Practicing Pandas Exercise Using Small Dataset.

Importing PYHTON Liabraries

PANDAS Exersice

[278]:
```
import pandas as pd
```

Created Dictionary for Imaginary Data

dict1 is a Dictionary and df is a DataFrame which stores data in the pandas.

[279]:
```
dict1 ={'Name':['Shweta','Neetu','Rina','Vedant','Amit',
                'Shree','Amy'],
                'Marks':[98,87,90,89,100,98,92],
                'Gender':['Female','Female','Female','Male','Male',
                        'Female','Female']
                }
df=pd.DataFrame(dict1)
df
```

[279]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

1. Display First 3 Rows of the Dataset

[280]:
```
df.head(3)
```

[280]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female

2. Display Last 3 Rows of the Dataset

[281]:
```
df.tail(3)
```

[281]:

	Name	Marks	Gender
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

3. Find shape of the Dataset(Find How many Numbers of Rows And Columns in the Dataset)

[282]:
```
df.shape
```

[282]:
```
(7, 3)
```

Above shape of Dataset Displays in the form of tuple and In above tuple 7 has '0' index number in the tuple and 3 has '1' so If I want to print it then I will use below code.

[283]:
```
print('Number of Rows',df.shape[0])
print('Number of Columns',df.shape[1])
```

```
Number of Rows 7
Number of Columns 3
```

4. Get all the information About Our Dataset like Number of rows and columns,Datatypes of Each Column and how much Memory Requied by dataset.

[284]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Name    7 non-null      object
 1   Marks   7 non-null      int64 
 2   Gender  7 non-null      object
dtypes: int64(1), object(2)
memory usage: 184.0+ bytes
```

5. Check if Null Values present in the Dataset or not.

[285]:
```
df.isnull().sum(axis=0)
```

[285]:
```
Name      0
```

```
Marks     0
Gender    0
dtype: int64
```

6. Find Overall Statistics Of the Dataframe.

[286]:
```
df.describe()
```

[286]:

	Marks
count	7.000000
mean	93.428571
std	5.159365
min	87.000000
25%	89.500000
50%	92.000000
75%	98.000000
max	100.000000

7. Find Unique Values from Gender Column.

[287]:
```
df
```

[287]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

[288]:
```
df['Gender'].unique()
```

[288]:
```
array(['Female', 'Male'], dtype=object)
```

8. Find the number of Unique Values from Gender Column.

[289]:
```
df['Gender'].nunique()
```

[289]:
```
2
```

9. Display count of Unique Values from Gender Column.

[290]:
```
df['Gender'].value_counts()
```

[290]:
```
Female    5
Male      2
Name: Gender, dtype: int64
```

10. Find Total Number of students having Marks Between 90 To 100.

[291]:
```
df['Marks']
```

[291]:
```
0     98
1     87
2     90
3     89
4    100
5     98
6     92
Name: Marks, dtype: int64
```

[292]:
```
df1=df[(df['Marks']>90) & (df['Marks']<100)]
df1
```

[292]:

	Name	Marks	Gender
0	Shweta	98	Female
5	Shree	98	Female
6	Amy	92	Female

[293]:
```
len(df1)
```

[293]:
```
3
```

IF I use between method then it will include 90 and 100 as well.

[294]:
```
df2=df[df['Marks'].between(90,100)]
df2
```

[294]:

	Name	Marks	Gender
0	Shweta	98	Female
2	Rina	90	Female
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

[295]:
```
len(df2)
```

[295]:
```
5
```

11. Find Average Marks

[296]:
```
df
```

[296]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

[297]:
```
df['Marks'].mean()
```

[297]:
```
93.42857142857143
```

12. Find a Name who has Highest marks

[298]:
```
df
```

[298]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

[299]:
```
#Finding Maximum marks
df['Marks'].max()
```

[299]:
```
100
```

[300]:
```
#finding name with maximum marks
df[df['Marks']==df['Marks'].max()]['Name'].values[0]
```

[300]:
```
'Amit'
```

13. Find a Name who has Lowest marks

[301]:
```
#Finding Minimum marks
df['Marks'].min()
```

[301]:
```
87
```

[302]:
```
#finding name with minimum marks
df[df['Marks']==df['Marks'].min()]['Name'].values[0]
```

[302]:
```
'Neetu'
```

14. Apply Method

[303]:
```
# Userdefined Function means function created by the user
def marks(x):
    return x/2
```

[304]:
```
#Apply above function on the Marks Columm in the dataset we use apply() method
df['Marks'].apply(marks)
```

[304]:
```
0    49.0
1    43.5
2    45.0
3    44.5
4    50.0
5    49.0
6    46.0
Name: Marks, dtype: float64
```

[305]:
```
# Add New columm name Half_marks
df['Half_marks']=df['Marks'].apply(marks)
```

[306]:
```
df
```

[306]:

	Name	Marks	Gender	Half_marks
0	Shweta	98	Female	49.0
1	Neetu	87	Female	43.5
2	Rina	90	Female	45.0
3	Vedant	89	Male	44.5
4	Amit	100	Male	50.0
5	Shree	98	Female	49.0
6	Amy	92	Female	46.0

[307]:
```
# Remove float number in Half_marks column
df['Half_marks']=df['Marks'].apply(lambda x:x//2)
df
```

[307]:

	Name	Marks	Gender	Half_marks
0	Shweta	98	Female	49

1	Neetu	87	Female	43
2	Rina	90	Female	45
3	Vedant	89	Male	44
4	Amit	100	Male	50
5	Shree	98	Female	49
6	Amy	92	Female	46

[308]:
```
df['Name'].apply(len)
```

[308]:
```
0    6
1    5
2    4
3    6
4    4
5    5
6    3
Name: Name, dtype: int64
```

15. Map Function

[309]:
```
df
```

[309]:

	Name	Marks	Gender	Half_marks
0	Shweta	98	Female	49
1	Neetu	87	Female	43
2	Rina	90	Female	45
3	Vedant	89	Male	44
4	Amit	100	Male	50
5	Shree	98	Female	49
6	Amy	92	Female	46

[310]:
```
df['Male_female']=df['Gender'].map({'Male':0,'Female':1})
df
```

[310]:

	Name	Marks	Gender	Half_marks	Male_female
0	Shweta	98	Female	49	1
1	Neetu	87	Female	43	1
2	Rina	90	Female	45	1
3	Vedant	89	Male	44	0
4	Amit	100	Male	50	0
5	Shree	98	Female	49	1
6	Amy	92	Female	46	1

16. Drop the unnecessary columns

[311]:
```
df
```

[311]:

	Name	Marks	Gender	Half_marks	Male_female
0	Shweta	98	Female	49	1
1	Neetu	87	Female	43	1
2	Rina	90	Female	45	1
3	Vedant	89	Male	44	0
4	Amit	100	Male	50	0
5	Shree	98	Female	49	1
6	Amy	92	Female	46	1

[312]:
```
#if only want to one column then
df.drop('Male_female',axis=1)
```

[312]:

	Name	Marks	Gender	Half_marks
0	Shweta	98	Female	49
1	Neetu	87	Female	43
2	Rina	90	Female	45
3	Vedant	89	Male	44
4	Amit	100	Male	50
5	Shree	98	Female	49
6	Amy	92	Female	46

[313]:
```
#I want to drop all unnecessary columns
df.drop(['Male_female','Half_marks'],axis=1)
```

[313]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female

6	Amy	92	Female

[314]:
```
df
```

[314]:

	Name	Marks	Gender	Half_marks	Male_female
0	Shweta	98	Female	49	1
1	Neetu	87	Female	43	1
2	Rina	90	Female	45	1
3	Vedant	89	Male	44	0
4	Amit	100	Male	50	0
5	Shree	98	Female	49	1
6	Amy	92	Female	46	1

[315]:
```
# we want to modify original dataframe then
df.drop(['Male_female','Half_marks'],axis=1,inplace=True)
```

[316]:
```
df
```

[316]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

17. Print Name of Columns

[317]:
```
df.columns
```

[317]:
```
Index(['Name', 'Marks', 'Gender'], dtype='object')
```

18. Sort The Dataframe As per the Marks Column

[318]:
```
df
```

[318]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

[319]:
```
df.sort_values(by='Marks',ascending=False)
```

[319]:

	Name	Marks	Gender
4	Amit	100	Male
0	Shweta	98	Female
5	Shree	98	Female
6	Amy	92	Female
2	Rina	90	Female
3	Vedant	89	Male
1	Neetu	87	Female

19.Display Name & Marks of the Male Students

[320]:
```
df
```

[320]:

	Name	Marks	Gender
0	Shweta	98	Female
1	Neetu	87	Female
2	Rina	90	Female
3	Vedant	89	Male
4	Amit	100	Male
5	Shree	98	Female
6	Amy	92	Female

[321]:
```
df[df['Gender']=='Female'][['Name','Marks']]
```

[321]:

	Name	Marks
0	Shweta	98
1	Neetu	87
2	Rina	90

5	Shree	98
6	Amy	92

[322]:
```
#Another method to solve above problem here let's do this for male
df[df['Gender'].isin(['Male'])][['Name','Marks']]
```

[322]:

	Name	Marks
3	Vedant	89
4	Amit	100

Pandas Exercise Complete

[ ]: