cardiovascular-disease-prediction

June 21, 2024

[1]:
```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

[3]:
```
# Loading the dataset
df = pd.read_csv("C:\\Users\\akash\\Downloads\\CVD_cleaned.csv")
df.head()
```

[3]:
```
  General_Health                  Checkup Exercise Heart_Disease Skin_Cancer  \
0           Poor  Within the past 2 years       No            No          No
1      Very Good     Within the past year       No           Yes          No
2      Very Good     Within the past year      Yes            No          No
3           Poor     Within the past year      Yes           Yes          No
4           Good     Within the past year       No            No          No

  Other_Cancer Depression Diabetes Arthritis     Sex Age_Category  \
0           No         No       No       Yes  Female        70-74
1           No         No      Yes        No  Female        70-74
2           No         No      Yes        No  Female        60-64
3           No         No      Yes        No    Male        75-79
4           No         No       No        No    Male          80+

   Height_(cm)  Weight_(kg)    BMI Smoking_History  Alcohol_Consumption  \
0        150.0        32.66  14.54             Yes                  0.0
1        165.0        77.11  28.29              No                  0.0
2        163.0        88.45  33.47              No                  4.0
3        180.0        93.44  28.73              No                  0.0
4        191.0        88.45  24.37             Yes                  0.0

   Fruit_Consumption  Green_Vegetables_Consumption  FriedPotato_Consumption
0               30.0                          16.0                     12.0
1               30.0                           0.0                      4.0
2               12.0                           3.0                     16.0
3               30.0                          30.0                      8.0
4                8.0                           4.0                      0.0
```

0.1 Data Preprocessing

[4]:
```
# Checking the shape of the dataset
df.shape
```

[4]:
```
(308854, 19)
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 308854 entries, 0 to 308853
Data columns (total 19 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   General_Health                308854 non-null  object
 1   Checkup                       308854 non-null  object
 2   Exercise                      308854 non-null  object
 3   Heart_Disease                 308854 non-null  object
 4   Skin_Cancer                   308854 non-null  object
 5   Other_Cancer                  308854 non-null  object
 6   Depression                    308854 non-null  object
 7   Diabetes                      308854 non-null  object
 8   Arthritis                     308854 non-null  object
 9   Sex                           308854 non-null  object
 10  Age_Category                  308854 non-null  object
 11  Height_(cm)                   308854 non-null  float64
 12  Weight_(kg)                   308854 non-null  float64
 13  BMI                           308854 non-null  float64
 14  Smoking_History               308854 non-null  object
 15  Alcohol_Consumption           308854 non-null  float64
 16  Fruit_Consumption             308854 non-null  float64
 17  Green_Vegetables_Consumption  308854 non-null  float64
 18  FriedPotato_Consumption       308854 non-null  float64
dtypes: float64(7), object(12)
memory usage: 44.8+ MB
```

[6]:
```
# Checking the datatypes
df.dtypes
```

[6]:
```
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
```

[7]:
```
df.duplicated().sum() # No duplicates found in dataset
```

[7]:
```
80
```

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

[9]:
```
# Checking for null/missing values
df.isnull().sum()
```

[9]:
```
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
BMI                             0
Smoking_History                 0
Alcohol_Consumption             0
Fruit_Consumption               0
Green_Vegetables_Consumption    0
FriedPotato_Consumption         0
dtype: int64
```

[10]:
```
#The dataset has columns - 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
```

[11]:
```
print(df.columns)
```

```
Index(['General_Health', 'Checkup', 'Exercise', 'Heart_Disease', 'Skin_Cancer',
       'Other_Cancer', 'Depression', 'Diabetes', 'Arthritis', 'Sex',
       'Age_Category', 'BMI', 'Smoking_History', 'Alcohol_Consumption',
       'Fruit_Consumption', 'Green_Vegetables_Consumption',
       'FriedPotato_Consumption'],
      dtype='object')
```

[12]:
```
# 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.
```

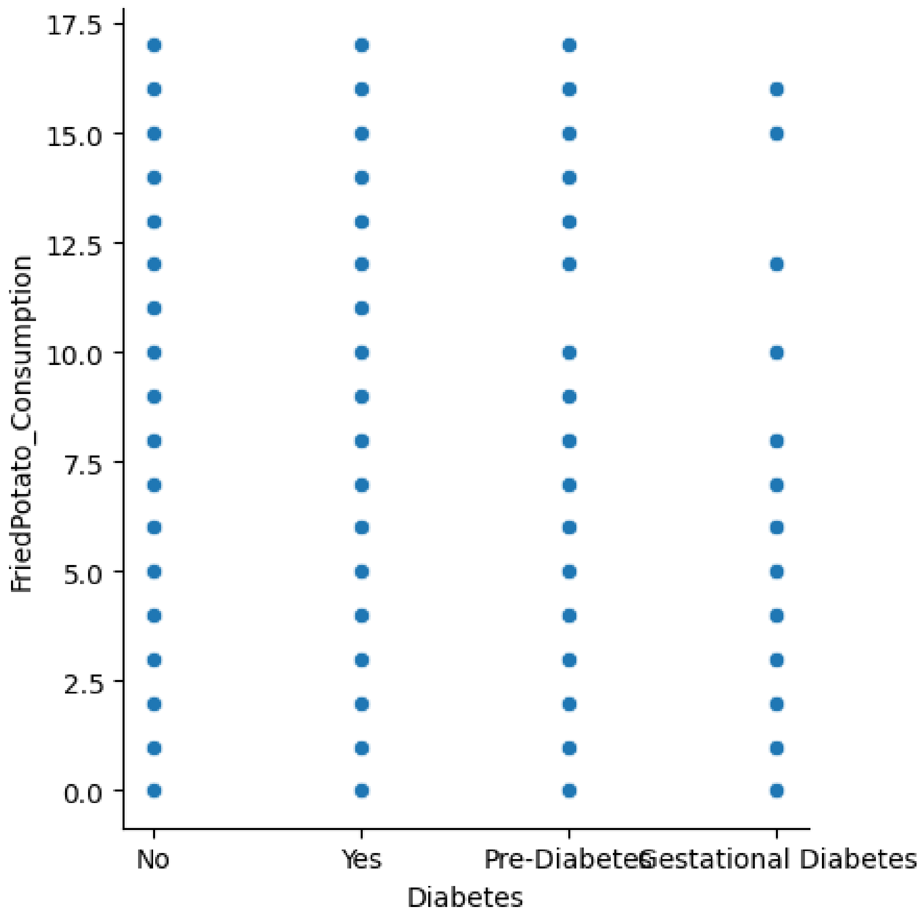
```
 99.  17.  18.  40.  56.  34.  36.  44. 100.  11.  64.  45.  80.  29.
 68.  26.  50.  22.  95.  23.  27. 112.  35.  31.  98.  96.  88.  92.
 19.  76.  49.  97. 128.  41.  37.  42.  52.  72.  46. 124.  84.]
```

[13]:
```
df['Diabetes'] = df['Diabetes'].map({
    'No, pre-diabetes or borderline diabetes': 'Pre-Diabetes',
    'Yes, but female told only during pregnancy': 'Gestational Diabetes',
    'Yes': 'Yes',
    'No': 'No'
})
```

[14]:
```
# columns for outlier removal
cols  = ['BMI', 'Alcohol_Consumption', 'Fruit_Consumption',␣
 ↪'Green_Vegetables_Consumption', 'FriedPotato_Consumption']

#IQR for the selected columns
Q1 = df[cols].quantile(0.25)
Q3 = df[cols].quantile(0.75)
IQR = Q3 - Q1

#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]

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

[15]:
```
sns.relplot(x="Diabetes",y="FriedPotato_Consumption",data=df)
```

[15]:
```
<seaborn.axisgrid.FacetGrid at 0x2d74cd20e60>
```



[16]:
```
df.describe()
```

[16]:
```
                 BMI  Alcohol_Consumption  Fruit_Consumption  \
count  186777.000000        186777.000000      186777.000000
mean       28.303577             2.505287          18.446104
std         5.433758             3.777076          10.898445
min        12.870000             0.000000           0.000000
25%        24.370000             0.000000           8.000000
50%        27.550000             0.000000          16.000000
75%        31.750000             4.000000          30.000000
max        43.280000            15.000000          56.000000

       Green_Vegetables_Consumption  FriedPotato_Consumption
count                 186777.000000            186777.000000
mean                      11.893440                 4.899565
std                        9.604871                 4.261893
min                        0.000000                 0.000000
25%                        4.000000                 2.000000
```

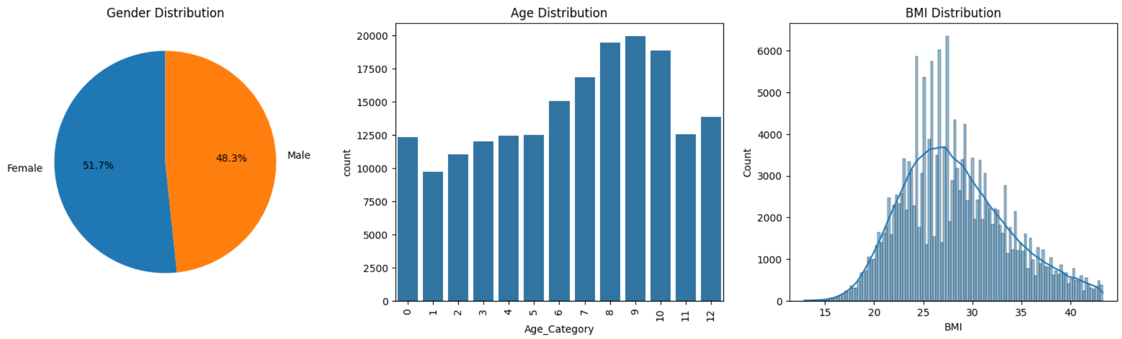
```
50%                          8.000000                   4.000000
75%                         16.000000                   8.000000
max                         44.000000                  17.000000
```

[17]:
```python
df.head()
```

[17]:
```
  General_Health                 Checkup Exercise Heart_Disease Skin_Cancer  \
0           Poor  Within the past 2 years       No            No          No
1      Very Good     Within the past year       No           Yes          No
2      Very Good     Within the past year      Yes            No          No
3           Poor     Within the past year      Yes           Yes          No
4           Good     Within the past year       No            No          No

  Other_Cancer Depression Diabetes Arthritis     Sex Age_Category    BMI  \
0           No         No       No       Yes  Female        70-74  14.54
1           No         No      Yes        No  Female        70-74  28.29
2           No         No      Yes        No  Female        60-64  33.47
3           No         No      Yes        No    Male        75-79  28.73
4           No         No       No        No    Male          80+  24.37

  Smoking_History  Alcohol_Consumption  Fruit_Consumption  \
0             Yes                  0.0               30.0
1              No                  0.0               30.0
2              No                  4.0               12.0
3              No                  0.0               30.0
4             Yes                  0.0                8.0

   Green_Vegetables_Consumption  FriedPotato_Consumption
0                          16.0                     12.0
1                           0.0                      4.0
2                           3.0                     16.0
3                          30.0                      8.0
4                           4.0                      0.0
```

[33]:
```python
fig, ax = plt.subplots(1,3,figsize=(20, 5))
ax[0].pie(df['Sex'].value_counts(), labels = ['Female', 'Male'], autopct='%1.
 ↪1f%%', startangle=90)
ax[0].set_title('Gender Distribution')
sns.countplot(x = 'Age_Category', data = df, ax = ax[1]).set_title('Age␣
 ↪Distribution')
ax[1].tick_params(axis='x', rotation=90)
sns.histplot(x = 'BMI', data = df, ax = ax[2], kde = True).set_title('BMI␣
 ↪Distribution')
```

[33]:
```
Text(0.5, 1.0, 'BMI Distribution')
```

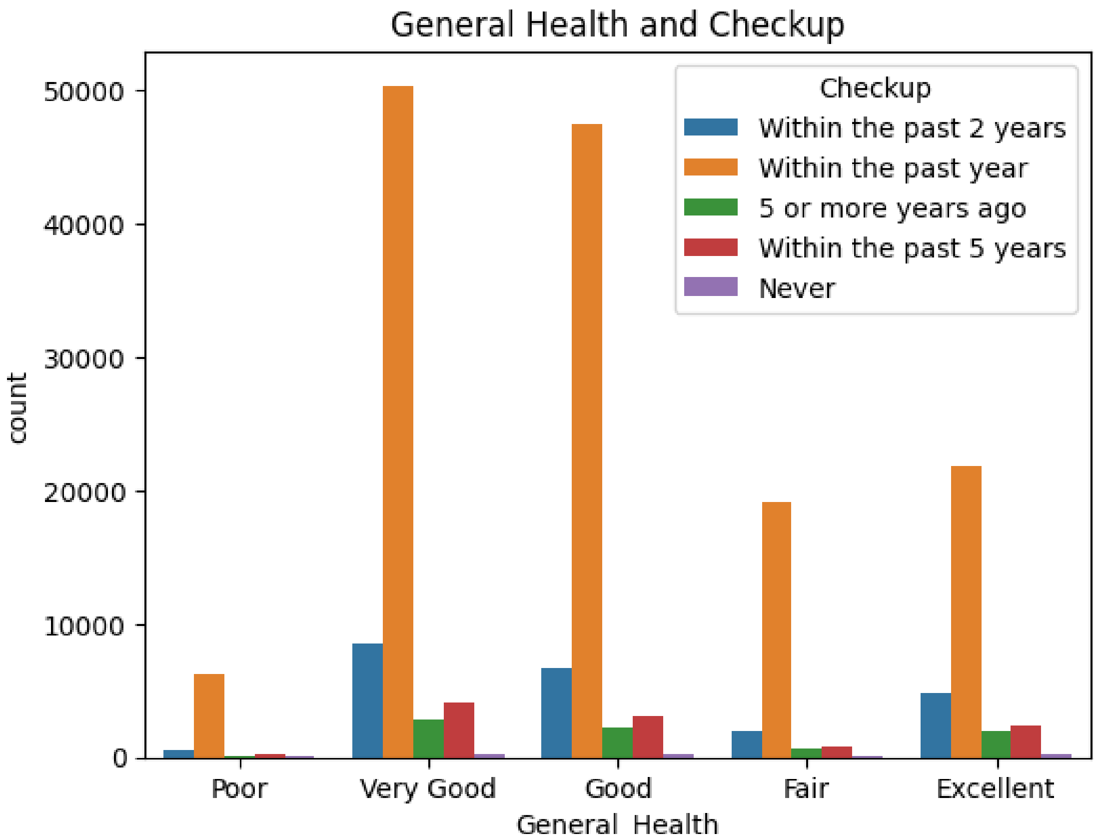


The above three graphs explains the patient demographics in the dataset. From the pie chart, it is clear that majority of ptients are male with 52% followed by females with 48%. Looking at the age distribution, we came to know that majority of patients are older than 45 years of age, this means that the dataset is skewed towards older patients. The histogram of BMI shows that the BMI of majority of patients is between 25 to 30. This means that majority of patients are overweight. Therefore, I build a hypothesis that the patients with higher BMI are more likely to have cardiovascular disease.

General Health and Last Checkup

[19]:
```python
sns.countplot(x = 'General_Health', data = df, hue  = 'Checkup').
 ↪set_title('General Health and Checkup')
```

[19]:
```
Text(0.5, 1.0, 'General Health and Checkup')
```

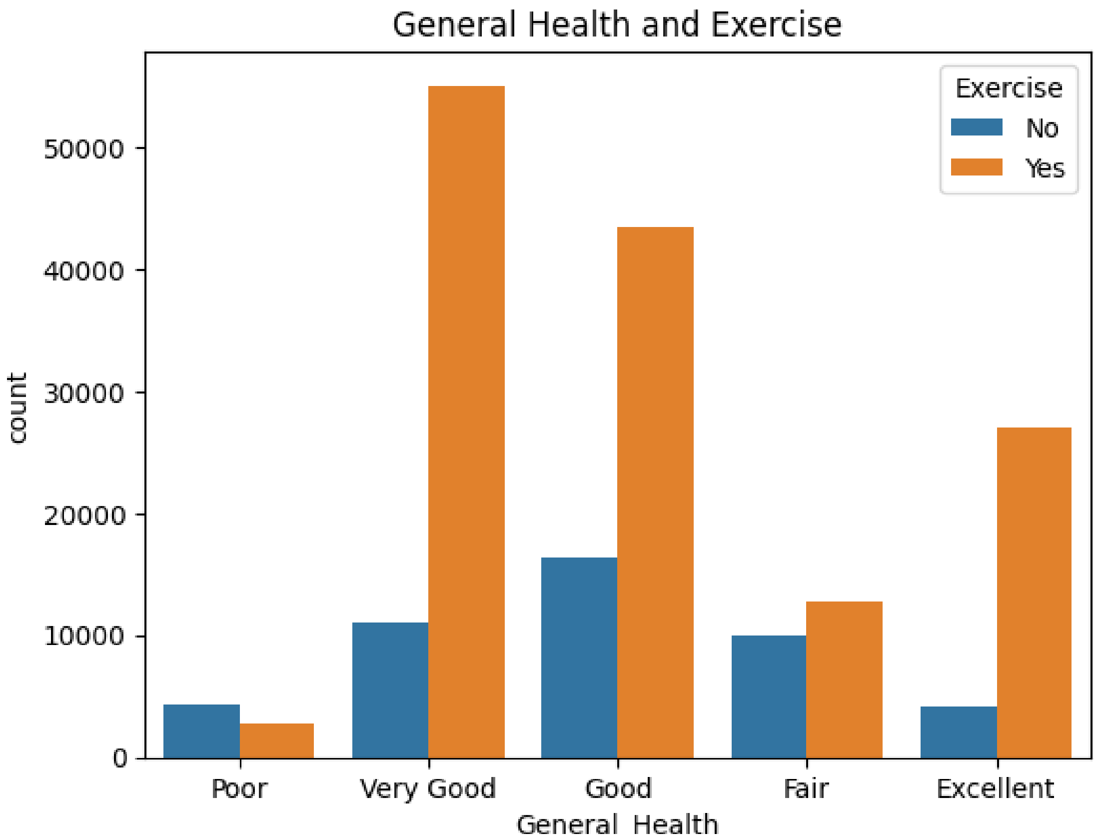


According to this graph most of the people are either in good or very good health, followed by excellent general health. This means that most of the people in the dataset are healthy. Very few of the people are poor general health. Lookinng at the last checkup, in all the general healths, most of the people have had their last checkup within the last year. However, there are still many people who have not had their last checkup within the last 5 years or more. This increases, the chances of having a potential cardiovascular disease.

Excersise and General Health

[20]:
```
sns.countplot(x = 'General_Health', data = df, hue  = 'Exercise').
↪set_title('General Health and Exercise')
```

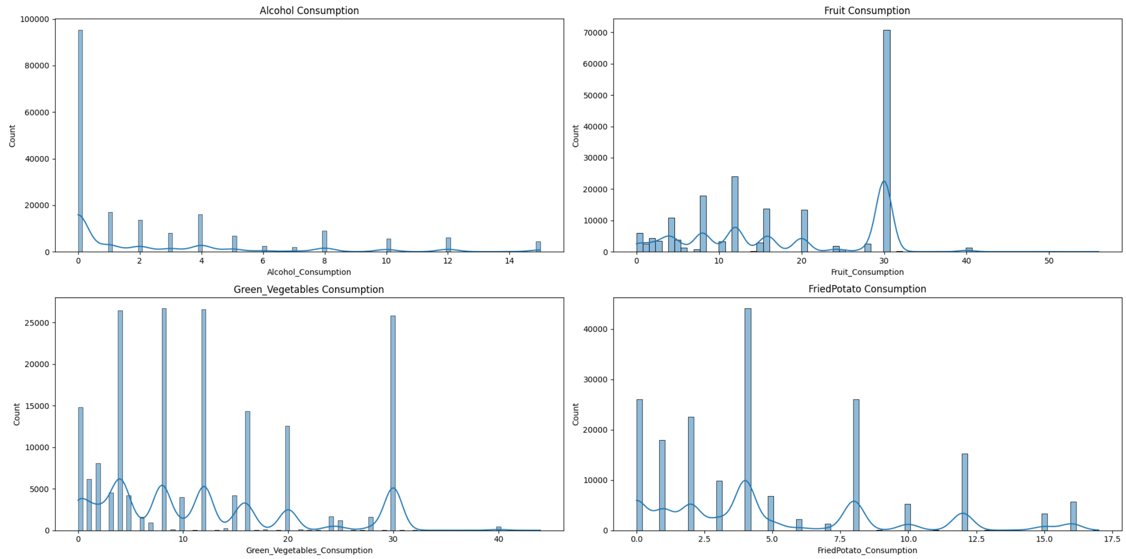
[20]:
```
Text(0.5, 1.0, 'General Health and Exercise')
```



The role of exercise in general health is evident thtough this graph. The people who excersise regularly are more likely to be in good or very good or even in excellent health. However, the people who do not excersise are more likely to be in poor health. This means that excersise plays an important role in maintaining good health.

Food Consumption

[21]:
```
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 = 'Fruit_Consumption', data = df, ax = ax[0,1], kde = True).
 ↪set_title('Fruit 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()
```

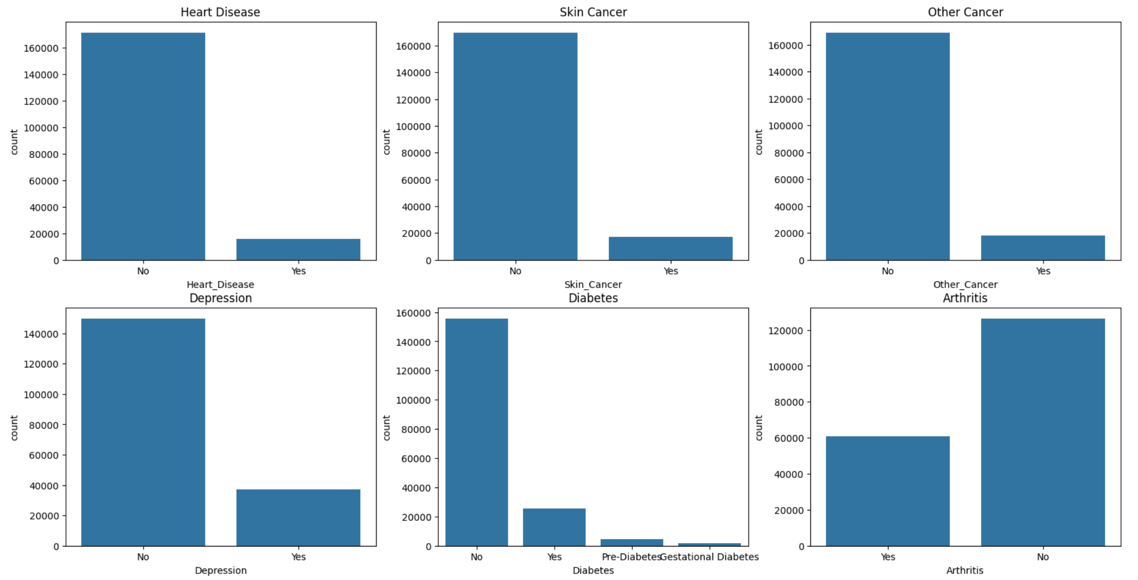


These plots visualizes the food and drinking habits of the patients. From these plots, it is clear that majority of the patients, do not consume alcohol. Coming to the food habits, most of the patients, consume higher amount of fruits and green vegetables which is good for health. However, most of the patients consume fried potatoes which is not good for health. This means that the patients who consume fried potatoes and alcohol are more likely to have cardiovascular disease.

Medical History

[22]:
```
fig, ax = plt.subplots(2,3,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 = 'Depression', data = df, ax = ax[1,0]).set_title('Depression')
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')
```

[22]:
```
Text(0.5, 1.0, 'Arthritis')
```

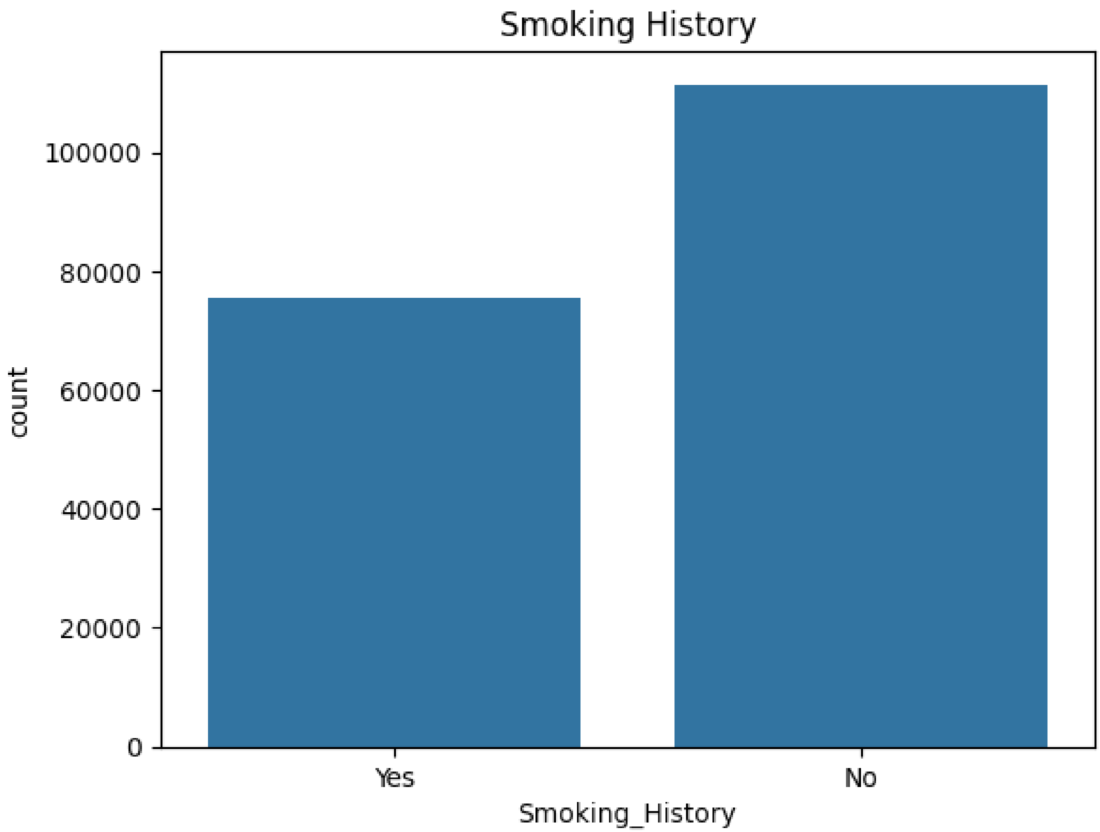


Most of the patients have no medical conditions. However, there are patients who have medical conditions like heart disease, skin cancer, other cancer, depression, diabetes and arthritis. In addition to that, there has been increased number of patients suffering ftom Depression as compared to other medical conditions. This means, the doctor should focus on mental health as well in addtion to physical health. There certain number of patients, who are pre-diabetic and some females suffer from gestational diabetes.

Patient's Smoking History

[23]:
```
sns.countplot(x = 'Smoking_History', data = df ).set_title('Smoking History')
```

[23]:
```
Text(0.5, 1.0, 'Smoking History')
```



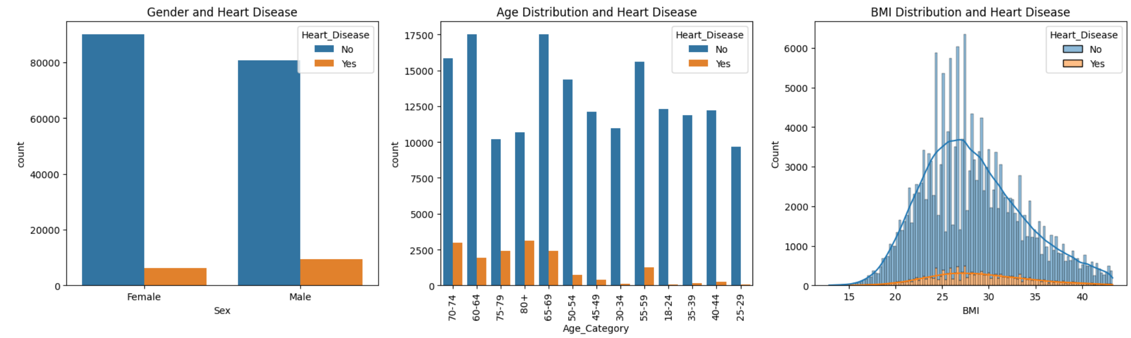
This graph shows the smoking history of the patients in the dataset. Majority of the patients have never smoked. However, there are patients in huge number who are current smokers. This means that the patients who are current smokers are more likely to have cardiovascular disease.

0.1.1 Target Variable and Independent Variables Visualization

Patient's Demographics and Heart Disease

[24]:
```
fig, ax = plt.subplots(1,3,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 Disease')
ax[1].tick_params(axis='x', rotation=90)
sns.histplot(x = 'BMI', data = df, ax = ax[2], kde = True, hue =␣
 ↪'Heart_Disease', multiple = 'stack').set_title('BMI Distribution and Heart␣
 ↪Disease')
```

[24]:
```
Text(0.5, 1.0, 'BMI Distribution and Heart Disease')
```

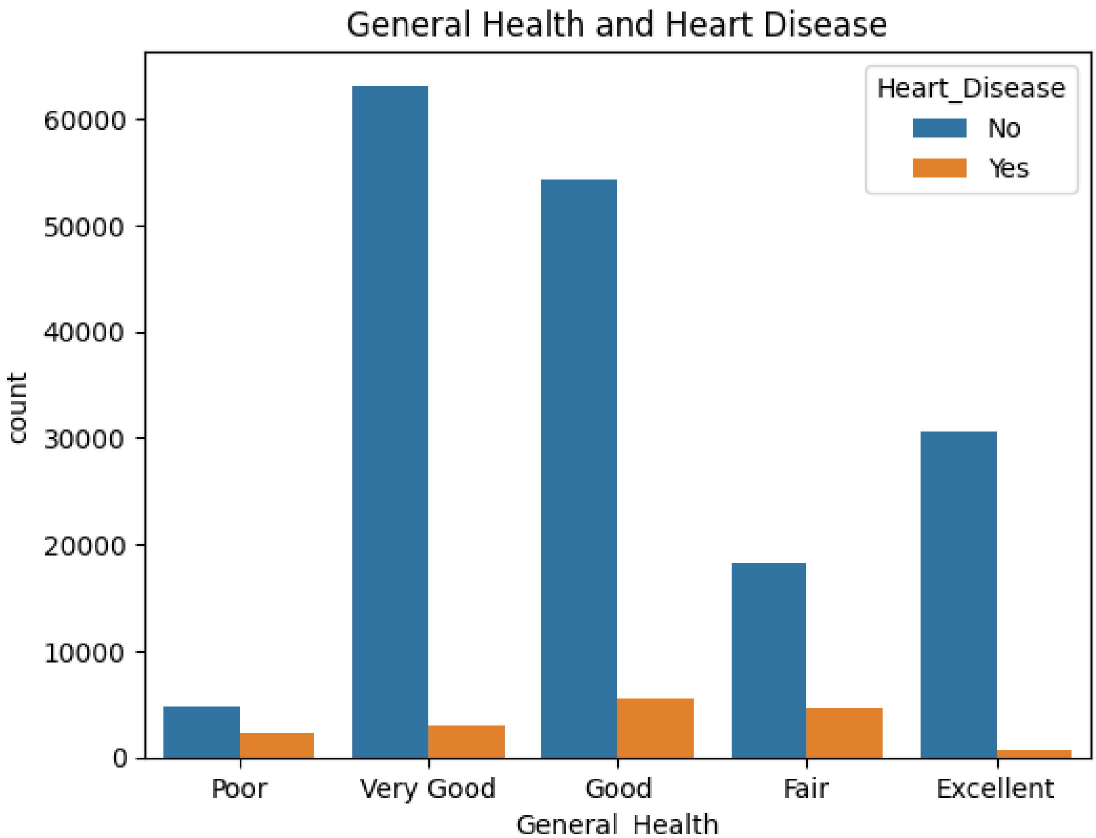


Visualizing the patient's demographics along with the heart disease, help us to know more about the relation of cardiovascular disease with patient. Firstly looking at the Gender graph, we can see that, males are more prone to heart disease as compared to females. The second graph reveals interesting facts about the data, where we can we that patientis with age higher than 55 years of age have increased instances of heart diseases, as compared to other age groups, with maximum heart disease cases in 80+ years of age patient. This means that patients older age are more prone to cardiovascular disease and the risk of cardivascular diases increases with age. The third graph, which is about BMI, shows that, patients with BMI between 25-30 i.e. overweight, have higher chances of heart disease.

General Health and Heart Disease

[25]:
```
sns.countplot(x = 'General_Health', data = df, hue  = 'Heart_Disease').
↪set_title('General Health and Heart Disease')
```

[25]:
```
Text(0.5, 1.0, 'General Health and Heart Disease')
```

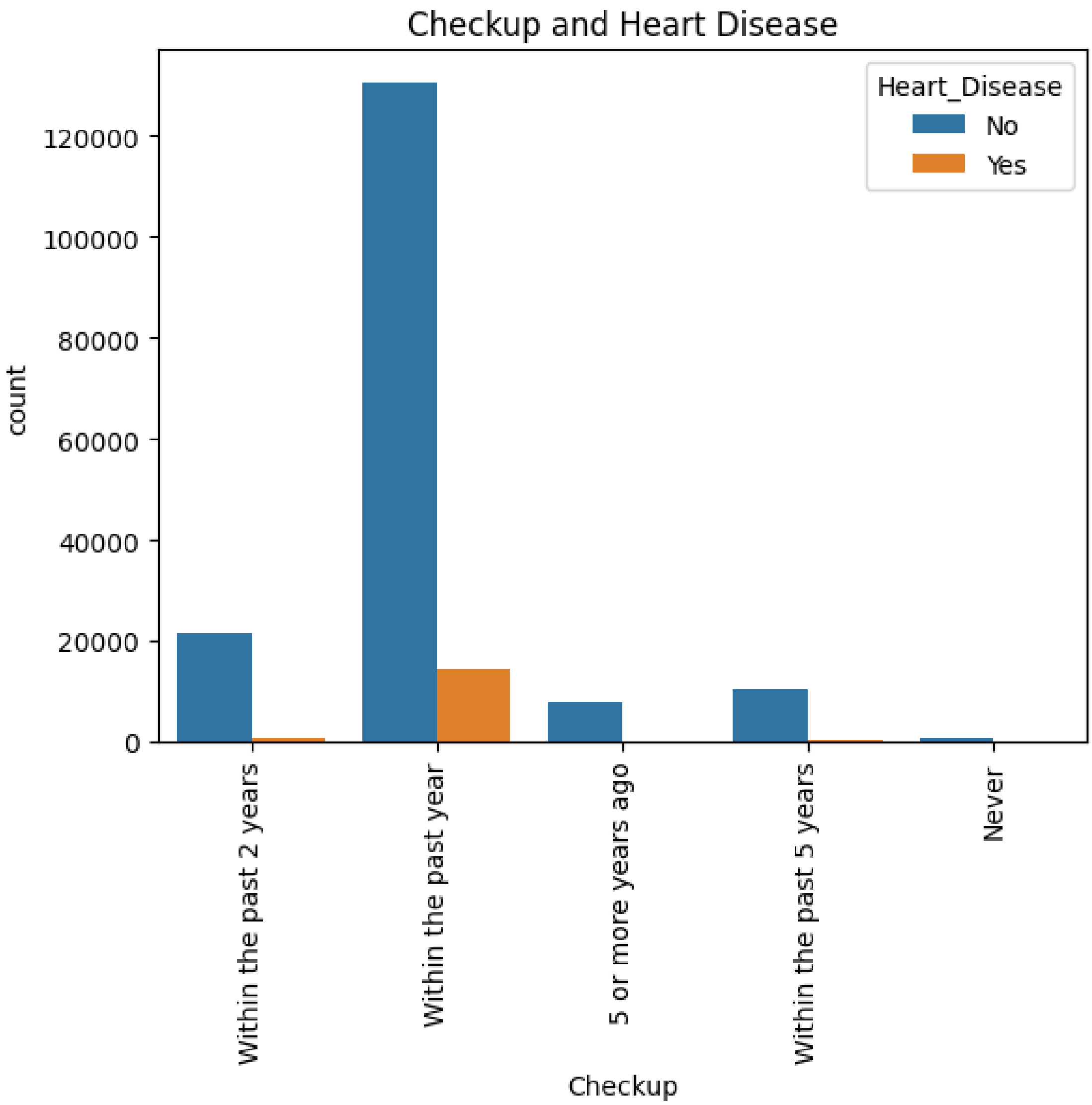


This graph is in contrast to my belief that, healthy patients are less prone to heart disease. However, this graph shows that patients with very good ot good general health have more chances of heart disease as compared to patients with poor general health.

Checkup and Heart Disease

[26]:
```
sns.countplot(x = 'Checkup', data = df, hue  = 'Heart_Disease').
 ↪set_title('Checkup and Heart Disease')
plt.xticks(rotation=90)
```

[26]:
```
([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')])
```



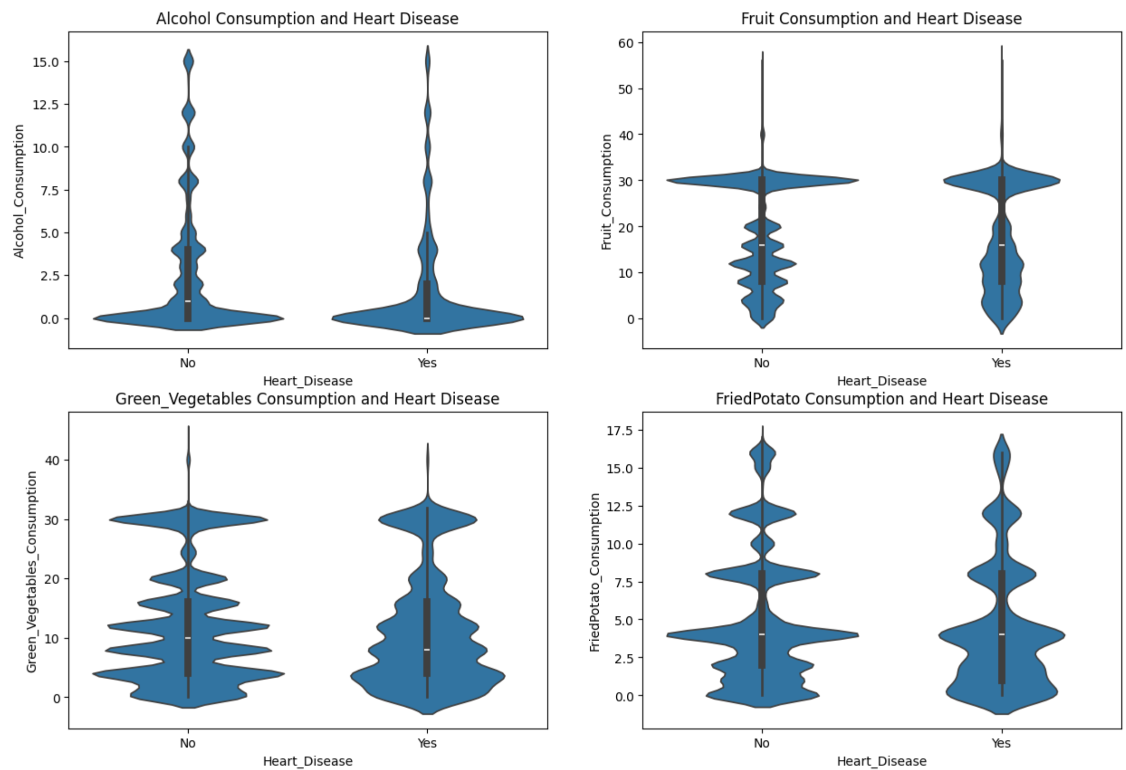
According to this graph, patients who have checkup in the last year have higher chances of having heart disease. This means that, patients who got themselves checked more often have higher chances of diagnosing cardiovascular disease at an early stage, as compared to patients who do not get themselves checked regularly.

Food Consumption and Heart Disease

[27]:
```
fig, ax = plt.subplots(2,2,figsize=(15, 10))
sns.violinplot(x = 'Heart_Disease', y = 'Alcohol_Consumption', data = df, ax =␣
 ↪ax[0,0]).set_title('Alcohol Consumption and Heart Disease')
sns.violinplot(x = 'Heart_Disease', y = 'Fruit_Consumption', data = df, ax =␣
 ↪ax[0,1]).set_title('Fruit Consumption and Heart Disease')
```

```
sns.violinplot(x = 'Heart_Disease', y = 'Green_Vegetables_Consumption', data =
↪df, ax = ax[1,0]).set_title('Green_Vegetables Consumption and Heart Disease')
sns.violinplot(x = 'Heart_Disease', y = 'FriedPotato_Consumption', data = df,
↪ax = ax[1,1]).set_title('FriedPotato Consumption and Heart Disease')
```

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



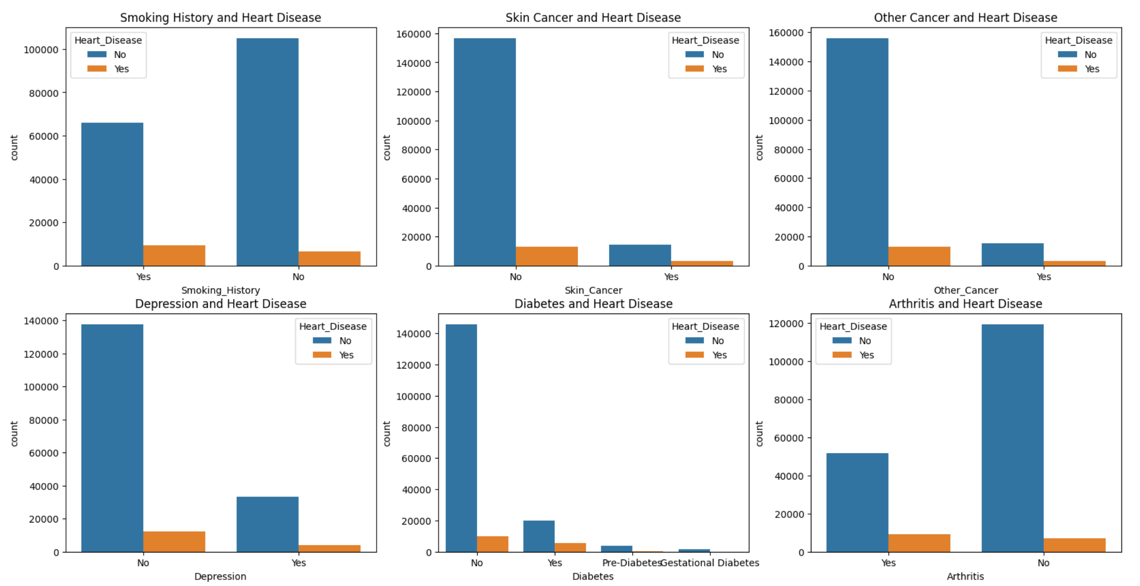
These graphs visualizes the patient's food and drinking habit along with their heart disease. Looking at the alcohol consumption graph, we can see that patients with increased alcohol consumption tend to have lower chances of heart disease. However, the patients with higher consumption on fruits and green vegetables, tend to have lower risk of heart diseases. In addition to that, patients with higher consumption of fried potatoes tend to have higher risk of heart disease.

Medical History and Heart Disease

[28]:
```
fig, ax = plt.subplots(2,3,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 = 'Skin_Cancer', data = df, ax = ax[0,1], hue =
↪'Heart_Disease').set_title('Skin Cancer and Heart Disease')
sns.countplot(x = 'Other_Cancer', data = df, ax = ax[0,2], hue =
↪'Heart_Disease').set_title('Other Cancer 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')
```

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



These graphs visualizes patient's medical history and its relation with heart disease. In the first graph, which is about smoking history, we can see that patients who smoke or used to smoke tend to have higher instances of having cardiovascular disease. In the second graph, we can see that patients with no skin cancer have higher cases of having heart disease as compared to its counterpart. In addtion to that it is evident from the third graph, that patient without any kind of cancer have higher cases of having a cardiovascular disease. In the fourth graph, we can see that patients with no depression have higher cases of having heart disease as compared to its counterpart. In the fifth graph, we can see that patients with no diabetes have higher cases of having heart disease and pre-diabetes or gestational diabetes have zero or no effect on heart diseases. In the last graph, we can see that patients with no arthritis have higher cases of having heart disease as compared to its counterpart.

From this, I conclude that, patients with medical history have no major effect on having a cardiovacular disease.

0.2 Data Preprocessing 2

Label Encoding the Categorical Variables

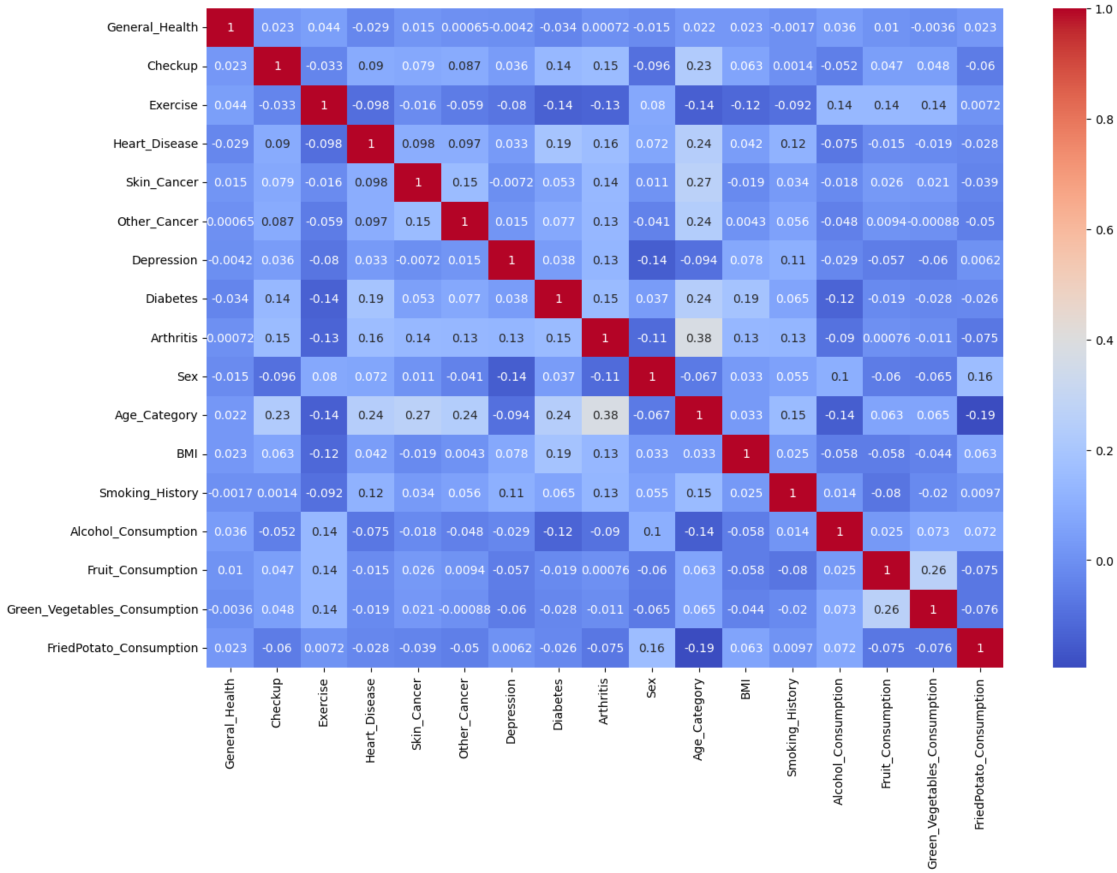
[29]:
```
from sklearn.preprocessing import LabelEncoder

# List of categorical variables
cols =␣
↪['General_Health','Checkup','Exercise','Heart_Disease','Skin_Cancer','Other_Cancer','Depres
↪'Smoking_History']

# Label encoding object
le = LabelEncoder()

for i in cols:
    le.fit(df[i])
    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]
```

0.3 Coorelation Matrix Heatmap

[30]:
```
plt.figure(figsize=(15,10))
sns.heatmap(df.corr(), annot = True, cmap = 'coolwarm')
```

[30]: <Axes: >



There is no major coorelation among the variales.

[32]:
```python
# Calculate the correlation matrix
correlation_matrix = df.corr()
correlation_long = correlation_matrix.reset_index().melt(id_vars='index')
correlation_long.columns = ['Variable 1', 'Variable 2', 'Correlation']
correlation_long.to_csv('C:\\Users\\akash\\Downloads\\correlation_matrix_long.
 ↪csv', index=False)
```

0.4 Train Test Split

[31]:
```python
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'], test_size = 0.2, random_state = 0)
```

0.5 Cardiovascular Disease Prediction

For predicting the cardiovascular disease, I have used the following classification models: 1. Random Forest Classifier 2. Decision Tree Classifier

0.5.1 Random Forest Classifier

[32]:
```python
from sklearn.ensemble import RandomForestClassifier

# Create Random Forest object
rfc = RandomForestClassifier(random_state=0, max_features='sqrt',␣
↪n_estimators=200, class_weight='balanced')
```

[33]:
```python
# Training the model
rfc.fit(X_train, y_train)
```

[33]:
```
RandomForestClassifier(class_weight='balanced', n_estimators=200,
                       random_state=0)
```

[34]:
```python
# Training accuracy
rfc.score(X_train, y_train)
```

[34]:
```
0.9999866150005688
```

[35]:
```python
# Predicting the test set results
rfc_pred = rfc.predict(X_test)
```

0.5.2 Decision Tree Classifier

[36]:
```python
from sklearn.tree import DecisionTreeClassifier

# Create Decision Tree object
dtc = DecisionTreeClassifier(random_state=0, max_depth= 12, min_samples_leaf=2,␣
↪min_samples_split=2, class_weight='balanced')
```

[37]:
```python
# Training the model
dtc.fit(X_train, y_train)
```

[37]:
```
DecisionTreeClassifier(class_weight='balanced', max_depth=12,
                       min_samples_leaf=2, random_state=0)
```

[38]:
```python
# Training accuracy
dtc.score(X_train, y_train)
```

[38]:
```
0.73877835110192
```

[39]:
```python
# Predicting the test set results
dtc_pred = dtc.predict(X_test)
```

0.5.3 Logistic Regression

[40]:
```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
```

[41]:
```
#Training the model
lr = LogisticRegression(max_iter=500)
lr.fit(X_train, y_train)
```

[41]: LogisticRegression(max_iter=500)

[42]:
```
#Training accuracy
lr.score(X_train, y_train)
```

[42]: 0.9141753836475462

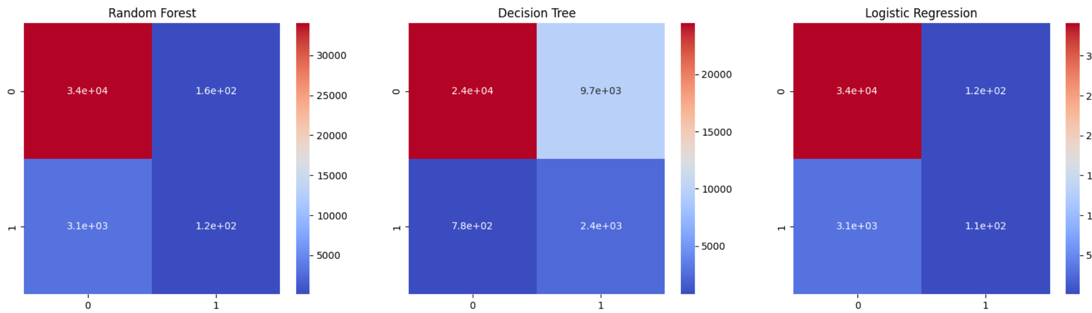
[43]:
```
#Predicting the test set results
lr_pred = lr.predict(X_test)
```

0.6 Model Evalution

0.6.1 Confusion Matrix

[44]:
```
from sklearn.metrics import confusion_matrix
fig, ax  = plt.subplots(1,3, figsize = (20,5))
sns.heatmap(confusion_matrix(y_test, rfc_pred), annot = True, cmap =␣
 ↪'coolwarm', ax = ax[0]).set_title('Random Forest')
sns.heatmap(confusion_matrix(y_test, dtc_pred), annot = True, cmap =␣
 ↪'coolwarm', ax = ax[1]).set_title('Decision Tree')
sns.heatmap(confusion_matrix(y_test, lr_pred), annot = True, cmap = 'coolwarm',␣
 ↪ax = ax[2]).set_title('Logistic Regression')
```

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



[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
```

[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
```

[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:  0.9146857265231824
Precision Score:  0.4732142857142857
Recall Score:  0.03338582677165354
F1 Score:  0.06237128567225655
```

CONCLUSION

From the exploratory data analysis, it was found the risk of having a cardiovascular disease increases with increasing age and the people with age above 55 are more prone to this disease, with maximum number patients with cardiovascular disease in 80+ years of age. In addition to that, the patients with higher BMI are more likely to have cardiovascular disease. The patients of older age who exercise are prone cardiovascular disease, which may be due to extensive pressure on the heart. The dietary habits of the patient also have some contribution to the cardiovascular disease. The patients who consume higher amount of fruits and green vegetables are less prone to cardiovascular disease. However, the patients who consume fried potatoes are more prone to cardiovascular disease. The patients who smoke or used to smoke are more prone to cardiovascular disease. But incontrast to my belief, any of the previous medical history such as cancer, arthritis, diabetes or depression have no major effect on cardiovascular disease.

```
[ ]:
```