# Pandas Data Analysis projects

March 4, 2024

[119]:
```
## import the library
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

[120]:
```
dict1 = {'Name':
 ↪['priyanka','aditya','krishna','vedanth','prashav','mital','archana'],
        'Marks':[98,89,78,88,99,66,77],
        'Gender':['female','male','male','male','male','male','female']
        }

df1 = pd.DataFrame(dict1)
df1
```

[120]:

| | Name | Marks | Gender |
|---|---|---|---|
| 0 | priyanka | 98 | female |
| 1 | aditya | 89 | male |
| 2 | krishna | 78 | male |
| 3 | vedanth | 88 | male |
| 4 | prashav | 99 | male |
| 5 | mital | 66 | male |
| 6 | archana | 77 | female |

[121]:
```
##Q1, Display top 3 rows of dataframe
df1.head(3)
```

[121]:

| | Name | Marks | Gender |
|---|---|---|---|
| 0 | priyanka | 98 | female |
| 1 | aditya | 89 | male |
| 2 | krishna | 78 | male |

[122]:
```
##Q2, Display last 3 rows of dataset
df1.tail(3)
```

[122]:

| | Name | Marks | Gender |
|---|---|---|---|
| 4 | prashav | 99 | male |
| 5 | mital | 66 | male |

```
6  archana     77  female
```

[123]:
```
##3, Find the shape of our dataset
df1.shape
```

[123]:
```
(7, 3)
```

[124]:
```
print('numbers of rows',df1.shape[0])
print('numbers of column',df1.shape[1])
```

```
numbers of rows 7
numbers of column 3
```

[125]:
```
##Q4,Get the information about dataset like total number of rows and total␣
 ↪numbers
##of columns datatype of each columnsans memory requirements.
df1.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: 296.0+ bytes
```

[126]:
```
##Q5, Check null vakues of the data set
df1.isnull().sum(axis=1)
```

[126]:
```
0    0
1    0
2    0
3    0
4    0
5    0
6    0
dtype: int64
```

[127]:
```
##Q6,Get overall statistic about the  dataframe.
df1.describe(include = 'all')
```

[127]:

| | Name | Marks | Gender |
|---|---|---|---|
| count | 7 | 7.0 | 7 |
| unique | 7 | NaN | 2 |
| top | priyanka | NaN | male |

```
freq           1    NaN      5
mean         NaN   85.0    NaN
std          NaN   12.0    NaN
min          NaN   66.0    NaN
25%          NaN   77.5    NaN
50%          NaN   88.0    NaN
75%          NaN   93.5    NaN
max          NaN   99.0    NaN
```

[128]:
```
##Q7, find the unique value from the gender columns
df1['Gender'].unique()
```

[128]:
```
array(['female', 'male'], dtype=object)
```

[129]:
```
##Q8, Find the numbers of unique values in Gender columns
df1['Gender'].nunique()
```

[129]:
```
2
```

[130]:
```
##Q9, Display the count of unique value Gender columns,
df1['Gender'].value_counts()
```

[130]:
```
male      5
female    2
Name: Gender, dtype: int64
```

[131]:
```
##Q10,Find the total numbers of students having marks between 90 to 100 (␣
 ↪inclussive)using method
df1[df1['Marks']>=90]
```

[131]:
```
       Name  Marks  Gender
0  priyanka     98  female
4   prashav     99    male
```

[68]:
```
## without using between method,
len(df1[(df1['Marks'] >= 90) & (df1['Marks'] <= 100)])
```

[68]:
```
2
```

[69]:
```
## with using between method,
sum(df1['Marks'].between(90,100))
```

[69]:
```
2
```

[70]:
```
##Q11, Find the everage marks.
df1
```

[70]:

| | Name | Marks | Gender |
|---|---|---|---|
| 0 | priyanka | 98 | female |
| 1 | aditya | 89 | male |
| 2 | krishna | 78 | male |
| 3 | vedanth | 88 | male |
| 4 | prashav | 99 | male |
| 5 | mital | 66 | male |
| 6 | archana | 77 | female |

[71]:
```
df1['Marks'].mean()
```

[71]: 85.0

[72]:
```
##Q12,APPLY METHODE.
def marks(x):
    return x/2
```

[73]:
```
df1['Half_Marks'] = df1['Marks'].apply(marks)
```

[74]:
```
df1
```

[74]:

| | Name | Marks | Gender | Half_Marks |
|---|---|---|---|---|
| 0 | priyanka | 98 | female | 49.0 |
| 1 | aditya | 89 | male | 44.5 |
| 2 | krishna | 78 | male | 39.0 |
| 3 | vedanth | 88 | male | 44.0 |
| 4 | prashav | 99 | male | 49.5 |
| 5 | mital | 66 | male | 33.0 |
| 6 | archana | 77 | female | 38.5 |

[75]:
```
## and  using lambda function
df1['Marks'].apply(lambda x:x//2)
```

[75]:
```
0    49
1    44
2    39
3    44
4    49
5    33
6    38
Name: Marks, dtype: int64
```

[76]:
```
##Q13, Find the lenghth of string availabe in Name columns
df1['Name'].apply(len)
```

[76]:
```
0    8
1    6
2    7
```

```
3    7
4    7
5    5
6    7
Name: Name, dtype: int64
```

[102]:
```
## Map function
df1
```

[102]:

| | Name | Marks | Half_Marks |
|---|---|---|---|
| 0 | priyanka | 98 | 49.0 |
| 1 | aditya | 89 | 44.5 |
| 2 | krishna | 78 | 39.0 |
| 3 | vedanth | 88 | 44.0 |
| 4 | prashav | 99 | 49.5 |
| 5 | mital | 66 | 33.0 |
| 6 | archana | 77 | 38.5 |

[84]:
```
## i want to convert gender column male is 1 and female is 0 using map function
df1['Male_Female'] = df1['Gender'].map({'male':1,'female':0})
df1
```

[84]:

| | Name | Marks | Gender | Half_Marks | Male_Female |
|---|---|---|---|---|---|
| 0 | priyanka | 98 | female | 49.0 | 0 |
| 1 | aditya | 89 | male | 44.5 | 1 |
| 2 | krishna | 78 | male | 39.0 | 1 |
| 3 | vedanth | 88 | male | 44.0 | 1 |
| 4 | prashav | 99 | male | 49.5 | 1 |
| 5 | mital | 66 | male | 33.0 | 1 |
| 6 | archana | 77 | female | 38.5 | 0 |

[85]:
```
# Define the mapping
gender_mapping = {'male': 1, 'female': 0}

# Apply the mapping to the 'Gender' column
df1['Gender'] = df1['Gender'].map(gender_mapping)
```

[86]:
```
df1
```

[86]:

| | Name | Marks | Gender | Half_Marks | Male_Female |
|---|---|---|---|---|---|
| 0 | priyanka | 98 | 0 | 49.0 | 0 |
| 1 | aditya | 89 | 1 | 44.5 | 1 |
| 2 | krishna | 78 | 1 | 39.0 | 1 |
| 3 | vedanth | 88 | 1 | 44.0 | 1 |
| 4 | prashav | 99 | 1 | 49.5 | 1 |
| 5 | mital | 66 | 1 | 33.0 | 1 |
| 6 | archana | 77 | 0 | 38.5 | 0 |

[87]:
```
##Q14, drop the single columns using tuple
df1.drop('Half_Marks',axis=1)
```

[87]:

| | Name | Marks | Gender | Male_Female |
|---|---|---|---|---|
| 0 | priyanka | 98 | 0 | 0 |
| 1 | aditya | 89 | 1 | 1 |
| 2 | krishna | 78 | 1 | 1 |
| 3 | vedanth | 88 | 1 | 1 |
| 4 | prashav | 99 | 1 | 1 |
| 5 | mital | 66 | 1 | 1 |
| 6 | archana | 77 | 0 | 0 |

[ ]:
```
## drop the multiple columns using  inside list
df1.drop(['Male_Female','Gender'],axis=1,inplace=True)
```

[94]:
```
df1
```

[94]:

| | Name | Marks | Half_Marks |
|---|---|---|---|
| 0 | priyanka | 98 | 49.0 |
| 1 | aditya | 89 | 44.5 |
| 2 | krishna | 78 | 39.0 |
| 3 | vedanth | 88 | 44.0 |
| 4 | prashav | 99 | 49.5 |
| 5 | mital | 66 | 33.0 |
| 6 | archana | 77 | 38.5 |

[96]:
```
##Q14, print the name of the columns,
df1.columns
```

[96]:
```
Index(['Name', 'Marks', 'Half_Marks'], dtype='object')
```

[97]:
```
##Q15, sort the dataframe as per the marks columns
df1.sort_values(by='Marks')
```

[97]:

| | Name | Marks | Half_Marks |
|---|---|---|---|
| 5 | mital | 66 | 33.0 |
| 6 | archana | 77 | 38.5 |
| 2 | krishna | 78 | 39.0 |
| 3 | vedanth | 88 | 44.0 |
| 1 | aditya | 89 | 44.5 |
| 0 | priyanka | 98 | 49.0 |
| 4 | prashav | 99 | 49.5 |

[98]:
```
df1.sort_values(by='Marks',ascending=False)
```

[98]:

| | Name | Marks | Half_Marks |
|---|---|---|---|
| 4 | prashav | 99 | 49.5 |
| 0 | priyanka | 98 | 49.0 |

```
1    aditya     89        44.5
3   vedanth     88        44.0
2   krishna     78        39.0
6   archana     77        38.5
5     mital     66        33.0
```

[100]:
```
df1.sort_values(by=['Marks','Half_Marks'],ascending=False)
```

[100]:
```
        Name  Marks  Half_Marks
4   prashav     99        49.5
0  priyanka     98        49.0
1    aditya     89        44.5
3   vedanth     88        44.0
2   krishna     78        39.0
6   archana     77        38.5
5     mital     66        33.0
```

[140]:
```
## Q16, display only name and marks female students
df1[df1['Gender'] == 'female'][['Name', 'Marks']]
```

[140]:
```
        Name  Marks
0  priyanka     98
6   archana     77
```

[142]:
```

```

[ ]:
```

```