

Import pandas

```
In [1]: import pandas as pd
```

Import csv

```
In [2]: df = pd.read_csv("./bmw.csv")
```

Ejercicio 1

```
In [3]: df[:10]
```

```
Out[3]:
```

	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

```
In [4]: serie = df["year"]
size = serie.size
tipo = serie.dtype
print("Tipo:", tipo, "; Size:", size)
```

Tipo: int64 ; Size: 10781

Ejercicio 3

```
In [5]: serie = df["mileage"]
serie[7::7]
```

```
Out[5]: 7          10401
        14          19057
        21          78957
        28          96213
        35          61818
        ...
        10752       41500
        10759       54008
        10766       54987
        10773       60372
        10780       59432
Name: mileage, Length: 1540, dtype: int64
```

Ejercicio 4

```
In [6]: serie = df["mileage"]
        serie.sample(frac=.4)
```

```
Out[6]: 9007       94347
        6060       6568
        7880       20671
        2849       11602
        8796       4971
        ...
        6587       6742
        3270        105
        35         61818
        1866       26965
        9726       19330
Name: mileage, Length: 4312, dtype: int64
```

Ejercicio 5

```
In [7]: serie = df["mileage"]
        indices = serie < 20000
        serie[indices]
```

```
Out[7]: 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

```
In [8]: serie = df["mpg"]
        pd.Series.sort_values(serie)
```

```
Out[8]: 6172      5.5
        1237      5.5
        6132      5.5
        6965      5.5
        2116      5.5
        ...
        7347    470.8
        5176    470.8
        10080    470.8
        8375    470.8
        9924    470.8
Name: mpg, Length: 10781, dtype: float64
```

Ejercicio 7

```
In [9]: serie = df["engineSize"]
media = serie.mean()
desv = serie.std()
minimum = serie.min()
maximum = serie.max()
print(media, desv, minimum, maximum)

2.1677673685186902 0.5520537772398375 0.0 6.6
```

Ejercicio 8

```
In [10]: filas,columnas = df.shape
print("Filas:",filas,"Columnas:",columnas)
print("Antepenúltimo registro:")
print(df.iloc[-3])

Filas: 10781 Columnas: 9
Antepenúltimo registro:
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

```
In [11]: nuevoDf = df[["mileage", "price", "mpg"]]
nuevoDf.sample(frac=.2)
```

Out [11]:

	mileage	price	mpg
532	8441	21498	48.7
9559	87743	8999	72.4
8319	9564	24950	60.1
6788	19164	15995	55.5
3352	103	33850	37.2
...
10211	18716	13900	65.7
241	5000	78000	31.4
4511	33546	16498	67.3
567	37842	17410	65.7
4262	19470	25000	51.4

2156 rows × 3 columns

Ejercicio 10

```
In [12]: indices = (df["mileage"]<10000) & (df["mpg"]>40)
df[indices]
```

Out [12]:

	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 × 9 columns

Ejercicio 11

```
In [13]: dfModificado = df.copy()
mapeo_modelo = {
    " 1 Series": "Serie 1",
    " 2 Series": "Serie 2",
    " 3 Series": "Serie 3",
    " 4 Series": "Serie 4",
    " 5 Series": "Serie 5",
    " 6 Series": "Serie 6",
    " 7 Series": "Serie 7",
    " 8 Series": "Serie 8",
    " 9 Series": "Serie 9"
}

dfModificado['model'] = dfModificado['model'].replace(mapeo_modelo)
dfModificado
```

Out [13]:

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	Serie 5	2014	11200	Automatic	67068	Diesel	125	57.6	2.0
1	Serie 6	2018	27000	Automatic	14827	Petrol	145	42.8	2.0
2	Serie 5	2016	16000	Automatic	62794	Diesel	160	51.4	3.0
3	Serie 1	2017	12750	Automatic	26676	Diesel	145	72.4	1.5
4	Serie 7	2014	14500	Automatic	39554	Diesel	160	50.4	3.0
...
10776	X3	2016	19000	Automatic	40818	Diesel	150	54.3	2.0
10777	Serie 5	2016	14600	Automatic	42947	Diesel	125	60.1	2.0
10778	Serie 3	2017	13100	Manual	25468	Petrol	200	42.8	2.0
10779	Serie 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 × 9 columns

Ejercicio 12

```
In [14]: dfNuevo = df.copy()
registro = {
    "model": " 3 Series",
    "year": 2023,
    "price" : 22572,
    "transmission" : "Automatic",
    "mileage" : 74120,
    "fuelType" : "Diesel",
    "tax" : 160,
    "mpg" : 58.4,
    "engineSize" : 2.0
}
nuevoReg = pd.DataFrame([registro])
dfNuevo = pd.concat([dfNuevo, nuevoReg], ignore_index=True)
dfNuevo
```

Out [14]:

	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 × 9 columns

Ejercicio 13

```
In [15]: ndarray = df.values
print(ndarray)
type(ndarray)
```

```
[[' 5 Series' 2014 11200 ... 125 57.6 2.0]
 [' 6 Series' 2018 27000 ... 145 42.8 2.0]
 [' 5 Series' 2016 16000 ... 160 51.4 3.0]
 ...
 [' 3 Series' 2017 13100 ... 200 42.8 2.0]
 [' 1 Series' 2014 9930 ... 30 64.2 2.0]
 [' X1' 2017 15981 ... 125 57.6 2.0]]
```

Out [15]: numpy.ndarray

Ejercicio 14

```
In [16]: import datetime
currentYear = datetime.datetime.now().year
dfNuevo = df.copy()
dfNuevo["mile/year"] = df["mileage"] / (currentYear - df["year"])
dfNuevo["mile/year"]
```

```
Out[16]: 0          6706.800000
         1          2471.166667
         2          7849.250000
         3          3810.857143
         4          3955.400000
         ...
        10776       5102.250000
        10777       5368.375000
        10778       3638.285714
        10779       4500.000000
        10780       8490.285714
Name: mile/year, Length: 10781, dtype: float64
```