DATA SCIENCE EAST AFRICA

INTRODUCTION TO SEABORN

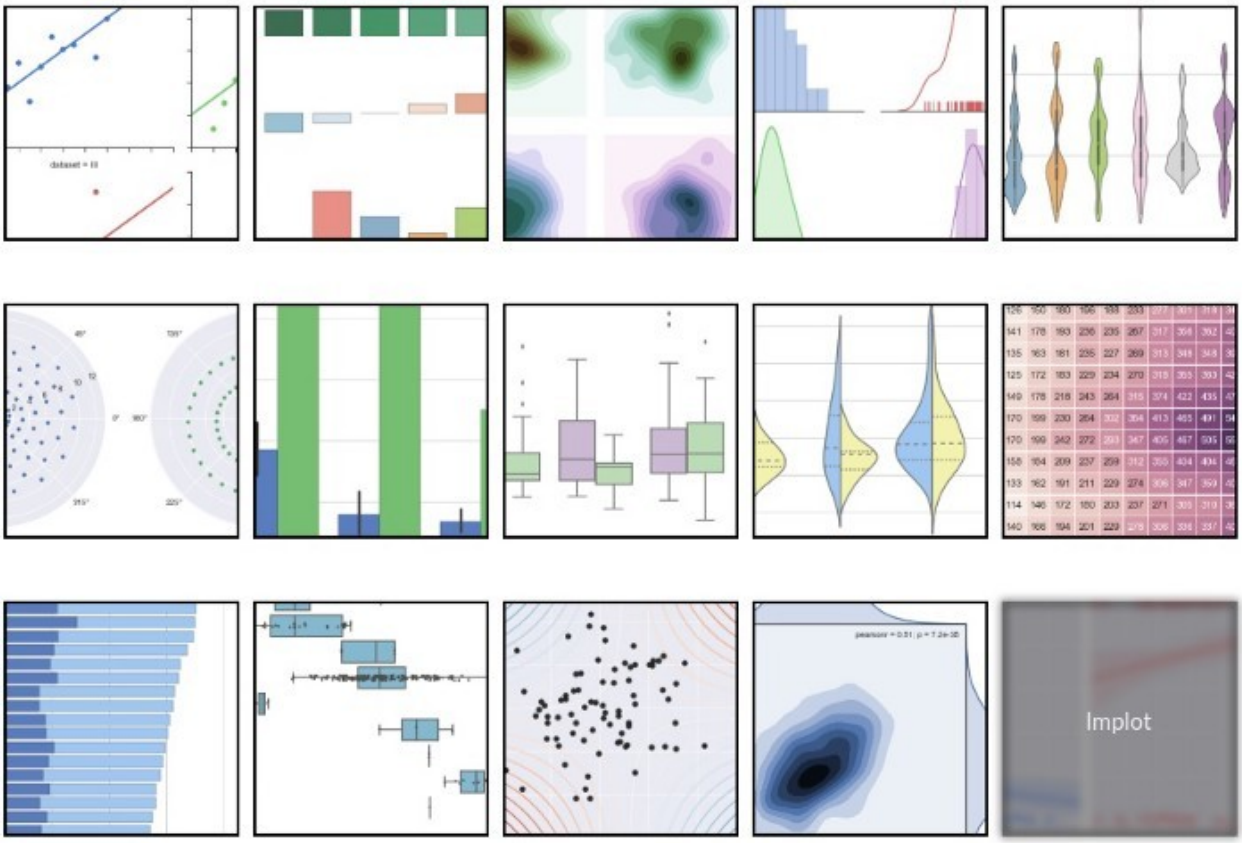
Day 18/20

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset–oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Different categories of plot in Seaborn.

Plots are basically used for visualizing the relationship between variables. Those variables can be either be completely numerical or a category like a group, class or division.



Seaborn divides plot into the below categories :-

- Relational plots: This plot is used to understand the relation between two variables.
- Categorical plots: This plot deals with categorical variables and how they can be visualized.
- Distribution plots: This plot is used for examining univariate and bivariate distributions
- Regression plots: The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
- Matrix plots: A matrix plot is an array of scatter plots.
- Multi-plot grids: It is an useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

Installation

For python environment : pip install seaborn

For conda environment : conda install seaborn

Some Dependencies needed while working with seaborn

Python **Numpy** **Scipy** **Pandas**

Matplotlib **Statsmodel**

Steps to plot different graphs using seaborn library

In the following examples, we are using a dataset of COVID-19 to make plots for their different columns present in a dataset(file name is COVID-19.csv).

Step 1: Importing pandas so to read the file.

Read a csv file by using read_csv () present in pandas.

```
In [72]: import pandas as pd

In [73]: dataset=pd.read_csv('COVID-19.csv')
```

Step 2: Printing the columns.

Get the columns so to determine which columns we want to use for plotting graphs.

```
In [75]: dataset.columns

Out[75]: Index(['Sno', 'age', 'gender', 'body temperature', 'Dry Cough', 'sour throat',
                'weakness', 'breathing problem', 'drowsiness', 'pain in chest',
                'travel history to infected countries', 'diabetes', 'heart disease',
                'lung disease', 'stroke or reduced immunity', 'symptoms progressed',
                'high blood pressue', 'kidney disease', 'change in appetide',
                'Loss of sense of smell', 'Corona result'],
               dtype='object')
```

Step 3: Importing seaborn so to make graphs.

```
In [84]: import seaborn as sns
```

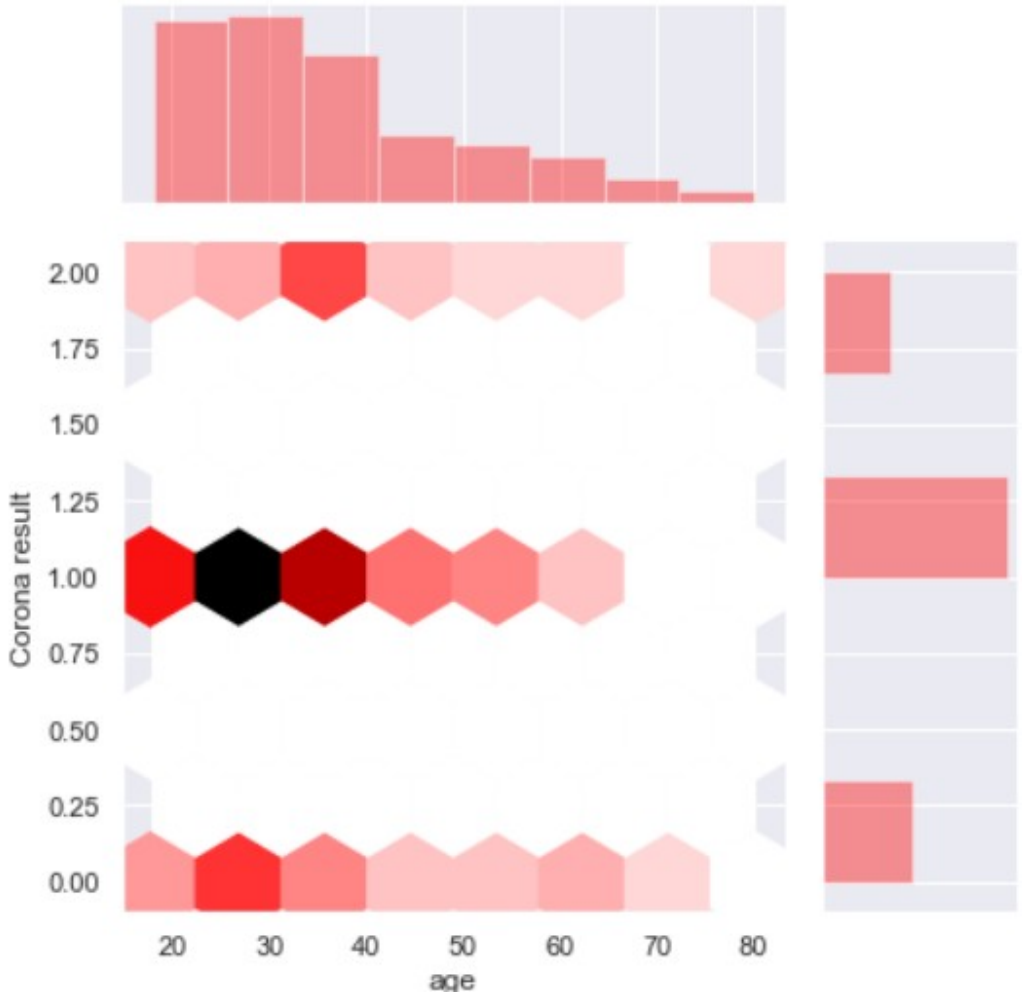
Step 4: Plotting graph to see distribution between 2 variables

Here we will use **jointplot** to draw a plot of two variables with bivariate and univariate graphs.

To see the parameters present in jointplot – Simply use Shift+Tab inside jointplot() so to use them according to our own choice. By clicking on + button, we can see all the parameters.

```
In [88]: sns.jointplot(data= dataset, x='age', y='Corona result',
                       kind='hex',color='red', height=6,ratio=3)
```

```
Out[88]: <seaborn.axisgrid.JointGrid at 0x277fb848308>
```

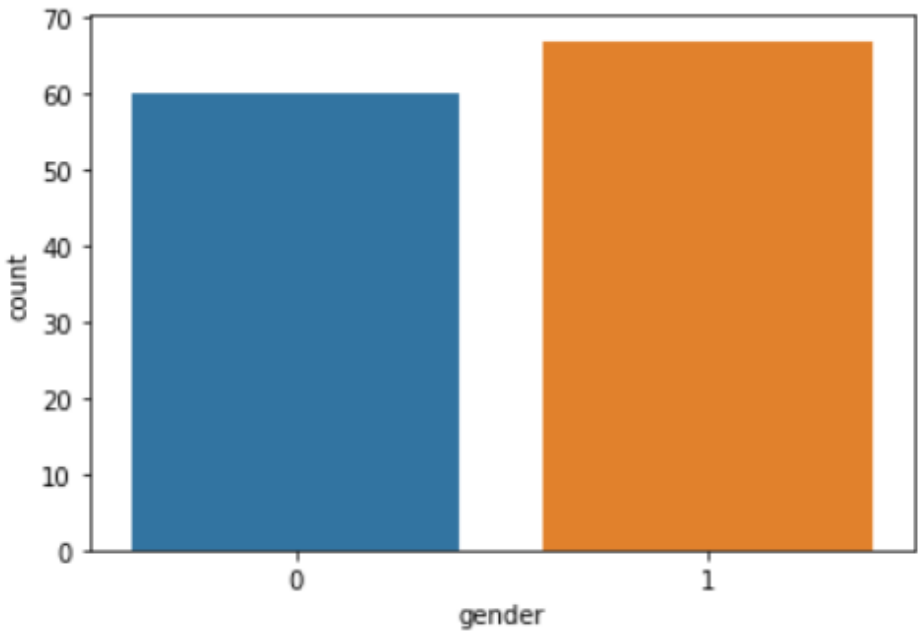


Explanation :

Using **age** and **Corona result**(0: **low**, 1: **medium**, 2:**high**) column to plot a graph and using kind of plot='hex' and also passing various parameters like color, height, ratio of the graph.

Plotting a graph for categorical variables like "gender"

```
In [33]: sns.countplot(x ='gender', data = dataset)
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1e8ba793648>
```



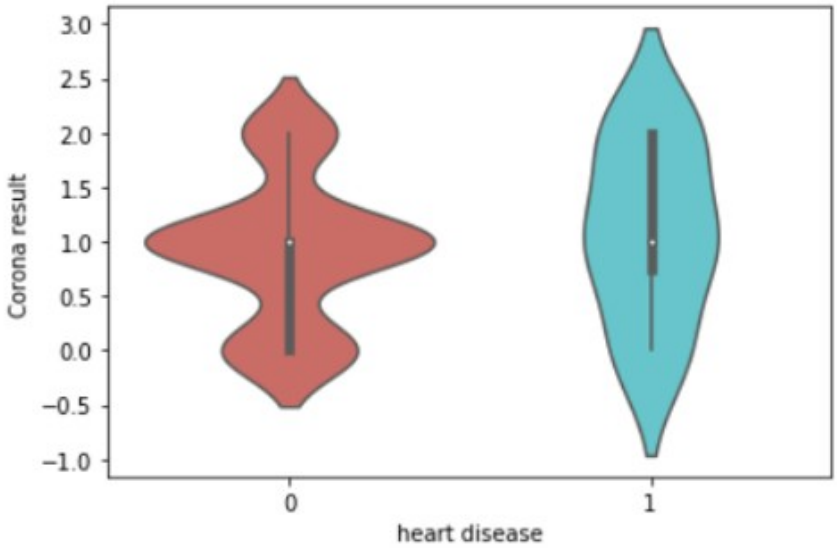
Explanation:

Using **countplot**, which basically counts the categories and returns a count of their occurrences. Here, 0: **male** and 1: **female** so we can see females(approx. 68) have more occurrences than males(approx. 59).

Plotting a graph to see relation between 2 variables.

```
In [66]: sns.violinplot(data= dataset, x='heart disease', y='Corona result'
                        dodge=True, bw='scott', palette="hls")

Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1e8bd834ac8>
```



Using **violinplot** to see the distribution of the quantitative data that represents the comparisons between variables and has advanced visualization to see a better description about the data distribution.

bw *{'scott', 'silverman', float}*–Either the name of a reference rule or the scale factor to use when computing the kernel bandwidth.

Dodge– Elements will be shifted along the categorical axis.

Here, the people who have heart disease, have more chances to get infected

Advantage of using seaborn library to make graphs.

- It uses fewer syntax and has easily interesting default themes.

- It provides a variety of visualization patterns.

- It specializes in statistics visualization and is used if one has to summarize data in visualizations and also show the distribution in the data.

- It is more integrated than matplotlib for working with Pandas data frames.

- It extends the Matplotlib library for creating beautiful graphics with Python using a more straightforward set of methods.

It is summarized that if Matplotlib “tries to make easy things easy and hard things possible”, Seaborn tries to make a well-defined set of hard things easy too.”

Seaborn helps resolve the two major problems faced by Matplotlib.

As Seaborn compliments and extends Matplotlib, the learning curve is quite gradual. If you know Matplotlib, you are already half way through Seaborn.