```
import pandas as pd
import numpy as np
```

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

	blood_sugar	bmi_dup	sex	age	blood_pressure	cholesterol	smoker	bmi	patient_id	disease
0	High	NaN	M	89.0	109.9	203.0	No	NaN	P40000	1.0
1	Lw	29.0	F	88.0	118.7	165.9	NO	29.0	NaN	0.0
2	HIGH	19.2	M	80.0	107.5	166.8	Yes	19.2	P40002	0.0
3	High	22.0	M	NaN	121.3	204.7	N	22.0	P40003	0.0
4	NORMAL	28.0	M	36.0	unknown	202.7	Yes	28.0	P40004	0.0
...	...	...	...	...	...	...	...	...	...	...
260	High	20.3	M	21.0	103.8	184.0	NO	20.3	P40006	NaN
261	HIGH	18.6	F	35.0	130.9	227.7	o	18.6	P40214	0.0
262	HIGH	18.6	F	35.0	130.9	227.7	o	18.6	P40214	0.0
263	Normal	26.6	M	50.0	110.6	253.5	NO	26.6	P40045	0.0
264	Normal	33.1	M	34.0	123.1	159.4	o	33.1	P40188	0.0

265 rows × 10 columns

```
print(df['blood_sugar'].unique())
df['blood_sugar'] = df['blood_sugar'].str.strip().str.lower().replace({ 'high': 'High', 'hig': 'High', 'hgh': 'High', 'hih': 'H
    'low': 'Low', 'lo': 'Low', 'lw': 'Low', 'ow': 'Low',
    'normal': 'Normal', 'norml': 'Normal', 'norma': 'Normal', 'noral': 'Normal',
    'nomal': 'Normal', 'nrmal': 'Normal', 'ormal': 'Normal'})
```

```
['High' 'Low' 'Normal' 'norml' 'ow' 'lo' 'hig' 'hih' 'ormal' 'nomal' 'igh'
 'noral' 'norma' 'nrmal' 'hgh']
```

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

```
df['smoker'] = df['smoker'].str.strip().str.lower().replace({'yes': 'Yes', 'no': 'No', 'o': 'No', 'es': 'Yes', 'ys': 'Yes','n':
print(df['smoker'].unique())
```

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

```
df['sex'] = df['sex'].str.strip().str.upper().replace({'F': 'F', 'M': 'M'})
```

```
#handling missing values
df.isnull().sum()
```

```
                 0
   blood_sugar   0
           sex   0
           age   0
blood_pressure   0
   cholesterol   0
        smoker   0
           bmi   0
    patient_id   0
       disease   0

dtype: int64
```

```
df['age'] = df['age'].fillna(df['age'].mean())
df['bmi'] = df['bmi'].fillna(df['bmi'].mean())
df['blood_pressure'] = df['blood_pressure'].fillna(df['blood_pressure'].mean())
```

```
df['blood_sugar'] = df['blood_sugar'].fillna(df['blood_sugar'].mode()[0])
df['smoker'] = df['smoker'].fillna(df['smoker'].mode()[0])
df['sex'] = df['sex'].fillna(df['sex'].mode()[0])
```

```
df.isnull().sum()
```

```
                 0
   blood_sugar   0
       bmi_dup  26
           sex   0
           age   0
blood_pressure   0
   cholesterol   0
        smoker   0
           bmi   0
    patient_id  22
       disease  28

dtype: int64
```

```
df['blood_pressure'] = pd.to_numeric(df['blood_pressure'], errors='coerce')
df['cholesterol'] = pd.to_numeric(df['cholesterol'], errors='coerce')
df['age'] = pd.to_numeric(df['age'], errors='coerce')
df['blood_pressure'] = pd.to_numeric(df['blood_pressure'], errors='coerce')
df['bmi'] = pd.to_numeric(df['bmi'], errors='coerce')
```

```
df = df.drop_duplicates(subset=['patient_id'], keep='first')
df = df.drop(columns=['bmi_dup'])
```

```
df['disease'] = df['disease'].fillna(df['disease'].mode()[0])
```

```
/tmp/ipython-input-3759652690.py:1: SettingWithCopyWarning:
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/stable/user_guide/indexing.html#returning-a-view-ver
  df['disease'] = df['disease'].fillna(df['disease'].mode()[0])
```

```
df = df.dropna(subset=['patient_id'])
df['cholesterol'] = df['cholesterol'].fillna(df['cholesterol'].mean())
```

```
chol_mean = df['cholesterol'].mean()

# Replace NaN values with the mean
df['cholesterol'] = df['cholesterol'].fillna(chol_mean)
```

```
df
```

	blood_sugar	sex	age	blood_pressure	cholesterol	smoker	bmi	patient_id	disease
0	High	M	89.0	109.9	203.0	No	26.1	P40000	1.0
2	High	M	80.0	107.5	166.8	Yes	19.2	P40002	0.0
3	High	M	51.0	121.3	204.7	No	22.0	P40003	0.0
4	Normal	M	36.0	118.5	202.7	Yes	28.0	P40004	0.0
5	Normal	F	3.0	118.5	217.2	No	21.9	P40005	0.0
...	...	...	...	...	...	...	...	...	...
245	Low	F	51.0	97.9	203.7	No	17.3	P40245	0.0
246	Normal	M	78.0	97.6	218.7	No	30.5	P40246	0.0
247	Low	F	80.0	130.9	248.4	No	19.1	P40247	0.0
248	High	M	51.0	107.7	229.2	No	16.9	P40248	1.0
249	Low	M	70.0	127.0	218.6	No	31.6	P40249	0.0

228 rows × 9 columns

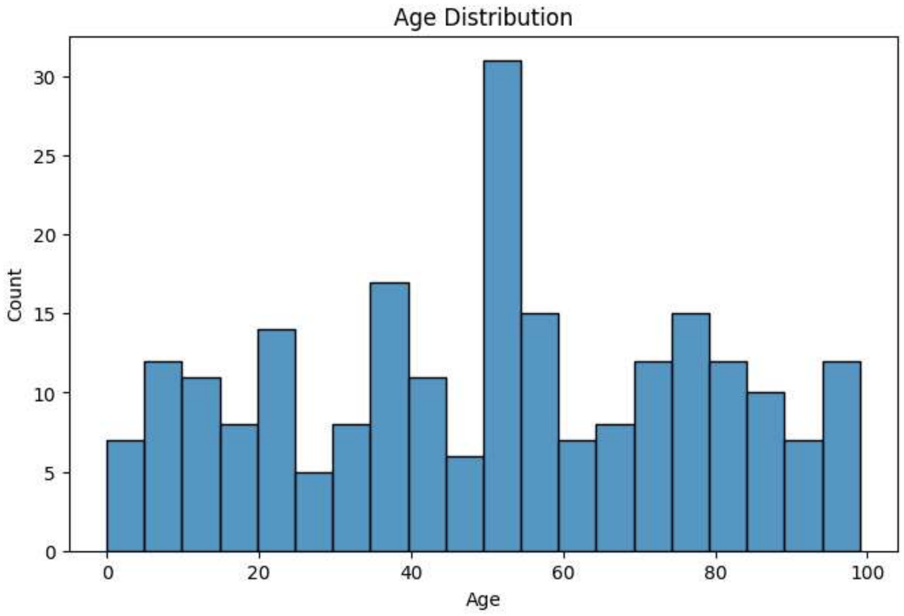
```
df2=pd.read_csv('/content/final_cleaned_dataset.csv')
df2
```

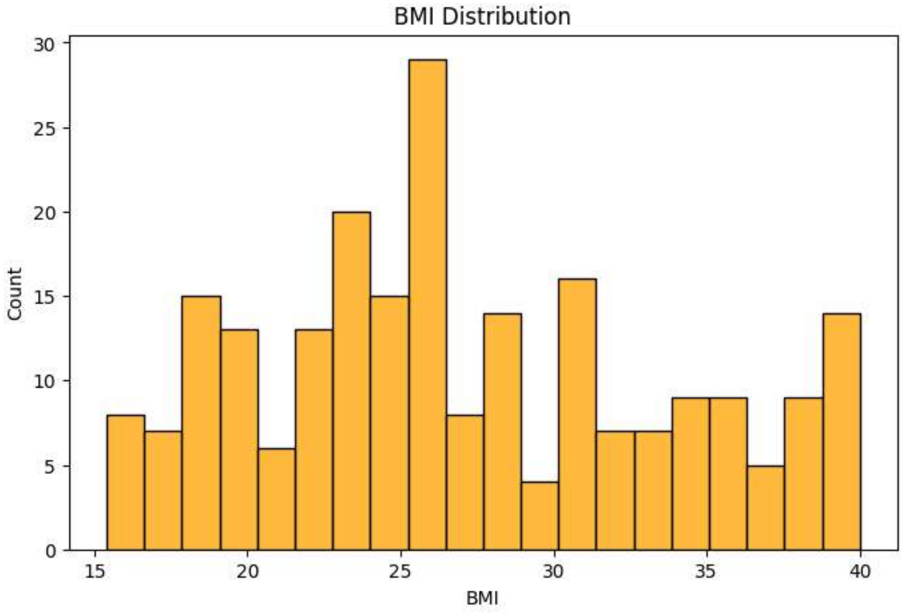
	blood_sugar	sex	age	blood_pressure	cholesterol	smoker	bmi	patient_id	disease
0	High	M	89	109.900000	203.0	No	27.117155	P40000	1.0
1	High	M	80	107.500000	166.8	Yes	19.200000	P40002	0.0
2	High	M	50	121.300000	204.7	No	22.000000	P40003	0.0
3	Normal	M	36	178.414523	202.7	Yes	28.000000	P40004	0.0
4	Normal	F	3	178.414523	217.2	No	21.900000	P40005	0.0
...	...	...	...	...	...	...	...	...	...
238	High	M	21	103.800000	184.0	No	20.300000	P40006	0.0
239	High	F	35	130.900000	227.7	No	18.600000	P40214	0.0
240	High	F	35	130.900000	227.7	No	18.600000	P40214	0.0
241	Normal	M	50	110.600000	253.5	No	26.600000	P40045	0.0
242	Normal	M	34	123.100000	159.4	No	33.100000	P40188	0.0

243 rows × 9 columns

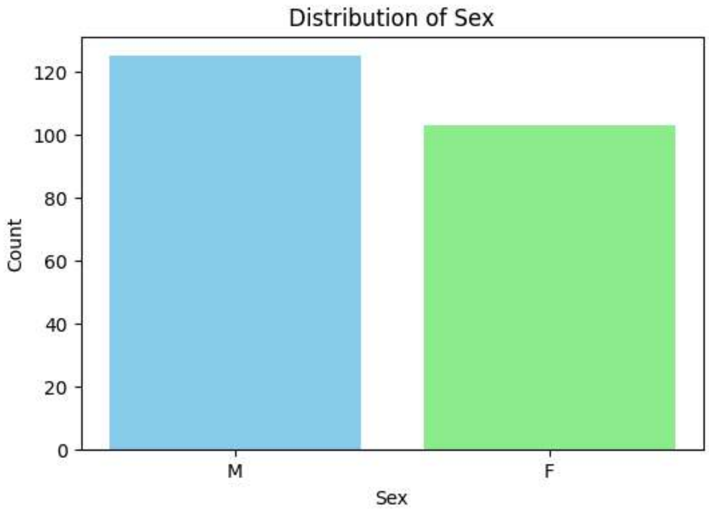
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(8,5))
sns.histplot(df['age'], bins=20)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
plt.figure(figsize=(8,5))
sns.histplot(df['bmi'], bins=20,  color='orange')
plt.title("BMI Distribution")
plt.xlabel("BMI")
plt.ylabel("Count")
plt.show()
```

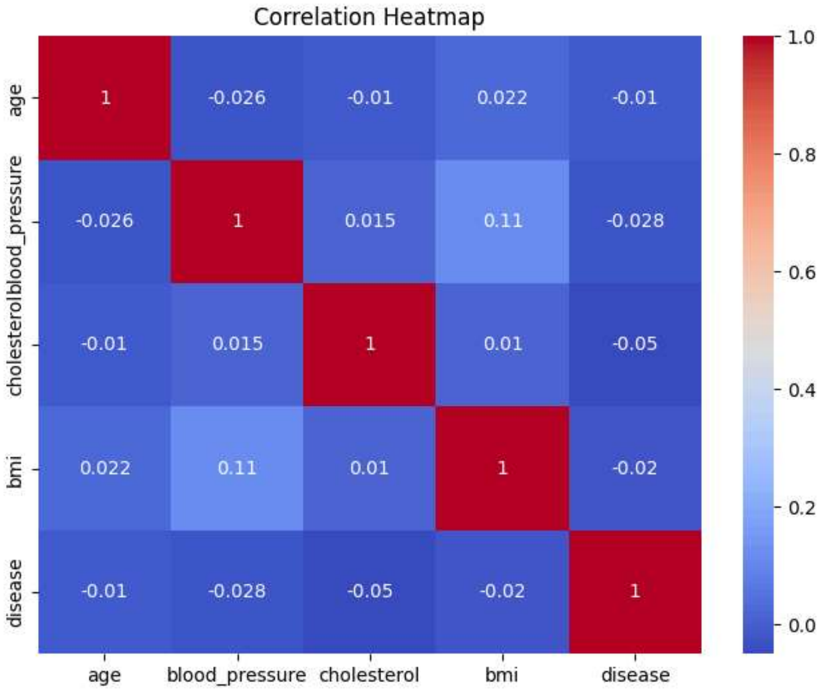




```
sex_counts = df['sex'].value_counts()

plt.figure(figsize=(6,4))
plt.bar(sex_counts.index, sex_counts.values, color=['skyblue','lightgreen'])
plt.xlabel('Sex')
plt.ylabel('Count')
plt.title('Distribution of Sex')
plt.show()
```

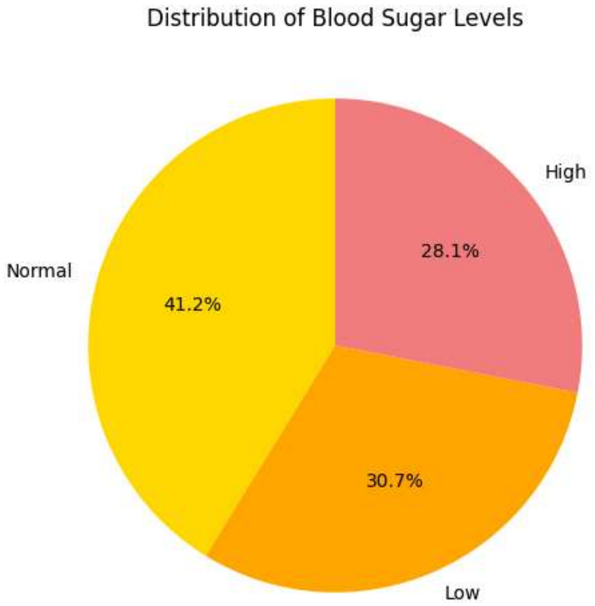


```
numeric_cols = ['age','blood_pressure','cholesterol','bmi','disease']
corr = df[numeric_cols].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

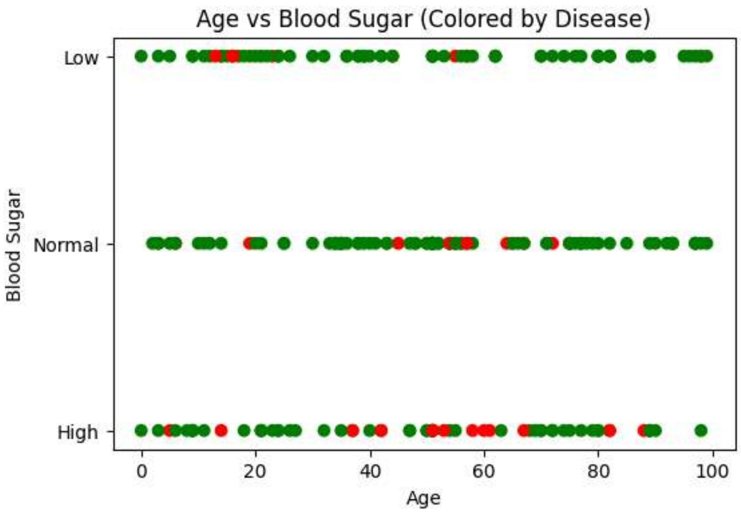


```
Start coding or generate with AI.
```

```
sugar_counts = df['blood_sugar'].value_counts()

plt.figure(figsize=(6,6))
plt.pie(sugar_counts, labels=sugar_counts.index, autopct='%1.1f%%', startangle=90, colors=['gold','orange','lightcoral'])
plt.title("Distribution of Blood Sugar Levels")
plt.show()
```



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

plt.figure(figsize=(6,4))
plt.scatter(df['age'], 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
40