Question

•Use Pandas to clean and preprocess a messy dataset, documenting the steps taken during the cleaning process.

PANDAS:- pandas is a popular python library used for data manipulation and analysis.It provides data structures and functions necessary to perform tasks such as reading and writing data, data cleaning,data exploration and data analysis. Pandas offers a wide range of functionalities for data manipulation, such as selecting specific columns, filtering data, handling missing values, merging datasets and much more.

Importing the pandas library:-

```
#Importing the pandas library
import pandas as pd
```

Checking the version of the pandas

```
#Cheaking the version of the pandas
print(pd.__version__)
```

```
1.5.3
```

Creating a pandas dataframe from a dictionary and performing basic operation.

```
#Creating a dictionary containing data
data={'Name':['Ukasha','Atika','Mustapha','Hafsa','Jalaluddeen','Zainab','Jabir','Aishatu','Abdulgat
```

Create a dataframe from the dictionary.

```
#Creating a dataframe from the dictionary
df=pd.DataFrame(data)
```

Display the dataframe

```
#Displaying the dataframe.
print(df)
```

```
          Name  Age  Gender
0       Ukasha   18    Male
```

```
1        Atika   12  Female
2     Mustapha   26    Male
3        Hafsa   30  Female
4  Jalaluddeen   21    Male
5       Zainab   15  Female
6        Jabir   20    Male
7      Aishatu   17  Female
8  Abdulgaffar   14    Male
9       Jamila   22  Female
```

Basic information about the dataframe.

```
#Basic information about the dataframe
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    10 non-null     object
 1   Age     10 non-null     int64
 2   Gender  10 non-null     object
dtypes: int64(1), object(2)
memory usage: 368.0+ bytes
None
```

Calculate descriptive statistics

```
#Calculating descriptive statistics
print(df.describe())
```

```
             Age
count  10.000000
mean   19.500000
std     5.542763
min    12.000000
25%    15.500000
50%    19.000000
75%    21.750000
max    30.000000
```

Filter rows based on a condition

```
#Filter rows based on a condition
#This line will allow me to filter the column of AGE that are greater than 20
filtered_df=df[df['Age']>20]


print(filtered_df)
```

```
           Name  Age  Gender
2     Mustapha   26    Male
3        Hafsa   30  Female
4  Jalaluddeen   21    Male
9       Jamila   22  Female
```

```
#Filter rows based on condition
#This line of code will allow me to filter the column of AGE that are less than 20
filtered_df=df[df['Age']<20]

print(filtered_df)
```

```
              Name  Age  Gender
    0       Ukasha   18    Male
    1        Atika   12  Female
    5       Zainab   15  Female
    7      Aishatu   17  Female
    8  Abdulgaffar   14    Male
```

Filtering the gender from our dataframe

```
#Filtering the gender of male
filtered_df=df[df['Gender']=="Male"]
print(filtered_df)
```

```
              Name  Age Gender
    0       Ukasha   18   Male
    2     Mustapha   26   Male
    4  Jalaluddeen   21   Male
    6        Jabir   20   Male
    8  Abdulgaffar   14   Male
```

```
#Filtering the gender of female
filtered_df=df[df['Gender']=="Female"]
print(filtered_df)
```

```
           Name  Age  Gender
    1    Atika   12  Female
    3    Hafsa   30  Female
    5   Zainab   15  Female
    7  Aishatu   17  Female
    9   Jamila   22  Female
```