Pandas:-

1)Dropna means:-Drop NA\Non Assign.....df.dropna()...it wont change original files.

2)import pandas as pd

```
df = pd.read_csv('data.csv')

df.dropna(inplace = True)

print(df.to_string())
```

where (inplace): It is a boolean which makes the changes in data frame itself ont True.....It wont return new data file insted of tbat it will changes original data files

3)Fillna means:-Fill Na.....it fills the given value in the place of null. or by defining "Mean","Median" or "Mode"

4)import pandas as pd

```
df = pd.read_csv('data.csv')

x = df["Calories"].mode()[0]..... //Mode()[0] bcz in data set  there will be chance of having more than 1 mode value,therefore we assign [0] to accept 1'st mode value in dataset.

print(df.to_string())
```

5)defination:-*in pandas single column is called as "series"....

```
import pandas as pd
a = [1, 7, 2]
myvar = pd.Series(a)   ##o/p is 0 1
a = [1, 7, 2]                  1 7
print(myvar)                   2 2
```

6)in Pandas are usually multi-dimensional tables, called DataFrames.i,e,,both rows and column

7)data with wrong format make it difficult to analys,therefore
step1:-we need to convert all format unique i,e,,to convert date column into same by the code:-to_datetime

```
import pandas as pd
df = pd.read_csv('data.csv')
df['Date'] = pd.to_datetime(df['Date'])    ##but this method will not give full
```

```
accuracy
print(df.to_string())

8)correct way is:-add the comment "dropna"
import pandas as pd
df = pd.read_csv('data.csv')
df['Date'] = pd.to_datetime(df['Date'])
df.dropna(subset=['Date'], inplace = True)   ##it give accuracy
print(df.to_string())

9)Cleaning wrong data:-
"Wrong data" does not have to be "empty cells" or "wrong format", it can just be
wrong, like if someone registered "199" instead of "1.99".example If you take a look
at our data set, you can see that in row 7, the duration is 450, but for all the
other rows the duration is between 30 and 60.

Example problems:-Specify to limit of selecting for over predict range i,e,,consider
duration column in that we put the limit of (If the value is higher than 120, set it
to 120)
for this above example programme was:- import pandas as pd
                                       df = pd.read_csv('data.csv')
                                        for x in df.index:
                                               if df.loc[x, "Duration"] > 120: ##it
tell compiler to compiler, if duratiom is >120;
                                                  df.loc[x, "Duration"] = 120  ##if
it is >120 then it makes that >120 values equal to/assign value 120
                                       print(df.to_string())

*Another way of doing this is to remove the rows that contains wrong data:-using
only "drop" not "dropna" bcz here we don't have Nul_value(na)
import pandas as pd
df = pd.read_csv('data.csv')
for x in df.index:
  if df.loc[x, "Duration"] > 120:  ##if compiler find >120 values next would happens

    df.drop(x, inplace = True)     ##>120 values found it will delete using keyword
dro1
print(df.to_string())

10)Removing Duplicates:-Duplicate rows are rows that have been registered more than
one time.
*By using "duplicated()" The duplicated() method returns a Boolean values for each
row
 Example:-Returns True for every row that is a duplicate, othwerwise False
i,e,,import pandas as pd
                                                                        df =
pd.read_csv('data.csv') ##o/p gives False,if the index don't had duplicate i,e0
False

print(df.duplicated())
```

*By using "drop_duplicates()" removes duplicate valuse instaed of duplicate() does
Example:-import pandas as pd
        df = pd.read_csv('data.csv')
        df.drop_duplicates(inplace = True)   ####Notice that duplicate row has been removed from the result
        print(df.to_string())

9)In pandas:-"std()" is used to get Standard deviation
        print(df.std())is the code
10)cov:-Covariance of value

__