Table of Contents

# 5.3 Customizing Axes, Figure Size & Aspect Ratio

## Axis labels and title

Use plt.xlabel() and plt.ylabel() to add axis labels, and plt.title() for a chart title. You can specify font size, colour and rotation.

## Axis limits

Set axis ranges using plt.xlim(min, max) and plt.ylim(min, max). Adjusting limits focuses on relevant regions and can improve interpretability.

## Figure size

Control the overall size of the figure using plt.figure(figsize=(width, height)), where width and height are measured in inches. Larger figures prevent clutter when plotting multiple subplots.

## Aspect ratio

The aspect ratio (width‑to‑height ratio) influences how your plot looks. You can create square plots or elongate them for better readability. Use the aspect parameter in axes or adjust the figure size accordingly.

## Example

Create a scatter plot of height versus weight with:

* x‑axis labelled “Height (cm)” and y‑axis labelled “Weight (kg)”,
* Title “Height vs Weight”,
* Axis limits set to focus on the range 150–200 cm and 50–100 kg, and
* Figure size of 8×6 inches.

## Summary

Customising axes and figure layout improves the readability and aesthetics of plots. Properly labelled axes and appropriately sized figures help viewers interpret results【98554664303776†L151-L182】.

## Reflection questions

1. Why is it important to label axes and title your charts?
2. How can adjusting axis limits change the perception of the data?
3. Describe how to create a square plot in Matplotlib.

## References