

SIMULACIÓN DEL EFECTO DE LA LEY DE LOS GRANDES NÚMEROS CON PURRR

Fabiola & Jimena

23 de abril de 2019

1. Cargar las librerías necesarias.

```
library(purrr)
library(dplyr)
library(reshape2)
library(ggplot2)
```

2. Crear un data frame conformado por 5000 muestras, cada columna es una muestra diferente.

```
df <- data.frame( n_100 = 1:5000 %>% map_dbl(~ rnorm(100) %>% mean),
                  n_300 = 1:5000 %>% map_dbl(~ rnorm(300) %>% mean),
                  n_500 = 1:5000 %>% map_dbl(~ rnorm(500) %>% mean),
                  n_700 = 1:5000 %>% map_dbl(~ rnorm(700) %>% mean),
                  n_1000 = 1:5000 %>% map_dbl(~ rnorm(1000) %>% mean) )

head(df)

##           n_100           n_300           n_500           n_700           n_1000
## 1 0.03019068 0.10637601 -0.054956136 0.049993176 -0.03168138
## 2 0.32341325 -0.01047090 0.002073066 0.053301766 -0.04764781
## 3 0.05583186 -0.01870543 0.015272225 0.021549158 0.00022493
## 4 0.04690703 0.05549385 0.002346898 -0.053000477 -0.04904445
## 5 0.16706523 0.04287960 0.077361462 0.002343872 -0.06194401
## 6 0.10534456 -0.08825272 -0.007285741 -0.007805330 -0.06490391
```

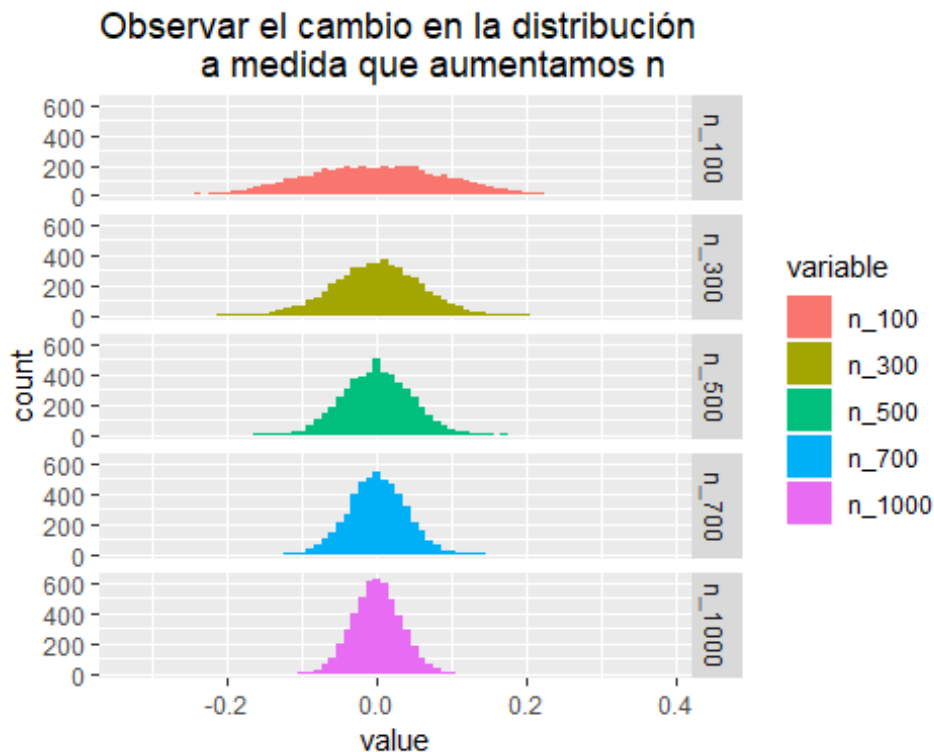
3. Verificación.

```
df %>% map(var)

## $n_100
## [1] 0.01002137
##
## $n_300
## [1] 0.003259781
##
## $n_500
## [1] 0.002003208
##
## $n_700
```

```
## [1] 0.001409067
##
## $n_1000
## [1] 0.00100473

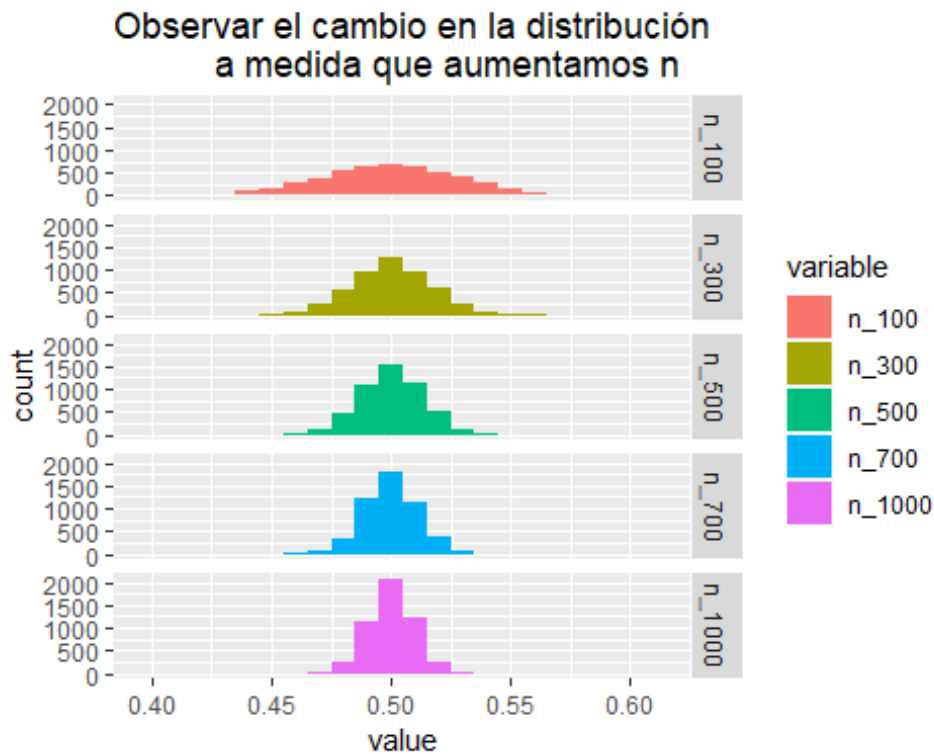
ggplot() +
  geom_histogram(data=df %>% melt, aes(value, fill = variable), binwidth
= 0.01) +
  facet_grid(variable ~ .) +
  ggtitle("Observar el cambio en la distribución
a medida que aumentamos n")
```



4. Observar que sin importar la distribución que se utilice, al aumentar el número de repeticiones la media de la distribución converge a una normal.

```
df <- data.frame( n_100 = 1:5000 %>% map_dbl(~ runif(100) %>% mean),
                  n_300 = 1:5000 %>% map_dbl(~ runif(300) %>% mean),
                  n_500 = 1:5000 %>% map_dbl(~ runif(500) %>% mean),
                  n_700 = 1:5000 %>% map_dbl(~ runif(700) %>% mean),
                  n_1000 = 1:5000 %>% map_dbl(~ runif(1000) %>% mean) )

ggplot() +
  geom_histogram(data=df %>% melt, aes(value, fill = variable), binwidth
= 0.01) +
  facet_grid(variable ~ .) +
  ggtitle("Observar el cambio en la distribución
a medida que aumentamos n")
```



Simular esto con purrr elimina la necesidad de muchos for loops, manteniendo las cosas más limpias.

5. Cómo se haría sin purrr.

```
sampling<-function(n_means,sample_means){

  x<-as.data.frame(c())

  for(i in 1:length(sample_means)){

    for(j in 1:length(n_means)){

      x[j,i] <- mean(rexp(sample_means[i])) #Cambiar dst:
runif , rexp , rpois(sample_means[i],2)
    }
  }

  return(x)
}

sample_means<-c(100,300,500,700,1000)
n_means<-1:1000

df<-sampling(n_means,sample_means)
```

```
ggplot() +
  geom_histogram(data=df %>% melt, aes(value, fill = variable), binwidth
= 0.01) +
  facet_grid(variable ~ .) +
  ggtitle("Observar el cambio en la distribución
a medida que aumentamos n")

## No id variables; using all as measure variables
```

