Instructions: During the class session, you will be divided in to groups for this exercise. This exercise will not be graded but will account for extra credit.

Use the provided dataset to answer the following questions:

1. Does the dataset have any missing information?
2. Does the dataset have any outliers?
3. Which of the variables within the dataset are not normally distributed?
4. Show the descriptive statistics of all the continuous variables within the dataset.

Group Members

1. Remmy Bisimbeko - B26099 - J24M19/011 My GitHub - https://github.com/RemmyBisimbeko/Data-Science

Recommended Libraries

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Read Excel File

In [ ]:
```
df = pd.read_excel("Data Sets/Assignment1_Cars-1.xlsx")
```

Display first five rows of data

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

Out[ ]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

Display bottom five rows (optional)

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

Out[ ]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
58	Acura	21.4	4	121.0	109	4.11	2.78	18.6	1	1	4	2
59	Toyota Tacoma	40.6	2	75.7	52	4.93	17.02	0.0	0	3	2	1
60	GMC Sierra	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2
61	Nissan Xtrail	17.3	8	275.8	180	3.07	3.73	17.6	0	0	3	3
62	Merc C-class	15.2	8	275.8	180	3.07	3.78	18.0	0	0	3	3

What is the Shape of our Data frame? (rows, cols)

In [ ]:
```
df.shape
```

Out[ ]:
```
(63, 12)
```

Column Names

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['model', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs',
'am',
       'gear', 'carb'],
      dtype='object')
```

Data types of each column

In [ ]:
```
df.dtypes
```

Out[ ]:
```
model     object
mpg      float64
cyl        int64
disp     float64
hp         int64
drat     float64
wt       float64
qsec     float64
vs         int64
am         int64
gear       int64
carb       int64
dtype: object
```

More information on the Data Frame

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   model   63 non-null     object 
 1   mpg     63 non-null     float64
 2   cyl     63 non-null     int64  
 3   disp    63 non-null     float64
 4   hp      63 non-null     int64  
 5   drat    63 non-null     float64
 6   wt      63 non-null     float64
 7   qsec    63 non-null     float64
 8   vs      63 non-null     int64  
 9   am      63 non-null     int64  
 10  gear    63 non-null     int64  
 11  carb    63 non-null     int64  
dtypes: float64(5), int64(6), object(1)
memory usage: 6.0+ KB
```

Describe the Data

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

Out[ ]:

	mpg	cyl	disp	hp	drat	wt	
count	63.000000	63.000000	63.000000	63.000000	63.000000	63.000000	63.
mean	20.207937	6.190476	234.887302	149.285714	3.594286	3.758873	17
std	6.755172	1.924856	122.607423	68.567703	0.558678	2.590846	3
min	10.400000	2.000000	71.100000	52.000000	2.760000	1.513000	0.
25%	15.200000	4.000000	130.900000	95.000000	3.075000	2.780000	16.
50%	18.700000	6.000000	225.000000	150.000000	3.690000	3.440000	17.
75%	22.800000	8.000000	334.000000	180.000000	3.920000	3.780000	18.
max	40.600000	8.000000	472.000000	335.000000	4.930000	17.020000	22.

My Observations are as follows: -We have 63 Vehicles -Highest Miles per gallon for all cars is 40.6

QN 1. Does the dataset have any missing information?

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

Out[ ]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
58	False	False	False	False	False	False	False	False	False	False	False	False
59	False	False	False	False	False	False	False	False	False	False	False	False
60	False	False	False	False	False	False	False	False	False	False	False	False
61	False	False	False	False	False	False	False	False	False	False	False	False
62	False	False	False	False	False	False	False	False	False	False	False	False

63 rows × 12 columns

How many are they?

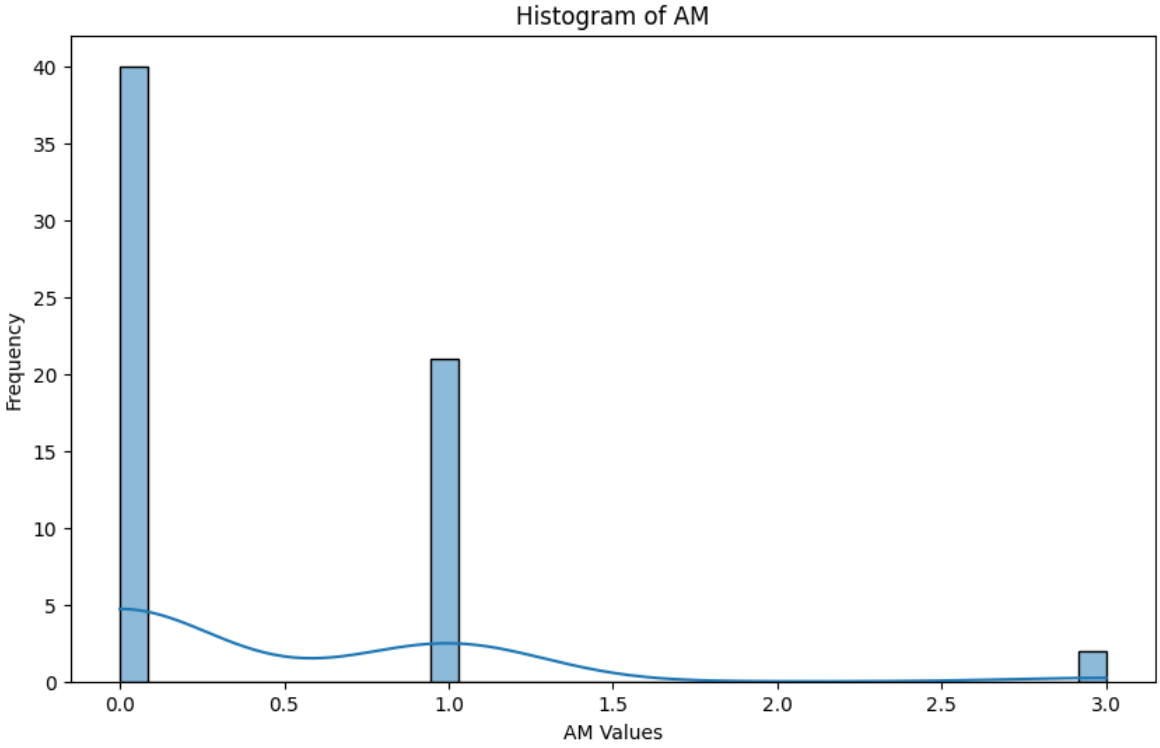
In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
model    0
mpg      0
cyl      0
disp     0
hp       0
drat     0
wt       0
qsec     0
vs       0
am       0
gear     0
carb     0
dtype: int64
```

Data Set appears to have no missing values!

QN 2. Does the dataset have any outliers?

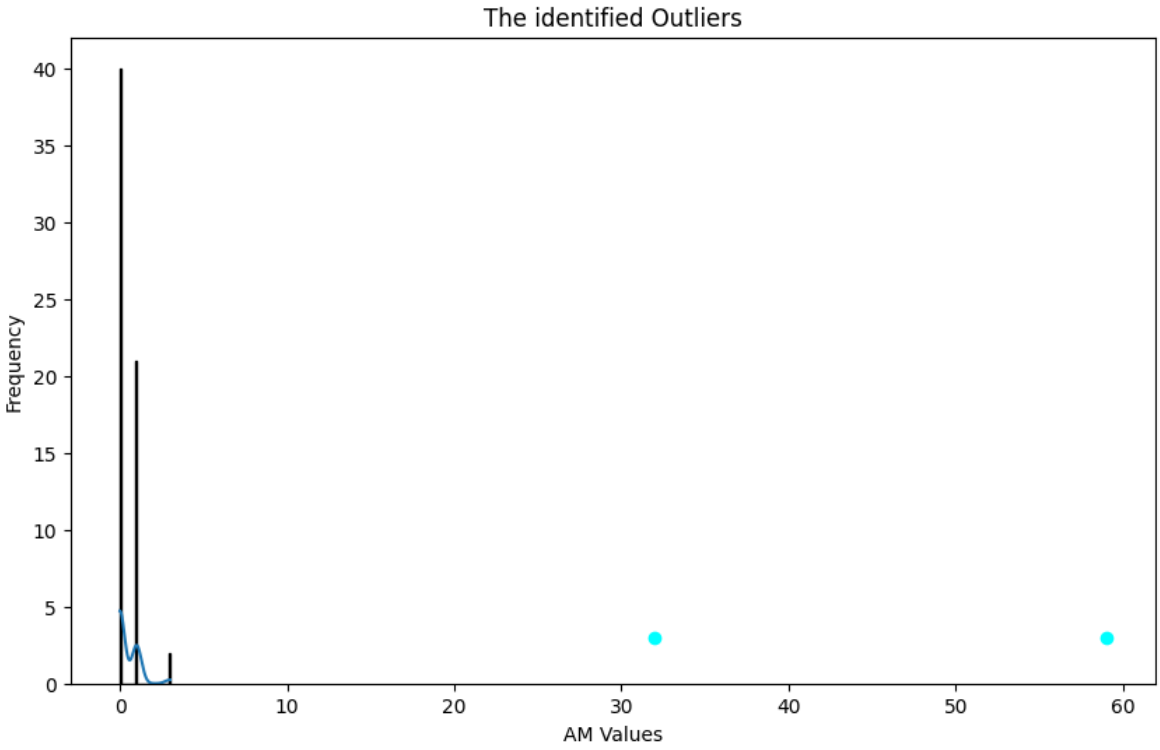
In [ ]:
```
# Using a Histogram to Visualise the Data
import seaborn as sns

plt.figure(figsize=(10,6))
sns.histplot(df['am'], bins=35, kde=True)
plt.title('Histogram of AM')
plt.xlabel('AM Values')
plt.ylabel('Frequency')
plt.show()
```



In [ ]:
```
# Detect the Outliers Using the Z-Score Approach
import scipy.stats as stats

zscores = stats.zscore(df['am'])
threshold = 3
outliers = np.where(np.abs(zscores) > threshold)[0]

print("Outliers:", outliers)
```

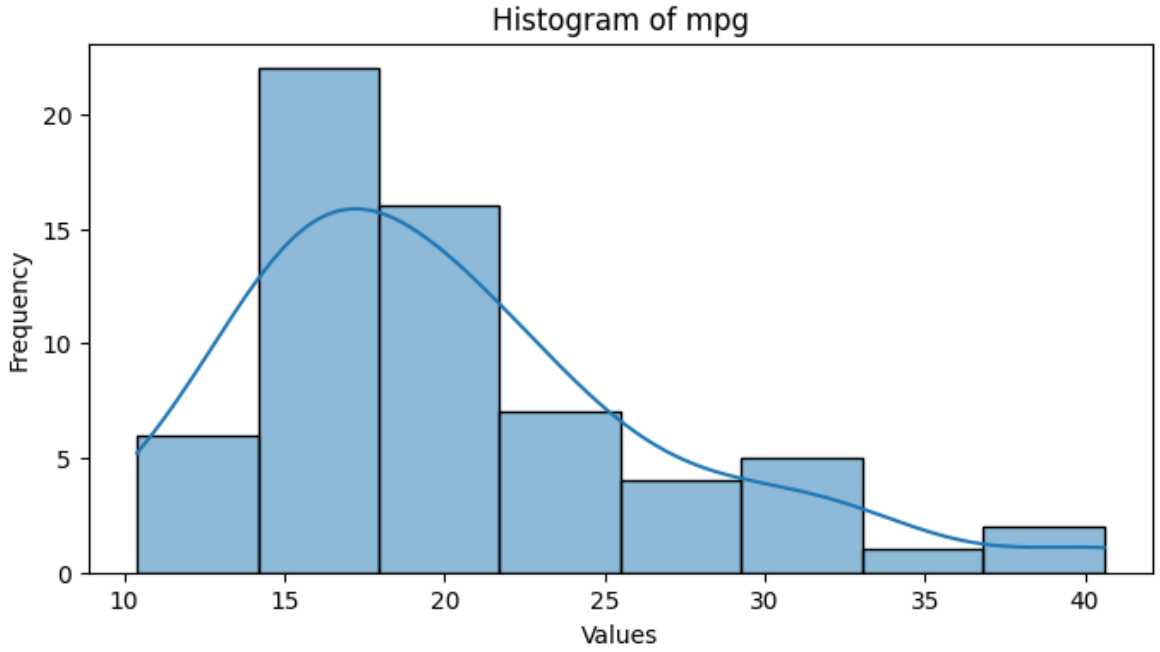
```
Outliers: [32 59]
```

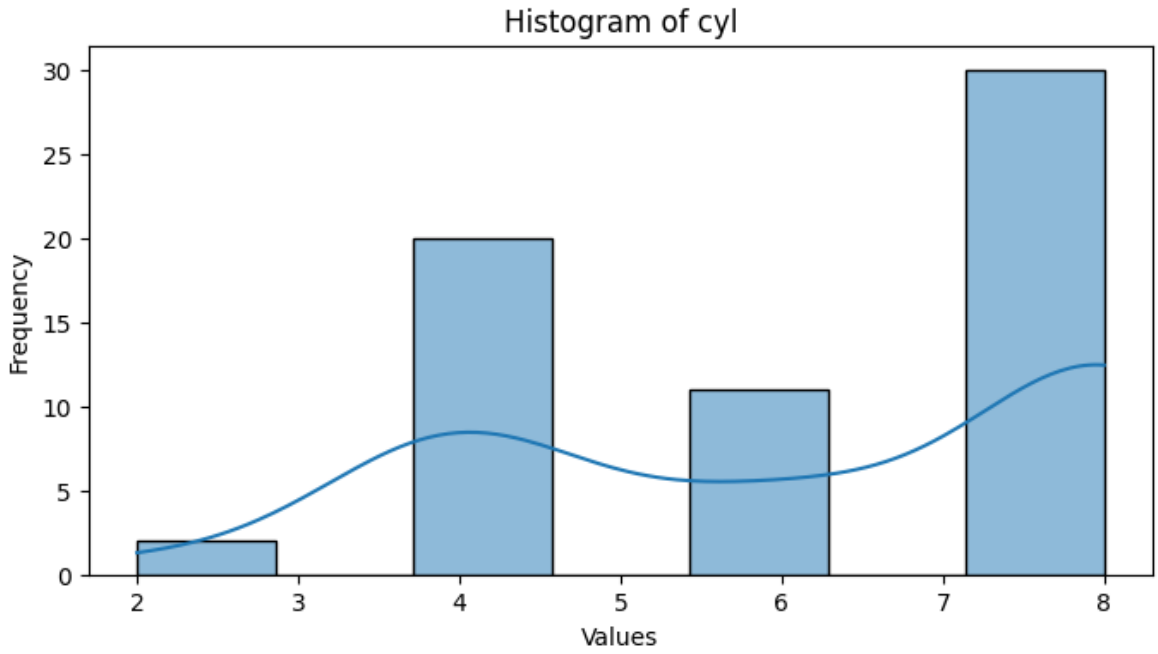
In [ ]:
```
# Visualise the Outliers
plt.figure(figsize=(10, 6))
sns.histplot(df['am'], bins=35, kde=True)
plt.scatter(outliers, df['am'].iloc[outliers], color='cyan', label='Outli
plt.title('The Identified Outliers')
plt.xlabel('AM Values')
plt.ylabel('Frequency')
plt.show()
```

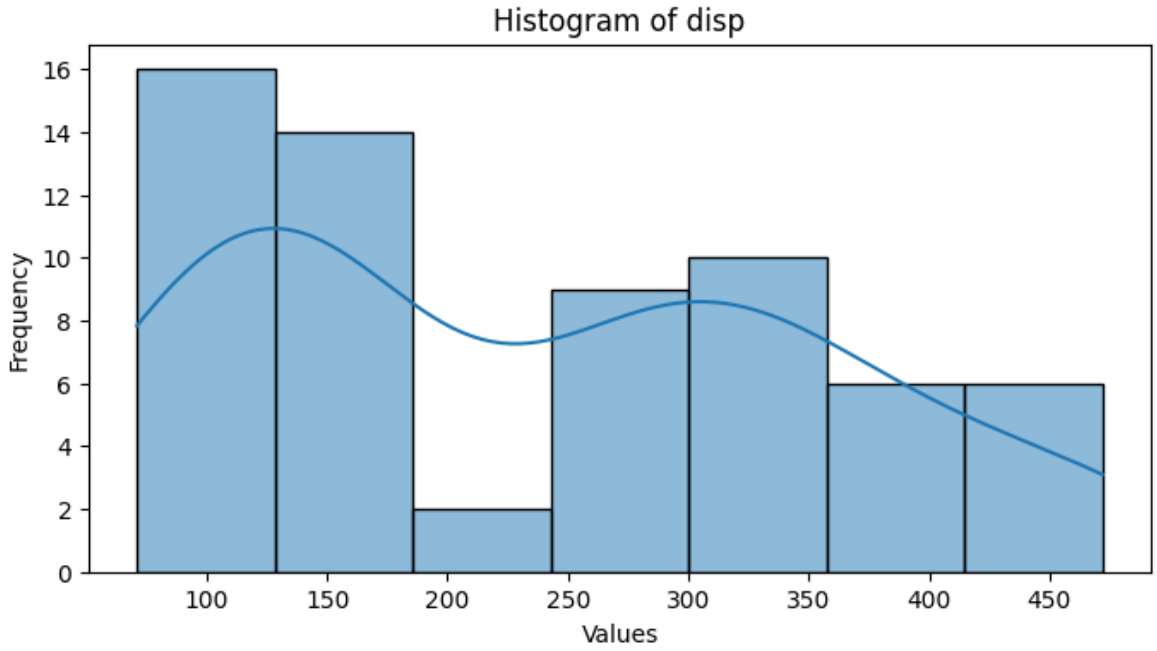


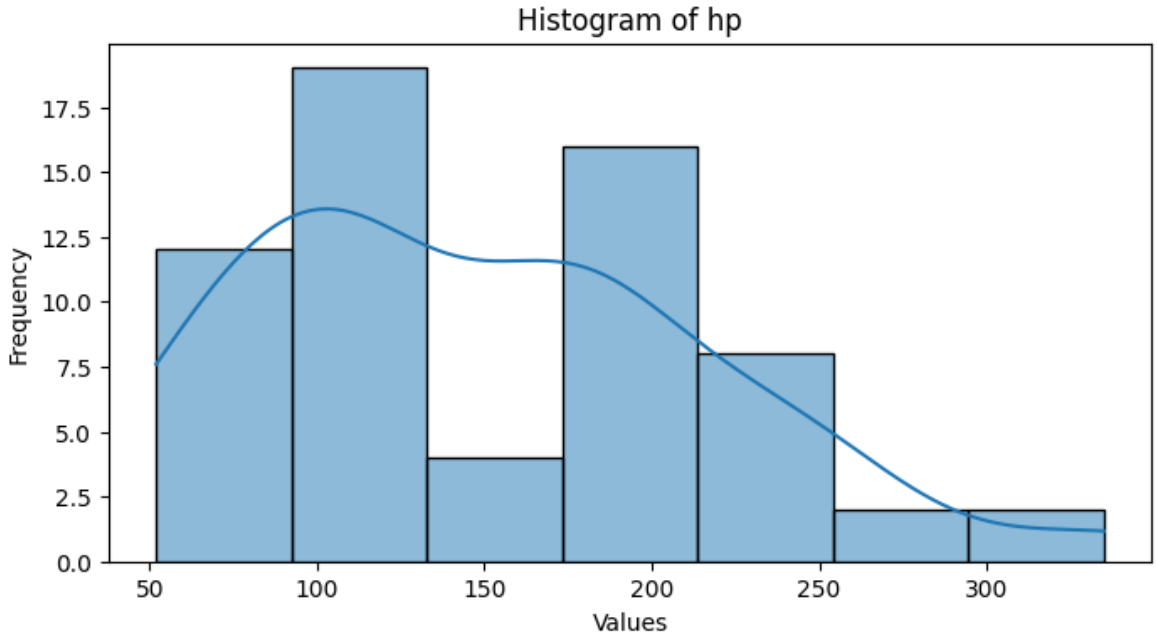
QN 3. Which of the variables within the dataset are not normally distributed?

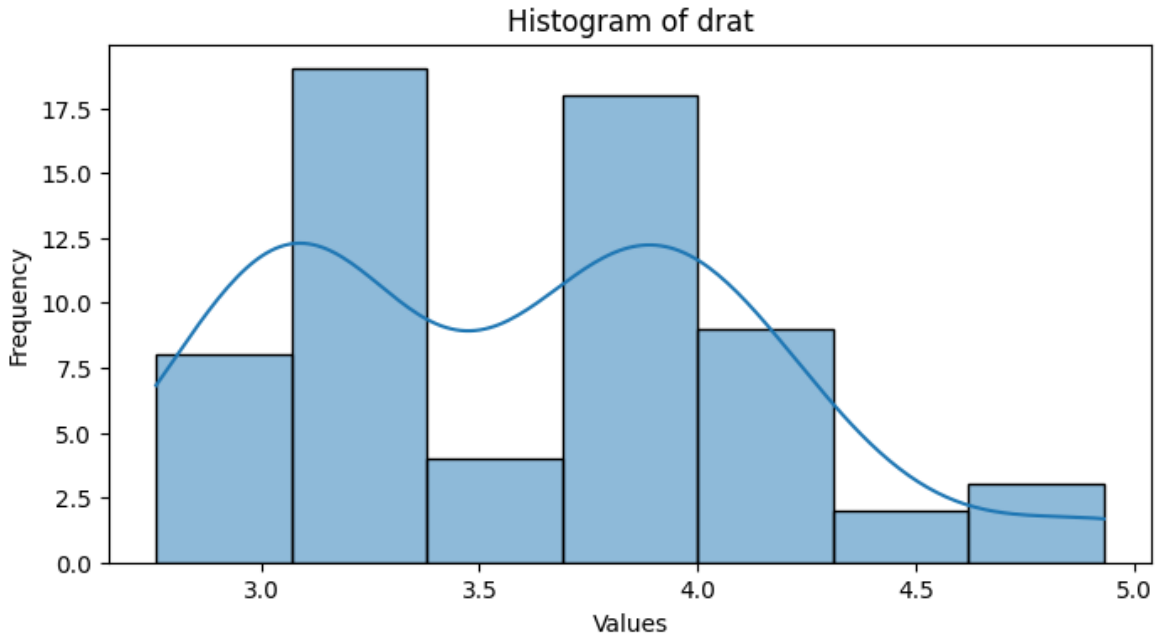
In [ ]:
```
# Remove non-numeric columns
numeric_columns = df.select_dtypes(include=[np.number]).columns
df_numeric = df[numeric_columns]

# Visual inspection of histograms
for column in df_numeric.columns:
    plt.figure(figsize=(8, 4))
    sns.histplot(df_numeric[column], kde=True)
    plt.title(f'Histogram of {column}')
    plt.xlabel('Values')
    plt.ylabel('Frequency')
    plt.show()
```

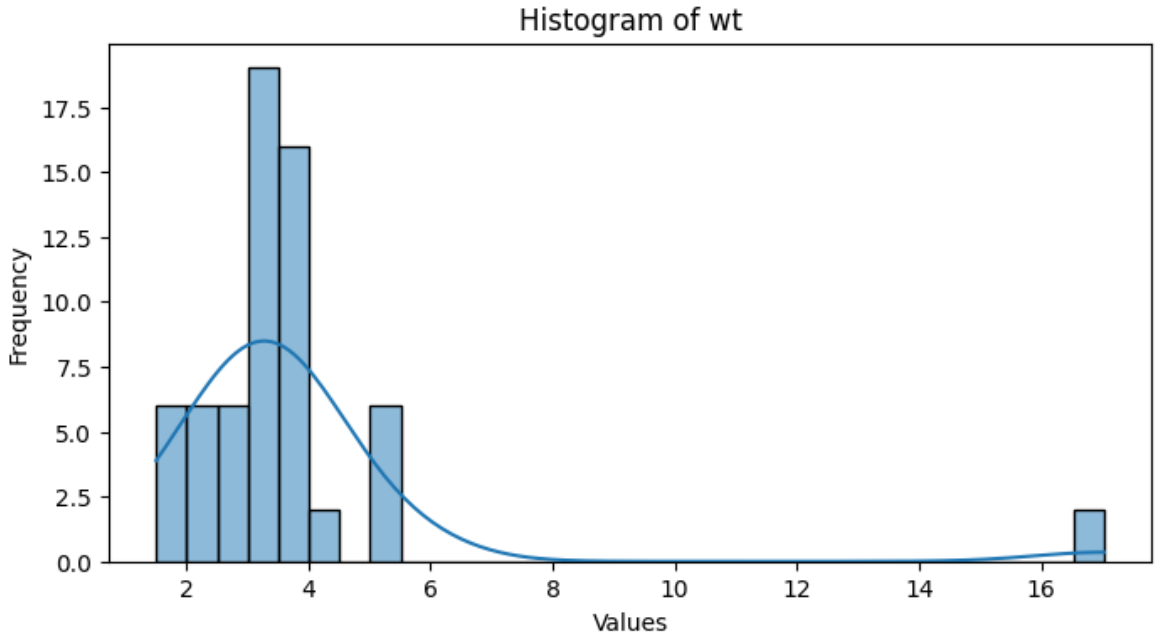


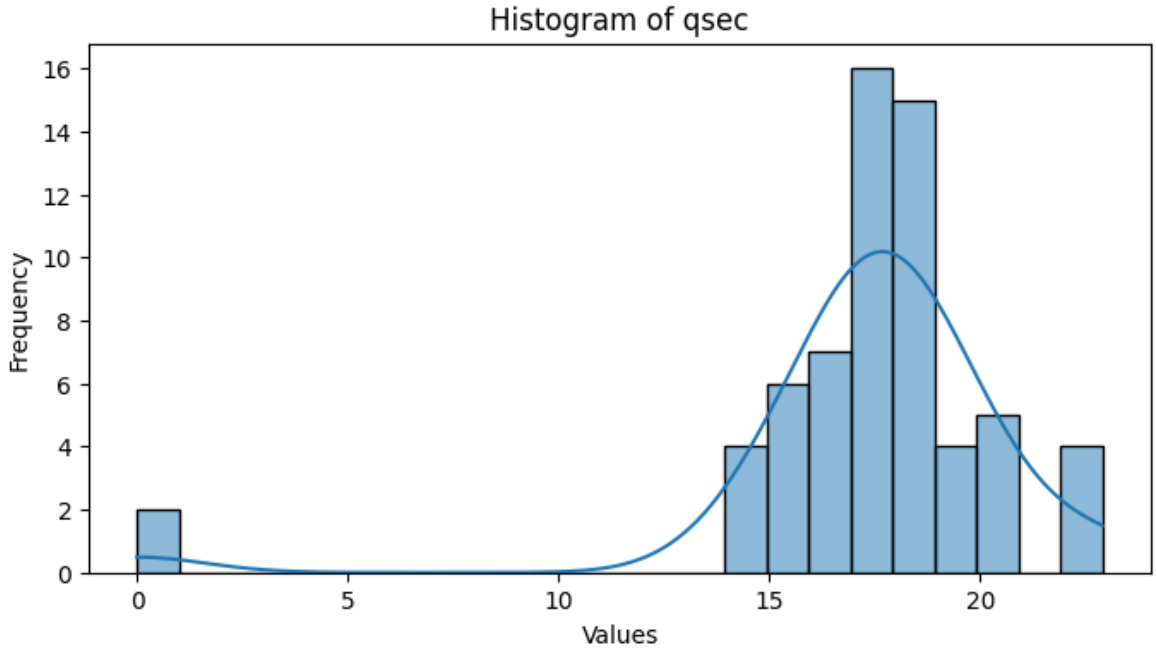

Histogram of cyl
Frequency
Values
0
5
10
15
20
25
30
2
3
4
5
6
7
8

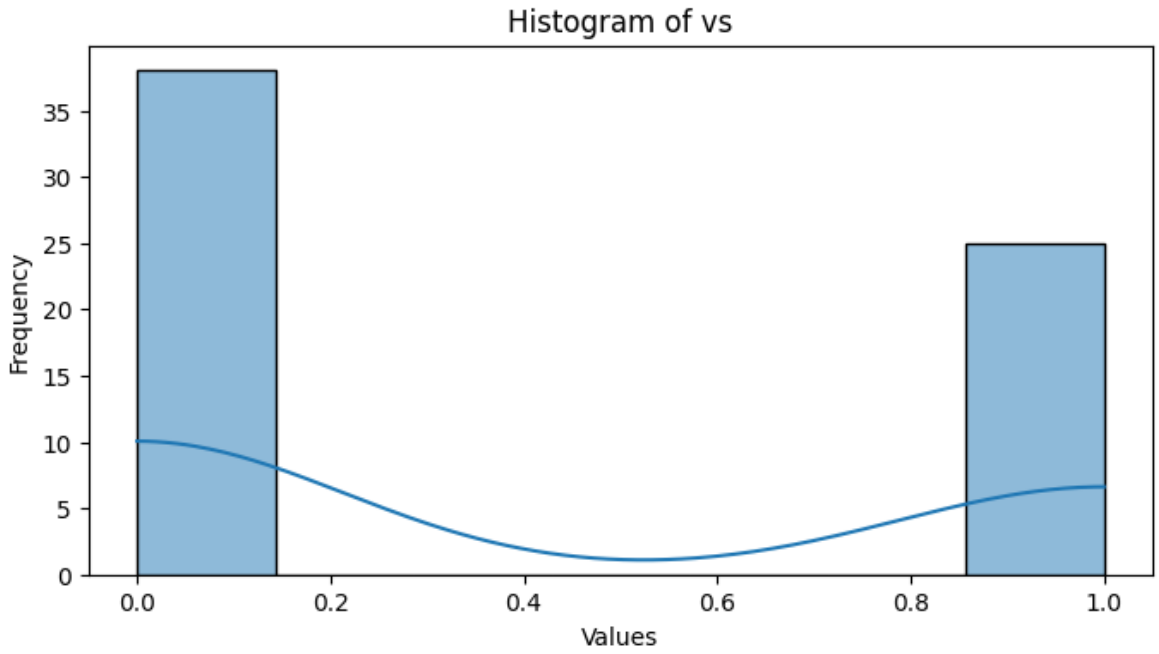

Histogram of disp
Frequency
Values
0
2
4
6
8
10
12
14
16
100
150
200
250
300
350
400
450

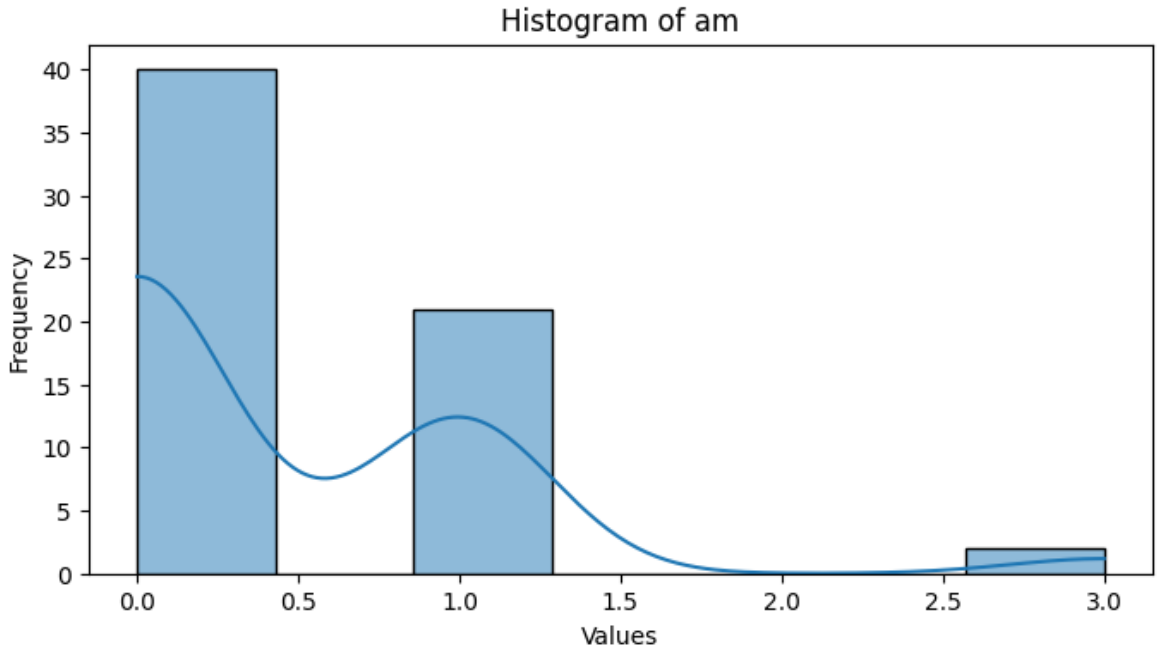

Histogram of hp
Frequency
Values
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
50
100
150
200
250
300

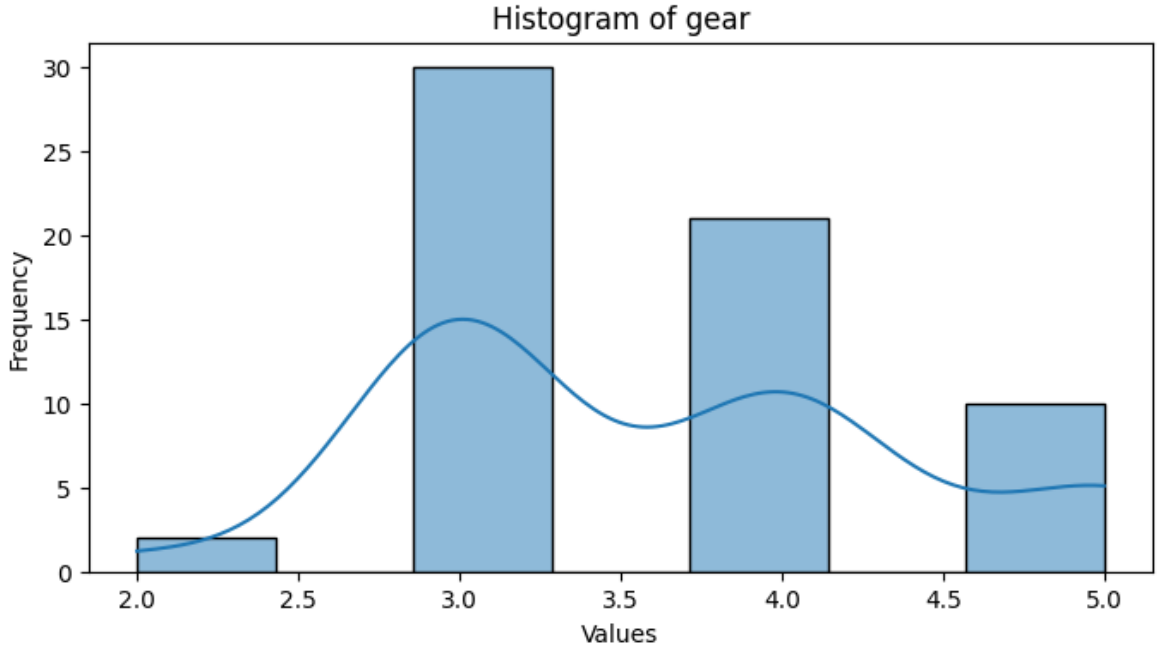

Histogram of drat
Frequency
Values
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
3.0
3.5
4.0
4.5
5.0

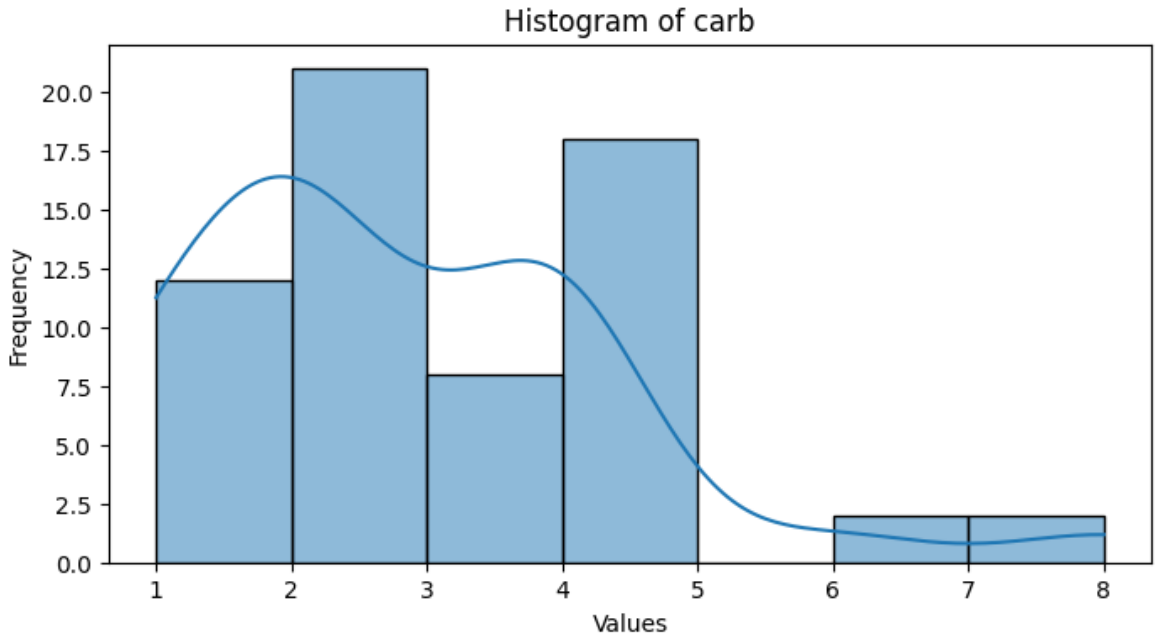

Histogram of wt
Frequency
Values
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
2
4
6
8
10
12
14
16


Histogram of qsec
Frequency
Values
0
2
4
6
8
10
12
14
16
0
5
10
15
20


Histogram of vs
Frequency
Values
0
5
10
15
20
25
30
35
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of am
Frequency
Values
0
5
10
15
20
25
30
35
40
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Histogram of gear
Frequency
Values
0
5
10
15
20
25
30
2.0
2.5
3.0
3.5
4.0
4.5
5.0



In [ ]:
```
# Using the Shapiro-Wilk test for normality
from scipy.stats import shapiro

# Shapiro-Wilk test
non_normal_variables = []
for column in df_numeric.columns:
    stat, p = shapiro(df_numeric[column])
    alpha = 0.05
    if p < alpha:
        non_normal_variables.append(column)
        print(f'{column} is not normally distributed (p-value={p})')
    else:
        print(f'{column} is normally distributed (p-value={p})')

print("\nVariables not normally distributed:", non_normal_variables)
```

```
mpg is not normally distributed (p-value=0.00021498681772254538)
cyl is not normally distributed (p-value=1.9259465725125302e-08)
disp is not normally distributed (p-value=0.0005896648328323517)
hp is not normally distributed (p-value=0.004189709579620521)
drat is not normally distributed (p-value=0.001560451108193203)
wt is not normally distributed (p-value=1.5314964146613204e-13)
qsec is not normally distributed (p-value=9.89061646923088e-11)
vs is not normally distributed (p-value=1.7410261772405094e-11)
am is not normally distributed (p-value=1.795177884425708e-11)
gear is not normally distributed (p-value=3.553517270238008e-07)
carb is not normally distributed (p-value=1.5340614007820725e-06)

Variables not normally distributed: ['mpg', 'cyl', 'disp', 'hp', 'drat',
'wt', 'qsec', 'vs', 'am', 'gear', 'carb']
```

QN 4. Show the descriptive statistics of all the continuous variables within the dataset.

In [ ]:
```
# Display descriptive statistics of continuous variables
continuous_variables = df.select_dtypes(include=['number'])
print(continuous_variables.describe())
```

	mpg	cyl	disp	hp	drat	wt
count	63.000000	63.000000	63.000000	63.000000	63.000000	63.000000
mean	20.207937	6.190476	234.887302	149.285714	3.594286	3.758873
std	6.755172	1.924856	122.607423	68.567703	0.558678	2.590846
min	10.400000	2.000000	71.100000	52.000000	2.760000	1.513000
25%	15.200000	4.000000	130.900000	95.000000	3.075000	2.780000
50%	18.700000	6.000000	225.000000	150.000000	3.690000	3.440000
75%	22.800000	8.000000	334.000000	180.000000	3.920000	3.780000
max	40.600000	8.000000	472.000000	335.000000	4.930000	17.020000

	qsec	vs	am	gear	carb
count	63.00000	63.000000	63.000000	63.000000	63.000000
mean	17.36127	0.396825	0.428571	3.619048	2.825397
std	3.71618	0.493169	0.665129	0.791662	1.571312
min	0.00000	0.000000	0.000000	2.000000	1.000000
25%	16.87000	0.000000	0.000000	3.000000	2.000000
50%	17.60000	0.000000	0.000000	3.000000	2.000000
75%	18.90000	1.000000	1.000000	4.000000	4.000000
max	22.90000	1.000000	3.000000	5.000000	8.000000

Sources:

https://www.kaggle.com/code/rtatman/data-cleaning-challenge-outliers
https://github.com/ipython/ipython https://etna-docs.netlify.app/tutorials/outliers.html
https://pythonguides.com/scipy-stats-zscore/ https://www.statology.org/z-score-python/ https://www.geeksforgeeks.org/scipy-stats-zscore-function-python/
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html

My Git Hub Repo https://github.com/RemmyBisimbeko/Data-Science
https://www.kaggle.com/remmybisimbeko