Module 5 Assignment

Rudy Fasano

11 / 30 / 2020

Exercise 2

2.1

Importing necessary packages for analyses:

In [2]:
```
import pymysql.cursors ### importing libraries
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
%matplotlib inline
```

Establishing connection to MySQL server in order to access data:

In [3]:
```
connection = pymysql.connect(host='localhost', #### connecting to MySQL server and importing data
                             user='root',
                             password='Clockwork2911!!',
                             db='auto',
                             charset='utf8mb4',
                             cursorclass=pymysql.cursors.DictCursor)
print(connection)
```

```
<pymysql.connections.Connection object at 0x0000029D78F005E0>
```

In [4]:
```
with connection.cursor() as cursor:
    df_head = "select * from mpg;"
    cursor.execute(df_head)
    ##
    df = pd.DataFrame(cursor.fetchall())
    connection.close()
```

Exploring data:

In [ ]:
```
df.head()
```

Exploring data to identify any null values, none were present:

In [110…
```
nulls=pd.isnull(df["car name"]) #### identifying NaN values per column
df[nulls]
```

Out[110…

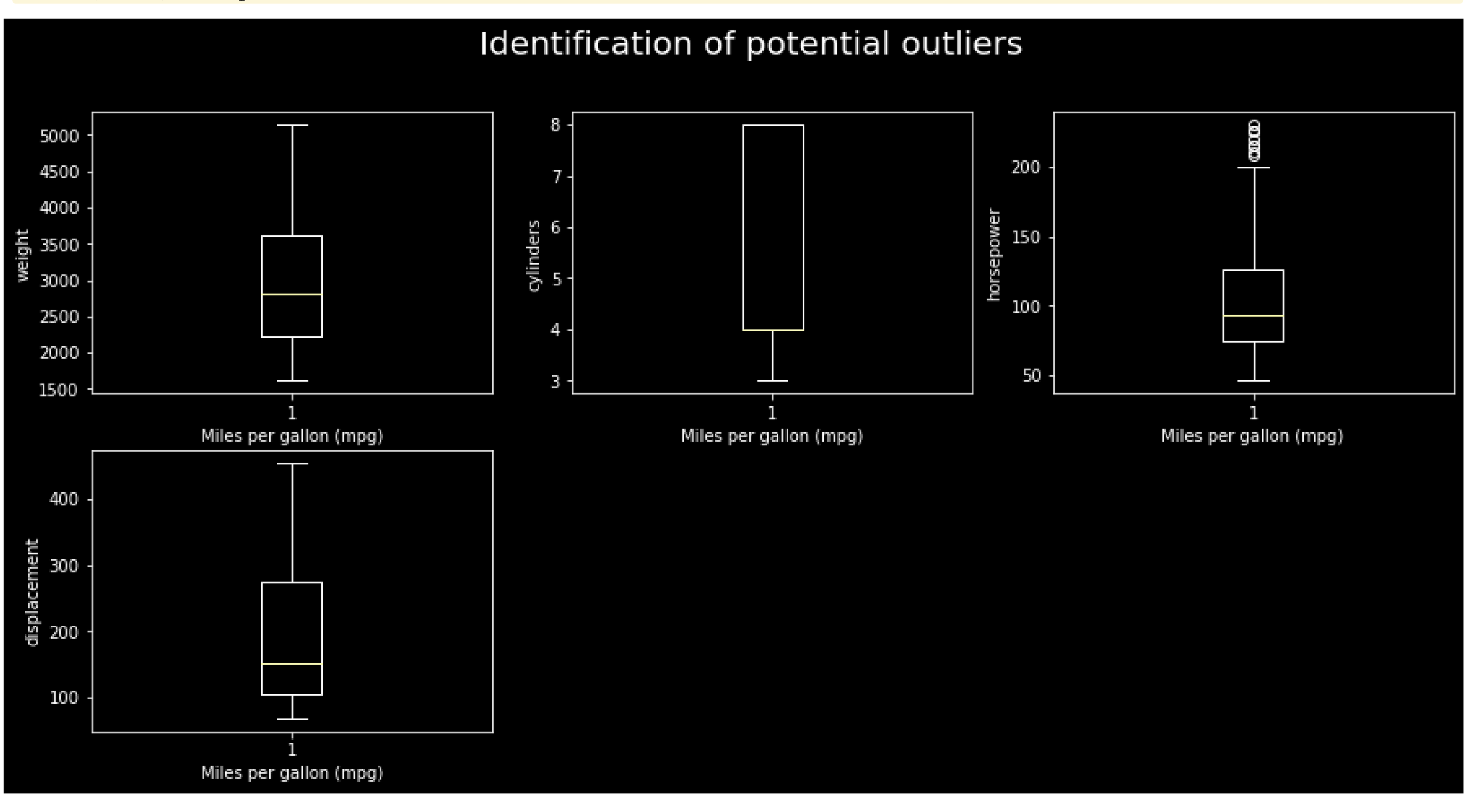
mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name

Exploring data to identify any potential outliers.

In [7]:
```
fig = plt.figure(figsize = (15,7))       ######### identifying potential outliers with boxplots
fig.suptitle('Identification of potential outliers',
            fontsize=20)

ax1 = fig.add_subplot(231)
ax1.boxplot(x="weight", data=df)
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('weight')
##
ax2 = fig.add_subplot(232)
ax2.boxplot(x="cylinders", data=df)
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('cylinders')
##
ax3 = fig.add_subplot(233)
ax3.boxplot(x="horsepower", data=df)
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('horsepower')
##
ax4 = fig.add_subplot(234)
ax4.boxplot(x="displacement", data=df)
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('displacement')
##
```

Out[7]: `Text(0, 0.5, 'displacement')`



No potential outliers were identified except for in the variable "horsepower" as noted above. Nevertheless, the potential values were not identified to be incorrect or innacurate data due to being high-performance cars.

In [24]:
```
q1, q3 = np.percentile(df["horsepower"], [25, 75])
iqr = q3 - q1
crim = df["horsepower"]

lower_b = q1 - (1.5 * iqr)
upper_b = q3 + (1.5 * iqr)


#############
sub = (df[df['horsepower'] > (1.5*iqr + q3)])

sub.head()
```

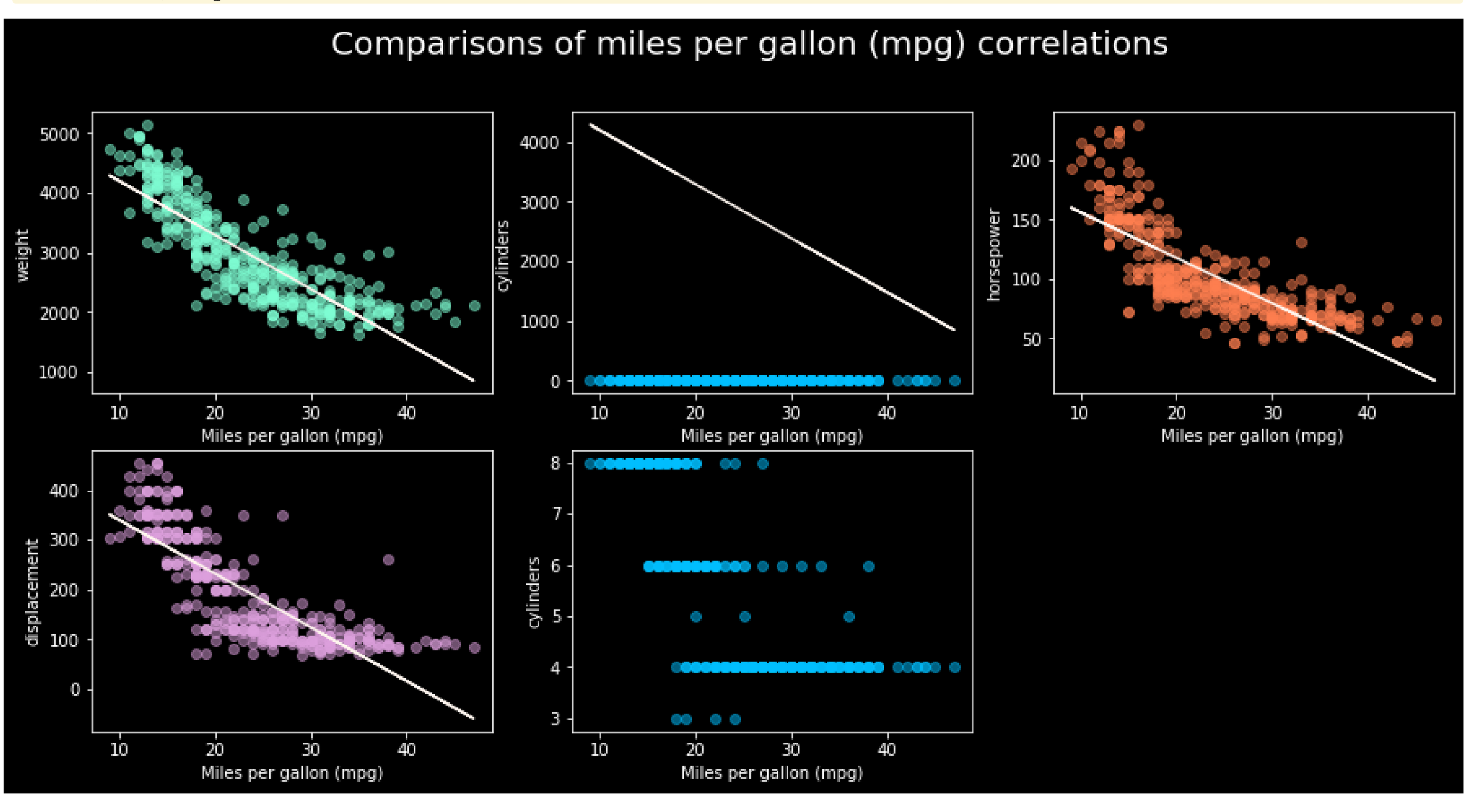
Out[24]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
6	14	8	454	220	4354	9	70	1	chevrolet impala
7	14	8	440	215	4312	8.5	70	1	plymouth fury iii
8	14	8	455	225	4425	10	70	1	pontiac catalina
13	14	8	455	225	3086	10	70	1	buick estate wagon (sw)
25	10	8	360	215	4615	14	70	1	ford f250

Regression models created to identify any trends or relationships:

In [25]:
```
fig = plt.figure(figsize = (15,7))
fig.suptitle('Comparisons of miles per gallon (mpg) correlations',
            fontsize=20)
ax1 = fig.add_subplot(231)
ax1.scatter(df['mpg'],df['weight'],alpha=0.5, c='aquamarine')
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('weight')
x=df['mpg']  ##### calculating for regression line
y=df['weight']
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b, c='seashell')
##
ax2 = fig.add_subplot(232)
ax2.scatter(df['mpg'],df['cylinders'],alpha=0.5, c='deepskyblue')
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('cylinders')
x1=df['mpg']
y1=df['cylinders']
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b, c='seashell')
##
ax3 = fig.add_subplot(233)
ax3.scatter(df['mpg'],df['horsepower'],alpha=0.5, c='coral')
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('horsepower')
x=df['mpg']
y=df['horsepower']
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b, c='seashell')
###
ax4 = fig.add_subplot(234)
ax4.scatter(df['mpg'],df['displacement'],alpha=0.5, c='plum')
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('displacement')
x=df['mpg']
y=df['displacement']
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b, c='seashell')
##
ax5 = fig.add_subplot(235)
ax5.scatter(df['mpg'],df['cylinders'],alpha=0.5, c='deepskyblue')
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('cylinders')
```

Out[25]: `Text(0, 0.5, 'cylinders')`



In [9]:
```
mpgweight = smf.ols('mpg~weight', data=df) # mpg ~ weight regression model
mpgweight = mpgweight.fit()
mpgweight.params
##
mpgcylinders = smf.ols('mpg~cylinders', data=df) # mpg ~ cylinder regression model
mpgcylinders = mpgcylinders.fit()
mpgcylinders.params
##
mpgdis = smf.ols('mpg~displacement', data=df)
mpgdis = mpgdis.fit()
mpgdis.params
##
mpghp = smf.ols('mpg~horsepower', data=df)
mpghp = mpghp.fit()
mpghp.params



print(mpgweight.params,
      '\n', mpgcylinders.params,
      '\n', mpgdis.params,
      '\n', mpghp.params
     )
```

```
Intercept    46.228738
weight       -0.007636
dtype: float64
 Intercept    42.928696
cylinders    -3.552004
dtype: float64
 Intercept       35.147741
displacement    -0.059952
dtype: float64
 Intercept     39.955805
horsepower    -0.157591
dtype: float64
```

The regression model consists of the dependent variable (y) being miles per gallon (mpg), was investigated with the independent variables (x) being weight, cylidners, displacement and horsepower. The visualizations indicate very similar negative relationships with mpg. The regression line equations are as follows: Mpg = 46.229 - 0.007 (x) Weight, Mpg = 42.929 - 3.552 (x) Cylinders, Mpg = 35.148 - 0.060 (x) Displacement, Mpg = 39.956 - 0.158 (x) Horsepower

With the independent variable experiencing these negative relationships, it can be inferred that as the dependent variable increases, the independent variable will duely decrease. Heavier vehicles with larger engines, heavier bodies as well as higher volume of piston displacement will result in having less miles per gallon.