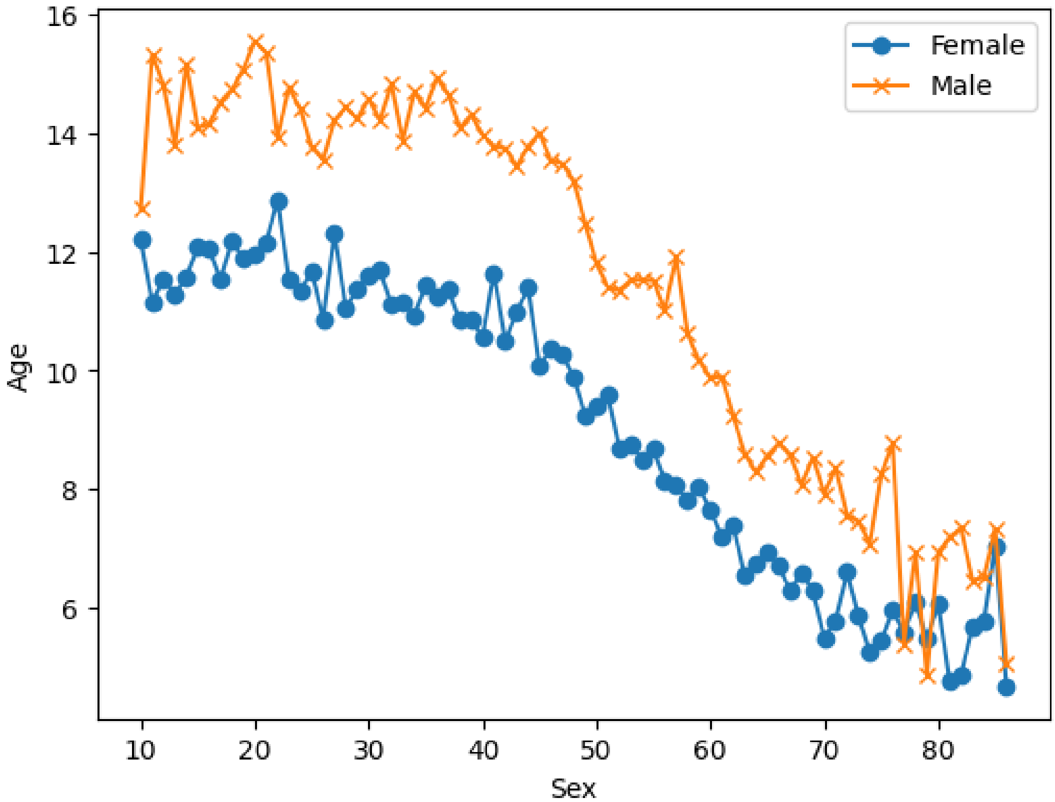
deathage

February 23, 2024

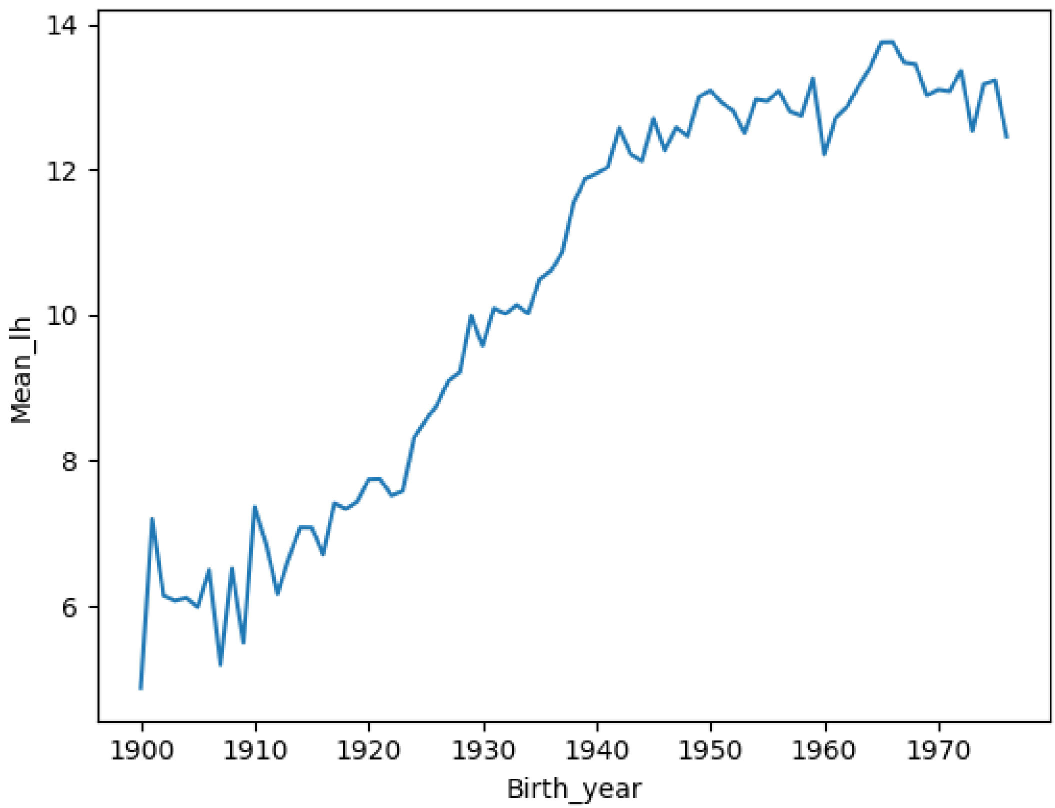
[1]:
```
# import libraries
# TASK 1 - Load the handedness data from the National Geographic survey and␣
 ↪create a scatter plot.
import pandas as pd
import matplotlib.pyplot as plt
# load the data
data_url_1 = "https://gist.githubusercontent.com/mbonsma/
 ↪8da0990b71ba9a09f7de395574e54df1/raw/
 ↪aec88b30af87fad8d45da7e774223f91dad09e88/lh_data.csv"
lefthanded_data = pd.read_csv(data_url_1)

# plot male and female left-handedness rates vs. age
%matplotlib inline
fig, ax = plt.subplots()
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()
ax.set_xlabel('Sex')
ax.set_ylabel('Age')
```

[1]:
```
Text(0, 0.5, 'Age')
```



[2]:
```
# TASK 2 - Add two new columns, one for birth year and one for mean␣
↪left-handedness, then plot the mean as afunction of 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 mean 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')
ax.set_ylabel('Mean_lh')
```

[2]:
```
Text(0, 0.5, 'Mean_lh')
```

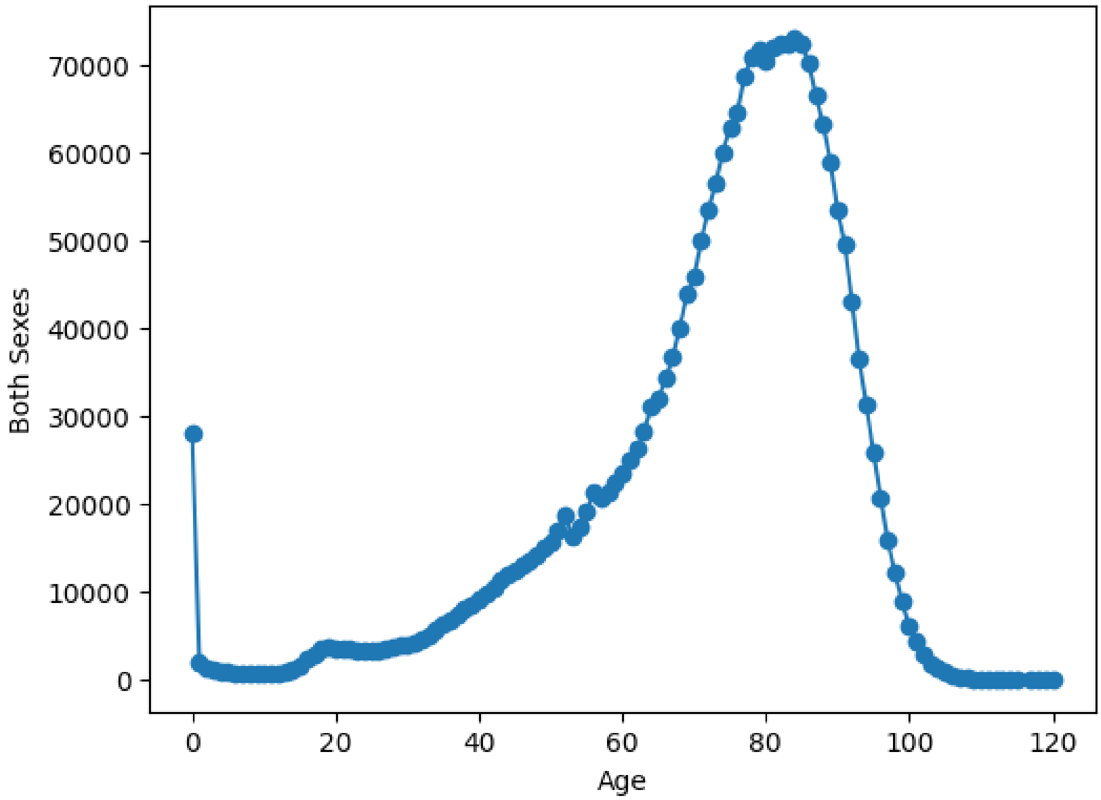


[6]:
```python
# TASK 3 - Create a function that will return P(LH | A) for particular ages of␣
↪death in a given study year.
# import library
import numpy as np
# create a function for P(LH | A)
def P_lh_given_A(ages_of_death, study_year = 1990):
    """ P(Left-handed | ages of death), calculated based 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
```

[8]:
```
# TASK 4 - Load death distribution data for the United States and plot it.
# 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')
```

[8]:
```
Text(0, 0.5, 'Both Sexes')
```



[9]:
```
# TASK 5 - Create a function called P_lh() which calculates the overall␣
↪probability of left-handedness in thepopulation for a given study year.
# sum over P_lh for each age group
def P_lh(death_distribution_data, study_year = 1990):
    """ 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

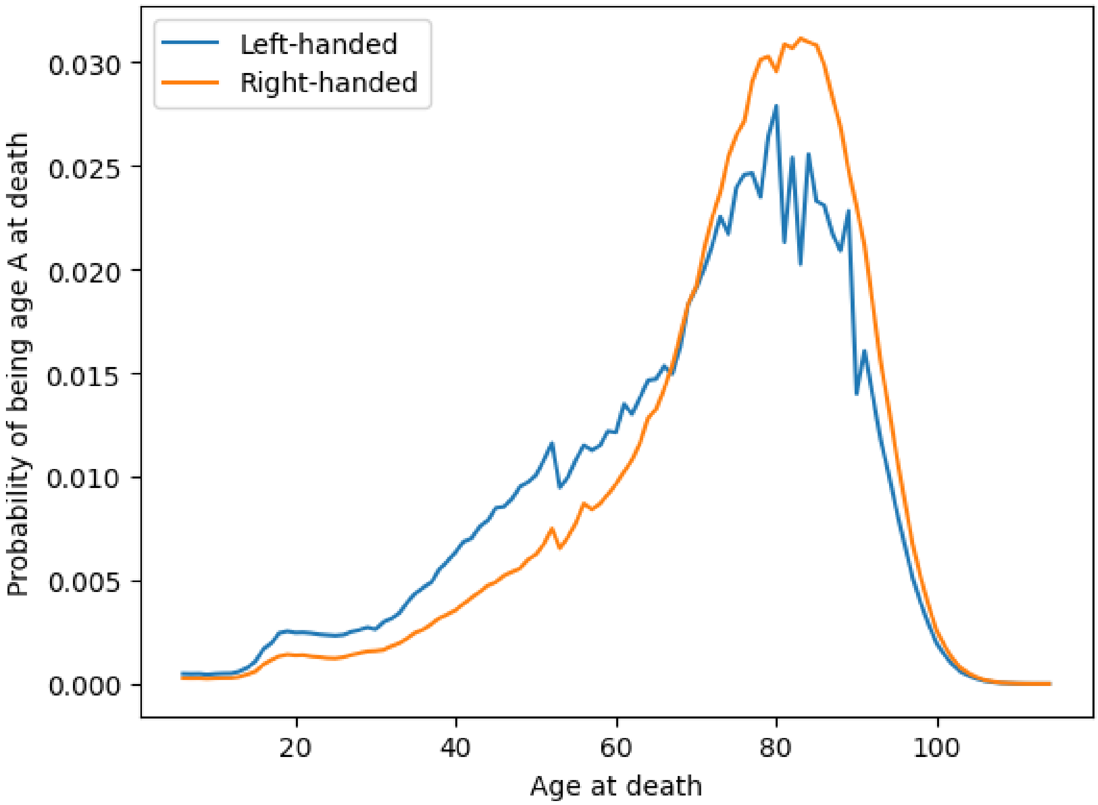
[10]:
```
# TASK 6 - Write a function to calculate P_A_given_lh().
def P_A_given_lh(ages_of_death, death_distribution_data, study_year = 1990):
```

```
    """ 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
```

[11]:
```
# TASK 7 - Write a function to calculate P_A_given_rh().
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
```

[14]:
```
# TASK 8 - Plot the probability of being a certain age at death given that␣
↪you're left- or right-handed for a rangeof ages.
ages = np.arange(6, 115, 1)

# 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()
ax.set_xlabel("Age at death")
ax.set_ylabel(r"Probability of being age A at death")
```

[14]:
```
Text(0, 0.5, 'Probability of being age A at death')
```



[17]:

```
# TASK 9 - Find the mean age at death for left-handers and right-handers.
# calculate mean 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)))
```

```
Average age of lefthanded67.24503662801027
Average age of righthanded72.79171936526477
The difference in average ages is 6
```

[19]:

```
# TASK 10 - Redo the calculation from Task 8, setting the study_year parameter␣
  ↪to 2018.
# 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.
```