Table of Contents

# 5.4 Exporting & Saving Visualisations

## Saving plots

Matplotlib can save plots to various formats using plt.savefig(filename, dpi=resolution) before plt.show(). Supported formats include PNG, PDF, SVG and JPEG.

* **PNG** – Lossless image format suitable for presentations.
* **PDF** – High‑quality vector format for reports and printing.
* **SVG** – Vector graphics ideal for web and publications.

Set dpi (dots per inch) to control image resolution (for example, 300 dpi for print).

## Workflow

1. Create and customise the plot as usual.
2. Call plt.savefig('figure.png', dpi=300, bbox\_inches='tight') to save the figure. The bbox\_inches parameter trims extra whitespace.
3. Optionally call plt.show() to display the plot in an interactive window.

## Example

Generate a histogram of exam scores and save it as a PDF:

import matplotlib.pyplot as plt  
plt.hist(scores, bins=10)  
plt.xlabel('Score')  
plt.ylabel('Frequency')  
plt.title('Distribution of Exam Scores')  
plt.savefig('exam\_scores.pdf', dpi=300, bbox\_inches='tight')

## Summary

Saving visualisations ensures that analyses are reproducible and shareable. Choosing appropriate file formats and resolutions is essential for professional reporting【98554664303776†L151-L182】.

## Reflection questions

1. When might you prefer saving a figure as SVG rather than PNG?
2. Why should plt.savefig() be called before plt.show() in scripts?
3. What does the dpi parameter control?

## References