Exploratory Data Analysis (EDA) techniques for kaggle competition beginners

medium.com/@nikkisharma536/exploratory-data-analysis-eda-techniques-for-kaggle-competition-beginners-be4237c3c3a9
Nikita sharma November 25, 2018

Nikita sharma
Nov 25

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.

[36]:
```
import pandas as pd
train_data = pd.read_csv("../input/train.csv")
train_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

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`

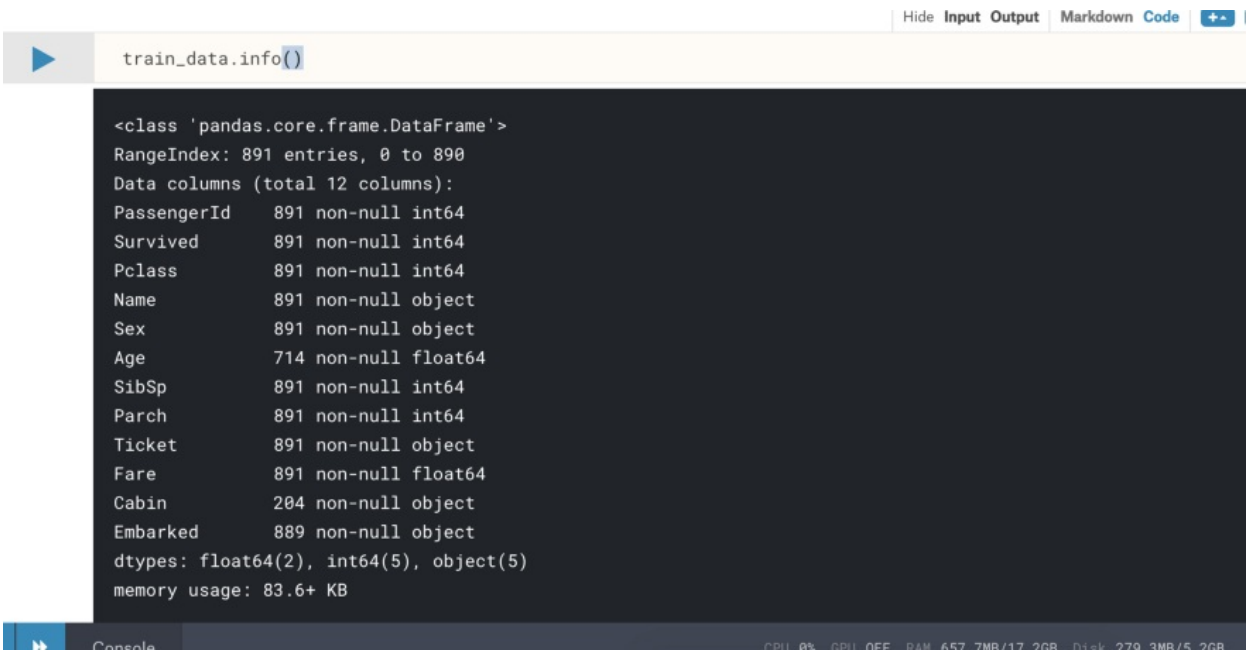
[39]:
```
train_data.dtypes
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

Markdown Code

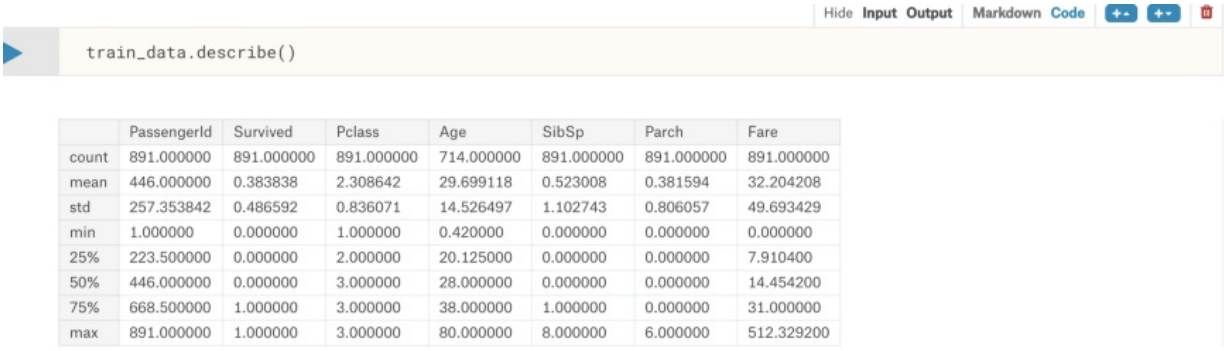
or

`train_data.info()`



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

```
train_data.describe()
```



	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

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.

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

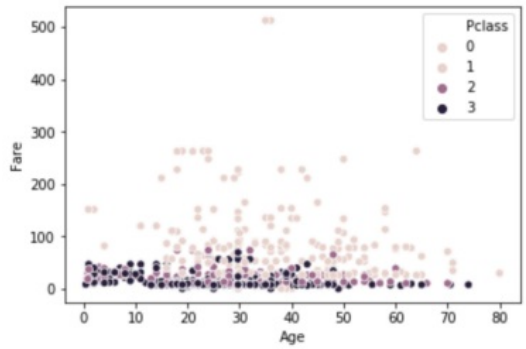
A scatter plot is a set of points plotted on a horizontal and vertical axes.

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)
```

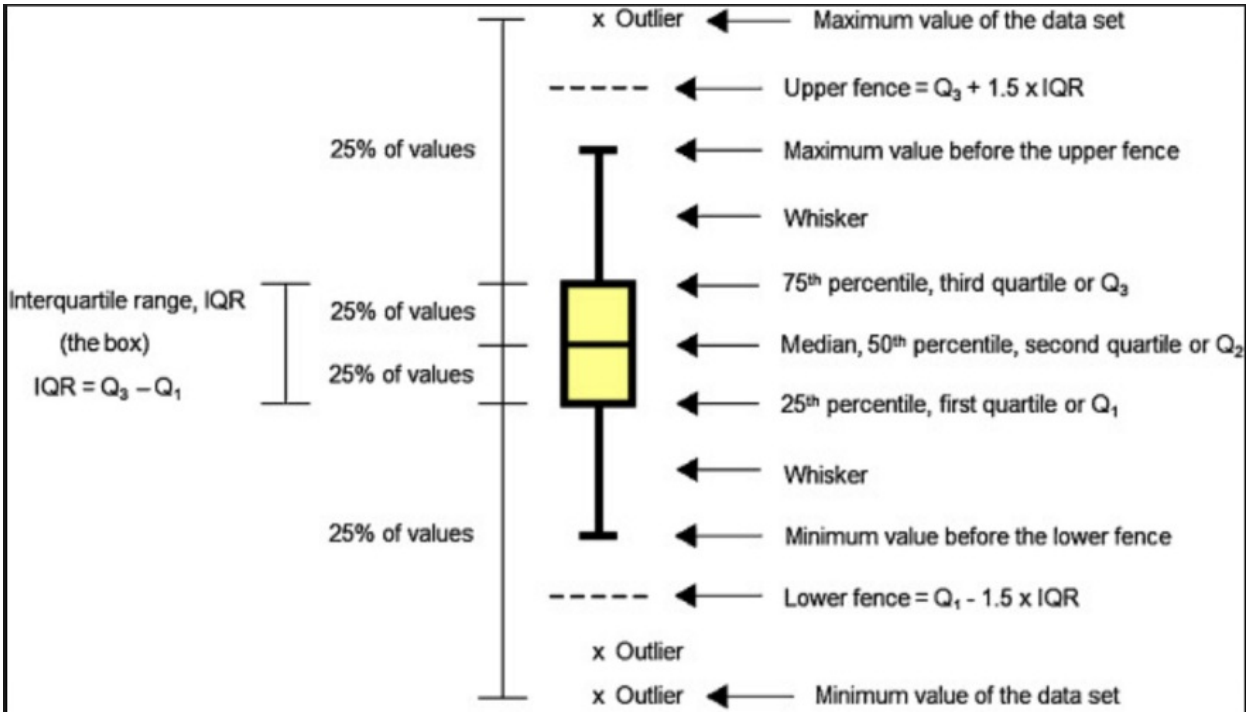
```
sns.scatterplot(x="Age", y="Fare", hue="Pclass", data=train_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa3a486e9e8>
```



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.

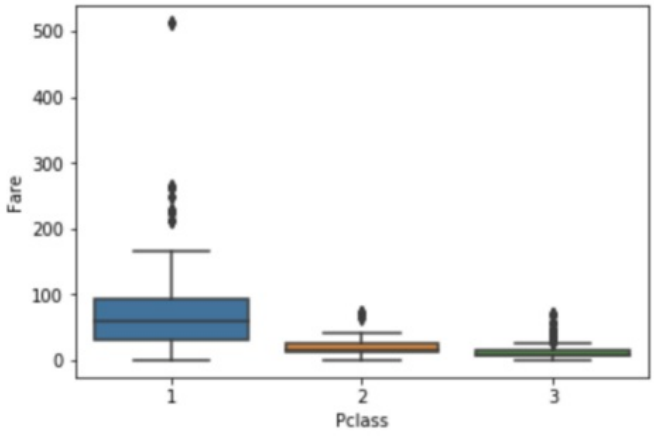


Box plot below shows how the passenger fare varies based on ticket class.

```
sns.boxplot(x="Pclass", y="Fare",data= train_data)
```

```
sns.boxplot(x="Pclass", y="Fare",data= train_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa3a3f6f8d0>
```



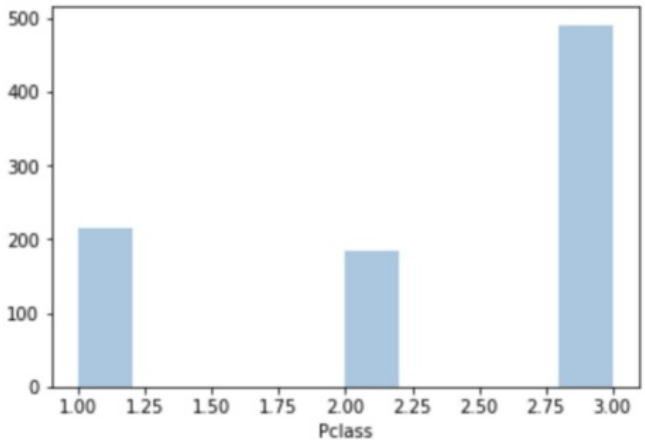
Histogram

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)
```

```
sns.distplot( train_data['Pclass'],  kde=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa3a3e13c18>
```



Cat Plot

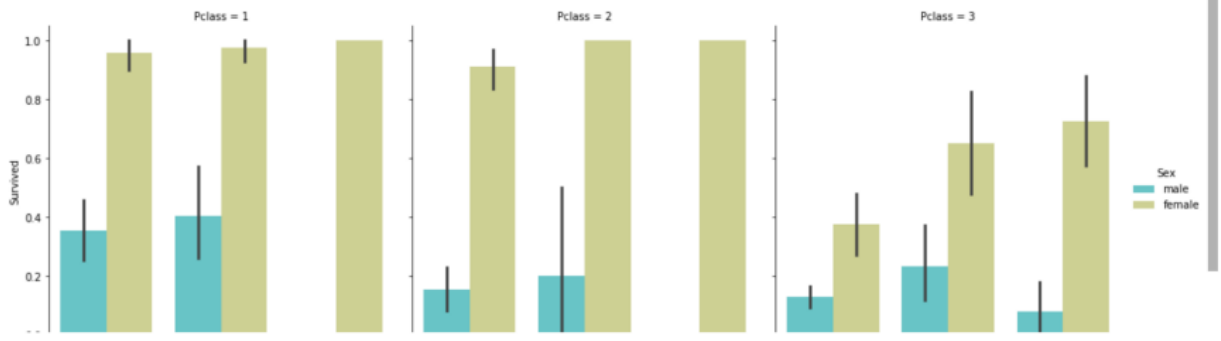
Cat plot provides access to several axes-level functions that show the relationship between a numerical and one or more categorical variables using one of several visual representations. We can used different kind of plot to draw (corresponds to the name of a categorical plotting function)Options are: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

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")
```

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

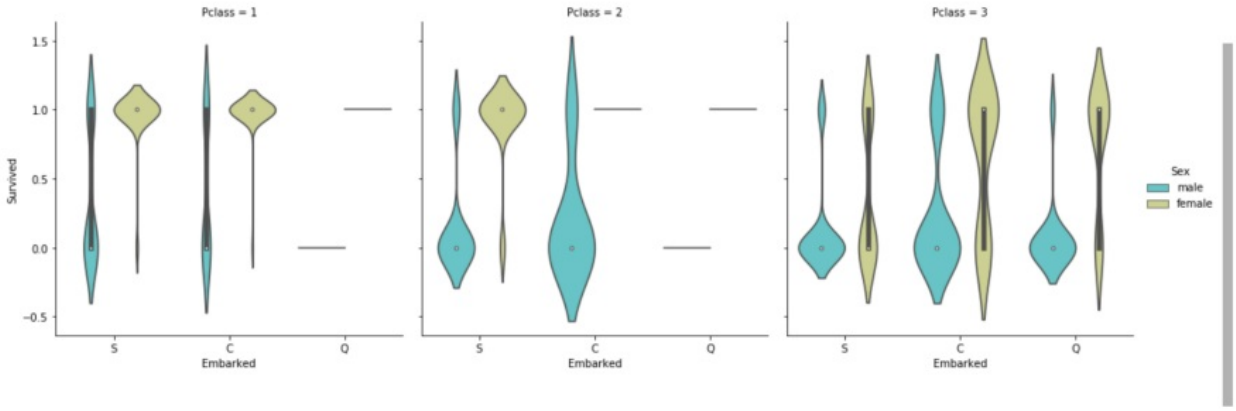
```
<seaborn.axisgrid.FacetGrid at 0x7fa3a5154e48>
```



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")
```

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



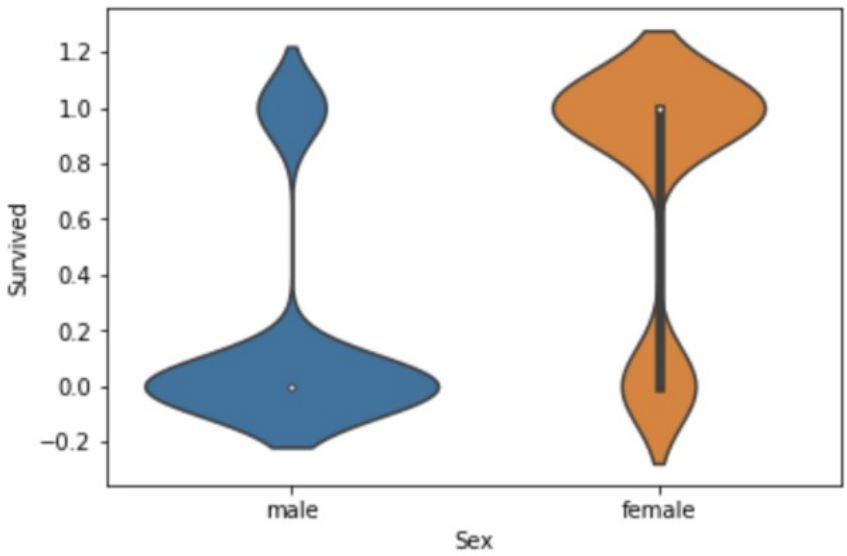
Violin Plot

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

```
sns.violinplot(x='Sex', y='Survived',data=train_data)
```

```
sns.violinplot(x='Sex', y='Survived',data=train_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa39e6300f0>
```



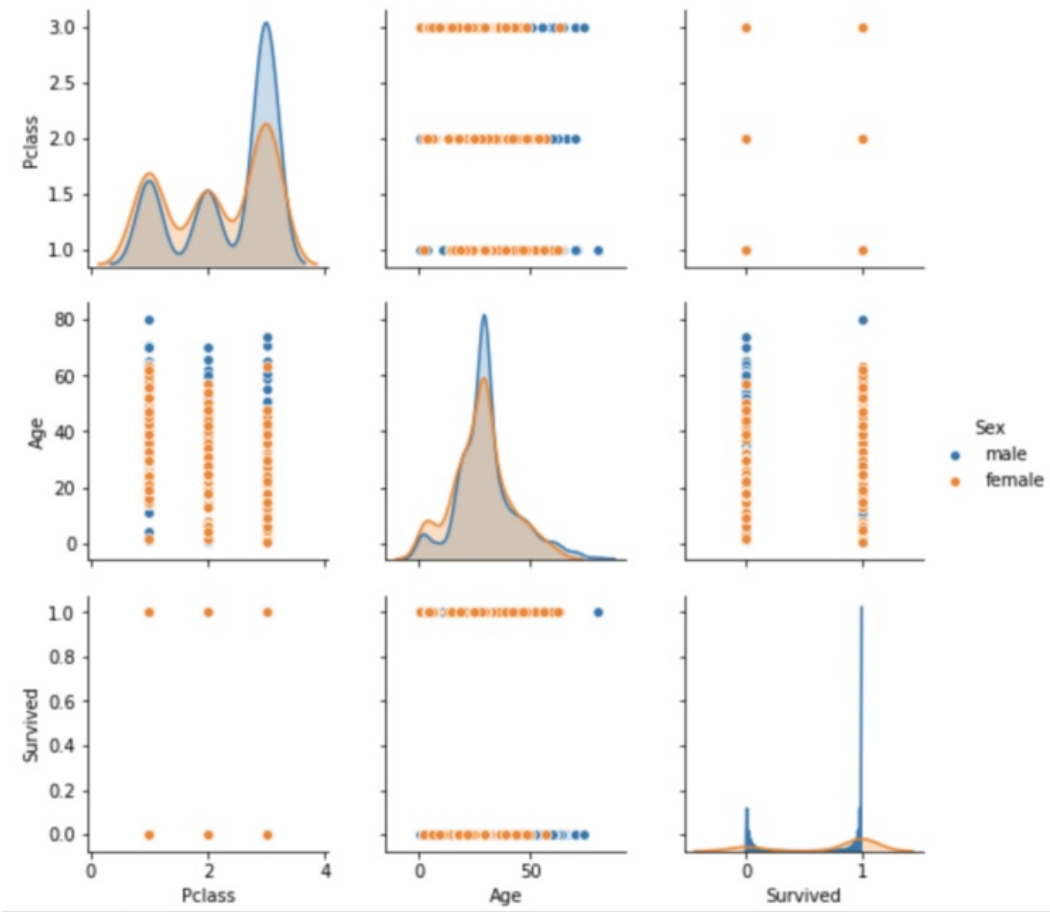
Pair Plot

Pair plot in seaborn only plots numerical columns although later we will use the categorical variables for coloring. More about pair plot is here.

```
sns.pairplot(train_data, hue="Sex")
```

```
sns.pairplot(train_data, hue="Sex")
```

```
<seaborn.axisgrid.PairGrid at 0x7fa394ac19b0>
```



Joint Plot

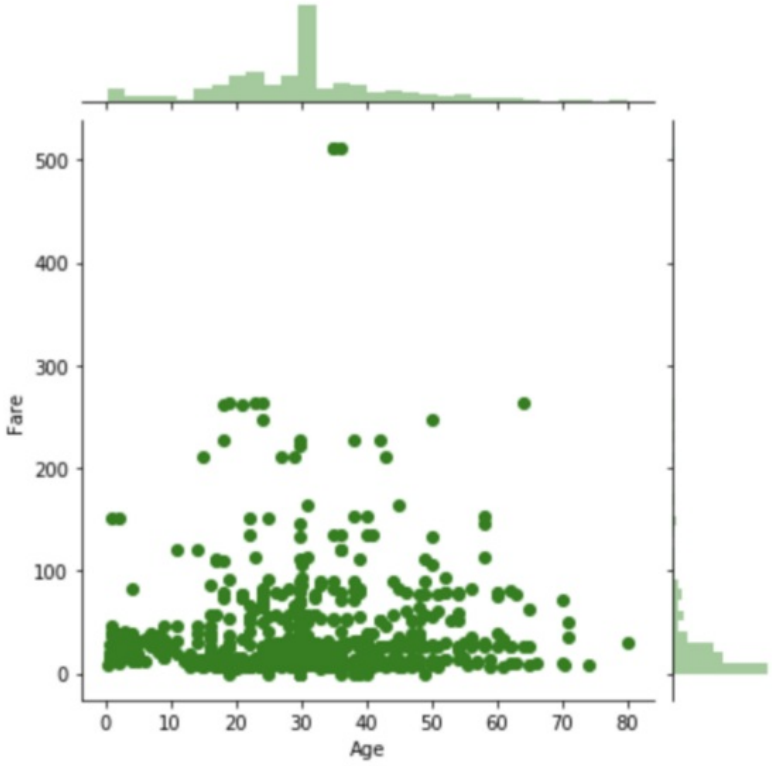
Jointplot is seaborn library specific and can be used to quickly visualize and analyze the relationship between two variables and describe their individual distributions on the same plot.

More about Joint plot is here.

```
sns.jointplot(x="Age", y="Fare", data=train_data, color ='green')
```

```
sns.jointplot(x="Age", y="Fare", data=train_data,color='green')
```

```
<seaborn.axisgrid.JointGrid at 0x7fa39105b710>
```



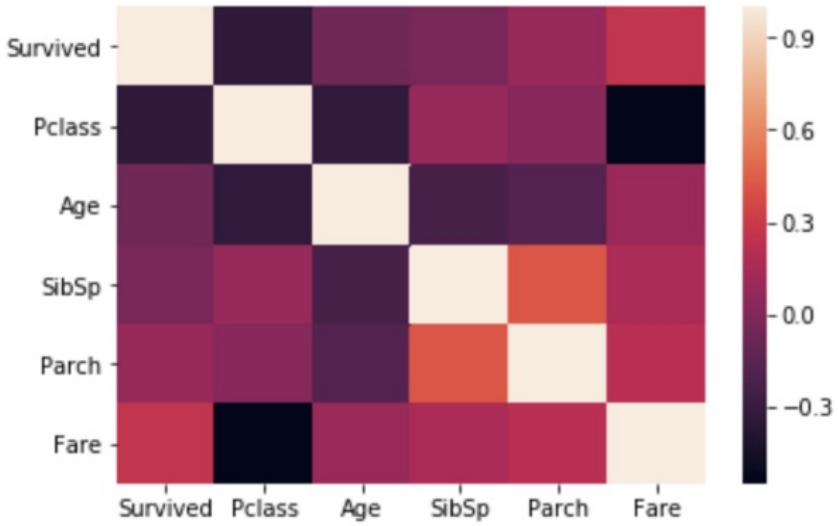
Heat Map

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")
```

```
sns.heatmap(train_data.corr(), fmt = ".2f")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa38f160a20>
```



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

Originally published at confusedcoders.com on November 25, 2018.