

Data Plotting Using matplotlib.pyplot as plt

Using plot container (Single Plot):

First-Way(Empty Plot):

```
import matplotlib.pyplot as plt
plt.plot()
plt.show()
```

Second-Way (Multiple Values, and adding xticks, Title, xlabel, ylabel):

```
plt.plot(x_values, y_values)
plt.xticks(rotation=90)
plt.xlabel('Month')
plt.ylabel('Unemployment Rate')
plt.title('Monthly Unemployment Trends, 1948')
```

Third-Way (With figsize, color, for loop, label and legend) (Single Plot):

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(12,12))

for i in range(5):
    start_index = i*12                # For starting index of the columns to be plotted
    end_index = (i+1)*12              # For starting index of the columns to be plotted
    subset = unrte[start_index:end_index] # Here unrte is a Dataframe and
    plt.plot(subset['DATE'], subset['VALUE'], c=colors[i]) # DATE & VALUE are it's columns
    label = str(1948 + i)              # Adds label for line in the plot

plt.legend(loc='upper left')          # At this locations
plt.show()
```

Using axes container (Multiple Plots):

First-Way:

```
import matplotlib.pyplot as plt
fig = plt.figure()
x_values = [0.0, 0.5, 1.0]
y_values = [10, 20, 40]

ax1 = fig.add_subplot(2, 1, 1)
ax1.plot(x_values, y_values)

ax2 = fig.add_subplot(2, 1, 2)
```

```
ax2.plot([0], [0])
plt.show()
```

Second-Way (With figsize, color and using for loop) (Multiple Plots):

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(12,12))
colors = ['red', 'blue', 'green', 'orange', 'black']

for i in range(5):
    ax = fig.add_subplot(5,1,i+1)          # For positioning Axes Objects
    start_index = i*12                    # For starting index of the columns to be plotted
    end_index = (i+1)*12                  # For starting index of the columns to be plotted
    subset = unrate[start_index:end_index] # Here unrate is a Dataframe and
    ax.plot(subset['DATE'], subset['VALUE'], c=colors[i]) # DATE & VALUE are it's columns

plt.show()
```

Adding More Options (Labels, Legend, Title, xlabel, ylabel) (Multiple Plots):

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(12,12))
colors = ['red', 'blue', 'green', 'orange', 'black']

for i in range(5):
    ax = fig.add_subplot(5,1,i+1)          # For positioning Axes Objects
    start_index = i*12                    # For starting index of the columns to be plotted
    end_index = (i+1)*12                  # For starting index of the columns to be plotted
    subset = unrate[start_index:end_index] # Here unrate is a Dataframe and
    ax.plot(subset['DATE'], subset['VALUE'], c=colors[i]) # DATE & VALUE are it's columns
    ax.legend([str(1948 + i)], loc='upper left')
    ax.set_xlabel('Year - ' + unrate.loc[i*12, 'DATE'].strftime("%Y")) # Choose either xlabel or title
    ax.set_ylabel("Unemployment Rate, Percent")
    ax.set_title("Monthly Unemployment Trends, " + unrate.loc[i*12, 'DATE'].strftime("%Y") )

plt.show()
```

Data Plotting for BarPlots, ScatterPlots, Histograms and BoxPlots

BarPlots:

First-Way:

```
fig, ax = plt.subplots()           # Quick way of creating a plot with single axes object

from numpy import arange           # arange gives array of numbers in a sequence and given range
bar_positions = arange(5) + 0.75   # left starting positions
num_cols = ['RT_user_norm', 'Metacritic_user_nom', 'IMDB_norm', 'Fandango_Ratingvalue',
            'Fandango_Stars']      # xticklabels

bar_heights = norm_reviews[num_cols].iloc[0].values    # height

ax.bar(bar_positions, bar_heights)
# ax.bar(left position, height, width)
# Default width is 0.8
plt.show()
```

Adding More Options (xlabel, ylabel, xticks, xticklabels, title):

```
fig, ax = plt.subplots()

bar_position = arange(5) + 0.75
bar_height = norm_reviews[num_cols].iloc[0].values

ax.bar(bar_position, bar_height, 0.5)

tick_positions = range(1, 6)
ax.set_xticks(tick_positions)           # Setting tick positions at x-axis

ax.set_xticklabels(num_cols, rotation=90) # Giving names/labels to the x-axis ticks

ax.set_xlabel("Rating Source")           # X-axis label
ax.set_ylabel("Average Rating")          # Y-axis label
ax.set_title("Average User Rating For Avengers: Age of Ultron (2015)") # Title of the axes Object
plt.show()

-----
-
# Incase of horizontal Bar Plot:
# fig, ax = plt.subplots()
```

```
# bar_position = arange(5) + 0.75
# bar_height = norm_reviews[num_cols].iloc[0].values

# ax.barh(bar_position, bar_height, 0.5, align="edge")

# ax.set_yticklabels(num_cols)
# ax.set_yticks(range(1, 6))

# plt.ylabel("Rating Source")
# plt.xlabel("Average Rating")

# plt.title("Average User Rating For Avengers: Age of Ultron (2015)")
# plt.show()
```

ScatterPlots:

First-Way:

```
fig, ax = plt.subplots()

ax.scatter(norm_reviews['Fandango_Ratingvalue'],
           norm_reviews['RT_user_norm'])

ax.set_xlabel("Fandango")
ax.set_ylabel("Rotten Tomatoes")
ax.set_xlim(0, 5)
ax.set_ylim(0, 5)
ax.set_title('f vs. r') # Use Either xlabel or title in case of multiple plots

plt.title("Fandango VS Rotten Tomatoes") # Title to be set at Either axes level or plot level
plt.show()
```

Histograms:

First-Way:

```
fig, ax = plt.subplots()

ax.hist(norm_reviews['Fandango_Ratingvalue'], bins=20, range=(0,5))
ax.set_ylim(0,50)
plt.title("Fandango_Ratingvalue Histogram")
plt.show()
```

BoxPlots:

First-Way:

```
fig, ax = plt.subplots()

ax.boxplot(norm_reviews['RT_user_norm'])
ax.set_ylim(0, 5)
ax.set_xticklabels(['Rotten Tomatoes'])
plt.show()
```

Second-Way(Multiple Plots):

```
fig, ax = plt.subplots()

num_cols = ['RT_user_norm', 'Metacritic_user_nom', 'IMDB_norm', 'Fandango_Ratingvalue',
            'Fandango_Stars']
ax.boxplot(norm_reviews[num_cols].values)
ax.set_ylim(0, 5)
ax.set_xticklabels(num_cols, rotation=90)
plt.show()
```

Data Plotting Using Pandas (Additional to others: Scatter Matrix)

First-Way

```
df.plot(x='Sample_size', y='Employed', kind='scatter')  
  
# More Options  
df.plot(x='Sample_size', y='Employed', kind='scatter', title='Employed vs. Sample_size',  
figsize=(5,10))
```

Second-Way (Using Axes Object Container)

```
ax = df.plot(x='Sample_size', y='Employed', kind='scatter')  
ax.set_title('Employed vs. Sample_size')
```

For Histograms

```
df['Sample_size'].plot(kind='hist')  
  
# With Bins:  
df['Sample_size'].hist(bins=25, range=(0,5000))
```

For Scatter Matrix:

```
scatter_matrix(df[['Women', 'Men']], figsize=(10,10))
```

For Bars:

```
df[:5]['Women'].plot(kind='bar')  
  
# If X-Labels needs to be specified or else index is used  
df[:5].plot.bar(x='Major', y='Women')
```