Task 1:

a)

TASK 1. GAINING A BASIC UNDERSTANDING OF PANDAS DATAFRAME

Storing tabular data as pandas dataframe:

(a) Data preprocessing is one of the steps in machine learning. The pandas library in python is suitable to deal with tabular data. Create a variable 'emissions' and assign to it the following data (Table 1) as padas DataFrame. Create an excel file 'emissions_from_pandas.xlsx' from the'emissions' variable using python.

In [1]:
```
import pandas as pd
from pandas import DataFrame


# Creates Table
emissions = DataFrame()
emissions["Low Altitude"] = [1.50, 1.48, 2.98, 1.4, 3.12, 0.25, 6.73, 5.3, 9.3, 6.96, 7.21, 0.87, 1.06, 7.39, 1.37]
emissions["High Altitude"] = [7.59,2.06,8.86,8.67,5.61,6.28,4.04,4.40,9.52,1.50,6.07,17.11,3.57,2.68,6.46]
emissions.to_excel("emissions_from_pandas.xlsx")
```

b)

In [2]:
```
display(emissions.head())
display(emissions.iloc[0,0])
display(emissions.iloc[1,1])
display(emissions.iloc[0:2,0:2])
display(emissions.iloc[2:4,:])
```

	Low Altitude	High Altitude
0	1.50	7.59
1	1.48	2.06
2	2.98	8.86
3	1.40	8.67
4	3.12	5.61

```
1.5
2.06
```

	Low Altitude	High Altitude
0	1.50	7.59
1	1.48	2.06

	Low Altitude	High Altitude
2	2.98	8.86
3	1.40	8.67

c)

File conversion:

(c) Create an xl file "emissions_from_pandas.xlsx" from the emissions variable using the .to_excel method. Paste the screenshot of the input command.

In [3]:
```
emissions.to_excel("emissions_from_pandas.xlsx")
```

d)

(d) Create an MS Excel file 'emissions_excel.xlsx' containing the data in Table 1 above with thecolumn header and save it on your computer. Create a variable 'emissions_from_excel' from the'emissions_excel.xlsx' file using pd_read function. Show the first five rows using .head(). Paste a screenshot with the input commands used.

In [4]:
```
emissions.to_excel("emissions_excel.xlsx")
emissions_from_excel = pd.read_excel("emissions_excel.xlsx")

emissions_from_excel.head()
```

Out[4]:

	Unnamed: 0	Low Altitude	High Altitude
0	0	1.50	7.59
1	1	1.48	2.06
2	2	2.98	8.86
3	3	1.40	8.67
4	4	3.12	5.61

Task 2:

TASK 2. NUMERICAL AND GRAPHICAL SUMMARY OF DATASETS USING PYTHON AND PANDAS

Report the following statistics for the emissions data (in Task 1) both at low and high altitude: sample size or count, sample mean, sample standard deviation (std), minimum, maximum, median, first and third quartile.

(a) Use python (any library, but pandas will be fast). You can past screenshot of both input and output (output will look like the following.)

In [5]:
```
emissions.describe()
```

Out[5]:

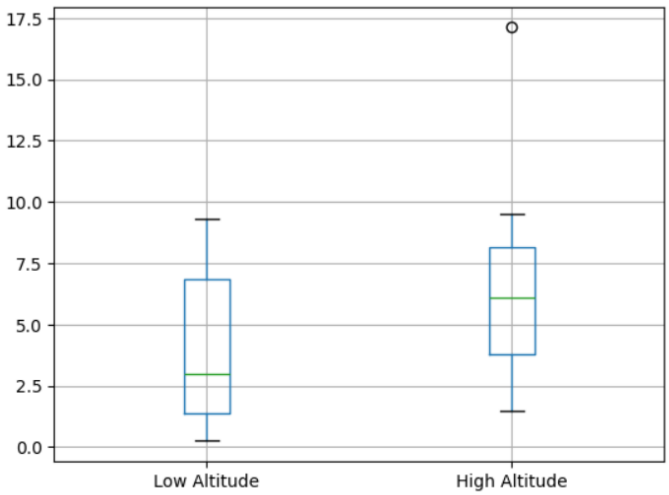
	Low Altitude	High Altitude
count	15.000000	15.000000
mean	3.794667	6.294667
std	3.020801	3.890521
min	0.250000	1.500000
25%	1.385000	3.805000
50%	2.980000	6.070000
75%	6.845000	8.130000
max	9.300000	17.110000

b)

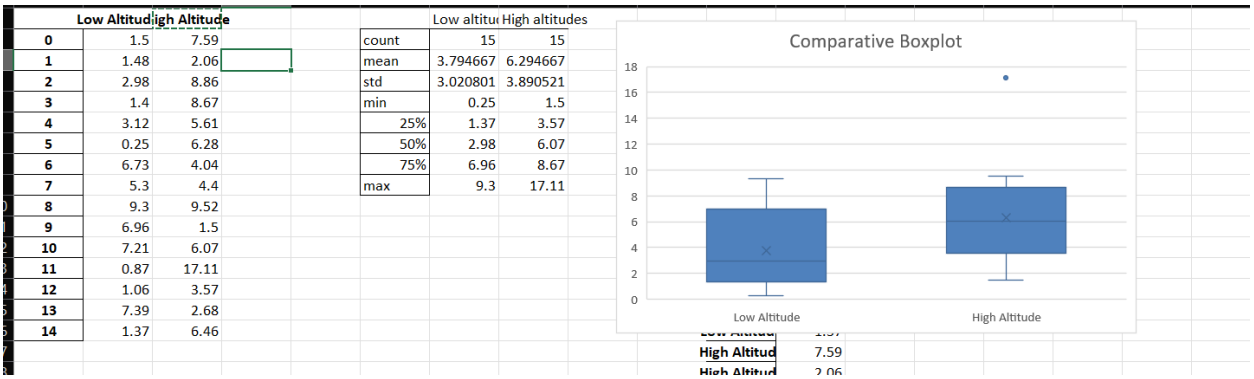
(b) Use the pandas library in python to generate a comparative boxplot of the emissions dataset. Interpret the boxplot (max 50 words)

In [6]:
```
emissions.boxplot()
'''
In general, the low altitude vehicles produce less emissions. However, it is also a much more densely packed dataset, with c
low altitude data is skewed right whereas the high altitude data is fairly symmetric.

'''
```

Out[6]: <Axes: >



c)



	Low Altitud	igh Altitude
0	1.5	7.59
1	1.48	2.06
2	2.98	8.86
3	1.4	8.67
4	3.12	5.61
5	0.25	6.28
6	6.73	4.04
7	5.3	4.4
8	9.3	9.52
9	6.96	1.5
10	7.21	6.07
11	0.87	17.11
12	1.06	3.57
13	7.39	2.68
14	1.37	6.46

	Low altitu	High altitudes
count	15	15
mean	3.794667	6.294667
std	3.020801	3.890521
min	0.25	1.5
25%	1.37	3.57
50%	2.98	6.07
75%	6.96	8.67
max	9.3	17.11

High Altitud	7.59
High Altitud	2.06

Task 3:

TASK 3. IMPORTANCE OF GRAPHS

(a) Define a variable 'dataset1' of type dataframe using the bivariate dataset I given above. Find the summary statistics using dataset1.describe()

In [2]:
```
import seaborn as sns
anscombe = sns.load_dataset('anscombe')
dataset1 = anscombe.iloc[0:11, :]
dataset1.describe()
```

Out[2]:

	x	y
count	11.000000	11.000000
mean	9.000000	7.500909
std	3.316625	2.031568
min	4.000000	4.260000
25%	6.500000	6.315000
50%	9.000000	7.580000
75%	11.500000	8.570000
max	14.000000	10.840000

b)

(b) Define a dataframe 'dataset2' of type dataframe using the bivariate dataset II given above. Find the summary statistics using dataset2.describe().

In [3]:
```
dataset2 = anscombe.iloc[11:22, :]
dataset2.describe()
```

Out[3]:

	x	y
count	11.000000	11.000000
mean	9.000000	7.500909
std	3.316625	2.031657
min	4.000000	3.100000
25%	6.500000	6.695000
50%	9.000000	8.140000
75%	11.500000	8.950000
max	14.000000	9.260000

c)

(c) Do you see any difference between the statistics that summarize the y variables in the two datasets?

In [4]:
```
'''
In the two datasets, the first three rows of data for y were nearly identical. With the only major difference being in th
and max of the two. The min and max for dataset one were both greater than dataset 2.

'''
```

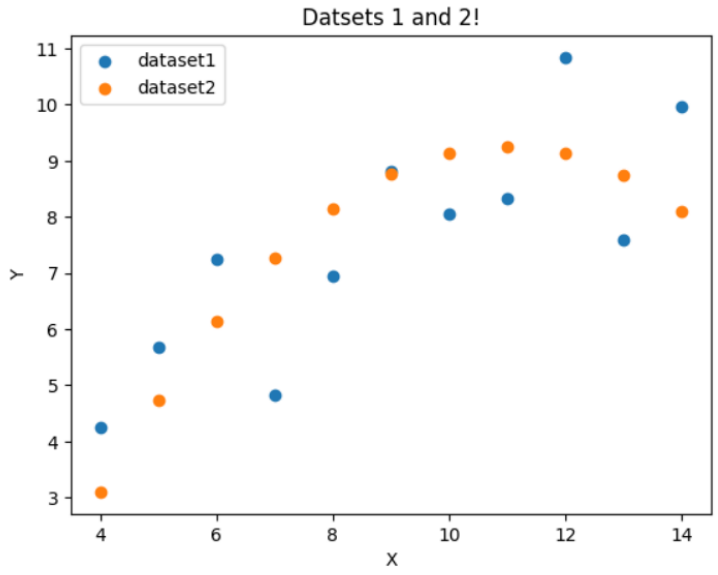
d) Code:

(d) Draw a diagram showing two scatterplots using the same axes to display dataset1 and dataset2 above. Do you see any difference between the two datasets. Comment using less than 50 words.

In [12]:

```
import matplotlib.pyplot as plt
import numpy as np
fig, axis = plt.subplots()

x_data1 = np.array(dataset1.iloc[:,1])
y_data1 = np.array(dataset1.iloc[:,2])

x_data2 = np.array(dataset2.iloc[:,1])
y_data2 = np.array(dataset2.iloc[:,2])

axis.scatter(x_data1, y_data1, label = "dataset1")
axis.scatter(x_data2, y_data2, label = "dataset2")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.title("Datsets 1 and 2!")

plt.show()
```

Graph/Explanation:



In [6]:

```
'''
Dataset 1 seems to be much more sporadic, with points all over the place. Contrastingly, dataset2 has a parabolic shape,
nice curve of data.

'''
```

Task 4:

a)

In [7]:
```
import sklearn as sk
from sklearn.datasets import load_digits

digits = load_digits()
print(digits.DESCR)
```

In [8]:
```
'''
I. Number of Instances: 1797
II. Number of Attributes: 64
III. Attribute Information: 8x8 image of integer pixels in the range 0..16.
IV. Missing Attribute Values: None
V. Creator: E. Alpaydin (alpaydin '@' boun.edu.tr)
VI. Date: July; 1998
VII. Class: 10 classes where each class refers to a digit.

'''
```

b)

In [9]:
```
from sklearn.datasets import load_breast_cancer

breast_cancer = load_breast_cancer()
print(breast_cancer.DESCR)
```

In [10]:
```
'''
I. Number of Instances: 569
II. Number of Attributes: 30 numeric, predictive attributes and the class
III. Attribute Information:
    - radius (mean of distances from center to points on the perimeter)
    - texture (standard deviation of gray-scale values)
    - perimeter
    - area
    - smoothness (local variation in radius lengths)
    - compactness (perimeter^2 / area - 1.0)
    - concavity (severity of concave portions of the contour)
    - concave points (number of concave portions of the contour)
    - symmetry
    - fractal dimension ("coastline approximation" - 1)
IV. Missing Attribute Values: None
V. Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian
VI. Date: Date: November, 1995
VII. Class:
    - WDBC-Malignant
    - WDBC-Benign

'''
```