

# Tarea 1 - CA0307: Estadística Actuarial II

Anthony Mauricio Jiménez Navarro | C24067      Henri Gerard Gabert Hidalgo | B93096  
Juan Pablo Morgan Sandí | C15319

2024-09-13

## Contents

<b>Setup</b>	<b>2</b>
<b>Ejercicio 1</b>	<b>2</b>
1 . . . . .	2
2 . . . . .	2
3 . . . . .	3
4 . . . . .	3
5 . . . . .	4
6 . . . . .	4
<b>Ejercicio 2</b>	<b>5</b>
1 . . . . .	5
2 . . . . .	6
3 . . . . .	12
<b>Ejercicio 3</b>	<b>13</b>
1 . . . . .	13
2 . . . . .	13
3 . . . . .	14
4 . . . . .	22
5 . . . . .	22
<b>Ejercicio 4</b>	<b>22</b>
1 . . . . .	22
2 . . . . .	23
3 . . . . .	23
<b>Referencias</b>	<b>25</b>

# Setup

Se cargan los paquetes necesarios

```
library(univariateML)
library(rriskDistributions)
library("fGarch")
library(dplyr)
library(ggplot2)
library(ks)
library(boot)
library(pander)
library(tinytex)
```

Se carga la base de datos

```
base_salarios <- read.csv("BaseSalarios.csv", header = TRUE, sep = ";",
                        stringsAsFactors = FALSE)
base_salarios <- base_salarios[ , !(names(base_salarios) %in% c("X"))]
base_salarios <- na.omit(base_salarios)

# Eliminar cualquier símbolo no numérico (excepto comas y puntos)
base_salarios$U..Salario <- gsub("[^0-9,\\.]", "", base_salarios$U..Salario)

# Eliminar espacios en blanco
base_salarios$U..Salario <- trimws(base_salarios$U..Salario)

# Reemplazar comas decimales por puntos decimales
base_salarios$U..Salario <- gsub(",", ".", base_salarios$U..Salario)

# Convertir a numérico
base_salarios$U..Salario <- as.numeric(base_salarios$U..Salario)
```

## Ejercicio 1

1

```
base_salarios$Nivel <- ifelse(base_salarios$Coutas < 150, 1, 2)
```

2

```
analisis_descriptivo <- base_salarios %>%
  group_by(Nivel) %>%
  summarise(
    Promedio_Salarios = mean(`U..Salario`, na.rm = TRUE),
    Varianza_Salarios = var(`U..Salario`, na.rm = TRUE),
    Cantidad_Salarios = n(),
    Max_Salarios = max(`U..Salario`, na.rm = TRUE),
```

```

Min_Salarios = min(`U..Salario`, na.rm = TRUE),

# Para Sexo, podemos contar la cantidad de hombres y mujeres por Nivel
Cantidad_Hombres = sum(Sexo == 1, na.rm = TRUE),
Cantidad_Mujeres = sum(Sexo == 2, na.rm = TRUE)
)

pander( analisis_descriptivo)

```

Table 1: Table continues below

Nivel	Promedio_Salarios	Varianza_Salarios	Cantidad_Salarios
1	792104	2.098e+11	3058
2	1362840	3.251e+11	2419

Max_Salarios	Min_Salarios	Cantidad_Hombres	Cantidad_Mujeres
3642975	10063	982	2076
6925643	158431	667	1752

### 3

Un boxplot es una representación gráfica que resume la distribución de un conjunto de datos, mostrando en el, los cuartiles, mínimo y máximo, siendo la caja el rango intercuartilico es decir la diferencia entre el Q3 y Q1, la línea dentro de la caja el Q2 es decir la mediana y las líneas que se alargan de las cajas, son conocidos como bigotes se extienden al máximo y mínimo, lo que esté por fuera son outliers. Por lo que los considero utiles por 3 motivos, 1. nos permiten ver más o menos una distribución de los datos, 2. es fácil ver o comparar esa distribución por grupos y 3. la fácil detección de outliers.

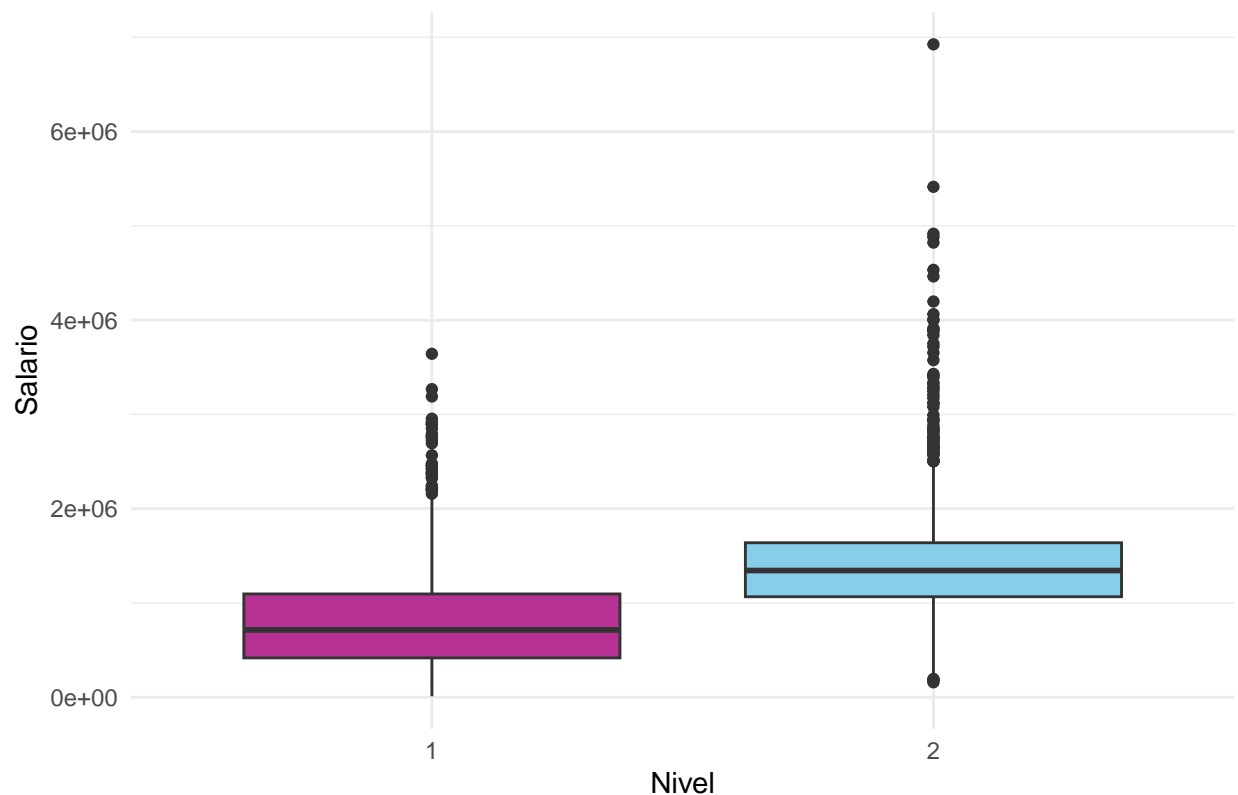
### 4

```

ggplot(base_salarios, aes(x = factor(Nivel), y = `U..Salario`, fill = factor(Nivel))) +
  geom_boxplot() +
  labs(
    x = "Nivel",
    y = "Salario",
    title = "Comparación de Salarios entre Niveles"
  ) +
  theme_minimal() +
  scale_fill_manual(values = c("#B63191", "skyblue")) +
  theme(legend.position = "none")

```

## Comparación de Salarios entre Niveles



### 5

Al observar el gráfico, me gustaría decir que entre más cuotas ha aportado, más salario se percibe, pero no es cierto. Aunque, si lo vemos desde otro punto de vista, la mediana del grupo 2 es mayor. El promedio que nos da el análisis descriptivo es más alto, y tanto el valor máximo como el mínimo son superiores en el grupo 2. Incluso, la caja de aquellos que han pagado más de 150 cuotas se ubica más arriba, salarialmente hablando, que los que han pagado menos de 150, esto significa que el 75% de las personas del nivel 2 perciben un salario mayor que el 75% de las personas del nivel 1. En general, se puede concluir que el grupo que ha aportado más cuotas tiene una percepción salarial superior.

Además, gracias al boxplot también se pueden observar varios outliers en ambos grupos, pero la cantidad es mayor en el grupo 2.

### 6

Vamos a realizar una prueba de hipótesis para comparar las medias salariales de aquellas personas que han aportado menos de 150 cuotas y de aquellos que han aportado más de 150.

```
# Filtrar los salarios según el nivel
salarios_nivel_1 <- base_salarios$U..Salario[base_salarios$Nivel == 1]
salarios_nivel_2 <- base_salarios$U..Salario[base_salarios$Nivel == 2]

# Realizar la prueba t
resultado_t_test <- t.test(salarios_nivel_1, salarios_nivel_2)
```

```
# Mostrar resultados de la prueba  
print(resultado_t_test)
```

```
##  
## Welch Two Sample t-test  
##  
## data: salarios_nivel_1 and salarios_nivel_2  
## t = -40.059, df = 4573.8, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -598667.7 -542804.2  
## sample estimates:  
## mean of x mean of y  
## 792103.6 1362839.6
```

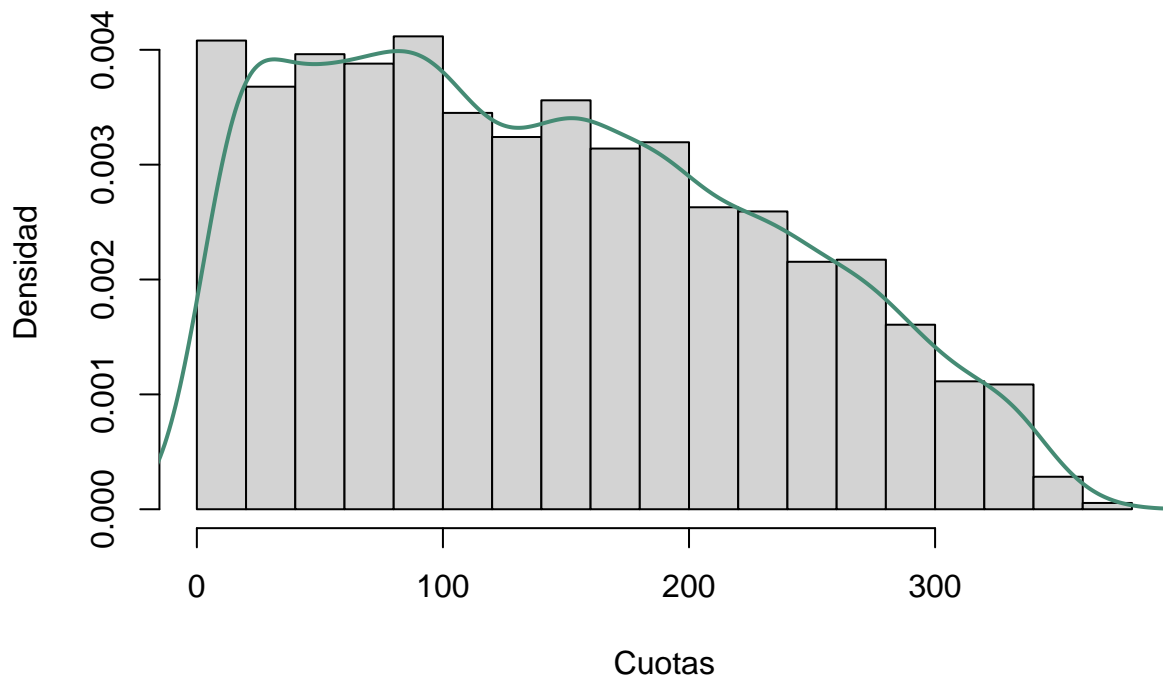
Al obtener un p-valor de  $2.2e-16$  podemos concluir que el promedio salarial de los grupos es completamente diferente.

## Ejercicio 2

1

```
# Creamos el histograma con la columna Cuotas-  
hist(base_salarios$Coutas, main = "Histograma de la Cuotas",  
      freq = F, xlab = "Cuotas", ylab = "Densidad")  
# Le añadimos la línea que representa la densidad.  
lines(density(base_salarios$Coutas), col = "aquamarine4", lwd = 2)
```

## Histograma de la Cuotas



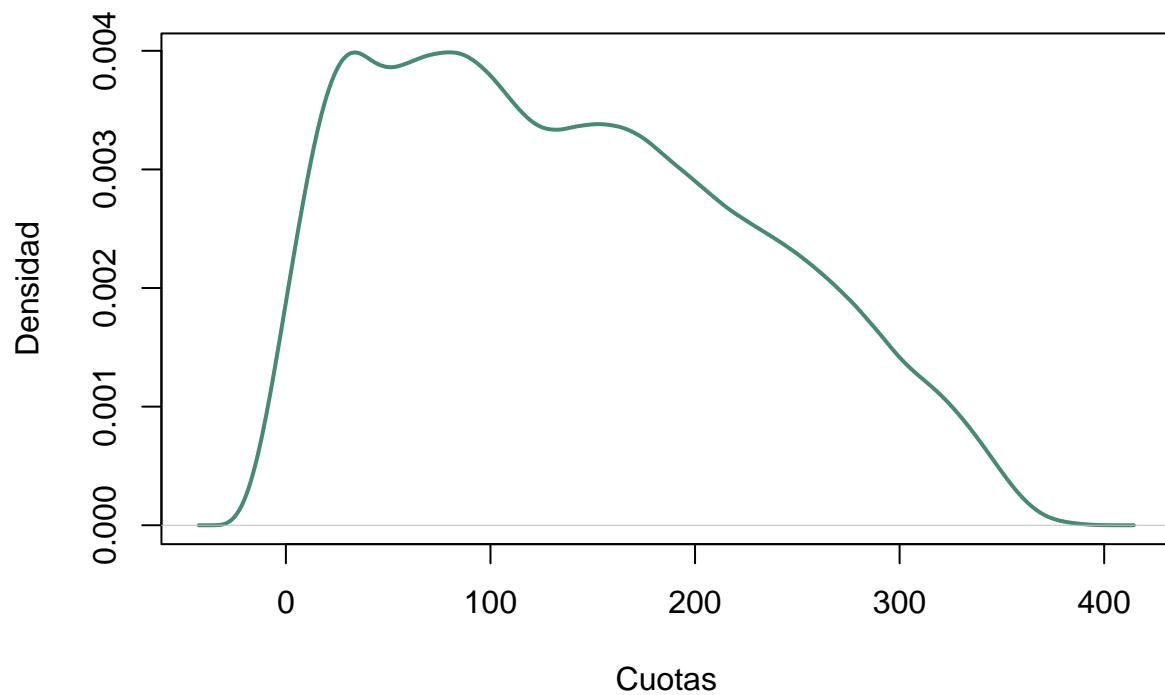
2

Ahora, graficaremos, para cada kernel estudiado, la densidad de la variable cuotas.

**Biweight**

```
# Graficamos con Kernel de Silverman.
plot(density(base_salarios$Coutas, bw = "nrd0", kernel = "biweight"),
     col = "aquamarine4", lwd = 2,
     main = "Densidad de las Cuotas según kernel Bandwidth de Silverman",
     xlab = "Cuotas", ylab = "Densidad")
```

## Densidad de las Cuotas según kernel Bandwidth de Silverman

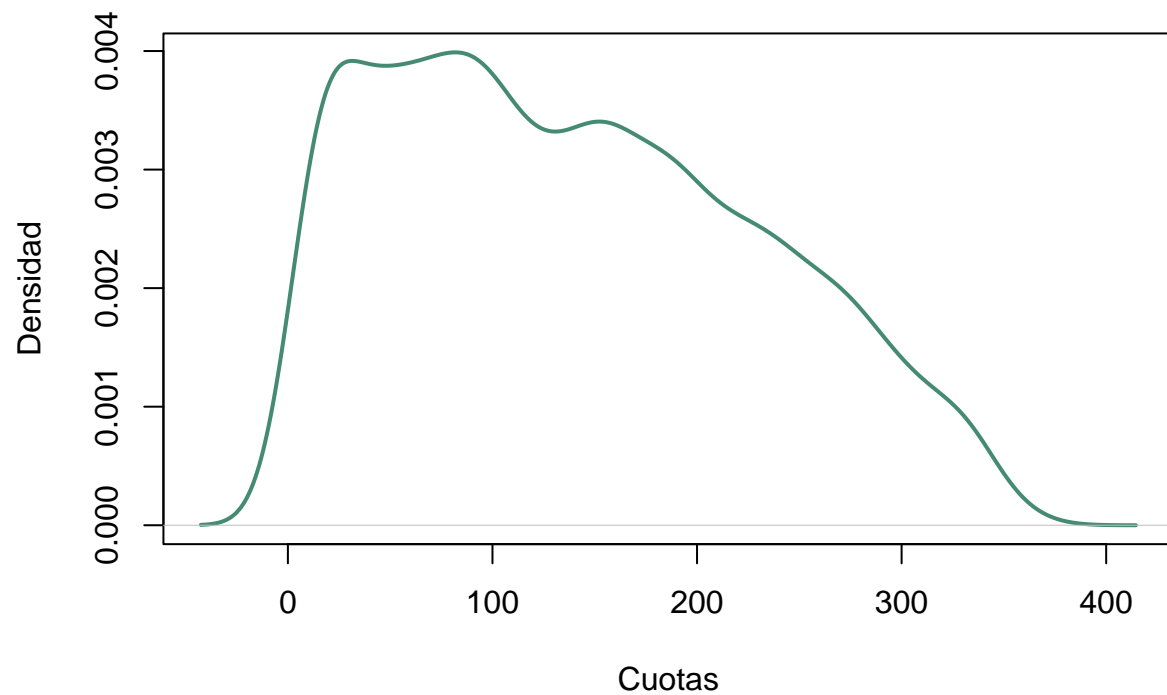


Respecto al “nrd0”, fue tomado de (Bandwidth Function - RDocumentation, s.f.)

### Gaussiana

```
# Graficamos con Kernel Gaussiano.  
plot(density(base_salarios$Coutas, kernel = "gaussian"),  
     col = "aquamarine4", lwd = 2,  
     main = "Densidad de las Cuotas según kernel Gaussiano",  
     xlab = "Cuotas", ylab = "Densidad")
```

## Densidad de las Cuotas según kernel Gaussiano

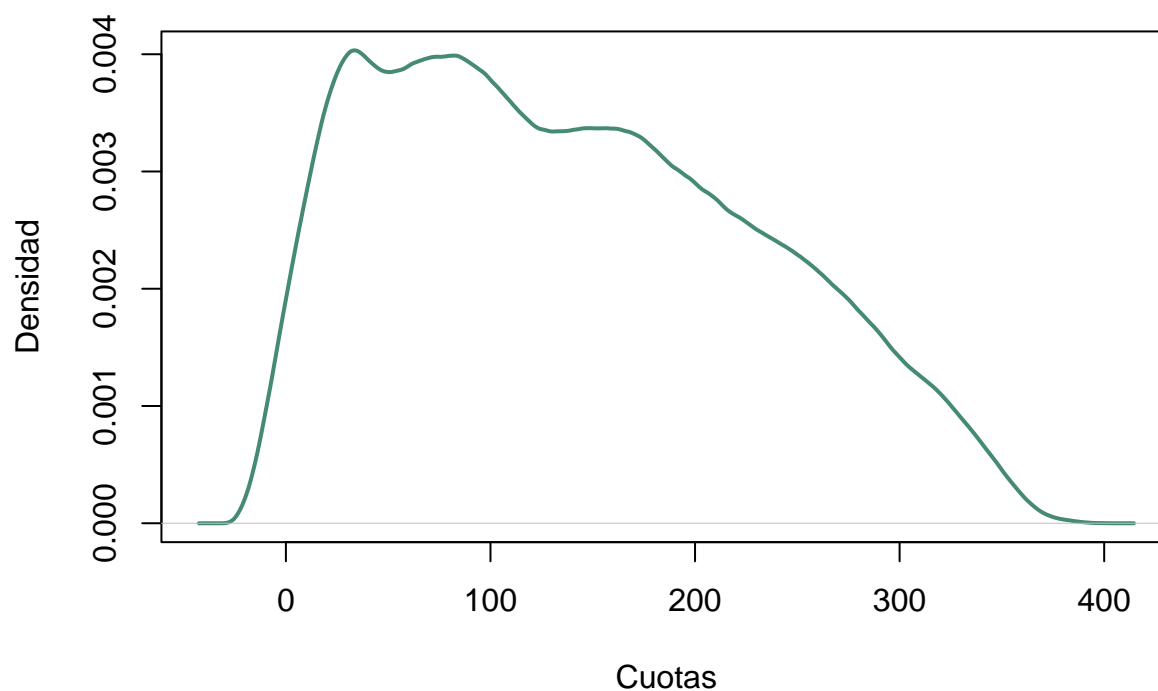


Epanechnikov

```
# Graficamos con Kernel de Epanechnikov.  
plot(density(base_salarios$Coutas, kernel = "epanechnikov"),  
     col = "aquamarine4", lwd = 2,  
     main = "Densidad de las Cuotas según kernel Epanechnikov",  
     xlab = "Cuotas", ylab = "Densidad")
```



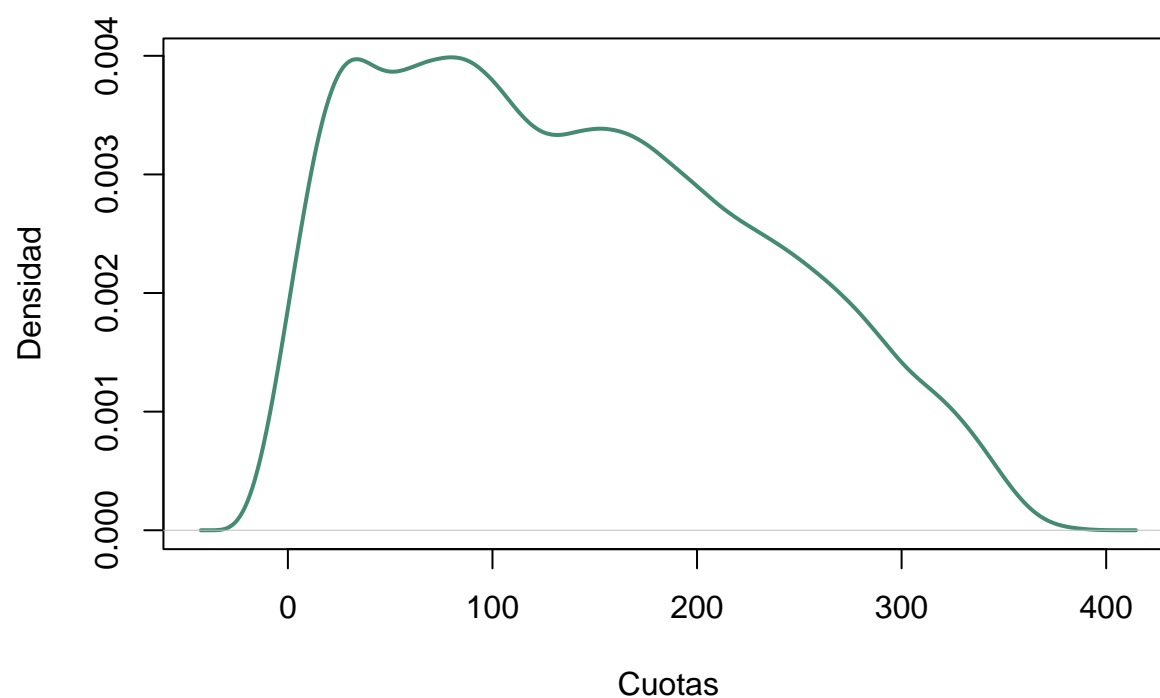
### Densidad de las Cuotas según kernel Epanechnikov



### Coseno

```
# Graficamos con Kernel Coseno.  
plot(density(base_salarios$Coutas, kernel = "cosine"),  
     col = "aquamarine4", lwd = 2,  
     main = "Densidad de las Cuotas según kernel Coseno",  
     xlab = "Cuotas", ylab = "Densidad")
```

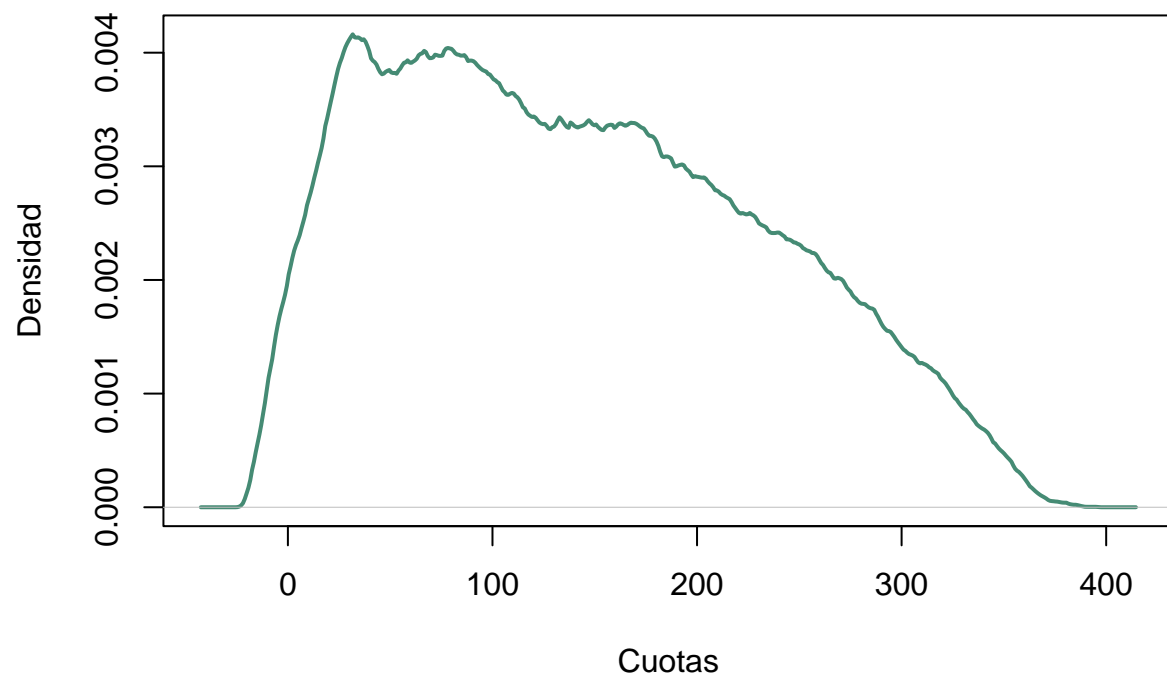
## Densidad de las Cuotas según kernel Coseno



## Uniforme

```
# Graficamos con Kernel Uniforme.  
plot(density(base_salarios$Coutas, kernel = "rectangular"),  
     col = "aquamarine4", lwd = 2,  
     main = "Densidad de las Cuotas según kernel Rectangular",  
     xlab = "Cuotas", ylab = "Densidad")
```

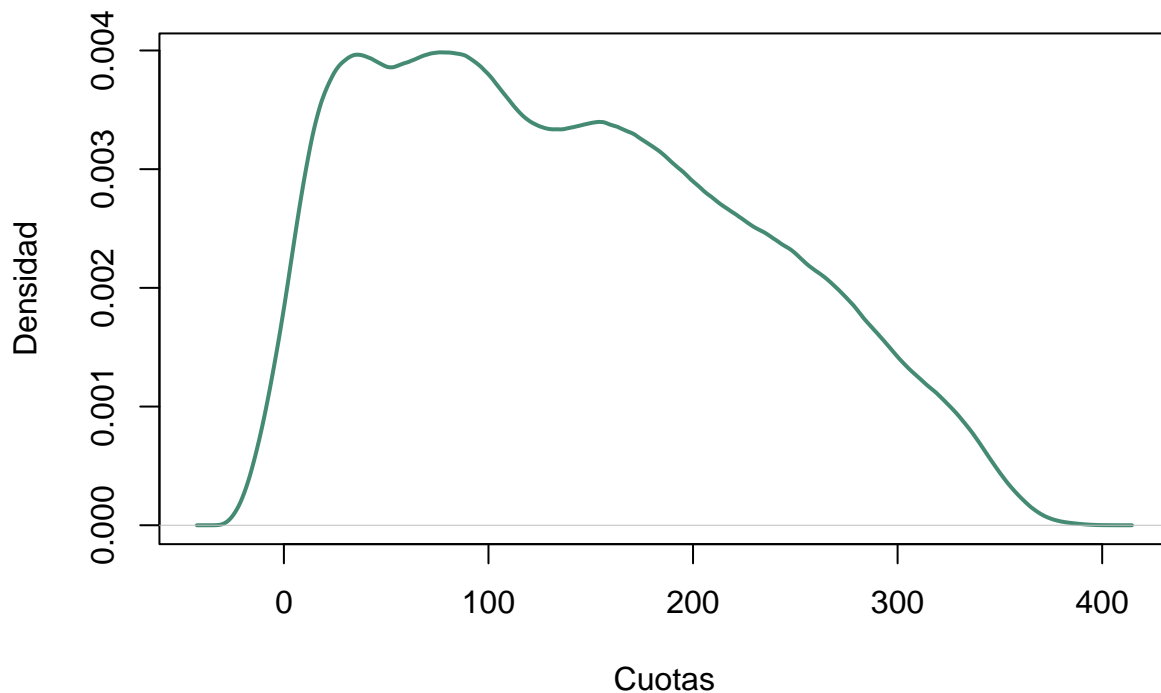
## Densidad de las Cuotas según kernel Rectangular



### Triangular

```
# Graficamos con Kernel Triangular.  
plot(density(base_salarios$Coutas, kernel = "triangular"),  
     col = "aquamarine4", lwd = 2,  
     main = "Densidad de las Cuotas según kernel Triangular",  
     xlab = "Cuotas", ylab = "Densidad")
```

## Densidad de las Cuotas según kernel Triangular



### 3

Creamos un vector con los nombres de todos los kernels disponibles para la función density.

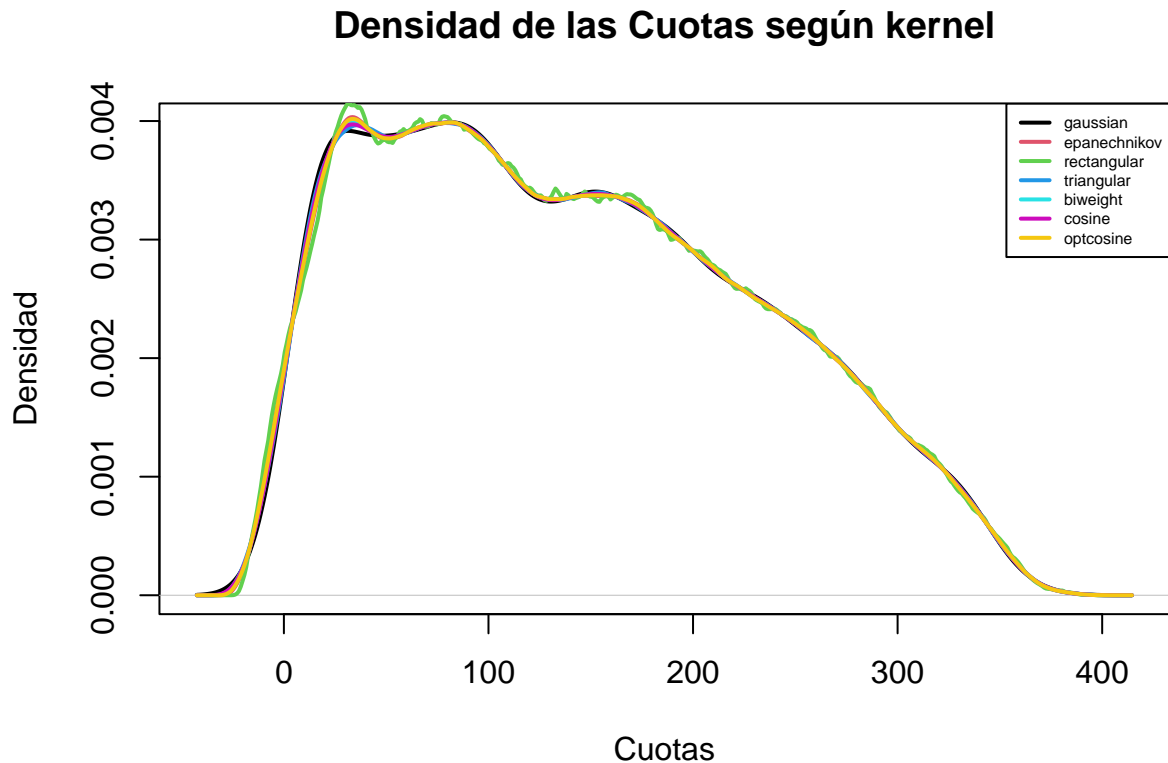
```
(kernels <- eval(formals(density.default)$kernel))
```

```
## [1] "gaussian"      "epanechnikov" "rectangular"  "triangular"   "biweight"
## [6] "cosine"       "optcosine"
```

Ahora, generamos el gráfico con todos a la vez, para apreciar las diferencias entre cada uno de estos.

```
# primero llamamos el kernel normal como base para el gráfico.
plot(density(base_salarios$Coutas, kernel = "gaussian"),
     col = "black", lwd = 2,
     main = "Densidad de las Cuotas según kernel",
     xlab = "Cuotas", ylab = "Densidad")

# creamos un ciclo for, para evaluar cada kernel, y añadirlo al gráfico.
for(i in 2:length(kernels))
  lines(density(base_salarios$Coutas, kernel = kernels[i]), col = i, lwd = 2)
legend("topright", legend = kernels, col = seq(kernels), lty = 1, lwd = 2, cex = 0.5)
```



## Ejercicio 3

1

El Criterio de Información de Akaike (AIC) evalúa la calidad de un modelo ajustado considerando tanto la cantidad de parámetros como la calidad del ajuste. La fórmula del AIC es:

$$AIC = 2K - 2\ln(L)$$

Donde k representa el número de parámetros del modelo y L es la función de máxima verosimilitud del modelo. El AIC penaliza los modelos más complejos, es decir, aquellos con más parámetros, para reducir el riesgo de sobreajuste. Se considera que el mejor modelo es aquel con el valor de AIC más bajo.

2

```
base_salarios <- read.csv("BaseSalarios.csv", header = TRUE, sep = ";",
                          stringsAsFactors = FALSE)
base_salarios <- base_salarios[, !names(base_salarios) %in% "X"]
base_salarios <- na.omit(base_salarios)
```

```
valores <- model_select(base_salarios$Coutas)
print(valores)
```

```
## Maximum likelihood estimates for the Skew Generalized Error model
##      mean      sd      nu      xi
## 145.352   87.772   5.253   6.219

mean_model <- as.numeric(valores["mean"])
sd_model <- as.numeric(valores["sd"])
nu_model <- as.numeric(valores["nu"])
xi_model <- as.numeric(valores["xi"])

L <- sum(dsged(base_salarios$Coutas, mean = mean_model, sd = sd_model,
              nu = nu_model, xi = xi_model, log = TRUE))
k <- 4
AIC_1 <- -2 * L + 2 * k

exp <- mlexp(base_salarios$Coutas)
gamma <- mlgamma(base_salarios$Coutas)
lognormal <- mllnorm(base_salarios$Coutas)
weibull <- mlweibull(base_salarios$Coutas)
loggamma <- mllgamma(base_salarios$Coutas+1)
uniform <- mlunif(base_salarios$Coutas)

AIC_Calculados <- data.frame(
  DistribuCION = c("Model_Select", "Exponential", "Gamma", "Lognormal",
                  "Weibull", "Loggamma", "Uniform"),
  AIC = c(AIC_1,
          AIC(exp),
          AIC(gamma),
          AIC(lognormal),
          AIC(weibull),
          AIC(loggamma),
          AIC(uniform))
)

print(AIC_Calculados)
```

```
##   DistribuCION      AIC
## 1 Model_Select 63470.77
## 2 Exponential 65171.78
## 3      Gamma 64325.05
## 4   Lognormal 65593.94
## 5     Weibull 64046.54
## 6   Loggamma 66570.95
## 7     Uniform 64780.51
```

De la tabla se logra observar que el menor valor de AIC corresponde a la distribución Weibull, de forma que siguiendo el Criterio de Información de Akaike es elige este.

### 3

```
ajuste_risk <- fit.cont(base_salarios$Coutas)
```

```

##
## Begin fitting distributions -----

## * fitting normal distribution ... OK

## * fitting Cauchy distribution ... OK

## * fitting logistic distribution ... OK

## * fitting beta distribution ... failed

## * fitting exponential distribution ... OK

## * fitting chi-square distribution ... OK

## * fitting uniform distribution ... OK

## * fitting gamma distribution ... OK

## * fitting lognormal distribution ... OK

## * fitting Weibull distribution ... OK

## * fitting F-distribution ... OK

## * fitting Student's t-distribution ... OK

## * fitting Gompertz distribution ... OK

## * fitting triangular distribution ... failed

## End fitting distributions -----

##          logL      AIC      BIC Chisq(value) Chisq(p) AD(value)
## Normal      -32430.73  64865.45  64878.67      1216.35      0      56.49
## Cauchy       -33822.33  67648.66  67661.88      3531.13      0     143.65
## Logistic     -32618.92  65241.83  65255.05      1500.51      0      59.21
## Exponential  -32584.89  65171.78  65178.39      1395.05      0     210.33
## Chi-square   -124028.73 248059.46 248066.06           Inf      0           Inf
## Uniform          NULL      NULL      NULL           Inf      0           Inf
## Gamma        -32160.53  64325.05  64338.27       803.46      0      58.07
## Lognormal    -32794.97  65593.94  65607.15      2189.18      0     149.71
## Weibull      -32021.27  64046.54  64059.75       593.93      0      40.13
## F            -40588.69  81181.37  81194.59     30957.52      0    2200.72
## Student      -44604.57  89211.15  89217.75     69809.16      0    4756.44
## Gompertz     -31811.61  63627.22  63640.43       200.06      0       9.10
##          H(AD) KS(value)  H(KS)
## Normal    rejected      0.07 rejected
## Cauchy     rejected      0.15 rejected
## Logistic   rejected      0.08 rejected

```

```
## Exponential rejected      0.12 rejected
## Chi-square      NULL      0.47 rejected
## Uniform         NULL      0.06 rejected
## Gamma           rejected   0.07 rejected
## Lognormal       rejected   0.11 rejected
## Weibull         rejected   0.05 rejected
## F               NULL      0.50 rejected
## Student         NULL      0.72 rejected
## Gompertz        NULL      0.03 rejected
```

```
##
## Chosen continuous distribution is: Normal (norm)
## Fitted parameters are:
##      mean      sd
## 141.0909  90.2261
```

```
print(ajuste_risk)
```

```
## $data2fit
##      [1] 294 191 255  75 213 201 302 287 306 135 173 326 269 209 249 200 307 243
##      [19] 267 274 319 234 297 102 191 270 283 319  61 157 266 271 222 221 310 212
##      [37] 251 345 169 323 289 300 127 295 315 315 329 271 296 277 313 356 324 261
##      [55] 331 321 339 259 326 334 344 315 244 306 315 265 326 293 330 234 363 253
##      [73] 345 279 265 277 271 278 221 251 264  44 213 299 282 220 229 223 204 319
##      [91] 304 258 246 346 239 334 313 337 334 256 296 321 298 265 337 294 282 284
##     [109] 136 305 296 310 279 172 349 262 330 309 265 265 275 332 271 309 320 293
##     [127] 281 297 255 249 150 275 271 334 311 342 233 317 308 266 342 252 311 264
##     [145] 227 101 323 233  70 254 335 341 275 340 280 259 318 250 292 324 295 337
##     [163]  49 343 204 316 317 336 300 305 312 309  69 317 125 262 236 325 329 333
##     [181]  49 286 361 290 263 332 241  57 339 177 326 303 191 371 276 319 262 309
##     [199] 307 236 161 337 224 186 111 328 261 225 329 335 321 290 282 215 237 235
##     [217] 340 283 235 217 276 330 275 278 227 336 297 250 319 325 276  84 285 247
##     [235] 272 305 317 271 328 289 293 322 326 258 285 294 313 304 269 299 318 122
##     [253] 322 266 221 327 163 294 303 243 179 318 282 282 293 302 233 279 318  40
##     [271] 306 328 191 310 344 258 278 307  99 233 248 262 274 287 293 139 286 139
##     [289] 237 260 219 257 220 259 272 248 242  34 332 264 274 253 311 270 203 217
##     [307] 274 282 280 229 216 290 271 321 268 322 335 267 299 268 257 253 106 241
##     [325] 265 201 234  87 285 267 273 325 267 294 170 271 289 273 282 213 237 273
##     [343] 303 303 278 273 195 245 232 241 161 267 285 229 279 279 209 146 253 268
##     [361] 268 273 259 258 241 121 277 257 291 288 208 239 228 248 213 308 251 215
##     [379] 293 280 278 251 271 261 236 243 266 253 270 226 269 287 247 261 285 260
##     [397] 269 242 230 184 248 252 275 260 221 253 281 258 263 274 212 254 251 215
##     [415] 237 191 264 198 249 232 287 233 262 253 222 262 223 249 271 207 251 249
##     [433] 247 162 304 237 263 220 257 272 133 259 258 203 239 250 272 262 271 261
##     [451] 272 242 244 221 156 254 253 250 240 153 223 257 250 251 281 307 283 264
##     [469] 254 272 337 312 230 253 250 171 311 291 108 273 243  24 357 269  65 280
##     [487] 335 321 304 275 302 319 287 262 203 336 302 273 159 271 264 248 342 340
##     [505] 153 283 263  53 275 274 316 254 282 322 289 336 330 328 339 324 337 294
##     [523] 293 241 293 251 103 264 341 256 285 289 286 311 293 286 233 307 291 262
##     [541] 232 274 330 321 195 260 303 274 293 335 299 225 313 248 270  20 332 289
##     [559] 249 307 234 300 246 283 203 326 275 283 288 294 300 301 245 338 220 262
##     [577] 295 300 131 288 263 264 267 249  94 200 282 272 255 287 217 251 250 258
##     [595] 245 263 275 301 297 249 261 263 267 156 268 247 221 251 273 277 233 146
```



```

## [613] 216 236 231 216 256 218 142 236 299 248 258 278 255 305 254 267 336 249
## [631] 337 334 282 280 297 292 347 253 327 132 259 290 223 317 96 336 276 254
## [649] 254 295 325 172 278 329 290 26 309 332 218 269 302 291 342 145 340 77
## [667] 344 308 312 272 291 262 276 278 246 306 300 285 218 262 296 275 288 259
## [685] 55 169 250 290 248 248 272 287 327 270 143 285 207 280 329 270 278 278
## [703] 240 203 244 280 204 287 282 250 270 249 119 240 250 246 247 118 208 248
## [721] 203 256 214 253 244 298 313 320 268 338 284 251 313 109 330 265 128 324
## [739] 309 308 299 298 340 311 346 282 260 285 305 261 53 330 119 281 311 300
## [757] 261 254 261 251 272 268 238 296 274 260 307 309 271 262 263 244 328 267
## [775] 329 302 203 272 238 314 246 265 296 237 300 302 362 277 299 293 329 194
## [793] 172 280 205 282 275 226 283 128 318 251 182 271 252 123 279 351 85 289
## [811] 296 221 311 268 337 252 303 316 284 186 358 253 322 280 290 79 127 252
## [829] 339 322 346 265 275 288 307 318 318 315 319 266 259 341 280 276 309 299
## [847] 346 329 240 311 241 246 247 297 307 247 329 277 263 293 310 261 279 287
## [865] 290 295 264 265 250 309 267 269 91 260 275 279 256 248 210 264 325 286
## [883] 262 266 296 268 205 276 253 231 252 239 287 258 286 250 248 265 247 265
## [901] 329 275 102 328 294 297 300 356 310 267 247 339 258 273 334 91 364 279
## [919] 227 247 249 213 324 197 164 271 243 283 310 257 153 267 323 240 244 276
## [937] 266 247 264 333 138 274 330 61 347 338 30 267 333 109 301 300 131 319
## [955] 264 321 332 278 212 247 345 352 326 298 278 321 278 218 297 278 296 282
## [973] 138 202 299 312 242 249 133 323 283 32 353 270 241 257 293 250 277 249
## [991] 250 253 251 240 273 263 307 249 255 257 248 258 218 234 278 274 242 262
## [1009] 265 252 239 146 95 248 85 262 281 296 157 82 269 304 285 330 335 319
## [1027] 318 342 290 286 263 335 257 257 284 202 271 242 293 33 261 321 252 278
## [1045] 333 292 289 204 317 111 247 319 285 313 288 266 272 245 114 234 271 244
## [1063] 245 262 157 290 282 259 231 277 245 264 251 233 245 261 292 256 305 295
## [1081] 129 270 131 187 298 201 334 271 302 61 88 290 296 115 250 247 263 336
## [1099] 286 327 330 268 361 294 258 247 287 303 258 269 180 247 237 238 203 247
## [1117] 247 238 237 244 184 204 234 235 235 87 235 235 236 254 244 205 238 243
## [1135] 231 197 242 246 185 245 234 134 243 245 212 234 246 241 236 244 235 238
## [1153] 253 245 108 231 246 240 226 208 236 239 183 242 232 240 235 214 246 247
## [1171] 224 225 236 237 238 239 215 190 234 228 238 241 237 219 241 224 242 224
## [1189] 197 242 241 200 235 226 235 239 238 246 238 221 196 173 239 243 212 240
## [1207] 185 200 237 233 238 228 241 234 230 167 235 237 231 97 234 234 237 189
## [1225] 241 246 222 235 237 241 64 235 110 228 227 207 235 237 232 231 225 231
## [1243] 213 198 224 241 193 228 230 176 229 213 235 241 232 189 124 232 233 231
## [1261] 190 236 187 228 238 236 132 229 235 241 121 237 238 149 236 234 228 356
## [1279] 311 235 179 238 232 235 118 227 96 224 239 195 134 223 227 126 226 229
## [1297] 234 222 226 224 228 230 223 211 229 211 209 231 124 227 224 232 224 223
## [1315] 228 221 198 228 220 213 269 228 231 56 224 237 228 199 224 231 233 224
## [1333] 215 234 202 228 217 59 226 224 210 217 225 225 230 221 232 231 226 221
## [1351] 231 224 164 170 193 225 226 221 227 227 204 198 232 236 198 227 206 233
## [1369] 232 228 226 217 194 226 195 185 247 210 226 231 227 229 232 191 229 223
## [1387] 227 196 227 224 224 216 188 225 201 232 226 221 223 222 230 235 226 213
## [1405] 214 205 199 220 213 210 228 221 219 229 166 213 208 128 230 67 144 220
## [1423] 189 192 229 226 259 228 211 201 205 225 185 221 220 219 194 216 220 221
## [1441] 220 207 226 210 183 213 175 34 210 195 225 196 216 217 221 170 225 204
## [1459] 88 204 63 221 169 211 183 198 225 207 224 177 220 214 216 221 215 84
## [1477] 230 131 213 183 223 218 213 217 213 218 211 218 163 211 145 220 208 211
## [1495] 155 207 198 214 215 215 198 182 208 270 223 203 212 211 195 151 222 215
## [1513] 217 182 222 217 172 221 222 216 215 211 219 211 192 221 218 215 222 216
## [1531] 208 223 213 174 218 218 211 207 220 211 143 211 216 221 218 179 266 212
## [1549] 221 220 176 215 204 219 343 198 219 210 209 210 201 138 209 205 187 222
## [1567] 172 218 216 78 219 199 218 66 223 212 217 173 213 68 149 208 214 184

```

```

## [1585] 215 97 189 206 122 215 153 177 213 199 218 212 215 218 127 217 150 202
## [1603] 204 217 216 136 216 214 186 185 207 220 200 121 217 208 280 208 138 153
## [1621] 218 210 215 166 169 206 151 209 212 204 210 50 185 186 209 181 207 206
## [1639] 174 196 205 176 198 210 197 209 170 196 249 196 198 201 202 215 172 154
## [1657] 189 182 217 200 94 187 209 134 190 248 201 150 180 207 196 188 209 211
## [1675] 203 207 207 199 210 202 210 204 208 196 211 201 203 180 207 95 205 202
## [1693] 207 182 203 202 141 201 170 206 205 198 141 193 163 194 206 202 193 144
## [1711] 208 199 203 203 203 111 202 200 209 162 210 168 202 204 196 157 175 205
## [1729] 198 211 201 204 201 205 112 186 198 198 204 48 203 97 200 195 205 197
## [1747] 206 199 195 199 211 199 209 204 124 204 200 189 204 209 204 195 211 182
## [1765] 209 153 205 196 197 202 189 202 209 198 187 192 168 171 187 207 171 160
## [1783] 201 198 208 150 136 145 180 199 199 200 192 182 185 191 187 175 194 195
## [1801] 203 195 199 174 197 203 162 82 192 205 188 147 193 197 187 190 169 192
## [1819] 172 201 194 202 163 185 194 137 196 195 154 207 89 97 201 153 156 121
## [1837] 199 127 192 193 183 183 51 193 134 196 190 199 196 110 62 190 197 192
## [1855] 192 193 186 113 197 156 176 158 97 298 189 143 98 189 179 187 195 147
## [1873] 194 191 191 24 191 196 197 197 173 198 195 197 171 196 193 183 187 197
## [1891] 197 195 133 193 176 151 193 157 156 188 197 187 190 39 189 193 191 192
## [1909] 194 192 180 194 191 175 162 188 186 167 191 191 183 176 195 168 190 191
## [1927] 185 191 178 192 193 188 197 181 189 175 176 181 186 183 193 191 194 188
## [1945] 196 193 186 42 188 167 151 177 180 168 195 142 192 196 151 184 128 190
## [1963] 184 189 99 195 187 186 180 194 183 192 195 173 17 192 178 144 190 194
## [1981] 186 194 185 170 240 188 172 183 184 170 188 192 187 186 160 163 171 191
## [1999] 149 189 189 182 185 179 176 92 45 192 185 191 177 120 183 177 185 267
## [2017] 154 186 186 184 159 187 181 185 104 107 187 152 186 179 185 146 180 127
## [2035] 156 158 178 189 183 108 184 181 182 163 190 183 141 185 100 183 188 184
## [2053] 185 69 190 148 168 186 113 170 179 180 184 185 183 183 177 152 184 184
## [2071] 187 192 181 252 189 181 181 140 157 180 181 182 163 183 181 163 183 175
## [2089] 192 180 187 181 178 186 104 178 188 186 128 130 163 177 182 183 181 187
## [2107] 173 187 175 182 180 187 140 168 180 108 141 185 175 182 186 174 178 181
## [2125] 178 111 167 180 183 142 185 179 87 179 297 100 110 172 119 181 96 181
## [2143] 185 146 147 175 176 131 171 178 175 179 184 172 179 180 182 118 179 154
## [2161] 181 153 170 165 177 175 183 178 172 180 172 170 176 172 180 172 180 186
## [2179] 179 173 169 105 167 177 184 174 156 177 177 181 179 162 173 182 170 169
## [2197] 183 176 59 168 166 165 182 184 32 173 163 123 166 154 165 172 148 138
## [2215] 132 163 168 155 175 171 178 154 171 174 171 114 90 178 124 110 157 179
## [2233] 178 168 164 171 183 172 156 176 151 152 186 182 172 169 176 170 177 240
## [2251] 149 175 174 134 169 178 172 169 155 154 177 176 169 180 216 174 177 172
## [2269] 171 176 167 169 169 160 161 157 167 119 168 147 155 170 127 170 238 169
## [2287] 172 161 131 166 171 168 166 138 165 146 170 126 171 177 91 155 175 149
## [2305] 163 171 171 115 153 165 167 126 70 139 148 174 154 143 73 167 174 144
## [2323] 170 168 116 167 155 137 153 164 97 40 163 137 171 175 174 171 168 157
## [2341] 165 160 160 19 170 170 170 156 42 161 167 74 171 162 161 163 163 164
## [2359] 155 163 162 127 164 174 174 171 164 170 166 169 170 162 174 153 171 93
## [2377] 167 162 165 158 157 111 158 158 171 163 151 177 166 169 168 139 103 168
## [2395] 172 168 162 165 120 133 154 123 143 157 166 163 171 167 166 156 162 166
## [2413] 147 168 119 169 145 150 164 136 162 161 168 160 138 169 166 138 109 165
## [2431] 132 162 156 164 118 166 147 169 150 141 172 167 100 167 156 157 161 161
## [2449] 154 163 165 141 156 161 145 170 137 133 164 155 165 163 106 159 157 158
## [2467] 159 128 160 149 163 62 49 135 158 158 145 163 83 159 121 161 163 76
## [2485] 158 161 156 158 120 112 124 161 166 161 147 144 120 165 201 162 163 126
## [2503] 156 164 136 139 143 138 160 145 165 134 161 159 157 160 88 153 163 160
## [2521] 160 157 161 155 156 158 9 149 136 143 156 157 160 140 150 159 164 157
## [2539] 282 154 158 149 147 147 121 32 137 156 157 151 78 130 158 146 156 139

```

```

## [2557] 160 148 159 157 163 151 160 153 157 158 151 151 157 29 130 145 72 155
## [2575] 153 152 136 149 186 163 148 158 152 98 149 155 81 156 161 155 156 157
## [2593] 147 122 150 29 128 159 144 100 150 157 157 158 154 133 156 152 157 32
## [2611] 141 151 156 153 162 160 153 160 85 155 154 163 126 68 155 154 148 149
## [2629] 146 144 156 150 154 156 154 154 153 155 152 161 159 156 158 150 70 151
## [2647] 159 146 134 139 155 154 151 150 146 113 150 153 150 133 131 151 154 156
## [2665] 157 141 104 151 65 152 139 267 129 155 106 150 141 157 160 92 147 147
## [2683] 156 140 123 157 152 127 146 148 147 82 145 108 157 129 154 133 136 146
## [2701] 152 151 140 155 144 127 145 152 149 147 151 157 139 128 116 141 154 147
## [2719] 157 155 144 139 140 136 152 141 74 78 173 152 149 119 134 104 150 126
## [2737] 133 102 133 149 59 133 66 157 150 113 148 137 146 145 144 151 147 131
## [2755] 140 153 147 44 108 89 140 147 151 112 150 84 151 145 148 142 143 77
## [2773] 292 154 53 91 141 106 20 87 100 50 145 108 92 144 149 89 150 143
## [2791] 140 140 145 141 139 136 101 150 153 94 147 136 141 103 299 145 148 139
## [2809] 135 142 150 148 150 129 131 149 31 143 143 110 124 143 149 143 116 144
## [2827] 173 146 145 143 147 135 130 135 85 72 138 101 148 114 141 132 149 141
## [2845] 134 143 145 146 139 137 145 148 145 148 134 142 149 114 118 68 146 137
## [2863] 142 144 137 147 121 100 98 135 133 56 142 146 29 132 150 141 144 124
## [2881] 134 138 142 127 60 122 128 145 139 135 140 146 144 129 136 147 143 137
## [2899] 134 141 134 147 117 137 135 123 142 143 142 148 132 112 54 124 134 124
## [2917] 139 130 135 101 130 110 119 91 139 140 135 93 126 133 132 141 146 139
## [2935] 116 209 56 108 138 138 113 120 133 130 24 133 135 138 144 101 146 144
## [2953] 124 135 143 120 137 134 132 140 33 241 246 175 118 128 119 56 105 134
## [2971] 140 132 115 147 318 131 132 111 19 134 288 246 304 123 123 96 71 123
## [2989] 190 198 126 110 132 134 118 133 134 140 136 98 138 126 138 135 132 93
## [3007] 100 9 137 134 134 139 84 126 147 114 128 89 114 122 133 110 75 133
## [3025] 129 97 137 31 134 137 129 117 128 78 163 110 97 60 133 100 97 128
## [3043] 131 117 132 133 135 107 130 126 91 111 138 56 123 127 128 138 116 129
## [3061] 128 134 125 121 94 137 132 128 129 123 122 124 131 130 129 128 125 128
## [3079] 132 139 126 107 191 319 191 134 126 127 81 128 132 133 131 113 105 126
## [3097] 119 124 119 162 120 122 91 125 134 127 124 127 133 130 180 137 84 83
## [3115] 126 124 134 188 123 109 109 117 110 131 115 132 104 134 127 123 124 118
## [3133] 106 63 121 132 102 125 89 132 127 85 113 123 109 79 118 125 119 39
## [3151] 121 125 122 125 86 130 90 124 131 127 118 122 123 129 100 184 140 68
## [3169] 102 90 96 126 113 122 92 106 121 122 114 126 128 107 119 117 124 126
## [3187] 109 110 115 194 126 111 81 105 125 119 119 20 126 116 122 120 122 100
## [3205] 111 117 129 120 59 152 190 124 110 127 106 123 117 115 88 89 119 116
## [3223] 111 50 122 298 122 112 117 115 81 116 119 124 88 126 91 121 111 46
## [3241] 119 69 117 107 108 111 118 116 125 111 118 118 121 126 116 123 116 116
## [3259] 106 117 120 107 98 90 122 121 113 110 103 120 115 115 75 116 121 106
## [3277] 98 24 115 120 115 121 85 115 114 88 118 107 97 121 114 111 90 125
## [3295] 116 114 123 115 111 93 117 113 113 123 120 119 106 123 109 122 114 97
## [3313] 101 105 97 121 114 114 68 109 217 79 113 89 59 119 99 123 110 96
## [3331] 43 108 101 51 116 94 112 123 21 104 77 101 111 99 64 114 120 110
## [3349] 96 80 93 25 121 93 101 11 105 66 107 49 109 98 105 114 84 105
## [3367] 114 53 108 96 119 108 109 112 112 111 119 108 113 105 116 107 97 95
## [3385] 48 115 112 117 88 88 114 108 54 113 95 106 114 85 80 116 79 102
## [3403] 68 106 111 78 118 42 104 80 112 34 114 104 104 96 104 109 84 50
## [3421] 114 88 107 110 104 55 109 99 111 104 96 109 106 113 105 102 109 106
## [3439] 102 103 109 104 119 67 19 105 54 101 112 110 105 95 112 105 109 108
## [3457] 89 56 106 113 111 103 85 102 110 59 100 108 98 101 101 74 95 114
## [3475] 109 102 104 112 114 102 97 102 103 97 87 60 96 110 110 112 102 99
## [3493] 100 104 74 111 90 109 104 98 102 114 103 103 99 109 102 111 110 96
## [3511] 89 76 102 102 110 111 104 89 96 103 111 107 101 104 92 104 104 85

```

```

## [3529] 108 82 107 91 105 95 44 71 110 110 68 109 72 55 99 102 110 89
## [3547] 75 101 71 99 102 95 103 104 104 32 102 99 93 97 67 97 46 98
## [3565] 91 81 102 99 97 13 103 91 101 79 102 89 101 95 98 100 106 97
## [3583] 104 66 109 100 96 88 99 25 18 98 107 81 101 102 106 91 99 96
## [3601] 105 94 100 105 93 107 89 62 96 95 83 89 107 99 82 101 105 106
## [3619] 99 91 98 77 20 108 48 93 94 88 98 99 95 56 86 72 102 91
## [3637] 102 102 97 79 102 92 97 92 102 93 92 97 74 96 96 97 98 91
## [3655] 87 99 91 88 99 58 94 81 88 94 88 59 93 96 95 100 76 89
## [3673] 85 49 48 80 90 93 48 95 43 97 90 22 79 101 71 74 89 102
## [3691] 101 102 96 24 100 92 63 95 91 79 96 81 96 92 21 90 90 90
## [3709] 88 102 86 95 91 75 49 93 93 91 61 91 94 93 69 57 57 91
## [3727] 83 98 100 101 99 82 90 82 92 82 96 183 95 99 96 81 98 93
## [3745] 93 90 87 37 27 93 101 83 93 78 82 101 87 93 94 81 84 96
## [3763] 99 79 89 64 96 97 71 23 67 72 72 88 88 77 38 89 96 89
## [3781] 90 91 93 88 97 92 52 71 94 75 97 97 99 97 94 82 79 100
## [3799] 88 80 95 88 96 83 22 77 97 85 57 88 86 174 93 24 95 89
## [3817] 92 94 71 81 94 87 87 72 83 81 70 77 15 87 92 89 89 74
## [3835] 40 95 85 95 75 85 93 91 21 77 90 87 92 64 79 76 19 92
## [3853] 95 83 77 94 93 76 75 86 86 94 87 92 88 73 89 93 90 82
## [3871] 57 88 85 88 94 86 86 91 255 83 89 88 87 87 88 69 89 69
## [3889] 84 86 91 88 83 35 44 54 8 91 83 77 82 85 87 82 79 69
## [3907] 81 74 74 84 89 80 86 91 86 80 76 79 53 45 65 78 68 77
## [3925] 86 82 78 79 80 87 77 82 85 68 81 84 80 81 79 76 90 60
## [3943] 91 87 60 81 83 78 91 79 42 79 81 83 63 85 82 79 86 82
## [3961] 84 71 89 68 85 59 88 61 89 86 81 83 85 72 82 83 81 216
## [3979] 79 78 87 83 85 39 78 66 71 80 83 64 89 59 84 71 86 78
## [3997] 76 27 84 62 79 76 73 74 78 81 80 79 49 71 76 86 83 74
## [4015] 75 81 81 79 78 79 78 74 87 86 75 72 87 83 81 66 194 72
## [4033] 83 77 81 79 62 73 68 82 86 58 76 73 194 47 54 80 75 85
## [4051] 69 62 63 73 20 75 81 86 135 85 86 70 63 76 75 77 81 73
## [4069] 83 62 65 68 41 72 75 79 73 84 79 142 43 75 58 61 73 74
## [4087] 79 74 53 72 69 72 72 80 83 60 48 83 65 62 72 83 77 52
## [4105] 78 54 69 73 82 83 67 53 65 75 65 77 57 43 71 63 69 78
## [4123] 70 53 73 76 52 72 80 77 75 76 50 63 78 72 69 67 73 76
## [4141] 72 75 75 77 70 68 69 36 76 73 74 70 70 67 40 72 47 75
## [4159] 66 73 74 77 37 68 59 72 65 69 71 73 70 76 62 60 76 59
## [4177] 71 72 67 76 35 75 73 70 67 66 69 62 75 72 66 77 68 77
## [4195] 76 27 69 66 47 63 66 77 55 59 68 64 66 78 76 76 73 65
## [4213] 23 70 65 66 64 65 37 72 69 66 73 63 75 72 71 51 40 68
## [4231] 75 77 42 77 56 67 73 63 63 67 47 65 72 62 72 66 67 51
## [4249] 58 73 69 74 74 66 41 65 71 75 72 74 74 55 65 70 67 67
## [4267] 70 69 69 72 62 74 59 59 67 68 28 63 59 59 34 56 26 66
## [4285] 70 47 56 61 69 67 25 68 45 50 63 60 67 28 69 67 54 66
## [4303] 62 64 62 67 59 69 34 60 18 63 52 73 62 131 59 55 65 27
## [4321] 68 54 69 60 45 42 70 61 42 68 70 62 65 48 60 43 70 53
## [4339] 50 69 54 64 62 62 26 10 61 68 58 62 43 59 60 23 35 64
## [4357] 49 52 45 64 31 67 66 57 106 37 64 58 51 60 66 22 61 57
## [4375] 56 58 43 55 57 59 57 57 63 41 40 61 58 47 56 49 58 57
## [4393] 47 63 56 63 59 59 58 58 67 64 61 65 57 55 61 50 60 56
## [4411] 54 64 55 60 62 49 61 57 52 57 100 64 57 49 58 33 55 27
## [4429] 62 60 60 60 54 18 38 55 35 57 235 55 19 90 62 20 43 48
## [4447] 62 56 57 27 30 33 63 53 53 64 54 59 54 62 27 46 60 60
## [4465] 56 59 56 52 61 58 163 60 29 60 239 62 29 55 56 38 61 31
## [4483] 54 28 58 56 62 54 55 48 65 50 59 58 52 43 62 29 58 62

```

```

## [4501] 25 52 56 63 55 19 59 55 25 48 59 58 55 25 75 57 19 55
## [4519] 41 56 44 34 60 22 54 55 55 55 46 48 19 45 53 34 43 47
## [4537] 57 38 59 56 14 58 45 37 29 47 45 52 21 49 55 53 44 47
## [4555] 49 56 54 48 53 33 59 51 51 76 32 52 54 57 43 44 56 50
## [4573] 48 57 50 45 52 50 38 45 51 52 53 170 49 49 45 52 47 43
## [4591] 55 46 54 47 48 44 55 30 46 34 38 45 40 49 48 42 32 43
## [4609] 27 43 48 52 42 42 42 45 40 45 46 43 43 53 43 49 15 46
## [4627] 43 52 43 46 48 43 49 45 50 47 41 46 54 28 51 14 37 47
## [4645] 42 46 34 40 45 44 45 43 43 42 50 36 29 44 49 53 49 40
## [4663] 42 40 41 52 47 48 17 45 51 302 42 42 47 44 25 28 43 40
## [4681] 27 46 43 51 42 45 46 19 47 50 52 29 41 45 48 18 50 37
## [4699] 46 50 125 48 50 19 38 50 45 40 47 46 32 21 10 45 36 43
## [4717] 44 41 11 44 30 40 31 27 41 39 11 38 23 42 40 22 48 43
## [4735] 35 39 46 44 25 41 46 49 47 18 34 47 13 47 43 42 46 28
## [4753] 16 43 47 41 38 30 38 38 49 36 46 37 34 38 42 43 34 17
## [4771] 39 14 36 36 38 35 35 35 188 36 35 42 37 34 43 48 41 39
## [4789] 38 48 49 45 42 26 46 45 37 24 40 38 40 31 42 37 19 35
## [4807] 43 35 36 42 43 36 32 15 33 47 44 35 40 37 40 16 40 153
## [4825] 46 13 34 44 47 13 38 17 34 36 34 40 40 45 39 40 22 40
## [4843] 40 39 40 25 28 23 42 53 33 21 41 18 41 20 34 32 39 32
## [4861] 35 25 43 44 43 38 44 16 33 33 37 38 37 25 32 28 31 37
## [4879] 40 34 32 37 36 33 34 30 30 18 34 19 38 37 33 39 36 34
## [4897] 31 31 38 34 34 33 26 29 32 42 39 34 36 17 32 31 4 28
## [4915] 32 38 40 30 39 36 36 27 37 13 27 28 36 38 32 38 18 31
## [4933] 27 36 38 28 32 34 38 26 26 19 31 25 38 35 26 16 23 35
## [4951] 7 13 28 32 37 31 23 23 27 30 24 31 34 27 5 35 32 22
## [4969] 24 32 32 32 35 12 20 27 15 30 25 33 27 28 26 24 28 27
## [4987] 26 24 28 30 21 17 26 16 26 18 20 28 28 23 25 25 29 25
## [5005] 22 27 27 23 19 31 11 24 20 26 19 18 21 30 25 25 30 27
## [5023] 18 27 21 25 20 18 28 21 21 15 19 11 15 20 14 23 25 15
## [5041] 12 17 24 12 25 25 29 18 28 26 21 21 22 19 25 17 25 23
## [5059] 19 18 16 24 26 20 23 19 18 24 7 22 18 25 21 27 26 22
## [5077] 26 18 12 118 19 27 27 26 25 23 24 25 27 22 13 16 68 26
## [5095] 14 14 16 20 15 18 14 19 19 25 105 19 20 62 22 20 21 16
## [5113] 15 7 15 24 24 18 24 23 14 17 13 19 21 20 22 21 21 17
## [5131] 18 19 12 15 25 16 22 21 18 14 16 20 12 20 20 21 22 15
## [5149] 11 10 13 12 11 24 23 18 12 17 12 25 24 25 25 11 10 10
## [5167] 19 16 15 6 22 14 18 19 21 15 12 21 14 18 16 19 18 13
## [5185] 20 15 9 18 18 9 123 15 11 13 12 23 13 12 12 17 10 19
## [5203] 5 14 15 8 15 17 10 19 15 13 15 11 15 9 21 9 19 7
## [5221] 16 18 15 17 14 20 20 21 11 15 18 10 9 19 9 10 8 18
## [5239] 10 12 7 9 9 19 16 13 13 12 15 8 15 11 15 17 15 7
## [5257] 17 18 9 8 15 6 16 12 8 15 18 18 9 14 17 17 8 18
## [5275] 11 15 14 5 28 14 10 15 11 18 13 7 13 13 6 16 15 7
## [5293] 17 5 6 18 15 7 9 9 18 17 10 9 18 3 17 14 9 7
## [5311] 9 17 7 5 22 4 15 14 15 10 33 12 14 16 9 12 6 15
## [5329] 11 9 13 23 7 14 4 14 16 13 14 6 5 8 12 16 10 8
## [5347] 9 9 12 15 16 12 12 3 7 10 12 3 3 12 7 9 14 7
## [5365] 7 7 8 14 4 8 5 7 9 6 11 6 10 10 4 8 9 8
## [5383] 15 5 11 10 5 7 4 14 10 6 15 6 14 7 4 3 9 14
## [5401] 5 7 8 7 13 12 13 8 9 8 16 4 8 10 4 12 13 10
## [5419] 3 4 11 12 6 9 11 8 5 13 7 15 8 7 10 5 4 4
## [5437] 7 6 7 10 14 8 13 5 13 5 5 14 16 13 10 13 6 7
## [5455] 11 4 12 5 21 26 3 7 12 9 12 7 6 12 4 2 2 2

```

```
## [5473]  1  13  13  10  7
##
## $chosenDistr
## [1] "norm"
##
## $fittedParams
##      mean      sd
## 141.0909  90.2261
```

## 4

Del punto 2 se observa que la distribución que mejor se ajuste corresponde a Weibull, pero del punto 3, al utilizar `fit.cont()`, podemos observar que la distribución Gompertz, presenta un valor de AIC menor al resto, de forma que resulta la más adecuada, pero ya que esta no formaba parte del inciso, se decide tomar la segunda mejor que corresponde a Weibull.

## 5

```
bootstrap <- bootstrapml(weibull, reps = 1000,
                        map = function(x) c(mean = mean(x), sd = sd(x)))

IC_mean <- c(bootstrap[1,1], bootstrap[1,2])
IC_sd <- c(bootstrap[2,1], bootstrap[2,2])

print(IC_mean)
```

```
## [1] 77.04197 79.92074
```

```
print(IC_sd)
```

```
## [1] 106.8551 110.8852
```

## Ejercicio 4

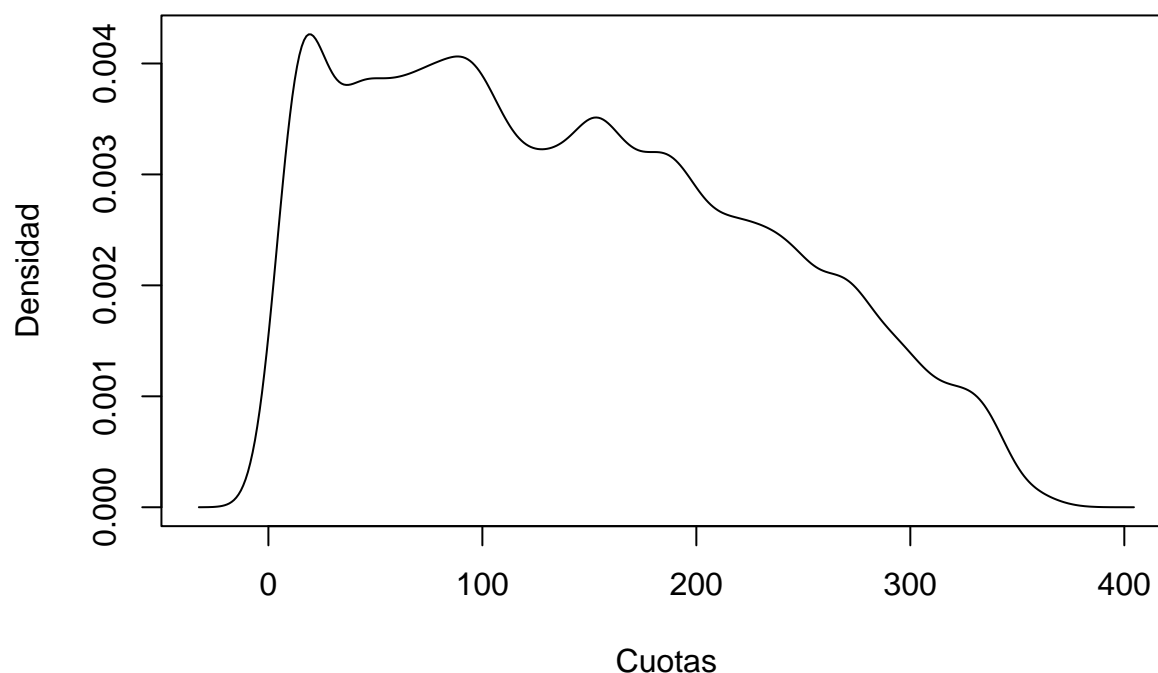
### 1

Del paquete `ks` de `r`, la función `kde` realiza una estimación de densidad mediante kernel, con bases de 1 a 6 variables, por lo que su la función nos facilita el estimar y visualizar la densidad de probabilidad de un conjunto de datos de manera no paramétrica, permitiendonos obtener una representación suave y continua de su distribución.

```
densidad_kde <- kde(base_salarios$Coutas)

plot(densidad_kde, xlab = "Cuotas", ylab = "Densidad",
     main = "Estimación de Densidad de Cuotas")
```

## Estimación de Densidad de Cuotas



2

Del paquete `boot` de `r`, la función `boot.ci` calcula 5 intervalos de confianza distintos, no paramétricos (la aproximación normal de primer orden, el intervalo bootstrap básico, el intervalo bootstrap estudentizado, el intervalo de percentil bootstrap y el intervalo de percentil bootstrap ajustado (BCa)), basados en el método de bootstrap.

3

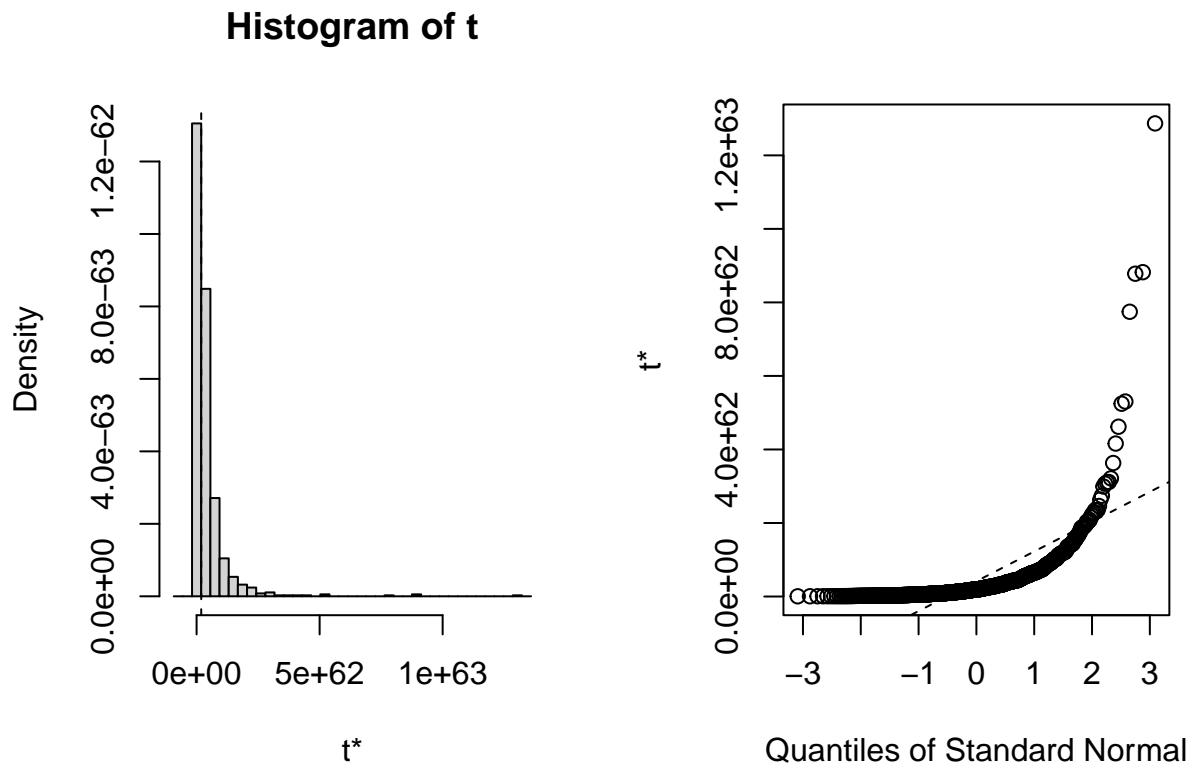
```
# Función para calcular la media

# Bootstrap de la media de Cuotas
result <- boot(data = base_salarios$Coutas,
               statistic=function(y,indices) exp(mean(y[indices])),
               R = 1000)

# Media original
media_original <- result$t0

# Media bootstrap (t)
media_bootstrap <- result$t
```

```
plot(result)
```



```
# Resultados
cat("Media original:", media_original, "\n")
```

```
## Media original: 1.883695e+61
```

```
print("Head de la media Bootstrap:")
```

```
## [1] "Head de la media Bootstrap:"
```

```
pander(head(media_bootstrap))
```

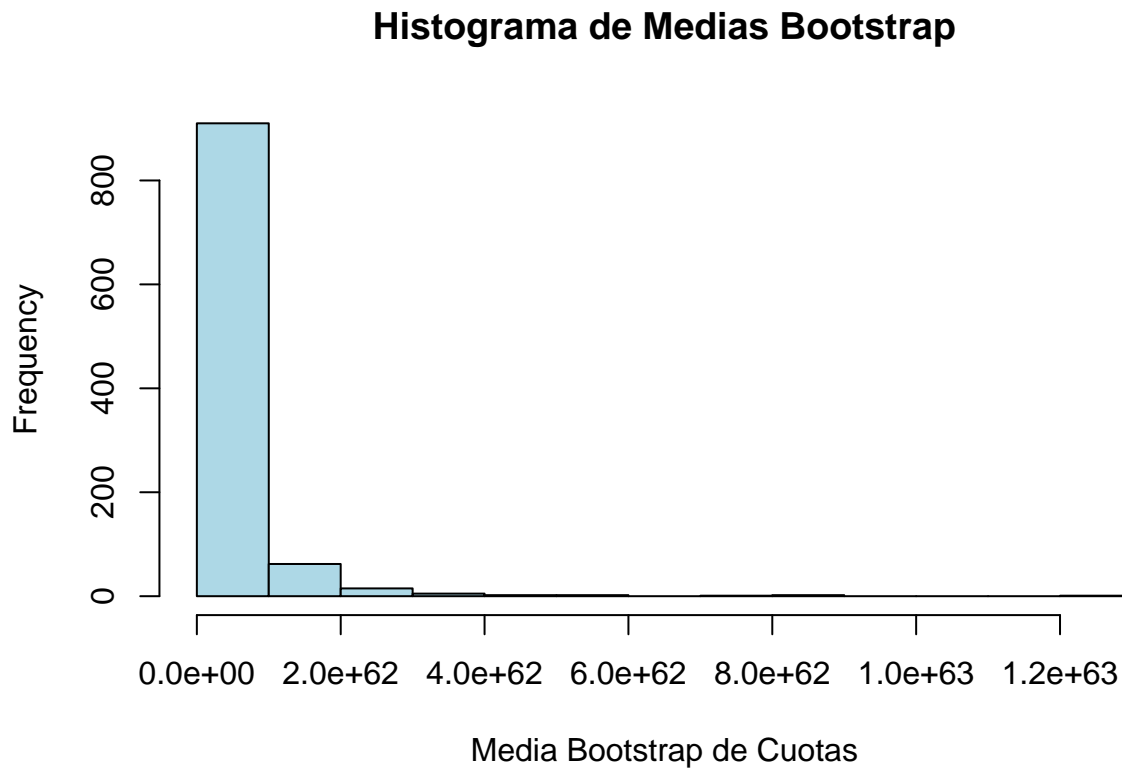
---

```
4.989e+60
1.246e+61
1.172e+62
4.412e+60
6.555e+61
1.331e+60
```

---



```
# Histograma de las medias bootstrap
hist(result$t,
      main = "Histograma de Medias Bootstrap",
      xlab = "Media Bootstrap de Cuotas",
      col = "lightblue",
      border = "black")
```



## Referencias

Angelo Canty(2024) Package ‘boot’: <https://cran.r-project.org/web/packages/boot/boot.pdf>

bandwidth function - RDocumentation. (s.f.). <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/bandwidth>

JMP Statistical Discovery LLC (2024) Diagrama de caja: [https://www.jmp.com/es\\_es/statistics-knowledge-portal/exploratory-data-analysis/box-plot.html](https://www.jmp.com/es_es/statistics-knowledge-portal/exploratory-data-analysis/box-plot.html)

Tarn Duong(2024) Package ‘ks’: <https://cran.r-project.org/web/packages/ks/ks.pdf>