

Ejercicio de los carros

Alonso Pizarro Lagunas

23/12/2021

Análisis de los coches con database 'mtcars' de ggplot

```
library(ggplot2)
mtcars <- mtcars
```

Carga de datos en Python.

```
import pandas as pd      # Pandas
import numpy as np       # Numpy
import matplotlib.pyplot as plt # Matplotlib

mtcars = r.mtcars        # El DF que se cargo desde R ahora lo cargamos en python
print(mtcars.shape)      # Vemos la dimensión de nuestra data
```

```
## (32, 11)
```

```
print(mtcars.head())     # Visualizamos las primeras líneas de data
```

```
##           mpg  cyl  disp    hp  drat    ...  qsec  vs  am  gear  carb
## Mazda RX4      21.0  6.0  160.0  110.0  3.90    ...  16.46  0.0  1.0   4.0   4.0
## Mazda RX4 Wag  21.0  6.0  160.0  110.0  3.90    ...  17.02  0.0  1.0   4.0   4.0
## Datsun 710     22.8  4.0  108.0   93.0  3.85    ...  18.61  1.0  1.0   4.0   1.0
## Hornet 4 Drive  21.4  6.0  258.0  110.0  3.08    ...  19.44  1.0  0.0   3.0   1.0
## Hornet Sportabout 18.7  8.0  360.0  175.0  3.15    ...  17.02  0.0  0.0   3.0   2.0
##
## [5 rows x 11 columns]
```

Medidas de Centralización

```
print(mtcars.mean())     # Media de cada una de las columnas
```

```
## mpg      20.090625
## cyl       6.187500
## disp     230.721875
## hp       146.687500
## drat      3.596563
## wt       3.217250
## qsec     17.848750
## vs       0.437500
## am       0.406250
## gear     3.687500
## carb     2.812500
```

```
## dtype: float64
print(mtcars.mean(axis = 1)) # Media por filas (Nota que esto no hace mucho sentido)
```

```
## Mazda RX4                29.907273
## Mazda RX4 Wag            29.981364
## Datsun 710                23.598182
## Hornet 4 Drive           38.739545
## Hornet Sportabout        53.664545
## Valiant                   35.049091
## Duster 360               59.720000
## Merc 240D                24.634545
## Merc 230                 27.233636
## Merc 280                 31.860000
## Merc 280C                31.787273
## Merc 450SE               46.430909
## Merc 450SL               46.500000
## Merc 450SLC              46.350000
## Cadillac Fleetwood       66.232727
## Lincoln Continental       66.058545
## Chrysler Imperial        65.972273
## Fiat 128                  19.440909
## Honda Civic               17.742273
## Toyota Corolla            18.814091
## Toyota Corona             24.888636
## Dodge Challenger          47.240909
## AMC Javelin               46.007727
## Camaro Z28                58.752727
## Pontiac Firebird          57.379545
## Fiat X1-9                 18.928636
## Porsche 914-2             24.779091
## Lotus Europa              24.880273
## Ford Pantera L            60.971818
## Ferrari Dino              34.508182
## Maserati Bora              63.155455
## Volvo 142E                26.262727
## dtype: float64
```

```
print(mtcars.median()) # Mediana para cada una de las columnas
```

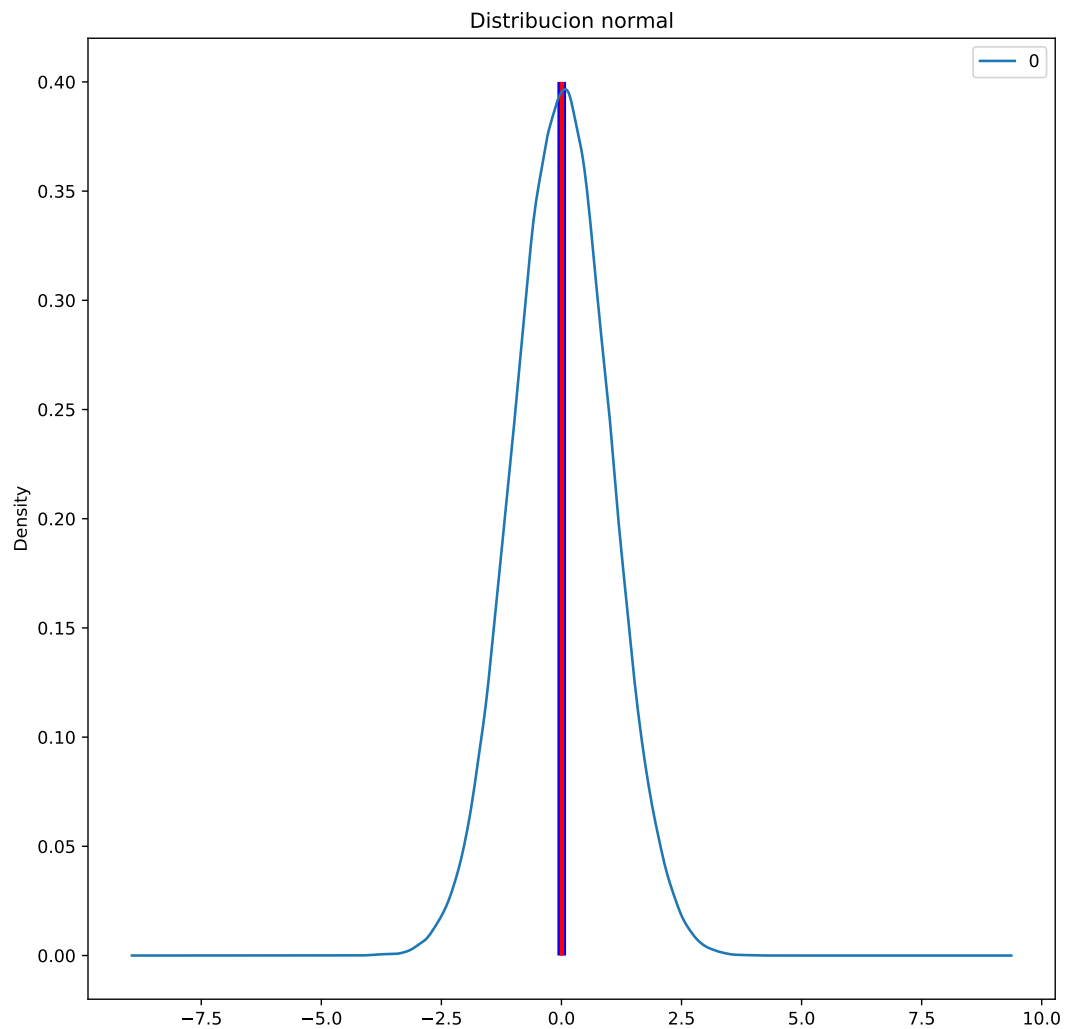
```
## mpg      19.200
## cyl       6.000
## disp     196.300
## hp       123.000
## drat       3.695
## wt        3.325
## qsec      17.710
## vs         0.000
## am         0.000
## gear       4.000
## carb       2.000
## dtype: float64
```

```
print(mtcars['mpg'].mean()) # media para la columna de 'mpg'
```

```
## 20.090624999999996
```

Medidas contra distribuciones

```
normData = pd.DataFrame(np.random.normal(size=100000)) # Generamos datos normalmente dist.  
  
normData.plot(kind = 'density', figsize=(10,10)) # Grafica  
plt.vlines(normData.mean(), ymin = 0, ymax = 0.4,  
linewidth=5.0, color = "blue")  
plt.vlines(normData.median(), ymin = 0, ymax = 0.4,  
linewidth=2.5, color = "red")  
plt.title('Distribucion normal')  
plt.show()
```



Muestras con sesgos

```
skewedData = pd.DataFrame(np.random.exponential(size=100000)) # Generamos datos normalmente dist.

skewedData.plot(kind = 'density', figsize=(10,10), xlim = (-1,5)) # Grafica
plt.vlines(skewedData.mean(), ymin = 0, ymax = 1.0,
linewidth=5.0, color = "blue")
plt.vlines(skewedData.median(), ymin = 0, ymax = 1.0,
linewidth=2.5, color = "red")
plt.title('Distribucion sesgada')
plt.show()
```

