Hoja 2 de problemas y prácticas con R

Estadística Computacional I. Grado en Estadística

Departamento de Estadística e Investigación Operativa. Universidad de Sevilla

Contents

1 Generar una muestra

De las calificaciones de 50 alumnos con el comando:

`` `sample(c("1S","2A","3N","4SB","5MH"),prob=c(0.3,0.35,0.2,0.1,0.05),50,rep=T)` ``.

```
set.seed(12345)
m1=sample(c("1S","2A","3N","4SB","5MH"),
          prob=c(0.3,0.35,0.2,0.1,0.05),50,rep=T)
m1
```

```
##  [1] "3N"  "4SB" "3N"  "4SB" "1S"  "2A"  "2A"  "1S"  "3N"  "5MH" "2A"  "2A"
## [13] "3N"  "2A"  "1S"  "1S"  "1S"  "1S"  "2A"  "5MH" "1S"  "2A"  "5MH" "3N"
## [25] "1S"  "1S"  "3N"  "1S"  "2A"  "1S"  "3N"  "2A"  "2A"  "3N"  "1S"  "1S"
## [37] "4SB" "4SB" "1S"  "2A"  "3N"  "1S"  "4SB" "3N"  "2A"  "2A"  "2A"  "2A"
## [49] "2A"  "1S"
```

1.1 Resumir valores

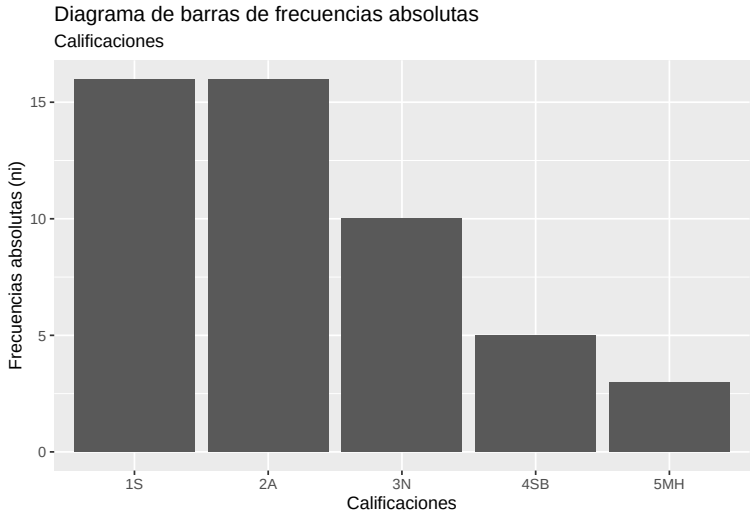
Mediante una tabla de frecuencias incluyendo frecuencias absolutas, frecuencias relativas, porcentajes, y los acumulados de las tres características.

```
tablafre=tibble(valores=m1) %>%
  group_by(valores) %>%
  summarise(
    ni=n() # Frecuencias absolutas
    ) %>%
  mutate(
    fi=ni /length(m1), #Frec rel
    pi=fi*100, #Porcentajes
    Ni=cumsum(ni),
    Fi=cumsum(fi),#Ni/length(m1)
    Pi=cumsum(pi)
    )

tablafre %>%
  kable(booktabs=TRUE) %>%
  kable_styling(latex_options = "striped"  )
```

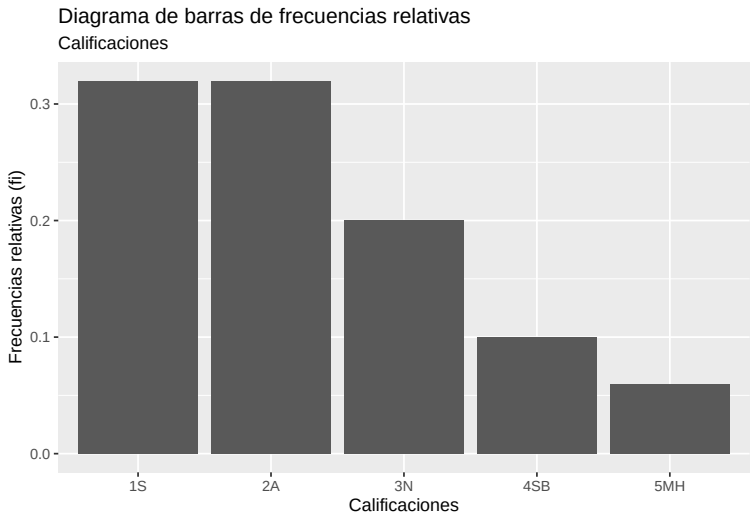
valores	ni	fi	pi	Ni	Fi	Pi
1S	16	0.32	32	16	0.32	32
2A	16	0.32	32	32	0.64	64
3N	10	0.20	20	42	0.84	84
4SB	5	0.10	10	47	0.94	94
5MH	3	0.06	6	50	1.00	100

1.2 Obtener representaciones gráficas adecuadas de las medidas anteriores.

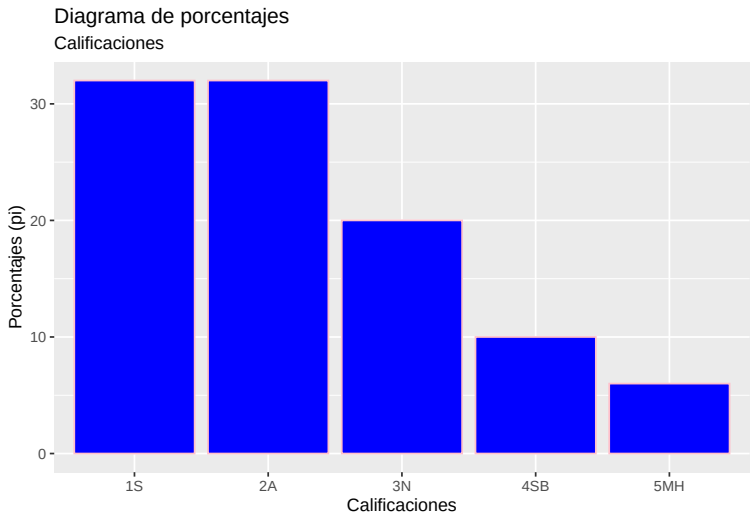
```
tablafre %>%
  ggplot(aes(x=valores , y = ni)) +
  geom_col()+
  labs(
    title = "Diagrama de barras de frecuencias absolutas",
    subtitle = "Calificaciones",
    y="Frecuencias absolutas (ni)",
    x="Calificaciones"
  )
```




```
tablafre %>%
  ggplot(aes(x=valores , y = fi)) +
  geom_col()+
  labs(
    title = "Diagrama de barras de frecuencias relativas",
    subtitle = "Calificaciones",
    y="Frecuencias relativas (fi)",
    x="Calificaciones"
  )
```




```
tablafre %>%
  ggplot(aes(x=valores , y = pi)) +
  geom_col( color = "pink", fill ="blue")+
  labs(
    title = "Diagrama de porcentajes",
    subtitle = "Calificaciones",
    y="Porcentajes (pi)",
    x="Calificaciones"
  )
```




```
tablafre %>%
  ggplot(aes(x=valores , y = Fi, group=1)) +
 # geom_col(color="blue") +
 # geom_line(color="pink")+ # Si no pongo group=1 no me hace la representación gráfica. Es para variabl
  geom_step(col="pink")+ #Variables discretas
  labs(
    title = "Polígono de Frecuencias rel acumuladas",
    subtitle = "Calificaciones",
    y="Frecuencias rel acumuladas (Fi)",
    x="Calificaciones"
  )
```

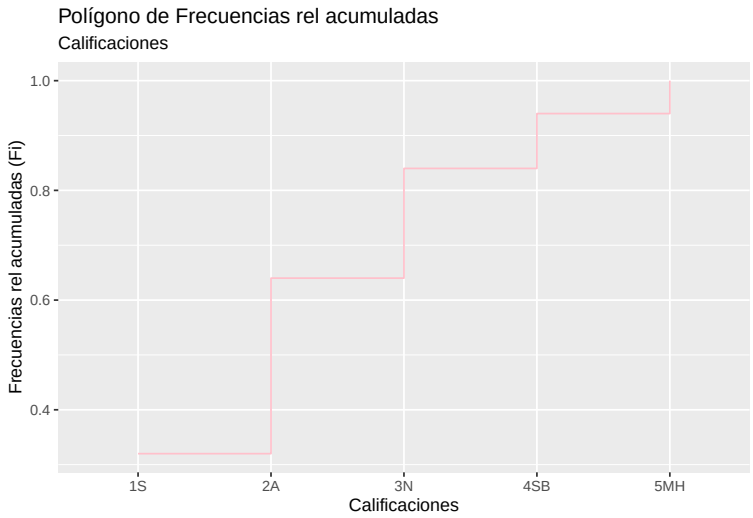


Diagrama de sectores con el sistema base:

```
tablafre$fi
```

```
## [1] 0.32 0.32 0.20 0.10 0.06
```

```
pie(tablafre$ni, labels = tablafre$valores)
```

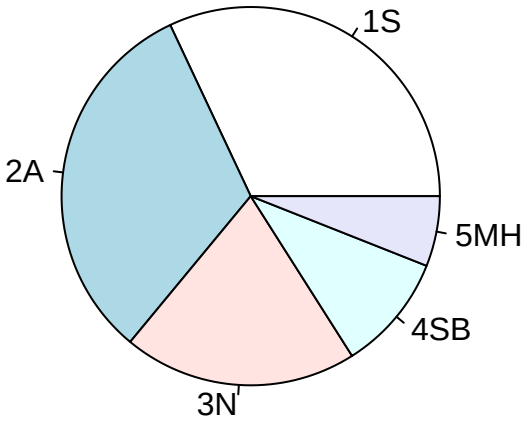
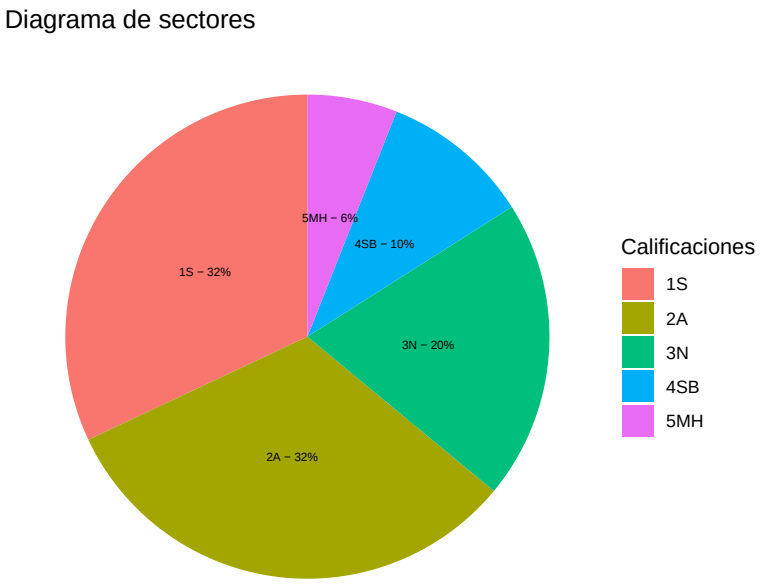


Diagrama de sectores con ggplot2:

```
tablafre %>%
  ggplot(aes(x="",y=pi, fill=factor(valores)))+
  geom_bar(width = 1, stat = "identity")+
  coord_polar("y",start = 0)+
  theme_void() +
  geom_text(aes(label=paste0(valores," - ",          round(pi,2), "%")),
   position=position_stack(vjust=0.5), size=2 ) +
  labs(
    title = "Diagrama de sectores",
    fill= "Calificaciones"
  )
```



2 Fichero "Familia.txt"

El fichero "Familia.txt" contiene el peso (kgs) y la altura (cms) de los integrantes de una familia.

2.1 Leer en R este fichero y calcular la media y la cuasidesviación típica de cada variable.

```
datos2=read.table(file="Familia.txt",sep=" ")
head(datos2) %>%
  kable(booktabs=TRUE) %>%
kable_styling(latex_options = "striped")
```

	Altura	Peso
Sobrina	120	22
Hijo	172	52
Abuelo	163	71
Hija	158	51
Sobrino	153	51
Abuela	148	60

```
datos2 %>%
  summarise(
    MediaAltura=mean(Altura),
    MediaPeso = mean(Peso),
    SdAltura=sd(Altura),
    SdPeso=sd(Peso)
  ) %>%
  kable(booktabs=TRUE) %>%
kable_styling(latex_options = "striped")
```

MediaAltura	MediaPeso	SdAltura	SdPeso
156.6	54.1	14.93839	13.56835

Otra forma:

```
datos2 %>%
  summarise_each(
   c( sd, mean)) %>%
  kable(booktabs=TRUE) %>%
kable_styling(latex_options = "striped")
```

Altura_fn1	Peso_fn1	Altura_fn2	Peso_fn2
14.93839	13.56835	156.6	54.1

Otra forma:

```
datos2 %>%
  summarise_all(
    list(mean,sd)
  )  %>%
kable(booktabs=TRUE) %>%
  kable_styling(latex_options = "striped")
```

Altura_fn1	Peso_fn1	Altura_fn2	Peso_fn2
156.6	54.1	14.93839	13.56835

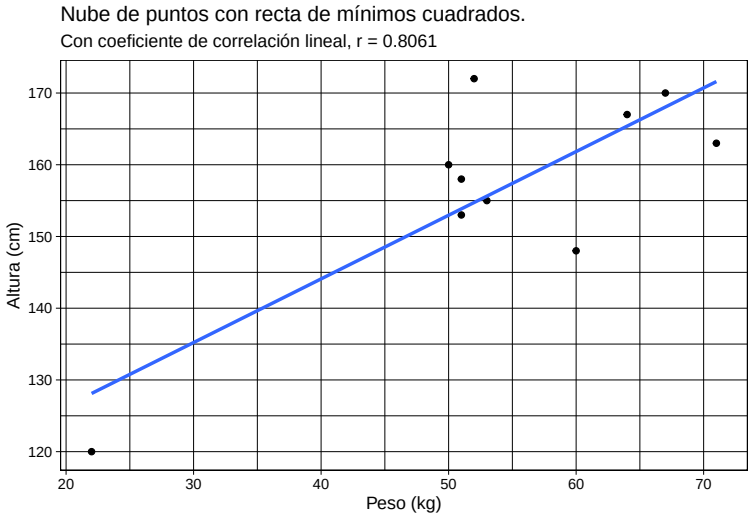
2.2 Nube de puntos y recta de mínimos cuadrados

Dibujar la nube de puntos (Peso, Altura) y superponer la recta de mínimos cuadrados. Calcular el coeficiente de correlación lineal entre ambas variables.

Coeficiente de correlación lineal

```
datos2 %>%
  summarise(
    Ccirlineal=cor(Peso,Altura)
    )-> ccl
```

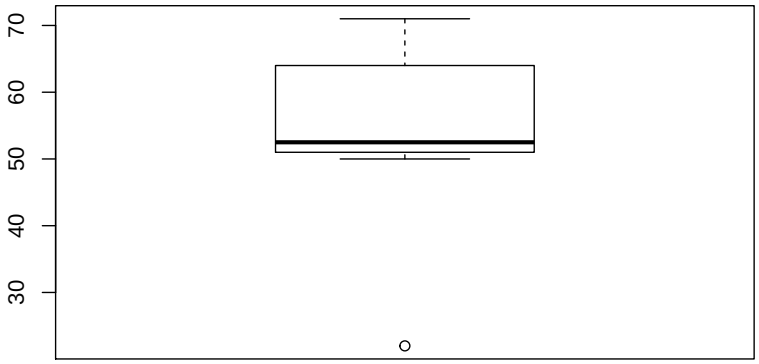
```
ggplot(data=datos2, aes(Peso,Altura)) +
  geom_point()+
  geom_smooth(method=lm, se=FALSE, formula = y~x)+  #Quito los IC
  labs(
    title = "Nube de puntos con recta de mínimos cuadrados.",
    subtitle = paste0("Con coeficiente de correlación lineal, r = " , round(ccl,4)),
    y="Altura (cm)",
    x="Peso (kg)" )+
  theme_linedraw()
```



2.3 Outliers

¿Qué observación es outlier para la variable peso? Repetir el apartado anterior sin esa persona.

```
res2=boxplot(datos2$Peso)
```




```
res2 # Mirlo las estadísticas del diagrama.
```

```
## $stats
##      [,1]
## [1,] 50.0
## [2,] 51.0
## [3,] 52.5
## [4,] 64.0
## [5,] 71.0
## attr(,"class")
##         1
## "integer"
##
## $n
## [1] 10
##
## $conf
##          [,1]
## [1,] 46.00468
## [2,] 58.99532
```

```
##
## $out
## [1] 22
##
## $group
## [1] 1
##
## $names
## [1] "1"
```

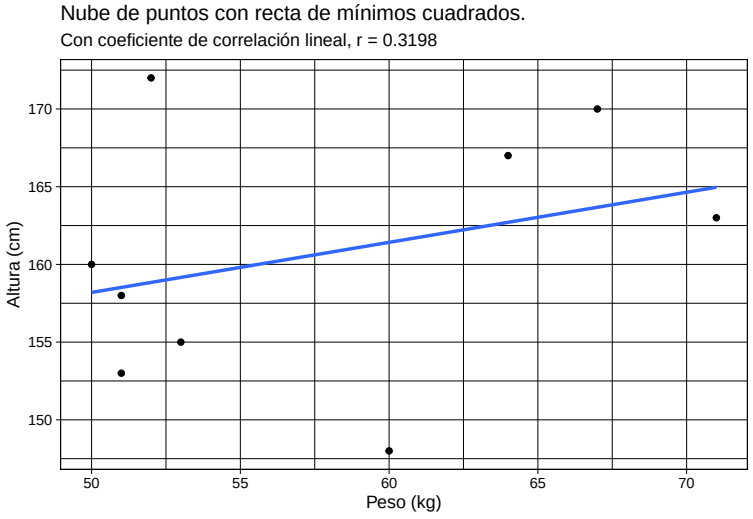
```
datos2 %>%
  arrange(Peso) %>%
  head(1)
```

```
##         Altura Peso
## Sobrina    120   22
```

```
datos2new=datos2[-1,]
```

```
datos2new %>%
  summarise(
    Ccirlineal=cor(Peso,Altura)
    )-> ccl
```

```
ggplot(data=datos2new, aes(Peso,Altura)) +
  geom_point()+
  geom_smooth(method=lm, se=FALSE, formula = y~x)+  #Quito los IC
  labs(
    title = "Nube de puntos con recta de mínimos cuadrados.",
    subtitle = paste0("Con coeficiente de correlación lineal, r = " , round(ccl,4)),
    y="Altura (cm)",
    x="Peso (kg)" )+
  theme_linedraw()
```



2.4 IMC

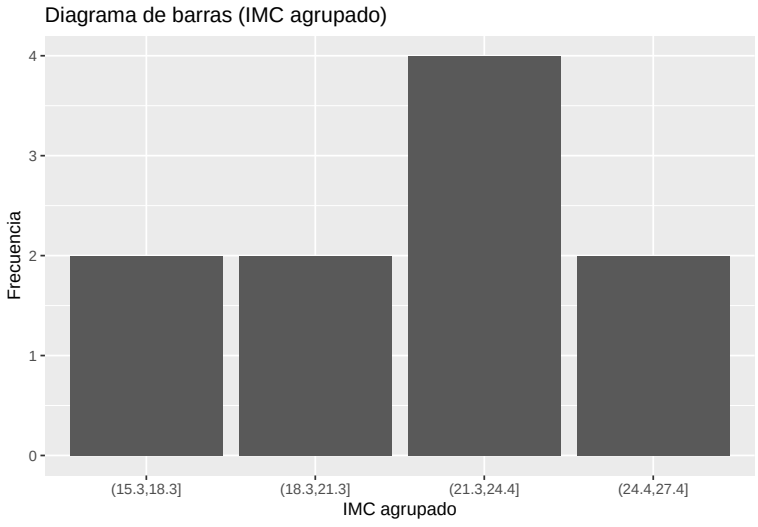
Calcular el Índice de Masa Corporal (IMC), definido como el cociente entre el peso y el cuadrado de la altura (en metros). Representarlo con un gráfico de barras.

```r
dat2imc= datos2 %>%
  mutate(IMC=Peso/((Altura/100)^2))

dat2imc %>%
  kable(booktabs=TRUE) %>%
kable_styling(latex_options = "striped")
```

	Altura	Peso	IMC
Sobrina	120	22	15.27778
Hijo	172	52	17.57707
Abuelo	163	71	26.72287
Hija	158	51	20.42942
Sobrino	153	51	21.78649
Abuela	148	60	27.39226
Tía	160	50	19.53125
Tío	170	67	23.18339
Madre	155	53	22.06035
Padre	167	64	22.94812

Vamos a definir cuatro intervalos para representar mis datos, empleamos el método del rango intercuartílico.

```r
dat2imc %>%
  mutate(
    IMCargu=cut(IMC,breaks=4)
  ) %>% ggplot(aes(x=IMCargu))+
  geom_bar()+
  labs(

  x="IMC agrupado",
  y="Frecuencia",
  title="Diagrama de barras (IMC agrupado)"
  )
```



2.5 Ordenar

Ordenar los familiares de mayor a menor IMC.

```
dat2imc %>%
  arrange(desc(IMC  ) ) %>%
  kable(booktabs=TRUE) %>%
  kable_styling(latex_options = "striped")
```

	Altura	Peso	IMC
Abuela	148	60	27.39226
Abuelo	163	71	26.72287
Tío	170	67	23.18339
Padre	167	64	22.94812
Madre	155	53	22.06035
Sobrino	153	51	21.78649
Hija	158	51	20.42942
Tía	160	50	19.53125
Hijo	172	52	17.57707
Sobrina	120	22	15.27778

3 Librería *ISwR*

Acceder al fichero alkfos de la librería ISwR:

```
library(ISwR)
data(alkfos) # Cargamos los datos
alkfos %>%
  kable(booktabs=TRUE,longtable=T,caption="Fichero alkfos") %>%
  kable_styling(latex_options = c("striped","repeat_header"))
```

Table 1: Fichero alkfos

grp	c0	c3	c6	c9	c12	c18	c24
1	142	140	159	162	152	175	148
1	120	126	120	146	134	119	116
1	175	161	168	164	213	194	221
1	234	203	174	197	289	174	189
1	94	107	146	124	128	98	114
1	128	97	113	203	NA	NA	NA
1	202	189	208	203	209	200	218
1	190	277	270	171	141	192	190
1	104	117	135	122	112	133	123
1	112	95	114	122	118	119	138
1	160	169	178	208	220	215	232
1	214	211	215	240	227	288	260
1	113	138	112	114	109	106	111
1	237	245	219	213	215	225	228
1	205	213	248	222	225	207	172
1	202	231	236	185	204	226	147
1	137	128	136	146	152	132	150

Table 1: Fichero alkfos *(continued)*

grp	c0	c3	c6	c9	c12	c18	c24
1	175	163	167	144	168	NA	NA
1	174	151	150	133	134	149	146
1	81	81	83	74	82	84	108
1	113	131	298	124	126	140	129
1	104	114	124	102	94	122	125
1	178	172	159	155	157	153	164
2	150	122	103	109	103	87	109
2	173	127	117	124	143	123	144
2	191	174	165	160	177	184	NA
2	191	159	157	161	150	187	215
2	230	150	144	153	125	124	152
2	145	134	167	141	112	212	194
2	128	92	89	78	83	78	80
2	102	86	80	76	82	79	68
2	180	124	116	117	124	NA	NA
2	153	96	97	96	93	156	110
2	115	79	79	79	73	69	72
2	150	113	124	102	100	109	101
2	182	147	156	79	135	NA	162
2	175	146	157	140	143	158	162
2	146	86	81	80	87	89	95
2	92	80	95	95	86	119	NA
2	228	177	185	181	190	182	192
2	178	119	107	NA	102	110	94
2	213	185	152	142	158	178	194
2	161	107	104	107	NA	118	129

3.1 Ver los primeros casos y los últimos.

```
alkfos[c(1,2,3,41,42,43),] %>%
  kable(booktabs=TRUE) %>%
  kable_styling(latex_options = "striped", stripe_index = c(1,2,5:6))
```

	grp	c0	c3	c6	c9	c12	c18	c24
1	1	142	140	159	162	152	175	148
2	1	120	126	120	146	134	119	116
3	1	175	161	168	164	213	194	221
41	2	178	119	107	NA	102	110	94
42	2	213	185	152	142	158	178	194
43	2	161	107	104	107	NA	118	129

```
alkfos %>% head() %>%  tail()
```

```
##   grp  c0  c3  c6  c9 c12 c18 c24
## 1   1 142 140 159 162 152 175 148
## 2   1 120 126 120 146 134 119 116
```

```
## 3   1 175 161 168 164 213 194 221
## 4   1 234 203 174 197 289 174 189
## 5   1  94 107 146 124 128  98 114
## 6   1 128  97 113 203  NA  NA  NA
```

3.2 Reformatear los datos a la estructura grp time c id.

Grupo, tiempo y valor observado.

```
alkfos_fl = alkfos %>%
  pivot_longer(names_to = "cid",
               values_to = "time",
               cols = -grp) %>%
  #para que aparezca primero el 3 luego el 6... sería conveniente quitar la c
  mutate(
    cid=as.integer(gsub("c","",cid))
  ) %>%
  select(grp,time,cid)
# View(alkfos_fl)
```

3.3 Ordenar el nuevo formato por grp, id y time, y mostrar las variables en el orden (grp, id, time, c).

```
alkfos_fl %>%
  arrange(grp,cid,time) %>%
  select(grp,cid,time) %>%
  head(6) %>%
  tail(6) %>%
  kable(booktabs=TRUE) %>%
  kable_styling("striped")
```

grp	cid	time
1	0	81
1	0	94
1	0	104
1	0	104
1	0	112
1	0	113

4 Fichero *"dietas.dat"*

Leer el fichero "dietas.dat", donde se distinguen 4 dietas alimenticias, y se mide el peso durante 6 meses.

```
dietas=read.table("dietas.dat",sep = " ", header = FALSE)
datos_dietas=dietas %>%
  rename(
    Dieta=V1,
    PesosM1=V2,
    PesosM2=V3,
    PesosM3=V4,
    PesosM4=V5,
    PesosM5=V6,
```

```
    PesosM6=V7
  )
datos_dietas %>%
  head()
```

```
##   Dieta PesosM1 PesosM2 PesosM3 PesosM4 PesosM5 PesosM6
## 1     1       3       1       1       2       1       6
## 2     1       2       2       1       2       1       3
## 3     1       2       1       1       1       1       2
## 4     1       1       1       1       1       1       1
## 5     1       2       2       1       4       2       5
## 6     1       2       2       2       2       2       4
```

Cada variable toma valores de 1 a 6, a mayor valor más lejos del peso ideal. Representar gráficamente la evolución de los pesos medios según la dieta.

Está en formato ancho, deberíamos pasarlo al formato largo para poder manipular mejor los datos.

```
datos_dietas_largo=datos_dietas %>%
  pivot_longer(
    names_to = "Mes",
    values_to = "Peso",
    cols=-Dieta
  ) %>%

  mutate(
    # Mes = as.numeric(gsub("PesosM","",Mes))
  Mes =  recode(Mes,
                "PesosM1"=1L,
                "PesosM2"=2L,
                "PesosM3"=3L,
                "PesosM4"=4L,
                "PesosM5"=5L,
                "PesosM6"=6L )
  ) %>%
  arrange(Dieta,Mes)
#L obliga a que el dato sea entero en lugar de numeric, aunque no pasaría nada.
#
datos_dietas_largo %>%
  head() %>%
  kable(booktabs=TRUE) %>%
  kable_styling("striped")
```

Dieta	Mes	Peso
1	1	3
1	1	2
1	1	2
1	1	1
1	1	2
1	1	2

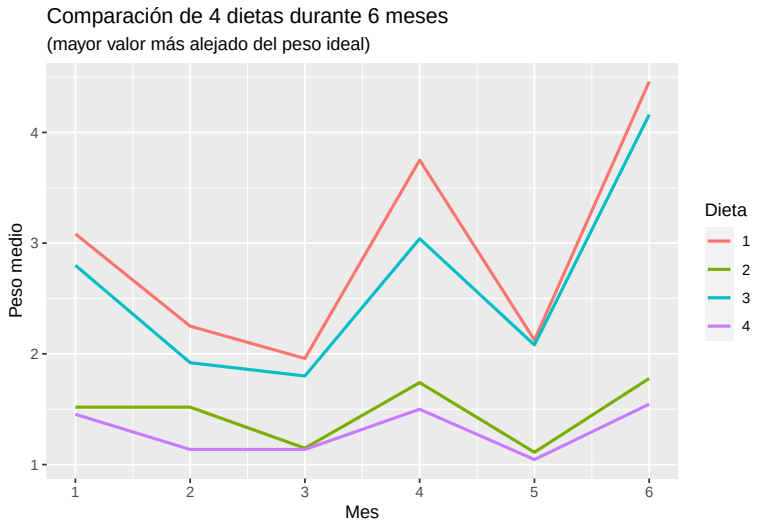
Vamos a calcular los pesos medios, por meses y para cada dieta.

```
media_dietas=
  datos_dietas_largo %>%
  group_by(Dieta,Mes) %>%
  summarise(PesoMedio=mean(Peso,na.rm=TRUE))

media_dietas %>%
  kable(booktabs=TRUE,longtable=TRUE) %>%
  kable_styling("striped")
```

Dieta	Mes	PesoMedio
1	1	3.083333
1	2	2.250000
1	3	1.958333
1	4	3.750000
1	5	2.125000
1	6	4.458333
2	1	1.518519
2	2	1.518519
2	3	1.148148
2	4	1.740741
2	5	1.111111
2	6	1.777778
3	1	2.800000
3	2	1.920000
3	3	1.800000
3	4	3.040000
3	5	2.080000
3	6	4.160000
4	1	1.454546
4	2	1.136364
4	3	1.136364
4	4	1.500000
4	5	1.045454
4	6	1.545454

Hacemos ahora la representación

```
media_dietas %>%
  ggplot(aes(x=Mes,y=PesoMedio,colour=factor(Dieta)))+
  geom_line(size=0.9)+
  scale_x_continuous(breaks = 1:6)+
  labs(
    title="Comparación de 4 dietas durante 6 meses",
    subtitle = "(mayor valor más alejado del peso ideal)",
    y="Peso medio",
    colour="Dieta" # Puedo hacerlo porque al definir le he llamado así.
  )
```



Dieta 3 un poco mejor que la 1.

La dieta 2 se estabiliza en valores bajos y sube y baja.

La que se mantiene siemrpe cerca e valores bajos es la 4, por lo que podría ser la mejor dieta.

5 Datos *Boston*

En este problema se trabajará con el conjunto de datos Boston de la librería MASS.

5.1 Acceder a los datos, en particular, averiguar qué información contiene y cuál es la dimensionalidad de los datos.

```
library(MASS)
?Boston
# gls(Boston)
glimpse(Boston)
```

```
## Rows: 506
## Columns: 14
## $ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985, 0.08829,~
## $ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5, 12.5, 1~
## $ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87, 7.87, 7.~
## $ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524, 0.524,~
## $ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172, 5.631,~
## $ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0, 85.9, 9~
## $ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605, 5.9505~
## $ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4,~
## $ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311, 311, 31~
## $ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2, 15.2, 15~
## $ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12, 395.60, 396.90~
## $ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.93, 17.10~
## $ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15~
```

5.2 Ordenar las variables según el valor absoluto de su coeficiente de correlación lineal con medv (variable a predecir en este conjunto de datos).

```
CorB=Boston %>%
  map_dbl(~cor(.x,Boston$medv)) #Calcula coef corr lineal con la vble medv
sort(abs(CorB))
```

```
##      chas       dis     black        zn       age       rad      crim       nox
## 0.1752602 0.2499287 0.3334608 0.3604453 0.3769546 0.3816262 0.3883046 0.4273208
##       tax     indus   ptratio        rm     lstat      medv
## 0.4685359 0.4837252 0.5077867 0.6953599 0.7376627 1.0000000
```

5.3 Comentarios de variables

5.3.1 Tasa de criminalidad

```
Boston %>%
  arrange(desc(crim)) %>%
    dplyr::select(crim,everything()) %>%
  slice(1:5,(nrow(Boston)-4):nrow(Boston)) # 5 primeras y 5 últimas. 506-4=504,
```

```
##        crim zn indus chas   nox    rm   age    dis rad tax ptratio  black lstat
## 1  88.97620  0 18.10    0 0.671 6.968  91.9 1.4165  24 666    20.2 396.90 17.21
## 2  73.53410  0 18.10    0 0.679 5.957 100.0 1.8026  24 666    20.2  16.45 20.62
## 3  67.92080  0 18.10    0 0.693 5.683 100.0 1.4254  24 666    20.2 384.97 22.98
## 4  51.13580  0 18.10    0 0.597 5.757 100.0 1.4130  24 666    20.2   2.60 10.11
## 5  45.74610  0 18.10    0 0.693 4.519 100.0 1.6582  24 666    20.2  88.27 36.98
## 6   0.01311 90  1.22    0 0.403 7.249  21.9 8.6966   5 226    17.9 395.93  4.81
## 7   0.01301 35  1.52    0 0.442 7.241  49.3 7.0379   1 284    15.5 394.74  5.49
## 8   0.01096 55  2.25    0 0.389 6.453  31.9 7.3073   1 300    15.3 394.72  8.23
## 9   0.00906 90  2.97    0 0.400 7.088  20.8 7.3073   1 285    15.3 394.72  7.85
## 10  0.00632 18  2.31    0 0.538 6.575  65.2 4.0900   1 296    15.3 396.90  4.98
##    medv
## 1  10.4
## 2   8.8
## 3   5.0
## 4  15.0
## 5   7.0
## 6  35.4
## 7  32.7
## 8  22.0
## 9  32.2
## 10 24.0
```

```
# de esa a la 506
```

5.3.2 Tasa de propiedad

```
Boston %>%
  arrange(desc(tax)) %>%
  dplyr::select(tax,everything()) %>%
  slice(1:5,(nrow(Boston)-4):nrow(Boston)) # Slice selecciona filas: slice(1) Da la 1 fila
```

```
##    tax    crim zn indus chas   nox    rm  age     dis rad ptratio  black lstat
## 1  711 0.15086  0 27.74    0 0.609 5.454 92.7  1.8209   4    20.1 395.09 18.06
```

```
## 2  711 0.18337  0 27.74    0 0.609 5.414 98.3  1.7554   4    20.1 344.05 23.97
## 3  711 0.20746  0 27.74    0 0.609 5.093 98.0  1.8226   4    20.1 318.43 29.68
## 4  711 0.10574  0 27.74    0 0.609 5.983 98.8  1.8681   4    20.1 390.11 18.07
## 5  711 0.11132  0 27.74    0 0.609 5.983 83.5  2.1099   4    20.1 396.90 13.35
## 6  188 0.15038  0 25.65    0 0.581 5.856 97.0  1.9444   2    19.1 370.31 25.41
## 7  188 0.09849  0 25.65    0 0.581 5.879 95.8  2.0063   2    19.1 379.38 17.58
## 8  188 0.16902  0 25.65    0 0.581 5.986 88.4  1.9929   2    19.1 385.02 14.81
## 9  188 0.38735  0 25.65    0 0.581 5.613 95.6  1.7572   2    19.1 359.29 27.26
## 10 187 0.01709 90  2.02    0 0.410 6.728 36.1 12.1265   5    17.0 384.46  4.50
##    medv
## 1  15.2
## 2   7.0
## 3   8.1
## 4  13.6
## 5  20.1
## 6  17.3
## 7  18.8
## 8  21.4
## 9  15.7
## 10 30.1
```

5.3.3 Ratio Alumno-Profesor

```
Boston %>%
  arrange(desc(ptratio)) %>%
  dplyr::select(ptratio,everything()) %>%
  slice(1:5,(nrow(Boston)-4):nrow(Boston))
```

```
##    ptratio    crim zn indus chas   nox    rm  age     dis rad tax  black lstat
## 1     22.0 0.04301 80  1.91    0 0.413 5.663 21.9 10.5857   4 334 382.80  8.05
## 2     22.0 0.10659 80  1.91    0 0.413 5.936 19.5 10.5857   4 334 376.04  5.57
## 3     21.2 0.25915  0 21.89    0 0.624 5.693 96.0  1.7883   4 437 392.11 17.19
## 4     21.2 0.32543  0 21.89    0 0.624 6.431 98.8  1.8125   4 437 396.90 15.39
## 5     21.2 0.88125  0 21.89    0 0.624 5.637 94.7  1.9799   4 437 396.90 18.34
## 6     13.0 0.57834 20  3.97    0 0.575 8.297 67.0  2.4216   5 264 384.54  7.44
## 7     13.0 0.54050 20  3.97    0 0.575 7.470 52.6  2.8720   5 264 390.30  3.16
## 8     12.6 0.04011 80  1.52    0 0.404 7.287 34.1  7.3090   2 329 396.90  4.08
## 9     12.6 0.04666 80  1.52    0 0.404 7.107 36.6  7.3090   2 329 354.31  8.61
## 10    12.6 0.03768 80  1.52    0 0.404 7.274 38.3  7.3090   2 329 392.20  6.62
##    medv
## 1  18.2
## 2  20.6
## 3  16.2
## 4  18.0
## 5  14.3
## 6  50.0
## 7  43.5
## 8  33.3
## 9  30.3
## 10 34.6
```

5.4 ¿Cuántos distritos son limítrofes con el río?

```
Boston %>%
  filter(chas==1) %>%
  summarise(
    CuantosDistritos=n()
  )
```

```
##   CuantosDistritos
## 1               35
```

5.5 Calcular las medias de crim y medv según chas.

```
Boston %>%
  group_by(chas) %>%
  summarise(
    Media_crim= mean(crim,na.rm=TRUE),
    Media_medv= mean(medv,na.rm=TRUE)
  ) %>%
  mutate(
    limitrofeRio=recode(chas,
                        '0'="No limítrofe",
                        '1'="Si limítrofe") #Podríamos haber recodificado chas.
  )
```

```
## # A tibble: 2 x 4
##    chas Media_crim Media_medv limitrofeRio
##   <int>      <dbl>      <dbl> <chr>
## 1     0       3.74       22.1 No limítrofe
## 2     1       1.85       28.4 Si limítrofe
```

5.6 Analizar la relación lineal entre lstat y medv.

```
regrelineal=Boston %>%
 dplyr:: select(lstat,medv) %>%
  lm(formula = lstat~medv)
regrelineal
```

```
##
## Call:
## lm(formula = lstat ~ medv, data = .)
##
## Coefficients:
## (Intercept)         medv
##     25.5589      -0.5728
```

```
# summary(regrelineal)
```

6 Teorema de Fisher

Comprobar empíricamente el Teorema de Fisher a partir de 5000 muestras de tamaño 10 de una ley N(0,1):

6.1 Analizar la relación lineal entre las medias y las cuasivarianzas.

```
set.seed(1234)
dat6=map_dfc(1:5000, ~rnorm(10))
dat6[1:6,1:6]
```

```
## # A tibble: 6 x 6
##     ...1    ...2   ...3   ...4   ...5   ...6
##    <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 -1.21  -0.477   0.134  1.10   1.45  -1.81
## 2  0.277 -0.998  -0.491 -0.476 -1.07  -0.582
## 3  1.08  -0.776  -0.441 -0.709 -0.855 -1.11
## 4 -2.35   0.0645  0.460 -0.501 -0.281 -1.01
## 5  0.429  0.959  -0.694 -1.63  -0.994 -0.162
## 6  0.506 -0.110  -1.45  -1.17  -0.969  0.563
```

Las medias

```
medias=map_dbl(dat6, mean); head(medias)
```

```
##       ...1       ...2       ...3       ...4       ...5       ...6
## -0.3831574 -0.1181707 -0.3879468 -0.7661931 -0.6097971 -0.2788647
```

Cuasivarianzas

```
cuasivar= dat6 %>%
  map_dbl(var)
head(cuasivar)
```

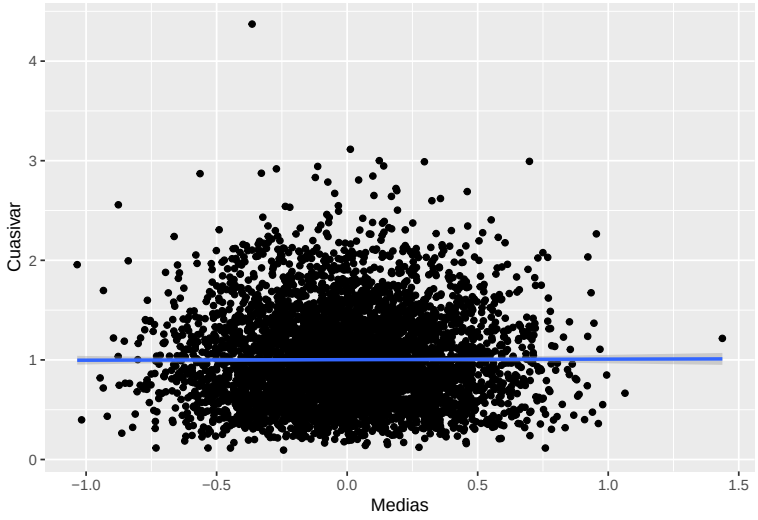
```
##      ...1      ...2      ...3      ...4      ...5      ...6
## 0.9915928 1.1392095 0.4435577 0.7996756 0.6196134 1.4065456
```

```
dat6c=tibble(
  Medias=medias ,
  Cuasivar= cuasivar
)

dat6c %>%
  head(6) %>%
  kable(booktabs=TRUE) %>%
  kable_styling("striped")
```

Medias	Cuasivar
-0.3831574	0.9915928
-0.1181707	1.1392095
-0.3879468	0.4435577
-0.7661931	0.7996756
-0.6097971	0.6196134
-0.2788647	1.4065456

Para el diagrama de dispersión

```
dat6c %>%
  ggplot(aes(x=Medias , y=Cuasivar))+
  geom_point()+
  geom_smooth(method = "lm") # Recta que mejor se adapta a la línea de puntos.
```



Los puntos crean como un círculo, no existe relación lineal ni tampoco ningún tipo de relación. La recta es horizontal, por lo que las variables están incorreladas.

Vamos a calcular el modelo lineal:

```
dat6c %>%
  lm(formula = Cuasivar~Medias ,data=.) %>%  # El punto coloca lo que está a la izquierda.
  summary()
```

```
##
## Call:
## lm(formula = Cuasivar ~ Medias, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9067 -0.3403 -0.0753  0.2751  3.3714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.002469   0.006512  153.93   <2e-16 ***
## Medias      0.005396   0.020734    0.26    0.795
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4605 on 4998 degrees of freedom
## Multiple R-squared:  1.355e-05,  Adjusted R-squared:  -0.0001865
## F-statistic: 0.06774 on 1 and 4998 DF,  p-value: 0.7947
```

Por tanto, el modelo es el siguiente: Cuasivar $= 0,99 - 0,0484$ Media. Vemos que $R^2 = 0,0008$, por lo que confirmamos la incorrelación e independencia de las variables.

6.2 Estudiar gráficamente si los cocientes (n-1)*cuasivar/(sigma^2) siguen una ley chi-cuadrado.

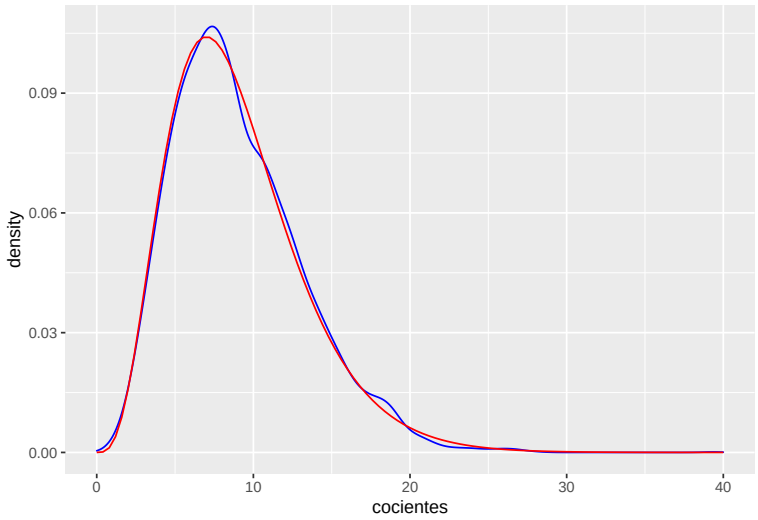
```
dat6c2=dat6c %>%
  mutate(
    cocientes = (10-1)*Cuasivar/(1^2) # Los datos provienen de una Normal(0,1)
  )          # Mutate añade nuevas columnas
```

```
dat6c2 %>%
  head(6)
```

```
## # A tibble: 6 x 3
##   Medias Cuasivar cocientes
##    <dbl>    <dbl>     <dbl>
## 1 -0.383    0.992      8.92
## 2 -0.118    1.14      10.3
## 3 -0.388    0.444      3.99
## 4 -0.766    0.800      7.20
## 5 -0.610    0.620      5.58
## 6 -0.279    1.41      12.7
```

Vamos a representar la *función de densidad* con un histograma:

```
dat6c2 %>%
  ggplot(aes(x=cocientes))+
  # geom_histogram()+
  geom_density(color="blue")+
  stat_function(aes(x=seq(0,40,length.out=5000)),
                fun = dchisq ,
                args = list(df=10-1), color="red") # Superpongo la curva de la Chi
```




```
 # geom_function(aes(x=seq(0,40,length.out=5000)),
   #              fun = dchisq ,
   #              args = list(df=10-1), color="red")



# Divide el intervalo 40 en 5000 valores
```

Lo hacemos para 20 muestras.

```
set.seed(1234)
dat6=map_dfc(1:20, ~rnorm(10))
dat6[1:6,1:6]
```

```
## # A tibble: 6 x 6
##     ...1    ...2   ...3   ...4   ...5   ...6
```

```
##    <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 -1.21  -0.477   0.134  1.10   1.45  -1.81
## 2  0.277 -0.998  -0.491 -0.476 -1.07  -0.582
## 3  1.08  -0.776  -0.441 -0.709 -0.855 -1.11
## 4 -2.35   0.0645  0.460 -0.501 -0.281 -1.01
## 5  0.429  0.959  -0.694 -1.63  -0.994 -0.162
## 6  0.506 -0.110  -1.45  -1.17  -0.969  0.563
```

Las medias

```
medias=map_dbl(dat6, mean); head(medias)
```

```
##       ...1       ...2       ...3       ...4       ...5       ...6
## -0.3831574 -0.1181707 -0.3879468 -0.7661931 -0.6097971 -0.2788647
```

Cuasivarianzas

```
cuasivar= dat6 %>%
  map_dbl(var)
head(cuasivar)
```

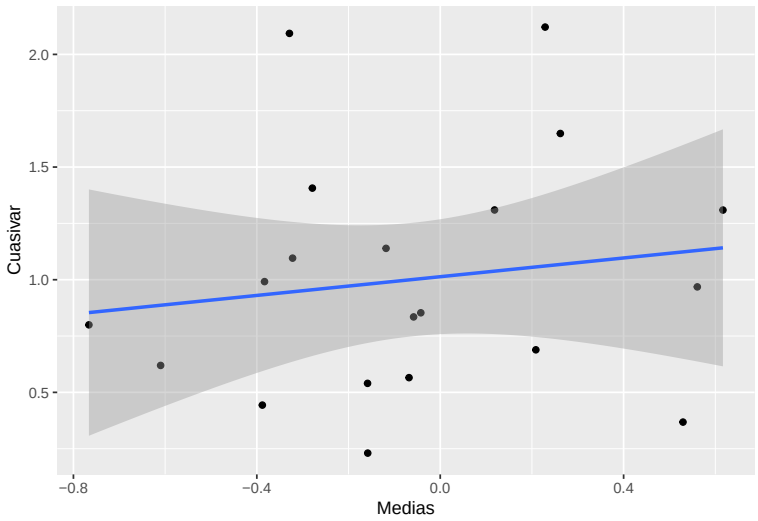
```
##      ...1      ...2      ...3      ...4      ...5      ...6
## 0.9915928 1.1392095 0.4435577 0.7996756 0.6196134 1.4065456
```

```
dat6c=tibble(
  Medias=medias ,
  Cuasivar= cuasivar
)

dat6c %>%
  head(6) %>%
  kable(booktabs=TRUE) %>%
  kable_styling("striped")
```

Medias	Cuasivar
-0.3831574	0.9915928
-0.1181707	1.1392095
-0.3879468	0.4435577
-0.7661931	0.7996756
-0.6097971	0.6196134
-0.2788647	1.4065456

Para el diagrama de dispersión

```
dat6c %>%
  ggplot(aes(x=Medias , y=Cuasivar))+
  geom_point()+
  geom_smooth(method = "lm") # Recta que mejor se adapta a la línea de puntos.
```



Los puntos crean como un círculo, no existe relación lineal ni tampoco ningún tipo de relación. La recta es horizontal, por lo que las variables están incorreladas.

Vamos a calcular el modelo lineal:

```
dat6c %>%
  lm(formula = Cuasivar~Medias ,data=.) %>%  # El punto coloca lo que está a la izquierda.
  summary()
```

```
##
## Call:
## lm(formula = Cuasivar ~ Medias, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7553 -0.3844 -0.1029  0.1936  1.1482
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0134     0.1214   8.350 1.33e-07 ***
## Medias        0.2078     0.3259   0.638    0.532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5362 on 18 degrees of freedom
## Multiple R-squared:  0.02209,    Adjusted R-squared:  -0.03224
## F-statistic: 0.4066 on 1 and 18 DF,  p-value: 0.5317
```

Por tanto, el modelo es el siguiente: Cuasivar $= 0,99 - 0,0484$ Media. Vemos que $R^2 = 0,0008$, por lo que confirmamos la incorrelación e independencia de las variables.

6.3 Estudiar gráficamente si los cocientes (n-1)*cuasivar/(sigma^2) siguen una ley chi-cuadrado.

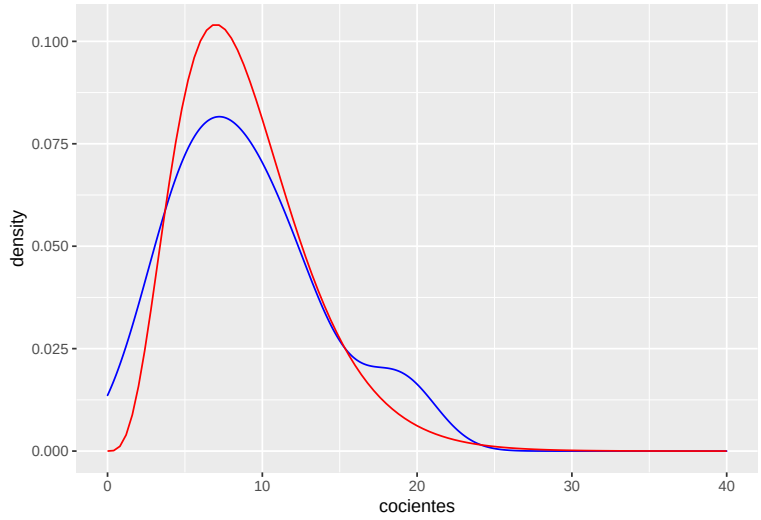
```
dat6c2=dat6c %>%
  mutate(
    cocientes = (10-1)*Cuasivar/(1^2) # Los datos provienen de una Normal(0,1)
  )          # Mutate añade nuevas columnas
```

```
dat6c2 %>%
  head(6)
```

```
## # A tibble: 6 x 3
##   Medias Cuasivar cocientes
##    <dbl>    <dbl>     <dbl>
## 1 -0.383    0.992      8.92
## 2 -0.118    1.14      10.3
## 3 -0.388    0.444      3.99
## 4 -0.766    0.800      7.20
## 5 -0.610    0.620      5.58
## 6 -0.279    1.41      12.7
```

Vamos a representar la *función de densidad* con un histograma:

```
dat6c2 %>%
  ggplot(aes(x=cocientes))+
  # geom_histogram()+
  geom_density(color="blue")+
  # stat_function(aes(x=seq(0,40,length.out=20)),
  #               fun = dchisq ,
  #               args = list(df=10-1), color="red") # Superpongo la curva de la Chi
  geom_function(aes(x=seq(0,40,length.out=20)),
                fun = dchisq ,
                args = list(df=10-1), color="red")
```




```
# Divide el intervalo 40 en 5000 valores
```

Vemos que con un tamaño 20 no se ajusta igual.

Hay una coincidencia bastante buena, por lo que con una muestra de tamaño 5000, se aproxima bastante.

7 Simulaciones

Comprobar mediante una simulación el ajuste de las distribuciones chi-cuadrado y la distribución F-Snedecor a partir de las cuasivarianzas muestrales para 10000 pares de muestras independientes. En cada par, la primera muestra será de tamaño 10 de la ley N(0,1), y la segunda muestra de tamaño 8 de la ley N(10,3).

Creamos 10000 muestras de tamaño 10 para la X y de tamaño 8 para la Y.

```r
set.seed(1234)
n=10000
nX=10
mdX=0
sigmaX=1

nY=8
mdY=10
sigmaY=sqrt(3)

dat7_NormalX=map_dfc(1:n , ~rnorm(nX,mean = mdX,sd=sigmaX))
dat7_NormalY=map_dfc(1:n , ~rnorm(nY,mean = mdY,sd=sigmaY))
```

Vamos a calcular ahora las cuasivarianzas una vez generadas las muestras.

```r
cuasivarX=dat7_NormalX %>%
  map_dbl(var)
cuasivarY=dat7_NormalY %>%
  map_dbl(var)


F= (cuasivarX/cuasivarY)*(sigmaY^2/sigmaX^2) # Cocientes muestrales que deben seguir una F

ChiX=((nX-1)*cuasivarX)/(sigmaX^2)
ChiY=((nY-1)*cuasivarY)/(sigmaY^2)
```

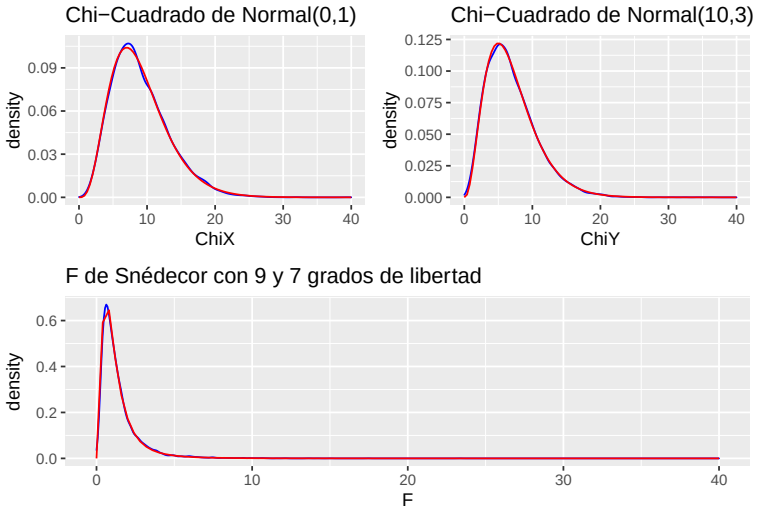
Vamos a ver si se ajustan bien a la teórica:

```r
# install.packages("patchwork")
library(patchwork)
```

```r
dat7=tibble(
  F=F ,
  ChiX=ChiX ,
  ChiY=ChiY
)
```

```r
p1=dat7 %>%
  ggplot(aes(x=ChiX))+
  geom_density(color="blue")+
  stat_function(aes(x=seq(0,40, length.out=n)),
                    fun=dchisq ,
                    args=list(df=nX-1),
                color="red")+
  labs(title="Chi-Cuadrado de Normal(0,1)")


p2=dat7 %>%
  ggplot(aes(x=ChiY))+
  geom_density(color="blue")+
  stat_function(aes(x=seq(0,40, length.out=n)),
                    fun=dchisq ,
                    args=list(df=nY-1),
                color="red")+ labs(title="Chi-Cuadrado de Normal(10,3)")
# F
```

```
p3=dat7 %>%
  ggplot(aes(x=F))+
  geom_density(color="blue")+
  stat_function(aes(x=seq(0,40, length.out=n)),
                    fun=df ,
                    args=list(df1=nX-1, df2=nY-1),
                color="red")+
  labs(title="F de Snédecor con 9 y 7 grados de libertad ")


(p1|p2)/p3
```




```
# p1/p2
#
```

Vemos que todos los modelos se ajustan muy bien.

8 Fichero *"salarios.txt"*

El fichero "salarios.txt" contiene datos sobre el salario (variable wage) y otras características para 3000 trabajadores.

8.1 Leer en R los datos.

```
salarios=read.table(file="salarios.txt",header = TRUE,sep = " ") # SEP no es necesario, ya que read.tab
salarios %>%
  head() %>%
  kable(booktabs=TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```

	year	age	sex	maritl	race	education	region	jobclass	health	health_ins	logwage	wage
231655	2006	18	1. Male	1. Never Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industrial	1. <=Good	2. No	4.318063	75.04315
86582	2004	24	1. Male	1. Never Married	1. White	4. College Grad	2. Middle Atlantic	2. Information	2. >=Very Good	2. No	4.255273	70.47602
161300	2003	45	1. Male	2. Married	1. White	3. Some College	2. Middle Atlantic	1. Industrial	1. <=Good	1. Yes	4.875061	130.98218
155159	2003	43	1. Male	2. Married	3. Asian	4. College Grad	2. Middle Atlantic	2. Information	2. >=Very Good	1. Yes	5.041393	154.68529
11443	2005	50	1. Male	4. Divorced	1. White	2. HS Grad	2. Middle Atlantic	2. Information	1. <=Good	1. Yes	4.318063	75.04315
376662	2008	54	1. Male	2. Married	1. White	4. College Grad	2. Middle Atlantic	2. Information	2. >=Very Good	1. Yes	4.845098	127.11574

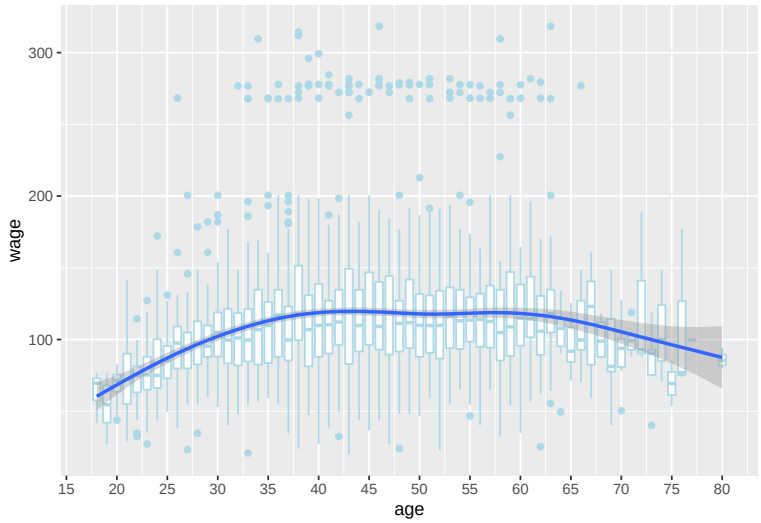
```
salarios %>% glimpse()
```

```
## Rows: 3,000
## Columns: 12
## $ year       <int> 2006, 2004, 2003, 2003, 2005, 2008, 2009, 2008, 2006, 2004,~
## $ age        <int> 18, 24, 45, 43, 50, 54, 44, 30, 41, 52, 45, 34, 35, 39, 54,~
## $ sex        <fct> 1. Male, 1. Male, 1. Male, 1. Male, 1. Male, 1. Male, 1. Ma~
## $ maritl     <fct> 1. Never Married, 1. Never Married, 2. Married, 2. Married,~
## $ race       <fct> 1. White, 1. White, 1. White, 3. Asian, 1. White, 1. White,~
## $ education  <fct> 1. < HS Grad, 4. College Grad, 3. Some College, 4. College ~
## $ region     <fct> 2. Middle Atlantic, 2. Middle Atlantic, 2. Middle Atlantic,~
## $ jobclass   <fct> 1. Industrial, 2. Information, 1. Industrial, 2. Informatio~
## $ health     <fct> 1. <=Good, 2. >=Very Good, 1. <=Good, 2. >=Very Good, 1. <=~
## $ health_ins <fct> 2. No, 2. No, 1. Yes, 1. Yes, 1. Yes, 1. Yes, 1. Yes, 1. Ye~
## $ logwage    <dbl> 4.318063, 4.255273, 4.875061, 5.041393, 4.318063, 4.845098,~
## $ wage       <dbl> 75.04315, 70.47602, 130.98218, 154.68529, 75.04315, 127.115~
```

8.2 Representar gráficamente los salarios según las variables age, year y education, y superponer estimaciones de la media del salario según cada variable.

Edad

```
salarios %>%
  ggplot(aes(x=age , y=wage))+
  geom_boxplot(aes(group=age), colour="lightblue")+
  scale_x_continuous(breaks=seq(0,90,by=5))+ # Voy a manipular el EJE X
  geom_smooth() # Para ver la tendencia. Para cada edad, representa el valor medio. Añade un IC, que se
```

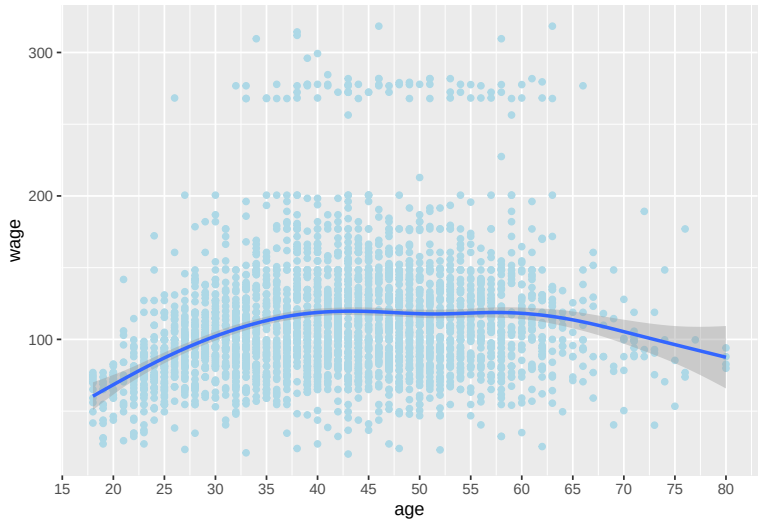



```
  # geom_smooth(method = "lm") # Con la recta de regresión
```

Con diagrama de puntos en vez de caja y bigote:

```
p8.1=salarios %>%
  ggplot(aes(x=age , y=wage))+
  geom_point(aes(group=age), colour="lightblue")+
  scale_x_continuous(breaks=seq(0,90,by=5))+ # Voy a manipular el EJE X
  geom_smooth() # Para ver la tendencia. Para cada edad, representa el valor medio. Añade un IC, que se
  # geom_smooth(method = "lm") # Con la recta de regresión
```

```
p8.1
```



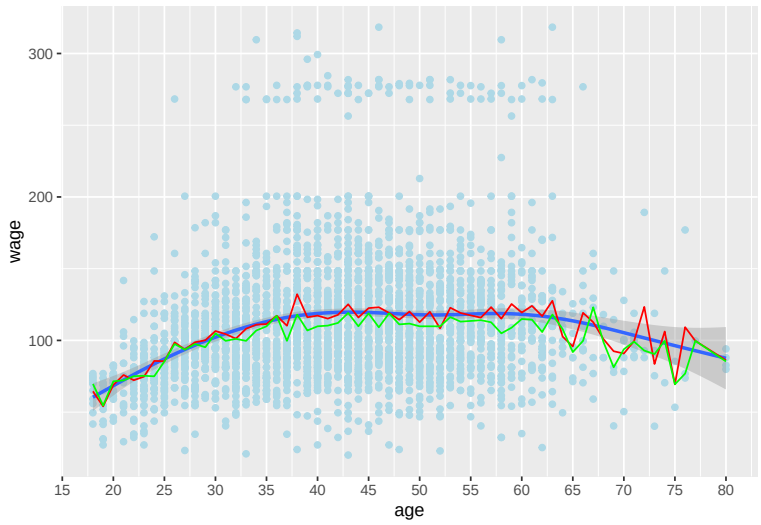
No tenemos el cálculo de la mediana de forma directa aquí. Podríamos calcular por cada edad la media, mediana,... y pintarlo.

```
part01=split(salarios,salarios$age) # Lista de 61 elementos, cada elemento tiene información para cada
dat8a= tibble(
  edades=as.numeric(names(part01)),
  medias = part01 %>% map_dbl(~mean(.x$wage, na.rm=T)), # Cada una de las listas de part01 y el conteni
  medianas=part01 %>% map_dbl(~median(.x$wage, na.rm=T))
)
dat8a %>% head()
```

```
## # A tibble: 6 x 3
##   edades medias medianas
##    <dbl>  <dbl>    <dbl>
## 1     18   64.5     69.6
## 2     19   54.0     54.6
## 3     20   69.0     71.5
## 4     21   75.9     72.2
## 5     22   72.3     75.0
## 6     23   74.7     75.4
```

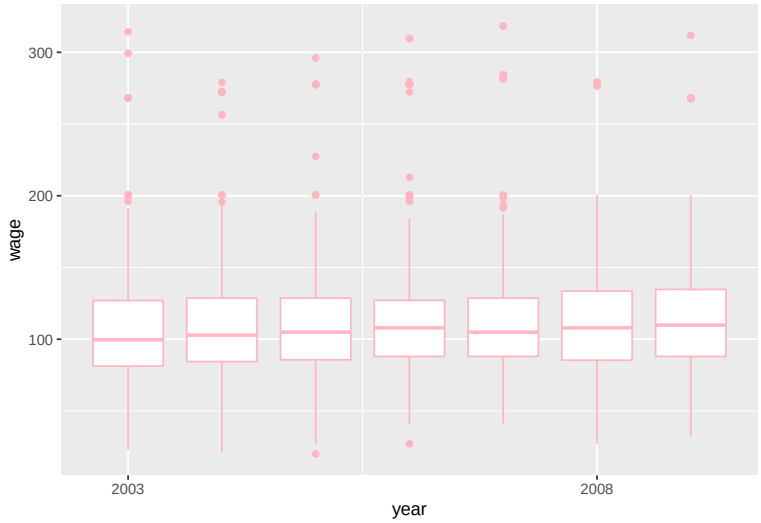
Para representar la evolución, superponemos al gráfico anterior:

```
p8.1+
  geom_line(data = dat8a, aes(x=edades,y=medias), color="red")+
  geom_line(data = dat8a, aes(x=edades,y=medianas), color="green")
```



Año

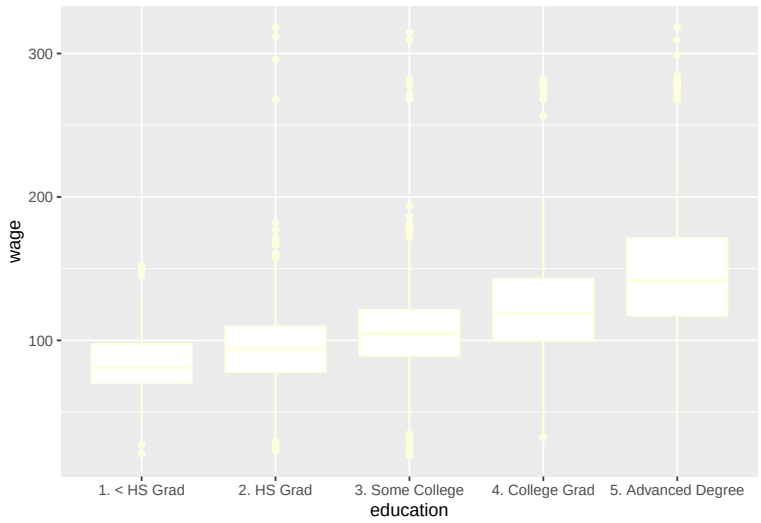
```
salarios %>%
  ggplot(aes(x=year , y=wage))+
  geom_boxplot(aes(group=year), colour="lightpink")+
  scale_x_continuous(breaks=seq(2003,2009,by=5))+ # Voy a manipular el EJE X
  geom_smooth() # Para ver la tendencia. Para cada edad, representa el valor medio. Añade un IC, que se
```




```
  # geom_smooth(method = "lm") # Con la recta de regresión
```

Nivel de educación

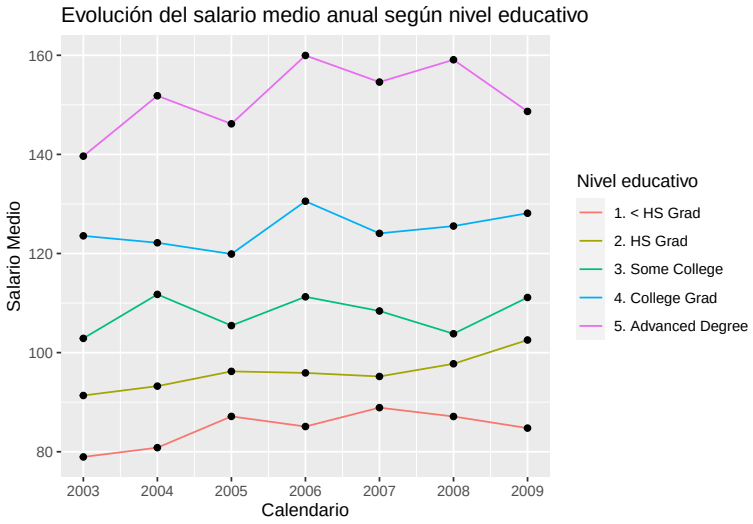
```
salarios %>%
  ggplot(aes(x=education , y=wage))+
  geom_boxplot(aes(group=education), colour="lightyellow")+
  # scale_x_continuous(breaks=seq(2003,2009,by=5))+ # Voy a manipular el EJE X
  geom_smooth() # Para ver la tendencia. Para cada edad, representa el valor medio. Añade un IC, que se
```




```
  # geom_smooth(method = "lm") # Con la recta de regresión
```

8.3 Dibujar la evolución anual del salario medio según el nivel educativo.

```
salario_med_nivel_educ=salarios %>%
  group_by(year, education) %>% # Porque esevolución ANUAL
  summarise(MedSalario=mean(wage,na.rm=TRUE))

salario_med_nivel_educ %>%
  ggplot(aes(x=year , y=MedSalario, group=education))+
  geom_line(aes(color=education))+
  geom_point()+
  scale_x_continuous(breaks=2003:2009)+
  labs(
    title="Evolución del salario medio anual según nivel educativo",
    x="Calendario",
    y="Salario Medio",
    color="Nivel educativo" # Porque he dicho en geom_line aes(color=)
  )
```



Parece que el comportamiento es parecido, pero a mayor nivel de educación, mayor es el salario.

8.4 Calcular los porcentajes de variación interanual del salario medio según nivel educativo.

```
salario_med_nivel_educ %>%
  head()
```

```
## # A tibble: 6 x 3
## # Groups:   year [2]
##    year education          MedSalario
##   <int> <fct>                   <dbl>
## 1  2003 1. < HS Grad             78.9
## 2  2003 2. HS Grad               91.3
## 3  2003 3. Some College         103.
## 4  2003 4. College Grad         124.
## 5  2003 5. Advanced Degree      140.
## 6  2004 1. < HS Grad             80.8
```

Salario medio de un año y le divido el salario medio del año anterior:

- 1: no hay variación
- 1.10: 10% de variación (crecimiento)
- 0.90: Ha disminuido un 90%.

Vamos a preparar los datos para que sea más fácil:

```
salario_med_nivel_educ %>%
  group_by(education) %>%
  mutate(año_anterior= lag(year,1), # Al agrupar por nivel de educación, calcula el salario medio y le
         MedSalAct= MedSalario,
         MedSalAnt=lag(MedSalario,1)) %>% # 1 es una posición
  head(15)
```

```
## # A tibble: 15 x 6
## # Groups:   education [5]
##     year education          MedSalario año_anterior MedSalAct MedSalAnt
##    <int> <fct>                   <dbl>        <int>     <dbl>     <dbl>
##  1  2003 1. < HS Grad             78.9           NA      78.9      NA
##  2  2003 2. HS Grad               91.3           NA      91.3      NA
##  3  2003 3. Some College         103.            NA     103.       NA
##  4  2003 4. College Grad         124.            NA     124.       NA
##  5  2003 5. Advanced Degree      140.            NA     140.       NA
##  6  2004 1. < HS Grad             80.8         2003      80.8      78.9
##  7  2004 2. HS Grad               93.2         2003      93.2      91.3
##  8  2004 3. Some College         112.          2003     112.      103.
##  9  2004 4. College Grad         122.          2003     122.      124.
## 10  2004 5. Advanced Degree      152.          2003     152.      140.
## 11  2005 1. < HS Grad             87.1         2004      87.1      80.8
## 12  2005 2. HS Grad               96.2         2004      96.2      93.2
## 13  2005 3. Some College         105.          2004     105.      112.
## 14  2005 4. College Grad         120.          2004     120.      122.
## 15  2005 5. Advanced Degree      146.          2004     146.      152.
```

```
# Si fuera mes, sería 12
```

Calculamos lo que nos piden:

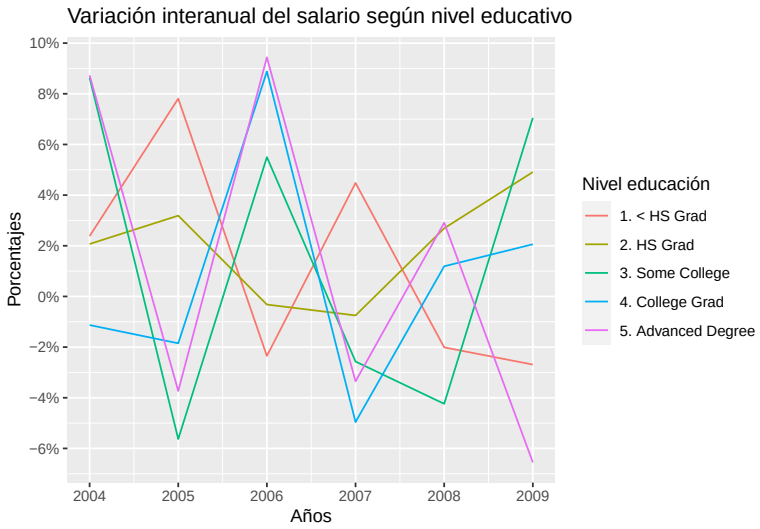
```
Var_Interanual=salario_med_nivel_educ %>%
  group_by(education) %>%
  mutate(año_anterior= lag(year,1),
         MedSalAct= MedSalario,
         MedSalAnt=lag(MedSalario,1),
         Cociente = MedSalAct/MedSalAnt,
         IncrAnualPorc=round(Cociente*100-100,2)
         ) %>%
  filter(year !=2003 )
```

```
head(Var_Interanual) %>%
  kable(booktabs=TRUE) %>%
  kable_styling(c("striped","scale_down"))
```

year	education	MedSalario	año_anterior	MedSalAct	MedSalAnt	Cociente	IncrAnualPorc
2004	1. < HS Grad	80.82215	2003	80.82215	78.94714	1.0237503	2.38
2004	2. HS Grad	93.23782	2003	93.23782	91.34757	1.0206929	2.07
2004	3. Some College	111.74949	2003	111.74949	102.87548	1.0862597	8.63
2004	4. College Grad	122.16354	2003	122.16354	123.56053	0.9886939	-1.13
2004	5. Advanced Degree	151.82982	2003	151.82982	139.64439	1.0872605	8.73
2005	1. < HS Grad	87.13149	2004	87.13149	80.82215	1.0780645	7.81

Vamos a representarlo grñaficamente:

```
g1=Var_Interanual %>%
  ggplot(aes(x=year,y=IncrAnualPorc)) +
  geom_line(aes(color=education))+
  scale_y_continuous(breaks = seq(-10,10,by=2),
                     labels =paste0( seq(-10,10,by=2), "%"))+
  labs(
    title="Variación interanual del salario según nivel educativo",
          y="Porcentajes",
          x="Años",
          color="Nivel educación")

g1
```



Interpretación

```
ggplot2::ggsave(filename = "gg1_visalario.png")
```

8.5 Ordenar el fichero de datos según año (creciente) y edad (decreciente).

```
salarios %>%
  arrange(year, desc(age)) %>%
  head() %>%
  kable(booktabs=TRUE) %>%
  kable_styling(c("scale_down","striped"))
```

	year	age	sex	maritl	race	education	region	jobclass	h
155488	2003	80	1. Male	2. Married	1. White	2. HS Grad	2. Middle Atlantic	2. Information	1
154262	2003	80	1. Male	2. Married	1. White	3. Some College	2. Middle Atlantic	2. Information	2
154263	2003	80	1. Male	2. Married	1. White	2. HS Grad	2. Middle Atlantic	1. Industrial	2
153846	2003	74	1. Male	3. Widowed	2. Black	4. College Grad	2. Middle Atlantic	2. Information	1
158165	2003	73	1. Male	2. Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industrial	2
153897	2003	72	1. Male	2. Married	1. White	5. Advanced Degree	2. Middle Atlantic	2. Information	1

9 Librería *MASS*

Acceder al data frame painters de la librería MASS.

```
painter=MASS::painters
glimpse(painter)
```

```
## Rows: 54
## Columns: 5
## $ Composition <int> 10, 15, 8, 12, 0, 15, 8, 15, 4, 17, 10, 13, 10, 15, 13, 12~
## $ Drawing     <int> 8, 16, 13, 16, 15, 16, 17, 16, 12, 18, 13, 15, 15, 14, 14,~
## $ Colour      <int> 16, 4, 16, 9, 8, 4, 4, 7, 10, 12, 8, 8, 6, 7, 10, 5, 6, 12~
## $ Expression  <int> 3, 14, 7, 8, 0, 14, 8, 6, 4, 18, 8, 8, 6, 10, 9, 8, 10, 6,~
## $ School      <fct> A, A, A, A, A, A, A, A, A, A, B, B, B, B, B, B, C, C, C, C~
```

```
?MASS::painters
```

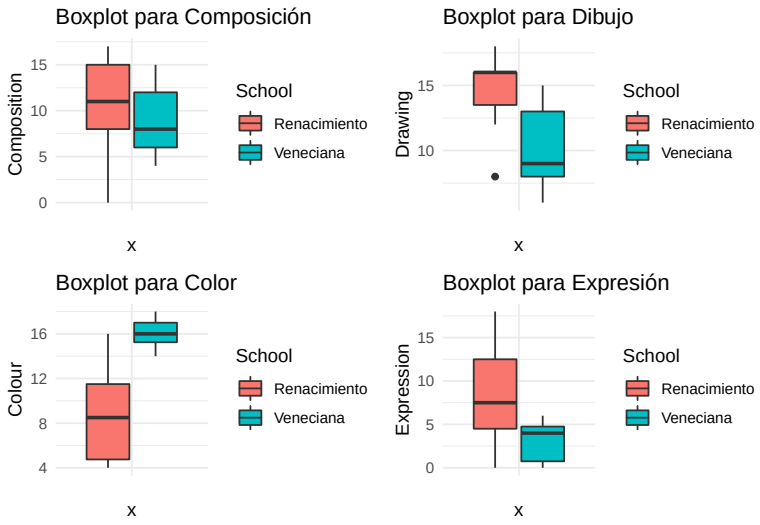
9.1 Interpretar y resumir la información contenida en este fichero de datos.

9.2 Seleccionar las escuelas del renacimiento y Veneciana para los siguientes apartados.

```
dat9=painter %>%
  filter(School %in% c("A","D")) %>%
  mutate(
    School= ifelse(School=="A","Renacimiento","Veneciana")
  )
view(dat9)
# También se puede hacer con recode y case when.
# ¡¡¡HACER!!!
```

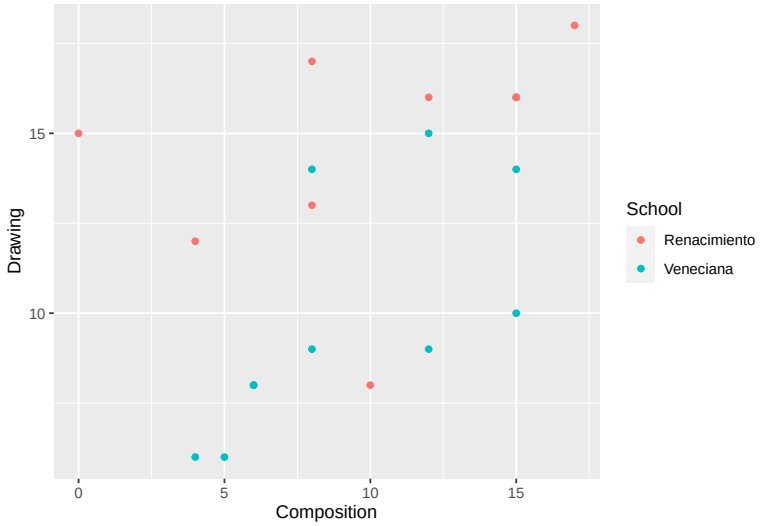
9.3 Generar en una sola pantalla los diagramas de caja y bigotes según la escuela.

```
p1=
ggplot(dat9) +
 aes(y = Composition, x="", fill = School) +
 geom_boxplot() +
 scale_fill_hue() +
    labs(title="Boxplot para Composición")+
 theme_minimal()

p2=ggplot(dat9) +
 aes(y = Drawing, x="", fill = School) +
 geom_boxplot() +
 scale_fill_hue() +
    labs(title="Boxplot para Dibujo")+
 theme_minimal()

p3=ggplot(dat9) +
 aes(y = Colour, x="", fill = School) +
 geom_boxplot() +
 scale_fill_hue() +
    labs(title="Boxplot para Color")+
 theme_minimal()

p4=ggplot(dat9) +
 aes( x=" ",y = Expression, fill = School) +
 geom_boxplot() +
 scale_fill_hue() +
  labs(title="Boxplot para Expresión")+
 theme_minimal()

library(patchwork)
(p1|p2)/(p3|p4)
```



9.4 Construir nubes de puntos en las que se distinga la escuela.

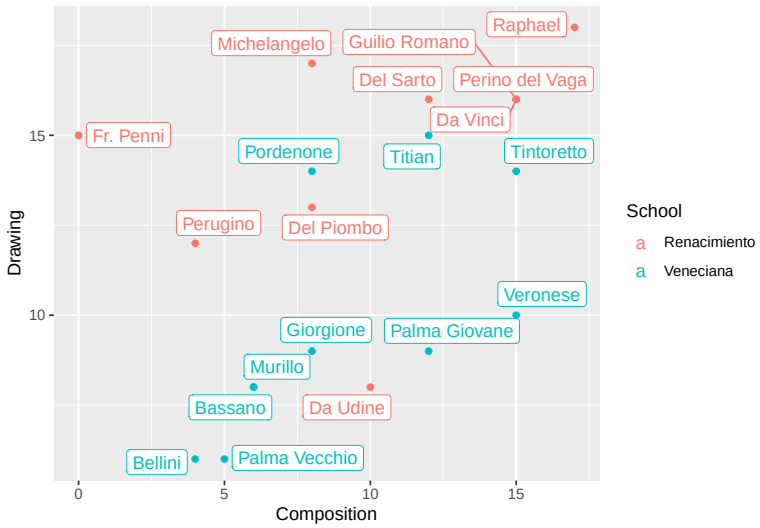
```
dat9 %>%
  ggplot(aes(x= Composition, y= Drawing, fill=School,colour=School))+
  geom_point()
```




```
dat9b=painter %>%
  filter(School %in% c("A","D")) %>%
  mutate(School= ifelse(School=="A","Renacimiento","Veneciana"),
         Nombres=rownames(.))
```

```
# install.packages("ggrepel")
library(ggrepel)
```

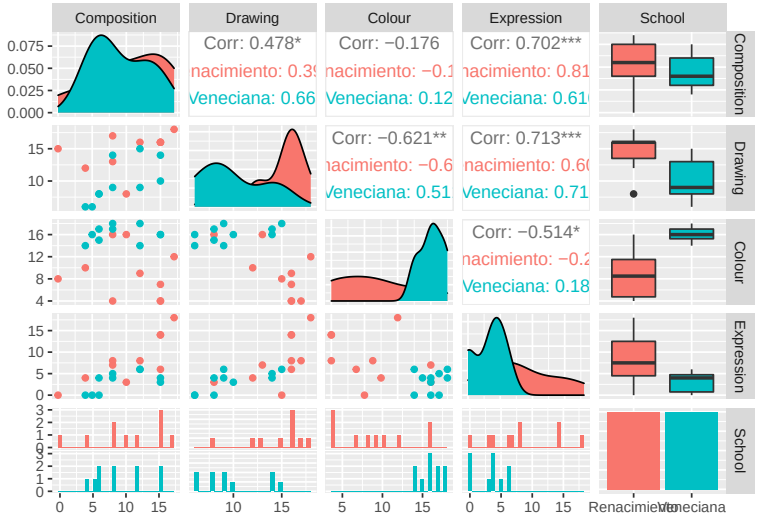
```
dat9b %>%
  ggplot(aes(x=Composition,y=Drawing, color=School, label=Nombres))+
  geom_point()+
#  geom_text() Al añadir la nueva librería puedo usar la siguiente función
# geom_text_repel() # Separa las coincidencias
  geom_label_repel() # Mejora aún la presentación de geom_text_repel
```



Este gráfico lo puedo hacer para todas las parejas, no solo composción y dibujo.

Vamos a intentarlo hacerlo con todos y así no tener que usar Patchword

```
library(GGally)
dat9 %>%
  ggpairs(colums=1:4, mapping = ggplot2::aes(colour=School))
```



9.5 Comparar mediante gráficos de barras las medias de ambas escuelas.

```
dat9c=dat9 %>%
  group_by(School) %>%
  summarise(
    Composición=mean(Composition,na.rm=TRUE),
    Drawing=mean(Drawing,na.rm=TRUE),
    Color=mean(Colour,na.rm=TRUE),
    Expression=mean(Expression,na.rm=TRUE)
  )
dat9c
```

```
## # A tibble: 2 x 5
```

```
##   School       Composición Drawing Color Expression
##   <chr>              <dbl>   <dbl> <dbl>      <dbl>
## 1 Renacimiento        10.4    14.7   9          8.2
## 2 Veneciana            9.1     9.9  16.1        3.2
```

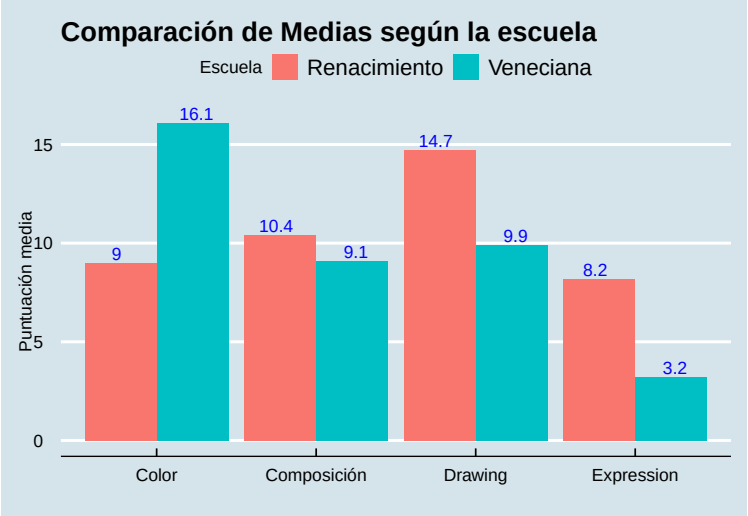
Para ggplot los necesitamos en formato largo, es decir, queremos las variables, las escuelas y los datos en columnasd.

```
dat9cLargo=
  dat9c %>%
  pivot_longer(names_to = "Variables",values_to = "Medias",
               cols=-School) # names_to  son: School , composición,...

dat9cLargo %>%
  kable(booktabs=TRUE) %>%
  kable_styling("striped")
```

School	Variables	Medias
Renacimiento	Composición	10.4
Renacimiento	Drawing	14.7
Renacimiento	Color	9.0
Renacimiento	Expression	8.2
Veneciana	Composición	9.1
Veneciana	Drawing	9.9
Veneciana	Color	16.1
Veneciana	Expression	3.2

Vamos a hacer el gráfico.

```
uno=dat9cLargo %>%
  ggplot(aes(x=Variables,y=Medias,fill=School))+
  geom_col(position="dodge")+
  geom_text(aes(label=Medias),
              size=4, hjust=0.5,vjust=-0.25,
              position=position_dodge(width = 1), color="blue"  )+
  labs(
    title="Comparación de Medias según la escuela",
    x="",
    y="Puntuación media",
    fill="Escuela"
  )+
  ggthemes::theme_economist()
dos=dat9cLargo %>%
  ggplot(aes(x=Variables,y=Medias,fill=School))+
  geom_col()
uno
```




```
uno/dos # Ver con dodge que ya no aparecen apiladas
```