```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("/content/medical_diagnosis_classification.csv")
df
```

	blood_sugar	bmi_dup	sex	age	blood_pressure	cholesterol	smoker	bmi
0	High	NaN	M	89.0	109.9	203.0	No	NaN
1	Lw	29.0	F	88.0	118.7	165.9	NO	29.0
2	HIGH	19.2	M	80.0	107.5	166.8	Yes	19.2
3	High	22.0	M	NaN	121.3	204.7	N	22.0
4	NORMAL	28.0	M	36.0	unknown	202.7	Yes	28.0
...	...	...	...	...	...	...	...	...
260	High	20.3	M	21.0	103.8	184.0	NO	20.3
261	HIGH	18.6	F	35.0	130.9	227.7	o	18.6
262	HIGH	18.6	F	35.0	130.9	227.7	o	18.6
263	Normal	26.6	M	50.0	110.6	253.5	NO	26.6
264	Normal	33.1	M	34.0	123.1	159.4	o	33.1

265 rows × 10 columns

Next steps: Generate code with df New interactive sheet

```
print(df['blood_sugar'].unique())
```

```
['High' 'Lw' 'HIGH' 'NORMAL' nan 'Normal' 'Norml' 'Low' 'ow' 'Lo' 'LOW'
 'Hig' 'Hih' 'ormal' 'Nomal' 'igh' 'Noral' 'Norma' 'Nrmal' 'Hgh']
```

```
import numpy as np

df['blood_sugar'] = (
    df['blood_sugar']
    .astype(str)
    .str.strip()
    .str.lower()
)
```

```
mapping = {
    'high': 'High', 'hig': 'High', 'hgh': 'High', 'hih': 'High', 'igh': 'High',
    'low': 'Low', 'lo': 'Low', 'lw': 'Low', 'ow': 'Low',
    'normal': 'Normal', 'norml': 'Normal', 'norma': 'Normal', 'noral': 'Normal'
    'nomal': 'Normal', 'nrmal': 'Normal', 'ormal': 'Normal'
}


df['blood_sugar'] = df['blood_sugar'].map(mapping)

df['blood_sugar'] = df['blood_sugar'].replace('nan', np.nan)


df = df.infer_objects(copy=False)

print(df['blood_sugar'].unique())
df['blood_sugar'] = df['blood_sugar'].replace(np.nan,df['blood_sugar'].mode()[0
df
```

```
['High' 'Low' 'Normal']
      blood_sugar  bmi_dup  sex   age  blood_pressure  cholesterol  smoker   bmi
  0          High      NaN    M  89.0           109.9        203.0      No   NaN
  1           Low     29.0    F  88.0           118.7        165.9      NO  29.0
  2          High     19.2    M  80.0           107.5        166.8     Yes  19.2
  3          High     22.0    M   NaN           121.3        204.7       N  22.0
  4        Normal     28.0    M  36.0         unknown        202.7     Yes  28.0
 ...          ...      ...  ...   ...             ...          ...     ...   ...
260          High     20.3    M  21.0           103.8        184.0      NO  20.3
261          High     18.6    F  35.0           130.9        227.7       o  18.6
262          High     18.6    F  35.0           130.9        227.7       o  18.6
263        Normal     26.6    M  50.0           110.6        253.5      NO  26.6
264        Normal     33.1    M  34.0           123.1        159.4       o  33.1

265 rows × 10 columns
```

Next steps: Generate code with df New interactive sheet

```
df= df.drop(columns=['bmi_dup'])
```

```
df
```

	blood_sugar	sex	age	blood_pressure	cholesterol	smoker	bmi	patient_i
0	High	M	89.0	109.9	203.0	No	NaN	P4000
1	Low	F	88.0	118.7	165.9	NO	29.0	Nal
2	High	M	80.0	107.5	166.8	Yes	19.2	P4000
3	High	M	NaN	121.3	204.7	N	22.0	P4000
4	Normal	M	36.0	unknown	202.7	Yes	28.0	P4000
...	...	...	...	...	...	...	...	.
260	High	M	21.0	103.8	184.0	NO	20.3	P4000
261	High	F	35.0	130.9	227.7	o	18.6	P4021
262	High	F	35.0	130.9	227.7	o	18.6	P4021
263	Normal	M	50.0	110.6	253.5	NO	26.6	P4004
264	Normal	M	34.0	123.1	159.4	o	33.1	P4018

265 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
print(df["bmi"].isnull().sum())
print(df["bmi"].unique())
```

```
26
[ nan 29.  19.2 22.  28.  21.9 20.3 17.2 18.9 34.2 33.  16.1 20.1 23.1
 35.6 34.6 37.3 29.1 23.8 30.2 15.4 25.2 20.7 30.4 39.9 19.4 28.7 18.
 38.4 22.3 37.  22.1 35.2 20.2 19.8 21.6 35.9 31.  25.1 39.3 26.6 39.2
 19.9 37.6 15.3 28.6 25.6 28.1 28.8 24.7 40.  39.4 39.  39.5 34.8 16.6
 38.8 24.1 18.8 23.2 29.6 22.9 30.3 31.8 18.2 27.5 31.3 36.  16.8 34.9
 24.9 37.9 20.8 34.5 30.6 23.9 22.4 24.4 28.5 33.5 25.  27.  23.6 31.2
 24.8 37.4 21.  32.3 39.1 23.7 35.8 23.  30.8 39.6 36.9 25.9 23.4 28.4
 22.2 24.5 27.1 31.7 37.1 15.5 23.5 37.8 26.7 34.1 25.8 15.8 18.5 39.7
 30.7 26.3 17.  18.4 28.3 20.4 27.8 29.3 33.1 18.7 26.8 31.1 38.1 32.7
 32.8 35.7 32.9 24.3 18.6 17.9 18.1 16.2 28.9 17.6 18.3 26.1 17.3 30.5
 19.1 16.9 31.6]
```

```
df["bmi"] = df["bmi"].replace(np.nan,df["bmi"].mean())
print(df["bmi"].unique())
print(df["bmi"].isnull().sum())
```

```
[27.11715481 29.         19.2        22.         28.         21.9
 20.3        17.2        18.9        34.2        33.         16.1
 20.1        23.1        35.6        34.6        37.3        29.1
 23.8        30.2        15.4        25.2        20.7        30.4
 39.9        19.4        28.7        18.         38.4        22.3
 37.         22.1        35.2        20.2        19.8        21.6
 35.9        31.         25.1        39.3        26.6        39.2
 19.9        37.6        15.3        28.6        25.6        28.1
 28.8        24.7        40.         39.4        39.         39.5
 34.8        16.6        38.8        24.1        18.8        23.2
 29.6        22.9        30.3        31.8        18.2        27.5
 31.3        36.         16.8        34.9        24.9        37.9
 20.8        34.5        30.6        23.9        22.4        24.4
 28.5        33.5        25.         27.         23.6        31.2
 24.8        37.4        21.         32.3        39.1        23.7
 35.8        23.         30.8        39.6        36.9        25.9
 23.4        28.4        22.2        24.5        27.1        31.7
 37.1        15.5        23.5        37.8        26.7        34.1
 25.8        15.8        18.5        39.7        30.7        26.3
 17.         18.4        28.3        20.4        27.8        29.3
 33.1        18.7        26.8        31.1        38.1        32.7
 32.8        35.7        32.9        24.3        18.6        17.9
 18.1        16.2        28.9        17.6        18.3        26.1
 17.3        30.5        19.1        16.9        31.6       ]
0
```

Double-click (or enter) to edit

```
print(df['sex'].unique())
print(df['sex'].isnull().sum())
df['sex'] = df['sex'].replace(np.nan,df['sex'].mode()[0])
print(df['sex'].unique())
print(df['sex'].isnull().sum())
```

```
['M' 'F' nan]
19
['M' 'F']
0
```

```
print(df['age'].unique())
print(df['age'].isnull().sum())
```

```
[89. 88. 80. nan 36.  3. 21. 77. 71. 97. 93. 85. 79. 37. 23. 99. 98.  5.
 67. 81. 25. 90. 40. 74. 82. 52. 14. 55. 30. 56. 50. 57. 72. 86.  9. 62.
 18. 20. 33. 17. 65. 64. 42. 47. 68. 75. 54. 15. 76.  6. 51. 39. 35. 34.
 87. 38. 19. 48. 95. 58. 32. 96. 44. 94. 16. 11. 22. 12. 27. 43. 45.  2.
 24. 60. 66. 26. 53.  8. 41. 70. 61. 92. 69. 10.  0. 13. 63. 78.]
20
```

```
Start coding or generate with AI.
```

```
df['age'] = df['age'].replace(np.nan,df['age'].mean())
print(df['age'].unique())
print(df['age'].isnull().sum())
```

```
[89.         88.         80.         50.9755102 36.          3.
 21.         77.         71.         97.         93.         85.
 79.         37.         23.         99.         98.          5.
 67.         81.         25.         90.         40.         74.
 82.         52.         14.         55.         30.         56.
 50.         57.         72.         86.          9.         62.
 18.         20.         33.         17.         65.         64.
 42.         47.         68.         75.         54.         15.
 76.          6.         51.         39.         35.         34.
 87.         38.         19.         48.         95.         58.
 32.         96.         44.         94.         16.         11.
 22.         12.         27.         43.         45.          2.
 24.         60.         66.         26.         53.          8.
 41.         70.         61.         92.         69.         10.
  0.         13.         63.         78.       ]
0
```

```
df['age'] = df['age'].astype(int)
```

```
print(df['blood_pressure'].unique())
print(df['blood_pressure'].isnull().sum())
df['blood_pressure'] = df['blood_pressure'].replace(r'^(?![\d\.\-]+$).*', np.na
df['blood_pressure'] = pd.to_numeric(df['blood_pressure'], errors='coerce')
df['blood_pressure'] = df['blood_pressure'].replace(np.nan,df['blood_pressure']
```

```
[  109.9          118.7          107.5          121.3
   178.41452282   103.8          122.5          119.1
    92.3          103.1          127.1           99.7
   118.9           88.2          118.5           89.2
   110.9          121.8          116.2          101.5
   101.7           95.8          123.1          122.4
   130.           112.5           94.           115.3
   120.2          102.5          104.           126.8
   121.5          137.           145.8           97.5
   113.7          121.9          125.5          135.2
   130.9          137.1          120.9          110.6
   149.2          122.3          116.4          103.9
   135.1          118.           123.7          116.1
   114.1          112.3          103.4          108.4
   108.            96.9          134.9          125.4
   139.6          130.8        13760.           134.6
   103.2          133.5          113.           135.
   114.9          112.           113.5          105.4
    97.8           82.5          104.2          159.5
   148.           115.2          145.7          105.7
   129.5          138.8           98.6          127.
    99.6          138.           123.5          129.4
   116.7           93.3          113.4          133.9
   149.9          124.2          109.7          128.
   132.6          122.           121.1          123.2
   120.7          124.5          147.4          106.
   137.7          117.9          128.4          112.1
   139.9          112.6           96.           140.5
   111.3          125.3          130.1         1080.
    91.4          116.5          119.2          111.9
   107.6          118.6          111.6          124.4
   136.6          137.2          139.3          115.
   101.           110.7           94.5           99.9
   121.2          127.2          102.1           94.4
   126.3          110.8          114.2          105.3
   111.1          132.3          147.5          122.1
    90.2          128.5          103.6          108.6
    98.8          103.5          147.           127.8
    83.6          158.1           92.8          112.2
    81.6          116.           113.1          124.6
   116.3          128.1          101.4          110.5
   114.4          124.1          112.4          118.4
   102.7          136.4          136.1          129.9
   127.3          137.8          106.1          105.2
   140.2          102.3          110.2          113.8
   122.6          142.4          109.2          119.3
    97.9           97.6          107.7       ]
0
```

```
print(df['cholesterol'].unique())
print(df['cholesterol'].isnull().sum())
df['cholesterol'] = df['cholesterol'].replace(r'^(?![\d\.\-]+$).*', np.nan, reg
df['cholesterol'] = df['cholesterol'].replace(np.nan,df['cholesterol'].mean())
```

```
[203.         165.9        166.8        204.7        202.7
 217.2        184.         170.4        203.8        249.7
 134.3        172.         220.7        207.6        221.2
 185.         143.6        258.5        237.9        199.2
 170.5        182.8        208.         191.2        199.5
 256.4        232.3        159.         181.8        201.11709402
 197.8        303.9        215.         195.3        264.5
 207.2        146.8        195.6        221.5        217.
 235.2        202.9        150.9        253.5        165.7
 292.         258.8        184.3        134.2        174.9
 255.3        206.2        152.8        227.4        192.7
 177.3        180.         245.7        219.1        148.
 212.9        187.1        169.7        176.8        210.1
 190.3        172.5        224.8        246.         219.6
 182.4        219.7        120.6        106.7        188.5
 222.2        197.         226.1        157.9        230.9
 257.         175.6        227.         200.8        215.5
 256.9        211.5        226.8        155.8        225.7
 167.7        188.7        224.         215.7        152.4
 205.8        188.4        247.5        200.2        234.4
 158.9        220.6        237.2        191.5        182.6
 154.4        169.6        184.2        165.8        173.1
 205.5        209.7        238.2        173.3        185.4
 182.9        188.9        208.8        140.3        236.9
 193.4        224.7        169.2        177.         217.5
 164.7        142.7        205.6        176.5        194.2
 161.7        181.2        202.2        213.7        136.8
 213.8        139.6        192.5        227.3        159.1
 203.7        128.7        260.4        204.3        145.7
 171.6        284.2        236.6        192.4        163.1
 248.1        212.1        229.4        159.4        264.6
 196.9        265.1        175.2        235.8        186.6
 219.8        235.5        207.9        144.1        161.4
 239.7        216.2        189.3        228.7        232.
 227.7        189.         234.5        213.5        166.
 130.4        235.1        308.5        242.3        205.7
 181.9        167.6        205.3        233.9        176.4
 151.6        202.5        180.1        245.1        256.7
 186.4        142.4        211.1        209.9        218.7
 248.4        229.2        218.6       ]
0
```

```
print(df['smoker'].unique())
smokemap = {
    'No':'No','NO':'No','o':'No','N':'No',
    'Yes':'Yes','YES':'Yes','es':'Yes','Ys':'Yes','Ye':'Yes'
}
df['smoker'] = df['smoker'].map(smokemap)
print(df['smoker'].unique())
print(df['smoker'].isnull().sum())
df['smoker'] = df['smoker'].replace(np.nan,df['smoker'].mode()[0])
print(df['smoker'].unique())
print(df['smoker'].isnull().sum())
```

```
['No' 'Yes']
['No' 'Yes']
0
['No' 'Yes']
0
```

```
df
```

	blood_sugar	sex	age	blood_pressure	cholesterol	smoker	bmi
0	High	M	89.00000	109.900000	203.0	No	27.117155
1	Low	F	88.00000	118.700000	165.9	No	29.000000
2	High	M	80.00000	107.500000	166.8	Yes	19.200000
3	High	M	50.97551	121.300000	204.7	No	22.000000
4	Normal	M	36.00000	178.414523	202.7	Yes	28.000000
...	...	...	...	...	...	...	...
260	High	M	21.00000	103.800000	184.0	No	20.300000
261	High	F	35.00000	130.900000	227.7	No	18.600000
262	High	F	35.00000	130.900000	227.7	No	18.600000
263	Normal	M	50.00000	110.600000	253.5	No	26.600000
264	Normal	M	34.00000	123.100000	159.4	No	33.100000

265 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
print(df['patient_id'].isnull().sum())
df = df.dropna(subset=['patient_id'])
print(df['patient_id'].isnull().sum())
```

```
0
0
```

```
df
```

	blood_sugar	sex	age	blood_pressure	cholesterol	smoker	bmi
0	High	M	89.00000	109.900000	203.0	No	27.117155
2	High	M	80.00000	107.500000	166.8	Yes	19.200000
3	High	M	50.97551	121.300000	204.7	No	22.000000
4	Normal	M	36.00000	178.414523	202.7	Yes	28.000000
5	Normal	F	3.00000	178.414523	217.2	No	21.900000
...	...	...	...	...	...	...	...
260	High	M	21.00000	103.800000	184.0	No	20.300000
261	High	F	35.00000	130.900000	227.7	No	18.600000
262	High	F	35.00000	130.900000	227.7	No	18.600000
263	Normal	M	50.00000	110.600000	253.5	No	26.600000
264	Normal	M	34.00000	123.100000	159.4	No	33.100000

243 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
print(df['disease'].unique())
print(df['disease'].isnull().sum())
df['disease'] = df['disease'].replace(np.nan,df['disease'].mode()[0])
print(df['disease'].unique())
print(df['disease'].isnull().sum())
```

```
[1. 0.]
0
[1. 0.]
0
<class 'pandas.core.series.Series'>
```

```
df.to_csv("final_cleaned_dataset.csv",index=False)
```

```
df2 = pd.read_csv("/content/cleaned_dataset")
print(df2.isnull().sum())
df2.columns
```

```
blood_sugar       0
sex               0
age               0
blood_pressure    0
cholesterol       0
smoker            0
```

```
bmi               0
patient_id        0
disease           0
dtype: int64
Index(['blood_sugar', 'sex', 'age', 'blood_pressure', 'cholesterol', 'smoker',
       'bmi', 'patient_id', 'disease'],
      dtype='object')
```

Charts

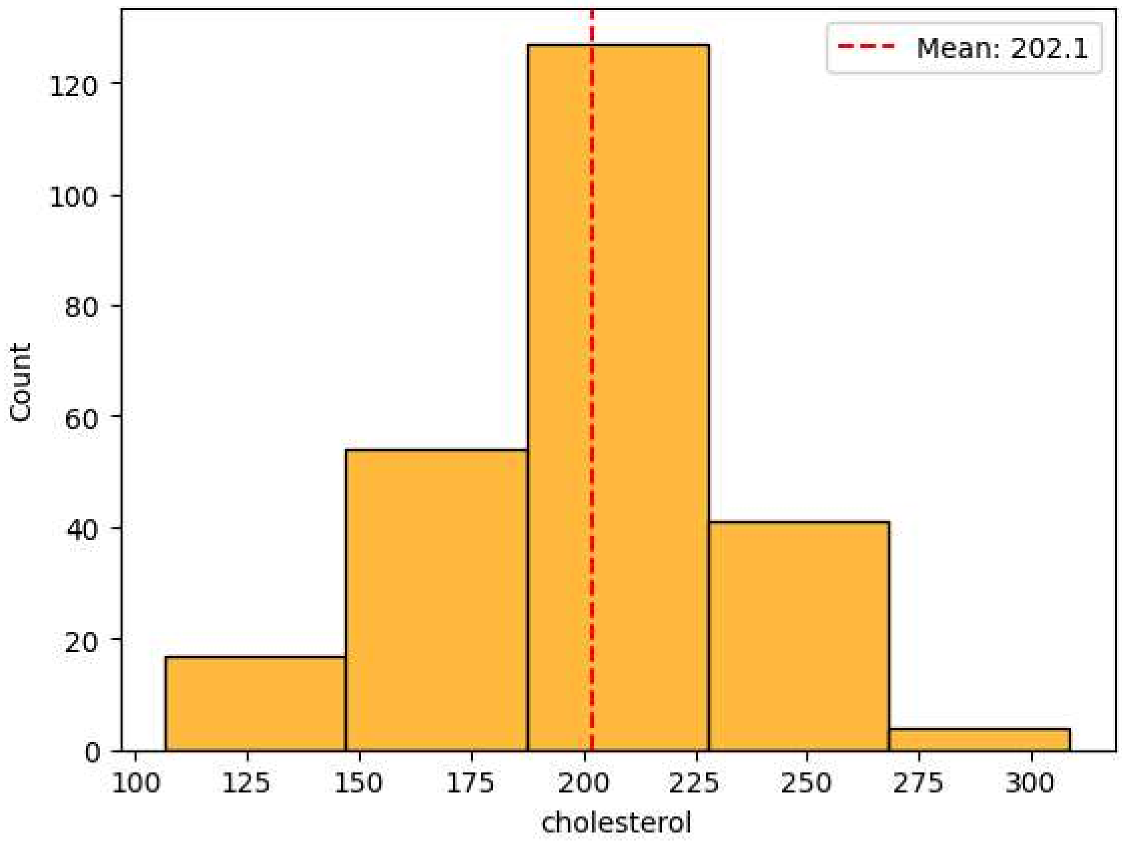
```
cleaned_df = pd.read_csv("/content/final_cleaned_dataset.csv")
cleaned_df
```

	blood_sugar	sex	age	blood_pressure	cholesterol	smoker	bmi	patie
0	High	M	89	109.900000	203.0	No	27.117155	F
1	High	M	80	107.500000	166.8	Yes	19.200000	F
2	High	M	50	121.300000	204.7	No	22.000000	F
3	Normal	M	36	178.414523	202.7	Yes	28.000000	F
4	Normal	F	3	178.414523	217.2	No	21.900000	F
...	...	...	...	...	...	...	...	
238	High	M	21	103.800000	184.0	No	20.300000	F
239	High	F	35	130.900000	227.7	No	18.600000	F
240	High	F	35	130.900000	227.7	No	18.600000	F
241	Normal	M	50	110.600000	253.5	No	26.600000	F
242	Normal	M	34	123.100000	159.4	No	33.100000	F

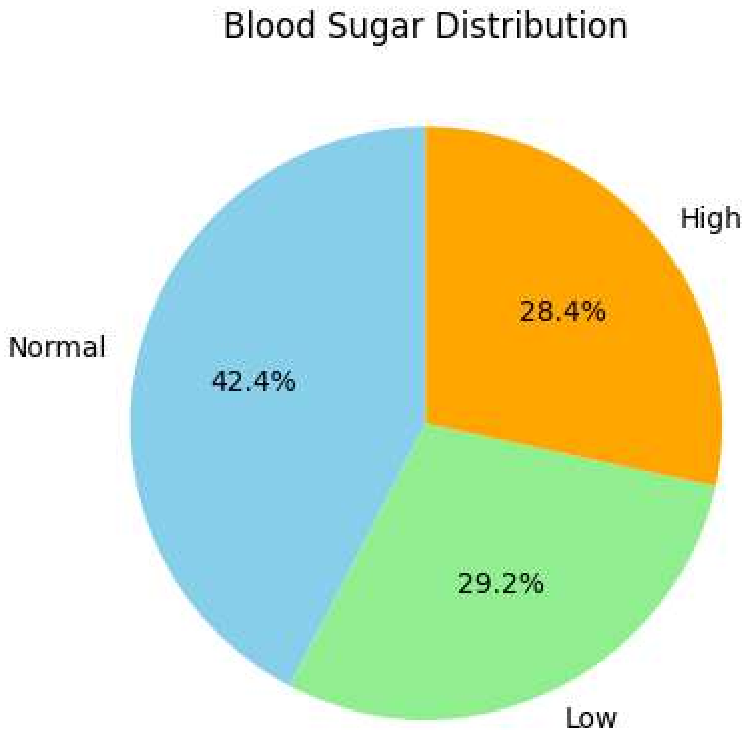
243 rows × 9 columns

Next steps: Generate code with cleaned_df New interactive sheet

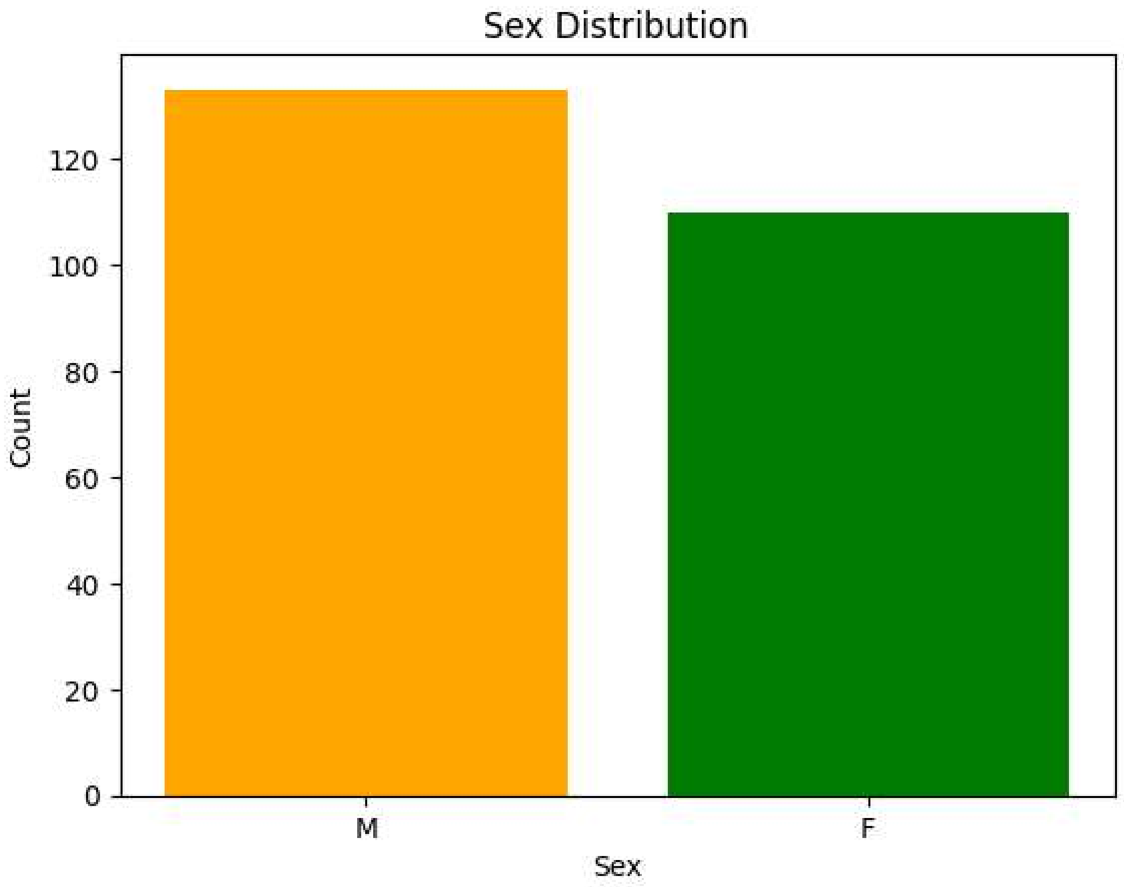
```
mean_val = cleaned_df['cholesterol'].mean()
sns.histplot(cleaned_df['cholesterol'], bins=5, color='orange')
plt.axvline(mean_val, color='red', linestyle='--', label=f'Mean: {mean_val:.1f}
plt.legend()
plt.show()
```



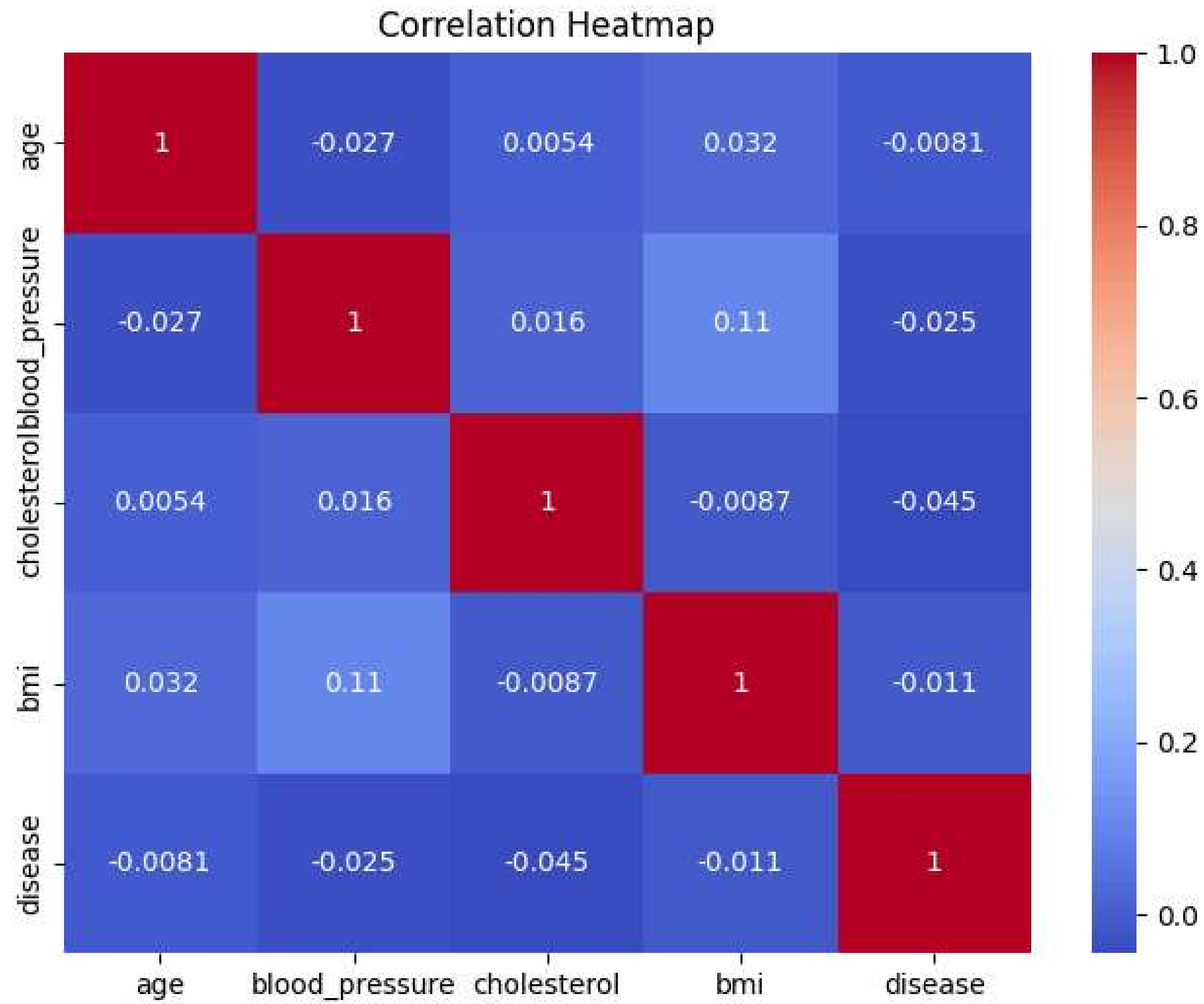
```
counts = cleaned_df['blood_sugar'].value_counts()
plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=90, colors=[
plt.title("Blood Sugar Distribution")
plt.show()
```



```
sex_count = cleaned_df['sex'].value_counts()
plt.bar(sex_count.index,sex_count.values,color=['orange','green'])
plt.xlabel("Sex")
plt.ylabel("Count")
plt.title("Sex Distribution")
plt.show()
```

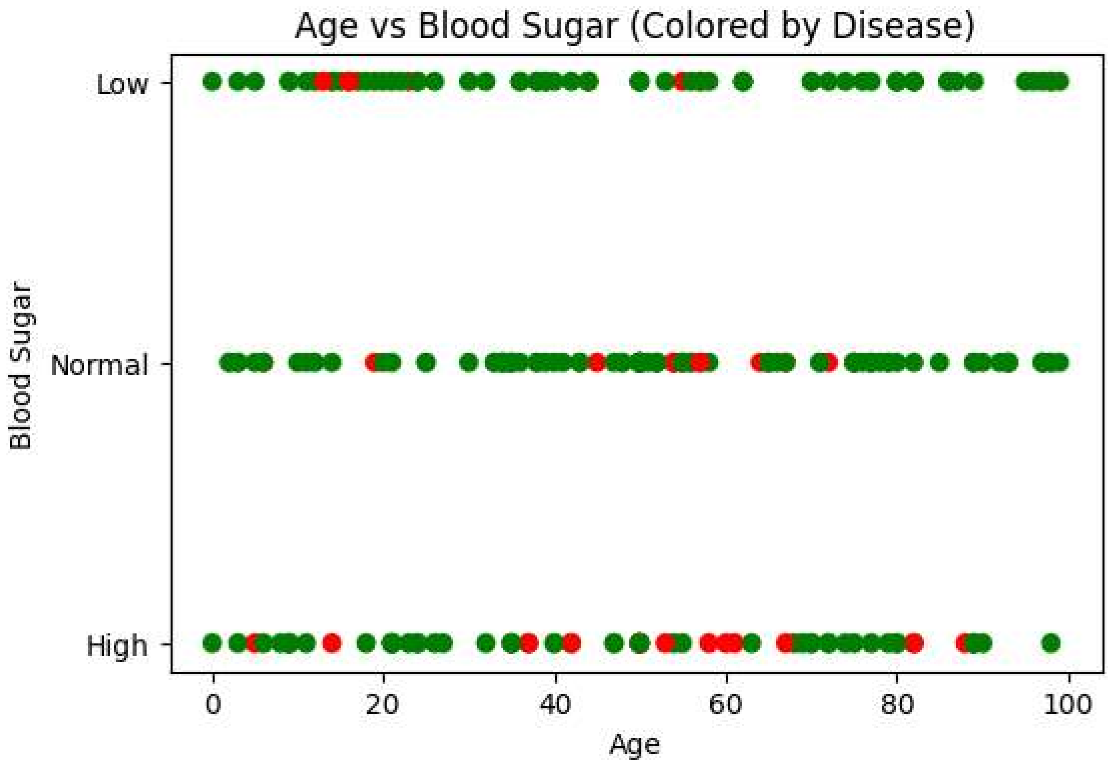


```
numeric_cols = ['age','blood_pressure','cholesterol','bmi','disease']
corr = cleaned_df[numeric_cols].corr()
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



```
colors = cleaned_df['disease'].map({0: 'green', 1: 'red'})

plt.figure(figsize=(6,4))
plt.scatter(cleaned_df['age'], cleaned_df['blood_sugar'], c=colors)
plt.xlabel('Age')
plt.ylabel('Blood Sugar')
plt.title('Age vs Blood Sugar (Colored by Disease)')
plt.show()
```



```
import matplotlib.pyplot as plt

df_sorted = df.sort_values('age')

plt.figure(figsize=(6,4))
plt.plot(df_sorted['age'], df['bmi'], marker='o', linestyle='-')
plt.xlabel('Age')
plt.ylabel('Blood Sugar')
plt.title('Blood Sugar Trend by Age')
plt.show()
```

Blood Sugar Trend by Age