

Simple linear regression - Exercise Solution

real estate dataset file: 'real_estate_price_size.csv'

Create a simple linear regression.

Apart from that:

- Create a scatter plot (with or without a regression line)
- Calculate the R-squared
- Display the intercept and coefficient(s)
- Using the model make a prediction about an apartment with size 750 sq.ft.

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

Libraries

```
In [15]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

from sklearn.linear_model import LinearRegression
```

Load the data

```
In [2]: data = pd.read_csv('real_estate_price_size.csv')
data.head()
```

```
Out[2]:
```

	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

```
In [3]: data.describe()
```

```
Out[3]:
```

	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

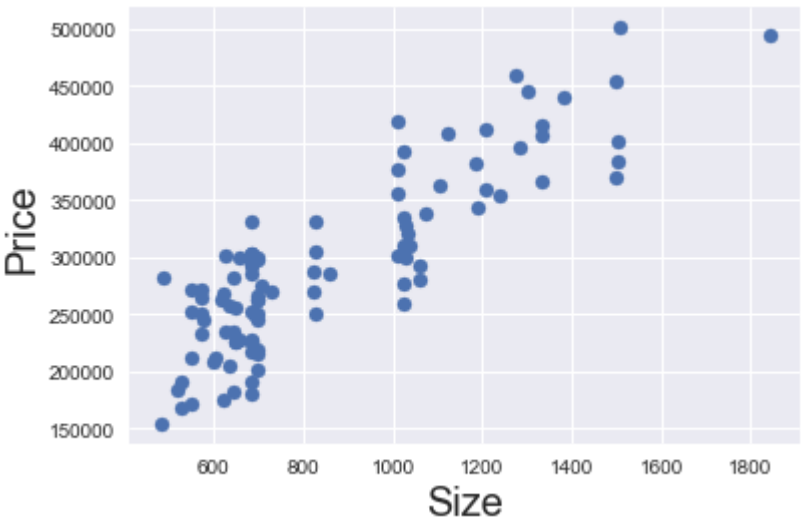
Create the regression

Declare the dependent and the independent variables

```
In [5]: x = data['size']
y = data['price']
```

Explore the data

```
In [6]: plt.scatter(x,y)
plt.xlabel('Size',fontsize=20)
plt.ylabel('Price',fontsize=20)
plt.show()
```



Transform the inputs into a matrix (2D object)

```
In [7]: x_matrix = x.values.reshape(-1,1)
```

Regression

```
In [9]: reg = LinearRegression()
reg.fit(x_matrix,y)
```

```
Out[9]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

R-squared

```
In [10]: reg.score(x_matrix,y)
```

```
Out[10]: 0.7447391865847586
```

Intercept

```
In [12]: reg.intercept_
```

```
Out[12]: 101912.60180122912
```

Coefficients

```
In [13]: reg.coef_
```

```
Out[13]: array([223.17874259])
```

Making predictions

You find an apartment online with a size of 750 sq.ft.

All else equal what should be its price according to the model?

```
In [14]: reg.predict(750)
```

```
Out[14]: array([269296.65874718])
```