greatlearning
Learning for Life

Data Visualization

Agenda

- Visualization using Seaborn
- Visualization using Plotly

Data visualization

- Representation of the data in a pictorial or graphical format
- First step of data analysis
- Allow us to get the intuitive understanding of the data
- Helps to visualize the patterns in the data

greatlearning
Learning for Life

Visualization using Seaborn

Introduction

- Seaborn is used for data visualization
- It is based on the matplotlib
- It provides a high-level interface for drawing attractive and informative statistical graphics

Functionalities of seaborn

- Allows comparison between multiple variables
- Supports multi-plot grids
- Univariate and bivariate visualization
- Availability of different color palettes
- Estimates and plots linear regression line

Installation

Open terminal program (for Mac user) or command line (for Windows) and install it using following command:

```
conda install seaborn
```

Or

```
pip install seaborn
```

Installation

- Alternatively, you can install seaborn in a jupyter notebook using below code:

```
!pip install seaborn
```

- To import the library, use the command:

```
Import seaborn as sns
```

Strip plot

- It is similar to the scatter plot with one categorical variable
- It is used to understand the underlying distribution of the data
- One axis represents the categorical variable and another represents the value corresponding to the categories

Read the data

Load the titanic data to create a strip plot

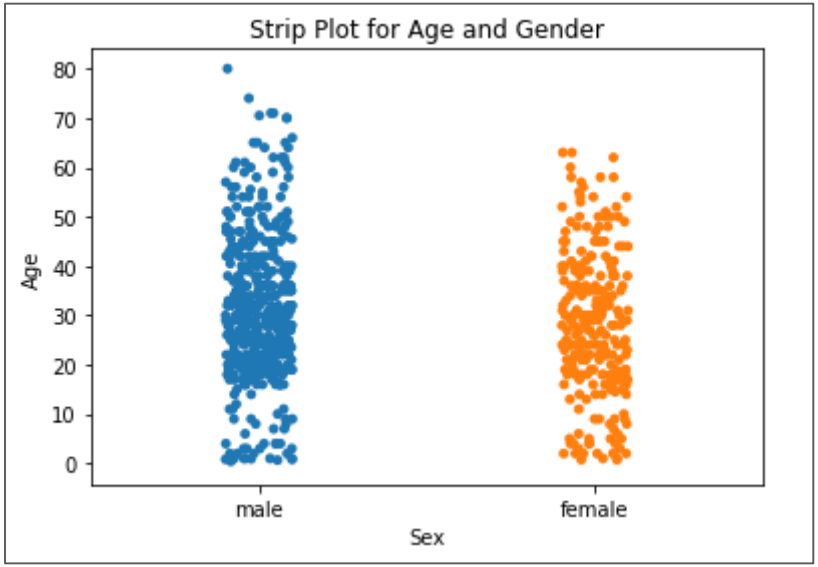
```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

# display first five rows
df_titanic.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

Strip plot

Check the distribution of age based on gender

```
# plot a strip plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.stripplot(x = 'Sex', y = 'Age', data = df_titanic)

# add the plot label
plt.title('Strip Plot for Age and Gender')

# display the plot
plt.show()
```



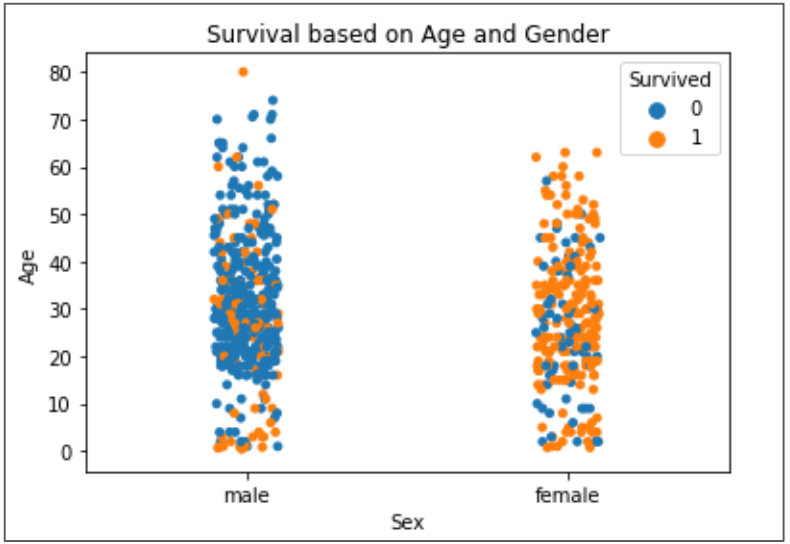
The plot shows that, the maximum age of males is higher than of females

Strip plot

Add the one more categorical variable to strip plot using the parameter, ‘hue’

```
# plot a strip plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'hue' adds one more variable to the plot
# 'data' represents the DataFrame
sns.stripplot(x = 'Sex', y = 'Age', hue = 'Survived' , data = df_titanic)

# add the plot label
plt.title('Survival based on Age and Gender')

# display the plot
plt.show()
```

Add a categorical variable



Proportion of female survivors is higher than males

Violin plot

- It is similar to a boxplot, that displays the kernel density estimator of the underlying distribution
- It shows the distribution of the quantitative data across categorical variables such that those distributions can be compared

Read the data

Load the titanic data to create a violin plot

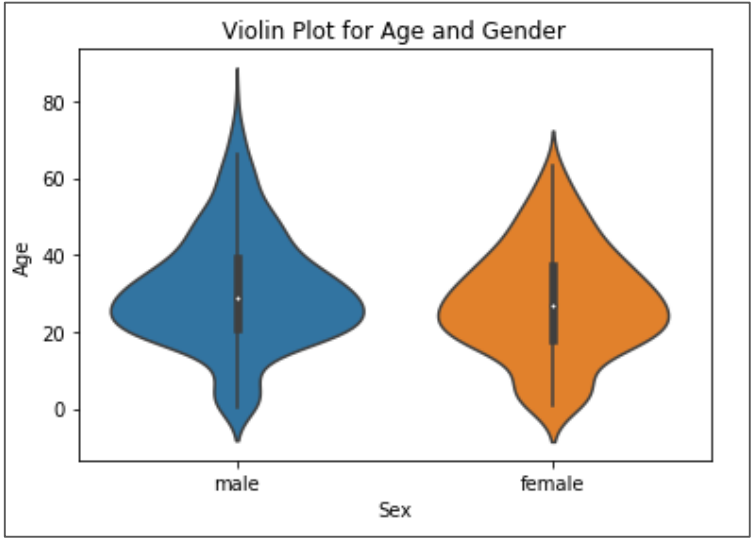
```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

# display first five rows
df_titanic.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

Violin plot

Plot the violin plot to compare the distribution of age based on gender

```
# plot a violin plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.violinplot(x = 'Sex', y = 'Age', data = df_titanic)

# add the plot label
plt.title('Violin Plot for Age and Gender')

# display the plot
plt.show()
```



The box plot is plotted inside the violin plot

Violin plot

Violin plot can be divided into two halves, where one half represents surviving while other half represents the non-surviving passenger

```
# plot a violin plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'hue' adds one more variable to the plot
# 'data' represents the DataFrame
# 'split' returns the plot splitted in two halves
sns.violinplot(x='Sex', y='Age', data=df_titanic, hue='Survived', split=True)

# add the plot label
plt.title('Survival based on Age and Gender')

# display the plot
plt.show()
```

Pass ‘True’ as value for the split parameter

Swarm plot

- It is the combination of strip and violin plots
- The points are adjusted in such a way that they don't overlap, which gives the better representation of the data

Read the data

Load the titanic data to create a swarm plot

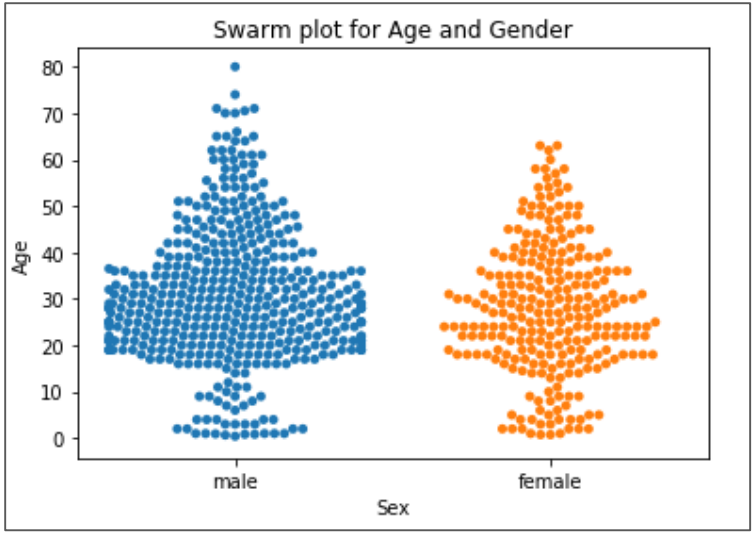
```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

# display first five rows
df_titanic.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

Swarm plot

Create a swarm plot for the distribution of age based on gender

```
# plot a swarm plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.swarmplot(x = 'Sex', y = 'Age', data = df_titanic)

# add the plot label
plt.title('Swarm plot for Age and Gender')

# display the plot
plt.show()
```

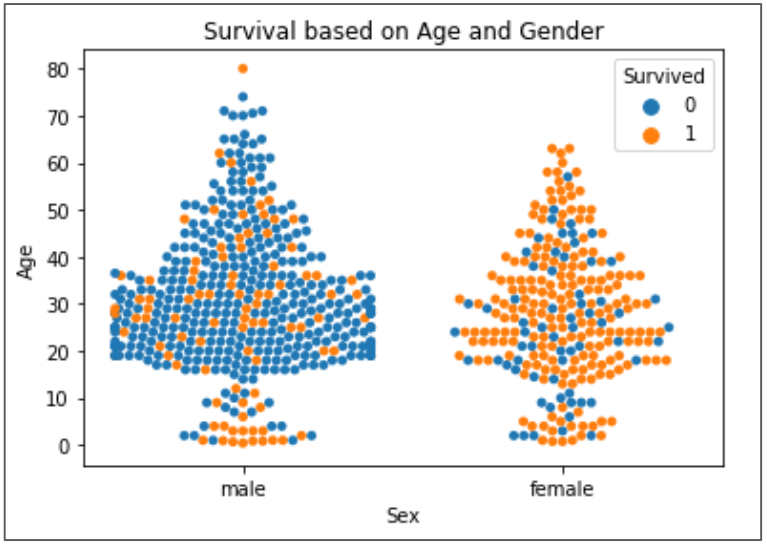


High proportion of males are in the age range 20 - 40

Swarm plot

Add one more categorical variable ‘Survived’ to the swarm plot using the parameter, ‘hue’

```
# plot a swarm plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'hue' adds one more variable to the plot
# 'data' represents the DataFrame
sns.swarmplot(x = 'Sex', y = 'Age', data = df_titanic, hue = 'Survived')

# add the plot label
plt.title('Survival based on Age and Gender')

# display the plot
plt.show()
```



Pair plot

- It displays the pairwise relationship between the numeric variables
- The pairplot() method creates a matrix; where the diagonal plots represent the univariate distribution of each variable and the off-diagonal plots represent the scatter plot of the pair of variables

Read the data

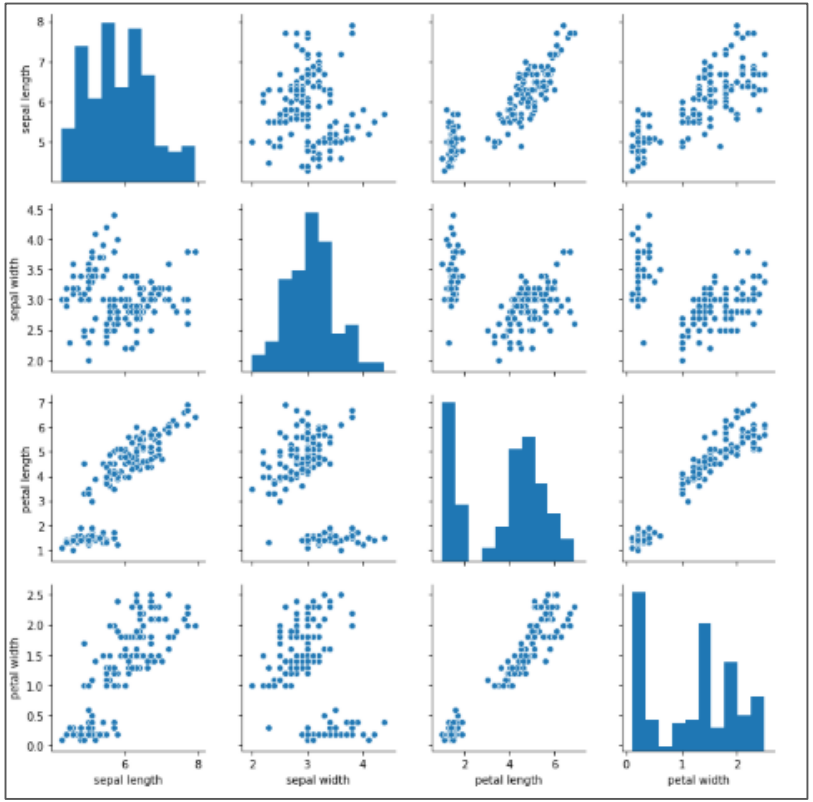
Use the iris data to create the pair plot

```
# load the csv file 'iris.csv'
df_iris = pd.read_csv('iris.csv')

# display first five rows
df_iris.head()
```

	sepal length	sepal width	petal length	petal width	class
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Pair plot

```
# plot a pair plot
# 'data' represents the data to plot the pair plot
sns.pairplot(data = df_iris)

# display the plot
plt.show()
```



Distribution plot

- It displays the distribution of the data
- It is a variation of histogram that uses kernel smoothing to plot values, allowing for smoother distributions by smoothing out the noise

Read the data

greatlearning
Learning for Life

Load the titanic data to create a distribution plot

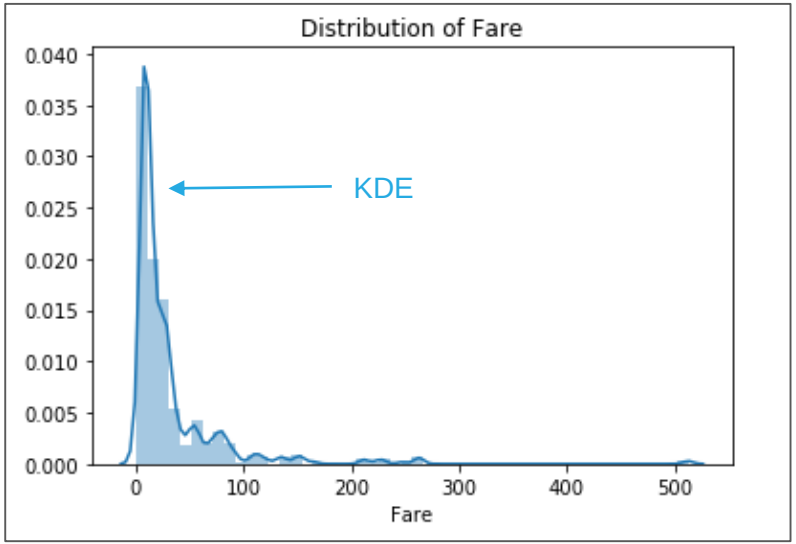
```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

# display first five rows
df_titanic.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

Distribution plot

The distplot() method plots the histogram with a Kernel Density Estimator (KDE), which is a used to estimate the probability distribution function of a random variable

```
# plot a distribution plot
# 'a' represents variable to plot a distribution plot
sns.distplot(a = df_titanic['Fare'])

# add the plot label
plt.title('Distribution of Fare')

# display the plot
plt.show()
```



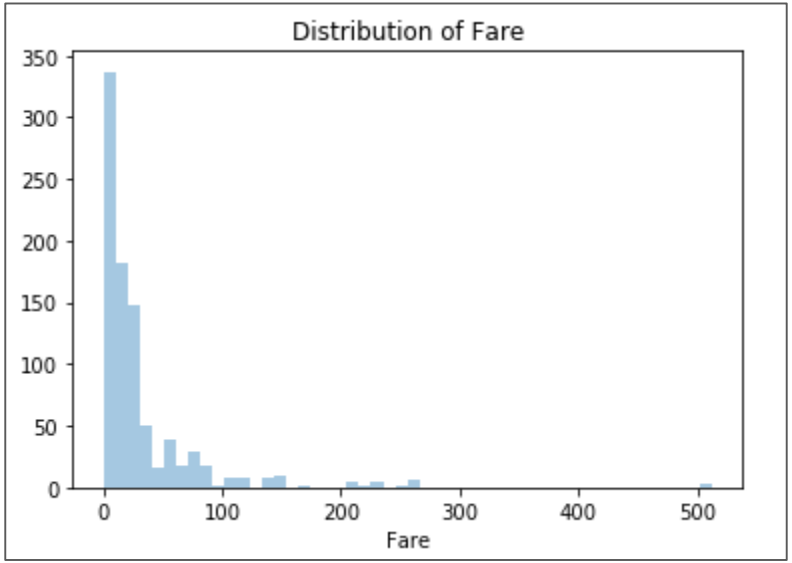
The plot shows the positive skewness of the ‘Fare’

Distribution plot

Plot the distribution of ‘Fare’ without the kernel density estimator (KDE)

```
# plot a distribution plot
# 'a' represents variable to plot a distribution plot
sns.distplot(a = df_titanic['Fare'], kde = False)

# add the plot label
plt.title('Distribution of Fare')

# display the plot
plt.show()
```

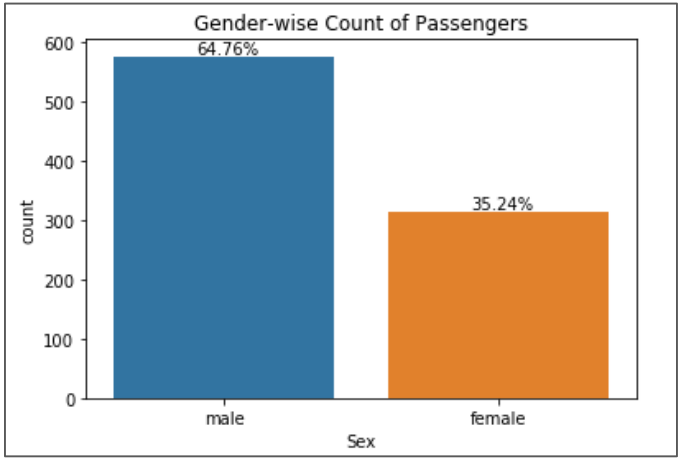
Returns the plot without kde



Count plot

It is similar to the bar plot. However, it shows the count of the categories in a specific variable

```
# plot a count plot
# 'x' represents variable on x-axis
# 'data' represents the DataFrame
sns.countplot(x = 'Sex', data = df_titanic)

# add text on the plot
# 'x' and 'y' represents the position of the text
# 's' represents the text
plt.text(x = -0.1, y = 580, s = str(round(df_titanic.Sex.value_counts()[0]/len(df_titanic)*100, 2)) + '%')
plt.text(x = 0.9, y = 320, s = str(round(df_titanic.Sex.value_counts()[1]/len(df_titanic)*100, 2)) + '%')

# add the plot label
plt.title('Gender-wise Count of Passengers')

# display the plot
plt.show()
```

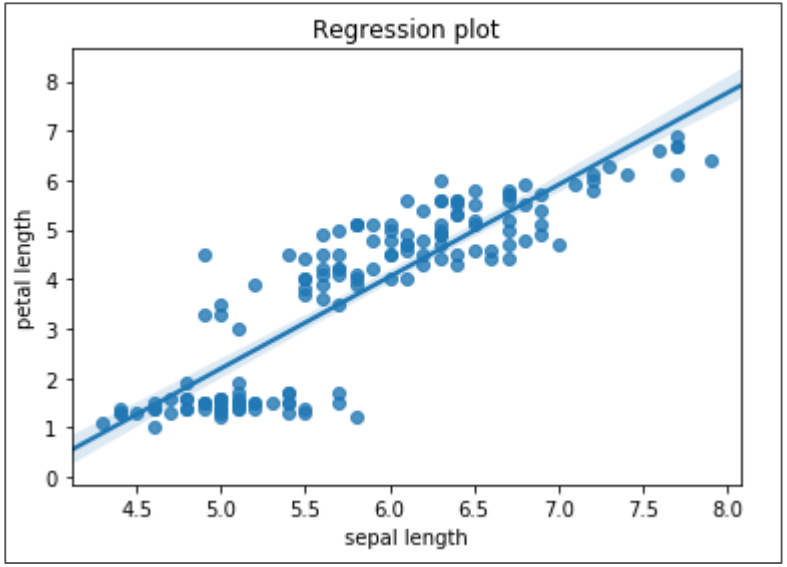


Calculate the gender-wise percentage upto 2 decimals

Regression plot

It is used to study the relationship between the two variables with the regression line

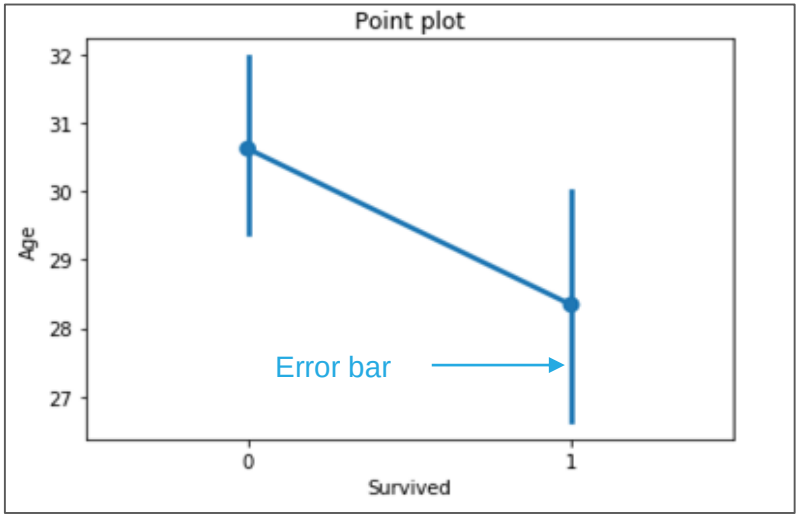
```
# plot a regression plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.regplot(x = 'sepal length', y = 'petal length', data = df_iris)

# add the plot label
plt.title('Regression plot')

# display the plot
plt.show()
```



Point plot

It represents an estimate of central tendency (by default, mean) by position of scatter points and provides the indication of the uncertainty around that estimate using error bars

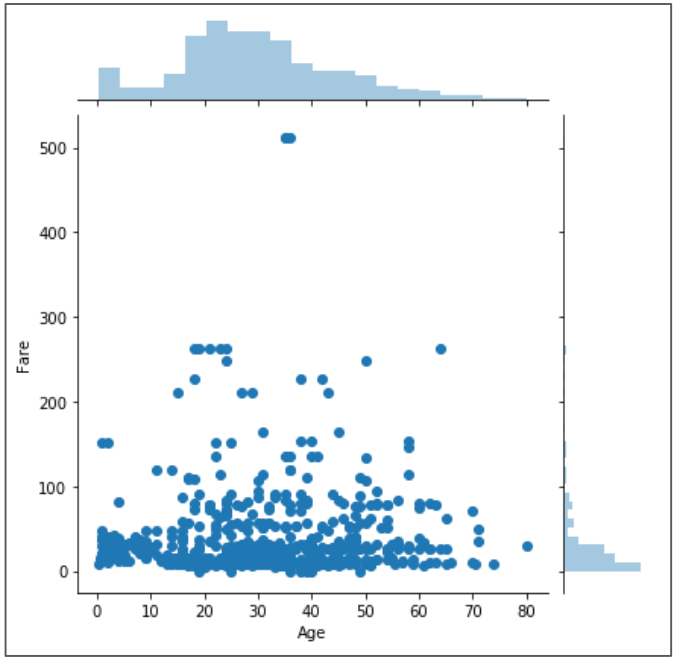
```
# plot a point plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.pointplot(x = 'Survived', y = 'Age', data = df_titanic)

# add the plot label
plt.title('Point plot')

# display the plot
plt.show()
```



Joint plot

A joint plot is a bivariate plot along with the distribution plot along the margins

```
# plot a joint plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.jointplot(x = 'Age', y = 'Fare', data = df_titanic)

# display the plot
plt.show()
```



Heatmap

- A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented by the different colors

- Heatmap for correlation shows the correlation between the variables on each axis

Read the data

Use the iris data to create the heatmap

```
# load the csv file 'iris.csv'
df_iris = pd.read_csv('iris.csv')

# display first five rows
df_iris.head()
```

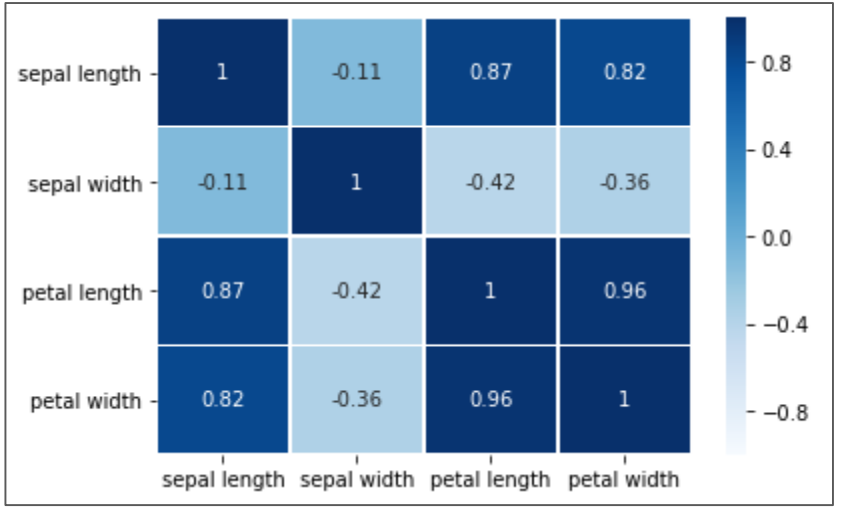
	sepal length	sepal width	petal length	petal width	class
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Heatmap

```
# plot heatmap to study correlation
# 'data' returns the data for heatmap
# 'annot' returns the correlation values on heatmap
# 'linewidth' add lines between each cells
# 'cmap' assigns the colors to each cell
# 'cbar' returns the color bar beside the heatmap
# 'vmin' and 'vmax' assigns the minimum and maximum values to anchor the color bar
sns.heatmap(data = df_iris.corr(), annot = True, linewidth=0.5,
            cmap = 'Blues', cbar = True, vmin = -1, vmax = 1 )

# display the plot
plt.show()
```

Assigns color to each cell

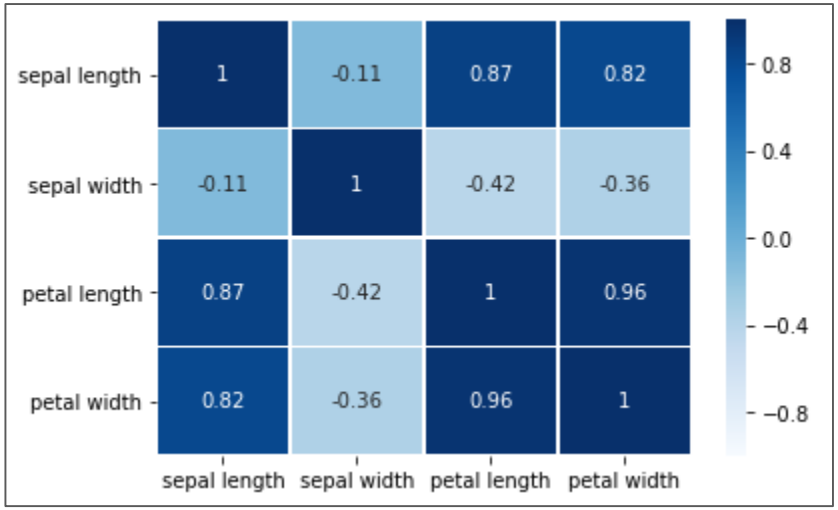
Add values to the heatmap



The variables ‘petal width’ and ‘petal length’ are highly positively correlated

Heatmap

- Diagonal cells represent the correlation of the variable with itself; thus, the value will always equal to 1
- The off-diagonal entries represent the correlation between the pair of variables
- The color bar beside the heatmap shows that the dark blue color represents the positive correlation (near to +1) and light blue color represent the negative correlation (near to -1)



Visualization using Plotly

Plotly

- Plotly allows us to make beautiful, interactive, explorable charts
- It is an open-source and browser-based graphing library
- We can get the values on the graph by hovering over the graph

Plotly

- Import the offline version of plotly as:

```
import plotly
plotly.offline.init_notebook_mode(connected=True)
```

- Import the subpackage ‘express’ as:

```
# import the library
import plotly.express as px
```

Read the data

greatlearning
Learning for Life

Load the titanic data for further manipulations:

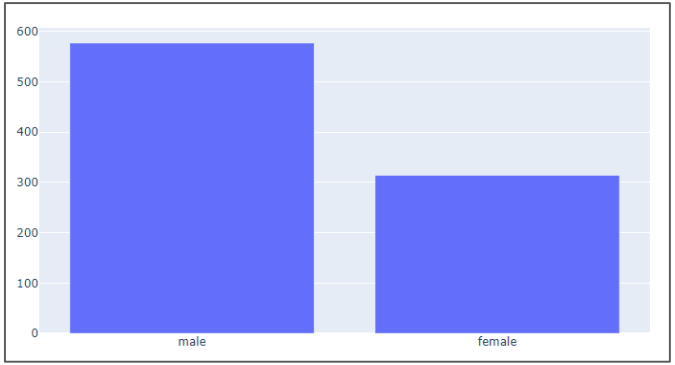
```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

# display first five rows
df_titanic.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

Bar plot

Plot a bar plot to get the gender-wise count of passengers

```
# import the required subpackage
import plotly.graph_objs as go

# instantiate the plot object
# set the size of plot
fig = go.Figure(layout={'autosize':False, 'height':500, 'width':800})

# plot the bar plot
# 'add_trace' adds the trace to the plot
# 'name' sets the trace name
fig.add_trace(go.Bar(x = df_titanic['Sex'], y = df_titanic.Sex.value_counts(), name = 'Sex'))
```

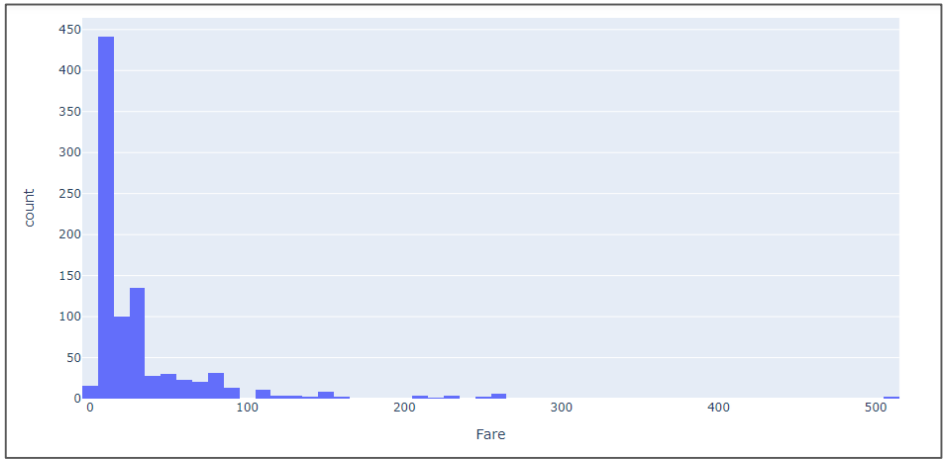


Male population is more in the data than female population

Histogram

Plot a distribution of fare of the passengers

```
# plot the histogram to check the distribution of fare
# 'x' assigns the variable to plot a histogram
fig = px.histogram(data_frame = df_titanic, x="Fare")

# display the plot
fig.show()
```



The histogram shows the positive skewness in the ‘Fare’. There are few extreme points near to Fare = 500

Read the data

greatlearning
Learning for Life

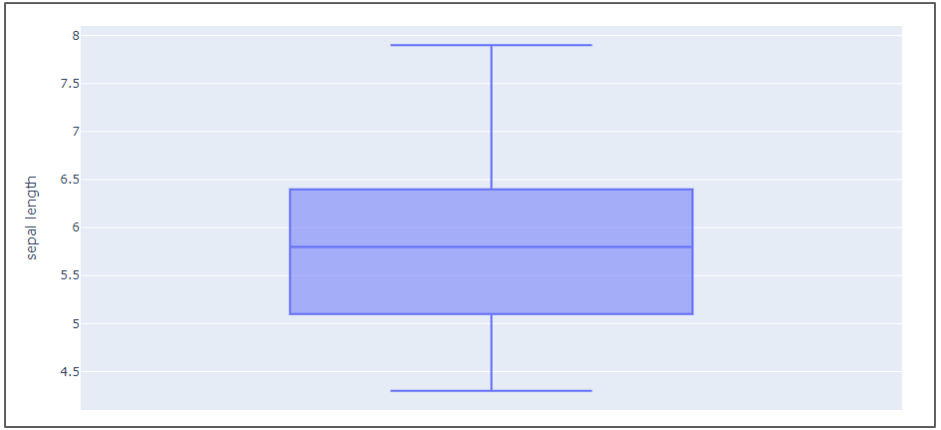
Use the iris data for further manipulations:

```
# load the csv file 'iris.csv'
df_iris = pd.read_csv('iris.csv')

# display first five rows
df_iris.head()
```

	sepal length	sepal width	petal length	petal width	class
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Boxplot

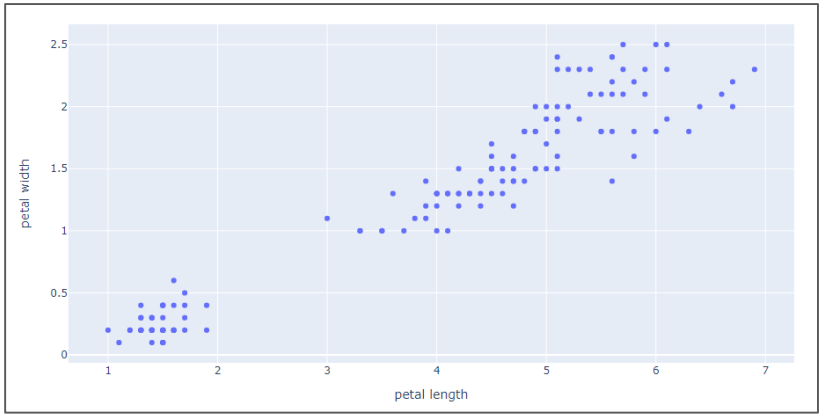
```
# plot a boxplot
# 'y' assigns the variable to plot a boxplot
fig = px.box(df_iris, y ='sepal length')

# display the plot
fig.show()
```



The boxplot shows that, the ‘sepal length’ is not significantly skewed

Scatter plot

```
# plot a boxplot
# 'data' assigns the DataFrame
# 'x' assigns the variable on x-axis
# 'y' assigns the variable on y-axis
fig = px.scatter(df_iris, x = 'petal length', y = 'petal width')

# display the plot
fig.show()
```



The plot shows that, the variables ‘petal length’ and ‘petal width’ are positively correlated

greatlearning
Learning for Life

Thank You