Multiple Linear Regression - Exercise

You are given a real estate dataset.

Real estate is one of those examples that every regression course goes through as it is extremely easy to understand and there is a (almost always) certain causal relationship to be found.

The data is located in the file: 'real_estate_price_size_year.csv'.

You are expected to create a multiple linear regression (similar to the one in the lecture), using the new data.

In this exercise, the dependent variable is 'price', while the independent variables are 'size' and 'year'.

Good luck!

Import the relevant libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn
seaborn.set()
```

Load the data

In [2]:

```
data = pd.read_csv('real_estate_price_size_year.csv')
```

In [3]:
```
data
```

Out[3]:
```
         price     size  year
0   234314.144   643.09  2015
1   228581.528   656.22  2009
2   281626.336   487.29  2018
3   401255.608  1504.75  2015
4   458674.256  1275.46  2009
...        ...      ...   ...
95  252460.400   549.80  2009
96  310522.592  1037.44  2009
97  383635.568  1504.75  2006
98  225145.248   648.29  2015
99  274922.856   705.29  2006

100 rows × 3 columns
```

In [4]:
```
data.head()
```

Out[4]:
```
        price     size  year
0  234314.144   643.09  2015
1  228581.528   656.22  2009
2  281626.336   487.29  2018
3  401255.608  1504.75  2015
4  458674.256  1275.46  2009
```

In [5]:

```
data.describe()
```

Out[5]:

	price	size	year
count	100.000000	100.000000	100.000000
mean	292289.470160	853.024200	2012.600000
std	77051.727525	297.941951	4.729021
min	154282.128000	479.750000	2006.000000
25%	234280.148000	643.330000	2009.000000
50%	280590.716000	696.405000	2015.000000
75%	335723.696000	1029.322500	2018.000000
max	500681.128000	1842.510000	2018.000000

Create the regression

Declare the dependent and the independent variables

In [7]:

```
y = data['price']
x1 = data[['size', 'year']]
```

Regression

In [8]:

```
x = sm.add_constant(x1)
results = sm.OLS(y,x).fit()
```

In [9]:
```
results.summary()
```

Out[9]:
```
OLS Regression Results

       Dep. Variable:            price          R-squared:      0.776
               Model:              OLS      Adj. R-squared:     0.772
              Method:    Least Squares         F-statistic:     168.5
                Date: Wed, 25 Aug 2021  Prob (F-statistic):  2.77e-32
                Time:         22:06:01      Log-Likelihood:   -1191.7
    No. Observations:              100                 AIC:     2389.
        Df Residuals:               97                 BIC:     2397.
            Df Model:                2
     Covariance Type:        nonrobust

                coef    std err        t  P>|t|     [0.025     0.975]
const  -5.772e+06   1.58e+06   -3.647  0.000  -8.91e+06  -2.63e+06
 size    227.7009     12.474   18.254  0.000    202.943    252.458
 year   2916.7853    785.896    3.711  0.000   1357.000   4476.571

        Omnibus: 10.083     Durbin-Watson:     2.250
  Prob(Omnibus):  0.006  Jarque-Bera (JB):     3.678
           Skew:  0.095          Prob(JB):     0.159
       Kurtosis:  2.080          Cond. No.  9.41e+05


Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.41e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:
```

```