## EJERCICIOS DE PANDAS

Cargue el fichero bmw.csv con el siguiente código:

```
import pandas as pd

# Conectar a Google Drive
from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('/content/drive/MyDrive/my_data/bmw.csv')
```

**Ejercicio 1:**

Muestre los primeros 10 registros de la base de datos.

```
import pandas as pd
df = pd.read_csv("bmw.csv")

print(df.iloc[0:10, :])
```

Resultado:

| | model | year | price | transmission | mileage | fuelType | tax | mpg | engineSize |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 5 Series | 2014 | 11200 | Automatic | 67068 | Diesel | 125 | 57.6 | 2.0 |
| 1 | 6 Series | 2018 | 27000 | Automatic | 14827 | Petrol | 145 | 42.8 | 2.0 |
| 2 | 5 Series | 2016 | 16000 | Automatic | 62794 | Diesel | 160 | 51.4 | 3.0 |
| 3 | 1 Series | 2017 | 12750 | Automatic | 26676 | Diesel | 145 | 72.4 | 1.5 |
| 4 | 7 Series | 2014 | 14500 | Automatic | 39554 | Diesel | 160 | 50.4 | 3.0 |
| 5 | 5 Series | 2016 | 14900 | Automatic | 35309 | Diesel | 125 | 60.1 | 2.0 |
| 6 | 5 Series | 2017 | 16000 | Automatic | 38538 | Diesel | 125 | 60.1 | 2.0 |
| 7 | 2 Series | 2018 | 16250 | Manual | 10401 | Petrol | 145 | 52.3 | 1.5 |
| 8 | 4 Series | 2017 | 14250 | Manual | 42668 | Diesel | 30 | 62.8 | 2.0 |
| 9 | 5 Series | 2016 | 14250 | Automatic | 36099 | Diesel | 20 | 68.9 | 2.0 |

**Ejercicio 2:**

Obtenga la serie correspondiente al atributo year, y a continuación obtenga el tipo de datos y el número de registros de dicha serie.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


print(pd.Series(df.loc[:, "year"]))
```

Resultado:

```
0       2014
1       2018
2       2016
3       2017
4       2014
       ...
10776   2016
10777   2016
10778   2017
10779   2014
10780   2017
Name: year, Length: 10781, dtype: int64
```

**Ejercicio 3:**

Obtenga la serie correspondiente al atributo mileage, y después seleccione los registros de posición múltiplo de 7 en dicha serie.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


serie = pd.Series(df.loc[:, "mileage"])

print(serie[serie.index%7==0])
```

Resultado:

```
0       67068

7       10401

14      19057

21      78957

28      96213

       ...

10752   41500

10759   54008

10766   54987

10773   60372

10780   59432

Name: mileage, Length: 1541, dtype: int64
```

**Ejercicio 4:**

Obtenga la serie correspondiente al atributo mileage, y después seleccione aleatoriamente el 40% de los registros de dicha serie.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


print(pd.Series(df["mileage"]).sample(frac=0.4))
```

Resultado:

```
8945    19750
5737      999
794     56423
6465     9888
9911   106000
       ...
6515     1000
2153    16202
7893     4000
7626    30000
463      2664
Name: mileage, Length: 4312, dtype: int64
```

**Ejercicio 5:**

Obtenga la serie correspondiente al atributo mileage, y después seleccione los registros de dicha serie con valor menor que 20000.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


serie = pd.Series(df["mileage"])

print(serie[serie<20000])
```

Resultado:

```
1       14827

7       10401

14      19057

15      16570

39       6522

       ...

10740    3551

10741    2784

10742    5634

10743   13165

10755   13955

Name: mileage, Length: 5610, dtype: int64
```

**Ejercicio 6:**

Obtenga la serie correspondiente al atributo mpg, y después ordene los registros de dicha serie.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


print(pd.Series(df["mpg"]).sort_values())
```

Resultado:

```
6965     5.5
6172     5.5
6132     5.5
6198     5.5
2116     5.5
       ...
7299    470.8
3628    470.8
6070    470.8
2352    470.8
7347    470.8
Name: mpg, Length: 10781, dtype: float64
```

**Ejercicio 7:**

Calcule la media, la desviación típica, el mínimo y el máximo del atributo engineSize.

```
import pandas as pd
df = pd.read_csv("bmw.csv")

serie=pd.Series(df["engineSize"])
print("Media: "+str(serie.mean()))
print("Desviacion estandar: "+str(serie.std()))
print("Minimo: "+str(serie.min()))
print("Maximo: "+str(serie.max()))
```

Resultado:

```
Media: 2.1677673685186902
Desviacion estandar: 0.5520537772398375
Minimo: 0.0
Maximo: 6.6
```

**Ejercicio 8:**

Obtenga el número de filas y columnas de la base de datos, así como el antepenúltimo registro.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


print("Filas: "+str(df.shape[0])+" Columnas: "+str(df.shape[1]))

print("Antepenultima fila: ")

print(df.iloc[df.shape[0]-3,:])
```

Resultado:

```
Filas: 10781 Columnas: 9

Antepenultima fila:

model           3 Series

year             2017

price           13100

transmission      Manual

mileage          25468

fuelType         Petrol

tax              200

mpg              42.8

engineSize         2.0

Name: 10778, dtype: object
```

**Ejercicio 9:**

Obtenga los atributos mileage, price y mpg en un nuevo DataFrame, y después seleccione aleatoriamente el 20% de los registros.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


df2 = df[["mileage", "price", "mpg"]]

print("Nuevo dataframe:")

print(df2)

print("Sample:")

print(df2.sample(frac=0.2))
```

Resultado:

```
Nuevo dataframe:
      mileage  price   mpg
0       67068  11200  57.6
1       14827  27000  42.8
2       62794  16000  51.4
3       26676  12750  72.4
4       39554  14500  50.4
...       ...    ...   ...
10776   40818  19000  54.3
10777   42947  14600  60.1
10778   25468  13100  42.8
10779   45000   9930  64.2
10780   59432  15981  57.6

[10781 rows x 3 columns]
Sample:
      mileage  price   mpg
3491     4003  30980  55.4
3591     5685  28980  52.3
```

7469  125000  4150  60.1

5785  60930  26462  47.1

9844  24999  17100  57.6

...  ...  ...  ...

5567  4830  41490  43.5

7136  6142  36995  39.2

3260  10  34990  54.3

4343  32351  19991  52.3

9474  27890  14999  57.6

[2156 rows x 3 columns]

**Ejercicio 10:**

Obtenga los registros que tengan un valor de mileage inferior a 10000 y un valor de mpg mayor que 40.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


df = df[df["mileage"]<10000]

df = df[df["mpg"]>40]

print(df)
```

Resultado:

```
        model  year  price transmission  mileage fuelType  tax   mpg  engineSize
131    1 Series  2017  14600    Automatic     5615   Petrol  145  58.9         1.5
148    1 Series  2016  13700       Manual     8719   Petrol  125  52.3         1.5
153    1 Series  2016  13750    Automatic     8707   Petrol   30  55.5         1.5
166          X1  2020  31498    Semi-Auto     1560   Diesel  145  60.1         2.0
167    2 Series  2020  27998       Manual     1580   Petrol  150  43.5         1.5
...         ...   ...    ...          ...      ...      ...  ...   ...         ...
10713  3 Series  2020  23899    Automatic     1255   Petrol  150  47.9         2.0
10739  3 Series  2019  23987    Automatic     1049   Petrol  150  47.9         2.0
10740  3 Series  2019  23454    Automatic     3551   Petrol  150  47.9         2.0
10741  3 Series  2019  23599    Automatic     2784   Petrol  145  47.9         2.0
10742  3 Series  2019  23499    Automatic     5634   Petrol  145  47.9         2.0

[3079 rows x 9 columns]
```

**Ejercicio 11:**

Modifique los valores del atributo model, de tal manera que los valores " x Series" pasen a ser "Serie x", siendo x un número entre 1 y 9.

```
import pandas as pd
df = pd.read_csv("bmw.csv")


def model_order(model):
   if("Series" in model):
      texto = model.split()
      return texto[1]+" "+texto[0]
   return model


df['model'] = df['model'].apply(model_order)
print(df)
```

Resultado:

```
       model  year  price transmission  mileage fuelType  tax   mpg  engineSize
0     Series 5  2014  11200    Automatic    67068   Diesel  125  57.6         2.0
1     Series 6  2018  27000    Automatic    14827   Petrol  145  42.8         2.0
2     Series 5  2016  16000    Automatic    62794   Diesel  160  51.4         3.0
3     Series 1  2017  12750    Automatic    26676   Diesel  145  72.4         1.5
4     Series 7  2014  14500    Automatic    39554   Diesel  160  50.4         3.0
...        ...   ...    ...          ...      ...      ...  ...   ...         ...
10776       X3  2016  19000    Automatic    40818   Diesel  150  54.3         2.0
10777  Series 5  2016  14600    Automatic    42947   Diesel  125  60.1         2.0
10778  Series 3  2017  13100       Manual    25468   Petrol  200  42.8         2.0
10779  Series 1  2014   9930    Automatic    45000   Diesel   30  64.2         2.0
10780       X1  2017  15981    Automatic    59432   Diesel  125  57.6         2.0

[10781 rows x 9 columns]
```

**Ejercicio 12:**

Inserte un nuevo registro con los siguientes datos: model=" 3 Series", year=2023, price = 22572, transmission = "Automatic", mileage = 74120, fuelType = "Diesel", tax = 160, mpg = 58.4, engineSize = 2.0

```
import pandas as pd

df = pd.read_csv("bmw.csv")


df.loc[df.shape[0]] = [" 3 Series", 2023, 22572, "Automatic",  74120, "Diesel",  160,  58.4,  2.0]

print(df)
```

Resultado:

```
        model  year  price transmission  mileage fuelType  tax   mpg  engineSize
0      5 Series  2014  11200    Automatic    67068   Diesel  125  57.6         2.0
1      6 Series  2018  27000    Automatic    14827   Petrol  145  42.8         2.0
2      5 Series  2016  16000    Automatic    62794   Diesel  160  51.4         3.0
3      1 Series  2017  12750    Automatic    26676   Diesel  145  72.4         1.5
4      7 Series  2014  14500    Automatic    39554   Diesel  160  50.4         3.0
...         ...   ...    ...          ...      ...      ...  ...   ...         ...
10777   5 Series  2016  14600    Automatic    42947   Diesel  125  60.1         2.0
10778   3 Series  2017  13100       Manual    25468   Petrol  200  42.8         2.0
10779   1 Series  2014   9930    Automatic    45000   Diesel   30  64.2         2.0
10780         X1  2017  15981    Automatic    59432   Diesel  125  57.6         2.0
10781   3 Series  2023  22572    Automatic    74120   Diesel  160  58.4         2.0


[10782 rows x 9 columns]
```

**Ejercicio 13:**

Convierta el DataFrame en un ndarray de numpy, e imprima el tipo de datos del ndarray obtenido.

```
import pandas as pd

df = pd.read_csv("bmw.csv")


df = df.to_numpy()

print(type(df))
```

Resultado:

```
<class 'numpy.ndarray'>
```

**Ejercicio 14:**

Calcule para cada registro el número medio de millas recorridas cada año.

```
import pandas as pd

df = pd.read_csv("bmw.csv")

print(df.groupby(["model", "year"])["mileage"].mean())
```

Resultado:

```
model     year
1 Series  2001     22633.0
          2004    112000.0
          2005     63547.0
          2006     89000.0
          2007    167000.0
                    ...
i8        2015     45075.0
          2016     10087.0
          2017     21233.0
          2018      9888.0
          2019      2062.0
Name: mileage, Length: 222, dtype: float64
```