Week 4

Tomás

13 de noviembre de 2019

### str()

La funcion str() es una alternativa a la tipica funcion summary nos entrega caracterisiticas del objeto que le entregamos, str se puede leer como estructura del objeto.

### Numeros Aleatorios

Se pueden recrear numeros aleatorios especialmente para simulacion y estadistica:

* rnorm : Numeros normales random
* rpois : Numeros poisson random

Estos numeros aleatorios pueden ser antecedidos por:

* d : density
* r : random number generation
* p : cumulative distribution
* q : quantile function

### Normal

dnorm(x, mean = 0, sd = 1, log = FALSE) # Densidad de probabilidad  
qnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE) # Distribucion acumulativa  
pnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE) # Inversa de qnorm  
rnorm(n, mean = 0, sd = 1) # Numeros random

Cada vez que se generan numeros aleatorios es importante plantar una semilla para poder generar los mismo numeros siempre

set.seed(1)  
rnorm(10,0,1)

## [1] -0.6264538 0.1836433 -0.8356286 1.5952808 0.3295078 -0.8204684  
## [7] 0.4874291 0.7383247 0.5757814 -0.3053884

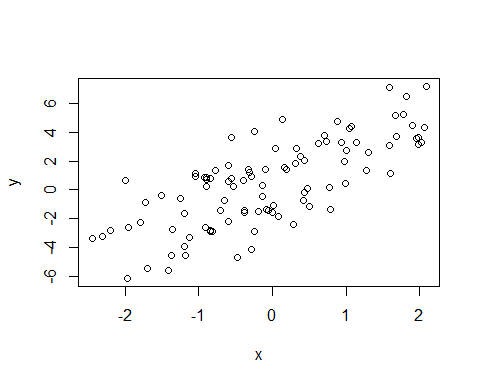
### Generando numeros aleatorios para una regresion lineal

Supongamos que queremos general un modelo lineal (Considerando x variable aleatoria normal):

set.seed(2)  
x <- rnorm(100)  
e <- rnorm(100,0,2)  
y <- 0.5 + 2\*x + e  
summary(y)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -6.1180 -1.5795 0.6741 0.4970 2.9112 7.1640

plot(x,y)

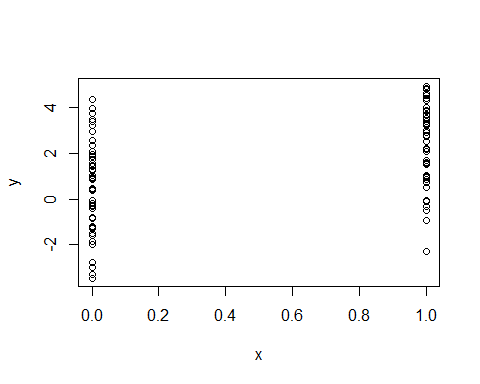


Supongamos que queremos general un modelo lineal (Considerando x variable aleatoria binomial):

set.seed(3)  
x <- rbinom(100,1, 0.5)  
e <- rnorm(100,0,2)  
y <- 0.5 + 2\*x + e  
summary(y)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.4647 -0.2239 1.4970 1.4364 3.2434 4.9341

plot(x,y)

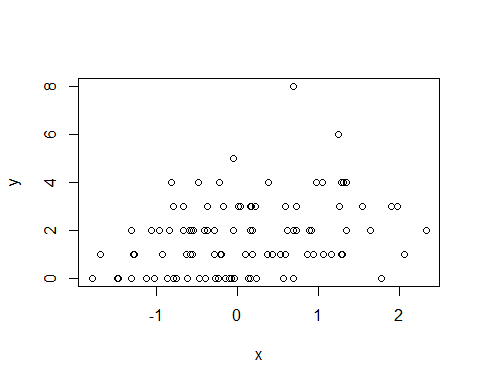


Supongamos que queremos general un modelo lineal (Considerando x una variable poisson):

set.seed(4)  
x <- rnorm(100)  
log.mu <- 0.5 + 0.3 \* x  
y <- rpois(100,exp(log.mu))  
summary(y)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 1.00 2.00 1.74 3.00 8.00

plot(x,y)



### Random Sampling

Esta funcion te permite escoger un numero o elemento al azar entre un conjunto de datos que se entrega

set.seed(5)  
sample(1:10,4)

## [1] 2 9 7 3

sample(1:10) #Obtengo una permutacion de los numeros

## [1] 9 10 5 6 3 7 2 4 1 8

sample(letters,5)

## [1] "l" "p" "e" "i" "v"

sample(1:10,replace = TRUE)

## [1] 10 5 10 10 3 1 10 6 8 7