This Document reports an Exploratory Data Analysis on a sample population with heart disease collected from Kaggle.com, provided by Mazaharul Hasnine Mirza. (2023). Heart Disease Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/5146672 (Mirza, 2023)

This Data Analysis was conducted using Python. The data was manipulated with Python in a Jupyter Notebook using VSCode. Several Open Source and publicly available Python Libraries where utilized to visualize, interpret and make calculations on the dataset (Rob Mulla, 2021). The modules used where:

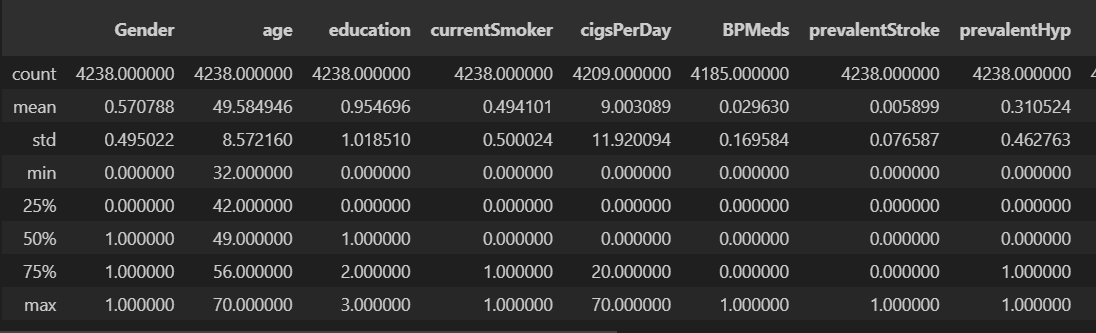
* Pandas – Data manipulation in ‘Pandas Dataframes’; https://pandas.pydata.org/
* Numpy – Linear algebra; https://numpy.org/
* Matplotlib – Data Visualization; https://matplotlib.org/
* SciPy – Statistical data visualization. Using the normal distribution and t-test modules <https://scipy.org/>
* Seaborn – Statistical data visualization <https://seaborn.pydata.org/>

The dataset was represented as a comma separated value file (csv).

The data considered 16 variables as possible causative factors for heart disease. The correlation between these variables was more easily found due to the large sample size of 4238 participants from differing backgrounds. Many columns existed and the data. Although the data required ‘cleaning’ before plotting, the data was still well-represented and readable.

A screenshot of a computer

Description automatically generated

**

*A screen shot of a black screen

Description automatically generated*

*Figure 1. Insights of the data with the pandas dataframe head() and describe() methods*

The Variables of the dataset, as retrieved, were interpreted as follows:

|  |  |  |
| --- | --- | --- |
| **Variable/Column** | **Interpretation:** | **DataType:** |
| age | The age of the individual | Integer |
| education | Highest level of education | String |
| currentSmoker | Whether the person smoked cigarettes | Boolean |
| cigsPerDay | How many cigarettes this person smokes in a day | Integer |
| BPMeds | Whether the person takes Blood Pressure Medication | Boolean |
| prevalentStroke | Whether the person has experienced brain stroke (reduced or blocked blood vessels) | Boolean |
| prevalentHyp | If the person has a high blood pressure relating to any of 3 causative factors: systolic blood pressure ≥140 mmHg , diastolic blood pressure ≥90 mmHg,  taking medication for hypertension (Riley, n.d.) | Boolean |
| diabetes | Whether this person has been diagnosed with any type of Diabetes | Boolean |
| totChol | Total Cholesterol present in blood measured in milligrams per deciliter (*Cholesterol Levels*, 2020) | Integer |
| sysBP | Systolic Blood Pressure. “The pressure exerted when the heartbeats” and “The normal range of systolic blood pressure should be 90 – 120 mm Hg.” (Byjus, n.d.) | Integer |
| diaBP | Diastolic Blood Pressure, as opposed to Systolic, is the pressure exerted when the heart is at rest. Normal range is 60-80 mm Hg. | Integer |
| BMI | Body mass index. A ratio of body weight to height | Float |
| heartRate | Heart rate measure in beats per minute | Integer |
| glucose | Glucose levels in the blood. Normal range is between 70 mg/dL and 100 mg/dL | Integer |
| Heart Stroke | Whether the person has experienced Heart Stroke (reduced or blocked blood vessels) | Boolean |

*Figure 2. Discerning the Datasets Variables*

After identifying the variables used in the dataset, the data was cleaned using various methods available in Python and Python modules (Rob Mulla, 2021). Certain functions and operations require the data types to be changed to integers only. All non-integer values were converted to a number that represented those values. These values included:

|  |  |  |
| --- | --- | --- |
| All Boolean values: | Gender: | Education: |
| 0 for no  1 for yes | Male: 0  Female: 1 | 0 for uneducated or ‘nan’ (null value)  1 for primary school  2 for graduate  3 for postgraduate |

*Figure 3. Translation of datatypes from strings (or objects) to integer values for analysis*

Other operations to ensure data consistency and usability were to remove outliers.

This was accomplished by calculating the statistical data of all variables in the dataset with pandas:

A black and white screen with numbers

Description automatically generated

*Figure 4. Average data of all variables of the dataset by the pandas describe() method*

The maximum values (refer to figure 4) where incredibly unusual and affected the mean values significantly.

To better represent the outliers of the entire dataset, Seaborn box plots where used to plot the columns that have numerical values to investigate outlier values and visualize them:

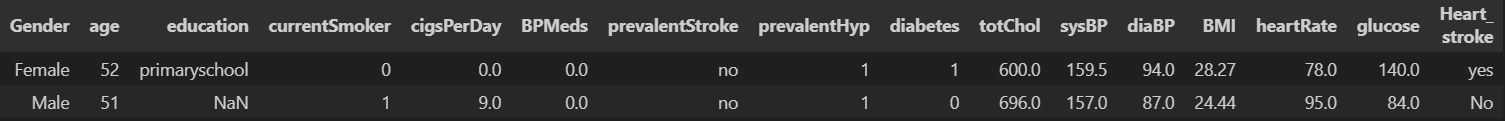
A graph with different colored lines and numbers

Description automatically generated with medium confidence

*Figure 5. Box plots of variables that had a high variance of numerical values.*

The Box Plot gave evidence that there were several outliers. Evidence was certain that the values 600 and 700 should be removed from totChol, as they may pose as extreme cases that should not necessarily represent the rest of the population. This was further investigated by inspecting the row of these outliers.

df[df['totChol'] > 500]



*Figure 6. totChol Outliars*

As indicated by the data anomalies, these individuals tended to score high in many variables, likely due to serious underlying conditions.

Likewise, another clear indication of an outlier is value 300 from sysBP. This individual also seemed to present as an extreme case, possibly following a severe incident due to high-scoring values:

df[df['sysBP'] > 270]

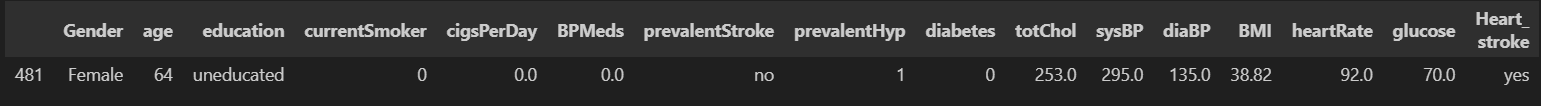
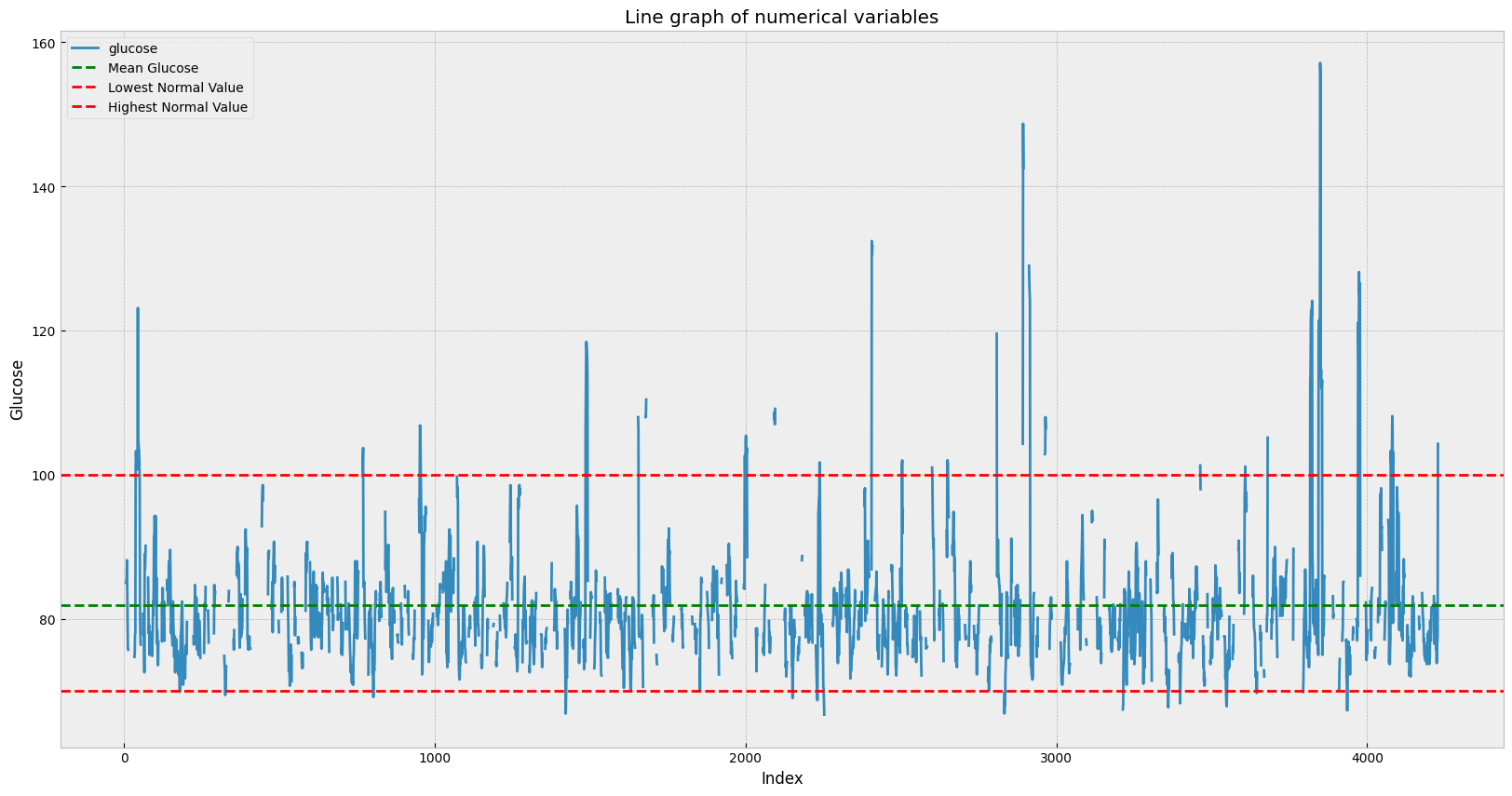


Figure 7. sysBP outliers

For other variables, it was dubious to discern the outlier values based on the box plot alone as they were more compact. It was also difficult to rule out high-scoring glucose values as glucose is volatile by nature, and the variable was observed to be prone to fluctuation compared to other variables.



*Figure 8. Line graph showing discrepancies between glucose levels of individuals*

Notice there were considerable amounts of people scoring above the threshold of what is considered a normal glucose level. These values are associated with the causative effect of other variables, proven by the brief analysis of a correlation graph using the Seaborn module in Python:

# Generate a correlation matrix of the DataFrame

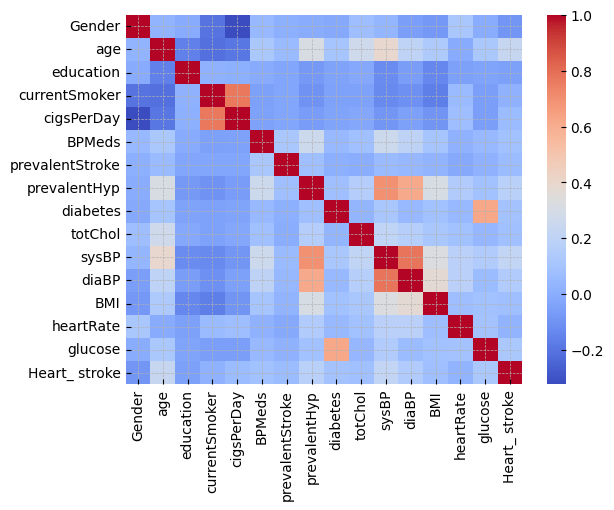
correlation\_matrix = df.corr()

# Create a heatmap using the correlation matrix

sns.heatmap(correlation\_matrix, annot=False, cmap='coolwarm')

# Show the plot

plt.show()



*Figure 9. Correlation Graph*

Glucose indeed has a strong correlation to diabetes.

It is well known that glucose levels are associated with diabetes. “Hyperglycemia affects people who have diabetes.” Furthermore: “In people who have diabetes, glucose tends to build up in the bloodstream.” (Mayo Clinic, 2022) The correlation between these glucose spikes and diabetes was found with Bayes theorem. Taking a subset of values above the normal threshold of glucose and another subset of that data with individuals with diabetes was taken, resulting in the intersection of individuals with diabetes and individuals with high glucose levels.

The number of people with diabetes where then filtered:

print(df['diabetes'].value\_counts())

A black background with white text

Description automatically generated

The percentage of people with diabetes in the total sampled population equated to ~2.56%:

print(df['diabetes'].value\_counts(normalize=True) \* 100)

A black background with white text

Description automatically generated

The total number of people with diabetes equated to:

print(df['diabetes'].value\_counts())

A black background with white text

Description automatically generated107

The number of people with diabetes given glucose levels are above 100 was found by sub setting:

Input:

correlation\_diabetes = df[df['glucose'] > 100]

print(len(correlation\_diabetes))

print(len(correlation\_diabetes[correlation\_diabetes['diabetes'] == 1]))

print(len(correlation\_diabetes[correlation\_diabetes['diabetes'] == 1]) / len(correlation\_diabetes) \* 100)

output:

A black background with white numbers

Description automatically generated

* The first print statement 309 was the number of rows (or number of individuals) with glucose levels over 100. The subset of glucose levels.
* The second print statement 85 was the number of rows (number of individuals) with diabetes.
* The third print statement (27.5) showed the percentage of individuals with diabetes within the subset of those who had glucose levels over 100. This indicates that those who have higher levels of glucose are 27.5% more likely to have diabetes.

So compared to the total population of people with diabetes, those with high glucose accounted for the majority of those with diabetes (~79%):

A screenshot of a computer program code


This operation proved there was a correlation between these in higher scoring glucose levels and diabetes.

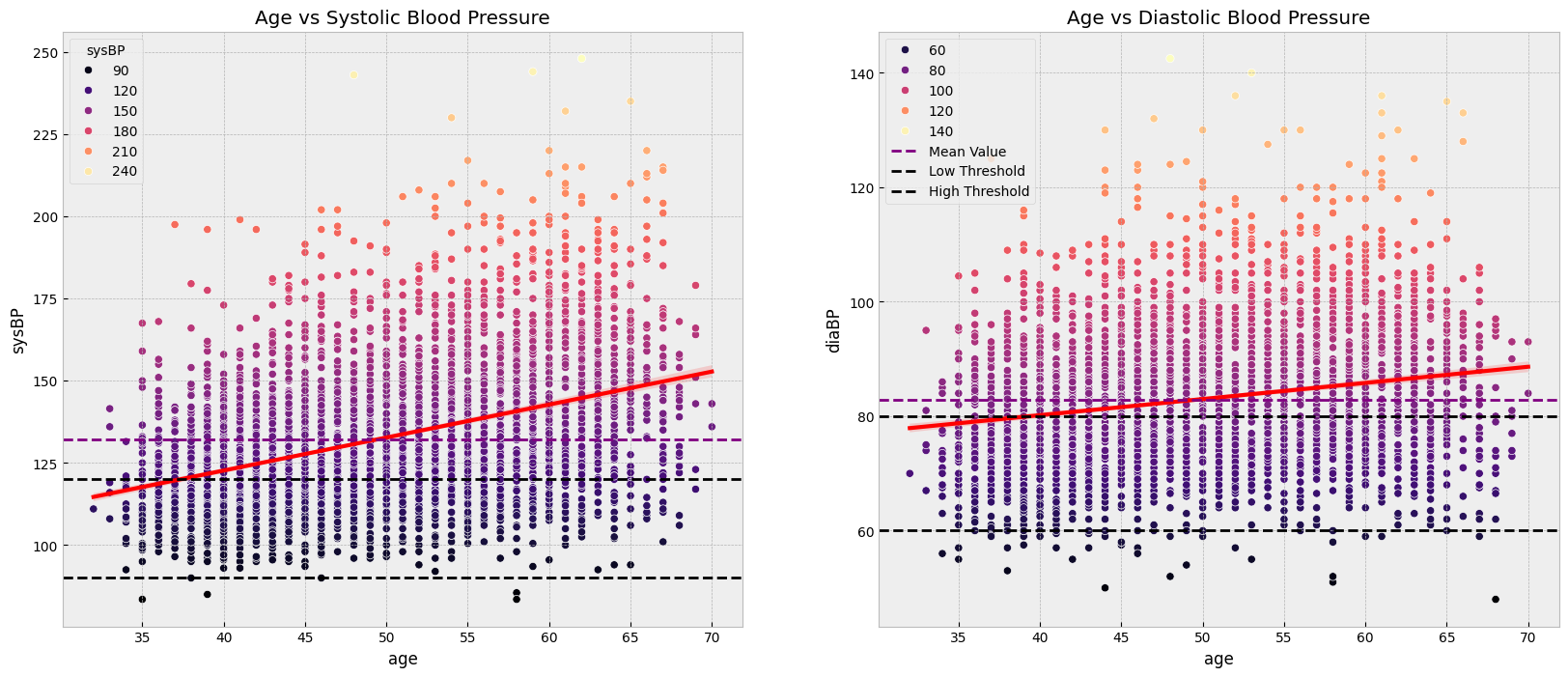
A graph of different colored lines

Description automatically generated

*Figure 10. Fluctuations of all numerical variables*

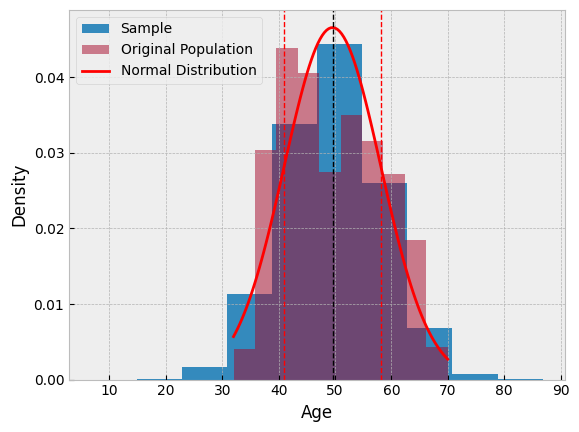
Juxtaposed against other variables, glucose fluctuations did show the most unusual signs of activity from the mean. Compared to the larger variance of the cholesterol levels.

The subsequent analysis was focused on the correlation of aging to various other factors of heart disease. Another correlation was apparent from Figure 9 concerning aging: Systolic blood pressure. To investigate this, linear regression was performed on the two variables with a scatter plot:



Diastolic Blood pressure was included as they are closely related to each other (as pointed out in Figure 2: variable interpretation chart) They both increase with age. Much of the population was well above the normal levels of Blood Pressure, providing evidence that these blood pressures, such as systolic or diastolic blood pressure, are a clear indication of many forms of heart disease. As age increases so does the average systolic blood pressure, by a considerable degree.

An analysis of the age distribution of individuals with heart disease was conducted using a normalized distribution, with the calculation of the probability density function (PDF) to examine the age occurrence pattern (W3Schools, n.d.)

**

*Figure 11. distribution of the original population, juxtaposed with the random, normalized sample of the population.*

A random sample of 500,000 observations was generated from the population's age distribution using NumPy's random.normal() function, with parameters μ (mu) and σ (sigma) defining the normal distribution.(Dickinson, 2017)

samples = np.random.normal(mu, sigma, size=500000)

Generating 500,000 samples to make an accurate representation of a normalized distribution.

With this data, the variance of the first standard deviation was observed to be at age 41 and 58, meaning ~68.0% of people with heart disease are between 40 and 58.

As indicated, elevated diastolic blood pressure levels are associated with other heart-related diseases. A proposed hypothesis is whether elevated diastolic blood pressures are also associated with individuals in the dataset who have experienced stroke as well.

A T-test was conducted on these samples to identify the relevance of these two instances and provide some evidence of causation. An alpha value of 5% was established as the critical value for the threshold for the rejection of this hypothesis.

The normal distribution was plotted on two subsets of the main population. One who hadhigher than normal diastolic blood pressures, and the other of the stroke population with hifh diastolic blood pressures. Below was the code used to produce these results:

def mainPopulation(dataset):

    # compare and overlay the normal distributions of the diaBP between the main dataframe and the stroke subset of the dataframe saved as stroke variable

    # Calculate the mean and standard deviation of the dataset

    mu = dataset['sysBP'].mean()

    sigma = dataset['sysBP'].std()

    print(f'Mean: {mu}, Standard Deviation: {sigma}')

    # Generate a large number of samples from the normal distribution (500,000)

    samples = np.random.normal(mu, sigma, size=500000)

    # Plot the histogram of the generated samples

    plt.hist(samples, bins=10, density=True, label='Sample')

    # Plot the histogram of the dataset

    plt.hist(dataset['sysBP'], bins=10, density=True, alpha=0.5, label='Original Population')

    # Plot the normalized distribution

    x = np.linspace(dataset['sysBP'].min(), dataset['sysBP'].max(), 100)  # Generate 100 points between the min and max values of the dataset

    y = norm.pdf(x, mu, sigma)                                 # Calculate the normal distribution of the generated points

    plt.plot(x, y, 'r-', label='Normal Distribution')          # Plot the normal distribution

    # Add a vertical line at the mean and variance

    plt.axvline(mu, color='k', linestyle='dashed', linewidth=1)         # Add a vertical line at the mean

    plt.axvline(mu + sigma, color='r', linestyle='dashed', linewidth=1) # Add a vertical line at the mean + variance

    plt.axvline(mu - sigma, color='r', linestyle='dashed', linewidth=1) # Add a vertical line at the mean - variance

    # Calculate the percentage of people between each variance

    # Then round the percentage to the closest whole number

    percentage = norm.cdf(mu + sigma, mu, sigma) - norm.cdf(mu - sigma, mu, sigma)

    percentage = round(percentage, 2)

    print(f'{percentage \* 100}% of people with heart disease are between {int(mu - sigma)} and {int(mu + sigma)}')

    # Add labels and legend

    plt.xlabel(f'Diastolic Blood Pressure for {"Main Population" if dataset is df else "Stroke Population"}')

    plt.ylabel('Density')

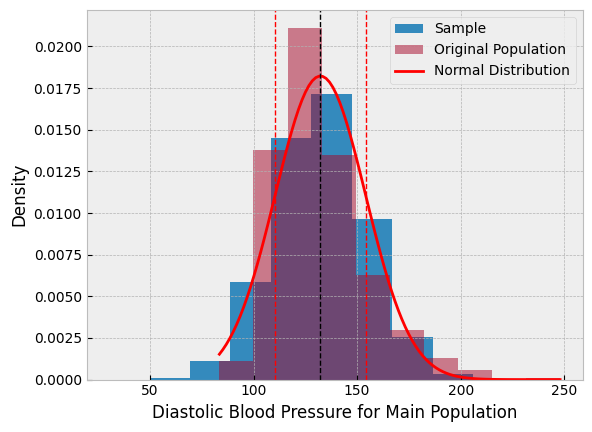
    plt.legend()

    # Show the plot

    plt.show()

mainPopulation(df)

mainPopulation(has\_stroke)



A graph with a red line

Description automatically generated

The T-test was performed with SciPy ttest\_ind()(Scipy, n.d.) with the two subsets as parameters.

# Perform a t-test on the diastolic blood pressure between people with and without a stroke

t\_stat, p\_val = ttest\_ind(stroke['diaBP'], df['diaBP'])

p\_val = 1 - p\_val

# Print the t-statistic and p-value

print(f'The t-statistic is: {t\_stat}')

print(f'The p-value is:  {p\_val}')

Which produced:

A black background with white numbers

Description automatically generated

“The p-value quantifies the probability of observing as or more extreme values assuming the null hypothesis” (Scipy, n.d.)

Therefore, the Null Hypothesis cannot be rejected.

A blue circle with a red line

Description automatically generated

Diastolic blood pressure levels seemed to be proportional to many heart conditions. They seem to point out underlying causes. Blood pressure levels make good suggestions for the diagnosis of heart problems.

Refer to the Jupyter Notebook source code attached for more information on the methods of this report.

**References:**

Byjus. (n.d.). *Blood pressure- Types of Blood Pressure and its Differences*. BYJUS. Retrieved 9 May 2024, from https://byjus.com/biology/systolic-and-diastolic-blood-pressure/

*Cholesterol Levels: By age, LDL, HDL, and More*. (2020, June 16). Healthline. https://www.healthline.com/health/high-cholesterol/levels-by-age

Dickinson, M. (2017, January 25). *Answer to ‘Generate random numbers with a given (numerical) distribution’*. Stack Overflow. https://stackoverflow.com/a/41852266

Mayo Clinic. (2022, August 20). *Hyperglycemia in diabetes-Hyperglycemia in diabetes—Symptoms & causes*. Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/hyperglycemia/symptoms-causes/syc-20373631

Mirza, M. H. (2023). *Heart Disease-UCI*. Kaggle. https://www.kaggle.com/datasets/mragpavank/heart-diseaseuci

Riley, L. (n.d.). *Indicator Metadata Registry Details*. Retrieved 9 May 2024, from https://www.who.int/data/gho/indicator-metadata-registry/imr-details/5657

Rob Mulla (Director). (2021, December 28). *A Gentle Introduction to Pandas Data Analysis (on Kaggle)*. https://www.youtube.com/watch?v=\_Eb0utIRdkw

Scipy. (n.d.). *scipy.stats.ttest\_ind—SciPy v1.13.0 Manual*. Retrieved 10 May 2024, from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest\_ind.html

W3Schools. (n.d.). *Normal (Gaussian) Distribution*. Retrieved 6 May 2024, from https://www.w3schools.com/python/numpy/numpy\_random\_normal.asp