**Problem Statement**

Heart disease continues to be the leading cause of death in the US, affecting people from a wide range of demographic backgrounds. Despite significant research and preventive measures, many individuals remain at risk due to factors like high blood pressure, smoking, obesity, and more. These factors continue to pose serious threats to public health, emphasizing the need for new and effective strategies to identify, understand, and tackle them.

This project aims to delve into the vast dataset comprising responses from over 30K+ adults, extracted as part of the 2022 annual CDC survey. This extensive dataset holds valuable insights waiting to be discovered, providing a deeper understanding of how different risk factors interact and affect heart health. By using advanced machine learning techniques like classification and regression algorithms, we aim to uncover important patterns and connections hidden within the data. Through thorough analysis and validation, the predictive models developed have the potential to transform early detection and intervention methods for addressing heart disease.

The main aim of this effort is to give healthcare professionals strong tools and knowledge to find people at higher risk of heart disease early on. This means they can act quickly by suggesting lifestyle changes or medical treatments, which could prevent serious heart problems and make people healthier overall. Also, creating precise prediction models could help shape public health policies and decide where to focus resources, so interventions and preventive actions can be customized for different groups of people. By employing advanced machine learning algorithms on vast datasets, our aim is to reveal concealed patterns and develop predictive models that could transform early detection and intervention methodologies. The central objective of this endeavor is to advance public health by comprehensively understanding the etiology of heart disease, providing proactive measures, and mitigating its prevalence.

# A. Background

According to the World Health Organization (WHO), cardiovascular diseases are responsible for approximately 17.9 million deaths annually. This prevalence underscores the urgent need to address the problem. Numerous factors contribute to the development of heart disease, including high blood pressure, high cholesterol levels, smoking, obesity, diabetes, and sedentary lifestyles. The increasing prevalence of these risk factors, often driven by changes in diet, physical activity, and lifestyle, has contributed to the rising incidence of heart disease.

Heart disease significantly decreases the standard of life for those who have it in addition to increasing the risk of dying young. Chest pain, breathing difficulties, exhaustion, and limited exercise are just a few of the symptoms that severely limit day-to-day activities and general health. Furthermore, one shouldn't underestimate the emotional toll that individuals and their families take. With its widespread occurrence and significant influence on both death and mortality, heart disease is an important public health issue. Efforts to prevent, detect, and treat heart disease serve wider public health objectives, such as reducing the burden of noncommunicable diseases and fostering healthier communities, in addition to improving individual health outcomes.

Moreover, specific demographic groups, including racial and ethnic minorities, socioeconomically disadvantaged populations, and rural communities, bear a disproportionate burden of heart disease and its associated risk factors. Addressing this issue necessitates tackling these disparities head-on and ensuring that all individuals have fair access to prevention, diagnosis, and treatment services. By prioritizing equitable healthcare access and interventions tailored to diverse populations, we can effectively combat the pervasive impact of heart disease and work towards fostering inclusive and healthier societies.The COVID-19 pandemic has further heightened concerns about cardiovascular health, emphasizing the need for research that explores the intersection of COVID-19, heart disease, and social determinants of health, particularly focusing on vulnerable populations. In summary, addressing heart disease is essential not only for improving individual health but also for promoting overall population health and well-being.

**B. Why is this study important?**

This study holds immense importance due to the pervasive impact of heart disease on global health. As one of the foremost causes of mortality worldwide, heart disease places a substantial burden on healthcare systems, stretching resources thin and compromising the quality of patient care. Through rigorous and comprehensive research efforts, scientists aim to delve deep into every facet of heart disease, from its underlying causes to the identification of risk factors and the development of effective treatment options. By gaining a thorough understanding of the intricacies of this complex condition, researchers aspire to unearth invaluable insights that will be pivotal in the ongoing battle against this formidable health challenge.

Furthermore, the significance of this study extends beyond the realm of healthcare systems and individual patient outcomes. Heart disease exacts a profound toll on society as a whole, imposing economic burdens and societal costs. By shedding light on the multifaceted nature of heart disease through robust research endeavors, we stand to enhance our ability to mitigate its impact on public health and well-being. Ultimately, this study represents a crucial step towards empowering healthcare providers, policymakers, and individuals alike with the knowledge and tools needed to effectively prevent, diagnose, and manage heart disease. Through collaborative and interdisciplinary research efforts, we can work towards reducing the global burden of heart disease and fostering healthier communities for generations to come.

# Data Sources

Data Source : <https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/data>

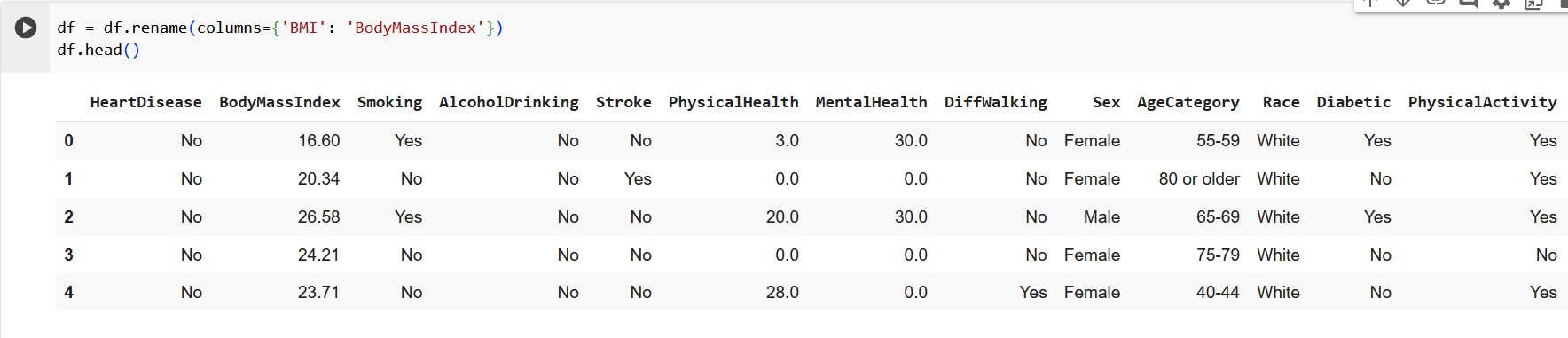
The dataset originally comes from the CDC and is a major part of the Behavioral Risk Factor Surveillance System (BRFSS). This system conducts yearly telephone surveys aimed at gathering information about the health status of residents in the United States. Specifically, the dataset used in this project is derived from the 2022 annual CDC survey, encompassing responses from over 30K+ adults regarding their health status. It comprises 18 variables, including the target variable.

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| HeartDisease | object | If the respondent had a heart disease or not. (Target  Variable) |
| BMI | float64 | Body Mass Index |
| Smoking | object | The respondent is a smoker or not. |
| AlcoholDrinking | object | Adults who reported having had at least one drink of alcohol in the past 30 days. |
| Stroke | object | If the respondent had a stroke or not. |
| PhysicalHealth | float64 | About physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good. |
| MentalHealth | float64 | About mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good? |
| DiffWalking | object | Have serious difficulty walking or climbing stairs or not. |
| Sex | object | Sex of respondent |
| AgeCategory | object | Age of respondent |
| Race | object | Race of respondent |
| Diabetic | object | If the respondent is diabetic or not. |
| PhysicalActivity | object | During the past month, other than your regular job, did the respondent participate in any physical activities or exercises such as running, calisthenics, golf, gardening, or walking for exercise? |
| GenHealth | object | General health condition of respondent |
| SleepTime | int64 | On average, how many hours of sleep respondent get in a 24-hour period |
| Asthma | object | If the respondent had asthma or not. |
| KidneyDisease | object | If the respondent had a kidney disease or not. |
| SkinCancer | object | If the respondent had skin cancer or not. |

**Data Cleaning/Processing**

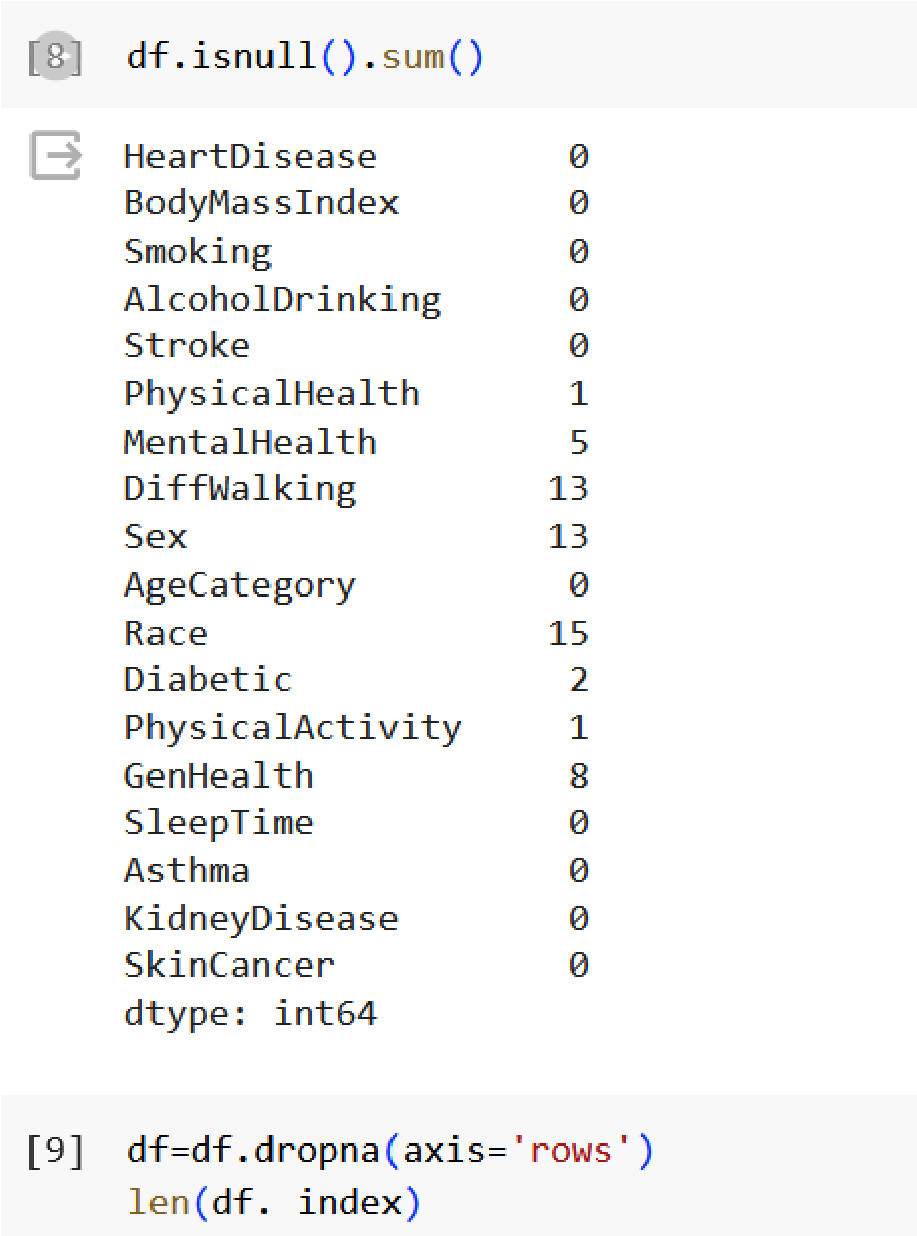
# Renaming the columns

This part of the code is a method call on the DataFrame df. It's using the .rename() method to rename the columns of the DataFrame. The columns parameter specifies a dictionary where the keys are the old column names, and the values are the new column names. In this case, it's renaming the column 'BMI' to 'BodyMassIndex'.



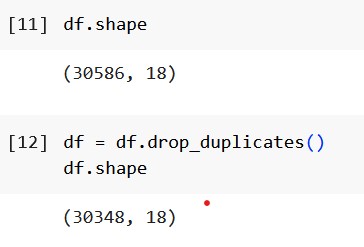
# Dealing NA values

Dealing with missing values before training the data and performing exploratory data analysis (EDA) is crucial to ensure data integrity, prevent bias in analysis, improve model performance, and facilitate meaningful insights. Handling missing values upfront helps maintain the reliability of the dataset, prevents errors in analysis, and ensures that machine learning models can effectively learn from the data. Additionally, addressing missing values early allows for a more focused and accurate exploration of meaningful patterns and relationships in the data during EDA. The DataFrame contains 30,000 rows. Initially, there were 48 null values present in the DataFrame. After dropping the rows with null values, the DataFrame now contains no null values. The null values were removed to ensure data integrity and accuracy in subsequent analyses. The DataFrame is now ready for further exploration or analysis without the presence of missing values.



# Dealing Duplicate Values

The process of removing duplicates from a DataFrame is essential for ensuring data integrity and accuracy in analyses. In the provided scenario, the DataFrame originally contained 30,586 rows and 18 columns, indicating a significant amount of data. However, after removing duplicates using the drop\_duplicates() function, the DataFrame was streamlined to 29,925 rows while retaining all 18 columns. This reduction in the number of rows suggests that there were instances of duplicated records within the dataset. Removing duplicates is critical because they can distort statistical analyses, such as calculating means or correlations, and lead to erroneous conclusions. By eliminating duplicate rows, analysts can work with a more precise and representative dataset, enhancing the reliability and validity of their findings. This process is particularly important in data preprocessing steps before conducting exploratory data analysis (EDA) or training machine learning models, as it ensures that subsequent analyses are based on accurate and non-redundant data. Overall, removing duplicates contributes to maintaining data quality and enhancing the robustness of analytical insights derived from the dataset.



# Dealing Data Values

Changing the data type of columns can be essential for several reasons. By converting these columns to strings, it ensures that the values within them are treated as text rather than numerical or categorical data. This transformation may be necessary to facilitate certain types of analyses or operations, such as string manipulation or textual comparisons. For instance, converting 'Race' and 'GenHealth' to strings could be beneficial if the values represent categories or labels that need to be handled as text. Additionally, converting data types can help in optimizing memory usage and improving computational efficiency, especially when dealing with large datasets. However, it's important to ensure that the chosen data type conversion accurately reflects the nature of the data and aligns with the intended analytical tasks to avoid unintended consequences in subsequent analyses or modeling efforts.



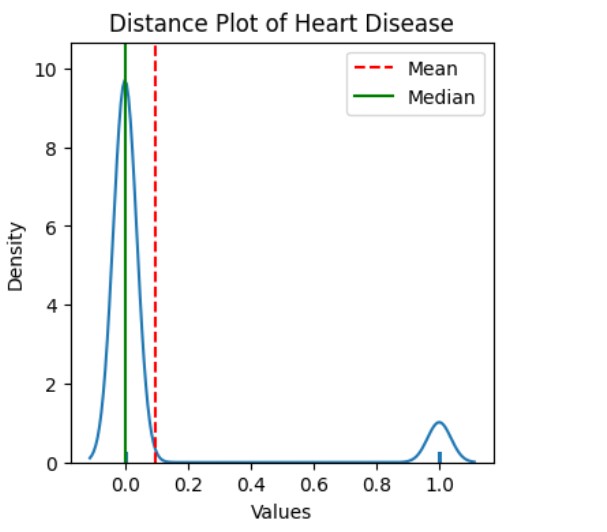
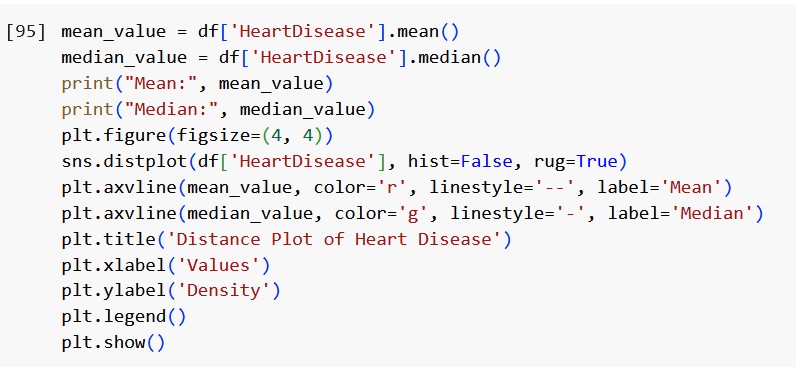
# Dealing Numerical Values

The provided code snippet replaces categorical values ('Yes' and 'No') with numerical equivalents (1 and 0, respectively) in several columns of a DataFrame. This transformation is crucial for preparing the data for future analysis and training machine learning models. By converting categorical variables to numerical representations, the data is standardized, facilitating model training and improving performance. Additionally, numerical representations enhance interpretability and ensure compatibility with machine learning algorithms designed to operate on numerical inputs. Overall, this conversion process streamlines data processing, making it more suitable for analytical tasks and machine learning applications.



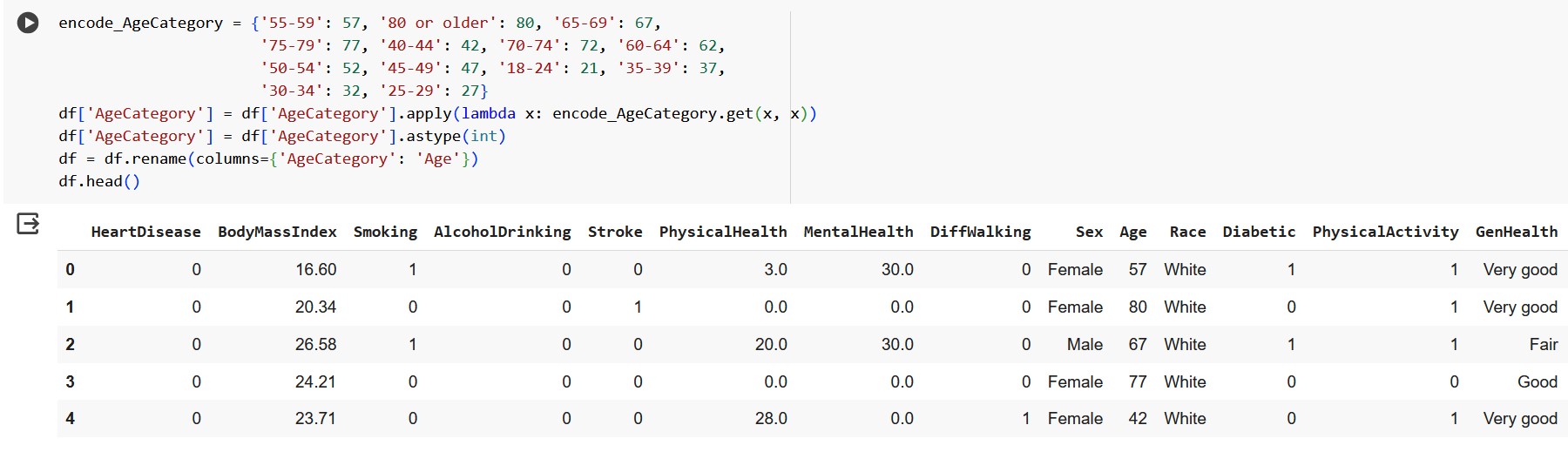
# Understanding the Data

The provided code snippet conducts an essential analysis and visualization process for the 'HeartDisease' column in the DataFrame. First, it computes both the mean and median values of the column, representing the average and middle values of the dataset, respectively. These statistics offer fundamental insights into the central tendency of the data distribution, crucial for understanding its overall behavior. By printing these values, analysts can compare them to assess the skewness or symmetry of the distribution and identify potential outliers or anomalies. Additionally, the code generates a density plot using Seaborn's distplot() function, illustrating the distribution of 'HeartDisease' values. This visual representation aids in detecting patterns, trends, or irregularities in the data. Moreover, vertical lines denoting the mean and median are overlaid onto the plot, providing a visual reference for their positions within the distribution. Comparing the mean and median visually allows for a quick assessment of the distribution's shape and the presence of outliers. Altogether, this step is critical for data cleaning and analysis, enabling analysts to gain insights into the dataset's characteristics, identify data quality issues, and make informed decisions during the preprocessing phase.



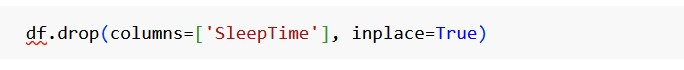
# Dealing Categorical Data

In the provided code snippet, categorical age categories are encoded into numerical values for analysis and modeling purposes. Initially, a dictionary named encode\_AgeCategory is constructed to map each age category to its corresponding numerical representation. Subsequently, this encoding scheme is applied to the 'AgeCategory' column of the DataFrame (df) using the .apply() function and a lambda expression. For each value in the 'AgeCategory' column, the lambda function looks up the corresponding numerical value from the dictionary, replacing each age category with its numerical equivalent. After encoding, the data type of the 'AgeCategory' column is converted to integer to ensure consistency and compatibility with numerical operations. Additionally, the column name is renamed to 'Age' for clarity and conciseness. This encoding process transforms categorical age data into a numerical format, facilitating statistical analysis, visualization, and machine learning tasks that require numerical inputs. Overall, this step contributes to data preprocessing efforts, enabling more robust and meaningful analysis of age-related trends and patterns within the dataset.



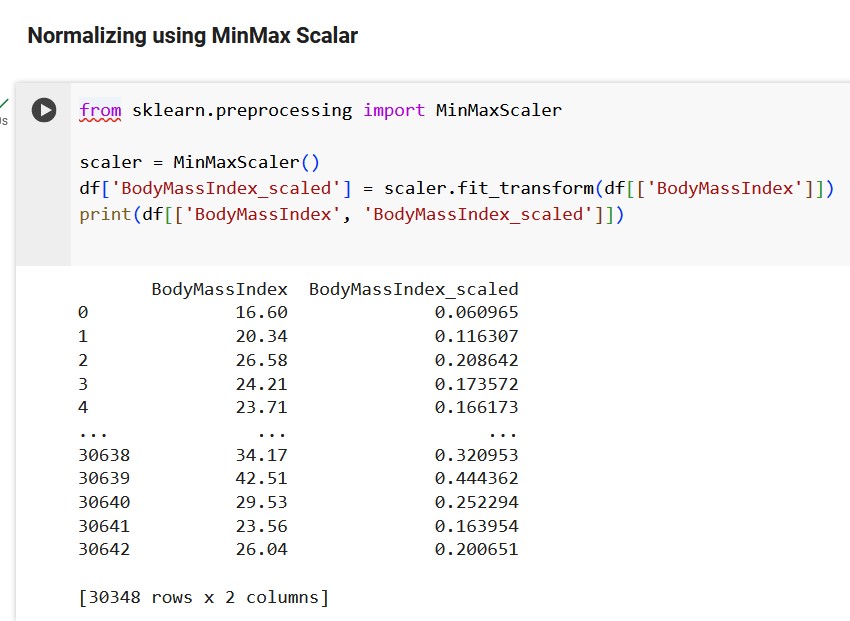
# Removing Unwanted Columns

The provided code snippet removes the 'SleepTime' column from the DataFrame (df) using the .drop() method with inplace=True. This method allows for the removal of specific columns from the DataFrame. By passing 'SleepTime' as an argument within the columns parameter, the method identifies and drops the column named 'SleepTime'. This action is permanent and directly modifies the original DataFrame df, effectively eliminating the specified column from the dataset. The removal of 'SleepTime' may be necessary for various reasons, such as if the column is deemed irrelevant to the analysis or modeling tasks at hand. By eliminating unnecessary columns, the DataFrame is streamlined, reducing complexity and potentially improving computational efficiency. It's important to ensure that the removal of the 'SleepTime' column does not result in the loss of critical information or adversely impact subsequent analyses or modeling efforts. Overall, this step contributes to data preprocessing by refining the DataFrame to include only relevant features for the intended analytical objectives.



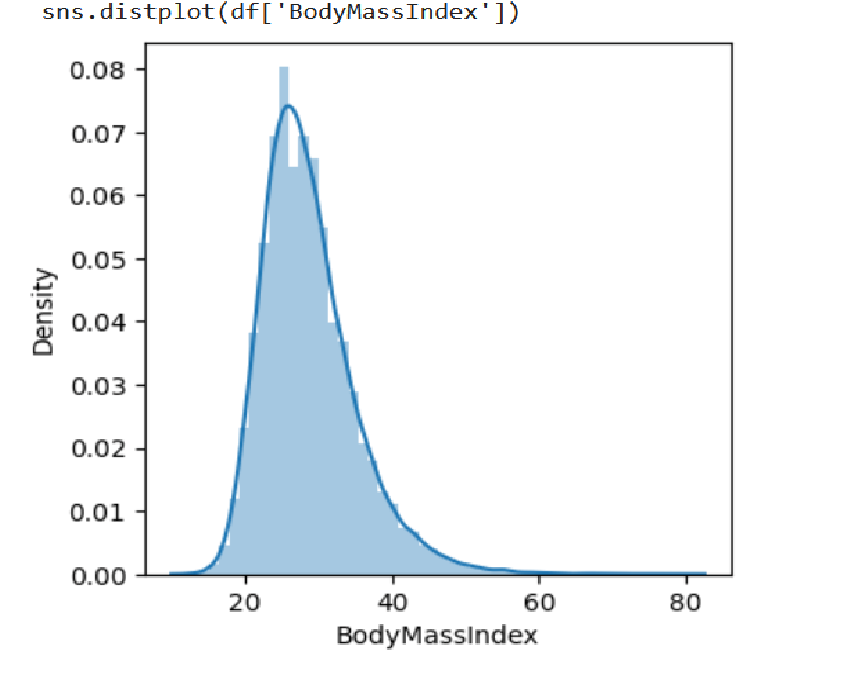
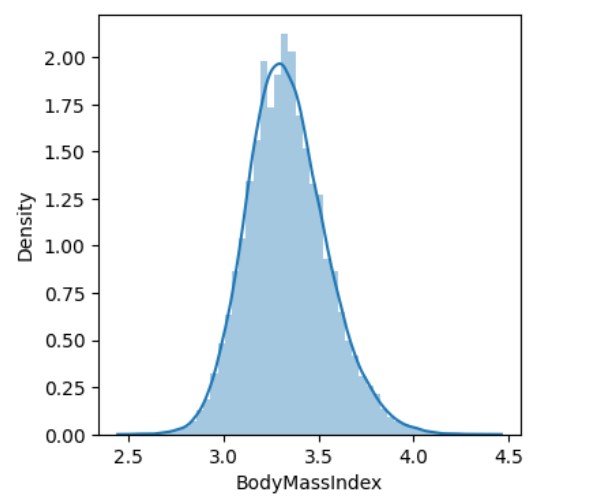
# Normalizing the data

The provided code snippet demonstrates the application of Min-Max scaling, a common technique used in data preprocessing, specifically to normalize the 'BodyMassIndex' column of a DataFrame (df). Min-Max scaling is employed to transform numerical features to a specified range, typically between 0 and 1, ensuring that all features are on a similar scale. In the code, the MinMaxScaler class from the sklearn.preprocessing module is imported, providing the necessary functionality for scaling the data. An instance of the MinMaxScaler class is then instantiated and stored in the variable scaler. This instance is subsequently used to perform the normalization process. The fit\_transform() method of the MinMaxScaler instance is applied to the 'BodyMassIndex' column of the DataFrame, computing the minimum and maximum values of the feature and scaling its values accordingly. The scaled values are then stored in a new column named 'BodyMassIndex\_scaled' within the DataFrame. Finally, the code prints out both the original 'BodyMassIndex' values and the corresponding scaled values for comparison, allowing for a visual assessment of the effect of the normalization process. Overall, Min-Max scaling ensures that features contribute equally to the analysis, prevents features with larger scales from dominating the learning process, and improves the performance of machine learning algorithms, particularly those sensitive to feature scales.



# Dealing with skewness

The provided code snippet addresses skewness in the 'BodyMassIndex' column of the DataFrame (df) through a transformation process and subsequent visualization. Skewness refers to the asymmetry of the distribution of data points, which can affect the validity of statistical analyses and modeling efforts. In this case, the code applies a logarithmic transformation using the natural logarithm (np.log) to the 'BodyMassIndex' column. This transformation aims to reduce the skewness of the data and make its distribution more symmetrical. Before and after applying the transformation, density plots of the 'BodyMassIndex' column are visualized using Seaborn's distplot() function, allowing for a visual assessment of the distribution's shape and skewness. Additionally, the code computes the skewness of the transformed 'BodyMassIndex' column to quantify the degree of skewness present in the data. By visualizing the effect of the transformation and computing the skewness, the code provides insights into the effectiveness of the transformation in reducing skewness and improving the normality of the data distribution. Overall, addressing skewness through appropriate transformations is essential for ensuring validity and accuracy.



**Exploratory Data Analysis (EDA)**

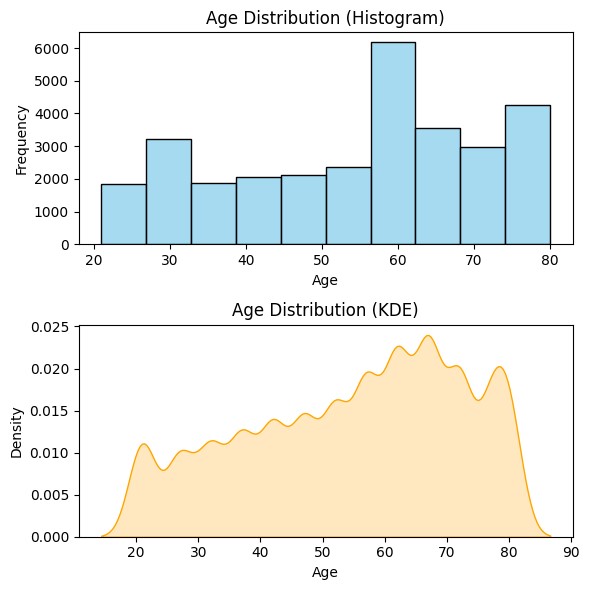
Exploratory Data Analysis is the process of analyzing, summarizing, and visualizing a dataset to gain insights and understanding of the data. EDA is done to understand big data before using expensive big data methodology. EDA is the first step in data analysis that identifies the patterns and anomalies in the data. Basic tools of EDA are plots, graphs, and summary stats. It is a method for systematically going through data, plotting distributions, plotting time series, looking at pairwise relationships using scatter plots, and generating summary stats. Eg: min, max, mean, upper, lower quartiles, identifying outliers.

Univariate, bivariate and multivariate are used to describe the number of variables that are being analyzed in any particular analysis. Choosing any of these analyses will depend on the type of data that needs to be analyzed.

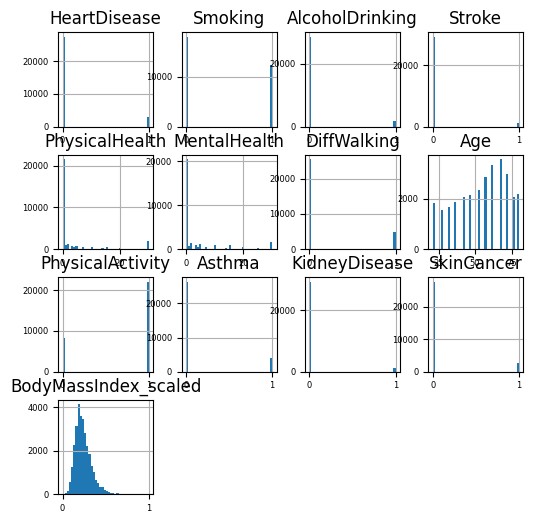
**EDA Univariate Analysis:**

Univariate analysis is based on analyzing a single variable at a time. This is used for understanding the distribution and spread of a variable. Most used approaches of univariate are histograms, box plots, and summary statistics.

1. **Histogram:** It helps in analyzing the distribution of one or more variables by counting the observations that fall within the discrete bins. It helps in finding the median and frequency of the data. If there are any gaps or outliers in the data they can be easily identified.
   * 1. The first subplot displays a histogram of the 'Age' column.The second subplot visualizes the kernel density estimate (KDE) of the 'Age' column.

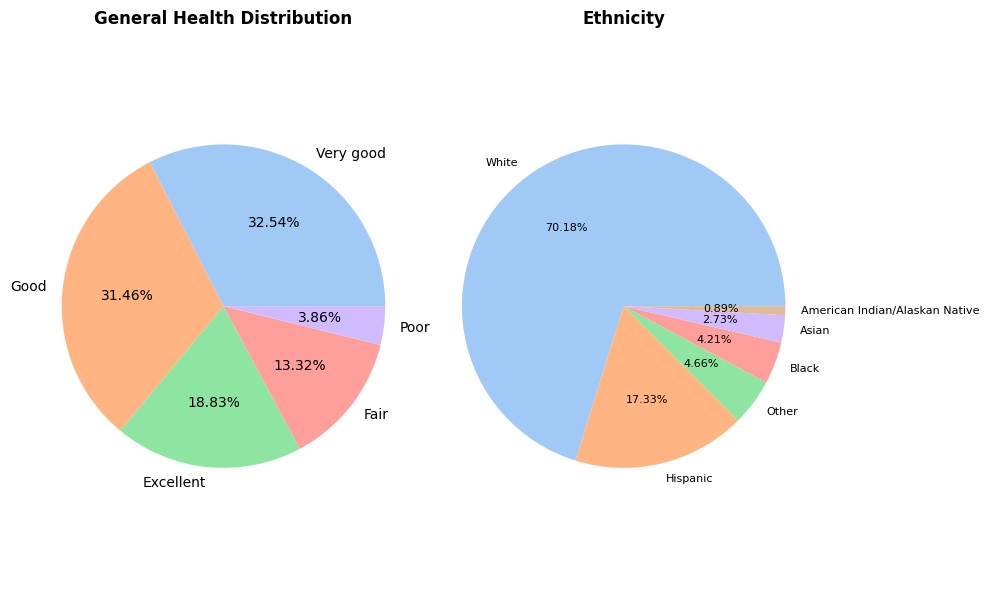


* + 1. The histogram plot efficiently represents a compact histogram visualization of the DataFrame's data distribution with customized sizing and labeling.



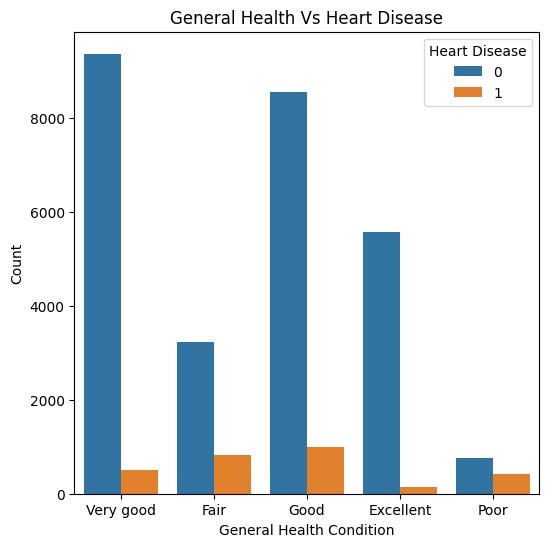
1. **Pie Chart:** pie chart displays categorical variable distributions. It visualizes each category's proportion relative to the whole dataset. It aids in understanding category frequencies and identifying dominant patterns within data, serving as a tool for individual variable exploration.

The first chart displays the distribution of general health values obtained from a DataFrame column labeled 'GenHealth'. The second chart represents the distribution of ethnicity values obtained from a DataFrame column labeled 'Race'. Both charts are set to have equal aspect ratios for accurate visualization.

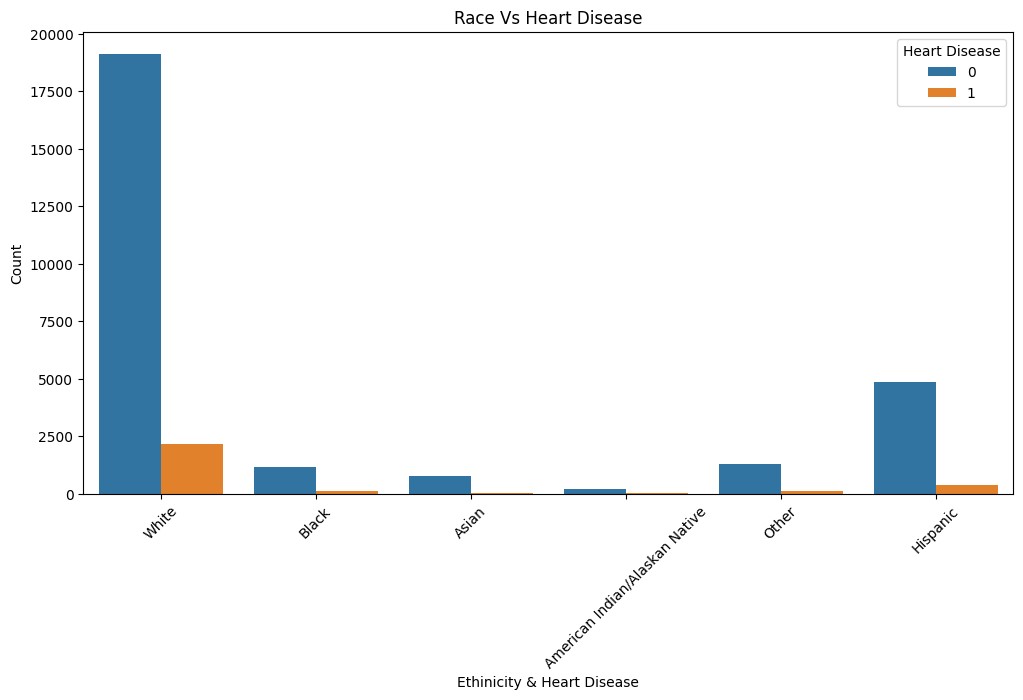


1. **Bar Plot:** Bar charts display the distribution of categorical variables by representing the frequency or proportion of each category as bars. They are effective for comparing the relative sizes of different categories and identifying patterns or trends within categorical data.

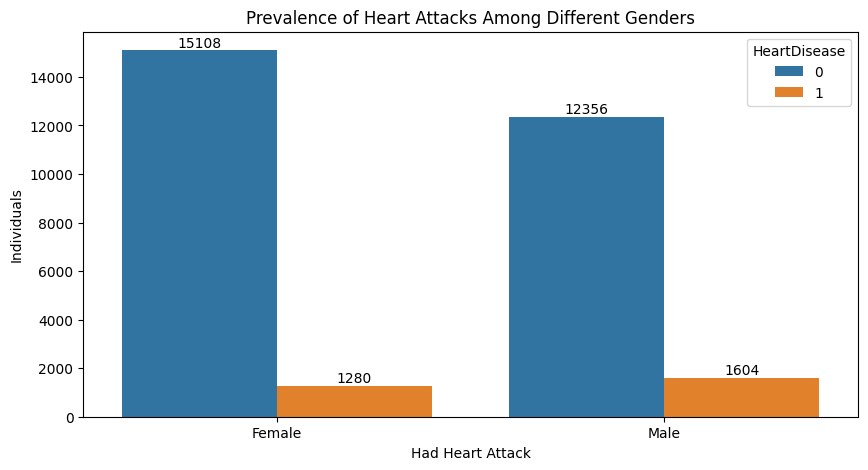
This bar plot compares the occurrences of heart disease across different general health conditions. The 'GenHealth' column is used on the x-axis to represent various general health conditions, while the 'HeartDisease' column is utilized for creating separate bars based on the presence or absence of heart disease.



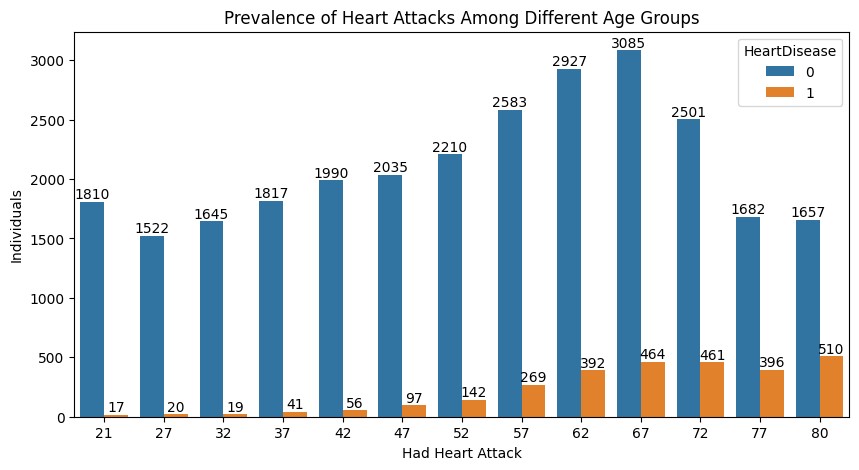
1. **Count Plot:** It represents the counts of the observation present in the categorical variable. It shows the visual depiction in the bar chart.
   1. The Count Plot with 'Race' on the x-axis and 'HeartDisease' on the hue, the x and y axes denoting 'Ethnicity & Heart Disease' and 'Count' respectively.This plot presents race distribution concerning heart disease occurrences.



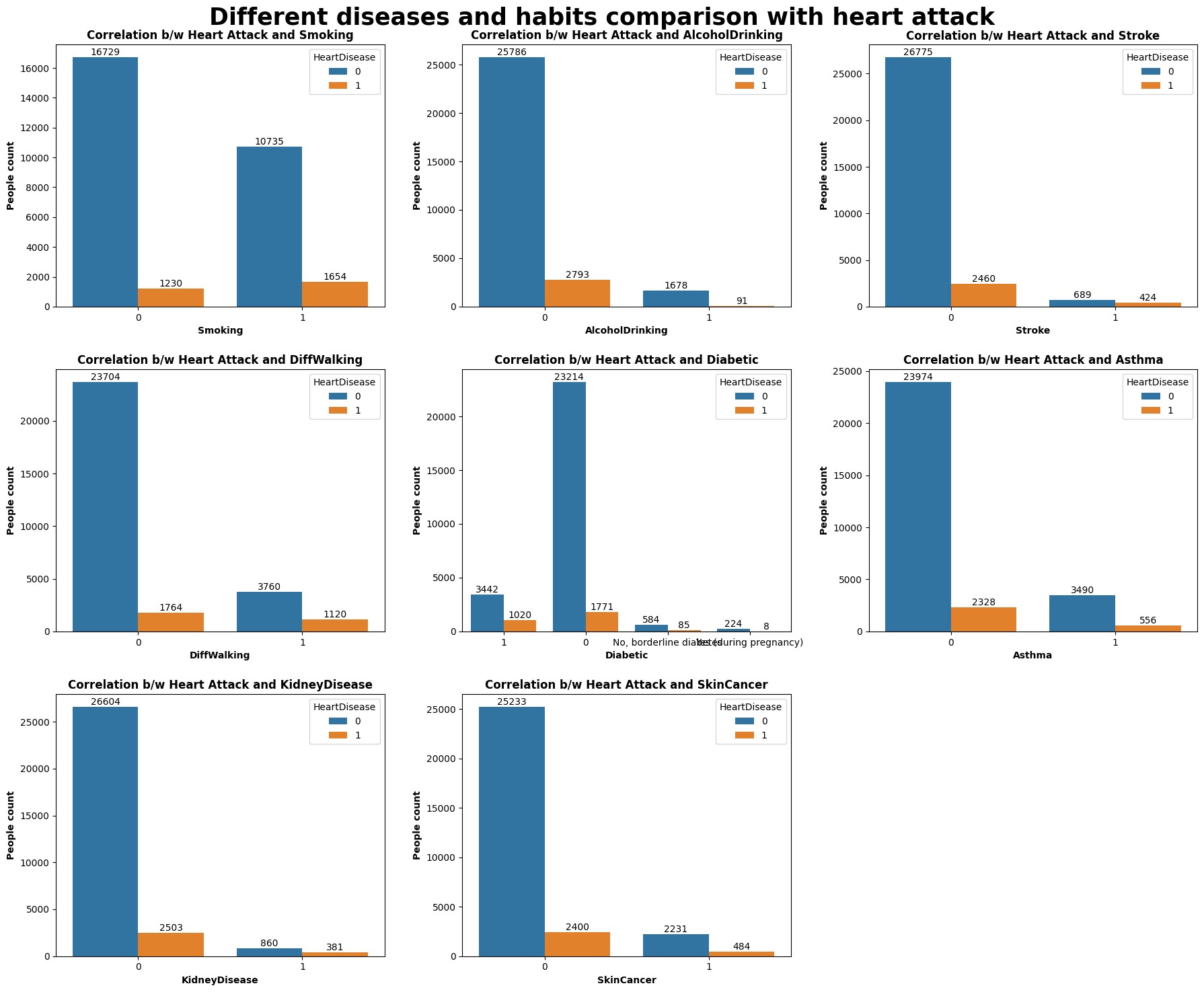
* 1. The Count Plot represents the prevalence of heart attacks across different genders. The countplot function from seaborn is utilized to create a bar chart that displays the count of individuals who had heart attacks, categorized by gender. The 'Sex' column is used as the x-axis variable, while the 'HeartDisease' column is specified as the hue for distinguishing heart attack occurrences.



* 1. The Count Plot represents the prevalence of heart attacks across various age groups.The countplot function from Seaborn is employed to create a bar chart that displays the count of individuals with and without heart disease categorized by age.



1. **Grid count plot:**The grid count plot conducts an analysis of various chronic diseases and lifestyle factors in relation to heart disease. Specifically, it selects columns related to heart disease, smoking, alcohol consumption, stroke, difficulty walking, diabetes, asthma, kidney disease, and skin cancer from the DataFrame. Each subplot displays the count of individuals with and without heart disease based on the specific chronic disease or lifestyle factor. The title of each subplot reflects the correlation between heart attacks and the corresponding factor.

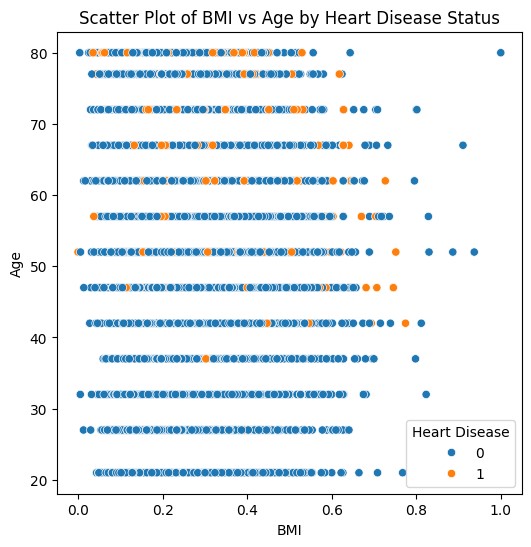


**EDA (Exploratory Data Analysis) Bi-Variate:**

Bivariate analysis is based on analyzing the relationship between two variables. This is used for understanding how one variable is related to another variable. Most used approaches are scatter plots, correlation analysis, and contingency tables.

1. **Scatter Plots**:The Scatter plots visualize the relationship between two numerical variables. Each data point represents a combination of values for the two variables, allowing for the observation of patterns, trends, and potential correlations between them.

This scatter plot depicts the relationship between two variables: Body Mass Index (BMI) and Age, differentiated by Heart Disease status.The 'BodyMassIndex\_scaled' column is used for the x-axis, 'Age' for the y-axis, and the 'HeartDisease' column dictates the hue of the data points, providing a visual representation of heart disease status. The scatter plot illustrates the relationship between BMI, Age, and Heart Disease status.

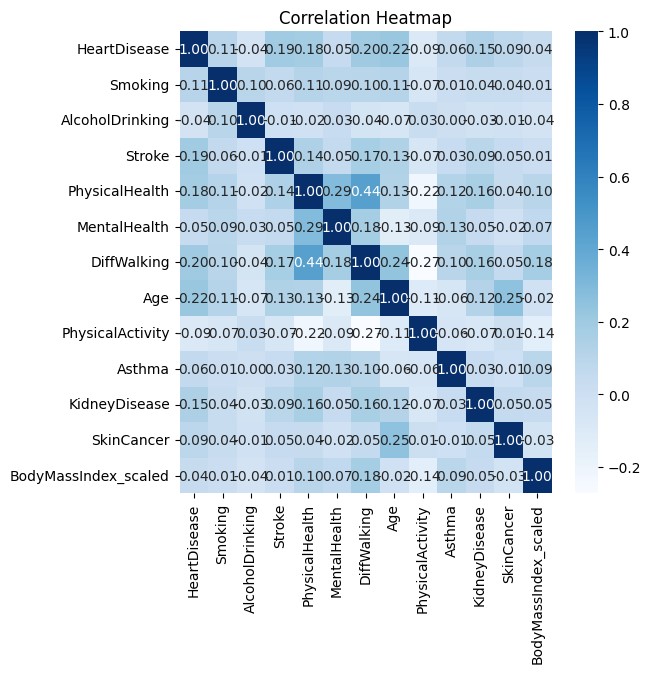


**EDA (Exploratory Data Analysis) Multi-Variate:**

Multivariate analysis is based on analyzing more than two variables simultaneously. This is used for understanding the relationships between multiple variables. Most used approaches are factor analysis, cluster analysis, and principal component analysis.

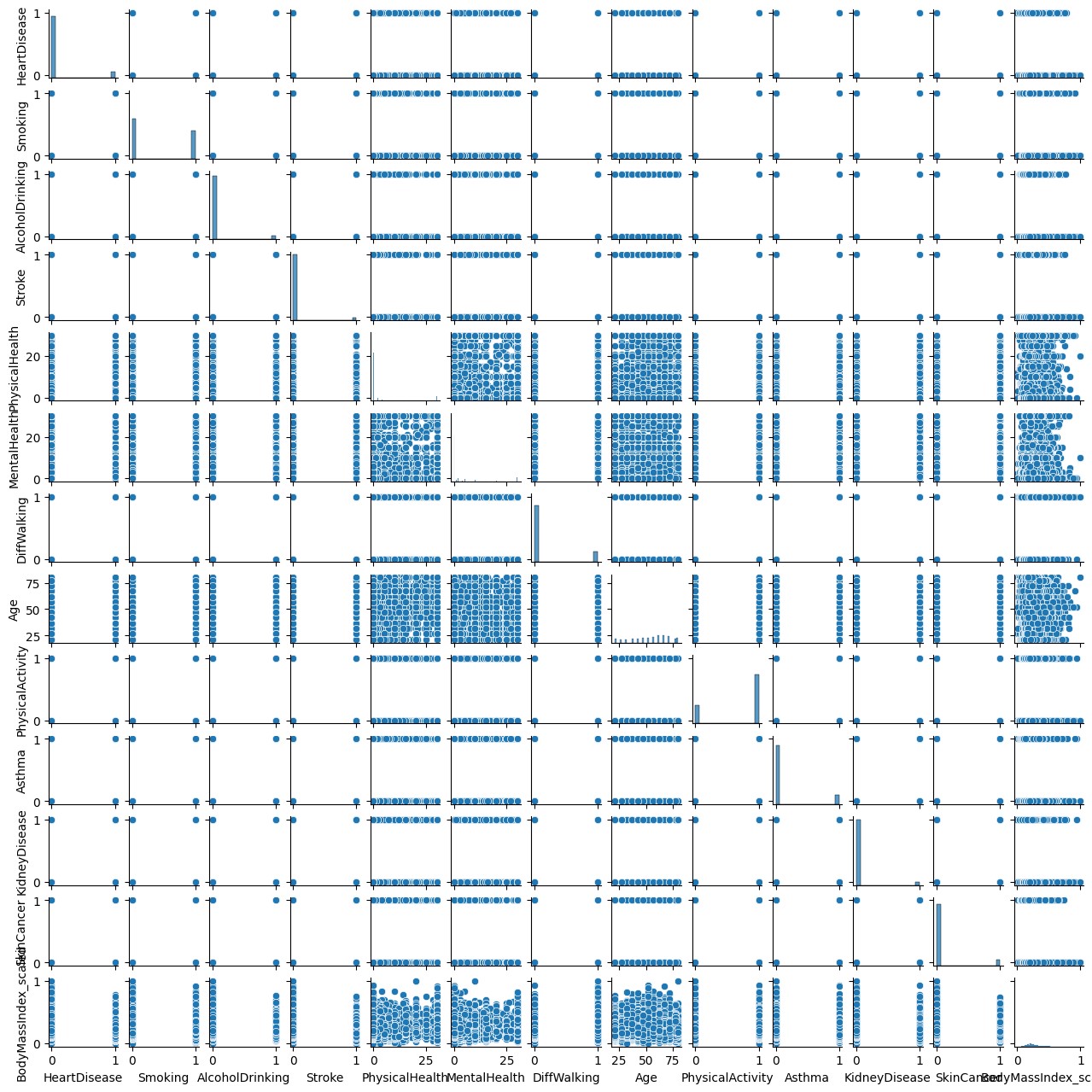
1. **Heatmaps:** Heatmaps depict relationships between either two categorical or two numerical variables through color representations, aiding in pattern and correlation identification, especially in extensive datasets.

This matrix illustrates variable interrelationships, employing libraries to render correlation matrices as heatmaps. Annotated with correlation values, the heatmap's color map indicates correlation strength and direction, presenting rectangular data in a color-encoded matrix format. Observing color shifts along each axis reveals patterns in variable values.



1. **Pair Plots:** Pair plots, also known as scatterplot matrices, display scatter plots for pairs of numerical variables in a grid format. They allow for the simultaneous visualization of relationships between multiple variables in the dataset, aiding in the identification of patterns and correlations.

In this pair-plot, the pairwise relationships between the two variables in the dataset can be plotted. It helps in creating a good visualization to understand the whole data in a single figure. And also univariate distribution can be drawn to show marginal distribution in each column.



**References**

<https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/code>