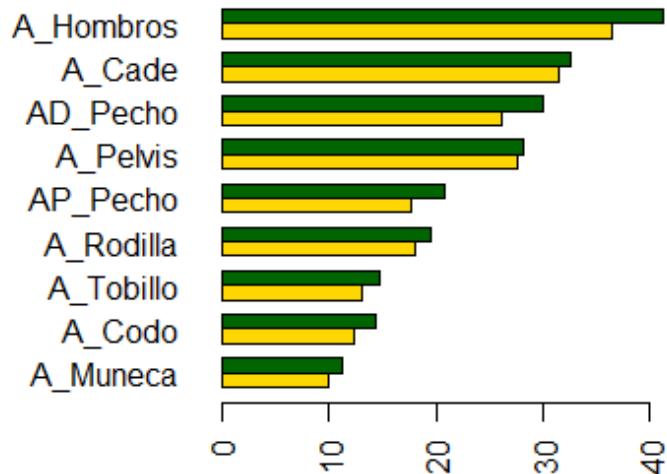
##           Mujer Hombre
## A_Muneca   9.874 11.25
## A_Codo     12.367 14.46
## A_Tobillo  13.027 14.74
## A_Rodilla  18.097 19.56
## AP_Pecho   17.725 20.81
## A_Pelvis   27.582 28.09
## AD_Pecho   26.097 29.95
## A_Cade     31.462 32.53
## A_Hombros  36.503 41.24

print(d2,digits=4)

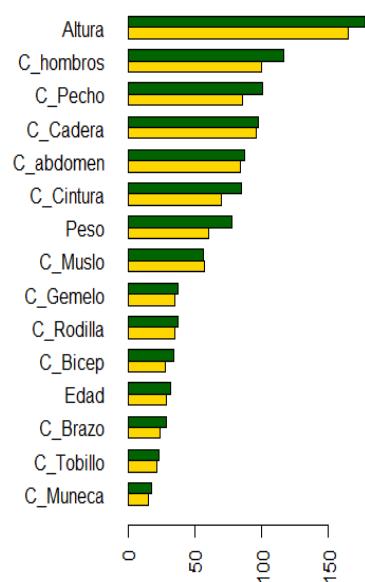
##           Mujer Hombre
## C_Muneca   15.06 17.19
## C_Tobillo  21.21 23.16
## C_Brazo    23.76 28.24
## Edad       28.77 31.67
## C_Bicep    28.10 34.40
## C_Rodilla  35.26 37.20
## C_Gemelo   35.01 37.21
## C_Muslo    57.20 56.50
## Peso        60.60 78.14
## C_Cintura  69.80 84.53
## C_abdomen  83.75 87.66
## C_Cadera   95.65 97.76
## C_Pecho    86.06 100.99
## C_hombros  100.30 116.50
## Altura     164.87 177.75

par(mar=c(5.1,6.1,4.1,2.1))
barplot(t1,beside=TRUE,horiz = TRUE,las=2,col=c("gold","darkgreen"))

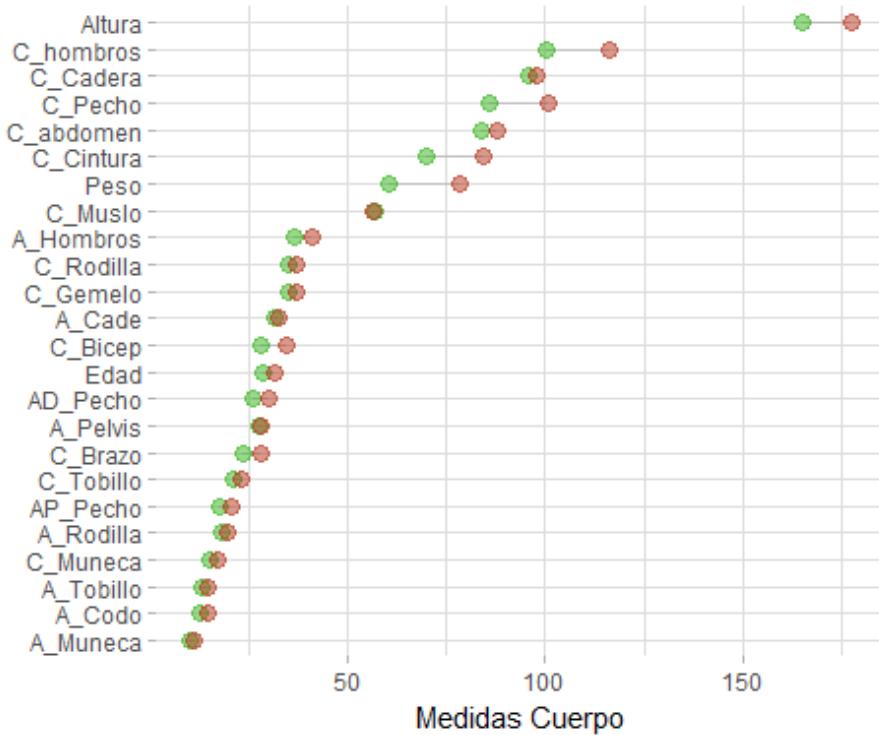
```



```
par(mar=c(5.1,6.1,4.1,2.1))
barplot(t2,beside=TRUE,horiz = TRUE,las=2,col=c("gold","darkgreen"))
```



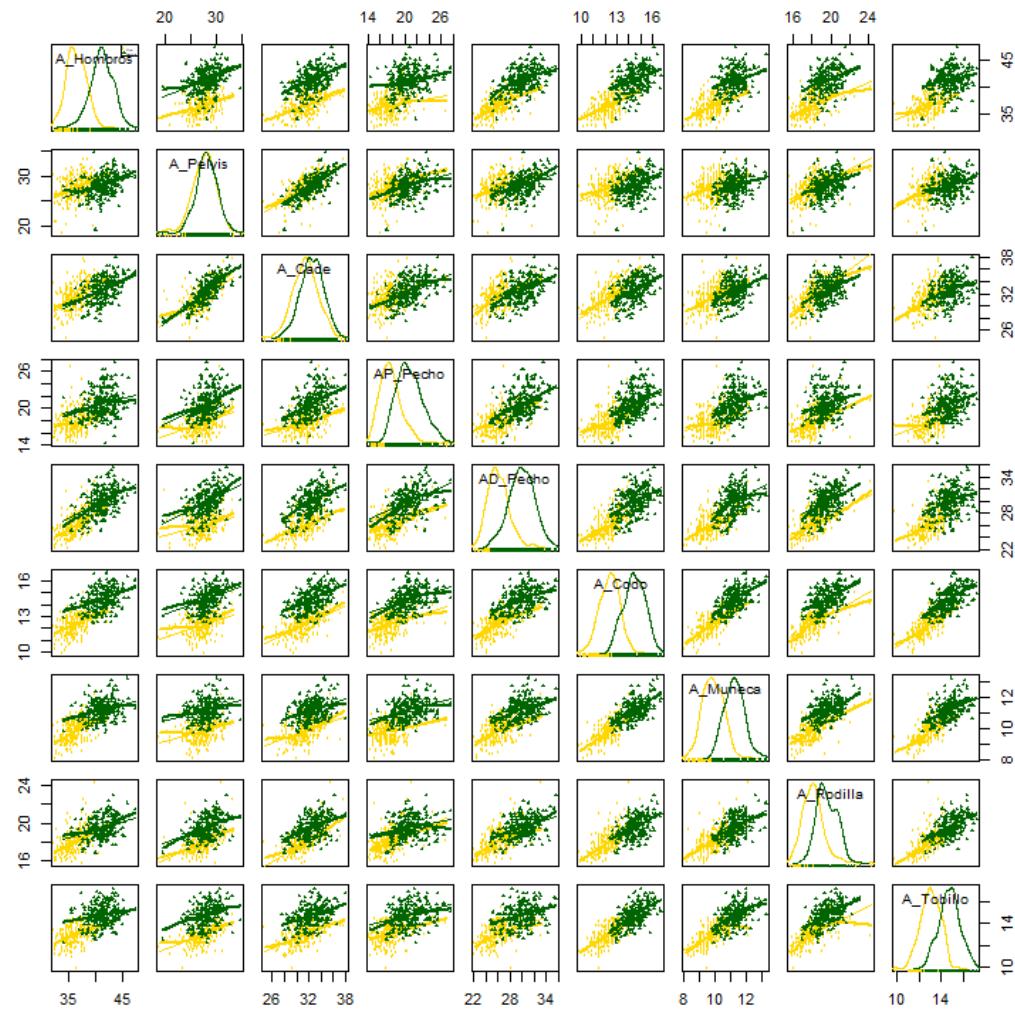
4. Otras Variables



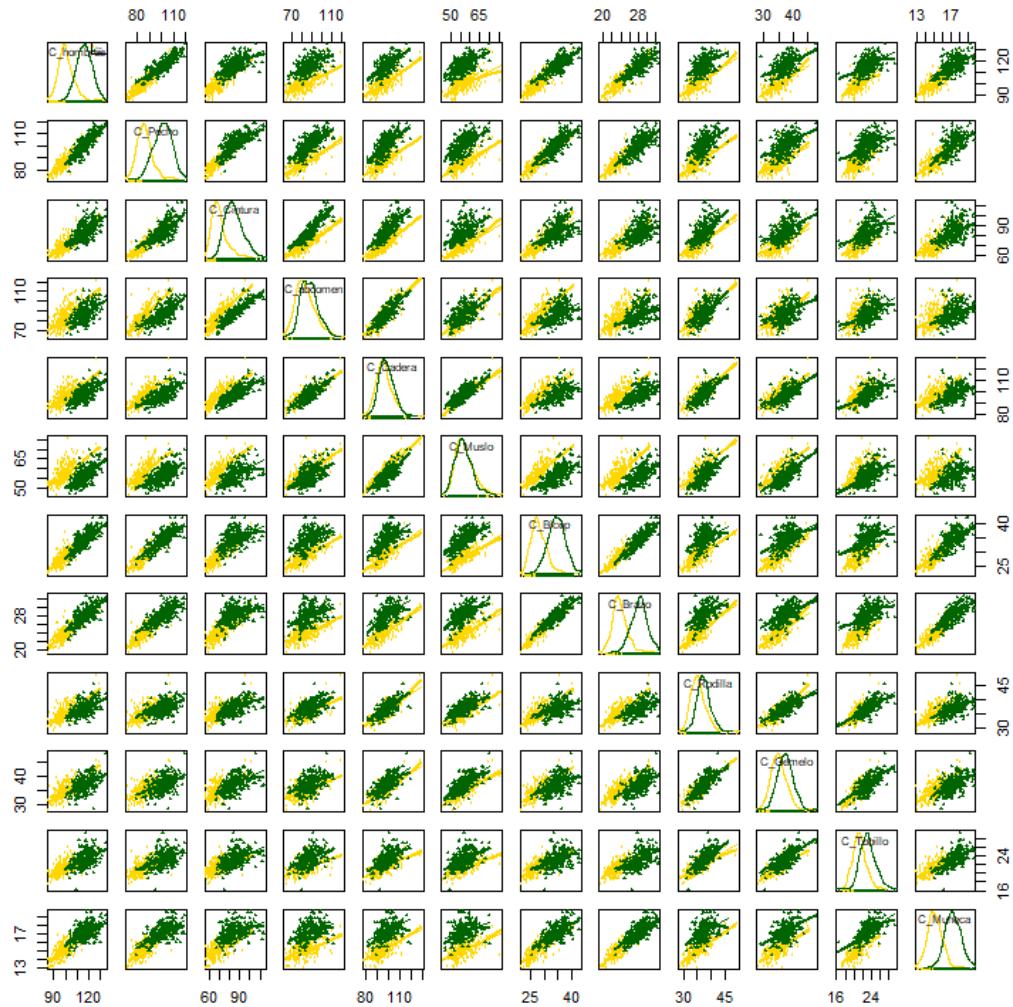
5 y 6 Correlaciones

```
library(car)

scatterplotMatrix(datos[1:9],groups=datos$Sexo,cex=.2,by.groups = TRUE,
l=c("gold","darkgreen"))
```



```
library(car)
scatterplotMatrix(datos[10:21], groups=datos$Sexo, cex=.2, by.groups = TRUE,
col=c("gold","darkgreen"))
```



```
# Calculo de las matrices de correlaciones
```

```
M1=cor(datos[datos$Sexo=="Mujer",1:9])
H1=cor(datos[datos$Sexo=="Hombre",1:9])
M2=cor(datos[datos$Sexo=="Mujer",10:21])
H2=cor(datos[datos$Sexo=="Hombre",10:21])
```

```
# Cambio las etiquetas de las columnas para reducir el espacio
```

```
nam1=abbreviate(names(datos)[1:9])
nam2=abbreviate(names(datos)[10:21])
colnames(M1)=nam1
colnames(H1)=nam1
colnames(M2)=nam2
colnames(H2)=nam2
rownames(M2)=nam2
rownames(H2)=nam2
```

```

# Muestro las matrices de correlaciones
print(M1, digits=2)

##          A_Hm A_P1 A_Cad AP_P AD_P A_Cod A_Mn A_Rd A_Tb
## A_Hombros 1.00 0.36  0.48 0.21 0.54  0.48 0.47 0.48 0.39
## A_Pelvis   0.36 1.00  0.63 0.35 0.27  0.38 0.29 0.49 0.48
## A_Cade     0.48 0.63  1.00 0.35 0.50  0.57 0.45 0.67 0.53
## AP_Pecho   0.21 0.35  0.35 1.00 0.36  0.36 0.26 0.43 0.31
## AD_Pecho   0.54 0.27  0.50 0.36 1.00  0.53 0.47 0.52 0.41
## A_Codo     0.48 0.38  0.57 0.36 0.53  1.00 0.68 0.65 0.65
## A_Muneca   0.47 0.29  0.45 0.26 0.47  0.68 1.00 0.60 0.60
## A_Rodilla  0.48 0.49  0.67 0.43 0.52  0.65 0.60 1.00 0.60
## A_Tobillo  0.39 0.48  0.53 0.31 0.41  0.65 0.60 0.60 1.00

print(H1, digits=2)

##          A_Hm A_P1 A_Cad AP_P AD_P A_Cod A_Mn A_Rd A_Tb
## A_Hombros 1.00 0.35  0.46 0.22 0.47  0.37 0.27 0.34 0.19
## A_Pelvis   0.35 1.00  0.72 0.38 0.44  0.36 0.28 0.39 0.32
## A_Cade     0.46 0.72  1.00 0.47 0.50  0.48 0.38 0.45 0.37
## AP_Pecho   0.22 0.38  0.47 1.00 0.46  0.40 0.34 0.23 0.30
## AD_Pecho   0.47 0.44  0.50 0.46 1.00  0.43 0.44 0.41 0.32
## A_Codo     0.37 0.36  0.48 0.40 0.43  1.00 0.60 0.52 0.61
## A_Muneca   0.27 0.28  0.38 0.34 0.44  0.60 1.00 0.49 0.49
## A_Rodilla  0.34 0.39  0.45 0.23 0.41  0.52 0.49 1.00 0.55
## A_Tobillo  0.19 0.32  0.37 0.30 0.32  0.61 0.49 0.55 1.00

print(M2, digits=2)

##          C_hm C_Pc C_Cn C_bd C_Cd C_Ms C_Bc C_Br C_Rd C_Gm C_Tb C_Mn
## C_hm 1.00 0.83 0.73 0.61 0.68 0.63 0.74 0.75 0.65 0.63 0.56 0.66
## C_Pc  0.83 1.00 0.86 0.77 0.74 0.68 0.82 0.77 0.62 0.58 0.51 0.64
## C_Cn  0.73 0.86 1.00 0.84 0.81 0.73 0.80 0.71 0.63 0.58 0.49 0.55
## C_bd  0.61 0.77 0.84 1.00 0.83 0.70 0.75 0.64 0.61 0.52 0.49 0.49
## C_Cd  0.68 0.74 0.81 0.83 1.00 0.90 0.77 0.75 0.76 0.69 0.60 0.62
## C_Ms  0.63 0.68 0.73 0.70 0.90 1.00 0.75 0.73 0.76 0.73 0.58 0.57
## C_Bc  0.74 0.82 0.80 0.75 0.77 0.75 1.00 0.87 0.68 0.67 0.57 0.68
## C_Br  0.75 0.77 0.71 0.64 0.75 0.73 0.87 1.00 0.75 0.74 0.64 0.81
## C_Rd  0.65 0.62 0.63 0.61 0.76 0.76 0.68 0.75 1.00 0.80 0.70 0.70
## C_Gm  0.63 0.58 0.58 0.52 0.69 0.73 0.67 0.74 0.80 1.00 0.74 0.65
## C_Tb  0.56 0.51 0.49 0.49 0.60 0.58 0.57 0.64 0.70 0.74 1.00 0.67
## C_Mn  0.66 0.64 0.55 0.49 0.62 0.57 0.68 0.81 0.70 0.65 0.67 1.00

print(H2, digits=2)

##          C_hm C_Pc C_Cn C_bd C_Cd C_Ms C_Bc C_Br C_Rd C_Gm C_Tb C_Mn
## C_hm 1.00 0.83 0.58 0.54 0.65 0.61 0.76 0.70 0.50 0.49 0.47 0.54
## C_Pc  0.83 1.00 0.71 0.68 0.68 0.60 0.78 0.70 0.48 0.48 0.47 0.55
## C_Cn  0.58 0.71 1.00 0.88 0.80 0.56 0.47 0.42 0.57 0.52 0.48 0.37
## C_bd  0.54 0.68 0.88 1.00 0.81 0.55 0.47 0.43 0.56 0.46 0.51 0.39
## C_Cd  0.65 0.68 0.80 0.81 1.00 0.79 0.57 0.54 0.70 0.65 0.59 0.45
## C_Ms  0.61 0.60 0.56 0.55 0.79 1.00 0.66 0.61 0.66 0.70 0.52 0.38

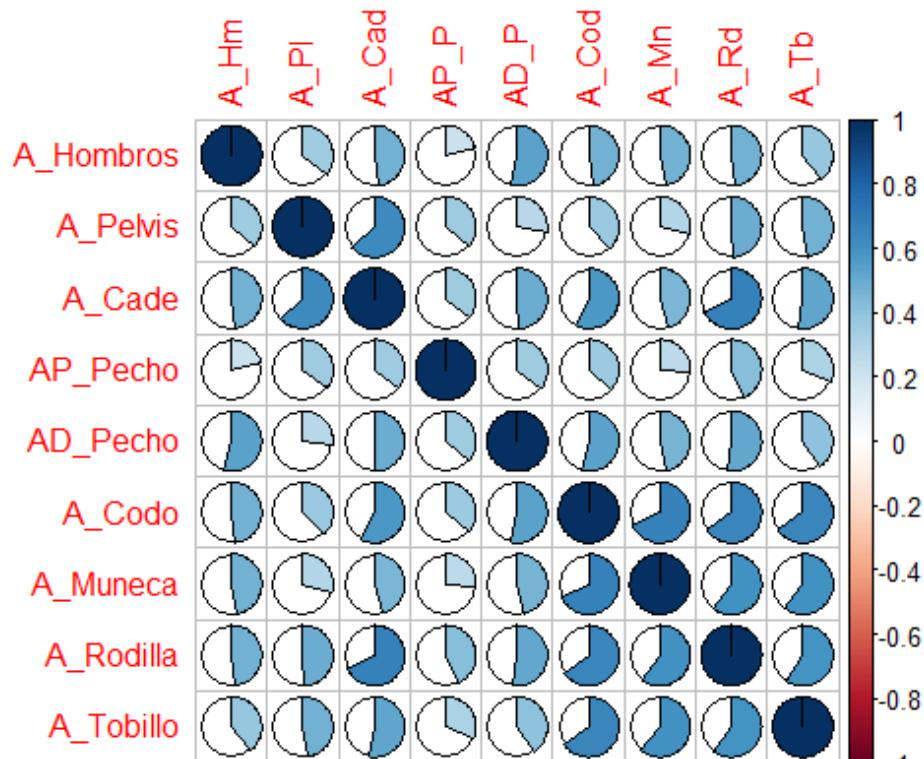
```

```
## C_Bc 0.76 0.78 0.47 0.47 0.57 0.66 1.00 0.86 0.43 0.47 0.43 0.61
## C_Br 0.70 0.70 0.42 0.43 0.54 0.61 0.86 1.00 0.53 0.55 0.51 0.71
## C_Rd 0.50 0.48 0.57 0.56 0.70 0.66 0.43 0.53 1.00 0.73 0.69 0.50
## C_Gm 0.49 0.48 0.52 0.46 0.65 0.70 0.47 0.55 0.73 1.00 0.70 0.54
## C_Tb 0.47 0.47 0.48 0.51 0.59 0.52 0.43 0.51 0.69 0.70 1.00 0.63
## C_Mn 0.54 0.55 0.37 0.39 0.45 0.38 0.61 0.71 0.50 0.54 0.63 1.00

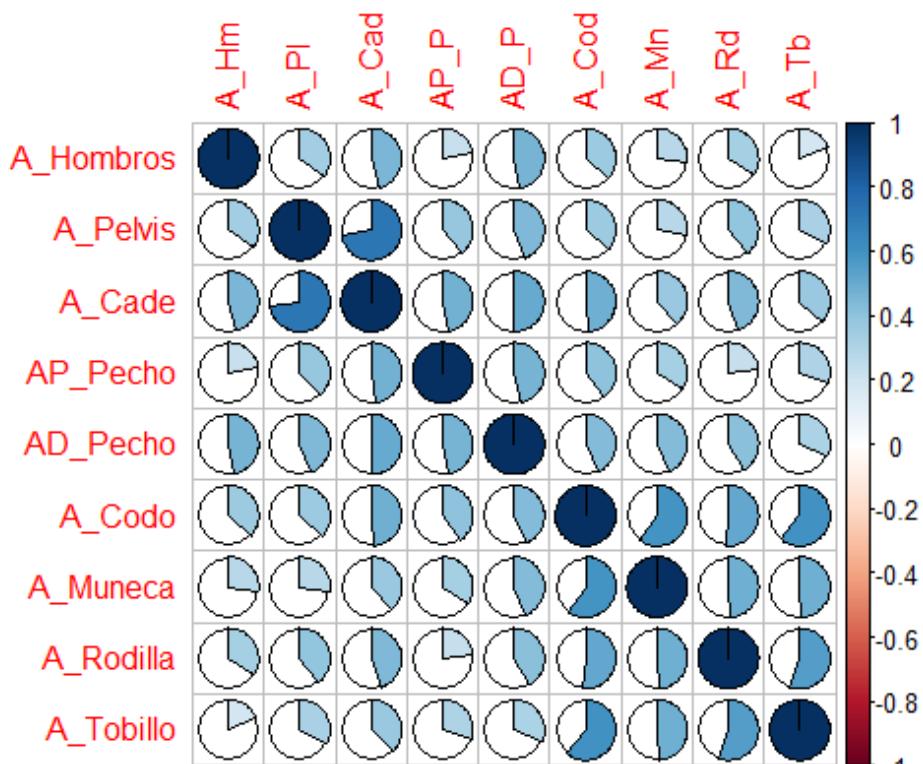
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.4.2

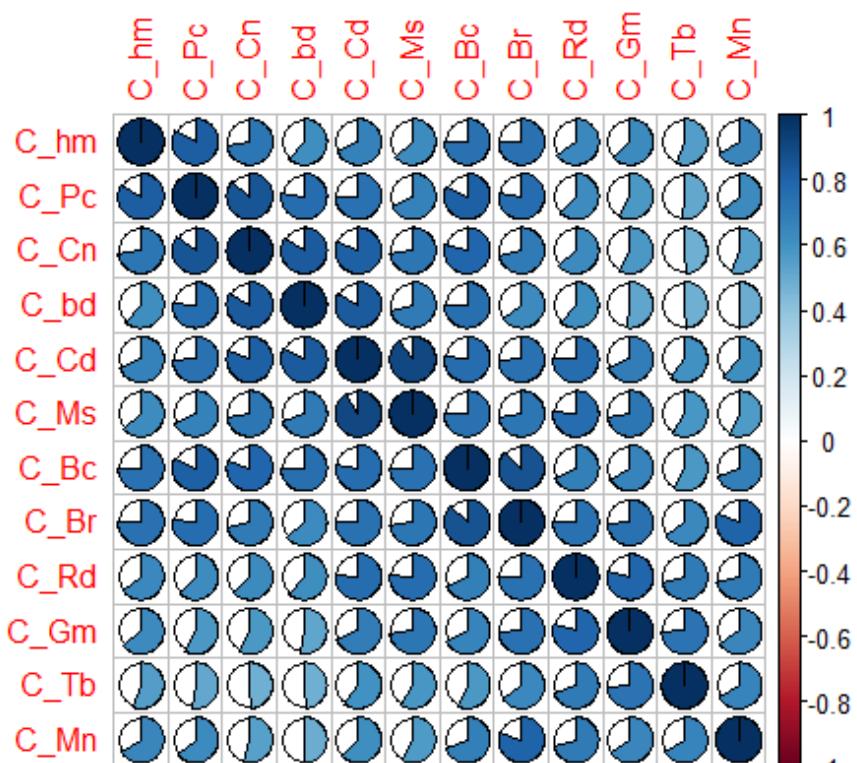
corrplot(M1,method="pie")
```



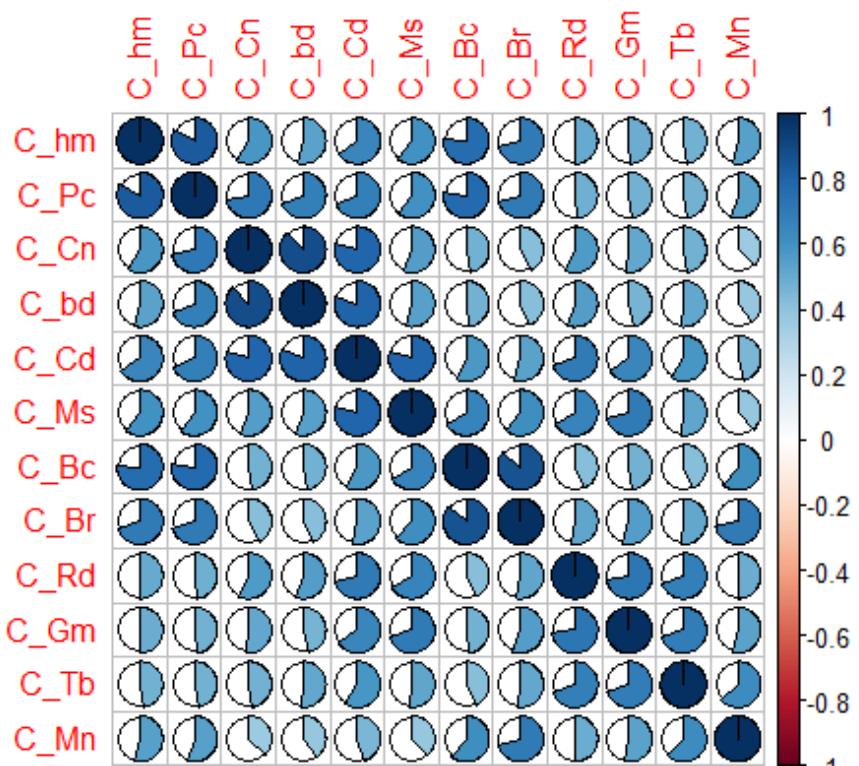
```
corrplot(H1,method="pie")
```



```
corrplot(M2,method="pie")
```

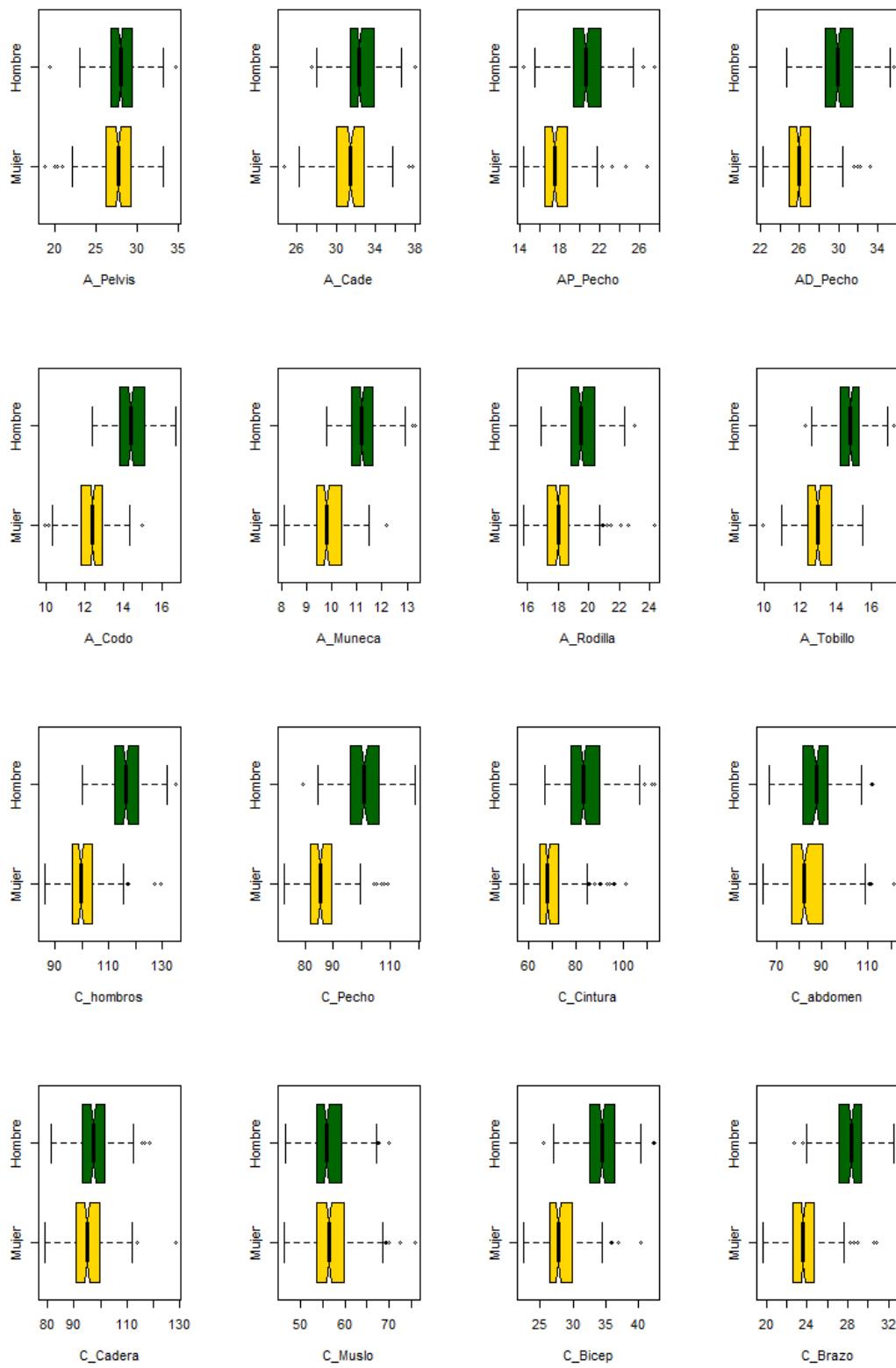


```
corrplot(H2,method="pie")
```

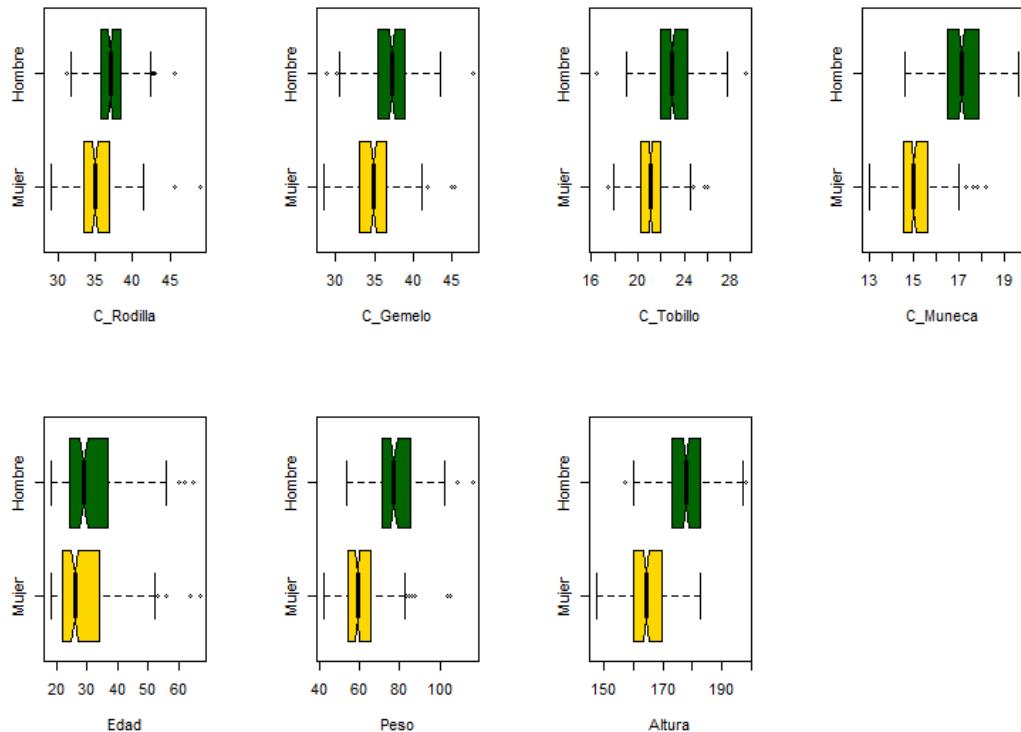


7. Boxplots de todas las variables

```
par(mfrow=c(4,4))
for (k in (2:17)){
  boxplot(datos[,k]~Sexo, data=datos, notch=TRUE,
           col=(c("gold","darkgreen")),
           horizontal=TRUE, xlab=names(datos)[k])
}
```



```
par(mfrow=c(2,4))
for (k in (18:24)){
  boxplot(datos[,k]~Sexo, data=datos, notch=TRUE,
           col=(c("gold","darkgreen")),
           horizontal=TRUE, xlab=names(datos)[k])
}
```



8 y 9. Correlación del Peso y las variables Musculares (Hombres y Mujeres)

```
sel = c(23,24,10:21)
M = cor(datos[datos$Sexo=='Mujer',sel])
H = cor(datos[datos$Sexo=='Hombre',sel])
M1 = M[,1]
H1 = H[,1]
d = as.data.frame(rbind(mujer=M1,hombre=H1))
print(t(d),digits=3)

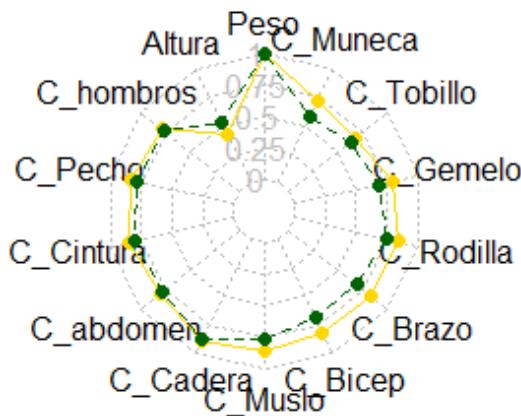
##          mujer hombre
## Peso      1.000  1.000
## Altura    0.431  0.535
## C_hombros 0.791  0.764
## C_Pecho   0.839  0.794
## C_Cintura 0.858  0.805
```

```

## C_abdomen 0.797 0.776
## C_Cadera  0.902 0.877
## C_Muslo   0.857 0.772
## C_Bicep   0.820 0.688
## C_Brazo   0.835 0.688
## C_Rodilla 0.840 0.744
## C_Gemelo  0.795 0.691
## C_Tobillo 0.669 0.636
## C_Muneca  0.721 0.576

library(fmsb)
# To use the fmsb package, I have to add 2 lines to the dataframe: the max and min of each topic to show on the plot!
d=rbind(rep(1,14) , rep(0.,14) , d)
colors_border=c('gold','darkgreen')
radarchart(d,axistype=1 ,pcol=colors_border,
            cglcol='grey',axislabcol="grey",
            caxislabels=seq(0,1,.25))

```



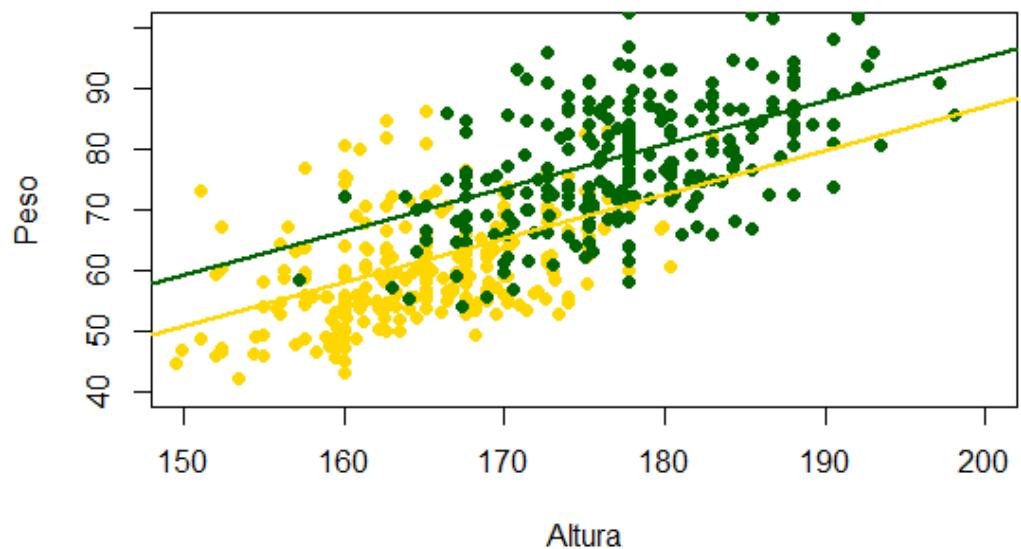
10. Estudio de Peso versus Estatura

```

plot(x=NULL, y=NULL, xlim=c(150,200),ylim=c(40,100),xlab='Altura',ylab='Peso')
#plot(datos$Altura,datos$Peso,col=dat)
points(datos$Altura[datos$Sexo=='Mujer'],datos$Peso[datos$Sexo=='Mujer'],
       pch=19,col='gold')

```

```
points(datos$Altura[datos$Sexo=='Hombre'], datos$Peso[datos$Sexo=='Hombre'],
],pch=19,col='darkgreen')
abline(c(-56.94+8.36,0.719),col='darkgreen',lw=2)
abline(c(-56.94,0.719),col='gold',lw=2)
```



Elecciones Comunidad de Madrid 2021

Análisis de Datos

Ejercicio de análisis clúster: MADRID 2021

En el archivo “Madrid_2021.txt” se proporciona los resultados por barrios de Madrid Capital de las elecciones a la Asamblea de Madrid celebradas el 4 de Mayo de 2021. Para cada barrio se proporciona el nombre, el censo y el porcentaje sobre el censo de votos a los partidos más votados: Vox, PP, Ciudadanos (Cs), PSOE, Más Madrid y Unidos-Podemos. Además se proporciona el porcentaje de votos a otras candidaturas y el porcentaje de abstención.

Los datos se han obtenido de la página web:

<https://datos.gob.es/es/catalogo/l01280796-elecciones-asamblea-de-madrid-1983-2019>

La variable **censo** se proporciona para completar la información, es muy relevante para el análisis de los resultados, pero en este ejercicio no se utiliza. Nos centraremos en las variables porcentaje de votos: vox, pp, cs, psoe, masmadrid, podemos, otros y abstención.

PREGUNTAS:

1. Describe mediante un boxplot múltiple los porcentajes de votos a **vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion**. Interpreta brevemente la gráfica. (Nota: si X es una tabla con múltiples columnas, boxplot(X) representa el diagrama de cajas de cada columna)
2. Describe cada variable (univariante)
3. Obten la matriz de correlaciones de las variables **vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion**. Representa mediante gráficos de dispersión las relaciones entre las variables. Interpreta los resultados.
4. Resume brevemente las conclusiones del análisis.

Análisis de Datos

2. Repaso Regresión con R

Datos

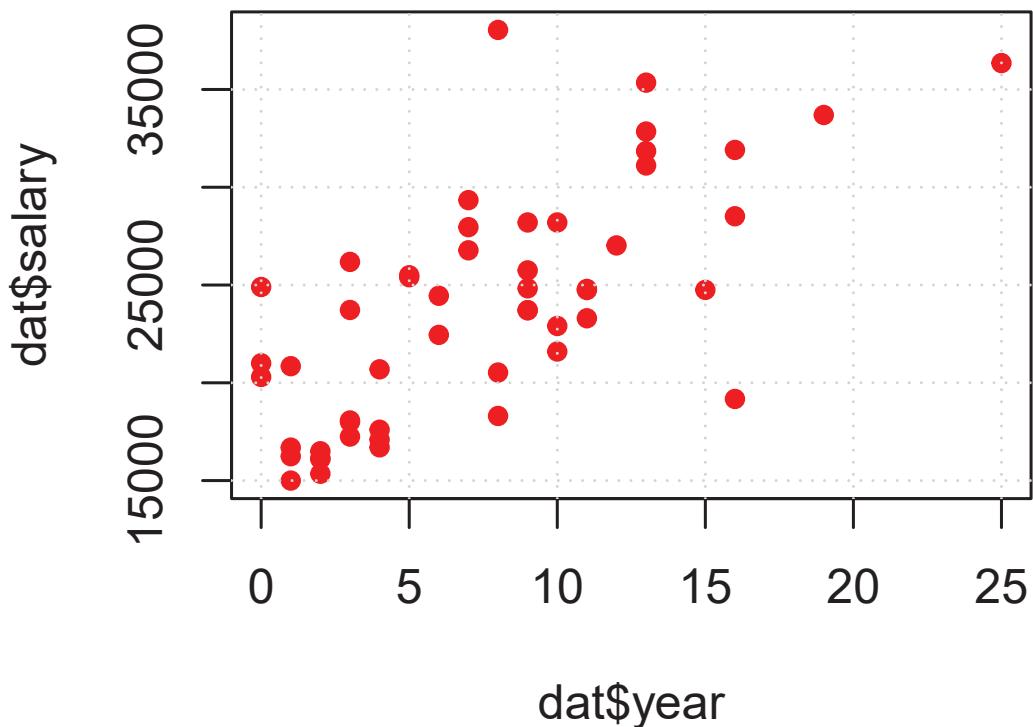
Para leer datos utilizo la instrucción `read.table()` y para ver las primeras files (observaciones) de los datos leidos, utilizo la instrucción `head()`

```
dat = read.table("data/salary.txt", header=T) # (1)  
head(dat) # (2)
```

```
##      degree rank      sex year ysdeg salary  
## 1 Masters Prof     Male   25    35  36350  
## 2 Masters Prof     Male   13    22  35350  
## 3 Masters Prof     Male   10    23  28200  
## 4 Masters Prof Female    7    27  26775  
## 5      PhD Prof     Male   19    30  33696  
## 6 Masters Prof     Male   16    21  28516
```

Relación entre variables: correlación y regresión lineal

```
par(mar=c(6.1, 4.1, 0.5, 2.1))
plot(dat$year,dat$salary, col="red", pch=19,cex=.8)
grid()
```



Correlación

La medida de asociación más habitual es la correlación

$$r = \frac{s_{xy}}{s_x s_y}$$

```
cor(dat$year,dat$salary)
```

```
## [1] 0.700669
```

Regresión

El modelo de regresión profundiza en la relación lineal

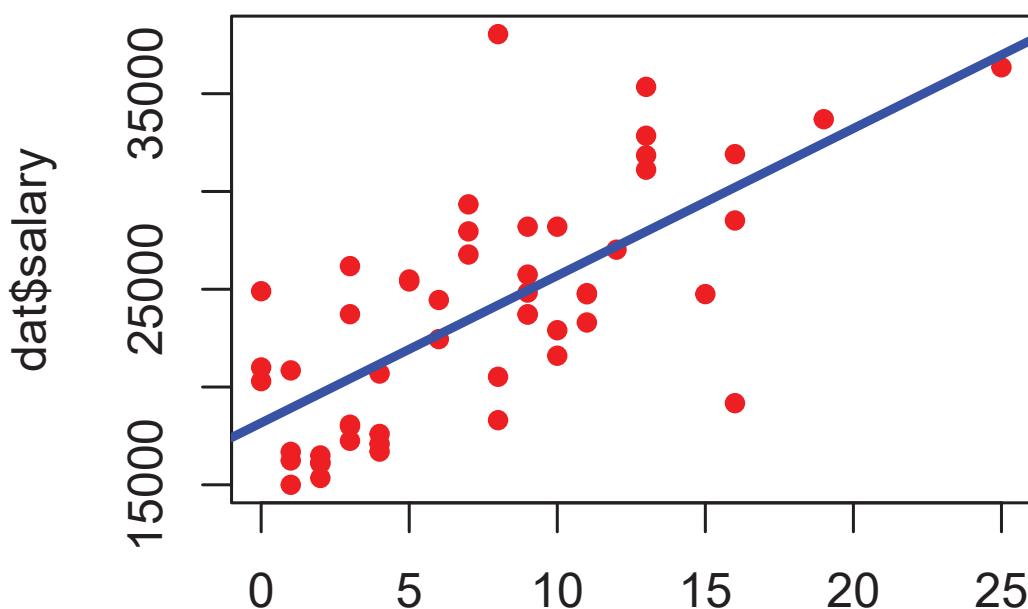
```
m = lm(salary ~ year, data = dat)
summary(m)
```

```
## 
## Call:
## lm(formula = salary ~ year, data = dat)
## 
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -11035.9 -3172.4  -561.7  3185.8 13856.5 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 18166.1    1003.7   18.100 < 2e-16 ***
## year        752.8     108.4    6.944 7.34e-09 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 4264 on 50 degrees of freedom
## Multiple R-squared:  0.4909, Adjusted R-squared:  0.4808 
## F-statistic: 48.22 on 1 and 50 DF,  p-value: 7.341e-09
```

La ecuación del modelo es:

$$\text{salary}_i = 18166.1 + 752.8 \times \text{year}_i + e_i$$

```
par(mar=c(6.1, 4.1, 0.5, 2.1))
plot(dat$year,dat$salary, col="red", pch=19,cex=.8)
abline(m,col = "blue",lwd = 3)
```



Interpretación del modelo:

- $\hat{\beta}_1 = 752.8$: Mide la relación entre la antigüedad (year) y el salario. Se ha estimado que el salario aumenta 752.8 \$ por cada año de antigüedad. El parámetro β_1 se ha estimado con un error de 108.4 \$/año, y los resultados indican que β_1 es significativamente distinto de cero (p-value=7.34e-09).
- Conocida la antigüedad (year) la ecuación anterior predice el salario anual de un profesor con un error medio de 4264 dólares.

$$\hat{s}_R = 4264 \text{ \$}$$

Interpretación del modelo:Continuación

- $R^2 = 0.4909$: La antigüedad explica el 49.09% de la variabilidad del salario. Se observa que personas con los mismos años de antigüedad tienen salarios diferentes. La variable “year” (antigüedad), solo explica el 49% del salario. Hay otras variables que influyen y no están en la ecuación que explican el $100 - 49.09 = 50.91\%$.
- $\beta_0 = 18166.1\$$ es el salario anual que estima el modelo para un profesor sin antigüedad (year=0). En este ejemplo se puede interpretar, pero en la mayoría de los casos es un parámetro que no tiene fácil interpretación. En general es un parámetro con menor interés.

Regresión Múltiple

- Vamos a poner otra variable (`ysdeg`)

```
m1 = lm(salary ~ year + ysdeg, data = dat)
summary(m1)
```

```
## 
## Call:
## lm(formula = salary ~ year + ysdeg, data = dat)
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -10321.2 -2347.2  -332.7  2298.8 12240.9 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16555.7   1052.4   15.732 < 2e-16 ***
## year        489.3    129.6    3.777 0.000431 ***  
## ysdeg       222.2     69.8    3.184 0.002525 **  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 3921 on 49 degrees of freedom
## Multiple R-squared:  0.5782, Adjusted R-squared:  0.561 
## F-statistic: 33.58 on 2 and 49 DF,  p-value: 6.532e-10
```

$$\text{salary}_i = 16555.7 + 489.3 \text{year}_i + 222.2 \text{ysdeg}_i + e_i$$

Interpretación del modelo:

- La nueva variable aumenta la capacidad explicativa del modelo. Con las dos variables (`year` y `ysdeg`, explicamos el **57.8%**).
- El salario se predice con un error medio de **3921 dólares** con el nuevo modelo (se ha reducido respecto al modelo anterior).
- Al introducir la nueva variable la estimación del efecto de la antigüedad (`year`) ha variado, ahora el parámetro $\hat{\beta}_1$ vale 489.3 \$/año. El efecto de `ysdeg` (los años desde que obtuvo el título de Master o Doctorado) es de 222.2 \$ por año. Los dos coeficientes son significativamente distintos de cero. SE dice que las variables `year` y `ysdeg` tienen un efecto significativo en el salario.

Desgraciadamente no podemos hacer una representación gráfica fácil del modelo.

Variables cualitativas en el modelo

Introducimos la variable `sexo` en el modelo que es una **variable cualitativa** que toma el valor 0 para mujeres y 1 para hombres.

```
m2 = lm(salary ~ year + ysdeg + sex, data = dat)
summary(m2)
```

```
## 
## Call:
## lm(formula = salary ~ year + ysdeg + sex, data = dat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -10298.6 -2175.5 - 383.5  2106.0 12736.5 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16120.84    1325.78   12.160 2.89e-16 ***
## year        456.17     143.90    3.170  0.00265 **  
## ysdeg       230.76     72.02    3.204  0.00241 **  
## sexMale     746.41    1366.81    0.546  0.58753    
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 3949 on 48 degrees of freedom
## Multiple R-squared:  0.5808, Adjusted R-squared:  0.5546 
## F-statistic: 22.17 on 3 and 48 DF,  p-value: 3.762e-09
```

$$\text{salary}_i = 16120.84 + 456.17 \text{year}_i + 230.76 \text{ysdeg}_i + 746.41 \text{sexo}_i + e_i$$

Interpretación del modelo:

- ① Según esto a igualdad de antiguedad y edad, los hombres cobran 746 dólares más al año que las mujeres. Pero el efecto no es significativo. Miramos que esta variable mejora muy poquito R², ahora es 58.08%. (Ojo, al añadir una variable al modelo **siempre** aumenta R², por tanto el criterio R² para comparar modelos no es útil. Para corregir ese problema se utiliza Adjusted-R². Se aprecia que el valor R² ajustado en este caso es peor el obtenido con el modelo m2).
- ② El resto de los parámetros del modelo no han cambiado mucho. La interpretación es similar al modelo anterior.

Variable cualitativa con más de dos grupos

Vamos a introducir la categoría del profesor (**rank**) en el modelo.

```
m3 = lm(salary ~ year + ysdeg + rank, data = dat)
summary(m3)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg + rank, data = dat)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -3413.7 -1218.5 -182.7  742.0  9483.3 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16317.46   667.86  24.433 < 2e-16 ***
## year        400.46    81.50   4.914 1.13e-05 ***
## ysdeg       -34.32    54.54  -0.629   0.532    
## rankAssoc   4619.12  1054.02   4.382 6.55e-05 ***
## rankProf    9864.30  1120.27   8.805 1.65e-11 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2417 on 47 degrees of freedom
## Multiple R-squared:  0.8462, Adjusted R-squared:  0.8331 
## F-statistic: 64.64 on 4 and 47 DF,  p-value: < 2.2e-16
```

$$\text{salary}_i = 16317.5 + 400.5 \text{year}_i - 34.3 \text{ysdeg}_i + 4619.1 \text{Assoc}_i + 9864.3 \text{Prof}_i + e_i$$

Interpretación:

- El modelo explica el 84.6% de la variabilidad salarial.
- La variable ysdeg no es significativa. Esto es frecuente en los modelos de regresión, al añadir nuevas variables al modelo, otras variables presentes en él dejan de ser significativas. Esto se debe a que las variables explicativas del modelo están relacionadas entre ellas. A este fenómeno se le denomina multicolinealidad (relaciones lineales entre los regresores).
- El error del modelo se ha reducido a 2417 dólares
- El salario de un profesor ASSISTANT tiene como base 16317 dólares anuales, más 400 dólares por año contratado. Los ASSOCIATES cobran 4619 más al año que los ASSISTANTS. Los PROFESSORS cobran 9864 dólares más que los ASSISTANTS. (Ojo los ASSISTANTS hacen de referencia)

Todas las variables

Vamos a estudiar el modelo que relaciona linealmente el salario anual con todas las variables.

```
m4 = lm(salary ~ year + ysdeg + sex + degree + rank, data = dat)
summary(m4)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg + sex + degree + rank, data = dat)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -4045.2 -1094.7  -361.5   813.2  9193.1 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16912.42    816.44  20.715 < 2e-16 ***
## year         476.31     94.91   5.018 8.65e-06 ***
## ysdeg        -124.57    77.49  -1.608   0.115    
## sexMale      -1166.37   925.57  -1.260   0.214    
## degreePhD    1388.61   1018.75   1.363   0.180    
## rankAssoc    5292.36   1145.40   4.621 3.22e-05 ***
## rankProf     11118.76  1351.77   8.225 1.62e-10 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2398 on 45 degrees of freedom
## Multiple R-squared:  0.855, Adjusted R-squared:  0.8357 
## F-statistic: 44.24 on 6 and 45 DF,  p-value: < 2.2e-16
```

Interpretación inicial

Observamos que el modelo tiene varias variables cuyo efecto no es significativo. Las variables *ysdeg*, *sexMale* y *degreePhD* son no significativas, que se interpreta como sigue: para explicar el salario es suficiente la antigüedad y la categoría del profesor, el resto de la información (grado, sexo y años con el diploma) no añaden información relevante.

Eliminación de variables del modelo

Vamos a quitar las NO.significativas con la función “STEP”

```
m5 = step(m4,trace=0)
summary(m5)

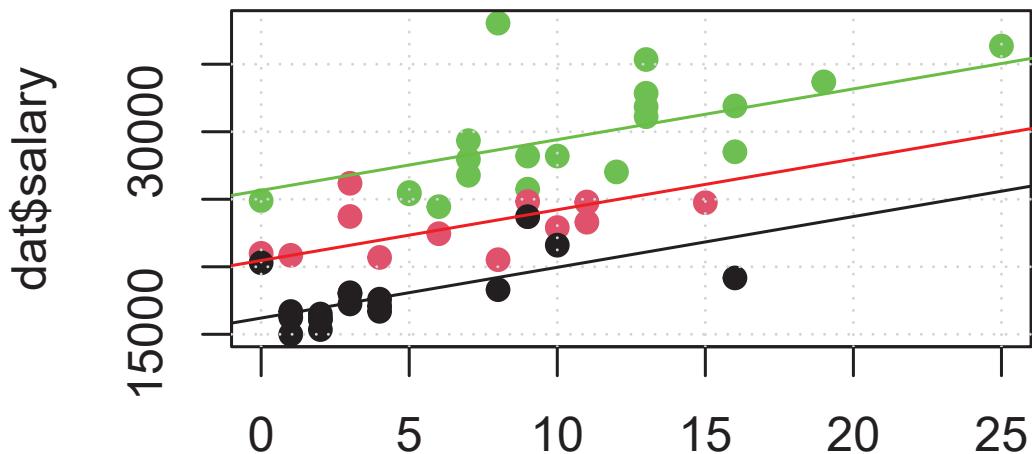
##
## Call:
## lm(formula = salary ~ year + rank, data = dat)
##
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -3462.0 -1302.8  -299.2   783.5 9381.6 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16203.27    638.68  25.370 < 2e-16 ***
## year        375.70     70.92   5.298 2.90e-06 ***
## rankAssoc   4262.28    882.89   4.828 1.45e-05 ***
## rankProf    9454.52    905.83  10.437 6.12e-14 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 2402 on 48 degrees of freedom
## Multiple R-squared:  0.8449, Adjusted R-squared:  0.8352 
## F-statistic: 87.15 on 3 and 48 DF,  p-value: < 2.2e-16
```

Interpretación

Según este modelo el salario está determinado por la antiguedad y la categoría (*rank*). Las demás variables no añaden información o no mejoran el modelo. En general, que una o más variables no entre en el modelo no significa necesariamente que no influyan en la variable respuesta. Puede ser que estén muy correlacionadas con algunos de los restantes regresores y su inclusión no es necesaria.

Gráfico del modelo

```
dat$rank = factor(dat$rank)
plot(dat$year,dat$salary,col = as.numeric(dat$rank),pch=19)
grid()
abline(c(16203.27,375.70))
abline(c(16203.27+4262.28,375.70),col="red")
abline(c(16203.27+9454.52,375.70),col="green")
```



Predicciones

En la siguiente tabla se muestra las predicciones y los errores (residuos) del modelo para cada uno de los datos de la muestra (solo se muestran los seis primeros).

```
dat$prediccion = fitted(m5)
dat$errores = residuals(m5)
head(dat)
```

	degree	rank	sex	year	ysdeg	salary	prediccion	errores
## 1	Masters	Prof	Male	25	35	36350	35050.18	1299.8175
## 2	Masters	Prof	Male	13	22	35350	30541.83	4808.1652
## 3	Masters	Prof	Male	10	23	28200	29414.75	-1214.7478
## 4	Masters	Prof	Female	7	27	26775	28287.66	-1512.6609
## 5	PhD	Prof	Male	19	30	33696	32796.01	899.9914
## 6	Masters	Prof	Male	16	21	28516	31668.92	-3152.9217

Interpretación

Según la ecuación del modelo $m5$, el valor previsto para el primer profesor, que tiene 25 años de antigüedad y que tiene la categoría de “Prof” es de 35058.18 \$, el dato observado es de 36350 \$, la diferencia $36350 - 35058.18 = 1299.81$ es el error que comete el modelo para este caso.



Diagnosis del modelo

Para comprobar las hipótesis del modelo:

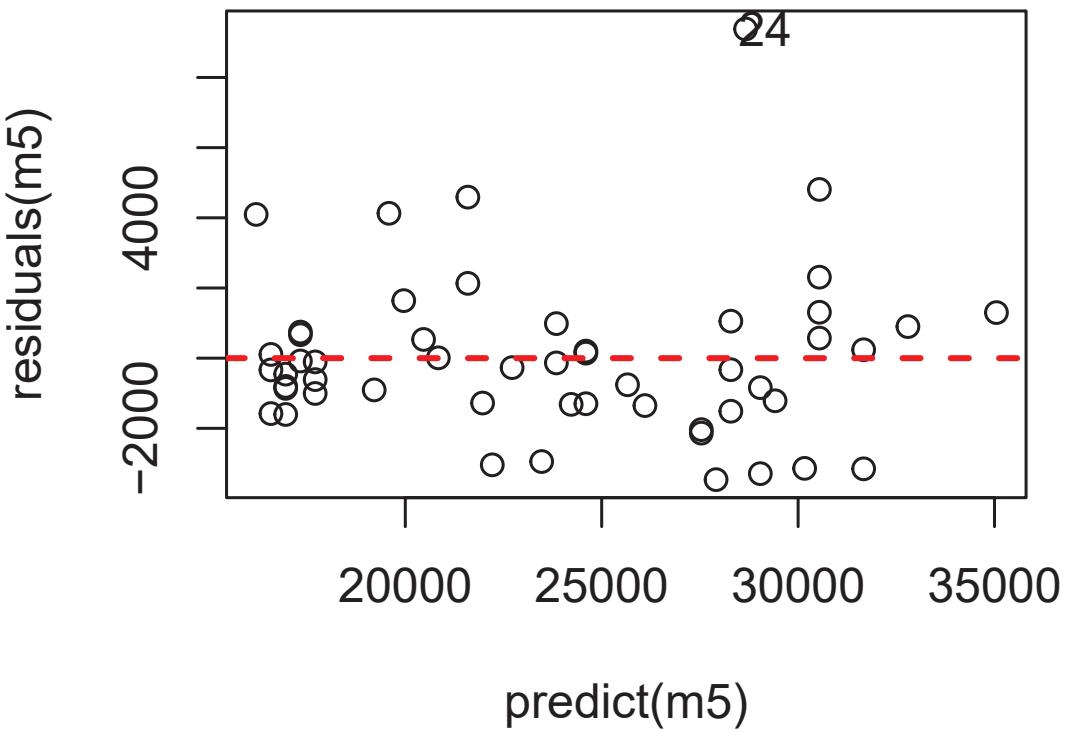
- linealidad,
- homocedasticidad y
- normalidad

Se utiliza la instrucción `plot(m5)`



Homocedasticidad, linealidad y valores atípicos

```
par(mar=c(6.1, 4.1, 0.5, 2.1))
plot(predict(m5),residuals(m5))
abline(h=0,col="red",lw=2,lt=2)
text(predict(m5)[24]+500,residuals(m5)[24],labels = "24")
```



Homocedasticidad, linealidad y atípicos (interpretación)

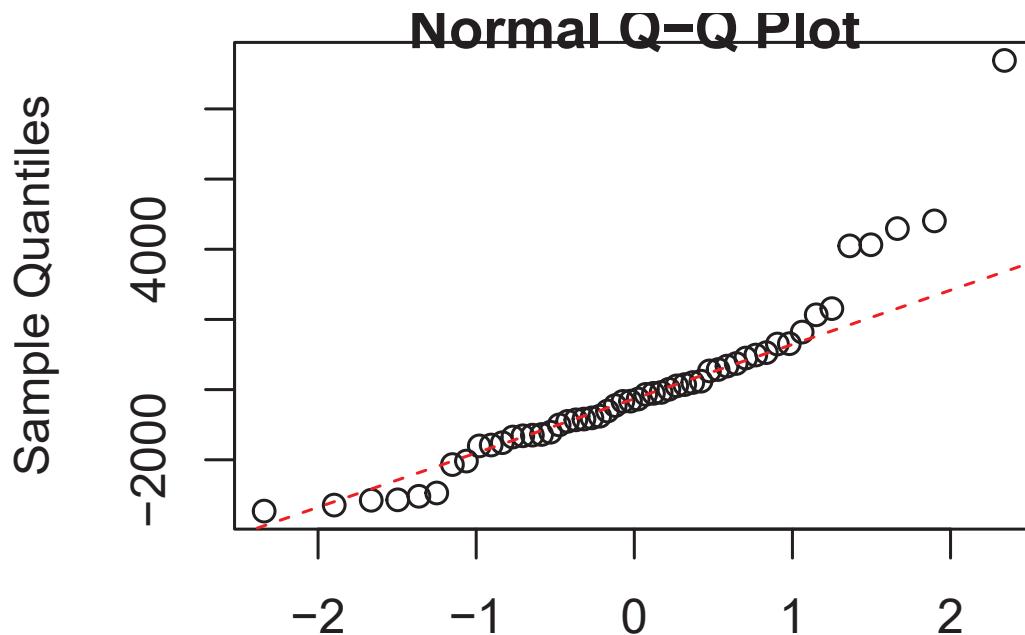
Se utiliza el gráfico de residuos frente a valores previstos. En la figura se observa que los residuos presentan una variabilidad uniforme alrededor de la linea central. Esto es señal de que los datos cumplen la hipótesis de linealidad y homocedasticidad.

Se aprecia una observación atípica, que corresponde a la observación número 24. Más adelante analizaremos este problema.

Normalidad

Se comprueba con el gráfico qq plot. En la figura se muestran los quantiles teóricos de la distribución normal que corresponderían a los residuos si estos tuvieran distribución normal. Se aprecia que a excepción de la observación 24, los restantes valores se comportan aceptablemente.

```
par(mar=c(6.1, 4.1, 0.5, 2.1))
qqnorm(residuals(m5))
qqline(residuals(m5), col="red", lty=2)
```



Nuevo modelo sin la observación 24

Cuando en el estudio aparecen algunas observaciones con residuos muy altos o que tienen valores de los regresores muy distintos a la mayoría de las observaciones, es importante ver su influencia en el modelo estimado. Una manera de estudiar su efecto es re-estimar el modelo sin estas observaciones y analizar como cambia.

```
m4_2 = lm(salary ~ year + ysdeg + sex + degree + rank,
           data = dat[c(-24,)]) # elimino la observación (fila) 24
summary(m4_2)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg + sex + degree + rank, data = dat[c(-24),
## ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3976.9   -902.1   -38.3    758.1   5464.1 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16412.51    660.84  24.836 < 2e-16 ***
## year        476.55     75.98   6.272 1.34e-07 ***
## ysdeg       -134.47    62.06  -2.167  0.0357 *  
## sexMale     -296.99    760.15  -0.391  0.6979    
## degreePhD   1742.34   818.46   2.129  0.0389 *  
## rankAssoc   5005.30   918.64   5.449 2.16e-06 ***
## rankProf    10533.63  1088.16   9.680 1.80e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1920 on 44 degrees of freedom
## Multiple R-squared:  0.8973, Adjusted R-squared:  0.8832 
## F-statistic: 64.04 on 6 and 44 DF,  p-value: < 2.2e-16
```

Nuevo modelo sin 24

Se aprecian cambios importantes respecto al modelo *m4*:

1. La desviación típica residual ha disminuido pasando a ser 1920 \$.

- ② R^2 ha aumentado y ahora es casi el 90 %
- ③ En este modelo la variable *degree* (el grado del profesor: Master o PhD) tiene efecto significativo, los profesores con el grado de doctor cobran 1742.34 \$ más al año en media que los que solo tienen el Máster.
- ④ La variable *ysdeg* también es significativa. Pero el signo negativo tiene menos lógica. Se interpreta como que a igualdad de categoría, de antigüedad (*year*) y grado, cuanto mayor es *ysdeg* menor es el salario (-134 \$/año). Esto es debido a la multicolinealidad entre los regresores.

Eliminando var. no significativas

Eliminando las variables no significativas, el modelo final podría ser:

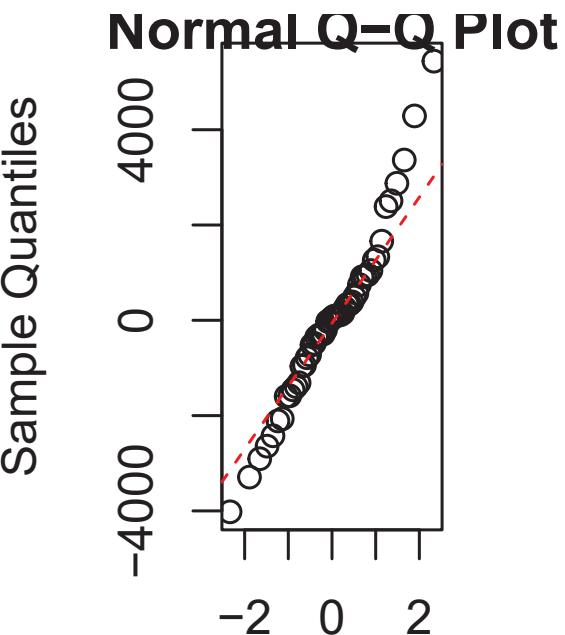
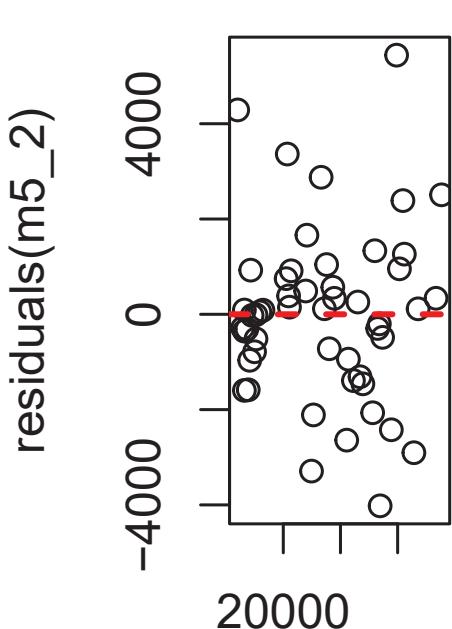
```
m5_2 = step(m4_2,trace = FALSE)
summary(m5_2)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg + degree + rank, data = dat[c(-24),
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4019.7  -953.1    51.4   831.2  5439.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16258.57    525.52  30.938 < 2e-16 ***
## year        462.10     65.74   7.029 9.24e-09 ***
## ysdeg       -124.15    55.63  -2.232  0.0306 *
## degreePhD   1669.38    789.33   2.115  0.0400 *
## rankAssoc   4860.47    832.58   5.838 5.43e-07 ***
## rankProf    10376.30   1001.33  10.363 1.68e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1902 on 45 degrees of freedom
## Multiple R-squared:  0.8969, Adjusted R-squared:  0.8854
## F-statistic: 78.3 on 5 and 45 DF,  p-value: < 2.2e-16
```

Diagnosis

```
par(mar=c(6.1, 4.1, 0.5, 2.1))
par(mfrow=c(1,2))          # figura doble 1 por 2
plot(predict(m5_2),residuals(m5_2))
abline(h=0,col="red",lw=2,lty=2)

qqnorm(residuals(m5_2))
qqline(residuals(m5_2),col="red",lty=2)
```



Conclusiones

- El modelo de regresión nos permite construir una ecuación que explica la influencia de un conjunto de variables (regresores) en otra variable (variable dependiente o variable respuesta)
- El proceso de estimación implica muchas decisiones complejas y en algún caso, subjetiva: transformar/eliminar variables y/o observaciones del modelo. El procedimiento puede dar lugar a modelos distintos que aunque en lo fundamental pueden coincidir pueden diferir en algunos detalles que serán o no importantes en función del objetivo del análisis.
- En la aplicación de estas técnicas es bueno tener presente que los resultados de la estimación dependen de la muestra que estamos analizando y que las conclusiones alcanzadas tienen que ser coherentes y reproducibles con otra muestra diferente.

Conclusiones (cont)

En el caso particular que estudiamos, el de los salarios de los profesores de universidad podemos concluir lo siguiente:

- El modelo predice el salario con $R^2 = 90\%$ (aprox).
- Las variables que más influyen son la antigüedad y la categoría del profesor. También influyen el grado y los años que lleva en posesión del título, aunque el coeficiente de esta última variable es difícil de interpretar.
- No se han encontrado diferencias significativas en el salario de hombres y mujeres.
- La ecuación del modelo permite predecir el salario anual con un error medio de 1726 \$
- En el estudio se ha eliminado una observación (número 24) que tenía un comportamiento muy diferente al resto. Este hecho implica que puede haber profesores cuyos salarios no se ajustan a las predicciones de este modelo. En nuestro estudio hay 1 caso en una muestra de 52 (el 2% de observaciones atípicas)

Instrucciones más importantes de Regresión con R

```
cars1 = read.table("data/cars.txt", header=TRUE)
head(cars1)
```

```
##   mpg engine horse weight accel origin cylinders
## 1 14     340   160   3609    8.0      1         8
## 2 14     440   215   4312    8.5      1         8
## 3 15     390   190   3850    8.5      1         8
## 4 14     454   220   4354    9.0      1         8
## 5 15     400   150   3761    9.5      1         8
## 6 16     400   230   4278    9.5      1         8
```

En las instrucciones siguientes se utiliza los datos del consumo mpg de 391 coches de distintas nacionalidades origin (1=USA,2=EUR,3=JAP). Además se incluyen las variables engine (cilindrada), horse (potencia en CV), weight (peso en libras), accel (tiempo en segundos para pasar de 0 a 60km/h) y cylinders (número de cilindros).

Regresión Simple

```
cars1 = read.table("data/cars.txt",header=TRUE)
# lee los datos cars.txt en el directorio \data

m0 = lm (mpg ~ horse, data = cars1)
# estima el modelo de regresión: mpg = b0 + b1 horse + u

summary(m0)
# proporciona los resultados del modelo m0

plot(cars1$horse,cars1$mpg)
# gráfico de dispersión entre horse (x) y mpg (y)

abline (m0,col="red",lw=2)
# dibuja la recta de reg. estimada en m0 (color rojo y grosor=2)
```

Regresión Múltiple

```
m1 = lm (mpg ~ horse + weight +
          accel, data = cars1)
# estima el modelo de regresión múltiple

m1a = lm (mpg ~ horse + I(horse^2) + weight +
           accel, data = cars1)
# incluye el término horse al cuadrado

m1b = lm (mpg ~ horse + weight + I(horse*weight) +
           accel, data = cars1)
# incluye el término horse*weight

m1c = lm (log(mpg) ~ horse + weight +
           accel, data = cars1)
# utiliza el log de mpg como variable respuesta
```

Regresión Múltiple con variables cualitativas

```
cars1$origin = factor( cars1$origin,
  labels = c("USA","EUR","JAP"))
# Convierte "origin" a tipo "factor" y se asignan etiquetas

m2 = lm (mpg ~ horse + weight + accel + origin,
          data = cars1)
# modelo con variable cualitativa (utiliza la primera como referencia)

cars1$origin = relevel(cars1$origin,
  ref = "EUR")
# Cambia el nivel de referencia (por defecto el primero)

m2a = lm (mpg ~ horse + weight + accel + origin,
          data = cars1)
# modelo con variable cualitativa con EUR como referencia

m2b = lm (mpg ~ weight + accel + origin + horse*origin,
          data = cars1)
# modelo con parámetros asociados a horse distintos para cada origen

m3 = lm (mpg ~ ., data = cars1)
# utiliza todas las variables en cars1 como regresores

anova(m3)
# análisis de la varianza del modelo m3
```

Diagnosis del modelo de regresión

```
plot(m3)
# diagnosis del modelo m3

resi = residuals(m3)
# residuos del modelo m3 para las observaciones en cars1

pred = fitted(m3)
# valores predichos de m3 (ajustados) para las observaciones en cars1

plot(pred,resi)
# Diagnosis: comprueba linealidad y homocedasticidad

qqnorm(resi)
# Diagnosis: comprueba normalidad

qqline(resi)
# añade recta al qqplot para comprobar normalidad
```

Predicción

```
xnueva = data.frame(engine=180, horse =100, weight=3000,  
                     accel =10, origin = "JAP",  
                     cylinders=4)  
# coche nuevo para hacer predicción del consumo  
  
predict(m3,xnueva,  
        interval = "confidence")  
# predicción e intervalo para la media  
  
predict(m3,xnueva,  
        interval = "prediction")  
# predicción e intervalo para una nueva observación
```

Otras instrucciones para Regresión

```
m4 = step(m3)  
# a partir de m3 selecciona el modelo utilizando STEPWISE  
  
coefficients(m4)  
# coeficientes del modelo  
  
confint(m4, level=0.95)  
# intervalo de confianza para los coef.  
  
vcov(m4)  
# matriz de varianza de los parámetros estimados  
  
out = influence(m4)  
# diagnosis sobre datos atípicos
```

Modelo de Regresión

Datos

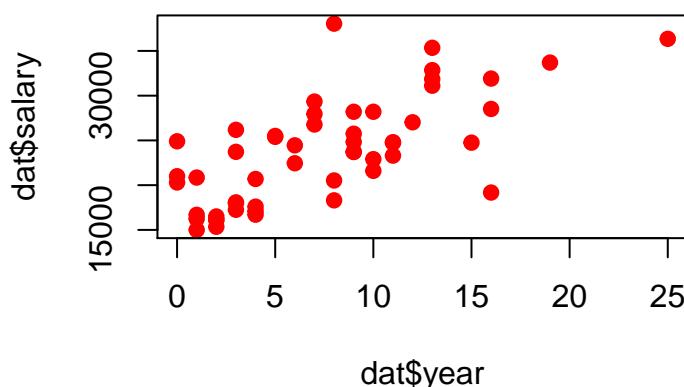
Para leer datos utilizo la instrucción `read.table` y para ver las primeras files (observaciones) de los datos leidos, utilizo la instrucción `head()`

```
dat = read.table("salary.txt", header=T) # (1)  
head(dat) # (2)
```

```
##      degree rank    sex year ysdeg salary  
## 1 Masters Prof   Male  25     35 36350  
## 2 Masters Prof   Male  13     22 35350  
## 3 Masters Prof   Male  10     23 28200  
## 4 Masters Prof Female  7     27 26775  
## 5     PhD Prof   Male  19     30 33696  
## 6 Masters Prof   Male  16     21 28516
```

Relación entre variables: correlación y regresión lineal

```
plot(dat$year, dat$salary, col="red", pch=19)
```



La medida de asociación más habitual es la correlación

```
cor(dat$year,dat$salary)
```

```
## [1] 0.700669
```

El modelo de regresión profundiza en la relación lineal

```
m = lm(salary ~ year, data = dat)
summary(m)
```

```
##
## Call:
## lm(formula = salary ~ year, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11035.9  -3172.4   -561.7   3185.8  13856.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18166.1     1003.7  18.100 < 2e-16 ***
## year         752.8      108.4   6.944 7.34e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4264 on 50 degrees of freedom
## Multiple R-squared:  0.4909, Adjusted R-squared:  0.4808
## F-statistic: 48.22 on 1 and 50 DF,  p-value: 7.341e-09
```

La ecuación del modelo es:

$$\text{salary}_i = 18166.1 + 752.8 \times \text{year}_i + e_i$$

Interpretación del modelo:

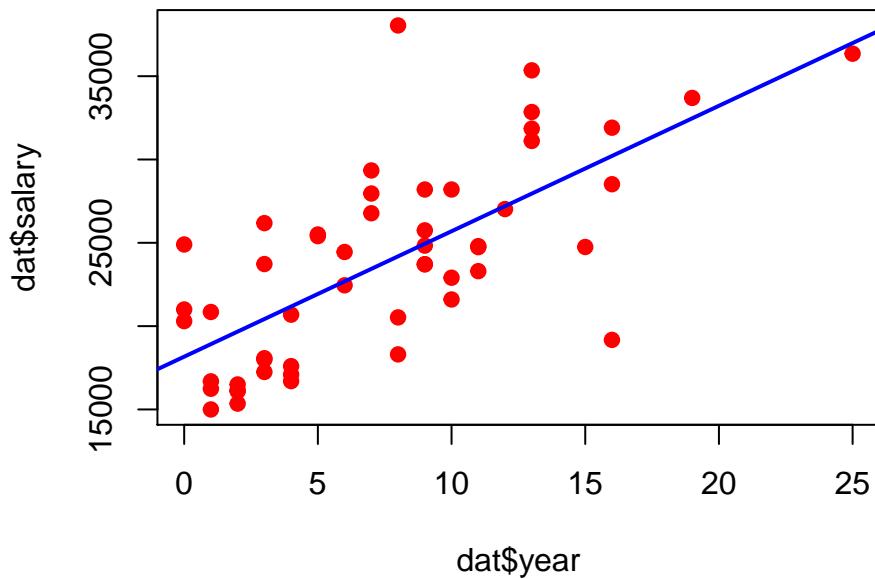
1. $\hat{\beta}_1 = 752.8$: Mide la relación entre la antigüedad (year) y el salario. Se ha estimado que el salario aumenta 752.8 \$ por cada año de antiguedad. El parámetro β_1 se ha estimado con un error de 108.4 \$/año, y los resultados indican que β_1 es significativamente distinto de cero (p-value=7.34e-09).
2. Conocida la antigüedad (year) la ecuación anterior predice el salario anual de un profesor con un error medio de 4264 dolares.

$$\hat{s}_R = 4264 \$$$

3. $R^2 = 0.4909$: La antigüedad explica el 49.09% de la variabilidad del salario. Se observa que personas con los mismos años de antigüedad tienen salarios diferentes. La variable "year" (antigüedad), solo explica el 49% del salario. Hay otras variables que influyen y no están en la ecuación que explican el $100 - 49.09 = 50.91\%$.

4. $\beta_0 = 18166.1$ \$ es el salario anual que estima el modelo para un profesor sin antigüedad (year=0). En este ejemplo se puede interpretar, pero en la mayoría de los casos es un parámetro que no tiene fácil interpretación. En general es un parámetro con menor interés.

```
plot(dat$year,dat$salary, col="red", pch=19)
abline(m,col = "blue",lwd = 2)
```



Vamos a poner la otra variable (años desde que consiguió el grado ysdeg)

```
m1 = lm(salary ~ year + ysdeg, data = dat)
summary(m1)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -10321.2  -2347.2  -332.7  2298.8 12240.9 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16555.7    1052.4  15.732 < 2e-16 ***
## year        489.3     129.6   3.777 0.000431 ***
## ysdeg       222.2      69.8   3.184 0.002525 **  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## Residual standard error: 3921 on 49 degrees of freedom
## Multiple R-squared:  0.5782, Adjusted R-squared:  0.561
## F-statistic: 33.58 on 2 and 49 DF,  p-value: 6.532e-10

```

$$salary_i = 16555.7 + 489.3year_i + 222.2ysdeg_i + e_i$$

Interpretación del modelo:

1. La nueva variable aumenta la capacidad explicativa del modelo. Con las dos variables (year y ysdeg, explicamos el 57.8%).
2. El salario anual se predice con un error medio de 3921 dólares con el nuevo modelo (se ha reducido respecto al modelo anterior).
3. Al introducir la nueva variable la estimación del efecto de la antigüedad (year) ha variado, ahora el parámetro $\hat{\beta}_1$ vale 489.3 \$/año. El efecto de *ysdeg* (los años desde que obtuvo el título de Master o Doctorado) es de 222.2 \$ por año. Los dos coeficientes son significativamente distintos de cero. SE dice que las variables *year* y *ysdeg* tienen un efecto significativo en el salario.

Desgraciadamente no podemos hacer una representación gráfica fácil del modelo.

Variables cualitativas en el modelo

Introducimos la variable *sexo* en el modelo que es una variable cualitativa que toma el valor 0 para mujeres y 1 para hombres.

```

m2 = lm(salary ~ year + ysdeg + sex, data = dat)
summary(m2)

```

```

##
## Call:
## lm(formula = salary ~ year + ysdeg + sex, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10298.6  -2175.5  -383.5  2106.0 12736.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16120.84    1325.78 12.160 2.89e-16 ***
## year        456.17     143.90   3.170  0.00265 **
## ysdeg       230.76      72.02   3.204  0.00241 **
## sexMale     746.41    1366.81   0.546  0.58753
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3949 on 48 degrees of freedom
## Multiple R-squared:  0.5808, Adjusted R-squared:  0.5546
## F-statistic: 22.17 on 3 and 48 DF,  p-value: 3.762e-09

```

Interpretación del modelo:

- Según esto a igualdad de antiguedad y edad, los hombres cobran 746 dólares más al año que las mujeres. Pero el efecto no es significativo. Miramos que esta variable mejora muy poquito R², ahora es 58.08%. (Ojo, al añadir una variable al modelo **siempre** aumenta R², por tanto el criterio R² para comparar modelos no es útil. Para corregir ese problema se utiliza Adjusted-R². Se aprecia que el valor R² ajustado en este caso es peor el obtenido con el modelo m2).
- El resto de los parámetros del modelo no han cambiado mucho. La interpretación es similar al modelo anterior.

Variable cualitativa con más de dos grupos

Vamos a introducir la categoría del profesor (*rank*) en el modelo. El resultado es el siguiente.

```
m3 = lm(salary ~ year + ysdeg + rank, data = dat)
summary(m3)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg + rank, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3413.7 -1218.5  -182.7   742.0  9483.3 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16317.46    667.86  24.433 < 2e-16 ***
## year        400.46     81.50   4.914 1.13e-05 ***
## ysdeg       -34.32     54.54  -0.629   0.532    
## rankAssoc   4619.12   1054.02   4.382 6.55e-05 ***
## rankProf    9864.30   1120.27   8.805 1.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2417 on 47 degrees of freedom
## Multiple R-squared:  0.8462, Adjusted R-squared:  0.8331 
## F-statistic: 64.64 on 4 and 47 DF,  p-value: < 2.2e-16
```

El modelo ha mejorado mucho!!!

Interpretación:

- El modelo explica el 84.6% de la variabilidad salarial.
- La variable ysdeg no es significativa. Esto es frecuente en los modelos de regresión, al añadir nuevas variables al modelo, otras variables presentes en él dejan de ser significativas. Esto se debe a que las variables explicativas del modelo están relacionadas entre ellas. A este fenómeno se le denomina multicolinealidad (relaciones lineales entre los regresores).
- El error del modelo se ha reducido a 2417 dólares
- El salario de un profesor ASSISTANT tiene como base 16317 dólares anuales, más 400 dólares por año contratado. Los ASSOCIATES cobran 4619 más al año que los ASSISTANTS. Los PROFESSORS cobran 9864 dólares más que los ASSISTANTS. (Ojo los ASSISTANTS hacen de referencia)

Todas las variables

Vamos a estudiar el modelo que relaciona linealmente el salario anual con todas las variables.

```
m4 = lm(salary ~ year + ysdeg + sex + degree + rank, data = dat)
summary(m4)
```

```
##
## Call:
## lm(formula = salary ~ year + ysdeg + sex + degree + rank, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -4045.2 -1094.7  -361.5   813.2  9193.1 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16912.42    816.44  20.715 < 2e-16 ***
## year        476.31     94.91   5.018 8.65e-06 ***
## ysdeg      -124.57    77.49  -1.608   0.115    
## sexMale    -1166.37   925.57  -1.260   0.214    
## degreePhD  1388.61   1018.75   1.363   0.180    
## rankAssoc   5292.36   1145.40   4.621 3.22e-05 ***
## rankProf   11118.76   1351.77   8.225 1.62e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2398 on 45 degrees of freedom
## Multiple R-squared:  0.855, Adjusted R-squared:  0.8357 
## F-statistic: 44.24 on 6 and 45 DF,  p-value: < 2.2e-16
```

Observamos que el modelo tiene varias variables cuyo efecto no es significativo. Las variables *ysdeg*, *sexMale* y *degreePhD* son no significativas, que se interpreta como sigue: para explicar el salario es suficiente la antigüedad y la categoría del profesor, el resto de la información (grado, sexo y años con el diploma) no añaden información relevante.

Vamos a quitar las NO.significativas con la función “STEPWISE”

```
m5 = step(m4,trace=0)
summary(m5)
```

```
##
## Call:
## lm(formula = salary ~ year + rank, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3462.0 -1302.8  -299.2   783.5  9381.6 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16203.27    638.68  25.370 < 2e-16 ***
## year         375.70     70.92   5.298 2.90e-06 ***
```

```

## rankAssoc    4262.28      882.89     4.828 1.45e-05 ***
## rankProf     9454.52      905.83    10.437 6.12e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2402 on 48 degrees of freedom
## Multiple R-squared:  0.8449, Adjusted R-squared:  0.8352
## F-statistic: 87.15 on 3 and 48 DF,  p-value: < 2.2e-16

```

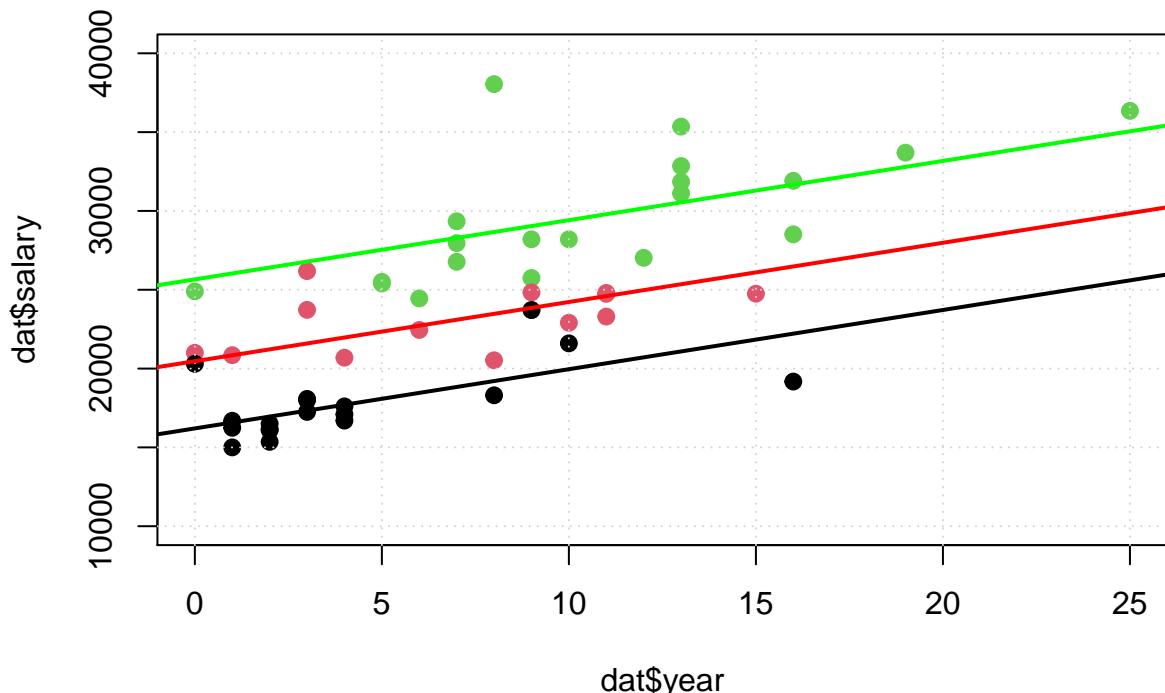
Según este modelo el salario está determinado por la antigüedad y la categoría (*rank*). Las demás variables no añaden información o no mejoran el modelo. En general, que una o más variables no entre en el modelo no significa necesariamente que no influyan en la variable respuesta. Puede ser que estén muy correlacionadas con algunos de los restantes regresores y su inclusión no es necesaria.

Podemos representar gráficamente el modelo, porque sólo hay una variable explicativa continua y la otra es cualitativa. Abajo se muestra la figura con las tres rectas, una para Professors (verde), otra para Associates (roja) y otra para Assistants (negra).

```

plot(dat$year,dat$salary,
      col= factor(dat$rank),
      pch=19,cex=1.1,ylim=c(10000,40000))
grid()
abline(c(16203.27,375.70),lwd=2)
abline(c(16203.27+4262.28,375.70),col="red",lwd=2)
abline(c(16203.27+9454.52,375.70),col="green",lwd=2)

```



Predicciones

En la siguiente tabla se muestra las predicciones y los errores (residuos) del modelo para cada uno de los datos de la muestra (solo se muestran los seis primeros).

```
dat$prediccion = fitted(m5)
dat$errores = residuals(m5)
head(dat)
```

```
##   degree rank   sex year ysdeg salary prediccion     errores
## 1 Masters Prof Male  25    35  36350  35050.18  1299.8175
## 2 Masters Prof Male  13    22  35350  30541.83  4808.1652
## 3 Masters Prof Male  10    23  28200  29414.75 -1214.7478
## 4 Masters Prof Female 7    27  26775  28287.66 -1512.6609
## 5     PhD Prof Male  19    30  33696  32796.01   899.9914
## 6 Masters Prof Male  16    21  28516  31668.92 -3152.9217
```

Según la ecuación del modelo $m5$, el valor previsto para el primer profesor, que tiene 25 años de antigüedad y que tiene la categoría de “Prof” es de 35058.18 \$, el dato observado es de 36350 \$, la diferencia 36350-35058.18 = 1299.81 es el error que comete el modelo para este caso.

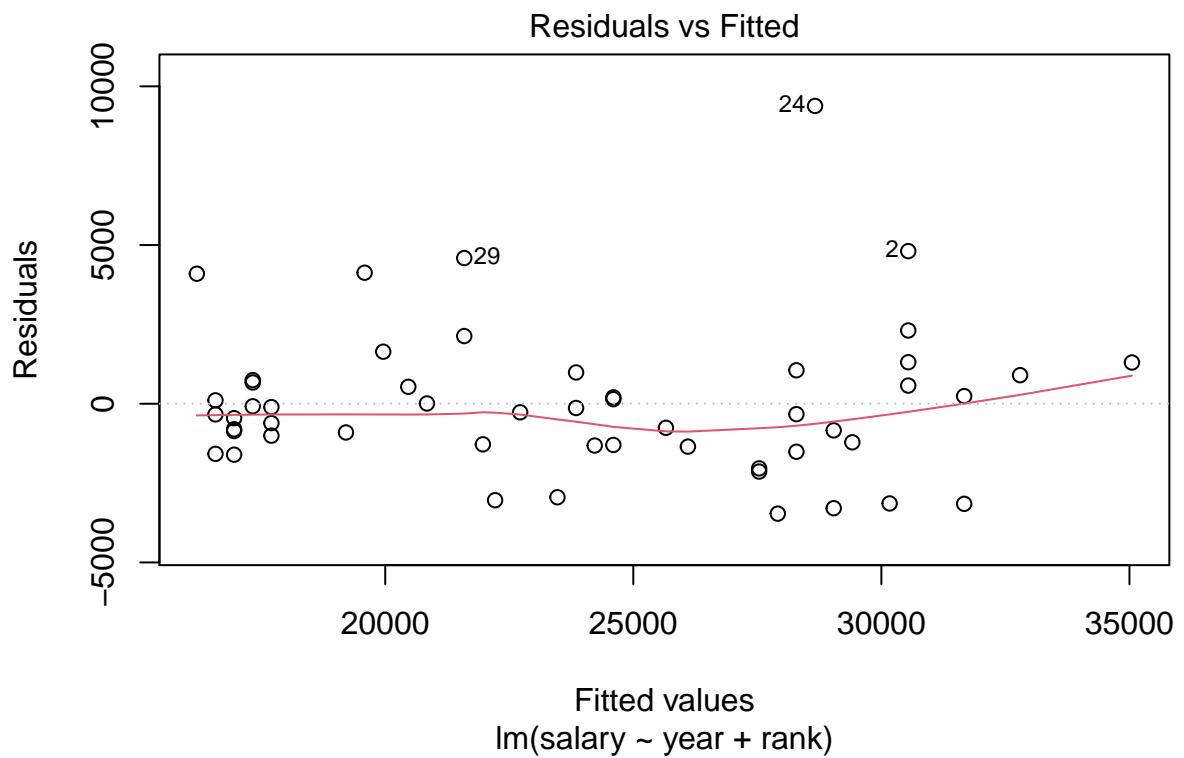
Diagnosis del modelo

Para comprobar las hipótesis del modelo: linealidad, homocedasticidad y la normalidad se utilizan dos gráficos.

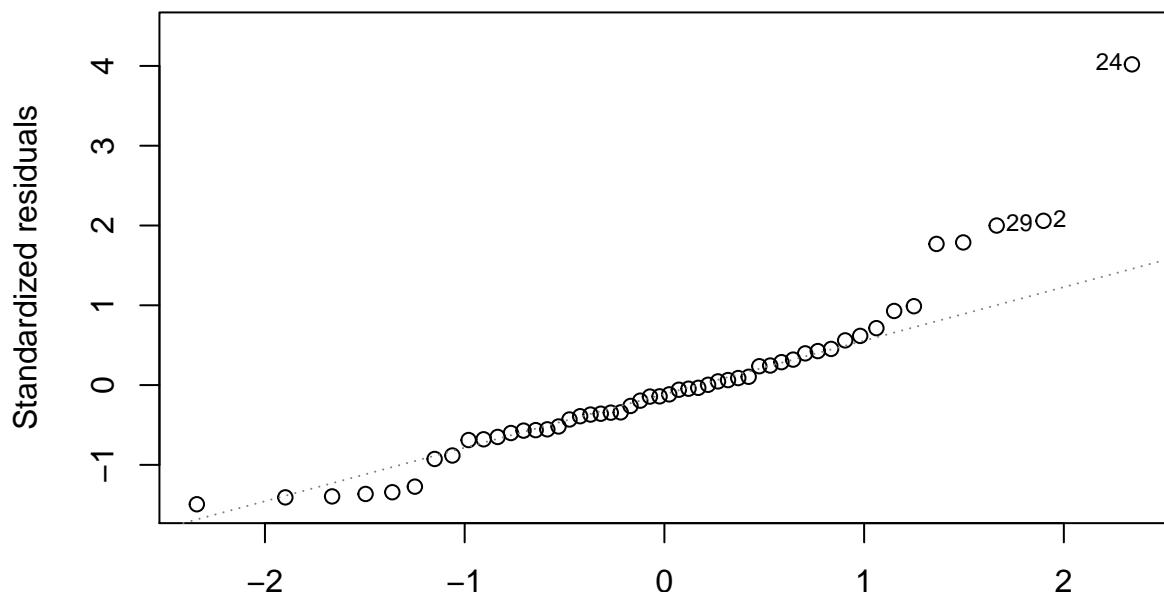
Homocedasticidad, linealidad y valores atípicos

Se utiliza el gráfico de residuos frente a valores previstos.

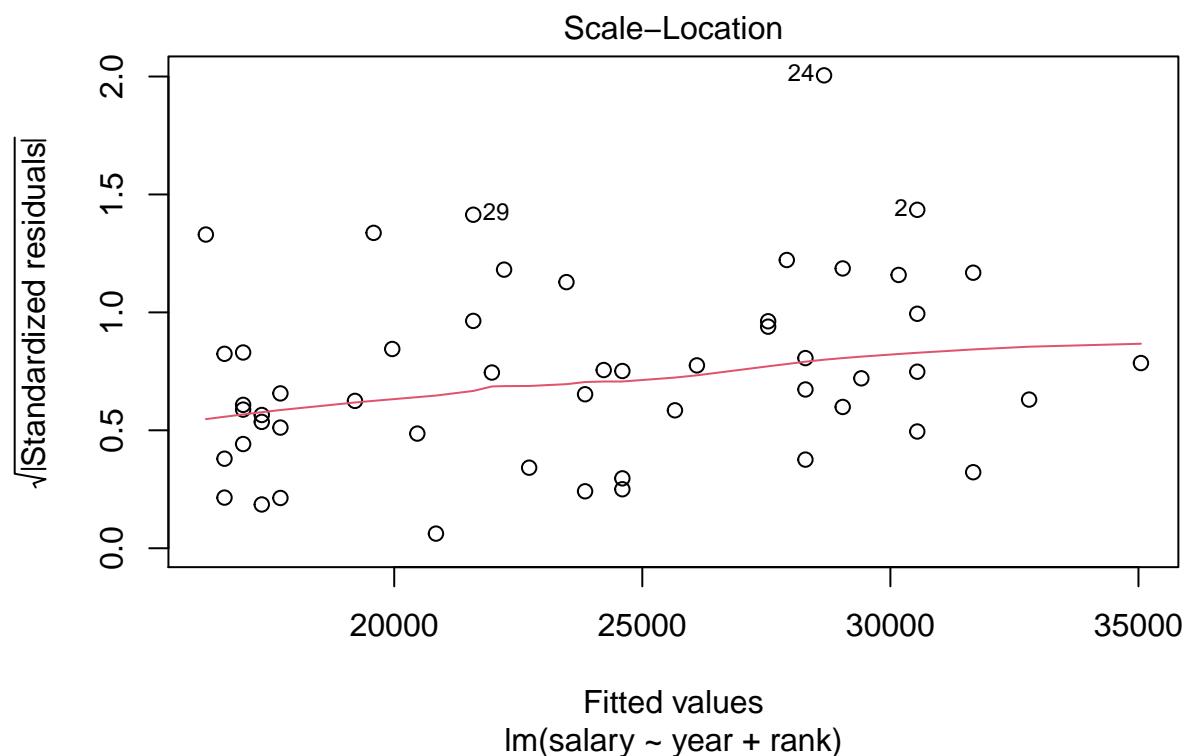
```
plot(m5)
```

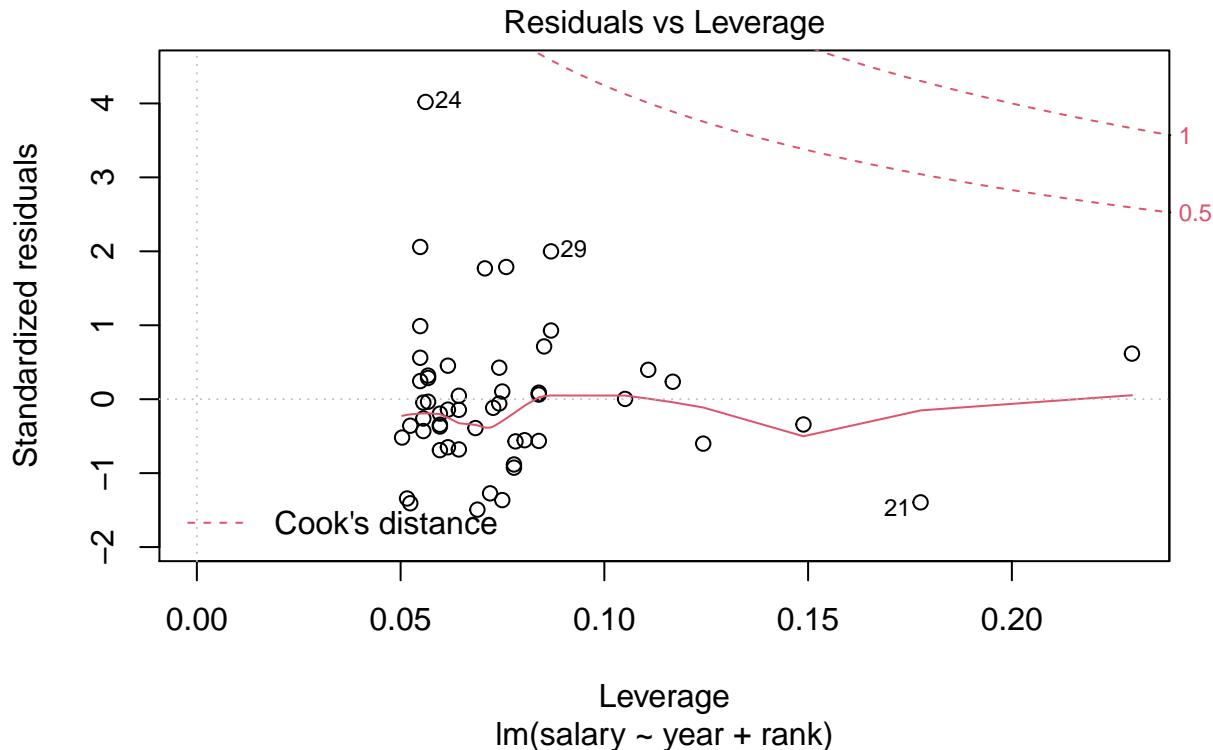


Normal Q-Q



Theoretical Quantiles
lm(salary ~ year + rank)





En la figura se observa que los residuos presentan una variabilidad uniforme alrededor de la linea central. Esto es señal de que los datos cumplen la hipótesis de linealidad y homocedasticidad.

Se aprecia una observación atípica, que corresponde a la observación número 24. Más adelante analizaremos este problema.

Normalidad

Se comprueba con el gráfico qq plot. En la figura se muestran los quantiles teóricos de la distribución normal que corresponderían a los residuos si estos tuvieran distribución normal. Se aprecia que a excepción de la observación 24, los restantes valores se comportan aceptablemente.

Nuevo modelo sin la observación 24

Cuando en el estudio aparecen algunas observaciones con residuos muy altos o que tienen valores de los regresores muy distintos a la mayoría de las observaciones, es importante ver su influencia en el modelo estimado.

Una manera de estudiar su efecto es re-estimar el modelo sin estas observaciones y analizar como cambia.

En la siguiente instrucción se estima el modelo sin la observación 24.

```
m4_2 = lm(salary ~ year + ysdeg + sex + degree + rank,
           data = dat[c(-24,)])
# elimino la observación (fila) 24
summary(m4_2)
```

```

## 
## Call:
## lm(formula = salary ~ year + ysdeg + sex + degree + rank, data = dat[c(-24),
##      ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -3976.9 -902.1 -38.3  758.1 5464.1 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16412.51   660.84  24.836 < 2e-16 ***
## year        476.55    75.98   6.272 1.34e-07 ***
## ysdeg       -134.47   62.06  -2.167  0.0357 *  
## sexMale     -296.99   760.15  -0.391  0.6979    
## degreePhD   1742.34   818.46   2.129  0.0389 *  
## rankAssoc   5005.30   918.64   5.449 2.16e-06 ***
## rankProf    10533.63  1088.16   9.680 1.80e-12 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 1920 on 44 degrees of freedom
## Multiple R-squared:  0.8973, Adjusted R-squared:  0.8832 
## F-statistic: 64.04 on 6 and 44 DF,  p-value: < 2.2e-16

```

Se aprecian cambios importantes respecto al modelo *m4*:

1.La desviación típica residual ha disminuido pasando a ser 1920 \$.

2. R^2 ha aumentado y ahora es casi el 90 %

3. En este modelo la variable *degree* (el grado del profesor: Master o PhD) tiene efecto significativo, los profesores con el grado de doctor cobran 1742.34 \$ más al año en media que los que solo tienen el Máster.
4. La variable *ysdeg* también es significativa. Pero el signo negativo tiene menos lógica. Se interpreta como que a igualdad de categoría, de antigüedad (*year*) y grado, cuanto mayor es *ysdeg* menor es el salario (-134 \$/año). Esto es debido a la multicolinealidad entre los regresores.

Eliminando las variables no significativas, el modelo final podría ser:

```

m5_2 = step(m4_2,trace = FALSE)
summary(m5_2)

```

```

## 
## Call:
## lm(formula = salary ~ year + ysdeg + degree + rank, data = dat[c(-24),
##      ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -4019.7 -953.1  51.4  831.2 5439.2 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16412.51   660.84  24.836 < 2e-16 ***
## year        476.55    75.98   6.272 1.34e-07 ***
## ysdeg       -134.47   62.06  -2.167  0.0357 *  
## sexMale     -296.99   760.15  -0.391  0.6979    
## degreePhD   1742.34   818.46   2.129  0.0389 *  
## rankAssoc   5005.30   918.64   5.449 2.16e-06 ***
## rankProf    10533.63  1088.16   9.680 1.80e-12 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 1920 on 44 degrees of freedom
## Multiple R-squared:  0.8973, Adjusted R-squared:  0.8832 
## F-statistic: 64.04 on 6 and 44 DF,  p-value: < 2.2e-16

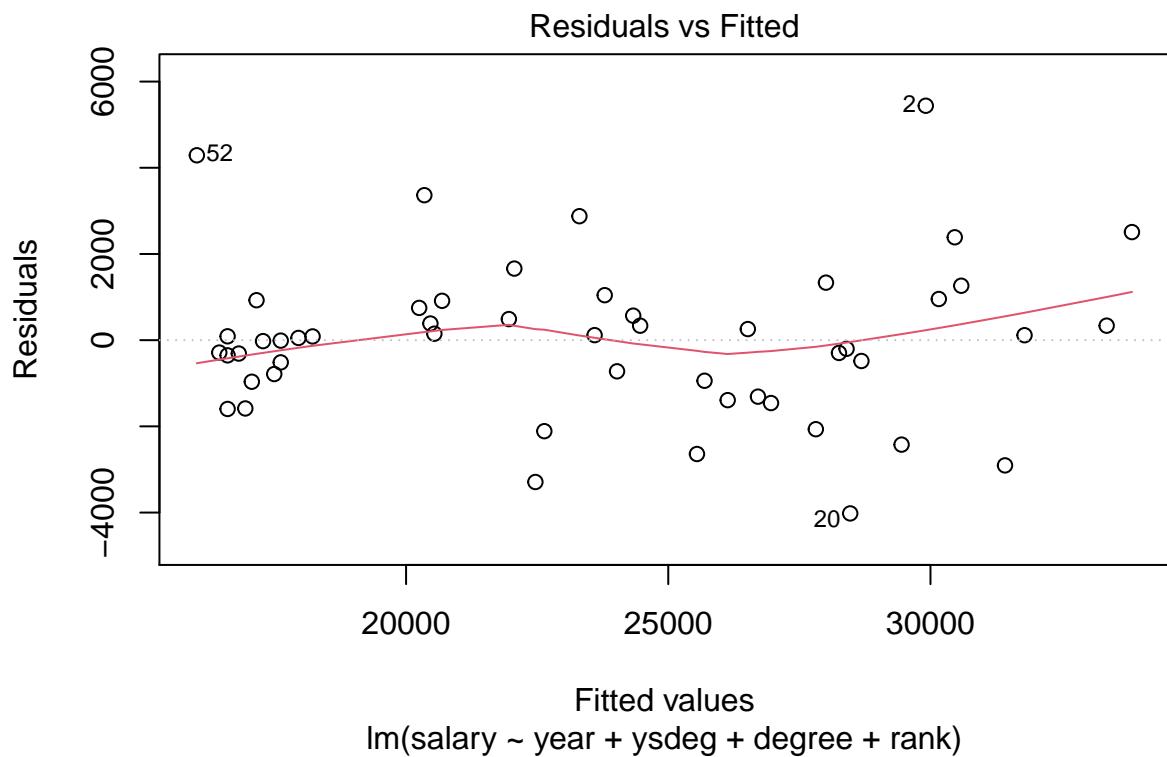
```

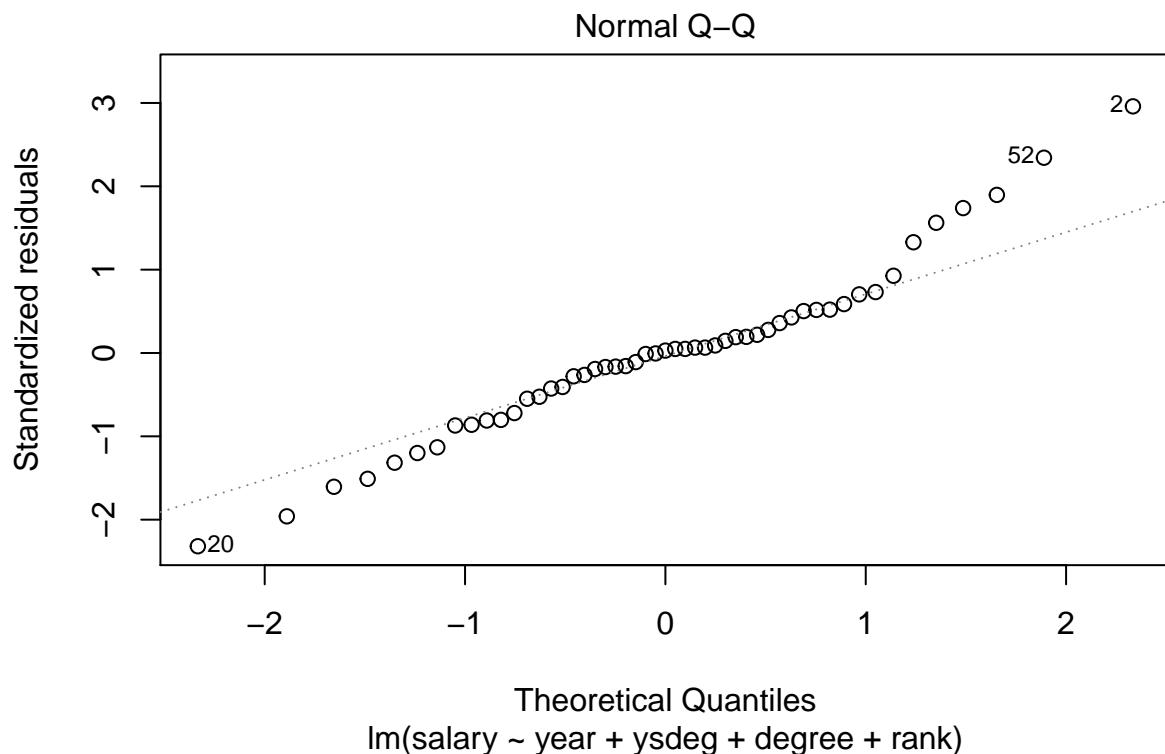
```

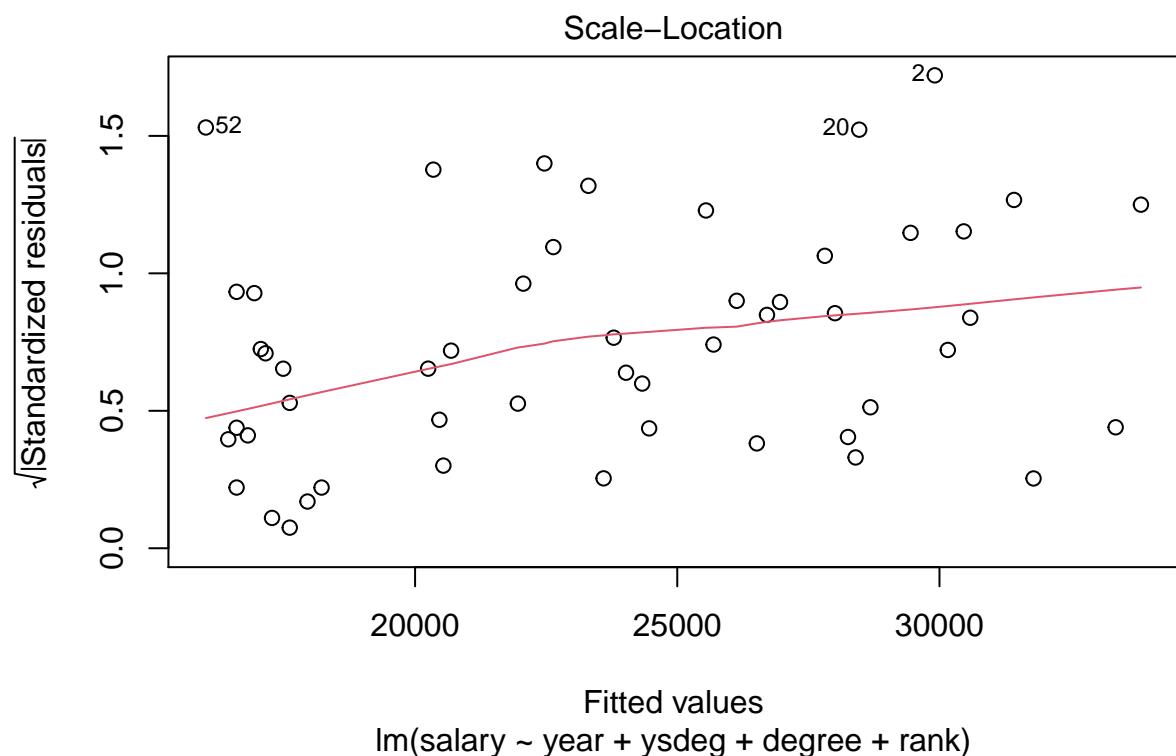
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16258.57     525.52 30.938 < 2e-16 ***
## year        462.10      65.74  7.029 9.24e-09 ***
## ysdeg       -124.15     55.63 -2.232  0.0306 *
## degreePhD   1669.38    789.33  2.115  0.0400 *
## rankAssoc   4860.47    832.58  5.838 5.43e-07 ***
## rankProf    10376.30   1001.33 10.363 1.68e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1902 on 45 degrees of freedom
## Multiple R-squared:  0.8969, Adjusted R-squared:  0.8854
## F-statistic:  78.3 on 5 and 45 DF,  p-value: < 2.2e-16

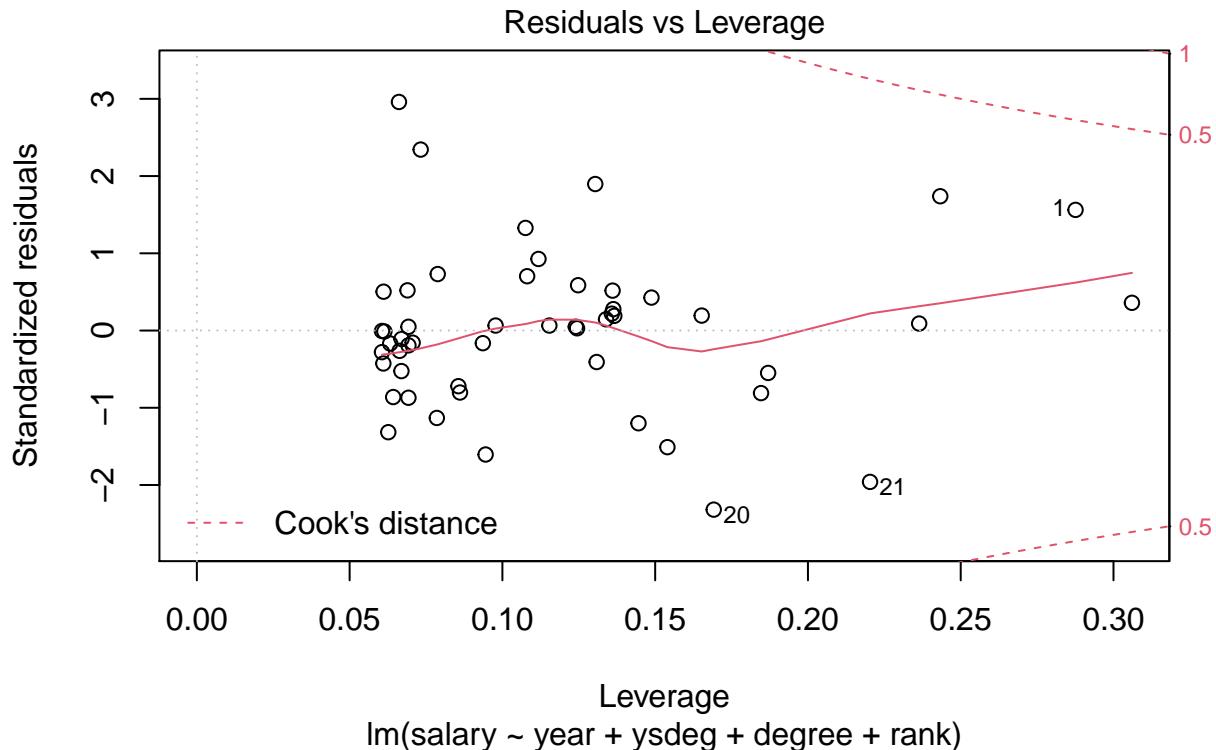
```

```
plot(m5_2)
```









En el gráfico de los residuos frente a los valores previstos no se aprecian problemas importantes. La observación 2 es ligeramente atípica, aunque no excesivamente. Se puede repetir el modelo eliminándola y se comprueba que no cambia sustancialmente.

Conclusiones

El modelo de regresión nos permite construir una ecuación que explica la influencia de un conjunto de variables (regresores) en otra variable (variable dependiente o variable respuesta).

El proceso de estimación implica muchas decisiones complejas y en algún caso, subjetiva: transformar/eliminar variables y/o observaciones del modelo. El procedimiento puede dar lugar a modelos distintos que aunque en lo fundamental pueden coincidir pueden diferir en algunos detalles que serán o no importantes en función del objetivo del análisis.

En la aplicación de estas técnicas es bueno tener presente que los resultados de la estimación dependen de la muestra que estamos analizando y que las conclusiones alcanzadas tienen que ser coherentes y reproducibles con otra muestra diferente.

En el caso particular que estudiamos, el de los salarios de los profesores de universidad podemos concluir lo siguiente:

1. El modelo predice el salario con $R^2 = 90\%$ (aprox.).
2. Las variables que más influyen son la antigüedad y la categoría del profesor. También influyen el grado y los años que lleva en posesión del título, aunque el coeficiente de esta última variable es difícil de interpretar.
3. No se han encontrado diferencias significativas en el salario de hombres y mujeres.

4. La ecuación del modelo permite predecir el salario anual con un error medio de 1726 \$
5. En el estudio se ha eliminado una observación (número 24) que tenía un comportamiento muy diferente al resto. Este hecho implica que puede haber profesores cuyos salarios no se ajustan a las predicciones de este modelo. En nuestro estudio hay 1 caso en una muestra de 52 (el 2% de observaciones atípicas)

TAREA 1: Medidas del Cuerpo Humano

Solución

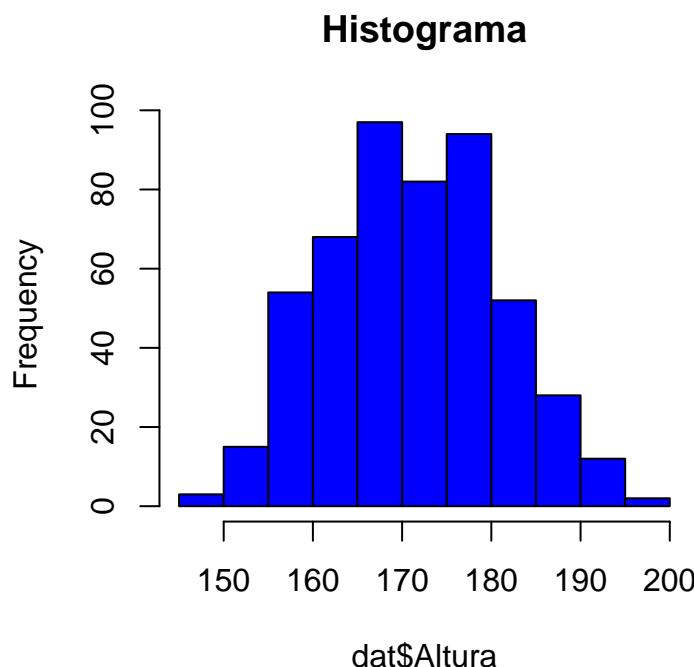
```
dat = read.table("cuerpo.txt", header=TRUE)
dat$Sexo = factor(dat$Sexo, labels = c("Mujer", "Hombre"))
```

Preguntas

Apartado 1

Realiza el histograma de la variable *Altura*, proporciona la media y la desviación típica. ¿Cuántas personas miden más de 180?

```
hist(dat$Altura, col="blue", main="Histograma")
```



```

m = mean(dat$Altura)
sprintf("La media es %.1f", m)

## [1] "La media es 171.1"

s = sd(dat$Altura)
sprintf("La desviación típica %.1f", s)

## [1] "La desviación típica 9.4"

sum(dat$Altura > 180)

## [1] 94

```

Hay 94 personas que miden igual o más de 180 cm.

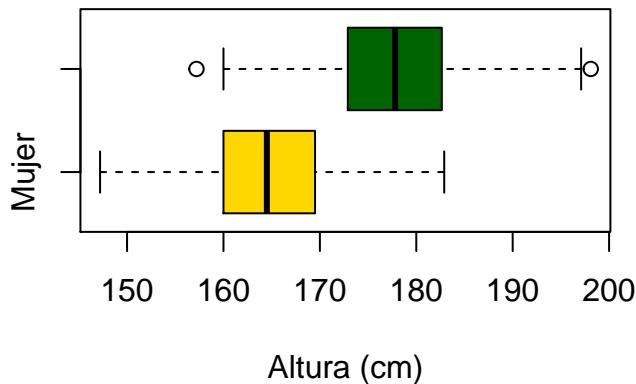
Apartado 2

Realiza un gráfico con el boxplot de la altura para hombres y mujeres. Describe las diferencias que se observan.

```

boxplot(dat$Altura ~ dat$Sexo, horizontal = TRUE,
        col= c("gold","darkgreen"),
        xlab = "Altura (cm)",
        ylab = "")

```



Se observa que los hombres son más altos en general que las mujeres. La dispersión de las dos distribuciones es muy parecida. En el gráfico está toda la información relevante. Por ejemplo, se observa que 75% de las mujeres tienen una estatura por debajo de 169.5. Sin embargo, el 75% de los hombres están por encima de 172.9 cm.

Abajo, se coloca información adicional con los valores más representativos de la figura (no es necesario incluirla en el trabajo).

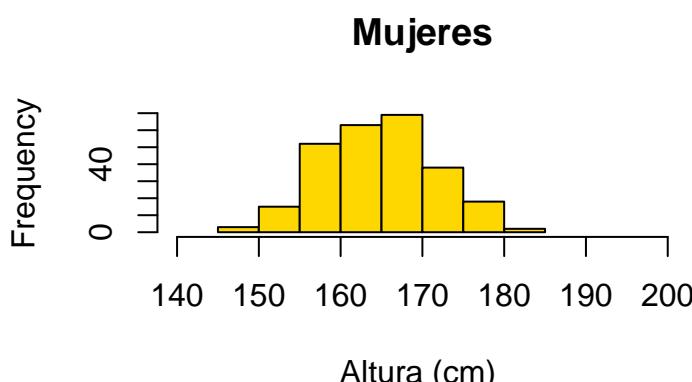
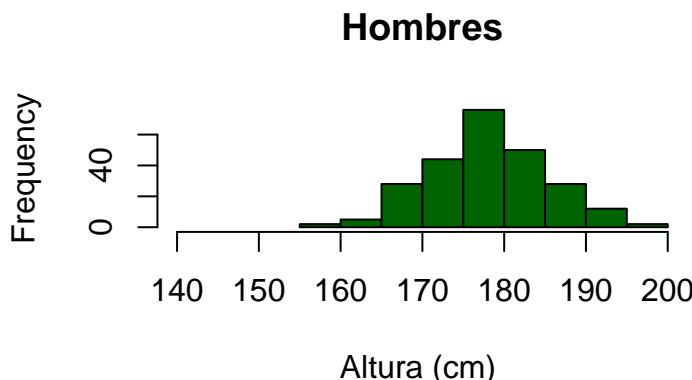
Apartado 3

Realiza un gráfico con dos histogramas, en la parte superior coloca el histograma de la *Altura* de hombres y en la inferior el de mujeres. Utiliza la misma escala en los dos histogramas. Calcula la media y la desviación típica asociados a cada uno de los histogramas. Interpreta los resultados.

```
par(mfrow = c(2,1))
hom = dat$Sexo == "Hombre"
muj = dat$Sexo == "Mujer"

hist(dat$Altura[hom], col = "darkgreen",
      xlim=c(140,200), main = "Hombres",
      xlab = "Altura (cm)")

hist(dat$Altura[muj], col = "gold",
      xlim=c(140,200), main = "Mujeres",
      xlab = "Altura (cm) ")
```



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

print("medias")
```

```

## [1] "medias"

tapply(dat$Altura,dat$Sexo,mean)

##      Mujer     Hombre
## 164.8723 177.7453

print("desviaciones típicas")

## [1] "desviaciones típicas"

tapply(dat$Altura,dat$Sexo,sd)

##      Mujer     Hombre
## 6.544602 7.183629

```

Este gráfico proporciona una información muy parecida al boxplot del apartado anterior. Se aprecia que las distribuciones de las estaturas siguen la forma “normal”, simétrica. La estatura media de los hombres es muy superior a la de las mujeres, cerca de 11 cm de diferencia. Las desviaciones típicas son similares.

Apartado 4

Compara las medias de hombres y mujeres para las medidas musculares (variables que empiezan por C, van de la 10 a la 21 ambas inclusive).

- (a) Realiza un boxplot (múltiple en función de la variable Sexo) para la variable donde haya una mayor diferencia entre la media de hombres y mujeres.
- (b) Realiza un boxplot (múltiple en función de la variable Sexo) para la variable donde la media de las medidas de las mujeres sea superior a la media de las medidas de los hombres.

```

m1 = sapply(dat[hom,10:21],mean)
m2 = sapply(dat[muj,10:21], mean)
t1 = cbind(Hombres=m1,Mujeres=m2,Dif=m1-m2) # hace una tabla "pegando" tres columnas
print(t1,digits=3)

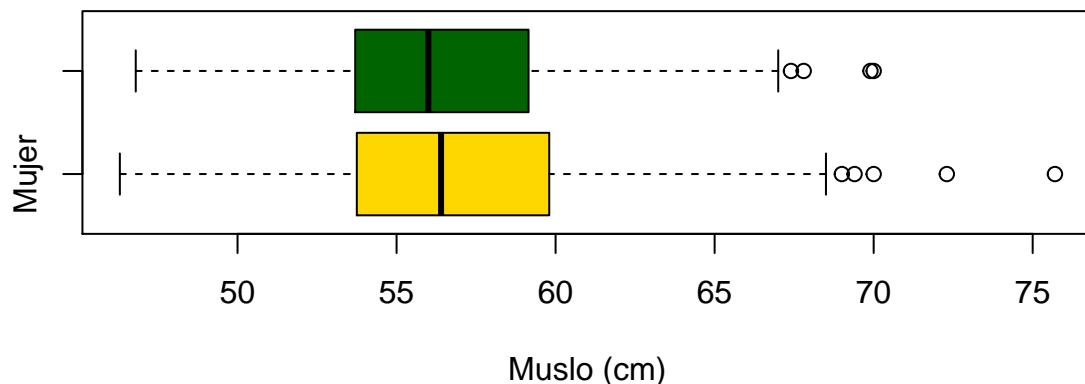
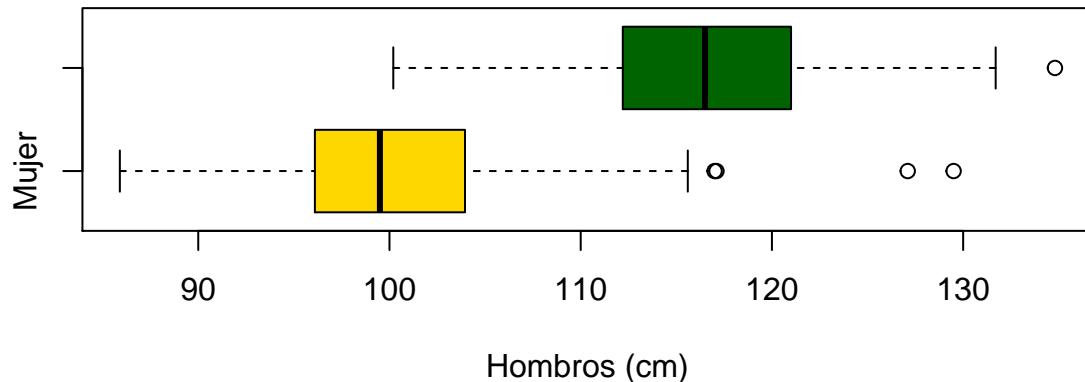
```

	Hombres	Mujeres	Dif
## C_Hombros	116.5	100.3	16.198
## C_Pecho	101.0	86.1	14.930
## C_Cintura	84.5	69.8	14.730
## C_Abdomen	87.7	83.7	3.917
## C_Cadera	97.8	95.7	2.110
## C_Muslo	56.5	57.2	-0.698
## C_Bicep	34.4	28.1	6.306
## C_Brazo	28.2	23.8	4.480
## C_Rodilla	37.2	35.3	1.936
## C_Gemelo	37.2	35.0	2.201
## C_Tobillo	23.2	21.2	1.953
## C_Muneca	17.2	15.1	2.131

La mayor diferencia entre las medias es en el perímetro de hombros y la menor diferencia es en el perímetro de muslos, en esta variable la media de las mujeres es ligeramente superior a la de los hombres.

```
par(mfrow=c(2,1))
boxplot(dat$C_Hombros~dat$Sexo, horizontal = TRUE,
        col= c("gold","darkgreen"),
        xlab = "Hombros (cm)",
        ylab = "")

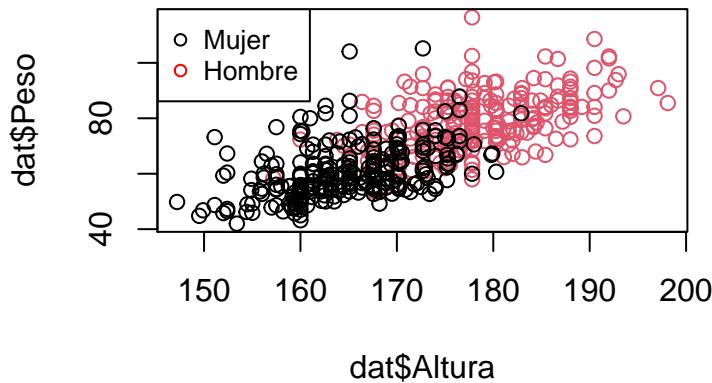
boxplot(dat$C_Muslo~dat$Sexo, horizontal = TRUE,
        col= c("gold","darkgreen"),
        xlab = "Muslo (cm)",
        ylab = "")
```



Apartado 5

- (a) Realiza un gráfico de dispersión del peso de una persona en función de la altura, utiliza un color diferente para hombres y mujeres. Calcula la correlación entre las dos variables con todos los datos, y para hombres y para mujeres. Comenta los resultados.

```
plot(dat$Peso ~ dat$Altura, col=dat$Sexo)
legend("topleft", legend=c("Mujer","Hombre"),
       col=c( "black","red"), pch=1,cex=0.8)
```



```
cor(dat$Peso,dat$Altura)
```

```
## [1] 0.7173011
```

```
cor(dat$Peso[hom] ,dat$Altura[hom])
```

```
## [1] 0.5347418
```

```
cor(dat$Peso[muj] ,dat$Altura[muj])
```

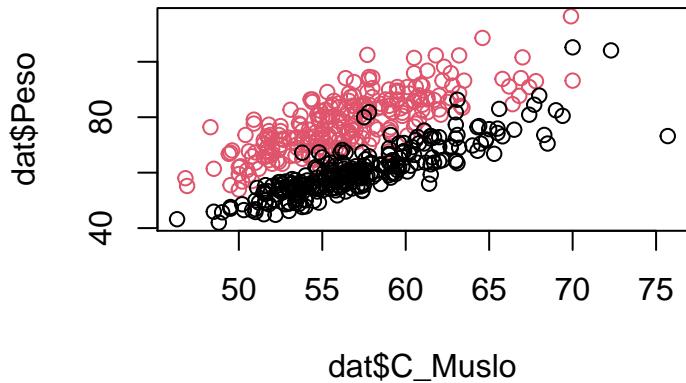
```
## [1] 0.4310593
```

En el gráfico de dispersión se aprecia dos “distribuciones” (nubes) diferentes. La nube de hombres (puntos rojos) está centrada en 177.7cm y 78.1kg. El centro de las mujeres es 164.9cm 60.6kg.

La correlación para el hombres es 0.53 y la de mujeres 0.43. La correlación cerca de cero indica que la nube es circular. Conforme la correlación aumenta, la nube es más elíptica. En el caso extremo de correlación 1 (o -1), la nube se concentra en una recta. Lo que se aprecia en la figura es que la nube conjunta, sin diferenciar hombres y mujeres, es más elíptica y por tanto tiene una correlación mayor que las otras dos por separado.

- (b) Repite el apartado 5 (a), pero utiliza ahora las variables *Peso* y *C_Muslo*. Explica las diferencias entre los resultados de (a) y (b).

```
plot(dat$Peso ~ dat$C_Muslo, col=dat$Sexo)
```



```
cor(dat$Peso,dat$C_Muslo)
```

```
## [1] 0.5585626
```

```
cor(dat$Peso[hom],dat$C_Muslo[hom])
```

```
## [1] 0.7722913
```

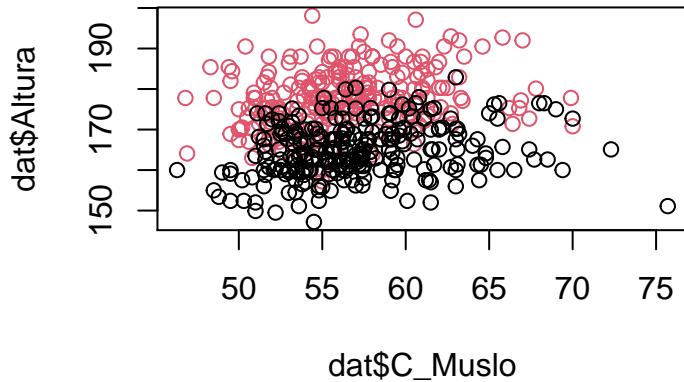
```
cor(dat$Peso[muj],dat$C_Muslo[muj])
```

```
## [1] 0.8567074
```

En este caso ocurre lo contrario. La nube (distribución conjunta) es más circular que cada una por separado. Por ejemplo el gráfico muestra que para el caso de las mujeres la nube es mucho más alargada, los puntos están más próximos al eje de la nube lo que indica una mayor correlación entre las dos variables.

- Repite el apartado 5(a) y 5(b) pero utiliza ahora las variables *Altura* y *C_Muslo*. Explica las diferencias entre los resultados de los tres apartados (a), (b) y (c).

```
plot(dat$Altura ~ dat$C_Muslo, col=dat$Sexo)
```



```
cor(dat$Altura,dat$C_Muslo)
```

```
## [1] 0.1163097
```

```
cor(dat$Altura[hom],dat$C_Muslo[hom])
```

```
## [1] 0.2354401
```

```
cor(dat$Altura[muj],dat$C_Muslo[muj])
```

```
## [1] 0.234161
```

Estas dos variables presentan bajas correlaciones y se aprecia en la forma de las nubes de puntos.

Apartado 6.

Estima el modelo de regresión para explicar el **Peso** de una persona en función de la **Altura** y el **Sexo**. Interpreta cada uno de los parámetros del modelo.

```
m1 = lm(Peso~Altura+Sexo, data=dat)
summary(m1)
```

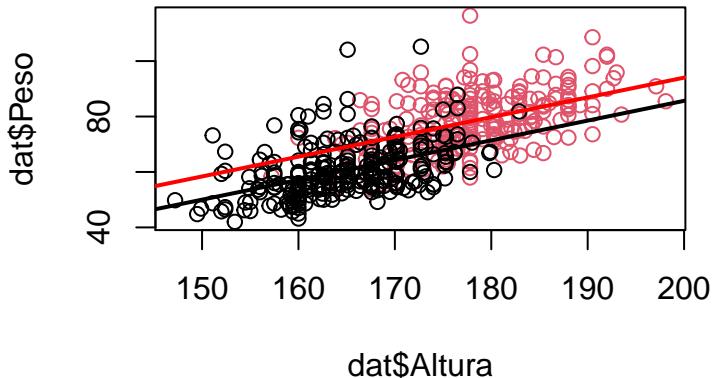
```
##
## Call:
## lm(formula = Peso ~ Altura + Sexo, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.184  -5.978  -1.356   4.709  43.337
##
## Coefficients:
```

```

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -56.94949   9.42444 -6.043 2.95e-09 ***
## Altura       0.71298   0.05707 12.494 < 2e-16 ***
## SexoHombre   8.36599   1.07296  7.797 3.66e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.802 on 504 degrees of freedom
## Multiple R-squared:  0.5668, Adjusted R-squared:  0.5651
## F-statistic: 329.7 on 2 and 504 DF, p-value: < 2.2e-16

plot(dat$Peso ~ dat$Altura, col=dat$Sexo)
abline(-56.94,0.71298,col="black",lw=2)
abline(-56.94 + 8.366,0.71298,col="red",lw=2)

```



Intercept se corresponde con la ordenada en el origen y vale -56.94, su interpretación estricta según la ecuación es el peso de una mujer de Altura = 0 cm. No tiene sentido. La explicación: El modelo que se estima tiene valor en el rango de los datos que hemos tomado. Como se aprecia en el gráfico las estaturas van de 140 cm a 200 cm, nuestro modelo es válido en este rango. Fuera de este rango no tenemos información y es arriesgado extrapolar. La ordenada del origen corresponde a un valor que se encuentra muy alejado del rango de validez del modelo. Ésto no significa que haya que suprimir el parámetro, es necesario para que la recta se ajuste a los datos en el rango de interés.

El coeficiente que afecta a la "Altura" es 0.71298 e indica que según el modelo el peso de una persona aumenta 0.712 kg por cada cm de estatura. El pvalor es muy pequeño y la conclusión es que su efecto es muy significativo.

En el modelo aparece una variable con el nombre SexoHombre que toma el valor 0 cuando la persona es una mujer y el valor 1 cuando la persona es un hombre. El coeficiente 8.36599 indica que los hombres a igualdad de Altura pesan 8.36599 kg más que las mujeres. El pvalor es muy pequeño y por tanto se concluye que es un efecto significativo.

La desviación típica residual es 8.802 kg, indica que este modelo predice el peso de una persona con un error medio de 8.802 kg.

$R^2 = 0.5668$, es el coeficiente de determinación, se interpreta como el porcentaje de variabilidad del peso que está explicado por el modelo. La Altura y el Sexo explican parcialmente el peso de una persona, hay otras variables como se verá más adelante que mejoran la capacidad predictiva del modelo.

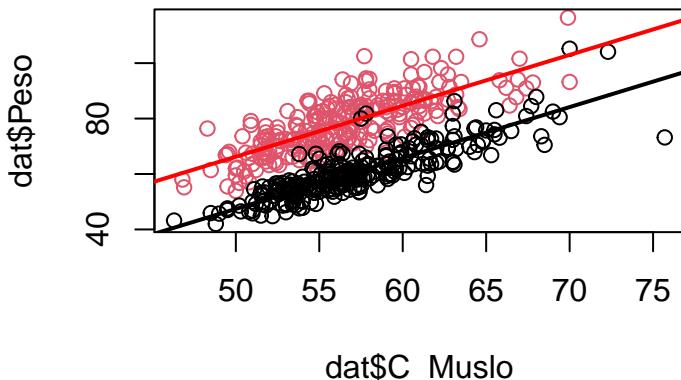
Apartado 7.

Estima el modelo del **Peso** como variable dependiente en función de **C_Muslo** y **Sexo**. Representa gráficamente el modelo en un gráfico de dispersión. Explica las diferencias entre el modelo del apartado 6 y 7. ¿Qué modelo es mejor?

```
m2 = lm(Peso~C_Muslo+Sexo, data=dat)
summary(m2)

##
## Call:
## lm(formula = Peso ~ C_Muslo + Sexo, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -21.3884  -3.6732  -0.2318   3.0220  22.1476 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -44.45503   3.37873 -13.16   <2e-16 ***
## C_Muslo      1.83677   0.05873  31.27   <2e-16 ***
## SexoHombre   18.82584   0.52350  35.96   <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.874 on 504 degrees of freedom
## Multiple R-squared:  0.8071, Adjusted R-squared:  0.8063 
## F-statistic: 1054 on 2 and 504 DF,  p-value: < 2.2e-16

plot(dat$Peso ~ dat$C_Muslo, col=dat$Sexo)
abline(-44.45,1.837,col="black",lw=2)
abline(-44.45+18.82,1.837,col="red",lw=2)
```



La interpretación del modelo es similar. Este modelo es mejor como se aprecia en: (1) la desviación típica residual es menor y el coeficiente de determinación es mucho más alto. El modelo explica el 80.7 % de la variabilidad del peso.

Apartado 8

Estima el modelo de regresión múltiple entre el **Peso** utilizando como regresores las medidas musculares (empiezan por C), la **Altura** y el **Sexo**. Interpreta brevemente los parámetros del modelo. Explica el parámetro estimado que afecta a la variable **Sexo**.

```
names(dat)
```

```
## [1] "A_Hombros" "A_Pelvis"   "A_Cade"      "AP_Pecho"    "AD_Pecho"    "A_Codo"
## [7] "A_Muneca"   "A_Rodilla"   "A_Tobillo"   "C_Hombros"   "C_Pecho"     "C_Cintura"
## [13] "C_Abdomen"  "C_Cadera"   "C_Muslo"     "C_Bicep"     "C_Brazo"     "C_Rodilla"
## [19] "C_Gemelo"   "C_Tobillo"  "C_Muneca"   "Edad"       "Peso"        "Altura"
## [25] "Sexo"
```

```
m3 = lm(Peso ~ ., data = dat[,c(10:21,23,24,25)])
summary(m3)
```

```
##
## Call:
## lm(formula = Peso ~ ., data = dat[, c(10:21, 23, 24, 25)])
##
## Residuals:
##      Min      1Q Median      3Q      Max
## -7.5941 -1.3736  0.0663  1.2392 10.4202
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -122.04195  2.66430 -45.806 < 2e-16 ***
## C_Hombros    0.08450  0.02991  2.825 0.004914 **
## C_Pecho      0.19382  0.03566  5.435 8.63e-08 ***
## C_Cintura    0.36426  0.02741 13.291 < 2e-16 ***
## C_Abdomen   -0.01240  0.02396 -0.517 0.605075
## C_Cadera     0.22961  0.04360  5.266 2.09e-07 ***
## C_Muslo      0.29047  0.05285  5.496 6.26e-08 ***
## C_Bicep      0.07594  0.08579  0.885 0.376481
## C_Brazo      0.58569  0.13965  4.194 3.25e-05 ***
## C_Rodilla    0.29186  0.07736  3.773 0.000181 ***
## C_Gemelo     0.39859  0.06992  5.700 2.06e-08 ***
## C_Tobillo   -0.00109  0.09976 -0.011 0.991290
## C_Muneca    -0.10790  0.19547 -0.552 0.581181
## Altura       0.33890  0.01657 20.452 < 2e-16 ***
## SexoHombre   -0.99262  0.52583 -1.888 0.059654 .
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.198 on 492 degrees of freedom
## Multiple R-squared:  0.9736, Adjusted R-squared:  0.9729
## F-statistic: 1297 on 14 and 492 DF,  p-value: < 2.2e-16
```

El modelo explica el 97.36% de la variabilidad del peso. Es un buen modelo. Con el se predice el peso de una persona con un error de 2.198 kg. La mayoría de las variables tienen efecto positivo y son significativas. Algunas variables no son significativas, tres de ellas tienen coeficiente negativo.

El coeficiente que afecta a la variable SexoHombre es -0.9926, tiene un pvalor = 0.59 lo que implica que no es significativo. Además ha cambiado de signo. Al incluir todas las variables musculares, las diferencias entre Hombres y mujeres desaparecen. A igualdad de medidas musculares no existen diferencias significativas entre hombres y mujeres en el peso. Si tomamos nivel de significación $\alpha = 0.10$, si existen diferencias significativas, y el parámetro indica que a igualdad del resto de las medidas, la mujer pesa casi un kilograma más (992 gr) que el hombre.)

Apartado 9

Utiliza el método de stepwise para eliminar las variables no importantes.

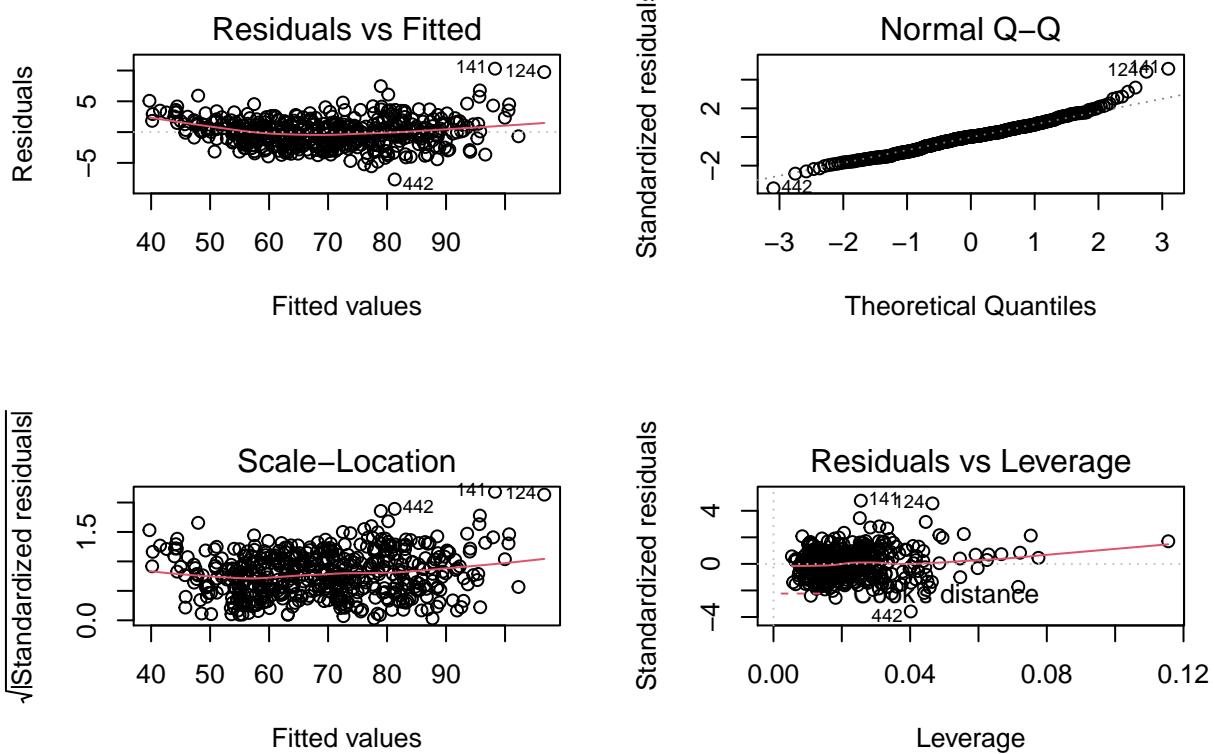
```
m4 = step(m3,trace = 0)
summary(m4)

##
## Call:
## lm(formula = Peso ~ C_Hombros + C_Pecho + C_Cintura + C_Cadera +
##      C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura + Sexo,
##      data = dat[, c(10:21, 23, 24, 25)])
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -7.6911 -1.3476  0.0486  1.2660 10.3018
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -122.58889   2.55775 -47.928 < 2e-16 ***
## C_Hombros     0.09026   0.02903   3.109 0.001983 **
## C_Pecho       0.19626   0.03409   5.758 1.50e-08 ***
## C_Cintura     0.35982   0.02452  14.675 < 2e-16 ***
## C_Cadera      0.21615   0.03886   5.562 4.38e-08 ***
## C_Muslo        0.31408   0.04789   6.559 1.36e-10 ***
## C_Brazo        0.62535   0.09978   6.267 7.99e-10 ***
## C_Rodilla      0.27480   0.07421   3.703 0.000237 ***
## C_Gemelo       0.38997   0.06356   6.136 1.74e-09 ***
## Altura         0.33535   0.01615  20.762 < 2e-16 ***
## SexoHombre     -0.89385   0.49751  -1.797 0.073002 .
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.192 on 496 degrees of freedom
## Multiple R-squared:  0.9736, Adjusted R-squared:  0.973
## F-statistic:  1826 on 10 and 496 DF,  p-value: < 2.2e-16
```

Apartado 10

Realiza la diagnosis del modelo final y explica si es válido. Indica qué condiciones del modelo no se cumplen.

```
par(mfrow=c(2,2))
plot(m4)
```



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

En el gráfico de residuos frente a valores previstos se observa falta de linealidad y ligera homocedasticidad. El modelo no es perfecto, aunque la desviación es muy poco importante.

El gráfico de normalidad no presenta grandes desviaciones, se puede aceptar la hipótesis como válida.

APÉNDICE (No se pedía en la tarea)

Extra: Como mejorar el modelo

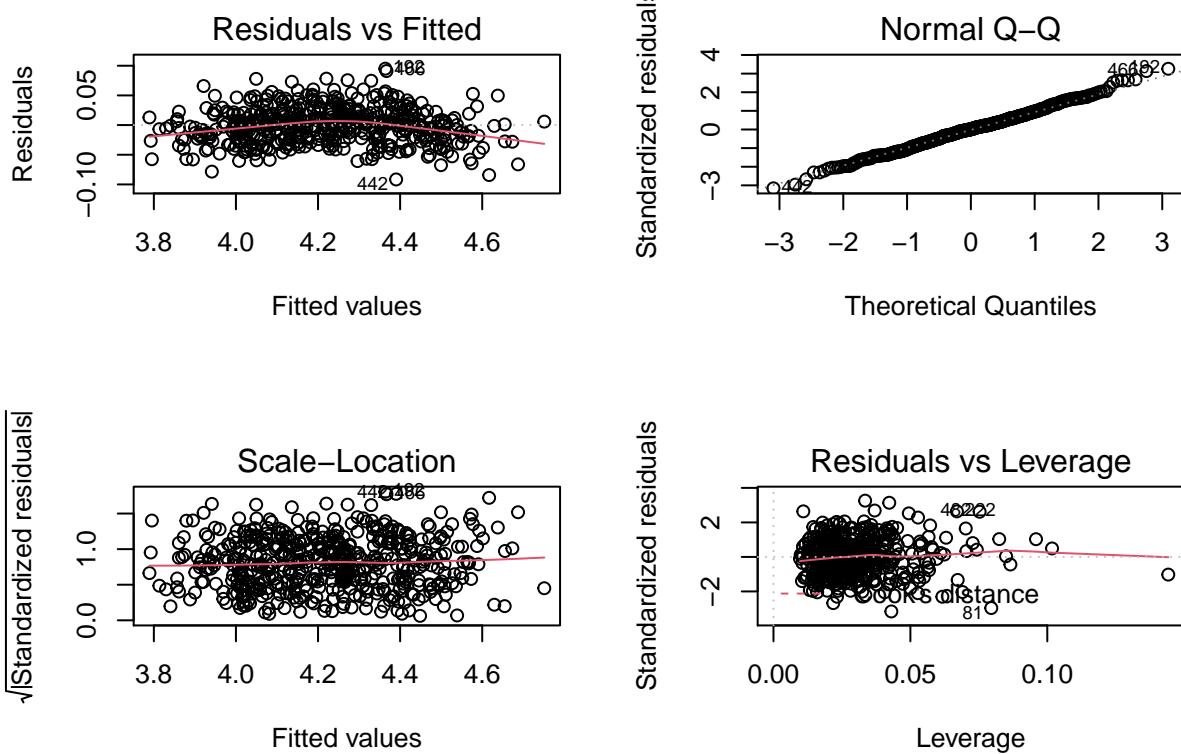
En estas páginas se presenta una forma de corregir las desviaciones. Estos apartados no se pedían en la tarea y no cuentan en la evaluación.

Transformación logarítmica

```
m6 = lm(log(Peso) ~ ., data = dat[,c(10:21,23,24,25)])
m7 = step(m6,trace=0)
summary(m7)

##
## Call:
## lm(formula = log(Peso) ~ C_Hombros + C_Pecho + C_Cintura + C_Abdomen +
##      C_Cadera + C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura,
##      data = dat[, c(10:21, 23, 24, 25)])
##
## Residuals:
##       Min     1Q Median     3Q    Max 
## -0.091613 -0.019229  0.000592  0.018635  0.094553
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.4235148  0.0319877 44.502 < 2e-16 ***
## C_Hombros   0.0012142  0.0003946  3.077 0.002208 ** 
## C_Pecho     0.0027825  0.0004687  5.936 5.48e-09 ***
## C_Cintura   0.0040378  0.0003274 12.333 < 2e-16 ***
## C_Abdomen   0.0004768  0.0003032  1.572 0.116494  
## C_Cadera    0.0035532  0.0005745  6.185 1.30e-09 *** 
## C_Muslo     0.0046344  0.0006359  7.288 1.25e-12 *** 
## C_Brazo     0.0103454  0.0012274  8.428 3.82e-16 *** 
## C_Rodilla   0.0034982  0.0010026  3.489 0.000528 *** 
## C_Gemelo    0.0055362  0.0008657  6.395 3.71e-10 *** 
## Altura      0.0049651  0.0002051 24.206 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02965 on 496 degrees of freedom
## Multiple R-squared:  0.9767, Adjusted R-squared:  0.9763 
## F-statistic:  2083 on 10 and 496 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(m6)
```

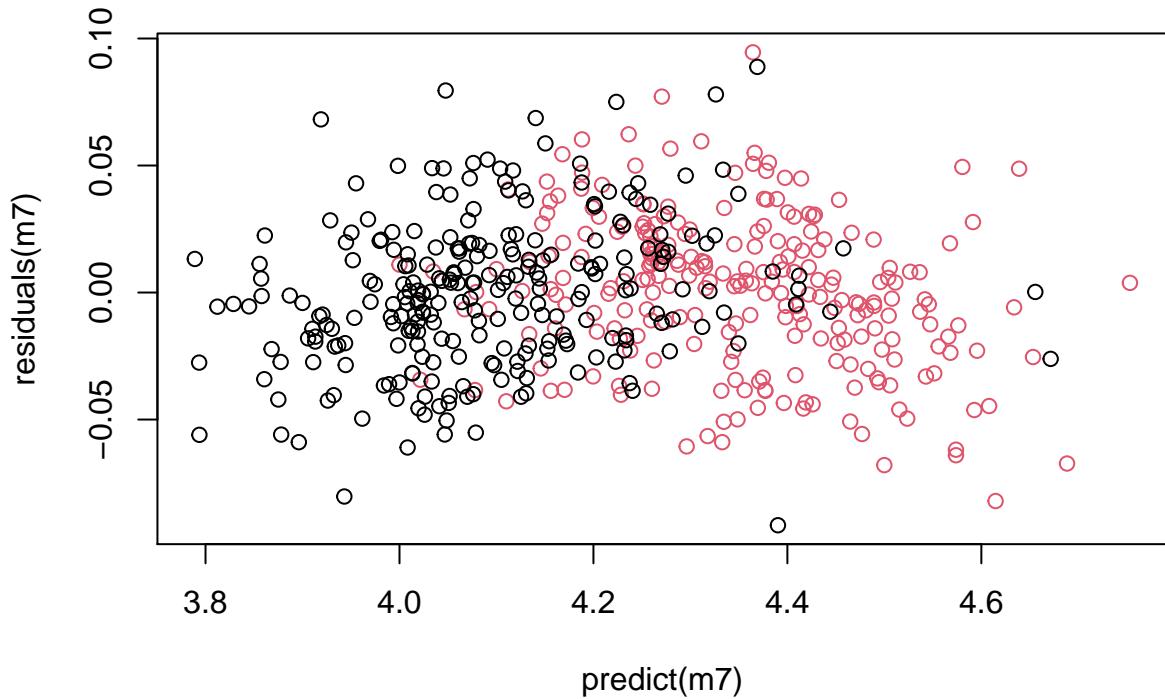


```
par(mfrow=c(1,1))
```

No parece que resuelva el problema.

En el gráfico de abajo se aprecia que la falta de linealidad es debida a que los residuos de hombres y mujeres tienen un comportamiento diferente.

```
plot(predict(m7),residuals(m7),col=dat$Sexo)
```



Una manera de controlar este comportamiento es incluyendo interacciones en el modelo: el efecto de las variables en el peso es diferente para hombres que para mujeres.

```
options(digits=3)
m8=lm(formula = log(Peso) ~ C_Hombros + C_Pecho + C_Cintura + C_Abdomen +
      C_Cadera + C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura+ Sexo*(C_Hombros + C_Pecho + C_Cintura +
      C_Cadera + C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura),
      data = dat[, c(10:21, 23, 24, 25)])
summary(m8)

##
## Call:
## lm(formula = log(Peso) ~ C_Hombros + C_Pecho + C_Cintura + C_Abdomen +
##     C_Cadera + C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura +
##     Sexo * (C_Hombros + C_Pecho + C_Cintura + C_Abdomen + C_Cadera +
##             C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura), data = dat[, ,
##             c(10:21, 23, 24, 25)])
##
## Residuals:
##       Min        1Q        Median        3Q       Max
## -0.11219 -0.01878  0.00055  0.01816  0.09082
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.34e+00  5.00e-02 26.80 < 2e-16 ***
## C_Hombros                1.44e-04  5.36e-04   0.27   0.7880
```

```

## C_Pecho          4.54e-03   7.33e-04    6.19  1.3e-09 ***
## C_Cintura        4.09e-03   5.76e-04    7.11  4.2e-12 ***
## C_Abdomen        1.42e-04   3.89e-04    0.37  0.7145
## C_Cadera         4.04e-03   8.24e-04    4.91  1.3e-06 ***
## C_Muslo          4.79e-03   9.87e-04    4.85  1.7e-06 ***
## C_Brazo           6.14e-03   2.11e-03    2.91  0.0037 **
## C_Rodilla         3.83e-03   1.38e-03    2.77  0.0058 **
## C_Gemelo          9.01e-03   1.25e-03    7.20  2.2e-12 ***
## Altura            4.81e-03   3.12e-04   15.41 < 2e-16 ***
## SexoHombre        2.03e-01   7.08e-02    2.87  0.0043 **
## C_Hombros:SexoHombre 1.58e-03   7.74e-04    2.05  0.0411 *
## C_Pecho:SexoHombre -2.51e-03   9.39e-04   -2.68  0.0077 **
## C_Cintura:SexoHombre 3.18e-04   7.63e-04    0.42  0.6770
## C_Abdomen:SexoHombre 9.87e-05   6.49e-04    0.15  0.8792
## C_Cadera:SexoHombre -8.44e-04   1.13e-03   -0.74  0.4568
## C_Muslo:SexoHombre -4.12e-04   1.29e-03   -0.32  0.7498
## C_Brazo:SexoHombre  4.71e-03   2.70e-03    1.75  0.0813 .
## C_Rodilla:SexoHombre -1.50e-03   1.95e-03   -0.77  0.4423
## C_Gemelo:SexoHombre -5.22e-03   1.69e-03   -3.08  0.0022 **
## Altura:SexoHombre   3.22e-04   4.24e-04    0.76  0.4491
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0284 on 485 degrees of freedom
## Multiple R-squared:  0.979, Adjusted R-squared:  0.978
## F-statistic: 1.08e+03 on 21 and 485 DF, p-value: <2e-16

```

```
# m9 se obtiene como m9 = step(m8)
```

```
m9 = lm(formula = log(Peso) ~ C_Hombros + C_Pecho + C_Cintura + C_Cadera +
C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura + Sexo +
C_Hombros:Sexo + C_Pecho:Sexo + C_Cadera:Sexo + C_Brazo:Sexo +
C_Gemelo:Sexo, data = dat[, c(10:21, 23, 24, 25)])
summary(m9)
```

```
##
## Call:
## lm(formula = log(Peso) ~ C_Hombros + C_Pecho + C_Cintura + C_Cadera +
##     C_Muslo + C_Brazo + C_Rodilla + C_Gemelo + Altura + Sexo +
##     C_Hombros:Sexo + C_Pecho:Sexo + C_Cadera:Sexo + C_Brazo:Sexo +
##     C_Gemelo:Sexo, data = dat[, c(10:21, 23, 24, 25)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11162 -0.01895  0.00012  0.01856  0.09152
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.321619  0.039357 33.58 < 2e-16 ***
## C_Hombros                 0.000086  0.000523  0.16  0.8694
## C_Pecho                    0.004423  0.000641  6.90  1.6e-11 ***
## C_Cintura                  0.004412  0.000324 13.63 < 2e-16 ***
## C_Cadera                   0.004248  0.000588  7.22  2.0e-12 ***
```

```

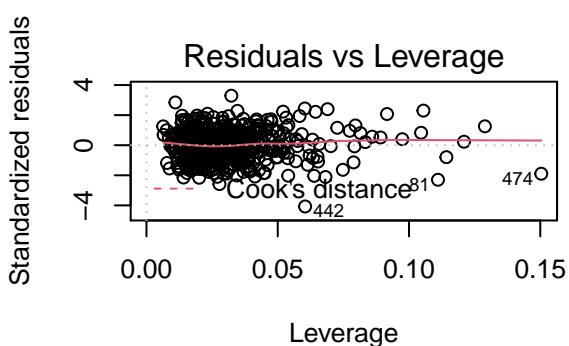
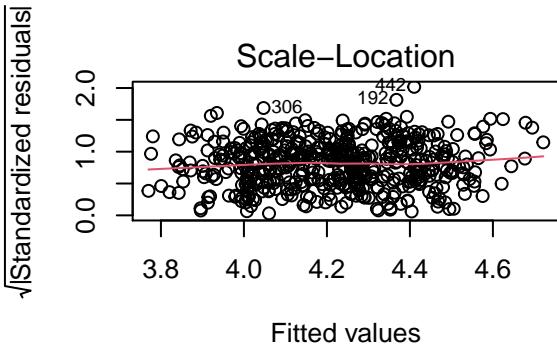
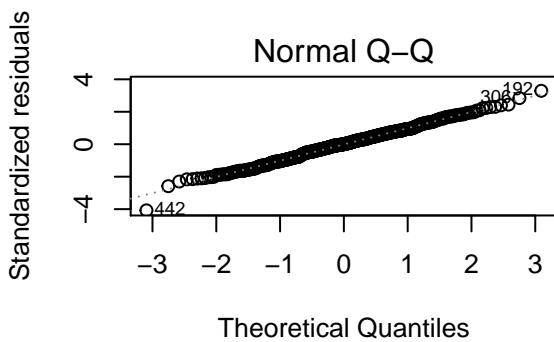
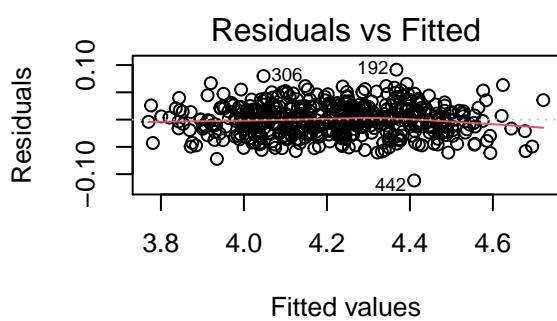
## C_Muslo          0.004472  0.000624    7.17  2.8e-12 ***
## C_Brazo          0.006462  0.002069    3.12  0.0019 **
## C_Rodilla        0.003138  0.000966    3.25  0.0012 **
## C_Gemelo         0.009284  0.001164    7.98  1.1e-14 ***
## Altura           0.004988  0.000210   23.75 < 2e-16 ***
## SexoHombre       0.233218  0.046367    5.03  6.9e-07 ***
## C_Hombres:SexoHombre 0.001587  0.000751    2.11  0.0350 *
## C_Pecho:SexoHombre -0.002254  0.000780   -2.89  0.0040 **
## C_Cadera:SexoHombre -0.000921  0.000660   -1.40  0.1635
## C_Brazo:SexoHombre  0.004155  0.002610    1.59  0.1121
## C_Gemelo:SexoHombre -0.005883  0.001455   -4.04  6.1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0283 on 491 degrees of freedom
## Multiple R-squared:  0.979, Adjusted R-squared:  0.978
## F-statistic: 1.53e+03 on 15 and 491 DF, p-value: <2e-16

```

```

par(mfrow=c(2,2))
plot(m9)

```



```

par(mfrow=c(1,1))

```

Ejercicio de Descriptiva y Regresión

TAREA 1: Análisis de Datos

Estudio de capacidad pulmonar de jóvenes

En este ejercicio se desea analizar la Capacidad Pulmonar en 654 jóvenes entre 3 y 19 año. Los datos han sido recogidos en Boston (USA) a finales de los 70. El archivo tiene 5 variables. La variable más importante es *fev* Forced Expiratory Volume (FEV) y proporciona el volumen de aire en litros exhalado en el primer segundo durante la espiración forzada tras una inspiración completa. Es un indicador muy utilizado por los especialistas en pulmón para evaluar la salud de una persona. Los valores normales de FEV en personas sanas dependen principalmente del sexo, la edad, la altura, el peso y la raza. En este estudio se proporciona para cada individuo la edad (*age*) en años, la altura (*ht*) en pulgadas, el género (*sex*) que toma el valor 0 si es una mujer y 1 si es hombre, y finalmente, si el individuo fuma o no (*smoke*) (No-fumador=0, fumador=1). El objetivo de este estudio era ver la relación entre la capacidad pulmonar (FEV) y el resto de las variables con especial interés en el hábito de fumar.

Fuente: Rosner, B. (1999), Fundamentals of Biostatistics, 5th Ed., Pacific Grove, CA: Duxbury

El archivo de datos se llama “fev.txt”.

Apartado 1 (1 punto)

Describe la variable *fev*. Compara gráfica y numéricamente la diferencia en la capacidad pulmonar (*fev*) de hombres y mujeres.

Apartado 2 (1 punto)

Estudia la relación entre las variables *ht*, *age* y *fev*. Explica gráficamente las relaciones entre estas tres variables.

Apartado 3. (1 punto)

Compara las tres variables *fev*, *ht* y *age* para fumadores y no fumadores. Utiliza gráficos y valores numéricos. Interpreta los resultados.

Apartado 4 (3 puntos)

- (a) Estima e interpreta el modelo de regresión simple entre *fev* (variable respuesta) y *ht* (altura) como regresor. Realiza la diagnosis del modelo.

- (b) Estima el modelo otra vez utilizando $\log(\text{fev})$ como variable respuesta. Realiza la diagnosis y comenta los resultados de la diagnosis.
- (c) Interpreta los coeficientes fundamentales del modelo estimado.

Apartado 5 (3 puntos)

- (a) Estima el modelo de regresión múltiple entre $\log(\text{fev})$ y el resto de las variables.
- (b) Realiza la diagnosis del modelo.
- (c) Interpreta el modelo obtenido.

Solución

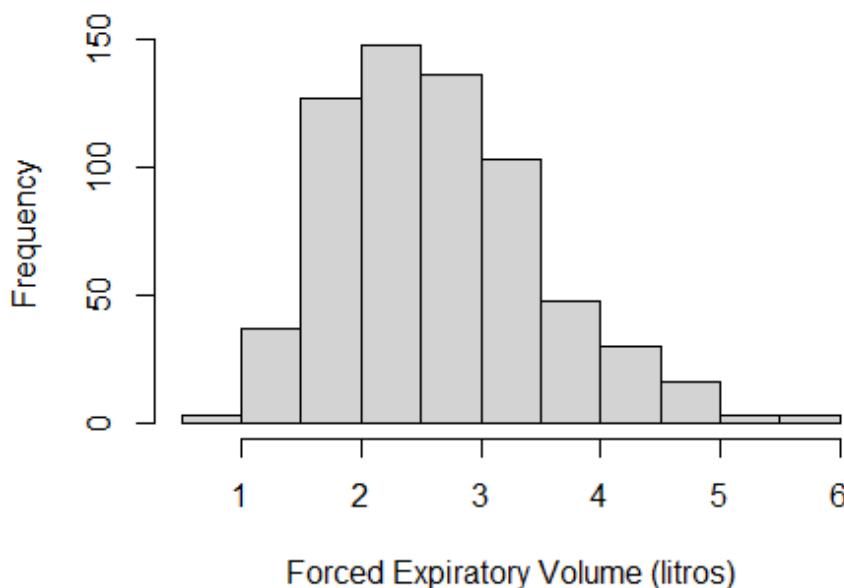
Apartado 1 (1 punto)

Describe la variable *fev*. Compara gráfica y numéricamente la diferencia en la capacidad pulmonar (*fev*) de hombres y mujeres. En este apartado no tengas en cuenta el resto de las variables.

```
dat = read.table("fev.txt",header=TRUE)
dat$sex=factor(dat$sex,labels=c("mujer","hombre"))
dat$smoke=factor(dat$smoke,labels=c("no","si"))
summary(dat)

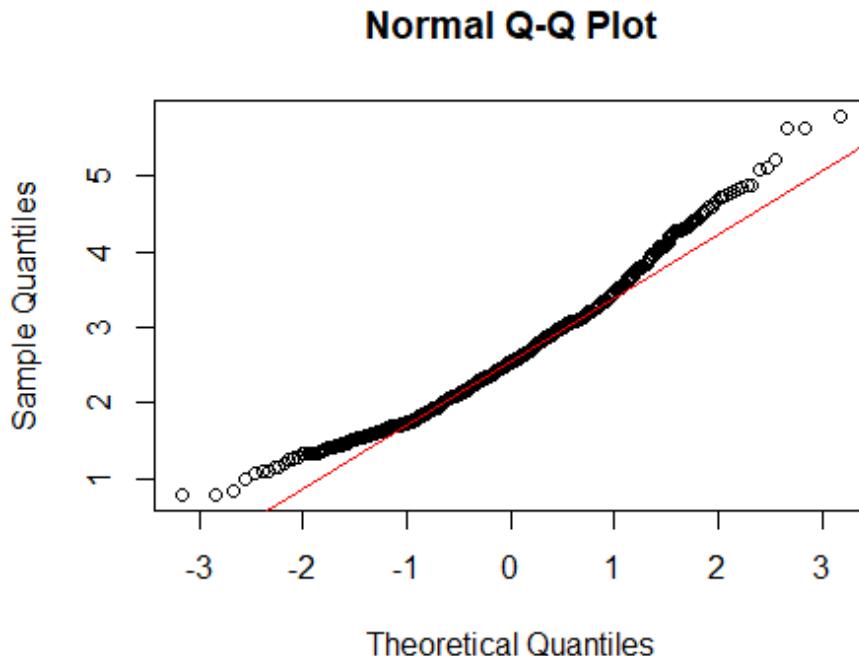
##      age          fev          ht          sex      smoke
##  Min.   : 3.000   Min.   :0.791   Min.   :46.00   mujer :318   no:589
##  1st Qu.: 8.000   1st Qu.:1.981   1st Qu.:57.00   hombre:336  si: 65
##  Median :10.000   Median :2.547   Median :61.50
##  Mean   : 9.931   Mean   :2.637   Mean   :61.14
##  3rd Qu.:12.000   3rd Qu.:3.119   3rd Qu.:65.50
##  Max.   :19.000   Max.   :5.793   Max.   :74.00

hist(dat$fev,main="",xlab="Forced Expiratory Volume (litros)")
```



Los datos tienen distribución ligeramente asimétrica. Se puede visualizar mejor la desviación respecto a la normal mediante el gráfico q-q.

```
qqnorm(dat$fev)
qqline(dat$fev, col="red")
```



La capacidad pulmonar media de los datos es 2.6 litros

```
mean(dat$fev)
```

```
## [1] 2.63678
```

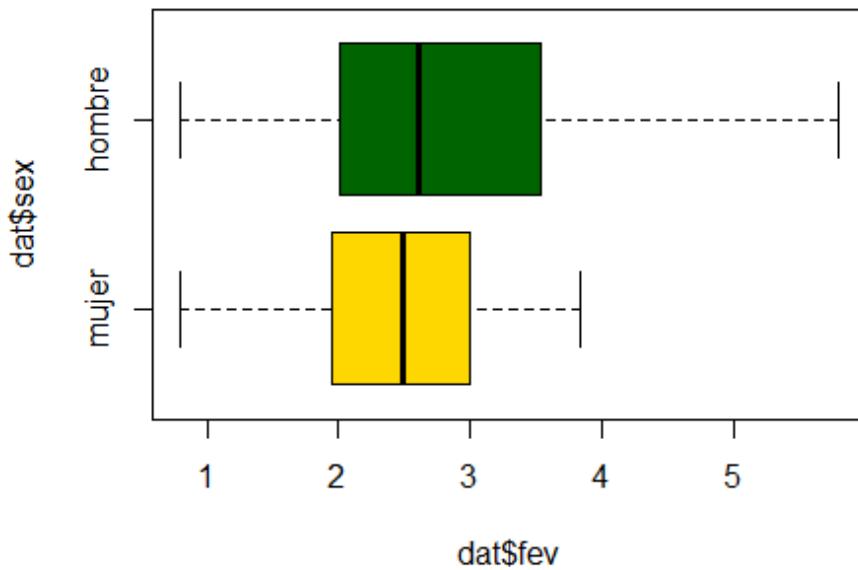
y la desviación típica 0.87 litros

```
sd(dat$fev)
```

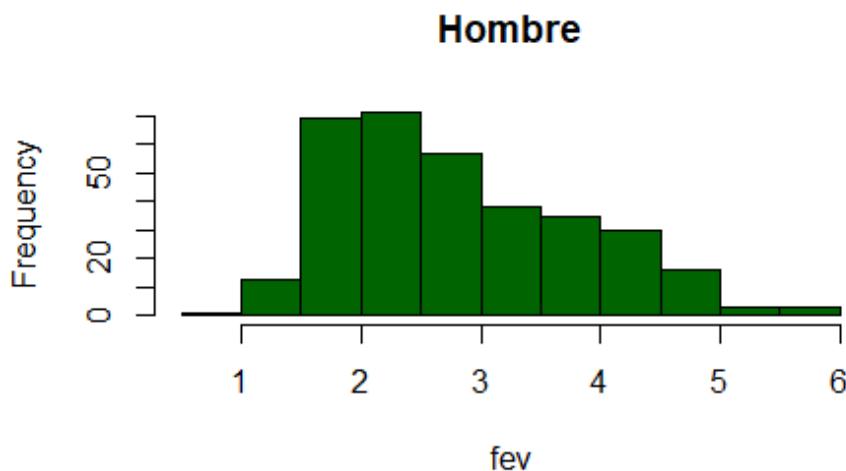
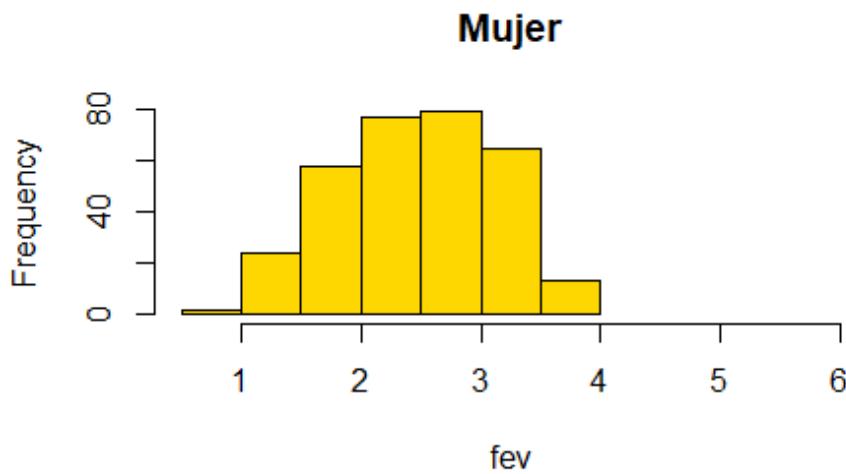
```
## [1] 0.8670591
```

Para comparar hombres y mujeres, el mejor gráfico es un boxplot

```
boxplot(dat$fev~dat$sex, col=c("gold", "darkgreen"), horizontal = TRUE)
```



```
par(mfrow=c(2,1))
h = dat$sex=="hombre"
hist(dat$fev[!h],col="gold",main="Mujer", xlim = c(0.5,6),xlab = "fev")
hist(dat$fev[h],col="darkgreen",main = "Hombre",xlim = c(0.5,6),xlab =
"fev")
```



```
tapply(dat$fev, dat$sex, mean)
##     mujer     hombre
## 2.451170 2.812446

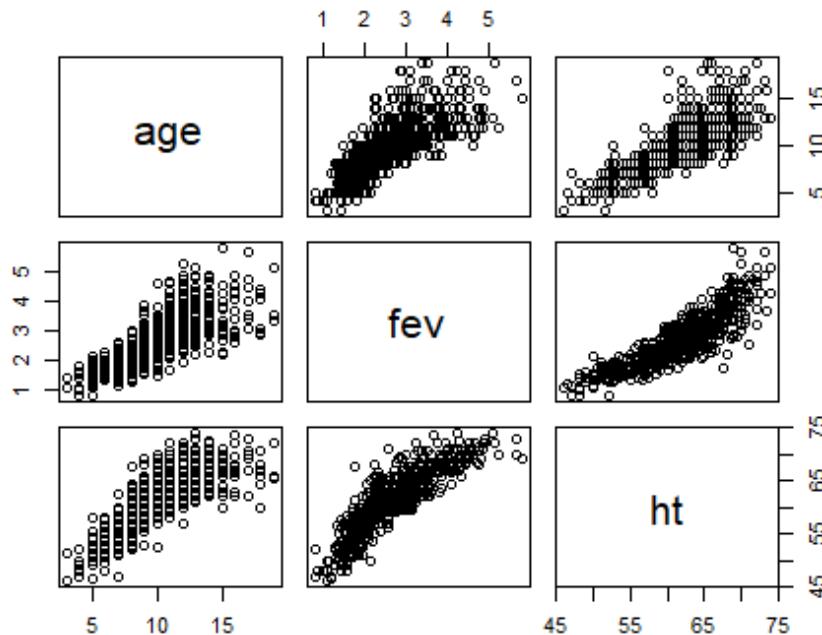
tapply(dat$fev, dat$sex, sd)
##     mujer     hombre
## 0.645736 1.003598
```

Se observa tanto en el boxplot, como en los histogramas y los valores numéricos que la capacidad pulmonar media de hombres y mujeres es similar, 2.45 litros para mujeres y 2.81 litros para hombres. La dispersión sin embargo es mayor para hombres (1 litro) que para mujeres (0.65 litros) como se aprecia en las figuras.

Apartado 2 (1 punto)

Estudia la relación entre las variables ht, age y fev. Explica gráficamente las relaciones entre estas tres variables.

```
pairs(dat[,1:3])
```



En las figuras se aprecia una clara relación entre las variables. La relación entre altura y edad es obvia. La capacidad pulmonar aumenta con la edad y también con la altura. La relación es más intensa entre fev y ht. La matriz de correlación de las tres variables es:

```
print(cor(dat[,1:3]))  
##           age      fev      ht  
## age  1.0000000 0.756459 0.7919436  
## fev  0.7564590 1.000000 0.8681350  
## ht   0.7919436 0.868135 1.0000000
```

Se detecta falta de linealidad en la relación de las variables. Se estudiará con detalle en los apartados 4 y 5.

Apartado 3. (1 punto)

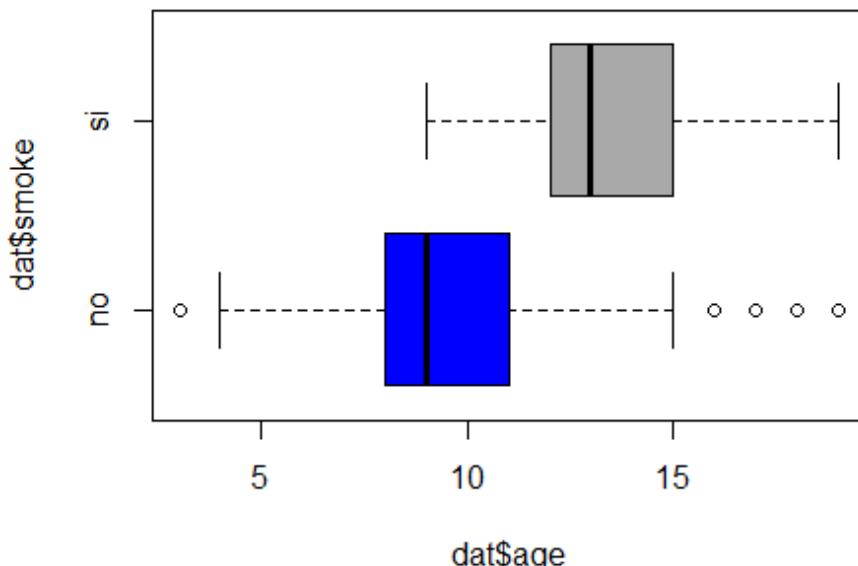
Compara las tres variables fev, ht y age para fumadores y no fumadores. Utiliza gráficos y valores numéricos. Interpreta los resultados.

De los 654 sólo (!) fuman 65 (el 10%). El porcentaje es muy alto con la mentalidad actual, en los años 70 la visión era distinta.

```
table(dat$smoke)  
##  
##  no   si  
## 589   65
```

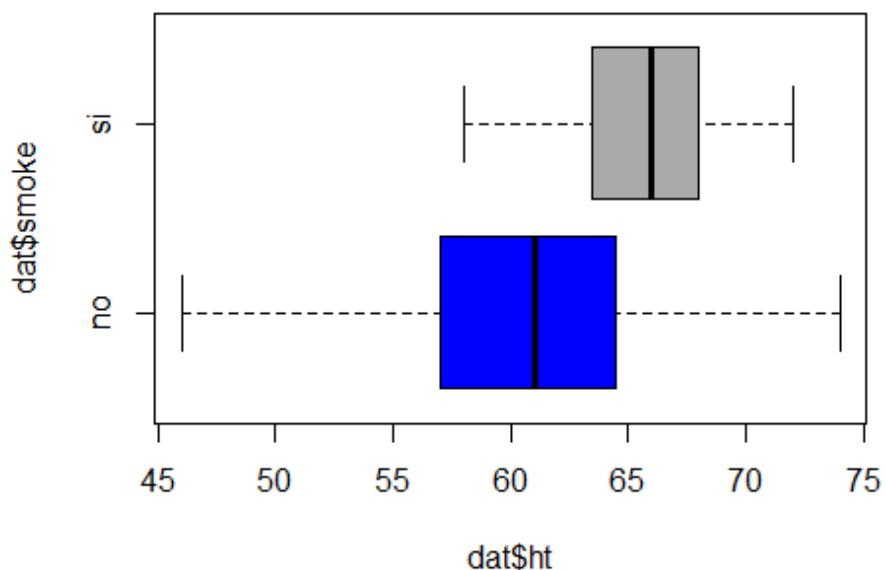
En el boxplot de la edad según **smoke** observamos que la distribución de edades de los dos grupos es muy distinta. Aunque sigue sorprendiendo la edad de algunos fumadores. Es curioso en el gráfico que en el grupo de no-fumadores son atípicas las edades por encima de 15 años.

```
boxplot(dat$age~dat$smoke, horizontal = TRUE, col=c("blue", "darkgrey"))
```

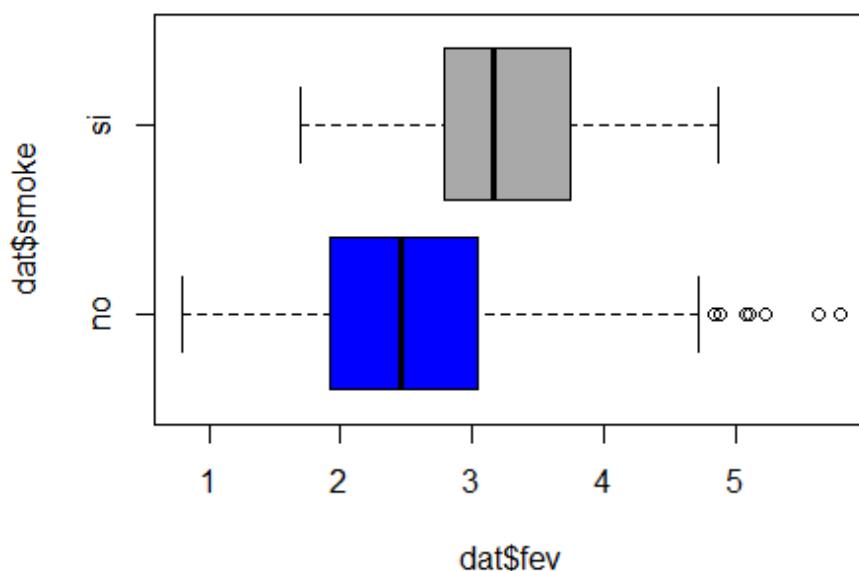


Teniendo en cuenta la diferencia de edades, es lógico que también haya diferencia en las otras dos variables **ht** y **fev**.

```
boxplot(dat$ht~dat$smoke, horizontal = TRUE, col=c("blue", "darkgrey"))
```



```
boxplot(dat$fev~dat$smoke, horizontal = TRUE, col=c("blue", "darkgrey"))
```



Se concluye que los fumadores son (en general) individuos de más edad, más altura y más capacidad pulmonar. Esto se puede medir con las medias de cada variable para cada uno de los grupos.

```
fuma = dat$smoke == "si"
m1 = sapply(dat[!fuma,1:3],mean)
m2 = sapply(dat[fuma,1:3],mean)
rbind(nofuma=m1,fuma=m2)

##           age      fev      ht
## nofuma  9.534805 2.566143 60.61273
## fuma    13.523077 3.276862 65.95385
```

Una pulgada son 2.54 cm, de manera que 60.6 pulgadas son 154.7 cm y 65.9 pulgadas son 167.4 cm.

Si se desea se proporciona los valores de las desviaciones típicas, aunque es suficiente con mirar los gráficos boxplot para ver que para age y fev son similares, y que son muy distintas para ht.

```
fuma = dat$smoke == "si"
s1 = sapply(dat[!fuma,1:3],sd)
s2 = sapply(dat[fuma,1:3],sd)
rbind(nofuma=s1,fuma=s2)

##           age      fev      ht
## nofuma  2.740642 0.8505215 5.672432
## fuma    2.339255 0.7499863 3.192671
```

Apartado 4 (3 puntos)

(a) Estima e interpreta el modelo de regresión simple entre fev (variable respuesta) y ht (altura) como regresor. Realiza la diagnosis del modelo.

```
dat = read.table("fev.txt",header=TRUE)
m1 = lm(fev~ht,data=dat)
summary(m1)

##
## Call:
## lm(formula = fev ~ ht, data = dat)
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -1.75167 -0.26619 -0.00401  0.24474  2.11936 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -5.432679  0.181460 -29.94   <2e-16 ***
## ht          0.131976  0.002955  44.66   <2e-16 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## 
## Residual standard error: 0.4307 on 652 degrees of freedom
## Multiple R-squared:  0.7537, Adjusted R-squared:  0.7533 
## F-statistic: 1995 on 1 and 652 DF,  p-value: < 2.2e-16

```

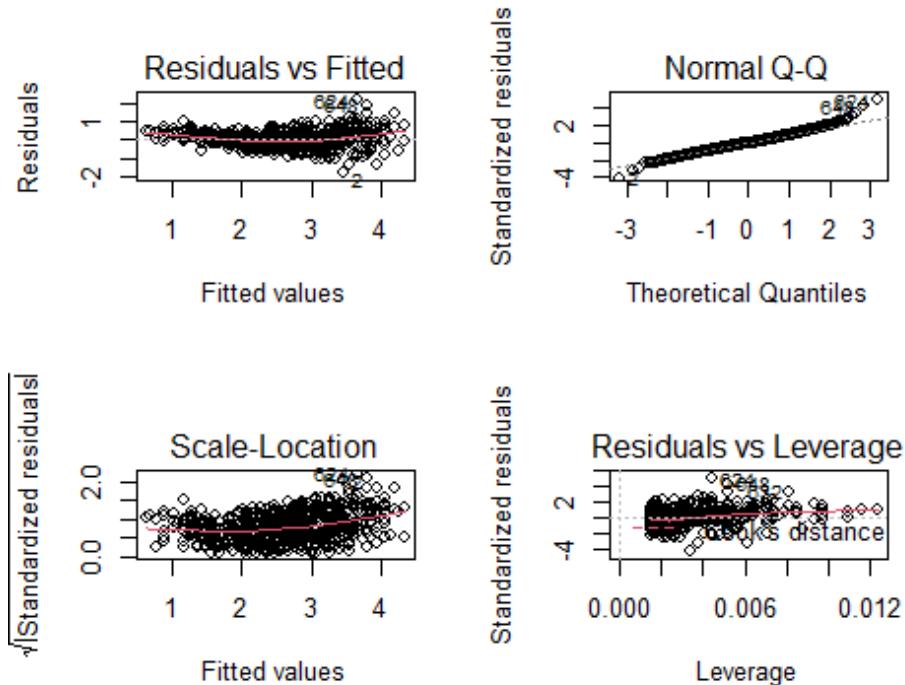
- El modelo tiene un coeficiente de determinación $R^2 = 75.4\%$. La altura explica el 75% de la variabilidad de la capacidad pulmonar.
- Con la altura predecimos la capacidad pulmonar con un error medio de 0.43 litros
- La altura tiene un efecto significativo en la capacidad pulmonar. La capacidad pulmonar aumenta 0.132 litros por cada pulgada de estatura
- Beta0 no tiene interés

Diagnosis del modelo.

```

par(mfrow=c(2,2))
plot(m1)

```



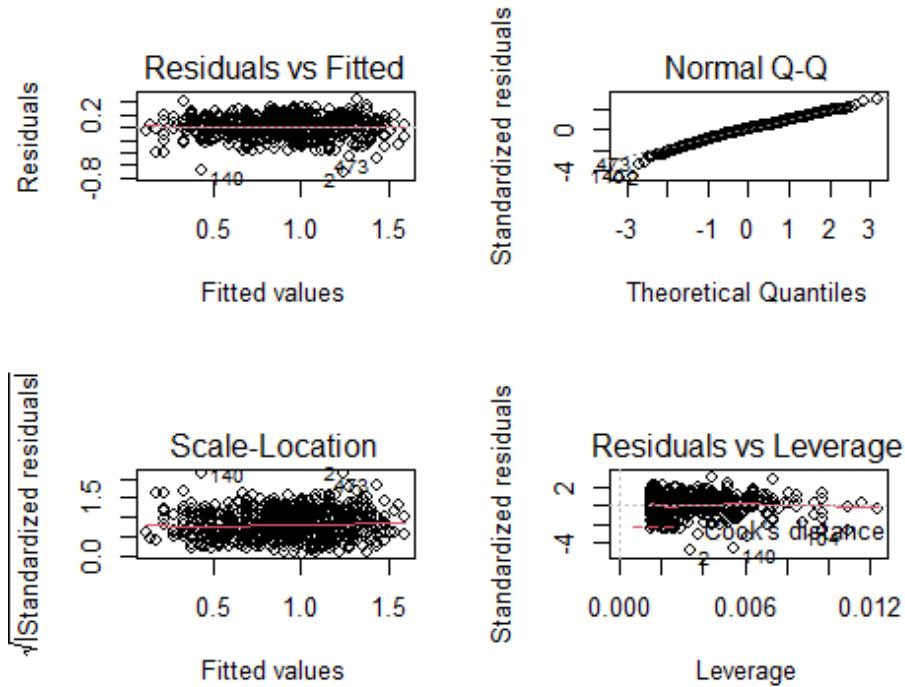
El modelo no cumple la hipótesis de linealidad ni de homocedasticidad, como se aprecia en la primera figura. Se recomienda la transformación logarítmica.

(b) Estima el modelo otra vez utilizando log(fev) como variable respuesta. Realiza la diagnosis y comenta los resultados de la diagnosis.

```
m2 = lm(log(fev)~ht,data=dat)
summary(m2)

##
## Call:
## lm(formula = log(fev) ~ ht, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.70208 -0.08986  0.01190  0.09337  0.43174 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.271312   0.063531 -35.75 <2e-16 ***
## ht          0.052119   0.001035  50.38 <2e-16 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.1508 on 652 degrees of freedom
## Multiple R-squared:  0.7956, Adjusted R-squared:  0.7953 
## F-statistic: 2538 on 1 and 652 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(m2)
```



La transformación corrige la falta de linealidad y homocedasticidad.

(c) Interpreta los coeficientes fundamentales del modelo estimado.

- El modelo tiene un coeficiente de determinación $R^2 = 79.56\%$. La altura explica el 79.6% de la variabilidad de la capacidad pulmonar en logaritmos.
- Con la altura predecimos la capacidad pulmonar con un error medio de 15%. En términos absolutos (medidos en litros), el error es más grande para capacidades pulmonares mayores.
- La altura tiene un efecto significativo en la capacidad pulmonar. La capacidad pulmonar aumenta un 5.2% por cada pulgada de altura.

Apartado 5 (3 puntos)

(a) Estima el modelo de regresión múltiple entre log(fev) y el resto de las variables.

```
m3 = lm(log(fev)~., data=dat)
summary(m3)

##
## Call:
## lm(formula = log(fev) ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.63278 -0.08657  0.01146  0.09540  0.40701 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.943998  0.078639 -24.721 < 2e-16 ***
## age          0.023387  0.003348   6.984 7.1e-12 ***
## ht           0.042796  0.001679  25.489 < 2e-16 ***
## sex          0.029319  0.011719   2.502  0.0126 *  
## smoke        -0.046068  0.020910  -2.203  0.0279 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1455 on 649 degrees of freedom
## Multiple R-squared:  0.8106, Adjusted R-squared:  0.8095 
## F-statistic: 694.6 on 4 and 649 DF,  p-value: < 2.2e-16
```

Podemos utilizar el método stepwise para seleccionar los regresores más importantes.

```
m4 = step(m3, trace=0)
summary(m4)

##
## Call:
## lm(formula = log(fev) ~ age + ht + sex + smoke, data = dat)
##
```

```

## Residuals:
##      Min      1Q   Median      3Q     Max
## -0.63278 -0.08657  0.01146  0.09540  0.40701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.943998  0.078639 -24.721 < 2e-16 ***
## age          0.023387  0.003348   6.984 7.1e-12 ***
## ht           0.042796  0.001679  25.489 < 2e-16 ***
## sex          0.029319  0.011719   2.502  0.0126 *  
## smoke        -0.046068  0.020910  -2.203  0.0279 *  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1455 on 649 degrees of freedom
## Multiple R-squared:  0.8106, Adjusted R-squared:  0.8095 
## F-statistic: 694.6 on 4 and 649 DF,  p-value: < 2.2e-16

```

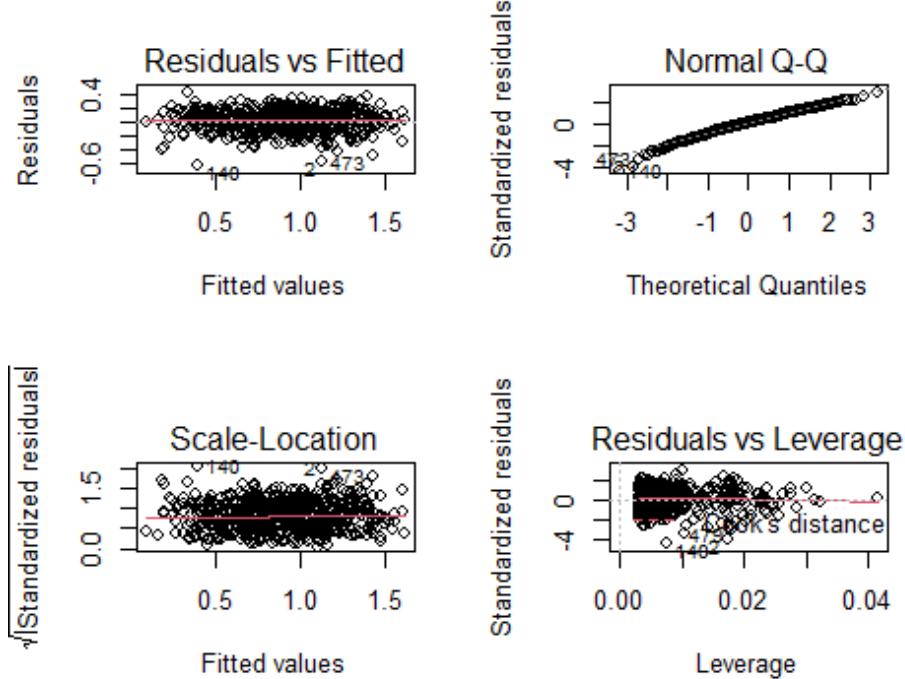
El modelo es el mismo. Todas las variables son significativas.

(b) Realiza la diagnosis del modelo.

```

par(mfrow=c(2,2))
plot(m4)

```



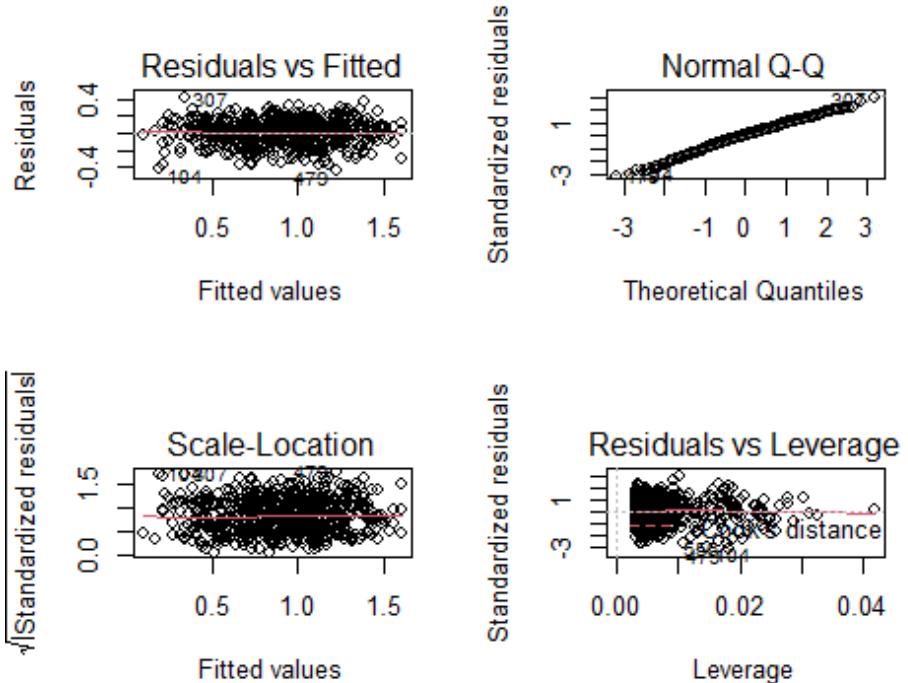
Hay algunas observaciones atípicas: 2, 140 y 473. El resto de las condiciones del modelo se cumplen razonablemente bien.

El modelo sin las observaciones atípicas no cambia sustancialmente.

```
m5 = lm(log(fev)~., data=dat[-c(2,140,473),])
summary(m5)

##
## Call:
## lm(formula = log(fev) ~ ., data = dat[-c(2, 140, 473), ])
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -0.43429 -0.08637  0.01137  0.08916  0.40989
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.981804  0.076442 -25.926 < 2e-16 ***
## age          0.021532  0.003250   6.625 7.34e-11 ***
## ht           0.043822  0.001633  26.829 < 2e-16 ***
## sex          0.022211  0.011356   1.956  0.0509 .  
## smoke        -0.048004  0.020195  -2.377  0.0177 *  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1404 on 646 degrees of freedom
## Multiple R-squared:  0.8209, Adjusted R-squared:  0.8198 
## F-statistic: 740.1 on 4 and 646 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(m5)
```



(c) Interpreta el modelo obtenido.

- **Fumar reduce significativamente la capacidad pulmonar de un joven.**
- El modelo explica el 81% de la variabilidad de la capacidad pulmonar (en logaritmos).
- Con la información de edad, altura, sexo y condición de fumador o no, se predice la capacidad pulmonar de un individuo con un error del 14.55 %.
- Todos los betas son significativos
- El hombre tiene una capacidad pulmonar de un 3% más que las mujeres a igualdad de edad, peso y de su condición de fumador o no.
- Los fumadores a igualdad del resto de las variables tienen un 4.6 % menos de capacidad pulmonar.

ANALISIS DE DATOS

Lección 3

COMPONENTES PRINCIPALES

1

Lección 3: Componentes Principales

1. ¿Cuándo se utilizan?
2. Datos Estandarizados
3. Componentes Principales (Concepto)
4. Cálculo
5. Matriz de Pesos e Importancia
6. Matriz de Cargas
7. Cuantos componentes utilizar
8. Gráficos Importantes
9. Componentes Principales con R
10. Ejemplo: Delitos USA

Objetivo

Transformar un conjunto de variables en otro conjunto sustancialmente menor que se denominan **componentes principales** y que mantienen la máxima información del conjunto original.

3

Ejemplo: EPF provincial

	PROVINCIA	CA	ALIMENTO	VESTIDO	VIVIENDA	SALUD	TRANSP.	CULTURA	Y1	Y2
1	ALAVA	País Vasco	2275	929	7781	638	1408	3524	4.306991786	-1.06325355
2	ALBACETE	Castilla - La Mancha	1253	449	2245	266	800	1182	-2.284585559	0.28191439
3	ALICANTE	Comunidad Valenciana	1512	598	3104	527	1401	2785	0.554928362	0.86528771
4	ALMERIA	Andalucía	1338	423	2218	287	818	1378	-2.104246576	0.31638767
5	AVILA	Castilla y León	1706	500	2180	256	883	1445	-1.455975004	-0.67203801
6	BADAJOZ	Extremadura	1098	366	1896	176	757	868	-3.1766335878	0.51197804
7	BALEARES	Baleares	1777	629	6211	438	2079	2802	2.303614246	0.58462114
8	BARCELONA	Cataluña	1977	649	4777	724	1909	3778	3.109370912	0.91622591
9	BURGOS	Castilla y León	1878	498	5132	266	1225	2499	0.447250546	-0.27736409
10	CACERES	Extremadura	1246	399	2157	240	1075	1104	-2.290465549	0.54282518
11	CADIZ	Andalucía	1325	429	2191	309	836	1278	-2.092872531	0.34089555
12	CASTELLÓN	Comunidad Valenciana	1676	620	3974	592	1412	2310	0.997979159	0.43729638
13	C.REAL	Castilla - La Mancha	1301	388	2663	241	582	967	-2.620892851	0.10459691
• • •										
48	VIZCAYA	País Vasco	2030	943	6406	632	2108	4130	4.563806393	0.04095601
49	ZAMORA	Castilla y León	1821	467	2628	189	833	1360	-1.509120624	-0.99614812
50	ZARAGOZA	Aragón	1879	545	4477	865	1410	3382	2.303179135	1.23246723
51	CEUTA	Ceuta	1332	498	2018	275	657	1554	-2.067855674	0.03370636
52	MELILLA	Melilla	1899	563	1857	515	569	1560	-0.813639021	-0.90919196

4

Dos componentes

Primer Componente

$$Y_{1i} = 0.375ALM_i + 0.408VES_i + 0.417VIV_i + 0.398SAL_i + 0.417TRA_i + 0.430CUL_i$$

Segundo Componente

$$Y_{1i} = 0.651ALM_i + 0.477VES_i - 0.047VIV_i - 0.358SAL_i - 0.295TRA_i - 0.361CUL_i$$

5

Cuándo interesan los CP

- Datos tienen muchas variables
- Las relaciones entre las variables es lineal (gráficos de dispersión)
- La correlación entre las variables es alta. (Matriz de correlaciones)

6

Datos Estandarizados

PROVINC	ALIMENT	Z1
1 ALAVA	2275	1.886406941
2 ALBACETE	1253	-1.347191574
3 ALICANTE	1512	-0.527717471
4 ALMERIA	1338	-1.078252196
5 AVILA	1706	0.086097108
6 BADAJOZ	1098	-1.837610439
7 BALEARES	1777	0.310740588
8 BARCELONA	1977	0.943539124
9 BURGOS	1878	0.630303849
10 CACERES	1246	-1.369339523
11 CADIZ	1325	-1.119384101
12 CASTELLON	1676	-0.006822672
13 C.REAL	1301	-1.195319925
• • •		
48 VIZCAYA	2030	1.111230735
49 ZAMORA	1821	0.449956266
50 ZARAGOZA	1879	0.633467841
51 CEUTA	1332	-1.097236152
52 MELILLA	1899	0.696747695

$$x_1, x_2, \dots, x_n \rightarrow \bar{x}, \hat{s}$$

$$z_i = \frac{x_i - \bar{x}}{\hat{s}}$$

$$\bar{z} = 0$$

$$\hat{s}_z = 1$$

7

Variables Estandarizadas

$$\begin{array}{cccc}
 \overline{X_1} & \overline{X_2} & \cdots & \overline{X_k} \\
 \hline
 x_{11} & x_{21} & \cdots & x_{k1} \\
 x_{12} & x_{22} & \cdots & x_{k2} \\
 \vdots & \vdots & \ddots & \vdots \\
 x_{1n} & x_{2k} & \cdots & x_{kn}
 \end{array}
 \xrightarrow{\hspace{1cm}}
 \begin{array}{cccc}
 \overline{Z_1} & \overline{Z_2} & \cdots & \overline{Z_k} \\
 \hline
 z_{11} & z_{21} & \cdots & z_{k1} \\
 z_{12} & z_{22} & \cdots & z_{k2} \\
 \vdots & \vdots & \ddots & \vdots \\
 z_{1n} & z_{2k} & \cdots & z_{kn}
 \end{array}$$

$$z_{ij} = \frac{x_{ij} - \bar{x}_i}{\hat{s}_i}$$

8

Primer componente

$$Y_{1j} = a_{11}Z_{1j} + a_{21}Z_{2j} + \cdots + a_{k1}Z_{kj}$$

$$\left. \begin{array}{l} \text{Corr}(Y_1, Z_1) = \lambda_{11} \\ \text{Corr}(Y_1, Z_2) = \lambda_{21} \\ \vdots \\ \text{Corr}(Y_1, Z_k) = \lambda_{k1} \end{array} \right\} \quad \lambda_1 = \lambda_{11}^2 + \lambda_{21}^2 + \cdots + \lambda_{k1}^2$$

$$\lambda_{i1} = a_{i1} \sqrt{\lambda_1}$$

No existe otra variable que tenga una mayor correlación (λ_1) con las Z_i

9

Segundo componente

$$Y_{2j} = a_{12}Z_{1j} + a_{22}Z_{2j} + \cdots + a_{k2}Z_{kj}$$

$$\left. \begin{array}{l} \text{Corr}(Y_2, Z_1) = \lambda_{12} \\ \text{Corr}(Y_2, Z_2) = \lambda_{22} \\ \vdots \\ \text{Corr}(Y_2, Z_k) = \lambda_{k2} \end{array} \right\} \quad \lambda_2 = \lambda_{12}^2 + \lambda_{22}^2 + \cdots + \lambda_{k2}^2$$

$$\lambda_{i2} = a_{i2} \sqrt{\lambda_2}$$

- Tiene correlación máxima con las Z_i
- Está incorrelacionada con Y_1

10

... y así sucesivamente hasta k .

$$Y_{mj} = a_{1m}Z_{1j} + a_{2m}Z_{2j} + \cdots + a_{km}Z_{kj}$$

$$\left. \begin{array}{l} \text{Corr}(Y_m, Z_1) = \lambda_{1m} \\ \text{Corr}(Y_m, Z_2) = \lambda_{2m} \\ \vdots \\ \text{Corr}(Y_m, Z_k) = \lambda_{km} \end{array} \right\} \quad \lambda_m = \lambda_{1m}^2 + \lambda_{2m}^2 + \cdots + \lambda_{km}^2$$

$$\lambda_{im} = a_{im} \sqrt{\lambda_m}$$

- Tiene correlación máxima con las Z_i
- Está incorrelacionada con Y_1, Y_2, \dots, Y_{m-1}

11

Cálculo: $R = P \Lambda P^T$

Valores y vectores propios

$$\begin{pmatrix} 1 & r_{12} & r_{13} & \cdots & r_{1k} \\ r_{12} & 1 & r_{23} & \cdots & r_{2k} \\ r_{13} & r_{23} & 1 & \cdots & r_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{1k} & r_{2k} & r_{3k} & \cdots & 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kk} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_k \end{pmatrix} \begin{pmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{k1} \\ a_{12} & a_{22} & a_{32} & \cdots & a_{k2} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{k3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1k} & a_{2k} & a_{3k} & \cdots & a_{kk} \end{pmatrix}$$

$$P = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kk} \end{pmatrix} \quad \Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_k \end{pmatrix}$$

Vectores Propios (columnas)

Valores Propios

12

Matriz de Pesos

		Nuevas Variables			
		Y_1	Y_2	\dots	Y_k
Variables Originales	Z_1	a_{11}	a_{12}	\dots	a_{1k}
	Z_2	a_{21}	a_{22}	\dots	a_{2k}
	\vdots	\vdots	\vdots	\ddots	\vdots
	Z_k	a_{k1}	a_{k2}	\dots	a_{kk}

$Y_{1i} = a_{11}Z_{1i} + a_{21}Z_{2i} + \dots + a_{k1}Z_{ki}$

Se pueden obtener **k** variables nuevas, tantas como variables originales.

13

Matriz de pesos

	Y1	Y2	Y3	Y4	Y5	Y6
ALIMENT	0.375	0.651	0.323	0.076	0.557	0.121
VESTIDO	0.408	0.477	-0.117	-0.230	-0.734	0.003
VIVIENDA	0.417	-0.041	-0.613	0.524	0.137	-0.394
SALUD	0.398	-0.358	0.685	0.185	-0.181	-0.421
TRANSP	0.418	-0.295	-0.193	-0.765	0.310	-0.137
CULTURA	0.431	-0.361	-0.023	0.215	-0.056	0.796

$$Y_{1i} = 0.375ALM_i + 0.408VES_i + 0.417VIV_i + 0.398SAL_i + 0.417TRA_i + 0.430CUL_i$$

$$Y_{2i} = 0.651ALM_i + 0.477VES_i - 0.041VIV_i - 0.358SAL_i - 0.295TRA_i - 0.361CUL_i$$

14

Matriz de Pesos

		Nuevas Variables			
		Y_1	Y_2	\dots	Y_k
Variables Originales	Z_1	a_{11}	a_{12}	\dots	a_{1k}
	Z_2	a_{21}	a_{22}	\dots	a_{2k}
	\vdots	\vdots	\vdots	\ddots	\vdots
	Z_k	a_{k1}	a_{k2}	\dots	a_{kk}

$$a_{1i}^2 + a_{2i}^2 + \dots + a_{ki}^2 = 1$$

Ortogonales

15

Varianza de los componentes

$$\lambda_i = \text{Var}(Y_i) = \frac{\sum_{j=1}^n (Y_{ij} - \bar{Y}_i)^2}{n} = \frac{\sum_{j=1}^n Y_{ij}^2}{n}$$

$$\text{Var}(Z_1) + \text{Var}(Z_2) + \dots + \text{Var}(Z_k) = k$$

$$\text{Var}(Y_1) + \text{Var}(Y_2) + \dots + \text{Var}(Y_k) = k$$

$$\text{Var}(Y_1) > \text{Var}(Y_2) > \dots > \text{Var}(Y_k)$$

Los componentes están ordenados según su importancia

16

Porcentaje de varianza

	Y1	Y2	Y3	Y4	Y5	Y6
ALIMENT	0.375	0.651	0.323	0.076	0.557	0.121
VESTIDO	0.408	0.477	-0.117	-0.230	-0.734	0.003
VIVIENDA	0.417	-0.041	-0.613	0.524	0.137	-0.394
SALUD	0.398	-0.358	0.685	0.185	-0.181	-0.421
TRANSP	0.418	-0.295	-0.193	-0.765	0.310	-0.137
CULTURA	0.431	-0.361	-0.023	0.215	-0.056	0.796
Varianza	4.468	0.656	0.368	0.223	0.167	0.118
% Varianza	74.472	10.931	6.140	3.709	2.784	1.963
% Var. Acum.	74.472	85.403	91.544	95.253	98.037	100.000

$$\frac{\lambda_i}{k} \times 100 \quad Ejem.: \frac{4.468}{6} \times 100 = 74.4\%$$

17

Matriz de Cargas

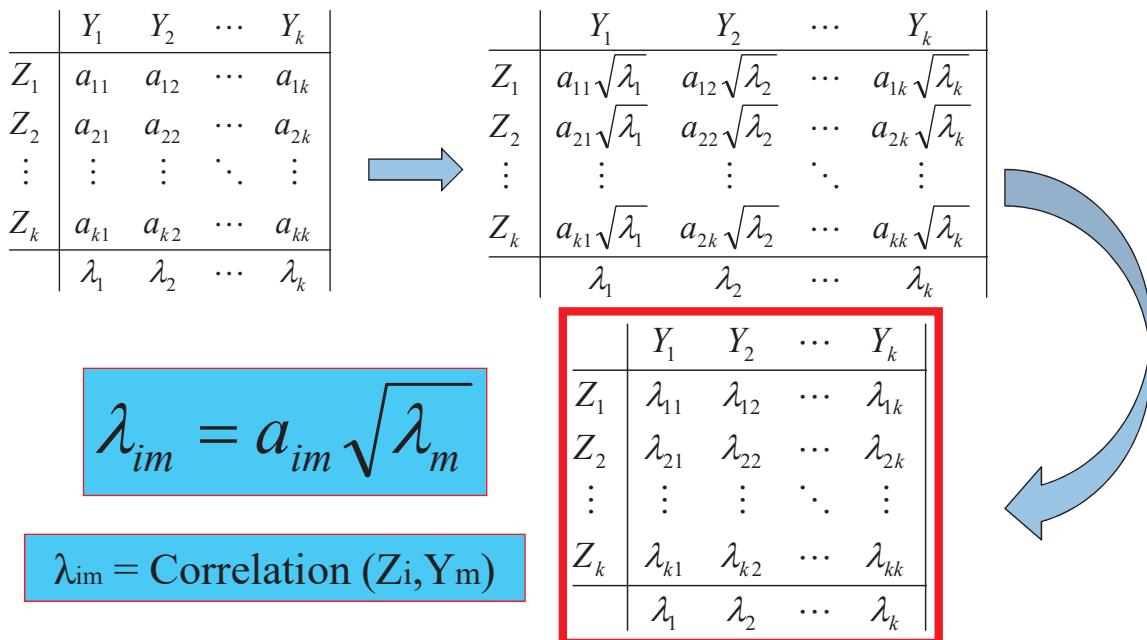
		Nuevas Variables			
		Y_1	Y_2	\dots	Y_k
Variables Originales	Z_1	λ_{11}	λ_{12}	\dots	λ_{1k}
	Z_2	λ_{21}	λ_{22}	\dots	λ_{2k}
	\vdots	\vdots	\vdots	\ddots	\vdots
	Z_k	λ_{k1}	λ_{k2}	\dots	λ_{kk}
		λ_1	λ_2	\dots	λ_k

$$\lambda_{ij} = \text{Corr}(Z_i, Y_j)$$

$$\lambda_i = \lambda_{1i}^2 + \lambda_{2i}^2 + \dots + \lambda_{ki}^2$$

18

Matriz de Cargas



19

Matriz de cargas (loadings)

	Y1	Y2	Y3	Y4	Y5	Y6
ALIMENT	0.793	0.527	0.196	0.036	0.228	0.041
VESTIDO	0.862	0.387	-0.071	-0.109	-0.300	0.001
VIVIENDA	0.882	-0.033	-0.372	0.247	0.056	-0.135
SALUD	0.842	-0.290	0.416	0.087	-0.074	-0.144
TRANSP	0.883	-0.239	-0.117	-0.361	0.127	-0.047
CULTURA	0.910	-0.293	-0.014	0.101	-0.023	0.273
Varianza	4.468	0.656	0.368	0.223	0.167	0.118
% Varianza	74.472	10.931	6.140	3.709	2.784	1.963
% Var. Acum.	74.472	85.403	91.544	95.253	98.037	100.000

1º

2º

3º

4º

5º

6º

20

Dos componentes principales

	Y1	Y2	Y3	Y4	Y5	Y6
ALIMENT	0.375	0.651	0.323	-0.076	0.557	0.121
VESTIDO	0.408	0.477	-0.117	-0.259	-0.734	0.003
VIVIENDA	0.417	-0.041	-0.613	0.524	0.137	-0.394
SALUD	0.398	-0.358	0.685	0.185	-0.181	-0.421
TRANSP	0.418	-0.295	-0.193	-0.765	0.310	-0.137
CULTURA	0.431	-0.361	-0.023	0.215	-0.056	0.796

$$Y_{1i} = 0.375ALM_i + 0.408VES_i + 0.417VIV_i + 0.398SAL_i + 0.417TRA_i + 0.430CUL_i$$

$$Y_{2i} = 0.651ALM_i + 0.477VES_i - 0.041VIV_i - 0.358SAL_i - 0.295TRA_i - 0.361CUL_i$$

21

Dos componentes principales

$$ALI_i = 0.375Y_{1i} + 0.651Y_{2i} + e_{ALIi}$$

$$VES_i = 0.408Y_{1i} + 0.477Y_{2i} + e_{VESi}$$

$$VIV_i = 0.417Y_{1i} - 0.041Y_{2i} + e_{VIVi}$$

$$SAL_i = 0.398Y_{1i} - 0.358Y_{2i} + e_{SALi}$$

$$TRA_i = 0.418Y_{1i} - 0.295Y_{2i} + e_{TRAi}$$

$$CUL_i = 0.430Y_{1i} - 0.361Y_{2i} + e_{CULi}$$

22

Interpretación 2 componentes

	Comp 1	Comp 2	communality	uniqueness
ALIMENT	0.793	0.527	0.907	0.093
VESTIDO	0.862	0.387	0.893	0.107
VIVIENDA	0.882	-0.033	0.779	0.221
SALUD	0.842	-0.290	0.793	0.207
TRANSP	0.883	-0.239	0.838	0.162
CULTURA	0.910	-0.293	0.914	0.086

Variance	4.468	0.656
Proportion	0.745	0.109
Cumulative	0.745	0.854

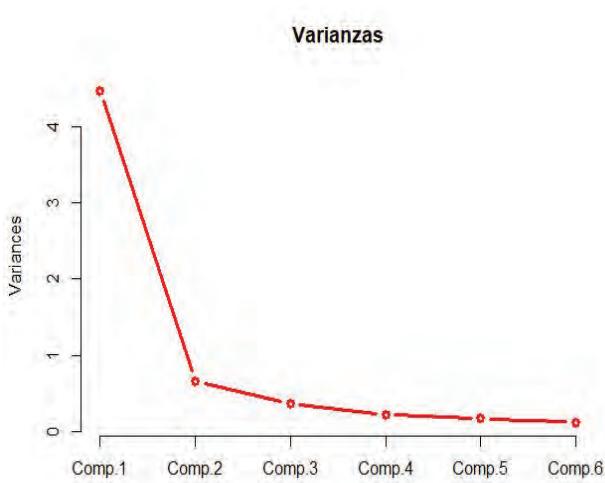
Comunalidad
(parte común)

Especificidad
(parte específica)

Reducimos de 6 a 2 y conseguimos retener el 85.4% de la información de los 6

23

¿Cuántos componentes?



- Cuantos menos mejor, cuanta más información mejor.
- KEISER (1961)
 $\lambda_i > 1$
- JOLLIFFE (1977)
 $\lambda_i > 0.7$
- CATTELL (1966)
(Gráfico sedimentación)

24

Lección 3 : Componentes Principales

8. GRÁFICOS IMPORTANTES

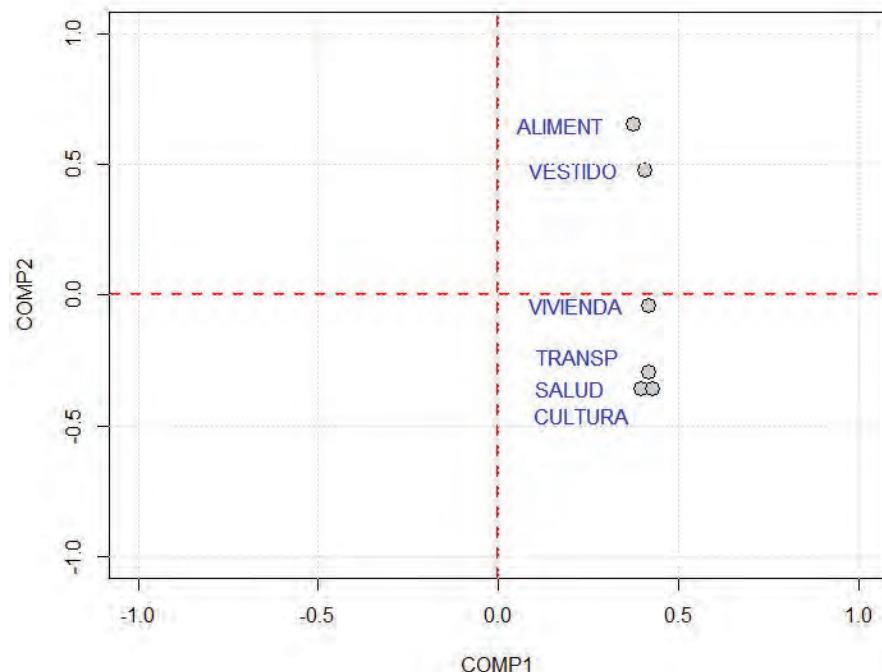
Scores (Y_{1i}, Y_{2i})

	Y1	Y2
ALIMENT	0.375	-0.651
VESTIDO	0.408	-0.477
VIVIENDA	0.417	0.041
SALUD	0.398	0.358
TRANSP	0.418	0.295
CULTURA	0.431	0.361

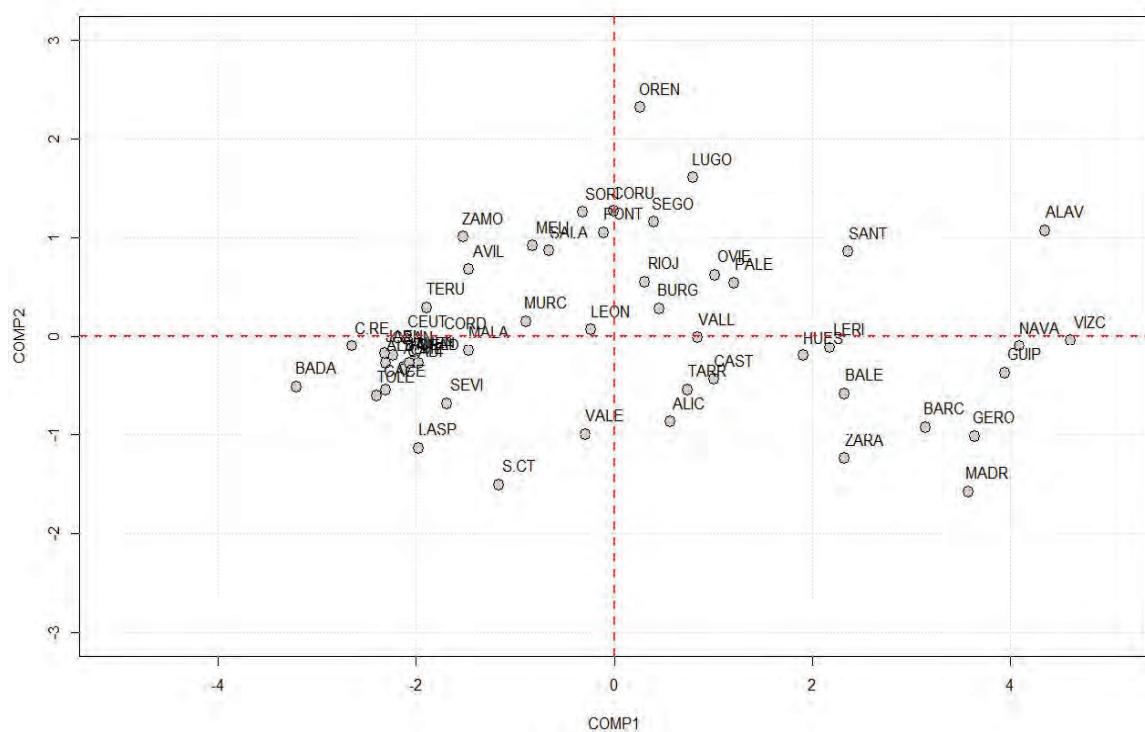
$$Y_{1i} = 0.375ALM_i + 0.408VES_i + 0.417VIV_i + 0.398SAL_i + 0.417TRA_i + 0.430CUL_i$$

$$Y_{2i} = 0.651ALM_i + 0.477VES_i - 0.047VIV_i - 0.358SAL_i - 0.295TRA_i - 0.361CUL_i$$

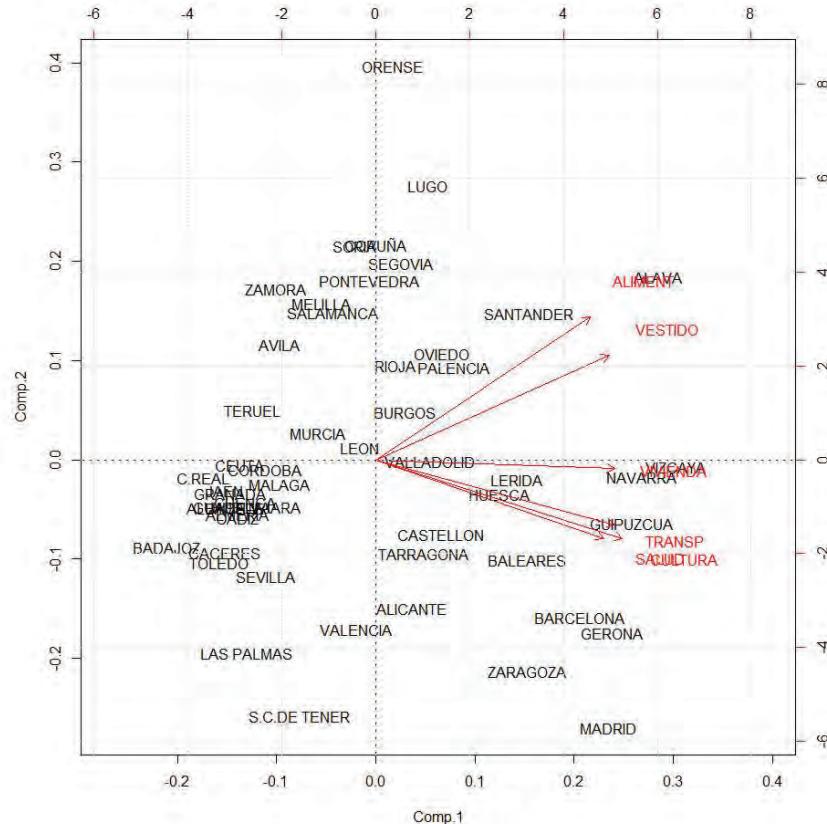
Pesos/Cargas



Scores (Y_{1i}, Y_{2i})



biplot



Lección 3 : Componentes Principales

8. EJEMPLO CON R



Datos y Correlaciones

```
renta0 = read.csv('gastos.csv', header=T)
renta2 = renta0[, 3:8]
row.names(renta2)=renta0$PROVINC
head(renta0)
```

```
##   PROVINC          CA ALIMENT VESTIDO VIVIENDA SALUD TRANSP CULTURA
## 1    ALAVA      País Vasco  2275     929    7781   638  1408  3524
## 2  ALBACETE Castilla - La Mancha 1253     449   2245   266   800 1182
## 3  ALICANTE Comunidad Valenciana 1512     598   3104   527  1401 2785
## 4   ALMERIA      Andalucía 1338     423   2218   287   818 1378
## 5    AVILA      Castilla y León 1706     500   2180   256   883 1445
## 6  BADAJOZ      Extremadura 1098     366   1896   176   757  868
```



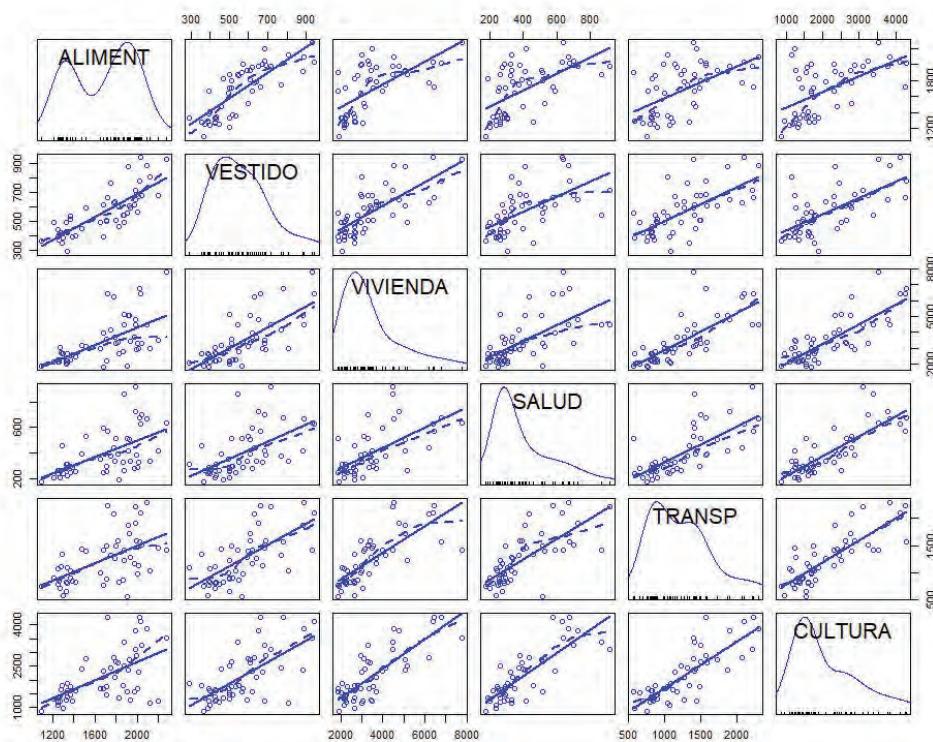
Correlaciones

```
cor(renta2)
```

```
##          ALIMENT VESTIDO VIVIENDA SALUD TRANSP CULTURA
## ALIMENT    1.000    0.801    0.625  0.576  0.565  0.574
## VESTIDO    0.801    1.000    0.730  0.597  0.679  0.669
## VIVIENDA   0.625    0.730    1.000  0.635  0.755  0.805
## SALUD      0.576    0.597    0.635  1.000  0.730  0.817
## TRANSP     0.565    0.679    0.755  0.730  1.000  0.823
## CULTURA    0.574    0.669    0.805  0.817  0.823  1.000
```

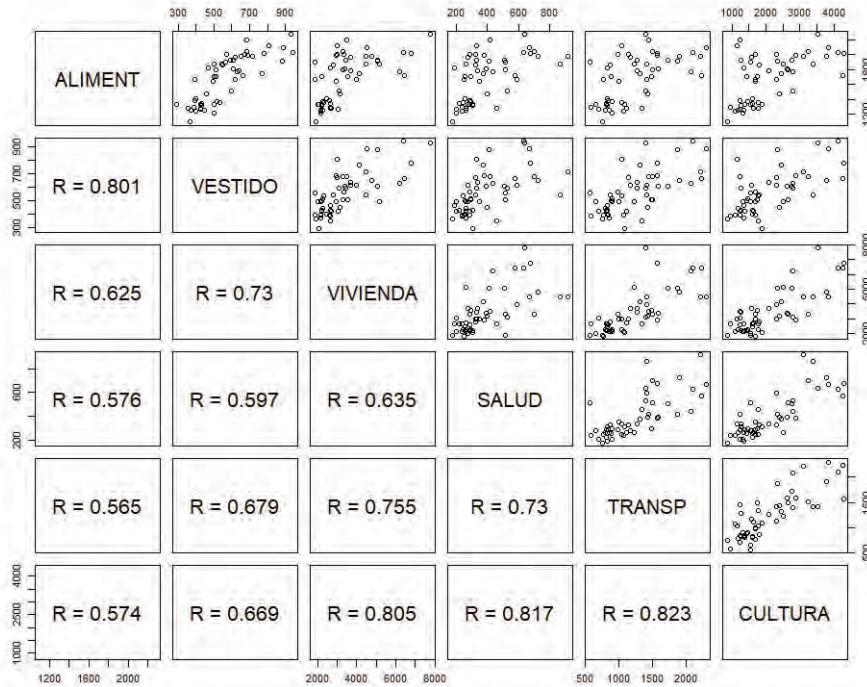
Gráficos de dispersión

```
library('car')
scatterplotMatrix(renta2, regLine=TRUE,
smooth=list(spread=FALSE))
```





pairs(renta2)



Componentes Principales

```
fit = princomp(renta2, cor = TRUE)
summary(fit)
```

```
## Importance of components:
##                               Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
## Standard deviation     2.1138449 0.8098562 0.60698331 0.47173282 0.40872965
## Proportion of Variance 0.7447234 0.1093112 0.06140479 0.03708864 0.02784332
## Cumulative Proportion  0.7447234 0.8540346 0.91543934 0.95252798 0.98037131
##                               Comp.6
## Standard deviation     0.34317950
## Proportion of Variance 0.01962869
## Cumulative Proportion  1.00000000
```



Pesos o Weights (loadings no es habitual)

```
loadings (fit)
```

```
##  
## Loadings:  
##          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6  
## ALIMENT    0.375  0.651  0.323           0.557  0.121  
## VESTIDO    0.408  0.477 -0.117 -0.230 -0.734  
## VIVIENDA   0.417           -0.613  0.524  0.137 -0.394  
## SALUD      0.398 -0.358  0.685  0.185 -0.181 -0.421  
## TRANSP     0.418 -0.295 -0.193 -0.765  0.310 -0.137  
## CULTURA    0.431 -0.361           0.215           0.796  
##  
##          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6  
## SS loadings  1.000  1.000  1.000  1.000  1.000  1.000  
## Proportion Var 0.167  0.167  0.167  0.167  0.167  0.167  
## Cumulative Var 0.167  0.333  0.500  0.667  0.833  1.000
```



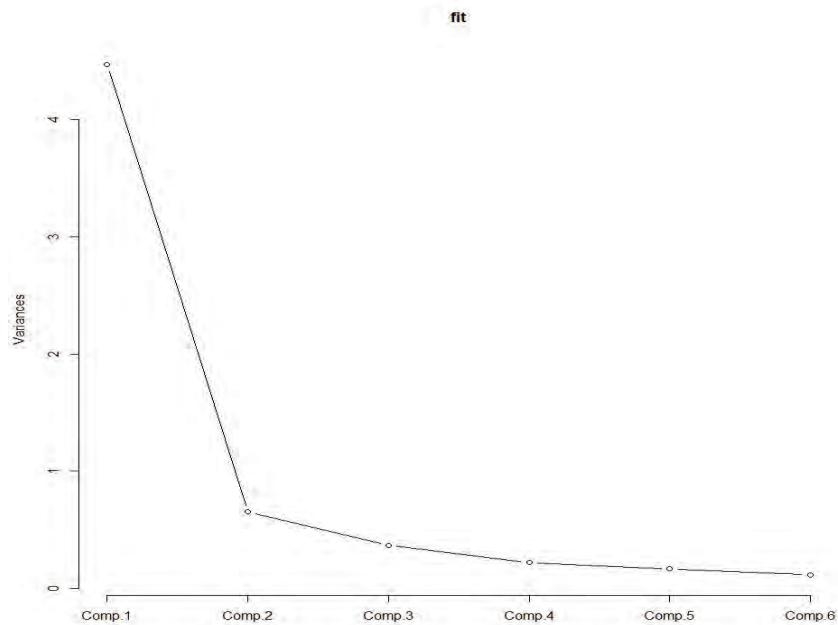
Matriz de pesos

```
unclass (loadings (fit))
```

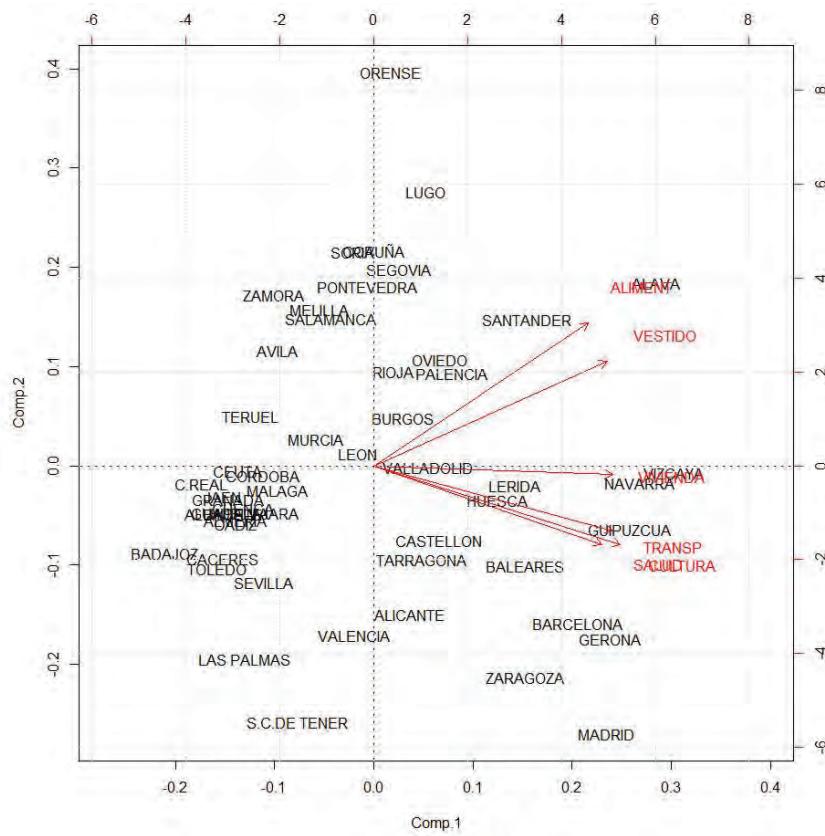
```
##          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6  
## ALIMENT    0.375  0.6513  0.3233  0.0759  0.5571  0.12069  
## VESTIDO    0.408  0.4774 -0.1169 -0.2300 -0.7342  0.00323  
## VIVIENDA   0.417 -0.0412 -0.6126  0.5243  0.1367 -0.39419  
## SALUD      0.398 -0.3584  0.6846  0.1853 -0.1806 -0.42095  
## TRANSP     0.418 -0.2954 -0.1931 -0.7654  0.3101 -0.13712  
## CULTURA    0.431 -0.3613 -0.0231  0.2151 -0.0559  0.79626
```



Número de componentes



biplot (fit)





```
prinfact <- function(x,r=2)
{
  k   <- ncol(x) # Número de variables
  comp=c('Comp 1'); for (i in 2:r) comp = c(comp,paste('Comp', i))
  nam <- c(comp,'communality','uniqueness')
  sol <- matrix(0,k,r+2) # matriz de correlaciones
  res <- matrix(0,3,r) # Varianza explicada
  rownames(sol)<-colnames(x) # Nombre de las variables

  # Cálculo

  eig <- eigen(cor(x))
  fit = princomp(x,cor=TRUE)
  eig$vectors = unclass(loadings(fit))
  imp <- eig$values/sum(eig$values)
  cum <- cumsum(imp)

  if ( eig$vectors[1,1] < 0 ) {eig$vectors = -eig$vectors}
  fac <- eig$vectors%*%diag(sqrt(eig$values))
  sol[1:k,1:r] <- fac[,1:r];
  com <- diag(fac[,1:r]%*%t(fac[,1:r]))
  uni <- 1- com

  z = scale(x) # estandariza las variables
  z_scores = z%*%eig$vectors # calcula los scores

  sol[1:k,r+1] = t(com)
  sol[1:k,r+2] = t(uni)
  res[1,1:r] = eig$values[1:r]
  res[2,1:r] = imp[1:r]
  res[3,1:r] = cum[1:r]
  colnames(sol)<-nam
  rownames(res)=c('Variance','Proportion','Cumulative')
  colnames(res)=comp
  list(loadings = sol,variances = res)
}
```

Resumen Componentes Principales



```
source('prinfact.R')
sol = prinfact(renta2,2)
sol$loadings
##          Comp 1 Comp 2 communality uniqueness
## ALIMENT    0.793  0.527      0.907     0.093
## VESTIDO    0.862  0.387      0.893     0.107
## VIVIENDA   0.882 -0.033      0.779     0.221
## SALUD      0.842 -0.290      0.793     0.207
## TRANSP     0.883 -0.239      0.838     0.162
## CULTURA    0.910 -0.293      0.914     0.086
sol$variances
##          Comp 1 Comp 2
## Variance    4.468  0.656
## Proportion  0.745  0.109
## Cumulative  0.745  0.854
```



Scores o Comp. Principales

```
row.names(sol$scores) = renta0$PROVINC
head(sol$scores)

##          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## ALAVA      4.309  1.063 -0.723  1.526 -0.462 -0.391
## ALBACETE   -2.285 -0.282 -0.119  0.022 -0.440 -0.204
## ALICANTE    0.555 -0.865  0.370 -0.185 -0.552  0.244
## ALMERIA    -2.104 -0.316  0.067  0.108 -0.192 -0.050
## AVILA       -1.456  0.672  0.254 -0.058  0.163  0.214
## BADAJOZ    -3.177 -0.512 -0.384 -0.117 -0.276 -0.213
```



Resumen Componentes Principales (3)

```
sol = prinfact(renta2, 3)
sol$loadings

##          Comp.1 Comp.2 Comp.3 communality uniqueness
## ALIMENT     0.793  0.527  0.196      0.945      0.055
## VESTIDO     0.862  0.387 -0.071      0.898      0.102
## VIVIENDA    0.882 -0.033 -0.372      0.917      0.083
## SALUD       0.842 -0.290  0.416      0.966      0.034
## TRANSP      0.883 -0.239 -0.117      0.851      0.149
## CULTURA     0.910 -0.293 -0.014      0.915      0.085
sol$variances

##          Comp.1 Comp.2 Comp.3
## Variance    4.468  0.656  0.368
## Proportion  0.745  0.109  0.061
## Cumulative 0.745  0.854  0.915
```

Comandos para Componente Principales



```
# Análisis de Componentes Principales
# datos utilizados RENTA2
# datos estandarizados (matriz de correlaciones)
fit <- princomp(renta2, cor=TRUE) # si ponemos
cor = FALSE, trabaja con la matriz de varianzas
summary(fit) # resumen
loadings(fit) # cargas /pesos
fit$scores # Los componentes principales
plot(fit,type="lines") # scree plot
biplot(fit) # gráfico de componentes
prinfact(renta2,2) # Resumen de análisis de CP
```

Lección 3 : Componentes Principales

9. EJEMPLO: DELITOS USA

EXERCISE:

'Crime rates per 100,000 pop by state'
(file: Usacrime.txt)

STATE	MURDER	RAPE	ROBBLA	ASSAU	BURGLA	LARCEN	AUTO
ALABAMA	14.2	25.2	96.8	278.3	1135.5	1881.9	280.7
ALASKA	10.8	51.6	96.8	284.0	1331.7	3369.8	753.3
ARIZONA	9.5	34.2	138.2	312.3	2346.1	4467.4	439.5
ARKANSAS	8.8	27.6	83.2	203.4	972.6	1862.1	183.4
CALIFORNIA	11.5	49.4	287.0	358.0	2139.4	3499.8	663.5
COLORADO	6.3	42.0	170.7	292.9	1935.2	3903.2	477.1
CONNECTICUT	4.2	16.8	129.5	131.8	1346.0	2620.7	593.2
DELAWARE	6.0	24.9	157.0	194.2	1682.6	3678.4	467.0
FLORIDA	10.2	39.6	187.9	449.1	1859.9	3840.5	351.4
GEORGIA	11.7	31.1	140.5	256.5	1351.1	2170.2	297.9
HAWAII	7.2	25.5	128.0	64.1	1911.5	3920.4	489.4
IDAHO	5.5	19.4	39.6	172.5	1050.8	2599.6	237.6
ILLINOIS	9.9	21.8	211.3	209.0	1085.0	2828.5	528.6
.
.
.

47



Burglary



Auto_theft



Larceny

Property crimes

Murder

Rape



Assault



Robbery

Violent Crimes

48

Correlation Matrix

	MURDER	RAPE	ROBBERY	ASSAULT	BURGLARY	LARCENY	AUTO
MURDER	1.0000	0.6012	0.4837	0.6486	0.3858	0.1019	0.0688
RAPE	0.6012	1.0000	0.5919	0.7403	0.7121	0.6140	0.3489
ROBBERY	0.4837	0.5919	1.0000	0.5571	0.6372	0.4467	0.5907
ASSAULT	0.6486	0.7403	0.5571	1.0000	0.6229	0.4044	0.2758
BURGLARY	0.3858	0.7121	0.6372	0.6229	1.0000	0.7921	0.5580
LARCENY	0.1019	0.6140	0.4467	0.4044	0.7921	1.0000	0.4442
AUTO	0.0688	0.3489	0.5907	0.2758	0.5580	0.4442	1.0000

If variables are not correlated, there would be no point in doing PCA.

The correlation matrix is symmetric, so we only need to inspect either the upper or lower triangular matrix.

49

Variances (Eigenvalues)

	Eigenvalue	Proportion	Cumulative	
PRIN1	4.11496	0.587851	0.58785	
PRIN2	1.23872	0.176960	0.76481	{ 76.5% }
PRIN3	0.72582	0.103688	0.86850	
PRIN4	0.31643	0.045205	0.91370	
PRIN5	0.25797	0.036853	0.95056	
PRIN6	0.22204	0.031720	0.98228	
PRIN7	0.12406	0.017722	1.00000	

50

Weights (Eigenvectors)

	PRIN1	PRIN2	PRIN3	PRIN4	PRIN5	PRIN6	PRIN7
MURDER	0.3002	-0.6291	0.1782	-0.2321	0.5381	0.2591	0.2675
RAPE	0.4317	-0.1694	-0.2441	0.0622	0.1884	-0.7732	-0.2964
ROBBERY	0.3968	0.0422	0.4958	-0.5579	-0.5199	-0.1143	-0.0039
ASSAULT	0.3966	-0.3435	-0.0695	0.6298	-0.5066	0.1723	0.1917
BURGLARY	0.4401	0.2033	-0.2098	-0.0575	0.1010	0.5359	-0.6481
LARCENY	0.3573	0.4023	-0.5392	-0.2348	0.0300	0.0394	0.6016
AUTO	0.2951	0.5024	0.5683	0.4192	0.3697	-0.0572	0.1470

- Do these eigenvectors mean anything?

- All crimes are positively correlated with the first component, which is therefore interpreted as a measure of overall crime rate.
- The 2nd component has positive loadings on AUTO, LARCENY and ROBBERY and negative loadings on MURDER, ASSAULT and RAPE. It is interpreted to measure the preponderance of property crime over violent crime.....

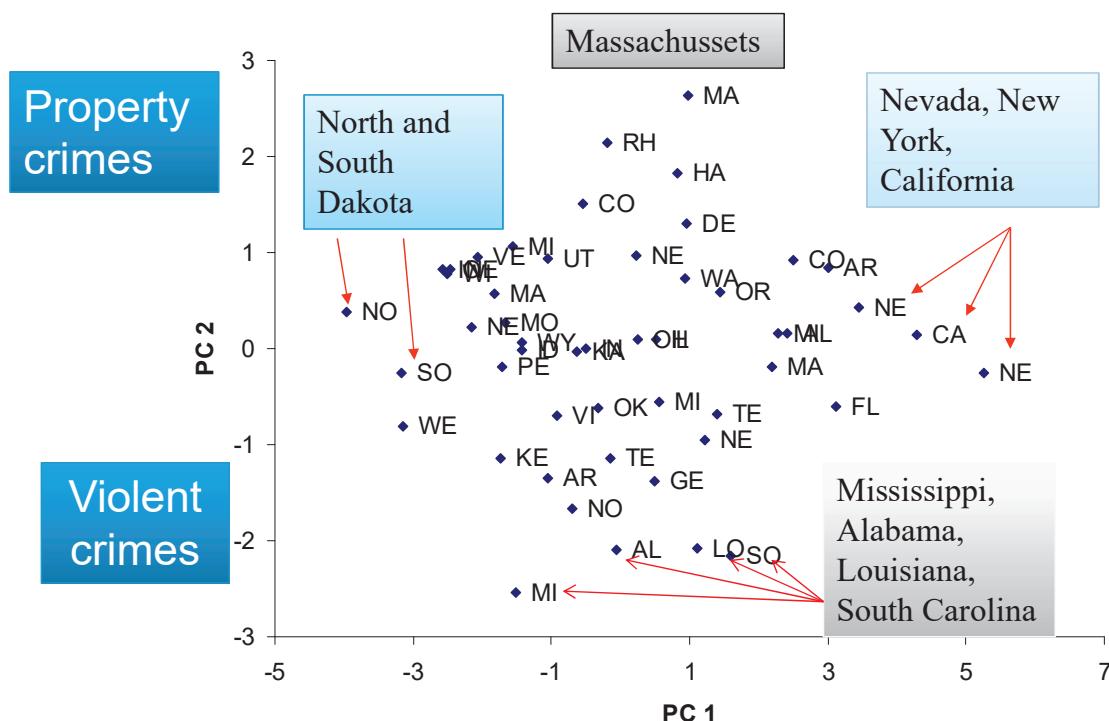
51

Resumen (Cargas y Comunalidad)

	Y1	Y2	Explicado	Residual
MURDER	0,609	-0,701	0,861	0,139
RAPE	0,875	-0,189	0,802	0,198
ROBBERY	0,804	0,047	0,649	0,351
ASSAULT	0,804	-0,383	0,793	0,207
BURGLARY	0,892	0,226	0,847	0,153
LARCENY	0,724	0,448	0,725	0,275
AUTO-THEFT	0,598	0,559	0,671	0,329
	4,11	1,24	0,764	0,236

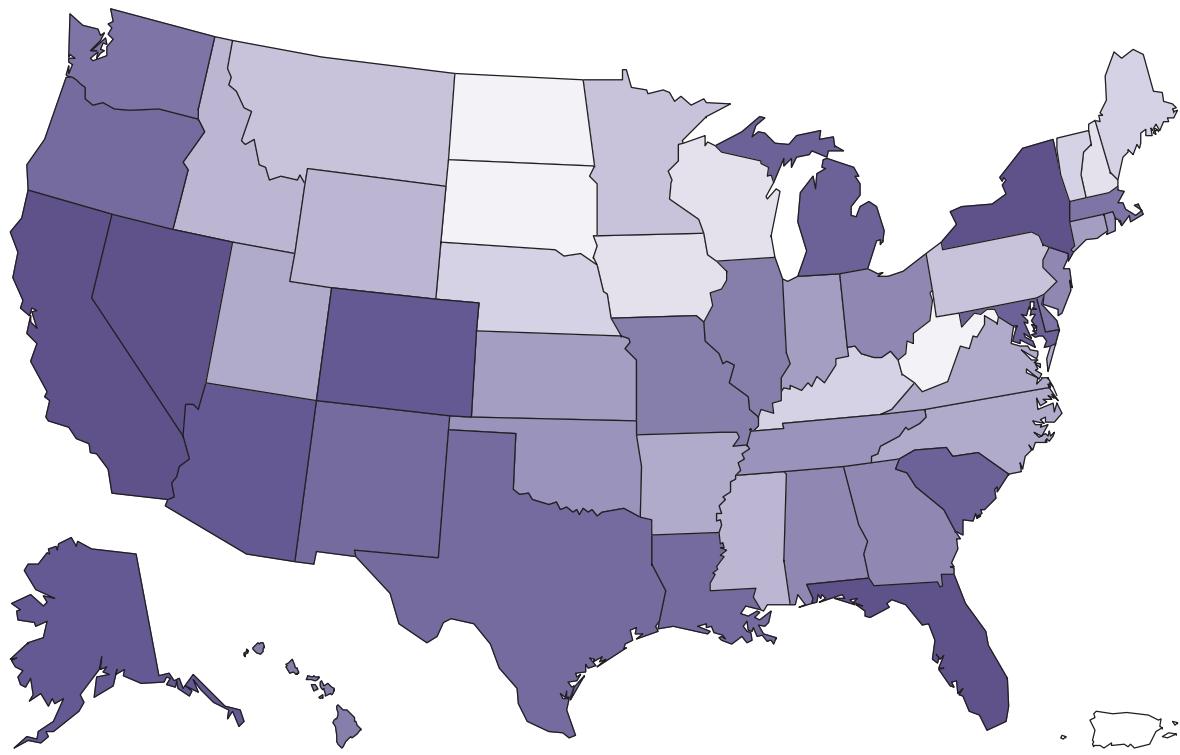
52

PC Plot: Crime Data



53

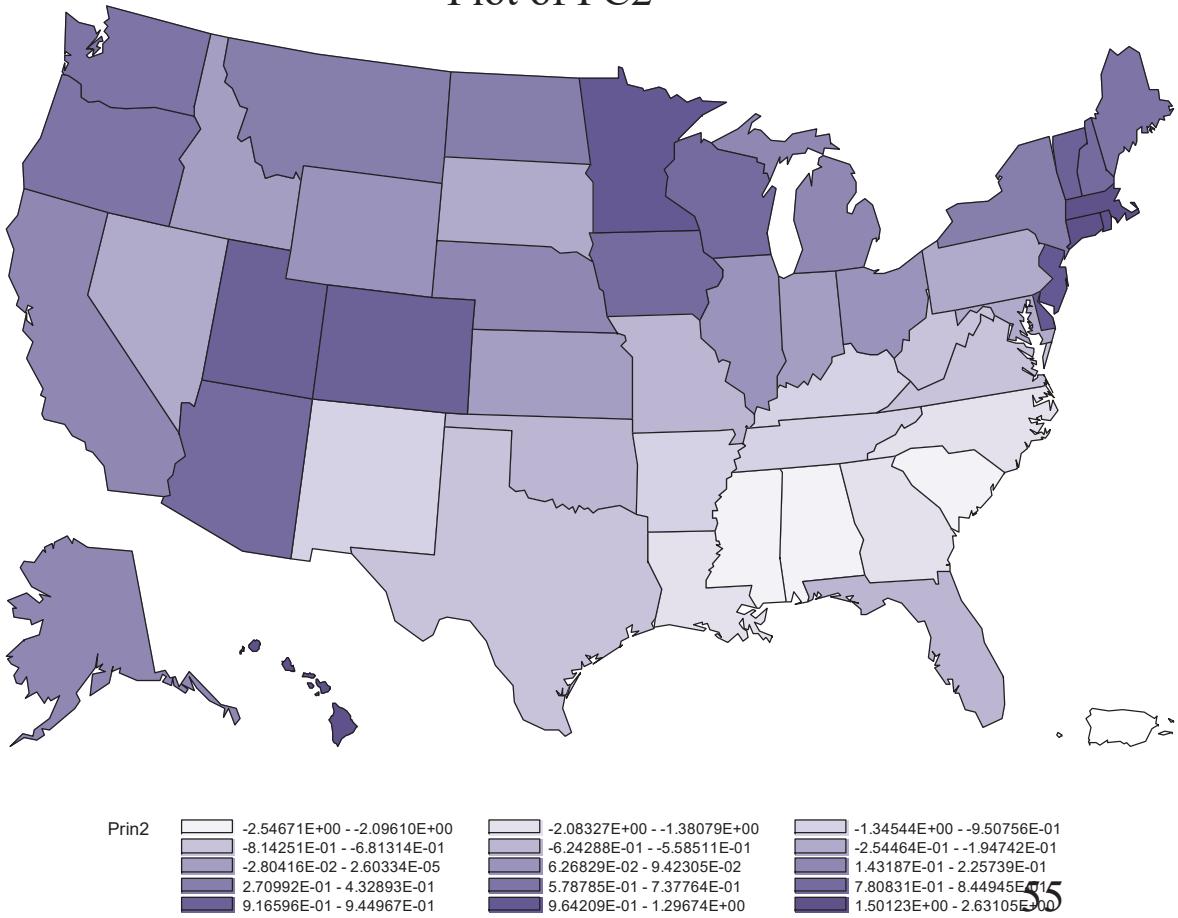
Plot of PC1



Prin1	-3.9640776 - -3.1477220	-2.5815619 - -2.4656229	-2.1507074 - -1.7269086	-1.7200694 - -1.5543424
	-1.5073580 - -1.4246347	-1.0544104 - -0.6992517	-0.6340669 - -0.4998955	-0.3213630 - -0.1365951
	-0.0498802 - 0.4904076	0.5129025 - 0.8231313	0.9305796 - 0.9784390	1.1202076 - 1.4490021
	1.6033606 - 2.2733344	2.4215150 - 3.0141383	3.1117540 - 5.2669853	

54

Plot of PC2



Decathlon: Seoul -1988

	100m	110m	400 m	1500 m	Long J.	High J.	Pole V.	Shot P.	Discus	Javaline
1	11.25	15.13	48.90	268.95	7.43	2.27	4.7	15.48	49.28	61.32
2	10.87	14.46	47.71	273.02	7.45	1.97	5.1	14.97	44.36	61.76
3	11.18	14.81	48.29	263.20	7.44	1.97	5.2	14.20	43.66	64.16
4	10.62	14.72	49.06	285.11	7.38	2.03	4.9	15.02	44.80	64.04
5	11.02	14.40	47.44	256.64	7.43	1.97	5.2	12.92	41.20	57.46
6	10.83	14.18	48.34	274.07	7.72	2.12	4.9	13.58	43.06	52.18
7	11.18	14.39	49.34	291.20	7.05	2.06	5.7	14.12	41.68	61.60
8	11.05	14.36	48.21	265.86	6.95	2.00	4.8	15.34	41.32	63.00
9	11.15	14.66	49.15	269.62	7.12	2.03	4.9	14.52	42.36	66.46
10	11.23	14.76	48.60	292.24	7.28	1.97	5.2	15.25	48.02	59.48
11	10.94	14.25	49.94	295.89	7.45	1.97	4.8	15.34	41.86	66.64
12	11.18	15.11	49.02	256.74	7.34	1.94	4.7	14.48	42.76	65.84
13	11.02	14.94	48.23	257.85	7.29	2.06	5.0	12.92	39.54	56.80
14	10.99	14.70	47.83	268.97	7.37	1.97	4.3	13.61	43.88	66.54
15	11.03	15.44	48.94	267.48	7.45	1.97	4.7	14.20	41.66	64.00
16	11.09	14.78	49.89	268.54	7.08	2.03	4.9	14.51	43.20	57.18
17	11.46	16.06	51.28	302.42	6.75	2.00	4.8	16.07	50.66	72.60

Decathlon: Seul -1988

	100m	110m	400 m	1500 m	Long J.	High J.	Pole V.	Shot P.	Discus	Javaline
18	11.57	15.00	49.84	286.04	7.00	1.94	4.9	16.60	46.66	60.20
19	11.07	14.96	47.97	262.41	7.04	1.94	4.5	13.41	40.38	51.50
20	10.89	15.38	49.68	277.84	7.07	1.79	4.9	15.84	45.32	60.48
21	11.52	15.64	49.99	266.42	7.36	1.94	4.6	13.93	38.82	67.04
22	11.49	15.22	50.60	262.93	7.02	2.03	4.7	13.80	39.08	60.92
23	11.38	14.97	50.24	272.68	7.08	2.00	4.4	14.31	46.34	55.68
24	11.30	15.38	49.98	277.84	6.97	2.15	4.6	13.23	38.72	54.34
25	11.00	14.96	49.73	285.57	7.23	2.03	4.5	13.15	38.06	52.82
26	11.33	15.39	48.37	270.07	6.83	2.06	4.6	11.63	37.52	55.42
27	11.10	15.13	48.63	261.90	6.98	1.82	4.7	12.69	38.04	49.52
28	11.51	15.18	51.16	303.17	7.01	1.94	4.6	14.17	45.84	56.28
29	11.26	15.61	48.24	272.06	6.90	1.88	4.4	12.41	38.02	52.68
30	11.50	15.56	49.27	293.85	7.09	1.82	4.5	12.94	42.32	53.50
31	11.43	15.88	51.25	294.99	6.22	1.91	4.6	13.98	46.18	57.84
32	11.47	15.00	50.30	293.72	6.43	1.94	4.0	12.33	38.72	57.26
33	11.57	16.20	50.71	269.98	7.19	1.91	4.1	10.27	34.36	54.94
34	12.12	17.05	52.32	281.24	5.83	1.70	2.6	9.71	27.10	39.10

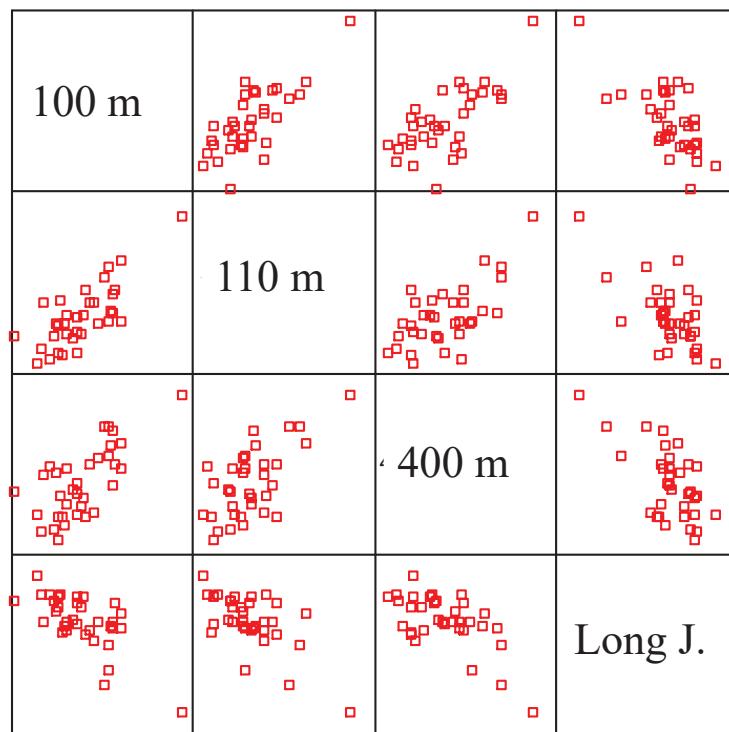
Decathlon: Seul -1988

Estadísticos descriptivos

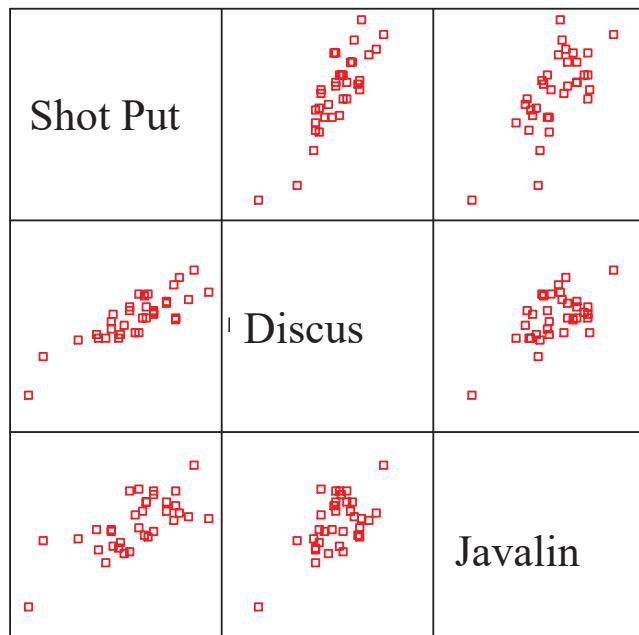
	Media	Desv. típ.
100 m	11,2235	,28723
110 m vallas	15,1076	,60566
400 m	49,3662	1,17555
1500 m	276,1915	13,47813
Salto Longitud	7,0950	,37387
Salto Altura	1,9744	,10448
Salto Pertiga	4,6765	,49302
Lanz. Peso	13,8509	1,50193
Lanz. Disco	41,9053	4,50071
Lanz. Jabalina	58,8406	6,43874

Correlations

	100 m	110 m vallas	400 m	1500 m	Salto Longit	Salto Altura	Salto Pertig	Lanz. Peso	Lanz. Disco	Lanz. Jabali
100 m	1 ,,000	,751 ,.000	,698 ,.000	,254 ,.148	-,691 ,.000	-,364 .034	-,627 .000	-,420 .013	-,353 .041	-,344 .046
110 m vallas	,751 ,.000	1 ,,000	,655 ,.380	,155 ,.000	-,654 ,.000	-,487 .004	-,709 .000	-,489 .003	-,403 .018	-,350 .042
400 m	,698 ,.000	,655 ,.000	1 ,,001	,554 ,.001	-,636 ,.000	-,275 .115	-,521 .002	-,142 .422	-,154 .383	-,150 .398
1500 m	,254 ,.148	,155 ,.380	,554 ,.001	1 ,,001	-,356 ,.039	-,132 .458	-,070 .693	,202 .252	,288 .098	,045 .801
Salto Longitud	-,691 ,.000	-,654 ,.000	-,636 ,.000	-,356 ,.039	1 ,,005	,471 ,.005	,632 .000	,391 .022	,375 .029	,446 .008
Salto Altura	-,364 ,.034	-,487 ,.004	-,275 .115	-,132 .458	,471 ,.005	1 ,,005	,472 .005	,321 .065	,376 .028	,338 .051
Salto Pertiga	-,627 ,.000	-,709 ,.000	-,521 ,.002	-,070 .693	,632 ,.000	,472 .005	1 ,,000	,643 .000	,620 .000	,557 .001
Lanz. Peso	-,420 ,.013	-,489 ,.003	-,142 .422	,202 .252	,391 ,.022	,321 .065	,643 .000	1 ,,000	,856 .000	,703 .000
Lanz. Disco	-,353 ,.041	-,403 ,.018	-,154 .383	,288 .098	,375 ,.029	,376 .028	,620 .000	,856 .000	1 ,,000	,618 .000
Lanz. Jabalina	-,344 ,.046	-,350 ,.042	-,150 .398	,045 .801	,446 ,.008	,338 .051	,557 .001	,703 .001	,618 .000	1 ,,000



Throwings



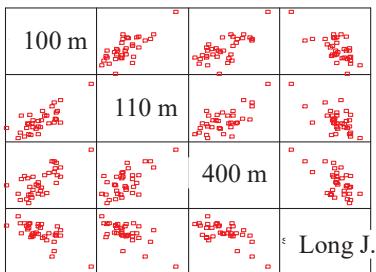
Correlations

	Long jump	Pole Vault	High Jump	Discus	Shot Put	Javelin	100m	110 m	400 m	1500 m
Long Jump	1	0.632	0.471	0.375	0.391	0.446	-0.691	-0.654	-0.636	-0.356
Pole Vault	0.632	1	0.472	0.620	0.643	0.557	-0.627	-0.709	-0.521	-0.070
High Jump	0.471	0.472	1	0.376	0.321	0.338	-0.364	-0.487	-0.275	-0.132
Discus	0.375	0.620	0.376	1	0.856	0.618	-0.353	-0.403	-0.154	0.288
Shot Put	0.391	0.643	0.321	0.856	1	0.703	-0.420	-0.489	-0.142	0.202
Javelin	0.446	0.557	0.338	0.618	0.703	1	-0.344	-0.350	-0.150	0.045
100 m	-0.691	-0.627	-0.364	-0.353	-0.420	-0.344	1	0.751	0.698	0.254
110 m	-0.654	-0.709	-0.487	-0.403	-0.489	-0.350	0.751	1	0.655	0.155
400 m	-0.636	-0.521	-0.275	-0.154	-0.142	-0.150	0.698	0.655	1	0.554
1500 m	-0.356	-0.070	-0.132	0.288	0.202	0.045	0.254	0.155	0.554	1

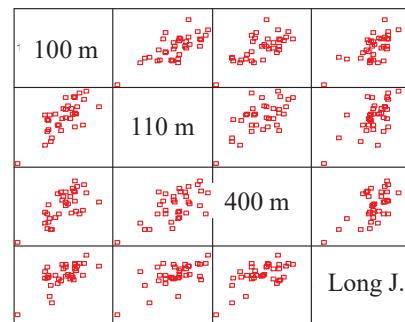
Correlations (speeds)

	Long jump	Pole Vault	High Jump	Discus	Shot Put	Javelin	100m	110 m	400 m	1500 m
Long Jump	1	0.632	0.471	0.375	0.391	0.446	0.682	0.642	0.625	0.357
Pole Vault	0.632	1	0.472	0.620	0.643	0.557	0.608	0.689	0.510	0.080
High Jump	0.471	0.472	1	0.376	0.321	0.338	0.350	0.475	0.266	0.129
Discus	0.375	0.620	0.376	1	0.856	0.618	0.338	0.381	0.146	-0.279
Shot Put	0.391	0.643	0.321	0.856	1	0.703	0.410	0.473	0.131	-0.198
Javelin	0.446	0.557	0.338	0.618	0.703	1	0.331	0.335	0.140	-0.034
100 m	0.682	0.608	0.350	0.338	0.410	0.331	1	0.736	0.682	0.247
110 m	0.642	0.689	0.475	0.381	0.473	0.335	0.736	1	0.638	0.151
400 m	0.625	0.510	0.266	0.146	0.131	0.140	0.682	0.638	1	0.557
1500 m	0.357	0.080	0.129	-0.279	-0.198	-0.034	0.247	0.151	0.557	1

Scatter plots (speeds)



Times



Speeds

	T100	T110	T400	LONG
T100	1.000	0.751	0.698	-0.691
T110	0.751	1.000	0.655	-0.654
T400	0.698	0.655	1.000	-0.636
LONG	-0.691	-0.654	-0.636	1.000

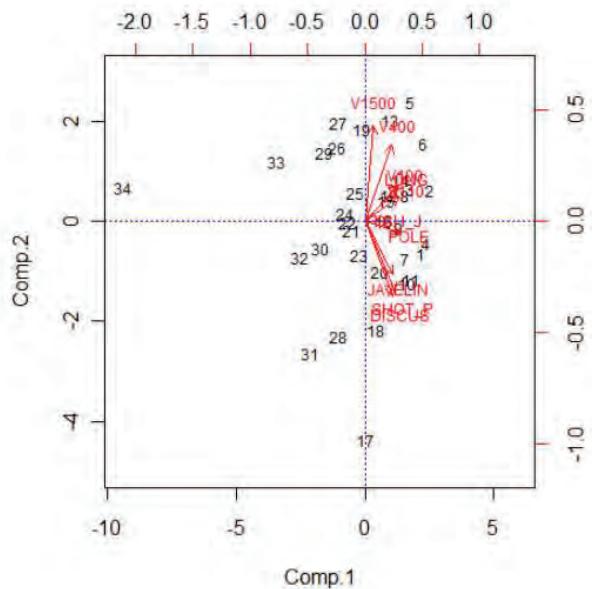
	V100	V110	V400	LONG
V100	1.0000	0.7358	0.6819	0.6816
V110	0.7358	1.0000	0.6385	0.6422
V400	0.6819	0.6385	1.0000	0.6254
LONG	0.6816	0.6422	0.6254	1.0000

Two Principal Components

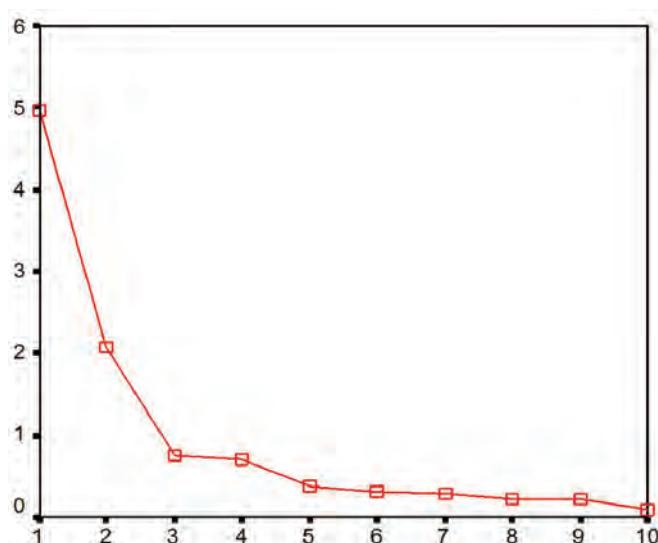
```
> prinfact(y,2)
[[1]]
      1           2 communality uniqueness
V100  0.7907982  0.300553407  0.7156941  0.2843059
V400  0.6483989  0.623630957  0.8093367  0.1906633
V110  0.8217556  0.200068702  0.7153097  0.2846903
V1500 0.1916215  0.777046112  0.6405194  0.3594806
LONG  0.8099560  0.283219050  0.7362418  0.2637582
HIGH_J 0.5983358  0.003538419  0.3580182  0.6419818
POLE   0.8700550 -0.089399516  0.7649879  0.2350121
SHOT_P 0.7265789 -0.569560356  0.8523159  0.1476841
DISCUS 0.6861732 -0.600460663  0.8313866  0.1686134
JAVELIN 0.6575130 -0.430750615  0.6178694  0.3821306

[[2]]
      1           2
Variance 4.9598878 2.0817920
Proportion 0.4959888 0.2081792
Cumulative 0.4959888 0.7041680
```

Biplot – Principal Components



Scree plot



Uso crime

Contents

Datos	1
Correlaciones	1
Componentes principales	3
Importancia de cada componentes	3
Cuantos componentes coger	4
Resumen del análisis con dos componentes	5
Gráficos importantes	6

Datos

```
library(corrplot)

## corrplot 0.84 loaded

dat = read.table("USAcime.txt", header=TRUE)
row.names(dat) = dat$STATE
dat = dat[,-1]
```

He quitado el nombre de los estados como variable y la he puesto como etiqueta de cada observación. Es útil para los gráficos finales.

Correlaciones

```
r = cor(dat)
print(r, digits = 3)
```

```

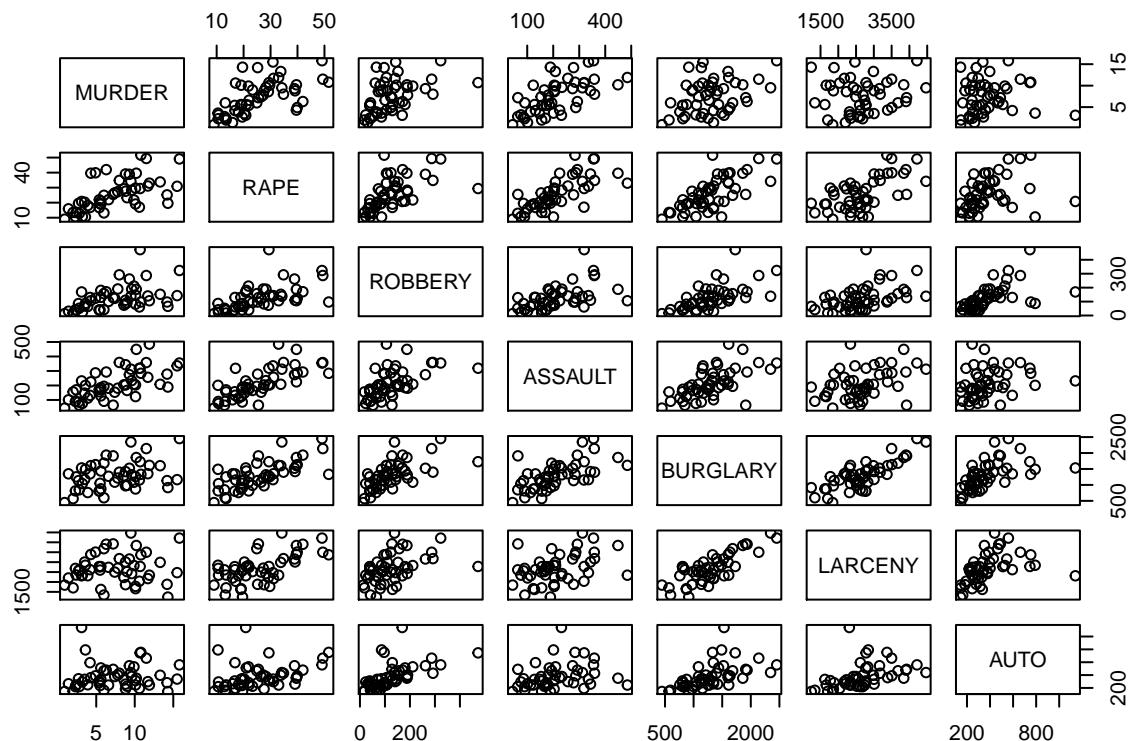
##          MURDER    RAPE ROBBERY ASSAULT BURGLARY LARCENY    AUTO
## MURDER  1.0000  0.601   0.484   0.649   0.386   0.102  0.0688
## RAPE    0.6012  1.000   0.592   0.740   0.712   0.614  0.3489
## ROBBERY 0.4837  0.592   1.000   0.557   0.637   0.447  0.5907
## ASSAULT 0.6486  0.740   0.557   1.000   0.623   0.404  0.2758
## BURGLARY 0.3858  0.712   0.637   0.623   1.000   0.792  0.5580
## LARCENY  0.1019  0.614   0.447   0.404   0.792   1.000  0.4442
## AUTO    0.0688  0.349   0.591   0.276   0.558   0.444  1.0000

```

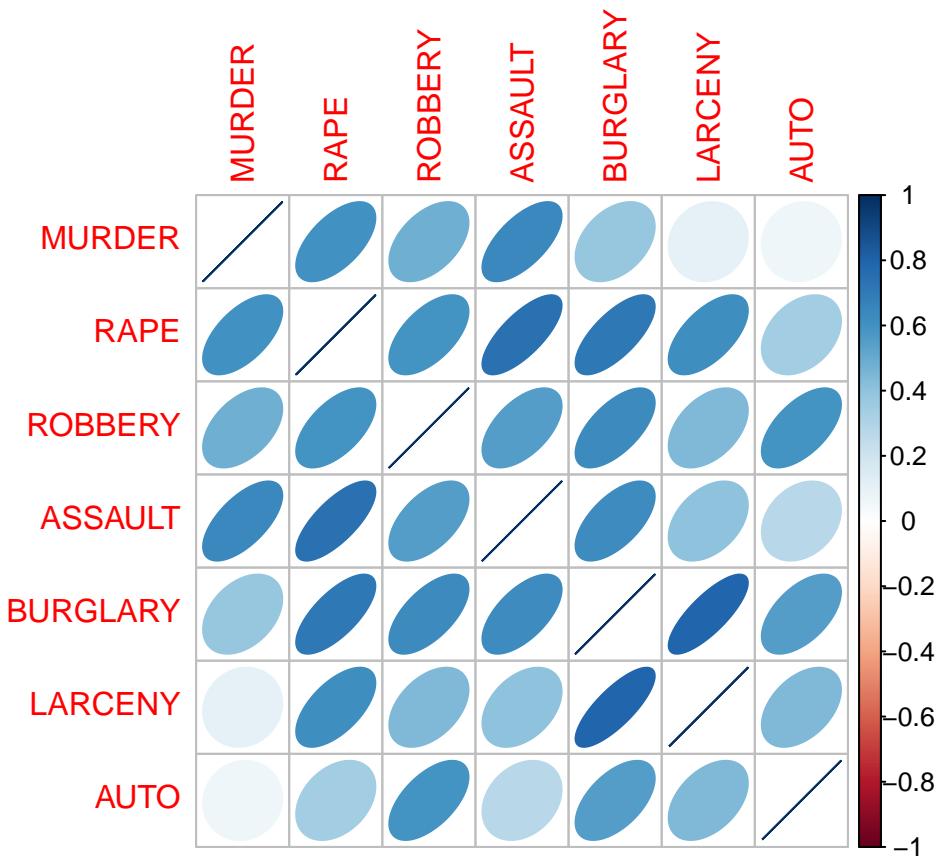
Las correlaciones son “importantes” y positivas.

Vamos a ver que las relaciones son lineales:

```
pairs(dat)
```



```
corrplot(r,method = "ellipse")
```



Componentes principales

```
m = princomp(dat, cor=TRUE)
names(m)
```

```
## [1] "sdev"      "loadings"   "center"     "scale"      "n.obs"      "scores"    "call"
```

Importancia de cada componentes

Se han calculado los seis componentes y están ordenados por orden de importancia. m\$sdev contienen las desviaciones típicas de cada componente, de mayor a menor.

```
m
```

```
## Call:
## princomp(x = dat, cor = TRUE)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 2.0285363 1.1129788 0.8519487 0.5625229 0.5079119 0.4712106 0.3522159
##
## 7 variables and 50 observations.
```

```
m$sdev # desviaciones de cada componente
```

```
##      Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 2.0285363 1.1129788 0.8519487 0.5625229 0.5079119 0.4712106 0.3522159
```

```
m$sdev^2 # varianzas de cada componente (suman 7)
```

```
##      Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 4.1149595 1.2387218 0.7258166 0.3164320 0.2579745 0.2220395 0.1240561
```

```
m$sdev^2/7*100# porcentaje que explica cada componente
```

```
##      Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 58.785136 17.696026 10.368809  4.520458  3.685349  3.171992  1.772229
```

```
cumsum(m$sdev^2/7*100) # porcentaje acumulado
```

```
##      Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 58.78514 76.48116 86.84997 91.37043 95.05578 98.22777 100.00000
```

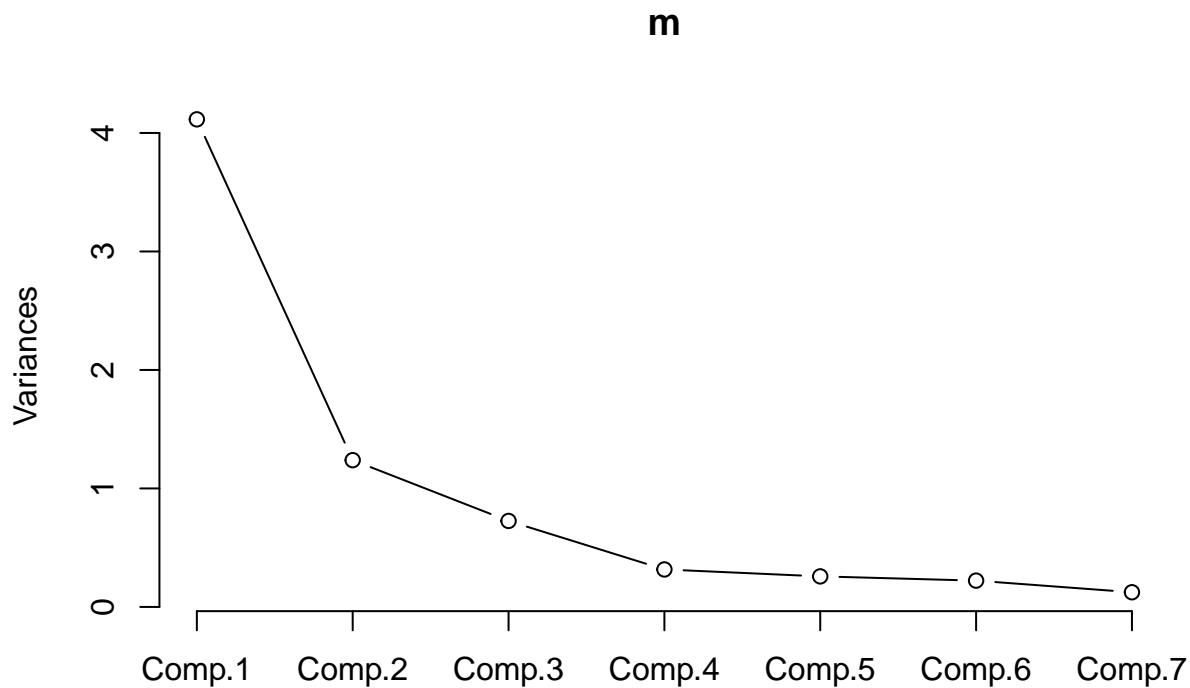
```
m$loadings
```

```
##
## Loadings:
##      Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## MURDER   0.300  0.629  0.178  0.232  0.538  0.259  0.268
## RAPE     0.432  0.169 -0.244          0.188 -0.773 -0.296
## ROBBERY  0.397          0.496  0.558 -0.520 -0.114
## ASSAULT  0.397  0.344          -0.630 -0.507  0.172  0.192
## BURGLARY 0.440 -0.203 -0.210          0.101  0.536 -0.648
## LARCENY  0.357 -0.402 -0.539  0.235          0.602
## AUTO     0.295 -0.502  0.568 -0.419  0.370          0.147
##
##      Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings    1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var 0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var 0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

Cuantos componentes coger

Es dudoso. Para simplificar la explicación cogemos dos y explicamos el 76.5% de la variabilidad total de los datos.

```
plot(m,type="lines")
```



Resumen del análisis con dos componentes

```

source("prinfect.R")
sol = prinfect(dat,2)
names(sol)

## [1] "loadings"  "variances" "scores"    "eig"

sol$loadings

##           Comp 1      Comp 2 communality uniqueness
## MURDER    0.6091272  0.70025782   0.8613969  0.1386031
## RAPE      0.8758395  0.18857770   0.8026564  0.1973436
## ROBBERY   0.8050763 -0.04701999   0.6503588  0.3496412
## ASSAULT   0.8046224  0.38233955   0.7936007  0.2063993
## BURGLARY  0.8928749 -0.22631376   0.8484435  0.1515565
## LARCENY   0.7249168 -0.44777266   0.7260047  0.2739953
## AUTO      0.5987769 -0.55918385   0.6712203  0.3287797

sol$variances

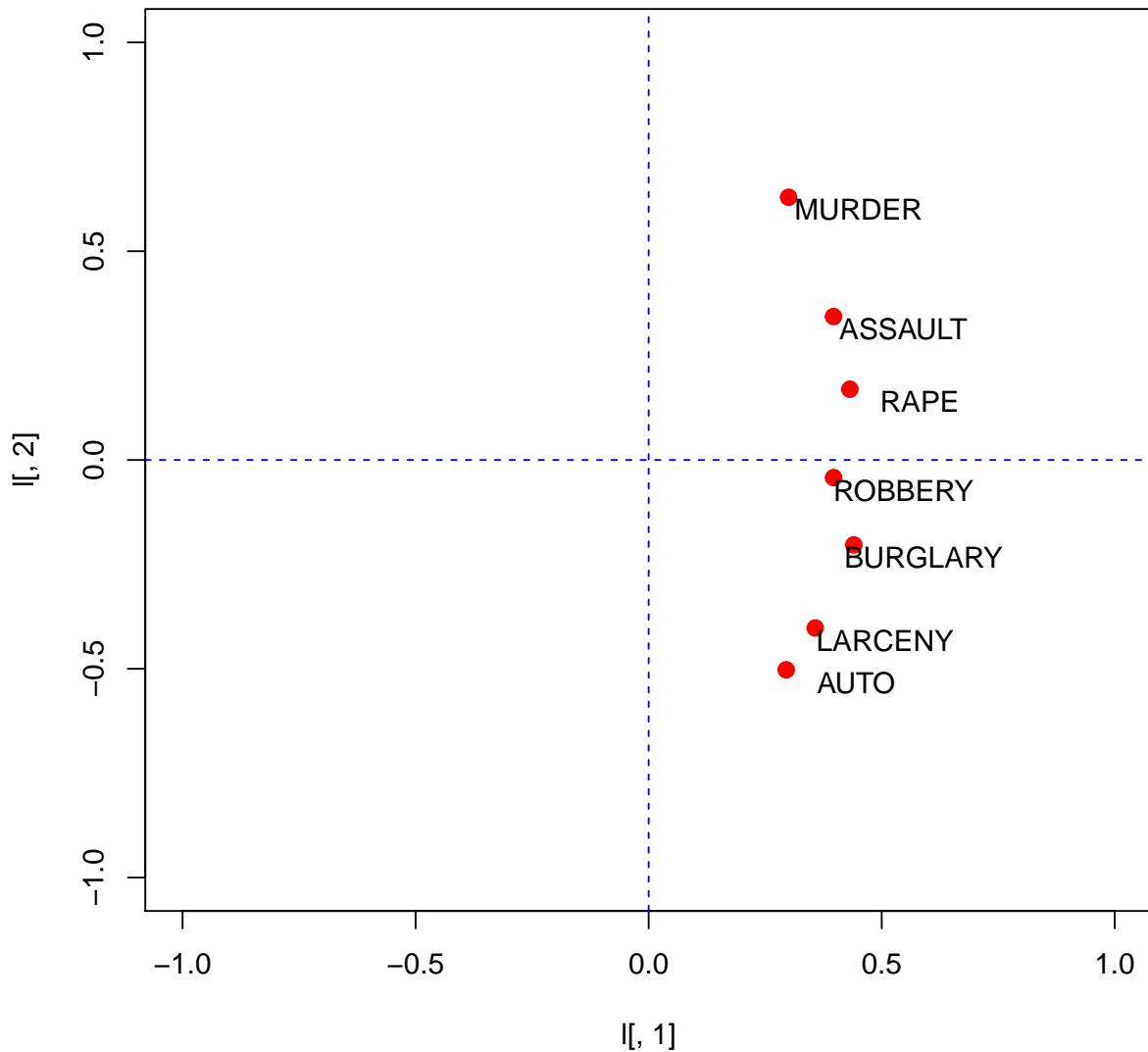
```

```
##          Comp 1     Comp 2
## Variance   4.1149595 1.2387218
## Proportion 0.5878514 0.1769603
## Cumulative 0.5878514 0.7648116
```

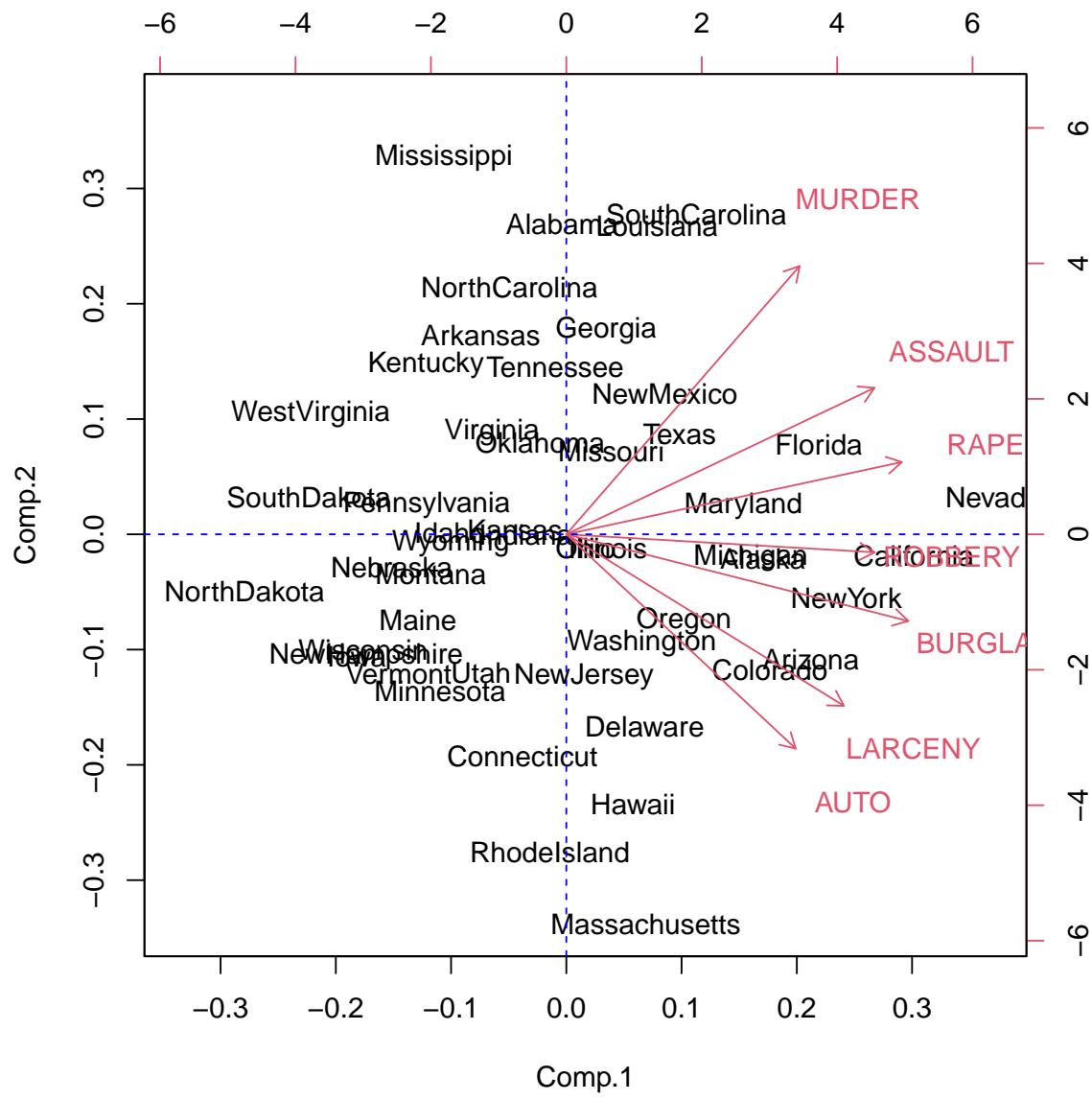
- Con la solución de dos componentes explicamos el 76.5% de la información total.
- No todas las variables originales están “igualmente” explicadas. Las mejores explicadas son BURGLARY (84.8%) y MURDER (86.1%). Las peores explicadas AUTO(67.1%) y ROBBERY(65%)

Gráficos importantes

```
l = loadings(m)
plot(l[,1],l[,2],col="red",cex=1.2,
      xlim=c(-1,1), ylim=c(-1,1),pch=19)
abline(h=0,col="blue",lty=2)
abline(v=0,col="blue",lty=2)
text(l[,1]+.15,l[,2]-.03,colnames(dat))
```



```
biplot(m)
abline(h=0,col="blue",lty=2)
abline(v=0,col="blue",lty=2)
```



Módulo de Análisis Multivariante

Componentes Principales

Ejercicio 1: Componentes Principales Cuerpo Humano

Con los datos del archivo **cuerpo.txt** vamos a realizar el análisis de componentes principales de las variables con medidas de contorno (empiezan por C)

```
## [1] "C_hombros" "C_pecho"    "C_cintura"  "C_abdomen" "C_cadera"   "C_muslo"
## [7] "C_biceps"   "C_brazo"     "C_rodilla"  "C_gemelo"  "C_tobillo"  "C_muneca"
```

En el documento adjunto “Descripción Dataset Cuerpo” se proporciona la información de los datos que contiene el archivo “cuerpo.txt”. Hay distintas versiones del archivo “cuerpo.txt”, importante utilizar el que se proporciona con este enunciado.

1. Para las 12 variables anteriores calcula la matriz de correlaciones utilizando solo los datos de mujeres. Comprueba con los gráficos de dispersión si las relaciones son lineales. Indica las tres correlaciones más altas (entre qué variables se producen). Indican las tres correlaciones más bajas (entre que variables se producen).
2. Repite el apartado 1 para los datos de hombres.
3. Realiza el análisis de componentes principales (con los datos estandarizados) para las 12 variables de las medidas de contorno de las mujeres. Toma la solución con dos componentes.
 - a. ¿Qué porcentaje de la variabilidad total de las 12 variables está explicada por los dos primeros componentes?
 - b. ¿Qué variable de las 12 está mejor explicada? Interpreta este resultado.
 - c. ¿Qué variable de las 12 está peor explicada? Interpreta este resultado.
 - d. Representa en un gráfico de dimensión 2 los scores de cada individuo. Interpreta la distribución de los puntos. Qué personas están a la derecha, a la izquierda, arriba y abajo del gráfico.
4. Repite el análisis de la pregunta 3 aplicado a hombres.

Ejercicio 2: Componentes Principales Olimpiadas de Seúl 1988

El Decathlon es una disciplina olímpica que se desarrolla en diez pruebas: 4 carreras (100,110 vayas, 400 y 1500 metros), tres lanzamientos (jabalina, disco y peso) y tres saltos (pértiga, altura y longitud). En el archivo **seoul1988.txt** tienes los resultados de 34 atletas en las diez pruebas realizadas en las Olimpiadas de Seúl en 1988.

Vamos a analizar las correlaciones entre las diez variables. Las unidades de las variables de saltos y lanzamientos son metros, las unidades de las carreras son los segundos que el atleta ha tardado en completar la distancia.

Para facilitar el análisis y la interpretación vamos a transformar las variables correspondientes a carreras (100, 110, 400 y 1500). Para cada atleta calcularemos la velocidad media con lo que ha realizado cada prueba. ($V100=100/T100$, $V400=400/T400$, $V110 = 110/T110$ y $V1500 = 1500/T1500$).

Realiza el análisis de componentes principales (con los datos estandarizados) para las 10 pruebas de Decathlon (emplea las nuevas variables de velocidad en lugar de las variables con los tiempos).

Toma la solución con dos componentes.

- a. ¿Qué porcentaje de la variabilidad total de las 10 variables está explicada por los dos primeros componentes?
- b. ¿Qué variable de las 10 está mejor explicada? Interpreta este resultado.
- c. ¿Qué variable de las 10 está peor explicada? Interpreta este resultado.
- d. Representa en un gráfico de dimensión 2 los scores de cada individuo. Interpreta la distribución de los puntos. Qué personas están a la derecha, a la izquierda, arriba y abajo del gráfico.
- e. Explica el significado del primer componente
- f. Explica el significado del segundo componente.
- g. Calcula la solución de tres componentes principales. Explica los cambios más relevantes entre esta solución y la solución de dos componentes.

Ejercicio 3. Componentes principales Elecciones Comunidad de Madrid 2021

En el archivo “Madrid_2021.txt” se proporciona los resultados por barrios de Madrid Capital de las elecciones a la Asamblea de Madrid celebradas el 4 de Mayo de 2021. Para cada barrio se proporciona el nombre, el censo y el porcentaje sobre el censo de votos a los partidos más votados: Vox, PP, Ciudadanos (Cs), PSOE, Más Madrid y Unidas-Podemos. Además se proporciona el porcentaje de votos a otras candidaturas y el porcentaje de abstención.

Los datos se han obtenido de la página web:

<https://datos.gob.es/es/catalogo/l01280796-elecciones-asamblea-de-madrid-1983-2019>

La variable **censo** se proporciona para completar la información, es muy relevante para el análisis de los resultados, pero en este ejercicio no se utiliza. Nos centraremos en las variables porcentaje de votos: **vox, pp, cs, psoe, masmadrid, podemos, otros y abstención**.

- a. Explica la solución de dos componentes
- b. ¿Cuántos componentes son necesarios? Utiliza el método gráfico (scree plot) para elegir el número de componentes.
- c. Realiza el gráfico de dimensión 2 con los scores (componentes principales). Explica e interpreta el gráfico. En el gráfico, pon el nombre de los barrios de Madrid para facilitar la interpretación.

Componentes Principales: Cuerpo Humano

Enunciado

Con los datos del archivo **cuerpo.txt** vamos a realizar el análisis de componentes principales de las variables con medidas de contorno (empiezan por C)

```
## [1] "C_hombros"  "C_pecho"     "C_cintura"   "C_abdomen"  "C_cadera"    "C_muslo"
## [7] "C_biceps"   "C_brazo"     "C_rodilla"   "C_gemelo"   "C_tobillo"   "C_muneca"
```

En el documento adjunto “Descripción Dataset Cuerpo” se proporciona la información de los datos que contiene el archivo “cuerpo.txt”. Hay distintas versiones del archivo “cuerpo.txt”, importante utilizar el que se proporciona con este enunciado.

1. Para las 12 variables anteriores calcula la matriz de correlaciones utilizando solo los datos de mujeres. Comprueba con los gráficos de dispersión si las relaciones son lineales. Indica las tres correlaciones más altas (entre qué variables se producen). Indican las tres correlaciones más bajas (entre que variables se producen). **(1.5 punto)**
2. Repite el apartado 1 para los datos de hombres. **(1.5 punto)**
3. Realiza el análisis de componentes principales (con los datos estandarizados) para las 12 variables de las medidas de contorno de las mujeres. Toma la solución con dos componentes. **(1.5 punto)**
 - a. ¿Qué porcentaje de la variabilidad total de las 12 variables está explicada por los dos primeros componentes?
 - b. ¿Qué variable de las 12 está mejor explicada? Interpreta este resultado.
 - c. ¿Qué variable de las 12 está peor explicada? Interpreta este resultado.
 - d. Representa en un gráfico de dimensión 2 los scores de cada individuo. Interpreta la distribución de los puntos. Qué personas están a la derecha, a la izquierda, arriba y abajo del gráfico.
4. Repite el análisis de la pregunta 3 aplicado a hombres. **(1.5 punto)**

Solución

```
source("prinfact.R")
library(MASS) # necesaria para Lda()
library(car) # necesaria para scatterPlotMatrix

## Loading required package: carData

dat = read.table("cuerpo.txt", header=TRUE)
dat$sexo = factor(dat$sexo, labels = c("Mujer", "Hombre"))

muj = dat$sexo == "Mujer"
dat0 = dat[muj,] # mujeres
dat1 = dat[!muj,] # hombres

sel = 10:21
```

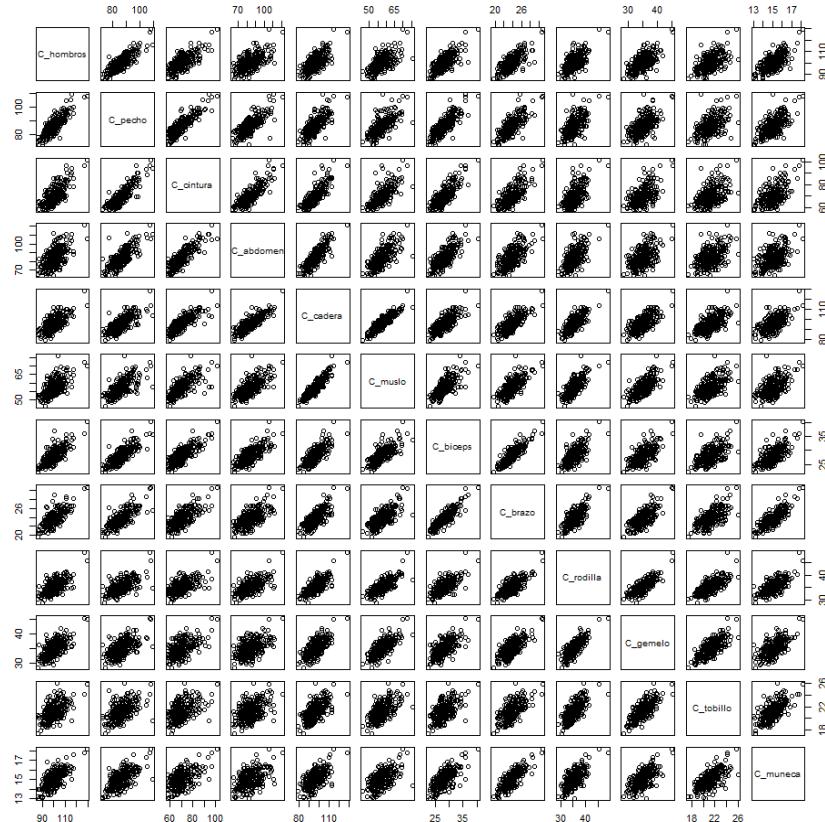
1- Para las 12 variables anteriores calcula la matriz de correlaciones utilizando solo los datos de mujeres. Comprueba con los gráficos de dispersión si las relaciones son lineales. Indica las tres correlaciones más altas (entre qué variables se producen). Indican las tres correlaciones más bajas (entre qué variables se producen).

```
r0 = cor(dat0[,sel])
print(r0, digits = 3)

##          C_hombros C_pecho C_cintura C_abdomen C_cadera C_muslo C_biceps
## C_hombros    1.000   0.827   0.726     0.613   0.679   0.630    0.744
## C_pecho      0.827   1.000   0.859     0.766   0.744   0.675    0.816
## C_cintura    0.726   0.859   1.000     0.835   0.812   0.728    0.797
## C_abdomen    0.613   0.766   0.835     1.000   0.830   0.701    0.751
## C_cadera     0.679   0.744   0.812     0.830   1.000   0.904    0.768
## C_muslo       0.630   0.675   0.728     0.701   0.904   1.000    0.749
## C_biceps     0.744   0.816   0.797     0.751   0.768   0.749    1.000
## C_brazo       0.747   0.767   0.708     0.640   0.747   0.726    0.868
## C_rodilla    0.647   0.620   0.631     0.613   0.762   0.764    0.680
## C_gemelo     0.632   0.576   0.577     0.520   0.693   0.729    0.666
## C_tobillo    0.555   0.515   0.487     0.487   0.602   0.582    0.570
## C_muneca     0.659   0.640   0.545     0.493   0.619   0.567    0.683
##          C_brazo C_rodilla C_gemelo C_tobillo C_muneca
## C_hombros   0.747   0.647   0.632   0.555   0.659
## C_pecho     0.767   0.620   0.576   0.515   0.640
## C_cintura   0.708   0.631   0.577   0.487   0.545
## C_abdomen   0.640   0.613   0.520   0.487   0.493
## C_cadera    0.747   0.762   0.693   0.602   0.619
## C_muslo     0.726   0.764   0.729   0.582   0.567
## C_biceps    0.868   0.680   0.666   0.570   0.683
## C_brazo     1.000   0.749   0.741   0.645   0.807
## C_rodilla   0.749   1.000   0.795   0.702   0.701
## C_gemelo    0.741   0.795   1.000   0.738   0.654
## C_tobillo   0.645   0.702   0.738   1.000   0.666
## C_muneca    0.807   0.701   0.654   0.666   1.000
```

Las correlaciones son todas positivas, están entre 0.48 y 0.90. Se observa que las relaciones son lineales.

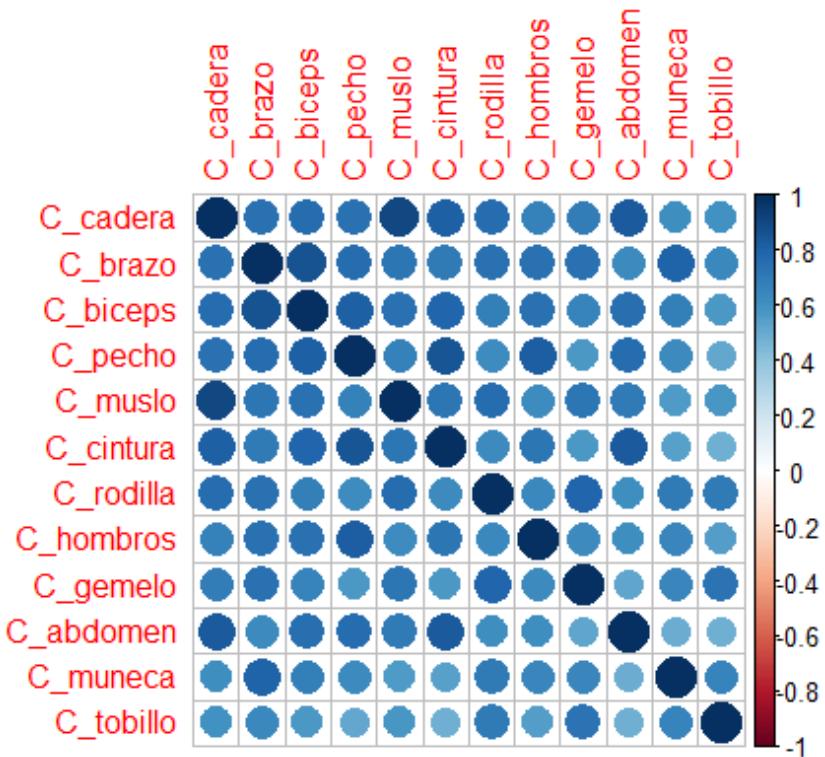
```
pairs(dat0[,sel])
```



Las correlaciones más altas se producen entre Muslo - Cadera (0.904), Biceps-Brazo (0.868) y Cintura-Pecho (0.8591)

Las correlaciones más baja entre Tobillo- Abdomen (0.487), Tobillo-Cintura (0.487) y Abdomen-Muñeca (0.493)

```
library(corrplot)
## corrplot 0.90 loaded
corrplot(r0,order="FPC")
```

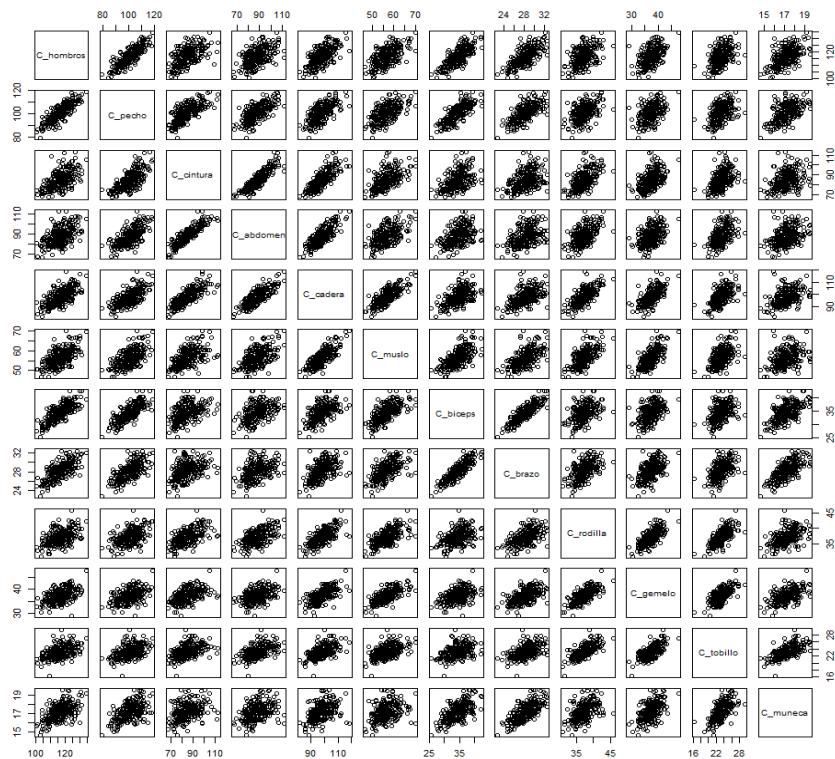


2. Repite el apartado 1 para los datos de hombres.

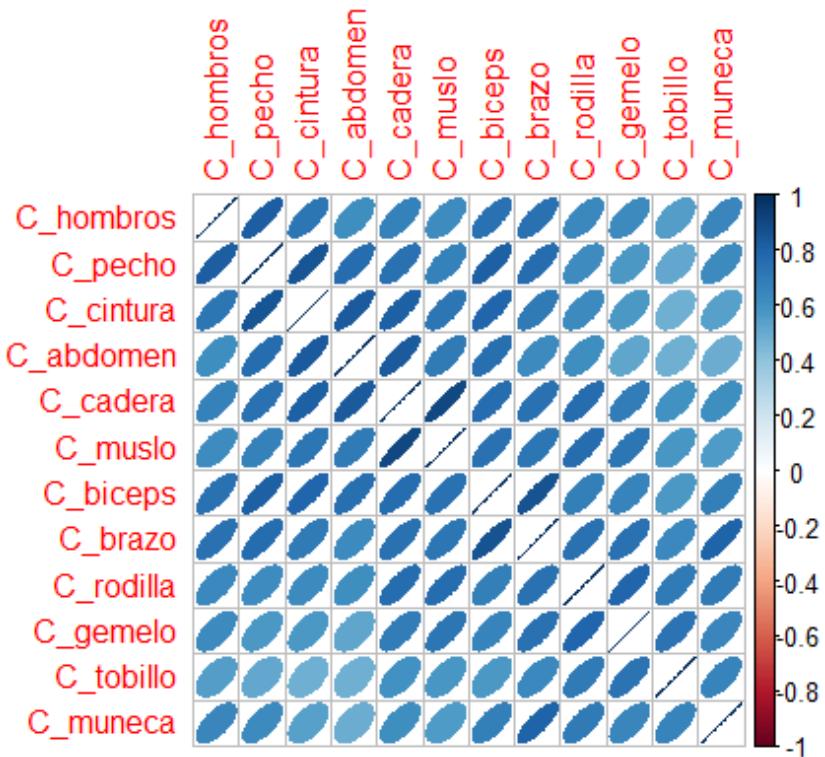
```
r1 = cor(dat1[,sel])
print(r1,digits = 3)

##          C_hombros C_pecho C_cintura C_abdomen C_cadera C_muslo C_biceps
## C_hombros    1.000   0.835   0.581     0.539   0.652   0.606   0.764
## C_pecho      0.835   1.000   0.713     0.685   0.682   0.602   0.778
## C_cintura    0.581   0.713   1.000     0.882   0.800   0.560   0.471
## C_abdomen    0.539   0.685   0.882     1.000   0.809   0.545   0.475
## C_cadera     0.652   0.682   0.800     0.809   1.000   0.791   0.572
## C_muslo      0.606   0.602   0.560     0.545   0.791   1.000   0.662
## C_biceps     0.764   0.778   0.471     0.475   0.572   0.662   1.000
## C_brazo       0.703   0.701   0.421     0.426   0.538   0.611   0.865
## C_rodilla     0.500   0.481   0.569     0.555   0.704   0.664   0.429
## C_gemelo     0.494   0.475   0.517     0.463   0.654   0.705   0.475
## C_tobillo     0.470   0.471   0.478     0.515   0.587   0.520   0.428
## C_muneca     0.541   0.546   0.369     0.389   0.450   0.382   0.611
##          C_brazo C_rodilla C_gemelo C_tobillo C_muneca
## C_hombros    0.703   0.500   0.494   0.470   0.541
## C_pecho      0.701   0.481   0.475   0.471   0.546
## C_cintura    0.421   0.569   0.517   0.478   0.369
## C_abdomen    0.426   0.555   0.463   0.515   0.389
## C_cadera     0.538   0.704   0.654   0.587   0.450
## C_muslo      0.611   0.664   0.705   0.520   0.382
## C_biceps     0.865   0.429   0.475   0.428   0.611
## C_brazo       1.000   0.528   0.554   0.513   0.708
## C_rodilla     0.528   1.000   0.727   0.689   0.497
## C_gemelo     0.554   0.727   1.000   0.695   0.539
## C_tobillo     0.513   0.689   0.695   1.000   0.629
## C_muneca     0.708   0.497   0.539   0.629   1.000

pairs(dat1[,sel])
```



```
library(corrplot)
corrplot(r0, method = "ellipse")
```



Las correlaciones mayores se dan entre Cintura y Abdomen (0.882), Brazo y Biceps (.8646) y Pecho y Hombro (0.8349). Las correlaciones menores se dan entre Muñeca y Cintura (0.3688), Muslo-Muñeca (0.3824) y Abdomen-Muñeca (0.3891)

Parece que tienen lógica estos resultados, las medidas de las partes del cuerpo más próximas se parecen más que las más alejadas.

3. Realiza el análisis de componentes principales (con los datos estandarizados) para las 12 variables de las medidas de contorno de las mujeres. Toma la solución con dos componentes.
 - a. ¿Qué porcentaje de la variabilidad total de las 12 variables está explicada por los dos primeros componentes?
 - b. ¿Qué variable de las 12 está mejor explicada? Interpreta este resultado.
 - c. ¿Qué variable de las 12 está peor explicada? Interpreta este resultado.
 - d. Representa en un gráfico de dimensión 2 los scores de cada individuo. Interpreta la distribución de los puntos. Qué personas están a la derecha, a la izquierda, arriba y abajo del gráfico.

```
m0 = prinfact(dat0[,sel],2)
m0$loadings

##           Comp 1      Comp 2 communality uniqueness
## C_hombros 0.8341144  0.07902883   0.7019923  0.2980077
## C_pecho   0.8704837  0.29837846   0.8467716  0.1532284
## C_cintura 0.8613778  0.38530336   0.8904304  0.1095696
## C_abdomen 0.8155069  0.39711806   0.8227543  0.1772457
## C_cadera  0.9049913  0.14088841   0.8388589  0.1611411
## C_muslo   0.8652339  0.03609471   0.7499325  0.2500675
## C_biceps  0.8990499  0.13742210   0.8271755  0.1728245
## C_brazo   0.9020096 -0.11200374   0.8261662  0.1738338
## C_rodilla 0.8520201 -0.27918052   0.8038800  0.1961200
## C_gemelo  0.8166777 -0.38485460   0.8150756  0.1849244
## C_tobillo 0.7358360 -0.47805596   0.7699921  0.2300079
## C_muneca  0.7879072 -0.32835170   0.7286126  0.2713874

m0$variances

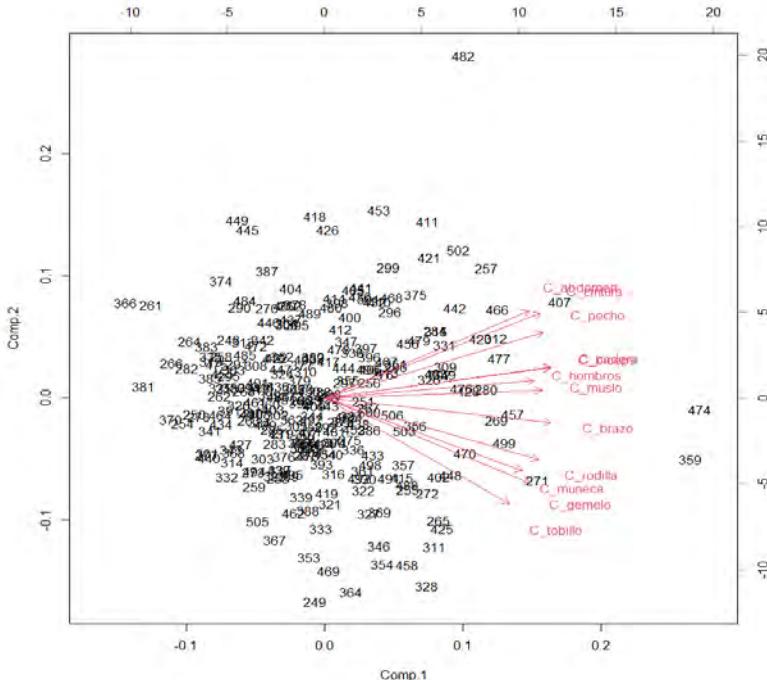
##           Comp 1      Comp 2
## Variance  8.6052160 1.01642589
## Proportion 0.7171013 0.08470216
## Cumulative 0.7171013 0.80180349
```

Los dos primeros componentes explican el 80.2% de la variabilidad total de las 12 variables.

La variable mejor explicada con los dos componentes es el contorno de la cintura: Parte común (communality) igual a 89% y parte específica (uniqueness) del 11%.

La peor explicada es el contorno de hombros que tiene una parte específica de 29.8% y una parte común del 70.2 %

```
p0 = princomp(dat0[,sel], cor=TRUE)
biplot(p0)
```



Para obtener el primer componente observamos que los pesos (o cargas) son todos positivos y de magnitud similar. Es una medida global del “tamaño” de la mujer. A la derecha se encuentran las mujeres de mayor tamaño y a la izquierda las de menor tamaño.

Para comprobar esto podemos proporcionar los datos de la observación 474 y de la 366.

```
dat[c(474,366),sel]

##      C_hombros C_pecho C_cintura C_abdomen C_cadera C_muslo C_biceps C_brazo
## 474      127.1    106.9     96.2     121.1    128.3    72.3     35.9    30.6
## 366      85.9     74.5     61.0     65.3     84.5    51.5     22.4    19.6
##      C_rodilla C_gemelo C_tobillo C_muñeca
## 474      49.0     45.4     24.1     17.8
## 366      30.4     28.4     17.4     13.2
```

La posición en el eje de ordenadas la proporciona la puntuación en el segundo componente (score). El segundo componente tiene pesos positivos altos para las medidas de Pecho, Cintura y Abdomen y pesos negativos (altos en valor absoluto) para Rodilla, Gemelo, Tobillo y Muñeca. Contraponen las medidas del tronco de la mujer y las de las extremidades. Arriba en el gráfico estarán las mujeres con valores altos (por encima de la media) de Pecho, Cintura y Abdomen y bajos (por debajo de la media) en

Rodilla, Gemelo, Tobillo y Muñeca. En la parte inferior del gráfico, al revés, estarán las mujeres con valores bajos (por debajo de la media) de Pecho, Cintura y Abdomen y altos (por encima de la media) en Rodilla, Gemelo, Tobillo y Muñeca. Para comprobar esto proporcionamos los valores de las observaciones 482 y 249.

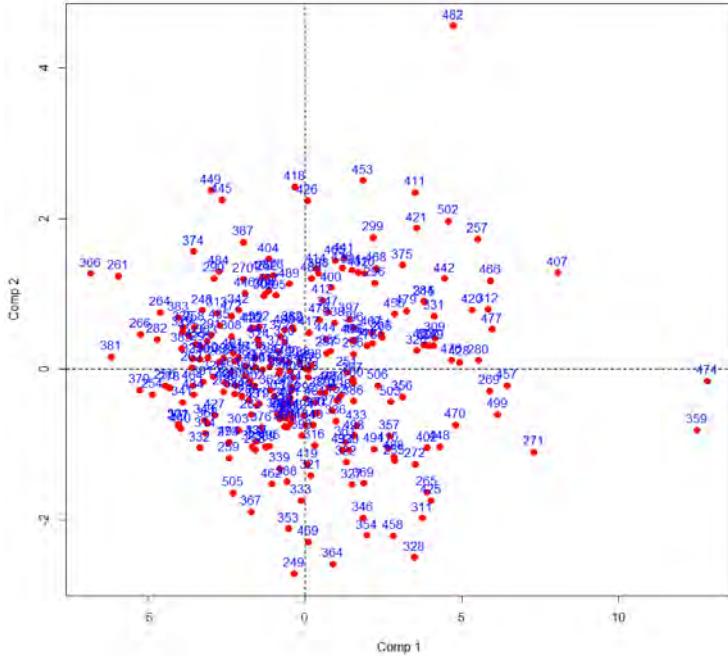
```
dat[c(482,249),c(11,12,13,18,19,20,21)]  
  
##      C_pecho C_cintura C_abdomen C_rodilla C_gemelo C_tobillo C_muneca  
## 482     109.0      94.2     110.5     34.8     35.5     19.5      15  
## 249      78.5      61.5      70.5     38.5     38.5     22.5      15
```

En la interpretación he destacado las variables con más peso, aunque en el cálculo intervienen todas y es preciso también considerar el valor estandarizado de cada variable.

La escala del gráfico “biplot” que proporciona por defecto R tiene una escala adaptada para facilitar la visualización conjunta de scores y weights. Los valores concretos no son muy importantes, lo relevante es la posición relativa de las observaciones.

El gráfico de las puntuaciones (scores) se puede obtener con las siguientes instrucciones:

```
plot(m0$scores[,1],m0$scores[,2],pch=19,  
      col="red",cex=1.2,xlab="Comp 1", ylab = "Comp 2")  
text(m0$scores[,1],m0$scores[,2],row.names(dat0),pos=3,col="blue")  
abline(h=0,lty=2)  
abline(v=0,lty=2)
```



4. Repite el análisis de la pregunta 3 aplicado a hombres.

```
m1 = prinfact(dat1[,sel],2)
m1$loadings
```

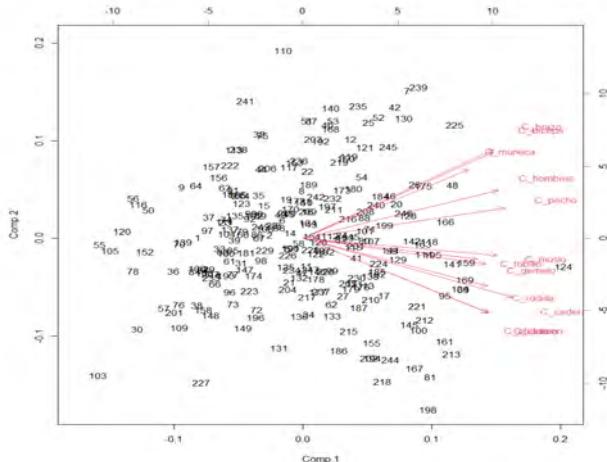
```
##          Comp 1      Comp 2 communality uniqueness
## C_hombros 0.8159192  0.26505647  0.7359790  0.2640210
## C_pecho   0.8465849  0.16630348  0.7443628  0.2556372
## C_cintura 0.7806692 -0.41073404  0.7781468  0.2218532
## C_abdomen 0.7720164 -0.40930353  0.7635386  0.2364614
## C_cadera  0.8745560 -0.32536989  0.8707138  0.1292862
## C_muslo   0.8121609 -0.09295544  0.6682461  0.3317539
## C_biceps  0.7977624  0.47477779  0.8618388  0.1381612
## C_brazo   0.7984699  0.49042456  0.8780704  0.1219296
## C_rodilla 0.7726689 -0.26330937  0.6663491  0.3336509
## C_gemelo  0.7668793 -0.14247683  0.6084034  0.3915966
## C_tobillo 0.7307568 -0.11313520  0.5468051  0.4531949
## C_muneca  0.6945913  0.38104343  0.6276512  0.3723488

m1$variances
```

```
##          Comp 1      Comp 2
## Variance 7.4878999 1.2622052
## Proportion 0.6239917 0.1051838
## Cumulative 0.6239917 0.7291754
```

Para el caso de hombres, la solución de dos componentes explica el 72.9% de la variabilidad total. Los dos primeros componentes tienen valor propio 7.49 y 1.26, respectivamente. La variable mejor explicada es el contorno del brazo (87.8%). Otras variables bien explicadas son Cadera y Biceps. El perímetro de la cintura que era la variable mejor explicada para mujeres, no es de las mejores en el caso de hombres. La variable que tiene menos en común con las demás es el perímetro de tobillo que está explicada al 54.7% con una unicidad del 45.3%.

```
p1 = princomp(dat1[,sel],cor=TRUE)
biplot(p1)
```



El primer componente es muy parecido al de las mujeres, es un componente que mide el tamaño del individuo. Tiene peso positivo de todas las variables. En el gráfico de scores los individuos se sitúan de derecha a izquierda según su tamaño. Podemos comparar las medidas de tres individuos, el 124 muy a la derecha (grandes dimensiones), el 226 en el centro (medidas medias) y el 55 muy a la izquierda (pequeño).

```
dat[c(124,226,55),10:21]
```

	C_hombros	C_pecho	C_cintura	C_abdomen	C_cadera	C_muslo	C_biceps	C_brazo
## 124	134.8	118.7	105.2	105.0	115.5	69.9	39.4	32.1
## 226	117.3	103.5	86.1	90.5	96.7	55.2	34.5	27.7
## 55	103.3	88.8	73.3	77.9	85.7	46.9	30.5	24.8
	C_rodilla	C_gemelo	C_tobillo	C_muneca				
## 124	42.2	47.7	27.0	19.2				
## 226	37.3	34.8	22.5	16.4				
## 55	31.1	30.5	19.0	15.0				

La posición según el eje vertical está determinado por el segundo componente, con pesos positivos para las variables biceps, brazo, muñeca y hombros, y pesos negativos para cintura, abdomen, cadera y rodilla. Es un componente de contraste que compara las medidas asociadas a los hombros y brazos frente a las del tronco y rodilla. Los individuos en la parte superior de gráfico destacan por tener valores altos en biceps, brazo, muñeca y hombros y bajos en cintura, abdomen, cadera y rodilla. Los individuos en la parte inferior tienen las características contarias, valores altos en el tronco (cintura, abdomen, cadera y rodilla.) y bajos en las extremidades superiores (biceps, brazo, muñeca y hombros). Para ilustrar los dos extremos y un caso medio elegimos al 110, 226 y 198.

```
dat[c(110,226,198),c(10,16,17,21,12,13,14,18)]
```

	C_hombros	C_biceps	C_brazo	C_muneca	C_cintura	C_abdomen	C_cadera	C_rodilla
## 110	117.7	36.3	32.5	18.4	73.3	82.1	89.3	34.3
## 226	117.3	34.5	27.7	16.4	86.1	90.5	96.7	37.3
## 198	120.7	37.1	27.8	15.9	98.6	111.7	118.7	37.5

Hay que tener presente en esta interpretación que el componente 1 es mucho más importante que el 2, utilizando el valor propio en esta comparación, el primero tiene un valor propio igual a 7.5 y el segundo a 1.26.

Seoul 1988

Contents

Datos	1
Correlaciones	2
Componentes principales	4
Importancia de cada componentes	4
Resumen del análisis con dos componentes	6
Gráficos importantes	7
Solución con tres componentes	10

Datos

```
library(corrplot)

## corrplot 0.84 loaded

library(corrplot)

dat = read.table("seoul1988.txt",header=TRUE)

dat$T100 = 100/dat$T100
dat$T110 = 110/dat$T110
dat$T400 = 400/dat$T400
dat$T1500 = 1500/dat$T1500

dat1 = dat[,c(3,7,8,12,4,6,10,5,9,11)]
```

Hay tres tipos de pruebas: carreras, saltos y lanzamientos. Los saltos y lanzamientos están medidos en metros. Cuanto más alto es el valor, mejor es el atleta en esa prueba. Sin embargo, las carreras están

medidas en segundos. Cuanto más alto es el valor, peor es la actuación del atleta. Vamos a corregir este efecto transformando las variables de carreras en lugar de segundos en velocidades, metros por segundo.

De esta forma conseguimos que los atletas buenos tengan un valor alto en esta prueba y los atletas menos buenos, tengan valores bajos. Ahora todas las variables están medidas con un criterio común, cuanto más alto el valor, mejor es el atleta en esa prueba.

Correlaciones

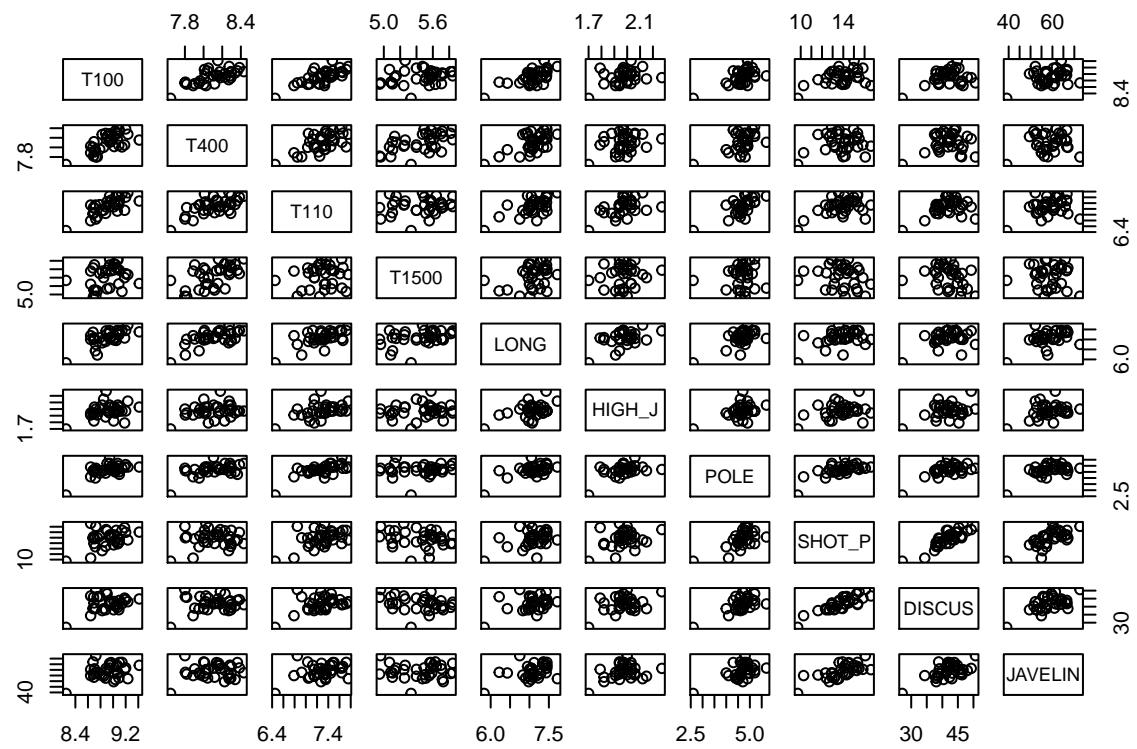
```
r = cor(dat1)
print(r,digits=3)
```

```
##          T100    T400    T110   T1500   LONG  HIGH_J   POLE  SHOT_P  DISCUS JAVELIN
## T100  1.000  0.682  0.736  0.247  0.682  0.350  0.608  0.410  0.338  0.331
## T400  0.682  1.000  0.638  0.557  0.625  0.266  0.510  0.131  0.146  0.140
## T110  0.736  0.638  1.000  0.151  0.642  0.475  0.689  0.473  0.381  0.335
## T1500 0.247  0.557  0.151  1.000  0.357  0.129  0.080 -0.198 -0.279 -0.034
## LONG  0.682  0.625  0.642  0.357  1.000  0.471  0.632  0.391  0.375  0.446
## HIGH_J 0.350  0.266  0.475  0.129  0.471  1.000  0.472  0.321  0.376  0.338
## POLE  0.608  0.510  0.689  0.080  0.632  0.472  1.000  0.643  0.620  0.557
## SHOT_P 0.410  0.131  0.473 -0.198  0.391  0.321  0.643  1.000  0.856  0.703
## DISCUS 0.338  0.146  0.381 -0.279  0.375  0.376  0.620  0.856  1.000  0.618
## JAVELIN 0.331  0.140  0.335 -0.034  0.446  0.338  0.557  0.703  0.618  1.000
```

Las correlaciones son “importantes” y positivas.

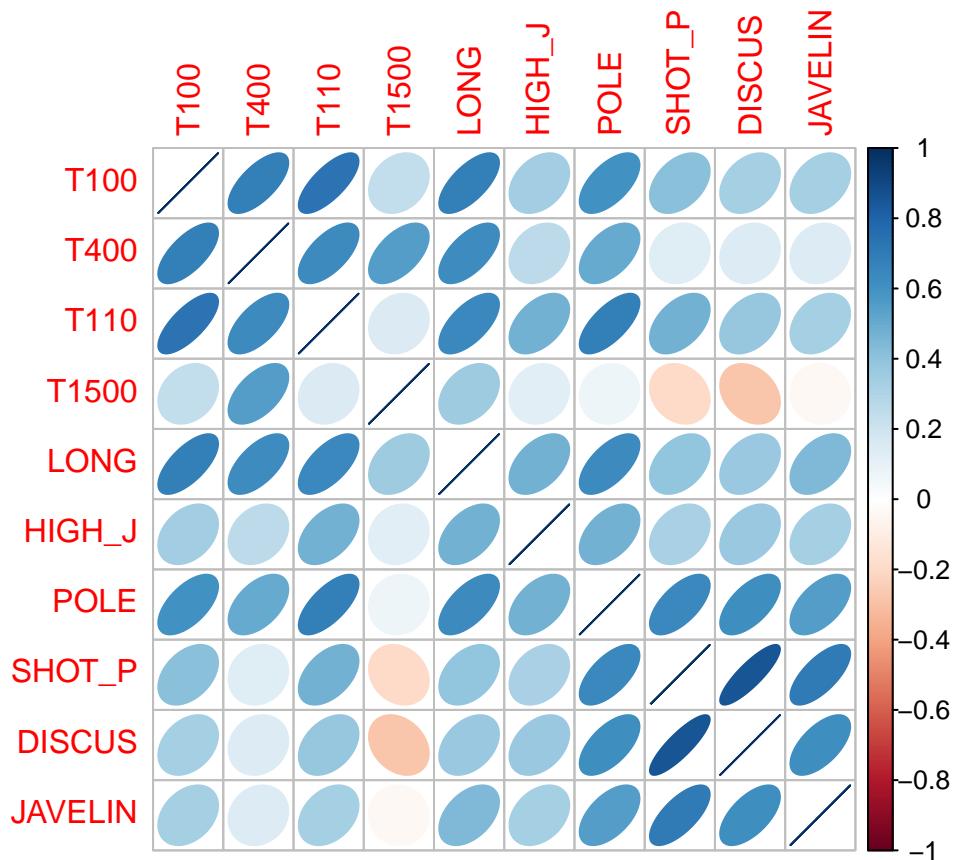
Vamos a ver que las relaciones son lineales:

```
pairs(dat1)
```



No hay desviaciones grandes a la condición de linealidad.

```
corrplot(r,method = "ellipse")
```



Casi todas las correlaciones son positivas. La prueba de 1500 tiene correlación negativas con los lanzamientos. 1500 es la prueba que menos relación tiene con las demás.

Las correlaciones más altas se dan en el grupo que incluye carreras, salto de longitud y pértiga, por un lado. Por otro lado en las tres pruebas de lanzamiento. Pértiga tiene correlaciones altas con los dos grupos.

Componentes principales

```
m = princomp(dat1, cor=TRUE)
names(m)
```

```
## [1] "sdev"      "loadings"   "center"     "scale"      "n.obs"     "scores"    "call"
```

Importancia de cada componentes

Se han calculado los seis componentes y están ordenados por orden de importancia. `m$sdev` contienen las desviaciones típicas de cada componente, de mayor a menor.

```
m
```

```
## Call:
## princomp(x = dat1, cor = TRUE)
##
```

```

## Standard deviations:
##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8
## 2.2270806 1.4428416 0.8629682 0.8387143 0.6200132 0.5547362 0.5408903 0.4791760
##   Comp.9   Comp.10
## 0.4609617 0.2887180
##
## 10 variables and 34 observations.

```

```
m$sdev # desviaciones de cada componente
```

```

##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8
## 2.2270806 1.4428416 0.8629682 0.8387143 0.6200132 0.5547362 0.5408903 0.4791760
##   Comp.9   Comp.10
## 0.4609617 0.2887180

```

```
m$sdev^2 # varianzas de cada componente (suman 7)
```

```

##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8
## 4.9598878 2.0817920 0.7447142 0.7034417 0.3844164 0.3077322 0.2925623 0.2296096
##   Comp.9   Comp.10
## 0.2124857 0.0833581

```

```
m$sdev^2/10*100# porcentaje que explica cada componente
```

```

##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8
## 49.598878 20.817920 7.447142 7.034417 3.844164 3.077322 2.925623 2.296096
##   Comp.9   Comp.10
## 2.124857 0.833581

```

```
cumsum(m$sdev^2/10*100) # porcentaje acumulado
```

```

##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8
## 49.59888 70.41680 77.86394 84.89836 88.74252 91.81984 94.74547 97.04156
##   Comp.9   Comp.10
## 99.16642 100.00000

```

```
m$loadings
```

```

##
## Loadings:
##   Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
## T100      0.355  0.208  0.344  0.128  0.310  0.166  0.482  0.192  0.541  0.104
## T400      0.291  0.432  0.214          -0.320          0.553 -0.399 -0.329
## T110      0.369  0.139  0.224  0.370          -0.430  0.114 -0.424 -0.437  0.292
## T1500     0.539 -0.335 -0.524 -0.308          0.200 -0.316  0.166  0.217
## LONG      0.364  0.196          0.431  0.576 -0.424 -0.285 -0.157
## HIGH_J    0.269          -0.773  0.505          0.167  0.154          -0.130
## POLE      0.391          -0.212 -0.365 -0.635          0.505
## SHOT_P    0.326 -0.395  0.102 -0.188 -0.224          0.306 -0.391          -0.627
## DISCUS    0.308 -0.416          -0.445  0.424          0.122 -0.117  0.564

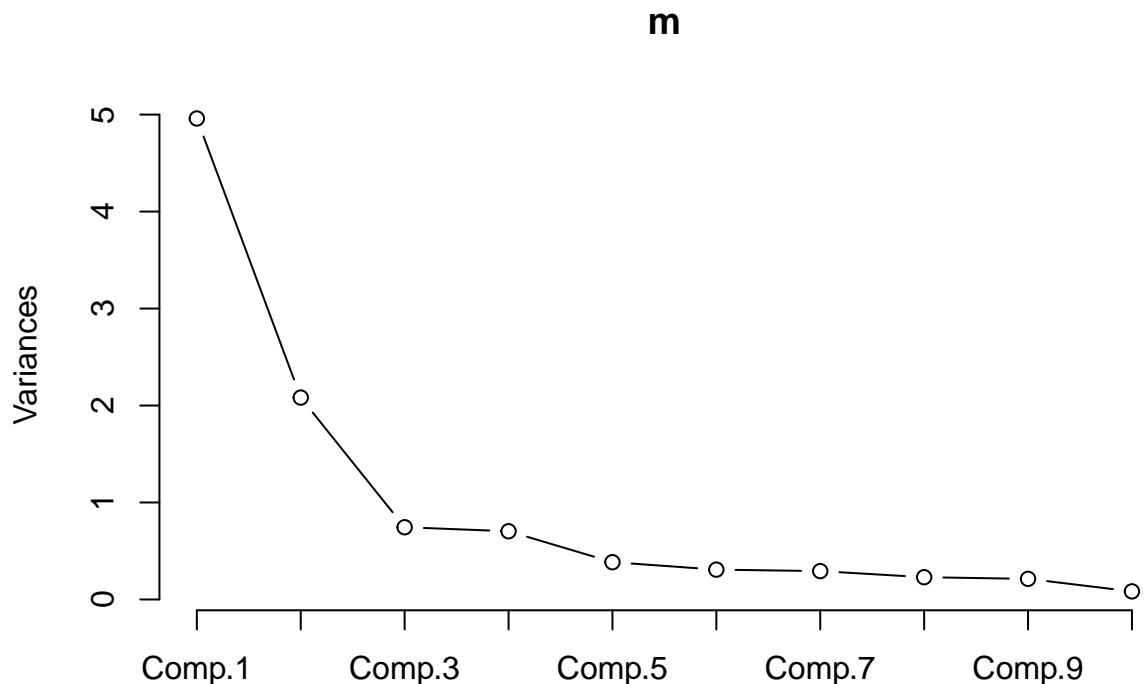
```

```

## JAVELIN  0.295 -0.299 -0.222 -0.513  0.470 -0.367          0.322 -0.182  0.109
##
##           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## SS loadings    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
## Proportion Var 0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
## Cumulative Var 0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9
##           Comp.10
## SS loadings     1.0
## Proportion Var 0.1
## Cumulative Var 1.0

plot(m,type="lines")

```



El gráfico indica que hay tres componentes importantes. Voy a interpretar la solución de dos y hablaré ligeramente del tercero. Si nos quedamos con dos componentes que explican el 70.4% de la variabilidad total. Con tres el 77.9%.

Resumen del análisis con dos componentes

```

source("prinfact.R")
sol = prinfact(dat1,2)
names(sol)

```

```

## [1] "loadings"  "variances" "scores"     "eig"

sol$loadings

##          Comp 1      Comp 2 communality uniqueness
## T100    0.7907982  0.300553407  0.7156941  0.2843059
## T400    0.6483989  0.623630957  0.8093367  0.1906633
## T110    0.8217556  0.200068702  0.7153097  0.2846903
## T1500   0.1916215  0.777046112  0.6405194  0.3594806
## LONG    0.8099560  0.283219050  0.7362418  0.2637582
## HIGH_J   0.5983358  0.003538419  0.3580182  0.6419818
## POLE    0.8700550 -0.089399516  0.7649879  0.2350121
## SHOT_P   0.7265789 -0.569560356  0.8523159  0.1476841
## DISCUS   0.6861732 -0.600460663  0.8313866  0.1686134
## JAVELIN  0.6575130 -0.430750615  0.6178694  0.3821306

```

```

sol$variances

##          Comp 1      Comp 2
## Variance 4.9598878 2.0817920
## Proportion 0.4959888 0.2081792
## Cumulative 0.4959888 0.7041680

```

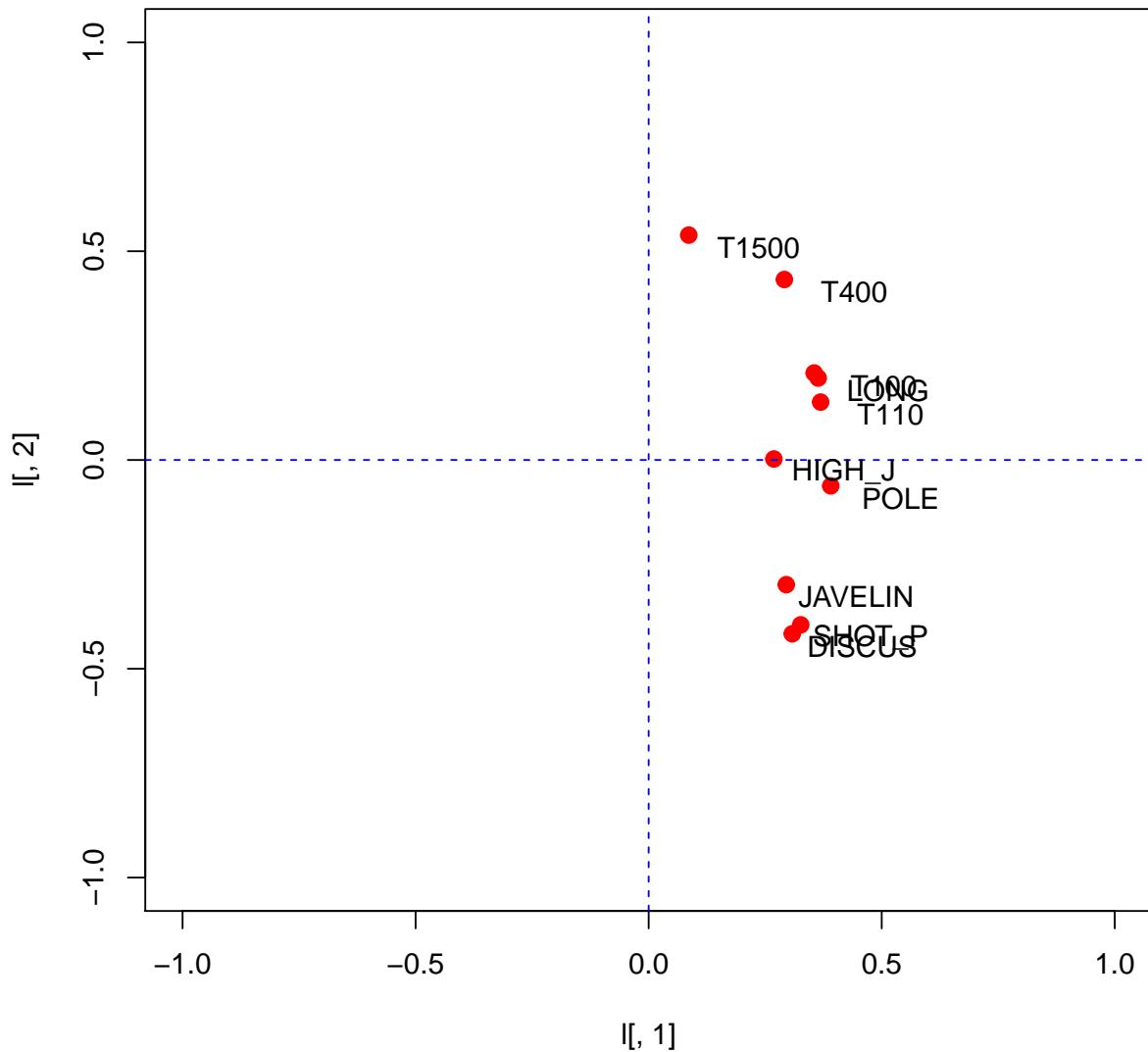
- Con la solución de dos componentes explicamos el 70.4% de la información total.
- No todas las variables originales están “igualmente” explicadas. Las mejores explicadas son SHOT_P (85.2%), DISCUS (83.1%) y T400 (80.9%).
- La peor explicada HIGH_J(35.8%). Según los datos, eso se interpreta como que es la prueba que menos tiene que ver con las demás.

Gráficos importantes

```

l = loadings(m)
plot(l[,1],l[,2],col="red",cex=1.2,
      xlim=c(-1,1), ylim=c(-1,1),pch=19)
abline(h=0,col="blue",lty=2)
abline(v=0,col="blue",lty=2)
text(l[,1]+.15,l[,2]-.03,colnames(dat1))

```

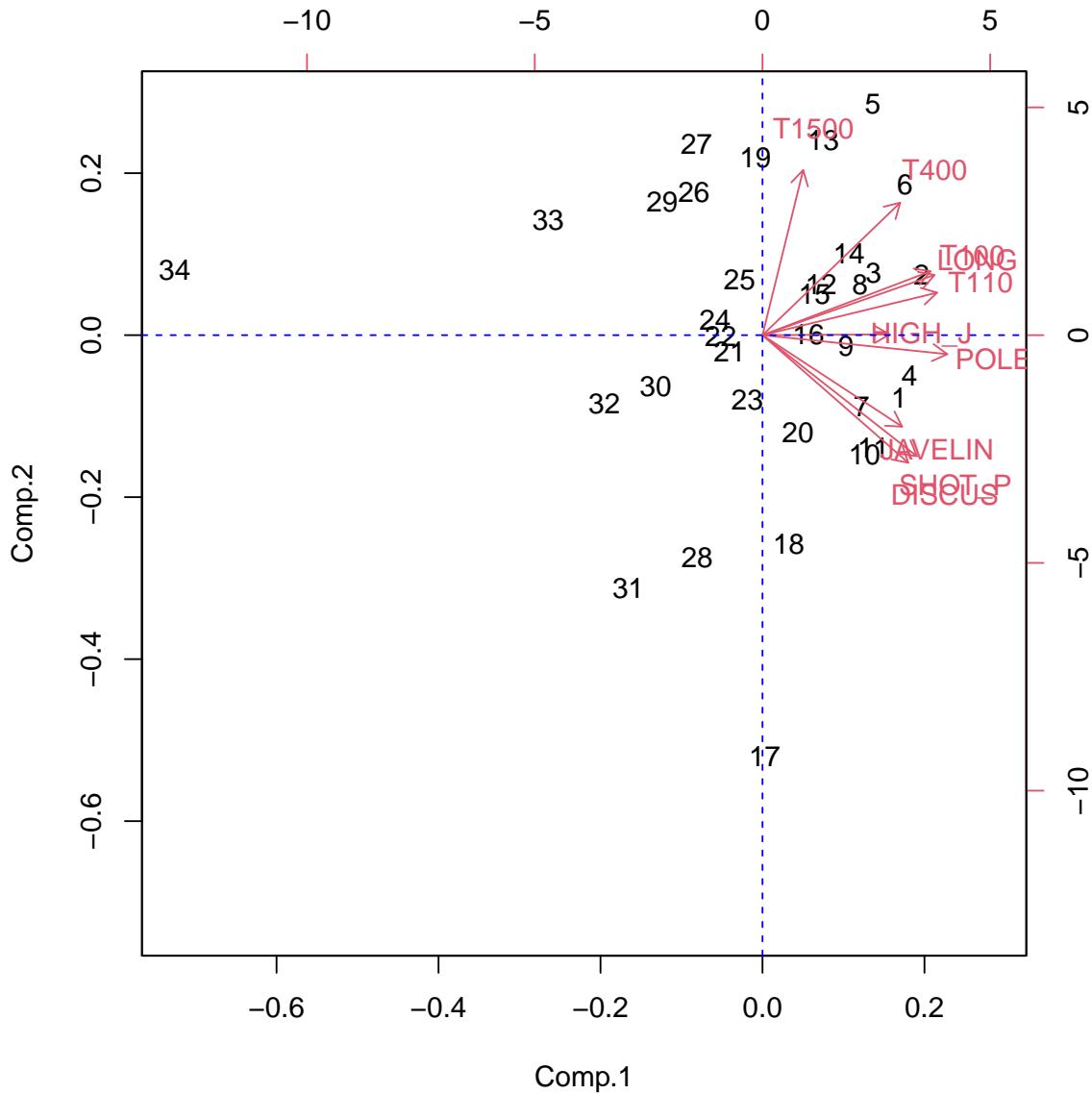


Esta figura muestra que las pruebas se agrupan de la siguiente forma:

- abajo: Están los tres lanzamientos. Son variables muy correlacionadas entre sí y que tienen comportamiento similar respecto al resto de las variables. Los atletas buenos en una son buenos en las otras dos.

- en medio: Lo mismo se puede decir de T100, T110 y LONG.
- arriba: T1500 se parece a T400, y las dos difieren bastante de las demás.
- HIGH_J y POLE son pruebas especiales

```
biplot(m)
abline(h=0,col="blue",lty=2)
abline(v=0,col="blue",lty=2)
```



La posición según eje x nos ordena los atletas de mejor a peor. El número es el orden en el archivo, que coincide con la posición que obtuvo el atleta en las Olimpiadas del 88. El atleta 34 es claramente inferior al resto de los participantes.

La posición en el eje "y" contrapone a dos tipos de atletas. Arriba son atletas que obtienen buenas puntuaciones en carreras y longitud y malas en lanzamientos. El 5, 13, 19 y 27 responden mejor a pruebas relacionadas con las "piernas" y peor a la que tiene que ver con los "brazos". El atleta 17 es un caso extremo, debe ser muy bueno en los lanzamientos y malo en las carreras (1500 y 400). En el centro está Antonio Peñalver (23) el representante español.

Solución con tres componentes

```
sol3 = prinfact(dat1,3)
sol3$loadings
```



```
##          Comp 1      Comp 2      Comp 3 communality uniqueness
## T100    0.7907982  0.300553407  0.29653534   0.8036273  0.1963727
## T400    0.6483989  0.623630957  0.18474832   0.8434687  0.1565313
## T110    0.8217556  0.200068702  0.19342527   0.7527231  0.2472769
## T1500   0.1916215  0.777046112 -0.28932941   0.7242310  0.2757690
## LONG    0.8099560  0.283219050 -0.08553478   0.7435580  0.2564420
## HIGH_J   0.5983358  0.003538419 -0.66744054   0.8034951  0.1965049
## POLE    0.8700550 -0.089399516  0.05907183   0.7684774  0.2315226
## SHOT_P   0.7265789 -0.569560356  0.08776848   0.8600192  0.1399808
## DISCUS   0.6861732 -0.600460663  0.03118783   0.8323593  0.1676407
## JAVELIN  0.6575130 -0.430750615 -0.19122142   0.6544350  0.3455650
```

```
sol3$variances
```

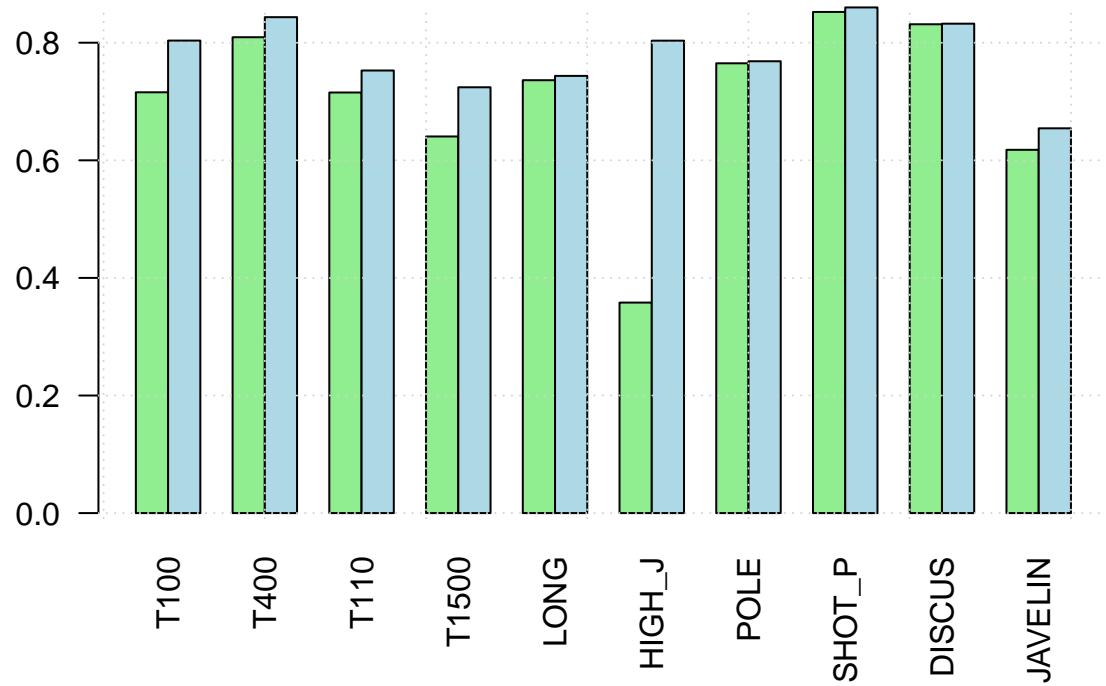
```
##          Comp 1      Comp 2      Comp 3
## Variance  4.9598878  2.0817920  0.74471417
## Proportion 0.4959888  0.2081792  0.07447142
## Cumulative 0.4959888  0.7041680  0.77863940
```

Si incluimos tres componentes explicamos el 77.8% de la variabilidad total.

Se observa que el tercer componente tiene carga (y peso) especialmente grande para HIGH_J, que era la variable peor explicada con la solución de 2 componentes.

El porcentaje de variabilidad explicada para cada variable original con la solución de 3 componentes es muy similar a la explicada con 2 componentes, excepto para la variable HIGH_J. En la figura inferior se puede observar lo anterior. Todas las variables originales están mejor explicadas con la solución de tres componentes (nunca se puede empeorar al añadir un componente), pero el **salto** importante se produce en HIGH_J.

```
barplot(rbind(sol$loadings[,3],sol3$loadings[,4]),
           beside=TRUE,col=c("lightgreen","lightblue"),las=2)
grid()
```



Análisis de Datos

Tema 4

ANÁLISIS DISCRIMINATE

1

Clasificación

- El modelo estrella de Estadística es el modelo de regresión que consiste en explicar la relación entre una variable respuesta **Y** (cuantitativa, el consumo de un coche) en función de una serie de variables explicativas X_1, X_2, \dots, X_k (regresores) que pueden ser variables cuantitativas o cualitativas (peso, númer. de cilindros, origen del coche, potencia, ...)
- En el análisis discriminante el problema es similar, pero en este caso, la variable respuesta **Y** es **cualitativa o categórica**. Predecir una respuesta cualitativa a partir de una serie de variables explicativas habitualmente se denomina **clasificar** una observación.
- Existen muchas **técnicas de clasificación**: análisis discriminante, regresión logística, árboles de clasificación, random forest, redes neuronales, support vector machine.

2

Problema de clasificación

Variables explicativas o regresores

Respuesta

X_1	X_2	...	X_k	G
X_{11}	X_{21}	...	X_{k1}	G_1
X_{12}	X_{22}	...	X_{k2}	G_2
...
X_{1i}	X_{2i}	...	X_{ki}	G_i
...
X_{1n}	X_{2n}	...	X_{kn}	G_n

$$G_i = \begin{cases} A \\ B \end{cases} \text{ Dos grupos}$$

$$G_i = \begin{cases} A \\ B \\ \vdots \\ K \end{cases} \text{ K grupos}$$

3

Ejemplo 1: dos grupos

Respuesta

Variables explicativas o regresores

	Altura	Peso	Pie	L_Brazo	A_Espal	D_Cabeza	Rod_Tob
Mujer	159,0	49,0	36,0	68,5	42,0	57,0	40,0
Mujer	172,0	65,0	38,0	75,0	48,0	58,0	44,0
Mujer	167,0	52,0	37,0	73,0	41,5	58,0	44,0
Mujer	164,0	51,0	36,0	71,0	44,5	54,0	40,0
Mujer	161,0	67,0	38,0	71,0	44,0	56,0	42,0
Mujer	168,0	48,0	39,0	72,5	41,0	54,5	43,0
Mujer	158,0	50,0	36,0	68,5	44,0	57,0	41,0
Mujer	156,0	65,0	36,0	68,0	46,0	58,0	41,0
Mujer	158,0	43,0	36,0	68,0	43,0	55,0	39,0
Mujer	162,0	68,0	39,0	72,0	44,0	59,0	42,0
Mujer	156,0	52,0	36,0	67,0	36,0	56,0	41,0
Mujer	152,0	45,0	34,0	66,0	40,0	55,0	38,0
Mujer	155,0	53,0	36,0	67,0	43,0	56,0	38,0
Mujer	170,0	70,0	38,0	73,0	45,0	56,0	43,0
Mujer	168,0	56,0	37,5	70,5	48,0	60,0	40,0
Hombre	164,0	62,0	39,0	73,0	44,0	55,0	44,0
Hombre	181,0	74,0	43,0	74,0	50,0	60,0	47,0
Hombre	183,0	74,0	41,0	79,0	47,5	59,5	47,0
Hombre	173,0	64,0	40,0	79,0	48,0	56,5	47,0
Hombre	178,0	74,0	42,0	75,0	50,0	59,0	45,0
Hombre	181,0	76,0	43,0	83,0	51,0	57,0	43,0
Hombre	182,5	91,0	41,0	83,0	53,0	59,0	43,0
Hombre	176,0	73,0	42,0	78,0	48,0	58,0	45,0
Hombre	181,0	80,0	43,0	76,0	49,0	57,0	46,0
Hombre	173,0	69,0	41,0	74,0	48,0	56,0	44,0
Hombre	189,0	87,0	45,0	82,0	53,0	61,0	52,0
Hombre	170,0	67,0	40,0	77,0	46,5	58,0	44,5

Ejemplo 2: Lirios (3 grupos)

1.- Virginica

2.- Versicolor

3.- Setosa

	Grupo	X1	X2	X3	X4
1	1	5,1	3,5	1,4	0,2
2	1	4,9	3,0	1,4	0,2
3	1	4,7	3,2	1,3	0,2
4	1	4,6	3,1	1,5	0,2
5	1	5,0	3,6	1,4	0,2
:	:	:	:	:	:
50	1	5,0	3,3	1,4	0,2
1	2	7,0	3,2	4,7	1,4
2	2	6,4	3,2	4,5	1,5
3	2	6,9	3,1	4,9	1,5
4	2	5,5	2,3	4,0	1,3
5	2	6,5	2,8	4,6	1,5
:	:	:	:	:	:
50	2	5,7	2,8	4,1	1,3
1	3	6,3	3,3	6,0	2,5
2	3	5,8	2,7	5,1	1,9
3	3	7,1	3,0	5,9	2,1
4	3	6,3	2,9	5,6	1,8
5	3	6,5	3,0	5,8	2,2
:	:	:	:	:	:
50	3	5,9	3,0	5,1	1,8

X_1 : Longitud sépalo

X_2 : Anchura sépalo

X_3 : Longitud pétalo

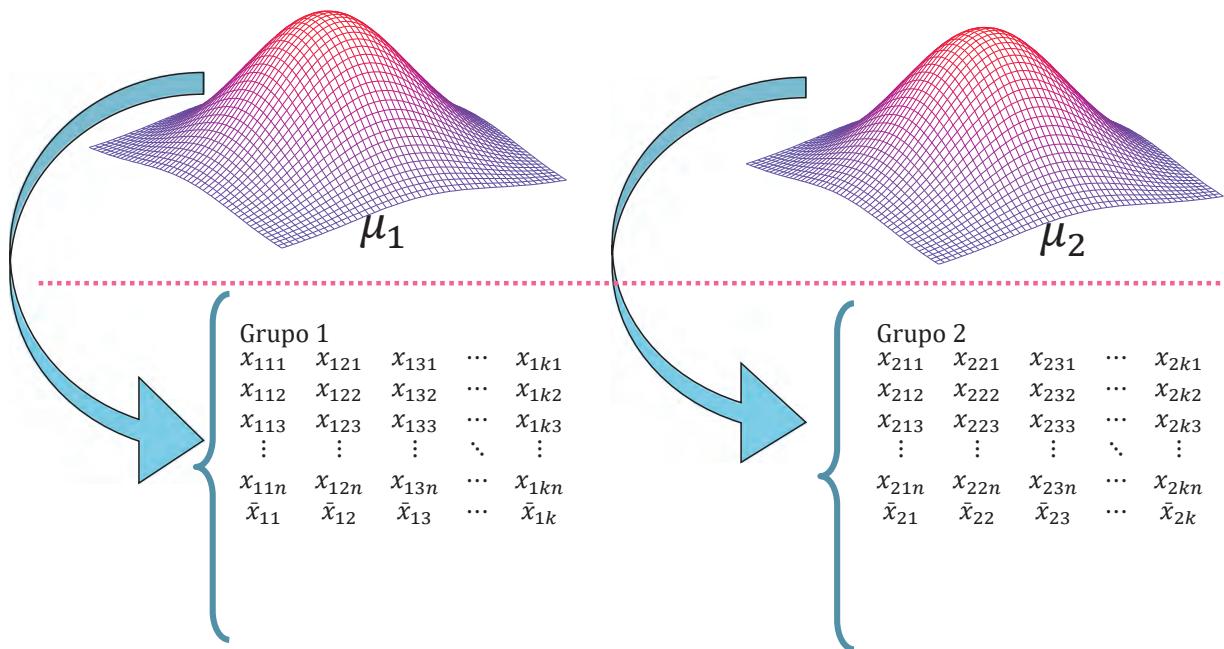
X_4 : Anchura pétalo

Lección 4 : Linear discriminant analysis (Ida)

1. DOS GRUPOS

Dos grupos

MODELO DATOS



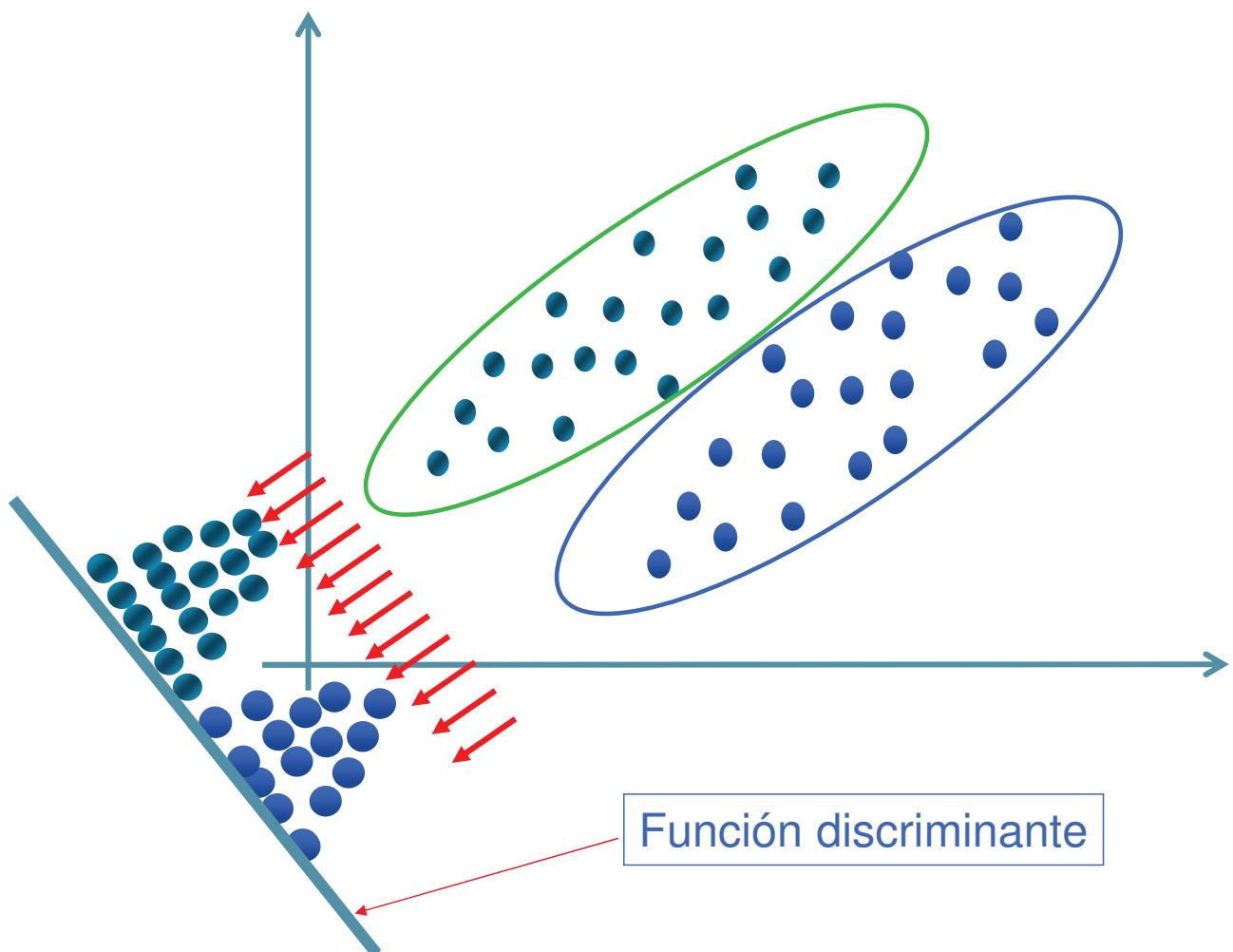
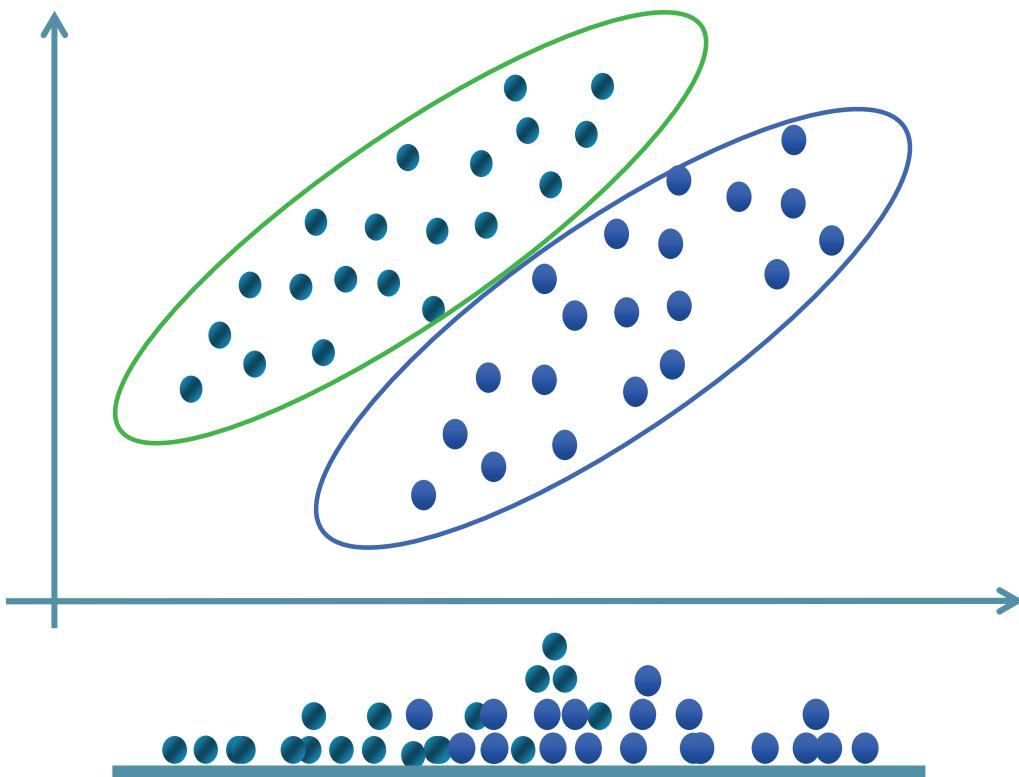
Objetivos del A. Discriminante

- Contrastar si los grupos son significativamente distintos
- Obtener la función discriminante Y que mejor separa los grupos

$$Y_i = a_0 + a_1 X_{1i} + a_2 X_{2i} + \cdots + a_k X_{ki}$$

- Definir una regla para clasificar nuevas observaciones
- Indicar qué variables son las que tienen más poder de discriminación.

Interpretación geométrica



Hipótesis del modelo

- Normalidad multivariante
- Homocedasticidad: misma matriz de varianzas
- Independencia

Estas condiciones son: (1) muy exigentes, (2) difíciles de verificar, (3) rara vez se cumplen.

A pesar de ello, la experiencia demuestra, que el modelo proporciona muy buenos resultados.

Puntuación discriminante: y_i

Grupo 1

$$\begin{matrix} x_{111} & x_{121} & x_{131} & \cdots & x_{1k1} \\ x_{112} & x_{122} & x_{132} & \cdots & x_{1k2} \\ x_{113} & x_{123} & x_{133} & \cdots & x_{1k3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{11n} & x_{12n} & x_{13n} & \cdots & x_{1kn} \\ \bar{x}_{11} & \bar{x}_{12} & \bar{x}_{13} & \cdots & \bar{x}_{1k} \end{matrix}$$

$$\mathbf{S}_1 = \begin{pmatrix} s_1^2 & s_{12} & \cdots & s_{1k} \\ s_{12} & s_2^2 & \cdots & s_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1k} & s_{2k} & \cdots & s_k^2 \end{pmatrix}$$

→ $\bar{\mathbf{X}}_1$

Grupo 2

$$\begin{matrix} x_{211} & x_{221} & x_{231} & \cdots & x_{2k1} \\ x_{212} & x_{222} & x_{232} & \cdots & x_{2k2} \\ x_{213} & x_{223} & x_{233} & \cdots & x_{2k3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{21n} & x_{22n} & x_{23n} & \cdots & x_{2kn} \\ \bar{x}_{21} & \bar{x}_{22} & \bar{x}_{23} & \cdots & \bar{x}_{2k} \end{matrix}$$

$$\mathbf{S}_2 = \begin{pmatrix} s'^2_1 & s'_{12} & \cdots & s'_{1k} \\ s'_{12} & s'^2_2 & \cdots & s'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s'_{1k} & s'_{2k} & \cdots & s'^2_k \end{pmatrix}$$

→ $\bar{\mathbf{X}}_2$

$$\mathbf{S}_T = \frac{n}{n+m} \mathbf{S}_1 + \frac{m}{n+m} \mathbf{S}_2$$

$$\mathbf{a} = \mathbf{S}_T^{-1} (\bar{\mathbf{X}} - \bar{\mathbf{Y}})$$

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_k \end{pmatrix}$$

$$Y_i = a_0 + a_1 x_{1i} + a_2 x_{2i} + \cdots + a_k x_{ki}$$

$$y_i = a_0 + a_1 x_{1i} + a_2 x_{2i} + \cdots + a_k x_{ki}$$

a_0 hace que media de y_i sea cero

Análisis Discriminante: lda()

```
library(haven)
dat <- read_sav("data/Consejo 1994.sav")
dat = as.data.frame(dat)
dat$SEXO = factor(dat$SEXO, labels=c("Mujer", "Hombre"))
pos = order(dat$SEXO)
dat = dat[pos,]
```

Ejemplo 1: Datos

	SEXO	Variables explicativas o regresores						
		Altura	Peso	Pie	L_Brazo	A_Espal	D_Cabeza	Rod_Tob
Mujer		159,0	49,0	36,0	68,5	42,0	57,0	40,0
Mujer		172,0	65,0	38,0	75,0	48,0	58,0	44,0
Mujer		167,0	52,0	37,0	73,0	41,5	58,0	44,0
Mujer		164,0	51,0	36,0	71,0	44,5	54,0	40,0
Mujer		161,0	67,0	38,0	71,0	44,0	56,0	42,0
Mujer		168,0	48,0	39,0	72,5	41,0	54,5	43,0
Mujer		158,0	50,0	36,0	68,5	44,0	57,0	41,0
Mujer		156,0	65,0	36,0	68,0	46,0	58,0	41,0
Mujer		158,0	43,0	36,0	68,0	43,0	55,0	39,0
Mujer		162,0	68,0	39,0	72,0	44,0	59,0	42,0
Mujer		156,0	52,0	36,0	67,0	36,0	56,0	41,0
Mujer		152,0	45,0	34,0	66,0	40,0	55,0	38,0
Mujer		155,0	53,0	36,0	67,0	43,0	56,0	38,0
Mujer		170,0	70,0	38,0	73,0	45,0	56,0	43,0
Mujer		168,0	56,0	37,5	70,5	48,0	60,0	40,0
Hombre		164,0	62,0	39,0	73,0	44,0	55,0	44,0
Hombre		181,0	74,0	43,0	74,0	50,0	60,0	47,0
Hombre		183,0	74,0	41,0	79,0	47,5	59,5	47,0
Hombre		173,0	64,0	40,0	79,0	48,0	56,5	47,0
Hombre		178,0	74,0	42,0	75,0	50,0	59,0	45,0
Hombre		181,0	76,0	43,0	83,0	51,0	57,0	43,0
Hombre		182,5	91,0	41,0	83,0	53,0	59,0	43,0
Hombre		176,0	73,0	42,0	78,0	48,0	58,0	45,0
Hombre		181,0	80,0	43,0	76,0	49,0	57,0	46,0
Hombre		173,0	69,0	41,0	74,0	48,0	56,0	44,0
Hombre		189,0	87,0	45,0	82,0	53,0	61,0	52,0
Hombre		170,0	67,0	40,0	77,0	46,5	58,0	44,5

Modelo lda()

```
library(MASS)
m1 = lda(SEXO ~ ., data = dat)
m1

## Call:
## lda(SEXO ~ ., data = dat)
##
## Prior probabilities of groups:
## Mujer Hombre
## 0.556 0.444
##
## Group means:
##          ESTATURA PESO  PIE L_BRAZO A_ESPALD D_CRÁNEO L_ROXTO
## Mujer      162 55.6 36.8    70.1    43.3    56.6    41.1
## Hombre     178 74.2 41.7    77.8    49.0    58.0    45.6
##
## Coefficients of linear discriminants:
##           LD1
## ESTATURA -0.08655
## PESO      -0.00504
## PIE       0.64480
## L_BRAZO   0.13579
## A_ESPALD  0.12237
## D_CRÁNEO  -0.20358
## L_ROXTO   0.09870
```

Modelo estimado

$$y_i = -18.4 - .086ESTA_i - .0050PESO + .644PIE_i + .135L_{BRA}i + .122 A_{ESP}i - .203D_{CRAN}i + .098LRXT_i$$

NOTAS:

- (1) a_0 se calcula para que la media de y_i sea cero (No es importante)

$$a_0 = .086\bar{x}_1 + .0050\bar{x}_2 - .644\bar{x}_3 + \dots - .098\bar{x}_7 = 18.4$$

- (2) Las puntuaciones y_i (scores) de hombres son muy diferentes a las de las mujeres (ese es el objetivo).
- (3) Los coeficientes a_i (pesos) son difíciles de interpretar y comparar (están en unidades distintas), los analizaremos más adelante

Función discriminante

	Altura	Peso	Pie	L_Brazo	A_Espal	D_Cabeza	Rod_Tob	Discrim_1	Medias
Mujer	159,0	49,0	36,0	68,5	42,0	57,0	40,0	-2,40	
Mujer	172,0	65,0	38,0	75,0	48,0	58,0	44,0	-0,51	
Mujer	167,0	52,0	37,0	73,0	41,5	58,0	44,0	-1,73	
Mujer	164,0	51,0	36,0	71,0	44,5	54,0	40,0	-1,59	
Mujer	161,0	67,0	38,0	71,0	44,0	56,0	42,0	-0,39	
Mujer	168,0	48,0	39,0	72,5	41,0	54,5	43,0	-0,02	
Mujer	158,0	50,0	36,0	68,5	44,0	57,0	41,0	-1,98	
Mujer	156,0	65,0	36,0	68,0	46,0	58,0	41,0	-1,91	
Mujer	158,0	43,0	36,0	68,0	43,0	55,0	39,0	-1,92	
Mujer	162,0	68,0	39,0	72,0	44,0	59,0	42,0	-0,31	
Mujer	156,0	52,0	36,0	67,0	36,0	56,0	41,0	-2,79	
Mujer	152,0	45,0	34,0	66,0	40,0	55,0	38,0	-3,44	
Mujer	155,0	53,0	36,0	67,0	43,0	56,0	38,0	-2,15	
Mujer	170,0	70,0	38,0	73,0	45,0	56,0	43,0	-0,69	
Mujer	168,0	56,0	37,5	70,5	48,0	60,0	40,0	-1,86	
Hombre	164,0	62,0	39,0	73,0	44,0	55,0	44,0	0,69	
Hombre	181,0	74,0	43,0	74,0	50,0	60,0	47,0	1,89	
Hombre	183,0	74,0	41,0	79,0	47,5	59,5	47,0	0,90	
Hombre	173,0	64,0	40,0	79,0	48,0	56,5	47,0	1,84	
Hombre	178,0	74,0	42,0	75,0	50,0	59,0	45,0	1,64	
Hombre	181,0	76,0	43,0	83,0	51,0	57,0	43,0	3,44	
Hombre	182,5	91,0	41,0	83,0	53,0	59,0	43,0	1,78	
Hombre	176,0	73,0	42,0	78,0	48,0	58,0	45,0	2,19	
Hombre	181,0	80,0	43,0	76,0	49,0	57,0	46,0	2,52	
Hombre	173,0	69,0	41,0	74,0	48,0	56,0	44,0	1,59	
Hombre	189,0	87,0	45,0	82,0	53,0	61,0	52,0	4,16	
Hombre	170,0	67,0	40,0	77,0	46,5	58,0	44,5	1,08	
								0,00	

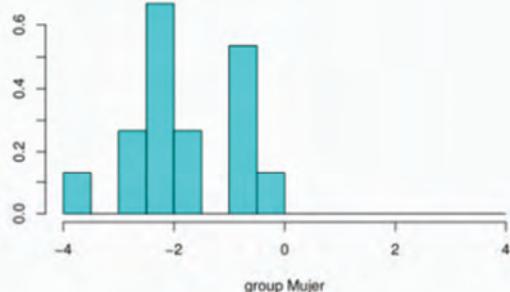
Centroide
Mujer

Centroide
Hombre

Gráfico

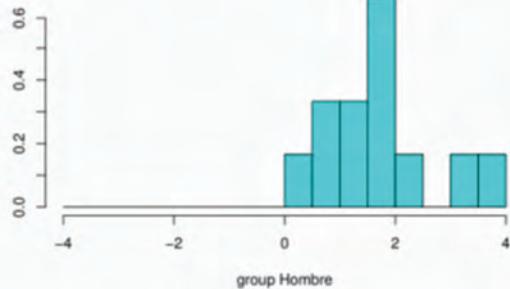
SCORES

plot(m1)



m2\$centroids

[1] -1.58 1.98



Método 1: Clasificación de una observación

- Dado una observación (conocemos las x 's)

$$U = (u_1, u_2, \dots, u_k)$$

- Calculamos su SCORE

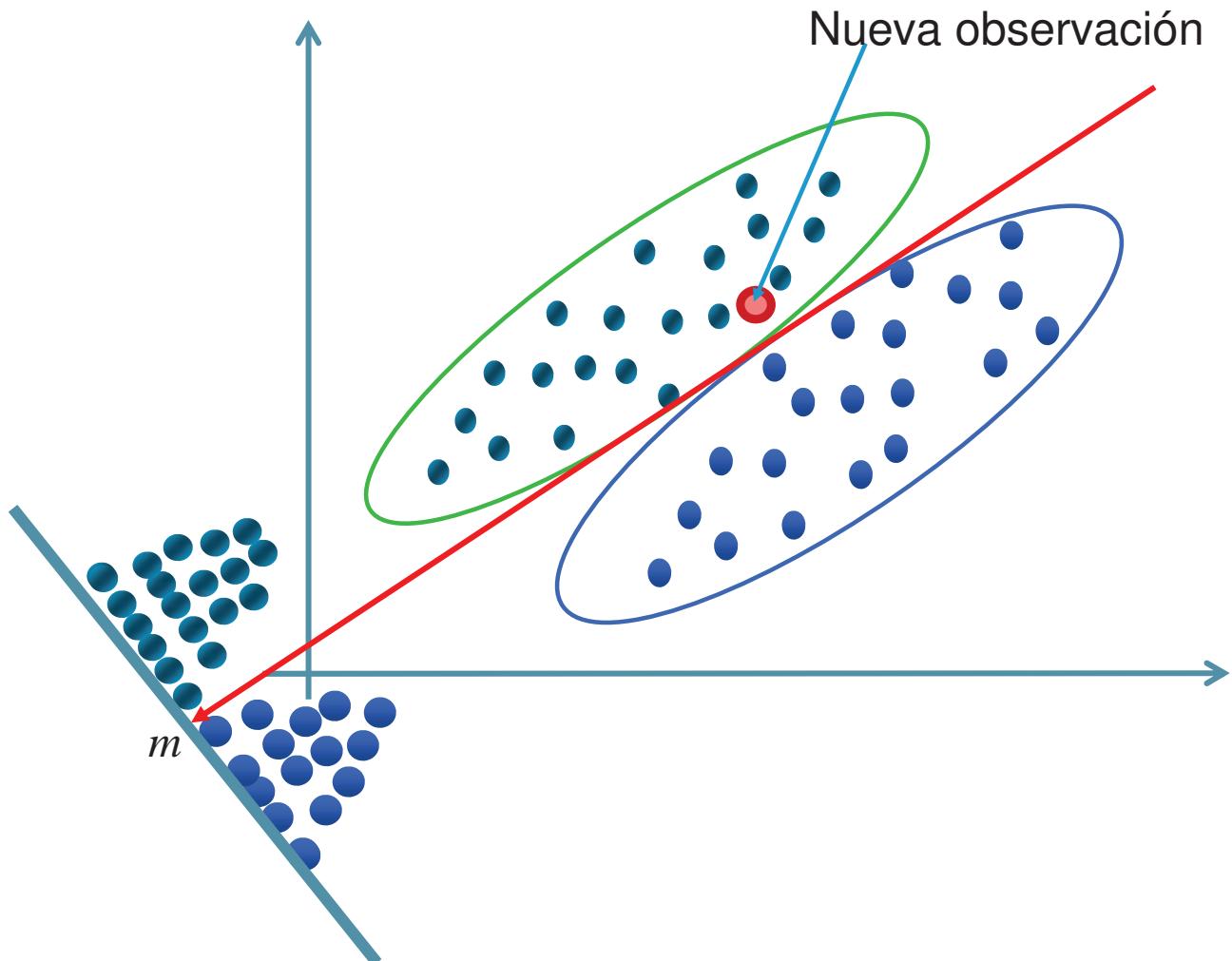
$$y_U = a_0 + a_1 u_1 + a_2 u_2 + \dots + a_k u_k$$

- Comparamos con la media de la función discriminante para cada grupo (centroides).

$$\bar{y}_1, \quad \bar{y}_2$$

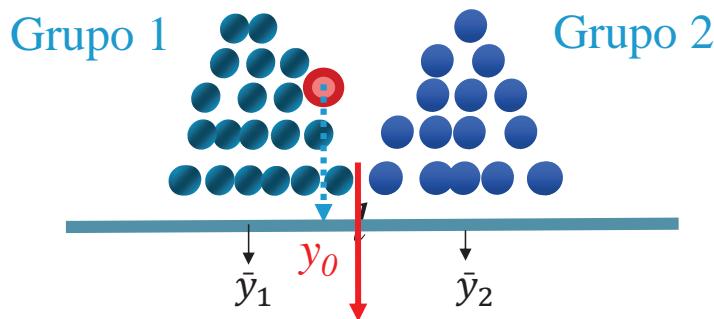
- Se asigna al más próximo:

$$d_1 = (\bar{y}_1 - y_0)^2, \quad d_2 = (\bar{y}_2 - y_0)^2,$$



$$l = \frac{m\bar{y}_1 + n\bar{y}_2}{n + m}$$

$$\bar{y}_1 < \bar{y}_2 \Rightarrow \begin{cases} y_0 < l \Rightarrow \text{grupo 1} \\ y_0 > l \Rightarrow \text{grupo 2} \end{cases}$$



Método 2: Clasificación de una observación

Probabilidad *a priori*

π_k = proporción de obs. en la clase k

Probabilidad de observar x en la clase k

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}$$

Bayes: Probabilidad de ser de la clase k si se ha observado x (*a posteriori*)

$$Pr(G = k | X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^K f_l(x)\pi_l}$$

Análisis Discriminante: predict()

```
pred_sex = predict(m1)  
names(pred_sex)
```

```
## [1] "class"      "posterior"   "x"
```

Resultados predict()

```
dat$PSEXO = pred_sex$class  
dat$PPROB = round(pred_sex$posterior, 3)  
dat$SCORE = pred_sex$x
```

SEXO	ESTATURA	PESO	PIE	L_BRAZO	A_ESPALD	D_CRÁNEO	L_ROXTO	PSEXO	PPROB["Mujer"]	PPROB["Hombre"]	SCORE["L01"]
1 Mujer	159	49	36.0	68.5	42.0	57.0	40.0	Mujer	1.000	0.000	-2.4035
3 Mujer	172	65	38.0	75.0	48.0	58.0	44.0	Mujer	0.940	0.060	-0.5117
4 Mujer	167	52	37.0	73.0	41.5	58.0	44.0	Mujer	0.999	0.001	-1.7252
5 Mujer	164	51	36.0	71.0	44.5	54.0	40.0	Mujer	0.999	0.001	-1.3902
6 Mujer	161	67	38.0	71.0	44.0	56.0	42.0	Mujer	0.911	0.089	-0.3926
7 Mujer	168	48	39.0	72.5	41.0	54.5	43.0	Mujer	0.728	0.272	-0.0172
10 Mujer	158	50	36.0	68.5	44.0	57.0	41.0	Mujer	1.000	0.000	-1.9786
11 Mujer	156	65	36.0	68.0	46.0	58.0	41.0	Mujer	1.000	0.000	-1.9078
13 Mujer	158	43	36.0	68.0	43.0	55.0	39.0	Mujer	1.000	0.000	-1.9238
18 Mujer	162	68	39.0	72.0	44.0	59.0	42.0	Mujer	0.885	0.115	-0.3143
19 Mujer	156	52	36.0	67.0	36.0	56.0	41.0	Mujer	1.000	0.000	-2.7946
20 Mujer	152	45	34.0	66.0	40.0	55.0	38.0	Mujer	1.000	0.000	-3.4416
23 Mujer	155	53	36.0	67.0	43.0	56.0	38.0	Mujer	1.000	0.000	-2.1526
25 Mujer	170	70	38.0	73.0	45.0	56.0	43.0	Mujer	0.967	0.033	-0.6940
27 Mujer	168	56	37.5	70.5	48.0	60.0	40.0	Mujer	0.999	0.001	-1.8553
2 Hombre	164	62	39.0	73.0	44.0	55.0	44.0	Hombre	0.178	0.822	0.6903
8 Hombre	181	74	43.0	74.0	50.0	60.0	47.0	Hombre	0.063	0.997	1.8859
9 Hombre	183	74	41.0	79.0	47.5	59.5	47.0	Hombre	0.094	0.906	0.8980
12 Hombre	173	64	40.0	79.0	48.0	56.5	47.0	Hombre	0.004	0.996	1.8410
14 Hombre	178	74	42.0	75.0	50.0	59.0	45.0	Hombre	0.007	0.993	1.6427
15 Hombre	181	76	43.0	83.0	51.0	57.0	43.0	Hombre	0.000	1.000	3.4362
16 Hombre	182	91	41.0	83.0	53.0	59.0	43.0	Hombre	0.005	0.995	1.7787
17 Hombre	176	73	42.0	78.0	48.0	58.0	45.0	Hombre	0.001	0.999	2.1870
21 Hombre	181	80	43.0	76.0	49.0	57.0	46.0	Hombre	0.000	1.000	2.5169
22 Hombre	173	69	41.0	74.0	48.0	56.0	44.0	Hombre	0.009	0.991	1.5874
24 Hombre	189	87	45.0	82.0	53.0	61.0	52.0	Hombre	0.000	1.000	4.1609
26 Hombre	170	67	40.0	77.0	46.5	58.0	44.5	Hombre	0.052	0.948	1.0783

Aciertos y fallos (Tabla de Confusión)

```
table(dat$SEXO,dat$PSEXO)
```

		Predicción	
		Mujer	Hombre
R E A L	Mujer	15	0
	Hombre	0	12

Análisis Discriminante: ldaPLUS()

```
library(multiUS)
m2 = ldaPlus(x = dat[,2:8], grouping = dat$SEXO)
names(m2)

## [1] "prior"                  "counts"
## [3] "means"                   "scaling"
## [5] "standCoefWithin"        "standCoefTotal"
## [7] "lev"                     "svd"
## [9] "N"                       "betweenGroupsWeights"
## [11] "call"                    "sigTest"
## [13] "eigModel"                "pred"
## [15] "centroids"               "corr"
## [17] "class"                   "classCV"
```

Función estandarizada

$$Y_i = a_1x_{1i} + a_2x_{2i} + \cdots + a_kx_{ki}$$

$$Y_i = a_1s_1\left(\frac{x_{1i}}{s_1}\right) + a_2s_2\left(\frac{x_{2i}}{s_2}\right) + \cdots + a_ks_k\left(\frac{x_{ki}}{s_k}\right)$$

Coeficientes estandarizados

$$(a_1s_1, \quad a_2s_2, \quad \cdots, \quad a_ks_k)$$

Coeficientes Estandarizados

```
m2$standCoefWithin
```

```
## LD1  
## ESTATURA -0.5562  
## PESO -0.0444  
## PIE 0.9779  
## L_BRAZO 0.4209  
## A_ESPALD 0.3515  
## D_CRÁNEO -0.3543  
## L_ROXTO 0.2167
```

```
## Coefficients of linear discriminants:  
## LD1  
## ESTATURA -0.08655  
## PESO -0.00504  
## PIE 0.64480  
## L_BRAZO 0.13579  
## A_ESPALD 0.12237  
## D_CRÁNEO -0.20358  
## L_ROXTO 0.09870
```



Modelo estimado (estandarizado)

$$y_i = -18.4 - .556ESTA'_i - .044PESO'_i + .977PIE'_i + .420LBRA'_i \\ + .351 A_ESP'_i - .354D'_{CRAN}_i + .216LRXT'_i$$

NOTAS: Sigue siendo difícil de interpretar

- (1) Cada variable x está dividida por su desviación típica ($x' = x/s_x$)
- (2) La variable que más discrimina es el PIE
- (3) Problemas de MULTICOLINEALIDAD (es difícil separar el efecto de cada variable)
- (4) Se interpreta como efectos marginales
- (5) Más adelante veremos si son significativas

29

Correlación

```
m2$corr
```

```
## LD1
## ESTATURA 0.695
## PESO 0.595
## PIE 0.896
## L_BRAZO 0.697
## A_ESPALD 0.555
## D_CRÁNEO 0.221
## L_ROXTO 0.584
```

- Nos permite comparar las correlaciones y ver el grado de relación de cada variable con la puntuación discriminante.
- Los coeficientes estandarizados de la función discriminante deben utilizarse para evaluar la importancia de la contribución única de cada variable independiente a la función discriminante.
- Una correlación baja (prox. a 0) indica que esa variable no es muy relevante. Puede ocurrir que el coeficiente estandarizado de la viable X en la función discriminante sea bajo y sin embargo el coeficiente de correlación de X con la función discriminante sea alto. Eso indica que X no añade información para discriminar (esa información está en el resto de las variables), aunque la variable X esté correlacionada con otras variables que sí son importantes en el modelo.

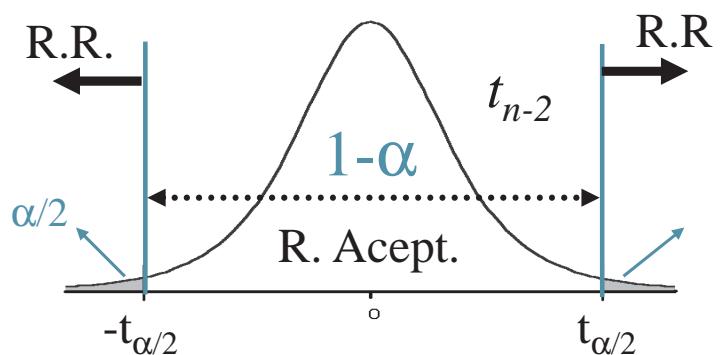
¿Son los Grupos Distintos?

Contraste de igualdad de medias

$$H_0: \mu_1 = \mu_2$$

$$H_1: \mu_1 \neq \mu_2$$

$$t_0 = \frac{\bar{y}_{1\bullet} - \bar{y}_{2\bullet}}{\hat{s}_R \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \rightarrow t_{n-2}$$

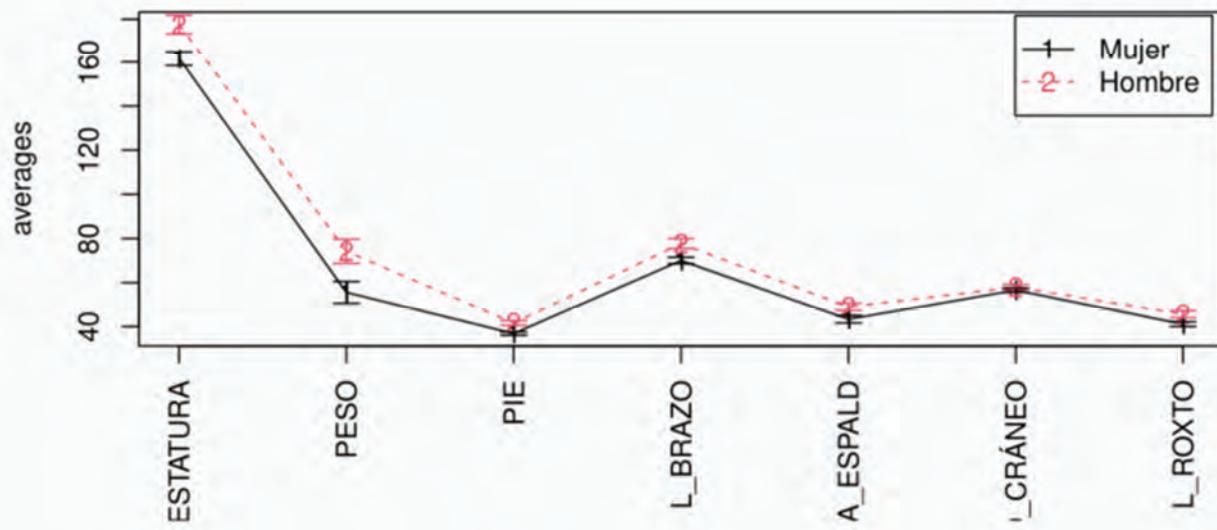


$|t_0| \leq t_{\alpha/2} \Rightarrow$ No se rechaza H_0

$|t_0| > t_{\alpha/2} \Rightarrow$ Se rechaza H_0

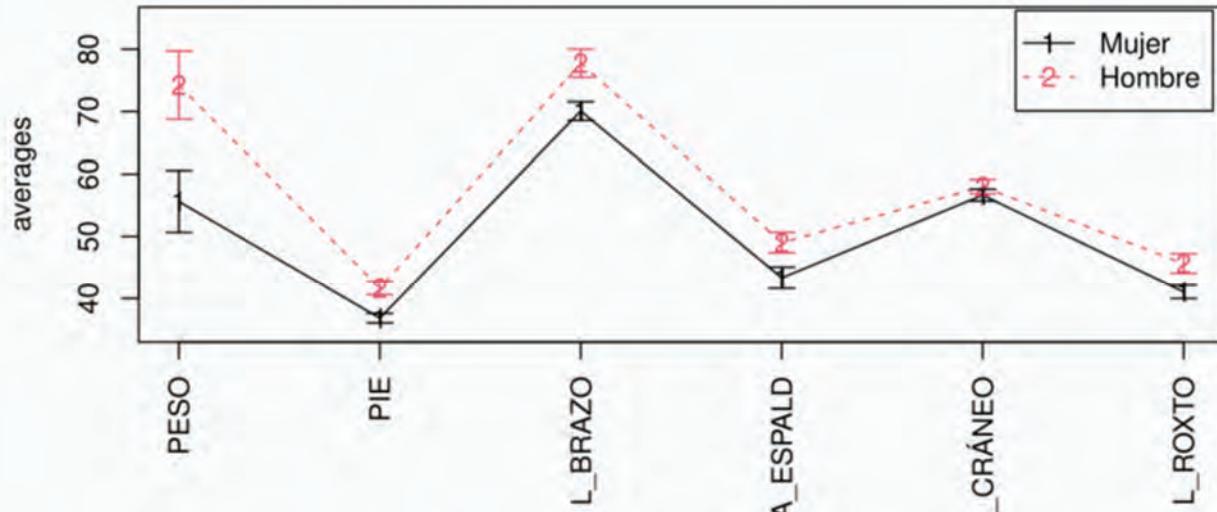
Diferencias entre medias

```
plotMeans(x = dat[,2:8], by = dat$SEXO,xleg="topright")
```



Diferencias entre medias (zoom)

```
plotMeans(x = dat[,3:8], by = dat$SEXO,xleg="topright",
          ylim=c(35,85))
```



Contraste global (lambda WILKS)

$$H_0: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

$$\mu_1 = \begin{pmatrix} \mu_{11} \\ \mu_{12} \\ \vdots \\ \mu_{1k} \end{pmatrix} \quad \mu_2 = \begin{pmatrix} \mu_{21} \\ \mu_{22} \\ \vdots \\ \mu_{2k} \end{pmatrix}$$

```
m2$sigTest
```

```
##           WilksL      F df1 df2      p
## 1 to 1  0.229 9.15     7 19 5.86e-05
```

Claramente
distintos

Tabla de confusión

```
m2$class$orgTab
```

```
##      pred
## orig   Mujer Hombre Sum
## Mujer    15     0  15
## Hombre     0    12  12
## Sum      15    12  27
```

```
m2$class$perTab
```

```
##      pred
## orig   Mujer Hombre Sum
## Mujer    100     0 100
## Hombre     0    100 100
```

```
m2$class$corPer
```

```
## [1] 100
```

Tabla de Confusión (CV)

```
m2$classCV$orgTab
```

```
##          pred
## orig      Mujer Hombre Sum
##   Mujer     13      2  15
##   Hombre     2     10  12
##   Sum       15     12  27
```

```
m2$classCV$perTab
```

```
##          pred
## orig      Mujer Hombre Sum
##   Mujer    86.7   13.3 100.0
##   Hombre   16.7   83.3 100.0
```

```
m2$classCV$corPer
```

```
## [1] 85.2
```



Discriminante

8. EJEMPLO CON R



Creditscoring: Solicitud de un crédito.

Los datos se analizan en el libro de Jeffrey Simonoff, "Analyzing Categorical Data", corresponden a la concesión o no de un crédito en 100 solicitudes con información económica del solicitante. El nombre de la variable aclara su significado, la variable *derogatory.reports* indica el número de problemas asociados a la tarjeta de crédito del solicitante. La variable respuesta es *Application.accepted*. El objetivo de este estudio es construir una función lineal que explique esta última variable en términos del resto.

```
credit = read.csv('creditscore.csv', header=T)
names(credit)

## [1] "Age"           "Income.per.dependent"
## [3] "Monthly.credit.card.exp" "Own.home"
## [5] "Self.employed"      "Derogatory.reports"
## [7] "Accepted"
```

Librería y datos

```
library(MASS)
library(multiUS)
library(klaR)
```

```
dat = read.csv("data/creditscore.csv", header=TRUE)
dat$Accepted =
  factor(dat$Accepted,
        labels=c("No", "Sí"))
```

LDA

```
m1 = lda(Accepted~., data=dat)
m1

## Call:
## lda(Accepted ~ ., data = dat)
##
## Prior probabilities of groups:
##   No    Sí
## 0.27 0.73
##
## Group means:
##   Age Income.per.dependent Monthly.credit.card.exp
## No 34.6           3.26          0
## Sí 31.2           3.41         259
##   Own.home Self.employed Derogatory.reports
## No    0.333      0.1111       1.04
## Sí    0.370      0.0274       0.11
##
## Coefficients of linear discriminants:
##                               LD1
## Age                      -0.05176
## Income.per.dependent     0.02795
## Monthly.credit.card.exp  0.00235
## Own.home                  -0.01418
## Self.employed              -1.94004
## Derogatory.reports        -0.73204
```

Función discriminante

$$y_i = a_0 - 0.051Age_i + 0.0279Income_i + 0.0023Monthly_i - 0.014Own_i - 1.94Self_i - 0.732Derogatory_i$$

```
m2 = ldaPlus(x = dat[,1:6], grouping = dat$Accepted)
m2$centroids
```

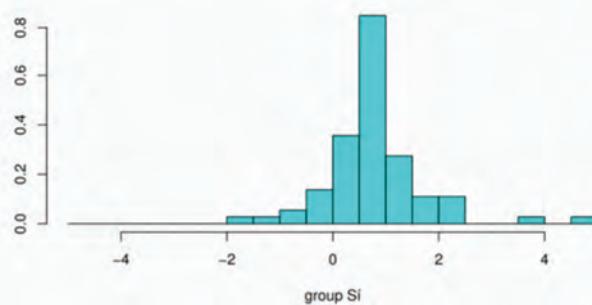
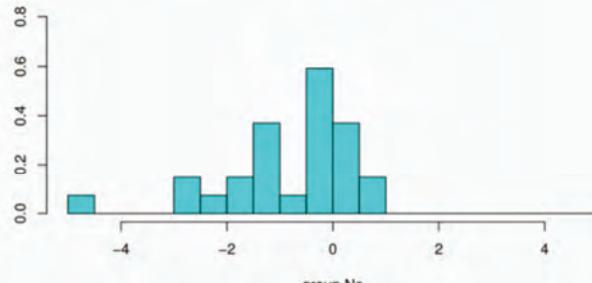
```
## [1] -1.191  0.441
```

```
a_0 = - sum(m1$scaling*sapply(dat[,1:6],mean))
a_0
```

```
## [1] 1.49
```

Gráfico de Scores

```
plot(m1)
```



Función Estandarizada: ldaPLUS()

```
m2 = ldaPlus(x = dat[,1:6], grouping = dat$Accepted)
```

```
m2$standCoefWithin
```

```
##                                     LD1
## Age                      -0.39933
## Income.per.dependent    0.04572
## Monthly.credit.card.exp 0.63898
## Own.home                 -0.00687
## Self.employed            -0.42086
## Derogatory.reports       -0.67808
```

$$y_i = 1.49 - .399Age'_i + .0457Income'_i + .638Monthly'_i - .0068Own'_i - .420Self'_i - .678Derogatory'_i$$

¿Hay diferencias significativas entre los dos grupos?

```
m2$sigTest
```

```
##          WilksL    F df1 df2      p
## 1 to 1  0.651 8.3    6   93 3.32e-07
```

Sí, muy significativas, $p - valor = 0.000000332$

Tablas de Confusión

```
m2$class$orgTab
```

```
##      pred
## orig  No  Sí Sum
##  No   12  15 27
##  Sí    2  71 73
##  Sum   14  86 100
```

```
m2$class$perTab
```

```
##      pred
## orig      No      Sí      Sum
##  No  44.44 55.56 100.00
##  Sí  2.74 97.26 100.00
```

```
m2$class$corPer
```

```
## [1] 83
```

El porcentaje de acierto es el 83% (acierta más cuando es "Sí" (97.2%) que cuando es "No" (44.4%))

Tablas de Confusión (CV)

```
m2$classCV$perTab
```

```
##      pred
## orig    No     Sí   Sum
##   No  37.04 62.96 100.00
##   Sí  5.48 94.52 100.00
```

```
m2$classCV$corPer
```

```
## [1] 79
```

Variables importantes (significativas)

```
m3 = greedy.wilks(Accepted ~ ., data=dat[,1:7])
m3
```

```
## Formula containing included variables:
##
## Accepted ~ Derogatory.reports + Monthly.credit.card.exp + Self.employed +
##             Age
## <environment: 0x000000002648d340>
##
##
## Values calculated in each step of the selection procedure:
##
##                               vars Wilks.lambda F.statistics.overall
## 1           Derogatory.reports      0.832          19.8
## 2   Monthly.credit.card.exp      0.725          18.4
## 3           Self.employed      0.687          14.6
## 4                 Age      0.652          12.7
##   p.value.overall F.statistics.diff p.value.diff
## 1      2.32e-05        19.76  2.32e-05
## 2      1.73e-07        14.28  2.70e-04
## 3      6.80e-08        5.34   2.30e-02
## 4      2.56e-08        5.18   2.50e-02
```

Modelo con las variables significativas

```
m4 = lda(m3$formula, data = dat[,1:7])
m4

## Call:
## lda(m3$formula, data = dat[, 1:7])
##
## Prior probabilities of groups:
##   No    Sí 
## 0.27 0.73 
##
## Group means:
##   Derogatory.reports Monthly.credit.card.exp Self.employed
##   No           1.04                  0       0.1111
##   Sí           0.11                 259     0.0274
##
##   Age
##   No 34.6
##   Sí 31.2
##
## Coefficients of linear discriminants:
##                               LD1
## Derogatory.reports      -0.72779
## Monthly.credit.card.exp  0.00241
## Self.employed            -1.89155
## Age                      -0.05059
```

$$y_i = 1.52 - .7277Derogatory_i + .0024Monthly_i - 1.89Self_i - 0.0505Age_i$$

Interpretación

```
m5 = ldaPlus(x = dat[,c(6,3,5,1)], grouping = dat$Accepted)
m5$centroids
```

```
## [1] -1.19  0.44
```

```
m5$standCoefWithin
```

```
##                               LD1
## Derogatory.reports      -0.674
## Monthly.credit.card.exp  0.654
## Self.employed            -0.410
## Age                      -0.390
```

- Valores altos de Derogatory.reports , Self.employed y Age empeoran la puntuación (perjudica la concesión del crédito)
- Valores altos de Monthly.credit mejoran la puntuación (favorece el crédito)

Correlación entre Scores y Variables

```
m5$corr
```

```
## LD1
## Derogatory.reports      -0.614
## Monthly.credit.card.exp 0.584
## Self.employed            -0.237
## Age                      -0.274
```

Ayuda a interpretar la función discriminante.

Nuevo cliente

Viene un nuevo cliente con 30 años (Age = 30), que es funcionario (Self.employed=0), No ha tenido ninguna reclamación en su tarjeta de crédito (Derogatory.reports=0) y tiene unos gastos medios mensuales en su tarjeta de crédito de 500 \$.

Su puntuación discriminante es

$$y_i = 1.52 - .7277Derogatory_i + .0024Monthly_i - 1.89Self_i - 0.0505Age_i$$

$$y_i = 1.52 - .7277 \times 0 + .0024 \times 500 - 1.89 \times 0 - 0.0505 \times 30 = 1.21$$

Nuevo cliente predict()

```
cliente=data.frame(Age=30,  
                   Monthly.credit.card.exp=500,  
                   Self.employed=0, Derogatory.reports=0)  
  
predict(m4,newdata = cliente)
```

```
## $class  
## [1] Sí  
## Levels: No Sí  
##  
## $posterior  
##          No      Sí  
## 1 0.0271 0.973  
##  
## $x  
##      LD1  
## 1 1.21
```

La clasificación final depende de la probabilidad (se asigna al grupo con mayor probabilidad). En este cálculo se tiene en cuenta la proporción inicial de cada clase.

```
tapply(m5$pred$x,dat$Accepted,mean)
```

```
##      No      Sí  
## -1.19  0.44
```

centroides

Nuevos clientes predict()

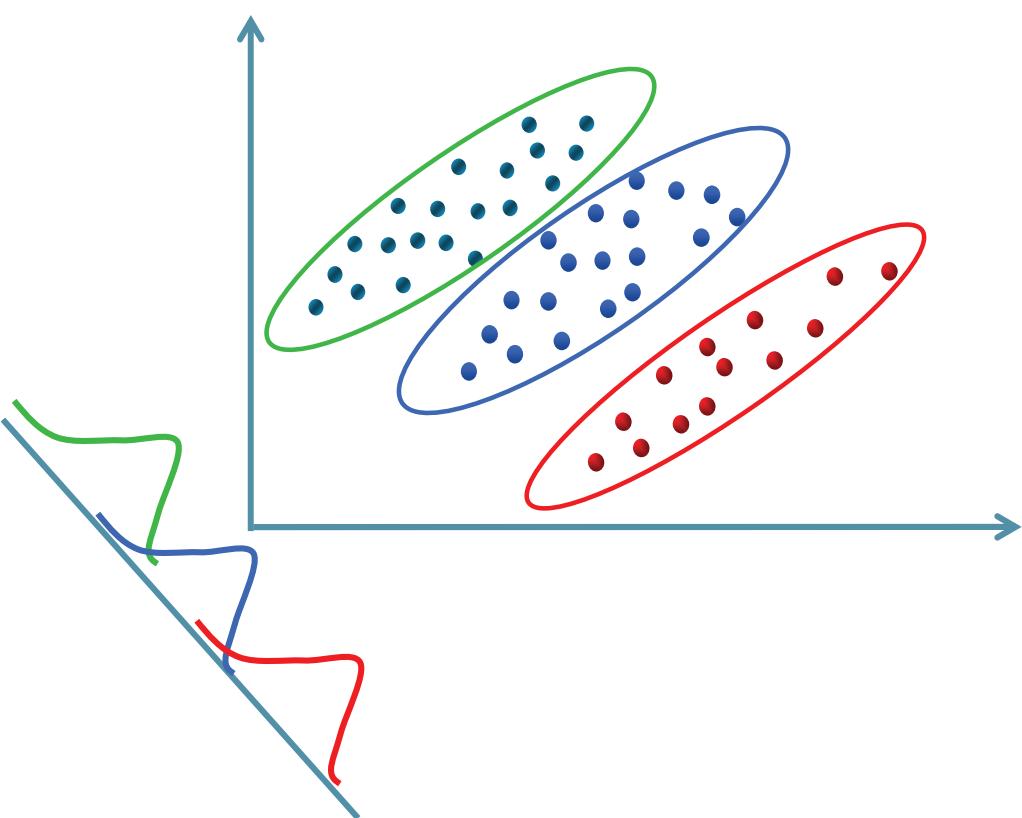
```
clientes=data.frame(Age=c(50,30,45),  
                   Monthly.credit.card.exp=c(100,500,1000),  
                   Self.employed=c(0,1,1),  
                   Derogatory.reports=c(0,1,0))  
  
predict(m4,newdata = clientes)
```

```
## $class  
## [1] Sí No Sí  
## Levels: No Sí  
##  
## $posterior  
##          No      Sí  
## 1 0.411 0.589  
## 2 0.666 0.334  
## 3 0.228 0.772  
##  
## $x  
##      LD1  
## 1 -0.764  
## 2 -1.409  
## 3 -0.237
```

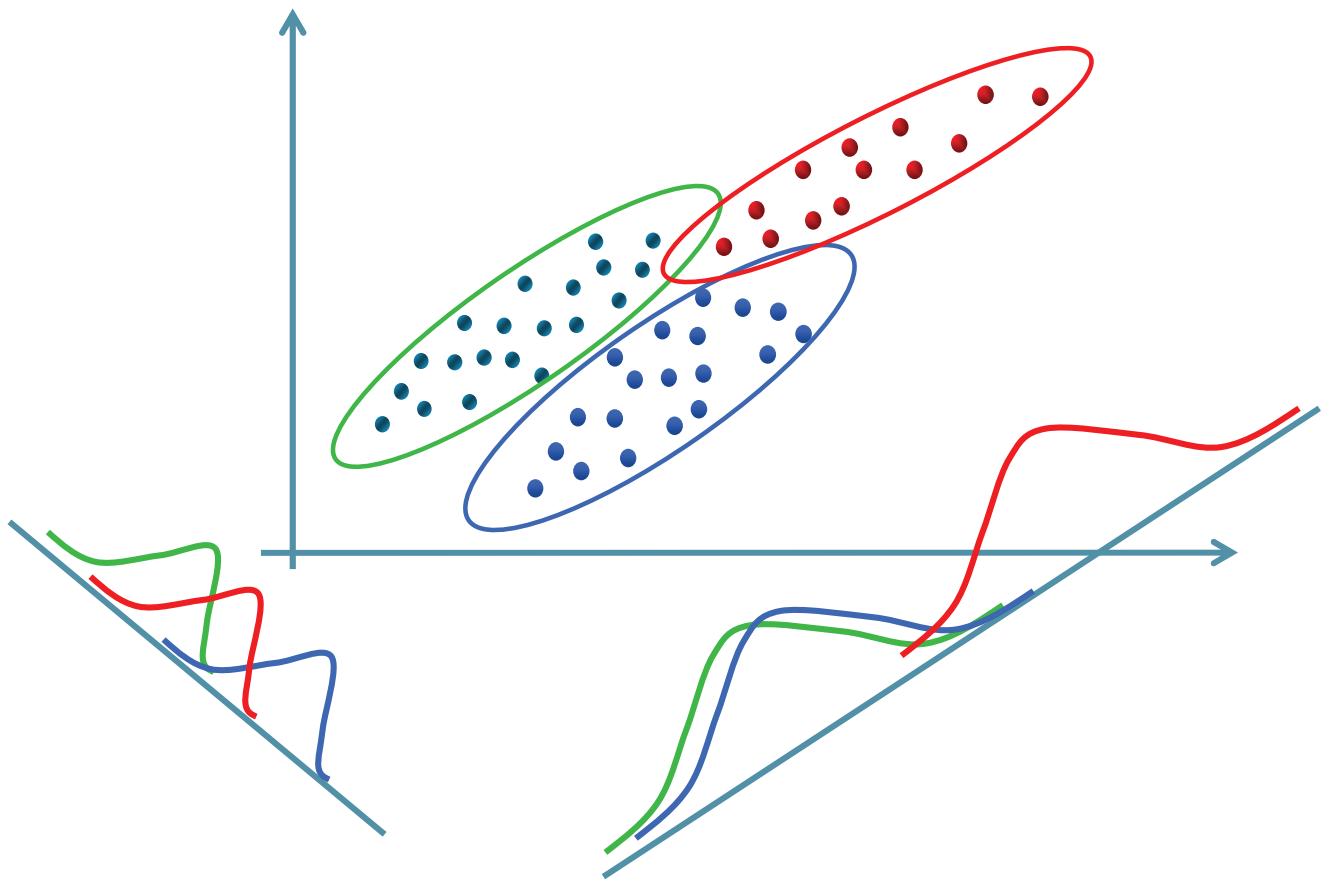
Análisis Discriminante

Más de dos grupos

Una función discriminante



Dos funciones discriminantes



Método de cálculo

Grupo 1					
x_{111}	x_{121}	x_{131}	...	x_{1k1}	
x_{112}	x_{122}	x_{132}	...	x_{1k2}	
x_{113}	x_{123}	x_{133}	...	x_{1k3}	
:	:	:	..	:	
x_{11n}	x_{12n}	x_{13n}	...	x_{1kn}	
\bar{x}_{11}	\bar{x}_{12}	\bar{x}_{13}	...	\bar{x}_{1k}	

Grupo 2					
x_{211}	x_{221}	x_{231}	...	x_{2k1}	
x_{212}	x_{222}	x_{232}	...	x_{2k2}	
x_{213}	x_{223}	x_{233}	...	x_{2k3}	
:	:	:	..	:	
x_{21n}	x_{22n}	x_{23n}	...	x_{2kn}	
\bar{x}_{21}	\bar{x}_{22}	\bar{x}_{23}	...	\bar{x}_{2k}	

Grupo p					
x_{p11}	x_{p21}	x_{p31}	...	x_{pk1}	
x_{p12}	x_{p22}	x_{p32}	...	x_{pk2}	
x_{p13}	x_{p23}	x_{p33}	...	x_{pk3}	
:	:	:	..	:	
x_{p1n}	x_{p2n}	x_{p3n}	...	x_{pkn}	
\bar{x}_{p1}	\bar{x}_{p2}	\bar{x}_{p3}	...	\bar{x}_{pk}	

$$\bar{X}_1$$

$$\bar{X}_2$$

...

$$\bar{X}_p$$

$$\mathbf{S}_1 = \begin{pmatrix} s_1^2 & s_{12} & \cdots & s_{1k} \\ s_{12} & s_2^2 & \cdots & s_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1k} & s_{2k} & \cdots & s_k^2 \end{pmatrix}$$

$$\mathbf{S}_2 = \begin{pmatrix} s'^2_1 & s'_{12} & \cdots & s'_{1k} \\ s'_{12} & s'^2_2 & \cdots & s'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s'_{1k} & s'_{2k} & \cdots & s'^2_k \end{pmatrix}$$

$$\mathbf{S}_p = \begin{pmatrix} s''^2_1 & s''_{12} & \cdots & s''_{1k} \\ s''_{12} & s''^2_2 & \cdots & s''_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s''_{1k} & s''_{2k} & \cdots & s''^2_k \end{pmatrix}$$

Método de cálculo (cont.)

$$\mathbf{S}_T = \frac{n_1}{n} \mathbf{S}_1 + \frac{n_2}{n} \mathbf{S}_2 + \cdots + \frac{n_p}{n} \mathbf{S}_p$$

$$\bar{\mathbf{X}}_{\bullet} = \frac{\bar{\mathbf{X}}_1 + \bar{\mathbf{X}}_2 + \cdots + \bar{\mathbf{X}}_p}{p} = \begin{pmatrix} \bar{x}_{\bullet 1} \\ \bar{x}_{\bullet 2} \\ \vdots \\ \bar{x}_{\bullet k} \end{pmatrix}$$

$$B = \frac{1}{p} \sum_{i=1}^p \begin{pmatrix} \bar{x}_{i1} - \bar{x}_{\bullet 1} \\ \bar{x}_{i2} - \bar{x}_{\bullet 2} \\ \vdots \\ \bar{x}_{ik} - \bar{x}_{\bullet k} \end{pmatrix} (\bar{x}_{i1} - \bar{x}_{\bullet 1} \quad \bar{x}_{i2} - \bar{x}_{\bullet 2} \quad \cdots \quad \bar{x}_{ik} - \bar{x}_{\bullet k})$$

Método de cálculo (cont.)

$$\mathbf{Q} = \mathbf{S}_T^{-1} \mathbf{B}$$

$$\mathbf{Q} = \mathbf{P} \mathbf{D} \mathbf{P}^{-1}$$

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1k} \\ q_{21} & q_{22} & q_{23} & \cdots & q_{2k} \\ q_{31} & q_{32} & q_{33} & \cdots & q_{3k} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ q_{k1} & q_{k2} & q_{k3} & \cdots & q_{kk} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kk} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_k \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kk} \end{pmatrix}^{-1}$$

Funciones discriminantes

		Funciones discriminantes			
		Y_1	Y_2	\dots	Y_r
Variables Originales	X_1	a_{11}	a_{12}	\dots	a_{1r}
	X_2	a_{21}	a_{22}	\dots	a_{2r}
	\vdots	\vdots	\vdots	\ddots	\vdots
	X_k	a_{k1}	a_{k2}	\dots	a_{kr}
Valores Propios		λ_1	λ_2	\dots	λ_r

Se pueden obtener $r = \min(k, p - 1)$ funciones discriminantes.

Poder de discriminación

Valores propios: $\mathbf{S}_T^{-1}\mathbf{B} = \{\lambda_1, \lambda_2, \dots, \lambda_r\}$

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$$

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \dots + \lambda_r}$$

Importancia del factor Y_i

Contraste de igualdad de medias

$$H_0: \mu_1 = \mu_2 = \cdots = \mu_p$$

$H_1:$ Alguna es distinta

$$\boldsymbol{\mu}_1 = \begin{pmatrix} \mu_{11} \\ \mu_{12} \\ \vdots \\ \mu_{1k} \end{pmatrix} \quad \boldsymbol{\mu}_2 = \begin{pmatrix} \mu_{21} \\ \mu_{22} \\ \vdots \\ \mu_{2k} \end{pmatrix} \quad \cdots \quad \boldsymbol{\mu}_p = \begin{pmatrix} \mu_{p1} \\ \mu_{p2} \\ \vdots \\ \mu_{pk} \end{pmatrix}$$

Contraste (Bartlett)

Si H_0 es cierto:

$$V = \left\{ (n - 1) - \frac{k + p}{2} \right\} \sum_{i=1}^r \log(1 + \lambda_i) \rightarrow \chi^2_{k(p-1)}$$

$$V_j = \left\{ (n - 1) - \frac{k + p}{2} \right\} \log(1 + \lambda_i) \rightarrow \chi^2_{k+p-2j}$$

¿Cuántas funciones discriminantes?

$$V_j = \left\{ (n - 1) - \frac{k + p}{2} \right\} \log(1 + \lambda_i) \rightarrow \chi^2_{k+p-2j}$$

H_0 : Los grupos son iguales

H_1 : Existen 1 o más,	V	$\rightarrow \chi^2_{k(p-1)}$
H_1 : Existen 2 o más,	$V - V_1$	$\rightarrow \chi^2_{(k-1)(p-1)}$
⋮	⋮	⋮
H_1 : Existen r ,	$V - \sum_{i=1}^{r-1} V_i$	$\rightarrow \chi^2_{(k-r)(p-1)}$

Lirios



Virginica



Versicolor



Setosa

Ejemplo: Lirios

1.- Virginica

2.- Versicolor

3.- Setosa

	Grupo	X1	X2	X3	X4
1	1	5,1	3,5	1,4	0,2
2	1	4,9	3,0	1,4	0,2
3	1	4,7	3,2	1,3	0,2
4	1	4,6	3,1	1,5	0,2
5	1	5,0	3,6	1,4	0,2
:	:	:	:	:	:
50	1	5,0	3,3	1,4	0,2
1	2	7,0	3,2	4,7	1,4
2	2	6,4	3,2	4,5	1,5
3	2	6,9	3,1	4,9	1,5
4	2	5,5	2,3	4,0	1,3
5	2	6,5	2,8	4,6	1,5
:	:	:	:	:	:
50	2	5,7	2,8	4,1	1,3
1	3	6,3	3,3	6,0	2,5
2	3	5,8	2,7	5,1	1,9
3	3	7,1	3,0	5,9	2,1
4	3	6,3	2,9	5,6	1,8
5	3	6,5	3,0	5,8	2,2
:	:	:	:	:	:
50	3	5,9	3,0	5,1	1,8

X_1 : Longitud sépalo

X_2 : Anchura sépalo

X_3 : Longitud pétalo

X_4 : Anchura pétalo

Datos y paquetes

```
library(MASS)
library(multiv)
```

```
iris = read.table('data/lirios.txt', header=T)
iris$Species=factor(iris$Species)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Análisis discriminante lda()

```
m1 = lda(Species~., data=iris)

## Call:
## lda(Species ~ ., data = iris)
##
## Prior probabilities of groups:
##   setosa versicolor virginica
##   0.333     0.333     0.333
##
## Group means:
##             Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa          5.01        3.43       1.46      0.246
## versicolor      5.94        2.77       4.26      1.326
## virginica       6.59        2.97       5.55      2.026
##
## Coefficients of linear discriminants:
##                 LD1     LD2
## Sepal.Length  0.829  0.0241
## Sepal.Width   1.534  2.1645
## Petal.Length -2.201 -0.9319
## Petal.Width  -2.810  2.8392
##
## Proportion of trace:
##    LD1     LD2
## 0.9912  0.0088
```

Coeficientes de las funciones discriminantes

```
m1$scaling
```

```
##                 LD1     LD2
## Sepal.Length  0.829  0.0241
## Sepal.Width   1.534  2.1645
## Petal.Length -2.201 -0.9319
## Petal.Width  -2.810  2.8392
```

Número de funciones discriminantes es $\min(\text{grupos} - 1, \text{variables}) = 2$

La primera función discriminante es:

$$Y_{1i} = 0.82X_{1i} + 1.53X_{2i} - 2.20X_{3i} - 2.81X_{4i}$$

La segunda función discriminante es:

$$Y_{2i} = 0.024X_{1i} + 2.16X_{2i} - 0.93X_{3i} + 2.83X_{4i}$$

Falta la constante, no es importante

Funciones discriminantes

Funciones canónicas discriminantes

	Función	
	1	2
L_SEPALO	-,829	,024
A_SÉPALO	-1,534	2,165
L_PÉTALO	2,201	-,932
A_PÉTALO	2,810	2,839
(Constante)	-2,105	-6,661

$$Y_{1i} = -2.105 - 0.829X_{1i} - 1.534X_{2i} + 2.201X_{3i} + 2.810X_{4i}$$

$$Y_{2i} = -6.661 + 0.024X_{1i} + 2.621X_{2i} - 0.932X_{3i} + 2.839X_{4i}$$

Solución SPSS: signos pueden cambiar !

Dos funciones discriminantes

Setosa

Versicolor

Virginica

LD1 LD2

1	1	5,1	3,5	1,4	0,2	8,06	0,30		
2	1	4,9	3,0	1,4	0,2	7,13	-0,79		
3	1	4,7	3,2	1,3	0,2	7,49	-0,27		
4	1	4,6	3,1	1,5	0,2	6,81	-0,67		
5	1	5,0	3,6	1,4	0,2	8,13	0,51		
:	:	:	:	:	:	:	:		
50	1	5,0	3,3	1,4	0,2	7,67	-0,13	7,61	0,22
1	2	7,0	3,2	4,7	1,4	-1,46	0,03		
2	2	6,4	3,2	4,5	1,5	-1,80	0,48		
3	2	6,9	3,1	4,9	1,5	-2,42	-0,09		
4	2	5,5	2,3	4,0	1,3	-2,26	-1,59		
5	2	6,5	2,8	4,6	1,5	-2,55	-0,47		
:	:	:	:	:	:	:	:		
50	2	5,7	2,8	4,1	1,3	-1,55	-0,59	-1,83	-0,73
1	3	6,3	3,3	6,0	2,5	-7,84	2,14		
2	3	5,8	2,7	5,1	1,9	-5,51	-0,04		
3	3	7,1	3,0	5,9	2,1	-6,29	0,47		
4	3	6,3	2,9	5,6	1,8	-5,61	-0,34		
5	3	6,5	3,0	5,8	2,2	-6,85	0,83		
:	:	:	:	:	:	:	:		
50	3	5,9	3,0	5,1	1,8	-7,68	0,33	-5,78	0,51

Centroides

Obtención de las puntuaciones Y_{1i} y Y_{2i} y más.

la función `predict(m1)` proporciona información importante para cada una de las observaciones (flores):

```
pred = predict(m1)
names(pred)

## [1] "class"      "posterior"   "x"
```

- `class`: la clase que según el modelo asigna (re-asigna) a cada flor
- `Posterior`: las probabilidades de pertenecer a cada clase según el modelo
- `x` : las puntuaciones discriminantes

Funciones discriminantes `pred$x`

Tiene tantas filas como observaciones. Aquí se muestran las 12 primeras

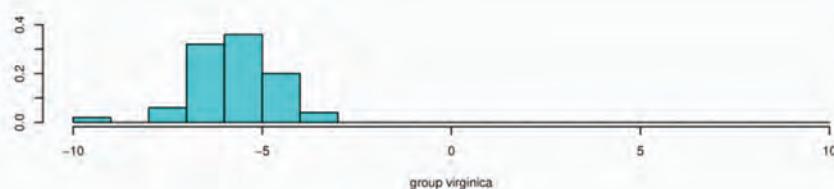
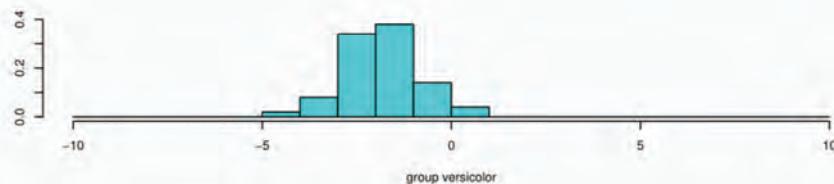
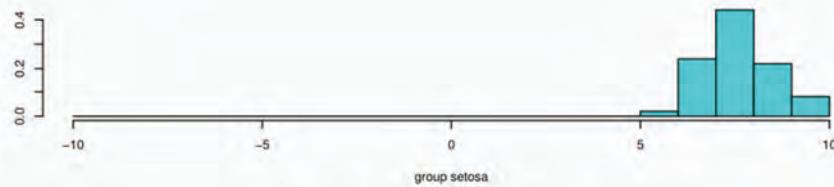
```
head(pred$x, 12)
```

```
##      LD1      LD2
## 1  8.06  0.3004
## 2  7.13 -0.7867
## 3  7.49 -0.2654
## 4  6.81 -0.6706
## 5  8.13  0.5145
## 6  7.70  1.4617
## 7  7.21  0.3558
## 8  7.61 -0.0116
## 9  6.56 -1.0152
## 10 7.34 -0.9473
## 11 8.40  0.6474
## 12 7.22 -0.1096
```

Gráfico de las puntuaciones

En el primer gráfico se representan los valores Y_{1i} . Se observan diferencias entre las tres variedades. Virginica y versicolor se parecen más, la setosa está más alejada (es más diferente).

```
plot(m1,dimen=1)
```



Las dos funciones discriminantes

```
plot(m1,dimen=2,col = as.integer(iris$Species),  
      abbrev=1,cex=1.5)
```

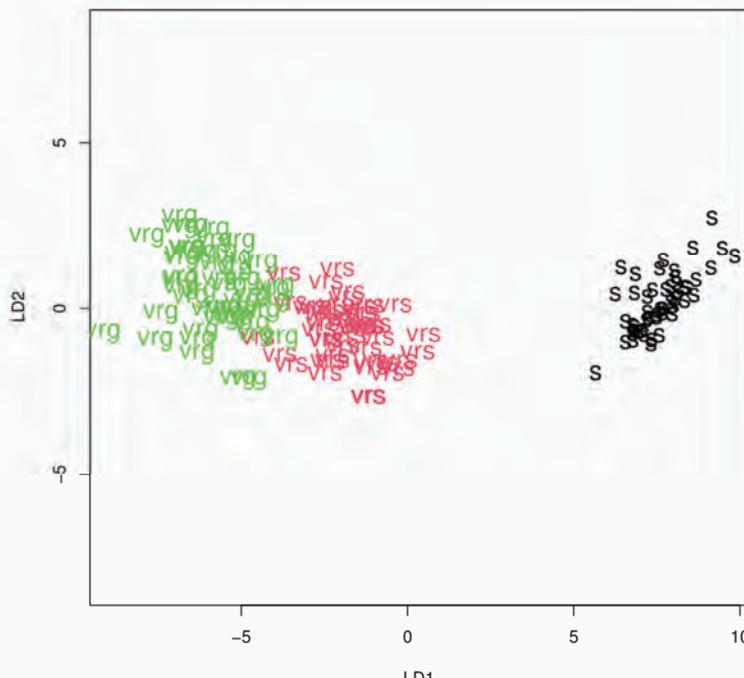


Gráfico personalizado (1)

```
plot(pred$x[,1],pred$x[,2],  
      col=as.integer(iris$Species), pch=19, cex=1.3)  
  
legend(1,2,legend=c("setosa","versicolor","virginia"),  
      col=c("black","red","green"),pch=19)  
  
grid()
```

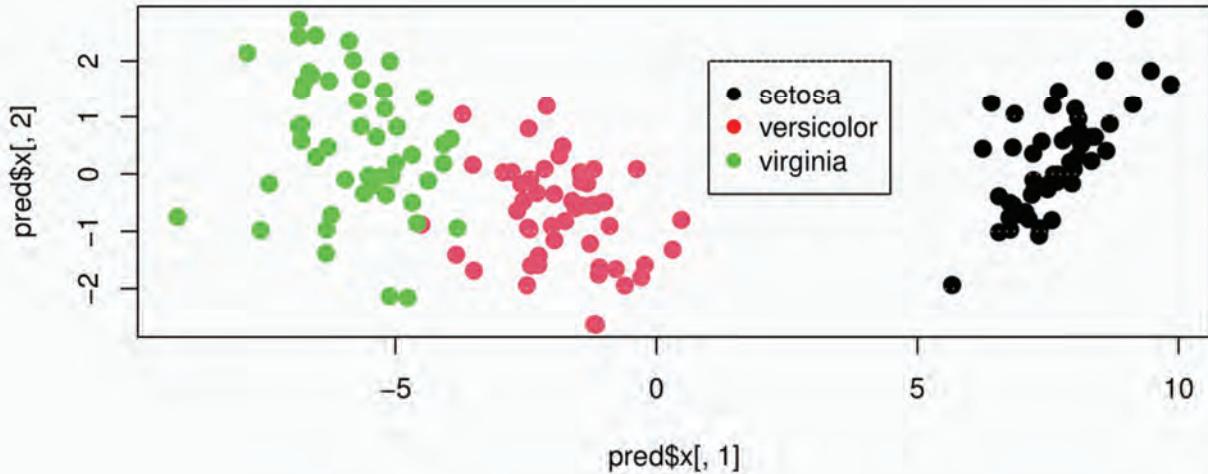
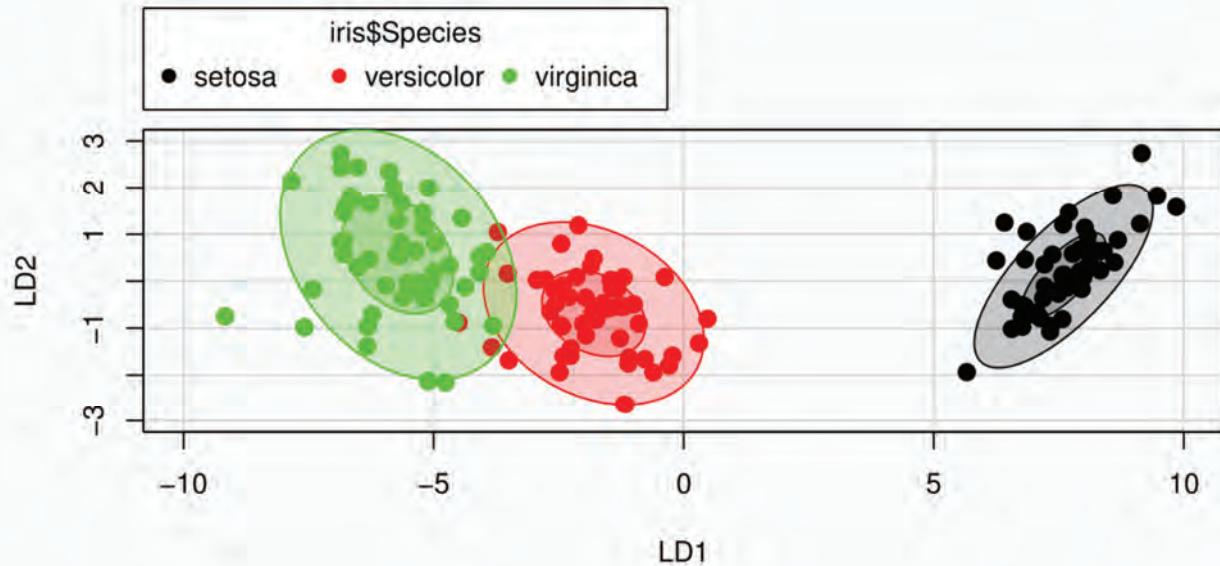


Gráfico personalizad (2)

```
library(car)  
scatterplot(pred$x[,1],pred$x[,2],groups= iris$Species,  
           pch=c(19,19,19),cex=1.3,  
           xlim=c(-10,10), ylim=c(-3,3),regLine=FALSE,  
           ellipse = TRUE,smooth = FALSE,  
           col=c("black","red","green"),  
           xlab="LD1", ylab="LD2")
```



Otros resultados de análisis discriminante: ldaPLUS()

```
library(multiUS)
m2 = ldaPlus(x = iris[,1:4], grouping = iris$Species)
names(m2)

## [1] "prior"           "counts"
## [3] "means"           "scaling"
## [5] "standCoefWithin" "standCoefTotal"
## [7] "lev"              "svd"
## [9] "N"                "betweenGroupsWeights"
## [11] "call"             "sigTest"
## [13] "eigModel"         "pred"
## [15] "centroids"        "corr"
## [17] "class"            "classCV"
```

En m2 está m1 y más (standCoefWithin, pred, sigTest, ...)

Pesos de las funciones discriminantes estandarizadas

Sirven para interpretar los coeficientes !

```
m2$standCoefWithin
```

```
##          LD1      LD2
## Sepal.Length 0.427  0.0124
## Sepal.Width   0.521  0.7353
## Petal.Length -0.947 -0.4010
## Petal.Width   -0.575  0.5810
```

La variable importante en LD1 es la longitud del pétalo (Petal.Length) y en la segunda LD la anchura del sépalo (Sepal.Width)

Interpretación de las funciones discriminantes

La primera función discriminante es:

$$Y_{1i} = 0.82X_{1i} + 1.53X_{2i} - 2.20X_{3i} - 2.81X_{4i} + k_1$$

con los coeficientes estandarizados

$$Y_{1i} = 0.42 \frac{X_{1i} - \bar{X}_1}{S_1} + 0.52 \frac{X_{2i} - \bar{X}_2}{S_2} - 0.94 \frac{X_{3i} - \bar{X}_3}{S_3} - 0.57 \frac{X_{4i} - \bar{X}_4}{S_4}$$

La media de Y_{1i} es 0. Para las flores de la variedad *setosa*, la media de Y_{1i} es 7.6, para las *versicolor* es -1.83 y para la *virginica* es -5.78. Una puntuación alta y positiva de Y_1 indicará que es *setosa* y una puntuación alta y negativa indicará que es *virginica*. Una puntuación intermedia que es *versicolor*. Una longitud grande del pétalo (peso -0.94) indicará que la flor es *virginica* y una longitud pequeña (por debajo de la media) que es *setosa*. La variable que primero tenemos que mirar para clasificar una flor es la longitud del pétalo. Las dos variables siguientes tienen un peso parecido, tienen una importancia similar. Una flor con el pétalo ancho indicará que es *virginica* y si es estrecho que es *setosa*. La interpretación una a una de las variables es aproximada, porque lo que cuenta es la combinación lineal de las cuatro variables, es decir tener en cuenta todas las variables a la vez.

La interpretación anterior se puede hacer observando las medias de las cuatro variables en los tres grupos. La longitud del pétalo es la que muestra mayores diferencias entre los tres grupos. Después la anchura del pétalo. En la anchura del sépalo, las variedades *versicolor* y *virgínica* son similares.

m2\$centroids

```
##           LD1     LD2
## setosa      7.61  0.215
## versicolor -1.83 -0.728
## virginica  -5.78  0.513
```

Contraste de igualdad de medias

Observando el gráfico de las puntuaciones discriminantes, es obvio que existen diferencias claras entre las cuatro especies. En alguna ocasión, los gráficos no muestran diferencias tan evidentes y es útil realizar el contraste de igualdad de los tres grupos.

m2\$sigTest

```
##        WilksL      F df1 df2      p
## 1 to 2 0.0234 199.1    8 288 1.37e-112
## 2 to 3 0.7780 13.8    3 145 5.79e-08
```

- La primera línea nos indica que existen diferencias muy significativas (`pvalue = 1.36e-112`) entre los grupos. Algun grupo (al menos) tiene media distinta. Por tanto necesitamos al menos una función discriminante.
- La segunda línea nos indica que "eliminado el efecto de la primera función discriminante", todavía existen diferencias significativas entre los tres grupos. Por tanto, también son "significativas" las diferencias detectadas por la segunda función discriminante.

Importancia de cada función

```
m2$eigModel
```

```
##          Eigenvalues      % Cum %   Cor Sq. Cor
## [1,]      32.192 99.121 99.1 0.985  0.970
## [2,]       0.285  0.879 100.0 0.471  0.222
```

El número de funciones posibles es dos. La primera tiene una importancia de 32.19 (valor propio) y la segunda de (0.285) (segundo valor propio). La importancia relativa de la primera es 99.12% y de la segunda 0.88%.

Clasificación

```
m2$class$orgTab
```

```
##          pred
## orig      setosa versicolor virginica Sum
## setosa     50        0        0    50
## versicolor 0        48        2    50
## virginica  0        1        49    50
## Sum        50        49        51 150
```

```
m2$class$perTab
```

```
##          pred
## orig      setosa versicolor virginica Sum
## setosa    100        0        0 100
## versicolor 0        96        4 100
## virginica  0        2        98 100
```

```
m2$class$corPer
```

```
## [1] 98
```

Una nueva flor ¿a qué especie pertenece?



Clasificación de una nueva flor

Nueva Flor:

$$\begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \\ 5.70 & 2.90 & 4.40 & 1.70 \end{array}$$

Valores de la función discriminante

$$Y_1 = +2.105 + 0.829X_1 + 1.534X_2 - 2.201X_3 - 2.810X_4 = -3.18$$

$$Y_2 = -6.661 + 0.024X_1 + 2.621X_2 - 0.932X_3 + 2.839X_4 = +0.479$$

Distancia a los grupos

$$D_1 = 10.79$$

$$D_2 = 1.81$$

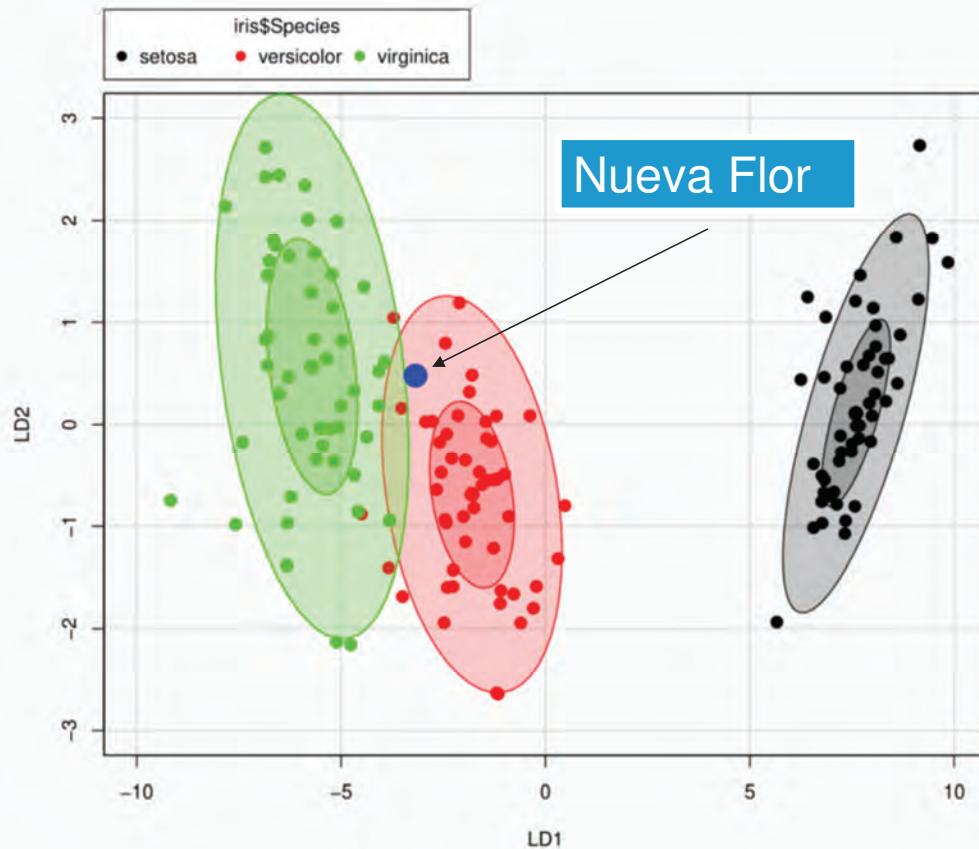
$$D_3 = 2.60$$

→ VERSICOLOR

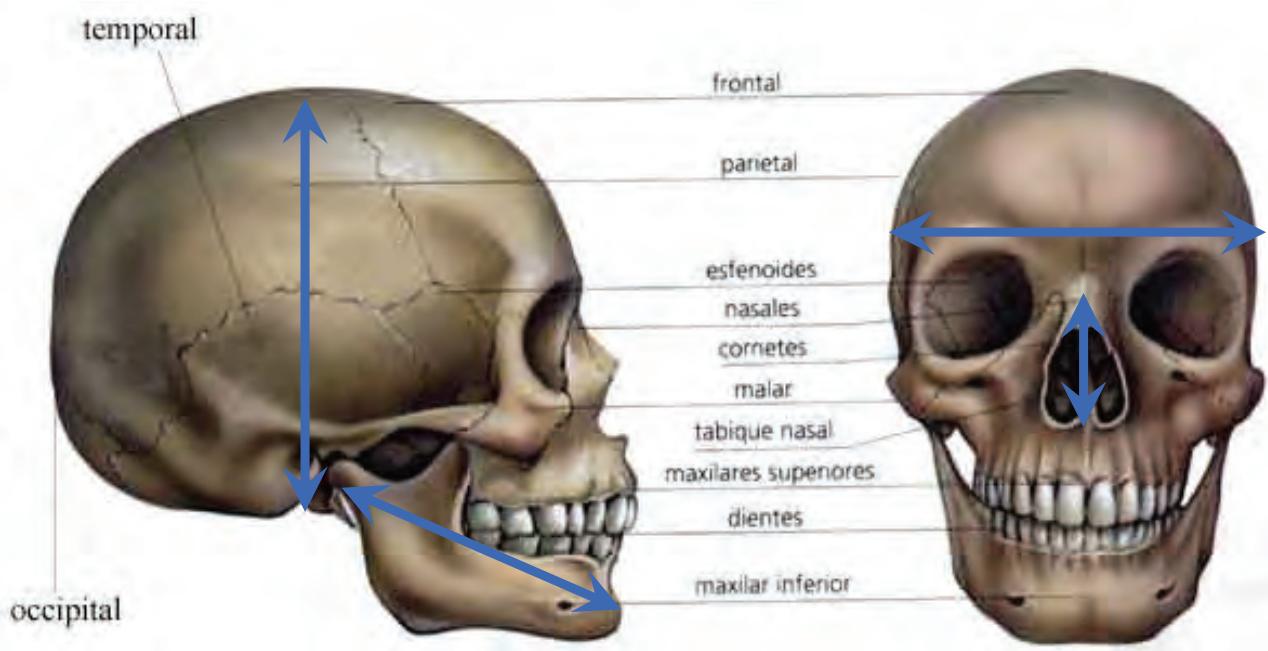
Clasificación de una nueva flor

```
flor = data.frame(Sepal.Length=5.7,  
                  Sepal.Width=2.9,  
                  Petal.Length=4.4,  
                  Petal.Width=1.7)  
predict(m1,newdata = flor)  
  
## $class  
## [1] versicolor  
## Levels: setosa versicolor virginica  
##  
## $posterior  
##      setosa versicolor virginica  
## 1 2.28e-25    0.85     0.15  
##  
## $x  
##      LD1     LD2  
## 1 -3.18  0.479
```

Posición Nueva flor



Ejemplo: Cráneos egipcios



Desarrollo del cráneo egipcio

- **Nombre del archivo de datos:** “craneos.txt”
- **Referencia:** Thomson, A. y Randall-Maciver, R. (1905) *Ancient Races of the Thebaid*, Oxford: Oxford University Press.
También se encuentra en: Hand, D.J., et al. (1994) *A Handbook of Small Data Sets*, New York: Chapman & Hall, pp. 299-301.
Manly, B.F.J. (1986) *Multivariate Statistical Methods*, Nueva York: Chapman & Hall.
- **Descripción:** Cuatro mediciones de cráneos egipcios masculinos de 5 períodos de tiempo diferentes. Se miden 30 cráneos de cada período.

Ejercicio Propuesto

En las siguientes transparencias se presenta el análisis discriminante de los datos correspondientes a las cuatro medidas de cráneos egipcios (variables explicativas) que corresponden a cinco periodos (variable respuesta) distintos que van del año 4000 antes de Cristo al 150 después de Cristo.

El ejercicio que se propone es repetir el análisis realizado utilizando R y contestar a las siguientes preguntas:

- (a) ¿Existen diferencias significativas en las medidas del cráneo en los diferentes períodos.?
- (b) ¿Qué dimensiones son las que más han variado en este tiempo.?
- (c) ¿Este modelo permite hacer una buena clasificación de los cráneos por periodo?

91

Descriptiva

Cinco períodos:
(30 cráneos en cada periodo)

- 1. 4000 BC
- 2. 3300 BC
- 3. 1850 BC
- 4. 200 BC
- 5. 150 AD

	Periodo									
	Predinástico Inicial (4000 BC)		Predinástico Tardío (3300 BC)		Dinastías 12 y 13 (1850 BC)		P. Tolemáico (200 BC)		P. Romano (150 AD)	
	Media	Desv. típ.	Media	Desv. típ.	Media	Desv. típ.	Media	Desv. típ.	Media	Desv. típ.
Anchura C.	131,4	5,13	132,40	4,92	134,47	3,48	135,50	3,92	136,17	5,35
Altura C.	133,6	4,47	132,70	4,65	133,80	4,98	132,30	5,13	130,33	4,97
Mandíbula	99,17	5,88	99,07	4,35	96,03	4,55	94,53	4,59	93,50	5,06
Nasal	50,53	2,76	50,23	2,96	52,23	9,51	51,97	2,82	51,30	3,63

AD : Valores propios (λ_i)

Análisis discriminante

Autovalores

Función	Autovalor	% de varianza	% acumulado	Correlación canónica
1	,409 ^a	89,2	89,2	,539
2	,047 ^a	10,2	99,4	,212
3	,003 ^a	,6	100,0	,053
4	,000 ^a	,0	100,0	,003

a. Se han empleado las 4 primeras funciones discriminantes canónicas en el análisis.

Contrastes

Lambda de Wilks

Contraste de las funciones	Lambda de Wilks	Chi-cuadrado	gl	Sig.
1 a la 4	,676	56,589	16	,000
2 a la 4	,953	7,029	9	,634
3 a la 4	,997	,411	4	,982
4	1,000	,001	1	,972

Sólo hay una función discriminante significativa

Primera función discriminante

Coeficientes de las funciones canónicas discriminantes

	Función			
	1	2	3	4
Anchura C.	-,134	-,002	,168	-,034
Altura C.	,028	,183	,005	-,103
Mandíbula	,148	-,048	,128	,054
Nasal	-,036	,087	-,009	,175
(Constante)	1,771	-23,824	-35,014	3,975

$$Y_1 = 1.771 - 0.134 \text{ AnchC} + 0.028 \text{ AltC} + 0.148 \text{ Man} - 0.036 \text{ Nar}$$

Estandarizados

	Función
	1
Anchura C.	-,134
Altura C.	,028
Mandíbula	,148
Nasal	-,036
(Constante)	1,771

	Función
	1
Anchura C.	-,617
Altura C.	,135
Mandíbula	,729
Nasal	-,184

Coeficientes no tipificados

Los más importantes son la Anchura del cráneo y la Mandíbula

Centroides

Centroides de los grupos

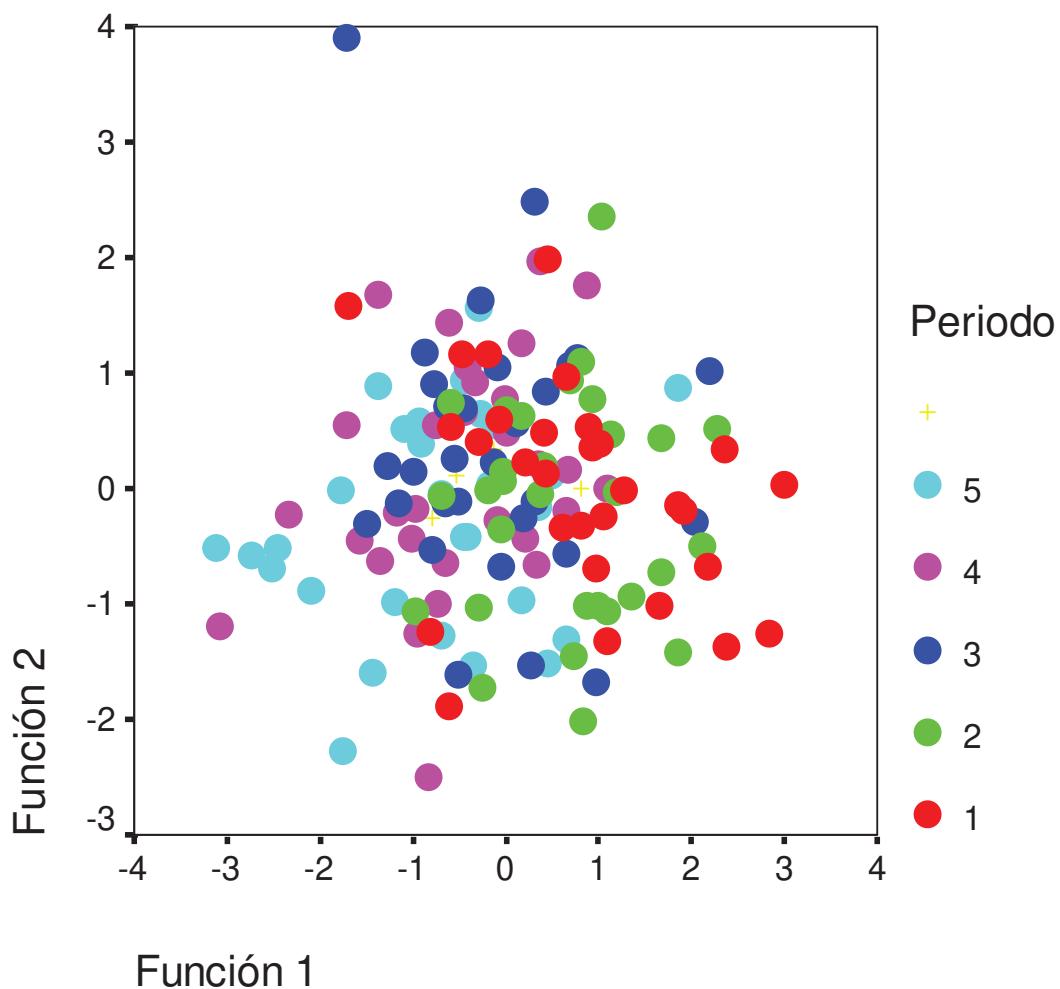
Periodo	Función
	1
Predinástico Inicial (4000 BC)	,806
Predinástico Tardío (3300 BC)	,639
Dinastías 12 y 13 (1850 BC)	-,129
P. Tolemáico (200 BC)	-,522
P. Romano (150 AD)	-,794

Clasificación

Resultados de la clasificación^a

Periodo	Grupo de pertenencia pronosticado					Total
	Predinástico Inicial (4000 BC)	Predinástico Tardío (3300 BC)	Dinastías 12 y 13 (1850 BC)	P. Tolemáico (200 BC)	P. Romano (150 AD)	
Predinástico Inicial (4000 BC)	14	7	5	2	2	30
Predinástico Tardío (3300 BC)	9	10	6	1	4	30
Dinastías 12 y 13 (1850 BC)	4	5	9	5	7	30
P. Tolemáico (200 BC)	2	4	9	3	12	30
P. Romano (150 AD)	2	4	6	5	13	30
Predinástico Inicial (4000 BC)	46,7	23,3	16,7	6,7	6,7	100,0
Predinástico Tardío (3300 BC)	30,0	33,3	20,0	3,3	13,3	100,0
Dinastías 12 y 13 (1850 BC)	13,3	16,7	30,0	16,7	23,3	100,0
P. Tolemáico (200 BC)	6,7	13,3	30,0	10,0	40,0	100,0
P. Romano (150 AD)	6,7	13,3	20,0	16,7	43,3	100,0

a. Clasificados correctamente el 32,7% de los casos agrupados originales.

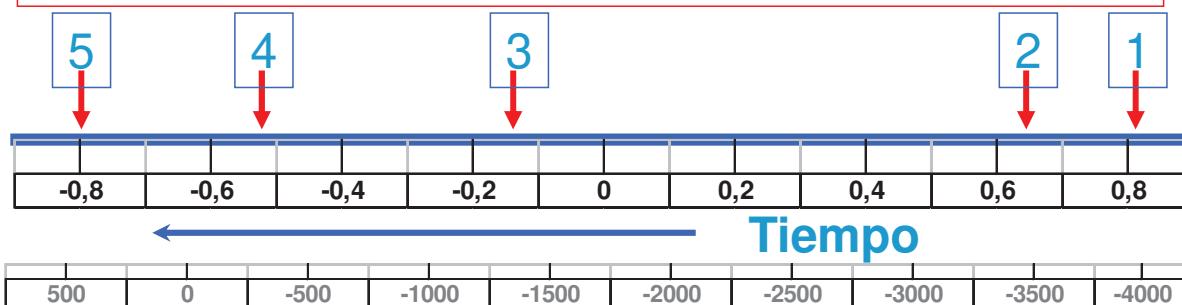


Interpretación

Centroides de los grupos

Periodo	Función
	1
Predinástico Inicial (4000 BC)	,806
Predinástico Tardío (3300 BC)	,639
Dinastías 12 y 13 (1850 BC)	-,129
P. Tolemáico (200 BC)	-,522
P. Romano (150 AD)	-,794

$$Y_1 = 1.771 - 0.134 \text{ AnchC} + 0.028 \text{ AltC} + 0.148 \text{ Man} - 0.036 \text{ Nar}$$



Lectura de datos

```
dat = read.table('data/craneos.txt', header=T)
dat$PERIODO = factor(dat$PERIODO)
head(dat)
```

	PERIODO	ANCHURA	ALTURA	MANDIB	NARIZ
## 1	1	131	138	89	49
## 2	1	125	131	92	48
## 3	1	131	132	99	50
## 4	1	119	132	96	44
## 5	1	136	143	100	54
## 6	1	138	137	89	56

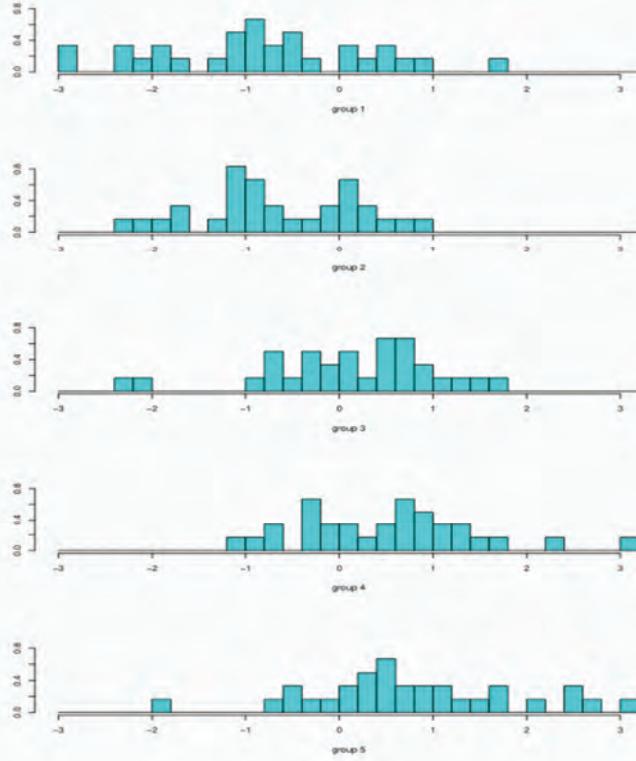
Análisis discriminante

```
library(MASS)
m1 = lda(PERIODO ~ ., data = dat)
m1

## Call:
## lda(PERIODO ~ ., data = dat)
##
## Prior probabilities of groups:
##   1   2   3   4   5 
## 0.2 0.2 0.2 0.2 0.2 
##
## Group means:
##   ANCHURA ALTURA MANDIB NARIZ
## 1     131     134    99.2   50.5
## 2     132     133    99.1   50.2
## 3     134     134    96.0   52.2
## 4     136     132    94.5   52.0
## 5     136     130    93.5   51.3
##
## Coefficients of linear discriminants:
##          LD1      LD2      LD3      LD4
## ANCHURA  0.1336 -0.00232 -0.16760 -0.0337
## ALTURA   -0.0278  0.18333 -0.00546 -0.1026
## MANDIB   -0.1483 -0.04806 -0.12757  0.0538
## NARIZ    0.0364  0.08722  0.00918  0.1746
##
## Proportion of trace:
##      LD1      LD2      LD3      LD4
## 0.8917  0.1021  0.0062  0.0000
```

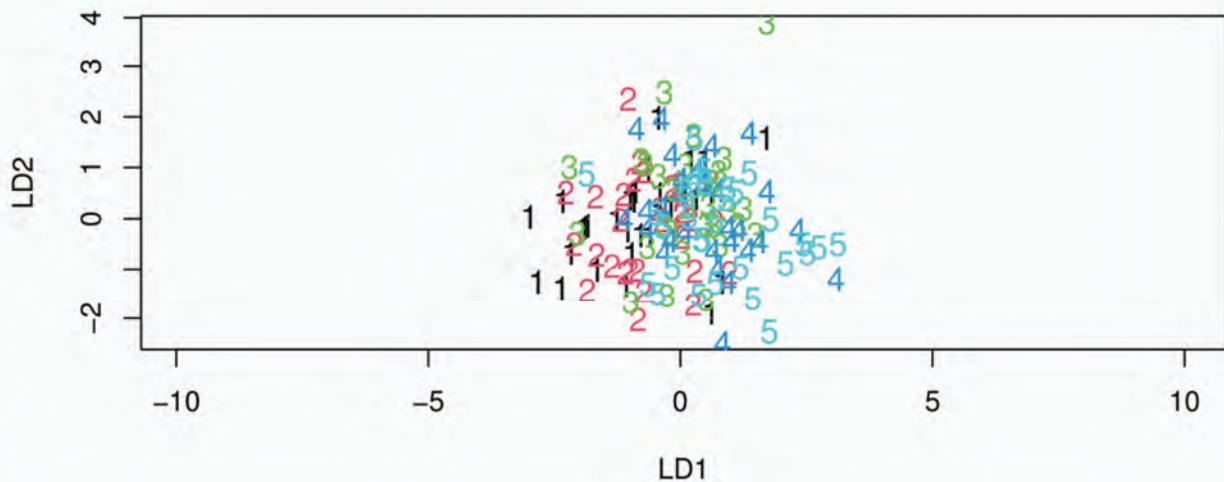
Gráficos 1

```
plot(m1,dimen=1)
```



Gráficos 2

```
plot(m1, dimen=2, col=as.numeric(dat$PERIODO), cex = 1.2)
grid()
```



ldaPlus(): ¿Cuantas funciones elegir?

```
library(multiUS)
m2 = ldaPlus(x = dat[,2:5], grouping=dat$PERIODO)
m2$sigTest
```

```
##          WilksL      F df1 df2      p
## 1 to 4  0.676 3.71367 16 434 1.95e-06
## 2 to 4  0.953 0.78104  9 348 6.34e-01
## 3 to 4  0.997 0.10236  4 288 9.82e-01
## 4 to 4  1.000 0.00122  1 145 9.72e-01
```

Sólo es necesario elegir 1 (las demás p-valor >.05)

Interpretación de la función discriminante

```
m2$standCoefWithin
```

```
##          LD1      LD2      LD3      LD4
## ANCHURA  0.617 -0.0107 -0.7741 -0.156
## ALTURA   -0.135  0.8885 -0.0264 -0.497
## MANDIB   -0.729 -0.2363 -0.6273  0.264
## NARIZ    0.184  0.4414  0.0464  0.884
```

```
m2$centroids
```

```
##          LD1      LD2      LD3      LD4
## 1 -0.806  0.00628  0.08036 -0.00013
## 2 -0.639 -0.18248 -0.07792 -0.00039
## 3  0.129  0.33464 -0.02498  0.00320
## 4  0.522  0.10608 -0.00108 -0.00499
## 5  0.794 -0.26451  0.02362  0.00231
```

- Los centroides están ordenadas por periodos (en LD1)
- Las variables con más peso son ANCHURA y MANDIB: Los cráneos han evolucionado reduciendo la mandíabula y aumentando la anchura del cráneo.

Clasificación

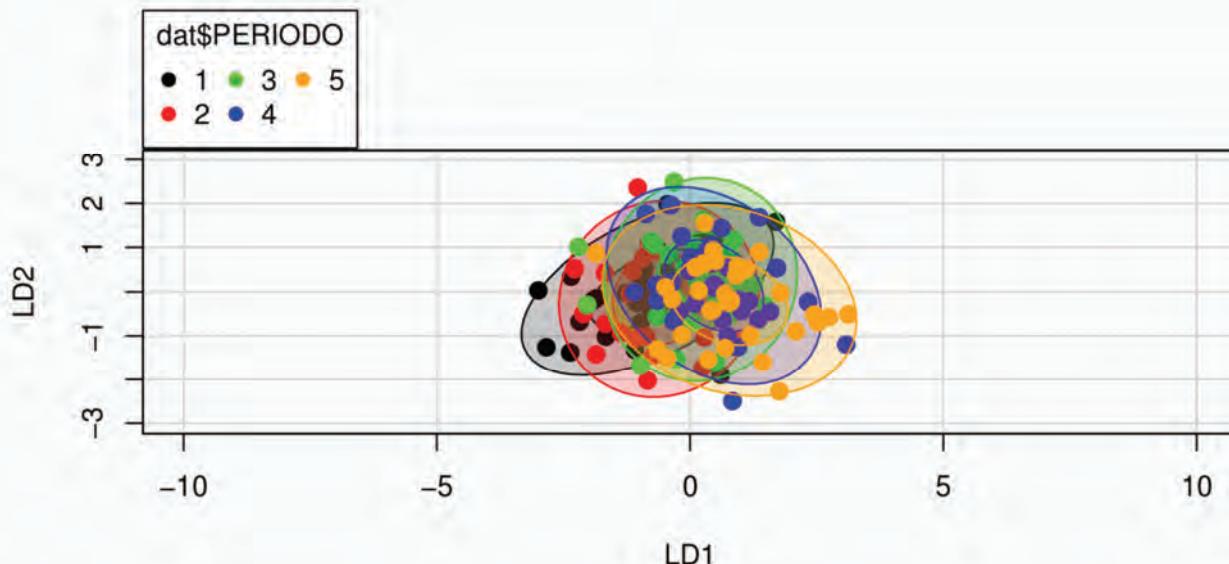
```
m2$class

## $orgTab
##   pred
## orig    1    2    3    4    5 Sum
## 1     14    7    5    2    2  30
## 2      9   10    6    1    4  30
## 3      4    5    9    5    7  30
## 4      2    4    9    3   12  30
## 5      2    4    6    5   13  30
## Sum   31   30   35   16   38 150
##
## $perTab
##   pred
## orig    1      2      3      4      5 Sum
## 1 46.67 23.33 16.67  6.67  6.67 100.00
## 2 30.00 33.33 20.00  3.33 13.33 100.00
## 3 13.33 16.67 30.00 16.67 23.33 100.00
## 4  6.67 13.33 30.00 10.00 40.00 100.00
## 5  6.67 13.33 20.00 16.67 43.33 100.00
##
## $corPer
## [1] 32.7
```

No sirve para clasificar

Gráfico 3

```
library(car)
pred= predict(m1)
scatterplot(pred$x[,1],pred$x[,2],groups= dat$PERIODO,
           pch=c(19,19,19,19,19),cex=1.3,
           xlim=c(-10,10), ylim=c(-3,3),regLine=FALSE,
           ellipse = TRUE,smooth = FALSE,
           col=c("black","red","green","blue","orange"),
           xlab="LD1", ylab="LD2")
```



Extra (paquete klaR)

```
library(klaR)
m3 = greedy.wilks(PERIODO ~ ., data =dat)
m3

## Formula containing included variables:
##
## PERIODO ~ MANDIB + ANCHURA
## <environment: 0x0000000021812338>
##
##
## Values calculated in each step of the selection procedure:
##
##      vars Wilks.lambda F.statistics.overall p.value.overall
## 1  MANDIB      0.814          8.31    4.64e-06
## 2  ANCHURA     0.718          6.50    8.71e-08
##      F.statistics.diff p.value.diff
## 1           8.31    4.64e-06
## 2           4.82    1.12e-03
```

Solo hay dos variables significativas MANDIBULA y ANCHURA

Intrucciones básica para Discriminante I

```
## ----LIBRARIES
library(MASS)
library(multiv)
## ----DATA
iris = read.table('data/lirios.txt',header=T)
iris$Species=factor(iris$Species)

## ---LDA
m1 = lda(Species~.,data=iris)
m1$scaling
## --- Gráficos
plot(m1,dimen=1)
plot(m1,dimen=2,col = as.integer(iris$Species),
     abbrev=1,cex=1.5)
## --- PREDICT
pred = predict(m1)
head(pred$x,12)
## --- PLANO DE MÁXIMA DISCRIMINACIÓN
plot(pred$x[,1],pred$x[,2],
      col=as.integer(iris$Species), pch=19, cex=1.3)
legend(1,2,legend=c("setosa","versicolor","virginia"),
       col=c("black","red","green"),pch=19)
```

```
## --- EXTRA
library(multius)
m2 = ldaPlus(x = iris[,1:4], grouping = iris$Species)
m2$standCoefWithin
m2$centroids
m2$sigTest
m2$eigModel
m2$class
library(klaR)
m3 = greedy.wilks(Species ~ ., data = iris)
## ---nueva observación
flor = data.frame(Sepal.Length=5.7,
                   Sepal.Width=2.9,
                   Petal.Length=4.4,
                   Petal.Width=1.7)
predict(m1,newdata = flor)
```

Discriminante: Conclusión

- El análisis discriminante es un problema similar al análisis de regresión múltiple, pero en este caso, la variable respuesta **Y** es **cualitativa o categórica**. Predecir una respuesta cualitativa a partir de una serie de variables explicativas habitualmente se denomina **clasificar** una observación.
- Existen muchas **técnicas de clasificación**: análisis discriminante (LDA), regresión logística, árboles de clasificación, random forest, redes neuronales, support vector machine.
- El modelo LDA está basado en la distribución normal multivariante de las variables explicativas con matrices de varianzas iguales para los distintos grupos. Éste es el modelo básico con versiones más generales.
- LDA es uno de los modelos fundamentales de clasificación. Proporciona en muchos ejemplos buenos resultados. Tiene la ventaja adicional de ser un modelo capaz de explicar las diferencias entre los distintos grupos.

EJERCICIO: Clasificación de tumores

CURSO MÉTODOLOGÍA DE INVESTIGACIÓN CUANTITATIVA
Análisis Multivariante

DESCRIPCIÓN

En el archivo “wdbc.txt” (Wisconsin (Diagnostic) Breast Cancer Data Set) se proporciona información sobre el análisis morfológico de las células de 569 tumores de cáncer de pecho.

Los datos han sido obtenidos de la siguiente página web

<http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin%28Diagnostic%29>

VARIABLES:

Para cada tumor se proporciona el número de identificación, la diagnosis y la media de 10 características físicas de las células analizadas (al final del documento se proporcionan referencias donde se describe las variables analizadas)

1. ID number
2. Diagnosis (M = malignant, B = benign)
3. radius (mean of distances from center to points on the perimeter)
4. texture (standard deviation of gray-scale values)
5. perimeter
6. area
7. smoothness (local variation in radius lengths)
8. compactness (perimeter² / area - 1.0)
9. concavity (severity of concave portions of the contour)
10. concave points (number of concave portions of the contour)
11. symmetry
12. fractal dimension ("coastline approximation" - 1)

Preguntas:

1. Obtén la función discriminante para diferenciar tumores malignos y benignos. Interpreta los coeficientes.
2. Realiza el contraste e indica si existen diferencias entre los dos tipos de tumores.
3. Representan gráficamente las puntuaciones (scores) de cada tumor en función de su malignidad.
4. Indica de las 10 variables, las más importantes para discriminar entre tumores malignos y benignos.
5. Utiliza la función anterior para clasificar los tumores e indica en una tabla los aciertos y errores que se observan.

Source:

Creators:

1. Dr. William H. Wolberg, General Surgery Dept.
University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu
2. W. Nick Street, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619
3. Olvi L. Mangasarian, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu

Donor:

Nick Street

Data Set Information:

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at [\[Web Link\]](#)

Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

```
ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/
```

Attribute Information:

- 1) ID number
- 2) Diagnosis (M = malignant, B = benign)
- 3-32)

Ten real-valued features are computed for each cell nucleus:

- a) radius (mean of distances from center to points on the perimeter)
- b) texture (standard deviation of gray-scale values)
- c) perimeter
- d) area
- e) smoothness (local variation in radius lengths)

- f) compactness (perimeter² / area - 1.0)
- g) concavity (severity of concave portions of the contour)
- h) concave points (number of concave portions of the contour)
- i) symmetry
- j) fractal dimension ("coastline approximation" - 1)

Relevant Papers:

First Usage:

W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.
[\[Web Link\]](#)

O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.
[\[Web Link\]](#)

Medical literature:

W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171.
[\[Web Link\]](#)

W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Image analysis and machine learning applied to breast cancer diagnosis and prognosis. Analytical and Quantitative Cytology and Histology, Vol. 17 No. 2, pages 77-87, April 1995.

W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian. Computerized breast cancer diagnosis and prognosis from fine needle aspirates. Archives of Surgery 1995;130:511-516.
[\[Web Link\]](#)

W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian. Computer-derived nuclear features distinguish malignant from benign breast cytology. Human Pathology, 26:792--796, 1995.
[\[Web Link\]](#)

EJERCICIO: Cuerpo humano

CURSO MÉTODOLOGÍA DE INVESTIGACIÓN CUANTITATIVA Análisis Multivariante

Enunciado

Con los datos del archivo **cuerpo.txt** vamos a realizar el análisis de componentes principales de las variables con medidas de contorno (empiezan por C)

```
## [1] "C_hombros" "C_pecho"    "C_cintura" "C_abdomen" "C_cadera"   "C_muslo"
## [7] "C_biceps"  "C_brazo"     "C_rodilla"  "C_gemelo"   "C_tobillo"  "C_muneca"
```

En el documento adjunto “Descripción Dataset Cuerpo” se proporciona la información de los datos que contiene el archivo “cuerpo.txt”. Hay distintas versiones del archivo “cuerpo.txt”, importante utilizar el que se proporciona con este enunciado.

1. Para las 12 variables anteriores calcula la matriz de correlaciones utilizando solo los datos de mujeres. Comprueba con los gráficos de dispersión si las relaciones son lineales. Indica las tres correlaciones más altas (entre qué variables se producen). Indican las tres correlaciones más bajas (entre qué variables se producen). **(1 punto)**
2. Repite el apartado 1 para los datos de hombres. **(1 punto)**
3. Realiza el análisis de componentes principales (con los datos estandarizados) para las 12 variables de las medidas de contorno de las mujeres. Toma la solución con dos componentes. **(1.5 punto)**
 - a. ¿Qué porcentaje de la variabilidad total de las 12 variables está explicada por los dos primeros componentes?
 - b. ¿Qué variable de las 12 está mejor explicada? Interpreta este resultado.
 - c. ¿Qué variable de las 12 está peor explicada? Interpreta este resultado.
 - d. Representa en un gráfico de dimensión 2 los scores de cada individuo. Interpreta la distribución de los puntos. Qué personas están a la derecha, a la izquierda, arriba y abajo del gráfico.
4. Repite el análisis de la pregunta 3 aplicado a hombres. **(1.5 punto)**
5. Realiza el análisis discriminante para comparar las 12 medidas de hombres y mujeres. Proporciona la función discriminante y la función discriminante estandarizada. Indica las 5 variables más importantes para discriminar entre mujeres y hombres. **(2 punto)**
6. Utiliza la instrucción *plot()* para representar las puntuaciones discriminante. Con ayuda del gráfico y de la función discriminante estandarizada interpreta las cinco variables que discriminan entre hombres y mujeres. **(1 punto)**

7. Obtén la *matriz de confusión* y explica si la función discriminante es útil para clasificar las observaciones entre hombres y mujeres. **(1 punto)**
8. Realiza la validación del método de clasificación siguiendo las siguientes instrucciones. **(1 punto)**
 - a. Toma una muestra **mtrain** al azar del 75% de las observaciones. Las restantes observaciones las denominaremos **mtest**.
 - b. Estima el modelo de clasificación utilizando **mtrain**
 - c. Aplica el modelo **mtrain** para clasificar las observaciones reservadas **mtest**
 - d. Obtén el porcentaje de observaciones mal clasificadas en el paso **c**.
 - e. Repite 100 veces los pasos **a** hasta **d** y guarda los errores de clasificación de cada simulación. Obtén la media de los errores. Proporciona el programa y el error medio obtenido en tú simulación.

Tarea 2: Clasificación de Tumores

Solución

1 Descripción

En el archivo “wdbc.txt” (Wisconsin Diagnostic Breast Cancer Data Set) se proporciona información sobre el análisis morfológico de las células de 569 tumores de cáncer de pecho. Los datos han sido obtenidos de la siguiente página web <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>

VARIABLES:

Para cada tumor se proporciona el número de identificación, la diagnosis y la media de 10 características físicas de las células analizadas (al final del documento se proporcionan referencias donde se describe las variables analizadas)

1. ID number
2. Diagnosis (M = malignant, B = benign)
3. radius (mean of distances from center to points on the perimeter)
4. texture (standard deviation of gray-scale values)
5. perimeter
6. area
7. smoothness (local variation in radius lengths)
8. compactness (perimeter² / area - 1.0)
9. concavity (severity of concave portions of the contour)
10. concave points (number of concave portions of the contour)
11. symmetry
12. fractal dimension (“coastline approximation” - 1)

```
library(candisc)
library(MASS)
library(klaR)
dat = read.table("wdbc.txt", header=TRUE)
```

2 Apartado 1

Obtén la función discriminante para diferenciar tumores malignos y benignos. Interpreta los coeficientes.

```
m1 = lda(dat$diagnosis ~ ., data=dat[, 2:12])
m1
```

```

## Call:
## lda(dat$diagnosis ~ ., data = dat[, 2:12])
##
## Prior probabilities of groups:
##      B      M
## 0.6274165 0.3725835
##
## Group means:
##      radius texture perimeter area smoothness compactness concavity
## B 12.14652 17.91476 78.07541 462.7902 0.09247765 0.08008462 0.04605762
## M 17.46283 21.60491 115.36538 978.3764 0.10289849 0.14518778 0.16077472
##      concave_points symmetry fractal_dimension
## B      0.02571741 0.174186      0.06286739
## M      0.08799000 0.192909      0.06268009
##
## Coefficients of linear discriminants:
##                               LD1
## radius                  2.173832578
## texture                 0.097479319
## perimeter                -0.243883158
## area                    -0.004235635
## smoothness                8.610211091
## compactness                0.431476344
## concavity                 3.592356858
## concave_points            28.529778564
## symmetry                  4.489073661
## fractal_dimension        -0.529214778

```

En el resumen del modelo **m1** se proporciona el vector con los pesos (coeficientes) de la función discriminante lineal. Se observa que el coeficiente mayor corresponde a la variable **concave_points** igual a 28.52. Esto no necesariamente indica que esa sea la variable “más importante” para discriminar entre tumores malignos y benignos. Los coeficientes dependen de las unidades con la que se mide cada una de las variables explicativas.

Para tener una idea de las variables importantes, un primer paso es obtener los coeficientes estandarizados. (las variables se han estandarizado y todas las variables tienen varianza 1). Los coeficientes son comparables, ahora son adimensionales. Hay varias formas de “estandarizar” los coeficientes:

Una manera sencilla es:

```
m1$scaling*sapply(dat[,3:12],sd)
```

```

##                               LD1
## radius                  7.660692144
## texture                 0.419262039
## perimeter                -5.926112235
## area                    -1.490579787
## smoothness                0.121095112
## compactness                0.022787456
## concavity                 0.286382002
## concave_points            1.107036571
## symmetry                  0.123064728
## fractal_dimension        -0.003736448

```

En este caso utiliza la desviación típica (total) de cada variable.

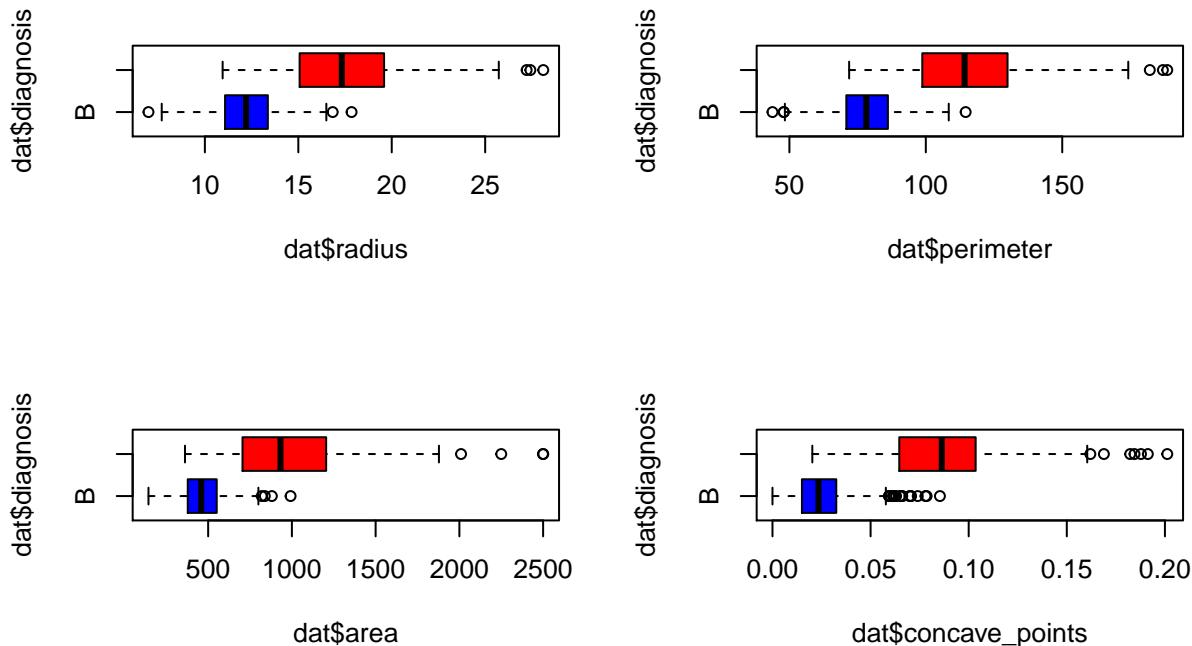
Otra alternativa utiliza como desviación típica, la media de las desviaciones típicas de las variables en cada grupo (pooled within-group variances). Para evitar hacer los cálculo, utilizo la que proporciona el paquete **candisc**, pero hay que tener cuidado, porque la solución tiene los signos cambiados respecto a la obtenida previamente:

```
m2 = manova(as.matrix(dat[,3:12])~dat$diagnosis)
m3 = candisc(m2)
-m3$coeffs.std
```

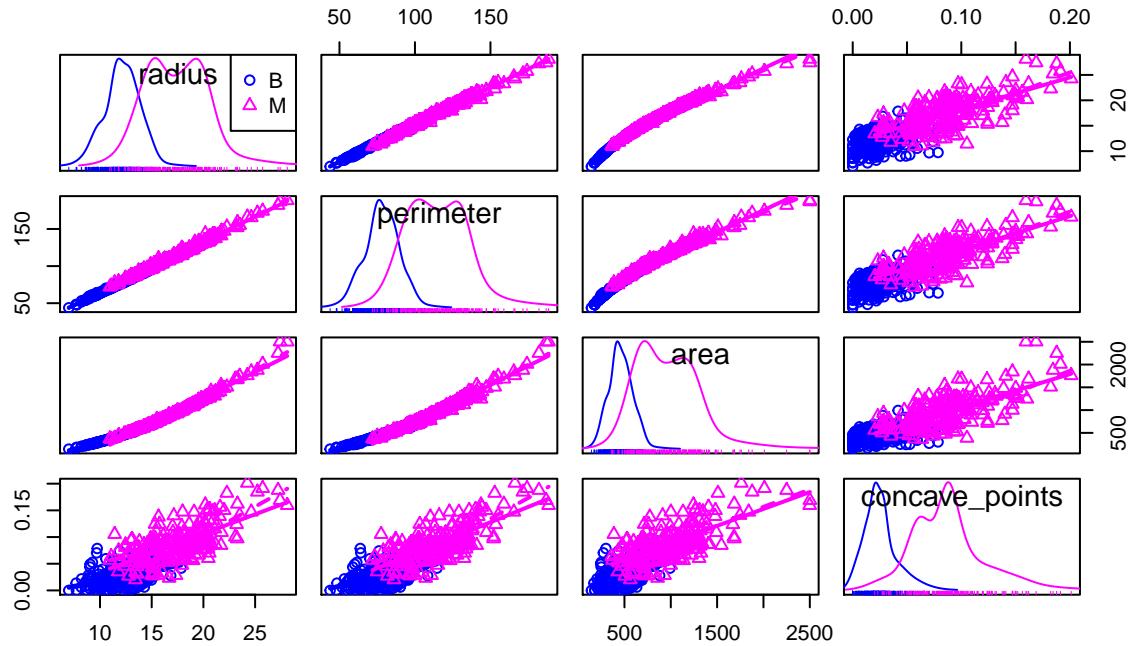
```
##                               Can1
## radius                  5.240059631
## texture                 0.381754333
## perimeter                -3.972190686
## area                   -1.052120298
## smoothness                0.113142739
## compactness                0.018305060
## concavity                  0.205715508
## concave_points              0.698022237
## symmetry                  0.116251670
## fractal_dimension          -0.003739434
```

La interpretación es similar, las variables con coeficientes mayores en valor absoluto son **radius**, **perimeter**, **area** y **concave_points**.

```
par(mfrow=c(2,2))
boxplot(dat$radius~dat$diagnosis,col=c("blue","red"),horizontal = TRUE)
boxplot(dat$perimeter~dat$diagnosis,col=c("blue","red"),horizontal = TRUE)
boxplot(dat$area~dat$diagnosis,col=c("blue","red"),horizontal = TRUE)
boxplot(dat$concave_points~dat$diagnosis,col=c("blue","red"),horizontal = TRUE)
```



```
scatterplotMatrix(dat[,c("radius","perimeter", "area", "concave_points")], groups = dat$diagnosis)
```



Interpretación:

- Se observa en el boxplot que los valores de las cuatro variables **radius**, **perimeter**, **area** y **concave_points** son diferentes para tumores malignos y benignos. En general, valores altos de cualquiera de las cuatro variables indican que el tumor es maligno.
- Como se aprecia en el gráfico de correlaciones, las cuatro variables están muy correlacionadas, especialmente las tres primeras **radius**, **perimeter** y **area**. Esta relación es obvia y lógica cuando las células tienen forma circular, aunque puede ser diferente para otras formas con perímetros más irregulares (esta es un poco la idea de los fractales).
- Cuando las variables explicativas están muy correlacionadas, como en este caso, los coeficientes y signos de la función discriminante (estandarizada o sin estandarizar) son difíciles de interpretar. Para su interpretación es preciso tener en cuenta que los coeficientes miden efectos marginales, a igualdad del resto de las variables en la ecuación. Así la variable **radius** tiene coeficiente (peso) positivo: a igualdad del resto de las variables, un mayor radio aumenta la probabilidad de ser maligno. El perímetro tiene coeficiente negativo, indica que a igualdad del resto, mayor área aumenta la probabilidad de ser benigno. Los signos son debidos a las correlaciones entre las variables explicativas. Veremos en el último apartado si la regla de clasificación es buena.

3 Apartado 2

Realiza el contraste e indica si existen diferencias entre los dos tipos de tumors.

```
summary(m2)
```

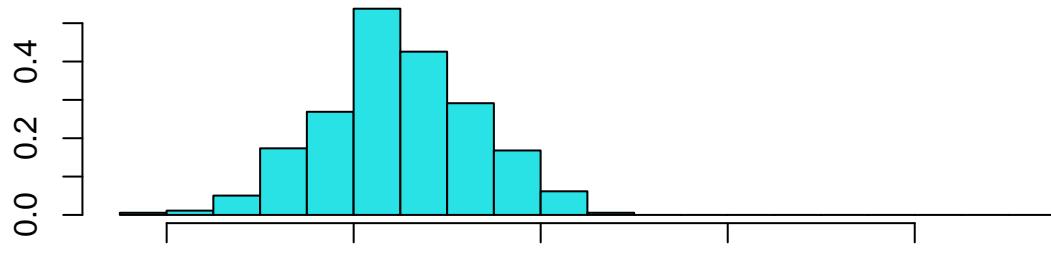
```
##               Df Pillai approx F num Df den Df    Pr(>F)
## dat$diagnosis   1 0.68276   120.09     10    558 < 2.2e-16 ***
## Residuals      567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Las variables medidas permiten discriminar significativamente entre los dos tipos de tumores B y M.

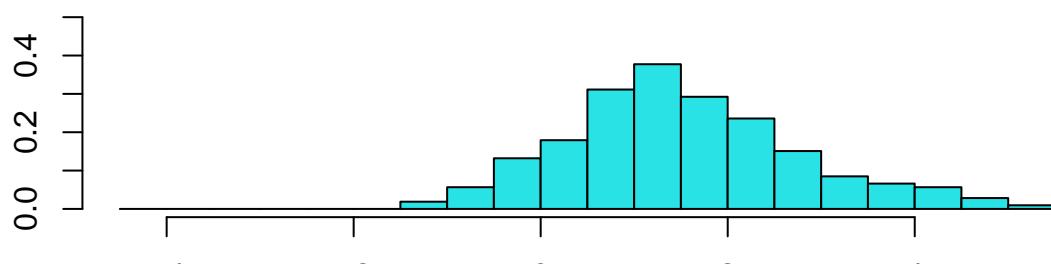
4 Apartado 3

Representan gráficamente las puntuaciones (scores) de cada tumor en función de su malignidad.

```
plot(m1)
```



group B



group M

Las puntuaciones o scores de cada individuo se obtiene utilizando la función `pred=predict(m1)`. Los valores están en el vector `pred$x`. La figura anterior son los histogramas de `pred$x` diferenciando los tumores benignos de los malignos.

Los centroides de los dos grupos se puede obtener como:

```
pred=predict(m1)
tapply(pred$x,dat$diagnosis,mean)
```

```
##          B          M
## -1.128531  1.900403
```

5 Apartado 4

Indica de las 10 variables, las más importantes para discriminar entre tumores malignos y benignos.

```
m4 = greedy.wilks(diagnosis~,data=dat[,2:12])
m4
```

```
## Formula containing included variables:
##
## diagnosis ~ concave_points + texture + radius + area + perimeter +
##           symmetry + concavity + smoothness
## <environment: 0x00000000191f10f8>
##
##
## Values calculated in each step of the selection procedure:
##
##           vars Wilks.lambda F.statistics.overall p.value.overall
## 1 concave_points    0.3968709      861.6760  7.101150e-116
## 2         texture    0.3584931      506.4155  8.280721e-127
## 3         radius    0.3409979      363.9672  1.551483e-131
## 4         area     0.3279516      288.9415  5.473830e-135
## 5        perimeter   0.3223809      236.6763  8.072064e-136
## 6        symmetry   0.3196355      199.3755  1.191176e-135
## 7       concavity   0.3185161      171.4703  6.589030e-135
## 8       smoothness   0.3172415      150.6521  2.929155e-134
##   F.statistics.diff p.value.diff
## 1      861.676020 7.101150e-116
## 2      60.592120 3.352874e-14
## 3      28.987779 1.068891e-07
## 4      22.436610 2.750510e-06
## 5      9.728670 1.906777e-03
## 6      4.827110 2.842227e-02
## 7      1.971425 1.608491e-01
## 8      2.250052 1.341725e-01
```

Las siguientes ocho variables **concave_points** , **texture** , **radius** , **area** , **perimeter**, **symmetry**, **concavity**, **smoothness** y en este orden son las que tienen poder de discriminación. Las variables **compactness** y **fractal_dimension** no tienen efecto significativo que contribuyan a mejorar el modelo y pueden eliminarse del mismo.

6 Apartado 6

Utiliza la función anterior para clasificar los tumores e indica en una tabla los aciertos y errores que se observan.

```
# m5 = lda(m4$formula,data=dat[,2:12])
m5 = lda(diagnosis ~ concave_points + texture + radius + area
          + perimeter + symmetry + concavity + smoothness,
          data = dat)
m5

## Call:
## lda(diagnosis ~ concave_points + texture + radius + area + perimeter +
##       symmetry + concavity + smoothness, data = dat)
##
## Prior probabilities of groups:
##           B         M
## 0.6274165 0.3725835
##
## Group means:
##   concave_points  texture    radius     area perimeter symmetry concavity
## B      0.02571741 17.91476 12.14652 462.7902 78.07541 0.174186 0.04605762
## M      0.08799000 21.60491 17.46283 978.3764 115.36538 0.192909 0.16077472
##   smoothness
## B 0.09247765
## M 0.10289849
##
## Coefficients of linear discriminants:
##                               LD1
## concave_points 28.573678864
## texture        0.097563416
## radius         2.134644004
## area          -0.004284658
## perimeter     -0.237144491
## symmetry       4.535711363
## concavity      3.636542194
## smoothness     8.753553786

pred = predict(m5)

t1=table(Obs=dat$diagnosis
          ,Prev=pred$class)
(t2 = addmargins(t1))

##          Prev
## Obs      B    M Sum
## B    351    6 357
## M    29 183 212
## Sum 380 189 569
```

```

aciertos= (t1[1,1]+t1[2,2])/sum(t1)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2), "% ")
## [1] "Aciertos = 93.85% -- Errores = 6.15% "

```

Vamos a comparar el % de errores de clasificación de cuatro modelos:

- 1 El modelo con las variables significativas (m5)
- 2 El modelo con todas las variables (m1)
- 3 El modelo con las cuatro variables con más peso en la función discriminante estandarizada (m6)
- 4 El modelo eliminando de m5 las variables area y perimeter que están muy correlacionadas con radius (m7)

A continuación se completa este cálculo:

```

p1=predict(m1)
p5 = predict(m5)
m6 = lda(diagnosis ~ concave_points + radius + area
          + perimeter , data = dat)
m7 = lda(diagnosis ~ concave_points + texture + radius +
          symmetry + concavity + smoothness, data = dat)
p6 = predict(m6)
p7 = predict(m7)
t1=table(Obs=dat$diagnosis,Prev=p1$class)
e1 = (t1[1,2]+t1[2,1])/sum(t1)
t5=table(Obs=dat$diagnosis,Prev=p5$class)
e5 = (t5[1,2]+t5[2,1])/sum(t5)
t6=table(Obs=dat$diagnosis,Prev=p6$class)
e6 = (t6[1,2]+t6[2,1])/sum(t6)
t7=table(Obs=dat$diagnosis,Prev=p7$class)
e7 = (t7[1,2]+t7[2,1])/sum(t7)
round(rbind(e1,e5,e6,e7)*100,2)

```

```

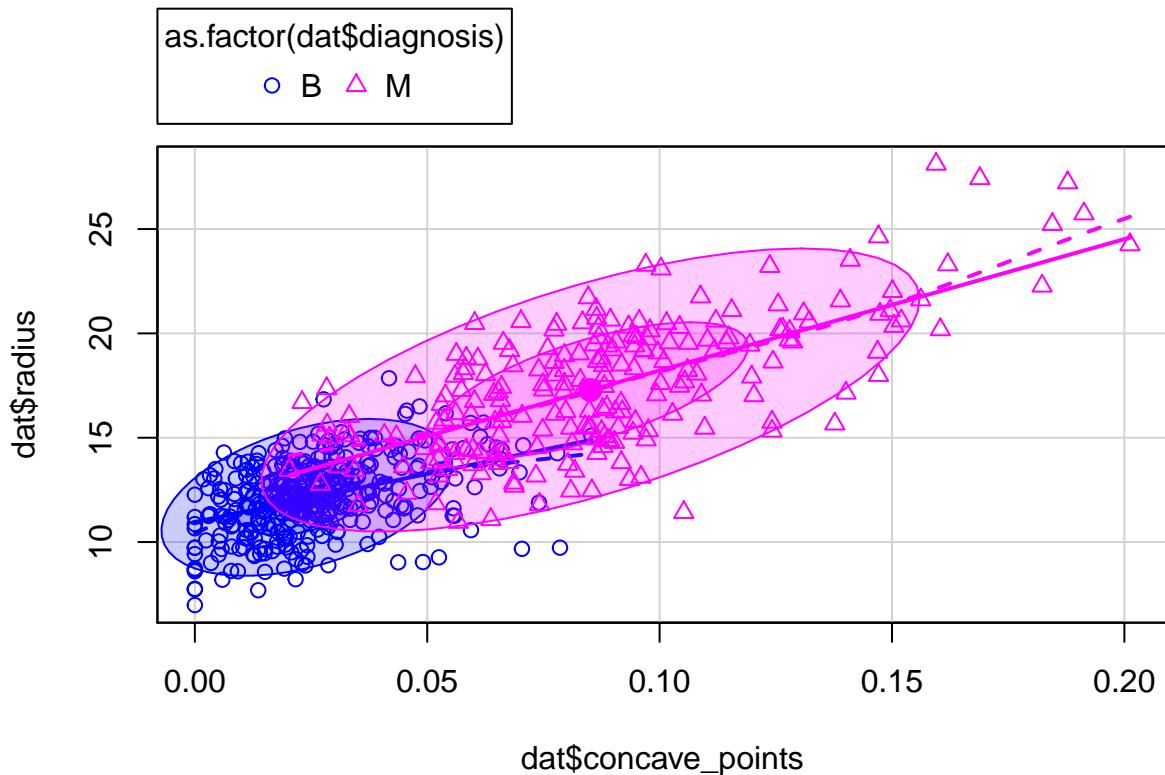
##      [,1]
## e1 6.15
## e5 6.15
## e6 9.31
## e7 6.68

```

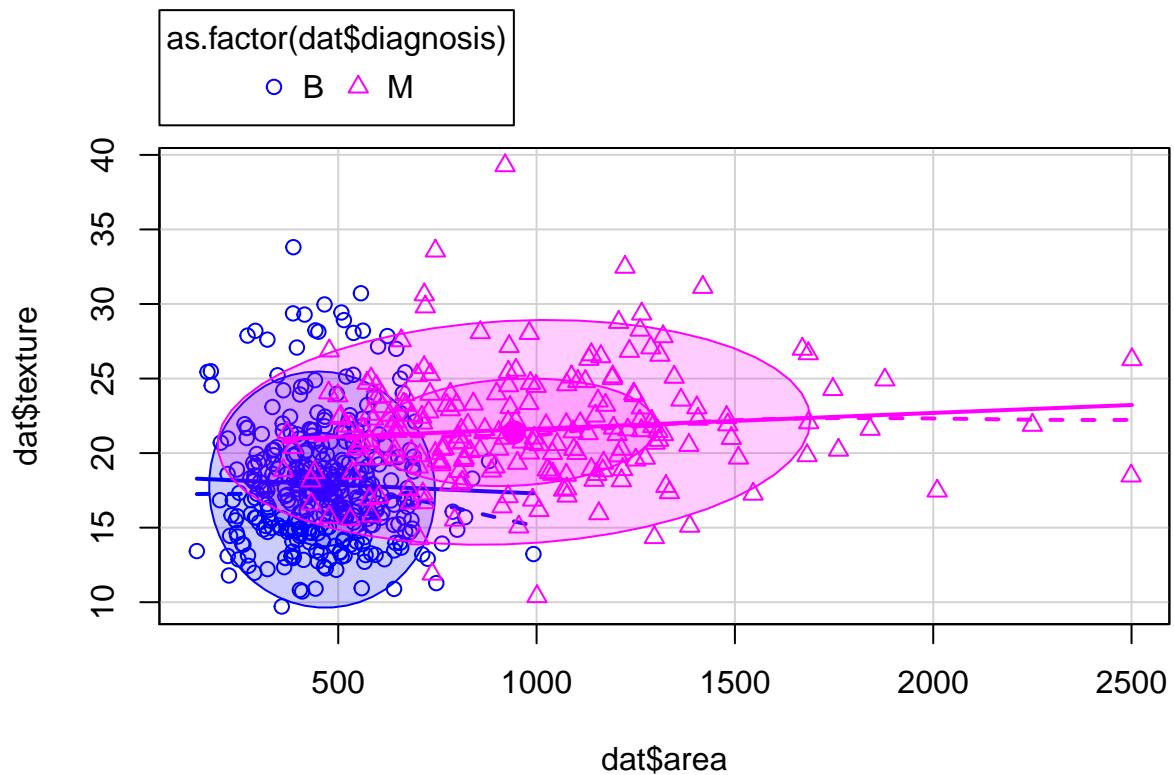
Entre el modelo m5 y m6 no existen diferencias. Confirma que las dos variables de más que tiene m1 no aportan información en la clasificación de tumores. El modelo m6 tiene un mayor porcentaje de error. El modelo m7 es parecido a m5, ligeramente peor (las variables area, perimeter y radius son prácticamente iguales a efectos de discriminación)

A continuación se hace dos gráficos de dispersión para ilustrar las diferencias entre los tumores en algunas de las variables.

```
par(mfrow=c(1,2))
scatterplot(dat$concave_points,dat$radius,
           groups =as.factor(dat$diagnosis),ellipse=TRUE)
```



```
scatterplot(dat$area, dat$texture,
           groups =as.factor(dat$diagnosis),ellipse=TRUE)
```



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

7 Extra: Técnicas de Validación cruzada (cross-validation)

El procedimiento anterior utilizado para valorar una regla de clasificación tiende a sobre-estimar el porcentaje de acierto. La razón es sencilla: utilizamos los mismos datos para evaluar la regla de clasificación que los utilizados para establecer (estimar) la regla de clasificación.

Para evitar este inconveniente, vamos a dividir la muestra en dos, una parte de ella la utilizamos para estimar el modelo y la otra parte para evaluar el modelo.

7.1 Modelo m5

```
set.seed(97654)
n = dim(dat)[1]
mue = sample(1:n, round(0.75*n))
mod = lda(diagnosis ~ concave_points + texture + radius
          + area + perimeter + symmetry + concavity
          + smoothness, data = dat[mue,])
pre = predict(mod, newdata = dat[-mue,])
tp=table(dat$diagnosis[-mue], pre$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
```

```

errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

```

```
## [1] "Aciertos = 92.96% -- Errores = 7.04% "
```

7.2 Modelo m1

```

mod = lda(diagnosis ~ concave_points + texture + radius
          + area + perimeter + symmetry + concavity
          + smoothness + compactness+ fractal_dimension, data = dat[mue,])
pre = predict(mod,newdata = dat[-mue,])
tp=table(dat$diagnosis[-mue],pre$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

```

```
## [1] "Aciertos = 92.96% -- Errores = 7.04% "
```

7.3 Modelo m6

```

mod = lda(diagnosis ~ concave_points + radius
          + area + perimeter , data = dat[mue,])
pre = predict(mod,newdata = dat[-mue,])
tp=table(dat$diagnosis[-mue],pre$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

```

```
## [1] "Aciertos = 88.73% -- Errores = 11.27% "
```

7.4 Modelo m7

```

mod = lda(diagnosis ~ concave_points + texture + radius
          + symmetry + concavity + smoothness, data = dat[mue,])
pre = predict(mod,newdata = dat[-mue,])
tp=table(dat$diagnosis[-mue],pre$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

```

```
## [1] "Aciertos = 90.85% -- Errores = 9.15% "
```

7.5 Validación (Leave one out)

Otra técnica de validación consiste en eliminar una observación de la muestra, estimar el modelo con las restantes y clasificar la observación extraída. La operación se repite para todas las observaciones de la muestra. A este método se denomina LEAVE ONE OUT (LOO). La función `lda()` la ejecuta si añadimos la opción `CV=TRUE`. Los resultados de la clasificación se muestran en la salida `mod$class`.

```
mod2 = lda(diagnosis ~ concave_points + texture + radius
           + area + perimeter + symmetry + concavity
           + smoothness + compactness + fractal_dimension, data = dat,CV=TRUE)
tp=table(dat$diagnosis, mod2$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

## [1] "Aciertos = 93.5% -- Errores = 6.5% "

mod2 = lda(diagnosis ~ concave_points + texture + radius
           + area + perimeter + symmetry + concavity
           + smoothness, data = dat,CV=TRUE)
tp=table(dat$diagnosis, mod2$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

## [1] "Aciertos = 93.5% -- Errores = 6.5% "

mod2 = lda(diagnosis ~ concave_points + radius
           + area + perimeter , data = dat,CV=TRUE)
tp=table(dat$diagnosis, mod2$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

## [1] "Aciertos = 90.69% -- Errores = 9.31% "

mod2 = lda(diagnosis ~ concave_points + texture
           + radius + symmetry + concavity +
           smoothness, data = dat,CV=TRUE)
tp=table(dat$diagnosis, mod2$class)
aciertos = (tp[1,1]+tp[2,2])/sum(tp)
errores = 1 - aciertos
paste0("Aciertos = ",round(100*aciertos,2),"% --", " Errores = ", round(100*errores,2),"% ")

## [1] "Aciertos = 93.15% -- Errores = 6.85% "
```

8 Source:

Creators:

1. Dr. William H. Wolberg, General Surgery Dept. University of Wisconsin, Clinical Sciences Center Madison, WI 53792 wolberg '@' eagle.surgery.wisc.edu
2. W. Nick Street, Computer Sciences Dept. University of Wisconsin, 1210 West Dayton St., Madison, WI 53706 street '@' cs.wisc.edu 608-262-6619
3. Olvi L. Mangasarian, Computer Sciences Dept. University of Wisconsin, 1210 West Dayton St., Madison, WI 53706 olvi '@' cs.wisc.edu

Donor:

Nick Street Data Set Information: Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at [Web Link]

Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server: `ftp ftp.cs.wisc.edu cd math-prog/cpo-dataset/machine-learn/WDBC/` Attribute Information: 1) ID number 2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

- a) radius (mean of distances from center to points on the perimeter)
- b) texture (standard deviation of gray-scale values)
- c) perimeter
- d) area
- e) smoothness (local variation in radius lengths)
- f) compactness ($\text{perimeter}^2 / \text{area} - 1.0$)
- g) concavity (severity of concave portions of the contour)
- h) concave points (number of concave portions of the contour)
- i) symmetry
- j) fractal dimension ("coastline approximation" - 1)

Relevant Papers: First Usage:

W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993. [Web Link]

O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995. [Web Link]

Medical literature:

W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171. [Web Link]

W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Image analysis and machine learning applied to breast cancer diagnosis and prognosis. *Analytical and Quantitative Cytology and Histology*, Vol. 17 No. 2, pages 77-87, April 1995.

W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian. Computerized breast cancer diagnosis and prognosis from fine needle aspirates. *Archives of Surgery* 1995;130:511-516. [Web Link]

W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian. Computer-derived nuclear features distinguish malignant from benign breast cytology. *Human Pathology*, 26:792–796, 1995. [Web Link]

Componentes Principales y Análisis Discriminante

Tarea 2: Análisis de datos

Enunciado

Con los datos del archivo **cuerpo.txt** vamos a realizar el análisis de componentes principales de las variables con medidas de contorno (empiezan por C)

```
## [1] "C_hombros" "C_pecho"    "C_cintura"  "C_abdomen" "C_cadera"   "C_muslo"
## [7] "C_biceps"   "C_brazo"     "C_rodilla"   "C_gemelo"  "C_tobillo"  "C_muneca"
```

En el documento adjunto “Descripción Dataset Cuerpo” se proporciona la información de los datos que contiene el archivo “cuerpo.txt”. Hay distintas versiones del archivo “cuerpo.txt”, importante utilizar el que se proporciona con este enunciado.

1. Para las 12 variables anteriores calcula la matriz de correlaciones utilizando solo los datos de mujeres. Comprueba con los gráficos de dispersión si las relaciones son lineales. Indica las tres correlaciones más altas (entre qué variables se producen). Indican las tres correlaciones más bajas (entre qué variables se producen). **(1.5 punto)**
2. Repite el apartado 1 para los datos de hombres. **(1.5 punto)**
3. Realiza el análisis de componentes principales (con los datos estandarizados) para las 12 variables de las medidas de contorno de las mujeres. Toma la solución con dos componentes. **(1.5 punto)**
 - a. ¿Qué porcentaje de la variabilidad total de las 12 variables está explicada por los dos primeros componentes?
 - b. ¿Qué variable de las 12 está mejor explicada? Interpreta este resultado.
 - c. ¿Qué variable de las 12 está peor explicada? Interpreta este resultado.
 - d. Representa en un gráfico de dimensión 2 los scores de cada individuo. Interpreta la distribución de los puntos. Qué personas están a la derecha, a la izquierda, arriba y abajo del gráfico.
4. Repite el análisis de la pregunta 3 aplicado a hombres. **(1.5 punto)**
5. Realiza el análisis discriminante para comparar las 12 medidas de hombres y mujeres. Proporciona la función discriminante y la función discriminante estandarizada. Indica las 5 variables que tienen más peso en la función discriminante. **(2 punto)**
6. Utiliza la instrucción *plot()* para representar las puntuaciones discriminante. Con ayuda del gráfico y de la función discriminante estandarizada interpreta las cinco variables que discriminan entre hombres y mujeres. **(1 punto)**
7. Obtén la *matriz de confusión* y explica si la función discriminante es útil para clasificar las observaciones entre hombres y mujeres. **(1 punto)**
8. Realiza la validación del método de clasificación siguiendo las siguientes instrucciones. **(Opcional)**
 - a. Toma una muestra **mtrain** al azar del 75% de las observaciones. Las restantes observaciones las denominaremos **mtest**.
 - b. Estima el modelo de clasificación utilizando **mtrain**
 - c. Aplica el modelo **mtrain** para clasificar las observaciones reservadas **mtest**

- d. Obtén el porcentaje de observaciones mal clasificadas en el paso **c**.
- e. Repite 100 veces los pasos **a** hasta **d** y guarda los errores de clasificación de cada simulación. Obtén la media de los errores. Proporciona el programa y el error medio obtenido en tú simulación.

SOLUCIÓN

1,2,3 y 4 : Apartados resueltos en el Tema 3 Componentes Principales

5. Realiza el análisis discriminante para comparar las 12 medidas de hombres y mujeres. Proporciona la función discriminante y la función discriminante estandarizada. Indica las 5 variables que tienen más peso en la función discriminante.

```
m = lda(dat$sexo~, data=dat[,c(sel,25)])
m

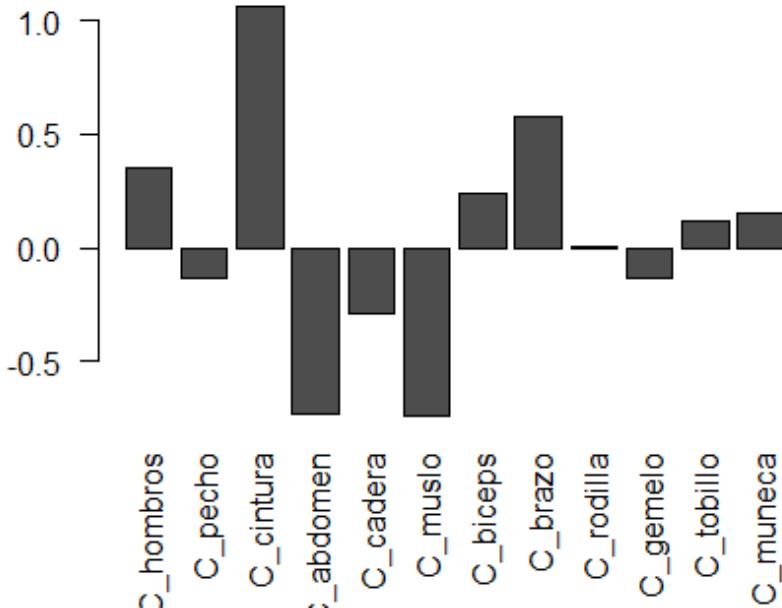
## Call:
## lda(dat$sexo ~ ., data = dat[, c(sel, 25)])
##
## Prior probabilities of groups:
##      Mujer      Hombre 
## 0.5128205 0.4871795 
##
## Group means:
##      C_hombros  C_pecho  C_cintura  C_abdomen  C_cadera  C_muslo  C_biceps 
## Mujer    100.3038  86.0600  69.80346  83.74577  95.65269  57.19577  28.09731 
## Hombre   116.5016 100.9899  84.53320  87.66235  97.76316  56.49798  34.40364 
##      C_brazo  C_rodilla  C_gemelo  C_tobillo  C_muneca 
## Mujer    23.76038  35.26000  35.00615  21.20577  15.05923 
## Hombre   28.24049  37.19555  37.20688  23.15911  17.19028 
##
## Coefficients of linear discriminants:
##                               LD1
##      C_hombros  0.054621610
##      C_pecho   -0.020475659
##      C_cintura  0.129471126
##      C_abdomen -0.079517756
##      C_cadera  -0.043776515
##      C_muslo   -0.165361431
##      C_biceps  0.084815722
##      C_brazo    0.335274258
##      C_rodilla  0.003955319
##      C_gemelo  -0.049934031
##      C_tobillo  0.073001483
##      C_muneca  0.169905976

s = sqrt((sapply(dat0[,sel],sd)^2*259+ sapply(dat1[,sel],sd)^2*246)/(259+246))
cbind(m$scaling,m$scaling*s)

##                               LD1          LD1
##      C_hombros  0.054621610  0.354164985
##      C_pecho   -0.020475659 -0.137115183
##      C_cintura  0.129471126  1.060550757
##      C_abdomen -0.079517756 -0.733731780
##      C_cadera  -0.043776515 -0.289063944
##      C_muslo   -0.165361431 -0.735958137
##      C_biceps  0.084815722  0.241344180
##      C_brazo    0.335274258  0.580097876
##      C_rodilla  0.003955319  0.009628226
##      C_gemelo  -0.049934031 -0.131265092
##      C_tobillo  0.073001483  0.115843379
##      C_muneca  0.169905976  0.149251789
```

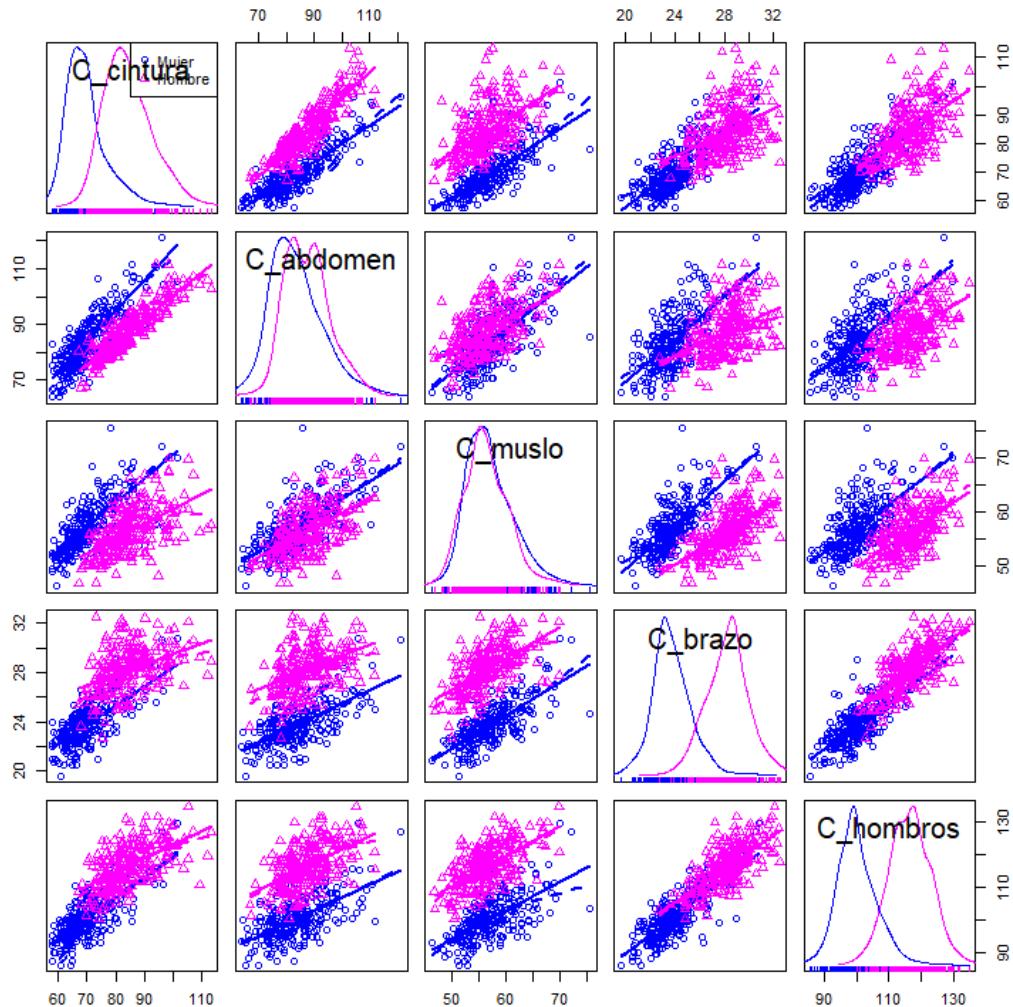
Las variables con más peso en la función discriminante son cintura (positivo), brazo (positivo), hombro (positivo), abdomen (negativo) y muslo (negativo).

```
barplot(t(m$scaling*s), las=2)
```

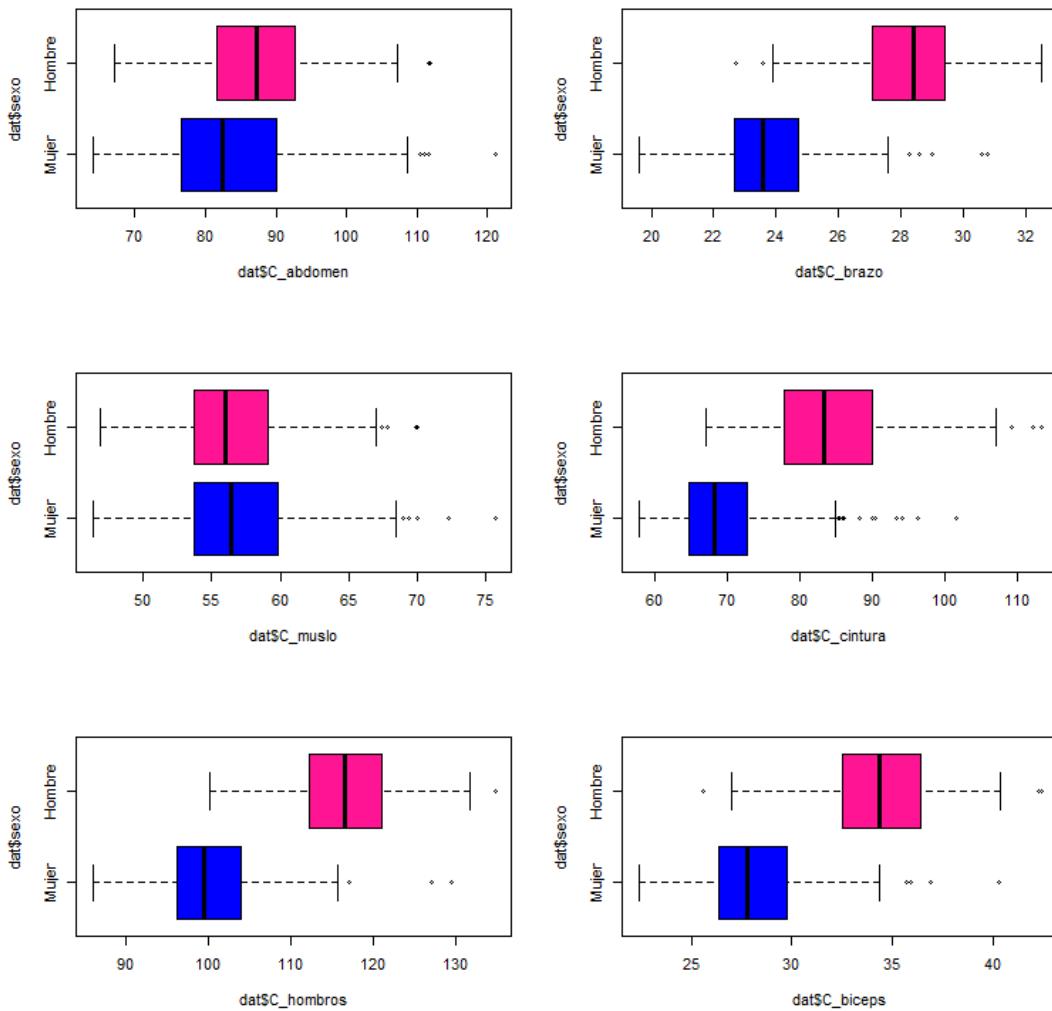


A continuación se proponen dos gráficos (no pedidos en la tarea) para ilustrar las diferencias entre mujeres y hombres en estas variables. Observa que hay variables que presentan diferencias importantes entre hombres y mujeres (muslo, por ejemplo) y que tienen mucho peso en la función discriminante que tiene en cuenta el efecto conjunto.

```
scatterplotMatrix(dat[,c("C_cintura", "C_abdomen", "C_muslo", "C_brazo", "C_hombros")], groups = dat$sexo)
```



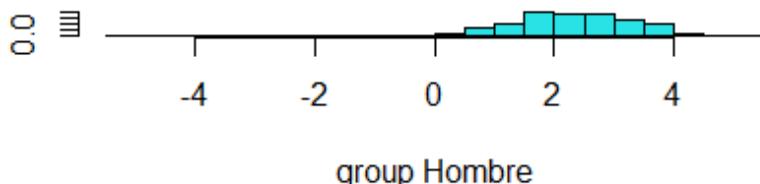
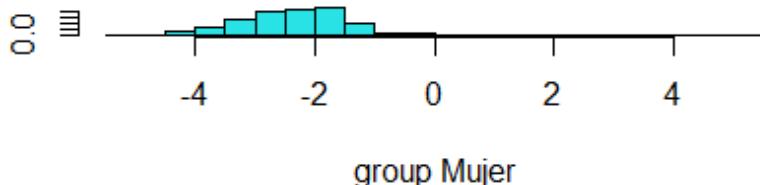
```
par(mfrow=c(3,2))
boxplot(dat$C_abdomen~dat$sexo,col=c("blue","deeppink"),horizontal = TRUE)
boxplot(dat$C_brazo~dat$sexo,col=c("blue","deeppink"),horizontal = TRUE)
boxplot(dat$C_muslo~dat$sexo,col=c("blue","deeppink"),horizontal = TRUE)
boxplot(dat$C_cintura~dat$sexo,col=c("blue","deeppink"),horizontal = TRUE)
boxplot(dat$C_hombros~dat$sexo,col=c("blue","deeppink"),horizontal = TRUE)
boxplot(dat$C_biceps~dat$sexo,col=c("blue","deeppink"),horizontal = TRUE)
```



```
par(mfrow=c(1,2))
```

6. Utiliza la instrucción `plot()` para representar las puntuaciones discriminante. Con ayuda del gráfico y de la función discriminante estandarizada interpreta las cinco variables que discriminan entre hombres y mujeres.

```
plot(m)
```



La media de la puntuación discriminante son -2.25 (Mujer) y 2.37 (Hombre). En el gráfico se aprecia que las puntuaciones discriminantes son muy diferentes para hombres y mujeres. Se pueden hacer contrastes formales que indicarían que la media de la función discriminante para mujeres es significativamente distinta que la de hombres, aunque el gráfico es suficiente para establecer esas diferencias.

Viendo los pesos de las función discriminante, a igualdad del resto de las variables, una medida alta de muslo es indicativo de mujer, una medida alta de abdomen también es de mujer, mientras que valores altos de hombros, cintura y brazo son indicadores de hombres. La función discriminante proporciona efectos marginales, es decir mide la diferencia entre hombres y mujeres a igualdad del resto de las variables. Por ejemplo, se puede comprobar con un boxplot que las distribuciones de las medidas de muslo en hombres y mujeres son similares. Ahora bien, si comparamos un hombre y una mujer con valores similares del resto de las variables (brazos, abdomen, etc), una medida alta del contorno de muslo indica que es más probable que esa persona sea una mujer.

```
pred = predict(m)
tapply(pred$x, dat$sexo, mean)

##      Mujer      Hombre
## -2.259083  2.377982
```

7. Obtén la *matriz de confusión* y explica si la función discriminante es útil para clasificar las observaciones entre hombres y mujeres.

La función *predict()* calcula las puntuaciones discriminantes y a partir de ellas predice en función de las variables explicativas si esa observación corresponde a un hombre o a una mujer.

```
pred = predict(m)
names(pred)

## [1] "class"     "posterior"  "x"
```

Los resultados de la función *predict()* son tres: **class** que nos proporciona la clase predicha para cada observación según la función discriminante, **posterior** que proporciona la probabilidad de pertenecer a la clase "mujer" u "Hombre" de cada observación (individuo) y **x** que es la puntuación discriminante.

La matriz de confusión se obtiene de la siguiente forma:

```
(t=table(Real=dat$sexo,Predicha=pred$class))

##      Predicha
## Real      Mujer Hombre
##   Mujer    257     3
##   Hombre     4    243
```

Añadiendo a la tabla el número total de filas y columnas,

```
(t1=addmargins(t))

##      Predicha
## Real      Mujer Hombre Sum
##   Mujer    257     3 260
##   Hombre     4    243 247
##   Sum      261    246 507
```

De las 260 mujeres, el método clasifica correctamente a 257 y se equivoca en 3. De los 247 hombres, 243 son clasificados correctamente y hay 4 erróneamente clasificados como mujeres. El porcentaje de fallos es de 7/507, igual a 1.4%. Por tanto acierta el 98.6%.

```
print(prop.table(t,1)*100,digits=3)

##      Predicha
## Real      Mujer Hombre
##   Mujer  98.85  1.15
##   Hombre  1.62  98.38
```

Esta manera de evaluar o el método es “sesgada”, utiliza los mismos datos para establecer primero la regla de clasificación y después para validar los resultados. Una opción para evitar este sesgo se proporciona en la siguiente pregunta.

8. Realiza la validación del método de clasificación siguiendo las siguientes instrucciones.
- Toma una muestra **mtrain** al azar del 75% de las observaciones. Las restantes observaciones las denominaremos **mtest**.
 - Estima el modelo de clasificación utilizando **mtrain**
 - Aplica el modelo **mtrain** para clasificar las observaciones reservadas **mtest** y obtén el porcentaje de observaciones mal clasificadas .
 - Repite 100 veces los pasos anteriores y guarda los errores de clasificación de cada simulación. Obtén la media de los errores.

Proporciona el programa y el error medio obtenido en tu simulación.

```
set.seed(7543) # semilla del generador de números aleatorios

n = dim(dat)[1] # número de observaciones

errores = NULL # inicialización del vector errores

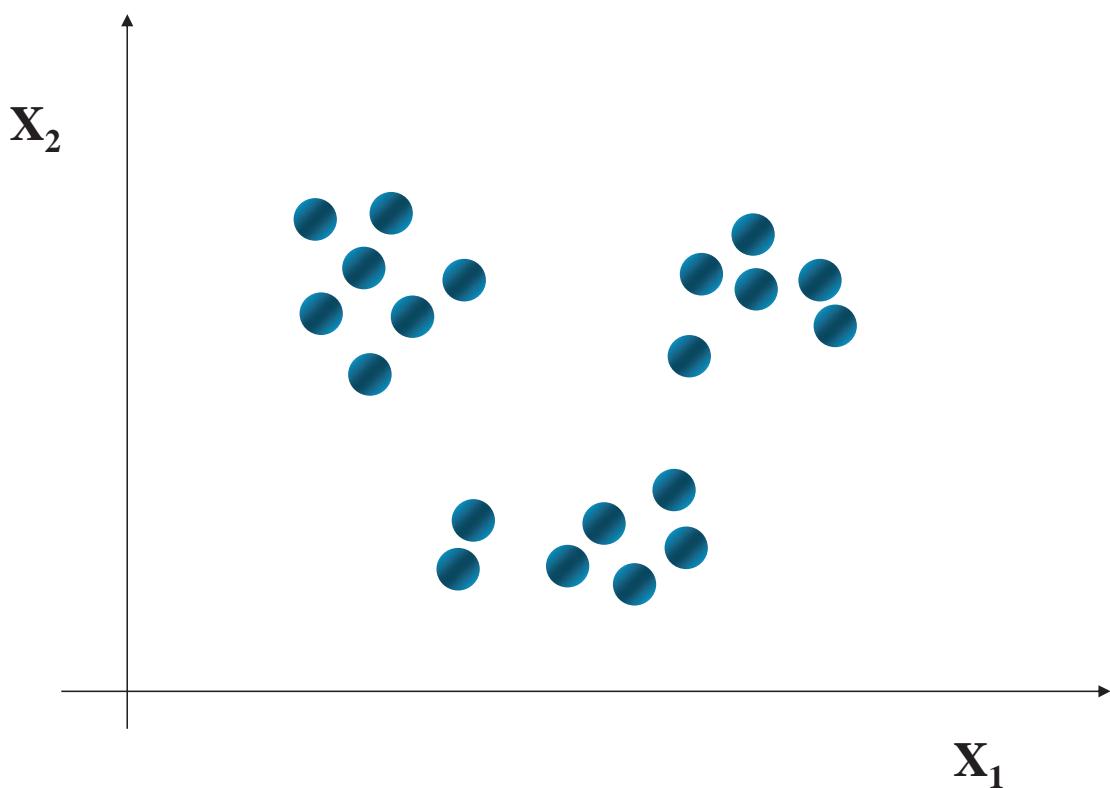
for (k in 1:100) {
  mue = sample(1:n,round(0.75*n)) # muestra para estimar el modelo en iter k
  mod = lda(sexo ~ ., data = dat[mue,c(sel,25)]) # modelo estimado en iter k
  pre = predict(mod,newdata = dat[-mue,]) # predicción con datos fuera de muestra
  tp=table(dat$sexo[-mue],pre$class) # tabla de confusión en iter k
  errores[k] = (tp[1,2]+tp[2,1])/sum(tp) # % errores de clasificación en la iter k
}
```

```
m_err = mean(errorres)  
paste0("Aciertos = ",round(100*(1-m_err),2),"% --", " Errores = ", round(100*m_err,2),"%")  
## [1] "Aciertos = 98.05% -- Errores = 1.95%"
```

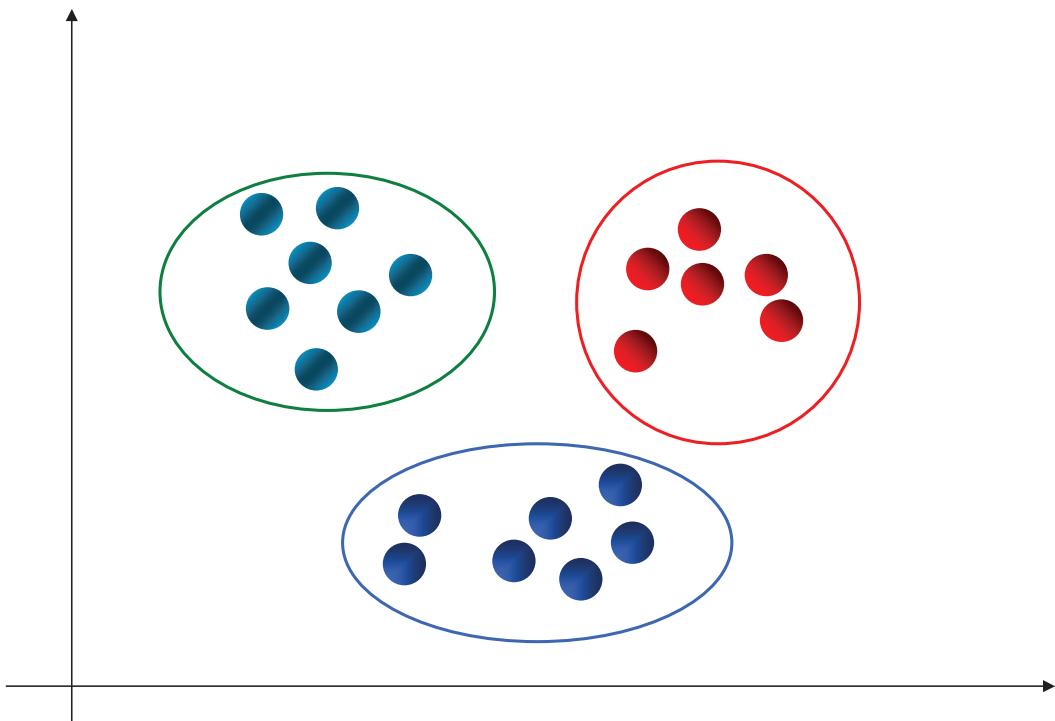
Lecc. 5 Análisis Cluster I. Cluster Jerárquico

1

¿Cómo formar grupos?

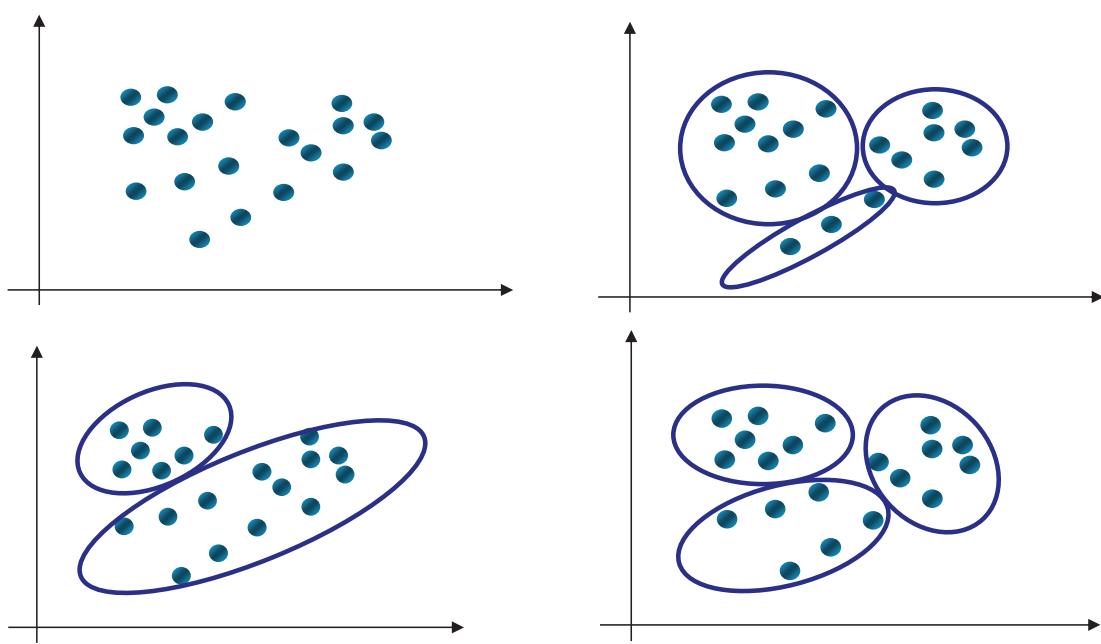


2



3

No siempre existe una solución obvia



4

Objetivo de A. Cluster

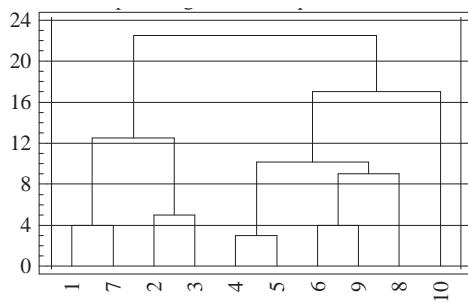
- Determinar si las observaciones se encuentran agrupadas de alguna forma.
- Formar grupos con las observaciones.

El primer objetivo es muy complicado en dimensión alta. El segundo objetivo puede tener muchas soluciones distintas.

5

Tipos de Algoritmos

- Jerárquicos



- No - Jerárquicos
K-means

6

Datos y distancias

	Álgebra	Cálculo	Estadística
1	7	6	8
2	6	6	5
3	8	7	5
4	5	7	7
5	4	6	6
6	5	4	5
7	9	6	8
8	2	6	5
9	3	4	5
10	6	4	3

$$D^2(1,2) = (7-6)^2 + (6-6)^2 + (8-5)^2 = 10$$

7

Distancia entre observaciones

Distancia eucídea :

$$D^2(x_i, x_j) = (x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \dots + (x_{ki} - x_{kj})^2$$

Distancia eucídea (datos estandarizados) :

$$D^2(x_i, x_j) = \left(\frac{x_{1i} - \bar{x}_{1j}}{\hat{s}_1}\right)^2 + \left(\frac{x_{2i} - \bar{x}_{2j}}{\hat{s}_2}\right)^2 + \dots + \left(\frac{x_{ki} - \bar{x}_{kj}}{\hat{s}_k}\right)^2$$

Distancia de Mahalanobis

$$D^2(x_i, x_j) = (x_{1i} - \bar{x}_{1j}, x_{2i} - \bar{x}_{2j}, \dots, x_{ki} - \bar{x}_{kj}) \begin{pmatrix} s_1^2 & s_{12} & \cdots & s_{1k} \\ s_{12} & s_2^2 & \cdots & s_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1k} & s_{2k} & \cdots & s_k^2 \end{pmatrix}^{-1} \begin{pmatrix} x_{1i} - \bar{x}_{1j} \\ x_{2i} - \bar{x}_{2j} \\ \vdots \\ x_{ki} - \bar{x}_{kj} \end{pmatrix}$$

8

Distancias

- **Distancia euclídea.** La raíz cuadrada de la suma de los cuadrados de las diferencias entre los valores de los elementos. Ésta es la medida por defecto para los datos de intervalo.
- **Distancia euclídea al cuadrado.** La suma de los cuadrados de las diferencias entre los valores de los elementos.
- **Correlación de Pearson.** La correlación producto-momento entre dos vectores de valores.
- **Coseno.** El coseno del ángulo entre dos vectores de valores.
- **Chebychev.** La diferencia absoluta máxima entre los valores de los elementos.
- **Bloque.** La suma de las diferencias absolutas entre los valores de los elementos. También se conoce como la distancia de Manhattan.
- **Minkowski.** La raíz p-ésima de la suma de las diferencias absolutas elevada a la potencia p-ésima entre los valores de los elementos.
- **Personalizada.** La raíz r-ésima de la suma de las diferencias absolutas elevada a la potencia p-ésima entre los valores de los elementos.

9

Matriz de Distancias²

	1	2	3	4	5	6	7	8	9	10
1										
2	10									
3	11	5								
4	6	6	13							
5	13	5	18	3						
6	17	5	18	13	6					
7	4	18	11	18	29	29				
8	34	16	37	14	5	13	58			
9	29	13	34	17	6	4	49	5		
10	30	8	17	26	17	5	38	24	13	

10

Distancias entre grupos

- Encadenamiento simple (vecino más próximo)
 - Encadenamiento completo (vecino más lejano)
 - Encadenamiento medio
 - Método del centroides
 - Método Ward (ESS)
-

11

Matriz de Distancias

	1	2	3	4	5	6	7	8	9	10
1										
2	10									
3	11	5								
4	6	6	13							
5	13	5	18	3						
6	17	5	18	13	6					
7	4	18	11	18	29	29				
8	34	16	37	14	5	13	58			
9	29	13	34	17	6	4	49	5		
10	30	8	17	26	17	5	38	24	13	

12

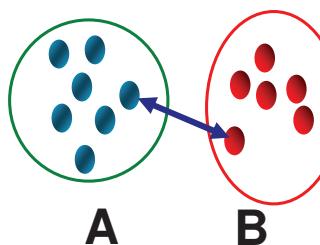
Matriz de distancias - 2

	4-5	1	2	3	6	7	8	9	10
4-5									
1	6								
2	5	10							
3	13	11	5						
6	6	17	5	18					
7	18	4	18	11	29				
8	5	34	16	37	13	58			
9	6	29	13	34	4	49	5		
10	17	30	8	17	5	38	24	13	

13

Encadenamiento simple

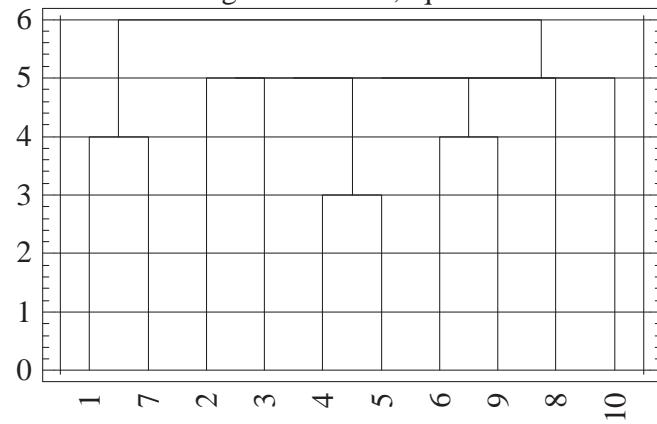
$$D(A, B) = \min\{D(i, j) : i \in A, j \in B\}$$



Distancia

Dendrograma

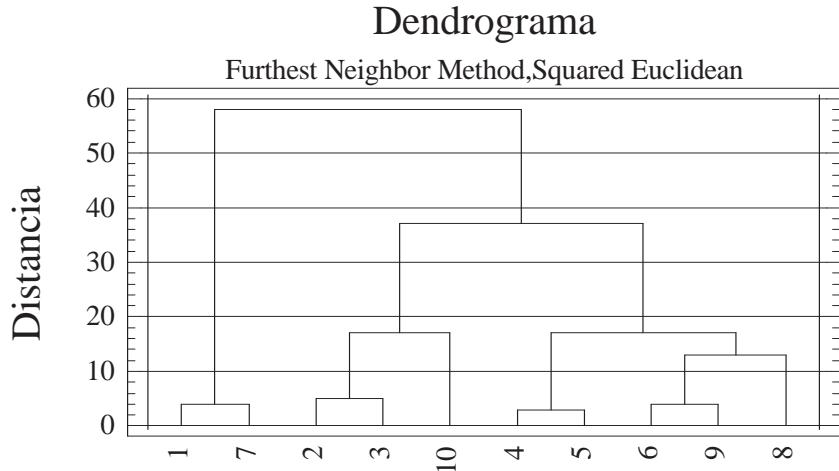
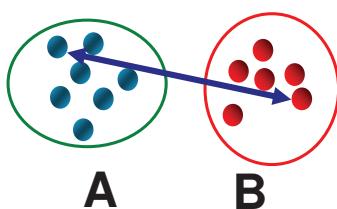
Nearest Neighbor Method, Squared Euclidean



14

Encadenamiento completo

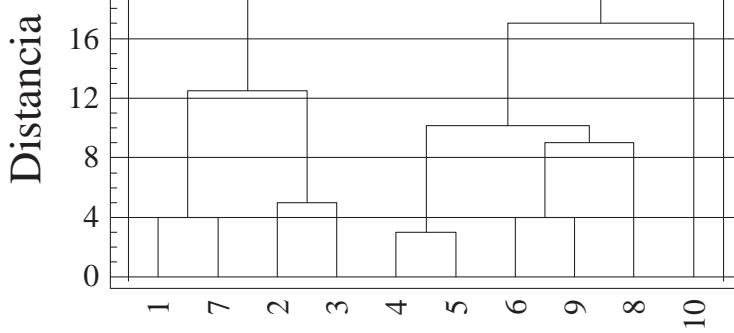
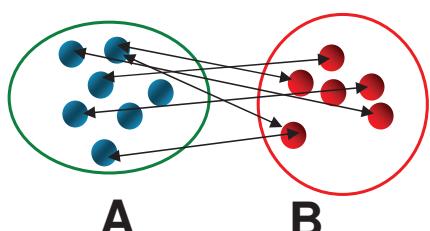
$$D(A, B) = \max\{D(i, j) : i \in A, j \in B\}$$



15

Encadenamiento medio

$$D(A, B) = \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{N_B} D(i, j)}{N_A N_B}$$



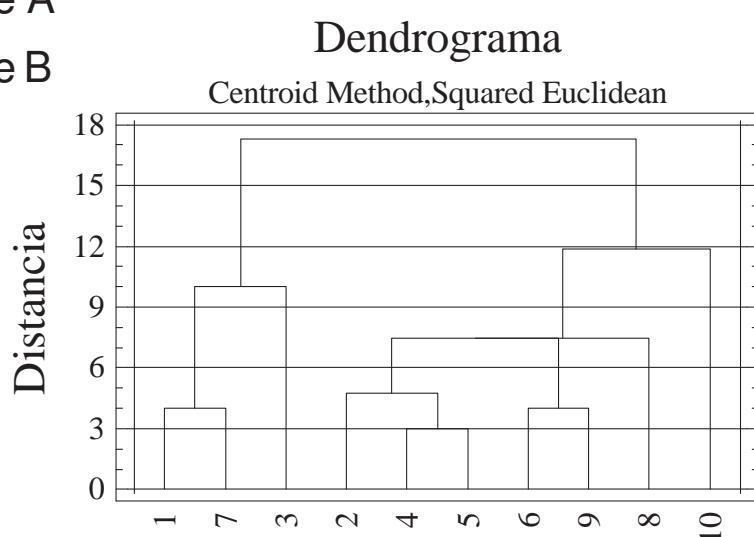
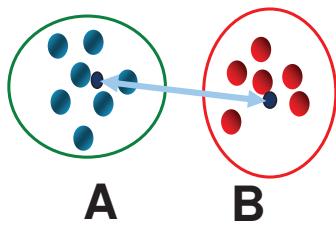
16

Centroide

$$D(A, B) = D(\bar{X}_A, \bar{X}_B)$$

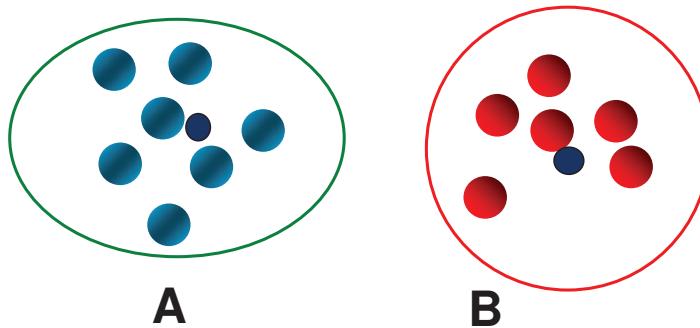
\bar{X}_A : Centroide o media de A

\bar{X}_B : Centroide o media de B



17

Ward



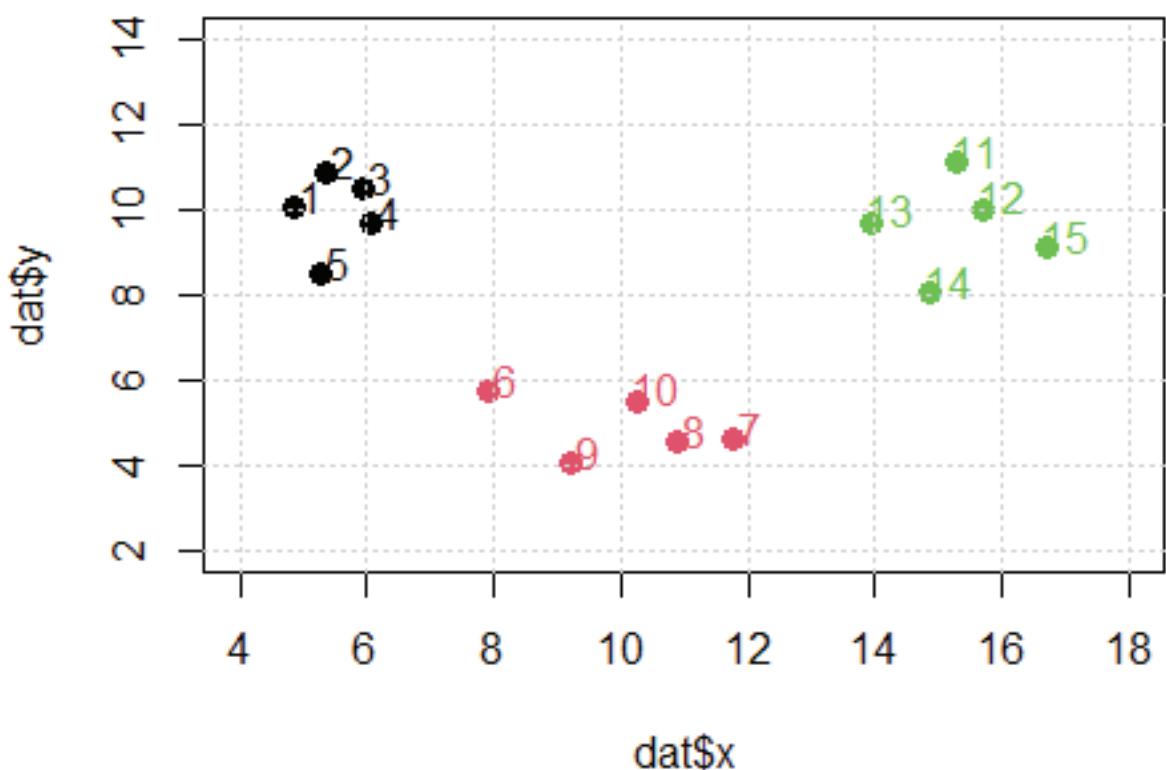
$$ESS(A) = \sum_{i=1}^k \sum_{j=1}^{N_A} (x_{ij} - \bar{x}_i)^2, \quad (x_{1j}, x_{2j}, \dots, x_{kj}) \in A$$

$$D(A, B) = ESS(A \cup B) - ESS(A) - ESS(B)$$

18

CON R

Ejemplo



```

d = dist(dat[,1:2],method = "euclidean",
          diag = T,upper = T)
print(d,digits = 1)

```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
## 1	0.0	1.0	1.2	1.3	1.6	5.3	8.8	8.1	7.4	7.1	10.5	10.9	9.1	10.2	11.9
## 2	1.0	0.0	0.7	1.4	2.4	5.7	8.9	8.4	7.8	7.3	9.9	10.4	8.6	9.9	11.5
## 3	1.2	0.7	0.0	0.8	2.1	5.1	8.2	7.7	7.2	6.6	9.3	9.8	8.0	9.2	10.8
## 4	1.3	1.4	0.8	0.0	1.4	4.4	7.6	7.0	6.4	5.9	9.3	9.6	7.8	8.9	10.6
## 5	1.6	2.4	2.1	1.4	0.0	3.8	7.6	6.9	5.9	5.8	10.3	10.6	8.8	9.6	11.4
## 6	5.3	5.7	5.1	4.4	3.8	0.0	4.0	3.2	2.1	2.4	9.1	8.9	7.2	7.4	9.4
## 7	8.8	8.9	8.2	7.6	7.6	4.0	0.0	0.9	2.6	1.7	7.4	6.7	5.5	4.6	6.7
## 8	8.1	8.4	7.7	7.0	6.9	3.2	0.9	0.0	1.8	1.1	7.9	7.3	6.0	5.3	7.4
## 9	7.4	7.8	7.2	6.4	5.9	2.1	2.6	1.8	0.0	1.8	9.3	8.8	7.4	6.9	9.0
## 10	7.1	7.3	6.6	5.9	5.8	2.4	1.7	1.1	1.8	0.0	7.5	7.1	5.6	5.3	7.4
## 11	10.5	9.9	9.3	9.3	10.3	9.1	7.4	7.9	9.3	7.5	0.0	1.2	1.9	3.1	2.4
## 12	10.9	10.4	9.8	9.6	10.6	8.9	6.7	7.3	8.8	7.1	1.2	0.0	1.8	2.2	1.3
## 13	9.1	8.6	8.0	7.8	8.8	7.2	5.5	6.0	7.4	5.6	1.9	1.8	0.0	1.9	2.8
## 14	10.2	9.9	9.2	8.9	9.6	7.4	4.6	5.3	6.9	5.3	3.1	2.2	1.9	0.0	2.1
## 15	11.9	11.5	10.8	10.6	11.4	9.4	6.7	7.4	9.0	7.4	2.4	1.3	2.8	2.1	0.0

```

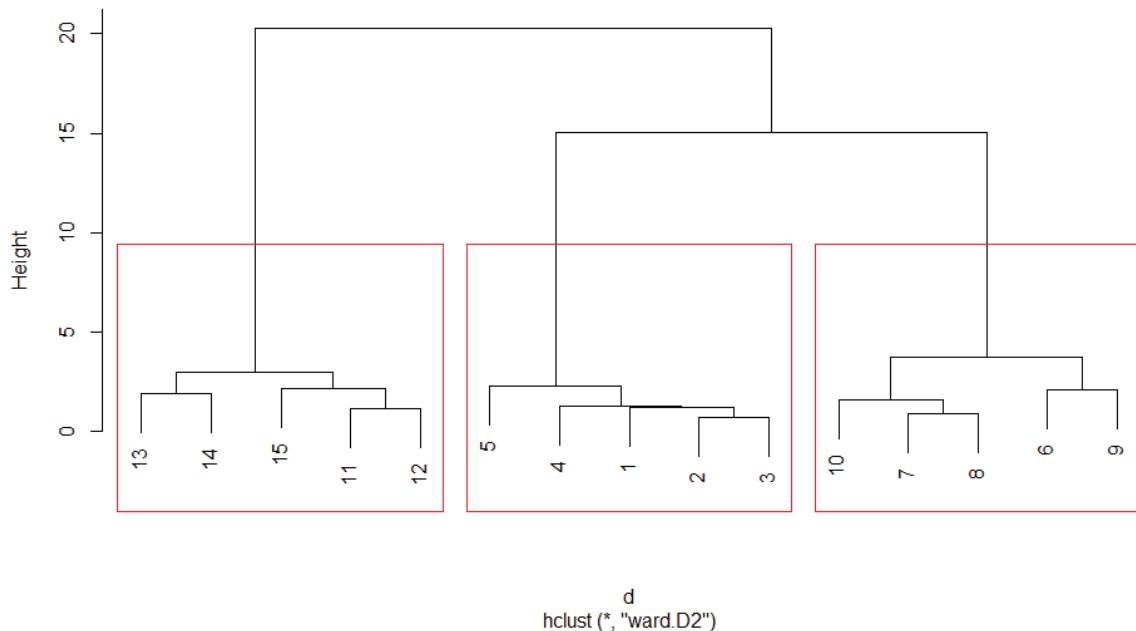
d = dist(dat[,1:2],method = "euclidean")
print(d,digits = 1)

```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
## 2	1.0													
## 3	1.2	0.7												
## 4	1.3	1.4	0.8											
## 5	1.6	2.4	2.1	1.4										
## 6	5.3	5.7	5.1	4.4	3.8									
## 7	8.8	8.9	8.2	7.6	7.6	4.0								
## 8	8.1	8.4	7.7	7.0	6.9	3.2	0.9							
## 9	7.4	7.8	7.2	6.4	5.9	2.1	2.6	1.8						
## 10	7.1	7.3	6.6	5.9	5.8	2.4	1.7	1.1	1.8					
## 11	10.5	9.9	9.3	9.3	10.3	9.1	7.4	7.9	9.3	7.5				
## 12	10.9	10.4	9.8	9.6	10.6	8.9	6.7	7.3	8.8	7.1	1.2			
## 13	9.1	8.6	8.0	7.8	8.8	7.2	5.5	6.0	7.4	5.6	1.9	1.8		
## 14	10.2	9.9	9.2	8.9	9.6	7.4	4.6	5.3	6.9	5.3	3.1	2.2	1.9	
## 15	11.9	11.5	10.8	10.6	11.4	9.4	6.7	7.4	9.0	7.4	2.4	1.3	2.8	2.1

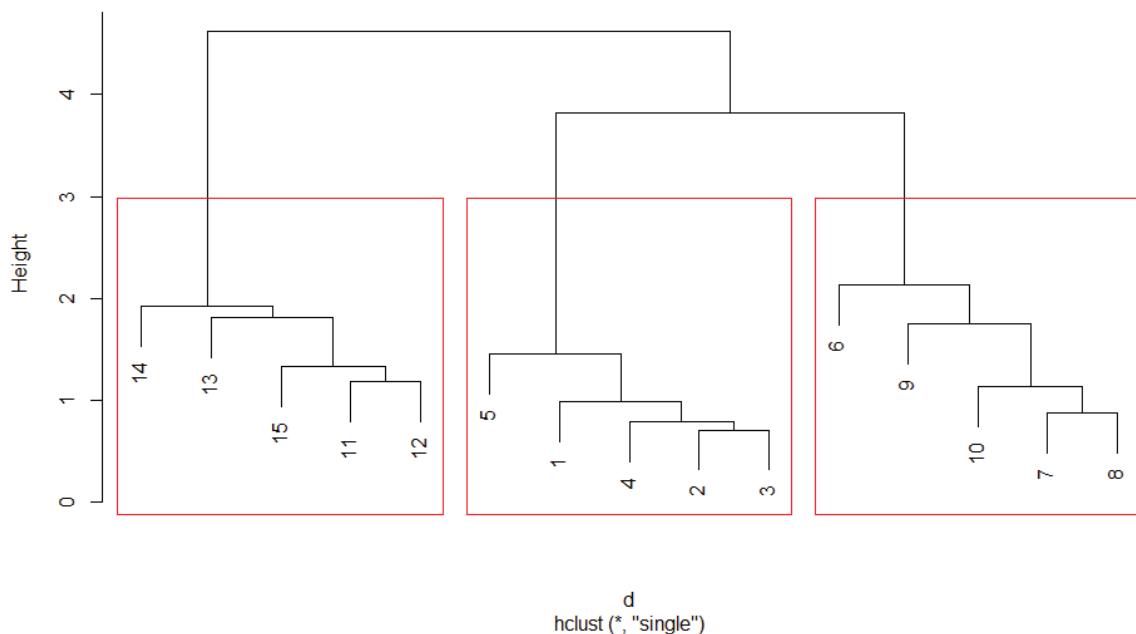
```
fit <- hclust(d, method="ward.D2")
plot(fit) # display dendrogram
groups <- cutree(fit, k=3) # cut tree into 3
clusters
# draw dendrogram with red borders around the 3
clusters
rect.hclust(fit, k=3, border="red")
```

Cluster Dendrogram

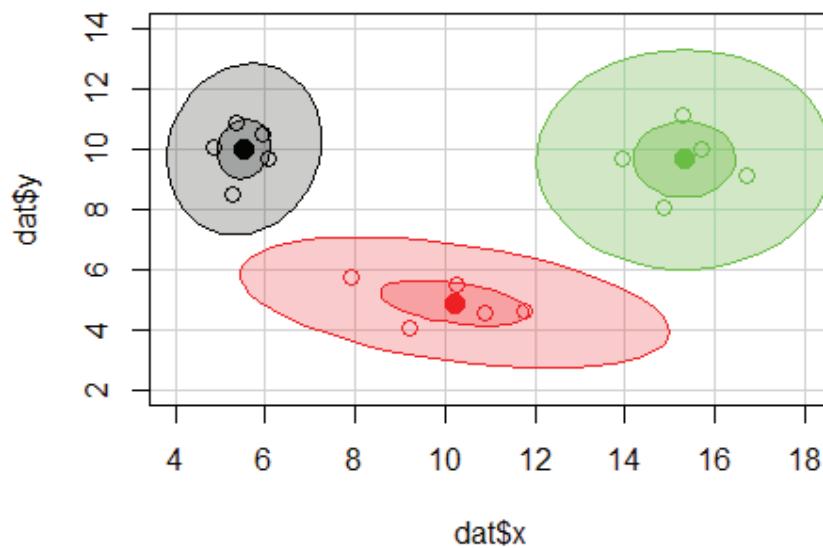


```
fit <- hclust(d, method="single")
plot(fit) # display dendrogram
groups <- cutree(fit, k=3) # cut tree into 3
# clusters
# draw dendrogram with red borders around the
# 3clusters
rect.hclust(fit, k=3, border="red")
```

Cluster Dendrogram



```
#library(car)
scatterplot(dat$x,dat$y,groups = dat$g,
            regLine = F,legend = F, ellipse =
T,col=c("black","red","green"),
            smooth = F,
xlim=c(4,18),ylim=c(2,14),pch=c(21,21,21),cex=1.2)
```

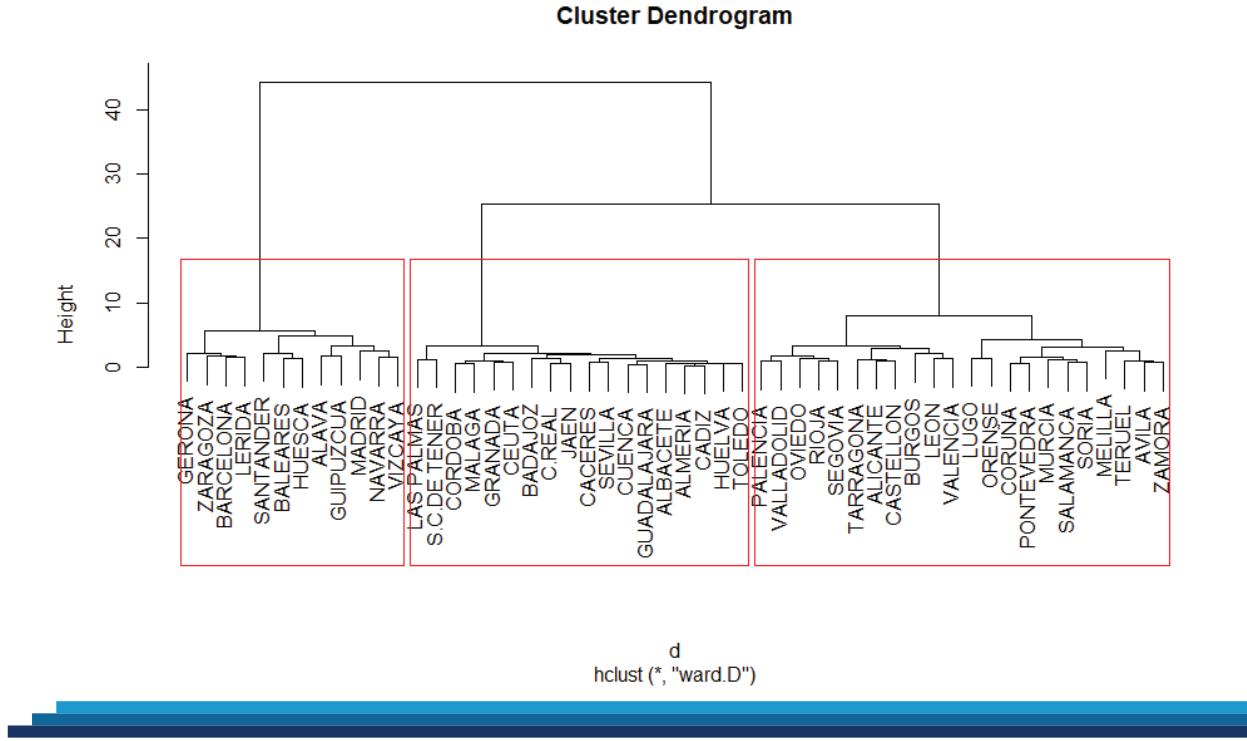


GASTOS PROVINCIALES

```
gastos = read.csv("gastos.csv", header=TRUE)
dat = gastos[,3:8]
row.names(dat)=gastos[,1]

dat = scale(dat)
d = dist(dat)

fit = hclust(d, method = "ward.D")
plot(fit)
groups <- cutree(fit, k=3)
rect.hclust(fit, k=3, border="red")
```



```

mu1 = sapply(dat0[groups==1, ], mean)
mu2 = sapply(dat0[groups==2, ], mean)
mu3 = sapply(dat0[groups==3, ], mean)

cbind(alto=mu1, medio=mu3, bajo=mu2)
##           alto     medio     bajo
## ALIMENT   1968.8333 1829.3636 1301.3889
## VESTIDO    743.0000  582.0455  422.2222
## VIVIENDA  5442.7500 3203.8636 2351.6667
## SALUD      645.9167  361.2727  266.5556
## TRANSP    1862.2500 1191.5000  887.7222
## CULTURA   3454.4167 1964.9091 1403.2222

```

```

alto=rownames(dat)[groups==1]
bajo=rownames(dat)[groups==2]
medio=rownames(dat)[groups==3]

sort(alto)
## [1] "ALAVA"      "BALEARES"    "BARCELONA"   "GERONA"      "GUIPUZCUA"   "HUESCA"
## [7] "LERIDA"      "MADRID"      "NAVARRA"     "SANTANDER"   "VIZCAYA"     "ZARAGOZA"

sort(medio)
## [1] "ALICANTE"   "AVILA"       "BURGOS"      "CASTELLON"   "CORUÑA"
## [6] "LEON"        "LUGO"        "MELILLA"     "MURCIA"      "ORENSE"
## [11] "OVIEDO"      "PALENCIA"   "PONTEVEDRA"  "RIOJA"       "SALAMANCA"
## [16] "SEGOVIA"     "SORIA"       "TARRAGONA"   "TERUEL"      "VALENCIA"
## [21] "VALLADOLID" "ZAMORA"

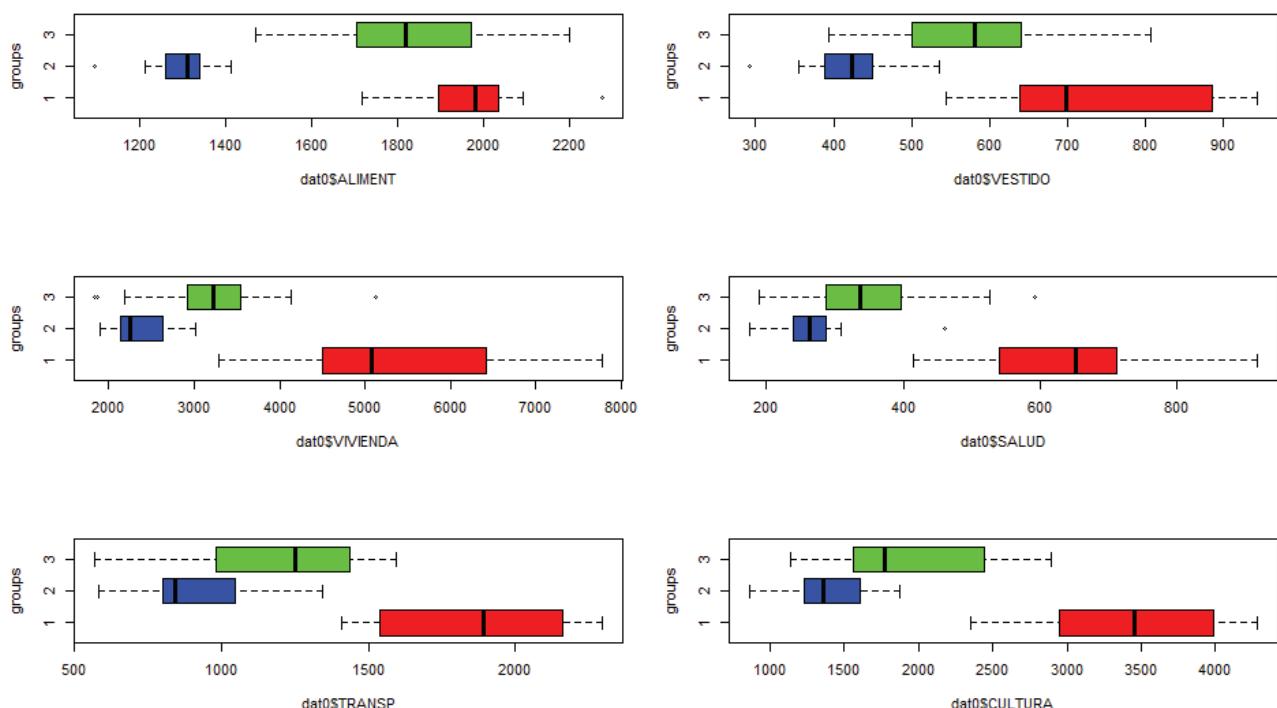
sort(bajo)
## [1] "ALBACETE"    "ALMERIA"     "BADAJOZ"     "C.REAL"      "CACERES"
## [6] "CADIZ"        "CEUTA"       "CORDOBA"     "CUENCA"      "GRANADA"
## [11] "GUADALAJARA" "HUELVA"     "JAEN"        "LAS PALMAS"  "MALAGA"
## [16] "S.C.DE TENER" "SEVILLA"    "TOLEDO"

```

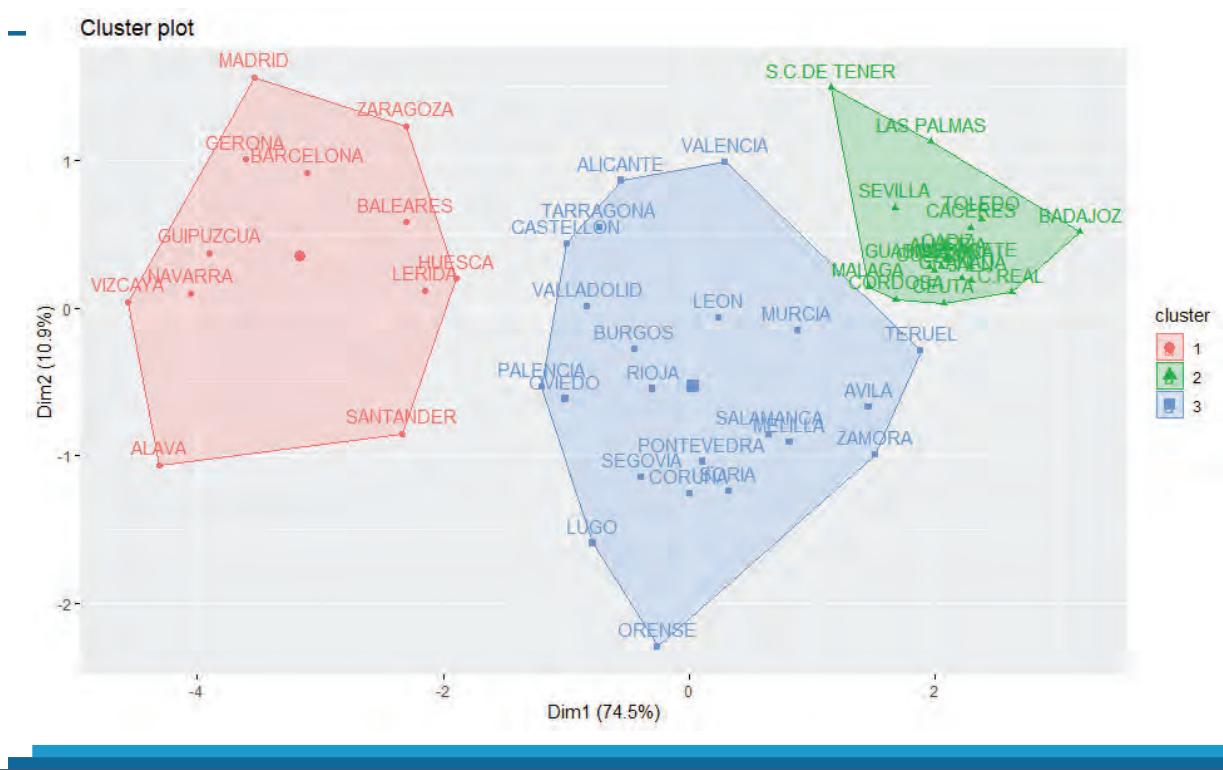
```

par(mfrow=c(3,2))
boxplot(dat0$ALIMENT~groups,
         col=c("red","blue","green"),horizontal = T)
boxplot(dat0$VESTIDO~groups,
         col=c("red","blue","green"),horizontal = T)
boxplot(dat0$VIVIENDA~groups,
         col=c("red","blue","green"),horizontal = T)
boxplot(dat0$SALUD~groups,
         col=c("red","blue","green"),horizontal = T)
boxplot(dat0$TRANSP~groups,
         col=c("red","blue","green"),horizontal = T)
boxplot(dat0$CULTURA~groups,
         col=c("red","blue","green"),horizontal = T)

```



```
fviz_cluster(list(data=dat, cluster=groups))
```

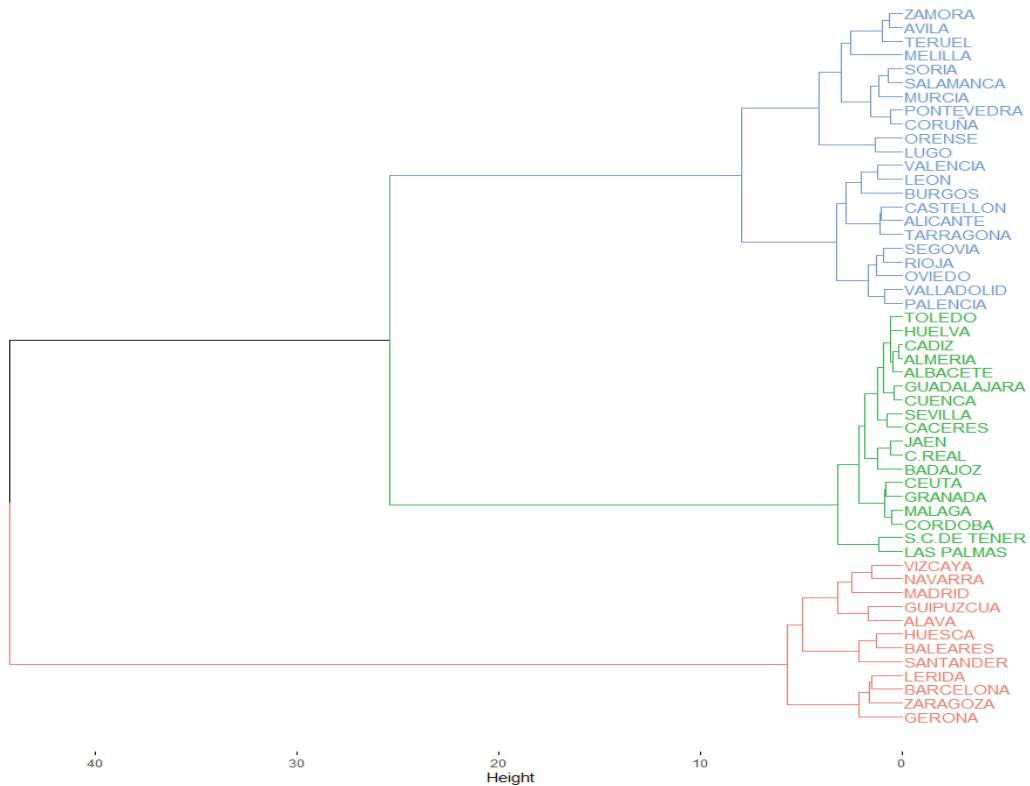


```

fit=eclust(dat, "hclust", k=3, graph=T
RUE, hc_method = "ward.D")
fviz_dend(fit, k=3, horiz=TRUE)

```

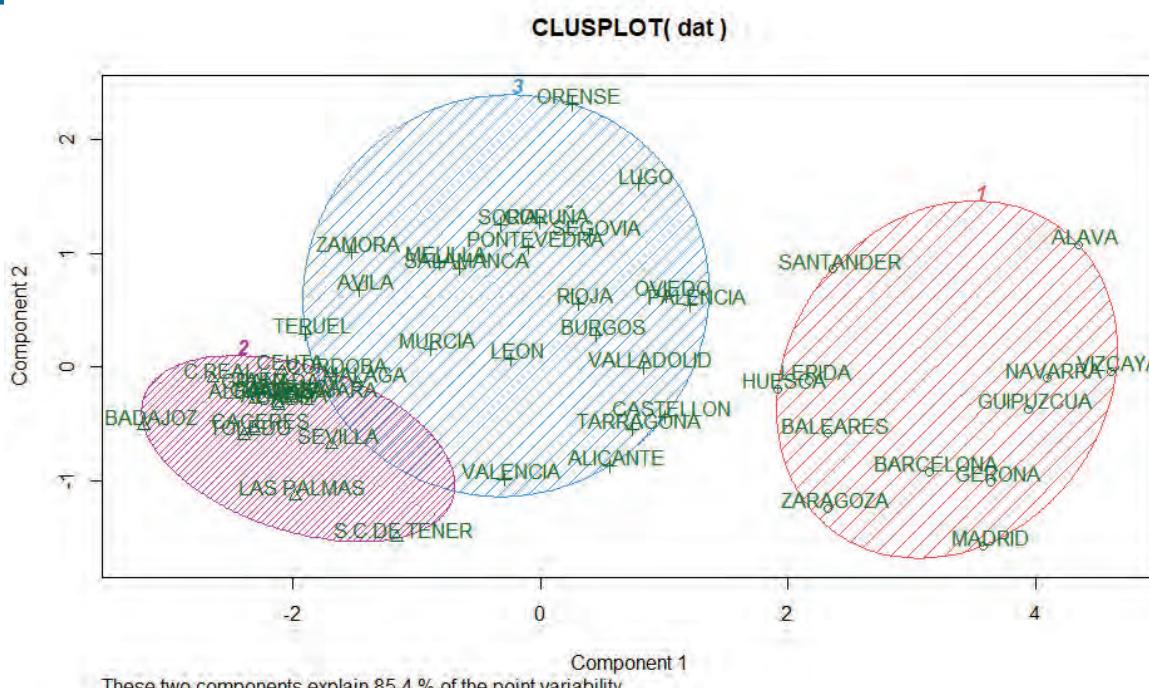
Cluster Dendrogram



```

clusplot(dat, groups, color=TRUE,
shade=TRUE,
labels=2, lines=0)

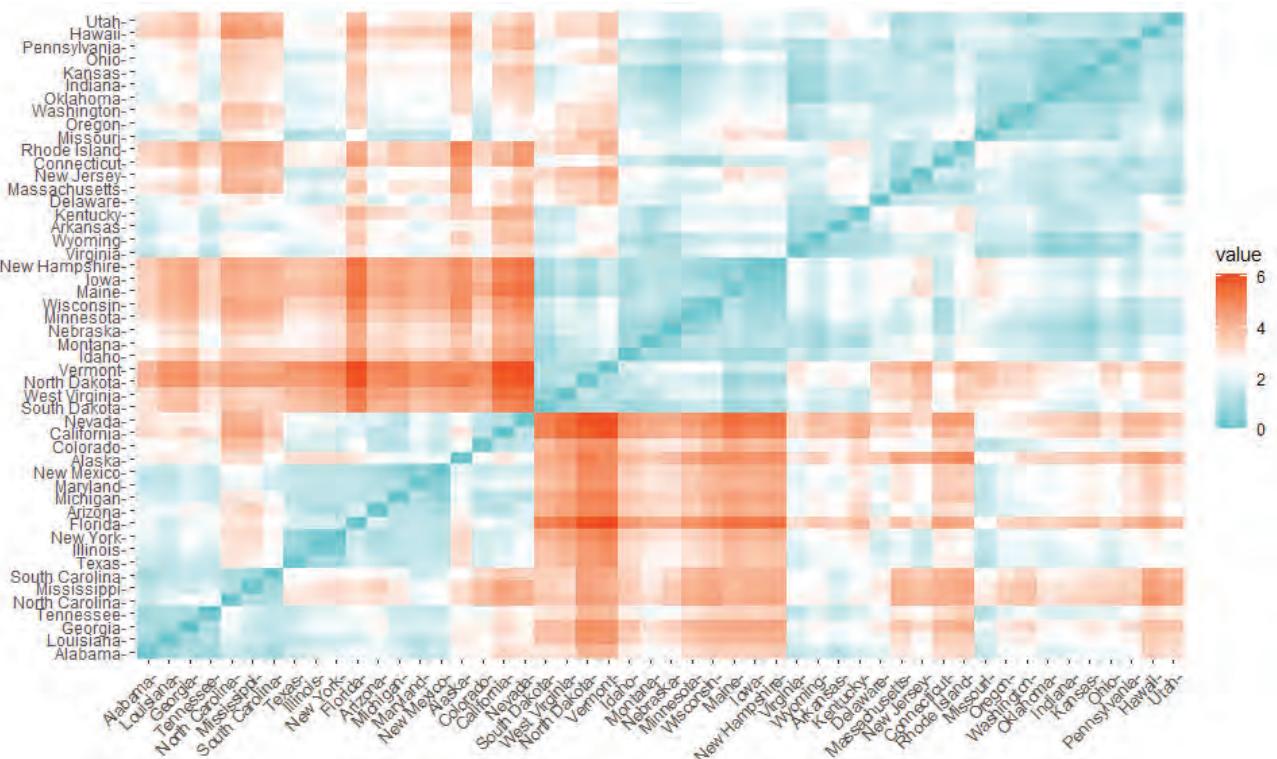
```



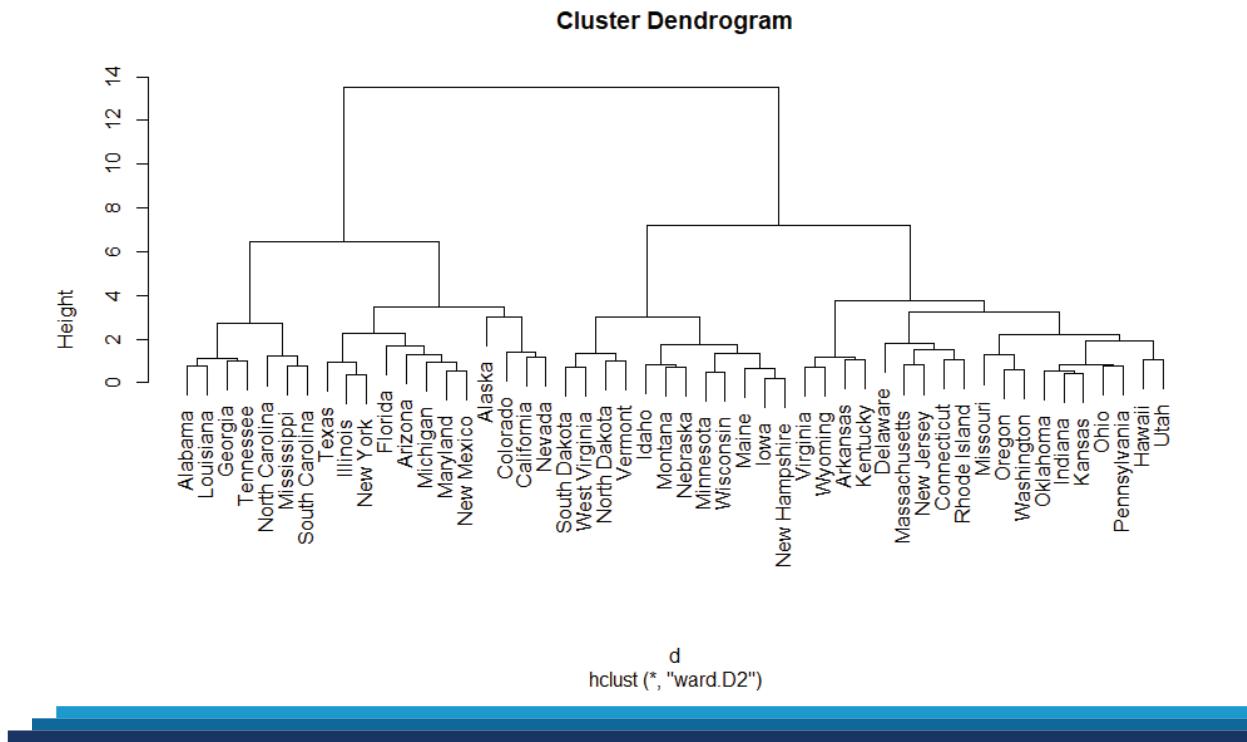
USA Arrests

data (USArrests)

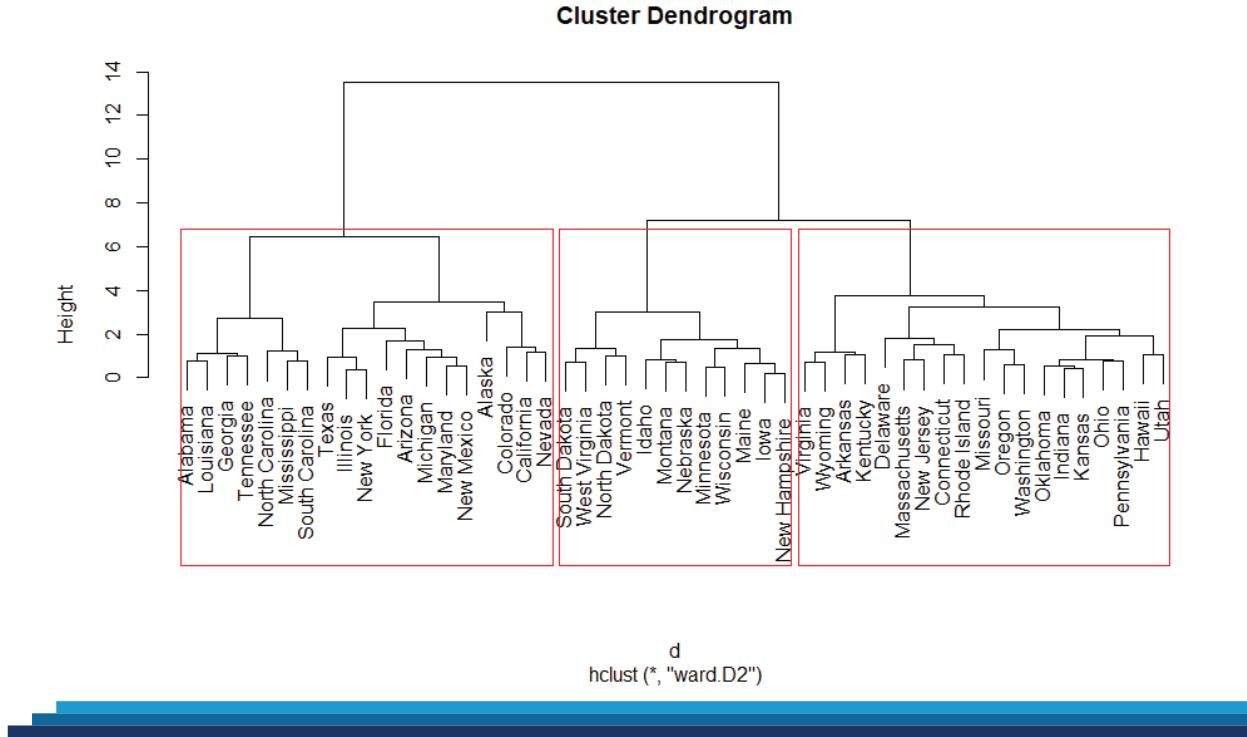
```
data(USArrests)
d <- dist(scale(USArrests),method = "euclidean")
fviz_dist(d,gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



```
fit <- hclust(d, method="ward.D2")
plot(fit) # display dendrogram
```



```
groups <- cutree(fit, k=3) # cut tree
into 5 clusters
# draw dendrogram with red borders around
the 5 clusters
plot(fit) # display dendrogram
rect.hclust(fit, k=3, border="red")
```



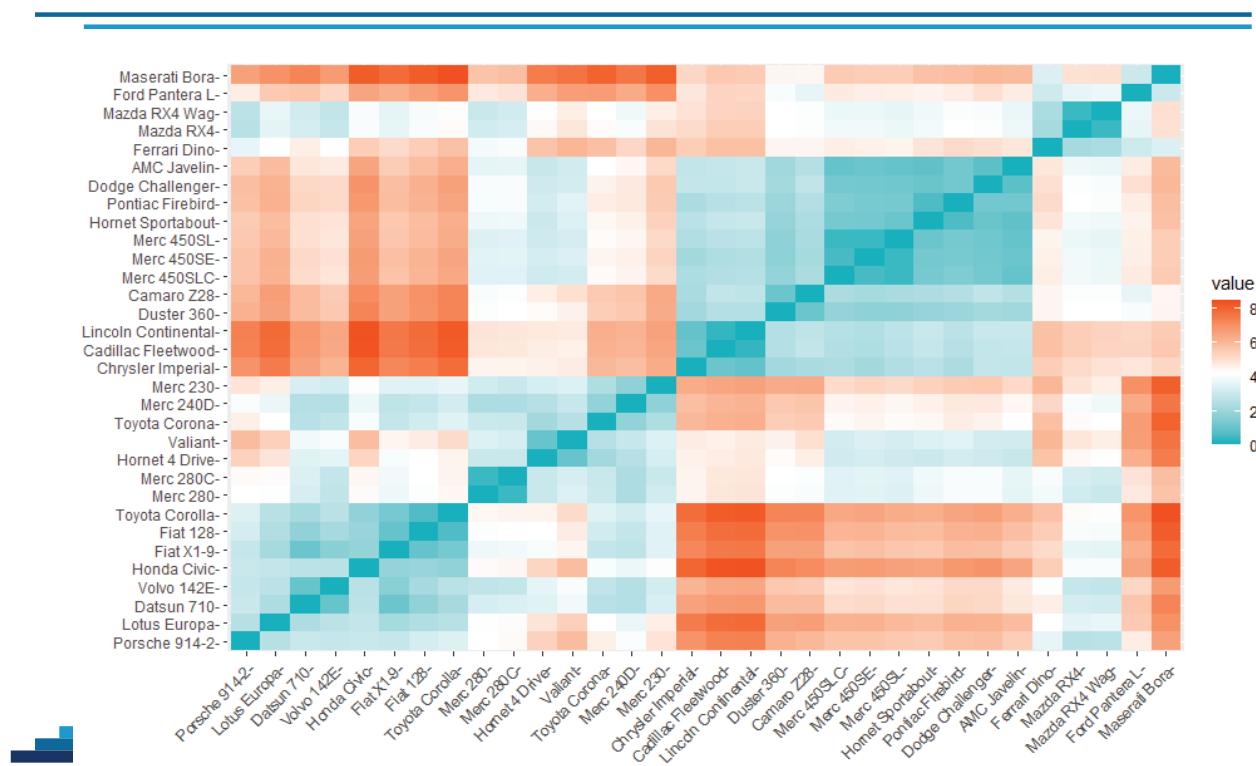
mtcars

```

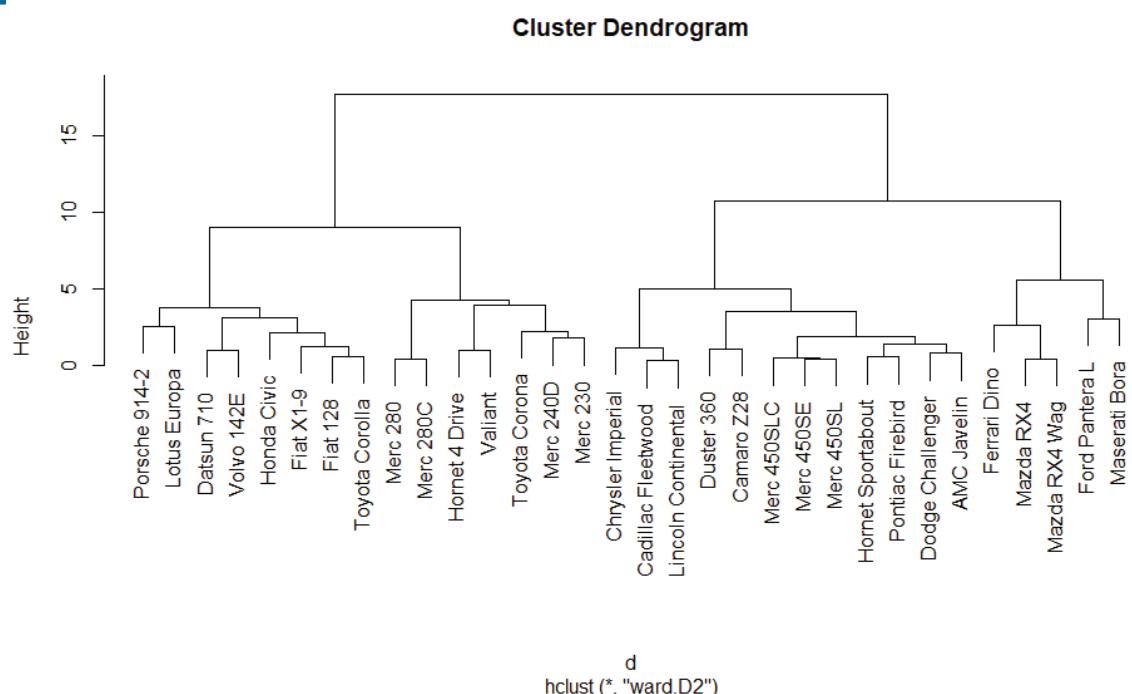
data("mtcars")
head(mtcars)
##          mpg cyl disp  hp drat    wt  qsec vs am
##          gear carb
## Mazda RX4     21.0   6 160 110 3.90 2.620 16.46  0  1
##                  4   4
## Mazda RX4 Wag 21.0   6 160 110 3.90 2.875 17.02  0  1
##                  4   4
## Datsun 710    22.8   4 108  93 3.85 2.320 18.61  1  1
##                  4   1
## Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0
##                  3   1
## Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02  0  0
##                  3   2
## Valiant       18.1   6 225 105 2.76 3.460 20.22  1  0
##                  3   1

```

```
d <- dist(scale(mtcars), method = "euclidean")
fviz_dist(d, gradient = list(low = "#00AFBB",
                             mid = "white", high = "#FC4E07"))
```



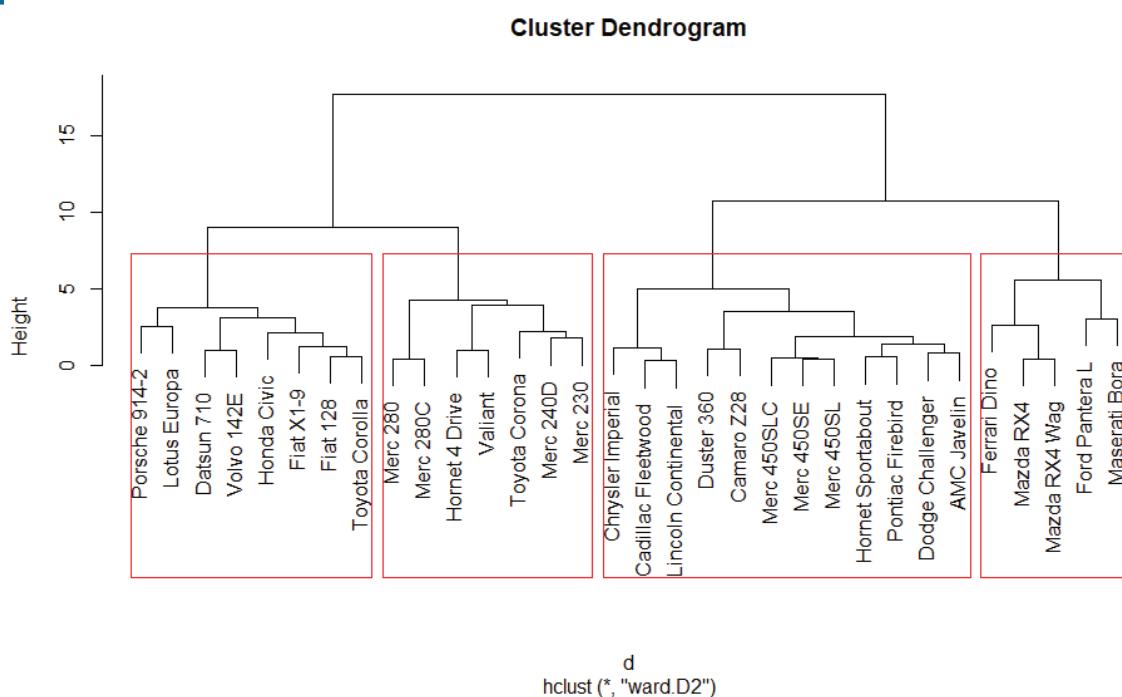
```
fit <- hclust(d,
method="ward.D2")
plot(fit) # display dendrogram
```



```

groups <- cutree(fit, k=4) # cut tree
into 5 clusters
# draw dendrogram with red borders around
the 5 clusters
plot(fit) # display dendrogram
rect.hclust(fit, k=4, border="red")

```



ANÁLISIS DE DATOS

Lecc. 5 Análisis Cluster II. K-means

1

Ejemplo: gastos.csv

▲	PROVINC	◆	ALIMENT	◆	VESTIDO	◆	VIVIENDA	◆	SALUD	◆	TRANSP	◆	CULTURA	◆
1	ALAVA		2275		929		7781		638		1408		3524	
2	ALBACETE		1253		449		2245		266		800		1182	
3	ALICANTE		1512		596		3104		527		1401		2785	
4	ALMERIA		1338		423		2218		287		818		1378	
5	AVILA		1706		500		2180		256		883		1445	
6	BADAJOZ		1098		366		1896		176		757		868	
7	BALEARES		1777		629		6211		438		2079		2802	
8	BARCELONA		1977		649		4777		724		1909		3778	
9	BURGOS		1878		496		5132		266		1225		2499	
10	CACERES		1246		399		2157		240		1075		1104	
11	CADIZ		1325		429		2191		309		836		1278	
12	CASTELLON		1676		620		3974		592		1412		2310	
13	C.REAL		1301		388		2663		241		582		967	

48	VIZCAYA	2030	943	•	6406	632	2108	4130
49	ZAMORA	1821	467	•	2628	189	833	1360
50	ZARAGOZA	1879	545	•	4477	865	1410	3382
51	CEUTA	1332	498	•	2018	275	657	1554
52	MELILLA	1899	563	•	1857	515	569	1560

2

Procedimiento K-Means

Sea $C = \{1, 2, 3, \dots, n\}$ el conjunto de las n observaciones:

Se define una partición a $\boxed{C_1, C_2, \dots, C_K}$

1. Cada C_k es un subconjunto de C
2. La unión de todos es C
3. La intersección de dos cualquiera de ellos es el vacío

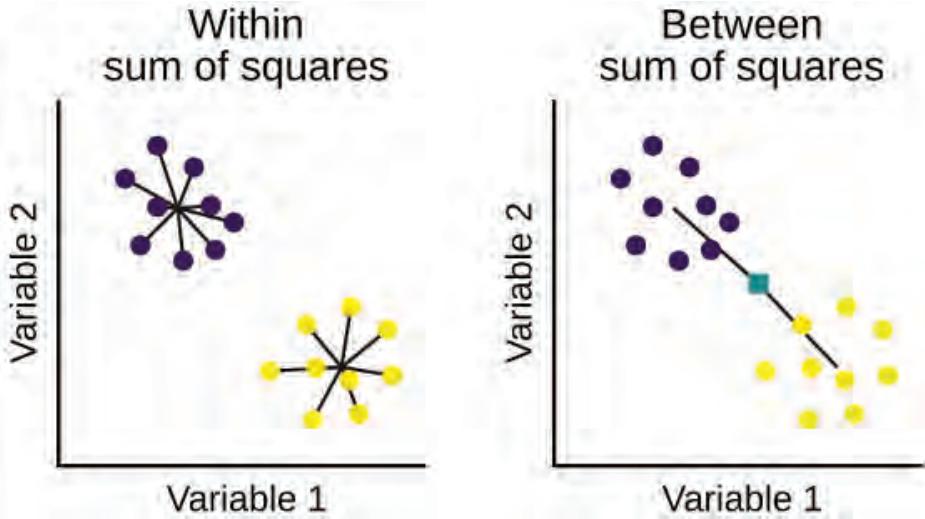
Within-cluster variation

$$W(C_k) = \frac{1}{n_k} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{x}_j)^2$$

n_k número de observaciones en C_k

Objetivo

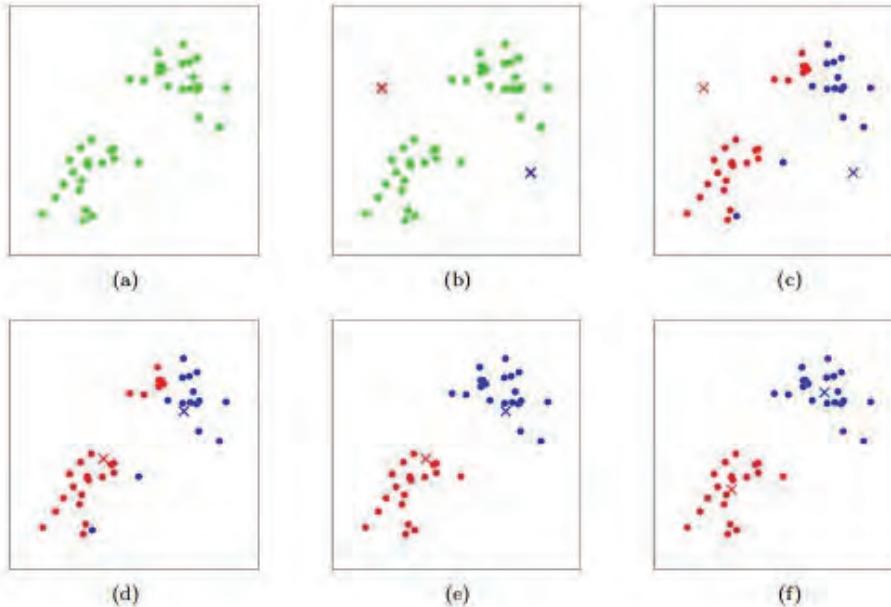
$$\text{minimizar}_{C_1, C_2, \dots, C_K} \sum_{k=1}^K W(C_k)$$



Algoritmo K-means

- (1) Se fija el número de grupos K
- (2) Se eligen K observaciones al azar que se denominan centroides de los clusters iniciales.
- (3) Se toma por orden las observaciones y se asignan al grupo cuyo centroide esté más próximo.
- (4) Se recalcula los centroides de los clusters.
- (5) Si los centroides han cambiado, se vuelve al paso (3).

Múltiples versiones (centros generados al azar)



Importante

K-means encuentra un óptimo local, el resultado final dependerá de la elección inicial de los centroides.

Es importante repetir el algoritmo varias veces con puntos de inicio diferentes y elegir la solución que proporciona la menor variación within-clusters.

K-MEANS



Ejemplo: gastos.csv

```
gastos = read.csv("gastos.csv", header=TRUE)
dat0 = gastos[,3:8]
row.names(dat0) = gastos[,1]
k3 = kmeans(dat0, centers = 3, nstart = 25)
```

	PROVINC	ALIMENT	VESTIDO	VIVIENDA	SALUD	TRANSP	CULTURA
1	ALAVA	2275	929	7781	638	1408	3524
2	ALBACETE	1253	449	2245	266	800	1182
3	ALICANTE	1512	598	3104	527	1401	2785
4	ALMERIA	1338	423	2218	287	818	1378
5	AVILA	1706	500	2180	256	883	1445
6	BADAJOZ	1098	366	1896	176	757	868
7	BALEARES	1777	629	6211	438	2079	2802
8	BARCELONA	1977	649	4777	724	1909	3778
9	BURGOS	1878	498	5132	266	1225	2499
10	CACERES	1246	399	2157	240	1075	1104
11	CADIZ	1325	429	2191	309	836	1278
12	CASTELLON	1676	620	3974	592	1412	2310
13	CREAL	1301	388	2663	241	582	967
48	VIZCAYA	2030	943	6406	632	2108	4130
49	ZAMORA	1821	467	2628	189	633	1360
50	ZARAGOZA	1879	545	4477	865	1410	3382
51	CEUTA	1332	498	2018	275	657	1554
52	MELILLA	1899	563	1857	515	569	1560

```
names(k2)
## [1] "cluster"      "centers"       "totss"        "withinss"      "tot.withinss"
## [6] "betweenss"    "size"          "iter"         "ifault"
```

- **cluster**: Un vector con enteros de 1 hasta K, indicando el cluster al que pertenece cada observación
- **centers**: Los centroides de cada cluster
- **totss**: total sum of squares
- **withinss**: within-cluster variación para cada cluster
- **tot.withinss**: suma de withinss
- **betweenss** : totss - tot.withinss
- **size**: número de observaciones en cada cluster
- **iter** : número de iteraciones
- **ifault**: problema del algoritmo (para expertos).

```
k3
## K-means clustering with 3 clusters of sizes 15, 25, 12
##
## Cluster means:
##           ALIMENT   VESTIDO  VIVIENDA     SALUD    TRANSP   CULTURA
## 1 1844.667 614.6000 3426.267 410.6667 1368.200 2220.333
## 2 1446.520 454.8400 2382.920 280.7600 878.000 1436.920
## 3 1955.333 727.5833 5596.750 609.8333 1838.833 3392.583
##
## Clustering vector:
##           ALAVA    ALBACETE    ALICANTE    ALMERIA    AVILA    BADAJOZ
## 1            3          2          1          2          2          2
## 2           BALEARES    BARCELONA    BURGOS    CACERES    CADIZ    CASTELLON
## 3            3          3          3          2          2          2
## 3           C.REAL     CORDOBA    CORUÑA    CUENCA    GERONA    GRANADA
## 2            2          2          1          2          3          2
## 2          GUADALAJARA    GUIPUZCUA    HUELVA    HUESCA    JAEN     LEON
## 2            2          3          2          3          2          1
## 1           LERIDA      RIOJA      LUGO      MADRID    MALAGA    MURCIA
## 1            1          1          1          3          2          2
## 2           NAVARRA     ORENSE     OVIEDO    PALENCIA    LAS PALMAS    PONTEVEDRA
## 3            3          2          1          1          2          1
## 2          SALAMANCA S.C.DE TENER    SANTANDER    SEGOVIA    SEVILLA    SORIA
## 2            2          2          3          1          2          1
## 2          TARRAGONA     TERUEL     TOLEDO    VALENCIA    VALLADOLID    VIZCAYA
## 1            1          2          2          1          1          3
## 2           ZAMORA      ZARAGOZA    CEUTA     MELILLA
## 2            2          3          2          2
##
## Within cluster sum of squares by cluster:
## [1] 7911007 7540866 20596084
## (between_SS / total_SS =  77.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"       "totss"        "withinss"      "tot.withinss"
## [6] "betweenss"    "size"          "iter"         "ifault"
```

```

k3
## K-means clustering with 3 clusters of sizes
15, 25, 12
##
## Cluster means:
##      ALIMENT    VESTIDO   VIVIENDA     SALUD     TRANSP    CULTURA
## 1 1844.667 614.6000 3426.267 410.6667 1368.200 2220.333
## 2 1446.520 454.8400 2382.920 280.7600 878.000 1436.920
## 3 1955.333 727.5833 5596.750 609.8333 1838.833 3392.583
##
```

Resulta principal: k3\$cluster

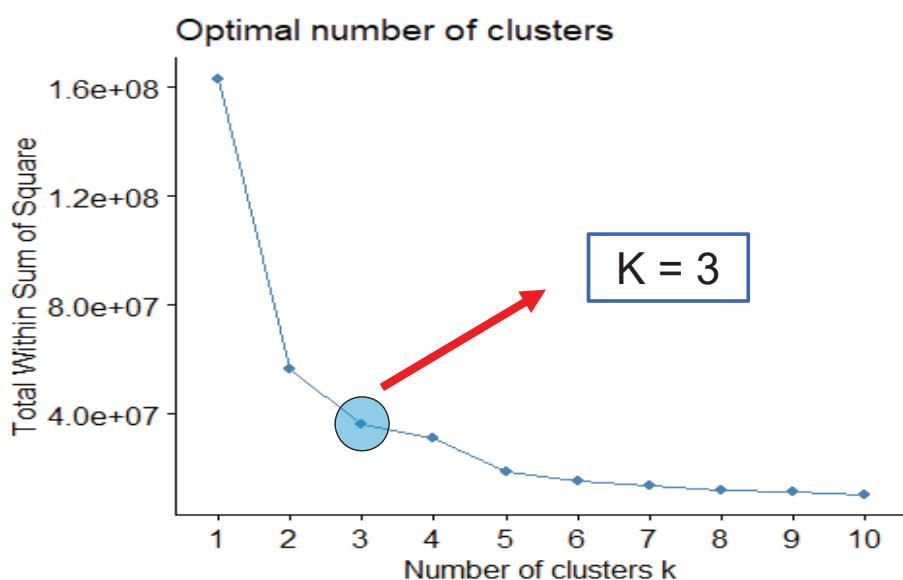
```

k3
## Clustering vector:
##      ALAVA    ALBACETE    ALICANTE    ALMERIA    AVILA    BADAJOZ
##      3          2          1          2          2          2          2
##      BALEARES  BARCELONA  BURGOS     CACERES    CADIZ    CASTELLON
##      3          3          3          2          2          2          1
##      C.REAL    CORDOBA  CORUÑA     CUENCA    GERONA  GRANADA
##      2          2          1          2          3          2          2
##      GUADALAJARA  GUIPUZCUA  HUELVA     HUESCA    JAEN    LEON
##      2          3          2          3          2          1
##      LERIDA    RIOJA     LUGO      MADRID    MALAGA  MURCIA
##      1          1          1          3          2          2          2
##      NAVARRA   ORENSE    OVIEDO     PALENCIA  LAS PALMAS  PONTEVEDRA
##      3          2          1          1          2          1
##      SALAMANCA S.C.DE TENER  SANTANDER  SEGOVIA  SEVILLA  SORIA
##      2          2          3          1          2          1
##      TARRAGONA  TERUEL    TOLEDO     VALENCIA  VALLADOLID  VIZCAYA
##      1          2          2          1          1          3
##      ZAMORA    ZARAGOZA  CEUTA     MELILLA
##      2          3          2          2
```

```
k3  
##  
## Within cluster sum of squares by cluster:  
## [1] 7911007 7540866 20596084  
## (between_SS / total_SS = 77.9 %)  
##  
## Available components:  
##  
## [1] "cluster"      "centers"       "totss"        "withinss"      "tot.withinss"  
## [6] "betweenss"    "size"         "iter"         "ifault"
```

¿Número de clusters ? (K)

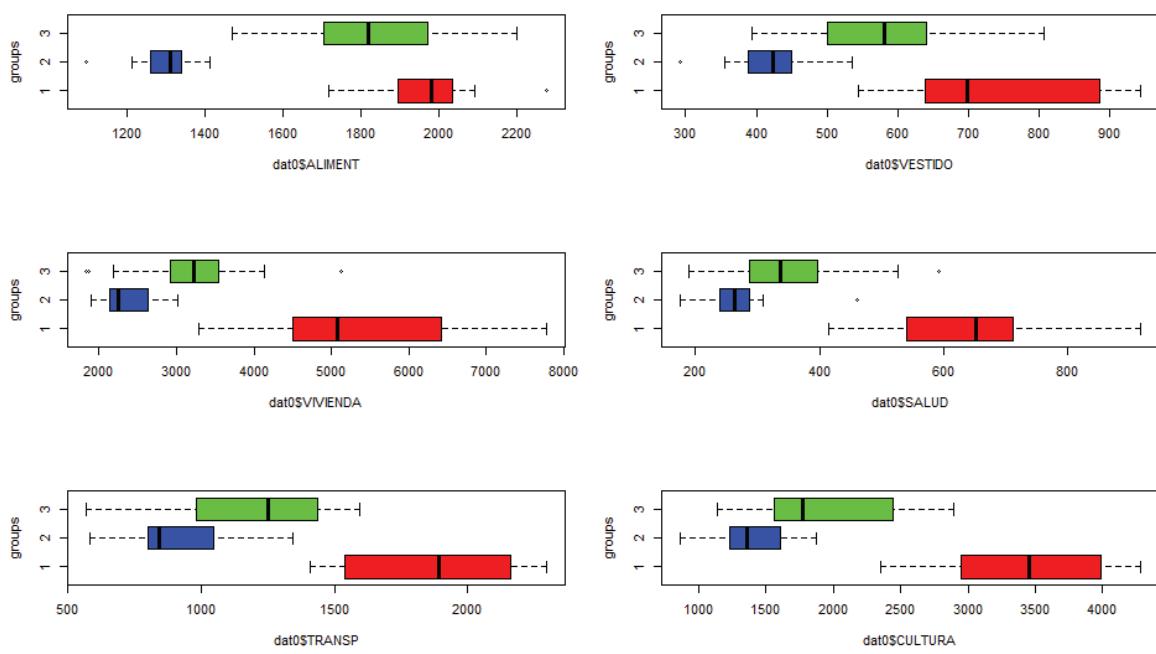
```
fviz_nbclust(dat0, kmeans, method = "wss")
```

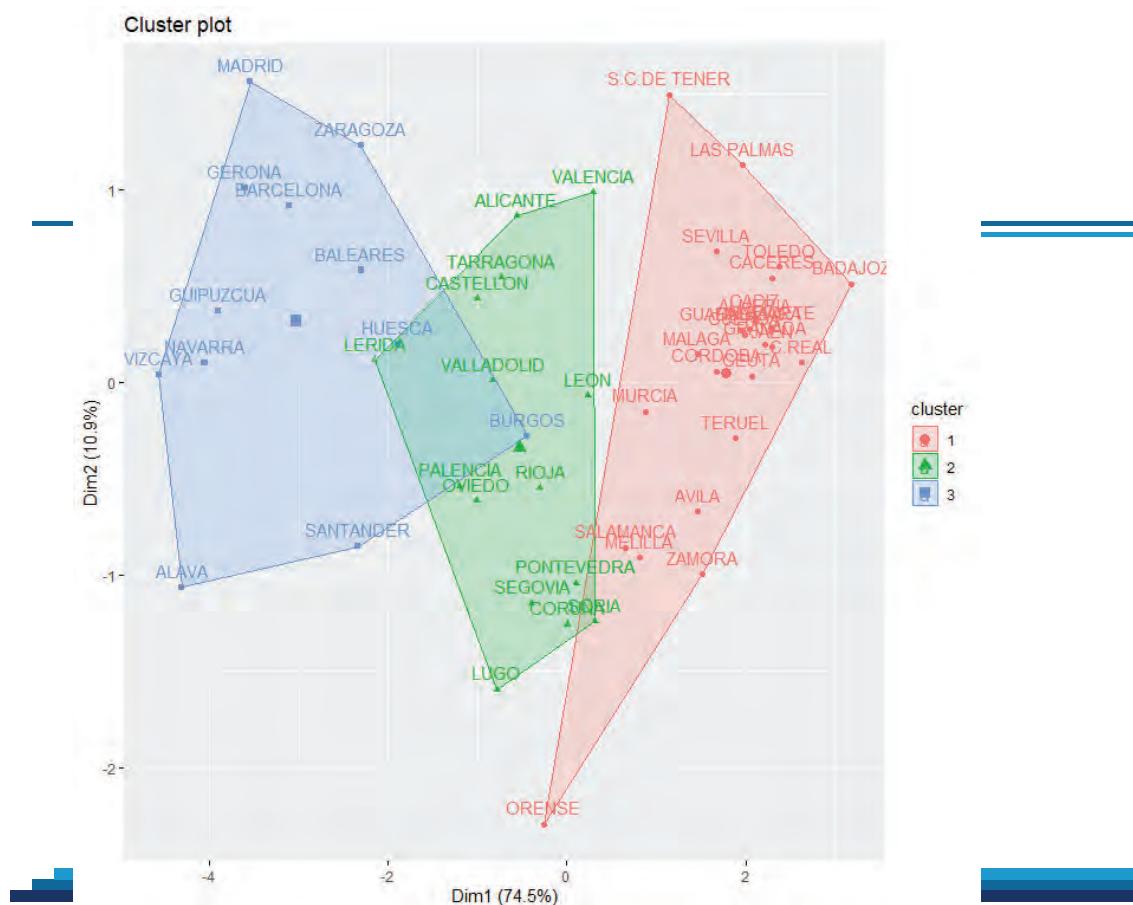


```

par(mfrow=c(3,2))
groups=k3$cluster
boxplot(dat0$ALIMENT~groups,col=c("red","blue","green"),
horizontal = T)
boxplot(dat0$VESTIDO~groups,col=c("red","blue","green"),
horizontal = T)
boxplot(dat0$VIVIENDA~groups,col=c("red","blue","green"),
horizontal = T)
boxplot(dat0$SALUD~groups,col=c("red","blue","green"),
horizontal = T)
boxplot(dat0$TRANSP~groups,col=c("red","blue","green"),
horizontal = T)
boxplot(dat0$CULTURA~groups,col=c("red","blue","green"),
horizontal = T)

```





Conclusiones

El Análisis Cluster incluye un conjunto de técnicas y algoritmos para clasificar observaciones en grupos que pueden proporcionar resultados muy diferentes. La interpretación y conclusiones que se derivan del análisis cluster deben ser tomadas con precaución.

Ejercicio 1: Wine Análisis Cluster

Wine dataset

El archivo “wine.txt” contiene los resultados del análisis químico de 178 vinos provenientes de una región italiana. Para cada vino se proporciona el contenido de varios compuestos y otras características, en total 13 variables (“Alcohol”, “Malic”, “Ash”, “Alcalinity”, “Magnesium”, “Phenols”, “Flavanoids”, “Nonflavanoids”, “Proanthocyanins”, “Color”, “Hue”, “Dilution” y “Proline”). La primera variable del archivo “Type” es una clasificación previa de los vinos que está basada en criterios subjetivos y que deseamos confirmar con nuestro análisis cluster. Solo será utilizada para evaluar los resultados del análisis cluster realizado.

Apartado 1.

Realizar el dendrograma con las 13 variables continuas utilizando la distancia euclídea y realizando el encadenamiento por el método de ward.D. Indica el número de observaciones que tiene si decidimos formar 3 clusters.

Apartado 2

Calcula la media de las 13 variables para cada grupo (centroide). Muestra gráficamente las diferencias entre los tres grupos para cada una de las variables utilizando gráficos boxplot o similares.

Apartado 3

Utiliza la función `fviz_cluster()` (del package `factoextra`) para mostrar en un gráfico de dimensión 2 la solución de tres grupos obtenida en el apartado 1

Apartado 4

Utilizando el procedimiento kmeans (con datos estandarizados) obtén la solución de tres grupos. Proyecta la solución en dimensión dos (utilizando la instrucción `fviz_cluster()`) y compara los resultados con la solución jerárquica de los apartados anteriores (ten en cuenta que el número asignado a cada cluster en cada solución es arbitrario)

Apartado 5

Compara la solución de kmeans con la clasificación inicial proporcionada en la variable Type. Explica las diferencias. Identifica las observaciones cuya asignación difiere entre uno y otro método.

Ejercicio 2 Análisis Clúster

Ejercicio de análisis clúster: MADRID 2021

En el archivo “Madrid_2021.txt” se proporciona los resultados por barrios de Madrid Capital de las elecciones a la Asamblea de Madrid celebradas el 4 de Mayo de 2021. Para cada barrio se proporciona el nombre, el censo y el porcentaje sobre el censo de votos a los partidos más votados: Vox, PP, Ciudadanos (Cs), PSOE, Más Madrid y Unidos-Podemos. Además se proporciona el porcentaje de votos a otras candidaturas y el porcentaje de abstención.

Los datos se han obtenido de la página web:

<https://datos.gob.es/es/catalogo/l01280796-elecciones-asamblea-de-madrid-1983-2019>

La variable **censo** se proporciona para completar la información, es muy relevante para el análisis de los resultados, pero en este ejercicio no se utiliza. Nos centraremos en las variables porcentaje de votos: vox, pp, cs, psoe, masmadrid, podemos, otros y abstención.

IMPORTANTE: El análisis se realizará con los datos sin estandarizar.

PREGUNTAS:

1. Describe mediante un boxplot múltiple los porcentajes de votos a vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion. Interpreta brevemente la gráfica. (Nota: si X es una tabla con múltiples columnas, boxplot(X) representa el diagrama de cajas de cada columna) **(1 punto)**
2. Obten la matriz de correlaciones de las variables vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion. Representa mediante gráficos de dispersión las relaciones entre las variables. Interpreta los resultados. **(1 punto)**
3. Realiza el dendrograma de los datos **sin estandarizar**, utilizando la distancia euclídea y el método de “ward.D2”. Según el árbol, obtén la solución con tres clusters. Indica el número de observaciones de cada cluster. **(1 punto)**
4. Calcula las medias del porcentaje de votos a cada partido (vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion) para cada cluster. Realiza un gráfico boxplot de los votos al pp para cada cluster. Lo mismo para psoe y abstencion. Asígnale un nombre a cada uno de los tres clusters y proporciona el nombre de los barrios del cluster con una media más alta de votos al PP. Utiliza la función *fviz_cluster()* del paquete *factoextra* para visualizar los tres grupos. En el gráfico identifica cada punto con el nombre del barrio. **(2 punto)**

5. Realiza el análisis cluster utilizando el método **kmeans**. Haz tres clusters (utiliza nstart=25). Se recomienda utilizar una semilla para inicializar el proceso de cálculo de manera que se puedan repetir las resultados. Proporciona el número de observaciones en cada cluster. **(1 punto)**
6. Repite los pasos del apartado 4 y describe la solución obtenida de kmeans. Compara los resultados de los dos métodos: hclust y kmeans. **(2 punto)**
7. Obten la solución de kmeans con 3, 4 y 5 grupos. Representa las tres soluciones utilizando la función *fviz_clustering()* e interpreta los resultados. **(1 punto)**
8. Resume brevemente las conclusiones del análisis cluster realizado en los apartados anteriores. Completa el análisis con aquello que te parezca interesante. **(1 punto)**

Enunciado: análisis clúster MADRID 2021

Solución

- 1 Boxplot
- 2 Correlaciones
- 3 Clúster Jerárquico
- 4 Clúster Jerárquico (desc)
- 5 Clúster Kmeans
- 6 Clúster kmeans (desc)
- 7 Más grupos
- 8 Conclusión

Tarea 3 Análisis Clúster

4ºGITI: Organización y Matemáticas

Prof: Eduardo Caro y Jesús Juan

Enunciado: análisis clúster MADRID 2021

En el archivo “Madrid_2021.txt” se proporciona los resultados por barrios de Madrid Capital de las elecciones a la Asamblea de la Comunidad de Madrid celebradas el 4 de Mayo de 2021. Para cada barrio el archivo incluye el nombre, el censo y el porcentaje sobre el censo de votos a los partidos más votados: Vox, PP, Ciudadanos (Cs), PSOE, Más Madrid y Unidas-Podemos. Además se proporciona el porcentaje de votos a otras candidaturas y el porcentaje de abstención.

Los datos se han obtenido de la página web:

[\(https://datos.gob.es/es/catalogo/I01280796-elecciones-asamblea-de-madrid-1983-2019\)](https://datos.gob.es/es/catalogo/I01280796-elecciones-asamblea-de-madrid-1983-2019)

La variable **censo** es el número de votantes en cada barrio, se proporciona para completar la información pero en este ejercicio no se utiliza. Nos centraremos en las variables porcentaje de votos: **vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion**

1. Describe mediante un boxplot múltiple los porcentajes de votos a vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion. Interpreta brevemente la gráfica. (Nota: si X es una tabla con múltiples columnas, boxplot(X) representa el diagrama de cajas de cada columna) (**1 punto**)
2. Obten la matriz de correlaciones de las variables vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion. Representa mediante gráficos de dispersión las relaciones entre las variables. Interpreta los resultados. (**1 punto**)
3. Realiza el dendrograma de los datos **sin estandarizar**, utilizando la distancia euclídea y el método de “ward.D2”. Según el árbol, obtén la solución con tres clústers. Indica el número de observaciones de cada clúster. (**1 punto**)

4. Calcula las medias del porcentaje de votos a cada partido (vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion) para cada clúster. Realiza un gráfico boxplot de los votos al pp para cada clúster. Lo mismo para psoe y abstencion. Asignale un nombre a cada uno de los tres clústers y proporciona el nombre de los barrios del clúster con una media más alta de votos al PP. Utiliza la función `fviz_cluster()` del paquete `factoextra` para visualizar los tres grupos. En el gráfico identifica cada punto con el nombre del barrio. **(2 punto)**
5. Realiza el análisis clúster utilizando el método **kmeans**. Haz tres clústers (utiliza `nstart=25`). Se recomienda utilizar una semilla para inicializar el proceso de cálculo de manera que se puedan repetir los resultados. Proporciona el número de observaciones en cada clúster. **(1 punto)**
6. Repite los pasos del apartado 4 y describe la solución obtenida de kmeans. Compara los resultados de los dos métodos: `hclust` y `kmeans`. **(2 punto)**
7. Obten la solución de kmeans con 3, 4 y 5 grupos. Representa las tres soluciones utilizando la función `fviz_clusuter()` e interpreta los resultados. **(1 punto)**
8. Resume brevemente las conclusiones del análisis clúster realizado en los apartados anteriores. Completa el análisis con aquello que te parezca interesante. **(1 punto)**

NOTA SOBRE LA EVALUACIÓN:

En este documento se proporciona la solución de la tarea 3. Esta solución incluyen detalles que no se solicitaban de forma específica en las preguntas y que lógicamente no son exigibles ni deben ser tenidas en cuenta a la hora de puntuar los ejercicios.

La solución obtenida mediante el método kmeans en algún trabajo puede ser diferente a la que se propone aquí. Debe tenerse en cuenta que el número asignado por el algoritmo kmeans a cada cluster es aleatorio y por tanto variará de una solución a otra, aunque la composición de los cluster (en general) deben parecerse bastante.

Solución

1 Boxplot

Describe mediante un boxplot múltiple los porcentajes de votos a vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion. Interpreta brevemente la gráfica. (Nota: si X es una tabla con múltiples columnas, boxplot(X) representa el diagrama de cajas de cada columna) **(1 punto)**

```
dat = read.table("madrid_2021.txt",header=TRUE)
names(dat)
```

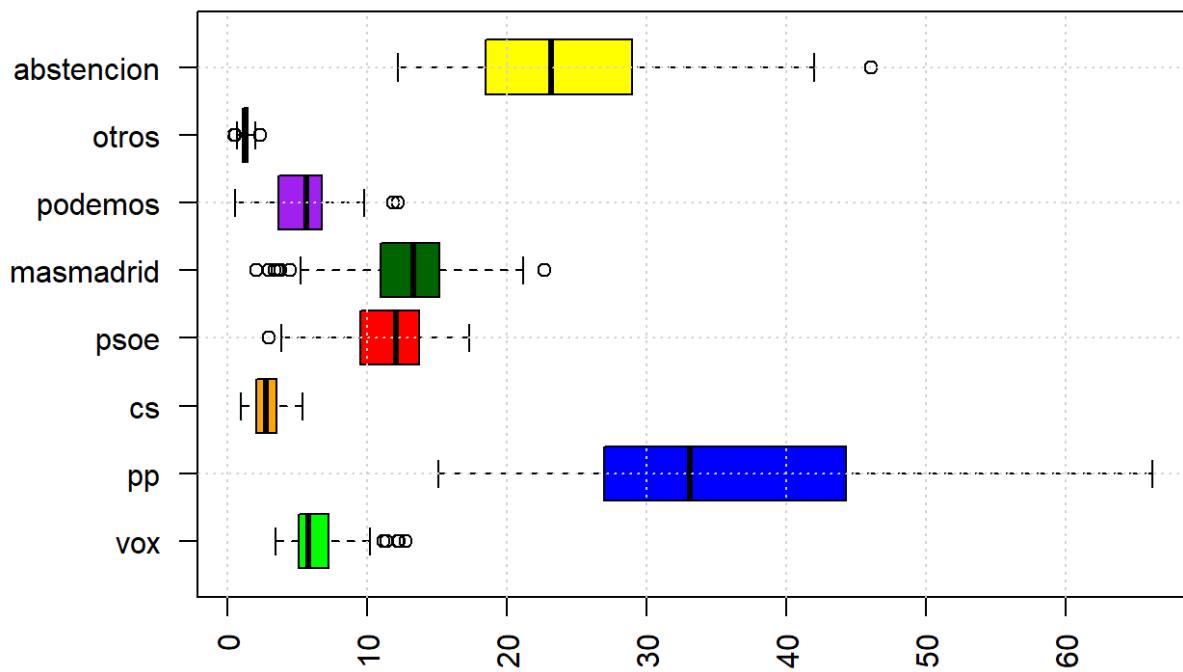
```
## [1] "barrio"      "censo"       "pp"          "psoe"        "cs"
## [6] "vox"         "masmadrid"    "podemos"     "otros"        "abstencion"
```

En las siguientes instrucciones hago tres cosas:

- Primero creo un data.frame (x) con las variables que voy a utilizar.
- REordeno las variables para facilitar la interpretación y
- Utilizo el nombre del barrio para identificar cada fila. Esto es clave si queremos que aparezca el nombre del barrio en los gráficos.

```
x = dat[,c(6,3,5,4,7,8,9,10)]
row.names(x)=dat$barrio
```

```
par(mar=c(5.1, 6.1, 4.1, 2.1)) # aumento el margen izquierdo de la figura
colores=c("green","blue","orange","red","darkgreen","purple","lightblue","yellow")
b=boxplot(x, horizontal = T, col = colores,las=2)
grid()
```



Boxplot de porcentaje de votos a los distintos partidos

El boxplot (Figura @ref(fig:boxplot)) es una representación magnífica para visualizar los resultados electorales por barrios. La victoria del PP es evidente. En el barrio que menos le ha votado obtuvo un 15.1% y el que más, el 66.2%. Las candidaturas de Más-Madrid y PSOE tienen resultados similares, y Podemos claramente inferiores a las dos anteriores. Los resultados de Vox son similares a Podemos. La abstención es muy elevada, entre el 12% y el 46.1%

La información del boxplot se puede obtener de la siguiente forma (previamente hemos hecho `b = boxplot()`).

```
b
```

```

## $stats
##      [,1] [,2] [,3]  [,4] [,5] [,6]  [,7] [,8]
## [1,]  3.5 15.1 1.00  3.90  5.3 0.60 0.70 12.2
## [2,]  5.1 27.0 2.10  9.55 11.0 3.65 1.15 18.5
## [3,]  5.8 33.1 2.80 12.10 13.3 5.70 1.30 23.2
## [4,]  7.3 44.3 3.55 13.75 15.2 6.75 1.50 29.0
## [5,] 10.2 66.2 5.40 17.30 21.2 9.80 2.00 42.0
##
## $n
## [1] 131 131 131 131 131 131 131 131
##
## $conf
##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
## [1,] 5.4963 30.71182 2.599834 11.52021 12.72021 5.27206 1.251684 21.75052
## [2,] 6.1037 35.48818 3.000166 12.67979 13.87979 6.12794 1.348316 24.64948
##
## $out
## [1] 11.2 12.2 11.4 12.8 12.3  3.0 22.7  3.6  3.4  3.8  3.0  2.1  4.5 12.2 11.9
## [16] 0.6  0.5  0.6  2.4 46.1
##
## $group
## [1] 1 1 1 1 1 4 5 5 5 5 5 5 6 6 7 7 7 7 8
##
## $names
## [1] "vox"        "pp"         "cs"         "psoe"       "masmadrid"
## [6] "podemos"    "otros"      "abstencion"

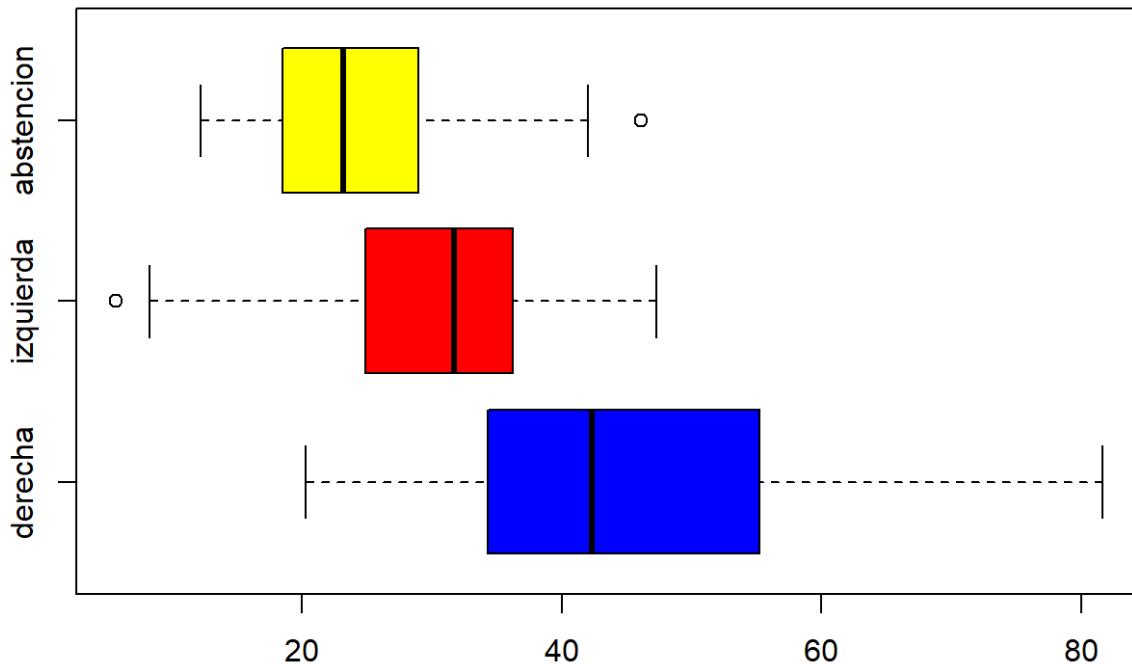
```

Agrupando las candidaturas en derecha e izquierda, el boxplot resultante es el de la figura @ref(fig:boxplot2). Se aprecia el apoyo mayoritario recibido por los partidos de derecha en Madrid capital (en este análisis no se tienen en cuenta el número de electores de cada barrio, ni tampoco los votos en los pueblos de la Comunidad de Madrid)

```

xder = x$vox+x$pp+x$cs
xizq = x$psoe + x$masmadrid + x$podemos
boxplot(cbind(derecha=xder, izquierda=xizq, abstencion = x$abstencion),
        col = c("blue","red","yellow"),horizontal=TRUE)

```



Boxplot de porcentaje de votos agregados en candidaturas de derecha, izquierda y abstención

Los barrios que más y menos votaron al PP fueron:

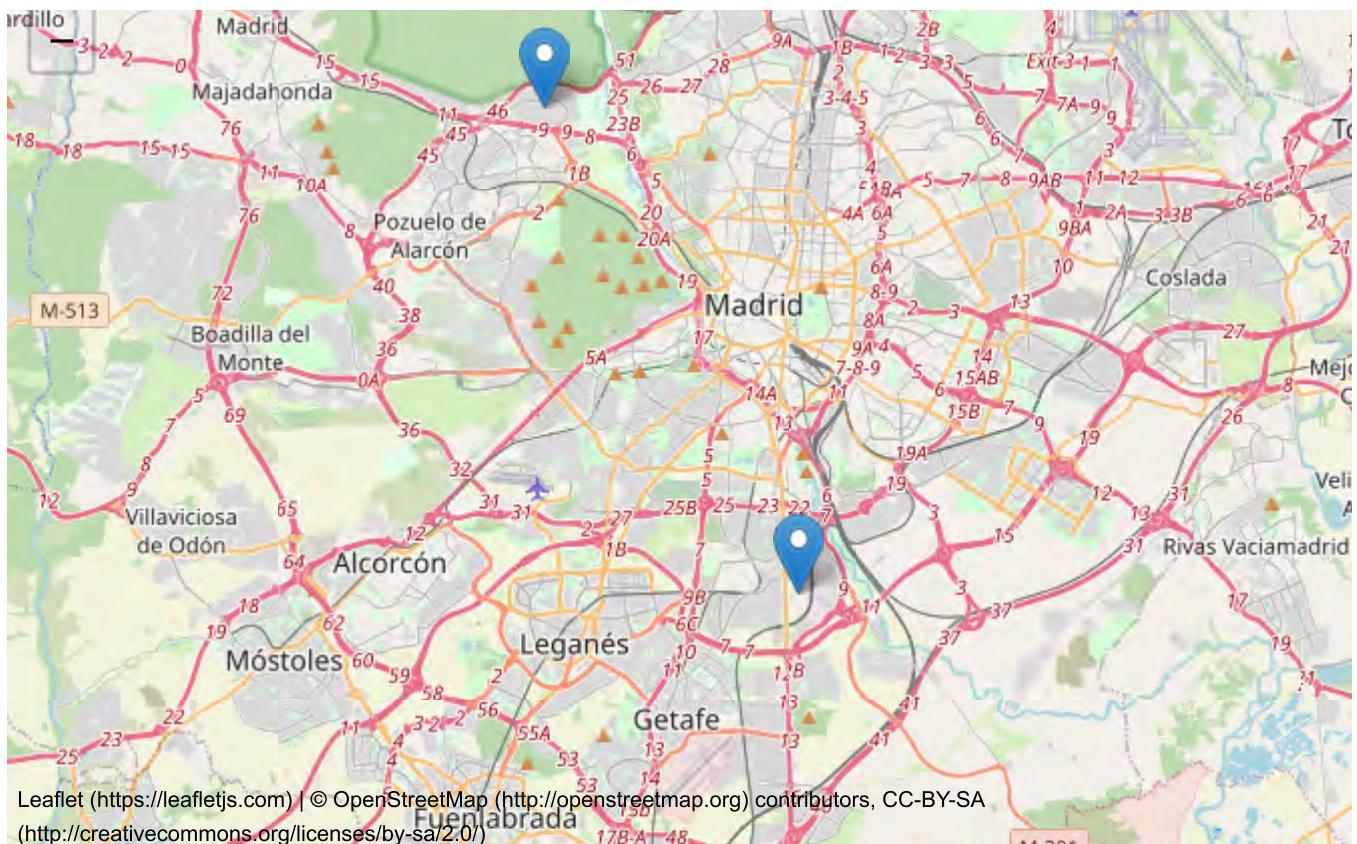
```
ppmax = which.max(x$pp)
ppmin = which.min(x$pp)
x[c(ppmax,ppmin),]
```

```
##          vox   pp   cs psoe masmadrid podemos otros abstencion
## Valdemarin 12.8 66.2 2.6  3.0      2.1     0.6   0.5    12.2
## San Cristobal 4.2 15.1 1.0 14.1     11.0     7.0   1.6   46.1
```

Los dos son barrios del extrarradio de Madrid, San Cristóbal de los Ángeles al sur cerca de Villaverde Bajo y Valdemarín al Noroeste, cerca de Pozuelo y Aravaca. Con R se puede incrustar en el documento un mapa con la situación de los dos barrios. Se requiere la latitud y longitud de cada barrio, que se consigue en Google Map. En la versión html de este documento se incluye el mapa que indica la situación de los dos barrios citados.

```
library(leaflet)
library(Rcpp)
m <- leaflet()
m <- addTiles(m)
m <- addMarkers(m, lng=c(-3.77590736943623,-3.68873), lat=c(40.46792712897,40.34077),
               popup=c("Valdemarín","San Cristóbal"), label =c("Valdemarín","San Cristóbal"))
setView(m,lng=-3.729, lat=40.40,zoom=11)
```



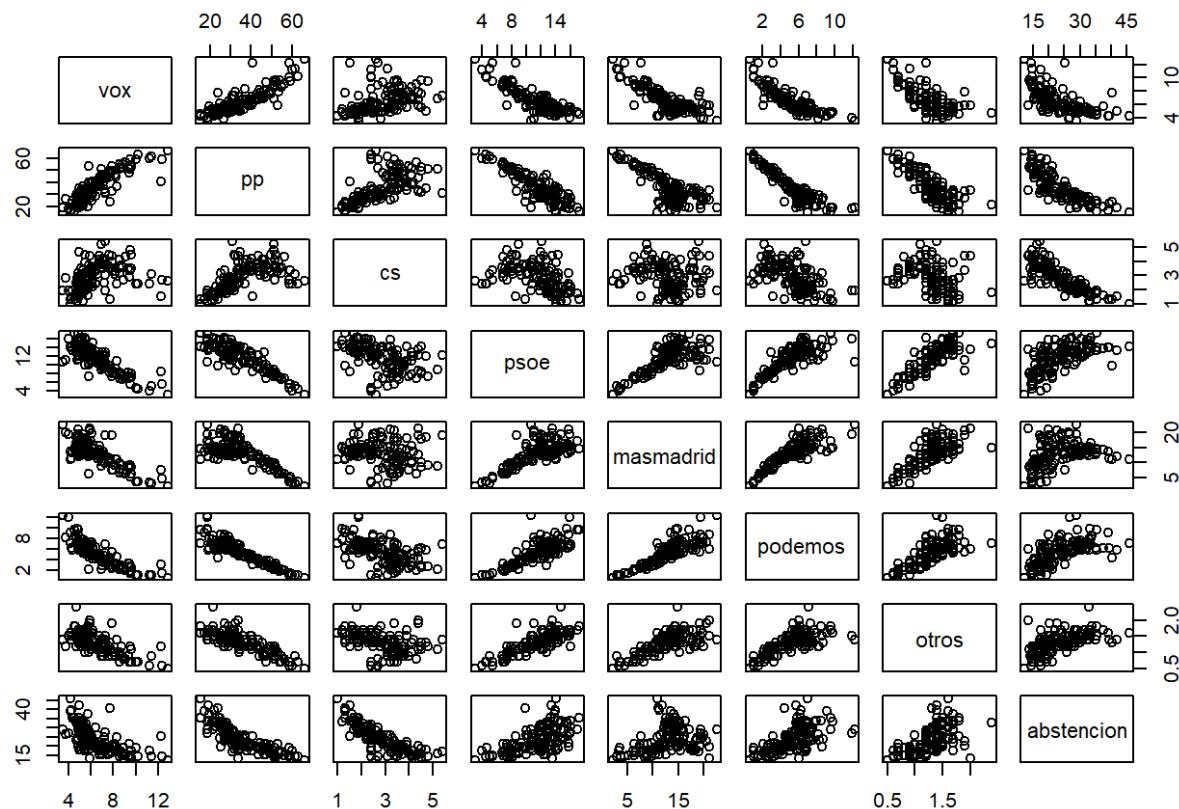


#m # Imprime el mapa

2 Correlaciones

Obten la matriz de correlaciones de las variables vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion. Representa mediante gráficos de dispersión las relaciones entre las variables. Interpreta los resultados. (1 punto)

```
pairs(x)
```



Gráficos de dispersión de porcentaje de votos a cada partido

Los gráficos @ref(fig:pairs) muestran relaciones bastante lineales entre las variables, con correlaciones positivas y negativas. A continuación se proporciona la matriz de correlaciones.

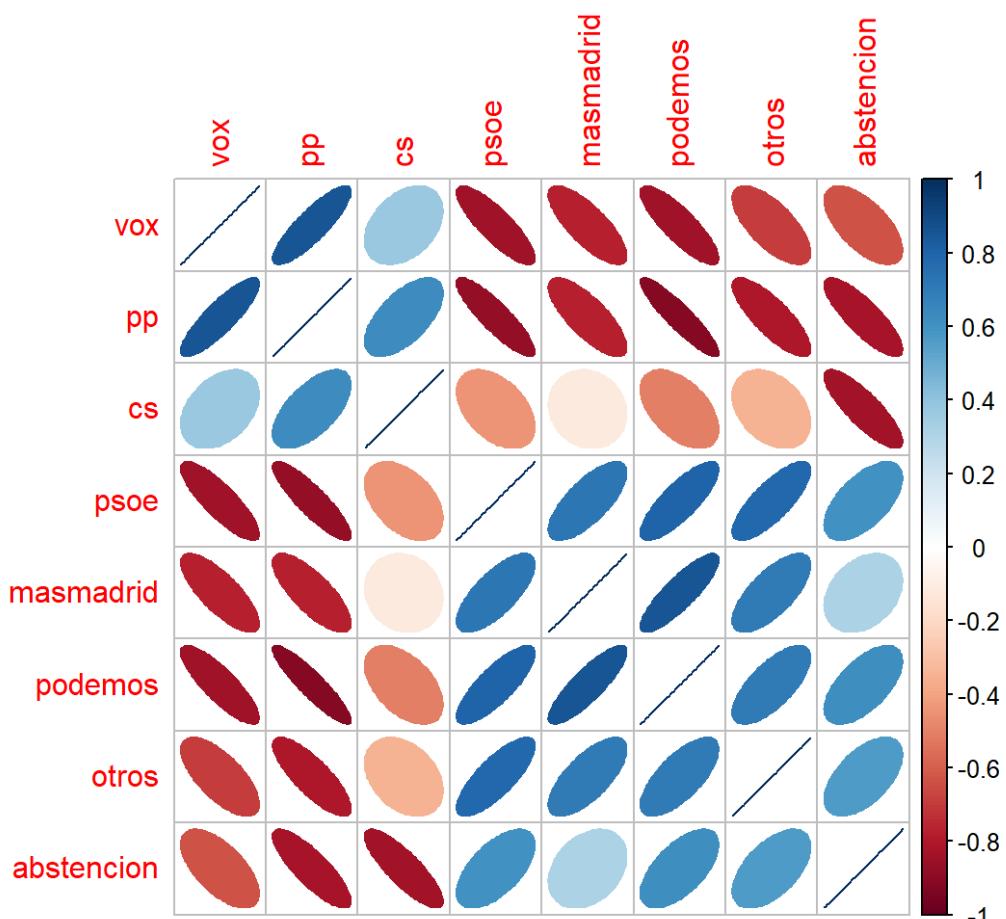
```
r=cor(x)  
print(r,digits = 3)
```

	vox	pp	cs	psoe	masmadrid	podemos	otros	abstencion
## vox	1.000	0.858	0.380	-0.842	-0.779	-0.843	-0.693	-0.637
## pp	0.858	1.000	0.620	-0.877	-0.772	-0.910	-0.809	-0.830
## cs	0.380	0.620	1.000	-0.447	-0.118	-0.501	-0.348	-0.835
## psoe	-0.842	-0.877	-0.447	1.000	0.729	0.809	0.785	0.605
## masmadrid	-0.779	-0.772	-0.118	0.729	1.000	0.854	0.707	0.316
## podemos	-0.843	-0.910	-0.501	0.809	0.854	1.000	0.706	0.619
## otros	-0.693	-0.809	-0.348	0.785	0.707	0.706	1.000	0.566
## abstencion	-0.637	-0.830	-0.835	0.605	0.316	0.619	0.566	1.000

Del análisis de correlaciones se puede destacar:

- Es muy llamativa la estructura de la matriz de correlaciones. Correlaciones positivas entre partidos de derecha (PP, VOX y CS) por un lado, correlaciones positivas entre partidos de izquierda (PSOE, Más Madrid y Podemos) por otro y correlaciones negativas entre partidos de ideología diferentes. Este efecto se aprecia en el gráfico realizado con `corrplot()`
- Existen correlaciones **positivas** muy altas entre partidos de misma ideología (derecha o izquierda). Por ejemplo entre PP y Vox, 0.858, y entre PSOE y Podemos 0.809, Podemos y Más Madrid 0.809. El coeficiente entre PP y Vox se interpreta como sigue: Los barrios que votan mucho al PP (por encima de la media), también votan mucho al Vox (por encima de la media), los que votan poco al PP, votan poco a Vox.
- Existen correlaciones **negativas** muy altas entre partidos de ideologías contrarias. Por ejemplo entre PSOE y Vox, -0.842, y entre PP y Podemos -0.910. Los barrios que votan mucho a un partido de izquierda, votan poco a los partidos de derecha.
- Visto lo anterior, parece que la Abstención está “asociada” a los partidos de izquierda. Los barrios que votan mucho a la izquierda, tienen abstención alta y los que votan poco a la izquierda (y por tanto votan a la derecha) tienen abstención baja.
- La variable “Otros” tiene un comportamiento semejante a la abstención. Lo que induce a pensar que son votantes de izquierda.

```
library(corrplot)
corrplot(r,method = "ellipse")
```



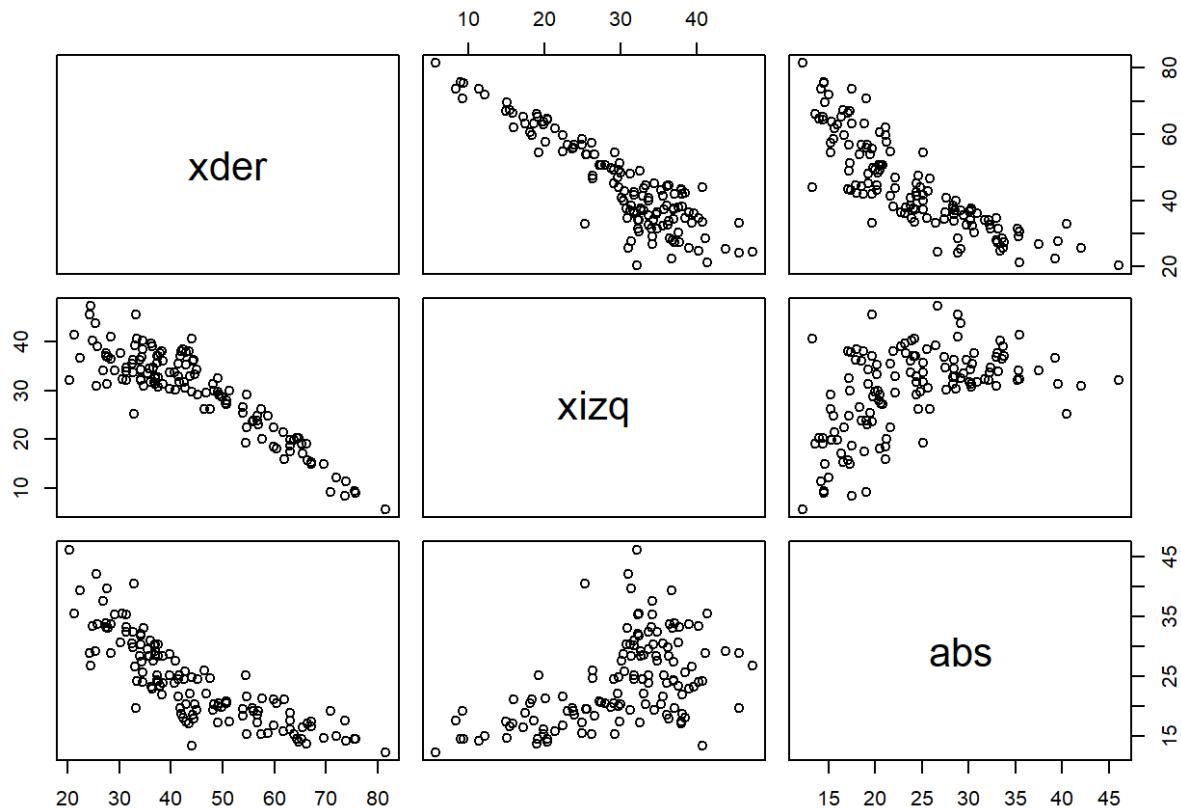
Si sumamos los votos de los partidos de derecha por una lado ($xder = pp + vox + cs$), los de izquierda por otro ($xizq = psoe + masmadrid + podemos$) y la abstención por otro, tenemos los siguientes resultados: Existe una correlación negativa muy fuerte entre votos a derecha e izquierda. La gráfica

muestra clara linealidad. La relación entre derecha y abstención, también es lineal y muy alta. Es curiosa la gráfica entre abstención e izquierda, con un primer tramo creciente lineal y segunda parte más horizontal. En este caso la relación no es lineal.

```
r3 = cor(cbind(xder,xizq,abs=x$abstencion))
r3
```

```
##          xder      xizq      abs
## xder  1.0000000 -0.8988595 -0.8396046
## xizq -0.8988595  1.0000000  0.5169506
## abs   -0.8396046  0.5169506  1.0000000
```

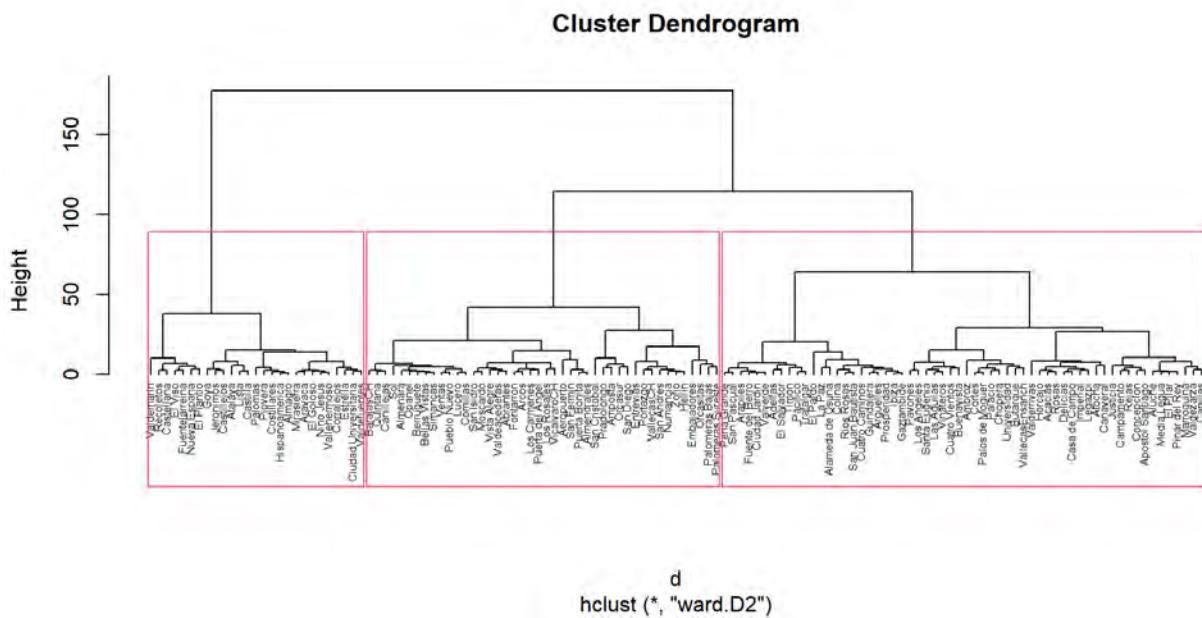
```
pairs(cbind(xder,xizq,abs=x$abstencion))
```



3 Clúster Jerárquico

Realiza el dendrograma de los datos sin estandarizar, utilizando la distancia euclídea y el método de “ward.D2”. Segundo el árbol, obtén la solución con tres clústeres. Indica el número de observaciones de cada clúster. (1 punto)

```
d = dist(x)
h = hclust(d,method = "ward.D2")
g3 = cutree(h,k=3)
plot(h,cex=.5,hang=-1)
rect.hclust(h,k=3)
```



```
table(g3)
```

```
## g3
## 1 2 3
## 60 44 27
```

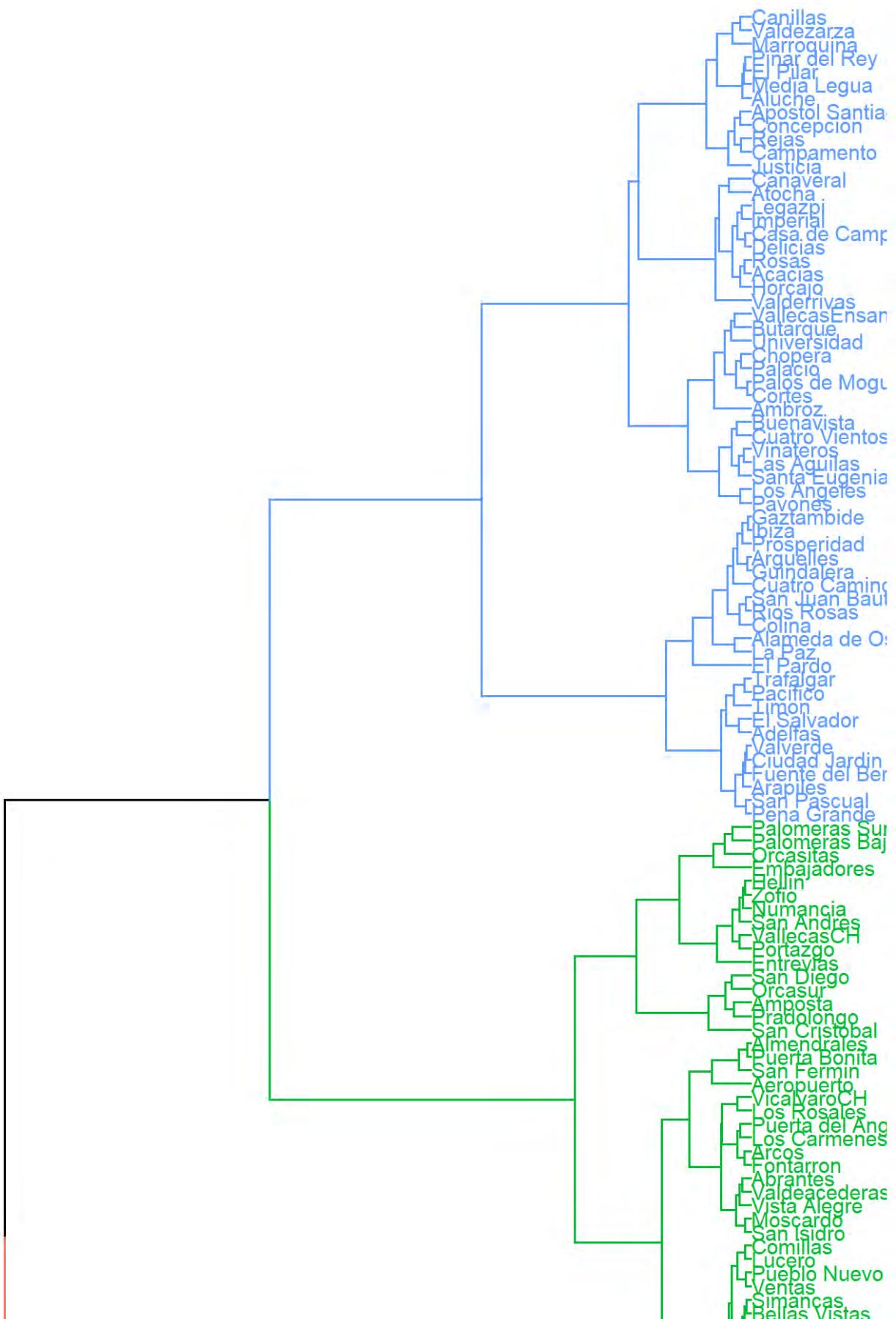
Viendo el dendrograma se aprecia que hay varias posibilidades de agrupamiento. Hay una posible solución de 3 clústeres, también podría ser interesante analizar la solución de 4 o 5 clústeres. La elección del número de clústeres depende de muchos aspectos. Cuando realmente las observaciones están *dispuestas en el espacio* formando grupos, lo ideal es encontrar la partición que corresponde a los grupos reales. En la mayoría de las ocasiones la disposición espacial no se corresponde con la existencia de grupos, aún así puede tener interés clasificar las observaciones. En este caso la elección del número de clústeres es más subjetiva. En el último apartado veremos una manera de elegir el número de clústeres.

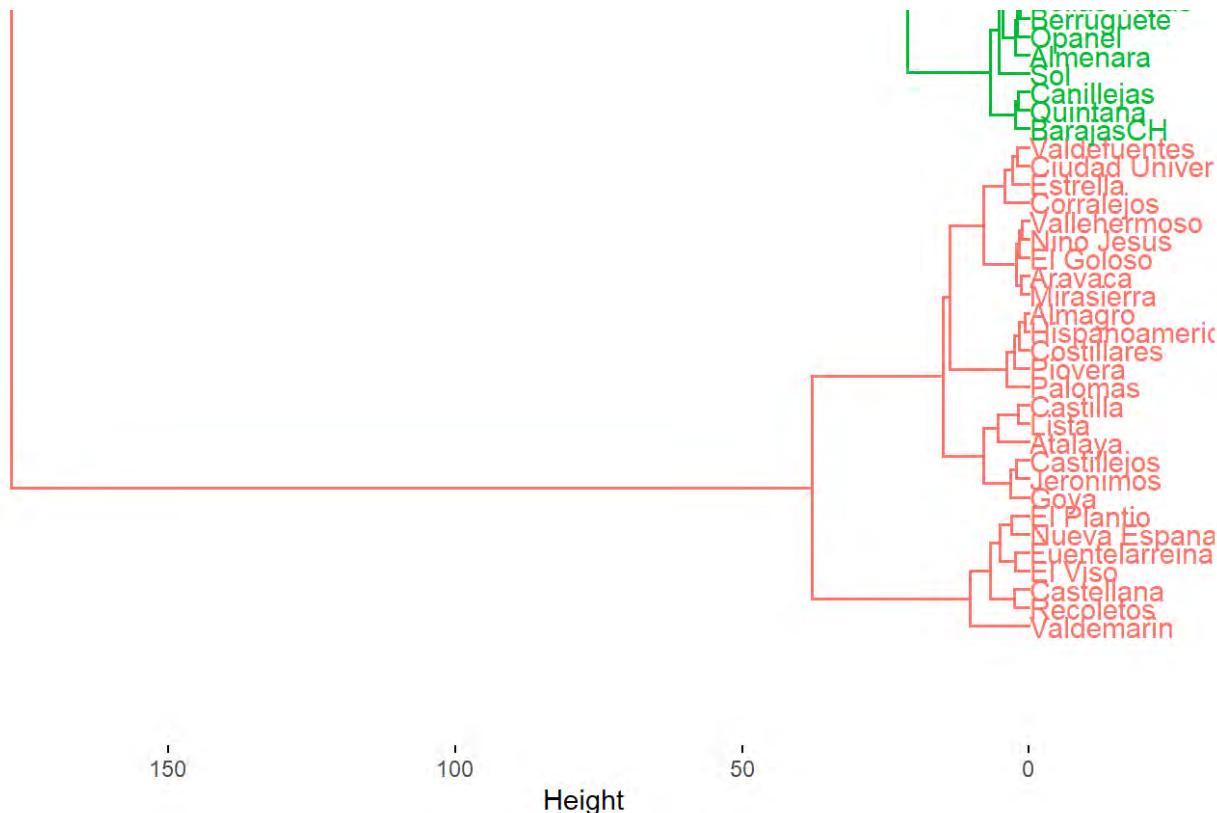
La solución de 3 clústeres es bastante razonable (aunque hay otras opciones). A continuación se muestran los barrios de cada uno de los clúster (estas etiquetas se pueden ver en el dendrograma). El clúster 1 es el más numeroso con 60 barrios, está a la derecha del dendrograma. El clúster 2 tiene 44 observaciones, está en el centro del dendrograma. El clúster 3 tiene 27 observaciones está a la izquierda del dendrograma. Si miramos el árbol, los clúster 2 y 3 están más próximos entre sí (las ramas se unen a una distancia de aprox. 100). El clúster 3 está a mayor distancia de los otros dos.

El dendrograma en horizontal permite leer las etiquetas más fácilmente.

```
library(factoextra)
fviz_dend(h, horiz = T, k=3, cex=.8)
```

Cluster Dendrogram





El nombre de los barrios de cada cluster se proporciona a continuación y el análisis de los clústers se hace en las siguientes secciones.

```
row.names(x)[g3==1]
```

```
## [1] "Palacio"          "Cortes"           "Justicia"
## [4] "Universidad"       "Imperial"         "Acacias"
## [7] "Choperia"          "Legazpi"          "Delicias"
## [10] "Palos de Moguer"   "Atocha"           "Pacifico"
## [13] "Adelfas"           "Ibiza"             "Fuente del Berro"
## [16] "Guindalera"        "Prosperidad"      "Ciudad Jardin"
## [19] "Cuatro Caminos"    "Gaztambide"       "Arapiles"
## [22] "Trafalgar"         "Rios Rosas"       "El Pardo"
## [25] "Pena Grande"       "El Pilar"          "La Paz"
## [28] "Valverde"          "Casa de Campo"    "Arguelles"
## [31] "Valdezarza"         "Aluche"            "Campamento"
## [34] "Cuatro Vientos"    "Las Aguilas"       "Buenavista"
## [37] "Pavones"            "Horcajo"           "Marroquina"
## [40] "Media Legua"        "Vinateros"          "Concepcion"
## [43] "San Pascual"        "San Juan Bautista" "Colina"
## [46] "Canillas"           "Pinar del Rey"     "Apostol Santiago"
## [49] "Butarque"           "Los Angeles"       "Santa Eugenia"
## [52] "VallecasEnsanch"   "Ambroz"            "Valderrivas"
## [55] "Canaveral"          "Rosas"              "Rejas"
## [58] "El Salvador"         "Alameda de Osuna"  "Timon"
```

```
row.names(x)[g3==2]
```

```

## [1] "Embajadores"      "Sol"           "Bellas Vistas"
## [4] "Almenara"         "Valdeacederas"   "Berruguete"
## [7] "Los Carmenes"     "Puerta del Angel" "Lucero"
## [10] "Comillas"        "Opanel"         "San Isidro"
## [13] "Vista Alegre"    "Puerta Bonita"   "Abrantes"
## [16] "Orcasitas"        "Orcasun"        "San Fermin"
## [19] "Almendrales"     "Moscardo"       "Zofio"
## [22] "Pradolongo"     "Entrevias"      "San Diego"
## [25] "Palomeras Bajas" "Palomeras Sureste" "Portazgo"
## [28] "Numancia"         "Fontarron"      "Ventas"
## [31] "Pueblo Nuevo"    "Quintana"       "San Andres"
## [34] "San Cristobal"   "Los Rosales"    "VallecasCH"
## [37] "VicalvaroCH"     "Simancas"       "Hellin"
## [40] "Amposta"         "Arcos"          "Canillejas"
## [43] "Aeropuerto"       "BarajasCH"      ""

```

```
row.names(x)[g3==3]
```

```

## [1] "Estrella"          "Jeronimos"        "Nino Jesus"
## [4] "Recoletos"         "Goya"             "Lista"
## [7] "Castellana"        "El Viso"          "Hispanoamerica"
## [10] "Nueva Espana"     "Castilla"         "Castillejos"
## [13] "Almagro"           "Vallehermoso"    "Fuentelarreina"
## [16] "Miras Sierra"     "El Goloso"        "Ciudad Universitaria"
## [19] "Valdemarin"       "El Plantio"      "Aravaca"
## [22] "Atalaya"          "Costillares"      "Palomas"
## [25] "Piovera"          "Valdefuentes"    "Corralejos"

```

4 Clúster Jerárquico (desc)

Calcula las medias del porcentaje de votos a cada partido (vox, pp, cs, psoe, masmadrid, podemos, otros y abstencion) para cada clúster. Realiza un gráfico boxplot de los votos al pp para cada clúster. Lo mismo para psoe y abstencion. Asígnale un nombre a cada uno de los tres clústers y proporciona el nombre de los barrios del clúster con una media más alta de votos al PP. Utiliza la función `fviz_cluster()` del paquete `factoextra` para visualizar los tres grupos. En el gráfico identifica cada punto con el nombre del barrio. (2 punto)

Vamos a calcular las medias de cada clúster.

```
K=3  
mg = NULL  
for(k in 1:K){  
  mg = rbind(mg,sapply(x[g3==k,],mean))  
}  
print(mg,digits = 3)
```

```
##      vox   pp   cs  psoe masmadrid podemos otros abstencion  
## [1,] 6.23 35.6 3.29 12.07    14.78     5.45 1.350     21.2  
## [2,] 5.14 24.3 1.88 13.56    14.30     6.96 1.518     32.3  
## [3,] 9.10 53.8 3.48  6.98     7.05     2.23 0.881     16.5
```

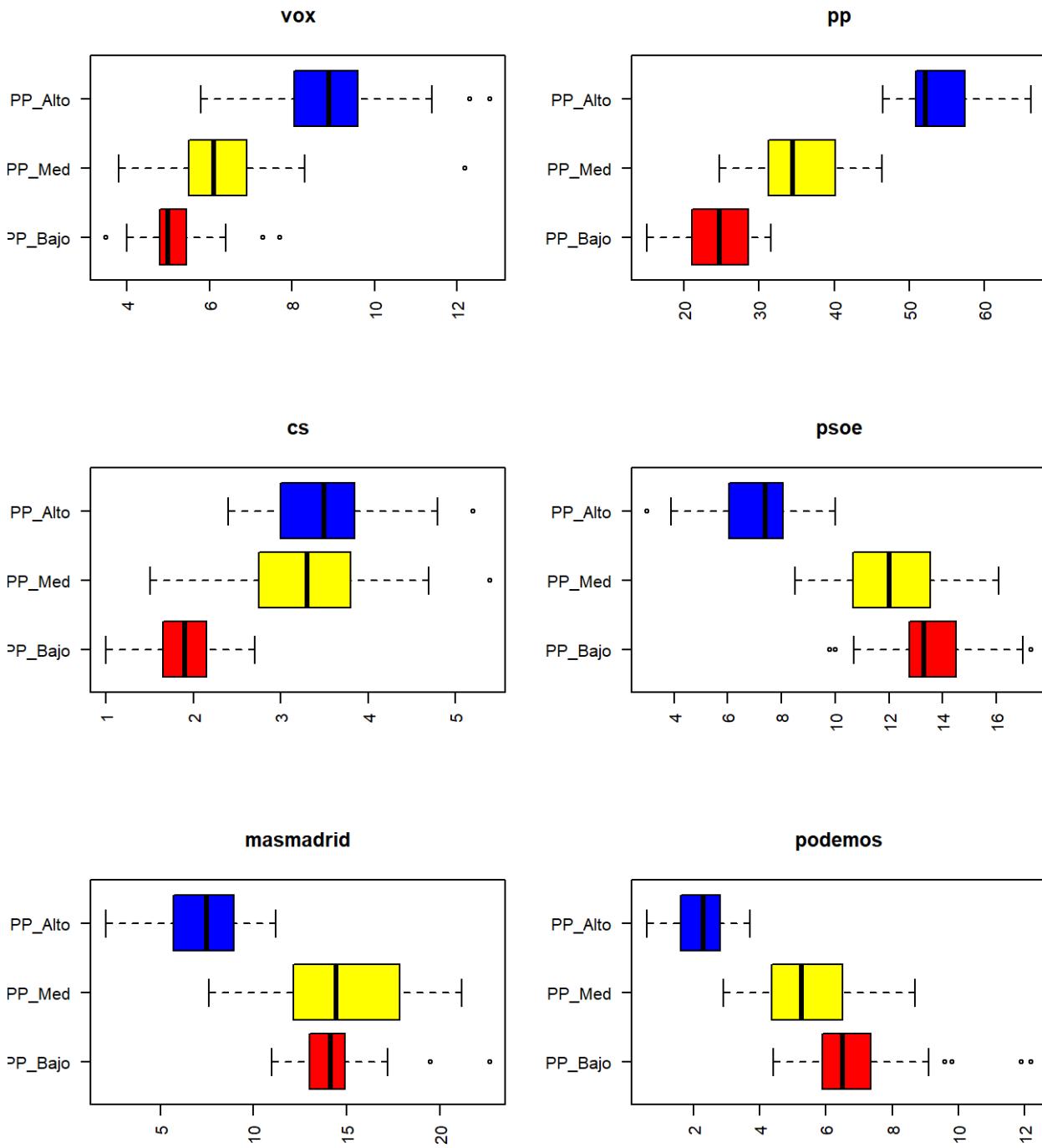
Vemos que el clúster 3 tiene un apoyo muy alto al PP con media del 53.8%. En el grupo 2 el apoyo es claramente menor, media 24.3% y el clúster 1 está a mitad de camino. Para identificar los clúster, voy a denominar al grupo 3, **PP_Alto**, al grupo 2, **PP_Bajo** y al grupo 1 **PP_Medio**. Se podrían haber elegido las etiquetas Derecha (PP_ALto), Centro (PP_Medio) e Izquierda (PP_Bajo).

```
row.names(mg) = c("PP_Med","PP_Bajo","PP_Alto")  
print(mg,digits = 3)
```

```
##      vox   pp   cs  psoe masmadrid podemos otros abstencion  
## PP_Med 6.23 35.6 3.29 12.07    14.78     5.45 1.350     21.2  
## PP_Bajo 5.14 24.3 1.88 13.56    14.30     6.96 1.518     32.3  
## PP_Alto 9.10 53.8 3.48  6.98     7.05     2.23 0.881     16.5
```

Los gráficos boxplot de cada variable para cada clúster nos ayuda a interpretar la clasificación.

```
par(mfrow=c(3,2))  
titulo = c("vox","pp","cs", "psoe", "masmadrid", "podemos")  
c3=factor(g3,labels = c("PP_Med","PP_Bajo","PP_Alto"))  
c3=relevel(c3,ref="PP_Bajo")  
for (k in 1:6){  
  boxplot(x[,k]~c3, horizontal = TRUE,  
          col=c("red","yellow","blue"),  
          main = titulo[k],las=2,ylab="",xlab="")  
}
```



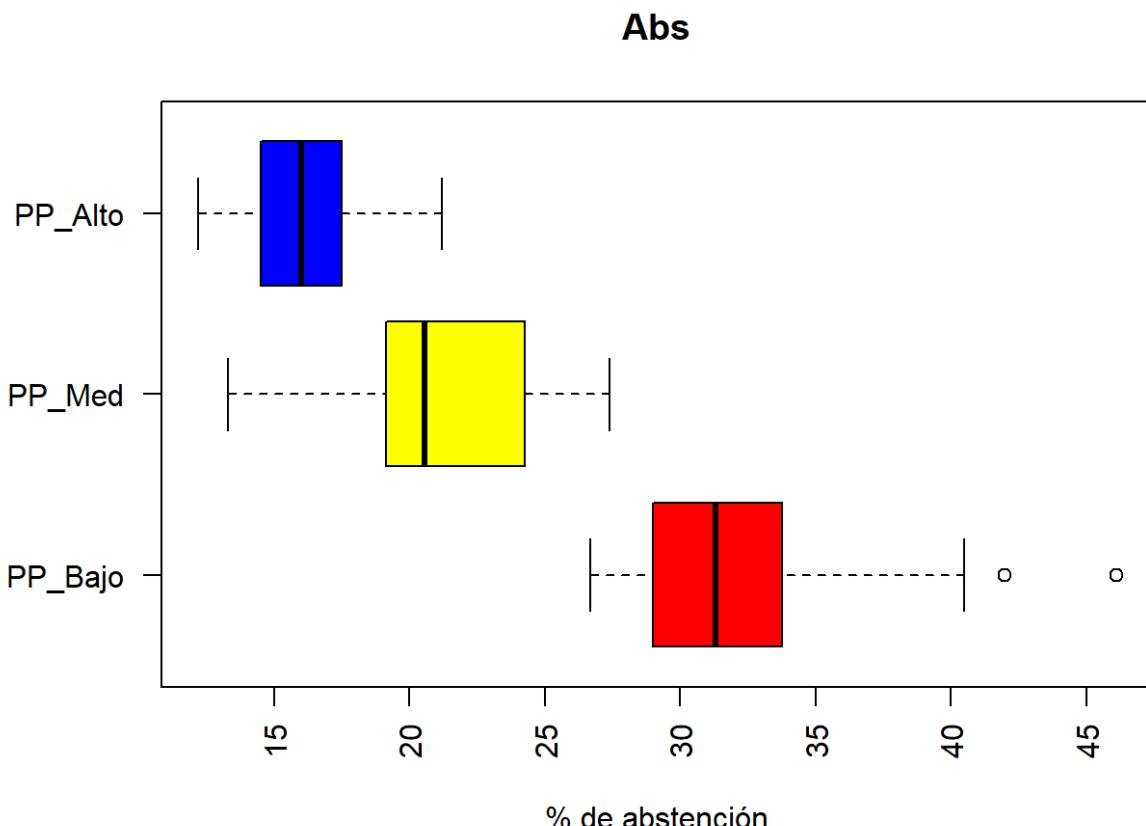
El boxplot del PP sirve de referencia, las tres gráficas de caja están de acuerdo con la denominación asignada: PP_Alto a la derecha, etc. La figura de Vox es similar. Los votos del PSOE es especular con el PP, en el cluster "PP_Alto" PSOE es bajo, etc. Lo mismo ocurre con Podemos. Ciudadanos solo obtiene porcentajes altos en los barrios donde hay voto alto o medio al PP. "Masmadrid" consigue más votos en los cluster PP_Medio y PP_Bajo. Importante tener en cuenta que las escalas de los ejes de abcisas en las figuras son muy diferentes: para el PP entre 25 y 65% mientras que para Cs, va del 1% al 6% (aprox).

El boxplot de la abstención tiene una interpretación clara y coincide con lo indicado en el análisis de correlaciones. La mayor abstención aparece en los barrios que votan en mayor proporción a los partidos de izquierda. La diferencia en el porcentaje de abstención en los tres clúster es enorme.

```

par(mar=c(5.1, 6.1, 4.1, 2.1))
boxplot(x[,8]~c3, horizontal = TRUE,
        col=c("red","yellow","blue"),main = "Abs",
        xlab = "% de abstención",las=2,
        ylab ="")

```



Con `fviz_cluseter()` presenta la proyección de los datos que están en dimensión 8 en el plano de dimensión 2 determinado por la solución de componentes principales de las variables estandarizadas. Se aprecia que los grupos se forman alineados con el primer componente principal. Los barrios que votan más a la derecha tienen puntuaciones positivas en el primer componente (y están a la derecha, de color azul) y en el otro extremo del eje, los que votan menos a la derecha con puntuaciones negativas en el primer componente (de color rojo).

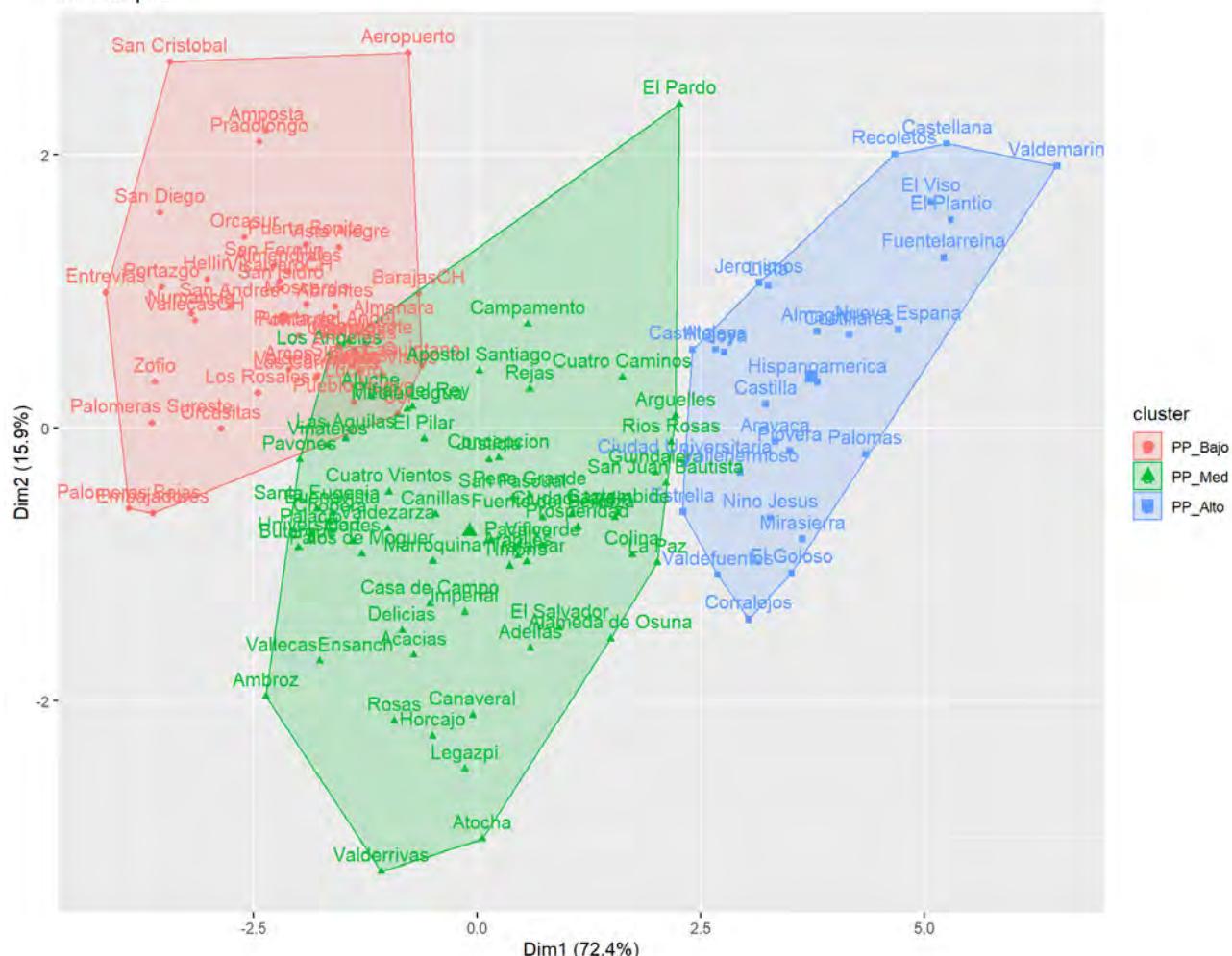
En este gráfico observamos qué barrios se parecen y tienen resultados similares. En el grupo de **PP_Alto** están Castellana, Recoletos, El Viso, Mirasierra, ... En el otro extremo, grupo **PP_Bajo**, Orcasitas, Vallecás, Entrevías, San Cristóbal, etc. También se aprecia que las observaciones no forman grupos aislados, hay barrios que se encuentran en la frontera de dos clúster y que podrían caer del lado de uno u otro, dependiendo el método de clasificación elegido elegido. Sería interesante utilizar variables socioeconómicas de cada barrio para comprobar su relación con la distribución de voto.

```

library(factoextra)
fviz_cluster(list(data=x,clusters=c3))

```

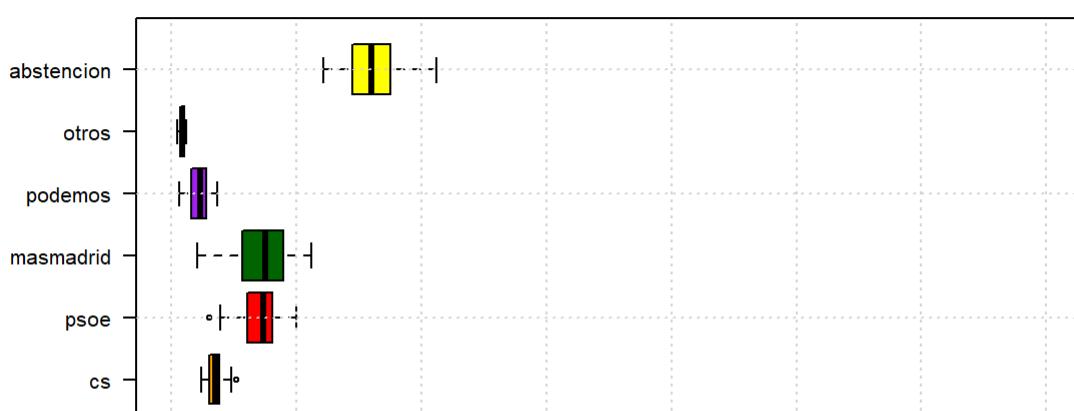
Cluster plot

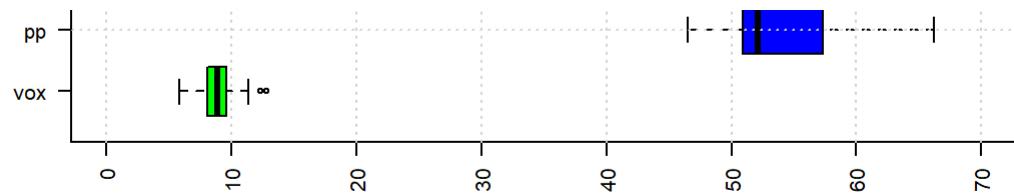


Otra forma de visualizar las diferencias entre los tres clusters es mediante los siguientes boxplots.

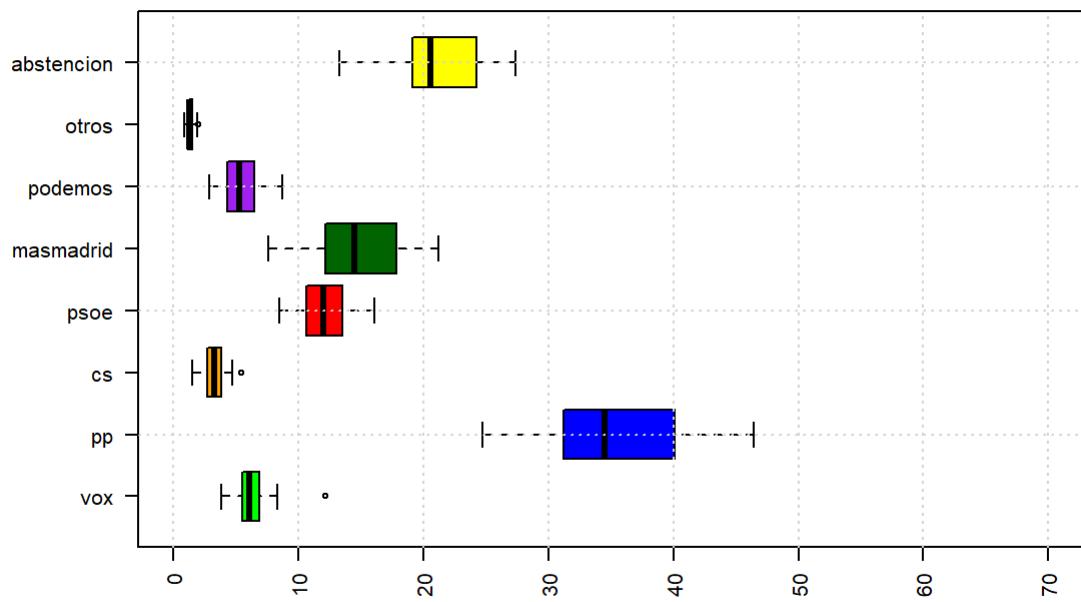
```
par(mfrow=c(3,1))
par(mar=c(5.1, 6.1, 4.1, 2.1))
boxplot(x[c3=="PP_Alto",],horizontal = TRUE,col=colores, las=2,ylim=c(0,70), main="PP ALTO")
grid()
boxplot(x[c3=="PP_Med",],horizontal = TRUE,col=colores, las=2, ylim=c(0,70), main = "PP Medio")
grid()
boxplot(x[c3=="PP_Bajo",],horizontal = TRUE,col=colores, las=2,ylim=c(0,70), main = "PP Bajo")
grid()
```

PP ALTO

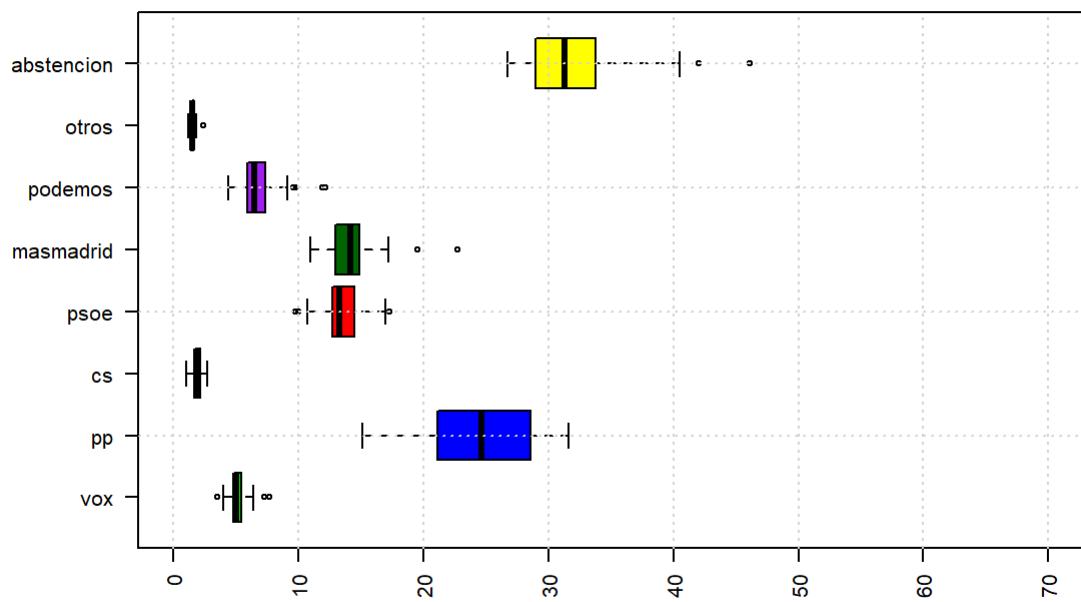




PP Medio



PP Bajo



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

5 Clúster Kmeans

Realiza el análisis clúster utilizando el método kmeans. Haz tres clústers (utiliza nstart=25). Se recomienda utilizar una semilla para inicializar el proceso de cálculo de manera que se puedan repetir las resultados. Proporciona el número de observaciones en cada clúster. (1 punto)

```
set.seed(-1779)
k3 = kmeans(x,centers = 3,nstart = 25)
```

```
names(k3)
```

```
## [1] "cluster"      "centers"       "totss"        "withinss"      "tot.withinss"
## [6] "betweenss"    "size"          "iter"          "ifault"
```

Un resumen del análisis se obtiene poniendo > k3 en R. No lo hacemos aquí para reducir el tamaño del documento.

```
k3$centers
```

```
##           vox      pp      cs      psOE masmadrid   podemos     otros abstencion
## 1 6.091667 34.03958 3.189583 12.362500 15.239583 5.729167 1.4125000 21.92083
## 2 5.060000 24.05111 1.904444 13.680000 14.702222 7.077778 1.5066667 32.02889
## 3 8.557895 51.12895 3.552632 7.910526 8.234211 2.631579 0.9473684 17.04737
```

También se puede obtener con las instrucciones:

```
K=3
mg = NULL
for(k in 1:K){
  mg = rbind(mg,sapply(x[k3$cluster==k,],mean))
}
print(mg,digits = 3)
```

```
##           vox      pp      cs      psOE masmadrid   podemos     otros abstencion
## [1,] 6.09 34.0 3.19 12.36      15.24      5.73 1.413      21.9
## [2,] 5.06 24.1 1.90 13.68      14.70      7.08 1.507      32.0
## [3,] 8.56 51.1 3.55  7.91      8.23      2.63 0.947      17.0
```

Utilizamos el mismo criterio para denominar los clústers. En este caso PP_Alto es el 3, PP_Bajo el 2 y PP_Medio el 1.

```
row.names(mg) = c("PP_Med","PP_Bajo","PP_Alto")
print(mg,digits = 3)
```

```
##           vox      pp      cs      psOE masmadrid   podemos     otros abstencion
## PP_Med 6.09 34.0 3.19 12.36      15.24      5.73 1.413      21.9
## PP_Bajo 5.06 24.1 1.90 13.68      14.70      7.08 1.507      32.0
## PP_Alto 8.56 51.1 3.55  7.91      8.23      2.63 0.947      17.0
```

```
table(k3$cluster)
```

```
##  
##  1  2  3  
## 48 45 38
```

6 Clúster kmeans (desc)

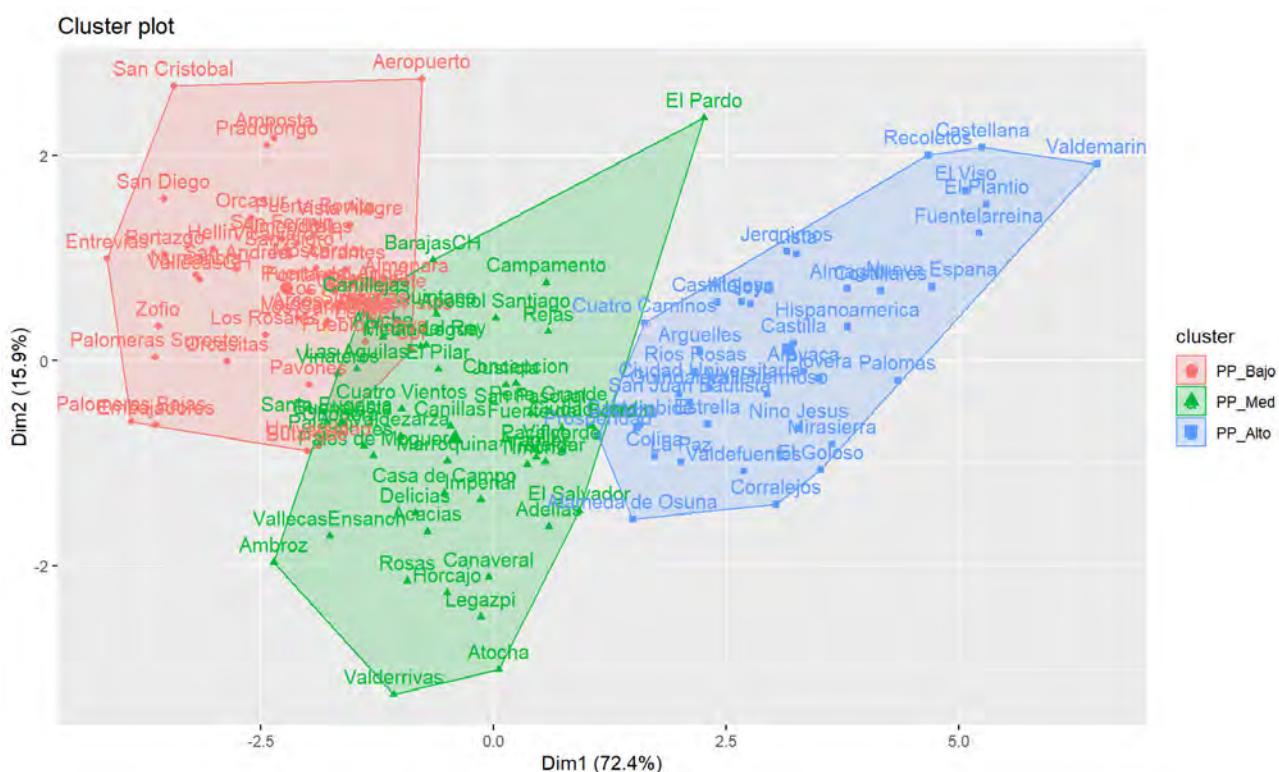
Repite los pasos del apartado 4 y describe la solución obtenida de kmeans. Compara los resultados de los dos métodos: hclust y kmeans. (2 punto)

En la tabla siguiente se comparan las dos soluciones. El grupo de **PP_Alto** de *kmeans* tiene más observaciones que el de *hclust*. Se observa que las 38 observaciones de PP_Alto de *kmeans* estan formadas por los 27 barrios de PP_Alto de *hclust* más 11 barrios que estaban en la frontera de los dos grupos (Argüelles, Cuatro Caminos, Ríos Rosa, por ejemplo). Esto se ve muy bien en el gráfico obtenido utilizando *fviz_cluster()*. En los otros dos grupos los cambios son menos importantes.

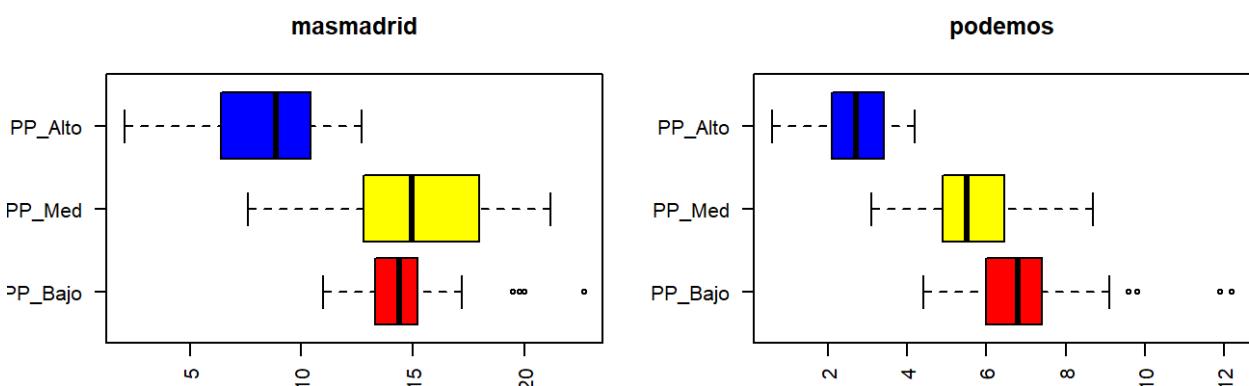
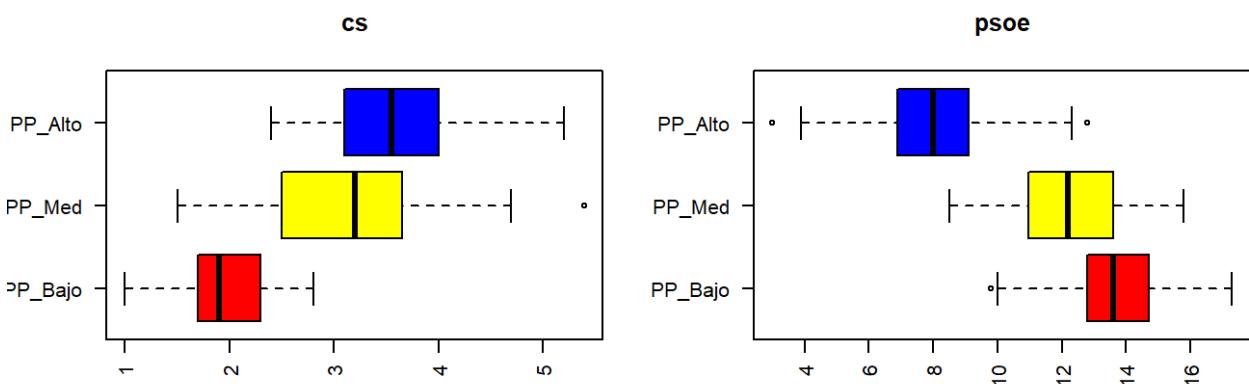
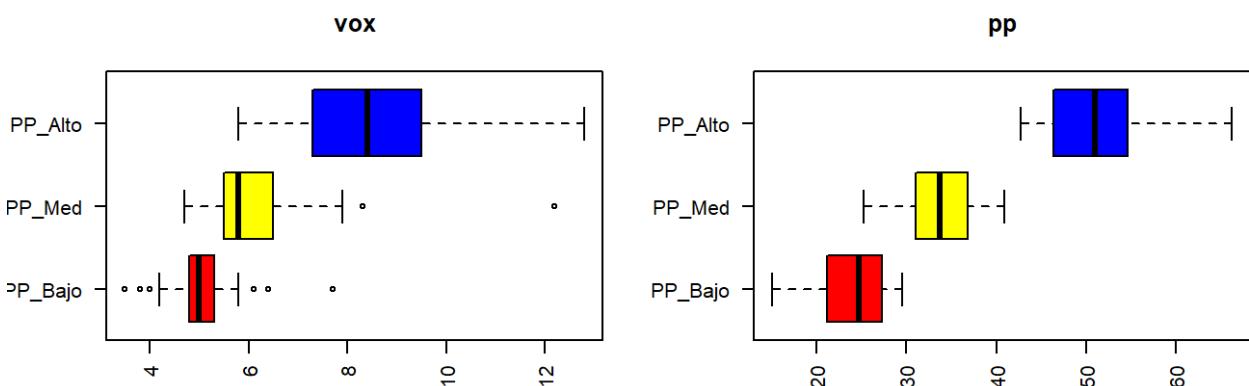
```
d3 = factor(k3$cluster,labels = c("PP_Med","PP_Bajo","PP_Alto"))
d3=relevel(d3,ref="PP_Bajo")
addmargins( table(kmean=d3,hclust=c3))
```

```
##          hclust
## kmean      PP_Bajo PP_Med PP_Alto Sum
## PP_Bajo     41      4      0   45
## PP_Med      3      45      0   48
## PP_Alto     0      11     27   38
## Sum        44      60     27 131
```

```
fviz_cluster(list(clusters=d3,data=x))
```



```
par(mfrow=c(3,2))
titulo = c("vox","pp","cs", "psoe", "masmadrid", "podemos")
for (k in 1:6){
  boxplot(x[,k]~d3, horizontal = TRUE,
    col=c("red","yellow","blue"),
    main = titulo[k],las=2,ylab="",xlab="")
}
```

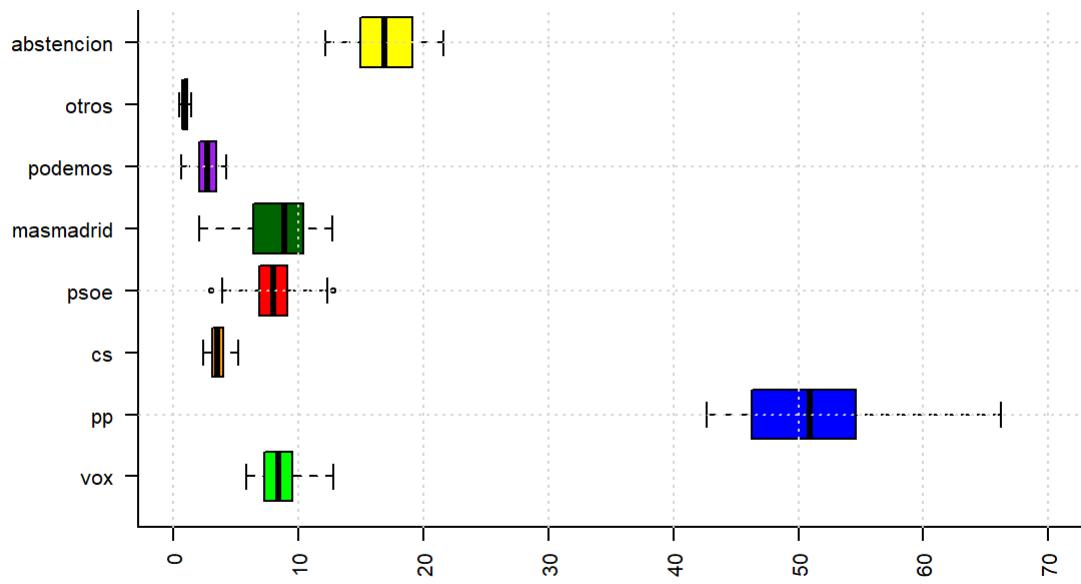


```

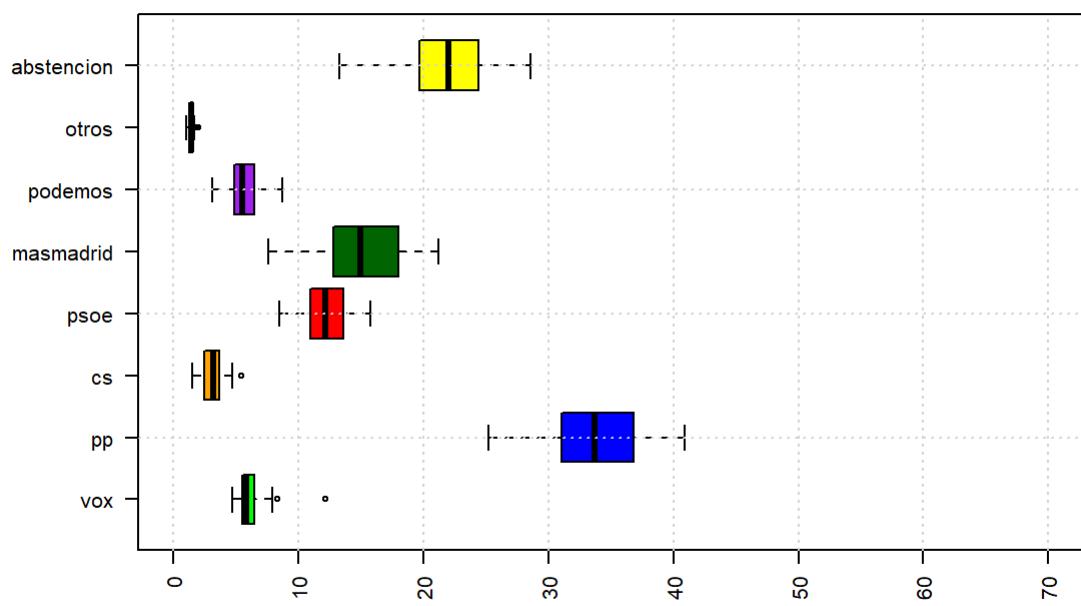
par(mfrow=c(3,1))
par(mar=c(5.1, 6.1, 4.1, 2.1))
boxplot(x[d3=="PP_Alto",],horizontal = TRUE,col=colores, las=2,ylim=c(0,70), main="PP ALTO")
grid()
boxplot(x[d3=="PP_Med",],horizontal = TRUE,col=colores, las=2, ylim=c(0,70), main = "PP Medio")
grid()
boxplot(x[d3=="PP_Bajo",],horizontal = TRUE,col=colores, las=2,ylim=c(0,70), main = "PP Bajo")
grid()

```

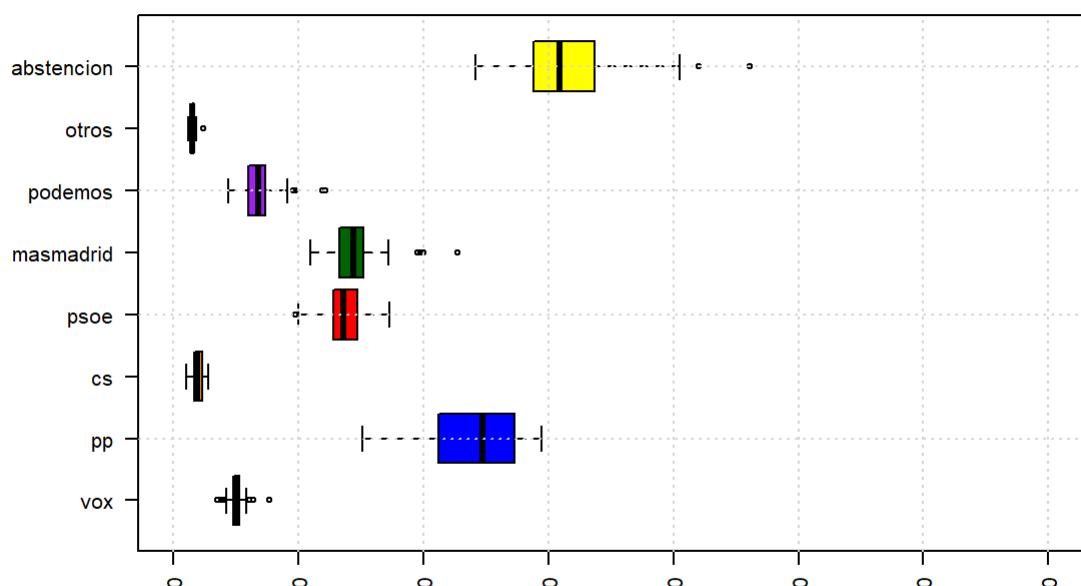
PP ALTO



PP Medio



PP Bajo

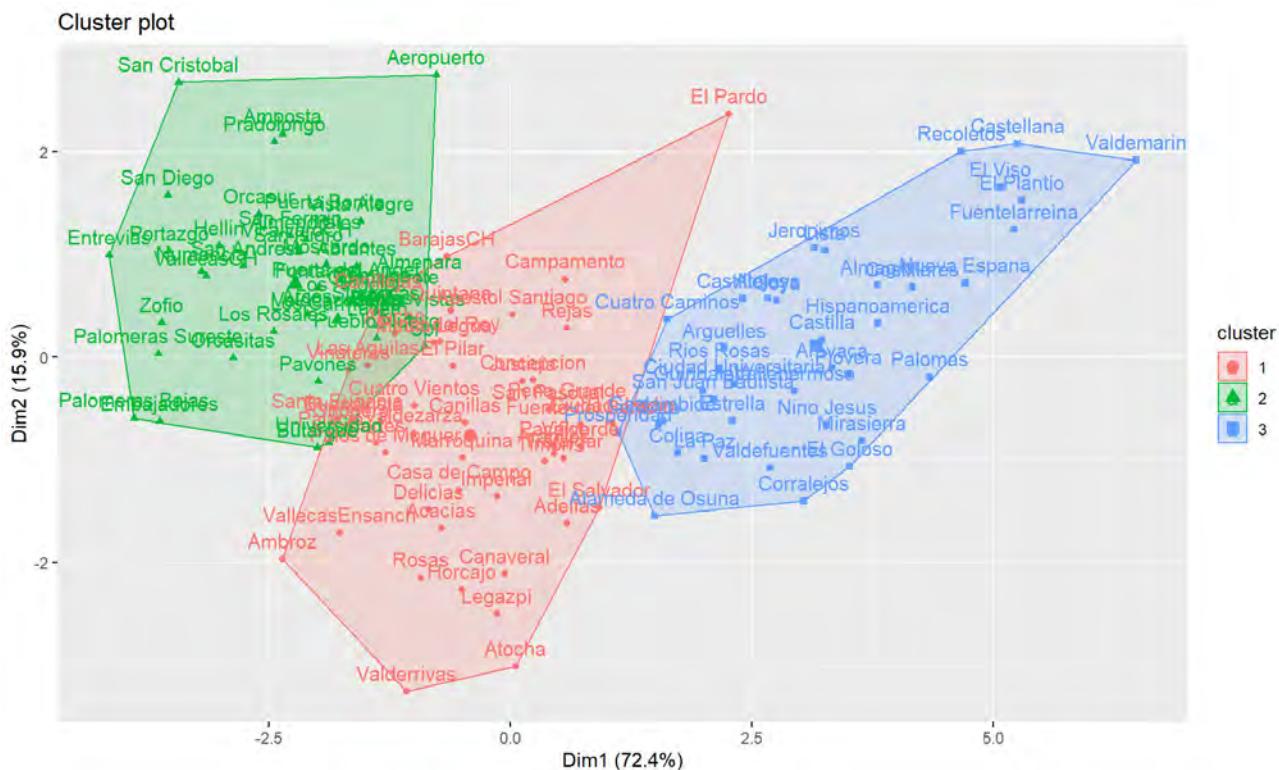


7 Más grupos

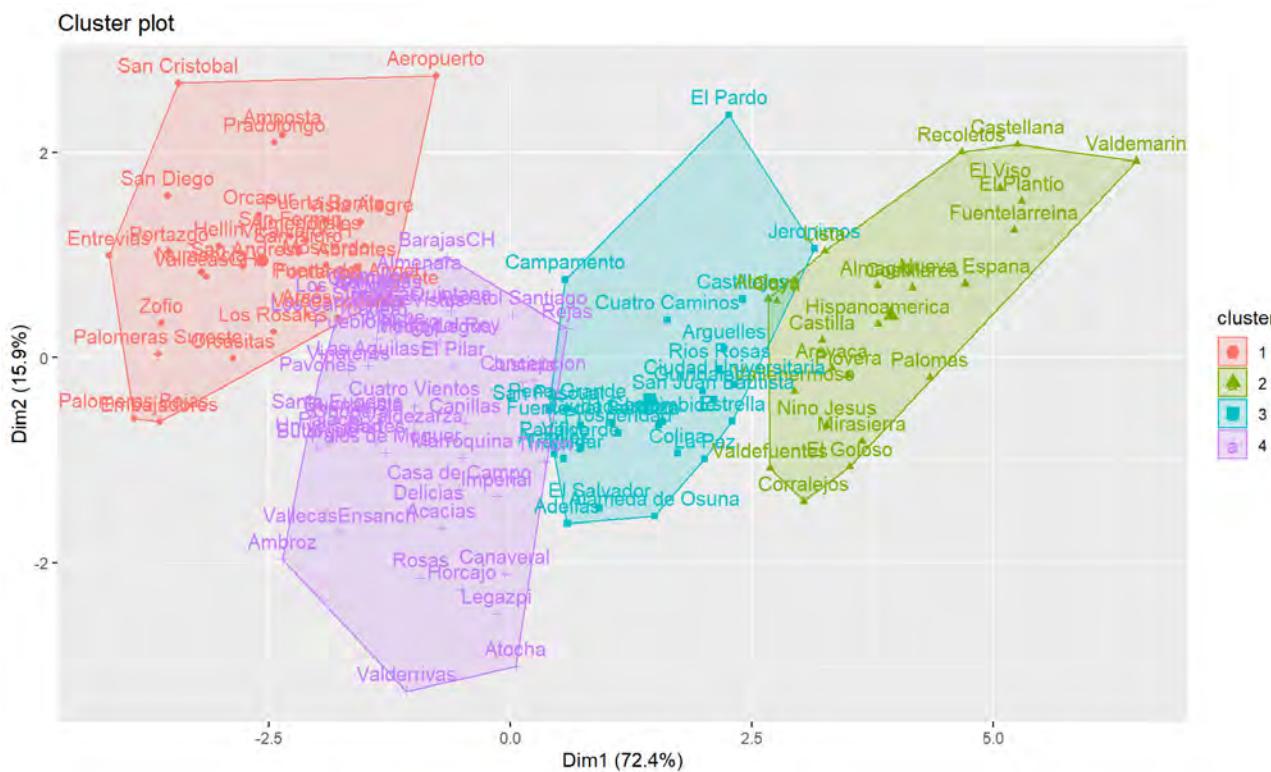
Obtén la solución de kmeans con 3, 4 y 5 grupos. Representa las tres soluciones utilizando la función `fviz_cluster()` e interpreta los resultados. (1 punto)

```
set.seed(-1779)
k3 = kmeans(x,centers = 3,nstart = 25)
k4 = kmeans(x,centers = 4,nstart = 25)
k5 = kmeans(x,centers = 5,nstart = 25)
```

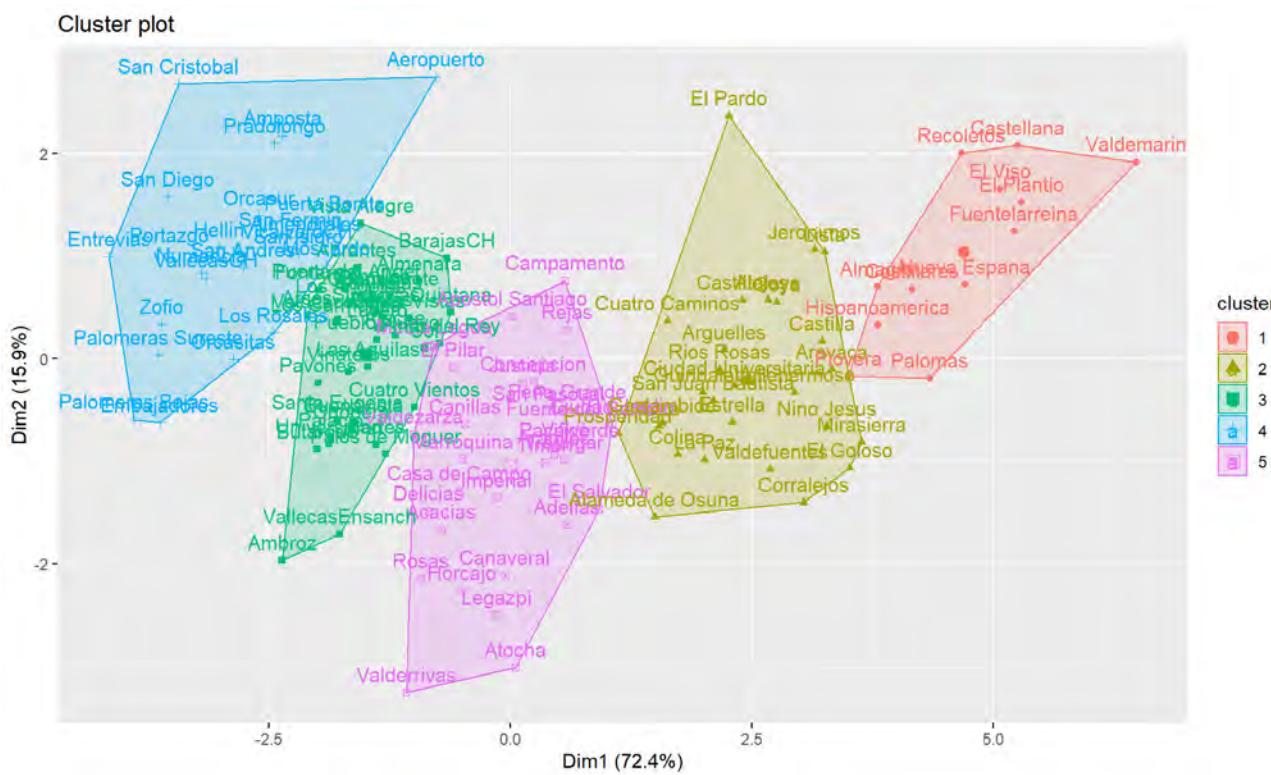
```
par(mfrow=c(3,1))
fviz_cluster(k3,data=x)
```



```
fviz_cluster(k4,data=x)
```

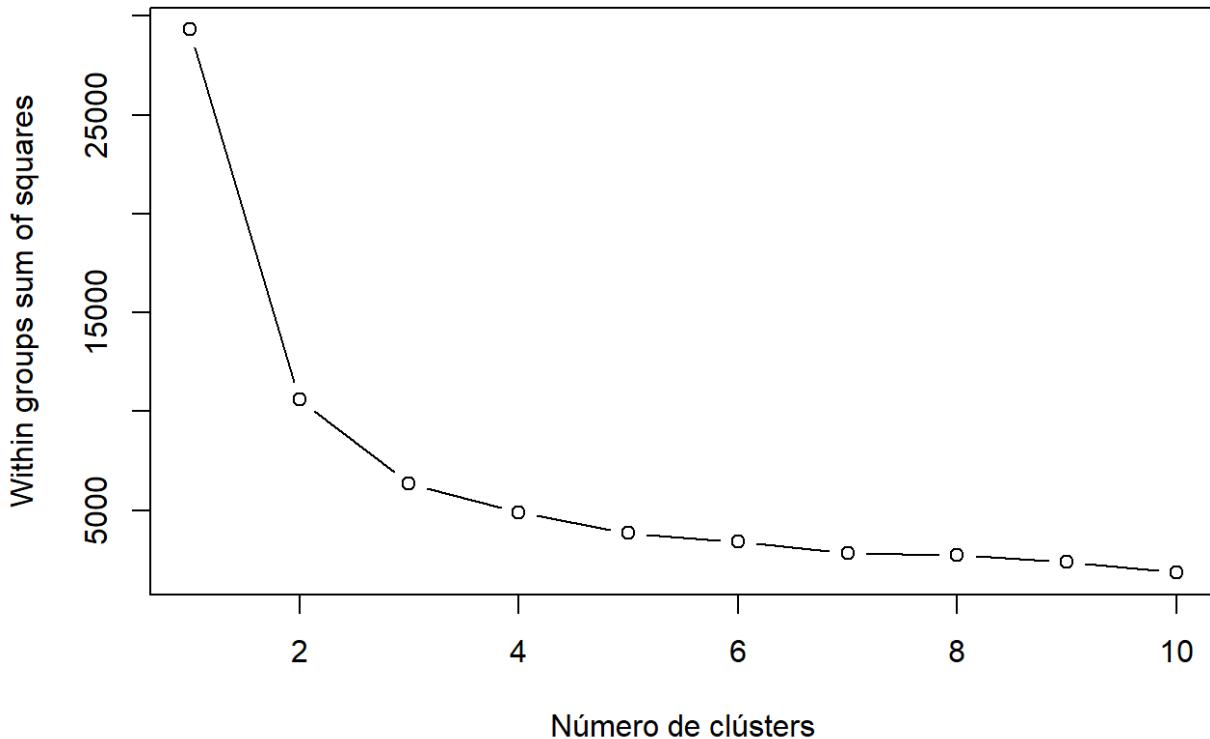


```
fviz_cluster(k5,data=x)
```



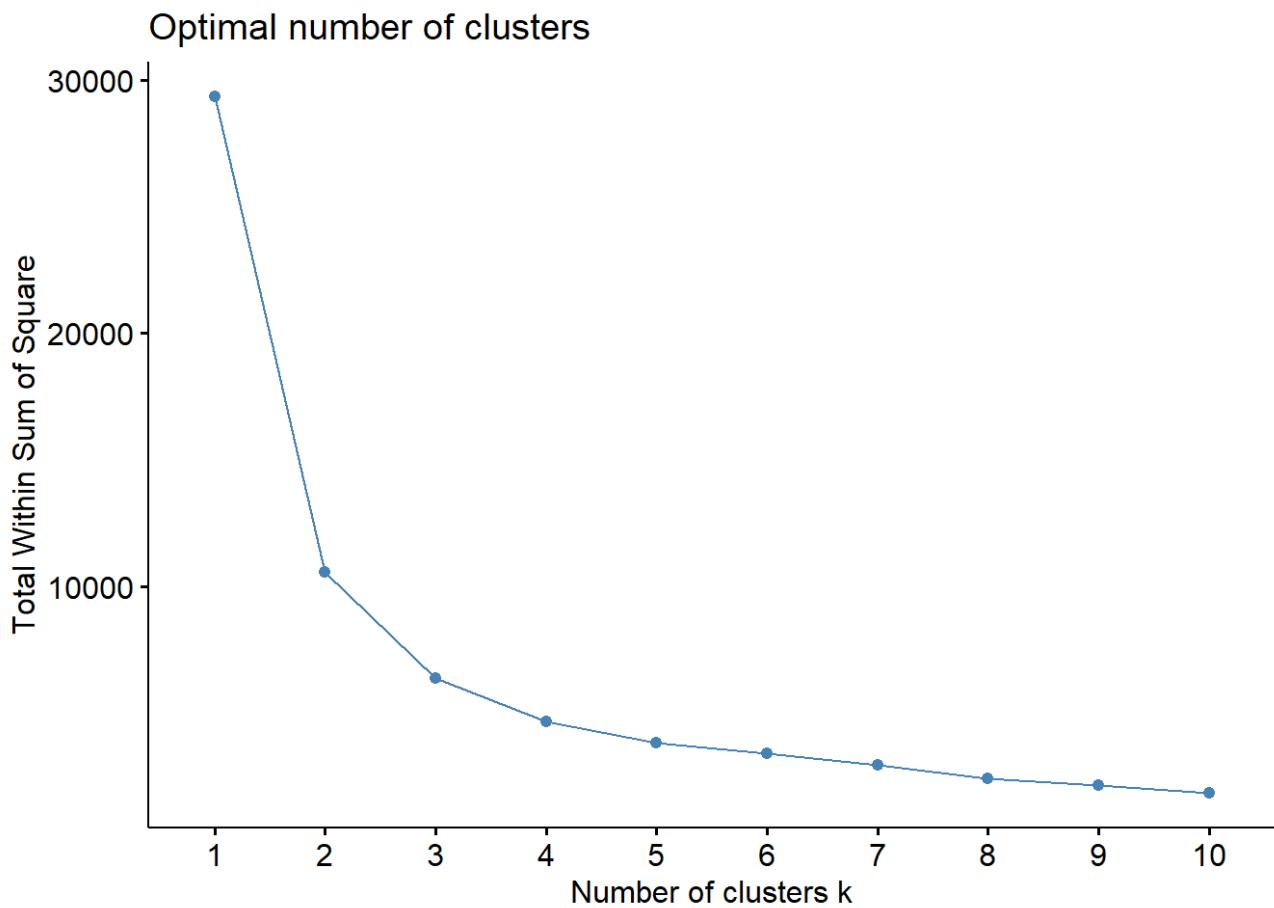
Vemos que conforme aumenta el número de clúster, los grupos tienden a ser más homogéneos, lógicamente. Una pregunta importante es cuando parar, ¿cuál es el número óptimo de clústers? Es una pregunta difícil y existen muchos métodos de seleccionar el número óptimo de clústers. La idea básica de los métodos es calcular la variabilidad interna de los grupos (suma de withinSS) y parar cuando el salto de k grupos a k+1 grupos sea poco importante.

```
wss=0
wss <- (nrow(x)-1)*sum(apply(x,2,var))
for (i in 2:10) wss[i] <- sum(kmeans(x,
                                         centers=i)$withinss)
plot(1:10, wss, type="b", xlab="Número de clústers",
     ylab="Within groups sum of squares")
```



El gráfico se puede obtener directamente con la función `fviz_nbclust()` del paquete `factoextra`.

```
fviz_nbclust(x, kmeans, method = "wss")
```



Según este gráfico el número óptimo de clúster es 3 o 4.

8 Conclusión

Resume brevemente las conclusiones del análisis clúster realizado en los apartados anteriores.

Completa el análisis con aquello que te parezca interesante. (1 punto)

El análisis descriptivo realizado con un boxplot muestra que en Madrid Capital los barrios votan mayoritariamente al PP. En el análisis de correlaciones es interesante comprobar el signo del coeficiente de correlación entre el porcentaje de votos según el partido sea de derecha o izquierda. También destaca la correlación positiva entre votos a candidaturas de izquierda y abstención. Los barrios que votan más a la izquierda son donde se produce una abstención mayor.

Con el análisis clúster realizado no hay evidencia de que existan realmente grupos en la disposición espacial de las observaciones. Se han obtenido varias soluciones con número de clústers 3, 4 y 5. El análisis de las soluciones muestra que el agrupamiento se corresponde con la orientación política derecha o izquierda de los votos. Era de esperar y el análisis lo confirma. En la proyección sobre el plano de componentes principales se observa la disposición de los clúster alineados según el primer componente: A la derecha los barrios correspondientes al grupo PP_Alto, en el medio PP_medio y a la izquierda PP_Bajo.

Una primera impresión al observar los barrios de cada clúster sugiere que existen diferencias socio-económicas importantes entre los grupos. Esto se podría valorar con información extra no disponible en los datos utilizados: nivel medio de renta del barrio, porcentaje de paro, precio medio del metro cuadrado de una vivienda en el barrio, porcentaje de inmigrantes, etc.

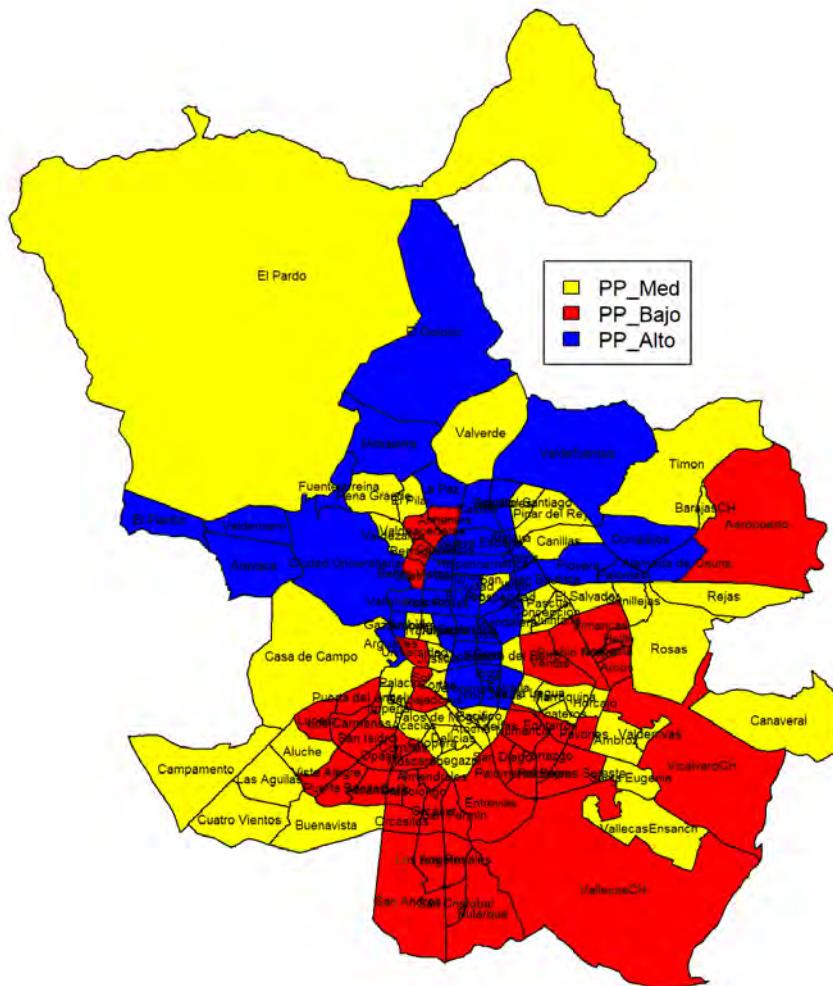
Extra: R permite incorporar mapas con cierta facilidad donde se puede incluir la información resultante del análisis realizado. En la última figura se muestran los barrios de Madrid que han sido coloreados de acuerdo al clúster al que pertenecen. Se observa que los barrios del clúster "PP_Alto" se encuentran en el centro, norte y oeste de Madrid y que el clúster de "PP_Bajo" se encuentra fundamentalmente en el sur.

```
library(sp)
library(maptools)
library(sf)
```

```
madrid=st_read("200001882.shp",quiet=TRUE)

plot(madrid$geometry,col=c("yellow","red","blue")[k3$cluster])

text(st_coordinates(st_centroid(madrid)),row.names(x),cex=.6)
legend(x=444872.4,y= 4489909, cex = 1, legend = c("PP_Med","PP_Bajo","PP_Alto"),
       col= c("yellow","red","blue"), fill = c("yellow","red","blue"))
```



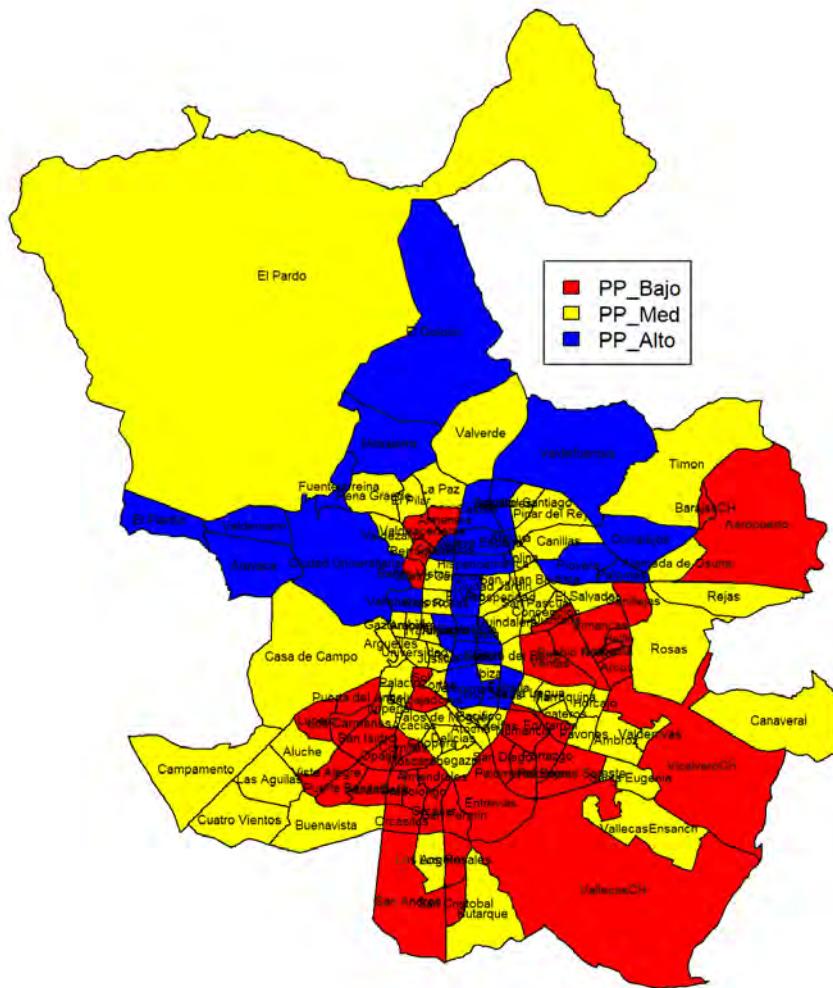
La disposición de los barrios según los clústers obtenidos con el dendrograma se puesta a continuación.

```

plot(madrid$geometry,col=c("red","yellow","blue")[as.numeric(c3)])

text(st_coordinates(st_centroid(madrid)),row.names(x),cex=.6)
legend(x=444872.4,y= 4489909, cex = 1, legend = c("PP_Bajo","PP_Med","PP_Alto"),
       col= c("red","yellow","blue"), fill = c("red","yellow","blue"))

```



El archivo con las delimitaciones de los barrios de Madrid se ha obtenido de <http://www.madrid.org/nomecalles/DescargaBDTCorte.icm>

(<http://www.madrid.org/nomecalles/DescargaBDTCorte.icm>) Hay muchas otras fuentes públicas que ofrecen estos mapas a nivel municipal, autonómico, nacional y mundial.

Wine dataset

Apartado 1.

Apartado 2

Apartado 3

Apartado 4

Apartado 5

Conclusiones del ejercicio

Tarea 3: Wine Análisis Cluster

GITI-Organización Análisis de Datos Cursom 020

Prof. Jesús Juan email: jesus.juan@upm.es (<mailto:jesus.juan@upm.es>)

Wine dataset

El archivo “wine.txt” contiene los resultados del análisis químico de 178 vinos provenientes de una región italiana. Para cada vino se proporciona el contenido de varios compuestos y otras características, en total 13 variables (“Alcohol”, “Malic”, “Ash”, “Alcalinity”, “Magnesium”, “Phenols”, “Flavanoids”, “Nonflavanoids”, “Proanthocyanins”, “Color”, “Hue”, “Dilution” y “Proline”). La primera variable del archivo “Type” es una clasificación previa de los vinos que está basada en criterios subjetivos y que deseamos confirmar con nuestro análisis cluster. Solo será utilizada para evaluar los resultados del análisis cluster realizado.

Apartado 1.

Realizar el dendrograma con las 13 variables continuas utilizando la distancia euclídea y realizando el encadenamiento por el método de ward.D. Indica el número de observaciones que tiene si decidimos formar 3 clusters.

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

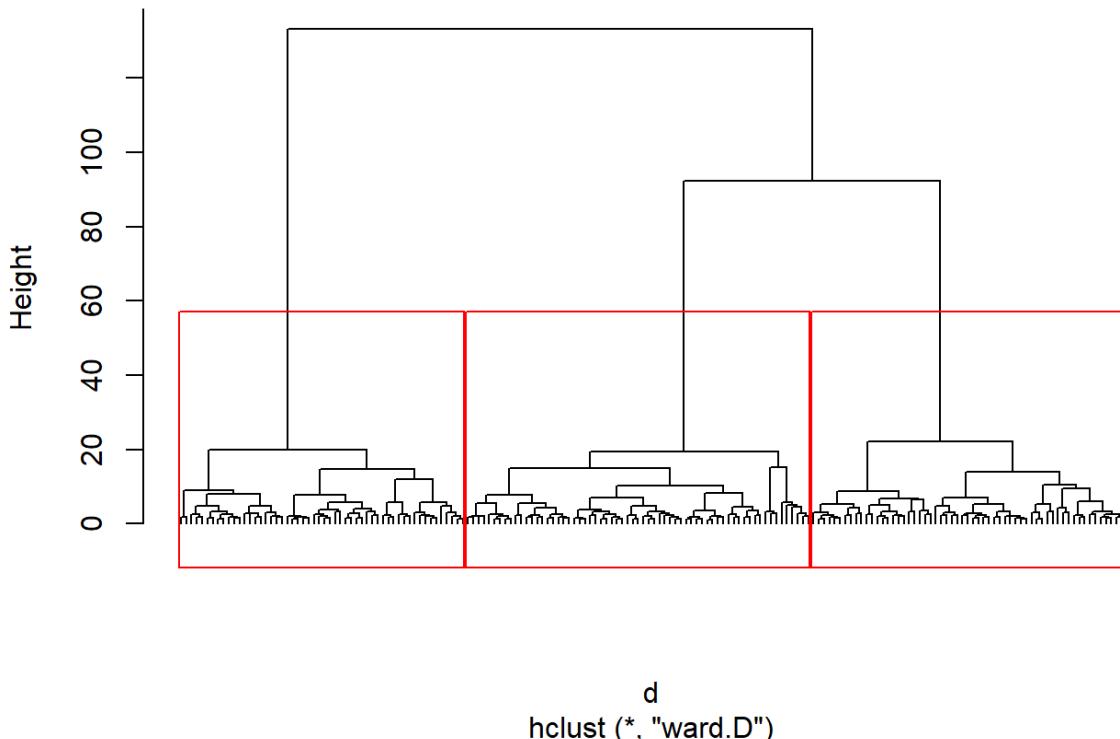
```
## Welcome! Want to learn more? See two factoextra-related books at http  
s://goo.gl/ve3WBa
```

```
library(car)
```

```
## Loading required package: carData
```

```
dat = read.table("wine.txt", header=TRUE)
x = scale(dat[,2:14])
d = dist(x)
fit = hclust(d, method = "ward.D")
plot(fit, hang = -1, labels = FALSE)
K=3
j3 <- cutree(fit, k=K) # elegimos tres grupos
rect.hclust(fit, k=K, border="red")
```

Cluster Dendrogram



En el gráfico no se ven los detalles pero se puede apreciar que hay tres grandes grupos de vinos.

```
table(grupos=j3)
```

```
## grupos
## 1 2 3
## 65 59 54
```

Apartado 2

Calcula la media de las 13 variables para cada grupo (centroide). Muestra gráficamente las diferencias entre los tres grupos para cada una de las variables utilizando gráficos boxplot o similares.

```
cent <- NULL
for (k in 1:K){
  cent <- rbind(cent,sapply(dat[j3==k,2:14],mean))
}
cent
```

```
##          Alcohol      Malic       Ash Alcalinity Magnesium Phenols Flavano
ids
## [1,] 13.62585 1.954462 2.458462 17.61538 108.81538 2.814615 2.9472
308
## [2,] 12.22220 1.966102 2.216271 20.12373 89.89831 2.263051 2.1101
695
## [3,] 13.09852 3.200556 2.420000 21.07037 99.57407 1.704815 0.8359
259
##          Nonflavanoids Proanthocyanins      Color         Hue Dilution Prol
ine
## [1,] 0.2924615 1.930308 5.310000 1.0743077 3.139385 1071.0
769
## [2,] 0.3564407 1.642034 3.004746 1.0523051 2.854746 503.3
898
## [3,] 0.4512963 1.126481 6.998333 0.7131481 1.710926 622.7
222
```

```
cent <- NULL
for(k in 1:K){
  cent <- rbind(cent, colMeans(dat[j3 == k,2:14]))
}
cent
```

```

##          Alcohol      Malic      Ash Alkalinity Magnesium   Phenols Flavano
ids
## [1,] 13.62585 1.954462 2.458462    17.61538 108.81538 2.814615  2.9472
308
## [2,] 12.22220 1.966102 2.216271    20.12373  89.89831 2.263051  2.1101
695
## [3,] 13.09852 3.200556 2.420000    21.07037  99.57407 1.704815  0.8359
259
##          Nonflavanoids Proanthocyanins      Color        Hue Dilution   Prol
ine
## [1,]      0.2924615           1.930308 5.310000 1.0743077 3.139385 1071.0
769
## [2,]      0.3564407           1.642034 3.004746 1.0523051 2.854746  503.3
898
## [3,]      0.4512963           1.126481 6.998333 0.7131481 1.710926  622.7
222

```

Para identificar a los tres grupos los caracterizaremos por su nivel de **Flavanoids**, Alto para grupo 1, Medio para el grupo 2 y Bajo para el grupo 3. Se podría utilizar otro criterio de denominación.

```

j3_names=factor(j3,labels=c("F-Alto","F-Medio","F-Bajo"))
table(Grupos=j3_names)

```

```

## Grupos
## F-Alto F-Medio F-Bajo
##       65      59      54

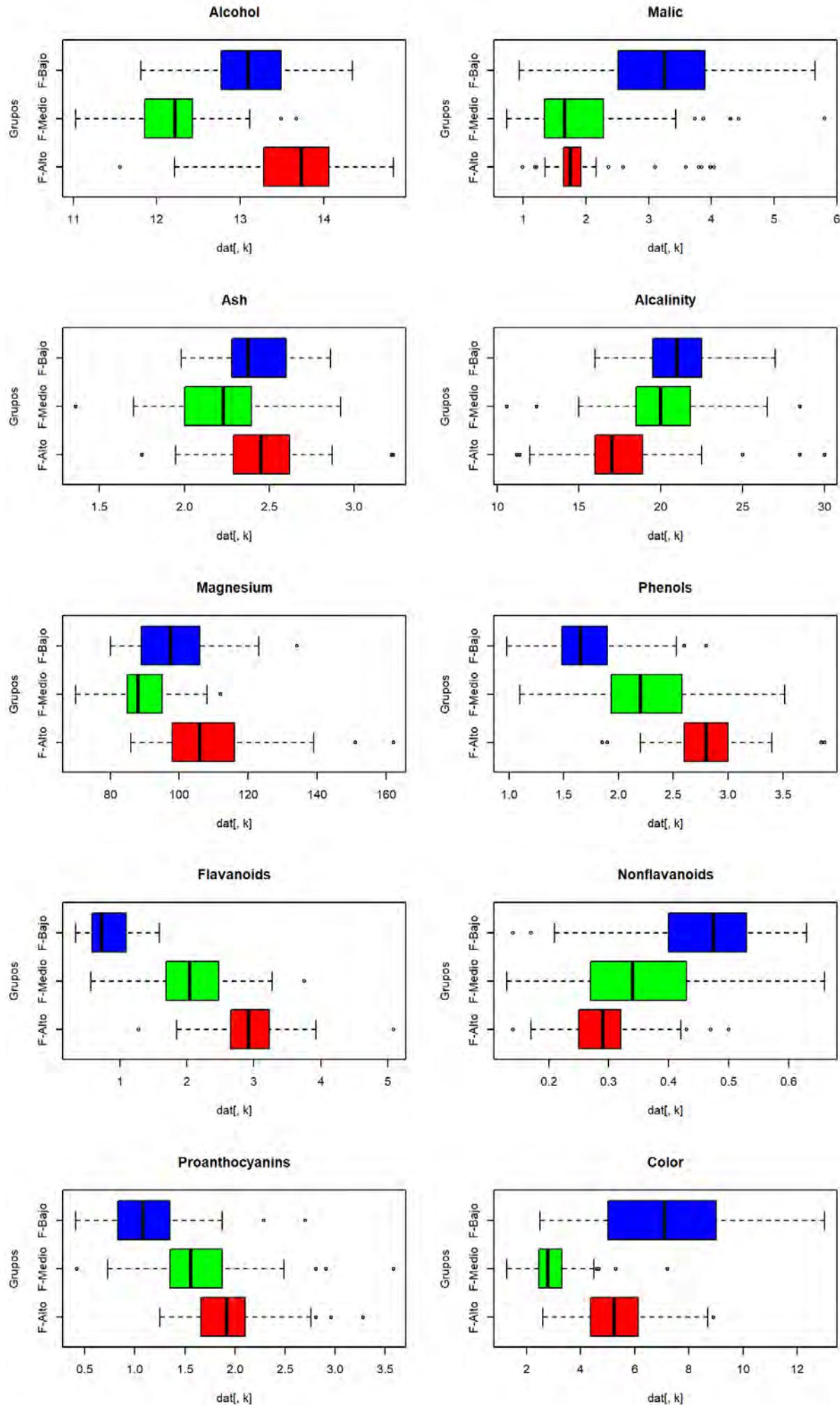
```

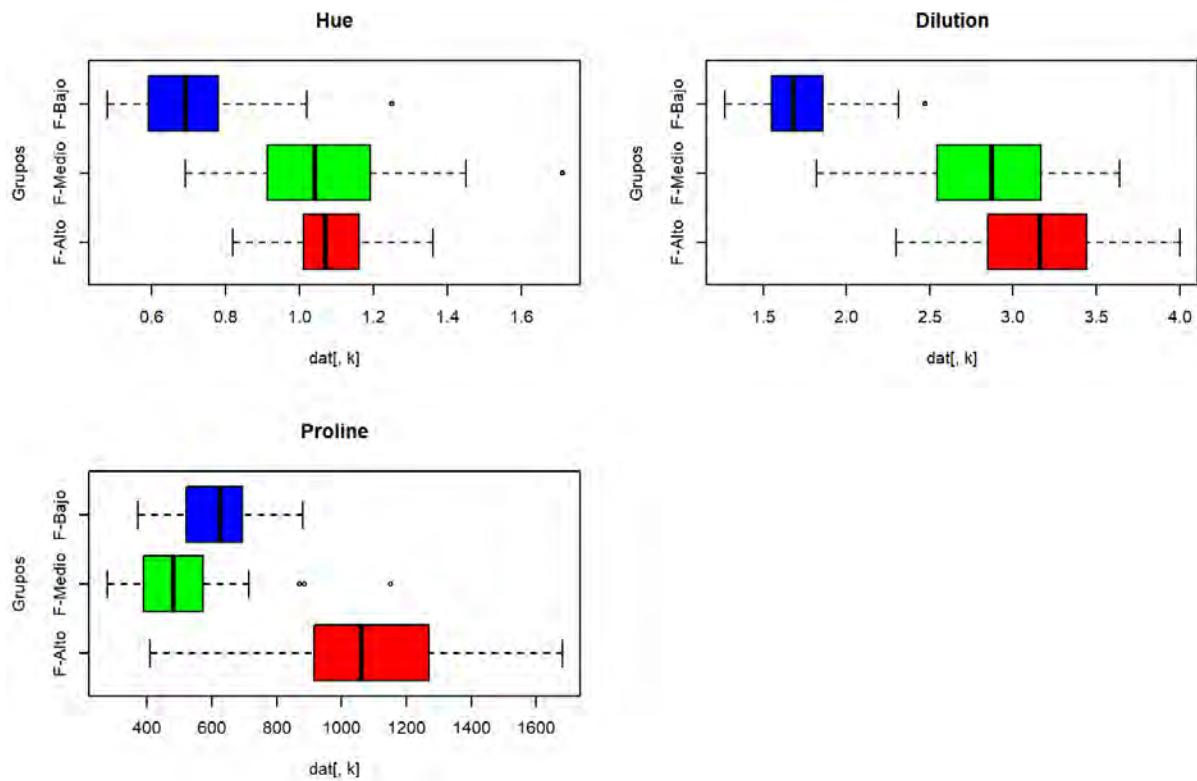
```

par(mfrow=c(3,2))
for (k in 2:14){
  boxplot(dat[,k]~j3_names, horizontal = TRUE,col=rainbow(K),main=names(d
at)[k],ylab="Grupos")

}

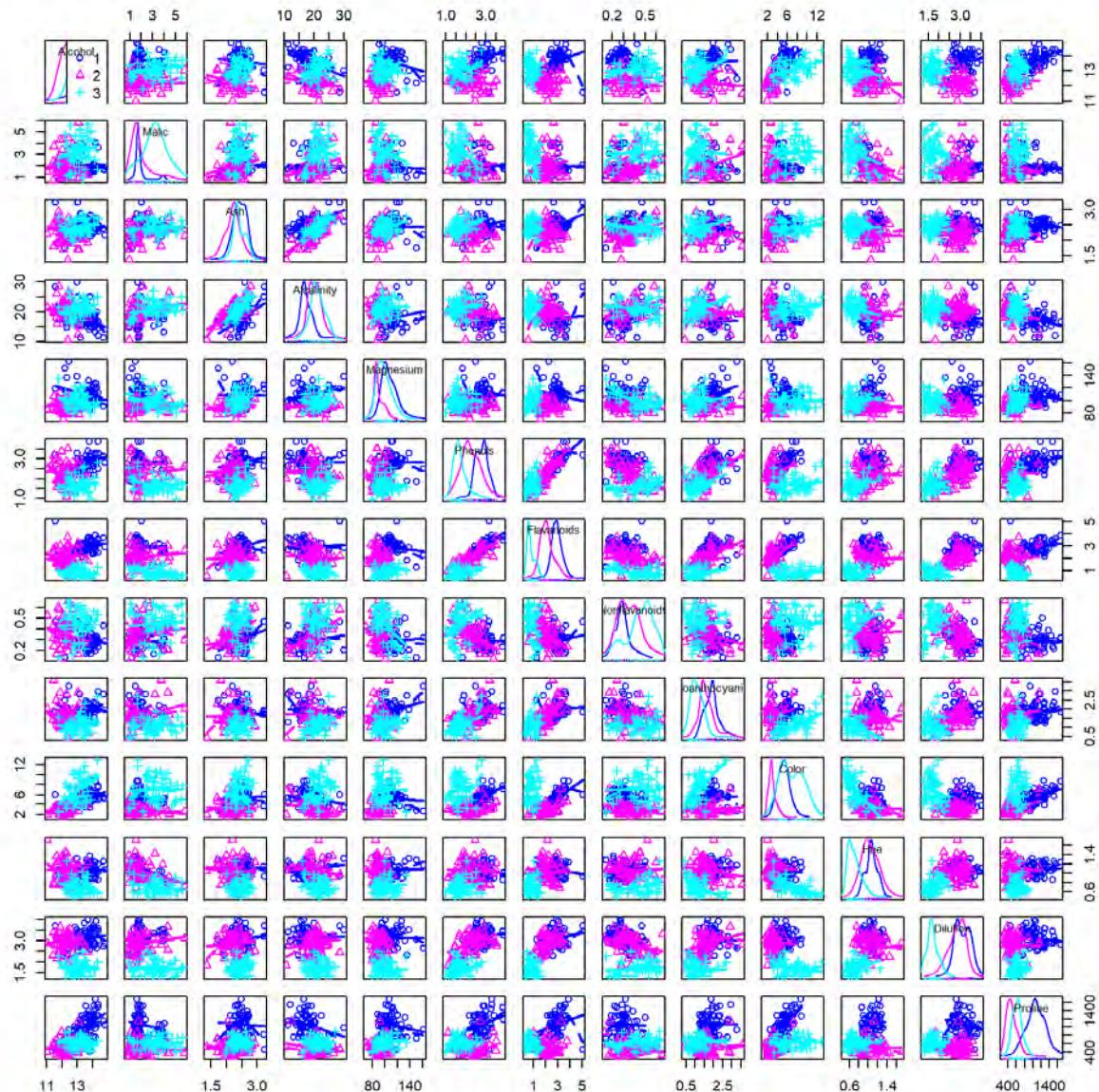
```



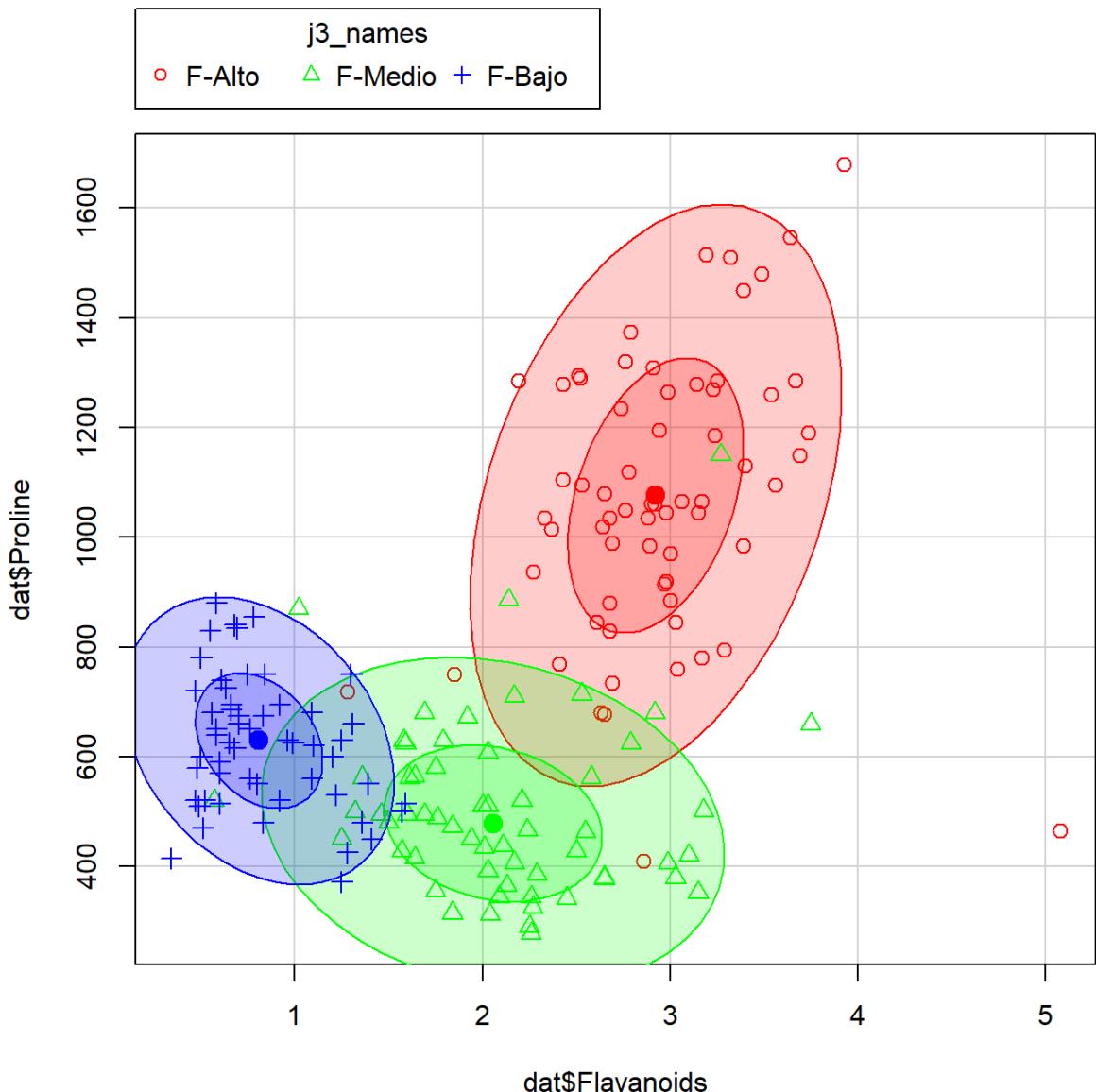
En los gráficos BOXPLOT se observan claramente la diferencia entre el contenido de cada componente. Por ejemplo, el contenido en **Phenols** del grupo 1 (F-Alto) es superior al grupo 2, F-Medio y los del grupo 2, también superiores al grupo 3 (F-Bajo). Viendo los gráficos uno a uno se puede valorar si existen diferencias entre los tres grupos para esa variable. En la mayoría de los casos se encuentran diferencias, no siempre con la misma ordenación. Por ejemplo el contenido en Alcohol es menor en el grupo “F-Medio”.

```
scatterplotMatrix(dat[,2:14],groups=j3)
```



Los gráficos de dispersión 2 a 2, también es una manera de visualizar las diferencias entre los tres grupos. Si miramos en la diagonal del gráfico, se muestra una estimación “no-paramétricas” (suave) del histograma de cada componente para cada grupo. Se aprecia fácilmente la diferencia entre los grupos. También se puede comprobar en los gráficos de dispersión. Por ejemplo en el gráfico de **Flavanoids** versus **Proline** se aprecia claramente que los grupos de vinos tienen diferencias importantes.

```
scatterplot(dat$Flavanoids,dat$Proline,
           groups = j3_names,
           regLine = F, smooth = F,
           ellipse = T,
           col=c("red","green","blue"), cex = 1.2)
```



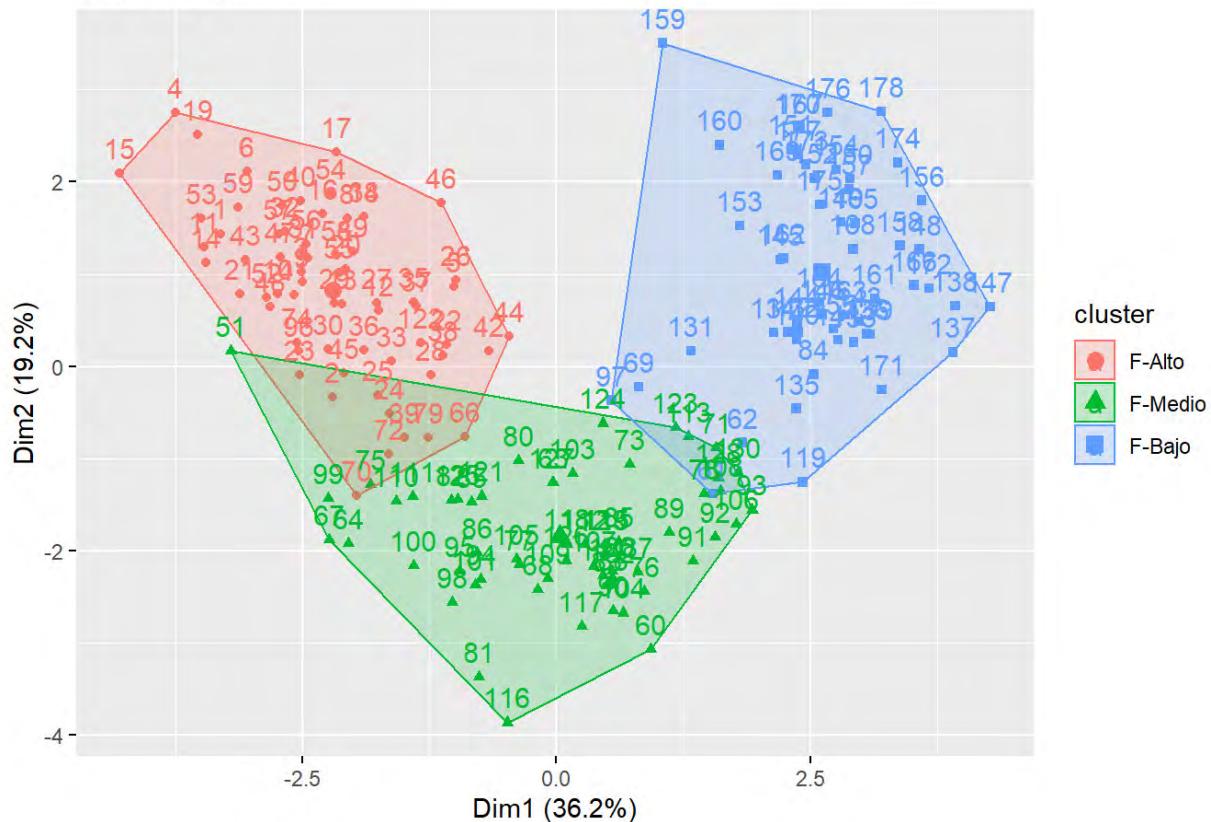
En el gráfico se aprecia que hay tres regiones bastante bien delimitadas para cada grupo con dos variables. Los puntos rojos (1-F-Alto) quedan identificado por un contenido grande de Proline. El grupo azul (3-F-Bajo) tiene bajo contenido en Proline y en Flavanoids. El grupo verde (2-F-Medio) tiene bajo contenido en Proline y alto en Flavanoids. Hay solapamiento, pero hay más variables que ayudan a diferenciar entre los tres grupos.

Apartado 3

Utiliza la función `fviz_cluster()` (del package `factoextra`) para mostrar en un gráfico de dimensión 2 la solución de tres grupos obtenida en el apartado 1. Utiliza la función `prinfact()` que estudiamos en el capítulo de componentes principales para **describir e interpretar** la gráfica (tened en cuenta que la proyección observada en el gráfico puede aparecer invertida respecto a los valores numéricos)

```
fviz_cluster(list(cluster=j3_names,data=dat[,2:14]))
```

Cluster plot



```
source('prinfect.R')
sol = prinfect(dat[,2:14],2)
sol$loadings
```

	Comp 1	Comp 2	communality	uniqueness
## Alcohol	0.313093350	0.764257253	0.6821166	0.3178834
## Malic	-0.531884726	0.355431713	0.4092331	0.5907669
## Ash	-0.004449362	0.499446109	0.2494662	0.7505338
## Alcalinity	-0.519157081	-0.016734916	0.2698041	0.7301959
## Magnesium	0.308022936	0.473476124	0.3190578	0.6809422
## Phenols	0.856136658	0.102774237	0.7435325	0.2564675
## Flavanoids	0.917470177	-0.005309113	0.8417797	0.1582203
## Nonflavanoids	-0.647607018	0.045476816	0.4214630	0.5785370
## Proanthocyanins	0.679921705	0.062103856	0.4661504	0.5338496
## Color	-0.192235968	0.837489383	0.7383431	0.2616569
## Hue	0.643662066	-0.441242229	0.6089956	0.3910044
## Dilution	0.816018903	-0.259933849	0.7334525	0.2665475
## Proline	0.622050797	0.576612723	0.7194294	0.2805706

```
sol$variances
```

```

##          Comp 1     Comp 2
## Variance   4.7058503 2.4969737
## Proportion 0.3619885 0.1920749
## Cumulative 0.3619885 0.5540634

```

- El componente 1 está determinado fundamentalmente por las variables **Phenols**(+), **Flavanoids**(+) y **Dilution**(+) y con menor peso **Proanthocyanins**(+), **Hue**(+), **Nonflavanoids**(-), **Proline**(+), **Malic**(-) y **Alcalinity**(-). La diferencia entre el Cluster 1 (F-Alto) y el 3 (F-Bajo), está marcado por este componente (estas variables). El grupo rojo (cluster F-Alto) tiene valores altos de las variables con peso (+) y bajo en las variables con peso (-)- El grupo azul (cluster 3) al revés. Si nos fijamos en esas variables, encontraremos diferencias entre los dos grupos . Cuidado porque en la proyección realizada el primer componente tiene los signos cambiados.
- En el componente 1, el grupo 2 (verde-“F-Medio”) se situa en medio. Vemos que en general en las variables anteriores tienen puntuaciones intermedias.
- El componente 2 está determinado por el contenido en **Alcohol**, **Color**, **Ash** y **Magnesium**. El grupo 1 (F-Alto) y 3 (F-Bajo) tienen altos valores de estas cuatro variables y el grupo 2 (F-Medio), destaca por bajo contenido en todas ellas, especialmente en **Alcohol**.
- Se ha hecho un resumen aproximado, en realidad cada componente es una combinación de todas las variables y todas afectan a la posición en el gráfico.
- Es importante destacar, que esta proyección (componentes principales) tiene como objetivo mostrar el plano donde hay una mayor dispersión de los datos. La proyección no necesariamente tiene por qué coincidir con el plano donde haya una mayor diferencias entre los tres grupos. En este caso, los dos componente, solo explican el 55% de la variabilidad total. Es decir que es una información muy incompleta del conjunto de las variables.

Apartado 4

Utilizando el procedimiento kmeans (con datos estandarizados) obtén la solución de tres grupos. Proyecta la solución en dimensión dos (utilizando la instrucción `fviz_cluster()`) y compara los resultados con la solución jerárquica de los apartados anteriores (ten en cuenta que el número asignado a cada cluster en cada solución es arbitrario)

```

set.seed(100)
k3 = kmeans(x,centers = 3,nstart = 25,iter.max = 25)
k3

```

```

## K-means clustering with 3 clusters of sizes 65, 62, 51
##
## Cluster means:
##      Alcohol      Malic      Ash Alcalinity   Magnesium   Phenols
## 1 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891
## 2  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724
## 3  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548
##      Flavanoids Nonflavanoids Proanthocyanins      Color       Hue     D
## 1  0.02075402    -0.03343924        0.05810161 -0.8993770  0.4605046  0.
## 2  0.97506900    -0.56050853        0.57865427  0.1705823  0.4726504  0.
## 3 -1.21182921     0.72402116       -0.77751312  0.9388902 -1.1615122 -1.
## 2887761
##      Proline
## 1 -0.7517257
## 2  1.1220202
## 3 -0.4059428
##
## Clustering vector:
## [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## 2 2 2 2 2
## [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1
## 1 1 1 1 2
## [75] 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 3 1 1 2 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## 3 3 3 3 3
## [149] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## 3
## Within cluster sum of squares by cluster:
## [1] 558.6971 385.6983 326.3537
## (between_SS / total_SS =  44.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"       "totss"         "withinss"      "tot.
## withinss"
## [6] "betweenss"    "size"          "iter"          "ifault"

```

Importante realizar el análisis kmeans utilizando nstart > 20, para obtener una buena solución. Aún así, la solución del método de kmeans puede (si los clusters no están bien definidos) cambiar ligeramente cada vez que repitamos el cálculo. También es importante tener en cuenta que el número asignado a cada cluster es arbitrario y puede cambiar con cada análisis. Es útil en ese caso inicializar la simulación estableciendo la

semilla que genera los números aleatorios `set.seed()`, si mantenemos este valor siempre nos dará la misma solución (es útil especialmente en un documento tipo Rmarkdown, para que los comentarios coincidan con los resultados mostrados).

```
fviz_cluster(k3,dat[,2:14])
```



El gráfico muestra que las soluciones del cluster jerárquico y kmeans son muy parecidas. En el cluster 1 de kmeans ha colocado a los “F-Medio” y en el cluster 2 de kmeans a los “F-Alto”. (Nota.- Esto puede ser diferente en cada ejecución)

```
table(Grupos_J = j3_names, Grupos_K=k3$cluster)
```

```
##           Grupos_K
## Grupos_J   1   2   3
##   F-Alto    4   61   0
##   F-Medio   58   1   0
##   F-Bajo    3   0   51
```

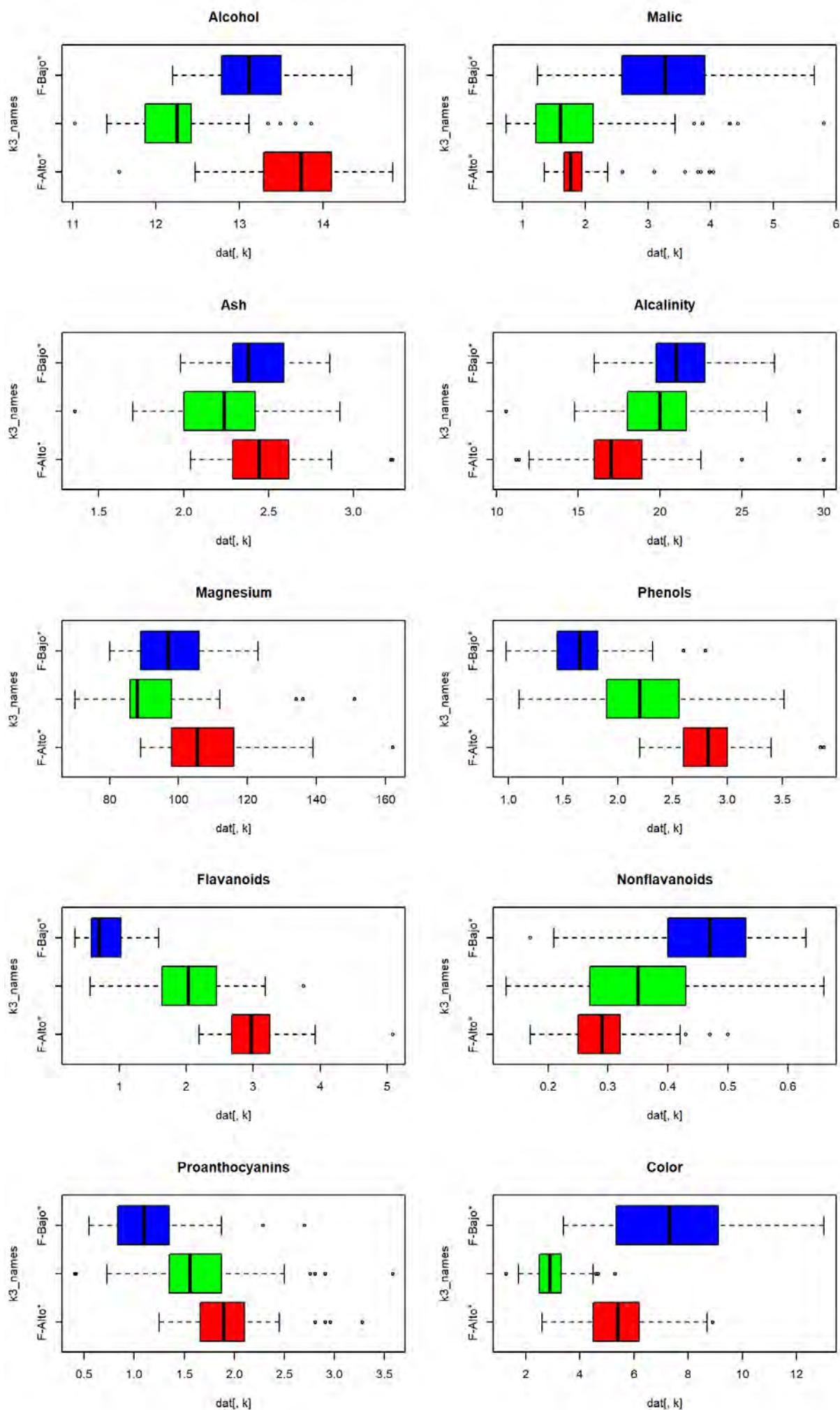
Renombrando la solución de **kmeans** de acuerdo con el mismo criterio utilizado en el cluster jerárquico y comparando, se tiene:

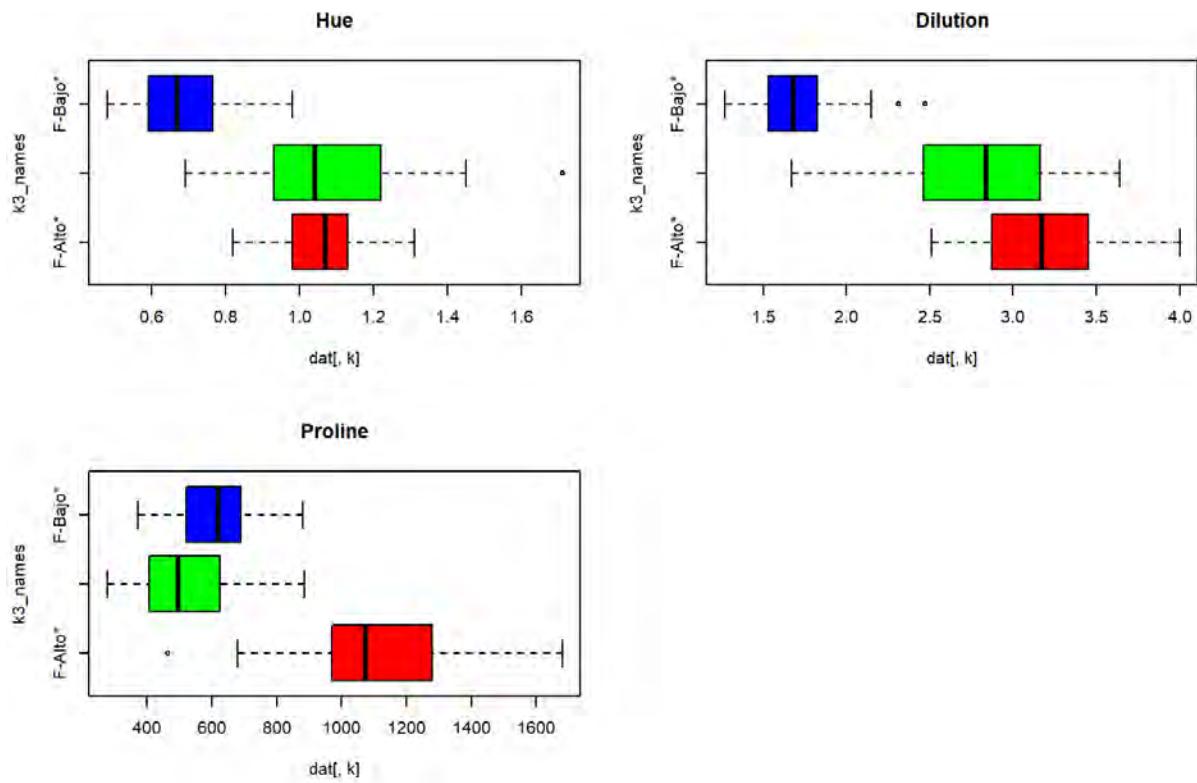
```
k3_names = factor(k3$cluster, labels = c("F-Medio*","F-Alto*","F-Bajo*"))
k3_names = relevel(k3_names, ref="F-Alto*")
table(Grupos_J=j3_names, Grupos_K=k3_names)
```

```
##          Grupos_K
## Grupos_J  F-Alto* F-Medio* F-Bajo*
##   F-Alto      61       4       0
##   F-Medio      1      58       0
##   F-Bajo      0       3      51
```

Los gráficos `boxplot` son similares.

```
par(mfrow=c(3,2))
for (k in 2:14){
  boxplot(dat[,k]~k3_names, horizontal = TRUE, col=rainbow(K), main=names(dat)[k])
}
```



Vamos a identificar los vinos con diferente clasificación en k3 y j3.

```
(sel32 = which(k3_names == "F-Alto*" & j3_names == "F-Medio"))
```

```
## [1] 51
```

```
(sel21 = which(k3_names == "F-Medio*" & j3_names == "F-Alto"))
```

```
## [1] 66 70 72 79
```

```
(sel23 = which(k3_names == "F-Medio*" & j3_names == "F-Bajo"))
```

```
## [1] 61 69 97
```

```
dat[c(sel32, sel21, sel23), c(1, 2, 7, 8, 11, 14)]
```

```
##      Type Alcohol Phenols Flavanoids Color Proline
## 51      1    13.05     2.72      3.27   7.20    1150
## 66      2    12.37     2.42      2.65   4.60     678
## 70      2    12.21     1.85      1.28   2.85     718
## 72      2    13.86     2.95      2.86   3.38     410
## 79      2    12.33     1.90      1.85   3.40     750
## 61      2    12.33     2.05      1.09   3.27     680
## 69      2    13.34     2.53      1.30   3.17     750
## 97      2    11.81     1.60      0.99   2.50     625
```

En el apartado siguiente se comenta someramente a qué se deben los errores de clasificación.

Apartado 5

Compara la solución de kmeans con la clasificación inicial proporcionada en la variable Type. Explica las diferencias. Identifica las observaciones que están mal clasificadas.

```
table(dat$Type, k3_names)
```

```
##      k3_names
##      F-Alto* F-Medio* F-Bajo*
## 1      59       0       0
## 2       3       65       3
## 3       0       0      48
```

El método **kmeans** coincide básicamente con la clasificación previa establecida en la variable Type. El tipo 1 corresponde con “F-Alto”, el tipo 2 con “F-Medio” y el tipo 3 con “F-Bajo”. Hay tres observaciones que kmeans asigna al grupo “F-Alto” que corresponden al tipo 2 y otras tres que kmeans asigna al grupo “F-Bajo” y que corresponden también al tipo 2. Si consideramos los tres grupos de manera secuencial, los errores se han producido en vinos del tipo 2, grupo intermedio, que unas veces los ha metido en un extremo y otros tres que ha metido en el otro extremo.

Son tipo 2 y están clasificadas como "F-Alto*"

```
(obs1 = which(dat$type == 2 & k3_names == "F-Alto*"))
```

```
## [1] 74 96 122
```

```
dat[obs1, c(1, 2, 7, 8, 11, 14)] # Elijo un reducido número de variables para  
reducir
```

```
##      Type Alcohol Phenols Flavanoids Color Proline  
## 74      2    12.99     3.30      2.89  3.35    985  
## 96      2    12.47     2.50      2.27  2.60    937  
## 122     2    11.56     3.18      5.08  6.00    465
```

Son tipo 2 y están clasificadas como "F-Bajo*" _

```
# obs = which(dat$type == 2 & (k3_names == "F-Alto*" | k3_names == "F-Bajo*"))  
(obs2 = which(dat$type == 2 & k3_names == "F-Bajo*"))
```

```
## [1] 62 84 119
```

```
dat[obs2, c(1, 2, 7, 8, 11, 14)]
```

```
##      Type Alcohol Phenols Flavanoids Color Proline  
## 62      2    12.64     2.02      1.41  5.75    450  
## 84      2    13.05     1.65      1.59  4.80    515  
## 119     2    12.77     1.63      1.25  3.40    372
```

Comparando los valores anteriores con el centro de los clusters, se puede explicar por qué estas observaciones están clasificadas en los clusters 1 y 3 en lugar de en el 2.

Conclusiones del ejercicio

Aplicando los dos métodos cluster (jerárquico y kmeans) se consiguen soluciones semejantes con muy pocas variaciones. Los resultados de las clasificaciones coinciden también con la clasificación previa de los vinos en tres tipos (declarados en la variable Type). Según esto, el método cluster es una buena técnica para identificar el origen de un vino.

ANALISIS DE DATOS

Lección 6

CART Y RANDOM FOREST PARA REGRESIÓN

1

Método basado en árboles

Se desea predecir :

- el **peso de una persona** y_i (variable dependiente) en función de
- **perímetro de su cintura** x_{1i} (regresor 1)
- **perímetro de su pecho.** x_{2i} (regresor 2)

El modelo de regresión lineal :

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + u_i$$

```

m1 = lm(Peso ~ C_Cintura + C_Pecho,
        data = hombre)

```

```
hombre = dat[dat$Sexo=="Hombre",]
```

```
summary(m1)
```

```

##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7614  -3.2976  -0.2276   3.3808  14.8801
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.89830  4.76320 -7.747 2.53e-13 ***
## C_Cintura    0.58198  0.05507 10.568 < 2e-16 ***
## C_Pecho      0.65201  0.06709  9.719 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.316 on 244 degrees of freedom
## Multiple R-squared:  0.7464, Adjusted R-squared:  0.7443
## F-statistic: 359.1 on 2 and 244 DF,  p-value: < 2.2e-16

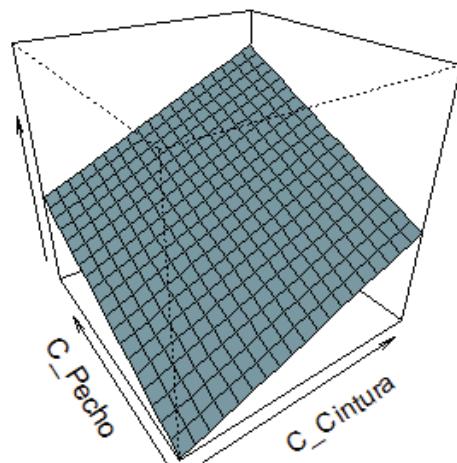
```

$$y_i = -36.9 + 0.58x_1 + 0.65x_2 + e_i,$$

$$s_R = 5.3 \text{ kg}, R^2 = 74.6\%$$

Regresión Lineal

C_Cintura: C_Pecho



$$y_i = -36.9 + 0.58x_1 + 0.65x_2 + e_i,$$

$$s_R = 5.3 \text{ kg}, R^2 = 74.6\%$$

```

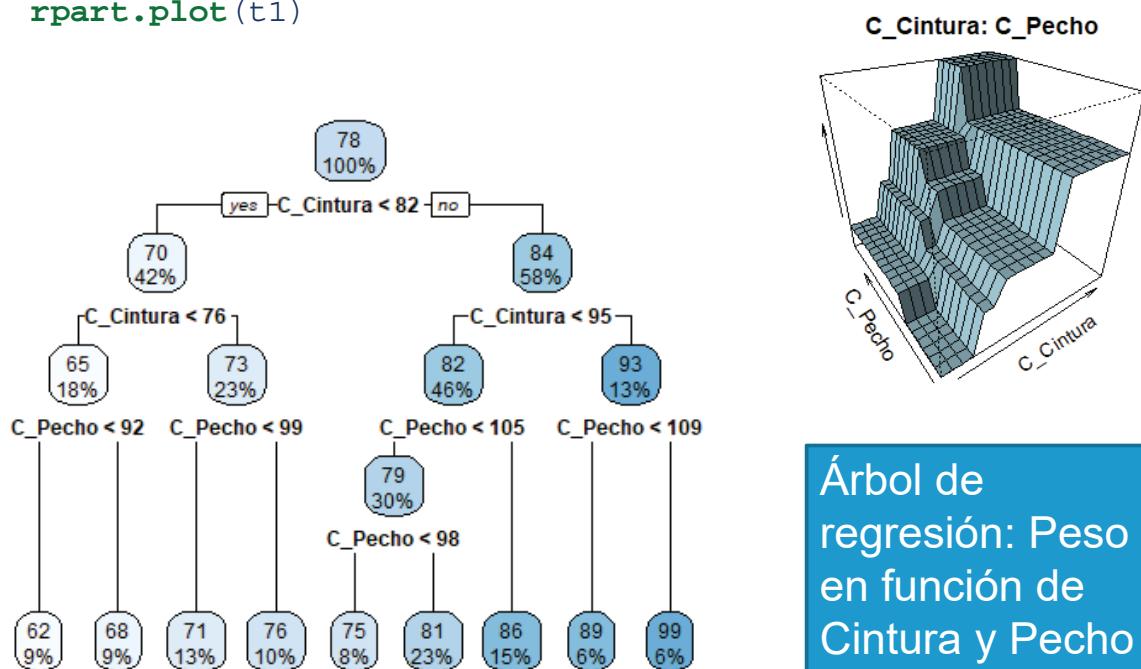
plotmo(m2, degree1 = FALSE, degree2 = TRUE,
       pt.color=1, jitter=.5, smooth.col=3,
       caption = "Regresión Lineal", npoints = 260)

```

Árboles de Regresión

```
t1 = rpart(Peso ~ C_Cintura + C_Pecho,  
            data = hombre)  
rpart.plot(t1)
```

valor previsto



Árbol de
regresión: Peso
en función de
Cintura y Pecho

Árbol

1. El árbol está formado por un conjunto de **nodos** y **ramas**. El nodo superior contiene el 100% de las observaciones y número superior 78 corresponde al peso medio de los datos en ese nodo.
2. La muestra se divide en dos grupos (del nodo superior salen dos ramas), el grupo que cumple **C_Cintura < 82**, y el complementario. Forman el segundo nivel del árbol, que tiene dos nodos el 2 y el 3. El nodo 2 (de la izquierda) contiene el 42% de las observaciones (103) con peso medio de 70 kg y el nodo de la derecha contiene el 58% de las observaciones (144) con una media de 84 kg.
3. Cada nodo se vuelve a subdividir. El nodo 2 se subdivide en dos y el nodo 3 en otros dos. El procedimiento continua hasta que se cumple un criterio de **parada** (por ejemplo que el nodo tenga menos de 10 observaciones).
4. Al final del árbol (nivel inferior) se encuentran los 9 nodos terminales (hojas del árbol), que es una partición de las 247 observaciones en 9 subconjuntos. El árbol está invertido, la raíz arriba y las hojas debajo.

Criterios de Partición y Parada

Para desarrollar el árbol es preciso definir dos aspectos:

- como se elige el criterio de partición de un nodo y
- el criterio de parada, es decir, cuando se termina el proceso de subdivisiones.

7

Criterio de partición

Supongamos que tenemos p variables explicativas o predictores x_1, x_2, \dots, x_p .

Tomamos la primera variable x_1 y buscamos el valor s que divide la muestra en dos regiones

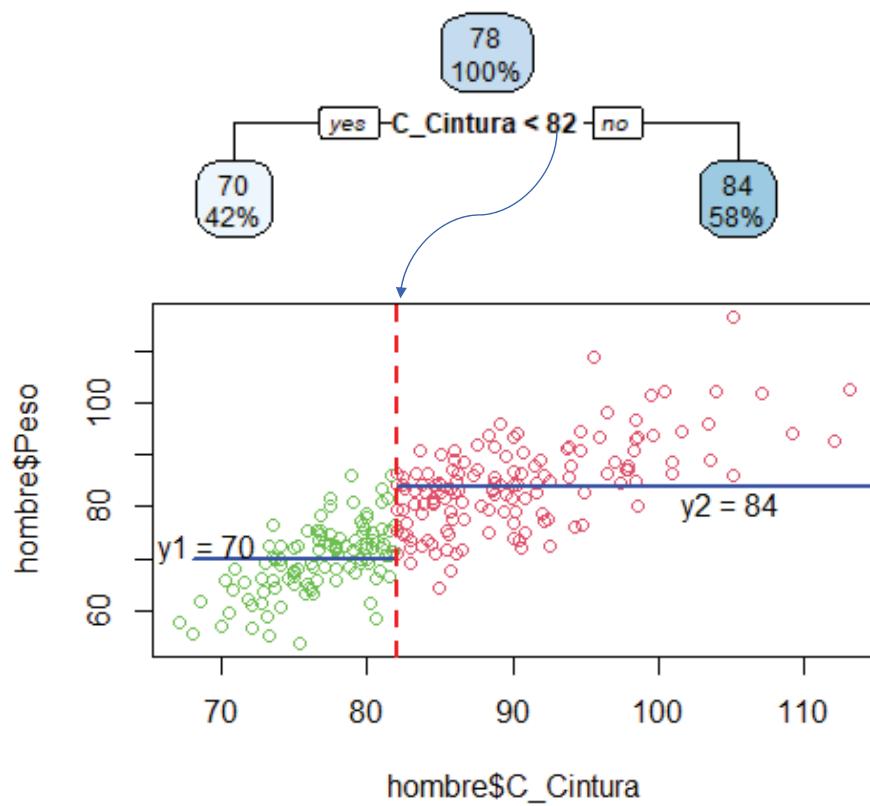
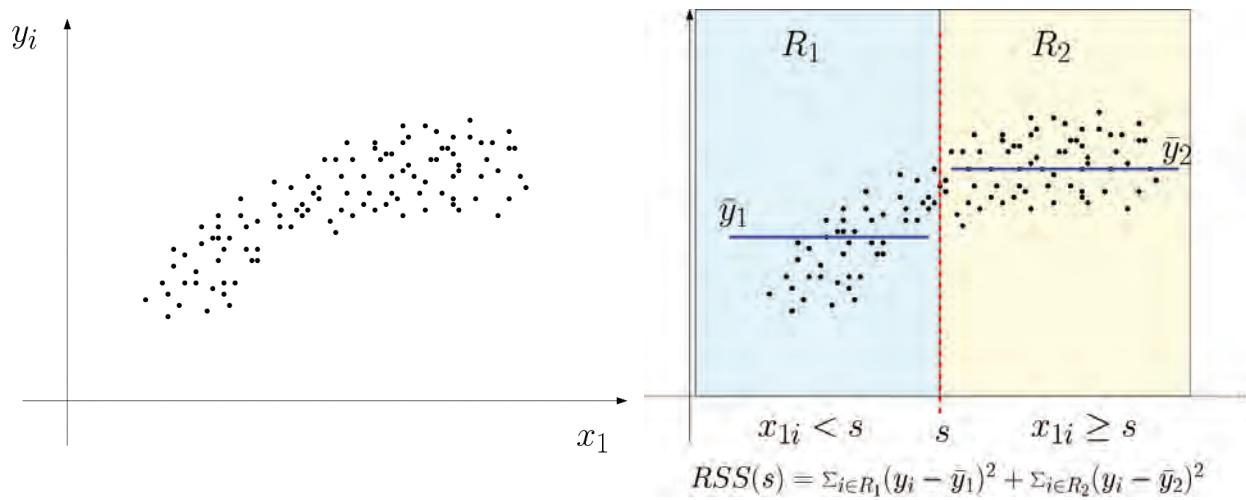
$$R_1 = \{x_{1i} | x_{1i} < s\} \quad y \quad R_2 = \{x_{1i} | x_{1i} \geq s\}$$

que hace mínimo

$$RSS_1(s) = \sum_{i \in R_1} (y_i - \bar{y}_{R_1})^2 + \sum_{i \in R_2} (y_i - \bar{y}_{R_2})^2$$

Siendo \bar{y}_{R_1} e \bar{y}_{R_2} las medias de la variable y (el peso en nuestro ejemplo) en las regiones R_1 y R_2 .

Partición binaria



Criterio de partición (cont)

- Elegido $s = s_1$ para x_1 , se repite el proceso con x_2, x_3, \dots, x_p . De las p variables se elige para hacer la partición la que tiene menor RSS_j .
- El proceso se repite , esta vez en lugar de dividir la muestra completa, se busca la mejor partición de los datos de R_1 y por otro lado de los datos de R_2 .
- El proceso continua hasta que se alcanza el criterio de parada.

Nodos terminales u hojas

Los nodos terminales los denominaremos $R_1^*, R_2^*, \dots, R_J^*$.

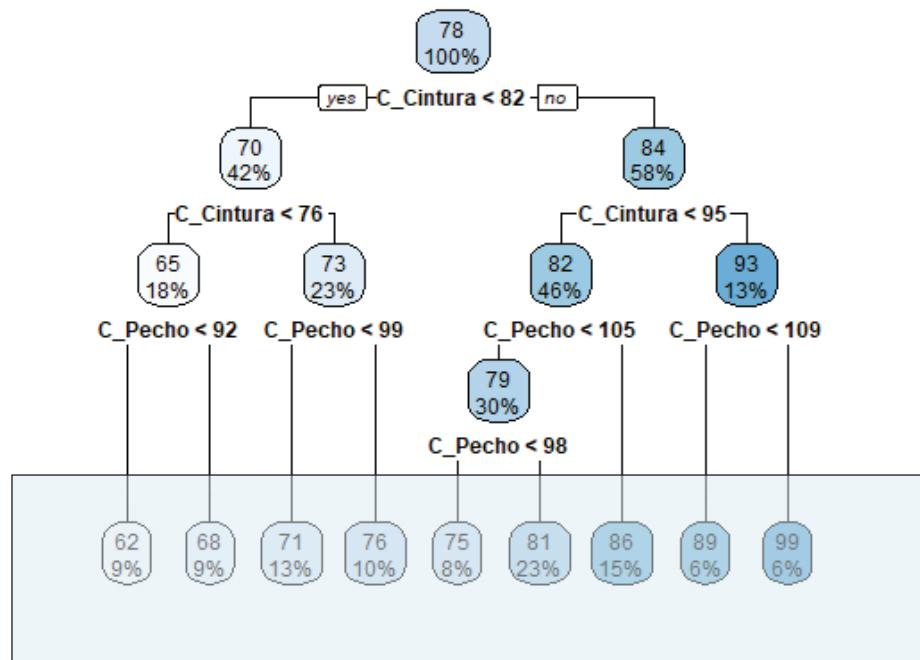
El objetivo es conseguir una partición $R_1^*, R_2^*, \dots, R_J^*$ que minimice

$$RSS_T = \sum_{j=1}^J \sum_{i \in R_j^*} (y_i - \bar{y}_{R_j^*})^2$$

El procedimiento descrito es rápido y los resultados son fáciles de interpretar.

El método no garantiza que no haya otra partición con J regiones que tenga menor RSS_T .

Nodos Terminales



13

Valor previsto

Si una observación

$$x_i = (x_{1i}, x_{2i}, \dots, x_{pi})^T$$

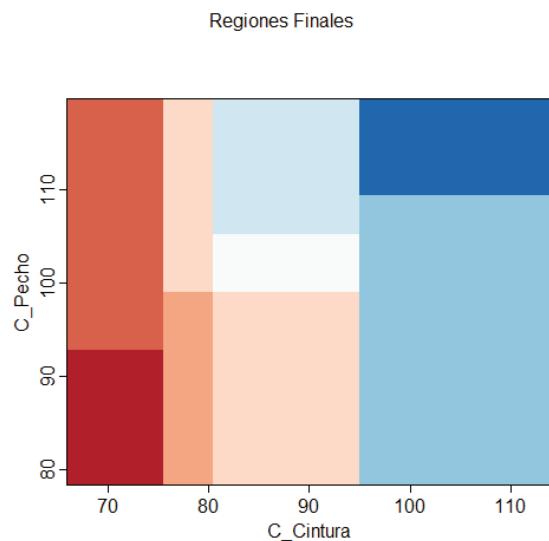
cae en la región R_j^* ,

el valor previsto \hat{y}_i para la variable respuesta (en nuestro caso el peso) de la observación i es $\bar{y}_{R_j^*}$.

14

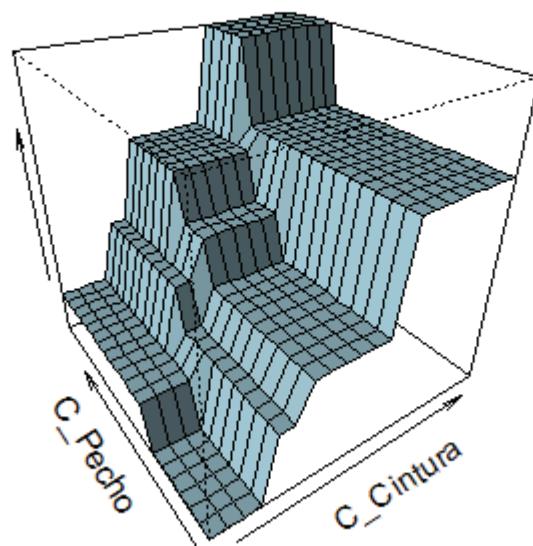
Regiones

En el caso de dos regresores la partición final (nodos terminales) son rectángulos en el espacio bidimensional.



valor previsto

C_Cintura: C_Pecho



```
plotmo(t1, degree1 = FALSE, degree2 = TRUE,  
caption = "valor previsto")
```

Regiones

```
y = hombre$Peso  
  
sol=cbind(  
  n=tapply(y,  
    t1$where, length),  
  perc=tapply(y, t1$where,  
    length)/247*100,  
  pred=tapply(y, t1$where,  
    mean),  
  RSS=tapply(y, t1$where,  
    sd))  
  
rownames(sol)=paste0("R",1:9)  
  
print(sol,digits=3)
```

	n	perc	pred	RSS
## R1	22	8.91	62.4	5.75
## R2	23	9.31	67.9	3.15
## R3	33	13.36	70.9	4.46
## R4	25	10.12	75.8	4.64
## R5	19	7.69	75.4	6.09
## R6	56	22.67	80.7	5.19
## R7	38	15.38	86.4	5.26
## R8	16	6.48	88.6	4.49
## R9	15	6.07	98.6	7.53

Criterio de parada: Complexity parameter (cp)

$$RSS_T = \sum_{j=1}^J \sum_{i \in R_j^*} (y_i - \bar{y}_{R_j^*})^2 = 6339.46$$

Residuo

$$e_i = y_i - \bar{y}_{R_{j(i)}^*}$$

Variabilidad Total

$$VT = \sum_{i=1}^n (y_i - \bar{y})^2 = 27188.13$$

Error relativo

$$\text{rel error} = \frac{RSS_T}{VT} = \frac{6339.46}{27188.13} = 0.23317$$

cp y Coeficiente de determinación

$$R^2 = 1 - \frac{RSS_T}{VT} = 1 - 0.23317 \\ = 0.76683$$

cp = " Aumento de R^2 al aumentar una rama del árbol "

xerror = "La estimación del error relativo por validación cruzada"

Xstd = "La desviación típica de los valores de errores relativos del proceso de validación cruzada"

Cp: complexity parameter

`printcp(t1)`

```
##  
## Root node error: 27188/247 = 110.07  
##  
## n= 247  
##  
##          CP nsplit rel error  xerror      xstd  
## 1 0.472811      0 1.00000 1.00479 0.093448  
## 2 0.122189      1 0.52719 0.56002 0.059511  
## 3 0.056994      2 0.40500 0.47178 0.044711  
## 4 0.046391      3 0.34801 0.41533 0.041334  
## 5 0.028357      4 0.30161 0.39946 0.041445  
## 6 0.014970      5 0.27326 0.37951 0.034625  
## 7 0.012607      6 0.25829 0.37096 0.032918  
## 8 0.012511      7 0.24568 0.35623 0.032019  
## 9 0.010000      8 0.23317 0.35483 0.031987
```

Criterios de parada

El árbol deja de crecer cuando se cumple alguna de estas condiciones:

- El incremento de R^2 al añadir una nueva rama es menor que cierto umbral **cp** (por defecto 0.01)
- Cuando el número de observaciones en el nodo es menor que cierto umbral **minsplit** (por defecto 20)
- Cuando la profundidad del árbol es superior a cierto umbral **maxdepth** (por defecto 30)

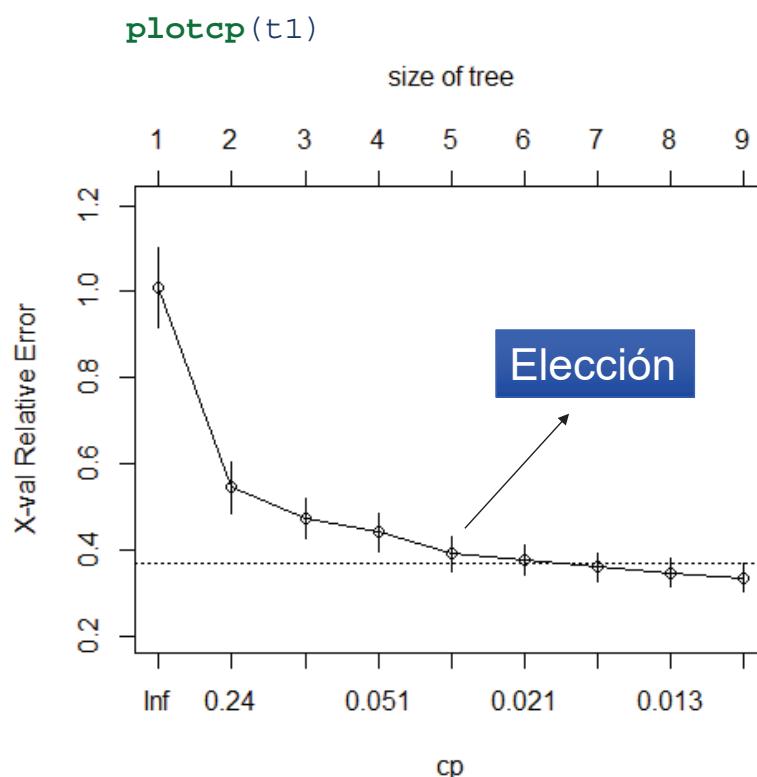
Podar el árbol (prune)

- La estrategia que se recomienda a la hora de construir un árbol de regresión es obtener un primer árbol con muchas ramas, utilizando criterios de parada muy generosos (por ejemplo $cp=0.01$).
- Analizar el árbol y eliminar las ramas que no implican un aumento suficiente de R^2 o una reducción significativa de los errores.
- Es muy habitual que los árboles proporcionen sobreajustes que son perjudiciales a la hora de predecir nuevas observaciones. Lo recomendable es utilizar el árbol más pequeño posible con un R^2 lo más alto posible.

Elección de **cp**

- Gráfico de *xerror* y se elige el árbol más pequeño que presenta un error similar.
- Se toma el valor de **cp** correspondiente a el árbol elegido
- Se poda el árbol con la instrucción **prune()**

Complexity parameter (plot)

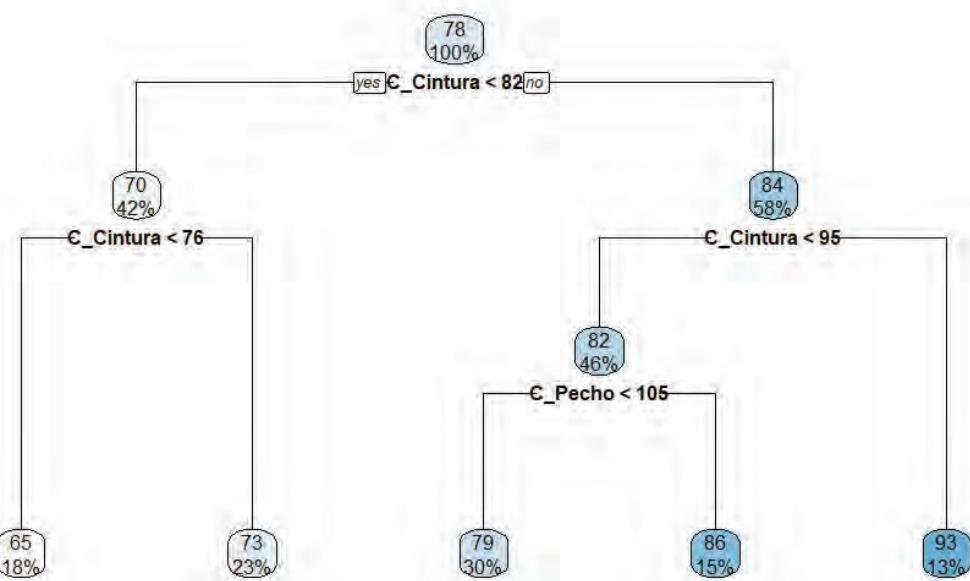


Cp: complexity parameter

```
printcp(t1)
```

```
##  
## Root node error: 27188/247 = 110.07  
##  
## n= 247  
##  
##          CP nsplit rel error xerror      xstd  
## 1 0.472811      0 1.00000 1.00479 0.093448  
## 2 0.122189      1 0.52719 0.56002 0.059511  
## 3 0.056994      2 0.40500 0.47178 0.044711  
## 4 0.046391      3 0.34801 0.41533 0.041334  
## 5 0.028357      4 0.30161 0.39946 0.041445  
## 6 0.014970      5 0.27326 0.37951 0.034625  
## 7 0.012607      6 0.25829 0.37096 0.032918  
## 8 0.012511      7 0.24568 0.35623 0.032019  
## 9 0.010000      8 0.23317 0.35483 0.031987
```

```
t2 = prune(t1, cp=0.036)  
rpart.plot(t2)
```



```

summary(t2)

## Call:
## rpart(formula = Peso ~ C_Cintura + C_Pecho, data = hombre)
##   n= 247
##
##           CP nsplit rel error     xerror      xstd
## 1 0.47281091      0 1.0000000 1.0047910 0.09344844
## 2 0.12218927      1 0.5271891 0.5600211 0.05951083
## 3 0.05699397      2 0.4049998 0.4717840 0.04471147
## 4 0.04639090      3 0.3480058 0.4153329 0.04133446
## 5 0.03600000      4 0.3016149 0.3994598 0.04144505
##
## Variable importance
## C_Cintura    C_Pecho
##       67        33
##

```

Todas las variables (hombres y mujeres)

- Dividir la muestra en dos partes
 - ▲ train : datos para estimar
 - ▲ test : datos para evaluar

```

set.seed(123)

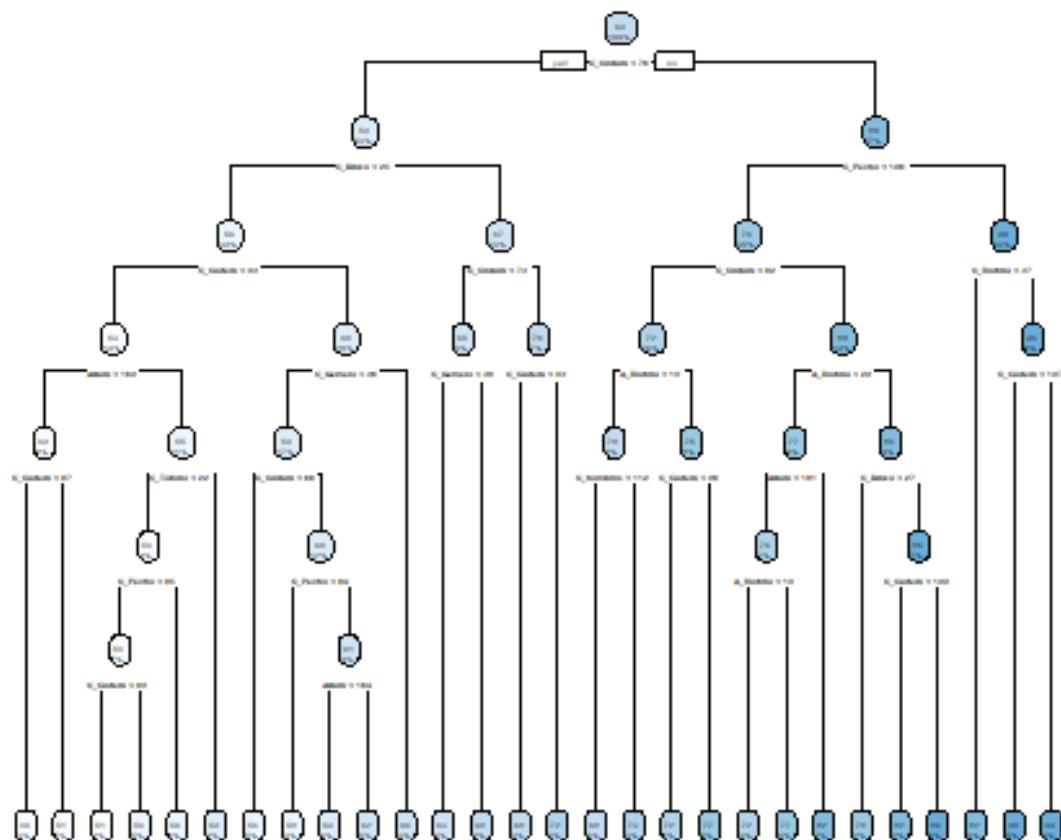
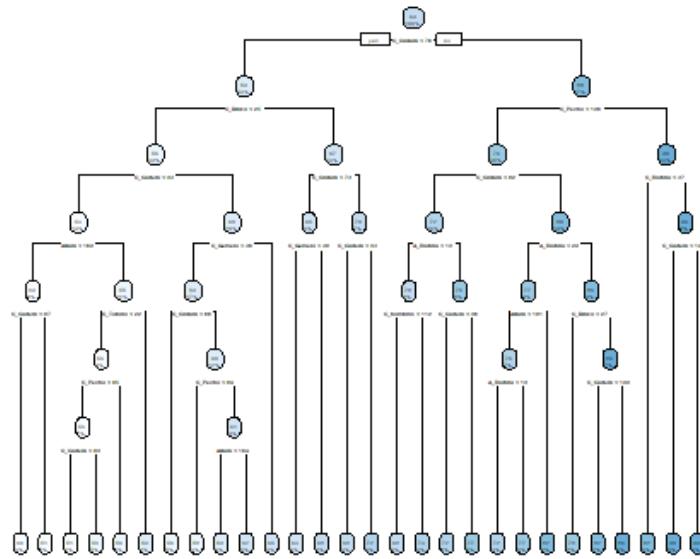
sel = sample(1:507, size=350, replace = FALSE)

train = dat[sel,]
test = dat[-sel,]

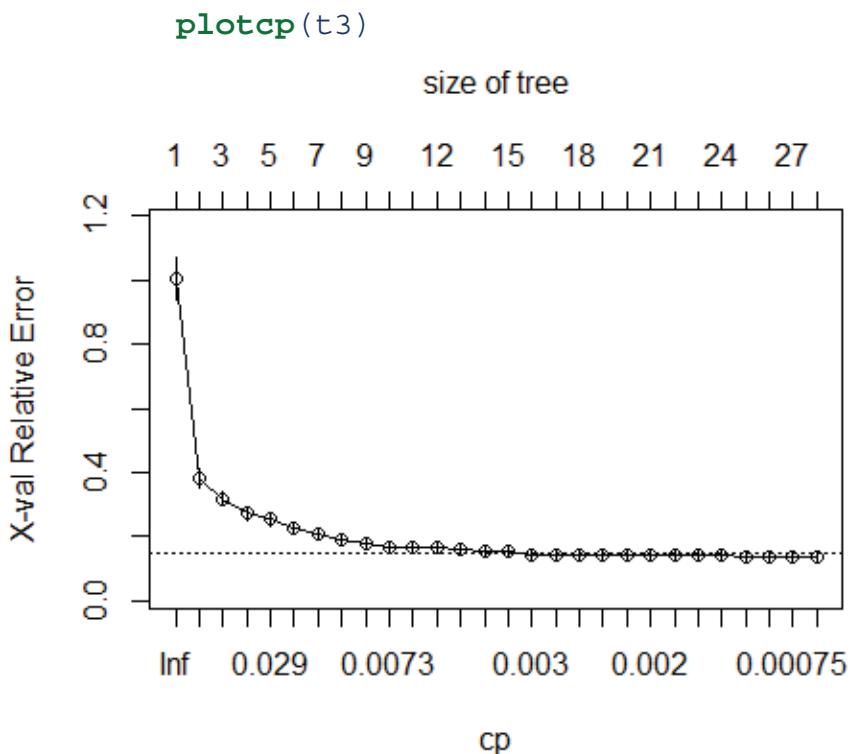
```

Estimación del árbol

```
t3 = rpart(Peso ~ ., data = train, cp=.0001)  
rpart.plot(t3)
```

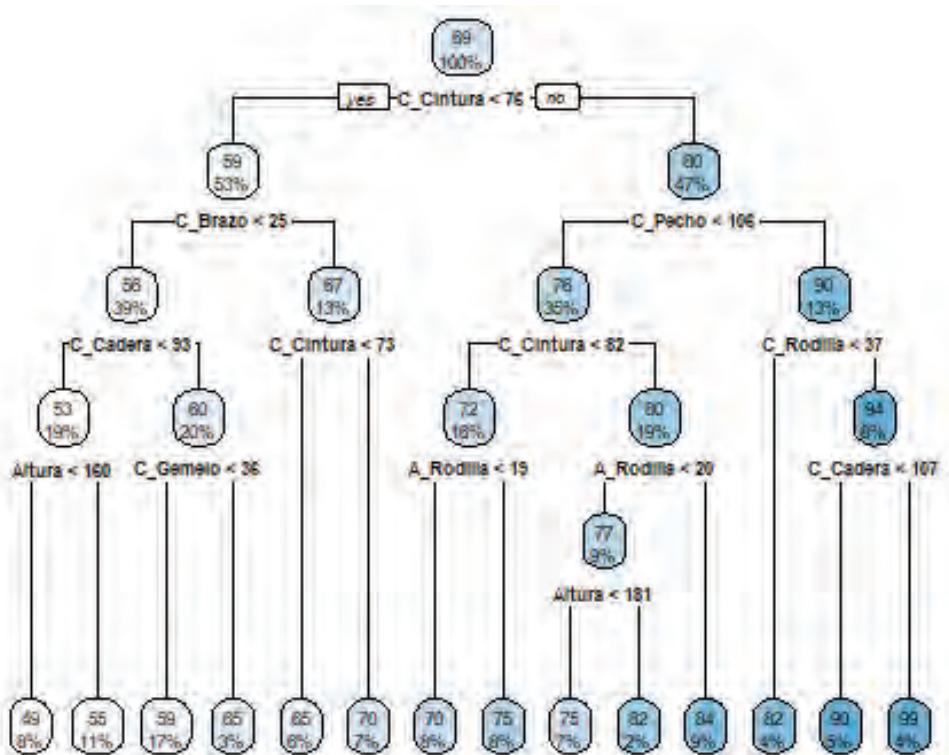


Selección del árbol



Poda con cp=0.004

```
t4 = prune(t3, 0.004)  
rpart.plot(t4)
```

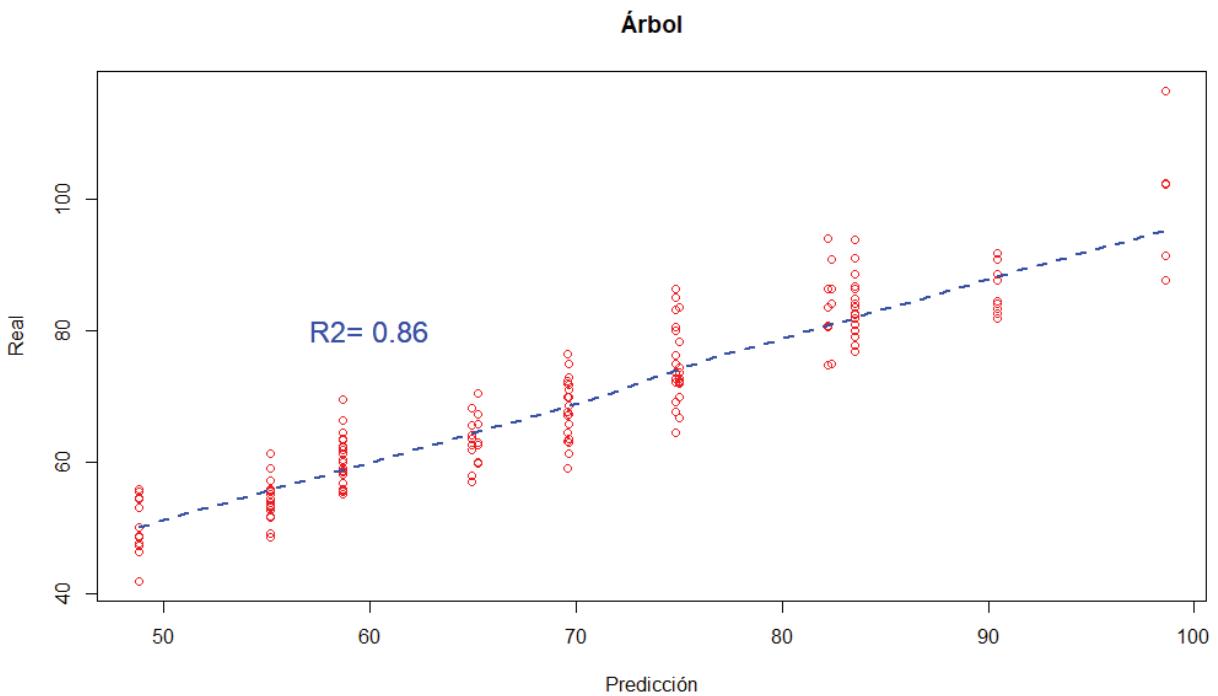


Bondad de ajuste

```
y = test$Peso  
  
VT = sum( (y-mean(y))^2 )  
  
yhat = predict(t4,newdata = test)  
  
e = y-yhat  
  
VNE = sum(e^2)  
  
R2 = 1 - VNE/VT
```

Bondad de ajuste Árbol

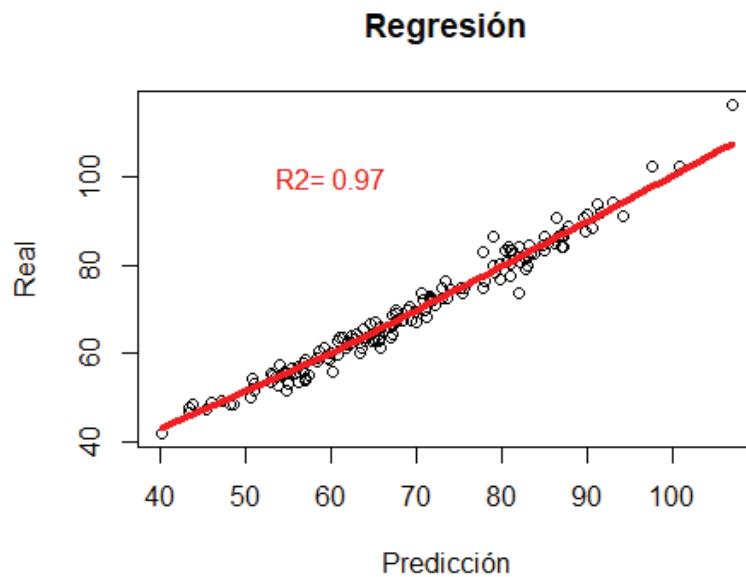
```
par(mfrow=c(2,1))  
scatter.smooth(yhat,y,lpars = list(col = "blue", lwd =  
4, lty = 1),  
               main="Árbol",  
               xlab = "Predicción",  
               ylab= "Real")  
text(60,100,paste("R2=",round(R2*100)/100),col="blue")
```



Regresión lineal

```
m2 = lm(Peso ~ ., data = train)
m3 = step(m2, trace = 0)
yhat = predict(m3, newdata = test)
e = y-yhat
VNE = sum(e^2)
R2 = 1 - VNE/VT
scatter.smooth(yhat,y,lpars = list(col =
"red", lwd = 4, lty = 1),
               main="Regresión",
               xlab = "Predicción",
               ylab= "Real")

text(60,100,paste("R2=",round(100*R2)/100
),col="red")
```



ANALISIS DE DATOS

Lección 6 Parte 2

CART Y RANDOM FOREST PARA REGRESIÓN

(Random Forest)

Random Forest para Regresión (Versión 1: bagging)

El objetivo predecir y_i una variable respuesta continua en función de p variables explicativas $x_i = (x_{1i}, x_{2i}, \dots, x_{pi})^T$. El número de observaciones es n . El conjunto para la estimación lo llamamos $d = (X, y)$.

Para $b = 1, 2, \dots, B$ hacemos lo siguiente:

bootstrap

1.- Tomamos una muestra al azar de n observaciones con reemplazamiento de las filas de d . Llamamos d_b^* a la muestra.

2.- Construimos un árbol que llamaremos $\hat{r}_b(x)$

3.- Guardamos d_b^* y $\hat{r}_b(x)$.

Se ha formado un conjunto de árboles por $\hat{r}_b(x)$, $b = 1, 2, \dots, B$. Al procedimiento se denomina **bagging**

Random Forest para Regresión (Versión 2: auténtico random forest)

Igual que en **bagging**: el objetivo predecir y_i una variable respuesta continua en función de p variables explicativas $x_i = (x_{1i}, x_{2i}, \dots, x_{pi})^T$. El número de observaciones es n . El conjunto para la estimación lo llamamos $d = (X, y)$.

Para $b = 1, 2, \dots, B$ hacemos lo siguiente:

1.- Tomamos una muestra al azar de n observaciones con reemplazamiento de las filas de d . Llamamos d_b^* a la muestra.

2.- Construimos un árbol que llamaremos $\hat{r}_b(x)$ pero no se utilizan todas las variables cada vez, en cada nodo de las p variables elegimos al azar $m < p$ para hacer la división. Una elección habitual es $m = \sqrt{p}$.

3.- Guardamos d_b^* y $\hat{r}_b(x)$.

Al conjunto formado por $\hat{r}_b(x)$, $b = 1, 2, \dots, B$ se denomina **random forest**

Predictión para

$$x_i = (x_{1i}, x_{2i}, \dots, x_{pi})^T$$

$$\hat{y}_i = \hat{r}_{rf}(x_i) = \frac{1}{B} \sum_{b=1}^B \hat{r}_b(x_i)$$

$$e_i = y_i - \hat{y}_i$$

$$MSE = \frac{1}{n} \sum_1^n e_i^2$$

$$R^2 = 1 - \frac{\sum_1^n e_i^2}{\sum_1^n (y_i - \bar{y})^2}$$

Random Forest

```
set.seed(123)
sel = sample(1:507, size=350, replace = FALSE)
train = dat[sel,]
test = dat[-sel,]

r0 = randomForest(Peso ~ A_Hombros+A_Pelvis+A_Cade+AP_Pecho+AD_Pecho,
data =train)
```

```
r1 = randomForest(Peso ~ ., data =train) # todas las variables
```

```
print(r1)
##
## Call:
##   randomForest(formula = Peso ~ ., data = train)
##   Type of random forest: regression
##   Number of trees: 500
##   No. of variables tried at each split: 8
##
##   Mean of squared residuals: 8.322204
##   % Var explained: 95.28
```

Estimación de error (datos de test)

La estrategia habitual con los algoritmos de **machine learning** es dividir el conjunto de datos en dos partes: parte de estimación o entrenamiento (75%) y parte de test o validación (25%).

Para una observación (y_i, x_i) del conjunto **test** se calcula el error

$$e_i = y_i - \hat{r}_{rf}(x_i)$$

y la deviación típica residual es

$$\hat{s}_R = \sqrt{\frac{e_i^2}{m-1}}$$

siendo m el número de observaciones en el conjunto de validación.

```
set.seed(123)
sel = sample(1:507, size=350, replace = FALSE)
train = dat[sel,]
test = dat[-sel,]
```

Out of the Bag error (OOB)

- Cada vez que tomamos una muestra de las observaciones para el árbol b, aproximadamente 1/3 de las obs. se quedan fuera (out of the bag)
- Se obtiene el **valor predicho** con el árbol b para las observaciones OOB
- Para cada observación se hace la media de las predicciones OOB (**predicción OOB**)
- Se obtiene el residuo o **OOB error** como la diferencia entre valor observado y predicho

names(r1)

- **call** : the original call to randomForest
- **type** : one of regression, classification, or unsupervised.
- **predicted** : the predicted values of the input data based on out-of-bag samples.
- **importance**: a matrix with two columns. The first column is the mean decrease in accuracy and the second the mean decrease in MSE. If importance=FALSE, the last measure is still returned as a vector.
- **importanceSD**: The “standard errors” of the permutation-based importance measure. For regression, a length p vector.

8

Names (r1) cont,

- **ntree** : number of trees grown.
- **mtry** : number of predictors sampled for splitting at each node.
- **forest** : (a list that contains the entire forest; NULL if randomForest is run in unsupervised mode or if keep.forest=FALSE.
- **oob.times**: number of times cases are ‘out-of-bag’ (and thus used in computing OOB error estimate)
- **mse**: vector of mean square errors: sum of squared residuals divided by n.
- **rsq** : “pseudo R-squared”: $1 - \text{mse} / \text{Var}(y)$.

9

Importancia de las variables

Un *random forest* es una caja negra que proporciona buenas predicciones, pero no proporciona infomación de la relación de cada variable explicativa y la variable dependiente.

Dos medidas:

- IncNodePurity
- %IncMSE

Importancia de las variables (IncNodePurity)

En cada árbol se seleccionan al azar *mtry* variables entre las *p* opciones. De las elegidas las más importantes (explican más variabilidad) serán usadas en varias ramas y las menos importantes, apenas aparecerán en el árbol. Cada vez que una variable es utilizada en un árbol (nodo), el algoritmo guarda la variabilidad explicada por la partición generada por la variable (varianza antes de la partición menos la suma de las varianzas de cada grupo). Estos valores se acumulan para cada variable utilizando todos los árboles del “*random forest*” y se denomina importancia de la variable.

Importancia de las variables (%IncMSE)

%IncMSE es más robusto e informativo. Mide el aumento en MSE (errores) si se sustituye la variable j por la variable j permutada al azar.

Pasos:

1.- Se construye el modelo random forest. Se calcula el OOB-MSE y lo denominamos MSE0

2.- Para cada variable, se permuta al azar los valores de la variable j, se construye un nuevo RF y se obtiene el OOB-MSEj

3.- %IncMSE de j es

$$\frac{MSE_j - MSE_0}{MSE_0} \times 100$$

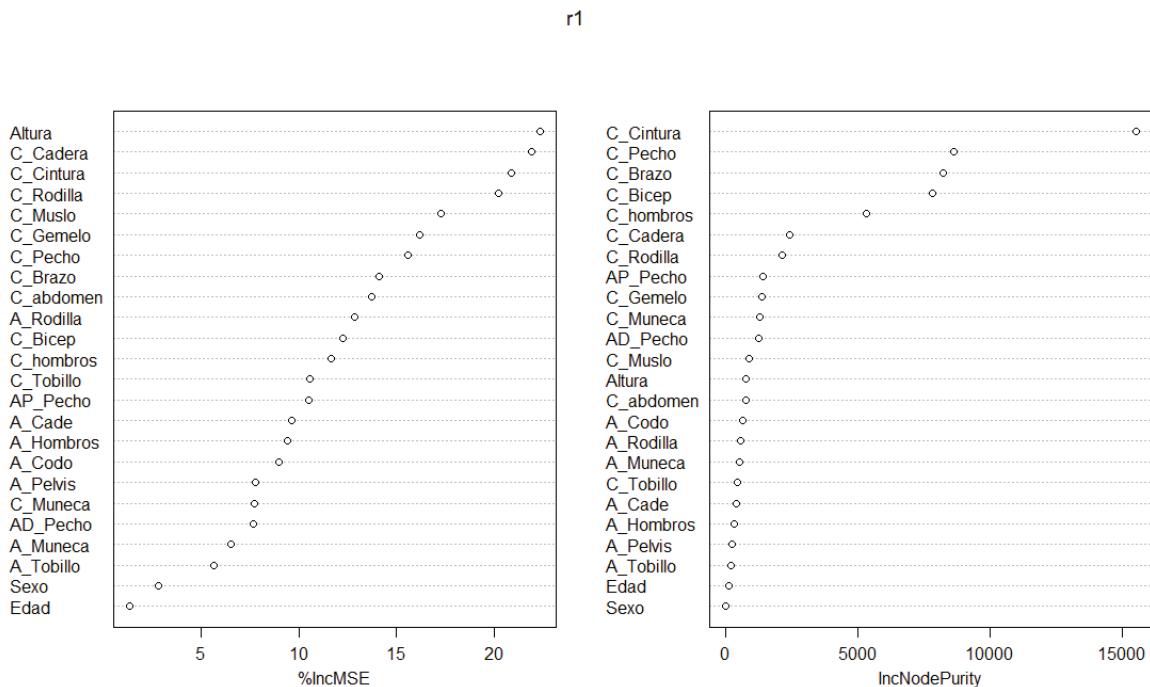
Cuanto mayor es el número, más importante es la variable

```
set.seed(123)
r1 = randomForest(Peso ~ ., data =train, importance=TRUE)
print(importance(r1), digits = 3)
```

	%IncMSE	IncNodePurity
## A_Hombros	9.43	308.9
## A_Pelvis	7.76	236.0
## A_Cade	9.67	410.6
## AP_Pecho	10.52	1400.4
## AD_Pecho	7.65	1239.8
## A_Codo	8.96	649.4
## A_Muneca	6.53	529.9
## A_Rodilla	12.86	545.6
## A_Tobillo	5.67	192.3
## C_hombros	11.67	5322.5
## C_Pecho	15.62	8632.5
## C_Cintura	20.90	15519.8

	%IncMSE	IncNodePurity
## C_abdomen	13.73	757.7
## C_Cadera	21.95	2421.0
## C_Muslo	17.32	902.7
## C_Bicep	12.24	7822.9
## C_Brazo	14.14	8212.9
## C_Rodilla	20.25	2117.3
## C_Gemelo	16.23	1352.0
## C_Tobillo	10.58	426.6
## C_Muneca	7.72	1276.1
## Edad	1.34	108.5
## Altura	22.37	780.6
## Sexo	2.80	11.6

varImpPlot (r1)



14

Elección de mtry

```

oob.err=double(13)
test.err=double(13)
set.seed(2354)
n = 507
sel = sample(1:n, size=round(n*.75))
##mtry is no of Variables randomly chosen at each split
for(mtry in 1:13)
{
  rf=randomForest(Peso ~ . , data = dat , subset = sel,
                  mtry=mtry,ntree=400)
  oob.err[mtry] = rf$mse[400]

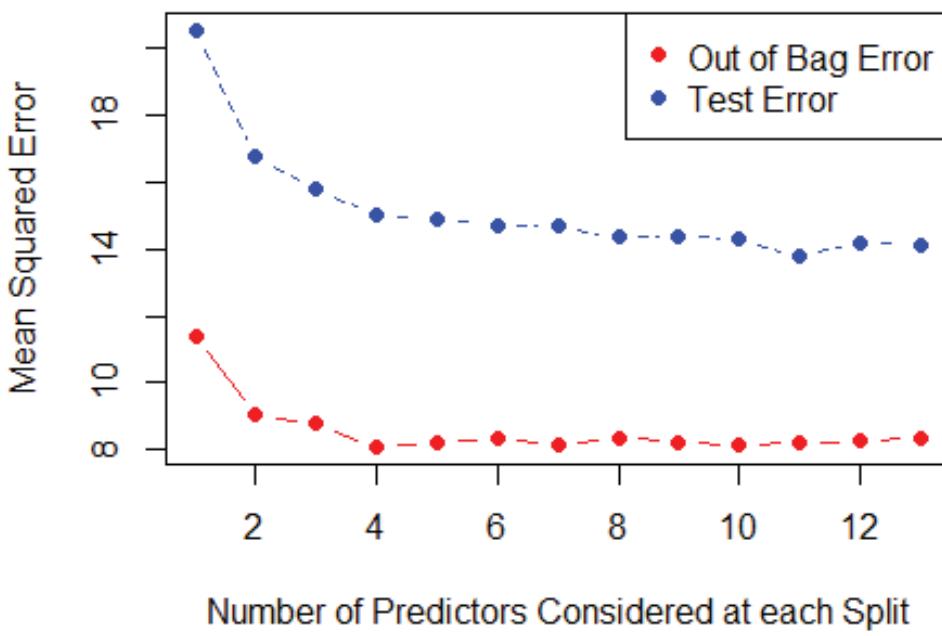
  pred<-predict(rf,dat[-sel,]) #Predictions on Test Set for each Tree
  test.err[mtry]= with(dat[-sel,], mean( (Peso - pred)^2)) #Mean Squared Test
Error

}

matplot(1:mtry , cbind(oob.err,test.err),
        pch=19 , col=c("red","blue"),
        type="b",ylab="Mean Squared Error",
        xlab="Number of Predictors Considered at each Split")
legend("topright",
       legend=c("Out of Bag Error","Test Error"),
       pch=19, col=c("red","blue"))

```

15



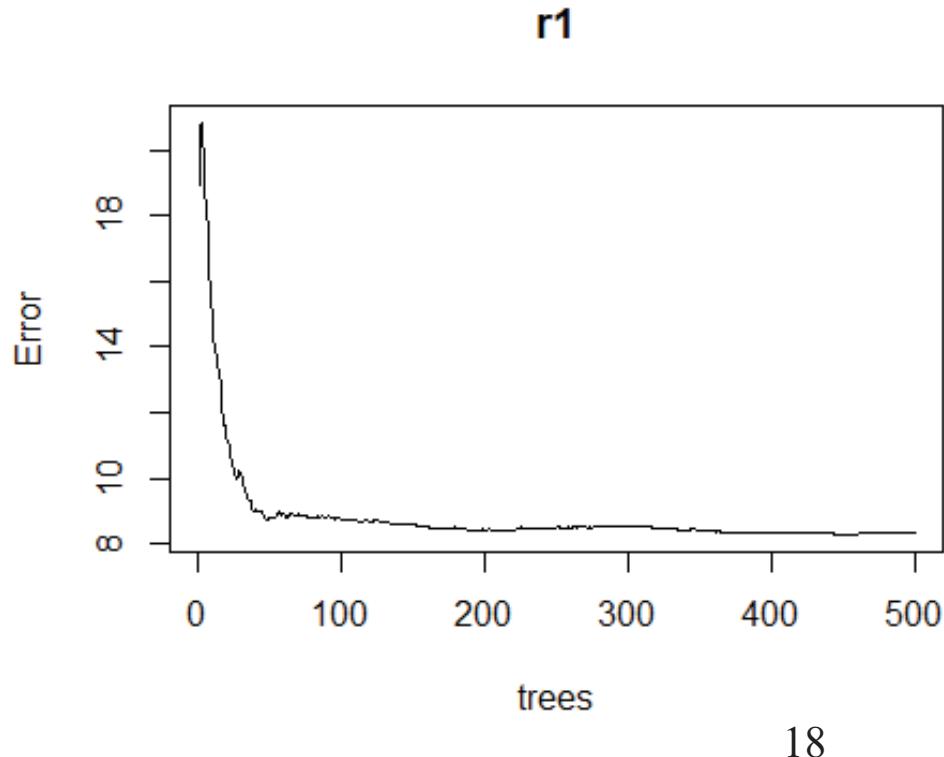
Los resultados con mtry = 4 no cambian en este caso

16

```
set.seed(123)
r2 = randomForest(Peso ~ ., data = train, mtry=4, importance=TRUE)
print(r2)

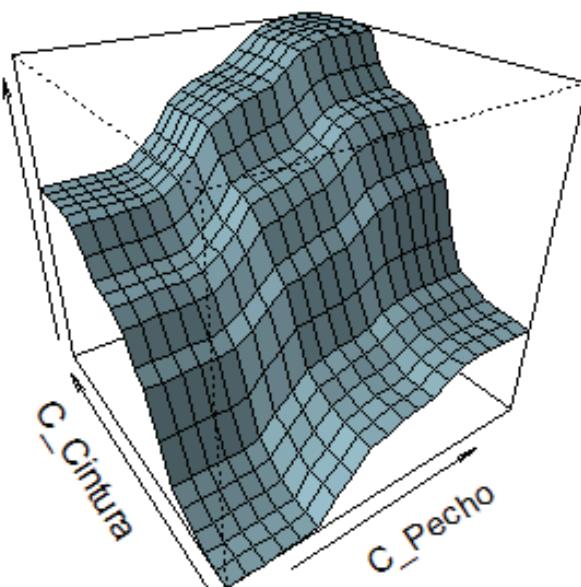
##
## Call:
##  randomForest(formula = Peso ~ ., data = train, mtry = 4, importance =
## TRUE)
##          Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##          Mean of squared residuals: 8.733008
##                         % Var explained: 95.05
```

Número de árboles en el bosque `plot(r1)`



```
Peso randomForest(Peso~, data=train)
```

C_Pecho: C_Cintura



Cuidado al interpretar este gráfico. El resto de las variables se dejan constante igual a su mediana.

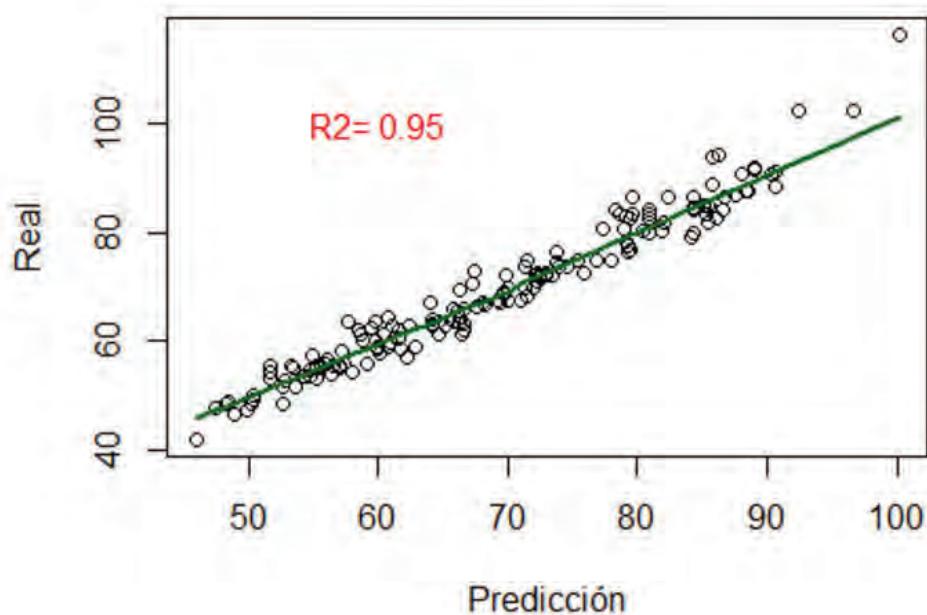
```
plotmo(r1, degree1 = FALSE, degree2 = 1)
```

Predicción RF

```
y = test$Peso
VT = sum((y-mean(y))^2)
yhat = predict(r2,newdata = test)
e = y-yhat
VNE = sum(e^2)
R2 = 1 - VNE/VT
scatter.smooth(yhat,test$Peso,
  lpars = list(col = "darkgreen", lwd = 2,lty = 1),
  main="Random Forest",
  xlab = "Predicción",
  ylab= "Real")

text(60,100,paste("R2=",round(100*R2)/100),
  col="red")
```

Random Forest



Predicción

```
y = test$Peso
VT = sum((y-mean(y))^2)
yhat = predict(r1,newdata = test)
e = y-yhat
VNE = sum(e^2)
R2 = 1 - VNE/VT
scatter.smooth(yhat,test$Peso,
    lpars = list(col = "green", lwd = 4,lty =
1),
    main="Random Forest",
    xlab = "Predicción",
    ylab= "Real")

text(60,100,paste("R2=",round(100*R2)/100) ,
    col="green")
```

22

Comparación con Regresión Lineal

```
m0 = lm(Peso ~ ., data = train)
m = step(m0,trace = 0)
summary(m)

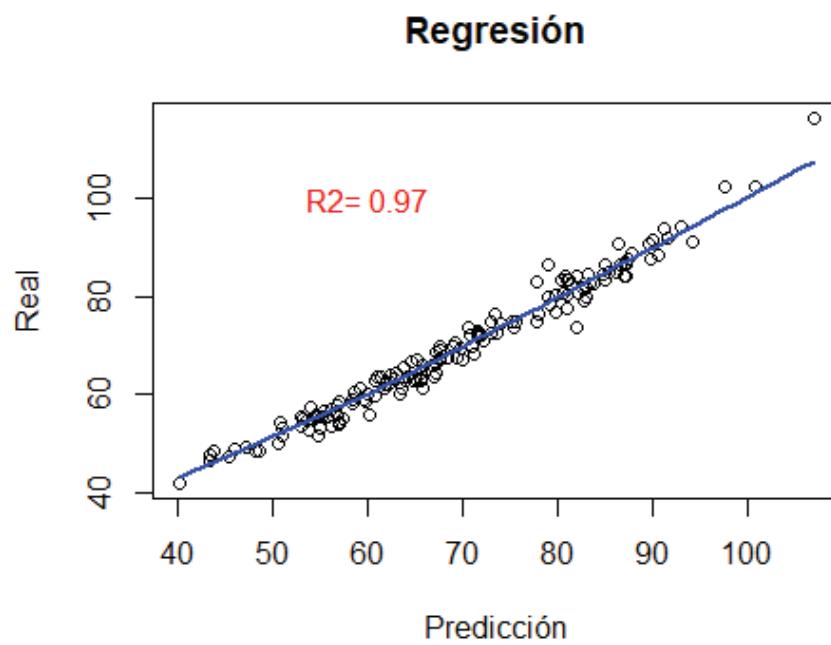
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -121.95836   2.97276 -41.025 < 2e-16 ***
## A_Pelvis     0.11328   0.06575   1.723 0.085848 .
## AP_Pecho     0.35501   0.07888   4.501 9.39e-06 ***
## AD_Pecho     0.16147   0.08784   1.838 0.066924 .
## A_Muneca     0.45237   0.24688   1.832 0.067796 .
## A_Rodilla    0.31762   0.14922   2.129 0.034026 *  
## C_hombros    0.07296   0.03492   2.090 0.037411 *  
## C_Pecho      0.13391   0.04085   3.278 0.001156 ** 
## C_Cintura    0.35394   0.03044   11.629 < 2e-16 ***
## C_Cadera     0.23413   0.04643   5.042 7.58e-07 ***
## C_Muslo      0.23980   0.06021   3.983 8.38e-05 ***
## C_Brazo       0.59802   0.13042   4.585 6.43e-06 ***
## C_Rodilla    0.36248   0.08682   4.175 3.81e-05 *** 
## C_Gemelo     0.27886   0.07262   3.840 0.000147 *** 
## C_Muneca    -0.40121   0.23023  -1.743 0.082317 .  
## Edad        -0.04582   0.01431  -3.203 0.001491 ** 
## Altura       0.30084   0.02066  14.560 < 2e-16 ***
## SexoHombre   -0.92716   0.57561  -1.611 0.108184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.027 on 332 degrees of freedom
## Multiple R-squared:  0.9779, Adjusted R-squared:  0.9768
## F-statistic: 863.7 on 17 and 332 DF,  p-value: < 2.2e-16
```

```

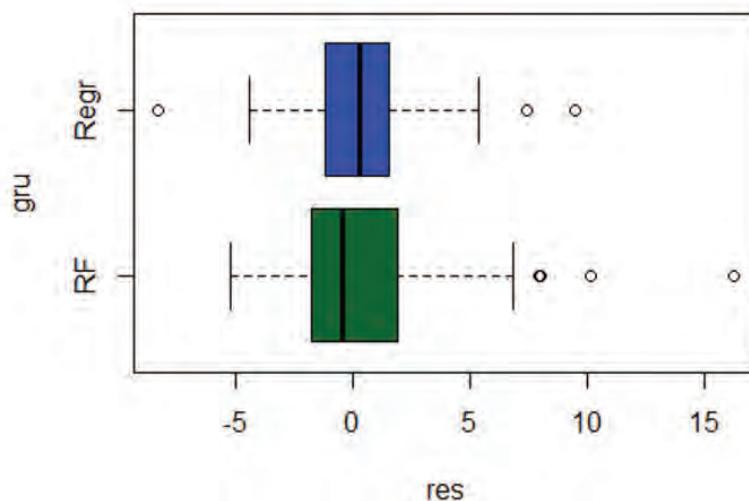
y = test$Peso
VT = sum((y-mean(y))^2)
yhat = predict(m,newdata = test)
e = y-yhat
VNE = sum(e^2)
R2 = 1 - VNE/VT
scatter.smooth(yhat,test$Peso,
  lpars = list(col = "blue", lwd = 2,lty = 1),
  main="Regresión",
  xlab = "Predicción",
  ylab= "Real")

text(60,100,paste("R2=",round(100*R2)/100),
  col="red")

```



Boxplot de residuos de los dos modelos



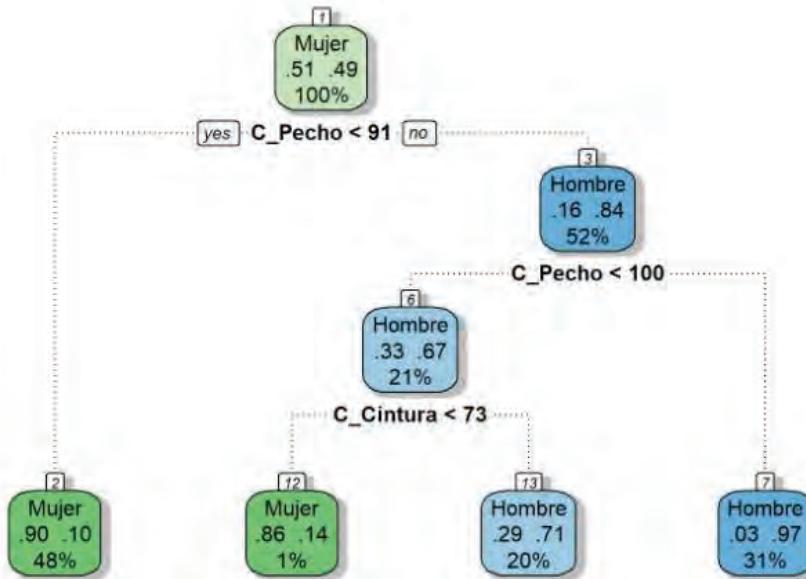
ANALISIS DE DATOS

Lección 6 Parte 3

CART Y RANDOM FOREST PARA CLASIFICACIÓN *(Random Forest)*

Árboles de Clasificación

```
t1 = rpart(Sexo ~ C_Cintura + C_Pecho, cp=.01,data =dat)
#rpart.plot(t1,extra = 2)
fancyRpartPlot(t1, caption = NULL)
```



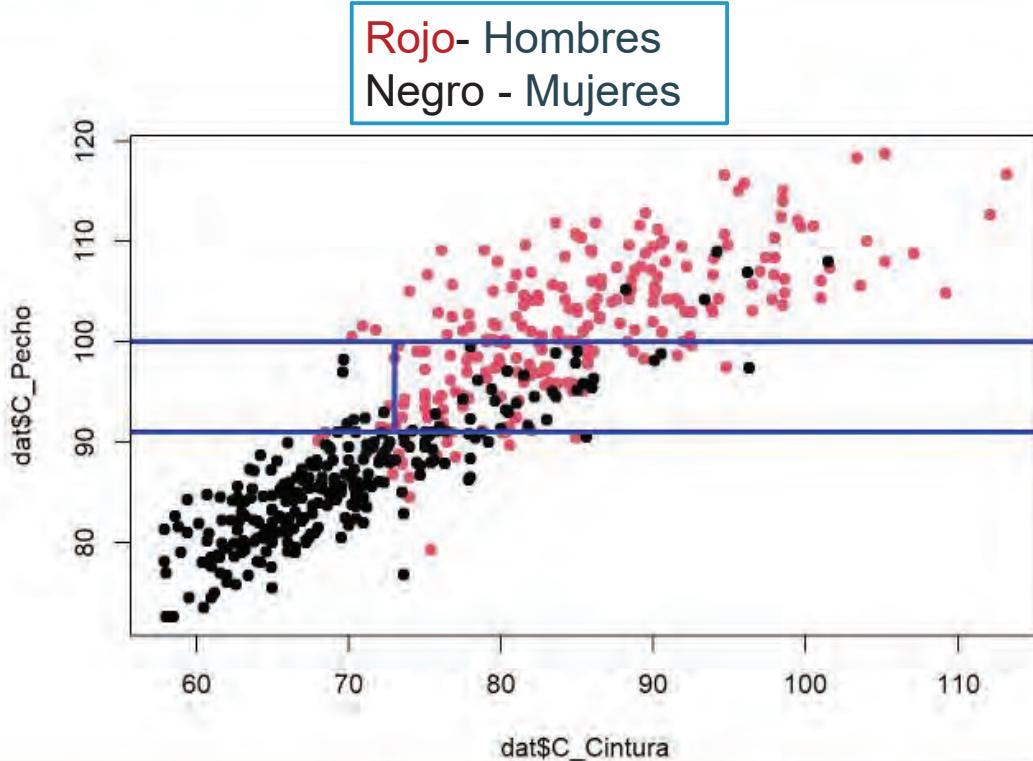
CART y Random Forest

Jesús Juan

1/9/2020 (updated: 2020-12-01)

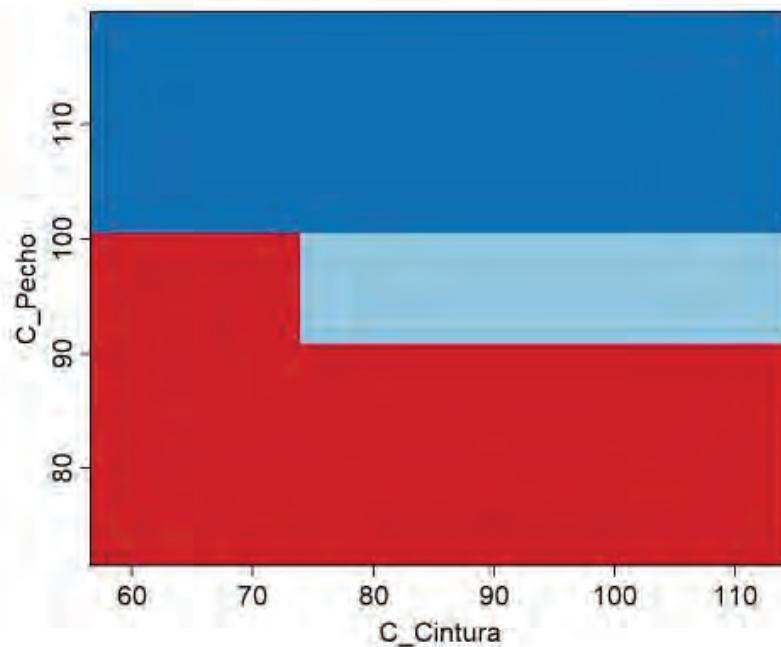
```
library(rpart)
library(rpart.plot)
library(plotma)
library("RColorBrewer")
library(randomForest)
library(gbm)
library(rattle)
dat <- read.table("../R_WORK/QUERPO.TXT", header=TRUE )
dat$Sexo = factor(dat$Sexo,labels=c("Mujer","Hombre"))
```

```
plot(dat$C_Cintura,dat$C_Pecho,col=dat$Sexo,pch=19)
abline(h=91,col="blue",lwd=3,lty=1)
abline(h=100,col="blue",lwd=3,lty=1)
segments(73,91,73,100,col="blue",lwd=3,lty=1)
```



Regiones Finales

Regiones Finales



Índice de Impureza de Gini

Es un indicador de **desorden** de un conjunto con información cualitativa. Por ejemplo

$$A = \{H, H, H, H, M, M, M\}$$

, las clases de este conjunto son **Hombre** (H) y **Mujer** (M).

Seleccionamos un elemento al azar. Luego lo etiquetamos al azar de acuerdo a la distribución de clases en el conjunto. Esto sería equivalente a etiquetar el elemento seleccionado tirando un dado de 7 caras con 4 caras H y 3 caras M. La probabilidad de que clasifiquemos erróneamente el elemento es igual a la probabilidad de que seleccionemos un elemento H multiplicada por la probabilidad de que lo etiquetemos como M más la probabilidad de que seleccionemos un elemento M multiplicada por la probabilidad de que lo etiquetemos como H. Esto es

$$I_G(A) = 3/7 \times 4/7 + 4/7 \times 3/7 = 24/49 = 0.490$$

Índice de impureza de Gini (General)

En un conjunto B con m clases, con proporciones p_1, p_2, \dots, p_m , el índice de Gini es

$$I_G(B) = \sum_{i=1}^m p_i(1 - p_i)$$

El índice de Gini para una conjunto con todos los elementos de la misma clase es 0 (máxima pureza).

El índice de Gini es máximo (máxima impureza) cuando hay la misma proporción de elementos de todas las clases. En el caso de dos clases equiprobables el índice de Gini es $1/2$. En el caso de muchas clase (cuando $m \rightarrow \infty$) el máximo valor es 1.

$$I_G(B) = \sum_{i=1}^m p_i(1 - p_i) = \sum_{i=1}^m p_i - \sum_{i=1}^m p_i^2 = 1 - \sum_{i=1}^m p_i^2$$

Partición de un conjunto (Ganancia)

$$A = \{H, H, H, H, M, M, M\}$$

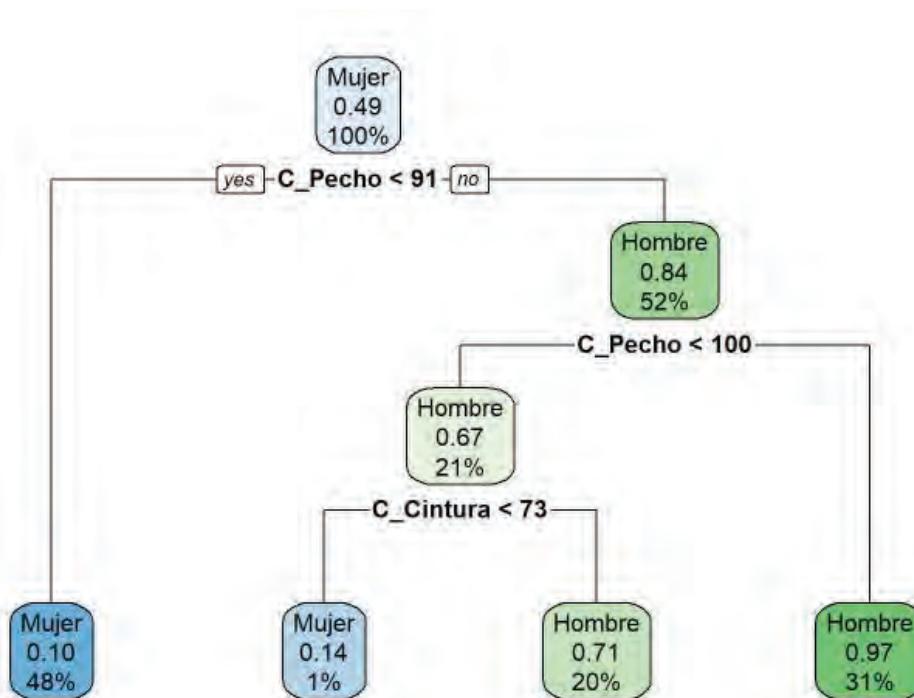
Supongamos que la condición $Peso < 65$ nos divide el conjunto A en $A_P = \{H, M, M\}$ $A_N = \{H, H, H, M\}$

$$I_G(A_P) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$I_G(A_N) = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.375$$

Ganancia de la regla $Peso < 65$

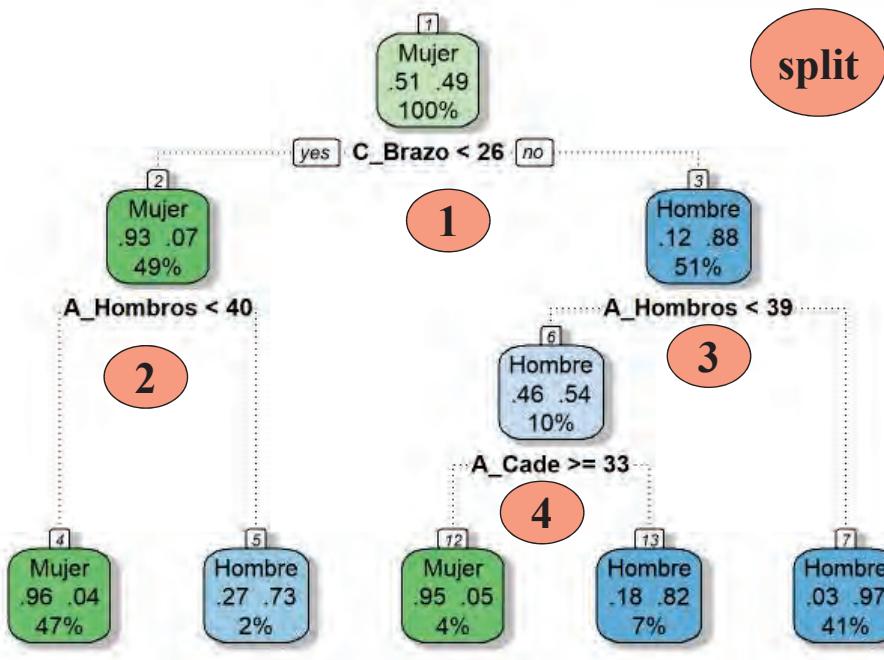
$$G(Peso < 65) = I_G(A) - [I_G(A_p) \cdot f_p + I_G(A_N) \cdot f_N]$$



Árbol

Con todas las variables

```
t2 = rpart(Sexo ~ ., cp=.01, data =dat)
#rpart.plot(t2)
fancyRpartPlot(t2, caption = NULL)
```



Complejidad del árbol

```
printcp(t2)

##
## Classification tree:
## rpart(formula = Sexo ~ ., data = dat, cp = 0.01)
##
## Variables actually used in tree construction:
## [1] A_Cade    A_Hombros C_Brazo
##
## Root node error: 247/507 = 0.48718
##
## n= 507
##
##      CP nsplits rel error  xerror     xstd
## 1 0.805668      0   1.00000 1.00000 0.045565
## 2 0.034416      1   0.19433 0.22672 0.028575
## 3 0.020243      3   0.12551 0.23077 0.028797
## 4 0.010000      4   0.10526 0.20648 0.027420
```

Clase Mayoritaria MUJER- error 0.49



Clase Mayoritaria
MUJER- error 0.07

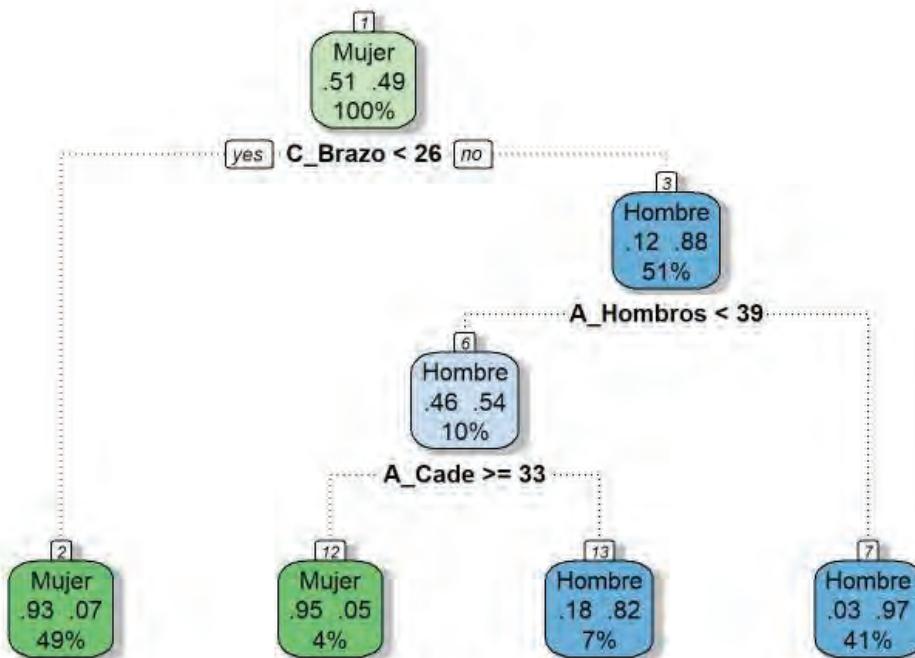
Clase Mayoritaria
HOMBRE- error 0.12

$$\text{ERROR} = 0.07 \times 0.49 + 0.12 \times 0.51 = 0.0955$$

$$\text{ERROR REL.} = 0.0955/0.49 = 0.1949$$

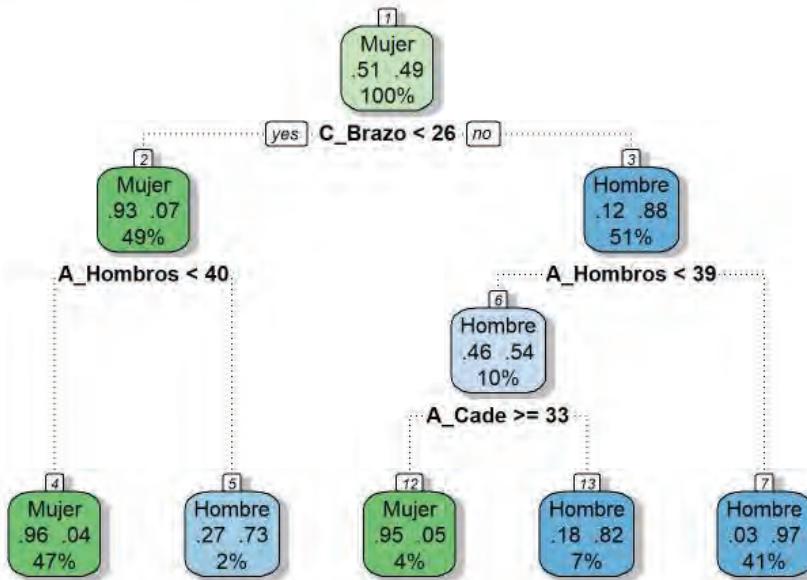
Poda de un árbol

```
t3 <- prune(t2, cp = 0.022)
fancyRpartPlot(t3, caption = NULL)
```



Estrategia: empezar con cp muy bajo

```
t3 = rpart(Sexo ~ ., cp=.0001,data =dat)
#rpart.plot(t1)
fancyRpartPlot(t2, caption = NULL)
```



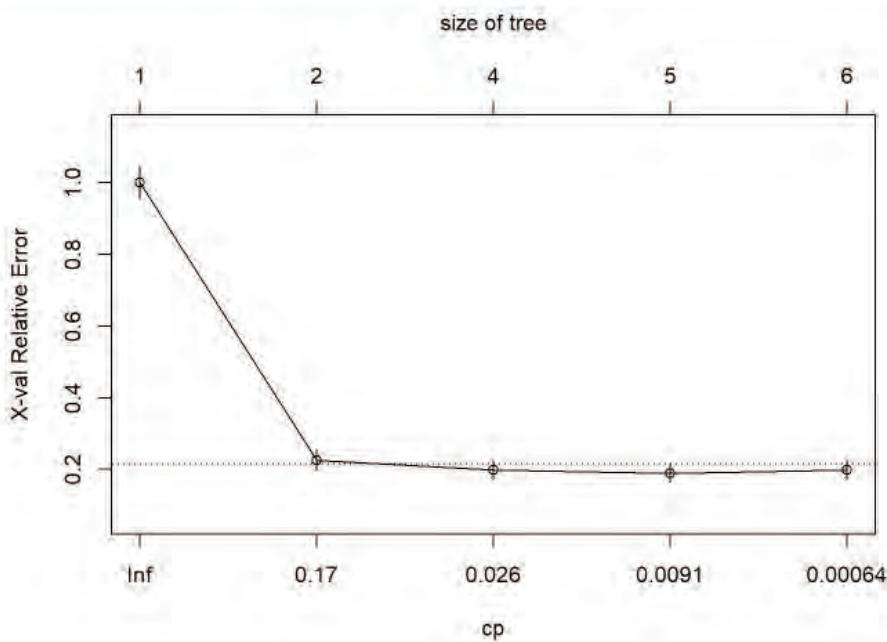
Complejidad del árbol

```
printcp(t3)
```

```
##  
## Classification tree:  
## rpart(formula = Sexo ~ ., data = dat, cp = 1e-04)  
##  
## Variables actually used in tree construction:  
## [1] A_Cade    A_Hombros Altura    C_Brazo  
##  
## Root node error: 247/507 = 0.48718  
##  
## n= 507  
##  
##      CP nsplitt rel.error xerror      xstd  
## 1 0.8056680     0  1.00000 1.00000 0.045565  
## 2 0.0344130     1  0.19433 0.22672 0.028575  
## 3 0.0202429     3  0.12551 0.19838 0.026936  
## 4 0.0040486     4  0.10526 0.19028 0.026438  
## 5 0.0001000     5  0.10121 0.19838 0.026936
```

Gráfico de coeficientes de complejidad

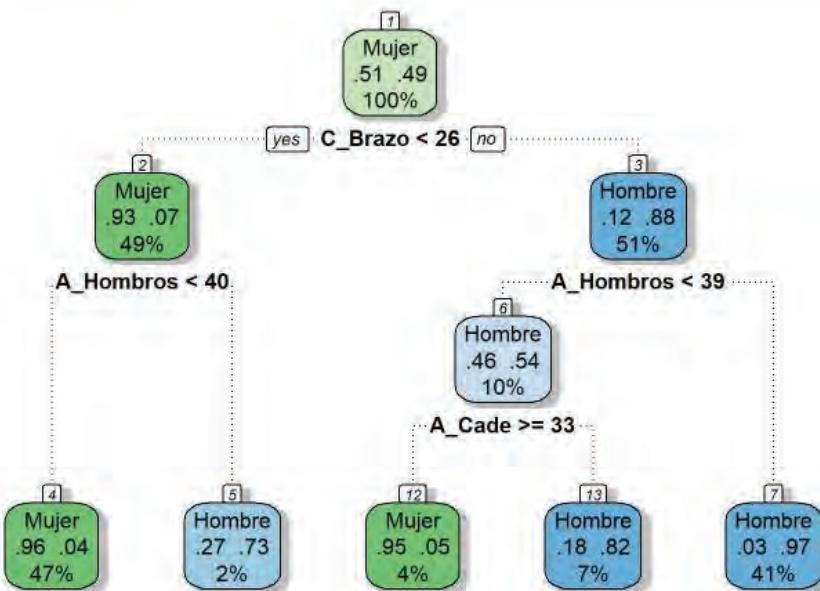
```
plotcp(t3)
```



Estrategia: Podar el árbol

Cuantas menos ramas mejor

```
t4 <- prune(t3, cp = 0.015)  
fancyRpartPlot(t4, caption = NULL)
```



Predicción con un árbol

```
ypred = predict(t4)
set.seed(198)
sel = sample(1:507,5)
ypred[sel,]

##          Mujer      Hombre
## [1,] 0.95780591 0.04219409
## [2,] 0.95780591 0.04219409
## [3,] 0.18181818 0.81818182
## [4,] 0.02898551 0.97101449
## [5,] 0.02898551 0.97101449

Sexop = factor(ypred[,1]<.5,labels=c("M","H"))
tabla1=table(Real=dat$Sexo,Pred=Sexop)
addmargins(tabla1)

##          Pred
## Real      M   H Sum
## Mujer    245 15 260
## Hombre    11 236 247
## Sum      256 251 507

(aciertos = (tabla1[1,1] + tabla1[2,2])/sum(tabla1))*100

## [1] 94.87179

(errores = (tabla1[1,2] + tabla1[2,1])/sum(tabla1))*100

## [1] 5.128205
```

```
tabla2 = prop.table(tabla1,1)*100
print(tabla2,digits=2)
```

```
##          Pred
## Real      M   H
## Mujer    94.2 5.8
## Hombre    4.5 95.5
```

Random Forest (Clasificación)

```
rl = randomForest(Sexo ~ C_Cintura + C_Pecho, cp=.001,data =dat)
rl

##
## Call:
##  randomForest(formula = Sexo ~ C_Cintura + C_Pecho, data = dat,      cp = 0.001)
##              Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##          OOB estimate of error rate: 14.6%
## Confusion matrix:
##             Mujer Hombre class.error
## Mujer     221     89  0.1500000
## Hombre     35     212  0.1417004
```

Dos variables

```
r2 = randomForest(Sexo ~ ., cp=.001,data =dat,mtry=5)
r2
```

```
##
## Call:
##  randomForest(formula = Sexo ~ ., data = dat, cp = 0.001, mtry = 5)
##              Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##          OOB estimate of error rate: 3.94%
## Confusion matrix:
##             Mujer Hombre class.error
## Mujer     251      9  0.03461538
## Hombre     11     236  0.04453441
```

Todas las variables

Train and test

```
set.seed(12456)
sel = sample(1:507, size=400)
train = dat[sel,]
test = dat[-sel,]
ttl = rpart(Sexo~., cp=.001, data=train)
printcp(ttl)

##
## Classification tree:
## rpart(formula = Sexo ~ ., data = train, cp = 0.001)
##
## Variables actually used in tree construction:
## [1] Altura    C_hombros C_Muneca C_Muslo
##
## Root node error: 194/400 = 0.485
##
## n= 400
##
##          CP nsplit rel error xerror      xstd
## 1 0.824742     0  1.000000 1.00000 0.051523
## 2 0.023196     1  0.175259 0.17526 0.028751
## 3 0.001000     5  0.082474 0.17526 0.028751
```

Errores de predicción (test-árbol)

```
ypred = predict(ttl,newdata = test)
test$Sexop = factor(ypred[,1]<.5,labels=c("M","H"))
solt=table(test$Sexo,test$Sexop)
err = matrix(0,2,1)
err[1]=solt[1,2]/(solt[1,1]+solt[1,2])
err[2]=solt[2,1]/(solt[2,1]+solt[2,2])
err = round(err,3)
cbind(addmargins(solt,2),error=err)

##
##      M  H Sum
## Mujer 45  9 54 0.167
## Hombre 2 51 53 0.038

error=round(1000*(solt[2,1]+solt[1,2])/sum(solt))/10
paste("error de clasificación", error, "%")

## [1] "error de clasificación 10.3 %"
```

Errores de predicción (test- Random Forest)

```
tr1 = randomForest(Sexo ~ ., cp=.001,data =train,importance=TRUE)
ypred2=predict(tr1,newdata = test)
Sexop2=ypred2
(solt2=table(test$Sexo,Sexop2))

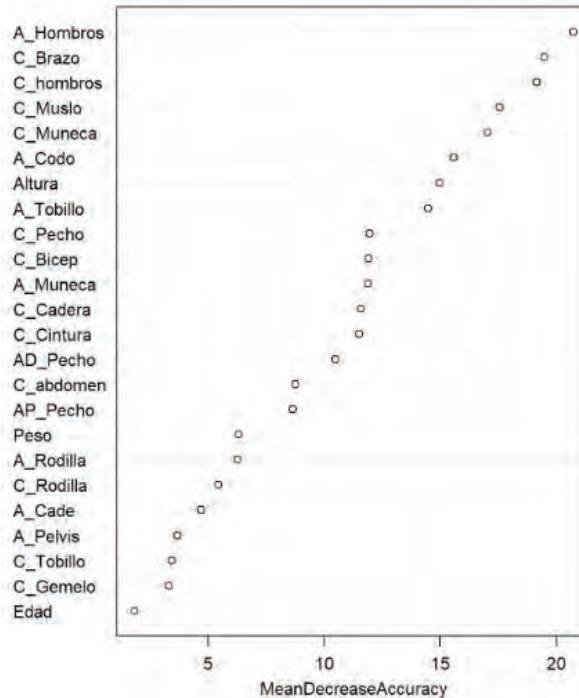
##          Sexop2
##          Mujer Hombre
## Mujer      51      3
## Hombre      0     53

error2=round(100*(solt2[2,1]+solt2[1,2])/sum(solt2),2)
paste("error de clasificación", error2, "%")

## [1] "error de clasificación 2.6 %"
```

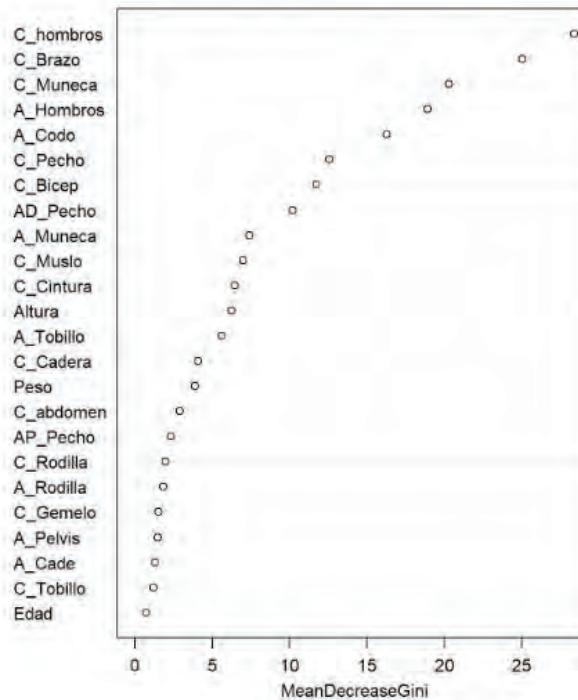
Importancia de las variables

```
varImpPlot(tr1)
```



Importancia de las variables

```
varImpPlot(tr1)
```



Penguins: Árboles de clasificación y Random Forest

Penguins y Lirios

Dos ejemplos fáciles de clasificar.

Penguins

```
library(rpart)
library(rpart.plot)
library(rattle)
library(randomForest)
library(treeheatr)
library(MASS)
library(factoextra)
library(car)
data(penguins)
names(penguins)

## [1] "species"           "island"              "culmen_length_mm"
## [4] "culmen_depth_mm"   "flipper_length_mm"  "body_mass_g"
## [7] "sex"
```

Data of three different species of penguins.

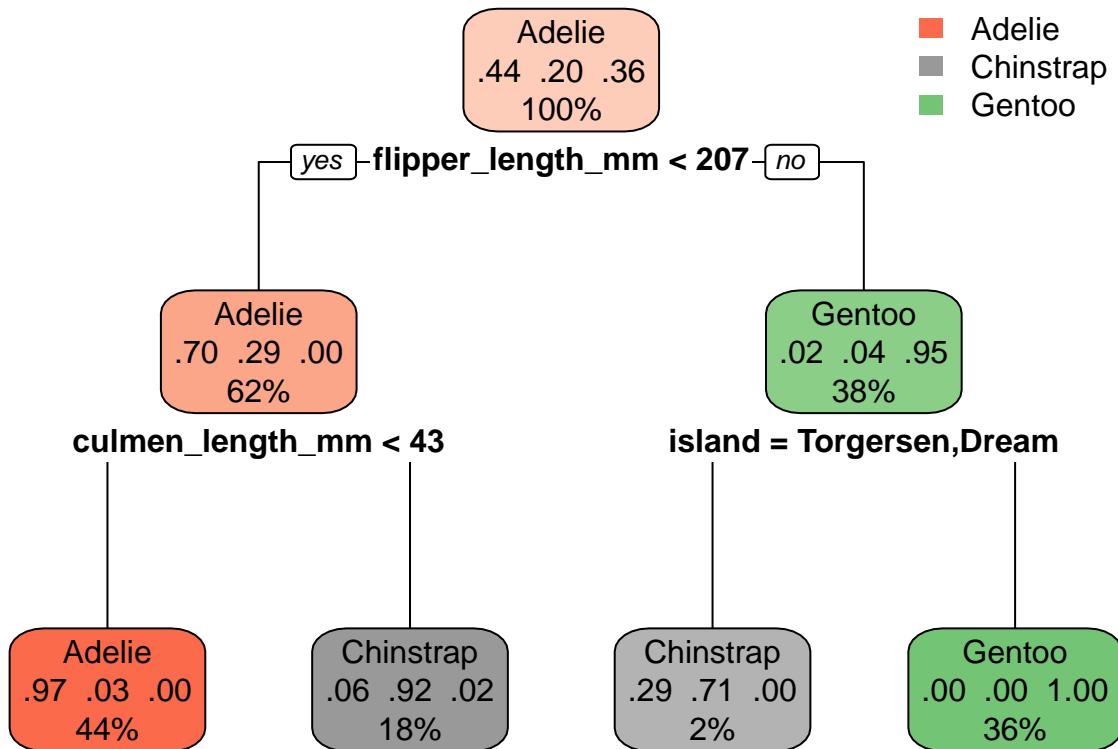
Variables:

- “species” : Adelie, Chinstrap, Gentoo
- “island” : Torgersen, Biscoe, Dream
- “culmen_length_mm” : longitud del pico
- “culmen_depth_mm” : anchura del pico
- “flipper_length_mm” : longitud de la aleta
- “body_mass_g” : peso
- “sex” : sexo

Description Collected and made available by Dr. Kristen Gorman and the Palmer Station, Antarctica LTER, a member of the Long Term Ecological Research Network.

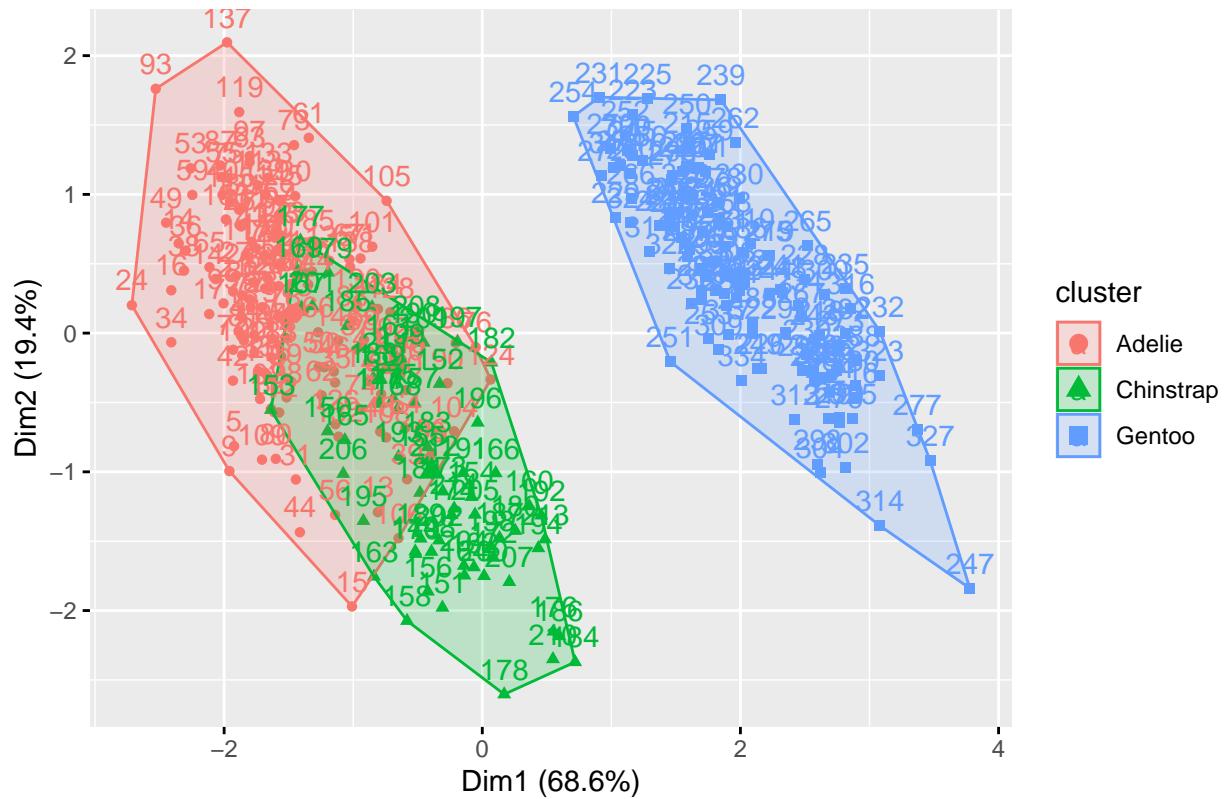
A data frame with 344 observations and 7 variables: species, island, culmen_length_mm, culmen_depth_mm, flipper_length_mm, body_mass_g and sex.

```
t1 = rpart(species ~ ., data = penguins, cp=0)
rpart.plot(t1)
```



```
casos = complete.cases(penguins)
dat = as.data.frame(penguins[casos,])
fviz_cluster(list(cluster=dat$species,data=dat[,3:6]))
```

Cluster plot



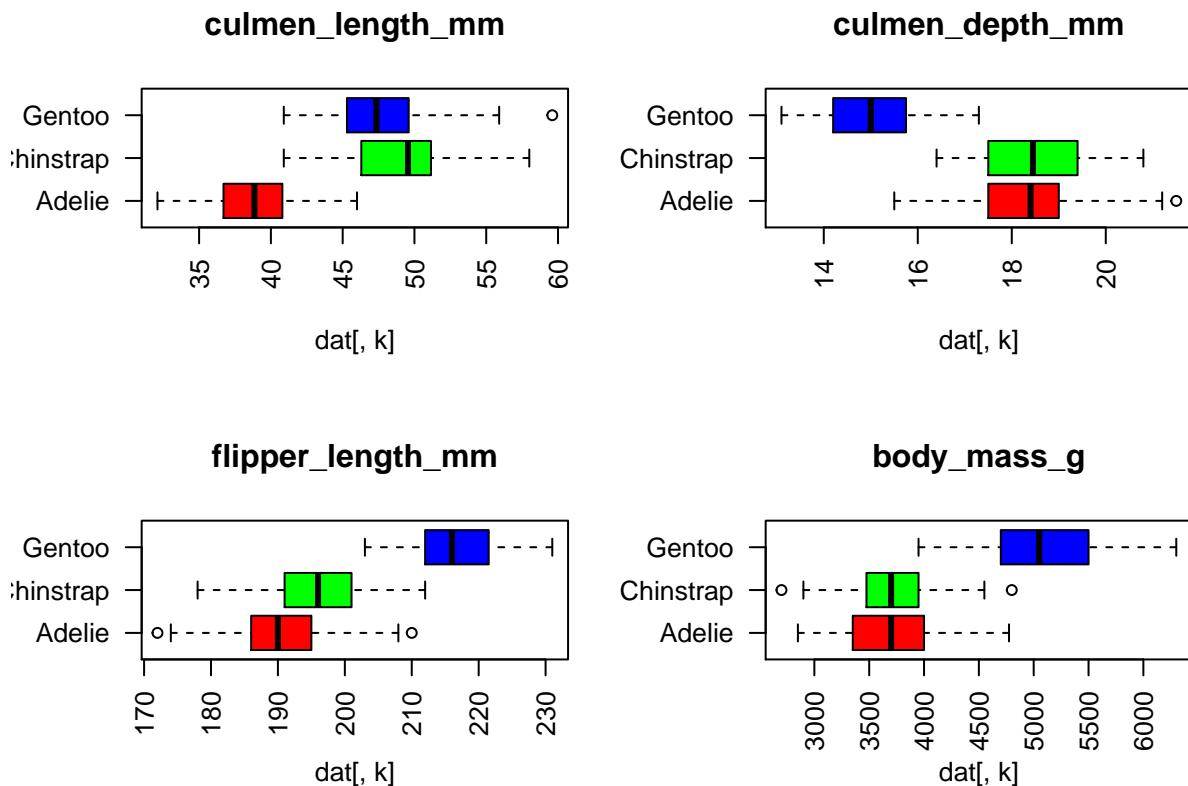
```
source("prinfact.R")
sol = prinfact(dat[,3:6], 2)
sol$loadings
```

```
##                               Comp 1      Comp 2 communality uniqueness
## culmen_length_mm   0.7511246  0.530984087  0.8461322 0.15386779
## culmen_depth_mm   -0.6620375  0.701063736  0.9297841 0.07021595
## flipper_length_mm  0.9557309  0.004516381  0.9134420 0.08655795
## body_mass_g        0.9109137  0.066942533  0.8342450 0.16575501
```

```
table(dat$species, dat$island)
```

```
##
##                               Torgersen Biscoe Dream
## Adelie                  47     44    55
## Chinstrap                0      0    68
## Gentoo                  0    120     0
```

```
par(mfrow=c(2,2))
for (k in 3:6) {
  boxplot(dat[,k] ~ dat[,1], horizontal = T, col=rainbow(3),
          main = colnames(dat)[k], las=2, ylab=NULL)
}
```



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

```
r1 = randomForest(species~., data = dat)
r1
```

```
##
## Call:
##   randomForest(formula = species ~ ., data = dat)
##   Type of random forest: classification
##   Number of trees: 500
##   No. of variables tried at each split: 2
##
##       OOB estimate of error rate: 0.9%
## Confusion matrix:
##             Adelie Chinstrap Gentoo class.error
## Adelie      146        0       0  0.00000000
## Chinstrap     3       65       0  0.04411765
## Gentoo       0       0      120  0.00000000
```

Fuente de los datos

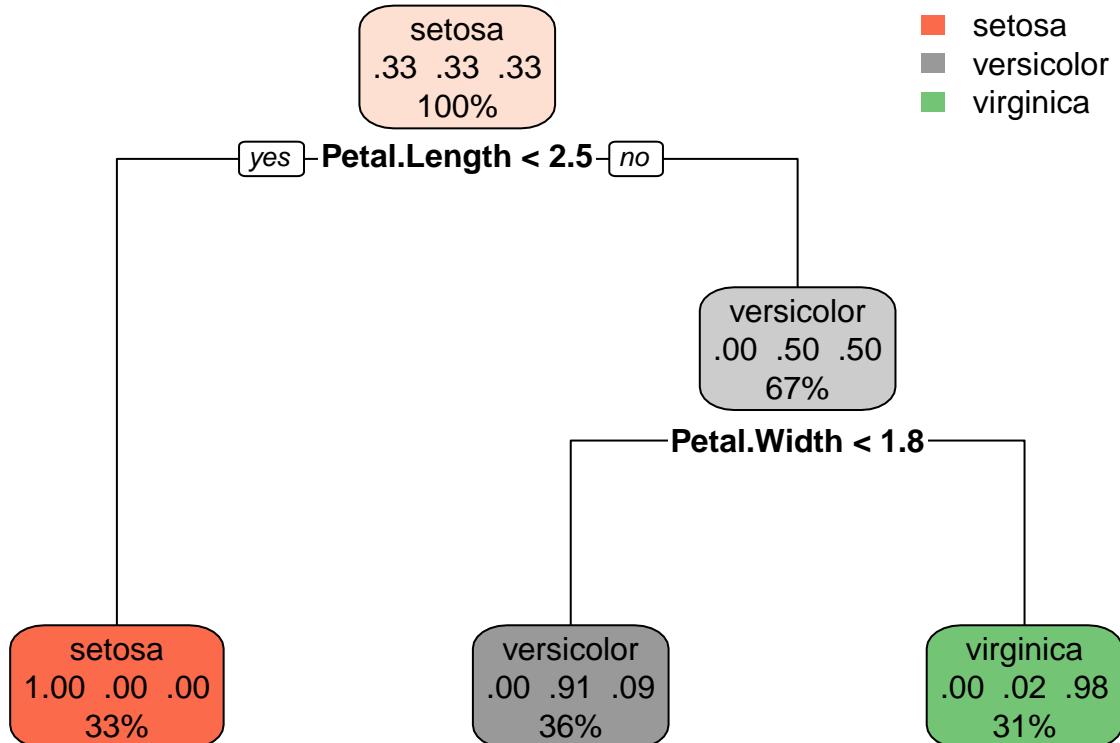
Gorman KB, Williams TD, Fraser WR (2014). Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus Pygoscelis). PLoS ONE 9(3): e90081. doi:10.1371/journal.pone.0090081

Details Fetched from <https://github.com/allisonhorst/penguins>.

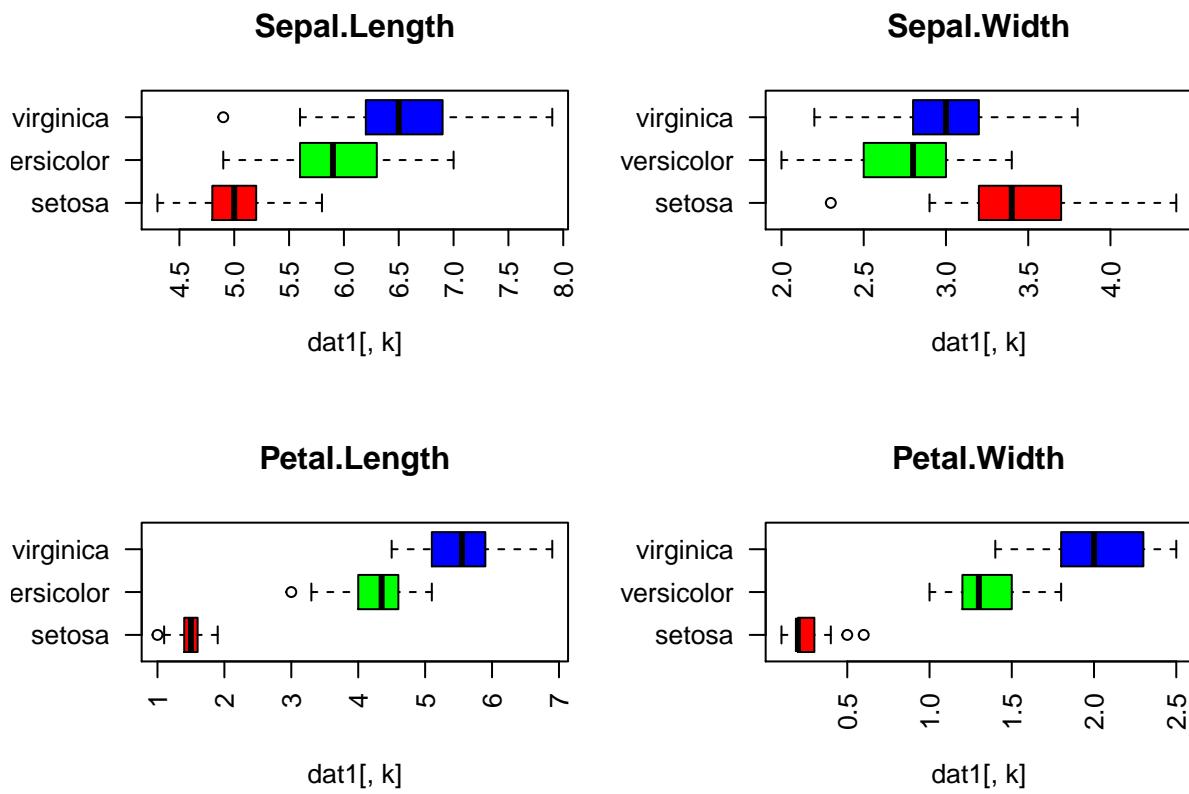
Lirios

```
dat1 = read.table("lirios.txt", header=T)
dat1$Species = factor(dat1$Species)
```

```
t2 = rpart(Species ~ ., data = dat1)
rpart.plot(t2)
```

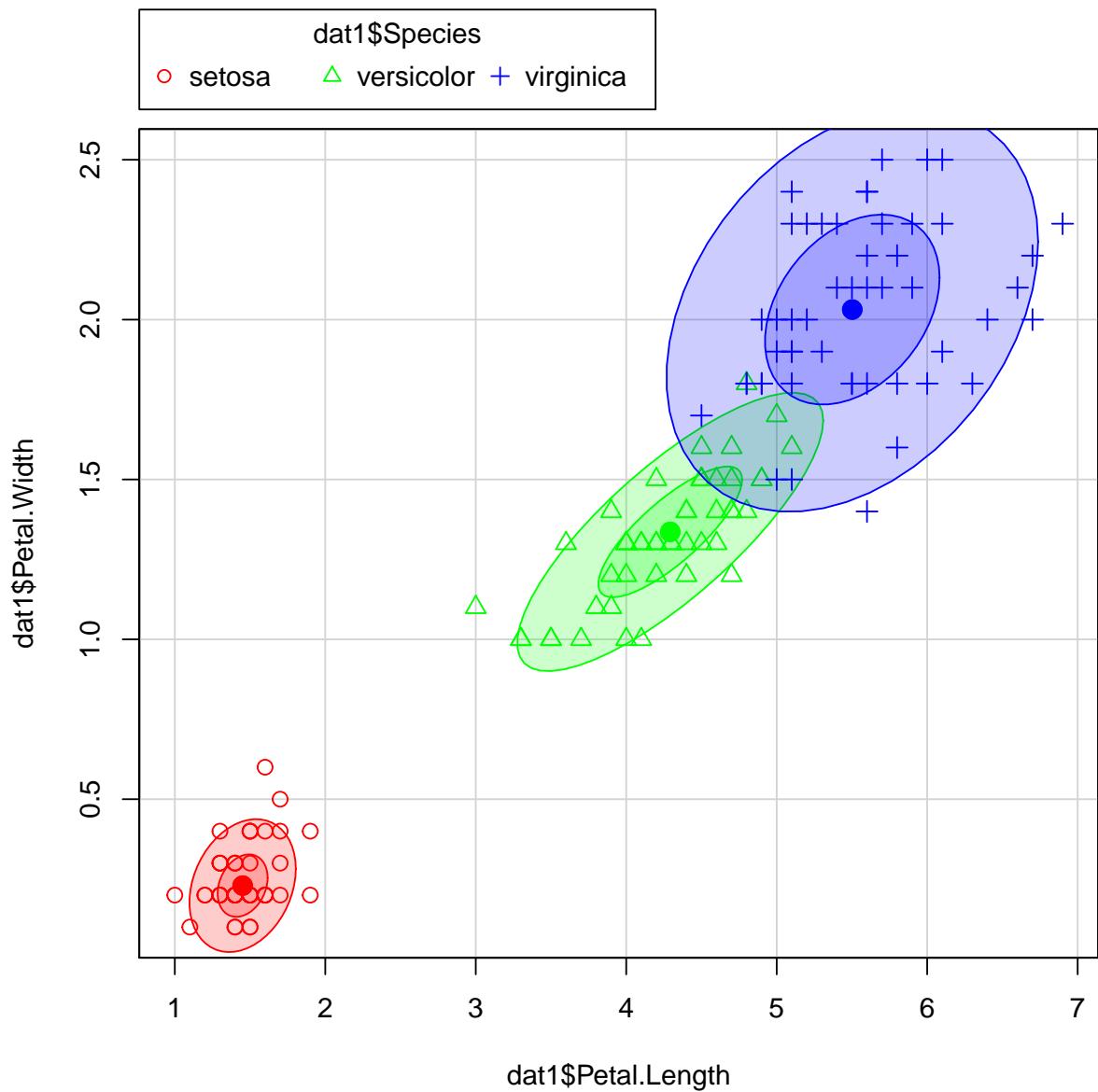


```
par(mfrow=c(2,2))
for (k in 1:4) {
  boxplot(dat1[,k] ~ dat1[,5], horizontal = T, col=rainbow(3),
           main = colnames(dat1)[k], las=2, ylab=NULL)
}
```

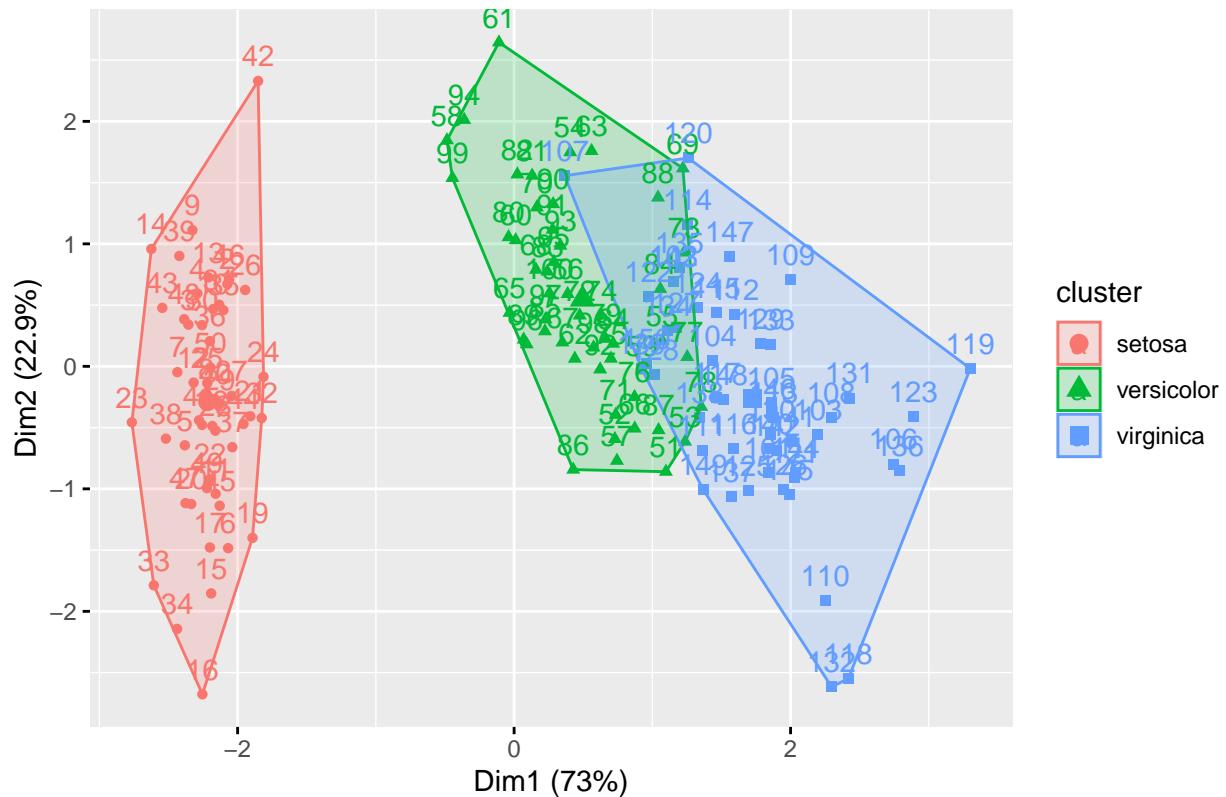


```
par(mfrow=c(1,1))
```

```
scatterplot(dat1$Petal.Length,dat1$Petal.Width,
           groups = dat1$Species,
           regLine = F, smooth = F,
           ellipse = T,
           col=c("red","green","blue"), cex = 1.2)
```



Cluster plot



```
r2 = randomForest(Species ~ ., data = dat1)
r2
```

```
##
## Call:
##   randomForest(formula = Species ~ ., data = dat1)
##   Type of random forest: classification
##   Number of trees: 500
##   No. of variables tried at each split: 2
##
##       OOB estimate of  error rate: 4.67%
## Confusion matrix:
##             setosa versicolor virginica class.error
## setosa      50        0        0     0.00
## versicolor    0       47        3     0.06
## virginica     0        4       46     0.08
```

TAREA 4 (Solución) : COVID 19

Jesus Juan

Covid 19

El repentino aumento de los casos de COVID-19 está ejerciendo una gran presión sobre los servicios de atención de la salud en todo el mundo. En la etapa actual, es vital una evaluación clínica rápida, precisa y temprana de la gravedad de la enfermedad. Para apoyar la toma de decisiones y la planificación logística en los sistemas de atención de la salud, este estudio aprovecha una base de datos de muestras de sangre de 461 pacientes infectados en la región de Wuhan (China) para identificar biomarcadores predictivos cruciales de la mortalidad de la enfermedad. Para ello, las herramientas de aprendizaje automático seleccionaron tres biomarcadores que predicen la mortalidad de pacientes individuales con una precisión superior al 90%: la deshidrogenasa láctica (LDH), los linfocitos y la proteína C reactiva de alta sensibilidad (hs-CRP). En particular, los niveles relativamente altos de LDH por sí solos parecen desempeñar un papel crucial para distinguir la gran mayoría de los casos que requieren atención médica inmediata. Este hallazgo es coherente con los conocimientos médicos actuales de que los altos niveles de LDH se asocian con la descomposición de tejidos que se produce en diversas enfermedades, incluidos trastornos pulmonares como la neumonía. En general, este documento sugiere una regla de decisión simple y operable para predecir rápidamente los pacientes de mayor riesgo, lo que permite priorizarlos y potencialmente reducir la tasa de mortalidad.

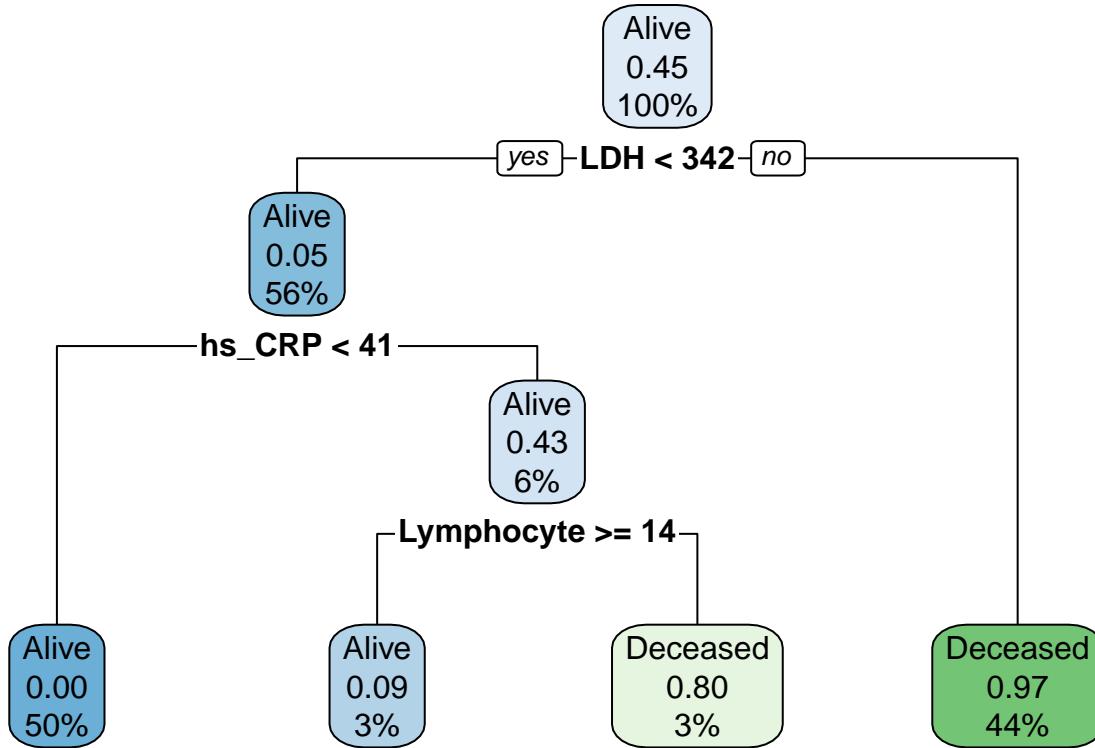
En el archivo “train_covid.txt” se encuentran los resultados de los análisis realizados a 351 pacientes con COVID para los tres biomarcadores “LDH”, “hs_CRP” y “Lymphocyte” y si el paciente sobrevivió o no al contagio: “Outcome”. Se han seleccionado al azar para construir el modelo. En el archivo “test_covid.txt” se encuentra datos de 110 pacientes que se han reservado para evaluar el modelo.

```
library(rpart)
library(rpart.plot)
library(rattle)
library(randomForest)
train_covid = read.table("train_covid.txt", header=TRUE)
test_covid = read.table("test_covid.txt", header=TRUE)
train_covid$Outcome = factor(train_covid$Outcome, labels = c("Alive", "Deceased"))
test_covid$Outcome = factor(test_covid$Outcome, labels = c("Alive", "Deceased"))
```

Responde a las siguientes preguntas:

1. Construye y representa el árbol de clasificación utilizando la variable **Outcome** como variable respuesta y las restantes tres variables como variables explicativas. (Nota.- la función `fancyRpartPlot()` se encuentra en el paquete `rattle` y la función `rpart.plot()` en el paquete `rpart.plot`) (utiliza el parámetro de complejidad `cp = 0.0001`). Explica el árbol detalladamente el árbol. Indica cuantos nodos finales tiene.

```
t1 = rpart(Outcome~., data = train_covid, cp=0.0001)
rpart.plot(t1)
```



El árbol tiene tres particiones y cuatro nodos finales. Los nodos finales son 3, 5, 6 y 7. El número de observaciones en cada nodo está en `t1$where`

```
table(t1$where)
```

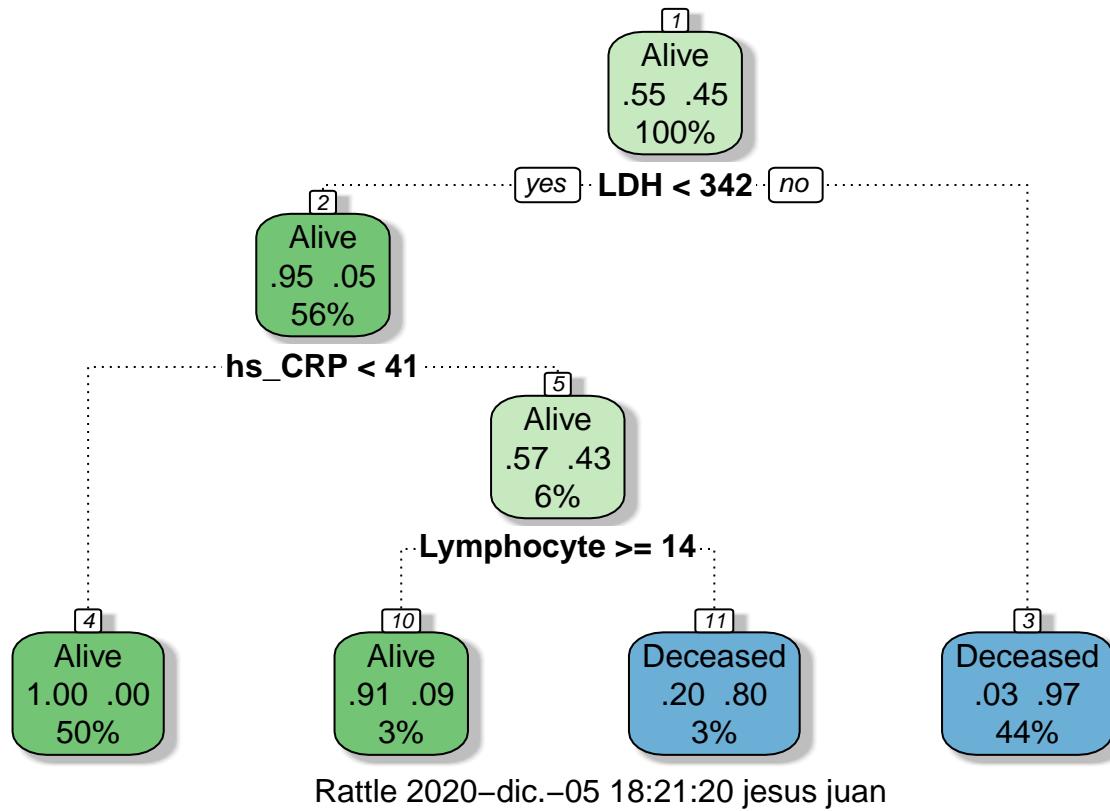
```
##
##      3      5      6      7
## 176   11   10  154
```

```
(tab0 = table(train_covid$Outcome, t1$where))
```

```
##
##            3   5   6   7
##  Alive    176 10   2   4
##  Deceased 0    1   8 150
```

Otra representación del árbol más fácil de interpretar se consigue con la función `fancyRpartPlot()` del paquete `rattle`.

```
fancyRpartPlot(t1)
```



Partimos de un conjunto de 351 observaciones (NODO 1) clasificadas en dos grupos: Alive and Deceased. El nodo inicial NODO1 está formado por los 351 pacientes, 45% de ellos murieron debido a la enfermedad y 55% sobrevivieron (192).

```
table(train_covid$Outcome)
```

```
##  
##      Alive Deceased  
##      192      159
```

- La primera partición (split) utiliza la regla $LDH < 342$. Crea dos nodos, NODO 3, corresponde al 44% de los pacientes tenían LDH igual o superior a 342, de este grupo murieron el 97% y sólo sobrevivió el 3%. En el NODO 2 están los pacientes con $LDH < 342$ (el 56%), sobrevivieron el 95% y perecieron el 5%.
- La segunda partición se realiza en el NODO 2, utilizando la regla $hs_CRP < 41$. Los que tuvieron un valor del biomarcador hs_CRP por debajo de 41 (NODO 4) forman el nodo 4. Son el 50% de la muestra y todos ellos sobrevivieron. Los pacientes del NODO 2 con hs_CRP igual o superior a 41 se encuentran en el NODO 5. Son el 6% de la muestra inicial. En este nodo hay una proporción muy alta, tanto de fallecidos (43%) como de sobrevivientes (57%).
- La tercera partición la realiza en el NODO 5 con el criterio $Lymphocyte \geq 14$. Del grupo de pacientes del NODO 5, la mitad están en el NODO 10 (3%) tenían Lymphocyte mayor o igual a 14, de este grupo sobrevivieron el 91% y perecieron el 9%. La otra mitad (3%) tenían valor de Lymphocyte menor que 14 y forman el nodo 11. En este nodo sobrevivieron el 20% y fallecieron el 80%.

2.- Utiliza la instrucción `printcp()` y explica como se obtienen los valores siguientes:

- (a) Root node error
- (b) La columna CP
- (c) La columna rel error

```
printcp(t1)
```

```
##  
## Classification tree:  
## rpart(formula = Outcome ~ ., data = train_covid, cp = 1e-04)  
##  
## Variables actually used in tree construction:  
## [1] hs_CRP      LDH       Lymphocyte  
##  
## Root node error: 159/351 = 0.45299  
##  
## n= 351  
##  
##          CP nsplit rel error  xerror     xstd  
## 1 0.918239      0  1.000000 1.000000 0.058654  
## 2 0.018868      1  0.081761 0.09434  0.023832  
## 3 0.000100      3  0.044025 0.09434  0.023832
```

De las dos clases de la variable **Outcome** utilizaremos **Alive** como la referencia.

El nodo inicial tiene mayoritariamente **Alive**, de manera que si asignamos a todos los pacientes el nivel **Alive** acertaremos con probabilidad 0.54701 y nos equivocaremos 0.45299. Es lo conocido como **Root node error** = 0.45299

```
table(train_covid$Outcome,train_covid$LDH<342)
```

```
##  
##          FALSE TRUE  
## Alive      4 188  
## Deceased 150    9
```

La primera columna de la tabla corresponde al NODO 3 y la segunda columna al NODO 2. Las observaciones del NODO 2 son declarados **Alive** y los del NODO 3, **Deceased**. El error con esta bifurcación es 13/351 = 0.037. El error relativo es

$$\text{error rel} = \frac{0.037}{0.45299} = \frac{13}{159} = 0.081761$$

Se denomina **Complexity parameter (cp)** a

$$cp = 1 - 0.081761 = 0.918239$$

Con tres splits (cuatro nodos finales) se tiene la siguiente tabla de errores

```
(tab0 = table(train_covid$Outcome,t1$where))
```

```
##          3   5   6   7
## Alive    176 10  2   4
## Deceased 0    1   8 150
```

El número de pacientes mál clasificados son 7, el error relativo es $7/159 = 0.044025$ y el parámetro de complejidad es

$$cp = \frac{0.081761 - 0.044025}{2} = 0.018868$$

El numerador 2 en este caso es debido a que el paso de la fila 2 a la 3 de la tabla aumentan en 2 el número de splits.

El árbol obtenido debe verificar las condiciones de control de árbol (número máximo de split, número mínimo de observaciones en los nodos, etc. Todo esto se controla en `rpart.control()`)

Las columnas `xerror` y `xstd` corresponde a la estimación del error y su desviación típica obtenido por validación cruzada. El interesado puede encontrar información en la documentación del paquete `rpart`.

3. Utiliza los datos del archivo `test_covid.txt` para evaluar el modelo. Utiliza `predict()` para predecir el **Outcome** de los datos e indica el % de aciertos y % fallos

```
yp = predict(t1,newdata = test_covid)
Outcome_pred = factor(yp[,1] < 0.5, labels = c("Alive","Deceased"))
(tab=table(test_covid$Outcome, Outcome_pred))
```

```
##          Outcome_pred
##                  Alive Deceased
## Alive        92       5
## Deceased     0      13
```

Otra forma de realizar predicciones con `predict()` y la opción `type = 'class'` es:

```
pred_class = predict(t1,newdata = test_covid, type = 'class')
(tab_bis=table(test_covid$Outcome,pred_class ))
```

```
##          pred_class
##                  Alive Deceased
## Alive        92       5
## Deceased     0      13

aciertos = (tab[1,1]+tab[2,2])/sum(tab)
fallos = (tab[1,2]+tab[2,1])/sum(tab)
paste('Fallos = ', round(100*fallos,1), '%')

## [1] "Fallos = 4.5 %"
```

```
paste('Aciertos = ', round(100*aciertos,1), '%')
```

```
## [1] "Aciertos = 95.5 %"
```

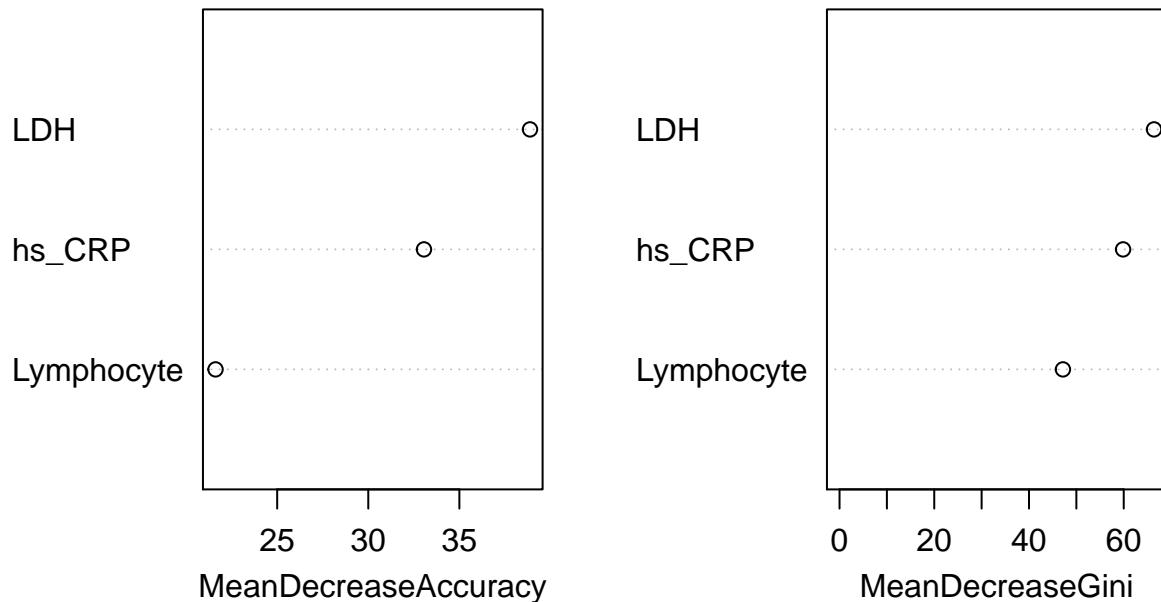
4.- Utiliza el modelo de *random forest* utilizando **Outcome** como variable respuesta y el resto como variables explicativas. Obtén el resumen del árbol y explica el significado de los valores que aparecen (OOB estimate of error rate y confusion matrix). Se recomienda utilizar la instrucción `set.seed(1122)` antes de ejecutar la instrucción de `randomForest()` para generar los mismos resultados). Realiza el gráfico de importancia de variables, para decidir qué variable es la más importante.

```
set.seed(1122)
r1 = randomForest(Outcome~., data = train_covid, importance=TRUE)
print(r1)
```

```
##
## Call:
##   randomForest(formula = Outcome ~ ., data = train_covid, importance = TRUE)
##   Type of random forest: classification
##   Number of trees: 500
##   No. of variables tried at each split: 1
##
##       OOB estimate of  error rate: 3.13%
## Confusion matrix:
##           Alive Deceased class.error
## Alive      185        7  0.03645833
## Deceased     4      155  0.02515723

varImpPlot(r1)
```

r1



Con los dos criterios, la variable más importante es LDH.

5. • Utiliza los datos del archivo `test_covid.txt` para evaluar el modelo de Random Forest. Utiliza `predict()` para predecir el **Outcome** de los datos e indica el % de aciertos y % fallos

```
pred = predict(r1, newdata = test_covid)
(tab2=table(test_covid$Outcome,pred))

##          pred
##          Alive Deceased
##    Alive      94       3
##    Deceased     0      13

aciertos = (tab2[1,1]+tab2[2,2])/sum(tab2)
fallos = (tab2[1,2]+tab2[2,1])/sum(tab2)
paste('Fallos = ', round(100*fallos,1), '%')

## [1] "Fallos = 2.7 %"

paste('Aciertos = ', round(100*aciertos,1), '%')

## [1] "Aciertos = 97.3 %"
```

e o de Casas

Jesús Juan

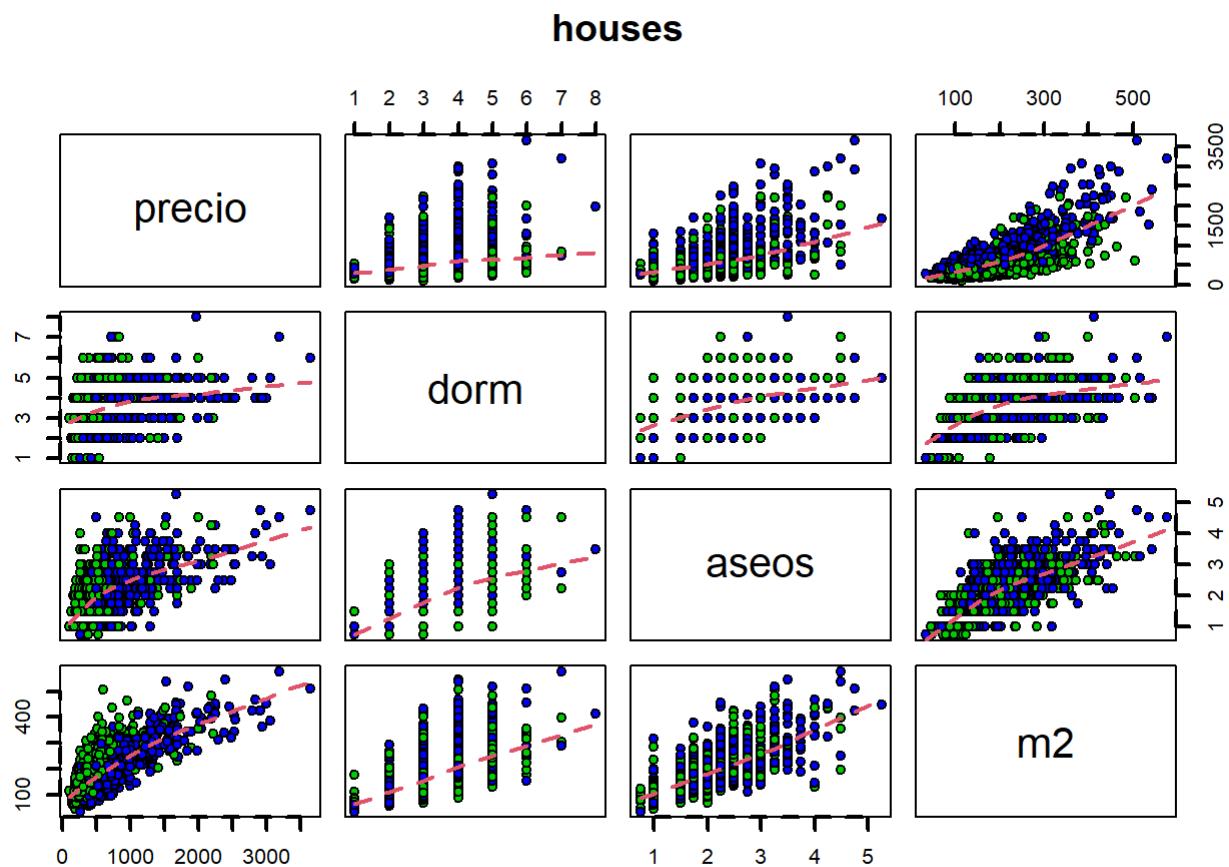
Áboles de Regresión

En el archivo "houses.txt" se proporciona el precio de venta en dólares de 1696 pisos en la ciudad de Seattle (USA), junto con el número de dormitorios (dorm), el número de baños o aseos (aseos), la superficie en metros cuadrados(m2), la variable cent que toma el valor 1 si el piso está en un barrio central de la ciudad y cero en el caso contrario y una puntuación (punt) que va de 1 a 4 que indica el estado del piso (1="mal", 2="regular", 3="bien" y 4="excelente").

Preparación de datos y paquetes

```
library(rpart)
library(rpart.plot)
library(randomForest)
library(plotmo)
dat <- read.table("houses.txt", header=TRUE )
dat$cent = factor(dat$cent,labels=c("No","Sí"))
dat$punt = factor(dat$punt, labels = c("mal","regular","bien","excelente"))
dat$precio = dat$precio/1000
n = dim(dat)[1]
sel = sample(1:n,size=round(n*.8))
train = dat[sel,]
test = dat[-sel,]
```

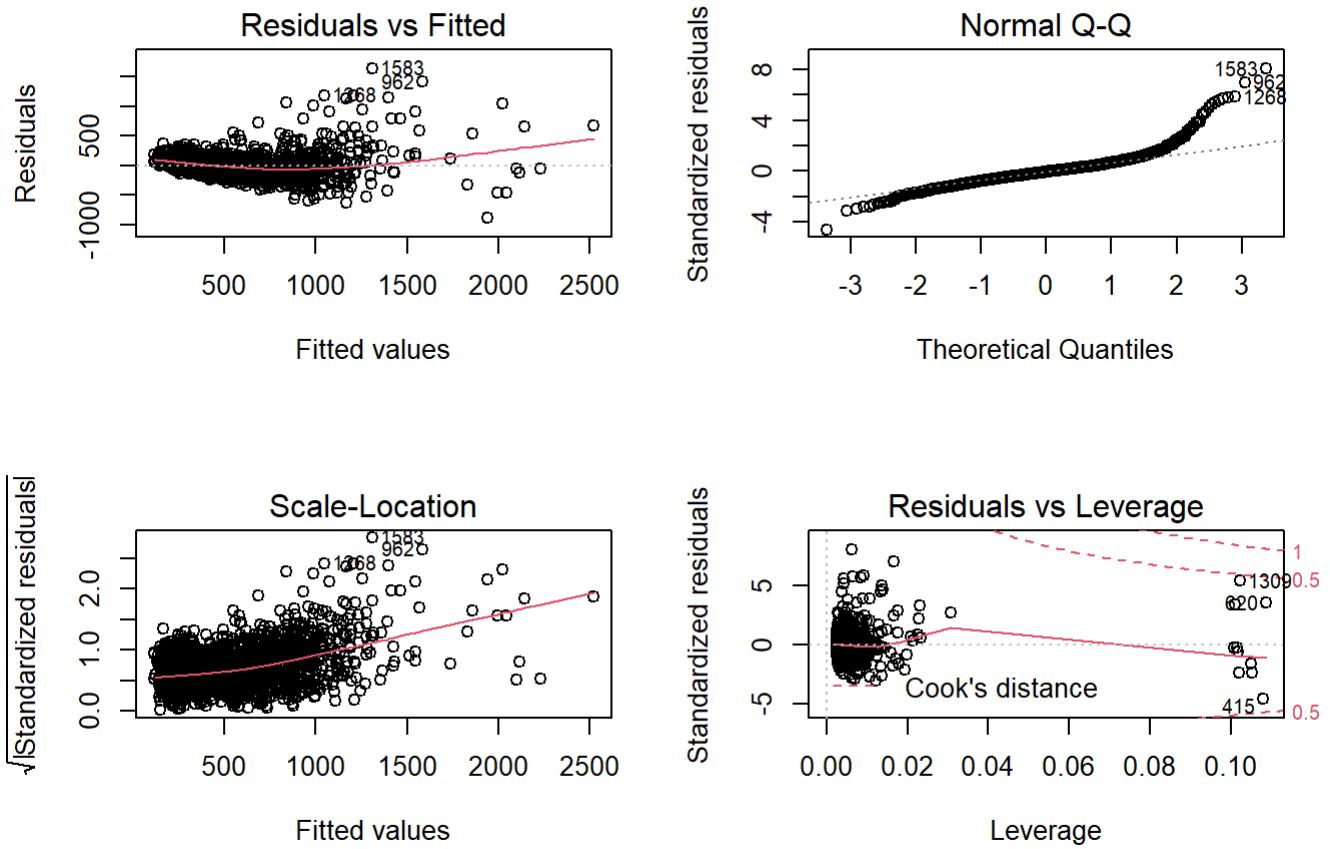
```
pairs(dat[,1:4], main="houses",
       panel=panel.smooth,
       pch=21,
       bg=c("green3","blue")[unclass(dat$cent)],
       lwd=2,lty=2)
```



```
m1 = lm(precio ~ ., data = train)
summary(m1)
```

```
##
## Call:
## lm(formula = precio ~ ., data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max 
## -889.02 -106.54 -12.22  77.35 1637.91 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 25.4179   23.9125   1.063   0.2880    
## dorm        -39.4057   7.9428  -4.961 7.90e-07 ***
## aseos         3.7860  11.8612   0.319   0.7496    
## m2           2.9778   0.1325  22.475 < 2e-16 ***
## centSí       230.1354  11.7333  19.614 < 2e-16 ***
## puntregular  -29.0205  17.4348  -1.665   0.0962 .  
## puntbien      126.8639  22.4638   5.647 1.98e-08 ***
## puntexcelente 804.7131  73.6361  10.928 < 2e-16 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 204.2 on 1349 degrees of freedom
## Multiple R-squared:  0.7143, Adjusted R-squared:  0.7128 
## F-statistic: 481.9 on 7 and 1349 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(m1)
```



```
m2 = lm(log(precio) ~ ., data =train)
summary(m2)
```

```

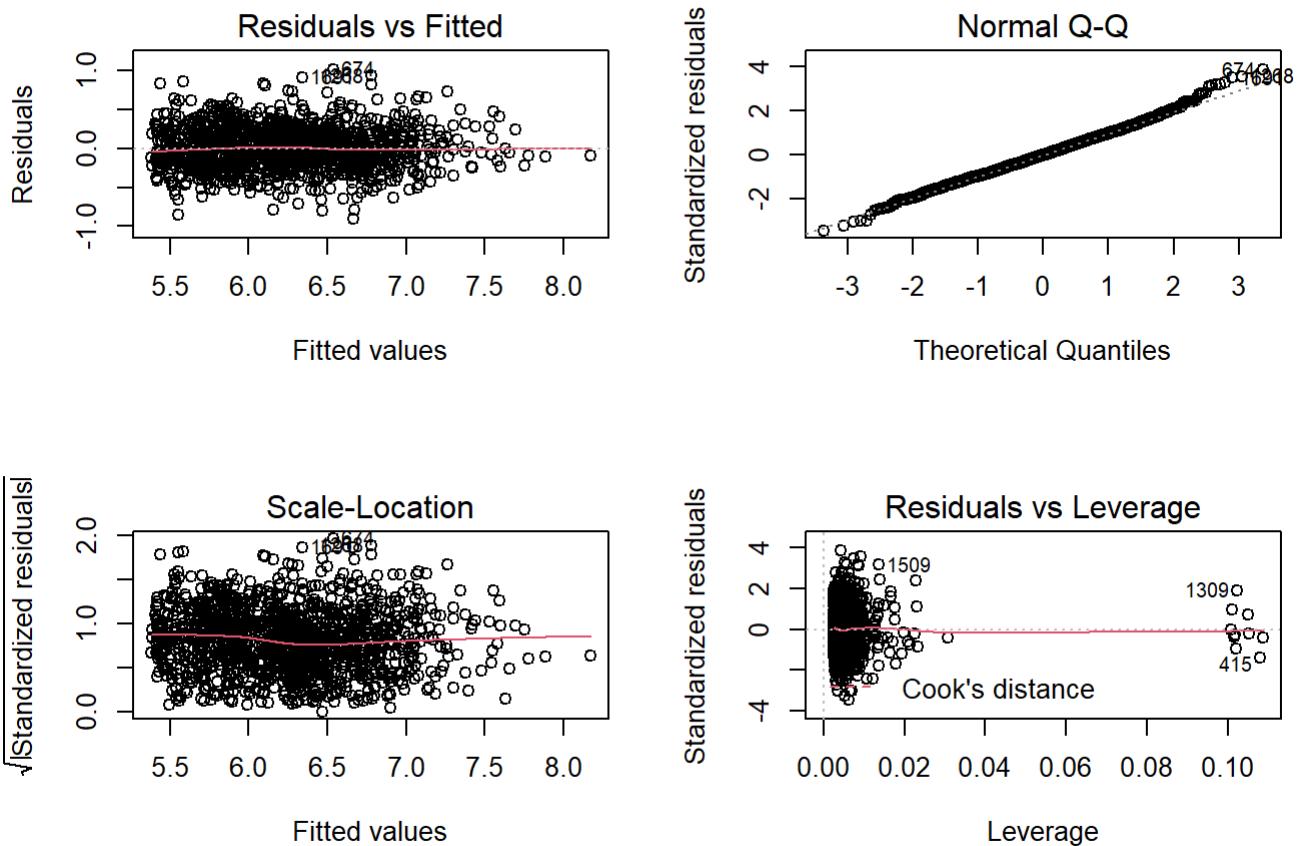
## 
## Call:
## lm(formula = log(precio) ~ ., data = train)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.90507 -0.17759 -0.00615  0.16983  1.01073
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.2375119  0.0306351 170.964 < 2e-16 ***
## dorm        -0.0355268  0.0101758  -3.491 0.000496 ***  
## aseos        0.0290492  0.0151958   1.912 0.056132 .    
## m2           0.0033694  0.0001697  19.850 < 2e-16 ***  
## centSí       0.4235643  0.0150319  28.178 < 2e-16 ***  
## puntregular  0.1656326  0.0223364   7.415 2.13e-13 *** 
## puntbien     0.4313249  0.0287792  14.987 < 2e-16 ***  
## puntexcelente 0.6840020  0.0943378   7.251 6.97e-13 *** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2616 on 1349 degrees of freedom
## Multiple R-squared:  0.7719, Adjusted R-squared:  0.7707 
## F-statistic: 652.1 on 7 and 1349 DF,  p-value: < 2.2e-16

```

```

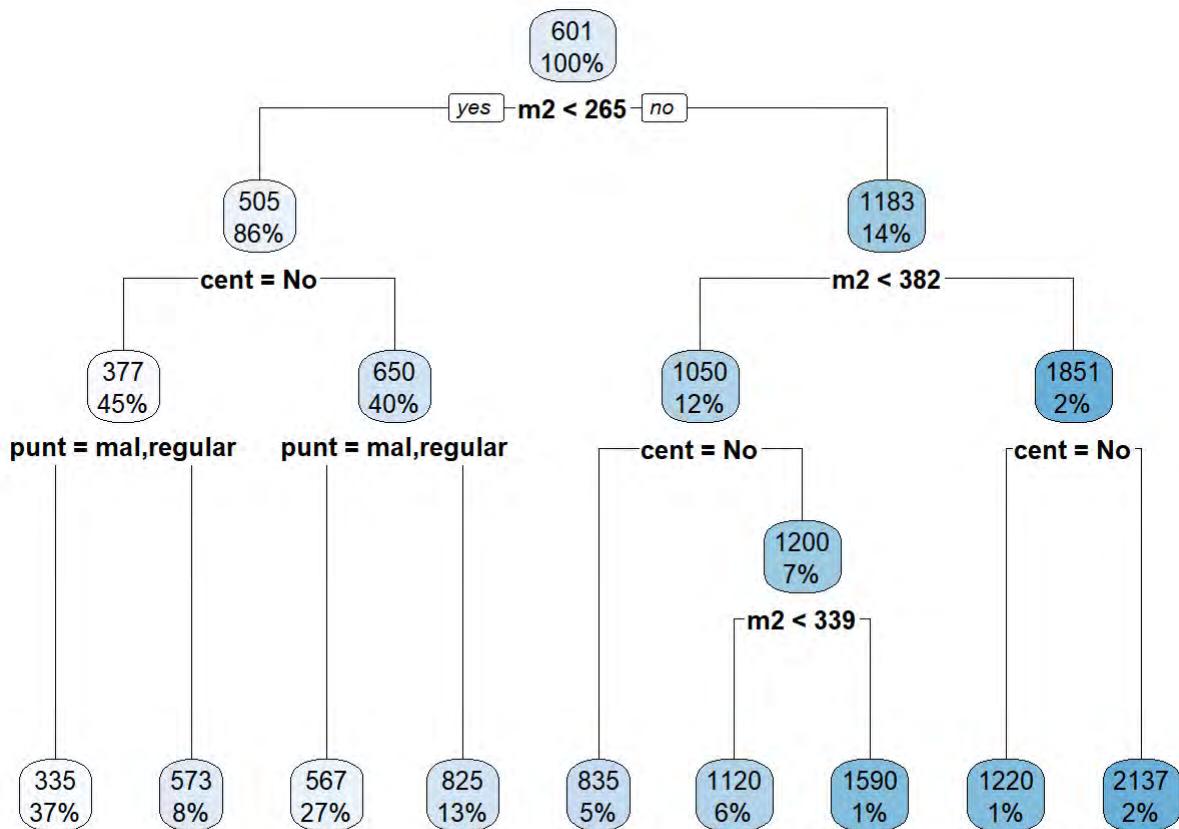
par(mfrow=c(2,2))
plot(m2)

```



Árboles de Regresión

```
t1 = rpart(precio ~., data =train)
rpart.plot(t1)
```



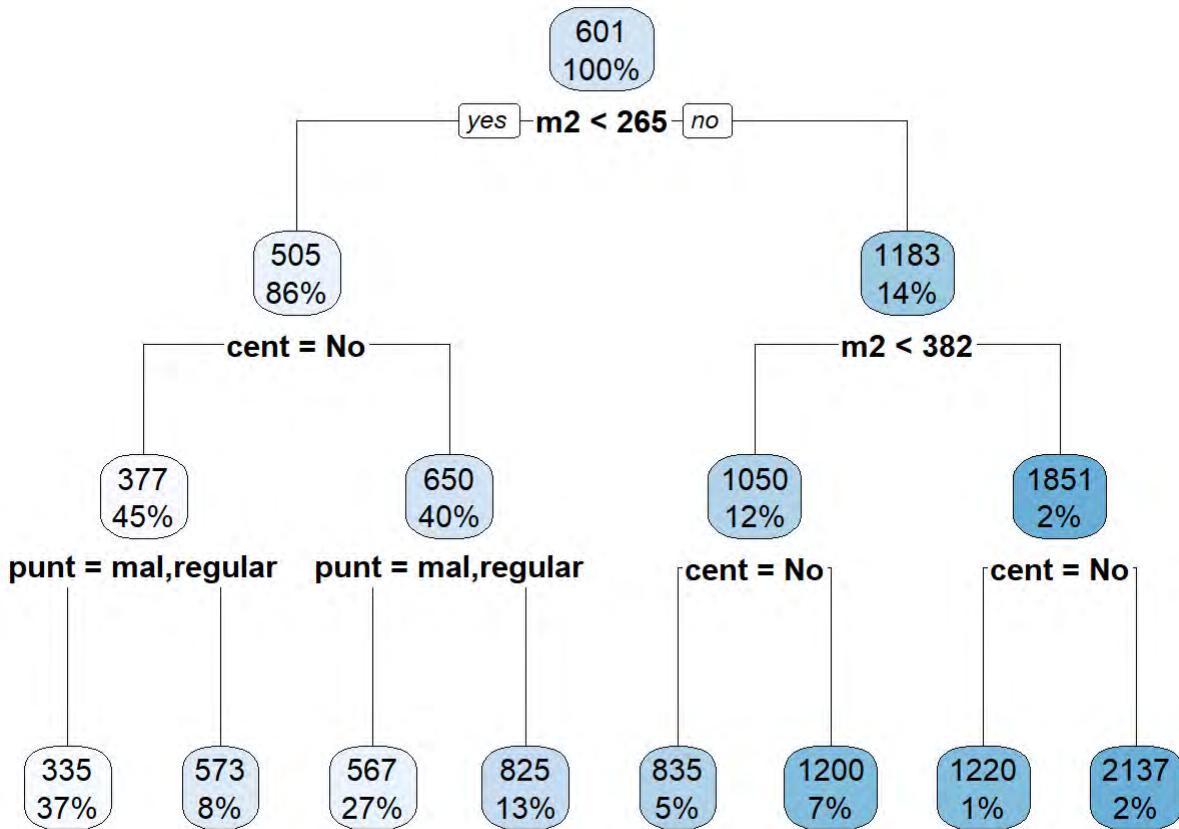
Complexity parameter (cp)

```
printcp(t1)
```

```
##
## Regression tree:
## rpart(formula = precio ~ ., data = train)
##
## Variables actually used in tree construction:
## [1] cent m2   punt
##
## Root node error: 196855935/1357 = 145067
##
## n= 1357
##
##          CP nsplit rel error  xerror     xstd
## 1 0.384569      0  1.00000 1.00202 0.089658
## 2 0.109360      1  0.61543 0.62980 0.048320
## 3 0.086914      2  0.50607 0.53170 0.049286
## 4 0.040568      3  0.41916 0.45379 0.038630
## 5 0.029398      4  0.37859 0.41876 0.038087
## 6 0.026253      5  0.34919 0.38690 0.035906
## 7 0.026146      6  0.32294 0.38303 0.035885
## 8 0.014861      7  0.29679 0.34464 0.033605
## 9 0.010000      8  0.28193 0.32364 0.030259
```

Podar el árbol

```
t2 = prune(t1,cp=0.02)
rpart.plot(t2)
```



Random Forest

```
r1 = randomForest(log(precio) ~ ., data = train, importance=TRUE)
print(r1)
```

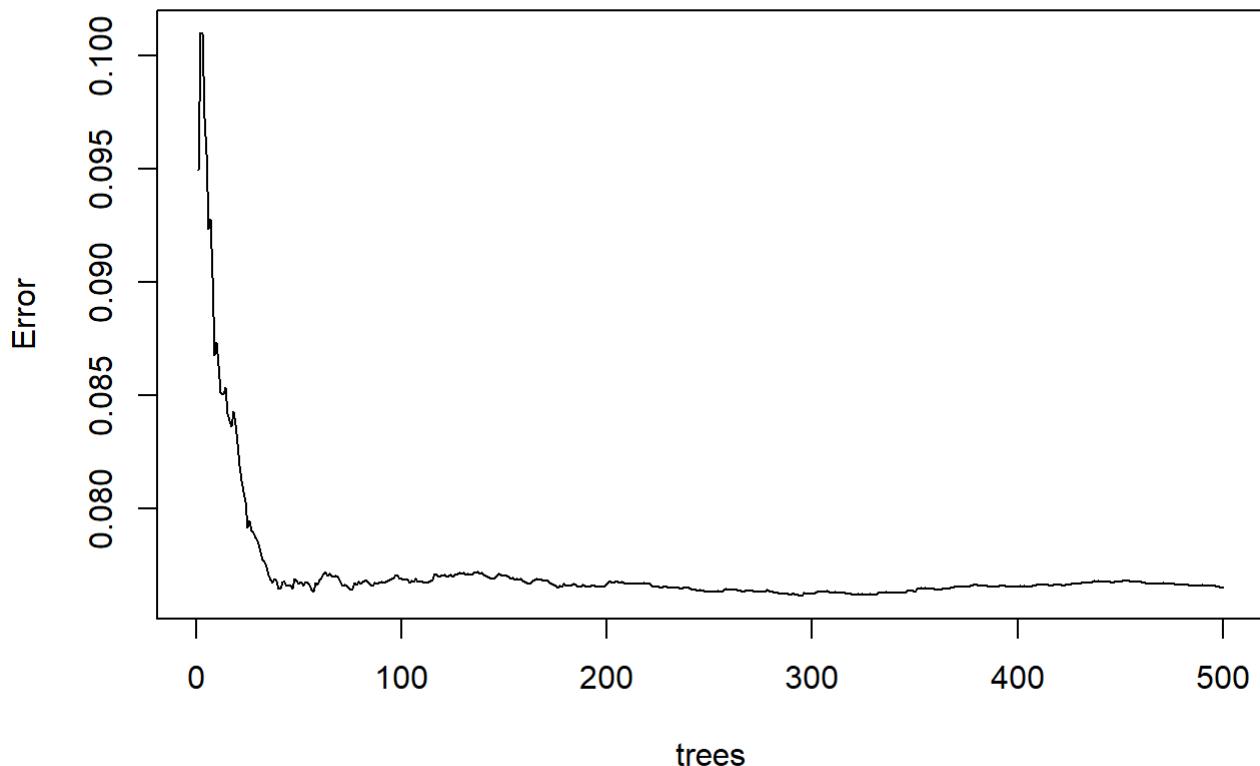
```
##
## Call:
##   randomForest(formula = log(precio) ~ ., data = train, importance = TRUE)
##   Type of random forest: regression
##   Number of trees: 500
##   No. of variables tried at each split: 1
##
##   Mean of squared residuals: 0.07651072
##   % Var explained: 74.34
```

class: center, middle

Número de árboles en el bosque

```
plot(r1)
```

r1

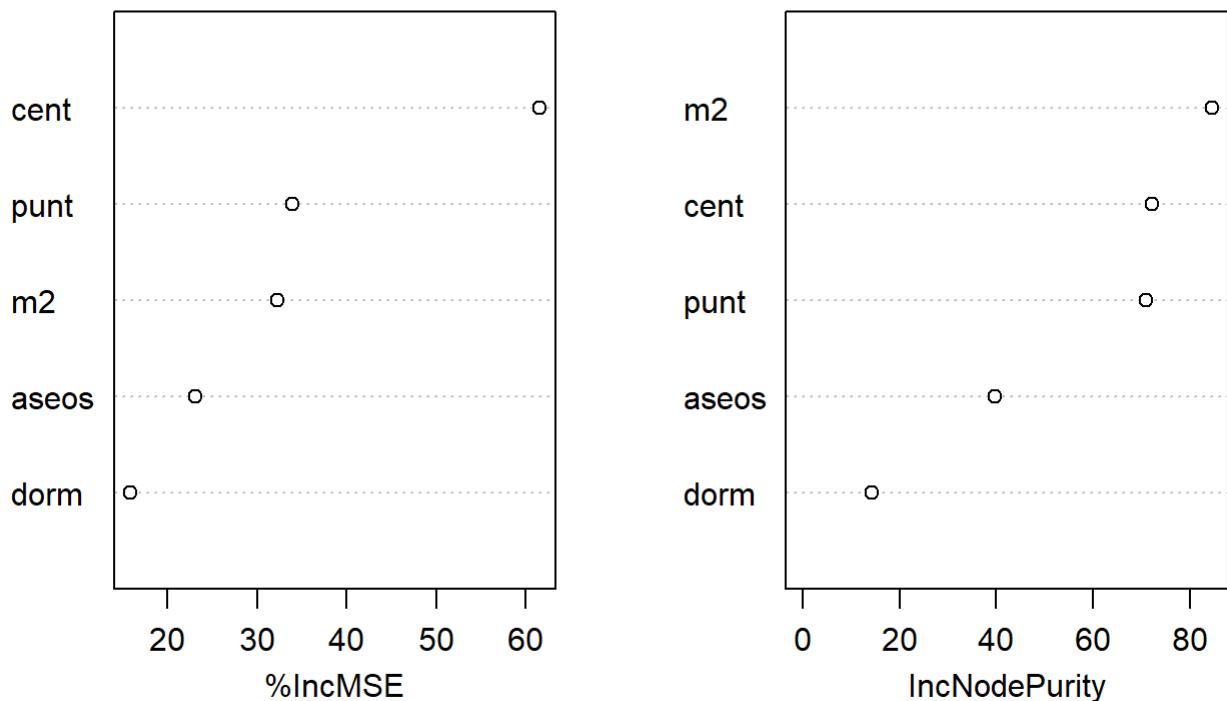


```
importance(r1)
```

```
##           %IncMSE IncNodePurity
## dorm     15.81155    14.27146
## aseos   23.07309    39.73250
## m2      32.24524    84.50895
## cent    61.53474    72.08478
## punt    33.96765    70.99063
```

```
varImpPlot(r1)
```

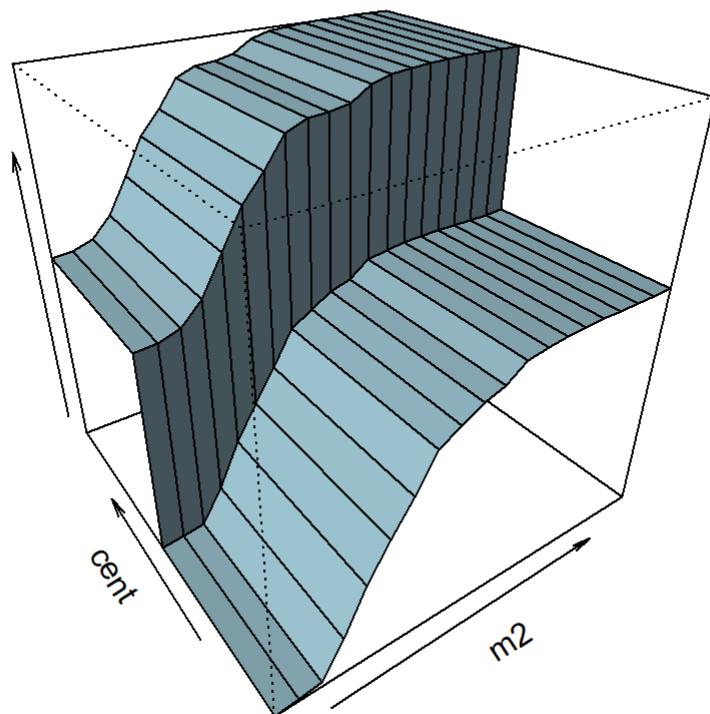
r1



```
plotmo(r1,degree1 = FALSE,degree2 = 8)
```

`log(precio) ~ randomForest(log(precio)~., data=train, importa...`

m2: cent



Housing Values in Suburbs of Boston

R Markdown

```
library(MASS)
library(rpart)
library(rpart.plot)
library(randomForest)
library(plotmo)
data(Boston)
## el nombre del data.frame es Boston. Por ejemplo names(Boston)
```

En **Boston** es el nombre de un conjunto de datos que se encuentra en el paquete **MASS**. Utiliza las instrucciones de arriba para cargar los datos. Utiliza `?Boston` para conocer las variables que contiene este conjunto de datos.

The Boston data frame has 506 rows and 14 columns.

This data frame contains the following columns:

- crim: per capita crime rate by town.
- zn: proportion of residential land zoned for lots over 25,000 sq.ft.
- indus: proportion of non-retail business acres per town.
- chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
- nox: nitrogen oxides concentration (parts per 10 million).
- rm: average number of rooms per dwelling.
- age: proportion of owner-occupied units built prior to 1940.
- dis: weighted mean of distances to five Boston employment centres.
- rad: index of accessibility to radial highways.
- tax: full-value property-tax rate per \$10,000.
- ptratio: pupil-teacher ratio by town.
- black: $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town.
- lstat: lower status of the population (percent).
- medv: median value of owner-occupied homes in \$1000s.

Toma el 80% de las observaciones para estimar los siguientes modelos y las restantes para comprobar la validez del mismo.

```
set.seed(789)
n = dim(Boston)[1]
sel = sample(1:n,.8*n)
train = Boston[sel,]
test = Boston[-sel,]
```

Responde a las siguientes preguntas:

1. Estima el modelo de regresión múltiple utilizando **medv** como variable respuesta y el resto como regresores.
2. Realiza la diagnosis y si es necesario, transforma los datos de manera adecuada para conseguir cumplir las condiciones del modelo.
3. Construye el árbol de regresión utilizando **medv** (en la misma escala de 2) como variable respuesta.
4. Poda el arbol utilizando el parámetro de complejidad.
5. Interpreta el árbol final
6. Estima el modelo de random forest utilizando la variable **medv** en la misma escala que 2. Indica las variables importantes del modelo.
7. Obtén los errores de los tres modelos con los datos reservados para validarlos y compáralos en un boxplot. Calcula R2 y sr para cada modelo. Elige el modelo que consideres más adecuado.

Apartado 1

```
m = lm(medv ~., data=train)
summary(m)
```

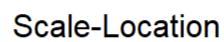
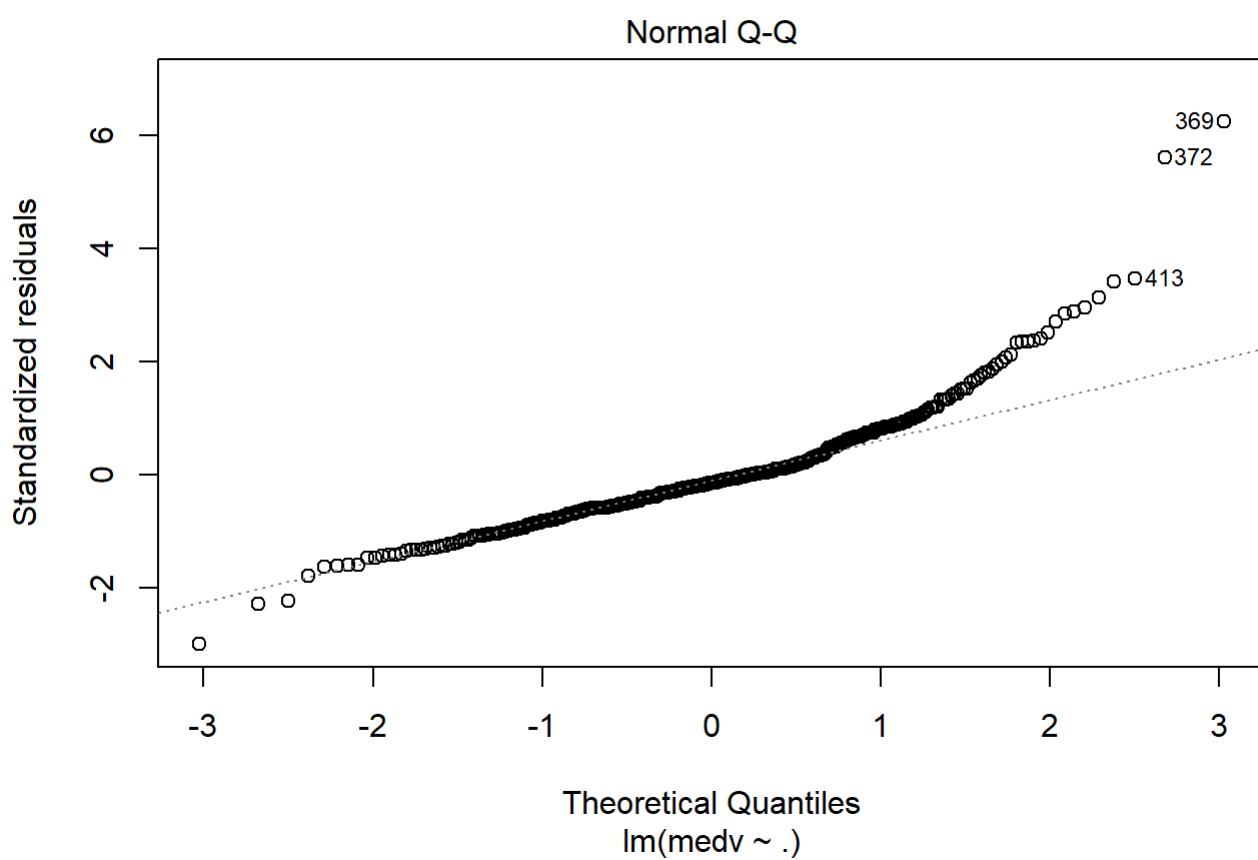
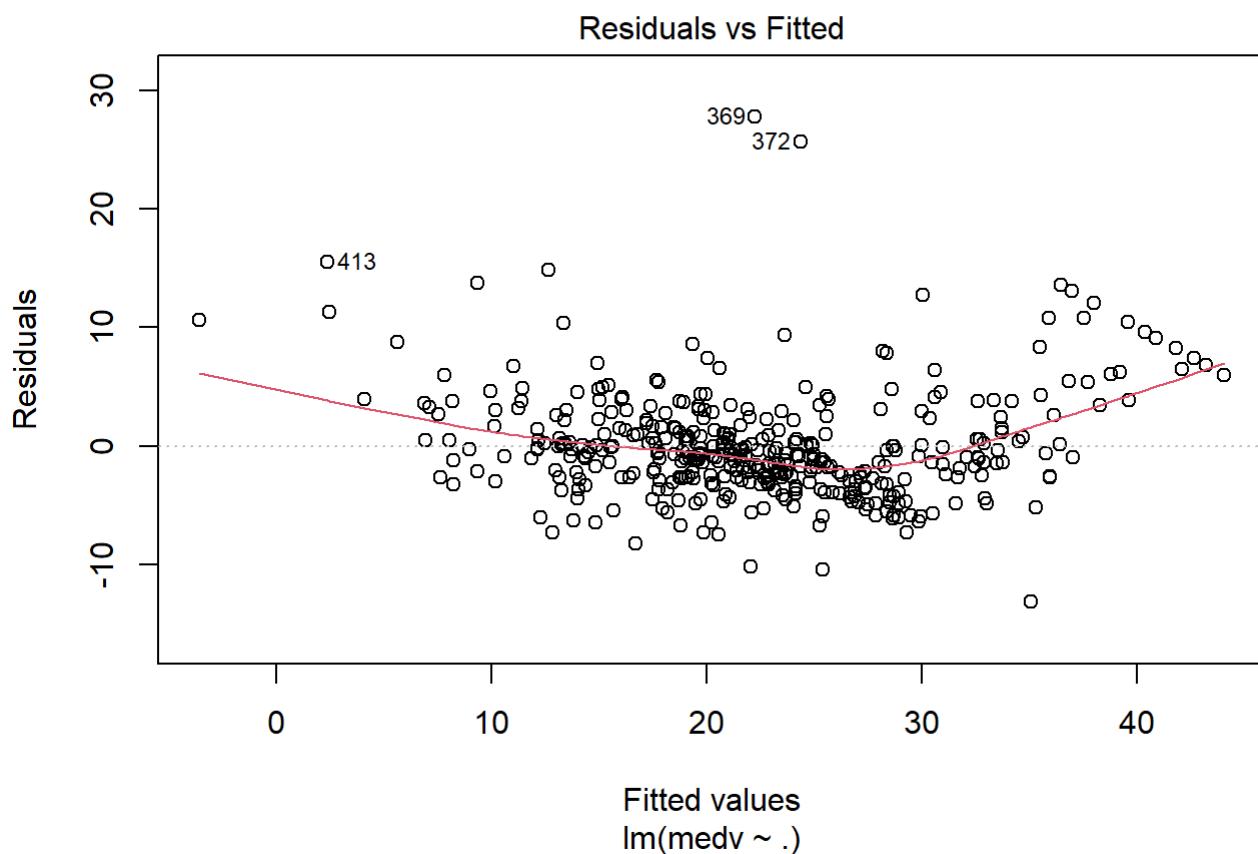
```

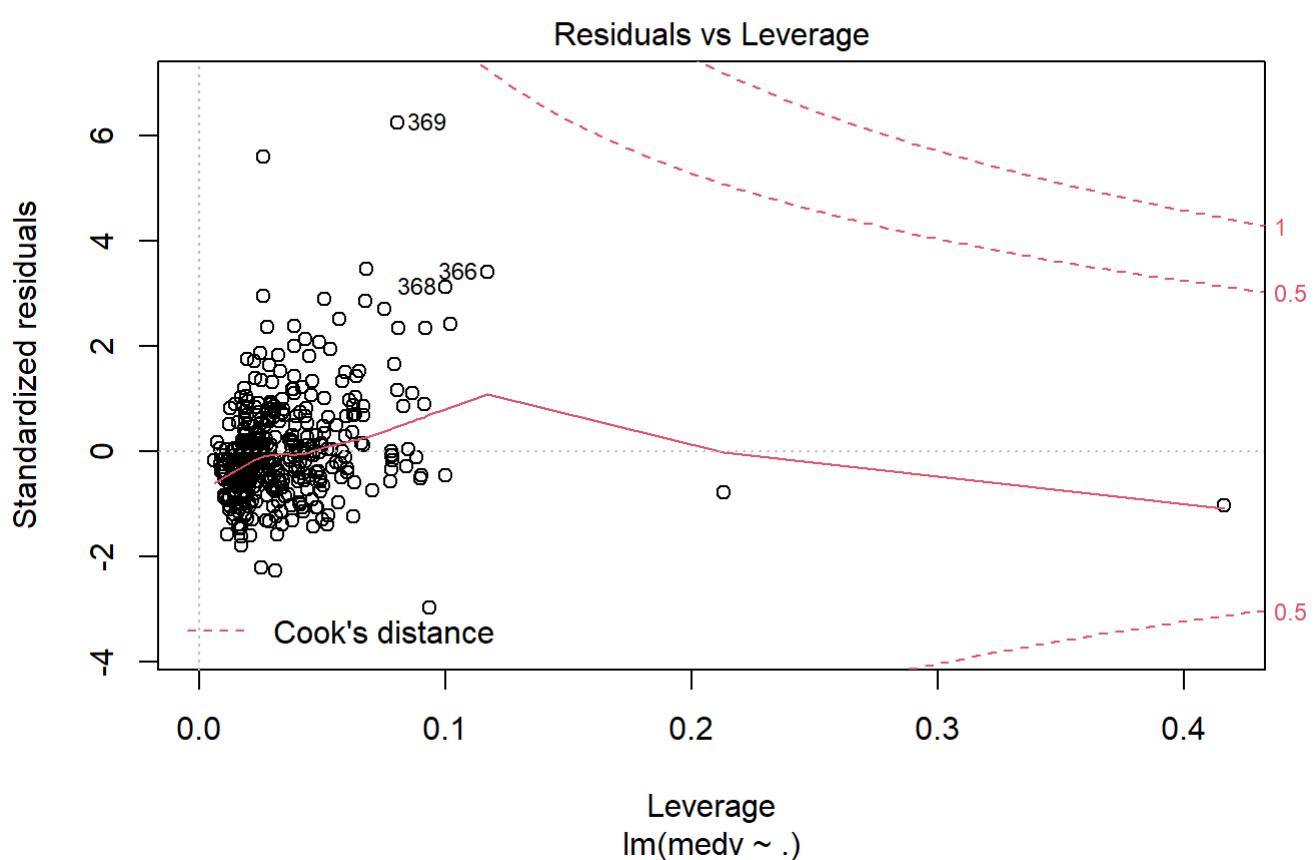
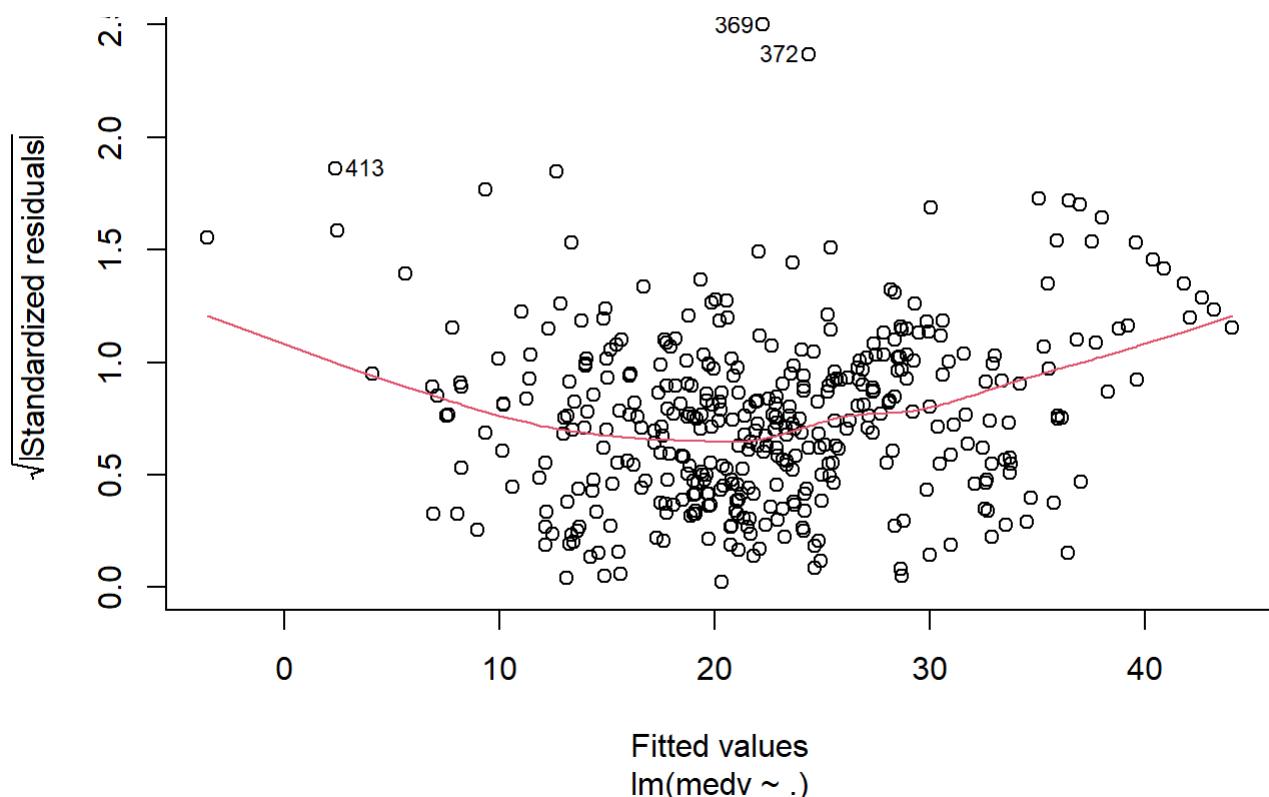
## 
## Call:
## lm(formula = medv ~ ., data = train)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -13.1648 -2.6434 -0.6312  1.7497 27.7533
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 35.566313  5.491643   6.476 2.83e-10 ***
## crim        -0.118214  0.037934  -3.116 0.001967 **  
## zn          0.039306  0.015425   2.548 0.011212 *   
## indus       0.002529  0.067730   0.037 0.970228    
## chas        0.521210  0.943143   0.553 0.580833    
## nox        -18.494366 4.081640  -4.531 7.81e-06 ***
## rm          4.105206  0.443666   9.253 < 2e-16 ***
## age        -0.003995  0.014391  -0.278 0.781445    
## dis        -1.529993  0.220022  -6.954 1.51e-11 ***
## rad         0.328137  0.070321   4.666 4.22e-06 ***
## tax        -0.014239  0.004056  -3.511 0.000499 *** 
## ptratio     -1.011582  0.143249  -7.062 7.59e-12 ***
## black       0.011538  0.002956   3.903 0.000112 *** 
## lstat      -0.420764  0.056038  -7.509 4.11e-13 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.63 on 390 degrees of freedom
## Multiple R-squared:  0.7438, Adjusted R-squared:  0.7353 
## F-statistic:  87.1 on 13 and 390 DF,  p-value: < 2.2e-16

```

Apartado 2

```
plot(m)
```





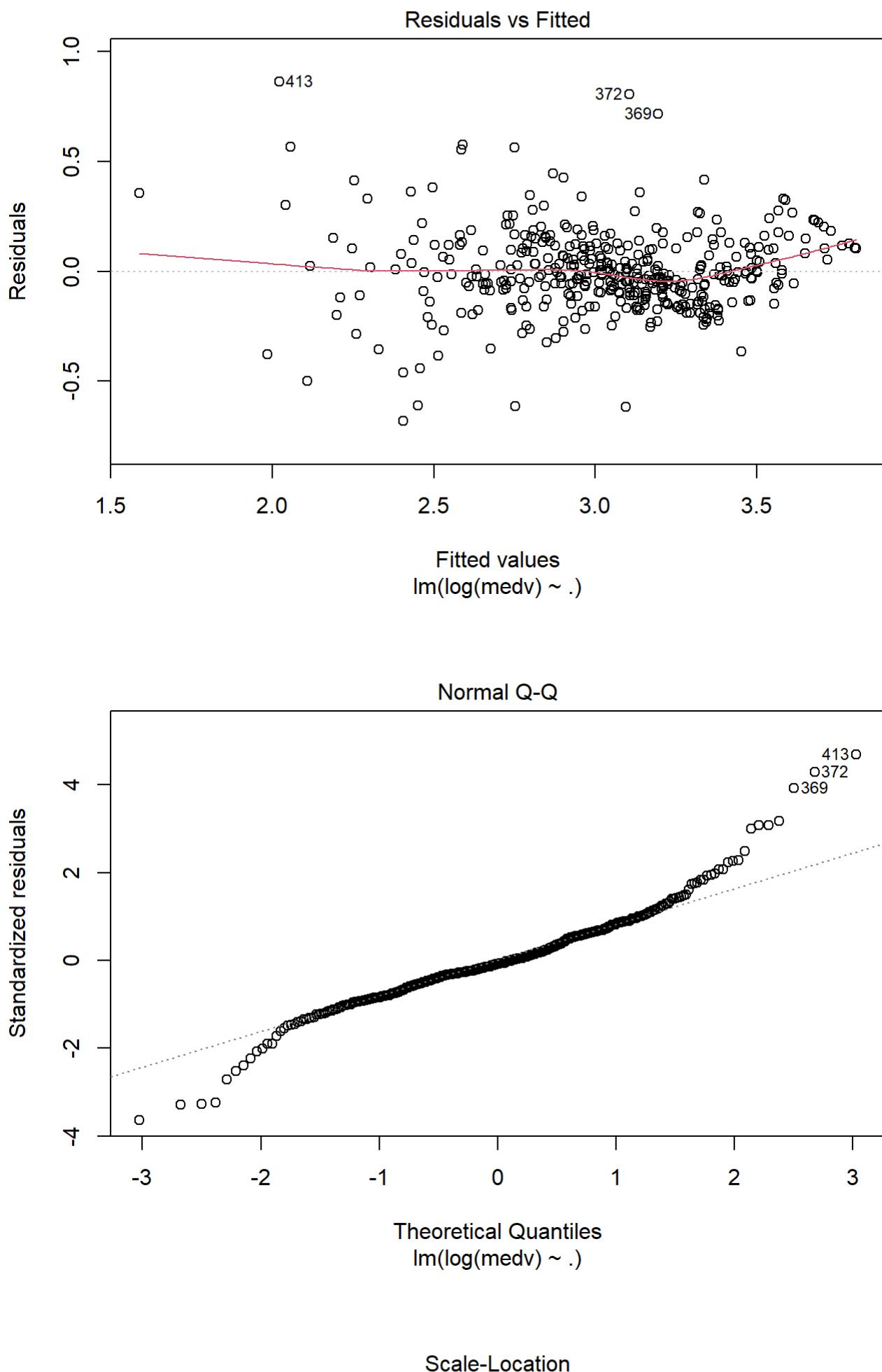
```
m1 = lm(log(medv) ~., data=train)
summary(m1)
```

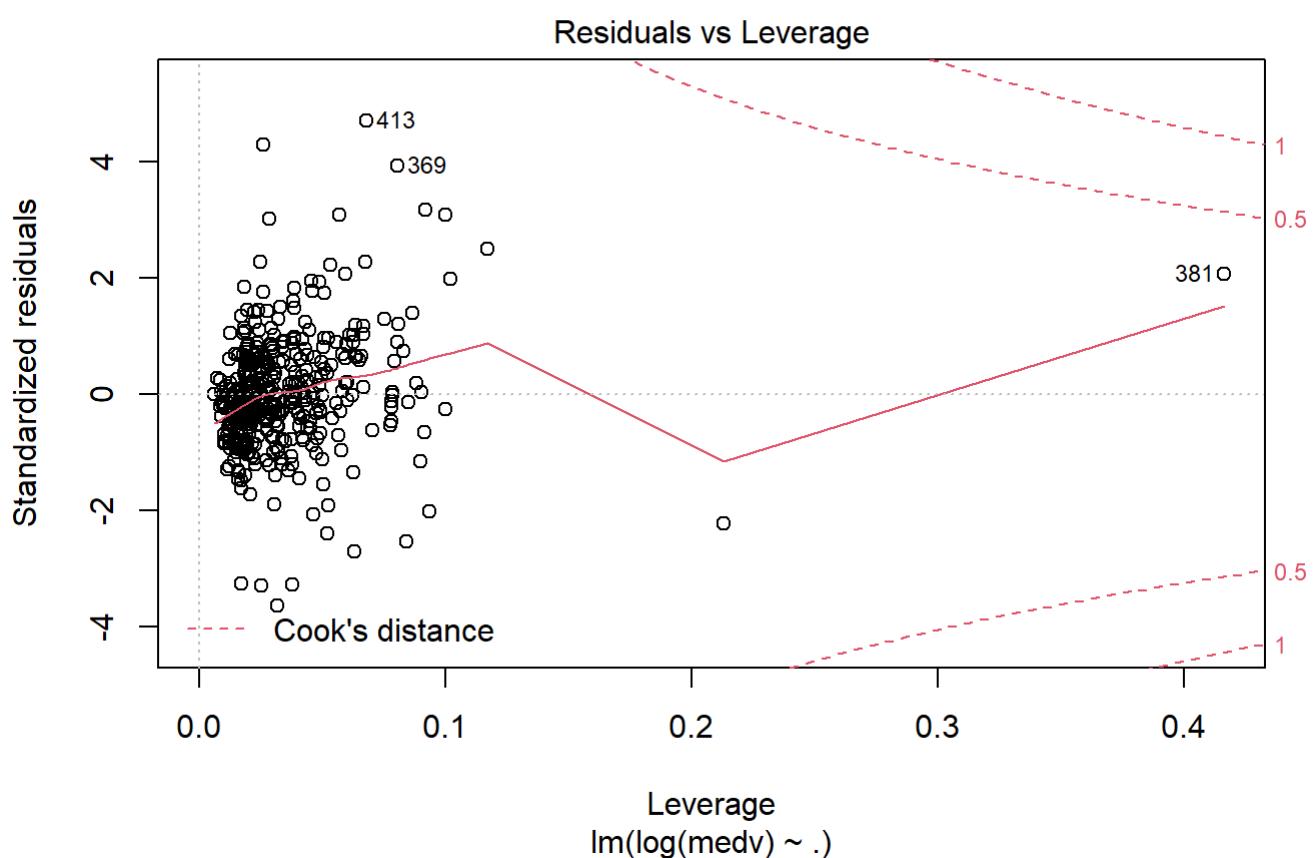
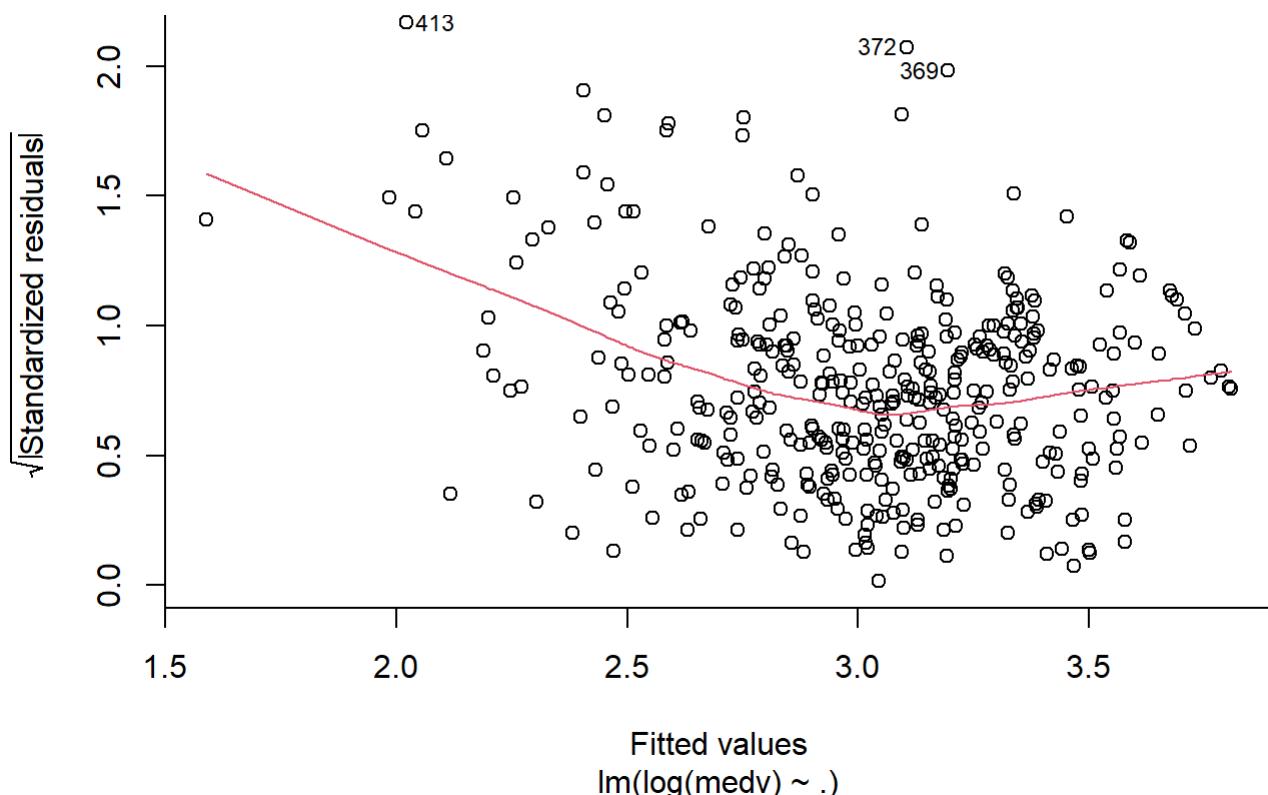
```

## 
## Call:
## lm(formula = log(medv) ~ ., data = train)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.68241 -0.10060 -0.01581  0.10453  0.86361
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.0700701  0.2257993 18.025 < 2e-16 ***
## crim        -0.0114260  0.0015597 -7.326 1.38e-12 ***
## zn          0.0009033  0.0006342  1.424   0.155    
## indus       0.0020529  0.0027849  0.737   0.461    
## chas        0.0388024  0.0387791  1.001   0.318    
## nox         -0.7851828  0.1678244 -4.679 3.99e-06 ***
## rm          0.0958318  0.0182422  5.253 2.46e-07 ***
## age         0.0001113  0.0005917  0.188   0.851    
## dis         -0.0503605  0.0090466 -5.567 4.84e-08 ***
## rad         0.0160213  0.0028914  5.541 5.54e-08 ***
## tax         -0.0007123  0.0001668 -4.271 2.45e-05 ***
## ptratio     -0.0404667  0.0058900 -6.870 2.54e-11 ***
## black       0.0005592  0.0001215  4.601 5.68e-06 ***
## lstat      -0.0258234  0.0023041 -11.208 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1904 on 390 degrees of freedom
## Multiple R-squared:  0.7848, Adjusted R-squared:  0.7777 
## F-statistic: 109.4 on 13 and 390 DF,  p-value: < 2.2e-16

```

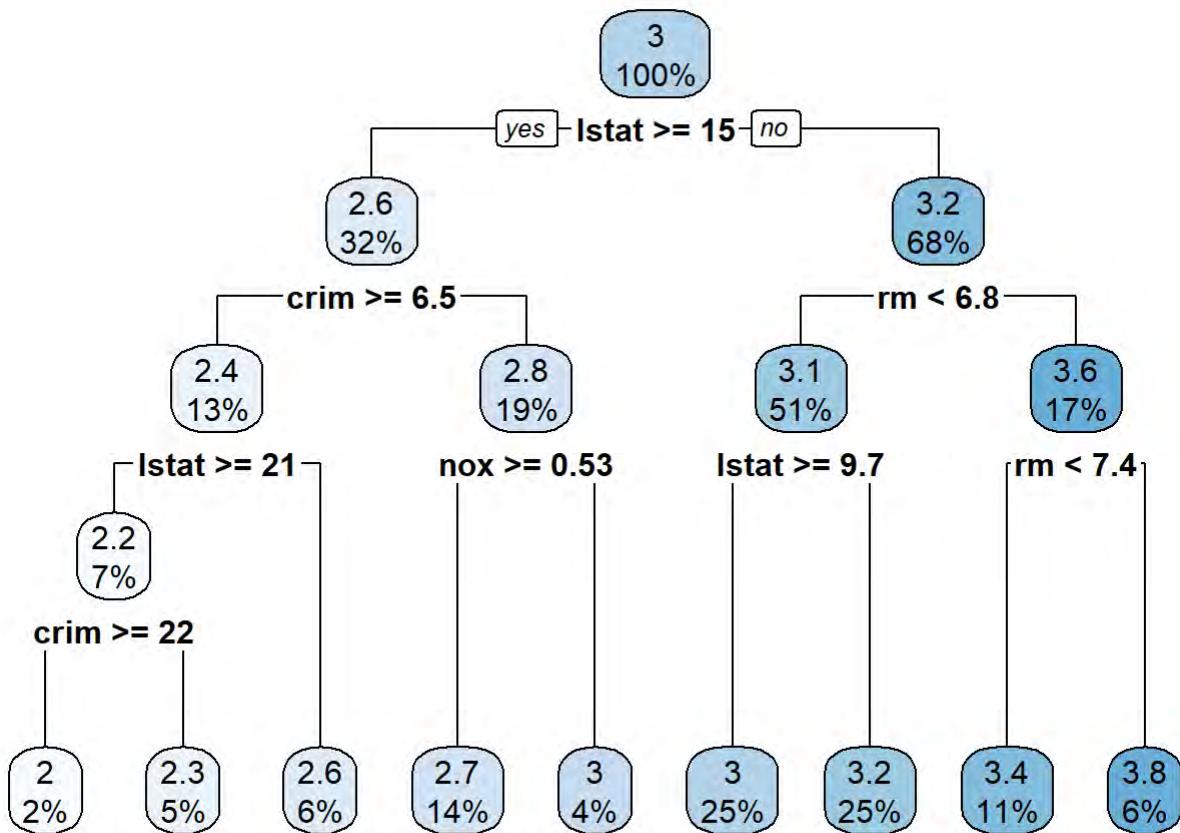
```
plot(m1)
```





Apartado 3

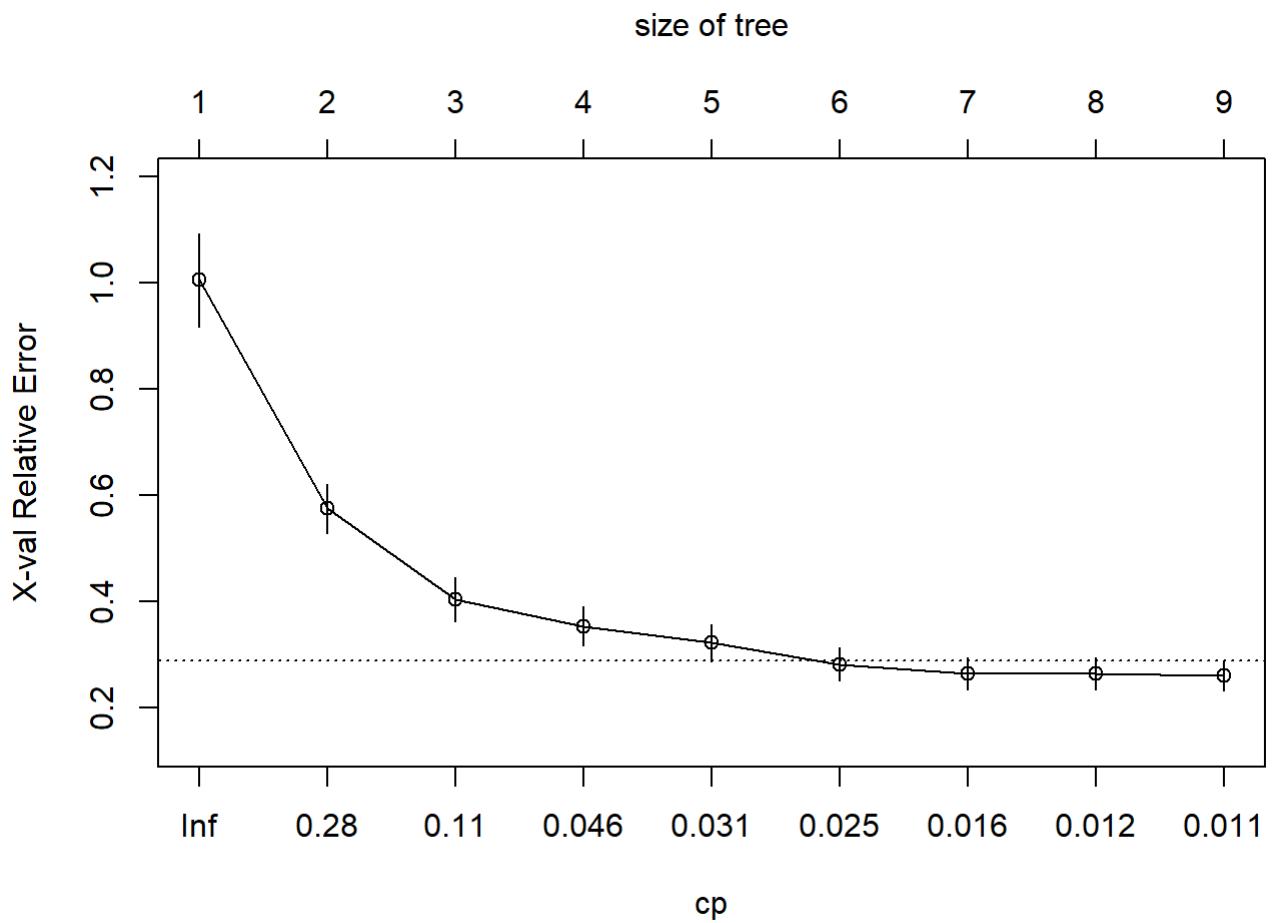
```
t = rpart(formula = log(medv) ~ ., data = train)
rpart.plot(t)
```



```
printcp(t)
```

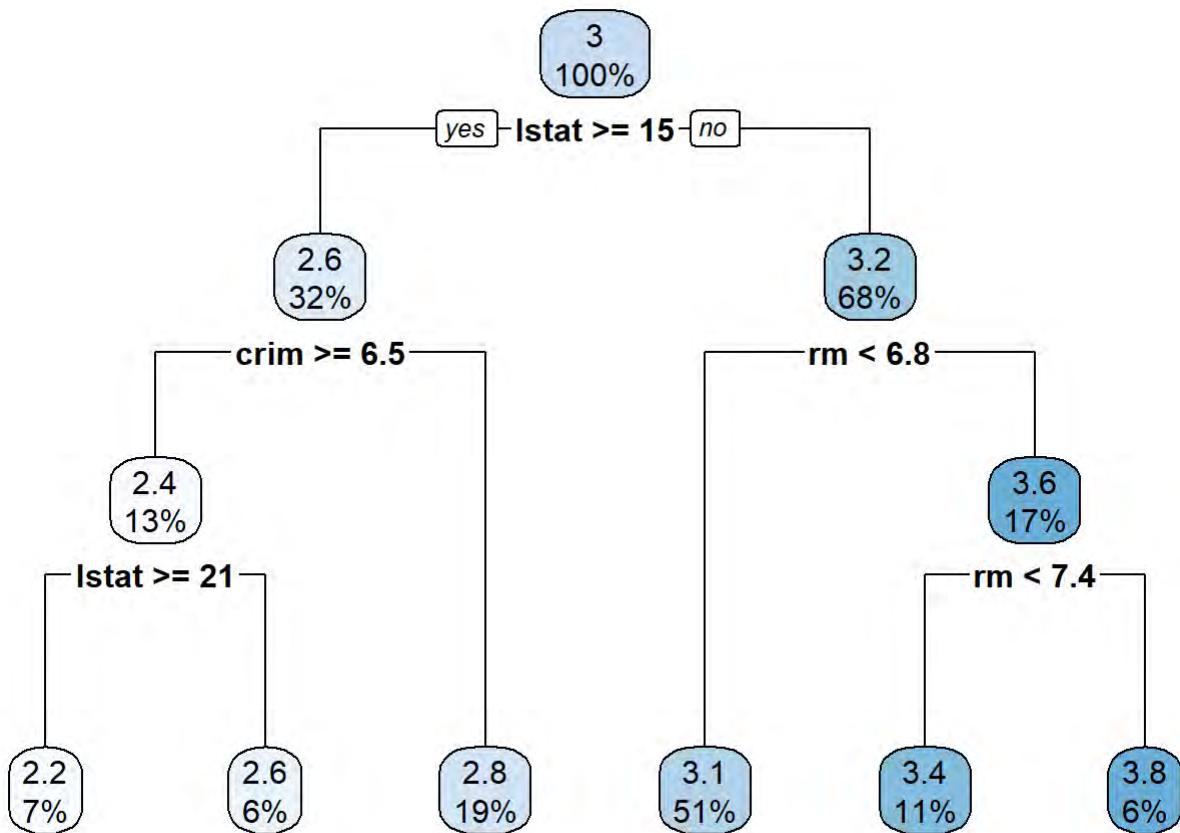
```
##
## Regression tree:
## rpart(formula = log(medv) ~ ., data = train)
##
## Variables actually used in tree construction:
## [1] crim lstat nox rm
##
## Root node error: 65.681/404 = 0.16258
##
## n= 404
##
##          CP nsplit rel error xerror      xstd
## 1 0.443071     0  1.00000 1.00428 0.087006
## 2 0.177220     1  0.55693 0.57413 0.046665
## 3 0.069768     2  0.37971 0.40346 0.042159
## 4 0.030815     3  0.30994 0.35278 0.036750
## 5 0.030398     4  0.27913 0.32143 0.034751
## 6 0.020185     5  0.24873 0.28116 0.030933
## 7 0.012583     6  0.22854 0.26288 0.029960
## 8 0.011931     7  0.21596 0.26331 0.030028
## 9 0.010000     8  0.20403 0.25966 0.028618
```

```
plotcp(t)
```



Apartado 4

```
t1 = prune(t,0.028)
rpart.plot(t1)
```



```
printcp(t1)
```

```
##
## Regression tree:
## rpart(formula = log(medv) ~ ., data = train)
##
## Variables actually used in tree construction:
## [1] crim  lstat rm
##
## Root node error: 65.681/404 = 0.16258
##
## n= 404
##
##          CP nsplit rel error  xerror     xstd
## 1 0.443071      0  1.00000 1.00428 0.087006
## 2 0.177220      1  0.55693 0.57413 0.046665
## 3 0.069768      2  0.37971 0.40346 0.042159
## 4 0.030815      3  0.30994 0.35278 0.036750
## 5 0.030398      4  0.27913 0.32143 0.034751
## 6 0.028000      5  0.24873 0.28116 0.030933
```

Apartado 5

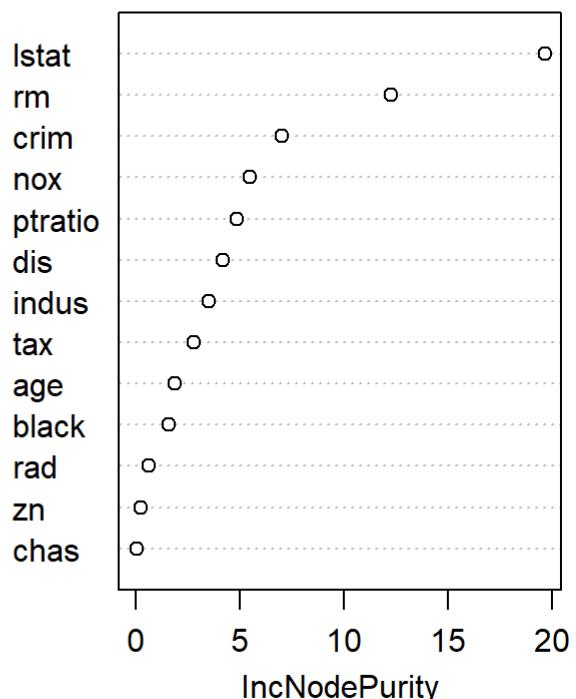
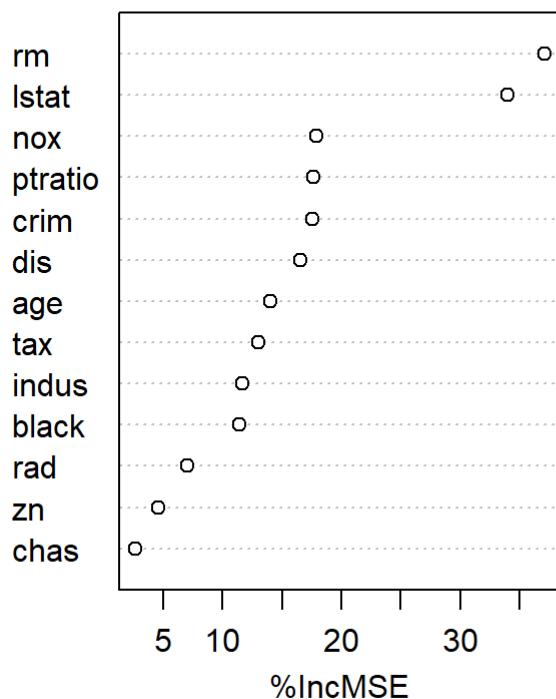
Apartado 6

```
r1 = randomForest(log(medv)~.,data=train,importance=TRUE)
print(r1)
```

```
##
## Call:
##   randomForest(formula = log(medv) ~ ., data = train, importance = TRUE)
##   Type of random forest: regression
##   Number of trees: 500
##   No. of variables tried at each split: 4
##
##   Mean of squared residuals: 0.02273219
##   % Var explained: 86.02
```

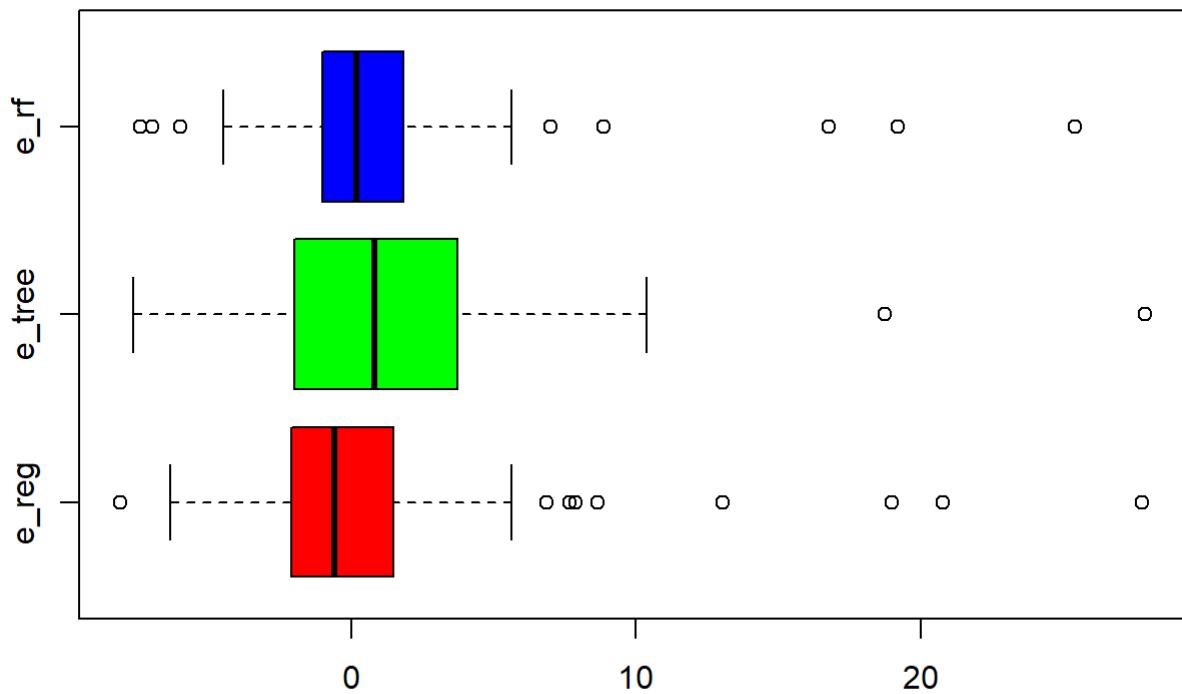
```
varImpPlot(r1)
```

r1



Apartado 7

```
y1 = exp(predict(m1,newdata=test))
y2 = exp(predict(t1,newdata=test))
y3 = exp(predict(r1,newdata=test))
e_reg = test$medv-y1
e_tree = test$medv-y2
e_rf = test$medv-y3
e=cbind(e_reg,e_tree,e_rf)
boxplot(e,horizontal=TRUE,col=rainbow(3))
```



```
sqrt(colMeans(e^2))
```

```
##      e_reg      e_tree      e_rf
## 5.121745 5.946054 4.416951
```

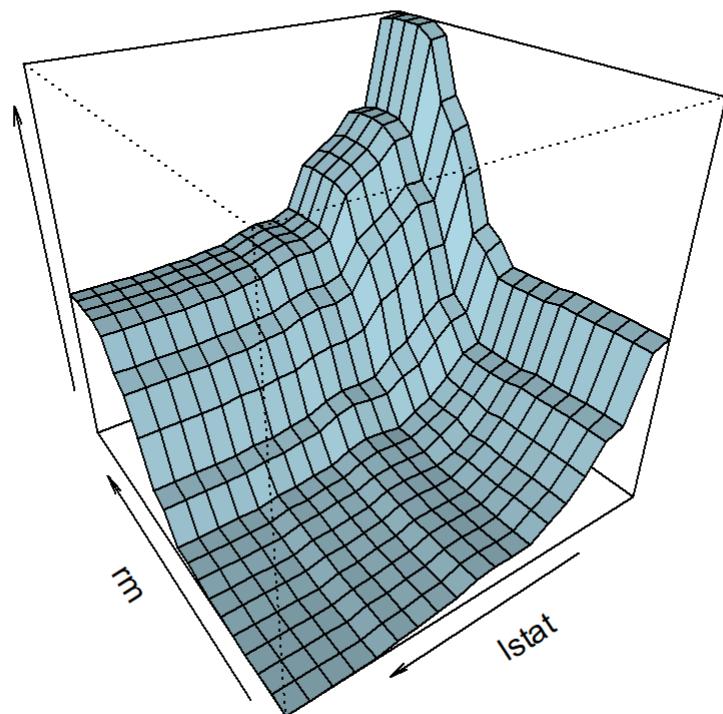
```
VT= sum((test$medv-mean(test$medv))^2)
1-colSums(e^2)/VT
```

```
##      e_reg      e_tree      e_rf
## 0.7341612 0.6417056 0.8022906
```

```
plotmo(r1,degree1 = FALSE,degree2 = 6)
```

```
log(medv)    randomForest(log(medv)~., data=train, importanc...
```

rm: lstat



Conclusiones:

Parece que el mejor modelo en este caso es el de Random Forest.

Los resultados del análisis dependen de varios pasos que se han realizado con muestreo aleatorio. Por ejemplo, la división entre datos para estimar y datos para validar es aleatoria, al cambiar de semilla, el resultado cambia. Conviene repetir el análisis con varias semillas, para ver que la comparación sigue proporcionando los mismos resultados. También, el propio Random Forest es intrínsecamente “aleatorio”, cada vez que obtenemos un modelo de RF, los resultados cambian. Para que las conclusiones sean sólidas, se recomienda repetir el proceso de comparación varias veces.

El modelo de regresión es muy mejorable, incluyendo combinación de variables (productos) que proporcionan un modelo mejor que el obtenido. Por ejemplo:

```
m2 = lm(log(medv) ~ . + lstat*rm, data=train)
summary(m2)
```

```

## 
## Call:
## lm(formula = log(medv) ~ . + lstat * rm, data = train)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.63097 -0.09307 -0.01045  0.08297  0.80110
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.2340851  0.2348979 13.768 < 2e-16 ***
## crim        -0.0130067  0.0014631 -8.890 < 2e-16 ***
## zn          0.0003531  0.0005934  0.595  0.55219  
## indus       0.0032692  0.0025923  1.261  0.20803  
## chas        0.0267416  0.0360670  0.741  0.45887  
## nox         -0.6989638  0.1563277 -4.471 1.02e-05 ***
## rm          0.2110271  0.0223407  9.446 < 2e-16 ***
## age         0.0005614  0.0005528  1.016  0.31047  
## dis         -0.0398277  0.0085111 -4.680 3.98e-06 ***
## rad         0.0164737  0.0026874  6.130 2.16e-09 ***
## tax         -0.0007112  0.0001550 -4.589 6.01e-06 ***
## ptratio     -0.0335160  0.0055431 -6.046 3.48e-09 ***
## black       0.0003774  0.0001152  3.275  0.00115 ** 
## lstat       0.0433186  0.0089929  4.817 2.09e-06 ***
## rm:lstat   -0.0124623  0.0015743 -7.916 2.58e-14 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1769 on 389 degrees of freedom
## Multiple R-squared:  0.8147, Adjusted R-squared:  0.808 
## F-statistic: 122.1 on 14 and 389 DF,  p-value: < 2.2e-16

```

```

y4 = exp(predict(m2,newdata=test))
e4=test$medv-y4
sqrt(mean(e4^2))

```

```

## [1] 4.677363

```

Los resultados de Random Forest se pueden utilizar para identificar combinaciones de variables para mejorar el modelo de regresión lineal.

El modelo de Random Forest también se puede mejorar ...

IMPORTANTE. Este archivo son notas rápidas sin depurar realizadas para los alumnos de 4º de GITI de Organización, por favor no distribuirlos. Puede contener errata, faltas y errores.

Source (de los datos):

Harrison, D. and Rubinfeld, D.L. (1978) Hedonic prices and the demand for clean air. J. Environ. Economics and Management 5, 81–102.

Belsley D.A., Kuh, E. and Welsch, R.E. (1980) Regression Diagnostics. Identifying Influential Data and Sources of Collinearity. New York: Wiley.

Tarea 4 Random Forest

Identificación de números escritos a mano

Prof: Eduardo Caro y JesúsJuan

Random Forest: Identificación de números escritos a mano

En esta tarea vamos a ver la aplicación de técnicas estadísticas al reconocimiento de caracteres manuscritos. Trabajaremos con un conjunto de datos muy popular en internet MNIST. Este ejemplo ha sido utilizado en muchas competiciones y universidades para enseñar diferentes técnicas de Machine Learning. En nuestro caso, utilizaremos Random Forest de clasificación.

La base de datos MNIST (<http://yann.lecun.com/exdb/mnist/>) de dígitos escritos a mano está formada por un conjunto de entrenamiento de 60.000 ejemplos (digit_train.csv) y un conjunto de validación de 10.000 ejemplos (digit_test.csv). Los dígitos han sido normalizados en tamaño y centrados en una imagen de tamaño fijo. Se trata de una buena base de datos para quienes deseen probar técnicas de aprendizaje y métodos de reconocimiento de patrones en datos del mundo real, dedicando un esfuerzo mínimo al preprocesamiento y al formateo. Cada imagen está formada por una matriz de 28x28 pixeles que contienen el nivel de gris en una escala de 0 (blanco) a 256 (negro).

```
library(randomForest)
train = read.csv("digit_train.csv", header=TRUE)
test = read.csv("digit_test.csv", header=TRUE)
```

Cada fila del archivo corresponde a la imagen de un dígito. La variable **digit** es el dígito que se representa y las 784 variables restantes (**X1, X2,...,X784**) el nivel de gris de los 28x28 pixeles de la imagen (almacenados por filas). Con las siguientes instrucciones reconstruyo la imagen del dígito numero 11715, dibujando los 784 pixeles.

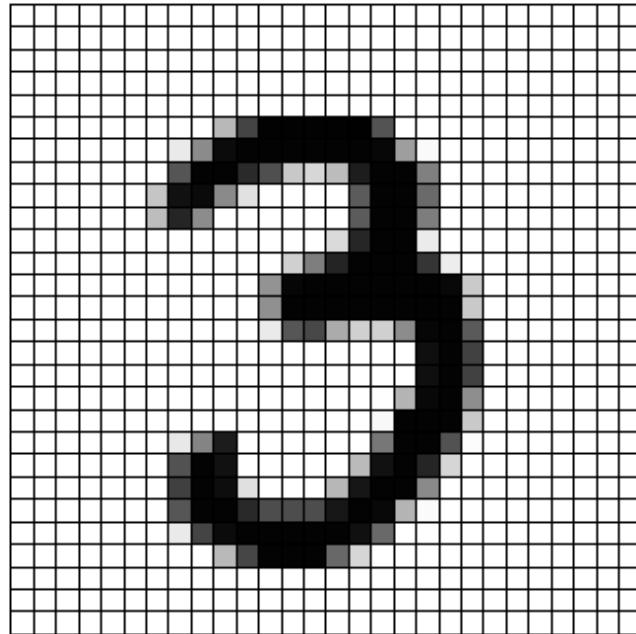
```
num = 11715
```

```
par(mar = rep(1,4))

xy = expand.grid(1:28, 1:28) # coordenadas en la rejilla de 28 x 28
z = as.numeric(train[num, 2:785])/256 # nivel de gris del número elegido

plot(0, 0, type = "n", xlab = "", ylab = "", axes = FALSE,
     xlim = c(0, 28), ylim = c(0, 28), asp = 1) #crea ejes y no dibuja
nada
```

```
rect(xy[,1]-1,27-xy[,2],xy[,1],28-xy[,2],col=gray(1-z)) #dibuja 784
rectangulos
```



A continuación se muestran 16 dígitos de la base de datos elegidos al azar.

```
set.seed(1345)
par(mfrow=c(4,4))
num=sample(1:60000,16)
par(mar = rep(0,4))
for (i in 1:16){

  plot(0, 0, type = "n", xlab = "", ylab = "", axes =FALSE,
    xlim = c(0, 28), ylim = c(0, 28),asp=1)

  z = as.numeric(train[num[i],2:785])/256

  rect(xy[,1] - 1, 27 - xy[,2], xy[,1] ,28 - xy[,2],
    col=gray(1-z), border = 'transparent' ) }
```

8	4	4	1
7	3	5	8
2	4	2	0
9	2	9	9

NOTA IMPORTANTE: El análisis que se propone en esta tarea supone trabajar con una matriz de 60000 x 285 y requiere bastante tiempo de cálculo. Si trabajamos con Rmarkdow, para evitar la repetición de cálculos realizados y ahorrar tiempo en la generación del documento es recomendable utilizar la opción **cache = TRUE** en los *chunks* que necesiten tiempos elevados de cálculo. Una posibilidad es declarar la opción por defecto al principio del documento: **knitr::opts_chunk\$set(echo = TRUE,cache = TRUE)**

Preguntas:

1. Construir un bosque aleatorio (RandomForest) con 50 árboles para clasificar los dígitos. ¿Cuál es el porcentaje de error de clasificación? Interpreta la matriz de confusión correspondiente a los errores OOB. Indica los dos dígitos más fáciles de identificar y los dos más difíciles según tú solución. (El ordenador puede tardar 10 minutos en hacer el cálculo de un RF de 50 árboles). Para los alumnos que no consigan la solución en tiempo razonable, pueden cargar una solución obtenida por mí que se encuentra en el archivo “bosque.rds”. Para leer el archivo utiliza la instrucción `rf = loadRDS("bosque.rds")`. Puedes contestar a las preguntas 1,2 y 3 con esta solución.
2. Con el RF obtenido, clasifica el conjunto “test” y obtén la matriz de confusión, proporcionando el porcentaje de error para cada dígito. Obtén el error medio de clasificación en el conjunto test. (Nota. si `t` es una matriz o tabla, `diag(t)` nos da el vector con los elementos diagonales y `rowSums(t)` el vector con la suma de cada fila)
3. Con las instrucciones usadas arriba para representar gráficamente las imágenes de los dígitos, muestra las imágenes de los dígitos que corresponden al 9 y que el algoritmo erróneamente los ha clasificado como 4.
4. Crea un `data.frame` que contenga solo las filas de los dígitos 4 y 9. Construye un Random Forest con 150 árboles para clasificar las observaciones. Asegúrate de que los niveles de la variable `digit` del nuevo `data.frame` sean 4 y 9, utilizando de nuevo la función `factor()` o la función `droplevels()`. Proporciona la matriz de confusión e interpreta los resultados.
5. Indica los 10 píxeles (variables x) más importantes utilizadas por el RF del apartado 4 para diferenciar entre 4 y 9. Indica gráficamente a qué coordenadas corresponden en la matriz de 28x28.
6. A partir del conjunto “test” crea el `data.frame` que contenga únicamente las observaciones correspondientes a los dígitos 4 y 9. Evalúa el modelo RF construido en el apartado anterior con este nuevo `data.frame` de test , calculando la matriz de confusión y el error de clasificación global.
7. Para entender la lógica que utiliza el algoritmo de Random Forest en la clasificación de dígitos realiza el siguiente análisis. Con el `data.frame` obtenido en el apartado 4, calcula y dibuja el árbol de clasificación con `cp = 0.01`. Interpreta el árbol.
8. Utiliza el conjunto test creado en el apartado 5 y evalúa el árbol estimado en el apartado anterior. Calcula la matriz de confusión y el error medio. Compara los resultados con los obtenidos en el apartado 5 e interpreta las diferencias.

9. Añade al estudio realizado algún aspecto “original” que consideres de interés. El alumno debe plantear una cuestión (de interés) diferente a las formuladas y que esté relacionada con los datos del problema y resolverla. Esta pregunta es abierta y tiene como objetivo completar el análisis realizado en la tarea. Debe ser breve, la misma extensión que el resto de las preguntas.
10. Haz un resumen del análisis realizado en esta tarea, indicando las conclusiones que consideres más relevantes.

Todos los apartados valen 1 punto.

SOLUCIÓN

Pregunta 1

Obtén el RandomForest con 50 árboles para clasificar los dígitos. ¿Cuál es el porcentaje de error de clasificación? Interpreta la matriz de confusión correspondiente a los errores OOB. Indica los dos dígitos más fáciles de identificar y los dos más difíciles según tú solución.

(La solución con 150 árboles puede tardar una hora y tiene una precisión muy similar a la de 50 árboles)

```
set.seed(19)
train$digit = factor(train$digit)
rf = randomForest(x=train[,2:785],y = train$digit,ntree = 50)
rf

##
## Call:
##   randomForest(x = train[, 2:785], y = train$digit, ntree = 50)
##             Type of random forest: classification
##                         Number of trees: 50
## No. of variables tried at each split: 28
##
##           OOB estimate of  error rate: 4.13%
## Confusion matrix:
##      0   1   2   3   4   5   6   7   8   9 class.error
## 0 5821   1  14   8   5   9  30   1  31   3  0.01722100
## 1   1 6632   35  16  12   9   7   9  16   5  0.01631563
## 2   32   14 5727   25  27   9  20  45  48  11  0.03877140
## 3   14    9  92 5732   7  88  16  54  84  35  0.06507911
## 4   12   13   14    2 5610   7  34  13  23 114  0.03971243
## 5   25    8   12   94  13 5140   49   9  44  27  0.05183545
## 6   24    8    9    2  16   48 5786   0  25   0  0.02230483
## 7    6   18   68   14  39    4    2 6014  14  86  0.04006385
## 8   15   33   63   66  36   64  30  12 5460   72  0.06682618
## 9   19   10   18   67  98   32    5   55  47 5598  0.05900151
```

En esta solución, los dígitos más difíciles de identificar son el 3 (6.5% de error) y el 8 (6.7% de error). El 9 también tiene una tasa de error alta (5.9%). El 3 se confundió mayoritariamente con el 2, el 5 y el 8. El 8 se confundió con 9, 2, 3 y 5. El 9 mayoritariamente con el 4. Importante tener en cuenta que esto varía de una solución a otra. Estos errores se corresponden con las observaciones que en las simulaciones de cada árbol estaban en el conjunto Out of the Bag.

La impresión general es que el algoritmo clasifica correctamente el 96% de los dígitos (OOB estimate of error rate: 4.13%).

Pregunta 2

Con el RF obtenido, clasifica el conjunto “test” y obtén la matriz de confusión, proporcionando el porcentaje de error para cada dígito (intenta reproducir la misma estructura que la matriz del apartado 1). Obtén el error medio de clasificación en el conjunto test. (Nota. si t es una matriz o tabla, $diag(t)$ nos da el vector con los elementos diagonales y $rowSums(t)$ el vector con la suma de cada fila)

```

pred = predict(rf,newdata = test)
t=table(test$digit,pred)
(t1 = cbind(t,class.error = (rowSums(t)-diag(t))/rowSums(t)))

##      0    1    2    3    4    5    6    7    8    9 class.error
## 0 967    0    0    0    1    3    3    2    4    0  0.01326531
## 1   0 1122    2    3    0    3    3    1    1    0  0.01145374
## 2   5    0 999    5    5    0    4    7    7    0  0.03197674
## 3   0    0 10 967    0  10    1  10    9    3  0.04257426
## 4   1    0    0    0 953    0    6    0    4  18  0.02953157
## 5   4    0    1  11    3 859    5    2    6    1  0.03699552
## 6   8    3    0    0    3    5 936    0    3    0  0.02296451
## 7   1    8  20    3    1    0    0 983    4    8  0.04377432
## 8   3    0    5  11    4    7    4    4 925   11  0.05030801
## 9   4    5    2    8   12    6    0    5    7 960  0.04856293

aciertos = sum(diag(t))/sum(t)
paste("Porcentaje de aciertos ", round(aciertos*100, digits=3))

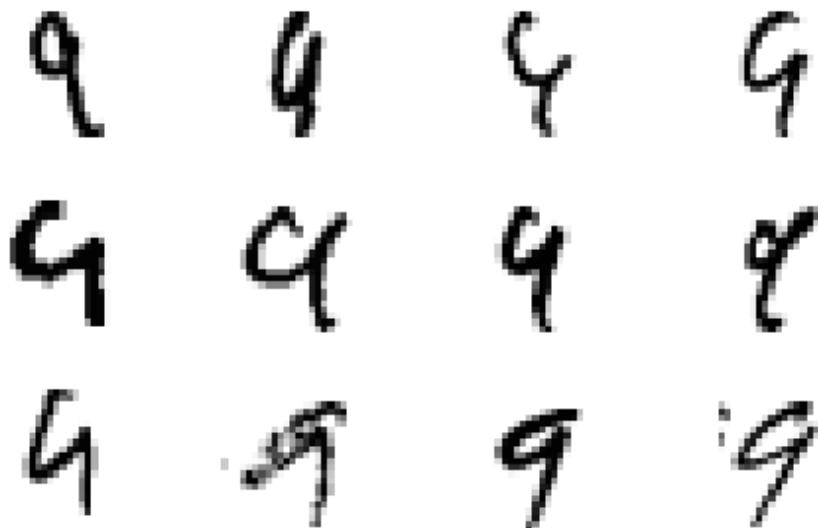
## [1] "Porcentaje de aciertos 96.71"

```

El error medio global en el conjunto test es 3.3%. Los dígitos que muestran más errores de clasificación son el 8 y el 9. La solución, en general, se parece a la obtenida con los errores OOB.

Pregunta 3

Con las instrucciones usadas arriba para representar gráficamente las imágenes de los dígitos, muestra las imágenes de los dígitos que corresponden al 9 y que el algoritmo los ha clasificado como 4



Como se ve en la tabla de confusión hay 12 que eran 9 y el algoritmo los clasificó como 4. En las imágenes se aprecia que en alguno de ellos la figura del dígito 9 no estaba perfectamente perfilada.

Aunque no lo pedían, a continuación, se muestra el error inverso. Dígitos que eran 4 y que el algoritmo los clasificó como 9. Como se aprecia en la matriz de confusión son 18. En la figura ponemos alguno de ellos.

4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4		

Pregunta 4

Crea un **data.frame** sólo con las imágenes de los números 4 y 9. Construye un Random Forest con 150 árboles para clasificar las observaciones. Asegúrate de que los niveles de la variable **digit** del nuevo **data.frame** sean 4 y 9, utilizando **factor()** o la función **droplevels()**. Proporciona la matriz de confusión e interpreta los resultados. Indica también los 10 píxeles (variables x) más importantes para diferenciar entre 4 y 9.

```
set.seed(87)
sel = train$digit %in% c("4", "9")
# otra opción -> sel = (train$digit == "4") | (train$digit == "9")
train49 = train[sel,]
train49$digit = factor(train49$digit)
start_time <- Sys.time()
rf49 = randomForest(digit ~ ., data = train49, ntree=150)
end_time <- Sys.time()
tiempo = end_time - start_time
rf49

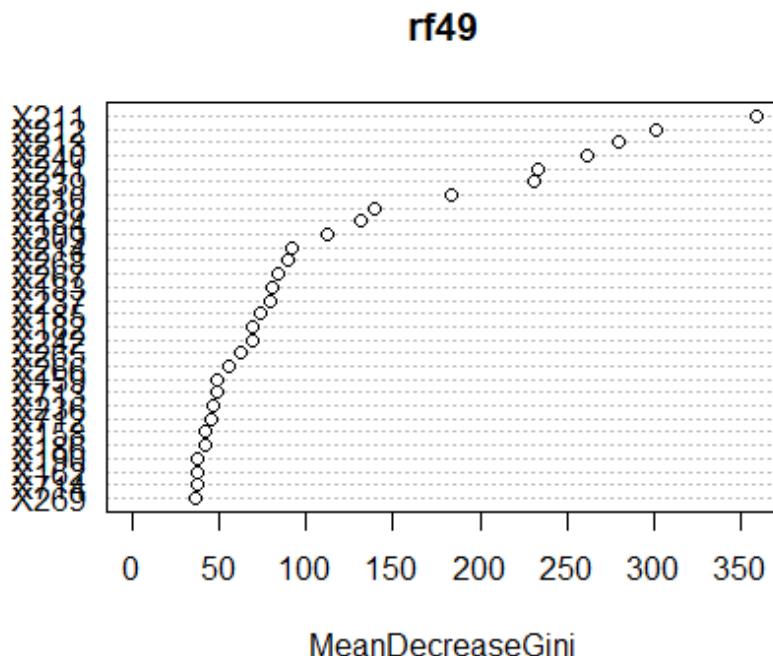
##
## Call:
## randomForest(formula = digit ~ ., data = train49, ntree = 150)
##           Type of random forest: classification
##                   Number of trees: 150
## No. of variables tried at each split: 28
##
##       OOB estimate of error rate: 1.27%
## Confusion matrix:
##      4   9 class.error
## 4 5757   85  0.01454981
## 9   65 5884  0.01092621
```

En mi ordenador la obtención de la solución tardó 2.98 minutos. El error de clasificación en este caso es del 1.27% (OOB). Los errores de clasificación en este problema se reducen respecto al caso con los 10 dígitos. El número de cuatros mal clasificados son 85, en el problema inicial hubo 114 cuatros que fueron clasificados como 9, y 98 nueves clasificados como 4.

Pregunta 5

Indica los 10 píxeles (variables x) más importantes utilizadas por el RF del apartado 4 para diferenciar entre 4 y 9. Indica gráficamente a qué coordenadas corresponden en la matriz de 28x28.

```
varImpPlot(rf49)
```



Los diez píxeles más importantes son:

```
var_imp=order(-rf49$importance)[1:10]
rownames(rf49$importance)[var_imp]

## [1] "X211" "X212" "X213" "X240" "X241" "X239" "X210" "X238" "X184" "X209"
```

Su importancia ha sido:

```
rf49$importance[var_imp]

## [1] 359.2700 300.5498 279.1668 261.4855 233.0833 231.3648 183.3135 139.3824
## [9] 131.5874 111.8809
```

Vamos a dibujar su posición:

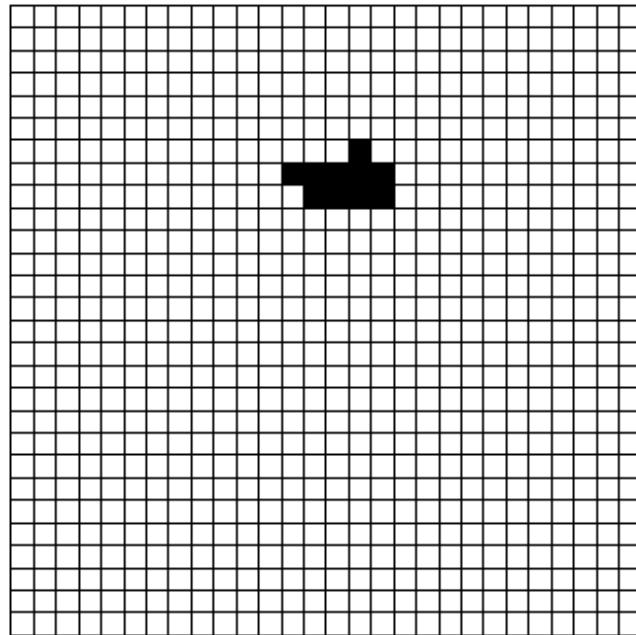
```
par(mar = rep(1,4))

xy = expand.grid(1:28, 1:28) # coordenadas en La rejilla de 28 x 28
```

```
z = rf49$importance/max(rf49$importance) # nivel de gris del número elegido
z = as.numeric(rf49$importance>111.88) # nivel de gris del número elegido

plot(0, 0, type = "n", xlab = "", ylab = "", axes =FALSE,
      xlim = c(0, 28), ylim = c(0, 28), asp =1) #crea ejes y no dibuja nada

rect(xy[,1]-1,27-xy[,2],xy[,1],28-xy[,2],col=gray(1-z)) #dibuja 784
rectangulos
```



La instrucción `as.numeric(rf49$importance>111.88)` crea un vector de 0 y 1. El valor 0 para las variables X (píxel) con importancia menor que 111.88 y 1 para las 10 más importantes. El límite 111.88 se ha obtenido de la importancia de las variables.

Pregunta 6

A partir del conjunto “test” crea el data.frame que contenga únicamente las observaciones correspondientes a los dígitos 4 y 9. Evalúa el modelo RF construido en el apartado anterior con este nuevo data.frame de test, calculando la matriz de confusión y el error de clasificación global.

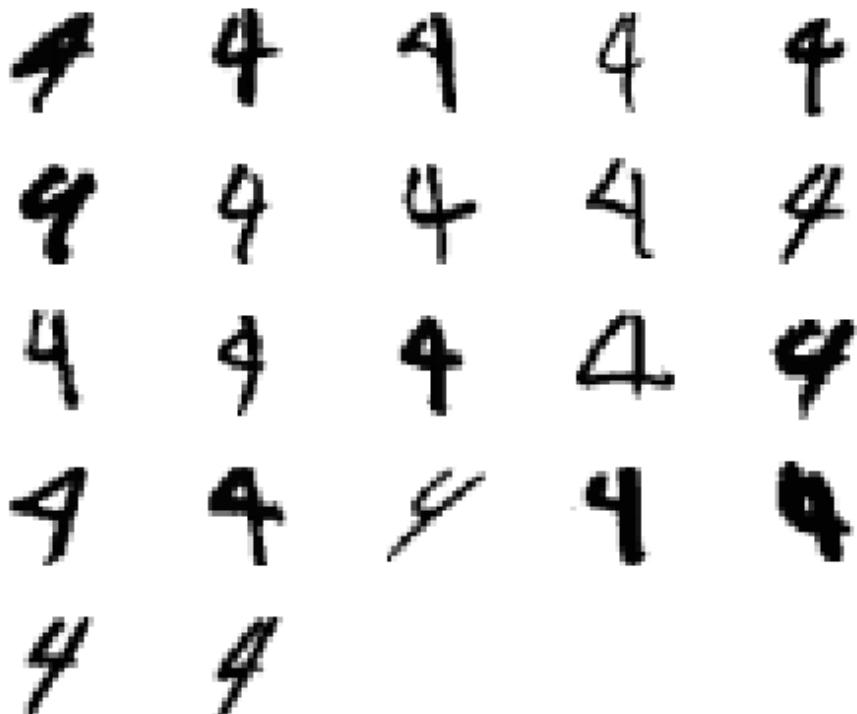
```
sel = test$digit %in% c("4", "9")
test49 = test[sel,]
test49$digit = factor(test49$digit)
pred49 = predict(rf49, newdata=test49)
t1=table(test49$digit,pred49)
(t2 = cbind(t1, class.error = (rowSums(t1)-diag(t1))/rowSums(t1)))

##      4   9 class.error
## 4 960  22  0.02240326
## 9 12 997  0.01189296

acerto2 = sum(diag(t2))/sum(t1)
paste0("El acierto global ", round(acerto2*100, 1), "%")

## [1] "El acierto global 98.3%"
```

El error medio es 1.7%. Hay más errores para clasificar el 4 (2.2%) que el 9 (1.2%).



q q g g
Q Q G G
q q G G

Pregunta 7

Para entender la lógica que utiliza el algoritmo de Random Forest en la clasificación de dígitos realiza el siguiente análisis. Con el data.frame obtenido en el apartado 4, calcula y dibuja el árbol de clasificación con $cp = 0.01$. Interpreta el árbol.

```
library(rpart)
library(rpart.plot)
library(rattle)

## Loading required package: tibble

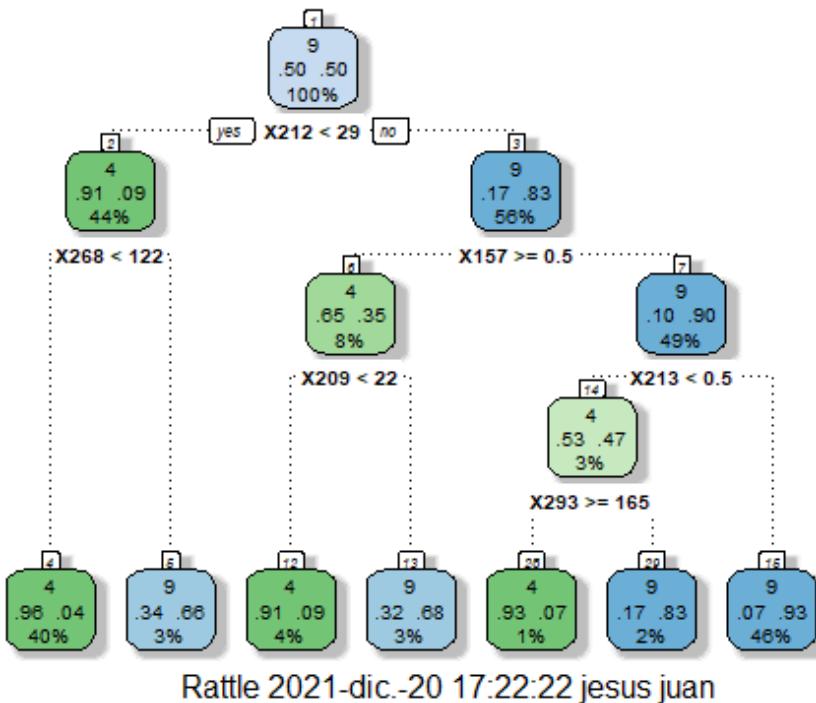
## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Versión 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Escriba 'rattle()' para agitar, sacudir y rotar sus datos.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
## 
##     importance

t49 = rpart(digit~, data=train49)
fancyRpartPlot(t49)
```



Las variables que utiliza el árbol son:

X157 X209 X212 X213 X268 X293

La interpretación es sencilla, aquí se hace un breve resumen, utilizar las notas de clase en caso de dudas:

- El nodo principal nos indica que hay una proporción muy parecida de 4 y 9. Ligeramente superior de 9, por eso identifica el nodo de partida con un 9.
- La primera variable que utiliza para clasificar es el nivel de gris del píxel 212 (X212), si es menor que 29 lo identifica como 4 y en caso contrario como 9. En el nodo 2 de la izquierda se encuentra el 44% de las observaciones, de las cuales en 91% son cuatros y el 9% son nueves. El nodo 3 de la derecha cumple que X212 es mayor que 29 y contiene el 56% de las observaciones, el 83% de las cuales son nueves y el resto cuatros.
- Aplica la regla de partición al nodo 2 y consigue el nodo 4 y 5.
- Aplica la regla de partición al nodo 3 y obtiene los nodos 6 y 7.
- El número de nodos finales (hojas) son 7. Son una partición de los datos. Cada nodo final nos da información de la clase mayoritaria (4 o 9), del porcentaje de cada una y de la proporción de observaciones que han caído en ese nodo. Por ejemplo, el nodo 4 contiene el 40% de las observaciones, son mayoritariamente cuatros (el 96%) y sólo el 4% son nueves. Si siguiendo las reglas de partición del árbol, una imagen que clasificada en el nodo 4, será identificada como un 4.

Un resumen del árbol se consigue con la instrucción `printcp()`

```
printcp(t49)

##
## Classification tree:
## rpart(formula = digit ~ ., data = train49)
##
## Variables actually used in tree construction:
## [1] X157 X209 X212 X213 X268 X293
##
## Root node error: 5842/11791 = 0.49546
##
## n= 11791
##
##          CP nsplit rel_error xerror      xstd
## 1 0.726292     0  1.00000 1.00000 0.0092932
## 2 0.047415     1  0.27371 0.27491 0.0063755
## 3 0.024820     2  0.22629 0.22749 0.0058780
## 4 0.021739     3  0.20147 0.20524 0.0056177
## 5 0.012752     4  0.17973 0.18110 0.0053121
## 6 0.010000     6  0.15423 0.16433 0.0050832
```

El error de clasificación es $0.1542 \times 0.4955 = 0.076$. Este valor se puede obtener calculando la matriz de confusión.

```
pred49a = predict(t49,type="class")
t1a=table(train49$digit,pred49a)
(t2a = cbind(t1a,class.error = (rowSums(t1a)-diag(t1a))/rowSums(t1a)))

##      4      9 class.error
## 4 5177  665  0.11383088
## 9  236 5713  0.03967053

acierto3 = sum(diag(t1a))/sum(t1a)
paste0("El acierto global ", round(acierto3*100, 1), "%")

## [1] "El acierto global 92.4%"
```

El error de un árbol es 7.6% Hay más errores para clasificar el 4 (11.4%) que el 9 (3.97%).

Pregunta 8

Utiliza el conjunto test creado en el apartado 6 y evalúa el árbol estimado en el apartado anterior. Calcula la matriz de confusión y el error medio. Compara los resultados con los obtenidos en el apartado 6 e interpreta las diferencias.

```
pred = predict(t49,newdata = test49,type = "class")
t1b=table(test49$digit,pred)
(t2b = cbind(t1b,class.error = (rowSums(t1b)-diag(t1b))/rowSums(t1b)))

##      4    9 class.error
## 4 846 136  0.13849287
## 9  44 965  0.04360753

acerto4 = sum(diag(t1b))/sum(t1b)
paste0("El acierto global ", round(acerto4*100, 1), "%")

## [1] "El acierto global 91%"
```

Los resultados son muy parecidos a los obtenidos en el apartado 6. En general deben ser ligeramente superiores los obtenidos con el subconjunto test respecto a los errores de entrenamiento (train) que es lo que ocurre aquí.

Los errores de clasificación del árbol son del 9%, y se reducen considerablemente utilizando el algoritmo random Forest donde el error es 1.7% como se vio en el apartado 6.

Pregunta 9

Añade al estudio realizado algún aspecto “original” que consideres de interés. El alumno debe plantear una cuestión (de interés) diferente a las formuladas y que esté relacionada con los datos del problema y resolverla. Esta pregunta es abierta y tiene como objetivo completar el análisis realizado en la tarea. Debe ser breve, la misma extensión que el resto de las preguntas.

Esta pregunta será evaluada de manera libre, se tendrá en cuenta la originalidad de la propuesta y su interés para completar al resto del ejercicio.

Pregunta 10

Haz un resumen del análisis realizado en esta tarea, indicando las conclusiones que consideres más relevantes.

- En este ejercicio se estudia el método de RandomForest para clasificar las cifras del cero al nueve escritas a mano. Los resultados indican que el método clasifica correctamente el 96.7 % de las observaciones y se equivoca el 3.3%.
- No todos los dígitos son igual de fácil de reconocer. Los más difíciles son el 8 y 9 que presentan cerca del 5% de errores de clasificación y los más sencillo el 0 y 1, para los cuales el error se reduce al 1.2% (aprox.).
- Si particularizamos el análisis a la clasificación de dos dígitos (4 y 9) los errores se reducen considerablemente. En principio el problema de clasificación es más sencillo debido al reducido número de opciones. El error medio es 1.7%. Hay más errores para clasificar el 4 (2.2%) que el 9 (1.2%).
- El estudio de un *árbol de clasificación* construido con las observaciones correspondientes a los dígitos 4 y 9 indica dos aspectos importantes: (1) Un único árbol tiene un porcentaje de error muy superior al randomForest. (2) Los píxeles que se utilizan para clasificar un dígito son muy escaso. En el ejemplo analizado del 4 y 9, el árbol con grado de complejidad $cp = 0.01$ sólo utiliza la información de seis píxeles.
- Para completar el análisis se debería comparar el método con otras alternativas de machine learning para seleccionar el procedimiento con menores errores de clasificación. También sería útil tener en cuenta aspectos computacionales: tiempo de cálculo y necesidades de memoria RAM. RandomForest (creo) que tanto en memoria como tiempo de cálculo presenta ventajas frente a las redes neuronales.