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


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


df = pd.read_csv('/content/drive/MyDrive/my_data/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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


df = pd.read_csv('/content/drive/MyDrive/my_data/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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


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


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

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

Resultado:

```
9       36099
10      74907
16      55594
19      47348
20      75124
       ...
10754    66500
10756    63000
10761    34769
10763    68810
10769    39347
Name: mileage, Length: 1383, 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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


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


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

Resultado:

```
3524     7734
2621      103
10266    26589
3960     1000
6910    24229
       ...
4995    11659
8617    34000
1802    51536
10231    89823
1649      123
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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


df = pd.read_csv('/content/drive/MyDrive/my_data/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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


df = pd.read_csv('/content/drive/MyDrive/my_data/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

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

df = pd.read_csv('/content/drive/MyDrive/my_data/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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


df = pd.read_csv('/content/drive/MyDrive/my_data/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


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


df = pd.read_csv('/content/drive/MyDrive/my_data/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

2758 12900 33775 53.3

9307 70000 9950 55.4

9200 71720 12795 68.9

2506 101 42124 39.2

7010 4509 59991 34.9

... ...

4165 20376 17480 52.3

5505 20400 20995 134.5

9170 20698 11988 78.5

6900 2150 32398 42.2

3632 9500 25295 39.8

[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


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


df = pd.read_csv('/content/drive/MyDrive/my_data/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


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


df = pd.read_csv('/content/drive/MyDrive/my_data/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

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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


df = pd.read_csv('/content/drive/MyDrive/my_data/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


# Conectar a Google Drive

from google.colab import drive

drive.mount('/content/drive')


df = pd.read_csv('/content/drive/MyDrive/my_data/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

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

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

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

Resultado:

```
year
1996     36000.000000
1997     49000.000000
1998     56500.000000
1999     66557.750000
2000     96104.500000
2001     57544.333333
2002     86014.833333
2003     53500.000000
2004     89496.666667
2005     85091.166667
2006     87275.857143
2007    103427.125000
2008     88601.434783
2009     90244.766667
2010     86244.609756
2011     77413.078431
```

2012 70765.563025

2013 59341.366947

2014 53221.033932

2015 47774.313449

2016 38237.280021

2017 28663.936665

2018 17135.560142

2019 5612.342324

2020 1495.342428

Name: mileage, dtype: float64