



Tutorial - Mi primera vez con R

Marisol Valencia Cárdenas

8/11/2019

```
knitr::opts_chunk$set(echo = FALSE)
```

Primera vez usando el R y Rstudio

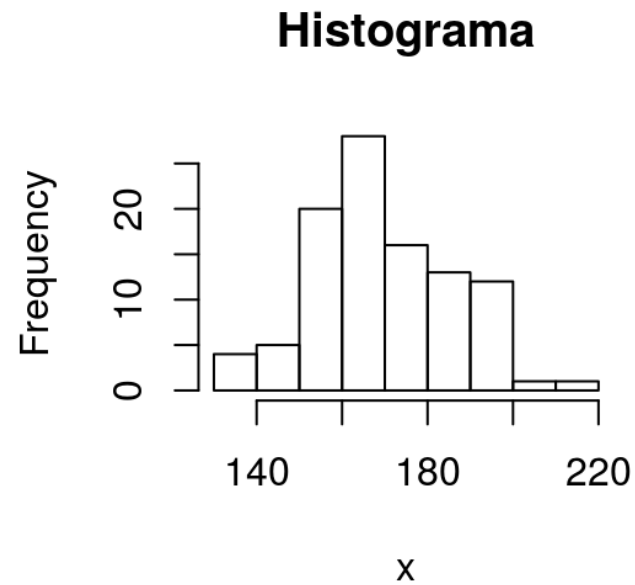
R es software libre y Rstudio tiene una versión gratuita.

La versión de R está disponible en www.cran.r-project.org, en versión de Windows o de Linux.

Simular variables y graficar.

Simular una variable aleatoria y crear un histograma con ésta variable, se puede hacer según se muestra en las siguientes líneas en gris.

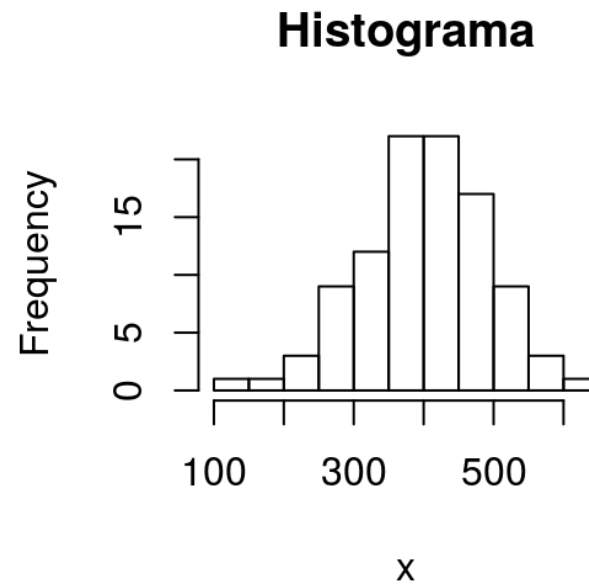
```
x <- rnorm(n=100, mean=170, sd=15)  
hist(x,main='Histograma')
```



Cambios en los parámetros

Simular una variable aleatoria y crear un histograma con ésta variable, se puede hacer según se muestra en las siguientes líneas en gris.

```
x <- rnorm(n=100, mean=400, sd=100)
hist(x, main='Histograma')
```



Simular y realizar histograma

Variable aleatoria con distribución normal.

```
x=rnorm(100,300,10)  
hist(x,breaks=10,col='pink',main='Histograma')
```

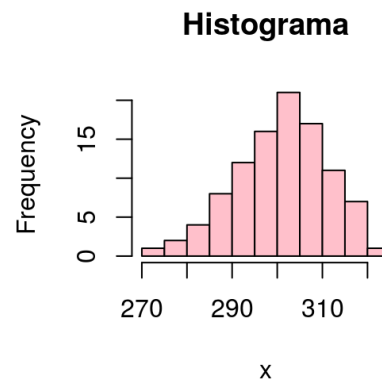
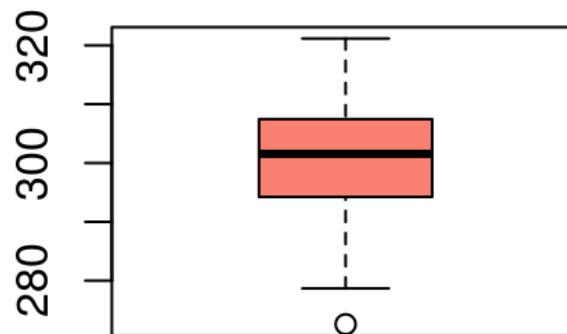


Diagrama de cajas y bigotes.

```
boxplot(x,col='salmon',main='Diagrama de Cajas y bigotes')
```

Diagrama de Cajas y bigotes



Simular y realizar histograma de distribución Poisson.

```
xp=rpois(1000,25)  
hist(xp,breaks=20,col='blue',main='Histograma')
```

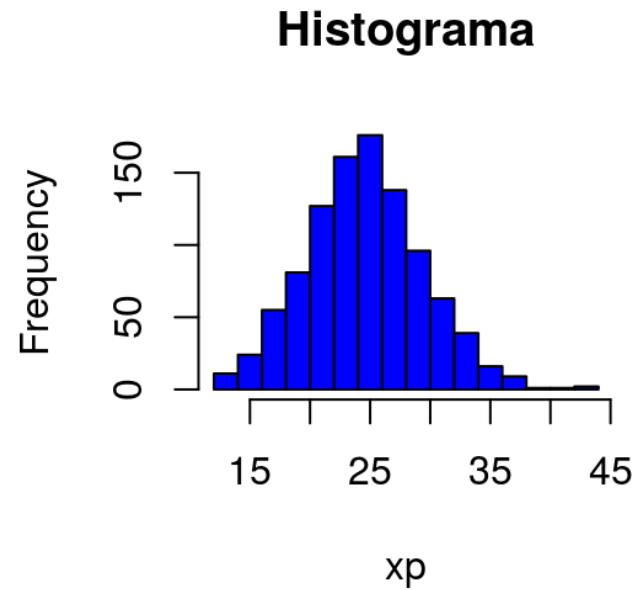
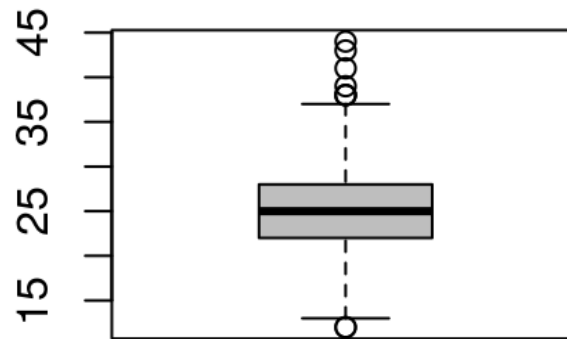


Diagrama de cajas y bigotes.

```
boxplot(xp,col='grey',main='Diagrama de Cajas y bigotes')
```

Diagrama de Cajas y bigotes



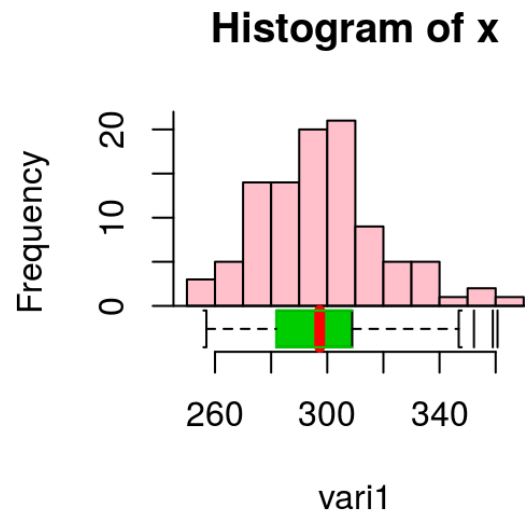
Instalar un paquete para otros estadísticos.

```
install.packages('sfsmisc')  
library(sfsmisc)
```

Histograma y diagrama de cajas y bigotes con el paquete sfsmisc

Variable con distribución T de Student.

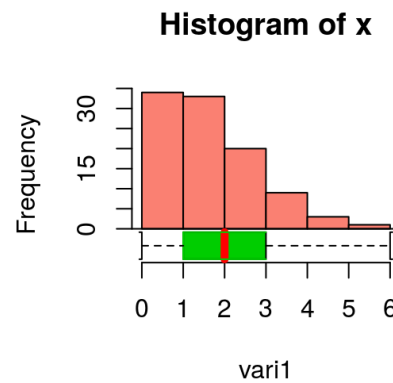
```
vari1=rt(100,99,300)  
histBxp(vari1,col='pink')
```



Simulación de variable con distribución Poisson

Variable con distribución Poisson.

```
vari1=rpois(100,2)  
histBxp(vari1,col='salmon')
```



Instalar un paquete para otras estadísticas.

El paquete fBasics es útil para estimar diferentes estadísticas básicas de una variable aleatoria determinada.

```
install.packages("fBasics")  
library(fBasics)
```

Resumen de estadísticas generales

```
x1=rt(100,99,1)  
summary(x1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## -1.9305  0.3714  1.0408  0.9274  1.4676  3.2842
```

Resumen de estadísticas generales con el paquete fBasics

La función `basicStats` estima diferentes estadísticas.

```
vari1=rt(100, 99,300)  
basicStats(vari1)
```

```
##              vari1  
## nobs          100.000000  
## NAs           0.000000  
## Minimum      266.858888  
## Maximum      347.710869  
## 1. Quartile   289.052547  
## 3. Quartile   314.483403  
## Mean          302.343849  
## Median        301.373781  
## Sum           30234.384855  
## SE Mean       1.886855  
## LCL Mean      298.599919  
## UCL Mean      306.087779  
## Variance      356.022234  
## Stdev         18.868551  
## Skewness       0.352429  
## Kurtosis      -0.408431
```

Resumen de estadísticas generales con el paquete fBasics

```
library(knitr)
vari1=rpois(200,3)
tablaes=basicStats(vari1)
kable(tablaes,digits=2)
```

| | vari1 |
|-------------|--------|
| nobs | 200.00 |
| NAs | 0.00 |
| Minimum | 0.00 |
| Maximum | 9.00 |
| 1. Quartile | 2.00 |
| 3. Quartile | 4.00 |
| Mean | 3.09 |

Hacer pruebas de diferencia de medias

```
x=rnorm(100,3,1)
y=rnorm(100,5,5)
t.test(x,y)

##
##  Welch Two Sample t-test
##
## data:  x and y
## t = -6.0161, df = 108.86, p-value = 2.441e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.739297 -1.886037
## sample estimates:
## mean of x mean of y
##  2.882455  5.695122
```


Pruebas de normalidad

shapiro.test es una función para aplicar la prueba de Shapiro Wilks.

```
x=rnorm(100,3,1)
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.99106, p-value = 0.7501
```

Pruebas de normalidad del paquete fBasics

```
x=rnorm(100,3,1)
normalTest(x, 'sw')
```

```
##
## Title:
##  Shapiro - Wilk Normality Test
##
## Test Results:
##  STATISTIC:
##    W: 0.9871
##  P VALUE:
##    0.4455
##
## Description:
##  Fri Nov  1 20:48:08 2019 by user:
```

Otras pruebas de normalidad

La función `normalTest(x)` de `fBasics` realiza cuatro pruebas de normalidad: Shapiro Wilks, Jarque Bera, Kolmogorov, D'Agostino.

Otras pruebas de normalidad

```
x=rpois(100,3)
normalTest(x, 'jb')

##
## Title:
##  Jarque - Bera Normalality Test
##
## Test Results:
##  STATISTIC:
##    X-squared: 24.5041
##    P VALUE:
##    Asymptotic p Value: 4.775e-06
##
## Description:
##  Fri Nov  1 20:48:08 2019 by user:
```

Prueba de bondad de ajuste Kolmogorov para Distribución Poisson

```
x=rpois(50,10)
ks.test(x, 'ppois', lambda=10)

##
## One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.20304, p-value = 0.03241
## alternative hypothesis: two-sided
```

Prueba de bondad de ajuste Chicuadrado

```
x1=rpois(50,10)
x2=rpois(50,10)
chisq.test(x1,x2)

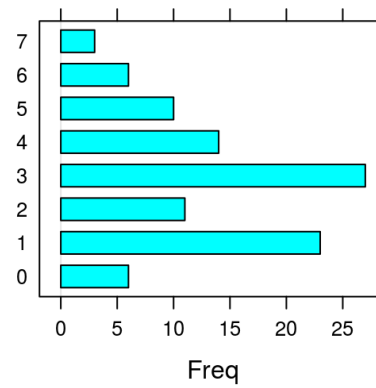
##
##  Pearson's Chi-squared test
##
## data:  x1 and x2
## X-squared = 151.9, df = 180, p-value = 0.937
```

Diagramas de barra

```
require(lattice)
```

```
## Loading required package: lattice
```

```
fax=factor(rpois(100,3))  
barchart(table(fax))
```



Crear vectores de datos

```
x1=c(1.2,4.1,4,15,12,16,16)
```

```
x2=c(0,1.1,2,13,12,15,12)
```

```
x1;x2
```

```
## [1]  1.2  4.1  4.0 15.0 12.0 16.0 16.0
```

```
## [1]  0.0  1.1  2.0 13.0 12.0 15.0 12.0
```


Creación de matrices

```
matri1=cbind(x1,x2)  
matri1
```

```
##           x1    x2  
## [1,]  1.2  0.0  
## [2,]  4.1  1.1  
## [3,]  4.0  2.0  
## [4,] 15.0 13.0  
## [5,] 12.0 12.0  
## [6,] 16.0 15.0  
## [7,] 16.0 12.0
```

Estimar Covarianza y Correlación

```
cov(x1,x2)
```

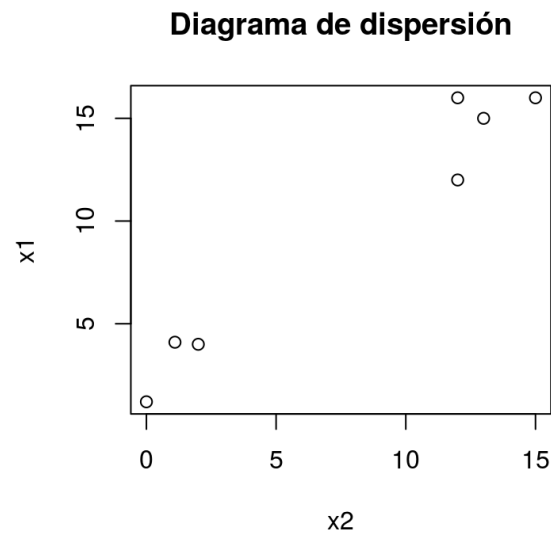
```
## [1] 40.9819
```

```
cor(x1,x2)
```

```
## [1] 0.9790429
```

Diagrama de dispersión

```
plot(x1~x2,main='Diagrama de dispersión')
```



Simular regresión

```
xt=rnorm(100,2,1)  
y=2-1.9*xt+rnorm(100,0,1)
```

Covarianza, Correlación y diagrama de dispersión de 'y' vs 'xt'

```
cov(xt,y)
```

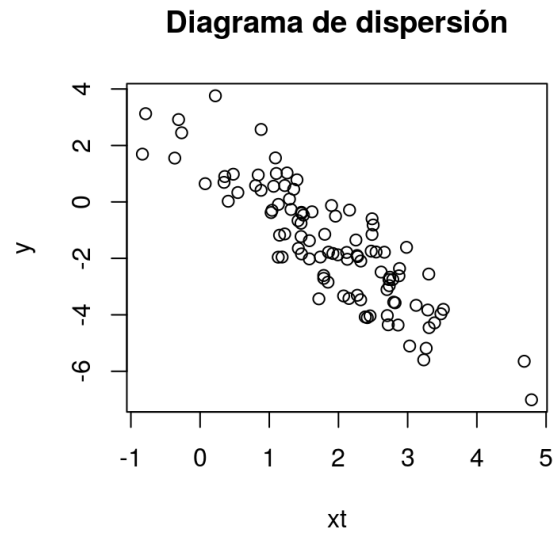
```
## [1] -1.956941
```

```
cor(xt,y)
```

```
## [1] -0.8721121
```

Diagrama de dispersión de 'y' vs 'xt'

```
plot(y~xt, main='Diagrama de dispersión')
```



Cargar una base de datos

Forma 1: Primero se guarda la base de datos en formato csv. Instrucciones de lectura:

```
dataset1 <- read.csv(file.choose(), sep=';', dec=',', head=T)  
View(dataset1)  
attach(dataset1)
```

Cargar una base de datos

Forma 2:

Si se tiene el archivo guardado con extensión en xlsx.

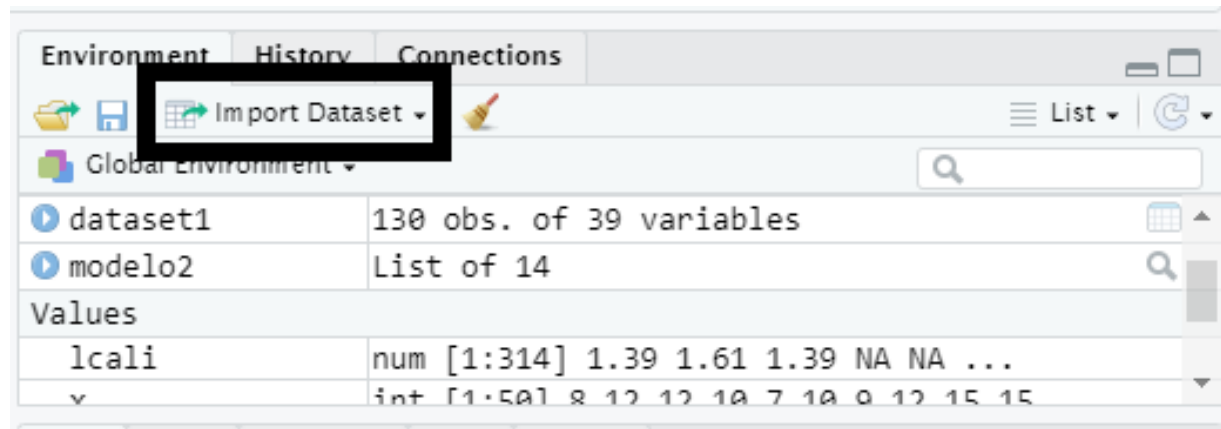
```
library(readxl)
dataset1 <- read_excel(file.choose())
View(dataset1)
attach(dataset1)
```

Cargar la base de datos asignada para el taller.

Cargar una base de datos

Forma 3:

Cargar la base desde la sección Enviroment de Rstudio, en el item Import Dataset. Se elige la opción From Excel y se busca la base de datos que ha guardado previamente.



Análisis de covarianza y correlación

```
plot(totalservicios,totalapps)  
cov(totalservicios,totalapps)  
cor(totalservicios,totalapps)
```

Realizar ejercicio en el taller.

Modelo de regresión lineal múltiple

```
y=log(totalservicios)
modelo2=lm(y~género+ingresos)
summary(modelo2)
Anova(modelo2,type='III')
```

Realizar ejercicio en taller.

Iniciando con Rmarkdown

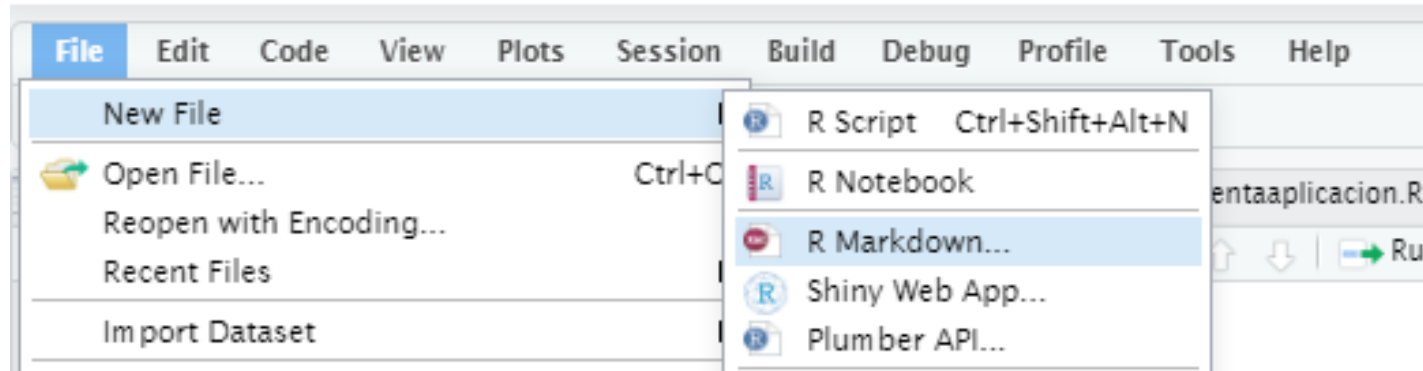
Para trabajar con R Markdown usted necesita: * Instalar R: [R](#) * Instalar la última versión de Rstudio: [RStudio](#)

Para trabajar con R Markdown usted necesita:

- Instalar R: [R](#)
- Instalar la última versión de Rstudio: [RStudio](#)
- Instalar el paquete knitr : `install.packages("knitr")`

Crear un documento en Rmarkdown

- Entrar por file (archivo), new file (nuevo archivo), R markdown.
- Se elige el formato en el cual se puede compilar: pdf, doc, html.



Gracias

- Marisol Valencia Cárdenas
- Docente
- Fundación Universitaria Autónoma de las Américas
- marisol.valencia@uam.edu.co

