Kevin Eibeck

861227549

Late days used for this assignment = 0

Total late days used = 0

Assignment 1

To run the program refer to the README.txt attached.

# Question 1: Feature distribution

## Histograms

Approach:

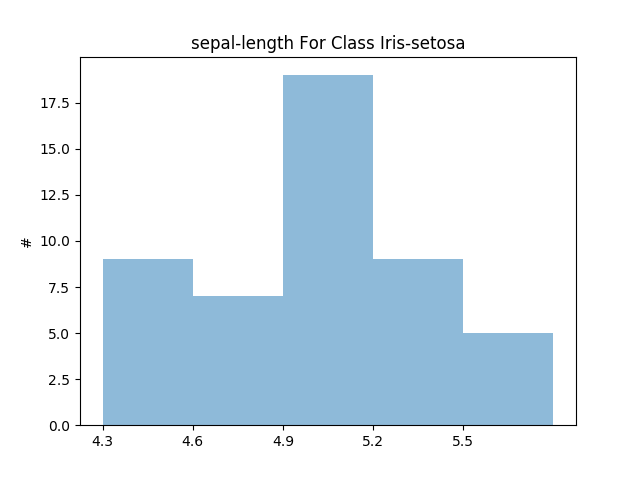
I use the min and max function to find out the range of values, then I divide by the binNum that the user asked for. I then iterate through a loop to initialize the list I use as the x-axis. Then I iterate through a second loop to fill out the histogram bins. Then I display it.

Dataset: Iris

Attribute: Sepal-length

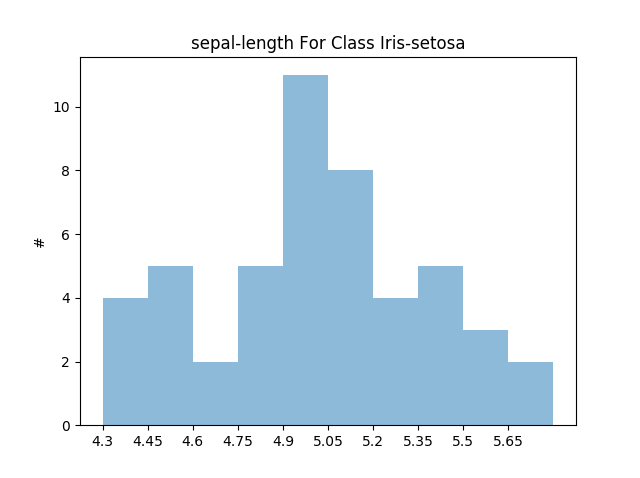
Class 1: Iris-setosa

Bin Size = 5



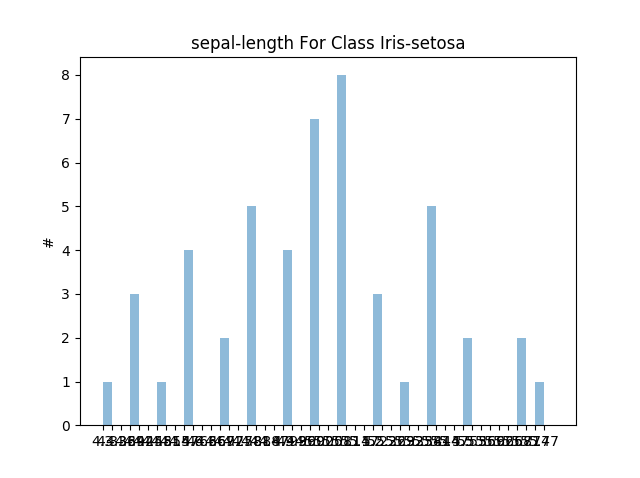
Mostly symmetric & mostly unimodal

Bin Size = 10



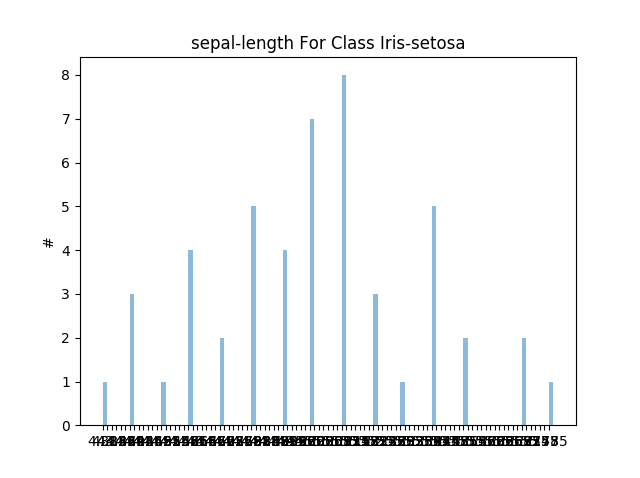
Mostly symmetric & mostly unimodal

Bin Size = 50



Mostly symmetric & mostly unimodal

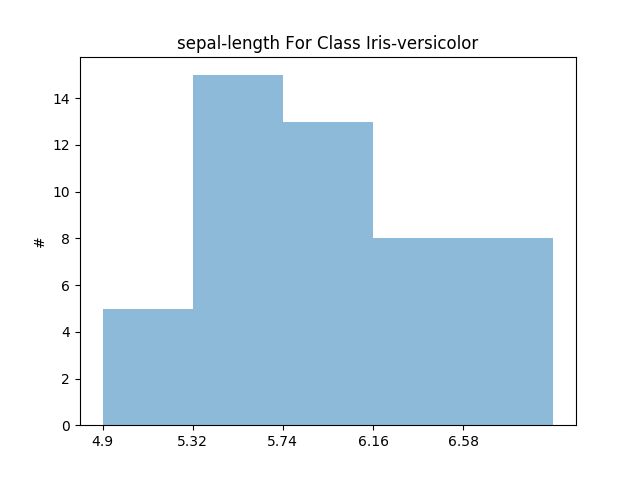
Bin Size = 100



Mostly symmetric & mostly unimodal

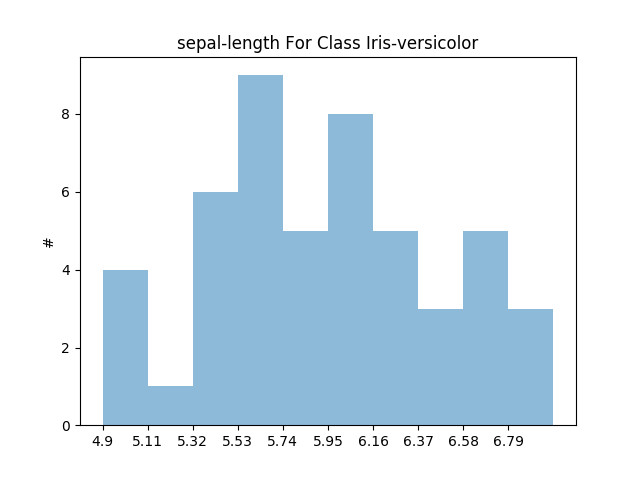
Class 2:

Bin Size = 5



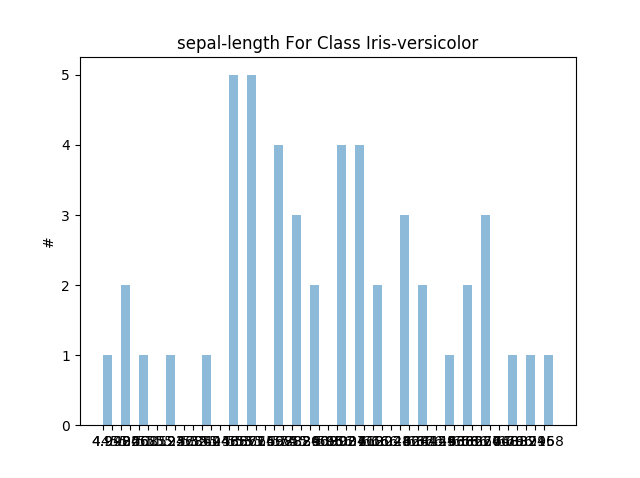
Mostly skewed & mostly unimodal

Bin Size = 10



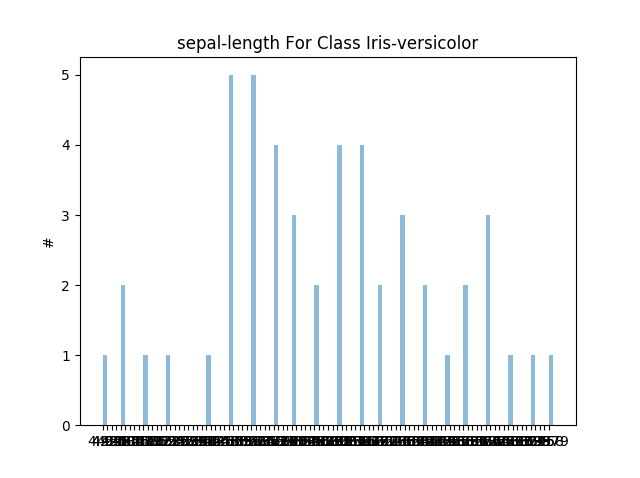
Mostly Skewed & mostly unimodal

Bin Size = 50



Mostly Skewed & mostly unimodal

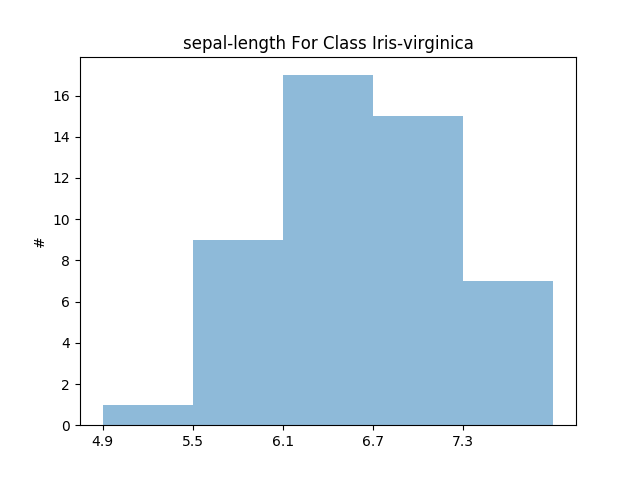
Bin Size = 100



Mostly Skewed & mostly unimodal

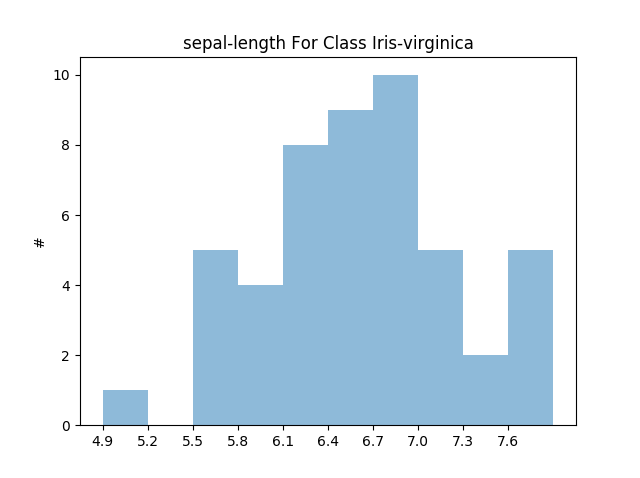
Class 3:

Bin Size = 5



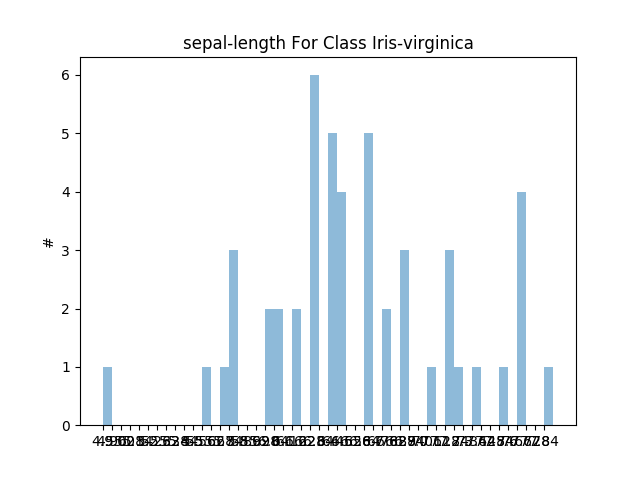
Mostly Skewed & mostly unimodal

Bin Size = 10



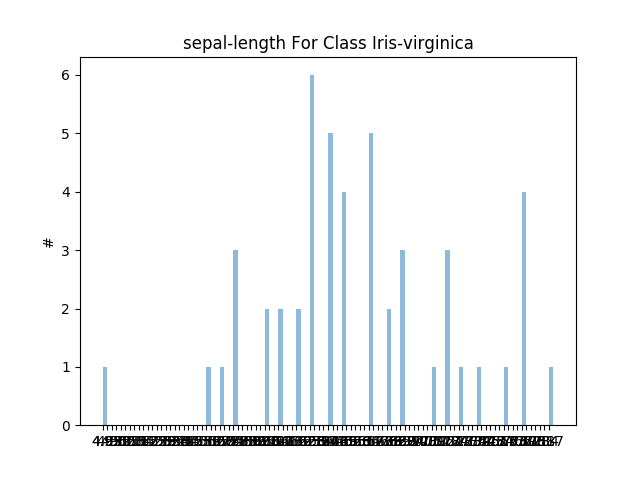
Mostly skewed & mostly bimodal

Bin Size = 50



Mostly skewed & mostly bimodal

Bin Size = 100

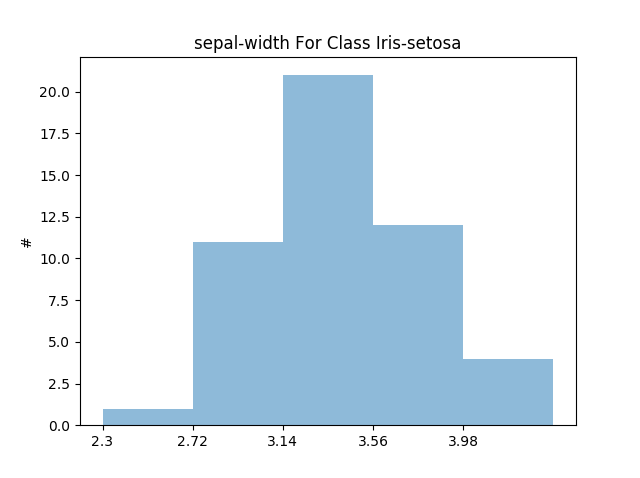


Mostly skewed & mostly bimodal

Attribute = Sepal Width

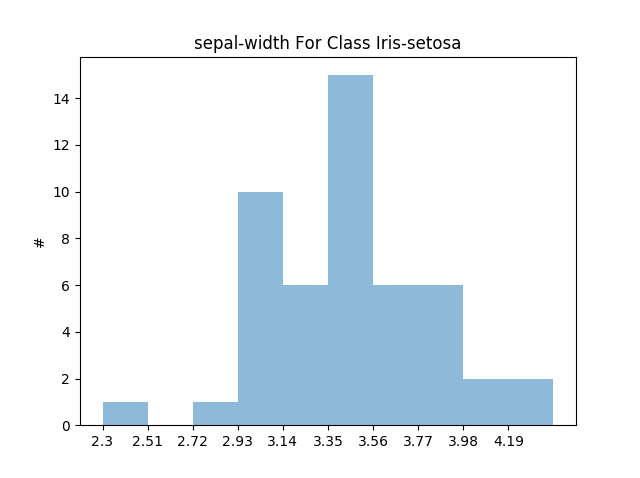
Class 1:

Bin Size = 5



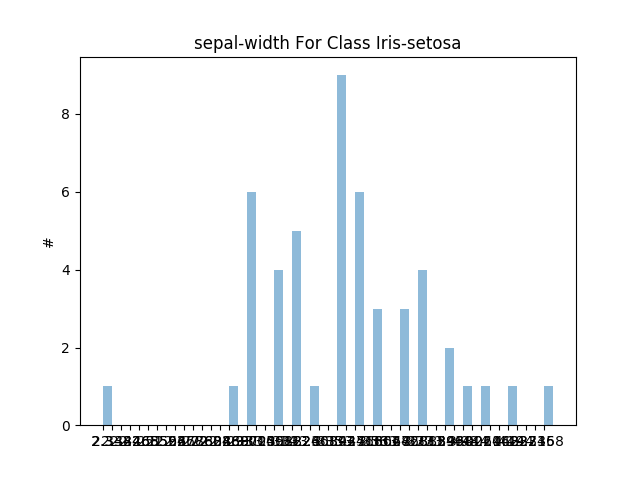
Mostly symmetric & mostly uniform

Bin Size = 10



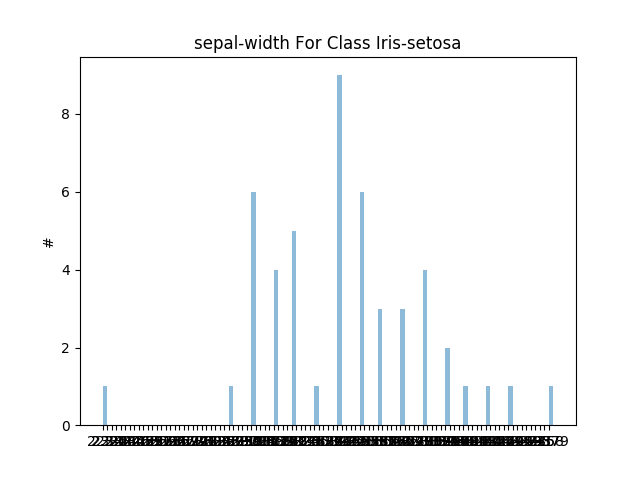
Mostly skewed and mostly bimodal

Bin Size = 50



Mostly skewed and mostly bimodal

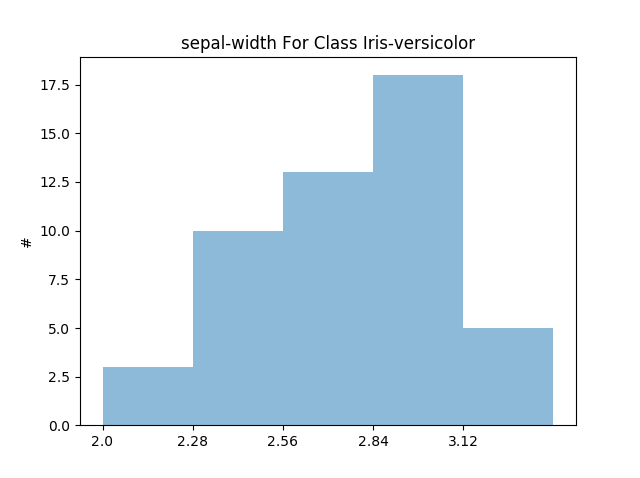
Bin Size = 100



Mostly skewed and mostly bimodal

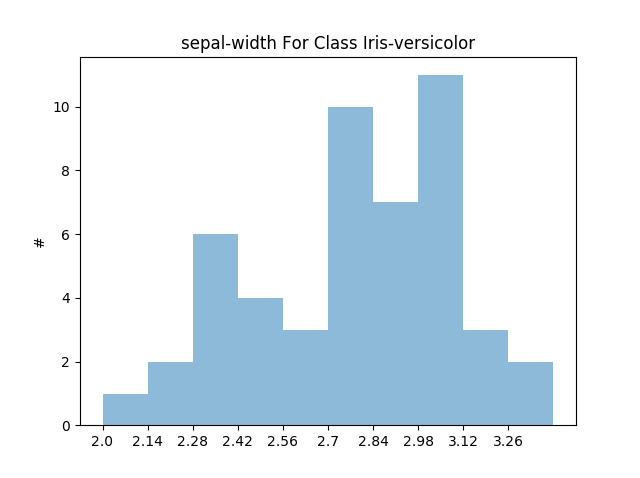
Class 2:

Bin Size = 5



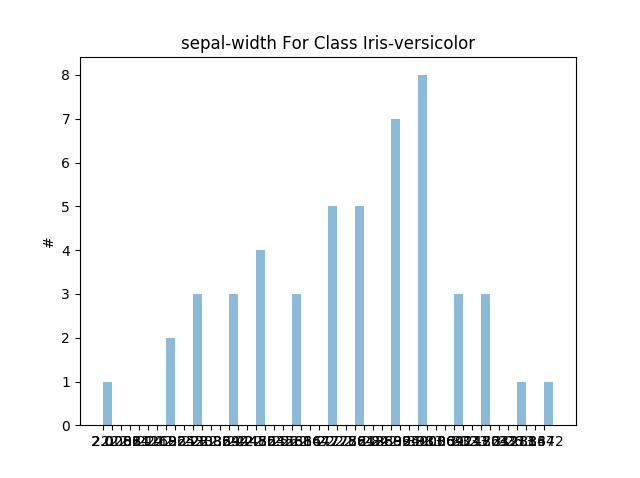
Mostly skewed and mostly unimodal

Bin Size = 10



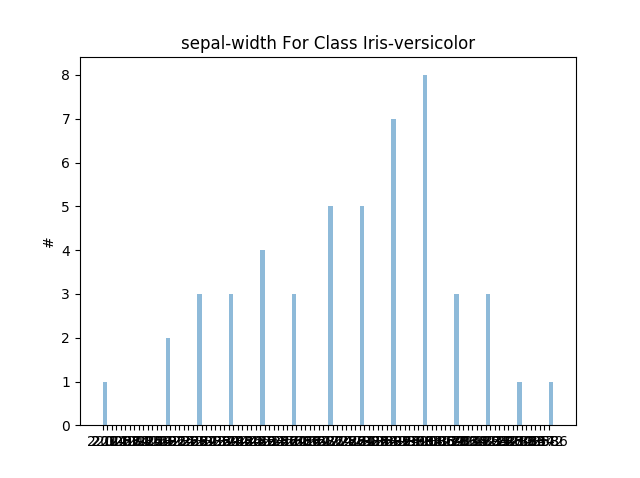
Mostly skewed & mostly multimodal

Bin Size = 50



Mostly skewed & mostly unimodal

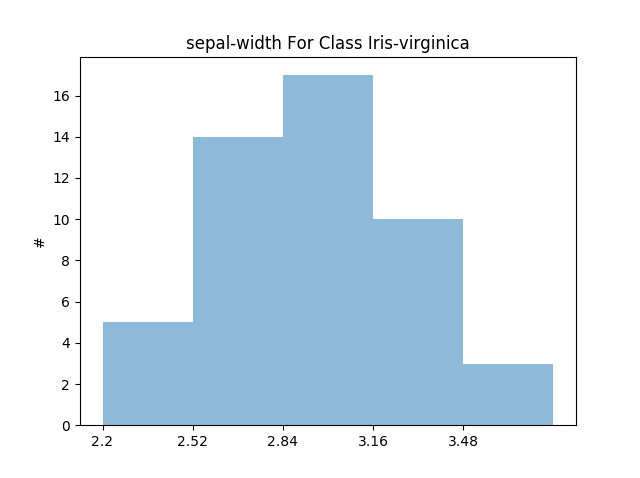
Bin Size = 100



Mostly skewed & mostly unimodal

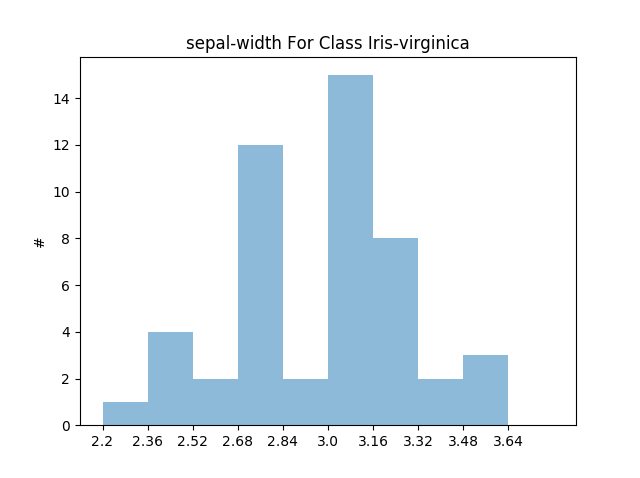
Class 3:

Bin Size = 5



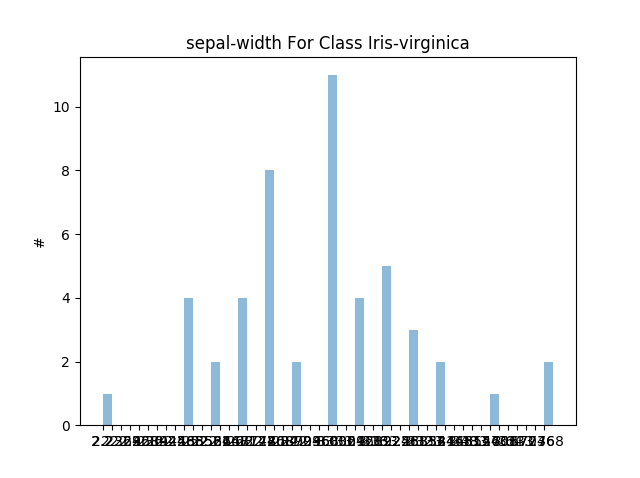
Mostly skewed & mostly unimodal

Bin Size = 10



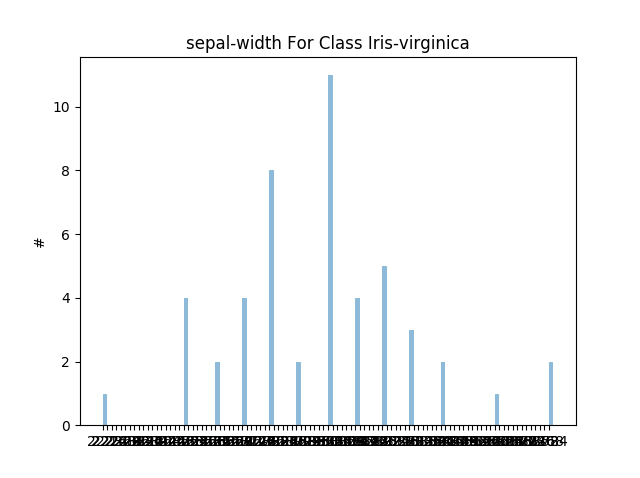
Mostly skewed & mostly multimodal

Bin Size = 50



Mostly symmetric & mostly unimodal

Bin Size = 100

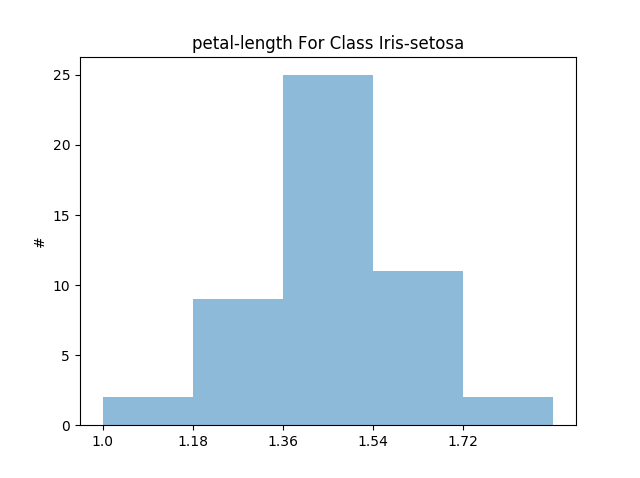


Mostly symmetric & mostly unimodal

Attribute = Pedal Length

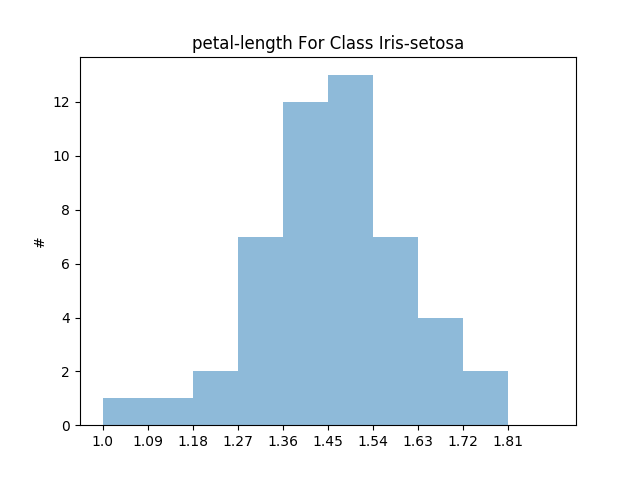
Class 1:

Bin Size = 5



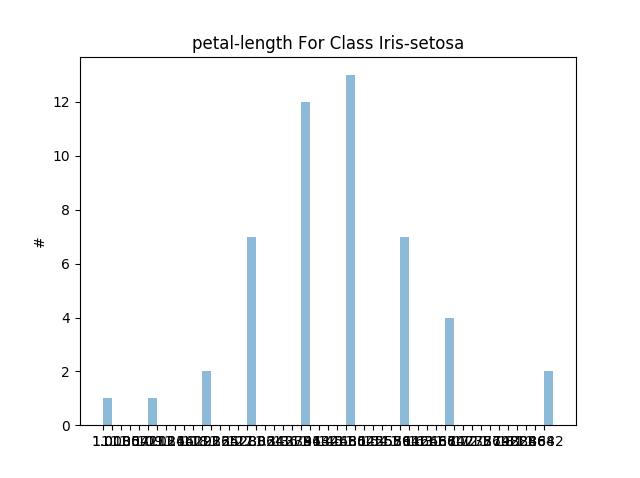
Mostly symmetric & mostly unimodal

Bin Size = 10



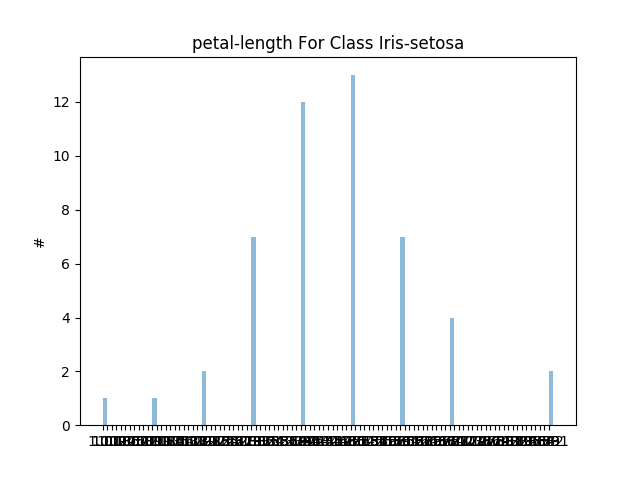
Mostly symmetric & mostly unimodal

Bin Size = 50



Mostly symmetric & mostly unimodal

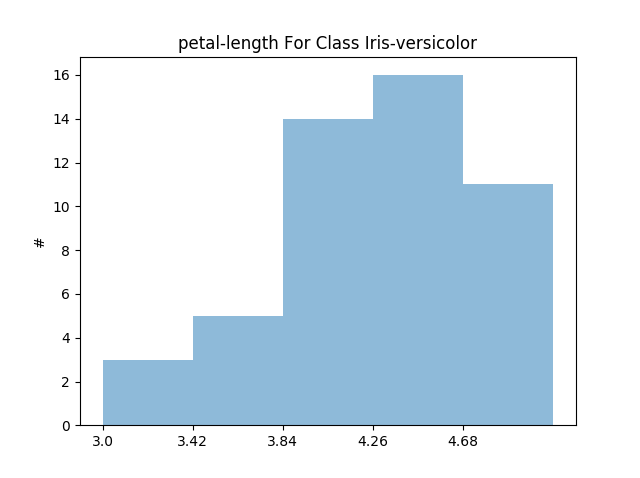
Bin Size = 100



Mostly symmetric & mostly unimodal

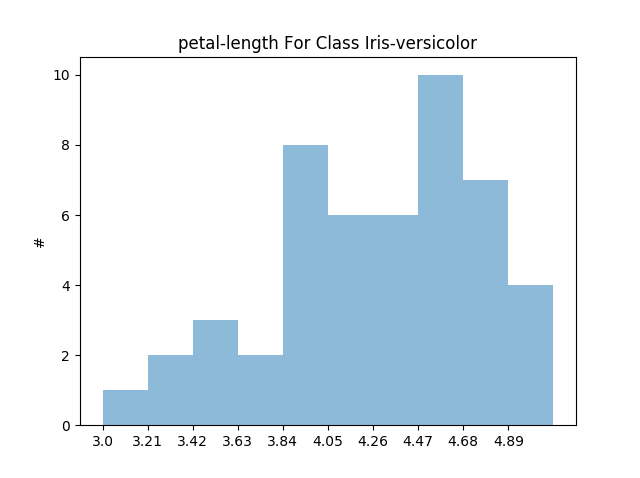
Class 2:

Bin Size = 5



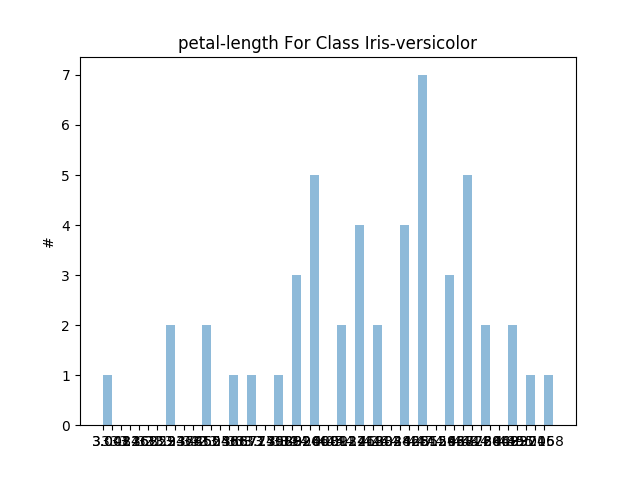
Mostly skewed & mostly unimodal

Bin Size = 10



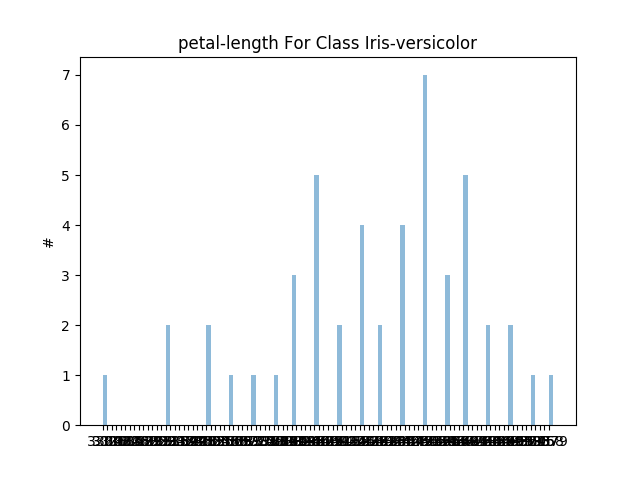
Mostly skewed & mostly bimodal

Bin Size = 50



Mostly skewed & mostly bimodal

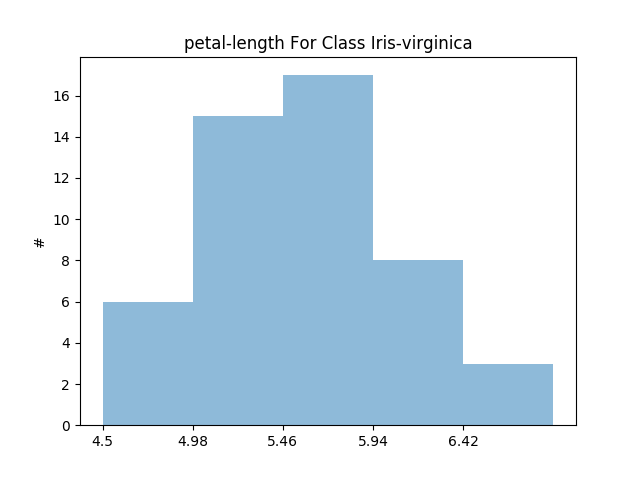
Bin Size = 100



Mostly skewed & mostly bimodal

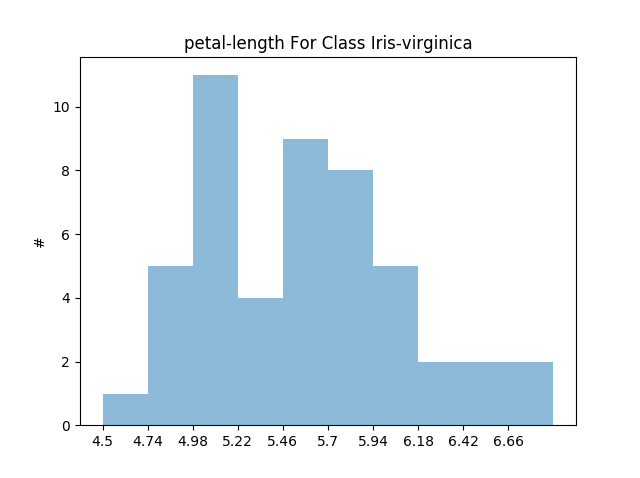
Class 3:

Bin Size = 5



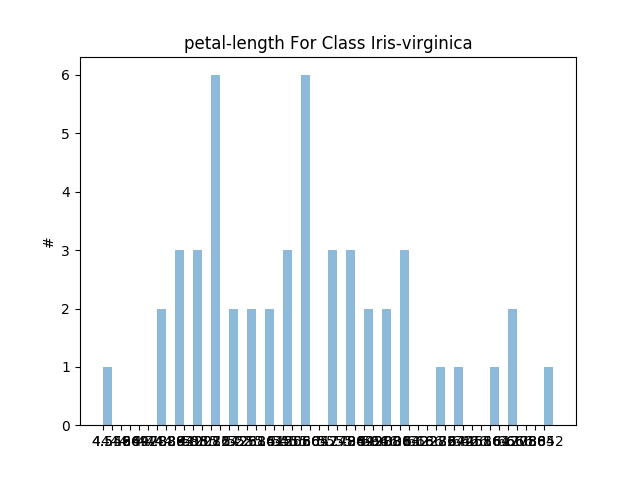
Mostly symmetric & mostly unimodal

Bin Size = 10



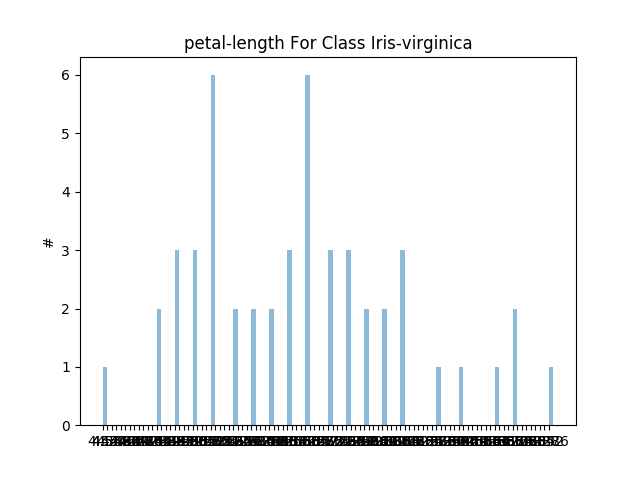
Mostly skewed & mostly bimodal

Bin Size = 50



Mostly skewed & mostly bimodal

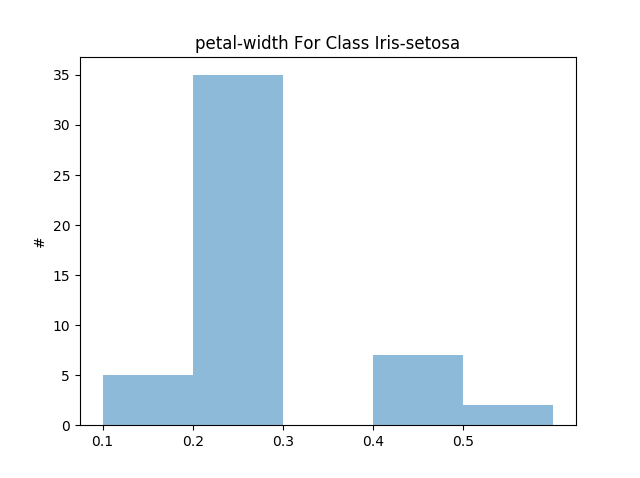
Bin Size = 100



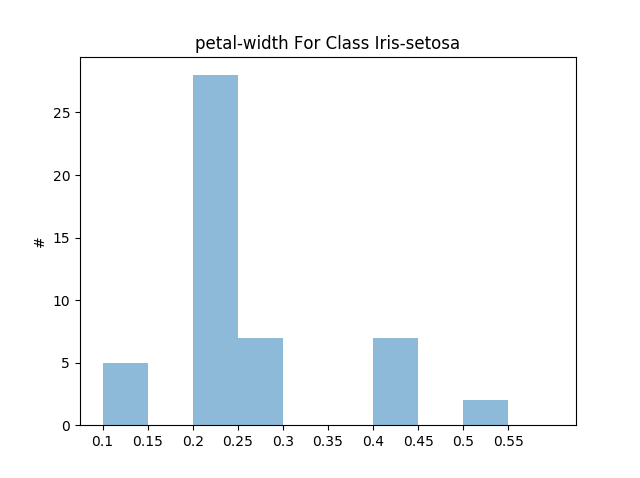
Mostly skewed & mostly bimodal

Attribute = Pedal Width

Class 1:

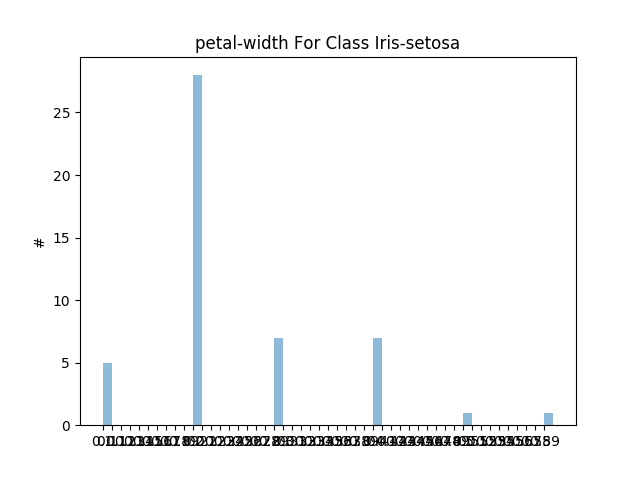
Bin Size = 5

Mostly skewed & mostly unimodal

Bin Size = 10

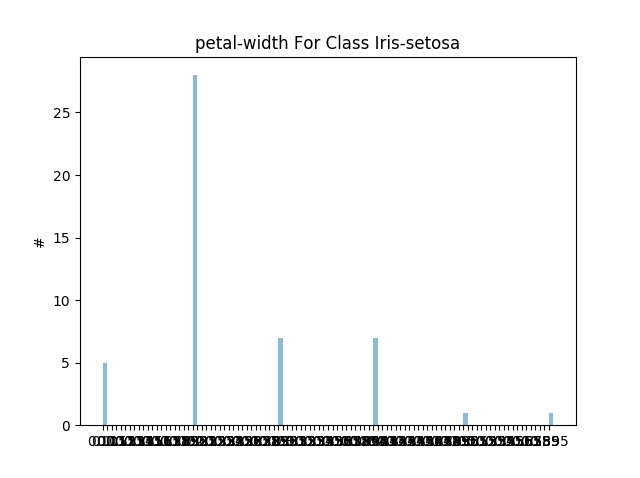
Mostly skewed & mostly unimodal

Bin Size = 50



Mostly skewed & mostly unimodal

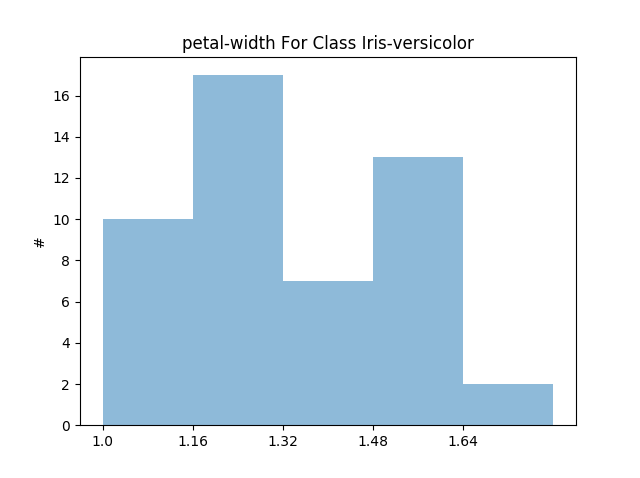
Bin Size = 100



Mostly skewed & mostly unimodal

Class 2:

Bin Size = 5



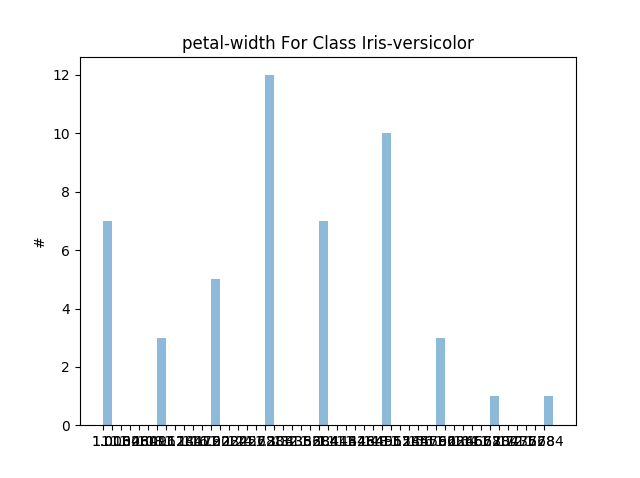
Mostly skewed & mostly bimodal

Bin Size = 10

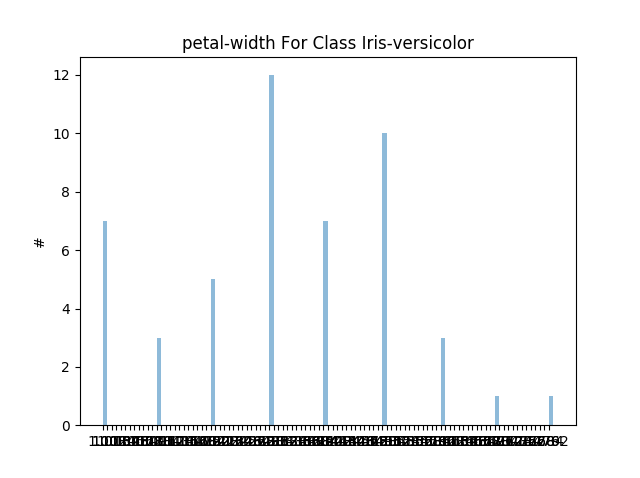


Mostly skewed & mostly multimodal

Bin Size = 50



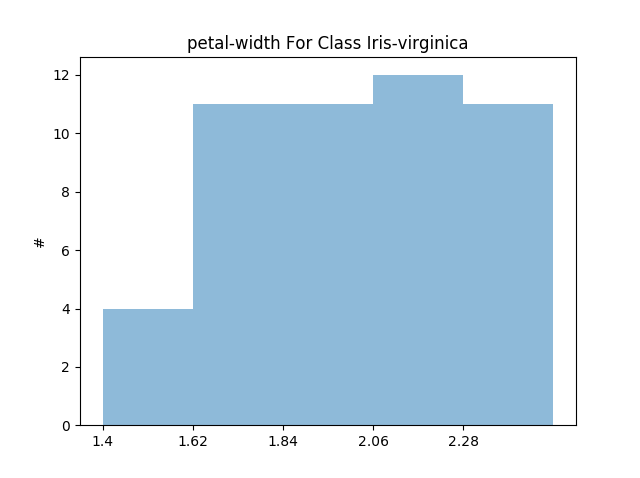
Mostly skewed & mostly multimodal

Bin Size = 100

Mostly skewed & mostly multimodal

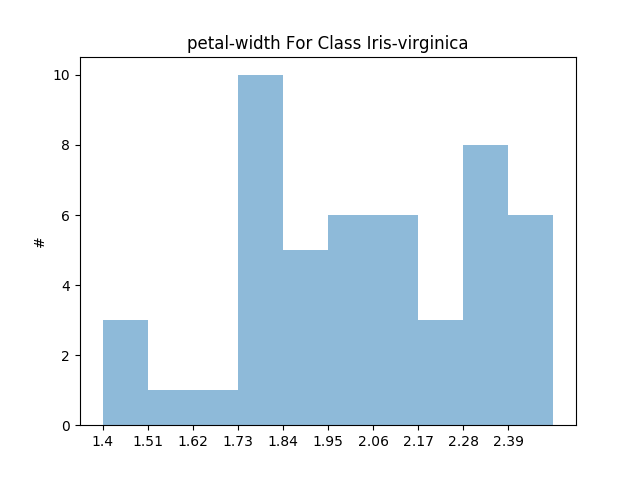
Class 3:

Bin Size = 5



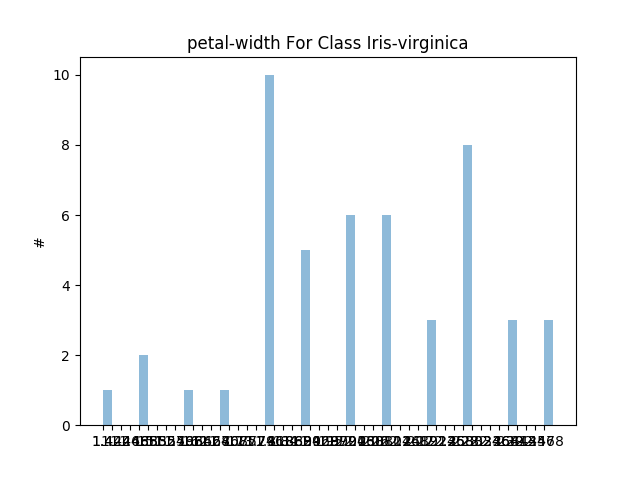
Mostly skewed & mostly uniform

Bin Size = 10



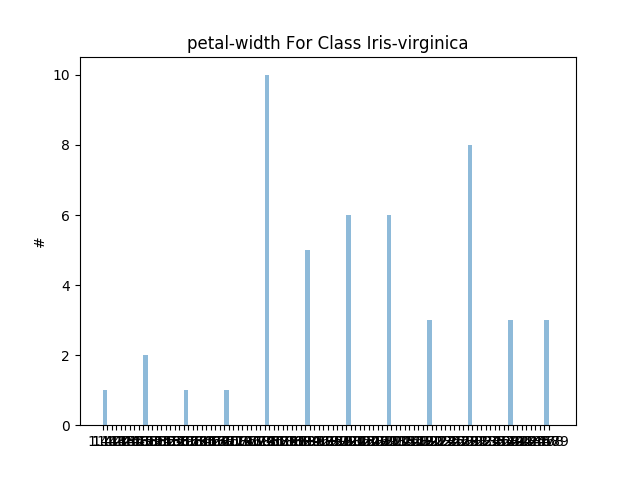
Mostly skewed & mostly multimodal

Bin Size = 50



Mostly skewed & mostly bimodal

Bin Size = 100



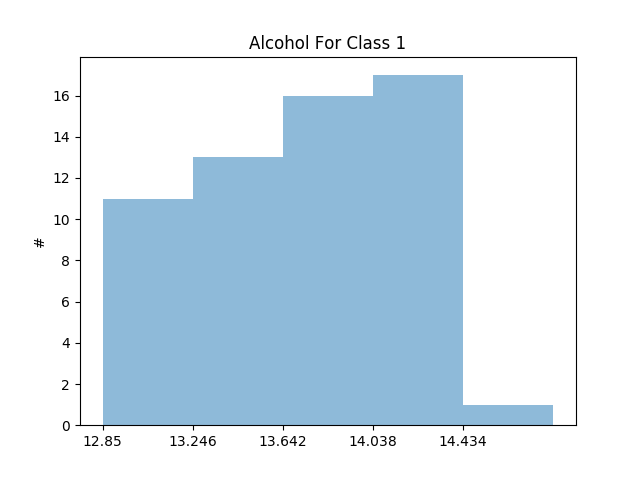
Mostly skewed & mostly bimodal

Dataset = Wine

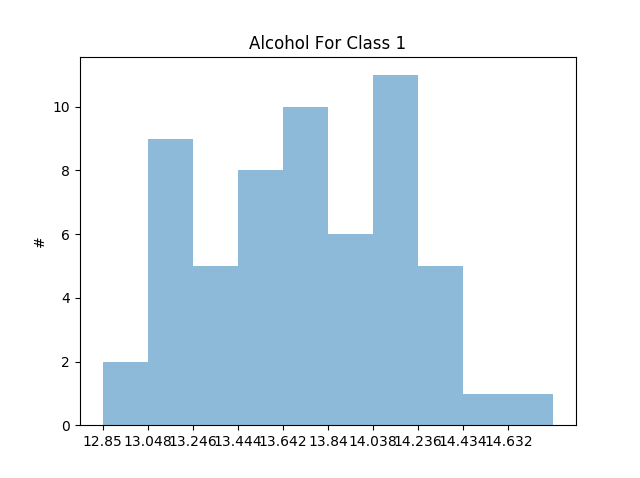
Attribute = Alcohol

Class 1:

Bin Size = 5

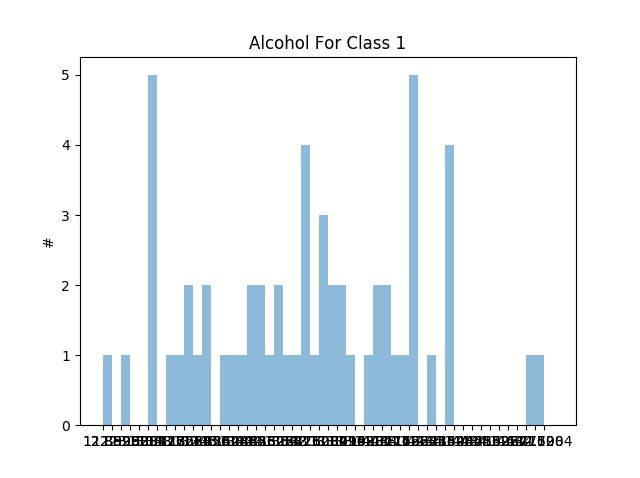


Mostly skewed & mostly uniform

Bin Size = 10

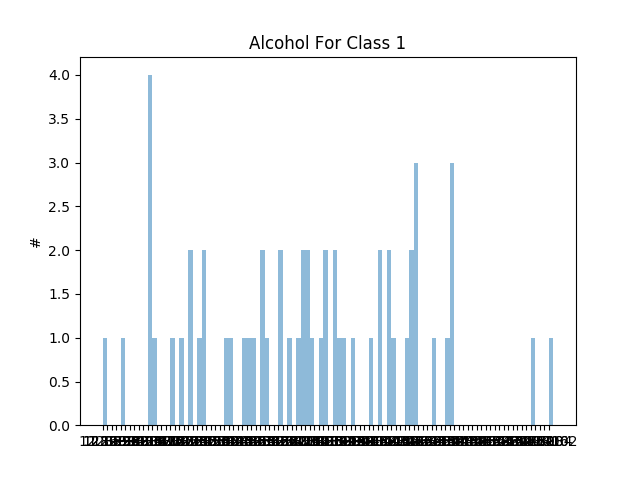
Mostly symmetric & mostly multimodal

Bin Size = 50



Mostly symmetric & mostly multimodal

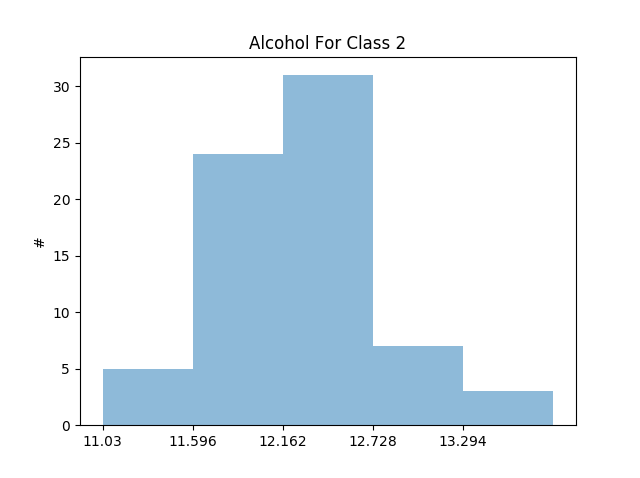
Bin Size = 100



Mostly skewed & mostly multimodal

Class 2:

Bin Size = 5



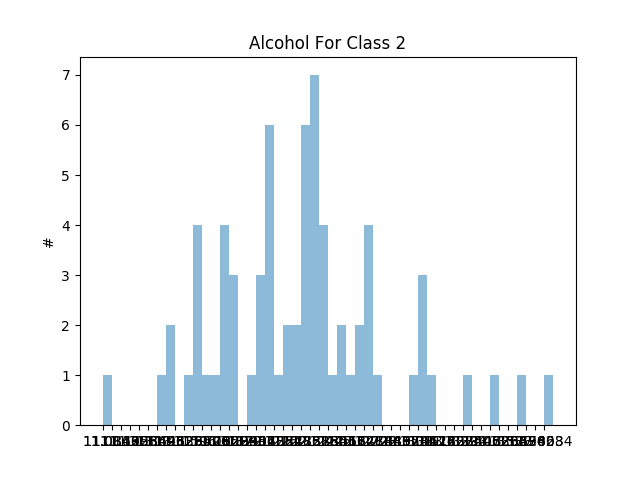
Mostly symmetric & mostly uniform

Bin Size = 10



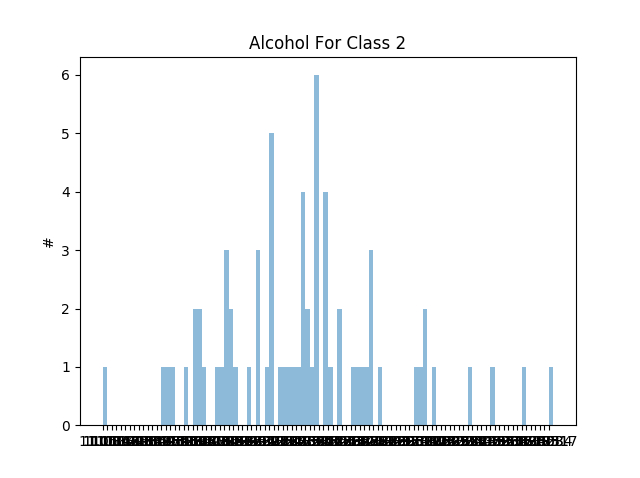
Mostly symmetric & mostly multimodal

Bin Size = 50



Mostly symmetric & mostly multimodal

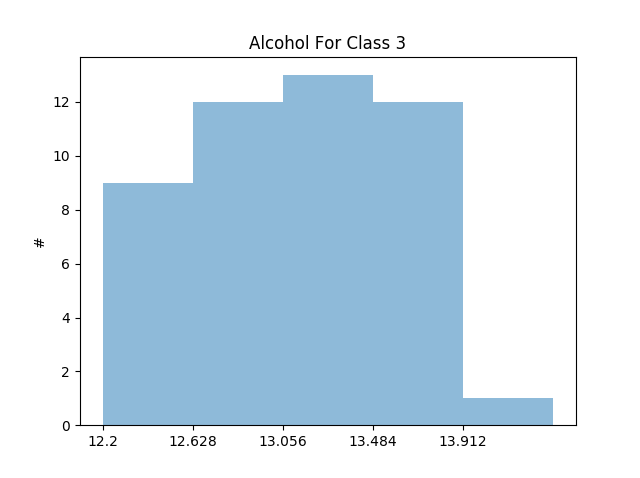
Bin Size = 100



Mostly symmetric & mostly multimodal

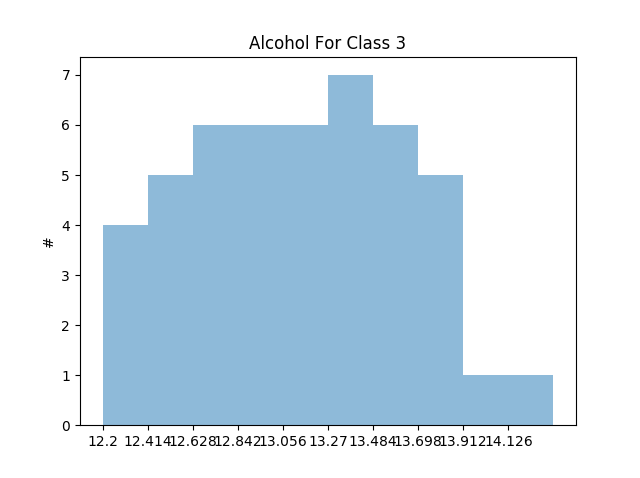
Class 3:

Bin Size = 5



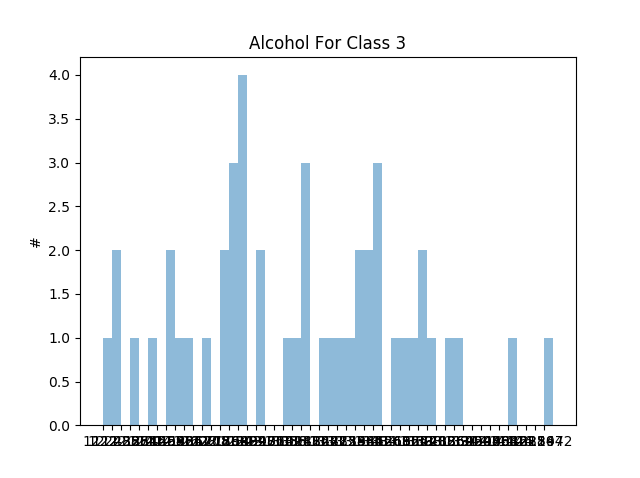
Mostly skewed & mostly uniform

Bin Size = 10



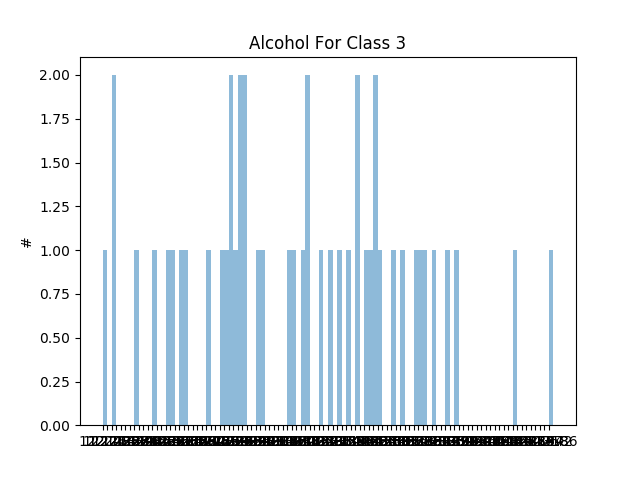
Mostly skewed & mostly uniform

Bin Size = 50



Mostly skewed & mostly multimodal

Bin Size = 100

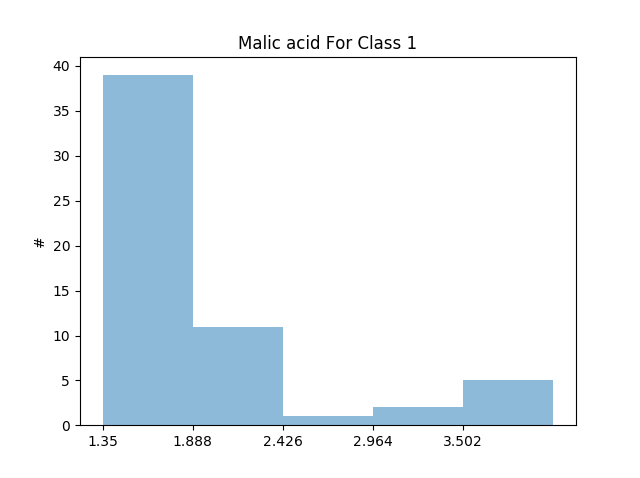


Mostly symmetric & mostly multimodal

Attribute = Malic Acid

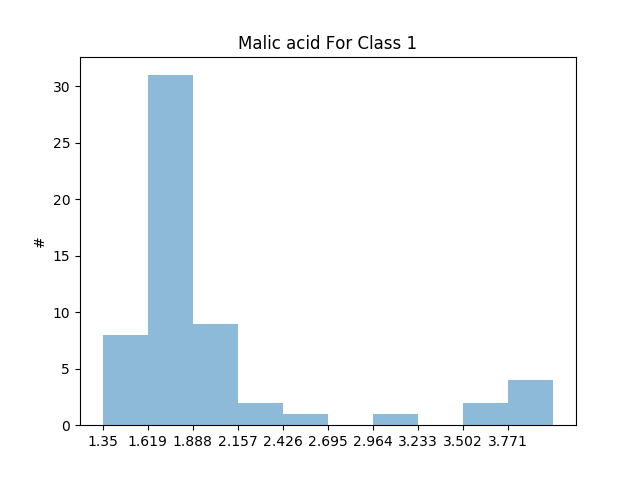
Class 1:

Bin Size = 5



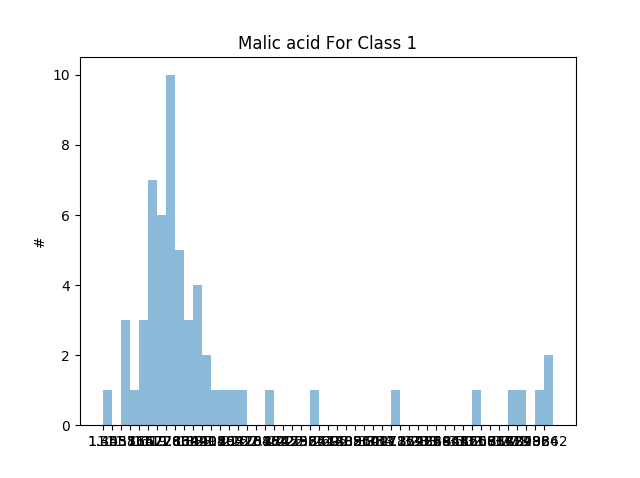
Mostly skewed & mostly unimodal

Bin Size = 10

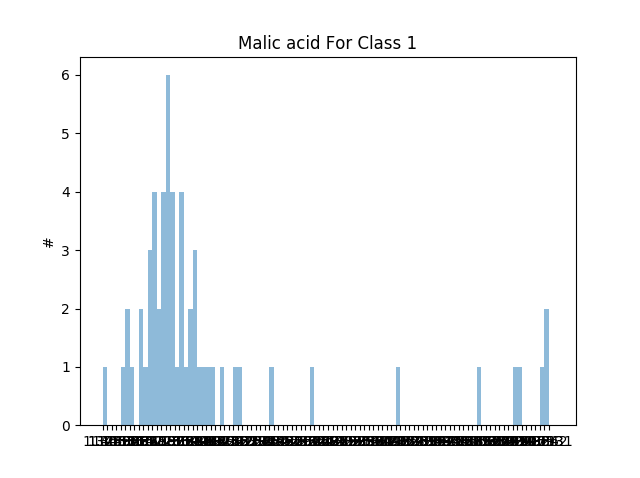


Mostly skewed & mostly unimodal

Bin Size = 50



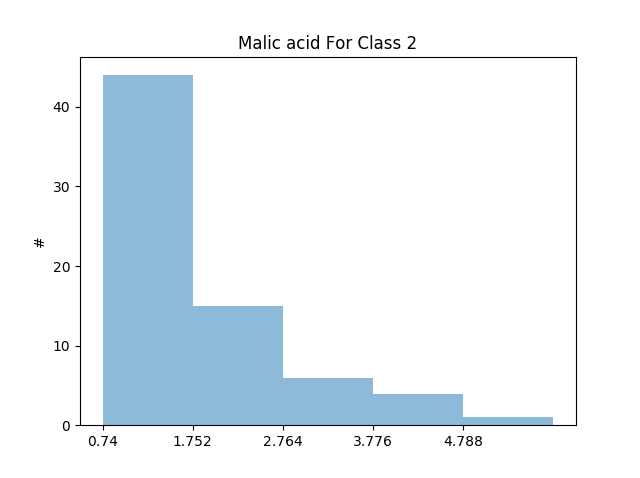
Mostly skewed & mostly unimodal

Bin Size = 100

Mostly skewed & mostly unimodal

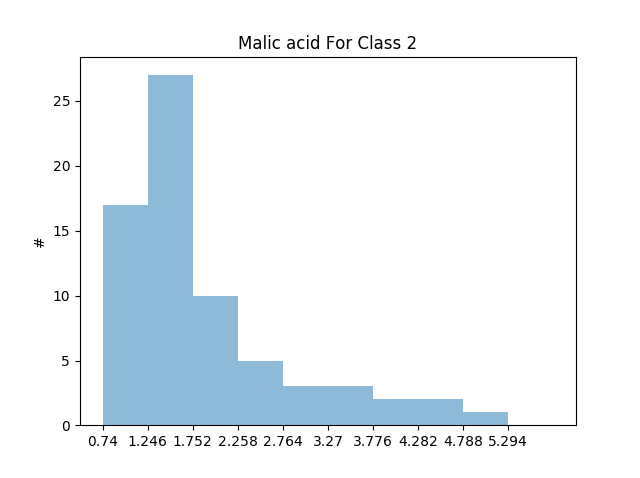
Class 2:

Bin Size = 5

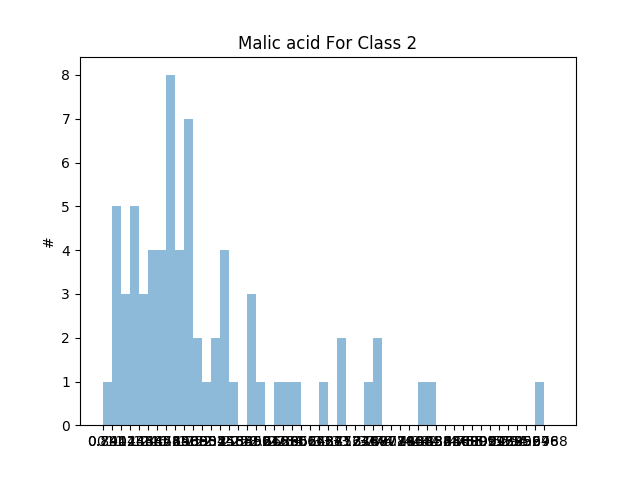


Mostly skewed & mostly unimodal

Bin Size = 10

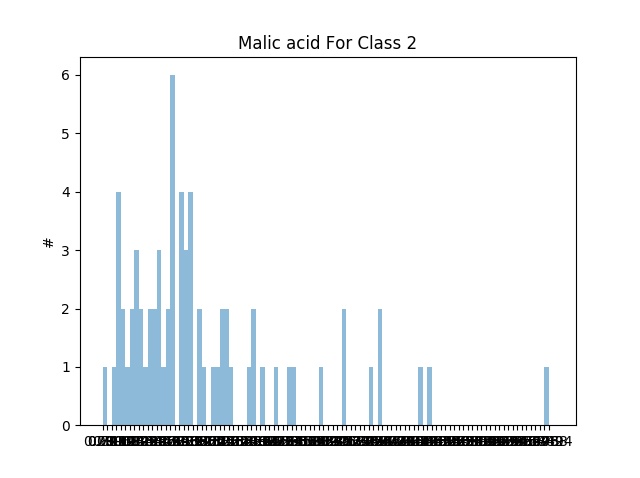


Mostly skewed & mostly unimodal

Bin Size = 50

Mostly skewed & mostly unimodal

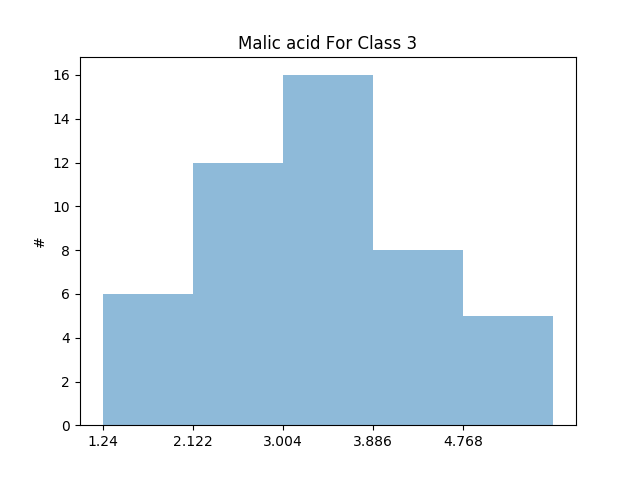
Bin Size = 100



Mostly skewed & mostly bimodal

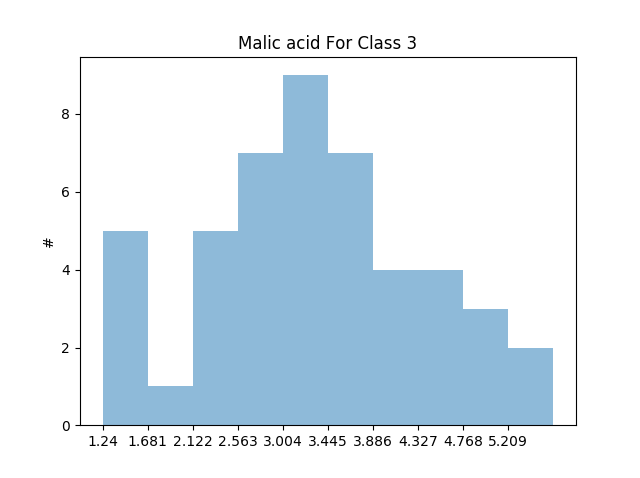
Class 3:

Bin Size = 5



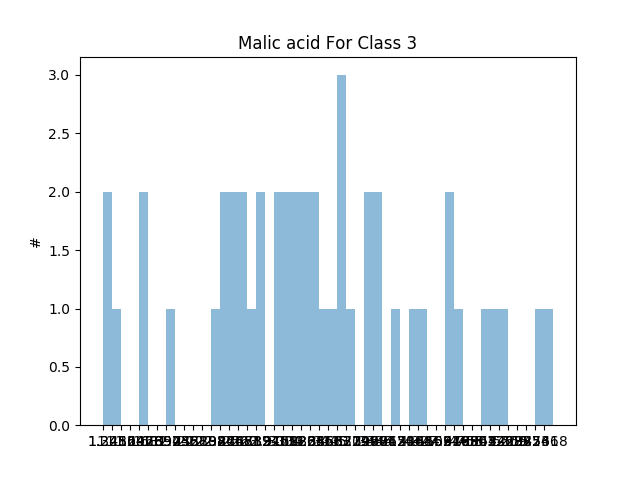
Mostly symmetric & mostly unimodal

Bin Size = 10



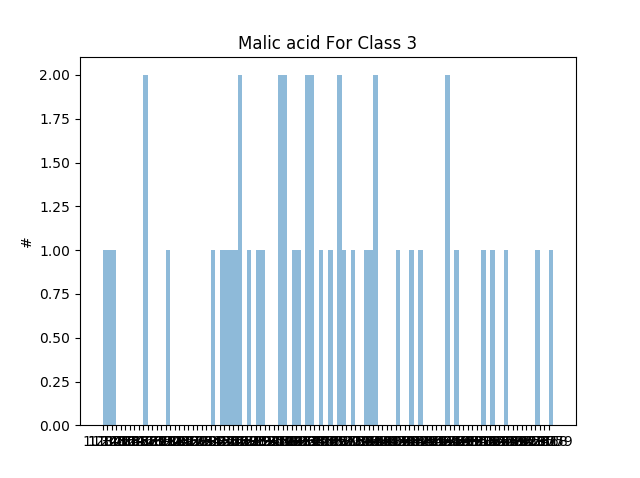
Mostly symmetric & mostly unimodal

Bin Size = 50



Mostly symmetric & mostly multimodal

Bin Size = 100

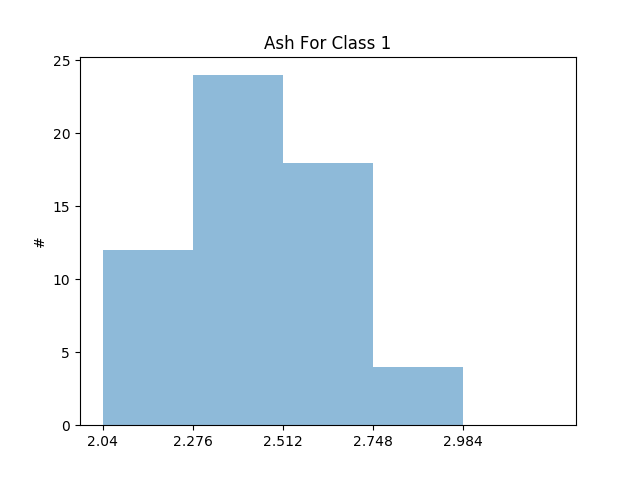


Mostly symmetric & mostly multimodal

Attribute = Ash

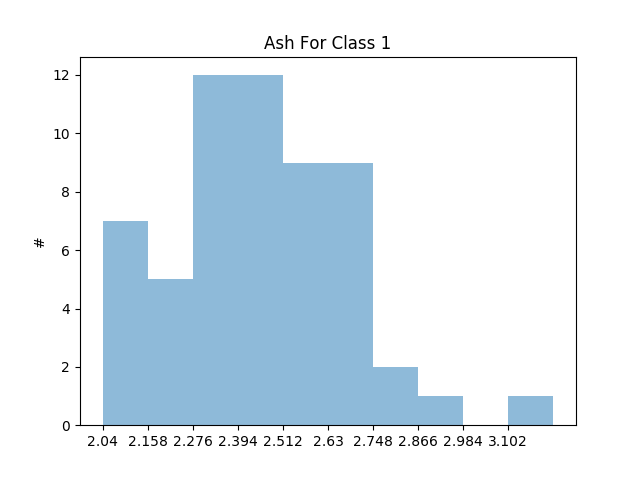
Class 1:

Bin Size = 5



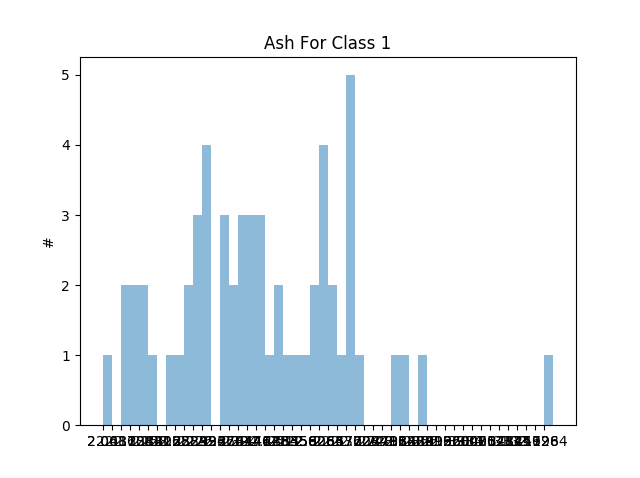
Mostly skewed & mostly unimodal

Bin Size = 10



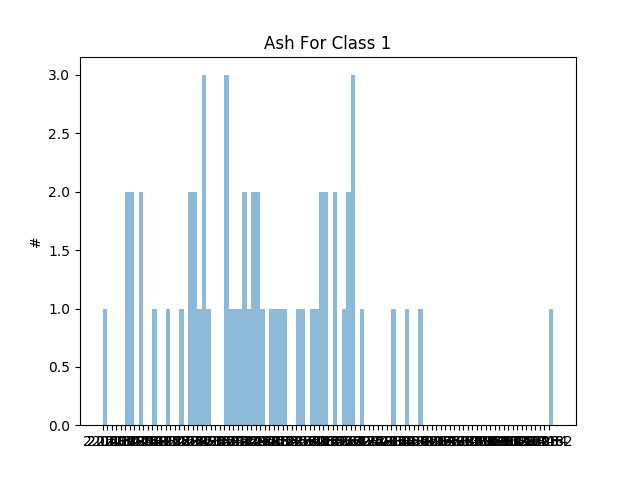
Mostly skewed & mostly unimodal

Bin Size = 50



Mostly skewed & mostly bimodal

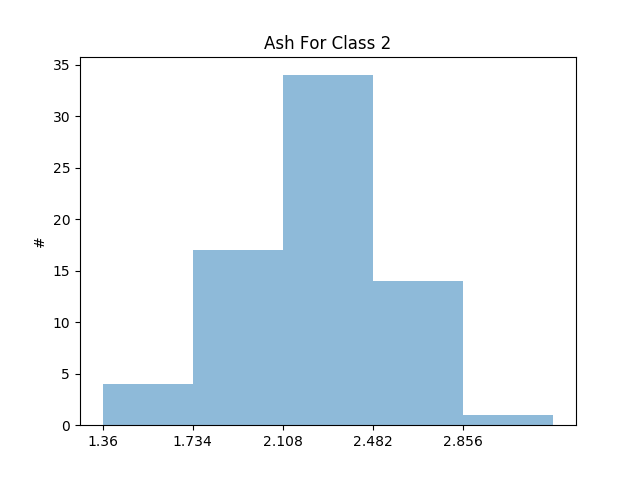
Bin Size = 100



Mostly skewed & mostly multimodal

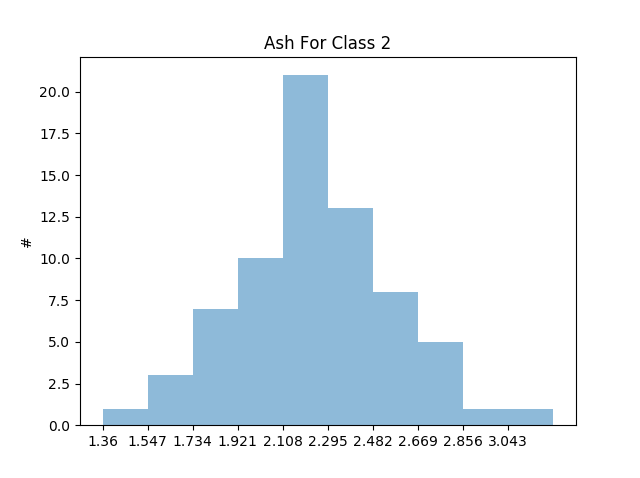
Class 2:

Bin Size = 5



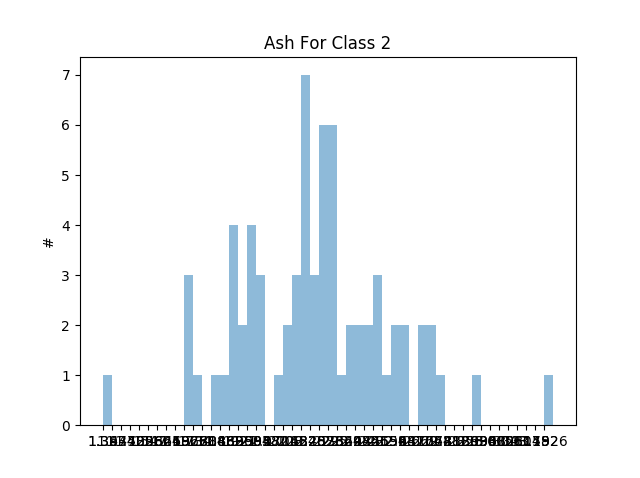
Mostly symmetric & mostly unimodal

Bin Size = 10



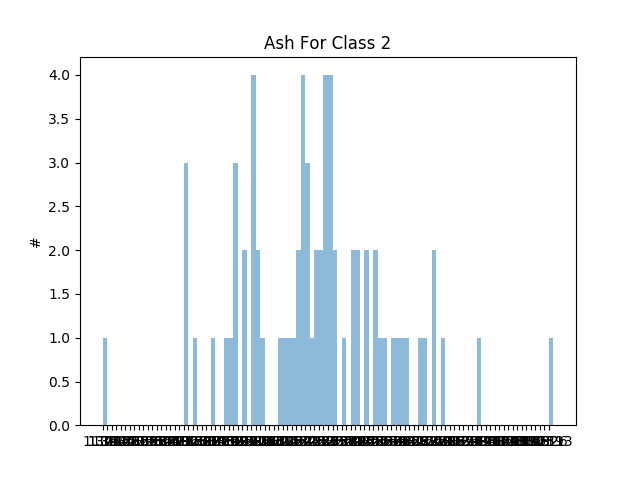
Mostly symmetric & mostly unimodal

Bin Size = 50



Mostly symmetric & mostly unimodal

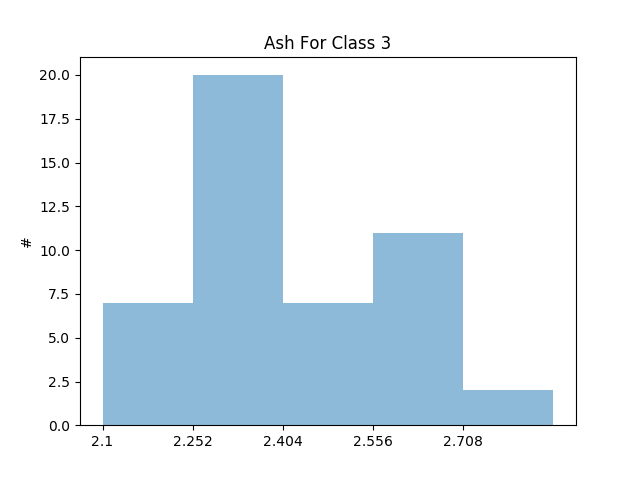
Bin Size = 100



Mostly skewed & mostly multimodal

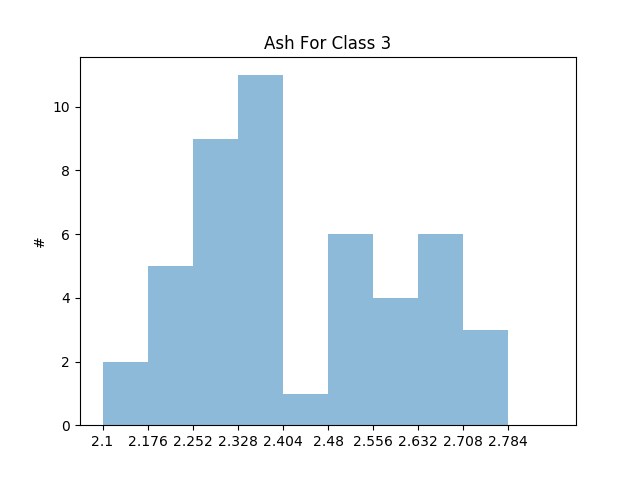
Class 3:

Bin Size = 5



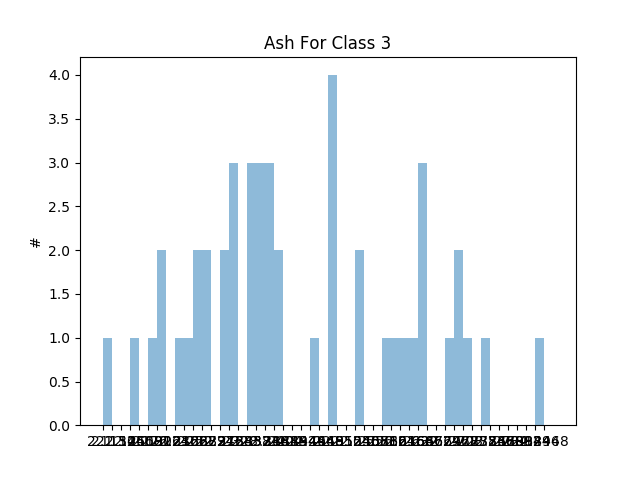
Mostly skewed & mostly unimodal

Bin Size = 10



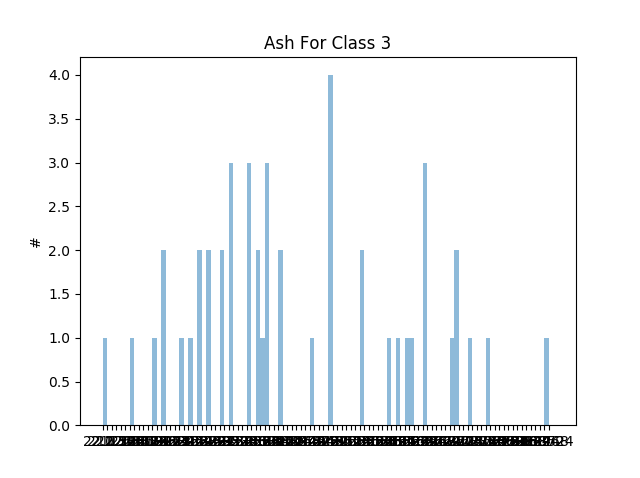
Mostly skewed & mostly multimodal

Bin Size = 50



Mostly skewed & mostly multimodal

Bin Size = 100



Mostly skewed & mostly multimodal

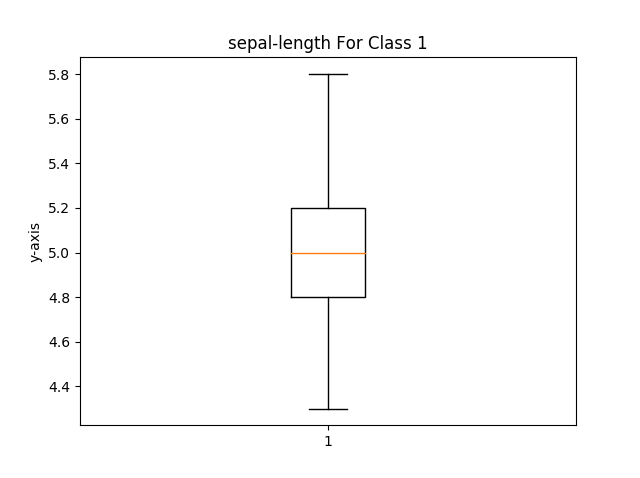
## Box Plots

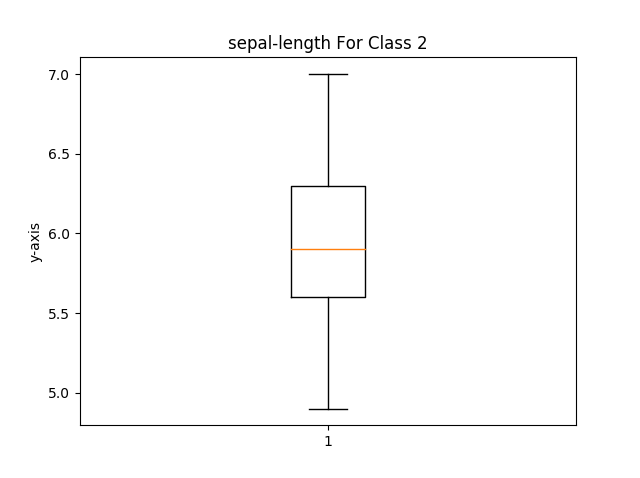
Approach:

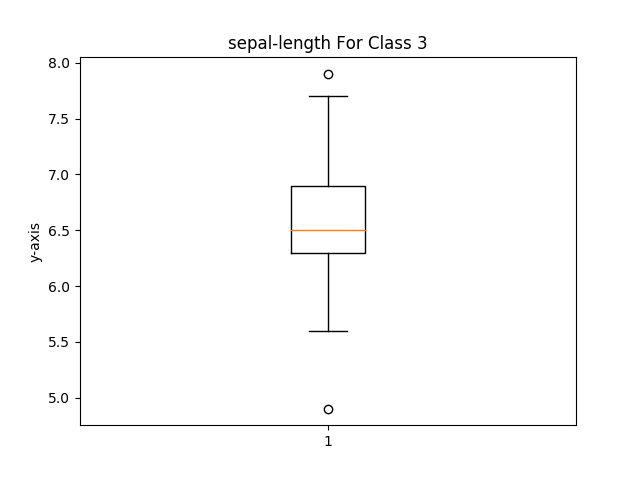
I pass my data into the matplotlib boxplot function

Iris

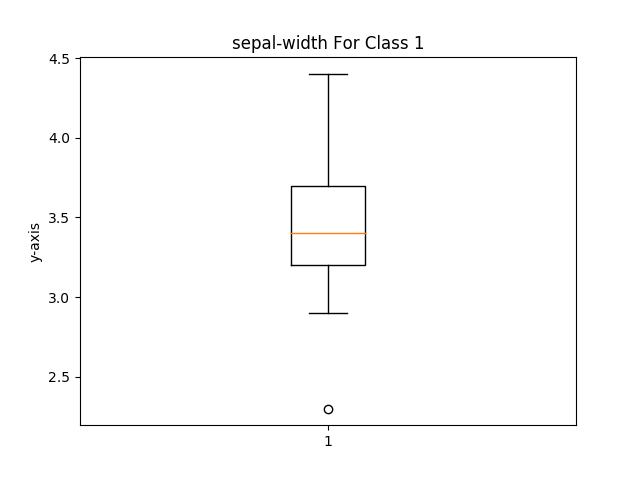
Sepal length

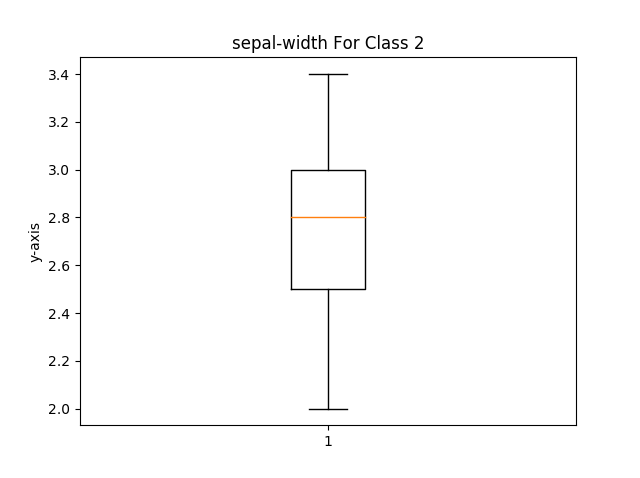
Class 1

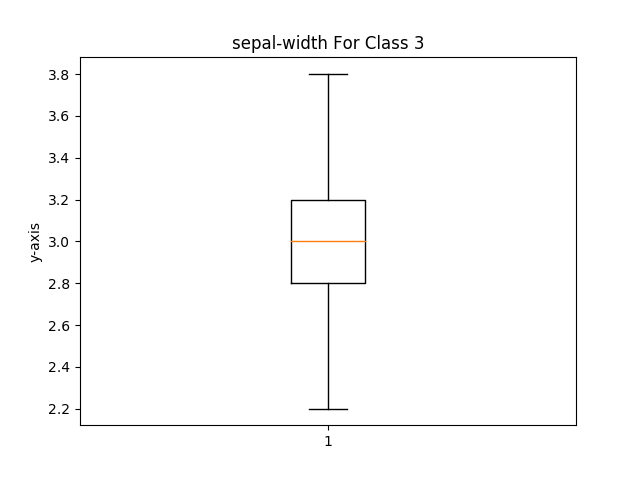
Class 2 

Class 3

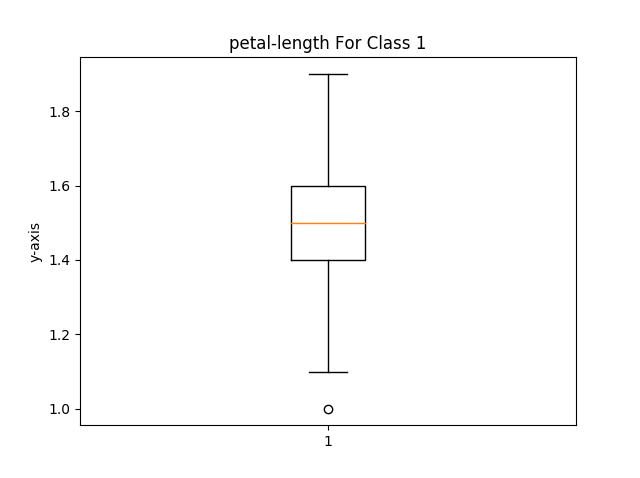
Sepal Width

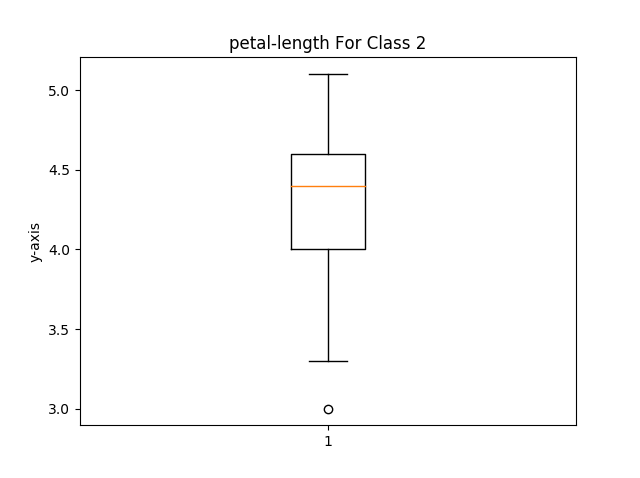
Class 1

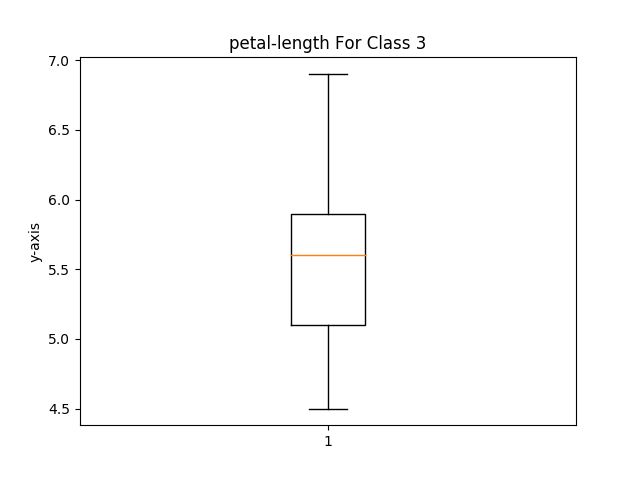
Class 2 

Class 3

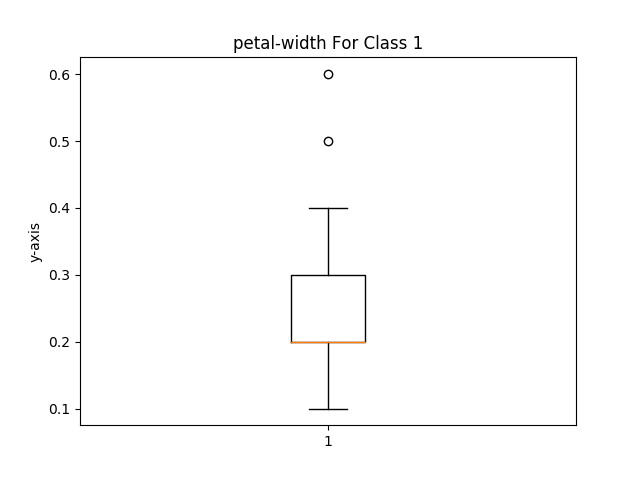
Petal Length

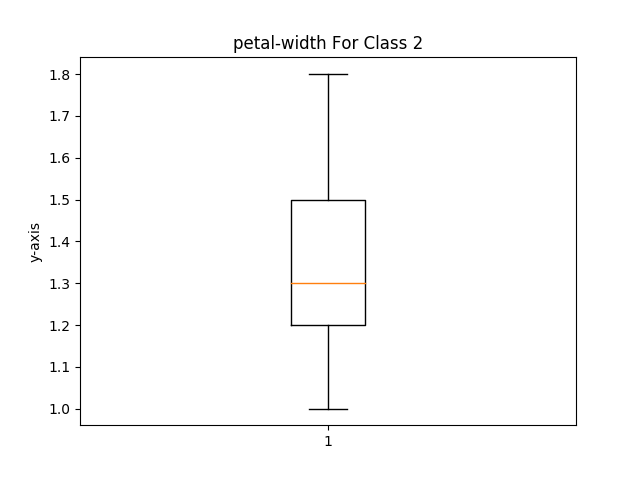
Class 1

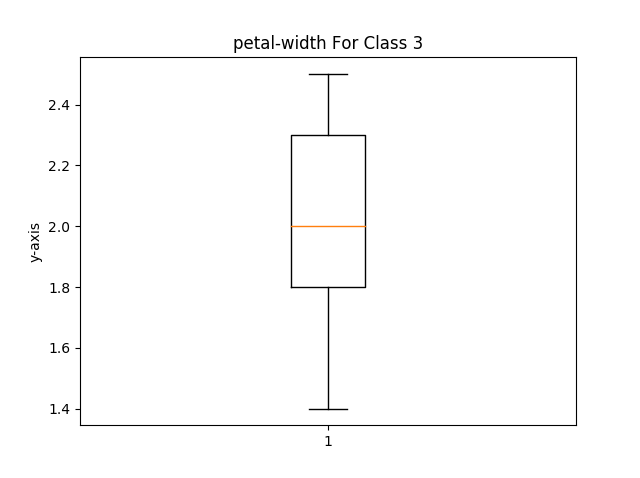
Class 2

Class 3

Petal Width

Class 1

Class 2 

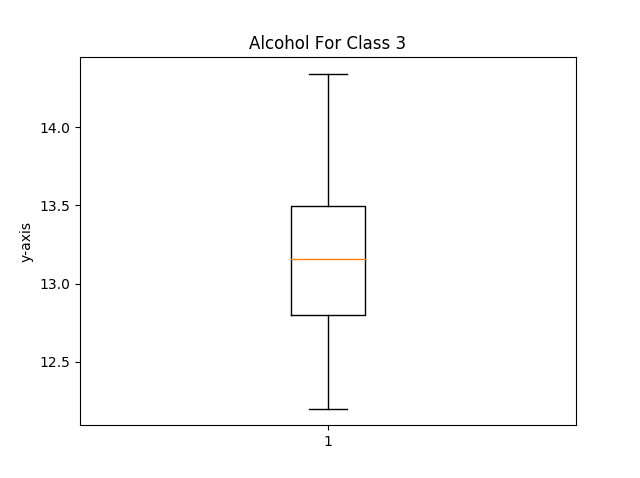
Class 3

Wine

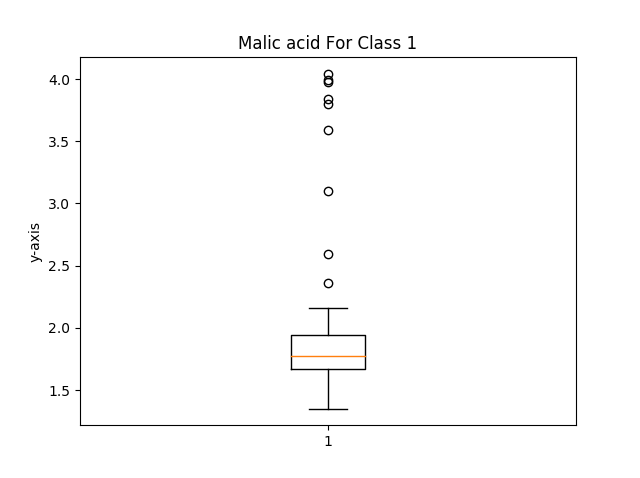
Alcohol

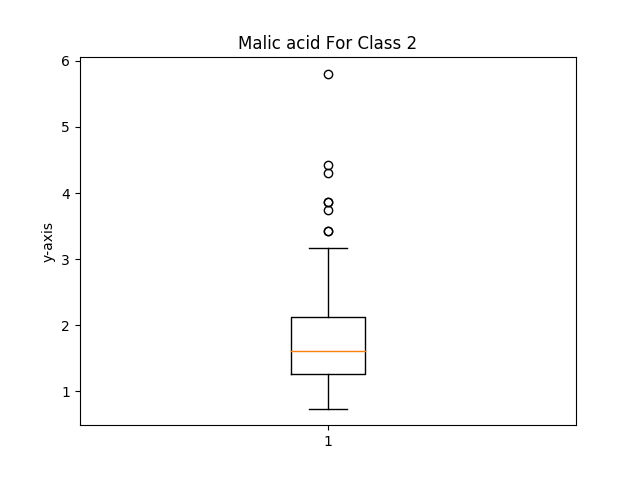
Class 1

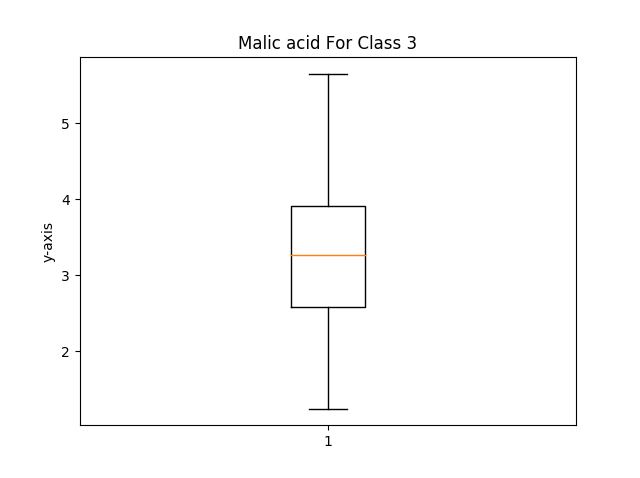
Class 2

Class 3

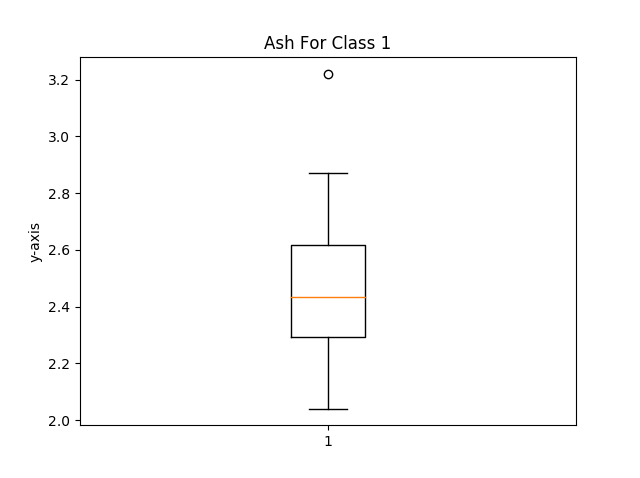
Malic Acid

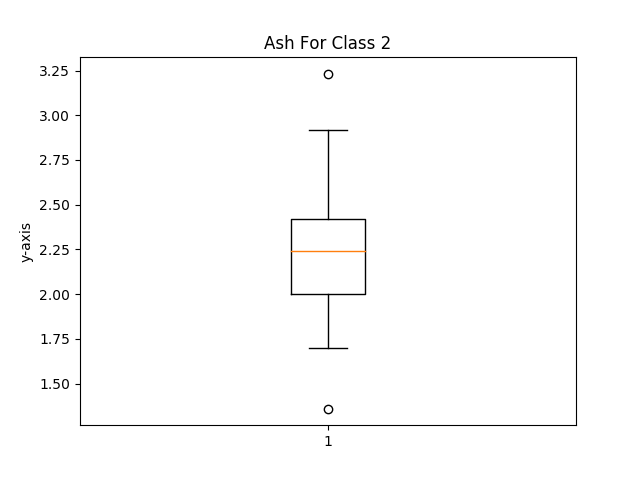
Class 1

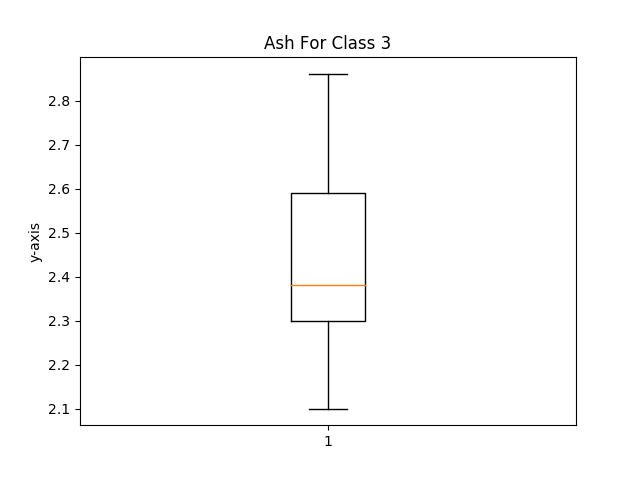
Class 2

Class 3

Ash

Class 1

Class 2

Class 3

## 

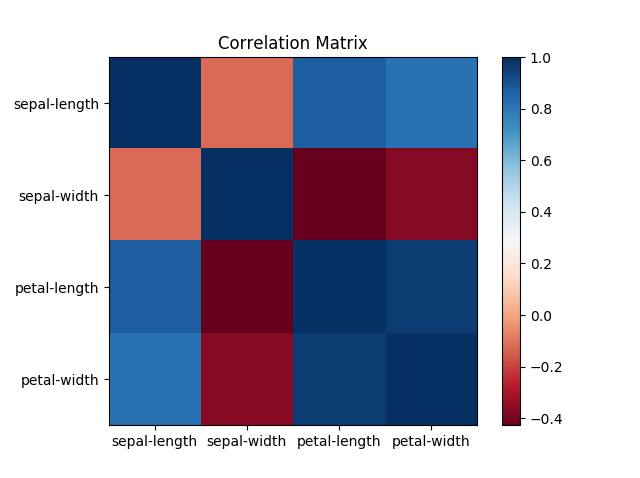
## Correlation Matrices

Approach:

I started with the correlation function, I had originally used the np.std() to get the standard deviation but it wasn’t giving me enough numbers pass the decimal so I had to code that myself. Next I made my corMatrix() function. It uses a nested for loop to pass the rows into the correlation() function. I had to use the list.pop() function as well as the np.delete() function to fix the wine data set so that it wouldn’t show the correlation of the class type.

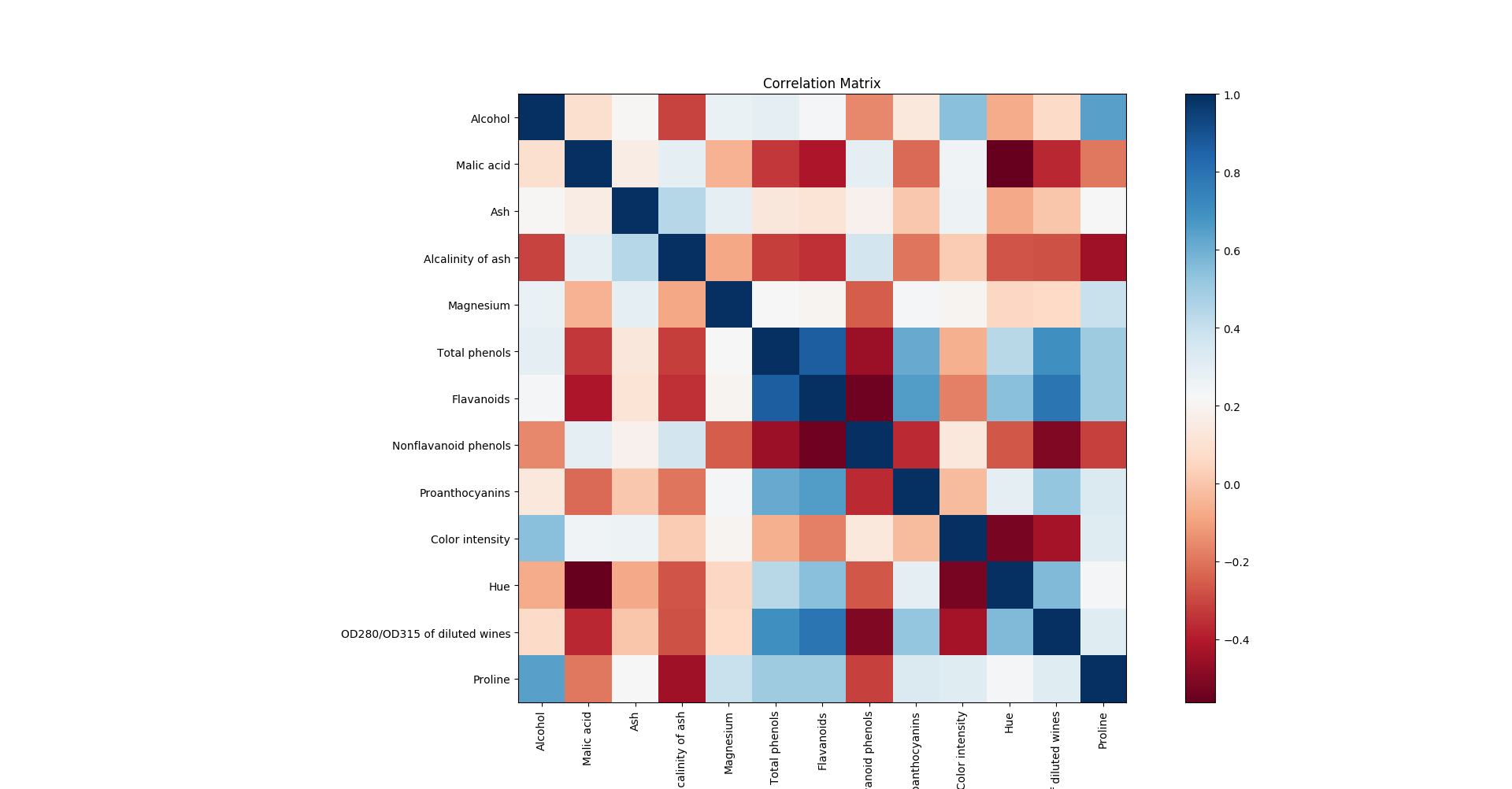
These are taking all three classes into account.

Iris



Petal-length and petal width are strongly correlated while sepal-width and both petal-length and petal-width are negatively correlated. Sepal-length and sepal-width have no correlation. This can be useful for someone breeding iris flowers, if the desired attribute is large petals, then the breeder should avoid a wide sepal.

Wine



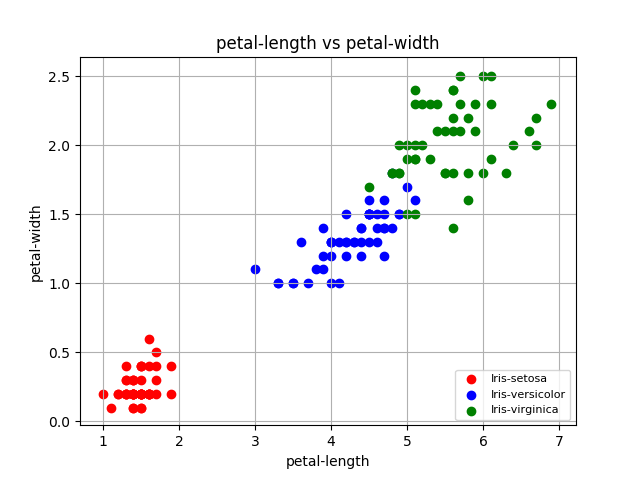
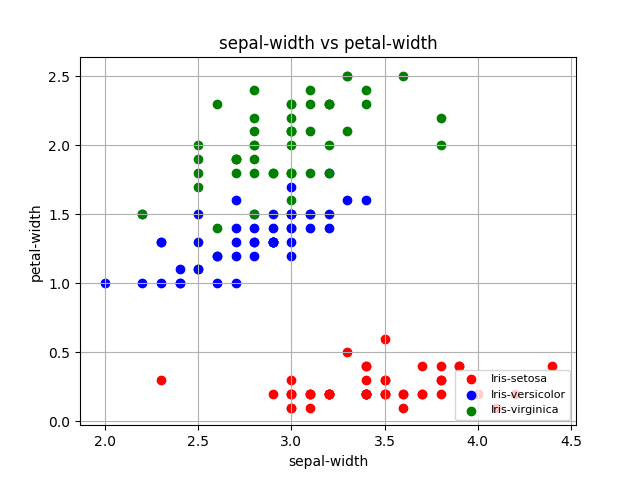
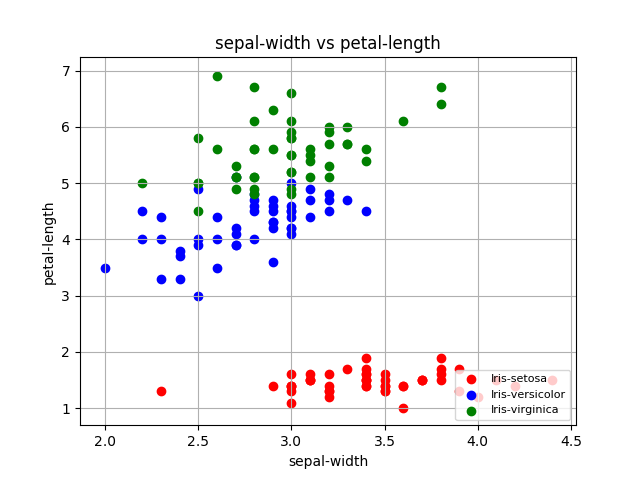
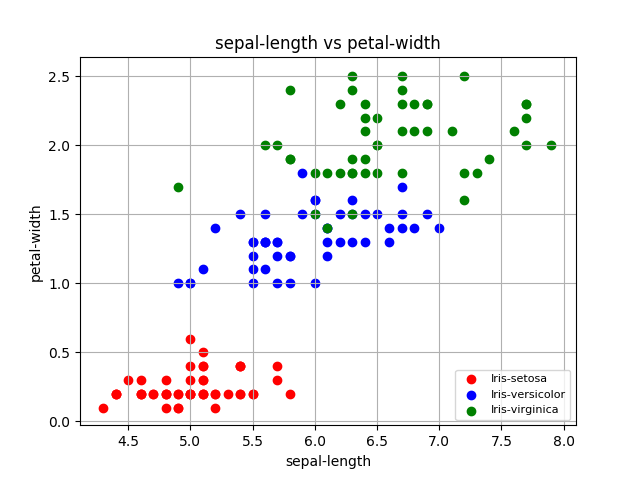
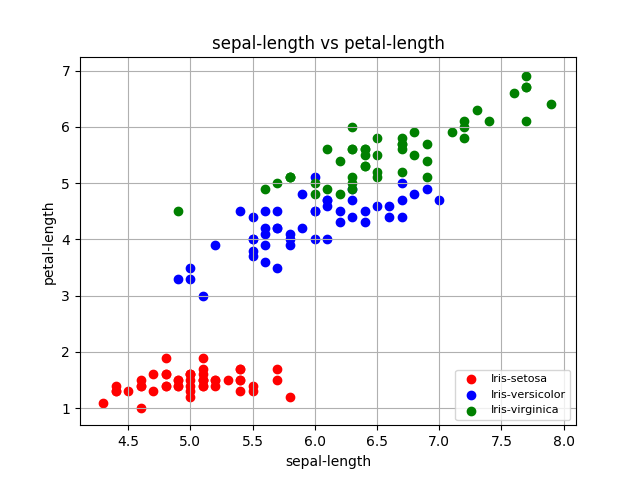
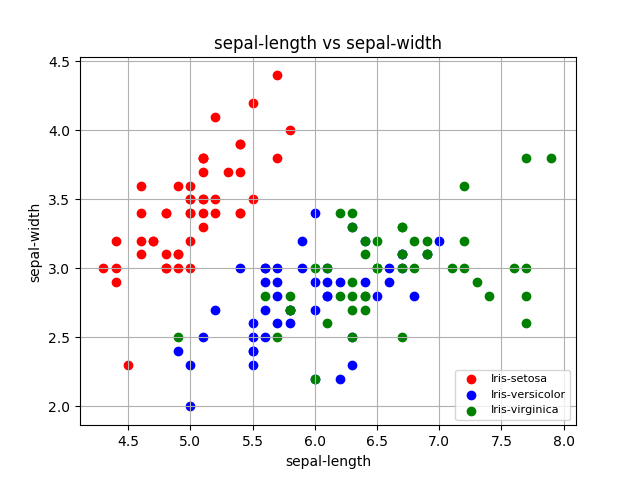
The first 5 attributes seem to have little correlation with any other factors, the others seem to have some correlation with a few exceptions such as Flavanoids and Nonflavanoid phenols having a fairly negative correlation as well as color intensity and hue being strongly negatively correlated.

The absolute minimum number of calls to the correlation function is (n)(n-1)/2 with n being the number of attributes. This is because the correlation between an attribute with itself is always one, and the correlation between two different attributes is equal to its inverse. So the first row needs to call n-1 times, followed by the second row call the function n-2 times. So by taking the sum of consecutive increasing numbers (which is (n)(n+1)/2) and taking away an n from the equation because you start with n-1 and not n.

## Scatter Plots

Approach:

Using the plt.scatter() function to graph specific subsets of the function passed in. The only part that was difficult was getting the legend to work.



All features seem to be discriminative towards the Iris-setosa, it seems that the Iris-setosa is much smaller than the other two classes in every dimension except sepal-width. This size difference makes it very easy to distinguish it from the other two classes.

## Distance

This accounts for all classes, also not that (0,0) is in the top left corner

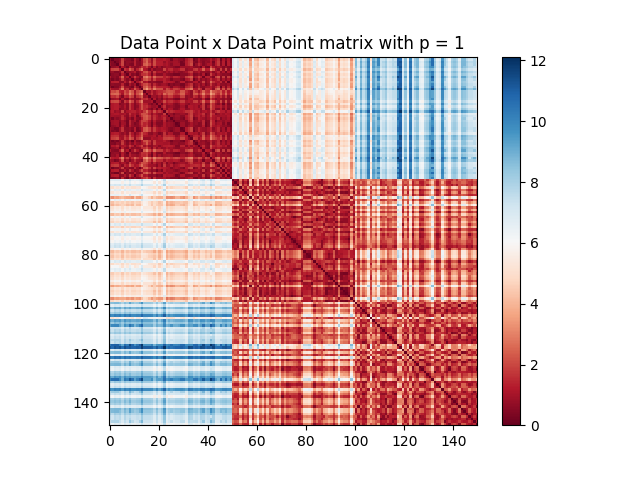
Approach:

I started with the distance() function. I had originally made the mistake of using \*\*(1/p) which made the program return 1 whenever p was greater than 1 since p was an integer. Eventually I realized that I needed to cas p to a float.

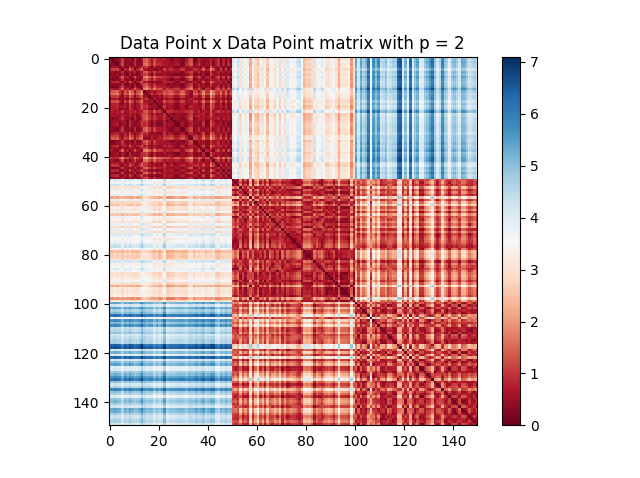
The dpMatrix() starts with a nested for loop that calls the distance function and fills out my variable called dpMat. The difficult part of this was part e of the problem of finding the non-trivial nearest data point. I created a second nested for loop that iterates through dpMat and fills out a 2D np array called ntndp which stands for Non Trivial Nearest Data Point. I have quite a bit of code that isn’t used as I had assumed the assignment had wanted me to print out a list of the nearest neighbors. So now it only prints the percentages asked for.

Iris

P = 1

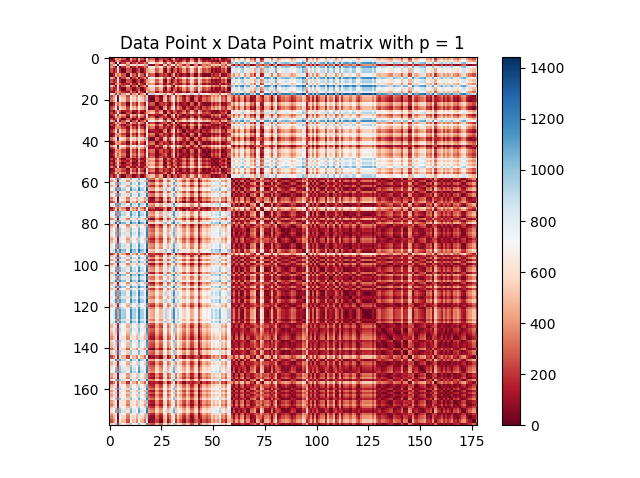


P = 2

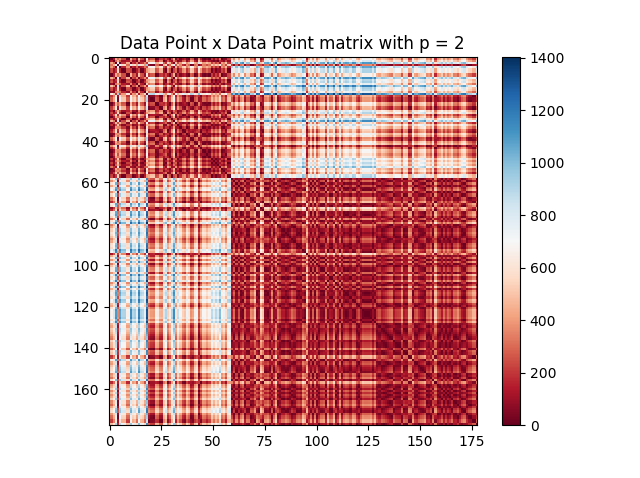


Wine

P = 1



P = 2



The absolute minimum number of calls to the distance function is the same as the correlation function (n)(n-1)/2.

For the nearest data point being from a different class:

Iris with p = 1

Percentage of nearest data points being from different classes 4.66666666667%

Iris with p = 2

Percentage of nearest data points being from different classes 4.0%

Wine with p = 1

Percentage of nearest data points being from different classes 14.0449438202%

Wine with p = 2

Percentage of nearest data points being from different classes 23.0337078652%

For when p changes the nearest data point:

Iris

Percentage of nearest data points changing when p changes 19.3333333333%

Wine

Percentage of nearest data points changing when p changes 17.9775280899%