Tarea 1 Simulación

Bernardo Mondragón Brozon

August 28, 2018

Problema 1

Sea X el número de 'unos' obtenido en 12 lanzamientos de un dado honesto. Entonces X tiene una distribución binomial. Calcular una tabla con los valores de la función de distribución para x = 0,1,...,12 por dos métodos: usando la función cumsum y usando la función pbinom. También determinar cuánto vale P(X > 7).

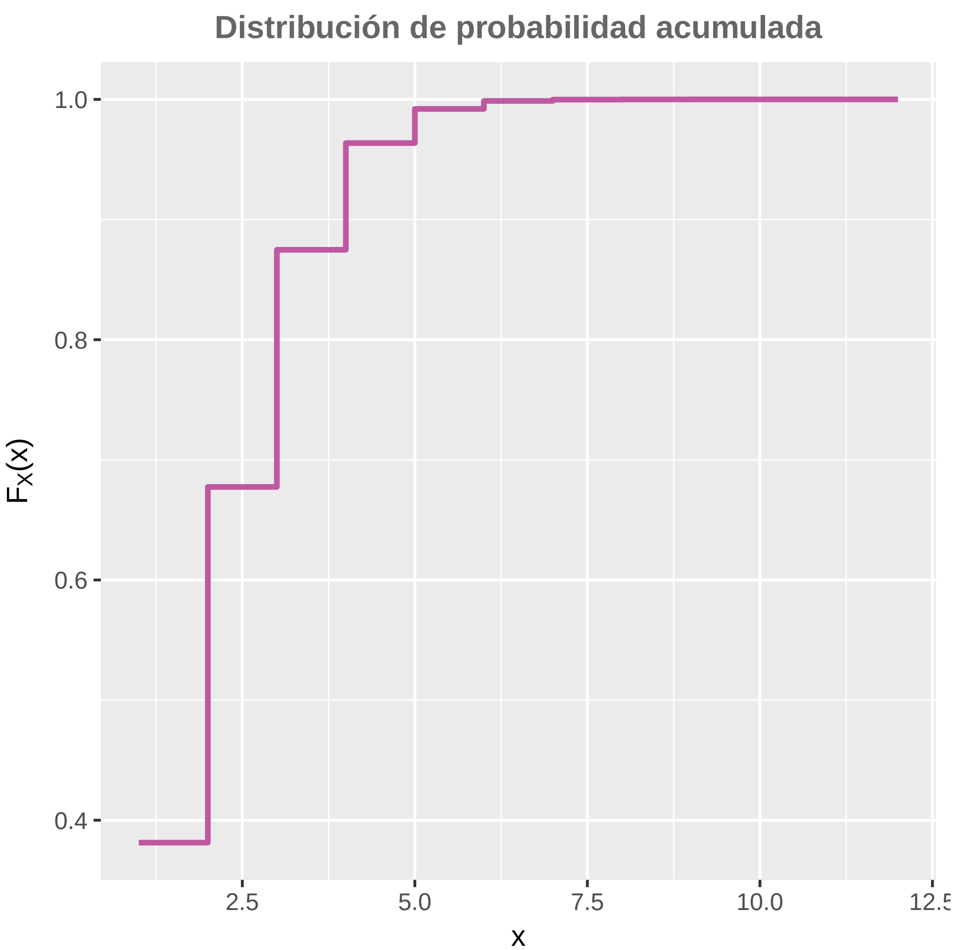
Solución

```
x <- c(1,2,3,4,5,6,7,8,9,10,11,12)
probs <- pbinom(q=x, size=12, prob=1/6)
table <- data.frame(x, probs)
names(table) <- c("x", "Probabilidad acumulada")
kable(table, caption="Distribución de X")
```

Table 1: Distribución de X

x	Probabilidad acumulada
1	0.3813326
2	0.6774262
3	0.8748219
4	0.9636500
5	0.9920750
6	0.9987075
7	0.9998445
8	0.9999866
9	0.9999992
10	1.0000000
11	1.0000000
12	1.0000000

```
ggplot(data = table, aes(x=table[,1], y=table[,2])) +
  geom_step(color=primary, size=1) +
  labs(x="x", y=TeX('$F_X(x)$')) +
  ggtitle("Distribución de probabilidad acumulada") +
  theme(plot.title = element_text(
    hjust = 0.5,
    color = "#666666",
    face = "bold",
    size = 12)
  )
```



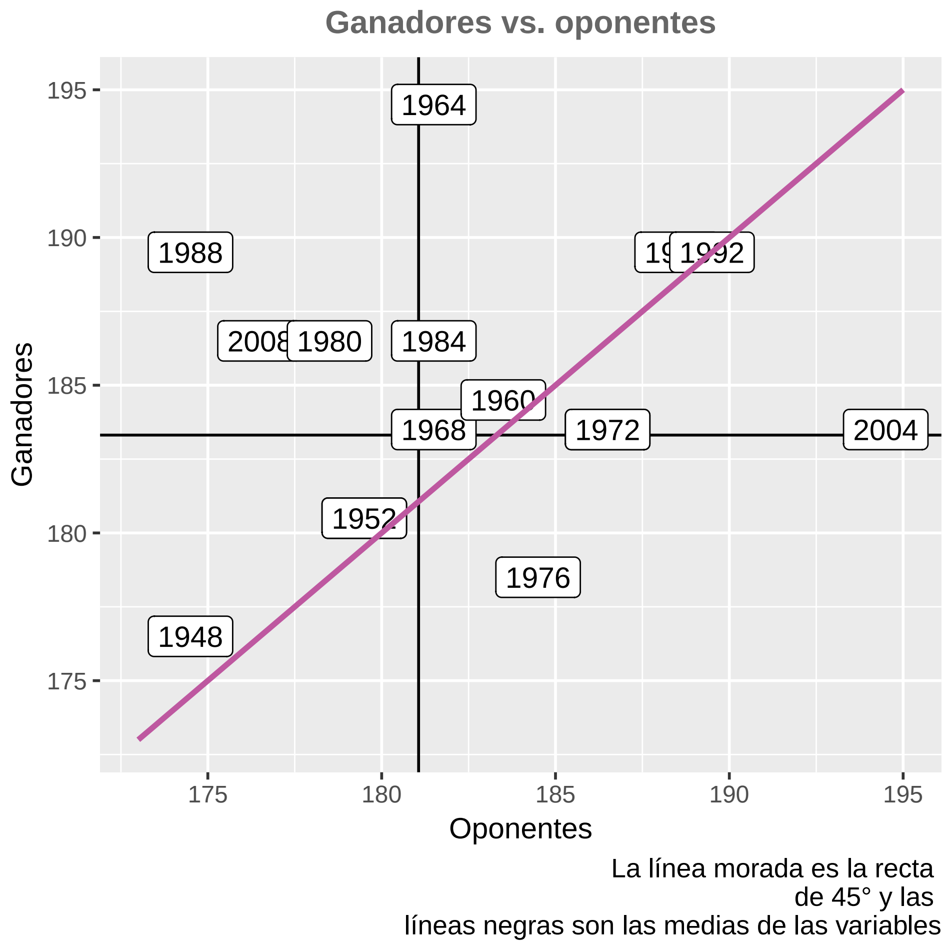
Problema 2

(Estaturas de presidentes gringos). En un artículo de Wikipedia, se reportan las estaturas de los Presidentes de los Estados Unidos y los de sus oponentes en elecciones. Se ha notado que mientras más alto seal el presidente típicamente gana la elección. Hagan una gráfica de dispersión de puntos con la estatura del perdedor vs. el ganador.

Solución

```
df <- read.csv("../HeightData.csv", sep=",", header = TRUE)
trim <- function (x) gsub("^\\s+|\\s+$", "", x)
getHeight <- function(Height) {
  res1 <- strsplit(x=as.character(Height), split="in")
  centimeters <- res1[1][[1]][2]
  res2 <- strsplit(x=centimeters, split="cm")
  number <- as.character(trim(res2))
  return(as.numeric(number))
}
winners_heights <- c()
losers_heights <- c()
for (i in 1:length(df$Height)) {
```

```
  height_w <- getHeight(df[i,]$Height)
  winners_heights <- c(winners_heights, height_w)
  height_l <- getHeight(df[i,]$Height.1)
  losers_heights <- c(losers_heights, height_l)
}
df <- data.frame(df$Year, df$Winner, winners_heights, df$Opponent, losers_heights)
names(df) <- c("Year", "Winner", "W.Height", "Opponent", "O.Height")
scatter_plot <- ggplot(data=df, mapping=aes(x=O.Height, y=W.Height)) +
  #geom_point(color=primary) +
  geom_hline(aes(yintercept=mean(df$W.Height))) +
  geom_vline(aes(xintercept=mean(df$O.Height))) +
  geom_label(label=df$Year, nudge_x = 1.5, nudge_y = 1.5) +
  stat_function(fun=function(x) {x}, size=1, color=primary) +
  ylim(c(173, 195)) +
  xlim(c(173, 195)) +
  labs(x="Oponentes", y="Ganadores", caption="La línea morada es la recta
       de 45° y las \n líneas negras son las medias de las variables") +
  ggtitle("Ganadores vs. oponentes") +
  theme(plot.title = element_text(
    hjust = 0.5,
    color = "#666666",
    face = "bold",
    size = 12))
scatter_plot
```



Problema 3

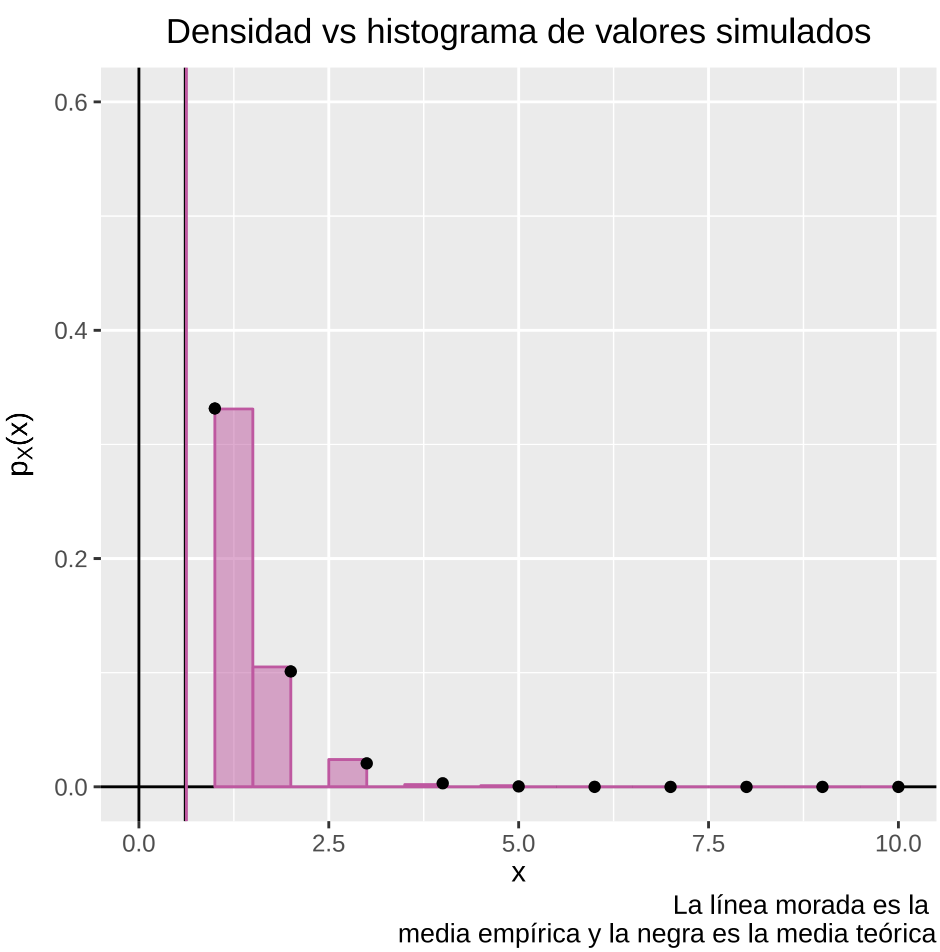
La función rpois genera observaciones aleatorias de una distribución Poisson. Usen la función rpois para simular un número grande (n=1000 y n=1000) muestras Poisson con parámetro $\lambda = 0.61$. Encuentren la función de masa de probabilidad, media, y varianza para las muestras. Comparen con los valores teóricos de la densidad Poisson.

Solución

Con n = 1000

```
obs_1 <- rpois(n=1000, lambda=0.61)
media_1 <- mean(obs_1)
variance_1 <- var(obs_1)
df_1 <- data.frame(obs_1)
vals <- dpois(seq(1, 10, by=1), lambda=0.61)
xs <- seq(1, 10, by=1)
df_pois <- data.frame(vals, xs)
hist_vs_den <- ggplot() +
  geom_hline(yintercept = 0, size = .5) +
```

```r
  geom_vline(xintercept = 0, size = .5) +
  labs(x = TeX('$x$'), y = TeX('$p_X(x)$'), caption="La línea morada es la
       media empírica y la negra es la media teórica") +
  ggtitle("Densidad vs histograma de valores simulados") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_vline(aes(xintercept=0.61)) +
  geom_vline(aes(xintercept=mean(obs_1)), color=primary) +
  xlim(0,10) +
  ylim(0,0.6) +
  geom_histogram(data=df_1, aes(x=obs_1, y=.001*..count..),
                 breaks=seq(1, 10, by=0.5),
                 color=primary, fill=primary, alpha=0.5) +
  geom_point(data=df_pois, aes(x=xs, y=vals))
hist_vs_den
```



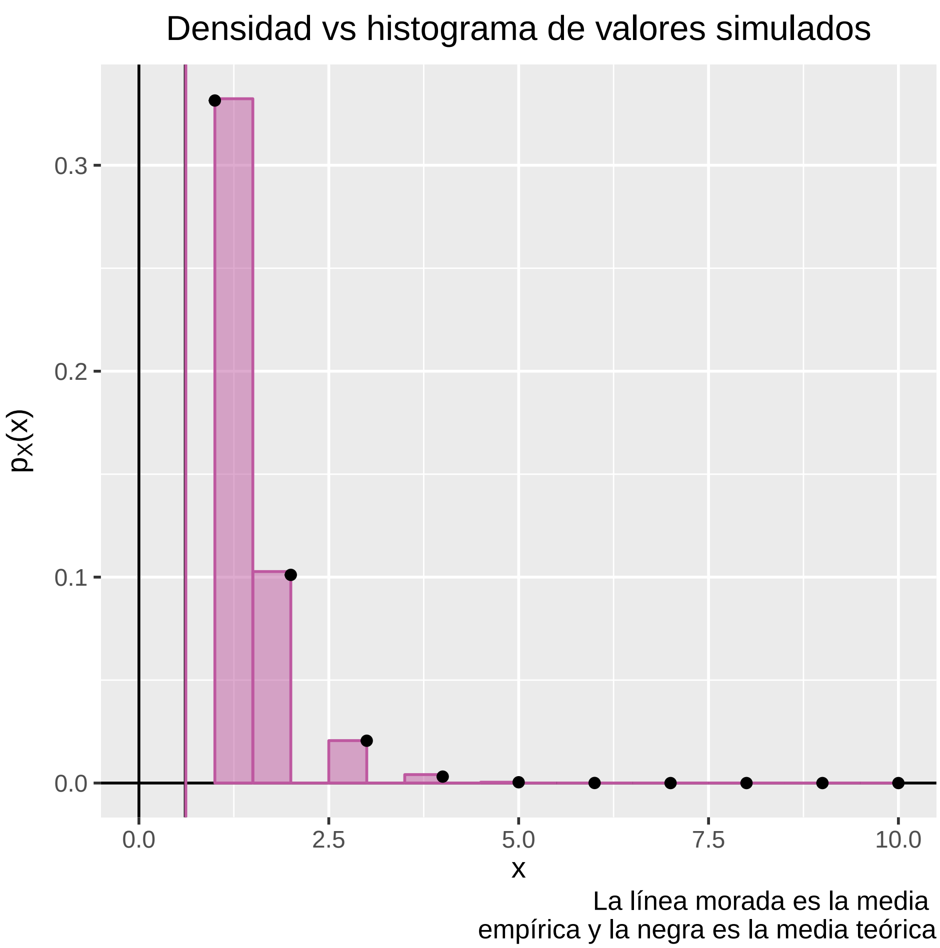

```r
tablita <- data.frame(
  c("Media", "Varianza"),
  c(mean(obs_1), var(obs_1)), c(0.61, 0.61))
names(tablita) <- c(" ", "Simulación", "Teóricos")
kable(tablita, caption="Comparación de los valores
      teóricos con los obtenidos por simulación")
```

Table 2: Comparación de los valores teóricos con los obtenidos por simulación

	Simulación	Teóricos
Media	0.6260000	0.61
Varianza	0.6327568	0.61

Con n = 10000

```
obs_2 <- rpois(n=10000, lambda=0.61)
media_2 <- mean(obs_2)
variance_2 <- var(obs_2)
df_2 <- data.frame(obs_2)
vals_2 <- dpois(seq(1, 10, by=1), lambda=0.61)
xs_2 <- seq(1, 10, by=1)
df_pois_2 <- data.frame(vals_2, xs_2)
hist_vs_den_2 <- ggplot() +
  geom_hline(yintercept = 0, size = .5) +
  geom_vline(xintercept = 0, size = .5) +
  labs(x = TeX('$x$'), y = TeX('$p_X(x)$'), caption="La línea morada es la media
       empírica y la negra es la media teórica") +
  ggtitle("Densidad vs histograma de valores simulados") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_vline(aes(xintercept=0.61)) +
  geom_vline(aes(xintercept=mean(obs_2)), color=primary) +
  # xlim(0,10) +
  # ylim(0,20) +
  geom_histogram(data=df_2,
                 aes(x=obs_2, y=.0001*..count..),
                 breaks=seq(1, 10, by=0.5), color=primary,
                 fill=primary, alpha=0.5) +
  geom_point(data=df_pois_2, aes(x=xs_2, y=vals_2))
hist_vs_den_2
```




```
tablita_2 <- data.frame(c("Media", "Varianza"), c(mean(obs_2), var(obs_2)), c(0.61, 0.61))
names(tablita_2) <- c(" ", "Simulación", "Teóricos")
kable(tablita_2, caption="Comparación de los valores
      teóricos con los obtenidos por simulación")
```

Table 3: Comparación de los valores teóricos con los obtenidos por simulación

	Simulación	Teóricos
Media	0.6179000	0.61
Varianza	0.6223618	0.61

Problema 4

Escriban una función en R llamada sd.n que regrese el valor estimado de $\hat{\sigma}$ de una muestra de tamaño n, utilizando la fórmula del estimado máximo verosímil de la varianza.

Solución

```
sd.n <- function(muestra) {
  aux <- 0
```

```
  for (i in 1:length(muestra)){
    aux <- aux + muestra[i]
  }
  return(aux/length(muestra))
}

muestra <- rpois(n = 100, lambda = 0.61)
sd.n(muestra)
```

```
## [1] 0.55
```

Problema 5

Escriban una función norma que calcule la norma Euclideana de un vector numérico de longitud n. Evaluar la norma de los vectores (0,0,0,1), (2,5,2,4) y (1,2,3,4,5,6,7,8,9,10).

Solución

```
vectorToString <- function(vec) {
  str <- paste("(", vec[1], sep="")
  for (i in 2:length(vec)) {
    str <- paste(str, ", ", vec[i], sep="")
  }
  str <- paste(str, ")", sep="")
  return(str)
}
norma <- function(vec) sqrt(sum(vec^2))
vec1 <- c(0,0,0,1)
vec2 <- c(2,5,2,4)
vec3 <- c(1,2,3,4,5,6,7,8,9,10)
vecData <- data.frame(
  vectors=c(vectorToString(vec1), vectorToString(vec2), vectorToString(vec3)),
  norms=c(norma(vec1), norma(vec2), norma(vec3)))
names(vecData) <- c("Vector", "Norma")
kable(vecData)
```

Vector	Norma
(0, 0, 0, 1)	1.00000
(2, 5, 2, 4)	7.00000
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	19.62142

Problema 6

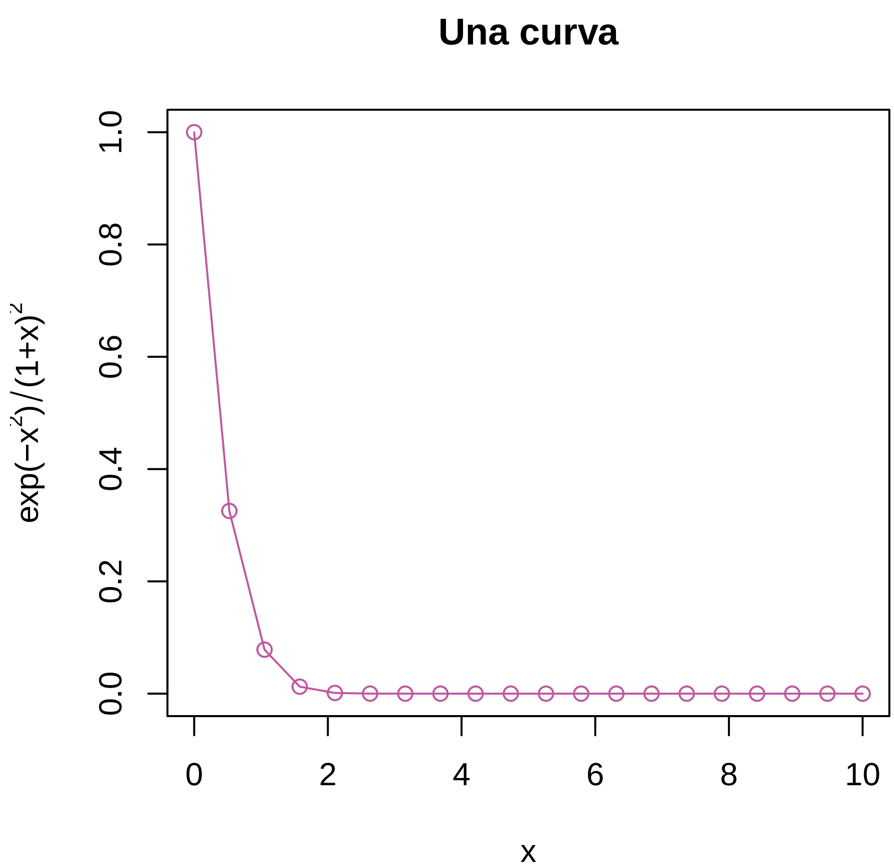
Usar la función curve para graficar la función f(x)=e-x2/(1+x2) en el intervalo $[0, 10]$. Luego usar la función integrate para calcular el valor de la integral

$$\int_0^\infty \frac{e^{-x^2}}{1+x^2}\mathrm{dx}$$

El límite superior se especifica usando el argumento upper=Inf en la función integrate.

Solución

```
myFunction <- function(x) exp(-x^2)/(1+x)^2
curve(myFunction,
      from=0, to=10,
      n=20, type="o",
      col=primary,
      xlab="x",
      ylab=TeX('exp(-x^2)/(1+x)^2'),
      main="Una curva")
```




```
integrate(myFunction,lower=0, upper=Inf)
```

```
## 0.4378135 with absolute error < 3e-05
```

Problema 7

Construir una matriz con 10 renglones y 2 columnas que contienen datos provenientes de una normal estándar:

$$x < -matrix(rnorm(20), 10, 2)$$

Esta es una muestra de 10 observaciones de una distribución normal bivariada. Usen la función apply y la función norma que crearon en un ejercicio anterior para calcular las normas euclideanas para cada una de las

10 observaciones.

Solución

```
x <- matrix(data=rnorm(n=20), nrow = 10, ncol = 2)
apply(x, 1, norma)
```

```
##  [1] 2.2852822 1.1899602 3.2868903 1.1045879 0.9049489 1.3203416 1.2814011
##  [8] 0.2574689 1.0576316 0.9226015
```

Problema 8

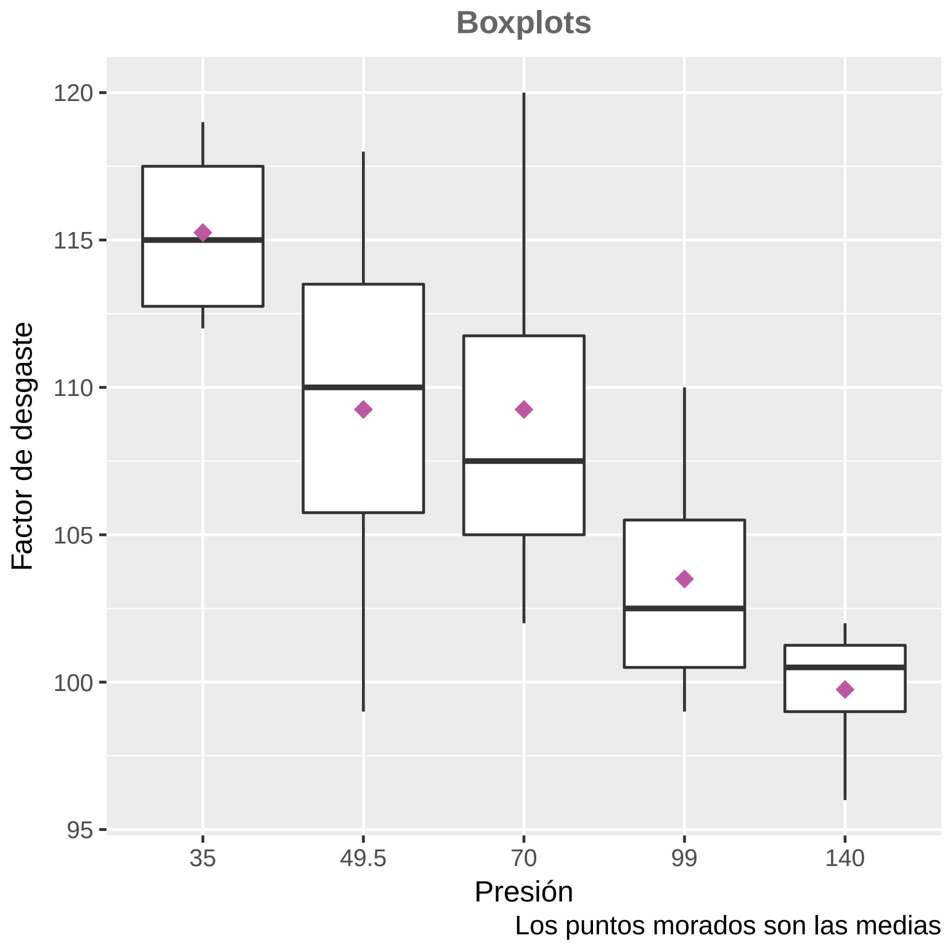
Los siguientes datos describen el factor de desgaste de papel manufacturado bajo diferentes presiones durante el prensado. Cuatro hojas de papel fueron seleccionadas y probadas para cada uno de los cinco lotes manufacturados:

Presión (lotes)	Factor de resistencia (hojas)
35.0	112 119 117 113
49.5	108 99 112 118
70.0	120 106 102 109
99.0	110 101 99 104
140.0	100 102 96 101

Metan estos datos en un dataframe con dos variables: factor de resistencia y presion. Hacer un boxplot para comparar los diferentes factores de resistencia para cada presión.

Solución

```
presion <- c(rep(35,4),rep(49.5,4),rep(70,4),rep(99,4),rep(140,4))
factorRes <- c(112,119,117,113,108,99,112,118,120,106,102,109,110,101,99,104,100,102,96,101)
sheetsData <- data.frame(presion,factorRes)
bxplt <- ggplot(data=sheetsData, aes(x = as.factor(presion), y = factorRes)) +
  geom_boxplot() +
  labs(x = "Presión", y = "Factor de desgaste",
       caption="Los puntos morados son las medias") +
  theme(plot.title = element_text(
    hjust = 0.5,
    color = "#666666",
    face = "bold",
    size = 12)) +
  stat_summary(fun.y = mean, colour = primary, geom = "point", shape = 18, size = 3) +
  ggtitle("Boxplots")
bxplt
```


Boxplots
120
115
110
105
100
95
Factor de desgaste
35
49.5
70
99
140
Presión
Los puntos morados son las medias