

Índice

| | |
|--|-----------|
| 1. Identificación de Modelo y Muestreo | 3 |
| a) Ajuste de modelo. | 3 |
| 1. Breve descripción de las variables sleeptime y steps | 3 |
| Data\$sleeptime | 3 |
| Data\$steps | 3 |
| 2. Ajustar todos los datos de sleeptime a una Normal, una Gamma y una Exponencial. | 4 |
| Data\$sleeptime | 4 |
| Data\$steps | 5 |
| 3. Test de Kolmogorov-Smirnov: ks.test | 6 |
| Data\$sleeptime | 6 |
| Data\$steps | 6 |
| 4. Gráfica: hist, dens, dens.teórica | 7 |
| Data\$sleeptime | 7 |
| Data\$steps | 8 |
| b) Muestreo y Distribución Muestral | 9 |
| 50 muestras | 10 |
| 100 muestras | 11 |
| 3. Idem con la Varianza muestral | 12 |
| 30 muestras | 12 |
| 50 muestras | 13 |
| 100 muestras | 14 |
| 4. Idem con la proporción de mujer/varones muestral | 15 |
| 30 muestras | 15 |
| 50 muestras | 16 |
| 100 muestras | 17 |
| 2. Estimación clásica (puntual, intervalos) | 18 |
| a) Estimación puntual | 18 |
| 1. Media y varianza de las variables sleeptime y steps | 18 |
| Data\$sleeptime Data\$steps | 18 |
| 2. Media y varianza entre mujer | 18 |
| Data\$sleeptime Data\$steps | 18 |
| 3. Media y varianza entre hombres | 19 |
| Data\$sleeptime Data\$steps | 19 |
| b) Estimación por intervalos, una población | 20 |
| 1. Estimación de la media con varianza conocida | 20 |
| Data\$sleeptime (mujer) | 20 |
| Data\$sleeptime (Hombres) | 21 |
| Data\$steps (mujer) | 23 |
| Data\$steps(Hombres) | 24 |

| | |
|---|-----------|
| 2. Estimación de la media con varianza desconocida | 26 |
| Data\$sleeptime (mujer) | 26 |
| Data\$sleeptime (Hombres) | 27 |
| Data\$steps (mujer) | 29 |
| Data\$steps (Hombres) | 30 |
| 3. Proporción de mujeres y hombres. | 32 |
| 4. Diferencia de medias | 34 |
| Data\$sleeptime | 34 |
| Data\$steps | 36 |
| 5. Razón de varianzas | 38 |
| Data\$sleeptime | 38 |
| Data\$steps | 39 |
| 6. Diferencia de proporciones | 40 |
| 3. Estimación bayesiana (puntual, intervalos) | 42 |
| 4. Contrastes (paramétricos y no paramétricos) | 44 |
| Test de Pearson | 46 |
| Test de Kolmogorov-Smirnov | 46 |
| Test de Durbin-Watson | 46 |
| Test de Wilcoxon | 47 |

Índice de Figuras

| | |
|--|----|
| Histograma y boxplot de Data\$sleeptime | 3 |
| Histograma y boxplot de Data\$steps | 3 |
| Histograma y curva de la distribución normal variable Data\$sleeptime | 4 |
| Histograma y curva de la distribución gamma variable Data\$sleeptime | 4 |
| Histograma y curva de la distribución exponencial variable Data\$sleeptime | 4 |
| Histograma y función de densidad variable Data\$sleeptime | 7 |
| Histograma función de densidad variable Data\$steps | 8 |
| Histograma y boxplot 30 muestras media variable Data\$Age | 9 |
| Histograma y boxplot 50 muestras media variable Data\$Age | 10 |
| Histograma y boxplot 100 muestras media variable Data\$Age | 11 |
| Histograma y boxplot 30 muestras varianza variable Data\$Age | 12 |
| Histograma y boxplot 50 muestras varianza variable Data\$Age | 13 |
| Histograma y boxplot 100 muestras varianza variable Data\$Age | 14 |
| Histograma y boxplot para proporción de mujeres y hombre 30 muestras | 15 |
| Histograma y boxplot para proporción de mujeres y hombre 50 muestras | 16 |
| Histograma y boxplot para proporción de mujeres y hombre 100 muestras | 17 |
| Comparación de función a priori, verosimilitud y a posteriori | 42 |

1. Identificación de Modelo y Muestreo

a) Ajuste de modelo.

1. Breve descripción de las variables *sleeptime* y *steps*

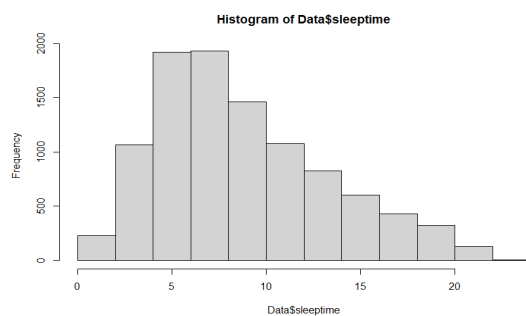
La variable ***sleeptime*** es una variable continua que recoge el tiempo de sueño expresado en horas por día.

Data\$sleeptime

`summary(Data$sleeptime)`

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|--------|---------|--------|--------|---------|---------|
| 0.6377 | 5.3843 | 7.8434 | 8.7821 | 11.5667 | 22.0813 |

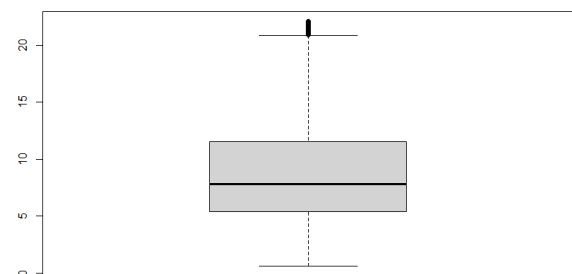
`hist(Data$sleeptime)`



`skewness(Data$sleeptime)`

[1] 0.7062983

`boxplot(Data$sleeptime)`



`kurtosis(Data$sleeptime)`

[1] -0.1803507

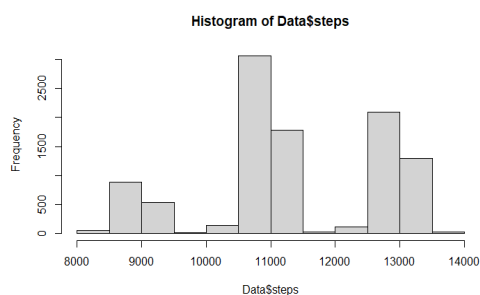
Steps es otra variable continua que indica la media de pasos dados en un intervalo de tiempo.

Data\$steps

`summary(Data$steps)`

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|------|---------|--------|-------|---------|-------|
| 8175 | 10738 | 11037 | 11332 | 12799 | 13755 |

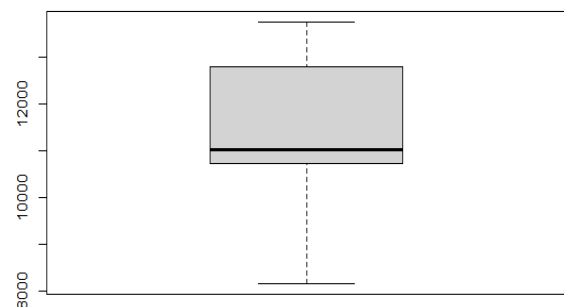
`hist(Data$steps)`



`skewness(Data$steps)`

[1] -0.2585471

`boxplot(Data$steps)`



`kurtosis(Data$steps)`

[1] -0.8003979

2. Ajustar todos los datos de *sleeptime* a una Normal, una Gamma y una Exponencial.

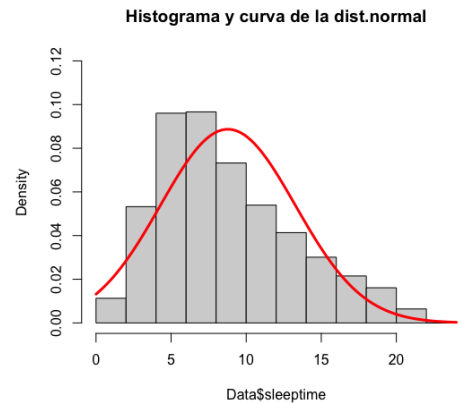
Data\$sleeptime

Ajuste de los datos de *sleeptime* a una distribución normal:

```
fitdistr(Data$sleeptime, c("normal"))  
      mean      sd  
8.78206610 4.49717612  
(0.04497176) (0.03179984)
```

A partir de la media y la desviación típica obtenidas tras el ajuste a la normal:

```
x=seq(0,1, 0.01)  
hist(Data$sleeptime, freq = FALSE, ylim=c(0, 0.12),  
main="Histograma y curva de la dist.normal")  
curve(dnorm(x, 8.78206610, 4.49717612, col = "red", lwd =  
3, add = TRUE)
```

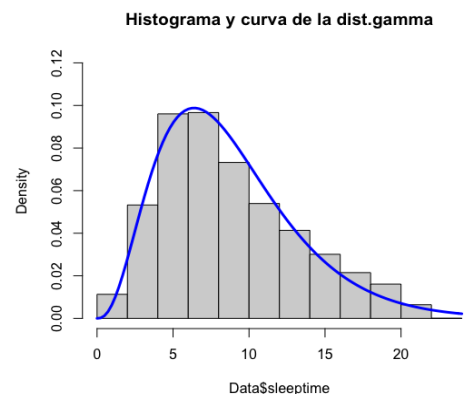


Ajuste de los datos de *sleeptime* a una distribución gamma:

```
fitdistr(Data$sleeptime, c("gamma"))  
      shape      rate  
3.660749175 0.416843756  
(0.049592063) (0.006052637)
```

Utilizando los valores *shape*=3.660749175 y *rate*=0.416843756 obtenidos tras el ajuste a gamma, obtenemos el histograma:

```
x=seq(0,1, 0.01)  
hist(Data$sleeptime, freq = FALSE, ylim=c(0, 0.12),  
main="Histograma y curva de la dist.gamma")  
curve(dgamma(x, 3.6607492, 0.4168438), col="blue", lwd=3,  
add=TRUE)
```

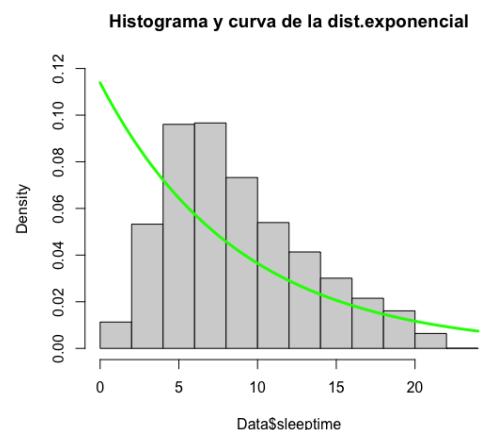


Ajuste de los datos de *sleeptime* a una distribución exponencial:

```
fitdistr(Data$sleeptime, c("exponential"))  
      rate  
0.113868421  
(0.001138684)
```

A partir del valor *rate*=0.113868421, obtenemos el histograma

```
hist(Data$sleeptime, freq = FALSE, ylim=c(0, 0.12),  
main="Histograma y curva de la dist.exponencial")  
curve(dexp(x, 0.113868421), col="blue", lwd=3, add=TRUE)
```



Data\$steps

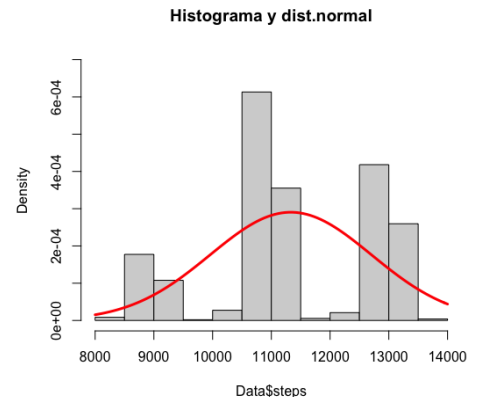
Ajuste de los datos de *steps* a una distribución normal:

```
fitdistr(Data$steps, c("normal"))
```

| mean | sd |
|--------------|-------------|
| 11331.744737 | 1373.466347 |
| (13.734663) | (9.711874) |

A partir de la media y la desviación típica obtenidas tras el ajuste a la normal:

```
hist(Data$steps, freq = FALSE, ylim=c(0, 7e-04),  
main="Histograma y dist.normal")  
curve(dnorm(x, 11331.744737, 1373.466347), col = "red",  
lwd = 3, add = TRUE)
```



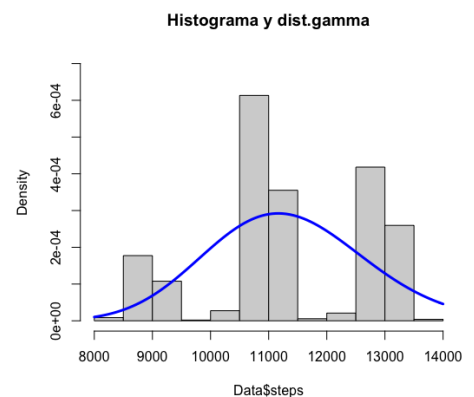
Ajuste de los datos de *steps* a una distribución gamma:

```
fitdistr(Data$steps, c("gamma"))
```

| shape | rate |
|----------------|----------------|
| 6.806330e+01 | 6.006529e-03 |
| (5.122055e-01) | (4.473506e-05) |

Utilizando los valores *shape*= 6.806330e+01 y *rate*= 6.006529e-03 obtenidos tras el ajuste a gamma, obtenemos el histograma:

```
hist(Data$steps, freq = FALSE, ylim=c(0, 7e-04),  
main="Histograma y dist.gamma")  
curve(dgamma(x, 6.806330e+01, 6.006529e-03),  
col="blue", lwd=3, add=TRUE)
```



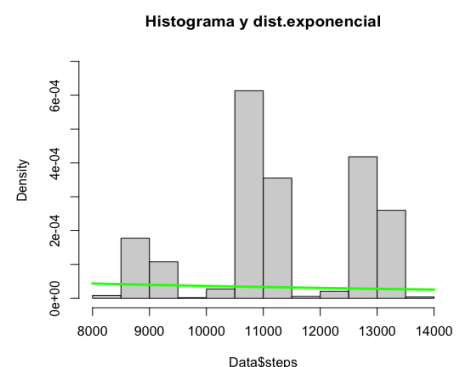
Ajuste de los datos de *steps* a una distribución exponencial:

```
fitdistr(Data$steps, c("exponential"))
```

| rate |
|----------------|
| 8.824766e-05 |
| (8.824766e-07) |

A partir del valor *rate*=8.824766e-05, obtenemos el histograma

```
hist(Data$steps, freq = FALSE, ylim=c(0, 7e-04),  
main="Histograma y dist.exponencial")  
curve(dexp(x, 8.824766e-05), col="green", lwd=3,  
add=TRUE)
```



3. Test de Kolmogorov-Smirnov: ks.test

Data\$sleeptime

```
ks.test(Data$sleeptime, pnorm, 8.78206610, 4.49717612)
```

One-sample Kolmogorov-Smirnov test

data: Data\$sleeptime
D = 0.085065, p-value < 2.2e-16
alternative hypothesis: two-sided

```
ks.test(Data$sleeptime, pgamma, 3.660749175, 0.416843756)
```

One-sample Kolmogorov-Smirnov test

data: Data\$sleeptime
D = 0.023454, p-value = 3.334e-05
alternative hypothesis: two-sided

```
ks.test(Data$sleeptime, "pexp", 0.113868421)
```

One-sample Kolmogorov-Smirnov test

data: Data\$sleeptime
D = 0.24804, p-value < 2.2e-16
alternative hypothesis: two-sided

Como $p < 0.1$, en todos los casos del test de Kolmogorov la variable *sleeptime* no siguen una distribución normal, gamma, ni exponencial.

Data\$steps

```
ks.test(Data$steps, "pnorm", 11331.744737, 1373.466347)
```

One-sample Kolmogorov-Smirnov test

data: Data\$steps
D = 0.14886, p-value < 2.2e-16
alternative hypothesis: two-sided

```
ks.test(Data$steps, "pgamma", 6.806330e+01, 6.006529e-03)
```

One-sample Kolmogorov-Smirnov test

data: Data\$steps
D = 0.15125, p-value < 2.2e-16
alternative hypothesis: two-sided

```
ks.test(Data$steps, "pexp", 8.824766e-05)
```

One-sample Kolmogorov-Smirnov test

data: Data\$steps

D = 0.52377, p-value < 2.2e-16

alternative hypothesis: two-sided

Como $p < 0.1$, en todos los casos del test de Kolmogorov la variable *steps* no siguen una distribución normal, gamma, ni exponencial.

4. Gráfica: hist, dens, dens.teórica

Data\$sleeptime

```
density(Data$sleeptime)
```

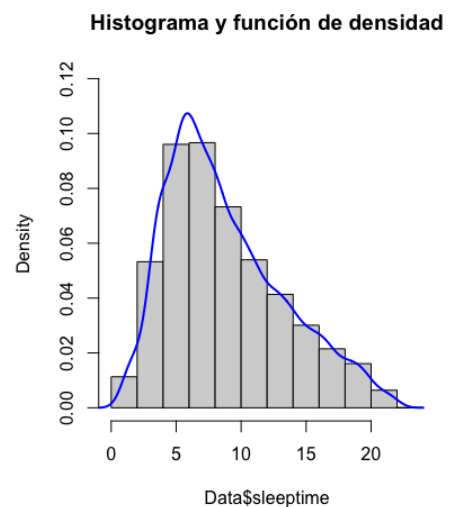
Call:

```
density.default(x = Data$sleeptime)
```

Data: Data\$sleeptime (10000 obs.); Bandwidth 'bw' = 0.6415

| x | y |
|----------------|-------------------|
| Min. :-1.287 | Min. :6.280e-06 |
| 1st Qu.: 5.036 | 1st Qu.:1.078e-02 |
| Median :11.359 | Median :2.994e-02 |
| Mean :11.359 | Mean :3.950e-02 |
| 3rd Qu.:17.683 | 3rd Qu.:6.578e-02 |
| Max. :24.006 | Max. :1.073e-01 |

```
hist(Data$sleeptime, freq = FALSE, ylim=c(0, 0.12),  
main="Histograma y función de densidad")  
dx <- density(Data$sleeptime)  
lines(dx, lwd=2, col="blue")
```



Data\$steps

`density(Data$steps)`

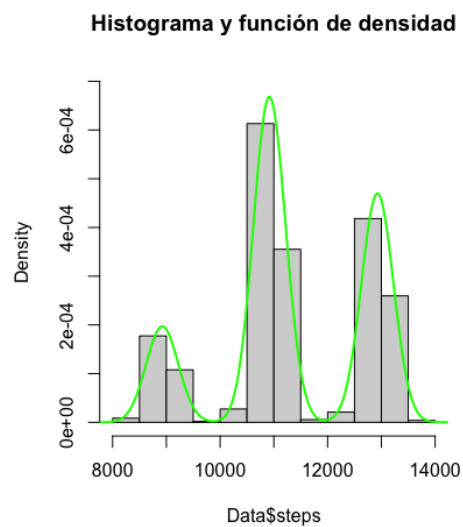
Call:

`density.default(x = Data$steps)`

Data: Data\$steps (10000 obs.); Bandwidth 'bw' = 195.9

| x | y |
|----------------|--------------------|
| Min. : 7587 | Min. : 5.000e-09 |
| 1st Qu.: 9276 | 1st Qu.: 7.213e-06 |
| Median : 10965 | Median : 6.446e-05 |
| Mean : 10965 | Mean : 1.479e-04 |
| 3rd Qu.: 12654 | 3rd Qu.: 2.056e-04 |
| Max. : 14343 | Max. : 6.682e-04 |

```
hist(Data$steps, freq = FALSE, ylim=c(0, 7e-04),  
main="Histograma y función de densidad")  
dx2<-density(Data$steps)  
lines(dx2, lwd=2, col="green")
```



b) Muestreo y Distribución Muestral

1 y 2. Se toman muestras de tamaño 200 para Age. Con 30, 50 y 100 muestras de tamaño 200 de Data, calcular las medias y representar hist y boxplot, ajustar a la distribución normal

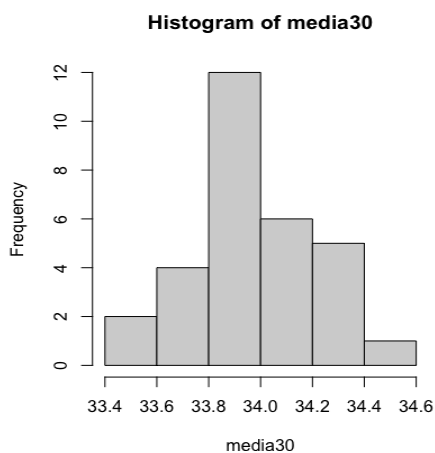
30 muestras

```
muestras30<-replicate(30, sample(Data$Age, 200))
```

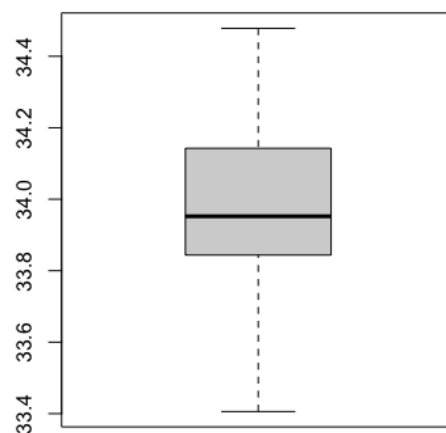
```
media30<-colMeans(muestras30)
```

```
[1] 34.07003 33.68843 33.81088 34.21099 34.40191 34.21802 33.82906 34.00219  
[9] 33.97317 34.27637 33.97789 33.43068 33.70236 33.97898 34.08542 33.88725  
[17] 34.28612 34.06937 34.51882 33.97965 33.98442 33.91211 33.49248 34.07214  
[25] 33.78563 34.05592 34.11158 34.44845 34.46972 33.47787
```

```
hist(media30)
```

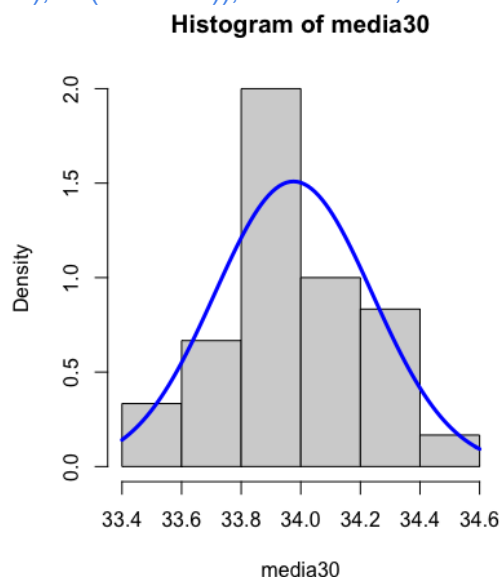


```
boxplot(media30)
```



```
hist(media30, freq=FALSE)
```

```
curve(dnorm(x, mean(media30), sd(media30)), col = "blue", lwd = 3, add = TRUE)
```



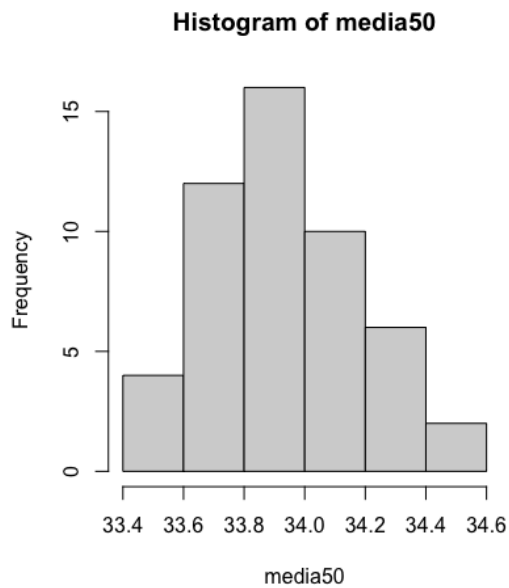
50 muestras

```
muestras50<-replicate(50, sample(Data$Age, 200))
```

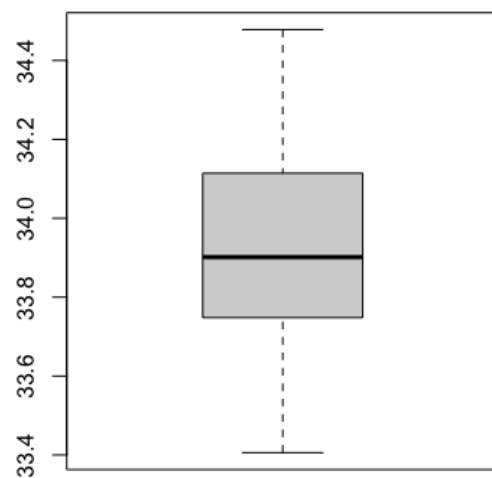
```
media50<-colMeans(muestras50)
```

```
[1] 33.85210 34.46560 33.53608 33.93677 34.33370 34.14514 34.25529 33.95513  
[9] 34.11465 33.86675 33.62834 33.88250 34.20426 33.72042 33.77700 33.67555  
[17] 33.89903 34.29230 34.28557 33.92803 34.19230 34.39449 34.47384 34.44356  
[25] 33.68390 33.90295 33.96609 34.15097 33.51190 34.13338 33.62475 33.94404  
[33] 34.13747 33.56012 34.30135 33.73178 34.02424 34.04907 33.98648 34.00304  
[41] 33.87431 34.38471 34.20744 33.89002 34.11241 33.69424 33.90131 33.82399  
[49] 33.88001 34.17112
```

```
hist(media50)
```

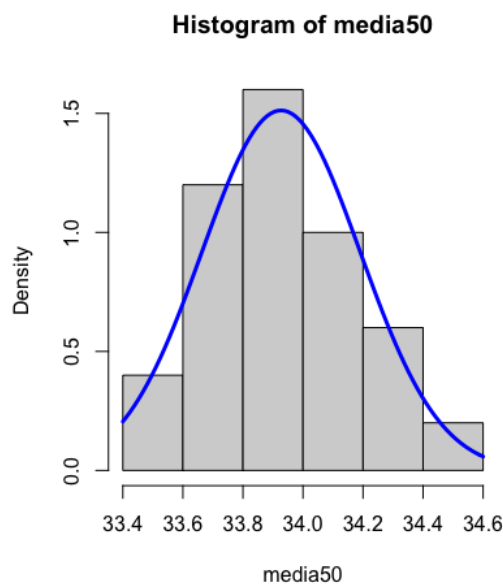


```
boxplot(media50)
```



```
hist(media50, freq=FALSE)
```

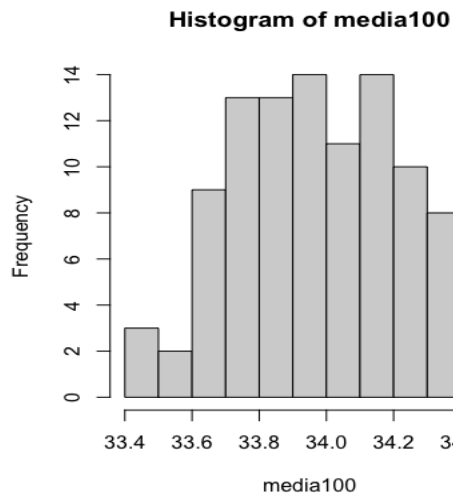
```
> curve(dnorm(x, mean(media50), sd(media50)), col = "blue", lwd = 3, add = TRUE)
```



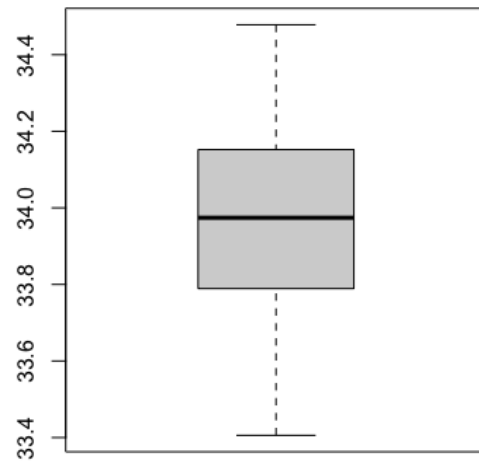
100 muestras

```
muestras100<-replicate(100, sample(Data$Age, 200))  
media100<-colMeans(muestras100)
```

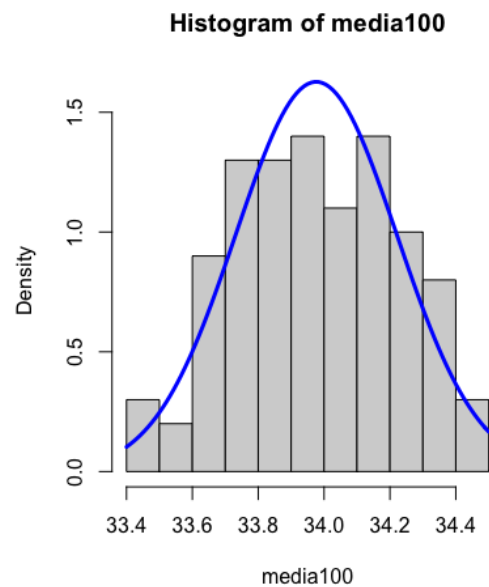
```
hist(media100)
```



```
boxplot(media100)
```



```
hist(media100, freq=FALSE, ylim=c(0, 1.5))  
curve(dnorm(x, mean(media100), sd(media100)), col = "blue", lwd = 3, add = TRUE)
```



Podemos observar que a medida que aumentamos el número de muestras, siendo en este caso máximo con 100 muestras, se aproximan más los valores a la media.

3. Idem con la Varianza muestral

Para hallar la varianza muestral. creamos una variable a la que le asignamos con un bucle las varianzas de la muestra 30 definida en el apartado anterior y la columna i

30 muestras

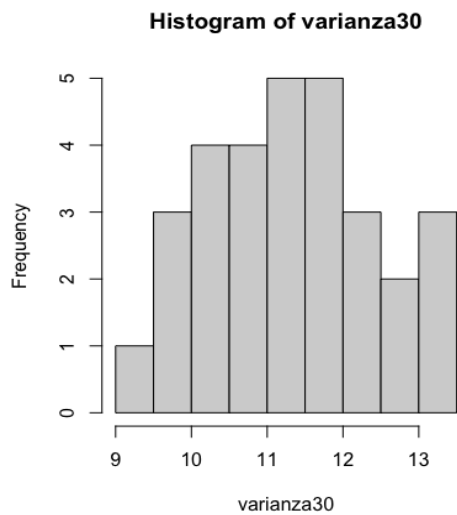
```
varianza30=1:30
```

```
for (i in 1:30) {varianza30[i]<-var(muestras30[1:200, i])}
```

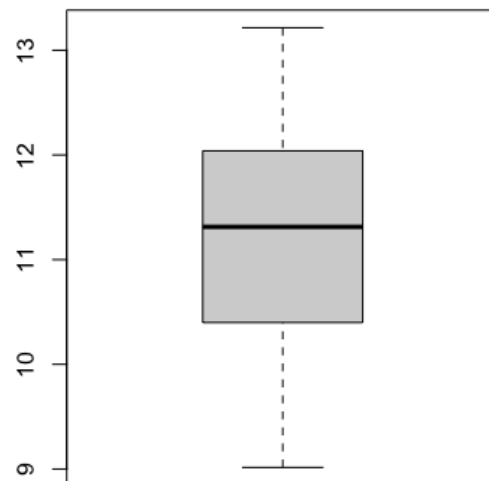
```
varianza30
```

```
[1] 11.120646 11.231837 11.215366 10.361570 11.019752 11.213799 11.240069 10.993101  
[9] 11.897978 11.069569 10.966613 10.327235 11.218188 9.682021 11.958018 11.574658  
[17] 10.555628 11.024767 13.070704 13.019740 12.980848 11.618010 10.378967 11.306521  
[25] 8.825753 11.892880 10.335009 11.736402 10.539344 11.604399
```

```
hist(varianza30)
```

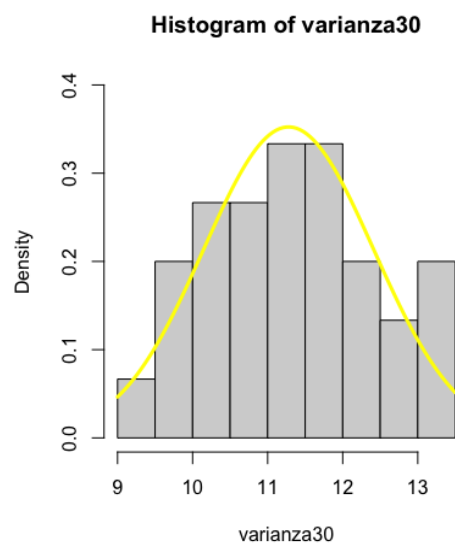


```
boxplot(varianza30)
```



```
hist(varianza30, freq=FALSE, ylim=c(0, 0.4))
```

```
curve(dnorm(x, mean(varianza30), sd(varianza30)), col = "yellow", lwd = 3, add = TRUE)
```



50 muestras

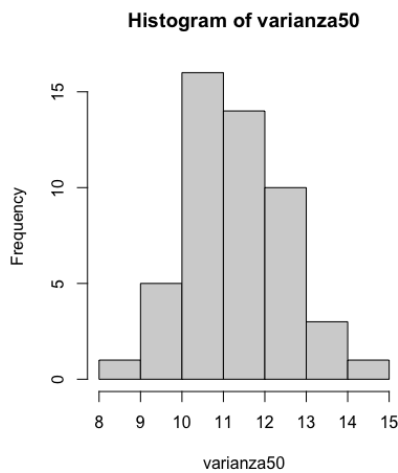
```
varianza50=1:50
```

```
for (i in 1:50) {varianza50[i]<-var(muestras50[1:200, i])}
```

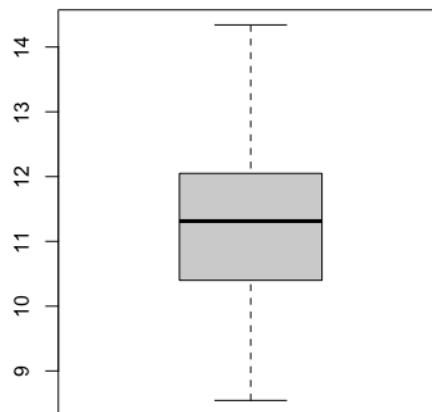
```
varianza50
```

```
[1] 9.504319 10.905797 9.505716 13.214799 9.015143 12.039078 11.314338 10.142841  
[9] 11.205700 10.674241 11.704014 11.200867 10.219696 11.311428 10.642182 10.289245  
[17] 11.826548 12.053492 11.443056 10.940889 13.160369 12.873750 12.685597 13.011185  
[25] 11.540348 12.273256 9.719649 11.627185 11.936956 10.399005 10.376174 9.361440  
[33] 8.544857 12.416831 12.408320 12.202757 12.230879 10.942440 11.442213 10.037489  
[41] 11.629021 11.573594 10.476358 14.340556 12.049221 11.559610 10.713468 10.098248  
[49] 10.777265 10.493471
```

```
hist(varianza50)
```

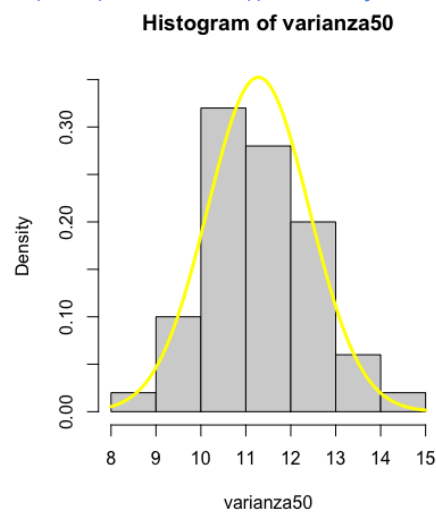


```
boxplot(varianza50)
```



```
hist(varianza50, freq=FALSE, ylim=c(0, 0.35))
```

```
curve(dnorm(x, mean(varianza50), sd(varianza50)), col = "yellow", lwd = 3, add = TRUE)
```

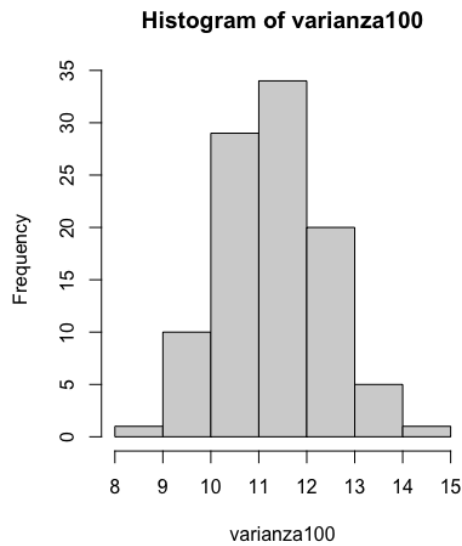


100 muestras

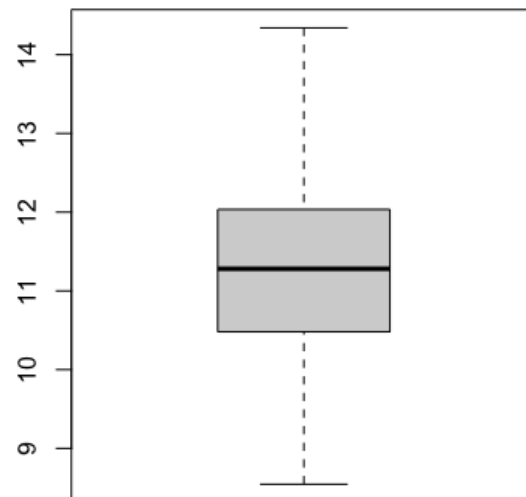
```
varianza100=1:100
```

```
for (i in 1:100) {varianza100[i]<-var(muestras100[1:200, i])}
```

```
hist(varianza100)
```

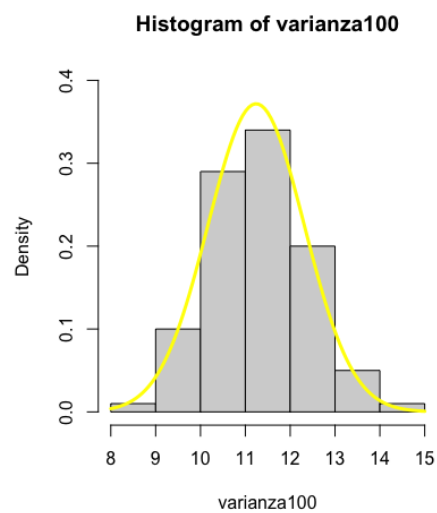


```
boxplot(varianza100)
```



```
hist(varianza100, freq=FALSE, ylim=c(0, 0.4))
```

```
curve(dnorm(x, mean(varianza100), sd(varianza100)), col = "yellow", lwd = 3, add = TRUE)
```

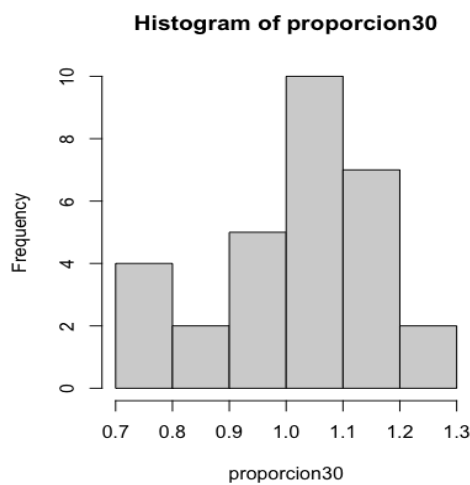


4. Idem con la proporción de mujer/varones muestral

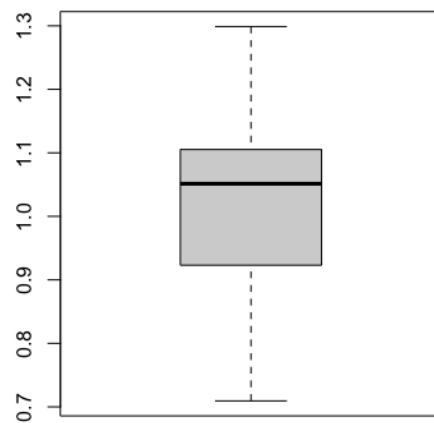
30 muestras

```
muestras30sexo<-replicate(30,sample(Data$Sex, 200))
sexo30=1:30
#Método para contar el número de varones
#Como se almacenan en vectores hacemos esto:
for (i in 1:30){sexo30[i] <- sum(muestras30sexo[(1+200*(i-1)):(200*i)]=="V")}
proporcion30=1:30
for(i in 1:30){proporcion30[i]<-sexo30[i]/(200-sexo30[i])}
```

hist(proporcion30)

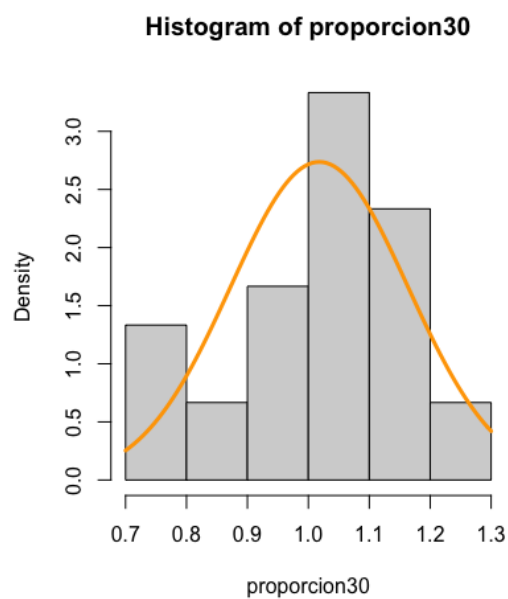


boxplot(proporcion30)



hist(proporcion30, freq=FALSE)

curve(dnorm(x, mean(proporcion30), sd(proporcion30)), col = "orange", lwd = 3, add = TRUE)

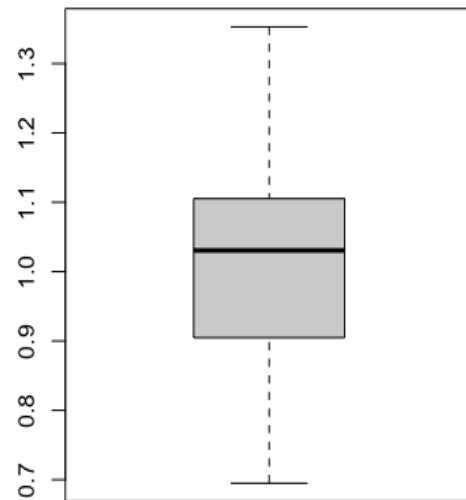
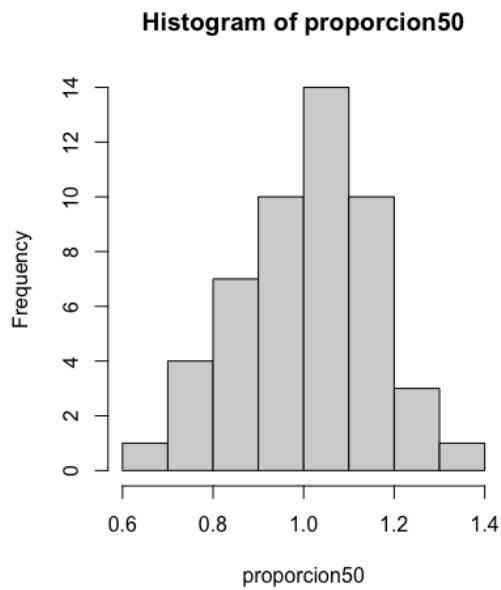


50 muestras

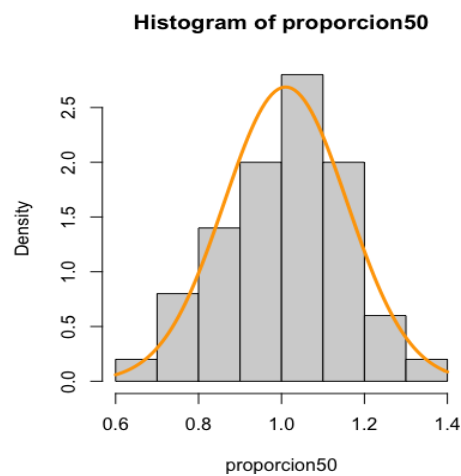
```
muestras50sexo<-replicate(50,sample(Data$Sex, 200))
sexo50=1:50
#Método para contar el número de varones
#Como se almacenan en vectores hacemos esto:
for (i in 1:50){sexo50[i] <- sum(muestras50sexo[(1+200*(i-1)):(200*i)]=="V")}
proporcion50=1:50
for(i in 1:50){proporcion50[i]<-sexo50[i]/(200-sexo50[i])}
```

hist(proporcion50)

boxplot(proporcion50)



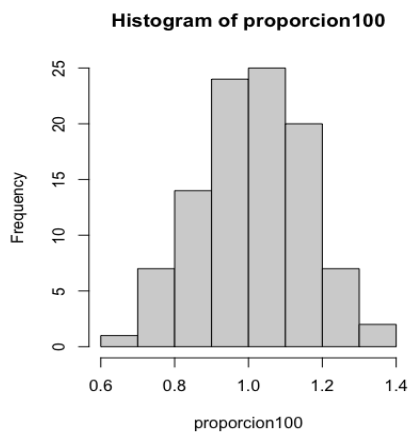
```
hist(proporcion50, freq=FALSE)
curve(dnorm(x, mean(proporcion50), sd(proporcion50)), col = "orange", lwd = 3, add = TRUE)
```



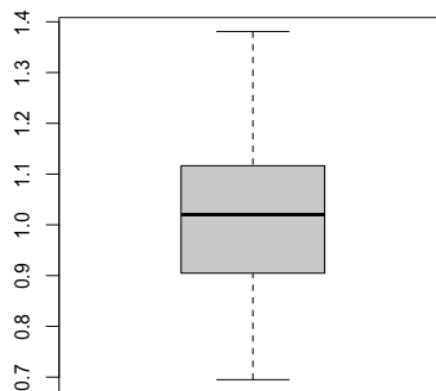
100 muestras

```
set.seed(2021)
muestras100sexo<-replicate(100,sample(Data$Sex, 200))
sexo100=1:100
for (i in 1:100){sexo100[i] <- sum(muestras100sexo[(1+200*(i-1)):(200*i)]=="V")}
proporcion100=1:100
for(i in 1:100){proporcion100[i]<-sexo100[i]/(200-sexo100[i])}
```

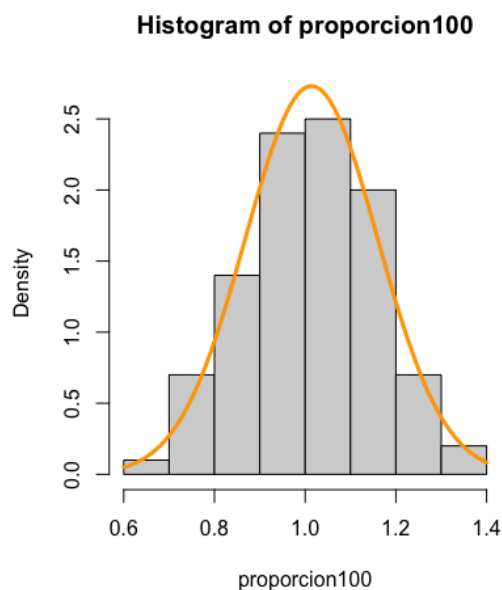
hist(proporcion100)



boxplot(proporcion100)



```
hist(proporcion100, freq=FALSE,ylim=c(0,2.7))
curve(dnorm(x, mean(proporcion100), sd(proporcion100)), col = "orange", lwd = 3, add = TRUE)
```



2. Estimación clásica (puntual, intervalos)

a) Estimación puntual

1. Media y varianza de las variables sleeptime y steps

Estimación de la media y la varianza a partir de una muestra de 200, y de los datos completos:

Data\$sleeptime

```
set.seed(2021)
muestra2<- sample(Data$sleeptime,200)
#Media de la muestra:
mean(muestra2)
[1] 8.832104
#Media de los datos completos de sleeptime
mean(Data$sleeptime)
[1] 8.782066
#Varianza de la muestra
var(muestra2)
[1] 20.94525
#Varianza de los datos completos sleeptime
var(Data$sleeptime)
[1] 20.22662
```

Data\$steps

```
set.seed(2021)
muestra3<- sample(Data$steps,200)
#Media de la muestra:
mean(muestra3)
[1] 11399.02
#Media de los datos completos de steps
mean(Data$steps)
[1] 11331.74
#Varianza de la muestra
var(muestra3)
[1] 1761013
#Varianza de los datos completos steps
var(Data$steps)
[1] 1886598
```

2. Media y varianza entre mujer

Data\$sleeptime

```
#Media entre las mujer de sleeptime
mean(Data$sleeptime[Data$Sex=="M"])
[1] 9.811281
#Varianza entre las mujer de sleeptime
var(Data$sleeptime[Data$Sex=="M"])
[1] 19.51217
```

```
set.seed(2021)
muestra_mujer_sleeptime<-
sample(Data$sleeptime[Data$Sex=="M"],200)
#Media entre las mujer de sleeptime
mean(muestra_mujer_sleeptime)
[1] 9.942984
#Varianza entre las mujer de sleeptime
var(muestra_mujer_sleeptime)
[1] 23.08107
```

Data\$steps

```
#Media entre las mujer de steps
mean(Data$steps[Data$Sex=="M"])
[1] 12336.67
#Varianza entre las mujer de steps
var(Data$steps[Data$Sex=="M"])
[1] 883412.2
```

```
set.seed(2021)
muestra_mujer_steps <-
sample(Data$steps[Data$Sex=="M"],200)
#Media entre las mujer de steps
mean(muestra_mujer_steps)
[1] 12351.12
#Varianza entre las mujer de steps
var(muestra_mujer_steps)
[1] 94081
```

3. Media y varianza entre hombres

Data\$sleeptime

```
#Media entre los varones de sleeptime
mean(Data$sleeptime[Data$Sex=="V"])
[1] 7.758189
#Varianza entre los varones de sleeptime
var(Data$sleeptime[Data$Sex=="V"])
[1] 18.83886
set.seed(2021)
muestra_varon_sleeptime<-
sample(Data$sleeptime[Data$Sex=="V"],200)
#Media entre las mujer de sleeptime
mean(muestra_varon_sleeptime)
[1] 7.537759
#Varianza entre las mujer de sleeptime
var(muestra_varon_sleeptime)
[1] 18.73971
```

Data\$steps

```
#Media entre los varones de steps
mean(Data$steps[Data$Sex=="V"])
[1] 10332.03
#Varianza entre los varones de steps
var(Data$steps[Data$Sex=="V"])
[1] 880504.
set.seed(2021)
muestra_varon_steps <-
sample(Data$steps[Data$Sex=="V"],200)
#Media entre las varones de steps
mean(muestra_varon_steps)
[1] 10312.36
#Varianza entre las varones de steps
var(muestra_varon_steps)
[1] 891499.7
```

b) Estimación por intervalos, una población

1. Estimación de la media con varianza conocida

Data\$sleeptime (mujer)

Suponiendo **normalidad** obtenemos:

#qnorm calcula el valor de z

```
CI_90<-qnorm(0.9)*var(muestra_mujer_sleeptime)/100
intervalo_90<-c(mean(muestra_mujer_sleeptime)-CI_90,
mean(muestra_mujer_sleeptime)+CI_90)
intervalo_90
[1] 9.647188 10.238780
```

```
CI_95 <- qnorm(0.95)*var(muestra_mujer_sleeptime)/100
intervalo_95<-c(mean(muestra_mujer_sleeptime)-CI_95,
mean(muestra_mujer_sleeptime)+CI_95)
intervalo_95
[1] 9.563334 10.322634
```

```
CI_99 <- qnorm(0.99)*var(muestra_mujer_sleeptime)/100
intervalo_99<-c(mean(muestra_mujer_sleeptime)-CI_99,
mean(muestra_mujer_sleeptime)+CI_99)
intervalo_99
[1] 9.406038 10.479930
```

Así para un nivel de 90% resulta el intervalo (8.522852, 9.041281); para 95%, (8.449368 9.114764); y para 99% (8.311525 9.252608).

Utilizando **Bootstrap**, y definiendo la función fMedia para su estimación obtenemos;

```
fMedia<- function(datos, i){
+   d<-datos[i]
+   return(mean(d))
+ }
```

```
set.seed(2021)
boot1 <- boot(muestra_mujer_sleeptime, statistic=fMedia, R=4000)
boot.ci(boot1, conf=0.9, type="norm")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 4000 bootstrap replicates
CALL :
boot.ci(boot.out = boot1, conf = 0.9, type = "norm")
```

Intervals :
Level Normal
90% (9.382, 10.506)
Calculations and Intervals on Original Scale

```
boot.ci(boot1, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 4000 bootstrap replicates
CALL :
boot.ci(boot.out = boot1, conf = 0.95, type = "norm")
Intervals :
Level Normal
95% (9.275, 10.613)
Calculations and Intervals on Original Scale

```
boot.ci(boot1, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 4000 bootstrap replicates
CALL :
boot.ci(boot.out = boot1, conf = 0.99, type = "norm")
Intervals :
Level Normal
99% (9.064, 10.824)
Calculations and Intervals on Original Scale

Los resultados obtenidos son a un nivel de 90%, (9.382, 10.506), a 95% (9.275, 10.613), y a 99% (9.064, 10.824) .

Data\$sleeptime (Hombres)

Suponiendo **normalidad** obtenemos:

```
CI_90<-qnorm(0.9)*var(muestra_varon_sleeptime)/100  
intervalo_90<-c(mean(muestra_varon_sleeptime)-CI_90,  
mean(muestra_varon_sleeptime)+CI_90)  
intervalo_90  
[1] 7.297599 7.777918  
CI_95 <- qnorm(0.95)*var(muestra_varon_sleeptime)/100  
intervalo_95<-c(mean(muestra_varon_sleeptime)-CI_95,  
mean(muestra_varon_sleeptime)+CI_95)  
intervalo_95  
[1] 7.229518 7.845999
```

```
CI_99 <- qnorm(0.99)*var(muestra_varon_sleeptime)/100  
intervalo_99<-c(mean(muestra_varon_sleeptime)-CI_99,  
mean(muestra_varon_sleeptime)+CI_99)  
intervalo_99  
[1] 7.101808 7.973709
```

Los intervalos solución para un nivel de confianza del 90% (7.297599 7.777918), del 95% (7.229518 7.845999), y del 99% (7.101808 7.973709)

Utilizando **Bootstrap**, y definiendo la función fMedia para su estimación obtenemos;

```
set.seed(2021)
```

```
boot2 <- boot(muestra_varon_sleeptime, statistic=fMedia, R=4000)
```

```
boot.ci(boot3, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot3, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (7.034, 8.040)

Calculations and Intervals on Original Scale

```
boot.ci(boot3, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot3, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (6.937, 8.137)

Calculations and Intervals on Original Scale

```
boot.ci(boot3, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot3, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (6.749, 8.325)

Calculations and Intervals on Original Scale

Los intervalos obtenidos para un nivel de confianza del 90% es (7.034, 8.040) , para 95% es (6.937, 8.137) y para 99% (6.749, 8.325) .

Data\$steps (mujer)

Suponiendo **normalidad** obtenemos;

```
CI_90<-qnorm(0.9)*var(muestra_mujer_steps)/100
intervalo_90<-c(mean(muestra_mujer_steps)-CI_90, mean(muestra_mujer_steps)+CI_90)
intervalo_90
[1] 294.0681 24408.1722
```

```
CI_95 <- qnorm(0.95)*var(muestra_mujer_steps)/100
intervalo_95<-c(mean(muestra_mujer_steps)-CI_95, mean(muestra_mujer_steps)+CI_95)
intervalo_95
[1] -3123.939 27826.179
```

```
CI_99 <- qnorm(0.99)*var(muestra_mujer_steps)/100
intervalo_99<-c(mean(muestra_mujer_steps)-CI_99, mean(muestra_mujer_steps)+CI_99)
intervalo_99
[1] -9535.551 34237.791
```

Los intervalos solución para 90% es (294.0681 24408.1722), para 95% (-3123.939 27826.179) y al 99% (-9535.551 34237.791)

Utilizando **Bootstrap**, y la función fMedia definida anteriormente, para su estimación obtenemos;

```
fMedia<- function(datos, i){
+   d<-datos[i]
+   return(mean(d))
+ }
```

```
set.seed(2021)
boot3 <- boot(muestra_mujer_steps, statistic=fMedia, R=4000)
boot.ci(boot2, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (12238, 12461)

Calculations and Intervals on Original Scale

```
boot.ci(boot2, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (12217, 12483)

Calculations and Intervals on Original Scale

```
boot.ci(boot2, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (12175, 12524)

Calculations and Intervals on Original Scale

Finalmente obtenemos los intervalos, para 90% (12238, 12461), 95% (12217, 12483) y para 99% (12175, 12524).

Data\$steps(Hombres)

Suponiendo **normalidad** obtenemos;

```
CI_90<-qnorm(0.9)*var(muestra_varon_steps)/100
```

```
intervalo_90<-c(mean(muestra_varon_steps)-CI_90, mean(muestra_varon_steps)+CI_90)
```

```
intervalo_90
```

```
[1] -1112.672 21737.384
```

```
CI_95 <- qnorm(0.95)*var(muestra_varon_steps)/100
```

```
intervalo_95<-c(mean(muestra_varon_steps)-CI_95, mean(muestra_varon_steps)+CI_95)
```

```
intervalo_95
```

```
[1] -4351.509 24976.221
```

```
CI_99 <- qnorm(0.99)*var(muestra_varon_steps)/100
```

```
intervalo_99<-c(mean(muestra_varon_steps)-CI_99, mean(muestra_varon_steps)+CI_99)
```

```
intervalo_99
```

```
[1] -10427.03 31051.74
```

Obtenemos los siguientes intervalos para 90% (-1112.672 21737.384), en 95% (-4351.509 24976.221) y para 99% (-10427.03 31051.74)

Utilizando **Bootstrap**, y la función `fMedia` definida anteriormente, para su estimación obtenemos;

```
set.seed(2021)
```

```
boot4 <- boot(muestra_varon_steps, statistic=fMedia, R=4000)
```

```
boot.ci(boot4, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot4, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (10200, 10424)

Calculations and Intervals on Original Scale

```
boot.ci(boot4, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot4, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (10179, 10445)

Calculations and Intervals on Original Scale

```
boot.ci(boot4, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot4, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (10137, 10487)

Calculations and Intervals on Original Scale

De esta forma, resultan los intervalos del 90% (10200, 10424) , para 95% (10179, 10445) y 99% (10137, 10487) .

2. Estimación de la media con varianza desconocida

Data\$sleeptime (mujer)

Suponiendo **normalidad** y a partir del `t.test()`, obtenemos;

```
t.test(muestra_mujer_sleeptime, conf.level = 0.90)
```

One Sample t-test

data: muestra_mujer_sleeptime

t = 29.269, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

90 percent confidence interval:

9.381591 10.504377

sample estimates:

mean of x

9.942984

```
t.test(muestra_mujer_sleeptime, conf.level = 0.95)
```

One Sample t-test

data: muestra_mujer_sleeptime

t = 29.269, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

9.273083 10.612884

sample estimates:

mean of x

9.942984

```
t.test(muestra_mujer_sleeptime, conf.level = 0.99)
```

One Sample t-test

data: muestra_mujer_sleeptime

t = 29.269, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

99 percent confidence interval:

9.05947 10.82650

sample estimates:

mean of x

9.942984

Los intervalos obtenidos son para el 90% (9.381591 10.504377), 95% resulta (9.273083 10.612884), y 99% (9.05947 10.82650)

Utilizando **Bootstrap**, y definimos `fVar`, para su estimación obtenemos;

```
fVar<- function(datos, i){
```

```
+   d<-datos[i]
```

```
+   return(var(d))
```

```
+ }
```

```
set.seed(2021)
```

```
boot1 <- boot(muestra_mujer_sleeptime, statistic=fVar, R=4000)
```

```
boot.ci(boot1, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot1, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (19.86, 26.61)

Calculations and Intervals on Original Scale

```
boot.ci(boot1, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot1, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (19.21, 27.25)

Calculations and Intervals on Original Scale

```
boot.ci(boot1, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot1, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (17.95, 28.52)

Calculations and Intervals on Original Scale

Los intervalos solución obtenidos son 90% (19.86, 26.61) , 95% (19.21, 27.25) y 99% (17.95, 28.52) .

Data\$sleeptime (Hombres)

Suponiendo **normalidad** y a partir del t.test(), obtenemos;

```
t.test(muestra_varon_sleeptime, conf.level = 0.90)
```

One Sample t-test

data: muestra_varon_sleeptime

t = 24.625, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

90 percent confidence interval:

7.031910 8.043607

sample estimates:

mean of x

7.537759

```
t.test(muestra_varon_sleeptime, conf.level = 0.95)
```

One Sample t-test

data: muestra_varon_sleeptime

t = 24.625, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

6.934138 8.141379

sample estimates:

mean of x

7.537759

```
t.test(muestra_varon_sleeptime, conf.level = 0.99)
```

One Sample t-test

data: muestra_varon_sleeptime

t = 24.625, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

99 percent confidence interval:

6.741660 8.333857

sample estimates:

mean of x

7.537759

Resultan finalmente el intervalo para 90% (7.031910 8.043607), para 95% (6.934138 8.141379) y 99% (6.741660 8.333857)

Utilizando **Bootstrap**, y definimos fVar, para su estimación obtenemos;

```
set.seed(2021)
```

```
boot2<-boot(muestra_varon_sleeptime, statistic = fVar, R=4000)
```

```
boot.ci(boot2, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (15.37, 22.18)

Calculations and Intervals on Original Scale

```
boot.ci(boot2, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (14.71, 22.83)

Calculations and Intervals on Original Scale

```
boot.ci(boot2, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (13.44, 24.11)

Calculations and Intervals on Original Scale

Finalmente resultan los intervalos para 90% (15.37, 22.18) , 95% (14.71, 22.83) y para 99% (13.44, 24.11) .

Data\$steps (mujer)

Suponiendo **normalidad** y a partir del t.test(), obtenemos;

```
t.test(muestra_mujer_steps, conf.level = 0.90)
```

One Sample t-test

data: muestra_mujer_steps

t = 180.08, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

90 percent confidence interval:

12237.78 12464.46

sample estimates:

mean of x

12351.12

```
t.test(muestra_mujer_steps, conf.level = 0.95)
```

One Sample t-test

data: muestra_mujer_steps

t = 180.08, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

12215.87 12486.37

sample estimates:

mean of x

12351.12

```
t.test(muestra_mujer_steps, conf.level = 0.99)
```

One Sample t-test

data: muestra_mujer_steps

t = 180.08, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

99 percent confidence interval:

12172.74 12529.50

sample estimates:

mean of x

12351.12

Los intervalos para 90% (12237.78 12464.46), 95% (12215.87 12486.37), y para 99% (12172.74 12529.50).

Utilizando **Bootstrap**, y definimos fVar, para su estimación obtenemos;

```
set.seed(2021)
```

```
boot3 <- boot(muestra_mujer_steps, statistic=fVar, R=4000)
```

```
boot.ci(boot3, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot3, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (845260, 1048567)

Calculations and Intervals on Original Scale

```
boot.ci(boot3, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot3, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (825786, 1068041)

Calculations and Intervals on Original Scale

```
boot.ci(boot3, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot3, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (787725, 1106102)

Calculations and Intervals on Original Scale

Los respectivos intervalos para 90% (845260, 1048567) , 95% (825786, 1068041) , y 99% (787725, 1106102) .

Data\$steps (Hombres)

Suponiendo **normalidad** y a partir del t.test(), obtenemos;

```
t.test(muestra_varon_steps, conf.level = 0.90)
```

One Sample t-test

data: muestra_varon_steps

t = 154.46, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

90 percent confidence interval:

10202.02 10422.69

sample estimates:

mean of x

10312.36

```
t.test(muestra_varon_steps, conf.level = 0.95)
```

One Sample t-test

data: muestra_varon_steps

t = 154.46, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

10180.70 10444.01

sample estimates:

mean of x

10312.36

```
t.test(muestra_varon_steps, conf.level = 0.99)
```

One Sample t-test

data: muestra_varon_steps

t = 154.46, df = 199, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

99 percent confidence interval:

10138.72 10485.99

sample estimates:

mean of x

10312.36

Obtenemos los intervalos 90% (10202.02 10422.69), 95% (10180.70 10444.01), y 99% (10138.72 10485.99).

Utilizando **Bootstrap**, y definimos fVar, para su estimación obtenemos;

```
set.seed(2021)
```

```
boot4<-boot(muestra_varon_steps, statistic = fVar, R=4000)
```

```
boot.ci(boot4, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot4, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (797596, 994624)

Calculations and Intervals on Original Scale

```
boot.ci(boot4, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot4, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (778723, 1013497)

Calculations and Intervals on Original Scale

```
boot.ci(boot4, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot4, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (741837, 1050382)

Calculations and Intervals on Original Scale

Finalmente los intervalos para 90% (797596, 994624), 95% (778723, 1013497) y 99% (741837, 1050382).

3. Proporción de mujeres y hombres.

```
set.seed(2021)
```

```
muestra_prop<-sample(Data$Sex, 200)
```

```
nMujeres<-0
```

```
#función que cuenta el número de mujeres en una muestra de 200
```

```
for (i in 1:200){
```

```
+ if(muestra_sleeptime[i]=="M")
```

```
+   nMujeres <- nMujeres+1
```

```
+ 
```

```
+ }
```

```
prop.test(nMujeres, 200, conf.level = 0.90)
```

1-sample proportions test with continuity correction

data: nMujeres out of 200, null probability 0.5

X-squared = 0.245, df = 1, p-value = 0.6206

alternative hypothesis: true p is not equal to 0.5

90 percent confidence interval:

0.4595364 0.5799069

sample estimates:

p

0.52

```
prop.test(nMujeres, 200, conf.level = 0.95)
```

1-sample proportions test with continuity correction

data: nMujeres out of 200, null probability 0.5

X-squared = 0.245, df = 1, p-value = 0.6206

alternative hypothesis: true p is not equal to 0.5

95 percent confidence interval:

0.4485723 0.5906469

sample estimates:

p
0.52

```
prop.test(nMujeres, 200, conf.level = 0.99)
```

1-sample proportions test with continuity correction

data: nMujeres out of 200, null probability 0.5

X-squared = 0.245, df = 1, p-value = 0.6206

alternative hypothesis: true p is not equal to 0.5

99 percent confidence interval:

0.4273959 0.6112850

sample estimates:

p
0.52

```
# Para calcular el intervalo de confianza de la proporción de hombres:
```

```
nHombres<-0
```

```
# Contar el número de hombres
```

```
for (i in 1:200){
```

```
  if(muestra_sleeptime[i]=="V")
```

```
    nHombres <- nHombres+1
```

```
}
```

```
prop.test(nHombres, 200, conf.level = 0.90)
```

1-sample proportions test with continuity correction

data: nHombres out of 200, null probability 0.5

X-squared = 0.245, df = 1, p-value = 0.6206

alternative hypothesis: true p is not equal to 0.5

90 percent confidence interval:

0.4200931 0.5404636

sample estimates:

p
0.48

```
prop.test(nHombres, 200, conf.level = 0.95)
```

1-sample proportions test with continuity correction

data: nHombres out of 200, null probability 0.5

X-squared = 0.245, df = 1, p-value = 0.6206

alternative hypothesis: true p is not equal to 0.5

95 percent confidence interval:

0.4093531 0.5514277

sample estimates:

p
0.48

```
prop.test(nHombres, 200, conf.level = 0.99)
```

1-sample proportions test with continuity correction

data: nHombres out of 200, null probability 0.5

X-squared = 0.245, df = 1, p-value = 0.6206

alternative hypothesis: true p is not equal to 0.5

99 percent confidence interval:

0.3887150 0.5726041

sample estimates:

p

0.48

4. Diferencia de medias

Data\$sleeptime

```
t.test (muestra_mujer_sleeptime, muestra_varon_sleeptime, var.equal = FALSE, conf.level = 0.9)
```

Welch Two Sample t-test

data: muestra_mujer_sleeptime and muestra_varon_sleeptime

t = 5.2599, df = 393.76, p-value = 2.372e-07

alternative hypothesis: true difference in means is not equal to 0

90 percent confidence interval:

1.651295 3.159156

sample estimates:

mean of x mean of y

9.942984 7.537759

```
t.test (muestra_mujer_sleeptime, muestra_varon_sleeptime, var.equal = FALSE, conf.level = 0.95)
```

Welch Two Sample t-test

data: muestra_mujer_sleeptime and muestra_varon_sleeptime

t = 5.2599, df = 393.76, p-value = 2.372e-07

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

1.506212 3.304239

sample estimates:

mean of x mean of y

9.942984 7.537759

```
t.test (muestra_mujer_sleeptime, muestra_varon_sleeptime, var.equal = FALSE, conf.level = 0.99)
```

Welch Two Sample t-test

```
data: muestra_mujer_sleeptime and muestra_varon_sleeptime  
t = 5.2599, df = 393.76, p-value = 2.372e-07  
alternative hypothesis: true difference in means is not equal to 0  
99 percent confidence interval:  
 1.221617 3.588833  
sample estimates:  
mean of x mean of y
```

Para que en la función fMedia definida previamente pueda calcular la resta de medias hemos hecho una matriz juntando dos muestras de 200 observaciones (una de mujeres y otra de hombres) que hemos pasado como parametro y devolvemos la resta de las medias

```
matriz_muestras<-rbind(muestra_mujer_sleeptime, muestra_varon_sleeptime)
```

```
fMedia<-function(datos,i){
```

```
+   mujeres<-datos[1,i]
```

```
+   varones<-datos[2,i]
```

```
+   return(mean(mujeres)-mean(varones))
```

```
+ }
```

```
set.seed(2021)
```

```
boot1<-boot(matriz_muestras, statistic = fMedia, R = 4000)
```

```
boot.ci(boot1, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot1, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (-2.16, -0.49)

Calculations and Intervals on Original Scale

```
> boot.ci(boot1, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot1, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (-2.320, -0.331)

Calculations and Intervals on Original Scale

```
boot.ci(boot1, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot1, conf = 0.99, type = "norm")
```

Intervals :
Level Normal
99% (-2.633, -0.018)
Calculations and Intervals on Original Scale

Data\$steps

```
t.test (muestra_mujer_steps, muestra_varon_steps, var.equal = FALSE, conf.level = 0.9)
```

Welch Two Sample t-test

data: muestra_mujer_steps and muestra_varon_steps
t = 21.3, df = 397.71, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
 1880.958 2196.571
sample estimates:
mean of x mean of y
 12351.12 10312.36

```
t.test (muestra_mujer_steps, muestra_varon_steps, var.equal = FALSE, conf.level = 0.95)
```

Welch Two Sample t-test

data: muestra_mujer_steps and muestra_varon_steps
t = 21.3, df = 397.71, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1850.591 2226.937
sample estimates:
mean of x mean of y
 12351.12 10312.36

```
t.test (muestra_mujer_steps, muestra_varon_steps, var.equal = FALSE, conf.level = 0.99)
```

Welch Two Sample t-test

data: muestra_mujer_steps and muestra_varon_steps
t = 21.3, df = 397.71, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 1791.027 2286.501
sample estimates:
mean of x mean of y
 12351.12 10312.36

Para que en la función fMedia definida previamente pueda calcular la resta de medias hemos hecho una matriz juntando dos muestras de 200 observaciones (una de mujeres y otra de hombres) que hemos pasado como parametro y devolvemos la resta de las medias

```
matriz_muestras<-rbind(muestra_mujer_steps, muestra_varon_steps)
```

```
set.seed(2021)
```

```
boot2<-boot(matriz_muestras, statistic = fMedia, R = 4000)
```

```
boot.ci(boot2, conf=0.9, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.9, type = "norm")
```

Intervals :

Level Normal

90% (-28.2, 1959.2)

Calculations and Intervals on Original Scale

```
boot.ci(boot2, conf=0.95, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.95, type = "norm")
```

Intervals :

Level Normal

95% (-218.6, 2149.5)

Calculations and Intervals on Original Scale

```
boot.ci(boot2, conf=0.99, type="norm")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 4000 bootstrap replicates

CALL :

```
boot.ci(boot.out = boot2, conf = 0.99, type = "norm")
```

Intervals :

Level Normal

99% (-590.6, 2521.6)

Calculations and Intervals on Original [Scale](#)

5. Razón de varianzas

Data\$sleeptime

```
var.test(muestra_mujer_sleeptime, muestra_varon_sleeptime, conf.level = 0.9)
```

F test to compare two variances

data: muestra_mujer_sleeptime and muestra_varon_sleeptime
F = 1.2317, num df = 199, denom df = 199, p-value = 0.1425
alternative hypothesis: true ratio of variances is not equal to 1
90 percent confidence interval:
0.9749285 1.5560140
sample estimates:
ratio of variances
1.231667

```
var.test(muestra_mujer_sleeptime, muestra_varon_sleeptime, conf.level = 0.95)
```

F test to compare two variances

data: muestra_mujer_sleeptime and muestra_varon_sleeptime
F = 1.2317, num df = 199, denom df = 199, p-value = 0.1425
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.9321087 1.6274952
sample estimates:
ratio of variances
1.231667

```
var.test(muestra_mujer_sleeptime, muestra_varon_sleeptime, conf.level = 0.99)
```

F test to compare two variances

data: muestra_mujer_sleeptime and muestra_varon_sleeptime
F = 1.2317, num df = 199, denom df = 199, p-value = 0.1425
alternative hypothesis: true ratio of variances is not equal to 1
99 percent confidence interval:
0.8535934 1.7771957
sample estimates:
ratio of variances
1.231667

Data\$steps

```
var.test(muestra_mujer_steps, muestra_varon_steps, conf.level = 0.9)
```

F test to compare two variances

data: muestra_mujer_steps and muestra_varon_steps

F = 1.0553, num df = 199, denom df = 199, p-value = 0.7045

alternative hypothesis: true ratio of variances is not equal to 1

90 percent confidence interval:

0.8353404 1.3332274

sample estimates:

ratio of variances

1.055319

```
var.test(muestra_mujer_steps, muestra_varon_steps, conf.level = 0.95)
```

F test to compare two variances

data: muestra_mujer_steps and muestra_varon_steps

F = 1.0553, num df = 199, denom df = 199, p-value = 0.7045

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

0.7986514 1.3944740

sample estimates:

ratio of variances

1.055319

```
var.test(muestra_mujer_steps, muestra_varon_steps, conf.level = 0.99)
```

F test to compare two variances

data: muestra_mujer_steps and muestra_varon_steps

F = 1.0553, num df = 199, denom df = 199, p-value = 0.7045

alternative hypothesis: true ratio of variances is not equal to 1

99 percent confidence interval:

0.7313777 1.5227407

sample estimates:

ratio of variances

1.055319

6. Diferencia de proporciones

```
set.seed(2021)
muestra1sex<-sample(Data$Sex, 200)
muestra2sex<-sample(Data$Sex, 200)
n1Mujeres<-0
n2Mujeres<-0
for (i in 1:200){
  if(muestra1sex[i]=="M"){
    n1Mujeres <- n1Mujeres+1}
  if(muestra2sex[i]=="M"){
    n2Mujeres <- n2Mujeres+1}
}
```

```
prop.test(x=c(n1Mujeres, n2Mujeres), n=c(200, 200), conf.level=0.90)
```

2-sample test for equality of proportions with continuity correction

data: c(n1Mujeres, n2Mujeres) out of c(200, 200)

X-squared = 0.64002, df = 1, p-value = 0.4237

alternative hypothesis: two.sided

90 percent confidence interval:

-0.04215834 0.13215834

sample estimates:

prop 1 prop 2

0.520 0.475

```
prop.test(x=c(n1Mujeres, n2Mujeres), n=c(200, 200), conf.level=0.95)
```

2-sample test for equality of proportions with continuity correction

data: c(n1Mujeres, n2Mujeres) out of c(200, 200)

X-squared = 0.64002, df = 1, p-value = 0.4237

alternative hypothesis: two.sided

95 percent confidence interval:

-0.0578977 0.1478977

sample estimates:

prop 1 prop 2

0.520 0.475

```
prop.test(x=c(n1Mujeres, n2Mujeres), n=c(200, 200), conf.level=0.99)
```

2-sample test for equality of proportions with continuity correction

data: c(n1Mujeres, n2Mujeres) out of c(200, 200)

X-squared = 0.64002, df = 1, p-value = 0.4237

alternative hypothesis: two.sided

99 percent confidence interval:

-0.08865939 0.17865939

sample estimates:

prop 1 prop 2

0.520 0.475


```

n1Hombres<-0
n2Hombres<-0
for (i in 1:200){
  if(muestra1sex[i]=="V"){
    n1Hombres <- n1Hombres+1}
  if(muestra2sex[i]=="V"){
    n2Hombres <- n2Hombres+1}
}

```

```
prop.test(x=c(n1Hombres, n2Hombres), n=c(200, 200), conf.level=0.90)
```

2-sample test for equality of proportions with continuity correction

data: c(n1Hombres, n2Hombres) out of c(200, 200)

X-squared = 0.64002, df = 1, p-value = 0.4237

alternative hypothesis: two.sided

90 percent confidence interval:

-0.13215834 0.04215834

sample estimates:

prop 1 prop 2

0.480 0.525

```
prop.test(x=c(n1Hombres, n2Hombres), n=c(200, 200), conf.level=0.95)
```

2-sample test for equality of proportions with continuity correction

data: c(n1Hombres, n2Hombres) out of c(200, 200)

X-squared = 0.64002, df = 1, p-value = 0.4237

alternative hypothesis: two.sided

95 percent confidence interval:

-0.1478977 0.0578977

sample estimates:

prop 1 prop 2

0.480 0.525

```
prop.test(x=c(n1Hombres, n2Hombres), n=c(200, 200), conf.level=0.99)
```

2-sample test for equality of proportions with continuity correction

data: c(n1Hombres, n2Hombres) out of c(200, 200)

X-squared = 0.64002, df = 1, p-value = 0.4237

alternative hypothesis: two.sided

99 percent confidence interval:

-0.17865939 0.08865939

sample estimates:

prop 1 prop 2

0.480 0.525

3. Estimación bayesiana (puntual, intervalos)

```
library(readr)
Data <- read_csv("Desktop/Probabilidad y Estadística/R/RStudio/PYE2DataSet92.csv")

set.seed(2021)
muestra<-sample(Data$Nation, 200)
sum(muestra=="SP")
sum(muestra=="SP")/200

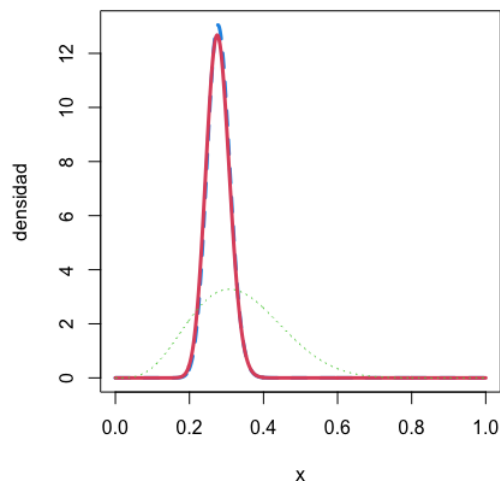
#Función a priori: distribución B(5,10)
#Función de verosimilitud: theta

x<-seq(0,1,length=500)
a=5
b=10
s=sum(muestra=="SP")
f=200-s
prior=dbeta(x,a,b)
verosimilitud=dbeta(x, s+1,f+1)
post=dbeta(x, a+s, b+f)
plot(x,post,type="l", ylab="densidad", lty=2,lwd=3,col=4) # función a posteriori en color azul
lines(x, verosimilitud, lty=1,lwd=3, col=2) # función de verosimilitud en rojo
lines(x, prior, lty=3, col=3) # función a priori en verde
```

Función a priori: $\beta(5,10)$

Función de verosimilitud: $\theta^{55}(1-\theta)^{145}$

Función a posteriori: $\theta^{59}(1-\theta)^{154} \rightarrow \beta(60,155)$



#Intervalo de confianza:

#Como nos piden el 95%, queda fuera un 5% y como tiene que ser simétrico, hallamos la qbeta de 0.025 y 0.975

```
qbeta(c(0.025, 0.975), a+s, b+f)
```

```
[1] 0.2212818 0.3407505
```

#Obtención de la muestra con franceses, italianos y españoles

```
muestra1<-sample(Data$height[Data$Nation=="FR"])
muestra2<-sample(Data$height[Data$Nation=="IT"])
muestra3<-sample(Data$height[Data$Nation=="SP"])
muestraTotal<-c(muestra1, muestra2, muestra3)
altura<-dnorm(170,7)
n=0
n0=var(muestraTotal)/49
desv=sqrt(var(muestraTotal))
n=length(muestraTotal)
desv=sqrt(var(muestraTotal))/sqrt(n+n0)
media=mean(muestraTotal)
mediaFinal=(n*media+n0*170)/(n+n0)
mediaFinal
[1] 167.9451
```

4. Contrastes (paramétricos y no paramétricos)

```
library(readr)
Data <- read_csv("Desktop/Probabilidad y Estadística/R/RStudio/PYE2DataSet92.csv")

set.seed(2021)
sample1<-sample(Data$IMC, 200)
sample2<-sample(Data$IMC, 200)

cuantil<-quantile(sample1)
t.test(sample1, alternative = "greater", mu=cuantil[2])
```

One Sample t-test

```
data: sample1
t = 8.7883, df = 199, p-value = 3.557e-16
alternative hypothesis: true mean is greater than 24.59114
95 percent confidence interval:
 25.00343      Inf
sample estimates:
mean of x
 25.09892
```

Se rechaza claramente la hipótesis nula a este nivel. Podemos afirmar media media es superior a Q_1

```
t.test(sample1, alternative="less", mu=cuantil[4])
```

One Sample t-test

```
data: sample1
t = -8.5155, df = 199, p-value = 2.021e-15
alternative hypothesis: true mean is less than 25.59093
95 percent confidence interval:
 -Inf 25.1944
sample estimates:
mean of x
 25.09892
```

Se rechaza claramente la hipótesis nula a este nivel. Podemos afirmar media es inferior a Q_3

```
sigma.test(sample1, sigmasq=1, alternative="greater")
```

One sample Chi-squared test for variance

data: sample1

X-squared = 132.87, df = 199, p-value = 0.9999

alternative hypothesis: true variance is greater than 1

95 percent confidence interval:

0.5704523 Inf

sample estimates:

var of sample1

0.6676638

Podemos asegurar dado que p-value = 0.9999 que la varianza es mayor que 1

```
t.test(sample1, sample2, paired=TRUE)
```

Paired t-test

data: sample1 and sample2

t = 1.0947, df = 199, p-value = 0.275

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-0.06973133 0.24376034

sample estimates:

mean of the differences

0.08701451

Como el p-valor $> \alpha=0.2$, no existe evidencia muestral para rechazar la igualdad de medias

```
var.test(sample1, sample2)
```

F test to compare two variances

data: sample1 and sample2

F = 1.1193, num df = 199, denom df = 199, p-value = 0.4272

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

0.8471083 1.4790815

sample estimates:

ratio of variances

1.119349

Como el p-valor $> \alpha=0.2$, no existe evidencia muestral para rechazar la igualdad de las varianzas

Test de Pearson

```
pearson.test(Sample)
```

Pearson chi-square normality test

data: Sample

P = 3.66, p-value = 0.9972

Como $p > 0.2$, no podemos rechazar la distribución normal con un nivel de significación del 5%

Test de Kolmogorov-Smirnov

```
set.seed(2021)
```

```
sample<-sample(Data$IMC, 200)
```

```
ks.test(sample, pnorm, mean(sample), sd(sample))
```

One-sample Kolmogorov-Smirnov test

data: sample

D = 0.028341, p-value = 0.9971

alternative hypothesis: two-sided

```
mean(sample)
```

25.09892

```
sd(sample)
```

0.817107

Como $p > 0.2$, no podemos rechazar la distribución normal con $\mu=25.09892$ y $\sigma=0.817107$ con un nivel de significación del 5%

Test de Durbin-Watson

```
set.seed(2021)
```

```
altura<-sample(Data$height, 200)
```

```
peso<-sample(Data$weight, 200)
```

```
set.seed(2021)
```

```
peso<-sample(Data$weight, 200)
```

```
mod1<-lm(sample ~ altura+peso)
```

```
dwtest(mod1)
```

Durbin-Watson test

data: mod1

DW = 2.1578, p-value = 0.8644

alternative hypothesis: true autocorrelation is greater than 0

Como $p > 0.2$, no podemos rechazar la hipótesis de independencia con un nivel de significación del 0.05.

Test de Wilcoxon

```
set.seed(2021)
sample1<-sample(Data$IMC, 200)
sample2<-sample(Data$IMC, 200)
wilcox.test(sample1, sample2)
```

Wilcoxon rank sum test with continuity correction

data: sample1 and sample2
W = 21080, p-value = 0.3505
alternative hypothesis: true location shift is not equal to 0

Como $p\text{-valor} > 0.2$, no podemos rechazar la hipótesis de homogeneidad con un nivel de significación del 0.05