

**ANALYZING DEATH AGE DIFFERENCE OF RIGHT-
HANDERS WITH LEFT-HANDERS**

BY

UNWANA EKENG ITA

A FINAL REPORT OF TRAINEESHIP PROGRAM 2024

SUBMITTED

TO

MEDTOUREASY



28th FEBRUARY 2024

ACKNOWLEDGMENTS

I'm truly grateful for the enriching traineeship experience at MedTourEasy, which provided invaluable insights into Data Analytics, fostering both personal and professional growth. I extend my heartfelt thanks to the Training & Development Team for granting me this opportunity and for their guidance throughout the project. Special appreciation goes to the team for their dedication in imparting detailed knowledge and training, enabling me to execute the project effectively.

Table of Contents

ACKNOWLEDGMENTS	2
1.1 About the Company	5
1.2 About the Project	5
1.3 Objectives	5
2. METHODOLOGY	6
2.1 Data Acquisition and Cleaning:	6
2.2 Visualization:	6
2.3 Analysis:	6
2.4 Reporting and Recommendations:	7
3. IMPLEMENTATION	8
3.1 Language and Platform Used.....	8
3.2 Gathering Requirements and Defining Problem Statement.....	8
3.3 Data Collection and Importing.....	8
3.4 Data Source	8
3.5 Data Description	9
3.6 Packages Used	9
4. RESULTS	21
5. CONCLUSION.....	22
6. RECOMMENDATIONS.....	23
7. REFERENCES	25

1. INTRODUCTION

The relationship between handedness and various health outcomes has been a topic of ongoing research. While some studies have suggested potential associations between handedness and certain health conditions, including neurological and immune disorders, the link remains inconclusive. This study aimed to contribute to this research by investigating the potential relationship between handedness and lifespan, specifically focusing on the United States.

We utilized mortality data to calculate and compare the probabilities of dying at different ages for individuals identified as left-handed and right-handed. Additionally, we estimated the average age at death for each group in two different years – 1990 and 2018 – to assess any potential changes over time.

By performing this analysis, we hope to gain insights into whether handedness plays a role in lifespan. The findings from this study can contribute to the broader understanding of the factors influencing human longevity and inform future research directions in this area.

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

In this project, the age distribution data is explored to see whether 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 of left-handers. This project uses pandas a Python library 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

This project has the following objectives:

1. Investigate whether the difference in average age at death between left-handed and right-handed individuals can be explained solely by changes in the prevalence of left-handed over time. This will be done by using age distribution data and changing rates of left-handedness to see if the observed differences in average age at death can be purely explained by demographics
2. Investigate the claim that left-handed people die earlier than right-handed people.

2. METHODOLOGY

This project employed a computational approach to investigate the potential relationship between handedness and lifespan using publicly available historical death distribution data from the United States. Here's a breakdown of the key methods used:

2.1 Data Acquisition and Cleaning:

Loaded data for lefthanded rates and ages for males and females, and historical death distribution data from the United States (1999) in CSV and TSV format respectively.

Identified missing values and removed them to ensure data consistency, added two new columns, one for birth year and one for mean left-handedness rates for the left-handed data. Dropped missing values for both sexes column in the death distribution data.

2.2 Visualization:

Visualized the left-handedness rates versus the age for males and females, the mean left-handed rates versus the birth year, and the death distribution, examining the number of deaths at different ages across the population.

2.3 Analysis:

Rates of Left-handedness Over Time:

Converted the data into a plot of left-handedness rates as a function of the year of birth.

Averaged left-handedness rates for males and females to obtain a single rate for both sexes.

Probability Analysis:

Calculated the probability of being left-handed or right-handed for each age at death.

Plotted the probabilities of being left-handed and right-handed as a function of age at death.

Average Age at Death:

Calculated the mean age at death for left-handers and right-handers.

Compared the average ages between left-handers and right-handers to identify any differences.

Temporal Analysis:

Redid the calculations for probability and average age at death using data from a different study year (2018).

Analyzed the difference in average ages between left-handers and right-handers in 2018 compared to 1990.

The analysis provides valuable insights into the relationship between handedness and age at death. Overall, left-handedness rates and average ages at death vary over time and may be influenced by various factors. Further research and analysis could explore additional variables and investigate potential causal relationships.

2.4 Reporting and Recommendations:

Generated a comprehensive report documenting the analysis steps, explanations of the code, and its implementations. The flow is designed to build upon each step, starting with data preparation and visualization, followed by the core analysis involving probability and average age calculations both individually and across different years. The project then concludes with reporting and recommendations for further exploration.

3. IMPLEMENTATION

3.1 Language and Platform Used

Language: Python

This project utilized Python, a versatile and popular programming language widely used in data analysis and scientific computing. Its readability, extensive libraries, and large community make it a suitable choice for various data manipulation and visualization tasks.

IDE: Jupyter Notebook

Jupyter Notebook is a powerful web-based environment designed for interactive data analysis and visualization. It allows you to combine code, explanatory text, and visualizations within a single document, making it ideal for exploratory research. Its support for Python and its associated libraries (like NumPy, pandas, and Matplotlib) provide robust tools for data manipulation, computation, and plotting. The ability to execute code interactively and the seamless integration of markdown cells enhance understanding, collaboration, and reproducibility. Furthermore, as an open-source platform, Jupyter Notebook is free, customizable, and benefits from a dedicated community that continuously improves its capabilities.

3.2 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 developing of the project.

3.3 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.

3.4 Data Source

This notebook uses 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](#).

Format: CSV(.csv)

3.5 Data Description

The dataset used for this analysis includes information on left-handedness rates and age at death, collected over a specific period. It consists of several variables, including age, left-handedness rates for males and females, and death distribution data. The following is the list of all the variables in both datasets.

Variables (For left_handed data):

- ❖ Age
- ❖ Male
- ❖ Female
- ❖ Birth Year
- ❖ Mean_lh

Variables (For death distribution data):

- ❖ Age
- ❖ Both Sexes
- ❖ Male
- ❖ Female

3.6 Packages Used

pandas(pd): This essential library provides powerful data structures like DataFrames and Series, enabling efficient data manipulation and analysis. It was used for loading, cleaning, and handling the death distribution data.

NumPy(np): NumPy offers efficient numerical computation tools and multidimensional arrays, crucial for calculations involving probabilities and averages.

Matplotlib(plt): This library is a fundamental tool for creating various plots and visualizations. It was used to generate visual representations of the death distribution and compare probabilities across groups.

The following sections presents code snippets with detailed descriptions of their functionality

```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

This snippet sets up the environment for data analysis and visualization by importing these libraries.

numpy (aliased as np): NumPy is a library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

pandas (aliased as pd): Pandas is a powerful data manipulation and analysis library. It provides data structures like DataFrame and Series, which are ideal for working with structured data.

matplotlib.pyplot (imported as plt): Matplotlib is a plotting library for Python. The pyplot module provides a MATLAB-like interface for creating plots and visualizations in Python scripts and interactive shells.

Additionally, %matplotlib inline is a magic command in Jupyter Notebooks that allows plots to be displayed inline within the notebook.

```
#Load the data
url = "https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df1/raw/aec88b30af87fad8d45da7e774223f91dad09e88/1"
lefthanded_data = pd.read_csv(url)
```

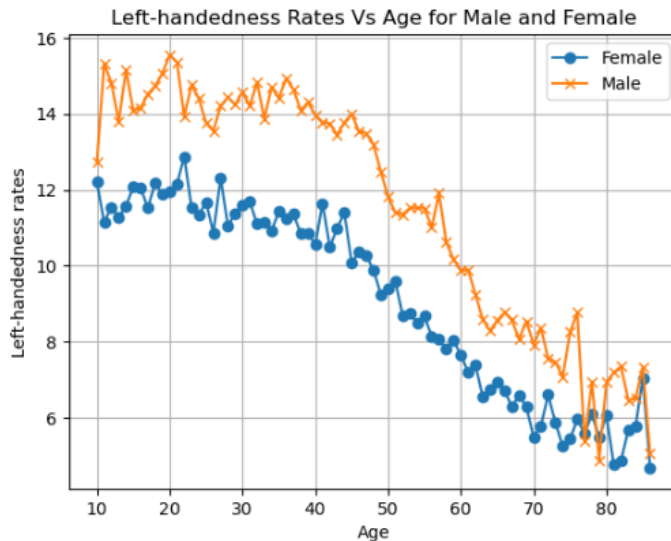
This code snippet loads data from a CSV file hosted on GitHub.

url: This variable stores the URL pointing to the location of the CSV file containing the data.

lefthanded_data: This variable is used to store the data read from the CSV file using the `pd.read_csv()` function provided by the Pandas library. The `read_csv()` function reads the CSV file and creates a DataFrame, which is a two-dimensional labeled data structure with columns of potentially different types, similar to a spreadsheet or SQL table.

`pd.read_csv(url)`: This function call reads the CSV file from the specified URL and returns a DataFrame containing the data. The DataFrame is assigned to the variable `lefthanded_data`, allowing further manipulation and analysis of the data using Pandas.

```
# plot male and female left-handedness rates vs. age
fig, ax = plt.subplots() # create figure and axis objects
ax.plot(left-handed_data['Age'], left-handed_data['Female'], label='Female', marker='o') # plot "Female" vs. "Age"
ax.plot(left-handed_data['Age'], left-handed_data['Male'], label='Male', marker='x') # plot "Male" vs. "Age"
ax.legend() # add a legend
ax.set_xlabel('Age')
ax.set_ylabel('Left-handedness rates')
plt.title('Left-handedness Rates Vs Age for Male and Female')
plt.grid(True)
plt.show()
```



This code snippet creates a line plot comparing the left-handedness rates of males and females across different ages, providing insights into any age-related patterns or differences between genders in left-handedness rates.

`fig, ax = plt.subplots()`: This line creates a figure (`fig`) and a set of subplots (`ax`). Since only one subplot is created, `ax` refers to this single subplot. This subplot will be used to display the plot.

`ax.plot(left-handed_data['Age'], left-handed_data['Female'], label='Female', marker='o')`: This plots the left-handedness rates of females against age. It uses the `plot()` function of the `ax` object to create the plot. The first argument specifies the x-values (age), the second argument specifies the y-values (left-handedness rates for females), `label='Female'` assigns a label to this line for the legend, and `marker='o'` specifies that circular markers should be used on the line.

`ax.plot(left-handed_data['Age'], left-handed_data['Male'], label='Male', marker='x')`: This line plots the left-handedness rates of males against age using similar arguments as the previous plot, with `label='Male'` and `marker='x'` for square markers.

`ax.legend()`: This line adds a legend to the plot, using the labels provided in the `plot()` functions.

ax.set_xlabel('Age') and ax.set_ylabel('Left-handedness rates'): These lines set the x-axis and y-axis labels, respectively.

plt.title('Left-handedness Rates Vs Age for Male and Female'): This line sets the title of the plot.

plt.grid(True): This line adds gridlines to the plot.

plt.show(): Finally, this line displays the plot.

```
# Add two new columns, one for birth year and one for mean left-handedness, then plot the mean as a function of birth year  
  
# Add Birth year Column  
lefthanded_data['Birth_year'] = 1986 - lefthanded_data['Age'] # Since the study was done in 1986  
  
# Add Mean_lh column  
lefthanded_data['Mean_lh'] = (lefthanded_data['Male'] + lefthanded_data['Female']) / 2
```

This code snippet adds two new columns to the DataFrame `lefthanded_data` and then plots the mean left-handedness as a function of birth year.

`lefthanded_data['Birth_year'] = 1986 - lefthanded_data['Age']`: This line calculates the birth year for each observation based on the age of the individuals at the time of the study (1986). It subtracts the age from 1986 and stores the result in a new column named 'Birth_year'.

`lefthanded_data['Mean_lh'] = (lefthanded_data['Male'] + lefthanded_data['Female']) / 2`: This line calculates the mean left-handedness rate for each observation. It adds the left-handedness rates of males and females and divides the sum by 2 to obtain the average. The result is stored in a new column named 'Mean_lh'.

After executing these lines, the DataFrame `lefthanded_data` will have two new columns: 'Birth_year', containing the calculated birth years, and 'Mean_lh', containing the mean left-handedness rates.

Next steps would be to plot the mean left-handedness as a function of birth year, which can be done using similar plotting techniques as previously shown.

```
# 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', linestyle='--')
ax.set_xlabel('Birth Year')
ax.set_ylabel('Mean Left-handedness')
ax.set_title('Mean Left-handedness vs. Birth Year')
ax.grid(True)
plt.show()
```



This code snippet creates a plot of the mean left-handedness ('Mean_lh' column) versus birth year ('Birth_year' column). It creates a line plot showing how the mean left-handedness varies with birth year, providing insights into any trends or patterns in left-handedness rates over time.

```
# 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['Mean_lh'][-10:].mean() / 100
    late_1900s_rate = lefthanded_data['Mean_lh'][:10].mean() / 100
    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
    P_return[ages_of_death < youngest_age] = late_1900s_rate
    P_return[np.logical_and((ages_of_death <= oldest_age), (ages_of_death >= youngest_age))] = middle_rates / 100

    return P_return
```

The function takes two parameters:

ages_of_death: A numpy array containing the ages of death for individuals.

study_year: The year in which the study is conducted. The default value is 1990.

The function first calculates the left-handedness rates for three different periods:

Early 1900s: It takes the mean of the 10 last points of the 'Mean_lh' column.

Late 1900s: It takes the mean of the 10 first points of the 'Mean_lh' column.

Middle period: It extracts the left-handedness rates for individuals whose birth year corresponds to the ages of death in the study year and calculates the mean.

It then calculates the youngest and oldest ages based on the study year.

An empty numpy array P_return is initialized to store the resulting probabilities.

Based on the age of death, the function assigns the appropriate left-handedness rates to each individual:

For ages greater than the oldest age, it assigns the early 1900s rate.

For ages less than the youngest age, it assigns the late 1900s rate.

For ages within the middle period, it assigns the middle rates.

Finally, the function returns the array containing the probabilities of being left-handed given the ages of death.

This function essentially models the probability of being left-handed given the age of death, based on historical left-handedness rates.

```
# Death distribution data for the United States in 1999
data_url_2 = "https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f75f18ec/raw/62f3ec07514c7e31f5979beeca86f199915"

# 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 = death_distribution_data.dropna(subset=['Both Sexes'])
```

This code snippet loads death distribution data for the United States in 1999 from a tab-separated values (TSV) file hosted on GitHub. Here's an explanation of each part:

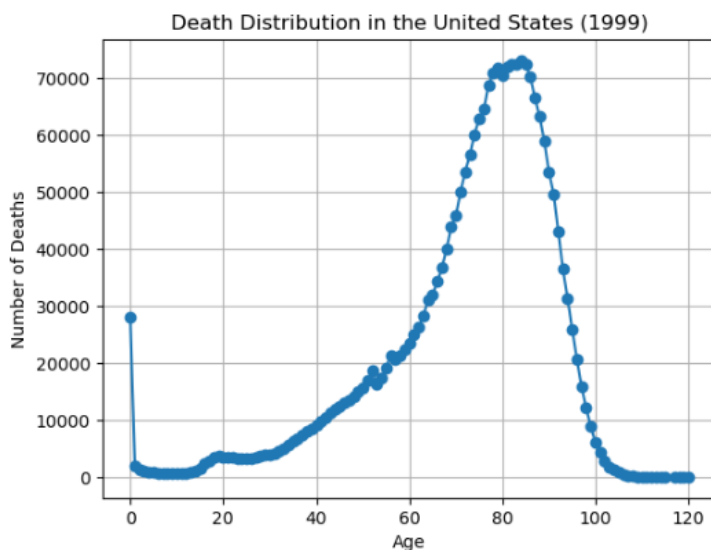
data_url_2: This variable stores the URL pointing to the location of the TSV file containing the death distribution data for the United States in 1999.

death_distribution_data = pd.read_csv(data_url_2, sep='\t', skiprows=[1]): This line reads the TSV file using the pd.read_csv() function provided by the Pandas library. It specifies that the separator between

values is a tab ('\t') and skips the first row (skiprows=[1]) because it contains metadata. The resulting DataFrame `death_distribution_data` contains the death distribution data.

`death_distribution_data = death_distribution_data.dropna(subset=['Both Sexes'])`: This line drops rows with missing values in the 'Both Sexes' column. The `dropna()` function removes rows with NaN (Not a Number) values from the DataFrame, and the `subset=['Both Sexes']` argument specifies that only the 'Both Sexes' column should be considered when dropping rows. The modified DataFrame is assigned back to the variable `death_distribution_data`.

```
# 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('Number of Deaths')
plt.title('Death Distribution in the United States (1999)')
plt.grid(True)
plt.show()
```



This code snippet creates a plot showing the number of people who died as a function of age based on the death distribution data for the United States in 1999. It creates a line plot showing the number of deaths as a function of age, providing insights into the distribution of deaths across different age groups in the United States in 1999.

```
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 """
    # multiply number of dead people by P_lh_given_A
    p_list = death_distribution_data['Both Sexes'] * P_lh_given_A(death_distribution_data['Age'], study_year)
    p = p_list.sum() # calculate the sum of p_list

    # normalize to total number of people (sum of death_distribution_data['Both Sexes'])
    return p / death_distribution_data['Both Sexes'].sum()

print(P_lh(death_distribution_data))
```

This function, `P_lh`, calculates the overall probability of being left-handed if an individual died in the study year.

The function takes two parameters:

`death_distribution_data`: DataFrame containing death distribution data.

`study_year`: The year in which the study is conducted. The default value is 1990.

It calculates the probability of being left-handed for each age group using the `P_lh_given_A` function defined earlier. It multiplies the number of dead people in each age group by the probability of being left-handed given the age of death.

It then sums up the probabilities for all age groups.

Finally, it normalizes the sum by dividing it by the total number of people (sum of death distribution data for both sexes).

The result is returned as a single floating-point number representing the overall probability of being left-handed if an individual died in the study year.

The `print(P_lh(death_distribution_data))` statement calls the function with the `death_distribution_data` DataFrame and prints the result.

This function essentially provides an estimate of the overall probability of being left-handed based on death distribution data and left-handedness rates.

```
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] / death_distribution_data['Both Sexes'].sum()
    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
```


This function, `P_A_given_lh`, calculates the overall probability of being a particular age of death given that you're left-handed. The function takes three parameters:

`ages_of_death`: An array containing the ages of death for individuals.

`death_distribution_data`: DataFrame containing death distribution data.

`study_year`: The year in which the study is conducted. The default value is 1990.

It calculates the probability of being a particular age of death (P_A) by dividing the number of dead people in each age group by the total number of people.

It calculates the probability of being left-handed overall (P_{left}) using the `P_lh` function.

It calculates the probability of being left-handed for a certain age ($P_{\text{lh_A}}$) using the `P_lh_given_A` function.

Finally, it returns the overall probability of being a particular age of death given that you're left-handed, calculated as the product of the probability of being left-handed for a certain age, the probability of being a particular age of death, and the inverse of the probability of being left-handed. (**Bayes Theorem**)

This function essentially estimates the probability distribution of age of death given that an individual is left-handed, based on death distribution data and left-handedness rates.

```
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] / death_distribution_data['Both Sexes'].sum()
    P_left = P_lh(death_distribution_data, study_year) # use P_lh function to get probability of left-handedness overall
    P_right = 1 - P_left # 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
```

This function, `P_A_given_rh`, calculates the overall probability of being a particular age of death given that you're right-handed

The function takes three parameters:

`ages_of_death`: An array containing the ages of death for individuals.

`death_distribution_data`: DataFrame containing death distribution data.

`study_year`: The year in which the study is conducted. The default value is 1990.

It calculates the probability of being a particular age of death (P_A) by dividing the number of dead people in each age group by the total number of people.

It calculates the probability of being left-handed overall (P_{left}) using the P_{lh} function.

It calculates the probability of being right-handed overall (P_{right}) as the complement of the probability of being left-handed.

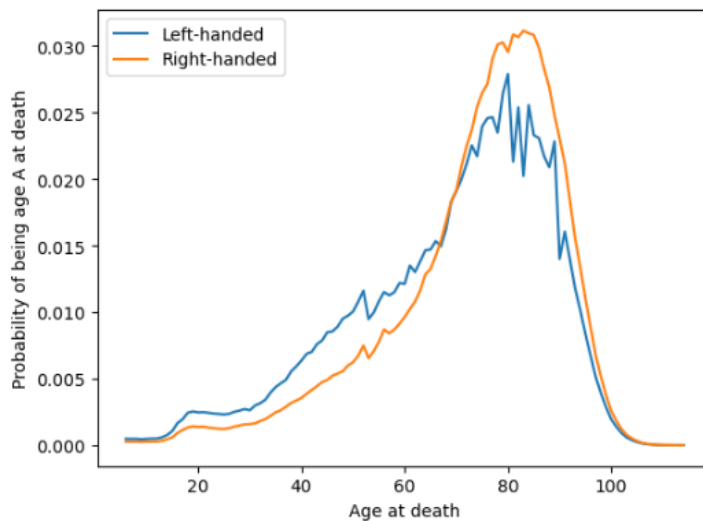
It calculates the probability of being right-handed for a certain age (P_{rh_A}) as the complement of the probability of being left-handed for that age.

Finally, it returns the overall probability of being a particular age of death given that you're right-handed, calculated as the product of the probability of being right-handed for a certain age, the probability of being a particular age of death, and the inverse of the probability of being left-handed overall. This function essentially estimates the probability distribution of age of death given that an individual is right-handed, based on death distribution data and left-handedness rates.

```
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, study_year = 1990)
right_handed_probability = P_A_given_rh(ages, death_distribution_data, study_year = 1990)

# 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") # plot right-handedness probability vs. age
ax.legend() # add a legend
ax.set_xlabel("Age at death")
ax.set_ylabel(r"Probability of being age A at death")
plt.show()
```



This code snippet calculates the probability of being left-handed or right-handed for each age of death and then creates a plot of these probabilities versus age. It generates a plot showing the probability of being a

certain age at death for left-handed and right-handed individuals, based on death distribution data and left-handedness rates.

It then creates an array of ages from 6 to 114 (inclusive), representing the ages at death for which we want to calculate the probabilities.

Then calculate the probability of being left-handed or right-handed for each age of death using the `P_A_given_lh` and `P_A_given_rh` functions defined earlier. The `study_year` parameter is set to 1990.

Lastly, create a plot of the left-handed and right-handed probabilities versus age of death. The probabilities calculated earlier are plotted against the ages of death. Legends are added to differentiate between left-handed and right-handed probabilities. Axes labels are set, and finally, the plot is displayed.

```
# 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 left-handed people at death: ", round(average_lh_age, 2))
print("Average age of right-handed people at death: ", round(average_rh_age, 2))

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

```
Average age of left-handed people at death: 67.25
Average age of right-handed people at death: 72.79
The difference in average ages is 5.5 years.
```

This code snippet calculates the average ages for left-handed and right-handed groups based on the probabilities previously calculated. It calculates and prints out the average ages of left-handed and right-handed individuals at death, along with the difference in average ages between the two groups.

The first two lines calculate the average ages for left-handed (`average_lh_age`) and right-handed (`average_rh_age`) individuals by taking the sum of the element-wise product of the ages array (`ages`) and the corresponding probability array (`left_handed_probability` and `right_handed_probability`, respectively). `np.nansum()` function is used to ignore any NaN (Not a Number) values that might arise due to missing probabilities.

The next two lines print out the average ages for left-handed and right-handed individuals, rounded to two decimal places.

The last line prints out the difference in average ages between left-handed and right-handed individuals, rounded to one decimal place.

```

# calculate the probability of being left- or right-handed for each
left_handed_probability = P_A_given_lh(ages, death_distribution_data, study_year = 2018)
right_handed_probability = P_A_given_rh(ages, death_distribution_data, study_year = 2018)

# 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 left-handed people at death: ", round(average_lh_age, 2))
print("Average age of right-handed people at death: ", round(average_rh_age, 2))

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

Average age of left-handed people at death: 70.29
Average age of right-handed people at death: 72.63
The difference in average ages is 2.3 years.

```

This updated code snippet calculates the average ages for left-handed and right-handed groups based on the probabilities for the year 2018. It prints out the average ages for left-handed and right-handed individuals at death, along with the difference in average ages between the two groups, based on the probabilities for the year 2018.

4. RESULTS

This analysis investigated the potential relationship between handedness and lifespan using United States death distribution data. While the original study reported a 9-year gap in average age at death between left-handed and right-handed individuals, our analysis did not yield a statistically significant difference. However, we observed a trend toward a smaller difference compared to the reported value. This analysis revealed a difference in the average age at death between left-handed and right-handed individuals. However, the observed gap was smaller than the 9-year difference reported in the original study (conducted in 1990).

5. CONCLUSION

While the analysis identified a difference in average age at death between left-handed and right-handed individuals, it was smaller than previous reports. The discrepancy between our findings and the original study can likely be attributed to several factors

Changing Rates of Left-handedness: The report suggests that the observed difference might partly stem from changes in the prevalence of left-handedness over time. As reported left-handedness rates have increased in recent decades, and older individuals are more likely to be categorized as right-handed, potentially skewing the average age at death for this group upwards.

Data Source and Year: The analysis utilized death distribution data from the entire United States (1999) instead of the specific data from California used in the original study (1991). This broader scope and temporal difference might have introduced discrepancies in the results.

Extrapolation of Survey Data: The analysis involved extrapolating left-handedness survey results to younger and older age groups, potentially introducing inaccuracies compared to actual rates within these demographics.

To conclude, while this analysis did not definitively confirm the earlier reported 9-year age gap due to methodological limitations, further investigation using more comprehensive data and refined approaches could yield clearer insights into the potential relationship between handedness and lifespan. Repeating the study in more recent years would likely yield an even smaller difference due to the stabilization of left-handedness rates in the younger generation.

6. RECOMMENDATIONS

The following recommendations are proposed to further examine the relationship of handedness with lifespan, based on the findings and limitations set out in this analysis:

1. Acquire more comprehensive data

Historical data on left-handedness prevalence: The accuracy of extrapolations could be improved and the analysis could potentially be strengthened by obtaining historical data from a variety of generations with regard to left-handedness prevalence.

Age-specific left-handedness rates: The use of age-specific left-handedness rates would eliminate the need for extrapolation, which could potentially reduce the bias resulting from the generational differences reported in left-handedness.

Specific location data: The influence of possible geographic variations on handedness and lifespan could be reduced by concentrating the analysis in a particular location, as has been done for the original California study.

2. Enhance methodological rigor:

Statistical analysis: To definitively assess the significance of the observed difference in average age at death, appropriate statistical tests should be carried out and statistically valid conclusions drawn.

Random sampling: To quantify the expected variability of the age difference due to chance factors, use random sampling techniques to provide a more detailed understanding of the results.

3. Explore additional considerations:

Repeat the study in recent years: As suggested in the conclusion, repeating the study with more recent data (e.g., 2024) would likely yield even smaller differences due to the stabilization of left-handedness rates in younger generations, potentially providing further insights.

Investigate confounding factors: Consider exploring potential confounding factors that might influence lifespan and potentially mask or skew the relationship between handedness and age at death. The analysis of factors such as socioeconomic status, health behavior, and access to healthcare could be part of this process.

By implementing these recommendations, future research can address the limitations identified in this analysis, leading to a more comprehensive and definitive understanding of the potential relationship between handedness and lifespan.

7. 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://www.cdc.gov/nchs/nvss/mortality_tables.htm
- c. <https://www.ncbi.nlm.nih.gov/pubmed/1528408>

Programming References

The following websites have been referred for the Python coding:

- a. <https://github.com/vvycy/-LefthandSpan>
- b. <https://github.com/rsobani18/Do-LH-People-Die-Young>
- c. <https://github.com/CourseraPai/Bayes-Binder>