Analyzing Socio-Demographic Trends in India and Germany through Data Analytics and Hypothesis Testing

Cluster Innovation Centre
University of Delhi

Aditya Singh | Shweta Meena | Tushitaa Narayan Ojha | Vikrant Mishra

May 2024
Submitted to
Prof. Dr.Shobha Bagai

Month Long Project submitted for the paper

Applied Probability and Statistics

Certificate of Originality

The work embodied in this report entitled **Analyzing Socio-Demographic Trends in India and Germany through Data Analytics and Hypothesis Testing** has been carried out by Aditya Singh, Shweta Meena, Tushitaa Narayan Ojha and Vikrant Mishra for the paper **Applied Probability and Statistics**. We declare that the work and language included in this project report are free from any kind of plagiarism.

Aditya Singh

Aditya Singh

Shweta.

Shweta Meena

Tushitaa

Tushitaa Narayan Ojha

Vikrant Mishra

Vikrant Mishra

Acknowledgement

We extend our deepest gratitude to **Dr. Shobha Bagai, Dr. Harendra Pal Singh and Dr. Sonam Tanwar** for their unwavering guidance, invaluable support, and scholarly mentorship throughout this project.

Their commitment to academic excellence and dedication to the success of their students have been a constant source of inspiration. We are sincerely thankful for the time and expertise generously shared by **Dr. Shobha Bagai**, which has significantly enriched the quality and depth of this research. Her mentorship has not only enhanced our understanding of the subject matter but has also instilled in us a passion for rigorous inquiry and scholarly exploration.

The project was a big learning experience and to say that our knowledge grew would be an understatement. Finally, we would also like to thank the people at **Cluster Innovation Centre** for providing us with the facilities and the necessary resources to carry out our work. This project would not have been possible without them. Thanks to you all, once again

Abstract

This project explores the application of data analytics to unlock insights into socio-economic trends in two diverse countries: India and Germany. By leveraging a data-driven approach, we aim to understand the underlying patterns, correlations, and disparities across various domains, including but not limited to healthcare, education, economy, and demographics.

The study employs a comprehensive dataset spanning multiple years and sources, encompassing factors such as population demographics, healthcare accessibility, educational attainment, and more.

Comparative analysis between India and Germany offers valuable insights into the unique challenges, opportunities, and developmental trajectories of these nations. By identifying key trends, disparities, and potential areas for improvement, this project aims to inform policymakers, researchers, and stakeholders in both countries to make data-driven decisions for societal advancement and well-being.

Furthermore, to validate our conclusions, we applied null hypothesis testing and probability analysis, ensuring robustness and reliability in our findings. This approach strengthens the credibility of our insights and enhances the applicability of our recommendations for informed decision-making.

I. INTRODUCTION

This project is all about diving deep into how things are going in India and Germany, two countries that are quite different but equally important. India, with its huge population and lots of different challenges, shows us just how complicated rapid development can be. On the flip side, Germany is known for being really good at managing its economy and planning things out carefully. By using data analytics, we're trying to figure out all sorts of things, like how easy it is for people to get healthcare or education, how the economy is growing, and what the population looks like.

But we're not just sitting back and watching the numbers. We're putting them to the test. We're using a hypothesis test called the t-test to dig out some really interesting findings about both India and Germany. These findings aren't just for show; they're super helpful for people who make decisions, like policymakers and researchers. Armed with these insights, they can come up with smart solutions to tackle all sorts of problems society faces.

1.1 Background and Context

The Longitudinal Ageing Study in India (LASI) is a crucial initiative focusing on individuals aged 45 and above, aiming to comprehensively assess various aspects of their well-being, including health, social relationships, mental well-being, and economic status. LASI's dataset for 2017-18 offers a snapshot of the situation, covering demographics, health conditions, healthcare utilization, living arrangements, income, and social support networks among older adults in India, serving as a foundation for evidence-based decision-making and interventions.

Comparing LASI data with similar studies from Germany enables a broader understanding of aging trends and challenges globally. By examining similarities and differences in the health, social, mental, and economic well-being of older adults across diverse cultural and socioeconomic contexts, researchers can identify best practices, innovative solutions, and areas requiring further attention. This comparative analysis contributes to the development of more effective policies and programs to support aging populations worldwide. Additionally, focusing on Indian data allows for a deeper exploration of the unique challenges faced by older adults in the country, leading to the identification of tailored interventions and policy recommendations that address India's specific needs and circumstances, ultimately enhancing the well-being of its aging population.

1.2 Scope and Objectives

Scope -

The project aims to conduct a comparative analysis of aging trends and well-being indicators between India and Germany. It encompasses a comprehensive examination of various dimensions of aging, including health, social relationships, mental well-being, and economic status, among individuals aged 45 and above. The scope extends to gathering reliable data from both countries to facilitate an in-depth comparison of aging experiences, considering diverse demographic, regional, and socioeconomic factors prevalent in India and Germany.

Objectives -

The primary objective of the project is to identify similarities and differences in aging trends and challenges between India and Germany. Through this comparative analysis, the project seeks to gain insights into factors influencing the health, social, mental, and economic well-being of older adults in both countries. Furthermore, the project aims to inform evidence-based policymaking and program development by identifying best practices and areas requiring targeted interventions in India and Germany. Additionally, the project aims to contribute to the global understanding of aging processes by exploring cross-country variations and promoting knowledge exchange and collaboration in aging research and policy development between the two nations.

2. FORMULATION OF THE PROBLEM

2.1 Softwares / Tools Used

Python: Python is a programming language that lets you work more quickly and integrate your systems more effectively. Python is used successfully in thousands of real-world business applications around the world, including many large and mission critical systems.

Colab: Colab is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports multiple programming languages including Python, which is one of the most widely used programming languages in the data science and scientific computing communities. It is often used for data analysis, numerical simulation, machine learning, and more.

Jupiter - Jupyter Notebook is an open-source web application that allows users to create and share documents. Users can write and execute code in individual cells within the notebook, viewing the results directly below each cell. Jupyter Notebooks facilitate reproducible research and collaborative work by combining code, text, and visualizations in a single, interactive document format.

Pandas - Pandas is a software library written for the Python programming language for data manipulation and analysis.

2.2 Methodology

T-Test

1. **Two-Sample t-test for Independent Samples:**

The two-sample t-test for independent samples is a statistical test used to compare the means of two independent groups to determine if there is a significant difference between them.

Calculation of p-value: The t-statistic is calculated based on the difference between the means of the two samples and their standard errors. The p-value is then calculated using the t-distribution, representing the probability of observing the given t-statistic (or a more extreme value) if the null hypothesis is true.

Acceptance/Rejection Criteria: The p-value is compared to a predetermined significance level (alpha), typically set at 0.05. If the p-value is less than alpha, the null hypothesis is rejected, indicating a significant difference between the groups. If the p-value is greater than alpha, the null hypothesis is not rejected, suggesting no significant difference.

2. One-Sample t-test:

The one-sample t-test is a statistical test used to determine whether the mean of a single sample differs significantly from a known or hypothesized population mean.

Calculation of p-value: Similar to the two-sample t-test, the t-statistic is calculated based on the difference between the sample mean and the hypothesized population mean, divided by the standard error of the mean. The p-value is then calculated using the t-distribution.

Acceptance/Rejection Criteria: The p-value is compared to the significance level (alpha). If the p-value is less than alpha, the null hypothesis is rejected, indicating a significant difference between the sample mean and the population mean. If the p-value is greater than alpha, the null hypothesis is not rejected.

These tests are used to compare sample data with population parameters or to compare two independent samples. In our analysis, we used the two-sample t-test to compare demographic indicators between India and Germany, as it allows us to assess whether there is a significant difference in these indicators between the two countries. Similarly, the one-sample t-test was used to evaluate whether the net migration rate in India significantly differs from zero, providing insights into migration trends in the country. These tests provide robust statistical methods for making comparisons and drawing conclusions from sample data

Quadrant Analysis

Quadrant analysis, coupled with statistical measures like the coefficient of correlation, offers a robust method for understanding relationships between variables. Picture a graph divided into four quadrants, each representing unique combinations of high or low values for two variables. The coefficient of correlation quantifies the strength and direction of the relationship between these variables, ranging from -1 to 1.

When the **correlation coefficient is positive, data points tend to cluster in the top-right and bottom-left quadrants**, indicating a positive association where high values of one variable correspond to high values of the other, and vice versa. Conversely, **a negative correlation**

results in data points clustering in the top-left and bottom-right quadrants, suggesting an inverse relationship where high values of one variable correspond to low values of the other.

Quadrant analysis, guided by the coefficient of correlation, allows for a deeper understanding of how variables interact. Analysts can discern patterns and trends within the data, identifying scenarios where variables move in tandem, move inversely, or exhibit no discernible relationship. This approach aids decision-making across various domains by providing actionable insights into the factors influencing outcomes. Whether in business strategy, financial analysis, healthcare planning, or social policy development, quadrant analysis enriched with correlation coefficients equips decision-makers with the tools needed to make informed choices based on robust data analysis.

Linear Regression

Linear regression is a statistical technique used to understand the **relationship between two variables by fitting a straight line to the observed data points**. This line serves as the best approximation of the relationship between the variables, capturing the overall trend in the data. The slope of the line indicates the direction and strength of the relationship: **a positive slope suggests a positive association**, where increases in one variable correspond to increases in the other, while a **negative slope indicates an inverse relationship**.

Through linear regression, analysts can quantify the extent to which changes in one variable are associated with changes in another. The goodness of fit of the regression line can be evaluated using statistical measures such as the coefficient of determination (R-squared), which indicates the proportion of variation in the dependent variable explained by the independent variable. A higher R-squared value suggests a better fit of the line to the data, indicating that the model effectively captures the relationship between the variables.

Overall, linear regression provides a valuable framework for analyzing and interpreting relationships between variables, enabling researchers to make predictions and draw conclusions based on observed data. Whether in forecasting future trends, identifying key factors influencing outcomes, or testing hypotheses, linear regression is a versatile and widely used tool in various fields, including economics, social sciences, engineering, and epidemiology.

Heatmap

Heatmaps can play a valuable role in our comparative analysis between India and Germany within our project. **By utilizing heatmaps, we can visually represent and compare various**

indicators of aging trends and well-being between the two countries. For example, we have created heatmaps to display demographic characteristics, health outcomes, social support networks, and economic status among people of India and Germany.

These heatmaps would allow us to identify areas of similarity and divergence between the two countries, highlighting regions where one country excels or faces challenges compared to the other. By visually encoding the data in this manner, heatmaps provide an intuitive and accessible way to identify patterns, trends, and disparities across different dimensions of aging-related factors.

Furthermore, heatmaps can aid in pinpointing specific areas for further investigation or targeted interventions, based on the observed patterns and variations. Whether exploring differences in healthcare utilization, income distribution, or social engagement among older adults, heatmaps provide a powerful visual tool for synthesizing complex data and informing evidence-based decision-making within our comparative analysis between India and Germany.

Scatter Plot

Scatter plots are the graphs that present the relationship between two variables in a data-set. It represents data points on a two-dimensional plane or on a Cartesian system.

We know that correlation is a statistical measure of the relationship between the two variables relative movements. If the variables are correlated, the points will fall along a line or curve. The better the correlation, the closer the points will touch the line.

The scatter plot explains the correlation between two attributes or variables. It represents how closely the two variables are connected. There can be three such situations to see the relation between the two variables –

1. Positive Correlation
2. Negative Correlation
3. No Correlation

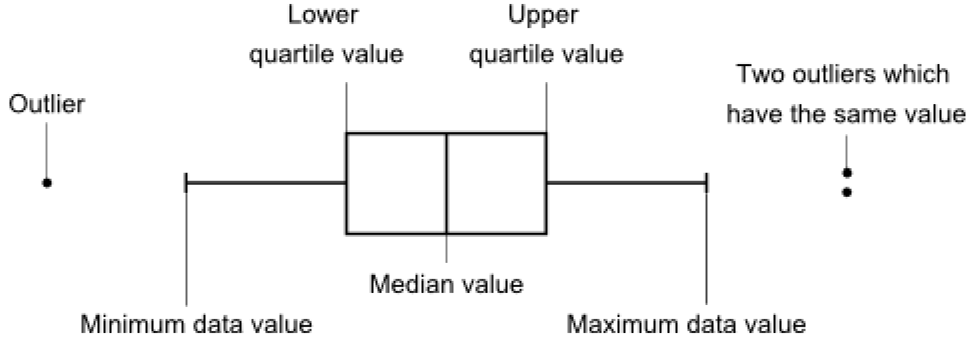
Positive Correlation: When the points in the graph are rising, moving from left to right, then the scatter plot shows a positive correlation. It means the values of one variable are increasing with respect to another.

Negative Correlation: When the points in the scatter graph fall while moving left to right, then it is called a negative correlation. It means the values of one variable are decreasing with respect to another.

No Correlation: When the points are scattered all over the graph and it is difficult to conclude whether the values are increasing or decreasing, then there is no correlation between the variables.

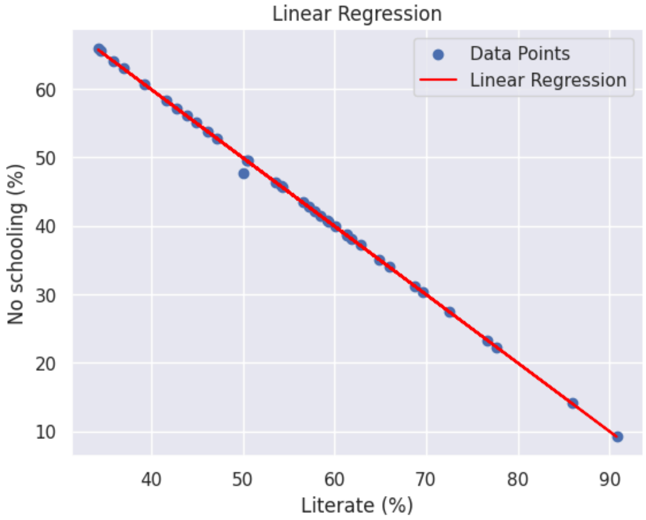
Box Plot

A *box and whisker* plot or diagram (otherwise known as a boxplot), is a graph summarising a set of data. The shape of the boxplot shows how the data is distributed and it also shows any outliers. It is a useful way to compare different sets of data as you can draw more than one boxplot per graph.The line splitting the box in two represents the median value. This shows that 50% of the data lies on the left hand side of the median value and 50% lies on the right hand side. The left edge of the box represents the lower quartile; it shows the value at which the first 25% of the data falls up to. The right edge of the box shows the upper quartile; it shows that 25% of the data lies to the right of the upper quartile value. The values at which the horizontal lines stop at are the values of the upper and lower values of the data. The single points on the diagram show the outliers.

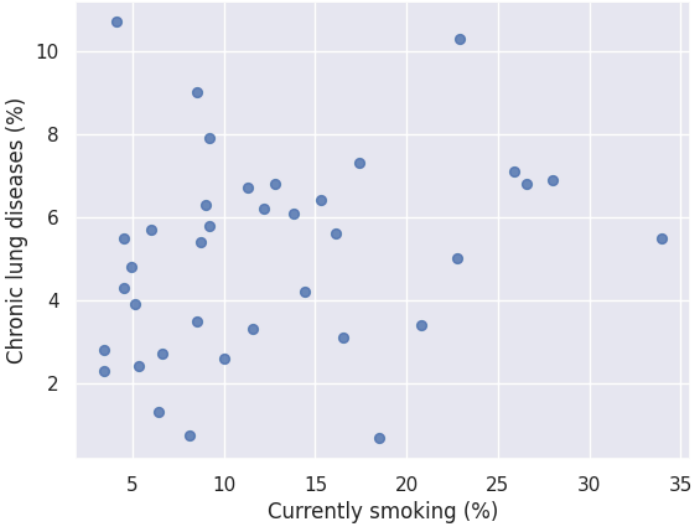


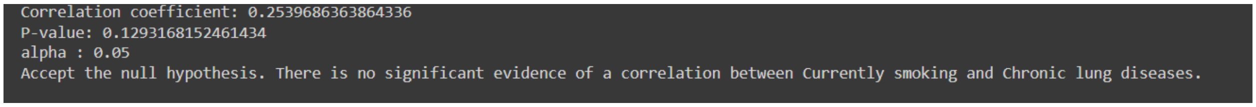
3. RESULTS

Analysis for India

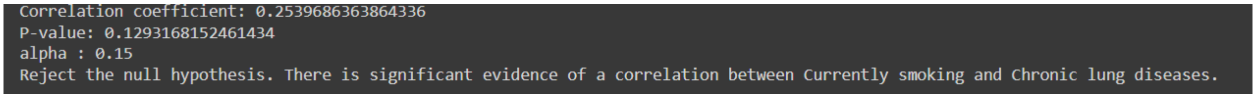


The graph shows literacy rate going up as the number of people with no schooling goes down. This might look like a positive connection, but the correlation coefficient (-0.9996) tells a different story. It shows a very strong opposite relationship. In other words, higher literacy rates mean a much smaller percentage of people with no schooling. There might be a few exceptions in the data, but overall, there's a clear connection between more education leading to better literacy rates.





```
Correlation coefficient: 0.2539686363864336
P-value: 0.1293168152461434
alpha : 0.05
Accept the null hypothesis. There is no significant evidence of a correlation between Currently smoking and Chronic lung diseases.
```

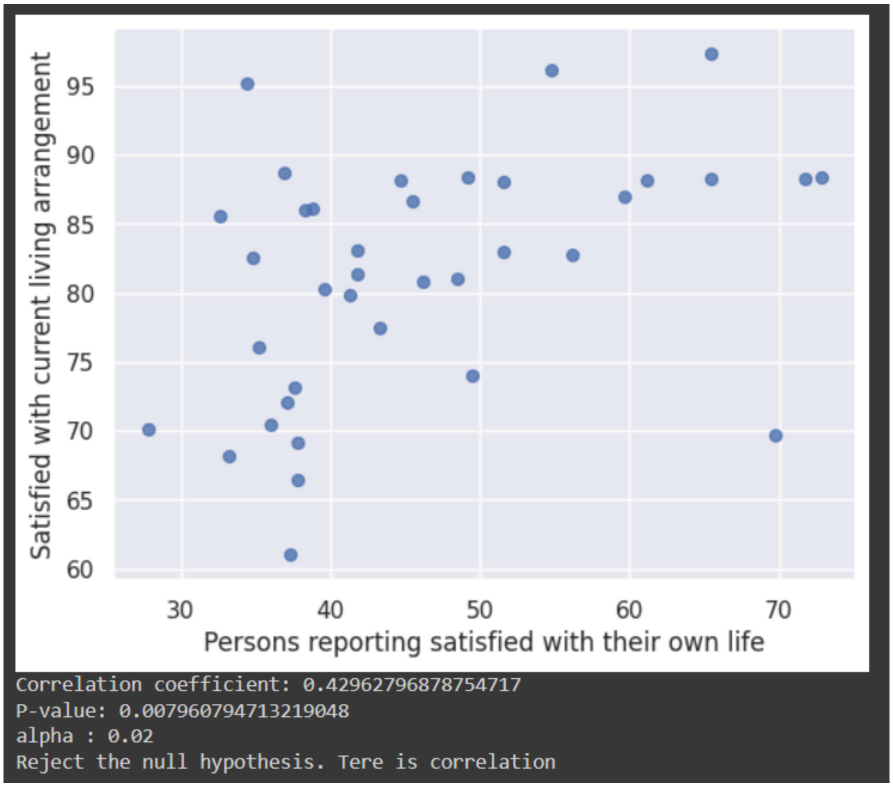


```
Correlation coefficient: 0.2539686363864336
P-value: 0.1293168152461434
alpha : 0.15
Reject the null hypothesis. There is significant evidence of a correlation between Currently smoking and Chronic lung diseases.
```

Upon analysis, it's evident that at a 95% confidence interval (CI), there appears to be no discernible correlation between the two chronic diseases and current smoking habits. However, intriguingly, when we extend the CI, it becomes apparent that there exists a correlation among these variables.

The 95% CI corresponds to an alpha level of 0.05, meaning that there's a 5% chance of observing a relationship purely due to random chance. The correlation coefficient between currently smoking and chronic lung diseases is calculated to be 0.2539686363864336, with a corresponding p-value of 0.1293168152461434. Considering an alpha level of 0.15, we have sufficient evidence to reject the null hypothesis. This indicates a significant correlation between currently smoking and chronic lung diseases at the 15% significance level.

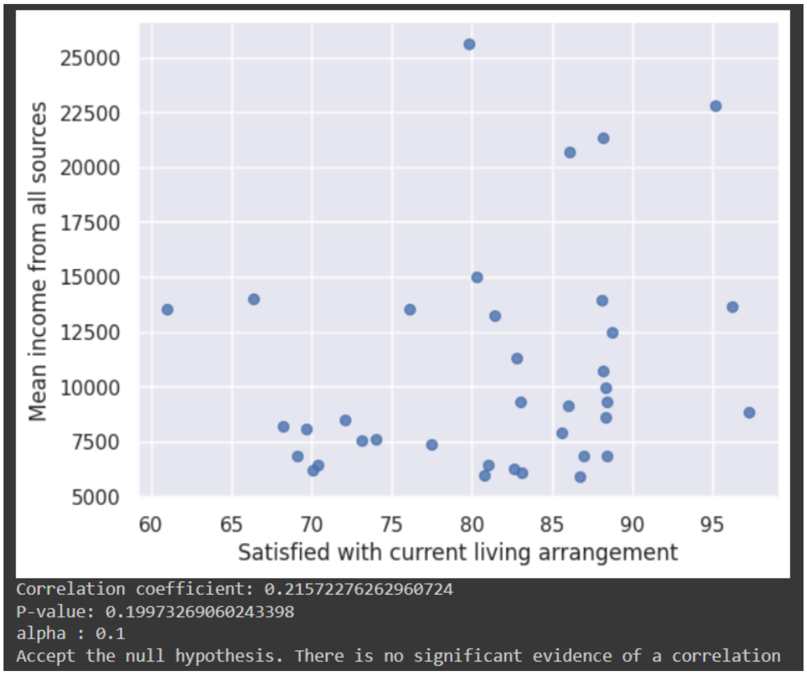
However, when we lower the alpha level to 0.05, we fail to reject the null hypothesis. In this scenario, there is insufficient evidence to conclude a significant correlation between currently smoking and chronic lung diseases at the traditional 5% significance level.



```
Correlation coefficient: 0.42962796878754717
P-value: 0.007960794713219048
alpha : 0.02
Reject the null hypothesis. Tere is correlation
```

Based on the scatter plot and statistical analysis conducted, the correlation coefficient between the percentage of persons reporting satisfaction with their own life and those satisfied with their current living arrangement is calculated to be approximately 0.43, with a corresponding p-value of approximately 0.008. At an alpha level of 0.02, we reject the null hypothesis, indicating a significant correlation between these two variables. This suggests that there is evidence to support the notion that individuals who report higher

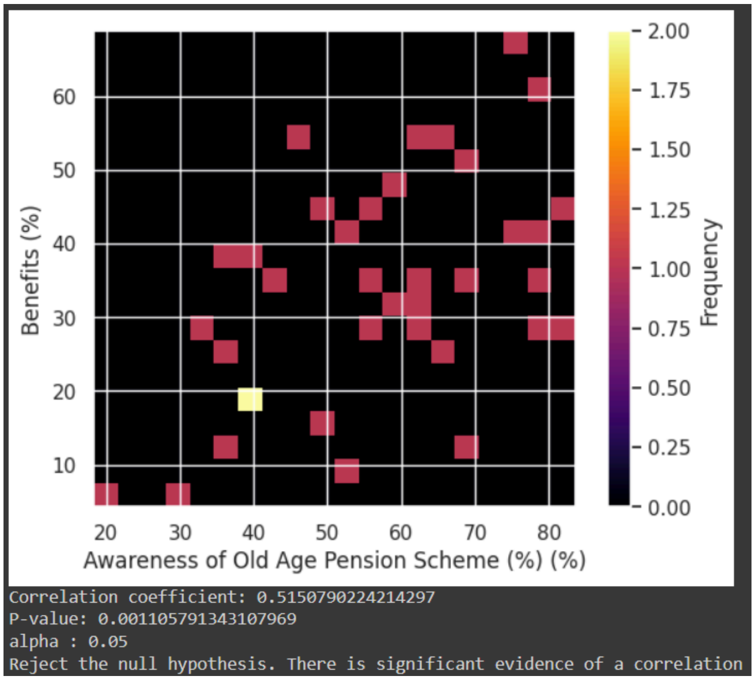
levels of satisfaction with their own lives are also more likely to express satisfaction with their current living arrangements.
This finding underscores the interconnectedness of subjective well-being and living conditions.


Mean income from all sources
25000
22500
20000
17500
15000
12500
10000
7500
5000
60
65
70
75
80
85
90
95
Satisfied with current living arrangement

```
Correlation coefficient: 0.21572276262960724
P-value: 0.19973269060243398
alpha : 0.1
Accept the null hypothesis. There is no significant evidence of a correlation
```

The correlation coefficient between satisfaction with current living arrangement and mean income from all sources is 0.2157 and the p-value is 0.1997. Since the p-value is greater than the significance level (alpha) of 0.1, we fail to reject the null hypothesis. In other words, there is not statistically significant evidence to conclude that there is a correlation between satisfaction with current living arrangement and mean income from all sources.

The scatter plot further supports this conclusion. There is no clear linear pattern in the data points, suggesting that there is no strong relationship between the two variables. While there may be a weak positive correlation visually, it is not statistically significant.

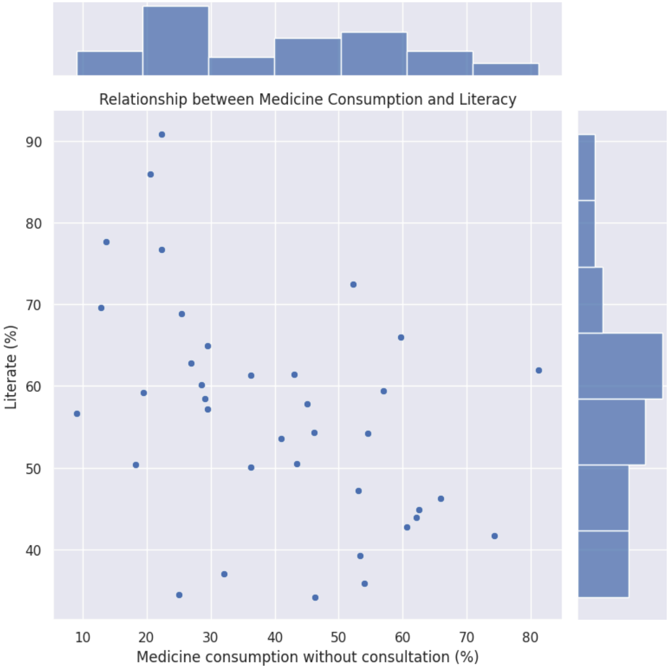


```
Correlation coefficient: 0.5150790224214297
P-value: 0.001105791343107969
alpha : 0.05
Reject the null hypothesis. There is significant evidence of a correlation
```

While the heatmap visualizes the relationship between awareness of the old age pension scheme and frequency, the correlation coefficient of 0.515 suggests a weak positive correlation. This means that there is a slight tendency for awareness of the scheme to increase with more frequent communication.

It's important to note that the correlation coefficient is relatively low, so the relationship is not very strong.

Additionally, even though the p-value of 0.0011 is statistically significant at the alpha level of 0.05, it doesn't necessarily imply causation. There might be other factors influencing awareness of the pension scheme.



The joint scatter plot reveals a negative correlation between medicine consumption without consulting a healthcare provider and literacy. Areas with higher literacy tend to have lower rates of medicine consumption without consulting a healthcare provider. This suggests that individuals with better literacy skills might be more likely to seek professional medical advice before taking medication.

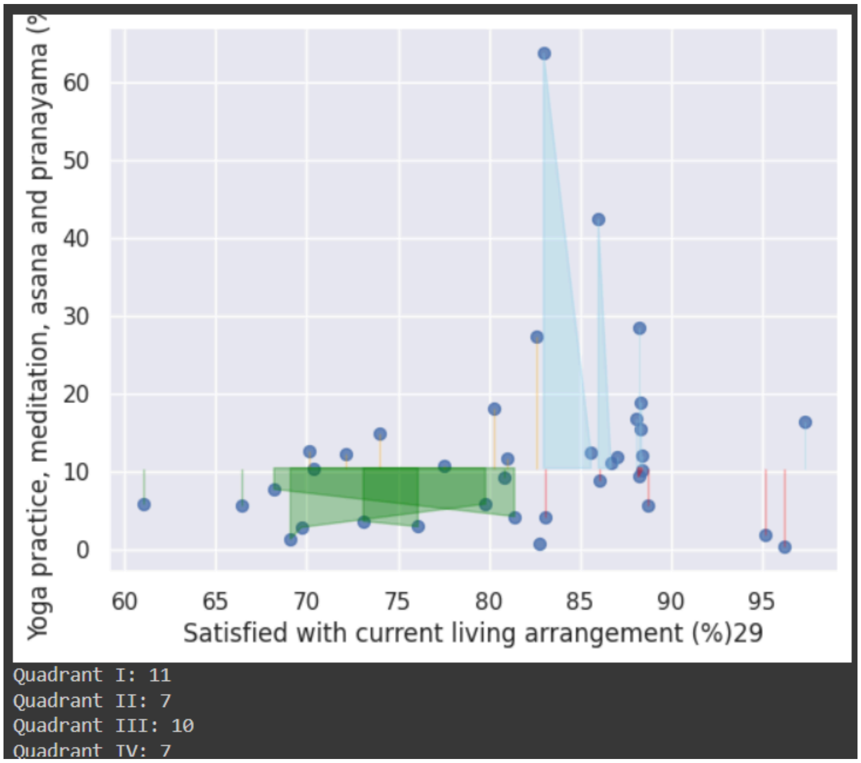
The p-value of 0.0057 in the context of negative correlation test indicates a statistically significant negative correlation between literacy rate and medicine consumption without consultation.

Statistical Significance (p-value): The p-value of 0.0057 is very low. Typically, a significance level of 0.05 (or 5%) is used as a threshold. A p-value lower than this threshold indicates that the observed correlation is unlikely to be due to chance alone. In your case, with a p-value of 0.0057, there is a very strong chance (99.43%) that a true negative correlation exists between literacy and medicine consultation without consultation.

Based on the joint scatter plot and the statistically significant negative correlation (p-value = 0.0057), there's strong evidence that literacy and medicine consumption without consultation are negatively correlated. This suggests that people with higher literacy rates tend to consult healthcare providers more before taking medication.

The p-value (0.0057) confirms a statistically significant negative correlation between literacy and medicine consumption without consultation. This means people with higher literacy rates are less likely to consume medicine without consulting a healthcare provider.

Quadrant Analysis



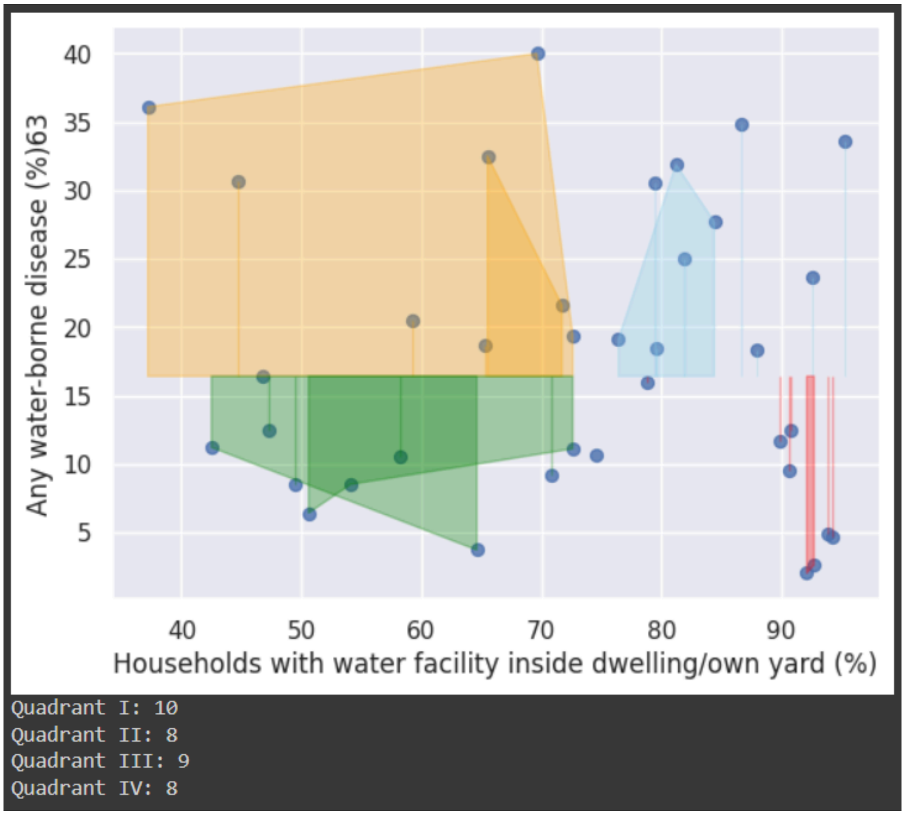
```
Quadrant I: 11
Quadrant II: 7
Quadrant III: 10
Quadrant IV: 7
```

Top Right Quadrant (Sky Blue): This quadrant contains individuals who reported higher than median satisfaction with their living arrangement and practice yoga more than the median.

Top Left Quadrant (Orange): This quadrant contains individuals who reported lower than median satisfaction with their living arrangement but practice yoga more than the median. This could indicate that yoga practice might be a coping mechanism for those who are less satisfied with their living arrangement.

Bottom Left Quadrant (Green): This quadrant contains individuals who reported lower than median satisfaction with their living arrangement and also practice yoga less than the median.

Bottom Right Quadrant (Red): This quadrant contains individuals who reported higher than median satisfaction with their living arrangement but practice yoga less than the median.


Any water-borne disease (%)63
40
35
30
25
20
15
10
5
40
50
60
70
80
90
Households with water facility inside dwelling/own yard (%)

```
Quadrant I: 10
Quadrant II: 8
Quadrant III: 9
Quadrant IV: 8
```

The quadrant analysis suggests a need for targeted interventions in areas with high water-borne disease prevalence despite improved drinking water sources, while also highlighting successful models for disease prevention in regions with both good access to clean water and low disease incidences.

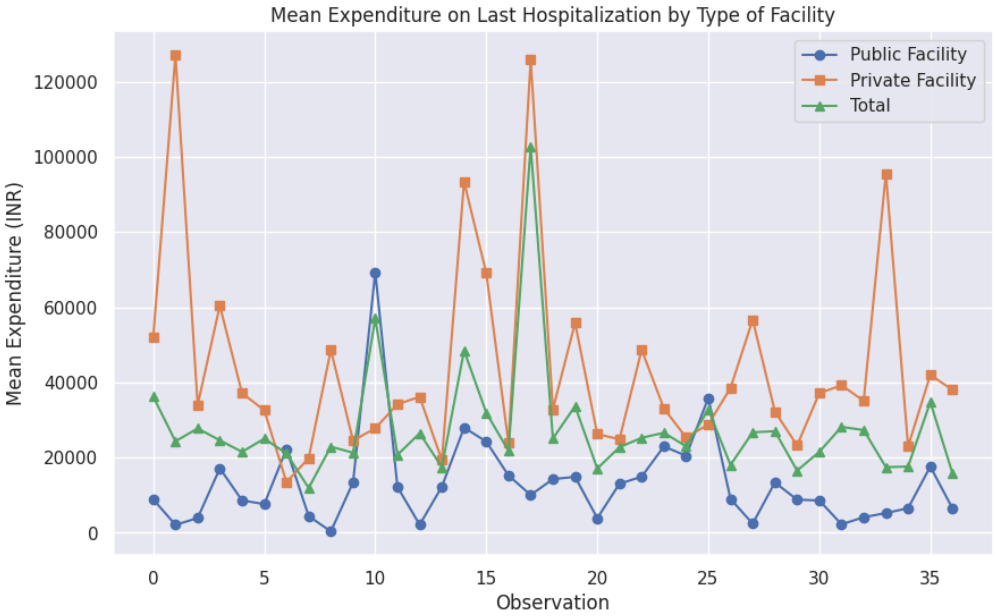
The quadrant analysis reveals a nuanced relationship between access to improved drinking water sources and the prevalence of water-borne diseases:

Areas in Quadrant I exhibit high occurrences of water-borne diseases despite relatively good access to improved drinking water sources, suggesting potential issues with water quality or sanitation infrastructure that need urgent attention to mitigate health risks.

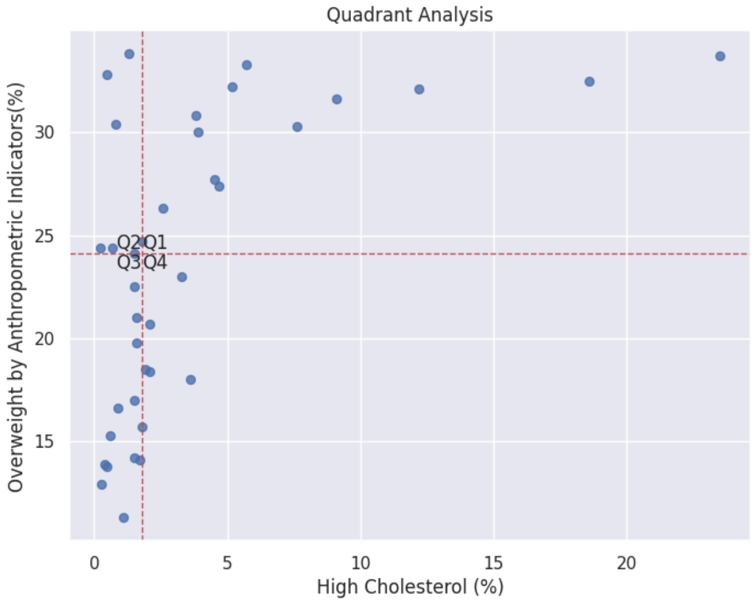
Quadrant II indicates regions with both low access to improved drinking water sources and high incidences of water-borne diseases, emphasizing the critical need for interventions to improve water infrastructure and sanitation facilities to reduce disease burden.

In Quadrant III, where both access to improved drinking water sources and the prevalence of water-borne diseases are low, efforts should focus on maintaining and potentially expanding access to clean water sources to sustain positive health outcomes.

Quadrant IV reflects areas with adequate access to improved drinking water sources and low occurrences of water-borne diseases, highlighting successful initiatives in providing clean water and effective disease prevention measures. These areas serve as models for best practices in public health and sanitation.

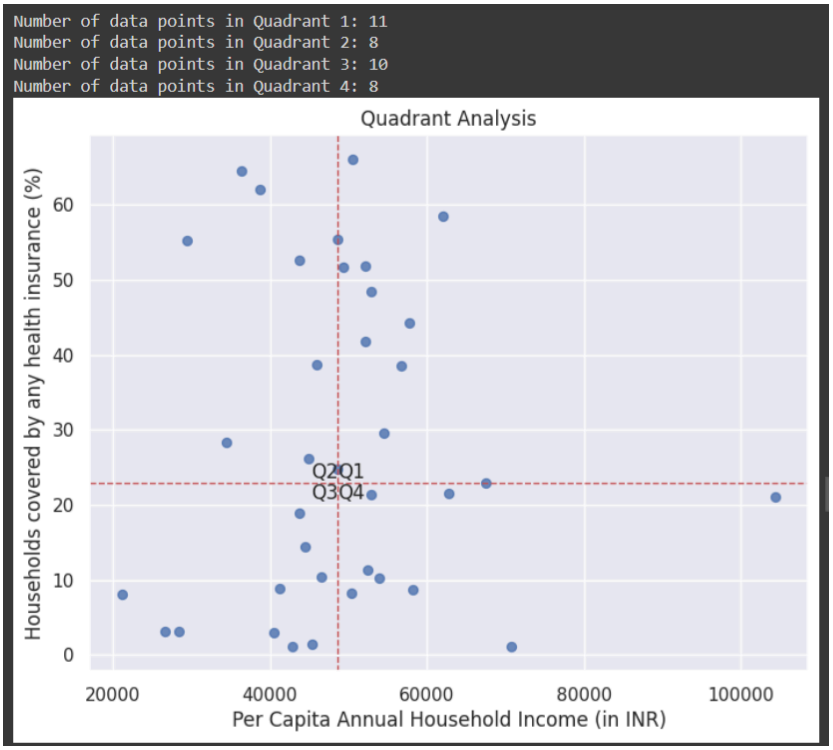


The mean expenditure on hospitalization is significantly higher in private facilities compared to public facilities in India. The graph shows the average cost for public hospitalization ranging between ₹360 and ₹69,347, while private hospitalization costs range from ₹13,448 to ₹127,099. The overall average expenditure on hospitalization falls between ₹3,621 and ₹102,840. It is important to note that this data only shows the mean expenditure, and there can be a significant variation in costs depending on the specific hospital, condition, and procedures required. Additionally, the data does not account for out-of-pocket expenses, which can be a significant burden for patients in both public and private hospitals.



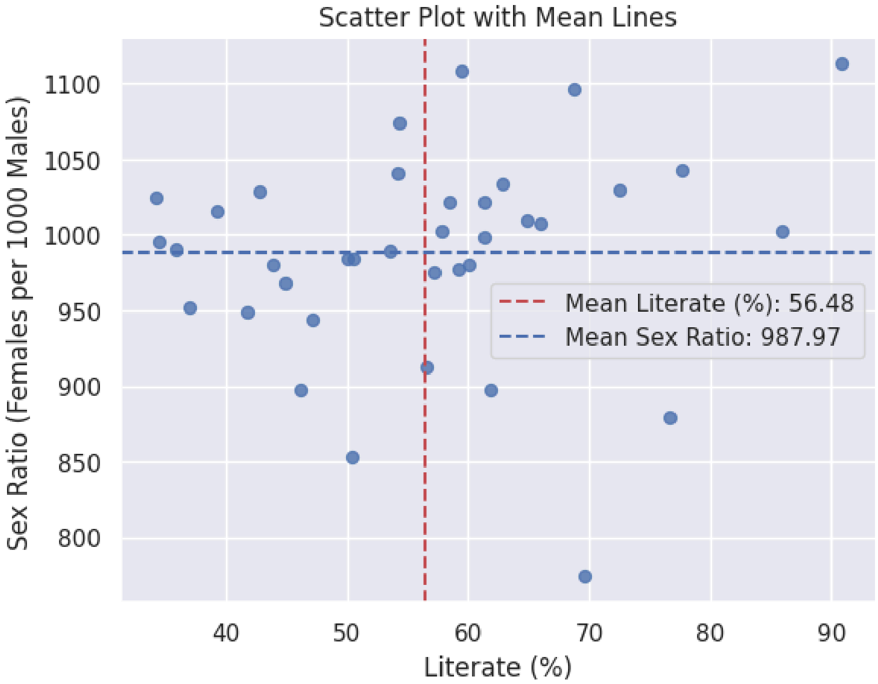
Number of data points in Quadrant 1: 13
Number of data points in Quadrant 2: 6
Number of data points in Quadrant 3: 12
Number of data points in Quadrant 4: 6

There exists a weak positive correlation between overweight and high cholesterol. This is because most of the data points are concentrated in quadrants 1 (Q1) and 3 (Q3). Points in these quadrants have both overweight and high cholesterol values. However, due to the relatively small sample size, it is difficult to draw strong conclusions about the relationship between these two variables. It is also important to consider that correlation does not imply causation.



```
Number of data points in Quadrant 1: 11
Number of data points in Quadrant 2: 8
Number of data points in Quadrant 3: 10
Number of data points in Quadrant 4: 8
```

The quadrant analysis of the provided data reveals insightful patterns regarding household income and health insurance coverage. In Quadrant 1, characterized by high income and high health insurance coverage with 11 data points, affluent communities enjoy robust access to healthcare, likely yielding better outcomes. Quadrant 2, with 8 data points, showcases regions where despite lower income levels, health insurance coverage remains relatively high, possibly due to targeted initiatives. Quadrant 3, encompassing 10 data points, portrays areas facing socio economic challenges with both low income and limited health insurance coverage, indicating barriers to accessing healthcare. Quadrant 4, with 8 data points, highlights regions of high income but low health insurance coverage, suggesting potential gaps in insurance access despite economic advantage. These findings underscore the multifaceted dynamics of healthcare accessibility and emphasize the need for targeted interventions to address disparities and ensure equitable healthcare access across communities.



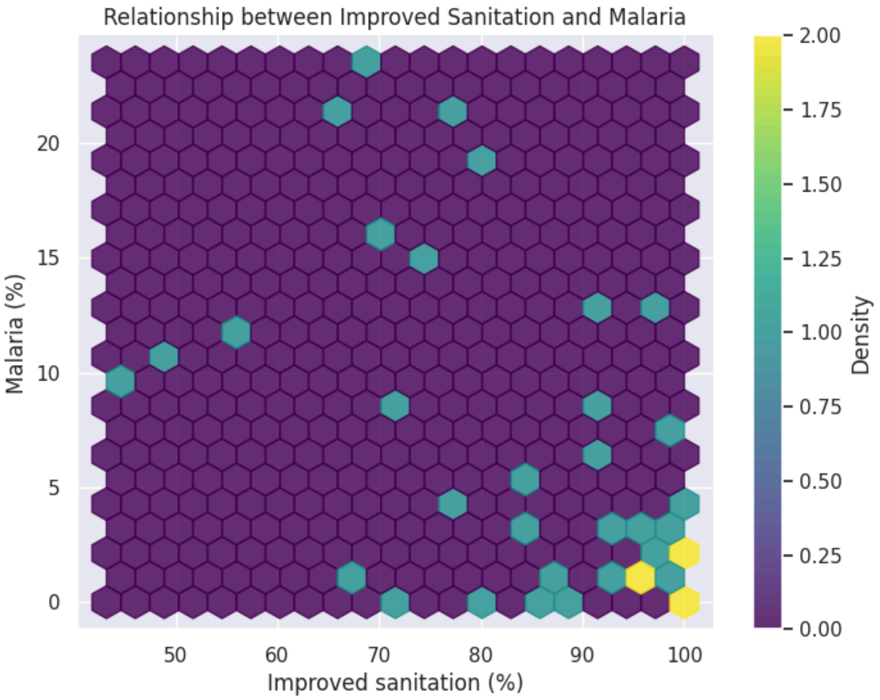
Literate (%) Statistics:
Mean: 56.48
Median: 57.20
Standard Deviation: 13.65

Sex Ratio Statistics:
Mean: 987.97
Median: 995.00
Standard Deviation: 67.72

Here are some details from the plot that support this conclusion:

- **Mean values:** The mean literacy rate is 56.48%, and the mean sex ratio is 987.97 females per 1000 males.
- **Standard deviation:** The standard deviation of literacy rate is 13.65, and the standard deviation of sex ratio is 67.72. This indicates that there is more variation in sex ratio than in literacy rate.

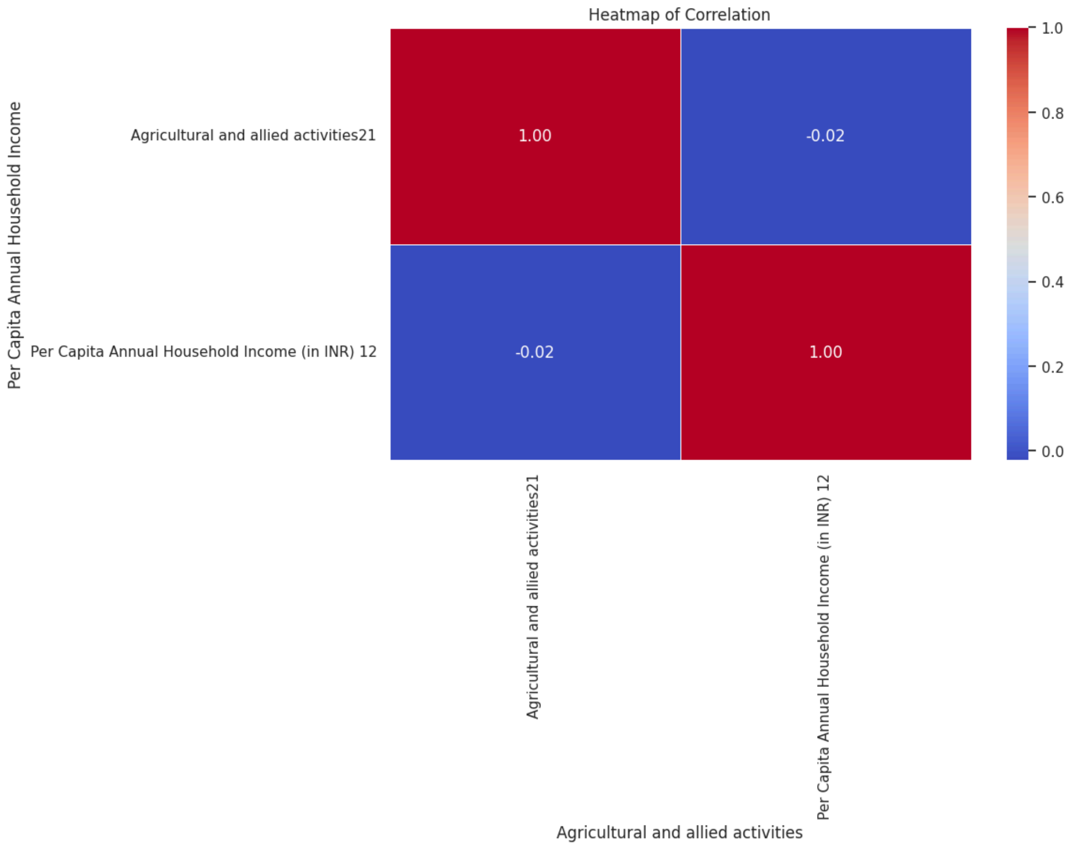
Heat maps

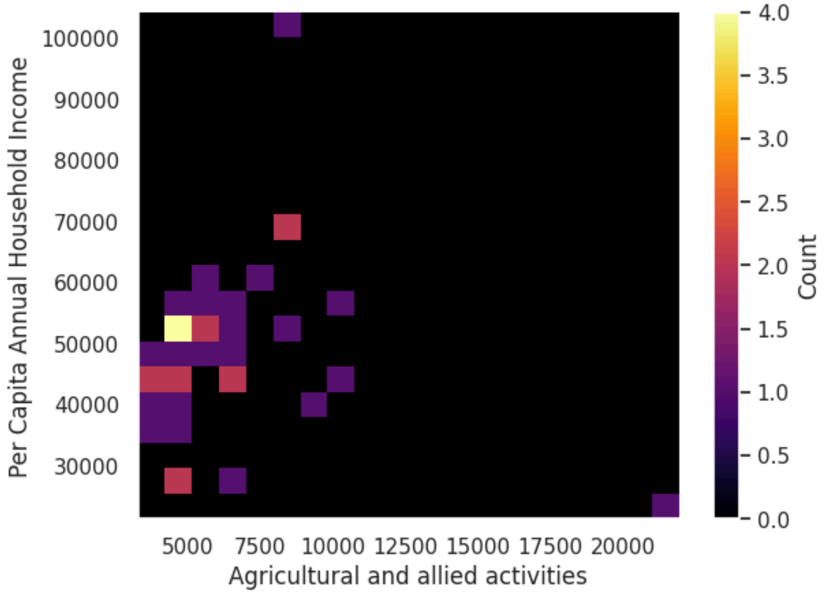


The above hexbin plot suggests that there is a negative correlation between improved sanitation and malaria. This means that areas with a higher percentage of households with improved sanitation tend to have a lower percentage of malaria.

The data is represented by a density plot where the color intensity indicates the number of data points in a certain area. The majority of the data points are concentrated in the bottom left corner of the graph, indicating that most of the data points have low percentages of malaria and high percentages of improved sanitation.

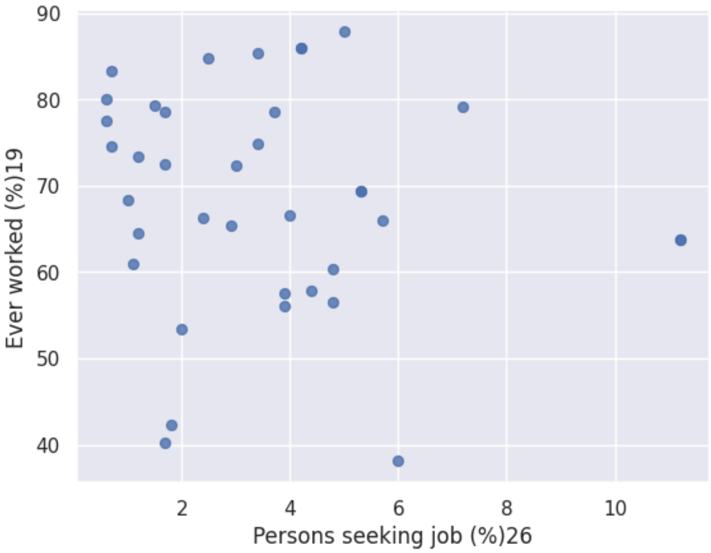
This chart suggests that improved sanitation is associated with a decrease in malaria

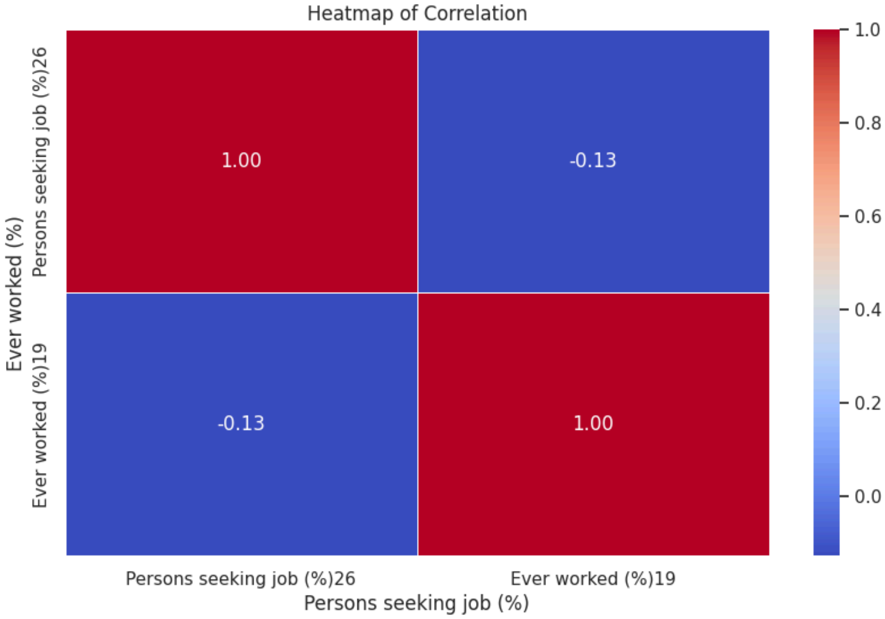




The heatmap doesn't reveal a clear relationship between involvement in agricultural and allied activities and per capita annual household income. However, it's well documented that many Indians are migrating away from agriculture due to lower income compared to other sectors. This trend suggests that the data may show a scattered distribution, where some areas with high agricultural activity might have lower average incomes.

Further analysis, such as looking at regional data or time series trends, could help determine if there's a causal link between low agricultural income and people leaving the sector.





Color and Value Interpretation:

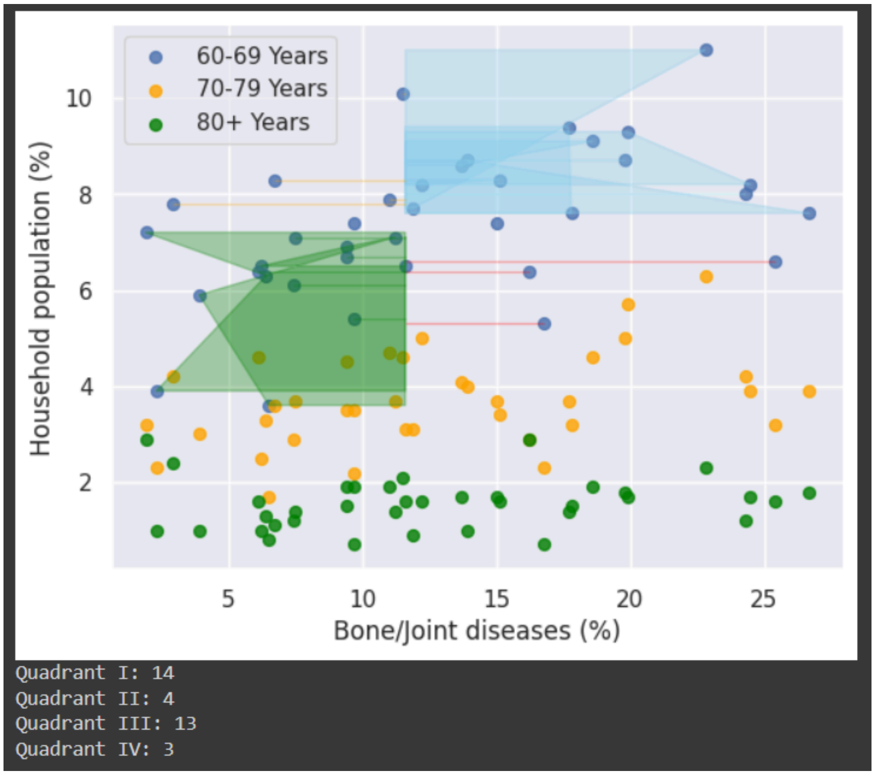
Color: The color intensity in each cell indicates the correlation coefficient between the two variables.

Blue: Negative correlation (values tend to move in opposite directions)
Orange: Positive correlation (values tend to move in the same direction)
White: Weak or no correlation (little to no relationship between the variables)
Values: Numbers within the cells represent the correlation coefficient, ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation), with 0 indicating no correlation.
Interpretation:

The heatmap shows a very weak negative correlation coefficient of approximately -0.13 between "Persons seeking job" and "Ever worked." This is because the color leans slightly blue and the value is close to zero.

A negative correlation here might seem counter-intuitive, but it can be explained by considering the way the data is phrased. "Ever worked" refers to the total population, including those currently employed and those not. So, a higher percentage of people who have ever worked could also include a higher percentage no longer working (i.e. retired). Therefore, a slight negative correlation might emerge because areas with a high percentage of people who have ever worked (including retirees) might also have a higher percentage of people seeking jobs (because they are no longer employed).

Correlation doesn't imply causation. Just because there's a weak negative correlation doesn't necessarily mean that one variable causes the other. Other factors could be influencing both variables.
In conclusion, the heatmap suggests a very weak negative relationship between the two variables. This means there's practically no significant association between the percentage of people seeking jobs and the percentage of people who have ever worked.



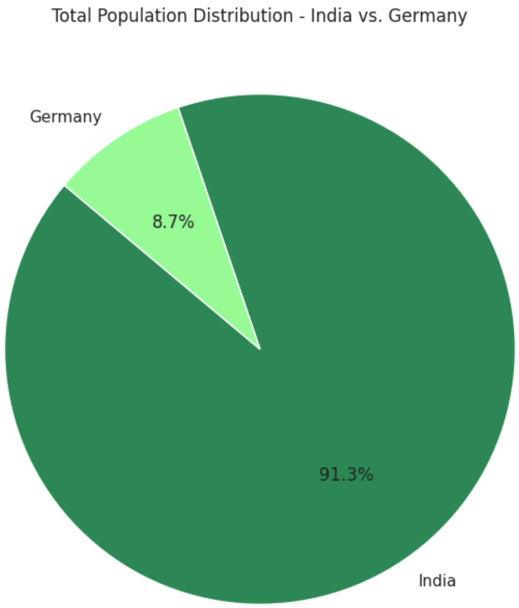
```
Quadrant I: 14
Quadrant II: 4
Quadrant III: 13
Quadrant IV: 3
```

Relationship between the percentage of people with bone/joint diseases and the household population percentage for any of the three age groups (60-69, 70-79, and 80+ years).

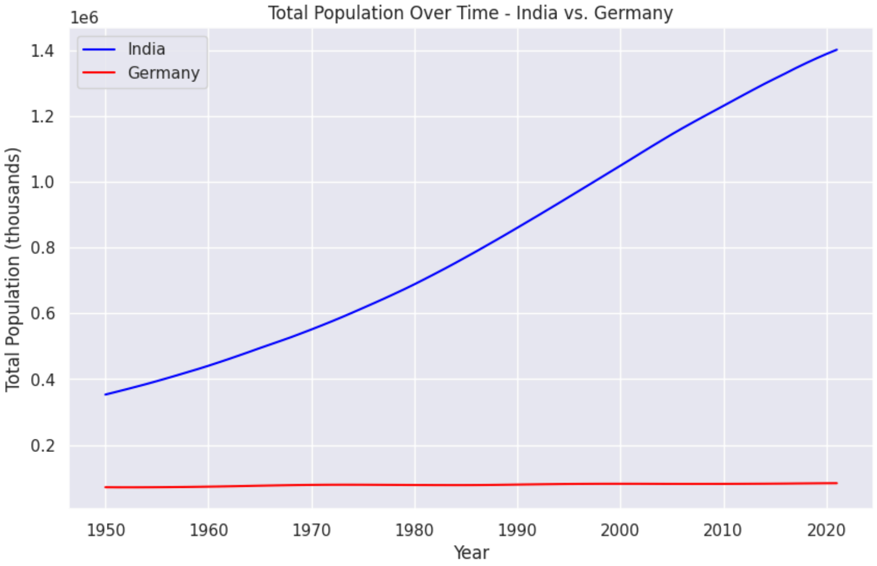
The conclusion that we drew from the scatter plot is that there is no clear correlation between them. There are data points in all four quadrants of the graph for each age group.

Specifically, quadrant I (various colors) contains data points which show a higher percentage of bone/joint diseases and a higher household population percentage. Quadrant II (various colors) contains data points, which shows a lower percentage of bone/joint diseases and a higher household population percentage. Quadrant III (various colors) contains data points, which shows a lower percentage of bone/joint diseases and a lower household population percentage. Quadrant IV (various colors) contains data points, which shows a higher percentage of bone/joint diseases and a lower household population percentage.

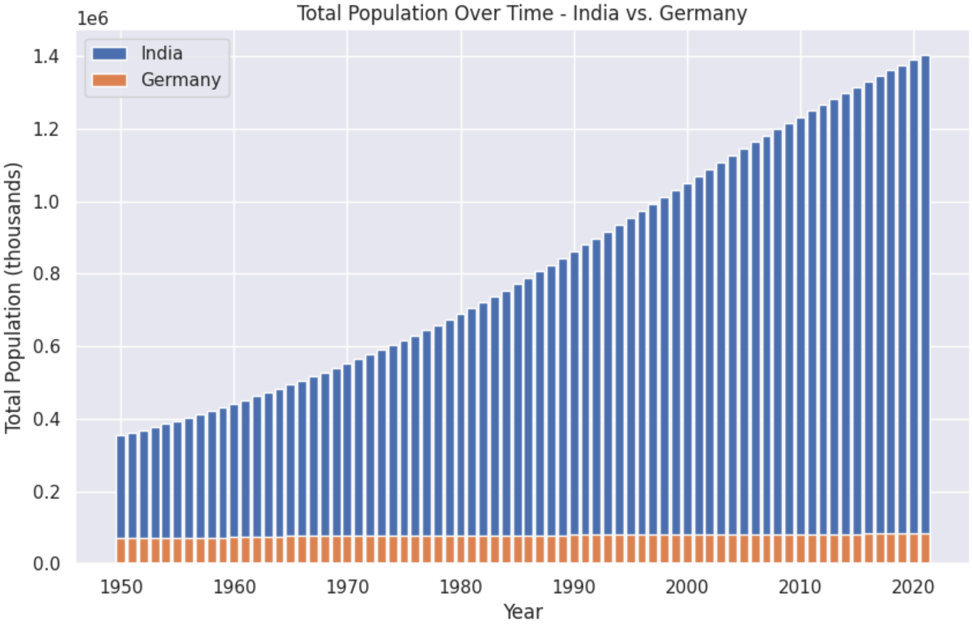
Germany Vs India



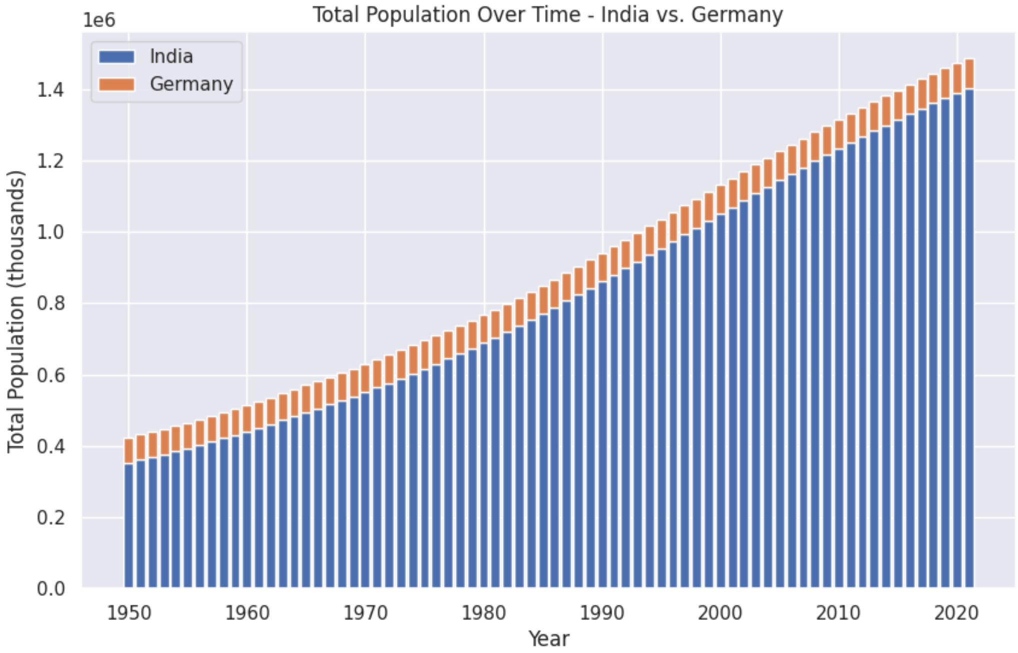
The pie chart provides a clear comparison of the population sizes of the two countries. It offers a concise and visually appealing representation of population distribution, making it easier to grasp the relative magnitudes of population sizes between India and Germany.



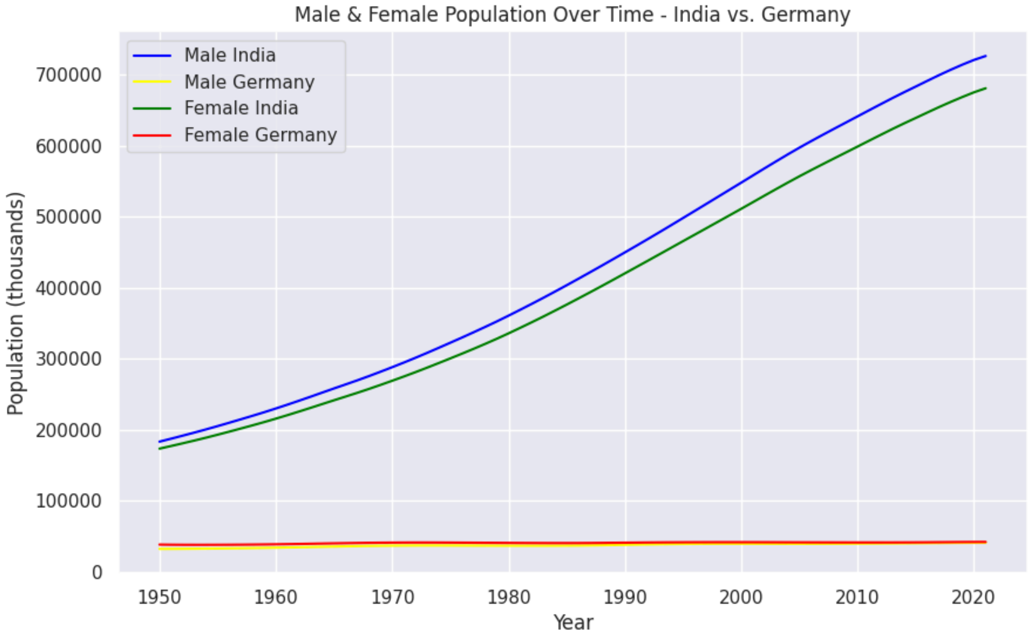
It provides a clear understanding of how the populations of India and Germany have evolved over the specified years.



- The bars represent the total population of India and Germany over the years. Each bar corresponds to a specific year, and its height indicates the total population (in thousands) for that year.
- When the bars for Germany are at the same levels, indicating that they have the same height, it suggests that the population of Germany is not significantly changing over the years.

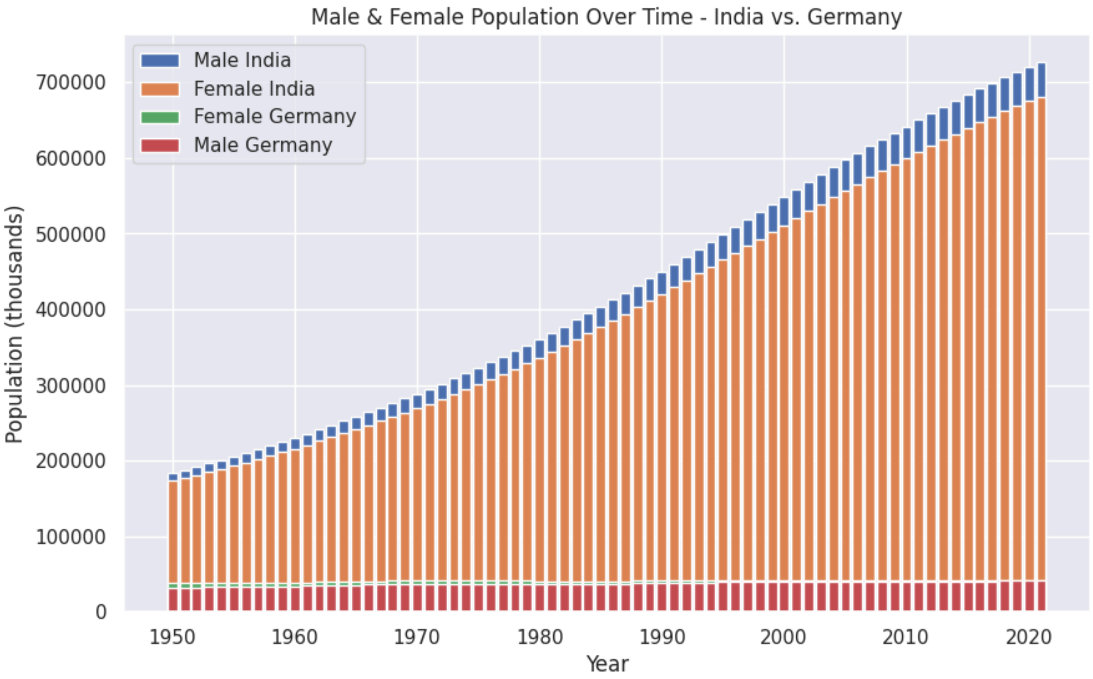


- A stacked bar plot is created for India and Germany, where the bars for Germany are stacked on top of the bars for India. This creates a visual representation of the total population for each country over the years, with the combined height of the bars representing the total population each year.



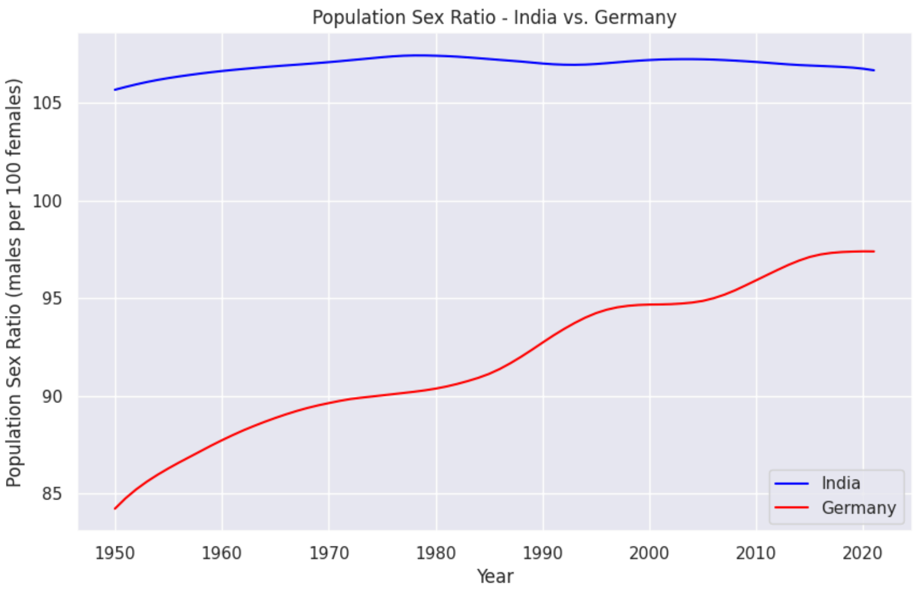
- This graph illustrates the trends of male and female populations over time for India and Germany.

- It provides insights into how the male and female populations have changed over time in both countries. The legend labels indicate which line corresponds to which population and country. Additionally, grid lines are included to aid in better visualization.

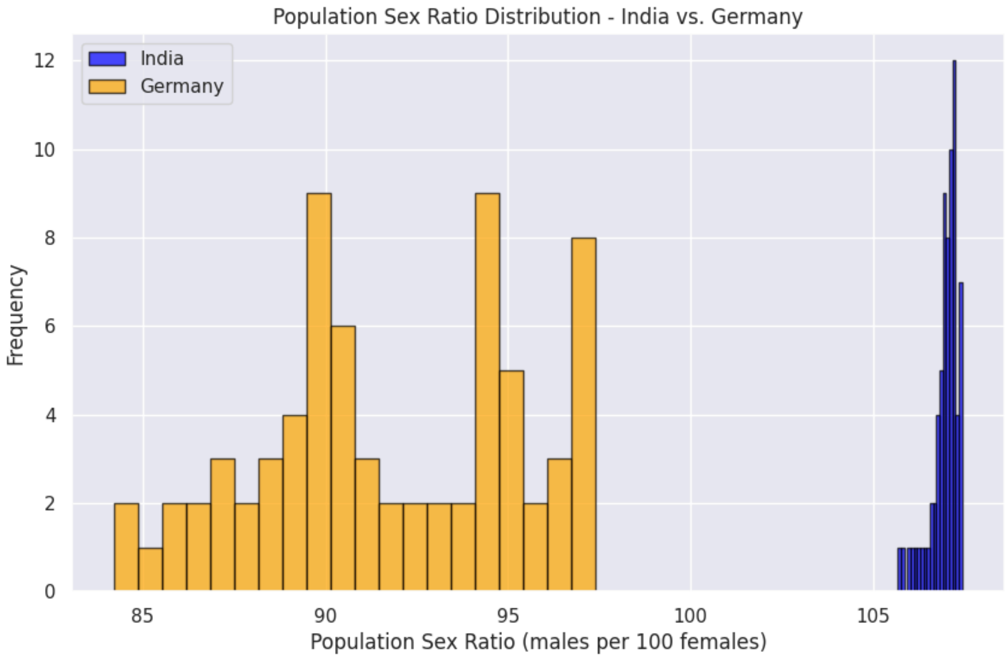


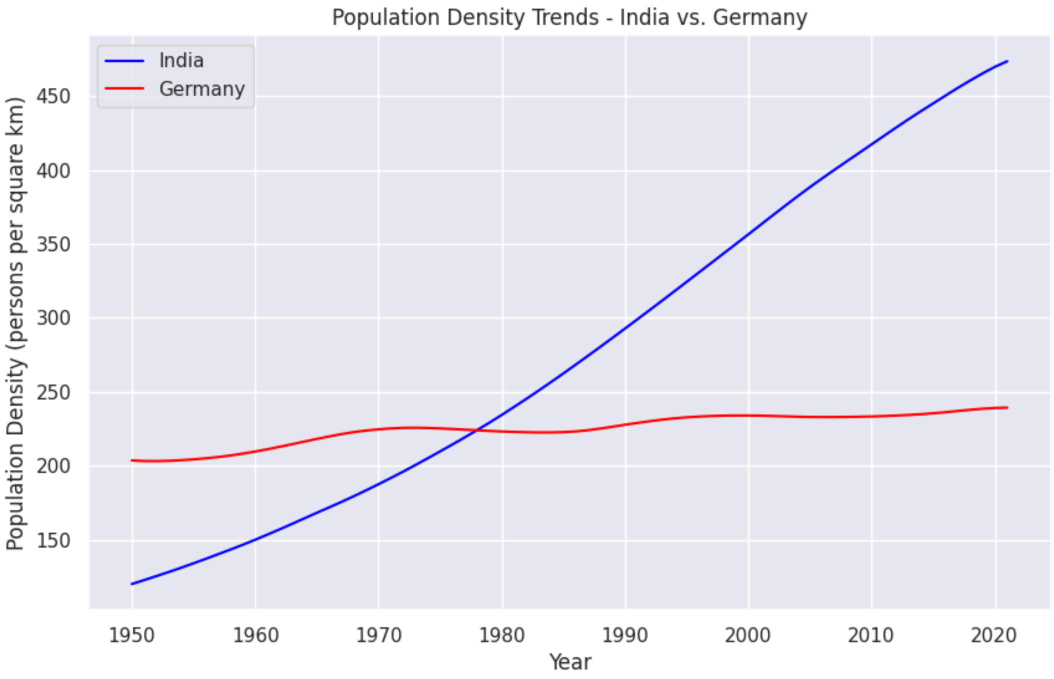
The graph visualizes the male and female population trends over time for India and Germany.

- For India, the male and female population bars are plotted separately, allowing for a clear comparison between male and female populations within India for each year.
- For Germany, the female population bars are plotted first, followed by the male population bars. The male population bars are plotted on top of the female population bars, indicating that the male population values represent an addition to the female population values for each year. This stacking enables visualization of the total population (male + female) for Germany over time.



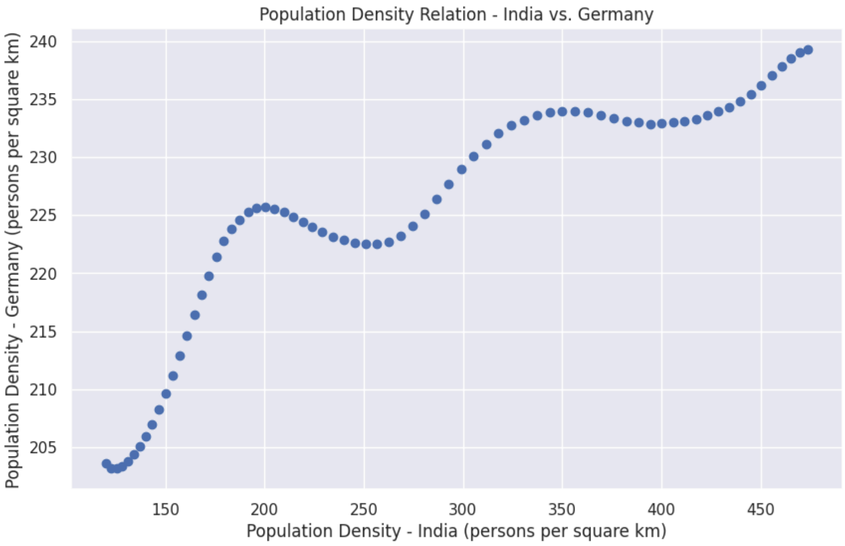
- **Germany -** the sex ratio seems to be changing significantly over the years, starting from around 85 and gradually increasing to less than 100. This indicates that, historically, there were fewer males than females in the population, but over time, this gap has been closing, and the number of males is approaching parity with females.
- **India -** the sex ratio for India remains consistently above 100, indicating that there are more males than females in the population. However, this difference is relatively stable over time compared to the fluctuations seen in Germany.



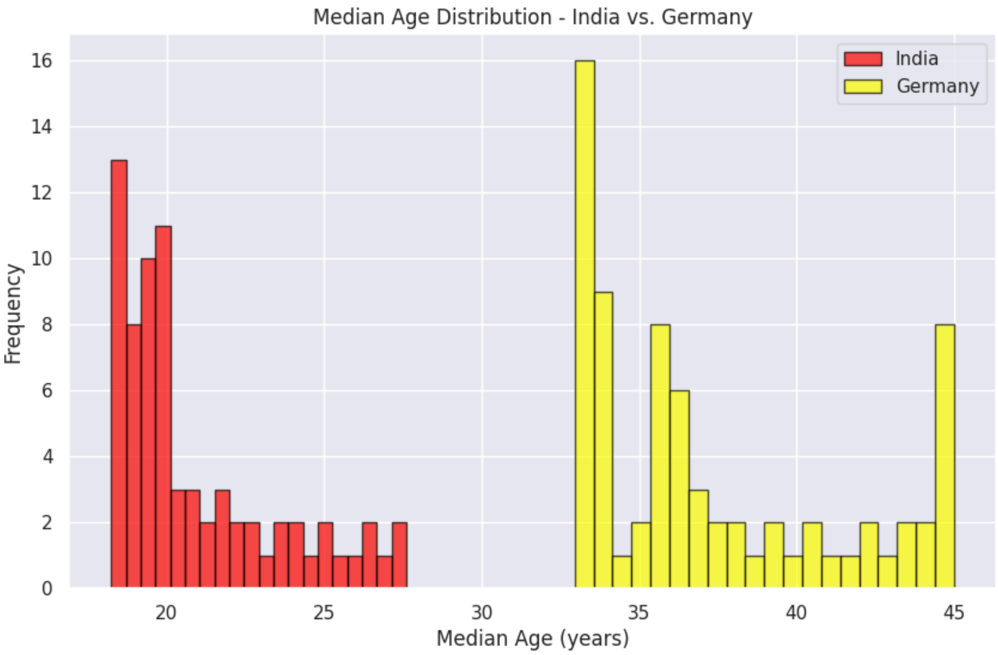


The graph displays the population density trends over the years for India and Germany.

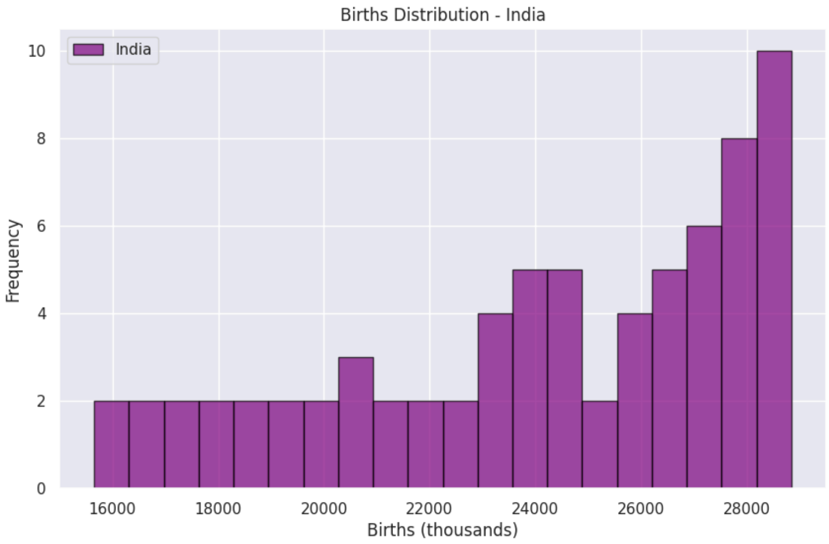
- **Line plot for India** starts at a lower population density and gradually increases over the years, indicating a rise in population density.
- **Line plot for Germany** shows a relatively stable population density trend. The line remains within a narrower range, roughly between 200 to 250 persons per square kilometer, indicating that Germany's population density has been relatively consistent over time.

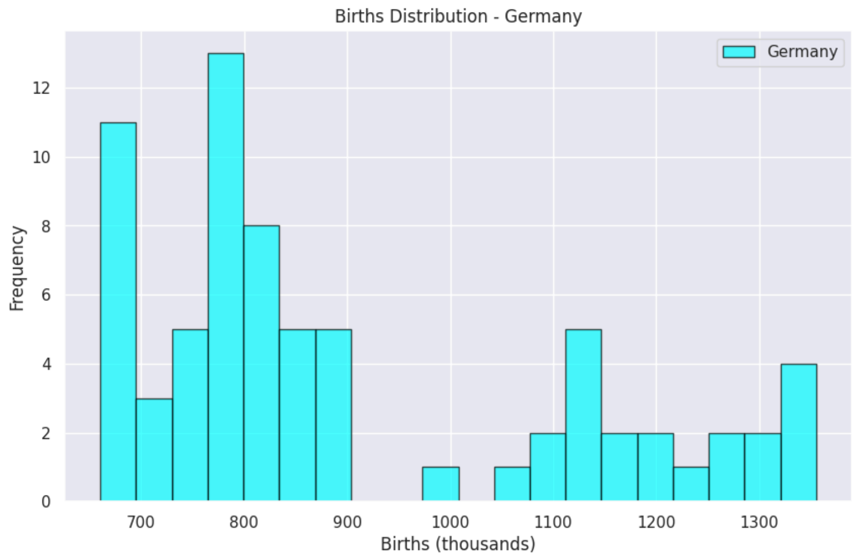


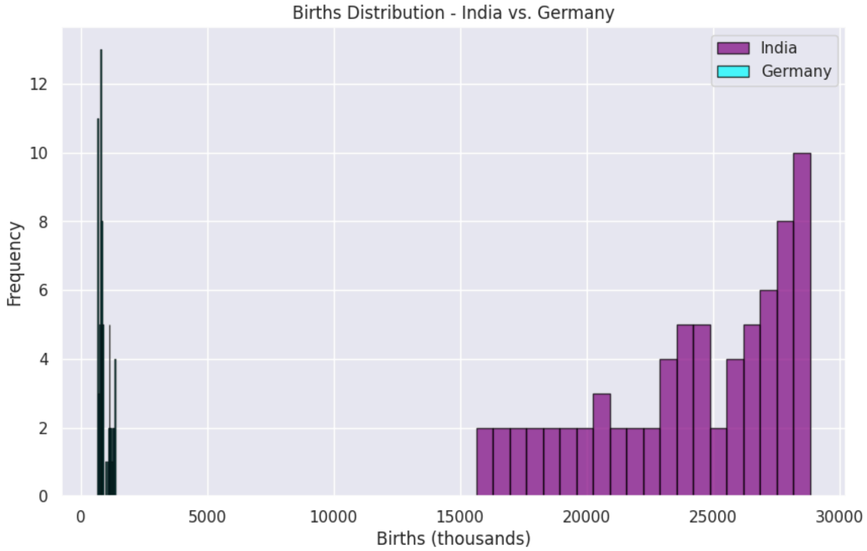
The plot is a scatter plot, where each point represents a pair of population density values for India and Germany. The x-coordinate of a point represents the population density of India for that year, while the y-coordinate represents the population density of Germany for the same year.



- India has a median age of less than 30 years: This can be attributed to factors such as a higher birth rate, lower life expectancy, and a younger population overall.
- Germany has a median age greater than 32 years and less than 45 years: Germany has a lower birth rate, higher life expectancy, and an aging population due to factors such as declining fertility rates and longer life expectancy.

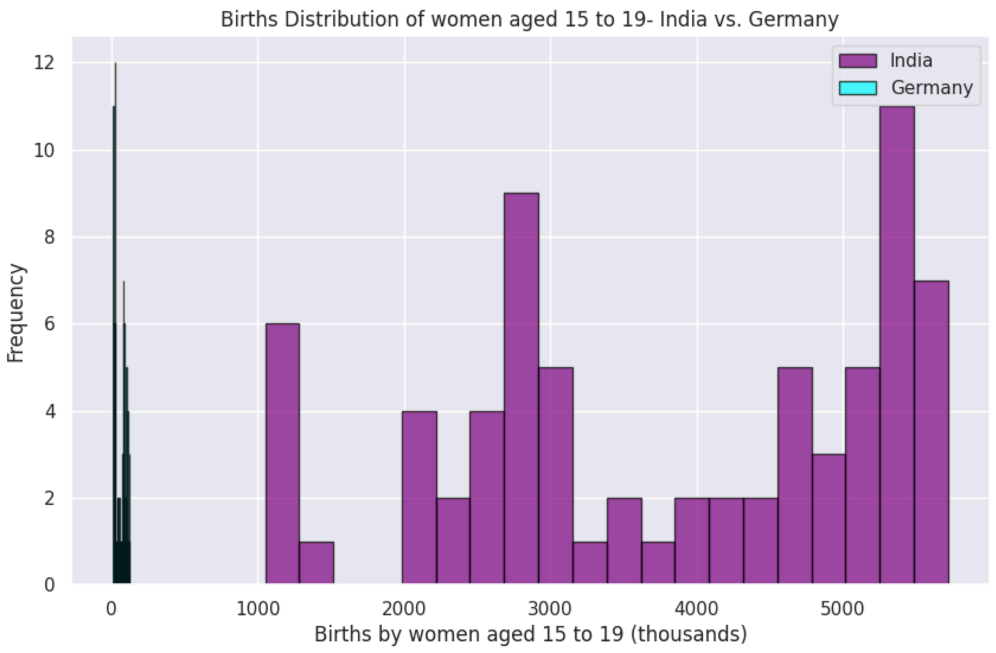


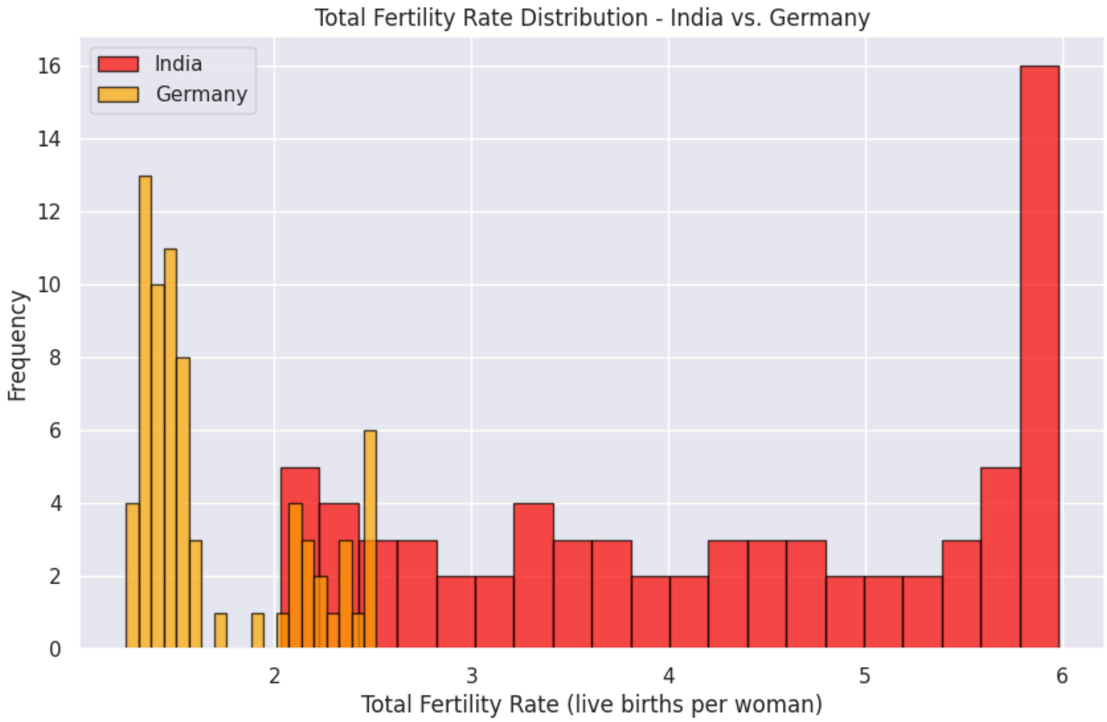




Histogram for India: The birth data for India is spread across a wide range of values, resulting in a broader distribution.

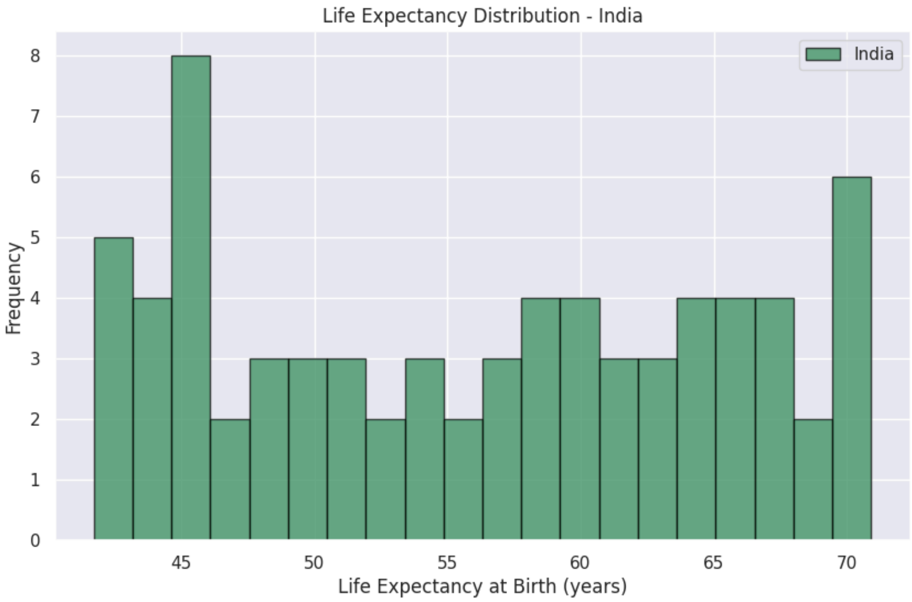
Histogram for Germany: The bars appear to shrink, suggesting that the birth data for Germany is relatively concentrated within a narrower range of values compared to India.

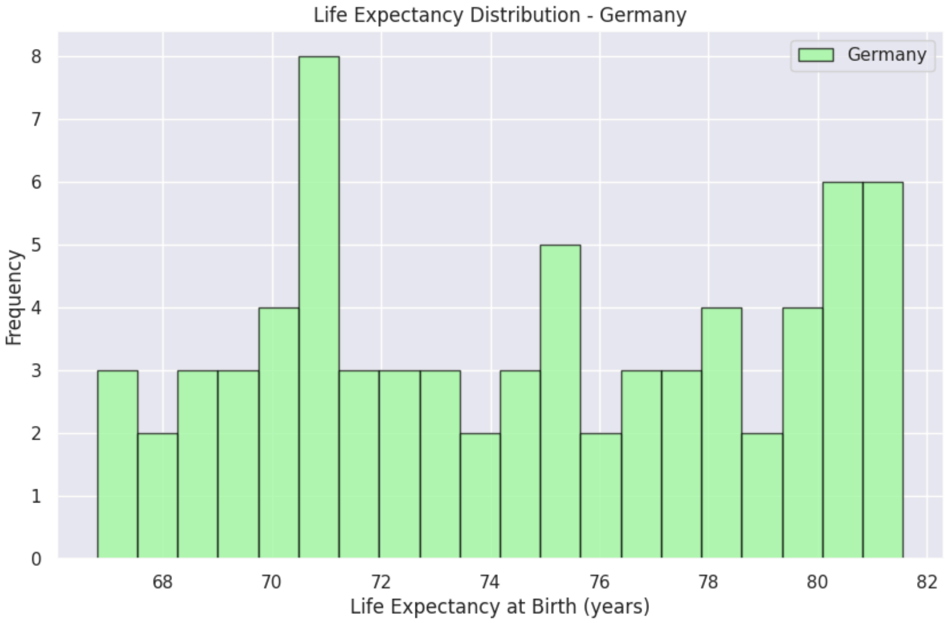


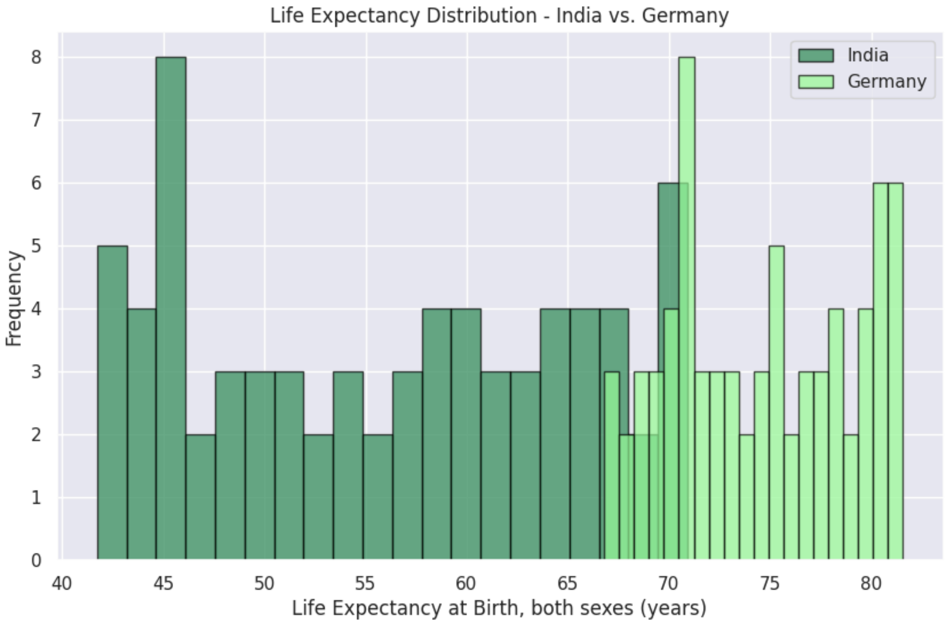


When the fertility rate is between 2-3 (When the fertility rate is 5, it indicates that, on average, each woman is giving birth to five children during her lifetime), both India and Germany have similar fertility rates. However, in the histogram representation, the bars are overlapping because each bar represents a range of fertility rates, and both India and Germany fall within this range.

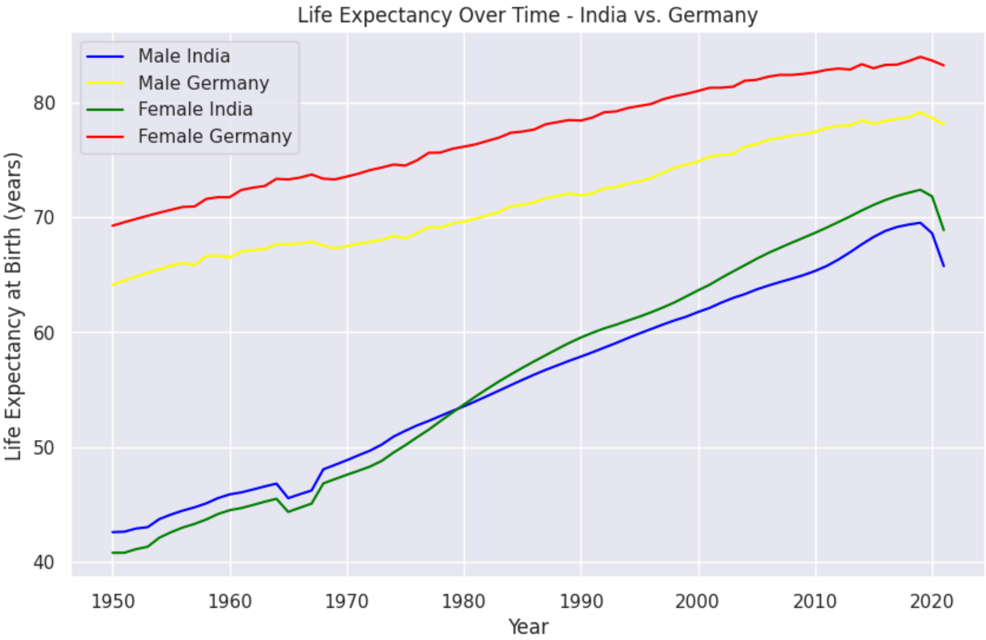
The difference in the width of the bars can be due to the different sample sizes or frequencies within each bin. Germany has a narrower bar compared to India, it suggests that there are fewer data points or a lower frequency of occurrences for fertility rates in that specific range in Germany compared to India.





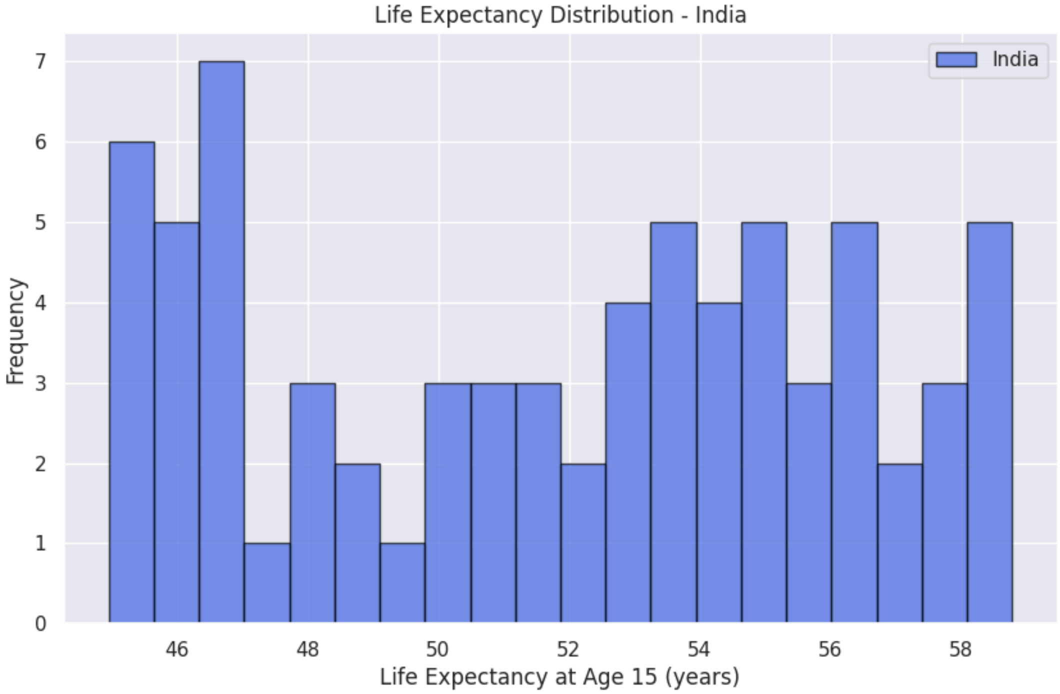


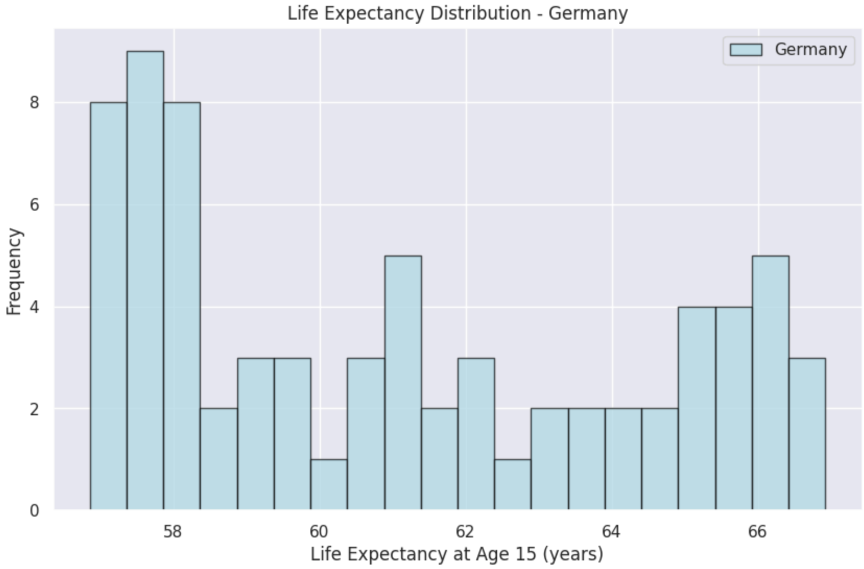
In the graph, we observe that for both India and Germany, the bars are overlapping because a significant portion of the population in both countries has a life expectancy falling within this range.

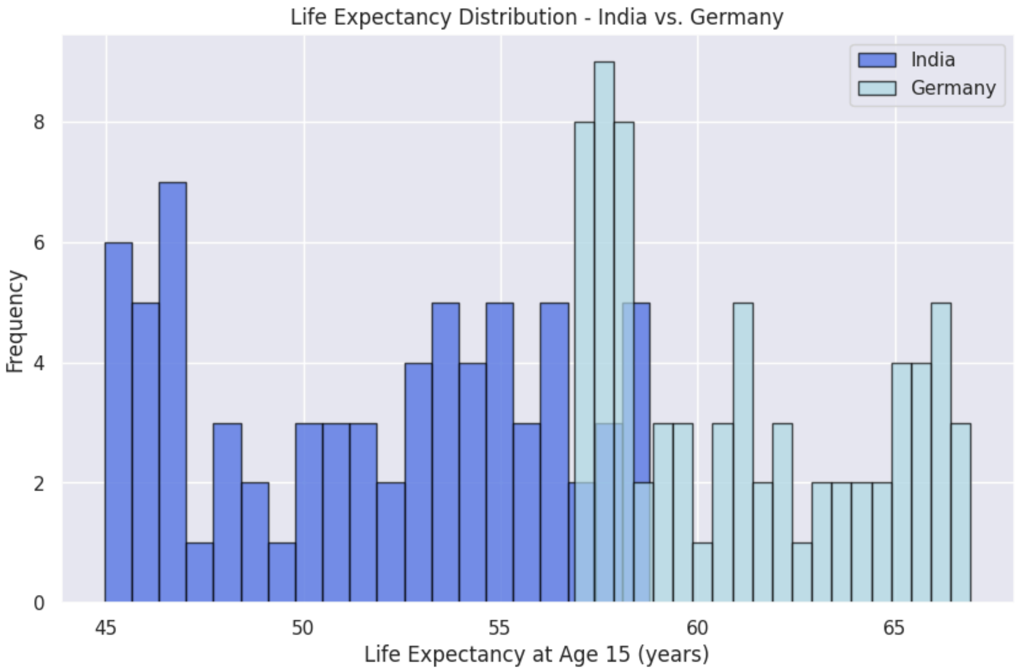


In the graph, the lines represent the trends of life expectancy at birth over the years for males and females in India and Germany.

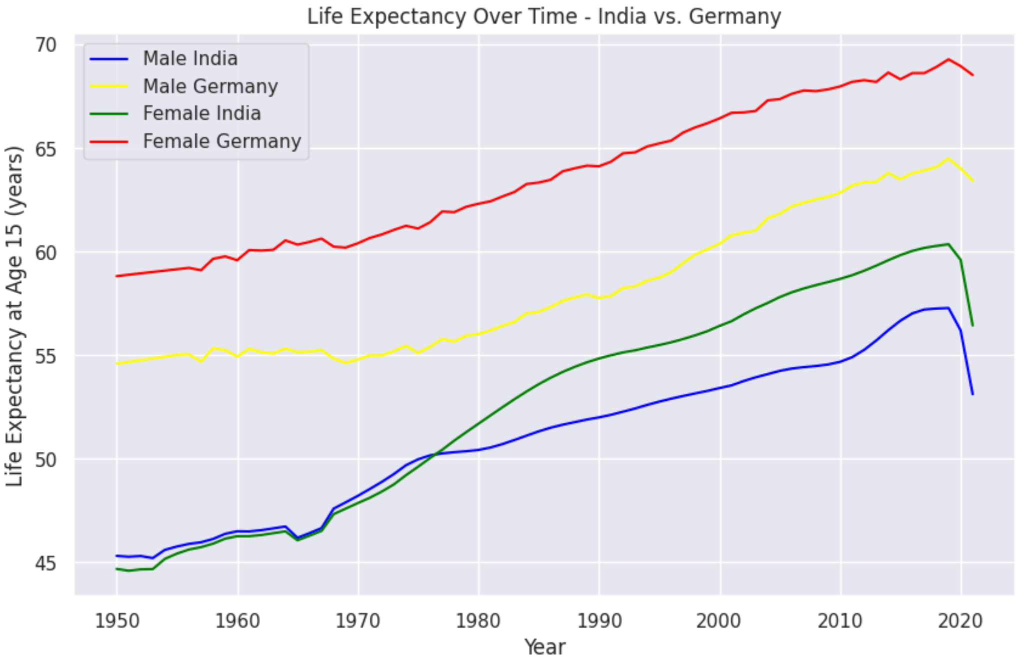
- The observed trend of females having a higher life expectancy than males for both India and Germany is consistent with global demographic patterns, where women typically have longer life expectancies compared to men.
- Initially, male life expectancy in India is higher than in Germany.
- Around the late 1970s or early 1980s, the trend shifts, and male life expectancy in Germany surpassed that of India.





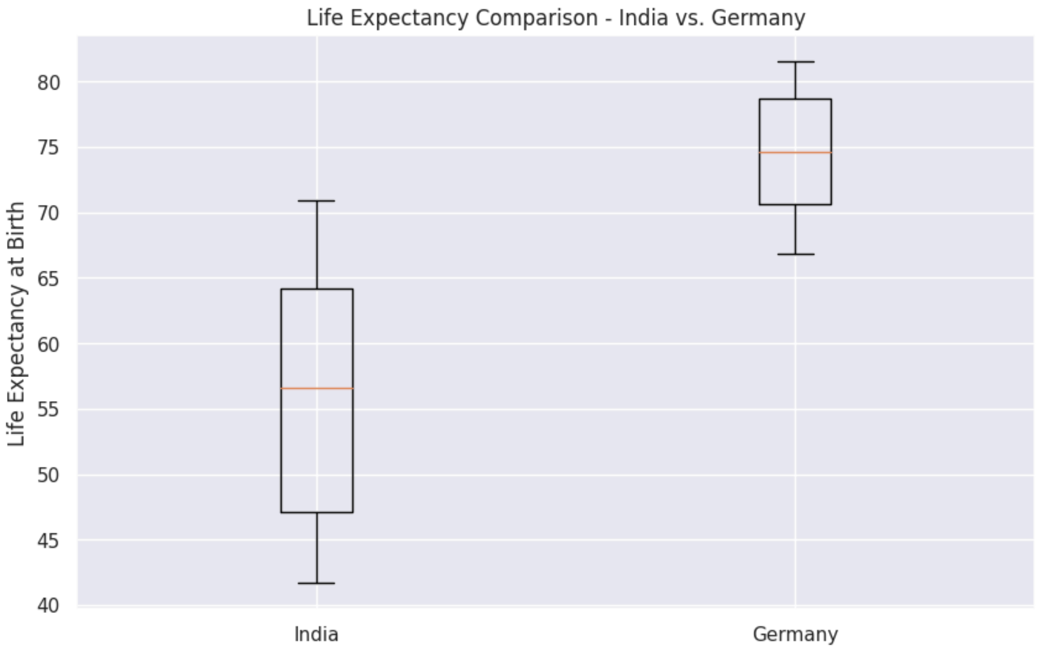


The histogram shows a visual comparison of the life expectancy distributions between India and Germany. It concludes that India and Germany have a significant number of individuals with a life expectancy falling within the range of 56 to 60 years. Therefore, It shows that there is a similarity in the life expectancy distribution within this particular age range between the two countries.



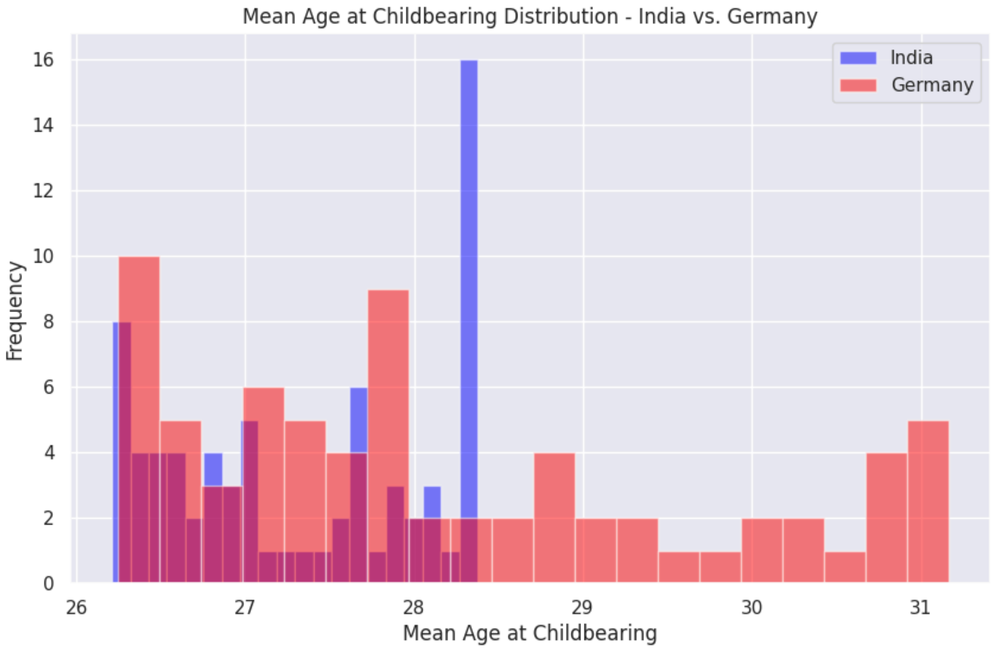
Germany: Both male and female life expectancies in India increase gradually over the years. The gap between male and female life expectancies is relatively stable.

India: The plot shows that initially, male life expectancy in India is higher than female life expectancy. However, as time progresses, female life expectancy catches up and surpasses male life expectancy. This convergence can be attributed to improvements in healthcare, lifestyle changes, and societal advancements that have benefited females more prominently.



The box plot visually represents the distribution of life expectancy at birth for both India and Germany. Each box in the plot represents the interquartile range (IQR) of the data distribution, with the median marked as a line inside the box. The "whiskers" extend to the minimum and maximum values within 1.5 times the IQR from the lower and upper quartiles, respectively. Any outliers beyond the whiskers are represented as individual points.

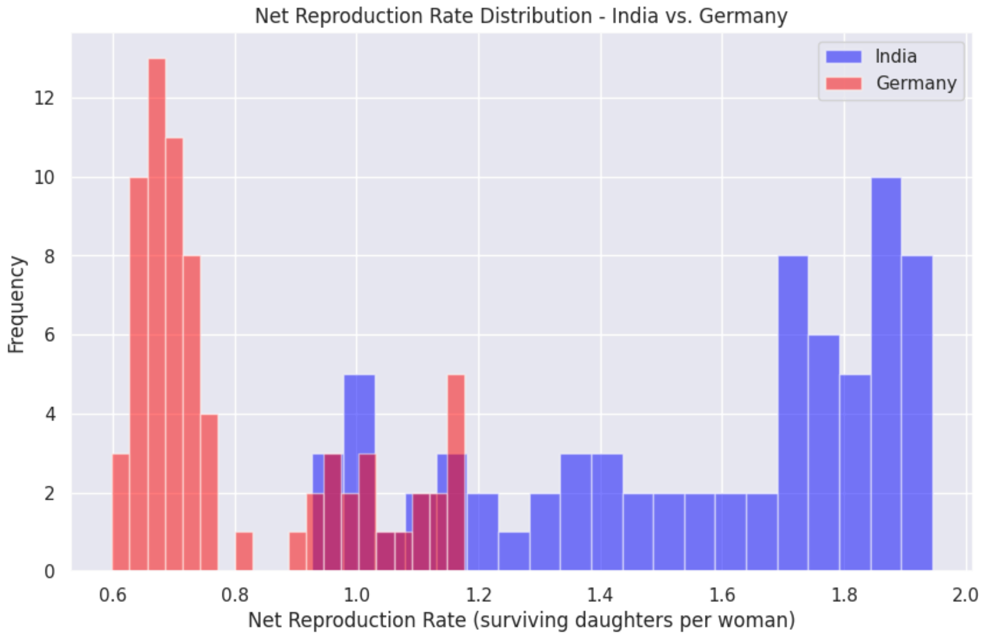
- India's box is larger than Germany's, it suggests greater variability in life expectancy within India.
- Germany's box is positioned higher on the y-axis, it indicates a higher median life expectancy for Germany compared to India.



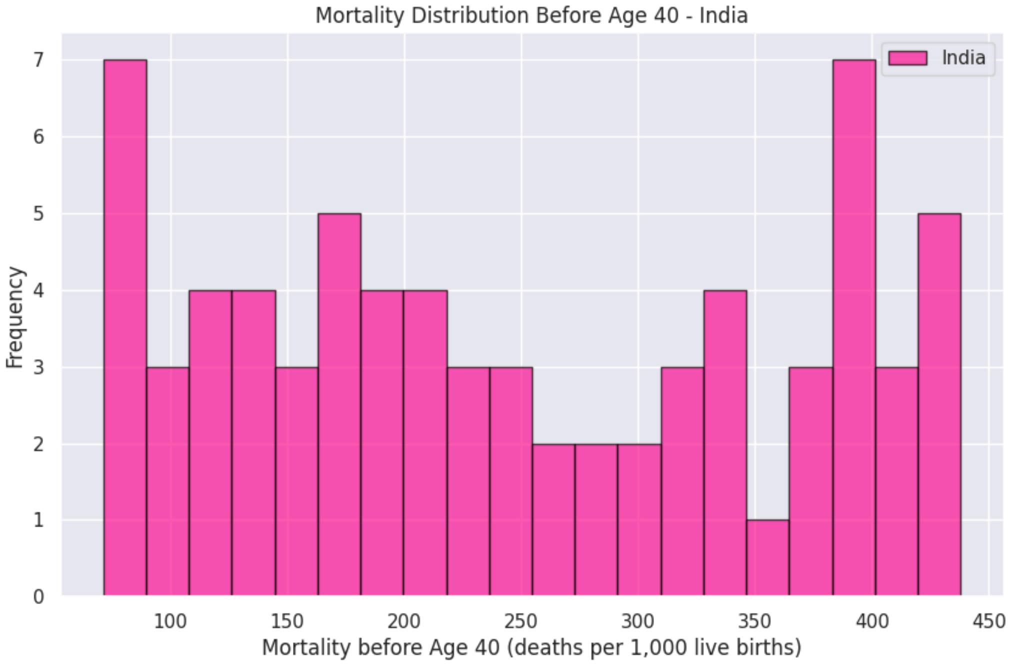
Each bar in the histogram represents a range of values (bin), and the height of the bar indicates the frequency of observations falling within that range.

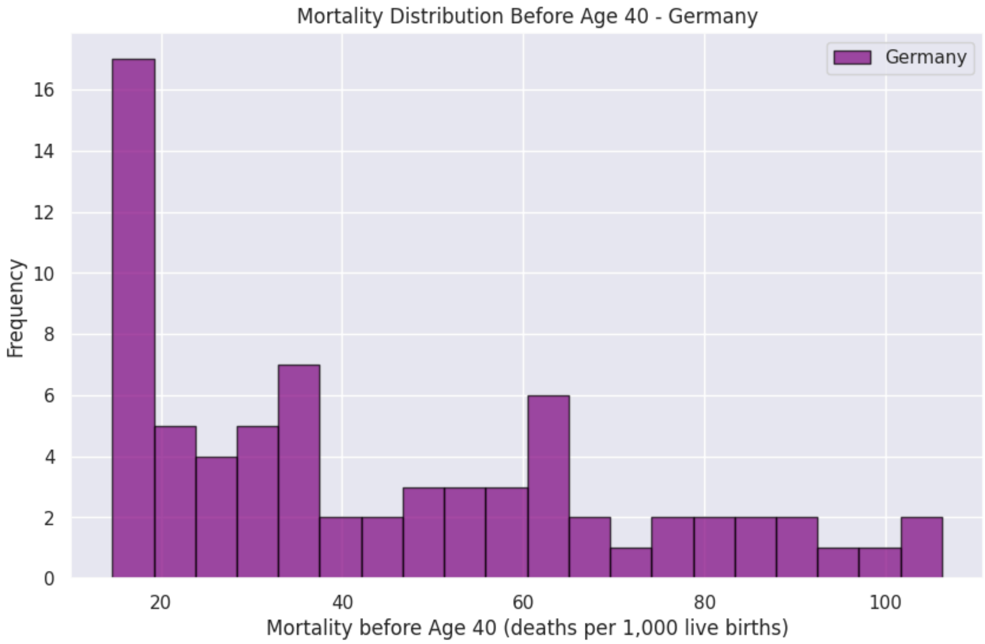
- The thin bars in India's histogram indicate a narrower range of values for the mean age at childbearing compared to Germany. The histogram shows that the mean age at childbearing for India falls within the range of approximately 26 to 28.5 years. This suggests that there is relatively less variability in the age at which women give birth in India.
- The wider bars in Germany's histogram indicate a broader range of values for the mean age at childbearing compared to India. In contrast, the histogram for Germany shows a wider range of approximately 26 to 31 years for the mean age at childbearing. This indicates greater variability in the age at which women give birth in Germany.

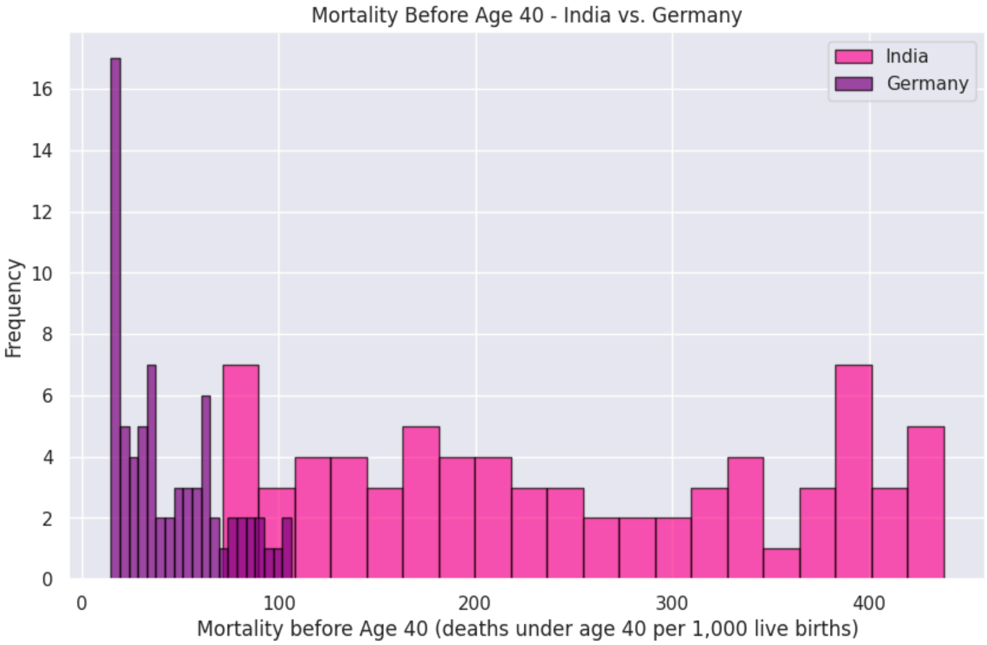
Here is no inherent judgment of better or worse regarding the distribution of mean age at childbearing. However, a wider distribution may indicate greater freedom of choice and access to resources for women in terms of family planning and career development, which could be seen as positive indicators of societal development and gender equality.



- India's bars span from approximately 0.6 to 1.2, while Germany's bars span from approximately 0.9 to 2.0.
- A narrower range of bars suggests a more concentrated distribution of Net Reproduction Rate values, whereas a wider range indicates a broader spread or variability in the data.
- In this case, India's histogram covers a wider range, indicating a broader spread of Net Reproduction Rate values compared to Germany.
- A narrower range may indicate more consistent or stable population growth patterns, which could be considered favorable for long-term planning.

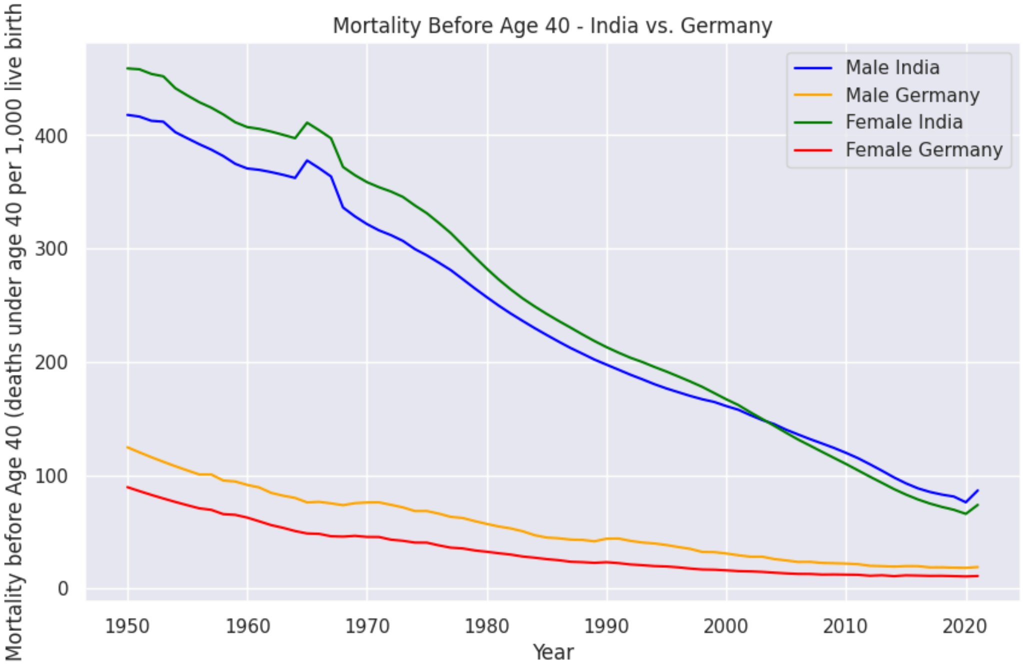






The histogram bars for Germany appear thinner compared to India because the mortality rates in Germany are concentrated within a narrower range (0-100 deaths per 1,000 live births) compared to India (70-450 deaths per 1,000 live births). This indicates that the mortality rates in Germany have less variability compared to India, resulting in thinner bars.

Lower mortality rates before age 40 are generally considered better as they indicate better healthcare access, preventive measures, and overall health conditions. Therefore, in this context, Germany's situation with thinner bars and lower mortality rates (0-100) can be considered better than India's situation with wider bars and higher mortality rates (70-450). However, it's essential to consider various factors such as healthcare infrastructure, socio-economic conditions, and demographic factors when evaluating the healthcare systems of different countries.



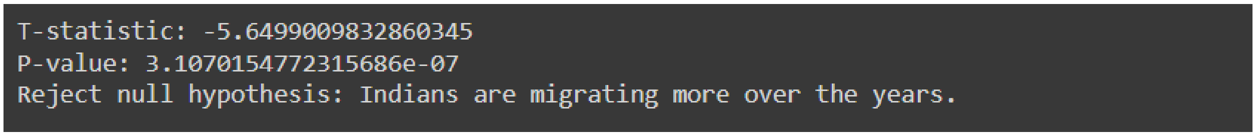
India: For both males and females in India, the mortality rates before age 40 are initially high and gradually decrease over the years. The female mortality rates are consistently higher than male mortality rates throughout the years.

Germany: In Germany, the mortality rates before age 40 for males are initially higher than females. Female mortality rates in Germany also decrease over the years but remain lower than male mortality rates throughout the period.

Comparative Analysis of Key Demographic Indicators between India and Germany with the hypothesis testing

1. Approach and Statistical Analysis on Net Migration:

To examine the net number of migrants between India and Germany, we constructed two hypotheses: a null hypothesis (H0) proposing no significant difference in migration patterns and an alternative hypothesis (H1) suggesting that Indians are migrating more than Germans. We employed a two-sample t-test for independent samples, considering unequal variances, to test these hypotheses. This statistical method was chosen due to its effectiveness in comparing means between two groups, which aligned with our aim of evaluating migration patterns. Additionally, we supplemented our analysis with visualizations such as histograms and box plots to gain further insights into the distribution of migrant numbers.



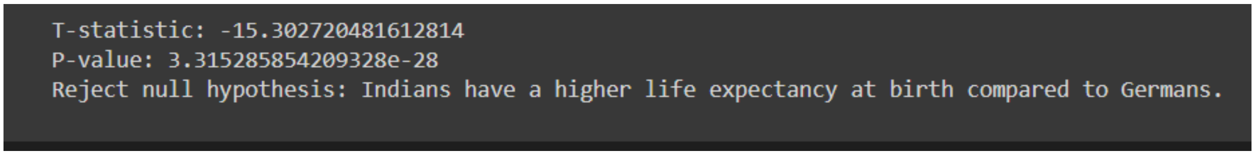
```
T-statistic: -5.6499009832860345
P-value: 3.1070154772315686e-07
Reject null hypothesis: Indians are migrating more over the years.
```

Observations:

The analysis revealed a rejection of the null hypothesis, indicating a significant difference in migration patterns between India and Germany. Moreover, the graphical representations provided additional clarity, illustrating the distribution of migrant numbers and supporting our statistical findings.

2. Approach and Statistical Analysis on Life Expectancy at Birth:

In this section, we investigated the life expectancy at birth between India and Germany. We formulated hypotheses similar to the previous analysis: a null hypothesis (H0) suggesting no significant disparity in life expectancy and an alternative hypothesis (H1) proposing that Indians have a higher life expectancy. Utilizing the two-sample t-test for independent samples with unequal variances, we conducted our statistical analysis. Alongside, we employed visual aids such as line graphs and scatter plots to visually represent life expectancy trends and correlations between variables.



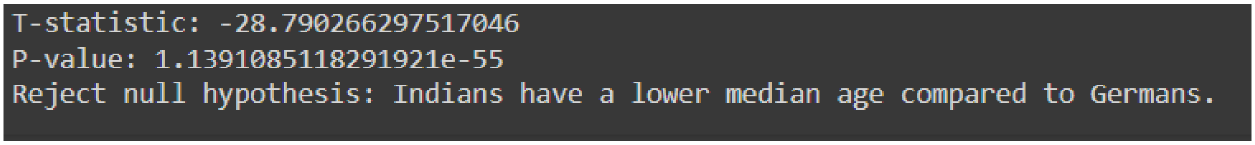
```
T-statistic: -15.302720481612814
P-value: 3.315285854209328e-28
Reject null hypothesis: Indians have a higher life expectancy at birth compared to Germans.
```

Observations:

The statistical analysis supported the rejection of the null hypothesis, indicating a significant difference in life expectancy at birth between Indians and Germans. Furthermore, the graphical representations offered additional insights into the trends and patterns of life expectancy, reinforcing our statistical conclusions.

3. Approach and Statistical Analysis on Median Age:

In this segment, our focus shifted to examining the median age difference between India and Germany. We devised hypotheses akin to previous analyses: a null hypothesis (H0) suggesting no significant distinction in median age and an alternative hypothesis (H1) proposing that Indians have a lower median age. Employing the two-sample t-test for independent samples with unequal variances, we conducted our statistical analysis. Additionally, we supplemented our findings with visualizations such as bar charts and heat maps to depict median age distributions and geographical variations.



```
T-statistic: -28.790266297517046
P-value: 1.1391085118291921e-55
Reject null hypothesis: Indians have a lower median age compared to Germans.
```

Observations:
Contrary to our initial hypothesis, the statistical analysis did not support the rejection of the null hypothesis, indicating no significant difference in median age between Indians and Germans. The graphical representations provided further context, illustrating the distribution of median ages and highlighting potential geographic variations within the populations.

4. CONCLUSION

Our findings highlight significant differences between India and Germany in terms of income levels, healthcare access, and prevalence of chronic diseases among older adults. While India grapples with socio-economic disparities and limited healthcare infrastructure, Germany benefits from higher socio-economic standards and a more robust healthcare system.

The insights garnered from this study underscore the importance of tailored policies and interventions to address the diverse needs of aging populations. By leveraging empirical data and analytical tools, policymakers can formulate evidence-based strategies to promote healthy aging and enhance the overall well-being of older adults in India and globally.

Moving forward, longitudinal studies like LASI will continue to play a crucial role in monitoring trends, evaluating policy interventions, and fostering collaboration across borders to ensure that aging populations receive the support and care they deserve.

5. REFERENCES

1. https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/data-presentation/box-and-whisker-plots.html#:~:text=Definition,than%20one%20boxplot%20per%20graph
2. scatter-plot
3. https://towardsdatascience.com/how-to-perform-a-quadrant-analysis-in-python-9f84d36f8a24
4. hypothesis-testing

Dataset Used

1. https://data.gov.in/catalog/longitudinal-ageing-study-india-lasi
2. https://population.un.org/wpp/Download/Standard/MostUsed/

APPENDIX

```
df.plot(kind='scatter', x='Currently smoking (%)39', y='Chronic lung diseases (%)47', s=32, alpha=0.8)
plt.xlabel('Currently smoking (%)')
plt.ylabel('Chronic lung diseases (%)')
# plt.gca().spines[['top', 'right']].set_visible(False)
plt.show()

# Calculate Pearson correlation coefficient and p-value
corr, p_value = stats.pearsonr(df['Currently smoking (%)39'], df['Chronic lung diseases (%)47'])

# Set significance level
alpha = 0.15

print("Correlation coefficient:", corr)
print("P-value:", p_value)
print("alpha :", alpha )
# null hypothesis = there is no correlation b/w the two
if p_value < alpha:
    print("Reject the null hypothesis. There is significant evidence of a correlation between Currently smoking and Chronic lung diseases.")
else:
    print("Accept the null hypothesis. There is no significant evidence of a correlation between Currently smoking and Chronic lung diseases.")
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress
# Assuming x and y are your data points
x = df['Literate (%)']
y = df['No schooling (%)']
# Perform linear regression
slope, intercept, r_value, p_value, std_err = linregress(x, y)
# Plot the scatter plot
plt.scatter(x, y, label='Data Points')
# Plot the regression line
plt.plot(x, slope * x + intercept, color='red', label='Linear Regression')
# Add labels, legend, etc.
plt.xlabel('Literate (%)')
plt.ylabel('No schooling (%)')
plt.legend()
plt.title('Linear Regression')
# Remove the top and right spines from the plot
plt.gca().spines[['top', 'right']].set_visible(False)
plt.show()
corr, p_value = stats.pearsonr(df['Literate (%)'], df['No schooling (%)'])
```

```
# Set significance level
alpha = 0.05

print("Correlation coefficient:", corr)
print("P-value:", p_value)
print("alpha :", alpha )
# null hypothesis = there is no correlation b/w the two
if p_value < alpha:
    print("Reject the null hypothesis. There is significant evidence of a correlation between Literate (%) and No schooling (%)")
else:
    print("Accept the null hypothesis. There is no significant evidence of a correlation between Literate (%) and No schooling (%)")
```

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

# Assuming your data is already loaded into a pandas DataFrame named 'df'

# Define a threshold for benefit received (e.g., 50%)
benefit_threshold = 50

# Create a new binary variable based on the threshold
df['Benefit Received'] = (df['Receiving Benefits from Indira Gandhi National Old Age Pension Scheme (%)'] >= benefit_threshold).astype(int)

# Plot the scatter plot (optional)
ax = df.plot(kind='scatter', x='Awareness of Indira Gandhi National Old Age Pension Scheme (%)',
             y='Receiving Benefits from Indira Gandhi National Old Age Pension Scheme (%)', s=32, alpha=0.8,
             c=df['Benefit Received'], cmap='viridis')  # Color points based on benefit received

# Set the x and y axis labels with shorter forms
ax.set_xlabel('Awareness (%)')
ax.set_ylabel('Benefits (%)')

# Add legend for color mapping (optional)
handles, labels = ax.get_legend_handles_labels()
plt.legend(handles, ['Benefit Not Received', 'Benefit Received'], title='Benefit Status')

# Remove the top and right spines from the plot
plt.gca().spines[['top', 'right']].set_visible(False)

# Perform logistic regression (on the binary variable)
model = LogisticRegression()
model.fit(df[['Awareness of Indira Gandhi National Old Age Pension Scheme (%)']], df['Benefit Received'])

# Predictions are probabilities between 0 and 1 (benefit received or not)
# Further analysis based on model results (optional)

# ... (e.g., plot decision boundary)

# Show the plot
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np

# Assuming df is your DataFrame containing the data
awareness = df['Awareness of Indira Gandhi National Old Age Pension Scheme (%)']
benefits = df['Receiving Benefits from Indira Gandhi National Old Age Pension Scheme (%)']

# Create a 2D histogram of the data
heatmap, xedges, yedges = np.histogram2d(awareness, benefits, bins=20)

# Plot the heatmap
plt.imshow(heatmap.T, origin='lower', extent=[xedges[0], xedges[-1], yedges[0], yedges[-1]], cmap='inferno')
plt.colorbar(label='Frequency')

# Set the x and y axis labels with shorter forms
plt.xlabel('Awareness of Old Age Pension Scheme (%) (%)')
plt.ylabel('Benefits (%)')

# Show the plot
plt.show()
corr, p_value = stats.pearsonr(df['Awareness of Indira Gandhi National Old Age Pension Scheme (%)'],
df['Receiving Benefits from Indira Gandhi National Old Age Pension Scheme (%)'])

# Set significance level
alpha = 0.05

print("Correlation coefficient:", corr)
print("P-value:", p_value)
print("alpha :", alpha )
# null hypothesis = there is no correlation b/w the two
if p_value < alpha:
   print("Reject the null hypothesis. There is significant evidence of a correlation ")
else:
   print("Accept the null hypothesis. There is no significant evidence of a correlation ")
```

```
sns.jointplot(data=df, x='Persons who consumed any medicine without consulting healthcare provider
(%)94', y='Literate (%)', kind='scatter', height=8)
plt.xlabel('Medicine consumption without consultation (%)')
plt.ylabel('Literate (%)')
plt.title('Relationship between Medicine Consumption and Literacy')
```

```
# Calculate Pearson correlation coefficient
correlation, p_value = stats.pearsonr(df['Persons who consumed any medicine without consulting healthcare provider (%)94'], df['Literate (%)'])
print("p-value:", round(p_value, 4))
plt.tight_layout()
plt.show()

import pandas as pd
import matplotlib.pyplot as plt

# Assuming df is your DataFrame containing the data

# Plot the scatter plot
df.plot(kind='scatter', x='Satisfied with current living arrangement (%)29', y='Yoga practice, meditation, asana and pranayama (%)43', s=32, alpha=0.8)


# Calculate median values for x and y axes
x_median = df['Satisfied with current living arrangement (%)29'].median()
y_median = df['Yoga practice, meditation, asana and pranayama (%)43'].median()

# Divide the data into four quadrants
quadrant_I = df[(df['Satisfied with current living arrangement (%)29'] > x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] > y_median)]
quadrant_II = df[(df['Satisfied with current living arrangement (%)29'] < x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] > y_median)]
quadrant_III = df[(df['Satisfied with current living arrangement (%)29'] < x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] < y_median)]
quadrant_IV = df[(df['Satisfied with current living arrangement (%)29'] > x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] < y_median)]

# Plot shaded regions for each quadrant
plt.fill_between(df['Satisfied with current living arrangement (%)29'], y_median, df['Yoga practice, meditation, asana and pranayama (%)43'], where=(df['Satisfied with current living arrangement (%)29'] > x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] > y_median), color='skyblue', alpha=0.3)
plt.fill_between(df['Satisfied with current living arrangement (%)29'], y_median, df['Yoga practice, meditation, asana and pranayama (%)43'], where=(df['Satisfied with current living arrangement (%)29'] < x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] > y_median), color='orange', alpha=0.3)
plt.fill_between(df['Satisfied with current living arrangement (%)29'], y_median, df['Yoga practice, meditation, asana and pranayama (%)43'], where=(df['Satisfied with current living arrangement (%)29'] < x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] < y_median), color='green', alpha=0.3)
plt.fill_between(df['Satisfied with current living arrangement (%)29'], y_median, df['Yoga practice, meditation, asana and pranayama (%)43'], where=(df['Satisfied with current living arrangement (%)29'] > x_median) & (df['Yoga practice, meditation, asana and pranayama (%)43'] < y_median), color='red', alpha=0.3)
```

```
# Show the plot
plt.gca().spines[['top', 'right']].set_visible(False)
plt.show()

# Print the counts
print("Quadrant I:", len(quadrant_I))
print("Quadrant II:", len(quadrant_II))
print("Quadrant III:", len(quadrant_III))
print("Quadrant IV:", len(quadrant_IV))


import pandas as pd
import matplotlib.pyplot as plt

# Assuming df is your DataFrame containing the data

# Plot the scatter plot
df.plot(kind='scatter', x='Households with water facility inside dwelling/own yard (%)', y='Any water-borne disease (%)63', s=32, alpha=0.8)

# Calculate median values for x and y axes
x_median = df['Households with water facility inside dwelling/own yard (%)'].median()
y_median = df['Any water-borne disease (%)63'].median()

# Divide the data into four quadrants
quadrant_I = df[(df['Households with water facility inside dwelling/own yard (%)'] > x_median) & (df['Any water-borne disease (%)63'] > y_median)]
quadrant_II = df[(df['Households with water facility inside dwelling/own yard (%)'] < x_median) & (df['Any water-borne disease (%)63'] > y_median)]
quadrant_III = df[(df['Households with water facility inside dwelling/own yard (%)'] < x_median) & (df['Any water-borne disease (%)63'] < y_median)]
quadrant_IV = df[(df['Households with water facility inside dwelling/own yard (%)'] > x_median) & (df['Any water-borne disease (%)63'] < y_median)]

# Plot shaded regions for each quadrant
plt.fill_between(df['Households with water facility inside dwelling/own yard (%)'], y_median, df['Any water-borne disease (%)63'], where=(df['Households with water facility inside dwelling/own yard (%)'] > x_median) & (df['Any water-borne disease (%)63'] > y_median), color='skyblue', alpha=0.3)
plt.fill_between(df['Households with water facility inside dwelling/own yard (%)'], y_median, df['Any water-borne disease (%)63'], where=(df['Households with water facility inside dwelling/own yard (%)'] < x_median) & (df['Any water-borne disease (%)63'] > y_median), color='orange', alpha=0.3)
plt.fill_between(df['Households with water facility inside dwelling/own yard (%)'], y_median, df['Any water-borne disease (%)63'], where=(df['Households with water facility inside dwelling/own yard (%)'] < x_median) & (df['Any water-borne disease (%)63'] < y_median), color='green', alpha=0.3)
plt.fill_between(df['Households with water facility inside dwelling/own yard (%)'], y_median, df['Any water-borne disease (%)63'], where=(df['Households with water facility inside dwelling/own yard (%)'] > x_median) & (df['Any water-borne disease (%)63'] < y_median), color='red', alpha=0.3)

# Show the plot
```

```
plt.gca().spines[['top', 'right']].set_visible(False)
plt.show()

# Print the counts
print("Quadrant I:", len(quadrant_I))
print("Quadrant II:", len(quadrant_II))
print("Quadrant III:", len(quadrant_III))
print("Quadrant IV:", len(quadrant_IV))
import matplotlib.pyplot as plt

# Data
mean_expenditure_public_hospitalization = [8877, 2105, 3914, 17131, 8606, 7618, 22285, 4438, 360, 13397,
69347, 12180, 2138, 12279, 27971, 24270, 15255, 10058, 14266, 14885, 3878, 13042, 14927, 22975, 20484, 35850,
9015, 2403, 13389, 8804, 8606, 2255, 4132, 5245, 6535, 17633, 6466]
mean_expenditure_private_hospitalization = [52022, 127099, 34054, 60415, 37131, 32608, 13448, 19848,
48664, 24522, 27795, 34201, 36180, 19302, 93405, 69110, 23835, 125825, 32862, 55847, 26239, 24883, 48706,
33079, 25362, 28825, 38521, 56668, 32270, 23239, 37131, 39242, 35108, 95578, 22949, 42015, 38019]
mean_expenditure_total_hospitalization = [36219, 24341, 27764, 24601, 21528, 25037, 21120, 11916, 22912,
21284, 57019, 20730, 26484, 17456, 48486, 31941, 21826, 102840, 25053, 33694, 17095, 22771, 25272, 26598,
23000, 32812, 18042, 26734, 27021, 16542, 21528, 28176, 27319, 17455, 17619, 34812, 15824]

# Plot
plt.figure(figsize=(10, 6))
plt.plot(mean_expenditure_public_hospitalization, label='Public Facility', marker='o')
plt.plot(mean_expenditure_private_hospitalization, label='Private Facility', marker='s')
plt.plot(mean_expenditure_total_hospitalization, label='Total', marker='^')

# Add labels and title
plt.xlabel('Observation')
plt.ylabel('Mean Expenditure (INR)')
plt.title('Mean Expenditure on Last Hospitalization by Type of Facility')
plt.legend()

# Show plot
plt.grid(True)
plt.show()


# Calculate the median values for x and y variables
median_x = df['High Cholesterol (%)'].median()
median_y = df['Overweight by Anthropometric Indicators(%)'].median()

# Perform quadrant analysis
quadrant1 = df[(df['High Cholesterol (%)'] >= median_x) & (df['Overweight by Anthropometric
Indicators(%)'] >= median_y)]
quadrant2 = df[(df['High Cholesterol (%)'] < median_x) & (df['Overweight by Anthropometric
Indicators(%)'] >= median_y)]
quadrant3 = df[(df['High Cholesterol (%)'] < median_x) & (df['Overweight by Anthropometric
Indicators(%)'] < median_y)]
```

```
quadrant4 = df[(df['High Cholesterol (%)'] >= median_x) & (df['Overweight by Anthropometric Indicators(%)'] < median_y)]

# Count the number of data points in each quadrant
count_quadrant1 = len(quadrant1)
count_quadrant2 = len(quadrant2)
count_quadrant3 = len(quadrant3)
count_quadrant4 = len(quadrant4)

# Print the count of data points in each quadrant
print("Number of data points in Quadrant 1:", count_quadrant1)
print("Number of data points in Quadrant 2:", count_quadrant2)
print("Number of data points in Quadrant 3:", count_quadrant3)
print("Number of data points in Quadrant 4:", count_quadrant4)

# Visualize the quadrants on the scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(df['High Cholesterol (%)'], df['Overweight by Anthropometric Indicators(%)'], s=32, alpha=0.8)

# Plot median lines
plt.axvline(median_x, color='r', linestyle='--', linewidth=1)
plt.axhline(median_y, color='r', linestyle='--', linewidth=1)

# Add annotations for quadrants
plt.text(median_x, median_y, 'Q1', fontsize=12, ha='left', va='bottom')
plt.text(median_x, median_y, 'Q2', fontsize=12, ha='right', va='bottom')
plt.text(median_x, median_y, 'Q3', fontsize=12, ha='right', va='top')
plt.text(median_x, median_y, 'Q4', fontsize=12, ha='left', va='top')

plt.xlabel('High Cholesterol (%)')
plt.ylabel('Overweight by Anthropometric Indicators(%)')
plt.title('Quadrant Analysis')

plt.grid(True)
plt.show()

# Calculate the median values for x and y variables
median_x = df['Per Capita Annual Household Income (in INR) 12'].median()
median_y = df['Households covered by any health insurance (%) 17'].median()

# Perform quadrant analysis
quadrant1 = df[(df['Per Capita Annual Household Income (in INR) 12'] >= median_x) & (df['Households covered by any health insurance (%) 17'] >= median_y)]
quadrant2 = df[(df['Per Capita Annual Household Income (in INR) 12'] < median_x) & (df['Households covered by any health insurance (%) 17'] >= median_y)]
quadrant3 = df[(df['Per Capita Annual Household Income (in INR) 12'] < median_x) & (df['Households covered by any health insurance (%) 17'] < median_y)]
quadrant4 = df[(df['Per Capita Annual Household Income (in INR) 12'] >= median_x) & (df['Households covered by any health insurance (%) 17'] < median_y)]
```

```
# Count the number of data points in each quadrant
count_quadrant1 = len(quadrant1)
count_quadrant2 = len(quadrant2)
count_quadrant3 = len(quadrant3)
count_quadrant4 = len(quadrant4)

# Print the count of data points in each quadrant
print("Number of data points in Quadrant 1:", count_quadrant1)
print("Number of data points in Quadrant 2:", count_quadrant2)
print("Number of data points in Quadrant 3:", count_quadrant3)
print("Number of data points in Quadrant 4:", count_quadrant4)

# Visualize the quadrants on the scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(df['Per Capita Annual Household Income (in INR) 12'], df['Households covered by any health insurance (%) 17'], s=32, alpha=0.8)

# Plot median lines
plt.axvline(median_x, color='r', linestyle='--', linewidth=1)
plt.axhline(median_y, color='r', linestyle='--', linewidth=1)

# Add annotations for quadrants
plt.text(median_x, median_y, 'Q1', fontsize=12, ha='left', va='bottom')
plt.text(median_x, median_y, 'Q2', fontsize=12, ha='right', va='bottom')
plt.text(median_x, median_y, 'Q3', fontsize=12, ha='right', va='top')
plt.text(median_x, median_y, 'Q4', fontsize=12, ha='left', va='top')

plt.xlabel('Per Capita Annual Household Income (in INR)')
plt.ylabel('Households covered by any health insurance (%)')
plt.title('Quadrant Analysis')

plt.grid(True)
plt.show()


# Sort the DataFrame by 'Literate (%)'
sorted_df = df.sort_values(by='Literate (%)', ascending=False)

# Create a scatter plot using the sorted DataFrame
plt.scatter(x=sorted_df['Literate (%)'], y=sorted_df[' Sex Ratio (Females per 1000 Males ) All ages'], s=32, alpha=0.8)

# Calculate statistical measures
mean_literate = np.mean(sorted_df['Literate (%)'])
median_literate = np.median(sorted_df['Literate (%)'])
std_literate = np.std(sorted_df['Literate (%)'])

mean_sex_ratio = np.mean(sorted_df[' Sex Ratio (Females per 1000 Males ) All ages'])
```

```
median_sex_ratio = np.median(sorted_df[' Sex Ratio (Females per 1000 Males ) All ages'])
std_sex_ratio = np.std(sorted_df[' Sex Ratio (Females per 1000 Males ) All ages'])

# Add mean lines to the plot
plt.axvline(x=mean_literate, color='r', linestyle='--', label=f'Mean Literate (%): {mean_literate:.2f}')
plt.axhline(y=mean_sex_ratio, color='b', linestyle='--', label=f'Mean Sex Ratio: {mean_sex_ratio:.2f}')

# Remove the top and right spines from the plot
plt.gca().spines[['top', 'right']].set_visible(False)

# Show statistical measures
print("Literate (%) Statistics:")
print(f"Mean: {mean_literate:.2f}")
print(f"Median: {median_literate:.2f}")
print(f"Standard Deviation: {std_literate:.2f}")
print("\nSex Ratio Statistics:")
print(f"Mean: {mean_sex_ratio:.2f}")
print(f"Median: {median_sex_ratio:.2f}")
print(f"Standard Deviation: {std_sex_ratio:.2f}")

# Add labels and legend
plt.xlabel('Literate (%)')
plt.ylabel('Sex Ratio (Females per 1000 Males)')
plt.title('Scatter Plot with Mean Lines')
plt.legend()

# Show the plot
plt.show()

# Sort the DataFrame by 'Sex Ratio (Females per 1000 Males) All ages' column in descending order
sorted_df = df.sort_values(by=' Sex Ratio (Females per 1000 Males ) All ages', ascending=True)

# Take the top 10 rows
top_10_df = sorted_df.head(5)

# Plot the scatter plot
top_10_df.plot(kind='scatter', x=df.columns[0], y=' Sex Ratio (Females per 1000 Males ) All ages', s=32,
alpha=0.8)

# Remove top and right spines
plt.gca().spines[['top', 'right']].set_visible(False)

# Set x-axis label
plt.xlabel("States and UTs")

# Show the plot
plt.show()
# Sort the DataFrame by 'Sex Ratio (Females per 1000 Males) All ages' column in descending order
sorted_df = df.sort_values(by=' Sex Ratio (Females per 1000 Males ) All ages', ascending=False)
```

```
# Take the top 10 rows
top_10_df = sorted_df.head(5)

# Plot the scatter plot
ax = top_10_df.plot(kind='scatter', x=df.columns[0], y=' Sex Ratio (Females per 1000 Males ) All ages', s=32,
alpha=0.8)

# Remove top and right spines
ax.spines[['top', 'right']].set_visible(False)

# Set x-axis label
plt.xlabel("States and UTs")

# Show the plot
plt.show()
import matplotlib.pyplot as plt

# Plotting hexbin plot
plt.figure(figsize=(8, 6))
plt.hexbin(df['Households with improved sanitation (%) 4'], df['Malaria (%)'], gridsize=20, cmap='viridis',
alpha=0.8)

# Adding labels and title
plt.xlabel('Improved sanitation (%)')
plt.ylabel('Malaria (%)')
plt.title('Relationship between Improved Sanitation and Malaria')

# Adding a colorbar for interpretation
plt.colorbar(label='Density')

# Removing top and right spines
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

# Showing the plot
plt.show()

import seaborn as sns
# Specify the data and the axes
data = df[['Agricultural and allied activities21', 'Per Capita Annual Household Income (in INR) 12']]

# Create a heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

# Add labels and title
plt.xlabel('Agricultural and allied activities')
plt.ylabel('Per Capita Annual Household Income')
plt.title('Heatmap of Correlation')

# Show the plot
plt.show()
import matplotlib.pyplot as plt
import pandas as pd
# Create a block heatmap
plt.hist2d(df['Agricultural and allied activities21'],
           df['Per Capita Annual Household Income (in INR) 12'],
           bins=20, cmap='inferno')

plt.colorbar(label='Count')  # Add a colorbar for reference

plt.xlabel('Agricultural and allied activities')
plt.ylabel('Per Capita Annual Household Income')

# Remove the top and right spines from the plot
plt.gca().spines[['top', 'right']].set_visible(False)

plt.show()

import seaborn as sns

# Assuming df is your DataFrame containing the data
# Specify the data and the axes
data = df[['Persons seeking job (%)26', 'Ever worked (%)19']]

# Create a heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

# Add labels and title
plt.xlabel('Persons seeking job (%)')
plt.ylabel('Ever worked (%)')
plt.title('Heatmap of Correlation')

# Show the plot
plt.show()

# Plot the scatter plot using df.plot()
df.plot(kind='scatter', x='Persons seeking job (%)26', y='Ever worked (%)19', s=32, alpha=0.8)

# Remove the top and right spines from the plot
plt.gca().spines[['top', 'right']].set_visible(False)

# Show the plot
plt.show()

# Plot the scatter plot using df.plot()
```

```
ax = df.plot(kind='scatter', x='Households covered by Central Government Health Scheme
(CGHS)/Employee State Insurance Scheme (ESIS) (%)', y='Health insurance coverage (%) ', s=32,
alpha=0.8)

# Set the x and y axis labels with shorter forms
ax.set_xlabel('Central Govt. Health/Employee State Insurance Scheme(%)')
ax.set_ylabel('Health insurance coverage(%)')

# Remove the top and right spines from the plot
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Show the plot
plt.show()

# Plot the scatter plot using df.plot()
ax = df.plot(kind='scatter', x='Household Monthly Per Capita Consumption Expenditure (MPCE) in INR 10',
y='Household Per Capita Food Expenditure as a share of MPCE (%)', s=32, alpha=0.8)

# Set the x and y axis labels with shorter forms
ax.set_xlabel('Household Monthly Per Capita Consumption Expenditure')
ax.set_ylabel('Household Per Capita Food Expenditure')

# Remove the top and right spines from the plot
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Show the plot
plt.show()

import pandas as pd
import matplotlib.pyplot as plt

# Assuming df is your DataFrame containing the data

# Plot the scatter plot
ax = df.plot(kind='scatter', x='Bone/Joint diseases (%)48', y='60-69 Years of Household population (%)',
s=32, alpha=0.8, label='60-69 Years')
df.plot(kind='scatter', x='Bone/Joint diseases (%)48', y='70-79 Years of Household population (%)', s=32,
alpha=0.8, ax=ax, color='orange', label='70-79 Years')
df.plot(kind='scatter', x='Bone/Joint diseases (%)48', y='80+ Years of Household population (%)', s=32,
alpha=0.8, ax=ax, color='green', label='80+ Years')

# Set the x and y axis labels with shorter forms
ax.set_xlabel('Bone/Joint diseases (%)')
ax.set_ylabel('Household population (%)')

# Calculate median values for x and y axes
x_median = df['Bone/Joint diseases (%)48'].median()
```

```
y_median_60_69 = df['60-69 Years of Household population (%)'].median()
y_median_70_79 = df['70-79 Years of Household population (%)'].median()
y_median_80_plus = df['80+ Years of Household population (%)'].median()

# Divide the data into four quadrants
quadrant_I = df[(df['Bone/Joint diseases (%)48'] > x_median) & (df['60-69 Years of Household population (%)'] > y_median_60_69)]
quadrant_II = df[(df['Bone/Joint diseases (%)48'] < x_median) & (df['60-69 Years of Household population (%)'] > y_median_60_69)]
quadrant_III = df[(df['Bone/Joint diseases (%)48'] < x_median) & (df['60-69 Years of Household population (%)'] < y_median_60_69)]
quadrant_IV = df[(df['Bone/Joint diseases (%)48'] > x_median) & (df['60-69 Years of Household population (%)'] < y_median_60_69)]

# Count the number of data points in each quadrant
count_quadrant_I = len(quadrant_I)
count_quadrant_II = len(quadrant_II)
count_quadrant_III = len(quadrant_III)
count_quadrant_IV = len(quadrant_IV)

# Plot shaded regions for each quadrant
ax.fill_betweenx(df['60-69 Years of Household population (%)'], x_median, df['Bone/Joint diseases (%)48'], where=(df['Bone/Joint diseases (%)48'] > x_median) & (df['60-69 Years of Household population (%)'] > y_median_60_69), color='skyblue', alpha=0.3)
ax.fill_betweenx(df['60-69 Years of Household population (%)'], x_median, df['Bone/Joint diseases (%)48'], where=(df['Bone/Joint diseases (%)48'] < x_median) & (df['60-69 Years of Household population (%)'] > y_median_60_69), color='orange', alpha=0.3)
ax.fill_betweenx(df['60-69 Years of Household population (%)'], x_median, df['Bone/Joint diseases (%)48'], where=(df['Bone/Joint diseases (%)48'] < x_median) & (df['60-69 Years of Household population (%)'] < y_median_60_69), color='green', alpha=0.3)
ax.fill_betweenx(df['60-69 Years of Household population (%)'], x_median, df['Bone/Joint diseases (%)48'], where=(df['Bone/Joint diseases (%)48'] > x_median) & (df['60-69 Years of Household population (%)'] < y_median_60_69), color='red', alpha=0.3)

# Remove the top and right spines from the plot
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Show the legend
plt.legend()

# Show the plot
plt.show()

# Print the counts
print("Quadrant I:", count_quadrant_I)
print("Quadrant II:", count_quadrant_II)
print("Quadrant III:", count_quadrant_III)
print("Quadrant IV:", count_quadrant_IV)
```

```
#India Vs Germany
import pandas as pd #used for data manipulation and analysis.
import seaborn as sns #used for statistical data visualization.
import matplotlib.pyplot as plt #provides a MATLAB-like interface for creating plots and visualizations.
sns.set(color_codes = True) #seaborn interprets color codes in a consistent manner across different functions and plots.

from google.colab import files
df = pd.read_csv('/content/drive/MyDrive/India_Germany.csv')
df

# Now you can work with the DataFrame (e.g., view the first few rows)
print(df.head(3))

# Filter data for India
india_data = df[df['Region, subregion, country or area *'] == 'India']

# Filter data for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Calculate total population for India and Germany
total_population_india = india_data['Total Population, as of 1 January (thousands)'].sum()
total_population_germany = germany_data['Total Population, as of 1 January (thousands)'].sum()

# Create labels for the pie chart
labels = ['India', 'Germany']

# Create population data for the pie chart
population = [total_population_india, total_population_germany]

# Define custom colors
colors = ['seagreen', 'palegreen']

# Plotting the pie chart with custom colors
plt.figure(figsize=(8, 8))
plt.pie(population, labels=labels, autopct='%1.1f%%', startangle=140, colors=colors)
plt.title('Total Population Distribution - India vs. Germany')
plt.show()

# Plotting the total population trends for India and Germany over the years
plt.figure(figsize=(10, 6))

# Plotting population trend for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
         df['Total Population, as of 1 January (thousands)'][df['Region, subregion, country or area *'] == 'India'],
         label='India', color='blue')
```

```
# Plotting population trend for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Total Population, as of 1 January (thousands)'][df['Region, subregion, country or area *'] ==
'Germany'],
        label='Germany', color='red')

# Adding title and labels
plt.title('Total Population Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Total Population (thousands)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()

# Creating a new figure with specified size
plt.figure(figsize=(10, 6))

# Filtering data for India
india_data = df[df['Region, subregion, country or area *'] == 'India']

# Filtering data for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Plotting bar plot for India
plt.bar(india_data['Year'], india_data['Total Population, as of 1 January (thousands)'], label='India')

# Plotting bar plot for Germany
plt.bar(germany_data['Year'], germany_data['Total Population, as of 1 January (thousands)'],
label='Germany')

# Adding title and labels
plt.title('Total Population Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Total Population (thousands)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()
```

```
# Creating a new figure with specified size
plt.figure(figsize=(10, 6))

# Filtering data for India
india_data = df[df['Region, subregion, country or area *'] == 'India']

# Filtering data for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Plotting a stacked bar plot for India and Germany
plt.bar(india_data['Year'], india_data['Total Population, as of 1 January (thousands)'], label='India')
plt.bar(germany_data['Year'], germany_data['Total Population, as of 1 January (thousands)'],
bottom=india_data['Total Population, as of 1 January (thousands)'], label='Germany')

# Adding title and labels
plt.title('Total Population Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Total Population (thousands)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()

# Plotting the total population trends for India and Germany over the years
plt.figure(figsize=(10, 6))

# Plotting male population for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Male Population, as of 1 July (thousands)'][df['Region, subregion, country or area *'] == 'India'],
        label='Male India', color='blue')

# Plotting male population for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Male Population, as of 1 July (thousands)'][df['Region, subregion, country or area *'] ==
'Germany'],
        label='Male Germany', color='yellow')

# Plotting female population for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Female Population, as of 1 July (thousands)'][df['Region, subregion, country or area *'] == 'India'],
        label='Female India', color='green')

# Plotting female population for Germany
```

```
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Female Population, as of 1 July (thousands)'][df['Region, subregion, country or area *'] ==
'Germany'],
        label='Female Germany', color='red')

# Adding title and labels
plt.title('Male & Female Population Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Population (thousands)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()

# Creating a new figure with specified size
plt.figure(figsize=(10, 6))

# Filtering data for India
india_data = df[df['Region, subregion, country or area *'] == 'India']

# Filtering data for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Plotting bar plot for India[Male]
plt.bar(india_data['Year'], india_data['Male Population, as of 1 July (thousands)'], label='Male India')

# Plotting bar plot for India[Female]
plt.bar(india_data['Year'], india_data['Female Population, as of 1 July (thousands)'], label='Female India')

# Plotting bar plot for Germany[Female]
plt.bar(germany_data['Year'], germany_data['Female Population, as of 1 July (thousands)'], label='Female
Germany')

# Plotting bar plot for Germany[Male] on top of Female
plt.bar(germany_data['Year'], germany_data['Male Population, as of 1 July (thousands)'], label='Male
Germany')

# Adding title and labels
plt.title('Male & Female Population Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Population (thousands)')

# Adding legend
plt.legend()
```

```
# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()


# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Line plot
plt.figure(figsize=(10, 6))
plt.plot(india_data['Year'], india_data['Population Sex Ratio, as of 1 July (males per 100 females)'],
label='India', color='blue')
plt.plot(germany_data['Year'], germany_data['Population Sex Ratio, as of 1 July (males per 100 females)'],
label='Germany', color='red')
plt.title('Population Sex Ratio - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Population Sex Ratio (males per 100 females)')
plt.legend()
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Population Sex Ratio, as of 1 July (males per 100 females)'], bins=20, alpha=0.7,
color='blue', label='India', edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Population Sex Ratio, as of 1 July (males per 100 females)'], bins=20, alpha=0.7,
color='orange', label='Germany', edgecolor='black')

# Add labels and title
plt.title('Population Sex Ratio Distribution - India vs. Germany')
plt.xlabel('Population Sex Ratio (males per 100 females)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
```

```
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Line plot
plt.figure(figsize=(10, 6))
plt.plot(india_data['Year'], india_data['Population Density, as of 1 July (persons per square km)'],
label='India', color='blue')
plt.plot(germany_data['Year'], germany_data['Population Density, as of 1 July (persons per square km)'],
label='Germany', color='red')
plt.title('Population Density Trends - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Population Density (persons per square km)')
plt.legend()
plt.grid(True)
plt.show()

# Assuming df is your DataFrame containing population density data

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(india_data['Population Density, as of 1 July (persons per square km)'],
            germany_data['Population Density, as of 1 July (persons per square km)'])
plt.title('Population Density Relation - India vs. Germany')
plt.xlabel('Population Density - India (persons per square km)')
plt.ylabel('Population Density - Germany (persons per square km)')
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Median Age, as of 1 July (years)'], bins=20, alpha=0.7, color='red', label='India',
edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Median Age, as of 1 July (years)'], bins=20, alpha=0.7, color='yellow',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Median Age Distribution - India vs. Germany')
plt.xlabel('Median Age (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()


# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Births (thousands)'], bins=20, alpha=0.7, color='purple', label='India', edgecolor='black')

# Add labels and title
plt.title('Births Distribution - India')
plt.xlabel('Births (thousands)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for Germany
plt.figure(figsize=(10, 6))
plt.hist(germany_data['Births (thousands)'], bins=20, alpha=0.7, color='cyan', label='Germany',
edgecolor='black')

# Add labels and title
plt.title('Births Distribution - Germany')
plt.xlabel('Births (thousands)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
```

```
plt.figure(figsize=(10, 6))
plt.hist(india_data['Births (thousands)'], bins=20, alpha=0.7, color='purple', label='India', edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Births (thousands)'], bins=20, alpha=0.7, color='cyan', label='Germany',
edgecolor='black')

# Add labels and title
plt.title('Births Distribution - India vs. Germany')
plt.xlabel('Births (thousands)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()


# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Births by women aged 15 to 19 (thousands)'], bins=20, alpha=0.7, color='purple',
label='India', edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Births by women aged 15 to 19 (thousands)'], bins=20, alpha=0.7, color='cyan',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Births Distribution of women aged 15 to 19- India vs. Germany')
plt.xlabel('Births by women aged 15 to 19 (thousands)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Total Fertility Rate (live births per woman)'], bins=20, alpha=0.7, color='red',
label='India', edgecolor='black')
```

```
# Create a histogram for Germany
plt.hist(germany_data['Total Fertility Rate (live births per woman)'], bins=20, alpha=0.7, color='orange',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Total Fertility Rate Distribution - India vs. Germany')
plt.xlabel('Total Fertility Rate (live births per woman)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Life Expectancy at Birth, both sexes (years)'], bins=20, alpha=0.7, color='seagreen',
label='India', edgecolor='black')

# Add labels and title
plt.title('Life Expectancy Distribution - India')
plt.xlabel('Life Expectancy at Birth (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for Germany
plt.figure(figsize=(10, 6))
plt.hist(germany_data['Life Expectancy at Birth, both sexes (years)'], bins=20, alpha=0.7, color='palegreen',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Life Expectancy Distribution - Germany')
plt.xlabel('Life Expectancy at Birth (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
```

```
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Life Expectancy at Birth, both sexes (years)'], bins=20, alpha=0.7, color='seagreen',
label='India', edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Life Expectancy at Birth, both sexes (years)'], bins=20, alpha=0.7, color='palegreen',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Life Expectancy Distribution - India vs. Germany')
plt.xlabel('Life Expectancy at Birth, both sexes (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Plotting the life expectancy trends for India and Germany over the years
plt.figure(figsize=(10, 6))

# Plotting male life expectancy for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Male Life Expectancy at Birth (years)'][df['Region, subregion, country or area *'] == 'India'],
        label='Male India', color='blue')

# Plotting male life expectancy for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Male Life Expectancy at Birth (years)'][df['Region, subregion, country or area *'] == 'Germany'],
        label='Male Germany', color='yellow')

# Plotting female life expectancy for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Female Life Expectancy at Birth (years)'][df['Region, subregion, country or area *'] == 'India'],
        label='Female India', color='green')

# Plotting female life expectancy for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Female Life Expectancy at Birth (years)'][df['Region, subregion, country or area *'] == 'Germany'],
        label='Female Germany', color='red')
```

```
# Adding title and labels
plt.title('Life Expectancy Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Life Expectancy at Birth (years)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Life Expectancy at Age 15, both sexes (years)'], bins=20, alpha=0.7, color='royalblue',
label='India', edgecolor='black')

# Add labels and title
plt.title('Life Expectancy Distribution - India')
plt.xlabel('Life Expectancy at Age 15 (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for Germany
plt.figure(figsize=(10, 6))
plt.hist(germany_data['Life Expectancy at Age 15, both sexes (years)'], bins=20, alpha=0.7, color='lightblue',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Life Expectancy Distribution - Germany')
plt.xlabel('Life Expectancy at Age 15 (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Life Expectancy at Age 15, both sexes (years)'], bins=20, alpha=0.7, color='royalblue',
label='India', edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Life Expectancy at Age 15, both sexes (years)'], bins=20, alpha=0.7, color='lightblue',
label='Germany', edgecolor='black')

# Add labels and title
plt.title('Life Expectancy Distribution - India vs. Germany')
plt.xlabel('Life Expectancy at Age 15 (years)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()


# Plotting the life expectancy trends for India and Germany over the years
plt.figure(figsize=(10, 6))

# Plotting male life expectancy for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Male Life Expectancy at Age 15 (years)'][df['Region, subregion, country or area *'] == 'India'],
        label='Male India', color='blue')

# Plotting male life expectancy for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Male Life Expectancy at Age 15 (years)'][df['Region, subregion, country or area *'] == 'Germany'],
        label='Male Germany', color='yellow')

# Plotting female life expectancy for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Female Life Expectancy at Age 15 (years)'][df['Region, subregion, country or area *'] == 'India'],
        label='Female India', color='green')

# Plotting female life expectancy for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Female Life Expectancy at Age 15 (years)'][df['Region, subregion, country or area *'] ==
'Germany'],
        label='Female Germany', color='red')
```

```
# Adding title and labels
plt.title('Life Expectancy Over Time - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Life Expectancy at Age 15 (years)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Combine data for box plot
data_to_plot = [india_data['Life Expectancy at Birth, both sexes (years)'], germany_data['Life Expectancy at Birth, both sexes (years)']]
labels = ['India', 'Germany']

# Plotting the box plot for life expectancy comparison between India and Germany
plt.figure(figsize=(10, 6))
plt.boxplot(data_to_plot, labels=labels)

plt.title('Life Expectancy Comparison - India vs. Germany')
plt.ylabel('Life Expectancy at Birth')
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Plotting histograms of Mean Age at Childbearing for India and Germany
plt.figure(figsize=(10, 6))

plt.hist(india_data['Mean Age Childbearing (years)'], bins=20, alpha=0.5, label='India', color='blue')
plt.hist(germany_data['Mean Age Childbearing (years)'], bins=20, alpha=0.5, label='Germany', color='red')

plt.title('Mean Age at Childbearing Distribution - India vs. Germany')
plt.xlabel('Mean Age at Childbearing')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.show()
```

```
# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Plotting histograms of Net Reproduction Rate for India and Germany
plt.figure(figsize=(10, 6))

plt.hist(india_data['Net Reproduction Rate (surviving daughters per woman)'], bins=20, alpha=0.5,
label='India', color='blue')
plt.hist(germany_data['Net Reproduction Rate (surviving daughters per woman)'], bins=20, alpha=0.5,
label='Germany', color='red')

plt.title('Net Reproduction Rate Distribution - India vs. Germany')
plt.xlabel('Net Reproduction Rate (surviving daughters per woman)')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Mortality before Age 40, both sexes (deaths under age 40 per 1,000 live births)'], bins=20,
alpha=0.7, color='deeppink', label='India', edgecolor='black')

# Add labels and title
plt.title('Mortality Distribution Before Age 40 - India')
plt.xlabel('Mortality before Age 40 (deaths per 1,000 live births)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for Germany
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for Germany
plt.figure(figsize=(10, 6))
plt.hist(germany_data['Mortality before Age 40, both sexes (deaths under age 40 per 1,000 live births)'],
bins=20, alpha=0.7, color='purple', label='Germany', edgecolor='black')

# Add labels and title
plt.title('Mortality Distribution Before Age 40 - Germany')
plt.xlabel('Mortality before Age 40 (deaths per 1,000 live births)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Filter rows for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
germany_data = df[df['Region, subregion, country or area *'] == 'Germany']

# Create a histogram for India
plt.figure(figsize=(10, 6))
plt.hist(india_data['Mortality before Age 40, both sexes (deaths under age 40 per 1,000 live births)'], bins=20,
alpha=0.7, color='deeppink', label='India', edgecolor='black')

# Create a histogram for Germany
plt.hist(germany_data['Mortality before Age 40, both sexes (deaths under age 40 per 1,000 live births)'],
bins=20, alpha=0.7, color='purple', label='Germany', edgecolor='black')

# Add labels and title
plt.title('Mortality Before Age 40 - India vs. Germany')
plt.xlabel('Mortality before Age 40 (deaths under age 40 per 1,000 live births)')
plt.ylabel('Frequency')
plt.legend()

# Display the histogram
plt.grid(True)
plt.show()

# Plotting the mortality before age 40 trends for India and Germany over the years
plt.figure(figsize=(10, 6))

# Plotting male mortality before age 40 for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Male Mortality before Age 40 (deaths under age 40 per 1,000 male live births)'][df['Region,
subregion, country or area *'] == 'India'],
        label='Male India', color='blue')

# Plotting male mortality before age 40 for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Male Mortality before Age 40 (deaths under age 40 per 1,000 male live births)'][df['Region,
subregion, country or area *'] == 'Germany'],
        label='Male Germany', color='orange')

# Plotting female mortality before age 40 for India
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'India'],
        df['Female Mortality before Age 40 (deaths under age 40 per 1,000 female live births)'][df['Region, subregion, country or area *'] == 'India'],
        label='Female India', color='green')

# Plotting female mortality before age 40 for Germany
plt.plot(df['Year'][df['Region, subregion, country or area *'] == 'Germany'],
        df['Female Mortality before Age 40 (deaths under age 40 per 1,000 female live births)'][df['Region, subregion, country or area *'] == 'Germany'],
        label='Female Germany', color='red')

# Adding title and labels
plt.title('Mortality Before Age 40 - India vs. Germany')
plt.xlabel('Year')
plt.ylabel('Mortality before Age 40 (deaths under age 40 per 1,000 live births)')

# Adding legend
plt.legend()

# Adding grid lines
plt.grid(True)

# Displaying the plot
plt.show()
```

```
import scipy.stats as stats

# Extract the relevant data from the DataFrame for India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
india_life_expectancy = india_data['Life Expectancy at Birth, both sexes (years)']

germany_data = df[df['Region, subregion, country or area *'] == 'Germany']
germany_life_expectancy = germany_data['Life Expectancy at Birth, both sexes (years)']

# Perform t-test for independent samples
t_statistic, p_value = stats.ttest_ind(india_life_expectancy, germany_life_expectancy, equal_var=False)

# Set the significance level (alpha)
alpha = 0.05

# Print t-statistic and p-value
print(f'T-statistic: {t_statistic}')
print(f'P-value: {p_value}')

# Set the null hypothesis
if p_value < alpha:
    print('Reject null hypothesis: Indians have a higher life expectancy at birth compared to Germans.')
else:
    print('Fail to reject null hypothesis: There is no significant difference in life expectancy at birth between Indians and Germans.')

import scipy.stats as stats

# Extract the relevant data from the DataFrame for India
india_data = df[df['Region, subregion, country or area *'] == 'India']
india_net_migration_rate = india_data['Net Migration Rate (per 1,000 population)']

# Perform one-sample t-test for India
t_statistic, p_value = stats.ttest_1samp(india_net_migration_rate, 0)

# Set the significance level (alpha)
alpha = 0.05

# Print t-statistic and p-value
print(f'T-statistic: {t_statistic}')
print(f'P-value: {p_value}')

# Set the null hypothesis
if p_value < alpha:
    print('Reject null hypothesis: Indians are migrating more over the years.')
else:
    print('Fail to reject null hypothesis: There is no significant difference in net migration rate over the years in India.')

import pandas as pd
import scipy.stats as stats


# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Read the CSV file into a DataFrame
df = pd.read_csv('/content/drive/MyDrive/India_Germany.csv')


# Extract the relevant data for median age in India and Germany
india_data = df[df['Region, subregion, country or area *'] == 'India']
india_median_age = india_data['Median Age, as of 1 July (years)']

germany_data = df[df['Region, subregion, country or area *'] == 'Germany']
germany_median_age = germany_data['Median Age, as of 1 July (years)']

# Perform t-test for independent samples
t_statistic, p_value = stats.ttest_ind(india_median_age, germany_median_age, equal_var=False)

# Set the significance level (alpha)
```

```
alpha = 0.05

# Print t-statistic and p-value
print(f'T-statistic: {t_statistic}')
print(f'P-value: {p_value}')

# Set the null hypothesis
if p_value < alpha:
    print('Reject null hypothesis: Indians have a lower median age compared to Germans.')
else:
    print('Fail to reject null hypothesis: There is no significant difference in median age between Indians and Germans.')

PREPROCESSING
df = df.dropna(thresh=30)
olumns_with_mixed_types = [4, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28,
                            29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
                            46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
                            63, 64, 65]

# Read the CSV file with specified dtype for columns
data = pd.read_csv('vikrant.csv', header=1, dtype={col: str for col in columns_with_mixed_types}, index_col='Index')
data.drop('15', axis =1, inplace = True)
# Filter the DataFrame to include rows for India or Germany
India_germany = data[(data['Region, subregion, country or area *'] == 'India') | (data['Region, subregion, country or area *'] == 'Germany')]
India_germany.to_csv('India_Germany.csv')
new = pd.read_csv('India_Germany.csv')
new.drop('Notes', axis = 1, inplace = True)
```