Final Report of Traineeship Program 2023

On

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

MEDTOUREASY

16th August 2023

ACKNOWLDEGMENTS

I am deeply thankful for the profound learning and personal growth I gained during my traineeship with MedTourEasy, specifically in the realm of Data Analytics. Engaging with professionals greatly expanded my knowledge and skills throughout this experience.

My sincere appreciation to MedTourEasy's Training & Development Team for providing me with this opportunity. Despite their demanding schedules, their guidance and training equipped me to excel in the project. I also extend my gratitude to my colleagues and the entire team for fostering a conducive working environment.

A special note of thanks to all those involved in the project "Analyzing the Age Difference between Right Handers and Left Handers." Your unwavering support significantly contributed to the success of this enriching journey.

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

This project delves into the intriguing intersection of left-handedness and age at death, aiming to uncover the relationship between these factors through a comprehensive data-driven analysis. Leveraging data from a National Geographic survey conducted in 1986, coupled with death distribution data from the year 1999 for the United States, the study investigates the evolving landscape of left-handedness rates across generations and its impact on the age at which individuals pass away.

The analysis begins by examining the changing rates of left-handedness over time, showing that left-handedness rates decrease with age, revealing a social shift in the acceptance of left-handedness rather than a direct correlation with age. Through a meticulous utilization of Bayes' theorem, the study calculates conditional probabilities that a person is of a certain age at death given their handedness, both for left-handed and right-handed individuals. The results are synthesized in a probability distribution that highlights the likelihood of dying at various ages for each handedness group.

The project also explores how the average ages at death differ between left-handed and right-handed individuals, shedding light on the potential impact of handedness on longevity. By comparing these findings with previous studies, the project offers insights into the complex interplay of handedness, societal shifts, and age-related factors that contribute to differences in the average ages at death.

In conclusion, this project showcases the power of data analysis in unraveling hidden patterns within seemingly unrelated variables. It highlights the evolving dynamics of left-handedness and its implications on age at death, serving as a testament to the multifaceted nature of human traits and the intricate influences shaping them over time.

INTRODUCTION

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

This project explores the intricate relationship between left-handedness and age at death through a rigorous data analysis approach. Utilizing data from a 1986 National Geographic survey and death distribution data from 1999 for the United States, the study delves into the changing rates of left-handedness across different age groups and its potential influence on the age at which individuals pass away.

By scrutinizing the historical trends of left-handedness rates, the project unveils a pattern where rates decline with age, not due to the aging process itself, but rather as a reflection of societal shifts in attitudes towards left-handedness. Leveraging Bayes' theorem, the study calculates conditional probabilities of specific age groups at death based on handedness, for both left-handed and right-handed individuals. The outcomes are synthesized into probability distributions, revealing the probabilities of dying at various ages for each handedness group.

The project also scrutinizes the average ages at death for left-handed and right-handed individuals, providing insights into potential disparities in longevity associated with handedness. By comparing findings with existing research, the project contributes to understanding the interplay between left-handedness, societal dynamics, and age-related phenomena, shedding light on the multifaceted factors influencing lifespan disparities.

In conclusion, this project underscores the value of data analysis in uncovering hidden connections between seemingly unrelated factors. It elucidates the evolving landscape of left-handedness and its potential implications for age at death, offering a glimpse into the intricate interplay between biological and sociocultural elements.

1.3 Objectives and Deliverables

1. ***Investigate Left-Handedness and Age at Death Relationship***: The primary objective of this project is to examine the potential relationship between left-handedness and age at death. By utilizing age distribution data and Bayesian statistics, we aim to determine whether differences in average age at death between left-handed and right-handed individuals can be attributed to the changing rates of left-handedness over time.

2. ***Analyze Changing Left-Handedness Rates***: Explore the historical trends of left-handedness rates over different age groups using data from a 1986 National Geographic survey. Investigate the hypothesis that left-handedness rates decline with age due to evolving societal attitudes, rather than a direct correlation with age.

3. ***Apply Bayesian Analysis***: Utilize Bayesian statistics to calculate conditional probabilities that an individual is of a certain age at death given their handedness. Employ Bayes' theorem to derive these probabilities for both left-handed and right-handed individuals.

4. **Create Probability Distributions**: Develop probability distributions that illustrate the likelihood of dying at various ages for left-handed and right-handed individuals based on their respective conditional probabilities. Visualize these distributions to gain insights into the age at death patterns.

The project consists of following deliverables:

1. **Exploratory Analysis of Left-Handedness Rates:** Provide an analysis of the historical trends in left-handedness rates using the 1986 National Geographic survey data. Visualize how left-handedness rates change across different age groups and explain the implications of these trends.

2. **Bayesian Analysis Implementation:** Present the implementation of Bayesian statistics to calculate conditional probabilities of being a specific age at death given left-handedness or right-handedness. Explain the methodology and mathematical concepts used in the analysis.

3. **Probability Distribution Visualizations:** Display probability distributions that showcase the likelihood of dying at various ages for left-handed and right-handed individuals. Include graphical representations of these distributions to illustrate the patterns.

4. **Comparison of Average Ages at Death:** Compare and contrast the average ages at death for left-handed and right-handed individuals based on the probability distributions. Analyze the implications of these comparisons on the claims of early death for left-handers.

5. **Discussion and Interpretation:** Provide a comprehensive discussion of the project's findings, interpreting the implications of changing left-handedness rates on age at death disparities. Address any limitations of the analysis and discuss the broader significance of the results.

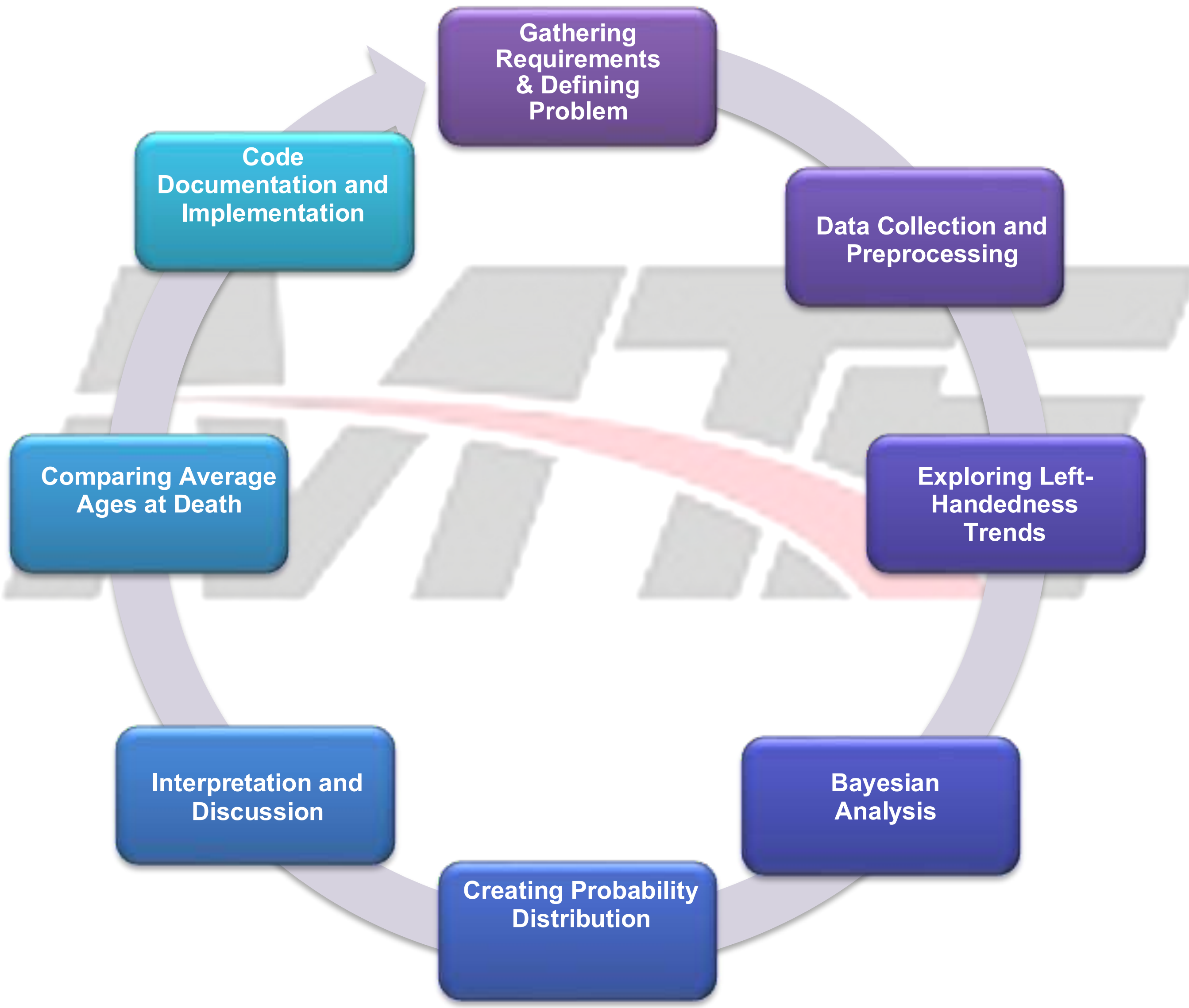
6. **Code Documentation:** Document the code used in the analysis, explaining the purpose and functions of key code segments. Ensure that the code is clear, well-organized, and easily understandable by other researchers.

7. **Project Conclusion:** Summarize the project's objectives, methodologies, findings, and implications in a coherent and concise conclusion. Reflect on the achievements of the project and suggest potential future research directions.

METHODOLOGY

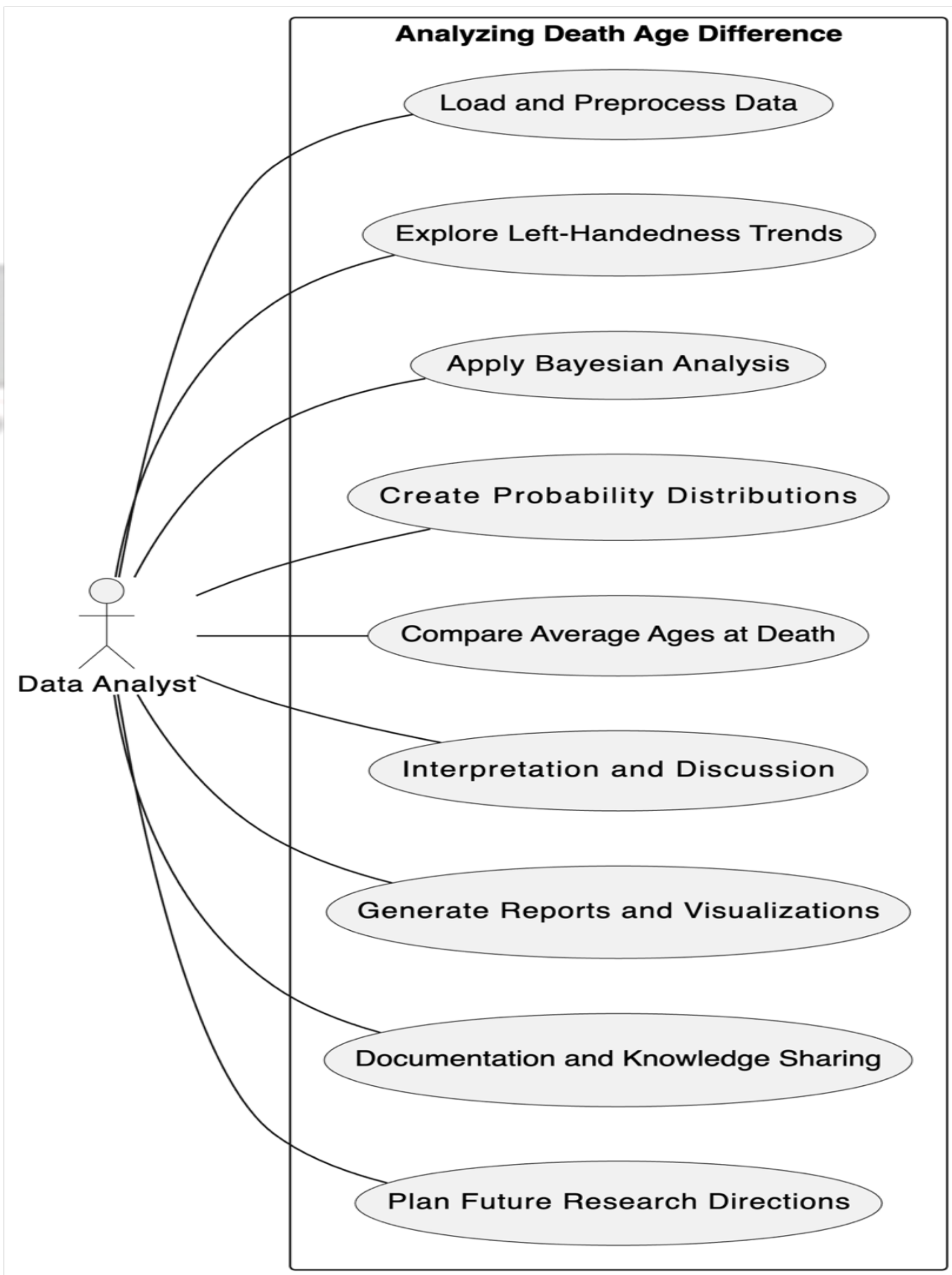
2.1 Flow of the Project

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

The use case diagram illustrates the interactions and functionalities involved in the project "Analyzing Death Age Difference of Right Handers with Left Handers." The primary actor is the "Data Analyst," who is responsible for performing various tasks to analyze the age difference in death between right-handed and left-handed individuals.



2.3 Language and Platform Used

2.3.1 Language: Python

Python is immensely helpful in Exploratory Data Analysis (EDA) due to its rich ecosystem of libraries, tools, and capabilities that streamline the process of data exploration, visualization, and analysis. The important features of Python are:

- **Rich Ecosystem of Libraries:** Python offers a vast array of libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn, which cover data manipulation, analysis, visualization, machine learning, and more. These libraries streamline the entire data analysis pipeline.
- **Data Visualization:** Python's libraries like Matplotlib, Seaborn, and Plotly provide extensive capabilities for creating static and interactive visualizations. Data analysts can create various charts, plots, and graphs to effectively communicate insights from the data.
- **Data Manipulation:** The Pandas library is particularly powerful for data manipulation tasks. Data analysts can easily load, clean, transform, and reshape data using DataFrames, making it efficient to prepare data for analysis.
- **Interactivity and Exploration:** Jupyter Notebooks offer an interactive environment where data analysts can execute code, visualize results, and provide explanations within a single document. This promotes exploratory data analysis and enables documentation of findings.
- **Statistical Analysis and Testing:** Python's Scipy library provides statistical functions and hypothesis testing tools for data analysts to perform a wide range of statistical analyses. This is crucial for validating assumptions and drawing meaningful insights from data.

2.3.2 IDE: Visual Studio Code

Visual Studio Code (VS Code) is a popular integrated development environment (IDE) that can be effectively used for data analysis tasks, especially with the right extensions and configurations. Here are some details on using VS Code for data analysis:

1. **Extensions for Data Analysis:** VS Code supports extensions that enhance its functionality. Some useful extensions for data analysis include:

    - Pandas and Data Visualization Extensions: There are extensions that enhance Pandas DataFrame exploration and provide interactive data visualization capabilities directly in VS Code.

2. **Integrated Terminal:** VS Code comes with an integrated terminal, allowing you to execute commands and scripts without leaving the IDE. This is useful for running Python scripts, executing shell commands, and managing virtual environments.

3. **Workspace Configuration:** You can configure VS Code workspaces to suit your data analysis projects. You can specify Python interpreter paths, set up virtual environments, and define project-specific settings.

2.3.3 Package: Jupyter

VS Code supports Jupyter notebooks, which allow you to create interactive documents. Jupyter notebooks are documents that can contain code cells (executable code), markdown cells (text explanations), and visualizations. They are particularly useful for data analysis, exploration, and sharing insights. To work with Jupyter notebooks in VS Code:

- Install the Jupyter extension if you haven't already.
- Open a Jupyter notebook (.ipynb file) in VS Code.
- Create and run code cells using the "Run Cell" button or keyboard shortcuts.
- Use markdown cells to provide explanations, headers, and text.

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.

The data for **Analyze Death Age Difference of Right Handers with Left Handers** has been collected through various GitHub repositories,mentioned as follows:

- death distribution data for the United States from the year 1999
- Rates of left-handedness digitized from a figure in this 1992 paper by Gilbert and Wysocki

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:

Secure Sockets Layer: The ssl module in Python provides the necessary tools to implement secure communication using SSL/TLS protocols. It is an integral part of ensuring data security and confidentiality when transmitting sensitive information over the internet.

pandas: Pandas is a foundational Python library that plays a crucial role in data analysis and manipulation. It provides high-level data structures and functions designed to make working with structured data intuitive and efficient. Pandas is widely used in data analysis projects to clean, transform, analyze, and visualize data, making it an essential tool for data scientists, analysts, and researchers.

pandas.read_csv(): “**pandas.read_csv”** is a powerful function provided by the Pandas library in Python that allows you to read data from a CSV (Comma-Separated Values) file and create a DataFrame, which is a two-dimensional tabular data structure. This function simplifies the process of importing structured data from CSV files for further analysis and manipulation.

Sample Code:

```
# import libraries
import ssl
import pandas as pd
import matplotlib.pyplot as plt


ssl._create_default_https_context = ssl._create_unverified_context

# load the data
data_url_1 = "https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df1/raw/aec88b30af87fad8d45da7e774223f91da
lefthanded_data = pd.read_csv(data_url_1)
```

3.2 Explore Left-Handedness Trends

The Data Analyst explores historical trends in left-handedness rates across different age groups. This involves analyzing the left-handedness rates dataset to identify patterns and trends.

Packages Used:

Matplotlib: Matplotlib is a popular Python library used for creating a wide range of static, interactive, and animated visualizations. It provides a flexible and customizable framework for generating high-quality graphs, charts, and plots for data visualization and exploration. Matplotlib's versatility and ease of use make it an essential tool for displaying data insights effectively.

Functions Used:

plt.subplot() : **The plt.subplot()** function in Matplotlib is used to create and arrange multiple subplots within a single figure. It provides a way to divide a figure into a grid of rows and columns and specify the position of each subplot within that grid. This allows you to create multiple plots side by side, facilitating easy comparison and visualization of different aspects of your data.

ax.plot() : **The ax.plot()** method is a fundamental function in Matplotlib that is used to create various types of plots on a specific set of axes within a figure. It is primarily used in the object-oriented interface of Matplotlib, where you create Figure and Axes objects to have greater control over plot customization and layout.

Sample Code:

```
# plot male and female left-handedness rates vs. age
%matplotlib inline
fig, ax = plt.subplots() # create figure and axis objects
ax.plot('Age', 'Female', data = lefthanded_data, marker = 'o') # plot "Female" vs. "Age"
ax.plot('Age', 'Male', data = lefthanded_data, marker = 'x') # plot "Male" vs. "Age"
ax.legend() # add a legend
ax.set_xlabel('Sex')
ax.set_ylabel('Age')
```

3.4 Rates of left-handedness over time

Since the study was done in 1986, the data after this conversion will be the percentage of people alive in 1986 who are left-handed as a function of the year they were born.

Sample Code:

```
# create a new column for birth year of each age
lefthanded_data['Birth_year'] = 1986 - lefthanded_data['Age']

# create a new column for the average of male and female
lefthanded_data['Average_lh'] = lefthanded_data[['Male', 'Female']].mean(axis=1)

# create a plot of the 'Mean_lh' column vs. 'Birth_year'
fig, ax = plt.subplots()
ax.plot('Birth_year', 'Average_lh', data = lefthanded_data) # plot 'Mean_lh' vs. 'Birth_year'
ax.set_xlabel('Birth_year') # set the x label for the plot
ax.set_ylabel('Average_lh') # set the y label for the plot
```

3.5 Apply Bayesian Analysis

Bayesian analysis is used to calculate the conditional probabilities of various scenarios related to age at death, handedness, and other factors.

The probability of dying at a certain age given that you're left-handed is **not** equal to the probability of being left-handed given that you died at a certain age. This inequality is why we need **Bayes' theorem**, a statement about conditional probability which allows us to update our beliefs after seeing evidence.

We want to calculate the probability of dying at age A given that you're left-handed. Let's write this in shorthand as P(A | LH). We also want the same quantity for right-handers: P(A | RH).

Here's Bayes' theorem for the two events we care about: left-handedness (LH) and dying at age A.

$$P(A|LH) = \frac{P(LH|A)P(A)}{P(LH)}$$

P(LH | A) is the probability that you are left-handed *given that* you died at age A. P(A) is the overall probability of dying at age A, and P(LH) is the overall probability of being left-handed. We will now calculate each of these three quantities, beginning with P(LH | A).

To calculate P(LH | A) for ages that might fall outside the original data, we will need to extrapolate the data to earlier and later years. Since the rates flatten out in the early 1900s and late 1900s, we'll use a few points at each end and take the mean to extrapolate the rates on each end. The number of points used for this is arbitrary, but we'll pick 10 since the data looks flat-ish until about 1910.

Sample Code:

```python
# import library
import numpy as np

# create a function for P(LH | A)
def P_lh_given_A(ages_of_death, study_year = 1990):
    """ P(Left-handed | ages of death), calculated based on the reported rates of left-handedness.
    Inputs: numpy array of ages of death, study_year
    Returns: probability of left-handedness given that subjects died in `study_year` at ages `ages_of_death` """

    # Use the mean of the 10 last and 10 first points for left-handedness rates before and after the start
    early_1900s_rate = lefthanded_data['Average_lh'][-10:].mean()
    late_1900s_rate = lefthanded_data['Average_lh'][:10].mean()
    middle_rates = lefthanded_data.loc[lefthanded_data['Birth_year'].isin(study_year - ages_of_death)]['Average_lh']
    youngest_age = study_year - 1986 + 10 # the youngest age is 10
    oldest_age = study_year - 1986 + 86 # the oldest age is 86

    P_return = np.zeros(ages_of_death.shape) # create an empty array to store the results
    # extract rate of left-handedness for people of ages 'ages_of_death'
    P_return[ages_of_death > oldest_age] = early_1900s_rate / 100
    P_return[ages_of_death < youngest_age] = late_1900s_rate / 100
    P_return[np.logical_and((ages_of_death <= oldest_age), (ages_of_death >= youngest_age))] = middle_rates / 100

    return P_return
```

3.6 When do people normally die?

To estimate the probability of living to an age A, we can use data that gives the number of people who died in a given year and how old they were to create a distribution of ages of death. If we normalize the numbers to the total number of people who died, we can think of this data as a probability distribution that gives the probability of dying at age A. The data we'll use for this is from the entire US for the year 1999 - the closest I could find for the time range we're interested in.

Function Used:

.dropna() : In Python, .dropna() is a method commonly used in pandas, a popular data manipulation library, to remove missing or NaN (Not a Number) values from a DataFrame or Series. This method is quite useful for cleaning and preprocessing data before analysis or modeling.

Sample Code:

```
# Death distribution data for the United States in 1999
data_url_2 = "https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f75f18ec/raw/62f3ec07514c7e31f5979beeca86f19991

# load death distribution data
death_distribution_data = pd.read_csv(data_url_2, sep='\t', skiprows=[1])
 (variable) death_distribution_data: DataFrame
death_distribution_data = death_distribution_data.dropna(subset = ['Both Sexes'])

# plot number of people who died as a function of age
fig, ax = plt.subplots()
ax.plot('Age', 'Both Sexes', data = death_distribution_data, marker='o') # plot 'Both Sexes' vs. 'Age'
ax.set_xlabel('Age')
ax.set_ylabel('Both Sexes')
```

3.7 The overall probability of left-handedness

Now that we have functions to calculate the probability of being age A at death given that you're left-handed or right-handed.

Packages Used:

NumPy: NumPy (Numerical Python) is a fundamental package in the Python scientific computing ecosystem. It provides support for large, multi-dimensional arrays and matrices, along with a wide range of mathematical functions to operate on these arrays efficiently. NumPy is widely used in various fields, including data analysis, scientific research, machine learning, and more.

Functions Used:

numpy.arange(): numpy.arange() is a function provided by the NumPy library in Python, and it is used to create an array of evenly spaced values within a specified range. It generates a sequence of numbers starting from a specified start value, incrementing by a specified step size, and stopping before a specified stop value.

Sample Code:

```
ages = np.arange(6, 115, 1) # make a list of ages of death to plot

# calculate the probability of being left- or right-handed for each
left_handed_probability = P_A_given_lh(ages, death_distribution_data)
right_handed_probability = P_A_given_rh(ages, death_distribution_data)

# create a plot of the two probabilities vs. age
fig, ax = plt.subplots() # create figure and axis objects
ax.plot(ages, left_handed_probability, label = "Left-handed")
ax.plot(ages, right_handed_probability, label = "Right-handed")
ax.legend() # add a legend
ax.set_xlabel("Age at death")
ax.set_ylabel(r"Probability of being age A at death")
```

3.8 Putting it all together

Now we have the means of calculating all three quantities we need: P(A), P(LH), and P(LH | A). We can combine all three using Bayes' rule to get P(A | LH), the probability of being age A at death (in the study year) given that you're left-handed. To make this answer meaningful, though, we also want to compare it to P(A | RH), the probability of being age A at death given that you're right-handed.

We're calculating the following quantity twice, once for left-handers and once for right-handers.

$$P(A|LH) = \frac{P(LH|A)P(A)}{P(LH)}$$

3.9 Moment of Truth

Finally, let's compare our results with the original study that found that left-handed people were nine years younger at death on average. We can do this by calculating the mean of these probability distributions in the same way we calculated P(LH) earlier, weighting the probability distribution by age and summing over the result.

$$\text{Average age of left-handed people at death} = \sum_A AP(A|LH)$$

$$\text{Average age of right-handed people at death} = \sum_A AP(A|RH)$$

Sample Code:

```
# calculate average ages for left-handed and right-handed groups
# use np.array so that two arrays can be multiplied
average_lh_age =  np.nansum(ages*np.array(left_handed_probability))
average_rh_age =  np.nansum(ages*np.array(right_handed_probability))

# print the average ages for each group
print("Average age of lefthanded" + str(average_lh_age))
print("Average age of righthanded" + str(average_rh_age))


# print the difference between the average ages
print("The difference in average ages is " + str(round(average_rh_age - average_lh_age, 1)) + " years.")
```

X sample_code + Tag

```
Average age of lefthanded67.24503662801027
Average age of righthanded72.79171936526477
The difference in average ages is 5.5 years.
```

4.1 Defining Visuals

Data visualization is presenting data in a graphical or pictorial format. It allows decision-makers to see visually presented analytics, so that they can grasp difficult concepts or identify new patterns. In interactive visualizations, technology can be used to dig in charts and graphs for more detail, interactively modifying what data one can see and how it works.

Because of the way in which the human brain processes information, it is easier to visualize large amounts of complex data using charts or graphs than to poring overspreadsheets or reports. Data visualization is a quick, easy and universal way of conveying concepts. Data visualization can also:

- Identify areas that need attention or improvement.
- Clarify which factors influence customer behaviour.
- Help you understand which products to place where.
- Predict sales volumes.

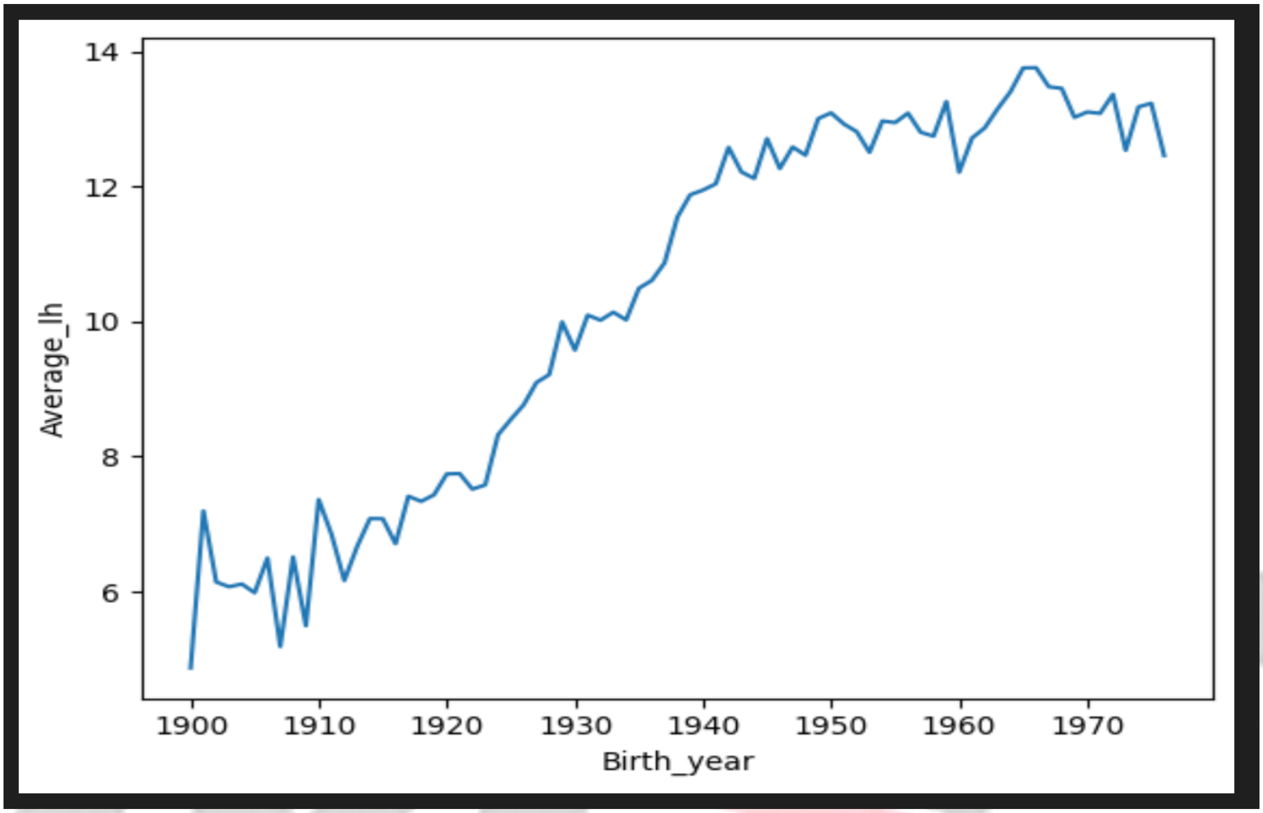
Visualization in Python is commonly done using various libraries that provide tools for creating graphical representations of data. Some of the most popular visualization libraries in Python include:

1. Matplotlib: Matplotlib is one of the foundational plotting libraries in Python. It provides a wide range of plotting functionalities, from basic line plots to complex visualizations. It can be used for creating static, animated, and interactive plots.

2. Seaborn: Seaborn is built on top of Matplotlib and provides a higher-level interface for creating attractive statistical visualizations. It's especially useful for creating complex visualizations with less code.

SAMPLE SCREENSHOTS AND OBSERVATIONS

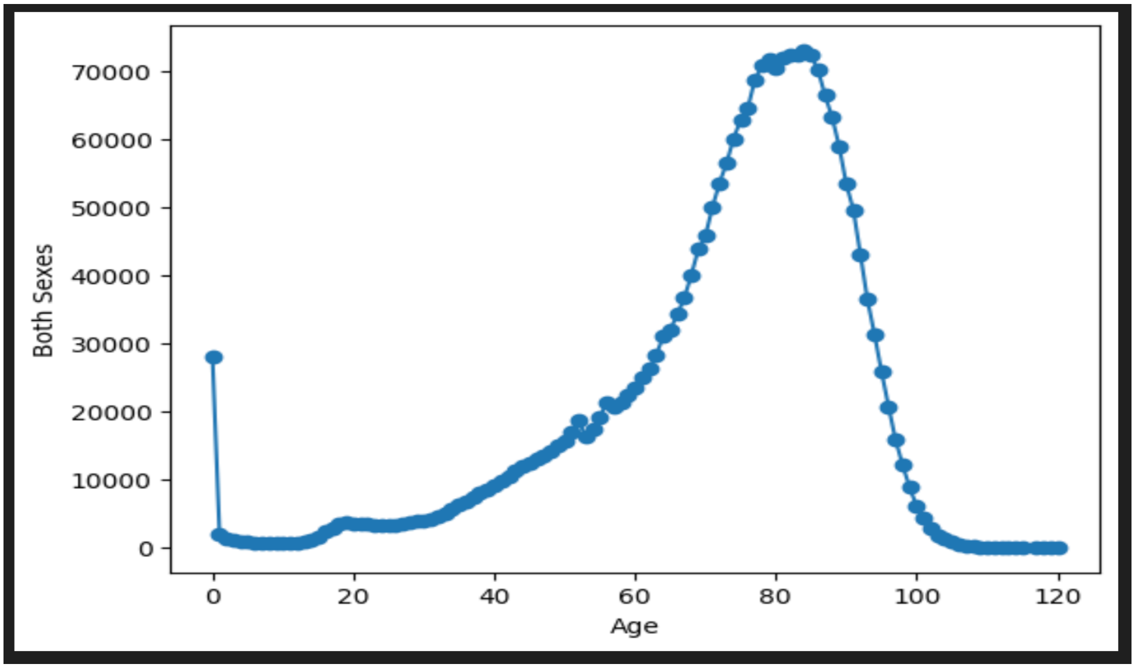
4.2 Rates of left-handedness over time



Observation:

The prevalence of left-handedness has risen over the specified duration. Notably, there has been a significant spike between the years 1930 and 1960.

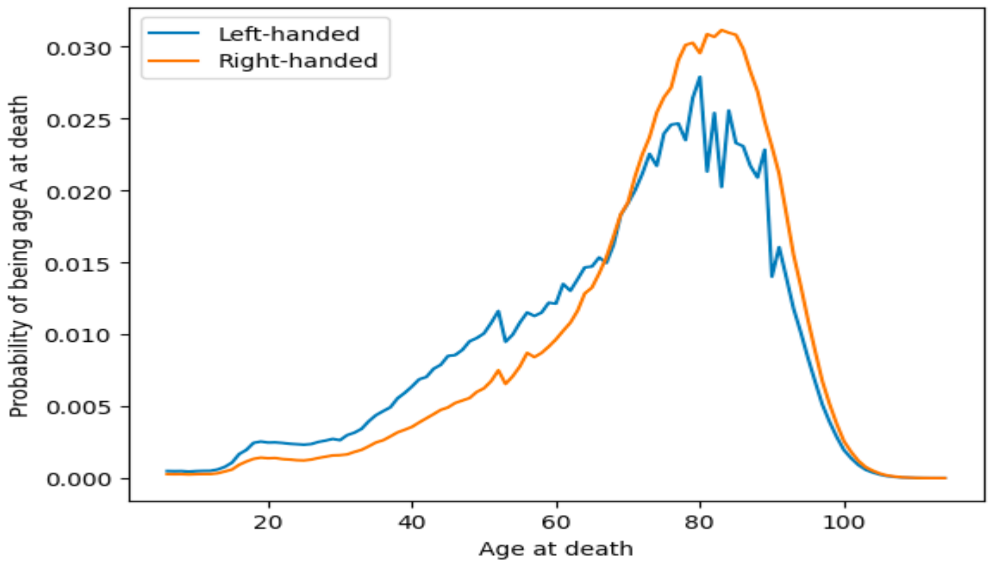
4.3 Death distribution for the US in 1999



Observation:

The age group with the highest number of deaths in 1999 was observed to be 85 years. This age had the highest count of deaths across all age categories, as indicated by the data. However, the rate of increase appears to slow down after a certain point, suggesting that there might be an age range where the mortality rate plateaus or increases at a slower pace. Overall, these observations provide insights into the distribution of deaths across age groups and gender for the United States in 1999. It's important to note that these observations are based on the available data and might be influenced by various factors such as population demographics and healthcare advancements.

4.4 Comparison of left-handed vs right-handed (age at death)



Observation:

The trend in the data unmistakably indicates that the age at which left-handed individuals pass away is lower compared to their right-handed counterparts. This suggests that left-handed individuals tend to experience a shorter lifespan. This could potentially be attributed to a range of factors, including genetic predispositions, lifestyle differences, or other external influences that might impact their overall health and mortality. Further investigation into these factors could provide deeper insights into the observed trend.

Average age of lefthanded67.24503662801027

Average age of righthanded72.79171936526477

The difference in average ages is 5.5 years.

CONCLUSION AND FUTURE SCOPE

We got a pretty big age gap between left-handed and right-handed people purely as a result of the changing rates of left-handedness in the population, which is good news for left-handers: you probably won't die young because of your sinisterness. The reported rates of left-handedness have increased from just 3% in the early 1900s to about 11% today, which means that older people are much more likely to be reported as right-handed than left-handed, and so looking at a sample of recently deceased people will have more old right-handers.

Our number is still less than the 9-year gap measured in the study. It's possible that some of the approximations we made are the cause:

We used death distribution data from almost ten years after the study (1999 instead of 1991), and we used death data from the entire United States instead of California alone (which was the original study).

We extrapolated the left-handedness survey results to older and younger age groups, but it's possible our extrapolation wasn't close enough to the true rates for those ages.

One thing we could do next is figure out how much variability we would expect to encounter in the age difference purely because of random sampling: if you take a smaller sample of recently deceased people and assign handedness with the probabilities of the survey, what does that distribution look like? How often would we encounter an age gap of nine years using the same data and assumptions? We won't do that here, but it's possible with this data and the tools of random sampling.

To finish off, let's calculate the age gap we'd expect if we did the study in 2018 instead of in 1990. The gap turns out to be much smaller since rates of left-handedness haven't increased for people born after about 1960. Both the National Geographic study and the 1990 study happened at a unique time - the rates of left-handedness had been changing across the lifetimes of most people alive, and the difference in handedness between old and young was at its most striking.

With regards to the future work, the firm aims at regularly updating the data with time and integrating it with their systems so as to continually draw conclusions and analyze the results.

REFERENCES

Data Collection

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

- https://www.khanacademy.org/math/statistics-probability
- https://www.probabilitycourse.com/
- https://cpcontents.adobe.com/public/newlearner/newlearner_1c07ce5f.html?i_qp_user_id=21140072&accountId=121793#/course/7384882/overview?ci_id=8030686
- https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df1/raw/aec88b30af87fad8d45da7e774223f91dad09e88/lh_data.csv
- https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f75f18ec/raw/62f3ec07514c7e31f5979beeca86f19991540796/cdc_vs00199_table310.tsv

Programming References

The following websites have been referred for Python coding and Visualization tutorials:

- https://matplotlib.org/stable/tutorials/index
- https://matplotlib.org/stable/api/index.html
- https://seaborn.pydata.org/examples/index.html
- https://www.w3schools.com/python/
- https://www.python.org/doc/

Project Notebook Link:

https://drive.google.com/file/d/1vC9czm75xote6Pa5IxOk76FNeeb3utpx/view?usp=sharing