Matplotlib Cheatsheet

*This cheatsheet also contains a section that introduces basic **Pandas** Visualization methods.*

Importing Matplotlib

```python
import matplotlib.pyplot as plt
```

Basic Line plot

```python
days = [0, 1, 2, 3, 4, 5, 6]               # x axis
money_spent = [10, 12, 12, 10, 14, 22, 24] # y axis
plt.plot(days, money_spent)
plt.show()
```

Subplots

```python
plt.subplot(rows, columns, index_of_subplot) # Create subplots

plt.subplot(1, 2, 1) # First Subplot
plt.plot(x, y, color='green') # Plot 1st subplot

plt.subplot(1, 2, 2) # Second Subplot
plt.plot(x, y, color='steelblue') # Plot 2nd Subplot
```

Styles

```python
plt.plot(x,
   y,
  color = 'green','red','blue'... etc # optional
  linestyle=  '.','-','--' or '-.'    # optional
  marker= 'o','*','s','x','d','h'     # optional
  linewidth= 1, 2, ...                # optional
  alpha= 0.1 - 1                      # optional
```

Legends

```python
# Create Legend
plt.legend(["first_line",
      "second_line",
            loc= 1 - 10])
```

Matplotlib Cheatsheet

*This cheatsheet also contains a section that introduces basic **Pandas** Visualization methods.*

Figures

```python
# Create Figure with custom size
plt.figure(figsize=(width, heigth))
plt.plot(x, y)
```

Python

Save Figures

```python
plt.savefig('tall_and_narrow.png/ .svg/ .pdf')
```

Python

Modify Ticks

```python
ax = plt.subplot(row, column, index) # Specify subplot to modify

ax.set_xticks([1, 2, 4]) # x Attributes
ax.set_yticks([0.1, 0.2, ...]) # y Attributes

ax.set_xticklabels(["Jan", "Feb", "Apr"],
     rotation=30) # rotation=degrees rotates the labels

ax.set_yticklabels(["10%", "20%", ...]) # labeling y axis
```

Python

Axis and Labels

```python
plt.axis(x_min, x_max, y_min, y_max) # Zoom in or out of the plot

plt.xlabel("This is a label ") # Labeling the x-Axis
plt.ylabel("This is a label" ) # Labeling the y-Axis
plt.title("This is a title") # Labeling the Plot
```

Python

Add Text to Graph

```python
plt.text(x_coord, y_coord, "text");
```

Python

Matplotlib Cheatsheet

*This cheatsheet also contains a section that introduces basic **Pandas** Visualization methods.*

Simple Bar Chart

```python
plt.bar(range(len(y_values)), y_values)
```

Scatter Plot

```python
plt.scatter(x_values, y_values)
```

Histogram

```python
plt.hist(dataset, range=(0,100), bins=20) # Create one Histogram
# Specifiy number of bins (default = 10)

# Create multiple Histograms
plt.hist(a, alpha=0.5, normed=True)
plt.hist(b, histtype='step', linewidth=2 normed=True)
```

Pandas- Visualization

Terms Used

df = Pandas DataFrame
series = Pandas Series
data = Pandas DataFrame or Series

Plotting a Dataframe

```python
df.plot() # Plot all the columns
    #in the dataFrame with Index as x-axis
```

Plotting a Series

```python
series.plot()
```

Matplotlib Cheatsheet

*This cheatsheet also contains a section that introduces basic **Pandas** Visualization methods.*

Plotting selected columns

```python
df.plot('Name of X column',
  'Name of Y column',
  kind = 'bar','scatter','hist','box','kde','area'...)
# The default plot is always a line plot unless kind is stated
```

Bar Plots

```python
data.plot.bar()  # Normal bar plot
data.plot.barh() # Horizontal plot

data.plot.bar(stacked=True) # Stacked bar plot
```

Area Plots

```python
data.plot.area() # Area plot
data.plot.area(stacked=False) # Non-Stacked area plot
```

Pie plots

```python
series.plot.pie() # Pie plot for Series

DataFrame.plot.pie(subplots=True) # Pie plot for DataFrame

series.plot.pie(labels= ['A','B','C'], colors= ['r','b','g'], autopct= '%.2f')
```

Scatter plot

```python
DataFrame.plot.scatter(x=' ', y=' ') Scatter plot

ax= df.plot.scatter(x='A',y='B',
        color='None',
        label='Group1')

df.plot.scatter(x='C',y='D',
    color='Other',
    label='Group2', ax= ax)# Multiple plots
```

Matplotlib Cheatsheet

*This cheatsheet also contains a section that introduces basic **Pandas** Visualization methods.*

Histograms

```python
data.plot.hist() #Histogram plot

data.plot.hist(stacked=True, bins=10) # Stacked and bins size

data.plot.hist(orientation='horizontal',
  cumulative=True)

data.diff().hist(color='g', alpha=0.5) # Subplots histograms
```