Lovely Professional University

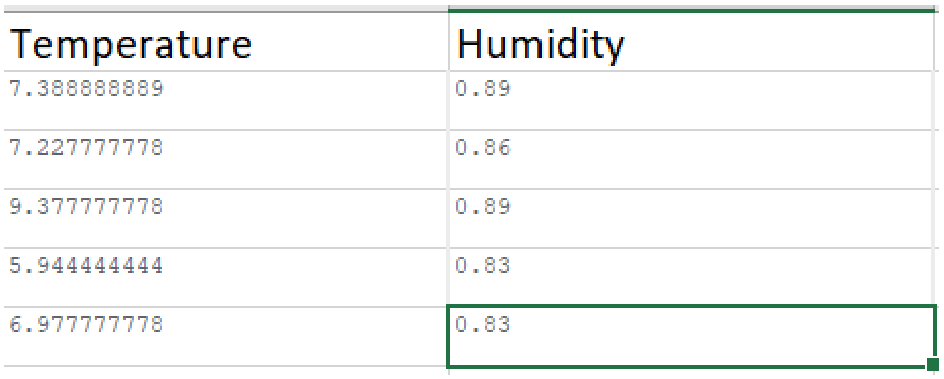
CA2

School: Mittal School of Business **Lovely Faculty of Business and Arts**
Name of the Faculty Member: Logesh Kumar, Rupesh
Course Code: MGNM801 **Course Title:** Business Analytics-I
Section: Q2240
Date of Allotment: 22/12/2022 **Date of Submission:** 29/12/2022
Attempt (Group/Individual): Individual **Max Marks** :30
Name: Nowneesh T **Roll No:** RQ2240A19
Reg No: 12202342

S No.	Objectives	Task	Marks	Evaluation Parameters
1	Test the understanding the concepts of Data Analysis using Pandas	Explain Pandas, assume you have a dataset with two columns, "Temperature" and "Humidity" in a csv file named "weather_data.csv". How would you perform the following operations on the dataset with pandas: 1. Read data 2. Separate the two columns and transform them into Pandas Series 3. Explain DataFrame() and Series() Function	10	Capability to solve problems – 3 marks Ability to write and describe code – 6 marks Ability to understand the problem – 1 mark
2	Test Data Visualization Capability of the students	What is Matplotlib? How would you perform the following actions in Matplotlib: 1. Make a simple line graph, scatter plot, bar chart and a histogram 2. Add Labels to each of the above charts 3. Save the graphs above using python.	10	Research of the topic – 3 marks Ability to write clean and robust code – 5 marks Ability to understand problem – 2 marks
3	Improve student's ability to understand Data Visualization concepts more effectively	Explain plotly? Create any three graph with the help of plotly.	10	Ability to write the code – 6 marks Ability to customise the graph – 3 marks Ability to understand problem – 1 mark

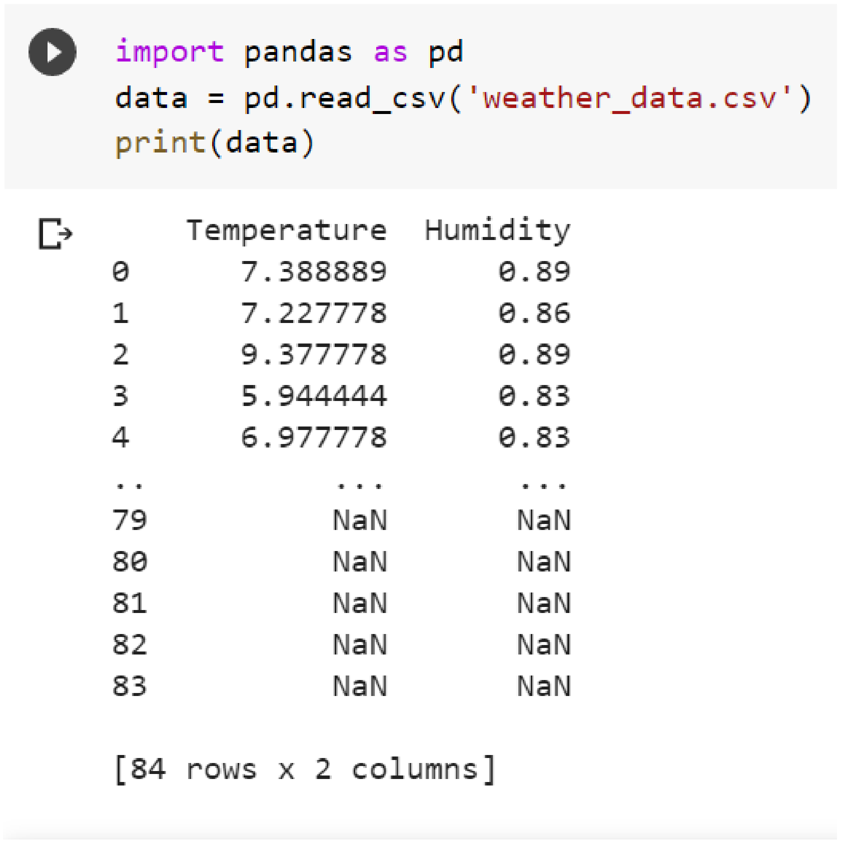
1.Pandas

The data is taken from Kaggle with 5 rows.



Temperature	Humidity
7.388888889	0.89
7.227777778	0.86
9.377777778	0.89
5.944444444	0.83
6.977777778	0.83

Reading the data



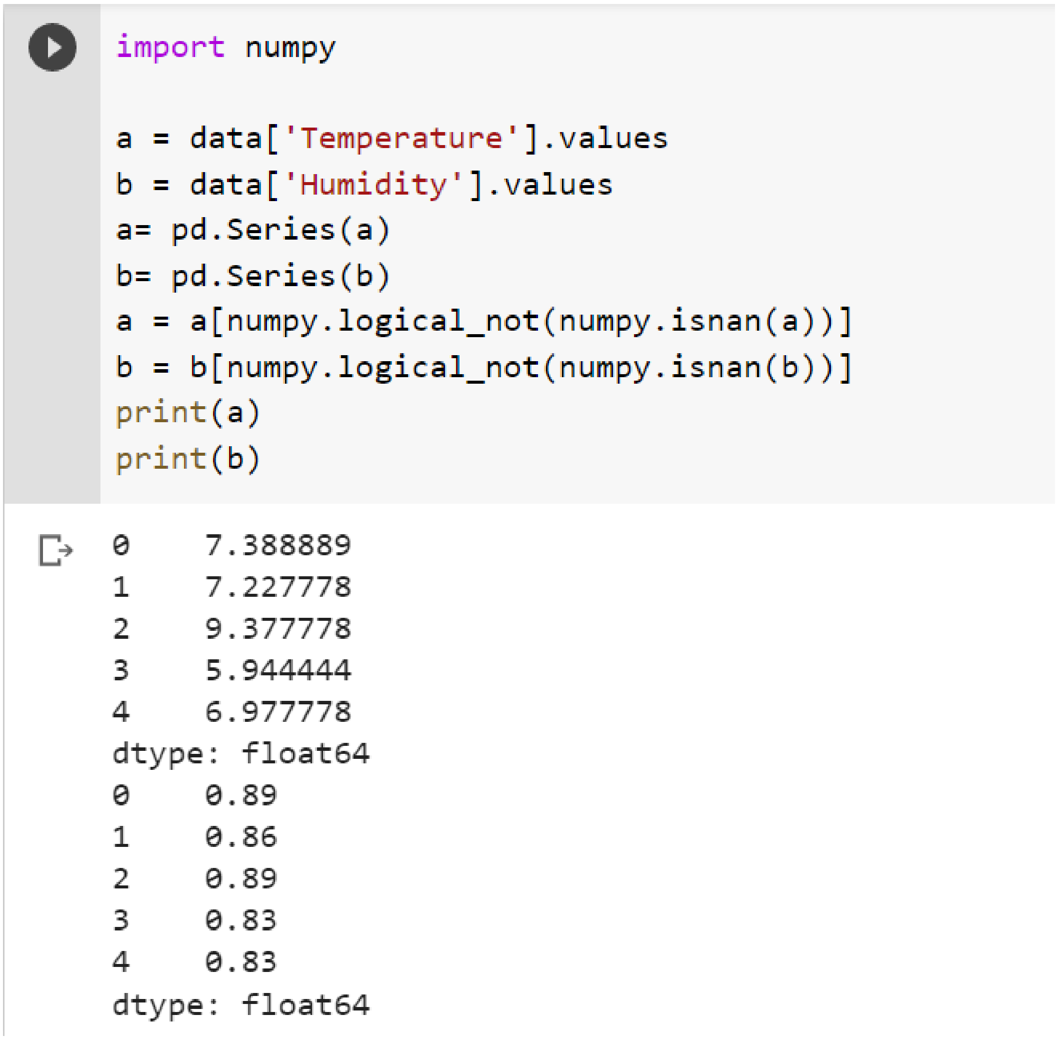
```python
import pandas as pd
data = pd.read_csv('weather_data.csv')
print(data)
```

```
    Temperature  Humidity
0      7.388889      0.89
1      7.227778      0.86
2      9.377778      0.89
3      5.944444      0.83
4      6.977778      0.83
..          ...       ...
79          NaN       NaN
80          NaN       NaN
81          NaN       NaN
82          NaN       NaN
83          NaN       NaN

[84 rows x 2 columns]
```

- First, we have to import pandas function.
- Second, we have to read the data from CSV file
- Third, we have to print the data for checking whether data is successfully read or not.

Columns into pandas Series



```
import numpy

a = data['Temperature'].values
b = data['Humidity'].values
a= pd.Series(a)
b= pd.Series(b)
a = a[numpy.logical_not(numpy.isnan(a))]
b = b[numpy.logical_not(numpy.isnan(b))]
print(a)
print(b)
```

```
0    7.388889
1    7.227778
2    9.377778
3    5.944444
4    6.977778
dtype: float64
0    0.89
1    0.86
2    0.89
3    0.83
4    0.83
dtype: float64
```

- After reading reading data, we have to separate those data.
- “Series” command is used to split the columns from data variable.
- “is nan” is used to delete unwanted or unfilled data.
- Atlast, we have to print the data.

A Pandas Series resembles a table's column.

- It is a one-dimensional array that can hold any kind of data.
- You are able to name your own labels using the index option.
- When you create labels, you can use the label to get to an item.
- When constructing a Series, you can also utilise a key/value object like a dictionary.

Create a simple Pandas Series from a list:

```
import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a)

print(myvar)
```

```
0    1
1    7
2    2
dtype: int64
```

A Pandas DataFrame is a two-dimensional data structure having rows and columns, similar to a two-dimensional array.

- The loc attribute is used by Pandas to return one or more specified rows (s)
- You are able to name your own indexes using the index argument.
- Pandas may load data sets that are kept in files into a DataFrame.

Create a simple Pandas DataFrame:

```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

```
   calories  duration
0       420        50
1       380        40
2       390        45
```

2. MatplotLib

A tool for visualising data, Matplotlib is a low level graph charting framework written in Python.
Since Matplotlib is open source, we are allowed to utilise it.
For platform compatibility, Matplotlib is primarily written in Python, with a small amount of code written in C, Objective-C, and Javascript.

LINE CHART

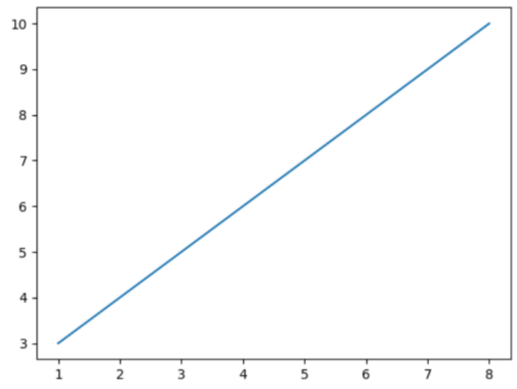
```
#Three lines to make our compiler able to draw:
import sys
import matplotlib
matplotlib.use('Agg')

import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints)
plt.show()

#Two  lines to make our compiler able to draw:
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```

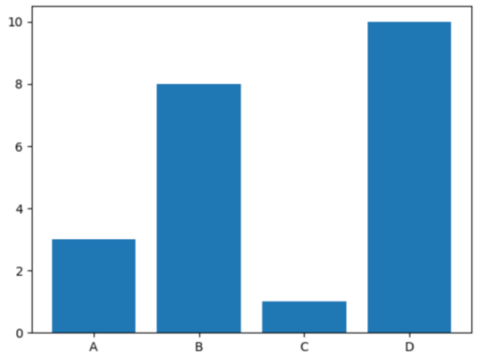


BAR CHART

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```



HISTOGRAM

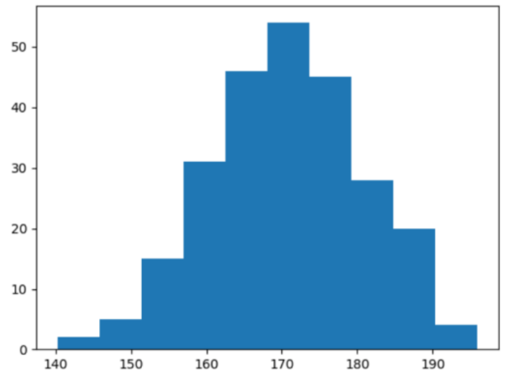
```
import sys
import matplotlib
matplotlib.use('Agg')

import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)

plt.hist(x)
plt.show()

#Two  lines to make our compiler able to draw:
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```

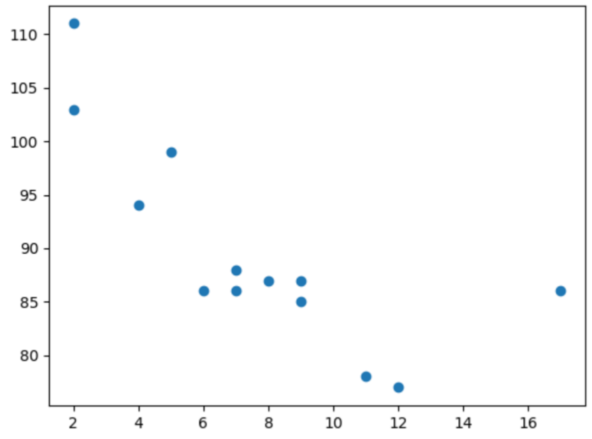


SCATTER PLOT

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)
plt.show()
```



3. Plotly

An open-source library called the Python Plotly Library can be used to quickly and easily visualise data and comprehend it. Plotly supports a number of different plot types, including line charts, scatter plots, histograms, and cox plots.

```
import plotly.express as px


# Creating the Figure instance
fig = px.line(x=[1,2, 3], y=[1, 2, 3])

# printing the figure instance
print(fig)
```

```
Figure({
    'data': [{'hovertemplate': 'x=%{x}<br>y=%{y}<extra></extra>',
              'legendgroup': '',
              'line': {'color': '#636efa', 'dash': 'solid'},
              'mode': 'lines',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([1, 2, 3]),
              'xaxis': 'x',
              'y': array([1, 2, 3]),
              'yaxis': 'y'}],
    'layout': {'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'x'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'y'}}}
})
```

```
import plotly.express as px


# Creating the Figure instance
fig = px.line(x=[1, 2, 3], y=[1, 2, 3])

# showing the plot
fig.show()
```

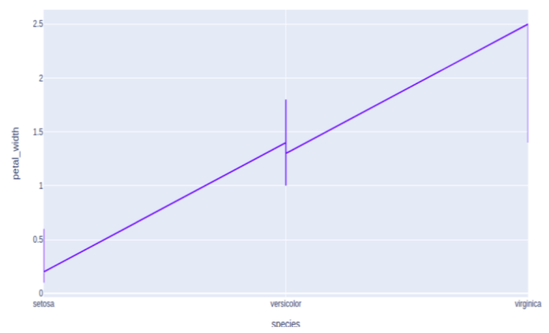
LINE CHART

```python
import plotly.express as px

# using the iris dataset
df = px.data.iris()

# plotting the line chart
fig = px.line(df, x="species", y="petal_width")

# showing the plot
fig.show()
```



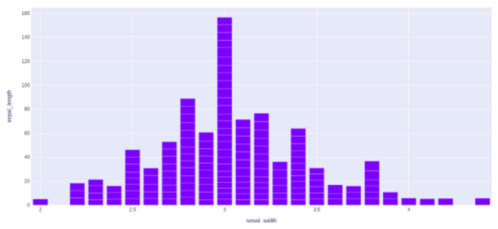
BAR CHART

```python
import plotly.express as px

# using the iris dataset
df = px.data.iris()

# plotting the bar chart
fig = px.bar(df, x="sepal_width", y="sepal_length")

# showing the plot
fig.show()
```



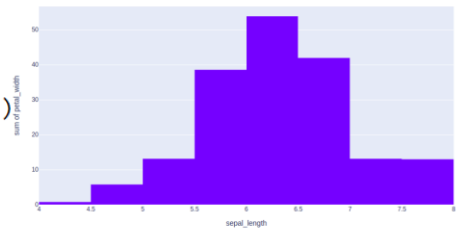
HISTOGRAM

```python
import plotly.express as px

# using the iris dataset
df = px.data.iris()

# plotting the histogram
fig = px.histogram(df, x="sepal_length", y="petal_width")

# showing the plot
fig.show()
```


sum of petal_width
sepal_length