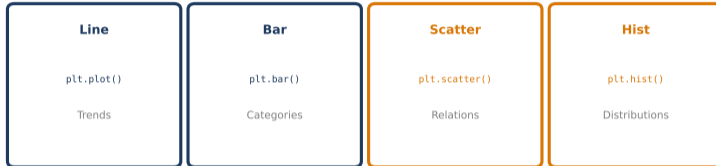


## Lesson 17 Summary: Matplotlib Basics

Data Science with Python – Key Concepts

Data Science Program

## Matplotlib Visualization



### Customization:

`figure()` | `subplots()` | `savefig()` | `rcParams`

*Object-oriented API for better control*

---

Matplotlib is Python's foundational visualization library

## Core visualization methods:

- **Line:** `plt.plot(x, y)` – trends over time
- **Bar:** `plt.bar(x, heights)` – category comparison
- **Scatter:** `plt.scatter(x, y)` – relationships
- **Histogram:** `plt.hist(data)` – distributions

## Always import:

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

---

Master these four plot types for most visualization needs

## Object-oriented API:

- **Figure:** The entire image/canvas
- **Axes:** Individual plot within the figure
- Better control than pyplot interface

## Pattern:

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(x, y)
ax.set_xlabel('Date')
ax.set_title('Stock Price')
```

---

Use object-oriented API for complex figures

## Essential styling options:

- **Colors:** Named colors, hex codes, RGB
- **Line styles:** '-', '--', ':', '-.'
- **Markers:** 'o', 's', '^', 'x'
- **Labels:** xlabel, ylabel, title, legend

## Example:

```
ax.plot(x, y, color='blue', linestyle='--', marker='o')
```

---

Consistent styling improves chart readability

## Multiple plots in one figure:

- **Grid:** `plt.subplots(2, 2)` – 2x2 grid
- **Shared axes:** `sharex=True, sharey=True`
- **Indexing:** `axs[0, 1]` for row 0, col 1

## Example:

```
fig, axs = plt.subplots(2, 2, figsize=(12, 8))  
axs[0, 0].plot(x, y)
```

---

Subplots enable comparison across multiple views

## Export to various formats:

- **PDF:** Vector format for publications
- **PNG:** Raster for web/presentations
- **SVG:** Vector for web graphics

## Best practice:

```
plt.savefig('chart.pdf', dpi=300, bbox_inches='tight')  
plt.close() # Free memory
```

---

Use PDF for print; PNG for screen at 150+ dpi

# Global Settings (rcParams)

## Configure default plot appearance:

- **Font size:** All text elements
- **Figure size:** Default dimensions
- **DPI:** Resolution for saving

## Benefits:

- Consistent styling across all plots
- Set once at script start

---

rcParams ensure professional, consistent charts

### Stock price with moving average:

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(df.index, df['Price'], label='Price')
ax.plot(df.index, df['MA20'], label='MA')
ax.fill_between(df.index, lower, upper, alpha=0.3)
ax.legend()
```

---

Layer multiple elements for comprehensive visualizations

### Essential Matplotlib Commands:

Operation	Syntax
Create figure	<code>fig, ax = plt.subplots()</code>
Line plot	<code>ax.plot(x, y)</code>
Bar chart	<code>ax.bar(x, heights)</code>
Scatter	<code>ax.scatter(x, y)</code>
Histogram	<code>ax.hist(data, bins=30)</code>
Labels	<code>ax.set_xlabel('X')</code>
Legend	<code>ax.legend()</code>
Save	<code>plt.savefig('file.pdf')</code>

---

Matplotlib provides complete control over visualizations