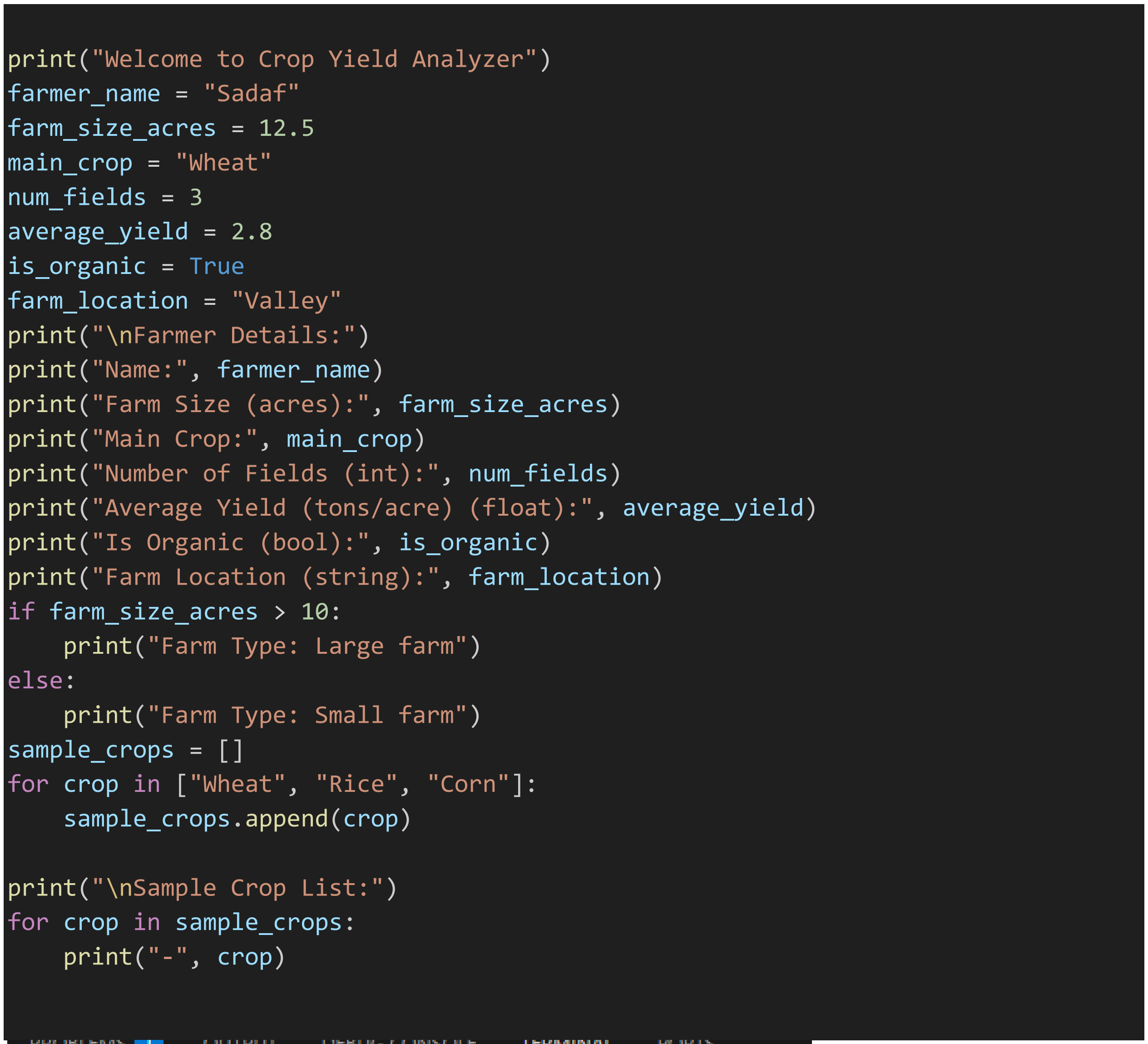
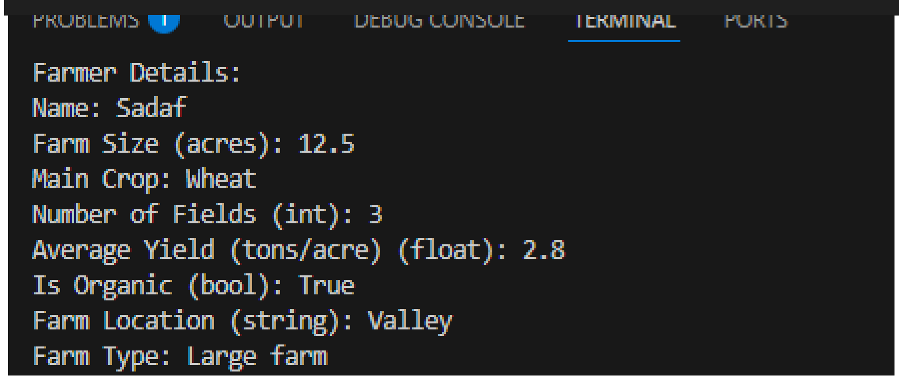
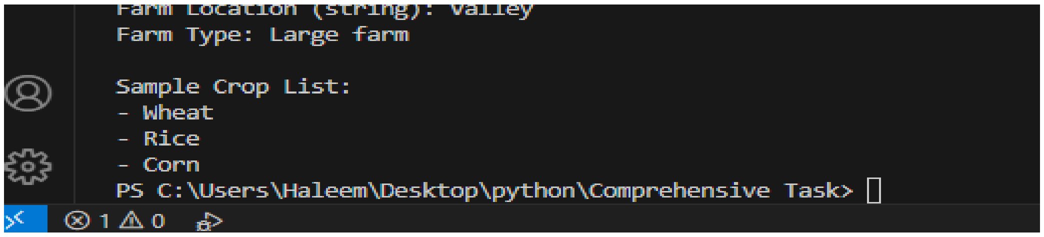


```python
print("Welcome to Crop Yield Analyzer")
farmer_name = "Sadaf"
farm_size_acres = 12.5
main_crop = "Wheat"
num_fields = 3
average_yield = 2.8
is_organic = True
farm_location = "Valley"
print("\nFarmer Details:")
print("Name:", farmer_name)
print("Farm Size (acres):", farm_size_acres)
print("Main Crop:", main_crop)
print("Number of Fields (int):", num_fields)
print("Average Yield (tons/acre) (float):", average_yield)
print("Is Organic (bool):", is_organic)
print("Farm Location (string):", farm_location)
if farm_size_acres > 10:
    print("Farm Type: Large farm")
else:
    print("Farm Type: Small farm")
sample_crops = []
for crop in ["Wheat", "Rice", "Corn"]:
    sample_crops.append(crop)

print("\nSample Crop List:")
for crop in sample_crops:
    print("-", crop)
```

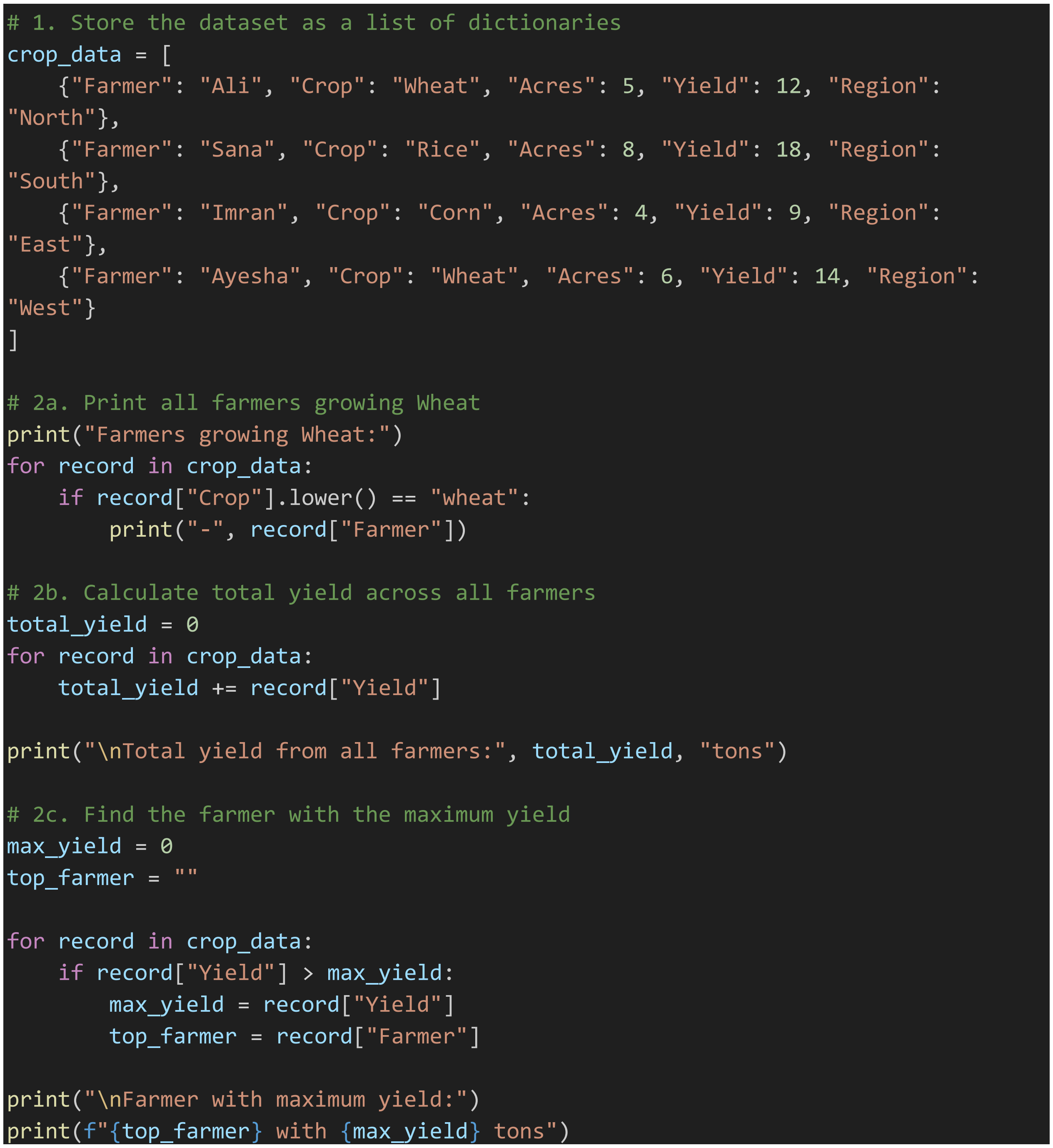


```
Farmer Details:
Name: Sadaf
Farm Size (acres): 12.5
Main Crop: Wheat
Number of Fields (int): 3
Average Yield (tons/acre) (float): 2.8
Is Organic (bool): True
Farm Location (string): Valley
Farm Type: Large farm
```



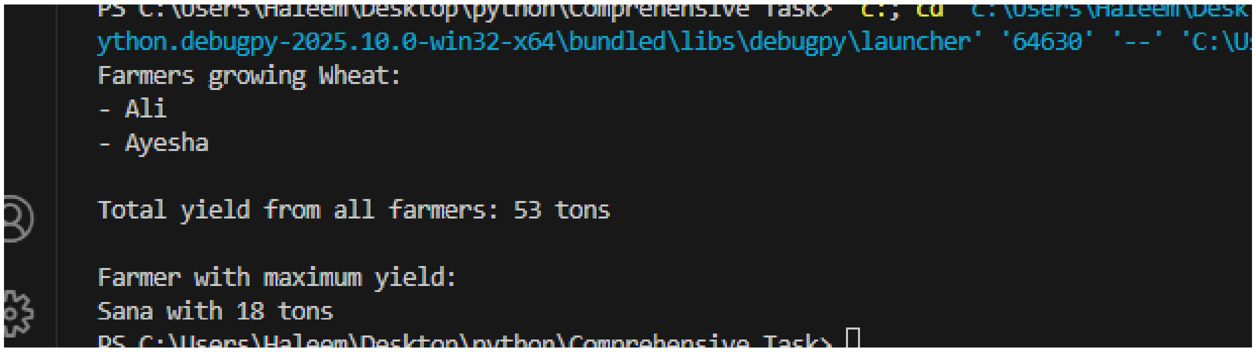
```
Farm Location (string): Valley
Farm Type: Large farm

Sample Crop List:
- Wheat
- Rice
- Corn
PS C:\Users\Haleem\Desktop\python\Comprehensive Task> 
```

CODE:02



```python
# 1. Store the dataset as a list of dictionaries
crop_data = [
    {"Farmer": "Ali", "Crop": "Wheat", "Acres": 5, "Yield": 12, "Region":
"North"},
    {"Farmer": "Sana", "Crop": "Rice", "Acres": 8, "Yield": 18, "Region":
"South"},
    {"Farmer": "Imran", "Crop": "Corn", "Acres": 4, "Yield": 9, "Region":
"East"},
    {"Farmer": "Ayesha", "Crop": "Wheat", "Acres": 6, "Yield": 14, "Region":
"West"}
]

# 2a. Print all farmers growing Wheat
print("Farmers growing Wheat:")
for record in crop_data:
    if record["Crop"].lower() == "wheat":
        print("-", record["Farmer"])

# 2b. Calculate total yield across all farmers
total_yield = 0
for record in crop_data:
    total_yield += record["Yield"]

print("\nTotal yield from all farmers:", total_yield, "tons")

# 2c. Find the farmer with the maximum yield
max_yield = 0
top_farmer = ""

for record in crop_data:
    if record["Yield"] > max_yield:
        max_yield = record["Yield"]
        top_farmer = record["Farmer"]

print("\nFarmer with maximum yield:")
print(f"{top_farmer} with {max_yield} tons")
```

output:



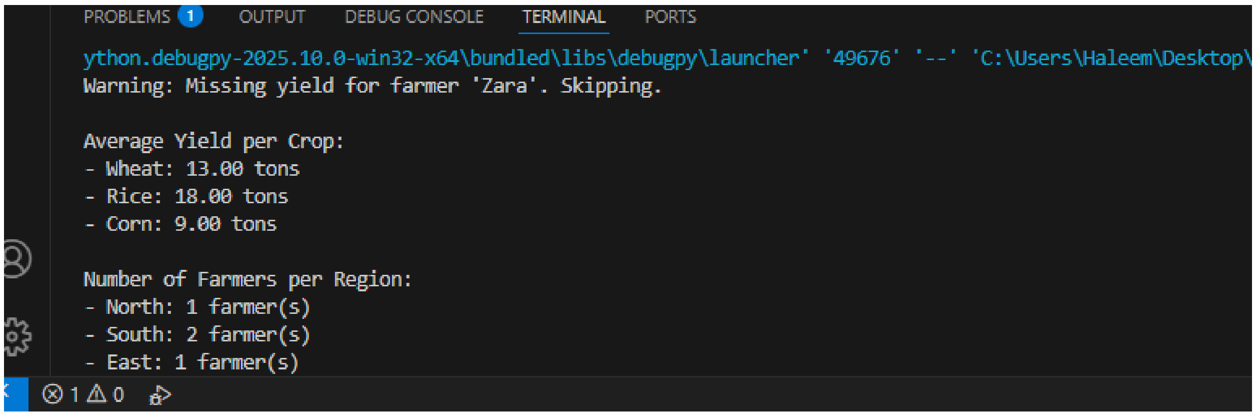
```
PS C:\Users\Haleem\Desktop\python\Comprehensive Task>  c:; cd 'c:\Users\Haleem\Desk
ython.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '64630' '--' 'C:\U
Farmers growing Wheat:
- Ali
- Ayesha

Total yield from all farmers: 53 tons

Farmer with maximum yield:
Sana with 18 tons
PS C:\Users\Haleem\Desktop\python\Comprehensive Task>
```

CODE:03

```
# Sample dataset (with one missing yield to demonstrate error handling)
crop_data = [
    {"Farmer": "Ali", "Crop": "Wheat", "Acres": 5, "Yield": 12, "Region":
"North"},
    {"Farmer": "Sana", "Crop": "Rice", "Acres": 8, "Yield": 18, "Region":
"South"},
    {"Farmer": "Imran", "Crop": "Corn", "Acres": 4, "Yield": 9, "Region":
"East"},
    {"Farmer": "Ayesha", "Crop": "Wheat", "Acres": 6, "Yield": 14, "Region":
"West"},
    {"Farmer": "Zara", "Crop": "Rice", "Acres": 7, "Region": "South"}  # Missing
'Yield'
]

# Function to calculate average yield per crop type
def average_yield_per_crop(data):
    crop_totals = {}
    crop_counts = {}

    for record in data:
        try:
            crop = record["Crop"]
            yield_value = record["Yield"]
            crop_totals[crop] = crop_totals.get(crop, 0) + yield_value
            crop_counts[crop] = crop_counts.get(crop, 0) + 1
        except KeyError:
            print(f"Warning: Missing yield for farmer '{record.get('Farmer',
'Unknown')}'. Skipping.")

    print("\nAverage Yield per Crop:")
```

```python
    for crop in crop_totals:
        avg = crop_totals[crop] / crop_counts[crop]
        print(f"- {crop}: {avg:.2f} tons")

# Function to count how many farmers are in each region
def count_farmers_per_region(data):
    region_counts = {}
    for record in data:
        region = record.get("Region", "Unknown")
        region_counts[region] = region_counts.get(region, 0) + 1

    print("\nNumber of Farmers per Region:")
    for region, count in region_counts.items():
        print(f"- {region}: {count} farmer(s)")

# Run the functions
average_yield_per_crop(crop_data)
count_farmers_per_region(crop_data)
```

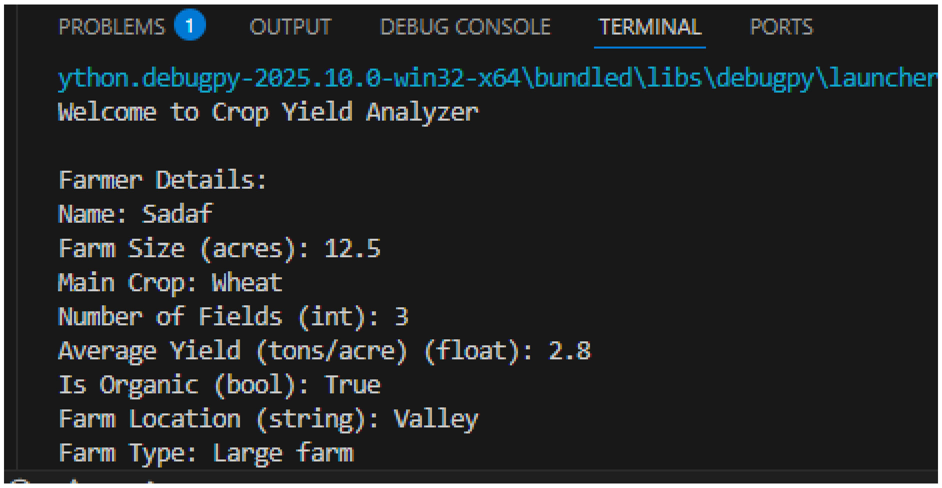
OUTPUT:



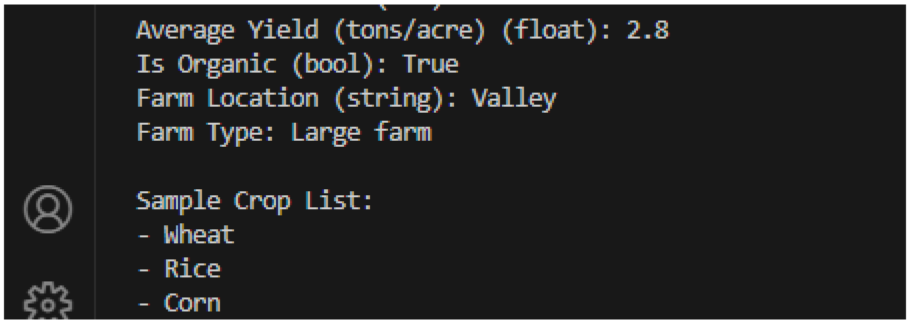
```
ython.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '49676' '--' 'C:\Users\Haleem\Desktop\
Warning: Missing yield for farmer 'Zara'. Skipping.

Average Yield per Crop:
- Wheat: 13.00 tons
- Rice: 18.00 tons
- Corn: 9.00 tons

Number of Farmers per Region:
- North: 1 farmer(s)
- South: 2 farmer(s)
- East: 1 farmer(s)
```

```python
# Script: Crop Yield Analyzer

# 1. Print a welcome message
print("Welcome to Crop Yield Analyzer")

# 2. Variables to store farmer details
farmer_name = "Sadaf"
farm_size_acres = 12.5  # float
main_crop = "Wheat"

# 3. Demonstrating data types
```

```python
num_fields = 3            # Integer
average_yield = 2.8       # Float
is_organic = True         # Boolean
farm_location = "Valley" # String

# Print farmer details and data types
print("\nFarmer Details:")
print("Name:", farmer_name)
print("Farm Size (acres):", farm_size_acres)
print("Main Crop:", main_crop)
print("Number of Fields (int):", num_fields)
print("Average Yield (tons/acre) (float):", average_yield)
print("Is Organic (bool):", is_organic)
print("Farm Location (string):", farm_location)

# 4. Conditional check on farm size
if farm_size_acres > 10:
    print("Farm Type: Large farm")
else:
    print("Farm Type: Small farm")

# 5. Loop to generate a sample list of crops
sample_crops = []
for crop in ["Wheat", "Rice", "Corn"]:
    sample_crops.append(crop)

print("\nSample Crop List:")
for crop in sample_crops:
    print("-", crop)
```

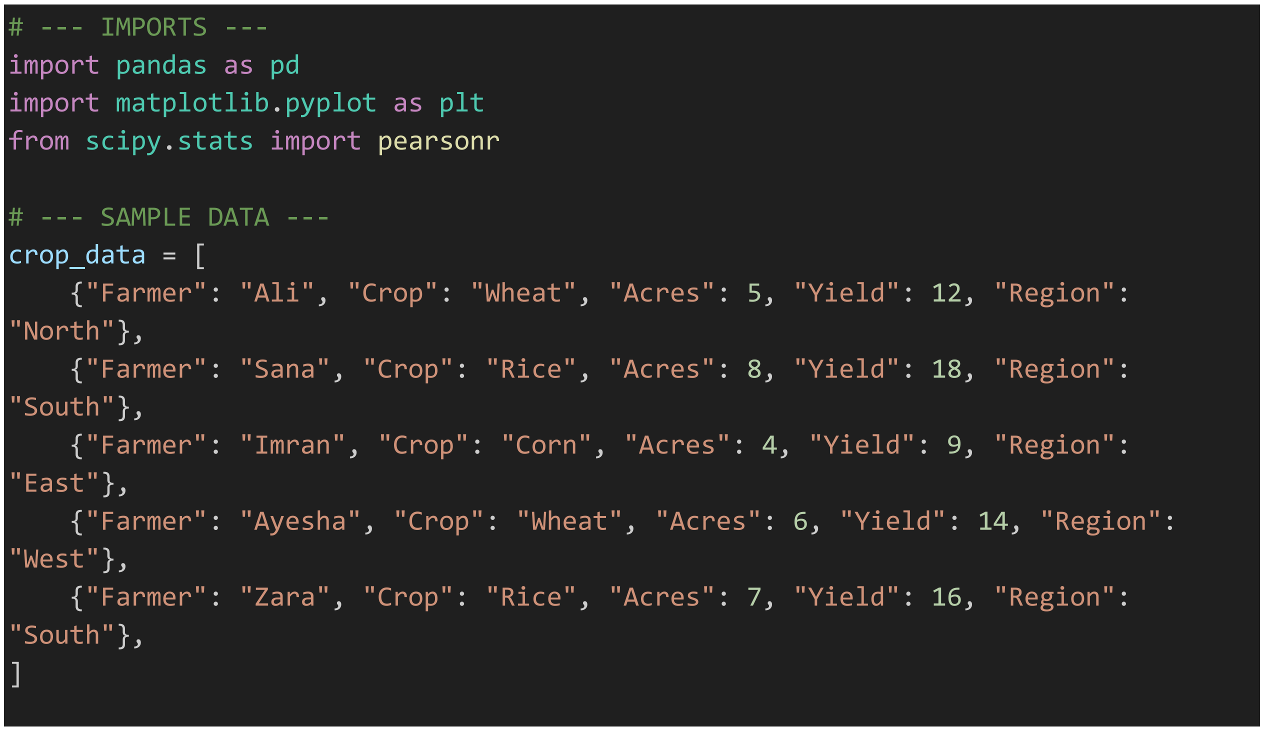

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ython.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher
Welcome to Crop Yield Analyzer

Farmer Details:
Name: Sadaf
Farm Size (acres): 12.5
Main Crop: Wheat
Number of Fields (int): 3
Average Yield (tons/acre) (float): 2.8
Is Organic (bool): True
Farm Location (string): Valley
Farm Type: Large farm
```



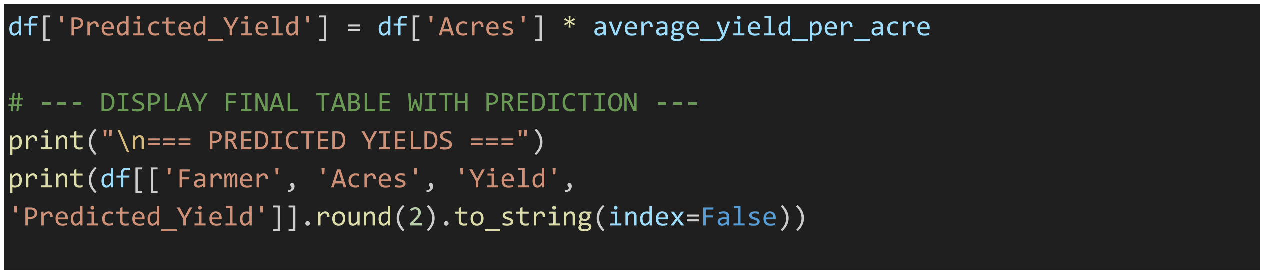
```
Average Yield (tons/acre) (float): 2.8
Is Organic (bool): True
Farm Location (string): Valley
Farm Type: Large farm

Sample Crop List:
- Wheat
- Rice
- Corn
```

CODE:04

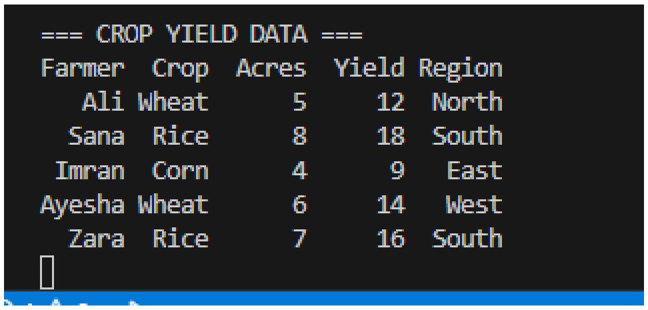


```python
# --- IMPORTS ---
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import pearsonr

# --- SAMPLE DATA ---
crop_data = [
    {"Farmer": "Ali", "Crop": "Wheat", "Acres": 5, "Yield": 12, "Region":
"North"},
    {"Farmer": "Sana", "Crop": "Rice", "Acres": 8, "Yield": 18, "Region":
"South"},
    {"Farmer": "Imran", "Crop": "Corn", "Acres": 4, "Yield": 9, "Region":
"East"},
    {"Farmer": "Ayesha", "Crop": "Wheat", "Acres": 6, "Yield": 14, "Region":
"West"},
    {"Farmer": "Zara", "Crop": "Rice", "Acres": 7, "Yield": 16, "Region":
"South"},
]
```

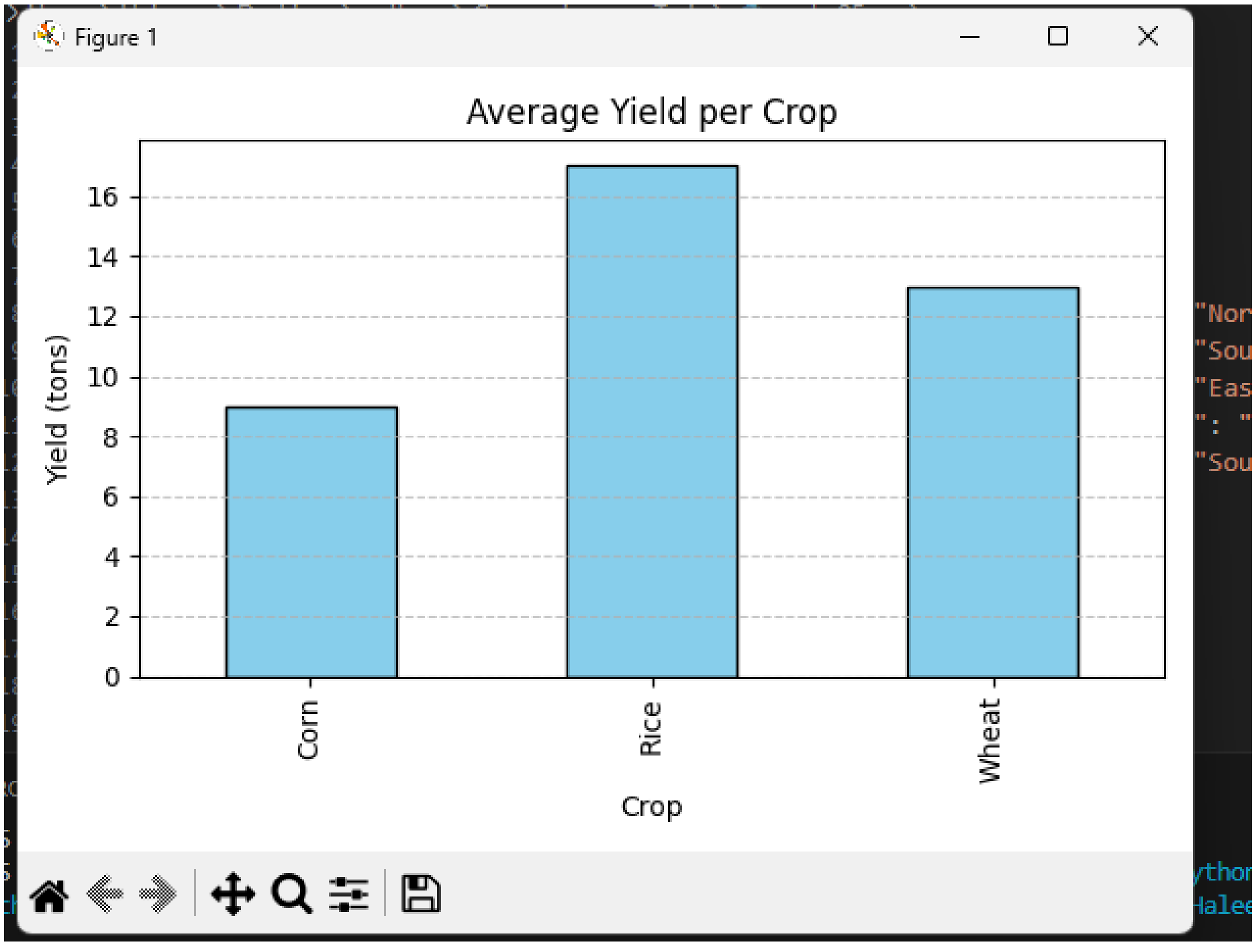
```python
# --- CONVERT TO PANDAS DATAFRAME ---
df = pd.DataFrame(crop_data)

# --- TABLE VIEW ---
print("\n=== CROP YIELD DATA ===")
print(df.to_string(index=False))

# --- 1. BAR CHART: Average Yield per Crop ---
avg_yield_per_crop = df.groupby('Crop')['Yield'].mean()
plt.figure(figsize=(6,4))
avg_yield_per_crop.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Average Yield per Crop')
plt.ylabel('Yield (tons)')
plt.xlabel('Crop')
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

# --- 2. PIE CHART: Percentage of Farmers by Region ---
region_counts = df['Region'].value_counts()
plt.figure(figsize=(6,6))
region_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90,
colors=plt.cm.Pastel1.colors)
plt.title('Percentage of Farmers by Region')
plt.ylabel('')
plt.tight_layout()
plt.show()

# --- 3. SCATTER PLOT: Acres vs Yield ---
plt.figure(figsize=(6,4))
plt.scatter(df['Acres'], df['Yield'], color='green')
plt.title('Acres vs Yield')
plt.xlabel('Acres')
plt.ylabel('Yield (tons)')
plt.grid(True)
plt.tight_layout()
plt.show()

# --- 4. CORRELATION ANALYSIS ---
correlation, _ = pearsonr(df['Acres'], df['Yield'])
print(f"\nCorrelation (Acres vs Yield): {correlation:.2f}")

# --- 5. SIMPLE YIELD PREDICTION MODEL ---
# Predicted Yield = Acres × Average Yield per Acre (dataset-wide)
average_yield_per_acre = (df['Yield'].sum()) / (df['Acres'].sum())
```



```python
df['Predicted_Yield'] = df['Acres'] * average_yield_per_acre

# --- DISPLAY FINAL TABLE WITH PREDICTION ---
print("\n=== PREDICTED YIELDS ===")
print(df[['Farmer', 'Acres', 'Yield',
'Predicted_Yield']].round(2).to_string(index=False))
```



```
=== CROP YIELD DATA ===
Farmer  Crop  Acres  Yield Region
   Ali Wheat      5     12  North
  Sana  Rice      8     18  South
 Imran  Corn      4      9   East
Ayesha Wheat      6     14   West
  Zara  Rice      7     16  South
```



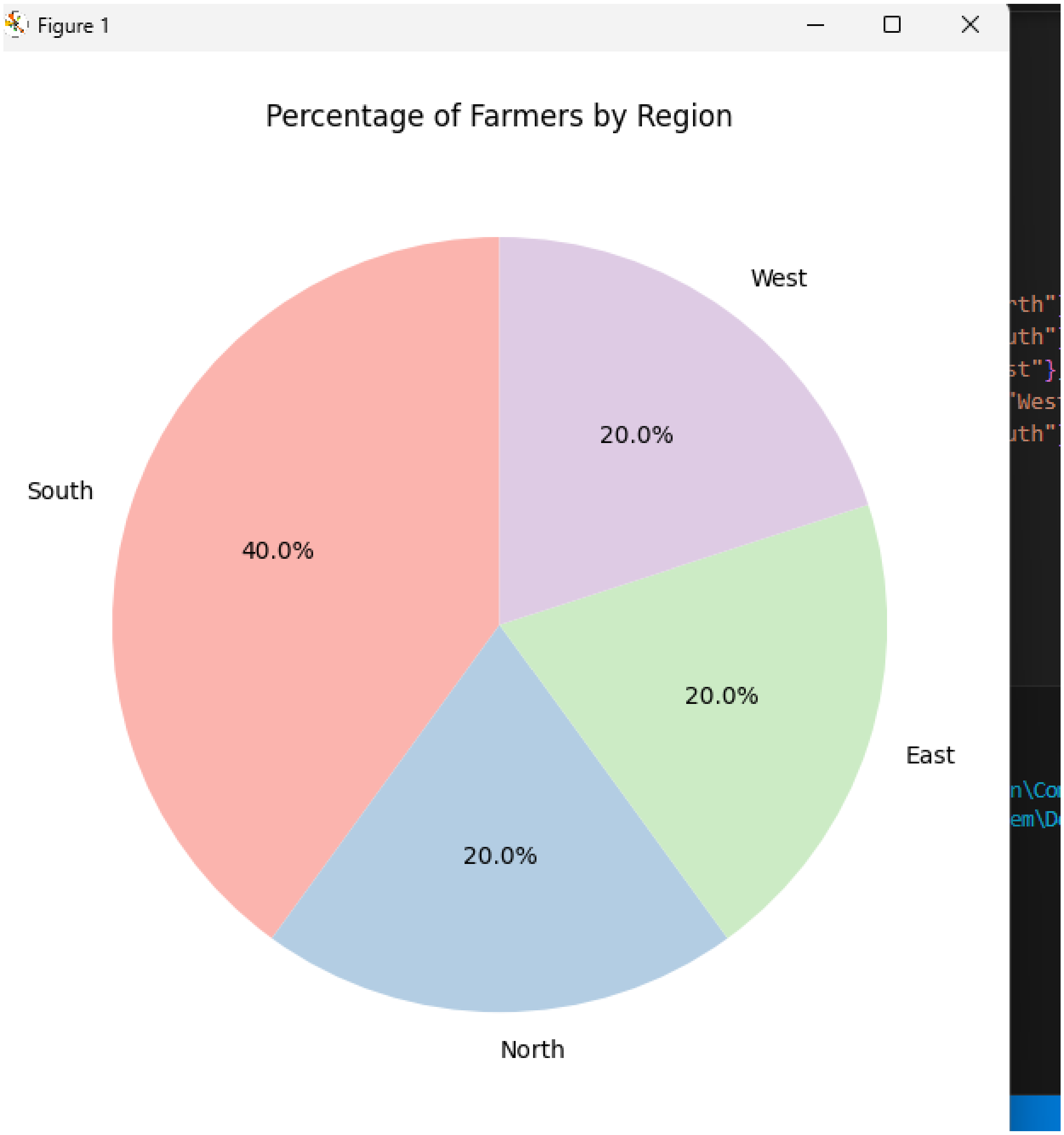


Figure 1
Percentage of Farmers by Region
West
20.0%
South
40.0%
20.0%
East
20.0%
North

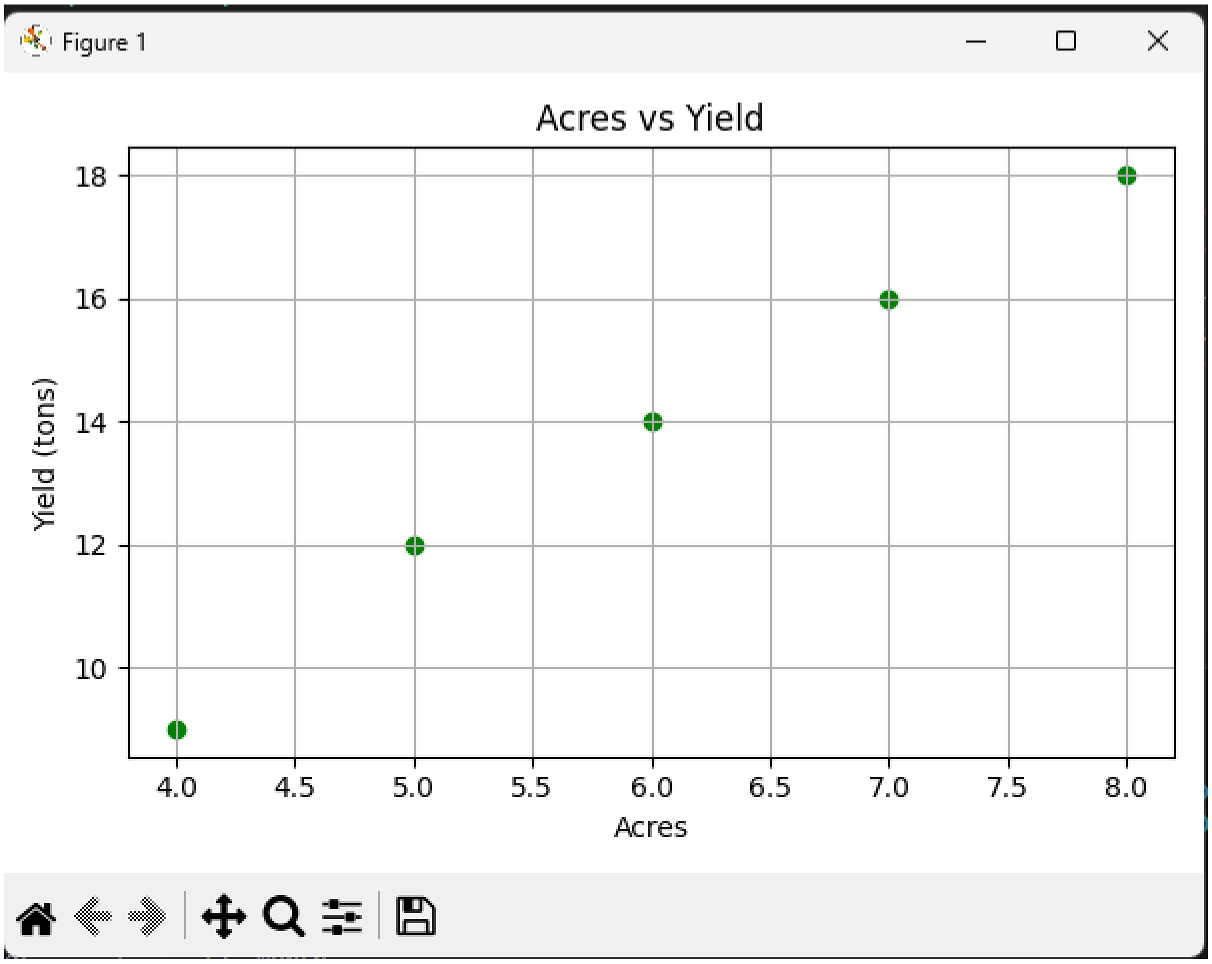


Figure 1
Acres vs Yield
Yield (tons)
18
16
14
12
10
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Acres