

Seaborn

- 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.
- Seaborn was built on matplotlib and closely integrated with pandas data structure.
- It was released in 2012.
- It includes a large number of plots dedicated towards providing better insights, along with myriad colors and palettes.
- There are various type of plots. It can be 2D or 3D.
- It is installed using : *pip3 install seaborn*

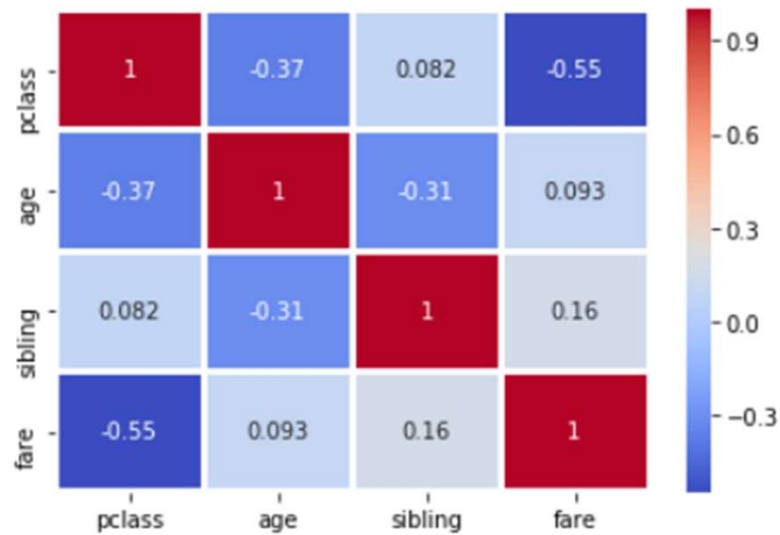
Heatmap

- **Syntax :** `seaborn.heatmap(data, annot)`
- This plot helps you to analyze continuous columns of the entire dataset.
- It will provide relationship between 2 columns.
- `data` : dataset with correlation parameter, or pivot table has to be passed
- `annot` (default = False) : If true, it displays the correlation values between columns.

Heatmap

- **Syntax :** `seaborn.heatmap(data, annot)`
- Dataset Taken : Titanic.csv

```
sns.heatmap(df.corr(), annot = True, cmap = 'coolwarm', linewidths= 2)  
plt.show()
```



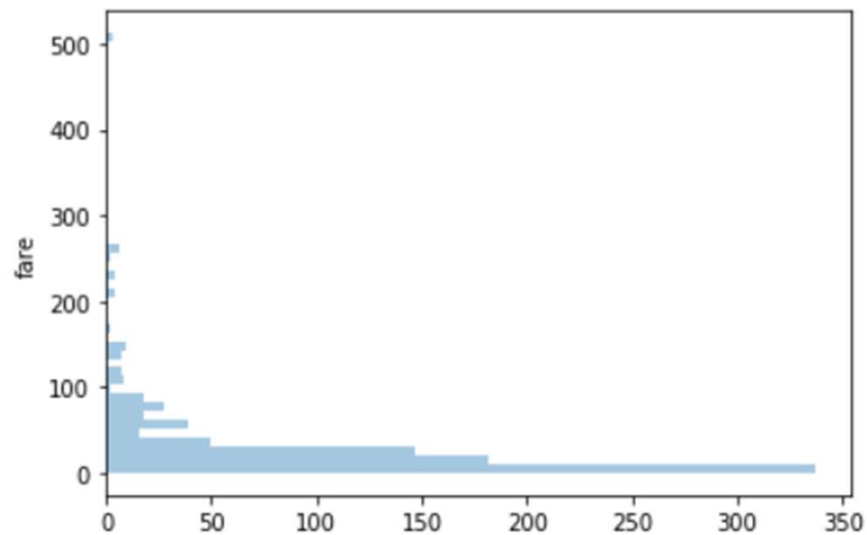
Distribution Plot

- **Syntax :** `seaborn.distplot(data, bins, kde)`
- This plot helps you to analyze single column of continuous data only.
- It will provide you the frequency of the data values.
- bins : Helps to get an idea of the trend of the column. Lesser the bins, less is the information, and vice-verse.
- kde (Kernel Density Estimation) : to estimate the probability density function of random variable

Distribution Plot

- **Syntax :** `seaborn.distplot(data, bins, kde)`
- Dataset Taken : Titanic.csv

```
sns.distplot(df['fare'], bins = 50, kde = False, vertical= True)  
plt.show()
```



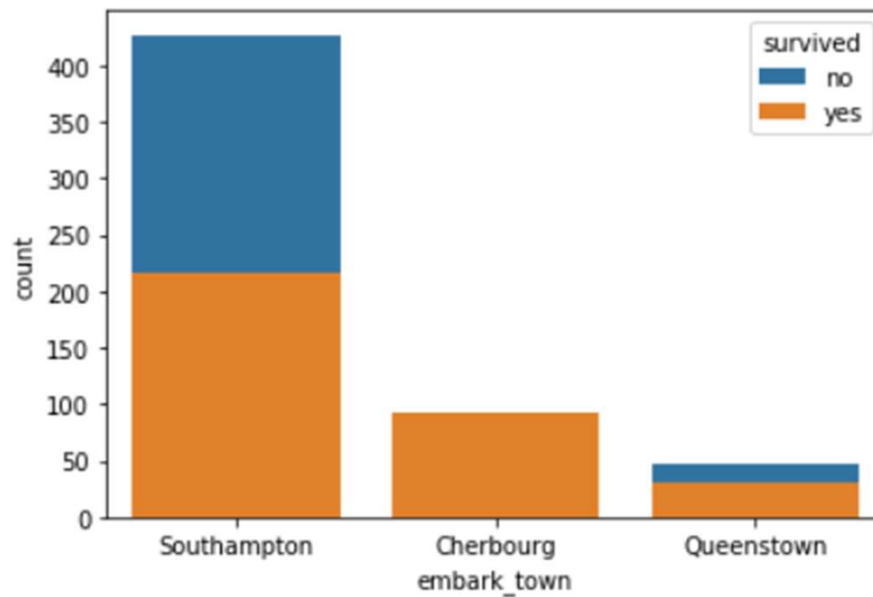
Countplot

- **Syntax :** `seaborn.countplot(data, dodge, hue)`
- This plot helps you to analyze single column of categorical data.
- It will provide you the frequency of the data values.
- `dodge` (default = True) : the splits occurring with respect to hue, are stacked over each other.
- `hue` : It means to compare the count of single column with respect to other categorical column.

Countplot

- **Syntax :** `seaborn.countplot(data, dodge, hue)`
- Dataset Taken : Titanic.csv

```
sns.countplot(df['embark_town'], hue = df['survived'], dodge=False)  
plt.show()
```



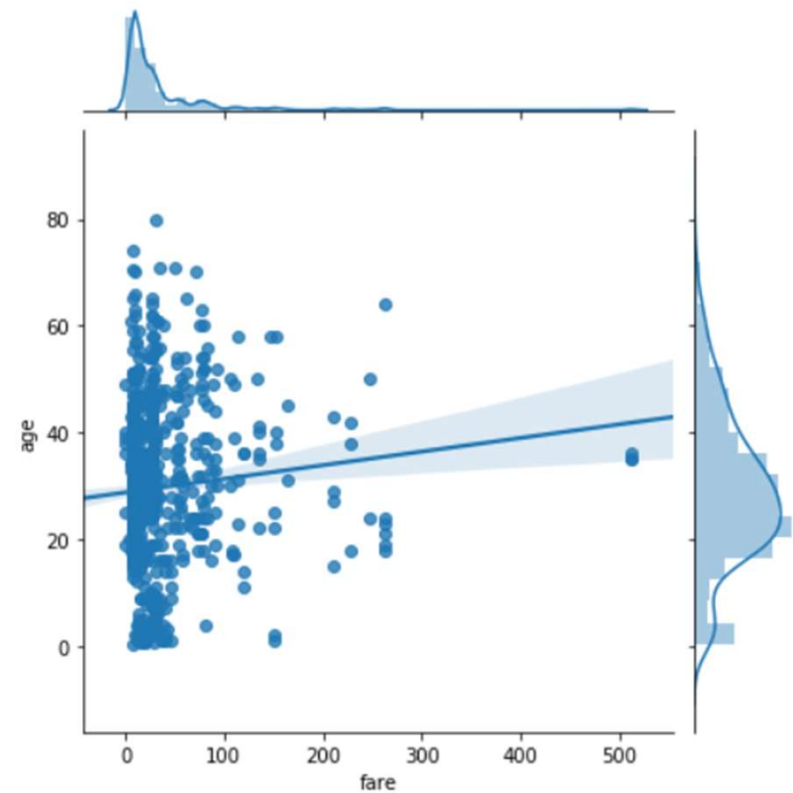
Jointplot

- **Syntax :** `seaborn.jointplot(x, y)`
- This plot helps you to analyze two columns of continuous data.
- It will provide you the insights or patterns between them.
- jointplot shows the comparison between two columns along with their distribution plot.
- x : x-axis columns
- y : y-axis columns

Jointplot

- **Syntax :** `seaborn.jointplot(x, y)`
- Dataset Taken : Titanic.csv

```
sns.jointplot(x = df['fare'], y = df['age'], kind= "reg")  
plt.show()
```



Barplot

- **Syntax :** `seaborn.barplot(x, y)`
- This plot helps you to analyze two columns, categorical vs continuous data.
- We can also plot a single column.
- It will provide you the insights or patterns between them.
- x : x-axis columns
- y : y-axis columns

Barplot

- **Syntax :** `seaborn.barplot(x, y)`
- Dataset Taken : Titanic.csv

```
sns.barplot(y = df['age'])  
plt.show()
```



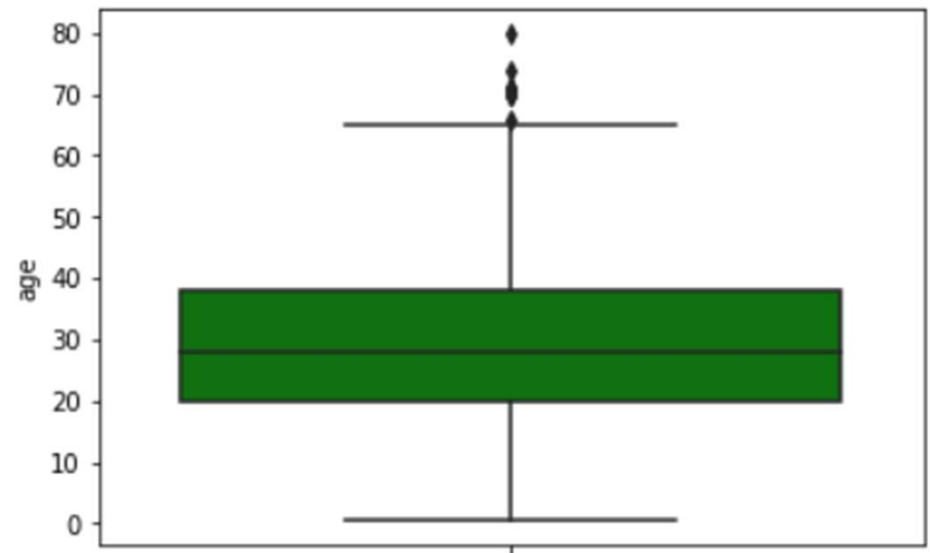
Boxplot

- **Syntax :** `seaborn.barplot(x, y)`
- This plot helps you to analyze two columns, categorical vs continuous data.
- We can also plot a single column, that helps us to identify outliers
- It will provide you the insights or patterns between them.
- x : x-axis columns
- y : y-axis columns

Boxplot

- **Syntax :** `seaborn.barplot(x, y)`
- Dataset Taken : Titanic.csv

```
sns.boxplot(y = df['age'], color = 'g')  
plt.show()
```



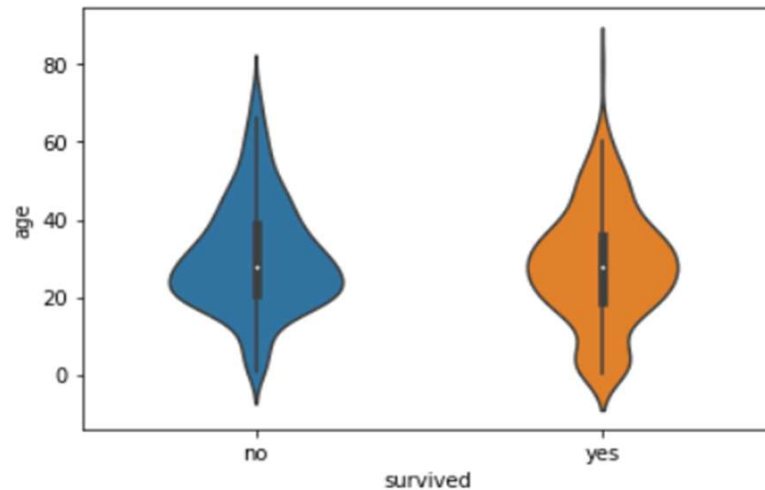
Violinplot

- **Syntax :** `seaborn.violinplot(x, y)`
- This plot helps you to analyze two columns, categorical vs continuous data.
- We can also plot a single column.
- It draws a combination of boxplot and kde.
- It will provide you the insights or patterns between them.
- x : x-axis columns
- y : y-axis columns

Violinplot

- **Syntax :** `seaborn.violinplot(x, y)`
- Dataset Taken : Titanic.csv
 - width (default = 0.8) :
 - changes the width of each violin.

```
sns.violinplot(x = df['survived'], y = df['age'], width = 0.5)  
plt.show()
```



Stripplot

- **Syntax :** `seaborn.stripplot(x, y)`
- This plot helps you to analyze two columns, categorical vs continuous data.
- We can also plot a single column.
- It draws a scatterplot.
- It will provide you the insights or patterns between them.
- x : x-axis columns
- y : y-axis columns

Stripplot

- **Syntax :** `seaborn.stripplot(x, y)`
- Dataset Taken : Titanic.csv

```
sns.stripplot(x = df['survived'], y = df['age'], jitter = False)  
plt.show()
```

- jitter (default = True)
- the columns are displayed with scattered datapoints.
- If False, the datapoints are stacked over one another.

