Practical No-1

Name :- Mete Omkar Navnath

Div :- A

Batch :- A3

Roll No :- 70

Data Wrangling - I

Perform the following operations using Python on any open source dataset (e.g., data.csv)

1.Import all the required Python Libraries.

2.Locate an open source data from the web (e.g. https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).

3.Load the Dataset into pandas data frame.

4.Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.

5.Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.

6.Turn categorical variables into quantitative variables in Python. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.

```
In [9]: import pandas as pd
```

```
In [11]: import numpy as np
```

```
In [13]: #!unzip archive.zip
```

```
In [17]: df = pd.read_csv("C:\\Users\\Omkar\\Desktop\\TE SEM-2\\Practical's\\DSBDA\\Pract
```

```
In [19]: df
```

Out[19]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24.0	Female	50.0	68.90	Quetta	Rejected
1	Waqar	21.0	Female	99.0	60.73	Karachi	NaN
2	Bushra	17.0	Male	89.0	NaN	Islamabad	Accepted
3	Aliya	17.0	Male	55.0	85.29	Karachi	Rejected
4	Bilal	20.0	Male	65.0	61.13	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	19.0	Female	85.0	78.09	Quetta	Accepted
153	Bilal	17.0	Female	81.0	84.40	Islamabad	Rejected
154	Fatima	21.0	Female	98.0	50.86	Multan	Accepted
155	Shoaib	-1.0	Male	91.0	80.12	Quetta	Accepted
156	Maaz	17.0	Male	88.0	86.85	Lahore	Accepted

157 rows × 7 columns

To find Null Values

In [22]:
```
df.isnull()
```

Out[22]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	True
2	False	False	False	False	True	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	True
...	...	...	...	...	...	...	...
152	False	False	False	False	False	False	False
153	False	False	False	False	False	False	False
154	False	False	False	False	False	False	False
155	False	False	False	False	False	False	False
156	False	False	False	False	False	False	False

157 rows × 7 columns

In [24]:
```
df.head(10)
```

Out[24]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24.0	Female	50.0	68.90	Quetta	Rejected
1	Waqar	21.0	Female	99.0	60.73	Karachi	NaN
2	Bushra	17.0	Male	89.0	NaN	Islamabad	Accepted
3	Aliya	17.0	Male	55.0	85.29	Karachi	Rejected
4	Bilal	20.0	Male	65.0	61.13	Lahore	NaN
5	Murtaza	23.0	Female	NaN	NaN	Islamabad	Accepted
6	Asad	18.0	Male	NaN	97.31	Multan	Accepted
7	Rabia	20.0	Female	82.0	55.67	Lahore	Accepted
8	Rohail	17.0	Male	64.0	NaN	Karachi	Accepted
9	Kamran	18.0	Male	53.0	98.98	Multan	Rejected

In [26]:
```
df.isnull().sum()
```

Out[26]:
```
Name                      10
Age                       10
Gender                    10
Admission Test Score      11
High School Percentage    11
City                      10
Admission Status          10
dtype: int64
```

To find Non-Null Values

In [29]:
```
df.notnull()
```

Out[29]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	True	True	True	True	True	True	True
1	True	True	True	True	True	True	False
2	True	True	True	True	False	True	True
3	True	True	True	True	True	True	True
4	True	True	True	True	True	True	False
...	...	...	...	...	...	...	...
152	True	True	True	True	True	True	True
153	True	True	True	True	True	True	True
154	True	True	True	True	True	True	True
155	True	True	True	True	True	True	True
156	True	True	True	True	True	True	True

157 rows × 7 columns

In [31]:
```
df.notnull().sum()
```

Out[31]:
```
Name                      147
Age                       147
Gender                    147
Admission Test Score      146
High School Percentage    146
City                      147
Admission Status          147
dtype: int64
```

To get Dataset Information

In [34]:
```
df.describe()
```

Out[34]:

	Age	Admission Test Score	High School Percentage
count	147.000000	146.000000	146.000000
mean	19.680272	77.657534	75.684726
std	4.540512	16.855343	17.368014
min	-1.000000	-5.000000	-10.000000
25%	18.000000	68.250000	65.052500
50%	20.000000	79.000000	77.545000
75%	22.000000	89.000000	88.312500
max	24.000000	150.000000	110.500000

```
In [36]: df.columns
```

```
Out[36]: Index(['Name', 'Age', 'Gender', 'Admission Test Score',
                'High School Percentage', 'City', 'Admission Status'],
               dtype='object')
```

```
In [38]: df.size
```

```
Out[38]: 1099
```

```
In [40]: df.shape
```

```
Out[40]: (157, 7)
```

```
In [42]: df.ndim
```

```
Out[42]: 2
```

```
In [44]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 7 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Name                    147 non-null    object 
 1   Age                     147 non-null    float64
 2   Gender                  147 non-null    object 
 3   Admission Test Score    146 non-null    float64
 4   High School Percentage  146 non-null    float64
 5   City                    147 non-null    object 
 6   Admission Status        147 non-null    object 
dtypes: float64(3), object(4)
memory usage: 8.7+ KB
```

to DROP NuLL Values

```
In [47]: #df=df.dropna()
```

To deal with Null Values (Replace with some other value like Mean of any column)

```
In [50]: df["Age"].replace(np.NaN,10,inplace=True)
```

```
C:\Users\Omkar\AppData\Local\Temp\ipykernel_16356\311224403.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


  df["Age"].replace(np.NaN,10,inplace=True)
```

In [52]:
```
df
```

Out[52]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24.0	Female	50.0	68.90	Quetta	Rejected
1	Waqar	21.0	Female	99.0	60.73	Karachi	NaN
2	Bushra	17.0	Male	89.0	NaN	Islamabad	Accepted
3	Aliya	17.0	Male	55.0	85.29	Karachi	Rejected
4	Bilal	20.0	Male	65.0	61.13	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	19.0	Female	85.0	78.09	Quetta	Accepted
153	Bilal	17.0	Female	81.0	84.40	Islamabad	Rejected
154	Fatima	21.0	Female	98.0	50.86	Multan	Accepted
155	Shoaib	-1.0	Male	91.0	80.12	Quetta	Accepted
156	Maaz	17.0	Male	88.0	86.85	Lahore	Accepted

157 rows × 7 columns

In [54]:
```
df.isnull().sum()
```

Out[54]:
```
Name                      10
Age                        0
Gender                    10
Admission Test Score      11
High School Percentage    11
City                      10
Admission Status          10
dtype: int64
```

To Change Data Type

In [57]:
```
df["Age"] = df["Age"].astype(int)
```

```
In [59]: df
```

Out[59]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24	Female	50.0	68.90	Quetta	Rejected
1	Waqar	21	Female	99.0	60.73	Karachi	NaN
2	Bushra	17	Male	89.0	NaN	Islamabad	Accepted
3	Aliya	17	Male	55.0	85.29	Karachi	Rejected
4	Bilal	20	Male	65.0	61.13	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	19	Female	85.0	78.09	Quetta	Accepted
153	Bilal	17	Female	81.0	84.40	Islamabad	Rejected
154	Fatima	21	Female	98.0	50.86	Multan	Accepted
155	Shoaib	-1	Male	91.0	80.12	Quetta	Accepted
156	Maaz	17	Male	88.0	86.85	Lahore	Accepted

157 rows × 7 columns

To replace Null Values by Mean

```
In [62]: Avg_percentage=df["High School Percentage"].astype('float').mean()

In [64]: Avg_percentage

Out[64]: 75.68472602739726

In [66]: df["High School Percentage"].replace(np.NaN,Avg_percentage,inplace=True)
```

```
C:\Users\Omkar\AppData\Local\Temp\ipykernel_16356\3697381515.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.



  df["High School Percentage"].replace(np.NaN,Avg_percentage,inplace=True)
```

```
In [68]: df.isnull().sum()
```

```
Out[68]: Name                      10
         Age                        0
         Gender                    10
         Admission Test Score      11
         High School Percentage     0
         City                      10
         Admission Status          10
         dtype: int64
```

```
In [70]: df
```

Out[70]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24	Female	50.0	68.900000	Quetta	Rejected
1	Waqar	21	Female	99.0	60.730000	Karachi	NaN
2	Bushra	17	Male	89.0	75.684726	Islamabad	Accepted
3	Aliya	17	Male	55.0	85.290000	Karachi	Rejected
4	Bilal	20	Male	65.0	61.130000	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	19	Female	85.0	78.090000	Quetta	Accepted
153	Bilal	17	Female	81.0	84.400000	Islamabad	Rejected
154	Fatima	21	Female	98.0	50.860000	Multan	Accepted
155	Shoaib	-1	Male	91.0	80.120000	Quetta	Accepted
156	Maaz	17	Male	88.0	86.850000	Lahore	Accepted

157 rows × 7 columns

To convert Categorial into Numerial Variable

```
In [73]: df["Gender"].replace({'Female':0,'Male':1},inplace = True)
```

```
C:\Users\Omkar\AppData\Local\Temp\ipykernel_16356\3985573958.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


  df["Gender"].replace({'Female':0,'Male':1},inplace = True)
C:\Users\Omkar\AppData\Local\Temp\ipykernel_16356\3985573958.py:1: FutureWarning:
Downcasting behavior in `replace` is deprecated and will be removed in a future v
ersion. To retain the old behavior, explicitly call `result.infer_objects(copy=Fa
lse)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_dow
ncasting', True)`
  df["Gender"].replace({'Female':0,'Male':1},inplace = True)
```

In [75]:
```
df
```

Out[75]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24	0.0	50.0	68.900000	Quetta	Rejected
1	Waqar	21	0.0	99.0	60.730000	Karachi	NaN
2	Bushra	17	1.0	89.0	75.684726	Islamabad	Accepted
3	Aliya	17	1.0	55.0	85.290000	Karachi	Rejected
4	Bilal	20	1.0	65.0	61.130000	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	19	0.0	85.0	78.090000	Quetta	Accepted
153	Bilal	17	0.0	81.0	84.400000	Islamabad	Rejected
154	Fatima	21	0.0	98.0	50.860000	Multan	Accepted
155	Shoaib	-1	1.0	91.0	80.120000	Quetta	Accepted
156	Maaz	17	1.0	88.0	86.850000	Lahore	Accepted

157 rows × 7 columns

In [77]:
```
df["Admission Status"].replace({'Rejected':0,'Accepted':1},inplace = True)
```

```
C:\Users\Omkar\AppData\Local\Temp\ipykernel_16356\999870716.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


  df["Admission Status"].replace({'Rejected':0,'Accepted':1},inplace = True)
C:\Users\Omkar\AppData\Local\Temp\ipykernel_16356\999870716.py:1: FutureWarning:
Downcasting behavior in `replace` is deprecated and will be removed in a future v
ersion. To retain the old behavior, explicitly call `result.infer_objects(copy=Fa
lse)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_dow
ncasting', True)`
  df["Admission Status"].replace({'Rejected':0,'Accepted':1},inplace = True)
```

In [79]:
```
df
```

Out[79]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	24	0.0	50.0	68.900000	Quetta	0.0
1	Waqar	21	0.0	99.0	60.730000	Karachi	NaN
2	Bushra	17	1.0	89.0	75.684726	Islamabad	1.0
3	Aliya	17	1.0	55.0	85.290000	Karachi	0.0
4	Bilal	20	1.0	65.0	61.130000	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	19	0.0	85.0	78.090000	Quetta	1.0
153	Bilal	17	0.0	81.0	84.400000	Islamabad	0.0
154	Fatima	21	0.0	98.0	50.860000	Multan	1.0
155	Shoaib	-1	1.0	91.0	80.120000	Quetta	1.0
156	Maaz	17	1.0	88.0	86.850000	Lahore	1.0

157 rows × 7 columns

To count

In [82]:
```
df["Gender"].value_counts()
```

Out[82]:
```
Gender
0.0    83
1.0    64
Name: count, dtype: int64
```

In [84]:
```
df["Age"].value_counts()
```

Out[84]:
```
Age
 17     24
 23     19
 19     18
 22     18
 24     17
 20     17
 21     15
 18     14
 10     10
-1       5
Name: count, dtype: int64
```

Concept Hierachy

In [87]:
```python
def fun1(value):
    if (value < 20):
        return "teenager"
    elif (value >= 20 and value < 40):
        return "young"
    elif (value >= 40 and value < 60):
        return "middle aged"
    elif (value >= 60):
        return "senior citizen"
    else:
        pass
```

In [89]:
```python
df["Age"] = df["Age"].apply(fun1)
```

In [91]:
```python
df
```

Out[91]:

	Name	Age	Gender	Admission Test Score	High School Percentage	City	Admission Status
0	Shehroz	young	0.0	50.0	68.900000	Quetta	0.0
1	Waqar	young	0.0	99.0	60.730000	Karachi	NaN
2	Bushra	teenager	1.0	89.0	75.684726	Islamabad	1.0
3	Aliya	teenager	1.0	55.0	85.290000	Karachi	0.0
4	Bilal	young	1.0	65.0	61.130000	Lahore	NaN
...	...	...	...	...	...	...	...
152	Ali	teenager	0.0	85.0	78.090000	Quetta	1.0
153	Bilal	teenager	0.0	81.0	84.400000	Islamabad	0.0
154	Fatima	young	0.0	98.0	50.860000	Multan	1.0
155	Shoaib	teenager	1.0	91.0	80.120000	Quetta	1.0
156	Maaz	teenager	1.0	88.0	86.850000	Lahore	1.0

157 rows × 7 columns