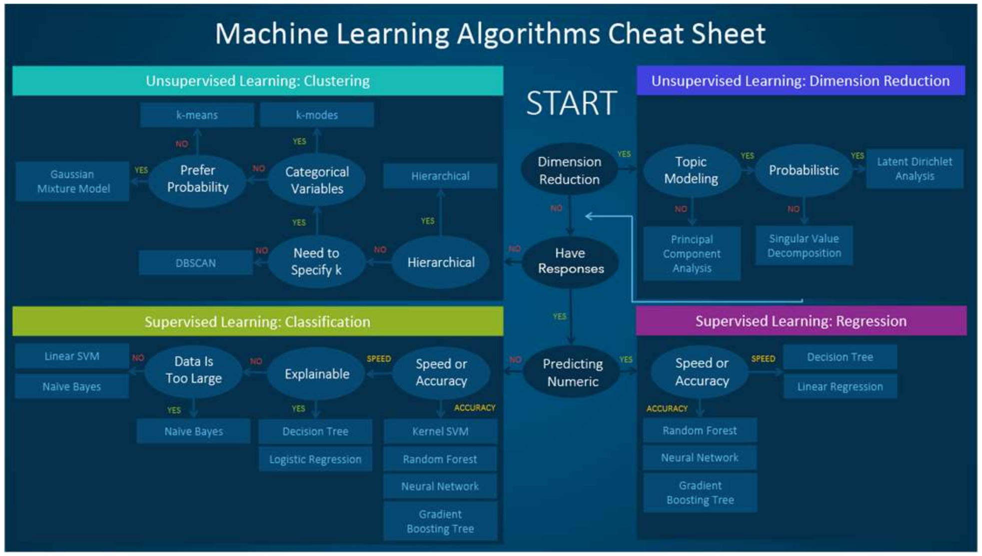
SYNTRA PXL

This year:

- Verder uitdiepen regression, tree based, clustering….



- Meer met github werken
- Meer model implementatie

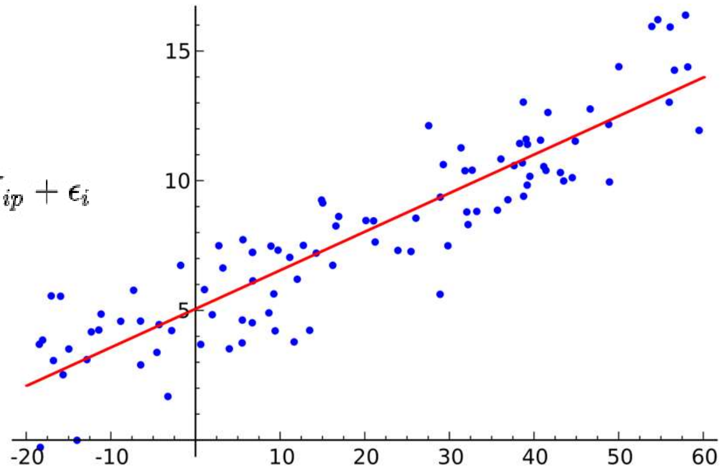
Simple Linear Regression

Simple Linear Regression

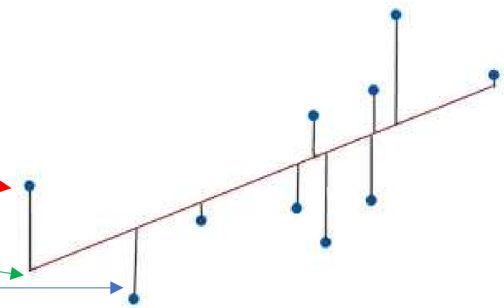
huisprijs

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + \epsilon_i$$

Factoren die prijzen bepalen zoals volume, bouwjaar, locatie



Lineaire regressie identificeert de vergelijking/lijn die het kleinste verschil oplevert tussen alle **waargenomen** waarden en hun **geschatte** waarden.
Om precies te zijn, vindt lineaire regressie de kleinste som van kwadratische **residuen** die mogelijk is voor de dataset.



SYNTRA PXL

Simple Linear Regression

What is a Cost Function?

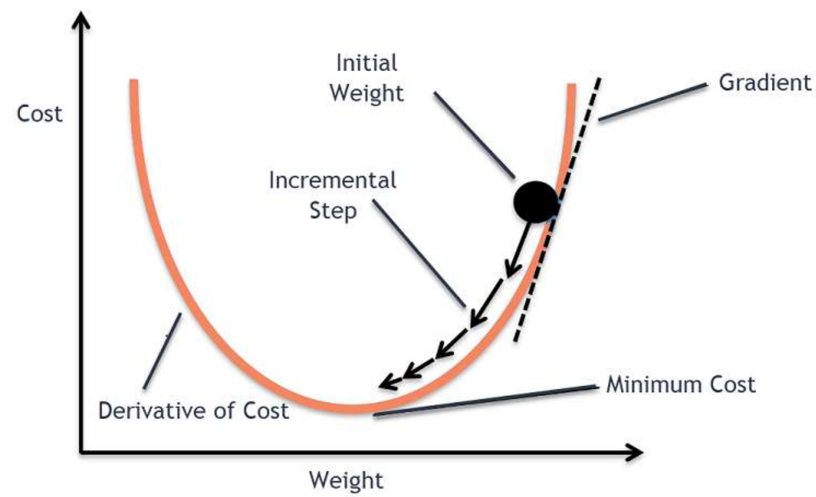
- It is a function that measures the performance of a model for any given data.
- Cost Function quantifies the **error between predicted values and expected values** and presents it in the form of a single real number.

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$



Simple Linear Regression

Opdracht:

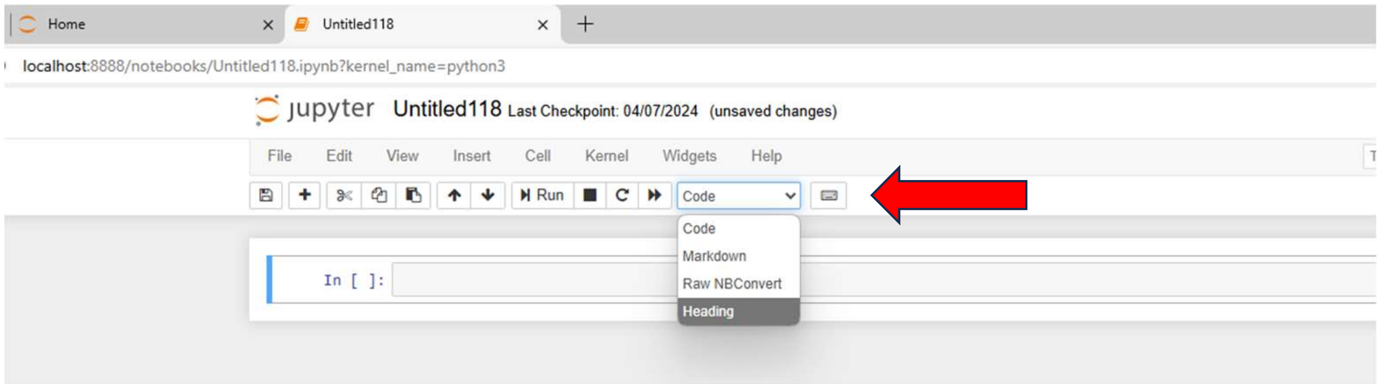
The real estate market is a dynamic and complex environment, and accurately predicting house prices is crucial for various stakeholders, including buyers, sellers, and investors. This project aims to develop a predictive model that can enhance decision-making in the real estate domain.

Objectives

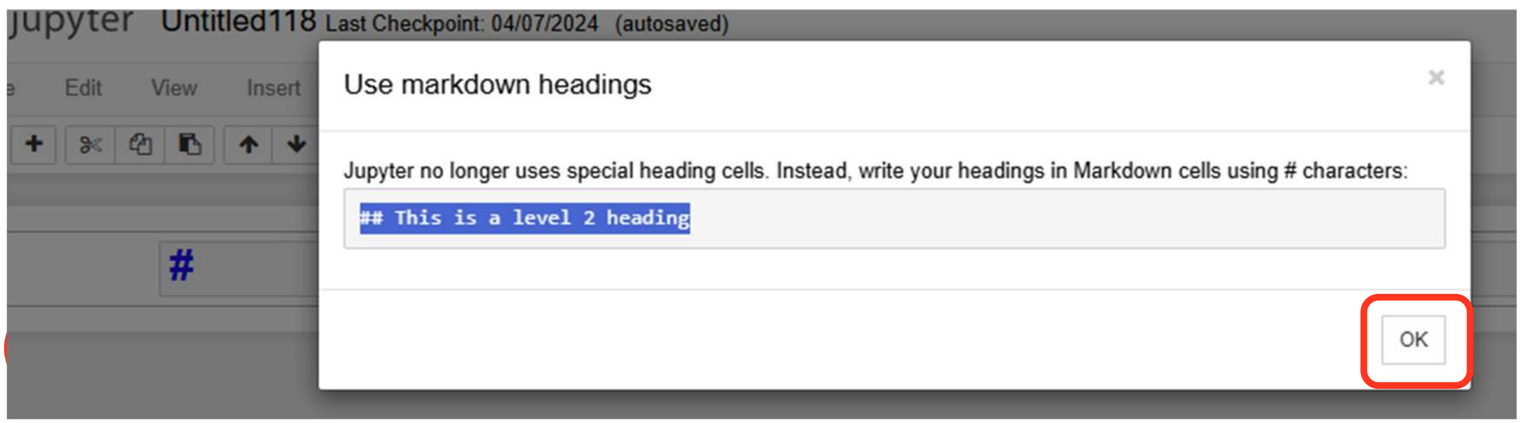
Develop a robust regression model for predicting house prices.
Implement effective outlier treatment of numerical variables and feature engineering techniques.
Explore and visualize data relationships through EDA (Exploratory Data Analysis).
Apply encoding methods for categorical variables.
Deploy the model and develop an estimation tool in Excel

Simple Linear Regression

- Open Jupyter notebook
- Choose Code > Heading

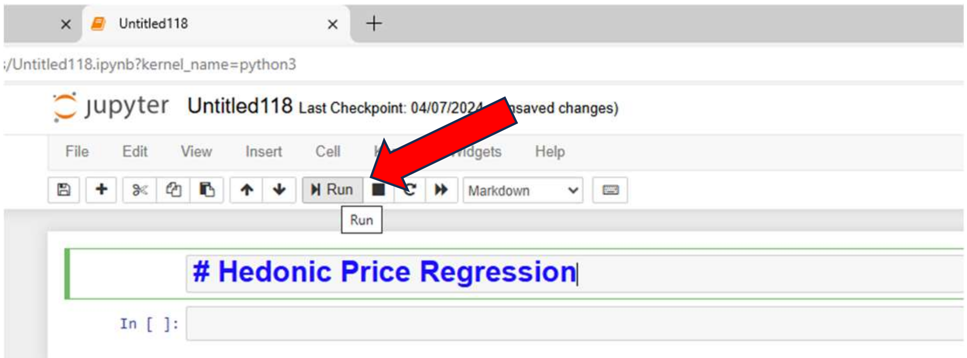


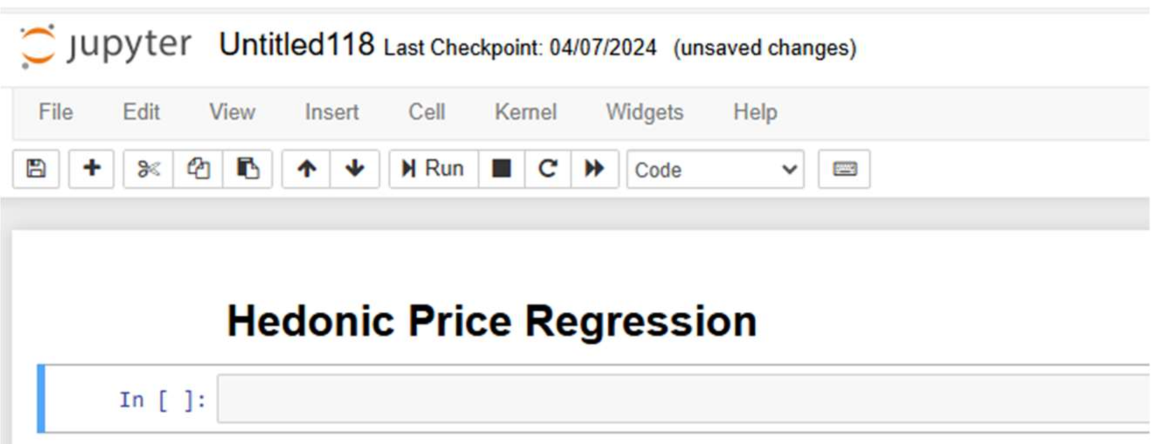
- Press “OK”



Simple Linear Regression

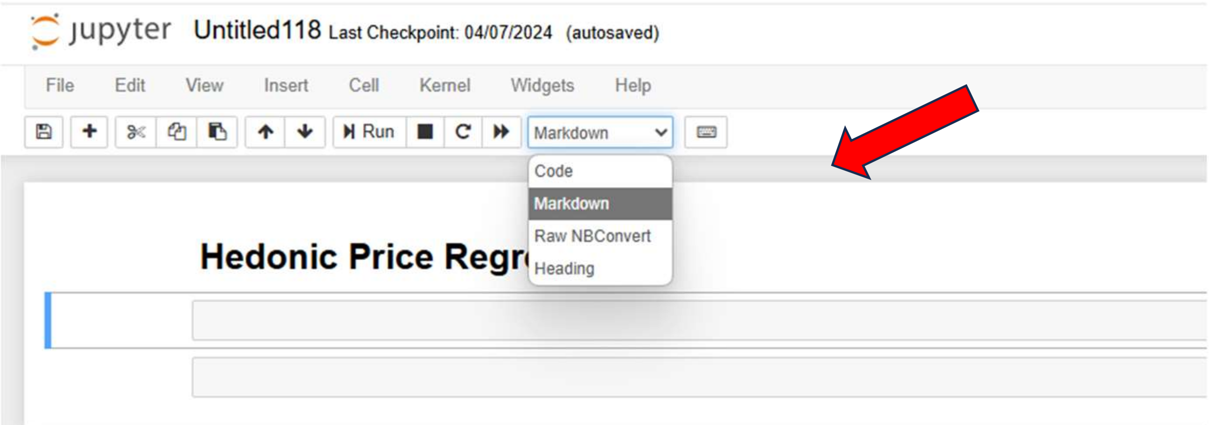
- Type in “Hedonic Price Estimation”
- Press “Run”



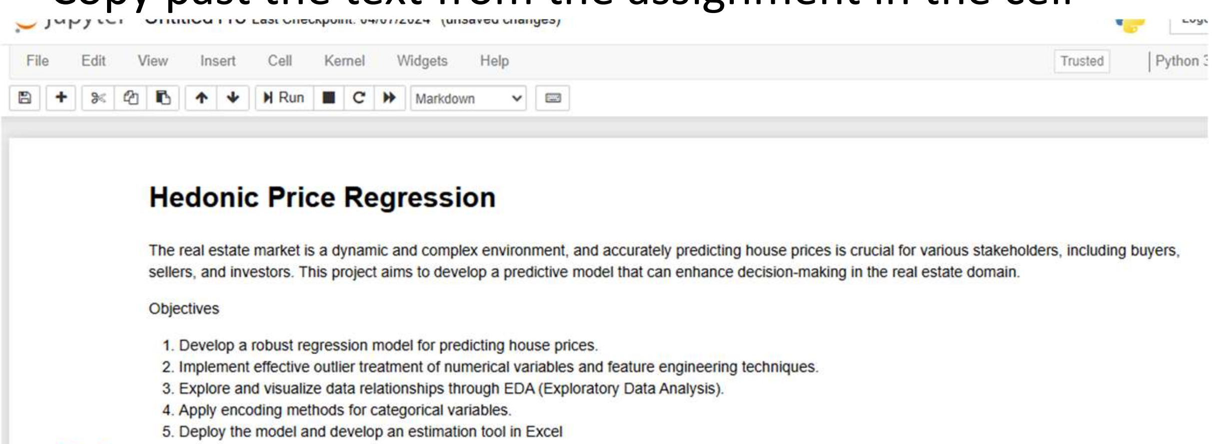


Simple Linear Regression

- Select “Markdown”



- Copy past the text from the assignment in the cell



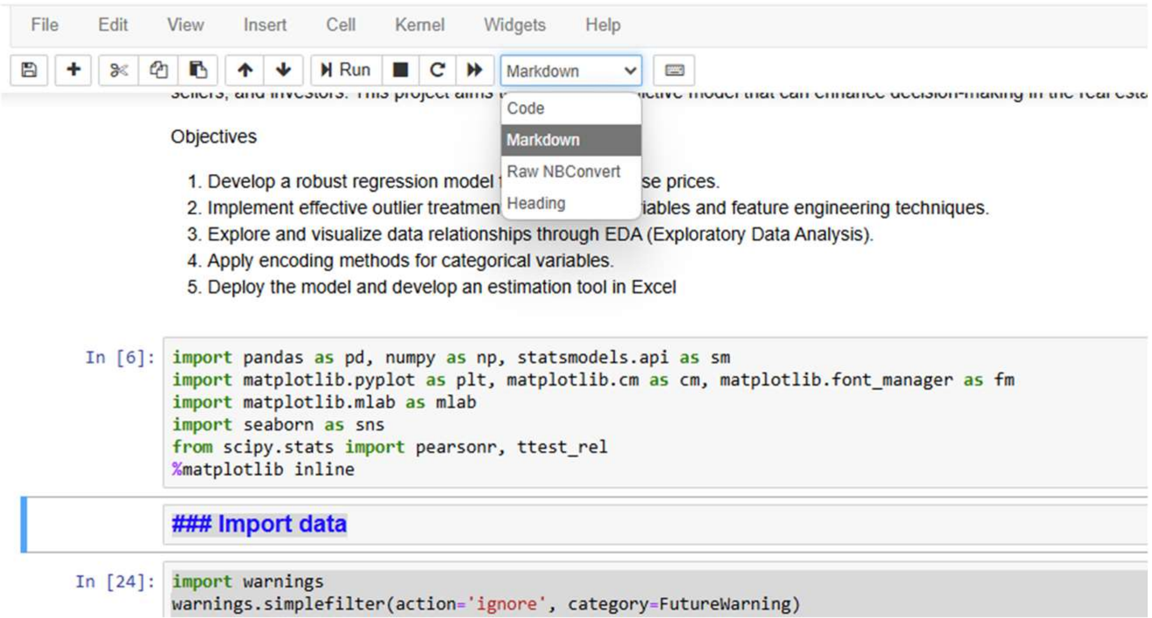
Simple Linear Regression

- Import libraries

```
import pandas as pd, numpy as np, statsmodels.api as sm
import matplotlib.pyplot as plt, matplotlib.cm as cm, matplotlib.font_manager as fm
import matplotlib.mlab as mlab
import seaborn as sns
from scipy.stats import pearsonr, ttest_rel
from scipy.stats import spearmanr
from scipy.stats import kendalltau
%matplotlib inline
```

Simple Linear Regression

- Import data




```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

data="C://Users/bours/Documents/Les_Syntra/DataScience2/les_1/bestanden/belgium_community_les1.csv"
immo = pd.read_csv(data,sep=';',header=0)
immo.head()
```

Simple Linear Regression

- Import data

	locality	type of property	subtype of property	price	sale type	number of rooms	area	furnished	open fire	terrace	terrace area	number of facades	building state
0	Mouscron	Appartement	None	204584	notariale	2	4	0	0	TRUE	40	2	2
1	Mouscron	Appartement	None	395000	notariale	6	212	0	0	None	None	2	1
2	Mouscron	Appartement	None	182500	notariale	2	50	0	0	None	None	2	1
3	Mouscron	Appartement	None	229500	notariale	2	70	0	0	None	None	2	1
4	Mouscron	Appartement	None	239500	notariale	3	50	0	0	None	None	2	1

Simple Linear Regression

- Check on missing values

```
immocopy = immo.copy()
immocopy.isnull().sum()
```

Simple Linear Regression

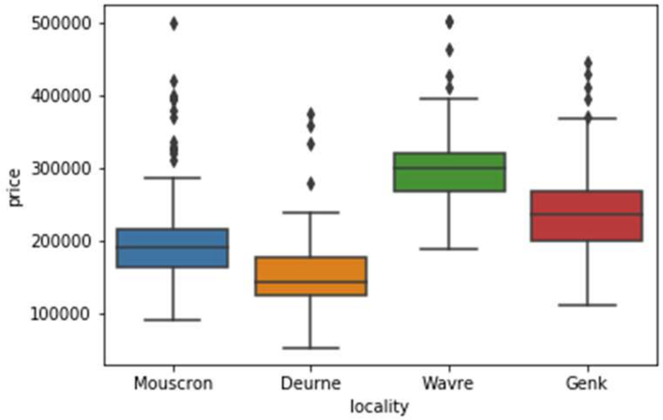
- EDA: significant prices difference between localities, number of rooms, building state
- Localities vs price

```
sns.boxplot(data=immo, x="locality", y="price")
```

1. localities vs price

```
sns.boxplot(data=immo, x="locality", y="price")
```

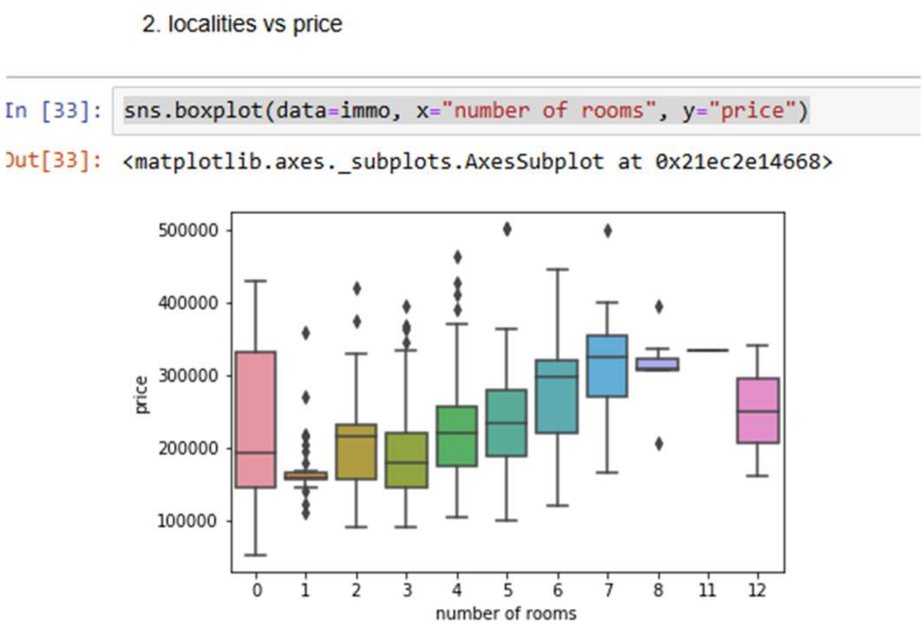
```
<matplotlib.axes._subplots.AxesSubplot at 0x21ec2da3860>
```



Simple Linear Regression

- EDA: significant prices difference between localities, number of rooms, building state
- number of rooms vs price

```
sns.boxplot(data=immo, x="number of rooms", y="price")
```



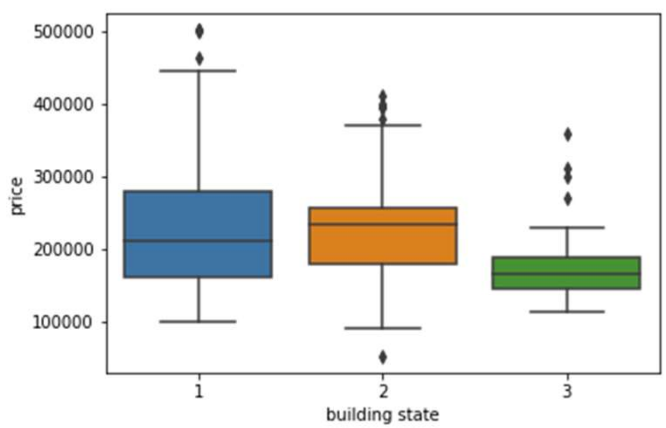
Simple Linear Regression

- EDA: significant prices difference between localities, number of rooms, building state
- Building state vs price

```
sns.boxplot(data=immo, x="number of rooms", y="price")
```

```
[35]: sns.boxplot(data=immo, x="building state", y="price")
```

```
:[35]: <matplotlib.axes._subplots.AxesSubplot at 0x21ec2f1dda0>
```

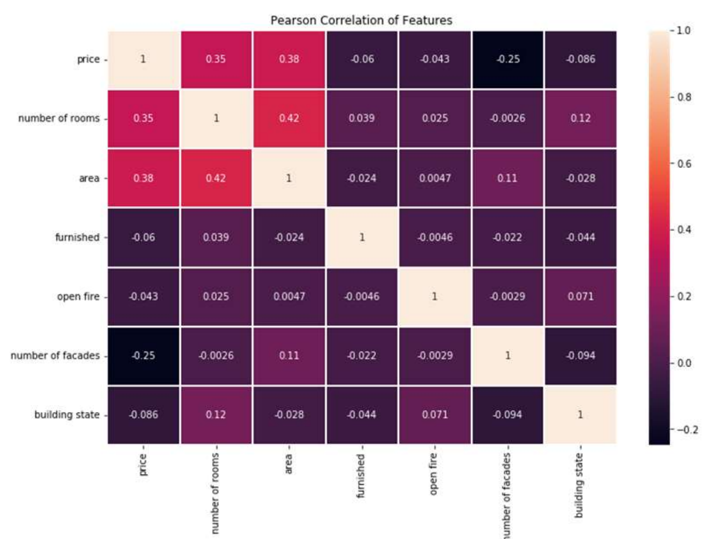


Simple Linear Regression

Correlation Matrix

- The correlation matrix shows the correlation between all the variables in the dataset. It will help you identify which variables are strongly correlated, for cases like simple linear regression or those that are not correlated, for cases when you're trying to models with multiple features adding to your model's performance.

```
corr = immo.corr()
f, ax = plt.subplots(figsize=(12, 8))
sns.heatmap(corr, annot=True, square=False, ax=ax, linewidth = 1)
plt.title('Pearson Correlation of Features')
```



Simple Linear Regression

- Correlation is a statistical measure that helps us understand the strength and direction of the linear relationship between two variables.

- The Pearson correlation coefficient, often used for continuous variables, ranges from -1 to 1.

  - A positive value indicates a positive correlation,
  - a negative value indicates a negative correlation
  - 0 indicates no correlation.

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

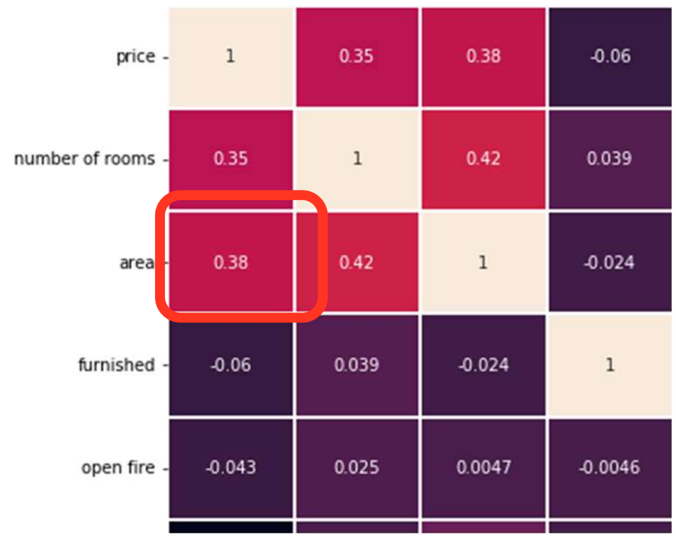
SYNTRA PXL

Simple Linear Regression

How is Correlation calculated ?

- Open **belgium_community_les1_calcs**

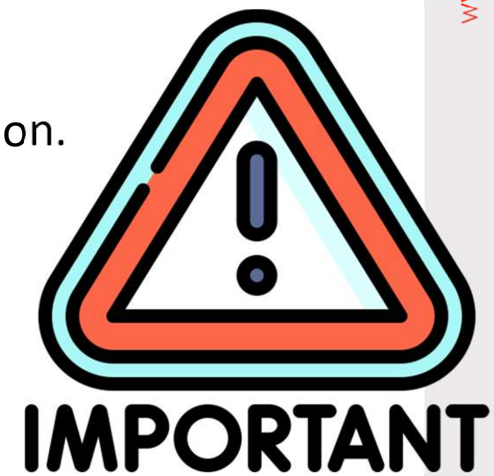
									Correlation	0,38			Average Price	222.813,49	Sum(SquarePrice)	3.073.902.768.827,41
													Average Area	93,31	Sum(Square Area)	1.264.683,25
type of property	subtype of property	price	sale type	number of rooms	area	furnished	open fire	terrace	terrace area	number of facades	building state		price	area	Square Price	Square Area
Appartement	None	204584	notariale	2	4	0	0	TRUE	40	2	2		- 18.229,49	- 89,31	332.314.192,32	7.975,43
Appartement	None	395000	notariale	6	212	0	0	None	None	2	1		172.186,51	118,69	29.648.195.296,50	14.088,45
Appartement	None	182500	notariale	2	50	0	0	None	None	2	1		- 40.313,49	- 43,31	1.625.177.225,34	1.875,34
Appartement	None	229500	notariale	2	70	0	0	None	None	2	1		6.686,51	- 23,31	44.709.457,55	543,13
Appartement	None	239500	notariale	3	50	0	0	None	None	2	1		16.686,51	- 43,31	278.439.719,72	1.875,34
Appartement	None	189500	notariale	2	45	0	0	None	None	2	1		- 33.313,49	- 48,31	1.109.788.408,86	2.333,40
Appartement	None	259900	notariale	2	104	0	0	None	None	2	1		37.086,51	10,69	1.375.409.454,56	114,38



Simple Linear Regression

- Significance and P-value: **will tell us if the correlation is significantly different from zero.**

A small p-value (< 0.05) suggests a statistically significant correlation.
A large p-value (> 0.05) suggests no significant correlation.



The test statistics for Pearson's correlation coefficient and Spearman's correlation coefficient have the same

$$t = \frac{r \times \sqrt{n-2}}{\sqrt{1-r^2}}$$

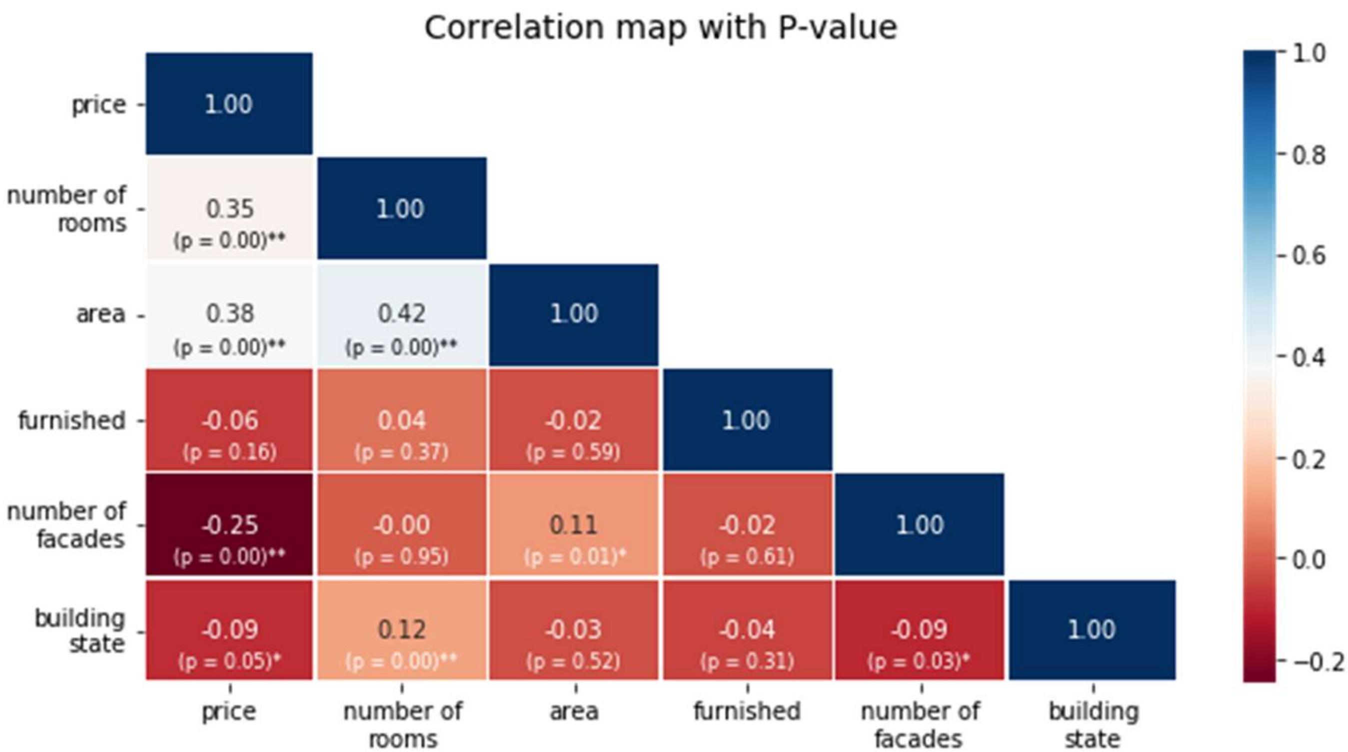
The p-value is 2 × P(T > t) where T follows a t distribution with n – 2 degrees of freedom.

Simple Linear Regression

- Generate a correlation matrix with significance

```
# Generate the correlation matrix afresh
corr = immo[['price','number of rooms','area','furnished','number of facades','building state']].corr()

# mask the correlation matrix to diagonal
mask = np.zeros_like(corr, dtype=bool)
mask[np.triu_indices_from(mask)] = True
np.fill_diagonal(mask, False)

fix,ax = plt.subplots(figsize=(10,5))
plt.title("Correlation map with P-value", fontsize=14)

# Generate heatmap
heatmap = sns.heatmap(corr,
            annot= True,
            annot_kws={"fontsize": 10},
            fmt='.2f',
            linewidths=0.5,
            cmap='RdBu',
            mask=mask,
            ax=ax)

# calculate and format p-values
p_values = np.full((corr.shape[0], corr.shape[1]), np.nan)
for i in range(corr.shape[0]):
  for j in range(i+1, corr.shape[1]):
    x = immocorr.iloc[:, i]
    y = immocorr.iloc[:, j]
    mask = ~np.logical_or(np.isnan(x), np.isnan(y))
    if np.sum(mask) > 0:
      p_values[i, j] = pearsonr(x[mask], y[mask])[1] #change to kendalltau or spearmanr

# Create a dataframe object for p_values
p_values = pd.DataFrame(p_values, columns=corr.columns, index=corr.index)

# Mask the p values
mask_pvalues = np.triu(np.ones_like(p_values), k=1)

# Generate maximum and minimum correlation coefficients for p-value annotation color
max_corr = np.max(corr.max())
min_corr = np.min(corr.min())

# Assign p-value annotations, include asterisks for significance
for i in range (p_values.shape[0]):
  for j in range(p_values.shape[1]):
    if mask_pvalues[i, j]:
      p_value = p_values.iloc[i, j]
      if not np.isnan(p_value):
        correlation_value = corr.iloc[i, j]
        text_color = 'white' if correlation_value >= (max_corr - 0.4) or correlation_value <= (min_corr + 0.4) else 'black'
        if p_value <= 0.01:
          #include double asterisks for p-value <= 0.01
          ax.text(i + 0.5, j + 0.8, f'(p = {p_value:.2f})**',
              horizontalalignment='center',
              verticalalignment='center',
              fontsize=8,
              color=text_color)
        elif p_value <= 0.05:
          #include single asterisk for p-value <= 0.05
          ax.text(i + 0.5, j + 0.8, f'(p = {p_value:.2f})*',
              horizontalalignment='center',
              verticalalignment='center',
              fontsize=8,
              color=text_color)
        else:
          ax.text(i + 0.5, j + 0.8, f'(p = {p_value:.2f})',
              horizontalalignment='center',
              verticalalignment='center',
              fontsize=8,
              color=text_color)

# Customize x-axis labels
x_labels = [textwrap.fill(label.get_text(), 13) for label in ax.get_xticklabels()]
ax.set_xticklabels(x_labels, rotation=0, ha="center")

# Customize y-axis labels
y_labels = [textwrap.fill(label.get_text(), 13) for label in ax.get_yticklabels()]
ax.set_yticklabels(y_labels, rotation=0, ha="right")

# Display the plot
plt.show()
```

Simple Linear Regression



** 99% are the times the correlation is significant from 0

* 95% are the times the correlation is significant from 0

Simple Linear Regression

Which are the most important variables to use ?

Simple Linear Regression

- Run the regression

```
#select predictor variables
x = immo[['area','number of facades','building state’]]

#select response variable
y = immo["price"]

#define response variable
y = y

#define predictor variables
x = x

#add constant to predictor variables
x = sm.add_constant(x)

#fit linear regression model
model = sm.OLS(y, x).fit()

#view model summary
print(model.summary())
```

Simple Linear Regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.239
Model:                            OLS   Adj. R-squared:                  0.235
Method:                 Least Squares   F-statistic:                     55.59
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           2.95e-31
Time:                        19:03:27   Log-Likelihood:                -6685.1
No. Observations:                 534   AIC:                         1.338e+04
Df Residuals:                     530   BIC:                         1.340e+04
Df Model:                           3
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const               2.34e+05   1.14e+04     20.446      0.000    2.11e+05    2.56e+05
area                638.7447     59.403     10.753      0.000     522.050     755.439
number of facades -3.423e+04   4357.544     -7.855      0.000   -4.28e+04   -2.57e+04
building state    -1.369e+04   5034.861     -2.719      0.007   -2.36e+04   -3797.258
==============================================================================
Omnibus:                       45.474   Durbin-Watson:                   1.307
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              107.223
Skew:                           0.455   Prob(JB):                     5.21e-24
Kurtosis:                       4.997   Cond. No.                         447.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates **the percentage of the variance in the dependent variable that the independent variables explain collectively**. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scal

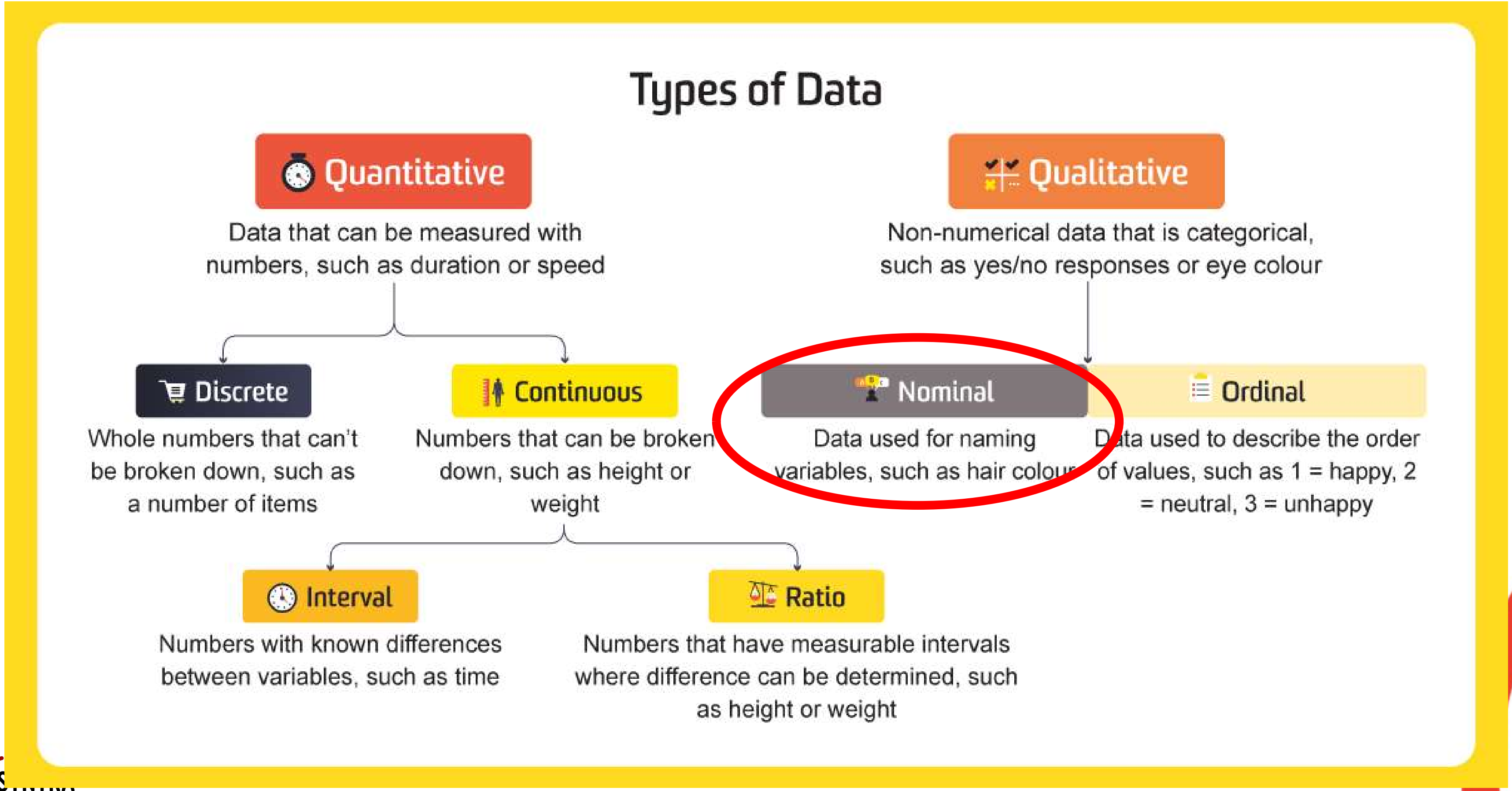
Simple Linear Regression

- 23% is not really good

- Maybe we can add **locality**

Simple Linear Regression

- **Locality** is like Deurne, Genk, Moucron, Wavre…



Simple Linear Regression

One Hot Encoding

- One hot encoding transforms categorical variables into a binary matrix, where each category is represented by a unique binary vector.

- This approach is particularly useful for algorithms that cannot work with categorical data directly, such as logistic regression.

- Essentially One-Hot encoding creates a binary column for each category, but only the active category is only set to 1 and all the other columns are set to 0.

Simple Linear Regression

- **One hot encoding**

Water	Temperature
A	Hot
B	Cold
C	Warm
D	Cold

Dummy Variables

Water	Temperature	var_hot	var_warm	var_cold
A	Hot	1	0	0
B	Cold	0	0	1
C	Warm	0	1	0
D	Cold	1	0	0

SYNTRA PXL

Simple Linear Regression

- **One hot encoding also known as Dummy Variable**

What is a Dummy Variable?
A dummy variable (is, an indicator variable) is a numeric variable that represents categorical data, such as gender, location, etc.

What are the benefits of a Dummy Variable?
Regression results are easiest to interpret when dummy variables are limited to two specific values, 1 or 0. Typically, 1 represents the presence of a qualitative attribute, and 0 represents the absence.

Simple Linear Regression

The sign of each coefficient indicates the direction of the relationship between a predictor variable and the response variable.

- A **positive** sign indicates that as the predictor variable increases, the Target variable also **increases**.

- A **negative** sign indicates that as the predictor variable increases, the Target variable **decreases**.

Simple Linear Regression

- Add locality and dummy variables

```
immo_dummy = pd.get_dummies(immo, columns = ['locality'])
immo_dummy.head()
```

[61]:

subtype of property	price	sale type	number of rooms	area	furnished	open fire	terrace	terrace area	number of facades	building state	locality_Deurne	locality_Genk	locality_Mouscron	locality_Wavre
None	204584	notariale	2	4	0	0	TRUE	40	2	2	0	0	1	0
None	395000	notariale	6	212	0	0	None	None	2	1	0	0	1	0
None	182500	notariale	2	50	0	0	None	None	2	1	0	0	1	0
None	229500	notariale	2	70	0	0	None	None	2	1	0	0	1	0
None	239500	notariale	3	50	0	0	None	None	2	1	0	0	1	0

[ ]:

SYNTRA PXL

www.syntrapxl.be

Simple Linear Regression

- ReRun the regression

```
x = immo_dummy[['area','number of facades','building
state','locality_Deurne','locality_Genk','locality_Mouscron','locality_Wavre']]

y = immo["price"]
import statsmodels.api as sm

#define response variable
y = y

#define predictor variables
x = x

#add constant to predictor variables
x = sm.add_constant(x)

#fit linear regression model
model = sm.OLS(y, x).fit()

#view model summary
print(model.summary())
```

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000     1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

R2 is 52,5%

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:
Date:                Sun, 08 Sep 2024   Prob (F-statistic):
Time:                        20:59:15   Log-Likelihood:                -65
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000     1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000      481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642    -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278    -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000    -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000     3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000      1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000     9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

OLS which stands for Ordinary Least Square. The model tries to find out a linear expression for the dataset which minimizes the sum of residual squares.

[Linear Regression - statsmodels 0.14.1]

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:
No. Observations:                 534   AIC:
Df Residuals:                     527   BIC:
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000     1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that the
strong multicollinearity problems or that the design matrix is singul
```

We have total 534 observation and 7 features. Out of 7 features,6 features are independent. DF Model is therefore 6. DF residual is calculated from total observation-DF model-1 which is 534-6-1 = 527 in our case.

	area	number of facades	building state	locality_Deurne	locality_Genk	locality_Mouscron	locality_Wavre
0	4	2	2	0	0	1	0
1	212	2	1	0	0	1	0
2	50	2	1	0	0	1	0
3	70	2	1	0	0	1	0
4	50	2	1	0	0	1	0

SYNTRA PXL

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                       coef    std err          t      P>|t|      [0.025
------------------------------------------------------------------------------------
const             1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area               574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state    4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne  -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk     4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron 1.965e+04   4416.044      4.449      0.000      1.1e+04    2.83e+04
locality_Wavre    1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                     1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

- Covariance type is typically **nonrobust** which means there is no elimination of data to calculate the covariance between features.
- Covariance shows how two variables move with respect to each other. If this value is greater than 0, both move in same direction and if this is less than 0, the variables mode in opposite direction.
- Covariance is difference from correlation. Covariance does not provide the strength of the relationship, only the direction of movement whereas, correlation value is normalized and ranges between -1 to +1 and correlation provides the strength of relationship.
- If we want to obtain robust covariance, we can declare cov_type=HC0/HC1/HC2/HC3.
- The usual covariance maximum likelihood estimate is very sensitive to the presence of outliers in the data set. In such a case, it would be better to use a robust estimator of covariance to guarantee that the estimation is resistant to “erroneous” observations in the data set

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000     1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000      1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

R-squared

- R-squared value is the coefficient of determination which indicates the percentage of the variability if the data explained by the selected independent variables.

Adj. R-squared

- As we add more and more independent variables to our model, the R-squared values increases but in reality, those variables do not necessarily make any contribution towards explaining the dependent variable.
- Therefore addition of each unnecessary variables needs some sort of penalty. The original R-squared values is adjusted when there are multiple variables incorporated. In essence, we should always look for adjusted R-squared value while performing multiple linear regression. For a single independent variable, both R-squared and adjusted R-squared value are same.

SYNTRA PXL

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                      coef    std err          t      P>|t|
------------------------------------------------------------------------------
const            1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area              574.8768     47.714     12.048      0.000     481.144     668.610
number of facades-1823.2323  3915.752     -0.466      0.642   -9515.632    5869.167
building state   4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk    4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron 1.965e+04  4416.044      4.449      0.000      1.1e+04    2.83e+04
locality_Wavre   1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

The coef column represents the coefficients for each independent variable along with intercept value.

Std err is the standard deviation of the corresponding variable's coefficient across all the data points.
When using only **one predicting variable**, the standard error can be obtained from this two dimensional space as shown below

$$Y = a + bx$$

$$a = \frac{[(\Sigma y)(\Sigma x^2) - (\Sigma y)(\Sigma xy)]}{[n(\Sigma x^2) - (\Sigma x)^2]}$$

$$b = \frac{[n(\Sigma xy) - (\Sigma x)(\Sigma y)]}{[n(\Sigma x^2) - (\Sigma x)^2]}$$

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04 [illegible]
locality_Genk      4.159e+04   4493.380      9.256      0.000 [illegible] [illegible]
locality_Mouscron  1.965e+04   4416.044      4.449      0.000 [illegible] [illegible]
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04 [illegible]
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

The t-column provides the t-values corresponding to to each independent variables.
T-statistics are used to calculate the p-values. Typically when p-value is less than 0.05, it indicates a **strong evidence against null hypothesis** which states that the corresponding independent variable has no effect on the dependent variable. **Or the independent variable coefficient is significantly different from zero.**

P-value of 0.642 for **Number of facades** says us that there is 64.2% chance that **Number of facades** variables has no effect on Price. It seems are got 0 p-value indicating that the data for area is statistically significant since is is less than the critical limit (0.05). In this case, we can reject the null hypothesis and say that area data is significantly controlling the Price.

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const               1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                 574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state      4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne    -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk       4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron   1.965e+04   4416.044      4.449      0.000     1.1e+04    2.83e+04
locality_Wavre      1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

F-test provides a way to check all the independent variables all together if any of those are related to the dependent variable.

- If **Prob(F-statistic)** is greater than 0.05, there is no evidence of relationship between any of the independent variable with the output.
- If it is less than 0.05, we can say that there is at least one variable which is significantly related with the output.

In our example, the p-value is less than 0.05 and therefore, one or more than one of the independent variable are related to output variable Price.

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000     1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

F-test provides a way to check all the independent variables all together if any of those are related to the dependent variable.

- If **Prob(F-statistic)** is greater than 0.05, there is no evidence of relationship between any of the independent variable with the output.
- If it is less than 0.05, we can say that there is at least one variable which is significantly related with the output.

In our example, the p-value is less than 0.05 and therefore, one or more than one of the independent variable are related to output variable Price.

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000      1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

The **log-likelihood** value is a measure for fit of the model with the given data. It is useful when we compare two or more models. The higher the value of log-likelihood, the better the model fits the given data. It can range from negative infinity to positive infinity.

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000     1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000     1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

AIC (stands for Akaike's Information Criteria developed by Japanese statistician Hirotugo Akaike) and BIC (stands for Bayesian Information Criteria) are also used as criteria for model robustness.
The goal is to minimize these values to get a better model.

$$AIC_k = n\ln(SSE) - n\ln(n) + 2(k+1)$$

$$BIC_k = n\ln(SSE) - n\ln(n) + (k+1)\ln(n)$$

Here, SSE is squared sum of error, n is number of records and k is number of variables incorporated in the model. In essence, **AIC and BIC penalize adding more variables to the model**.

When developing a model, the goal is to minimize the values of AIC and BIC whereas if we use R^2 as the metric, the goal is to increase its value.

The challenge is to find out a model that minimizes AIC/BIC or increase R^2. If the dataset is small, we can find out all possible models but this approach is not feasible if the dataset is large. It is also computationally expensive. We only need to find out those models which have the highest R^2 value or the lowest AIC/BIC.

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const               1.342e+05   7967.219     16.846      0.000    1.19e+05    1.5e+05
area                 574.8768     47.714     12.048      0.000     481.144    [illegible]
number of facades  -1823.2323   3915.752     -0.466      0.642   -9515.632  [illegible]
building state      4572.6226   4208.775      1.086      0.278  [illegible]  [illegible]
locality_Deurne    -3.471e+04   5424.176     -6.398      0.000  [illegible]  [illegible]
locality_Genk       4.159e+04   4493.380      9.256  [illegible]  [illegible]  5.04e+04
locality_Mouscron   1.965e+04   4416.044      4.449  [illegible]    1.1e+04    2.83e+04
locality_Wavre      1.077e+05   4555.817     23.637  [illegible]    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Omnibus test checks the normality of the residuals once the model is deployed. If the value is zero, it means the residuals are perfectly normal. Here, in the example prob(Omnibus) is 0 indicating that there is 0% chance that the residuals the normally distributed.
For a model to be robust, besides checking R-squared and other rubrics, the residual distribution is also required to be normal ideally. In other words, the residual should not follow any pattern when plotted against the fitted values.

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144      668.6
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632       58
building state     4572.6226   4208.775      1.086      0.278   -3695.413
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000    -4.54e
locality_Genk      4.159e+04   4493.380      9.256      0.000    3           4e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000                2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.0         7e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Wat                       1.782
Prob(Omnibus):                  0.000   Jarque    a (JB):              607.628
Skew:                           0.594   Pr   (JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Skew values tells us the **skewness** of the residual distribution. Normally distributed variables have 0 skew values.
Kurtosis is a measure of light-tailed or heavy-tailed distribution compared to normal distribution. High kurtosis indicates the distribution is too narrow and low kurtosis indicates the distribution is too flat. A kurtosis value between -2 and +2 is good to prove normalcy.

www.syntrapxl.be

SYNTRA PXL

Simple Linear Regression

- ReRun the regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000     1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+0
locality_Mouscron  1.965e+04   4416.044      4.449      0.000     1.1e+04    2.83e
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.1   05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Durbin-Watson statistic provides a measure of autocorrelation in the residual. If the residual values are autocorrelated, the model becomes biased and it is not expected. This simply means that one value should not be depending on any of the previous values. An ideal value for this test ranges from 0 to 4.

Jarque-Bera (JB) and Prob(JB) is similar to Omni test measuring the normalcy of the residuals.

High condition number indicates that there are possible multicollinearity present in the dataset. If only one variable is used as predictor, this value is low and can be ignored. We can proceed like stepwise regression and see if there is any multicollinearity added when additional variables are included.

Simple Linear Regression

How to improve the model ?

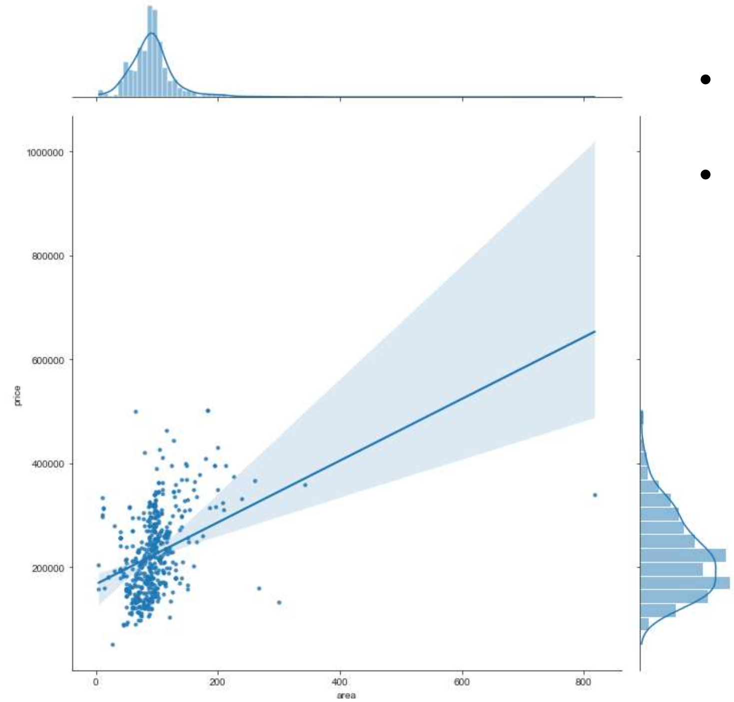
- Remove outliers. Some observations are not relevant for our business
- Remove features (number of facades, building state…)
- Variable transformation

Simple Linear Regression

How to improve the model ?

- Make a jointplot to illustrate the relation between area and price

```
g = sns.jointplot("area", "price", data=immo, kind="reg", scatter_kws={"s": 10}, size=10)
```



- Remove observations with price>400.000 and area>200
- Rerun the jointplot

Simple Linear Regression

```
x = immo_dummy[['area','locality_Deurne','locality_Genk','locality_Mouscron','locality_Wavre']]
y = immo_dummy["price"]
import statsmodels.api as sm

#define response variable
y = y

#define predictor variables
x = x

#add constant to predictor variables
x = sm.add_constant(x)

#fit linear regression model
model = sm.OLS(y, x).fit()

#view model summary
print(model.summary())
```

Simple Linear Regression

When developing a model, the goal is to minimize the values of AIC and BIC whereas if we use R^2 as the metric, the goal is to increase its value.

The challenge is to find out a model that minimizes AIC/BIC or increase R^2.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.525
Model:                            OLS   Adj. R-squared:                  0.520
Method:                 Least Squares   F-statistic:                     97.23
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           5.04e-82
Time:                        20:59:15   Log-Likelihood:                -6559.2
No. Observations:                 534   AIC:                         1.313e+04
Df Residuals:                     527   BIC:                         1.316e+04
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.342e+05   7967.219     16.846      0.000    1.19e+05     1.5e+05
area                574.8768     47.714     12.048      0.000     481.144     668.610
number of facades -1823.2323   3915.752     -0.466      0.642   -9515.632    5869.167
building state     4572.6226   4208.775      1.086      0.278   -3695.413    1.28e+04
locality_Deurne   -3.471e+04   5424.176     -6.398      0.000   -4.54e+04    -2.4e+04
locality_Genk      4.159e+04   4493.380      9.256      0.000    3.28e+04    5.04e+04
locality_Mouscron  1.965e+04   4416.044      4.449      0.000      1.1e+04    2.83e+04
locality_Wavre     1.077e+05   4555.817     23.637      0.000    9.87e+04    1.17e+05
==============================================================================
Omnibus:                       95.240   Durbin-Watson:                   1.782
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              607.628
Skew:                           0.594   Prob(JB):                    1.14e-132
Kurtosis:                       8.089   Cond. No.                     1.09e+18
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.96e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.615
Model:                            OLS   Adj. R-squared:                  0.611
Method:                 Least Squares   F-statistic:                     202.5
Date:                Sun, 08 Sep 2024   Prob (F-statistic):          1.09e-103
Time:                        23:09:58   Log-Likelihood:                -6190.0
No. Observations:                 513   AIC:                         1.239e+04
Df Residuals:                     508   BIC:                         1.241e+04
Df Model:                           4
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.122e+05   4801.840     23.361      0.000    1.03e+05    1.22e+05
area                887.7203     64.959     13.666      0.000     760.098    1015.342
locality_Deurne   -3.703e+04   3528.536    -10.494      0.000    -4.4e+04   -3.01e+04
locality_Genk      3.758e+04   3390.807     11.084      0.000    3.09e+04    4.42e+04
locality_Mouscron  1.315e+04   3302.230      3.983      0.000    6663.829    1.96e+04
locality_Wavre     9.847e+04   3805.262     25.878      0.000      9.1e+04    1.06e+05
==============================================================================
Omnibus:                       14.255   Durbin-Watson:                   1.765
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               15.427
Skew:                           0.351   Prob(JB):                     0.000447
Kurtosis:                       3.479   Cond. No.                     9.62e+17
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.78e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Simple Linear Regression

How to improve the model ?

Exponents and logarithms

- A logarithm is a mathematical function used to determine the power to which a number, also known as the base, must be raised to obtain another number. Expressed mathematically:

- If b^y = x, then log_b(x) = y.

- The “b” is the base, “x” is the number we are trying to find the logarithm of, and “y” is the result of the operation.

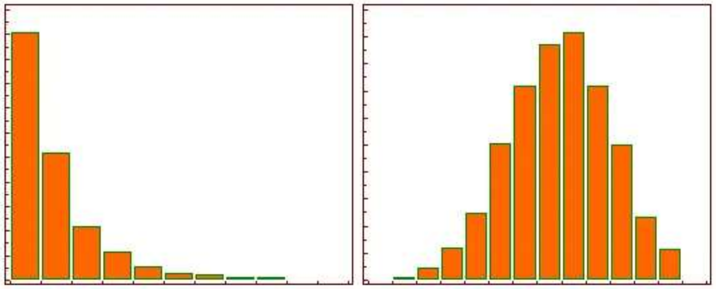
- To understand the practical application, consider the example 2^3 = 8. The logarithm base 2 of 8 is 3, which can be written as log2(8) = 3.

Simple Linear Regression

How to improve the model ?

Why logarithms

- Quite often data arising in real studies are so skewed that standard statistical analyses of these data yield invalid results.
- Many methods have been developed to test the normality assumption of observed data.
- When the distribution of the continuous data is non-normal, transformations of data are applied to make the data as "normal" as possible and, thus, **increase the validity** of the associated statistical analyses.
- popular use of the log transformation is to reduce the **variability of data**, especially in data sets that include outlying observations



Simple Linear Regression

How to improve the model ?

Rules for interpretation

•**Only the dependent/response variable is log-transformed**. Exponentiate the coefficient. This gives the multiplicative factor for every one-unit increase in the independent variable. Example: the coefficient is 0.198. exp(0.198) = 1.218962. For every one-unit increase in the independent variable, our dependent variable increases by a factor of about 1.22, or 22%. Recall that multiplying a number by 1.22 is the same as increasing the number by 22%. Likewise, multiplying a number by, say 0.84, is the same as decreasing the number by 1 – 0.84 = 0.16, or 16%.

•**Only independent/predictor variable(s) is log-transformed**. Divide the coefficient by 100. This tells us that a 1% increase in the independent variable increases (or decreases) the dependent variable by (coefficient/100) units. Example: the coefficient is 0.198. 0.198/100 = 0.00198. For every 1% increase in the independent variable, our dependent variable increases by about 0.002. For x percent increase, multiply the coefficient by log(1.x). Example: For every 10% increase in the independent variable, our dependent variable increases by about 0.198 * log(1.10) = 0.02.

•**Both dependent/response variable and independent/predictor variable(s) are log-transformed**. Interpret the coefficient as the percent increase in the dependent variable for every 1% increase in the independent variable. Example: the coefficient is 0.198. For every 1% increase in the independent variable, our dependent variable increases by about 0.20%. For x percent increase, calculate 1.x to the power of the coefficient, subtract 1, and multiply by 100. Example: For every 20% increase in the independent variable, our dependent variable increases by about ($1.20^{0.198}$ - 1) * 100 = 3.7 percent.

Simple Linear Regression

- Let us do some transformation

  - Calculate price per sqm (price / area)
  - Take the log of area and price / area

```
immo_dummy['log_price_sqm']= np.log(immo_dummy['price']/immo_dummy['area'])
immo_dummy['log_area']= np.log(immo_dummy['area'])
```

Simple Linear Regression

- Rerun the regression

```
x = immo_dummy[['log_area','locality_Deurne','locality_Genk','locality_Mouscron','locality_Wavre']]
y = immo_dummy["log_price_sqm"]
import statsmodels.api as sm

#define response variable
y = y

#define predictor variables
x = x

#add constant to predictor variables
x = sm.add_constant(x)

#fit linear regression model
model = sm.OLS(y, x).fit()

#view model summary
print(model.summary())
```

Simple Linear Regression

before

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.615
Model:                            OLS   Adj. R-squared:                  0.611
Method:                 Least Squares   F-statistic:                     202.5
Date:                Sun, 08 Sep 2024   Prob (F-statistic):          1.09e-103
Time:                        23:09:58   Log-Likelihood:                -6190.0
No. Observations:                 513   AIC:                         1.239e+04
Df Residuals:                     508   BIC:                         1.241e+04
Df Model:                           4
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1.122e+05   4801.840     23.361      0.000    1.03e+05    1.22e+05
area                887.7203     64.959     13.666      0.000     760.098    1015.342
locality_Deurne   -3.703e+04   3528.536    -10.494      0.000    -4.4e+04   -3.01e+04
locality_Genk      3.758e+04   3390.807     11.084      0.000    3.09e+04    4.42e+04
locality_Mouscron  1.315e+04   3302.230      3.983      0.000    6663.829    1.96e+04
locality_Wavre     9.847e+04   3805.262     25.878      0.000      9.1e+04    1.06e+05
==============================================================================
Omnibus:                       14.255   Durbin-Watson:                   1.765
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               15.427
Skew:                           0.351   Prob(JB):                     0.000447
Kurtosis:                       3.479   Cond. No.                     9.62e+17
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.78e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

after

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          log_price_sqm   R-squared:                       0.781
Model:                            OLS   Adj. R-squared:                  0.780
Method:                 Least Squares   F-statistic:                     453.7
Date:                Sun, 08 Sep 2024   Prob (F-statistic):          4.11e-166
Time:                        23:26:15   Log-Likelihood:                 56.616
No. Observations:                 513   AIC:                            -103.2
Df Residuals:                     508   BIC:                            -82.03
Df Model:                           4
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 9.1444      0.079    116.176      0.000       8.990       9.299
log_area             -0.8164      0.022    -36.685      0.000      -0.860      -0.773
locality_Deurne       1.9391      0.026     75.263      0.000       1.889       1.990
locality_Genk         2.3645      0.027     88.276      0.000       2.312       2.417
locality_Mouscron     2.2324      0.024     91.546      0.000       2.184       2.280
locality_Wavre        2.6084      0.027     96.666      0.000       2.555       2.661
==============================================================================
Omnibus:                        5.834   Durbin-Watson:                   1.757
Prob(Omnibus):                  0.054   Jarque-Bera (JB):                8.007
Skew:                           0.028   Prob(JB):                       0.0183
Kurtosis:                       3.610   Cond. No.                     3.83e+16
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 7.29e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Simple Linear Regression

- Implement the model

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          log_price_sqm   R-squared:                       0.781
Model:                            OLS   Adj. R-squared:                  0.780
Method:                 Least Squares   F-statistic:                     453.7
Date:                Sun, 08 Sep 2024   Prob (F-statistic):          4.11e-166
Time:                        23:26:15   Log-Likelihood:                 56.616
No. Observations:                 513   AIC:                            -103.2
Df Residuals:                     508   BIC:                            -82.03
Df Model:                           4
Covariance Type:            nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
const                  9.1444      0.079    116.176      0.000       8.990       9.299
log_area              -0.8164      0.022    -36.685      0.000      -0.860      -0.773
locality_Deurne        1.9391      0.026     75.263      0.000       1.889       1.990
locality_Genk          2.3645      0.027     88.276      0.000       2.312       2.417
locality_Mouscron      2.2324      0.024     91.546      0.000       2.184       2.280
locality_Wavre         2.6084      0.027     96.666      0.000       2.555       2.661
==============================================================================
Omnibus:                        5.834   Durbin-Watson:                   1.757
Prob(Omnibus):                  0.054   Jarque-Bera (JB):                8.007
Skew:                           0.028   Prob(JB):                       0.0183
Kurtosis:                       3.610   Cond. No.                     3.83e+16
==============================================================================
```

Paste Copy Format Painter Clipboard | Font | Merge & Center Alignment | Number | Formatting Table Styles | Cells

OGEST | =EXP($O$1+$O$2*LN(G8)+VLOOKUP(A8;$R$1:$S$4;2;FALSE))

N	O
const	9,14444
log_liv_area	-0,8164

R	S
Deurne	1,9391
Genk	2,3645
Mouscron	2,2324
Wavre	2,6084

locality	type of pr	subtype o	price	sale t	number o	area	furnished	open fire	terrace	terrace ar	number o	building state	Actual Price SQM	Predixted Price SQM	Diff%
Mouscron	Appartem	None	204584	notar	2	4	0	0	TRUE	40	2	2	51146	=EXP($O$1+$O$2*LN(G8)+VLOOKUP(A8;$	
Mouscron	Appartem	None	395000	notar	6	212	0	0	None	None	2	1	1863,207547	1100,723495	69,27%
Mouscron	Appartem	None	182500	notar	2	50	0	0	None	None	2	1	3650	3579,807899	1,96%
Mouscron	Appartem	None	229500	notar	2	70	0	0	None	None	2	1	3278,571429	2719,949205	20,54%
Mouscron	Appartem	None	239500	notar	3	50	0	0	None	None	2	1	4790	3579,807899	33,81%
Mouscron	Appartem	None	189500	notar	2	45	0	0	None	None	2	1	4211,111111	3901,360991	7,94%
Mouscron	Appartem	None	259900	notar	2	104	0	0	None	None	2	1	2499,038462	1968,760173	26,93%
Mouscron	Appartem	None	196500	notar	2	40	0	0	None	None	2	1	4912,5	4295,137353	14,37%
Mouscron	Appartem	None	157500	notar	1	50	0	0	None	None	1	1	3150	3579,807899	-12,01%
Mouscron	Appartem	None	157500	notar	1	50	0	0	None	None	1	1	3150	3579,807899	-12,01%
Mouscron	Appartem	None	139500	notar	1	70	0	0	None	None	1	1	1992,857143	2719,949205	-26,73%
Mouscron	Appartem	None	157500	notar	1	50	0	0	None	None	1	1	3150	3579,807899	-12,01%

Simple Linear Regression

- Perform a cross validation

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from numpy import mean
from numpy import absolute
from numpy import sqrt
import pandas as pd

#define predictor and response variables
X = immo_dummy[['log_area','locality_Deurne','locality_Genk','locality_Mouscron','locality_Wavre']]
y = immo_dummy["log_price_sqm"]

#define cross-validation method to use
cv = KFold(n_splits=3, random_state=1, shuffle=True)

#build multiple linear regression model
model = LinearRegression()

#use k-fold CV to evaluate model
scores = cross_val_score(model, X, y, scoring='r2',
                    cv=cv, n_jobs=None)

#view mean absolute error
print(scores)
mean(absolute(scores))
```

Simple Linear Regression

- Perform a cross validation

```
[0.68627212 0.74201595 0.85051939]
]: 0.7596024856593727
```

Simple Linear Regression

- Make a salary estimator