

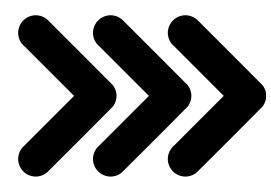


**Tajamul Khan**

# Matplotlib cheat sheet



**@Tajamulkhan**



# Basic Plots

- **plt.plot(x, y):** Line plot for trends and continuous data.
- **plt.scatter(x, y):** Visualize relationships between two variables using scatter plots.
- **plt.bar(x, height):** Bar chart to compare categorical data.
- **plt.barh(y, width):** Horizontal bar chart for easier comparison.
- **plt.hist(data, bins=10):** Histogram to analyze data distribution.
- **plt.pie(sizes, labels=labels):** Show proportions of categories using a pie chart.
- **plt.boxplot(data):** Summarize distributions and detect outliers.
- **plt.violinplot(data):** Show density and distribution of data.



@Tajamulkhan



# Customizing Plots

- **plt.title('Title')**: Set the title of the plot.
- **plt.xlabel('X Label')**: Set the label for the x-axis.
- **plt.ylabel('Y Label')**: Set the label for the y-axis.
- **plt.legend()**: Display the legend for the plot.
- **plt.grid(True)**: Enable gridlines on the plot.
- **plt.xlim([xmin, xmax])**: Set the x-axis limits.
- **plt.ylim([ymin, ymax])**: Set the y-axis limits.
- **plt.xticks(rotation=45)**: Rotate the x-axis labels for better readability.



@Tajamulkhan



# Subplots & Layout

- **plt.subplot(nrows, ncols, index):** Create a subplot in a grid.
- **plt.subplots\_adjust(left, right, top, bottom):** Adjust subplot spacing.
- **fig, ax = plt.subplots():** Create a figure and axes object for finer control.
- **fig.tight\_layout():** Automatically adjust subplot parameters to fit the figure area.
- **plt.subplot(121):** Create a subplot in a 1x2 grid (first position).
- **plt.subplots(2, 2):** Create a 2x2 grid of subplots.
- **ax.plot(x, y):** Use ax (axes) to plot directly into a specific subplot.
- **ax.set\_title('Subplot Title'): Set a title for a specific subplot.**



@Tajamulkhan



# Styles & Color

- **plt.style.use('ggplot')**: Apply a specific style to the plot (e.g., 'ggplot', 'seaborn').
- **plt.plot(x, y, 'ro')**: Customize the plot style and markers (red circles).
- **plt.plot(x, y, linestyle='--', color='g')**: Set the line style and color (green dashed line).
- **plt.scatter(x, y, c='blue', marker='x')**: Set color & marker for scatter plot.
- **plt.bar(x, height, color='skyblue')**: Customize bar chart colors.
- **plt.hist(data, bins=9, color='purple')**: Set color for histogram bars.
- **plt.plot(x, y, alpha=0.5)**: Adjust transparency of plot elements.
- **plt.pie(sizes, colors=['red', 'green', 'blue'])**: Set specific colors for pie chart sections.



@Tajamulkhan



# Annotations & Text

- **plt.annotate('Text', xy=(x, y), xytext=(xtext, ytext), arrowprops=dict(arrowstyle="->")):** Annotate the plot with text and optional arrows.
- **plt.text(x, y, 'Text'):** Place text at specified coordinates on the plot.
- **plt.title('Title', fontsize=14):** Set title with customized font size.
- **plt.xlabel('X Label', fontsize=12):** Set x-axis label with font size.
- **plt.ylabel('Y Label', fontsize=12):** Set y-axis label with font size.
- **plt.xticks([0, 1, 2], ['A', 'B', 'C']):** Change x-tick labels to custom values.
- **plt.yticks([10, 30], ['Low', 'High']):** Customize y-tick labels.
- **plt.text(2, 3, 'Annotation'):** Add simple annotation text at specific coordinates.



@Tajamulkhan



# Saving & Exporting

- **plt.savefig('plot.png')**: Save the current plot as a PNG file.
- **plt.savefig('plot.pdf', dpi=300)**: Save plot as a PDF with specified resolution.
- **plt.savefig('plot.svg')**: Save plot as an SVG file for scalable graphics.
- **plt.savefig('plot.png', transparent=True)**: Save plot with transparent background.
- **fig.savefig('figure.png')**: Save a specific figure object to a file.
- **plt.savefig('plot.png', bbox\_inches='tight')**: Save plot with tight bounding box (no extra space).
- **plt.savefig('plot.png', pad\_inches=1)**: Control padding around saved figure.
- **plt.savefig('plot.png', dpi=300, quality=95)**: Save with custom quality and resolution.



@Tajamulkhan



# 3D Plotting

- **from mpl\_toolkits.mplot3d import Axes3D:** Import 3D plotting toolkit.
- **fig = plt.figure():** Create a figure object for 3D plotting.
- **ax = fig.add\_subplot(111, projection='3d'):** Create 3D axes.
- **ax.plot3D(x, y, z):** Plot a 3D line graph.
- **ax.scatter3D(x, y, z):** Create a 3D scatter plot.
- **ax.plot\_surface(x, y, z, cmap='viridis'):** Create a 3D surface plot with color map.
- **ax.contour3D(x, y, z, 50):** Create 3D contour plot.
- **ax.bar3d(x, y, z, dx, dy, dz):** Create a 3D bar plot.



@Tajamulkhan



# Multiple Plots

- **plt.plot(x, y)**: Create a basic plot.
- **plt.subplot(1, 2, 1)**: Create subplots in a single row, first plot.
- **plt.subplot(1, 2, 2)**: Create subplots in a single row, second plot.
- **plt.subplots(2, 2)**: Create a 2x2 grid of subplots.
- **ax1.plot(x, y), ax2.plot(x, z)**: Plot on different axes in subplots.
- **fig.tight\_layout()**: Adjust layout of multiple subplots automatically.
- **plt.subplots\_adjust(left, right, top, bottom)**: Customize the spacing between subplots.
- **plt.subplot(111)**: Create a single full-sized plot.



@Tajamulkhan



# Found Helpful?

---

# Repost



**Follow for more!**