Cheat sheets

# Matplotlib Cheat sheet

|  |  |
| --- | --- |
| Create a figure with a single subplot | **fig, ax = plt.subplots()** |
| Create a figure with multiple subplots | **fig, axs = plt.subplots(nrows=2, ncols=2)** |
| Plot a line graph | **ax.plot(x\_data, y\_data)** |
| Plot a scatter plot | **ax.scatter(x\_data, y\_data)** |
| Add labels to the x and y axes. | **ax.set\_xlabel('X Label')**  **ax.set\_ylabel('Y Label')** |
| Add a title to the plot | **ax.set\_title('Title')** |
| Add a legend to the plot | **ax.legend()** |
| Change the line colour | **ax.plot(x\_data, y\_data, color='red')** |
| Change the line style | **ax.plot(x\_data, y\_data, linestyle='--')** |
| Change the marker shape | **ax.scatter(x\_data, y\_data, marker='o')** |
| Set the x and y-axis limits. | **ax.set\_xlim([0, 10])**  **ax.set\_ylim([0, 20])** |
| Add text to the plot | **ax.text(x, y, 'Text')** |
| Save the plot to a file | **plt.savefig('filename.png')** |
| Show the plot | **plt.show()** |
| # Create a bar chart | **ax.bar(x\_data, y\_data)** |
| Add a color map to a plot | **sc = ax.scatter(x\_data, y\_data, c=z\_data, cmap='viridis')**  **plt.colorbar(sc)** |
| Add error bars to a plot | **ax.errorbar(x\_data, y\_data, yerr=error\_data)** |
| Add a grid to a plot | **ax.grid(True)** |
| Change the font size of x and y axis labels | **ax.set\_xlabel('X Label', fontsize=14)**  **ax.set\_ylabel('Y Label', fontsize=14)** |
| Change the font size of the title | **ax.set\_title('Title', fontsize=16)** |
| Create a horizontal bar chart | **ax.barh(x\_data, y\_data)** |
| Create a histogram | **ax.hist(data, bins=10)** |
| Customize the x-tick labels | **ax.set\_xticks([1, 2, 3, 4, 5])**  **ax.set\_xticklabels(['One', 'Two', 'Three', 'Four', 'Five'])** |
| Add a horizontal or vertical line to a plot | **ax.axhline(y\_value, color='gray', linestyle='--')**  **ax.axvline(x\_value, color='gray', linestyle='--')** |
| Add annotations to a plot | **ax.annotate('Annotation', xy=(x, y), xytext=(x\_text, y\_text), arrowprops=dict(facecolor='black', arrowstyle='->'))** |
| Create a subplot with shared x-axis or y-axis | **fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True) # shared x-axis**  **fig, (ax1, ax2) = plt.subplots(1, 2, sharey=True) # shared y-axis** |
| Create a 3D plot | **from mpl\_toolkits import mplot3d**  **ax = plt.axes(projection='3d')**  **ax.plot3D(x\_data, y\_data, z\_data)** |
| Create a polar plot | **ax = plt.subplot(111, projection='polar')**  **ax.plot(theta, r)** |
| Create a violin plot | **ax.violinplot(data)** |
| Create a box plot | **ax.boxplot(data)** |
| Create a heat map | **ax.imshow(data, cmap='viridis')**  **plt.colorbar()** |
| Create a pie chart | **ax.pie(data, labels=label\_data)** |
| Fill the area between two lines | **ax.fill\_between(x\_data, y1\_data, y2\_data, alpha=0.2)** |
| Add a background grid to a plot | **ax.grid(color='gray', linestyle='--', linewidth=0.5)** |
| Rotate x-axis tick labels | **plt.xticks(rotation=45)** |
| Add a horizontal or vertical shaded region to a plot | **ax.axhspan(ymin, ymax, alpha=0.2, color='gray')**  **ax.axvspan(xmin, xmax, alpha=0.2, color='gray')** |
| Add a vertical text label to a plot | **ax.text(x, y, 'Label', rotation=90, verticalalignment='center')** |
| Change the font family of text | **plt.rcParams['font.family'] = 'Arial'** |
| Add a legend outside of the plot | **ax.legend(bbox\_to\_anchor=(1.05, 1), loc='upper left')** |
| Add a dashed or dotted line to a plot | **ax.plot(x\_data, y\_data, linestyle='--') # dashed**  **ax.plot(x\_data, y\_data, linestyle=':') # dotted** |
| Add a secondary y-axis to a plot | **ax2 = ax.twinx()**  **ax2.plot(x\_data, y2\_data, color='red')** |
| Set the font size of tick labels | **ax.tick\_params(labelsize=12)** |
| Remove the top and right spines of a plot | **ax.spines['top'].set\_visible(False)**  **ax.spines['right'].set\_visible(False)** |
| Change the background color of a plot  Create a logarithmic scale on the x or y axis | **ax.set\_facecolor('lightgray')**  **ax.set\_xscale('log')**  **ax.set\_yscale('log')** |
| Set the figure size | **fig.set\_size\_inches(8, 6)** |
| Save the plot with a transparent background | **plt.savefig('filename.png', dpi=300, bbox\_inches='tight', transparent=True)** |
|  |  |
|  |  |