Coches

Hernández Martínez Oscar Gerardo

2/7/2020

```
import os
os.environ['QT_QPA_PLATFORM_PLUGIN_PATH'] = 'C:/Users/jxsje/anaconda3/Library/plugins/platforms'
```

Análisis de los coches (mtcars)

```
pycars = r.data
```

Medidas de centralización

```
print(pycars.mean()) #Media por 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(pycars.mean()) #Media por filas
```

```
## 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(pycars.median()) #Mediana
```

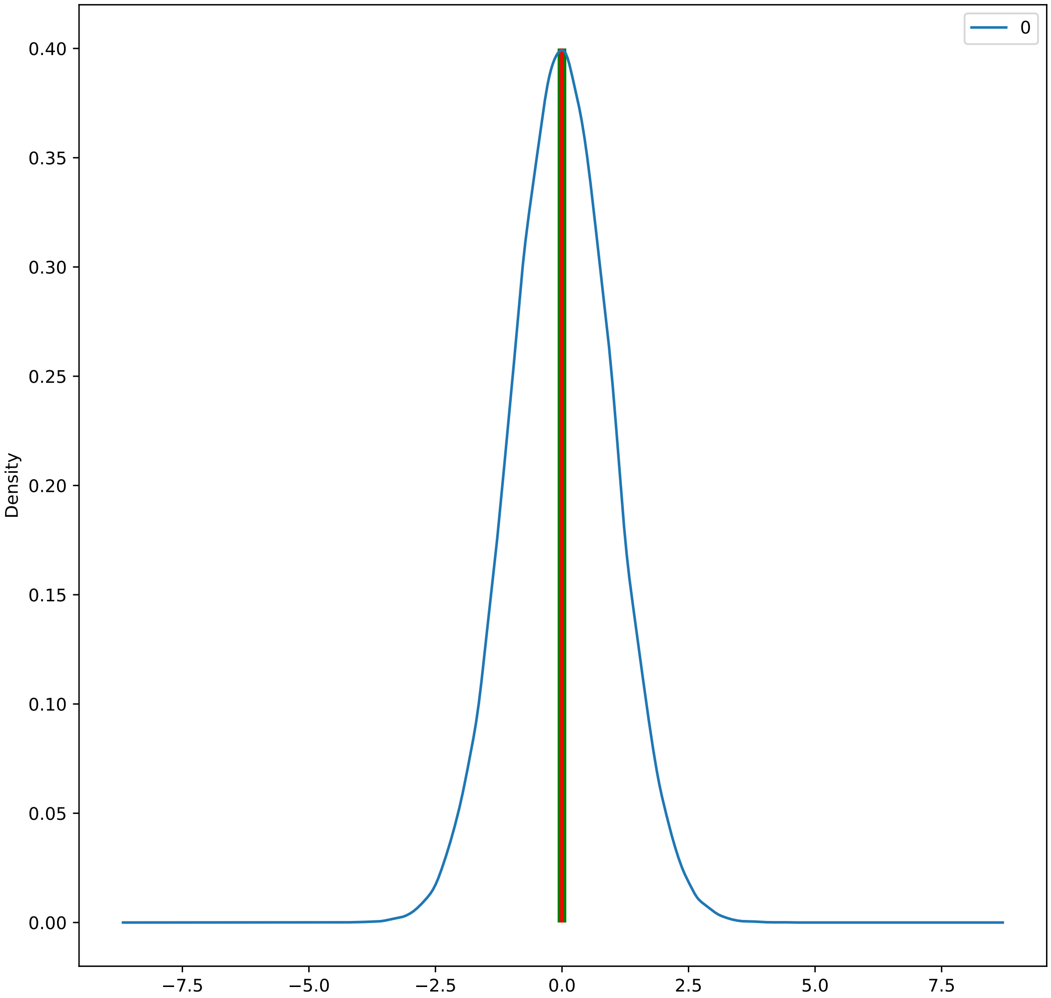
```
## mpg      19.200
## cyl       6.000
## disp    196.300
## hp      123.000
## drat      3.695
## wt        3.325
## qsec     17.710
## vs        0.000
## am        0.000
## gear      4.000
## carb      2.000
## dtype: float64
```

```
print(pycars.mode()) #Moda
```

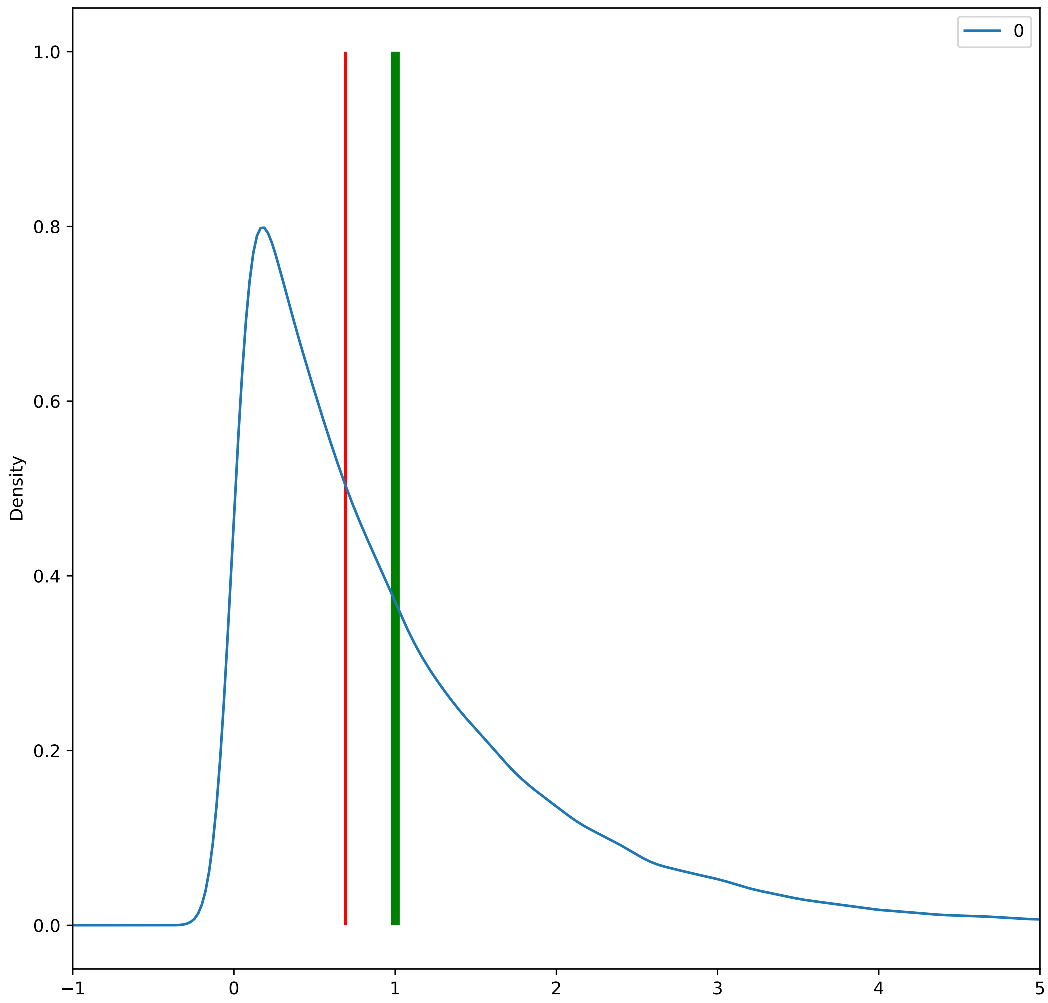
```
##     mpg  cyl   disp     hp  drat    wt   qsec   vs   am  gear  carb
## 0  10.4  8.0  275.8  110.0  3.07  3.44  17.02  0.0  0.0   3.0   2.0
## 1  15.2  NaN    NaN  175.0  3.92   NaN  18.90  NaN  NaN   NaN   4.0
## 2  19.2  NaN    NaN  180.0   NaN   NaN    NaN  NaN  NaN   NaN   NaN
## 3  21.0  NaN    NaN    NaN   NaN   NaN    NaN  NaN  NaN   NaN   NaN
## 4  21.4  NaN    NaN    NaN   NaN   NaN    NaN  NaN  NaN   NaN   NaN
## 5  22.8  NaN    NaN    NaN   NaN   NaN    NaN  NaN  NaN   NaN   NaN
## 6  30.4  NaN    NaN    NaN   NaN   NaN    NaN  NaN  NaN   NaN   NaN
```

Medidas vs Distribuciones

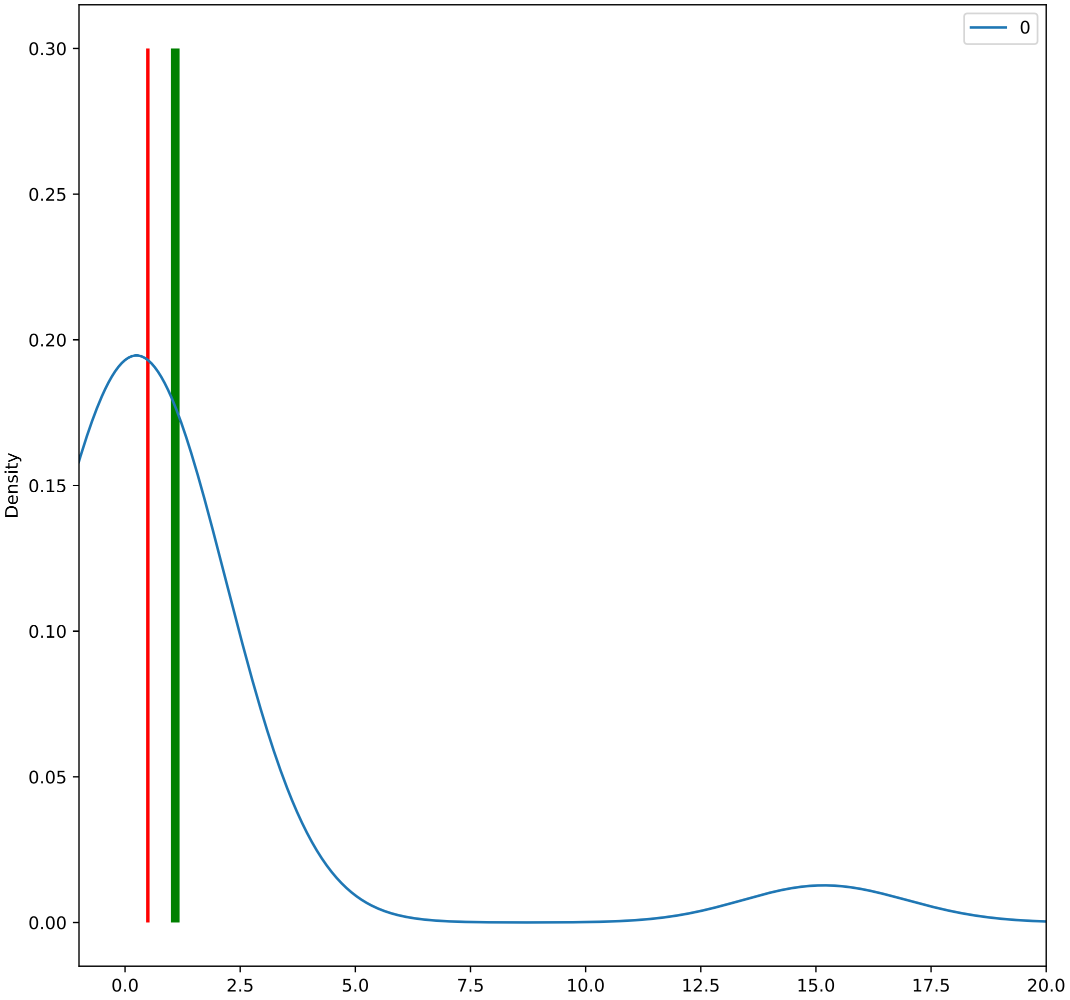
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

norm_data = pd.DataFrame(np.random.normal(size=100000))
norm_data.plot(kind="density", figsize=(10,10))
plt.vlines(norm_data.mean(), ymin=0, ymax=0.4, linewidth=5.0, color="green")
plt.vlines(norm_data.median(), ymin=0, ymax=0.4, linewidth=2.0, color="red")
plt.show()
```



```
plt.clf()
skewed_data = pd.DataFrame(np.random.exponential(size=100000))
skewed_data.plot(kind="density", figsize=(10,10), xlim=(-1,5))
plt.vlines(skewed_data.mean(), ymin=0, ymax=1.0, linewidth=5.0, color = "green")
plt.vlines(skewed_data.median(), ymin=0, ymax=1.0, linewidth=2.0, color="red")
plt.show()
```



```
plt.clf()
norm_data = np.random.normal(size=50)
outliers = np.random.normal(15, size = 3)
combined_data = pd.DataFrame(np.concatenate((norm_data, outliers), axis=0))
combined_data.plot(kind="density", figsize=(10,10), xlim=(-1,20))
plt.vlines(combined_data.mean(), ymin=0, ymax=0.3, linewidth=5.0, color = "green")
plt.vlines(combined_data.median(), ymin=0, ymax=0.3, linewidth=2.0, color="red")
plt.show()
```



Medidas de dispersión

- Rango de `mpg`

```
max(pycars["mpg"]) - min(pycars["mpg"])
```

```
## 23.5
```

```
five_nums = [pycars["mpg"].quantile(0),
pycars["mpg"].quantile(0.25),
pycars["mpg"].quantile(0.5),
pycars["mpg"].quantile(0.75),
pycars["mpg"].quantile(1.0)]

print(five_nums)
```

```
## [10.4, 15.425, 19.2, 22.8, 33.9]
```

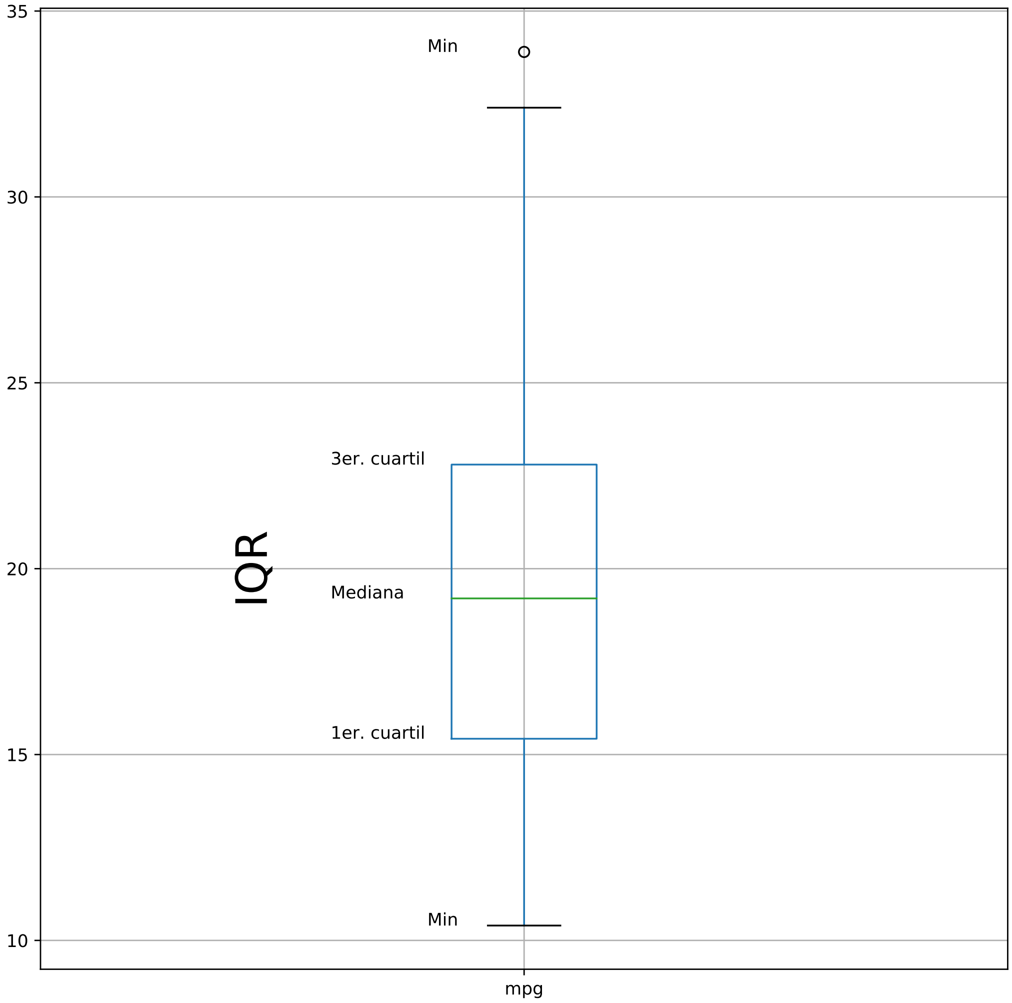
```
def five_numbers(var):
    quantiles = []
    for i in range(0, 5):
        quantiles.append(var.quantile(i/4))
    return quantiles

five_numbers(pycars["mpg"])
```

```
## [10.4, 15.425, 19.2, 22.8, 33.9]
```

```
print(pycars["mpg"].quantile(0.75) - pycars["mpg"].quantile(0.25))
```

```
## 7.375
```

```
plt.clf()
pycars.boxplot(column = "mpg", return_type = "axes", figsize = (10,10))
plt.text(x=0.8, y=pycars["mpg"].quantile(0.25), s ="1er. cuartil")
plt.text(x=0.8, y=pycars["mpg"].quantile(0.5), s ="Mediana")
plt.text(x=0.8, y=pycars["mpg"].quantile(0.75), s ="3er. cuartil")
plt.text(x=0.9, y=pycars["mpg"].quantile(0), s="Min")
plt.text(x=0.9, y=pycars["mpg"].quantile(1), s="Min")
plt.text(x=0.7, y=pycars["mpg"].quantile(0.5), s="IQR", rotation=90, size=25)
plt.show()
```



Varianza y desviación estándar

```
print(pycars["mpg"].var())
```

```
## 36.32410282258065
```

```
print(pycars["mpg"].std())
```

```
## 6.026948052089105
```

```
mad = abs(pycars["mpg"]-pycars["mpg"].median())
k = 1.4826
print("La desviación mediana absoluta es %f"%(mad.median()*k))
```

```
## La desviación mediana absoluta es 5.411490
```

El Sesgo y la Curtosis

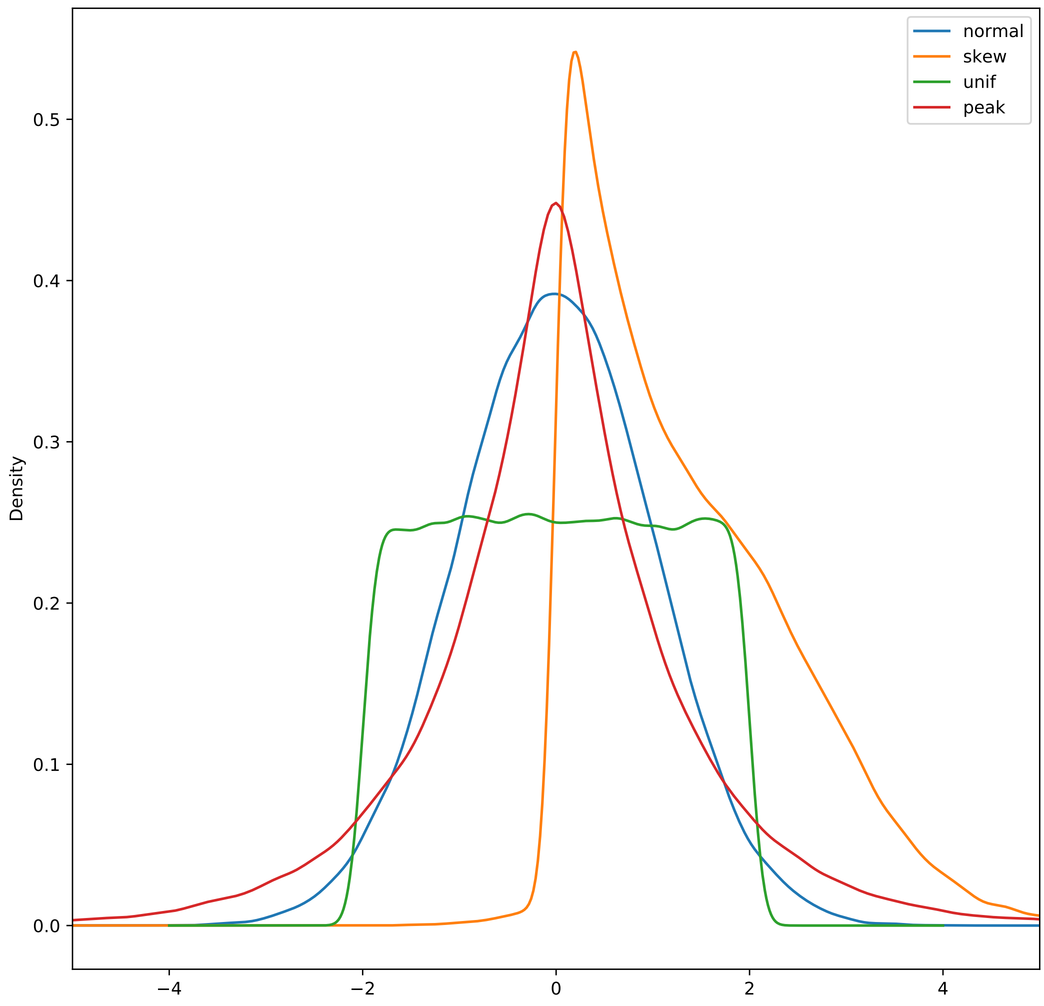
```
print(pycars["mpg"].skew())
```

```
## 0.6723771376290805
```

```
print(pycars["mpg"].kurt())
```

```
## -0.0220062914240855
```

```
norm = np.random.normal(size=100000)
skew = np.concatenate((np.random.normal(size=35000)+2,
                      np.random.exponential(size=65000)),
                      axis=0)
unif = np.random.uniform(-2,2, size=100000)
peak = np.concatenate((np.random.exponential(size=50000),
                      np.random.exponential(size=50000)*(-1)),
                      axis=0)
data = pd.DataFrame({
  "normal":norm,
  "skew":skew,
  "unif":unif,
  "peak":peak
})

plt.clf()
data.plot(kind="density", figsize = (10,10), xlim=(-5,5))
plt.show()
```



```
print("Normal, Sesgo:%f, Curtosis%f"%(data["normal"].skew(),data["normal"].kurt()))
```

```
## Normal, Sesgo:-0.001634, Curtosis-0.008884
```

```
print("Normal+Exp, Sesgo:%f, Curtosis%f"%(data["skew"].skew(),data["skew"].kurt()))
```

```
## Normal+Exp, Sesgo:0.989169, Curtosis1.166281
```

```
print("Uniforme, Sesgo:%f, Curtosis%f"%(data["unif"].skew(),data["unif"].kurt()))
```

```
## Uniforme, Sesgo:0.000414, Curtosis-1.193650
```

```
print("Suma de Exp, Sesgo:%f, Curtosis%f"%(data["peak"].skew(),data["peak"].kurt()))
```

```
## Suma de Exp, Sesgo:0.057468, Curtosis2.967783
```