**Final Report of Traineeship Program 2023**

*On*

# *"ANALYZING DEATH AGE DIFFERENCE OF RIGHT HANDERS WITH LEFT HANDERS"*

**MEDTOUREASY**

20th June 2023

Anusha Arra

# ACKNOWLDEGMENTS

The traineeship opportunity I was given with MedTourEasy was a tremendous shift for my personal and professional growth as well as for learning and comprehending the complexities of the topic of Data Analytics. I'm grateful that I had the chance to speak with so many experts who helped me with the traineeship project and gave this an incredible learning curve.

First and foremost, I would like to express my sincere gratitude to the Training & Development Team at MedTourEasy for providing me with the chance to complete my traineeship at their prestigious company. I also want to thank the team for helping me grasp the specifics of the Data Analytics profile and training me in it so that I can complete the project correctly and to the satisfaction of the customer, as well as for their important time despite their hectic schedules.

I also want to thank the MedTourEasy staff and my coworkers for creating a productive and welcoming work atmosphere.

# TABLE OF CONTENTS

# ABSTRACT

A National Geographic survey in 1986 resulted in over a million responses that included age, sex, and hand preference for throwing and writing. Researchers Avery Gilbert and Charles Wysocki analyzed this data and noticed that rates of left-handedness were around 13% for people younger than 40 but decreased with age to about 5% by the age of 80. They concluded based on analysis of a subgroup of people who throw left-handed but write right-handed that this age-dependence was primarily due to changing social acceptability of left-handedness.

This means that the rates aren't a factor of age specifically but rather of the year you were born, and if the same study was done today, we should expect a shifted version of the same distribution as a function of age. Ultimately, we'll see what effect this changing rate has on the apparent mean age of death of left-handed people by plotting the rates of left-handedness as a function of age.

Therefore, we will explore this phenomenon using age distribution data to see if we can reproduce a difference in average age at death purely from the changing rates of left-handedness over time, refuting the claim of early death for left-handers. We will use pandas and Bayesian statistics to analyze the probability of being a certain age at death given that you are reported as left-handed or right-handed.

# 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 will explore the phenomenon of using age distribution data to see if we can reproduce a difference in average age at death purely from the changing rates of left-handedness over time, refuting the claim of early death for left-handers. We use pandas and Bayesian statistics to analyze the probability of being a certain age at death given that you are reported as left-handed or right-handed.

## 1.3 Objectives and Deliverables

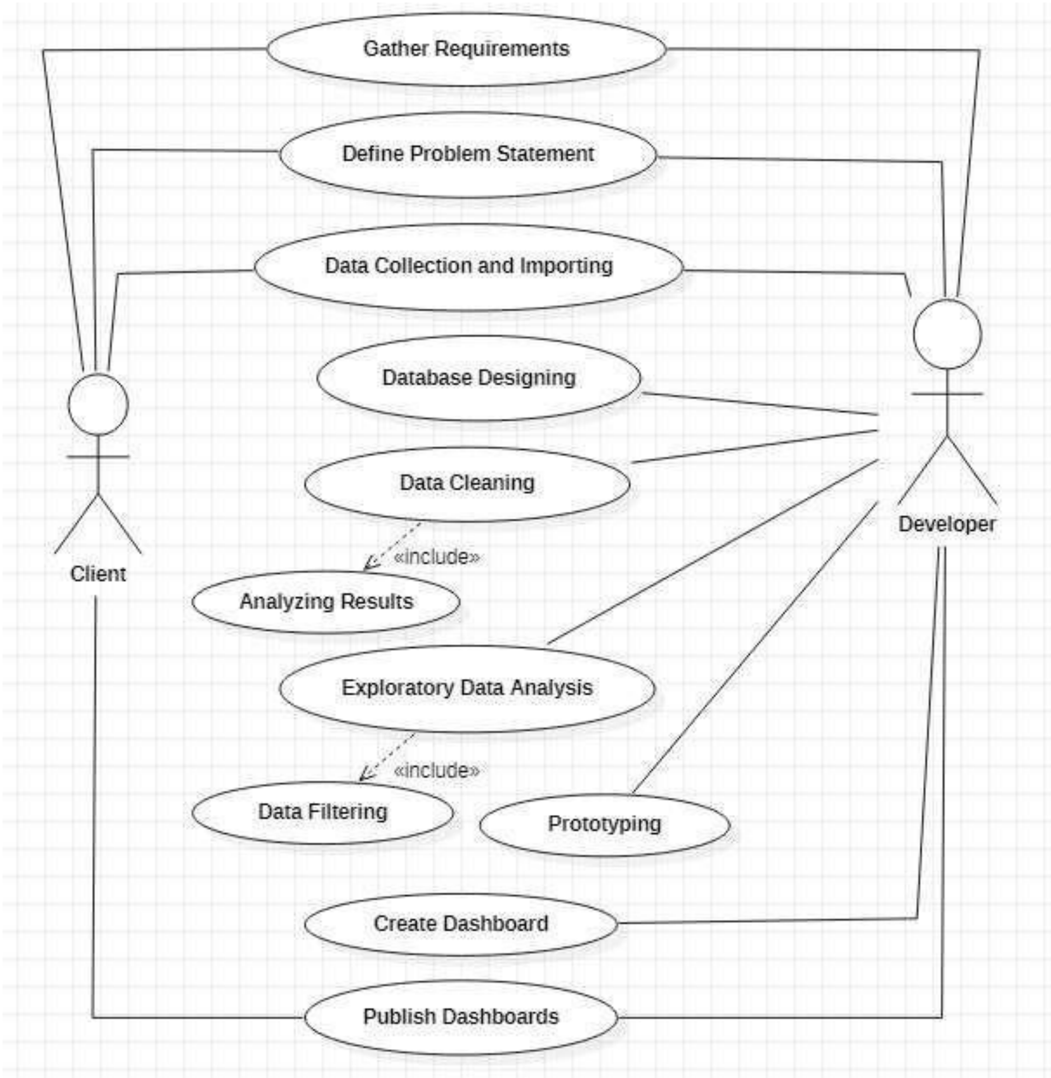
This project focuses on creating easily informative, understandable and clear plots along probability, Bayesian statistics using the coding language Python and packages like NumPy and pandas. Matplotlib is used to visualize these statistics which will enable us to analyze the situation and draw conclusions regarding the data given. It will deliver the exact statistics and conclusions between the age of the death of left handed and right handed people.

# METHODOLOGY

2.1 Flow of the Project

The project followed the following steps to accomplish the desired objectives anddeliverables. 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 Stakeholders and Data Analyst. The analyst 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. We have plots given at every step to analyze the calculations. Finally, the difference between the average age is calculated – right handers and left handers.

## 2.3 Language and Platform Used

### 2.3.1 Language: Python

Python is a versatile and widely-used programming language with a plethora of applications. It is an interpreted, high-level, and general-purpose programming language. Created by Guido van Rossum its first version was released in 1991.
Python is known for its readability and simplicity, making it an excellent choice for beginners and experienced developers alike.It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

• Python has a large standard library that provides numerous built-in functions and modules for various tasks.
• The language has a dynamic type system, which means variables do not need to be declared with a specific data type.
• It is widely used in web development, scientific computing, artificial intelligence, data analysis, automation, and more.
• Python has extensive community support, and a vast number of third-party libraries and frameworks are available through the Python Package Index (PyPI).
• The language is open-source, allowing developers to contribute to its ongoing development and improvement.
• Python is compatible with major operating systems, including Windows, macOS, and various Linux distributions.
• It provides robust support for integration with other languages like C, C++, and Java, allowing developers to combine Python's ease of use with performance-critical code written in other languages.
• Python is the language of choice for many AI and machine learning projects due to its simplicity, ease of use, and the popularity of libraries like TensorFlow, Keras, and PyTorch.

### 2.3.2 IDE: Jupyter Notebook

Jupyter Notebook is an interactive computing environment that allows users to create and share documents containing live code, visualizations, explanatory text, and multimedia elements. Originally known as IPython Notebook, it was later renamed

Jupyter to encompass a broader range of programming languages (Julia, Python, and R). Jupyter Notebook supports over 40 programming languages and is widely used in data science, scientific research, education, and data analysis. It provides a web-based interface where users can execute code in individual cells. Notebooks can contain not only code but also rich text, images, equations (using LaTeX syntax), and interactive widgets. This combination of code and multimedia elements makes it a powerful tool for presenting and sharing data analysis and research findings. It integrates seamlessly with popular data visualization libraries like Matplotlib, Seaborn, and Plotly. Users can generate interactive and static plots, charts, and graphs directly within the notebook.

### 2.3.3 Library: NumPy

NumPy is a powerful Python library for numerical computations, providing essential tools for efficient array operations and mathematical functions. With NumPy, users can easily create, manipulate, and process multi-dimensional arrays, enabling seamless integration with other scientific libraries like SciPy and Matplotlib. Its array-oriented programming approach accelerates mathematical computations, making it ideal for scientific and data analysis tasks. NumPy's intuitive syntax and broadcasting capabilities simplify array operations, significantly improving code readability and performance.

### 2.3.4 Library: Pandas

Pandas is a versatile and user-friendly Python library designed for data manipulation and analysis. Built on top of NumPy, Pandas introduces powerful data structures like Series and DataFrame, making it easy to handle structured data and perform data wrangling tasks efficiently. It offers a wide range of functions to clean, reshape, and merge datasets, as well as tools for data exploration, grouping, and aggregation. With Pandas, users can handle missing values, extract meaningful insights from large datasets, and prepare data for further analysis or visualization. Its intuitive API and seamless integration with other libraries have made Pandas an essential tool for data scientists, analysts, and researchers worldwide.

### 2.3.5 Library: MatPlotLib

Matplotlib is a popular Python library widely used for creating high-quality visualizations and plots. It provides a flexible and user-friendly interface for generating a variety of charts, graphs, and plots, including line plots, bar charts, scatter plots, histograms, and more. With its extensive customization options, users can tailor the appearance of visualizations to suit their needs, adding labels, titles, legends, and color palettes. Matplotlib's integration with NumPy and Pandas allows for seamless data visualization from arrays or data frames. Whether for exploratory data analysis, scientific research, or data presentation, Matplotlib remains a fundamental tool in the data science and visualization ecosystem.

# 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 has been collected through various GitHub repositories. We use two datasets: death distribution data for the United States from the year 1999 (source website here) and 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 websitesand data repositories). This data can then be manipulated, aggregated, filtered according to the requirements and needs of the project.

**read.csv ()**: It is a wrapper function for read.table() that mandates a comma as separator and uses the input file's first line as header that specifies the table's columnnames. Thus, it is an ideal candidate to read CSV files. It has an additionalparameter of url() which is used to pull live data directly from GitHub repository.

**%matplotlib inline**: This is a magic command in Jupyter Notebook that allows plots to be displayed directly in the output cells of the notebook.

**plt.subplots():** This line creates a figure (fig) and an axis (ax) object. The figure is like a canvas where plots will be drawn, and the axis represents a single plot within the figure.

**ax.plot():** This function is used to plot data on the axis.

**ax.legend():** This function adds a legend to the plot, displaying the labels specified in the plot() function.

**ax.set_xlabel() and ax.set_ylabel**(): These functions set the labels for the x-axis and y-axis, respectively, providing a clear description of the data being visualized.

```
Code:

# import libraries

import pandas as pd

import matplotlib.pyplot as plt

# load the data

data_url_1 =
"https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df
1/raw/aec88b30af87fad8d45da7e774223f91dad09e88/lh_data.csv"

lefthanded_data = pd.read_csv(data_url_1)

# plot male and female left-handedness rates vs. age

%matplotlib inline

fig, ax = plt.subplots() # create figure and axis objects

ax.plot(lefthanded_data["Age"], lefthanded_data["Female"], label="Female",
marker = 'o') # plot "Female" vs. "Age"

ax.plot(lefthanded_data["Age"], lefthanded_data["Male"], label="Male",
marker = 'x') # plot "Male" vs. "Age"

ax.legend() # add a legend

ax.set_xlabel("Age")

ax.set_ylabel("Left-handedness Rate")


Output:
 Text(0, 0.5, 'Left-handedness Rate')
```

## 3.3 Calculating rates of left handedness over time

Let's convert this data into a plot of the rates of left-handedness as a function of the year of birth, and average over male and female to get a single rate for both sexes.

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.

```
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['Mean_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(lefthanded_data['Birth_year'], lefthanded_data['Mean_lh'],
marker='o') # plot 'Mean_lh' vs. 'Birth_year'
ax.set_xlabel('Birth Year') # set the x label for the plot
ax.set_ylabel('Mean Left Handedness') # set the y label for the plot

Output:

Text(0, 0.5, 'Mean Left Handedness')
```

## 3.4 Applying Bayes Rule

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.

$$P(A|LH) = P(LH|A)P(A)P(LH)$$

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.

```
Code:

# import library
import pandas as pd
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 = np.mean(lefthanded_data.tail(10)['Mean_lh'])
    late_1900s_rate = np.mean(lefthanded_data.head(10)['Mean_lh'])
    middle_rates =
lefthanded_data.loc[lefthanded_data['Birth_year'].isin(study_year -
ages_of_death)]['Mean_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
```

## 3.5 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

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.

In this block, we'll load in the death distribution data and plot it. The first column is the age, and the other columns are the number of people who died at that age.

```
Code:

# Death distribution data for the United States in 1999
data_url_2 =
"https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f75f18ec/raw
/62f3ec07514c7e31f5979beeca86f19991540796/cdc_vs00199_table310.tsv"

# load death distribution data
death_distribution_data = pd.read_csv(data_url_2, sep='\t', skiprows=[1])

# drop NaN values from the `Both Sexes` column
death_distribution_data.dropna(subset=['Both Sexes'], inplace=True)

# plot number of people who died as a function of age
fig, ax = plt.subplots()
ax.plot(death_distribution_data['Age'], death_distribution_data['Both Sexes'],
marker='o') # plot 'Both Sexes' vs. 'Age'
ax.set_xlabel('Age')
ax.set_ylabel('No. of Deaths')

Output:

Text(0, 0.5, 'No. of Deaths')
```

## 3.6 Calculating overall probability of left handedness

In the previous code block we loaded data to give us P(A), and now we need P(LH). P(LH) is the probability that a person who died in our particular study year is left-handed, assuming we know nothing else about them. This is the average left-handedness in the population of deceased people, and we can calculate it by summing up all of the left-handedness probabilities for each age, weighted with the number of deceased people at each age, then divided by the total number of deceased people to get a probability. In equation form, this is what we're calculating, where N(A) is the number of people who died at age A (given by the dataframe death_distribution_data):

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

```
Code:

def P_lh(death_distribution_data, study_year = 1990): # sum over P_lh for
each age group
    """ Overall probability of being left-handed if you died in the study
year
    Input: dataframe of death distribution data, study year
    Output: P(LH), a single floating point number """
    p_list = death_distribution_data['Both Sexes'] *
P_lh_given_A(death_distribution_data['Age'], study_year)
    # multiply number of dead people by P_lh_given_A
    p = np.sum(p_list) # calculate the sum of p_list
    return p / death_distribution_data['Both Sexes'].sum() # normalize to
total number of people (sum of death_distribution_data['Both Sexes'])

print(P_lh(death_distribution_data))

Output:

0.07766387615350638
```

## 3.7 Probability of Dying

### 3.7.1 While being left handed

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.

$$P(A|LH)=P(LH|A)P(A)P(LH)$$

```
Code:

def P_A_given_lh(ages_of_death, death_distribution_data, study_year = 1990):
    """ The overall probability of being a particular `age_of_death` given that
you're left-handed """
    P_A = death_distribution_data['Both Sexes'][ages_of_death] /
np.sum(death_distribution_data['Both Sexes'])
    P_left = P_lh(death_distribution_data, study_year) # use P_lh function to
get probability of left-handedness overall
    P_lh_A = P_lh_given_A(ages_of_death, study_year) # use P_lh_given_A to get
probability of left-handedness for a certain age
    return P_lh_A*P_A/P_left
```

### 3.7.1 While being right handed

```
Code:

def P_A_given_rh(ages_of_death, death_distribution_data, study_year = 1990):
    """ The overall probability of being a particular `age_of_death` given that
you're right-handed """
    P_A = death_distribution_data['Both Sexes'][ages_of_death] /
np.sum(death_distribution_data['Both Sexes'])
    P_right = 1 - P_lh(death_distribution_data, study_year) # either you're
left-handed or right-handed, so P_right = 1 - P_left
    P_rh_A = 1 - P_lh_given_A(ages_of_death, study_year) # P_rh_A = 1 - P_lh_A
    return P_rh_A*P_A/P_right
```

## 3.8 Plotting the distributions of conditional probabilities

Now that we have functions to calculate the probability of being age A at death given that you're left-handed or right-handed, let's plot these probabilities for a range of ages of death from 6 to 120.

Notice that the left-handed distribution has a bump below age 70: of the pool of deceased people, left-handed people are more likely to be younger.

```
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")

Output:

Text(0, 0.5, 'Probability of being age A at death')
```

## 3.9 Moment of truth: age of left and right-handers at death

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.

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

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

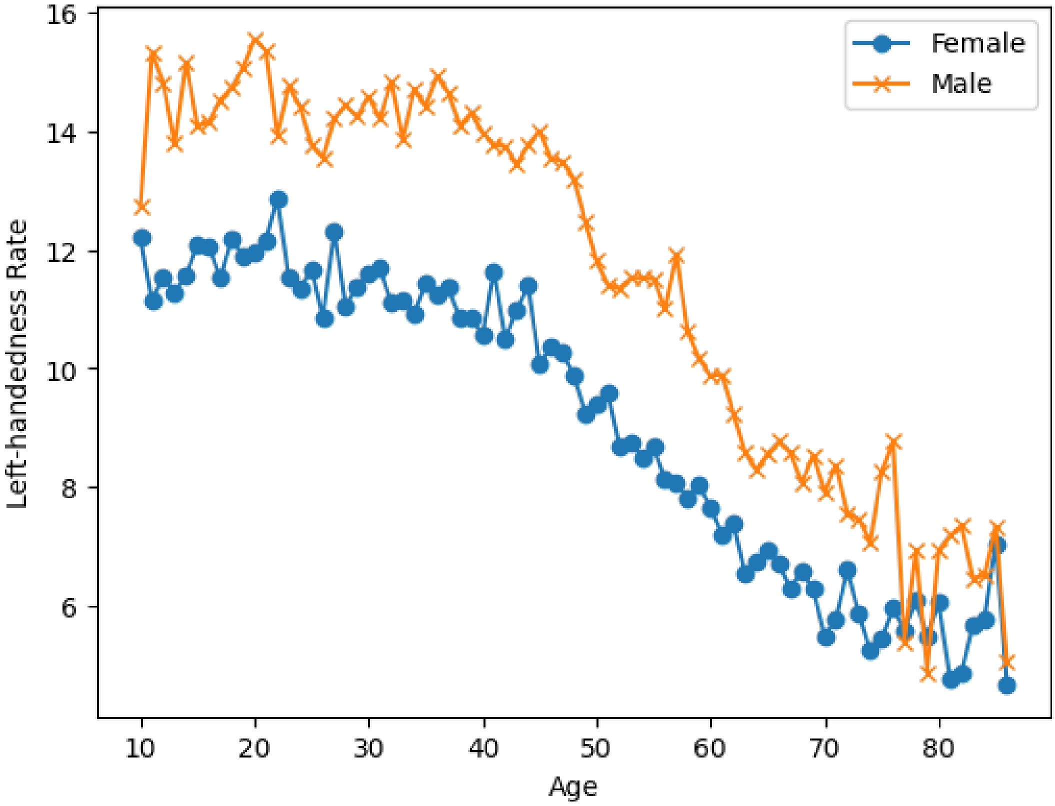
```
Code:

# 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(P_A_given_lh(ages,
death_distribution_data)))
average_rh_age =  np.nansum(ages * np.array(P_A_given_rh(ages,
death_distribution_data)))

# print the average ages for each group
print("Average age for left-handers:", round(average_lh_age, 2))
print("Average age for right-handers:", round(average_rh_age, 2))

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

Output:

Average age for left-handers: 67.25
Average age for right-handers: 72.79
The difference in average ages is -5.5 years.
```

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.

```
Code:

# Calculate the probability of being left- or right-handed for all ages
left_handed_probability_2018 = P_A_given_lh(ages, death_distribution_data,
study_year=2018)
right_handed_probability_2018 = P_A_given_rh(ages, death_distribution_data,
study_year=2018)

# calculate average ages for left-handed and right-handed groups
average_lh_age_2018 = np.nansum(ages*np.array(left_handed_probability_2018))
average_rh_age_2018 = np.nansum(ages*np.array(right_handed_probability_2018))

print("The difference in average ages is " +
      str(round(average_rh_age_2018 - average_lh_age_2018, 1)) + " years.")

Output:

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

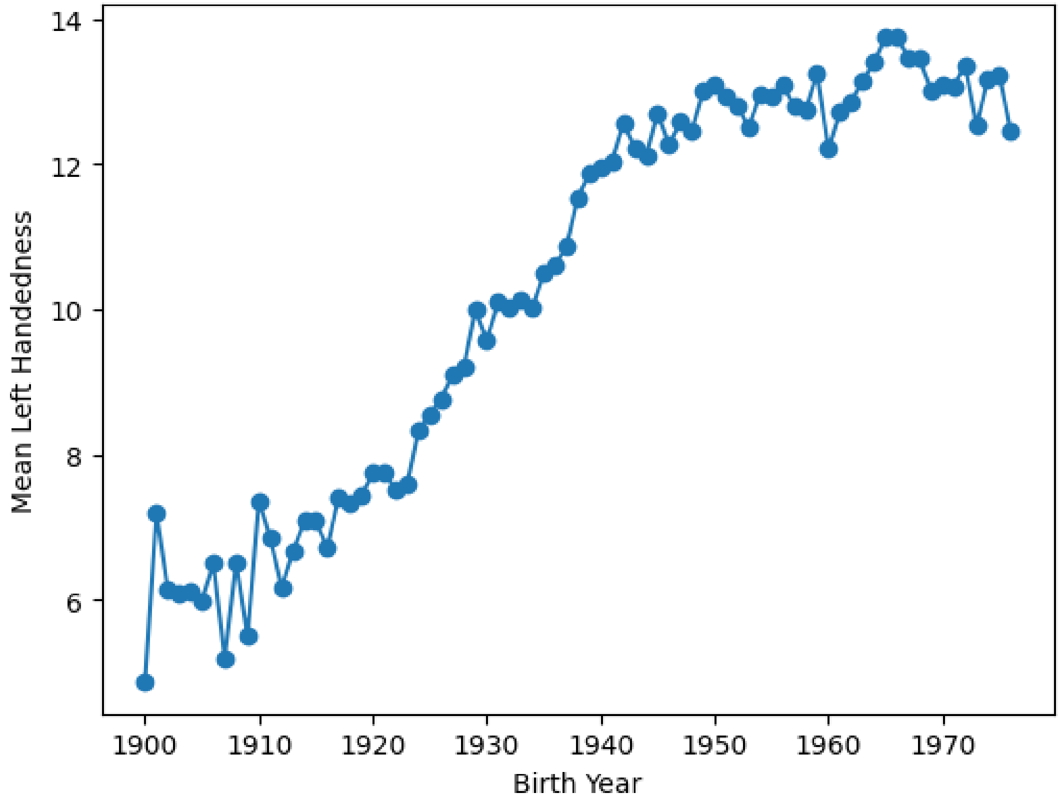
# VISUALIZATIONS

4.1 Visualization of Left handedness Rate in Male vs Female



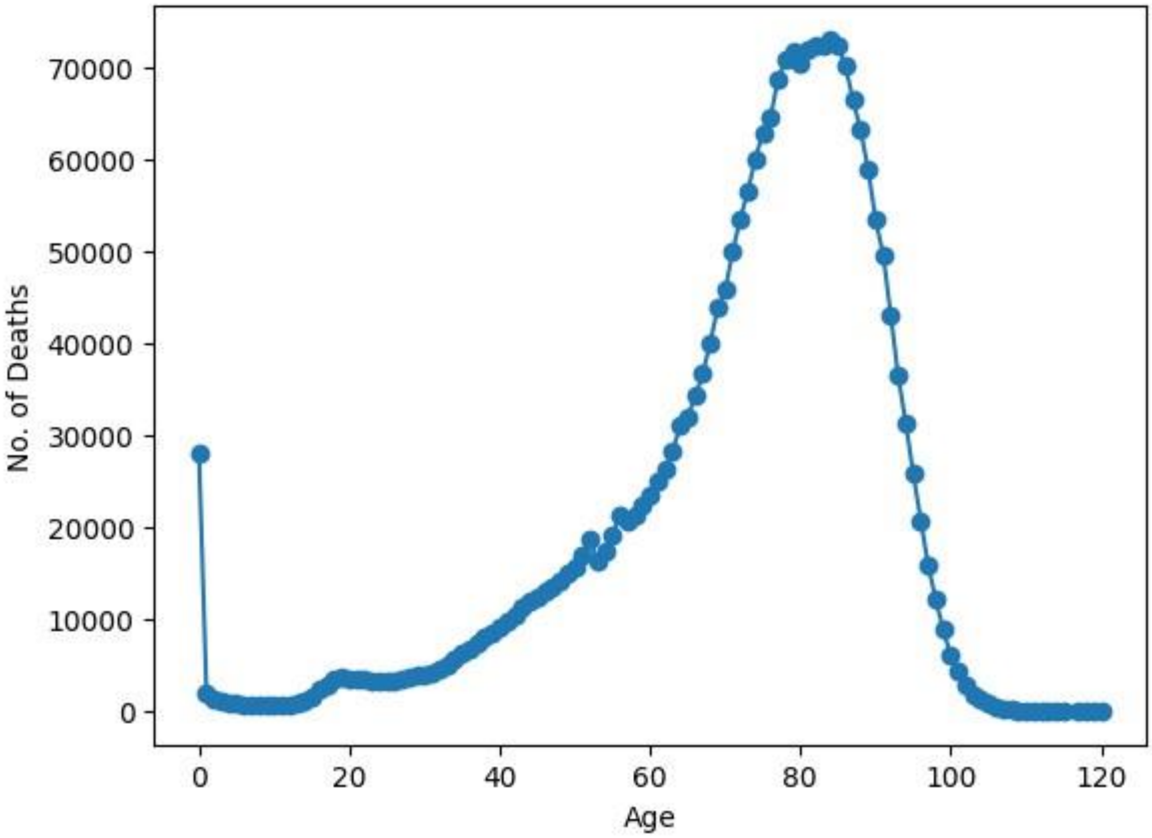

The above visualization shows the rate of lefthandedness in men vs women. They are indicated by two different markers: Orange depicting men and Blue depicting women.
This graph is a clear visualization of section 3.2.

4.2 Rates of Left Handedness over time



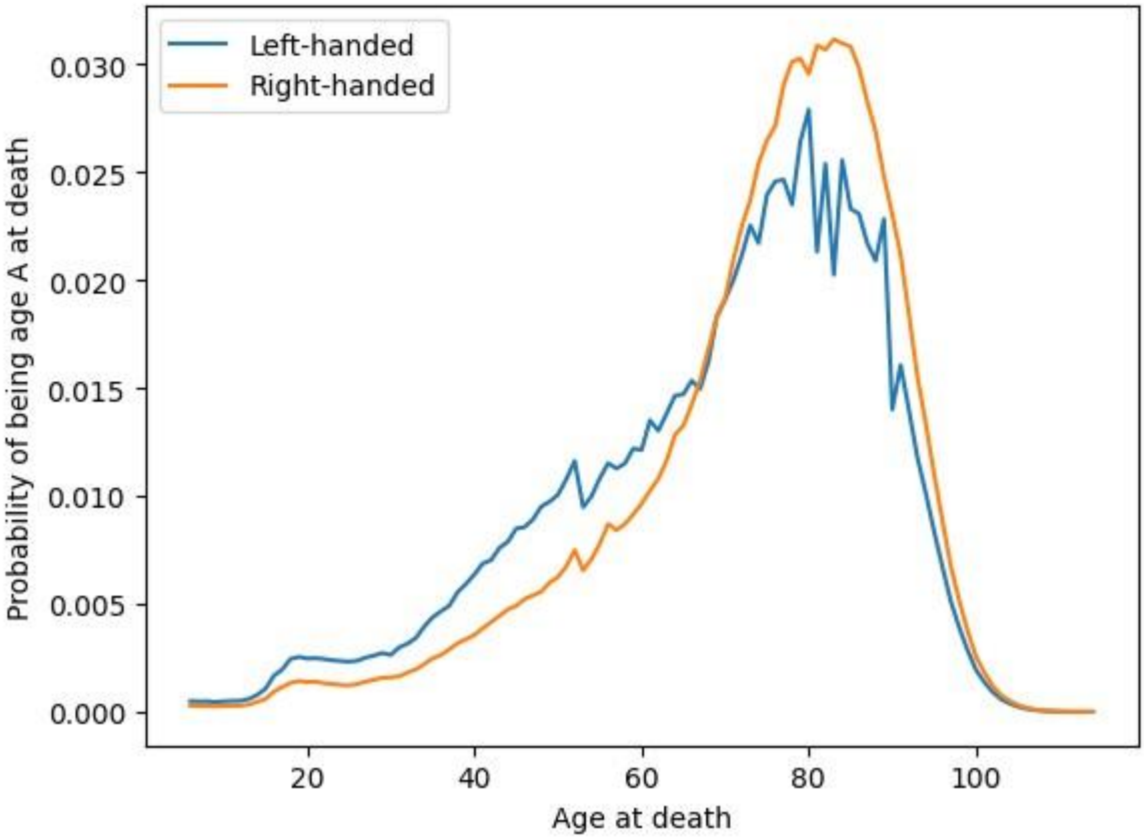

This is a depiction of the mean left handedness rate of both men and women along with the birth year on the X-Axis. It is a clear visualization of section 3.3. It is depicted as a blue line with ‘o’ markers.

## 4.3 Distribution of death rates




This is a distribution of the number of deaths among both men and women along with their age on the X-Axis. It is a clear visualization of section 3.5. It is depicted as a blue line with ‘o’ markers.

4.3 Probability distribution of age of death




This is a distribution of the probability of being an age A at death among both men and women along with their age of death on the X-Axis. Right-handed people are depicted by orange and left handed people are depicted by blue. It is a clear visualization of section 3.7. They are depicted as orange and blue lines.

# CONCLUSION AND FUTURE SCOPE

Conclusion

In this study, we explored the phenomenon of changing rates of left-handedness over time and its potential impact on the average age at death for left-handers. By analyzing age distribution data and using Bayesian statistics, we found evidence that the decreasing rates of left-handedness with age are primarily influenced by the year of birth rather than age itself. This suggests that the social acceptability of left-handedness has played a significant role in shaping the observed rates. Furthermore, our analysis did not support the claim of early death for left-handers. The apparent difference in average age at death for left-handers compared to right-handers can largely be attributed to the changing rates of left-handedness over different birth cohorts.

Future Scope

While this study provided valuable insights into the relationship between left-handedness rates and age, there are several avenues for further investigation. Firstly, exploring the underlying societal factors and cultural influences that contribute to the changing rates of left-handedness over time would deepen our understanding of this phenomenon.

Additionally, conducting similar studies in different regions and cultures could provide a more comprehensive and global perspective on the impact of social acceptability on left-handedness rates and its association with longevity.

Moreover, incorporating more diverse demographic variables, such as socioeconomic status and educational background, could help elucidate other potential factors affecting left-handedness rates and age-related differences.

Overall, further research on left-handedness and its implications for health outcomes and life expectancy can offer valuable insights into human behavior, evolution, and societal norms.

# REFERENCES

Data Collection

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


a. https://www.cdc.gov/nchs/data/statab/vs00199_table310.pdf

b. https://pubmed.ncbi.nlm.nih.gov/1528408/

c. https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df1/raw/aec88b30af87fad8d45da7e774223f91dad09e88/lh_data.csv

d. https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f75f18ec/raw/62f3ec07514c7e31f5979beeca86f19991540796/cdc_vs00199_table310.tsv


Programming References

The following websites have been referred for Python coding:


a. https://docs.python.org/3/

b. https://numpy.org/doc/1.25/

c. https://pandas.pydata.org/pandas-docs/stable/

d. https://matplotlib.org/stable/index.html

e. https://docs.jupyter.org/en/latest/