Proyecto Beisbol en R

Universidad Simón Bolivar
Estadística para matemáticos-CO3322
Profesor: Pedro Ovalles
Enero-Marzo 2022

Eduardo Gavazut
Luis Riera
Miguel Cordero

Planteamiento del problema

Descripción de los datos

Realizar un análisis descriptivo de los datos

```r
library(readxl)
Baseball <- read_excel("~/GitHub/data/Baseball.xlsx")
head(Baseball, n=5)
```

```
# A tibble: 5 x 6
     X1    X2    X3    X4    X5    X6
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 0.283 0.144 0.049 0.012 0.013 0.086
2 0.276 0.125 0.039 0.013 0.002 0.062
3 0.281 0.141 0.045 0.021 0.013 0.074
4 0.328 0.189 0.043 0.001 0.03  0.032
5 0.29  0.161 0.044 0.011 0.07  0.076
```

¿Qué clase es la base de datos?

```r
class(Baseball)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

Por los resultados obtenidos con la ayuda de la función `class` de `R`, la base de datos es del tipo `tbl_df`, que es una subclase de la clase `data.frame`. `tbl_df` cumple con tener propiedades diferentes por defecto y se suele referir a ellas como `tibble`. Es una clase eficiente para trabajar con bases de datos grandes y su visualización.

Utilice el comand `str` para explicar las variables y el tipo de cada variable de la base de datos

```r
str(Baseball)
```

```
tibble [45 x 6] (S3: tbl_df/tbl/data.frame)
 $ X1: num [1:45] 0.283 0.276 0.281 0.328 0.29 0.296 0.248 0.228 0.305 0.254 ...
 $ X2: num [1:45] 0.144 0.125 0.141 0.189 0.161 0.186 0.106 0.117 0.174 0.094 ...
 $ X3: num [1:45] 0.049 0.039 0.045 0.043 0.044 0.047 0.036 0.03 0.05 0.041 ...
```

```
 $ X4: num [1:45] 0.012 0.013 0.021 0.001 0.011 0.018 0.008 0.006 0.008 0.005 ...
 $ X5: num [1:45] 0.013 0.002 0.013 0.03 0.07 0.05 0.012 0.003 0.061 0.014 ...
 $ X6: num [1:45] 0.086 0.062 0.074 0.032 0.076 0.007 0.095 0.145 0.112 0.124 ...
```

Se tienen 6 variables, `X1,X2,X3,X4,X5,X6`. Cada una con 45 valores de tipo `double` o número decimal, que representan las 45 observaciones aleatorias realizadas a jugadores de la (MLB).

Cada variable representa la siguiente información:

- `X1`: tasa de bateo, en (hit/veces al bate).
- `X2`: carreras anotadas/veces al bate.
- `X3`: dobles/ veces al bate.
- `X4`: tripes/ veces al bate.
- `X5`: jonrones/ veces al bate.
- `X6`: ponches/ veces al bate.

Calcule por cada variable los estadisticos para cada variable numérica

Para obtener los estadísticos de las seis (6) variables de esta base de datos, se inicia por guardar las 45 observaciones en un vector que represente a cada variable:

```
X1<- Baseball$X1
X2<- Baseball$X2
```

Con los datos vectorizados y con apoyo de las funciones de `R` obtener las siguiente información

```
# Media
media <- c(mean(X1), mean(X2))
# Mediana
mediana<- c(median(X1), median(X2))
# Cuartile 1: 25%
q1 <- c(quantile(X1,0.25), quantile(X2,0.25))
# Cuartile 2: 50% = Mediana
q2 <- c(quantile(X1,0.5), quantile(X2,0.5))
# Cuartile 3: 75%
q3 <- c(quantile(X1,0.75), quantile(X2,0.75))
# Rango Intercuartile
ric <- c(IQR(X1), IQR(X2))
# Varianza
varianza <- c(var(X1), var(X2))
# Desviación estándar
stad <- c(sd(X1), sd(X2))
# Coeficiente de variación
coef_var <-  stad/media
```

Guardamos los datos en un array

```
# Unimos los valores obtenidos
estadisticos <- cbind(round(q1, digits = 4), round(media, digits=4),
                      round(mediana,digits=4), round(q2, digits=4),
                      round(ric, digits=4), round(q3, digits=4),
                      round(varianza, digits=4), round(stad, digits=4),
                      round(coef_var, digits=4))
# Definimos los nombres de las columnas y filas
rownames(estadisticos) <- c("X1", "X2")
colnames(estadisticos) <- c("25%", "Media", "Mediana", "50%","75%", "RIC",
                            "Varianza", "Desv. Estándar","Coef. variación")
```

```r
# Mostramos el arreglo
estadisticos
```

```
     25%  Media Mediana  50%  75%   RIC Varianza Desv. Estándar Coef. variación
X1 0.248 0.2805    0.29 0.29 0.06 0.308   0.0019          0.044          0.1569
X2 0.119 0.1509    0.15 0.15 0.07 0.189   0.0018          0.042          0.2784
```

Elabora diagramas de caja para cada conjunto de datos

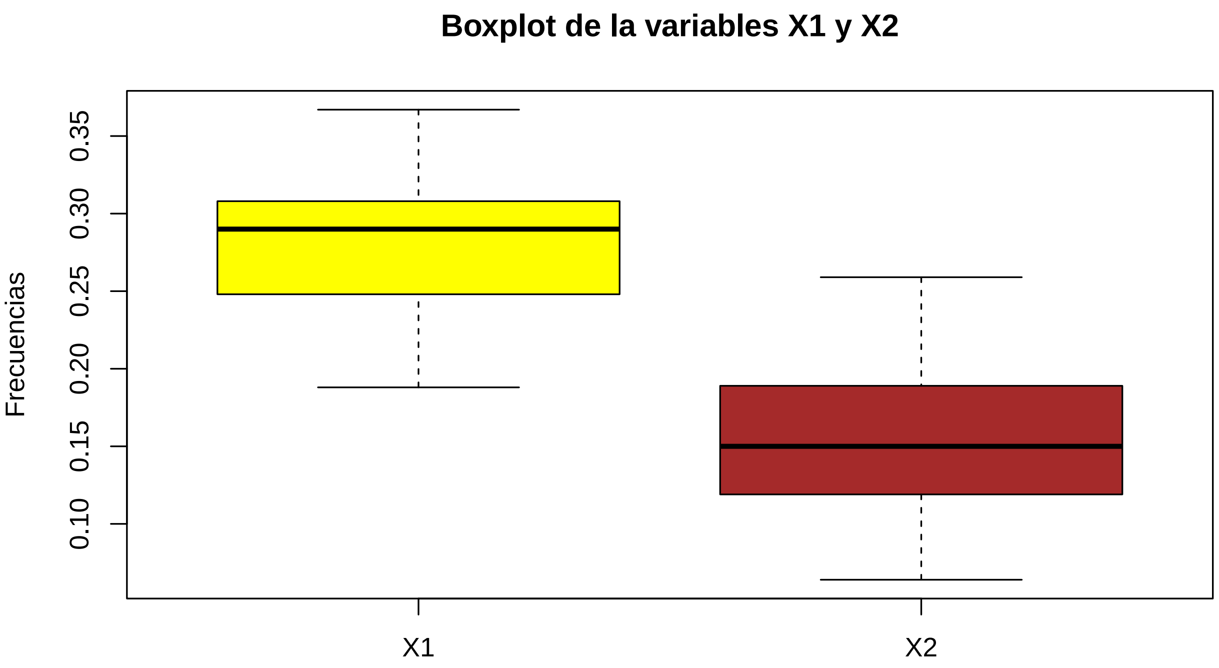

Figura 1: Diagrama de caja para las variables X1 y X2

Intervalo de confianza del 97% para la media

Por los histogramas obtenidos en ??, se puede suponer que la variables `X1` y `X2` siguen alguna de las siguientes distribuciones de probabilidad: Poisson, Normal, Gamma y Chi-Cuadrado.

Para comprobar cual de estas funciones se ajusta mejor a las variables se utiliza la función `fitdistr` de la librería `MASS`. Que calcula los parámetros de las distribuciones seleccionadas que hacen mímina la varianza entre los datos reales y los estimados utilizando tales parámetros.

```r
library(MASS) # Cargamos la libreria
```

Para X1

```
pois_fit_1 = fitdistr(X1,"poisson")
pois_fit_1
```

```
     lambda
  0.28046667
 (0.07894677)
```

```
chi_fit_1 = fitdistr(X1,"chi-squared",start=list(df=1))
chi_fit_1
```

```
      df
  0.9944336
 (0.1335736)
```

```
norm_fit_1 = fitdistr(X1,"normal")
norm_fit_1
```

```
      mean           sd
  0.280466667   0.043510203
 (0.006486118) (0.004586378)
```

```
gam_fit_1= fitdistr(X1,"gamma")
gam_fit_1
```

```
     shape         rate
   39.409111   140.512690
 (  8.273037) ( 29.685499)
```

Con estos estimadores, se crean 8 simulaciones por cada cada distribución y se escoge aquella que mejor se ajuste a los datos al comparar los histogramas con la variable objetivo:

```
# Se fija el seed para que se repliquen los datos en cada ejecución
set.seed(777)
sim_pois <- matrix(nrow = 8, ncol = 100) # Simulaciones con poisson
sim_chi <- matrix(nrow = 8, ncol = 100)  # Simulaciones con chi-cuadrado
sim_norm <- matrix(nrow = 8, ncol = 100) # Simulaciones con normal
sim_gam <- matrix(nrow = 8, ncol = 100)  # Simulaciones con gamma

# Se realizan las simulaciones en un loop for
for(i in 1:8){
  #Para poisson
  sim_pois[i,]=rpois(n = 100, lambda = pois_fit_1$estimate)
  #Para chi-cuadrada
  sim_chi[i,]=rchisq(n=100,df=chi_fit_1$estimate)
  #Para normal
  sim_norm[i,]=rnorm(n=100,mean=norm_fit_1$estimate[1],sd=norm_fit_1$estimate[2])
  #Para gamma
  sim_gam[i,]=rgamma(n=100,shape = gam_fit_1$estimate[1],rate=gam_fit_1$estimate[2])
}
```

Se comparan los histogramas de las simulaciones y el original

De estas simulaciones se pudo concluir que la distribución que mejor se adapta a los datos originales de la variable `X1`, es la distribución normal con media, $\mu = 0.280$ y desviación estándar, $\sigma = 0.044$.

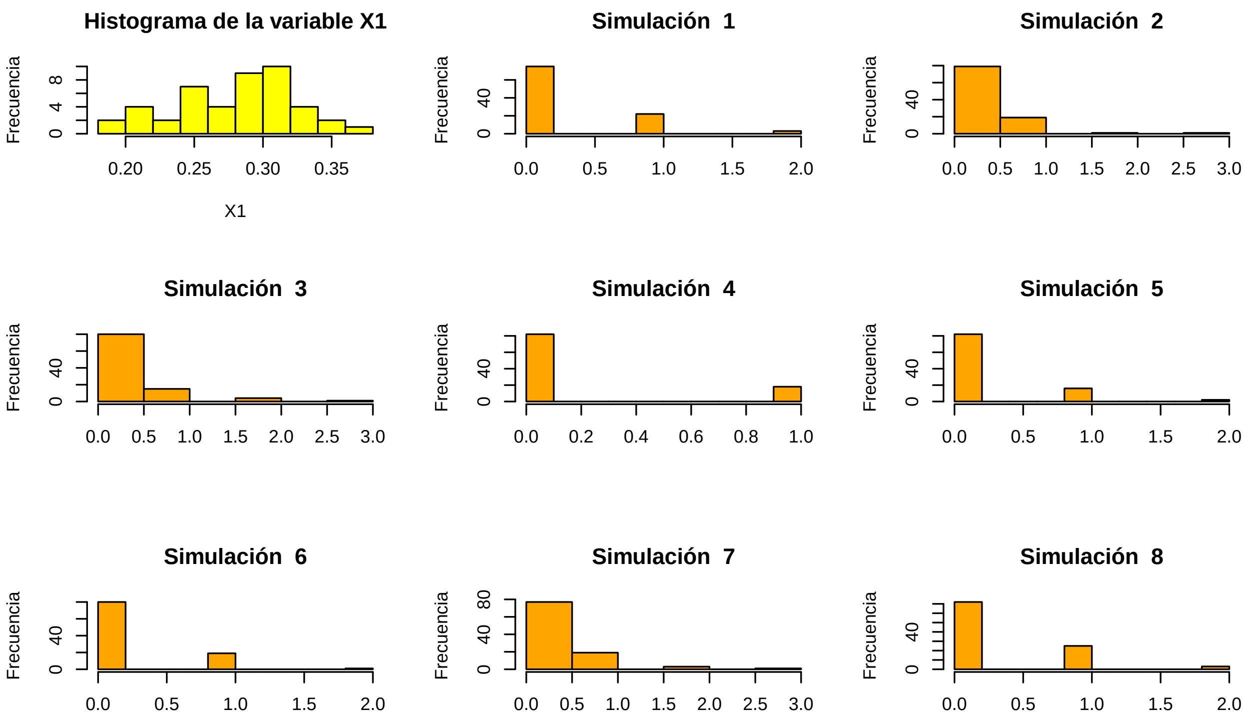

Figura 2: Comparación valores originales con 8 imulaciones de una distribución Poisson.

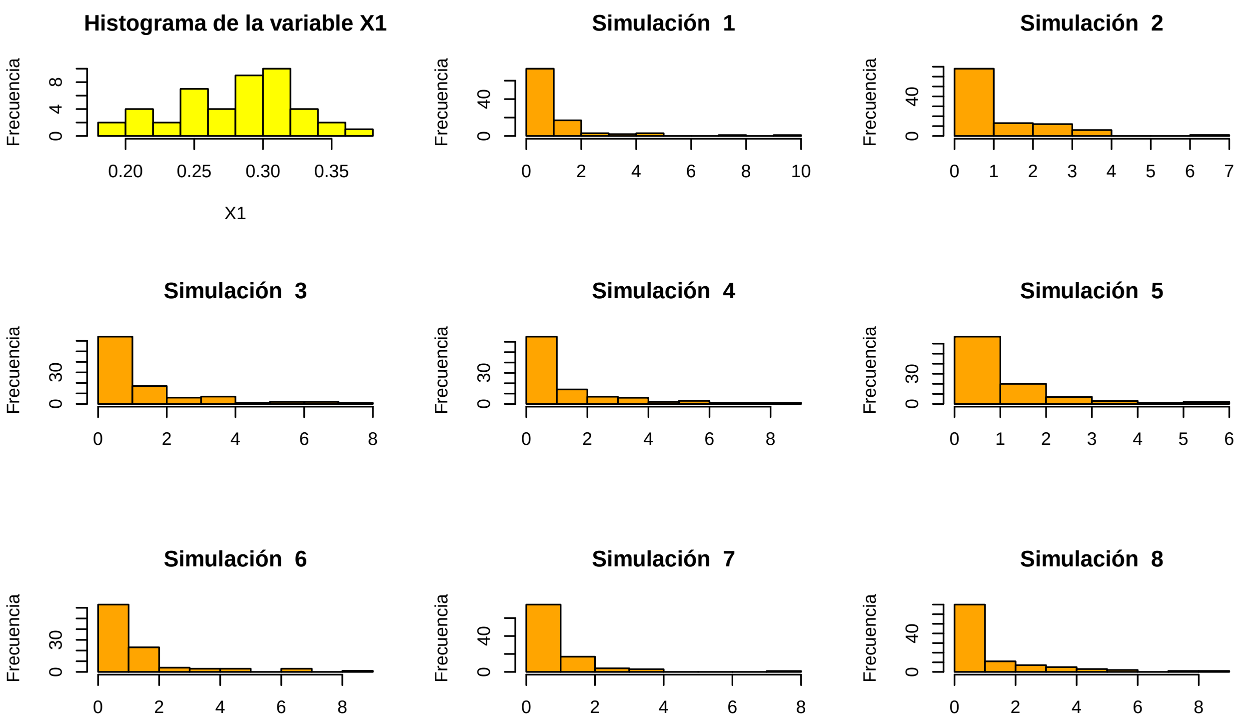

Figura 3: Comparación valores originales con 8 imulaciones de una distribución Chi-Cuadrado.

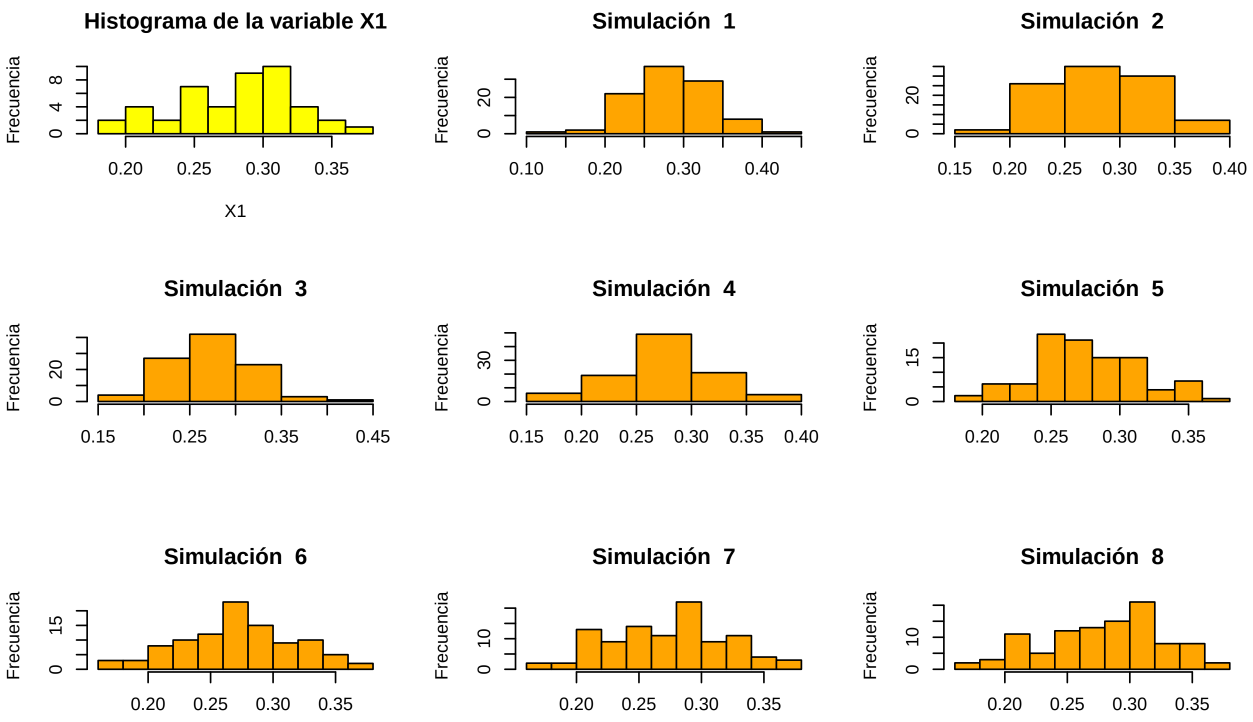

Figura 4: Comparación valores originales con 8 imulaciones de una distribución Normal.

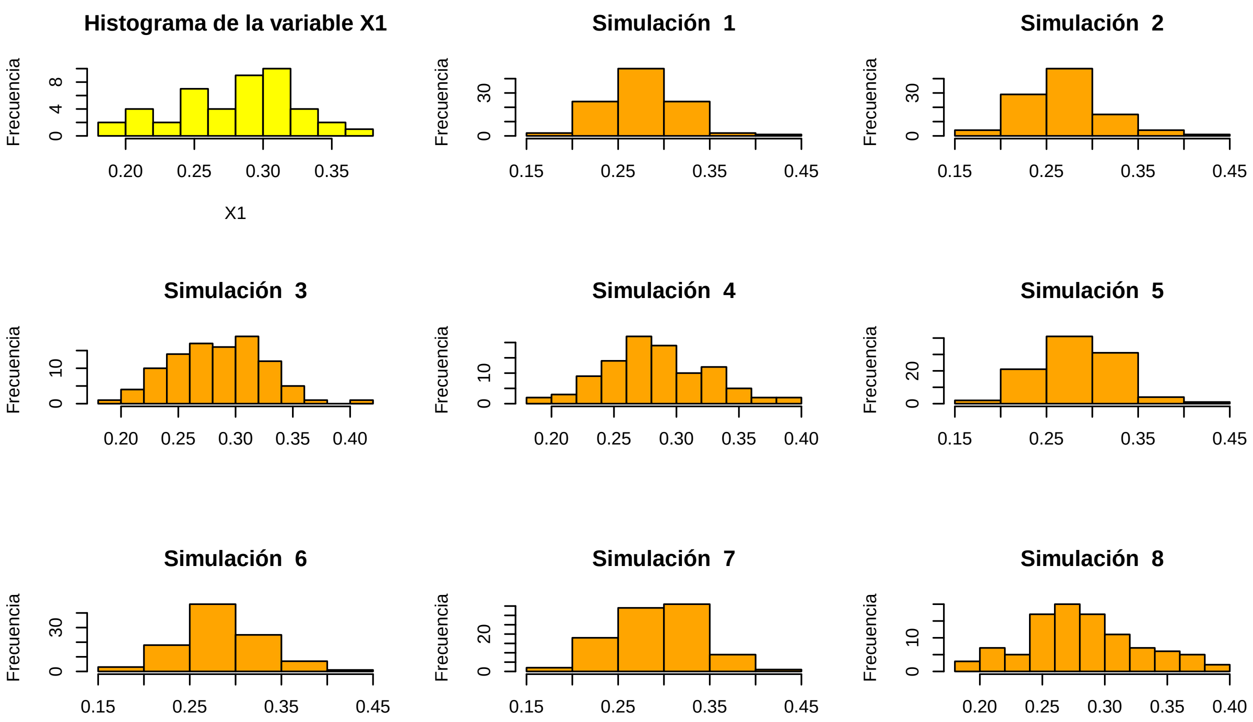

Figura 5: Comparación valores originales con 8 imulaciones de una distribución Gamma.

Para X2

```
pois_fit_2 = fitdistr(X2,"poisson")
pois_fit_2
```

```
     lambda
  0.15088889
 (0.05790584)
```

```
chi_fit_2 = fitdistr(X2,"chi-squared",start=list(df=1))
chi_fit_2
```

```
      df
  0.7826904
 (0.1084082)
```

```
norm_fit_2 = fitdistr(X2,"normal")
norm_fit_2
```

```
      mean           sd
  0.150888889   0.041539391
 (0.006192327) (0.004378636)
```

```
gam_fit_2= fitdistr(X2,"gamma")
gam_fit_2
```

```
     shape       rate
  12.283553   81.407930
 ( 2.555185) (17.284575)
```

Se crearon 8 simulaciones por cada cada distribución y se escoge la que mejor se ajuste a los datos al comparar los histogramas

```
# Se fija el seed para que se repliquen los datos en cada ejecución
set.seed(777)
sim_pois_2 <- matrix(nrow = 8, ncol = 100) # Simulaciones con poisson
sim_chi_2 <- matrix(nrow = 8, ncol = 100)  # Simulaciones con chi-cuadrado
sim_norm_2 <- matrix(nrow = 8, ncol = 100) # Simulaciones con normal
sim_gam_2 <- matrix(nrow = 8, ncol = 100)  # Simulaciones con gamma

# Se realizan las simulaciones en un loop for
for(i in 1:8){
  #Para poisson
  sim_pois_2[i,]=rpois(n = 100, lambda = pois_fit_2$estimate)
  #Para chi-cuadrada
  sim_chi_2[i,]=rchisq(n=100,df=chi_fit_2$estimate)
  #Para normal
  sim_norm_2[i,]=rnorm(n=100,mean=norm_fit_2$estimate[1],sd=norm_fit_2$estimate[2])
  #Para gamma
  sim_gam_2[i,]=rgamma(n=100,shape = gam_fit_2$estimate[1],rate=gam_fit_2$estimate[2])
}
```

Se comparan los histogramas de las simulaciones y el original

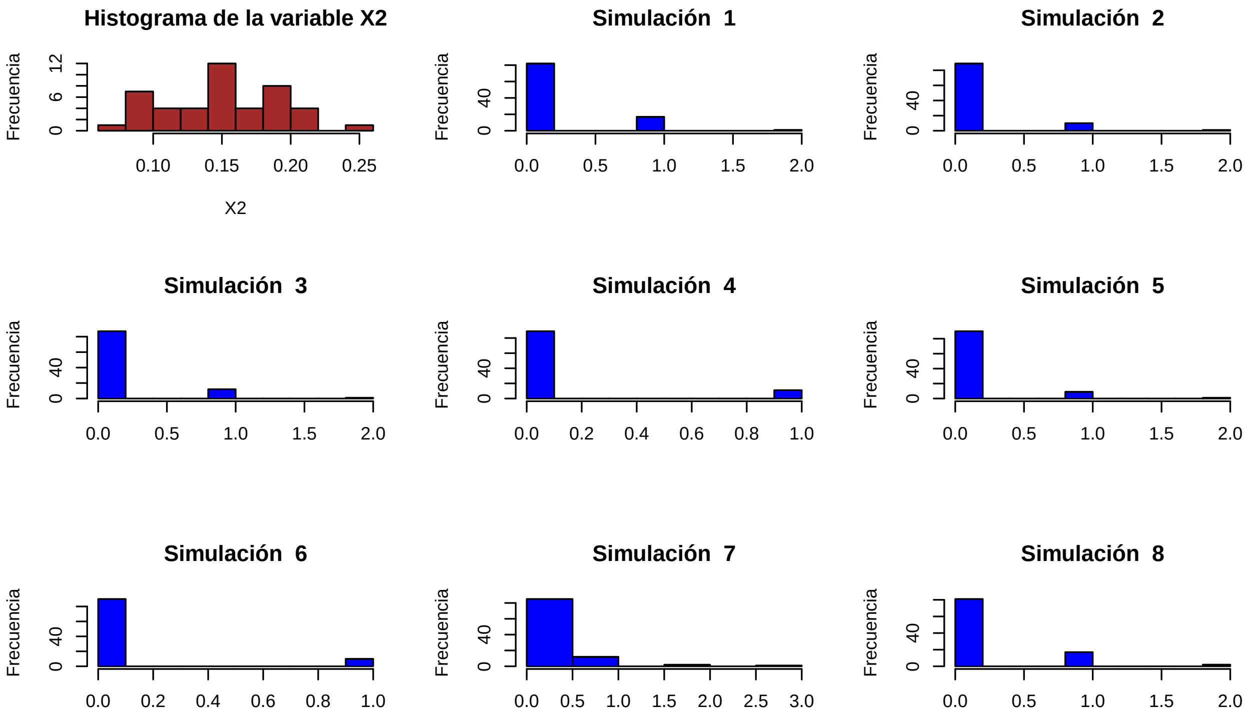

Figura 6: Comparación valores originales con 8 imulaciones de una distribución Poisson.

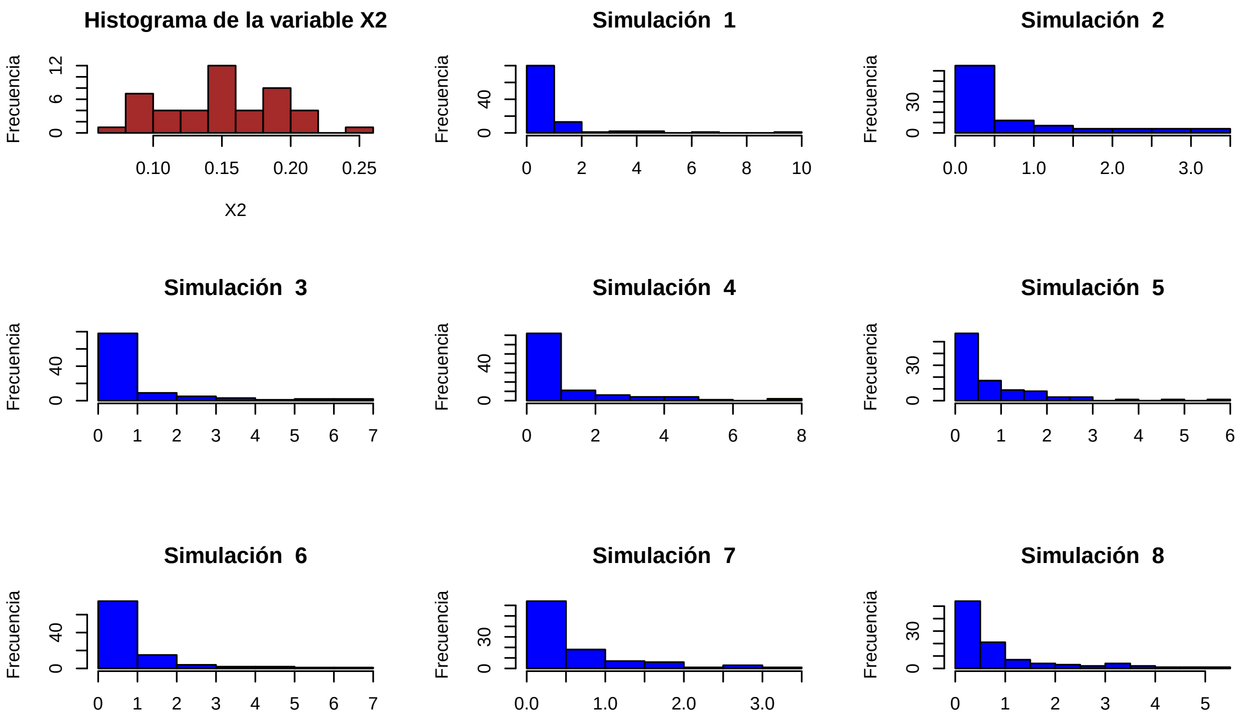

Figura 7: Comparación valores originales con 8 imulaciones de una distribución Chi-Cuadrado.

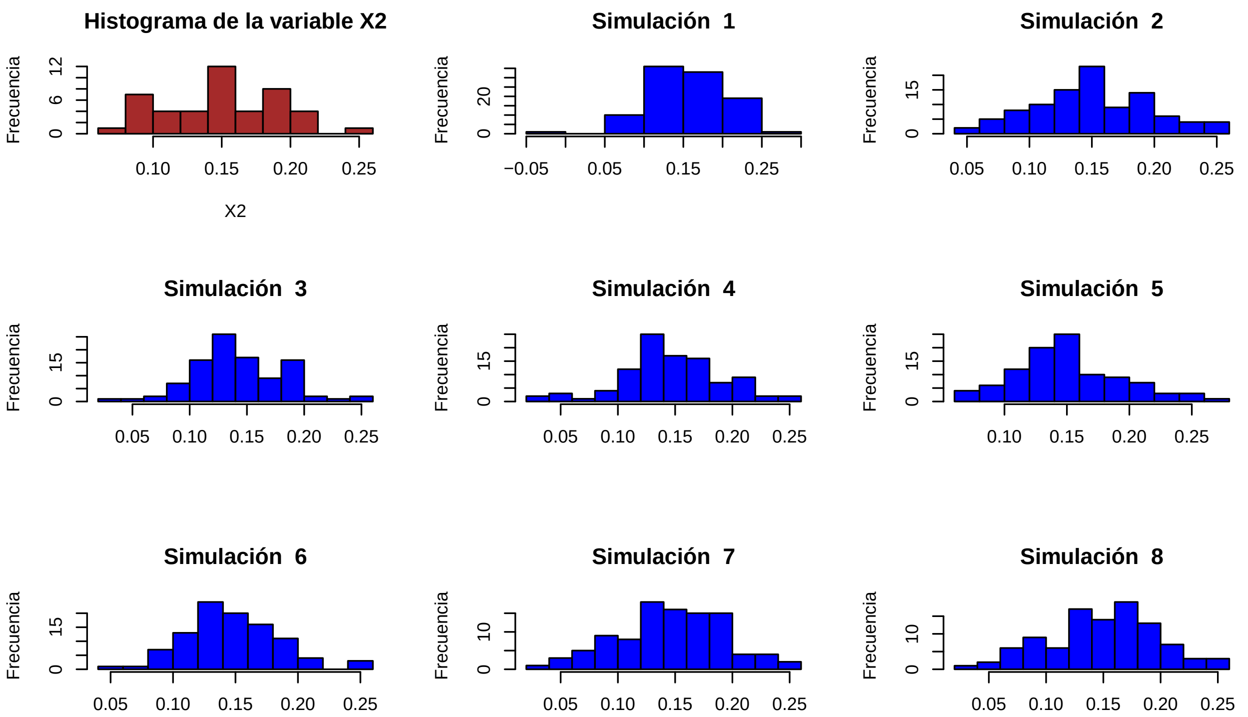

Figura 8: Comparación valores originales con 8 imulaciones de una distribución Normal.

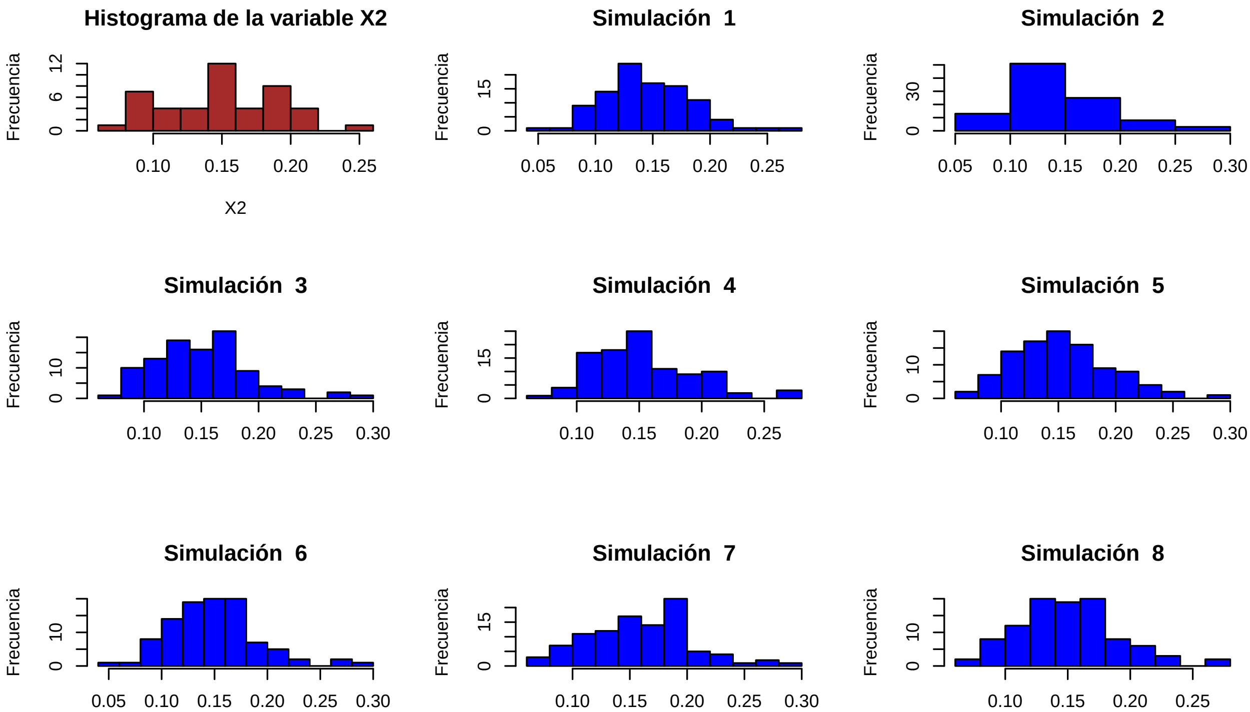

Figura 9: Comparación valores originales con 8 imulaciones de una distribución Gamma.

De estas simulaciones se pudo concluir que la distribución que mejor se adapta a los datos originales de la variable X2, es la distribución normal con media, $\mu = 0.151$ y desviación estándar, $\sigma = 0.042$.

Con estos resultados es posible calcular un intervalo de confiaza para la media de la siguiente manera:

```
# Intervalo de confianza para una muestra grande (mayor a 30) de X1
t.test( X1, conf.level = 0.97 )$conf.int
```

```
[1] 0.2657556 0.2951778
attr(,"conf.level")
[1] 0.97
```

Note que el valor obtenido por los estimadores coincide con el intervalo de confianza pues $0.280 \in (0.2658, 0.2952)$.

```
# Intervalo de confianza para una muestra grande (mayor a 30) de X2
t.test( X2, conf.level = 0.97 )$conf.int
```

```
[1] 0.1368441 0.1649337
attr(,"conf.level")
[1] 0.97
```

Note que el valor obtenido por los estimadores coinicide con el intervalo de confianza pues $0.151 \in (0.1368, 0.1649)$.