A Project On

"Analyzing Health Profile of Dhaka Cities People in Bangladesh: Employing Python for Statistical Inference and Linear Regression Insights."

Course Code: WM-ASDS04

Course Title: Introduction to Data Science with Python

SUBMITTED BY

Salaha Uddin Chowdhury Shaju ID:20231147
Mir Khalid Hassan ID:20231167
Md. Habibur Rahman ID:20231183

11th Batch Section B (Summer 2023)
Masters in Applied Statistics and Data Science (MS-ASDS)

Supervisor

Ms. Farhana Afrin Duty
Assistant Professor
Department of Statistics and Data Science
Jahangirnagar University

Department of Statistics and Data Science
Jahangirnagar University, Savar, Dhaka-1342, Bangladesh

22nd December, 2023

To

Ms. Farhana Afrin Duty

Assistant Professor

Department of Statistics and Data Science
Jahangirnagar University

Subject: Submission of project titled "Analyzing Health Profile of Dhaka Cities People in Bangladesh: Employing Python for Statistical Inference and Linear Regression Insights."

Dear Madam,

This is to inform you that we have prepared the project titled "Analyzing Health Profile of Dhaka Cities People in Bangladesh: Employing Python for Statistical Inference and Linear Regression Insights.", which has been prepared as a requirement for the completion of the course Introduction to Data Science and Python.

We have tried our level best to prepare the project report to the optimal standard under your valuable direction. Therefore, we hope that this will meet the standard of your judgment.

Thank you, for giving us the opportunity to write a formal report. It is an experience which will certainly help us in future. This will implement our analytical skills and also fulfill the requirement for the successful completion of our course.

Sincerely yours

Salaha Uddin Chowdhury Shaju (ID:20231147)

Mir Khalid Hassan (ID: 20231167)

Md. Habibur Rahman (ID:20231183)

“Analyzing Health Profile of Dhaka Cities People in Bangladesh: Employing Python for Statistical Inference and Linear Regression Insights.”

Abstract

This project delves into the intricate relationship among gender, hypertension stages, and body mass index (BMI) among visitors to parks in Dhaka, Bangladesh. In urban environments, such as Dhaka, where parks serve as vital recreational spaces, understanding the health dynamics of diverse populations engaging in outdoor activities is essential. The primary objective is to unravel the complex interplay between gender, hypertension, and BMI, providing a nuanced understanding through meticulous demographic analysis.

The study involves a comprehensive exploration of gender-based disparities in hypertension stages and BMI, aiming to quantify and characterize high-risk groups. Statistical inference techniques are employed to discern significant differences in BMI between genders and among various hypertension stages. A robust linear regression model is developed to quantify the strength and nature of the relationships between gender, hypertension stages, and BMI.

Demographic analysis, encompassing factors like age, enhances the contextual understanding of these health dynamics. The project not only identifies trends and patterns but also seeks to derive actionable insights for public health interventions. By disseminating findings to the public, policymakers, and healthcare professionals, this research aims to contribute to health awareness and inform strategies tailored to specific gender and hypertension subgroups. Ultimately, this project provides valuable insights into the health profiles of Dhaka Park visitors, contributing to a broader understanding of public health in urban recreational spaces.

Key words: *Body Mass Index (BMI), Demographic Variable, Hypertension, Linear Regression, Height, Weight*

ACKNOWLEDGEMENT

At the very beginning, we would like to convey our sincere appreciation to almighty Allah for giving me the strength and the ability to finish this project report.

It is with immense pleasure that we express our sincere gratitude to Farhana Afrin Duty, Assistant Professor in the Department of Statistics and Data Science at Jahangirnagar University, Savar, Dhaka. Throughout our academic journey, her unwavering support, continuous motivation, and valuable suggestions have been instrumental. We are privileged to have such a dynamic and punctual supervisor guiding our project. We remain indebted to her for her significant contributions and encouragement. Wishing her a life filled with health and prosperity.

Our gratitude extends to Interactive Professional Development Initiatives (IPDI Foundation) for generously providing the dataset and offering unwavering support throughout our endeavor.

Finally, we must express our very profound gratitude to our parents and to our family member for providing us with unfailing support and continuous encouragement throughout of study and through the process of researching and writing this project. This accomplishment would not have been possible without them.

Thanks to ALL!

Project Outline

1. Introduction

This research sets out to explore the complex web of health dynamics among visitors to Dhaka, Bangladesh, which lies away within the vibrant wallpaper of the city. These parks are essential spaces for leisure activities in the middle of busy cities life, drawing a diverse population interested in outdoor activities. among light of the vital role of these areas, this study explores the intricate relationships that exist between body mass index (BMI), gender, and stages of hypertension among park visitors in Dhaka.

Our primary objective is to put a spotlight on the complicated relationships that occur within this group by using statistically appropriate methods and analyses that are based on Python. By carefully examining demographic data, we hope to identify and describe subgroups at greater risk among park visitors in addition to studying gender disparities in BMI and hypertension stages.
We want to identify significant differences in BMI between genders and stages of hypertension by using advanced statistical inference techniques and building a strong linear regression model. Age-enriched demographic analysis will offer a contextual lens through which to view the complexities at work.

This project aims to provide practical insights for public health solutions, going beyond data exploration. Through the dissemination of findings to a wide range of participants, including consumers, policymakers, and healthcare professionals, our objective is to enhance public knowledge of diseases and customize interventions for certain gender and hypertension subgroups. Finally, this project seeks to bring light on the health characteristics of park visitors in Dhaka and improve our understanding of public health dynamics in urban areas.

1.1 Background

We want to identify significant differences in BMI between genders and stages of hypertension by using advanced statistical inference techniques and building a strong linear regression model. Age-enriched demographic analysis will offer a contextual lens through which to view the complexities at work.

The goal of this project is to investigate the complex interaction between body mass index (BMI) of park visitors, gender, and hypertension stages against the backdrop of Dhaka's busy metropolitan environment. Attracted to these essential green areas, city dwellers comprise a distinct population that covers a wide range of ages, professions, and lifestyles. It becomes essential to comprehend the interactions among BMI complexities, variations in hypertension, and gender-specific health inequities in order to support focused public health activities.

The main goal of the project is to analyze and interpret the health nuances contained in the information by using linear regression analysis and statistical inference based on Python. The research intends to assess high-risk populations and identify patterns that can guide customized solutions through thorough demographic analysis. This research aims to promote health awareness and impact methods that are relevant to the unique requirements of park visitors in Dhaka by sharing insights with a wide range of stakeholders, including the general public, policymakers, and healthcare professionals. In the end, this research aims to advance knowledge of public health dynamics in urban recreational settings, opening the door for well-informed and significant health initiatives in Bangladesh's capital of Dhaka.

1.2 Motivation of this project/ Scope

Health and well-being are paramount aspects of human life, and understanding the intricate interplay between lifestyle, demographic factors, and prevalent health conditions is crucial for informed decision-making and public health interventions. The motivation behind this project stems from the desire to delve deep into the health and lifestyle patterns within a specific community, shedding light on factors that influence overall well-being.

1.2.1 Key Motivations

Key motivations of the project are given below:

1.2.1.1 Public Health Impact

The project aims to contribute valuable insights to public health initiatives by examining the prevalence of lifestyle-related health conditions such as hypertension, diabetes, and obesity within the studied population.

1.2.1.2 Demographic Analysis

Understanding the demographic composition of the population allows for targeted health interventions, considering factors like age, gender, occupation, and education.

1.2.1.3 Lifestyle Choices

Analyzing lifestyle factors such as smoking habits provides a nuanced perspective on how individual choices contribute to overall health and well-being.

1.2.1.4 BMI and Health Correlation

Investigating the correlation between Body Mass Index (BMI) and health conditions helps in comprehending the impact of weight on prevalent diseases.

1.2.1.5 Predictive Modeling

The implementation of linear regression modeling seeks to predict health outcomes, offering a practical tool for assessing potential risk factors and making informed decisions.

1.2.1.6 Contribution to Research

The project contributes to the existing body of research by providing a comprehensive analysis of the relationships between demographic factors, lifestyle choices, and health outcomes.

1.2.1.7 Community-Specific Insights

Focusing on a specific community allows for tailored recommendations and interventions, addressing the unique health challenges faced by that population.

By uncovering these insights, the project aspires to foster a better understanding of health determinants, promoting evidence-based practices for healthier communities. This motivation drives the exploration of data and the subsequent analysis, aiming to make a positive impact on individual well-being and public health strategies.

2. Objectives

There are two types of objectives in this project. These are given below:

2.1 primary objective

The primary objective is to examine BMI variations which Scrutinize the distribution of BMI across different demographic groups, with a particular focus on gender-based variations.

2.2 Secondary Objective

There are many secondary objectives in this project which are:

1. Analyze and quantify any existing gender disparities in terms of hypertension stages and BMI among Dhaka Park visitors.

2. Investigate the distribution of hypertension stages within the studied population, discerning patterns and prevalence.

3. Identify and characterize groups that may be at a higher risk concerning hypertension and BMI, considering both gender and hypertension stages.

4. Conduct statistical tests to determine if there are significant differences in BMI between genders and among different hypertension stages.

5. Develop a robust linear regression model to quantify the relationship between gender, hypertension stages, and BMI, providing insights into the strength and nature of these associations.

6. Conduct a comprehensive demographic analysis, including age and other relevant factors, to contextualize the relationship between gender, hypertension stages, and BMI.

7. Uncover and interpret trends in the data, such as age-related variations or changes in BMI across different hypertension stages.

8. Derive actionable insights that can inform public health strategies and interventions targeted at specific gender and hypertension subgroups.

9. Contribute to health awareness by disseminating findings to the public, policymakers, and healthcare professionals, fostering a better understanding of the factors influencing health in urban recreational spaces like Dhaka parks.

3. Data Collection and Methodology

There are some steps for data collection and methodology. These are given below:

3.1 Source of Data

The dataset for this study was collected from the IPDI Foundation, which provided useful information about persons exercising in three known Dhaka parks: Ramna Park, Hatirjeel Park, and Dhanmondi Lake Park.

3.1.1 Data Provided by IPDI Foundation

The IPDI Foundation was instrumental in contributing the dataset, ensuring that the information collected agreed with the foundation's focus on health and well-being.

3.1.2 Participant Recruitment

The study involved contacting and welcoming participants to participate in person while they were in the park. Because participation was completely voluntary, anybody who genuinely wanted to be included in the study could.

3.1.3 Ethical Considerations

It is considered that ethical considerations were followed during the data gathering procedure to protect participant privacy and confidentiality. The IPDI Foundation has additional information about ethical practices.

3.1.4 Limitations

While the convenience sampling method made data collection more practical, it is important to note that the findings may primarily represent individuals who exercise in these specific park environments, potentially limiting generalizability to the broader population.

3.1.5 Data Quality Assurance

To assure the accuracy and reliability of the dataset provided by the IPDI Foundation, rigorous quality control methods may have been undertaken throughout data collection.

3.2. Sampling Method

The data was collected using a convenience sample strategy, which targeted persons who actively participated in exercise activities within the chosen park locations. This methodology was selected due to its applicability and effectiveness in gathering perspectives from people who are easily reachable in these leisure spaces.

3.2.1 Health and Lifestyle Factors

The dataset includes demographic details such as name, age, gender, occupation, and education level, providing a comprehensive profile of the participants. Economic status, height, weight, and BMI (Body Mass Index) are crucial health indicators that contribute to a holistic understanding of participants' well-being. Information on lifestyle factors and prevalent health conditions is included, such as smoking habits and the presence of diseases like hypertension, diabetes, dyslipidemia, stroke, and ischemic heart disease (IHD).

3.2.2 Categorization and Grouping

The data has been categorized into groups based on age, BMI, and hypertension stage. This grouping facilitates a more nuanced analysis and interpretation of the results.

4. Data Exploration and Visualization

Data exploration and visualization follows some technique. These are given below:

4.1 Dataset Overview

1. The dataset comprises 22 columns and 322 entries, representing various individual qualities.
2. Floating-point values are used for numerical variables such as age, height, weight, BMI, systolic and diastolic blood pressure, and RBS.
3. Categorical attributes, including name, sex, occupation, education, and health status, are indicated as objects.
4. The dataset provides a comprehensive overview of participants' demographics, health, and lifestyle traits.

Out[9]:

	Serial_No	Name	Age	Sex	Occupation	Education	Economy	Hight	weight	BMI	...	Smoking	HTN	DM	Dyslipidemia	Stroke	IHD	Age
0	DH1	Md.Sharif Hossain	69.0	1.0	2	5	4	171.0	69.0	23.597004	...	2.0	2.0	2.0	2.0	2.0	2.0	
1	DH2	Nasrin Malek	60.0	2.0	4	4	1	145.0	65.0	30.915577	...	2.0	1.0	1.0	1.0	2.0	1.0	
2	DH3	Golam Rabbani	54.0	1.0	5	4	3	175.0	75.0	24.489796	...	2.0	1.0	1.0	1.0	1.0	2.0	
3	DH4	Md.Salauddin Ahmed	65.0	1.0	5	5	4	170.0	71.0	24.567474	...	1.0	1.0	2.0	2.0	2.0	2.0	
4	DH5	Md.Johurul Islam	54.0	1.0	1	5	4	172.0	68.0	22.985398	...	2.0	2.0	1.0	2.0	2.0	2.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
317	HA127	Shekhon Das	43.0	1.0	2	5	4	171.0	78.0	26.674874	...	2.0	2.0	2.0	2.0	2.0	2.0	
318	HA128	Raja Mia	30.0	1.0	1	5	2	160.0	66.0	25.781250	...	1.0	2.0	2.0	2.0	2.0	2.0	
319	HA129	Soiib	25.0	1.0	5	5	1	171.0	76.0	25.990903	...	2.0	2.0	2.0	2.0	2.0	2.0	

4.2 Summary Statistics

1. The summary statistics table presents numerical characteristics, including distribution, central tendency, and variability of values.

2. Age ranges from 17 to 83 years, with an average age of approximately 51.12 years.

3. Standard deviations indicate the dataset's variability for each variable.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 322 entries, 0 to 321
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Serial_No          322 non-null    object
 1   Name               322 non-null    object
 2   Age                322 non-null    float64
 3   Sex                322 non-null    float64
 4   Occupation         322 non-null    object
 5   Education          322 non-null    object
 6   Economy            322 non-null    object
 7   Hight              322 non-null    float64
 8   weight             322 non-null    float64
 9   BMI                322 non-null    float64
 10  Systolic_Upper     322 non-null    float64
 11  Diastolic_Lower    322 non-null    float64
 12  RBS                320 non-null    float64
 13  Smoking            322 non-null    float64
 14  HTN                322 non-null    float64
 15  DM                 322 non-null    float64
 16  Dyslipidemia       322 non-null    float64
 17  Stroke             322 non-null    float64
 18  IHD                322 non-null    float64
 19  Age_group          322 non-null    float64
 20  BMI_Gr             322 non-null    float64
 21  Ag_g               322 non-null    float64
 22  Hypertenson_stage  322 non-null    float64
dtypes: float64(18), object(5)
memory usage: 58.0+ KB
```

4.3 Data Cleaning

1. During data processing, missing values were identified in the 'RBS' variable (2 entries).
2. Rows with missing values were removed to ensure data integrity.
3. The cleaned dataset, labeled as "df1," retains the same columns but may have fewer rows than the original 320

```
Original shape: (322, 23)
Cleaned shape: (320, 23)
```

4.4 Variables in Summary Table

1. The summary table includes statistics for height, weight, BMI, systolic and diastolic blood pressure, RBS, and age.
2. This table serves as a valuable tool for understanding the spread and distribution characteristics of each variable among the 320 dataset entries.

	Age	Hight	weight	BMI	Systolic_Upper	Diastolic_Lower	RBS
count	320.000000	320.000000	320.000000	320.000000	320.00000	320.000000	320.000000
mean	51.118750	164.803125	69.800625	25.809787	120.71875	78.750000	6.895219
std	11.634384	8.668141	10.035067	4.394856	15.05837	8.691869	2.917158
min	17.000000	103.000000	7.200000	2.324380	90.00000	60.000000	2.000000
25%	42.750000	160.000000	64.000000	23.447564	110.00000	70.000000	5.075000
50%	52.000000	166.000000	69.000000	25.417751	120.00000	80.000000	6.000000
75%	60.000000	170.000000	76.000000	27.665487	130.00000	80.000000	7.600000
max	83.000000	190.000000	95.000000	75.407673	170.00000	110.000000	19.400000

4.5 Quartile Analysis

1. The 25th percentile (1st quartile) age is approximately 42.75 years, signifying that 25% of participants are younger.

2. The median age is 52 years, providing insight into the central trend of the dataset.

3. Three-quarters of participants fall below the 75th percentile (3rd quartile) age of 60 years.

4.6 The distribution of key variables

```
Index(['Serial_No', 'Name', 'Age', 'Sex', 'Occupation', 'Education', 'Economy',
       'Hight', 'weight', 'BMI', 'Systolic_Upper', 'Diastolic_Lower', 'RBS',
       'Smoking', 'HTN', 'DM', 'Dyslipidemia', 'Stroke', 'IHD', 'Age_group',
       'BMI_Gr', 'Ag_g', 'Hypertenson_stage'],
      dtype='object')
```

The dataset contains the following columns:

1. Serial No: A serial number or identifier for each entry.
2. Name: The name of the individual represented in the entry.
3. Age: The age of the individual.
4. Sex: Gender of the individual (Male/Female).
5. Occupation: The occupation of the individual.

6. Education: Educational background or level of the individual.

7. Economy: Economic status of the individual.

8. Height: The height of the individual.

9. Weight: The weight of the individual.

10. BMI: Body Mass Index, a numerical measure of an individual's body fat based on their height and weight.

11. Systolic Upper: Systolic blood pressure reading.

12. Diastolic Lower: Diastolic blood pressure reading.

13. RBS: Random Blood Sugar level.

14. Smoking: Indicates whether the individual is a smoker.

15. HTN: Presence of Hypertension (High Blood Pressure).

16. DM: Presence of Diabetes Mellitus.

17. Dyslipidemia: Presence of abnormal lipid levels.

18. Stroke: Indicates whether the individual has a history of stroke.

19. IHD: Indicates whether the individual has ischemic heart disease.

20. Age group: Categorized age groups for better analysis.

21. BMI Gr: Categorized groups based on Body Mass Index.

22. Hypertension stage: Categorized stages of hypertension.

These columns provide a comprehensive overview of individual characteristics, health metrics, and demographic information, offering a rich dataset for exploring the health profiles of Dhaka Park visitors in relation to gender, hypertension stages, and BMI.

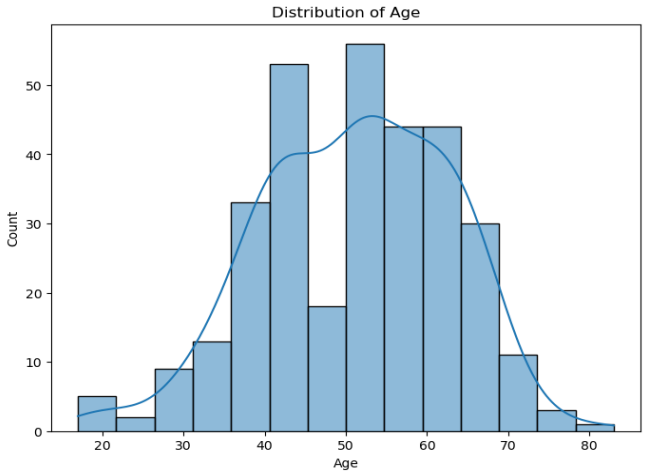


Figure 1: Distribution of Age

The distribution of Age chart shows us that the data are normally distributed. And the age group are between 20 years to 80 years with the mean value 50 years.

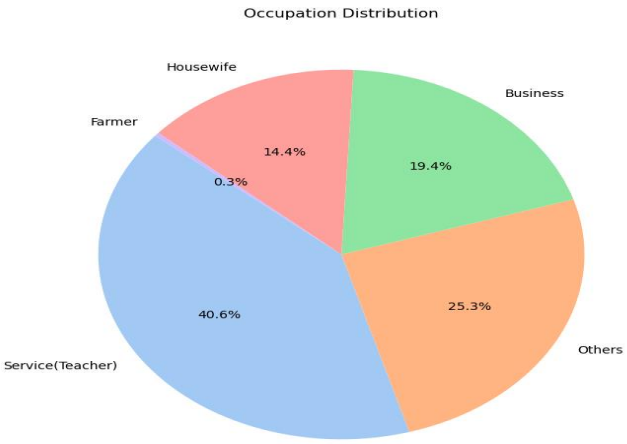


Figure 2: Pie Chart of Occupation Distribution

The Occupation Distribution pie chart shows us majority of the respondents are from service sector as a teacher (40.6%), others occupation (25.3%), Business (19.4%), Housewife (14.4%) and Farmer (0.3%) respectively.

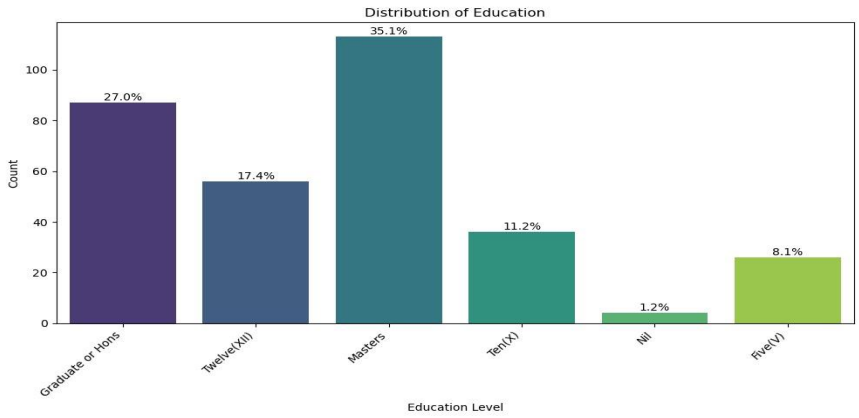


Figure 3: Distribution of Education

Distribution of education column chart shows us the respondents from different education background. These are Masters (35.1%), Graduate in Honors(27%), class twelve (17.4%), class Ten (11.2%), class Five (8.1%) and no Education (1.2%) respectively.

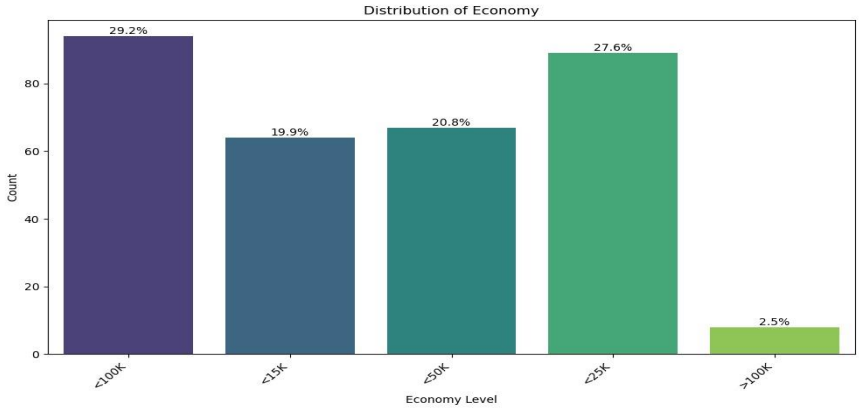


Figure 4: Distribution of Economy

According to the distribution of Economy chart we can find out the different level income respondents. Most of the data taken from less than 100k income (29.2%), less than 25k income (27.6%), less than 50k income (20.8%), less than 15k income (19.9%) and more than 100k income (2.5%) respectively.

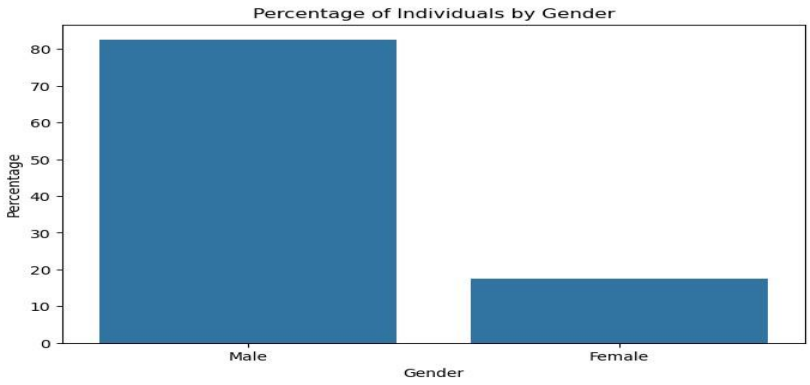


Figure 5 : Percentage of Individuals by Gender

Gender distribution column chart shows us that the male respondents' (81%) are greater than female respondents (19%).

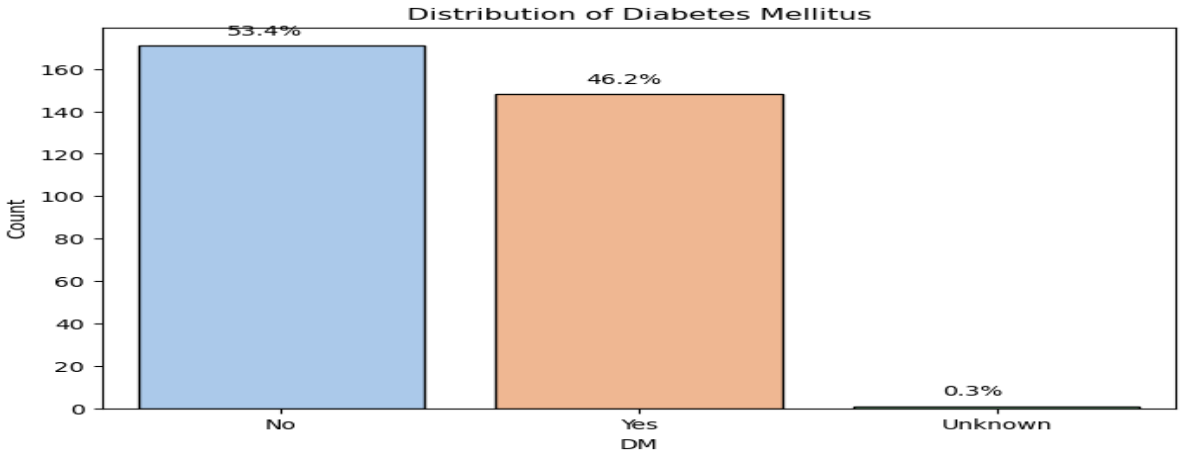


Figure 6: Distribution of Diabetes Mellitus

Regarding Diabetes Mellitus (DM), 148 participants (46.2%) reported having the condition, whereas 173 participants (53.4%) did not have a history of DM. Only one participant (0.3%) fell into the "Unknown" category.

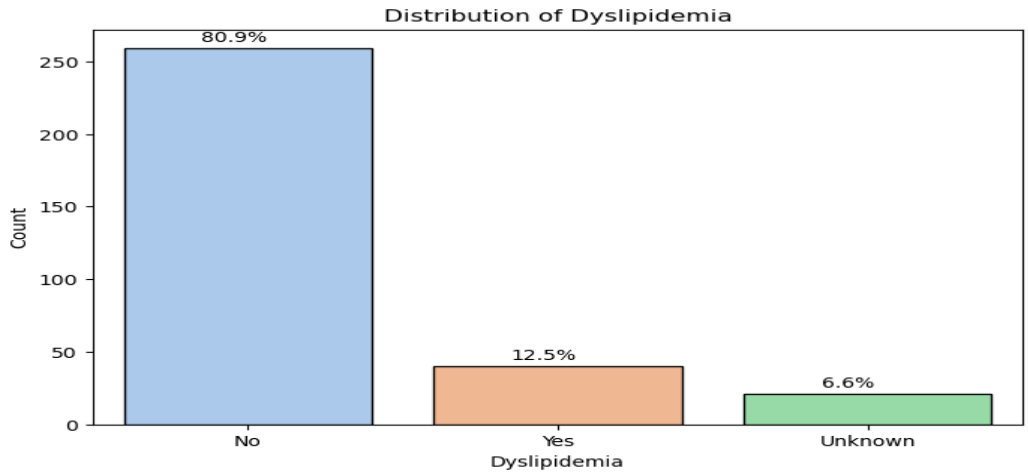


Figure 7: Distribution of Dyslipidemia

Concerning dyslipidemia, 40 participants (12.5%) acknowledged having this condition, while 261 participants (80.9%) did not report a history of dyslipidemia. Additionally, 21 participants (6.6%) were categorized as "Unknown."

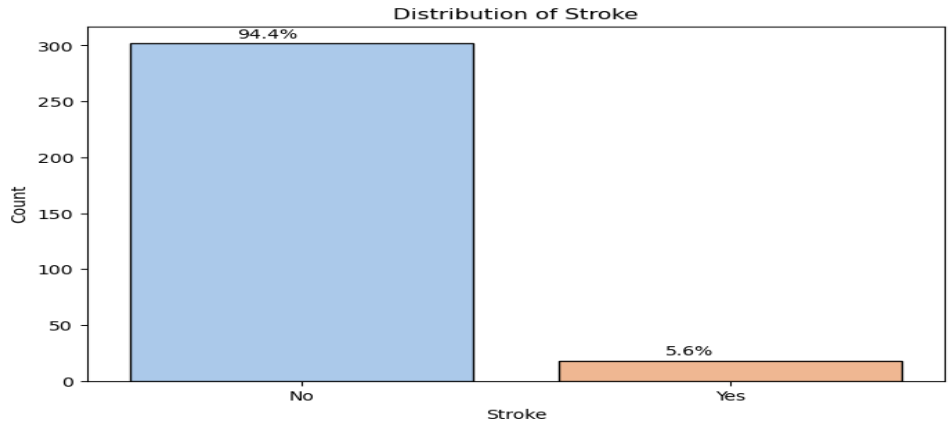


Figure 8: Distribution of Stroke

For those reporting a history of stroke, 18 participants (5.6%) indicated having experienced a stroke, whereas a significant majority of 304 participants (94.4%) did not report such a history.

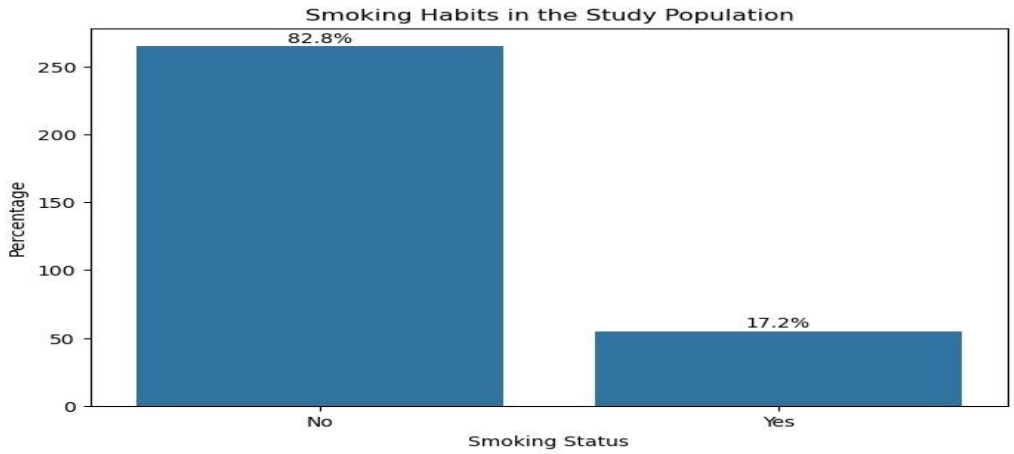


Figure 9: Smoking Habits in the Study Population

Among the participants, 55 (17.1%) reported being smokers. A significant majority, constituting 267 participants (82.9%), indicated that they do not smoke.

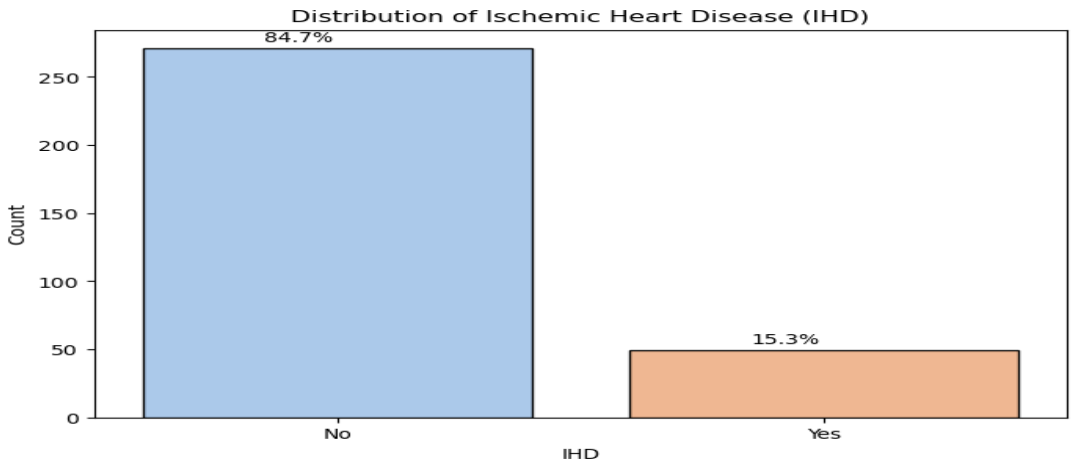


Figure 10: Distribution of Ischemic Heart Disease

In terms of Ischemic Heart Disease (IHD), 49 participants (15.3%) reported having IHD, while 273 participants (84.7%) did not have a history of IHD.

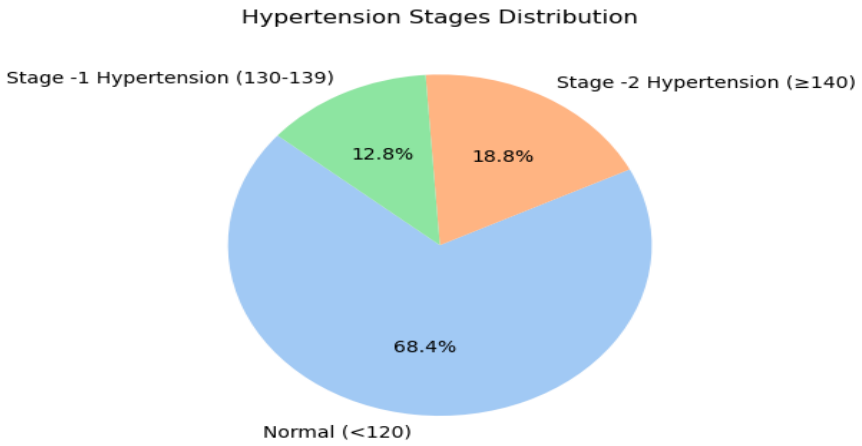


Figure 11: Pie Chart on Hypertension Stages Distribution

Hypertension Stages Distribution pie chart shows us that majority respondents pressure (68.4%) are in normal condition whereas stage 1 pressure (12.8%) and stage 2 pressure (18.8%) respectively.

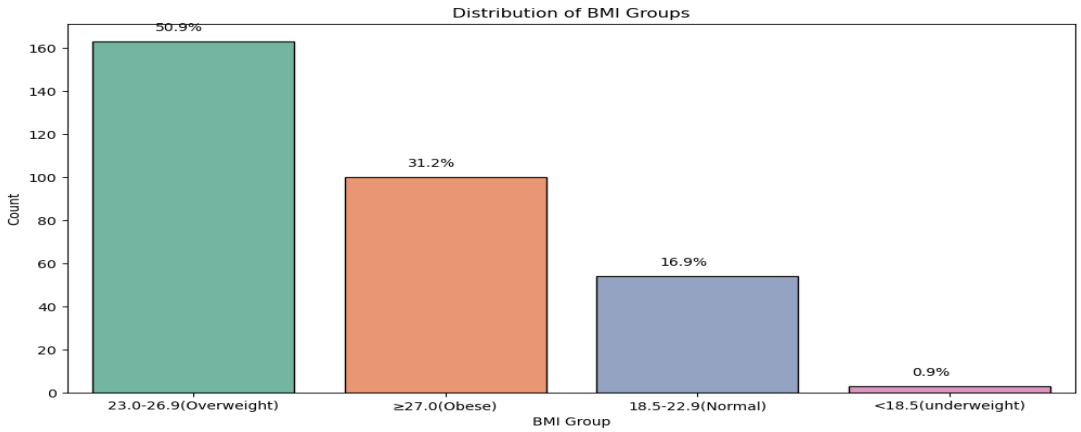


Figure 12: Distribution of BMI Groups

The Distribution chart shows us the BMI of different groups these are overweight (50.9%), Obese (31.2%), Normal (16.9%) and underweight (0.9%) respectively.

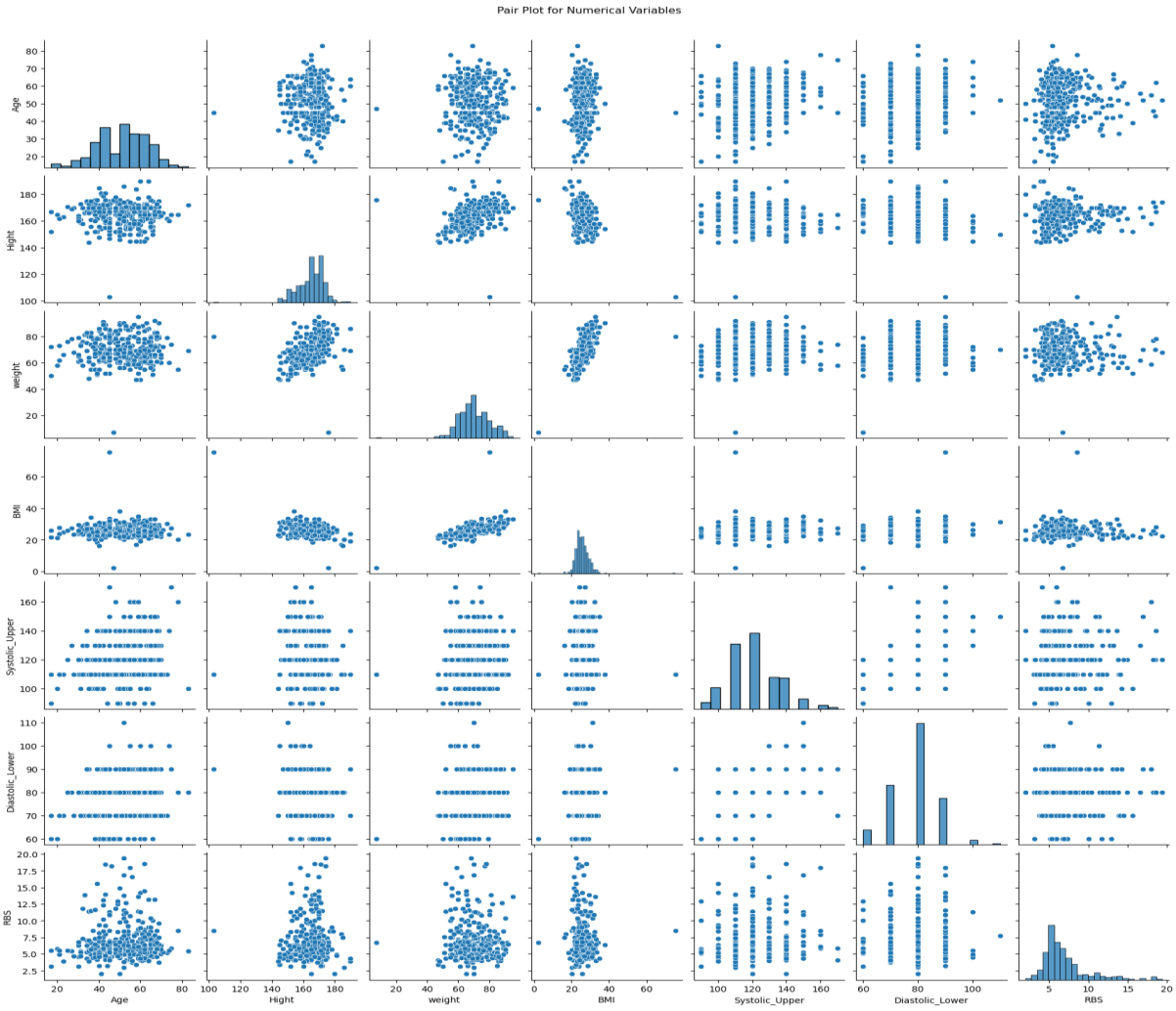


Figure 13: Pair Plot of Different Variables

Interpretation of pair plot represents; the off-diagonal plots are scatterplots for pairs of numerical variables. Each point in the scatterplot represents a data point, and the position of the point is determined by the values of the two variables on the x and y axes. These scatterplots help us visually assess the relationship between pairs of variables on the data set. By examining these scatterplots, we can identify potential relationships, patterns, or trends between pairs of numerical variables in our dataset. This visual exploration is helpful for understanding the underlying structure of our data and guiding further analyses or modeling decisions. We can visualize the relationship among the variables from this pair plot easily.

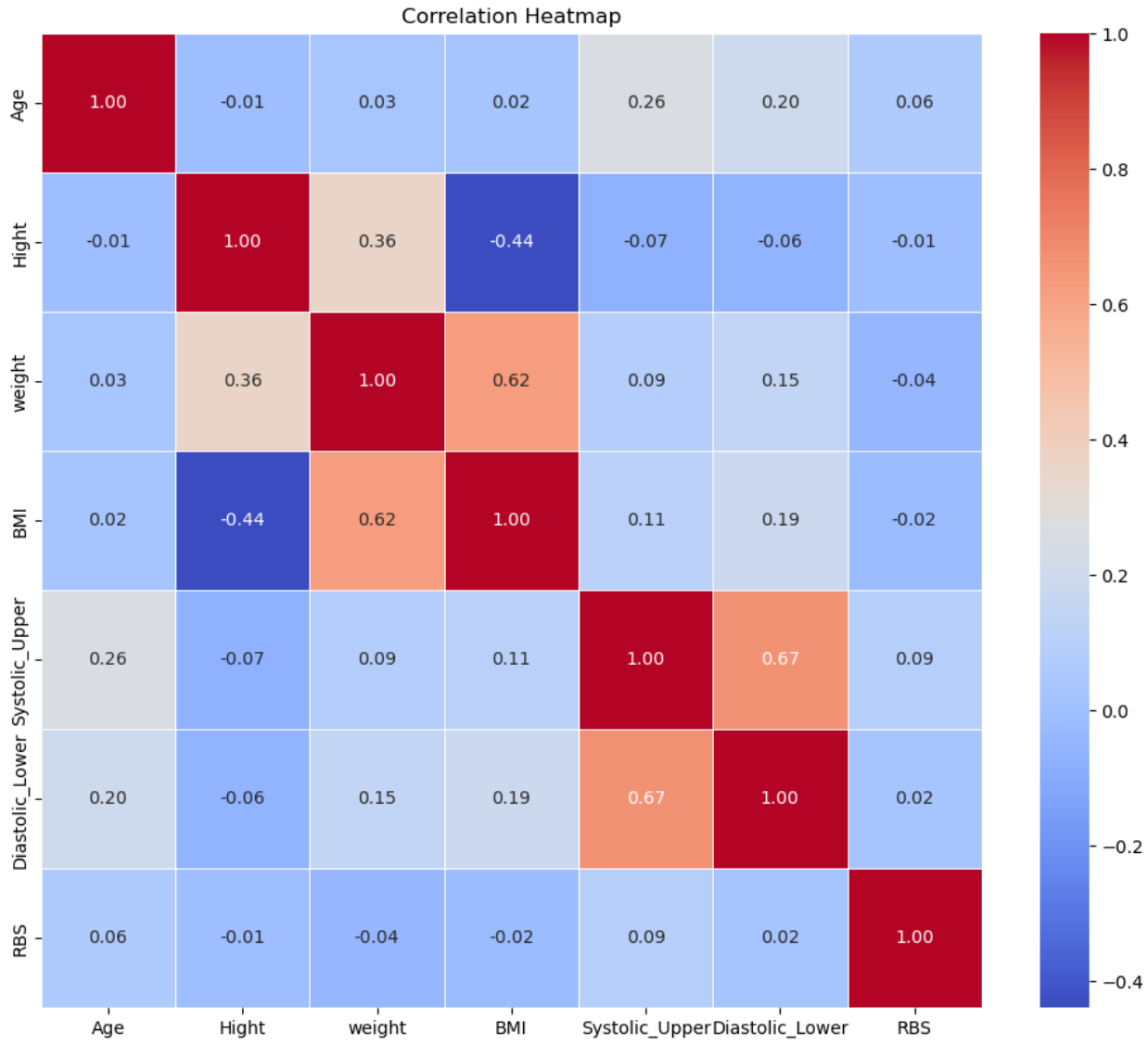


Figure 14: Correlation Heat map among the variables

The Correlation Heat map interpreting the relationships among the variables easily to identify the strong relation. Here the data between -1 to +1. Our heat map represents us that the BMI is strongly positive correlation to the weight (0.62), whereas the strong negative correlation with height (-0.44). And the systolic upper pressure is strongly correlate to the diastolic lower pressure (0.62). We can infer from the heat map to choose the independent variables for changing the BMI as a dependent variable in regression model.

5. Result Analysis

The result analysis steps are following some statistical technique and regression analysis. These are given below**:**

5.1 Statistical Inference

In our investigation using the chi-squared test, we explored the potential connection between two key categorical variables in our dataset: 'Sex' and 'BMI_Gr' (BMI Group). The results showed a chi-squared value of approximately 10.61 and a corresponding p-value of around 0.014. As our chosen significance level is 0.05, the p-value falls below this threshold, indicating a significant association between gender and BMI groups. This suggests that the distribution of BMI categories varies notably between males and females within our study cohort. These findings offer meaningful insights into the nuanced relationship between gender and BMI classification, presenting a noteworthy aspect within the scope of our research project.

```
Chi-squared value: 10.609143337159699
P-value: 0.014038502234043449
There is a significant relationship between Sex and BMI_Gr.
```

In the performed chi-squared test, we investigated the potential association between two categorical variables, 'Hypertenson_stage' (Hypertension Stage) and 'Sex' in our dataset.

```
Chi-squared value: 9.457102647881129
P-value: 0.008839267003376509
There is a significant relationship between Hypertenson_stage and Sex.
```

The results unveiled a chi-squared value of approximately 9.46 and a corresponding p-value of about 0.009. Given our chosen significance level of 0.05, the obtained p-value is below this threshold, indicating a statistically significant relationship between hypertension stage and gender. This suggests that the distribution of hypertension stages varies significantly between males and females within our study population. These findings provide valuable insights into the interrelation between gender and hypertension stages, offering a noteworthy aspect within the context of our research project.

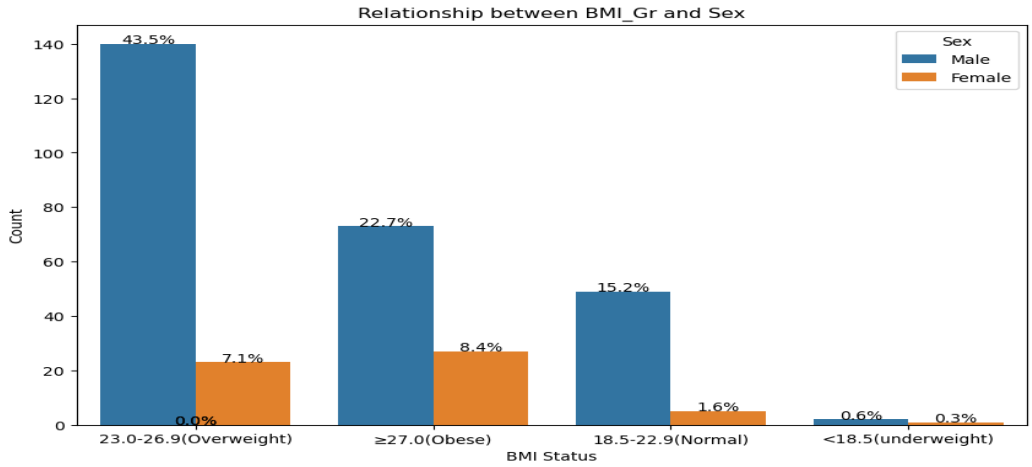


Figure 15: Relationship Chart between BMI Group and Sex

The relationship between BMI group and Sex chart shows us in every groups percentage of Male BMI are greater than Female BMI.

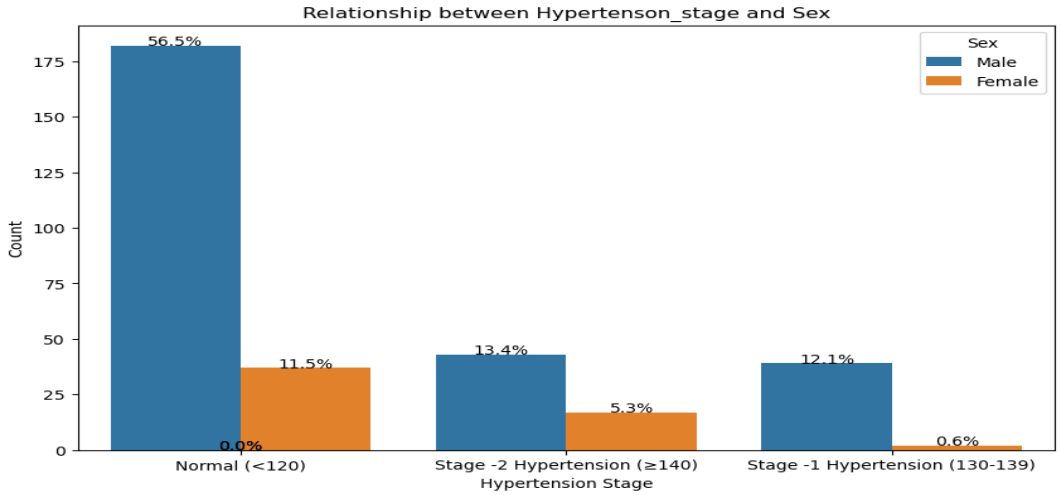


Figure 16: Comparison between Hypertension_Stage and Sex

In terms of hypertension (HTN), 102 participants (31.7%) were diagnosed with this condition, while 218 participants (67.7%) did not have a history of hypertension. Two participants (0.6%) fell under the "Unknown" category.

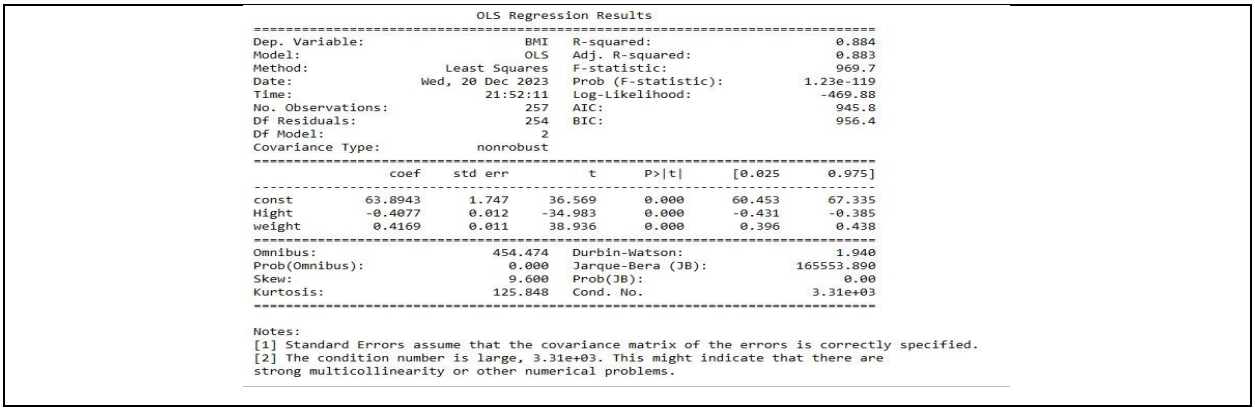
5.2 Splitting Data Set

In the provided code, the dataset (df2) is split into training and testing sets using the train_test_split function from the scikit-learn library. The features (X) and the target variable (y) are defined based on specified columns in the dataset. This separation is crucial to train the machine learning model on one subset of the data (training set) and evaluate its performance on another, unseen subset (testing set).

The linear regression model (Linear Regression) is then initialized and trained using the training set (X_train, y_train). Training involves adjusting the model's parameters to best fit the relationship between the features and the target variable. After training, the model is used to make predictions (y_pred) on the testing set (X_test). The performance of the model is evaluated using two metrics:

5.3 Regression Analysis and Model Evaluation

The linear regression model, trained on a subset of the dataset and evaluated on a separate testing set, exhibits promising performance. The Mean Squared Error (MSE), a measure of prediction accuracy, is approximately 1.14, indicating that, on average, the model's predictions closely align with the actual values. Furthermore, the R-squared (R2) value, which gauges the model's ability to explain variance in the target variable, stands at around 0.884. This high R2 value suggests that the model captures about 88.4% of the variability in the target variable. These metrics collectively imply that the linear regression model demonstrates robust predictive capabilities and effectively explains the observed patterns in the data. However, it's crucial to interpret these results within the specific context of your project goals and criteria for model performance.



OLS Regression Results

Dep. Variable:	BMI	R-squared:	0.884
Model:	OLS	Adj. R-squared:	0.883
Method:	Least Squares	F-statistic:	969.7
Date:	Wed, 20 Dec 2023	Prob (F-statistic):	1.23e-119
Time:	21:52:11	Log-Likelihood:	-469.88
No. Observations:	257	AIC:	945.8
Df Residuals:	254	BIC:	956.4
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	63.8943	1.747	36.569	0.000	60.453	67.335
Hight	-0.4077	0.012	-34.983	0.000	-0.431	-0.385
weight	0.4169	0.011	38.936	0.000	0.396	0.438

Omnibus:	454.474	Durbin-Watson:	1.940
Prob(Omnibus):	0.000	Jarque-Bera (JB):	165553.890
Skew:	9.600	Prob(JB):	0.00
Kurtosis:	125.848	Cond. No.	3.31e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.31e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

The interpretation about comparison of line graph, by comparing the circles (actual values) with the crosses (predicted values), we can visually assess how well the regression model performs. Ideally, the crosses should closely align with the circles, indicating that the model's predictions are in good agreement with the actual BMI values. From the line graph we have seen that maximum predicted values of BMI are closely related to the actual values of BMI. So we can say that the model is best fitted for the given data we have observed.

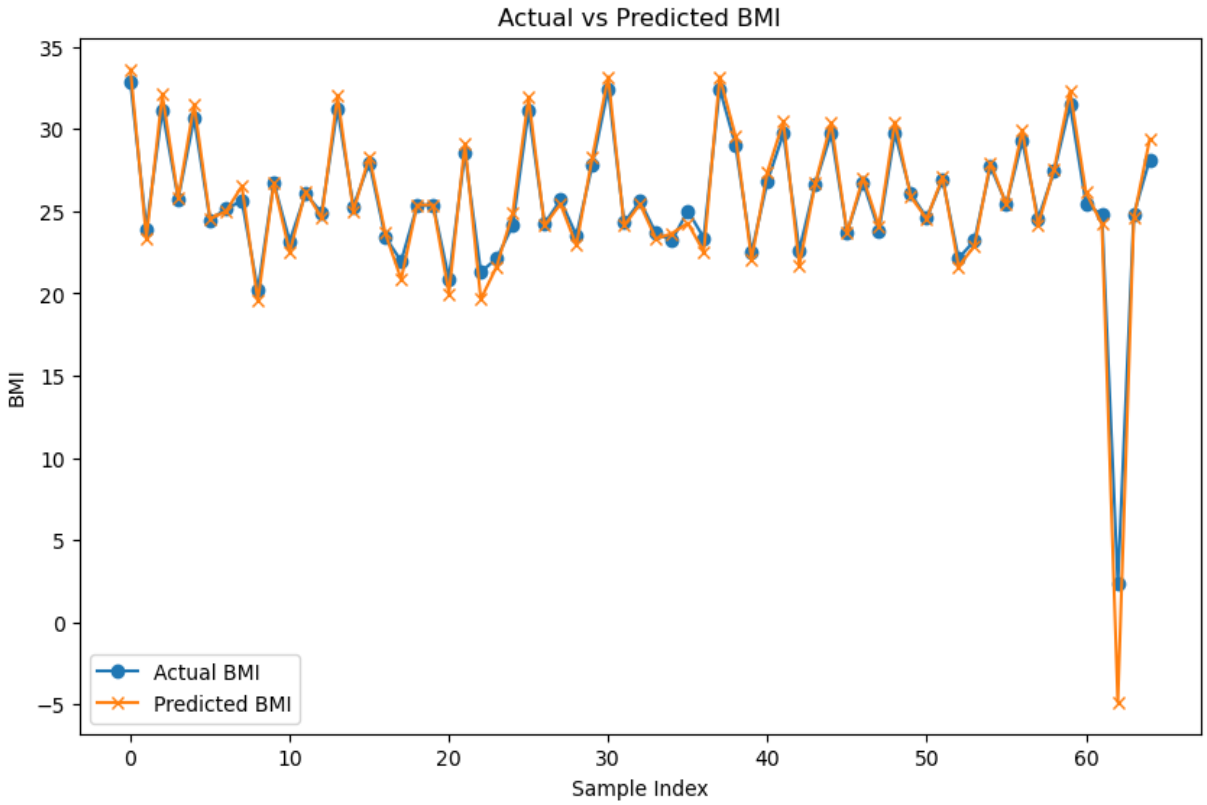


Figure 17: Comparison between Actual vs Predicted Values

We also find out the comparison between actual value vs predicted value which is shown in figure 17 and figure 18. We have seen that most of the time predicted value is closer to the actual values.

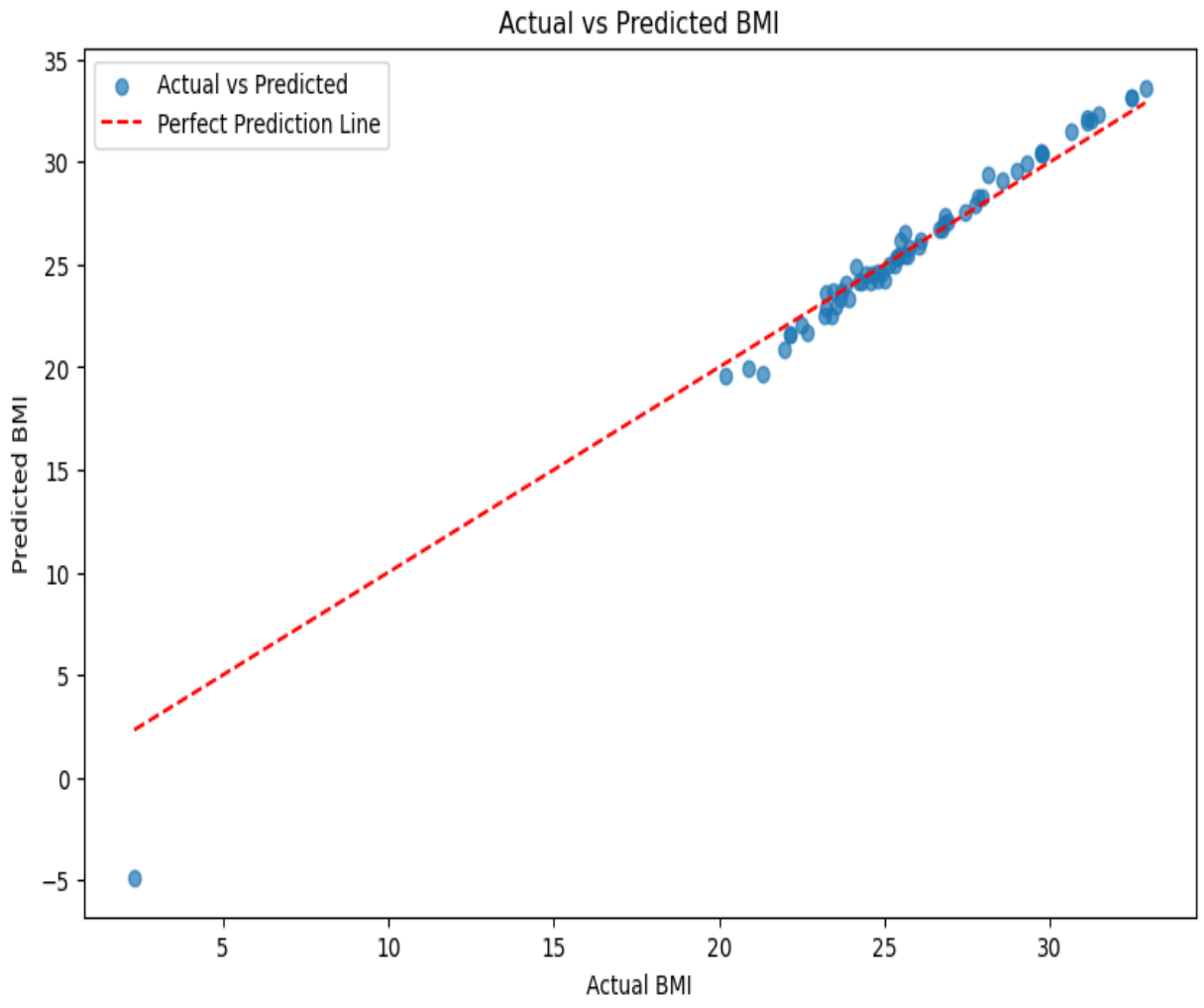


Figure 18: Linear Regression on Actual vs Predicted BMI

Interpretation of the linear regression model, the red regression line represents the linear relationship between the actual and predicted BMI values. The dotted red line is the predicted regression model and blue circle represent the actual BMI of observed data. This line is the result of the linear regression model's attempt to best fit the data. In an ideal scenario, the points should be close to this line, indicating that the predicted values closely match the actual values. A well-fitted model would show a tight cluster of points around the regression line, indicating a strong correlation between actual and predicted values

6. Conclusion

Our study presents a successful development and evaluation of a robust linear regression model for BMI prediction, demonstrating strong performance with an MSE of 1.14 and an R2 value of 0.884. Notably, chi-squared tests revealed significant associations between gender and BMI groups ($p = 0.014$) and hypertension stages ($p = 0.009$), emphasizing the relevance of gender-specific considerations in health classifications. These insights contribute valuable knowledge to the interplay between demographic variables and health outcomes. While our study provides a solid foundation, we recognize the potential for further investigation with additional variables and a larger dataset to enhance generalizability. Our research aims to guide healthcare interventions for specific demographic groups, making meaningful contributions to the field of health analytics and warranting consideration for international journal submission

7. Reference

1. Dementia [Internet]. World Health Organization; [cited 2023 Dec 21]. Available from: https://www.who.int/news-room/fact-sheets/detail/dementia
2. [Internet]. U.S. Department of Health and Human Services; [cited 2023 Dec 21]. Available from: https://www.nih.gov/
3. Desai RA, Manley M, Desai MM, Potenza MN. Gender differences in the association between body mass index and psychopathology. CNS Spectrums. 2009;14(7):372–83. doi:10.1017/s1092852900023026
4. Zhang J, Xu L, Li J, Sun L, Qin W, Ding G, et al. Gender differences in the association between body mass index and health-related quality of life among adults:a cross-sectional study in Shandong, China. BMC Public Health. 2019;19(1). doi:10.1186/s12889-019-7351-7
5. Connelly PJ, Currie G, Delles C. Sex differences in the prevalence, outcomes and management of hypertension. Current Hypertension Reports. 2022;24(6):185–92. doi:10.1007/s11906-022-01183-8
6. Reckelhoff JF. Gender differences in the regulation of Blood Pressure. Hypertension. 2001;37(5):1199–208. Doi: 10.1161/01.hyp.37.5.1199
7. Oscullo, Less, Kishi, Armanini, More, Yarlioglues, et al. [Internet]. 2023 [cited 2023 Dec 21]. Available from: https://journals.lww.com/jhypertension/pages/default.aspx

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pyreadstat
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:
```
# Read the SPSS data file
df, meta = pyreadstat.read_sav('project.sav')

# Display the DataFrame
print(df.head())

# Display the metadata (variable labels, value labels, etc.)
print(meta)
```

```
  Serial_No                Name   Age  Sex Occupation Education Economy  \
0       DH1   Md.Sharif Hossain  69.0  1.0          2         5       4
1       DH2        Nasrin Malek  60.0  2.0          4         4       1
2       DH3       Golam Rabbani  54.0  1.0          5         4       3
3       DH4  Md.Salauddin Ahmed  65.0  1.0          5         5       4
4       DH5    Md.Johurul Islam  54.0  1.0          1         5       4

   Hight  weight        BMI  ...  RBS  Smoking  HTN   DM  Dyslipidemia  \
0  171.0    69.0  23.597004  ...  3.7      2.0  2.0  2.0           2.0
1  145.0    65.0  30.915577  ...  5.0      2.0  1.0  1.0           1.0
2  175.0    75.0  24.489796  ...  7.4      2.0  1.0  1.0           1.0
3  170.0    71.0  24.567474  ...  5.1      1.0  1.0  2.0           2.0
4  172.0    68.0  22.985398  ...  6.7      2.0  2.0  1.0           2.0

   Stroke  IHD  Age_group  BMI_Gr  Hypertenson_stage
0     2.0  2.0        6.0     3.0                1.0
1     2.0  1.0        5.0     4.0                4.0
2     1.0  2.0        5.0     3.0                4.0
3     2.0  2.0        6.0     3.0                1.0
4     2.0  2.0        5.0     3.0                3.0

[5 rows x 22 columns]
<pyreadstat._readstat_parser.metadata_container object at 0x000001DE64C1A910>
```

In [3]:
```
# Check for missing values
print(df.isnull().sum())
```

```
Serial_No           0
Name                0
Age                 0
Sex                 0
Occupation          0
Education           0
Economy             0
Hight               0
weight              0
BMI                 0
Systolic_Upper      0
Diastolic_Lower     0
RBS                 2
Smoking             0
HTN                 0
DM                  0
Dyslipidemia        0
Stroke              0
IHD                 0
Age_group           0
BMI_Gr              0
Hypertenson_stage   0
dtype: int64
```

In [22]:
```
# Drop rows with missing values
df1 = df.dropna()

# Display the shape of the cleaned DataFrame
print("Original shape:", df1.shape)
print("Cleaned shape:", df1.shape)
```

```
Original shape: (320, 22)
Cleaned shape: (320, 22)
```

In [66]:
```
df1
```

4	DH5	Md.Johurul Islam	54.0	Male	Service(Teacher)	Graduate or Hons	<100K	172.0	6
...	...	...	...	...	...	...	...	...	
316	HA126	Sofiqul Islam	62.0	Male	Others	Five(V)	<15K	165.0	7
318	HA128	Raja Mia	30.0	Male	Service(Teacher)	Graduate or Hons	<25K	160.0	6
319	HA129	Sojib	25.0	Male	Others	Graduate or Hons	<15K	171.0	7
320	HA130	Amdad Hossain	65.0	Male	Others	Graduate or Hons	<25K	170.0	7
321	HA131	Masuma Begum	50.0	Female	Housewife	Five(V)	<15K	165.0	7

320 rows × 23 columns

In [ ]:
```

```

In [23]:
```
# Display the DataFrame
print(df1.head())

# Display the metadata (variable labels, value labels, etc.)
print(meta.variable_value_labels)
```

```
  Serial_No                Name   Age     Sex        Occupation  \
0       DH1   Md.Sharif Hossain  69.0    Male          Business
1       DH2         Nasrin Malek  60.0  Female         Housewife
2       DH3       Golam Rabbani  54.0    Male            Others
3       DH4  Md.Salauddin Ahmed  65.0    Male            Others
4       DH5    Md.Johurul Islam  54.0    Male  Service(Teacher)

          Education Economy  Hight  weight        BMI  ...  RBS  Smoking  HTN
\
0  Graduate or Hons   <100K  171.0    69.0  23.597004  ...  3.7       No   No
1       Twelve(XII)    <15K  145.0    65.0  30.915577  ...  5.0       No  Yes
2       Twelve(XII)    <50K  175.0    75.0  24.489796  ...  7.4       No  Yes
3  Graduate or Hons   <100K  170.0    71.0  24.567474  ...  5.1      Yes  Yes
4  Graduate or Hons   <100K  172.0    68.0  22.985398  ...  6.7       No   No

    DM Dyslipidemia Stroke  IHD Age_group                 BMI_Gr  \
0   No           No     No   No     61-70  23.0-26.9(Overweight)
1  Yes          Yes     No  Yes     51-60          ≥27.0(Obese)
2  Yes          Yes    Yes   No     51-60  23.0-26.9(Overweight)
3   No           No     No   No     61-70  23.0-26.9(Overweight)
4  Yes           No     No   No     51-60  23.0-26.9(Overweight)

                Hypertenson_stage
0                  Normal (<120)
1     Stage -2 Hypertension (≥140)
2     Stage -2 Hypertension (≥140)
3                  Normal (<120)
4  Stage -1 Hypertension (130-139)

[5 rows x 22 columns]
{'Sex': {1.0: 'Male', 2.0: 'Female'}, 'Occupation': {'1': 'Service(Teacher)',
'2': 'Business', '3': 'Farmer', '4': 'Housewife', '5': 'Others'}, 'Educatio
n': {'1': 'Nil', '2': 'Five(V)', '3': 'Ten(X)', '4': 'Twelve(XII)', '5': 'Gra
duate or Hons', '6': 'Masters'}, 'Economy': {'1': '<15K', '2': '<25K', '3':
'<50K', '4': '<100K', '5': '>100K'}, 'Smoking': {1.0: 'Yes', 2.0: 'No'}, 'HT
N': {1.0: 'Yes', 2.0: 'No', 3.0: 'Unknown'}, 'DM': {1.0: 'Yes', 2.0: 'No', 3.
0: 'Unknown'}, 'Dyslipidemia': {1.0: 'Yes', 2.0: 'No', 3.0: 'Unknown'}, 'Stro
ke': {1.0: 'Yes', 2.0: 'No', 3.0: 'Unknown'}, 'IHD': {1.0: 'Yes', 2.0: 'No',
3.0: 'Unknown'}, 'Age_group': {1.0: '<20', 2.0: '20-30', 3.0: '31-40', 4.0:
'41-50', 5.0: '51-60', 6.0: '61-70', 7.0: '71-80', 8.0: '>80'}, 'BMI_Gr': {1.
0: '<18.5(underweight)', 2.0: '18.5-22.9(Normal)', 3.0: '23.0-26.9(Overweigh
t)', 4.0: '≥27.0(Obese)'}, 'Hypertenson_stage': {1.0: 'Normal (<120)', 2.0:
'Elevated (120-129)', 3.0: 'Stage -1 Hypertension (130-139)', 4.0: 'Stage -2
Hypertension (≥140)', 5.0: 'Hypertensive Crisis (>180)'}}
```

In [68]:
```
# Display the DataFrame with value labels
value_labels = meta.variable_value_labels

for column, labels in value_labels.items():
    df1[column] = df1[column].replace(labels)

# Display the updated DataFrame
print(df1.head())
```

```
  Serial_No                Name   Age     Sex        Occupation  \
0       DH1   Md.Sharif Hossain  69.0    Male          Business
1       DH2        Nasrin Malek  60.0  Female         Housewife
2       DH3       Golam Rabbani  54.0    Male            Others
3       DH4  Md.Salauddin Ahmed  65.0    Male            Others
4       DH5    Md.Johurul Islam  54.0    Male  Service(Teacher)

          Education Economy  Hight  weight        BMI  ...  Smoking  HTN   DM
\
0  Graduate or Hons   <100K  171.0    69.0  23.597004  ...       No   No   No
1       Twelve(XII)    <15K  145.0    65.0  30.915577  ...       No  Yes  Yes
2       Twelve(XII)    <50K  175.0    75.0  24.489796  ...       No  Yes  Yes
3  Graduate or Hons   <100K  170.0    71.0  24.567474  ...      Yes  Yes   No
4  Graduate or Hons   <100K  172.0    68.0  22.985398  ...       No   No  Yes

  Dyslipidemia Stroke  IHD Age_group                 BMI_Gr   Ag_g  \
0           No     No   No     61-70  23.0-26.9(Overweight)  61-70
1          Yes     No  Yes     51-60           ≥27.0(Obese)  51-60
2          Yes    Yes   No     51-60  23.0-26.9(Overweight)  51-60
3           No     No   No     61-70  23.0-26.9(Overweight)  61-70
4           No     No   No     51-60  23.0-26.9(Overweight)  51-60

                Hypertenson_stage
0                   Normal (<120)
1     Stage -2 Hypertension (≥140)
2     Stage -2 Hypertension (≥140)
3                   Normal (<120)
4  Stage -1 Hypertension (130-139)

[5 rows x 23 columns]

C:\Users\ipdi2\AppData\Local\Temp\ipykernel_21116\1640471111.py:5: SettingWit
hCopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df1[column] = df1[column].replace(labels)
```

In [13]:
```
df1
```

Out[13]:

	Serial_No	Name	Age	Sex	Occupation	Education	Economy	Hight	weight	
0	DH1	Md.Sharif Hossain	69.0	1.0	2	5	4	171.0	69.0	23
1	DH2	Nasrin Malek	60.0	2.0	4	4	1	145.0	65.0	30
2	DH3	Golam Rabbani	54.0	1.0	5	4	3	175.0	75.0	24
3	DH4	Md.Salauddin Ahmed	65.0	1.0	5	5	4	170.0	71.0	24
4	DH5	Md.Johurul Islam	54.0	1.0	1	5	4	172.0	68.0	22
...	...	...	...	...	...	...	...	...	...	...
316	HA126	Sofiqul Islam	62.0	1.0	5	2	1	165.0	70.0	25
318	HA128	Raja Mia	30.0	1.0	1	5	2	160.0	66.0	25
319	HA129	Sojib	25.0	1.0	5	5	1	171.0	76.0	25

In [24]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 320 entries, 0 to 321
Data columns (total 22 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial_No          320 non-null    object 
 1   Name               320 non-null    object 
 2   Age                320 non-null    float64
 3   Sex                320 non-null    object 
 4   Occupation         320 non-null    object 
 5   Education          320 non-null    object 
 6   Economy            320 non-null    object 
 7   Hight              320 non-null    float64
 8   weight             320 non-null    float64
 9   BMI                320 non-null    float64
 10  Systolic_Upper     320 non-null    float64
 11  Diastolic_Lower    320 non-null    float64
 12  RBS                320 non-null    float64
 13  Smoking            320 non-null    object 
 14  HTN                320 non-null    object 
 15  DM                 320 non-null    object 
 16  Dyslipidemia       320 non-null    object 
 17  Stroke             320 non-null    object 
 18  IHD                320 non-null    object 
 19  Age_group          320 non-null    object 
 20  BMI_Gr             320 non-null    object 
 21  Hypertenson_stage  320 non-null    object 
dtypes: float64(7), object(15)
memory usage: 57.5+ KB
```

In [11]:
```
# Assuming your DataFrame is named 'df'
data_info = pd.DataFrame({
    'Column': df1.columns,
    'Non-Null Count': df.count(),
    'Dtype': df1.dtypes
})

display(data_info)
```

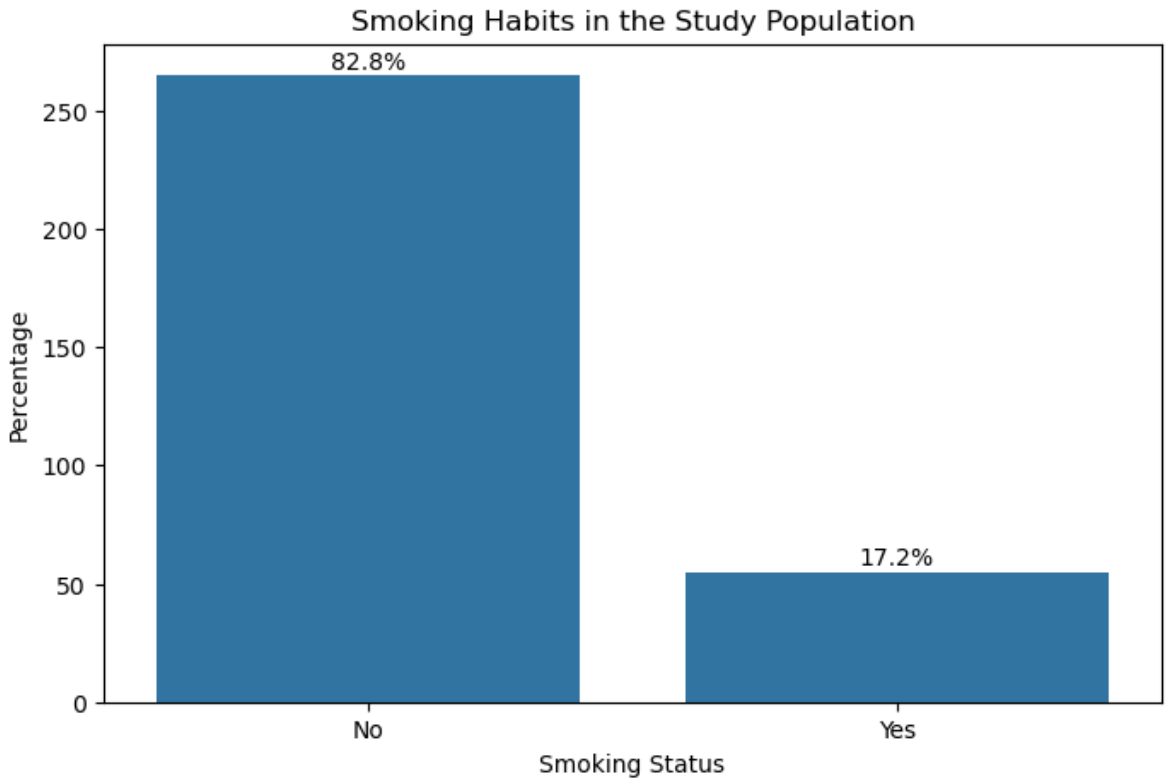
	Column	Non-Null Count	Dtype
Serial_No	Serial_No	322	object
Name	Name	322	object
Age	Age	322	float64
Sex	Sex	322	float64
Occupation	Occupation	322	object
Education	Education	322	object
Economy	Economy	322	object
Hight	Hight	322	float64
weight	weight	322	float64
BMI	BMI	322	float64
Systolic_Upper	Systolic_Upper	322	float64
Diastolic Lower	Diastolic Lower	322	float64

In [38]:
```
import pandas as pd

# Assuming your data is stored in a DataFrame called df
# Replace the column names accordingly

variables_of_interest = ['Smoking', 'HTN', 'DM', 'Dyslipidemia', 'Stroke', 'IH

for variable in variables_of_interest:
    frequency_table = df[variable].value_counts().reset_index().rename(columns
    percentage_table = pd.DataFrame(df[variable].value_counts(normalize=True)

    result_table = pd.merge(frequency_table, percentage_table, on=variable)

    print(f"\n{variable}:\n{result_table}\n{'-' * 50}")
```

```
Smoking:
  Smoking  Frequency  Percentage
0      No        267   82.919255
1     Yes         55   17.080745
--------------------------------------------------

HTN:
       HTN  Frequency  Percentage
0       No        218   67.701863
1      Yes        102   31.677019
2  Unknown          2    0.621118
--------------------------------------------------

DM:
        DM  Frequency  Percentage
0       No        173   53.726708
1      Yes        148   45.962733
2  Unknown          1    0.310559
--------------------------------------------------

Dyslipidemia:
  Dyslipidemia  Frequency  Percentage
0           No        261   81.055901
1          Yes         40   12.422360
2      Unknown         21    6.521739
--------------------------------------------------

Stroke:
  Stroke  Frequency  Percentage
0     No        304   94.409938
1    Yes         18    5.590062
--------------------------------------------------

IHD:
   IHD  Frequency  Percentage
0   No        273   84.782609
1  Yes         49   15.217391
--------------------------------------------------

Age_group:
  Age_group  Frequency  Percentage
0     51-60         95   29.503106
1     41-50         89   27.639752
2     61-70         70   21.739130
3     31-40         49   15.217391
4     20-30         10    3.105590
5     71-80          5    1.552795
6       <20          3    0.931677
7       >80          1    0.310559
--------------------------------------------------

BMI_Gr:
                  BMI_Gr  Frequency  Percentage
0  23.0-26.9(Overweight)        164   50.931677
1           ≥27.0(Obese)        100   31.055901
2      18.5-22.9(Normal)         55   17.080745
```

```
3     <18.5(underweight)           3    0.931677
--------------------------------------------------

Hypertenson_stage:
                 Hypertenson_stage  Frequency  Percentage
0                    Normal (<120)        220   68.322981
1     Stage -2 Hypertension (≥140)         60   18.633540
2  Stage -1 Hypertension (130-139)         42   13.043478
--------------------------------------------------
```

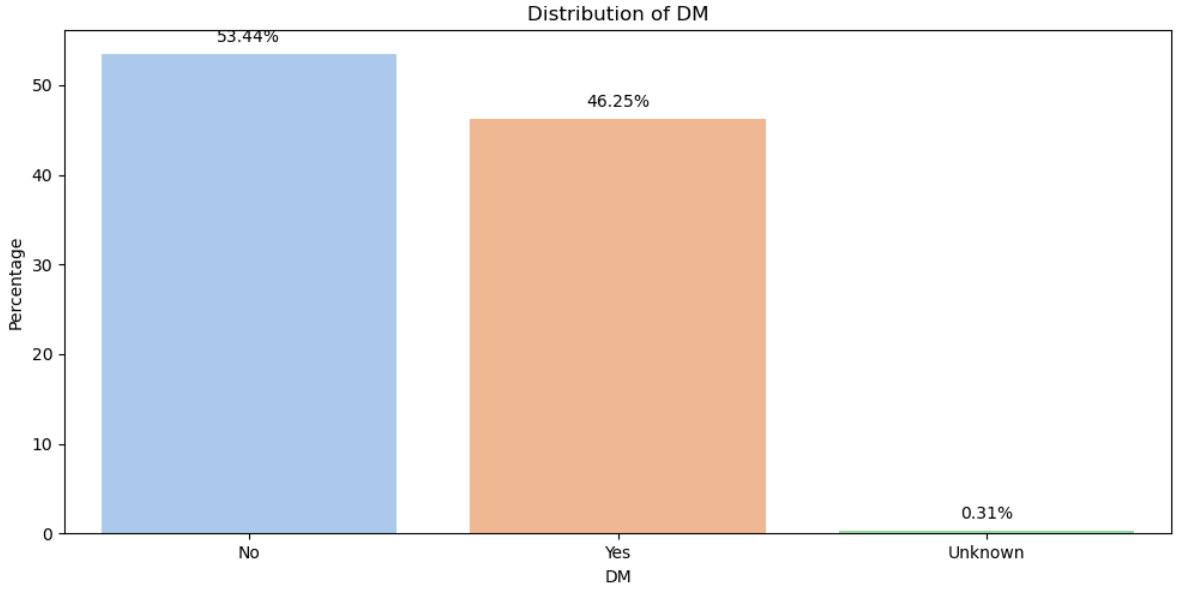
In [39]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Countplot of Smoking with percentage
plt.figure(figsize=(8, 5))
total = len(df1['Smoking'])
ax = sns.countplot(x='Smoking', data=df1)

for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width() / 2., height + 3, f'{height/total:.1%}',

plt.title('Smoking Habits in the Study Population')
plt.xlabel('Smoking Status')
plt.ylabel('Percentage')
plt.show()
```

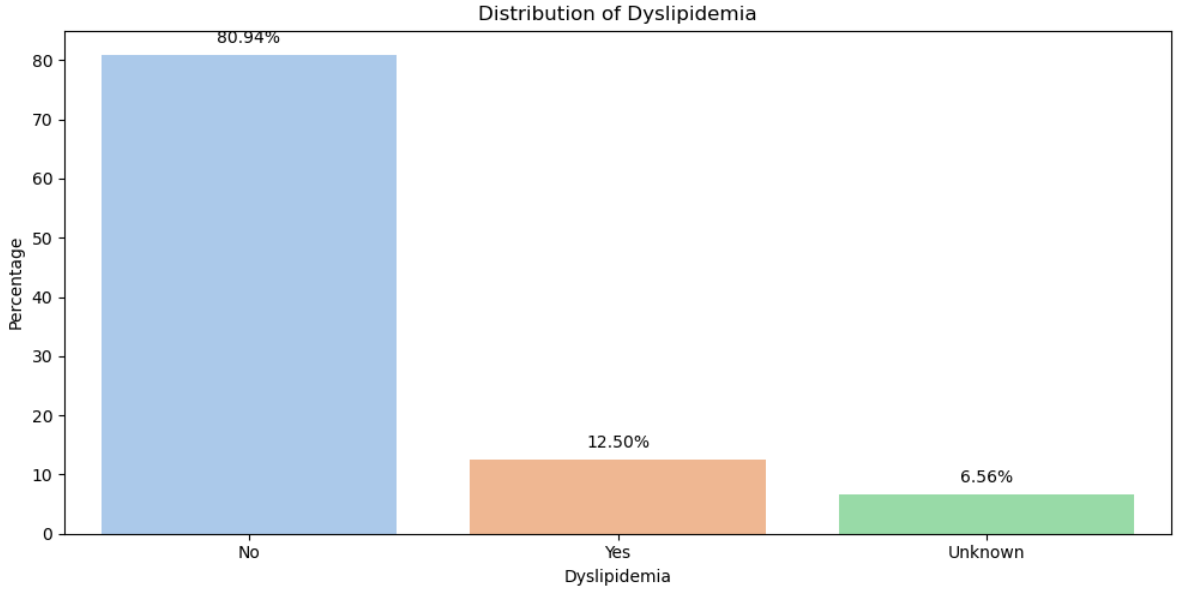


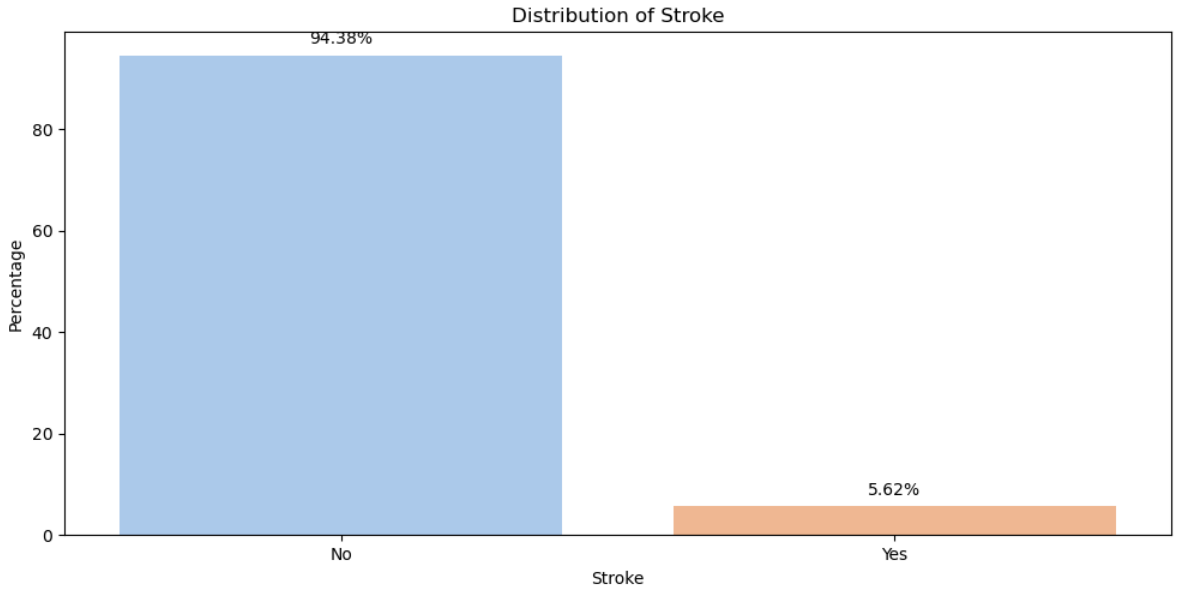
In [ ]:
```
```

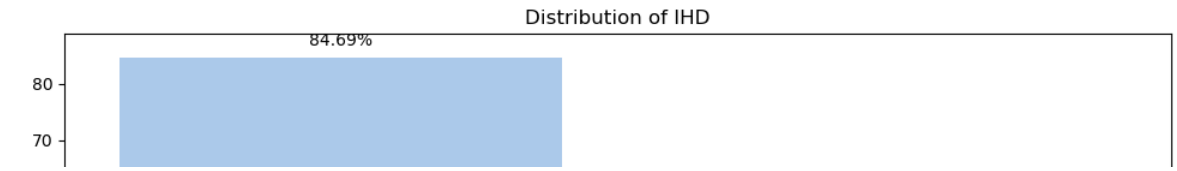
In [31]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Select the relevant columns
selected_columns = ['DM', 'Dyslipidemia', 'Stroke', 'IHD', 'Age_group', 'BMI_G

# Create a new DataFrame with selected columns
selected_df = df1[selected_columns]

# Set up subplots
fig, axes = plt.subplots(nrows=len(selected_columns), ncols=1, figsize=(10, 5

# Loop through each column and create a grouped bar chart
for i, column in enumerate(selected_columns):
    counts = selected_df[column].value_counts(normalize=True) * 100

    # Create a bar chart
    sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])

    # Annotate with percentage values
    for p in axes[i].patches:
        axes[i].annotate(f'{p.get_height():.2f}%', (p.get_x() + p.get_width()
                         ha='center', va='center', xytext=(0, 10), textcoords=

    axes[i].set_title(f'Distribution of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Percentage')

# Adjust layout
plt.tight_layout()
plt.show()
```

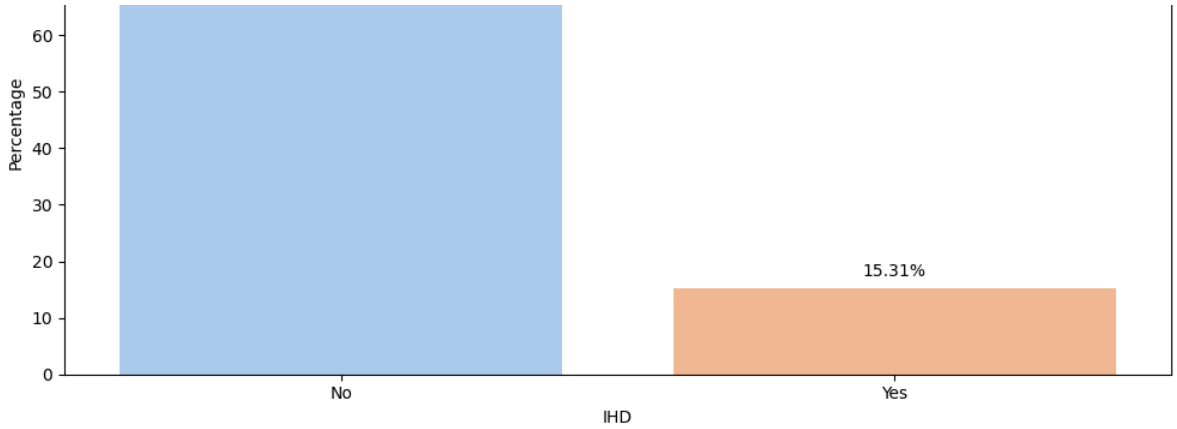
```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\931388240.py:18: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\931388240.py:18: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\931388240.py:18: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\931388240.py:18: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\931388240.py:18: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\931388240.py:18: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.barplot(x=counts.index, y=counts.values, palette='pastel', ax=axes[i])
```

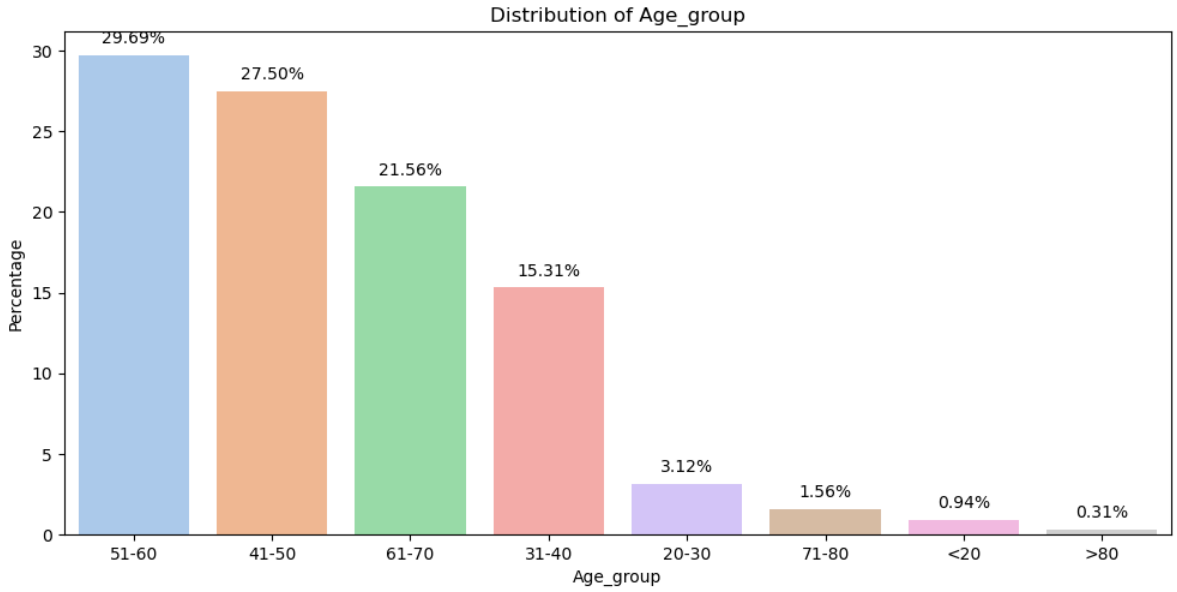

Distribution of DM
53.44%
46.25%
0.31%
Percentage
0
10
20
30
40
50
No
Yes
Unknown
DM

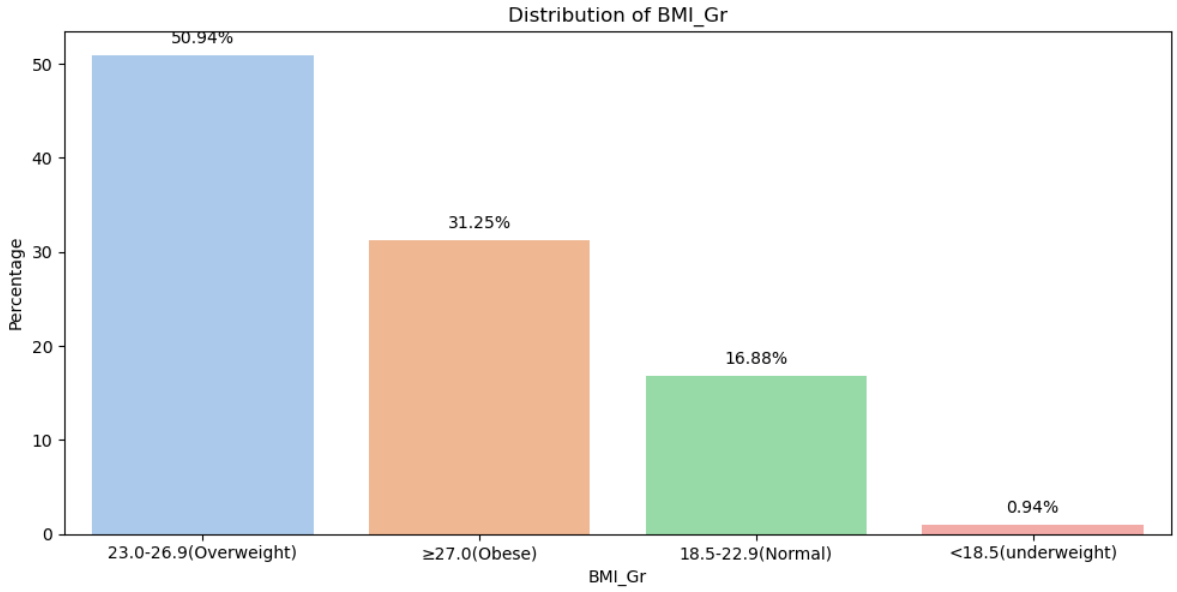

Distribution of Dyslipidemia
80.94%
12.50%
6.56%
Percentage
0
10
20
30
40
50
60
70
80
No
Yes
Unknown
Dyslipidemia


Distribution of Stroke
94.38%
5.62%
Percentage
0
20
40
60
80
No
Yes
Stroke


Distribution of IHD
84.69%
70
80

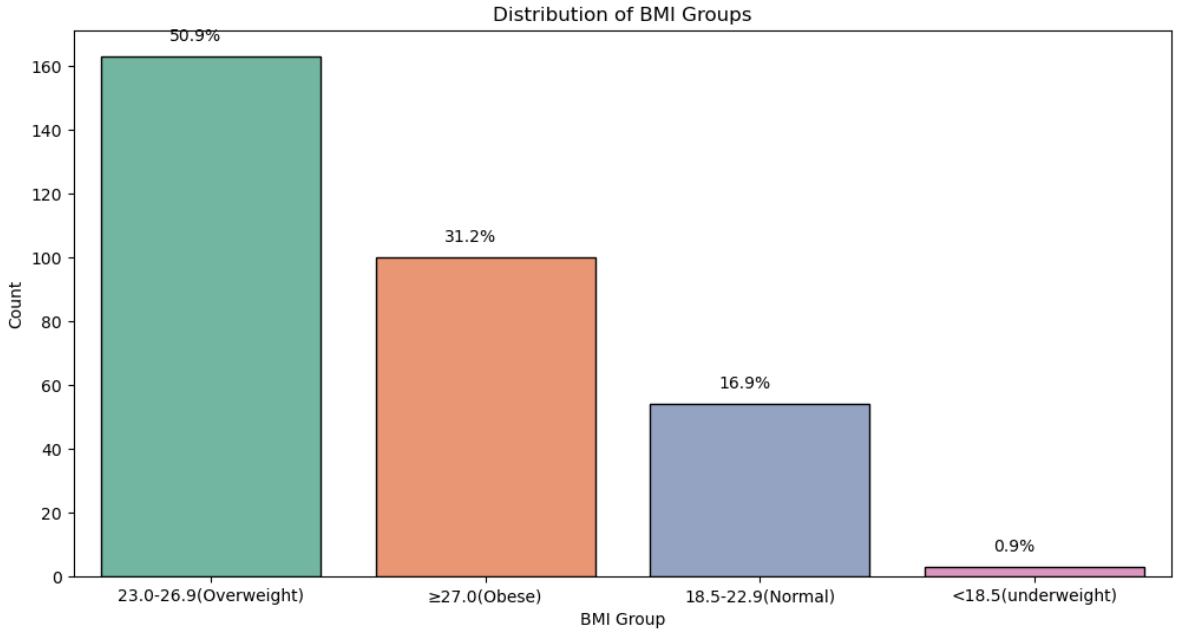

Percentage
60
50
40
30
20
10
0
15.31%
No
Yes
IHD


Distribution of Age_group
Percentage
30
25
20
15
10
5
0
29.69%
27.50%
21.56%
15.31%
3.12%
1.56%
0.94%
0.31%
51-60
41-50
61-70
31-40
20-30
71-80
<20
>80
Age_group


Distribution of BMI_Gr
Percentage
50
40
30
20
10
0
50.94%
31.25%
16.88%
0.94%
23.0-26.9(Overweight)
≥27.0(Obese)
18.5-22.9(Normal)
<18.5(underweight)
BMI_Gr

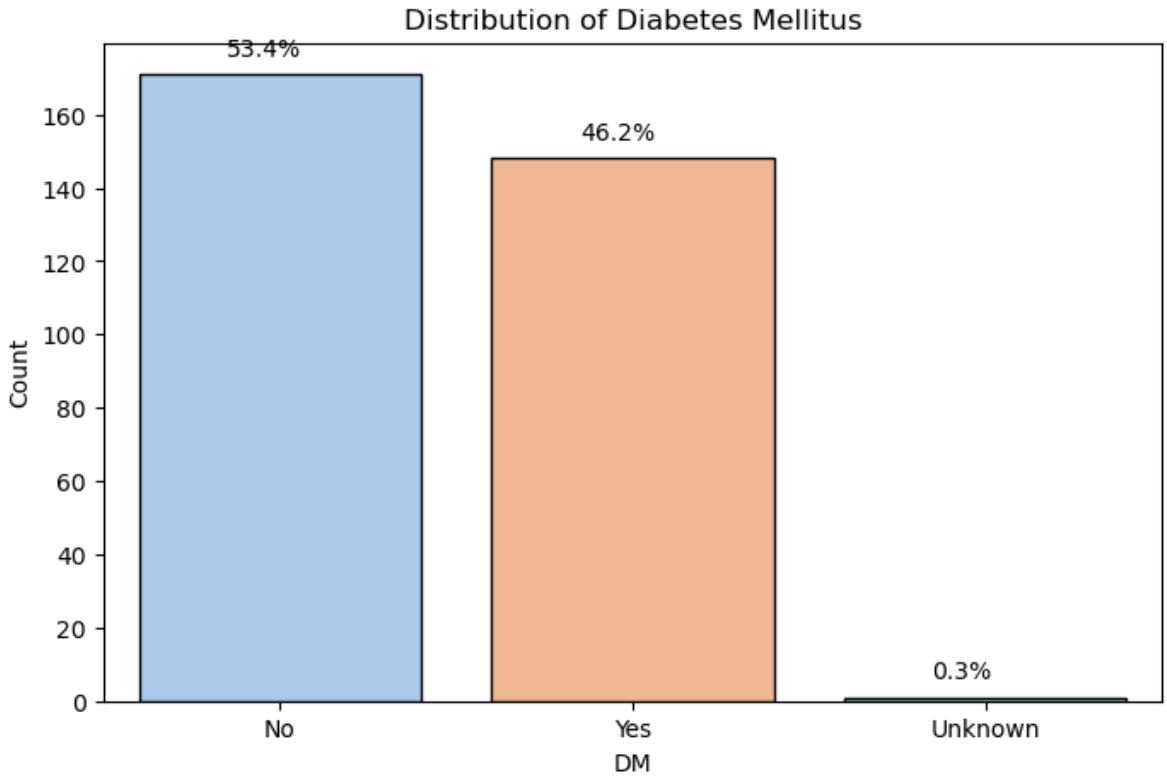
In [32]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'BMI_Gr' is the variable of interest
plt.figure(figsize=(12, 6))
sns.countplot(x='BMI_Gr', data=df1, palette='Set2', edgecolor='black', saturat

# Annotate with percentage values
total = len(df1['BMI_Gr'])
for p in plt.gca().patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total)
    x = p.get_x() + p.get_width() / 2 - 0.15
    y = p.get_height() + 5
    plt.gca().annotate(percentage, (x, y), fontsize=10)

plt.title('Distribution of BMI Groups')
plt.xlabel('BMI Group')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\1124107320.py:6: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.countplot(x='BMI_Gr', data=df1, palette='Set2', edgecolor='black', satu
ration=0.75)
```



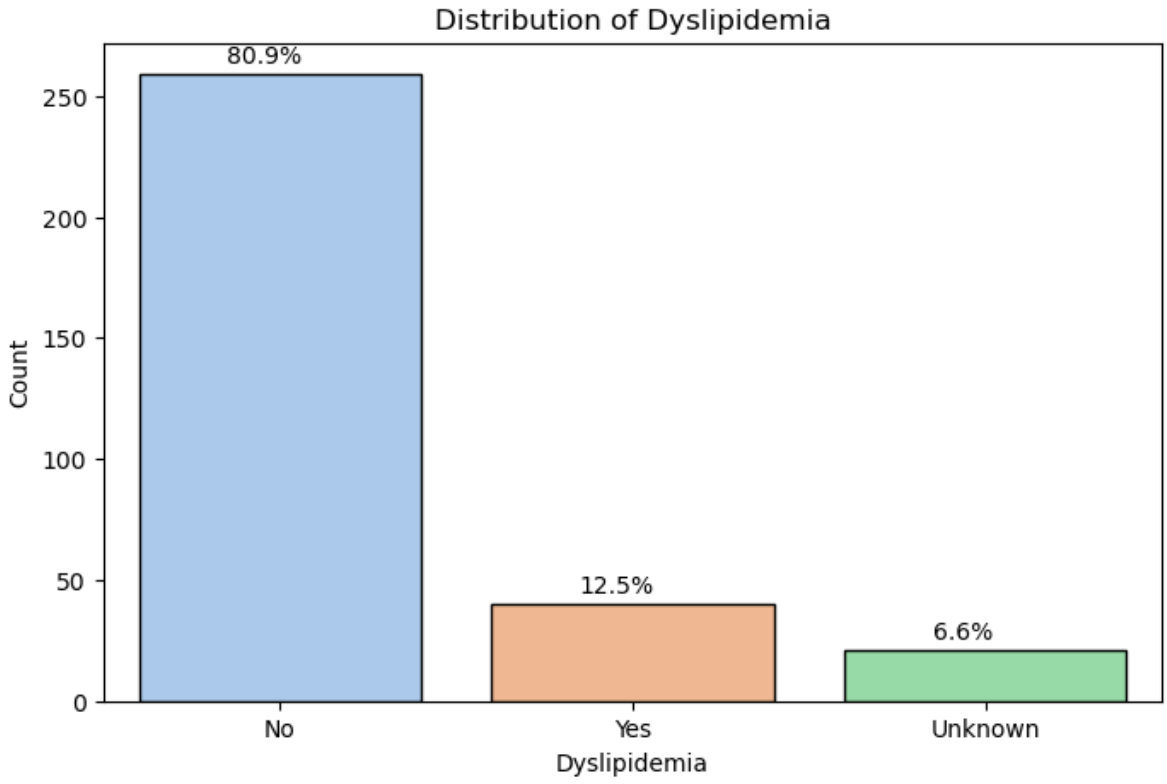
In [33]:
```
plt.figure(figsize=(8, 5))
sns.countplot(x='DM', data=df1, palette='pastel', edgecolor='black', saturatio

# Annotate with percentage values
total_dm = len(df1['DM'])
for p in plt.gca().patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total_dm)
    x = p.get_x() + p.get_width() / 2 - 0.15
    y = p.get_height() + 5
    plt.gca().annotate(percentage, (x, y), fontsize=10)

plt.title('Distribution of Diabetes Mellitus')
plt.xlabel('DM')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\3921584942.py:2: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.countplot(x='DM', data=df1, palette='pastel', edgecolor='black', satura
tion=0.75)
```



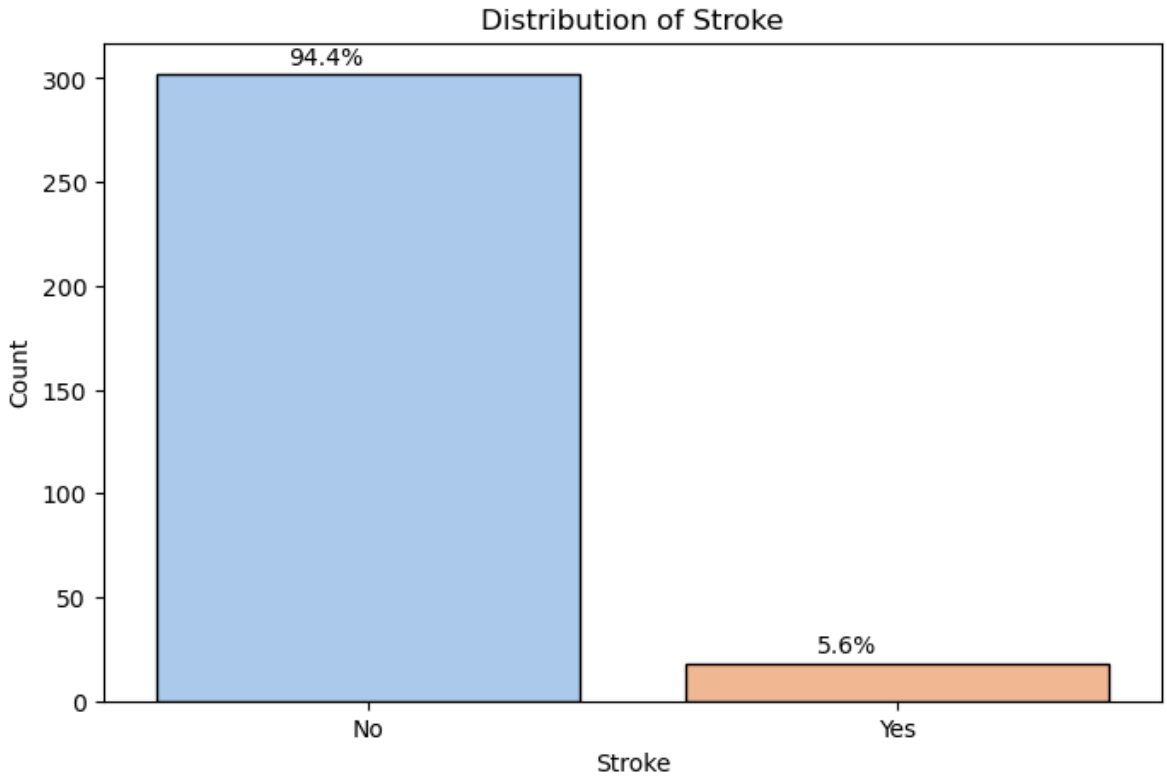
In [34]:
```
plt.figure(figsize=(8, 5))
sns.countplot(x='Dyslipidemia', data=df1, palette='pastel', edgecolor='black',

# Annotate with percentage values
total_dyslipidemia = len(df1['Dyslipidemia'])
for p in plt.gca().patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total_dyslipidemia)
    x = p.get_x() + p.get_width() / 2 - 0.15
    y = p.get_height() + 5
    plt.gca().annotate(percentage, (x, y), fontsize=10)

plt.title('Distribution of Dyslipidemia')
plt.xlabel('Dyslipidemia')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\1435658765.py:2: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.countplot(x='Dyslipidemia', data=df1, palette='pastel', edgecolor='blac
k', saturation=0.75)
```



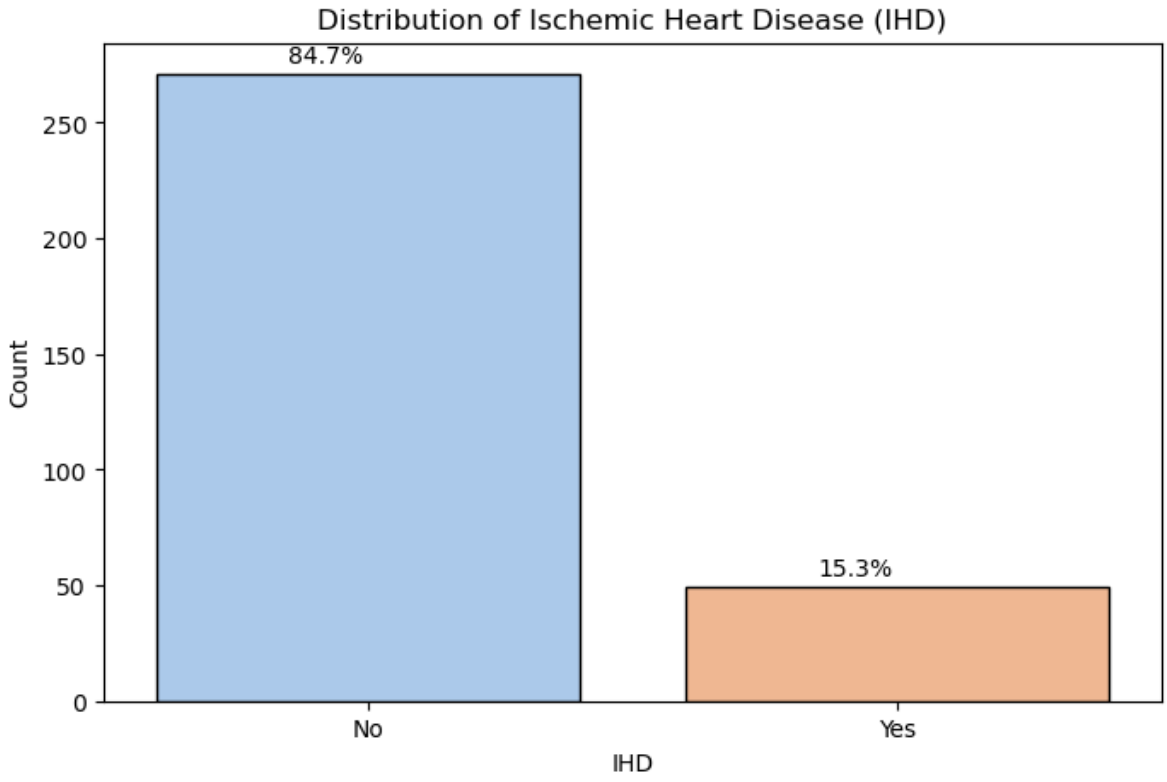
In [35]:
```
plt.figure(figsize=(8, 5))
sns.countplot(x='Stroke', data=df1, palette='pastel', edgecolor='black', satur

# Annotate with percentage values
total_stroke = len(df1['Stroke'])
for p in plt.gca().patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total_stroke)
    x = p.get_x() + p.get_width() / 2 - 0.15
    y = p.get_height() + 5
    plt.gca().annotate(percentage, (x, y), fontsize=10)

plt.title('Distribution of Stroke')
plt.xlabel('Stroke')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\1685772313.py:2: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.countplot(x='Stroke', data=df1, palette='pastel', edgecolor='black', sa
turation=0.75)
```



In [36]:
```
plt.figure(figsize=(8, 5))
sns.countplot(x='IHD', data=df1, palette='pastel', edgecolor='black', saturati

# Annotate with percentage values
total_ihd = len(df1['IHD'])
for p in plt.gca().patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total_ihd)
    x = p.get_x() + p.get_width() / 2 - 0.15
    y = p.get_height() + 5
    plt.gca().annotate(percentage, (x, y), fontsize=10)

plt.title('Distribution of Ischemic Heart Disease (IHD)')
plt.xlabel('IHD')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\391336208.py:2: FutureWarni
ng:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.countplot(x='IHD', data=df1, palette='pastel', edgecolor='black', satur
ation=0.75)
```



In [ ]:

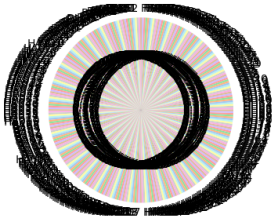
In [ ]:

In [ ]:

In [ ]:

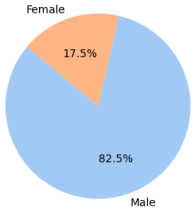
In [28]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Get a list of categorical columns
categorical_columns = df1.select_dtypes(include=['object']).columns

# Set up subplots
num_plots = len(categorical_columns)
num_cols = 3  # You can adjust the number of columns in the subplot grid
num_rows = (num_plots + num_cols - 1) // num_cols

fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 5 * num_rows))

# Flatten axes in case there's only one row
axes = axes.flatten()

# Loop through each categorical column and create a pie chart
for i, col in enumerate(categorical_columns):
    counts = df1[col].value_counts()

    axes[i].pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=140
    axes[i].set_title(f'{col} Distribution')

# Adjust layout
plt.tight_layout()
plt.show()
```

Serial_No Distribution

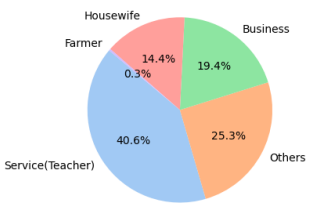


Name Distribution

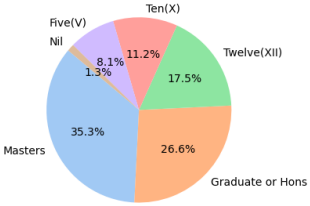
Sex Distribution



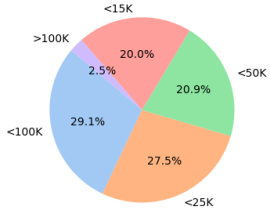
Occupation Distribution



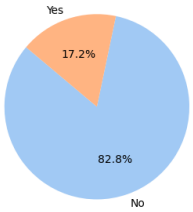
Education Distribution



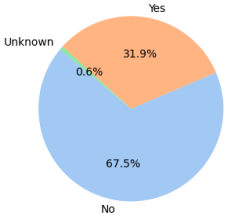
Economy Distribution



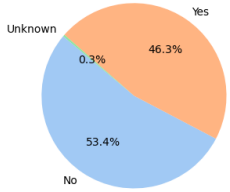
Smoking Distribution



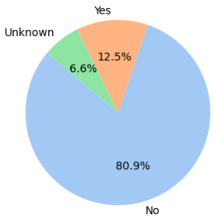
HTN Distribution



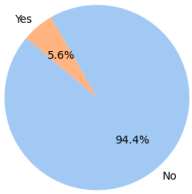
DM Distribution



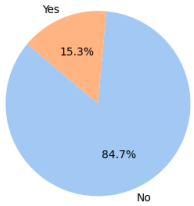
Dyslipidemia Distribution



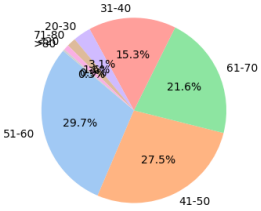
Stroke Distribution



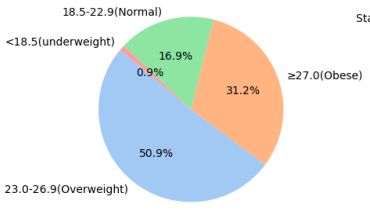
IHD Distribution



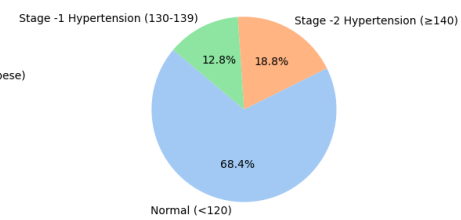
Age_group Distribution



BMI_Gr Distribution

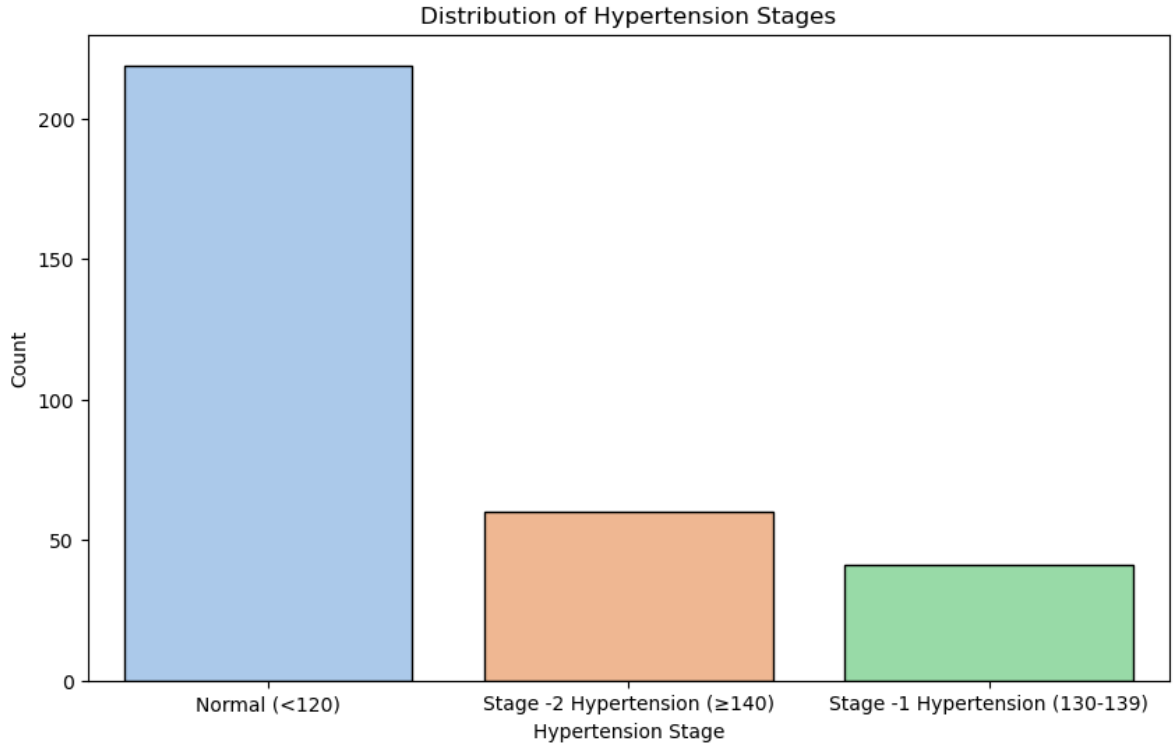


Hypertenson_stage Distribution

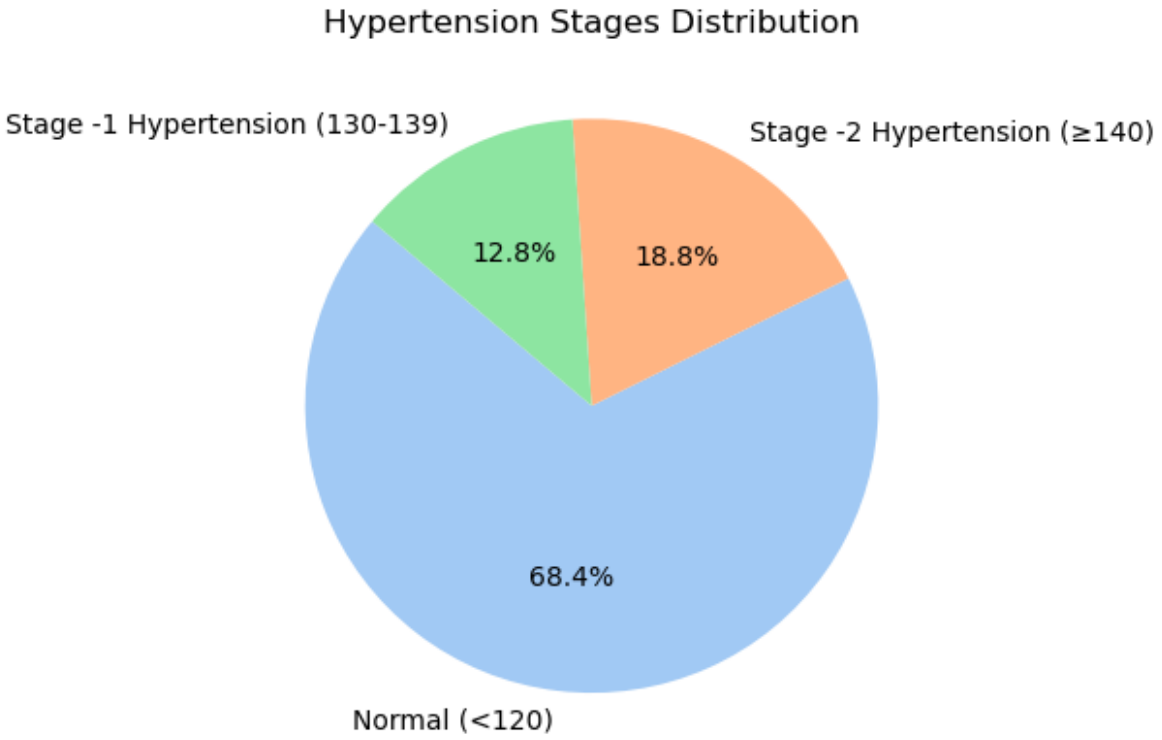


In [29]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'Hypertenson_stage' is the variable of interest
plt.figure(figsize=(10, 6))
sns.countplot(x='Hypertenson_stage', data=df1, palette='pastel', edgecolor='bl
plt.title('Distribution of Hypertension Stages')
plt.xlabel('Hypertension Stage')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ipdi2\AppData\Local\Temp\ipykernel_11784\3348328464.py:6: FutureWarn
ing:

Passing `palette` without assigning `hue` is deprecated and will be removed i
n v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sa
me effect.

  sns.countplot(x='Hypertenson_stage', data=df1, palette='pastel', edgecolor
='black', saturation=0.75)
```



In [30]:
```
hypertension_counts = df1['Hypertenson_stage'].value_counts()
plt.pie(hypertension_counts, labels=hypertension_counts.index, autopct='%1.1f%
plt.title('Hypertension Stages Distribution')
plt.show()
```



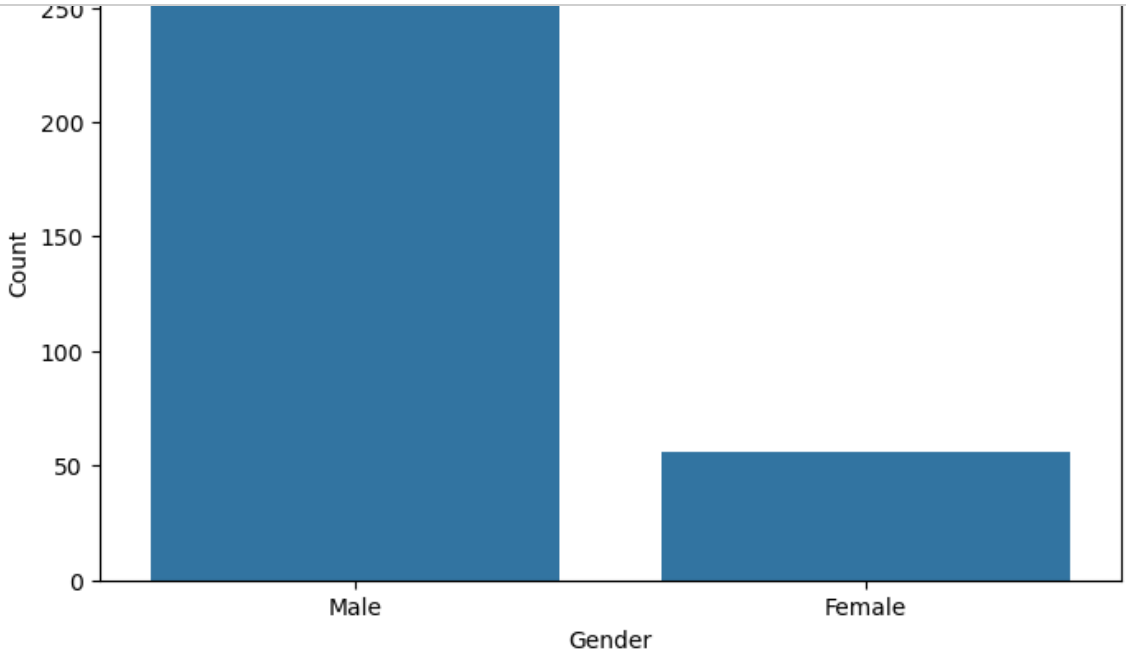
In [ ]:
```

```

In [71]:
```
df1.describe()
```

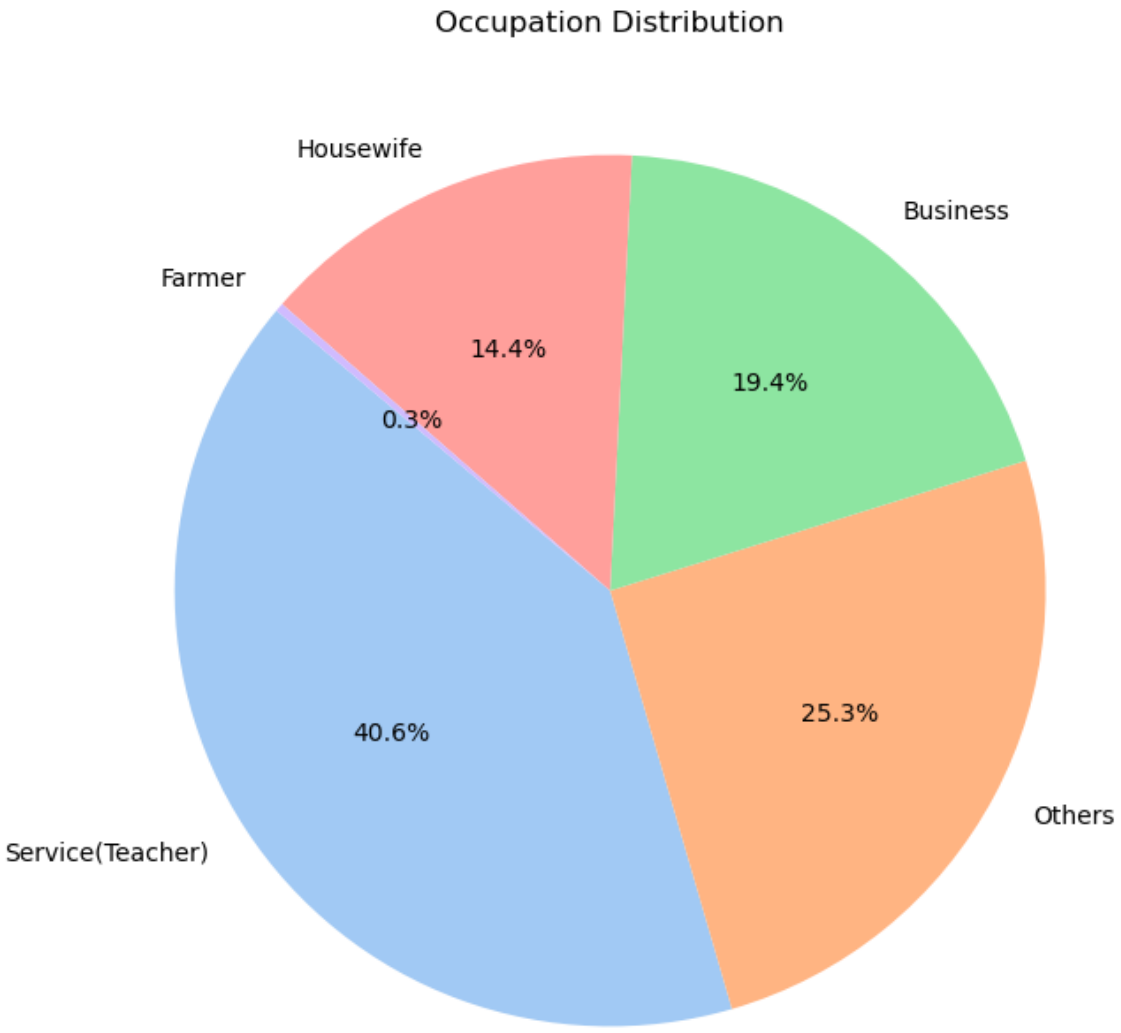
Out[71]:

	Age	Hight	weight	BMI	Systolic_Upper	Diastolic_Lower	R
count	320.000000	320.000000	320.000000	320.000000	320.00000	320.000000	320.000
mean	51.118750	164.803125	69.800625	25.809787	120.71875	78.750000	6.895
std	11.634384	8.668141	10.035067	4.394856	15.05837	8.691869	2.917
min	17.000000	103.000000	7.200000	2.324380	90.00000	60.000000	2.000
25%	42.750000	160.000000	64.000000	23.447564	110.00000	70.000000	5.075
50%	52.000000	166.000000	69.000000	25.417751	120.00000	80.000000	6.000
75%	60.000000	170.000000	76.000000	27.665487	130.00000	80.000000	7.600
max	83.000000	190.000000	95.000000	75.407673	170.00000	110.000000	19.400

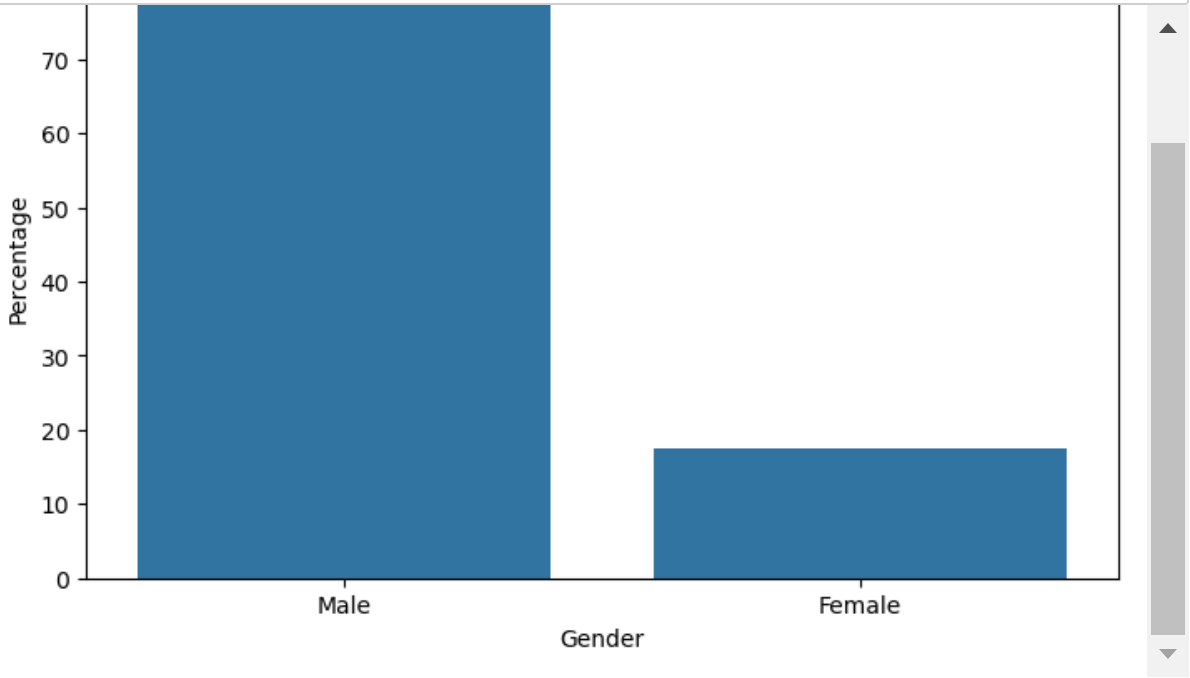
In [25]:
```
# Countplot of Gender
plt.figure(figsize=(8, 5))
sns.countplot(x='Sex', data=df1)
plt.title('Count of Individuals by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



In [27]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data (replace this with your data)
occupation_counts = df1['Occupation'].value_counts()

# Plotting
plt.figure(figsize=(10, 8))
plt.pie(occupation_counts, labels=occupation_counts.index, autopct='%1.1f%%',
plt.title('Occupation Distribution')
plt.show()
```



In [26]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate percentages
total = len(df1)  # Total number of entries
gender_counts = df1['Sex'].value_counts()
percentage = (gender_counts / total) * 100

# Plotting
plt.figure(figsize=(8, 5))
sns.barplot(x=percentage.index, y=percentage.values)
plt.title('Percentage of Individuals by Gender')
plt.xlabel('Gender')
plt.ylabel('Percentage')
plt.show()
```



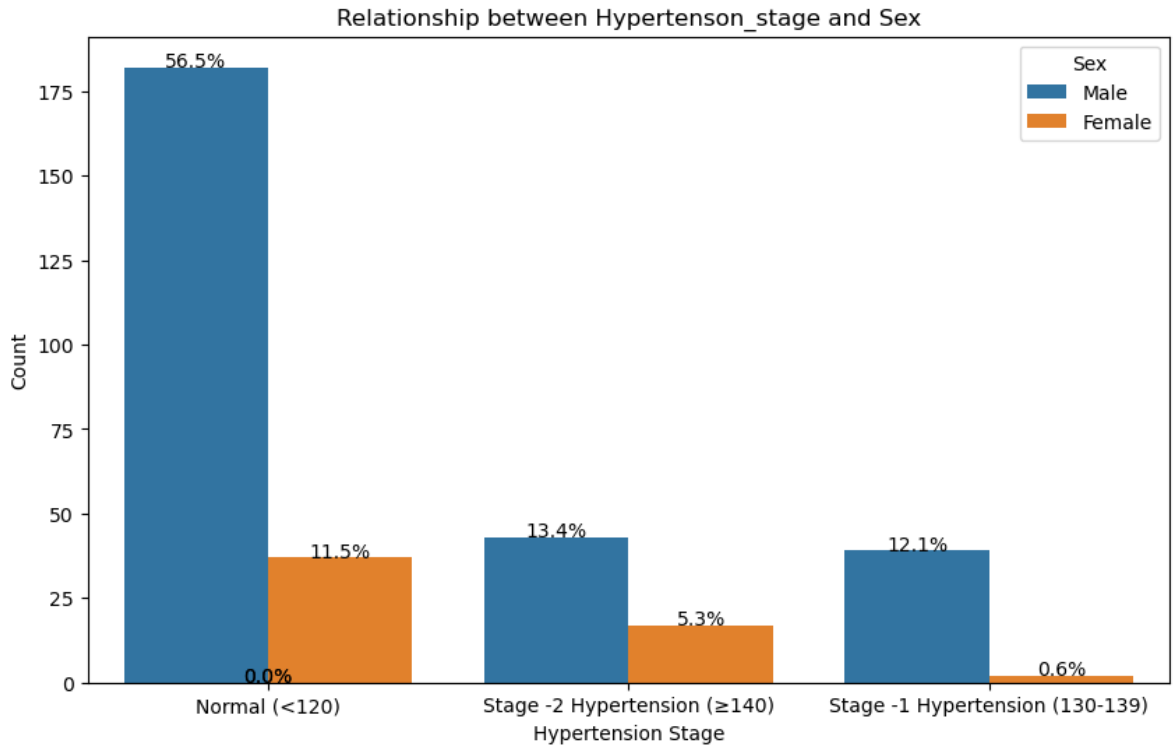
In [73]:
```
# Example: Count the occurrences of each category in the 'Sex' column
sex_counts = df1['Sex'].value_counts()
print(sex_counts)
```

```
Male      264
Female     56
Name: Sex, dtype: int64
```

In [75]:
```
import pandas as pd
from scipy.stats import chi2_contingency

# Create a contingency table
contingency_table = pd.crosstab(df1['Hypertenson_stage'], df1['Sex'])

# Perform the chi-squared test
chi2, p, _, _ = chi2_contingency(contingency_table)

# Display the results
print(f"Chi-squared value: {chi2}")
print(f"P-value: {p}")

# Check for significance at a certain level (e.g., 0.05)
alpha = 0.05
if p < alpha:
    print("There is a significant relationship between Hypertenson_stage and S
else:
    print("There is no significant relationship between Hypertenson_stage and
```

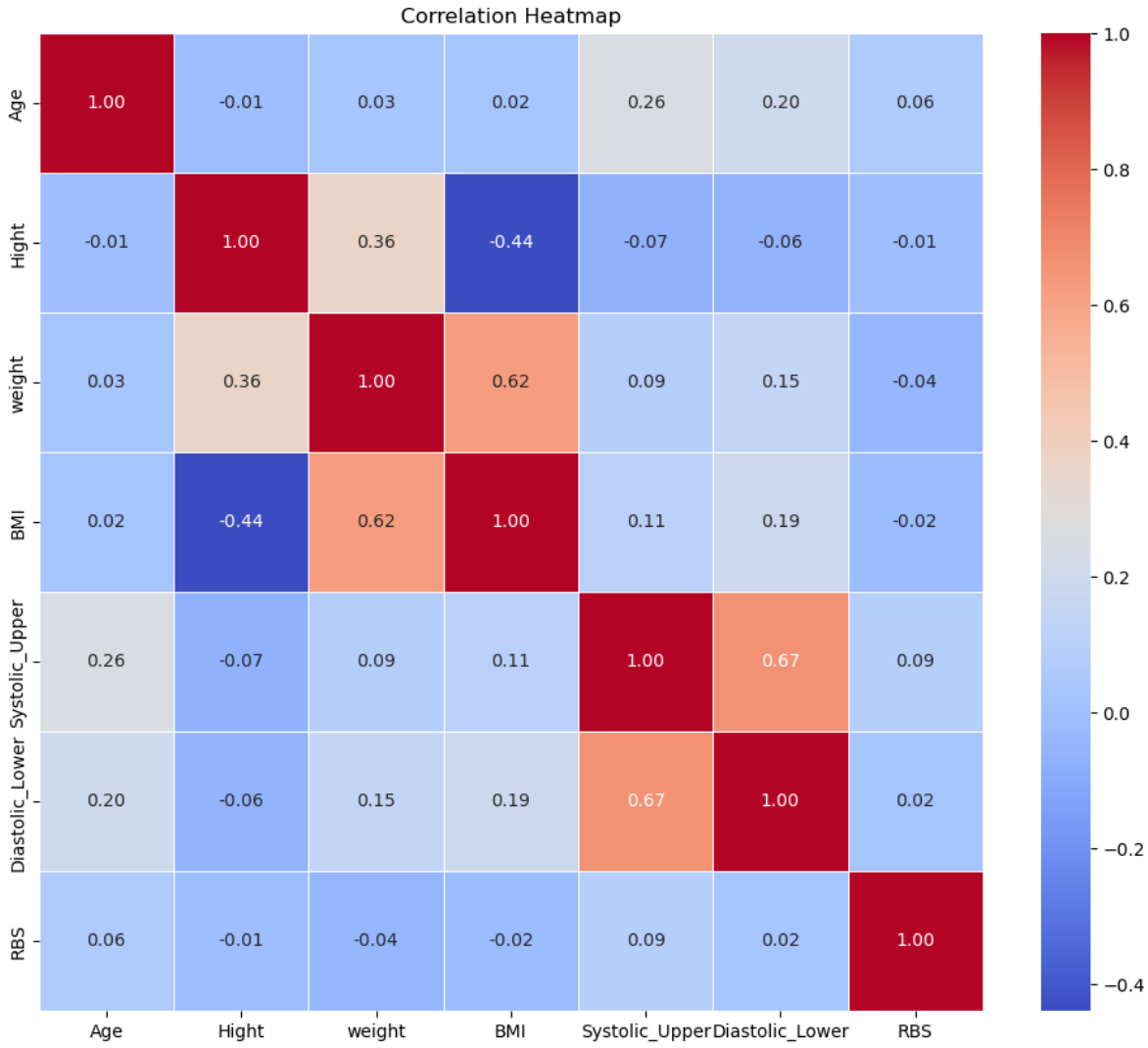
```
Chi-squared value: 9.457102647881129
P-value: 0.008839267003376509
There is a significant relationship between Hypertenson_stage and Sex.
```

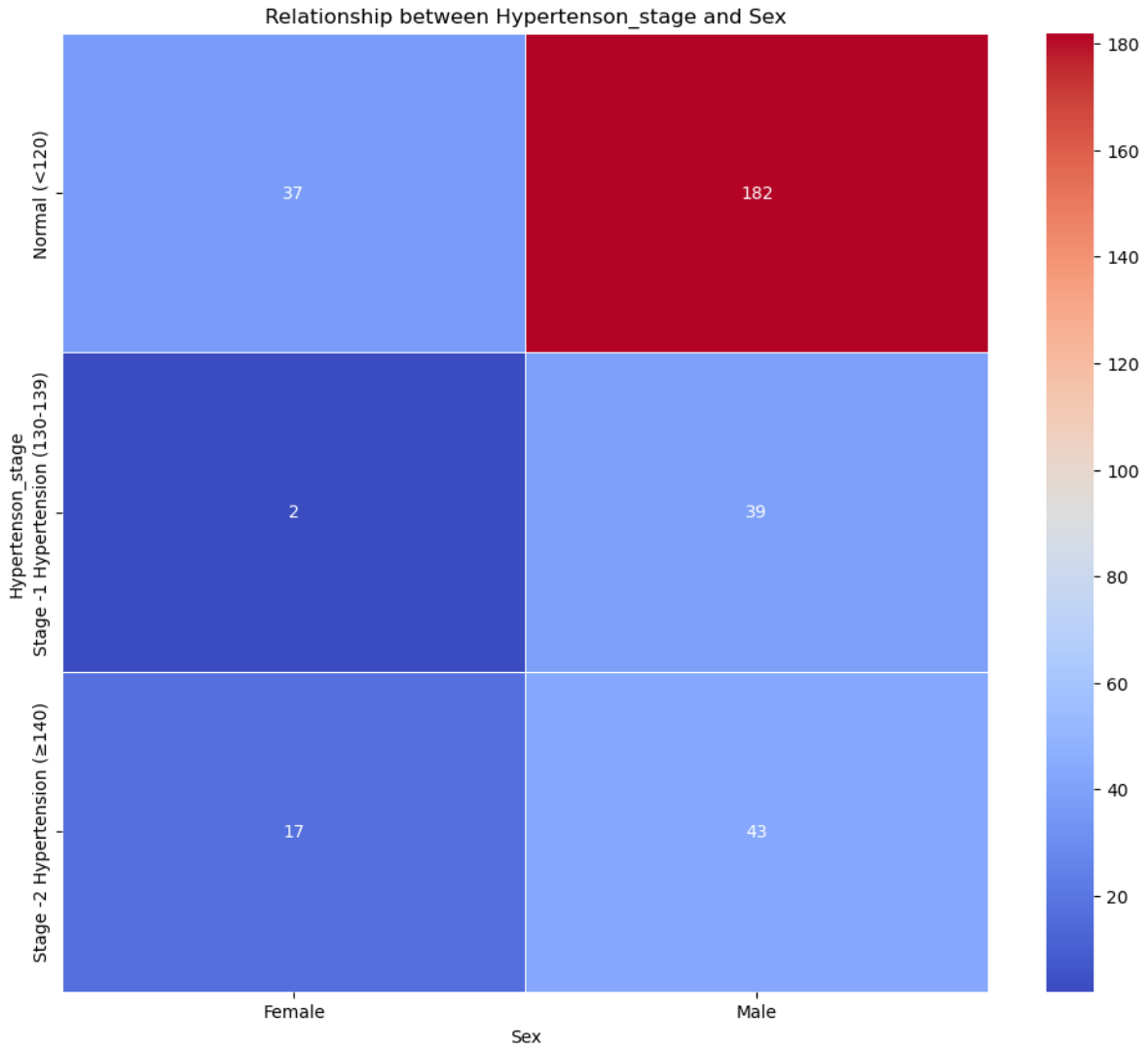
In [76]:
```
# Visualize the relationship using a stacked bar chart with improved labels
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='Hypertenson_stage', hue='Sex', data=df1)

# Customize the plot for better visibility of category names
ax.set_title('Relationship between Hypertenson_stage and Sex')
ax.set_xlabel('Hypertension Stage')
ax.set_ylabel('Count')
ax.legend(title='Sex')

# Display the percentage labels on top of each bar
total = float(len(df))
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width() / 2., height + 0.1,
            f'{height/total:.1%}', ha="center")

plt.show()
```



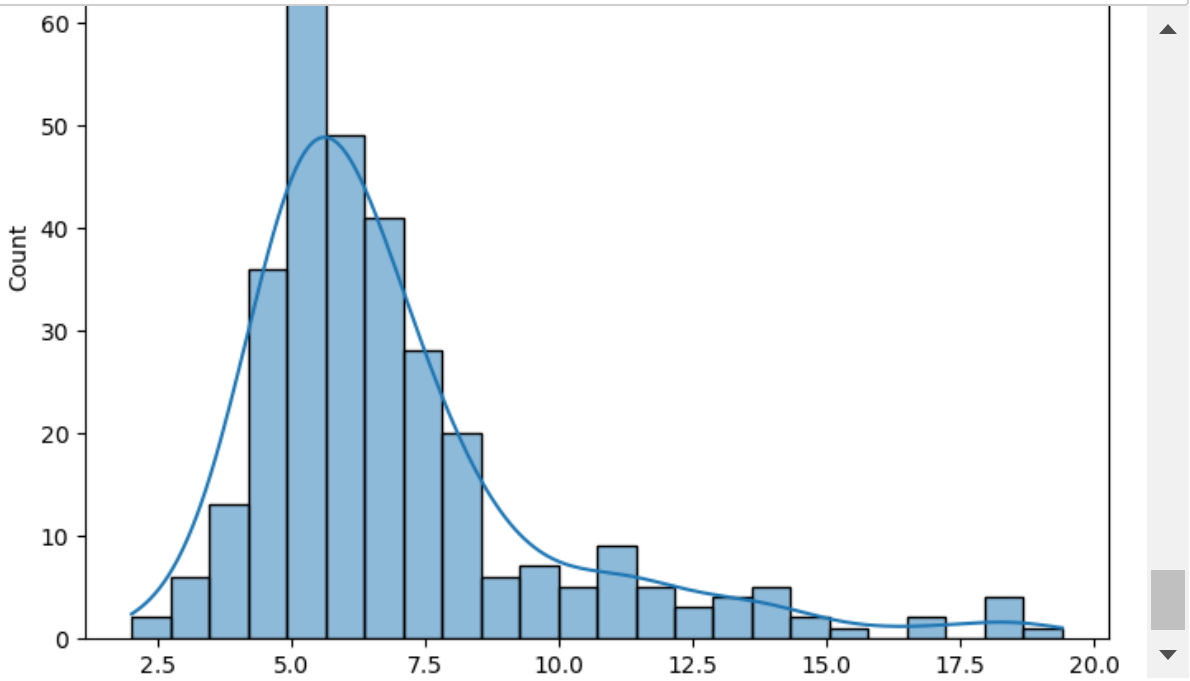
In [ ]:
```
# Calculate percentages for each categorical variable
categorical_columns = df1.select_dtypes(include=['object']).columns
categorical_percentages = df1[categorical_columns].apply(lambda x: x.value_cou

# Display the table
print(categorical_percentages)
```

In [77]:
```
# Correlation matrix for numerical variables
correlation_matrix = df1.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewi
plt.title('Correlation Heatmap')
plt.show()

# Relationship between categorical variables
plt.figure(figsize=(12, 10))
sns.heatmap(pd.crosstab(df1['Hypertenson_stage'], df1['Sex']), annot=True, cma
plt.title('Relationship between Hypertenson_stage and Sex')
plt.show()
```

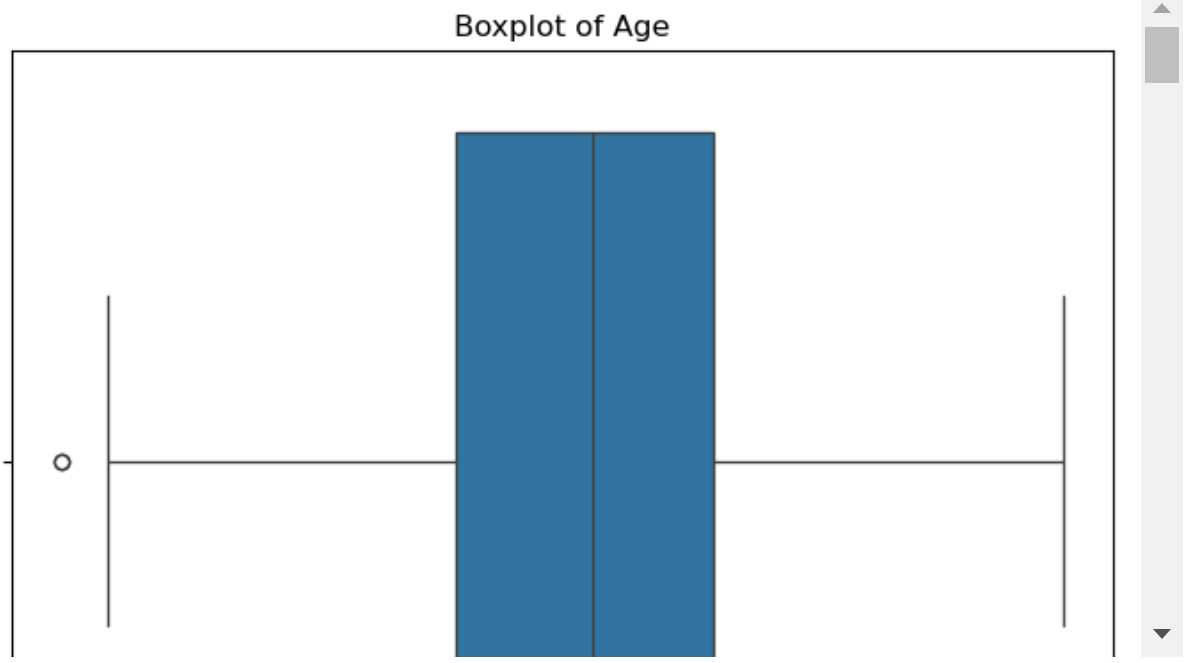



Relationship between Hypertenson_stage and Sex
Hypertenson_stage
Normal (<120)
Stage -1 Hypertension (130-139)
Stage -2 Hypertension (≥140)
37
182
2
39
17
43
Female
Male
Sex
180
160
140
120
100
80
60
40
20

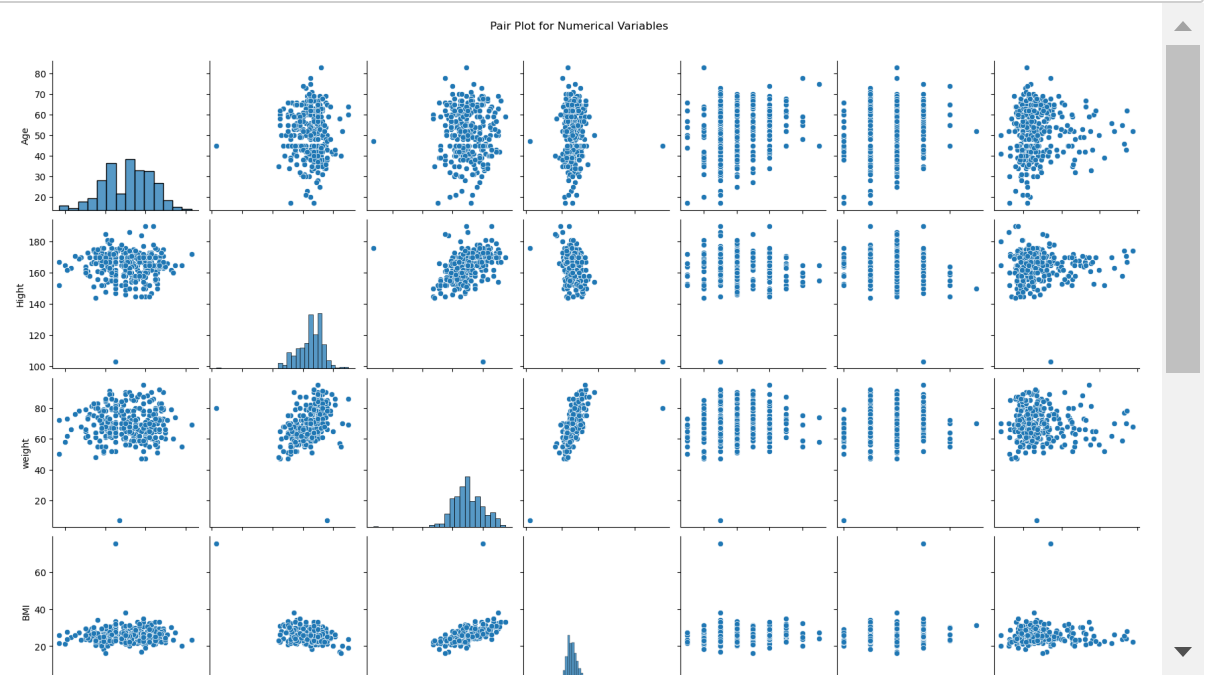
In [78]:
```
# Distribution of numerical variables
numerical_columns = df1.select_dtypes(include=['float64', 'int64']).columns
for column in numerical_columns:
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.show()
```



In [21]:
```
# Detect and visualize outliers using boxplots
for column in numerical_columns:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=column, data=df)
    plt.title(f'Boxplot of {column}')
    plt.show()
```



In [79]:
```
# Pair plot for numerical variables
sns.pairplot(df1[numerical_columns])
plt.suptitle('Pair Plot for Numerical Variables', y=1.02)
plt.show()
```



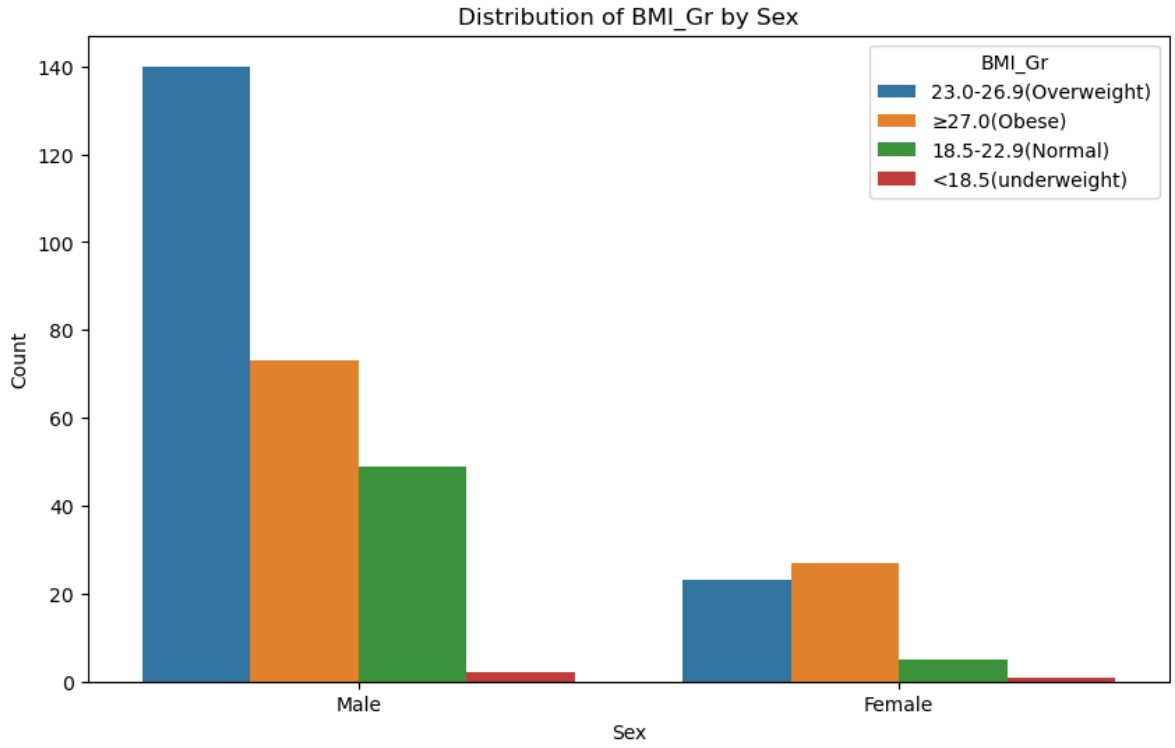
In [80]:
```
# Display the column names in the DataFrame
print(df1.columns)
```

```
Index(['Serial_No', 'Name', 'Age', 'Sex', 'Occupation', 'Education', 'Econom
y',
       'Hight', 'weight', 'BMI', 'Systolic_Upper', 'Diastolic_Lower', 'RBS',
       'Smoking', 'HTN', 'DM', 'Dyslipidemia', 'Stroke', 'IHD', 'Age_group',
       'BMI_Gr', 'Ag_g', 'Hypertenson_stage'],
      dtype='object')
```

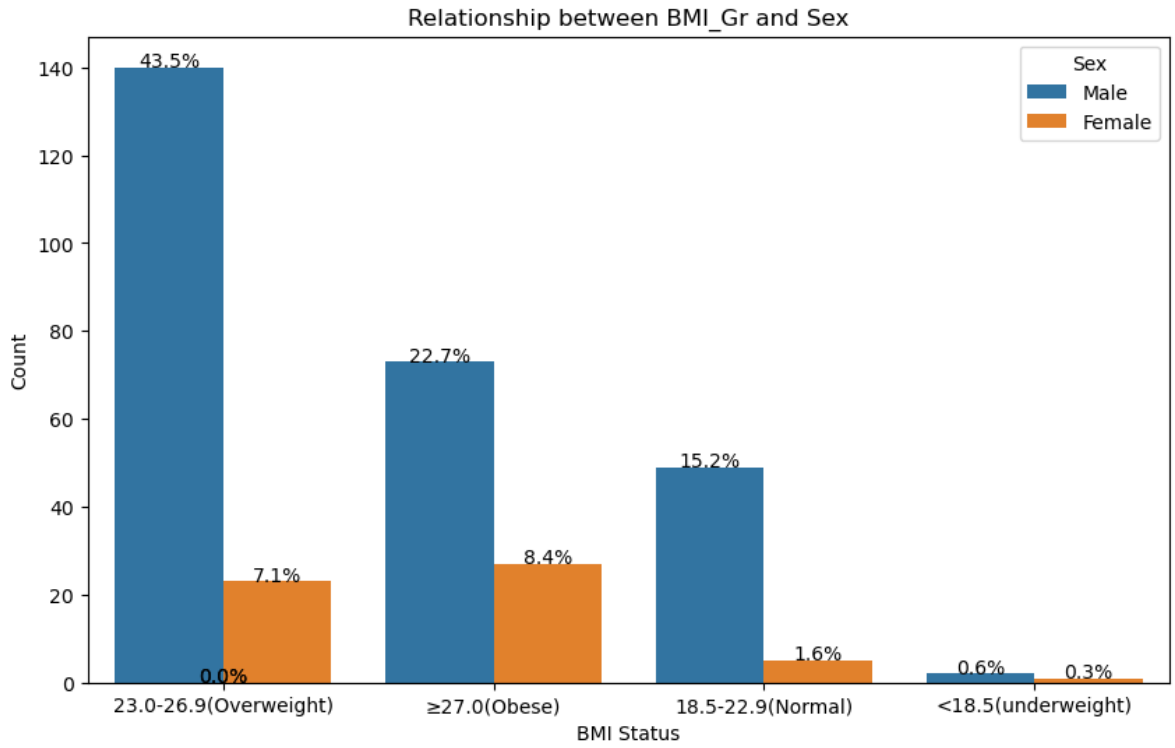
In [81]:
```
import pandas as pd
from scipy.stats import chi2_contingency

# Create a contingency table
contingency_table = pd.crosstab(df1['Sex'], df1['BMI_Gr'])

# Perform the chi-squared test
chi2, p, _, _ = chi2_contingency(contingency_table)

# Display the results
print(f"Chi-squared value: {chi2}")
print(f"P-value: {p}")

# Check for significance at a certain level (e.g., 0.05)
alpha = 0.05
if p < alpha:
    print("There is a significant relationship between Sex and BMI_Gr.")
else:
    print("There is no significant relationship between Sex and BMI_Gr.")
```

```
Chi-squared value: 10.609143337159699
P-value: 0.014038502234043449
There is a significant relationship between Sex and BMI_Gr.
```

In [82]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a contingency table
contingency_table = pd.crosstab(df1['Sex'], df1['BMI_Gr'])

# Plot a stacked bar chart
plt.figure(figsize=(10, 6))
sns.countplot(x='Sex', hue='BMI_Gr', data=df1)
plt.title('Distribution of BMI_Gr by Sex')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.show()
```



In [83]:
```
# Visualize the relationship using a stacked bar chart with improved labels
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='BMI_Gr', hue='Sex', data=df1)

# Customize the plot for better visibility of category names
ax.set_title('Relationship between BMI_Gr and Sex')
ax.set_xlabel('BMI Status')
ax.set_ylabel('Count')
ax.legend(title='Sex')

# Display the percentage labels on top of each bar
total = float(len(df))
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width() / 2., height + 0.1,
            f'{height/total:.1%}', ha="center")

plt.show()
```



In [87]:
```
# Print the column names of df1
print(df1.columns)
```

```
Index(['Serial_No', 'Name', 'Age', 'Sex', 'Occupation', 'Education', 'Econom
y',
       'Hight', 'weight', 'BMI', 'Systolic_Upper', 'Diastolic_Lower', 'RBS',
       'Smoking', 'HTN', 'DM', 'Dyslipidemia', 'Stroke', 'IHD', 'Age_group',
       'BMI_Gr', 'Ag_g', 'Hypertenson_stage'],
      dtype='object')
```

In [44]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Assuming your DataFrame is named 'df'
# Replace 'df' with the actual DataFrame name

# Select the features and target variable
features = ['Hight', 'weight']  # Replace with your selected features
target_bmi = 'BMI'

# Selecting the relevant columns
df2 = df[features + [target_bmi]].dropna()
```

In [45]:
```
df2
```

Out[45]:

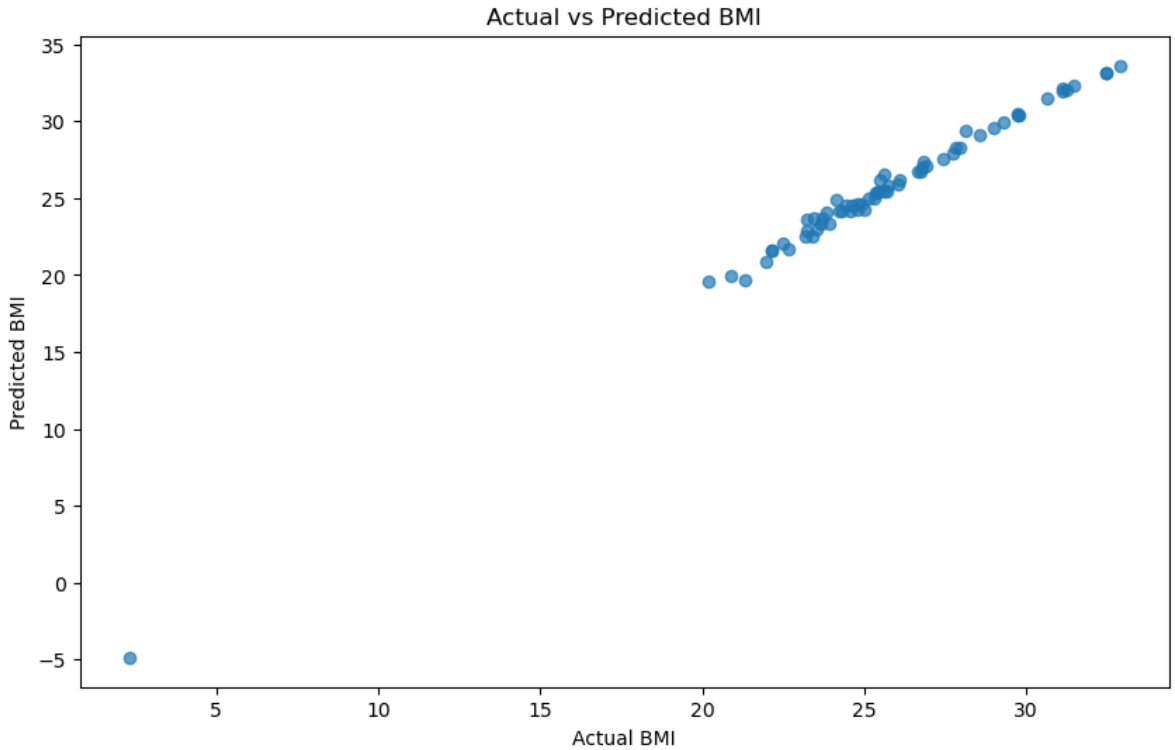
	Hight	weight	BMI
0	171.0	69.0	23.597004
1	145.0	65.0	30.915577
2	175.0	75.0	24.489796
3	170.0	71.0	24.567474
4	172.0	68.0	22.985398
...	...	...	...
317	171.0	78.0	26.674874
318	160.0	66.0	25.781250
319	171.0	76.0	25.990903
320	170.0	73.0	25.259516
321	165.0	70.0	25.711662

322 rows × 3 columns

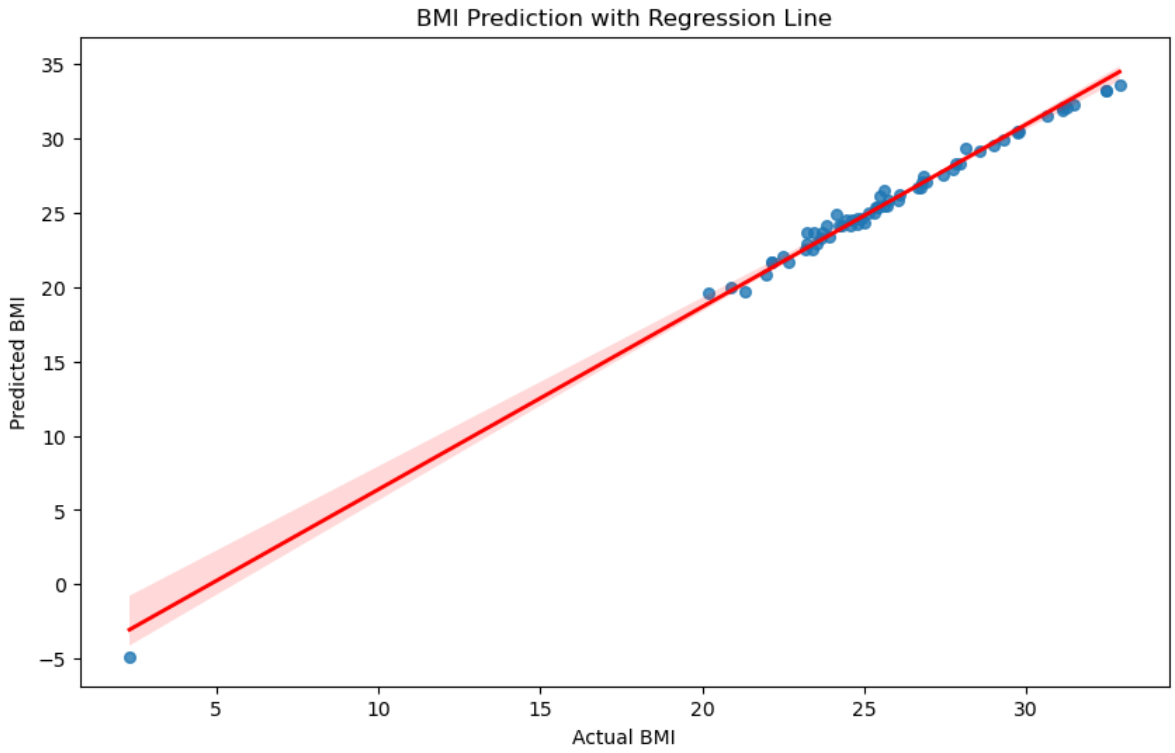
In [101]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    df2[features], df2[target_bmi], test_size=0.2, random_state=42
)

# Initialize a linear regression model
model = LinearRegression()

# Fit the model
model.fit(X_train, y_train)

# Predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print model evaluation metrics
print(f'Mean Squared Error: {mse}, R-squared: {r2}')
```

```
Mean Squared Error: 1.138525369421954, R-squared: 0.9351202487234338
```

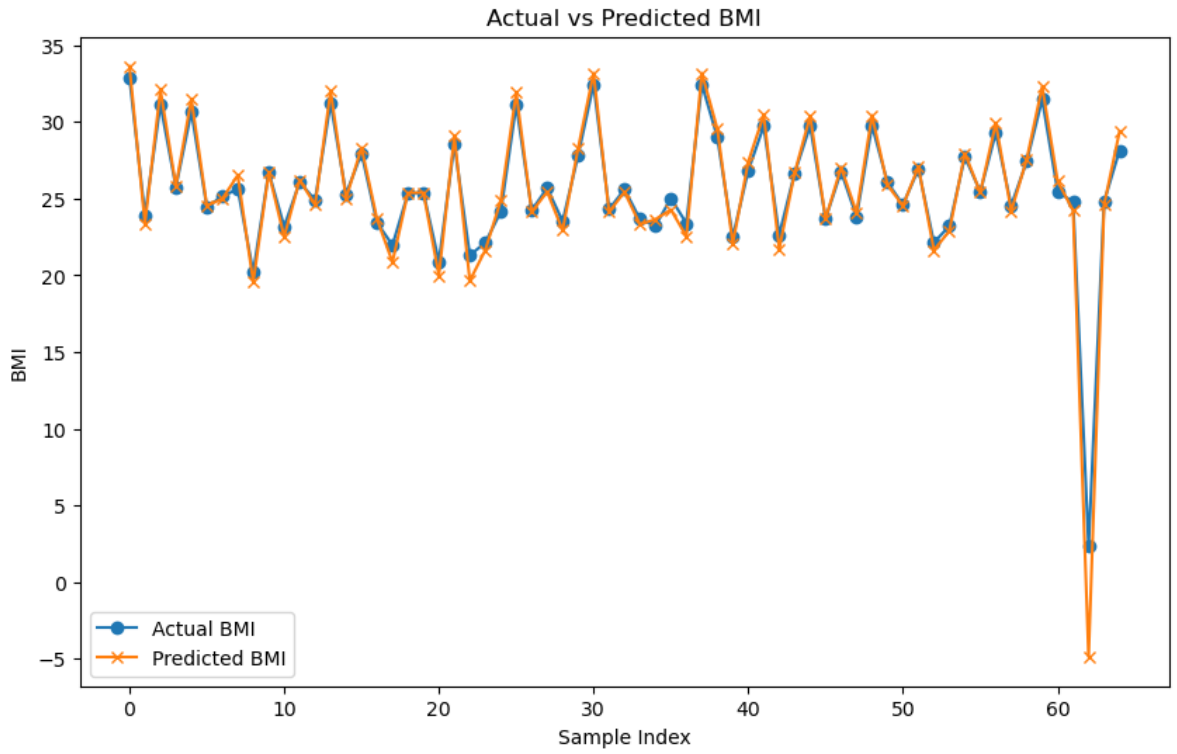
In [99]:
```
# Visualize actual and predicted values in a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.7)
plt.xlabel('Actual BMI')
plt.ylabel('Predicted BMI')
plt.title('Actual vs Predicted BMI')
plt.show()
```



In [100]:
```
# Visualize predictions vs actual values with a regression line
plt.figure(figsize=(10, 6))
sns.regplot(x=y_test, y=y_pred, scatter_kws={'s': 30}, line_kws={'color': 'red
plt.xlabel('Actual BMI')
plt.ylabel('Predicted BMI')
plt.title('BMI Prediction with Regression Line')
plt.show()
```



In [95]:
```
import matplotlib.pyplot as plt

# Visualize actual vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual BMI', marker='o')
plt.plot(y_pred, label='Predicted BMI', marker='x')
plt.xlabel('Sample Index')
plt.ylabel('BMI')
plt.title('Actual vs Predicted BMI')
plt.legend()
plt.show()
```



In [40]:
```
pip install statsmodels
```

```
Note: you may need to restart the kernel to use updated packages.Requirement
already satisfied: statsmodels in c:\users\ipdi2\anaconda3\lib\site-packages
(0.13.2)
Requirement already satisfied: numpy>=1.17 in c:\users\ipdi2\anaconda3\lib\si
te-packages (from statsmodels) (1.21.5)
Requirement already satisfied: scipy>=1.3 in c:\users\ipdi2\anaconda3\lib\sit
e-packages (from statsmodels) (1.9.1)
Requirement already satisfied: pandas>=0.25 in c:\users\ipdi2\anaconda3\lib\s
ite-packages (from statsmodels) (1.4.4)
Requirement already satisfied: patsy>=0.5.2 in c:\users\ipdi2\anaconda3\lib\s
ite-packages (from statsmodels) (0.5.2)
Requirement already satisfied: packaging>=21.3 in c:\users\ipdi2\anaconda3\li
b\site-packages (from statsmodels) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\ipdi2\ana
conda3\lib\site-packages (from packaging>=21.3->statsmodels) (3.0.9)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\ipdi2\anaco
nda3\lib\site-packages (from pandas>=0.25->statsmodels) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\ipdi2\anaconda3\lib\s
ite-packages (from pandas>=0.25->statsmodels) (2022.1)
Requirement already satisfied: six in c:\users\ipdi2\anaconda3\lib\site-packa
ges (from patsy>=0.5.2->statsmodels) (1.16.0)
```

In [48]:
```
print(df2.columns)
```

```
Index(['Hight', 'weight', 'BMI'], dtype='object')
```

In [51]:
```
# Add a constant term to the independent variables
X = sm.add_constant(df2[['Hight', 'weight']])
```

In [53]:
```
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    df2[['Hight', 'weight']], df2['BMI'], test_size=0.2, random_state=42
)

# Add a constant term to the independent variables
X_train = sm.add_constant(X_train)

# Initialize a linear regression model
model = sm.OLS(y_train, X_train)

# Fit the model
results = model.fit()

# Display the regression summary
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    BMI   R-squared:                       0.88
4
Model:                            OLS   Adj. R-squared:                  0.88
3
Method:                 Least Squares   F-statistic:                     969.
7
Date:                Wed, 20 Dec 2023   Prob (F-statistic):          1.23e-11
9
Time:                        21:52:11   Log-Likelihood:                -469.8
8
No. Observations:                 257   AIC:                             945.
8
Df Residuals:                     254   BIC:                             956.
4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         63.8943      1.747     36.569      0.000      60.453      67.33
5
Hight         -0.4077      0.012    -34.983      0.000      -0.431      -0.38
5
weight         0.4169      0.011     38.936      0.000       0.396       0.43
8
==============================================================================
Omnibus:                      454.474   Durbin-Watson:                   1.94
0
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           165553.89
0
Skew:                           9.600   Prob(JB):                         0.0
0
Kurtosis:                     125.848   Cond. No.                     3.31e+0
3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 3.31e+03. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

In [ ]: