EXPERIMENT - 3

CREATING DATAFRAMES & ANALYZING PROPERTIES

AIM:
To create Pandas DataFrames using NumPy arrays, CSV/Excel files and analyze the various properties of the DataFrames for real world structured data analysis.

PREREQUISITES & REQUIREMENTS:

1. Computer with Python Installed
2. Jupytor Notebook
3. Knowledge on Python, Numpy & Pandas Libraries

Step 1: Install and Import the Libraries using commands:

pip install numpy
pip install pandas

In [172…
```
# importing NumPy & Pandas
import numpy as np
import pandas as pd
```

Step 2: Create a 2-D array

In [173…
```
# Considering a 4 weeks bike sales data in three different branches
# rows --> branches, columns --> weeks

sales = np.array([[150, 200, 180, 200],
                  [160, 210, 160, 230],
                  [170, 220, 200, 240]])
```

In [174…
```
# sales array properties
sales.shape, sales.ndim
```

Out[174…
```
((3, 4), 2)
```

Step 3: Converting the created 2-D array into DataFrame using Pandas

In [175…
```
df = pd.DataFrame(sales)
df
```

Out[175…

	0	1	2	3
0	150	200	180	200
1	160	210	160	230
2	170	220	200	240

In [176…
```
# Creating the Index & Columns for the DataFrame
# Index --> Branches, Columns --> Weeks

df_index = ['Branch_1', 'Branch_2', 'Branch_3']
df_columns = ['Week_1', 'Week_2', 'Week_3', 'Week_4']
```

In [177…
```
# Assigning the index & columns to the dataframe
df = pd.DataFrame(data=sales, index=df_index, columns=df_columns)
```

In [178…
```
# Created DataFrame
df
```

Out[178…

	Week_1	Week_2	Week_3	Week_4
Branch_1	150	200	180	200
Branch_2	160	210	160	230
Branch_3	170	220	200	240

In [179…

```python
# Adding Week_5 Data (Column)
# Must match number of rows (3 branches)
df['Week_5'] = [210, 220, 230]
df
```

Out[179…

	Week_1	Week_2	Week_3	Week_4	Week_5
Branch_1	150	200	180	200	210
Branch_2	160	210	160	230	220
Branch_3	170	220	200	240	230

In [181…

```python
# Add Branch_4 (Row)

branch_4 = pd.DataFrame({
    'Week_1': [155],
    'Week_2': [205],
    'Week_3': [175],
    'Week_4': [210],
    'Week_5': [225]
}, index=['Branch_4'])

df = pd.concat([df, branch_4])
df
```

Out[181…

	Week_1	Week_2	Week_3	Week_4	Week_5
Branch_1	150	200	180	200	210
Branch_2	160	210	160	230	220
Branch_3	170	220	200	240	230
Branch_4	155	205	175	210	225

Step 4: Creating a DataFrame with Existing CSV/EXCEL File

In [182…

```python
# Incase, if file is not in the same folder, full path is necessary
# use read_csv method

stu_df = pd.read_csv('students_data.csv')
stu_df
```

Out[182…

	Name	Age	City	Math Marks	Science Marks	English Marks	History Marks	Geography Marks	Computer Marks	Attendance	Hobbies
0	Naina	23	Lucknow	88	88	65	75	73	75	99.811461	Reading
1	Vivaan	18	Ahmedabad	96	84	90	74	88	97	99.597535	Sports
2	Aadya	24	Bangalore	67	97	60	86	69	83	99.573261	Music
3	Kriti	21	Jaipur	90	90	78	71	61	97	99.448342	Sports
4	Gautam	23	Kolkata	72	80	72	73	97	90	99.339785	Music
...	...	...	...	...	...	...	...	...	...	...	...
295	Pihu	20	Pune	79	95	90	88	81	63	70.507363	Reading
296	Grisha	18	Mumbai	67	76	74	70	75	94	70.303313	Traveling
297	Sana	24	Delhi	84	64	80	93	97	84	70.235032	Reading
298	Prisha	24	Pune	84	89	99	60	88	88	70.204657	Music
299	Aadrika	24	Mumbai	63	95	66	90	81	100	70.173928	Sports

300 rows × 11 columns

Step 5: Analyze the Important Properties of the DataFrames

In [183…

```
# To view first 5 data points
stu_df.head()
```

Out[183…

	Name	Age	City	Math Marks	Science Marks	English Marks	History Marks	Geography Marks	Computer Marks	Attendance	Hobbies
0	Naina	23	Lucknow	88	88	65	75	73	75	99.811461	Reading
1	Vivaan	18	Ahmedabad	96	84	90	74	88	97	99.597535	Sports
2	Aadya	24	Bangalore	67	97	60	86	69	83	99.573261	Music
3	Kriti	21	Jaipur	90	90	78	71	61	97	99.448342	Sports
4	Gautam	23	Kolkata	72	80	72	73	97	90	99.339785	Music

In [184…

```
# Shape of the DataFrame
stu_df.shape
```

Out[184…

```
(300, 11)
```

In [185…

```
# Index Range of the DataFrame
stu_df.index
```

Out[185…

```
RangeIndex(start=0, stop=300, step=1)
```

In [187…

```
# Data Types in the DataFrame
stu_df.dtypes
```

Out[187…

```
Name                 object
Age                   int64
City                 object
Math Marks            int64
Science Marks         int64
English Marks         int64
History Marks         int64
Geography Marks       int64
Computer Marks        int64
Attendance          float64
Hobbies              object
dtype: object
```

In [188…

```
# Retrieve the columns (feature) names of the DataFrames
stu_df.columns
```

Out[188…

```
Index(['Name', 'Age', 'City', 'Math Marks', 'Science Marks', 'English Marks',
       'History Marks', 'Geography Marks', 'Computer Marks', 'Attendance',
       'Hobbies'],
      dtype='object')
```

In [189…

```
# Information about the DataFrame
stu_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Name             300 non-null    object 
 1   Age              300 non-null    int64  
 2   City             300 non-null    object 
 3   Math Marks       300 non-null    int64  
 4   Science Marks    300 non-null    int64  
 5   English Marks    300 non-null    int64  
 6   History Marks    300 non-null    int64  
 7   Geography Marks  300 non-null    int64  
 8   Computer Marks   300 non-null    int64  
 9   Attendance       300 non-null    float64
 10  Hobbies          300 non-null    object 
dtypes: float64(1), int64(7), object(3)
memory usage: 25.9+ KB
```

In [190…

```
# Collecting Descriptive statistics of numerical data in a DataFrame.
# It provides summary statistics such as count, mean, standard deviation, Min etc..
```

```
stu_df.describe()
```

Out[190…

	Age	Math Marks	Science Marks	English Marks	History Marks	Geography Marks	Computer Marks	Attendance
count	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000
mean	21.450000	81.353333	81.926667	80.290000	80.480000	79.820000	80.156667	85.253308
std	2.242975	11.860187	12.085037	12.316312	11.827195	12.308232	12.087977	8.947083
min	18.000000	60.000000	60.000000	60.000000	60.000000	60.000000	60.000000	70.173928
25%	20.000000	71.000000	72.000000	69.000000	70.000000	69.000000	69.000000	77.488971
50%	21.500000	81.000000	84.000000	81.000000	81.000000	79.500000	80.000000	85.249150
75%	23.000000	92.000000	93.000000	91.000000	90.000000	91.250000	90.000000	93.351863
max	25.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	99.811461

In [191…

```
# Transposing Rows & Columns
stu_df.head(10).transpose()
```

Out[191…

	0	1	2	3	4	5	6	7	8	
Name	Naina	Vivaan	Aadya	Kriti	Gautam	Yash	Dhruv	Niharika	Dev	Is
Age	23	18	24	21	23	25	25	23	23	
City	Lucknow	Ahmedabad	Bangalore	Jaipur	Kolkata	Chennai	Delhi	Kolkata	Lucknow	Koll
Math Marks	88	96	67	90	72	98	77	93	83	
Science Marks	88	84	97	90	80	87	88	95	81	
English Marks	65	90	60	78	72	82	71	93	91	
History Marks	75	74	86	71	73	63	71	71	83	
Geography Marks	73	88	69	61	97	71	74	80	64	
Computer Marks	75	97	83	97	90	86	79	62	69	
Attendance	99.811461	99.597535	99.573261	99.448342	99.339785	99.262248	99.122242	98.93384	98.782453	98.772
Hobbies	Reading	Sports	Music	Sports	Music	Music	Sports	Sports	Music	Reac

RESULT:
By using Pandas, DataFrames were successfully created from NumPy arrays and CSV files. Various DataFrame properties such as shape, column names, index, data types and statistical summaries were analyzed.