Entrega de ejercicios Tema 3

Blanca Cano Camarero

9 de noviembre de 2022

Indice de contenidos

Ejercicio 5

```
library(ggplot2)
library(purrr)
library(nortest) # Para ejemplo del Lillie test

set.seed(100)
```

La función (ksnoest) calcula el estadístico de Kolmogorov-Smirnov cuando estamos interesados en contrastar si nuestros datos siguen una distribución normal estándar:

```
ksnoest <- function(datos){
    y <- ks.test(datos,pnorm)$statistic
    return(y)
}
```

Supongamos que queremos contrastar la hipótesis nula de que los datos son normales (con valores arbitrarios de la media y la desviación típica). Una posibilidad es estimar los parámetros de la normal y comparar la función de distribución empírica F_n con la función de distribución de una variable $N(\hat{\mu}, \hat{\sigma}^2)$. La siguiente función calcula el correspondiente estadístico de Kolmogorov-Smirnov:

```
ksest <- function(datos){
    mu <- mean(datos)
    stdev <- sd(datos)
    y <- ks.test(datos, pnorm, mean=mu, sd=stdev)$statistic
    return(y)
}
```

Apartado 1. Genera 1000 muestras de tamaño 20 y calcula ambos estadísticos (ksnoest y ksest) para cada una de ellas.

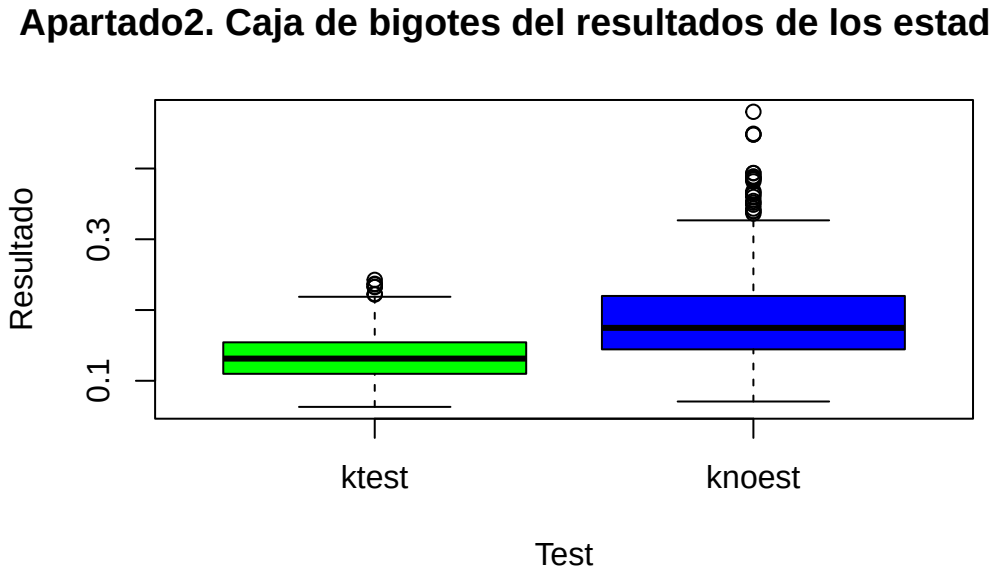
```
mean <- 0
sd <- 1
number_of_samples <- 1000
sample_size <- 20

samples <-  matrix(
   rnorm( number_of_samples * sample_size),#, mean=mean, sd= sd*),
   nrow = number_of_samples
)
```

```r
# Cálculo de ksnoest
knoest_results <- apply(X = samples,
                        MARGIN = 1, # rows 1, columns = 1
                       FUN = ksnoest
                    )
# Cálculo de ksest
kest_results <- apply(samples,1, ksest )
```

Apartado 2. Mediante diagramas de cajas, compara las distribuciones de ambos estadísticos. ¿En cuál de los dos casos se obtienen en media valores menores? ¿Podrías dar una razón intuitiva?

Solución

```r
boxplot(kest_results,knoest_results,
        main = "Apartado2. Caja de bigotes del resultados de los estadísticos",
        xlab = "Test",
        ylab = "Resultado",
        names = c("ktest", "knoest"),
        col = c("green","blue ")
)
```



En *boxplot* la media viene indicada como la línea negra central, a la vista del gráfico en ktest se obtiene una media menor.

La motivación de esto es la siguiente: de acorde a la documentación de R (ejecute `help(ks.test)`) el estadístico de test es

$$D^+ = \max_u (F_x(u) - F_y(u))$$

Que en nuestro caso se corresponde con la diferencia entre la función de distribución empírica y la de una normal.

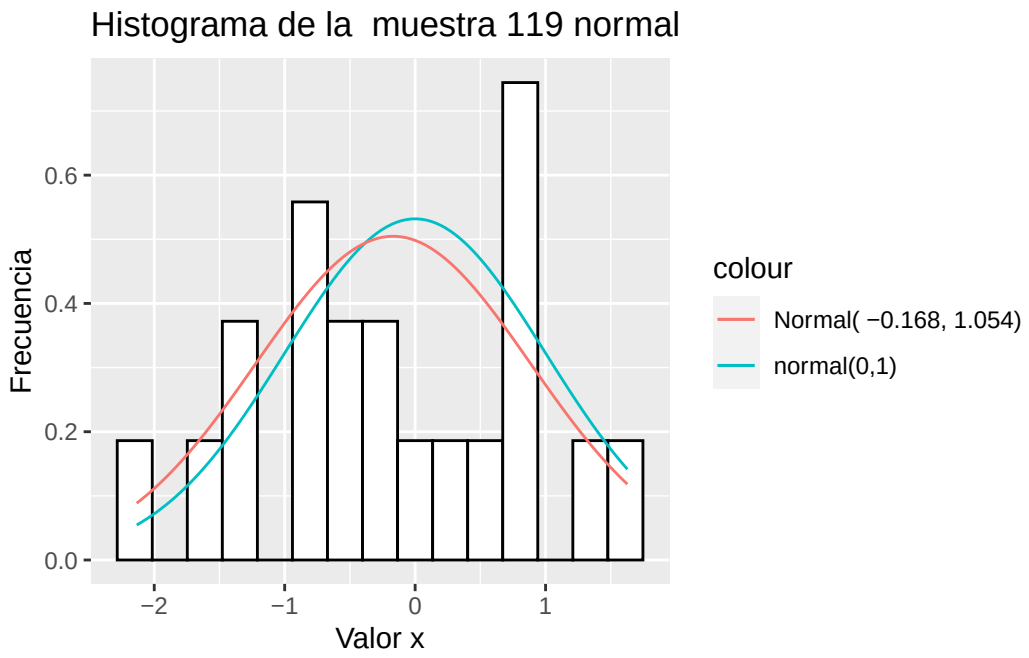
Notemos que por defecto en *knoest* se está comparando con una normal de media 0 y desviación típica 1 mientras que en *ktest*, la normal que se utiliza tiene como media la media y como desviación típica la calculada de los datos. Al ser el tamaño de muestra relativamente pequeño aunque los datos idealmente pertenecen a tal distribución.

Para ejemplificar lo dicho vamos a mostrar un histograma de los resultados junto dos las dos distribuciones normales planteadas.

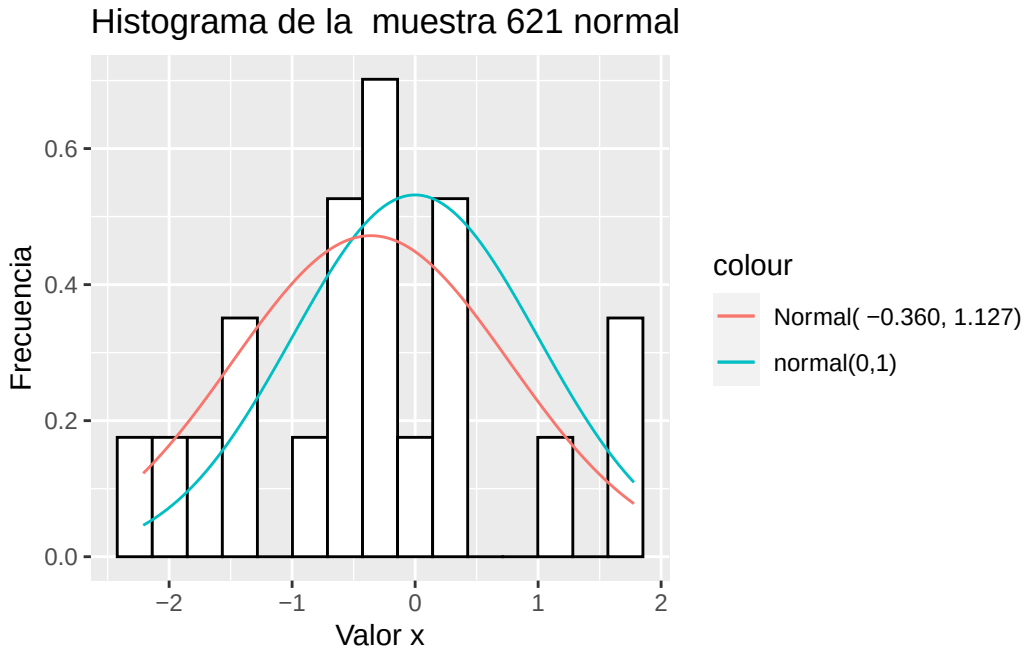
```
bins = 15

plot_hist_and_density <- function (index){

df <- data.frame(PF = samples[index,])
  scale <- sample_size/bins
  # get adapted norm
  scaled_norm <- function (x) scale*dnorm(x)
  scaled_norm_adapted <- function (x) scale*dnorm(x,mean = mean(df$PF), sd = sd(df$PF) )

  title = sprintf("Normal( %.3f, %.3f)", mean(df$PF), sd(df$PF))
  big_title =  sprintf("Histograma de la  muestra %i normal ", index)

  ggplot(df, aes(x = PF)) +
    ggtitle(big_title ) +
    xlab("Valor x") + ylab("Frecuencia") +
      geom_histogram(aes(y =..density..),
                     colour = "black",
                     fill = "white",
                     bins = bins) +
  geom_function(aes(colour = "normal(0,1)"), fun = scaled_norm ) +
  geom_function(aes(colour = title), fun = scaled_norm_adapted)

}
```

```
number_of_plots = 2
map(sample(1:number_of_samples, number_of_plots), plot_hist_and_density)
```

```
[[1]]
```




```
[[2]]
```



Apartado 3. Imagina que estimamos los parámetros y usamos las tablas de la distribución del estadístico de Kolmogorov-Smirnov para hacer el contraste a nivel α. El verdadero nivel de significación, ¿es mayor o menor que α?

Solución

El nivel de significación será por lo general menor, porque nosotros estamos suponiendo que es igual a una normal de media y distribución de parámetros estimados; que son diferentes a los ideal desconocidos.

Vamos a proceder a poner un ejemplo para los datos que ya hemos calculado, para ello tengamos presente que uno de los atributos de `ks.test` es el p-valor; la probabilidad de aceptar la hipótesis nula cuando esta es verdadera. Por otra parte se tiene que el nivel de significación α es la probabilidad de rechazar la hipótesis nula siendo verdadera.

Por lo que $\alpha = 1 - p-$valor.

Todo Dibujar las gráficas de las ditribuciones reales y compararlas

```
ksnoest_alpha <- function(datos){
    p_value <- ks.test(datos,pnorm)$p
    return(1 - p_value)
}

ksest_alpha <- function(datos){
```

```
    mu <- mean(datos)
    stdev <- sd(datos)
    p_value <- ks.test(datos, pnorm, mean=mu, sd=stdev)$p
    return(1 - p_value)
}

significacion_sin_estimar <- mean(apply(samples, 2, ksnoest_alpha))
significacion_estimada <- mean(apply(samples, 2,ksest_alpha))
sprintf('La media de nivel de signficación (para ksnoest) sin estimar los parámetros es %.
```

```
[1] "La media de nivel de signficación (para ksnoest) sin estimar los parámetros es 0.4318."
```

```
sprintf('La media de nivel de signficación para ksest es: %.4f.', significacion_estimada)
```

```
[1] "La media de nivel de signficación para ksest es: 0.2094."
```

```
veces_menor = significacion_sin_estimar / significacion_estimada * 100
sprintf('Donde para el estimado es un  %.3f por ciento menor', veces_menor)
```

```
[1] "Donde para el estimado es un  206.200 por ciento menor"
```

Podemos observar como para parámetro estimado es menor en nuestro caso.

Apartado 4. Para resolver el problema se ha estudiado la distribución en el caso de muestras normales con parámetros estimados. Es lo que se conoce como contraste de normalidad de Kolmogorov-Smirnov-Lilliefors (KSL). Según la tabla del estadístico KSL, el nivel crítico para $\alpha = 0.05$ y $n = 20$ es 0.190. Esto significa que el porcentaje de valores de ksest mayores que 0.19 en nuestra simulación debe ser aproximadamente del 5%. Compruébalo a partir de los resultados de los apartados anteriores.

Solución

```
nivel_critico <- 0.19
# Sacamos en una lista los mayores
numero_de_mayores <-  length(kest_results[kest_results > nivel_critico])
porcentaje <- 100 * numero_de_mayores / number_of_samples

cat('El porcentaje de mayores es un ', porcentaje, '%' )
```

```
El porcentaje de mayores es un  5.2 %
```

```r
# Podemos comprobarlo además haciendo

lillie.test(samples)
```

```
    Lilliefors (Kolmogorov-Smirnov) normality test

data:  samples
D = 0.0041207, p-value = 0.5647
```

Apartado 5. Haz una pequeña simulación para aproximar el nivel de significación del contraste KSL cuando se utiliza un valor crítico 0.12 para muestras de tamaño 40.

Solución

Planteamiento de la simulación. Para aproximar el nivel de significación lo que haremos será generar `number_of_samples <- 1000` muestras de tamaño `sample_size <- 40`. Calcularemos el porcentaje esas muestras se quedan por encima del valor crítico `nivel_critico <- 0.12`

```r
nivel_critico <- 0.12
number_of_samples <- 1000
sample_size <- 40

samples <-  matrix(
   rnorm( number_of_samples * sample_size),
   nrow = number_of_samples
)
kest_results <- apply(samples,1, ksest)

numero_de_mayores <-  length(kest_results[kest_results > nivel_critico])
porcentaje <- 100 * numero_de_mayores / number_of_samples

cat('El porcentaje de mayores es ', porcentaje)
```

```
El porcentaje de mayores es  13.7
```

```r
cat('\nEs decir alpha es approximadamente ', porcentaje/100)
```

```
Es decir alpha es approximadamente  0.137
```

```r
library(purrr)
library(ggplot2)
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.0      v stringr 1.4.1
v readr   2.1.2      v forcats 0.5.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
set.seed(5)
```

Ejercicio 7

```r
muestra <- c(1, 2, 3.5, 4, 7, 7.3, 8.6, 12.4, 13.8, 18.1)
varianza <- var(muestra)
```

Apartado 1. Usa bootstrap para determinar el error típico de este estimador de σ^2.

Solución Generaremos nuevas muestras a partir de las que ya tenemos, calcularemos sus varianzas y finalemente el error típico de éstas.

```r
size <- length(muestra)
number_of_samples <- 1000

# Paso 1: Remuestro de los datos
remuestreo <- matrix(
  sample(muestra, size*number_of_samples, replace=TRUE),
  nrow = number_of_samples
)
# Paso 2: Cálculo de la varianza de cada remuestreo
varianzas_remuestreo <- apply(remuestreo, 1, var)

# Paso 3: Cálculo del error típico
et <- sd(varianzas_remuestreo)

cat('El error típico de remuestreo es ', et)
```

```
El error típico de remuestreo es  10.34001
```

Apartado 2 Compara el resultado con el error típico que darías si, por ejemplo, supieras que los datos proceden de una distribución normal. **Solución** Bajo hipótesis de normalidad podría aplicarse el lema de Fisher, que dice así:

Sean $X_1, X_2, \dots, X_n$ variables aleatorias independientes e identicamente distribuidas de una normal $\mathcal{N}(\mu, \sigma^2)$. Entonces:

1. $\bar{X}$ y S^2 son independientes.
2. $$\frac{(n-1)}{\sigma^2} S^2 \sim \chi^2_{n-1}$$
3. $\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma^2}{n})$.

Puesto que nuestro objetivo es encontrar un estimador de la $Var(S^2)$ tomando la varianza de ambos miembros (2) resulta:

$$Var\left(\frac{n-1}{\sigma^2} S^2\right) = Var(\chi^2_{n-1})$$

que por las propiedades de la variaza y que $var(\chi^2_k) = 2k$ se tiene

$$\frac{(n-1)^2}{\sigma^4} Var\left(S^2\right) = 2(n-1),$$

Por lo que concluímos que

$$Var\left(S^2\right) = \frac{2\sigma^4}{n-1}.$$

Por el apartado anterior σ puede ser estimada con S, por lo que finalmente podemos concluir que

$$\hat{Vat(S^2)} = \frac{2S^4}{n-1}$$

::: {.cell}

```
# El estimador de la varianza de una normal
2*(varianza ^4)/(size - 1)
```

```
[1] 201083.5
```

::: Notemso que esto solo se puede utilizar para una normal.

Apartado 3 Calcula un intervalo de confianza para σ^2 usando el método bootstrap híbrido. Fija $1-\alpha = 0.95$.

Solución

Para explicar la idea que subyace en el diseño del algoritmo de *boostrap híbrido*, comenzaremos con las siguientes

Se define la proporción como

$$\tilde{H}_n(x) = \frac{1}{B}\sum_{b}^{B} I_{T^{*(b)} \leq x}.$$

Sea

$$H_n(x) = P_F\left(\sqrt{n}\left(\bar{X} - \mu\right) \leq x\right)$$

que por no ser conocido aproximaremos como

$$\hat{H}_n(x) = P_F\left(\sqrt{n}\left(\bar{X}^* - \bar{X}\right) \leq x\right)$$

$$1-\alpha = P\left\{H_n^{-1}\left(\frac{\alpha}{2}\right) \leq \sqrt{n}\left(\hat{\theta}-\theta\right) \leq H_n^{-1}\left(1-\frac{\alpha}{2}\right)\right\}$$

dando lugar al intervalo de confianza

$$\left[\hat{\theta} - \sqrt{n}H_n^{-1}\left(1-\frac{\alpha}{2}\right), \hat{\theta} - \sqrt{n}H_n^{-1}\left(\frac{\alpha}{2}\right)\right]$$

Puesto que H_n no es conocido los sustituiremos por el estimador de *bootstrap* $\hat{H}_n$ y es el llamado *método híbrido*.

De esta manera resulta:

```
# --- Funciones auxiliares ---
# Construción de la inversa de H(H, muestra_ordenada, B^{-1})
H_inv <- function (alpha, muestra_ordenada, B_inv, acumulado = 0, index = 0) {
  if(acumulado < alpha){
    return (H_inv(alpha, muestra_ordenada, B_inv, acumulado + B_inv, index+1 ))
  }
  else{
    return(muestra_ordenada[index])
  }
}
# En lugar de emplear esta función utilizaremos la función `quantile`
```

```r
# \hat \theta: Estimador de la varianza
```

```r
# Parámetros
a = 0.05 # alpha
B = length(muestra) # tamaño del reemuestro
numero_remuestreos = 100
repeticiones_experimento = 100

## variable auxiliares
B_inv = 1/B
acierto <- NULL
intervalo <- NULL

for(i in 1:repeticiones_experimento){

  muestras_boostrap <- matrix(
    sample(muestra, B*numero_remuestreos, rep=TRUE),
    nrow = numero_remuestreos
  )

  varianzas_bootstrap = apply(muestras_boostrap, 1, var)

  muestras_normalizadas <- sqrt(B)*(varianzas_bootstrap - varianza)


  ic_min <-varianza - quantile(muestras_normalizadas, 1-a/2)/sqrt(B)

  ic_max <-varianza - quantile(muestras_normalizadas, a/2)/sqrt(B)

  intervalo <- rbind(intervalo, c(ic_min, ic_max))
}

df <- data.frame(
  ic_min <-intervalo[, 1],
  ic_max <- intervalo[, 2],
  ind = 1:numero_remuestreos
)
df
```

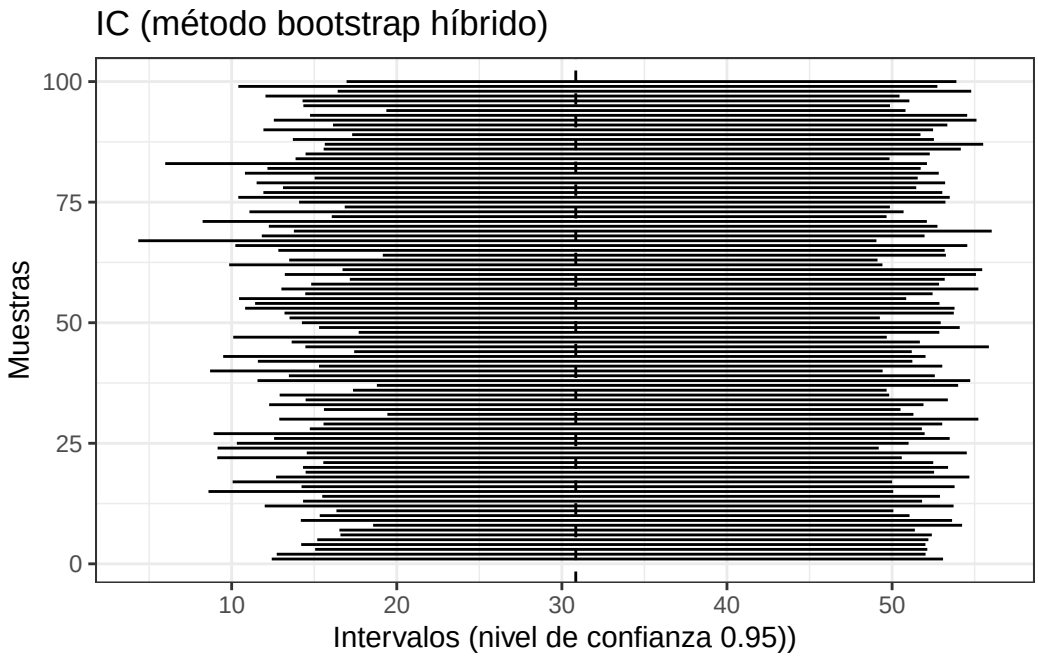
```
  ic_min....intervalo...1. ic_max....intervalo...2. ind
1                12.427733                  53.08718   1
```

2	12.731058	52.02833	2
3	15.055500	52.13480	3
4	14.210158	52.02075	4
5	15.184111	52.19874	5
6	16.588208	52.40801	6
7	16.526733	51.38136	7
8	18.568656	54.24099	8
9	14.188297	53.63517	9
10	15.334375	51.05644	10
11	16.345042	50.07880	11
12	12.002458	53.72508	12
13	14.324019	51.81609	13
14	15.485444	52.90046	14
15	8.594097	50.06843	15
16	14.231442	53.78672	16
17	10.059903	50.00227	17
18	12.688089	54.68146	18
19	14.479486	52.54397	19
20	14.322256	53.39010	20
21	15.549133	52.49265	21
22	9.125656	50.58728	22
23	14.547767	54.52394	23
24	9.147911	49.19143	24
25	10.315167	50.99920	25
26	12.569147	53.49012	26
27	8.905744	51.97293	27
28	14.733778	51.80416	28
29	15.564736	53.03680	29
30	12.883089	55.22549	30
31	19.432569	51.29322	31
32	15.588189	50.51143	32
33	12.270597	51.90271	33
34	14.478325	53.37726	34
35	12.906036	49.82532	35
36	17.350056	49.66899	36
37	18.794722	53.99940	37
38	11.565200	54.73307	38
39	13.466367	52.58790	39
40	8.692258	49.42760	40
41	15.291308	53.04028	41
42	11.587956	51.22836	42
43	9.486389	52.02636	43
44	17.419786	51.19337	44

45	14.464633	55.86329	45
46	13.633556	51.68294	46
47	10.095408	49.67898	47
48	17.692678	52.86159	48
49	15.293244	54.09589	49
50	14.254658	52.94582	50
51	13.505597	49.26109	51
52	13.201222	53.73847	52
53	10.814967	53.78312	53
54	11.419511	52.87139	54
55	10.448622	50.85506	55
56	14.455700	52.45361	56
57	13.012367	55.22641	57
58	14.815492	52.84524	58
59	17.155689	53.18893	59
60	13.218019	55.08066	60
61	16.713344	55.45980	61
62	9.848703	49.41549	62
63	13.484311	49.11637	63
64	19.154719	53.25938	64
65	12.826647	53.18517	65
66	10.222797	54.55500	66
67	4.343622	49.05241	67
68	11.824847	51.96812	68
69	13.767397	56.03463	69
70	12.250667	52.74901	70
71	8.238489	52.10568	71
72	16.060711	49.66685	72
73	11.077125	50.69639	73
74	16.847825	49.87041	74
75	14.082867	53.23271	75
76	10.401819	53.49212	76
77	11.920078	53.04233	77
78	13.107558	51.46367	78
79	11.511469	53.21249	79
80	15.025333	51.55306	80
81	10.803444	52.82646	81
82	12.165425	51.73244	82
83	5.973953	52.11164	83
84	13.869297	49.84673	84
85	14.474978	52.28304	85
86	15.571392	54.16359	86
87	15.639944	55.51732	87

88	13.707067	52.53504	88
89	17.297556	51.71985	89
90	11.922411	52.48233	90
91	16.136569	53.34643	91
92	12.552611	55.11189	92
93	14.746078	54.54900	93
94	19.369775	50.81307	94
95	14.343489	49.87388	95
96	14.294622	51.04764	96
97	12.047653	50.45321	97
98	16.420833	54.79243	98
99	10.403108	52.74234	99
100	16.959500	53.89461	100

```
ggplot(df) +
  geom_linerange(aes(xmin = ic_min, xmax = ic_max, y = ind)) +
  geom_vline(aes(xintercept = varianza), linetype = 2) +
  theme_bw() +
  labs( y= 'Muestras', x = 'Intervalos (nivel de confianza 0.95))',
        title = 'IC (método bootstrap híbrido)'
  )
```



#