PRACTICAL-1

Objective- Study and document the different phases of a data analytics project (Data Collection, Cleaning, Processing, Analysis, and Visualization).

Phases of a Data Analytics Project A data analytics project typically involves several interconnected phases: Data Collection, Cleaning, Processing, Anaysis, and Visualization. Each phase plays a critical role in transforming raw data into actionable insights. Below is a detailed study of these phases:

1. **Data Collection :-** This phase involves gathering data from various sources to address specific questions or objectives. Steps: Identify the questions or objectives the data needs to answer. Choose appropriate data collection methods, such as surveys, interviews, observations, or secondary sources like government reports. Determine the required amount of data and the sampling method (random, systematic, stratified). Ensure the data source is trustworthy and reliable. Tools: Online forms, APIs, databases, sensors, and manual collection methods.

2. **Data Cleaning :-** Data cleaning ensures the accuracy, consistency, and reliability of the dataset by removing errors and irrelevant information. Steps: Detect and correct inaccuracies such as missing values or duplicates. Fix structural errors (e.g., inconsistent formatting). Filter outliers and irrelevant observations. Automate repetitive cleaning tasks using scripts or tools. Importance: Clean data improves analysis accuracy, efficiency, and reliability. Tools: Python libraries (Pandas), R scripts, Excel.

3. **Data Processing : -** This phase transforms raw data into a usable format for analysis. Steps: Input cleaned data into systems like CRMs or databases. Apply algorithms to process the data based on its intended use (e.g., customer insights, trend analysis). Translate processed data into readable formats like tables or graphs. Importance: Enables structured interpretation of raw data for further analysis. Tools: SQL databases, cloud platforms (AWS), machine learning frameworks.

4. **Data Analysis :-** Data analysis involves applying statistical or computational techniques to extract insights from processed data. Types of Analysis: Descriptive Analysis: Summarizes historical data. Diagnostic Analysis: Explains why certain events occurred. Predictive Analysis: Forecasts future trends. Prescriptive Analysis: Suggests actions based on predictions. Process: Perform exploratory analysis to understand patterns in the dataset. Validate findings through statistical tests or modeling techniques. Tools: Python (NumPy, SciPy), R, Tableau, Power BI.

5. **Visualization :-** Visualization presents insights in a graphical format for easier interpretation by stakeholders. Methods: Use charts (bar graphs, pie charts), dashboards, and infographics to summarize findings visually. Focus on clarity and relevance to ensure

stakeholders can derive actionable insights quickly. Importance: Enhances decision- making by simplifying complex datasets into digestible formats. Tools: Tableau, Power BI, Matplotlib (Python).

Practical - 2

OBJECTIVE :- Load a dataset, handle missing values, remove duplicates, and normalize/scale data (Data Exploration)

In [39]:
```
import pandas as pd
df=pd.read_csv('nyc_weather.csv')
```

In [13]:
```
df.head()
```

Out[13]:

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedMPH	Pr
0	01-01-16	38	23	52	30.03	10	8.0	
1	01-02-16	36	18	46	30.02	10	7.0	
2	01-03-16	40	21	47	29.86	10	8.0	
3	01-04-16	25	9	44	30.05	10	9.0	
4	01-05-16	20	-3	41	30.57	10	5.0	

In [15]:
```
df.tail()
```

Out[15]:

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedM
26	1/27/2016	41	22	45	30.03	10	
27	1/28/2016	37	20	51	29.90	10	
28	1/29/2016	36	21	50	29.58	10	
29	1/30/2016	34	16	46	30.01	10	
30	1/31/2016	46	28	52	29.90	10	

In [17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   EST                   31 non-null     object
 1   Temperature           31 non-null     int64
 2   DewPoint              31 non-null     int64
 3   Humidity              31 non-null     int64
 4   Sea Level PressureIn  31 non-null     float64
 5   VisibilityMiles       31 non-null     int64
 6   WindSpeedMPH          28 non-null     float64
 7   PrecipitationIn       31 non-null     object
 8   CloudCover            31 non-null     int64
 9   Events                9 non-null      object
 10  WindDirDegrees        31 non-null     int64
dtypes: float64(2), int64(6), object(3)
memory usage: 2.8+ KB
```

In [19]:
```
df.shape
```

Out[19]: (31, 11)

In [21]:
```
df.describe() #statistatical values
```

Out[21]:

	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedMPH	Clo
count	31.000000	31.000000	31.000000	31.000000	31.000000	28.000000	3
mean	34.677419	17.838710	51.677419	29.992903	9.193548	6.892857	
std	7.639315	11.378626	11.634395	0.237237	1.939405	2.871821	
min	20.000000	-3.000000	33.000000	29.520000	1.000000	2.000000	
25%	29.000000	10.000000	44.500000	29.855000	9.000000	5.000000	
50%	35.000000	18.000000	50.000000	30.010000	10.000000	6.500000	
75%	39.500000	23.000000	55.000000	30.140000	10.000000	8.000000	
max	50.000000	46.000000	78.000000	30.570000	10.000000	16.000000	

In [23]:
```
df2=df.dropna() #data cleaning
```

In [25]:
```
#df4=df.fillna("a")
df4=df['WindSpeedMPH'].fillna("a")
```

In [27]:
```
df.columns
```

Out[27]:
```
Index(['EST', 'Temperature', 'DewPoint', 'Humidity', 'Sea Level PressureIn',
       'VisibilityMiles', 'WindSpeedMPH', 'PrecipitationIn', 'CloudCover',
       'Events', 'WindDirDegrees'],
      dtype='object')
```

In [29]:
```
df3=df.drop_duplicates()
```

In [31]:
```
df3.shape
```

Out[31]:
```
(31, 11)
```

In [33]:
```
df4.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 31 entries, 0 to 30
Series name: WindSpeedMPH
Non-Null Count  Dtype
--------------  -----
31 non-null     object
dtypes: object(1)
memory usage: 380.0+ bytes
```

In [35]:
```
df4.replace({'WindSpeedMPH':'ab'})
```

Out[35]:
```
0      8.0
1      7.0
2      8.0
3      9.0
4      5.0
5      4.0
6      2.0
7      4.0
8      8.0
9        a
10       a
11     6.0
12    10.0
13     5.0
14     5.0
15     7.0
16     6.0
17    12.0
18    11.0
19     6.0
20     6.0
21       a
22    16.0
23     6.0
24     3.0
25     7.0
26     7.0
27     5.0
28     8.0
29     7.0
30     5.0
Name: WindSpeedMPH, dtype: object
```

In [53]:
```
w_avg=df['WindSpeedMPH'].mean()
```

In [55]:
```
w_avg
```

Out[55]:
```
6.892857142857143
```

In [57]:
```
df.fillna("b")
```

Out[57]:

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedM
0	01-01-16	38	23	52	30.03	10	
1	01-02-16	36	18	46	30.02	10	
2	01-03-16	40	21	47	29.86	10	
3	01-04-16	25	9	44	30.05	10	
4	01-05-16	20	-3	41	30.57	10	
5	01-06-16	33	4	35	30.50	10	
6	01-07-16	39	11	33	30.28	10	
7	01-08-16	39	29	64	30.20	10	
8	01-09-16	44	38	77	30.16	9	
9	01-10-16	50	46	71	29.59	4	
10	01-11-16	33	8	37	29.92	10	
11	01-12-16	35	15	53	29.85	10	
12	1/13/2016	26	4	42	29.94	10	1
13	1/14/2016	30	12	47	29.95	10	
14	1/15/2016	43	31	62	29.82	9	
15	1/16/2016	47	37	70	29.52	8	
16	1/17/2016	36	23	66	29.78	8	
17	1/18/2016	25	6	53	29.83	9	1
18	1/19/2016	22	3	42	30.03	10	1
19	1/20/2016	32	15	49	30.13	10	
20	1/21/2016	31	11	45	30.15	10	
21	1/22/2016	26	6	41	30.21	9	
22	1/23/2016	26	21	78	29.77	1	1
23	1/24/2016	28	11	53	29.92	8	
24	1/25/2016	34	18	54	30.25	10	
25	1/26/2016	43	29	56	30.03	10	
26	1/27/2016	41	22	45	30.03	10	
27	1/28/2016	37	20	51	29.90	10	
28	1/29/2016	36	21	50	29.58	10	

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedM
29	1/30/2016	34	16	46	30.01	10	
30	1/31/2016	46	28	52	29.90	10	

In [62]:
```
x=df['WindSpeedMPH'].fillna(w_avg)
```

In [64]:
```
df['WindSpeedMPH']=x
```

In [72]:
```
df['WindSpeedMPH']=df['WindSpeedMPH'].fillna(df['WindSpeedMPH'].mean())
```

In [74]:
```
df
```

Out[74]:

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedM
0	01-01-16	38	23	52	30.03	10	8.000(
1	01-02-16	36	18	46	30.02	10	7.000(
2	01-03-16	40	21	47	29.86	10	8.000(
3	01-04-16	25	9	44	30.05	10	9.000(
4	01-05-16	20	-3	41	30.57	10	5.000(
5	01-06-16	33	4	35	30.50	10	4.000(
6	01-07-16	39	11	33	30.28	10	2.000(
7	01-08-16	39	29	64	30.20	10	4.000(
8	01-09-16	44	38	77	30.16	9	8.000(
9	01-10-16	50	46	71	29.59	4	6.8928
10	01-11-16	33	8	37	29.92	10	6.8928
11	01-12-16	35	15	53	29.85	10	6.000(
12	1/13/2016	26	4	42	29.94	10	10.000(
13	1/14/2016	30	12	47	29.95	10	5.000(
14	1/15/2016	43	31	62	29.82	9	5.000(
15	1/16/2016	47	37	70	29.52	8	7.000(
16	1/17/2016	36	23	66	29.78	8	6.000(
17	1/18/2016	25	6	53	29.83	9	12.000(
18	1/19/2016	22	3	42	30.03	10	11.000(
19	1/20/2016	32	15	49	30.13	10	6.000(
20	1/21/2016	31	11	45	30.15	10	6.000(
21	1/22/2016	26	6	41	30.21	9	6.8928
22	1/23/2016	26	21	78	29.77	1	16.000(
23	1/24/2016	28	11	53	29.92	8	6.000(
24	1/25/2016	34	18	54	30.25	10	3.000(
25	1/26/2016	43	29	56	30.03	10	7.000(
26	1/27/2016	41	22	45	30.03	10	7.000(
27	1/28/2016	37	20	51	29.90	10	5.000(
28	1/29/2016	36	21	50	29.58	10	8.000(

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedM
29	1/30/2016	34	16	46	30.01	10	7.0000
30	1/31/2016	46	28	52	29.90	10	5.0000