Final Report of Traineeship Program 2023

On

"ANALYZE DEATH AGE DIFFERENCE OF RIGHT HANDERS WITH LEFT"

By

NAVEEN A

27th MARCH 2023

ACKNOWLDEGMENTS

The traineeship opportunity that I had with MedTourEasy was a great change for learning and understanding the intricacies of the subject of Data Visualizations in Data Analytics; and, for personal as well as professional development. I am very obliged for having a chance to interact with so many professionals who guided me throughout the traineeship project and made it a great learning curve for me.

Firstly, I express my deepest gratitude and special thanks to the Training & Development Team of MedTourEasy who gave me an opportunity to carry out my traineeship at their esteemed organization. Also, I express my thanks to the team for making me understand the details of the Data Analytics profile and training me in the same so that I can carry out the project properly and with maximum client satisfaction and for spearing his valuable time despite his busy schedule.

I would also like to thank the team of MedTourEasy and my colleagues who made the working environment productive and very conducive.

TABLE OF CONTENTS

ABSTRACT

Barack Obama is left-handed. So are Bill Gates and Oprah Winfrey; so were Babe Ruth and Marie Curie. A 1991 study reported that left-handed people die on average nine years earlier than right-handed people. Nine years! Could this really be true?

In this notebook, 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 analyse 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 analysed 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 (source website here) and rates of left-handedness digitized from a figure in this 1992 paper by Gilbert and Wysocki.

I. INTRODUCTION

1.1 About the Company

MedTourEasy, a global healthcare company, provides you with the informationalresources needed to evaluate your global options. MedTourEasy provides analytical solutions to our partner healthcare providers globally.

1.2 About the Project

In this notebook, 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

In that reference, it is extremely crucial to create visualizations which help firms to analyze this situation and to prepare themselves for the future. Additionally, MedTourEasy, being one of the globally upcoming tele-medicine company in global healthcare, it is important for the firm to understand the situation of death rate between them so as to gain more insights on the intensity of the same, the response of all categories and the impact it will have on their market. Also, depending on the results of the analysis, this may be used for increasing their market presence and capacity planning.

Hence, 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.

,

- *Analysis of the problem*: 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.
- 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.
- Each of the above sub-sections has been represented in the form of dashboards which are created using R language on RStudio IDE and R Markdown package. These dashboards use a wide array of functions and packages in R to create intuitive and drillable dashboards, which can then be used by the firm to analyze the situation and draw conclusions about the same.

1.3 Objectives and Deliverables

This project focuses on creating easily understandable, interactive and dynamic dashboards by gathering data from various sources like 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. etc. and using the coding language R and packages like readr, dplyr, ggplot, ggplot2, flex dashboard and other R Shiny Packages to visualize these statistics which will enable the firm to analyze the situation and draw conclusions regarding the pandemic. The prototype for all the dashboards will be created using Power BI (primarily to create dynamic visualizations like world map, heat maps, forecasting, slicers etc.)

The project consists of 2 dashboards detailed as follows:

a. Analysis of the problem: This dashboard focuses on analyzing the data regarding the problem. It highlights the following points and displays them through various types of visualizations:
    - Comparison of between male and female in cases.
    - Country wise comparison of variation of cases with respect to age group, gender, health, ethnicity.
    - Analysis of most vulnerable areas Worldwide.

b. Analysis: This dashboard focuses on preparing graphs for following:
    - Where are all the old left-handed people?
    - Rates of left handedness over time.
    - Applying Bayes' rule (theorem).
    - When do people normally die.
    - The overall probability of left handers.
    - Moment of truth: age of left and right-handers at death.

II. METHODOLOGY

2.1 Flow of the Project

The project followed the following steps to accomplish the desired objectives and deliverables. Each step has been explained in detail in the following section.

2.2 Language and Platform Used

2.2.1 Language: R

It is a programming language and software environment for statistical analysis, representation of graphics, and reports. R was developed in the University of Auckland, New Zealand by Ross Ihaka and Robert Gentleman, and is currently being developed by the R Technology Core Team. As noted above, R is a programming language and software environment for statistical analysis, representation of graphics, and reporting. The important features of R are:

- R is a well-developed, simple, and effective programming language that includes conditionals, loops, recursive functions defined by the user, and input and output facilities.
- R has efficient data processing and storage facilities.
- R includes a set of operators for arrays, lists, vectors, and matrix calculations.
- R offers a detailed, coherent, and organized data analysis tool set.
- R provides graphical data analysis facilities and displays either directly on the computer or printing on papers.

Installation:

```
# import libraries
# ... YOUR CODE FOR TASK 1 ...
import pandas as pd
import matplotlib.pyplot as plt
```

III. 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 must 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 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, mentioned as follows:

- death distribution data
- 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, and filtered according to the requirements and needs of the project.

Packages Used:

Readr: The goal of readr is to provide a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. To accurately read a rectangular dataset with readr, one needs to combine two pieces: a function that parses the overall file, and a column specification.

.csv (comma separated values): a .csv file is a plain text file that contains a list of data. They mostly use commas to separate (or delimit) data, but sometimes they use other characters, like semicolons.

.tsv (tab separated values): a .tsv file stores a data table in which the columns of data are separated by tabs. For example, a database table or spreadsheet data.

.fwf (fixed width files): a .fwf file has a specific format that allows for the saving of textual data in an organized fashion.

```
Sample Code:

library(readxl)
library(readr)

x <- data.frame(read.csv(url(ECDC_ConfirmPath),stringsAsFactors = F))

vb <- data.frame(read_csv("D:/Traineeship/MedTourEasy - 6th April
2020/Project – COVID/Input Data/JHU/05-28-2020.csv"))

world1 <- data.frame(readxl::read_excel("D:/Traineeship/MedTourEasy - 6th
April 2020/Project - COVID/Input Data/ECDC/daily-cases-covid-
region.xlsx"))COVID/Input Data/JHU/05-28-2020.csv"))
```

3.3 Designing Databases

Once the data has been collected and imported into the R environment, it is important to design the structure of the database tables so as to identify the constraints in the data, keys, dependencies and relations between various tables.

Once the data is imported in the environment, it is converted into a data frame (data type in R) which makes it easy to maintain the data in form of tables. The various tables which have been created are mentioned as follows:

Attribute	Data type	Size	Extra
Sepal_length	INT	5	Primary Key
Sepal_width	INT	5	Not Null, Unique
Petal_length	INT	5	Not Null
Petal_width	INT	5	Not Null
Species	VARCHAR	25	Not Null

3.4 Data Cleaning

"Quality data beats fancy algorithms"

Data is the most imperative aspect of Analytics and Machine Learning. Everywhere in computing or business, data is required. But many a times, the data may be incomplete, inconsistent or may contain missing values when it comes to the real world. If the data is corrupted, then the process may be impeded or inaccurate results may be provided. Hence, Data cleaning is considered a foundational element of thebasic data science.

Data Cleaning means the process by which the incorrect, incomplete, inaccurate, irrelevant, or missing part of the data is identified and then modified, replaced or deleted as needed.

Packages Used:

Tidyverse: It is a collection of essential data science R-packages. Under the tidyverse umbrella, the packages help perform and interact with the data. There are a whole host of things one can do with data, like sub setting, transforming, visualizing and so on.

Dplyr: dplyr is a grammar of data manipulation, providing a consistent set of verbs that help solve the most common data manipulation challenges. It is simply the most useful package in R for data manipulation with the greatest advantage being the use the pipe function "*%>%"* to combine different functions in R. From filtering to grouping the data, this package does it all. It offers various functions like select, filter, group_by, summarize etc.

Functions Used:

Is.na(): In R, missing values are represented by the symbol **NA** (not available). Impossible values (e.g., dividing by zero) are represented by the symbol **NaN** (not a number). This function is used to check if a dataset contains NA values or not.

Unique(): This function is used to filter out redundant data and keep only unique values from the data frame.

Mutate(): This function adds new variables that are functions of existing variables

As.date(): This function is used to convert between character representations and objects of class “Date” representing calendar dates.

```
Sample Code:

library(tidyverse)
library(dplyr)

cfr[is.na(cfr)] <-  0
cfr$Date <- anydate(cfr$Date)

xtot <- unique((x %>% filter(date >= as.Date('2020-02-15')))$location)

na.omit(s$Total.Confirmed)
```

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 sub setting ordrillingl down data wherein data is extracted with respect to certaindefined logical conditions. Filtering is used for the following tasks:

- Analyzing results for a particular period.
- Calculating results for particular groups of interest.
- Exclude erroneous or "bad" observations from an analysis.
- Train and validate statistical models.

3.6 Packages Used:

Tidyverse: It is a collection of essential data science R-packages. Under the tidyverse umbrella, the packages help perform and interact with the data. There are a whole host of things one can do with data, like sub setting, transforming, visualizing and so on.

Dplyr: dplyr is a grammar of data manipulation, providing a consistent set of verbs that help solve the most common data manipulation challenges. It is simply the most useful package in R for data manipulation with the greatest advantage being the use the pipe function "*%>%"* to combine different functions in R. From filtering to grouping the data, this package does it all. It offers various functions like select, filter, group_by, summarize etc.

Functions Used:

Slice(): This function is used to extract rows by position.

Filter(): This function is used to extract rows that meet a certain logical criteria.

Logical Comparisons:

<: for less than

>: for greater than

<=: for less than or equal to

>=: for greater than or equal to

==: for equal to each other

!=: not equal to each other

%in%: group membership. For example, "value **%in%** c(2, 3)" means that value can takes 2 or 3.

Filter_all(), filter_at(): filter rows within a selection of variables. These functions replicate the logical criteria over all variables or a selection of variables.

Sample_n(): This function randomly select n rows

Top_n(): This function selects top n rows ordered by a variable

```
Sample Code:

library(tidyverse)
library(dplyr)

x1 <- x %>% filter(date >= as.Date('2020-02-15'))
x2 <- x %>% filter(date == as.Date('2020-06-07'))
x2 <- x2 %>% top_n(10, total_cases_per_million)
d <- x %>% filter(date == '2020-06-07')
```

3.7 Packages Used for visualization:

Ggplot2: Ggplot2 is a declarative graphics development framework focused on The Grammar of Graphics. Once the user provides the data and tells ggplot2 how to map aesthetic variables and what graphic primitives to use, it takes care of the details. In most cases, one starts with ggplot(), supplies a dataset and aesthetic mapping (with aes()), the adds on layers (like geom_point() or geom_histogram()), scales, faceting specifications (like facet_wrap()) and coordinate systems (like coord_flip()).

Plotly: This is a complement to the ggplot package which includes JavaScript libraries to provide more interactive visuals.

IV. TASKS

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

```
Code:
# import libraries
import pandas as pd
import matplotlib.pyplot as plt
# load the data
data_url_1 =
"https://gist.githubusercontent.com/mbonsma/8da0990b71ba9a09f7de395574e54df1/raw/aec88b30af87fad8d45da
7e774223f91dad09e88/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')
```

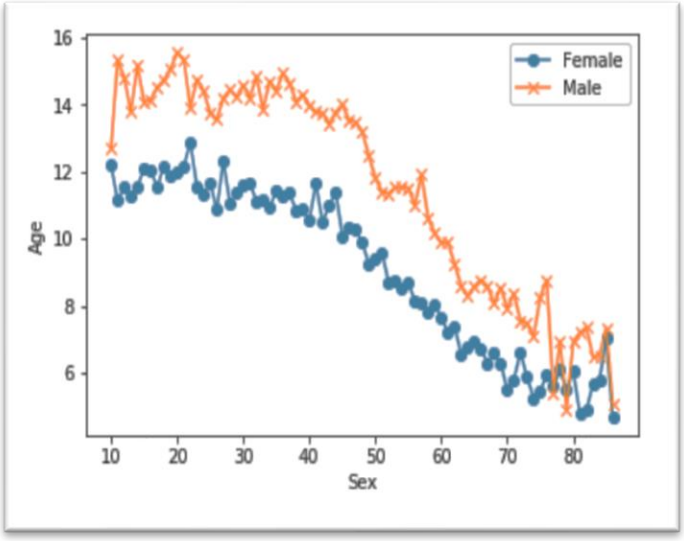

Fig 1. Refers to the graph Sex v/s Age

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

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
# ... YOUR CODE FOR TASK 2 ...
lefthanded_data['Birth_year'] = 1986 - lefthanded_data['Age']
# create a new column for the average of male and female
# ... YOUR CODE FOR TASK 2 ...
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
```

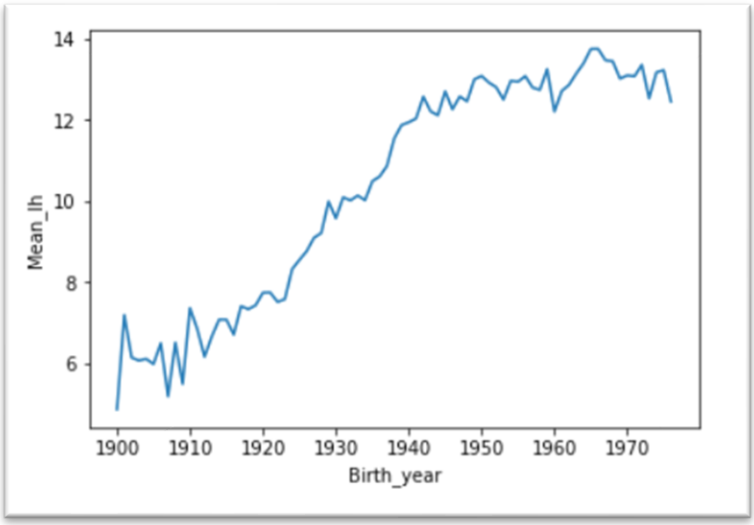

Fig 2. Refers to the code in python

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

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).

Code:

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

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
# ... YOUR CODE FOR TASK 4 ...
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')
```

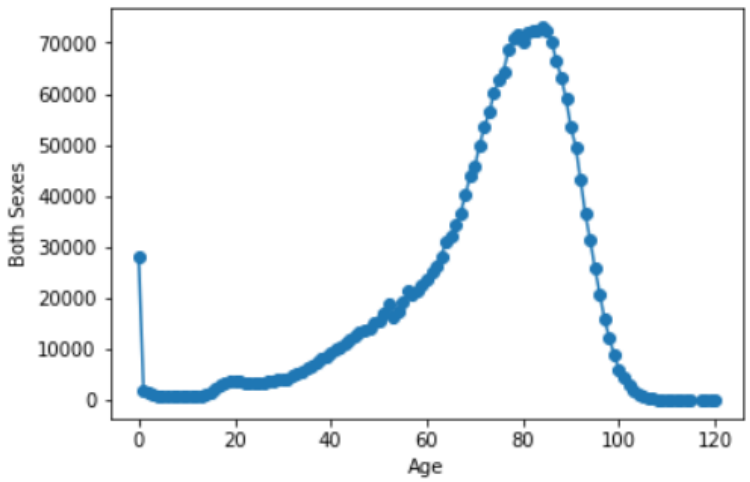

Fig: Graph of Age v/s Both Sexes

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

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 / 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))
```

4.6: Write a function to calculate `P_A_given_lh()`.

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.

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
```

4.7: Write a function to calculate `P_A_given_rh()`.

And now for right-handers.

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
```

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

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")
```

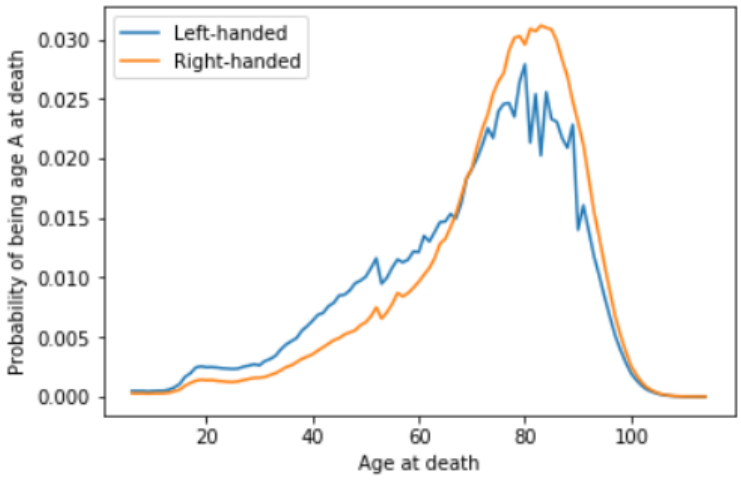

Fig: Graph of Age of Death v/s Probability

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

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(left_handed_probability))
average_rh_age =  np.nansum(ages*np.array(right_handed_probability))

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

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

Observations:

- Average age of lefthanded 67.24503662801027
- Average age of righthanded 72.79171936526477
- The difference in average ages is 5.5 years.

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

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, 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.")
```

Observations:

- We can conclude that, the difference in average ages is 2.3 years.

V. CONCLUSION AND FUTURE SCOPE

We got a 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.

VI. 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://github.com/rrmolin/Do-Left-handed-People-Really-Die-Young-DataCamp-project/blob/master/notebook.ipynb

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