
EJERCICIOS RESUELTOS

10.1.- Generar 100 muestras de tamaño 200 de una variable aleatoria $N(0,1)$. Calcular los correspondientes intervalos de confianza para la media al nivel del 95%, y el número de ellos que no la contienen.

SOLUCIÓN 10.1

10.2.- Generar una lista de 20 valores provenientes de una población normal de media 10 y varianza 4. Utilizando estos datos calcular un intervalo de confianza al 99% para la media de la población y otro al 95% para la varianza de la población.

SOLUCIÓN 10.2

10.3.- Para el conjunto de datos `vitcap`, perteneciente al paquete `IswR`, estimar la diferencia entre las medias de la variable `vital.capacity` cuando la variable `group` toma los valores 1 y 3. Se supone normalidad.

SOLUCIÓN 10.3

9.4.- Se quiere contrastar si el precio medio del metro cuadrado de vivienda nueva en el municipio de Getxo es 2850€ o, por el contrario, ha aumentado. Para ello se han tomado 12 viviendas al azar, obteniéndose los valores siguientes en miles de euros por metro cuadrado. Se supone normalidad.

4.01 3.87 4.68 2.83 3.88 4.92 4.46 5.64 4.91 2.35 4.12 1.11

SOLUCIÓN 10.4

SOLUCIÓN 10.1

```
> # Generaremos en primer lugar las 100 muestras;
tendremos en total 20000 valores aleatorios de una
N(0,1)
> valores.aleatorios<-rnorm(20000)

> # Organizamos los valores anteriores en una
matriz de 100 filas y 200 columnas (100 muestras
de tamaño 200)
> x<-matrix(valores.aleatorios,nrow=100)

> # Vamos a generar los vectores lim.inf y lim.sup
que inicialmente son los vectores nulos y que
representan los extremos inferiores y superiores
de los intervalos de confianza
> lim.inf<-c(rep(0,100))
> lim.sup<-c(rep(0,100))

> # Ahora, mediante un bucle, iremos asignando los
verdaderos valores de los extremos de los
intervalos de confianza
> for(i in 1:100)
+ {IC<-t.test(x[i,])$conf
+ lim.inf[i]<-IC[1];lim.sup[i]<-IC[2]}
> lim.inf
 [1] -0.168994189 -0.102966814 -0.054229619 -
0.043249995 -0.176382207
 [6] -0.088762829 -0.229321399 -0.226344429 -
0.174678971 -0.129027282
...
[96] -0.129770998 -0.029739334 -0.206517583 -
0.196698420 -0.141947694
> lim.sup
 [1] 0.129248004 0.173187224 0.208254361
0.245687977 0.117862739
 [6] 0.188612968 0.078127331 0.045632215
0.093436683 0.151957060
...
[96] 0.145744218 0.247266788 0.086298731
0.095172164 0.135060296
> IntConf<-matrix(c(lim.inf,lim.sup),nrow=100)
> IntConf
           [,1]      [,2]
 [1,] -0.168994189 0.129248004
 [2,] -0.102966814 0.173187224
 ...
 [99,] -0.196698420 0.095172164
[100,] -0.141947694 0.135060296

> plot(lim.inf,type="n",ylim=c(-0.3,0.3),xlab="
",ylab=" ")
```

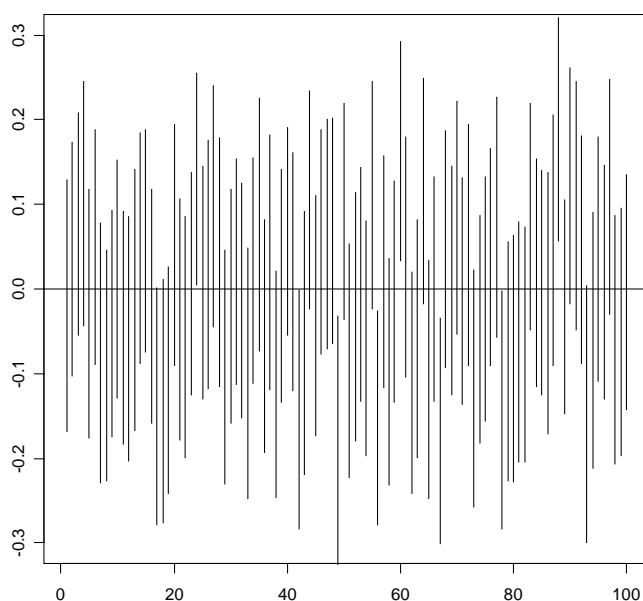
```

> # Con la opción type="n" no dibujamos el
gráfico, sólo los ejes

> # A continuación dibujamos los segmentos que
representan los diferentes IntConf
> coordx<-1:100
> segments(coordx,lim.inf,coordx,lim.sup)

> # Dibujamos la línea de la media (cero)
> abline(h=0)
> # El gráfico de los 200 IC es

```



```

> # Calculemos cuántos IntConf no contienen a la
media
> a<-c(rep(0,100))
> for (i in 1:100)
+{if (IntConf[i,1]>0 | IntConf[i,2]<0) a[i]<-1
else 0}
> a
  [1] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  [38] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
  [75] 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0
> # El número de IntConf que no contienen a la
media es:
> sum(a)
 [1] 8

> # Aviso al lector: como es lógico, los
resultados serán diferentes para cada simulación

```

SOLUCIÓN 10.2

```
> # Generamos la lista de los 20 valores
aleatorios
> valores<-rnorm(20,10,2)
> valores
 [1]  8.141324  9.271187  7.397786  9.097048
 7.567801 13.616481  9.459822
 [8]  7.459563 11.781118  8.836804 10.624325
 6.267555  6.697785  7.014521
[15]  6.641317  7.966985 12.921125 11.857177
13.766293 12.578453

> # Calculamos en primer lugar el intervalo de
confianza para la varianza al nivel 99%
> t.test(valores,conf.level=0.99)$conf
[1]  7.85318 11.04327
attr(,"conf.level")
[1] 0.99

> # Este intervalo puede ser calculado
alternativamente con la fórmula adecuada
> xraya<-mean(valores)
> destip<-sd(valores)
>
                                IC.media<-c(xraya-qt(1-
0.01/2,19)*destip/sqrt(20),xraya+qt(1-
0.01/2,19)*destip/sqrt(20))
> IC.media
[1]  7.85318 11.04327

> # Calculamos el intervalo de confianza para la
varianza al nivel 95%
> IC.varianza<-
c(9*var(valores)/qchisq(0.975,9),9*var(valores)/qc
hisq(0.025,9))
> IC.varianza
[1]  2.941223 20.719306
```

SOLUCIÓN 10.3

```
> library(ISwR)
Loading required package: survival
Loading required package: splines
Attaching package: 'ISwR'
The following object(s) are masked from
package:survival :
    lung

> data(vitcap)
> attach(vitcap)
> vitcap
```

	group	age	vital.capacity
1	1	39	4.62
2	1	40	5.29
3	1	41	5.52
4	1	41	3.71
5	1	45	4.02
6	1	49	5.09
7	1	52	2.70
8	1	47	4.31
9	1	61	2.70
10	1	65	3.03
11	1	58	2.73
12	1	59	3.67
13	3	27	5.29
14	3	25	3.67
15	3	24	5.82
16	3	32	4.77
17	3	23	5.71
18	3	25	4.47
19	3	32	4.55
20	3	18	4.61
21	3	19	5.86
22	3	26	5.20
23	3	33	4.44
24	3	27	5.52

```

> vc1<-vital.capacity[group==1]
> vc3<-vital.capacity[group==3]
> vc1
[1] 4.62 5.29 5.52 3.71 4.02 5.09 2.70 4.31 2.70
3.03 2.73 3.67
> vc3
[1] 5.29 3.67 5.82 4.77 5.71 4.47 4.55 4.61 5.86
5.20 4.44 5.52

> mean(vc1)-mean(vc3)
[1] -1.043333

> var.test(vc1,vc3)$conf
[1] 0.6651437 8.0260128
attr(,"conf.level")
[1] 0.95
> # Como el valor 1 está contenido en el anterior
intervalo de confianza podemos considerar iguales
las varianzas
> t.test(vc1,vc3,var.equal=T)$conf
[1] -1.783637 -0.303030
attr(,"conf.level")
[1] 0.95

```

SOLUCIÓN 10.4

```
>
                                                    datos<-
c(4.01,3.87,4.68,2.83,3.88,4.92,4.46,5.64,4.91,2.3
5,4.12,1.11)
> t.test(datos,mu=2.85,alternative = c("greater"))
```

One Sample t-test

```
data:  datos
t = 2.8872, df = 11, p-value = 0.00739
alternative hypothesis: true mean is greater than
2.85
95 percent confidence interval:
 3.246242      Inf
sample estimates:
mean of x
 3.898333
```

```
> #Como se trata de un contraste unilateral, en la
función t.test se ha puesto el argumento
alternative = c("greater"), ya que la hipótesis
alternativa es que el precio medio ha aumentado
```

```
> t.test(datos)$conf
[1] 3.099149 4.697517
attr(,"conf.level")
[1] 0.95
```

```
> #Conclusión: como el valor-p es muy pequeño
(menor que 0.01) se concluye que el precio medio
ha aumentado, siendo una estimación puntual de ese
precio medio 3898€ y una estimación por intervalo
[3.099149;4.697517]
```