Assignment 1

Name	: Swapnil Santosh Kanade
PRN	: 2019BTECS00114
Batch	: T5
Course	: Software Engineering Tools Lab

1. Weka is a GUI workbench that empowers data wranglers to assemble machine learning pipelines, train models, and run predictions without having to write code.

Using Weka tool perform below tasks such as data preprocessing, data classification (use any appropriate ML algorithm) and data visualization efficiently on given dataset.

Use the Iris dataset given- https://drive.google.com/file/d/1A3Fxsfzm6BSfhFZGDrjI47RTe45bSgYP/view

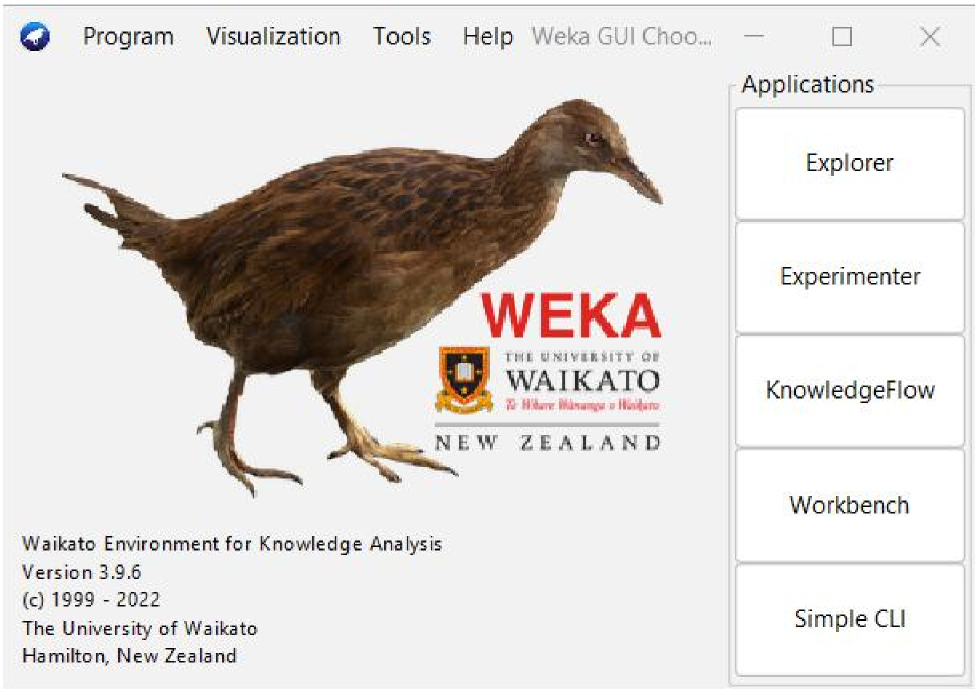
Note-provide screen shots for every task

Create a report which will illustrate the details of tasks performed (for e.g to perform preprocessing of data provide details of navigation and selection of appropriate parameters)

Ans.

Steps

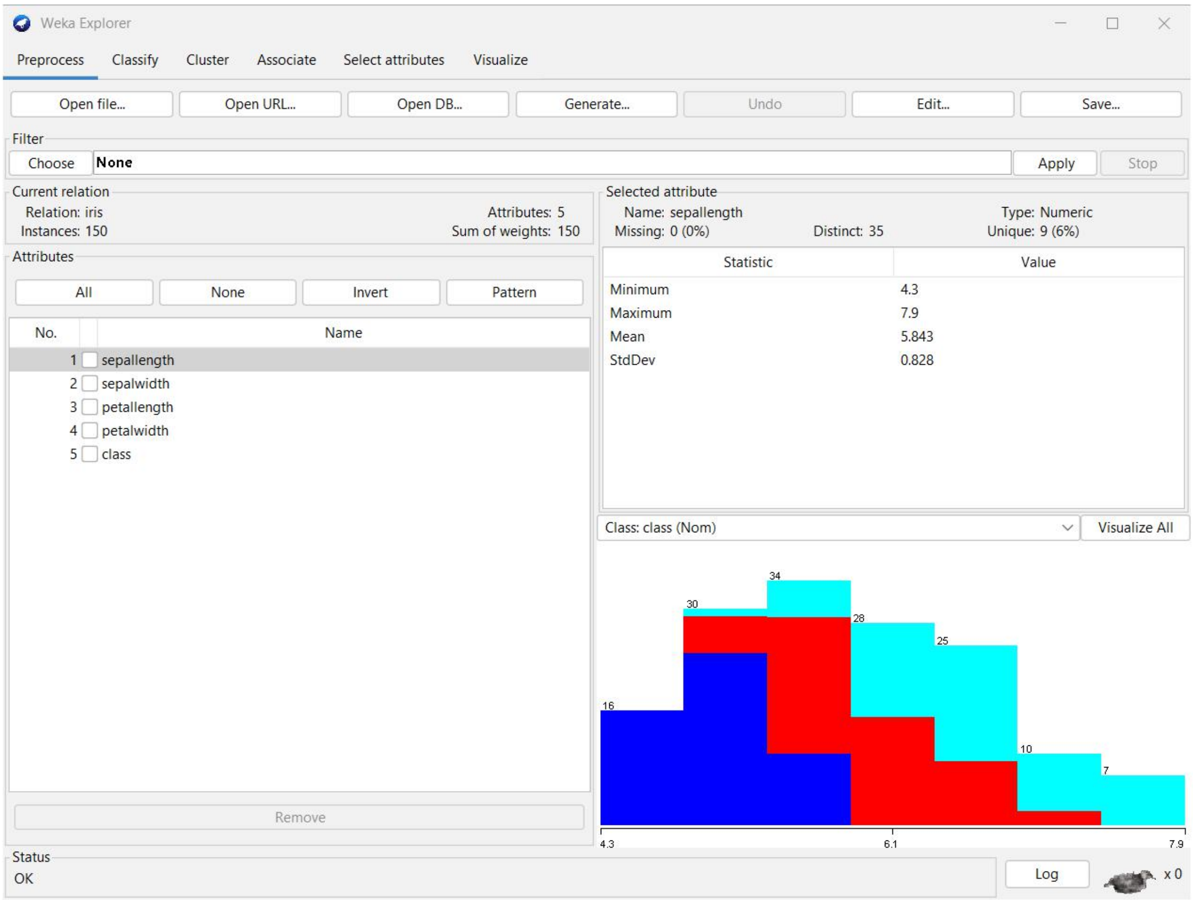
1) Downloading setup from https://sourceforge.net/projects/weka/files/weka-3-9/3.9.6/weka-3-9-6-azul-zulu-windows.exe/download?use_mirror=onboardcloud and installing it.



Waikato Environment for Knowledge Analysis
Version 3.9.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

Weka GUI interface

Data Preprocessing:



Filter: None

Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

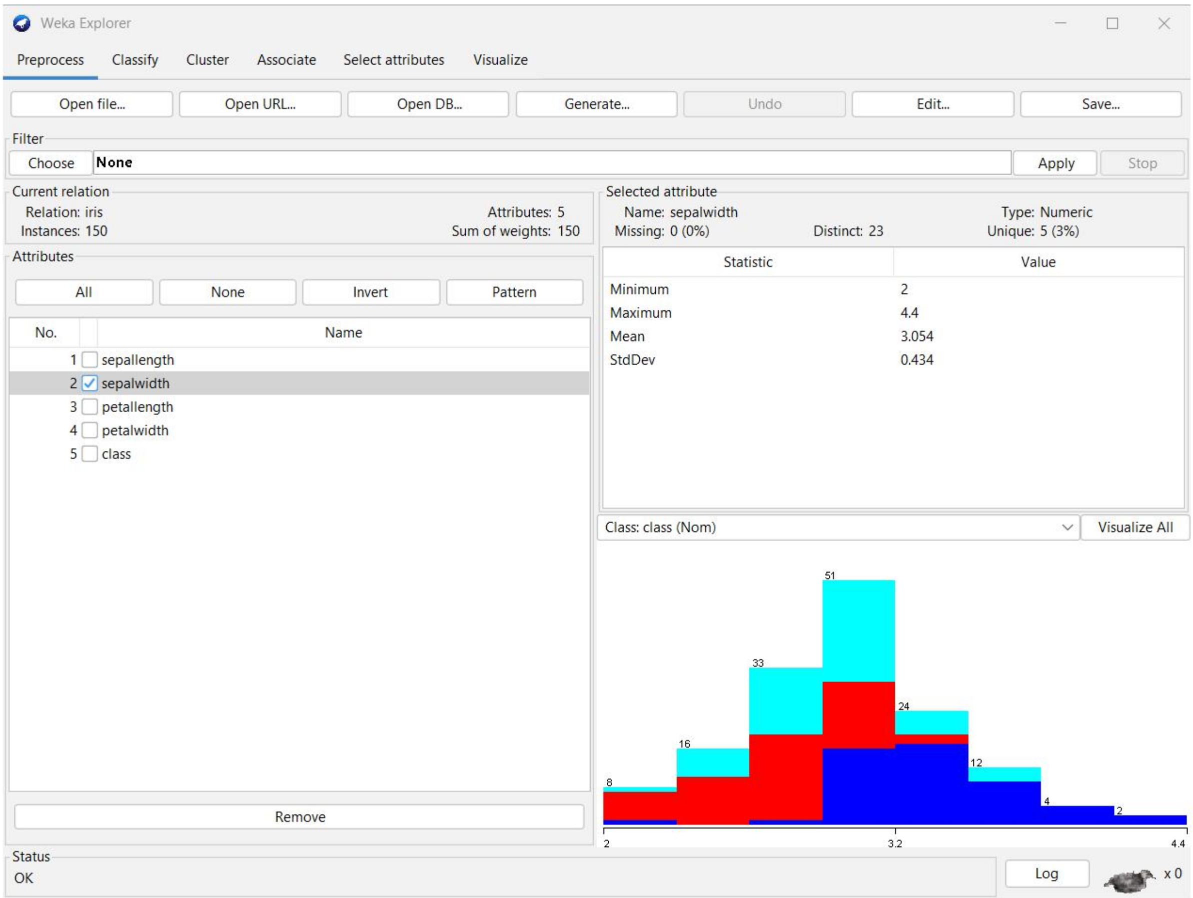
No.	Name
1	□ sepallength
2	□ sepalwidth
3	□ petallength
4	□ petalwidth
5	□ class

Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Class: class (Nom)

OK



Filter: None

Relation: iris
Attributes: 5
Instances: 150
Sum of weights: 150

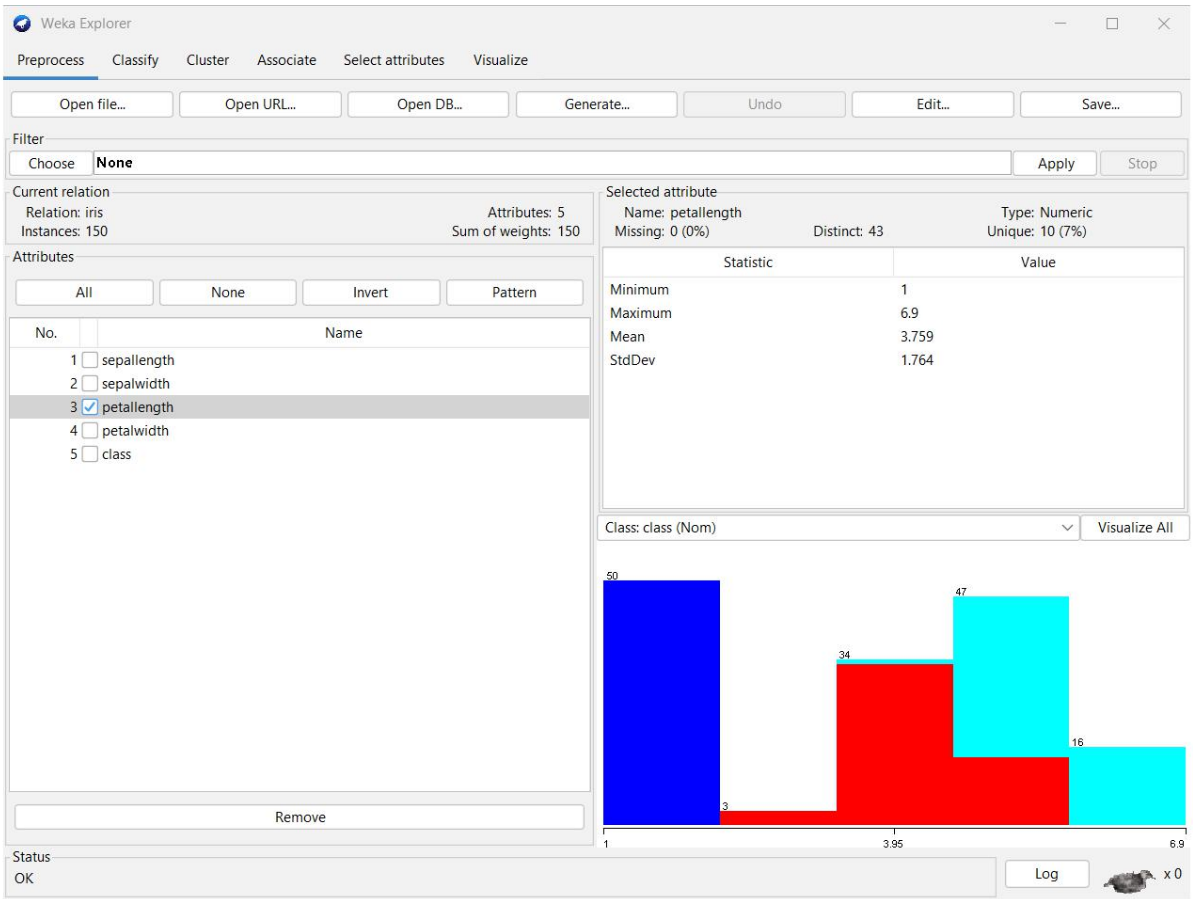
No.		Name
1	□	sepallength
2	☑	sepalwidth
3	□	petallength
4	□	petalwidth
5	□	class

Name: sepalwidth
Type: Numeric
Missing: 0 (0%)
Distinct: 23
Unique: 5 (3%)

Statistic	Value
Minimum	2
Maximum	4.4
Mean	3.054
StdDev	0.434

Class: class (Nom)

Status
OK



Filter: None

Relation: iris
Attributes: 5
Instances: 150
Sum of weights: 150

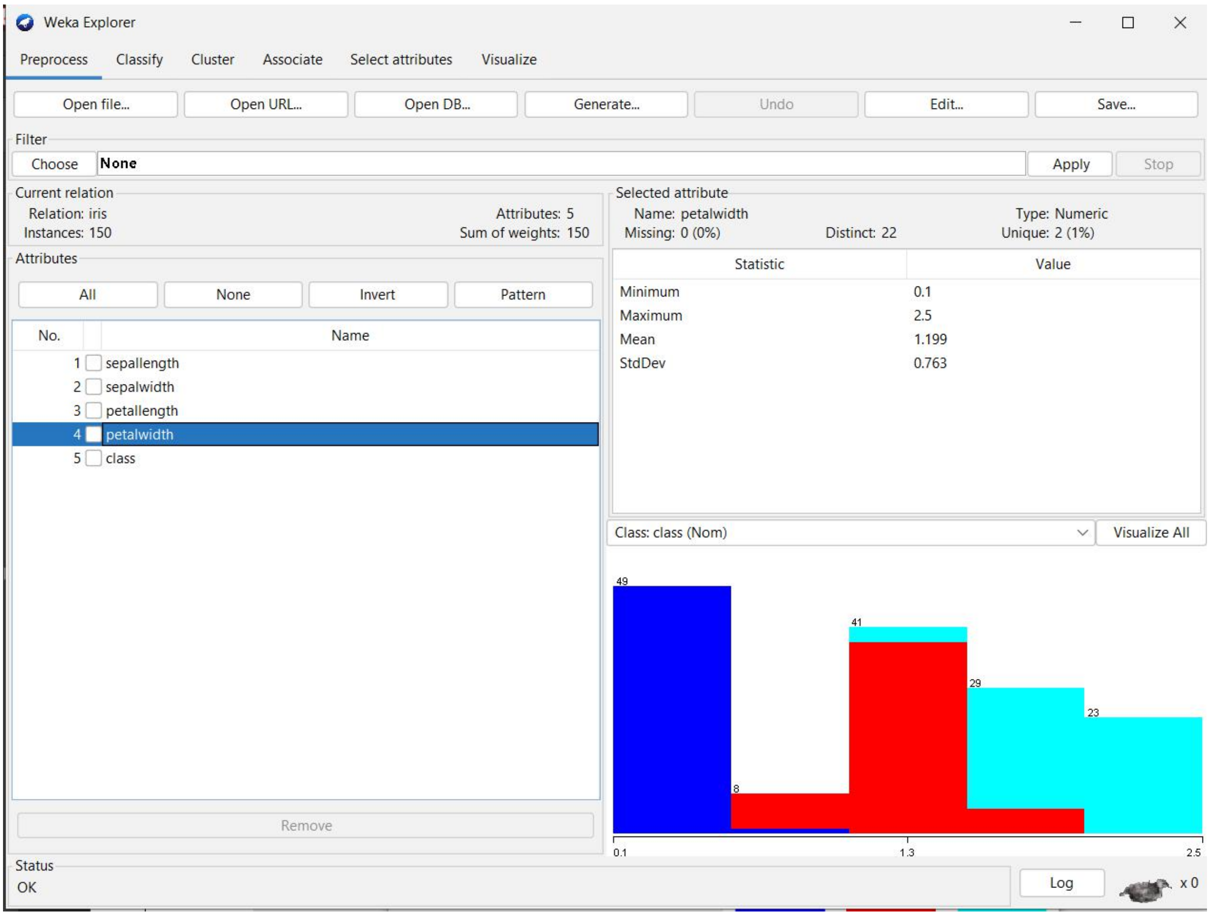
No.		Name
1	□	sepallength
2	□	sepalwidth
3	☑	petallength
4	□	petalwidth
5	□	class

Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)

Statistic	Value
Minimum	1
Maximum	6.9
Mean	3.759
StdDev	1.764

Class: class (Nom)

Status
OK



Filter: None

Current relation
Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

Attributes

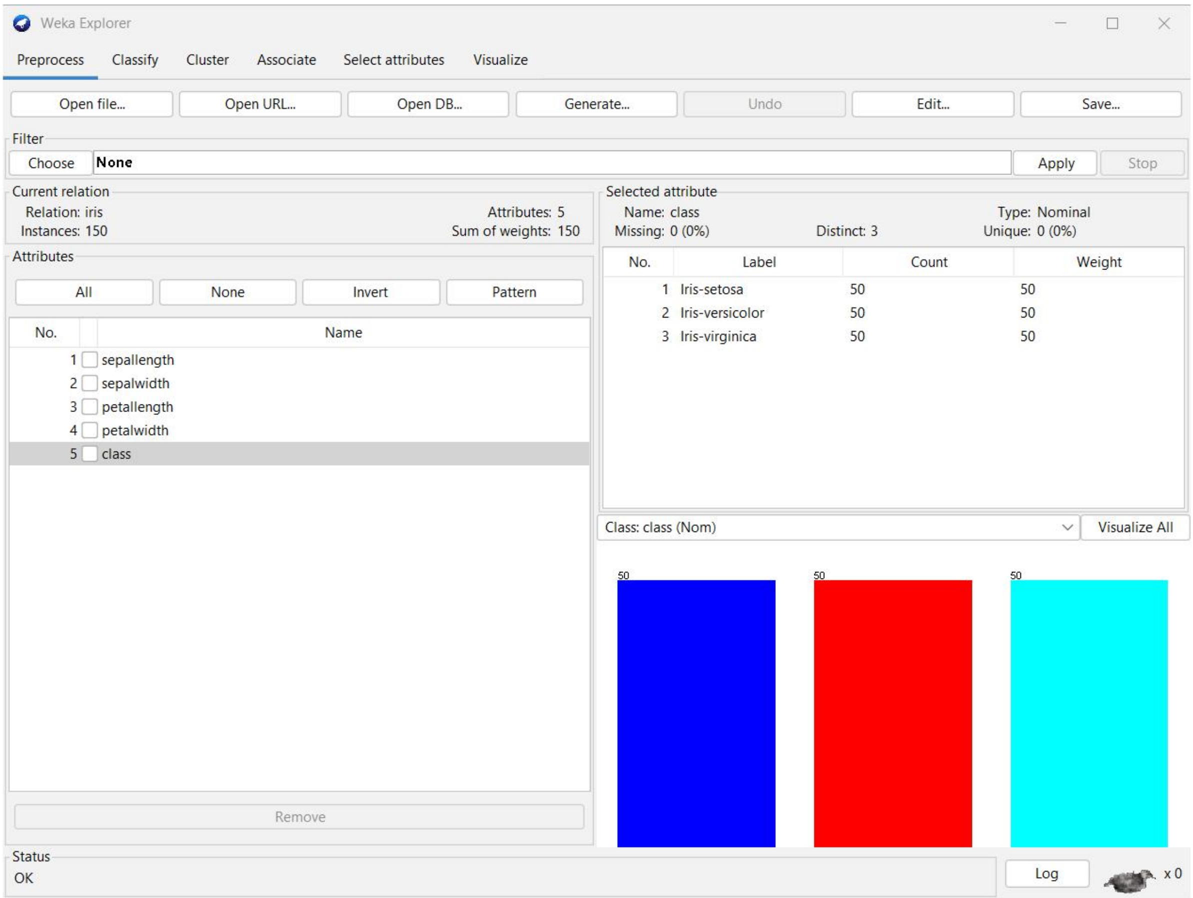
No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Selected attribute
Name: petalwidth
Type: Numeric
Missing: 0 (0%)
Distinct: 22
Unique: 2 (1%)

Statistic	Value
Minimum	0.1
Maximum	2.5
Mean	1.199
StdDev	0.763

Class: class (Nom)

Status
OK



Filter: None

Current relation
Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

Attributes

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Selected attribute
Name: class
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)

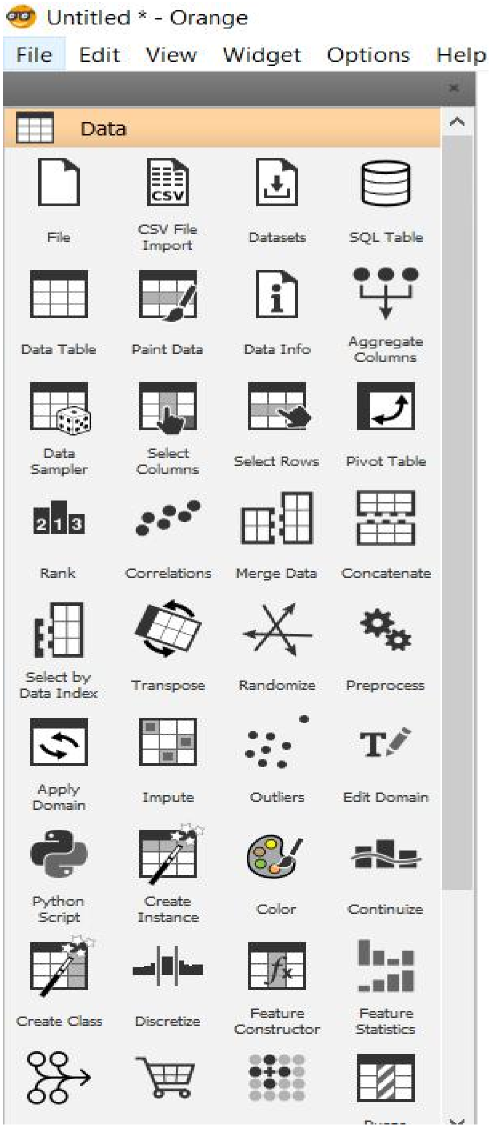
No.	Label	Count	Weight
1	Iris-setosa	50	50
2	Iris-versicolor	50	50
3	Iris-virginica	50	50

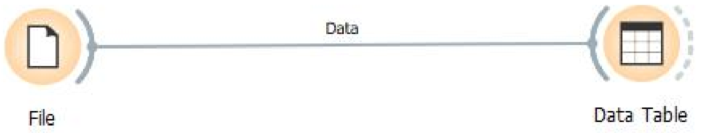
Class: class (Nom)

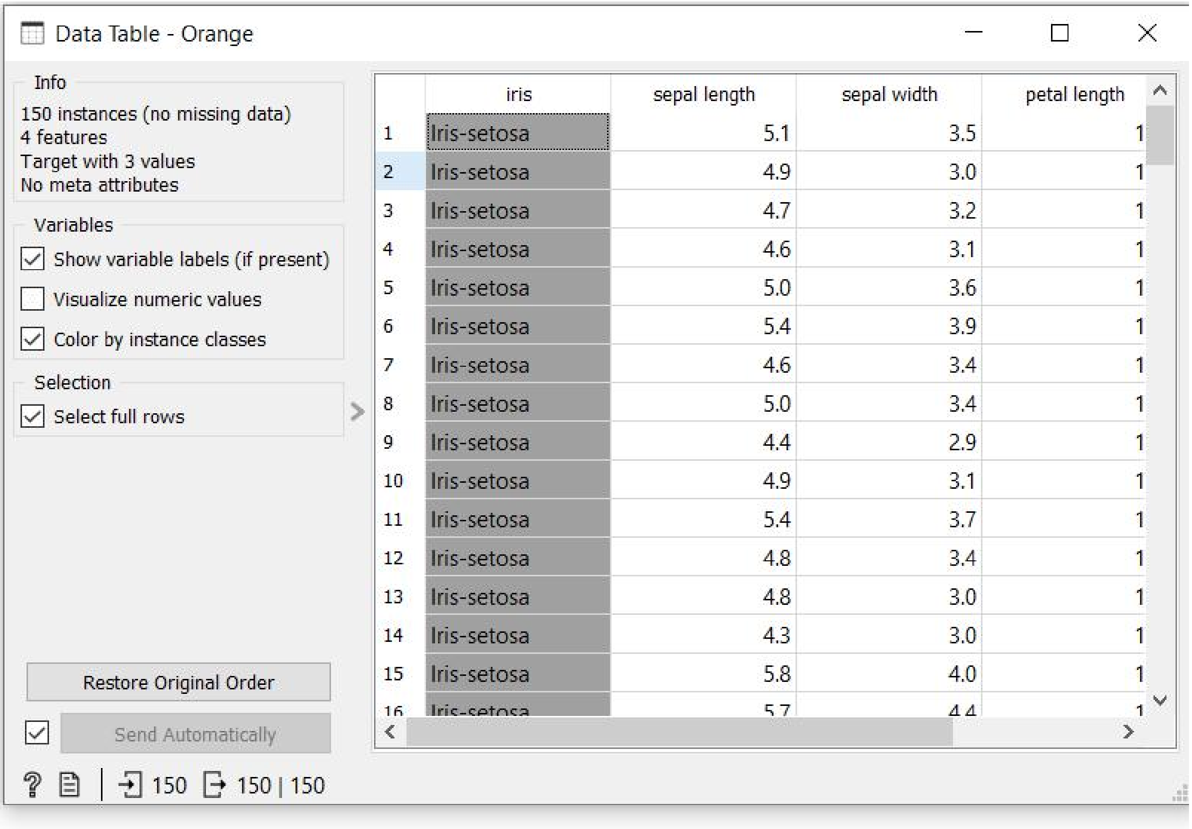
Status
OK

2) Orange is an easy-to-use data visualization tool with a large toolkit. In spite of being a GUI-based beginner-friendly tool, you mustn’t mistake it for a light-weight one. It can do statistical distributions and box plots as well as decision trees, hierarchical clustering and linear projections. a. Install orange b. Show data distribution c. Show linear projection d. Show FreeViz Use dataset https://drive.google.com/file/d/1m6sKI1Dap0XK6Bw1edUd5PohwpPwXnd9/view

=> 1) Download and install Orange from https://orangedatamining.com/download/#windows .







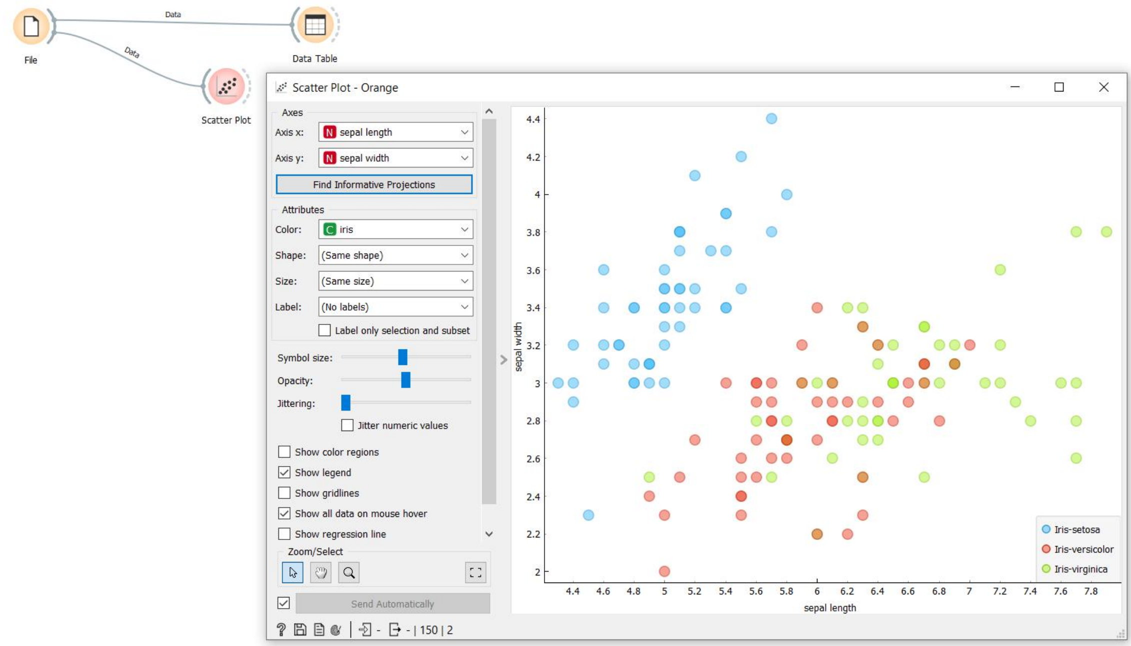
Info
150 instances (no missing data)
4 features
Target with 3 values
No meta attributes

Variables
☑ Show variable labels (if present)
☐ Visualize numeric values
☑ Color by instance classes

Selection
☑ Select full rows

	iris	sepal length	sepal width	petal length
1	Iris-setosa	5.1	3.5	1
2	Iris-setosa	4.9	3.0	1
3	Iris-setosa	4.7	3.2	1
4	Iris-setosa	4.6	3.1	1
5	Iris-setosa	5.0	3.6	1
6	Iris-setosa	5.4	3.9	1
7	Iris-setosa	4.6	3.4	1
8	Iris-setosa	5.0	3.4	1
9	Iris-setosa	4.4	2.9	1
10	Iris-setosa	4.9	3.1	1
11	Iris-setosa	5.4	3.7	1
12	Iris-setosa	4.8	3.4	1
13	Iris-setosa	4.8	3.0	1
14	Iris-setosa	4.3	3.0	1
15	Iris-setosa	5.8	4.0	1
16	Iris-setosa	5.7	4.4	1

Scatter plot:



Scatter Plot - Orange

Axes

Axis x: N sepal length

Axis y: N sepal width

Find Informative Projections

Attributes

Color: C iris

Shape: (Same shape)

Size: (Same size)

Label: (No labels)

□ Label only selection and subset

Symbol size:

Opacity:

Jittering:

□ Jitter numeric values

□ Show color regions

☑ Show legend

□ Show gridlines

☑ Show all data on mouse hover

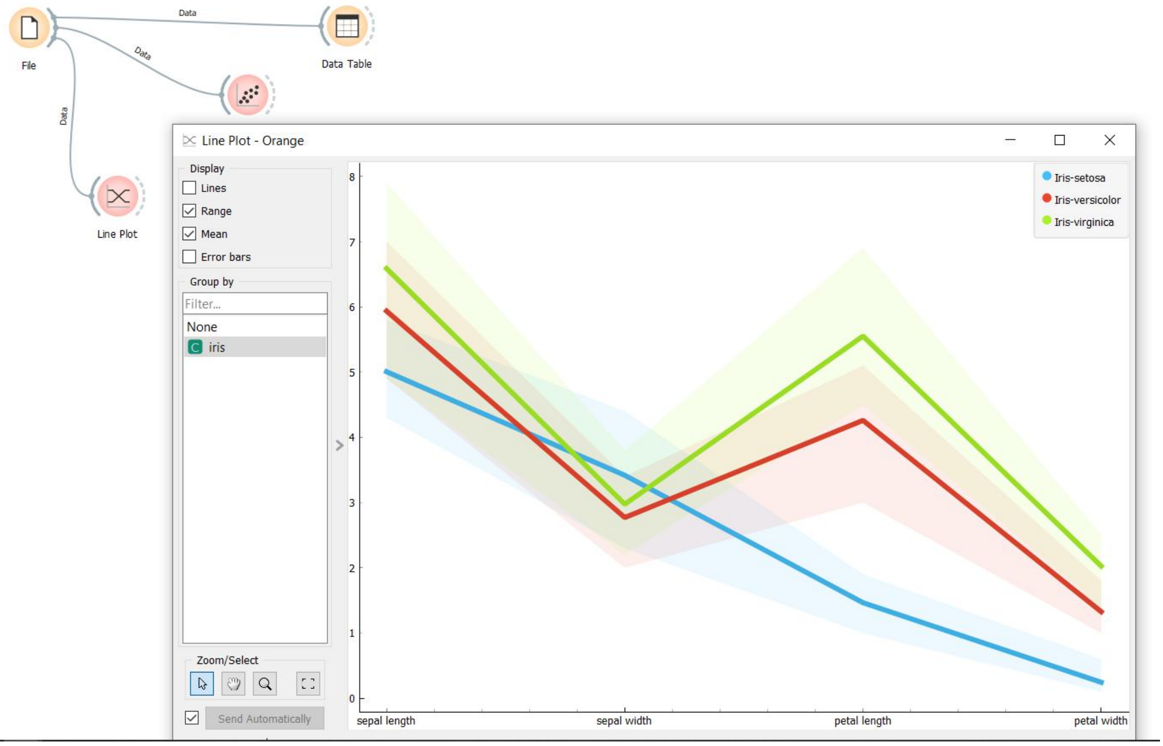
□ Show regression line

Zoom/Select

☑ Send Automatically

| 150 | 2

Line plot:



Line Plot - Orange

Display

□ Lines

☑ Range

☑ Mean

□ Error bars

Group by

Filter...

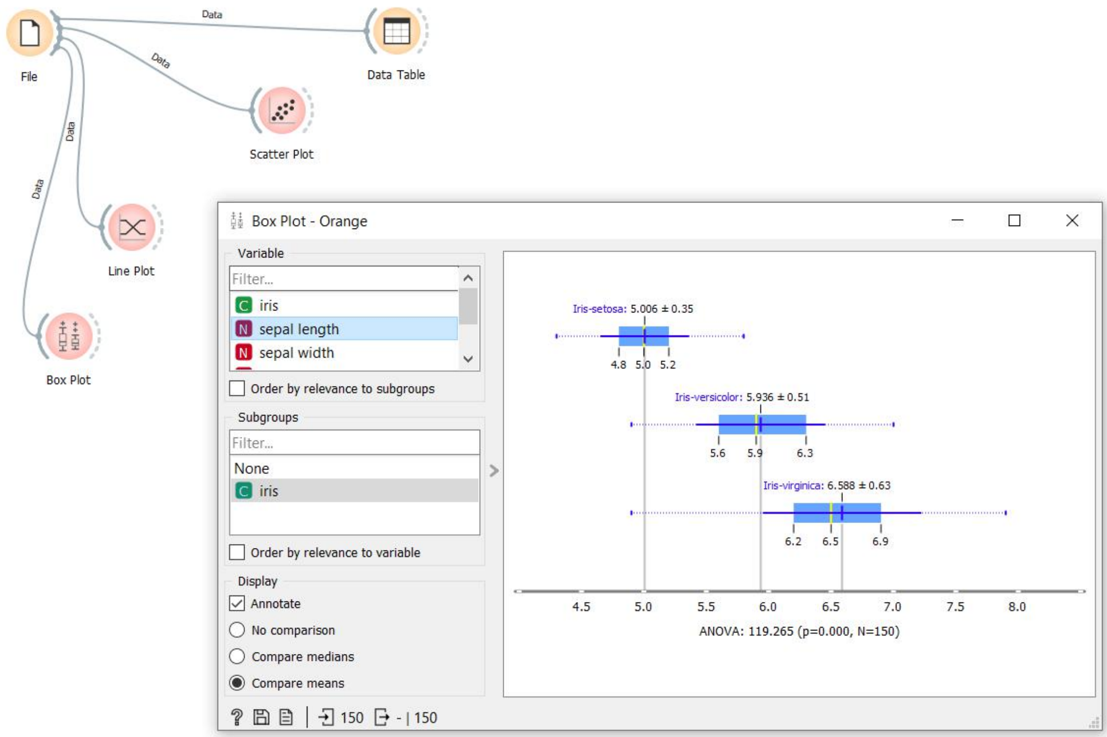
None

C iris

Zoom/Select

☑ Send Automatically

Box Plot:



□ Order by relevance to subgroups

□ Order by relevance to variable

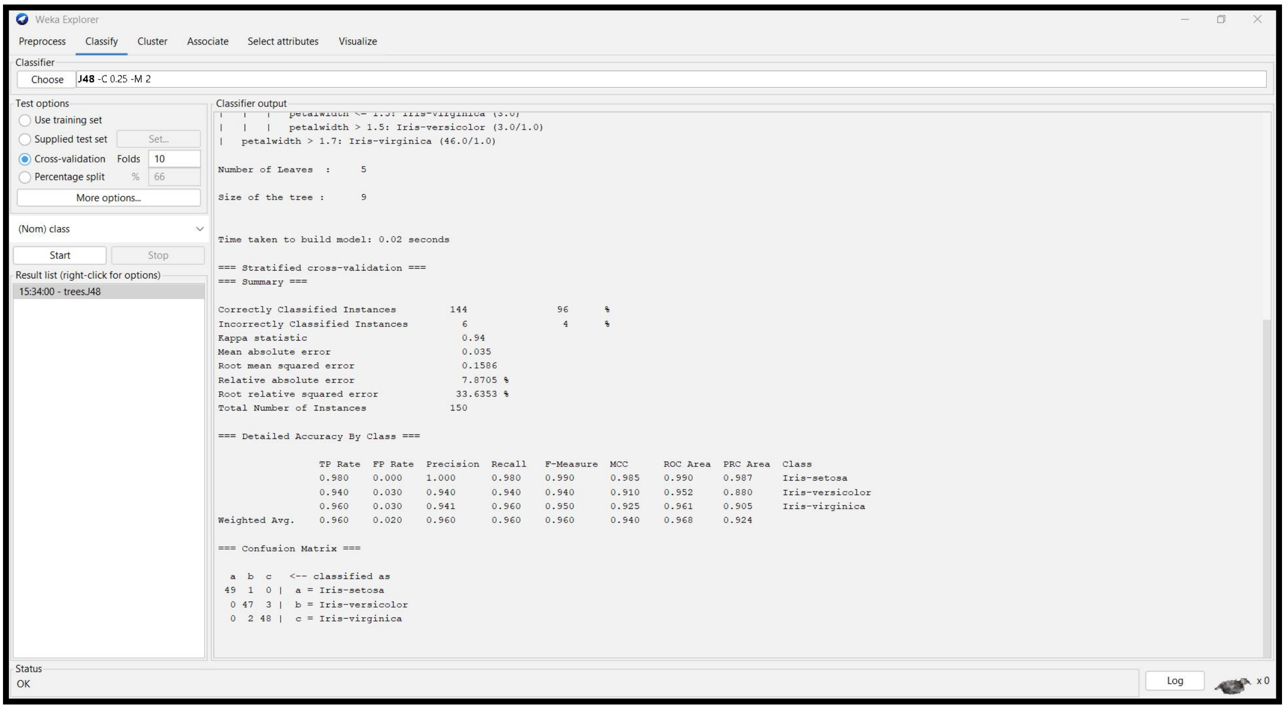
☑ Annotate

○ No comparison

○ Compare medians

● Compare means

ANOVA: 119.265 (p=0.000, N=150)



J48 -C 0.25 -M 2

○ Use training set

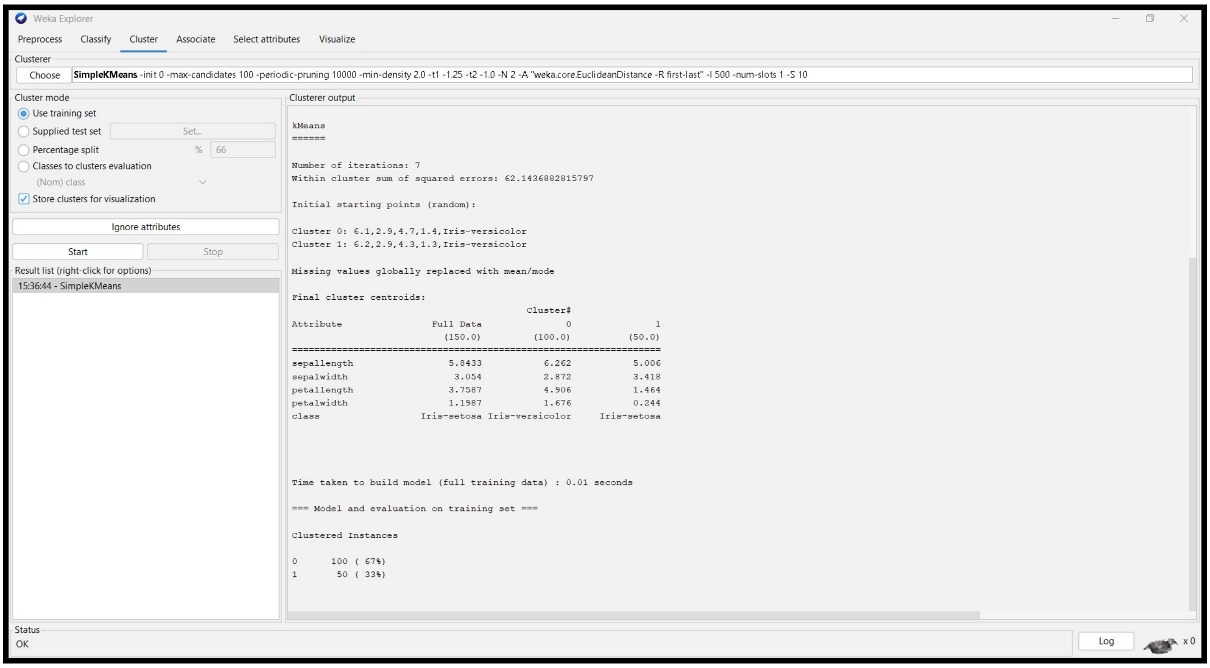
○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

```
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-setosa
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-versicolor
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-virginica
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```



Clusterer

SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

Result list (right-click for options)

15:36:44 - SimpleKMeans

Clusterer output

```
kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 62.1436882815797

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute                 Full Data            0             1
                            (150.0)      (100.0)        (50.0)
==============================================================
sepallength                  5.8433        6.262         5.006
sepalwidth                    3.054        2.872         3.418
petallength                  3.7587        4.906         1.464
petalwidth                   1.1987        1.676         0.244
class                   Iris-setosa Iris-versicolor Iris-setosa




Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      100 ( 67%)
1       50 ( 33%)
```

Status
OK

Q.3) Differentiate in between free software, Open-source software and proprietary software with respect to its properties.

Ans.

Proprietary software

Proprietary software (sometimes referred to as closed source software) is software that legally remains the property of the organization, group, or individual who created it. The organization that owns the rights to the product usually does not release the source code, and may insist that only those who have purchased a special license key can use it.

Free software

Free software (also called freeware) is licensed at no cost, or for an optional fee. It is usually closed source.

Open-source software

Open-source software is free and openly available to everyone. People who create open-source products publish the code and allow others to use and modify it. Communities of programmers often work together to develop the software and to support users. Open-source products are usually tested in public by online contributors.

Large companies such as Twitter, Facebook and the BBC make use of open-source technology. For example, the BBC makes use of MySQL and it creates open-source software, such as the program to improve the compatibility of iPlayer on smart TVs.

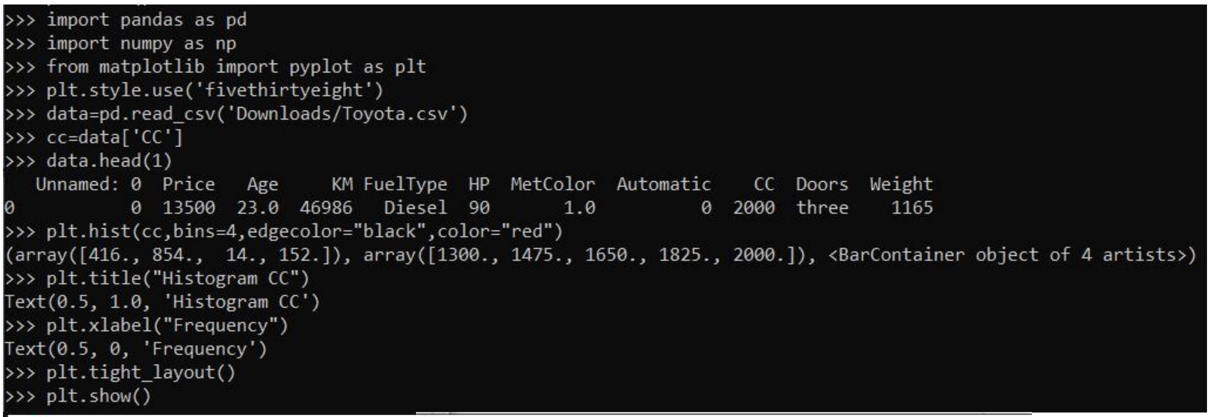
4. Using Anaconda Python create Histogram, Scatter plot and Bar plot for the dataset given below. Dataset- https://drive.google.com/file/d/1i11BZFe8Xj9kNq7eeE9KOa_Iz1KhEdXJ/view

a. Scatter plot- Scatter plot of Price Vs Age

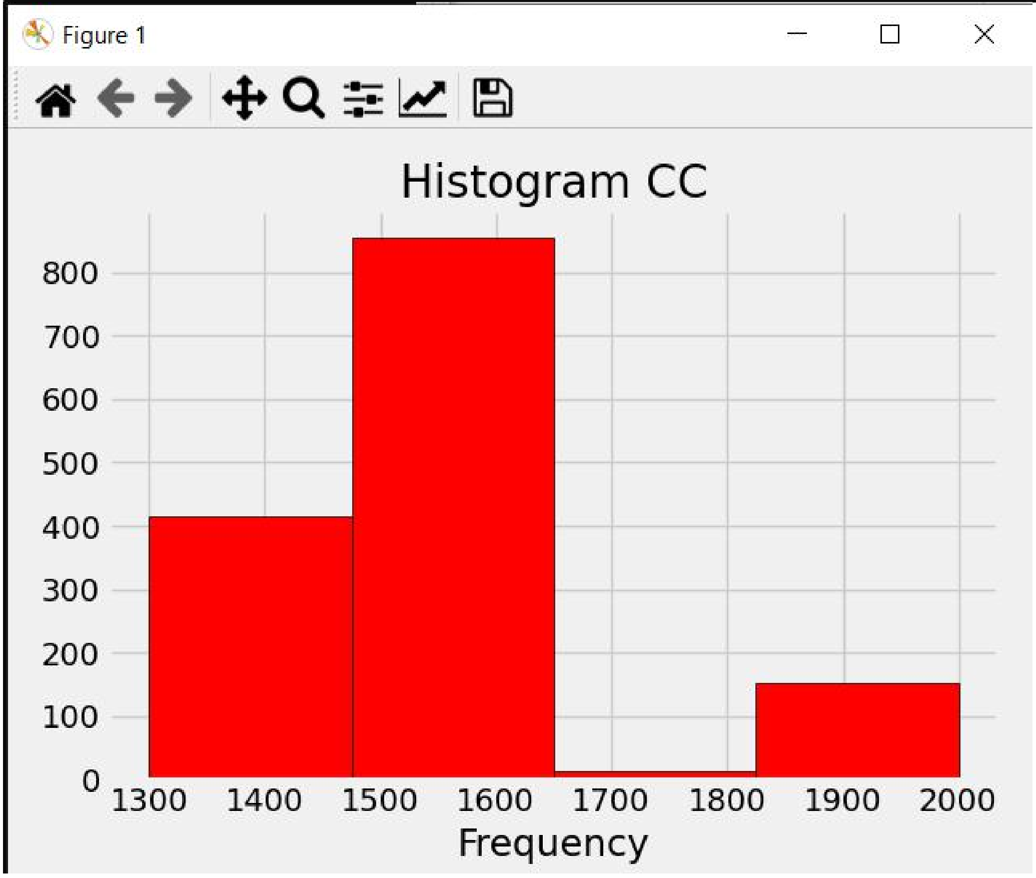
b. Histogram- for Kilometer and CC

c. Bar plot- Bar plot for different fuel types

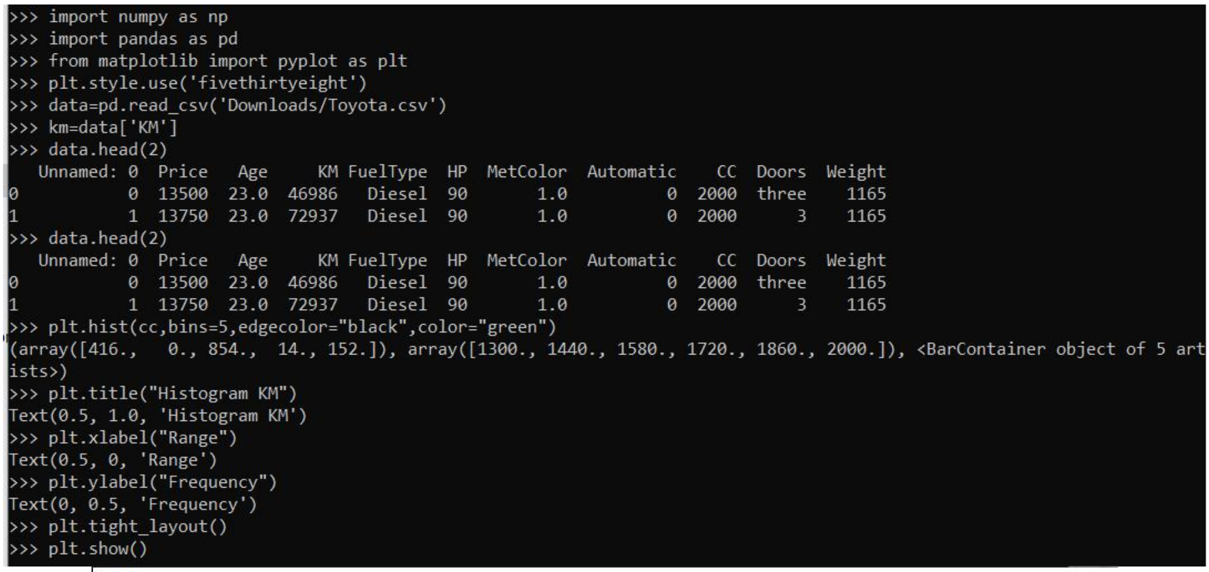
Ans=>



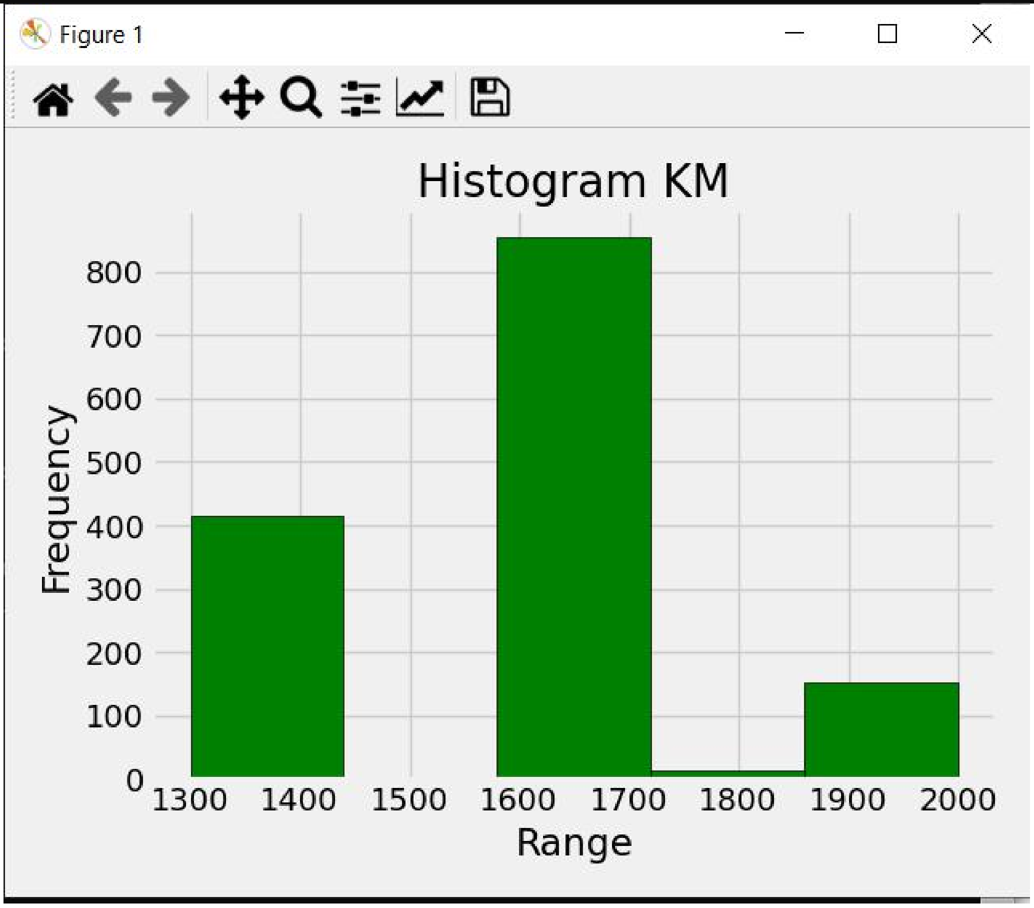
```
>>> import pandas as pd
>>> import numpy as np
>>> from matplotlib import pyplot as plt
>>> plt.style.use('fivethirtyeight')
>>> data=pd.read_csv('Downloads/Toyota.csv')
>>> cc=data['CC']
>>> data.head(1)
   Unnamed: 0  Price   Age     KM FuelType  HP  MetColor  Automatic    CC  Doors  Weight
0           0  13500  23.0  46986   Diesel  90       1.0          0  2000  three    1165
>>> plt.hist(cc,bins=4,edgecolor="black",color="red")
(array([416., 854.,  14., 152.]), array([1300., 1475., 1650., 1825., 2000.]), <BarContainer object of 4 artists>)
>>> plt.title("Histogram CC")
Text(0.5, 1.0, 'Histogram CC')
>>> plt.xlabel("Frequency")
Text(0.5, 0, 'Frequency')
>>> plt.tight_layout()
>>> plt.show()
```

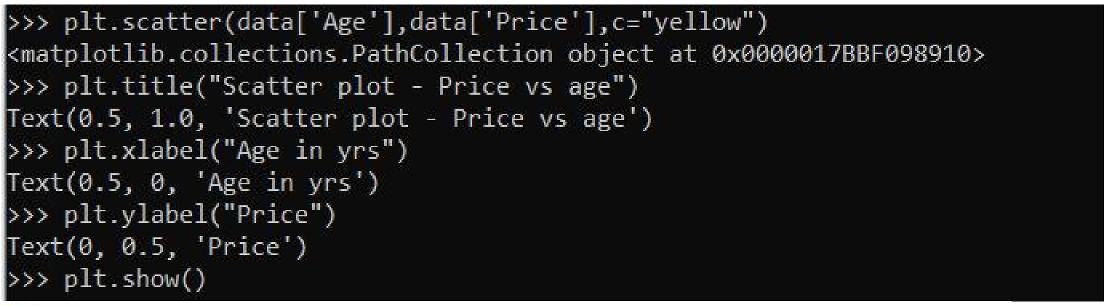


b. Histogram- for Kilometer and CC

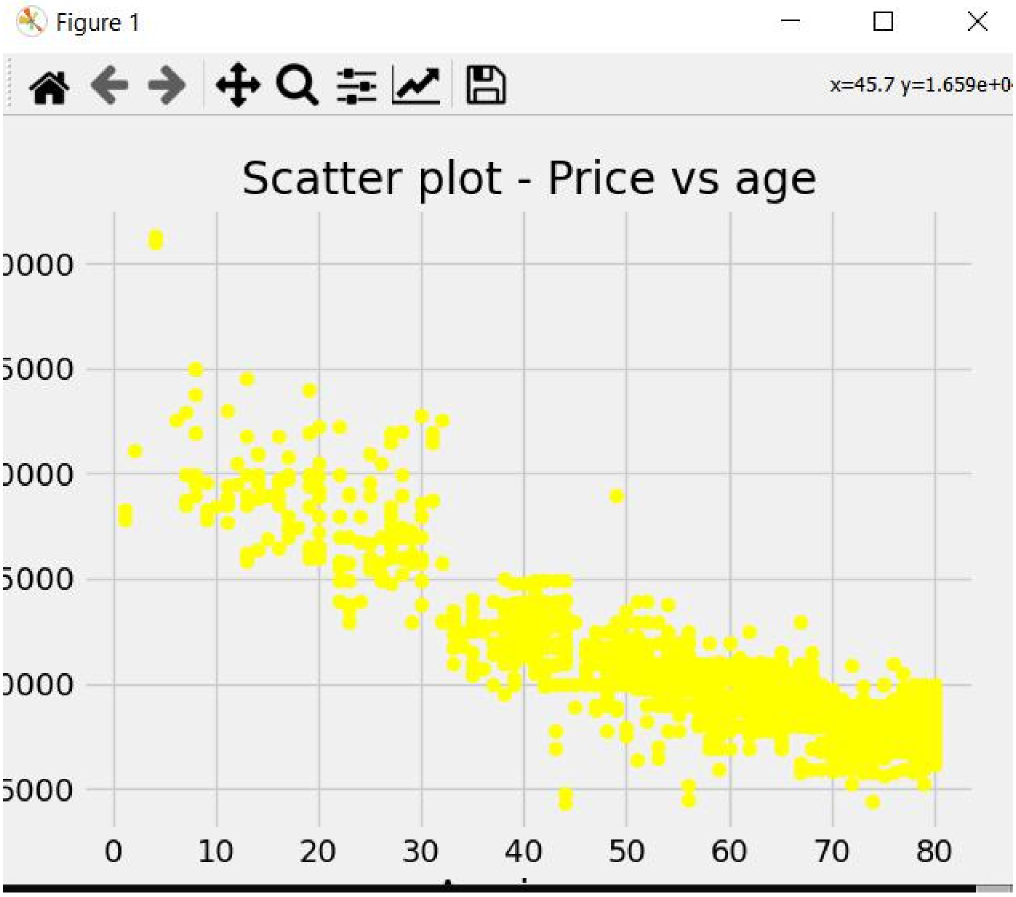


```
>>> import numpy as np
>>> import pandas as pd
>>> from matplotlib import pyplot as plt
>>> plt.style.use('fivethirtyeight')
>>> data=pd.read_csv('Downloads/Toyota.csv')
>>> km=data['KM']
>>> data.head(2)
   Unnamed: 0  Price   Age     KM FuelType  HP  MetColor  Automatic    CC  Doors  Weight
0           0  13500  23.0  46986   Diesel  90       1.0          0  2000  three    1165
1           1  13750  23.0  72937   Diesel  90       1.0          0  2000      3    1165
>>> data.head(2)
   Unnamed: 0  Price   Age     KM FuelType  HP  MetColor  Automatic    CC  Doors  Weight
0           0  13500  23.0  46986   Diesel  90       1.0          0  2000  three    1165
1           1  13750  23.0  72937   Diesel  90       1.0          0  2000      3    1165
>>> plt.hist(cc,bins=5,edgecolor="black",color="green")
(array([416.,   0., 854.,  14., 152.]), array([1300., 1440., 1580., 1720., 1860., 2000.]), <BarContainer object of 5 art
ists>)
>>> plt.title("Histogram KM")
Text(0.5, 1.0, 'Histogram KM')
>>> plt.xlabel("Range")
Text(0.5, 0, 'Range')
>>> plt.ylabel("Frequency")
Text(0, 0.5, 'Frequency')
>>> plt.tight_layout()
>>> plt.show()
```

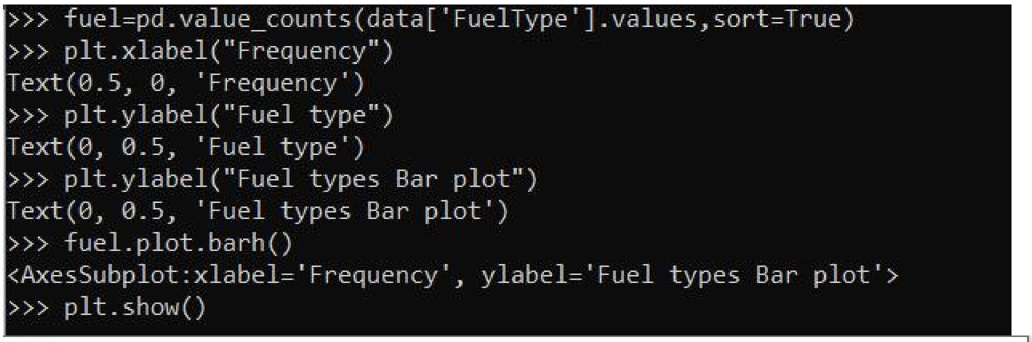




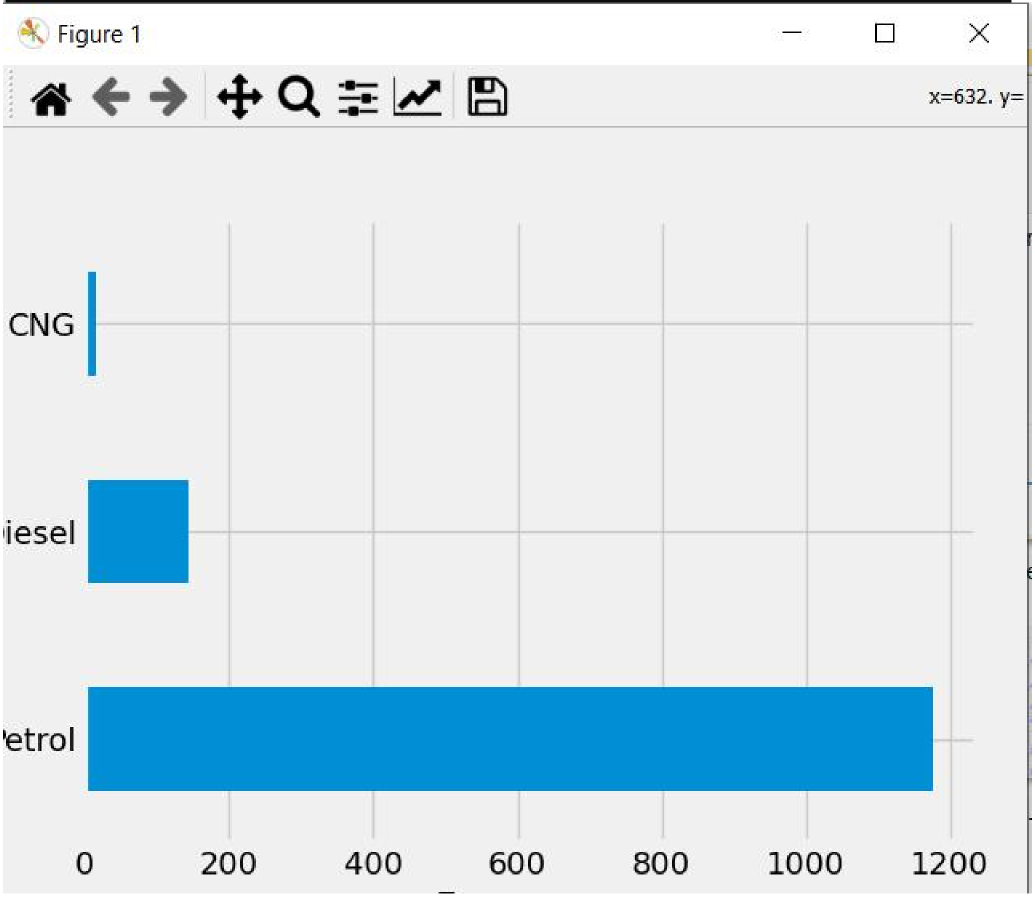
```
>>> plt.scatter(data['Age'],data['Price'],c="yellow")
<matplotlib.collections.PathCollection object at 0x0000017BBF098910>
>>> plt.title("Scatter plot - Price vs age")
Text(0.5, 1.0, 'Scatter plot - Price vs age')
>>> plt.xlabel("Age in yrs")
Text(0.5, 0, 'Age in yrs')
>>> plt.ylabel("Price")
Text(0, 0.5, 'Price')
>>> plt.show()
```



c. Bar plot- Bar plot for different fuel type
s



```
>>> fuel=pd.value_counts(data['FuelType'].values,sort=True)
>>> plt.xlabel("Frequency")
Text(0.5, 0, 'Frequency')
>>> plt.ylabel("Fuel type")
Text(0, 0.5, 'Fuel type')
>>> plt.ylabel("Fuel types Bar plot")
Text(0, 0.5, 'Fuel types Bar plot')
>>> fuel.plot.barh()
<AxesSubplot:xlabel='Frequency', ylabel='Fuel types Bar plot'>
>>> plt.show()
```



5. Enlist some examples along with its purpose and properties (at least 10) of FOSS and proprietary software with respect to database.

Examples of Open-Source S/W

- VLC Media Player
- Mozilla Firefox
- GIMP
- VNC

- Apache Web Server
- JQuery
- Weka
- Orange
- Anaconda Python

Examples of Free S/W

- Linux Kernel
- GNU Compiler Collection
- C Library
- MYSQL relational database
- Apache web server
- Sendmail mail transport agent
- Emacs text editor
- LaTex