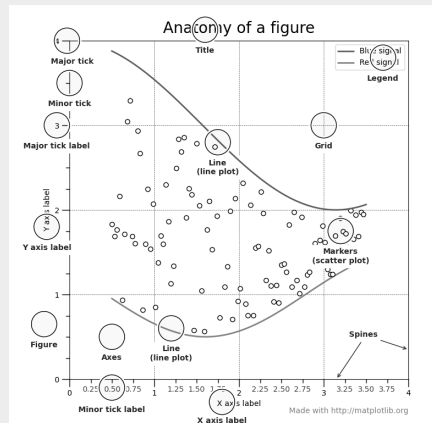


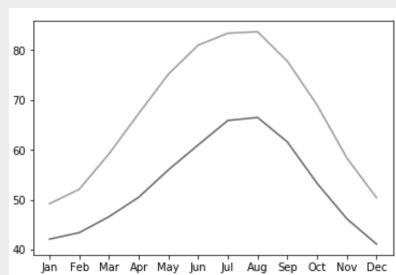
Import

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

Anatomy of a figure



Main principle



Example

```
fig, ax = plt.subplots() # Create one subplot
ax.plot(x1, y1) # each ax represents one plot on the figure object
ax.plot(x2, y2) # another plot is added to the figure object (with another color)
plt.show()
```

Customizing data appearance

<code>ax.plot(x,y, marker = "")</code>	# indicate each data point on the line	https://matplotlib.org/3.2.0/gallery/lines_bars_and_markers/marker_reference.html#sphx-glr-gallery-lines-bars-and-markers-marker-reference-py
<code>ax.plot(x,y, linestyle = "")</code>	# change linestyle	https://matplotlib.org/3.2.0/gallery/lines_bars_and_markers/linestyles.html#sphx-glr-gallery-lines-bars-and-markers-linestyles-py
<code>ax.plot(x,y, color = "")</code>	# change color	https://matplotlib.org/3.2.0/gallery/color/named_colors.html#sphx-glr-gallery-color-named-colors-py
<code>ax.set_xlabel(" ")</code>	# name the x axis	
<code>ax.set_ylabel(" ")</code>	# name the y axis	

Customizing data appearance (cont)

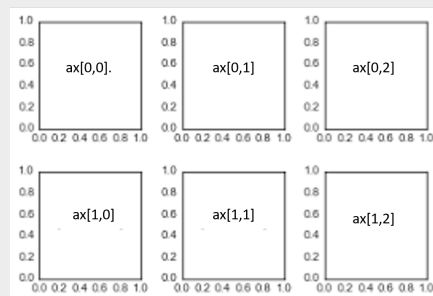
<code>ax.set_title("")</code>	# give figure a title	
<code>ax.tick_params("x"/"y", colors = "")</code>	# change the color of the ticks	
<code>ax.set_xticks(list())</code>	# change the tick values of the X axis	Use list comprehension
<code>ax.set_yticks(list())</code>	# change the tick values of the y axis	Use list comprehension

scatterplot

```
fig, ax = plt.scatter(x, y, s=area, c=colors, alpha=0.5)
ax.plot(x, y, linestyle = "None")
plt.show()
```

Subplot

```
fig, axes = plt.subplots(3,2)
```



Create the subplots

```
fig, ax = plt.subplots(3, 2)
```

First Subplot

```
ax[0, 0].plot(x, y, color = "green")
```

Second Subplot

```
ax[0, 1].plot(x, y, color = "blue")
```

Subplot

<code>fig, ax = plt.subplots(rows, columns, index_of_subplot)</code>	fig = container holding one to more axes; ax = individual plot
<code>ax.plot([x], y, color = "green")</code>	[X] = X coordinates; Y = Y coordinates
<code>plt.show()</code>	Plot the figure.
<code>plt.subplots(rows, columns, sharey = True)</code>	All subplots have same range of Y-axis values

Time series

Once there is column with a DateTime datatype then matplotlib will recognize that this is a variable that represents time.



By [deleted]

cheatography.com/deleted-121124/

Not published yet.

Last updated 21st March, 2020.

Page 2 of 4.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Time series

Plot time series	<code>ax.plot(time_variable, variableOfInterest)</code>
slice time variable	<code>timeVariable["start-date":"end-date"]</code>

Time-series with different variables

```
fig, ax = plt.subplots()
ax.plot(time_variable, variable1, color = "blue")
ax.set_xlabel()
ax.set_ylabel()
ax2 = ax.twinx() # same x-axis but different y-axis
ax2.plot(time_variable, variable2, color = "red")
ax2.set_xlabel()
ax2.set_ylabel()
plt.show()
```

Add annotations

```
# First argument is the text of the annotation
# xy argument is the xy of the dataset to which the annotation has to refer (pandas object)
# xytext argument is the xy coordinate of the text
# arrowprops a dictionary that defines the properties of the arrow we would like to use
ax.annotate("", xy = , xytext = , arrowprops = {})
# the arguments for the arrowprops
https://matplotlib.org/tutorials/text/annotations.html
```

Annotations are small pieces of text referring to a particular part of the graph.

Bar chart

```
fig, ax = plt.subplots()
ax.bar(x-axis, y-axis)
plt.show()
```

barchart

<code>ax.set_xticklabels(names)</code>	change the names on the x axis
<code>ax.set_ylabel("")</code>	Change name of the y axis
<code>ax.bar(x,y, bottom = , label = "")</code>	Important that you use the same x values, set bottom= equal to the pd.series containing the data whereon you want to stack.
<code>ax.legend()</code>	
<code>ax.set_xticklabels(x, rotation = 90)</code>	If you want to turn the x labels 90 degrees.

You can keep adding bars to eachother by calling `ax.bar(bottom)` multiple times. However, you have to add the different series in bottom: `"medals["Gold"] + medals["Silver"]"`



Histogram

```
fig, ax = plt.subplots()
ax.hist(x, label = "", bins = value/list, histtype = , label = "")
ax.set_xlabel("")
ax.set_ylabel("")
```

Boxplot

```
fig, ax = plt.subplots()
ax.boxplot(x)
ax.set_ylabel()
# Create multiple boxplots
fig, ax = plt.subplots()
ax.boxplot([mens_rowing["Height"], mens_gymnastics["Height"]])
ax.set_xticklabels(["Rowing", "Gymnastics"])
```

Scatterplot

```
fig, ax = plt.subplots()
ax.scatter(x, y, color = " " , label = "")
# create mutiple scatters on the same plot with different colors
fig, ax = plt.subplots()
ax.scatter(x, y, color = "red", label = "")
ax.scatter(x, y, color = "blue", label = "")
ax.legend()
ax.set_xlabel()
ax.set_ylabel()
# encoding a third variable by color
fig, ax = plt.subplots()
ax.scatter(x, y, c = )
```

Changing plot style

```
# This has to be before the subplots method
plt.style.use("ggplot")
# The available styles
https://matplotlib.org/gallery/style\_sheets/style\_sheets\_reference.html
```

Sharing your visualizations

```
# save figure
fig.savefig("name.png")
```

