**Exploratory Data Analysis (EDA) techniques for Kaggle competition beginners**

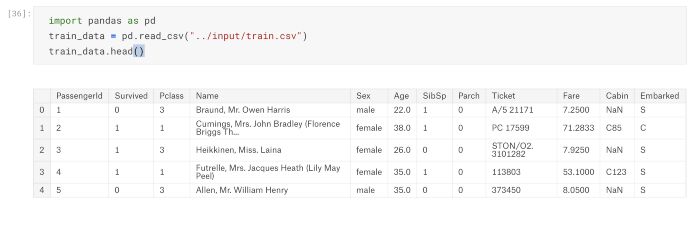
A hands on guide for beginners on EDA and Data Science competitions

E**xploratory Data Analysis** (**EDA**) is an approach to analysing data sets to summarize their main characteristics, often with visual methods. Following are the different steps involved in EDA :

1. Data Collection
2. Data Cleaning
3. Data Preprocessing
4. Data Visualisation

**Data Collection**

**Data collection** is the process of gathering information in an established systematic way that enables one to test hypothesis and evaluate outcomes easily.



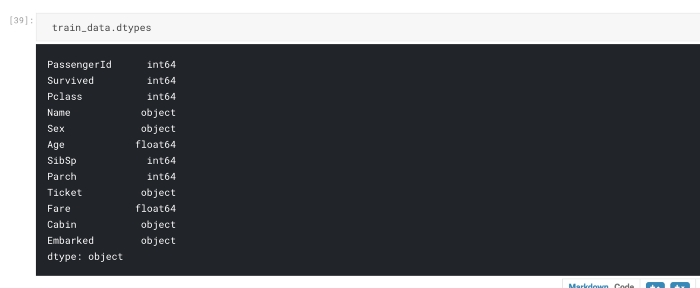
After getting data we need to check the data-type of features.

There are following types of features :

* numeric
* categorical
* ordinal
* datetime
* coordinates

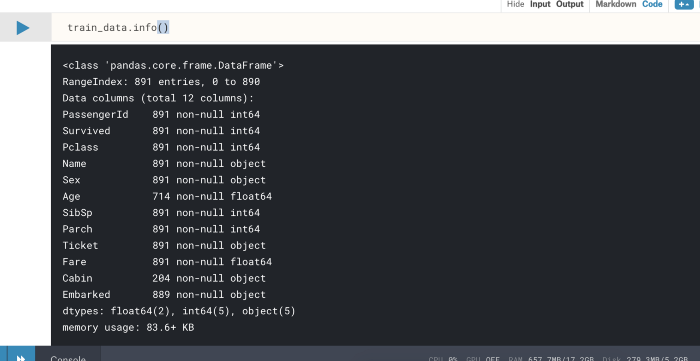
In order to know the data types/features of data, we need to run following command:

train\_data.dtypes



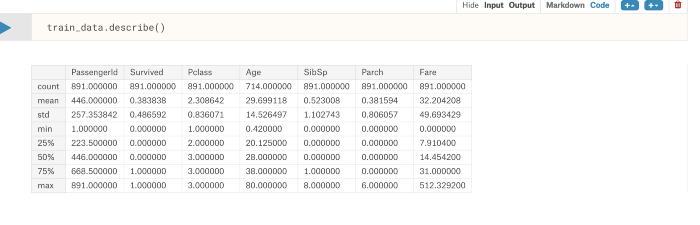
or

train\_data.info()



Let’s have a look to the statistical summary about our dataset.

train\_data.describe()



**Data Cleaning**

**Data cleaning** is the process of ensuring that your data is correct and useable by identifying any errors in the data, or missing data by correcting or deleting them. Refer to [this link for data cleaning.](https://confusedcoders.com/data-science/cleaning-data-for-data-visualisation" \t "_blank)

Once the data is clean we can go further for data preprocessing.

**Data Preprocessing**

**Data preprocessing** is a data mining technique that involves transforming raw data into an understandable format. It includes normalisation and standardisation, transformation, feature extraction and selection, etc. The product of data preprocessing is the final training dataset.

**Data Visualisation**

**Data visualisation** is the graphical representation of information and data. It uses statistical graphics, plots, information graphics and other tools to communicate information clearly and efficiently.

Here we will focus on commonly used Seaborn visualisation. Seaborn is a Python data visualisation library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Following are the common used seaborn visualisation :-

1. Scatter Plot
2. Box Plot
3. Histogram
4. Cat Plot
5. Violin Plot
6. Pair Plot
7. Joint plot
8. Heat Map

# import seaborn library

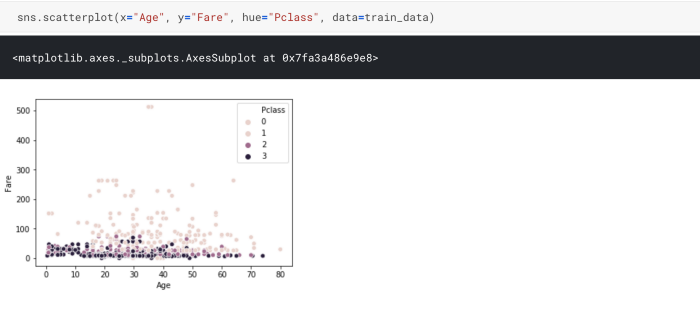
import seaborn as sns

**Scatter Plot**

A scatter plot is a set of points plotted on a horizontal and vertical axes.

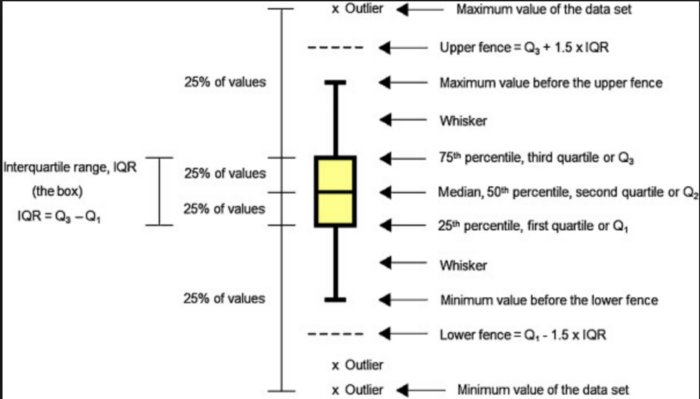
Scatter plot below shows the relationship between the passenger age and passenger fare based on pclass (Ticket class) from data taken from Titanic dataset

sns.scatterplot(x="Age", y="Fare", hue = 'Pclass', data=train\_data)



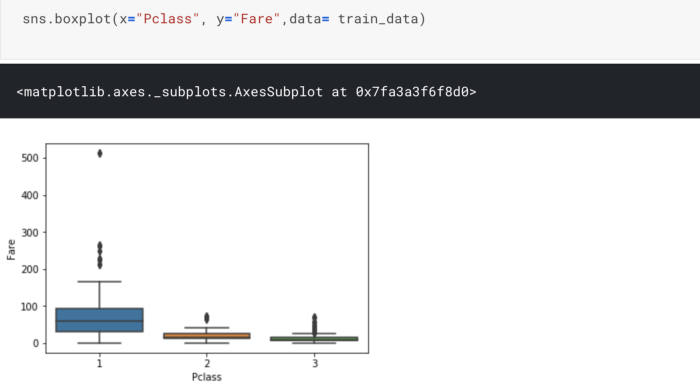
**Box Plot**

Box plot is a simple way of representing statistical data on a plot in which a rectangle is drawn to represent the second and third quartiles, usually with a vertical line inside to indicate the median value. The lower and upper quartiles are shown as horizontal lines either side of the rectangle.



Box plot below shows how the passenger fare varies based on ticket class.

sns.boxplot(x="Pclass", y="Fare",data= train\_data)



**Histogram**

A histogram is an accurate representation of the distribution of numerical data. It is an estimate of the probability distribution of a continuous variable

sns.distplot( train\_data['Pclass'], kde=False)



**Cat Plot**

Cat plot provides access to several axes-level functions that show the relationship between a numerical and one or more categorical variables using one of several visual representations. We can used different kind of plot to draw (corresponds to the name of a categorical plotting function)Options are: “point”, “bar”, “strip”, “swarm”, “box”, or “violin”. More details about Cat plot is [here](https://seaborn.pydata.org/generated/seaborn.catplot.html" \t "_blank)

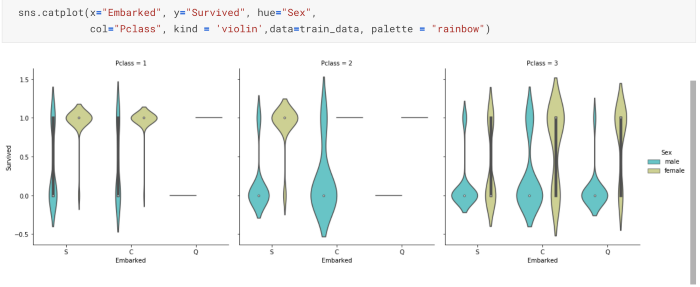
Below we do a cat plot with bar kind

sns.catplot(x="Embarked", y="Survived", hue="Sex",col="Pclass", kind = 'bar',data=train\_data, palette = "rainbow")



Let’s have a look on same cat plot with violin kind

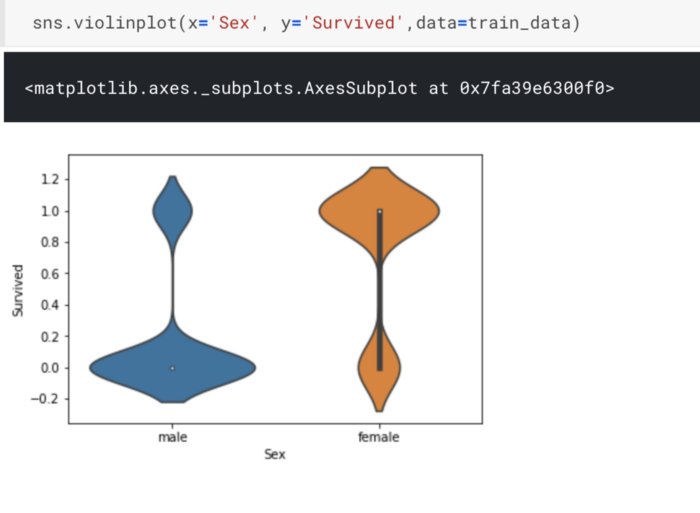
sns.catplot(x="Embarked", y="Survived", hue="Sex",col="Pclass", kind = 'violin',data=train\_data, palette = "rainbow")



**Violin Plot**

A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. More details about Violin plot is [here](https://seaborn.pydata.org/generated/seaborn.violinplot.html" \t "_blank)

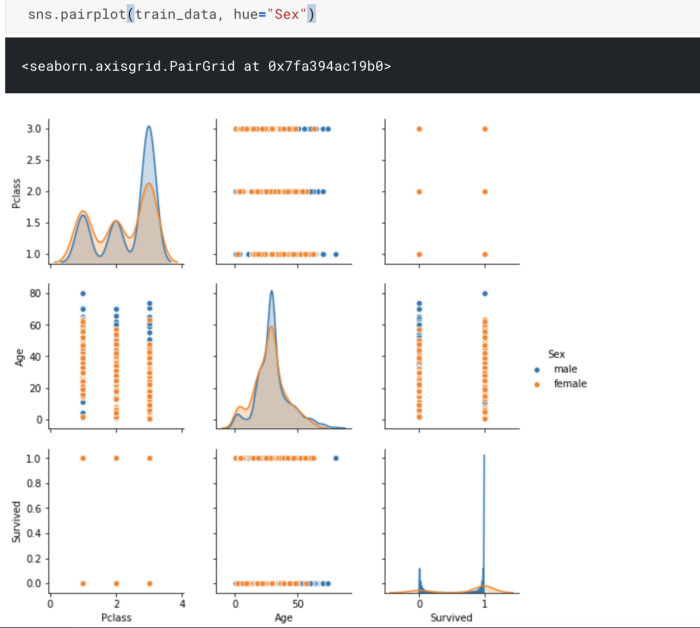
sns.violinplot(x='Sex', y='Survived',data=train\_data)



**Pair Plot**

Pair plot in seaborn only plots numerical columns although later we will use the categorical variables for coloring. More about pair plot is [here](https://seaborn.pydata.org/generated/seaborn.pairplot.html" \t "_blank).

sns.pairplot(train\_data, hue="Sex")

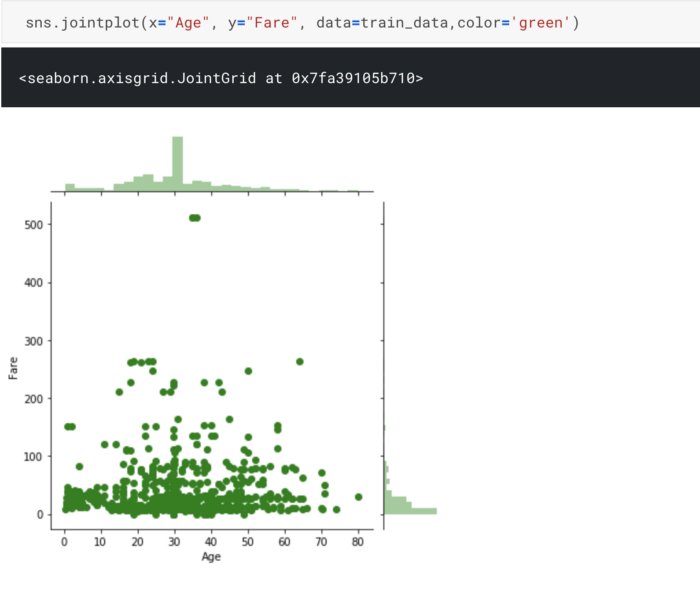


**Joint Plot**

Jointplot is seaborn library specific and can be used to quickly visualize and analyze the relationship between two variables and describe their individual distributions on the same plot.

More about Joint plot is [here.](https://seaborn.pydata.org/generated/seaborn.jointplot.html" \t "_blank)

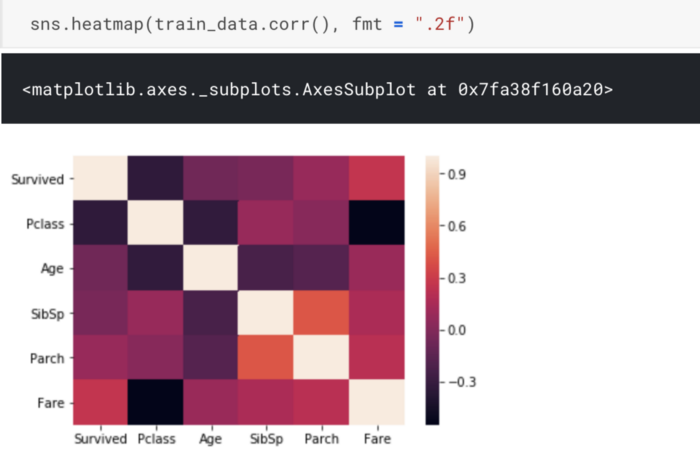
sns.jointplot(x="Age", y="Fare", data=train\_data, color ='green')



**Heat Map**

Heat map is a representation of data in the form of a map or diagram in which data values are represented as colours.

sns.heatmap(train\_data.corr(), fmt = ".2f")



That’s all for this post, hope it was helpful. Cheers!