

# Cardiovascular Disease Data EDA and Machine Learning Prediction

```
In [4]: # Loading the dataset
df = pd.read_csv('C:/Users/zizhe/Desktop/Leo Zhao/06Cardiovascular Disease Prediction/Data.csv')
df.head(8)
```

	General_Health	Checkup	Exercise	Heart_Disease	Skin_Cancer	Other_Cancer	Depression	Diabetes	Arthritis	Sex	Age_Category	Heig
0	Poor	Within the past 2 years	No	No	No	No	No	No	Yes	Female	70-74	
1	Very Good	Within the past year	No	Yes	No	No	No	Yes	No	Female	70-74	
2	Very Good	Within the past year	Yes	No	No	No	No	Yes	No	Female	60-64	
3	Poor	Within the past year	Yes	Yes	No	No	No	Yes	No	Male	75-79	
4	Good	Within the past year	No	No	No	No	No	No	No	Male	80+	
5	Good	Within the past year	No	No	No	No	Yes	No	Yes	Male	60-64	
6	Fair	Within the past year	Yes	Yes	No	No	No	No	Yes	Male	60-64	
7	Good	Within the past year	Yes	No	No	No	No	No	Yes	Female	65-69	

# Data Preprocessing

```
In [5]: # Checking the shape of the dataset  
df.shape
```

```
Out[5]: (308854, 19)
```

```
In [6]: # Checking for null/missing values  
df.isnull().sum()
```

```
Out[6]: General_Health      0  
Checkup                  0  
Exercise                 0  
Heart_Disease            0  
Skin_Cancer               0  
Other_Cancer              0  
Depression                0  
Diabetes                 0  
Arthritis                 0  
Sex                       0  
Age_Category              0  
Height_(cm)               0  
Weight_(kg)                0  
BMI                      0  
Smoking_History           0  
Alcohol_Consumption        0  
Fruit_Consumption          0  
Green_Vegetables_Consumption 0  
FriedPotato_Consumption    0  
dtype: int64
```

```
In [7]: # Checking the datatypes  
df.dtypes
```

```
Out[7]: General_Health          object
Checkup                  object
Exercise                 object
Heart_Disease            object
Skin_Cancer               object
Other_Cancer              object
Depression                object
Diabetes                 object
Arthritis                object
Sex                      object
Age_Category              object
Height_(cm)               float64
Weight_(kg)               float64
BMI                      float64
Smoking_History           object
Alcohol_Consumption       float64
Fruit_Consumption         float64
Green_Vegetables_Consumption float64
FriedPotato_Consumption   float64
dtype: object
```

The dataset has columns for weight, height, and BMI. However, the BMI column is calculated using the weight and height columns. Hence, the weight and height columns are dropped from the dataset.

```
In [8]: # Drop Column
df.drop(columns=['Weight_(kg)', 'Height_(cm)'], inplace=True)
```

```
In [9]: # Unique values in each column
for i in df.columns:
    print(i, df[i].unique())
```

```
General_Health ['Poor' 'Very Good' 'Good' 'Fair' 'Excellent']
Checkup ['Within the past 2 years' 'Within the past year' '5 or more years ago'
         'Within the past 5 years' 'Never']
Exercise ['No' 'Yes']
Heart_Disease ['No' 'Yes']
Skin_Cancer ['No' 'Yes']
Other_Cancer ['No' 'Yes']
Depression ['No' 'Yes']
Diabetes ['No' 'Yes' 'No, pre-diabetes or borderline diabetes'
          'Yes, but female told only during pregnancy']
Arthritis ['Yes' 'No']
Sex ['Female' 'Male']
Age_Category ['70-74' '60-64' '75-79' '80+' '65-69' '50-54' '45-49' '18-24' '30-34'
              '55-59' '35-39' '40-44' '25-29']
BMI [14.54 28.29 33.47 ... 63.83 19.09 56.32]
Smoking_History ['Yes' 'No']
Alcohol_Consumption [ 0.  4.  3.  8.  30.  2.  12.  1.  5.  10.  20.  17.  16.  6.  25.  28.  15.  7.
                     9. 24. 11. 29. 27. 14. 21. 23. 18. 26. 22. 13. 19.]
Fruit_Consumption [ 30.  12.  8.  16.  2.  1.  60.  0.  7.  5.  3.  6.  90.  28.
                    20.  4.  80.  24.  15.  10.  25.  14.  120.  32.  40.  17.  45.  100.
                    9.  99.  96.  35.  50.  56.  48.  27.  72.  36.  84.  26.  23.  18.
                    21.  42.  22.  11.  112.  29.  64.  70.  33.  76.  44.  39.  75.  31.
                    92.  104.  88.  65.  55.  13.  38.  63.  97.  108.  19.  52.  98.  37.
                    68.  34.  41.  116.  54.  62.  85.]
Green_Vegetables_Consumption [ 16.  0.  3.  30.  4.  12.  8.  20.  1.  10.  5.  2.  6.  60.
                               28.  25.  14.  40.  7.  22.  24.  15.  120.  90.  19.  13.  11.  80.
                               27.  17.  56.  18.  9.  21.  99.  29.  31.  45.  23.  100.  104.  32.
                               48.  75.  36.  35.  112.  26.  50.  33.  96.  52.  76.  84.  34.  97.
                               88.  98.  68.  92.  55.  95.  64.  124.  61.  65.  77.  85.  44.  39.
                               70.  93.  128.  37.  53.]
FriedPotato_Consumption [ 12.  4.  16.  8.  0.  1.  2.  30.  20.  15.  10.  3.  7.  28.
                          5.  9.  6.  120.  32.  14.  60.  33.  48.  25.  24.  21.  90.  13.
                          99.  17.  18.  40.  56.  34.  36.  44.  100.  11.  64.  45.  80.  29.
                          68.  26.  50.  22.  95.  23.  27.  112.  35.  31.  98.  96.  88.  92.
                          19.  76.  49.  97.  128.  41.  37.  42.  52.  72.  46.  124.  84.]
```

The diabetes column contains four values: 'Yes,' 'No,' 'No pre-diabetes or borderline diabetes,' and 'Yes, but female told only during pregnancy.' We will replace the last two values with 'pre-diabetes' and 'gestational diabetes,' respectively.

```
In [10]: df['Diabetes'] = df['Diabetes'].map({'No, pre-diabetes or borderline diabetes': 'Pre-Diabetes', 'Yes, but female told o
```

## Outliner removal

```
In [11]: # columns for outlier removal
cols = ['BMI', 'Alcohol_Consumption', 'Fruit_Consumption', 'Green_Vegetables_Consumption', 'FriedPotato_Consumption']

#Threshold for outlier removal
threshold = 1.5

#Find index of outliers
index = np.where((df[cols] < (Q1 - threshold * IQR)) | (df[cols] > (Q3 + threshold * IQR)))[0]
```

```
In [12]: #Drop outliers
df = df.drop(df.index[index])
```

Descriptive Statistics

```
In [15]: df.describe()
```

```
Out[15]:
```

	BMI	Alcohol_Consumption	Fruit_Consumption	Green_Vegetables_Consumption	FriedPotato_Consumption
<b>count</b>	186777.000000	186777.000000	186777.000000	186777.000000	186777.000000
<b>mean</b>	28.303577	2.505287	18.446104	11.893440	4.899565
<b>std</b>	5.433758	3.777076	10.898445	9.604871	4.261893
<b>min</b>	12.870000	0.000000	0.000000	0.000000	0.000000
<b>25%</b>	24.370000	0.000000	8.000000	4.000000	2.000000
<b>50%</b>	27.550000	0.000000	16.000000	8.000000	4.000000
<b>75%</b>	31.750000	4.000000	30.000000	16.000000	8.000000
<b>max</b>	43.280000	15.000000	56.000000	44.000000	17.000000

```
In [14]: df.head()
```

Out[14]:

	General_Health	Checkup	Exercise	Heart_Disease	Skin_Cancer	Other_Cancer	Depression	Diabetes	Arthritis	Sex	Age_Category	BM
0	Poor	Within the past 2 years	No	No	No	No	No	No	Yes	Female	70-74	14.5%
1	Very Good	Within the past year	No	Yes	No	No	No	Yes	No	Female	70-74	28.2%
2	Very Good	Within the past year	Yes	No	No	No	No	Yes	No	Female	60-64	33.4%
3	Poor	Within the past year	Yes	Yes	No	No	No	Yes	No	Male	75-79	28.7%
4	Good	Within the past year	No	No	No	No	No	No	No	Male	80+	24.3%

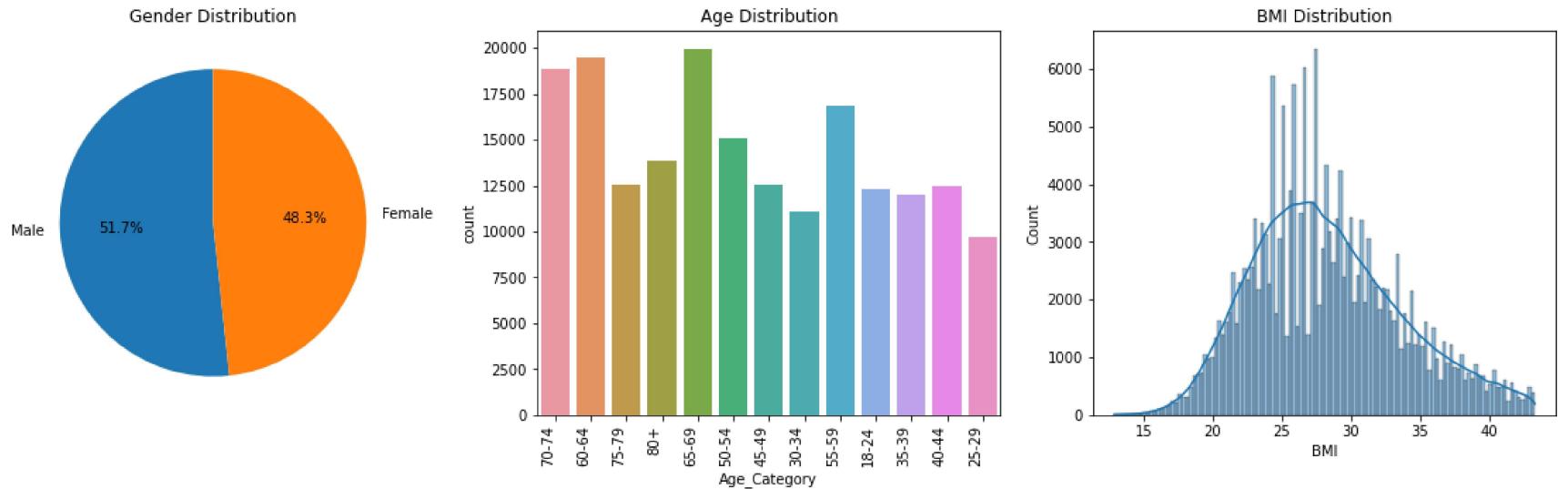
## Exploratory Data Analysis

In the exploratory data analysis, I will examine the dataset to understand it better and analyze the relationship between the features and the target variable. I'll start by assessing the data's distribution across all variables and then investigate the relationship between the features and the target variable.

### Patient demographics

```
In [16]: fig, ax = plt.subplots(1,3,figsize=(20, 5))
ax[0].pie(df['Sex'], labels = ['Male', 'Female'], startangle=90)
ax[0].set_title('Gender Distribution')
sns.countplot(x = 'Age_Category', data = df, ax = ax[1]).set_title('Age Distribution')
sns.histplot(x = 'BMI', data = df, ax = ax[2], kde = True).set_title('BMI Distribution')
```

Out[16]: Text(0.5, 1.0, 'BMI Distribution')

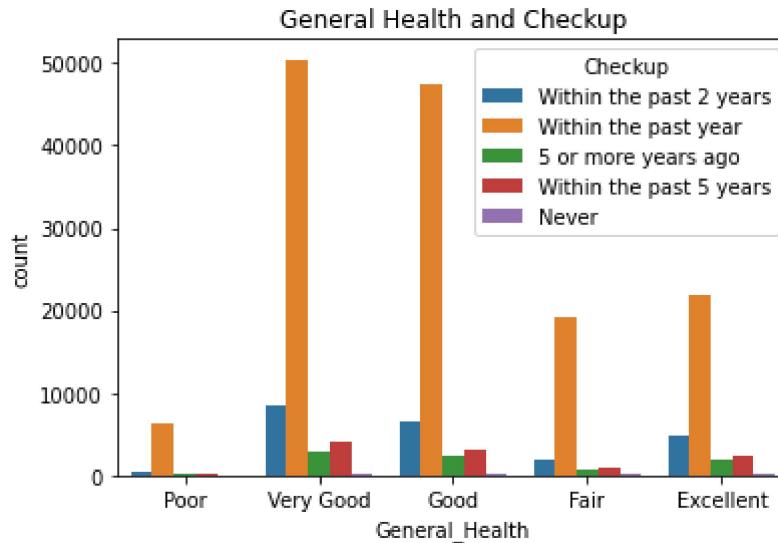


The three graphs above illustrate the patient demographics in the dataset. The pie chart indicates that the majority of patients are male (51.7%), followed closely by females (48.3%). Examining the age distribution, we observe that most patients are older than 45 years, indicating a skew toward older individuals in the dataset. The histogram of BMI reveals that most patients fall within the BMI range of 25 to 30, indicating that a majority are overweight. Based on these observations, I hypothesize that patients with higher BMI values are more likely to have cardiovascular disease.

## General Health and Last Checkup

```
In [17]: sns.countplot(x = 'General_Health', data = df, hue = 'Checkup').set_title('General Health and Checkup')
```

```
Out[17]: Text(0.5, 1.0, 'General Health and Checkup')
```



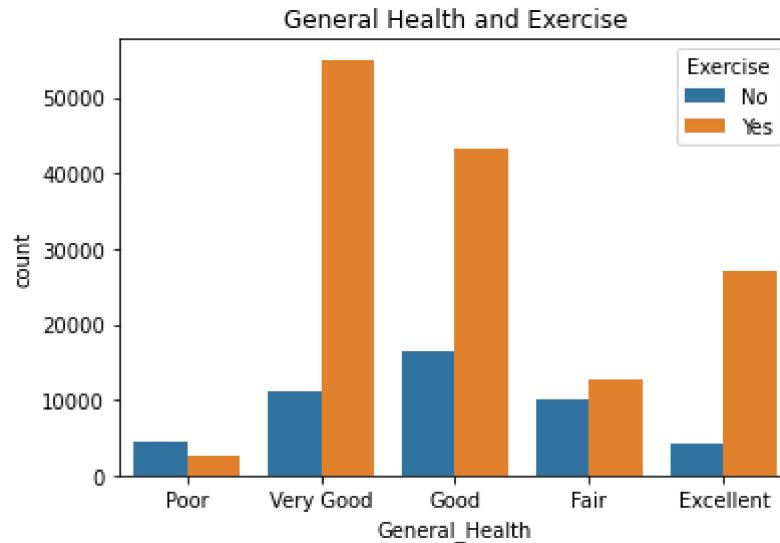
Based on this graph, the majority of people in the dataset report either good or very good general health, with a notable number reporting excellent general health. This suggests that most individuals in the dataset are in good health, while very few report poor general health.

When examining the timing of their last checkup across different general health statuses, it becomes apparent that, for all categories of general health, the majority of people have had their last checkup within the past year. However, there is still a significant portion of the population who have not had a checkup in the last 5 years or more. This increases the likelihood of potential cardiovascular disease.

## Excercise and General Health

```
In [18]: sns.countplot(x = 'General_Health', hue = 'Exercise').set_title('General Health and Exercise')
```

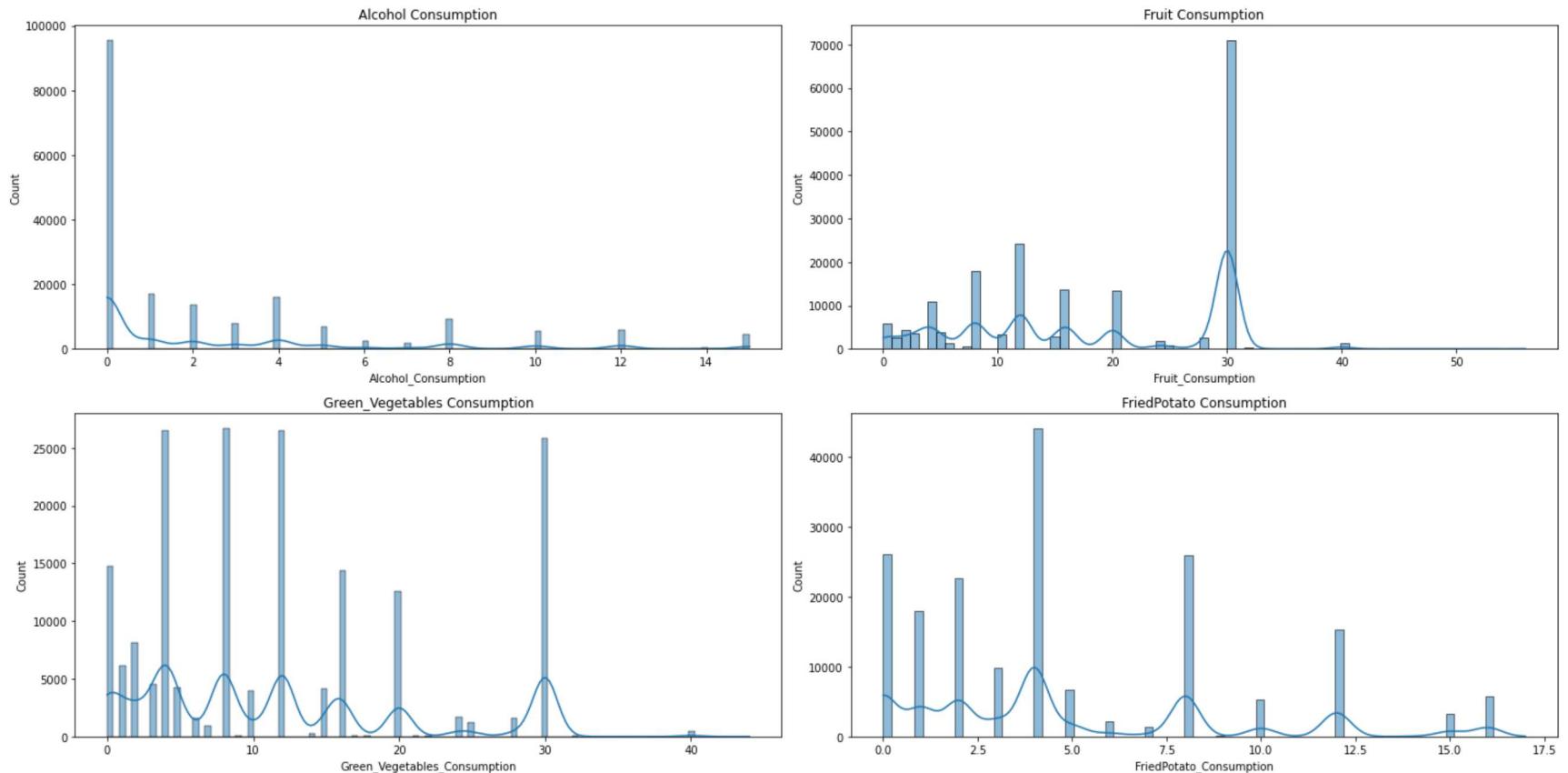
```
Out[18]: Text(0.5, 1.0, 'General Health and Exercise')
```



The role of exercise in general health is evident from this graph. Individuals who exercise regularly are more likely to report good, very good, or excellent health. Conversely, those who do not exercise are more likely to report poor health. This underscores the importance of exercise in maintaining good health.

## Food Consumption

```
In [19]: fig, ax = plt.subplots(2,2,figsize=(20, 10))
sns.histplot(x = 'Alcohol_Consumption', data = df, ax = ax[0,0], kde = True).set_title('Alcohol Consumption')
sns.histplot(x = 'Green_Vegetables_Consumption', data = df, ax = ax[1,0], kde = True).set_title('Green_Vegetables Consumption')
sns.histplot(x = 'FriedPotato_Consumption', data = df, ax = ax[1,1], kde = True).set_title('FriedPotato Consumption')
plt.tight_layout()
```

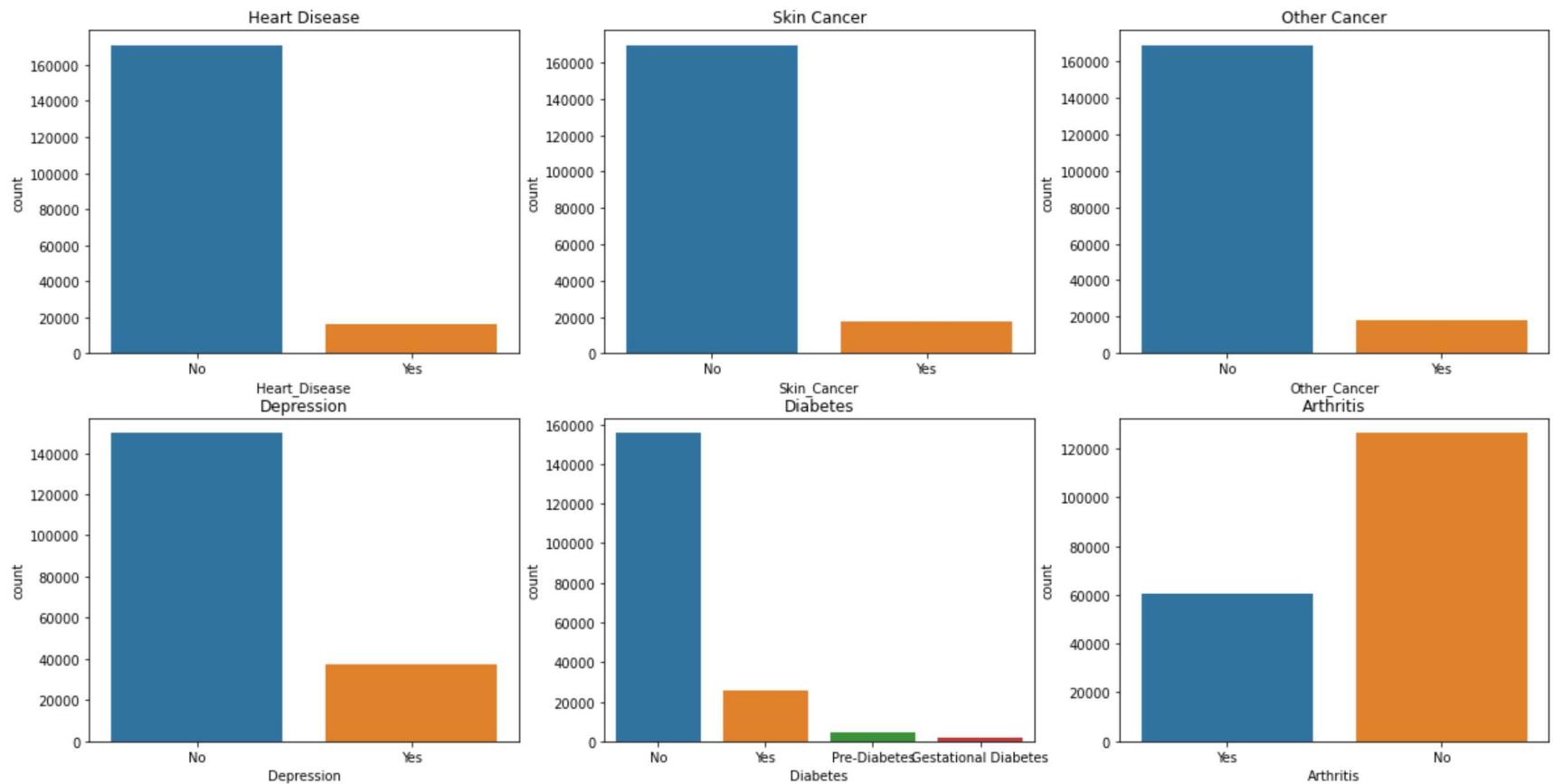


These plots visualize the food and drinking habits of the patients. It's evident that the majority of patients do not consume alcohol. Regarding food habits, most patients consume a higher amount of fruits and green vegetables, which is beneficial for health. However, a significant number of patients also consume fried potatoes, which is less healthy. This suggests that patients who consume fried potatoes and alcohol may be more likely to have cardiovascular disease.

## Medical History

```
In [20]: fig, ax = plt.subplots(figsize=(20, 10))
sns.countplot(x = 'Heart_Disease', data = df, ax = ax[0,0]).set_title('Heart Disease')
sns.countplot(x = 'Skin_Cancer', data = df, ax = ax[0,1]).set_title('Skin Cancer')
sns.countplot(x = 'Other_Cancer', data = df, ax = ax[0,2]).set_title('Other Cancer')
sns.countplot(x = 'Diabetes', data = df, ax = ax[1,1]).set_title('Diabetes')
sns.countplot(x = 'Arthritis', data = df, ax = ax[1,2]).set_title('Arthritis')
```

```
Out[20]: Text(0.5, 1.0, 'Arthritis')
```

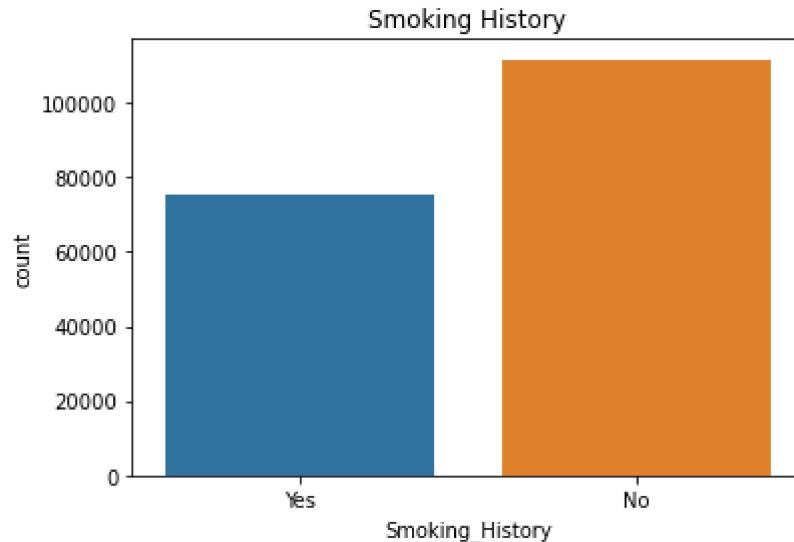


Most patients have no medical conditions, but some do have conditions such as heart disease, skin cancer, other cancers, depression, diabetes, and arthritis. Notably, there is an increased number of patients suffering from depression compared to other medical conditions. This suggests the importance of focusing on mental health in addition to physical health. Additionally, there are patients who are pre-diabetic, and some females suffer from gestational diabetes

## Patient's Smoking History

```
In [21]: sns.countplot(x = 'Smoking_History', data = df )('Smoking History')
```

```
Out[21]: Text(0.5, 1.0, 'Smoking History')
```



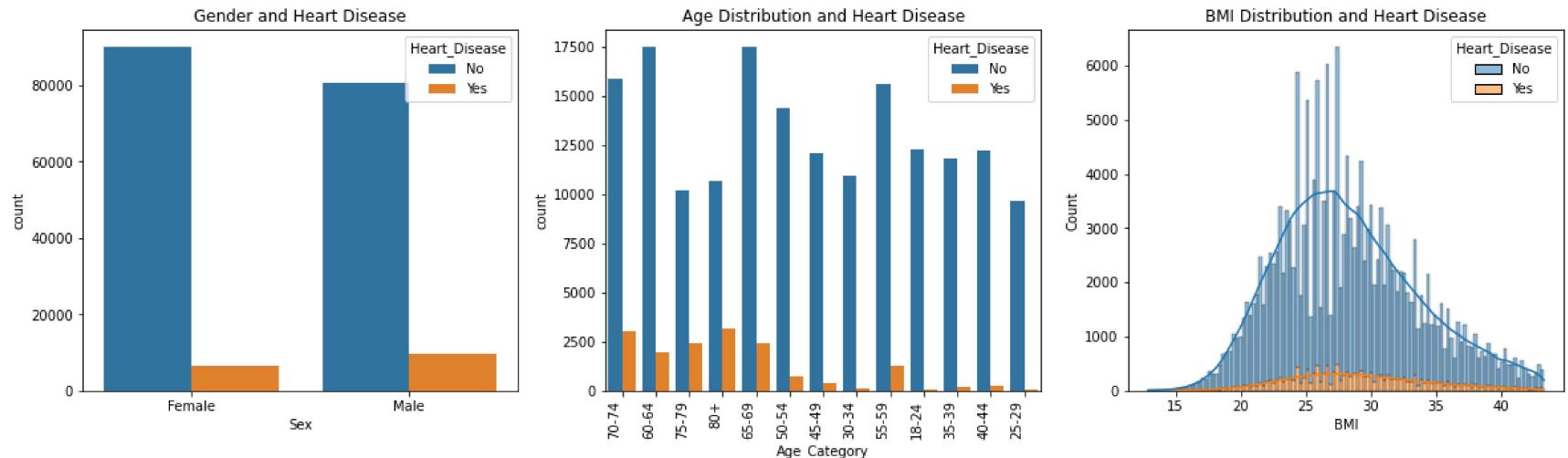
This graph displays the smoking history of the patients in the dataset. The majority of patients have never smoked, but there is a significant number of current smokers. This suggests that patients who are current smokers may be more likely to have cardiovascular disease.

## Target Variable and Independent Variables Visualization

### Patient's Demographics and Heart Disease

```
In [22]: fig, ax = plt.subplots(figsize=(20, 5))
sns.countplot(x = 'Sex', data = df, hue = 'Heart_Disease', ax = ax[0]).set_title('Gender and Heart Disease')
sns.countplot(x = 'Age_Category', data = df, ax = ax[1], hue = 'Heart_Disease').set_title('Age Distribution and Heart D
sns.histplot(x = 'BMI', data = df, ax = ax[2], kde = True, hue = 'Heart_Disease', multiple = 'stack').set_title('BMI Di
```

```
Out[22]: Text(0.5, 1.0, 'BMI Distribution and Heart Disease')
```



Visualizing patient demographics alongside heart disease data provides insights into the relationship between cardiovascular disease and patient characteristics.

First, the Gender graph indicates that males are more susceptible to heart disease compared to females.

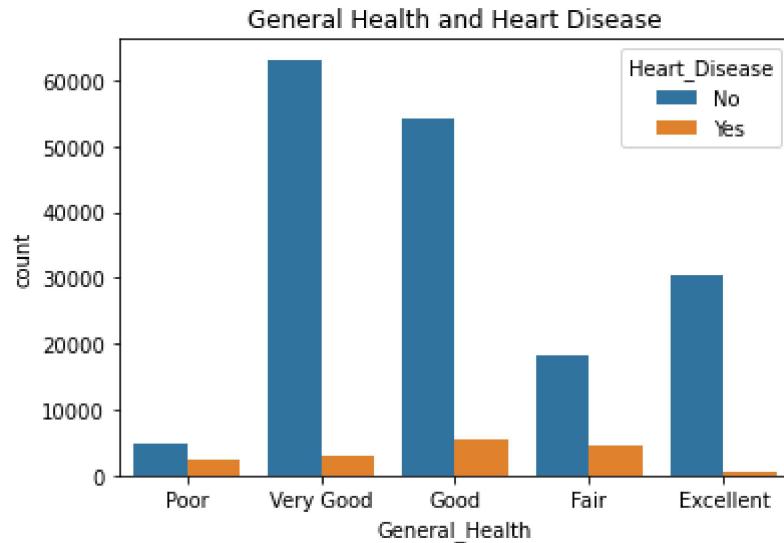
Second, the Age graph reveals that patients over 55 years of age, particularly those in the 80+ age group, have a higher incidence of heart disease, indicating that cardiovascular disease risk increases with age.

Third, the BMI graph shows that patients with a BMI between 25-30 (overweight) have a higher likelihood of heart disease.

## General Health and Heart Disease

```
In [23]: sns.countplot(x = 'General_Health', hue = 'Heart_Disease').set_title('General Health and Heart Disease')
```

```
Out[23]: Text(0.5, 1.0, 'General Health and Heart Disease')
```

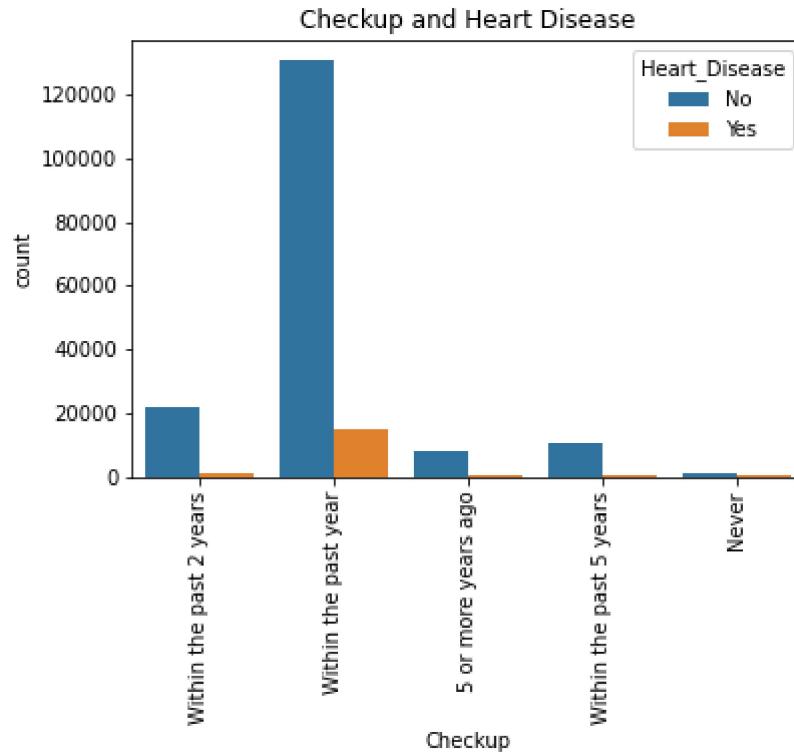


This graph contradicts the belief that healthy patients are less prone to heart disease. Instead, it reveals that patients with very good or good general health have a higher likelihood of heart disease compared to patients with poor general health.

## Checkup and Heart Disease

```
In [24]: sns.countplot(x = 'Checkup').set_title('Checkup and Heart Disease')
plt.xticks(rotation=90)
```

```
Out[24]: (array([0, 1, 2, 3, 4]),
[Text(0, 0, 'Within the past 2 years'),
 Text(1, 0, 'Within the past year'),
 Text(2, 0, '5 or more years ago'),
 Text(3, 0, 'Within the past 5 years'),
 Text(4, 0, 'Never')])
```

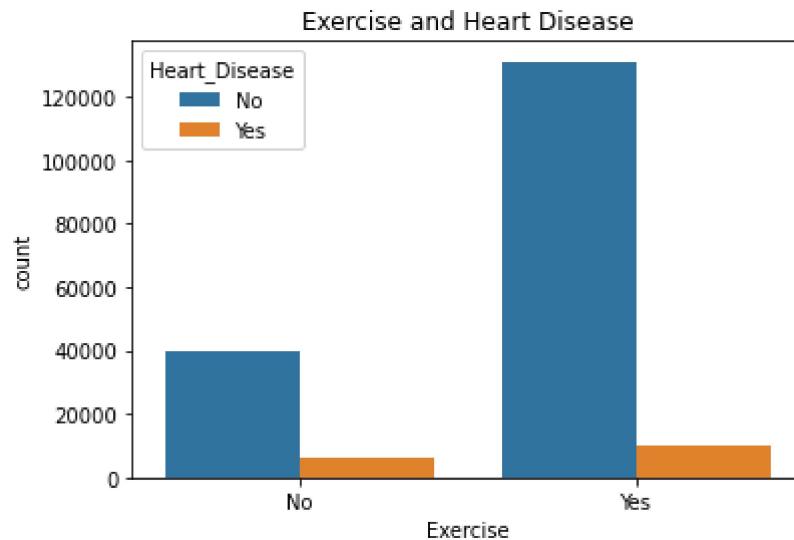


According to this graph, patients who had a checkup in the last year have a higher likelihood of having heart disease. This suggests that patients who undergo regular checkups are more likely to detect cardiovascular disease at an early stage compared to those who do not.

## Excercise and Heart Disease

```
In [25]: sns.countplot(x = 'Exercise', hue = 'Heart_Disease').set_title('Exercise and Heart Disease')
```

```
Out[25]: Text(0.5, 1.0, 'Exercise and Heart Disease')
```

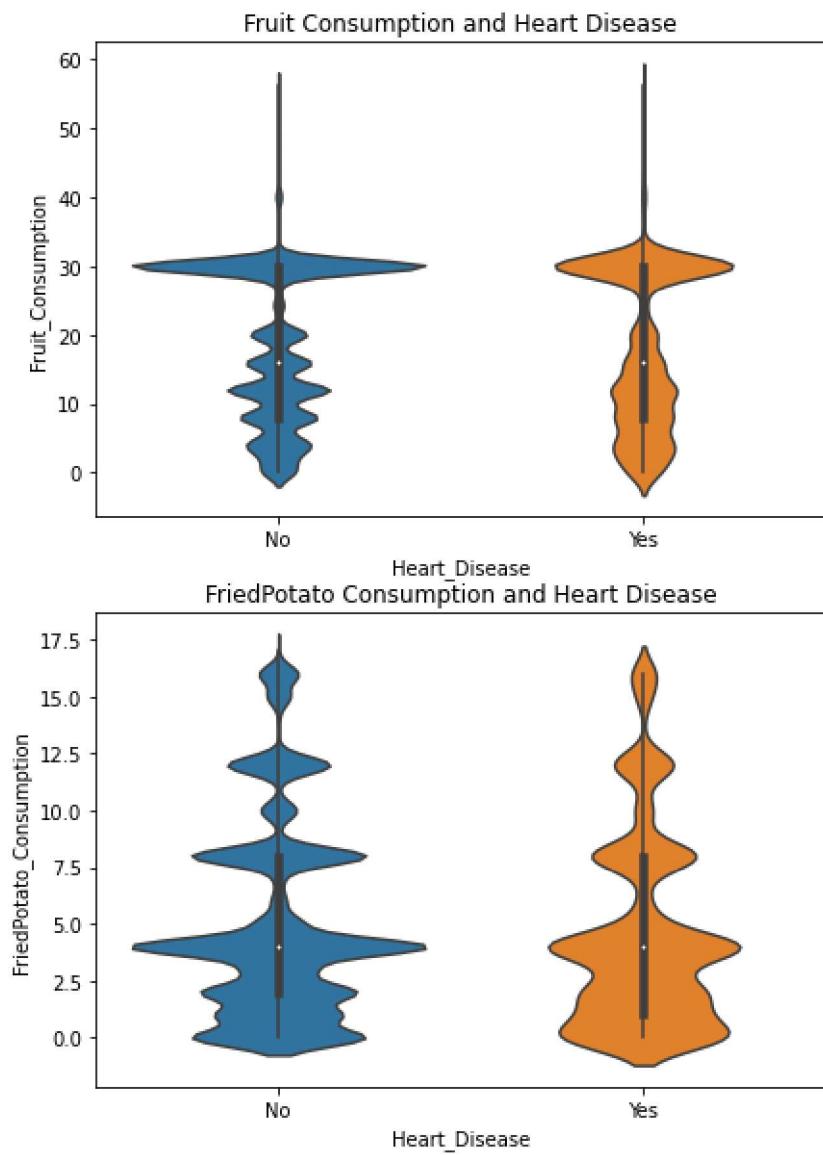
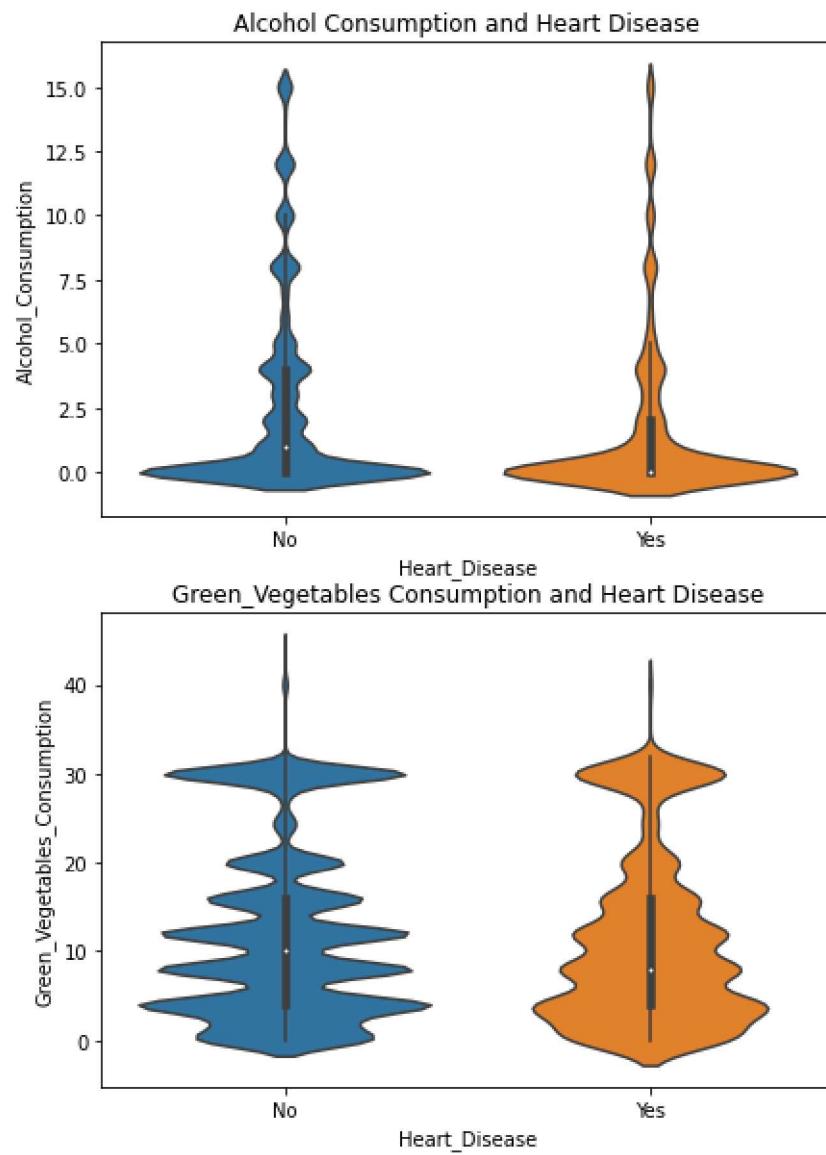


Interestingly, patients who exercise tend to have higher rates of heart disease, contrary to the belief that regular exercise reduces the risk of heart disease. Conversely, this graph indicates that patients who do not exercise are less prone to heart disease. It's possible that patients with weakened hearts exert excessive pressure on their hearts through exercise, potentially contributing to heart disease.

## Food Consumption and Heart Disease

```
In [26]: fig, ax = plt.subplots(figsize=(15, 10))
sns.violinplot(x = 'Heart_Disease', y = 'Alcohol_Consumption', data = df, ax = ax[0,0]).set_title('Alcohol Consumption and Disease')
sns.violinplot(x = 'Heart_Disease', y = 'FriedPotato_Consumption', data = df, ax = ax[1,1]).set_title('FriedPotato Consumption and Heart Disease')
```

Out[26]: Text(0.5, 1.0, 'FriedPotato Consumption and Heart Disease')



These graphs visualize the relationship between patient food and drink habits and heart disease.

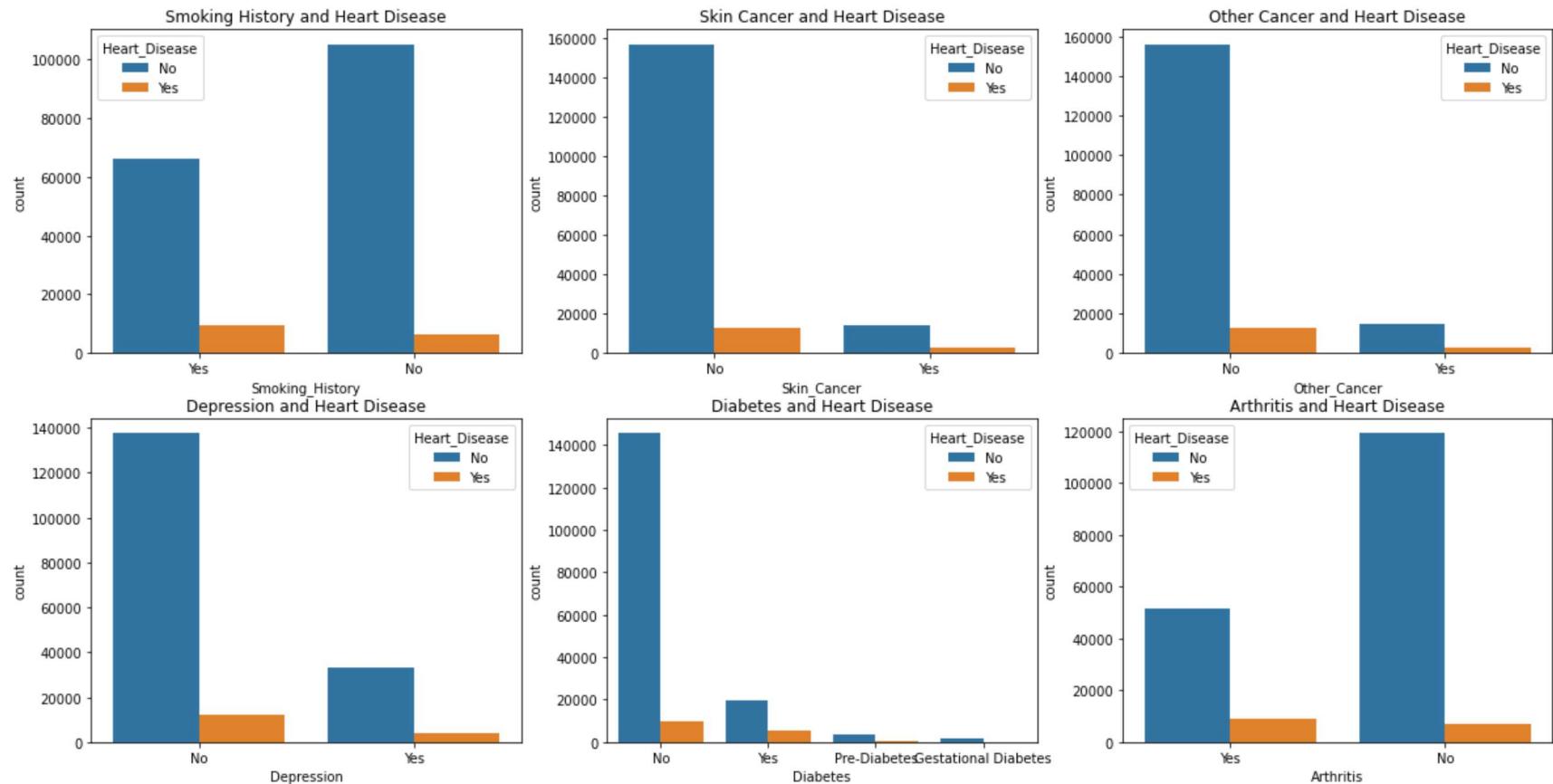
In the alcohol consumption graph, patients with higher alcohol consumption have lower chances of heart disease.

Regarding food habits, patients with increased consumption of fruits and green vegetables tend to have a lower risk of heart disease. Conversely, those with higher consumption of fried potatoes tend to have a higher risk of heart disease.

## Medical History and Heart Disease

```
In [27]: fig, ax = plt.subplots(figsize=(20, 10))
sns.countplot(x = 'Smoking_History', data = df, ax = ax[0,0], hue = 'Heart_Disease').set_title('Smoking History and Heart Disease')
sns.countplot(x = 'Depression', data = df, ax = ax[1,0], hue = 'Heart_Disease').set_title('Depression and Heart Disease')
sns.countplot(x = 'Diabetes', data = df, ax = ax[1,1], hue = 'Heart_Disease').set_title('Diabetes and Heart Disease')
sns.countplot(x = 'Arthritis', data = df, ax = ax[1,2], hue = 'Heart_Disease').set_title('Arthritis and Heart Disease')
```

Out[27]: Text(0.5, 1.0, 'Arthritis and Heart Disease')



These graphs illustrate the relationship between patient medical history and heart disease.

1. In the first graph on smoking history, patients who smoke or used to smoke tend to have a higher incidence of cardiovascular disease.

2,The second graph shows that patients without skin cancer have a higher likelihood of having heart disease compared to those with skin cancer.

3,The third graph indicates that patients without any type of cancer are more likely to have cardiovascular disease.

4,In the fourth graph, patients without depression have a higher incidence of heart disease compared to those with depression.

5,The fifth graph reveals that patients without diabetes are more likely to have heart disease, while pre-diabetes or gestational diabetes seem to have little or no effect on heart disease.

6,The final graph demonstrates that patients without arthritis are more likely to have heart disease compared to those with arthritis.

Based on the information presented above, it appears that patients' medical histories do not have a major effect on the likelihood of having cardiovascular disease. However, it's important to note that this conclusion is based on the data available and may not account for all possible factors or interactions.

## Data Preprocessing 2

### Label Encoding the Categorical Variables

In [28]:

```
from sklearn.preprocessing import LabelEncoder

# List of categorical variables
cols = ['General_Health', 'Checkup', 'Exercise', 'Heart_Disease', 'Skin_Cancer', 'Other_Cancer', 'Depression', 'Diabetes', 'Art

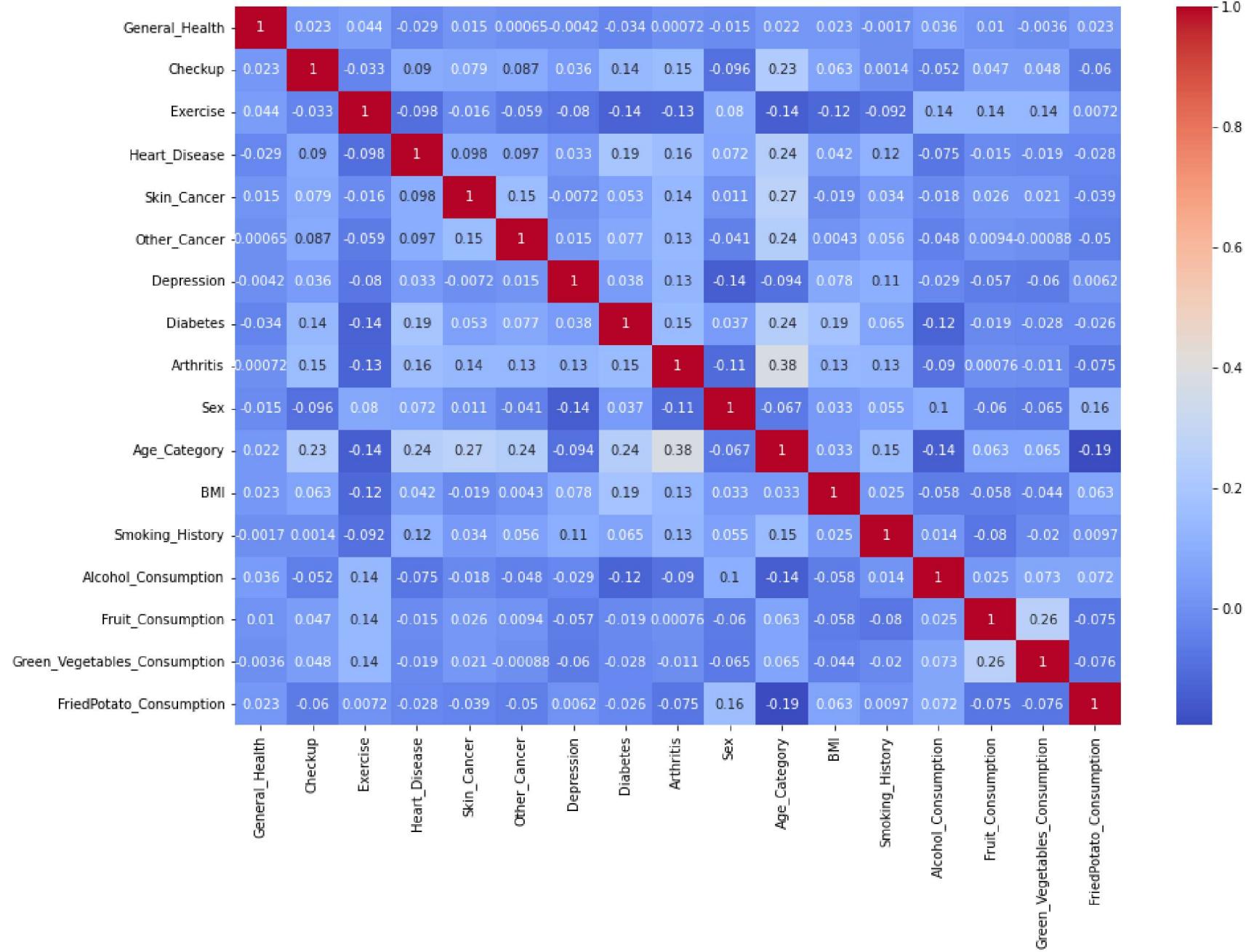
# Label encoding object
le = LabelEncoder()
df[i] = le.transform(df[i])
print(i, df[i].unique())
```

```
General_Health [3 4 2 1 0]
Checkup [2 4 0 3 1]
Exercise [0 1]
Heart_Disease [0 1]
Skin_Cancer [0 1]
Other_Cancer [0 1]
Depression [0 1]
Diabetes [1 3 2 0]
Arthritis [1 0]
Sex [0 1]
Age_Category [10 8 11 12 9 6 5 2 7 0 3 4 1]
Smoking_History [1 0]
```

## Correlation Matrix Heatmap

```
In [29]: plt.figure(figsize=(15,10))
heatmap(df.corr(), cmap = 'coolwarm')
```

```
Out[29]: <AxesSubplot:>
```



There is no strong or significant correlation among the variables.

# Train Test Split

```
In [30]: from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(df.drop(columns = ['Heart_Disease']), df['Heart_Disease'])
```

## Cardiovascular Disease Prediction

For predicting the cardiovascular disease, I have used the following classification models:

1. Random Forest Classifier
2. Decision Tree Classifier
3. Logistic Regression

## Random Forest Classifier

```
In [31]: from sklearn.ensemble import RandomForestClassifier  
  
# Create Random Forest object  
rfc = RandomForestClassifier(random_state=0, max_features='sqrt', class_weight='balanced')
```

```
In [32]: # Training the model  
rfc.fit(X_train, y_train)
```

```
Out[32]: RandomForestClassifier(class_weight='balanced', max_features='sqrt',  
n_estimators=200, random_state=0)
```

```
In [33]: # Training accuracy  
rfc.score(X_train, y_train)
```

```
Out[33]: 0.9999866150005688
```

```
In [43]: # Predicting the test set results  
rfc_pred= rfc.predict(X_test)
```

## Decision Tree Classifier

```
In [34]: from sklearn.tree import DecisionTreeClassifier  
  
# Create Decision Tree object  
dtc = DecisionTreeClassifier(random_state=0, max_depth= 12, min_samples_split=2, class_weight='balanced')
```

```
In [35]: # Training the model  
dtc.fit(X_train, y_train)
```

```
Out[35]: DecisionTreeClassifier(class_weight='balanced', max_depth=12,  
min_samples_leaf=2, random_state=0)
```

```
In [36]: # Training accuracy  
dtc.score(X_train, y_train)
```

```
Out[36]: 0.73877835110192
```

```
In [37]: # Predicting the test set results  
dtc_pred = dtc.predict(X_test)
```

## Logistic Regression

```
In [38]: from sklearn.linear_model import LogisticRegression  
  
lr = LogisticRegression()
```

```
In [39]: #Training the model  
lr.fit(X_train, y_train)
```

```
C:\Users\zizhe\New folder\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to  
converge (status=1):  
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max\_iter) or scale the data as shown in:  
<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:  
[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
Out[39]: LogisticRegression()
```

```
In [40]: #Training accuracy  
lr.score(X_train, y_train)
```

```
Out[40]: 0.91439623613816
```

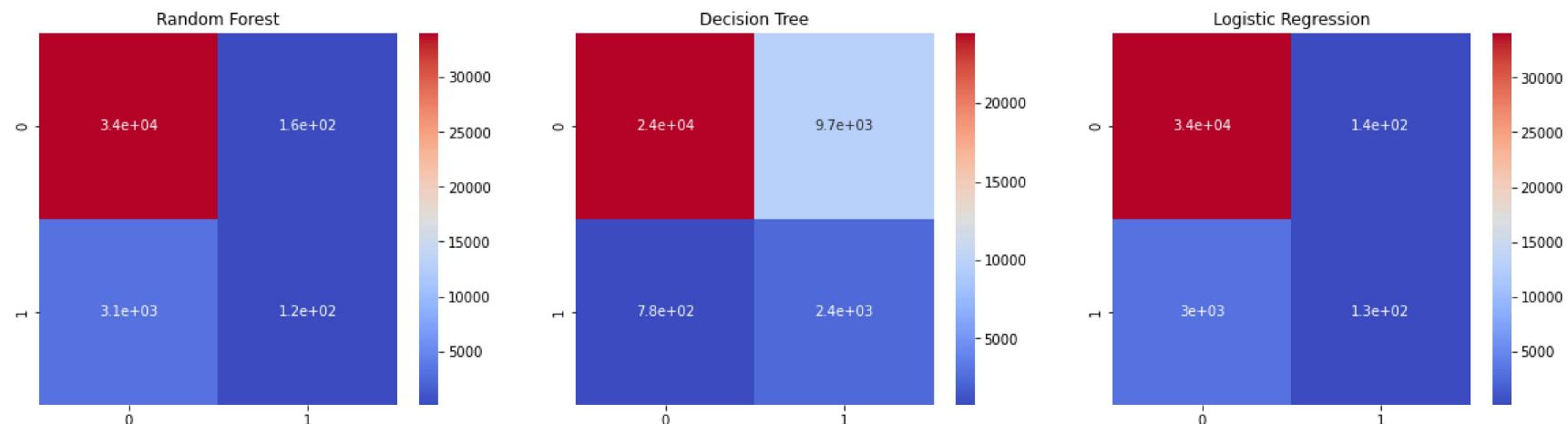
```
In [41]: #Predicting the test set results  
lr_pred = lr.predict(X_test)
```

## Model Evaluation

### Confusion Matrix

```
In [44]: from sklearn.metrics import confusion_matrix  
fig, ax = plt.subplots(figsize = (20,5))  
sns.heatmap(confusion_matrix(y_test, rfc_pred), cmap = 'coolwarm', ax = ax[0]).set_title('Random Forest')  
sns.heatmap(confusion_matrix(y_test, dtc_pred), cmap = 'coolwarm', ax = ax[1]).set_title('Decision Tree')  
sns.heatmap(confusion_matrix(y_test, lr_pred), cmap = 'coolwarm', ax = ax[2]).set_title('Logistic Regression')
```

```
Out[44]: Text(0.5, 1.0, 'Logistic Regression')
```



```
In [45]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score  
print('Random Forest')  
print('Accuracy Score: ', accuracy_score(y_test, rfc_pred))  
print('Precision Score: ', precision_score(y_test, rfc_pred))  
print('Recall Score: ', recall_score(y_test, rfc_pred))  
print('F1 Score: ', f1_score(y_test, rfc_pred))
```

```
Random Forest  
Accuracy Score: 0.9137755648356355  
Precision Score: 0.4166666666666667  
Recall Score: 0.03622047244094488  
F1 Score: 0.06664734859461026
```

```
In [46]: print('Decision Tree')  
print('Accuracy Score: ', accuracy_score(y_test, dtc_pred))  
print('Precision Score: ', precision_score(y_test, dtc_pred))  
print('Recall Score: ', recall_score(y_test, dtc_pred))  
print('F1 Score: ', f1_score(y_test, dtc_pred))
```

```
Decision Tree  
Accuracy Score: 0.718920655316415  
Precision Score: 0.19753902056321745  
Recall Score: 0.7533858267716536  
F1 Score: 0.31300706621303326
```

```
In [47]: print('Logistic Regression')  
print('Accuracy Score: ', accuracy_score(y_test, lr_pred))  
print('Precision Score: ', precision_score(y_test, lr_pred))  
print('Recall Score: ', recall_score(y_test, lr_pred))  
print('F1 Score: ', f1_score(y_test, lr_pred))
```

```
Logistic Regression  
Accuracy Score: 0.9147660349073776  
Precision Score: 0.4830188679245283  
Recall Score: 0.04031496062992126  
F1 Score: 0.07441860465116279
```

## Conclusion

From the exploratory data analysis, we observed that the risk of cardiovascular disease increases with age, with individuals above 55 years being particularly prone to this condition. The highest number of cardiovascular disease cases was found in patients aged 80 and above. Additionally, higher BMI levels were associated with an increased likelihood of cardiovascular disease.

Interestingly, older patients who exercise appeared to be more prone to cardiovascular disease, possibly due to the strain on the heart. Dietary habits also played a role, as patients who consumed larger quantities of fruits and vegetables had a lower risk, while those who consumed fried potatoes had a higher risk.

Furthermore, smoking or a history of smoking was linked to an increased risk of cardiovascular disease. Surprisingly, previous medical history related to cancer, arthritis, diabetes, or depression had no major effect on the likelihood of cardiovascular disease.

