

# Creating Different plots:- Line Plot, Scatter Plot

## Topics Covered:

- Introduction to Line plot
- Creating and Customization of Line Plot
- Introduction to Scatter plot
- Creating and Customization of Scatter Plot

## Introduction to Line Plot

### What is Line Plot ? Why do we use Line Plot?

Line charts are used to **represent the relation between two data X and Y on a different axis.**

Line plots are **used to display numerical, discrete data only, not the continuous data.** Line plots organize the data by indicating the occurrences of each value on a number line. These graphs are easily constructed with small data sets, and allow for interpretation based on the frequency patterns that are revealed. It is used to represent the variation of continuous data over time. However, it can be used with categorical columns as well

For more information on pandas please refer to the official [Matplotlib Documentation for Line Plot.](#)

## Creating and Customization of Line Plot

We can create Line Plot in python using Matplotlib and the syntax for the same is given below.

### Syntax:-

```
plt.plot(x,y,color,lw,label,linestyle,marker)
```

Where,

- **plt** is the alias name for pyplot
- The **plot()** function takes arguments that describe the layout of the line plot.
- **x** - data(column) to be represented on the x-axis
- **y** - data(column) to be represented on the y-axis
- **color** - Used to specify color of the line
- **lw** - Used to specify width of the line
- **Label** - Used to define the label for the line chart
- **linestyle**- Used to define the style of the edge of the line chart

Linestyle	Description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line

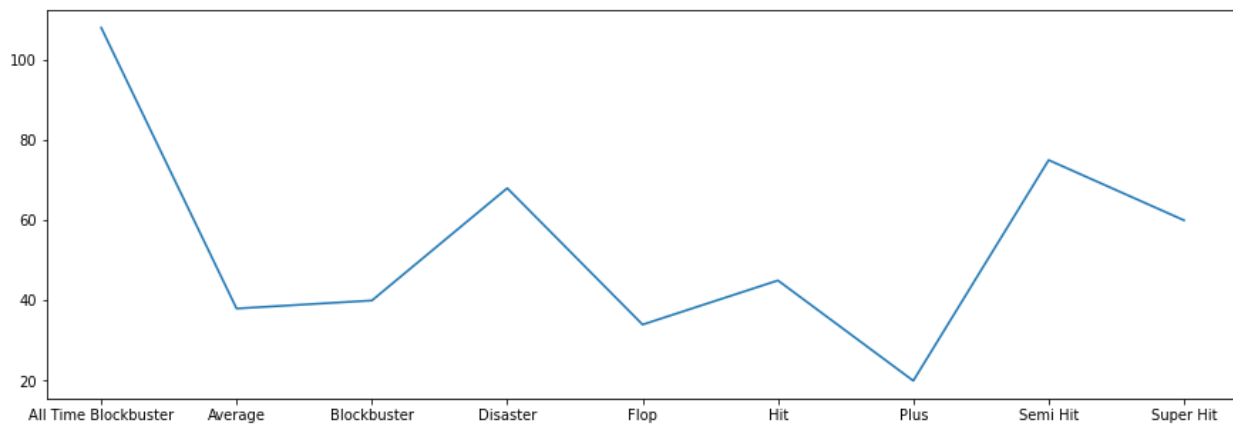
- **marker**- Used to highlight the connection between x and y. To get information about different types of markers refer to the official [matplotlib markers](#) website.
- **markersize**- Determines the size of the marker

## Plotting a normal Line plot

### Method 1 :-

```
# METHOD 1
# creating a simple line plot with figure size (15,5)
plt.figure(figsize=(15,5)) # Setting figure size
plt.plot(df_budget['Box_Office_Verdict'],df_budget['Budget'])
```

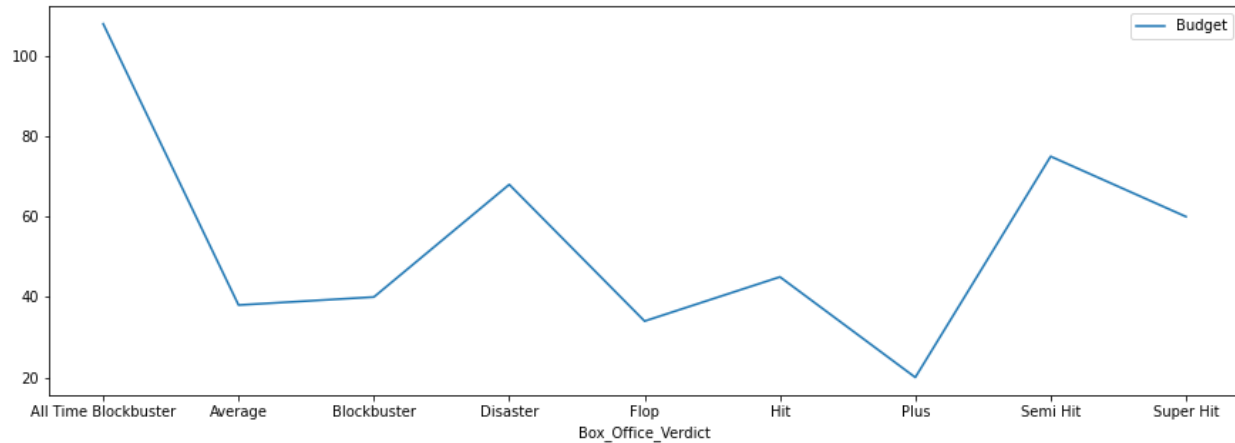
### Output:-



### Method 1 :-

```
# METHOD 2
# creating a simple line plot with figure size (15,5)
lines = df_budget.plot.line(x='Box_Office_Verdict', y='Budget',figsize=(15,5))
# Note:- This method automatically create the legend for the plot
```

## Output:-



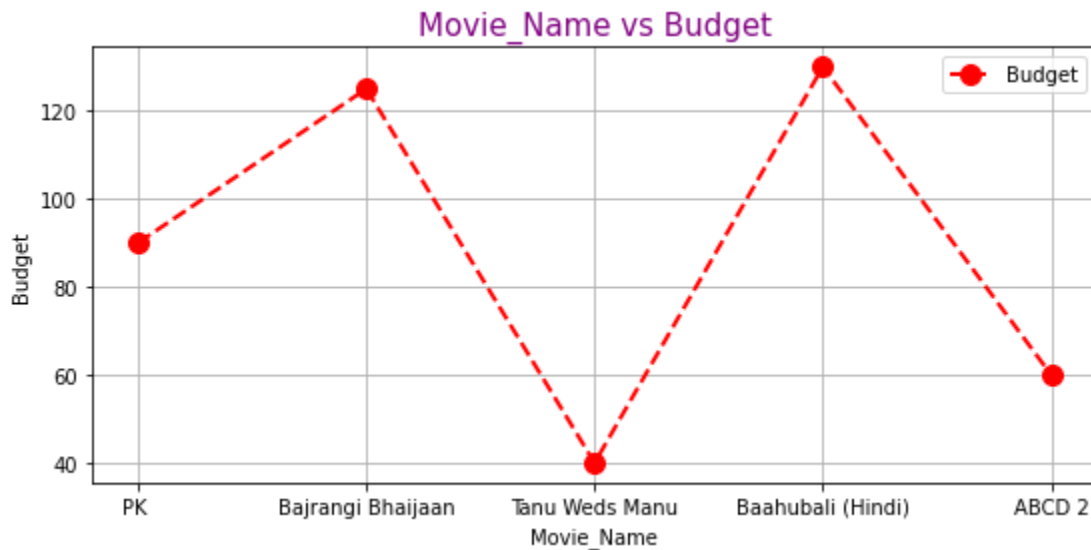
## Customizing plot with markers linewidth and linestyle

We can add markers, customize the marker size, color, type. Customize the line width of the line plot by using the linewidth function. Customize the line type by using the linestyle function.

## Code example:-

```
plt.figure(figsize=(9,4)) # setting plt figure
# Customizing plot by setting line color, marker, label, markersize, linewidth(lw) and
# linestyle
plt.plot(df_top_movies.index,df_top_movies['Budget'],color='red',marker='o',
        label='Budget',markersize=10,lw=2,linestyle='--')
plt.xlabel('Movie_Name') # Setting x-label
plt.ylabel('Budget') # Setting y-label
plt.legend() # Showing Legend for the plot
# Setting title for the plot
plt.title('Movie_Name vs Budget',fontsize=15,color='purple')
plt.grid() # Showing grid for the plot
plt.show() # Showing the plot
```

### Output:-



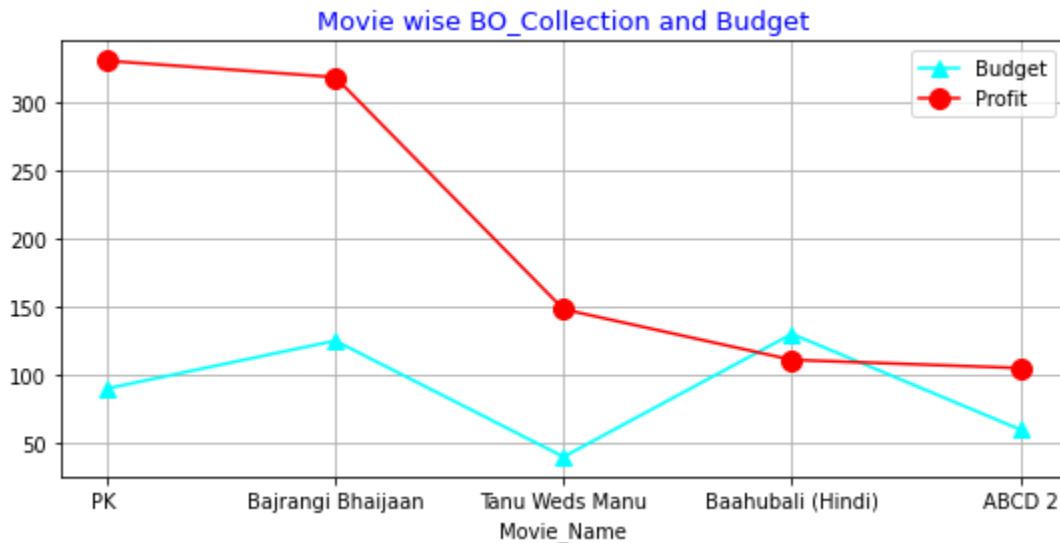
### Plotting multiple Line Plot in a single plot

We can plot two or more variables in the same line plot and also add grid to the plot using `grid()` function. The syntax and example for that is given below.

### Code example:-

```
plt.figure(figsize=(9,4)) # setting plt figure
# Customizing first variable line with color, axis, width and position
plt.plot(df_top_movies.index,df_top_movies['Budget'],color='cyan',marker='^',label='
Budget',markersize=8)
# Customizing second variable line with color, axis, width and position
plt.plot(df_top_movies.index,df_top_movies['BO_Collection'],color='red',marker='o',l
abel='Profit',markersize=10)
# Setting title for the plot
plt.title('Movie wise BO_Collection and Budget',fontsize=13,color='Blue')
plt.xlabel('Movie_Name') # Setting x-label
plt.legend() # Showing Legend for the plot
plt.grid() # Showing grid for the plot
plt.show() # Showing the plot
```

## Output:-



## Introduction to Scatter Plot

### What is a Scatter Plot ? Why do we use Scatter Plot?

Scatter plots are used to **observe relationships between variables** and use dots to represent the relationship between them. The **scatter()** method in the matplotlib library is used to draw a scatter plot. Scatter plots are widely used to represent relations among variables and how change in one affects the other.

Scatter plots are used to plot data points on a horizontal and a vertical axis in the attempt **to show how much one variable is affected by another**. Each row in the data table is represented by a marker whose position depends on its values in the columns set on the X and Y axes.

For more information on pandas please refer to the official [Matplotlib Documentation for Scatter Plot](#).

## Creating and Customization of Scatter Plot

We can create Scatter Plot in python using Matplotlib and the syntax for the same is given below.

### Syntax:-

```
plt.scatter(x,y,color,label,marker,s)
```

Where,

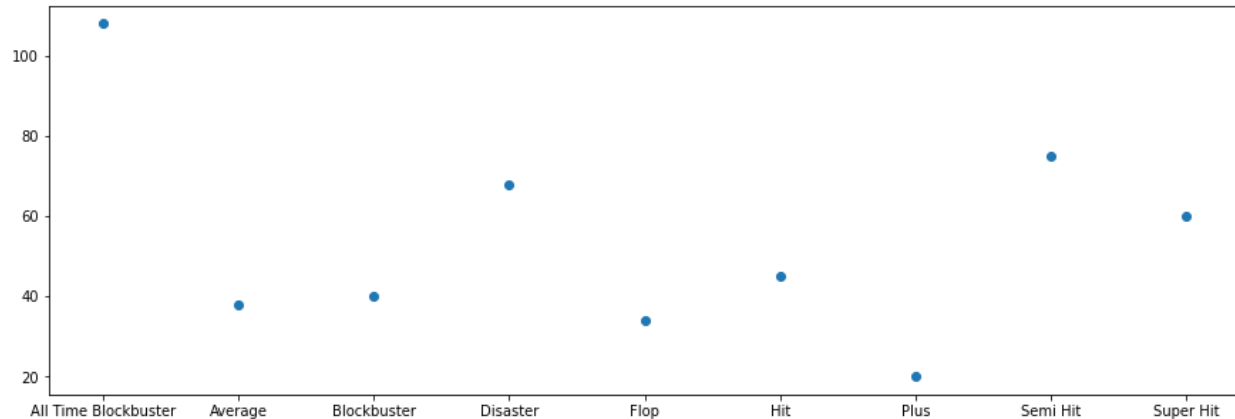
- **plt** is the alias name for pyplot
- The **scatter()** function takes arguments that describe the layout of the scatter plot.
- **x** - data(column) to be represented on the x-axis
- **y** - data(column) to be represented on the y-axis
- **color** - Used to specify color of the marker
- **Label** - Used to define the label for the scatter chart
- **marker**- Used to highlight the connection between x and y. To get information about different types of markers refer to the official [matplotlib markers](#) website.
- **s** - Used to define the size for the markers in scatter chart

### Plotting a normal Scatter plot

#### Code example:-

```
plt.figure(figsize=(15,5)) # setting plt figure
plt.scatter(df_budget['Box_Office_Verdict'],df_budget['Budget'])
```

### Output:-



### Customizing plot with markers and grid

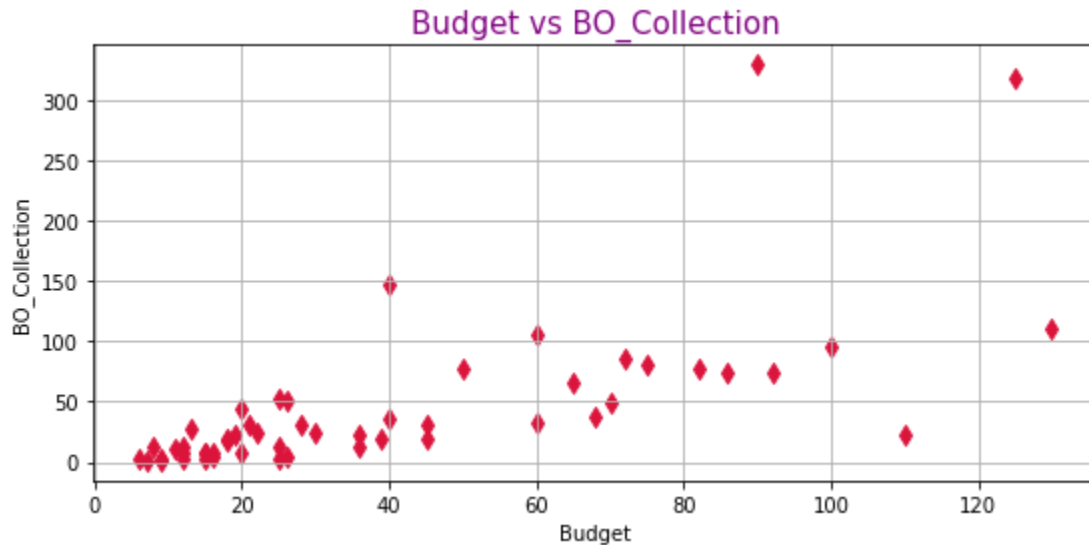
We can add markers, customize the marker size, color, type. and also add grid to the plot using `grid()` function. The syntax and example for that is given below.

### Code example:-

```
plt.figure(figsize=(9,4)) # setting plt figure
# Customizing plot by setting marker type, size and color
plt.scatter(df['Budget'],df['BO_Collection'],marker='d',s=50,color='crimson')
plt.xlabel('Budget') # Setting x-label
plt.ylabel('BO_Collection') # Setting y-label
# Setting title for the plot
plt.title('Budget vs BO_Collection',fontsize=15,color='purple')
plt.grid() # Showing grid for the plot
plt.show() # Showing the plot
```



### Output:-



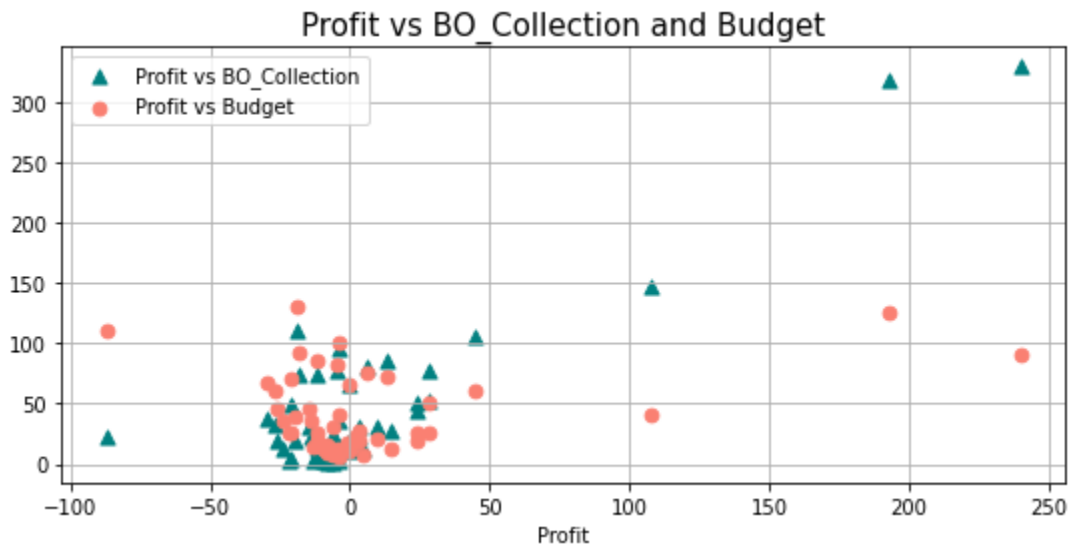
### Plotting two variables in single Plot

We can plot two or more variables in the same scatter plot and the syntax and example for that is given below.

### Code example:-

```
plt.figure(figsize=(9,4)) # setting plt figure
# Customizing first variable markers with type, size, color and label
plt.scatter(df['Profit'],df['BO_Collection'],marker='^',s=50,color='teal',label='Profit vs
BO_Collection')
# Customizing second variable markers with type, size, color and label
plt.scatter(df['Profit'],df['Budget'],marker='o',s=45,color='salmon',label='Profit vs
Budget')
plt.xlabel('Profit') # Setting x-label
# Setting title for the plot
plt.title('Profit vs BO_Collection and Budget',fontsize=15,color='black')
plt.grid() # Showing grid for the plot
plt.legend() # Showing Legend for the plot
plt.show() # Showing the plot
```

### Output:-



## Reference Notebook & Dataset

- [Jupyter Notebook Download Link](#)
- [Bollywood Movies Dataset Download Link](#)