Final Report of Traineeship Program 2023

On

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

MEDTOUREASY

26th September 2023

Prepared By-

Mukesh Chandra Kamila

ACKNOWLEDGEMENT

This internship opportunity that I had with MedTourEasy was a great opportunity and shift in my career for learning and understanding the significance of Data Analytics. It helped me in personal as well as professional development.

I have done this project under the guidance of MedTourEasy. I take this opportunity to express our gratitude. First and foremost, I would like to express our sincere gratitude to my mentor Ankit Hasija for his continued support and guidance which led to the completion of the project work. All the interactive conversations I have had with him during this period have been inspiring and rewarding for me. It was a great pleasure to work with him as he was exceptionally cooperative, helpful, modest and caring. This project would not have been completed without his guidance. And I am deeply grateful for MedTourEasy's support and the opportunity they have provided me. Their assistance allowed me to focus on my internship and my professional development, and I am thankful for their commitment to helping me succeed, their support and love during this journey will live in my memory.

TABLE OF CONTENTS

1. 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. **MedTourEasy's** mission is to provide access to quality healthcare for everyone, regardless of location, time frame or budget.

1.2 Project Description

In this 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 lefthanded or right-handed.

1.3 Objectives and Deliverables

This project is all about focusing and carrying out the in-depth analysis by gathering data of right-handers and left-handers from various sources like 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. Then, using Python and it's packages like Pandas, NumPy and Matplotlib.Pyplot to analyze and visualize this project which will provide actionable insights that can help company make informed decisions.

The project consists of deliverables as follows:

- Rates of left-handedness over time.
- When do people normally die?
- The overall probability of left-handedness
- Age of left and right-handers at death
- Differences of average age of left-handed people and right-handed people at death.

2. METHODOLOGY

2.1 Approach to the Project

First, I have gathered data of right-handers and left-handers from various sources like 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.

Then I have loaded the datasets in Python. Then I have done the EDA part using Python with the help of Pandas a data analysis library. And this analysis was done with the help of various libraries, functions and formula.

The project followed the following steps to accomplish the desired objectives and deliverables:

- Gathering Requirements & Defining Problem
- Data collection and importing
- Cleaning the datasets
- Analyze the datasets
- Exploratory Data Analysis (EDA)

2.1 Language and Platform Used

Language: Python

There are many programming languages available, but Python is popularly used by statisticians, engineers, scientists and analyst to perform data analytics.

Here are some of the reasons why Data Analytics using Python has become popular:

1. Python is easy to learn and understand and has a simple syntax.
2. The programming language is scalable and flexible.
3. It has a vast collection of libraries for numerical computation and data manipulation.
4. Python provides libraries for graphics and data visualization to build plots.
5. It has broad community support to help solve many kinds of queries.

IDE: Google Colab

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. Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. To be precise, Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup.

Libraries: Pandas, NumPy, Matplotlib

One of the main reasons why Data Analytics using Python has become the most preferred and popular mode of data analysis is that it provides a range of libraries.

NumPy: NumPy supports n-dimensional arrays and provides numerical computing tools. It is useful for Linear algebra and Fourier transform.

Pandas: Pandas provides functions to handle missing data, perform mathematical operations, and manipulate the data.

Matplotlib: Matplotlib library is commonly used for plotting data points and creating interactive visualizations of the data.

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

```
# import library
import numpy as np
```

3. IMPLEMENTATION

3.1 Gathering Requirements and Defining Problem Statement

So, this is the first step where all the instruction and requirements are received from MedTourEasy to understand what needs to be done in this project and all the questions are being asked by MedTourEasy need to be answered to reach the deliverables during this project, after this the final step is the problem statement which is defined which has to be followed while development of the project.

3.2 Data Collection and Importing

The data of Right-Handers and Left-Handers has been collected through various sources, 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.
- https://www.cdc.gov/nchs/nvss/mortality_tables.htm

Data importing is something that let us upload the required data into the programing language from external sources (online websites and data repositories). Then the data can be manipulated, aggregated, filtered as per the analysis requirements and needs of the project.

Read_csv: CSV files are the Comma Separated Files. To access data from the CSV file, we require a function read_csv() from Pandas that retrieves data in the form of the data frame.

```
# load the data
data_url_1 =
"https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574
e54df1/raw/aec88b30af87fad8d45da7e774223f91dad09e88/lh_data.csv"
lefthanded_data = pd.read_csv(data_url_1)
```

```
# Death distribution data for the United States in 1999
data_url_2 =
"https://gist.githubusercontent.com/mbonsma/2f4076aab6820ca1807f4e29f7
5f18ec/raw/62f3ec07514c7e31f5979beeca86f19991540796/cdc_vs00199_table3
10.tsv"

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

3.3 Designing Databases

Once the data is collected and imported into the Python environment, it is necessary to design the structure of the database tables as to clearly recognize the columns, rows and datatype in the data.

Once the data is imported into Python environment, it is converted into pandas' data frame which makes it easy to maintain the data in form of tables. The following are the various tables which have been created as pandas' data frame:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77 entries, 0 to 76
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Age     77 non-null     int64
 1   Male    77 non-null     float64
 2   Female  77 non-null     float64
dtypes: float64(2), int64(1)
memory usage: 1.9 KB
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 120 entries, 0 to 120
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          120 non-null    int64
 1   Both Sexes   120 non-null    float64
 2   Male         115 non-null    float64
 3   Female       120 non-null    float64
dtypes: float64(3), int64(1)
memory usage: 4.7 KB
```

4. ANALYSIS AND OBSERVATIONS

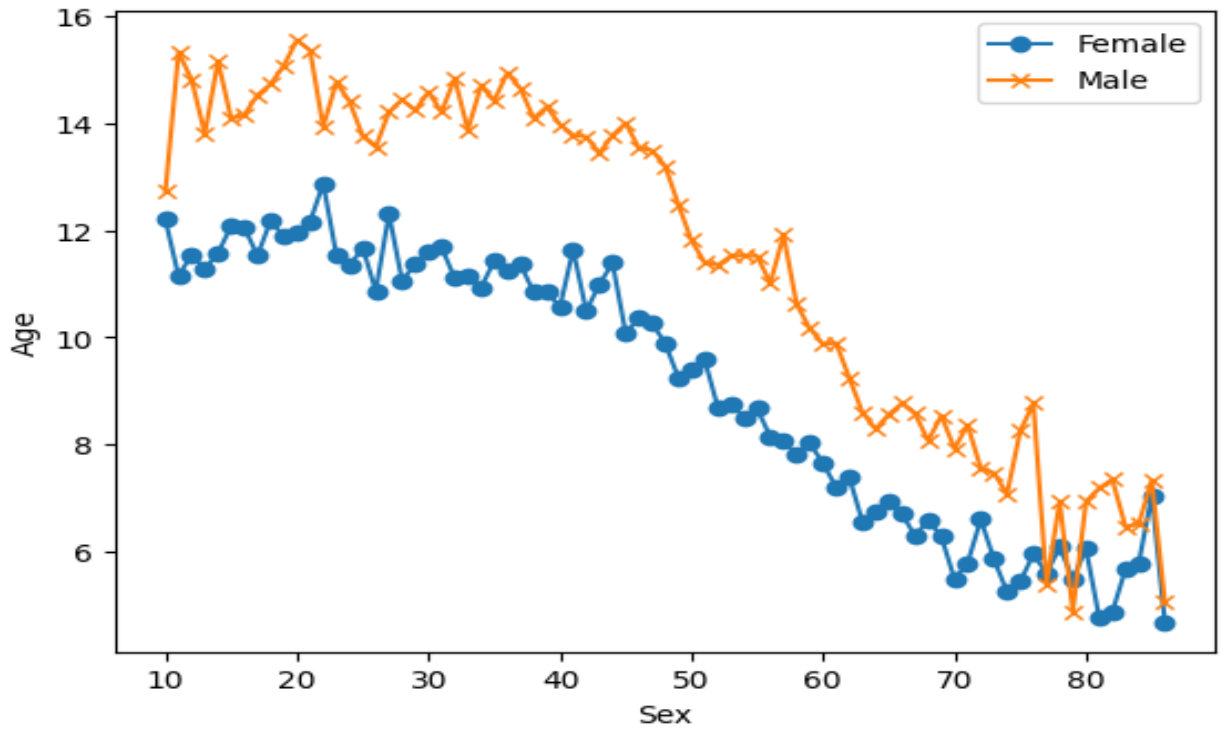
4.1 Where are the old left-handed people?

Load the handedness data from the National Geographic survey and create a scatter plot.

- Import pandas as pd and matplotlib.pyplot as plt.
- Load the data into a pandas DataFrame named lefthanded_data using the provided data_url_1. Note that the file is a CSV file.
- Use the .plot() method to create a plot of the "Male" and "Female" columns vs. "Age".

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

# load the data
data_url_1 =
"https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de
395574e54df1/raw/aec88b30af87fad8d45da7e774223f91dad09e88/lh_dat
a.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')
```

Observations: In the below scatter plot, it shows the effect of changing rate has on the apparent mean age of death of left-handed people, the rates of left-handedness as a function of age of both male and female is going down as mean age of death of left-handed is down.



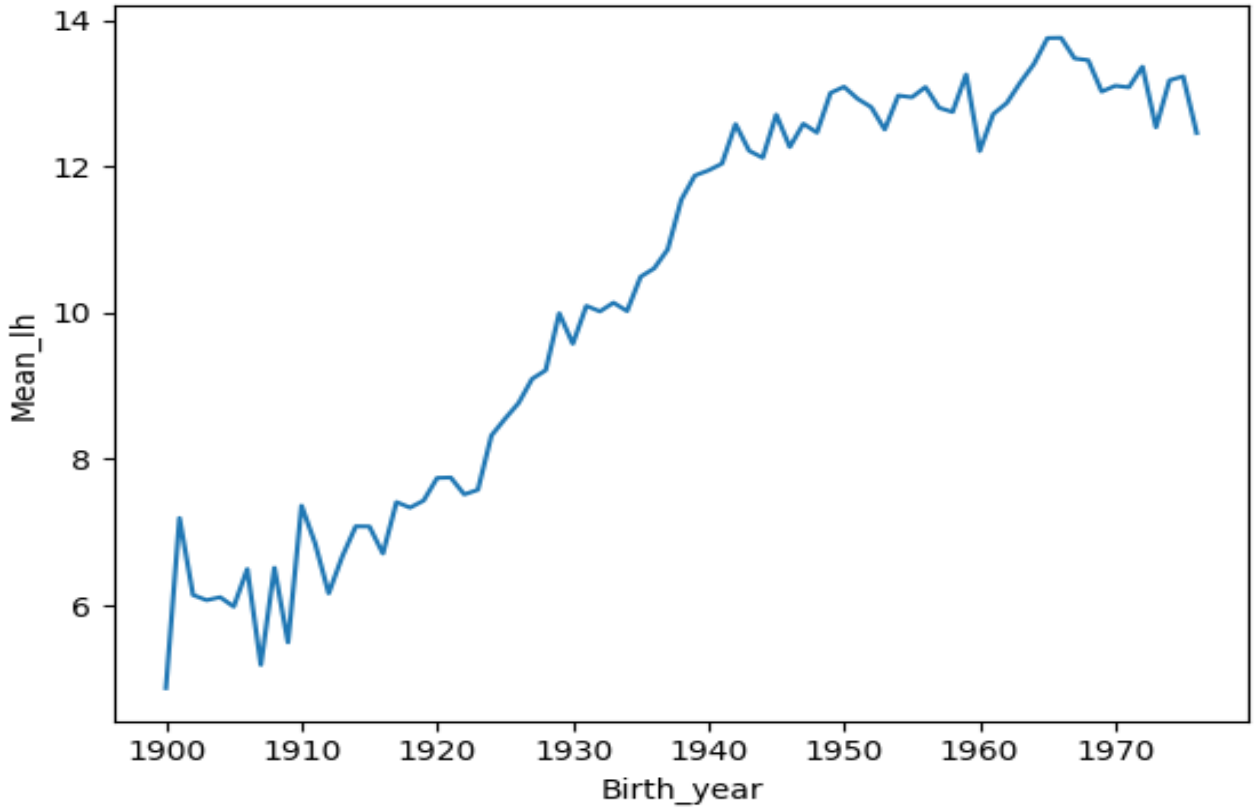
4.2 Rates of left-handedness over time

Add two new columns, one for birth year and one for mean left-handedness, then plot the mean as a function of birth year.

- Create a column in lefthanded_data called Birth_year, which is equal to 1986 - Age (since the study was done in 1986).
- Create a column in lefthanded_data called Mean_lh which is equal to the mean of the Male and Female columns.
- Use the .plot() method to plot Mean_lh vs. Birth_year.

```
# 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
```

Observations: This data converted 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. This chart indicates that there has been a significant increase in the rates of left-handedness when they were born between 1910-1940 birth year. But from 1941 to 1986 birth year, the rates of left-handedness have been steady, there has been no change in rates.



4.3 Applying Bayes' rule

Create a function that will return P(LH | A) for particular ages of death in a given study year.

- Import the numpy package aliased as np.
- Use the last ten Mean_lh data points to get an average rate for the early 1900s. Name the resulting DataFrame early_1900s_rate.
- Use the first ten Mean_lh data points to get an average rate for the late 1900s. Name the resulting DataFrame late_1900s_rate.
- For the early 1900s ages, fill in P_return with the appropriate left-handedness rates for ages_of_death. That is, input early_1900s_rate as a fraction, i.e., divide by 100.
- For the late 1900s ages, fill in P_return with the appropriate left-handedness rates for ages_of_death. That is, input late_1900s_rate as a fraction, i.e., divide by 100.

When we are calculating early_1900s_rate and late_1900s_rate, we remember that because the original data was from youngest age to oldest age, that means that the data is organized from latest birth year to earliest birth year. You will use the first ten Mean_lh data points to get an average rate for the late 1900s and the last ten for the early 1900s.

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)

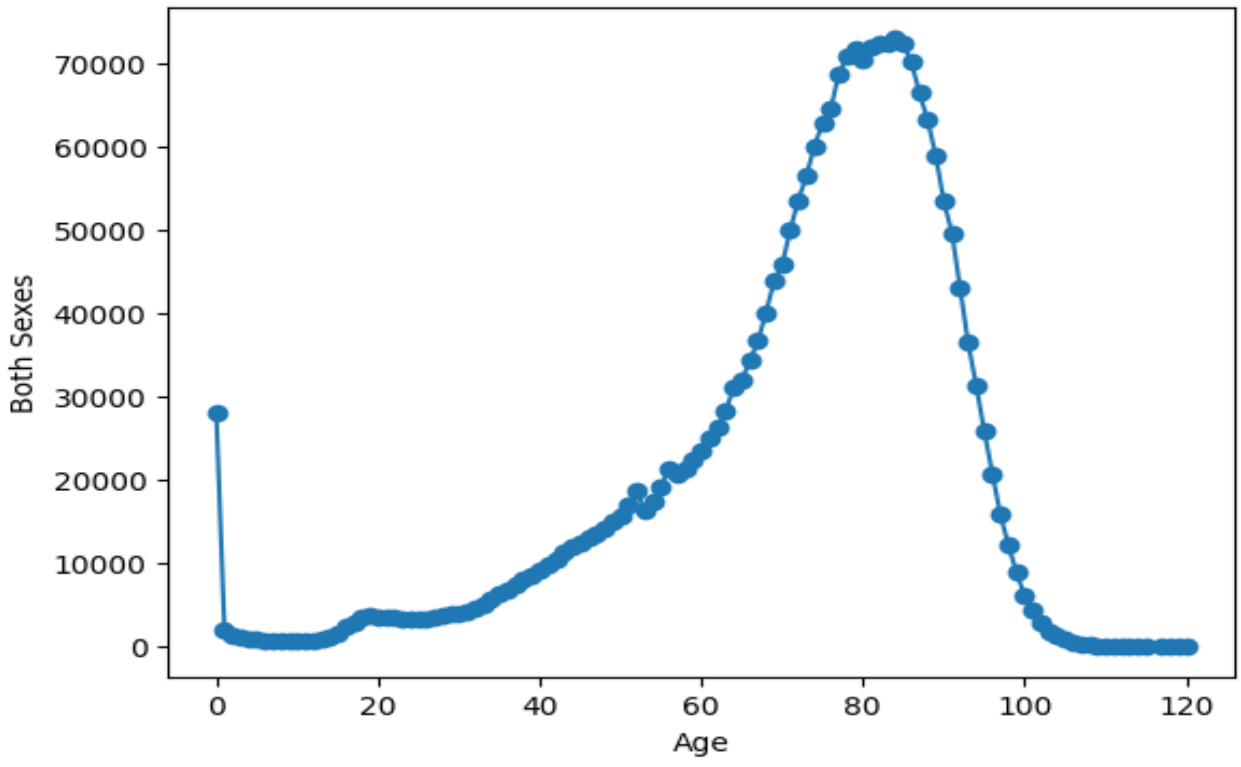
```
# import library
import numpy as np


# 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
```

4.4 When do people normally die?

Load death distribution data for the United States and plot it.

- Load death distribution data in the provided data_url_2 into death_distribution_data, setting sep = '\t' and skiprows=[1] to account for the dataset's format.
- Drop the NaN values from the Both Sexes column.
- Use the .plot() method to plot the number of people who died as a function of their age.

Observations: This chat indicates that there has been a linear increase in the total number of deaths from age of 20 to 80. Even at age of 80, maximum number deaths there, which is around 70000. But after at age of 80, the number of deaths has been decreased.



4.5 The overall probability of left-handedness

Create a function called P_lh() which calculates the overall probability of left-handedness in the population for a given study year.

- Create a series, p_list, by multiplying the number of dead people in the Both Sexes column with the probability of their being lefthanded using P_lh_given_A().
- Set the variable p equal to the sum of that series.
- Divide p by the total number of dead people by summing death_distribution_data over the Both Sexes column. Return result from the function.
- P(LH | A) was defined in Task 3. N(A) is the value of Both Sexes in the death_distribution_data DataFrame where the Age column is equal to A. The denominator is total number of dead people, which you can get by summing over the entire data frame in the Both Sexes column.

In equation form, this is what we're calculating, where N(A) is the number of people who died at age A:

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

Observations: The overall probability of left-handedness is 0.07766387615350638.

```
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))
```

```
0.07766387615350638
```

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

Write a function to calculate P_A_given_lh().

- Calculate P_A, the overall probability of dying at age A, which is given by death_distribution_data at age A divided by the total number of dead people (the sum of the Both Sexes column of death_distribution_data).
- Calculate the overall probability of left-handedness P(LH) using the function defined in Task 4.5.
- Calculate P(LH | A) using the function defined in Task 4.3.

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.

First, for lefthanders: P(A|LH) = P(LH|A) * P(A) / P(LH)

```
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
```

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

Write a function to calculate P_A_given_rh().

- Calculate P_A, the overall probability of dying at age A, which is given by death_distribution_data at age A divided by the total number of dead people. (This value is the same as in task 4.6)
- Calculate the overall probability of right-handedness P(RH), which is 1-P(LH).
- Calculate P(RH | A), which is 1 - P(LH | A).

And now for right-handers:

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

```
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
```

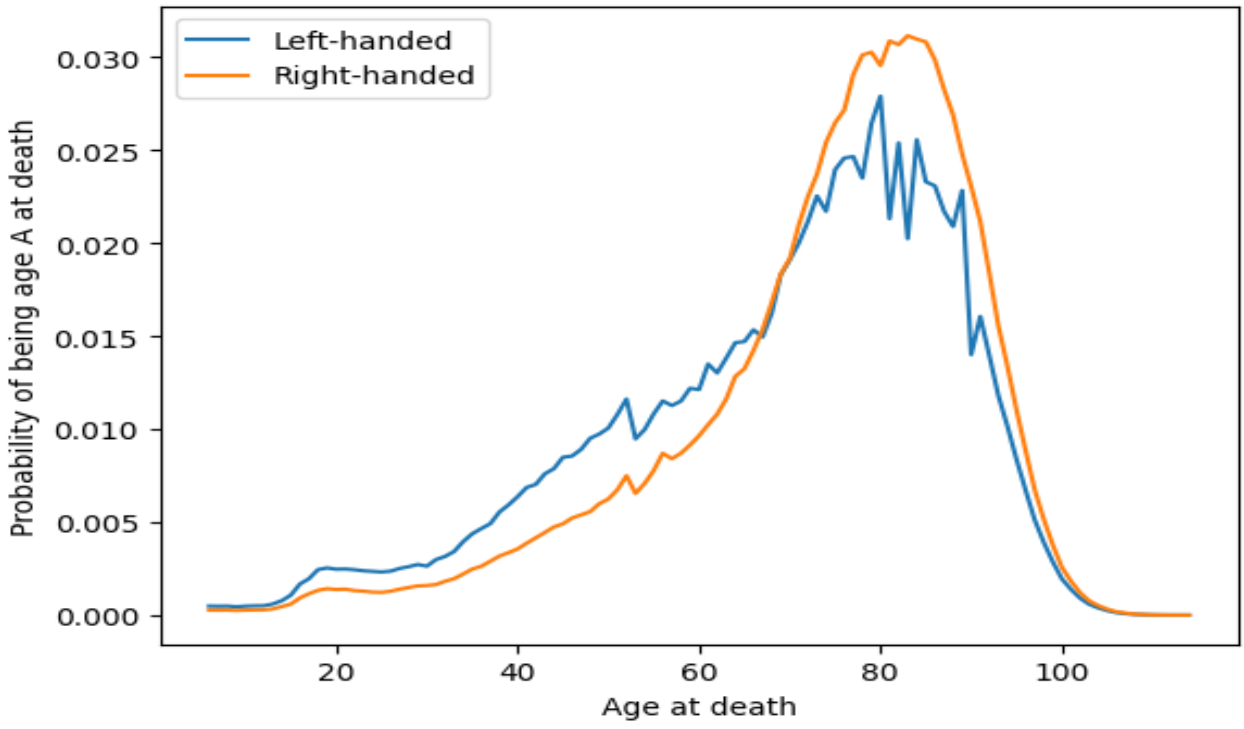
4.8 Plotting the distributions of conditional probabilities

Plot the probability of being a certain age at death given that you're left- or right-handed for a range of ages.

- Calculate P_A_given_lh and P_A_given_rh using the functions defined in Task 4.6
- Use the .plot() method to plot the results versus age.

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.

Observations: So, the probabilities of ages of death for both left-handed and right-handed are spiked at age between 70 to 90, but if we compare the probabilities of ages of death of right-handed are higher than the left-handed. the probabilities of ages of death of left-handed are bit high than right-handed at age between 20 to 60.



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

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

- Multiply the ages list by the left-handed probabilities of being those ages at death, then use np.nansum to calculate the sum. Assign the result to average_lh_age.
- Do the same with the right-handed probabilities to calculate average_rh_age.
- Print average_lh_age and average_rh_age.
- Calculate the difference between the two average ages and print it.

To make your printed output prettier, tried using the round() function to round the results to two decimal places.

Average age of left-handed people at death=∑AAP(A|LH)

Average age of right-handed people at death=∑AAP(A|RH)

Observations: From the below calculation, we found that the average age of Lefthanders is 67.25 and the average age of Righthanders is 72.79 and the difference in average ages is 5.5 years.

```
# 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("The average age of Lefthanders " + str(round(average_lh_age,2)))
print("The average age of Righthanders " + str(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.")
```

```
The average age of Lefthanders 67.25
The average age of Righthanders 72.79
The difference in average ages is 5.5 years.
```

4.10 Final comments

Redo the calculation from Task 8, setting the study_year parameter to 2018.

- In the call to P_A_given_lh, set age_of_death to ages, death_distribution_data to death_distribution_dat a, and study_year to 2018.
- Do the same for P_A_given_rh.

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.

Observations: The difference in average ages is 2.3 years if we did the study in 2018 instead of in 1990.

```
# 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.")
```

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

5. CONCLUSION

If we compare our results with the original study, we found that left-handed people were nine years younger at death on average. But as per the analysis of this project, in the above calculation, we found that the average age of Lefthanders is 67.25 and the average age of Righthanders is 72.79 and the difference in average ages is 5.5 years. And the difference in average ages is 2.3 years if we did the study in 2018 instead of in 1990.

This project aimed at analyzing the analyze death age difference of right handers with left handers. Currently, the project is done with analysis and all the questions and tasks has been answered.

6. REFERENCES

1. https://www.cdc.gov/nchs/data/statab/vs00199_table310.pdf
2. https://www.cdc.gov/nchs/nvss/mortality_tables.htm
3. https://www.ncbi.nlm.nih.gov/pubmed/1528408

Colab Notebook Link- https://colab.research.google.com/drive/1GmncInX-UBSaZBXasPIGr6pYSvsXc_-Y?usp=sharing