Lesson Reflection

Key Points

- DataFrames are 2D labeled data structures, like spreadsheets
- Columns store values of the same type, rows hold observations
- Use iloc and loc for indexing rows and columns
- Create DataFrames from lists, dicts, Series, or external data
- Indexes provide efficient access and operations

Reflection Questions

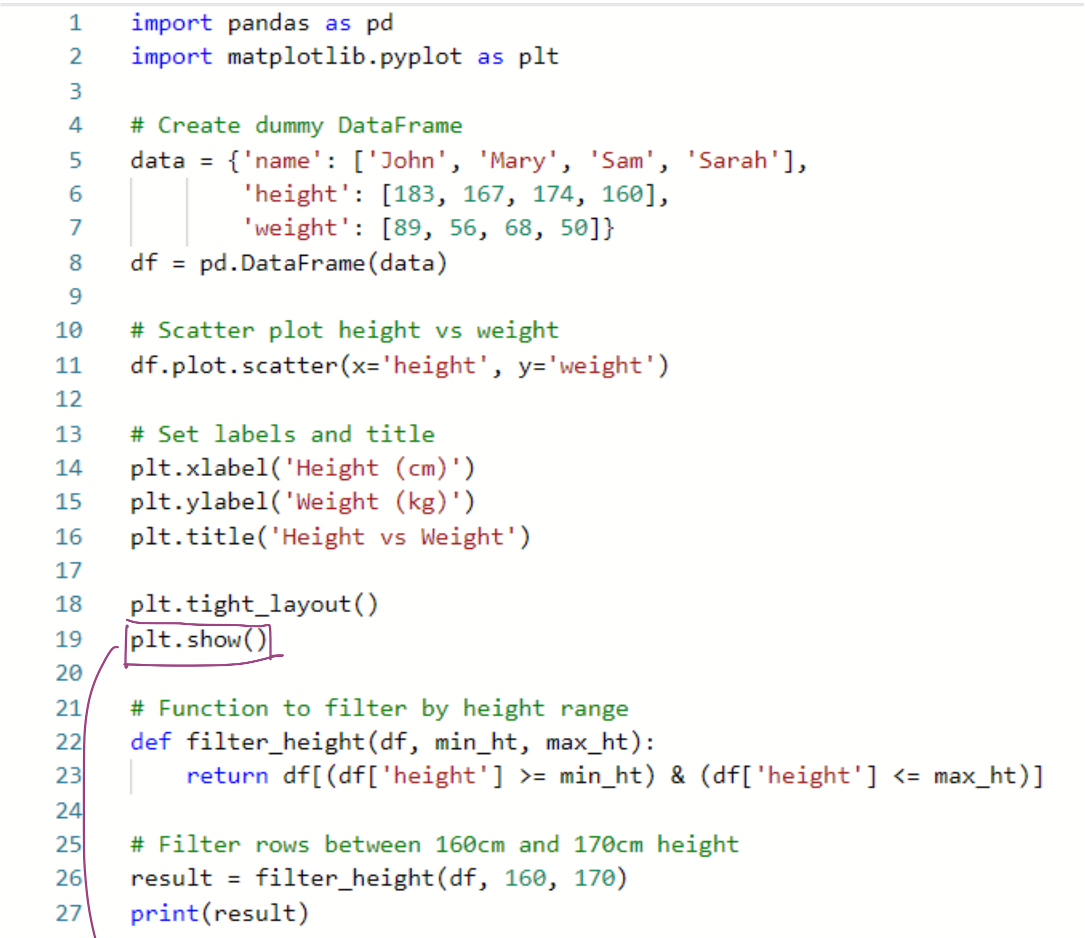
- What real-world data could you model with a DataFrame?
- How are DataFrame columns similar to variables in Python?
- When would you use iloc vs loc for selecting data?
- What are some DataFrame creation options you can use?
- How might DataFrame indexes improve performance?

Challenge Exercises

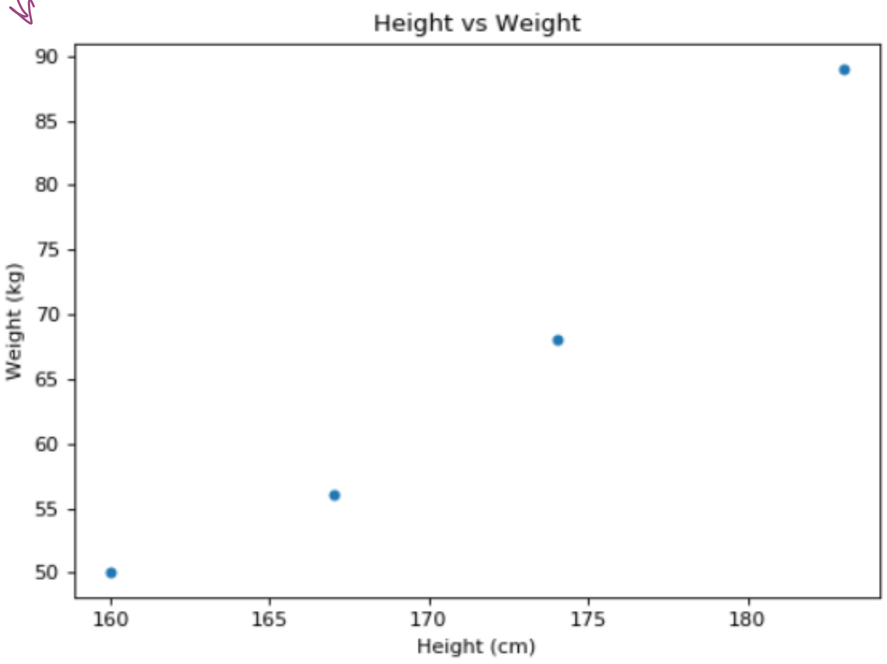
- Create a DataFrame for storing weather data
- Select a row from mid-point using iloc
- Filter rows matching certain numeric criteria with loc
- Load a CSV file into a DataFrame
- Set the index to another column like dates

Code Examples

Using Pandas to Plot Height vs Weight



```python
import pandas as pd
import matplotlib.pyplot as plt

# Create dummy DataFrame
data = {'name': ['John', 'Mary', 'Sam', 'Sarah'],
        'height': [183, 167, 174, 160],
        'weight': [89, 56, 68, 50]}
df = pd.DataFrame(data)

# Scatter plot height vs weight
df.plot.scatter(x='height', y='weight')

# Set labels and title
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.title('Height vs Weight')

plt.tight_layout()
plt.show()

# Function to filter by height range
def filter_height(df, min_ht, max_ht):
    return df[(df['height'] >= min_ht) & (df['height'] <= max_ht)]

# Filter rows between 160cm and 170cm height
result = filter_height(df, 160, 170)
print(result)
```


Height vs Weight
Weight (kg)
Height (cm)