CSE 587 - Data Intensive Computing
Project Phase 1 Report

Mrunmayee Vijay Rane UBIT : mrane
Steve Thomas UBIT : stevetho

Problem Statement

Analyze various NHANES datasets and predict people who are at risk of developing diabetes and depression

a. Discuss the background of the problem leading to your objectives. Why is it a significant problem?

Diabetes and Depression affects around 8% and 10% of the world population. Diabetes is one of the leading causes of death in the US. Depression is one of the most common mental health issues in the US. Depression can lead to suicide. It is estimated that 60% of the suicides are committed by people who were suffering from mood disorders.

Economic costs of both these diseases are huge. The global economic cost of diabetes was estimated to $1.3 trillion dollars in 2015 and is projected to reach 2.1 trillion dollars by 2030. It was estimated that poor mental health costs the world economy 2.5 trillion dollars per year in 2010, this cost is expected to rise to 6 trillion dollars by 2030.

Diabetes and depression obviously are two of the top health issues in the world. Our project attempts to find how hours of sleep, blood pressure, hearing, body measurements, age, gender, and ethnicity are connected to diabetes and depression.

b. Explain the potential of your project to contribute to your problem domain. Discuss why this contribution is crucial ?

According to “Diabetes and Depression”; a technical paper by Antonio Campbayo and Antonio Lobo. A bidirectional relationship has been proposed between the depression and diabetes. There have been studies that explore the relationship between blood pressure and depression, sleep and depression, age and depression, gender, and depression. Some risk factors for diabetes include age, gender, ethnicity, and body measurements among others. There have been studies that have linked hearing loss to diabetes. In this project, we are aiming to analyze all the possible factors for cause and effect of diabetes and create a model that predicts the risk of a person suffering from depression and diabetes.

Data Sources

We have used National Health and Nutritional Examination Survey (NHANES) dataset released by Centers for Disease Control and Prevention (CDC). NHANES is a survey conducted by the National Center for Health Statistics, a unit of CDC. This survey uses interviews, examinations, and laboratory tests for building the dataset.

For our study, we have used demographics, audiometry, blood pressure, glycohemoglobin, plasma fasting glucose, diabetes, mental health - depression screener and sleep disorders datasets for years 2015 - 2016 and 2017 - March 2020.

Data Cleaning

1. **Drop irrelevant columns**

   Demographics dataset has 29 columns of which sequence number, age, gender and ethnicity are relevant to our project.

```
# load data
demographics_df=pd.read_sas('/content/drive/MyDrive/P_DEMO.XPT')
demographics_df.head()
```

	SEQN	SDDSRVYR	RIDSTATR	RIAGENDR	RIDAGEYR	RIDAGEMN	RIDRETH1	RIDRETH3	RIDEXMON	DMDBORN4	...	FIAINTRP	MIALANG	MIAPROXY	MIAINTRP	AIALANGA	WTINTPRP	WTMECPRP	S
0	109263.0	66.0	2.0	1.0	2.0	NaN	5.0	6.0	2.0	1.0	...	2.0	NaN	NaN	NaN	NaN	7891.762435	8.951816e+03	
1	109264.0	66.0	2.0	2.0	13.0	NaN	1.0	1.0	2.0	1.0	...	2.0	1.0	2.0	2.0	1.0	11689.747264	1.227116e+04	
2	109265.0	66.0	2.0	1.0	2.0	NaN	3.0	3.0	2.0	1.0	...	2.0	NaN	NaN	NaN	NaN	16273.825939	1.665876e+04	
3	109266.0	66.0	2.0	2.0	29.0	NaN	5.0	6.0	2.0	2.0	...	2.0	1.0	2.0	2.0	1.0	7825.646112	8.154968e+03	
4	109267.0	66.0	1.0	2.0	21.0	NaN	2.0	2.0	NaN	2.0	...	2.0	NaN	NaN	NaN	NaN	26379.991724	5.397605e-79	

5 rows × 29 columns

Sample of original demographics dataset

```
demographics_df=demographics_df[['SEQN','RIAGENDR','RIDAGEYR','RIDRETH3']]
demographics_df.head()
```

	SEQN	RIAGENDR	RIDAGEYR	RIDRETH3
0	109263.0	1.0	2.0	6.0
1	109264.0	2.0	13.0	1.0
2	109265.0	1.0	2.0	3.0
3	109266.0	2.0	29.0	6.0
4	109267.0	2.0	21.0	2.0

Sample of the demographics dataset after dropping irrelevant rows

2. **Datatype changes of column**

   The columns of depression screen questions were of the type float64 which was not required since the values were of type integers. This also made understanding the values difficult. All the 9 columns were converted into type int64.

```
mental_health_df.head()
```

	SEQN	DPQ010	DPQ020	DPQ030	DPQ040	DPQ050	DPQ060	DPQ070	DPQ080	DPQ090
0	109266.0	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79
1	109271.0	2.000000e+00	1.000000e+00	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	2.000000e+00	5.397605e-79	5.397605e-79
2	109273.0	2.000000e+00	2.000000e+00	2.000000e+00	2.000000e+00	2.000000e+00	2.000000e+00	2.000000e+00	1.000000e+00	5.397605e-79
3	109274.0	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79	5.397605e-79
4	109282.0	5.397605e-79	1.000000e+00	5.397605e-79	1.000000e+00	5.397605e-79	5.397605e-79	5.397605e-79	3.000000e+00	5.397605e-79

Before changing the datatype (float64)

```
# convert data type into int64
mental_health_df['DPQ010'] = mental_health_df['DPQ010'].astype(np.int64)
mental_health_df['DPQ020'] = mental_health_df['DPQ020'].astype(np.int64)
mental_health_df['DPQ030'] = mental_health_df['DPQ030'].astype(np.int64)
mental_health_df['DPQ040'] = mental_health_df['DPQ040'].astype(np.int64)
mental_health_df['DPQ050'] = mental_health_df['DPQ050'].astype(np.int64)
mental_health_df['DPQ060'] = mental_health_df['DPQ060'].astype(np.int64)
mental_health_df['DPQ070'] = mental_health_df['DPQ070'].astype(np.int64)
mental_health_df['DPQ080'] = mental_health_df['DPQ080'].astype(np.int64)
mental_health_df['DPQ090'] = mental_health_df['DPQ090'].astype(np.int64)
```

```
# sum the scores of each questions
cols = ['DPQ010','DPQ020','DPQ030','DPQ040','DPQ050','DPQ060','DPQ070','DPQ080','DPQ090']
mental_health_df['SCORE'] = mental_health_df[cols].sum(axis=1)
mental_health_df.head()
```

	SEQN	DPQ010	DPQ020	DPQ030	DPQ040	DPQ050	DPQ060	DPQ070	DPQ080	DPQ090	SCORE
0	109266.0	0	0	0	0	0	0	0	0	0	0
1	109271.0	2	1	0	0	0	0	2	0	0	5
2	109273.0	2	2	2	2	2	2	2	1	0	15
3	109274.0	0	0	0	0	0	0	0	0	0	0
4	109282.0	0	1	0	1	0	0	0	3	0	5

After changing the datatype to int64.

3. Removal of missing data

The audiometry dataset had 487 rows which were missing data out of 5147 rows. These rows were removed.

```
# number of rows with missing data
audiometry_df.isnull().any(axis=1).sum()
```

```
487
```

```
# total number of rows
audiometry_df.shape[0]
```

```
5147
```

Number of rows with missing data and total number of rows

```
# drop rows with missing data
audiometry_df=audiometry_df.dropna()
# total number of rows
audiometry_df.shape[0]
```

```
4660
```

Number of rows after removing rows with missing values

4. Rows concatenation

There are two blood tests that are used to determine if a person is suffering from diabetes. They are glycohemoglobin test (A1C test) and fasting blood sugar test. NHANES has one dataset each corresponding to each test. So, to combine two datasets, one dataset was concatenated to another after determining if a person was suffering from diabetes.

```
fasting_glucose_df.head()
```

	SEQN	Diabetes
0	109264.0	0
1	109271.0	0
2	109274.0	1
3	109277.0	0
4	109282.0	0

```
# number of rows
fasting_glucose_df.shape[0]
```

```
4744
```

Sample of fasting glucose dataset and number of rows in the dataset

```
aic_test_df.head()
```

	SEQN	Diabetes
0	109264.0	0
1	109266.0	0
2	109271.0	0
3	109273.0	0
4	109274.0	0

```
# number of rows
aic_test_df.shape[0]
```

```
9737
```

Sample of glycohemoglobin dataset and number of rows in the dataset

```
# concat two dataframes
frames = [aic_test_df, fasting_glucose_df]
diabetes_df = pd.concat(frames)
```

```
diabetes_df.head()
```

	SEQN	Diabetes
0	109264.0	0
1	109266.0	0
2	109271.0	0
3	109273.0	0
4	109274.0	0

```
# number of rows
diabetes_df.shape[0]
```

```
14481
```

Sample of combined dataset and number of rows in the dataset

5. Removing duplicate rows

After concatenating the two A1C and fasting blood sugar datasets, there were duplicates in the combined dataset. These duplicated were removed using sequence number.

```
# number of rows
diabetes_df.shape[0]
```

```
14481
```

Number of rows in combined dataset before removing duplicates

```
# drop duplicates
diabetes_df = diabetes_df.drop_duplicates('SEQN')
# number of rows
diabetes_df.shape[0]
```

```
9749
```

Number of rows in combined dataset after removing duplicates

6. Summation of values

Each column in the Mental Health - Depression Screener corresponds to a question in the Patient Health Questionnaire. Depending on the response, each question/column is assigned a value between 0 and 3. The sum of all columns is then taken to determine if the person suffers from depression.

```
#sum the scores of each questions
cols = ['DPQ010','DPQ020','DPQ030','DPQ040','DPQ050','DPQ060','DPQ070','DPQ080','DPQ090']
mental_health_df['SCORE'] = mental_health_df[cols].sum(axis=1)
mental_health_df.head()
```

	SEQN	DPQ010	DPQ020	DPQ030	DPQ040	DPQ050	DPQ060	DPQ070	DPQ080	DPQ090	SCORE
0	109266.0	0	0	0	0	0	0	0	0	0	0
1	109271.0	2	1	0	0	0	0	2	0	0	5
2	109273.0	2	2	2	2	2	2	2	1	0	15
3	109274.0	0	0	0	0	0	0	0	0	0	0
4	109282.0	0	1	0	1	0	0	0	3	0	5

Sample of the dataset after summing columns and saving total in new column 'Score'

7. Averaging the values

Two columns corresponding to diastolic and systolic readings were calculated by find the mean of all diastolic and systolic readings.

```
# create columns with averages of diastolic and systolic pressure
blood_pressure_df['Diastolic'] = blood_pressure_df[['BPXODI1','BPXODI2','BPXODI3']].mean(axis=1)
blood_pressure_df['Systolic'] = blood_pressure_df[['BPXOSY1','BPXOSY2','BPXOSY3']].mean(axis=1)
blood_pressure_df.head()
```

	SEQN	BPXOSY1	BPXODI1	BPXOSY2	BPXODI2	BPXOSY3	BPXODI3	Diastolic	Systolic
0	109264.0	109.0	67.0	109.0	68.0	106.0	66.0	67.000000	108.000000
1	109266.0	99.0	56.0	99.0	55.0	99.0	52.0	54.333333	99.000000
2	109270.0	123.0	73.0	124.0	77.0	127.0	70.0	73.333333	124.666667
3	109271.0	102.0	65.0	108.0	68.0	111.0	68.0	67.000000	107.000000
4	109273.0	116.0	68.0	110.0	66.0	115.0	68.0	67.333333	113.666667

Sample of dataset after finding average diastolic and systolic readings

8. Categorizing values in a column

The value obtained by the summing scores of mental health depression screener dataset is used to find out the depression levels of each person. If the score is between 0 and 4, it is determined as no depression, 5 to 9 is considered as mild, 10 to 14 is considered as moderate, 15 to 19 is considered as moderately severe and 20 to 27 is considered as severe depression. Each of these categories is assigned a number that is 0 for none, 1 for mild, 2 for moderate, 3 for moderately severe and 4 severe.

```
mental_health_df['DEPRESSION']=pd.cut(mental_health_df['SCORE'], bins=[0, 4, 9, 14, 19, 27], include_lowest=True, labels=['0', '1', '2', '3','4'])
mental_health_df.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
```

	SEQN	SCORE	DEPRESSION
0	109266.0	0	0
1	109271.0	5	1
2	109273.0	15	3
3	109274.0	0	0
4	109282.0	5	1

Sample of the dataset after categorizing the score

9. **Removing irrelevant data from columns**

In mental health depression screener dataset, rows with values 7 or 9 corresponds to 'refused' and 'don't know' respectively. These values cannot be considered while calculating the total score of the depression screener questions. All these values converted to None datatype and then removed.

```
# number of rows
mental_health_df.shape[0]
```

```
8302
```

Number of rows before removing 7 and 9 values.

```
# replace 7 and 9 with none and drop those rows
mental_health_df = mental_health_df.replace({7: None})
mental_health_df = mental_health_df.replace({9: None})
mental_health_df=mental_health_df.dropna()
mental_health_df.shape[0]
```

```
8276
```

Converting 7 and 9 to none datatype and dropping them. Number of rows after removing 7 and 9

10. **Merging of data**

In demographics dataset, people are categorized as Mexican Hispanic and non-Mexican Hispanic. We have merged these data together into a single category of Hispanic to find the effect of being to the Hispanic ethnicity.

```
# merge hispanics together
demographics_df['RIDRETH3'].replace({2: 1},inplace=True)
demographics_df.head()
```

	SEQN	RIAGENDR	RIDAGEYR	RIDRETH3
0	109263.0	1.0	2.0	6.0
1	109264.0	2.0	13.0	1.0
2	109265.0	1.0	2.0	3.0
3	109266.0	2.0	29.0	6.0
4	109267.0	2.0	21.0	1.0

Sample of dataset after merging Mexican and non-Mexican Hispanic category into a single category

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method or philosophy for data analysis that makes use of a range of tools (mainly graphical) to maximize understanding of a data collection; reveal the fundamental structure; extrapolate crucial variables; find abnormalities and outliers; test underlying hypotheses; create frugally priced models; and choose the best factor settings. We started with an aim to detect diabetes and its

factors causing elements. (National Health and Nutrition – NHANES) dataset. To analyze the dataset and extract relevant features related to Diabetes such as age, gender, ethnicity, sleeping hours, weight, height, BMI, diabetes, depression scores, blood pressure components i.e., diastolic, and systolic blood pressure. Additionally, we also acquire audiometry data such as high frequency hearing loss and speech frequency hearing loss. We also tried to take in account smoking data, physical activity data and Insulin data but dropped them as there was very little data available to draw inference between people who smoke and have diabetes.

We divide our exploratory data analysis process according to John Tukey's exploratory data analysis (EDA) process into statistical and graphical analysis.

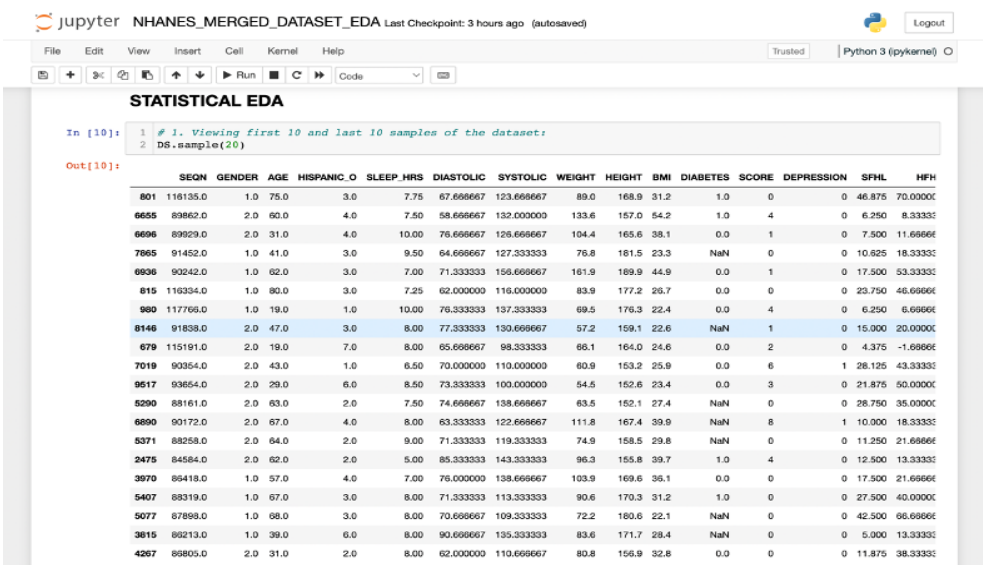
Statistical Exploratory Data Analysis:

1. Sample – *data.sample()*

Explanation:

In order to view the entire data, we used the sample() method with passing the number of items to sample in order to get a view of the first few and last few rows of the dataset.

Screenshot:



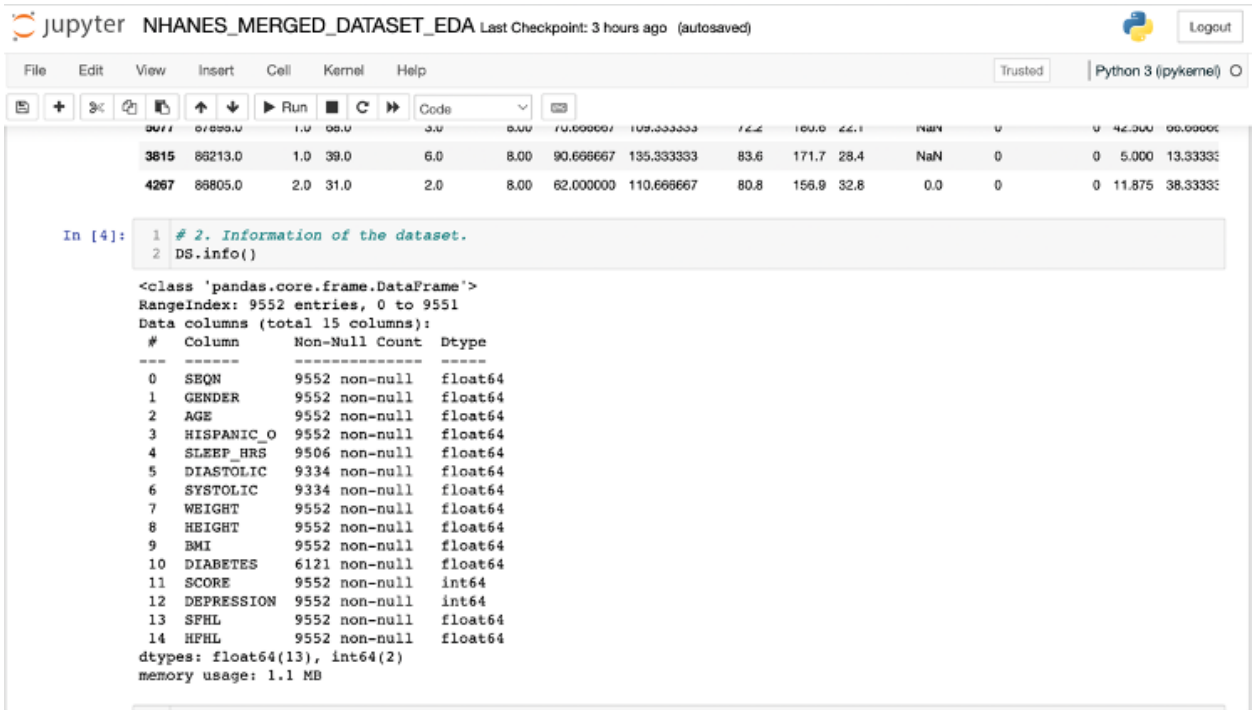
	SEQN	GENDER	AGE	HISPANIC_O	SLEEP_HRS	DIASTOLIC	SYSTOLIC	WEIGHT	HEIGHT	BMI	DIABETES	SCORE	DEPRESSION	SFHL	HFH
801	116135.0	1.0	75.0	3.0	7.75	67.666667	123.666667	89.0	168.9	31.2	1.0	0	0	46.875	70.00000
6655	89862.0	2.0	60.0	4.0	7.50	58.666667	132.000000	133.6	157.0	54.2	1.0	4	0	6.250	8.33333
6696	89929.0	2.0	31.0	4.0	10.00	76.666667	126.666667	104.4	165.6	38.1	0.0	1	0	7.500	11.66666
7865	91452.0	1.0	41.0	3.0	9.50	64.666667	127.333333	76.8	181.5	23.3	NaN	0	0	10.625	18.33333
6936	90242.0	1.0	62.0	3.0	7.00	71.333333	156.666667	161.9	189.9	44.9	0.0	1	0	17.500	53.33333
815	116334.0	1.0	80.0	3.0	7.25	62.000000	116.000000	83.9	177.2	26.7	0.0	0	0	23.750	46.66666
980	117766.0	1.0	19.0	1.0	10.00	76.333333	137.333333	69.5	176.3	22.4	0.0	4	0	6.250	6.66666
8146	91838.0	2.0	47.0	3.0	8.00	77.333333	130.666667	57.2	159.1	22.6	NaN	1	0	15.000	20.00000
679	115191.0	2.0	19.0	7.0	8.00	65.666667	98.333333	66.1	164.0	24.6	0.0	2	0	4.375	-1.66666
7019	90354.0	2.0	43.0	1.0	6.50	70.000000	110.000000	60.9	153.2	25.9	0.0	6	1	28.125	43.33333
9517	93654.0	2.0	29.0	6.0	8.50	73.333333	100.000000	54.5	152.6	23.4	0.0	3	0	21.875	50.00000
5290	88161.0	2.0	63.0	2.0	7.50	74.666667	138.666667	63.5	152.1	27.4	NaN	0	0	28.750	35.00000
6890	90172.0	2.0	67.0	4.0	8.00	63.333333	122.666667	111.8	167.4	39.9	NaN	8	1	10.000	18.33333
5371	88258.0	2.0	64.0	2.0	9.00	71.333333	119.333333	74.9	158.5	29.8	NaN	0	0	11.250	21.66666
2475	84584.0	2.0	62.0	2.0	5.00	85.333333	143.333333	96.3	155.8	39.7	1.0	4	0	12.500	13.33333
3970	86418.0	1.0	57.0	4.0	7.00	76.000000	138.666667	103.9	169.6	36.1	0.0	0	0	17.500	21.66666
5407	88319.0	1.0	67.0	3.0	8.00	71.333333	113.333333	90.6	170.3	31.2	1.0	0	0	27.500	40.00000
5077	87898.0	1.0	68.0	3.0	8.00	70.666667	109.333333	72.2	160.6	22.1	NaN	0	0	42.500	66.66666
3815	86213.0	1.0	39.0	6.0	8.00	90.666667	135.333333	83.6	171.7	28.4	NaN	0	0	5.000	13.33333
4267	86805.0	2.0	31.0	2.0	8.00	62.000000	110.666667	80.8	156.9	32.8	0.0	0	0	11.875	38.33333

2. Information – *data.info()*

Explanation:

In order, to get complete information about the data set such as number of columns, name of columns, total number of rows in every column, type of data contained in it, we used *data.info()*

Screenshot:

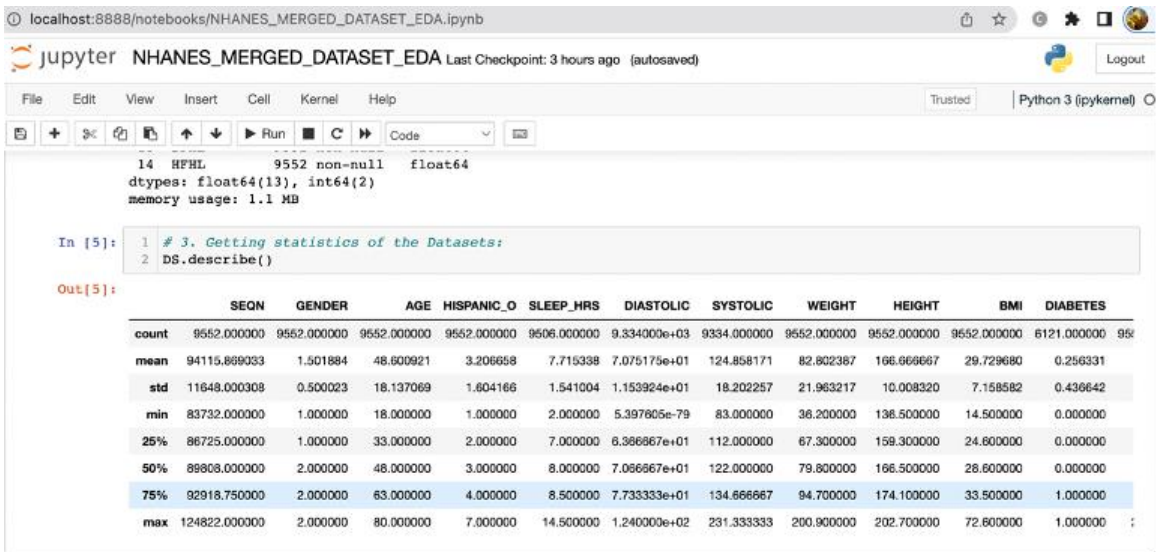


3. Describe – *data.describe()*

Explanation:

In order to get descriptive statistics of the entire dataset such as count, mean, standard deviation, minimum, maximum, 25%, 75% i.e., lower and upper percentiles and 50% i.e., median of data in each and every column. It summarizes the range of values in all columns as different columns have different ranges of measure. For e.g., blood pressure data such as systolic and diastolic has mmhg (millimeters of mercury) It also gives information about the shape of the distribution.

Screenshot:



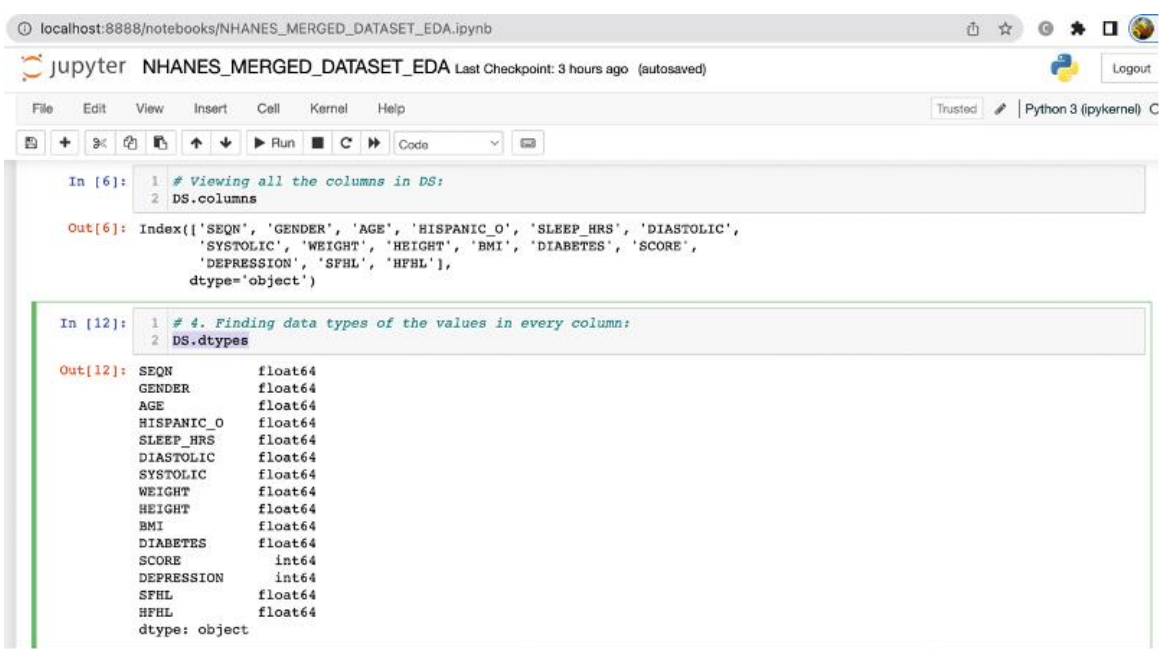
	SEQN	GENDER	AGE	HISPANIC_O	SLEEP_HRS	DIASTOLIC	SYSTOLIC	WEIGHT	HEIGHT	BMI	DIABETES
count	9552.000000	9552.000000	9552.000000	9552.000000	9506.000000	9.334000e+03	9334.000000	9552.000000	9552.000000	9552.000000	6121.000000
mean	94115.869033	1.501884	48.600921	3.206658	7.715338	7.075175e+01	124.858171	82.802387	166.666667	29.729680	0.256331
std	11648.000308	0.500023	18.137069	1.604166	1.541004	1.153924e+01	18.202257	21.963217	10.008320	7.158582	0.436642
min	83732.000000	1.000000	18.000000	1.000000	2.000000	5.397605e-79	83.000000	36.200000	138.500000	14.500000	0.000000
25%	86725.000000	1.000000	33.000000	2.000000	7.000000	6.366667e+01	112.000000	67.300000	159.300000	24.600000	0.000000
50%	89808.000000	2.000000	48.000000	3.000000	8.000000	7.066667e+01	122.000000	79.800000	166.500000	28.600000	0.000000
75%	92918.750000	2.000000	63.000000	4.000000	8.500000	7.733333e+01	134.666667	94.700000	174.100000	33.500000	1.000000
max	124822.000000	2.000000	80.000000	7.000000	14.500000	1.240000e+02	231.333333	200.900000	202.700000	72.600000	1.000000

4. Data types – *data.dtypes*

Explanation:

In order to find the datatypes of every column, we used data.dtypes(). It gives information about the data type of the value contained in every column. This dataset includes weight(kg), height(cm) or hours(hrs.), which can be integer or float values. In order to work with this dataset, it is essential to know its data types.

Screenshot:

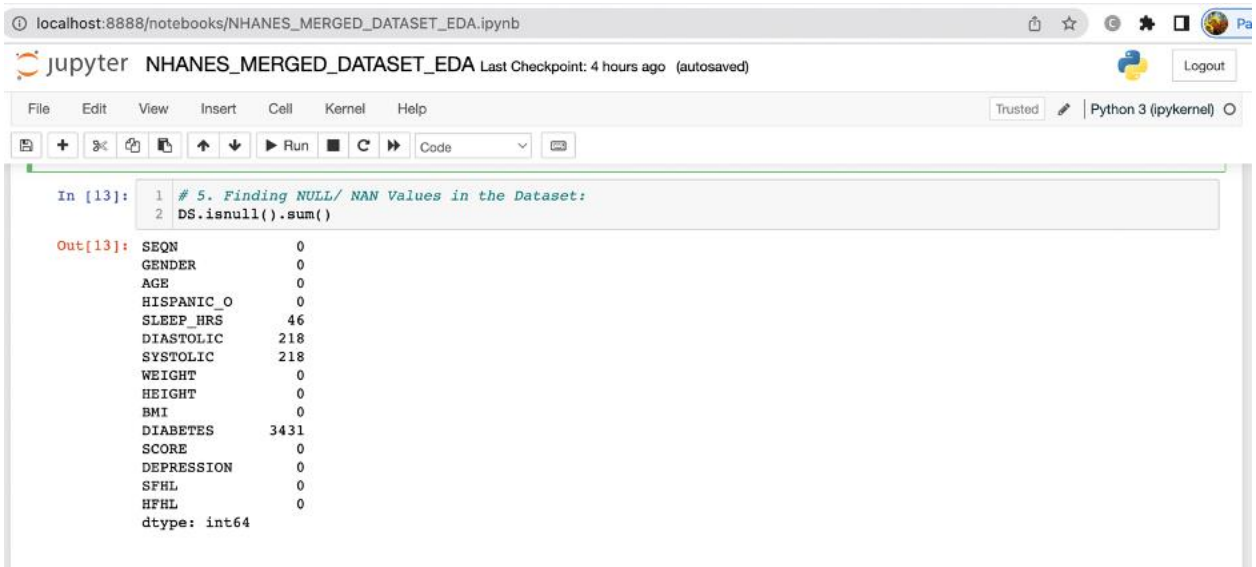


5. Summarization of the NaN values – *data.isnull().sum()*

Explanation:

To find if there's any missing values in the dataset, we first check if it contains any null value using isnull ()and then summarize it using sum(). Hence, it identifies the number of NaN values in each column of the dataset.

Screenshot:

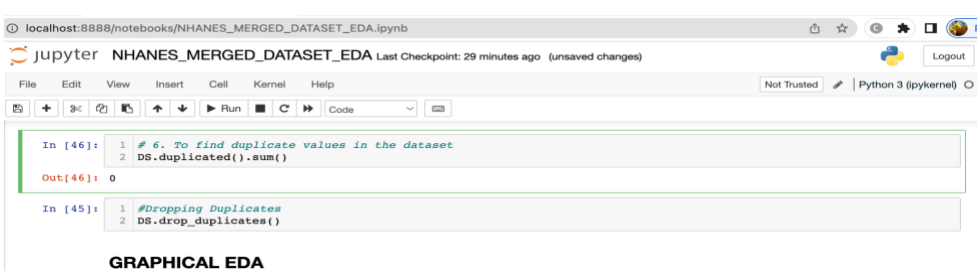


6. Duplicates - *data.duplicated().sum()*

Explanation:

In order to find duplicate values in the dataset, we used data.duplicates().sum(); which gives the count of the duplicate values in the dataset. Duplicates contaminate training or test data. Hence, it is essential to remove duplicate values.

Screenshot:



Graphical Exploratory Data Analysis:

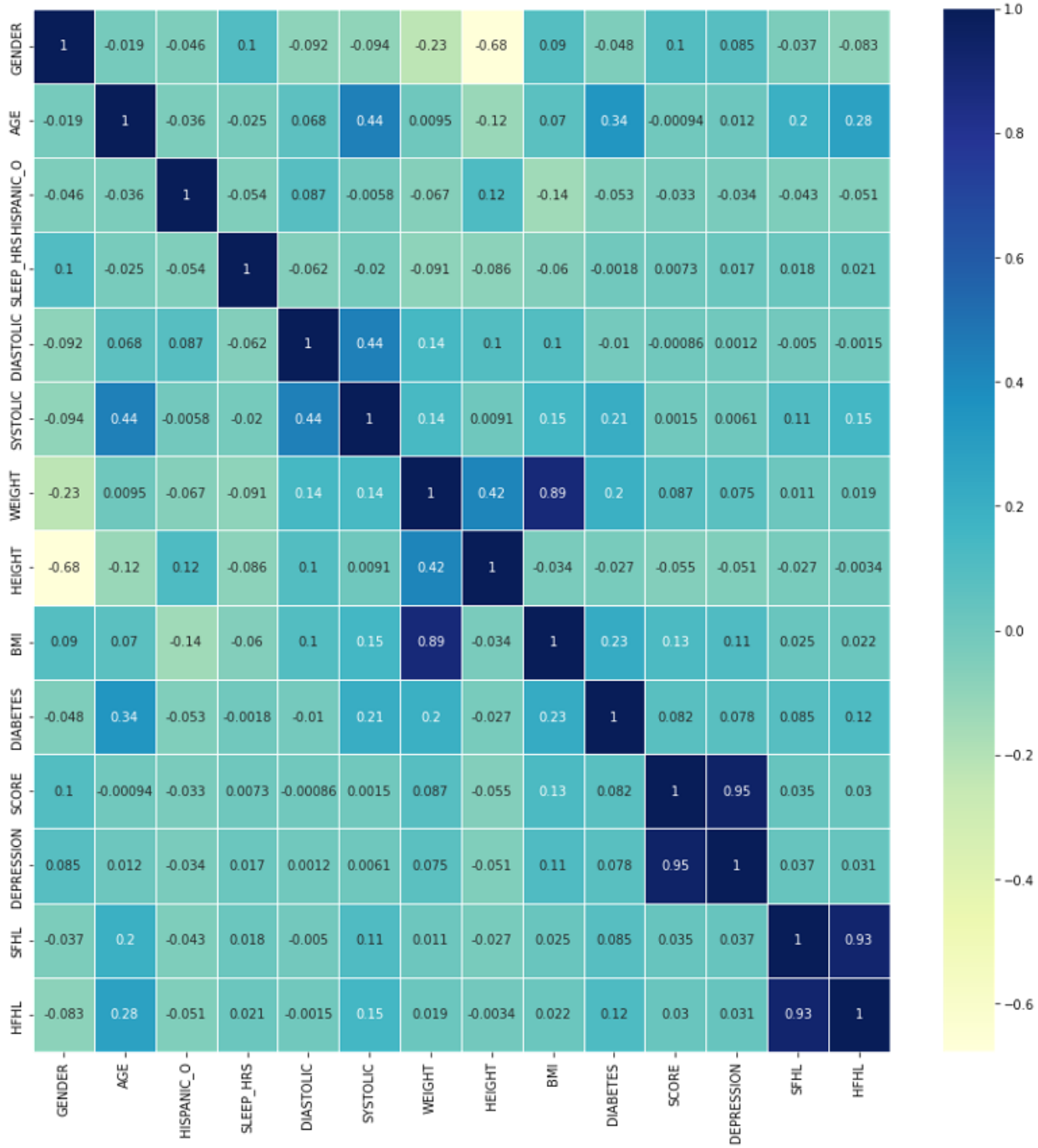
7. Correlation Matrix - *data.corr(args**))*

Explanation:

To understand the relationship between any two features in our data set, we used correlation matrix. According to this correlation matrix, we were able to derive some inferences such as the relationship between age and BMI(Body to Mass Index).

Ethnicity, height and diastolic ; sleep hours and genders; weight and diabetes; diabetes, age, and BMI; depression and BMI; Speech frequency hearing loss, High frequency hearing loss , systolic and age. Above, relationships can be used for phase 2 feature modeling.

Screenshot:

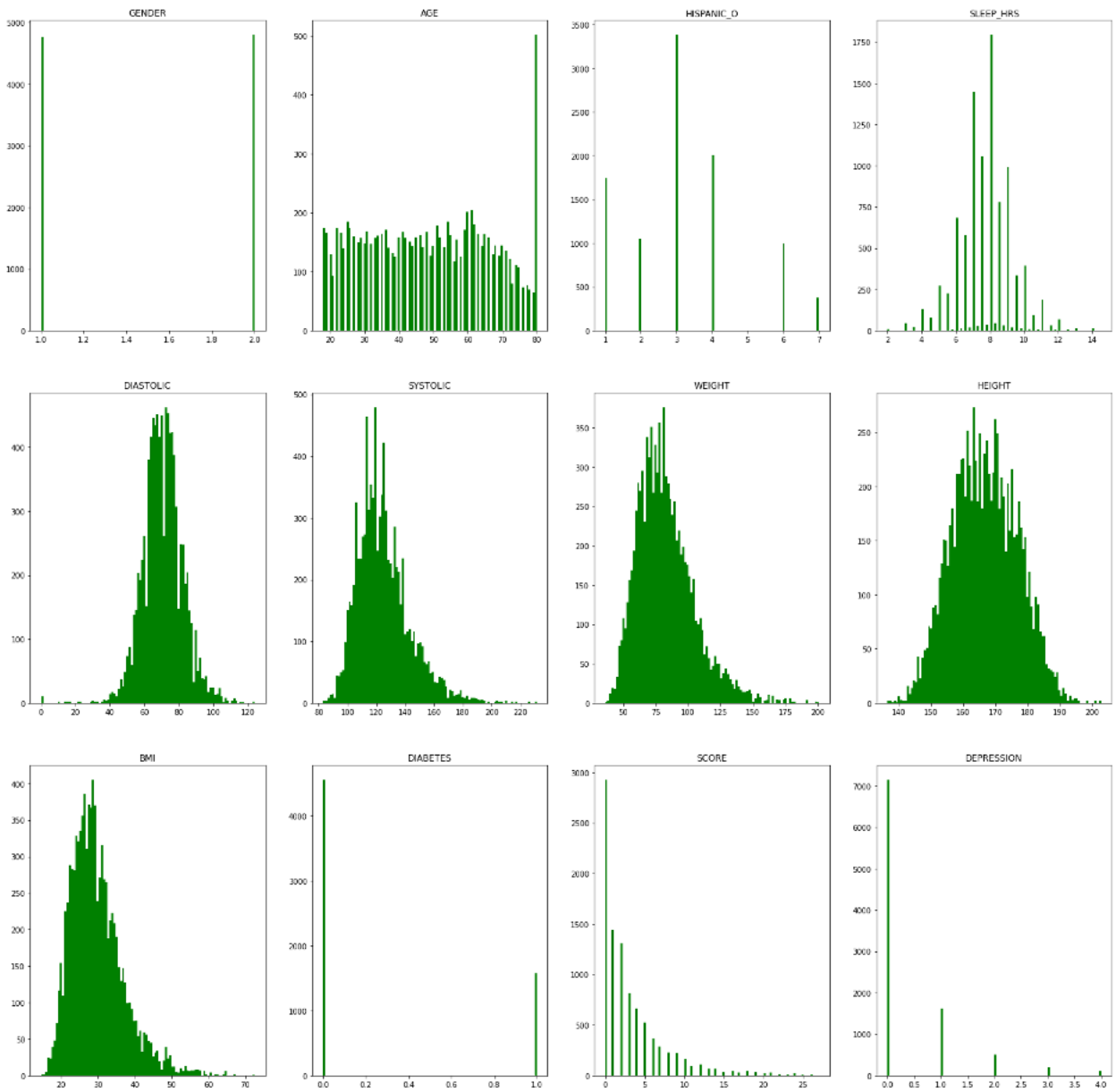

GENDER 1 -0.019 -0.046 0.1 -0.092 -0.094 -0.23 -0.68 0.09 -0.048 0.1 0.085 -0.037 -0.083
AGE -0.019 1 -0.036 -0.025 0.068 0.44 0.0095 -0.12 0.07 0.34 -0.00094 0.012 0.2 0.28
HISPANIC_O -0.046 -0.036 1 -0.054 0.087 -0.0058 -0.067 0.12 -0.14 -0.053 -0.033 -0.034 -0.043 -0.051
SLEEP_HRS 0.1 -0.025 -0.054 1 -0.062 -0.02 -0.091 -0.086 -0.06 -0.0018 0.0073 0.017 0.018 0.021
DIASTOLIC -0.092 0.068 0.087 -0.062 1 0.44 0.14 0.1 0.1 -0.01 -0.00086 0.0012 -0.005 -0.0015
SYSTOLIC -0.094 0.44 -0.0058 -0.02 0.44 1 0.14 0.0091 0.15 0.21 0.0015 0.0061 0.11 0.15
WEIGHT -0.23 0.0095 -0.067 -0.091 0.14 0.14 1 0.42 0.89 0.2 0.087 0.075 0.011 0.019
HEIGHT -0.68 -0.12 0.12 -0.086 0.1 0.0091 0.42 1 -0.034 -0.027 -0.055 -0.051 -0.027 -0.0034
BMI 0.09 0.07 -0.14 -0.06 0.1 0.15 0.89 -0.034 1 0.23 0.13 0.11 0.025 0.022
DIABETES -0.048 0.34 -0.053 -0.0018 -0.01 0.21 0.2 -0.027 0.23 1 0.082 0.078 0.085 0.12
SCORE 0.1 -0.00094 -0.033 0.0073 -0.00086 0.0015 0.087 -0.055 0.13 0.082 1 0.95 0.035 0.03
DEPRESSION 0.085 0.012 -0.034 0.017 0.0012 0.0061 0.075 -0.051 0.11 0.078 0.95 1 0.037 0.031
SFHL -0.037 0.2 -0.043 0.018 -0.005 0.11 0.011 -0.027 0.025 0.085 0.035 0.037 1 0.93
HFHL -0.083 0.28 -0.051 0.021 -0.0015 0.15 0.019 -0.0034 0.022 0.12 0.03 0.031 0.93 1
GENDER AGE HISPANIC_O SLEEP_HRS DIASTOLIC SYSTOLIC WEIGHT HEIGHT BMI DIABETES SCORE DEPRESSION SFHL HFHL
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6

8. Distribution Plotting - Histograms - *data.hist(args**))*

Explanation:

The histogram graphically shows features such as center (i.e., the location) of the data; spread (i.e., the scale) of the data; skewness of the data; presence of outliers; and presence of multiple modes in the data. These features provide strong indications of the proper distributional model for the data.
It helped us in visualizing and understanding the dataset in a better way in terms of training and testing phase.
E.g., We can infer from the histogram that the average height of a person is between 160 to 170cm. Average sleep hours are between 8 to 9 hours.

Screenshot :



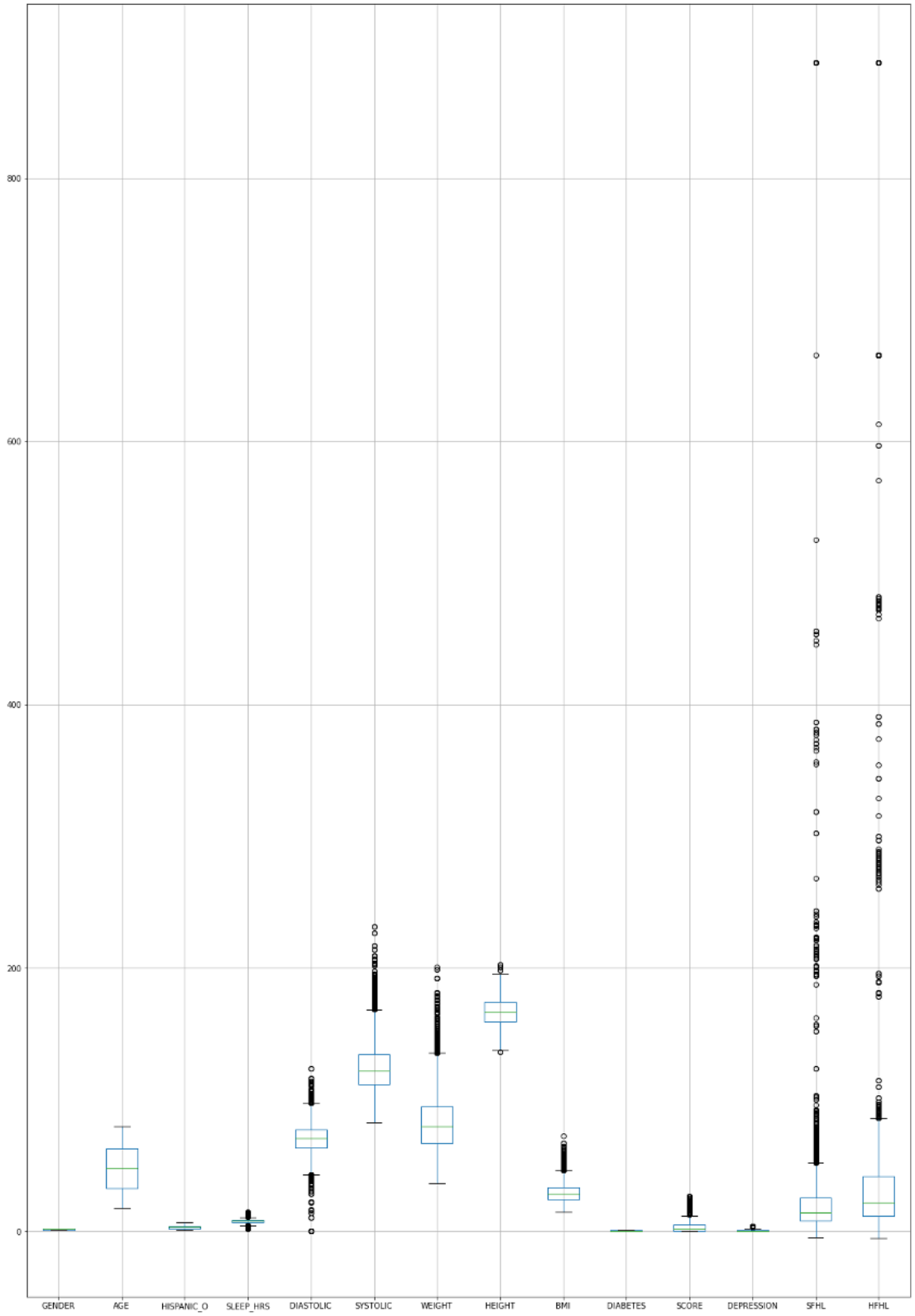
9. Box Plot – *data.boxplot(args**)*

Explanation:

Boxplots are a standardized way of displaying the distribution of data based on first quartile, median and third quartile. To understand outlier in the dataset, we use box plot. By default, the boxplot chart excludes any values outside of the 1.5 IQR (Interquartile Range) range, which eliminates the outlier values.

Box plot crucially identified the outliers in speech frequency hearing loss and high frequency hearing loss. This will help in dealing with the outliers in phase 2.

Screenshot:

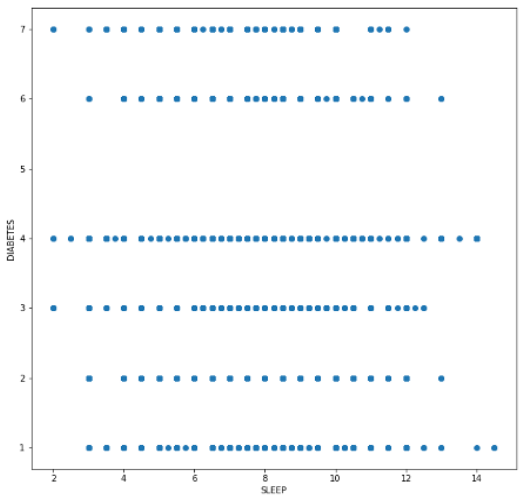

800
600
400
200
0
GENDER
AGE
HISPANIC_O
SLEEP_HRS
DIASTOLIC
SYSTOLIC
WEIGHT
HEIGHT
BMI
DIABETES
SCORE
DEPRESSION
SFHL
HFHL

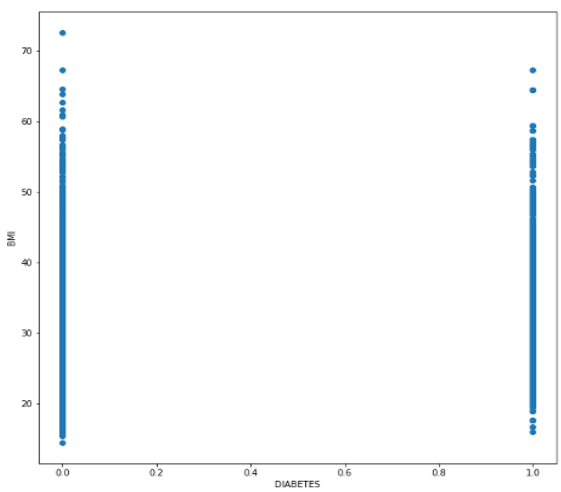
10. Scatter Plots:

Explanation:

Scatter plots help in determining the relationship between two variables. We tried to visualize the correlations which we found in the correlation matrix – heat map here. It is helpful in determining the relationship between sleep and diabetes and diabetes and BMI which will be helpful in the modeling phase.

Screenshot:





REFERENCES:

1. https://www.itl.nist.gov/div898/handbook/eda/section3/eda33.htm
2. https://pandas.pydata.org/docs/user_guide/visualization.html
3. https://seaborn.pydata.org/generated/seaborn.heatmap.html
4. https://pubmed.ncbi.nlm.nih.gov/24743941/
5. https://pubmed.ncbi.nlm.nih.gov/21052874/