Artificial Intelligence (Machine Learning & Deep Learning) [Course]

Week 4 - Day 2 - Seaborn

[See examples / code in GitHub code repository]

It is not about Theory, it is 20% Theory and 80% Practical – Technical/Development/Programming [Mostly Python based]

SeaBorn - Overview

seaborn

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

Python Seaborn Module

- **Data visualization** is considered as the best way to depict and analyze the data
- Python Seaborn module basically serves the purpose of Data Visualization at an ease with higher efficiency.
- It supports **NumPy** and **Pandas** data structure to represent the data sets.
- Seaborn stands out to have a better set of functions to carry out data visualization than **Matplotlib** in an optimized and efficient manner.

SeaBorn – Key Features

seaborn

Key Features of Seaborn

Aesthetic Themes

Provides appealing themes and palettes

Simplified Plot Creation

Reduces code complexity for creating plots

DataFrame Handling

Directly works with data frames for analysis

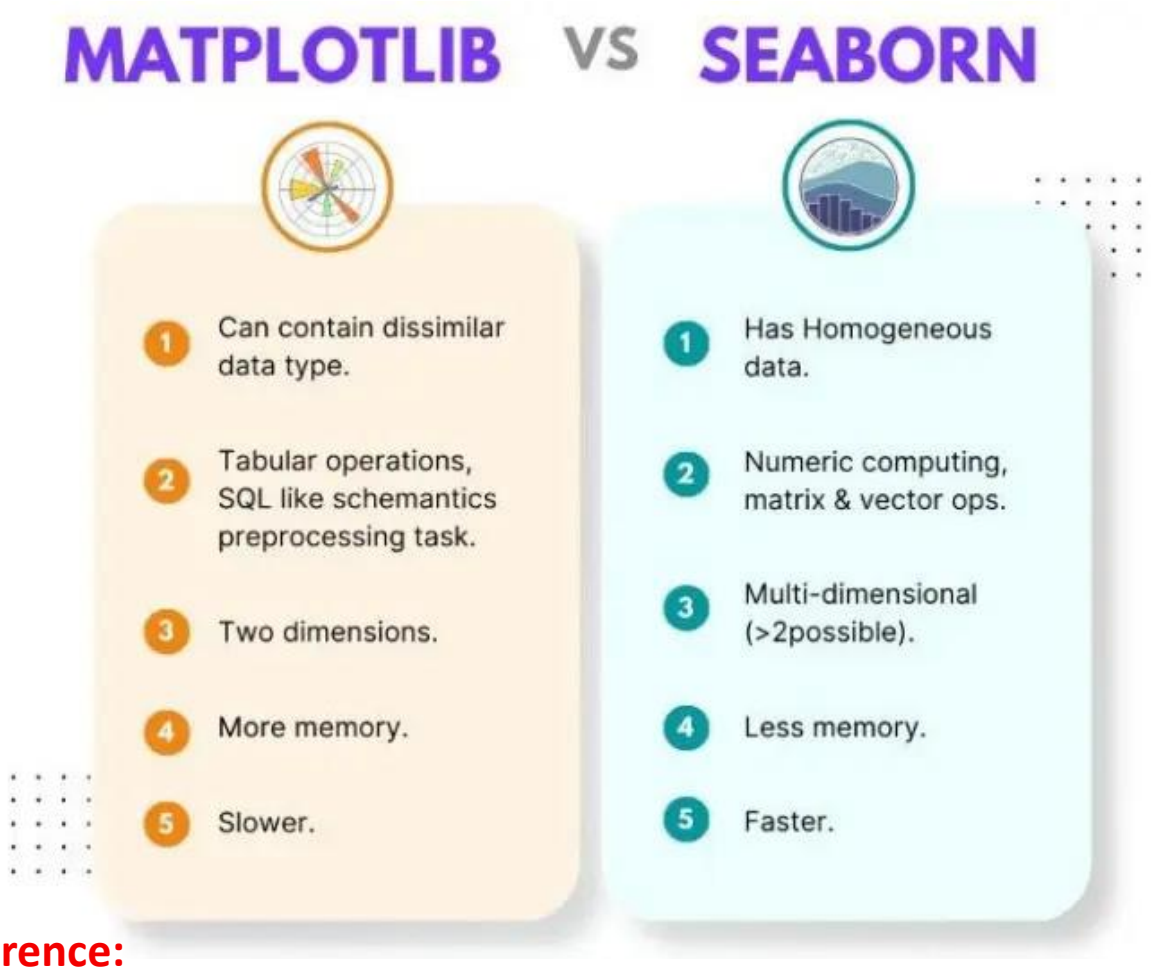
Statistical Visualization Focus

Emphasizes visualizing statistical data

python

MatPlotLib vs SeaBorn

seaborn

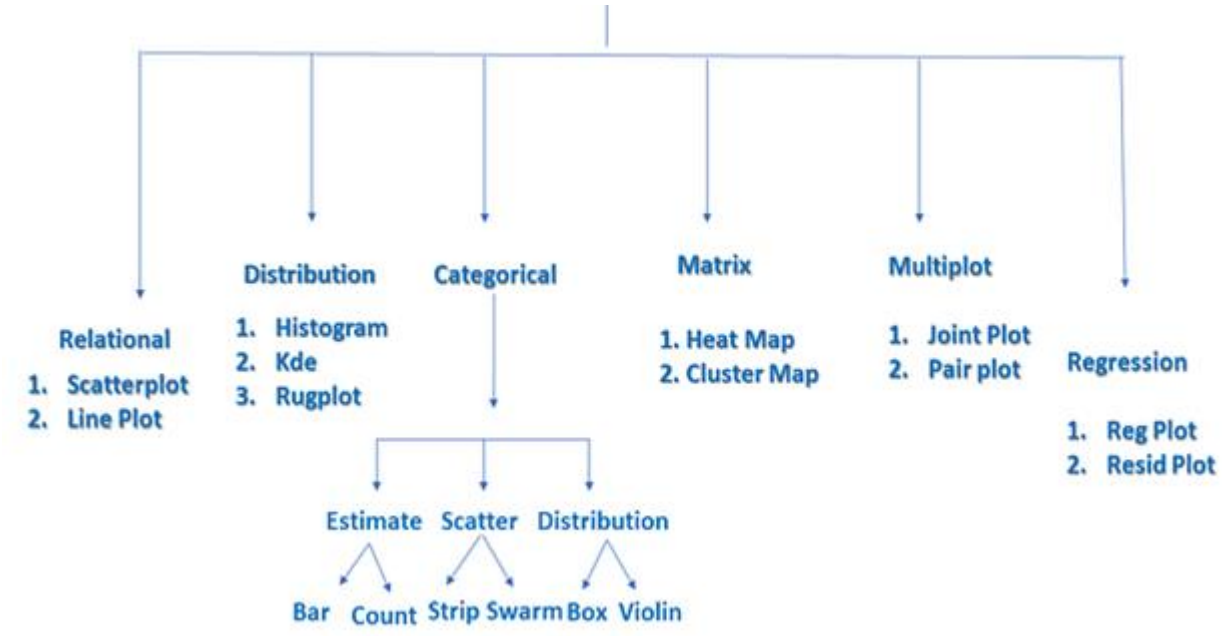


Reference:

https://docs.kanaries.net/topics/Seaborn/seaborn-vs-matplotlib

https://www.newhorizons.com/resources/blog/how-to-choose-between-seaborn-vs-matplotlib

SeaBorn - Plot Types



Reference:

https://medium.com/womenintechnology/complete-seaborn-tutorial-a5e184089c76
https://towardsdatascience.com/14-data-visualization-plots-of-seaborn-14a7bdd16cd7/
https://www.analyticsvidhya.com/blog/2021/12/12-data-plot-types-for-visualization/

Seaborn – cheat sheets

seaborn

3 Plotting With Seaborn

Axis Grids

```
>>> g = sns.FacetGrid(titanic, #Subplot grid for plotting conditional relationships
                      col="survived",
                      row="sex")
>>> g = g.map(plt.hist,"age")
>>> sns.factorplot(x="pclass", #Draw a categorical plot onto a Facetgrid
                   y="survived",
                   hue="sex",
                   data=titanic)
>>> sns.lmplot(x="sepal_width", #Plot data and regression model fits across a FacetGrid
               y="sepal_length",
               hue="species",
               data=iris)
>>> h = sns.PairGrid(iris) #Subplot grid for plotting pairwise relationships
>>> h = h.map(plt.scatter)
>>> sns.pairplot(iris) #Plot pairwise bivariate distributions
>>> i = sns.JointGrid(x="x", #Grid for bivariate plot with marginal univariate plots
                      y="y",
                      data=data)
>>> i = i.plot(sns.regplot,
               sns.distplot)
>>> sns.jointplot("sepal_length", #Plot bivariate distribution
                  "sepal_width",
                  data=iris,
                  kind='kde')
```

Regression Plots

```
>>> sns.regplot(x="sepal_width", #Plot data and a linear regression model fit
                y="sepal_length",
                data=iris,
                ax=ax)
```

Distribution Plots

```
>>> plot = sns.distplot(data.y, #Plot univariate distribution
                        kde=False,
                        color="b")
```

Matrix Plots

```
>>> sns.heatmap(uniform_data,vmin=0,vmax=1) #Heatmap
```

Categorical Plots

Scatterplot

```
>>> sns.stripplot(x="species", #Scatterplot with one categorical variable
                  y="petal_length",
                  data=iris)
>>> sns.swarmplot(x="species", #Categorical scatterplot with non-overlapping points
                  y="petal_length",
                  data=iris)
```

Bar Chart

```
>>> sns.barplot(x="sex", #Show point estimates & confidence intervals with scatterplot glyphs
                y="survived",
                hue="class",
                data=titanic)
```

Count Plot

```
>>> sns.countplot(x="deck", #Show count of observations
                  data=titanic,
                  palette="Greens_d")
```

Point Plot

4 Further Customizations

Also see Matplotlib

Axisgrid Objects

```
>>> g.despine(left=True) #Remove left spine
>>> g.set_ylabels("Survived") #Set the labels of the y-axis
>>> g.set_xticklabels(rotation=45) #Set the tick labels for x
>>> g.set_axis_labels("Survived", #Set the axis labels
                      "Sex")
>>> h.set(xlim=(0,5), #Set the limit and ticks of the x-and y-axis
          ylim=(0,5),
          xticks=[0,2.5,5],
```

Python Seaborn Cheat Sheet

https://www.datacamp.com/cheat-sheet/python-seaborn-cheat-sheet

https://cheatography.com/justin1209/cheat-sheets/seaborn/

https://book-of-gehn.github.io/articles/2021/06/05/Seaborn-Cheatsheet.html

python

SeaBorn - Exercises

seaborn

See code here: https://github.com/ShahzadSarwar10/AI-ML-Explorer
You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. "DEBUGGING IS BEST STRATEGY TO LEARN A LANAGUAGE." So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.
Thanks.
Shahzad – Your AI – ML Instructor

Exercises

Thank you - for listening and participating

- ❑**Questions / Queries**
- ❑**Suggestions/Recommendation**
- ❑**Ideas…..?**

Shahzad Sarwar
Cognitive Convergence
https://cognitiveconvergence.com
shahzad@cognitiveconvergence.com
voice: +1 4242530744 (USA) +92-3004762901 (Pak)