**Malnutrition, Obesity and Physical Inactivity Analysis and Surveillance**

CIS 5810

Information Systems

Project in Python

Guided by Professor Dr. Shilpa Balan

Shanmathi Arul Murugan

CIN: 306594267

Kaushik Sridharan

CIN: 306611492

**A. DATA SET URL’S AND DATA SET DESCRIPTION**

**DATASET SOURCE URL:**

<https://catalog.data.gov/dataset/nutrition-physical-activity-and-obesity-youth-risk-behavior-surveillance-system>

**DATA SET DESCRIPTION:**

This dataset includes data on adolescent's diet, physical activity, and weight status from Youth Risk Behavior Surveillance System acquired from the government website data.gov. It is published by the ‘Centers for Disease Control and Prevention’. This data is used for DNPAO's Data, Trends, and Maps database, which provides national and state specific data on obesity, nutrition, physical activity, and breastfeeding. The dataset contains about 31 columns and about 30,500 unique values(rows) for each column.

The dataset contains the values ranging from the year 2001 to 2015. It also contains the geological information of the affected population across the different states of Unites States of America along with their geospatial locations. This dataset categorizes the affected population into five different classes based on their cause of illness. The classes are as follows: fruits and vegetables behaviors, obesity and weight status, sugar drinks behavior, television viewing behavior and physical activity behaviors. This dataset also contains various questions regarding to the percentage of students in grade 9-12 who have a set of behaviors such as high consumption of beverages, having no or less vegetables and fruits in a day, less participation in physical activities, etc. The dataset also numeric values such as data value, confidence limits and sample size which specifies the disposition of illness over the population in a location.

The dataset can further be classified based of categories such as gender and race/ethnicity. Based on the above attributes two different stratifications have been gained. The stratification varies between gender, race and student grades of the people suffering from the various illness such as Malnutrition, Obesity and Physical Inactivity. With the help of python, we can now review the cause of youth risk factors by comparing the types of behaviors which causes obesity and malnutrition. We can also analyze if the specific behavior is related to gender, data rate and sample size.

**B. DATA CLEANING:**

**DATA CLEANING 1:** Removal of Null values in the columns

**PYTHON CODE:**

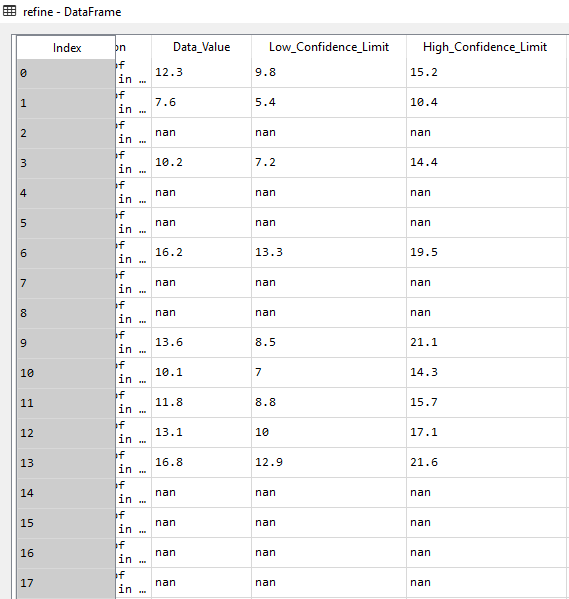
**#Implementation of DATAFRAME**

refine1=refine.dropna(subset=['Data\_Value','Low\_Confidence\_Limit'])

refine1=pd.DataFrame(refine1)

print(refine1)

**PRE-CLEANING:**



**POST-CLEANING:**



**EXPLANATION:**

The dataset obtained contains columns such as Data\_Value, Low\_Confidence\_Limit and High\_Confidence\_Limit which contains empty values. To improve the quality of the data on which the analysis must be done, these values have been removed from the dataset with the help of an inbuilt function *‘dropna’*. The category **Pandas Data Frame** have been used in this data cleaning division.

**DATA CLEANING 2:** Providing relevant values to the misfed Values

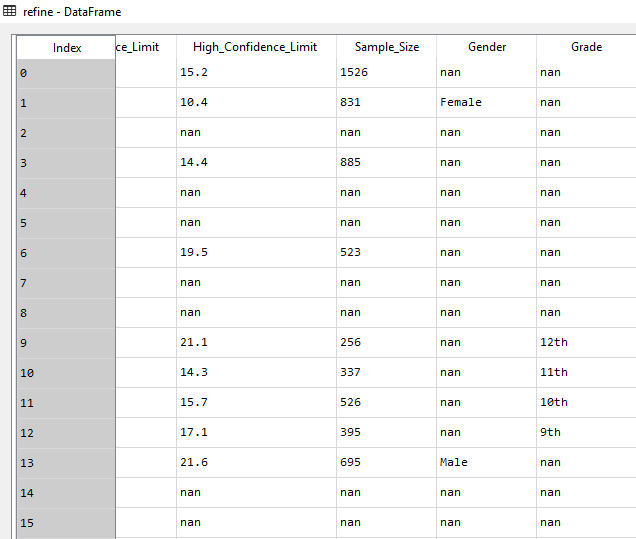
**PYTHON CODE:**

refine1[['Gender']]=refine1['Gender'].fillna('Do not wish to Disclose')

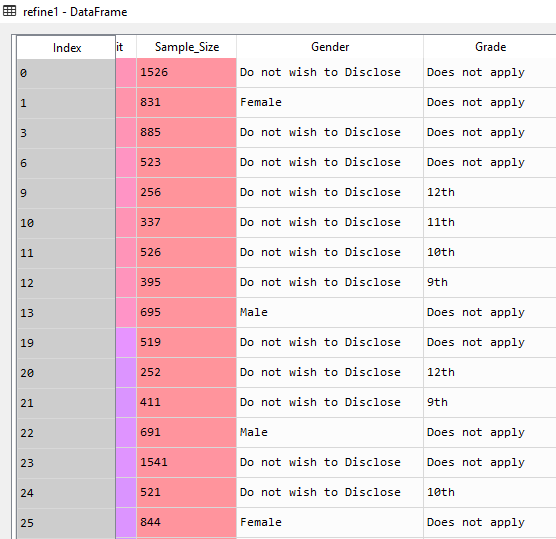
refine1[['Grade']]=refine1['Grade'].fillna('Does not apply')

print(refine1)

**PRE-CLEANING:**



**POST-CLEANING:**



**EXPLANATION:**

The dataset categorizes the affected people in terms of their Gender and the Grade of students who are among the affected population. In the Gender column, the misfed null values are replaced with a relevant value ‘Do not wish to Disclose’. Since only a certain number of population are students, the Grade column for other people are empty. Hence, those columns are filled with a value called ‘Does not apply’ with the help of an inbuilt function *‘fillna’*.

**DATA CLEANING 3:**

**PYTHON CODE:**

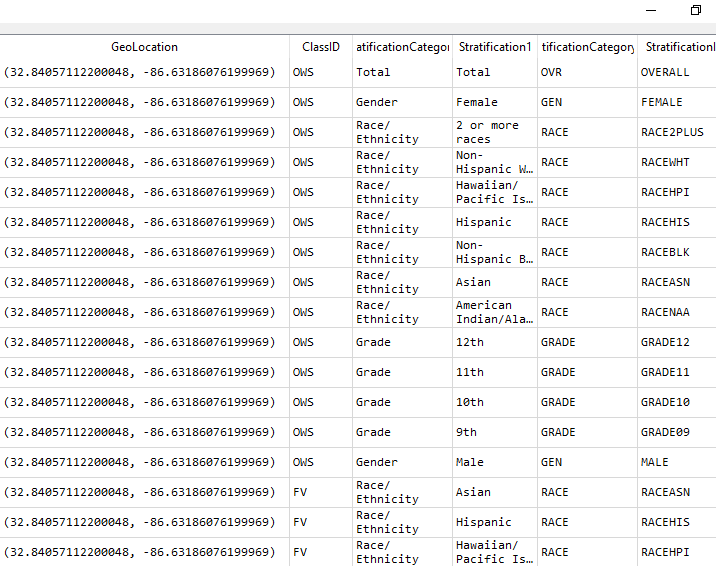
**#Implementation of STRINGS**

refine1['GeoLocation']=refine1['GeoLocation'].str.strip('()')

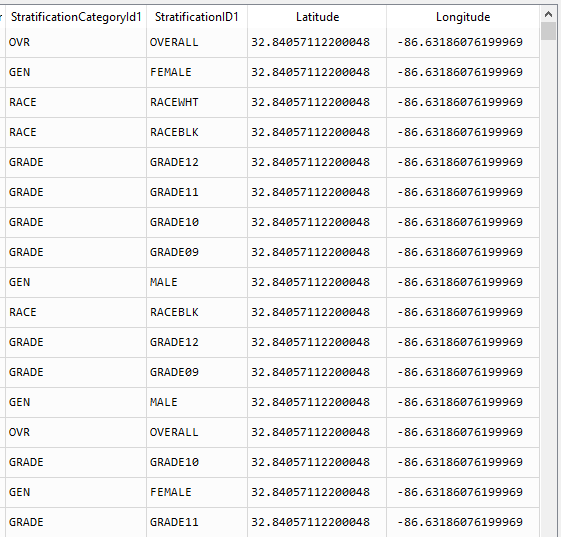
refine1[['Latitude','Longitude']]=refine1['GeoLocation'].str.split(',',expand=True)

print(refine1)

**PRE-CLEANING:**



**POST-CLEANING:**



**EXPLANATION:**

In the dataset downloaded from data.gov, the column Geolocation contains both the values of Latitude and Longitude together. Using the **String** functions *‘strip’* and *‘split’* the column is subdivided into two columns as Latitude and Longitude respectively to attain a clear understanding of the data.

**ADDITIONAL:** Removal of columns that are irrelevant to the analysis

**PYTHON CODE:**

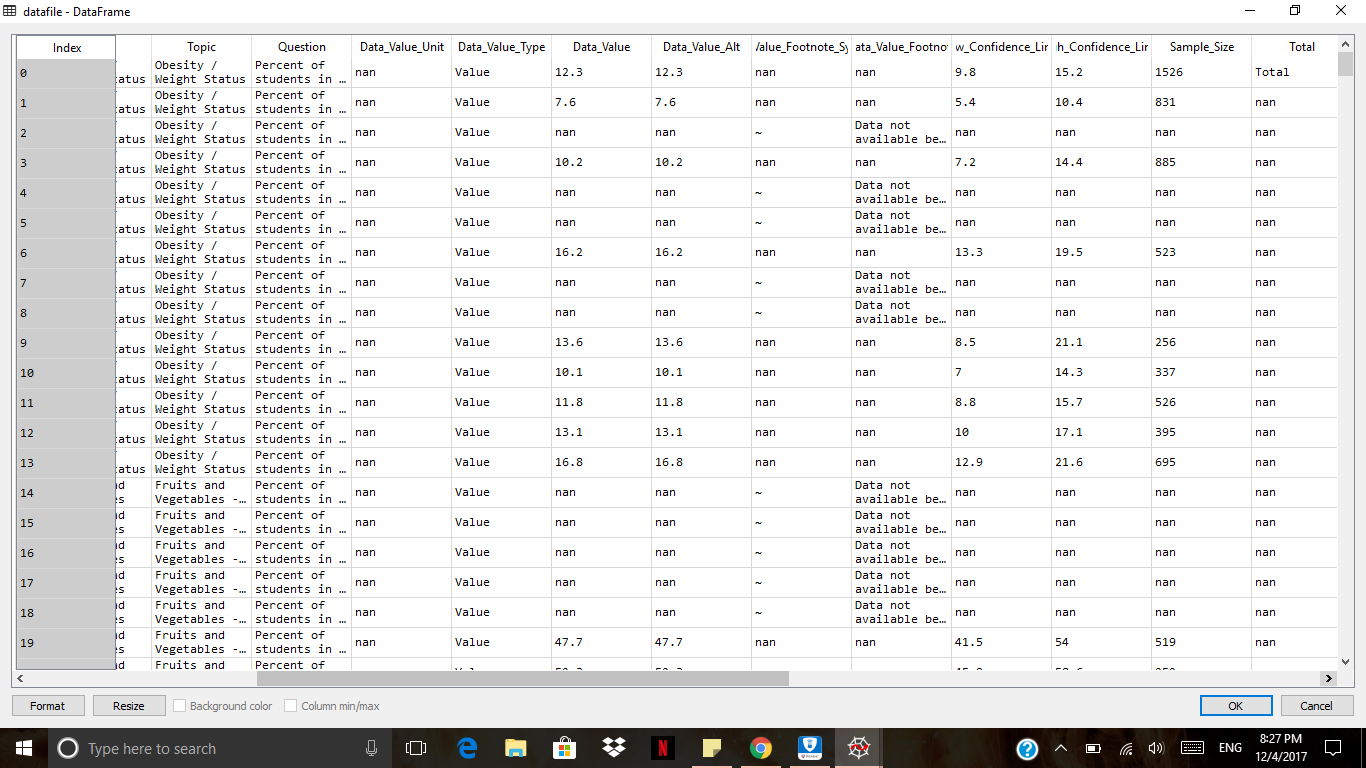
**#Implementation of LIST, DATAFRAME**

refine=datafile.drop(['YearEnd','Datasource','Data\_Value\_Unit','Data\_Value\_Type','Data\_Value\_Alt','Data\_Value\_Footnote\_Symbol','Data\_Value\_Footnote','Total','TopicID','DataValueTypeID','QuestionID','LocationID'],axis=1)

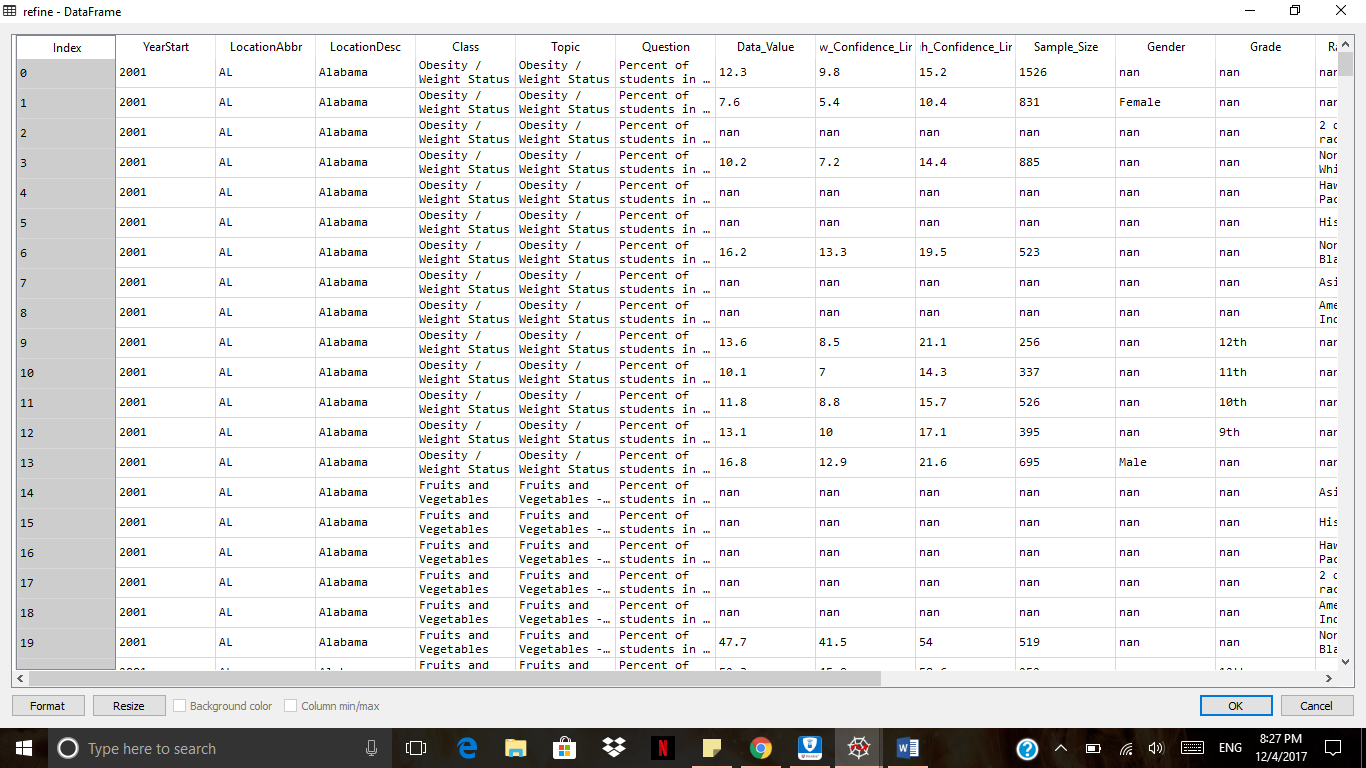
refine=pd.DataFrame(refine)

print(refine)

**PRE-CLEANING:**



**POST-CLEANING:**



**EXPLANATION:**

The dataset contains columns that either contain duplicate values and null values which are not required to perform analysis. The columns YearEnd, Datasource, Data\_Value\_Unit, Data\_Value\_Type, Data\_Value\_Alt, Data\_Value\_Footnote\_Symbol, Data\_Value\_Footnote, Total, TopicID, DataValueTypeID, QuestionID and LocationID have been removed with the help of the inbuilt function drop.

**C. SUMMARY STATISTICS:**

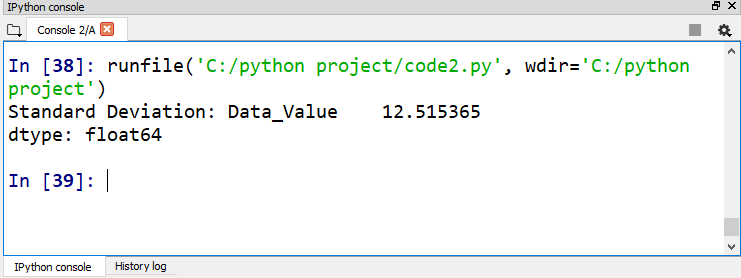
**STANDARD DEVIATION:**

**PYTHON CODE:**

sd=datafile1[['Data\_Value']].std()

print("Standard Deviation:",sd)

**OUTPUT:**



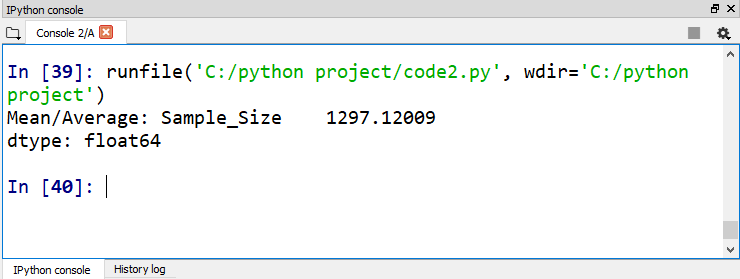
**MEAN/AVERAGE:**

**PYTHON CODE:**

avg=datafile1[['Sample\_Size']].mean()

print("Mean/Average:",avg)

**OUTPUT:**



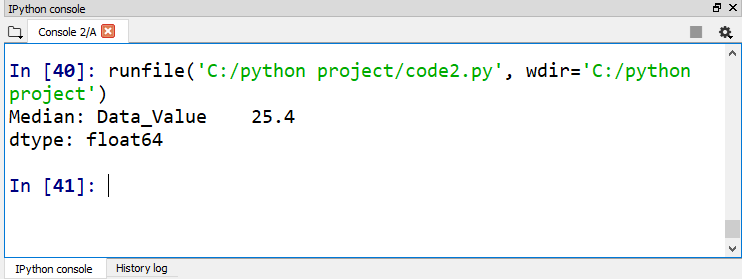
**MEDIAN:**

**PYTHON CODE:**

med=datafile1[['Data\_Value']].median()

print("Median:",med)

**OUTPUT:**



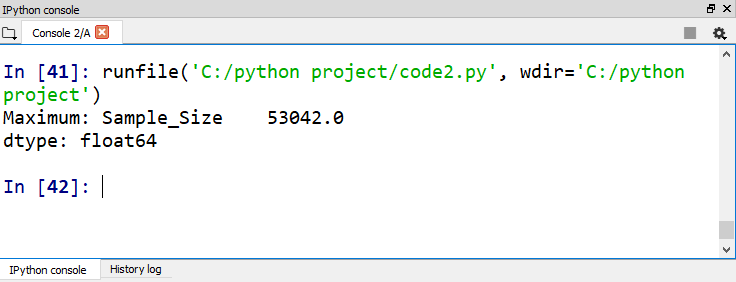
**MAXIMUM:**

**PYTHON CODE:**

maxi=datafile1[['Sample\_Size']].max()

print("Maximum:",maxi)

**OUTPUT:**



**D. ANALYSIS & VISUALIZATIONS:**

**ANALYSIS 1: Which class has caused the illness in most of population?**

**PYTHON CODE:**

import matplotlib.pyplot as plt

import pandas as pd

from pylab import \*

vis=pd.read\_csv('post\_refinement.csv',delimiter='|')

vis=pd.DataFrame(vis)

vis1=vis[['Topic']]

new = vis1['Topic'].value\_counts()

cnt = new.iloc[0:].tolist()

figure(1, figsize=(3,3))

**#Implemenation of LIST**

labels = ['Obesity / Weight Status','Fruits and Vegetables - Behavior','Physical Activity - Behavior','Television Viewing - Behavior','Sugar Drinks - Behavior']

**#Implementation of TUPLES**

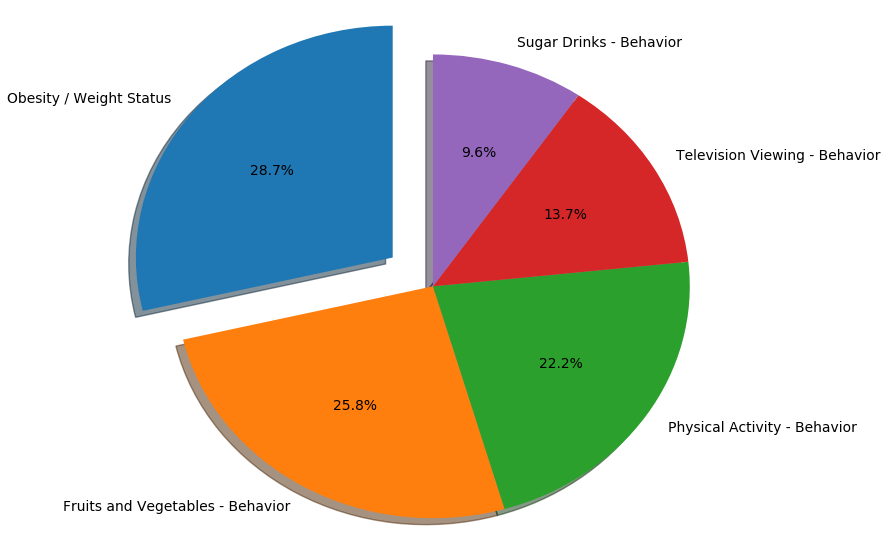
explode=(0.2, 0, 0, 0,0)

pie(cnt, labels=labels,explode=explode,

autopct='%1.1f%%', shadow=True, startangle=90)

plt.show()

**VISUALIZATION 1:**



**EXPLANATION:**

From the above Pie Chart, it can be observed that most of the population who are affected are suffering from Obesity or their weight status. To perform this analysis two categories have been used, namely **List and Tuples.** The labels for the Pie chart are provided with the help of List and Tuples are used for the Explode function which will slice out the highest percentage from the pie. It can also be analyzed that Fruits and Vegetables behavior is the major cause for people suffering from malnutrition.

**ANALYSIS 2: How has the trend of Data Values varied over the year?**

**PYTHON CODE:**

import matplotlib.pyplot as plt

import pandas as pd

vis=pd.read\_csv('post\_refinement.csv',delimiter='|')

vis=pd.DataFrame(vis)

vs = vis[['Data\_Value','YearStart']]

vs1 = vs.groupby('YearStart')['Data\_Value'].mean()

print(vs1)

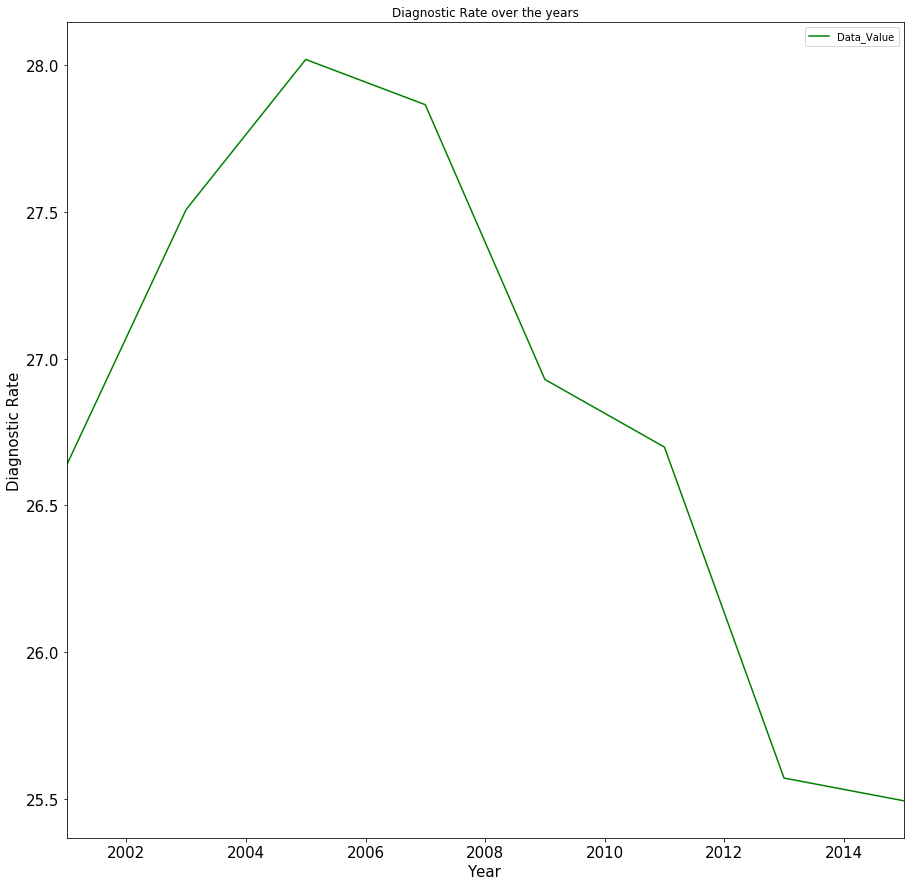
ax=vs1.plot(kind='line', title="Diagnostic Rate over the years", figsize=(15,15), legend=True, fontsize=15, color='green')

ax.set\_xlabel("Year", fontsize=15)

ax.set\_ylabel("Diagnostic Rate", fontsize=15)

plt.show()

**VISUALIZATION 2:**

****

**EXPLANATION:**

The above visualization is line chart that displays the trend of diagnostic rate or data value rate from the year 2001 to 2015. It can be inferred that in the year 2005, the data value is higher when compared to the other years. The number of people suffering from obesity and malnutrition have been reportedly high from the year 2005 to 2007. But it can also be seen that the data value has gradually decreased over the years and it has been the least in the year 2015 due to advance in the healthcare system.

**ANALYSIS 3: What is the classification of classes based on Gender?**

**PYTHON CODE:**

import matplotlib.pyplot as plt

import plotly as plot

import pandas as pd

import plotly.offline as offline

vis=pd.read\_csv('post\_refinement.csv',delimiter='|')

vis=pd.DataFrame(vis)

vs = vis[['Gender','Topic','Sample\_Size']]

vs3 = pd.DataFrame(vs.groupby(['Gender','Topic'])['Sample\_Size'].sum())

**#Implementation of FILE concept**

vs3.to\_csv('question3.csv',sep='|',encoding='utf-8')

vs3 = pd.read\_csv('question3.csv',delimiter='|')

ssdnd = vs3['Sample\_Size'][(vs3['Gender'] == "Do not wish to Disclose")]

ssfem = vs3['Sample\_Size'][(vs3['Gender'] == "Female")]

ssmal = vs3['Sample\_Size'][(vs3['Gender'] == "Male")]

**#Implementation of DICTIONARY**

data1 = [{"x": vs3['Topic'] , "y": ssfem, 'name':"Female", 'type':'bar'},

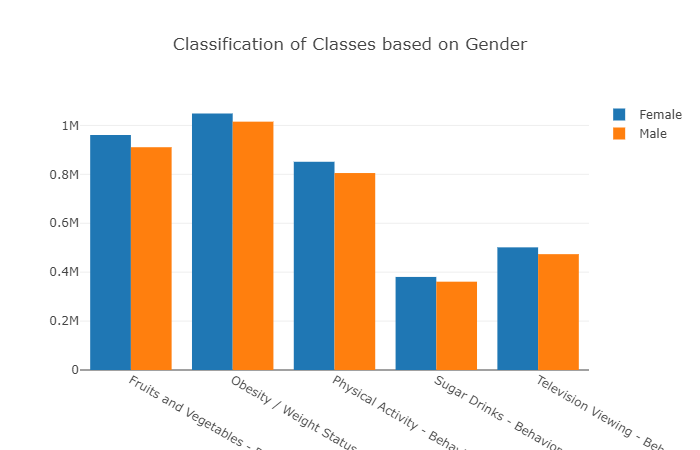
{"x": vs3['Topic'] , "y": ssmal, 'name':"Male", 'type':'bar' }]

layout1 = dict(title = 'Classification of Classes based on Gender')

fig = dict(data=data1, layout=layout1)

offline.plot(fig)

**VISUALIZATION 3:**



**EXPLANATION:**

The bar chart displays the classification of classes or the cause of diseases with respect to the gender. It can be observed that there is an only slight difference in values between male and female. The female gender has been majorly suffering from obesity and malnutrition. We can analyze that improper weight status being a major cause of the illness, followed by insufficient intake of fruits and vegetables in their regular diet and physical inactivity. An awareness can be created among the women to motivate them to indulge more in physical activity and following a healthy diet. The categories **File and Dictionary** have been used in this analysis. The inbuilt function to\_csv and read\_csv have been used to access the post\_refinement CSV file. Dictionary is used to specify the format of the bar chart.

**ANALYSIS 4: What is the comparison between Data value and Sample Size based on each State?**

**PYTHON CODE:**

import matplotlib.pyplot as plt

import plotly as plot

import pandas as pd

import plotly.offline as offline

import plotly.graph\_objs as go

**#Implementation of FUNCTION**

def stateanalysis(statevalue):

ss\_name = vs['Sample\_Size'][(vs['LocationDesc'] == statevalue)]

dv\_name = vs['Data\_Value'][(vs['LocationDesc'] == statevalue)]

trace = go.Scatter(

x = ss\_name,

y = dv\_name,

mode = 'markers',

marker=dict(size='16',color='crimson',colorscale='Viridis')

)

return(trace)

vis=pd.read\_csv('post\_refinement.csv',delimiter='|')

vis=pd.DataFrame(vis)

vs = vis[['LocationDesc','Data\_Value','Sample\_Size']]

statename = input("Enter the name of the state to analyze:")

trace = stateanalysis(statename)

data = [trace]

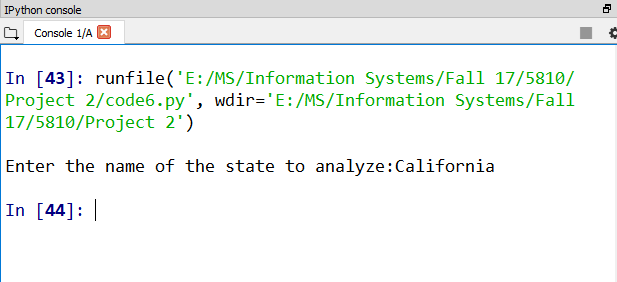
**#Implementation of DICTIONARY**

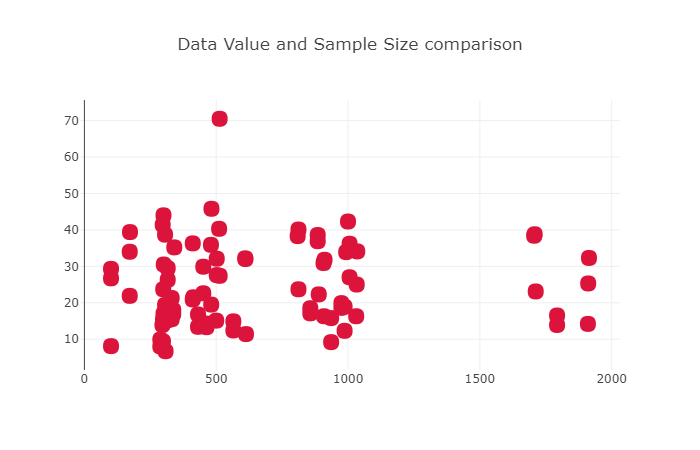
layout = dict(title = 'Data Value and Sample Size comparison')

fig = dict(data=data,layout=layout)

offline.plot(fig)

**VISUALIZATION 4:**



****

**EXPLANATION:**

The visualization is a scatter plot comparing the values of data value and sample size. The data value is plotted along Y- Axis and the sample size is plotted along X- Axis. The above program takes the value ‘State’ as an input and compares its data value and sample size of illness reported in their region. We can observe that there is not much variation in data rate except for one instance where the diagnostic rate is close to 70. Similarly, we can perform this analysis for different states and take necessary actions for the corresponding states to prevent obesity and malnutrition sickness. The concepts of **User-defined function and Dictionary** have been used in this analysis. The function is used to define the design of the scatter plot and its format.

**PYTHON FILES:**

* Data Cleaning



* Summary Statistics



* Analysis & Visualizations – 1



* Analysis & Visualizations – 2



* Analysis & Visualizations – 3



* Analysis & Visualizations – 4

