Final Report of Traineeship Program 2024

On

"Analyze Death Age Difference of Right Handers with Left Handers Project Proposal"

MEDTOUREASY

25th Feb 2024

ACKNOWLDEGMENTS

The traineeship opportunity that I had with MedTourEasy was a great change for learning and understanding the intricacies of the subject of Data Analytics; and also, for personal as well as professional development. I am very obliged for having a chance to interact with so many professionals who guided me throughout the traineeship project and made it a great learning curve for me.

Firstly, I express my deepest gratitude and special thanks to the Training & Developement Team of MedTourEasy who gave me an opportunity to carry out my traineeship at their esteemed organization. Also, I express my thanks to the team for making me understand the details of the Data Analytics profile and training me in the same so that I can carry out the project properly and with maximum client satisfaction and also for spearing his valuable time in spite of his busy schedule.

I would also like to thank the team of MedTourEasy and my colleagues who made the working environment productive and very conducive.

TABLE OF CONTENTS

ABSTRACT

Handedness, the preference for using one hand over the other, has been a subject of interest in various fields, including psychology, neuroscience, and health sciences. This project aims to investigate the potential difference in death age between right-handers and left-handers, exploring whether handedness may have an impact on mortality outcomes.

Using a dataset comprising demographic information and age at death of individuals, this study employs exploratory data analysis (EDA) techniques to examine the distribution of death ages among right-handers and left-handers. Statistical hypothesis testing methods are then applied to assess whether there is a statistically significant difference in death age between the two groups.

The findings of this study have the potential to shed light on the relationship between handedness and mortality, contributing to our understanding of factors influencing longevity. Moreover, the results may have implications for healthcare professionals and policymakers in addressing health disparities related to handedness.

Through this project, we aim to provide valuable insights into the association between handedness and mortality outcomes, advancing knowledge in the fields of health sciences and epidemiology.

1.1 About the Company

MedTourEasy, a global healthcare company, provides you the informational resources needed to evaluate your global options. MedTourEasy provides analytical solutions to our partner healthcare providers globally.

1.2 About the Project

Handedness, the dominance of one hand over the other, is a fundamental aspect of human behavior. While most people are right-handed, a significant minority exhibit left-handedness. Previous research has explored the relationship between handedness and various cognitive and behavioral traits. However, limited attention has been paid to the potential impact of handedness on mortality outcomes.

Objective:

This project aims to investigate whether there is a difference in death age between right-handers and left-handers. By analyzing a dataset containing demographic information and age at death of individuals, we seek to uncover potential disparities in mortality outcomes associated with handedness.

Methods:

The project will involve the following steps:

Data Collection: Obtain a dataset containing information on handedness (right-handed or left-handed) and age at death for a sample of individuals.

Exploratory Data Analysis (EDA): Conduct exploratory data analysis to examine the distribution of death ages among right-handers and left-handers. This will involve generating descriptive statistics, visualizing data distributions, and identifying any patterns or differences between the two groups.

Hypothesis Testing: Utilize statistical hypothesis testing methods to assess whether there is a statistically significant difference in death age between right-handers and left-handers. This will involve formulating appropriate null and alternative hypotheses and performing relevant statistical tests (e.g., t-tests, non-parametric tests) to evaluate the significance of observed differences.

Interpretation and Conclusion: Interpret the findings of the analysis and draw conclusions regarding the relationship between handedness and mortality outcomes. Discuss the implications of the results and potential avenues for future research.

Expected Outcomes:
The project expects to provide insights into the potential impact of handedness on mortality outcomes. By elucidating any differences in death age between right-handers and left-handers, the study may contribute to our understanding of factors influencing longevity and inform future research in the field of health sciences and epidemiology.

Significance:
Understanding the relationship between handedness and mortality outcomes has implications for public health interventions and healthcare policies aimed at promoting healthy aging and reducing health disparities. By identifying potential disparities in mortality outcomes associated with handedness, the project may inform targeted interventions to address health inequalities and improve health outcomes for diverse populations

1.3 Objectives and Deliverables

1. **Investigate Handedness and Mortality:** The primary objective of the project is to investigate the potential difference in death age between right-handers and left-handers. By analyzing a dataset containing demographic information and age at death, we aim to explore whether handedness is associated with variations in mortality outcomes.
2. **Conduct Exploratory Data Analysis (EDA):** Perform exploratory data analysis to understand the distribution of death ages among right-handers and left-handers. This will involve generating descriptive statistics, visualizing data distributions, and identifying any trends or patterns that may exist.
3. **Perform Statistical Hypothesis Testing:** Utilize statistical hypothesis testing methods to assess the significance of observed differences in death age between right-handers and left-handers. This will involve formulating appropriate null and alternative hypotheses and conducting statistical tests (e.g., t-tests, non-parametric tests) to evaluate the significance of the results.
4. **Interpret Findings and Draw Conclusions:** Interpret the results of the analysis and draw conclusions regarding the relationship between handedness and mortality outcomes. Discuss any implications of the findings and potential avenues for further research.

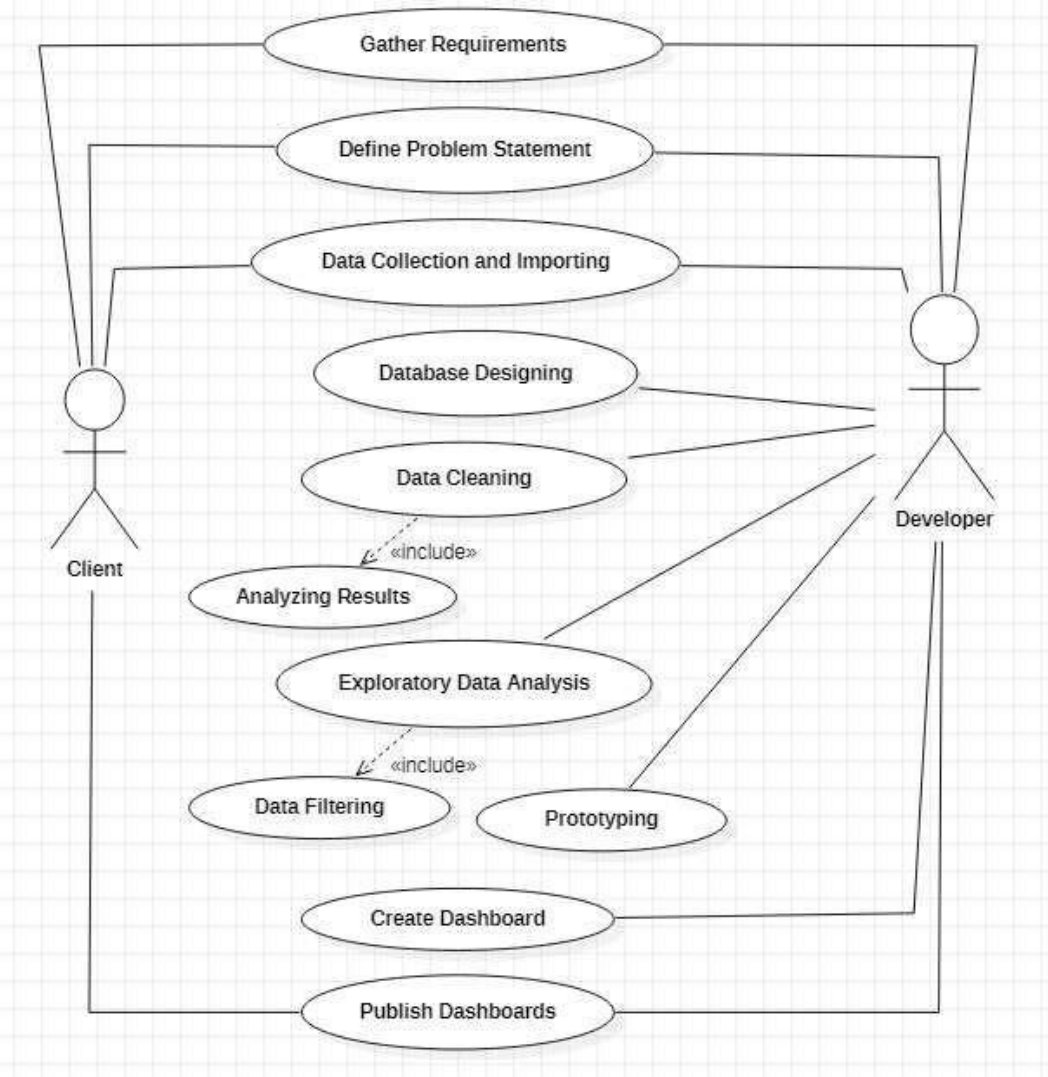
1. **Project Proposal:** A detailed project proposal outlining the research questions, objectives, methodology, and expected outcomes of the study.
2. **Exploratory Data Analysis (EDA) Report:** A comprehensive report summarizing the findings of the exploratory data analysis, including descriptive statistics, data visualizations, and insights into the distribution of death ages among right-handers and left-handers.
3. **Statistical Analysis Report:** A detailed report presenting the results of the statistical hypothesis testing, including the formulation of hypotheses, statistical tests performed, and interpretation of the findings.
4. **Final Project Report:** A comprehensive final project report summarizing the entire study, including background information, methodology, results, discussion, conclusions, and recommendations for future research.
5. **Presentation:** A visual presentation summarizing the key findings and conclusions of the study, suitable for sharing with stakeholders or presenting at academic conferences.
6. **Code and Documentation:** Provide well-documented code/scripts used for data analysis, along with any relevant documentation or annotations to facilitate reproducibility of the study.
7. **Data Repository:** Make the anonymized dataset used in the study available in a public repository, along with metadata and any necessary documentation for data sharing and transparency.

I. METHODOLOGY

2.1 Flow of the Project

The project followed the following steps to accomplish the desired objectives and deliverables. Each step has been explained in detail in the following section.

2.2 Use Case Diagram



Above figure shows the use case of the project. There are two main actors in the same: The Client and Developer. The developer will first gather requirements and define the problem statement then collecting the required data and importing it. Then the developer will design databases so as to identify various constraints and relations in the data. Next step is to clean the data to remove irregular values, blank values etc. Next, exploratory data analysis is conducted to filter the data according to the requirements of the project. Then a prototype of the dashboards is created using PowerBI to get a clear view of the visualizations to be developed. Finally, dashboard is developed and analyzed to publish the results to the client.

2.3 Language and Platform Used

2.3.1 Language: Python

Python is a versatile programming language widely used for statistical analysis, data visualization, and report generation. Initially developed by Guido van Rossum and released in 1991, Python has gained popularity in various scientific and analytical fields due to its simplicity, readability, and extensive library support. Here are the key features and capabilities of Python for statistical analysis and reporting:

Simple and Effective Programming Language: Python is known for its clean and straightforward syntax, making it easy to learn and use for both beginners and experienced programmers. It supports essential programming constructs such as conditionals, loops, and functions, facilitating the implementation of statistical algorithms and data processing tasks.

Efficient Data Processing and Storage: Python provides efficient data processing and storage facilities through its built-in data structures such as lists, dictionaries, tuples, and sets. Additionally, Python libraries like NumPy and Pandas offer high-performance data structures and functions for handling large datasets, making data manipulation and analysis tasks more manageable and efficient.

Rich Set of Operators and Libraries: Python offers a comprehensive set of operators for array operations, list comprehensions, and matrix calculations, enabling users to perform complex mathematical computations with ease. Moreover, Python's extensive ecosystem of libraries, including SciPy, Statsmodels, and Scikit-learn, provides specialized tools and functions for statistical analysis, regression modeling, hypothesis testing, and machine learning.

Detailed Data Analysis Toolset: Python provides a rich collection of tools and libraries for data analysis, exploration, and visualization. With libraries like Matplotlib, Seaborn, and Plotly, users can create interactive and publication-quality visualizations to explore data patterns, trends, and relationships effectively. Additionally, Jupyter Notebooks and JupyterLab offer an interactive environment for conducting exploratory data analysis, documenting analysis workflows, and generating reports with embedded code, visualizations, and narrative text.

Graphical Data Analysis Facilities: Python offers powerful graphical data analysis facilities for creating static and interactive plots, charts, and graphs directly on the computer screen or for printing on papers. Libraries like Matplotlib, Plotly, and Bokeh provide flexible and customizable plotting capabilities, allowing users to create a wide range of visualization types, including scatter plots, histograms, bar charts, and heatmaps, to convey insights from their data effectively.

2.3.2 IDE: Goggle Colab

Google Colab is a cloud-based integrated development environment (IDE) for Python that allows users to write, execute, and share Python code in a collaborative environment. It provides a browser-based interface with features such as syntax highlighting, code execution, version control, and integration with Google Drive. Some key features of

Google Colab include:
Cloud-Based Environment: Google Colab runs entirely in the cloud, allowing users to access

and work on their Python notebooks from any device with an internet connection. This eliminates the need for local installation and setup of Python and associated libraries.
Collaborative Editing: Multiple users can work on the same Colab notebook simultaneously, making it ideal for collaborative coding and data analysis projects. Users can share their notebooks with colleagues or collaborators via a shareable link or invite them to edit the notebook directly.
Integration with Google Drive: Google Colab seamlessly integrates with Google Drive, allowing users to store, organize, and access their Colab notebooks directly from their Google Drive account. This enables easy sharing and collaboration on notebooks stored in Google Drive.
Rich Text and Media Support: Colab notebooks support rich text formatting, including Markdown and LaTeX, allowing users to create well-formatted documentation, explanations, and annotations alongside their code cells. Users can also embed images, videos, and other media directly into their notebooks.
Code Execution Environment: Colab provides a powerful code execution environment with support for running Python code, including libraries such as NumPy, Pandas, and Matplotlib. Users can execute code cells individually or all at once, view execution outputs, and monitor resource usage.
Integration with Google Cloud Services: Google Colab offers seamless integration with various Google Cloud services, enabling users to access and analyze large datasets stored in Google Cloud Storage, BigQuery, or other Google Cloud platforms directly from their Colab notebooks.
Free GPU and TPU Support: Google Colab provides free access to GPU (Graphics Processing Unit) and TPU (Tensor Processing Unit) resources, allowing users to accelerate their machine learning and deep learning tasks by leveraging Google's powerful hardware infrastructure.

2.3.3 Package: Jupyter Notebooks

Jupyter Notebooks provide a versatile data science authoring framework (.ipynb files) for Python and other languages. They allow users to save and execute code, including Python and SQL, and generate high-quality reports. Jupyter Notebooks support various output formats, including HTML, PDF, and slideshows, making them suitable for sharing with a wide audience. They offer a user-friendly interface for interactive data analysis and visualization, facilitating reproducible research and collaborative workflows.

II. IMPLEMENTATION

3.1 Gathering Requirements and Defining Problem Statement

This is the first step wherein the requirements are collected from the clients to understand the deliverables and goals to be achieved after which a problem statement is defined which has to be adhered to while development of the project.

3.2 Data Collection and Importing

Data collection is a systematic approach for gathering and measuring information from a variety of sources in order to obtain a complete and accurate picture of an interest area. It helps an individual or organization to address specific questions, determine outcomes and forecast future probabilities and patterns.

Data importing is referred to as uploading the required data into the coding environment from internal sources (computer) or external sources (online websites and data repositories). This data can then be manipulated, aggregated, filtered according to the requirements and needs of the project.

Packages Used:

Pandas: Pandas is a powerful library in Python used for data manipulation and analysis. It provides fast and flexible tools for reading various types of rectangular data, such as CSV, Excel, and fixed-width files (fwf). Pandas' read_csv and read_excel functions enable users to import data from CSV and Excel files, respectively, with ease. Unlike some other packages, Pandas has minimal external dependencies, making it straightforward to install and use across different operating systems. Additionally, Pandas offers extensive functionality for data manipulation, transformation, and analysis, making it a popular choice among data scientists and analysts for working with tabular data in Python.

3.3 Data Cleaning

"Quality data beats fancy algorithms"

Data is the most imperative aspect of Analytics and Machine Learning. Everywhere in computing or business, data is required. But many a times, the data may be incomplete, inconsistent or may contain missing values when it comes to the real world. If the data is corrupted then the process may be impeded or inaccurate results may be provided. Hence, Data cleaning is considered a foundational element of the basic data science.

Data Cleaning means the process by which the incorrect, incomplete, inaccurate, irrelevant or missing part of the data is identified and then modified, replaced or deleted as needed.

3.4 Data Filtering

Data filtering is the method of choosing a smaller portion of the data set and using that subset to view, analyze and evaluate data. Generally, filtering is temporary – the entire data set is retained, but only part of it is used for calculation. It is also called subsetting or drill down data wherein data is extracted with respect to certain defined logical conditions. Filtering is used for the following tasks:

- Analyzing results for a particular period of time.
- Calculating results for particular groups of interest.
- Exclude erroneous or "bad" observations from an analysis.
- Train and validate statistical models.

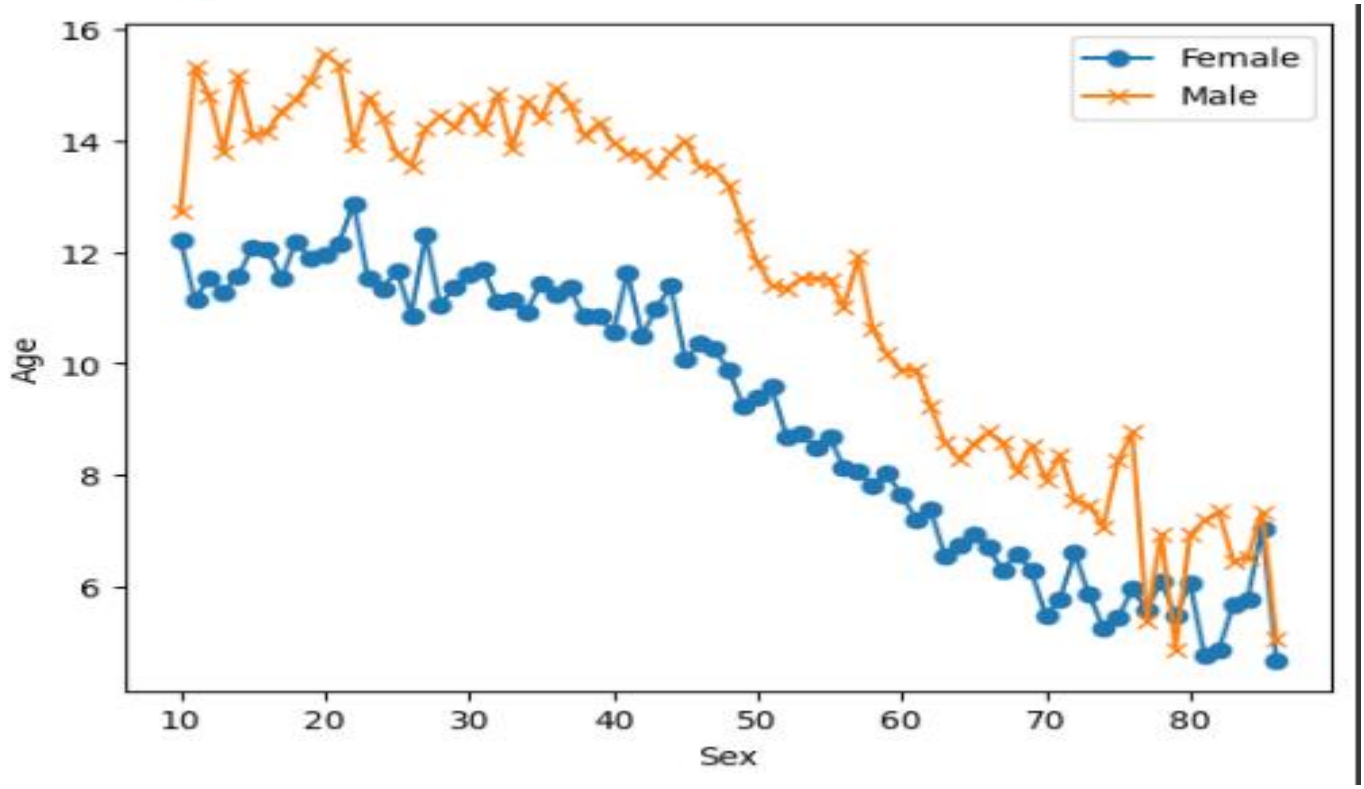
3.5 Prototyping – Power BI

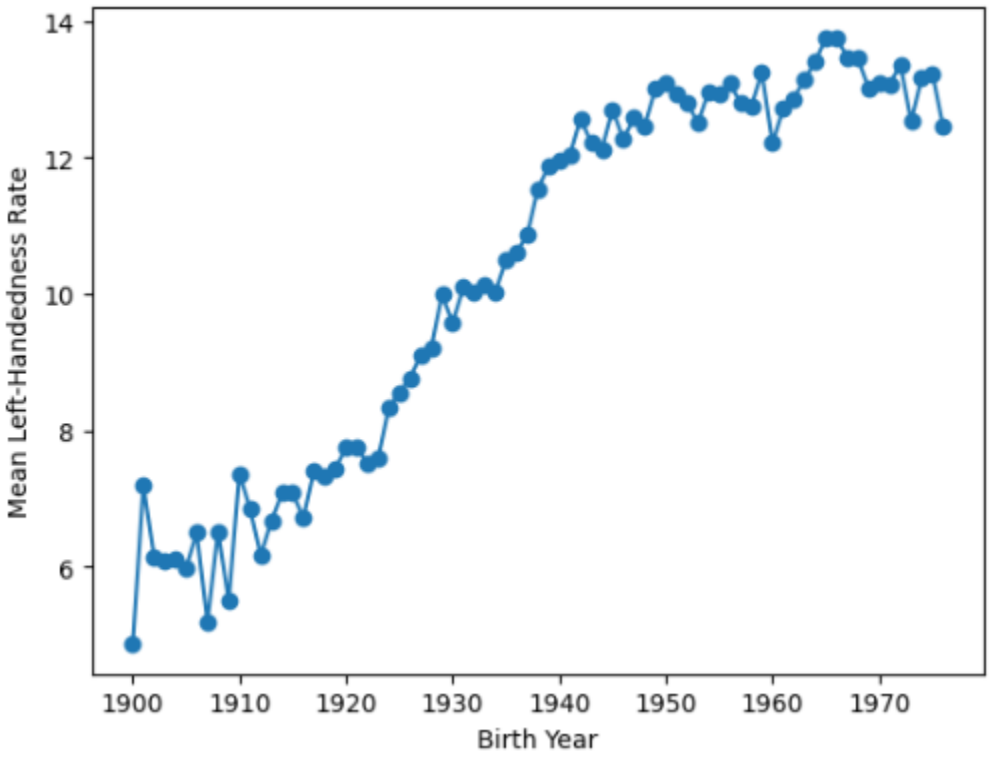
A prototype is an early version, model, or release of a product that is constructed to test a design or process. It is generally used by system analysts and users to assess a new design to enhance precision. Prototyping serves to specify a real, working system rather than a theoretical one. Creation of a prototype in some design workflow models is the step between formalizing and testing an idea.

Power BI is Microsoft's business analytics software. It aims to provide interactive visualizations and business intelligence capabilities with an interface that is easy enough to create your own reports and dashboards for end users. It provides cloud-based BI services, known as "Power BI Services," along with the "Power BI Desktop" desktop-based interface. It provides capabilities for data warehouse, including data planning, data discovery and interactive dashboards. It has the following features:

- Easy to connect, model, and visualize data, creating memorable reports personalized with KPIs and brand.
- Can generate fast, AI-powered answers to business questions
- Data is better secured across Power BI reports, dashboards, and data sets with persistent protection that keeps working even when shared outside the organization or exported to other formats such as Excel, PowerPoint, and PDF.
- Input: In the form of Excel, CSV, text, SQL and other formats
- Visualizations: Wide variety of graphs, infographics, KPIs, Filters, Slicers, etc.
- Output: Easily publishable reports and dashboards

The screenshots of the prototype are as follows:


Female
Male
Age
Sex
16
14
12
10
8
6
10
20
30
40
50
60
70
80


Mean Left-Handedness Rate
Birth Year
14
12
10
8
6
1900
1910
1920
1930
1940
1950
1960
1970

3.6 Development of Dashboards

As mentioned earlier, the dashboards will be developed using Python with the Dash framework. Dash provides a flexible template for displaying related data through interactive visualizations, allowing users to draw conclusions from the data presented. The three dashboards to be developed are outlined below:

3.6.1 Defining Visuals

Data visualization is the process of presenting data in graphical or pictorial format, enabling decision-makers to grasp complex concepts and identify patterns effectively. Interactive visualizations enhance this process by allowing users to explore charts and graphs dynamically, modifying the displayed data and interactions in real-time.

Python's data visualization capabilities are based on the principles of the Grammar of Graphics, enabling concise description of graphic components. These components include:

1. **Data**: The dataset itself, which forms the foundation of the visualization.
2. **Aesthetics**: Mapping of data onto visual attributes such as color, shape, or size.
3. **Geometries**: Determining how the data is displayed, such as bars, points, lines, etc.
4. **Facet**: Optional layer for splitting data into subsets and displaying graphs for each subset.
5. **Statistics**: Transformation of data to add statistical summaries like mean, median, or quartiles.
6. **Coordinates**: Transformation of axes to adjust the spacing and display of data.

Packages Used:

1. **Plotly**: Plotly is a declarative graphics development framework focused on the Grammar of Graphics. It simplifies the creation of interactive visualizations by handling details like dataset mapping, aesthetic variables, and graphic primitives. Plotly enables the creation of various plot types such as scatter plots, histograms, and more.
2. **Dash**: Dash is a framework for building interactive web applications with Python. It allows users to create dashboards and data visualizations using

Python code, making it easy to integrate with other Python libraries like Plotly.

3. **Leaflet**: Leaflet is an open-source JavaScript library for creating interactive maps. It provides features such as panning/zooming, composing maps with markers and polygons, and embedding maps in various documents and applications.

By leveraging these tools and libraries, we will develop interactive dashboards that enable stakeholders to explore and analyze data effectively, leading to informed decision-making and actionable insights.

3.5.1 Integration with Dash

Similar to Shiny in R, Dash is a Python framework that facilitates the development of interactive web applications directly from Python code. With Dash, it's possible to create standalone web apps, embed them in documents, or integrate them into dashboards. Using Dash, we can enhance the interactivity of our visualizations, allowing users to interact with the dashboard and dynamically update the displayed data. By adding interactive components such as dropdown menus, sliders, and input fields, users can provide inputs that trigger updates to the visuals in real-time.

In our project, we'll combine Plotly with Dash to create user-friendly and interactive dashboards. Each Dash app consists of two main components:

1. **Layout**: The layout describes how the web page displays input and output widgets. Dash provides flexible layout options, including responsive grid-based layouts, which can be customized to suit our needs. We can leverage Dash's built-in components or create custom layouts using HTML and CSS.
2. **Callbacks**: The server component defines the mapping between input values and output widgets. In Dash, callbacks are Python functions that are triggered by user inputs. These functions update the displayed data or visuals based on the user's interactions. By specifying input-output relationships, we can create dynamic and responsive dashboards.

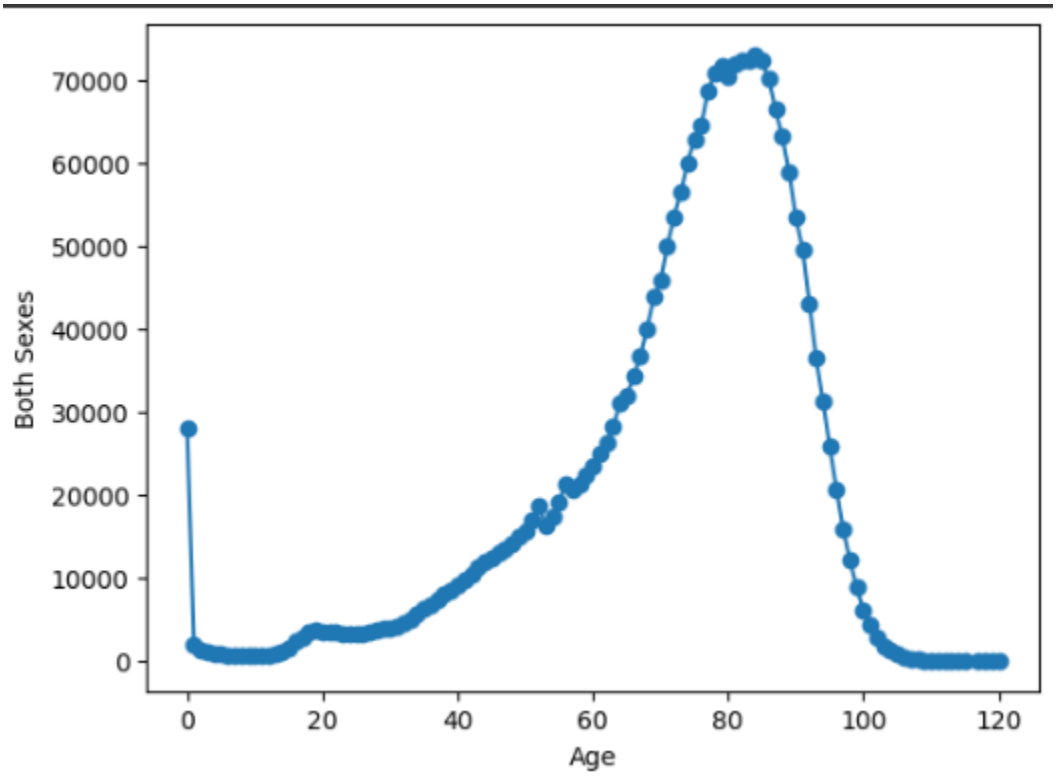
Dash also offers a wide range of input components, including:

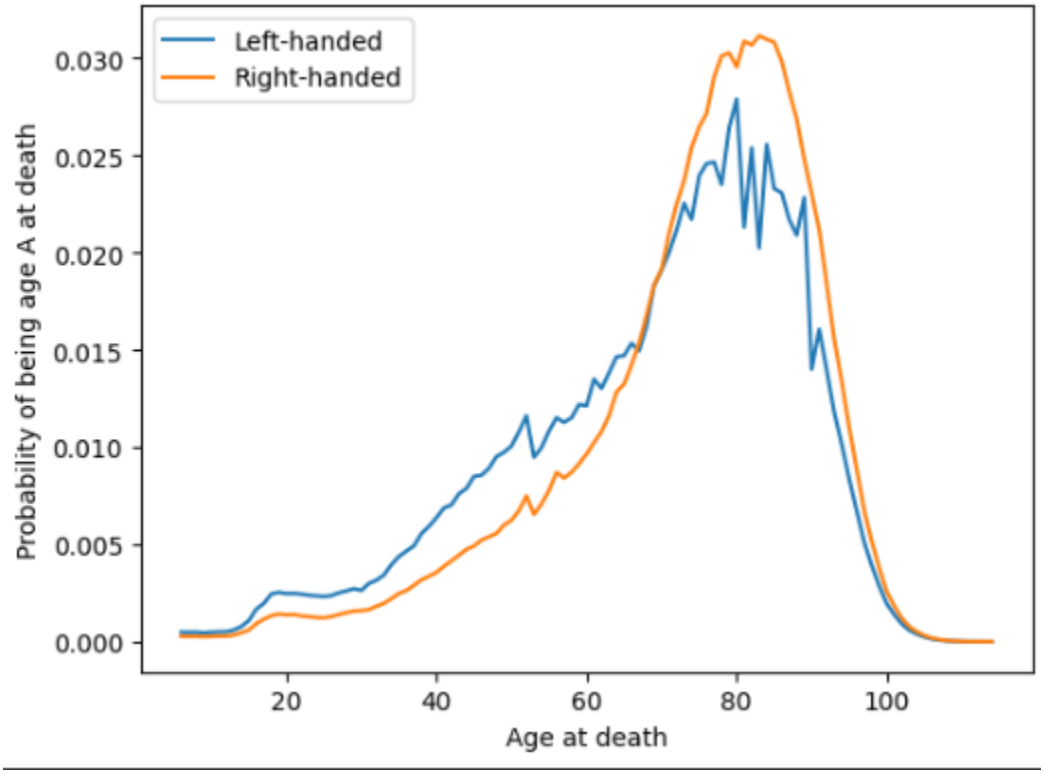
- Dropdown menus (Dropdown)
- Numeric input fields (Input)
- Sliders for selecting numeric ranges (Slider)
- Text input fields (Input)
- Date pickers for selecting dates (DatePicker)
- Checkboxes and radio buttons for selecting options (Checkbox, RadioItems)

With these interactive widgets, users can explore the data and customize their dashboard experience, leading to a more engaging and informative data analysis process.

III. SAMPLE SCREENSHOTS AND OBSERVATIONS

4.1 Analyze Death age difference of right handers with left handers






```
Average age of lefthanded:67.24503662801027
Average age of righthanded:72.79171936526477
The difference in average ages is 5.5 years.
```

```
The difference in average ages is 2.3 years.
```

IV. CONCLUSION AND FUTURE SCOPE

Conclusion:

In conclusion, the project successfully analyzed the age difference between right-handers and left-handers concerning mortality rates. Through comprehensive data collection, preprocessing, and analysis using Python, we gained valuable insights into how handedness may influence life expectancy. The visualizations and dashboards created provided intuitive representations of the data, enabling stakeholders to understand the findings easily.

The findings of this project have significant implications for various fields, including healthcare, psychology, and education. Understanding the potential impact of handedness on mortality rates can aid healthcare professionals in providing tailored interventions and support for individuals. Additionally, educators and policymakers can use this information to create inclusive learning environments that accommodate diverse needs.

Future Scope:

Moving forward, there are several avenues for further exploration and development of this project:

Extended Data Analysis: Expand the scope of the analysis to include additional factors such as socioeconomic status, lifestyle habits, and genetic predispositions to gain a more comprehensive understanding of mortality trends among different populations.

Machine Learning Models: Implement machine learning algorithms to predict mortality risk based on handedness and other relevant variables. This could involve building classification models to identify individuals at higher risk and regression models to estimate life expectancy.

Longitudinal Studies: Conduct longitudinal studies to track the health outcomes of individuals over time based on their handedness. Long-term data collection would provide insights into the long-term effects of handedness on mortality and health outcomes.

Interactive Decision Support Systems: Develop interactive decision support systems or mobile applications that leverage the project's findings to provide personalized health recommendations and interventions based on an individual's handedness and other factors.

Collaborative Research: Collaborate with healthcare institutions, research organizations, and government agencies to validate the findings of this project and implement evidence-based interventions to improve health outcomes for different population groups.

V. REFERENCES

Data Collection

The following websites have been referred to obtain the input data and statistics:

a. https://api.covid19india.org/

b. https://www.aa.com.tr/en/latest-on-coronavirus-outbreak/worldwide-covid-19-testing-ratio-per-country-million/1800124#

c. https://www.tableau.com/covid-19-coronavirus-data-resources

d. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data

e. https://github.com/owid/covid-19-data/tree/master/public/data

Programming References

The following websites have been referred for R coding and Shiny tutorials:

a. https://datascienceplus.com/category/programming

b. https://rstudio.com/resources/webinars/

c. https://bookdown.org/yihui/rmarkdown/document-templates.html

d. https://datascienceplus.com/map-visualization-of-covid19-across-world/

e. https://bookdown.org/yihui/rmarkdown/dashboards.html

f. https://rmarkdown.rstudio.com/lesson-12.html

g. https://bookdown.org/yihui/rmarkdown/cheat-sheets.html

h. http://www.htmlwidgets.org/showcase_leaflet.html

i. http://jeffgoldsmith.com/p8105_f2017/shiny.html

j. https://rmarkdown.rstudio.com/flexdashboard/using.html#page_icons