DA warmup exercise

In [1]:
```
import pandas as pd
```

In [3]:
```
df=pd.read_csv("marks.csv")
```

In [4]:
```
df
```

Out[4]:

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

In [5]:
```
df['Gender']=['Male','Female','Female','Male','Male','Female','Female']
```

In [6]:
```
df
```

Out[6]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
1	Aadhya	89	AB	Female
2	Krisha	99	AA	Female
3	Vedant	87	AB	Male
4	Parshv	90	AC	Male
5	Mittal	83	BA	Female
6	Archana	82	BB	Female

In [7]:
```
df.head(3)
```

Out[7]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
1	Aadhya	89	AB	Female
2	Krisha	99	AA	Female

In [8]:
```
df.tail(3)
```

Out[8]:

	Name	Marks	Grades	Gender
4	Parshv	90	AC	Male
5	Mittal	83	BA	Female
6	Archana	82	BB	Female

In [9]:
```
# Shape of the dataframe
```

In [11]:
```
df.shape                          #donot use () as it is an attribute and not a method
```

Out[11]:
```
(7, 4)
```

In [12]:
```
# information about the whole dataframe
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Name    7 non-null      object
 1   Marks   7 non-null      int64 
 2   Grades  7 non-null      object
 3   Gender  7 non-null      object
dtypes: int64(1), object(3)
memory usage: 356.0+ bytes
```

Check for null values in data frame

In [14]:
```
df.isnull()
```

Out[14]:

	Name	Marks	Grades	Gender
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
5	False	False	False	False
6	False	False	False	False

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
Name      0
Marks     0
Grades    0
Gender    0
dtype: int64
```

In [17]:
```
df.isnull().sum(axis=1)      #with axis 1 it shows for column values
```

Out[17]:
```
0    0
1    0
2    0
3    0
4    0
5    0
6    0
dtype: int64
```

overall statistics of dataframe

In [19]:
```
df.describe()    # for all numerical only
```

Out[19]:

	Marks
count	7.000000
mean	89.714286
std	6.676184
min	82.000000
25%	85.000000
50%	89.000000
75%	94.000000
max	99.000000

In [20]:
```
df.describe(include='all')  #includes non int values as well
```

Out[20]:

	Name	Marks	Grades	Gender
count	7	7.000000	7	7
unique	7	NaN	5	2
top	Priyang	NaN	AA	Female
freq	1	NaN	2	4
mean	NaN	89.714286	NaN	NaN
std	NaN	6.676184	NaN	NaN
min	NaN	82.000000	NaN	NaN
25%	NaN	85.000000	NaN	NaN
50%	NaN	89.000000	NaN	NaN
75%	NaN	94.000000	NaN	NaN
max	NaN	99.000000	NaN	NaN

Find unique values from gender column

In [21]:
```
df['Gender'].unique()
```

Out[21]:
```
array(['Male', 'Female'], dtype=object)
```

In [22]:
```
df['Gender'].nunique()   #how many unique values in columns
```

Out[22]:
```
2
```

count of uniques values

In [23]:
```
df['Gender'].value_counts()
```

Out[23]:
```
Gender
Female    4
Male      3
Name: count, dtype: int64
```

Find number of students with marks 90 to 100 inclusive

method 1

In [24]:
```
df
```

Out[24]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
1	Aadhya	89	AB	Female
2	Krisha	99	AA	Female
3	Vedant	87	AB	Male
4	Parshv	90	AC	Male
5	Mittal	83	BA	Female
6	Archana	82	BB	Female

In [28]:
```
df2=df.loc[df['Marks']>90]
```

In [29]:
```
df2
```

Out[29]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
2	Krisha	99	AA	Female

In [33]:
```
df2['Marks'].value_counts()
```

Out[33]:
```
Marks
98    1
99    1
Name: count, dtype: int64
```

Find number of students with marks 90 to 100 inclusive

method 2 using length method

In [34]:
```
df
```

Out[34]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
1	Aadhya	89	AB	Female
2	Krisha	99	AA	Female
3	Vedant	87	AB	Male
4	Parshv	90	AC	Male
5	Mittal	83	BA	Female
6	Archana	82	BB	Female

In [36]:
```
df[df['Marks']>=90]
```

Out[36]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
2	Krisha	99	AA	Female
4	Parshv	90	AC	Male

In [39]:
```
df[(df['Marks']>=90)&(df['Marks']<100)]          #for using end or operator (parenthesis is must)
```

Out[39]:

	Name	Marks	Grades	Gender
0	Priyang	98	AA	Male
2	Krisha	99	AA	Female
4	Parshv	90	AC	Male

In [40]:
```
len(df[(df['Marks']>=90)&(df['Marks']<100)])    # length function shows the lengh of df
```

Out[40]: 3

In [ ]:
```

```

Find number of students with marks 90 to 100 inclusive

method 3 using between method

In [42]:
```
sum(df['Marks'].between(90,100))      #this includes the 90 and 100
```

Out[42]: 3

Find average marks

In [50]:
```
df['Marks'].mean()
```

Out[50]: 89.71428571428571

In [51]:
```
df['Marks'].min()
```

Out[51]: 82

In [52]:
```
df['Marks'].max()
```

Out[52]:
```
99
```

Apply method (use own method) and new columns

In [54]:
```
def half_marks(x):
    return x/2
```

In [56]:
```
df['Marks'].apply(half_marks)
```

Out[56]:
```
0    49.0
1    44.5
2    49.5
3    43.5
4    45.0
5    41.5
6    41.0
Name: Marks, dtype: float64
```

In [57]:
```
df['halfmarks']=df['Marks'].apply(half_marks)
```

In [58]:
```
df
```

Out[58]:

	Name	Marks	Grades	Gender	halfmarks
0	Priyang	98	AA	Male	49.0
1	Aadhya	89	AB	Female	44.5
2	Krisha	99	AA	Female	49.5
3	Vedant	87	AB	Male	43.5
4	Parshv	90	AC	Male	45.0
5	Mittal	83	BA	Female	41.5
6	Archana	82	BB	Female	41.0

using lambda function

In [59]:
```
df['double marks']=df['Marks'].apply(lambda x:x*2)
```

In [60]:
```
df
```

Out[60]:

	Name	Marks	Grades	Gender	halfmarks	double marks
0	Priyang	98	AA	Male	49.0	196
1	Aadhya	89	AB	Female	44.5	178
2	Krisha	99	AA	Female	49.5	198
3	Vedant	87	AB	Male	43.5	174
4	Parshv	90	AC	Male	45.0	180
5	Mittal	83	BA	Female	41.5	166
6	Archana	82	BB	Female	41.0	164

Map function (e.g. Change male to 1 and females to 0)

In [62]:
```
df['MF_mapping']=df['Gender'].map({'Male':1,'Female':0})
```

In [63]:
```
df
```

Out[63]:

	Name	Marks	Grades	Gender	halfmarks	double marks	MF_mapping
0	Priyang	98	AA	Male	49.0	196	1
1	Aadhya	89	AB	Female	44.5	178	0
2	Krisha	99	AA	Female	49.5	198	0
3	Vedant	87	AB	Male	43.5	174	1
4	Parshv	90	AC	Male	45.0	180	1
5	Mittal	83	BA	Female	41.5	166	0
6	Archana	82	BB	Female	41.0	164	0

Drop the columns

In [67]:
```
df.drop(columns=['halfmarks','double marks'],inplace=True)      #or df.drop(['halfmarks','double marks'],axis=1)
```

In [68]:
```
df
```

Out[68]:

	Name	Marks	Grades	Gender	MF_mapping
0	Priyang	98	AA	Male	1
1	Aadhya	89	AB	Female	0
2	Krisha	99	AA	Female	0
3	Vedant	87	AB	Male	1
4	Parshv	90	AC	Male	1
5	Mittal	83	BA	Female	0
6	Archana	82	BB	Female	0

sorting as per marks columns

In [72]:
```
df.sort_values(['Marks'],ascending=True)
```

Out[72]:

	Name	Marks	Grades	Gender	MF_mapping
6	Archana	82	BB	Female	0
5	Mittal	83	BA	Female	0
3	Vedant	87	AB	Male	1
1	Aadhya	89	AB	Female	0
4	Parshv	90	AC	Male	1
0	Priyang	98	AA	Male	1
2	Krisha	99	AA	Female	0

Display ONLY name and marks of Female students

In [79]:
```
df[df['Gender']=='Female'][['Name','Marks']]
```

Out[79]:

	Name	Marks
1	Aadhya	89
2	Krisha	99
5	Mittal	83
6	Archana	82

In [ ]:
```

```