In this project we will consolidate the concepts of Data Analysis using various python libraries . We are provided with CSV of two datasets on which we will be performing following analysis –

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

df=pd.read\_csv(‘NCHS\_-\_Leading\_Causes\_of\_Death\_\_United\_States.csv’)

df​.head(2)

df1 =pd.read\_csv('NCHS\_-\_Drug\_Poisoning\_Mortality\_by\_State\_\_United\_States.csv')

df1​.head(2)

1. Get the Metadata from the above files.

2. Get the row names from the above files.

3. Change the column name from any of the above files.

4. Change the column name from any of the above file and store the changes made permanently.

5. Change the names of multiple columns.

6. Arrange values of a particular column in ascending order.

7. Arrange multiple column values in ascending order.

8. Make state as the first column of dataframe.

9. Get the column array using a variable.

10. Get the subset rows 11, 22, 33,44,55.

11. Get the subset rows excluding 11, 22, 33.44 and 55

teens=pd.read\_csv('NCHS\_\_Teen\_Birth\_Rates\_for\_Females\_by\_Age\_Group\_\_Race\_\_and\_Hispanic\_Origin\_\_United\_States.csv')

pregnancy=pd.read\_csv('NCHS\_\_Pregnancy\_Rates\_\_by\_Age\_for\_Hispanic\_Women\_\_United\_States\_\_1990-2010.csv')

Drugs=pd.read\_csv('NCHS\_-\_Drug\_Poisoning\_Mortality\_by\_County\_\_United\_States.csv')

mortality=pd.read\_csv('NCHS\_-\_Infant\_Mortality\_Rates\_\_by\_Race\_\_United\_States\_\_1915-2013.csv')

12. Join teens to pregnancy, keeping all rows from transactions and only matching rows from teens(left join)

# 13. Which pregnancy have a Race Ethnicity not in teens?

14. Join teens to pregnancy, keeping only rows from pregnancy and teens that match via Race Ethnicity(inner join)

15. Join teens to pregnancy, displaying all matching rows AND all non-matching rows (full outer join)

16.Determine different Age Groups marrital Status on same year

17. Test to see if we can drop columns

18. **Code with Output :**

my\_columns = list(data.columns)

my\_columns

['Year\_x',

'Race Ethnicity',

'Marrital Status',

'All pregnancies',

'Live birth',

'Year\_y',

'Age Group',

'Birth Rate']

list(data.dropna(thresh=int(data.shape[0] \* .9), axis=1).columns)

['Year\_x',

'Race Ethnicity',

'Marrital Status',

'All pregnancies',

'Live birth']

missing\_info = list(data.columns[data.isnull().any()])

missing\_info

['Year\_y', 'Age Group', 'Birth Rate']

**//for col in missing\_info:**

num\_missing = data[data[col].isnull() == True].shape[0]

print('number missing for column {}: {}'.format(col, num\_missing))

number missing for column Year\_y: 78

number missing for column Age Group: 78

number missing for column Birth Rate: 78

for col in missing\_info:

percent\_missing = data[data[col].isnull() == True].shape[0] / data.shape[0]

print('percent missing for column {}: {}'.format(

col, percent\_missing))

**Output of percentage missing data:**

percent missing for column Year\_y: 0.4105263157894737

percent missing for column Age Group: 0.4105263157894737

percent missing for column Birth Rate: 0.4105263157894737

19.Handling missing values using scikit learn