# Simple Linear Regression Exercise

July 19, 2024

## 1 Simple linear regression

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.csv'.

In this exercise, the dependent variable is 'price', while the independent variable is 'size'.

### 1.1 Import the relevant libraries

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

### 1.2 Load the data

[ ]:
```
data = pd.read_csv('real_estate_price_size.csv')
```

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

[ ]:

| | price | size |
|---|---|---|
| 0 | 234314.144 | 643.09 |
| 1 | 228581.528 | 656.22 |
| 2 | 281626.336 | 487.29 |
| 3 | 401255.608 | 1504.75 |
| 4 | 458674.256 | 1275.46 |

[ ]:
```
data.describe()
```

[ ]:

| | price | size |
|---|---|---|
| count | 100.000000 | 100.000000 |
| mean | 292289.470160 | 853.024200 |
| std | 77051.727525 | 297.941951 |
| min | 154282.128000 | 479.750000 |
| 25% | 234280.148000 | 643.330000 |
| 50% | 280590.716000 | 696.405000 |
| 75% | 335723.696000 | 1029.322500 |

```
max     500681.128000  1842.510000
```

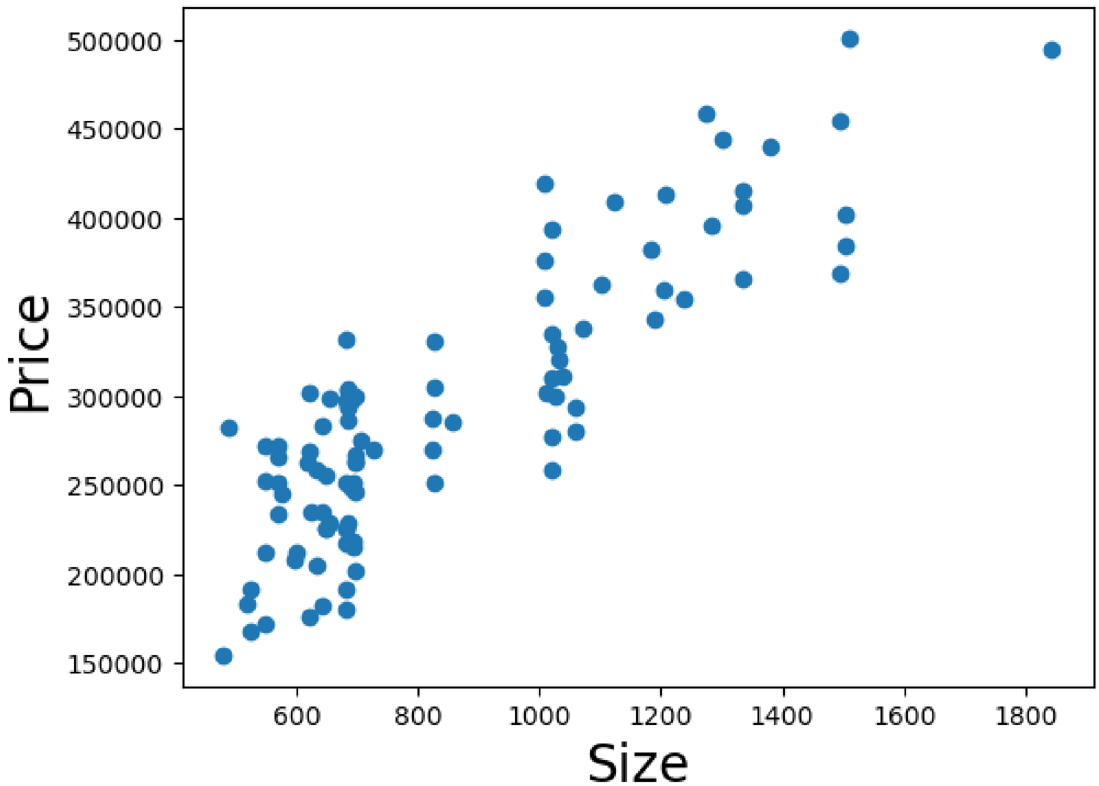
## 1.3 Create the regression

### 1.3.1 Declare the dependent and the independent variables

[ ]:
```
y = data['price']
x1 = data['size']
```

### 1.3.2 Explore the data

[ ]:
```
plt.scatter(x1,y)
plt.xlabel('Size',fontsize=20)
plt.ylabel('Price',fontsize=20)
plt.show()
```




### 1.3.3 Regression itself

[ ]:
```
x = sm.add_constant(x1)
results = sm.OLS(y,x).fit()
results.summary()
```

[ ]:

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.745 |
| **Model:** | OLS | **Adj. R-squared:** | 0.742 |
| **Method:** | Least Squares | **F-statistic:** | 285.9 |
| **Date:** | Fri, 19 Jul 2024 | **Prob (F-statistic):** | 8.13e-31 |
| **Time:** | 12:02:17 | **Log-Likelihood:** | -1198.3 |
| **No. Observations:** | 100 | **AIC:** | 2401. |
| **Df Residuals:** | 98 | **BIC:** | 2406. |
| **Df Model:** | 1 | | |
| **Covariance Type:** | nonrobust | | |

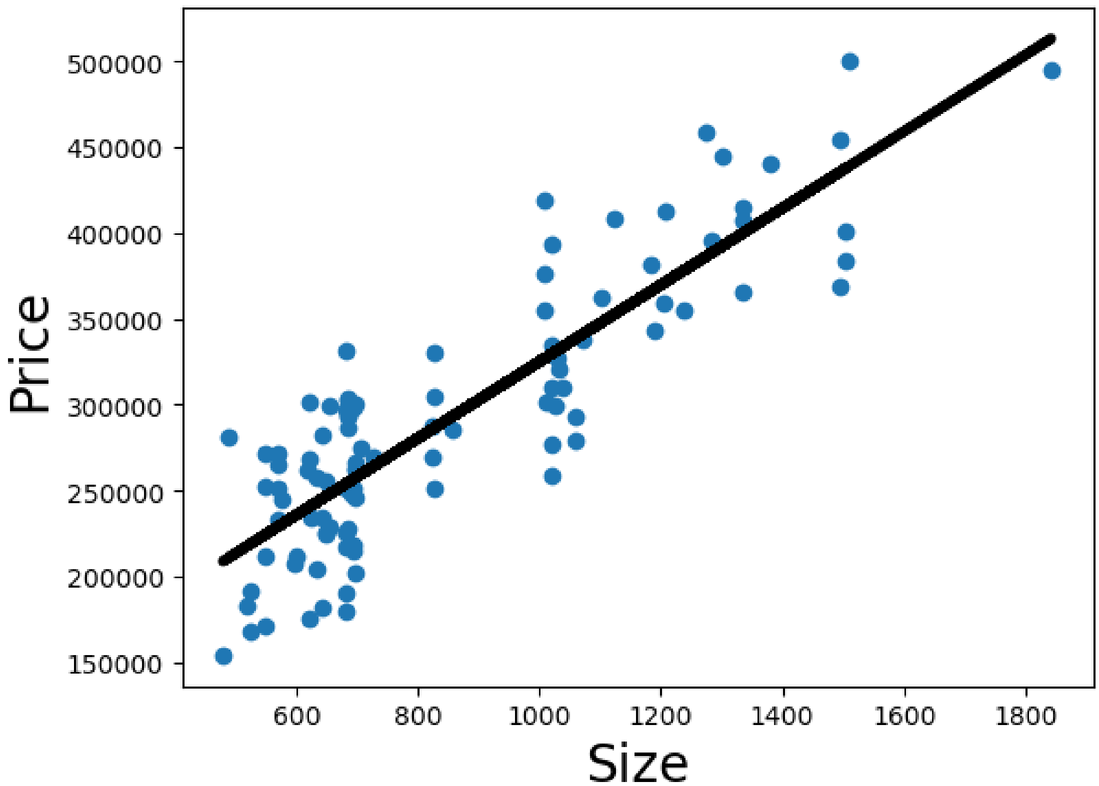
| | **coef** | **std err** | **t** | **P>\|t\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **const** | 1.019e+05 | 1.19e+04 | 8.550 | 0.000 | 7.83e+04 | 1.26e+05 |
| **size** | 223.1787 | 13.199 | 16.909 | 0.000 | 196.986 | 249.371 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 6.262 | **Durbin-Watson:** | 2.267 |
| **Prob(Omnibus):** | 0.044 | **Jarque-Bera (JB):** | 2.938 |
| **Skew:** | 0.117 | **Prob(JB):** | 0.230 |
| **Kurtosis:** | 2.194 | **Cond. No.** | 2.75e+03 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.75e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

[ ]:

```
plt.scatter(x1,y)
yhat = x1*223.1787+101900
fig = plt.plot(x1,yhat, lw=4, c='black', label ='regression line')
plt.xlabel('Size', fontsize = 20)
plt.ylabel('Price', fontsize = 20)
plt.show()
```



Price
Size
500000
450000
400000
350000
300000
250000
200000
150000
600
800
1000
1200
1400
1600
1800


[ ]: