



SDEV 1001

Programming Fundamentals

Packages and virtual environments - 3

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

Expectations - What I expect from you

- No Late Assignments
- No Cheating
- Be a good classmate
- Don't waste your time
- Show up to class

Agenda

On the right is what we will cover today.

Jupyter, Pandas & Matplotlib: Exploring Weather Data

Why Use These Tools?

Setting Up Your Environment

Loading Weather Data

Exploring the Data

Filtering and Analyzing

Visualizing Temperature Trends

Visualizing Precipitation

Exercises

Summary

Next Steps

Jupyter, Pandas & Matplotlib: Exploring Weather Data

- Learn how to use Jupyter Notebooks for interactive data analysis
- Use Pandas for data wrangling and Matplotlib for visualization

Why Use These Tools?

- Jupyter: run code, see results, and document your process in one place
- Pandas: clean, filter, and analyze tabular data
- Matplotlib: create charts and visualizations to reveal insights

Useful Documentation:

- [Matplotlib Documentation](#)
- [Pandas Documentation](#)

Setting Up Your Environment

- Create a virtual environment and install the necessary packages:

```
python -m venv venv  
venv\Scripts\activate  
pip install jupyter pandas matplotlib  
pip freeze > requirements.txt
```

- This keeps your project organized and dependencies isolated

Loading Weather Data

- Imagine you have a CSV file: `data/weather.csv`
- Example contents:

```
Date, Temperature, Precipitation, Wind  
2025-11-01, 12.3, 0.0, 15  
2025-11-02, 10.1, 2.5, 12  
2025-11-03, 8.7, 0.0, 20  
2025-11-04, 7.2, 5.1, 10  
2025-11-05, 6.8, 0.0, 8
```

- Load the data:

```
import pandas as pd  
weather = pd.read_csv('data/weather.csv')  
weather.head()
```

Exploring the Data

- Get a summary:

```
weather.info()
```

- See the shape (rows, columns):

```
weather.shape
```

- Basic statistics:

```
weather.describe()
```


Filtering and Analyzing

- Find rainy days:

```
rainy_days = weather[weather['Precipitation'] > 0]  
rainy_days
```

- Find the coldest day:

```
coldest = weather.loc[weather['Temperature'].idxmin()]  
print(coldest)
```

Visualizing Temperature Trends

- Plot temperature over time:

```
import matplotlib.pyplot as plt

plt.plot(weather['Date'], weather['Temperature'], marker='o')
plt.title('Temperature Over Time')
plt.xlabel('Date')
plt.ylabel('Temperature (°C)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Visualizing Precipitation

- Bar chart of precipitation:

```
plt.bar(weather['Date'], weather['Precipitation'])  
plt.title('Daily Precipitation')  
plt.xlabel('Date')  
plt.ylabel('Precipitation (mm)')  
plt.xticks(rotation=45)  
plt.tight_layout()  
plt.show()
```

Exercises

- Try these in new cells:
 1. What was the average wind speed on rainy days?
 2. Which day had the highest precipitation?
 3. Plot a scatter plot of temperature vs wind speed.

Summary

- You set up Jupyter, Pandas, and Matplotlib in a virtual environment
- Loaded and explored a weather dataset
- Filtered, analyzed, and visualized the data
- These skills are essential for data analysis and reporting

Next Steps

- Try with your own dataset (e.g., fitness tracker, sales data)
- Explore more Pandas and Matplotlib features: grouping, custom plots, missing data handling