**Introduction :**

Heart disease is a major global health concern. Python's data analysis and machine learning capabilities offer a robust approach to understanding this complex issue. By utilizing libraries like Pandas, NumPy, Matplotlib, and Seaborn, we can explore, visualize, and model heart disease data. This analysis can lead to early detection, improved prevention, and more effective treatments.

**Essential Python Libraries for Data Analysis**

**Import Statements:**

import pandas as pd (Data manipulation)

import numpy as np ( Numerical operations)

import matplotlib.pyplot as plt (Plotting)

import seaborn as sns ( Statistical visualizations)

To effectively analyze data, Python relies on several key libraries:

* **Pandas:** Handles data manipulation and analysis, making it easy to work with large datasets.
* **NumPy:** Provides support for numerical computations, essential for mathematical operations on data.
* **Matplotlib:** Creates a variety of visualizations to explore data patterns.
* **Seaborn:** Builds on Matplotlib, offering high-level statistical graphics for deeper insights.

**Loads heart disease data**

Imports a CSV file containing heart disease data into a Pandas DataFrame named data. This DataFrame can be used for further analysis.

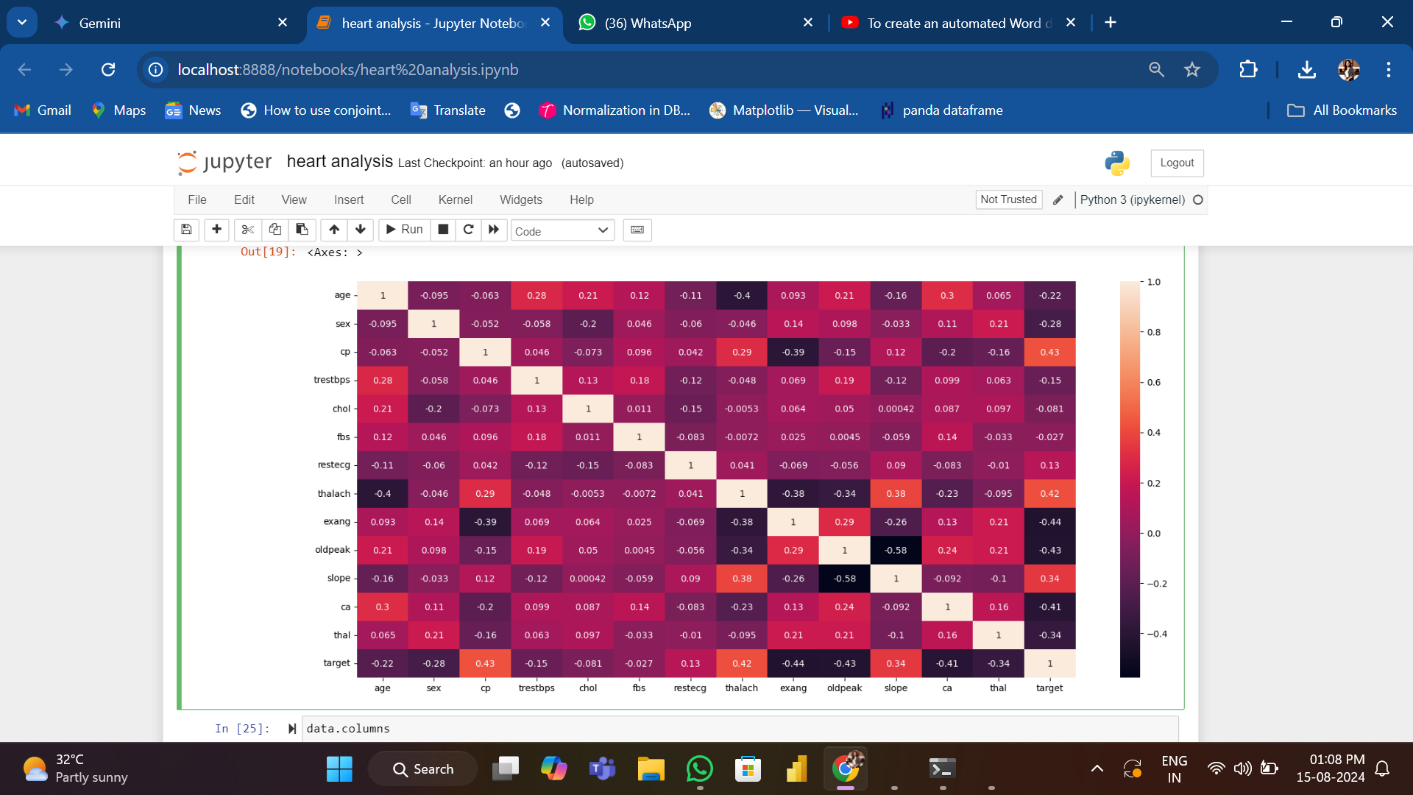
**Explanation of Data Exploration Code**

This code explores and cleans your heart disease data (data DataFrame).

* **data.head(), data.tail():** Show the first/last few rows for a quick look.
* **data.shape:** Reveals the DataFrame's dimensions (rows, columns).
* **data.info():** Provides data type and missing value information for each column.
* **data\_dup = data.duplicated().any(), print(data\_dup):** Checks for duplicate rows (True if any exist).
* **data = data.drop\_duplicates():** Removes duplicate rows.
* **data.isnull().sum():** Counts missing values (NaN) in each column.
* **data.describe():** Generates summary statistics for numerical columns.

**Correlation Matrix**

plt.figure(figsize=(18,8))

sns.heatmap(data.corr(),annot=True)

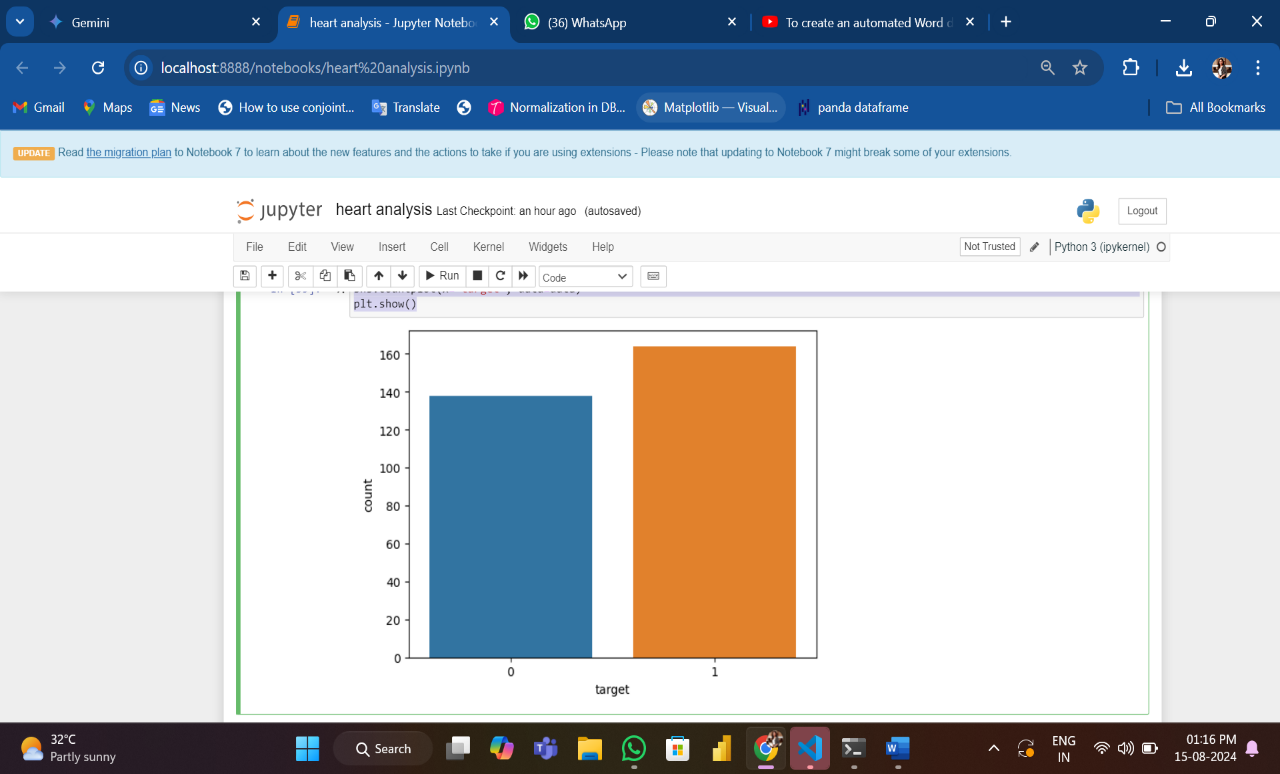
creates a larger space for the image and displays a color-coded chart showing these relationships with numbers.

Counting Heart Disease Cases

data['target'].value\_counts()

sns.countplot(x='target', data=data)

plt.show()



The code calculates the frequency of heart disease cases and visualizes the results. It first determines the number of people with and without heart disease, then creates a bar chart to illustrate this distribution.

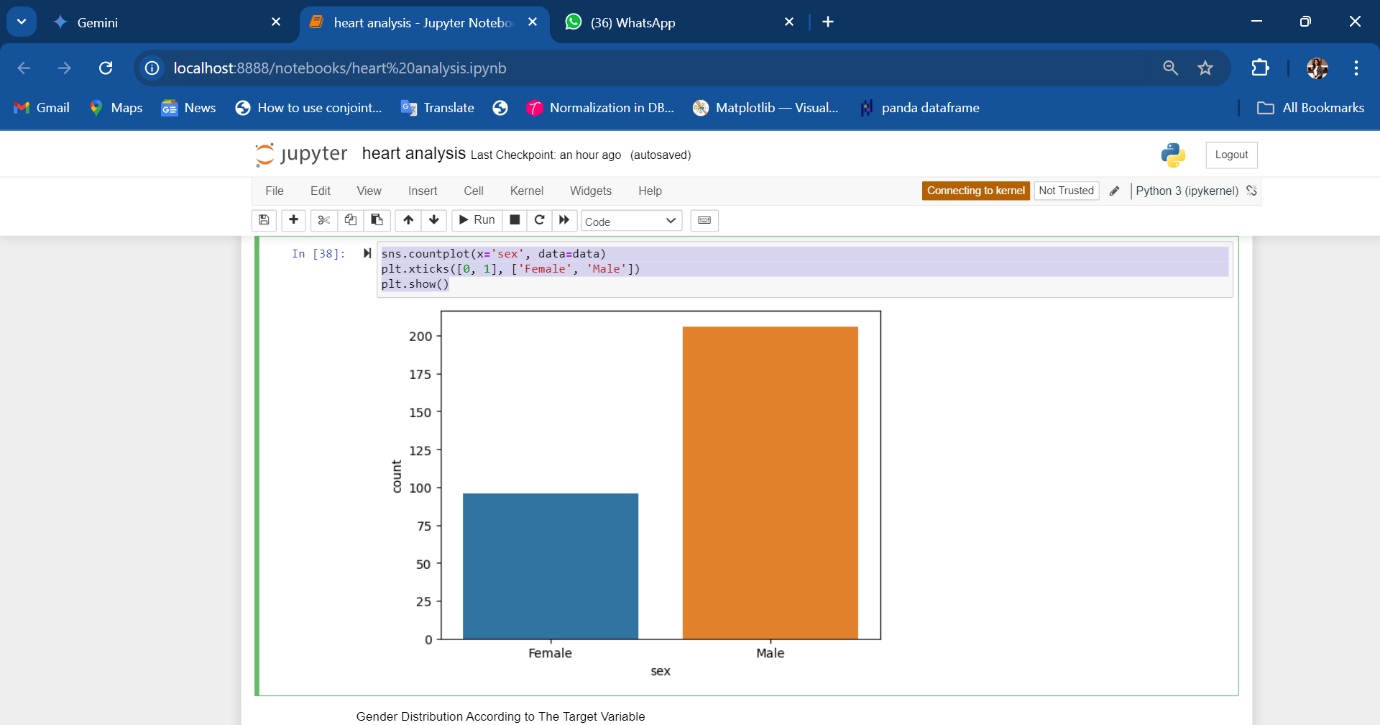
**Count of Male & Female in this Dataset**

data['sex'].value\_counts()

sns.countplot(x='sex', data=data)

plt.xticks([0, 1], ['Female', 'Male'])

plt.show()



This code counts the number of males and females in the dataset and creates a bar chart to visualize it.

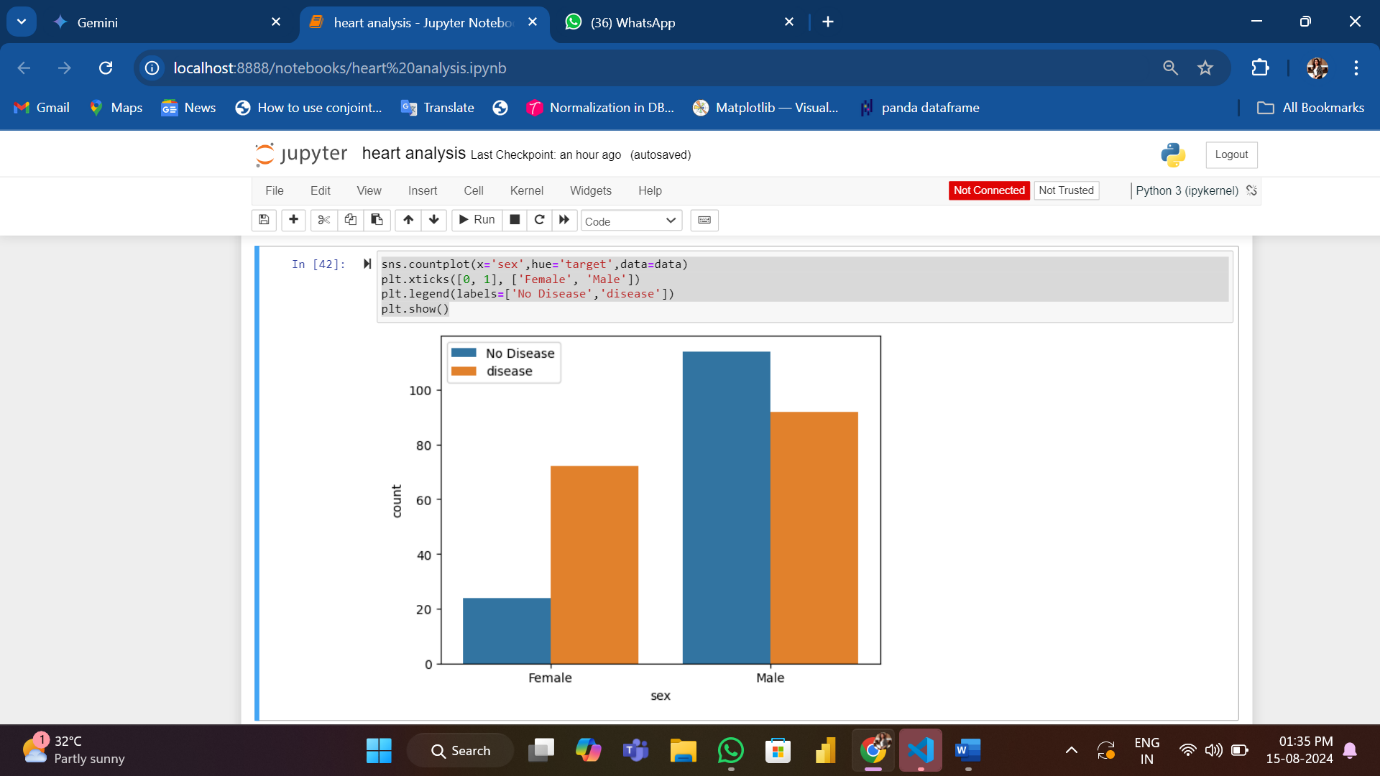
**Gender Distribution According to The Target Variable**

sns.countplot(x='sex',hue='target',data=data)

plt.xticks([0, 1], ['Female', 'Male'])

plt.legend(labels=['No Disease','disease'])

plt.show()



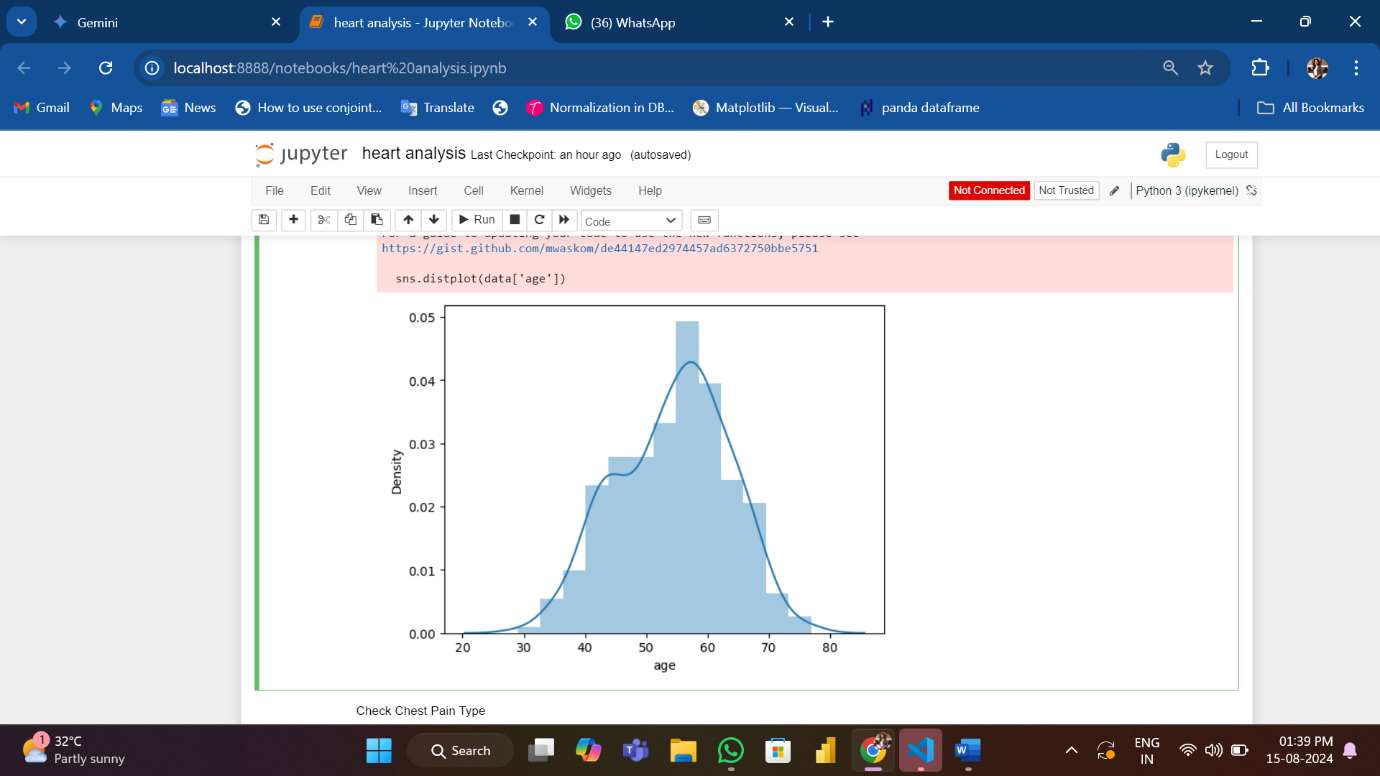
* X-axis: Shows sex (likely male/female).
* Bars: Count individuals in each sex category.
* Color: Represents presence/absence of heart disease (adjust legend labels if needed).

This helps visualize how heart disease cases are distributed across genders.

**Age Distribution Visualization :**

sns.distplot(data['age'])

plt.show()



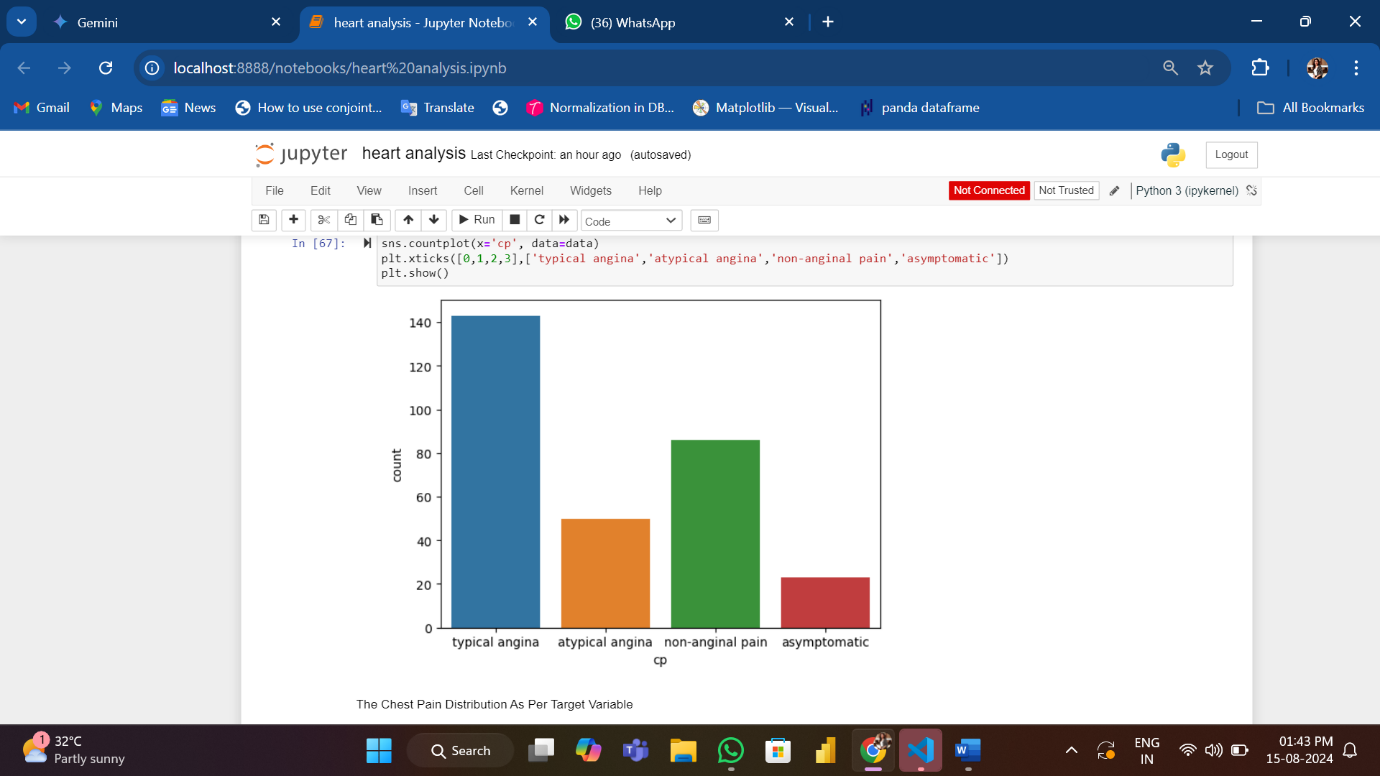
The code creates a histogram to visualize the distribution of ages in your data. This helps understand how many people fall into different age groups.

**Chest Pain Distribution**

sns.countplot(x='cp', data=data)

plt.xticks([0,1,2,3],['typical angina','atypical angina','non-anginal pain','asymptomatic'])

plt.show()



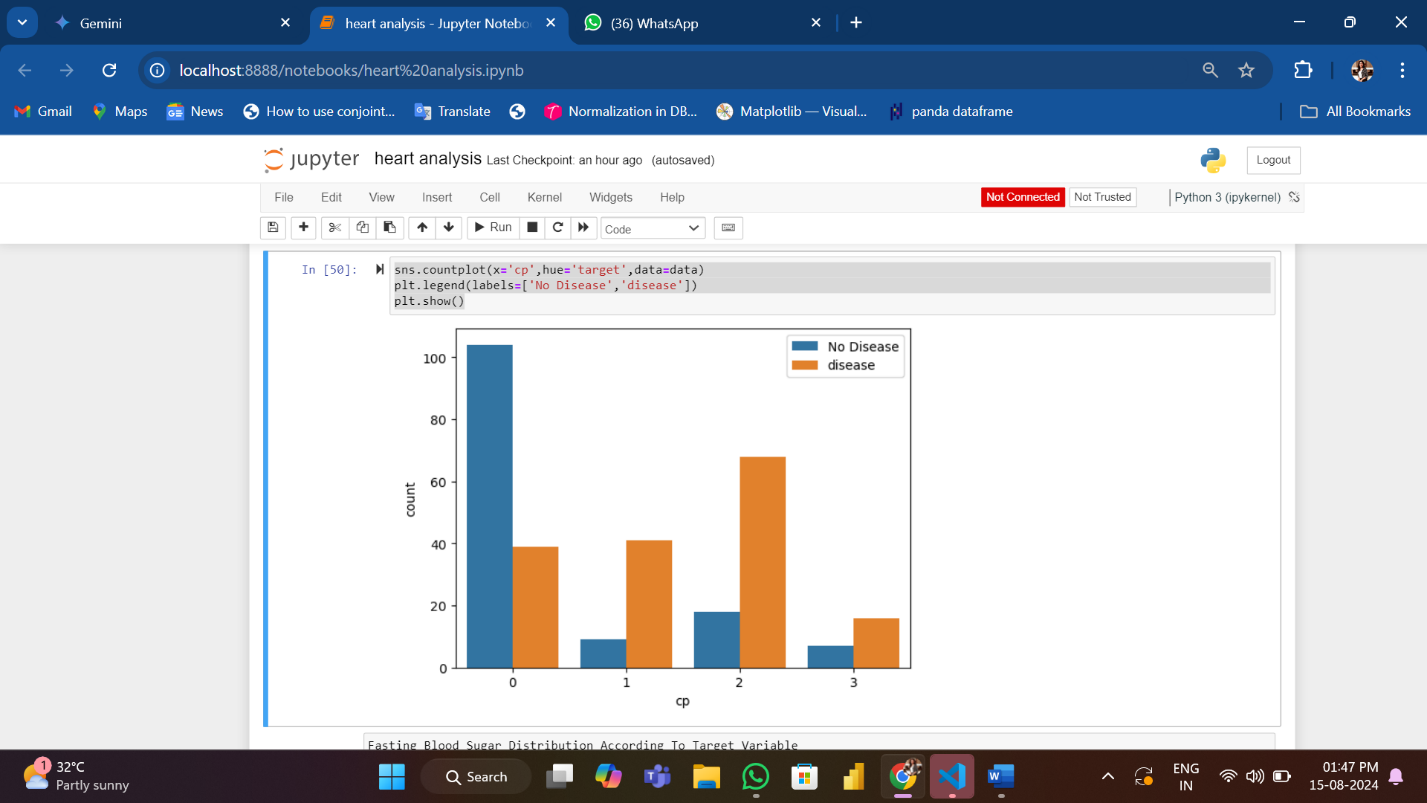
This code creates a countplot to visualize how many people have each type of chest pain in the data. The x-axis categories likely represent classifications used in diagnosing heart disease.

**Chest Pain by Heart Disease**

sns.countplot(x='cp',hue='target',data=data)

plt.legend(labels=['No Disease','disease'])

plt.show()



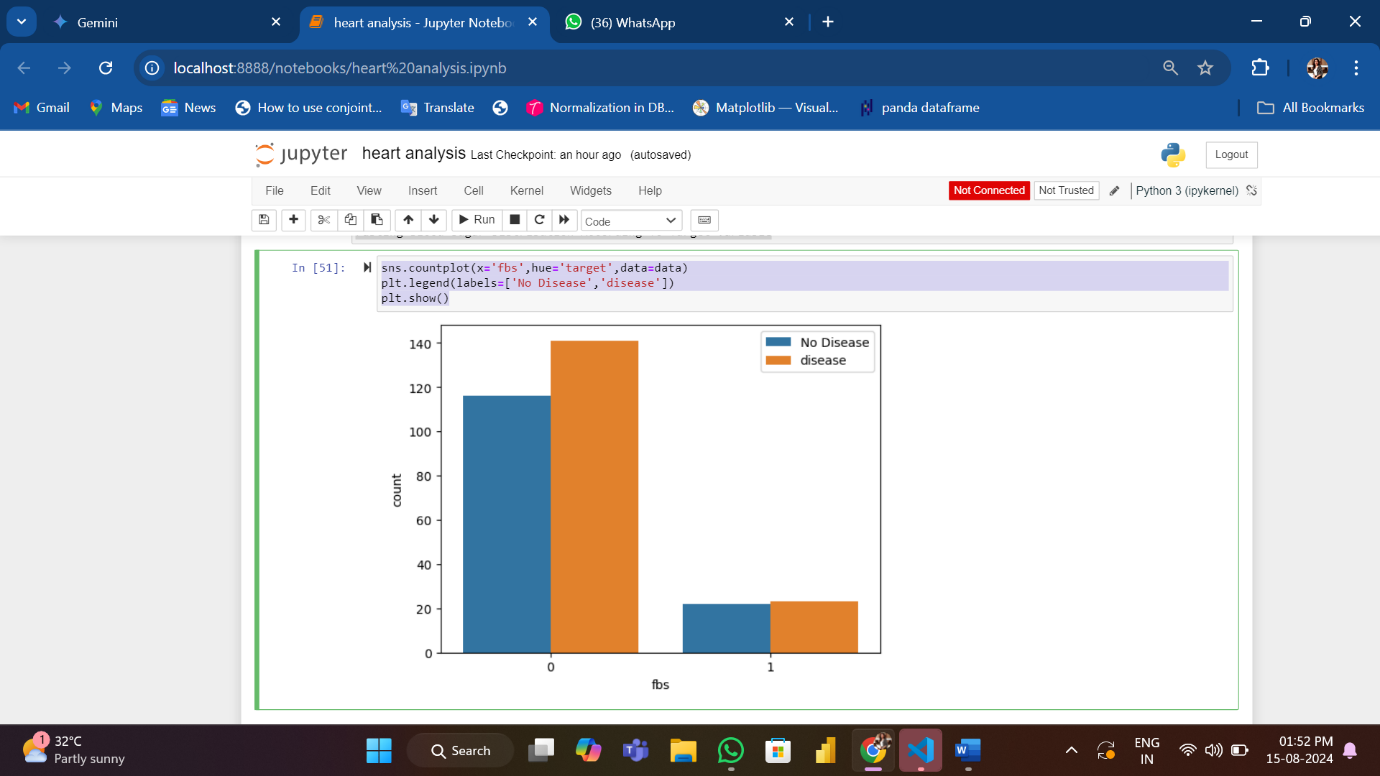
This code creates a colored bar chart to explore the link between chest pain types (cp) and heart disease (target). The colors show how many people with each chest pain type have or don't have heart disease.

**Fasting Blood Sugar (FBS) by Heart Disease**

sns.countplot(x='fbs',hue='target',data=data)

plt.legend(labels=['No Disease','disease'])

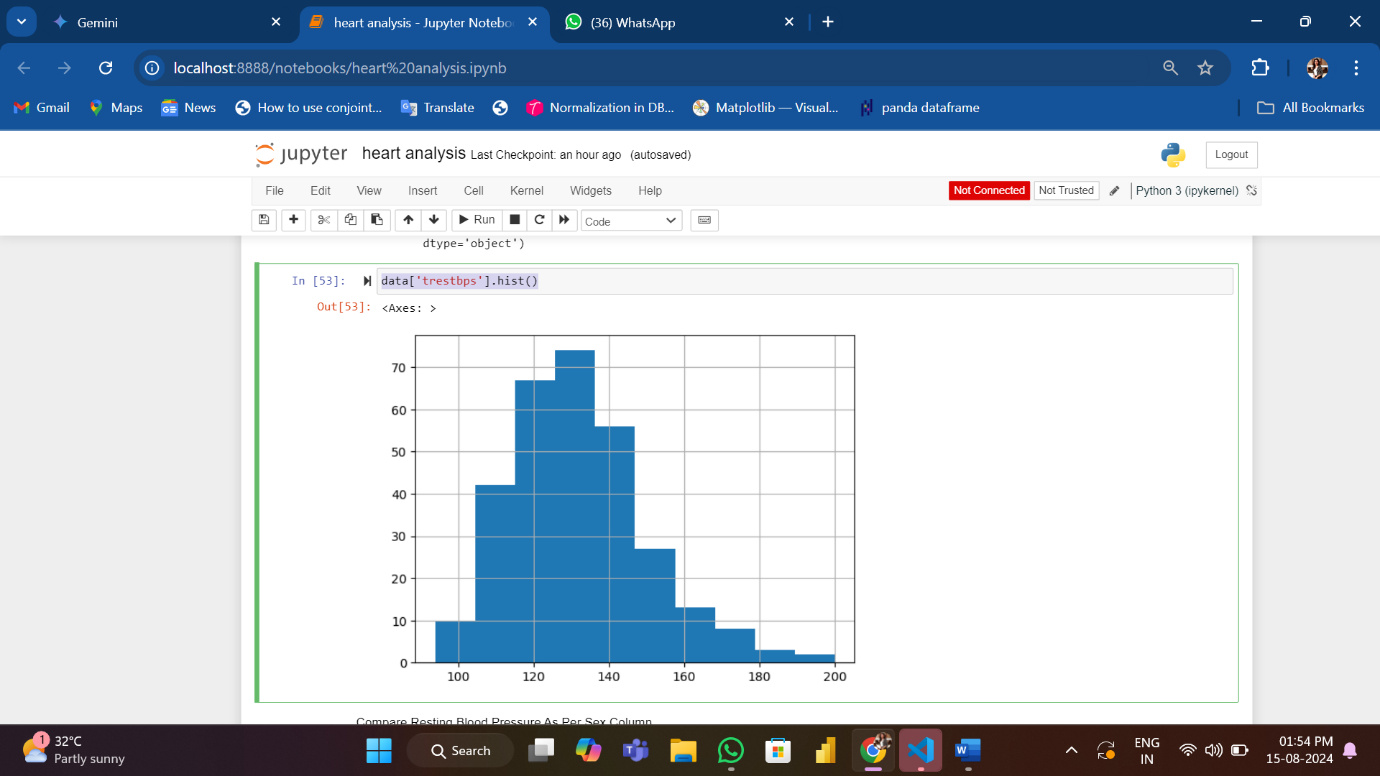
plt.show()



This code creates a colored bar chart to see how fasting blood sugar (FBS) levels relate to heart disease. The colors show how many people with different FBS ranges have or don't have heart disease (adjust legend labels if needed).

**Visualizing Resting Blood Pressure**

data['trestbps'].hist()



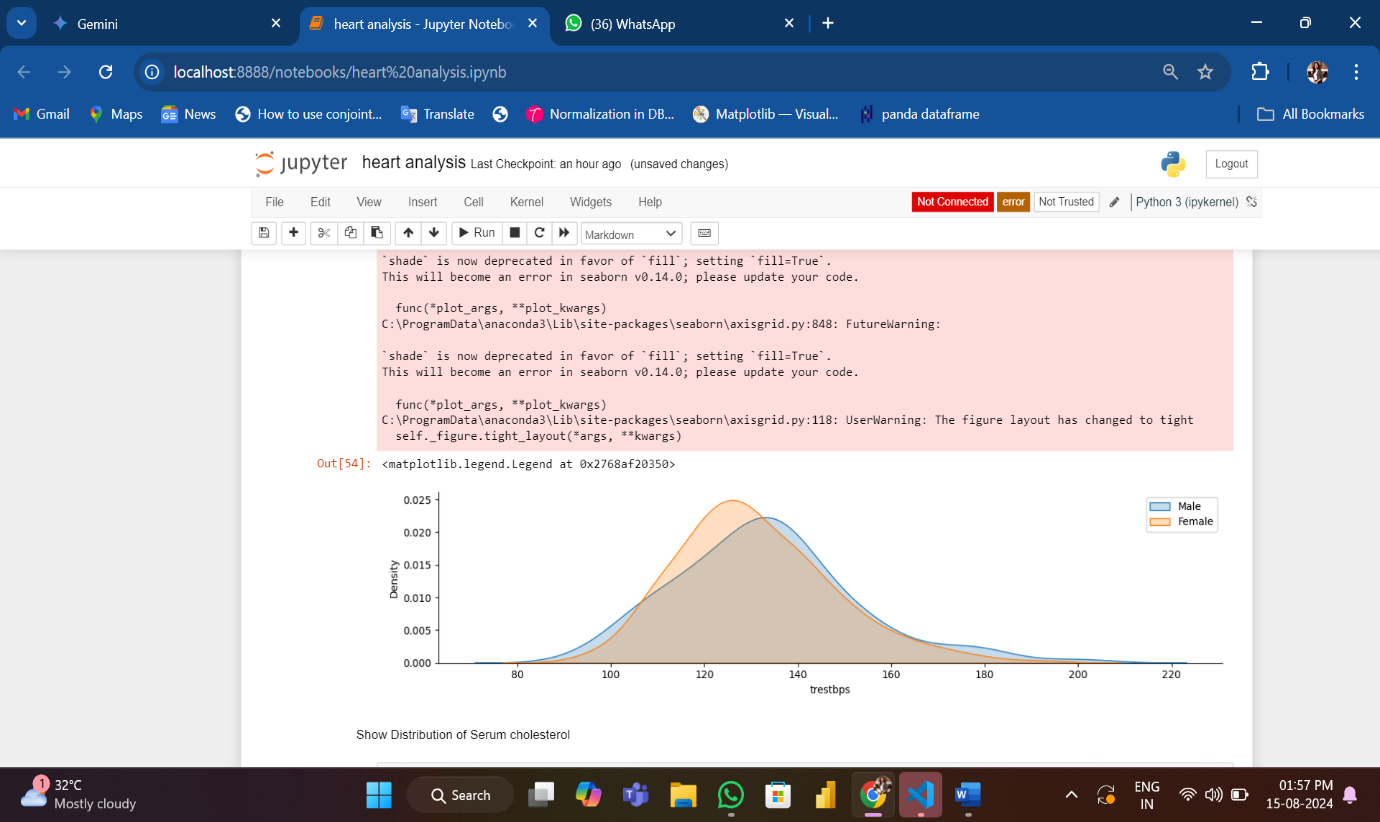
The code generates a histogram to show how resting blood pressure values are distributed across the dataset.

**Sex vs. Resting Blood Pressure Distribution**

g= sns.FacetGrid (data,hue="sex", aspect=4)

g.map(sns.kdeplot, 'trestbps', shade=True)

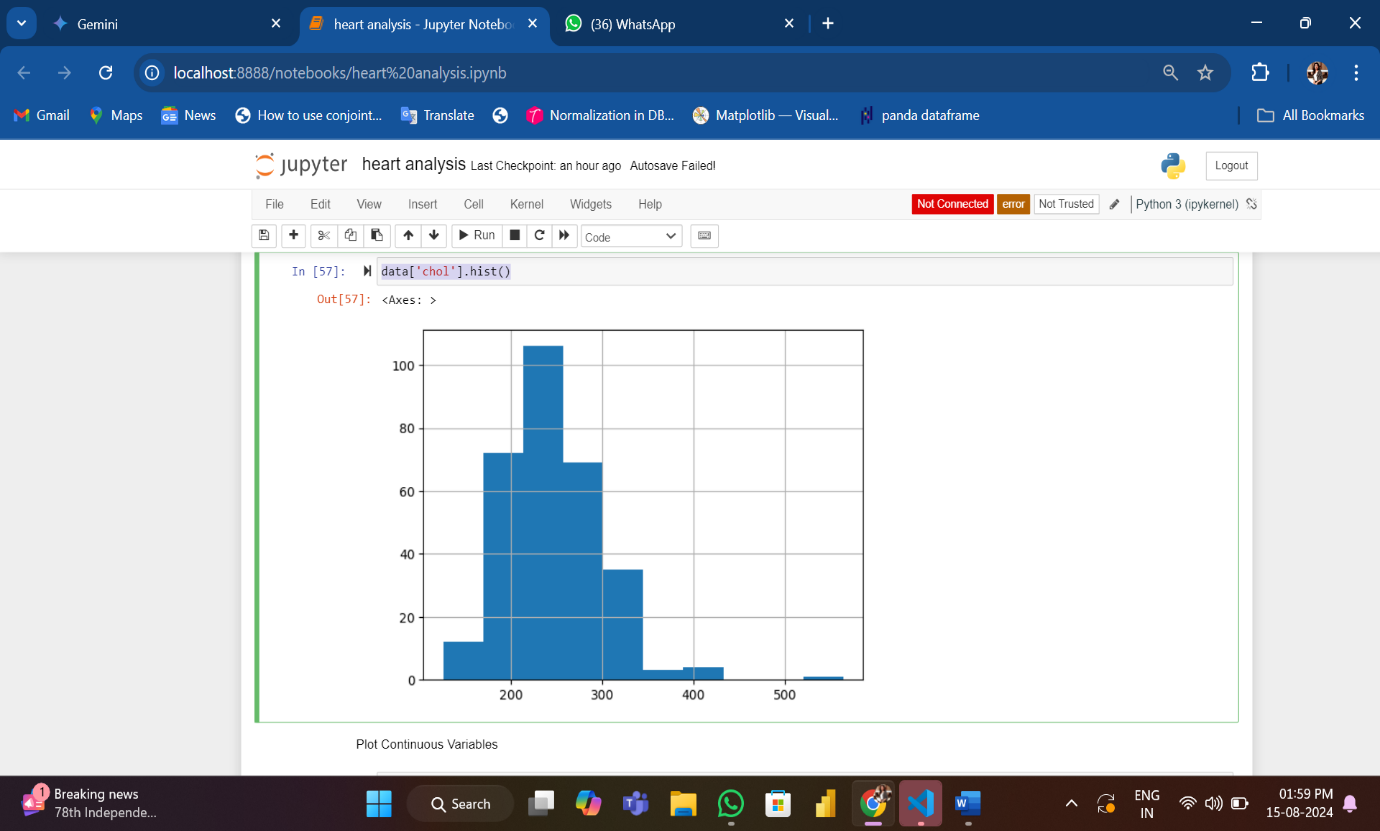
plt.legend(labels=[ 'Male', 'Female' ] )



This code creates a grid comparing resting blood pressure distributions for males and females. It uses kernel density plots with shading to visualize the probability density of each group. This helps understand how blood pressure patterns might differ between genders.

**Visualizing Serum Cholesterol Distribution**

data['chol'].hist()



The code generates a histogram to show how cholesterol levels are spread across the patient population. This helps understand the range and frequency of cholesterol values in the dataset.

**Visualizing Continuous Variables**

cate\_val=[]

cont\_val=[]

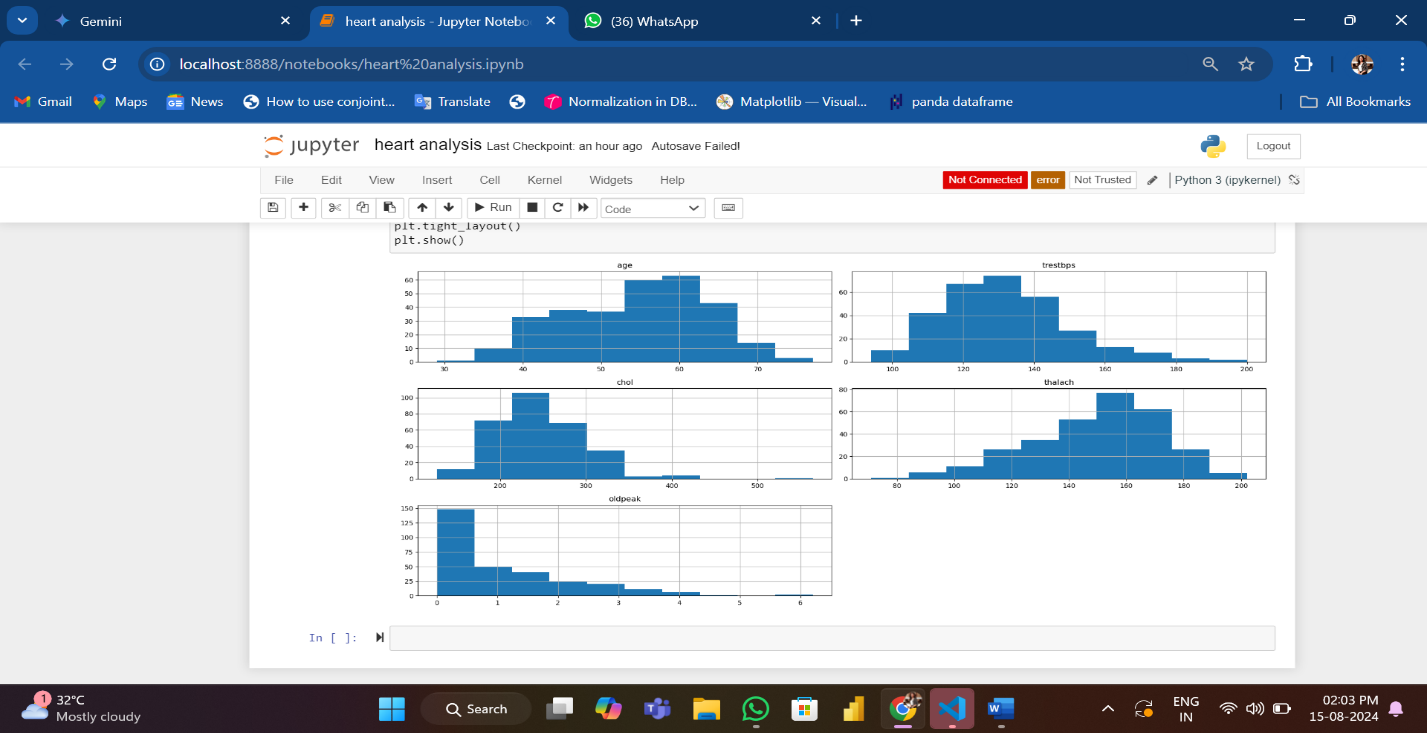
for column in data.columns:

if data[column].nunique() <=10:

cate\_val.append(column)

else:

cont\_val. append(column)



This code identifies continuous variables in your heart disease data and creates histograms to show their distribution. Histograms help understand how frequently values occur across a range.

**Understanding the Data Through Visualization**

By combining descriptive statistics and visual exploration, we gain valuable insights into the heart disease dataset:

* **Data Distribution:** Histograms for age, cholesterol, and blood pressure reveal their distribution patterns, identifying potential outliers or unexpected trends.
* **Categorical Variable Analysis:** Count plots for sex, chest pain type, and fasting blood sugar showcase the frequency of different categories within these variables.
* **Relationships Between Variables:** Correlation matrices and visualizations like scatter plots (not explicitly shown in the provided code) can uncover potential relationships between variables.
* **Target Variable Exploration:** Count plots for the target variable (heart disease presence) and its interaction with other variables (e.g., sex, chest pain) provide crucial information about the disease's distribution within different subgroups.

Overall, these visualizations offer a holistic view of the data, helping to identify patterns, anomalies, and potential areas for further investigation. This initial exploratory phase is essential for building effective predictive models and understanding the factors contributing to heart disease.

**Conclusion**

This exploratory data analysis provides valuable insights into the factors associated with heart disease. By examining variables such as age, sex, chest pain, blood pressure, cholesterol, and blood sugar levels, we can identify potential correlations and trends. These findings serve as a foundation for further in-depth analysis and predictive modeling to better understand and potentially predict heart disease risk.

While this analysis offers preliminary observations, building predictive models and conducting more sophisticated statistical tests would be necessary to draw definitive conclusions about the causes and risk factors of heart disease.