@author: Carlos Andrés Tejada Ramírez
Point 4 of Homework 1 from statistics III

# Data Exploration AND PCA (Point 4)

## Use this data from cars to do the following tasks

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import shapiro

df = pd.read_csv("../Docs/Data/CARS.csv")
df["Invoice"] = df["Invoice"].replace('[\\$,]', '',
regex=True).astype(float)
```

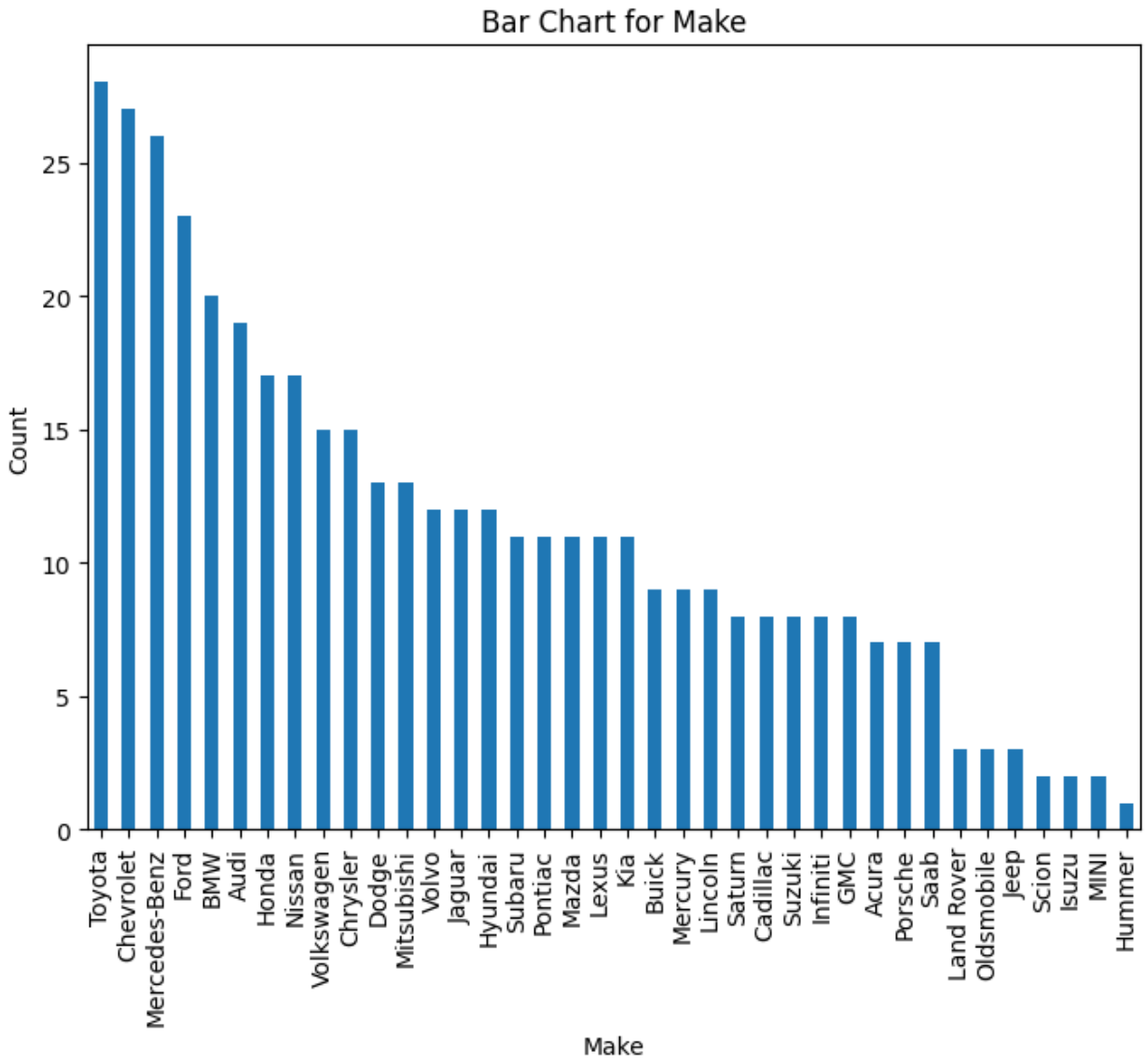
'Model' and 'MSRP' variables are excluded because they have too many different values or unique values

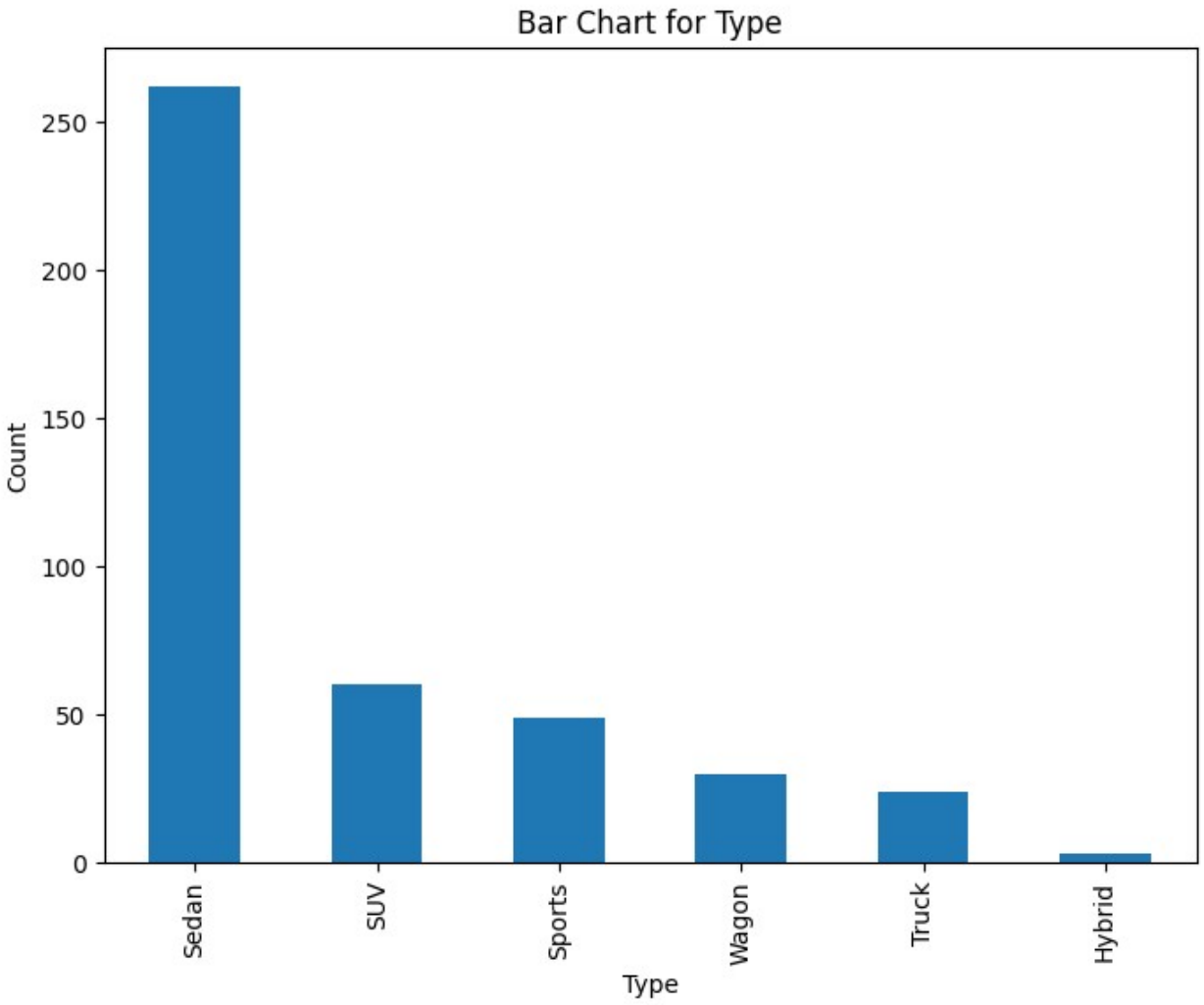
## 1. Distribution of each variable

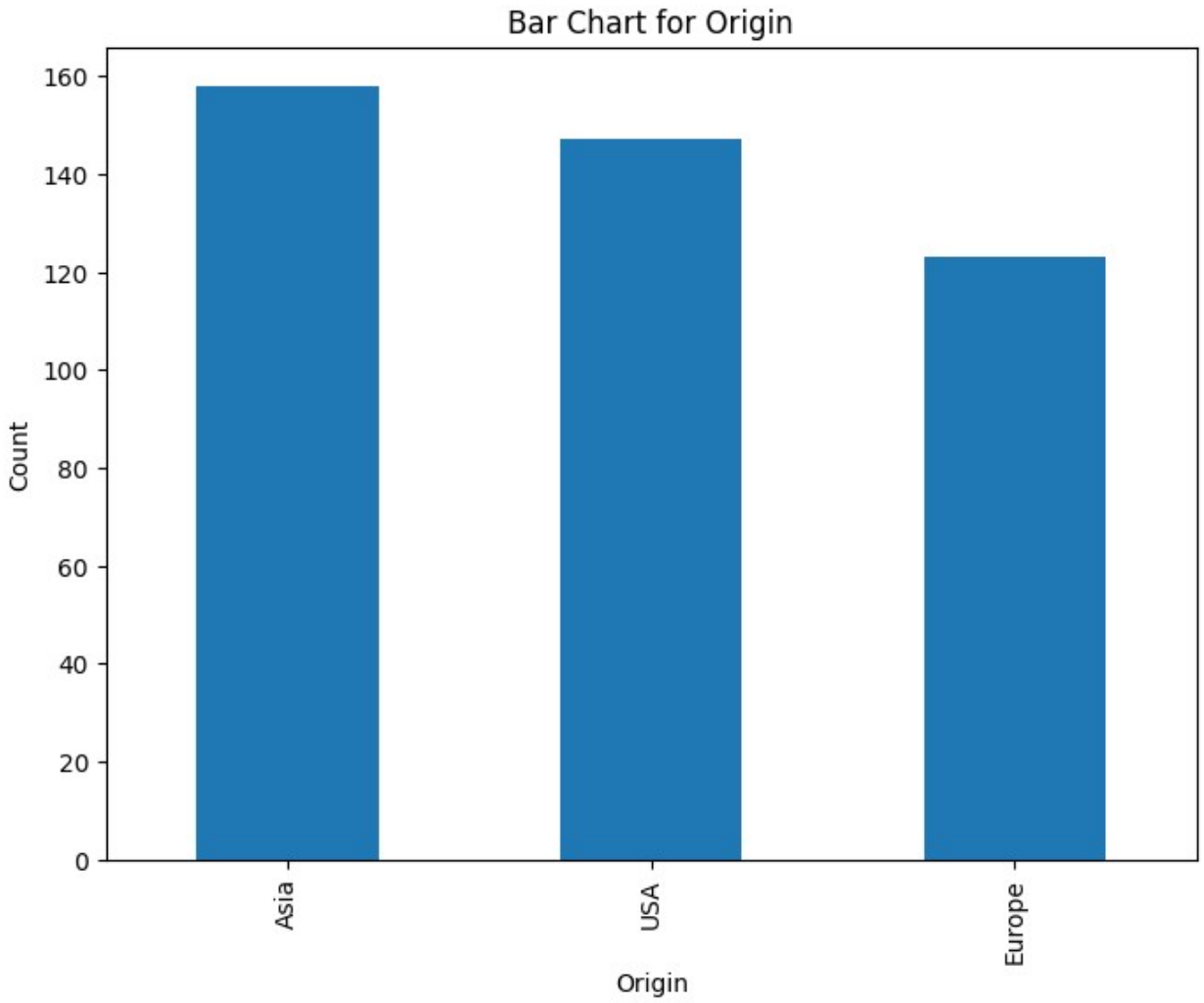
```
categoricalVariables =
df.select_dtypes(include='object').drop(columns=['Model', 'MSRP'])
numericVariables = df.select_dtypes(include=['int64', 'float64'])
```

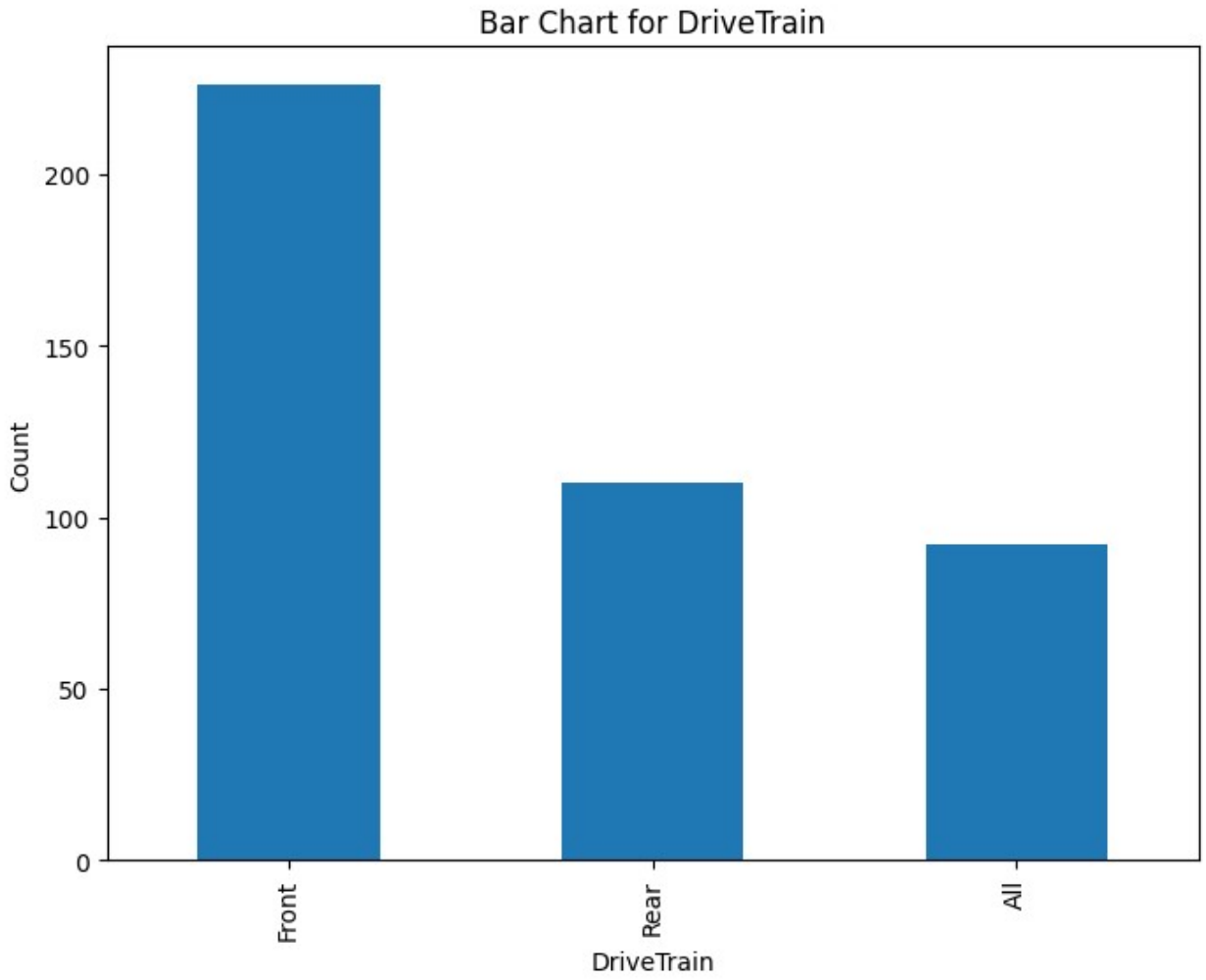
# 1.1. For categorical variables a bar graph. Category number of observations.

```
def graphBarChart(column):
    plt.figure(figsize=(8, 6))
    df[column].value_counts().plot(kind='bar')
    plt.title(f'Bar Chart for {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.show()

for column in categoricalVariables.columns:
    graphBarChart(column)
```


Bar Chart for Make
Count
0
5
10
15
20
25
Toyota
Chevrolet
Mercedes-Benz
Ford
BMW
Audi
Honda
Nissan
Volkswagen
Chrysler
Dodge
Mitsubishi
Volvo
Jaguar
Hyundai
Subaru
Pontiac
Mazda
Lexus
Kia
Buick
Mercury
Lincoln
Saturn
Cadillac
Suzuki
Infiniti
GMC
Acura
Porsche
Saab
Land Rover
Oldsmobile
Jeep
Scion
Isuzu
MINI
Hummer
Make


Bar Chart for Type
Count
250
200
150
100
50
0
Sedan
SUV
Sports
Wagon
Truck
Hybrid
Type

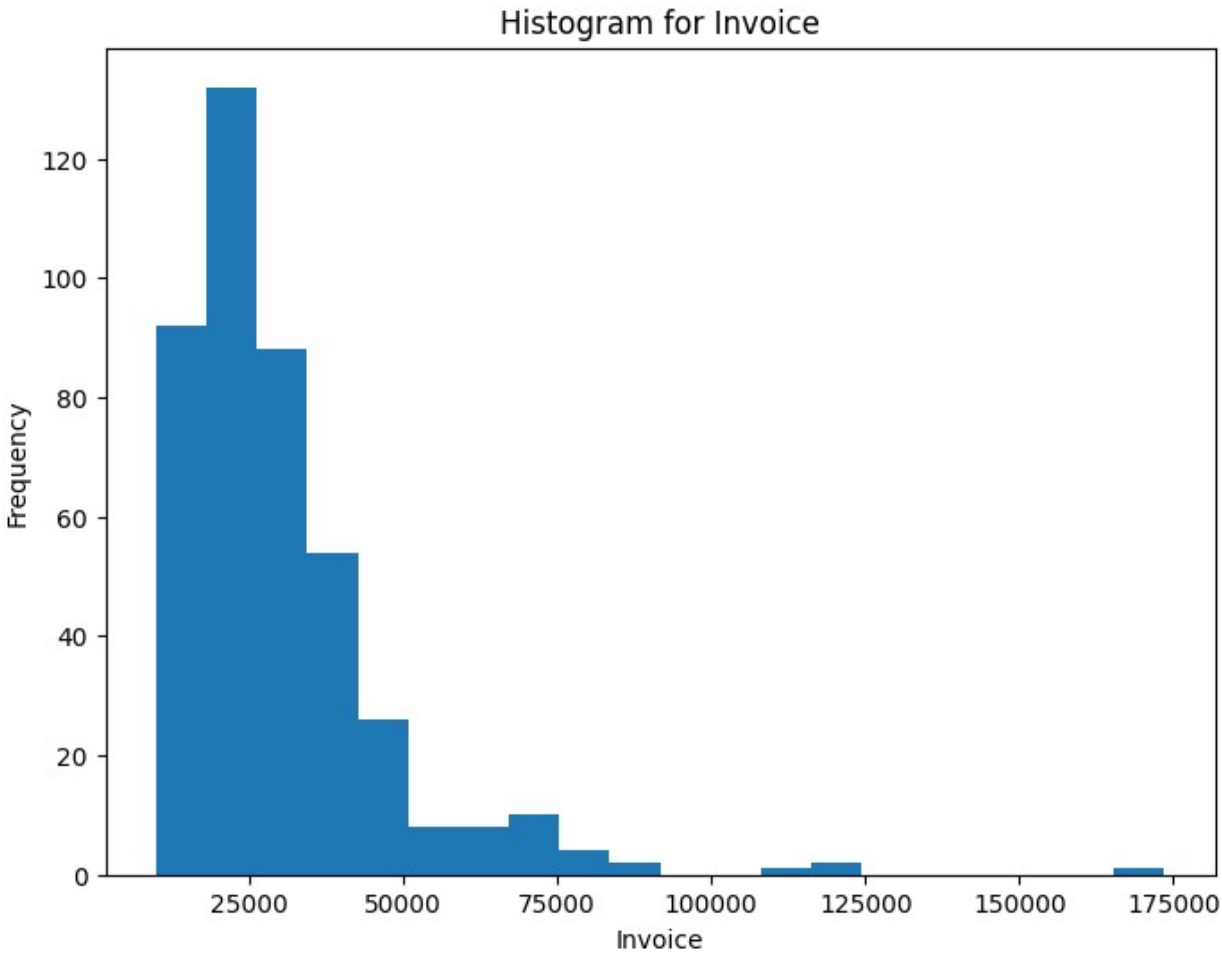

Bar Chart for Origin
160
140
120
100
80
60
40
20
0
Count
Asia
USA
Europe
Origin

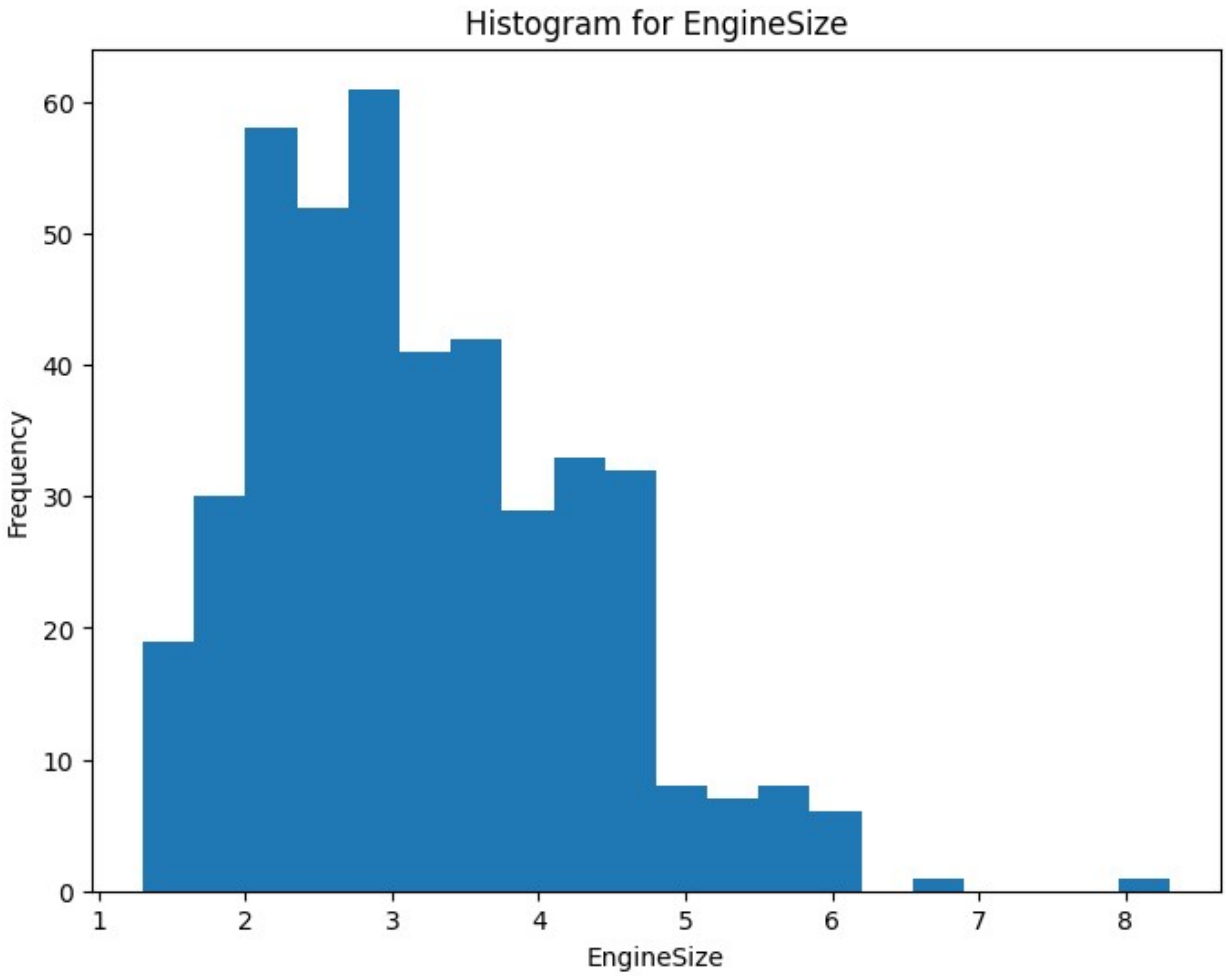


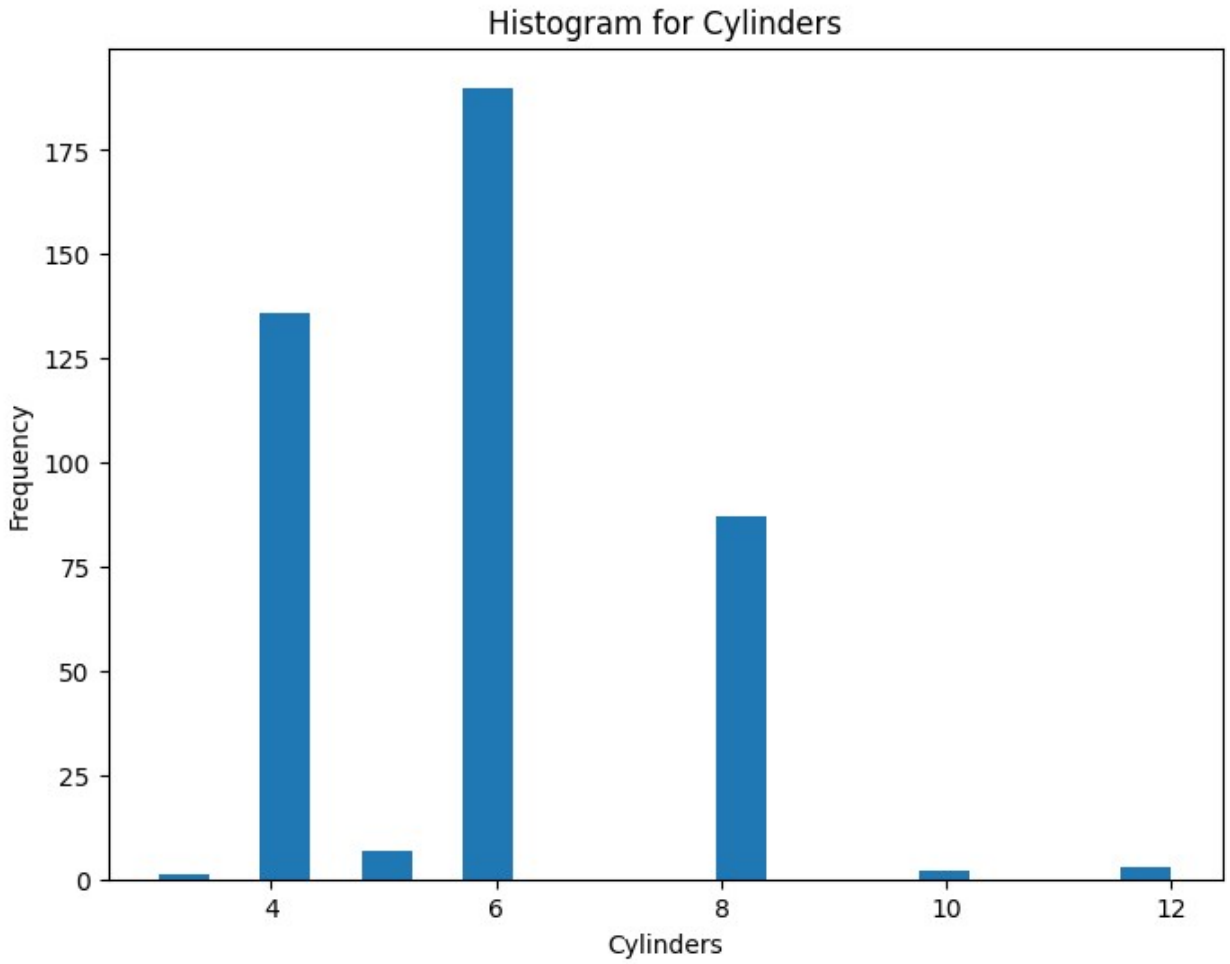

For numerical variables create histograms. List the car models that are further away from 5 deviation standards, and would be considered outliers. Test if it is a distribution normal or not.

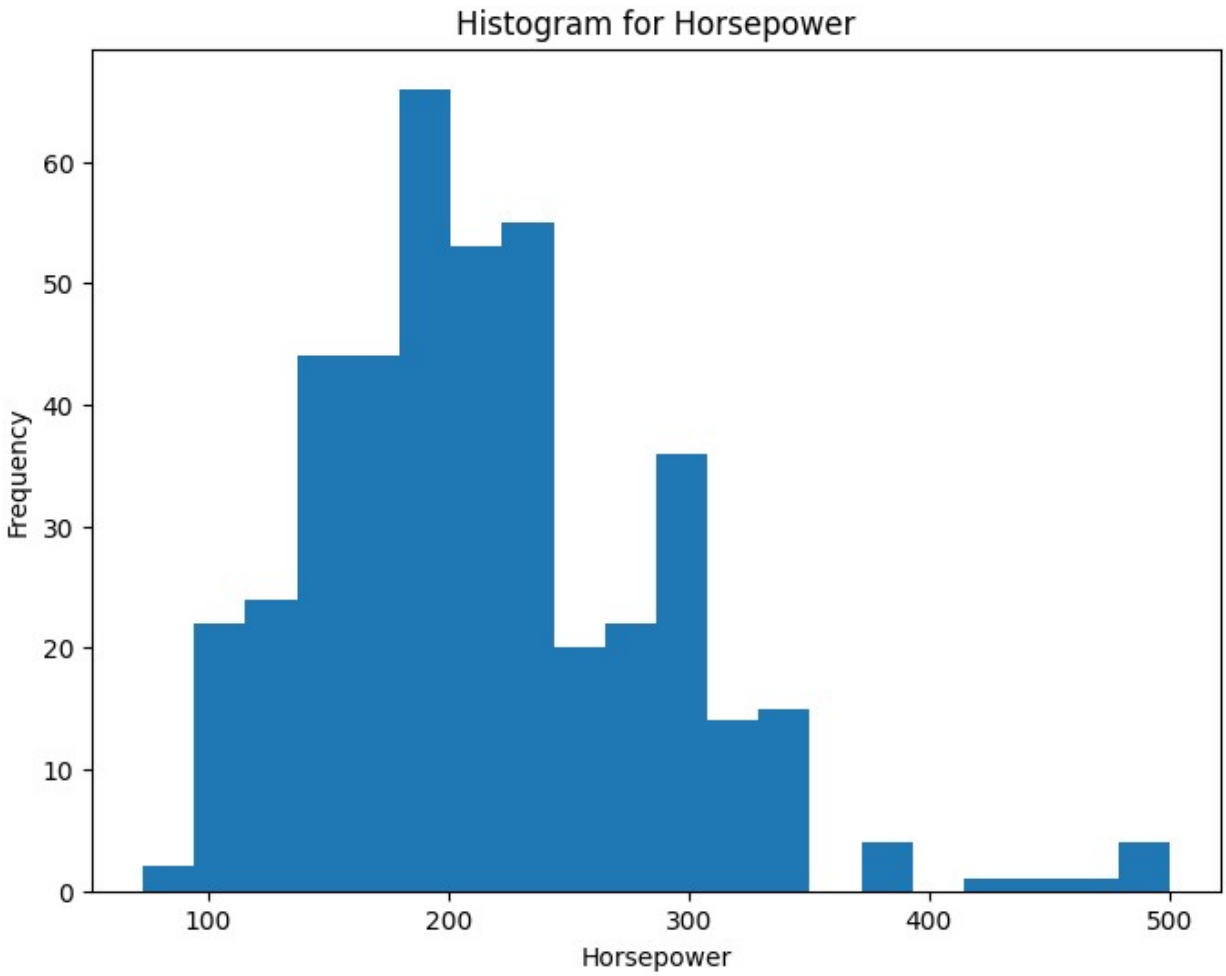
Histograms

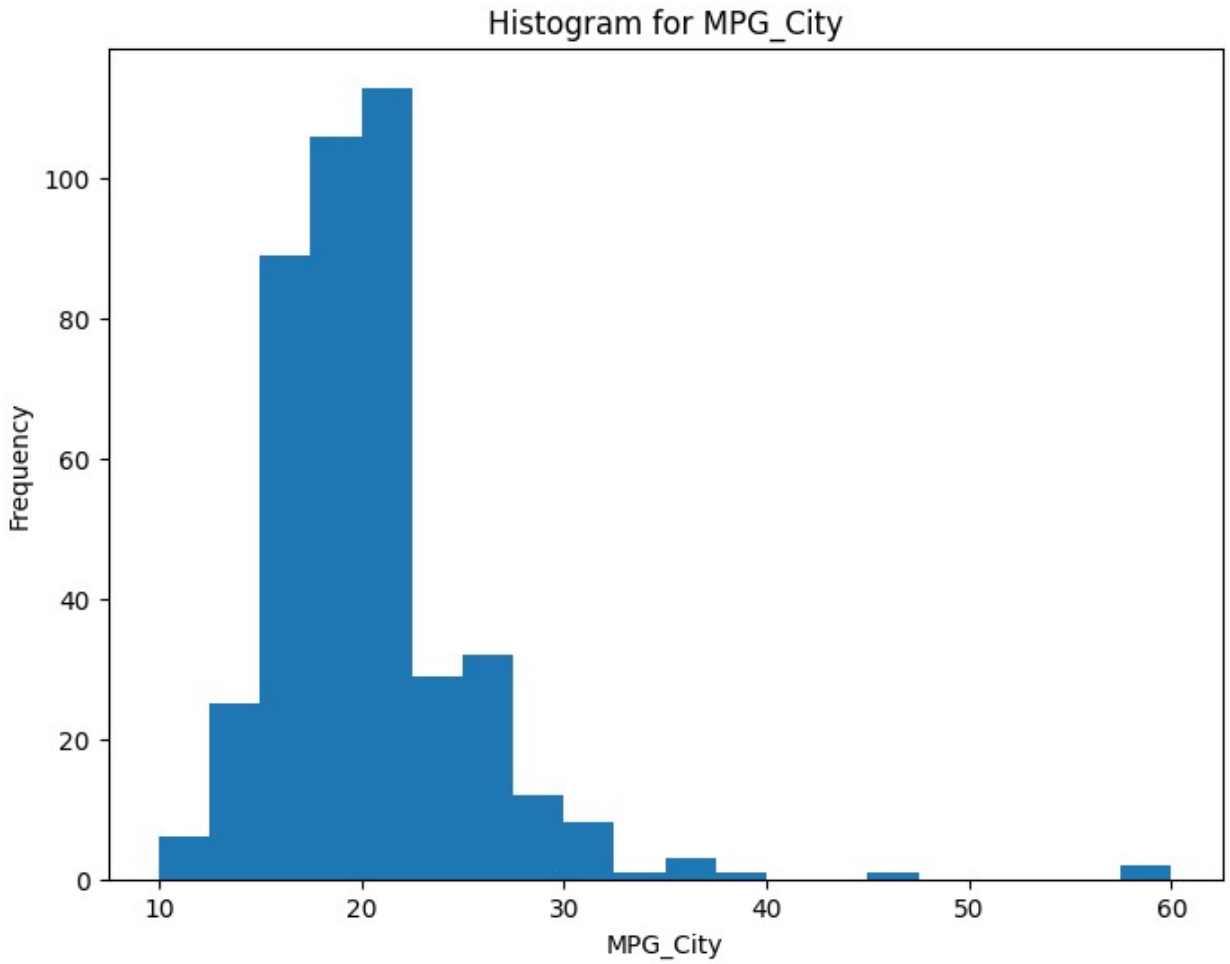
```
def graphHistogram(numericVariables):
    for column in numericVariables.columns:
        plt.figure(figsize=(8, 6))
        plt.hist(df[column], bins=20)
        plt.title(f'Histogram for {column}')
        plt.xlabel(column)
        plt.ylabel('Frequency')
        plt.show()

graphHistogram(numericVariables)
```

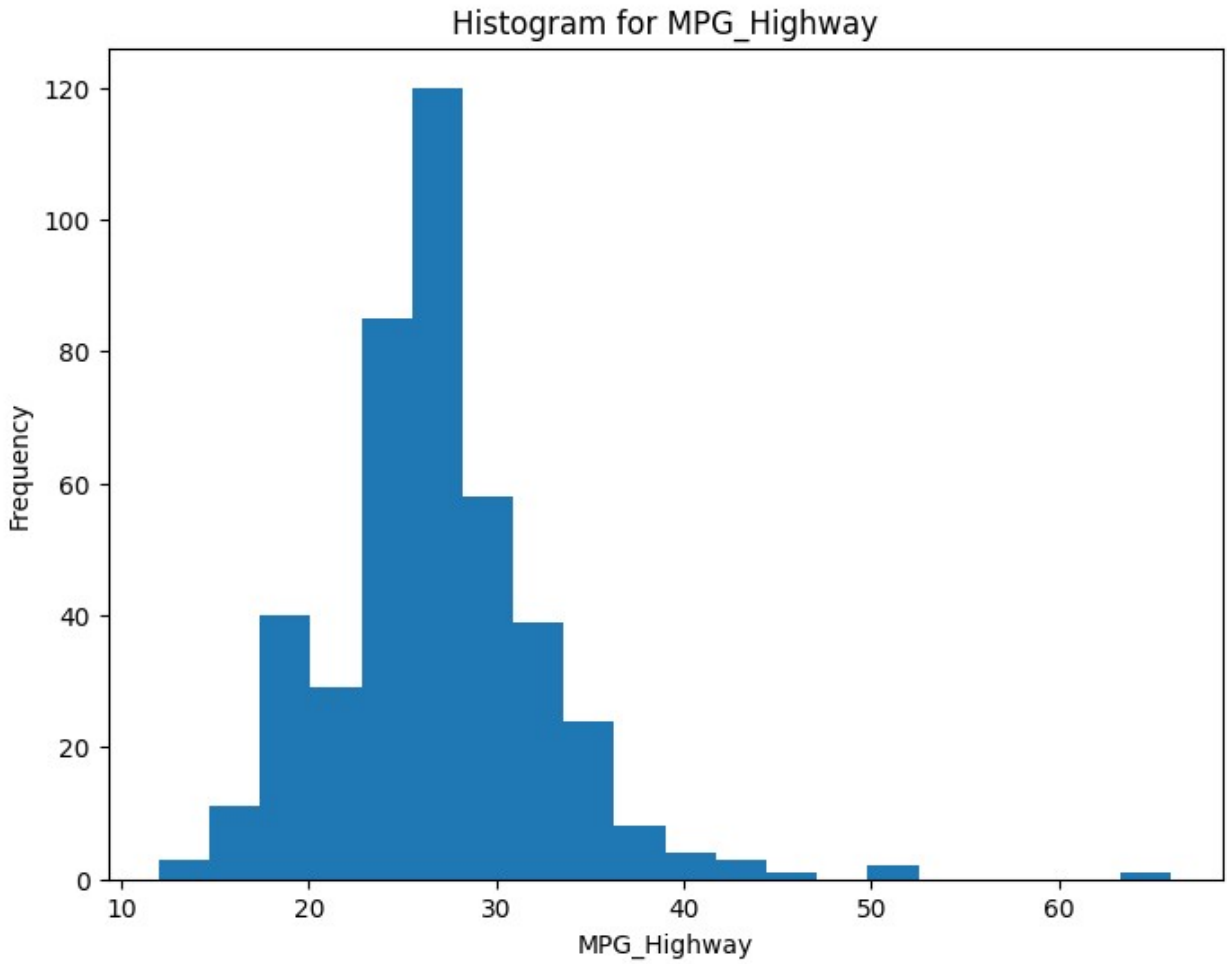

Histogram for Invoice
Frequency
0
20
40
60
80
100
120
25000
50000
75000
100000
125000
150000
175000
Invoice

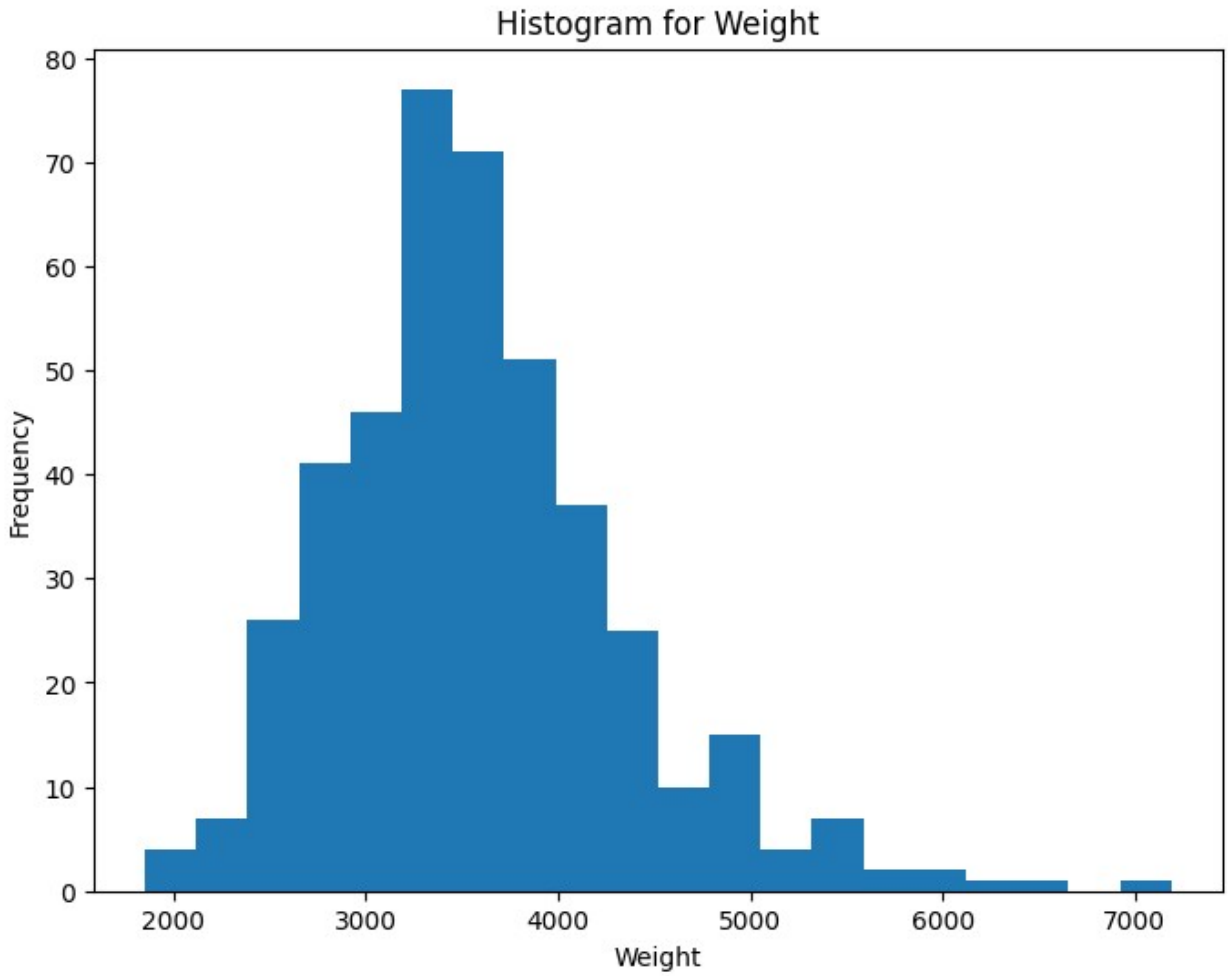

Histogram for EngineSize
Frequency
EngineSize
0
10
20
30
40
50
60
1
2
3
4
5
6
7
8

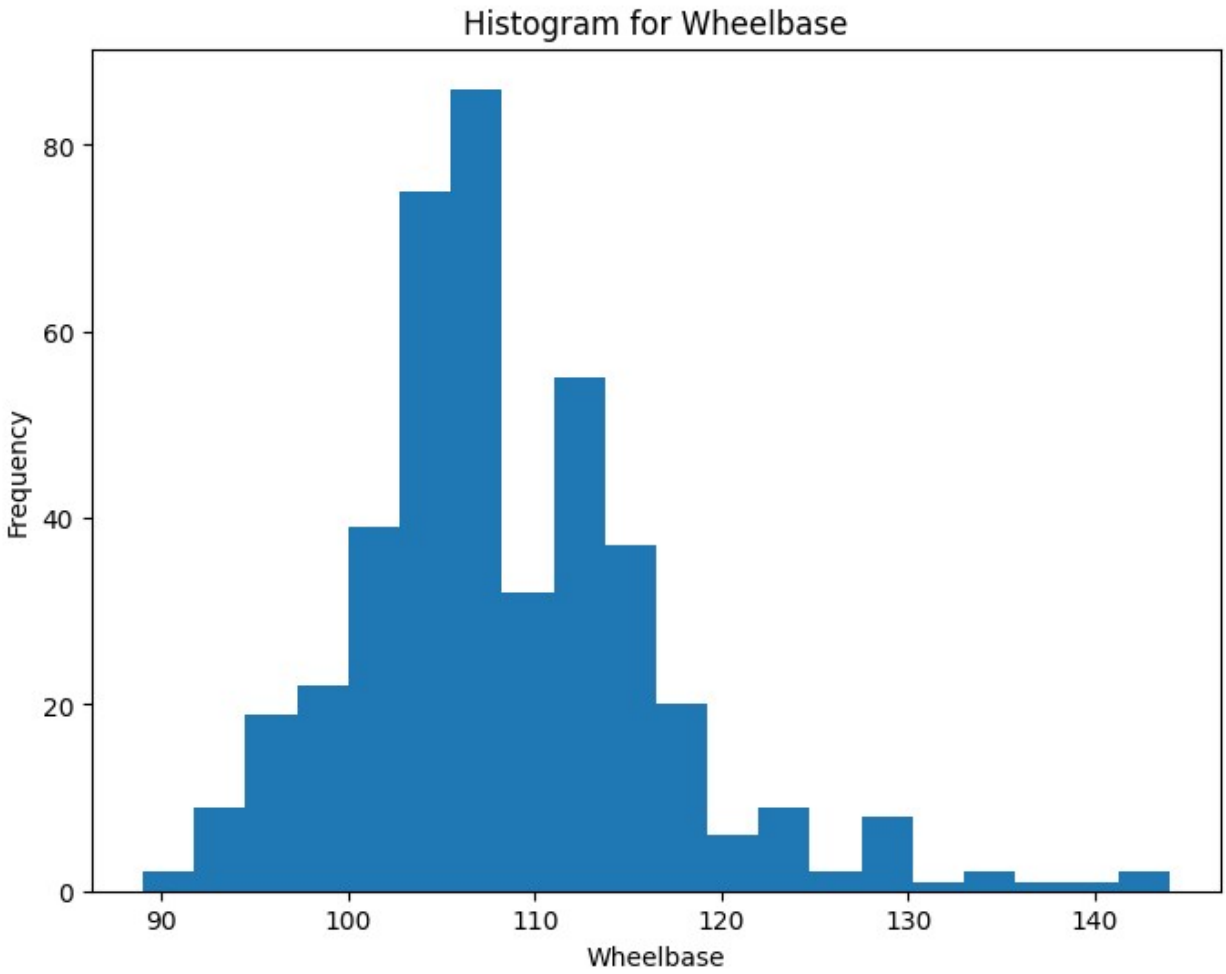

Histogram for Cylinders
Frequency
175
150
125
100
75
50
25
0
4
6
8
10
12
Cylinders

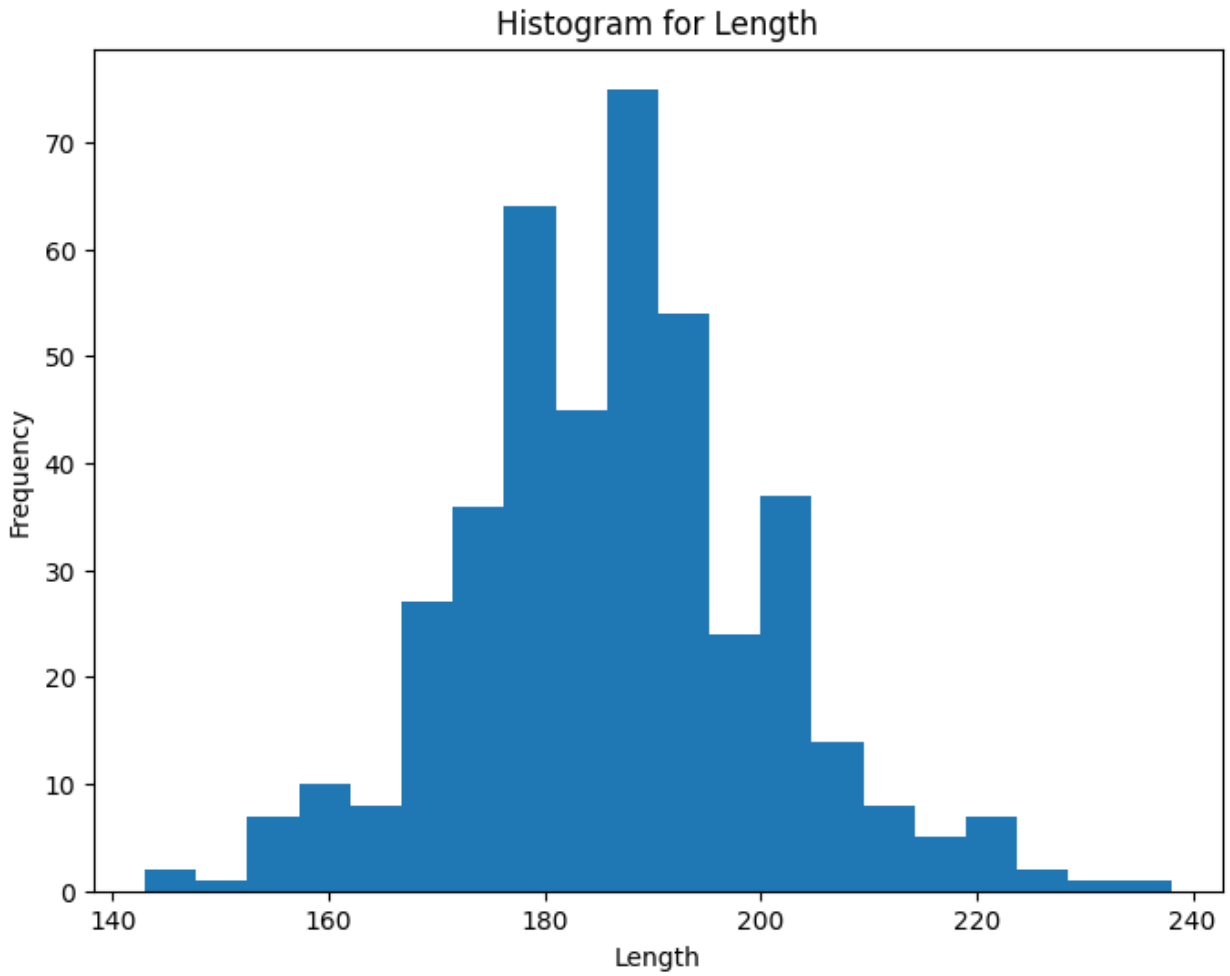

Histogram for Horsepower
Frequency
0
10
20
30
40
50
60
100
200
300
400
500
Horsepower


Histogram for MPG_City
Frequency
0
20
40
60
80
100
10
20
30
40
50
60
MPG_City


Histogram for MPG_Highway
Frequency
MPG_Highway
0
20
40
60
80
100
120
10
20
30
40
50
60


Histogram for Weight
Frequency
Weight
0
10
20
30
40
50
60
70
80
2000
3000
4000
5000
6000
7000


Histogram for Wheelbase
Frequency
0
20
40
60
80
90
100
110
120
130
140
Wheelbase




## Outliers

```
def scaleData(df, column):
    return (df[column] - df[column].mean()) / df[column].std()

def findOutliers(df, columns):
    outliers = pd.DataFrame()
    for column in columns:
        mean = np.mean(df[column])
        std_dev = np.std(df[column])
        lower_bound = mean - (5 * std_dev)
        upper_bound = mean + (5 * std_dev)
        outliers = pd.concat([outliers, df[(df[column] < lower_bound)
| (df[column] > upper_bound)]], axis=0)
    return outliers.drop_duplicates()

scaledDf = scaleData(df, numericVariables.columns)
findOutliers(scaledDf, numericVariables.columns)
```

```
      Invoice  EngineSize  Cylinders  Horsepower  MPG_City
MPG_Highway  \
```

```
262  5.077922    2.077649   3.973511    3.857597 -1.347929    -
1.366170
334  8.136512    0.363768   0.123513    3.634868 -0.584311    -
0.495272
150 -0.686068   -1.079501  -1.801485   -1.989051  7.624588
6.820271
373 -0.628536   -1.530522  -1.159819   -1.473989  7.433684
4.207577
```

```
       Weight  Wheelbase    Length
262  1.179271   0.703312  0.671253
334 -0.588884  -1.823213 -0.791347
150 -2.276669  -1.582592 -2.184299
373 -0.906414  -0.259174 -0.791347
```

## Normal tests

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

def getNormalityTest(numericVariables):
    results = pd.DataFrame(columns=['Variable', 'Statistical', 'p-
value', 'Normal distribution'])

    for column in numericVariables.columns:
        stat, pValue = shapiro(numericVariables[column])
        normalDistribution = pValue > 0.05
        results = results._append({
            'Variable': column,
            'Statistical': stat,
            'P-value': pValue,
            'Normal distribution': 1 if normalDistribution else 0
        }, ignore_index=True)

    return results

print(getNormalityTest(numericVariables))
print(f"\nNormal Distribution:\n\t- 1: It's a normal distribution.\n\
t- 0: It isn't a normal distribution")
```

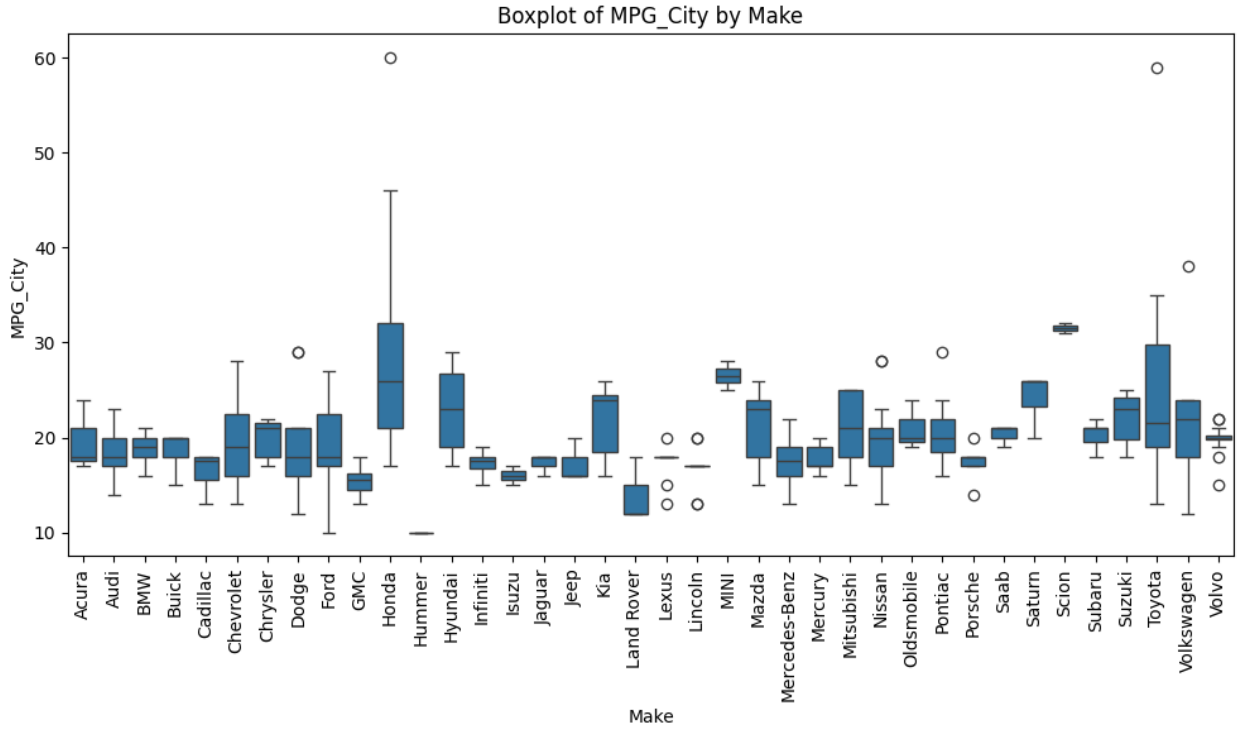
```
      Variable  Statistical p-value Normal distribution       P-value
0      Invoice     0.772256     NaN                   0  6.070299e-24
1   EngineSize     0.958561     NaN                   0  1.298519e-09
2    Cylinders          NaN     NaN                   1  1.000000e+00
3   Horsepower     0.949922     NaN                   0  7.410131e-11
4     MPG_City     0.807840     NaN                   0  3.389670e-22
5  MPG_Highway     0.929870     NaN                   0  2.776713e-13
6       Weight     0.958916     NaN                   0  1.472114e-09
7    Wheelbase     0.949997     NaN                   0  7.587275e-11
8       Length     0.991180     NaN                   0  1.183925e-02
```

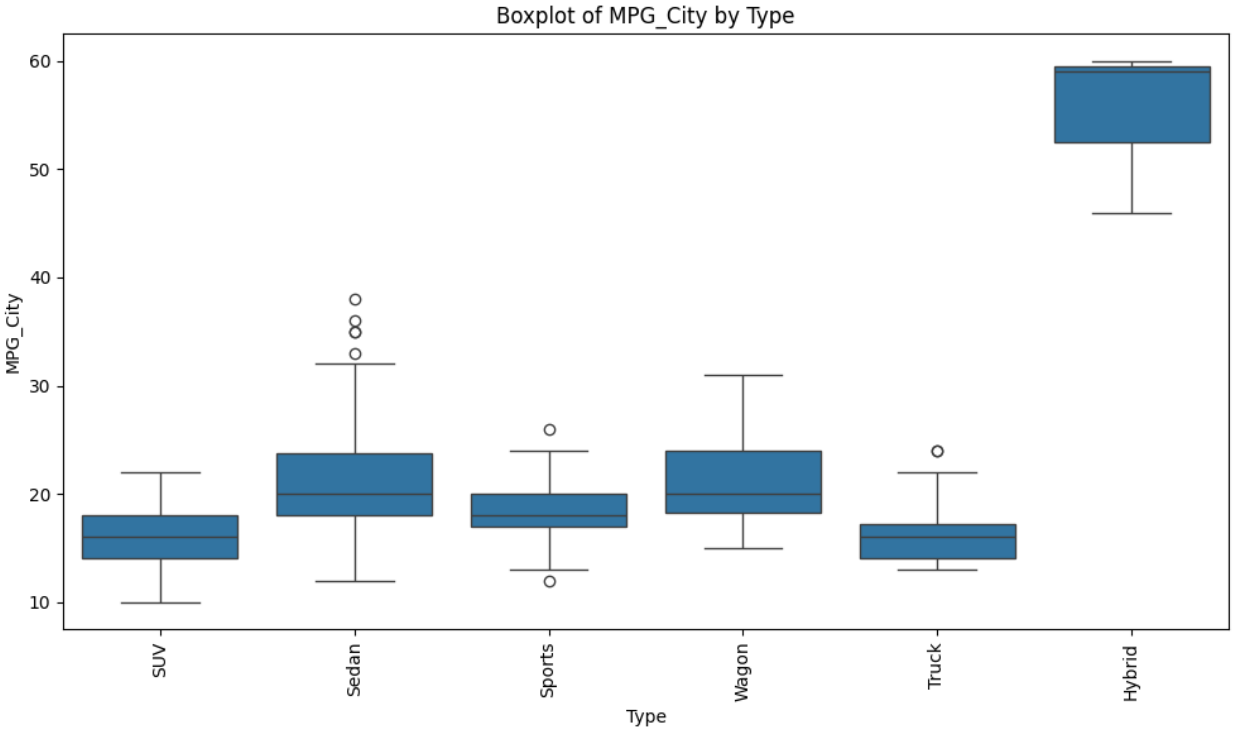
```
Normal Distribution:
     - 1: It's a normal distribution.
     - 0: It isn't a normal distribution
```

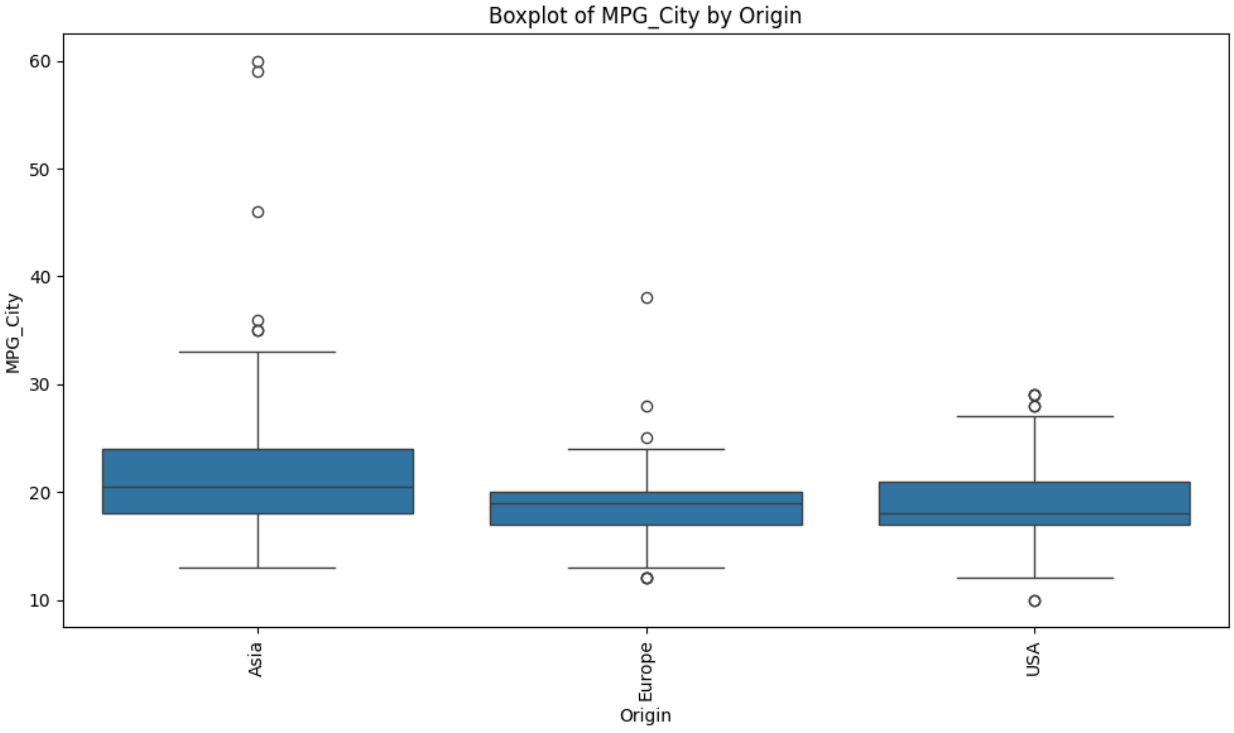
# 2. Graph of the relationship of each variable with respect to MPG_City

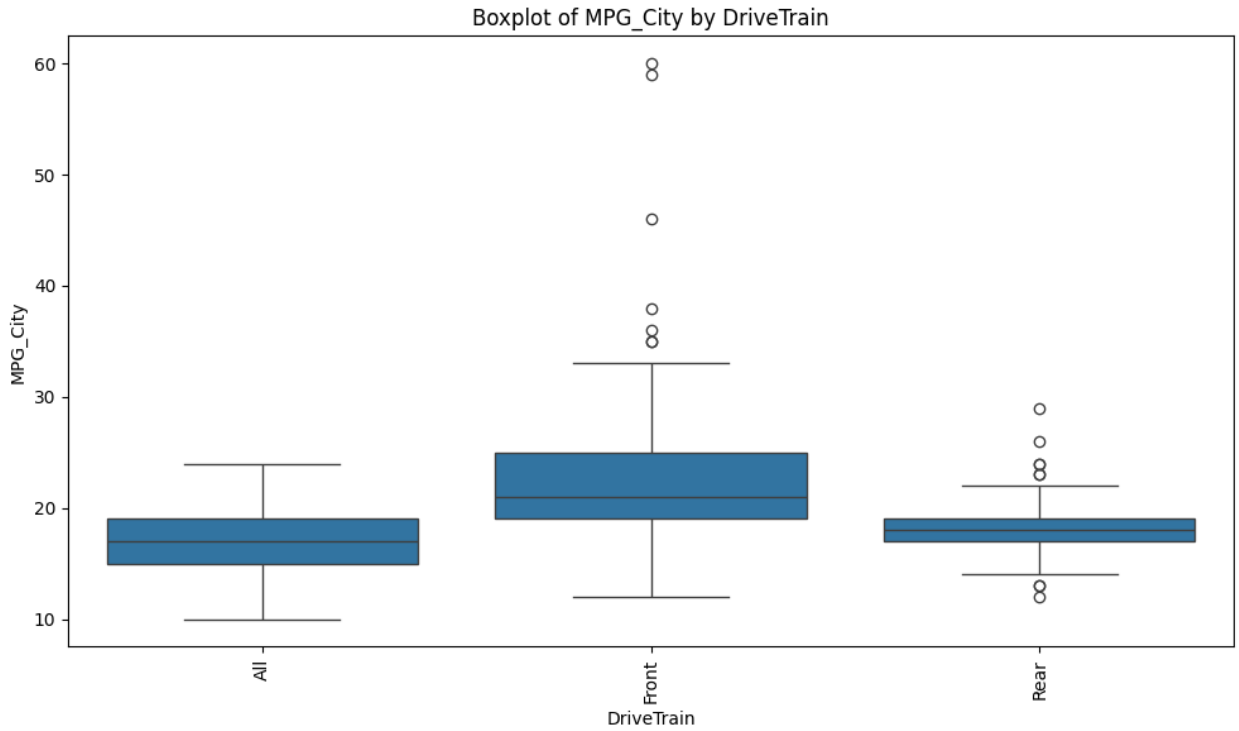
## 2.1. Categorical variables you must create a boxplot. Explain how you interpret the graph

```
import seaborn as sns
selectedVariable = 'MPG_City'

def graphVsBoxplots(variables, selectedVariable):
    for column in variables.columns:
        plt.figure(figsize=(10, 6))
        sns.boxplot(x=column, y=selectedVariable, data=df)
        plt.title(f'Boxplot of {selectedVariable} by {column}')
        plt.xlabel(column)
        plt.ylabel(selectedVariable)
        plt.xticks(rotation=90)
        plt.tight_layout()
        plt.show()

graphVsBoxplots(categoricalVariables, selectedVariable)
```




Boxplot of MPG_City by Type
MPG_City
60
50
40
30
20
10
SUV
Sedan
Sports
Wagon
Truck
Hybrid
Type


Boxplot of MPG_City by Origin
MPG_City
60
50
40
30
20
10
Asia
Europe
USA
Origin




## OBSERVATIONS:

### MPG_City BY make

- Hummer, Lexus, Lincoln, Scion, Volvo have the least data variance
- The highest MPG_City performance is achieved by Toyota and Honda

### MPG_City BY Type

- Hybrid cars are significantly more efficient in the city
- Sedans have too many upper outliers

### MPG_City BY Origin

- Cars of Asian origin have the best MPG City performance and the greatest number of outliers

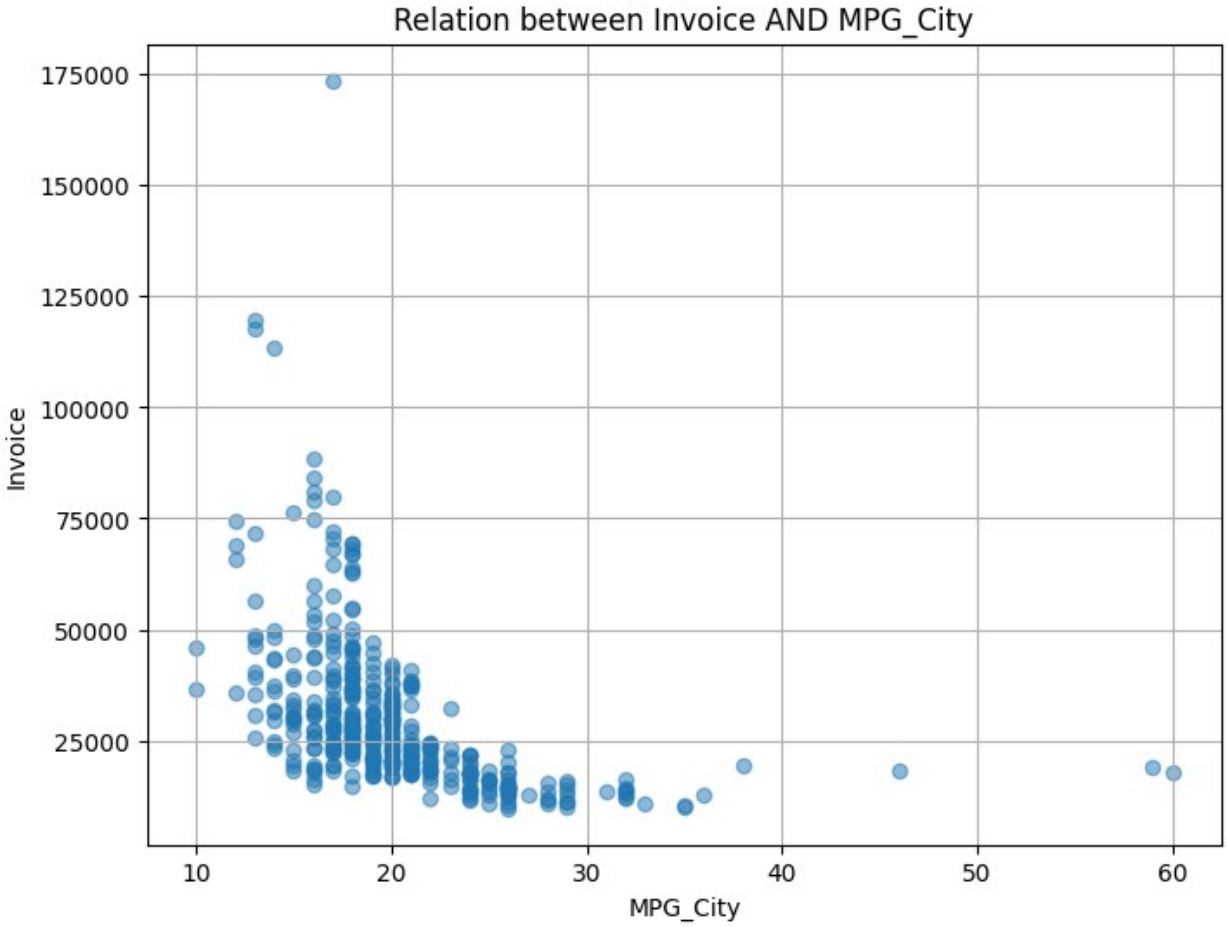
### MPG_City BY DriveTrain

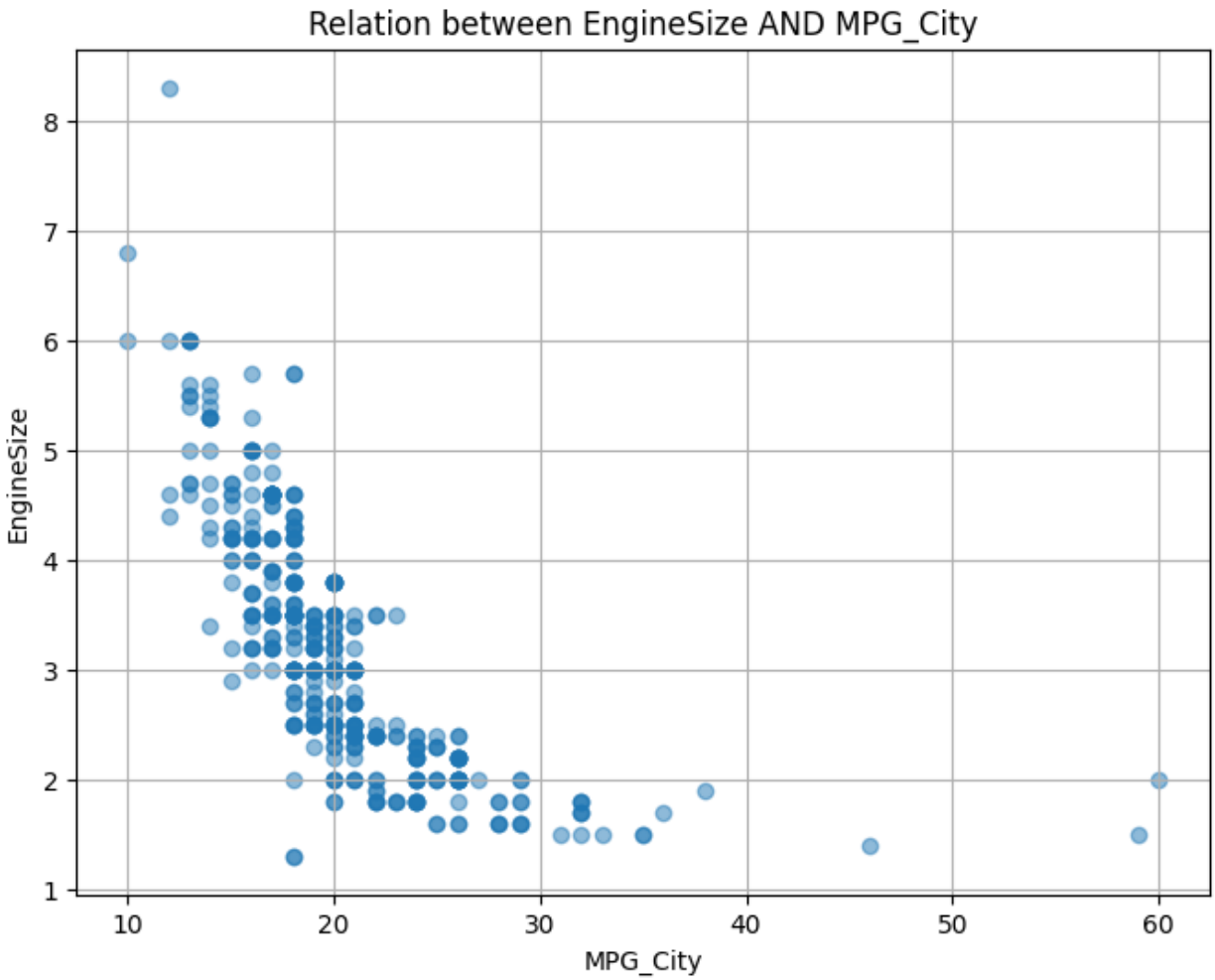
- The all drivetrain is the variable with the fewest outliers
- The front drivetrain has the best MPG_city performance, and also has the greatest number of outliers

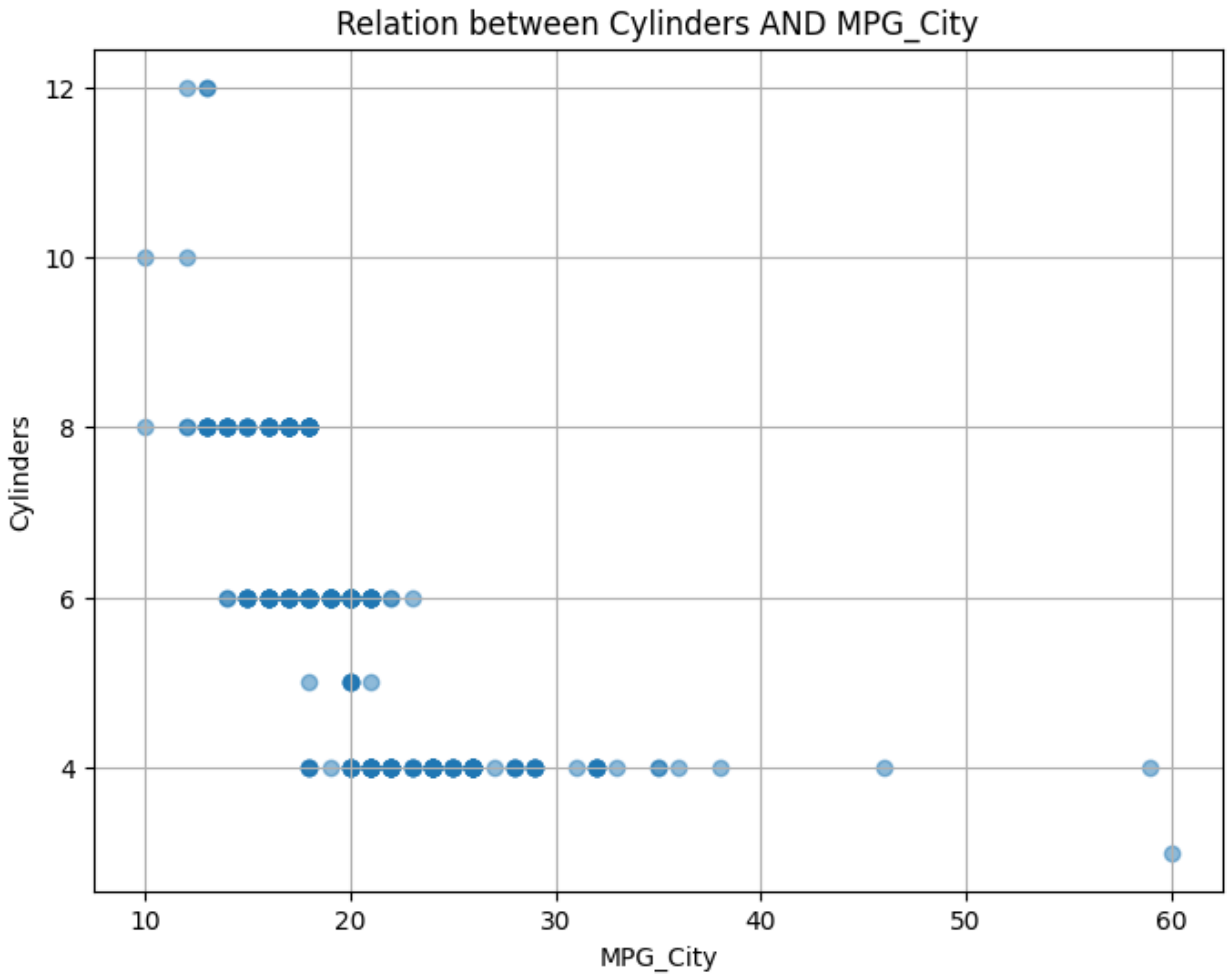
## 2.2. For numerical variables you are going to create a scatter plot. Explain how you interpret the graph

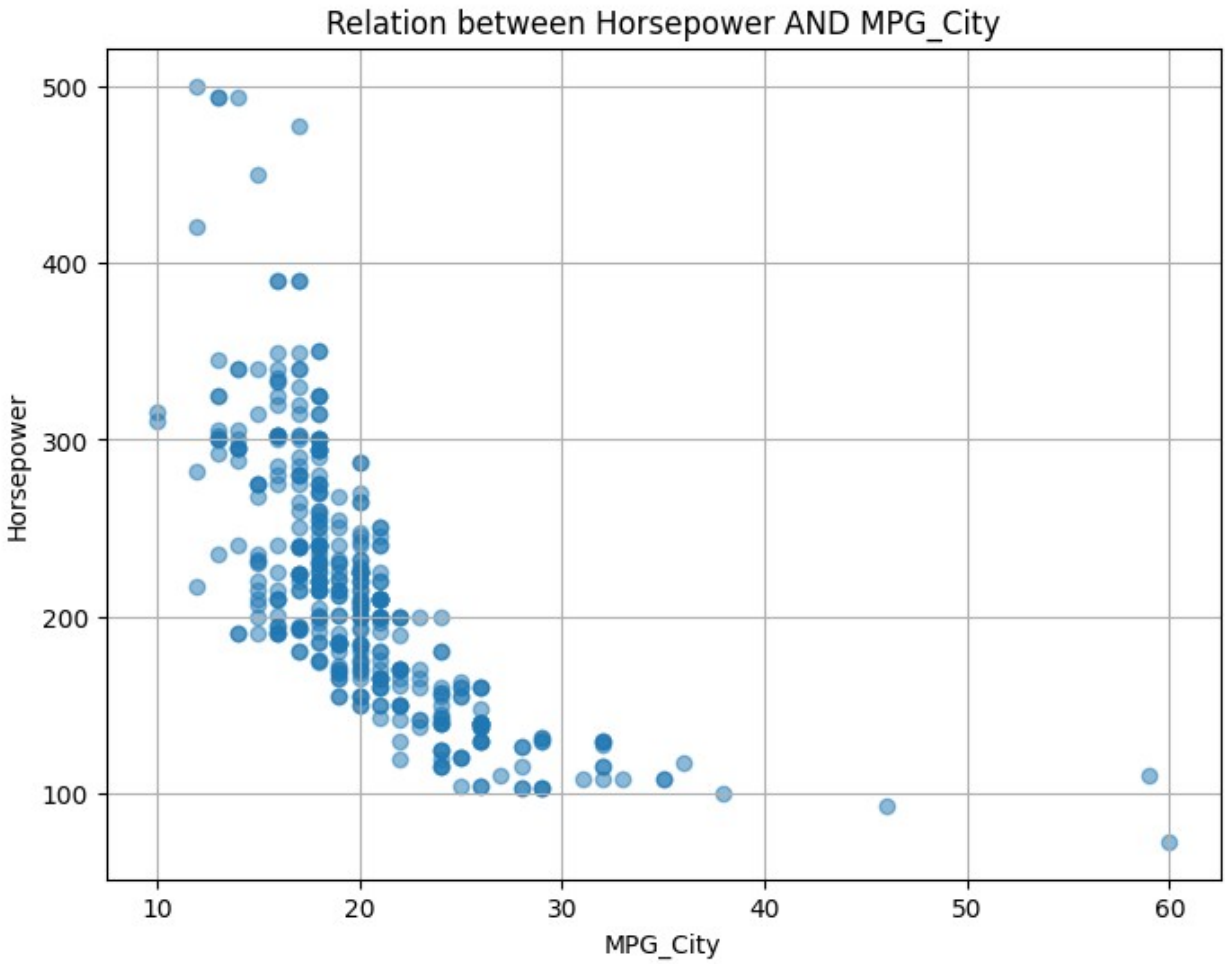
```
def graphScatter(variables, selectedVariable):
    for variable in variables:
        if variable != selectedVariable:
            plt.figure(figsize=(8, 6))
```

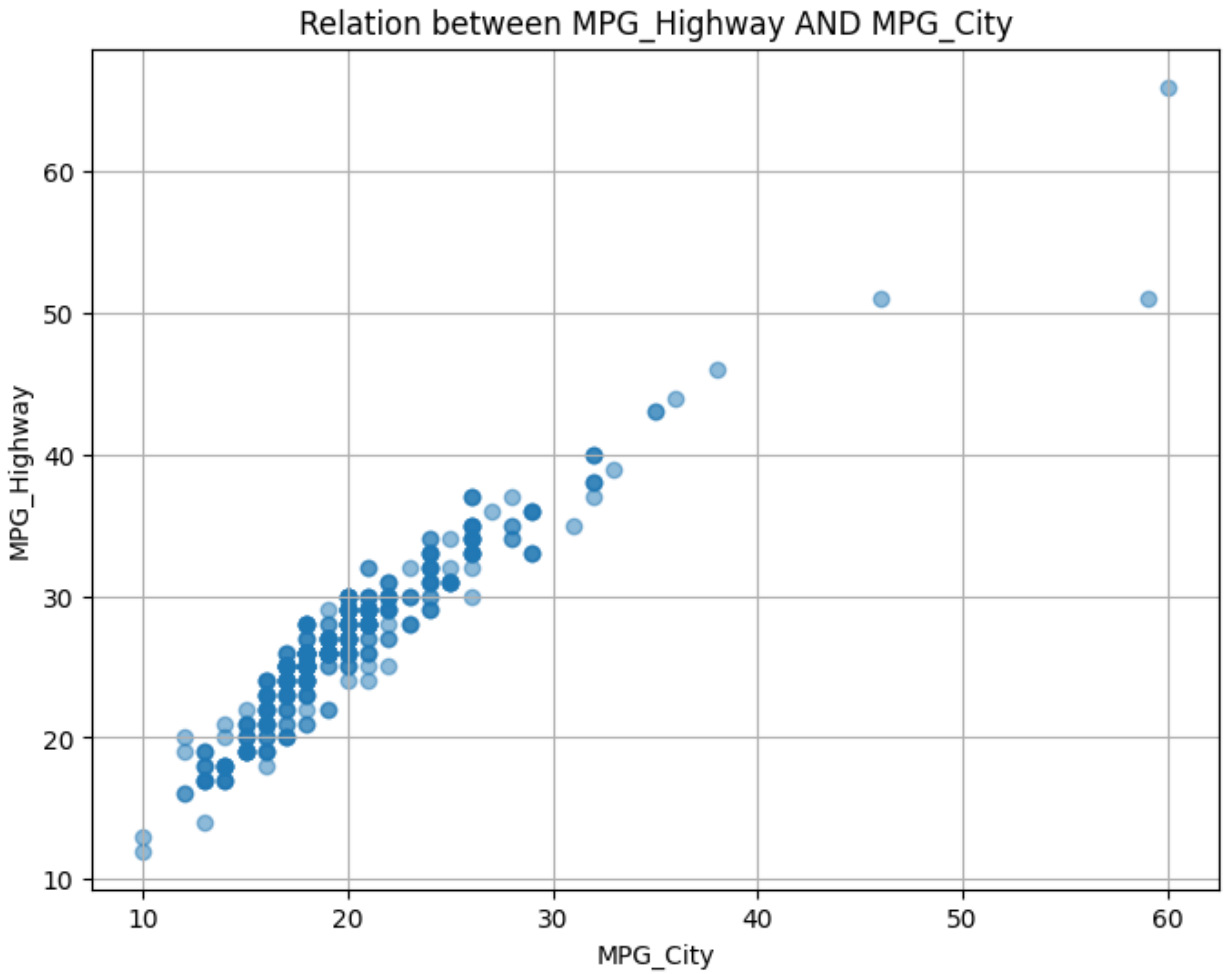
```
            plt.scatter(numericVariables[selectedVariable],
numericVariables[variable], alpha=0.5)
            plt.title(f'Relation between {variable} AND
{selectedVariable}')
            plt.xlabel(selectedVariable)
            plt.ylabel(variable)
            plt.grid(True)
            plt.show()

variablesToCompare = numericVariables.columns.to_list
graphScatter(numericVariables, selectedVariable)
```

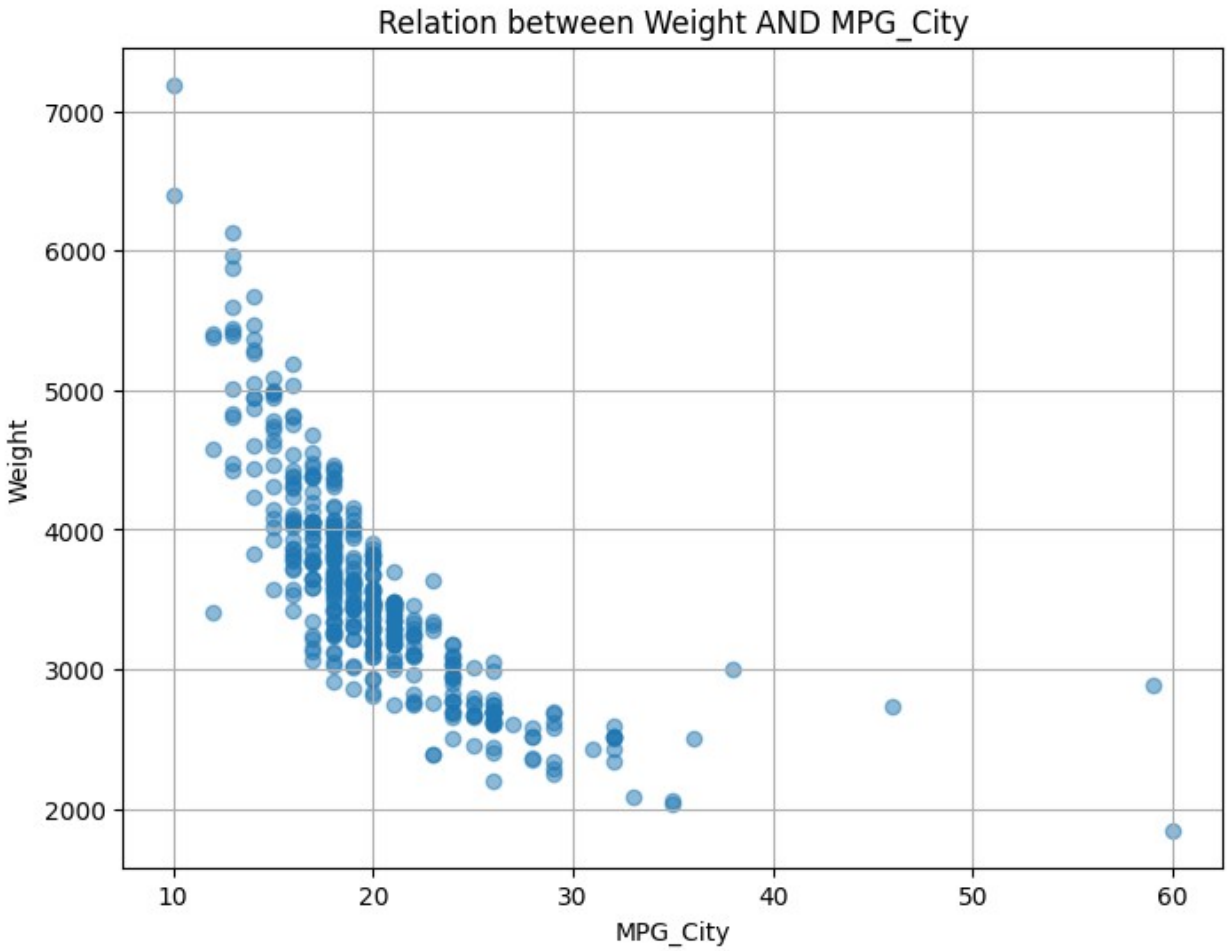


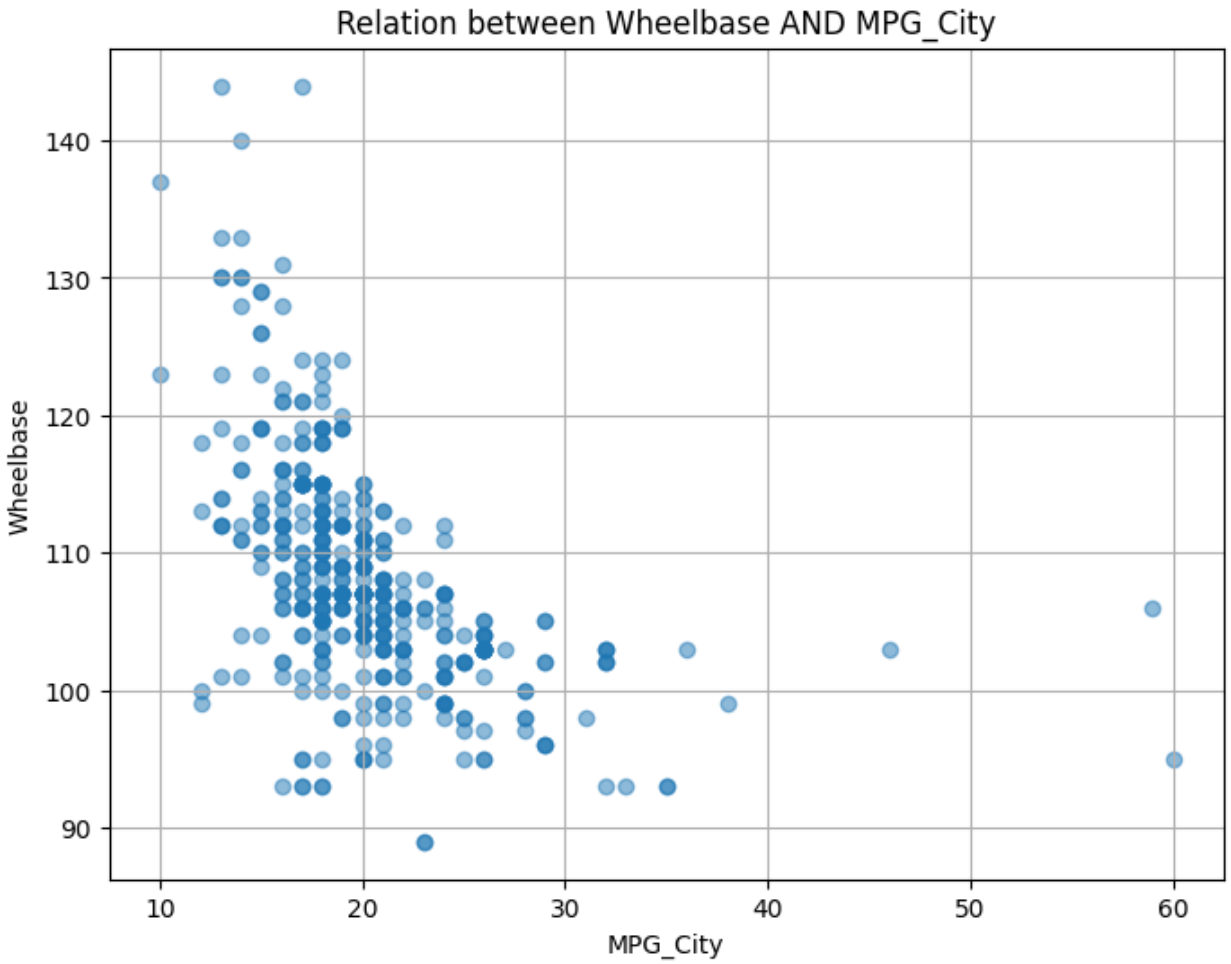

Relation between EngineSize AND MPG_City
EngineSize
MPG_City
1
2
3
4
5
6
7
8
10
20
30
40
50
60

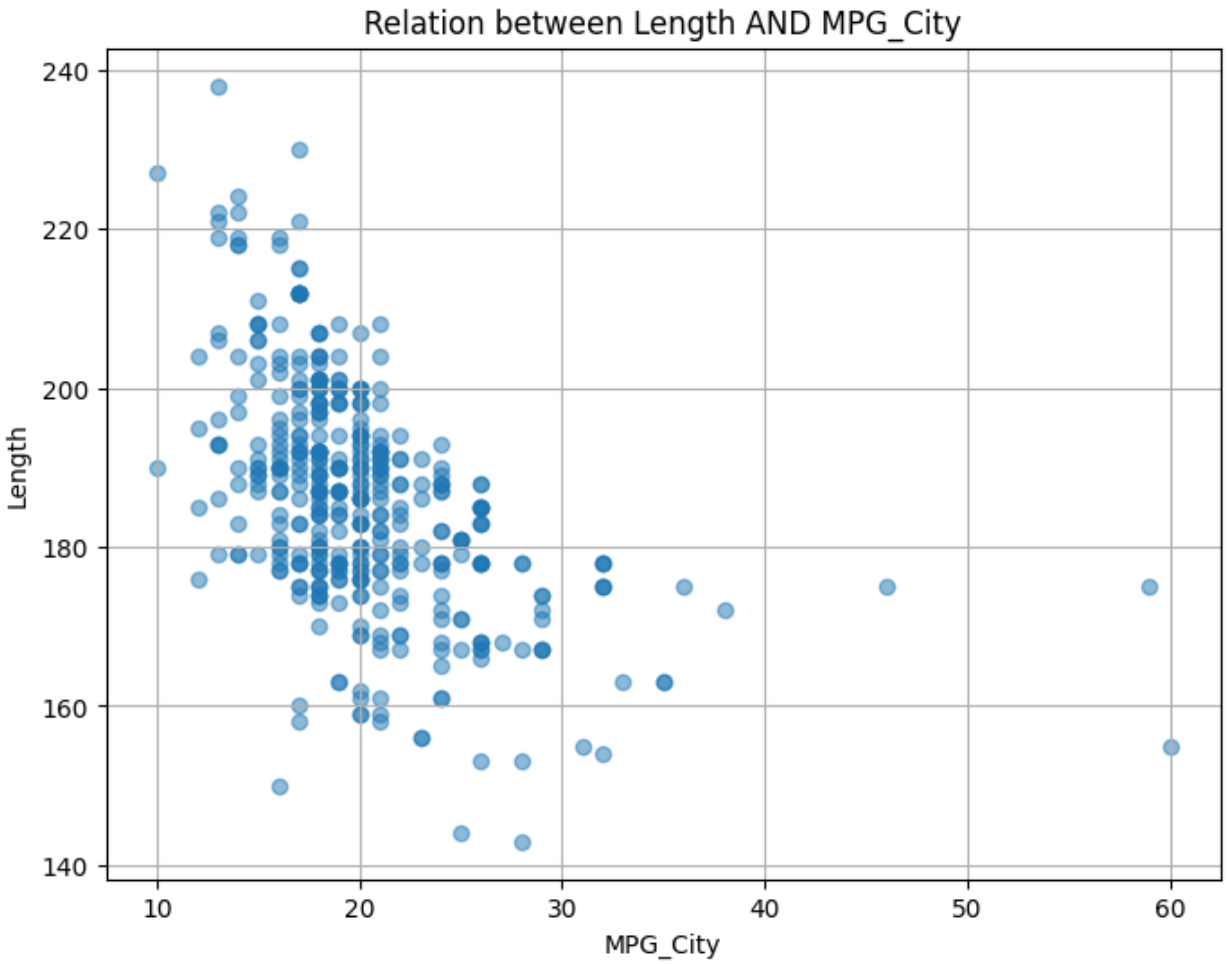

Relation between Cylinders AND MPG_City
Cylinders
MPG_City
12
10
8
6
4
10
20
30
40
50
60


Relation between Horsepower AND MPG_City
Horsepower
MPG_City
500
400
300
200
100
10
20
30
40
50
60


Relation between MPG_Highway AND MPG_City
MPG_Highway
MPG_City
10
20
30
40
50
60


Relation between Weight AND MPG_City
Weight
MPG_City
7000
6000
5000
4000
3000
2000
10
20
30
40
50
60


Relation between Wheelbase AND MPG_City
Wheelbase
MPG_City
140
130
120
110
100
90
10
20
30
40
50
60




## OBSERVATIONS:

### MPG_City AND Invoice

- The higher the price of the car, the lower the MPG_City performance may be.

### MPG_City AND EngineSize

- The larger the engine size, the lower the highway performance, the lower the MPG_City performance.

### MPG_City AND Cylinders

- The greater the number of cylinders, the lower the MPG_City performance.

### MPG_City AND Horsepower

- The lower the horsepower, the higher the MPG_City performance.

### MPG_City AND MPG_Highway

- The relationship is almost 1 to 1, the more MPG_Highway performance the more MPG_City performance.

MPG_City AND Weight

- The higher the weight there is a slightly lower performance in MPG_City.

MPG_City AND Wheelbase

- The greater the distance between the tires there is a slightly lower performance in MPG_City, since the separated tires are usually for trucks.

MPG_City AND Length

- The larger the size, there is a slightly lower performance in MPG_City.

# 3. Correlation Matrix

## 3.1. Create the correlation matrix, which are the most important variables to explain the variability of MPG_City. Explain why the coefficient is negative or positive

```
pd.concat([numericVariables, df[selectedVariable]]).corr()
```

```
              Invoice  EngineSize  Cylinders  Horsepower  MPG_City  \
Invoice      1.000000    0.564498   0.645226    0.823746 -0.470442
EngineSize   0.564498    1.000000   0.908002    0.787435 -0.709471
Cylinders    0.645226    0.908002   1.000000    0.810341 -0.684402
Horsepower   0.823746    0.787435   0.810341    1.000000 -0.676699
MPG_City    -0.470442   -0.709471  -0.684402   -0.676699  1.000000
MPG_Highway -0.434585   -0.717302  -0.676100   -0.647195  0.941021
Weight       0.442332    0.807867   0.742209    0.630796 -0.737966
Wheelbase    0.148328    0.636517   0.546730    0.387398 -0.507284
Length       0.166586    0.637448   0.547783    0.381554 -0.501526

             MPG_Highway    Weight  Wheelbase    Length
Invoice        -0.434585  0.442332   0.148328  0.166586
EngineSize     -0.717302  0.807867   0.636517  0.637448
Cylinders      -0.676100  0.742209   0.546730  0.547783
Horsepower     -0.647195  0.630796   0.387398  0.381554
MPG_City        0.941021 -0.737966  -0.507284 -0.501526
MPG_Highway     1.000000 -0.790989  -0.524661 -0.466092
Weight         -0.790989  1.000000   0.760703  0.690021
Wheelbase      -0.524661  0.760703   1.000000  0.889195
Length         -0.466092  0.690021   0.889195  1.000000
```

1. **Invoice (≈-0.47):** There is a moderate negative correlation between the price of the car (Invoice) and the mileage per gallon in the city (MPG_City). This means that more expensive vehicles could use more fuel in the city.

2. **EngineSize (≈-0.71):** There is a strong negative correlation between engine size (EngineSize) and city miles per gallon (MPG_City). This means that larger engines tend to use more fuel in the city.

3. **Cylinders (≈-0.68):** There is a strong negative correlation between the number of cylinders (Cylinders) and miles per gallon in the city (MPG_City). This is consistent with the previous observation because engines with more cylinders tend to be larger and consume more fuel.

4. **MPG_Highway (≈0.94):** There is a very strong positive correlation between highway miles per gallon (MPG_Highway) and city miles per gallon (MPG_City). This means that vehicles that are more efficient on the highway will also be more efficient in the city.

5. **Weight (≈-0.73):** There is a strong negative correlation between vehicle weight and city miles per gallon (MPG_City). This means that the more weight the vehicle has, the more gasoline it will use, therefore it will be less efficient in the city.

6. **Wheelbase (≈-0.50):** There is a moderate negative correlation between wheelbase (Wheelbase) and city miles per gallon (MPG_City). The further apart the tires are, the larger and heavier the vehicles can be, therefore they will be less efficient in the city.

7. **Length (≈-0.50):** Vehicle length (Length) shows a moderate negative correlation with city miles per gallon (MPG_City). This indicates that longer vehicles may perform less efficiently in the city.

## 3.2 Create dummy variables for all categorical variables and generate the correlation matrix again. What is the value of categorical variable with the highest correlation?

```
dummiesVariablees = pd.DataFrame(df[categoricalVariables.columns])
pd.get_dummies(dummiesVariablees[categoricalVariables.columns]).corr()
```

```
                  Make_Acura  Make_Audi  Make_BMW  Make_Buick  \
Make_Acura          1.000000  -0.027792 -0.028549   -0.018898
Make_Audi          -0.027792   1.000000 -0.047720   -0.031589
Make_BMW           -0.028549  -0.047720  1.000000   -0.032449
Make_Buick         -0.018898  -0.031589 -0.032449    1.000000
Make_Cadillac      -0.017796  -0.029746 -0.030557   -0.020227
Make_Chevrolet     -0.033459  -0.055927 -0.057451   -0.038030
Make_Chrysler      -0.024574  -0.041076 -0.042194   -0.027931
Make_Dodge         -0.022822  -0.038147 -0.039186   -0.025940
Make_Ford          -0.030729  -0.051363 -0.052762   -0.034926
Make_GMC           -0.017796  -0.029746 -0.030557   -0.020227
Make_Honda         -0.026225  -0.043835 -0.045029   -0.029807
Make_Hummer        -0.006240  -0.010430 -0.010714   -0.007093
Make_Hyundai       -0.021900  -0.036607 -0.037604   -0.024892
Make_Infiniti      -0.017796  -0.029746 -0.030557   -0.020227
Make_Isuzu         -0.008835  -0.014768 -0.015170   -0.010042
Make_Jaguar        -0.021900  -0.036607 -0.037604   -0.024892
Make_Jeep          -0.010834  -0.018108 -0.018602   -0.012313
Make_Kia           -0.020943  -0.035006 -0.035959   -0.023804
```

```
Make_Land Rover      -0.010834  -0.018108 -0.018602   -0.012313
Make_Lexus           -0.020943  -0.035006 -0.035959   -0.023804
Make_Lincoln         -0.018898  -0.031589 -0.032449   -0.021480
Make_MINI            -0.008835  -0.014768 -0.015170   -0.010042
Make_Mazda           -0.020943  -0.035006 -0.035959   -0.023804
Make_Mercedes-Benz   -0.032793  -0.054814 -0.056307   -0.037272
Make_Mercury         -0.018898  -0.031589 -0.032449   -0.021480
Make_Mitsubishi      -0.022822  -0.038147 -0.039186   -0.025940
Make_Nissan          -0.026225  -0.043835 -0.045029   -0.029807
Make_Oldsmobile      -0.010834  -0.018108 -0.018602   -0.012313
Make_Pontiac         -0.020943  -0.035006 -0.035959   -0.023804
Make_Porsche         -0.016627  -0.027792 -0.028549   -0.018898
Make_Saab            -0.016627  -0.027792 -0.028549   -0.018898
Make_Saturn          -0.017796  -0.029746 -0.030557   -0.020227
Make_Scion           -0.008835  -0.014768 -0.015170   -0.010042
Make_Subaru          -0.020943  -0.035006 -0.035959   -0.023804
Make_Suzuki          -0.017796  -0.029746 -0.030557   -0.020227
Make_Toyota          -0.034116  -0.057025 -0.058578   -0.038776
Make_Volkswagen      -0.024574  -0.041076 -0.042194   -0.027931
Make_Volvo           -0.021900  -0.036607 -0.037604   -0.024892
Type_Hybrid          -0.010834  -0.018108 -0.018602   -0.012313
Type_SUV              0.000992  -0.087030 -0.025628    0.034631
Type_Sedan            0.027029   0.031875  0.017199    0.049818
Type_Sports           0.011490   0.065012  0.059463   -0.052698
Type_Truck           -0.031428  -0.052533 -0.053963   -0.035721
Type_Wagon           -0.035402   0.029691 -0.017425   -0.040238
Origin_Asia           0.168563  -0.164878 -0.169368   -0.112114
Origin_Europe        -0.081886   0.339401  0.348644   -0.093072
Origin_USA           -0.093264  -0.155891 -0.160137    0.202632
DriveTrain_All       -0.022631   0.218596  0.018889   -0.037048
DriveTrain_Front      0.048108  -0.068914 -0.234187    0.105939
DriveTrain_Rear      -0.033684  -0.126765  0.249780   -0.086198

                    Make_Cadillac  Make_Chevrolet  Make_Chrysler
Make_Dodge  \
Make_Acura              -0.017796       -0.033459      -0.024574   -
0.022822
Make_Audi               -0.029746       -0.055927      -0.041076   -
0.038147
Make_BMW                -0.030557       -0.057451      -0.042194   -
0.039186
Make_Buick              -0.020227       -0.038030      -0.027931   -
0.025940
Make_Cadillac            1.000000       -0.035812      -0.026302   -
0.024427
Make_Chevrolet          -0.035812        1.000000      -0.049452   -
0.045926
Make_Chrysler           -0.026302       -0.049452       1.000000   -
0.033730
```

```
Make_Dodge                -0.024427       -0.045926       -0.033730
1.000000
Make_Ford                 -0.032889       -0.061837       -0.045416   -
0.042178
Make_GMC                  -0.019048       -0.035812       -0.026302   -
0.024427
Make_Honda                -0.028069       -0.052773       -0.038759   -
0.035996
Make_Hummer               -0.006679       -0.012557       -0.009223   -
0.008565
Make_Hyundai              -0.023440       -0.044071       -0.032368   -
0.030060
Make_Infiniti             -0.019048       -0.035812       -0.026302   -
0.024427
Make_Isuzu                -0.009457       -0.017780       -0.013058   -
0.012127
Make_Jaguar               -0.023440       -0.044071       -0.032368   -
0.030060
Make_Jeep                 -0.011595       -0.021801       -0.016012   -
0.014870
Make_Kia                  -0.022416       -0.042144       -0.030953   -
0.028746
Make_Land Rover           -0.011595       -0.021801       -0.016012   -
0.014870
Make_Lexus                -0.022416       -0.042144       -0.030953   -
0.028746
Make_Lincoln              -0.020227       -0.038030       -0.027931   -
0.025940
Make_MINI                 -0.009457       -0.017780       -0.013058   -
0.012127
Make_Mazda                -0.022416       -0.042144       -0.030953   -
0.028746
Make_Mercedes-Benz        -0.035099       -0.065991       -0.048467   -
0.045011
Make_Mercury              -0.020227       -0.038030       -0.027931   -
0.025940
Make_Mitsubishi           -0.024427       -0.045926       -0.033730   -
0.031325
Make_Nissan               -0.028069       -0.052773       -0.038759   -
0.035996
Make_Oldsmobile           -0.011595       -0.021801       -0.016012   -
0.014870
Make_Pontiac              -0.022416       -0.042144       -0.030953   -
0.028746
Make_Porsche              -0.017796       -0.033459       -0.024574   -
0.022822
Make_Saab                 -0.017796       -0.033459       -0.024574   -
0.022822
Make_Saturn               -0.019048       -0.035812       -0.026302   -
```

```
0.024427
Make_Scion               -0.009457       -0.017780      -0.013058   -
0.012127
Make_Subaru              -0.022416       -0.042144      -0.030953   -
0.028746
Make_Suzuki              -0.019048       -0.035812      -0.026302   -
0.024427
Make_Toyota              -0.036515       -0.068653      -0.050422   -
0.046827
Make_Volkswagen          -0.026302       -0.049452      -0.036320   -
0.033730
Make_Volvo               -0.023440       -0.044071      -0.032368   -
0.030060
Type_Hybrid              -0.011595       -0.021801      -0.016012   -
0.014870
Type_SUV                  0.043653        0.005950      -0.076952   -
0.032251
Type_Sedan               -0.031766       -0.030138       0.099547
0.001175
Type_Sports               0.004557       -0.032934      -0.028622   -
0.020880
Type_Truck                0.041347        0.145619      -0.046450
0.134392
Type_Wagon               -0.037891       -0.033598      -0.002558   -
0.048592
Origin_Asia              -0.105576       -0.198498      -0.145786   -
0.135392
Origin_Europe            -0.087644       -0.164783      -0.121024   -
0.112396
Origin_USA                0.190816        0.358760       0.263491
0.244705
DriveTrain_All           -0.030222       -0.018804      -0.099723   -
0.026328
DriveTrain_Front          0.026806        0.072057       0.129273
0.003696
DriveTrain_Rear          -0.002214       -0.064642      -0.053937
0.020528

                    Make_Ford  Make_GMC  ...  Type_Sedan  Type_Sports
\
Make_Acura          -0.030729 -0.017796  ...    0.027029     0.011490

Make_Audi           -0.051363 -0.029746  ...    0.031875     0.065012

Make_BMW            -0.052762 -0.030557  ...    0.017199     0.059463

Make_Buick          -0.034926 -0.020227  ...    0.049818    -0.052698

Make_Cadillac       -0.032889 -0.019048  ...   -0.031766     0.004557
```

| | | | | | |
|---|---|---|---|---|---|
| Make_Chevrolet | -0.061837 | -0.035812 | ... | -0.030138 | -0.032934 |
| Make_Chrysler | -0.045416 | -0.026302 | ... | 0.099547 | -0.028622 |
| Make_Dodge | -0.042178 | -0.024427 | ... | 0.001175 | -0.020880 |
| Make_Ford | 1.000000 | -0.032889 | ... | -0.065482 | 0.011937 |
| Make_GMC | -0.032889 | 1.000000 | ... | -0.137982 | -0.049625 |
| Make_Honda | -0.048466 | -0.028069 | ... | 0.014571 | -0.035554 |
| Make_Hummer | -0.011532 | -0.006679 | ... | -0.060797 | -0.017401 |
| Make_Hyundai | -0.040474 | -0.023440 | ... | 0.077097 | -0.016617 |
| Make_Infiniti | -0.032889 | -0.019048 | ... | 0.039045 | -0.049625 |
| Make_Isuzu | -0.016329 | -0.009457 | ... | -0.086081 | -0.024637 |
| Make_Jaguar | -0.040474 | -0.023440 | ... | 0.019003 | 0.116738 |
| Make_Jeep | -0.020022 | -0.011595 | ... | -0.105551 | -0.030210 |
| Make_Kia | -0.038705 | -0.022416 | ... | 0.068676 | -0.058399 |
| Make_Land Rover | -0.020022 | -0.011595 | ... | -0.105551 | -0.030210 |
| Make_Lexus | -0.038705 | -0.022416 | ... | -0.022231 | -0.012027 |
| Make_Lincoln | -0.034926 | -0.020227 | ... | 0.049818 | -0.052698 |
| Make_MINI | -0.016329 | -0.009457 | ... | 0.054540 | -0.024637 |
| Make_Mazda | -0.038705 | -0.022416 | ... | -0.082836 | 0.127091 |
| Make_Mercedes-Benz | -0.060605 | -0.035099 | ... | 0.001688 | 0.062159 |
| Make_Mercury | -0.034926 | -0.020227 | ... | 0.049818 | -0.052698 |
| Make_Mitsubishi | -0.042178 | -0.024427 | ... | -0.054707 | 0.064638 |
| Make_Nissan | -0.048466 | -0.028069 | ... | -0.034534 | 0.002019 |
| Make_Oldsmobile | -0.020022 | -0.011595 | ... | 0.066876 | -0.030210 |
| Make_Pontiac | -0.038705 | -0.022416 | ... | 0.038373 | -0.012027 |
| Make_Porsche | -0.030729 | -0.017796 | ... | -0.161996 | 0.300762 |
| Make_Saab | -0.030729 | -0.017796 | ... | 0.064834 | -0.046365 |

```
Make_Saturn          -0.032889 -0.019048  ...    0.039045    -0.049625
Make_Scion           -0.016329 -0.009457  ...   -0.015771    -0.024637
Make_Subaru          -0.038705 -0.022416  ...   -0.022231     0.034346
Make_Suzuki          -0.032889 -0.019048  ...    0.003640    -0.049625
Make_Toyota          -0.063050 -0.036515  ...   -0.022111    -0.035779
Make_Volkswagen      -0.045416 -0.026302  ...    0.047397    -0.068525
Make_Volvo           -0.040474 -0.023440  ...    0.048050    -0.061069
Type_Hybrid          -0.020022 -0.011595  ...   -0.105551    -0.030210
Type_SUV              0.023150  0.093344  ...   -0.507281    -0.145188
Type_Sedan           -0.065482 -0.137982  ...    1.000000    -0.451726
Type_Sports           0.011937 -0.049625  ...   -0.451726     1.000000
Type_Truck            0.077024  0.266304  ...   -0.306204    -0.087638
Type_Wagon            0.015740 -0.037891  ...   -0.344918    -0.098718
Origin_Asia          -0.182298 -0.105576  ...   -0.027023    -0.016556
Origin_Europe        -0.151335 -0.087644  ...    0.028668     0.144611
Origin_USA            0.329481  0.190816  ...    0.000142    -0.120989
DriveTrain_All       -0.023808  0.053771  ...   -0.330548    -0.098833
DriveTrain_Front     -0.003007 -0.076867  ...    0.390494    -0.262733
DriveTrain_Rear       0.025816  0.037265  ...   -0.135369     0.393055

                     Type_Truck  Type_Wagon  Origin_Asia  Origin_Europe
\
Make_Acura            -0.031428   -0.035402     0.168563      -0.081886
Make_Audi             -0.052533    0.029691    -0.164878       0.339401
Make_BMW              -0.053963   -0.017425    -0.169368       0.348644
Make_Buick            -0.035721   -0.040238    -0.112114      -0.093072
Make_Cadillac          0.041347   -0.037891    -0.105576      -0.087644
Make_Chevrolet         0.145619   -0.033598    -0.198498      -0.164783
```

| | | | | |
|---|---|---|---|---|
| Make_Chrysler | -0.046450 | -0.002558 | -0.145786 | -0.121024 |
| Make_Dodge | 0.134392 | -0.048592 | -0.135392 | -0.112396 |
| Make_Ford | 0.077024 | 0.015740 | -0.182298 | -0.151335 |
| Make_GMC | 0.266304 | -0.037891 | -0.105576 | -0.087644 |
| Make_Honda | -0.049570 | -0.055837 | 0.265862 | -0.129154 |
| Make_Hummer | -0.011795 | -0.013286 | -0.037020 | -0.030732 |
| Make_Hyundai | -0.041396 | -0.046630 | 0.222023 | -0.107857 |
| Make_Infiniti | -0.033638 | 0.097254 | 0.180415 | -0.087644 |
| Make_Isuzu | -0.016700 | -0.018812 | 0.089570 | -0.043512 |
| Make_Jaguar | -0.041396 | -0.046630 | -0.129924 | 0.267449 |
| Make_Jeep | -0.020478 | -0.023067 | -0.064271 | -0.053354 |
| Make_Kia | -0.039586 | 0.013242 | 0.212315 | -0.103141 |
| Make_Land Rover | -0.020478 | -0.023067 | -0.064271 | 0.132301 |
| Make_Lexus | -0.039586 | 0.013242 | 0.212315 | -0.103141 |
| Make_Lincoln | -0.035721 | -0.040238 | -0.112114 | -0.093072 |
| Make_MINI | -0.016700 | -0.018812 | -0.052415 | 0.107897 |
| Make_Mazda | 0.088769 | -0.044591 | 0.212315 | -0.103141 |
| Make_Mercedes-Benz | -0.061985 | 0.045116 | -0.194545 | 0.400471 |
| Make_Mercury | -0.035721 | 0.023547 | -0.112114 | -0.093072 |
| Make_Mitsubishi | -0.043138 | 0.004735 | 0.231367 | -0.112396 |
| Make_Nissan | 0.054430 | -0.008978 | 0.265862 | -0.129154 |
| Make_Oldsmobile | -0.020478 | -0.023067 | -0.064271 | -0.053354 |
| Make_Pontiac | -0.039586 | 0.013242 | -0.124244 | -0.103141 |
| Make_Porsche | -0.031428 | -0.035402 | -0.098640 | 0.203051 |
| Make_Saab | -0.031428 | 0.036751 | -0.098640 | 0.203051 |
| Make_Saturn | -0.033638 | 0.029682 | -0.105576 | -0.087644 |

| | | | | |
|---|---|---|---|---|
| Make_Scion | -0.016700 | 0.115379 | 0.089570 | -0.043512 |
| Make_Subaru | 0.024591 | 0.071075 | 0.212315 | -0.103141 |
| Make_Suzuki | -0.033638 | 0.029682 | 0.180415 | -0.087644 |
| Make_Toyota | 0.058728 | -0.035628 | 0.345862 | -0.168016 |
| Make_Volkswagen | -0.046450 | 0.096971 | -0.145786 | 0.300101 |
| Make_Volvo | -0.041396 | 0.064245 | -0.129924 | 0.267449 |
| Type_Hybrid | -0.020478 | -0.023067 | 0.109830 | -0.053354 |
| Type_SUV | -0.098416 | -0.110859 | 0.039751 | -0.107711 |
| Type_Sedan | -0.306204 | -0.344918 | -0.027023 | 0.028668 |
| Type_Sports | -0.087638 | -0.098718 | -0.016556 | 0.144611 |
| Type_Truck | 1.000000 | -0.066917 | -0.018094 | -0.154781 |
| Type_Wagon | -0.066917 | 1.000000 | -0.001418 | 0.068322 |
| Origin_Asia | -0.018094 | -0.001418 | 1.000000 | -0.485791 |
| Origin_Europe | -0.154781 | 0.068322 | -0.485791 | 1.000000 |
| Origin_USA | 0.165894 | -0.063670 | -0.553289 | -0.459312 |
| DriveTrain_All | 0.169126 | 0.056840 | 0.000441 | 0.120163 |
| DriveTrain_Front | -0.257806 | -0.033752 | 0.151006 | -0.289049 |
| DriveTrain_Rear | 0.135531 | -0.014875 | -0.172924 | 0.217251 |

| | Origin_USA | DriveTrain_All | DriveTrain_Front | \ |
|---|---|---|---|---|
| Make_Acura | -0.093264 | -0.022631 | 0.048108 | |
| Make_Audi | -0.155891 | 0.218596 | -0.068914 | |
| Make_BMW | -0.160137 | 0.018889 | -0.234187 | |
| Make_Buick | 0.202632 | -0.037048 | 0.105939 | |
| Make_Cadillac | 0.190816 | -0.030222 | 0.026806 | |
| Make_Chevrolet | 0.358760 | -0.018804 | 0.072057 | |
| Make_Chrysler | 0.263491 | -0.099723 | 0.129273 | |
| Make_Dodge | 0.244705 | -0.026328 | 0.003696 | |
| Make_Ford | 0.329481 | -0.023808 | -0.003007 | |
| Make_GMC | 0.190816 | 0.053771 | -0.076867 | |
| Make_Honda | -0.147099 | -0.019052 | 0.096418 | |
| Make_Hummer | 0.066908 | 0.092483 | -0.051188 | |

| | | | |
|---|---|---|---|
| Make_Hyundai | -0.122843 | -0.088873 | 0.160570 |
| Make_Infiniti | -0.099822 | 0.011775 | -0.111424 |
| Make_Isuzu | -0.049558 | 0.047545 | -0.003848 |
| Make_Jaguar | -0.122843 | -0.019964 | -0.179648 |
| Make_Jeep | 0.116161 | 0.092387 | -0.032768 |
| Make_Kia | -0.117472 | -0.084987 | 0.153550 |
| Make_Land Rover | -0.060768 | 0.160562 | -0.088868 |
| Make_Lexus | -0.117472 | 0.022842 | -0.142217 |
| Make_Lincoln | 0.202632 | -0.037048 | -0.122402 |
| Make_MINI | -0.049558 | -0.035854 | 0.064779 |
| Make_Mazda | -0.117472 | -0.013101 | -0.053487 |
| Make_Mercedes-Benz | -0.183941 | 0.009792 | -0.269000 |
| Make_Mercury | 0.202632 | -0.076690 | 0.008079 |
| Make_Mitsubishi | -0.128013 | -0.026328 | 0.112784 |
| Make_Nissan | -0.147099 | -0.048175 | 0.072453 |
| Make_Oldsmobile | 0.116161 | -0.043963 | 0.079431 |
| Make_Pontiac | 0.224555 | -0.049044 | 0.064820 |
| Make_Porsche | -0.093264 | 0.022212 | -0.136391 |
| Make_Saab | -0.093264 | -0.067473 | 0.121907 |
| Make_Saturn | 0.190816 | -0.030222 | 0.095922 |
| Make_Scion | -0.049558 | -0.035854 | 0.064779 |
| Make_Subaru | -0.117472 | 0.310387 | -0.171794 |
| Make_Suzuki | -0.099822 | 0.011775 | 0.061364 |
| Make_Toyota | -0.191361 | -0.023432 | 0.098709 |
| Make_Volkswagen | -0.137840 | -0.068794 | 0.154724 |
| Make_Volvo | -0.122843 | 0.083399 | 0.018813 |
| Type_Hybrid | -0.060768 | -0.043963 | 0.079431 |
| Type_SUV | 0.062251 | 0.411247 | -0.130524 |
| Type_Sedan | 0.000142 | -0.330548 | 0.390494 |
| Type_Sports | -0.120989 | -0.098833 | -0.262733 |
| Type_Truck | 0.165894 | 0.169126 | -0.257806 |
| Type_Wagon | -0.063670 | 0.056840 | -0.033752 |
| Origin_Asia | -0.553289 | 0.000441 | 0.151006 |
| Origin_Europe | -0.459312 | 0.120163 | -0.289049 |
| Origin_USA | 1.000000 | -0.114962 | 0.122003 |
| DriveTrain_All | -0.114962 | 1.000000 | -0.553481 |
| DriveTrain_Front | 0.122003 | -0.553481 | 1.000000 |
| DriveTrain_Rear | -0.031306 | -0.307756 | -0.622102 |

| | DriveTrain_Rear |
|---|---|
| Make_Acura | -0.033684 |
| Make_Audi | -0.126765 |
| Make_BMW | 0.249780 |
| Make_Buick | -0.086198 |
| Make_Cadillac | -0.002214 |
| Make_Chevrolet | -0.064642 |
| Make_Chrysler | -0.053937 |
| Make_Dodge | 0.020528 |
| Make_Ford | 0.025816 |

```
Make_GMC                 0.037265
Make_Honda              -0.092238
Make_Hummer             -0.028462
Make_Hyundai            -0.099891
Make_Infiniti            0.116223
Make_Isuzu              -0.040299
Make_Jaguar              0.223998
Make_Jeep               -0.049414
Make_Kia                -0.095524
Make_Land Rover         -0.049414
Make_Lexus               0.140996
Make_Lincoln             0.174660
Make_MINI               -0.040299
Make_Mazda               0.073419
Make_Mercedes-Benz       0.298102
Make_Mercury             0.062864
Make_Mitsubishi         -0.104095
Make_Nissan             -0.037484
Make_Oldsmobile         -0.049414
Make_Pontiac            -0.027947
Make_Porsche             0.134934
Make_Saab               -0.075839
Make_Saturn             -0.081171
Make_Scion              -0.040299
Make_Subaru             -0.095524
Make_Suzuki             -0.081171
Make_Toyota             -0.090738
Make_Volkswagen         -0.112087
Make_Volvo              -0.099891
Type_Hybrid             -0.049414
Type_SUV                -0.237484
Type_Sedan              -0.135369
Type_Sports              0.393055
Type_Truck               0.135531
Type_Wagon              -0.014875
Origin_Asia             -0.172924
Origin_Europe            0.217251
Origin_USA              -0.031306
DriveTrain_All          -0.307756
DriveTrain_Front        -0.622102
DriveTrain_Rear          1.000000

[50 rows x 50 columns]
```

By eye, the highest correlation value is "Drivetrain front" with -0.622102

## 3.3 Create the correlation matrix again by removing all car models that were classified as an outlier. (You can use .query('Model in["MDX","TSX 4dr"]'). There is some variation in correlation

### Correlation matrix after filtering outliers

```
from scipy.odr import Model

numericFilteredColumns = df.query('Model in["MDX","TSX
4dr"]').select_dtypes(include=['float64', 'int64'])
new = numericFilteredColumns.corr()
numericFilteredColumns.corr()
```

| | Invoice | EngineSize | Cylinders | Horsepower | MPG_City | \ |
|---|---|---|---|---|---|---|
| Invoice | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |
| EngineSize | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |
| Cylinders | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |
| Horsepower | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |
| MPG_City | -1.0 | -1.0 | -1.0 | -1.0 | 1.0 | |
| MPG_Highway | -1.0 | -1.0 | -1.0 | -1.0 | 1.0 | |
| Weight | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |
| Wheelbase | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |
| Length | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | |

| | MPG_Highway | Weight | Wheelbase | Length |
|---|---|---|---|---|
| Invoice | -1.0 | 1.0 | 1.0 | 1.0 |
| EngineSize | -1.0 | 1.0 | 1.0 | 1.0 |
| Cylinders | -1.0 | 1.0 | 1.0 | 1.0 |
| Horsepower | -1.0 | 1.0 | 1.0 | 1.0 |
| MPG_City | 1.0 | -1.0 | -1.0 | -1.0 |
| MPG_Highway | 1.0 | -1.0 | -1.0 | -1.0 |
| Weight | -1.0 | 1.0 | 1.0 | 1.0 |
| Wheelbase | -1.0 | 1.0 | 1.0 | 1.0 |
| Length | -1.0 | 1.0 | 1.0 | 1.0 |

### Correlation matrix before filtering outliers

```
numericVariables.corr()
```

| | Invoice | EngineSize | Cylinders | Horsepower | MPG_City | \ |
|---|---|---|---|---|---|---|
| Invoice | 1.000000 | 0.564498 | 0.645226 | 0.823746 | -0.470442 | |
| EngineSize | 0.564498 | 1.000000 | 0.908002 | 0.787435 | -0.709471 | |
| Cylinders | 0.645226 | 0.908002 | 1.000000 | 0.810341 | -0.684402 | |
| Horsepower | 0.823746 | 0.787435 | 0.810341 | 1.000000 | -0.676699 | |
| MPG_City | -0.470442 | -0.709471 | -0.684402 | -0.676699 | 1.000000 | |
| MPG_Highway | -0.434585 | -0.717302 | -0.676100 | -0.647195 | 0.941021 | |
| Weight | 0.442332 | 0.807867 | 0.742209 | 0.630796 | -0.737966 | |

```
Wheelbase    0.148328    0.636517   0.546730    0.387398 -0.507284
Length       0.166586    0.637448   0.547783    0.381554 -0.501526

             MPG_Highway    Weight  Wheelbase    Length
Invoice        -0.434585  0.442332   0.148328  0.166586
EngineSize     -0.717302  0.807867   0.636517  0.637448
Cylinders      -0.676100  0.742209   0.546730  0.547783
Horsepower     -0.647195  0.630796   0.387398  0.381554
MPG_City        0.941021 -0.737966  -0.507284 -0.501526
MPG_Highway     1.000000 -0.790989  -0.524661 -0.466092
Weight         -0.790989  1.000000   0.760703  0.690021
Wheelbase      -0.524661  0.760703   1.000000  0.889195
Length         -0.466092  0.690021   0.889195  1.000000
```