* EDA (Exploratory Data Analysis)
  + Summarize main characteristic
  + Understanding of the data
  + Relationships
  + Extract important variables
* Descriptive Statistics
  + Describe basic features of the data
    - Using pandas – **describe()** method
      * Ex: **df.describe()**
    - Categorical data summary
      * Using **Value\_Counts()**
        + ex: **drive\_wheels\_counts=df[“drive-wheels”].value\_counts().to\_frame()**
      * Box plots
        + **Chart, box and whisker chart

          Description automatically generatedex: sns.boxplot(x= “drive-wheels”, y=”price”, data=df)**
      * Scatter Plot
      * Can use **.set(title='Title of Plot')**
        + Predictor on the x-axis
        + Target on the y-axis
        + Ex:

**Y=df[“price]**

**X=df[“engine-size”]**

**Plt.scatter(x,y)**

**Plt.title(“Scatterplot of engine size vs price”)**

**Plt.xlabel(“Engine Size”)**

**Plt.ylabel(“Price”)**

* Grouping Data
  + In panda **dataframe.Groupby()**
    - Can be applied on categorical variables.
    - Groups data into categories (Use **.unique()** )
    - Ex:
      * **df\_test = df[[‘drive-wheels’ , ’body-style’ , ’price’]]**
      * **df\_grp = df\_test.groupby([‘drive-wheels’ , ’body-style’], as\_index = False).mean()**
  + Pivot table in python (pandas)
    - Ex:
      * **df\_pivot = df\_grp.pivot (index = ‘drive\_wheels’, columns = ‘body-style’ )**
  + Returning specified values with **.get\_group**
    - Ex: **grouped\_test2.get\_group('4wd')['price']**
  + Heat map plot
    - Chart

      Description automatically generatedEx:
* Correlation (.corr() method)
  + Ex: **df[['peak-rpm','highway-mpg','price']].corr()**
  + Chart, scatter chart

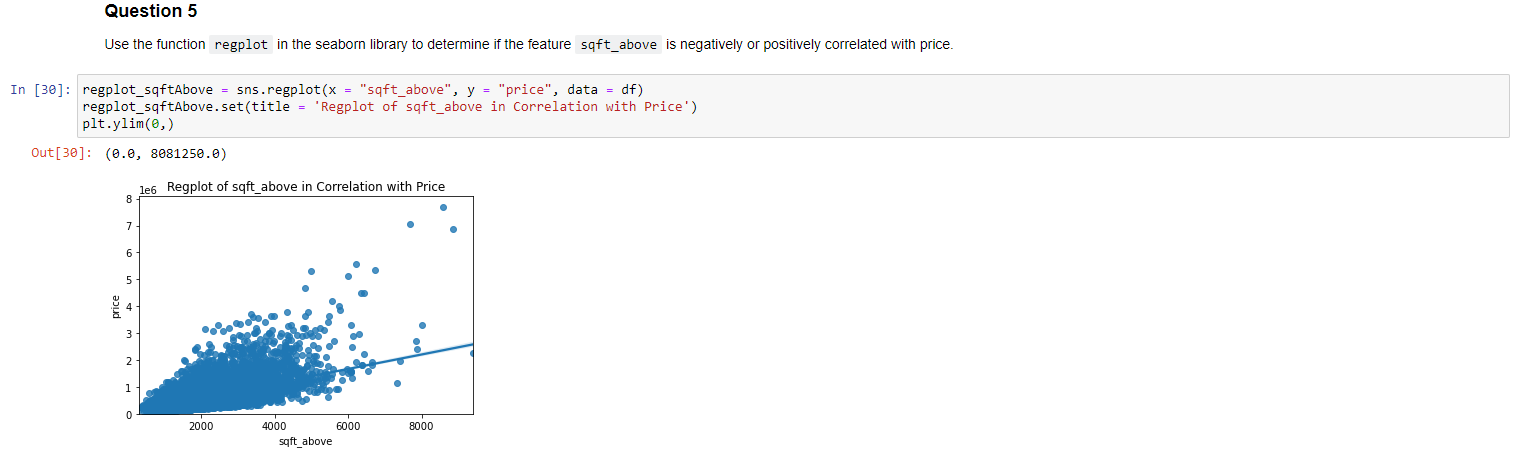
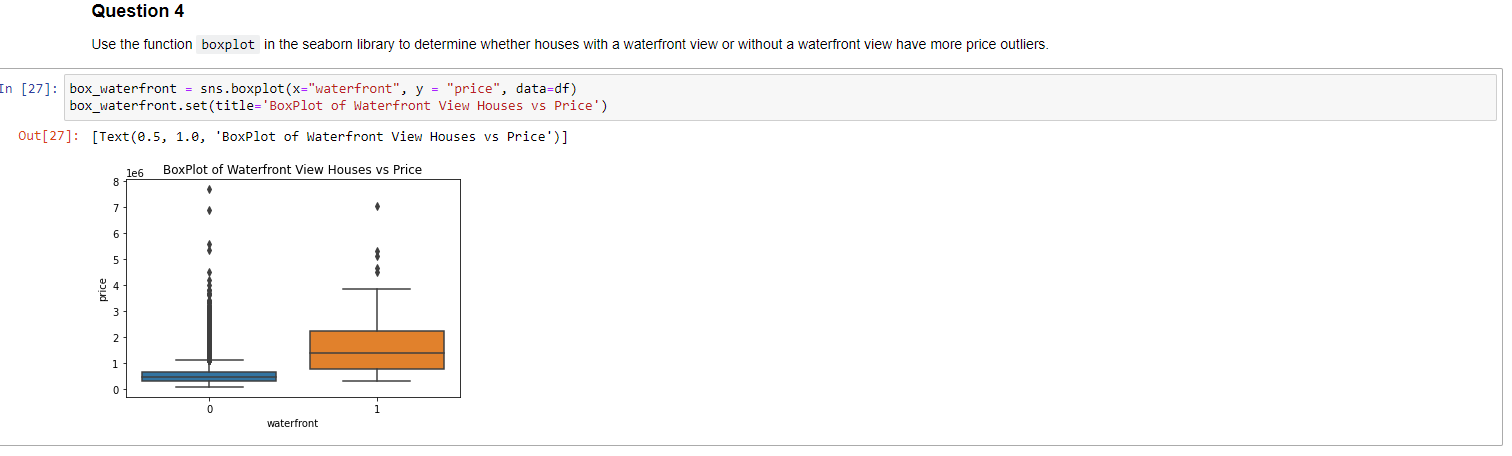
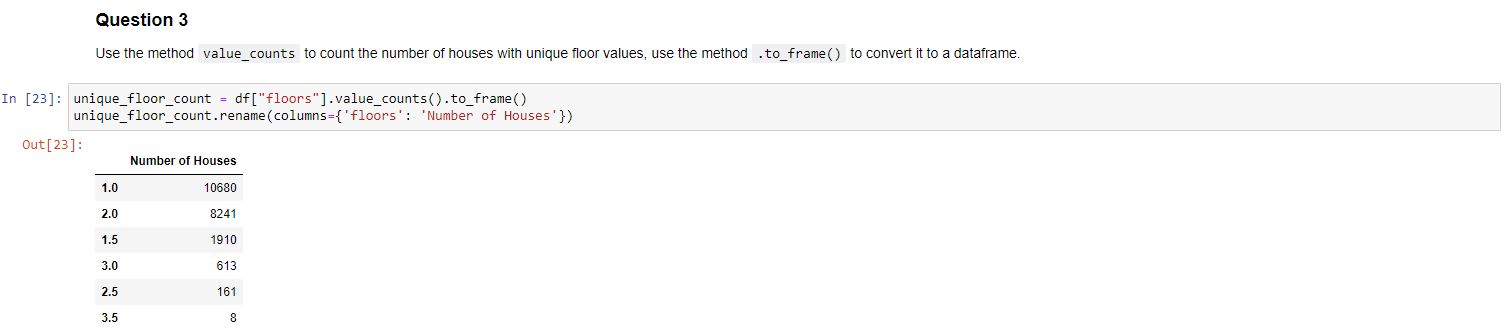
    Description automatically generatedMeasure to what extent different variables are interdependent. (Using Seaborn as sns)
  + Statistical methods
    - Correlation Coefficient
      * Close to +1: Large Positive Relationship
      * Close to -1: Large Negative Relationships
      * Close to 0: No Relationship
    - P-Value
      * P-Value < 0.001 Strong
      * P-Value < 0.05 Moderate
      * P-Value < 0.1 Weak
      * P-Value > 0.1 No
    - Using scipy package
      * Ex:
        + Chart

          Description automatically generated**pearson\_coef, p-value = stats.pearsonr(df[‘horsepower’], df[‘price’]**
  + Chi-Square Test for Association (denoted as x^2)
    - Test a null hypothesis that the variables are independent.
    - A picture containing text

      Description automatically generatedTable

      Description automatically generatedA picture containing diagram

      Description automatically generatedThe test doesn’t tell you what type of relationship that exist but rather that a relationship exists.
    - Ex:
      * **scipy.stats.chi2\_contingency(con\_table, correction = True)**

****