1) Importing Pandas lib and reading data set from url.

```
import pandas as pd
```

2)Creating a data frame(2D)

```
data = {'Country': ['Belgium',  'India',  'Brazil'],

'Capital': ['Brussels',  'New Delhi',  'Brasilia'],

'Population': [11190846, 1303171035, 207847528]}


df = pd.DataFrame(data,columns=['df',  'Capital',  'Population'])
```

Subset of a dataframe.

```
df[1:]
```

```
    df    Capital  Population
1  NaN  New Delhi  1303171035
2  NaN   Brasilia   207847528
```

Building data frame.

```
dframe = pd.DataFrame([[1,'Bob', 'Builder'],
                  [2,'Sally', 'Baker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker'],
                  [3,'Scott', 'Candle Stick Maker']
                  ],
columns=['id','name', 'occupation'])
```

```
dframe
```

```
   id   name          occupation
0   1    Bob             Builder
1   2  Sally               Baker
2   3  Scott  Candle Stick Maker
```

Getting top and bottom 5 rows

```
# from a dataset

dframe.head()
```

```
   id   name          occupation
0   1    Bob             Builder
1   2  Sally               Baker
2   3  Scott  Candle Stick Maker
3   3  Scott  Candle Stick Maker
4   3  Scott  Candle Stick Maker
```

```
dframe.tail()
```

```
    id   name          occupation
7    3  Scott  Candle Stick Maker
8    3  Scott  Candle Stick Maker
9    3  Scott  Candle Stick Maker
10   3  Scott  Candle Stick Maker
11   3  Scott  Candle Stick Maker
```

Getting statistics

```
dframe.describe()
```

```
              id
count  12.000000
mean    2.750000
std     0.621582
min     1.000000
25%     3.000000
50%     3.000000
75%     3.000000
max     3.000000
```

Information of dataframe

```
dframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   id          12 non-null     int64 
 1   name        12 non-null     object
 2   occupation  12 non-null     object
dtypes: int64(1), object(2)
memory usage: 416.0+ bytes
```

Get counts of values

```
dframe.value_counts()
```

```
id  name   occupation
3   Scott  Candle Stick Maker    10
```

```
1   Bob    Builder                1
2   Sally  Baker                  1
dtype: int64
```

Importing a dataframe and storing it it base dataframe

```
base =
pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/Exerci
se.csv")
```

Info and description

```
base.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  90 non-null     int64 
 1   id          90 non-null     int64 
 2   diet        90 non-null     object
 3   pulse       90 non-null     int64 
 4   time        90 non-null     object
 5   kind        90 non-null     object
dtypes: int64(3), object(3)
memory usage: 4.3+ KB
```

```
base.describe()
```

```
       Unnamed: 0         id       pulse
count   90.000000  90.000000   90.000000
mean    44.500000  15.500000   99.700000
std     26.124701   8.703932   14.858471
min      0.000000   1.000000   80.000000
25%     22.250000   8.000000   90.250000
50%     44.500000  15.500000   96.000000
75%     66.750000  23.000000  103.000000
max     89.000000  30.000000  150.000000
```

```
base.head()
```

```
   Unnamed: 0  id     diet  pulse    time  kind
0           0   1  low fat     85   1 min  rest
1           1   1  low fat     85  15 min  rest
2           2   1  low fat     88  30 min  rest
3           3   2  low fat     90   1 min  rest
4           4   2  low fat     92  15 min  rest
```

Getting Columns and Rows

```
base.loc[1, "diet"]
```

```
{"type":"string"}
```

All Rows two columns

```
base.loc[:,[ "time", "kind"]]
```

```
      time     kind
0    1 min     rest
1   15 min     rest
2   30 min     rest
3    1 min     rest
4   15 min     rest
..     ...      ...
85  15 min  running
86  30 min  running
87   1 min  running
88  15 min  running
89  30 min  running

[90 rows x 2 columns]
```

Concatenating 2 dataframes. (dframe + base)

```
pd.concat([dframe, base], ignore_index=True)
```

```
     id   name          occupation  Unnamed: 0    diet  pulse    time   
kind
0     1    Bob             Builder         NaN     NaN    NaN     NaN   
NaN
1     2  Sally               Baker         NaN     NaN    NaN     NaN   
NaN
2     3  Scott  Candle Stick Maker         NaN     NaN    NaN     NaN   
NaN
3     3  Scott  Candle Stick Maker         NaN     NaN    NaN     NaN   
NaN
4     3  Scott  Candle Stick Maker         NaN     NaN    NaN     NaN   
NaN
..   ..    ...                 ...         ...     ...    ...     ...   
...
97   29    NaN                 NaN        85.0  no fat  135.0  15 min  
running
98   29    NaN                 NaN        86.0  no fat  130.0  30 min  
running
99   30    NaN                 NaN        87.0  no fat   99.0   1 min  
running
100  30    NaN                 NaN        88.0  no fat  111.0  15 min  
running
101  30    NaN                 NaN        89.0  no fat  150.0  30 min  
running

[102 rows x 8 columns]
```

Dropinnng columns in dataframe.

```
# Droping from base
```

```
base.drop(["pulse"], axis =1)
```

```
    Unnamed: 0  id     diet    time     kind
0            0   1  low fat   1 min     rest
1            1   1  low fat  15 min     rest
2            2   1  low fat  30 min     rest
3            3   2  low fat   1 min     rest
4            4   2  low fat  15 min     rest
..         ...  ..      ...     ...      ...
85          85  29   no fat  15 min  running
86          86  29   no fat  30 min  running
87          87  30   no fat   1 min  running
88          88  30   no fat  15 min  running
89          89  30   no fat  30 min  running

[90 rows x 5 columns]
```

Adding columns in dataframe "base"

```
consistency = base["id"]*2
```

```
consistency
```

```
0      2
1      2
2      2
3      4
4      4
      ..
85    58
86    58
87    60
88    60
89    60
Name: id, Length: 90, dtype: int64
```

```
# Adding consisitency column
```

```
base["consisitency"] = base["id"]*2
```

```
base
```

```
    Unnamed: 0  id     diet  pulse    time     kind  consisitency
0            0   1  low fat     85   1 min     rest             2
1            1   1  low fat     85  15 min     rest             2
2            2   1  low fat     88  30 min     rest             2
3            3   2  low fat     90   1 min     rest             4
4            4   2  low fat     92  15 min     rest             4
```

```
..          ...  ..      ...    ...     ...      ...            ...
85           85  29   no fat    135  15 min  running             58
86           86  29   no fat    130  30 min  running             58
87           87  30   no fat     99   1 min  running             60
88           88  30   no fat    111  15 min  running             60
89           89  30   no fat    150  30 min  running             60

[90 rows x 7 columns]
```

Sorting Dataframe

```
# Sorting "pulse" col

base.sort_values("pulse", ascending = True)
```

```
    Unnamed: 0  id     diet  pulse    time     kind  consisitency
9            9   4  low fat     80   1 min     rest             8
10          10   4  low fat     82  15 min     rest             8
11          11   4  low fat     83  30 min     rest             8
16          16   6   no fat     83  15 min     rest            12
15          15   6   no fat     83   1 min     rest            12
..         ...  ..      ...    ...     ...      ...           ...
85          85  29   no fat    135  15 min  running            58
80          80  27   no fat    140  30 min  running            54
83          83  28   no fat    140  30 min  running            56
77          77  26   no fat    143  30 min  running            52
89          89  30   no fat    150  30 min  running            60

[90 rows x 7 columns]
```

Cleaning Setting NaN cells to some value (0)

```
base.fillna(0)
```

```
    Unnamed: 0  id     diet  pulse    time     kind  consisitency
0            0   1  low fat     85   1 min     rest             2
1            1   1  low fat     85  15 min     rest             2
2            2   1  low fat     88  30 min     rest             2
3            3   2  low fat     90   1 min     rest             4
4            4   2  low fat     92  15 min     rest             4
..         ...  ..      ...    ...     ...      ...           ...
85          85  29   no fat    135  15 min  running            58
86          86  29   no fat    130  30 min  running            58
87          87  30   no fat     99   1 min  running            60
88          88  30   no fat    111  15 min  running            60
89          89  30   no fat    150  30 min  running            60

[90 rows x 7 columns]
```

Taking Sample from a big dataframe.

```
base.sample(frac=0.25)
```

```
    Unnamed: 0  id     diet  pulse    time     kind  consisitency
62          62  21  low fat    110  30 min  running            42
74          74  25  low fat    116  30 min  running            50
70          70  24  low fat    132  15 min  running            48
72          72  25  low fat     94   1 min  running            50
10          10   4  low fat     82  15 min     rest             8
0            0   1  low fat     85   1 min     rest             2
60          60  21  low fat     93   1 min  running            42
86          86  29   no fat    130  30 min  running            58
80          80  27   no fat    140  30 min  running            54
9            9   4  low fat     80   1 min     rest             8
31          31  11  low fat     86  15 min  walking            22
87          87  30   no fat     99   1 min  running            60
29          29  10   no fat    100  30 min     rest            20
3            3   2  low fat     90   1 min     rest             4
15          15   6   no fat     83   1 min     rest            12
30          30  11  low fat     86   1 min  walking            22
5            5   2  low fat     93  30 min     rest             4
79          79  27   no fat    126  15 min  running            54
69          69  24  low fat     87   1 min  running            48
18          18   7   no fat     87   1 min     rest            14
46          46  16   no fat     86  15 min  walking            32
24          24   9   no fat     97   1 min     rest            18
```

Splitting the dataframe for Regression purposes

```
y = base["pulse"]
```

```
x = base[["diet"]]
```

```
x
```

```
       diet
0   low fat
1   low fat
2   low fat
3   low fat
4   low fat
..      ...
85   no fat
86   no fat
87   no fat
88   no fat
89   no fat

[90 rows x 1 columns]
```

```
y
```

```
0      85
1      85
2      88
```

```
3      90
4      92
     ...
85    135
86    130
87     99
88    111
89    150
Name: pulse, Length: 90, dtype: int64
```