# Analyzing Data with Python

## Importing Databases

## Data Wrangling

## Exploratory Data Analysis

## 4 Model Development

After we checked correlation between predictor parameters and target parameter we can start making model.

We tried different regression models. Single parameter linear regression, single parameter polynomial regression, multiple parameter linear regression. For some reason multiple parameter polynomial regression was not covered in the lab.

### Model validation using visualization

The most obvious one is to plot scatter plot of actual values and the fitted curve.

Also, you can plot on x-axis the predictor, and on y-axis the error of prediction ŷ -y. The plot is called Residual Plot.



Ideally the error points should be evenly distributed from the x-axis. Residual plot works only with one predictor (x-axis).

width = 12

height = 10

plt.figure(figsize=(width, height))

sns.residplot(df['highway-mpg'], df['price'])

plt.show()

Distribution plot can help visualize prediction accuracy for Multiple Linear Regression.

plt.figure(figsize=(width, height))

ax1 = sns.distplot(df['price'], hist=False, color="r", label="Actual Value")

sns.distplot(Y\_hat, hist=False, color="b", label="Fitted Values" , ax=ax1)

plt.title('Actual vs Fitted Values for Price')

plt.xlabel('Price (in dollars)')

plt.ylabel('Proportion of Cars')

plt.show()

plt.close()

Chart

Description automatically generated

### Model validation using statistical indicators

The way to validate the accuracy of prediction is to look at specific statistical coefficients:

R2 – R squared, coefficient of determination. Show how close the real data is to the fitted curve. Percentage of the points which position is explained by the polynomial fit model (e.g. 0.8 is 80%).

Mean Squared Error (MSE) – the average distance between actual value (y) and predicted value (ŷ). Basically (ŷ -y)2