

# Ejercicios LAB 1 Y 2

## Cuestiones L1-2 y L3

MÉTODOS ESTADÍSTICOS

AARÓN HERNÁNDEZ ÁLVAREZ

## Contenido

EJERCICIOS LABORATORIO .....	4
Ejercicio 1 .....	4
Apartado a) .....	4
Apartado b) .....	5
Apartado c) .....	6
Ejercicio 2 .....	7
Disposiciones preliminares .....	7
Apartado a) .....	7
Apartado b) .....	8
Apartado c) .....	9
Apartado d) .....	11
Apartado e) .....	13
Adición de clase .....	16
Parte 1 .....	16
Parte 2 .....	17
Parte 3 .....	17
Conclusiones finales del ejercicio .....	18
Ejercicio 3 .....	19
Disposiciones Preliminares .....	19
Apartado a) .....	20
Apartado b) .....	20
Apartado c) .....	21
Apartado d) .....	22
Apartado e) .....	24
Conclusiones finales del ejercicio. ....	25
Ejercicio 4 .....	27
Disposiciones preliminares. ....	27
Apartado a) .....	27
Apartado b) .....	29
Apartado c) .....	30
Apartado d) .....	31
Apartado e) .....	32
Apartado f) .....	33
Apartado g) .....	34
Ejercicio 5 .....	35
Disposiciones preliminares .....	35

Apartado a) .....	35
Apartado b) .....	37
Adición del alumno .....	38
Ejercicio 6 .....	40
Disposiciones preliminares .....	40
Apartado a) .....	40
Apartado b) .....	42
Apartado c) .....	44
Conclusiones finales del ejercicio .....	46
Ejercicio 7 .....	47
Disposiciones preliminares .....	47
Apartado a) .....	47
Apartado b) .....	48
Apartado c) .....	49
Apartado d) .....	50
Apartado e) .....	51
Apartado f) .....	52
Ejercicio 8 .....	55
Disposiciones preliminares .....	55
Apartados a) y b) .....	55
Apartado c) .....	56
Apartados d) y e) .....	57
Ejercicio 9 .....	59
Apartado a) .....	59
Apartados b), c) y d) .....	60
Conclusiones finales del ejercicio .....	70
CUESTIONES LECTURAS .....	71
Cuestión L2.1 .....	71
Apartado a) .....	71
Apartado b) .....	72
Apartado c) .....	73
Apartado d) .....	73
Apartado e) .....	75
Cuestión L2.2 .....	76
Apartado a) .....	76
Apartado b) .....	78
Apartado c) .....	78
Apartado d) .....	78

Apartado e) .....	79
Apartado f) .....	79
Apartado g) .....	79
Cuestion L2.3 .....	80
Conclusiones finales del ejercicio .....	84
Cuestion 3.....	85

## EJERCICIOS LABORATORIO

### Ejercicio 1

**Ejercicio 1:** Analizar con el comando **search()** los paquetes presentes en el entorno de trabajo. Con **library(help=package)**, seleccionar el paquete **datasets**, y, dentro de los distintos conjuntos de datos, visualizar en la consola los contenidos de varios de ellos con distintas características (tipos de variables, series, etc.).

- Analizar cómo están estructurados los datos para familiarizarse con ellos.
- Distinguir claramente en su contenido aquellos que contengan factores y vectores.
- Visualizar y direccionar su contenido y realizar algunos cálculos sencillos sobre el mismo.
- Generar, utilizando **R Markdown**, un report de laboratorio que recoja la sesión y explicar en él los resultados que se han obtenido. Utilizar aquellos trozos de código R empotrados (code chunks) con sintaxis **knitr** que se consideren necesarios para este fin.

#### Apartado a)

Creemos una variable “A”, en insertamos en ella el resultante del comando search.

```
A<-search()
A
## [1] ".GlobalEnv"          "package:stats"       "package:graphics"
## [4] "package:grDevices"   "package:utils"       "package:datasets"
## [7] "package:methods"     "Autoloads"           "package:base"
```

Probamos el comando list sin elementos, e insertamos sus resultados en una nueva variable B.

```
B<-list()
B
## list()
```

Borramos la variable B, insertando previamente sus valores en otra nueva denominada C.

```
C<-B
C
## list()
rm(B)
```

Revisamos los objetos creados con objects()

```
objects()
```

```
## [1] "A" "C"
```

Cargamos una librería nueva y empleamos `search()` para verificar que se ha creado.

```
library(Matrix)
search()
```

```
## [1] ".GlobalEnv"      "package:Matrix"    "package:stats"
## [4] "package:graphics" "package:grDevices" "package:utils"
## [7] "package:datasets" "package:methods"   "Autoloads"
## [10] "package:base"
```

Apartado b)

Cargamos un dataset en una variable. Usaremos la tabla de datos BEPS, del dataset `carData` como ejemplo:

```
BEPS<-read.csv(url("https://vincentarelbundock.github.io/Rdatasets/csv/carData/BEPS.csv"))
```

Mostramos el cabecero de la tabla con `head()`.

```
head(BEPS)
```

```
##      X                vote age economic.cond.national economic.cond
.household
## 1 1 1 Liberal Democrat  43                                3
3
## 2 2                Labour  36                                4
4
## 3 3                Labour  35                                4
4
## 4 4                Labour  24                                4
2
## 5 5                Labour  41                                2
2
## 6 6                Labour  47                                3
4
##      Blair Hague Kennedy Europe political.knowledge gender
## 1      4      1      4      2                2 female
## 2      4      4      4      5                2  male
## 3      5      2      3      3                2  male
## 4      2      1      3      4                0 female
## 5      1      1      4      6                2  male
## 6      4      4      2      4                2  male
```

Insertamos el cabecero en una nueva variable `BEPS_head`

```
BEPS_head<-head(BEPS)
objects()
```

```
## [1] "A"          "BEPS"       "BEPS_head" "C"
```

#### Apartado c)

Con el comando subset(), seleccionamos aquellos cuyo genero sea masculino de la tabla e insertamos el encabezado de la tabla resultante en la variable "BEPS\_male\_head"

```
BEPS_male_head<-subset(BEPS, gender=="male")
View(BEPS_male_head)
```

## Ejercicio 2

**Ejercicio 2:** El Data Set “*MplsStops*” de la librería *carData* contiene datos de incidencias de personas implicadas en actuaciones policiales por el Departamento de Policía de Minneapolis en 2017. Se pide:

- Analizar su contenido y visualizar los factores y vectores.
- Explicar el uso del comando ***subset()*** y emplearlo para obtener un subconjunto de este data set que con tenga los vectores *race*, *gender* y *neighborhood* para el caso de actuaciones derivadas de accidentes de tráfico:

```
datos_seleccionados<-subset(datos[problem=="traffic",],
                             select = c(race, gender, neighborhood))
```

- Utilizando el comando ***fable()*** analizar los diferentes porcentajes de accidentes de tráfico según raza y género.
- Visualizar con el comando gráfico ***pie()*** los resultados del apartado anterior.
- Encontrar en qué zona de Minneapolis se registraron más accidentes.

### Disposiciones preliminares

```
#Ubicamos el directorio de trabajo.
setwd("F:/ME/Practicas/Entrega_1/Ejercicios R")
library(carData)
library(knitr)
help(MplsStops)
```

```
## starting httpd help server ... done
```

### Apartado a)

Cargamos la librería *carData*, preparamos el dataSet *MplsStops* y realizamos ***summary*** sobre éste para verificar las características.

El dataset dispone de más de 50000 registros sobre detenciones, indicando sexo, raza, motivo de la detención fecha y hora de la detención entre otros. Asimismo, no indica la comisaría donde se ejecutó la detención pero si el barrio.

```
#APARTADO A
#Cargamos y anclamos la librería
GEN_DATA<-MplsStops
summary(GEN_DATA)
```

```
##           idNum           date           proble
m
## 17-000003:      1   Min.       :2017-01-01 00:00:42   suspicious:2
5822
## 17-000007:      1   1st Qu.:2017-03-29 08:35:09   traffic      :2
6098
## 17-000073:      1   Median  :2017-06-17 18:46:47
## 17-000092:      1   Mean    :2017-06-23 19:57:49
## 17-000098:      1   3rd Qu.:2017-09-18 18:32:06
## 17-000111:      1   Max.    :2017-12-31 23:52:35
```



```
## (Other) :51914
##      MDC      citationIssued personSearch vehicleSearch
## MDC :43699    NO :15899      NO :38462    NO :40579
## other: 8221   YES : 3211      YES : 5237    YES : 3120
##              NA's:32810      NA's: 8221    NA's: 8221
##
##
##
##
##      preRace      race      gender
## Unknown      :28337   Black      :15220   Female :10015
## Black        : 6805   White      :11703   Male   :27131
## White        : 6004   Unknown    : 9219   Unknown: 6492
## Native American: 908   East African: 2188   NA's    : 8282
## Latino       : 528    Latino     : 1858
## (Other)      : 1117   (Other)    : 3511
## NA's        : 8221   NA's       : 8221
##      lat      long      policePrecinct      neig
hborhood
## Min.      :44.89   Min.      :-93.33   Min.      :1.000   Downtown We
st: 4409
## 1st Qu.:44.95   1st Qu.: -93.29   1st Qu.:2.000   Whittier
: 3328
## Median :44.98   Median  :-93.28   Median :3.000   Near - Nort
h : 2256
## Mean    :44.97   Mean    :-93.27   Mean    :3.257   Lyndale
: 2154
## 3rd Qu.:45.00   3rd Qu.: -93.25   3rd Qu.:4.000   Jordan
: 2075
## Max.    :45.05   Max.    :-93.20   Max.    :5.000   Hawthorne
: 2031
##
##                                     (Other)
:35667
```

```
attach(GEN_DATA)
```

### Apartado b)

Necesitamos un filtro previo para la realización del ejercicio, ya que son más de 50000 registros. Para el caso escogemos la raza, el género y el vecindario. Generamos nuestro subset filtrando estos campos por el problema “tráfico”. Una vez hecho, desenganchamos el dataset. Mediante el uso del comando kable, mostramos la tabla.

### #APARTADO B

#### #Hacemos Attach y creamos nuestro subset

```
DATA_PR_2<-subset(GEN_DATA[problem=="traffic",], select = c(race
,gender,neighborhood))
DATA_PR_2_HEAD<-head(DATA_PR_2)
detach(GEN_DATA)
```

```
#Mostramos las columnas y cambiamos su nombre  
kable(DATA_PR_2[1:10,])
```

	race	gender	neighborhood
6825	White	Female	Whittier
6827	White	Female	Downtown West
6828	East African	Male	Downtown West
6830	Other	Female	Marcy Holmes
6831	White	Male	Nicollet Island - East Bank
6832	Black	Male	Folwell
6834	NA	NA	Cedar Riverside
6835	Black	Male	St. Anthony East
6836	Black	Male	Downtown West
6841	White	Male	Logan Park

```
names(DATA_PR_2)<-c("raza","genero","barrio")  
kable(DATA_PR_2[1:10,])
```

	raza	genero	barrio
6825	White	Female	Whittier
6827	White	Female	Downtown West
6828	East African	Male	Downtown West
6830	Other	Female	Marcy Holmes
6831	White	Male	Nicollet Island - East Bank
6832	Black	Male	Folwell
6834	NA	NA	Cedar Riverside
6835	Black	Male	St. Anthony East
6836	Black	Male	Downtown West
6841	White	Male	Logan Park

```
# Volvemos a anclar  
attach(DATA_PR_2)
```

#### Apartado c)

Creamos un gráfico de tipo pie mostrando los accidentes que existen por raza. Para eso, generamos en una variable nueva un nuevo dataframe, que partirá del original, y que consistirá en una tabla de frecuencias de accidentes de tráfico según raza. Una vez hecho esto, generamos el gráfico resultante usando como variable la frecuencia y como label la raza.

```
# Tabla de frecuencias de accidentes  
ftable(raza)
```

```
## raza Black White Unknown East African Latino Native American
Other Asian
##
##      9762  8167      756      1545  1327      339
811    507
```

```
DATA_PR_2_RACE<-data.frame(ftable(raza))
kable(DATA_PR_2_RACE)
```

raza	Freq
Black	9762
White	8167
Unknown	756
East African	1545
Latino	1327
Native American	339
Other	811
Asian	507

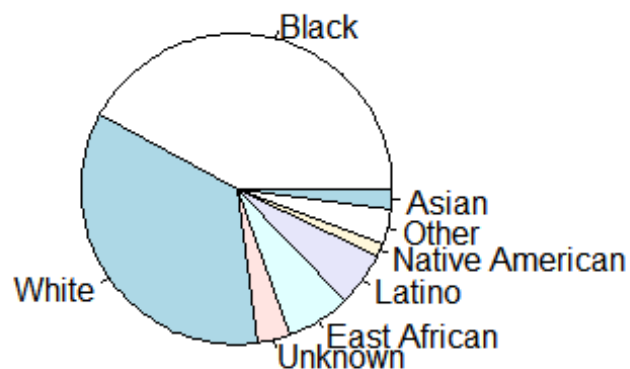
```
# APARTADO D
# Grafico de accidentes por raza
attach(DATA_PR_2_RACE)
```

```
## The following object is masked from DATA_PR_2:
```

```
##
```

```
##      raza
```

```
pie(Freq, labels=raza)
```



#### Apartado d)

Ahora realizamos un gráfico de tipo pie representando la frecuencia de accidentes por género. En primer lugar, creamos una tabla de frecuencias basándonos en género. A continuación creamos un dataframe de dicho resultado y lo insertamos en una variable. Seguidamente realizamos una visión más estilizada de la tabla con kable. Por último generamos el gráfico pie.

```
detach()
```

```
# Grafico de accidentes por genero
```

```
fable(genero)
```

```
## genero Female Male Unknown
```

```
##
```

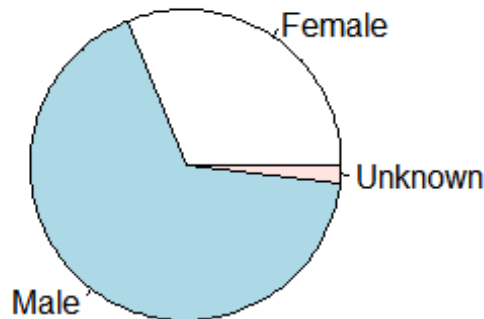
```
##           7200 15564      435
```

```
DATA_PR_2_GENDER<-data.frame(fable(genero))
```

```
kable(DATA_PR_2_GENDER)
```

genero	Freq
Female	7200
Male	15564
Unknown	435

```
pie(DATA_PR_2_GENDER$Freq, labels=DATA_PR_2_GENDER$genero)
```



A continuación realizaremos un nuevo dataframe para visualizar el porcentaje de accidentes respecto a cada raza. En primer lugar inicializamos nuestra variable donde irán todas las frecuencias de raza. Luego, en otra variable realizamos la conversión porcentual. A continuación renombramos las columnas de nuestra nueva tabla. Seguidamente insertamos los valores de raza de la tabla original en la nueva tabla para que no quede con NA. Realizamos Kable para verificar y generamos el gráfico pie.

```
# Calculamos suma de frecuencia de accidentes
ACC_SUM<-sum(DATA_PR_2_RACE$Freq)
# Creamos un dataframe con el porcentaje respecto al total de accidentes
PCTG_RACE<-100*DATA_PR_2_RACE/ACC_SUM

## Warning in Ops.factor(left, right): '*' not meaningful for factors

names(PCTG_RACE)<-c("raza", "porcentaje")
attach(PCTG_RACE)

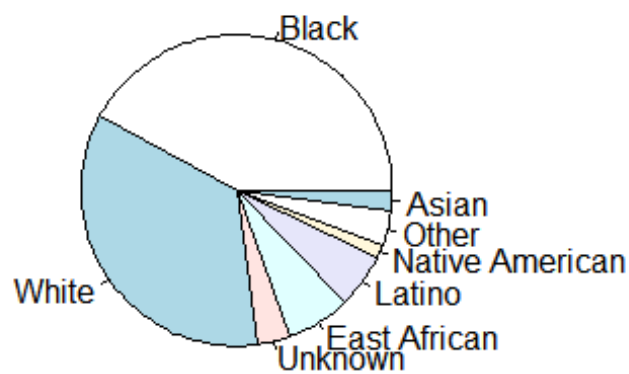
## The following object is masked from DATA_PR_2:
##
##      raza

raza<-DATA_PR_2_RACE$raza
kable(PCTG_RACE)
```

raza	porcentaje
NA	42.052210

```
NA 35.181356
NA 3.256655
NA 6.655467
NA 5.716378
NA 1.460326
NA 3.493581
NA 2.184027
```

```
pie(porcentaje, labels=raza)
```



#### Apartado e)

Creamos un dataframe con el porcentaje respecto al total de accidentes. A continuación generamos la tabla correspondiente con kable.

```
#APARTADO E
# Maximo barrio por accidentes.
# Tabla de accidentes por barrio
DATA_PR_2_NEIGH<-data.frame(ftable(barrio))
kable(DATA_PR_2_NEIGH)
```

barrio	Freq
Armatage	12
Audubon Park	348
Bancroft	21
Beltrami	158

Bottineau	281
Bryant	19
Bryn - Mawr	47
Camden Industrial	22
CARAG	325
Cedar - Isles - Dean	99
Cedar Riverside	262
Central	304
Cleveland	150
Columbia Park	87
Como	314
Cooper	56
Corcoran	140
Diamond Lake	34
Downtown East	119
Downtown West	1071
East Harriet	89
East Isles	418
East Phillips	428
ECCO	120
Elliot Park	120
Ericsson	31
Field	19
Folwell	745
Fulton	40
Hale	13
Harrison	156
Hawthorne	1246
Hiawatha	74
Holland	850
Howe	50
Humboldt Industrial Area	6
Jordan	1262
Keewaydin	10
Kenny	53
Kenwood	145
King Field	597
Lind - Bohanon	72

Linden Hills	94
Logan Park	276
Longfellow	162
Loring Park	193
Lowry Hill	126
Lowry Hill East	888
Lyndale	1282
Lynnhurst	159
Marcy Holmes	1281
Marshall Terrace	262
McKinley	457
Mid - City Industrial	201
Midtown Phillips	252
Minnehaha	18
Morris Park	3
Near - North	1319
Nicollet Island - East Bank	782
North Loop	466
Northeast Park	196
Northrop	17
Page	9
Phillips West	260
Powderhorn Park	309
Prospect Park - East River Road	384
Regina	34
Seward	130
Sheridan	217
Shingle Creek	26
St. Anthony East	152
St. Anthony West	347
Standish	32
Steven's Square - Loring Heights	559
Sumner - Glenwood	53
Tangletown	449
University of Minnesota	134
Ventura Village	330
Victory	233
Waite Park	100



Webber - Camden	271
Wenonah	11
West Calhoun	35
Whittier	1977
Willard - Hay	675
Windom	221
Windom Park	303

Una vez realizado el dataframe, buscamos en éste el barrio que mas accidentes registra con `which.max`.

```
# Calculos del maximo de accidentes y su barrio
max(DATA_PR_2_NEIGH$Freq)

## [1] 1977

which.max(DATA_PR_2_NEIGH$Freq)

## [1] 84

DATA_PR_2_NEIGH$barrio[which.max(DATA_PR_2_NEIGH$Freq)]

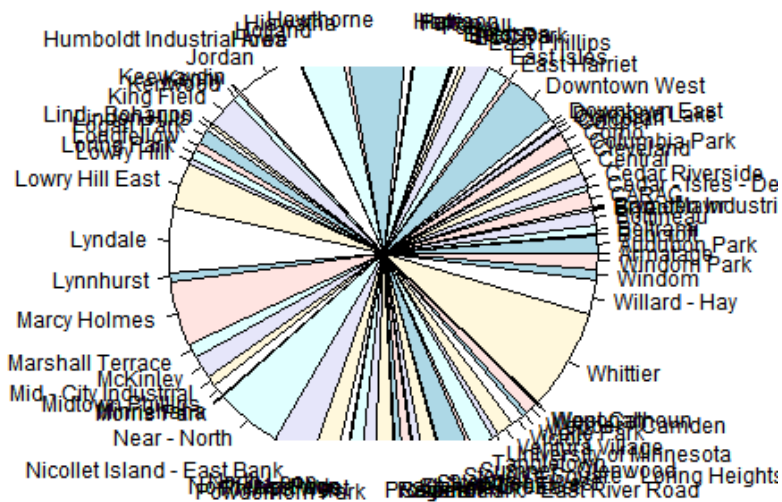
## [1] Whittier
## 87 Levels: Armatage Audubon Park Bancroft Beltrami Bottineau
... Windom Park
```

Adición de clase

Parte 1

Asimismo creamos además, otro gráfico extra, que constará de los barrios. Ejecutamos la función `pie` sobre éste, pero sale demasiado sobrecargado.

```
# Grafico de accidentes por barrio
pie(DATA_PR_2_NEIGH$Freq, labels=DATA_PR_2_NEIGH$barrio, cex=0.7
5, radius=1.25)
```



## Parte 2

Para arreglarlo, vamos a crear un gráfico pie para indicar los barrios que más accidentes tengan, incluyendo a aquellos que tengan accidentes por debajo de una cantidad. Para ello, filtramos aquellos cuya frecuencia de accidente sea mayor que 1000.

```
#Nuevo Dataset con Los barrios ajustados por encima de 1000
DATA_PR_2_NEIGH_RED<-DATA_PR_2_NEIGH[DATA_PR_2_NEIGH$Freq > 1000
,]
kable(DATA_PR_2_NEIGH_RED)
```

	barrio	Freq
20	Downtown West	1071
32	Hawthorne	1246
37	Jordan	1262
49	Lyndale	1282
51	Marcy Holmes	1281
58	Near - North	1319
84	Whittier	1977

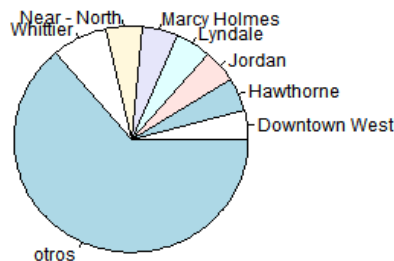
## Parte 3

Una vez creado el filtro para aquellos barrios por debajo de 1000, mostramos el gráfico pie resultante partiendo del nuevo dataframe creado para tal.

```
#Sumamos y añadimos al Dataset DATA_PR_2_NEIGH
DPR2_N<-sum(DATA_PR_2_NEIGH$Freq)
DPR2_NR<-sum(DATA_PR_2_NEIGH_RED$Freq)
DATA_PR_2_NEIGH_NEO<-rbind(DATA_PR_2_NEIGH_RED, data.frame(barri
```

```
o="otros", Freq=DPR2_N-DPR2_NR))
pie(DATA_PR_2_NEIGH_NEO$Freq, labels=DATA_PR_2_NEIGH_NEO$barrio,
cex=0.75, main = "Accidentes por barrio")
```

### Accidentes por barrio



### Conclusiones finales del ejercicio

Según el estudio realizado, se pueden esclarecer las siguientes conclusiones:

Respecto a los accidentes por raza:

- Las dos mayorías étnicas representadas en el informe en la ciudad, la caucásica y la afroamericana no presentan especial diferencia en cuanto a la frecuencia de accidentes. Si bien es verdad que no se tiene acceso a los datos poblacionales concretos para realizar una valoración precisa de la situación en función a la demografía.
- De entre las minorías étnicas presentes, destacan latinos y africanos orientales, ostentando casi el 40% de los accidentes restantes si se obviasen las dos etnias mencionadas en el punto anterior. De igual manera no se disponen de datos demográficos fiables.

Respecto a los accidentes por género:

- En este caso existe una clara diferencia entre géneros, puesto que se dan muchos más accidentes entre hombres que entre mujeres, suponiendo éstos un 67,1% de los detenidos por accidentes o infracciones de tráfico en la ciudad, contra un escaso 31,1%, y aquellos de género no indicado un casi inapreciable 1,8% del total. Nuevamente resaltar que no se disponen de más datos que puedan ayudar a verificar si esta situación puede o no estar condicionada por algún otro factor.

### Ejercicio 3

**Ejercicio 3:** Utilizar el Data Set “*Davis*” de la librería *carData*, que proporciona los datos de hombres y mujeres que realizan ejercicio regularmente de peso y altura, tanto medidos como comunicados por los/las afectados/as. El Data Set contiene datos no disponibles (NA’s). Analizar la estructura de los datos correspondientes y:

- Estudiar y aplicar posibles soluciones para los NA's.
- Encontrar las variaciones de altura y peso reales en función del género. Calcular las medias, medianas y desviación estándar correspondientes.
- Analizar las variaciones de altura y peso comunicadas en función del género. Calcular las medias, medianas y desviación estándar correspondientes.
- Visualizar gráficamente, utilizando **boxplot()**, una comparativa de los datos de peso medido y peso declarado por un lado y de la altura medida y la altura declarada por otro. Establecer justificadamente las conclusiones.
- Encontrar si hay diferencias significativas entre lo medido y declarado según el género y analizar las posibles formas de corregirla

## Disposiciones Preliminares

Iniciamos el directorio de trabajo de nuestro proyecto, cargamos la librería y vemos lo que existe en el DataSet Davis.

```
#Iniciamos el espacio de trabajo en Nuestro directorio de proyec
to
```

```
setwd("F:/ME/Practicas/Entrega 1/Ejercicios R")
```

```
library(knitr)
```

```
library(carData)
```

```
help(Davis)
```

```
## starting httpd help server ... done
```

## summary(Davis)

##	sex	weight	height	repwt
##	F:112	Min. : 39.0	Min. : 57.0	Min. : 41.00
.	:148.0			
##	M: 88	1st Qu.: 55.0	1st Qu.:164.0	1st Qu.: 55.00
Qu.:160.5				
##		Median : 63.0	Median :169.5	Median : 63.00
ian :168.0				
##		Mean : 65.8	Mean :170.0	Mean : 65.62
n :168.5				
##		3rd Qu.: 74.0	3rd Qu.:177.2	3rd Qu.: 73.50
Qu.:175.0				
##		Max. :166.0	Max. :197.0	Max. :124.00
.	:200.0			
##				NA's :17
s :17				

### Apartado a)

Existe un problema con los valores NA, y es que estos pueden desencadenar problemas de cálculo y aproximación a la hora de realizar las consultas y estimaciones de nuestros informes.

Para evitar esto tenemos varias opciones:

- Eliminarlos del muestreo final
- Omitirlos
- Ajustarlos a los percentiles mayores indicando que es un valor máximo

Para esta práctica optaremos por la tercera opción. Ajustaremos en segunda instancia los valores obtenidos en las gráficas, con los NA incluidos, al percentil 90 de las muestras y comprobaremos la diferencia.

*#Soluciones NA's.*

*#1. Eliminarlos #2. Omitirlos --> opción 2*

```
datos<-na.omit(Davis)
names(datos)<-c("sexo", "peso", "altura", "peso_m", "altura_m")
attach(datos)
summary(datos)
```

##	sexo		peso		altura		peso_m		a
			ltura_m						
##	F:99	Min.	: 39.0	Min.	: 57.0	Min.	: 41.00	Min.	:
			:148.0						
##	M:82	1st Qu.:	56.0	1st Qu.:	164.0	1st Qu.:	55.00	1st	:
			Qu.:161.0						
##		Median	: 63.0	Median	:169.0	Median	: 63.00	Medi	:
			an :168.0						
##		Mean	: 66.3	Mean	:170.2	Mean	: 65.68	Mean	:
			:168.7						
##		3rd Qu.:	75.0	3rd Qu.:	178.0	3rd Qu.:	74.00	3rd	:
			Qu.:175.0						
##		Max.	:166.0	Max.	:197.0	Max.	:124.00	Max.	:
			:200.0						

### Apartado b)

Creamos los dataset correspondientes para generar los boxplots necesarios para la práctica. Generaremos 4 en función del sexo, de pesos y alturas declarados por los participantes. Compilaremos la media, la media y la desviación estándar mediante el uso de la función *aggregate()* nuevamente.

*#Media con aggregate, en función del sexo, de los pesos y alturas medidas del DF datos*

```
aggregate(cbind(peso_m, altura_m)~sexo, datos, mean)
```

```
##   sexo  peso_m altura_m
## 1    F 56.66667 162.3636
## 2    M 76.56098 176.2561
```

*#Mediana con aggregate, en función del sexo, de los pesos y alturas medidas del DF datos*

```
aggregate(cbind(peso_m,altura_m)~sexo,datos,median)
```

```
##   sexo peso_m altura_m
## 1    F     56     163
## 2    M     75     175
```

*#Desviación media con aggregate, en función del sexo, de los pesos y alturas medidas del DF datos*

```
aggregate(cbind(peso_m,altura_m)~sexo,datos,sd)
```

```
##   sexo  peso_m altura_m
## 1    F 6.753684 5.791363
## 2    M 12.286797 6.977585
```

#### Apartado c)

Creamos los dataset correspondientes para generar los boxplots necesarios para la práctica. Generaremos 4 en función del sexo, de pesos y alturas medidos en los participantes. Compilaremos la media, la media y la desviación estándar mediante el uso de la función *aggregate()* nuevamente.

*#Media con aggregate, en función del sexo, de los pesos y alturas declarados del DF datos*

```
aggregate(cbind(peso,altura)~sexo,datos,mean)
```

```
##   sexo  peso  altura
## 1    F 58.29293 163.6465
## 2    M 75.97561 178.0122
```

*#Mediana con aggregate, en función del sexo, de los pesos y alturas declarados del DF datos*

```
aggregate(cbind(peso,altura)~sexo,datos,median)
```

```
##   sexo peso altura
## 1    F   56   165
## 2    M   75   178
```

*#Desviación media con aggregate, en función del sexo, de los pesos y alturas declarados del DF datos*

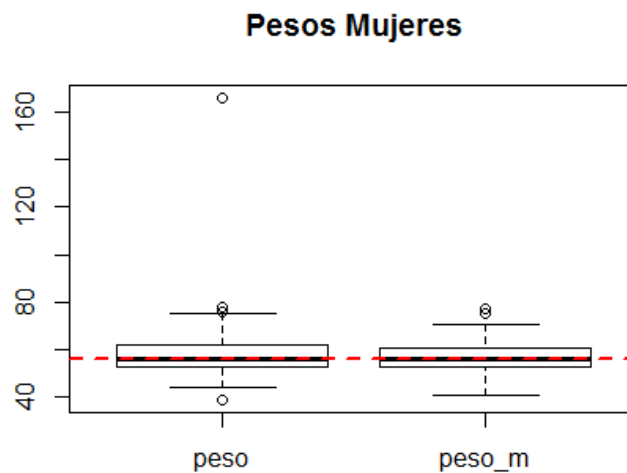
```
aggregate(cbind(peso,altura)~sexo,datos,sd)
```

```
##   sexo  peso  altura
## 1    F 12.92384 12.142280
## 2    M 12.15636 6.649039
```

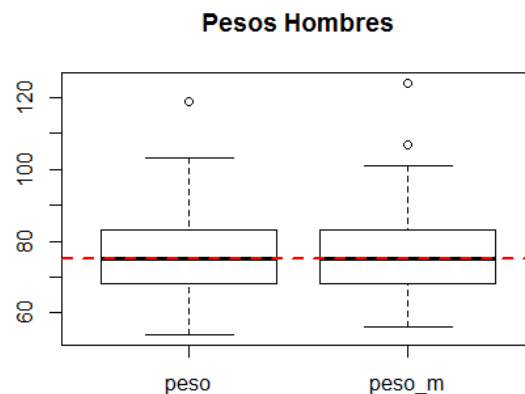
#### Apartado d)

Generamos los gráficos de cajas correspondientes. Se puede apreciar la diferencia en las mediciones, causada por los outliers, datos que salen en sobremanera del espacio muestral.

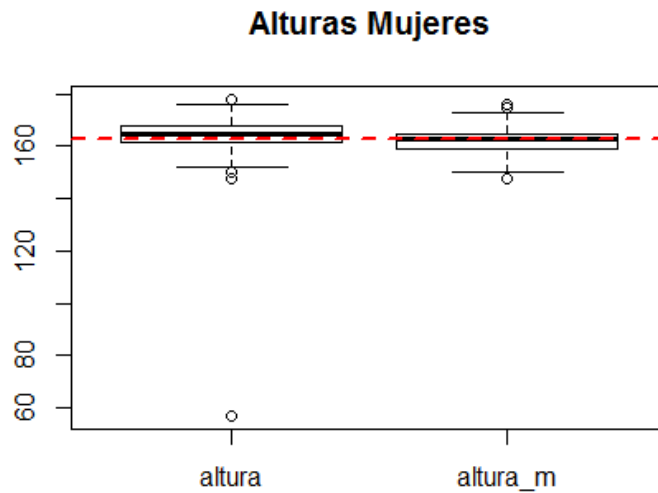
```
#Apartado D (Pesos mujeres)
#Boxplot de los pesos de las mujeres -- OPCION B
boxplot(subset(datos, select = c(peso,peso_m))[sexo=="F",], main
="Pesos Mujeres")
abline(h=median(peso_m[sexo=="F"]), col="red", lty=2, lwd=2)
```



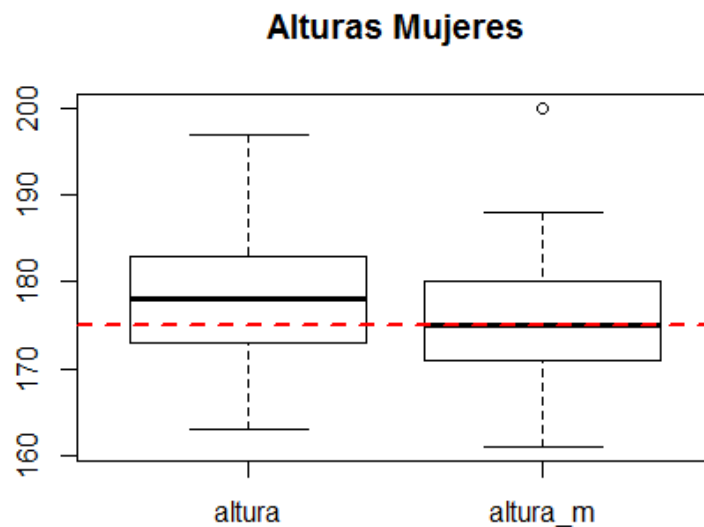
```
#Aparatdo D (Pesos hombres)
#Boxplot de los pesos de las mujeres -- OPCION B
boxplot(subset(datos, select = c(peso,peso_m))[sexo=="M",], main
="Pesos Hombres")
abline(h=median(peso_m[sexo=="M"]), col="red", lty=2, lwd=2)
```



```
#Apartado D (Alturas mujeres)
#Boxplot de los pesos de Las mujeres -- OPCION B
boxplot(subset(datos, select = c(altura,altura_m))[sexo=="F",],
main="Alturas Mujeres")
abline(h=median(altura_m[sexo=="F"]), col="red", lty=2, lwd=2)
```



```
#Apartado D (Alturas hombres)
#Boxplot de los pesos de Las mujeres -- OPCION B
boxplot(subset(datos, select = c(altura,altura_m))[sexo=="M",],
main="Alturas Hombres")
abline(h=median(altura_m[sexo=="M"]), col="red", lty=2, lwd=2)
```



Nota: es título debe ser *Altura Hombres*.



#### Apartado e)

Creemos 4 nuevos vectores, que contendrán los pesos y alturas medidos y declarados en las mujeres. Los emplearemos para verificar el cómo la corrección de outliers puede ser beneficiosa para la representación y estudio finales, pese a que *de facto* sean valores modificados artificialmente.

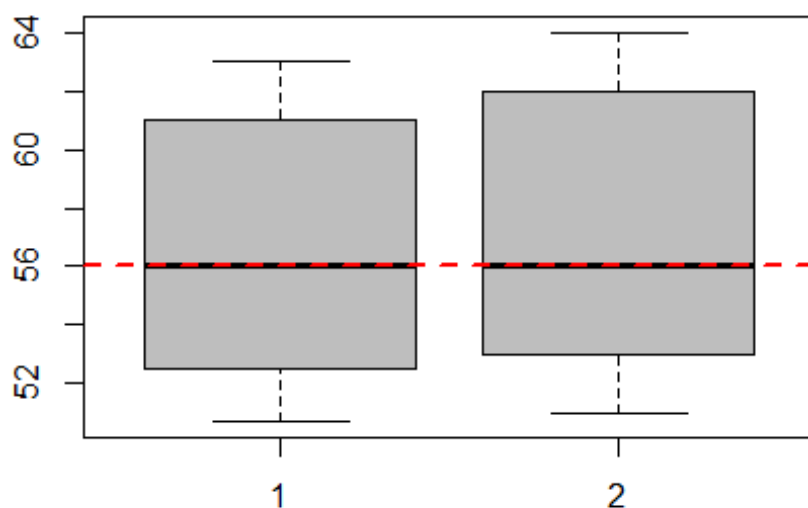
##### *#Apartado E (Pesos mujeres)*

*#Plot con las muestras en función del peso declarado con el medido*

```
x <- peso_m[sexo=="F"]
limitesup<-quantile(x,.85)
limiteinf<-quantile(x,.15)
p_corregido<-x
p_corregido[p_corregido<limiteinf] <-limiteinf
p_corregido[p_corregido>limitesup] <-limitesup

x2 <- peso[sexo=="F"]
limitesup_p2<-quantile(x2,.85)
limiteinf_p2<-quantile(x2,.15)
p2_corregido<-x2
p2_corregido[p2_corregido<limiteinf_p2] <-limiteinf_p2
p2_corregido[p2_corregido>limitesup_p2] <-limitesup_p2

boxplot(p_corregido,p2_corregido, col="grey")
abline(h=median(p2_corregido), col="red", lty=2, lwd=2)
```

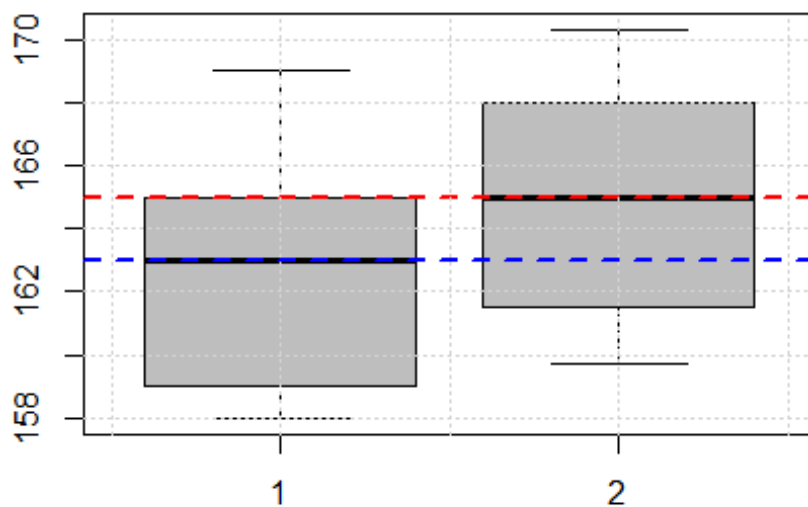


*#Plot con las muestras en funcion de la altura declarada con la medida*

```
x3 <- altura_m[sexo=="F"]
limitesup_a<-quantile(x3,.85)
limiteinf_a<-quantile(x3,.15)
a_corregido<-x3
a_corregido[a_corregido<limiteinf_a] <-limiteinf_a
a_corregido[a_corregido>limitesup_a] <-limitesup_a

x4 <- altura[sexo=="F"]
limitesup_a2<-quantile(x4,.85)
limiteinf_a2<-quantile(x4,.15)
a2_corregido<-x4
a2_corregido[a2_corregido<limiteinf_a2] <-limiteinf_a2
a2_corregido[a2_corregido>limitesup_a2] <-limitesup_a2

boxplot(a_corregido,a2_corregido, col="grey")
abline(h=median(a2_corregido), col="red", lty=2, lwd=2)
abline(h=median(a_corregido), col="blue", lty=2, lwd=2)
grid()
```



#### Conclusiones finales del ejercicio.

En este ejercicio se han realizado los diferentes estudios acerca de dos factores concretos en un espacio muestral, y se ha verificado como la toma de datos puede estar comprometida desde el inicio del estudio en sí. En lo concerniente a los apartados del ejercicio se puede decir que:

Respecto a los datos tomados:

- Se verifica una diferencia considerable tanto entre pesos como alturas declarados respecto a sus medidos, lo cual indica que la certeza de las características físicas es más bien poca por parte de los encuestados. Esto podría deberse a diversos factores. Asimismo destacar que ésta diferencia se acrecienta más en el caso de las alturas tanto para hombres como mujeres, que en la encuesta de pesos.

Respecto a la corrección de outliers:

- Es evidente el cambio en las gráficas una vez corregidos los valores disidentes (outliers). Esto resalta aún más en el caso de las mujeres, el cual disponía de varios de ellos que hacían poco visible la diferencia en los gráficos originales, destacando particularmente el caso de las alturas. Una vez corregidos, el gráfico de alturas se muestra nítido y la diferencia entre ambas medianas es completamente evidente y fácilmente apreciable.

## Ejercicio 4

**Ejercicio 4 (Opcional):** Utilizar la siguiente secuencia de comandos para leer los ficheros "empleados.txt" y "salarios.txt".

```
setwd("C:/Users/Antonio/Documents/R/Scripts R")
empleados <- read.table("empleados.txt", sep = ",", header = TRUE)
salarios <- read.table("salarios.txt", sep = "\t", header = TRUE)
names(empleados)
[1] "Num_Empleado" "Fecha_nacimiento" "Nombre" "Apellido" "Genero" "Fecha_Contrato"
names(salarios)
[1] "Num_Empleado" "Salario" "Desde_Fecha" "Hasta_Fecha"
```

Estos ficheros contienen los datos de los empleados y salarios de una empresa de Ingeniería vinculados por un campo común "Num\_Empleado".

- Analizar el contenido de los Data Frames con los comandos **tail()** y **head()**
- Razonar sobre los tipos de datos que lo integran (factores y vectores).
- Encontrar las medias, medianas y desviaciones estándar de la variable "Salario" agrupada por la variable "Num\_empleado" y encontrar el empleado que más cobra y el que menos en promedio.
- Visualizar utilizando **boxplot()** las variaciones de "Salario" dependiendo del empleado.
- Utilizar el comando **merge()** para unir los dos data frames unificados por "Num\_empleado" y repetir los apartados c) y d) para el data frame resultante.
- Con los comandos **interval()**, **now()** y **ymd()** del paquete **lubridate**, determinar la edad de los diez empleados y añadir una nueva columna con el campo "Edad" al data frame resultante del apartado anterior
- Añadir un nuevo registro al data frame del apartado e). Explicar en detalle el proceso.

Disposiciones preliminares.

```
setwd("F:/ME/Practicas/Entrega_1/Ejercicios R/Archivos")
datos_empleados <- read.table("empleados.txt", sep = ",", header = TRUE)
datos_salarios <- read.table("salarios.txt", sep = "\t", header = TRUE)
```

Apartado a)

Revisamos la tabla de empleados mediante los comando **head()** y **tail()**. Verificamos que la tabla de empleados cuenta con 6 columnas destacando los datos de cada empleado. De éstas, encontramos una ID, dos fechas, sexo, Nombres y apellido. Se pueden considerar factores evidentes el ID empleado, el sexo y las fechas.

*Datos\_salarios* por otra parte cuenta con información acerca de los contratos de los diferentes empleados mencionados arriba. Destacan los factores Numempleado, y las fechas, y el numérico salario.

```
##Analizar el contenido de Los Data Frames con Los comandos tail
() y head()
head(datos_empleados)
```

```
##   Num_Empleado Fecha_nacimiento Nombre Apellido Genero Fecha_Contrato
## 1          10001      1983-09-02   Mario Rodriguez      M
2018-06-23
## 2          10002      1984-06-02    Berta Santana      F
2011-08-02
## 3          10003      1986-12-03    Pedro Brito      M
2011-12-01
## 4          10004      1984-05-01 Carmelo Ortega      M
2018-11-28
## 5          10005      1991-01-21   Oscar Suarez      M
2018-09-10
## 6          10006      1983-04-20     Ana Priego      F
2011-08-02
```

```
tail(datos_empleados)
```

```
##   Num_Empleado Fecha_nacimiento Nombre Apellido Genero Fecha_Contrato
## 5          10005      1991-01-21   Oscar Suarez      M
2018-09-10
## 6          10006      1983-04-20     Ana Priego      F
2011-08-02
## 7          10007      1987-05-23 Cristina Bautista      F
2018-02-08
## 8          10008      1988-02-19   Samuel Lopez      M
2010-03-10
## 9          10009      1982-04-19     Sara Perez      F
2018-02-15
## 10         10010      1993-06-01     Dori Pelaez      F
2011-11-23
```

```
head(datos_salarios)
```

```
##   Num_Empleado Salario Desde_Fecha Hasta_Fecha
## 1          10001   60117  2006-06-26  2007-06-26
## 2          10001   62102  2007-06-26  2008-06-25
## 3          10001   66074  2008-06-25  2009-06-25
## 4          10001   66596  2009-06-25  2010-06-25
## 5          10001   66961  2010-06-25  2011-06-25
## 6          10001   71046  2011-06-25  2012-06-24
```

```
tail(datos_salarios)
```

```
##   Num_Empleado Salario Desde_Fecha Hasta_Fecha
## 206         10021   74759  2017-02-07  2018-02-07
## 207         10021   77519  2018-02-07  2019-02-07
## 212         10022   40000  2009-09-03  2010-09-02
## 213         10022   39935  2010-09-02  2011-09-02
## 214         10022   41348  2011-09-02  2018-01-01
## 215         10023   47883  2009-09-27  2018-09-26
```

#### Apartado b)

Realizamos un `summary` para verificar los cuartiles, percentiles, el IQR, medias y frecuencias pertinentes de cada campo según sea factor o numérico. Éste nos permite verificar que los empleados aquí indicados son de larga duración y con un salario medio bastante alto, entre otros factores.

Asimismo `summary()` puede realizar automáticamente el cálculo del IQR y, por ende, de los cuartiles 1 y 3

*##Razonar sobre Los tipos de datos que lo integran (factores y vectores).*

`summary(datos_empleados)`

```
##   Num_Empleado      Fecha_nacimiento      Nombre      Apellido
Genero
##   Min.      :10001      1982-04-19:1      Ana      :1      Bautista:1
F:5
##   1st Qu.:10003      1983-04-20:1      Berta     :1      Brito     :1
M:5
##   Median :10006      1983-09-02:1      Carmelo   :1      Lopez     :1
##   Mean    :10006      1984-05-01:1      Cristina  :1      Ortega    :1
##   3rd Qu.:10008      1984-06-02:1      Dori      :1      Pelaez    :1
##   Max.     :10010      1986-12-03:1      Mario     :1      Perez     :1
##              (Other)      :4      (Other)   :4      (Other)   :4
##   Fecha_Contrato
##   2011-08-02:2
##   2010-03-10:1
##   2011-11-23:1
##   2011-12-01:1
##   2018-02-08:1
##   2018-02-15:1
##   (Other)    :3
```

`summary(datos_salarios)`

```
##   Num_Empleado      Salario      Desde_Fecha      Hasta_Fecha
##   Min.      :10001      Min.      :39935      2006-08-03: 2      2018-01-01:
10
##   1st Qu.:10005      1st Qu.:47892      2007-08-03: 2      2007-08-03:
2
##   Median :10010      Median :60755      2008-08-03: 2      2008-08-03:
2
##   Mean    :10011      Mean    :61471      2009-08-03: 2      2009-08-03:
2
##   3rd Qu.:10016      3rd Qu.:72749      2010-08-02: 2      2010-08-02:
2
##   Max.     :10023      Max.     :99651      2011-08-02: 2      2011-08-02:
2
```

```
##                                (Other)    :174    (Other)    :  
166
```

```
##analizamos
```

### Apartado c)

Creamos tres tablas anexas unas de otras, con la media, medianas y desviación estándar de los salarios de cada uno de los empleados. De la primera extraemos el empleado que más cobra de promedio en comparación con los demás. Usamos los comandos *which.max()* y *which.min()*.

```
tapply(datos_salarios$Salario,datos_salarios$Num_Empleado,mean)
```

```
##      10001      10002      10003      10004      10005      10006      1000  
7      10008  
## 72117.54 68854.50 43030.29 53055.31 85790.90 50514.92 65945.2  
0 49307.67  
##      10009      10010      10011      10012      10013      10014      1001  
5      10016  
## 74006.79 76723.00 49782.00 46902.80 49452.86 52990.33 40000.0  
0 74995.40  
##      10017      10018      10019      10020      10021      10022      1002  
3  
## 87064.70 64313.83 47007.25 43278.20 64219.18 40427.67 47883.0  
0
```

```
tapply(datos_salarios$Salario,datos_salarios$Num_Empleado,median  
)
```

```
##      10001      10002      10003      10004      10005      10006      10007      100  
08      10009  
## 74333.0 68450.0 43478.0 52119.0 85811.0 50086.0 64019.0 48584  
.0 73023.0  
##      10010      10011      10012      10013      10014      10015      10016      100  
17      10018  
## 76799.5 50927.0 46472.5 48502.5 53228.0 40000.0 76381.0 87888  
.0 62467.0  
##      10019      10020      10021      10022      10023  
## 46860.5 43800.0 61117.0 40000.0 47883.0
```

```
tapply(datos_salarios$Salario,datos_salarios$Num_Empleado,sd)
```

```
##      10001      10002      10003      10004      10005      10006  
10007  
## 7153.5493 2932.2706 1339.9334 9033.3284 3666.0696 7154.6668 5  
891.2429  
##      10008      10009      10010      10011      10012      10013  
10014  
## 3063.2944 8621.0080 3118.0512 5229.2545 5148.5697 8116.8538 5
```

```

151.0021
##      10015      10016      10017      10018      10019      10020
10021
##      NA 2956.9745 9793.7160 5313.1197 2357.0130 2942.7560 7
296.1022
##      10022      10023
## 797.6944      NA

```

*#Creamos una tabala nueva con la media de los salarios por numero de empleado. Luego ejecutamos el which.max y which.min*

```

VectorSalario<-c(tapply(datos_salarios$Salario,datos_salarios$Num_Empleado,mean))
SueldoMax<-which.max(VectorSalario)
SueldoMin<-which.min(VectorSalario)

```

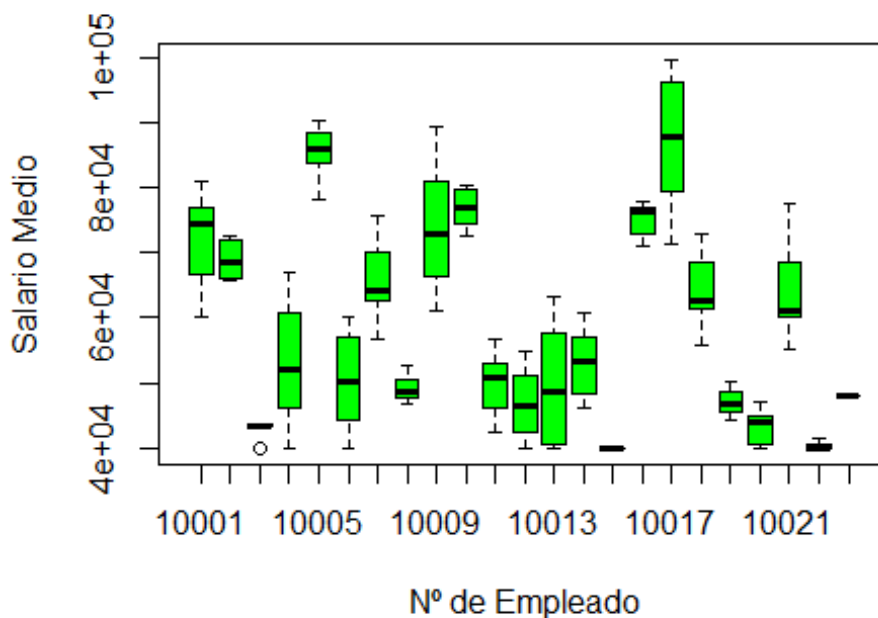
Apartado d)

Convertimos los datos de salario de la tabla *datos\_salario* en numéricos. Una vez realizada esta operación, creamos un boxplot combinado en el que aparecen todos los empleados y sus salarios medios. En éste se puede observar que el empleado de ID 10017 es el que más cobra de media con diferencia y el 10021 el que menos.

```

VectorSalario<-as.numeric(datos_salarios$Salario)
VectorEmpleado<-as.numeric(datos_salarios$Num_Empleado)
boxplot(VectorSalario~VectorEmpleado,ylab="Salario Medio",xlab="
Nº de Empleado", col="green")

```





#### Apartado e)

Repetimos el procedimiento anterior combinando las dos tablas mediante el uso del comando `merge()`. Éste nos combina automáticamente ambas tablas siguiendo una estructura de agrupación a través de un parámetro común, el ID empleado en este caso.

```
#Combinamos las dos tablas en una nueva y sobre esta un dataframe
```

```
tablanew<-merge(datos_salarios,datos_empleados)
datos_c<-data.frame(tablanew)
```

```
#Lo anclamos
```

```
attach(tablanew)
```

```
#apartado c repetido
```

```
tapply(Salario,Num_Empleado,mean)
```

```
##      10001      10002      10003      10004      10005      10006      1000
7      10008
```

```
## 72117.54 68854.50 43030.29 53055.31 85790.90 50514.92 65945.2
0 49307.67
```

```
##      10009      10010
```

```
## 74006.79 76723.00
```

```
tapply(Salario,Num_Empleado,median)
```

```
##      10001      10002      10003      10004      10005      10006      10007      100
08      10009
```

```
## 74333.0 68450.0 43478.0 52119.0 85811.0 50086.0 64019.0 48584
.0 73023.0
```

```
##      10010
```

```
## 76799.5
```

```
tapply(Salario,Num_Empleado,sd)
```

```
##      10001      10002      10003      10004      10005      10006      1000
7      10008
```

```
## 7153.549 2932.271 1339.933 9033.328 3666.070 7154.667 5891.24
3 3063.294
```

```
##      10009      10010
```

```
## 8621.008 3118.051
```

```
VSalario<-c(tapply(Salario,Num_Empleado,mean))
```

```
PMax<-which.max(VSalario)
```

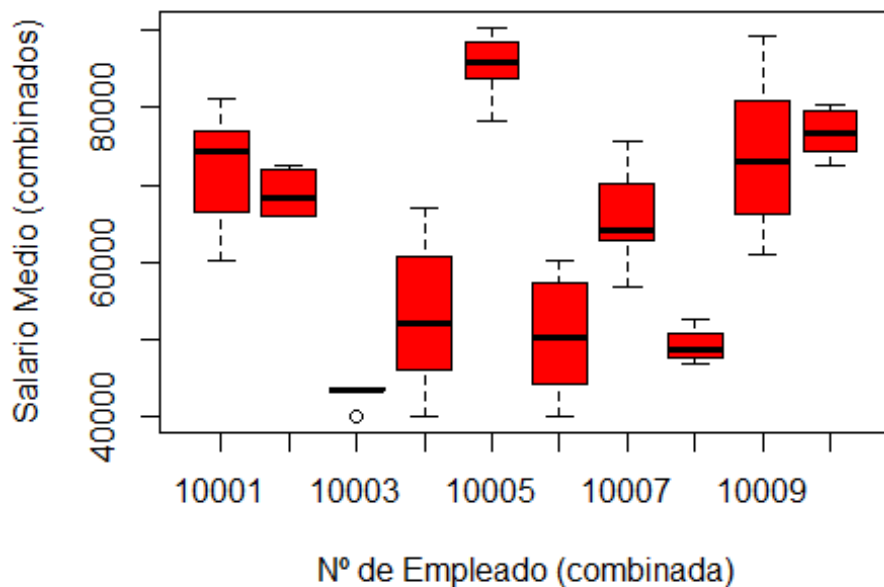
```
PMin<-which.min(VSalario)
```

```
#apartado d repetido
```

```
VectorSalarionew<-as.numeric(Salario)
```

```
VectorEmpleadonew<-as.numeric(Num_Empleado)
```

```
boxplot(VectorSalarionew~VectorEmpleadonew,
        ylab="Salario Medio (combinados)",
        xlab="Nº de Empleado (combinada)",
        col="red")
```



#### Apartado f)

Mediante el uso del paquete *lubridate()* crearemos una nueva tabla en la que comprobaremos el número que tiene cada empleado, teniendo en cuenta que habrá que filtrar los datos antes de realizar la tabla final.

Empleamos para ello las funciones indicadas en el enunciado.

```
attach(tablanew)
```

```
library(lubridate)
```

```
##
```

```
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
```

```
##
```

```
##      date
```

```
current<-as.Date(now())
```

```
vectorFechas<-c(as.Date(Fecha_nacimiento))
```

```
Anos<-c(year(current)-year(vectorFechas))
```

```
datosEdad<-data.frame(Num_Empleado=Num_Empleado,Edad=Anos)
```

```
tablanew<-merge(tablanew,datosEdad)
```

*##podemos comprobarlo creando un data frame filtrando por empleado y*

*##haciendo las medias de cada empleado (es su edad).*

```

prueba<-data.frame(tablanew)
detach()
attach(tablanew)

## The following objects are masked from tablanew (pos = 3):
##
##      Apellido, Desde_Fecha, Fecha_Contrato, Fecha_nacimiento,
##      Genero, Hasta_Fecha, Nombre, Num_Empleado, Salario

aggregate(Edad~Num_Empleado,prueba,mean)

##      Num_Empleado  Edad
## 1          10001    36
## 2          10002    35
## 3          10003    33
## 4          10004    35
## 5          10005    28
## 6          10006    36
## 7          10007    32
## 8          10008    31
## 9          10009    37
## 10         10010    26

detach()

```

#### Apartado g)

Mediante el uso del paquete *lubridate()* crearemos una nueva columna en la que comprobaremos el número de semanas laborales que cada empleado lleva trabajando en la empresa, teniendo en cuenta que habrá que filtrar los datos antes de realizar la tabla final.

Empleamos para ello las funciones indicadas en el enunciado. Posteriormente añadimos mediante el uso del comando *cbind()* la nueva columna, que empleará los registros de Fecha\_nacimiento para autogenerarse.

```

##Tiempo de trabajo en semanas según el contrato de cada cliente
en semanas Laborales
attach(tablanew)

## The following objects are masked from tablanew (pos = 3):
##
##      Apellido, Desde_Fecha, Fecha_Contrato, Fecha_nacimiento,
##      Genero, Hasta_Fecha, Nombre, Num_Empleado, Salario

from<-c(as.Date(Desde_Fecha))
to<-c(as.Date(Fecha_Contrato))
workingWeeks<-c(difftime(to,from, units = "days"))
workingWeeksArray<-as.numeric(workingWeeks,"days")
workingWeeksArray<-workingWeeksArray/5
tablanew<-cbind(tablanew,Semanas_Contrato=workingWeeksArray)

```

## Ejercicio 5

**Ejercicio 5:** Ejercicio: Leer el fichero *"casas.txt"* que incluye el precio medio de viviendas en miles de euros por localizaciones en España. Generar un vector "Precios" a partir de los datos indicados en el fichero. Realizar a continuación las siguientes operaciones:

```
A<-rank(Precios)
B<- sort(Precios)
C<- order(Precios)
Comparativa<-data.frame(Precios,A,B,C)
Comparativa
```

Explicar la diferencia entre las diferentes columnas que resultan en cada caso y obtener las casas de precio medio superior a 190.000 €

### Disposiciones preliminares

Para esta práctica, el alumno ha decidido emplear el paquete *sqldf()* para realizar los dataframes correspondientes empleando el lenguaje de consultas SQL.

```
#Iniciamos el espacio de trabajo en Nuestro directorio de proyecto
setwd("F:/ME/Practicas/Entrega_1/Ejercicios R")
library(sqldf)

## Loading required package: gsubfn
## Loading required package: proto
## Loading required package: RSQLite

library(knitr)
#Cargamos el dataset Correspondiente de casa.txt en una variable
Datcasas<-read.table("/ME/Practicas/Entrega_1/Ejercicios R/Archivos/casas.txt", header=T, dec=',', sep="\t")
#Anclamos la variable Datcasas para tener su datos disponibles
attach(Datcasas)
```

### Apartado a)

Creamos los vectores correspondientes para ver cómo reacciona cada uno de ellos ante el Sort, el Rank y el Order.

Como se puede observar, order realiza un orden correspondiente de la tabla en orden descendente. Sort ordena según su relevancia y Rank crea un ranking de registros en función de mayores y menores.

Una vez comprobadas, generamos un dataframe con el resultado final de las funciones en el que se nos indican las ciudades ordenadas con sus precios de menor a mayor y el ranking que ostenta cada una de ellas.

```

#Creamos nuestro vector precio
Datca_precios<-c(Precio)
#Generamos una tabla de frecuencias de precios
fable(Datca_precios)

## Datca_precios 95 101 117 121 157 162 164 188 201 211 325
##
##              1   1   1   1   1   1   1   2   1   1   1

#Generamos una variable de A con la funci?n RANK y hacemos kable sobre ella
pre_rango<-rank(Datca_precios)
kable(pre_rango)

#Genarmos una variable de B con la funci?n SORT y hacemos kable sobre ella
pre_sort<-sort(Datca_precios)
kable(pre_sort)

#Genarmos una variable de C con la funci?n ORDER y hacemos kable sobre ella
c<-order(Datca_precios)
Loc_orden<-Localizacion[order(Precio)]
kable(Loc_orden)

```

x

---

Cadiz  
 Zaragoza  
 Albacete  
 Lanzarote  
 Barcelona  
 Castellon  
 Badalona  
 Teruel  
 Tenerife  
 Salamanca  
 Malaga  
 Madrid

```

#Generamos un dataframe final con lo anterior.
pre_orden<-data.frame(Loc_orden,pre_sort,pre_rango)
#cambiamos los nombres de las columnas
names(pre_orden)<-c("Localizacion","Precios","Ranking")
kable(pre_orden)

```

Localizacion	Precios	Ranking
Cadiz	95	12.0
Zaragoza	101	10.0
Albacete	117	5.0
Lanzarote	121	6.0
Barcelona	157	7.0
Castellon	162	2.0
Badalona	164	11.0
Teruel	188	8.5
Tenerife	188	1.0
Salamanca	201	3.0
Malaga	211	8.5
Madrid	325	4.0

#### Apartado b)

Ahora, usando sqldf, generamos tres tablas nuevas que provengan cada una de una consulta SQL para aquellas cuyo precio sea mayor que 190, todas clasificadas por precio de mayor a menor y ordenadas por orden y rango, respectivamente.

```
library(sqldf)
Compara_mas190<-sqldf("SELECT * FROM pre_orden WHERE Precios > 190 ORDER BY Precios DESC")
kable(Compara_mas190)
```

Localizacion	Precios	Ranking
Madrid	325	4.0
Malaga	211	8.5
Salamanca	201	3.0

```
Compara_Pre_maymen<-sqldf("SELECT * FROM pre_orden ORDER BY Precios DESC")
kable(Compara_Pre_maymen)
```

Localizacion	Precios	Ranking
Madrid	325	4.0
Malaga	211	8.5
Salamanca	201	3.0
Teruel	188	8.5
Tenerife	188	1.0
Badalona	164	11.0
Castellon	162	2.0
Barcelona	157	7.0

Lanzarote	121	6.0
Albacete	117	5.0
Zaragoza	101	10.0
Cadiz	95	12.0

```
Compara_Pre_rank<-sqldf("SELECT * FROM pre_orden ORDER BY Ranking ASC")
kable(Compara_Pre_rank)
```

Localizacion	Precios	Ranking
Tenerife	188	1.0
Castellon	162	2.0
Salamanca	201	3.0
Madrid	325	4.0
Albacete	117	5.0
Lanzarote	121	6.0
Barcelona	157	7.0
Teruel	188	8.5
Malaga	211	8.5
Zaragoza	101	10.0
Badalona	164	11.0
Cadiz	95	12.0

```
#Por ultimo crearemos una columna que añadiremos al nuevo dataframe donde se indique la desviacion
#porcentual respecto al precio medio
detach()
```

### Adición del alumno

En este añadido se plantea crear un nuevo dataframe empleando sqldf() para mostrar las ciudades cuyo precio de venta medio que se encuentra en el rango del 15% superior o inferior respecto a la media general.

Para ello, generamos un vector nuevo con la media, y la desviación respecto a la media, lo añadimos a la tabla original, creamos un dataframe con el resultado y por último filtramos con consultas SQL los registros que nos interesen.

```
#Añadido al ejercicio para comprobación de SQL
```

```
attach(pre_orden)
#Calculamos la media y la desviacion porcentual respecto al precio medio
mediapre<-mean(Precios)
mediapre

## [1] 169.1667
```

```

desv_pre<-mediapre-Precios
porcen<-(mediapre-Precios)/Precios
desv<-porcen*100
# Añadimos la columna a un nuevo dataframe
Compara_desv<-data.frame(Localizacion,Precios,Ranking)
Compara_desv<-cbind(Compara_desv,desv,desv_pre)
names(Compara_desv)<-c("Localizaci?n","Precios","Ranking","Desv",
,"Montante")
kable(Compara_desv)

```

Localizaci?n	Precios	Ranking	Desv	Montante
Cadiz	95	12.0	78.070175	74.166667
Zaragoza	101	10.0	67.491749	68.166667
Albacete	117	5.0	44.586895	52.166667
Lanzarote	121	6.0	39.807162	48.166667
Barcelona	157	7.0	7.749469	12.166667
Castellon	162	2.0	4.423868	7.166667
Badalona	164	11.0	3.150406	5.166667
Teruel	188	8.5	-10.017731	-18.833333
Tenerife	188	1.0	-10.017731	-18.833333
Salamanca	201	3.0	-15.837479	-31.833333
Malaga	211	8.5	-19.826224	-41.833333
Madrid	325	4.0	-47.948718	-155.833333

*# Buscamos las viviendas cuya desviaci?n de precio est? e un rango del 15% respecto a la media*

*# Generamos una tabla.*

```

x<-sqldf("SELECT * FROM Compara_desv where Desv < 15 AND Desv >
-15 ORDER BY DESV")
kable(x)

```

Localizaci?n	Precios	Ranking	Desv	Montante
Teruel	188	8.5	-10.017731	-18.833333
Tenerife	188	1.0	-10.017731	-18.833333
Badalona	164	11.0	3.150406	5.166667
Castellon	162	2.0	4.423868	7.166667
Barcelona	157	7.0	7.749469	12.166667



## Ejercicio 6

**Ejercicio 6:** El fichero *"Accidentes\_1969\_1984\_UK.txt"* contiene datos de series temporales referidas a conductores fallecidos o con lesiones graves en UK entre los años 1966 y 1984. En enero de 1983 entró en vigor la ley que obliga a la utilización del cinturón de seguridad. Entre otras variables se dispone de las siguientes:

- *DriversKilled* : conductores de automóvil muertos.
- *front*: Pasajeros asientos delanteros muertos o gravemente heridos.
- *rear*: Pasajeros asientos delanteros muertos o gravemente heridos.
- *VanKilled*: número de conductores de furgonetas
- *law*: vigencia (0/1) de obligatoriedad del cinturón

Se pide:

- a) Analizar la serie temporal de fallecidos en accidentes, encontrar sus zonas de máximo valor y visualizar el efecto de entrada en vigor de la ley.
- b) Analizar las relaciones existentes entre los conductores fallecidos y las víctimas según estuvieran en los asientos delanteros o traseros. Explicar y estudiar en detalle el alcance de las suposiciones establecidas en los posibles modelos.
- c) Analizar y evaluar el efecto que tienen las furgonetas ligeras (tipo Van) en el conjunto de accidentes mortales antes y después de la aplicación de la ley. Justificar las respuestas

### Disposiciones preliminares

```
setwd("F:/ME/Practicas/Entrega_1/Ejercicios R/")
library(knitr)
accidentes<-read.table("/ME/Practicas/Entrega_1/Ejercicios R/Arc
hivos/Accidentes_1969_1984_UK.txt", header=T, dec=',', sep=",")
```

### Apartado a)

Para el primer apartado, es necesario tener en cuenta que la aplicación de la ley viene definida a partir de febrero de 1983, por lo cual será necesario especificar a partir de qué fecha comienzan las conclusiones de la aplicación de la ley. Generaremos un grafica que muestre el número de accidentes por año que han ido sucediéndose durante el período que se muestra en el dataset() empleado en el ejercicio.

Para hacerlo, construimos un plot simple desde cero, que llegue desde el principio hasta el fin de los registros de fallecidos en accidentes de tráfico. A continuación, establecemos la fecha límite a partir de la cual generaremos dos líneas medias que representarán el número de fallecidos antes y después de la aprobación de la ley. Asimismo, añadiremos a la gráfica una línea vertical que represente el momento de aplicación.

Una vez realizado esto, crearemos un gráfico de caja para mostrar la diferencia de una forma más visual.

```

#Anclamos el dataframe
attach(accidentes)

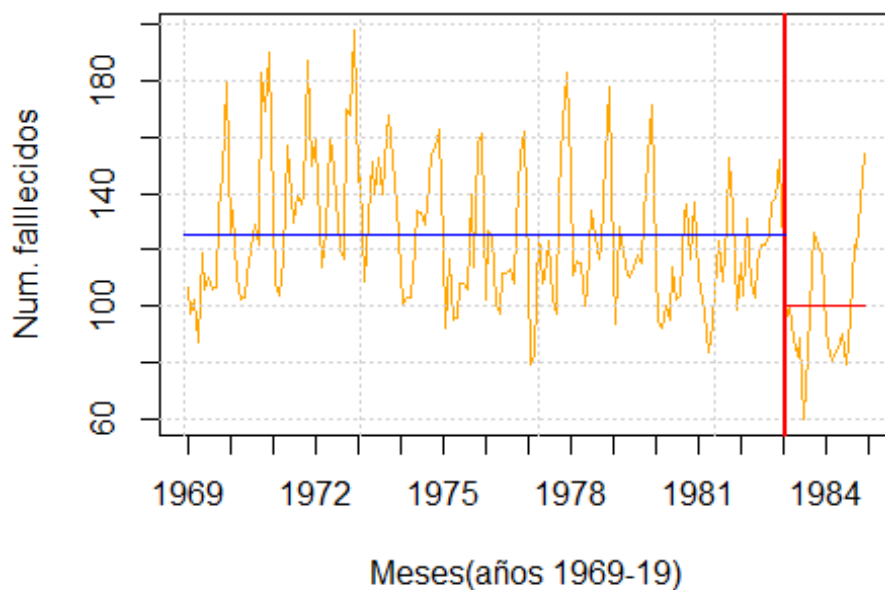
#Generamos el plot.
plot(1:length(DriversKilled), DriversKilled,
     xaxt="n", type="l",
     col="orange",xlab="Meses(años 1969-19)",
     ylab="Num. fallecidos")
years<-seq(1969,1985,1)
axis(1,at=seq(1,length(DriversKilled)+12,12),
     labels <-as.character(years))
grid()

#Establecemos la fecha de aplicación de la ley.
Febrero_83<-(1983-1969)*12+2
abline(v=Febrero_83, col="red",lwd=2)

#Creamos las medias de la aplicación de la ley.
media_no_ley<-mean(DriversKilled[1:Febrero_83])
media_ley<-mean(DriversKilled[Febrero_83:length(DriversKilled)])

#Line(c(x1x2),c(y1,y2),)
lines(c(0,Febrero_83),c(media_no_ley,media_no_ley), col="blue")
lines(c(Febrero_83,192),c(media_ley,media_ley), col="red")

```



```

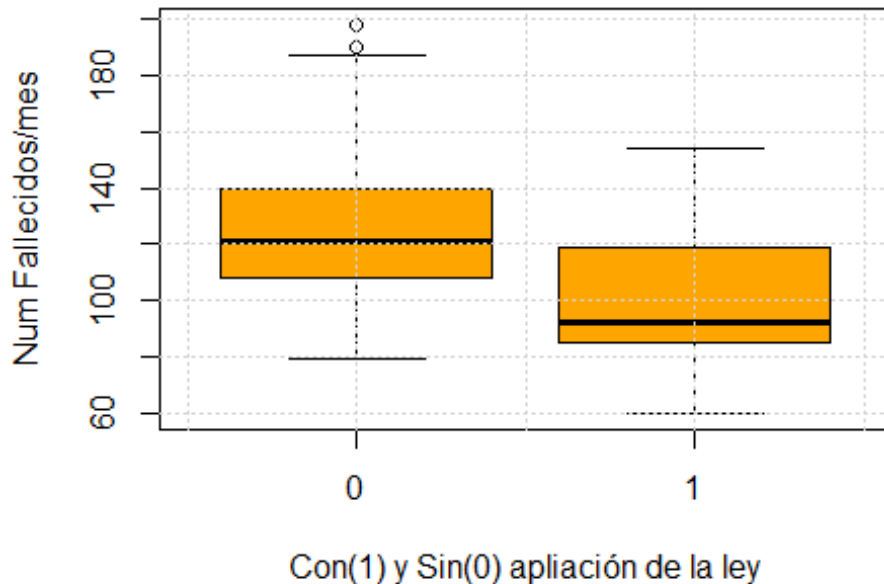
#Creamos un boxplot de mabas antes y después de la aplicación de
la ley.
boxplot(DriversKilled~law,

```

```

xlab="Con(1) y Sin(0) aplicación de la ley",
ylab="Num Fallecidos/mes", col="orange")
grid()

```



#### Apartado b)

Al igual que en el apartado a, generaremos dos graficas con dos modelos en cada una, que aglutinen la regresión lineal en función de las muertes en la parte frontal y trasera de los vehículos, tanto antes como después de la ley. Éstas partirán de una inicial que

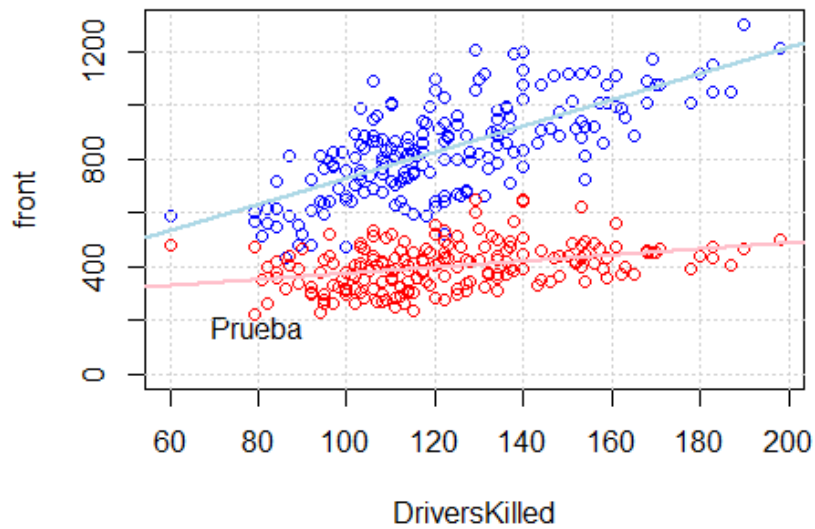
Para ello emplearemos el uso de la función `lm()`, que generará un modelo lineal en función de los fallecidos por año computado.

Asimismo probaremos la opción de etiquetar las gráficas usando una etiqueta "Prueba", que irá por coordenadas.

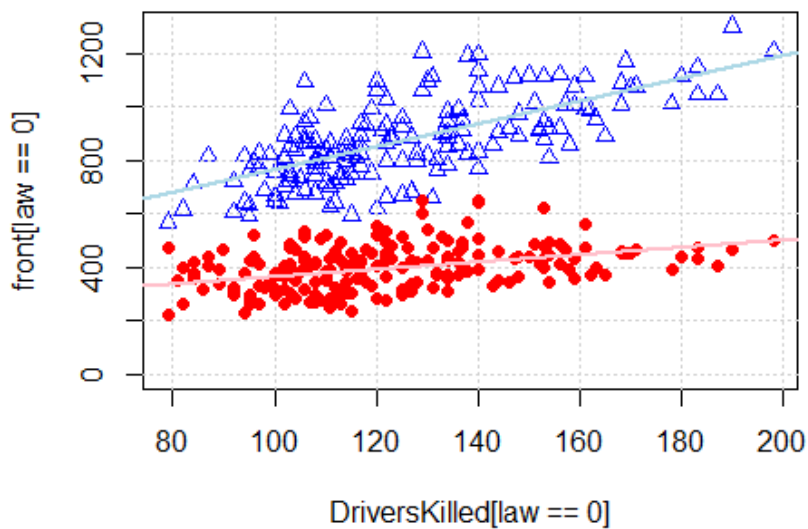
```

#Generamos el plot.
plot(DriversKilled,front, col="blue",
      ylim=c(0,max(front)), text(80,180,label="Prueba"))
grid()
modelo1<-lm(front~DriversKilled)
abline(modelo1,col="lightblue",lwd=2)
points(DriversKilled,rear, col="red")
modelo2<-lm(rear~DriversKilled)
abline(modelo2,col="pink",lwd=2)

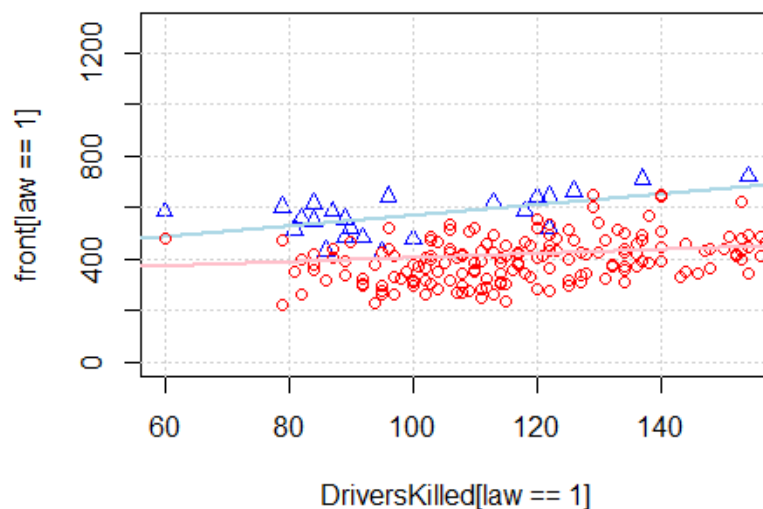
```



```
#Generamos el segundo plot con los puntos.(ANTES DE LA LEY)
plot(DriversKilled[law==0],front[law==0], col="blue",
      ylim=c(0,max(front)), pch=2)
grid()
modelo1<-lm(front[law==0]~DriversKilled[law==0])
abline(modelo1,col="lightblue",lwd=2)
points(DriversKilled,rear, col="red", pch=19)
modelo2<-lm(rear[law==0]~DriversKilled[law==0])
abline(modelo2,col="pink",lwd=2)
```



```
#Generamos el segundo plot con los puntos.(DESPUES DE LA LEY)
plot(DriversKilled[law==1],front[law==1], col="blue",
      ylim=c(0,max(front)), pch=2)
grid()
modelo1<-lm(front[law==1]~DriversKilled[law==1])
abline(modelo1,col="lightblue",lwd=2)
points(DriversKilled,rear, col="red", pch=1)
modelo2<-lm(rear[law==1]~DriversKilled[law==1])
abline(modelo2,col="pink",lwd=2)
```



#### Apartado c)

Repetimos el procedimiento del apartado a), teniendo en cuenta que en este caso no se tendrán en cuenta los conductores sino el tipo de vehículo, que serán furgonetas.

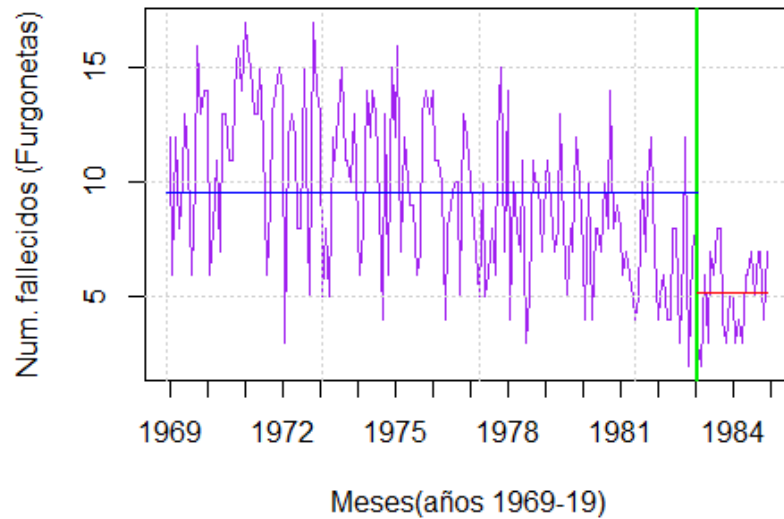
```
#Generamos el plot.
plot(1:length(VanKilled), VanKilled,
      xaxt="n", type="l",
      col="purple",xlab="Meses(años 1969-19)",
      ylab="Num. fallecidos (Furgonetas)")
years<-seq(1969,1985,1)
axis(1,at=seq(1,length(VanKilled)+12,12),
      labels <-as.character(years))
grid()

#Generamos el boxplot de furgonetas
#Establecemos la fecha de aplicación de la ley.
Febrero_83<-(1983-1969)*12+2
abline(v=Febrero_83, col="green2",lwd=2)

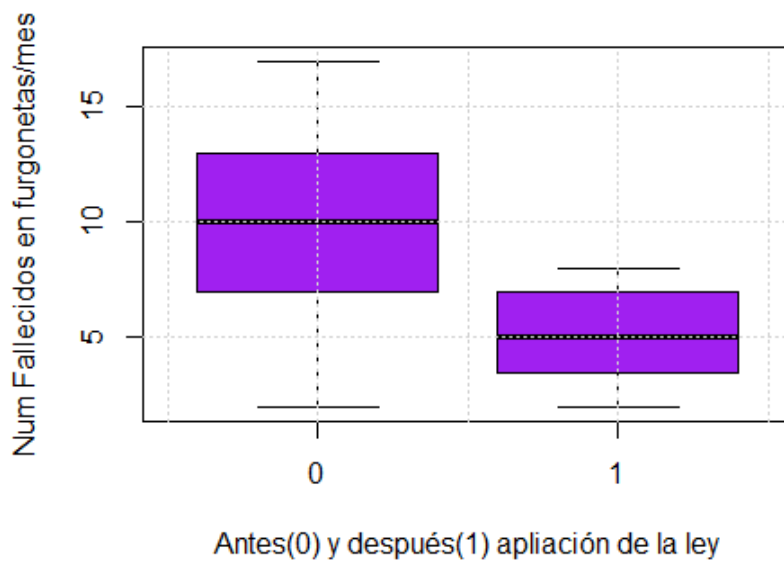
#Creamos las medias de la aplicación de la ley.
media_no_ley_f<-mean(VanKilled[1:Febrero_83])
```

```
media_ley_f<-mean(VanKilled[Febrero_83:length(DriversKilled)])

#Creamos las líneas de las medias.
lines(c(0,Febrero_83),c(media_no_ley_f,media_no_ley_f), col="blue")
lines(c(Febrero_83,192),c(media_ley_f,media_ley_f), col="red")
```



```
boxplot(VanKilled~law,
        xlab="Antes(0) y después(1) aplicación de la ley",
        ylab="Num Fallecidos en furgonetas/mes", col="purple")
grid()
```



### Conclusiones finales del ejercicio

En este ejercicio se ha podido comprobar el efecto de la aplicación de una ley sobre una población, y se pueden extraer una serie de conclusiones concretas respecto a los efectos de su aplicación:

- La aplicación de la normativa ha conllevado un descenso considerable en el número de fallecidos, tanto en accidentes de automóvil, como en accidentes de furgoneta. Esto puede estar debido a que la obligatoriedad de emplear un cinturón de seguridad, sumada a los estándares de homologación que esto conlleva en las inspecciones técnicas, y a la reticencia de los conductores de hacer frente al pago de las posibles sanciones que la nueva normativa pudiera haber implicado.
- En referencia al número de fallecidos es importante destacar el hecho de que desde la entrada en vigor, tanto los accidentes en la parte trasera, como en la frontal se han reducido considerablemente, llegando a casi desaparecer las injerencias en el caso de los conductores situados en la segunda. Esto puede ser debido a que ante la ausencia de contención ante un impacto, los ocupantes de la parte delantera podían fallecer más fácilmente por las consecuencias de la propia inercia del impacto (ser expelidos fuera de la cabina, impactar contra los airbags o el salpicadero, etc)
- En el caso de las furgonetas el descenso es notable, reduciéndose la media de forma clara en cerca de un 50% respecto a cotas anteriores a la aplicación de la ley, por los motivos anteriormente explicado.

## Ejercicio 7

**Ejercicio 7:** El fichero "Ventas\_Provincia.txt" contiene datos de ventas en euros de una empresa productora de cereales a distintas provincias españolas durante el año 2012. Se desea realizar un análisis de estos datos para valorar los procesos. Se pide:

- Cantidades totales y las medias anuales de ventas por provincia.
- Provincia en la que más se vende y en la que menos.
- Estudiar la evolución de las ventas de las provincias de Cáceres, Madrid y Barcelona en el segundo semestre de 2012.
- Utilizando los comandos gráficos de base de R, visualizar la evolución temporal de los datos del apartado c)
- Alternativamente, utilizando **ggplot2()** realizar una visualización de la evolución mensual de los datos del apartado c), tanto absolutos como relativos al total de ventas de la empresa. Explicar las distintas soluciones adoptadas.
- Realizar cambios en la estética, la escala y el tema en el apartado e). Explicar las ventajas y diferencias en cada caso.

### Disposiciones preliminares

*#Iniciamos el espacio de trabajo en Nuestro directorio de proyecto*

```
setwd("F:/ME/Practicas/Entrega_1/Ejercicios R")
```

```
library(knitr)
```

*#Cargamos el dataset Correspondiente de casa.txt en una variable*  
`Ventas<-read.table("Archivos/Ventas_Provincia.txt", header=T, sep=',', dec='.', stringsAsFactors=F)`

```
attach(Ventas)
```

```
names(Ventas)
```

```
## [1] "Year_Mes"      "Provincia"     "Total_Ventas"
```

### Apartado a)

*#Creamos el factor determinante de la tabla*

```
provincias<-(Provincia)
```

```
fprovincia<-as.factor(provincias)
```

```
levels(fprovincia)
```

```
## [1] "Albacete" "Alicante" "Almeria" "Asturias" "Avila"
```

```
## [6] "Badajoz" "Barcelona" "Caceres" "Gerona" "Huelva"
```

```
## [11] "Jaen" "Madrid" "Oviedo"
```

*#Tabla con las medias anuales*

```
t_medias<-aggregate(Total_Ventas~Provincia,Ventas,mean)
```

```
kable(t_medias)
```

Provincia	Total_Ventas
Albacete	60684.380



Alicante	9005.855
Almeria	37549.568
Asturias	35828.517
Avila	17322.423
Badajoz	36697.344
Barcelona	34684.695
Caceres	30688.796
Gerona	13441.506
Huelva	2449.362
Jaen	15866.704
Madrid	14132.735
Oviedo	3275.552

*#Tabla con las sumas anuales*

```
t_suma<-aggregate(Total_Ventas~Provincia,Ventas,sum)
kable(t_suma)
```

Provincia	Total_Ventas
Albacete	728212.56
Alicante	99064.40
Almeria	450594.81
Asturias	429942.21
Avila	207869.08
Badajoz	440368.13
Barcelona	416216.34
Caceres	368265.55
Gerona	161298.07
Huelva	29392.34
Jaen	190400.45
Madrid	169592.82
Oviedo	26204.42

```
detach()
```

Apartado b)

*#Seleccionamos la provincia que vende mas y la que menos vende*

```
attach(t_suma)
t_suma[which.max(t_suma[,2]),]
```

```
## Provincia Total_Ventas
## 1 Albacete 728212.6
```

```
#max<-Provincia[which.max(Total_Ventas)]
```

```
#max
```

```
t_suma[which.min(t_suma[,2]),]
```

```
##      Provincia Total_Ventas
```

```
## 13      Oviedo      26204.42
```

```
#min<-Provincia[which.min(Total_Ventas)]
```

```
#min
```

```
detach()
```

Apartado c)

Con `sqldf()` creamos una nueva tabla filtrada en base a los valores que solicita el enunciado.

*#Seleccionamos de la tabla una nuevo dataframe con SQL según los requerimientos pedidos*

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
Ventas2dosem<-sqldf("SELECT * FROM Ventas where Provincia='Barcelona' OR Provincia='Caceres' OR Provincia='Madrid'")
```

```
Ventas2dosem<-sqldf("SELECT * FROM Ventas2dosem where Year_Mes > '2012-06-01'")
```

```
kable(Ventas2dosem)
```

Year_Mes	Provincia	Total_Ventas
2012-07-01	Barcelona	22484.075
2012-07-01	Caceres	1719.600
2012-07-01	Madrid	8576.504
2012-08-01	Barcelona	65244.190
2012-08-01	Caceres	19879.714
2012-08-01	Madrid	12550.331
2012-09-01	Barcelona	15946.362
2012-09-01	Caceres	26386.495
2012-09-01	Madrid	21899.312
2012-10-01	Barcelona	38593.389
2012-10-01	Caceres	40658.198
2012-10-01	Madrid	20456.170
2012-11-01	Barcelona	34123.558
2012-11-01	Caceres	39202.173
2012-11-01	Madrid	49213.368

2012-12-01	Barcelona	34753.981
2012-12-01	Caceres	16284.664
2012-12-01	Madrid	888.380

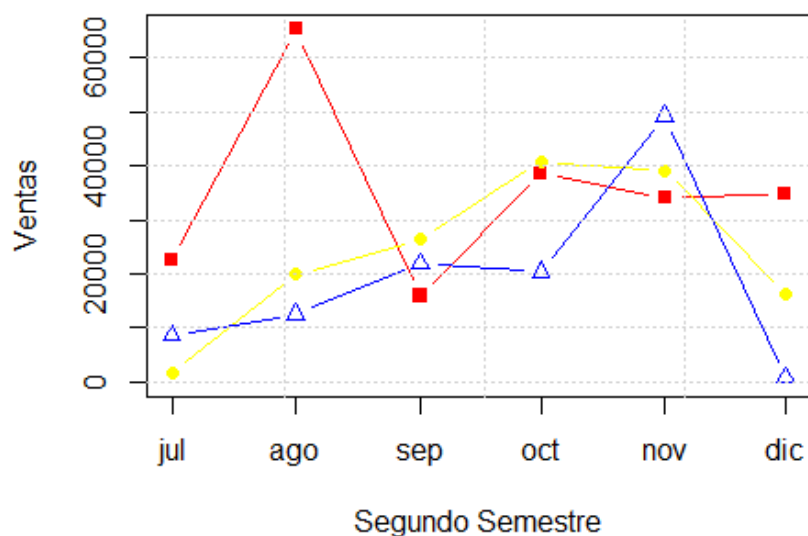
```
detach()
```

#### Apartado d)

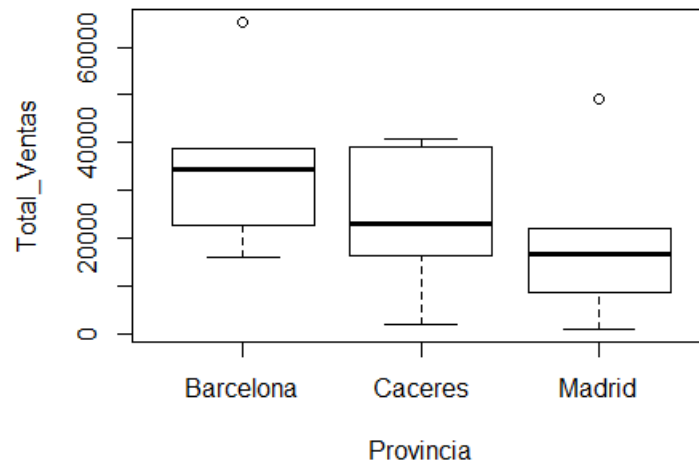
Generamos el plot con `points()` y líneas para mostrar como es el comportamiento de las ventas en cada ciudad durante los últimos 6 meses agrupada por ciudad.

A continuación generamos un boxplot equivalente.

```
attach(Ventas2dosem)
mesdata<-as.Date(Year_Mes)
meses<-unique(mesdata)
#Puntos Barcelona
plot(meses,Total_Ventas[Provincia=="Barcelona"],
     ylim=c(0,max(Total_Ventas)),type = "b",
     col = "red", xlab = "Segundo Semestre",
     ylab = "Ventas", pch=15)
#Puntos Caceres
points(meses,Total_Ventas[Provincia=="Caceres"],
       ylim=c(0,max(Total_Ventas)),type = "b",
       col = "yellow", pch=19)
#Puntos Madrid
points(meses,Total_Ventas[Provincia=="Madrid"],
       ylim=c(0,max(Total_Ventas)),type = "b",
       col = "blue", pch=2)
grid()
```



```
## Boxpot del Ventas2osem
boxplot(Total_Ventas~Provincia)
```

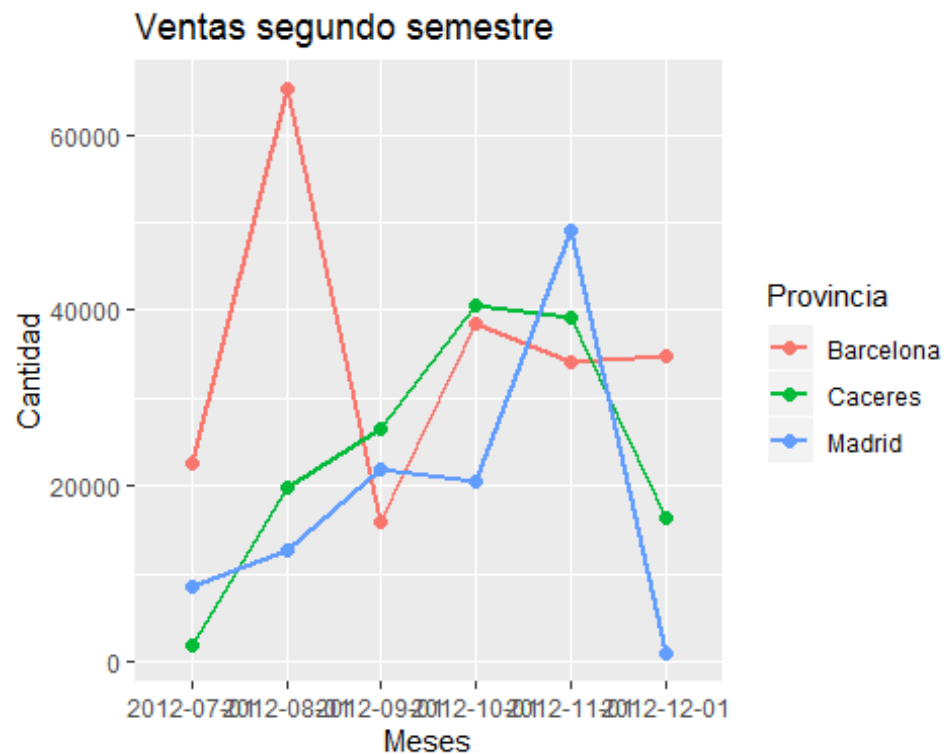


#### Apartado e)

Con ggplot, repetimos el apartado anterior, creando una gráfica lineal agrupada por ciudades empleando gramática de gráficos.

```
#Creamos la variable g
library(ggplot2)
g<-ggplot(data=Ventas2dosem, aes(x=Year_Mes,
                                y=Total_Ventas,
                                group=Provincia,
                                colour=Provincia))

g2<-g + geom_point(size=2)
g2<-g2 + geom_line(size=1)
g2<-g2 + xlab("Meses")
g2<-g2 + ylab("Cantidad")
g2<-g2 + ggtitle("Ventas segundo semestre")
g2
```



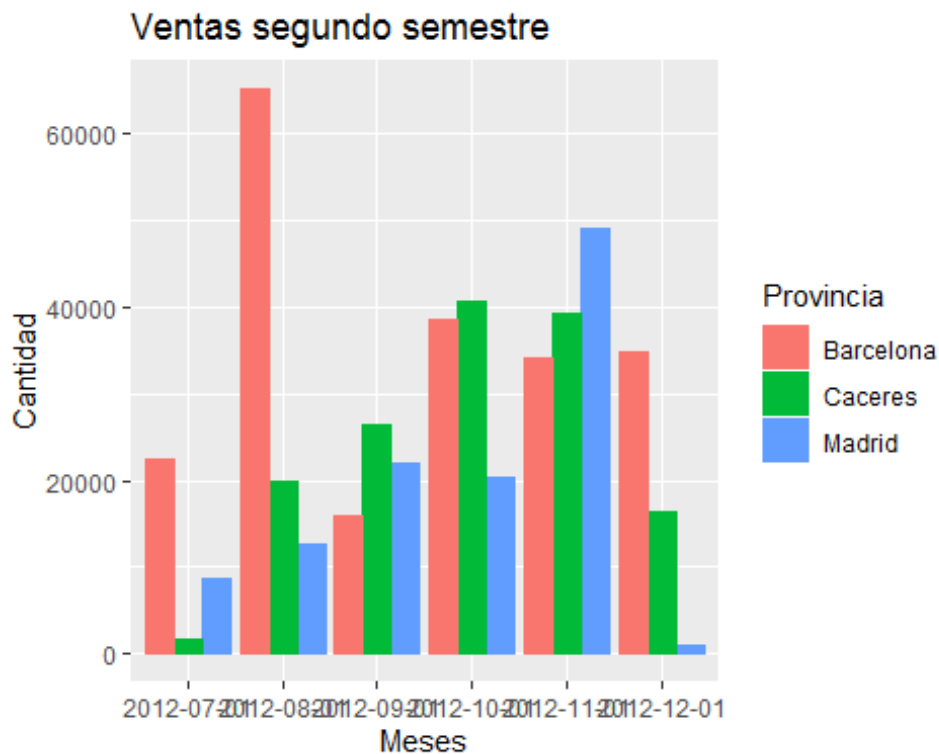
#### Apartado f)

Modificamos la gráfica del apartado anterior a una generada por barras. Lo hacemos en una nueva variable para no perder el progreso anterior.

```
g3<-ggplot(data=Ventas2dosem, aes(x=Year_Mes,
                                   y=Total_Ventas,
                                   group=Provincia,
                                   colour=Provincia))

g3<-g3 + geom_bar(stat="identity",
                  aes(fill=Provincia),
                  position="dodge")

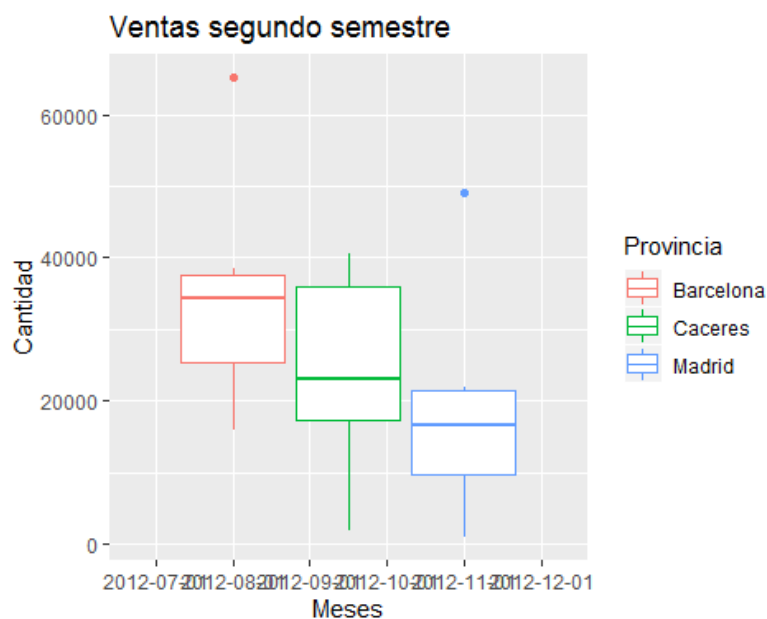
g3<-g3 + xlab("Meses")
g3<-g3 + ylab("Cantidad")
g3<-g3 + ggtitle("Ventas segundo semestre")
g3
```



Realizamos los mismo pero con una gráfica tipo boxplot y otra tipo violin.

```
g4<-ggplot(data=Ventas2dosem, aes(x=Year_Mes,
                                   y=Total_Ventas,
                                   group=Provincia,
                                   colour=Provincia))

g4<-g4 + geom_boxplot()
g4<-g4 + xlab("Meses")
g4<-g4 + ylab("Cantidad")
g4<-g4 + ggtitle("Ventas segundo semestre")
g4
```

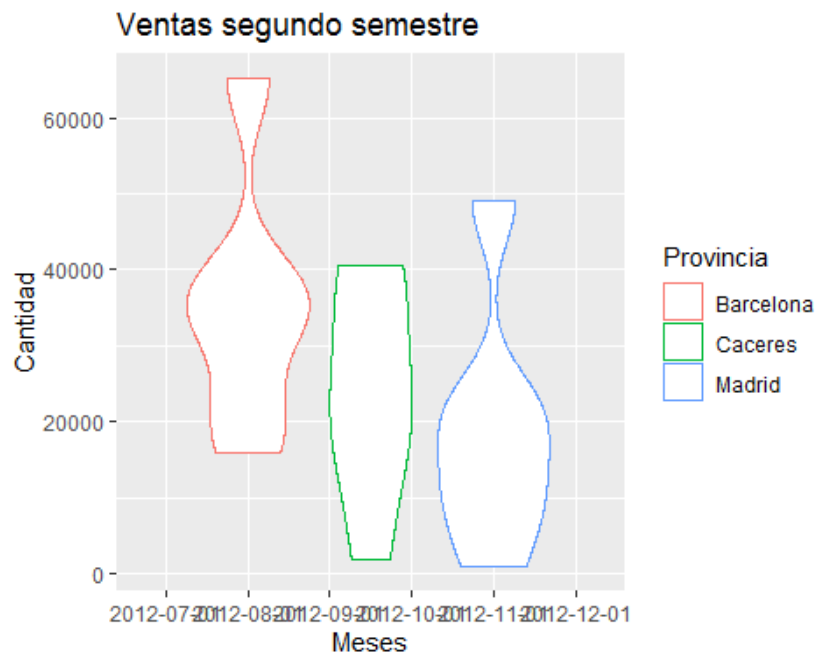


```

g5<-ggplot(data=Ventas2dosem, aes(x=Year_Mes,
                                   y=Total_Ventas,
                                   group=Provincia,
                                   colour=Provincia))

g5<-g5 + geom_violin()
g5<-g5 + xlab("Meses")
g5<-g5 + ylab("Cantidad")
g5<-g5 + ggtitle("Ventas segundo semestre")
g5

```



## Ejercicio 8

**Ejercicio 8 (Opcional):** Diseñar con los gráficos de base de **R** una función que rellene el área entre una curva dada por un vector y el eje de ordenadas, desde una posición inicial (*xlimite1*) a otra final (*xlimite2*). Se deberá incluir:

- Parámetros de entrada: el vector de ordenadas **x** y de abscisas **y**, las posiciones inicial y final de las ordenadas (*xlimite1*, *xlimite2*) y el color de relleno y el tipo. Opcionalmente puede incluirse una función como entrada que establezca la relación  $y=f(x)$ .
- Comprobación de los parámetros de entrada que garanticen la estabilidad de la función.
- Introducir mejoras en la llamada de la función con la posibilidad de incluir algún texto adicional en alguno de los ejes o en la propia gráfica.
- Plantear una función alternativa utilizando **ggplot2**).
- Introducir mejoras en la estética, el tema y la escala en la propia llamada de la función.
- Explicar el uso de ambas funciones en al menos tres casos de ejemplo

### Disposiciones preliminares

Iniciamos el espacio de trabajo en Nuestro directorio de proyecto

```
setwd("H:/ME/Practicas/Entrega_1/Ejercicios R")
library(knitr)
library(carData)
library(ggplot2)
```

### Apartados a) y b)

Generamos una función muy simple, se trata de una función parabólica adscrita al valor de la x. Recibe 5 parámetros, dos para la función y 3 para rellenar el polígono que se crea sobre ésta. Dispone de condicionantes para evitar problemas en la representación y errores de escala

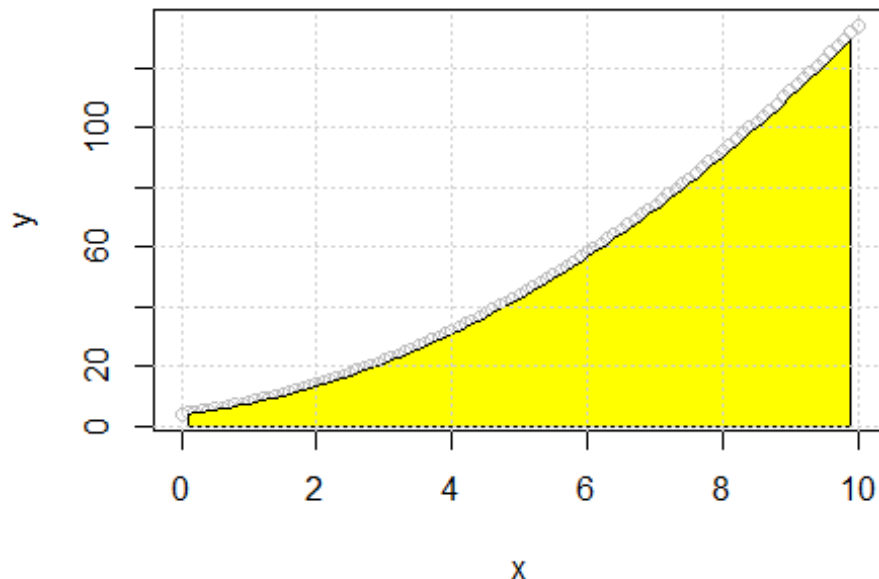
```
xaxis <- seq(from=0, to=10, by=0.1)
yaxis<- (xaxis)^2 + 3*xaxis + 4
cX<-xaxis
cY<-yaxis
xinf<-0.1
xsup<-9.9
incr<-0.1
plot_scale <- function(x,y,x1,x2,inc){
  if(length(x) != length(y)) stop("La dimensión en los vectores
no es la misma")
  if(!is.numeric(x1)) stop("ERROR: Entrada no numerica")
  if(!is.numeric(x2)) stop("ERROR:Entrada no numérica")
  plot(x,y, col = 'grey')
  n1<-as.integer(x1/incr)
```



```

n2<-as.integer(x2/inc)
#Generamos las coordenadas de las x para el poligono.
curva_x<- seq(x1,x2,inc)
traza_x<-c(x1, curva_x,x2)
#Generamos las coordenadas de las y.
curva_y <- y[n1:(n1 + length(curva_x)-1)]
traza_y<-c(0,curva_y, 0)
polygon(traza_x, traza_y, col = '#ffff00')
}
plot_scale(xaxis,yaxis,xinf,xsup,incr)
grid()

```



#### Apartado c)

Añadimos condicionantes y elementos como texto, título y demás al gráfico original.

```

# FUNCION MEJORADA
plot_scale <- function(x,y,x1,x2,inc){
  if(length(x) != length(y)) stop("La dimensión en los vectores
no es la misma")
  if(!is.numeric(x1)) stop("ERROR: Entrada no numerica")
  if(!is.numeric(x2)) stop("ERROR:Entrada no numérica")
  plot(x,y, main = 'Gráfica de ejemplo', xlab = 'X axis', ylab =
'Y axis', col = 'grey')
  n1<-as.integer(x1/inc)
  n2<-as.integer(x2/inc)
  curva_x<- seq(x1,x2,inc)
  curva_y <- y[n1:(n1 + length(curva_x)-1)]
  traza_x<-c(x1, curva_x,x2)

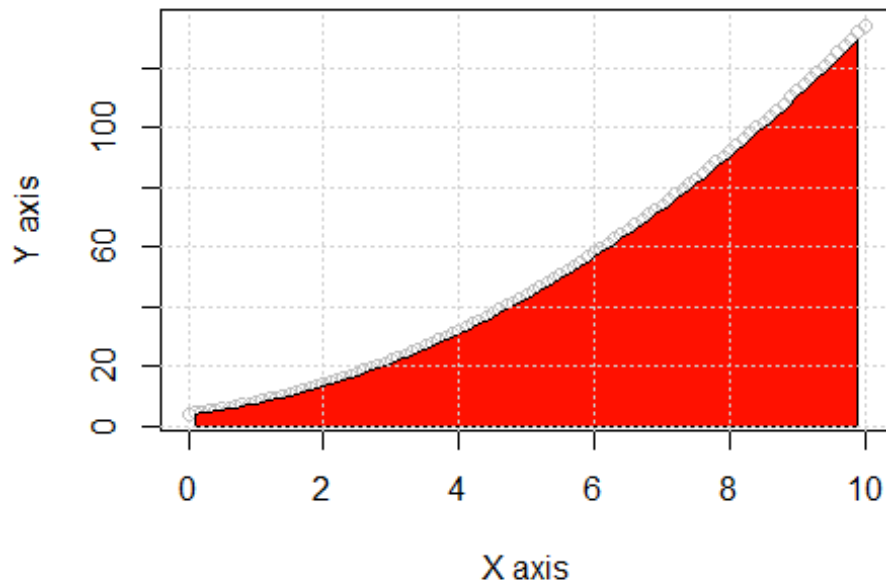
```

```

traza_y<-c(0,curva_y, 0)
polygon(traza_x, traza_y, col = '#ff1100')
}
plot_scale(xaxis,yaxis,xinf,xsup,incr)
grid()

```

### Gráfica de ejemplo



#### Apartados d) y e)

Realizamos lo anterior pero con ggplot. Para el área en cuestión, generamos un dataframe producto del original, pero filtrado a los valores que vamos a pintar.

```

xaxis<-seq(from=0, to=10, by=0.5)
yaxis<-(xaxis)^2 + 3*xaxis + 4
funcion_parab<-data.frame(xaxis,yaxis)
funcion_parab_ajustada<-funcion_parab[funcion_parab$xaxis>2,]
funcion_parab_ajustada<-funcion_parab_ajustada[funcion_parab_ajustada$xaxis<4,]

```

```
attach(funcion_parab)
```

```

## The following objects are masked _by_ .GlobalEnv:
##
##      xaxis, yaxis

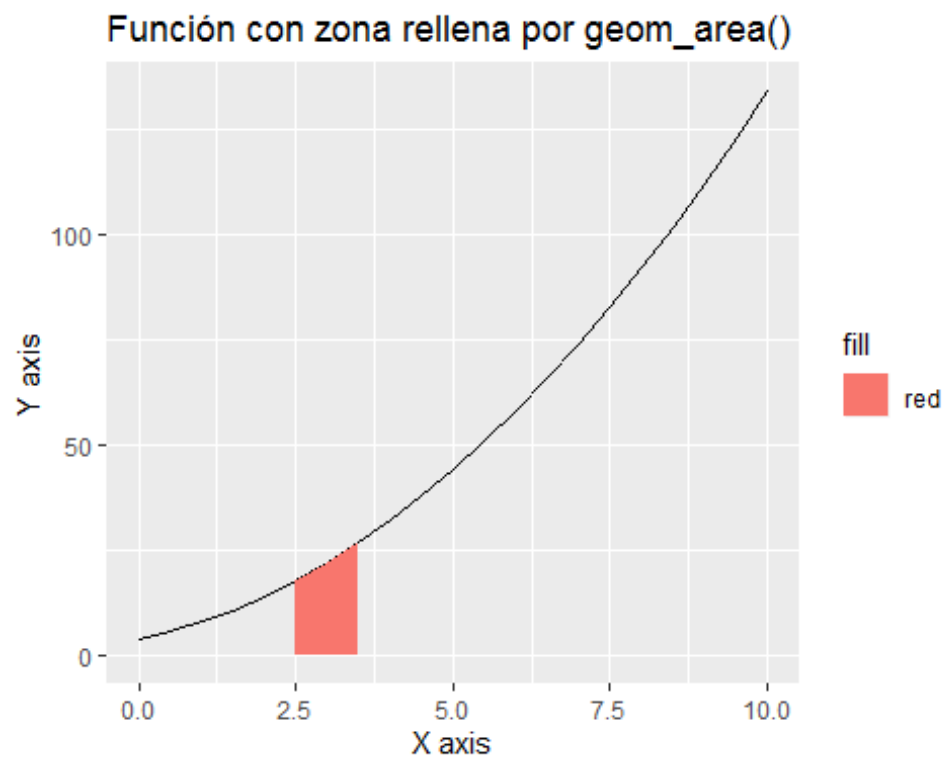
```

```

g<-ggplot(data = funcion_parab, mapping = aes(x=xaxis, y=yaxis))
g2<-g + geom_line()
g2<-g2 + xlab("X axis")
g2<-g2 + ylab("Y axis")
g2<-g2 + ggtitle("Función con zona rellena por geom_area()")
g2<-g2 + geom_area(data=funcion_parab_ajustada, mapping = aes(fi

```

```
l1 = 'red'))  
g2
```



## Ejercicio 9

**Ejercicio 9 (Opcional):** El fichero *"Datos\_hsb.txt"* contiene datos de resultados de aprendizaje de un estudio longitudinal de estudiantes USA no universitarios (High School and Beyond, 1980, National Center for Education Statistics) relativos a sus progresos en varias materias (lectura, escritura, matemáticas, sociales y ciencias) y clasificados por categorías (sexo, raza, estatus socioeconómico, tipo de centro y de programa). Se desea realizar un análisis de estos datos utilizando **ggplot2()** y combinado con diferentes estadísticos resumen, para ello se pide:

- Calcular las medias de progreso por las distintas categorías y razonar posibles conclusiones.
- Evaluar la influencia del centro en los resultados de aprendizaje de la época
- Estudiar posibles relaciones entre la capacidad de lectura y el aprendizaje de otras disciplinas.
- Visualizar si el sexo o la raza de origen en 1980 tenían influencias significativas en el nivel de aprendizaje.

#PRELIMINARES

```
setwd("H:/ME/Practicas/Entrega_1/Ejercicios R")
library(ggplot2)
library(knitr)
datosestudiantes<-read.table("Archivos/Datos_hsb.txt", sep = ",",
,dec = ".",header = T)
```

Apartado a)

Montamos el dataframe principal, sobre el cual generaremos los plots del ejercicio.

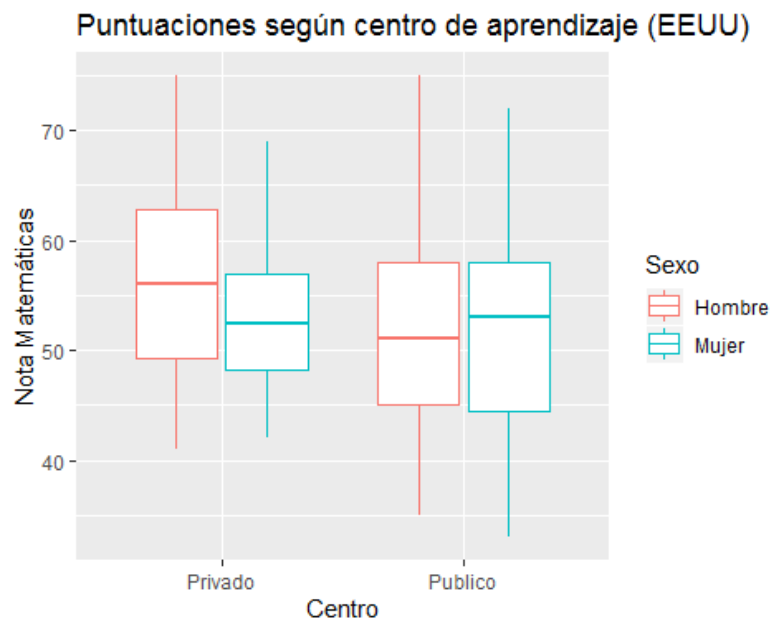
```
media_mt<-aggregate(cbind(P_Lectura,P_Matematicas,P_Ciencias,P_E
scritura,P_Sociales)~Raza+Sexo+Estatus_Socieconomico,data = dato
sestudiantes,mean)
kable(media_mt)
```

Raza	Sexo	Estatus	P_Lectura	P_Matematicas	P_Ciencias	P_Escritura	P_Sociales
Blanca	Hombre	alto	56.76000	55.88000	56.36000	54.44000	58.80000
Hispana	Hombre	alto	48.66667	51.66667	43.00000	44.00000	53.00000
Negra	Hombre	alto	50.00000	39.00000	49.00000	40.00000	47.00000
Asiática	Mujer	alto	50.33333	56.66667	53.66667	61.66667	52.66667
Blanca	Mujer	alto	57.82609	57.86957	57.43478	59.17391	56.91304
Hispana	Mujer	alto	73.00000	57.00000	55.00000	61.00000	66.00000
Negra	Mujer	alto	54.00000	54.50000	46.00000	51.50000	52.50000
Asiática	Hombre	bajo	52.00000	49.00000	55.00000	44.00000	41.00000
Blanca	Hombre	bajo	49.33333	48.55556	51.88889	46.77778	43.22222
Hispana	Hombre	bajo	52.33333	45.00000	48.00000	45.33333	41.00000
Negra	Hombre	bajo	43.50000	46.50000	40.50000	49.00000	48.50000
Asiática	Mujer	bajo	61.00000	68.00000	56.00000	65.00000	63.50000
Blanca	Mujer	bajo	50.20000	52.26667	51.86667	56.00000	49.33333
Hispana	Mujer	bajo	42.00000	45.33333	42.66667	46.50000	47.16667
Negra	Mujer	bajo	44.66667	45.00000	38.77778	47.88889	47.11111
Asiática	Hombre	medio	52.50000	63.50000	52.00000	61.50000	51.00000
Blanca	Hombre	medio	54.23529	54.44118	55.58824	50.17647	52.17647
Hispana	Hombre	medio	44.57143	50.00000	45.57143	44.14286	44.57143
Negra	Hombre	medio	47.75000	46.25000	49.25000	47.75000	49.75000
Asiática	Mujer	medio	47.00000	49.33333	44.66667	52.00000	44.33333
Blanca	Mujer	medio	52.02564	51.94872	51.12821	55.10256	53.89744
Hispana	Mujer	medio	45.00000	42.25000	46.50000	49.50000	51.00000
Negra	Mujer	medio	49.00000	52.00000	44.00000	50.50000	58.50000

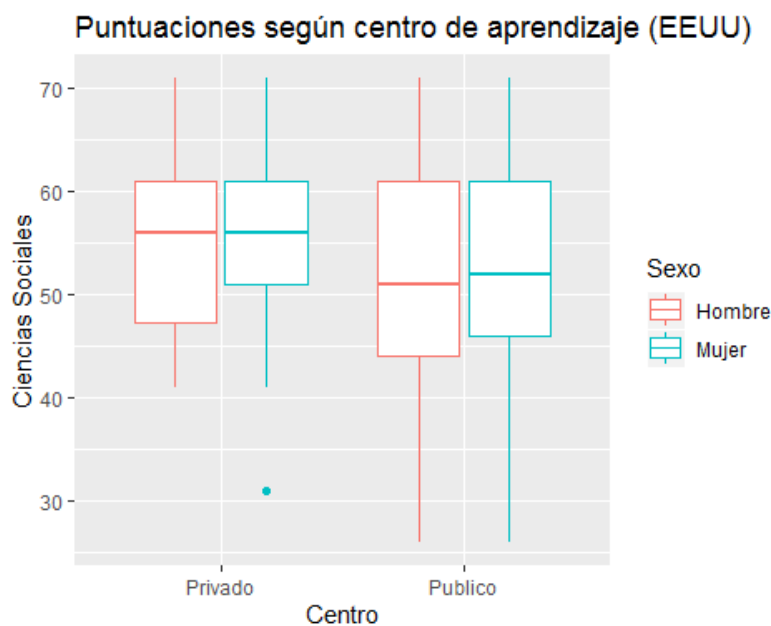
Apartados b), c) y d)

Iremos generando cada uno de los gráficos tipo caja necesarios, como se ha explicado en ejercicios anteriores, empleando *ggplot()*. Se mostrará el código adscrito a cada uno de ellos.

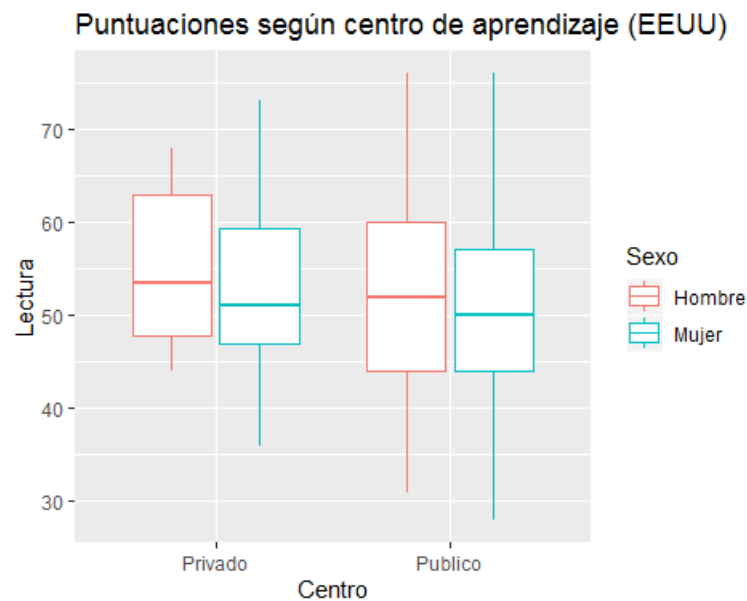
```
#Generamos cada uno de nuestros plots
g0<-ggplot(data = datosestudiantes, aes(x=Tipo_Escuela,y=P_Matem
aticas))
g1<- g0+geom_boxplot()+aes(colour=Sexo)+xlab("Centro")+ylab("Not
a Matemáticas")+ggtitle("Puntuaciones según centro de aprendizaj
e (EEUU)")
g1
```



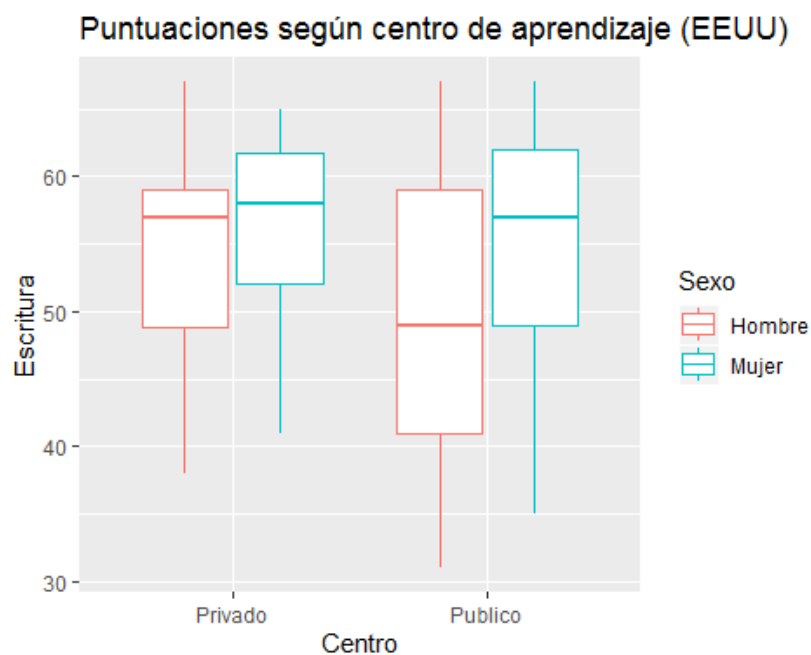
```
g2<-ggplot(data = datosestudiantes, aes(x=Tipo_Escuela,y=P_Sociales))
g3<-g2+geom_boxplot()+aes(colour=Sexo)+xlab("Centro")+ylab("Ciencias Sociales")+ggtitle("Puntuaciones según centro de aprendizaje (EEUU)")
g3
```



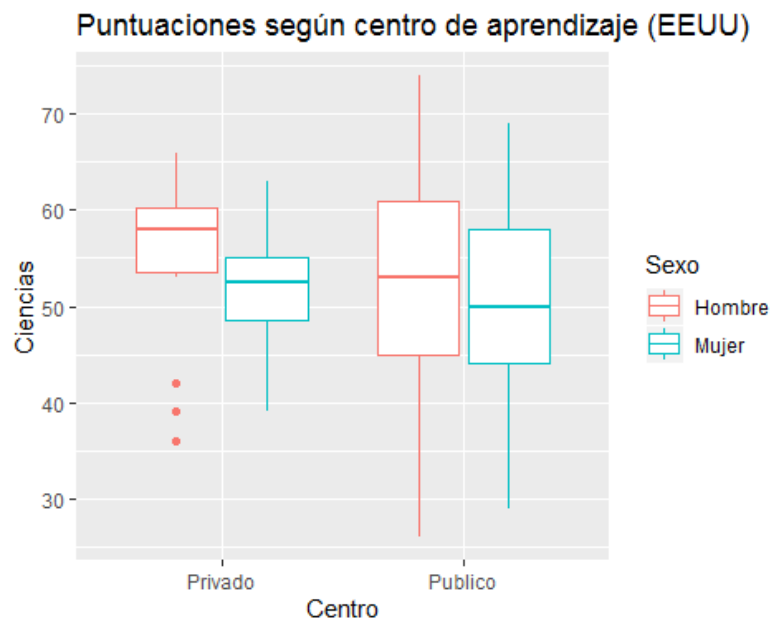
```
g4<-ggplot(data = datosestudiantes, aes(x=Tipo_Escuela,y=P_Lectura))
g5<-g4+geom_boxplot()+aes(colour=Sexo)+xlab("Centro")+ylab("Lectura")+ggtitle("Puntuaciones según centro de aprendizaje (EEUU)")
g5
```



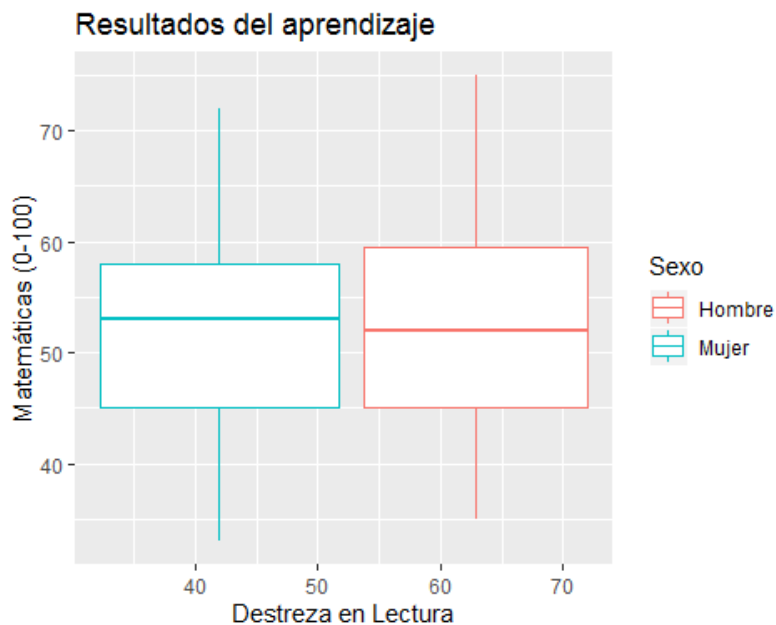
```
g6<-ggplot(data = datosestudiantes, aes(x=Tipo_Escuela,y=P_Escri
tura))
g7<-g6+geom_boxplot()+aes(colour=Sexo)+xlab("Centro")+ylab("Escr
itura")+ggtitle("Puntuaciones según centro de aprendizaje (EEUU)
")
g7
```



```
g8<-ggplot(data = datosestudiantes, aes(x=Tipo_Escuela,y=P_Cienc
ias))
g9<-g8+geom_boxplot()+aes(colour=Sexo)+xlab("Centro")+ylab("Cien
cias")+ggtitle("Puntuaciones según centro de aprendizaje (EEUU)"
)
g9
```

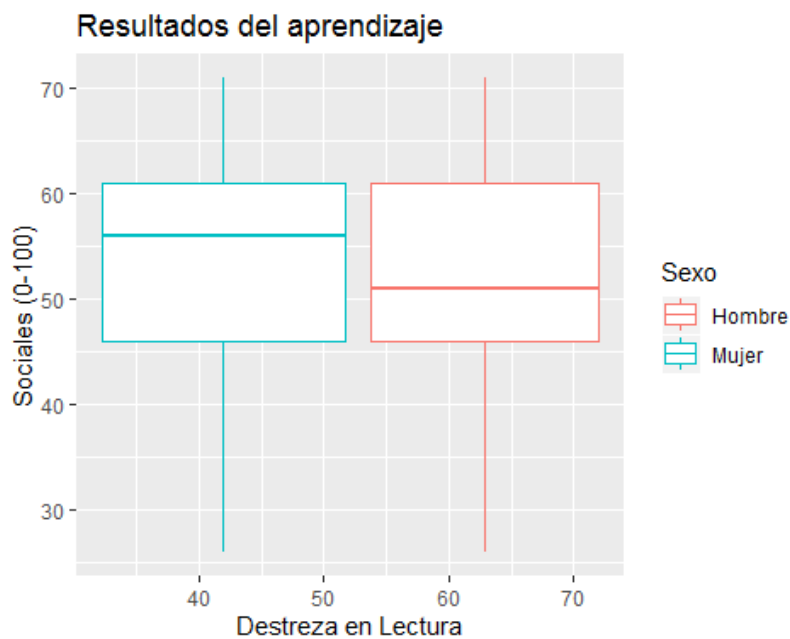


```
g10<-ggplot(data = datosestudiantes, aes(x=P_Lectura, y=P_Matematicas))
g11<- g10+geom_boxplot()+aes(colour=Sexo)+xlab("Destreza en Lectura")+ylab("Matemáticas (0-100)")
g11+ggtitle("Resultados del aprendizaje")
```

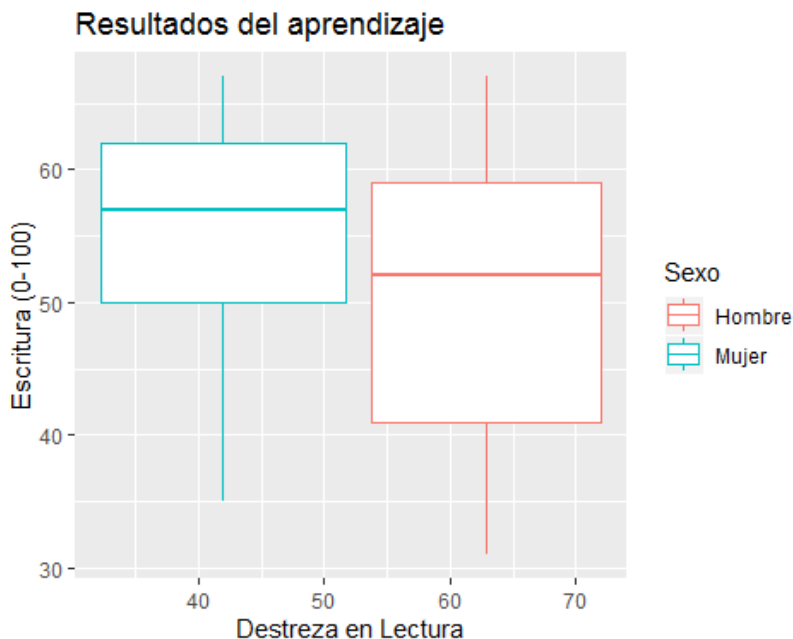


```
g12<-ggplot(data = datosestudiantes, aes(x=P_Lectura,y=P_Sociales))
g13<-g12+geom_boxplot()+aes(colour=Sexo)+xlab("Destreza en Lectura")+ylab("Sociales (0-100)")
g13+ggtitle("Resultados del aprendizaje")
```

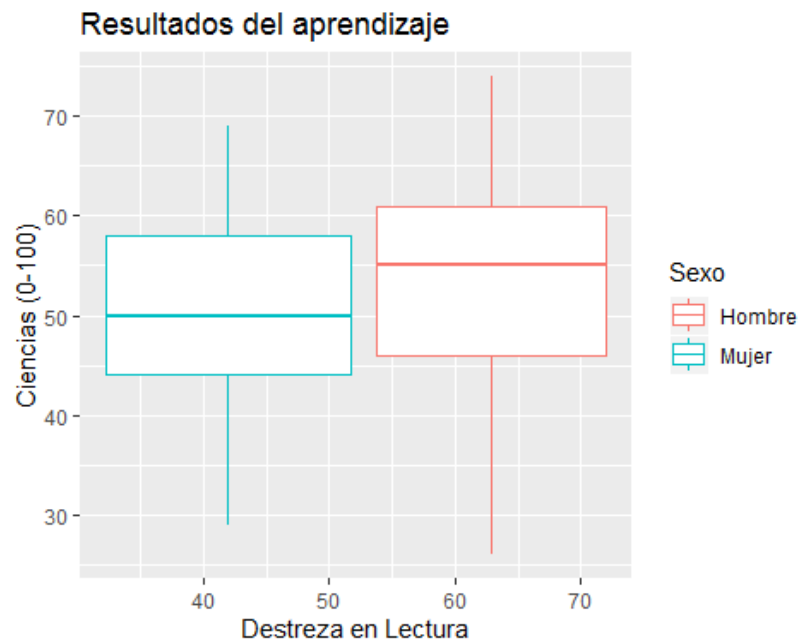




```
g14<-ggplot(data = datosestudiantes, aes(x=P_Lectura,y=P_Escritura))
g15<-g14+geom_boxplot()+aes(colour=Sexo)+xlab("Destreza en Lectura")+ylab("Escritura (0-100)")
g15+ggtitle("Resultados del aprendizaje")
```

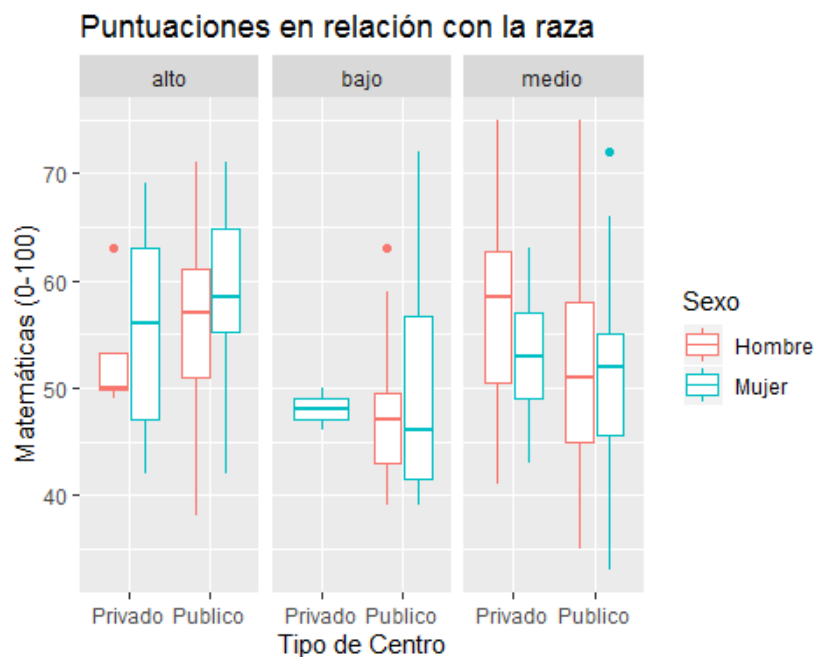


```
g16<-ggplot(data = datosestudiantes, aes(x=P_Lectura,y=P_Ciencias))
g17<-g16+geom_boxplot()+aes(colour=Sexo)+xlab("Destreza en Lectura")+ylab("Ciencias (0-100)")
g17+ggtitle("Resultados del aprendizaje")
```

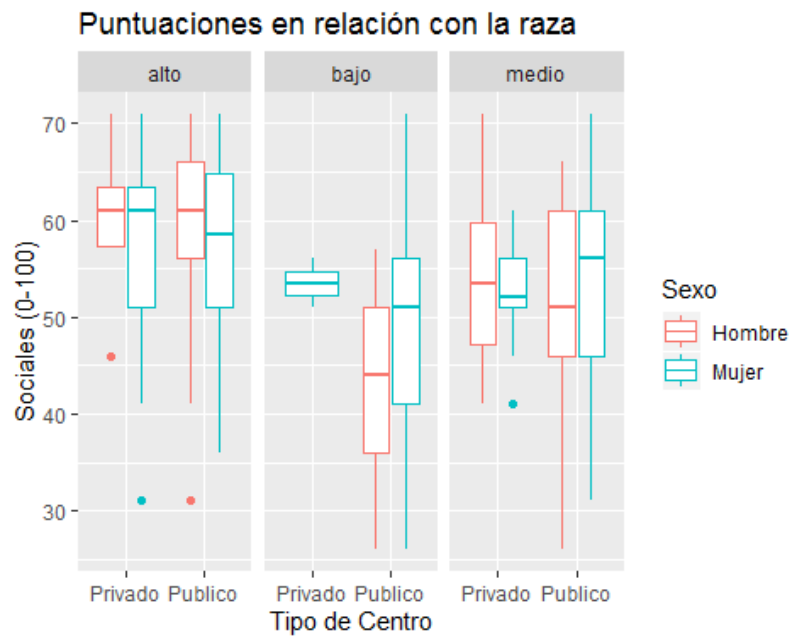


4. Visualizar si el sexo o la raza de origen en 1980 tenían influencias significativas en el nivel de aprendizaje.

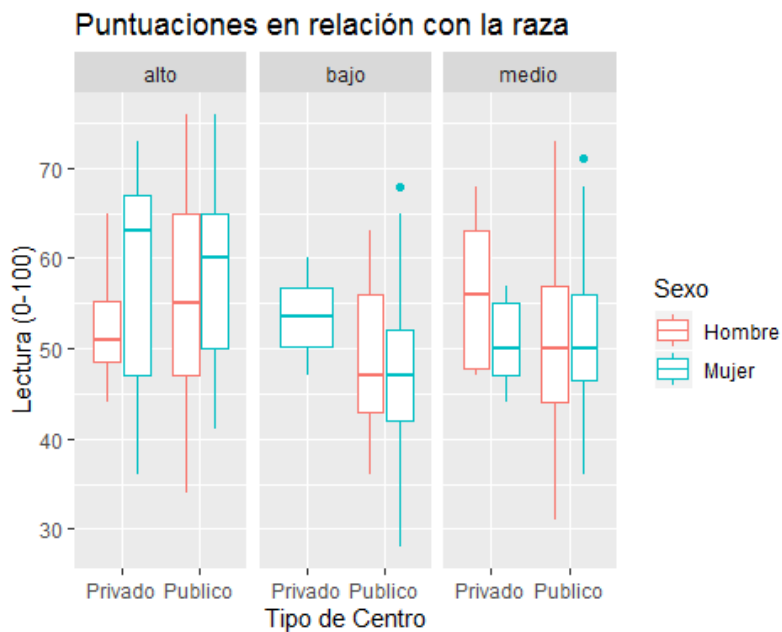
```
g18<- g0+geom_boxplot()+xlab("Tipo de Centro")+ylab("Matemáticas (0-100)")
+ggtitle("Puntuaciones en relación con la raza")
g18 <- g18+aes(colour=Sexo)+facet_wrap(~Estatus_Socioeconomico)
g18
```



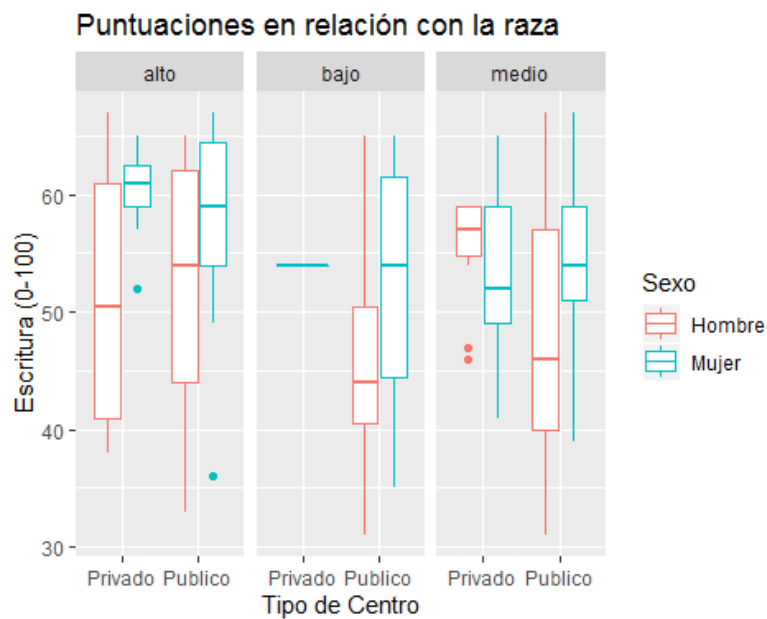
```
g19<- g2+geom_boxplot()+xlab("Tipo de Centro")+ylab("Sociales (0-100)")
+ggtitle("Puntuaciones en relación con la raza")
g19<-g19+aes(colour=Sexo)+facet_wrap(~Estatus_Socioeconomico)
g19
```



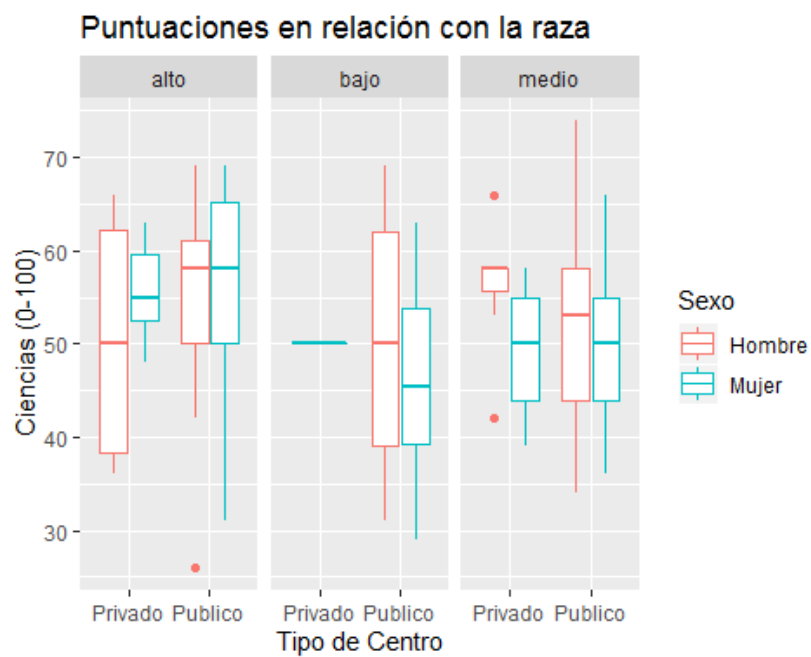
```
g20<-g4+geom_boxplot()+xlab("Tipo de Centro")+ylab("Lectura (0-100)")
ggtitle("Puntuaciones en relación con la raza")
g20<-g20+aes(colour=Sexo)+facet_wrap(~Estatus_Socioeconomico)
g20
```



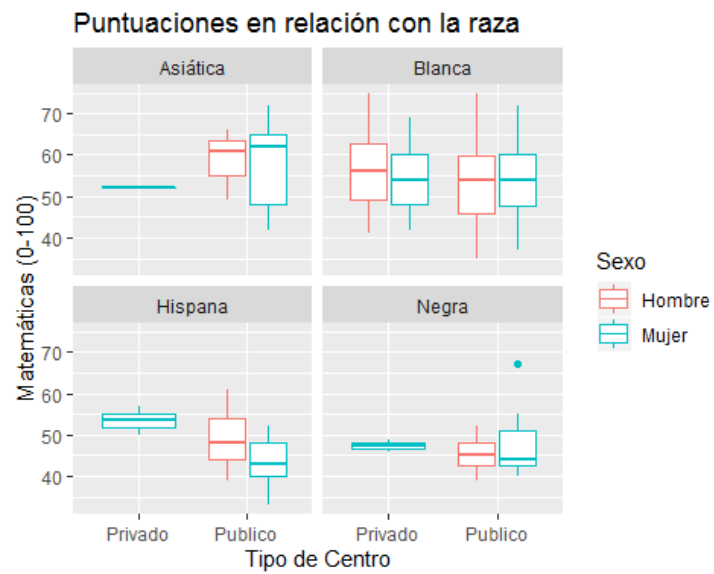
```
g21<-g6+geom_boxplot()+xlab("Tipo de Centro")+ylab("Escritura (0-100)")
ggtitle("Puntuaciones en relación con la raza")
g21<-g21+aes(colour=Sexo)+facet_wrap(~Estatus_Socioeconomico)
g21
```



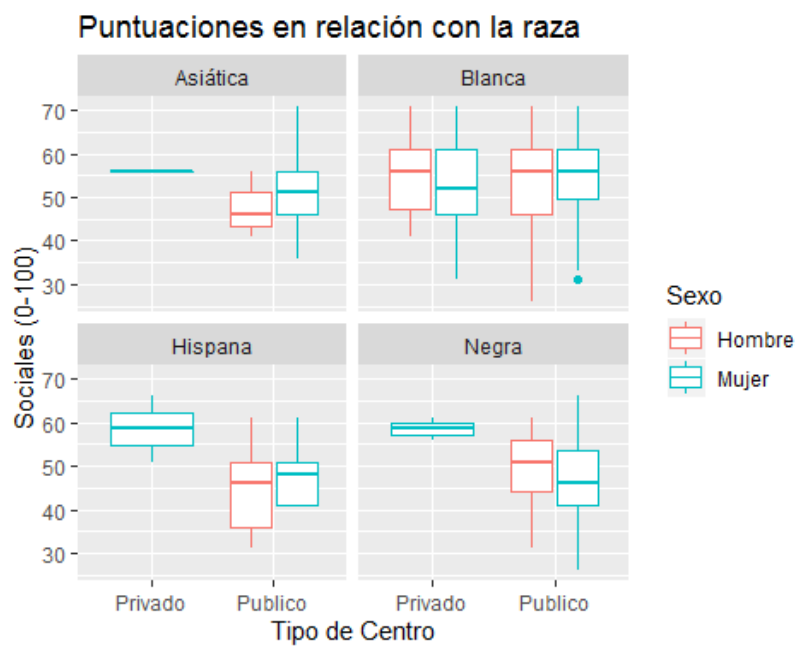
```
g22<-g8+geom_boxplot()+xlab("Tipo de Centro")+ylab("Ciencias (0-100)")
ggtitle("Puntuaciones en relación con la raza")
g22<-g22+aes(colour=Sexo)+facet_wrap(~Estatus_Socioeconomico)
g22
```



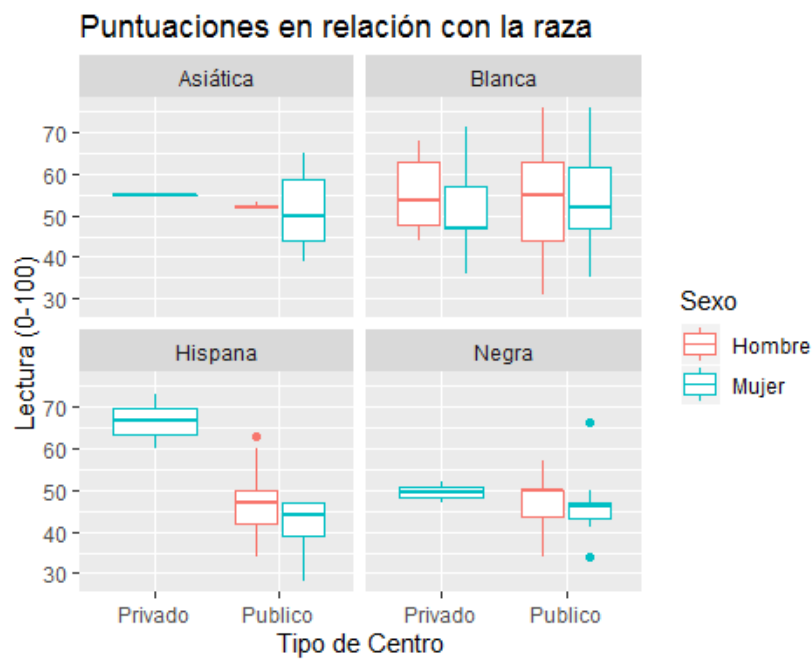
```
g30<-g18+facet_wrap(~Raza)
g30
```



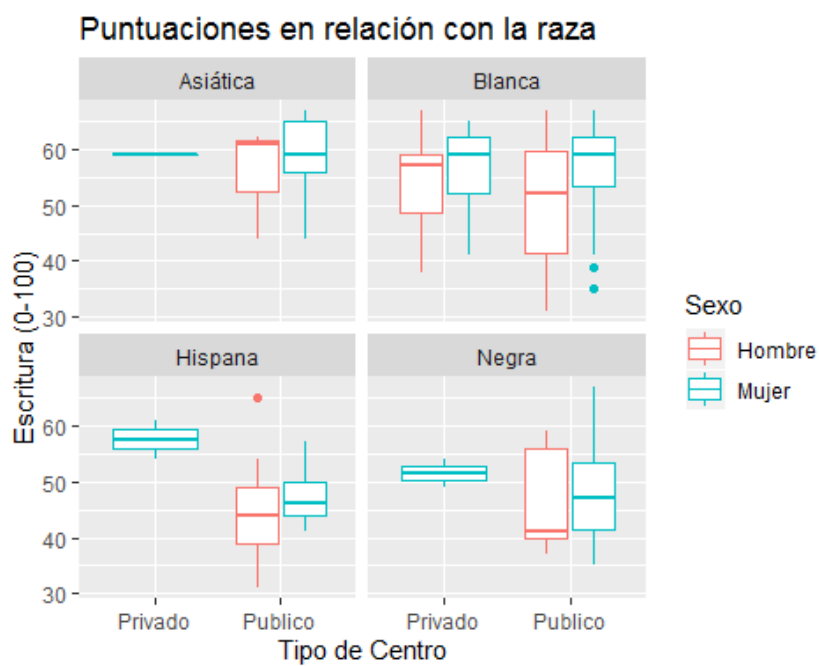
```
g31<-g19+facet_wrap(~Raza)
g31
```



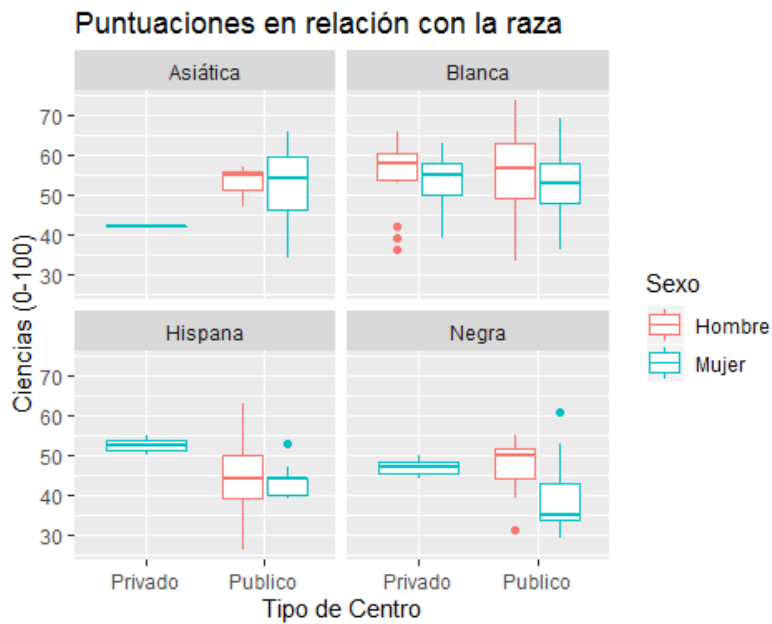
```
g32<-g20+facet_wrap(~Raza)
g32
```



```
g33<-g21+facet_wrap(~Raza)
g33
```



```
g34<-g22+facet_wrap(~Raza)
g34
```



### Conclusiones finales del ejercicio

Como se ha podido comprobar mediante el estudio anterior, no se experimentan diferencias significativas entre sexo y raza a la hora de computar las diferentes resultantes.

Si bien es determinante el factor socioeconómico, pues se verifica que aquellos cuyo poder adquisitivo es mayor, obtienen mejores resultados debido a que tienen acceso a una mejor educación.

Asimismo la situación social en 1980 en los Estados Unidos es completamente diferente a la actual y en algunos estados seguía presente la segregación racial (de facto), que podría haber impedido a gente de etnia no blanca el acceso a puestos de trabajo y educación superiores, repitiendo a su vez el ciclo de resultados en la generación sucesora

## CUESTIONES LECTURAS

### Cuestión L2.1

**Cuestión 1:** Los datos siguientes representan el peso en kilos de una muestra de 80 personas de la Escuela de Ingeniería Informática:

50	73	73	68	67	74	73	67	71	79
74	74	77	74	71	80	72	74	77	75
71	73	75	76	77	71	81	68	66	73
91	75	89	77	93	57	66	83	86	90
55	77	78	91	82	83	87	96	85	88
101	97	80	73	76	80	89	76	78	99
80	85	84	72	65	69	79	84	92	83
86	76	80	81	74	73	72	79	55	66

- Obtégase una distribución de datos en intervalos de amplitud del 5% de la distribución, construir una tabla de frecuencias absolutas y relativas y definir cada representante de la clase.
- Calcular la media muestral y la desviación estándar muestral.
- Encontrar la mediana, los cuartiles y el rango intercuartílico.
- Elaborar un histograma con los datos.
- Construir un diagrama de caja y mostrar los casos atípicos y otros elementos relevantes del mismo.
- Analizar los valores fuera de rango y la posible corrección de estos “outliers”.
- Construir un programa R para verificar los distintos apartados anteriores.

```
setwd("/")  
#Cuestion 1: Losdatos siguientes representan el peso en kilos de  
#una muestra  
#de 80 personas de La Escuela de Ingeniería Informática  
  
vector2<-c(50, 73, 73, 68, 67, 74, 73, 67, 71, 79, 74, 74, 77, 7  
4, 71, 80, 72,  
          74, 77, 75, 71, 73, 75, 76, 77, 71, 81, 68, 66, 73, 91  
, 75, 89, 77,  
          93, 57, 66, 83, 86, 90, 55, 77, 78, 91, 82, 83, 87, 96  
, 85, 88, 101,  
          97, 80, 73, 76, 80, 89, 76, 78, 99, 80, 85, 84, 72, 65  
, 69, 79, 84,  
          92, 83, 86, 76, 80, 81, 74, 73, 72, 79, 55, 66)
```

Apartado a)

```
vector<-read.table("/Archivos/PesosEstEII.txt")  
pesoDF<-data.frame(vector)
```



```
names(pesoDF)<-c("Intervalo", "FrecAbs", "FrecRel")
pesoDF
```

```
##      Intervalo FrecAbs FrecRel
## 1      50-55        3        3
## 2      55-60        1        4
## 3      60-65        1        5
## 4      65-70        8       13
## 5      70-75       23       36
## 6      75-80       19       55
## 7      80-85       10       65
## 8      85-90        7       72
## 9      90-95        4       76
## 10     95-100        3       79
## 11    100-105        1       80
```

Intervalo donde se experimenta la mayor frecuencia en la muestra el peso más común oscila entre 70 y 75 Kgs

```
attach(pesoDF)
Intervalo[(which.max(FrecAbs))]

## [1] 70-75
## 11 Levels: 100-105 50-55 55-60 60-65 65-70 70-75 75-80 80-85
... 95-100

detach(pesoDF)
```

Apartado b)

```
mean(vector2)
```

```
## [1] 77.3375
```

Aplicamos la función `var()` para descubrir la varianza.

```
varianzapeso<-var(vector2)
varianzapeso
```

```
## [1] 93.74541
```

Aplicamos raíz cuadrada y recuperamos la desviación estándar.

```
desv_est<-sqrt(varianzapeso)
desv_est
```

```
## [1] 9.682221
```

Comprobamos con la función `sd`.

```
desv_est2<-sd(vector2)
desv_est2

## [1] 9.682221
```

Apartado c)

```
median(vector2)
```

```
## [1] 76.5
```

*###Aplicamos quantile para verificar los cuantiles*

```
quantile(vector2)
```

```
##      0%      25%      50%      75%     100%
## 50.00  72.75  76.50  83.00 101.00
```

*###Aplicamos tambien con un ratio de 0.x para encontrar los deciles*

```
quants<-quantile(vector2, c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,
1))
quants
```

```
##   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
## 66.9  71.0  73.0  74.0  76.5  79.0  81.0  85.0  90.1 101.0
```

*#Aplicamos la fórmula para resolver el IQR y IDR*

```
rec_iq<-quantile(vector2,0.75)-quantile(vector2,0.25)
rec_iq
```

```
##      75%
## 10.25
```

```
rec_idc<-quantile(vector2,0.9)-quantile(vector2,0.1)
rec_idc
```

```
##      90%
## 23.2
```

*#Comprobamos con la fórmula IQR*

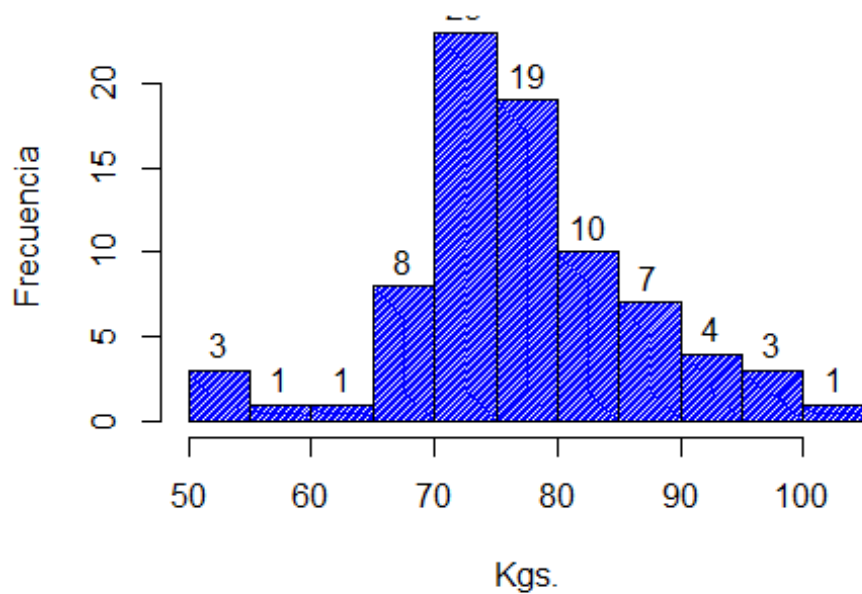
```
RIC<-IQR(vector2)
RIC
```

```
## [1] 10.25
```

Apartado d)

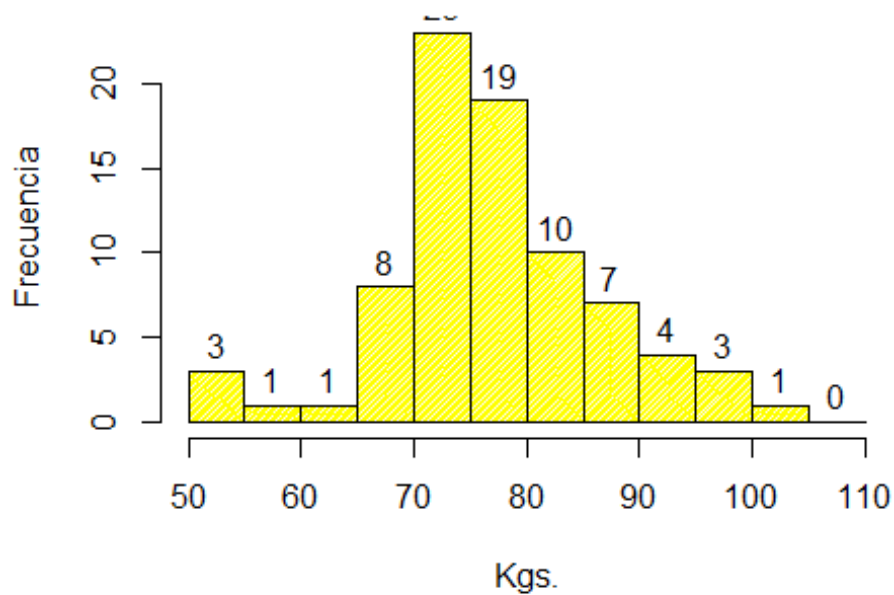
```
hist(vector2, main="Histograma sin ajustar outliers", xlab = "Kgs
.", ylab = "Frecuencia",
      col="blue", density= 100.0, border= "black", labels= TRUE)
```

**Histograma sin ajustar outliers**



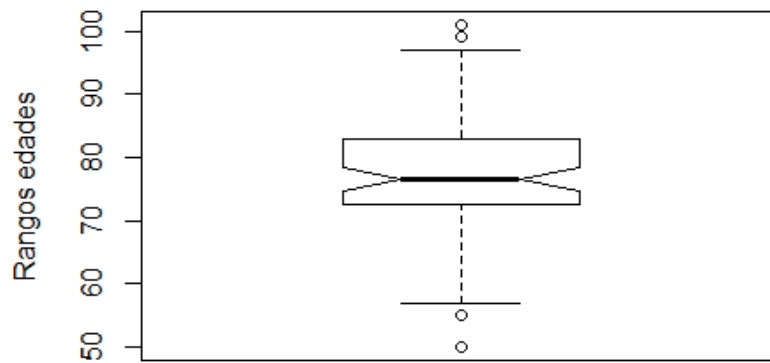
```
#Ajustamos los cortes al 5% respecto al máximo  
cortes<-c(50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110)  
hist(vector2, breaks=cortes, main="Histograma ajustando outliers"  
,xlab = "Kgs.", ylab = "Frecuencia",  
    col="yellow", density= 100.0, border= "black", labels= TRUE  
)
```

**Histograma ajustando outliers**



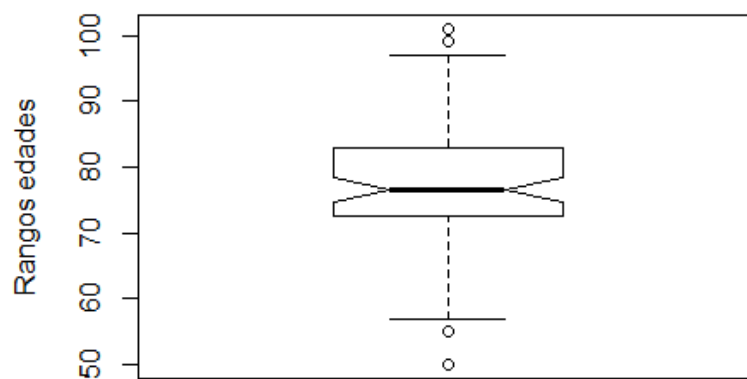
Apartado e)

```
boxplot(vector2, ylab = "Rangos edades", labels= TRUE, notch=TRUE, outliers=TRUE)
```



f) Analizar los valores fuera de rango y la posible corrección de estos “outliers”.

```
boxplot(vector2, ylab = "Rangos edades", labels= TRUE, notch=TRUE, outliers=FALSE)
```



## Cuestión L2.2

**Cuestión 2:** Los salarios mensuales en euros de los 230 empleados de una empresa del sector turístico ubicada en el Polígono de Arinaga se resume en la siguiente tabla:

Salarios (€/mes)	Empleados
600-900	5
900-1500	60
1500-2000	125
2000-2500	25
2500-3000	15

- Construir una tabla de datos de la distribución con los distintos tipos de frecuencias, realizar para ello un histograma (de frecuencias absolutas y relativas) y un diagrama de caja. Razonar sobre los distintos elementos de estos y explicar su contenido.
- Calcular el salario medio, mediano, el más frecuente por empleado y la variabilidad de los salarios. ¿Cuál se considera el salario más adecuado para representar la distribución?, justificar las respuestas.
- ¿Cuántos empleados cobran más de 1800 €/mes?
- ¿Cuál es el mínimo salario que cobra el 25% de los trabajadores mejor pagados?
- Calcular el tercer cuartil salarial y el percentil 95 y el rango interdecil.
- Estudiar analíticamente la concentración de los salarios y explicarla.
- Calcular los factores de forma (asimetría y curtosis) de la distribución.
- Corroborar con **R** los resultados.

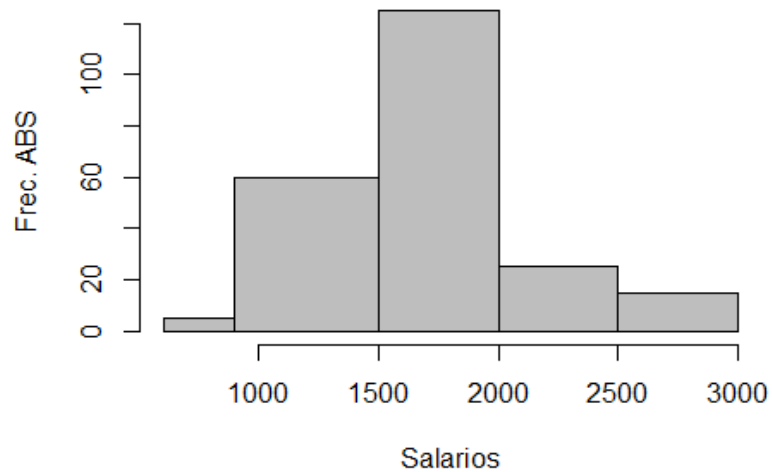
```
setwd("/")  
library(knitr)  
library(e1071)  
library(ineq)
```

Apartado a)

```
salarios <- c(rep(750,5),rep(1200,60),rep(1750,125),rep(2250,25)  
,rep(2750,15))  
Empleados<- c(5,60,125,25,15)  
N<- length(salarios)  
Freq_relativa <- Empleados/N  
hist(salarios,breaks = c(600,900,1500,2000,2500,3000) ,xlab ="Sa  
larios" , ylab = "Frec. ABS",  
      main = "Histograma Frecuencia Absoluta",col = "grey",freq =  
T)  
  
## Warning in plot.histogram(r, freq = freq1, col = col, border  
= border,  
## angle = angle, : the AREAS in the plot are wrong -- rather us
```

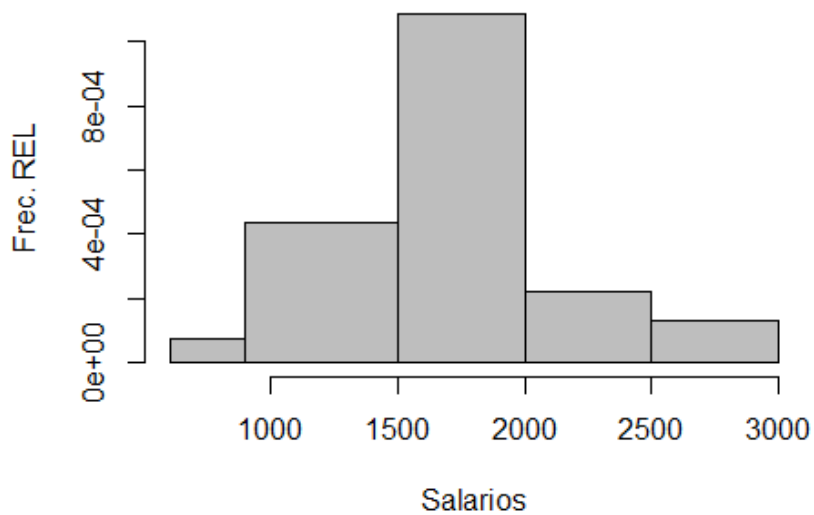
```
e 'freq =  
## FALSE'
```

**Histograma Frecuencia Absoluta**

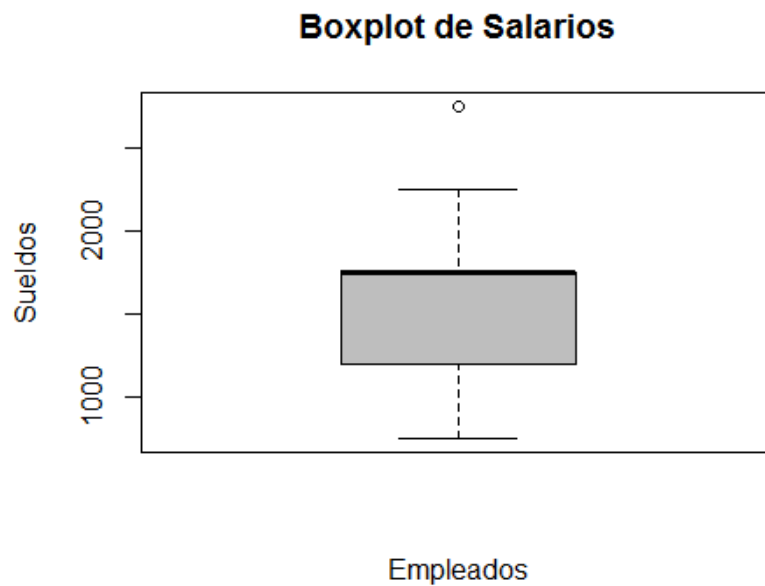


```
hist(salarios,breaks = c(600,900,1500,2000,2500,3000) ,xlab ="Sa  
larios" ,ylab ="Frec. REL" ,  
      main = "Histograma Frecuencia relaitva",col = "grey",freq =  
F)
```

**Histograma Frecuencia relaitva**



```
boxplot(salarios, main = "Boxplot de Salarios",col="grey", ylab=  
"Sueldos", xlab="Empleados")
```



Apartado b)

```
mean(salarios)
```

```
## [1] 1704.348
```

```
median(salarios)
```

```
## [1] 1750
```

```
Moda<-function(x){  
  ux<-unique(x)  
  ux[which.max(tabulate(match(x,ux)))]  
}
```

```
Moda(salarios)
```

```
## [1] 1750
```

```
var(salarios)
```

```
## [1] 191793.2
```

```
sd(salarios)
```

```
## [1] 437.9421
```

Apartado c)

```
empleados_salariomax <- salarios[salarios > 1800]
```

Apartado d)

```
Tercer_Q <- quantile(salarios,0.75)
```

```
kable(Tercer_Q)
```

	x
75%	1750

Apartado e)

```
percentil_95 <- quantile(salarios,0.95)
Recorrido_interdecil <- quantile(salarios,0.90)-quantile(salarios,0.10)
```

Apartado f)

```
# Concentracion = GiNi
Gini(salarios,corr = F)

## [1] 0.1310725
```

Apartado g)

```
skewness(salarios,na.rm = FALSE)

## [1] 0.4751668

kurtosis(salarios)

## [1] 0.3306589
```



## Cuestion L2.3

**Cuestión 3.** Los datos siguientes se corresponden con las causas más frecuentes de suspenso o abandono de la asignatura de Métodos Estadísticos:

- Falta de motivación por la asignatura (10)
- Escasa base matemática para comprender los conceptos (25).
- Horario del semestre (15).
- Carga de trabajo del curso donde se ubica la asignatura excesiva (37)
- Laboratorios deficientes (15)
- Prácticas muy laboriosas (8)
- Poco tiempo para el trabajo de curso (28)
- Explicaciones en clases teóricas no satisfactorias (10)
- Otras causas (6)

Se pide:

- a) Construir un diagrama de Pareto y evaluar los porcentajes de causas que se pueden explicar por categorías.
- b) Analizar gráficamente el problema y establecer conclusiones y recomendaciones para el/la profesor/a, el departamento y el centro.
- c) Construir un programa **R** que permita realizar los distintos apartados anteriores.

```
library(qcc)

## Package 'qcc' version 2.7

## Type 'citation("qcc")' for citing this R package in publications.

datos <- read.table(text = "Causas Suspensos", header = TRUE)
attach(datos)

Causas <- c("Falta de motivación por la asignatura",
            "Escasa base matemática para comprender los conceptos",
            "Horario del semestre",
            "Carga de trabajo del curso donde se ubica la asignatura excesiva",
            "Laboratorios deficientes",
            "Prácticas muy laboriosas",
            "Poco tiempo para el trabajo de curso",
            "Explicaciones en clases teóricas no satisfactorias",
            ,
            "Otras causas")
Suspensos <- c(10,25,15,37,15,8,28,10,6)
names(Causas) <- Suspensos
Causas

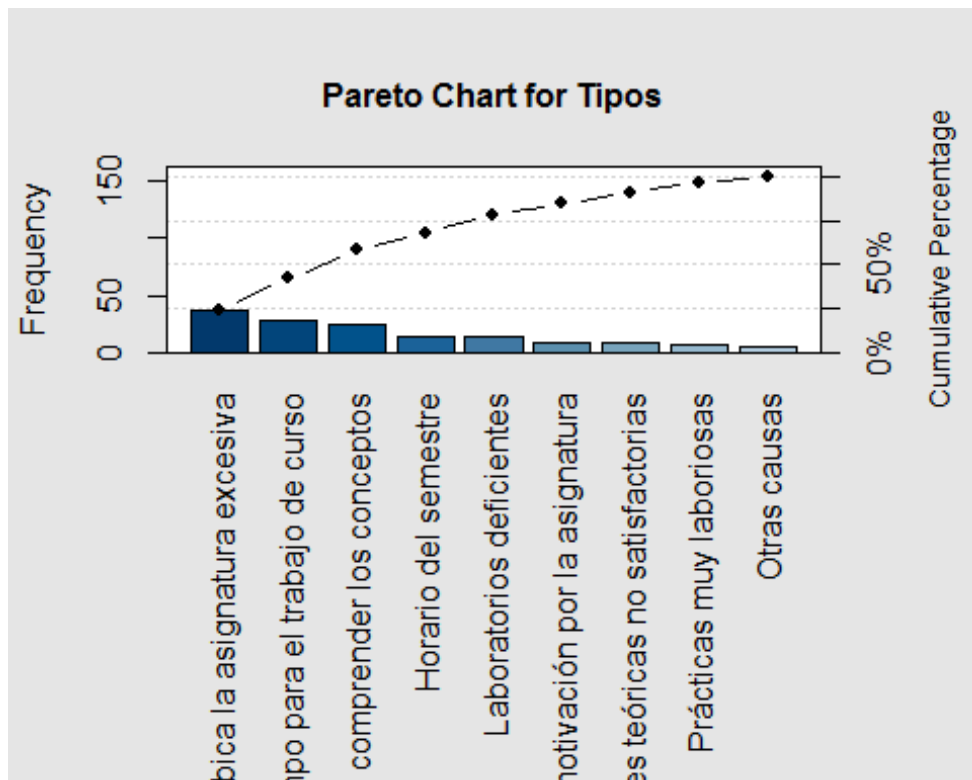
##
10
```

```

##                                "Falta de motivación por la asigna
tura"
##
25
##                                "Escasa base matemática para comprender los conce
ptos"
##
15
##                                "Horario del seme
stre"
##
37
## "Carga de trabajo del curso donde se ubica la asignatura exce
siva"
##
15
##                                "Laboratorios deficie
ntes"
##
8
##                                "Prácticas muy labori
osas"
##
28
##                                "Poco tiempo para el trabajo de c
urso"
##
10
##                                "Explicaciones en clases teóricas no satisfacto
rias"
##
6
##                                "Otras ca
usas"

Tipos <- Suspensos
names(Tipos) <- Causas
pareto.chart(Tipos)

```



```
##
## Pareto chart analysis for Tipos
##
Frequency
## Carga de trabajo del curso donde se ubica la asignatura exc
esiva 37.000000
## Poco tiempo para el trabajo de curso
28.000000
## Escasa base matemática para comprender los conceptos
25.000000
## Horario del semestre
15.000000
## Laboratorios deficientes
15.000000
## Falta de motivación por la asignatura
10.000000
## Explicaciones en clases teóricas no satisfactorias
10.000000
## Prácticas muy laboriosas
8.000000
## Otras causas
6.000000
##
## Pareto chart analysis for Tipos
##
Cum.Freq.
## Carga de trabajo del curso donde se ubica la asignatura exc
esiva 37.000000
```

```

## Poco tiempo para el trabajo de curso
65.000000
## Escasa base matemática para comprender los conceptos
90.000000
## Horario del semestre
105.000000
## Laboratorios deficientes
120.000000
## Falta de motivación por la asignatura
130.000000
## Explicaciones en clases teóricas no satisfactorias
140.000000
## Prácticas muy laboriosas
148.000000
## Otras causas
154.000000
##
## Pareto chart analysis for Tipos
##
Percentage
## Carga de trabajo del curso donde se ubica la asignatura exc
esiva 24.025974
## Poco tiempo para el trabajo de curso
18.181818
## Escasa base matemática para comprender los conceptos
16.233766
## Horario del semestre
9.740260
## Laboratorios deficientes
9.740260
## Falta de motivación por la asignatura
6.493506
## Explicaciones en clases teóricas no satisfactorias
6.493506
## Prácticas muy laboriosas
5.194805
## Otras causas
3.896104
##
## Pareto chart analysis for Tipos
##
Cum.Percent.
## Carga de trabajo del curso donde se ubica la asignatura exc
esiva 24.025974
## Poco tiempo para el trabajo de curso
42.207792
## Escasa base matemática para comprender los conceptos
58.441558
## Horario del semestre

```

68.181818  
## Laboratorios deficientes  
77.922078  
## Falta de motivación por la asignatura  
84.415584  
## Explicaciones en clases teóricas no satisfactorias  
90.909091  
## Prácticas muy laboriosas  
96.103896  
## Otras causas  
100.000000

### Conclusiones finales del ejercicio

De acuerdo a lo indicado en el ejercicio, y según los resultados del diagrama de Pareto, es posible extraer las siguientes conclusiones:

- Un gran porcentaje de los alumnos dejan o suspenden la asignatura debido tanto a la sobrecarga de trabajo como al poco tiempo que les queda para el trabajo final.
- Del restante, un bloque considerable de alumnos suspenden debido a carencias en los conocimientos básicos necesarios para la asignatura.
- Obviando el factor de los horarios de la asignatura, es importante reseñar que el mal estado de los laboratorios es acuciante y supone un motivo importante en el cómputo total.

Las medidas que deberían tomarse por parte del profesor son, por orden de prioridades:

- Reducir la carga de trabajo, lo cual ayudaría a mitigar el segundo causante, que es el poco tiempo disponible para el trabajo final, por efecto dominó.
- Procurar mejorar los horarios, proponiendo al rectorado mejores horarios para su asignatura.
- Solicitar mejoras en el equipamiento disponible, que abarquen laboratorios, materiales y demás enseres necesarios.

### Cuestion 3

#### Cuestionario 3

Aarón Hernández Álvarez.

1) a)

b)  $S = A \cap B \cap C$

c)  $S_1 = (A \cap B \cap \bar{C}) \cup (A \cap \bar{B} \cap C) \cup (\bar{A} \cap B \cap C)$

d)  $S_2 = (\bar{A} \cap \bar{B} \cap \bar{C}) \cup (\bar{A} \cap \bar{B} \cap C) \cup (\bar{A} \cap B \cap \bar{C}) \cup (A \cap \bar{B} \cap \bar{C})$

A = Pedido 1 a tiempo

B = Pedido 2 a tiempo

C = Pedido 3 a tiempo

$P(A) = 0,95$   $P(\bar{A}) = 0,05$

$P(B) = 0,95$   $P(\bar{B}) = 0,05$

$P(C) = 0,95$   $P(\bar{C}) = 0,05$

b)  $P(S) = 0,95^3 = 0,86 \rightarrow 86\%$

c)  $P(S_1) = (0,95^2 \cdot 0,05) + (0,95^2 \cdot 0,05) + (0,95^2 \cdot 0,05) = 3 \cdot (0,95^2 \cdot 0,05) = 0,14 \rightarrow \approx 14\%$

d)  $P(S_2) = (0,05^3) + (0,05^2 \cdot 0,95) + (0,05^2 \cdot 0,95) + (0,05^2 \cdot 0,95) = 0,0071 \approx 0,7\%$

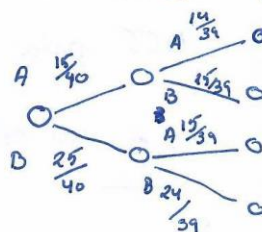
2) A: Media Markt (1er Monto)  
B: Corte Ingles (2do Monto)

a)  $P(A) = \frac{15}{40}$

b)  $P(B/A) = \frac{25}{39}$  \*\*

c)  $P(B/A) = \frac{15}{40} \cdot \frac{25}{39} = 0,24$

d)  $P(B/A) = 1 - P(\bar{B}/A) = 1 - \frac{125}{312} = 0,60$



\*  $P(\bar{B}/A) = P(\bar{B}/A) \cdot P(A) = \frac{125}{312}$

\*\*  $P(B/A) = P(B/A) / P(A)$

3) Jamosos del Sur. } Suministradores Mercadona  
Eubutidos Extremadura.

A la primera : 4.5 veces más que a la segunda  $\Rightarrow 1 + 4.5 = 5.5$

6% JS en mal estado

12% EE en mal estado

$P(JS)$  mal estado

$P(EE)$  mal estado

$$a) \begin{cases} P(H/JS) = 0,06 \\ P(H/EE) = 0,18 \end{cases}$$

$$P(H) = P(JS) \cdot (P(H/JS)) + P(EE) \cdot P(H/EE) \quad 7,2\%$$

$$P(H) = 0,06 \cdot 0,82 + 0,18 \cdot 0,12 = 0,0708 = 7,08\% \text{ en mal estado}$$

$$P(\bar{H}) = 1 - P(H) = 0,928 = 92,8\% \text{ en buen estado}$$

$$b) \frac{N}{P(\bar{H})} = \frac{1350}{0,928} = 1455 \text{ para saber cuanto es el total}$$

$$-1350 + 1455 = 105 \text{ en mal estado.}$$

54)



4)  $P \rightarrow$  En paro  
 Mujer  $\rightarrow 59,5\%$   
 Hombre  $\rightarrow 40,5\%$

$\begin{cases} 48,5\% \text{ Paro} \\ 52,5\% \text{ No paro} \end{cases}$   
 $\begin{cases} 27\% \text{ Paro} \\ 73\% \text{ No paro} \end{cases}$

$$P(H) = 59,5\%$$

$$P(H) = 40,5\%$$

$$P(P/H) = 27\%$$

$$P(P/H) = 47,5\%$$

Solución

$$P(H/P) = \frac{P(P \cap H)}{P(P)} = \frac{0,283}{0,39} = 0,725 = 72,5\%$$

$$P(P) = P(H) \cdot P\left(\frac{P}{H}\right) + P(H) \cdot P\left(\frac{P}{H}\right) = 0,405 \cdot 0,27 + 0,595 \cdot 0,475 = 0,39$$

$$P(P \cap H) = P(H) \cdot P\left(\frac{P}{H}\right) = 0,595 \cdot 0,475 = 0,283$$



6)  $P =$  Es Paro  $P(A) = 0,25$   $P(P) = 0,40$   $P(A \cap P) = 0,14$

$$a) P(P/A) = \frac{P(A \cap P)}{P(A)} = \frac{0,14}{0,25} = 0,56 \approx 56\%$$

$$b) P(A/P) = P(P \cap A) / P(P) = \frac{0,14}{0,4} = 0,35 \approx 35\%$$



7)  $P(A) = 0,9$   $P(\bar{A}) = 0,1$

$$P(B) = 0,9$$
  $P(\bar{B}) = 0,1$

$$a) P(\bar{A} \cap \bar{B}) = (0,1) \cdot (0,1) = 0,01$$

$$P(A \cup B) = 1 - P(\bar{A} \cap \bar{B}) = 0,99$$