

# mod4\_assignment-uc3

October 21, 2024

## 1 Module4: Numpy, Pandas, Matplotlib

### 1.1 Assignment: Use-Case III

```
[54]: #1
import pandas as pd
from matplotlib import pyplot as plt

# Load the data
data = pd.read_csv('../dataset/Hurricanes.csv')

# Print basic information about the data (Optional)
print('DATA INFO: ')
print(data.info())
print('\nDATA HEAD: ')
print(data.head())

# Plotting the bar graph
plt.figure(figsize=(12, 5)) # Adjusted the figure size for better clarity

# Bar plot for the number of hurricanes per year
plt.bar(data['Year'], data['Hurricanes'], width=0.6, color="blue")

# Adding labels and title
plt.xlabel('Year')
plt.ylabel('Number of Hurricanes')
plt.title('Number of Hurricanes per Year (US Atlantic Coast)')

# Set x-ticks at intervals of 10 years for better readability
plt.xticks(ticks=range(data['Year'].min(), data['Year'].max() + 1, 10))

# Add a grid for easier interpretation
plt.grid(axis='y', linestyle='--', alpha=0.7)

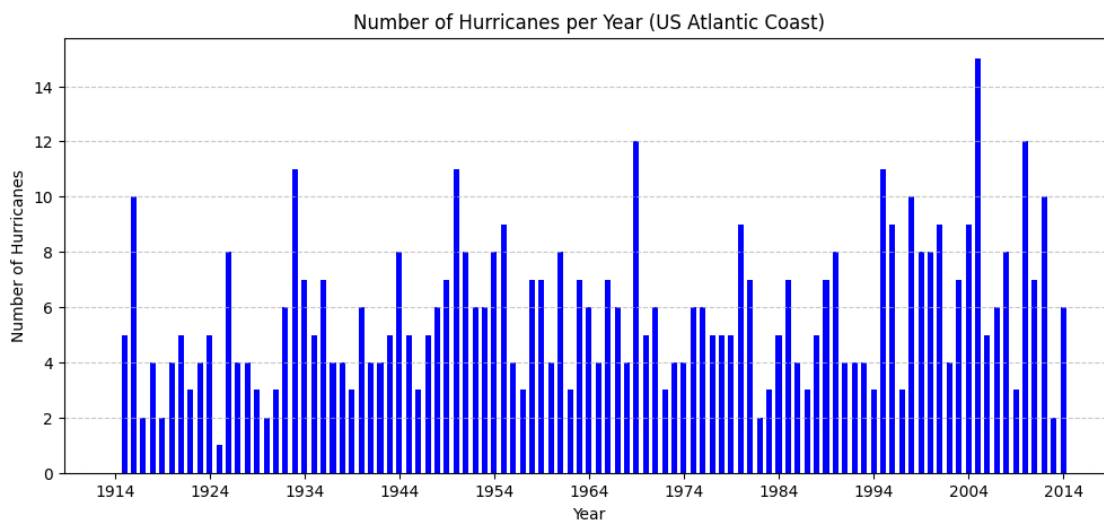
# Display the plot
plt.show()
```

DATA INFO:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Year        101 non-null    int64
1   Hurricanes  101 non-null    int64
dtypes: int64(2)
memory usage: 1.7 KB
None
```

DATA HEAD:

	Year	Hurricanes
0	1914	0
1	1915	5
2	1916	10
3	1917	2
4	1918	4



```
[57]: #2
import pandas as pd
from matplotlib import pyplot as plt

# Load the dataset
temp_data = pd.read_csv('../dataset/CityTemps.csv')

# Checking data information and the first few rows
print(temp_data.info()) # No null or empty values
print(temp_data.head())
```

```

# Bins for temperature ranges
bins = [-10, -5, 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]

# Create a figure for the histogram
plt.figure(figsize=(8,6))

# Plot histogram for Moscow
plt.hist(temp_data['Moscow'], bins, rwidth=0.8, color="blue", alpha=0.7,
        label='Moscow')

# Plot histogram for San Francisco
plt.hist(temp_data['San Francisco'], bins, rwidth=0.8, color="red", alpha=0.7,
        label='San Francisco')

# Adding labels and title
plt.xlabel('Temperature (°C)')
plt.ylabel('Frequency')
plt.title('Temperature Distribution in Moscow and San Francisco (2014-2015)')
plt.legend()

# Show the plot
plt.show()

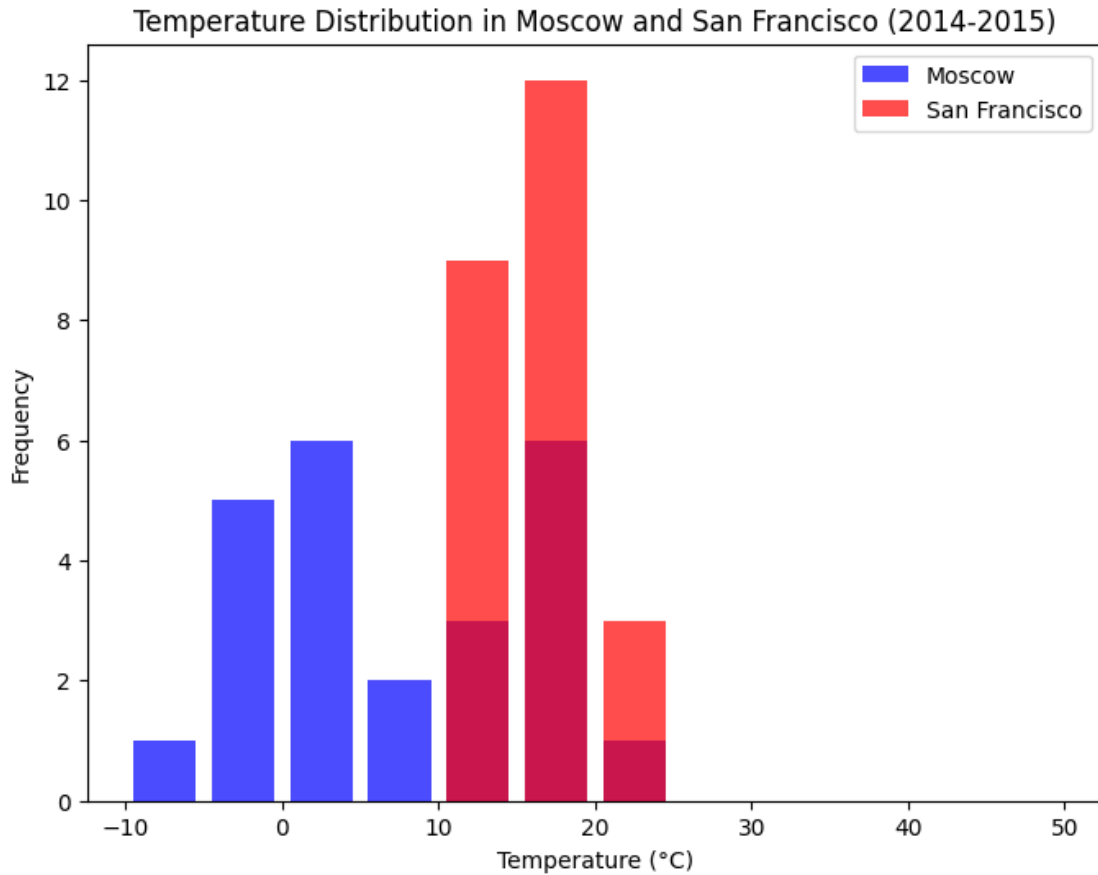
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year            24 non-null    int64
1   Month           24 non-null    int64
2   Moscow          24 non-null    float64
3   Melbourne       24 non-null    float64
4   San Francisco   24 non-null    float64
dtypes: float64(3), int64(2)
memory usage: 1.1 KB
None

```

	Year	Month	Moscow	Melbourne	San Francisco
0	2014	1	-8.6	22.0	12.8
1	2014	2	-1.9	21.5	12.7
2	2014	3	2.8	19.5	15.0
3	2014	4	7.0	15.8	15.7
4	2014	5	16.0	13.9	17.5



```
[62]: #3
temp_data.to_csv('M4_assign_dataset.csv', index=False)
re_data=pd.read_csv( 'M4_assign_dataset.csv')
print(re_data.info())
print(re_data.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year            24 non-null    int64
1   Month           24 non-null    int64
2   Moscow          24 non-null    float64
3   Melbourne       24 non-null    float64
4   San Francisco   24 non-null    float64
dtypes: float64(3), int64(2)
memory usage: 1.1 KB
None
   Year  Month  Moscow  Melbourne  San Francisco
```

0	2014	1	-8.6	22.0	12.8
1	2014	2	-1.9	21.5	12.7
2	2014	3	2.8	19.5	15.0
3	2014	4	7.0	15.8	15.7
4	2014	5	16.0	13.9	17.5

```
[78]: import pandas as pd
from matplotlib import pyplot as plt
import numpy as np

#4. Let the x-axis data points and y-axis data points are
x = [1,2,3,4]
y = [20, 21, 20.5, 20.8]
#5.1: Draw a Simple plot
plt.plot(x,y)

#5.2: Configure the line and markers in a simple plot
plt.plot(x,y, marker='^', linestyle='dashed')
#5.3: configure the axes
# Configure the axes using min and max from the data
plt.xlim(min(x) - 1, max(x) + 1) # Set x-axis limits slightly beyond min and
    ↪max values
plt.ylim(min(y) - 1, max(y) + 1) # Set y-axis limits slightly beyond min and
    ↪max values
# Optionally, add grid lines
plt.grid(True)
#5.4: Give the title of Graph & labels of the x axis and y axis
plt.xlabel('x')
plt.ylabel('y')
plt.title('y=f(x)')
plt.show()

#5.5: Give error bar if y_error = [0.12, 0.13, 0.2, 0.1]
y_error = [0.12, 0.13, 0.2, 0.1]

#5.6: define width, and height as figsize=(4,5) DPI and adjust plot dpi=100
plt.figure(figsize=(4, 5), dpi=100)

#5.7: Give a font size of 14
plt.xlabel('x', fontsize=14)
plt.ylabel('y', fontsize=14)
plt.title('y = f(x) with Error Bars', fontsize=14)

# 5.5: Add error bars
plt.errorbar(x, y, yerr=y_error, marker='^', linestyle='dashed', capsize=5,
    ↪label='y=f(x)')
```

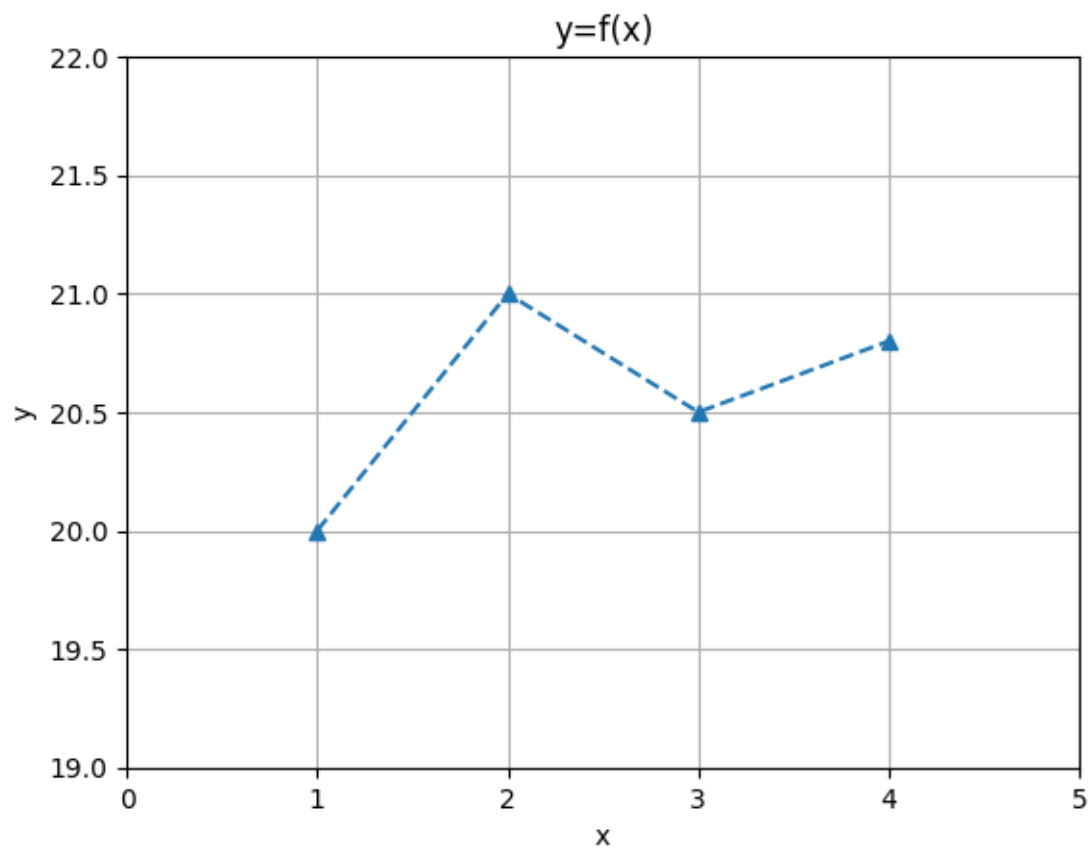
```

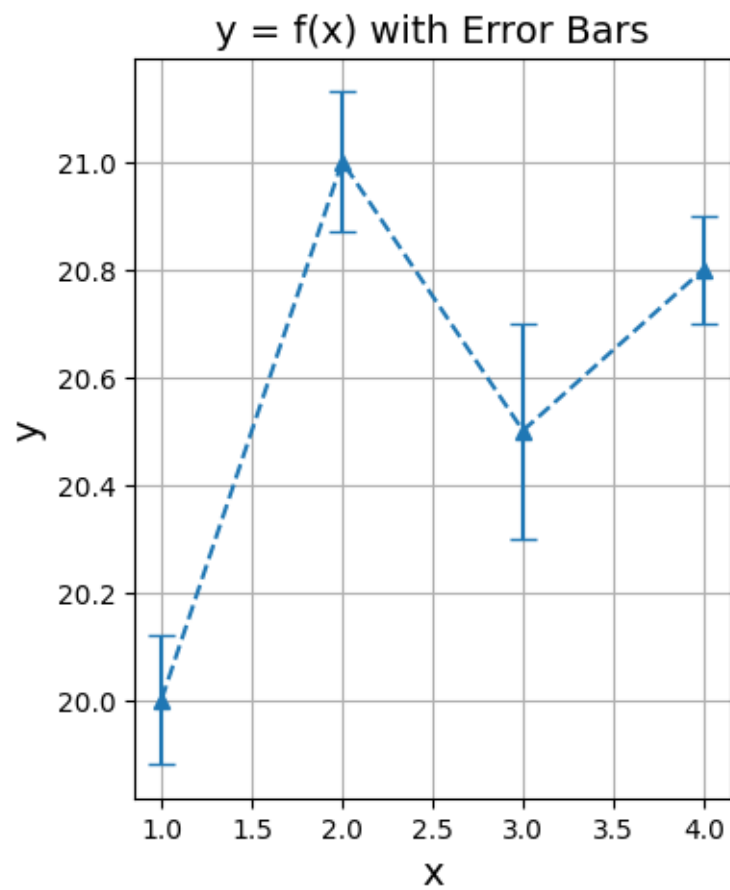
plt.grid(True)
plt.show()

#5.8: Draw a scatter graph of any 50 random values of the x and y axis
np.random.seed(0) # For reproducibility
random_x = np.random.rand(50) * 100 # Generate 50 random x values
random_y = np.random.rand(50) * 100 # Generate 50 random y values
plt.figure(figsize=(4, 5), dpi=100)
plt.scatter(random_x, random_y, color='blue', label='Random Scatter')
plt.xlabel('Random X', fontsize=14)
plt.ylabel('Random Y', fontsize=14)
plt.title('Random Scatter Plot', fontsize=14)
plt.grid(True)
plt.show()

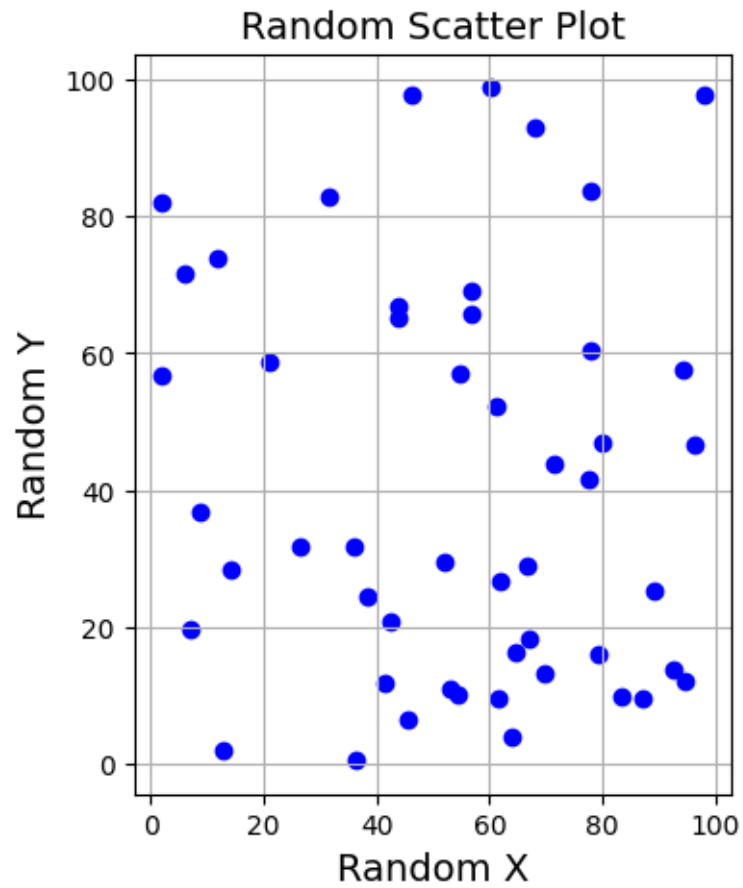
#5.9: Create a dataframe from the following data
dict_data = {'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
             'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
             'female': [0, 1, 1, 0, 1],
             'age': [42, 52, 36, 24, 73],
             'preTestScore': [4, 24, 31, 2, 3],
             'postTestScore': [25, 94, 57, 62, 70] }
df = pd.DataFrame(dict_data)
df

```









```
[78]: first_name last_name female age preTestScore postTestScore
0      Jason    Miller      0  42           4          25
1     Molly  Jacobson      1  52          24          94
2      Tina      Ali       1  36          31          57
3      Jake    Milner      0  24           2          62
4      Amy     Cooze       1  73           3          70
```