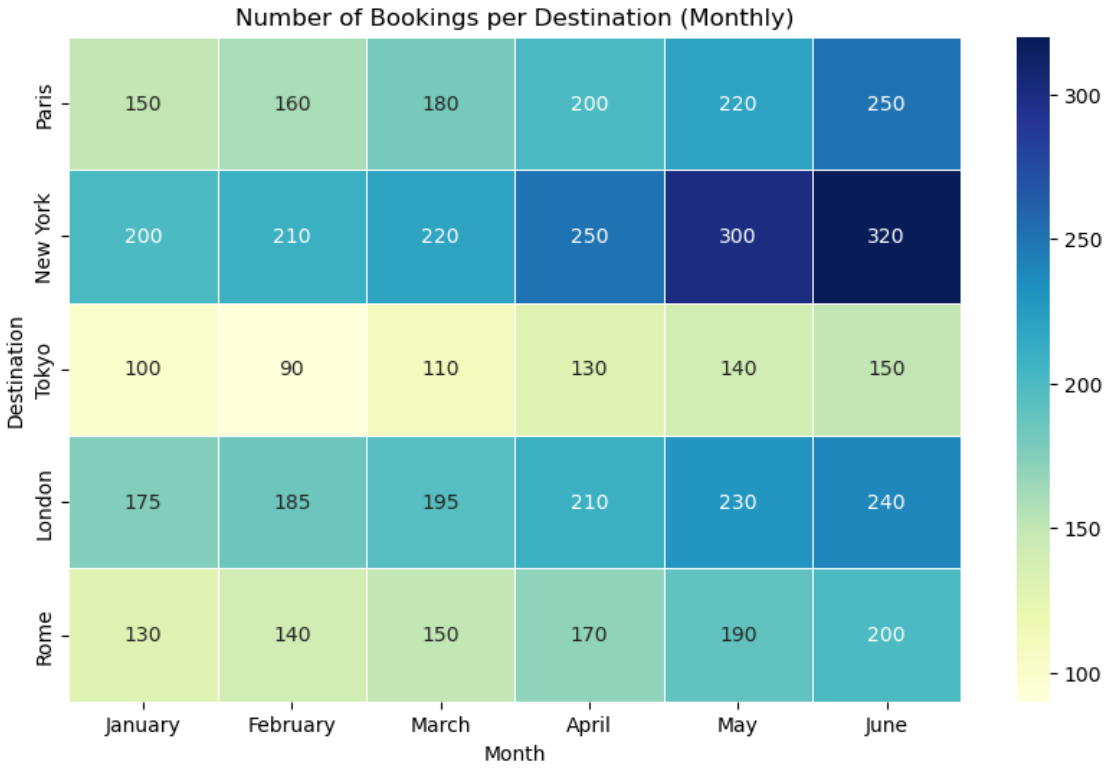
matplotlin4

September 23, 2024

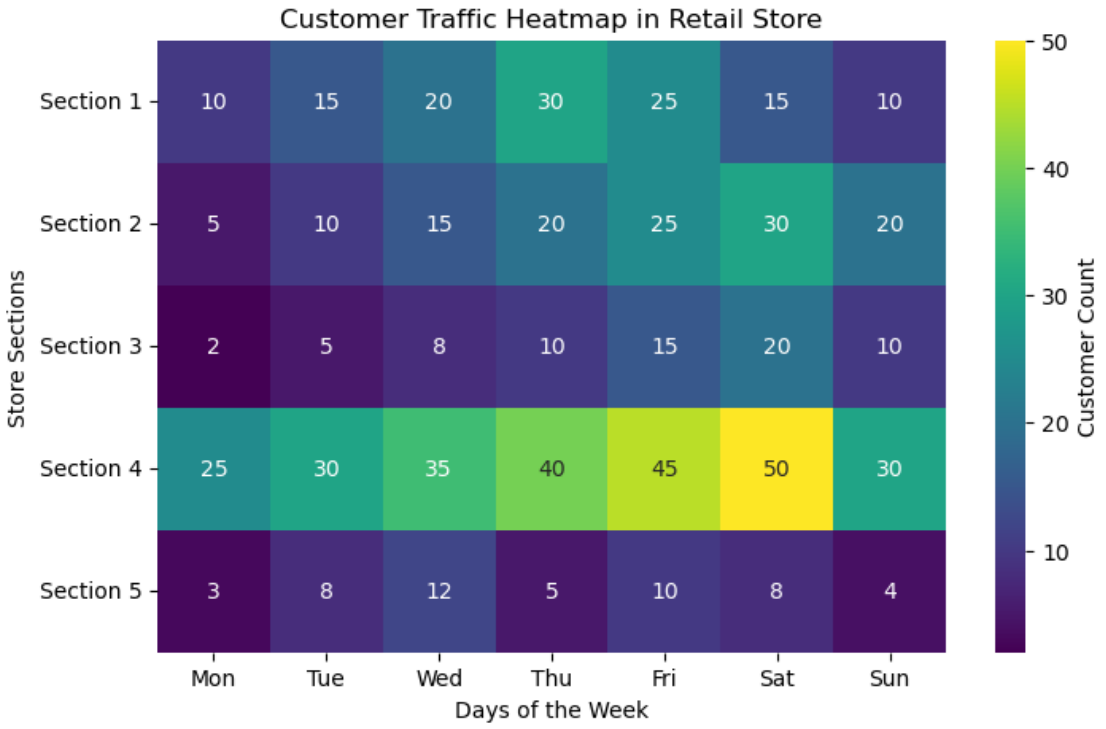
[ ]:

[20]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Sample dataset: Number of bookings per destination per month
data = {
    'Destination': ['Paris', 'New York', 'Tokyo', 'London', 'Rome'],
    'January': [150, 200, 100, 175, 130],
    'February': [160, 210, 90, 185, 140],
    'March': [180, 220, 110, 195, 150],
    'April': [200, 250, 130, 210, 170],
    'May': [220, 300, 140, 230, 190],
    'June': [250, 320, 150, 240, 200]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Set the 'Destination' as the index
df.set_index('Destination', inplace=True)

# Create the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df, annot=True, cmap='YlGnBu', fmt='d', linewidths=.5)
plt.title('Number of Bookings per Destination (Monthly)')
plt.xlabel('Month')
plt.ylabel('Destination')
plt.show()
```



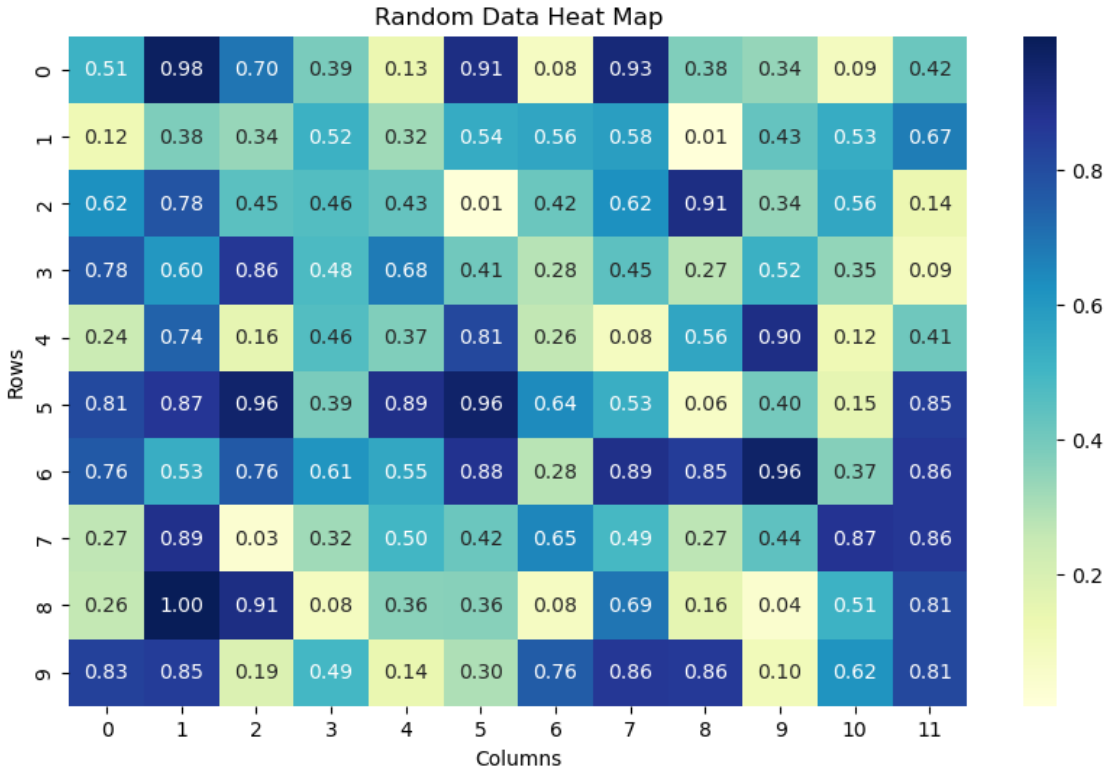
[15]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Simulated data for customer traffic in different sections of a store
# Rows represent different sections and columns represent days of the week
data = np.array([
    [10, 15, 20, 30, 25, 15, 10],  # Section 1
    [5, 10, 15, 20, 25, 30, 20],   # Section 2
    [2, 5, 8, 10, 15, 20, 10],     # Section 3
    [25, 30, 35, 40, 45, 50, 30],  # Section 4 (high traffic area)
    [3, 8, 12, 5, 10, 8, 4],       # Section 5
])

# Create the heatmap
plt.figure(figsize=(8, 5))
sns.heatmap(data, cmap='viridis', annot=True, fmt='d', cbar_kws={'label':␣
 ↪'Customer Count'})

# Set labels and title
plt.title('Customer Traffic Heatmap in Retail Store')
plt.xlabel('Days of the Week')
```

```
plt.ylabel('Store Sections')
plt.xticks(ticks=np.arange(7) + 0.5, labels=['Mon', 'Tue', 'Wed', 'Thu', 'Fri',␣
 ↪'Sat', 'Sun'])
plt.yticks(ticks=np.arange(5) + 0.5, labels=[f'Section {i+1}' for i in␣
 ↪range(5)], rotation=0)

# Show the plot
plt.show()
```

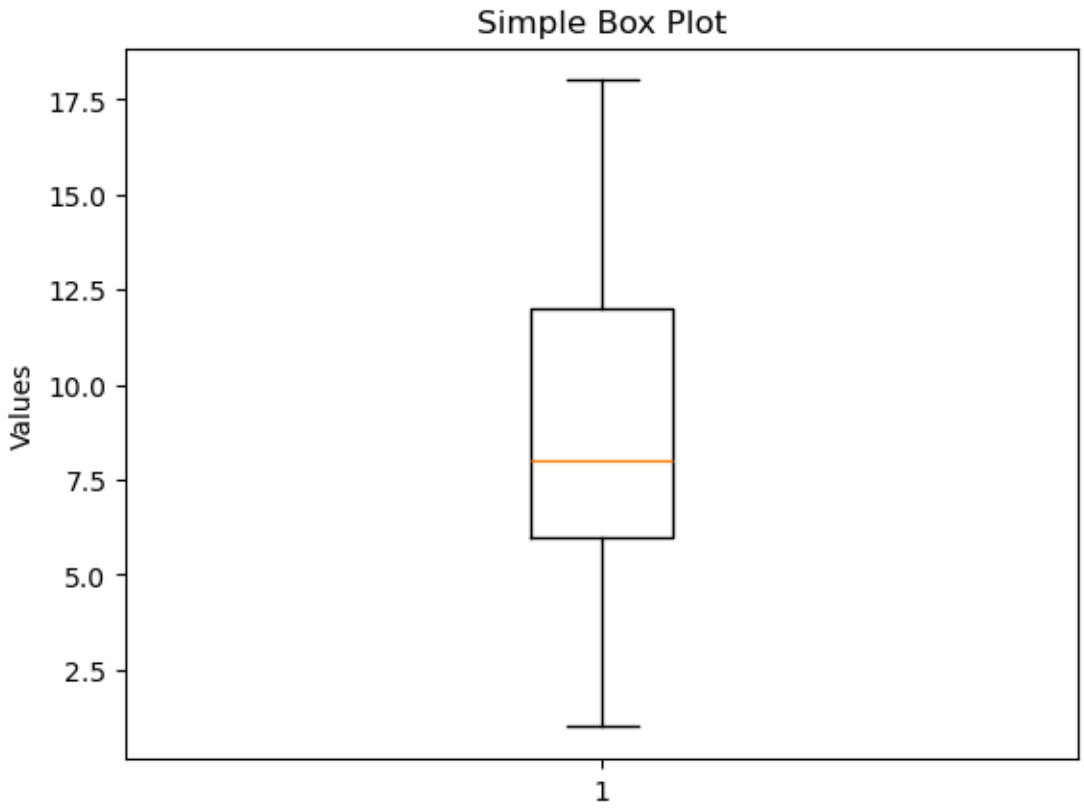


[16]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Generate random data
data = np.random.rand(10, 12)  # 10 rows, 12 columns

# Create a heat map
plt.figure(figsize=(10, 6))
sns.heatmap(data, cmap='YlGnBu', annot=True, fmt='.2f')

# Add labels and title
plt.title('Random Data Heat Map')
plt.xlabel('Columns')
```

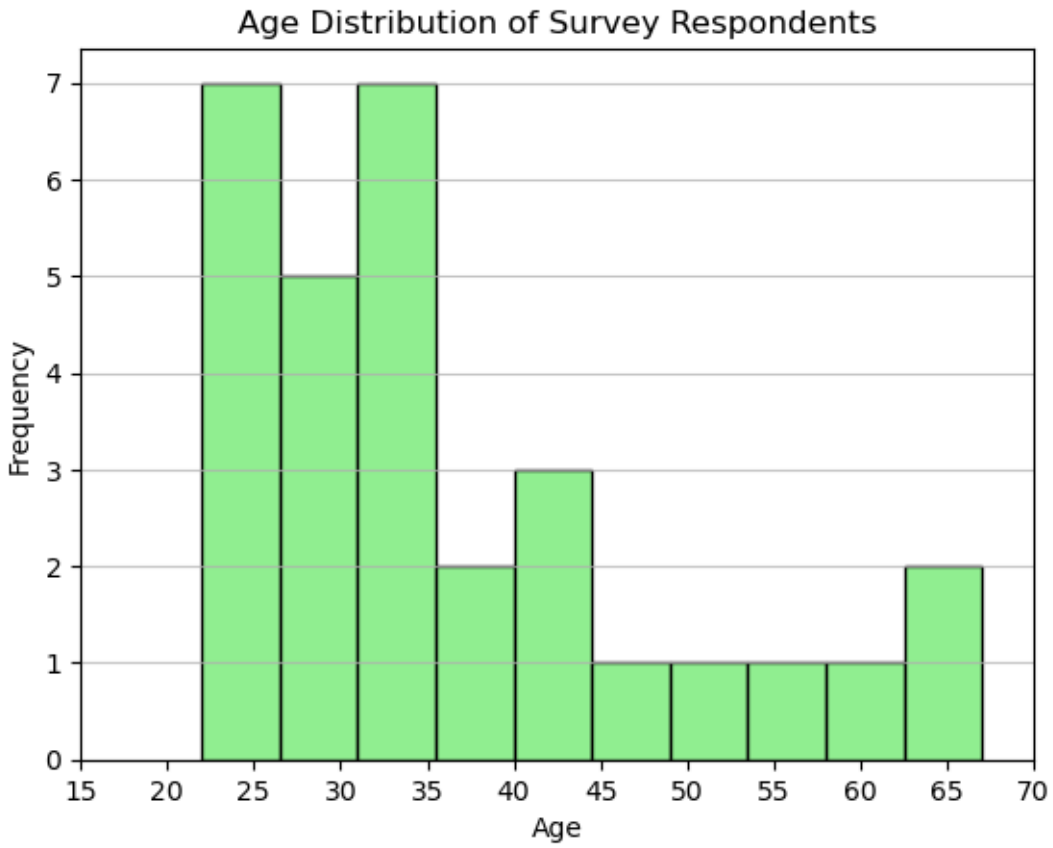
```
plt.ylabel('Rows')

# Show the plot
plt.show()
```



[10]:
```
import matplotlib.pyplot as plt

# Sample data
data = [1, 2, 5, 6, 7, 8, 8, 9, 10, 12, 14, 15, 18]

# Create a box plot
plt.boxplot(data)

# Add titles and labels
plt.title('Simple Box Plot')
plt.ylabel('Values')

# Show the plot
plt.show()
```



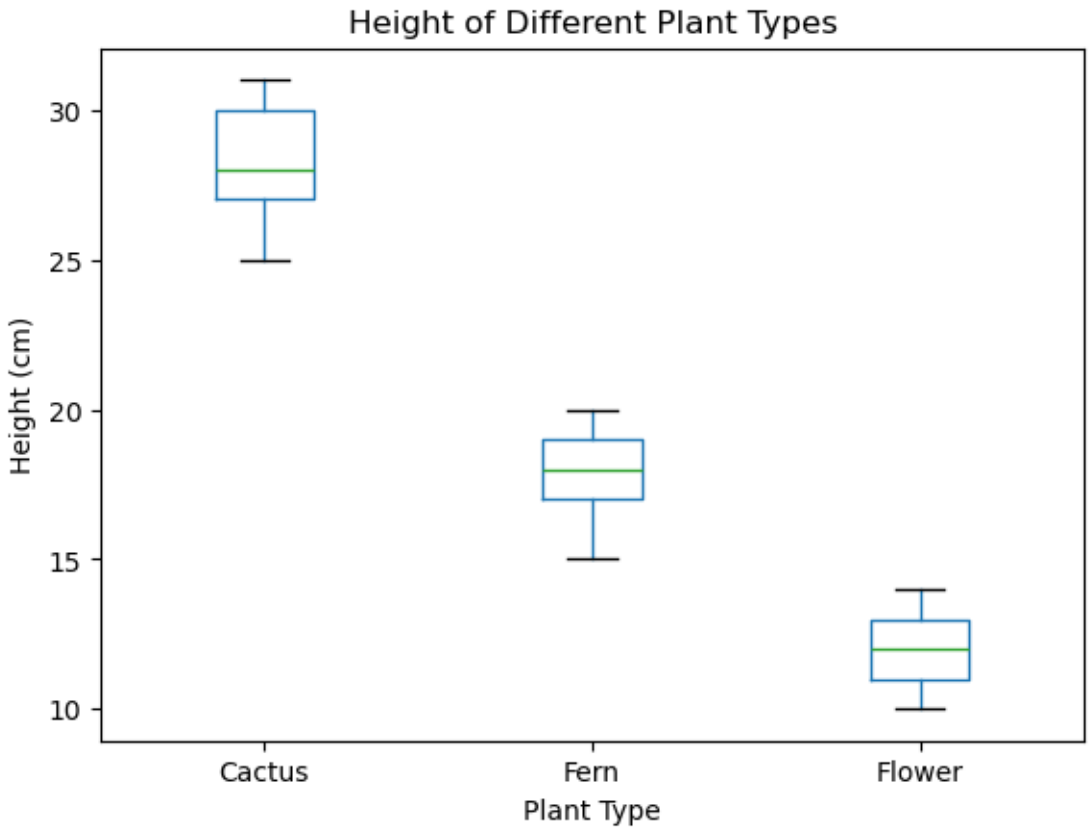
[18]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Sample dataset of ages
data = {
    'Age': [23, 25, 31, 35, 42, 28, 37, 30, 45, 23,
            29, 41, 22, 33, 26, 38, 40, 27, 35, 34,
            50, 55, 60, 63, 67, 22, 25, 29, 31, 33]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Plot the histogram
plt.hist(df['Age'], bins=10, color='lightgreen', edgecolor='black')
plt.title('Age Distribution of Survey Respondents')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.xticks(range(15, 75, 5))  # Set x-ticks for better readability
plt.grid(axis='y', alpha=0.75)
plt.show()
```



[19]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Sample dataset of plant heights (in cm)
data = {
    'Plant Type': ['Fern', 'Fern', 'Fern', 'Fern', 'Fern',
                   'Cactus', 'Cactus', 'Cactus', 'Cactus', 'Cactus',
                   'Flower', 'Flower', 'Flower', 'Flower', 'Flower'],
    'Height': [15, 18, 20, 17, 19,
               25, 30, 28, 27, 31,
               10, 12, 14, 11, 13]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Create a box plot
plt.figure(figsize=(8, 5))
df.boxplot(column='Height', by='Plant Type', grid=False)
plt.title('Height of Different Plant Types')
```

```
plt.suptitle('')  # Suppress the default title
plt.xlabel('Plant Type')
plt.ylabel('Height (cm)')
plt.show()
```

```
<Figure size 800x500 with 0 Axes>
```

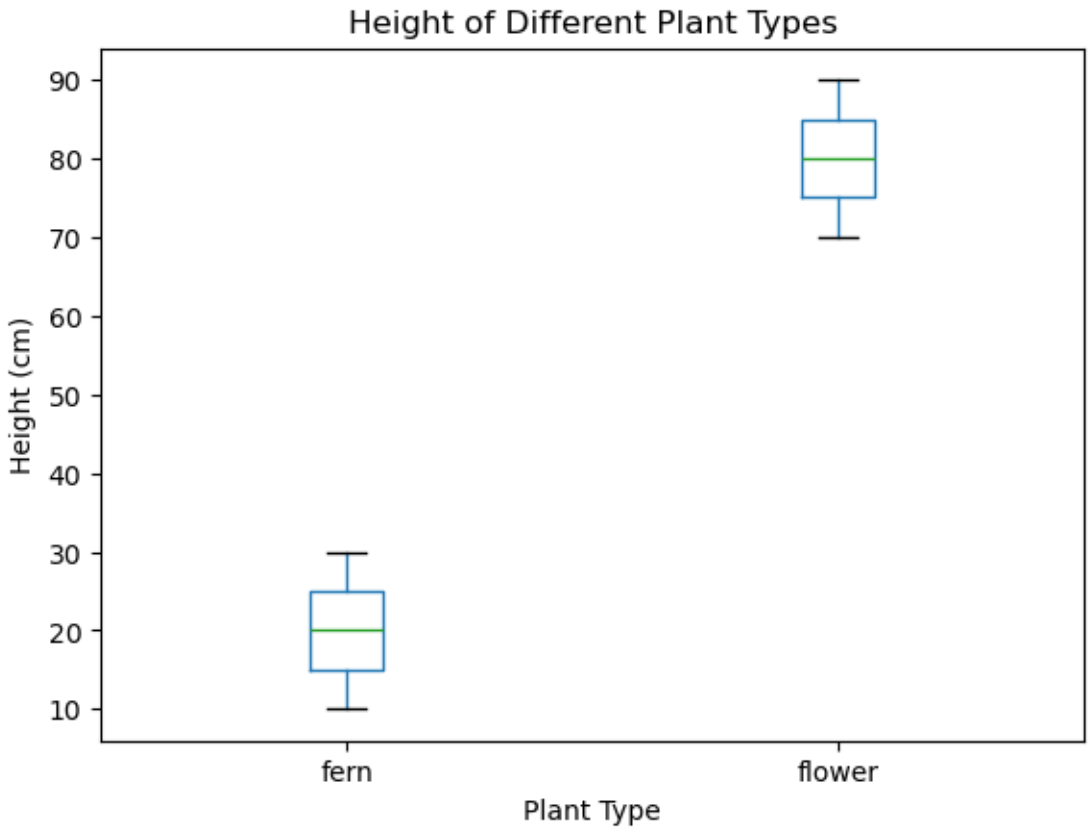


[3]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Initialize an empty list to hold the data
data = {
    'Plant Type': [],
    'Height': []
}

# Function to collect user input
def collect_data():
    while True:
        plant_type = input("Enter the plant type (or 'done' to finish): ")
```

```
        if plant_type.lower() == 'done':
            break
        try:
            height = float(input(f"Enter the height (in cm) for {plant_type}:␣
 ↪"))
            data['Plant Type'].append(plant_type)
            data['Height'].append(height)
        except ValueError:
            print("Please enter a valid height.")

# Collect data from the user
collect_data()

# Create a DataFrame
df = pd.DataFrame(data)

# Create a box plot if there is data
if not df.empty:
    plt.figure(figsize=(8, 5))
    df.boxplot(column='Height', by='Plant Type', grid=False)
    plt.title('Height of Different Plant Types')
    plt.suptitle('')  # Suppress the default title
    plt.xlabel('Plant Type')
    plt.ylabel('Height (cm)')
    plt.show()
else:
    print("No data entered.")
```

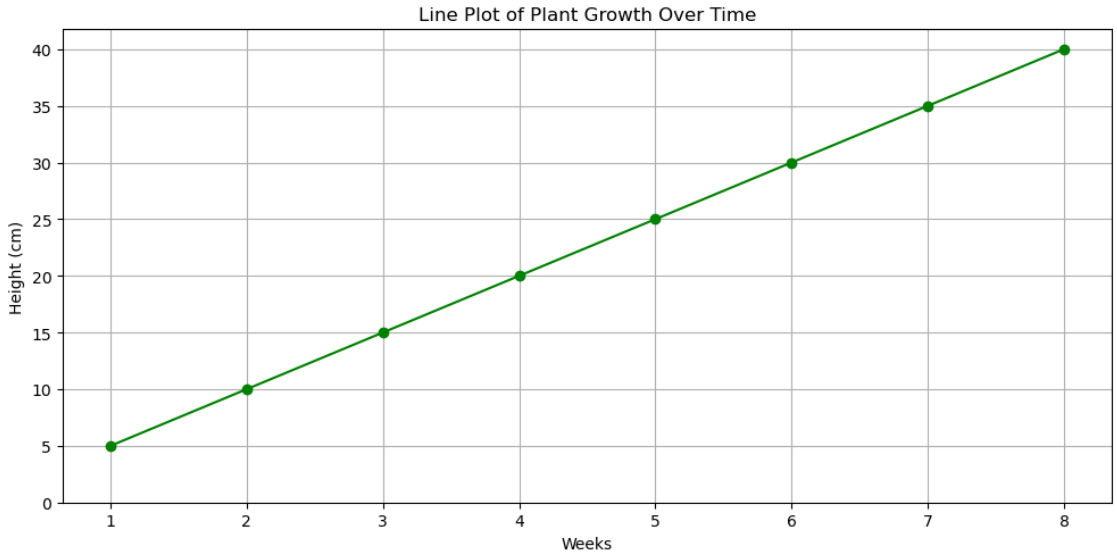
```
Enter the plant type (or 'done' to finish):  fern
Enter the height (in cm) for fern:  10
Enter the plant type (or 'done' to finish):  fern
Enter the height (in cm) for fern:  20
Enter the plant type (or 'done' to finish):  fern
Enter the height (in cm) for fern:  30
Enter the plant type (or 'done' to finish):  flower
Enter the height (in cm) for flower:  70
Enter the plant type (or 'done' to finish):  flower
Enter the height (in cm) for flower:  80
Enter the plant type (or 'done' to finish):  flower
Enter the height (in cm) for flower:  90
Enter the plant type (or 'done' to finish):  done
```

```
<Figure size 800x500 with 0 Axes>
```

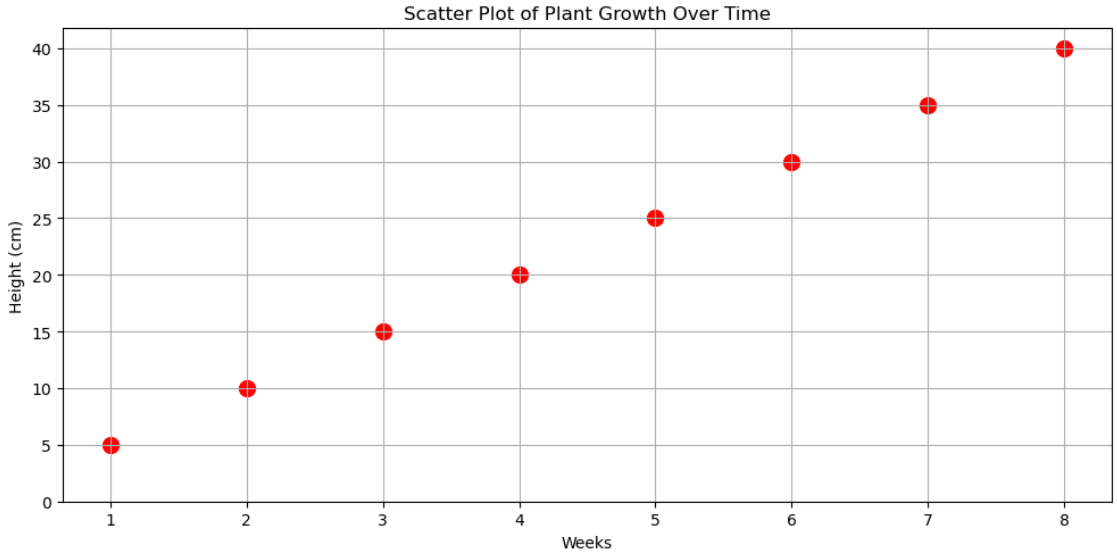


[4]:
```
import matplotlib.pyplot as plt

# Sample data: Weeks and corresponding heights of a plant
weeks = [1, 2, 3, 4, 5, 6, 7, 8]
heights = [5, 10, 15, 20, 25, 30, 35, 40]  # Plant height in cm

# Create a figure with multiple subplots
fig, axs = plt.subplots(3, 1, figsize=(10, 15))

# Line Plot
axs[0].plot(weeks, heights, marker='o', linestyle='-', color='green')
axs[0].set_title('Line Plot of Plant Growth Over Time')
axs[0].set_xlabel('Weeks')
axs[0].set_ylabel('Height (cm)')
axs[0].set_xticks(weeks)
axs[0].set_yticks(range(0, 45, 5))
axs[0].grid(True)

# Bar Plot
axs[1].bar(weeks, heights, color='lightblue')
```

```
axs[1].set_title('Bar Plot of Plant Growth Over Time')
axs[1].set_xlabel('Weeks')
axs[1].set_ylabel('Height (cm)')
axs[1].set_xticks(weeks)
axs[1].set_yticks(range(0, 45, 5))
axs[1].grid(axis='y')

# Scatter Plot
axs[2].scatter(weeks, heights, color='red', s=100)
axs[2].set_title('Scatter Plot of Plant Growth Over Time')
axs[2].set_xlabel('Weeks')
axs[2].set_ylabel('Height (cm)')
axs[2].set_xticks(weeks)
axs[2].set_yticks(range(0, 45, 5))
axs[2].grid(True)

# Adjust layout
plt.tight_layout()

# Show the plots
plt.show()
```

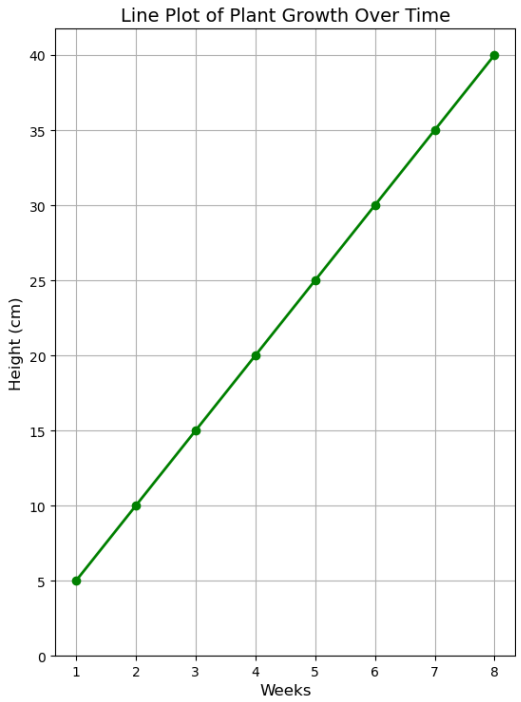

Line Plot of Plant Growth Over Time
Height (cm)
Weeks
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
8

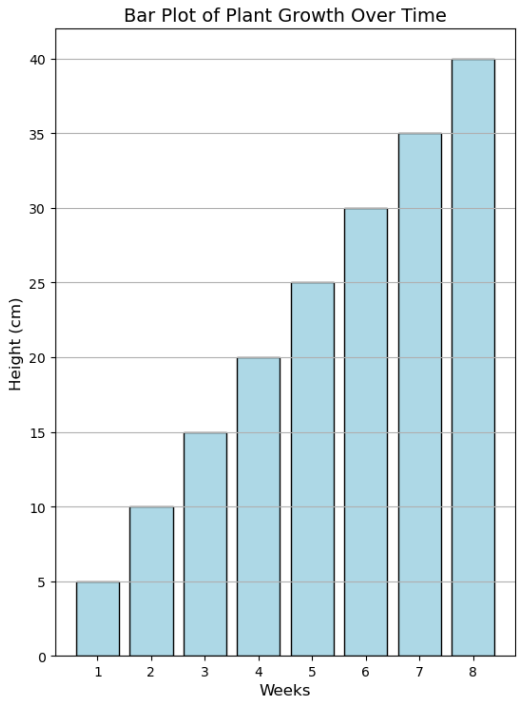

Bar Plot of Plant Growth Over Time
Height (cm)
Weeks
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
8

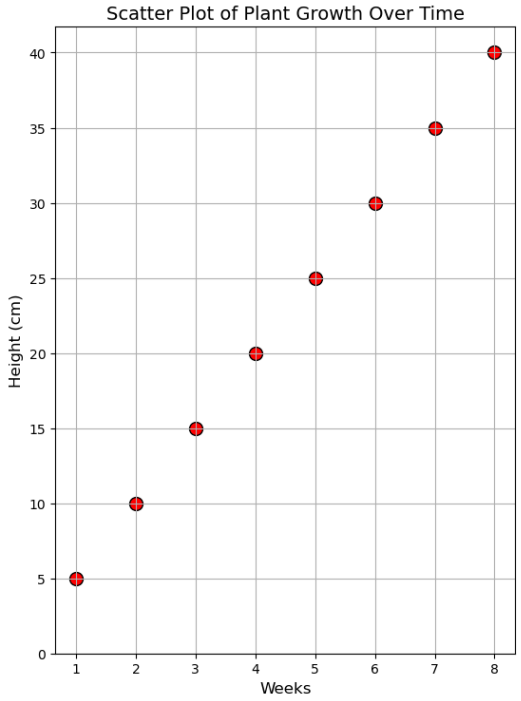

Scatter Plot of Plant Growth Over Time
Height (cm)
Weeks
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
8

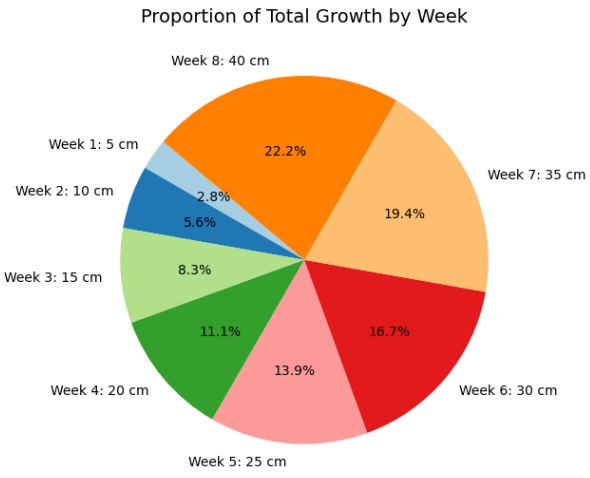
[6]:
```
import matplotlib.pyplot as plt

# Sample data: Weeks and corresponding heights of a plant
weeks = [1, 2, 3, 4, 5, 6, 7, 8]
heights = [5, 10, 15, 20, 25, 30, 35, 40]  # Plant height in cm

# Create a figure with multiple subplots
fig, axs = plt.subplots(2, 2, figsize=(12, 15))

# Line Plot
axs[0, 0].plot(weeks, heights, marker='o', linestyle='-', color='green',␣
 ↪linewidth=2)
axs[0, 0].set_title('Line Plot of Plant Growth Over Time', fontsize=14)
axs[0, 0].set_xlabel('Weeks', fontsize=12)
axs[0, 0].set_ylabel('Height (cm)', fontsize=12)
axs[0, 0].set_xticks(weeks)
axs[0, 0].set_yticks(range(0, 45, 5))
axs[0, 0].grid(True)

# Bar Plot
axs[0, 1].bar(weeks, heights, color='lightblue', edgecolor='black')
axs[0, 1].set_title('Bar Plot of Plant Growth Over Time', fontsize=14)
axs[0, 1].set_xlabel('Weeks', fontsize=12)
axs[0, 1].set_ylabel('Height (cm)', fontsize=12)
axs[0, 1].set_xticks(weeks)
axs[0, 1].set_yticks(range(0, 45, 5))
axs[0, 1].grid(axis='y')

# Scatter Plot
axs[1, 0].scatter(weeks, heights, color='red', s=100, edgecolor='black')
axs[1, 0].set_title('Scatter Plot of Plant Growth Over Time', fontsize=14)
axs[1, 0].set_xlabel('Weeks', fontsize=12)
axs[1, 0].set_ylabel('Height (cm)', fontsize=12)
axs[1, 0].set_xticks(weeks)
axs[1, 0].set_yticks(range(0, 45, 5))
axs[1, 0].grid(True)

# Pie Chart
total_growth = sum(heights)
growth_sizes = [(height / total_growth) * 100 for height in heights]
labels = [f'Week {week}: {height} cm' for week, height in zip(weeks, heights)]

axs[1, 1].pie(growth_sizes, labels=labels, autopct='%1.1f%%', startangle=140,␣
 ↪colors=plt.cm.Paired.colors)
```

```
axs[1, 1].set_title('Proportion of Total Growth by Week', fontsize=14)

# Adjust layout
plt.tight_layout()

# Show the plots
plt.show()
```



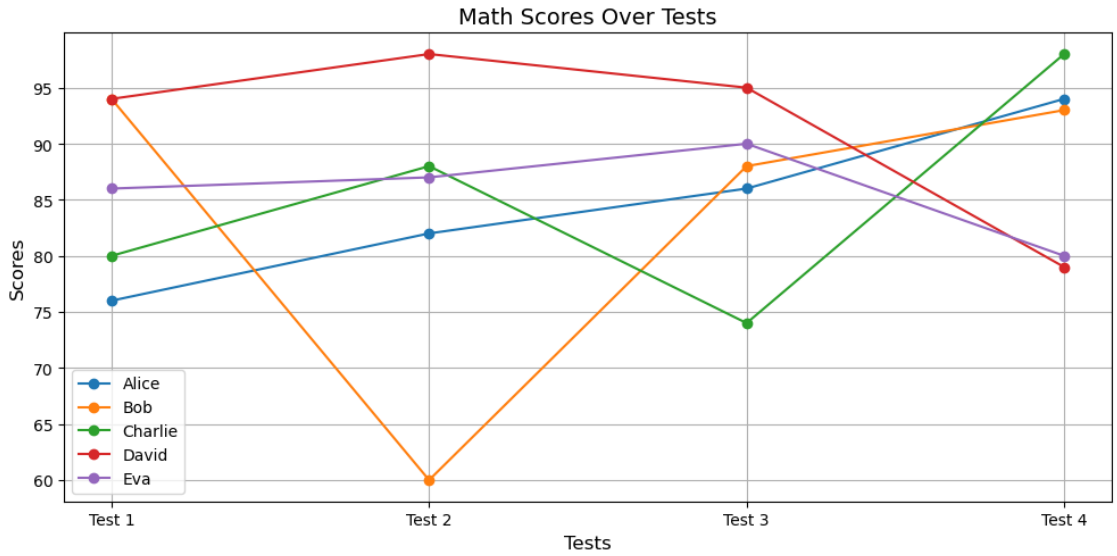


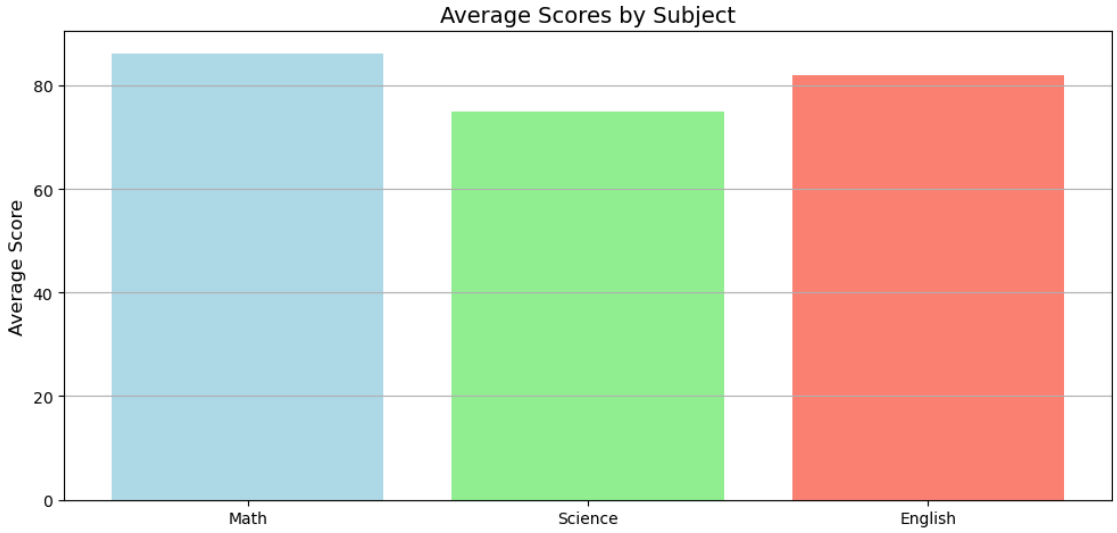


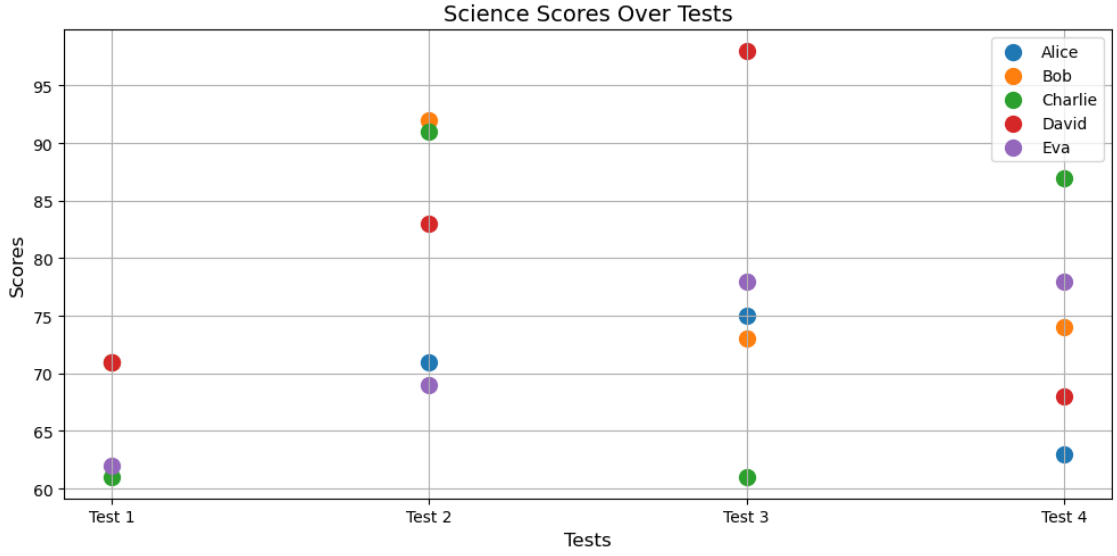


[7]:
```
import matplotlib.pyplot as plt
import numpy as np

# Sample data
students = ['Alice', 'Bob', 'Charlie', 'David', 'Eva']
tests = ['Test 1', 'Test 2', 'Test 3', 'Test 4']
math_scores = np.random.randint(60, 100, size=(5, 4))
science_scores = np.random.randint(60, 100, size=(5, 4))
english_scores = np.random.randint(60, 100, size=(5, 4))

# Create a figure with subplots
fig, axs = plt.subplots(3, 1, figsize=(10, 15))

# Line Plot for Math Scores
for i, student in enumerate(students):
    axs[0].plot(tests, math_scores[i], marker='o', label=student)
axs[0].set_title('Math Scores Over Tests', fontsize=14)
axs[0].set_xlabel('Tests', fontsize=12)
axs[0].set_ylabel('Scores', fontsize=12)
axs[0].legend()
axs[0].grid(True)

# Bar Plot for Average Scores
average_scores = [np.mean(scores) for scores in [math_scores, science_scores,␣
 ↪english_scores]]
subjects = ['Math', 'Science', 'English']
axs[1].bar(subjects, average_scores, color=['lightblue', 'lightgreen',␣
 ↪'salmon'])
axs[1].set_title('Average Scores by Subject', fontsize=14)
axs[1].set_ylabel('Average Score', fontsize=12)
axs[1].grid(axis='y')

# Scatter Plot for Science Scores
for i, student in enumerate(students):
    axs[2].scatter(tests, science_scores[i], s=100, label=student)
axs[2].set_title('Science Scores Over Tests', fontsize=14)
axs[2].set_xlabel('Tests', fontsize=12)
axs[2].set_ylabel('Scores', fontsize=12)
axs[2].legend()
axs[2].grid(True)

# Adjust layout
plt.tight_layout()
```

```
# Show the plots
plt.show()
```


Math Scores Over Tests
Scores
Tests
Test 1
Test 2
Test 3
Test 4
Alice
Bob
Charlie
David
Eva


Average Scores by Subject
Average Score
Math
Science
English


Science Scores Over Tests
Scores
Tests
Test 1
Test 2
Test 3
Test 4
Alice
Bob
Charlie
David
Eva

[ ]: