Final Report of Traineeship Program 2023

On

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

MEDTOUREASY

26th August 2023

By Karamveer Singh

ACKNOWLEDGEMENTS

The traineeship opportunity that I had with MedTourEasy has been an exceptional catalyst for both personal and professional growth, allowing me to delve into the intricacies of Data Visualizations in Data Analytics while expanding my knowledge in the field. I am truly grateful for the guidance and support I received throughout this journey.

First and foremost, I would like to extend my deepest gratitude to the Training & Development Team of MedTourEasy for granting me this invaluable opportunity to undertake my traineeship at your esteemed organization. Your belief in my potential and willingness to provide me with hands-on experience have truly been instrumental in shaping my understanding of the Data Analytics profile. I am immensely grateful for your unwavering support and for investing your valuable time in mentoring me.

Additionally, I would like to express my sincere thanks to the entire team of MedTourEasy. The collaborative and nurturing environment you created made my experience immensely enriching. The opportunity to interact with professionals from diverse backgrounds and expertise has broadened my perspective and sharpened my skills. Your collective guidance and encouragement throughout the traineeship project have been invaluable, and I am truly grateful for the knowledge and insights you shared.

TABLE OF CONTENTS

ABSTRACT

A survey of 1,177,507 U.S. men and women between the ages of 10 and 86 included questions regarding hand preference for writing and throwing. Three effects were observed. Individuals with at least some left motoric bias comprised a smaller percent of the population with advancing age. This finding provides large-scale confirmation of a previously described phenomenon. Among sinistral, concordance for writing and throwing was 2.2 times as prevalent as left-writing with right-throwing, and 4.1 times as prevalent as right-writing with left-throwing. These sinistral subpopulations displayed distinct and stable prevalence prior to age 50 and changing patterns of prevalence subsequent to age 50. The results confirm a decrease with age in the prevalence of sinistrality, but indicate that age-specific rates of mixed- and left-handedness are distinct. The implications for hypotheses regarding age-related change in the prevalence of sinistrality are discussed.

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

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. This notebook uses 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.

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, but let's start by plotting the rates of left-handedness as a function of age.

This notebook uses two datasets: death distribution data for the United States from the year 1999 and rates of left-handedness digitized from a figure in this 1992 paper by Gilbert and Wysocki.

Language and Platform Used

Language: Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.Python is a programming language that lets you work quickly and integrate systems more effectively.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Platform: Google Colab

Google Colab is a web-based integrated development environment (IDE) for Python. It is different from regular script-based code editors like PyCharm, Eclipse, and Visual Studio Code. Colab is designed to enable machine learning with storage on the cloud.

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

Jupyter is the open-source project on which Colab is based. Colab allows you to use and share Jupyter notebooks with others without having to download, install, or run anything.

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

1. zero configuration required
2. Access to GPUs free of charge
3. Easy sharing

Whether you're a student, a data scientist or an AI researcher, Colab can make your work easier.

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.

This project uses two datasets: death distribution data for the United States from the year 1999 (source website: NVSS - Mortality Tables (cdc.gov)) 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 websites and data repositories). This data can then be manipulated, aggregated, filtered according to the requirements and needs of the project.

Libraries imported:

1. Pandas
2. NumPy
3. Matplotlib

1. Pandas: Pandas is a powerful library for data manipulation and analysis. It provides data structures like Data Frames and Series that allow you to efficiently work with structured data. Pandas offers functionalities for reading and writing data from various file formats (CSV, Excel, SQL databases), cleaning and preprocessing data, handling missing values, merging and joining datasets, and performing aggregations and statistical operations. It is widely used for data wrangling tasks and data exploration.

2. NumPy: NumPy, short for Numerical Python, is a fundamental library for scientific computing in Python. It provides a multidimensional array object (ndarray) that enables efficient storage and manipulation of large arrays of homogeneous data. NumPy offers a wide range of mathematical functions, linear algebra operations, random number generation, and tools for working with arrays. It is the foundation for many other libraries in the scientific Python ecosystem and is essential for performing numerical computations and data analysis.

3. Matplotlib: Matplotlib is a plotting library that enables you to create a variety of static, animated, and interactive visualizations in Python. It provides a MATLAB-like interface and allows you to generate high-quality plots, histograms, scatter plots, bar charts, line plots, and more. Matplotlib offers extensive customization options for adjusting colours, styles, labels, titles, and annotations. It can be used for exploratory data analysis, data visualization in research papers, creating dashboards, and presenting insights to stakeholders.

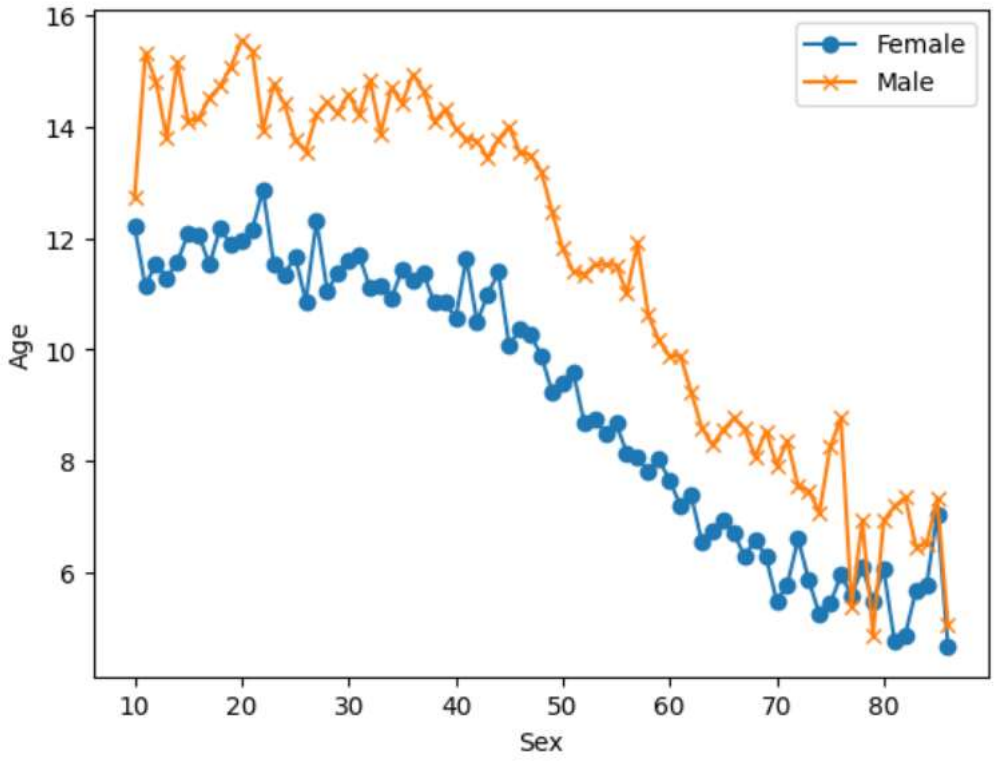
Together, these three libraries form a powerful toolkit for data analysis, manipulation, and visualization in Python.

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

```
# load the data
data_url_1 = "https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df1/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('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')
```

```
Text(0, 0.5, 'Age')
```



3.3 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.

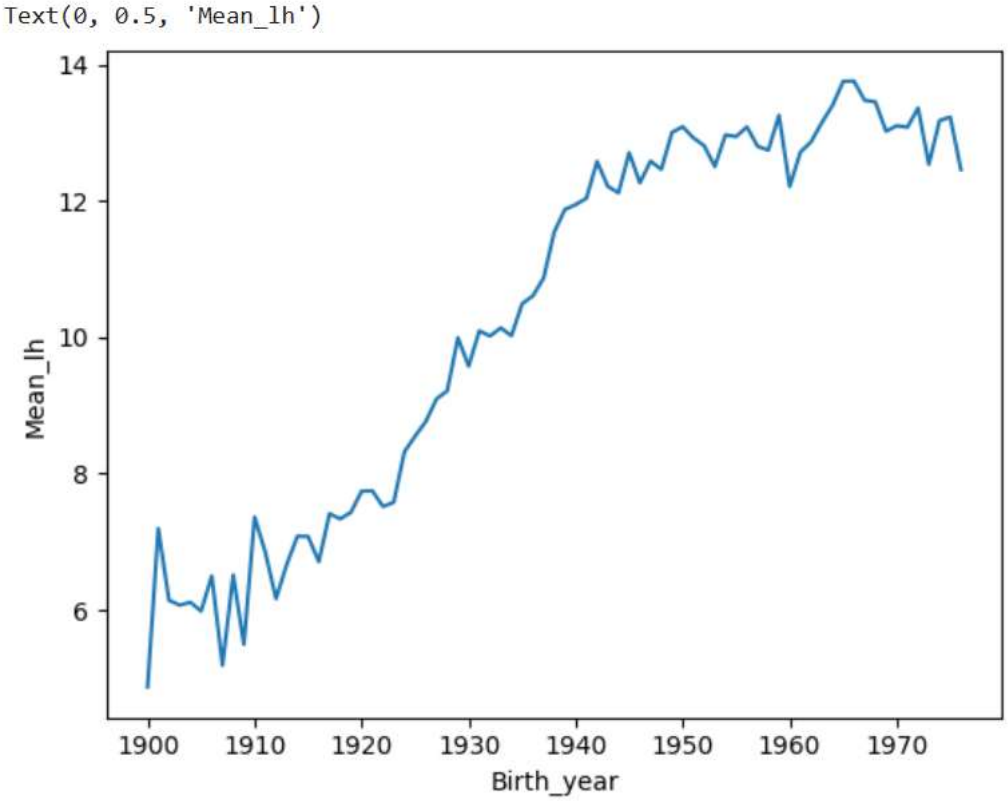
```
# create a new column for birth year of each age

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('Birth_year', 'Mean_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('Mean_lh') # set the y label for the plot
```



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.

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

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.

```
# 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['Mean_lh'][-10:].mean()
    late_1900s_rate = lefthanded_data['Mean_lh'][:10].mean()
    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 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.

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.

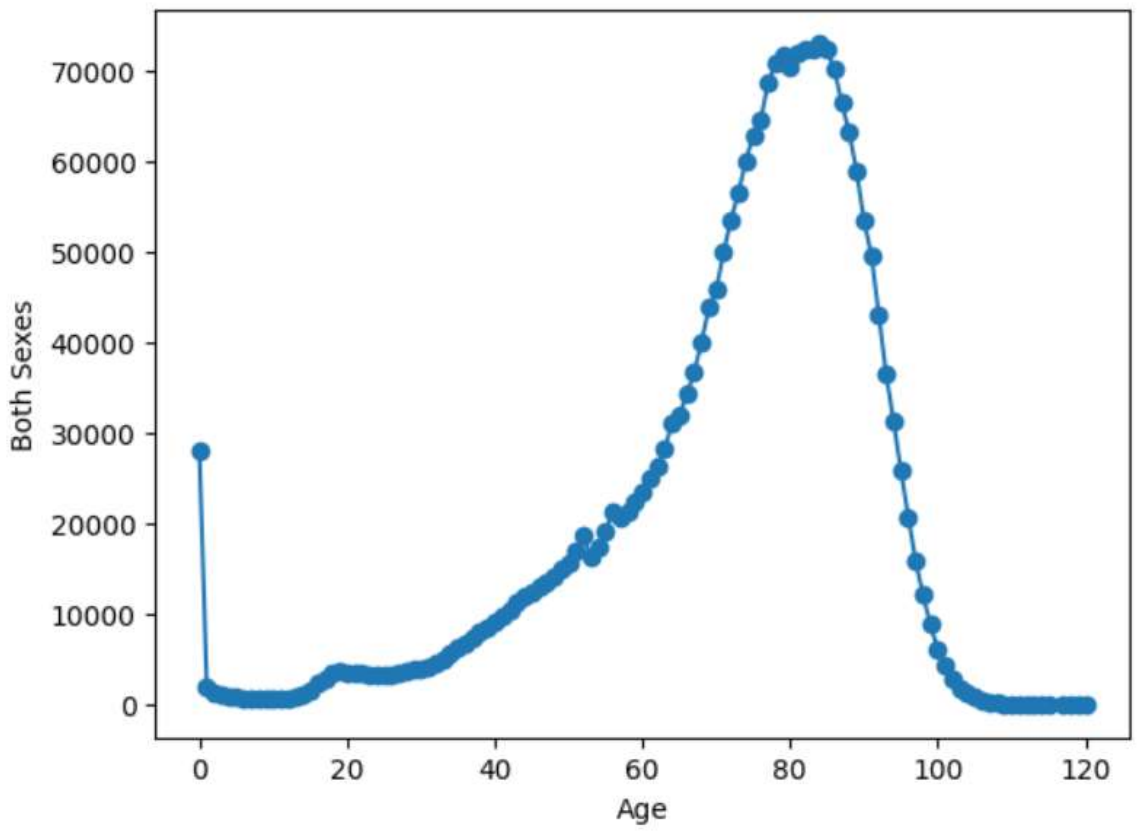
```
# Death distribution data for the United States in 1999
data_url_2 = "https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f75f18ec/raw/
62f3ec07514c7e31f5979beeca86f19991540796/cdc_vs00199_table310.tsv"

# 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 = 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')
```

```
Text(0, 0.5, 'Both Sexes')
```



3.6 The 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 data frame- death_distribution_data):

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

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

print(P_lh(death_distribution_data))
```

Result-

```
0.07766387615350638
```

3.7 Putting it all together: dying while left-handed(i)

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.

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

First, for left handers:

```
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
```

3.8 Putting it all together: dying while left-handed(ii)

And now for right handers:

```
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
```

3.9 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.

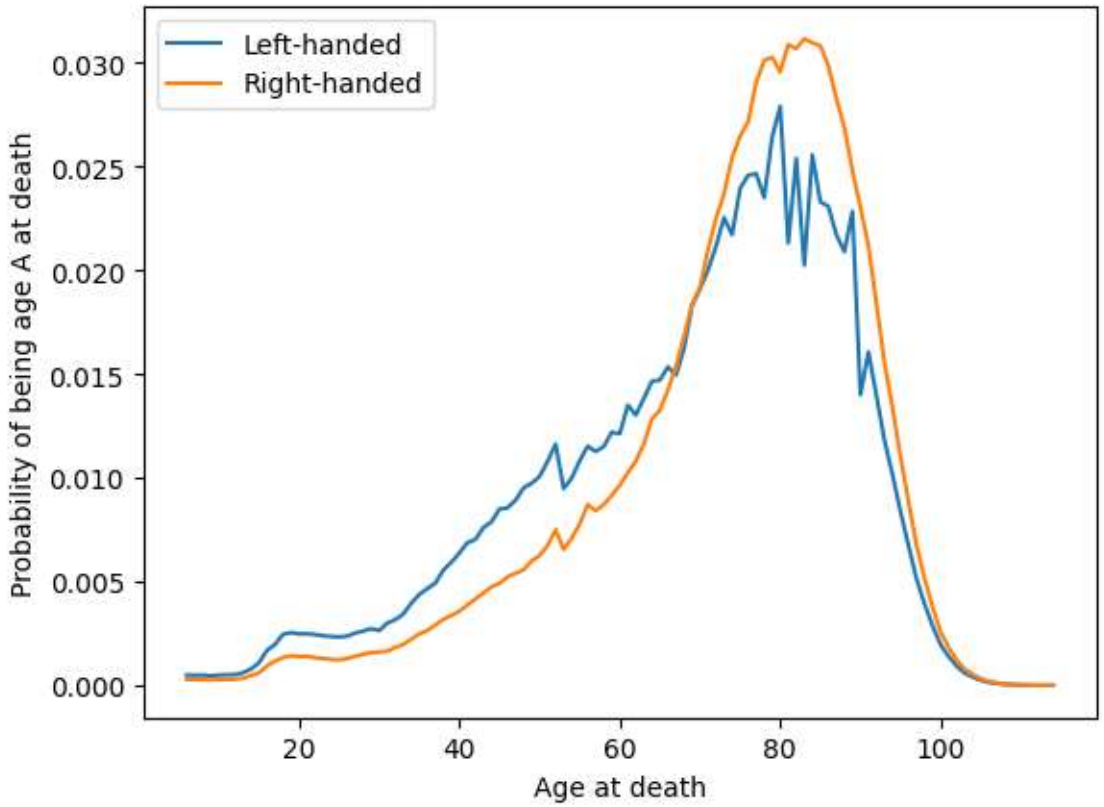
Notice that the left-handed distribution has a bump below age 70: of the pool of deceased people, left-handed people are more likely to be younger.

```
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")
```

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



3.10 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.

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

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

```
# 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
# ... YOUR CODE FOR TASK 9 ...
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.")
```

Result-

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

Conclusion

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:

1. 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).
2. 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.

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

Result-

The difference in average ages is 2.3 years.

REFERENCES

Data Collection

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

1. Deaths By Single Years of Age, Race, and Sex: United States, 1999 (cdc.gov)
2. NVSS - Mortality Tables (cdc.gov)
3. Hand preference and age in the United States - PubMed (nih.gov)

Training

The following training was completed prior to the completion of this project:

1. Google Data Analytics Professional Certificate | Coursera