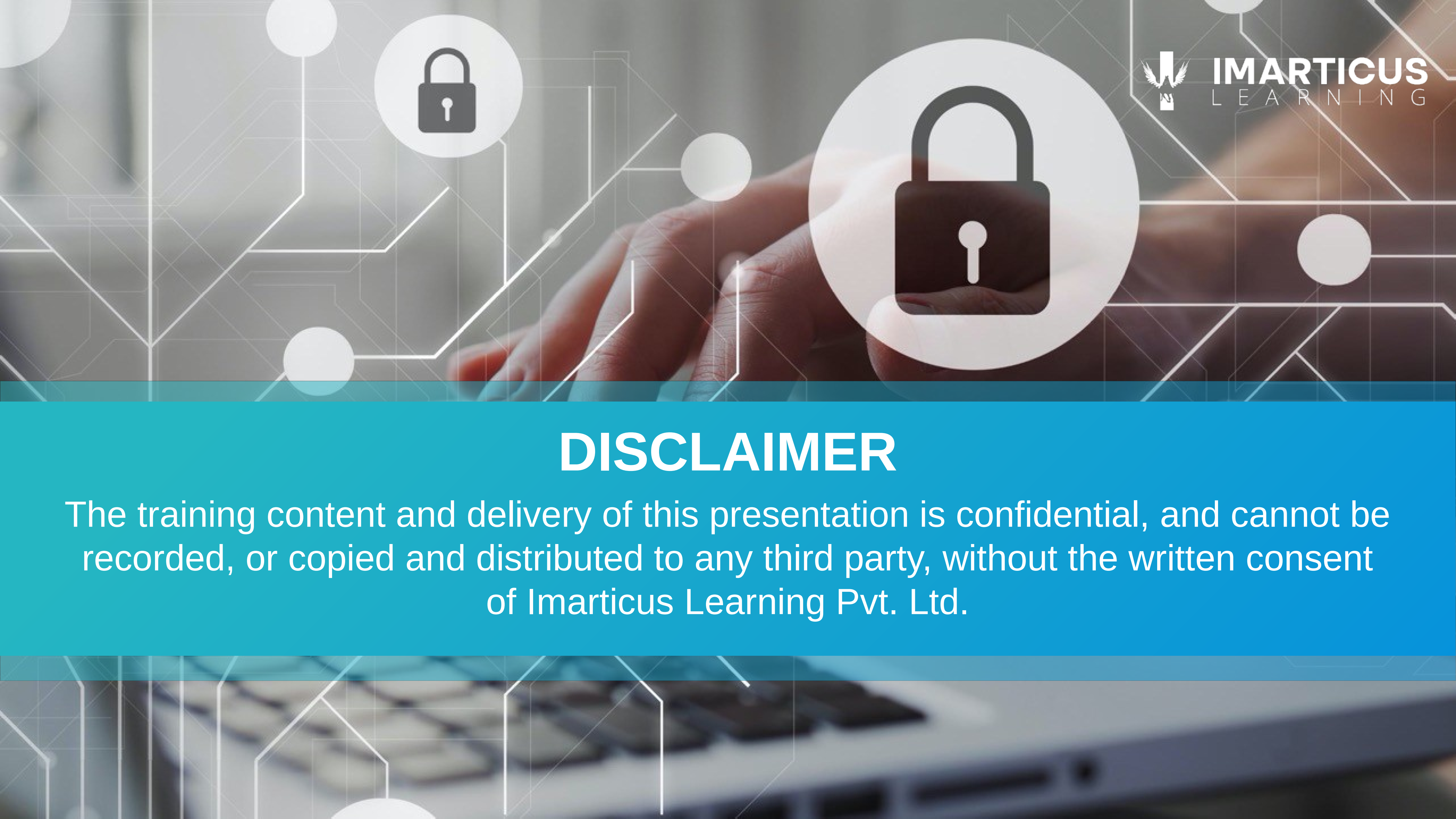
IMARTICUS
LEARNING

Python Programming

Visualization-Part 2



IMARTICUS
LEARNING

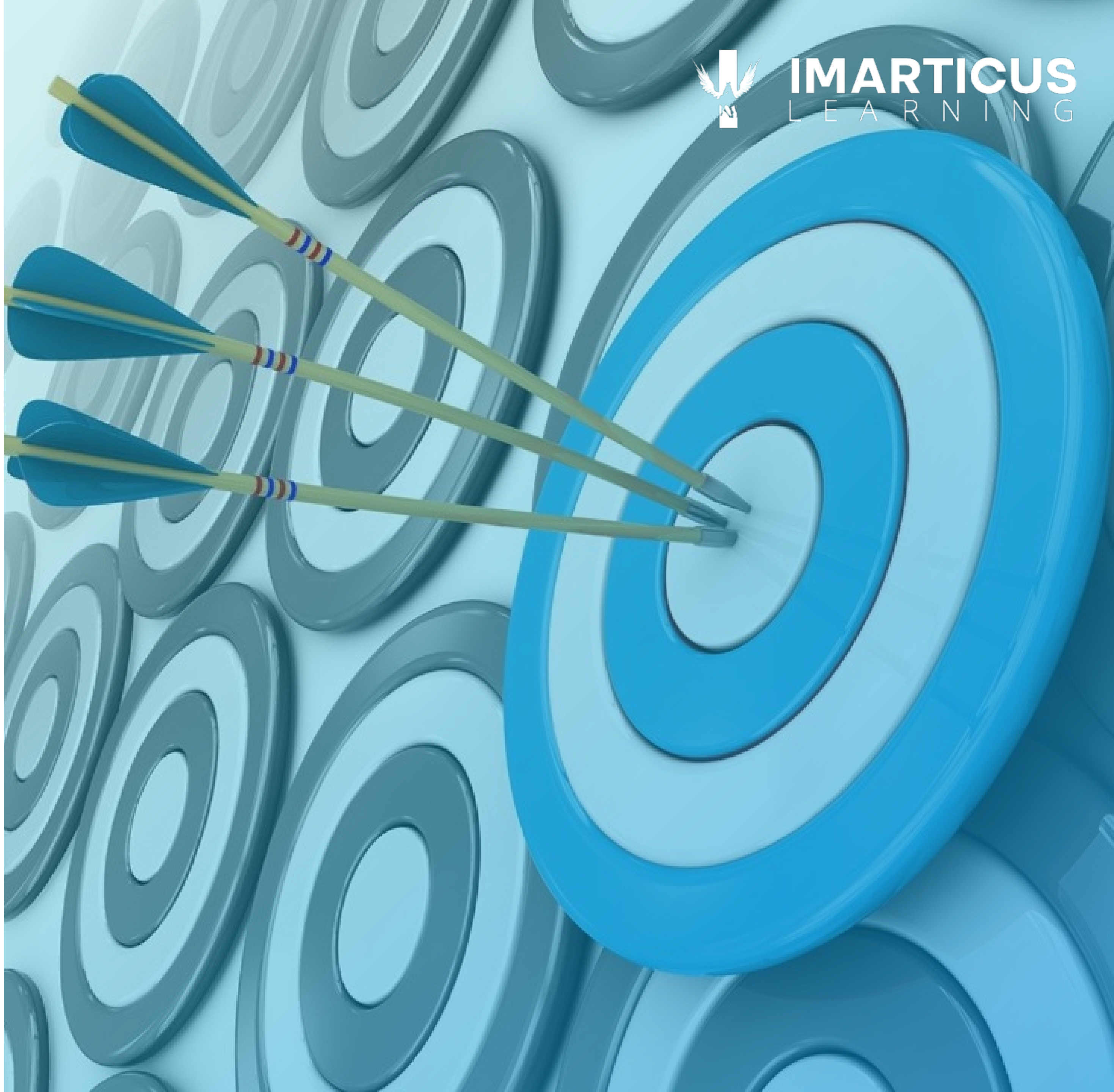
DISCLAIMER

The training content and delivery of this presentation is confidential, and cannot be recorded, or copied and distributed to any third party, without the written consent of Imarticus Learning Pvt. Ltd.

LEARNING OBJECTIVES

At the end of this session, you will learn:

- Visualization using Seaborn
- Strip Plot
- Distribution plot
- Joint plot
- Violin plot
- Swarm plot
- Pair plot
- Count plot
- Heatmap



Visualization using Seaborn

INTRODUCTION

Seaborn is a data visualization library built on top of Matplotlib

FUNCTIONALITIES OF SEABORN

1. Allows comparison between multiple variables
2. Supports multi-plot grids
3. Univariate and bivariate visualization
4. Availability of different color palettes
5. 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
```

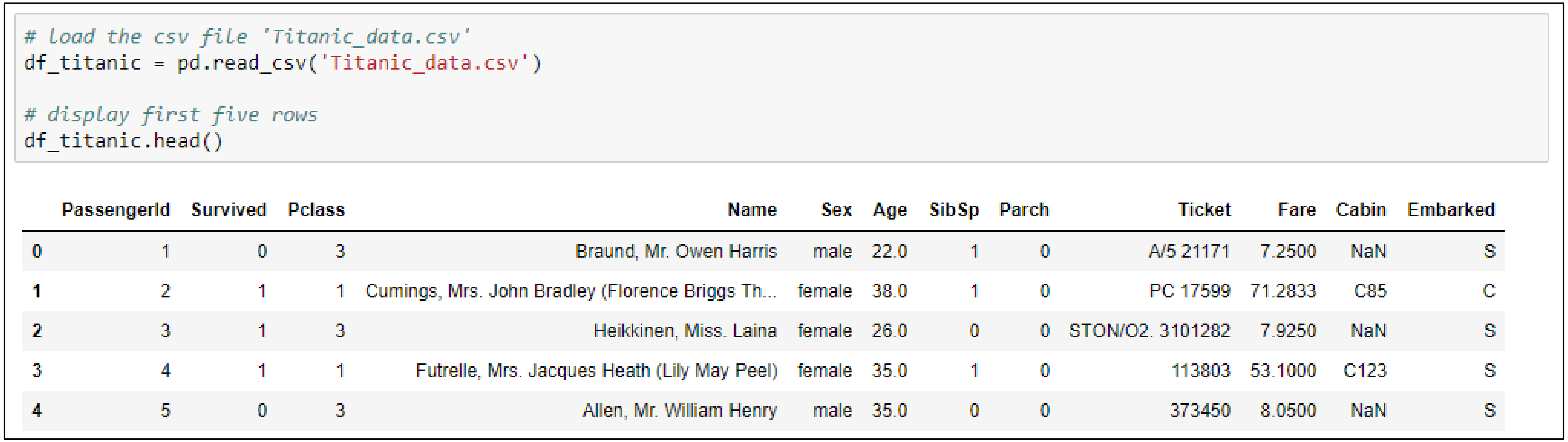
Strip Plot

STRIP PLOT

1. It is similar to the scatter plot with one categorical variable
2. It is similar to the scatter plot with one categorical variable
3. One axis represents the categorical variable and another represents the value corresponding to the categories

STRIP PLOT

Load the titanic data to create a strip plot

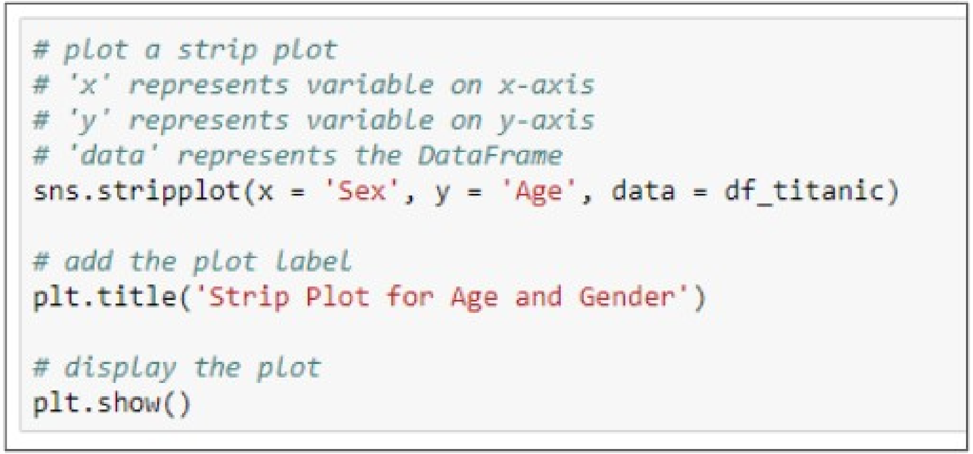


```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

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

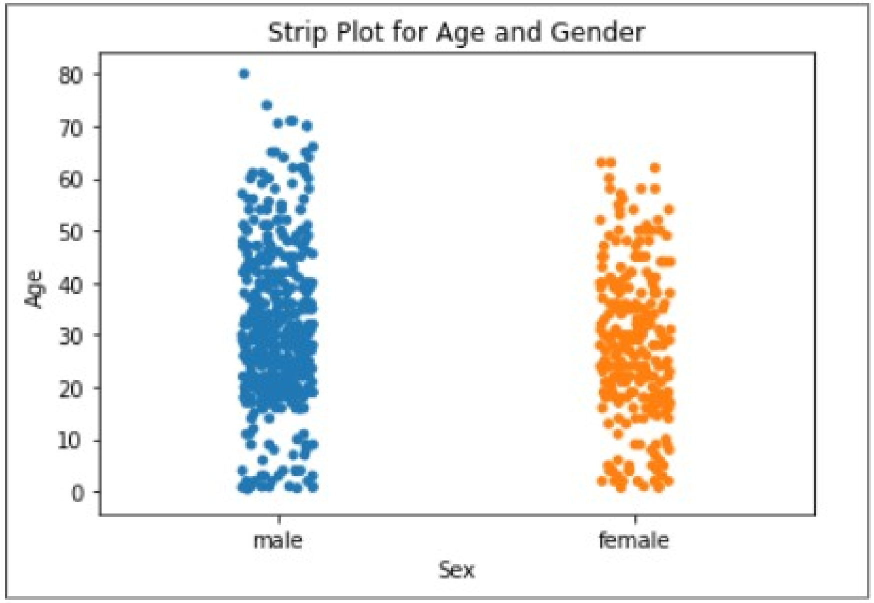
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 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



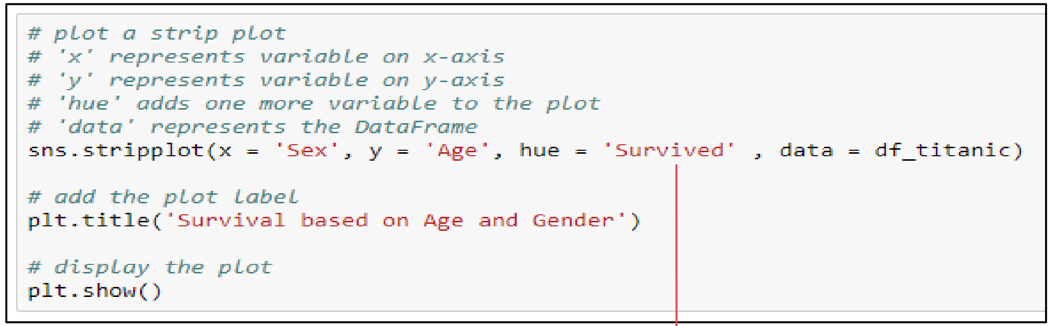
```
# plot a strip plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.stripplot(x = 'Sex', y = 'Age', data = df_titanic)

# 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

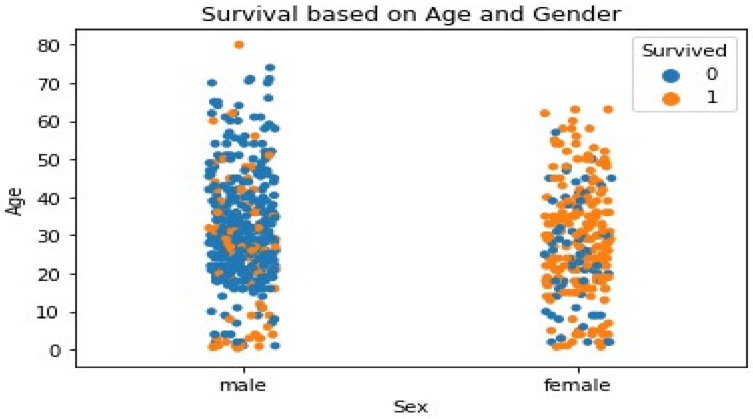
STRIP PLOT

Add 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

Distribution Plot

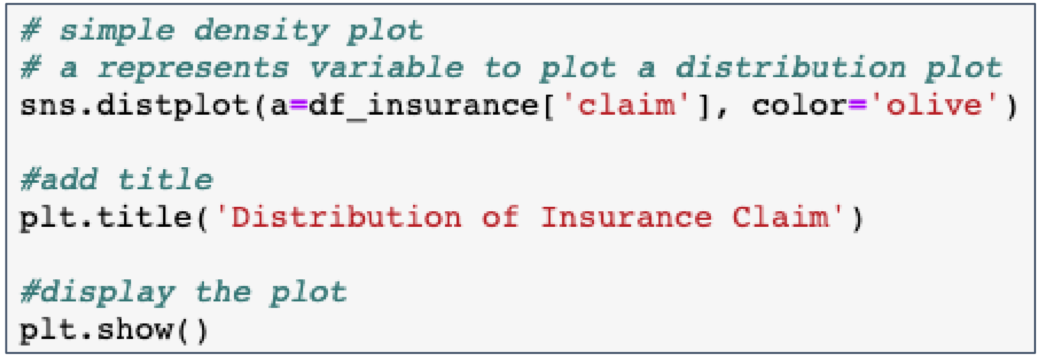
DISTRIBUTION PLOT

1. It displays the distribution of the data

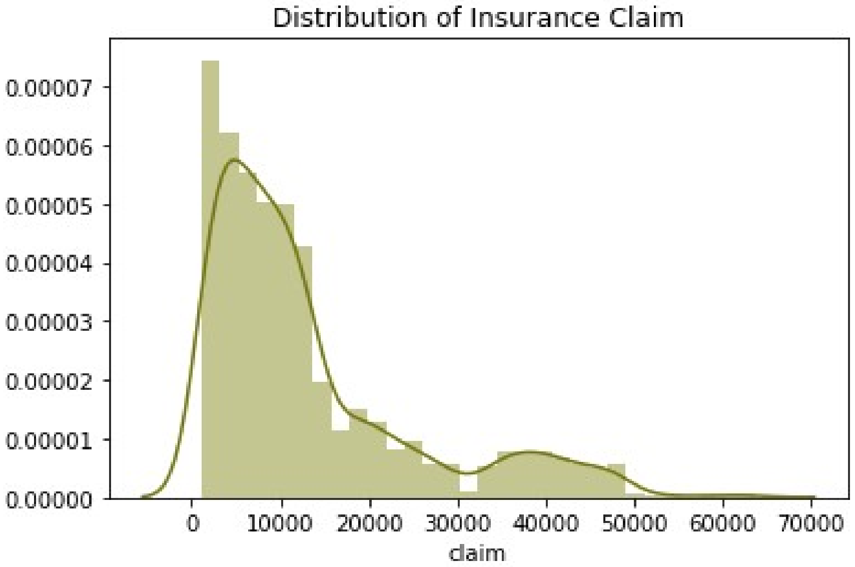
2. It is a variation of histogram that uses kernel smoothing to plot values, allowing for smoother distributions by smoothing out the noise

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



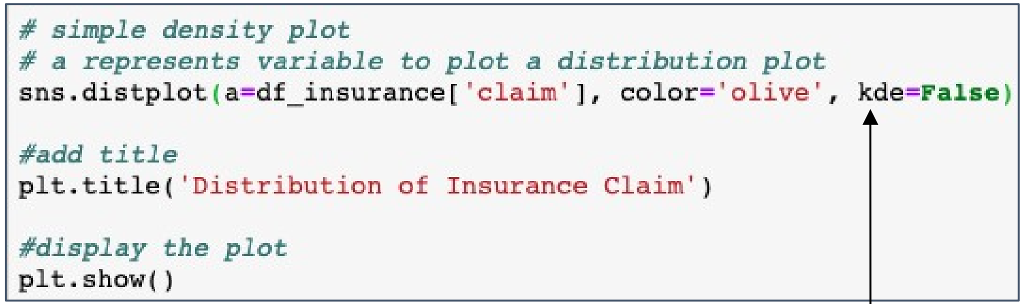
```
# simple density plot
# a represents variable to plot a distribution plot
sns.distplot(a=df_insurance['claim'], color='olive')

#add title
plt.title('Distribution of Insurance Claim')

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



The plot shows the positive skewness of the 'claim' variable

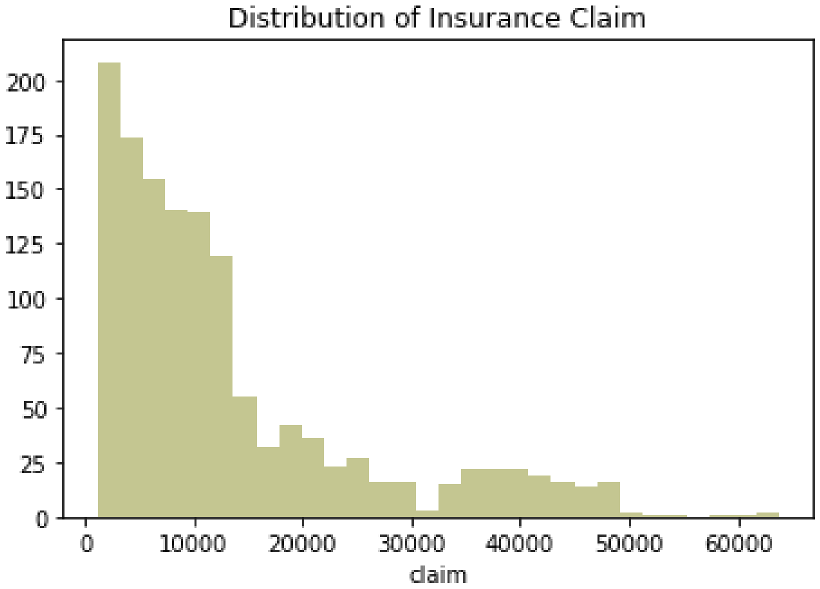
DISTRIBUTION PLOT

Plot the distribution of Sales without the kernel density estimator (KDE)



```
# simple density plot
# a represents variable to plot a distribution plot
sns.distplot(a=df_insurance['claim'], color='olive', kde=False)

#add title
plt.title('Distribution of Insurance Claim')

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

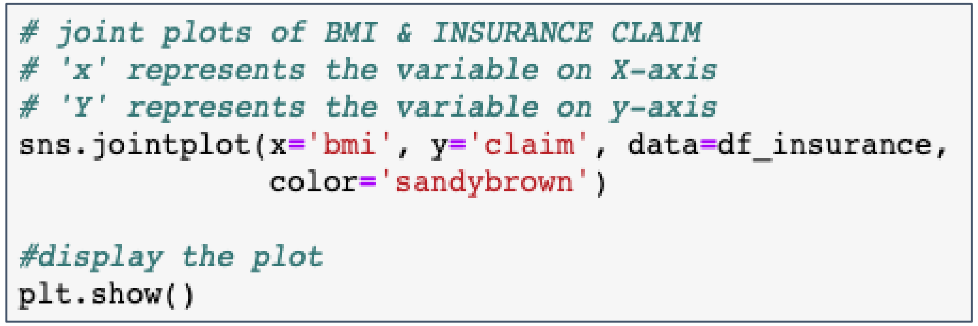
Returns the plot without kde



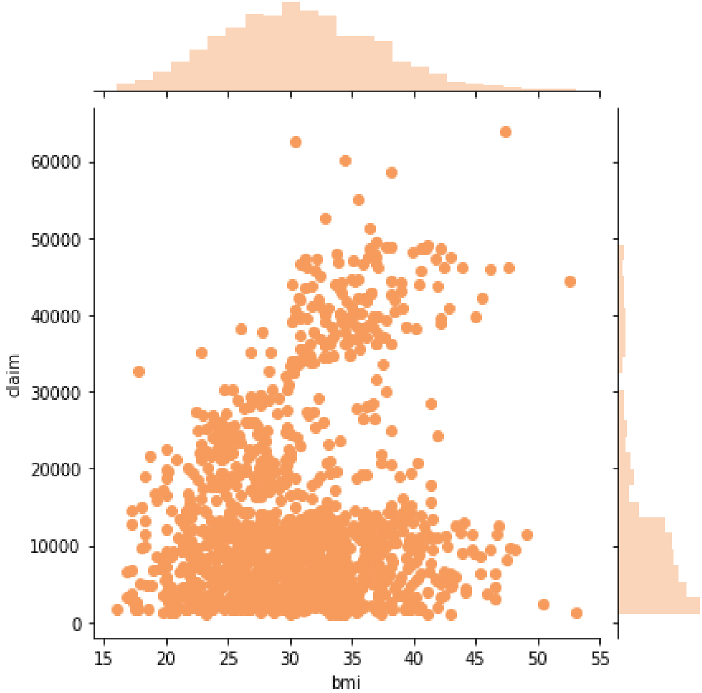
Joint Plot

JOINT PLOT

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



```
# joint plots of BMI & INSURANCE CLAIM
# 'x' represents the variable on X-axis
# 'Y' represents the variable on y-axis
sns.jointplot(x='bmi', y='claim', data=df_insurance,
              color='sandybrown')

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



Violin Plot

VIOLIN PLOT

1

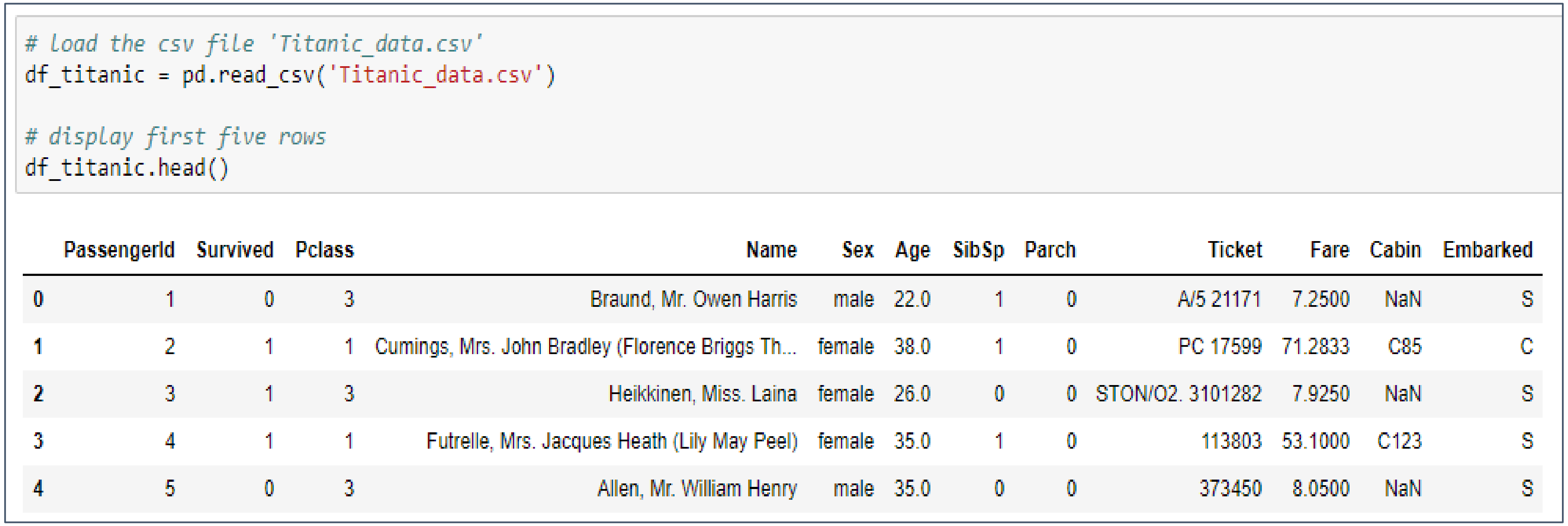
- It is similar to a boxplot, that displays the kernel density estimator of the underlying distribution

2

- 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

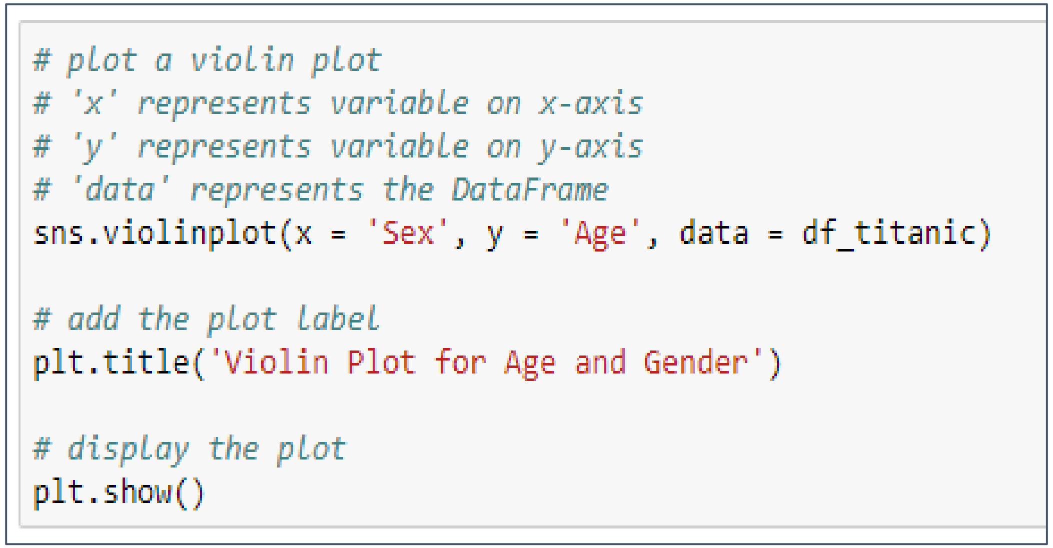


```python
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

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

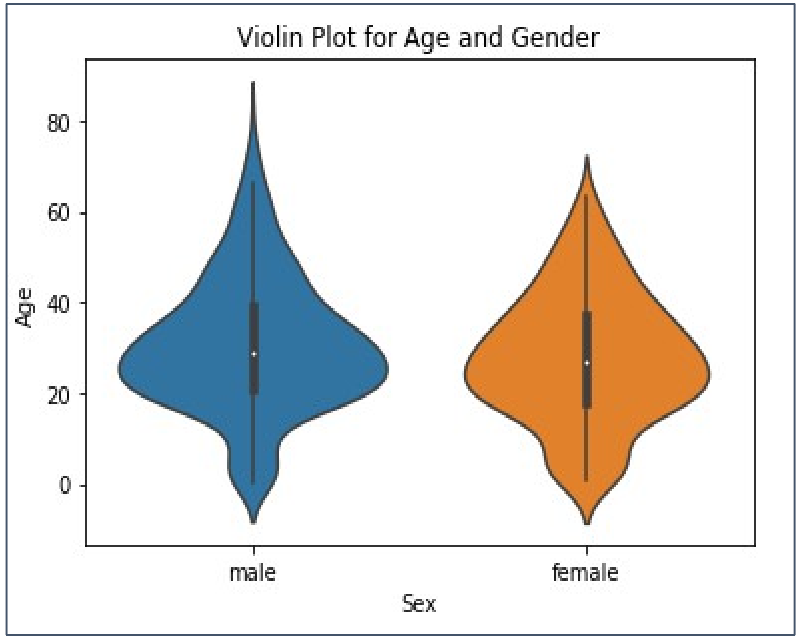
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 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

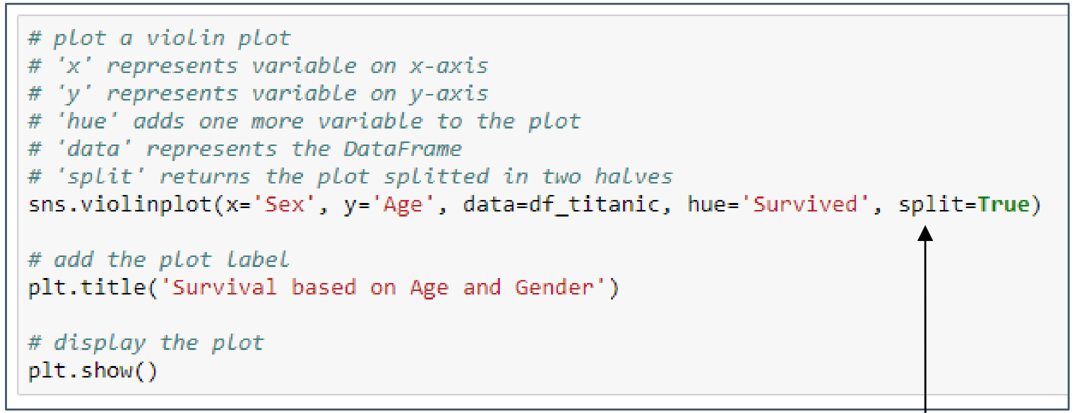


```
# plot a violin plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.violinplot(x = 'Sex', y = 'Age', data = df_titanic)

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

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



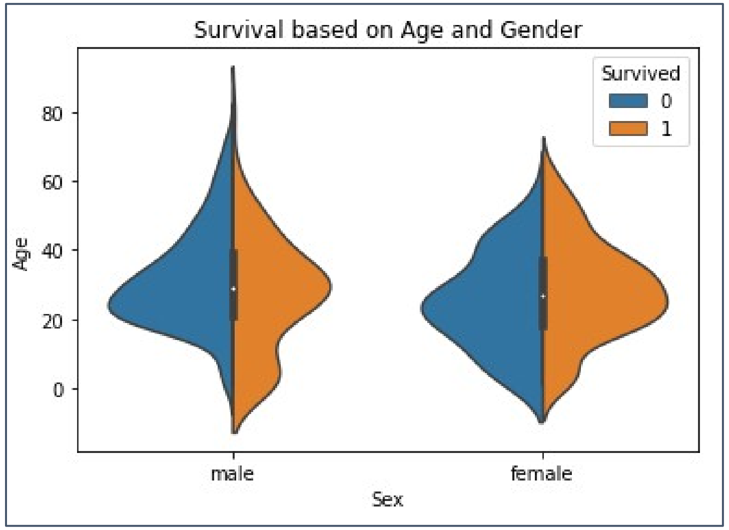
VIOLIN PLOT

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

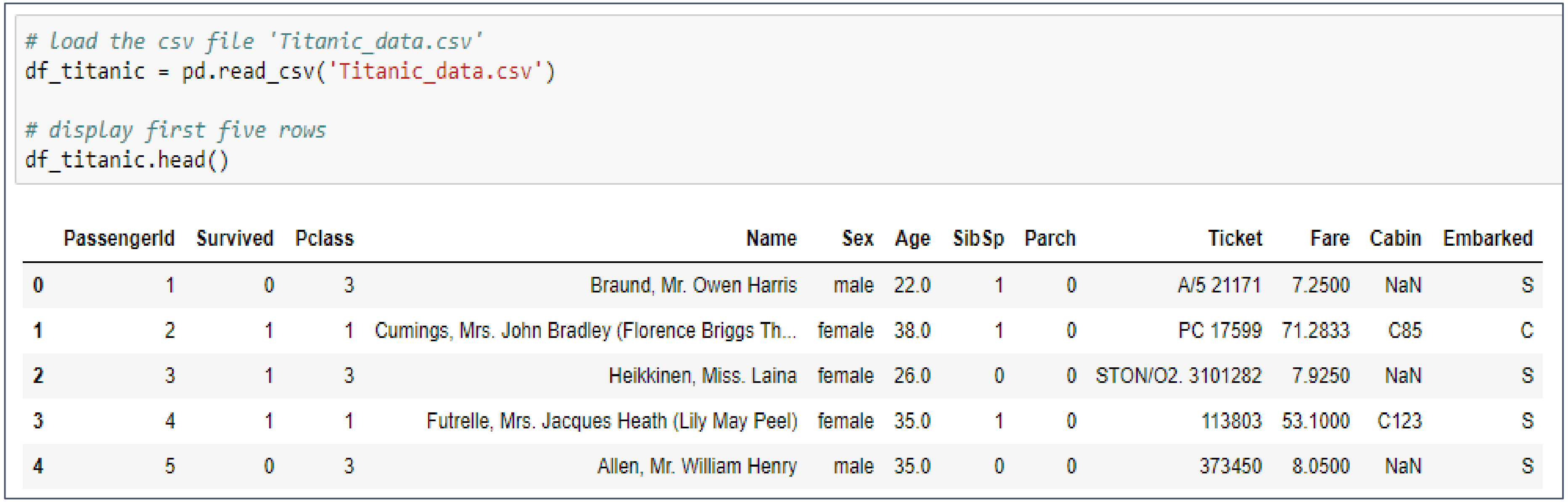
SWARM PLOT

1 It is the combination of strip and violin plots

2 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

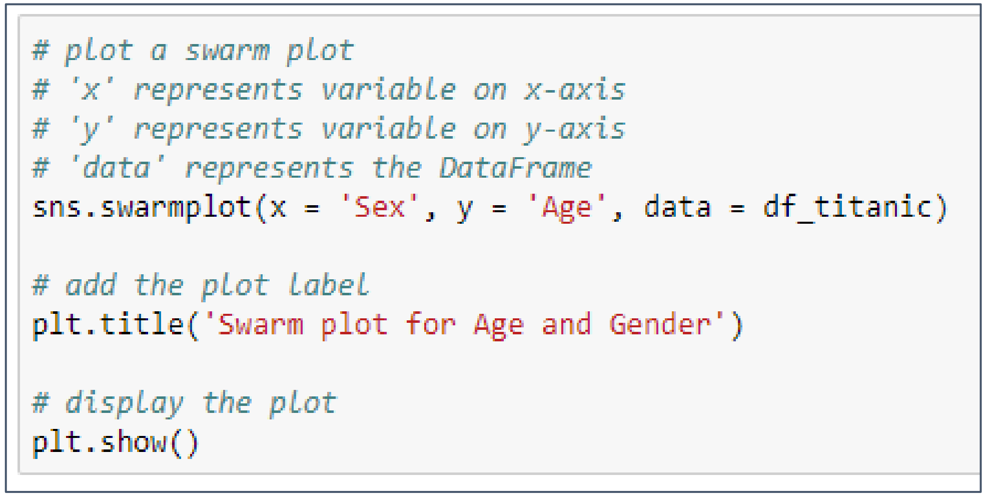


```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

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

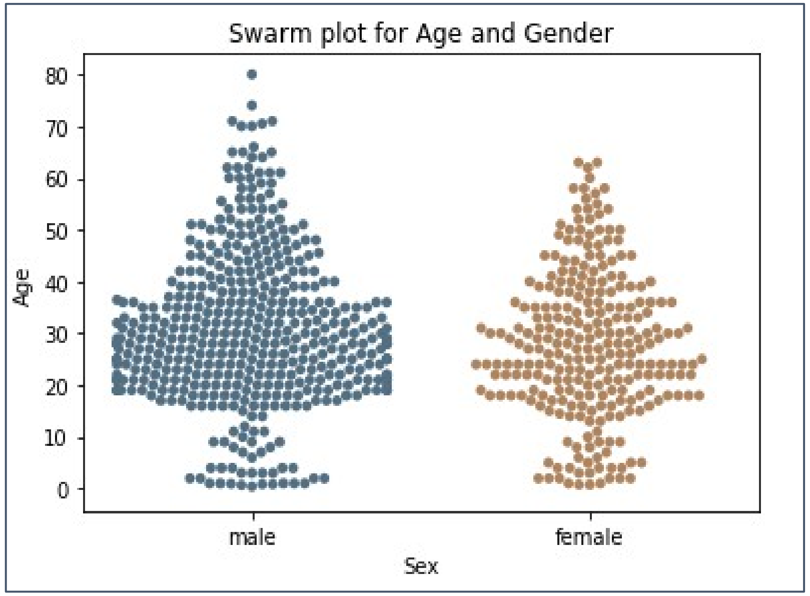
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 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

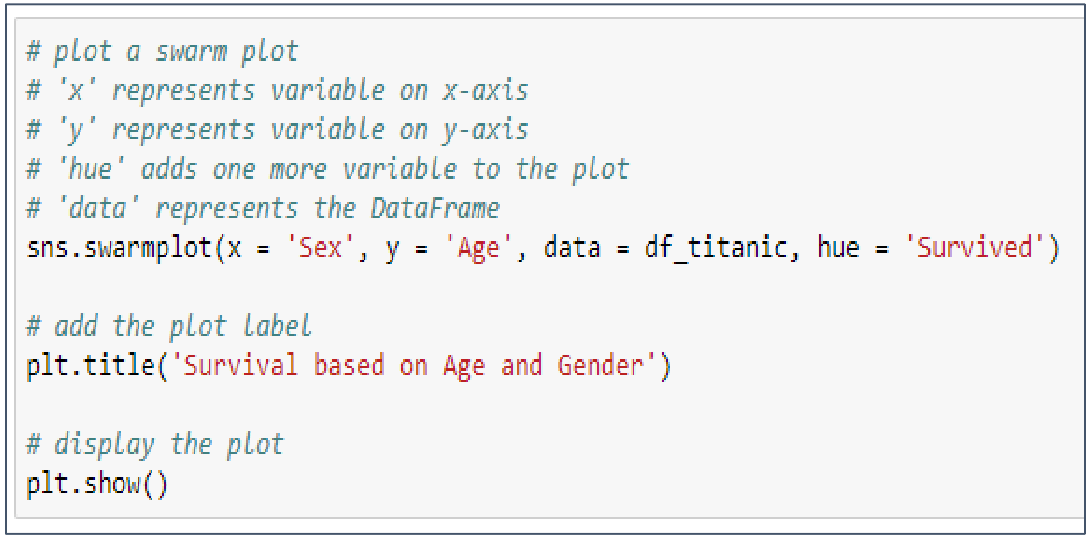


```
# plot a swarm plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.swarmplot(x = 'Sex', y = 'Age', data = df_titanic)

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

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

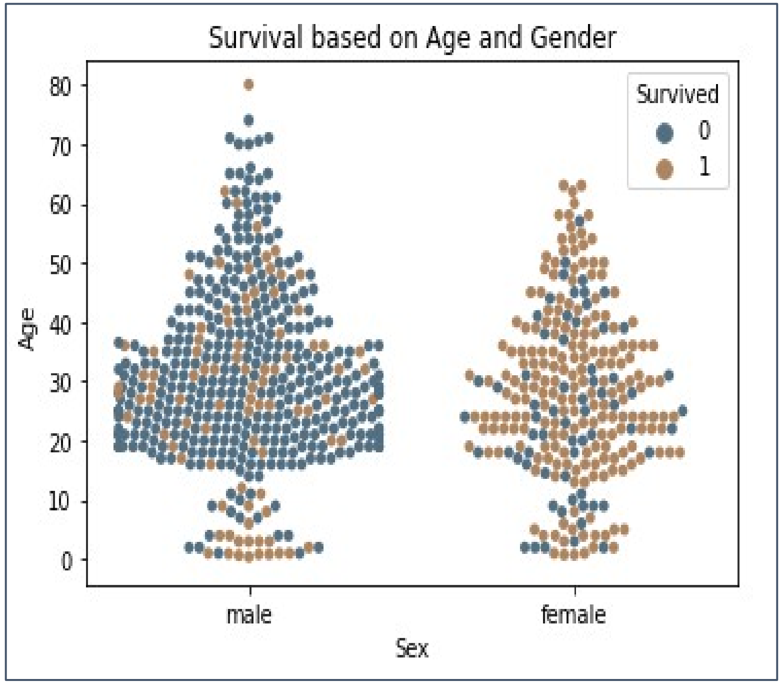


SWARM PLOT

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

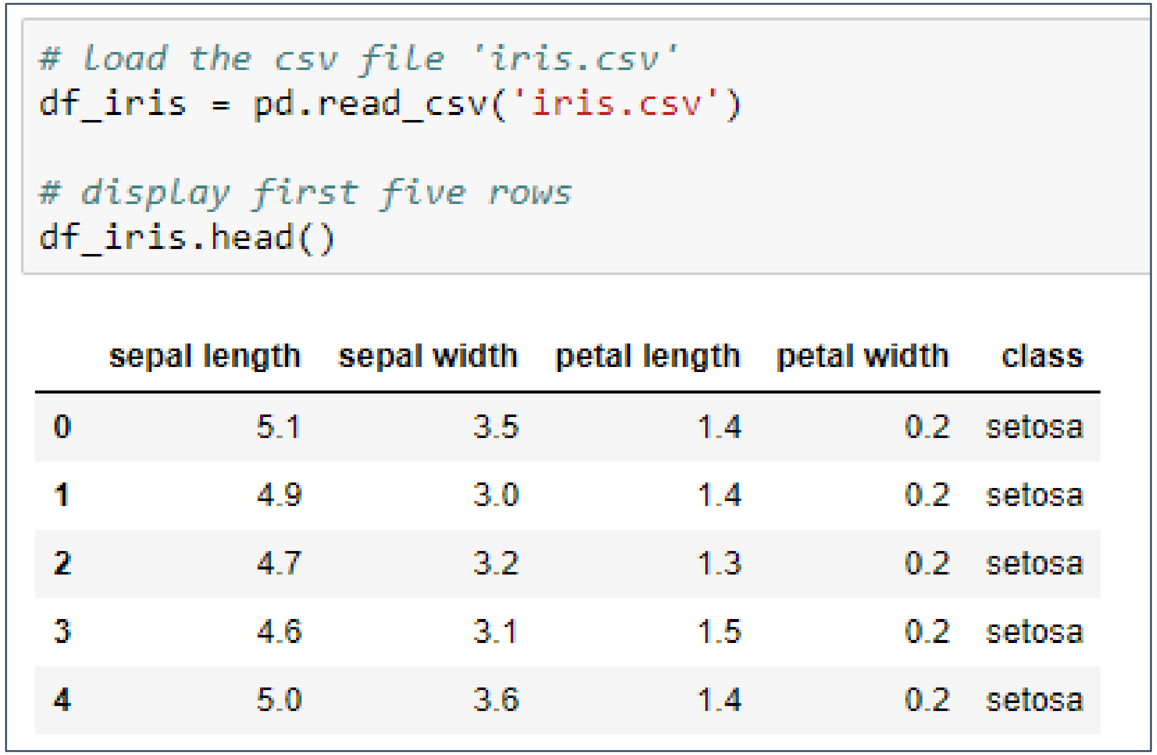
PAIR PLOT

1 It displays the pairwise relationship between the numeric variables

2 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

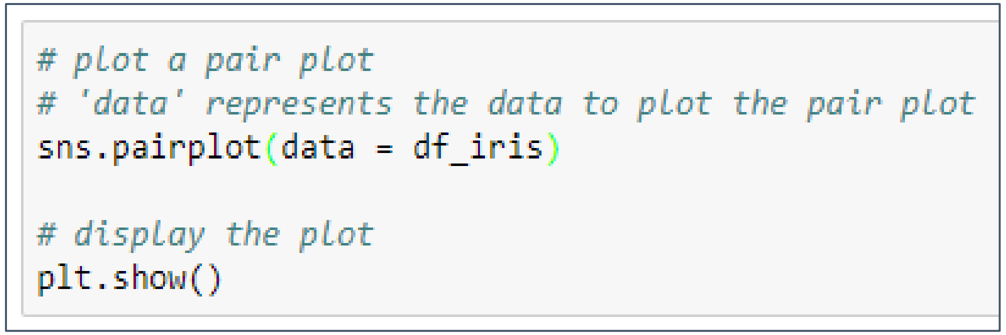
Use the iris data to create the pair plot



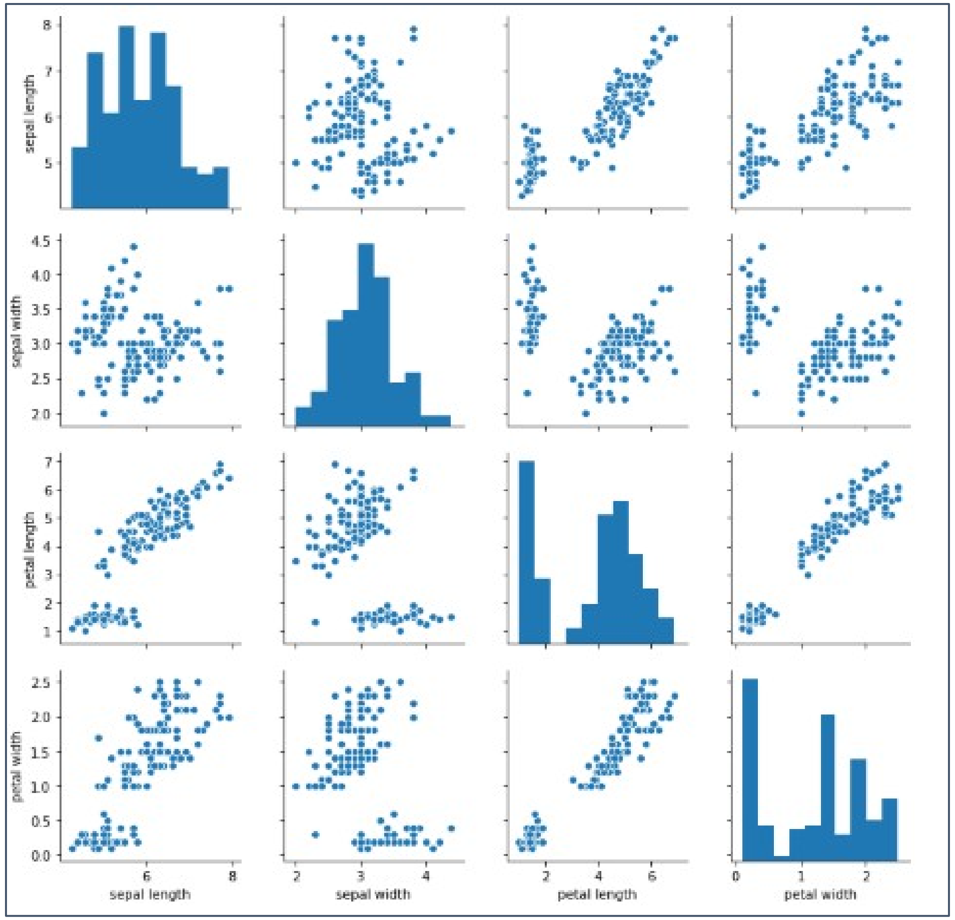
```
# load the csv file 'iris.csv'
df_iris = pd.read_csv('iris.csv')

# 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

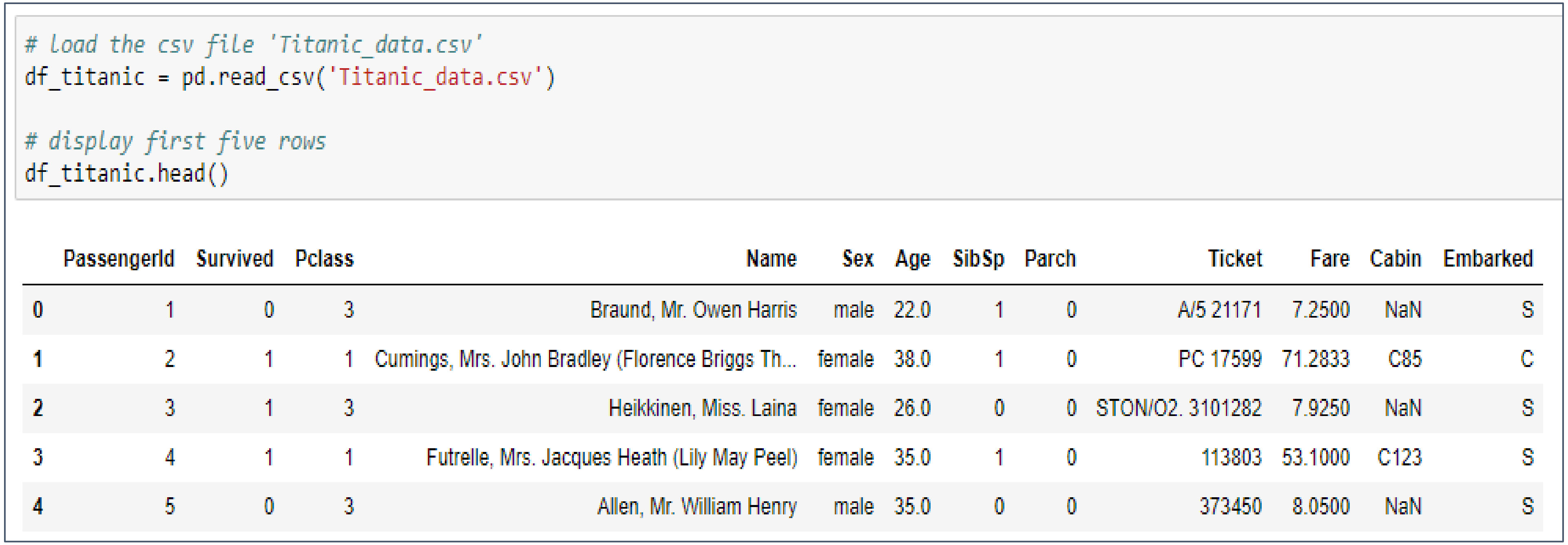


```
# plot a pair plot
# 'data' represents the data to plot the pair plot
sns.pairplot(data = df_iris)

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



READ THE DATA

Load the titanic data to create a count plot



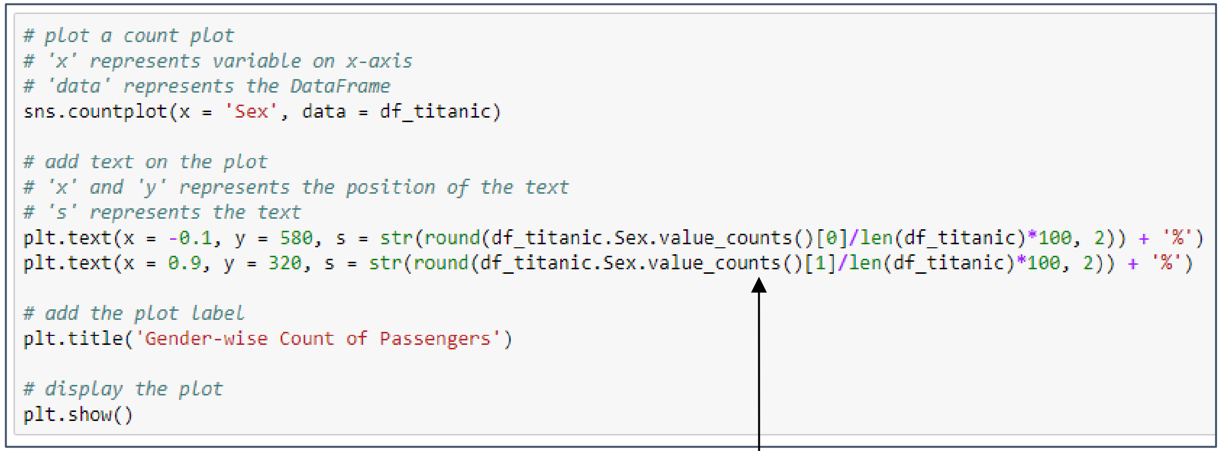
```
# load the csv file 'Titanic_data.csv'
df_titanic = pd.read_csv('Titanic_data.csv')

# 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

Count Plot

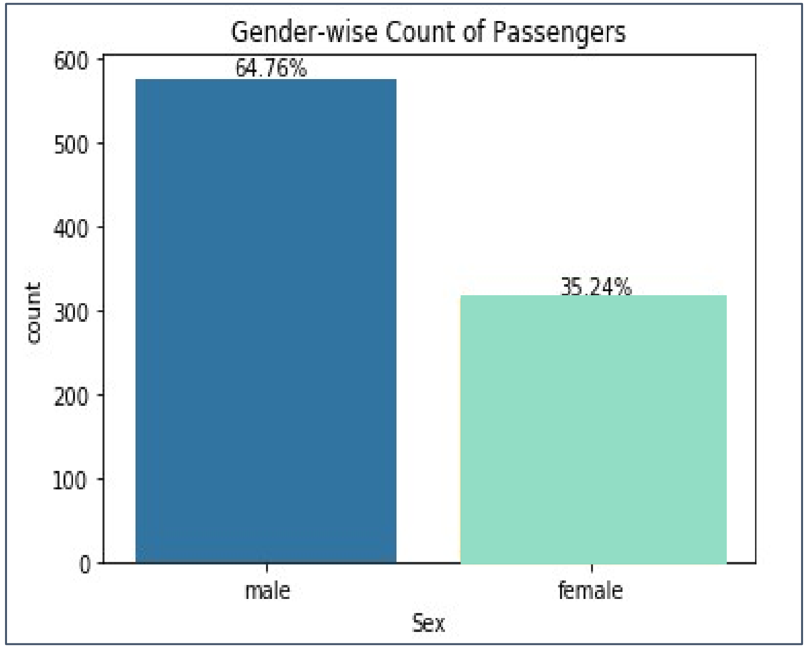
COUNT PLOT

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



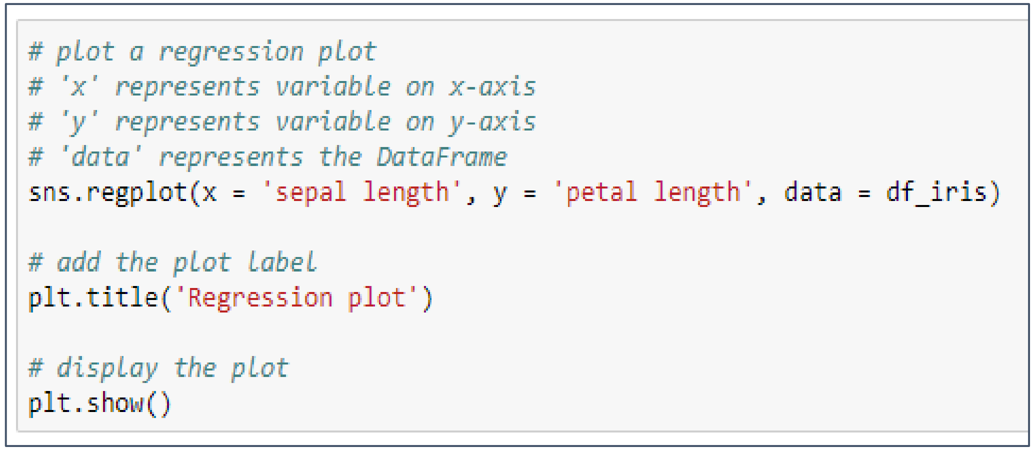
```
# plot a count plot
# 'x' represents variable on x-axis
# 'data' represents the DataFrame
sns.countplot(x = 'Sex', data = df_titanic)

# 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

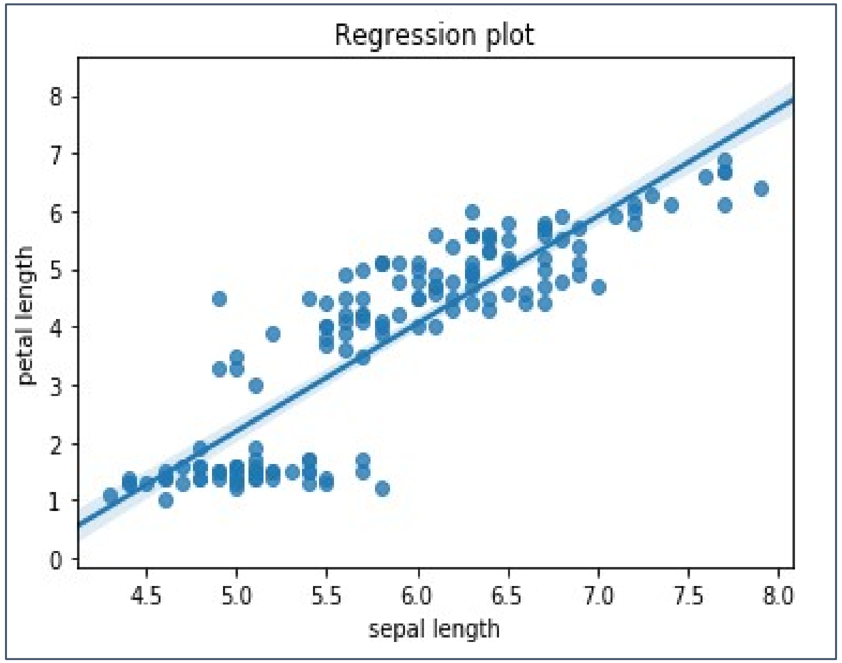


REGRESSION PLOT

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

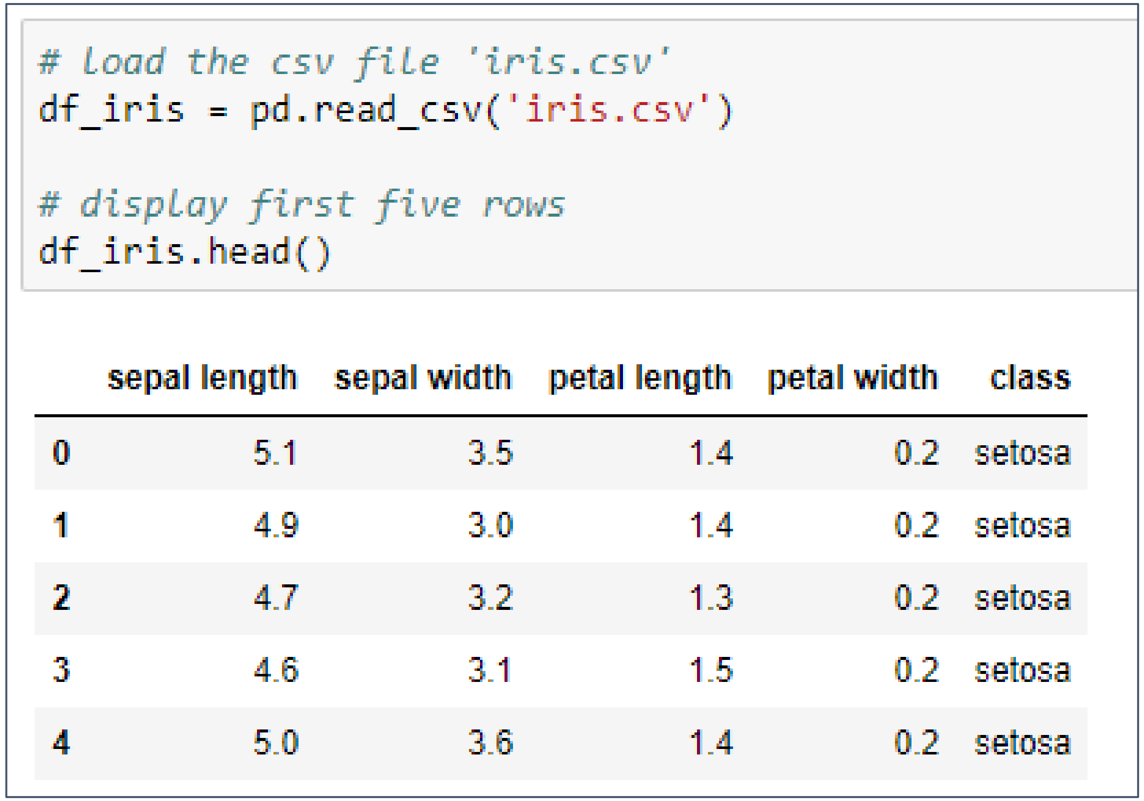


```
# plot a regression plot
# 'x' represents variable on x-axis
# 'y' represents variable on y-axis
# 'data' represents the DataFrame
sns.regplot(x = 'sepal length', y = 'petal length', data = df_iris)

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

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



READ THE PLOT

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

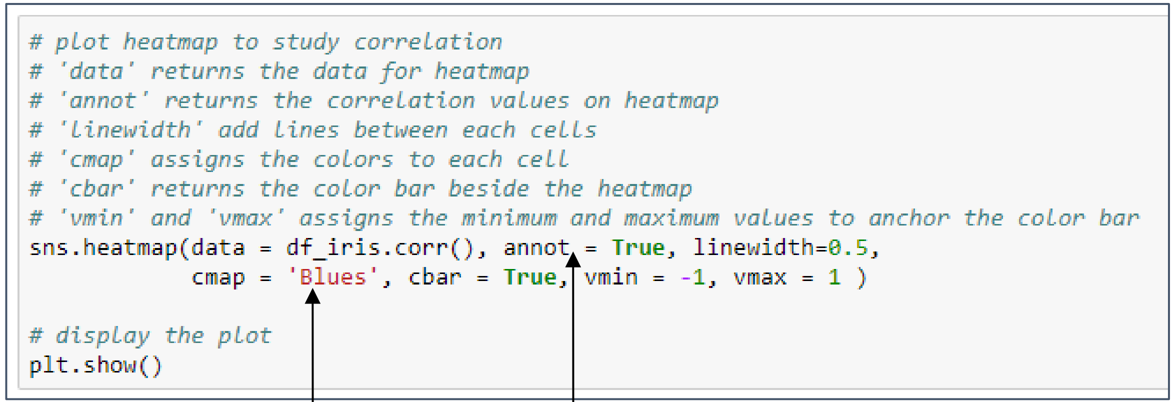
Heatmap

HEATMAP

1 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

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

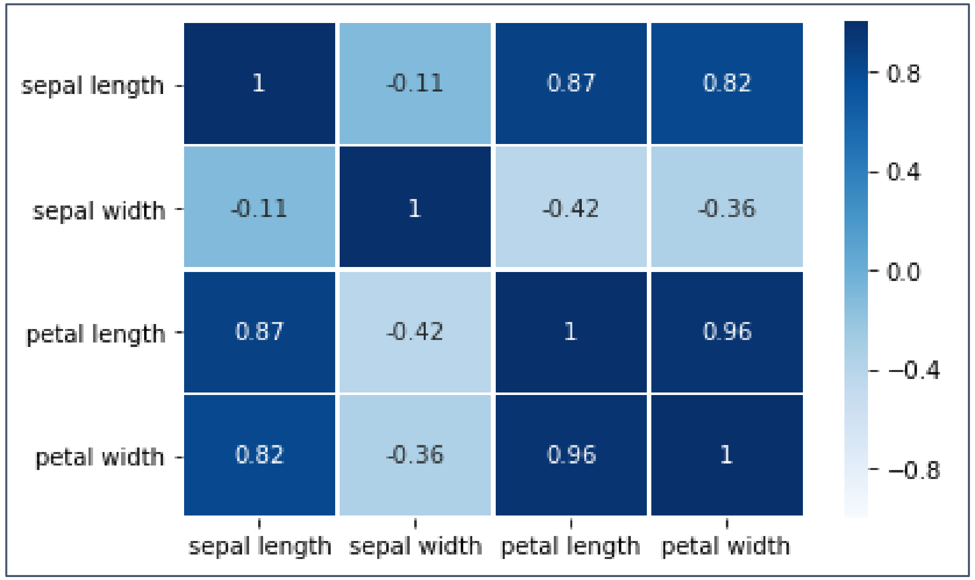
HEATMAP



```
# 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

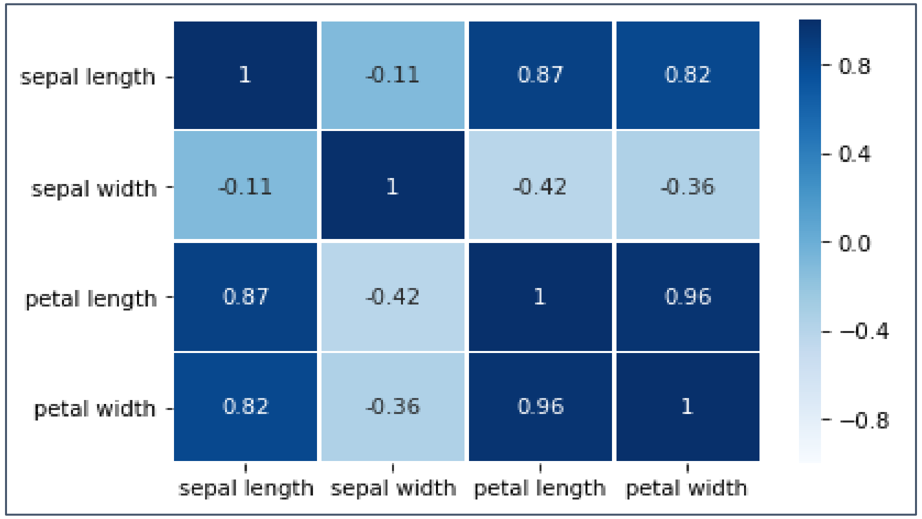


	sepal length	sepal width	petal length	petal width
sepal length	1	-0.11	0.87	0.82
sepal width	-0.11	1	-0.42	-0.36
petal length	0.87	-0.42	1	0.96
petal width	0.82	-0.36	0.96	1

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)



	sepal length	sepal width	petal length	petal width
sepal length	1	-0.11	0.87	0.82
sepal width	-0.11	1	-0.42	-0.36
petal length	0.87	-0.42	1	0.96
petal width	0.82	-0.36	0.96	1

Seaborn is a complement, not a substitute, for Matplotlib. There are some tweaks that still require Matplotlib

We’re committed to empower you to be

#FutureReady

through powerful training solutions.

IMARTICUS LEARNING

We build the workforce of the future.

250+
Corporate Clients

30,000+
Learners Trained

25000+
Learners Placed

FREE WEBINARS

Sign up for free webinars with industry experts every fortnight!

Enroll Now!

Contact us today!
https://imarticus.org/