Contents

[Summary of Day 2 2](#_Toc96464409)

[Functions Cheatsheet ( Day 2 ) - 2](#_Toc96464410)

[Some additional functions – 3](#_Toc96464411)

[Changing the default figure size- 3](#_Toc96464412)

[Pyplot Styles 3](#_Toc96464413)

[Pandas Visualizations 3](#_Toc96464414)

[Importing a sample dataset 3](#_Toc96464415)

[Iris Dataset 4](#_Toc96464416)

[DataFrame.plot 5](#_Toc96464417)

[Boxplot 7](#_Toc96464418)

[Histogram 7](#_Toc96464419)

[KDE Plot 7](#_Toc96464420)

[Advanced Panda tools - pandas.plotting 8](#_Toc96464421)

[Scatter Matrix 8](#_Toc96464422)

[Parallel Coordinates 9](#_Toc96464423)

[Seaborn 9](#_Toc96464424)

[Introduction 9](#_Toc96464425)

[Distplot 10](#_Toc96464426)

[Jointplot 10](#_Toc96464427)

[Pairplot 13](#_Toc96464428)

[Swarmplot and Violinplot 13](#_Toc96464429)

[CountPlot 14](#_Toc96464430)

[Bar Plot 15](#_Toc96464431)

[Box Plot 15](#_Toc96464432)

[Correlation plot using Heatmap 16](#_Toc96464433)

[Spurious Correlations 17](#_Toc96464434)

[Scatter Plot, Line Plot 19](#_Toc96464435)

[Editing Charts for Seaborn plots 19](#_Toc96464436)

[Changing chart parameters using Pyplot 20](#_Toc96464437)

[Changing chart parameters using Axes Object- 20](#_Toc96464438)

[MCQ (10 minutes)- 21](#_Toc96464439)

[Homework 22](#_Toc96464440)

# Summary of Day 2

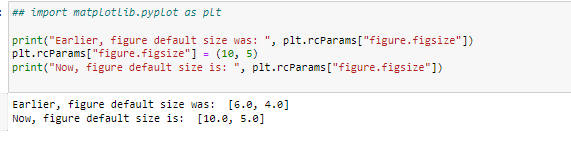
## Functions Cheatsheet ( Day 2 ) -

* Plotting functions -
  + Scatter Plot - plt.scatter()
  + Linechart - plt.plot()
  + Barchart - plt.bar()
  + Piechart - plt.pie()
  + Histogram - plt.hist()
  + Boxplot - plt.boxplot()
  + Heatmap - plt.hist2d(), plt.imshow()
* Functions for editing chart objects -
  + plt.xlabel('X-label') # For X-axis label
  + plt.ylabel('Y-label') # For Y-axis label
  + plt.title('A random scatter chart for Practice') # For Chart label
  + plt.xticks(x\_int,rotation = -5) # For managing x-axis ticks
  + plt.grid() # For toggling between grid on/off for x/y axes
  + plt.savefig('first\_matplotlib\_plot.png') # Saves the image as png in Jupyter library
  + plt.legend() # Adds legend
  + plt.colorbar() # Adds legend in heatmap
* Functions associated with Axes object -
  + plt.gca() # Gets the current axes
  + .get\_children() # Shows children objects of an axes object
  + .set\_title('Chart Title set by Axes Object')
  + .set\_xlabel('X Label set by Axes Object')
  + .set\_ylabel('Y Label set by Axes Object')
  + .plot() # Plots a line chart for a given axes object
  + .scatter() # Plots a scatter chart for a given axes object
  + .bar() # Plots a bar chart for a given axes object
* Subplots
  + fig, ax = plt.subplots(2, 2, sharex=True) # Assign figure object to fig and Axes objects to ax
  + plt.subplot(x,y,z) # Selecting subplot z of a x\*y subplot matrix

## Some additional functions –

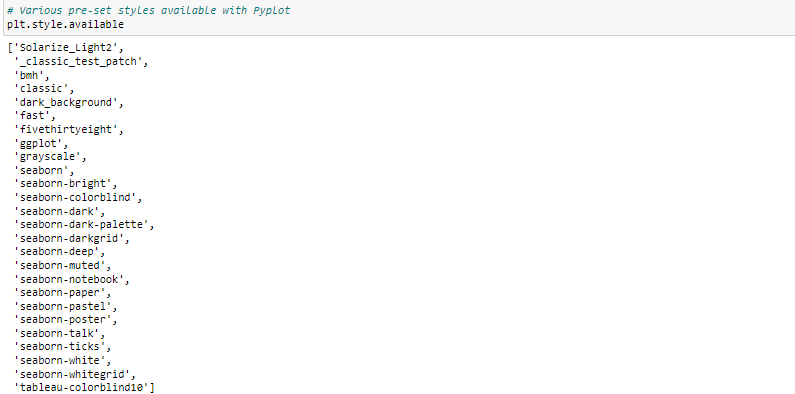
### Changing the default figure size-

We can configure the default figure size using rcParams



### Pyplot Styles

Style Reference link - <https://matplotlib.org/stable/gallery/style_sheets/style_sheets_reference.html>

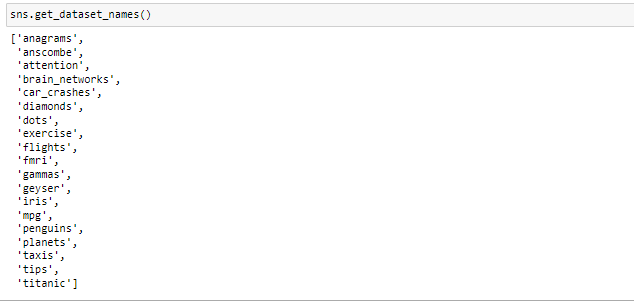




# Pandas Visualizations

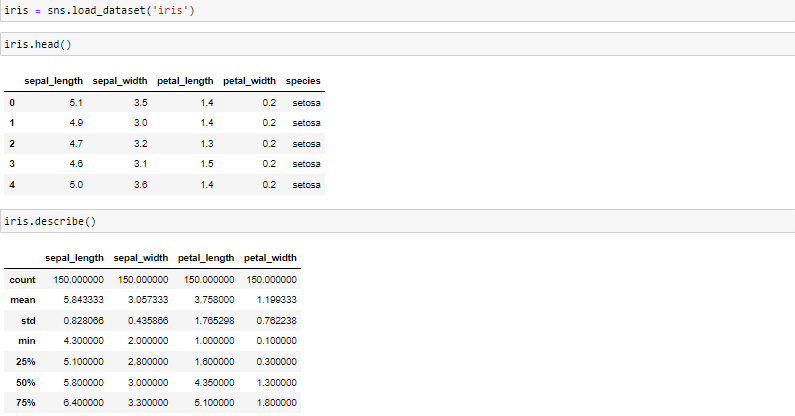
### Importing a sample dataset

Seaborn library allows us to load some great example datasets for practice. Let's have a look at the various options that we get using get\_dataset\_names() function. We'll be working mostly with "Iris" dataset in this notebook.



### Iris Dataset

It is a dataset of 3 different flowers of Iris species regarding their physical characteristics. The distribution mix of these 3 flowers is same – i.e. 50-50 datapoints each. Let's load the dataset and look at the data using head() function



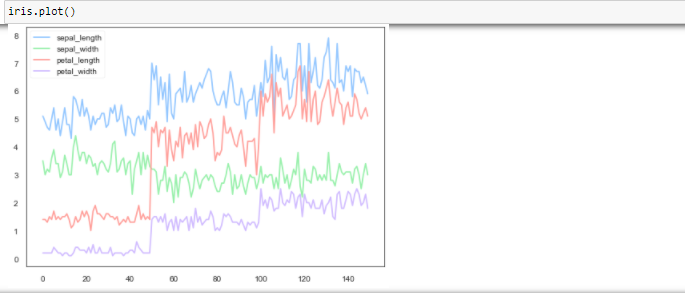
We can see that the dataset has 5 variables -

1. sepal\_length : length of the sepal of flower ( Range 4.3 -7.9 )
2. Sepal\_width : width of the sepal of flower ( Range 2 -4.4 )
3. petal\_length : length of the petal of flower ( Range 1 -6.9 )
4. petal\_width : width of the petal of flower (0.1 - 2.5)
5. species : Name of the flower species ( Categorical , 3 levels )

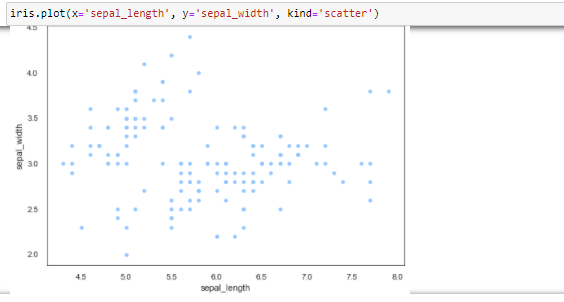
## DataFrame.plot

.

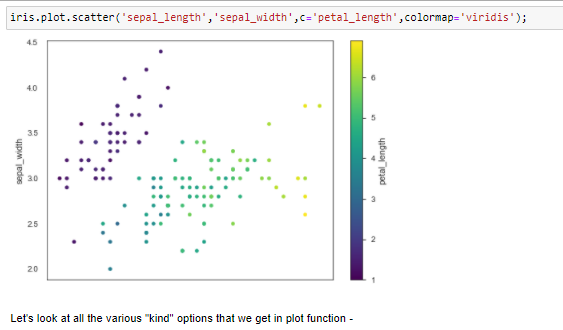
If nothing is specified, the functions plots all the columns by default as a linechart –



scatter plot by providing "scatter" in kind parameter-



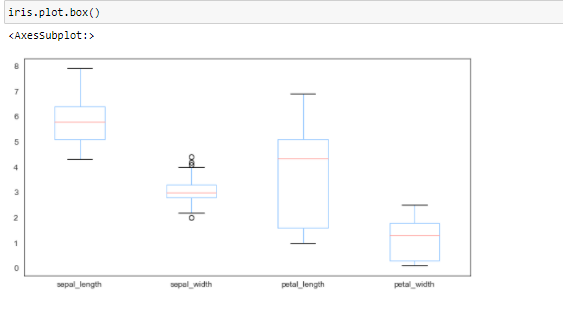
Alternatively, we can make the same scatter plot by using df.plot.scatter function.



kind parameter options :

* 'line' : line plot (default)
* 'barh' : horizontal bar plot
* 'hist' : histogram
* 'pie' : pie plot
* 'scatter' : scatter plot
* 'hexbin' : hexbin plot
* 'box' : boxplot
* 'kde' : Kernel Density Estimation plot
* 'density' : same as 'kde'
* 'area' : area plot
* 'bar' : vertical bar plot

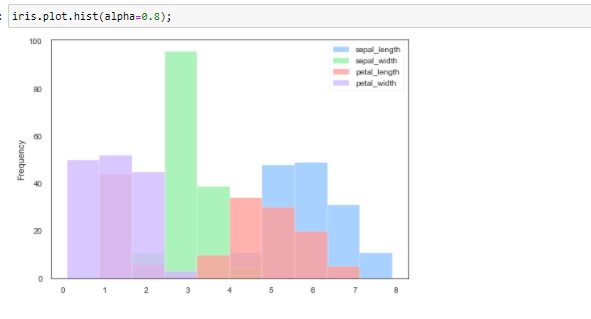
### Boxplot



Inferences –

* Sepal Length has the highest mean and takes higher range of value while Petal width the lowest.
* Petal length has the flattest distribution and a high range, while Sepal width is concentrated around its mean, thus more outliers.

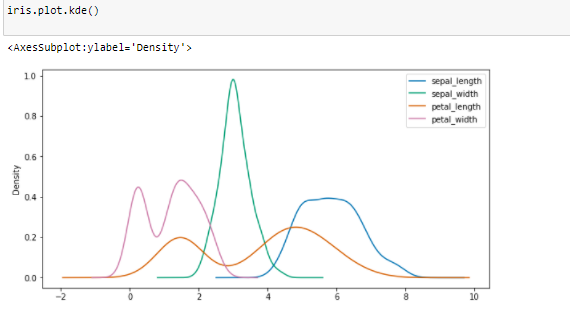
### Histogram



### KDE Plot

KDE stands for Kernal Density Estimate.

It plots an estimate of the Probability distribution sample of a variable. This graph has great application in statistics and feature engineering.



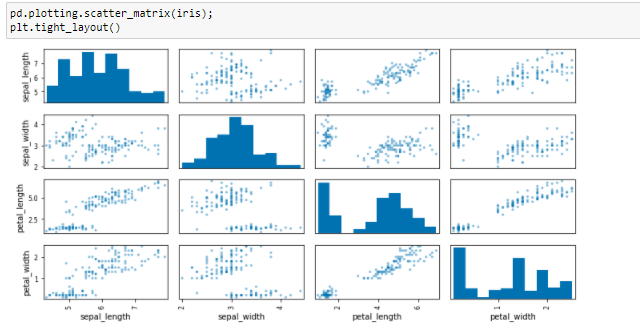
Inferences -

* Sepal length takes highest values compared to the 4 variables, meaning Sepal length is longest by absolute value in general. Opposite is true for Petal Width
* Most plants have more or less similar Sepal Width (Close to it mean)
* Petal length’s range is maximum and this can be a major determining factor in the identity of a flower from another

## Advanced Panda tools - pandas.plotting

### Scatter Matrix

A scatter matrix visualises each variable in the dataset with another, in a pairwise fashion -



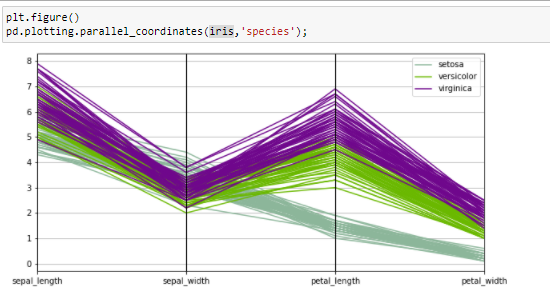
Pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters.

This is extremely useful in feature selection or rejection because we can see all the patterns among data instantly. It is also used to choose best set of variables for a classification problem.

### Parallel Coordinates

Parallel coordinates are another great way to analyse multivariate data and find patterns.

Here, each of the variable is represented as parallel lines and data points are plotted and connected. The color coding helps us visualise the data points with respect to a target categorical variable.



Parallel coordinates are similar to time-series plots except that there is no time on the x-axis and there is no linear relationship among points on x-axis.

These have applications to visualise very high dimensional data.

# Seaborn

## Introduction

Seaborn is the most used visualization library in analytics world. It is built on top of Matplotlib, so if anyone is proficient in Matplotlib, Seaborn is just a set of new functions to learn.

Although Matplotlib is a powerful library, it has 2 main drawback that Seaborn resolves -

* Default Parameters
* Advanced support for DataFrames

Seaborn not only resolves both of these, but also have following benefits -

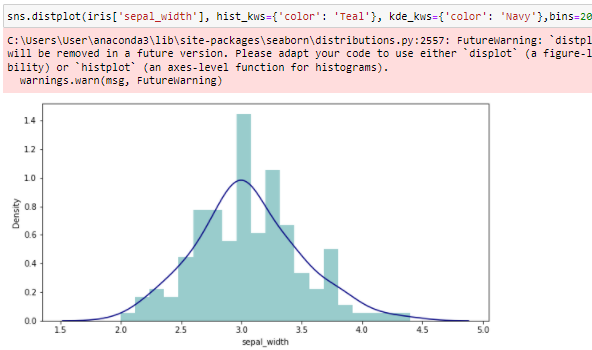
* Support for Pre-defined themes
* Great support for some additional chart useful for Univariate and Bivariate data
* Visualization of Linear Regression models
* Plotting time-series data

## Distplot

Distplot is used for checking the distribution of a continuous variable. It is a histogram, with a KDE plot layered on top pf histogram.

Distplot function of Seaborn library can be used to build this plot –



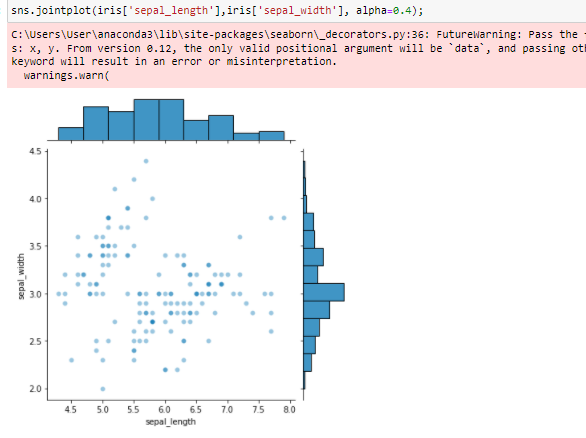


## Jointplot

Jointplot combines Univariate and Bivariate analysis in a unique way.

The relationship between 2 variables is shown in the plot, with histograms of individual variable on the axes.

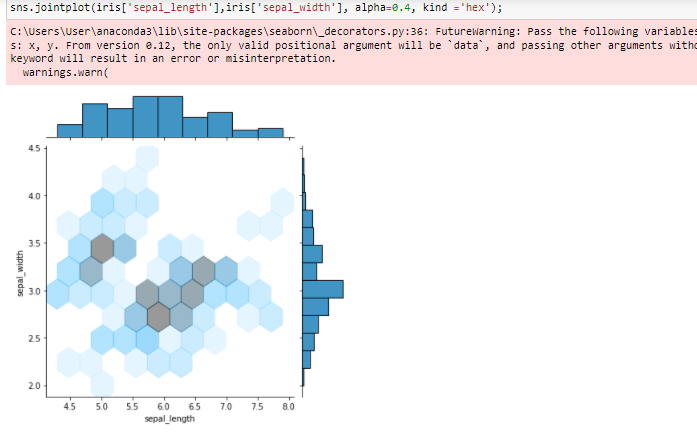
The "Kind" parameter helps in specifying one of the many alterations of this plot, with "Hex" being a commonly used one. Using "Hex" bins the data into hexagonal shapes on the chart as shown in example below.



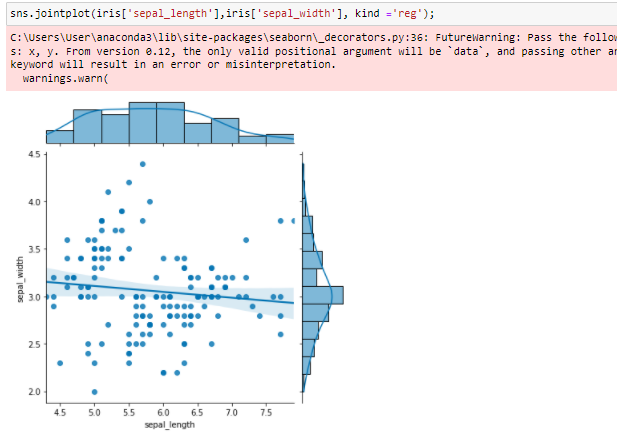
Inferences –

Mean of the Sepal length and Sepal width lie – around 6 and 3 respectively.

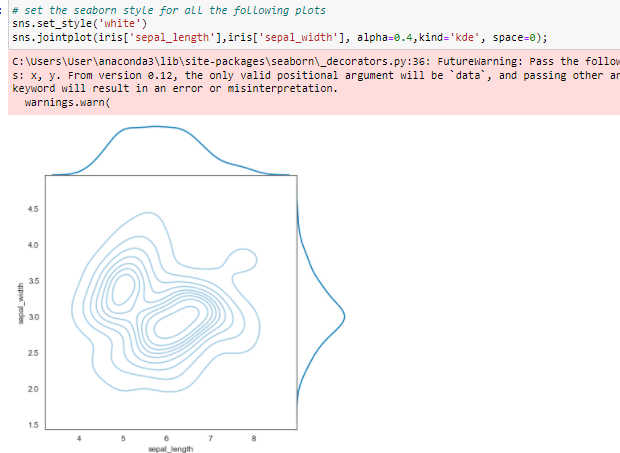
There are 2 clusters of areas where density is more compared to other part.



'Reg' in the 'kind' parameter plots a regression line. We had also talked about how Seaborn has support for Linear Regression, this is one example for the same.



'kde' in the 'kind' parameter plots KDE in 2D and 1D on the plot and axes.



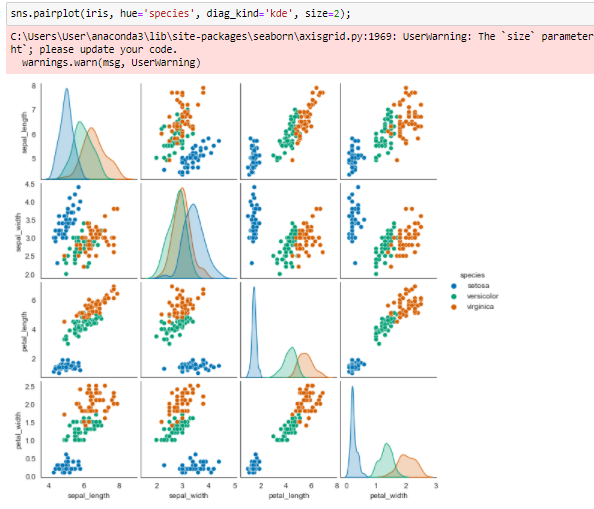
## Pairplot

Pairplots are analogous in Seaborn to Scatter Matrix discussed earlier.

It plots each variable against another in a pairwise manner.

A 'hue' parameter allows for categorising the data-point based on a target variable for a much more insightful chart.

'diag\_kind' dictates what kind of plot should be shown on diagonals where it's a univariate chart.



Inferences –

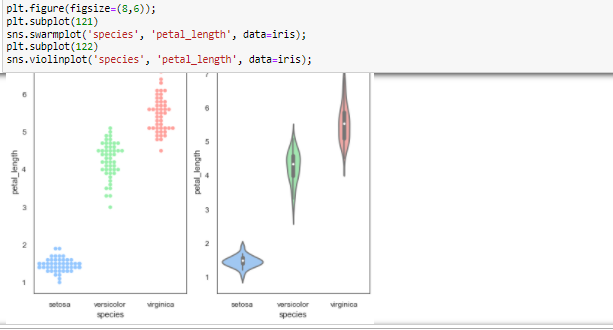
* Following are +vely correlated –
  + Petal Length & Petal Width
  + Petal Width & Sepal Length
  + Petal Length & Sepal Length
* Virginica is a bigger flower with more Length and Width across parameters
* Versicolor is a middle sized flower among the three
* Setosa is the smallest flower among the three

## Swarmplot and Violinplot

Swarmplot and Violinplot are similar to Boxplots, except that they provide addtional information.

A Swarmplot is a box plot, but all the data points are also shown as dots in a symmetrical fashion.

A Violinplot is a box plot, but it alos plots KDE graph of the variable



Inferences -

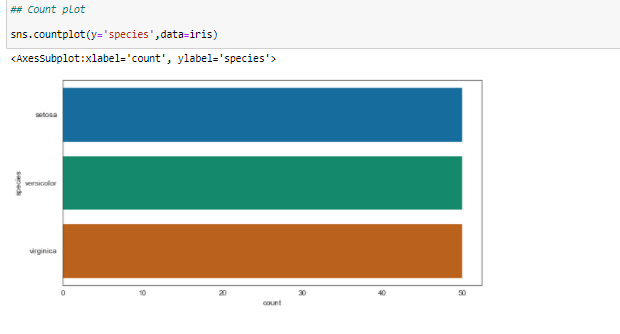
* Setosa has the smallest Petal length of all flowers, while Virginica has longest.
* Setosa also have a sharper curve, meaning most of the flowers had length close to the mean. The same is not true for other 2 flowers having a flatter distribution on Petal length.

## CountPlot

Provides count of distinct values in a variable in the form of bar graph. It can be made horizontal or Vertical by specifying the axis .

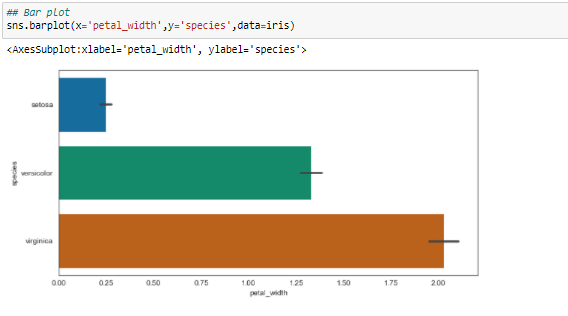


Inference – All flowers have equal data points in the data ( 50 each ).



## Bar Plot

Similar to the bar-chart discussed in last class. Barplot function is used here for plotting the bars.



Inferences –

* Setosa have the smallest Petal width, Virginica the largest
* The error bars on top represent that Setosa also have lower deviation from mean as compared to other 2 flower for Petal Width

## Box Plot

sns.boxplot() is used for plotting this chart.

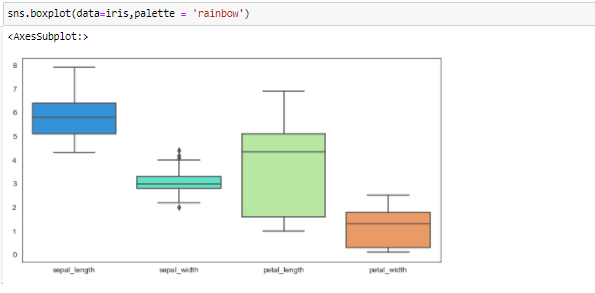
Let’s recall from Statistics session about what the horizontal lines in a box plot represent –

The colored portion represent the “Interquartile range (IQR)” in a box plot.

* **Q1** : Lowest line in IQR is called Q1. It represents the 25th percentile in data.
* **Median** : Mid line in IQR is called Median. It represents the 50th percentile datapoint.
* **Q3** : Highest line in IQR is called Q3. It represents the 75th percentile.

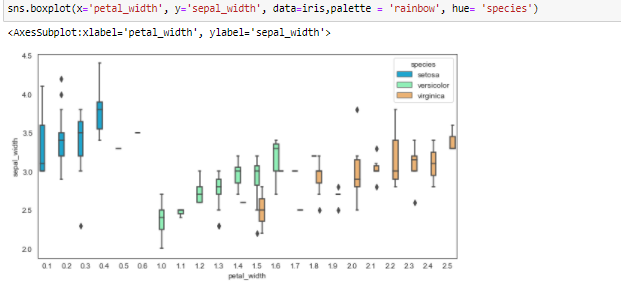
Apart from this, we can notice that there is on topmost line and a bottom line-

* **Minimum** : It is the lowest line in boxplot. The formula for this datapoint is (Q1 – 1.5\*IQR)
* **Maximum** : Topmost horizontal line. Formula – (Q3+ 1.5\*IQR)



Inferences -

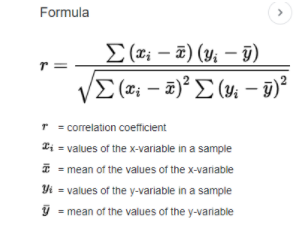
* Sepal Length has the highest mean and takes higher range of value while Petal width the lowest.
* Petal length has the flattest distribution and a high range, while Sepal width is concentrated around its mean, thus more outliers.



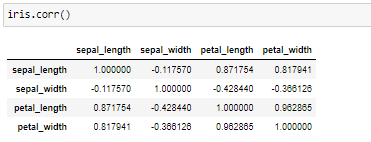
## Correlation plot using Heatmap

As discussed in Statistics session, we know that correlation is a statistical measure of how a variable responds in "coordination" to the another. In other words, it measures the extent of linear association between 2 continuous variables.

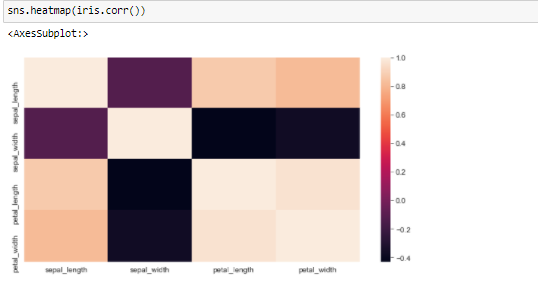
Below is the formula for calculating correlation -



For a Panda dataframe, we can get a correlation matrix of all the quantitative variables in the dataset using corr() function as below –



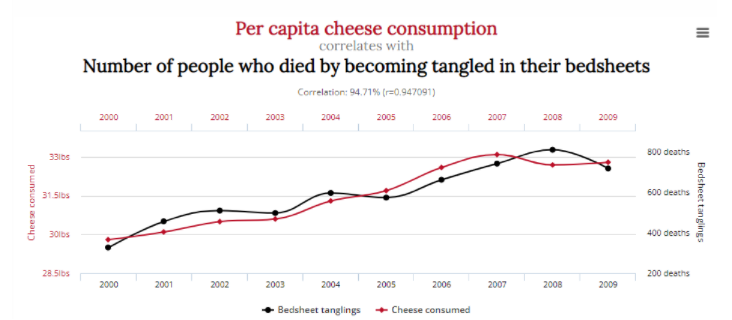
This matrix can be visually plotted using the heatmap() function of Seaborn

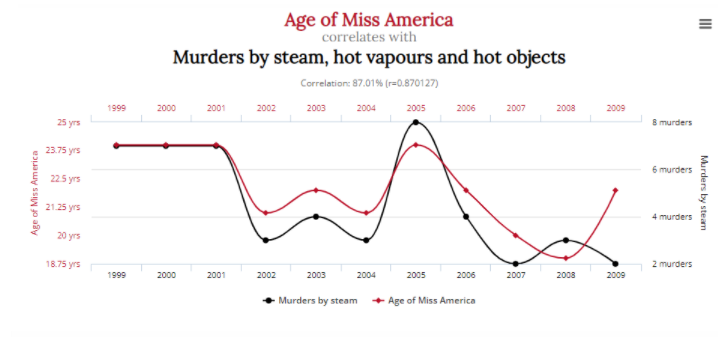


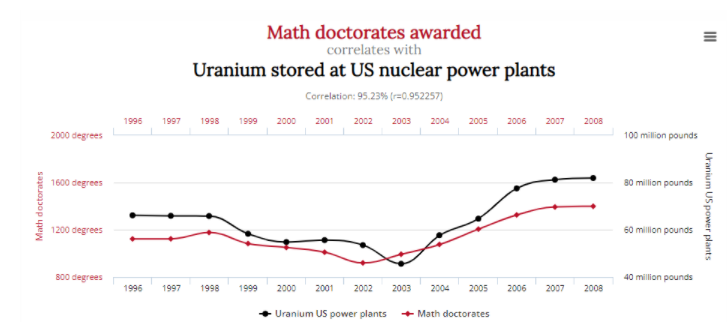
Inference example-  **Sepal Width and Petal Length are inversely correlated** ( One decreases with the increase in another and vice versa )

## Spurious Correlations

Spurious Correlation is a situation when 2 variables appear to be extremely related, but in reality they are not. The said correlation is just a coincidence in such scenarios, and these may lead to faulty insights/models.

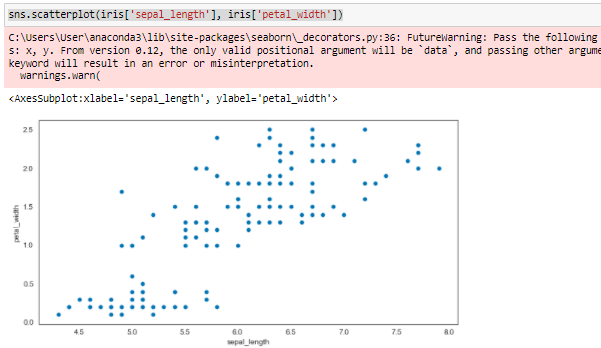


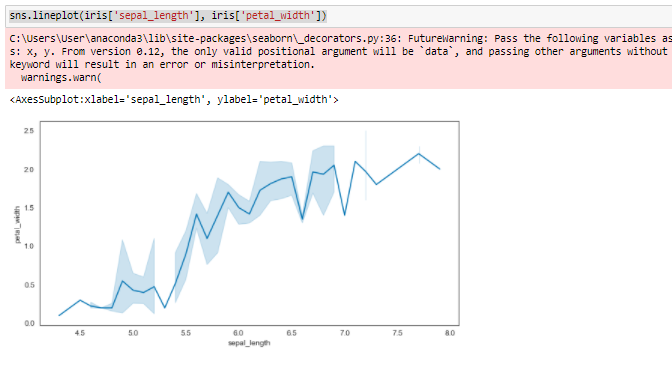




It is evident that although statistical evidence suggests extreme proportionality and linkage in all of these cases, none of the 2 factors can have any effect on each other in reality.

## Scatter Plot, Line Plot



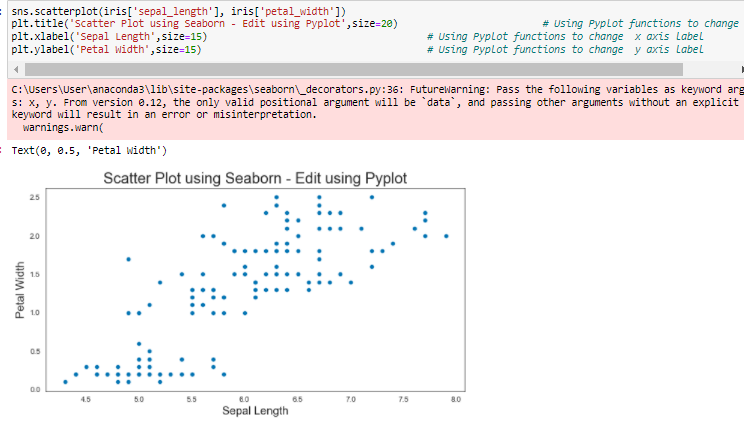


## Editing Charts for Seaborn plots

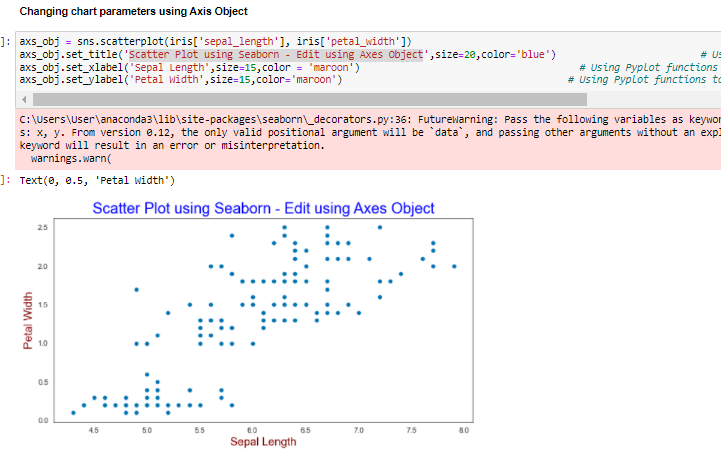
Seaborn is nothing but an another scripting layer on top of Matplotlib and Pyplot. Hence we can use ether use Axis object as well as pyplot functions discussed earlier in the last class for editing the charts.

For this, we'll need to assign plot to a variable so that we can use all the methods associated with the Axes object and Pyplot.

### Changing chart parameters using Pyplot



### Changing chart parameters using Axes Object-



**Seaborn or Matplotlib ?**

Both Seaborn and Matplotlib are widely accepted libraries in data science world. The choice of using the either one wholly depends upon the motivation and application behind plotting the graphs. If the focus is more on plotting standardised plots with great aesthetics ; Seaborn is a good option since it takes cares of overlapping objects.

But if there is a need of highly customised graph, Matplotlib offers great customisations flexibility with the use of objects and methods.

# MCQ (10 minutes)-

Q1. Which plot can be drawn to find out whether 2 variables are roughly correlated or not?

* + Box Plot
  + Bar plot
  + Scatter Plot
  + Histogram

Q2. Which plot combines histogram and kde plot ?

* + Scatter plot
  + Violinplot
  + Swarmplot
  + Distplot

Q3. Which plot combines features of a scatter plot and histogram?

* + Scatter plot
  + Violinplot
  + Swarmplot
  + Jointplot

Q4. Which of the following plot cannot be used for univariate ( 1 variable’s) analysis –

* + Jointplot
  + Histogram
  + KDE plot
  + Box plot

Q5. A heatmap is a visual representation of a correlation matrix.

* + True
  + False

Q6. Which of the pair are analogous to each other ( provide similar info/visual ) -

* Squareplot and Scatter plot
* Heatmap and Histogram
* Scatter Matrix and Pairplot
* KDE and lineplot

Q7. Which of the pair are analogous to each other ( provide similar info/visual ) -

* Swarmplot and Violinplot
* Lineplot and Barchart
* Jointplot and KDE plot
* Histogram and Dendogram

Q8. Which one is correct order of layer/component from lower to higher -

* Pyplot -> Seaborn -> Axes Object
* Axes Object -> Seaborn -> Pyplot
* Axes Object -> Pyplot -> Seaborn
* Seaborn -> Pyplot -> Axes Object

Q9. Which of the following plot would only work with quantitative data -

* Scatter Plot
* Jointplot
* Heatmap
* All of above

Q10. Which plot is useful for visualising multi-dimensional data –

* Box Plot
* Jointplot
* KDE
* Parallel Coordinates

# Homework

Load “mpg” dataset from sns.load\_dataset() and utilise the charts discussed today to find patterns in the data.

Practice following –

* Are there any correlated fields in the data?
* Study distribution of all the fields – Derive insights if any
* Is there any trend with respect to time ?