

# # Data Analytics with Python on class marks Data set

In [1]:

```
import pandas as pd
```

In [2]:

```
df=pd.read_csv("class_marks.csv")
```

In [3]:

```
df
```

Out[3]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	NaN	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	NaN	NaN	3.0	6.0	9.0	NaN
2	33	4.0	5.0	5.0	1.0	5.0	5.0	NaN	NaN	8.0	NaN
3	25	4.0	6.0	6.0	3.0	2.0	2.0	NaN	NaN	NaN	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	NaN	NaN	10.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	NaN	NaN	NaN	4.0
82	27	2.0	2.0	5.0	3.0	NaN	NaN	NaN	NaN	7.0	3.0
83	37	4.0	6.0	6.0	2.0	NaN	NaN	NaN	NaN	9.0	4.0
84	28	4.0	NaN	5.0	4.0	5.0	4.0	NaN	NaN	6.0	NaN
85	30	4.0	6.0	NaN	NaN	NaN	NaN	3.0	5.0	7.0	1.0

86 rows × 12 columns

In [4]:

```
df.shape
```

Out[4]:

(86, 12)

Here shape function returns the no.of rows and no.of columns contains of dataset

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86 entries, 0 to 85
Data columns (total 12 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Total     86 non-null    int64  
 1   Q1aM4    85 non-null    float64 
 2   Q1bM6    73 non-null    float64 
 3   Q2aM6    72 non-null    float64 
 4   Q2bM4    75 non-null    float64 
 5   Q3aM5    52 non-null    float64 
 6   Q3bM5    51 non-null    float64 
 7   Q4aM3    32 non-null    float64 
 8   Q4bM7    26 non-null    float64 
 9   Q5M10    56 non-null    float64 
 10  Q6aM4    33 non-null    float64 
 11  Q6bM6    28 non-null    float64 
dtypes: float64(11), int64(1)
memory usage: 8.2 KB
```

```
# info() function returns the each column information
```

In [6]:

```
df.dtypes
```

Out[6]:

```
Total      int64
Q1aM4    float64
Q1bM6    float64
Q2aM6    float64
Q2bM4    float64
Q3aM5    float64
Q3bM5    float64
Q4aM3    float64
Q4bM7    float64
Q5M10    float64
Q6aM4    float64
Q6bM6    float64
dtype: object
```

dtypes returns data types of all the columns in a data frame

In [7]:

```
df.head()
```

Out[7]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	Q
0	37	4.0	5.0	6.0	4.0	2.0	1.0	NaN	5.0	8.0	4.0	
1	32	4.0	3.0	4.0	3.0	NaN	NaN	3.0	6.0	9.0	NaN	
2	33	4.0	5.0	5.0	1.0	5.0	5.0	NaN	NaN	8.0	NaN	
3	25	4.0	6.0	6.0	3.0	2.0	2.0	NaN	NaN	NaN	2.0	
4	36	3.0	6.0	4.0	4.0	5.0	4.0	NaN	NaN	10.0	NaN	

head() function returns the first five rows data

In [8]:

```
df.head(7)
```

Out[8]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	Q
0	37	4.0	5.0	6.0	4.0	2.0	1.0	NaN	5.0	8.0	4.0	
1	32	4.0	3.0	4.0	3.0	NaN	NaN	3.0	6.0	9.0	NaN	
2	33	4.0	5.0	5.0	1.0	5.0	5.0	NaN	NaN	8.0	NaN	
3	25	4.0	6.0	6.0	3.0	2.0	2.0	NaN	NaN	NaN	2.0	
4	36	3.0	6.0	4.0	4.0	5.0	4.0	NaN	NaN	10.0	NaN	
5	20	4.0	6.0	6.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	
6	25	3.0	4.0	NaN	2.0	5.0	5.0	NaN	NaN	6.0	NaN	

here head(7) functions returns the first seven rows of data. Based on input as numeric value

In [9]:

```
df.tail()
```

Out[9]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
81	34	3.0	6.0	3.0	4.0	5.0	3.0	NaN	NaN	NaN	4.0
82	27	2.0	2.0	5.0	3.0	NaN	NaN	NaN	NaN	7.0	3.0
83	37	4.0	6.0	6.0	2.0	NaN	NaN	NaN	NaN	9.0	4.0
84	28	4.0	NaN	5.0	4.0	5.0	4.0	NaN	NaN	6.0	NaN
85	30	4.0	6.0	NaN	NaN	NaN	NaN	3.0	5.0	7.0	1.0

`tail()` function returns the last five rows of data

In [10]:

```
df.tail(8)
```

Out[10]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
78	33	2.0	3.0	6.0	4.0	5.0	5.0	NaN	NaN	8.0	NaN
79	27	2.0	6.0	NaN	3.0	2.0	5.0	NaN	NaN	9.0	NaN
80	31	4.0	6.0	6.0	2.0	2.0	5.0	NaN	NaN	6.0	NaN
81	34	3.0	6.0	3.0	4.0	5.0	3.0	NaN	NaN	NaN	4.0
82	27	2.0	2.0	5.0	3.0	NaN	NaN	NaN	NaN	7.0	3.0
83	37	4.0	6.0	6.0	2.0	NaN	NaN	NaN	NaN	9.0	4.0
84	28	4.0	NaN	5.0	4.0	5.0	4.0	NaN	NaN	6.0	NaN
85	30	4.0	6.0	NaN	NaN	NaN	NaN	3.0	5.0	7.0	1.0

Here `tail(8)` functions returns the last eight rows of data. Based on input as numeric value

In [11]:

```
df.shape
```

Out[11]:

(86, 12)

In [12]:

```
df=pd.read_csv("class_marks.csv",skiprows = 1)
```

In [13]:

```
df
```

Out[13]:

	37	4	5	6	4.1	2	1	Unnamed: 7	5.1	8	4.2	6.1	
0	32	4.0	3.0	4.0	3.0	NaN	NaN		3.0	6.0	9.0	NaN	NaN
1	33	4.0	5.0	5.0	1.0	5.0	5.0		NaN	NaN	8.0	NaN	NaN
2	25	4.0	6.0	6.0	3.0	2.0	2.0		NaN	NaN	NaN	2.0	NaN
3	36	3.0	6.0	4.0	4.0	5.0	4.0		NaN	NaN	10.0	NaN	NaN
4	20	4.0	6.0	6.0	4.0	NaN	NaN		NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...		...	...	...	...	...
80	34	3.0	6.0	3.0	4.0	5.0	3.0		NaN	NaN	NaN	4.0	6.0
81	27	2.0	2.0	5.0	3.0	NaN	NaN		NaN	NaN	7.0	3.0	5.0
82	37	4.0	6.0	6.0	2.0	NaN	NaN		NaN	NaN	9.0	4.0	6.0
83	28	4.0	NaN	5.0	4.0	5.0	4.0		NaN	NaN	6.0	NaN	NaN
84	30	4.0	6.0	NaN	NaN	NaN	NaN		3.0	5.0	7.0	1.0	4.0

85 rows × 12 columns

Here skiprows = 1, means delete one row. By default, it will delete one row from the top.

In [14]:

```
df=pd.read_csv("class_marks.csv",skiprows = 5)
```

In [15]:

```
df.shape
```

Out[15]:

(81, 12)

Here first five rows are deleted from DataFrame

In [16]:

```
df=pd.read_csv("class_marks.csv",skiprows =[2,4,6])
```

Here 3 rows are deleted whose row index value is 2,4,6

In [17]:

```
df=pd.read_csv("class_marks.csv",skipfooter=3)
```

```
<ipython-input-17-7e37e6a9ae1d>:1: ParserWarning: Falling back to the 'python' engine because the 'c' engine does not support skipfooter; you can avoid this warning by specifying engine='python'.
```

```
df=pd.read_csv("class_marks.csv",skipfooter=3)
```

In [18]:

```
df=pd.read_csv("class_marks.csv",header = None)
```

In [19]:

```
df
```

Out[19]:

	0	1	2	3	4	5	6	7	8	9	10	C
0	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	G
1	37	4	5	6	4	2	1	NaN	5	8	4	
2	32	4	3	4	3	NaN	NaN	3	6	9	NaN	
3	33	4	5	5	1	5	5	NaN	NaN	8	NaN	
4	25	4	6	6	3	2	2	NaN	NaN	NaN	2	
...	...	...	...	...	...	...	...	...	...	...	...	
82	34	3	6	3	4	5	3	NaN	NaN	NaN	4	
83	27	2	2	5	3	NaN	NaN	NaN	NaN	7	3	
84	37	4	6	6	2	NaN	NaN	NaN	NaN	9	4	
85	28	4	NaN	5	4	5	4	NaN	NaN	6	NaN	
86	30	4	6	NaN	NaN	NaN	NaN	3	5	7	1	

87 rows × 12 columns



usecols: Sometimes all the columns of the dataset may not be necessary for the analysis. Especially when the dataset is too big, it always makes sense to present part of it. Otherwise, it becomes too hard to understandb. Here is a way to import a few columns.

In [20]:

```
df=pd.read_csv("class_marks.csv",usecols=['Total','Q1aM4','Q1bM6'])
```

In [21]:

```
df
```

Out[21]:

	Total	Q1aM4	Q1bM6
<b>0</b>	37	4.0	5.0
<b>1</b>	32	4.0	3.0
<b>2</b>	33	4.0	5.0
<b>3</b>	25	4.0	6.0
<b>4</b>	36	3.0	6.0
...	...	...	...
<b>81</b>	34	3.0	6.0
<b>82</b>	27	2.0	2.0
<b>83</b>	37	4.0	6.0
<b>84</b>	28	4.0	NaN
<b>85</b>	30	4.0	6.0

86 rows × 3 columns

In [22]:

```
df=pd.read_csv("class_marks.csv", names=['Time', 'X', 'Y', 'Z'])
```

In [23]:

```
df
```

Out[23]:

Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	Q6bM
37	4	5	6	4	2	1	NaN	5	8	4	
32	4	3	4	3	NaN	NaN	3	6	9	NaN	NaN
33	4	5	5	1	5	5	NaN	NaN	8	NaN	NaN
25	4	6	6	3	2	2	NaN	NaN	NaN	2	NaN
...	...	...	...	...	...	...	...	...	...	...	...
34	3	6	3	4	5	3	NaN	NaN	NaN	4	
27	2	2	5	3	NaN	NaN	NaN	NaN	7	3	
37	4	6	6	2	NaN	NaN	NaN	NaN	9	4	
28	4	NaN	5	4	5	4	NaN	NaN	6	NaN	NaN
30	4	6	NaN	NaN	NaN	NaN	3	5	7	1	

87 rows × 4 columns

In [24]:

```
df=pd.read_csv("class_marks.csv", dtype={'BAR': 'S10'})
```

In [25]:

```
df
```

Out[25]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	NaN	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	NaN	NaN	3.0	6.0	9.0	NaN
2	33	4.0	5.0	5.0	1.0	5.0	5.0	NaN	NaN	8.0	NaN
3	25	4.0	6.0	6.0	3.0	2.0	2.0	NaN	NaN	NaN	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	NaN	NaN	10.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	NaN	NaN	NaN	4.0
82	27	2.0	2.0	5.0	3.0	NaN	NaN	NaN	NaN	7.0	3.0
83	37	4.0	6.0	6.0	2.0	NaN	NaN	NaN	NaN	9.0	4.0
84	28	4.0	NaN	5.0	4.0	5.0	4.0	NaN	NaN	6.0	NaN
85	30	4.0	6.0	NaN	NaN	NaN	NaN	3.0	5.0	7.0	1.0

86 rows × 12 columns

In [26]:

```
df=pd.read_csv("class_marks.csv")
df
```

Out[26]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	NaN	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	NaN	NaN	3.0	6.0	9.0	NaN
2	33	4.0	5.0	5.0	1.0	5.0	5.0	NaN	NaN	8.0	NaN
3	25	4.0	6.0	6.0	3.0	2.0	2.0	NaN	NaN	NaN	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	NaN	NaN	10.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	NaN	NaN	NaN	4.0
82	27	2.0	2.0	5.0	3.0	NaN	NaN	NaN	NaN	7.0	3.0
83	37	4.0	6.0	6.0	2.0	NaN	NaN	NaN	NaN	9.0	4.0
84	28	4.0	NaN	5.0	4.0	5.0	4.0	NaN	NaN	6.0	NaN
85	30	4.0	6.0	NaN	NaN	NaN	NaN	3.0	5.0	7.0	1.0

86 rows × 12 columns

In [27]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86 entries, 0 to 85
Data columns (total 12 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Total     86 non-null    int64  
 1   Q1aM4    85 non-null    float64 
 2   Q1bM6    73 non-null    float64 
 3   Q2aM6    72 non-null    float64 
 4   Q2bM4    75 non-null    float64 
 5   Q3aM5    52 non-null    float64 
 6   Q3bM5    51 non-null    float64 
 7   Q4aM3    32 non-null    float64 
 8   Q4bM7    26 non-null    float64 
 9   Q5M10    56 non-null    float64 
 10  Q6aM4    33 non-null    float64 
 11  Q6bM6    28 non-null    float64 
dtypes: float64(11), int64(1)
memory usage: 8.2 KB
```

In [28]:

```
#df=pd.read_csv("class_marks.csv", dtype={'Q1aM4': 'int64'})
```

# Null Values

In [29]:

```
df.isnull()
```

Out[29]:

Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	False	False	False	False	False	False	True	False	False	False
1	False	False	False	False	False	True	True	False	False	True
2	False	False	False	False	False	False	True	True	False	True
3	False	False	False	False	False	False	True	True	True	False
4	False	False	False	False	False	False	True	True	False	True
...	...	...	...	...	...	...	...	...	...	...
81	False	False	False	False	False	False	True	True	True	False
82	False	False	False	False	False	True	True	True	False	False
83	False	False	False	False	False	True	True	True	False	False
84	False	False	True	False	False	False	True	True	False	True
85	False	False	False	True	True	True	True	False	False	False

86 rows × 12 columns

In [30]:

```
df.isnull().sum()
```

Out[30]:

```
Total      0
Q1aM4     1
Q1bM6    13
Q2aM6    14
Q2bM4    11
Q3aM5    34
Q3bM5    35
Q4aM3    54
Q4bM7    60
Q5M10    30
Q6aM4    53
Q6bM6    58
dtype: int64
```

In [31]:

```
df.fillna(0)
```

Out[31]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 12 columns



In [32]:

```
df.isnull().sum()
```

Out[32]:

```
Total      0
Q1aM4     1
Q1bM6    13
Q2aM6    14
Q2bM4    11
Q3aM5    34
Q3bM5    35
Q4aM3    54
Q4bM7    60
Q5M10    30
Q6aM4    53
Q6bM6    58
dtype: int64
```

In [33]:

```
df.fillna(0,inplace=True)
```

In [34]:

```
df
```

Out[34]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 12 columns

In [35]:

```
df
```

Out[35]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 12 columns

In [36]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86 entries, 0 to 85
Data columns (total 12 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Total     86 non-null    int64  
 1   Q1aM4    86 non-null    float64 
 2   Q1bM6    86 non-null    float64 
 3   Q2aM6    86 non-null    float64 
 4   Q2bM4    86 non-null    float64 
 5   Q3aM5    86 non-null    float64 
 6   Q3bM5    86 non-null    float64 
 7   Q4aM3    86 non-null    float64 
 8   Q4bM7    86 non-null    float64 
 9   Q5M10    86 non-null    float64 
 10  Q6aM4    86 non-null    float64 
 11  Q6bM6    86 non-null    float64 
dtypes: float64(11), int64(1)
memory usage: 8.2 KB
```

In [37]:

```
df
```

Out[37]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 12 columns

for more parameters in read\_csv, please refer

[https://pandas.pydata.org/docs/reference/api/pandas.read\\_csv.html](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html)  
[\(https://pandas.pydata.org/docs/reference/api/pandas.read\\_csv.html\)](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html)

in the same way, while reading excel file can manipulate data

some parameters are same in csv and excel. please refer below document

[https://pandas.pydata.org/docs/reference/api/pandas.read\\_excel.html](https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html)  
[\(https://pandas.pydata.org/docs/reference/api/pandas.read\\_excel.html\)](https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html)

In [38]:

```
df.Total
```

Out[38]:

```
0    37  
1    32  
2    33  
3    25  
4    36  
..  
81   34  
82   27  
83   37  
84   28  
85   30  
Name: Total, Length: 86, dtype: int64
```

In [39]:

```
df.Total.count
```

Out[39]:

```
<bound method Series.count of 0      37  
1    32  
2    33  
3    25  
4    36  
..  
81   34  
82   27  
83   37  
84   28  
85   30  
Name: Total, Length: 86, dtype: int64>
```

In [40]:

```
df.Total.sum()
```

Out[40]:

2518

In [41]:

```
df.Total.mean()
```

Out[41]:

29.27906976744186

In [42]:

```
df["Total"]
```

Out[42]:

```
0    37  
1    32  
2    33  
3    25  
4    36  
     ..  
81   34  
82   27  
83   37  
84   28  
85   30  
Name: Total, Length: 86, dtype: int64
```

In [43]:

```
df["Total"].mean()
```

Out[43]:

```
29.27906976744186
```

In [44]:

```
df[df.Total>30]
```

Out[44]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
9	35	2.0	4.0	5.0	4.0	5.0	5.0	0.0	0.0	0.0	4.0
10	37	3.0	5.0	6.0	4.0	0.0	0.0	3.0	6.0	0.0	4.0
13	33	3.0	3.0	6.0	4.0	3.0	5.0	0.0	0.0	9.0	0.0
15	33	3.0	0.0	2.0	1.0	5.0	5.0	3.0	7.0	0.0	4.0
17	36	3.0	4.0	6.0	4.0	0.0	0.0	0.0	0.0	9.0	4.0
21	32	4.0	6.0	5.0	3.0	5.0	5.0	0.0	3.0	0.0	4.0
24	33	1.0	6.0	6.0	3.0	5.0	5.0	3.0	3.0	7.0	3.0
26	33	4.0	6.0	6.0	3.0	4.0	0.0	0.0	0.0	0.0	4.0
27	31	2.0	6.0	6.0	1.0	5.0	5.0	3.0	3.0	0.0	0.0
28	38	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
32	32	4.0	6.0	6.0	4.0	2.0	0.0	0.0	0.0	10.0	0.0
33	40	0.0	0.0	6.0	4.0	5.0	5.0	3.0	7.0	0.0	4.0
36	39	2.0	0.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
37	31	4.0	4.0	6.0	4.0	0.0	0.0	0.0	0.0	10.0	0.0
42	38	4.0	6.0	6.0	4.0	5.0	5.0	3.0	5.0	0.0	0.0
43	33	4.0	5.0	0.0	0.0	0.0	0.0	3.0	4.0	8.0	4.0
44	36	3.0	6.0	6.0	2.0	0.0	0.0	2.0	7.0	10.0	0.0
45	34	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	6.0	0.0
46	36	4.0	5.0	6.0	4.0	5.0	5.0	0.0	0.0	7.0	0.0
47	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
49	39	3.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
50	34	4.0	5.0	0.0	0.0	5.0	5.0	0.0	0.0	5.0	4.0
53	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
54	31	2.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	0.0	4.0
55	38	3.0	5.0	6.0	4.0	0.0	0.0	0.0	0.0	10.0	4.0
56	35	2.0	6.0	0.0	0.0	0.0	0.0	3.0	7.0	7.0	4.0
59	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
61	34	4.0	5.0	6.0	4.0	0.0	5.0	0.0	0.0	0.0	4.0
62	34	3.0	4.0	6.0	4.0	5.0	5.0	0.0	0.0	7.0	0.0
64	36	1.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	9.0	0.0
65	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
<b>66</b>	31	4.0	5.0	5.0	2.0	5.0	3.0	1.0	5.0	7.0	0.0
<b>70</b>	35	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	5.0	0.0
<b>71</b>	36	4.0	5.0	6.0	4.0	5.0	5.0	0.0	0.0	7.0	0.0
<b>73</b>	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0
<b>78</b>	33	2.0	3.0	6.0	4.0	5.0	5.0	0.0	0.0	8.0	0.0
<b>80</b>	31	4.0	6.0	6.0	2.0	2.0	5.0	0.0	0.0	6.0	0.0
<b>81</b>	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
<b>83</b>	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0

<b>In [45]:</b>
-----------------

```
df[(df.Total>30) & (df.Q1aM4>3)]
```

**Out[45]:**

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
<b>0</b>	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
<b>1</b>	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
<b>2</b>	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
<b>21</b>	32	4.0	6.0	5.0	3.0	5.0	5.0	0.0	3.0	0.0	4.0
<b>26</b>	33	4.0	6.0	6.0	3.0	4.0	0.0	0.0	0.0	0.0	4.0
<b>32</b>	32	4.0	6.0	6.0	4.0	2.0	0.0	0.0	0.0	10.0	0.0
<b>37</b>	31	4.0	4.0	6.0	4.0	0.0	0.0	0.0	0.0	10.0	0.0
<b>42</b>	38	4.0	6.0	6.0	4.0	5.0	5.0	3.0	5.0	0.0	0.0
<b>43</b>	33	4.0	5.0	0.0	0.0	0.0	0.0	3.0	4.0	8.0	4.0
<b>46</b>	36	4.0	5.0	6.0	4.0	5.0	5.0	0.0	0.0	7.0	0.0
<b>50</b>	34	4.0	5.0	0.0	0.0	5.0	5.0	0.0	0.0	5.0	4.0
<b>53</b>	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
<b>61</b>	34	4.0	5.0	6.0	4.0	0.0	5.0	0.0	0.0	0.0	4.0
<b>65</b>	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
<b>66</b>	31	4.0	5.0	5.0	2.0	5.0	3.0	1.0	5.0	7.0	0.0
<b>70</b>	35	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	5.0	0.0
<b>71</b>	36	4.0	5.0	6.0	4.0	5.0	5.0	0.0	0.0	7.0	0.0
<b>73</b>	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0
<b>80</b>	31	4.0	6.0	6.0	2.0	2.0	5.0	0.0	0.0	6.0	0.0
<b>83</b>	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0

<b>In [46]:</b>
-----------------

## Loc&iloc

In [46]:

```
df.loc[3]
```

Out[46]:

```
Total    25.0
Q1aM4    4.0
Q1bM6    6.0
Q2aM6    6.0
Q2bM4    3.0
Q3aM5    2.0
Q3bM5    2.0
Q4aM3    0.0
Q4bM7    0.0
Q5M10    0.0
Q6aM4    2.0
Q6bM6    0.0
Name: 3, dtype: float64
```

In [47]:

```
df.loc[3,"Total"]
```

Out[47]:

```
25
```

In [48]:

```
df.loc[3,[ "Total", "Q6bM6"]]
```

Out[48]:

```
Total    25.0
Q6bM6    0.0
Name: 3, dtype: float64
```

In [49]:

```
df.loc[2:4, ]
```

Out[49]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	Q
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0	
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0	
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0	

◀ ▶

In [50]:

```
df.loc[2:4, ["Total", "Q6bM6"]]
```

Out[50]:

	Total	Q6bM6
2	33	0.0
3	25	0.0
4	36	0.0

In [51]:

```
df.iloc[3]
```

Out[51]:

```
Total      25.0
Q1aM4      4.0
Q1bM6      6.0
Q2aM6      6.0
Q2bM4      3.0
Q3aM5      2.0
Q3bM5      2.0
Q4aM3      0.0
Q4bM7      0.0
Q5M10      0.0
Q6aM4      2.0
Q6bM6      0.0
Name: 3, dtype: float64
```

In [52]:

```
df.iloc[3:5]
```

Out[52]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	Q
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0	
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0	

In [53]:

```
df.iloc[3:5,2]
```

Out[53]:

```
3      6.0
4      6.0
Name: Q1bM6, dtype: float64
```

In [54]:

```
df.iloc[3:5,2:6]
```

Out[54]:

	Q1bM6	Q2aM6	Q2bM4	Q3aM5
3	6.0	6.0	3.0	2.0
4	6.0	4.0	4.0	5.0

In [55]:

```
df.iloc[3:4,2:6]
```

Out[55]:

	Q1bM6	Q2aM6	Q2bM4	Q3aM5
3	6.0	6.0	3.0	2.0

In [56]:

```
df.loc[(df.Total == 40)]
```

Out[56]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
33	40	0.0	0.0	6.0	4.0	5.0	5.0	3.0	7.0	0.0	4.0
47	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
53	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
59	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
65	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
73	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0



In [57]:

```
df.loc[(df.Total == 40) & (df.Q2bM4 == 4)]
```

Out[57]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
33	40	0.0	0.0	6.0	4.0	5.0	5.0	3.0	7.0	0.0	4.0
47	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
53	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
59	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
65	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0



In [58]:

```
df.loc[9:15]
```

Out[58]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
9	35	2.0	4.0	5.0	4.0	5.0	5.0	0.0	0.0	0.0	4.0
10	37	3.0	5.0	6.0	4.0	0.0	0.0	3.0	6.0	0.0	4.0
11	8	2.0	2.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0
12	29	4.0	4.0	5.0	3.0	2.0	2.0	2.0	1.0	9.0	2.0
13	33	3.0	3.0	6.0	4.0	3.0	5.0	0.0	0.0	9.0	0.0
14	30	4.0	6.0	6.0	2.0	4.0	5.0	3.0	0.0	0.0	0.0
15	33	3.0	0.0	2.0	1.0	5.0	5.0	3.0	7.0	0.0	4.0

In [59]:

```
df.loc[9:15:2]
```

Out[59]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
9	35	2.0	4.0	5.0	4.0	5.0	5.0	0.0	0.0	0.0	4.0
11	8	2.0	2.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0
13	33	3.0	3.0	6.0	4.0	3.0	5.0	0.0	0.0	9.0	0.0
15	33	3.0	0.0	2.0	1.0	5.0	5.0	3.0	7.0	0.0	4.0

In [60]:

```
df.iloc[1:5]
```

Out[60]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4	Q
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0	
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0	
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0	
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0	

In [61]:

```
#Other Functions
```

In [62]:

```
df.Total.unique()
```

Out[62]:

```
array([37, 32, 33, 25, 36, 20, 35, 8, 29, 30, 24, 23, 15, 9, 22, 31, 38,
       27, 21, 40, 19, 39, 28, 34, 18, 3, 17], dtype=int64)
```

In [63]:

```
df.Total.nunique()
```

Out[63]:

```
27
```

In [64]:

```
df.Total.value_counts()
```

Out[64]:

```
25    8
33    7
40    6
36    6
29    6
31    5
34    5
20    4
27    4
32    3
37    3
38    3
30    3
21    3
28    3
35    3
39    2
23    2
22    2
24    1
19    1
18    1
17    1
15    1
9     1
8     1
3     1
Name: Total, dtype: int64
```

In [65]:

```
df.dropna()
```

Out[65]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 12 columns

In [66]:

```
df.shape
```

Out[66]:

(86, 12)

In [67]:

```
df.replace(0, -999999)
```

Out[67]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	
0	37	4.0	5.0	6.0	4.0	2.0	1.0	-999999.0	5.0	
1	32	4.0	3.0	4.0	3.0	-999999.0	-999999.0	3.0	6.0	
2	33	4.0	5.0	5.0	1.0	5.0	5.0	-999999.0	-999999.0	
3	25	4.0	6.0	6.0	3.0	2.0	2.0	-999999.0	-999999.0	-
4	36	3.0	6.0	4.0	4.0	5.0	4.0	-999999.0	-999999.0	
...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	-999999.0	-999999.0	-
82	27	2.0	2.0	5.0	3.0	-999999.0	-999999.0	-999999.0	-999999.0	
83	37	4.0	6.0	6.0	2.0	-999999.0	-999999.0	-999999.0	-999999.0	
84	28	4.0	-999999.0	5.0	4.0	5.0	4.0	-999999.0	-999999.0	
85	30	4.0	6.0	-999999.0	-999999.0	-999999.0	-999999.0	3.0	5.0	

86 rows × 12 columns

In [68]:

```
df.replace({0:8,38:40})
```

Out[68]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	8.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	8.0	8.0	3.0	6.0	9.0	8.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	8.0	8.0	8.0	8.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	8.0	8.0	8.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	8.0	8.0	10.0	8.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	8.0	8.0	8.0	4.0
82	27	2.0	2.0	5.0	3.0	8.0	8.0	8.0	8.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	8.0	8.0	8.0	8.0	9.0	4.0
84	28	4.0	8.0	5.0	4.0	5.0	4.0	8.0	8.0	6.0	8.0
85	30	4.0	6.0	8.0	8.0	8.0	8.0	3.0	5.0	7.0	1.0

86 rows × 12 columns

## Drop duplicates

In [69]:

```
emp = {"Name": ["Parker", "Smith", "William", "Parker"], "Age": [21, 32, 29, 21]}
info = pd.DataFrame(emp)
print(info)
```

```
      Name  Age
0    Parker   21
1     Smith   32
2   William   29
3    Parker   21
```

In [70]:

```
info.drop_duplicates()
```

Out[70]:

	Name	Age
0	Parker	21
1	Smith	32
2	William	29

In [71]:

```
info.drop_duplicates(inplace=True)
```

In [72]:

```
info
```

Out[72]:

	Name	Age
0	Parker	21
1	Smith	32
2	William	29

## New column creation

In [73]:

```
df
```

Out[73]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 12 columns

```
◀ ▶
```

In [74]:

```
df["Q1"] = df["Q1aM4"] + df["Q1bM6"]
```

In [75]:

```
df
```

Out[75]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 13 columns

```
◀ ▶
```

In [76]:

```
df["Grade"] = None
```

In [77]:

```
df
```

Out[77]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

```
◀ ▶
```

In [78]:

```
df.loc[df["Total"]>36, ["Grade"]] = "First"
```

In [79]:

```
df
```

Out[79]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

In [80]:

```
df.loc[(df["Total"]>30) & (df["Total"]<=36), ["Grade"]] = "Second"
```

In [81]:

```
df
```

Out[81]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

In [82]:

```
df.loc[(df["Total"]>20) & (df["Total"]<=30),["Grade"]]="Third"
df.loc[(df["Total"]>=16) & (df["Total"]<=20),["Grade"]]="Pass"
df.loc[(df["Total"]<16)& (df["Total"]>=0),["Grade"]]="Fail"
```

In [83]:

```
df
```

Out[83]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

## isin New columns with for loop

## Append

In [84]:

```
one = pd.DataFrame({  
    'Name': ['Parker', 'Smith', 'Allen', 'John', 'Parker'],  
    'subject_id':['sub1','sub2','sub4','sub6','sub5'],  
    'Marks_scored':[98,90,87,69,78]},  
    index=[1,2,3,4,5])  
  
two = pd.DataFrame({  
    'Name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],  
    'subject_id':['sub2','sub4','sub3','sub6','sub5'],  
    'Marks_scored':[89,80,79,97,88]},  
    index=[1,2,3,4,5])
```

In [85]:

```
print (one.append(two))
```

```
Name subject_id Marks_scored
1 Parker      sub1      98
2 Smith       sub2      90
3 Allen       sub4      87
4 John        sub6      69
5 Parker      sub5      78
1 Billy        sub2      89
2 Brian        sub4      80
3 Bran         sub3      79
4 Bryce        sub6      97
5 Betty        sub5      88
```

In [86]:

```
df.columns
```

Out[86]:

```
Index(['Total', 'Q1aM4', 'Q1bM6', 'Q2aM6', 'Q2bM4', 'Q3aM5', 'Q3bM5', 'Q4aM
3',
       'Q4bM7', 'Q5M10', 'Q6aM4', 'Q6bM6', 'Q1', 'Grade'],
      dtype='object')
```

In [87]:

```
df.Total[0]
```

Out[87]:

37

In [88]:

```
#df=d["MASV4_score"]
#dfT=df.
```

In [89]:

```
#dfT
```

In [90]:

```
my_list=[]
```

In [91]:

```
len(df)
```

Out[91]:

86

In [92]:

```
for i in range(len(df)):
    a=df.Q1aM4[i]+df.Q1bM6[i]
    my_list.append(a)
```

In [93]:

```
dfr = pd.DataFrame (my_list,columns=['1Q'])
```

In [94]:

```
dfr.count()
```

Out[94]:

```
1Q      86
dtype: int64
```

In [95]:

```
dfr.value_counts()
```

Out[95]:

```
1Q
10.0     19
8.0      17
9.0      16
7.0       9
4.0       7
6.0       5
5.0       5
3.0       3
2.0       2
0.0       2
1.0       1
dtype: int64
```

In [96]:

```
dff=pd.concat([df,dfr],axis=1)
```

In [97]:

```
dff
```

Out[97]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 15 columns

In [98]:

```
list_2Q=[]
list_3Q=[]
list_4Q=[]
list_6Q=[]
```

In [99]:

```
for i in range(len(df)):
    list_2Q.append(dff.Q2aM6[i]+dff.Q2bM4[i])
    list_3Q.append(dff.Q3aM5[i]+dff.Q3bM5[i])
    list_4Q.append(dff.Q4aM3[i]+dff.Q4bM7[i])
    list_6Q.append(dff.Q6aM4[i]+dff.Q6bM6[i])
```

In [100]:

```
len(list_3Q)
```

Out[100]:

86

In [101]:

```
sum_Q = pd.DataFrame(  
    {'Q2': list_2Q,  
     'Q3': list_3Q,  
     'Q4': list_4Q,  
     'Q6': list_6Q  
)
```

In [ ]:

In [102]:

```
sum_Q.columns
```

Out[102]:

```
Index(['Q2', 'Q3', 'Q4', 'Q6'], dtype='object')
```

In [103]:

```
sum_Q
```

Out[103]:

	Q2	Q3	Q4	Q6
0	10.0	3.0	5.0	10.0
1	7.0	0.0	9.0	0.0
2	6.0	10.0	0.0	0.0
3	9.0	4.0	0.0	2.0
4	8.0	9.0	0.0	0.0
...	...	...	...	...
81	7.0	8.0	0.0	10.0
82	8.0	0.0	0.0	8.0
83	8.0	0.0	0.0	10.0
84	9.0	9.0	0.0	0.0
85	0.0	0.0	8.0	5.0

86 rows × 4 columns

In [104]:

```
dfQ=pd.concat([dff,sum_Q],axis=1)
```

In [105]:

```
dfQ
```

Out[105]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
<b>0</b>	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
<b>1</b>	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
<b>2</b>	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
<b>3</b>	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
<b>4</b>	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
<b>81</b>	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
<b>82</b>	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
<b>83</b>	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
<b>84</b>	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
<b>85</b>	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 19 columns

## Drop columns

In [106]:

```
dfQ.drop("1Q",axis=1)
```

Out[106]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 18 columns

In [107]:

```
dfQ
```

Out[107]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 19 columns

In [108]:

```
#dfQ.drop(["1Q", "2Q", "3Q", "4Q", "6Q"], axis=1)
```

In [109]:

```
#dfQ.drop(columns=[ "Q1", "2Q", "3Q", "4Q", "6Q" ])
```

In [110]:

```
dfQ.drop([0, 1])
```

Out[110]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
<b>2</b>	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
<b>3</b>	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
<b>4</b>	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
<b>5</b>	20	4.0	6.0	6.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>6</b>	25	3.0	4.0	0.0	2.0	5.0	5.0	0.0	0.0	6.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
<b>81</b>	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
<b>82</b>	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
<b>83</b>	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
<b>84</b>	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
<b>85</b>	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

84 rows × 19 columns

In [111]:

```
dfQ.drop(["Grade", "1Q"], axis=1, inplace=True)
```

In [112]:

```
dfQ
```

Out[112]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 17 columns

In [113]:

```
list=[453,456]
```

In [114]:

```
import random
n = random.randint(450,540)
if n not in(list):
    list.append(n)
print(n)
```

455

## dtype conversion

In [115]:

```
import numpy as np
```

In [116]:

```
df['Q1aM4'].astype(np.int64)
```

Out[116]:

```
0      4  
1      4  
2      4  
3      4  
4      3  
..  
81     3  
82     2  
83     4  
84     4  
85     4  
Name: Q1aM4, Length: 86, dtype: int64
```

In [117]:

```
dfQ=dfQ.astype(np.int64)  
dfQ
```

Out[117]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4	5	6	4	2	1	0	5	8	4
1	32	4	3	4	3	0	0	3	6	9	0
2	33	4	5	5	1	5	5	0	0	8	0
3	25	4	6	6	3	2	2	0	0	0	2
4	36	3	6	4	4	5	4	0	0	10	0
..	...	...	...	...	...	...	...	...	...	...	...
81	34	3	6	3	4	5	3	0	0	0	4
82	27	2	2	5	3	0	0	0	0	7	3
83	37	4	6	6	2	0	0	0	0	9	4
84	28	4	0	5	4	5	4	0	0	6	0
85	30	4	6	0	0	0	0	3	5	7	1

86 rows × 17 columns

◀ ▶

In [118]:

```
dfQ
```

Out[118]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4	5	6	4	2	1	0	5	8	4
1	32	4	3	4	3	0	0	3	6	9	0
2	33	4	5	5	1	5	5	0	0	8	0
3	25	4	6	6	3	2	2	0	0	0	2
4	36	3	6	4	4	5	4	0	0	10	0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3	6	3	4	5	3	0	0	0	4
82	27	2	2	5	3	0	0	0	0	7	3
83	37	4	6	6	2	0	0	0	0	9	4
84	28	4	0	5	4	5	4	0	0	6	0
85	30	4	6	0	0	0	0	3	5	7	1

86 rows × 17 columns

In [119]:

```
dfQ["Q5"] = dfQ["Q5M10"]
```

In [120]:

```
dfQ
```

Out[120]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4	5	6	4	2	1	0	5	8	4
1	32	4	3	4	3	0	0	3	6	9	0
2	33	4	5	5	1	5	5	0	0	8	0
3	25	4	6	6	3	2	2	0	0	0	2
4	36	3	6	4	4	5	4	0	0	10	0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3	6	3	4	5	3	0	0	0	4
82	27	2	2	5	3	0	0	0	0	7	3
83	37	4	6	6	2	0	0	0	0	9	4
84	28	4	0	5	4	5	4	0	0	6	0
85	30	4	6	0	0	0	0	3	5	7	1

86 rows × 18 columns

In [121]:

```
dfQ.drop("Q5M10",axis=1)
```

Out[121]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q6aM4	Q6bM6
0	37	4	5	6	4	2	1	0	5	4	6
1	32	4	3	4	3	0	0	3	6	0	0
2	33	4	5	5	1	5	5	0	0	0	0
3	25	4	6	6	3	2	2	0	0	2	0
4	36	3	6	4	4	5	4	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3	6	3	4	5	3	0	0	4	6
82	27	2	2	5	3	0	0	0	0	3	5
83	37	4	6	6	2	0	0	0	0	4	6
84	28	4	0	5	4	5	4	0	0	0	0
85	30	4	6	0	0	0	0	3	5	1	4

86 rows × 17 columns

In [ ]:

## replace()

In [122]:

```
df
```

Out[122]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

In [123]:

```
df.replace(37, 38)
```

Out[123]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	38	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	38	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

In [124]:

```
df.Total.value_counts()
```

Out[124]:

```
25    8
33    7
40    6
36    6
29    6
31    5
34    5
20    4
27    4
32    3
37    3
38    3
30    3
21    3
28    3
35    3
39    2
23    2
22    2
24    1
19    1
18    1
17    1
15    1
9     1
8     1
3     1
Name: Total, dtype: int64
```

In [125]:

```
df.replace([37,38,39],40)
```

Out[125]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	40	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	40	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

```
◀ ▶
```

In [126]:

```
df.replace([37,38,39],40).Total.value_counts()
```

Out[126]:

```
40    14
25     8
33     7
36     6
29     6
34     5
31     5
27     4
20     4
21     3
28     3
30     3
32     3
35     3
23     2
22     2
15     1
8      1
9      1
24     1
17     1
18     1
19     1
3      1
Name: Total, dtype: int64
```

In [127]:

```
df.replace(0,np.NaN)
```

Out[127]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	NaN	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	NaN	NaN	3.0	6.0	9.0	NaN
2	33	4.0	5.0	5.0	1.0	5.0	5.0	NaN	NaN	8.0	NaN
3	25	4.0	6.0	6.0	3.0	2.0	2.0	NaN	NaN	NaN	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	NaN	NaN	10.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	NaN	NaN	NaN	4.0
82	27	2.0	2.0	5.0	3.0	NaN	NaN	NaN	NaN	7.0	3.0
83	37	4.0	6.0	6.0	2.0	NaN	NaN	NaN	NaN	9.0	4.0
84	28	4.0	NaN	5.0	4.0	5.0	4.0	NaN	NaN	6.0	NaN
85	30	4.0	6.0	NaN	NaN	NaN	NaN	3.0	5.0	7.0	1.0

86 rows × 14 columns



## Sorting

In [128]:

```
df.sort_values("Total")
```

Out[128]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
69	3	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
11	8	2.0	2.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0
23	9	4.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
22	15	4.0	4.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
76	17	2.0	3.0	4.0	2.0	4.0	2.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
59	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
53	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
47	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
33	40	0.0	0.0	6.0	4.0	5.0	5.0	3.0	7.0	0.0	4.0
73	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0

86 rows × 14 columns



In [129]:

```
df.sort_values("Total", ascending=True)
```

Out[129]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
69	3	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
11	8	2.0	2.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0
23	9	4.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
22	15	4.0	4.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
76	17	2.0	3.0	4.0	2.0	4.0	2.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
59	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
53	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
47	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
33	40	0.0	0.0	6.0	4.0	5.0	5.0	3.0	7.0	0.0	4.0
73	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0

86 rows × 14 columns

In [130]:

```
df.sort_values(["Total", "Q1aM4"], ascending=True)
```

Out[130]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
69	3	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
11	8	2.0	2.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0
23	9	4.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
22	15	4.0	4.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
76	17	2.0	3.0	4.0	2.0	4.0	2.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
47	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
59	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
53	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
65	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
73	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0

86 rows × 14 columns

In [131]:

```
df.sort_values(["Total", "Q1aM4"], ascending=[1,0])
```

Out[131]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
<b>69</b>	3	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
<b>11</b>	8	2.0	2.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0
<b>23</b>	9	4.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
<b>22</b>	15	4.0	4.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>76</b>	17	2.0	3.0	4.0	2.0	4.0	2.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
<b>65</b>	40	4.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	0.0
<b>73</b>	40	4.0	6.0	0.0	0.0	5.0	5.0	3.0	0.0	10.0	4.0
<b>47</b>	40	2.0	6.0	6.0	4.0	5.0	5.0	3.0	7.0	10.0	0.0
<b>59</b>	40	2.0	6.0	6.0	4.0	5.0	5.0	0.0	0.0	10.0	4.0
<b>33</b>	40	0.0	0.0	6.0	4.0	5.0	5.0	3.0	7.0	0.0	4.0

86 rows × 14 columns

## Rename

In [132]:

```
df.rename(columns={'Total':'TOTAL'},inplace=True)
```

In [133]:

```
df
```

Out[133]:

	TOTAL	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

```
◀ | ▶
```

In [134]:

```
dfQ.columns
```

Out[134]:

```
Index(['Total', 'Q1aM4', 'Q1bM6', 'Q2aM6', 'Q2bM4', 'Q3aM5', 'Q3bM5', 'Q4aM3',  
       'Q4bM7', 'Q5M10', 'Q6aM4', 'Q6bM6', 'Q1', 'Q2', 'Q3', 'Q4', 'Q6', 'Q5'],  
      dtype='object')
```

In [135]:

```
my_list =[dfQ.Q1[1],dfQ.Q2[1],dfQ.Q3[1],dfQ.Q4[1],dfQ.Q5[1],dfQ.Q6[1]]
```

In [136]:

```
dfQ
```

Out[136]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4	5	6	4	2	1	0	5	8	4
1	32	4	3	4	3	0	0	3	6	9	0
2	33	4	5	5	1	5	5	0	0	8	0
3	25	4	6	6	3	2	2	0	0	0	2
4	36	3	6	4	4	5	4	0	0	10	0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3	6	3	4	5	3	0	0	0	4
82	27	2	2	5	3	0	0	0	0	7	3
83	37	4	6	6	2	0	0	0	0	9	4
84	28	4	0	5	4	5	4	0	0	6	0
85	30	4	6	0	0	0	0	3	5	7	1

86 rows × 18 columns

In [137]:

```
#df["FTotal"] =sum(sorted([dfQ.Q1,dfQ.Q2,dfQ.Q3,dfQ.Q4,dfQ.Q5,dfQ.Q6])[2:])
```

In [138]:

```
dfQ
```

Out[138]:

	Total	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4	5	6	4	2	1	0	5	8	4
1	32	4	3	4	3	0	0	3	6	9	0
2	33	4	5	5	1	5	5	0	0	8	0
3	25	4	6	6	3	2	2	0	0	0	2
4	36	3	6	4	4	5	4	0	0	10	0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3	6	3	4	5	3	0	0	0	4
82	27	2	2	5	3	0	0	0	0	7	3
83	37	4	6	6	2	0	0	0	0	9	4
84	28	4	0	5	4	5	4	0	0	6	0
85	30	4	6	0	0	0	0	3	5	7	1

86 rows × 18 columns

In [139]:

```
df
```

Out[139]:

	TOTAL	Q1aM4	Q1bM6	Q2aM6	Q2bM4	Q3aM5	Q3bM5	Q4aM3	Q4bM7	Q5M10	Q6aM4
0	37	4.0	5.0	6.0	4.0	2.0	1.0	0.0	5.0	8.0	4.0
1	32	4.0	3.0	4.0	3.0	0.0	0.0	3.0	6.0	9.0	0.0
2	33	4.0	5.0	5.0	1.0	5.0	5.0	0.0	0.0	8.0	0.0
3	25	4.0	6.0	6.0	3.0	2.0	2.0	0.0	0.0	0.0	2.0
4	36	3.0	6.0	4.0	4.0	5.0	4.0	0.0	0.0	10.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
81	34	3.0	6.0	3.0	4.0	5.0	3.0	0.0	0.0	0.0	4.0
82	27	2.0	2.0	5.0	3.0	0.0	0.0	0.0	0.0	7.0	3.0
83	37	4.0	6.0	6.0	2.0	0.0	0.0	0.0	0.0	9.0	4.0
84	28	4.0	0.0	5.0	4.0	5.0	4.0	0.0	0.0	6.0	0.0
85	30	4.0	6.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0	1.0

86 rows × 14 columns

