MATH 442
Class Project 1

Author:
Mia Marche

Date:
October 24, 2023

MATH 442
Class Project 1

"1.9 Project" from pages 45-47 of chapter 1 of the book by Brauer and Castillo-Chavez (2nd Edition) follows on next two pages. Grading for the project will follow the same rubric as the homework, so everything should be clearly explained.

In addition, you cannot use any built-in regression models to do the least squares approximation. You must work out the math/programming yourself. Programming should be done in Python.

Some key points that must be included in your submission:

An introduction and a conclusion must be included.

An appendix which contains snippets of relevant code. (I may ask you to email your code to me so that I can check if it actually runs.)

Relevant plots should be included. E.g. plots of the data with best fit curves.

For your convenience, Table 1.1 referenced in Problems 2 & 5 is given below.

Year	Population
1790	3,900,000
1800	5,300,000
1810	7,200,000
1820	9,600,000
1830	12,900,000
1840	17,100,000
1850	23,100,000
1860	31,400,000
1870	38,600,000
1880	50,200,000
1890	62,900,000

Year	Population
1900	76,000,000
1910	92,000,000
1920	105,700,000
1930	122,800,000
1940	131,700,000
1950	150,700,000
1960	179,000,000
1970	205,000,000
1980	226,500,000
1990	248,700,000

TABLE 1.1. Census data of US from 1790 to 1990.

Introduction

The rate of population growth is subject to various changing factors that over the past decades, scientists have attempted to model using various mathematical functions. In this project, we follow in their footsteps to use different mathematical models in conjunction with historical population data of the U.S. from 1790 to 1990 in order to predict future trends in population growth.

There are two fundamental assumptions that must be established: first, populations grow exponentially under positive per capita population rates, and second, the extent of population growth is restricted by the constraints of limited resources. Thus, we take this into consideration when working with our models, beginning with the logistic population model: $x(t) = \frac{Kx_0e^{rt}}{K-x_0+x_0e^{rt}}$, where $x(t)$ denotes the size of the population at time t, x_0 is the initial population size for which the growth rate depends on, r is the intrinsic growth rate, t is time in years, and K is the carrying capacity of the population. This model anticipates a sharp initial growth rate for $0 < x_0 < K$, followed by a decline over time, ultimately leading to a convergence towards a limit.

Throughout this experiment, we will also utilize the method of least squares to determine the best straight line that minimizes the sum of the squares of the vertical distances between the data points and the line: $\frac{\sum(t_i-\bar{t})(y_i-\bar{y})}{\sum(t_i-t)^2}$.

Lastly, for estimating the rate of change in population size $x'(t)$ we employ two different methods with two different equations. The first method uses the following equation; $x'(t) = \frac{X_{i+1}-X_i}{h}$ where x_i and x_{i+1} are two consecutive measurements taken with a single time interval h, which we substitute for 10. The second method, which is said to be a more accurate than the first, uses the equation; $x'(t) = \frac{X_{i+1}-X_{i-1}}{2h}$ which works with the values x_{i-1} and x_{i+1} and the variable $2h$. We continue with such equations to test the advantages and disadvantages of using mathematical models to predict population growth.

Problem 1.9.1: Derive the relation

$$\log\left(\frac{K-x}{x}\right) = \log\left(\frac{K-x_0}{x_0}\right) - rt$$

from

$$x(t) = \frac{Kx_0e^{rt}}{K-x_0+x_0e^{rt}}.$$

Thus, if we plot $\log \frac{K-x}{x}$ against t we should obtain a straight line. However, there is a problem: We do not know the value of K. We may try to avoid this problem by estimating K by eye from the graph of the data points(t, x). If we obtain similar curves when we fit the data to a logistic curve for several different values of K, so that our results are not very sensitive to changes in K, then we may have some confidence in these results.

To begin the derivation we will write what we were given:

$$x(t) = \frac{Kx_0e^{rt}}{K-x_0+x_0e^{rt}},$$

and multiply both sides by $(K - x_0 + x_0e^{rt})$ (1). Then, we divide both sides by x_0e^{rt} and x (2). Next, we subtract the left side by $\frac{x_0e^{rt}}{x_0e^{rt}}$ and the right side by $\frac{x}{x}$, which both are equal to one, keeping the equations constant (3). We can now take the log of both sides (4) and simplify the equation (5), resulting in the final equation of:

$$\log\left(\frac{K-x_0}{x_0}\right) - rt = \log\left(\frac{K-x}{x}\right)$$

$$\begin{aligned}
x(t) = \frac{Kx_0e^{rt}}{K-x_0+x_0e^{rt}} &\implies x(t) \times (K - x_0 + x_0e^{rt}) = Kx_0e^{rt} && (1)\\
&\implies \frac{K-x_0+x_0e^{rt}}{x_0e^{rt}} = \frac{K}{x} && (2)\\
&\implies \frac{K-x_0+x_0e^{rt}}{x_0e^{rt}} - \frac{x_0e^{rt}}{x_0e^{rt}} = \frac{K}{x} - \frac{x}{x} && (3)\\
&\implies \frac{K-x_0}{x_0e^{rt}} = \frac{K-x}{x} && (4)\\
&\implies \log\left(\frac{K-x_0}{x_0} \times \frac{1}{e^{rt}}\right) = \log\left(\frac{K-x}{x}\right) && (5)\\
&\implies \log\left(\frac{K-x_0}{x_0}\right) - rt = \log\left(\frac{K-x}{x}\right) && (6)
\end{aligned}$$

Problem 1.9.2: For each of the values $K = 200$, $K = 250$, $K = 300$, use the data of Table 1.1 to plot $\log \frac{K-x}{x}$ as a function of t.

Solution.

The appendix will provide an explanation for the creation of these plots of the logistic growth model for different values of K (the carrying capacity). Note the populations used in all calculations were multiplied by 10^{-6}.

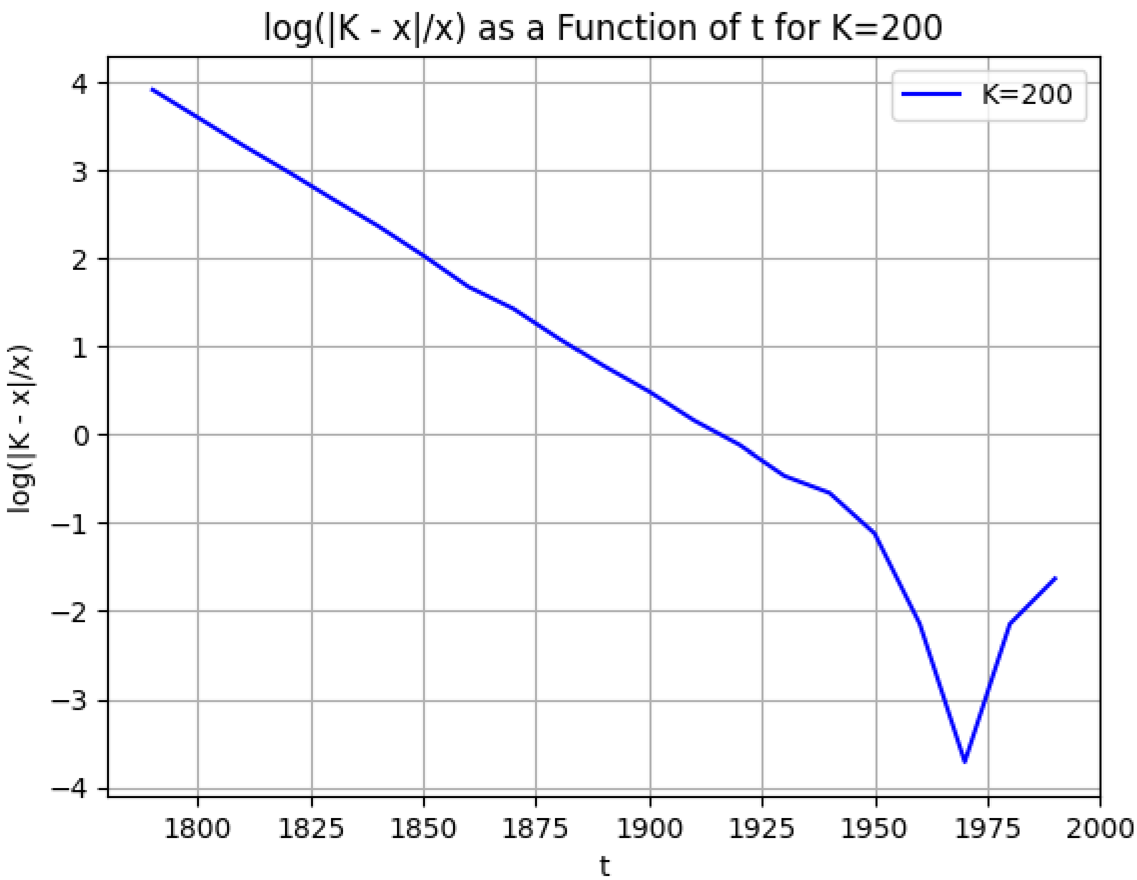

Figure 1: Plot $\log(\frac{K-x}{s})$ versus t for K=200.

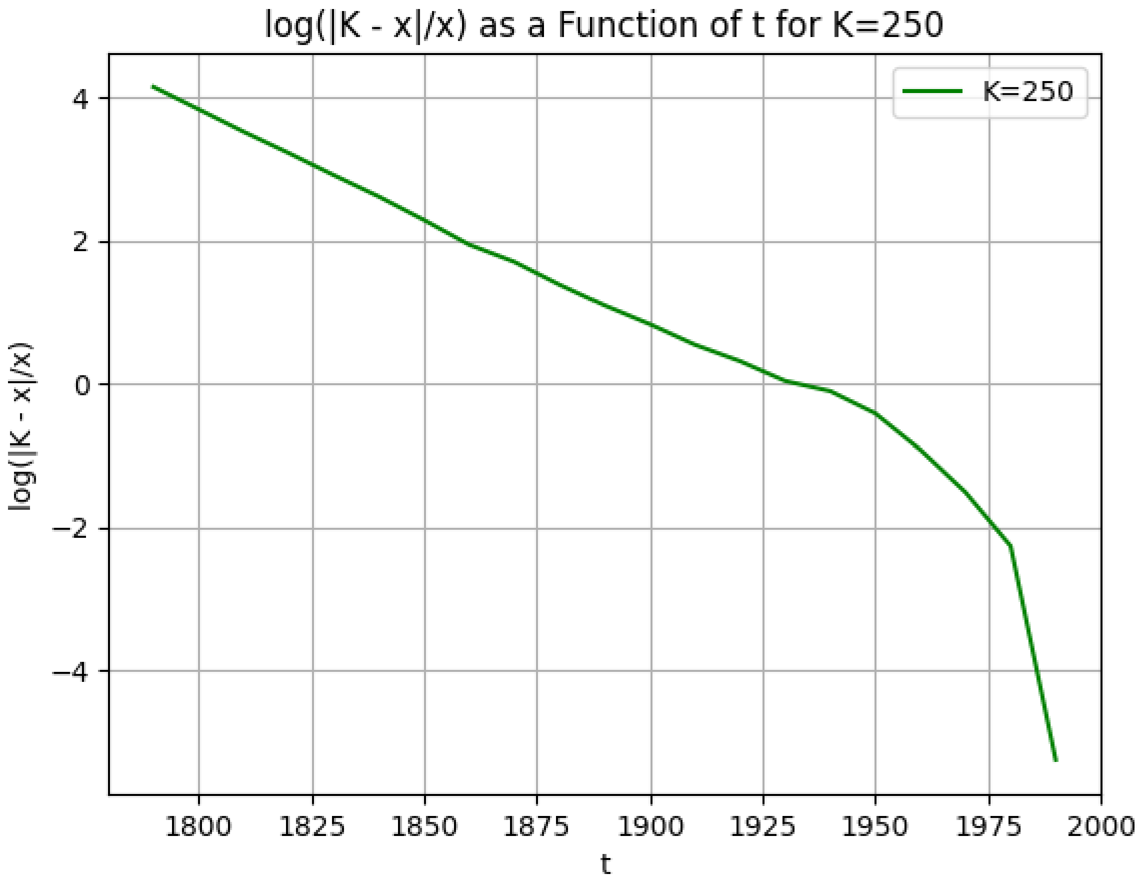

Figure 2: Plot $\log(\frac{K-x}{s})$ versus t for K=250.

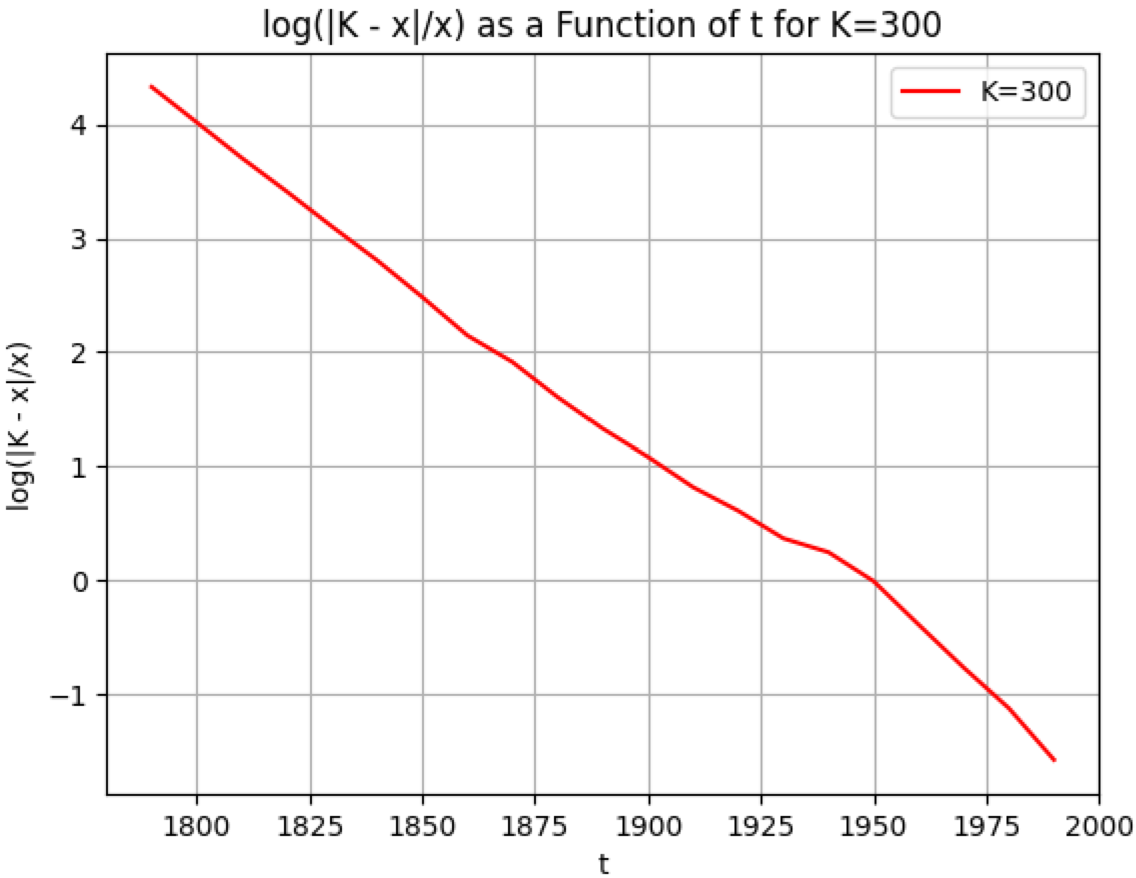

Figure 3: Plot $\log(\frac{K-x}{s})$ versus t for K=300.

Problem 1.9.3: For each of the values $K = 200$, $K = 250$, $K = 300$, use the method of least squares (Exercise 12,Section1.1) to estimate the slope $-r$ and the intercept $\log \frac{K-x_0}{x_0}$.

Solution.

The appendix will provide the code used to estimate the slope and the intercept. But I will discuss the methods used to estimate these values below. Firstly, what we typically call x and y are t and $\log \frac{K-x_0}{x_0}$ respectively. This value y=$\log \frac{K-x_0}{x_0}$ was derived in Problem 1.9.1. So for each of the values of K, $K = 200$, $K = 250$, $K = 300$, you will calculate y by plugging in the each of the values for K and each population data point from Table 1.1. We will call $\log \frac{K-x_0}{x_0}$ the letter y moving forward. Second, you will need to calculate the average of t and the average of y (There are 3 different y averages since there are 3 different K values). Then, for the slope for each value of K you calculate:

$$\frac{\sum (t_i - \bar{t})(y_i - \bar{y})}{\sum (t_i - \bar{t})^2}$$

And for the y-intercept that is calculated as follows:

$$\log \left(\frac{K - x_0}{x_0} \right)$$

Both of those equations can be found be simply rearrange what was given in Problem 1.9.1:

$$\log \left(\frac{K - x}{x} \right) = \log \left(\frac{K - x_0}{x_0} \right) - rt$$

So the solutions are as follows:
For K = 200:
Estimated slope (-r): 0.03318943784836223
Estimated intercept: 3.9176481800667

For K = 250:
Estimated slope (-r): 0.03486806660807836
Estimated intercept: 4.144761404261409

For K = 300:
Estimated slope (-r): 0.027929197618063527
Estimated intercept: 4.329720681971945

Problem 1.9.4: For each of the values $K = 200$, $K = 250$, $K = 300$, use the results obtained in Question 3 to give a function describing the population size, and use this function to predict the result of the Year 2000 census.

Method II: From the logistic model, we have

$$\frac{x'}{x} = r\left(1 - \frac{x}{K}\right)$$

Thus, if we plot $\frac{x'(t)}{x(t)}$ as a function of x, we should obtain a straight line with x-intercept K and slope $\frac{-r}{K}$. The problem here is that our data describes x, not x'. However, we can use the date to estimate x'. If x_i and x_{i+1} are two consecutive measurements taken with a time interval h, we may approximate x'_i by $\frac{x_{i+1}-x_i}{h}$. Actually, the approximation $\frac{x_{i+1}-x_{i-1}}{2h}$ is an approximation to x'_i which is significantly more accurate.

Solution.
To create a function to predict the result of the Year 2000 census use:
$K = 200$; $K = 250$; $K = 300$; $r = .0331894378$; $r = .034868067$;
$r = .0279291976$; $x_0 = 3.9$ $t = 2000 - 1790 = 210$

We will substitute these values in to the following equation:

$$x(t) = \frac{Kx_0e^{rt}}{K - x_0 + x_0e^{rt}}.$$

Next substitute in the appropriate values. We will start with $K = 200$:

$$x(t = 210) = \frac{200 * 3.9 * e^{(.0331894378)(210)}}{200 - 3.9 + 3.9e^{(.0331894378)(210)}}$$

$$x(t = 210) = 190.63274629243722$$

For $K = 250$:

$$x(t = 210) = \frac{250 * 3.9 * e^{(.034868067)(210)}}{250 - 3.9 + 3.9e^{(.034868067)(210)}}$$

$$x(t = 210) = 239.69096796064346$$

For $K = 300$:

$$x(t = 210) = \frac{300 * 3.9 * e^{(.0279291976)(210)}}{300 - 3.9 + 3.9e^{(.0279291976)(210)}}$$

$$x(t = 210) = 245.69209694956865$$

Problem 1.9.5: Use the data of Table 1.1 to estimate x_i' and then plot $\dfrac{x_i'}{x_i}$ against x_i.

Solution.

Below are the equations used to find $x'(t)$. Our value for h (time interval) is 10 in both methods.

Method 1:

$$x'(t) = \frac{X_{i+1} - X_i}{h}$$

Method 2:

$$x'(t) = \frac{X_{i+1} - X_{i-1}}{2h}$$

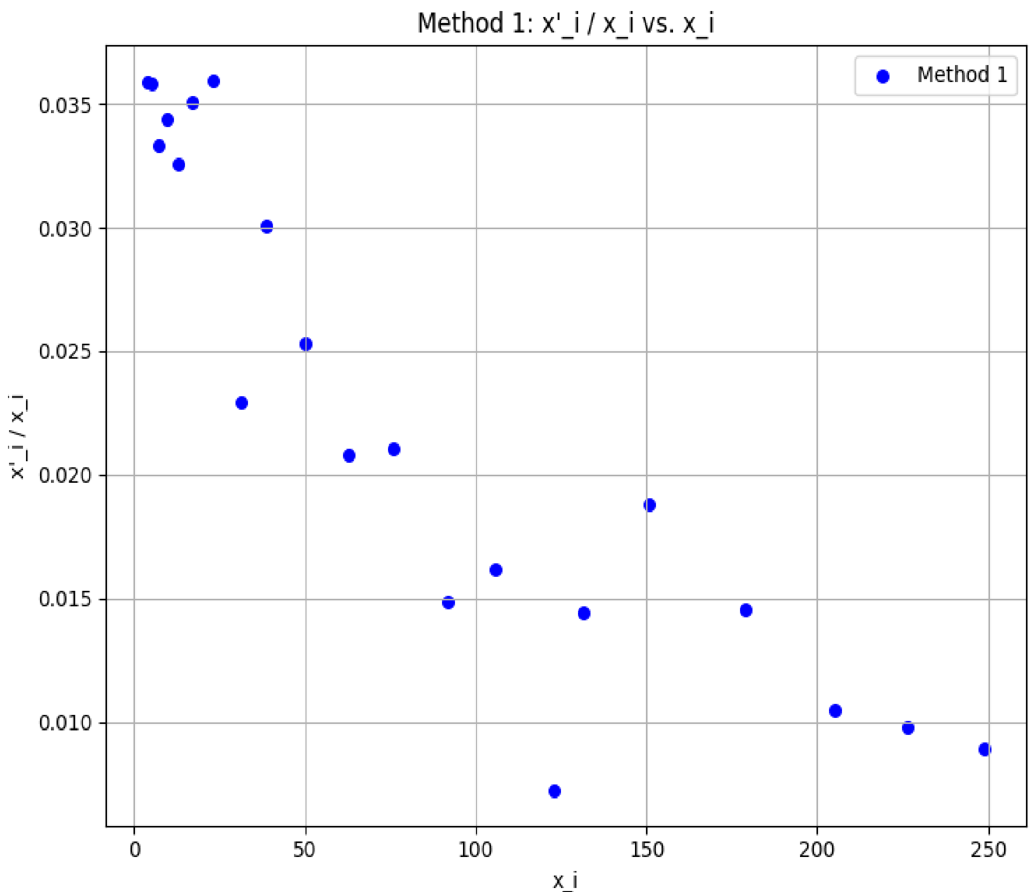

Figure 4: Plot of $\frac{x_i'}{x_i}$ against x_i using method I.

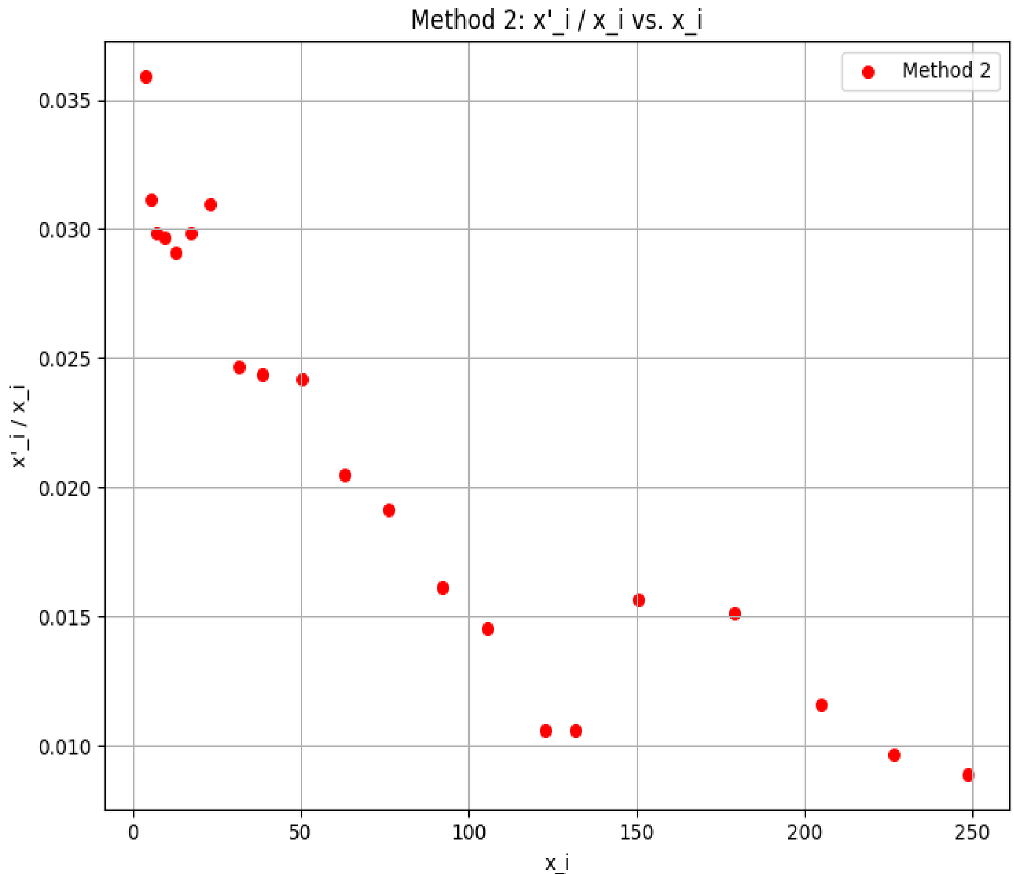

Figure 5: Plot of $\frac{x_i'}{x_i}$ against x_i using method II.

Method 1

x	x_i	x_i/x
3.9	0.14	0.0358974
5.3	0.19	0.0358491
7.2	0.24	0.0333333
9.6	0.33	0.034375
12.9	0.42	0.0325581
17.1	0.6	0.0350877
23.1	0.83	0.0359307
31.4	0.72	0.0229299
38.6	1.16	0.0300518
50.2	1.27	0.0252988
62.9	1.31	0.0208267
76	1.6	0.0210526
92	1.37	0.0148913
105.7	1.71	0.0161779
122.8	0.89	0.00724756
131.7	1.9	0.0144267
150.7	2.83	0.018779
179	2.6	0.0145251
205	2.15	0.0104878
226.5	2.22	0.00980132
248.7	2.22	0.00892642

This table provides each value of x, x_i, and $\frac{x_i}{x}$ using Method 1.

Method 2

x	x_i	x_i/x
3.9	0.14	0.0358974
5.3	0.165	0.0311321
7.2	0.215	0.0298611
9.6	0.285	0.0296875
12.9	0.375	0.0290698
17.1	0.51	0.0298246
23.1	0.715	0.0309524
31.4	0.775	0.0246815
38.6	0.94	0.0243523
50.2	1.215	0.0242032
62.9	1.29	0.0205087
76	1.455	0.0191447
92	1.485	0.0161413
105.7	1.54	0.0145695
122.8	1.3	0.0105863
131.7	1.395	0.0105923
150.7	2.365	0.0156934
179	2.715	0.0151676
205	2.375	0.0115854
226.5	2.185	0.0096468
248.7	2.22	0.00892642

This table provides each value of x, x_i, and $\frac{x_i}{x}$ using Method 2.

Problem 1.9.6: Use the method of least squares to estimate r and K, and use your result to predict the result of the Year 2000 census.

In question 6, you should find that the data is quite close to your straight line up to 1940 but not for 1950 and later.

Solution.
To begin this problem, we must understand the following relationship:

$$(\frac{x_i'}{x_i}) = \frac{-r}{K}x + r \approx y = ax + b$$

We will use the previous values for $\frac{x_i'}{x_i}$ for method 1 and method 2 which you can refer to in Problem 1.9.5. to use as the "y-data". Additionally, we used the population values given in Table 1.1 as the "x-data".

Then, we will calculate the slope in this case using the following equation:

$$\frac{-r}{K} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

Finally, we calculated the intercept using the following equation:

$$r = \bar{y} - \hat{a}\bar{x}$$

Since we need the K-value to predict the Year 2000 population, we needed to isolate K. We did this by dividing the intercept by the slope as shown below:

$$-\frac{r}{\frac{-r}{K}} = K$$

We substituted the calculated r and K-values along with $x_o = 3.9$ and $t = 210$ into the following equation:

$$x(t) = \frac{Kx_0e^{rt}}{K - x_0 + x_0e^{rt}}.$$

The appendix will provide the code used to estimate the following solutions:
<u>For method 1</u>:
The r-value is: 0.03176461037536661
The k-value is: 328.55533840463147
The estimated population for Year 2000 is: 297.193187497686
<u>For method 2</u>:
The r-value is: 0.03500017017556448
The k-value is: 359.66864848004843
The estimated population for Year 2000 is: 339.7533485941879

Problem 1.9.7: Using only the data from 1950 on, estimate x_i' and plot $\dfrac{x_i'}{x_i}$ against x_i. Then use the method of least squares to estimate r and K, and use your result to predict the result of the Year 2000 census.

Solution.

Since the problem is asking us for the data from 1950-1990, we adjusted the population and time values accordingly. In order to estimate the $\frac{x_i'}{x_i}$ values, we used the same equations you can refer to in problem 1.9.5. This in turn, provided us with data to make two separate plots for method 1 and method 2.

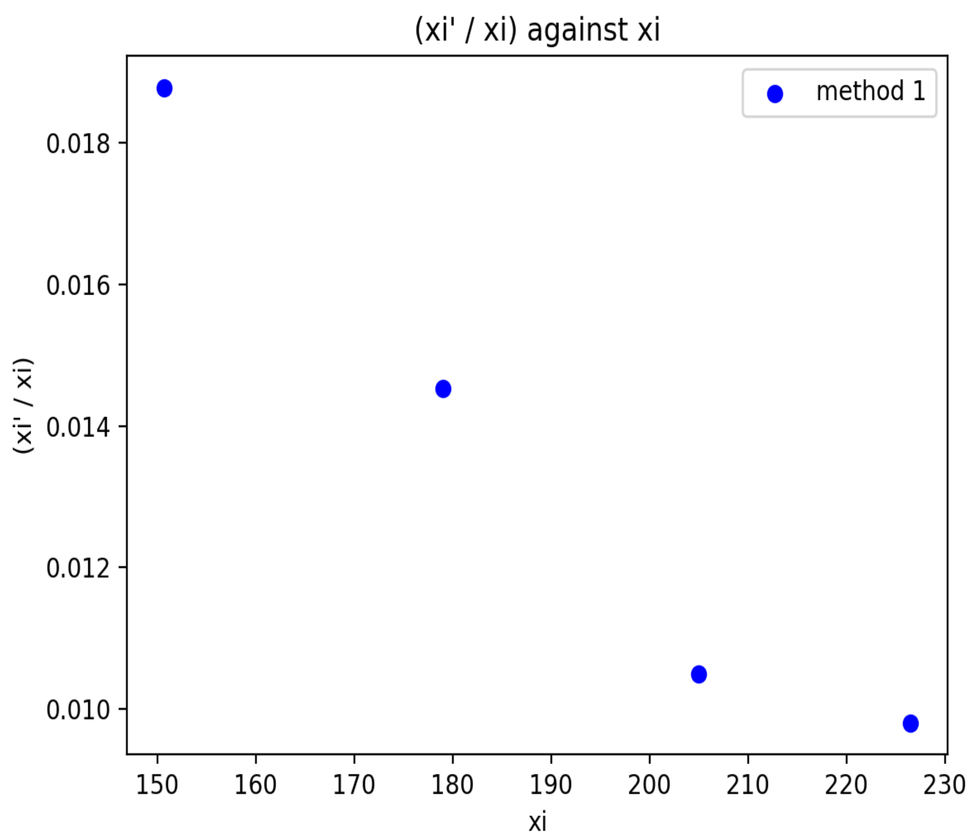

Figure 6: Plot $\frac{x_i'}{x}$ against x_i using method 1

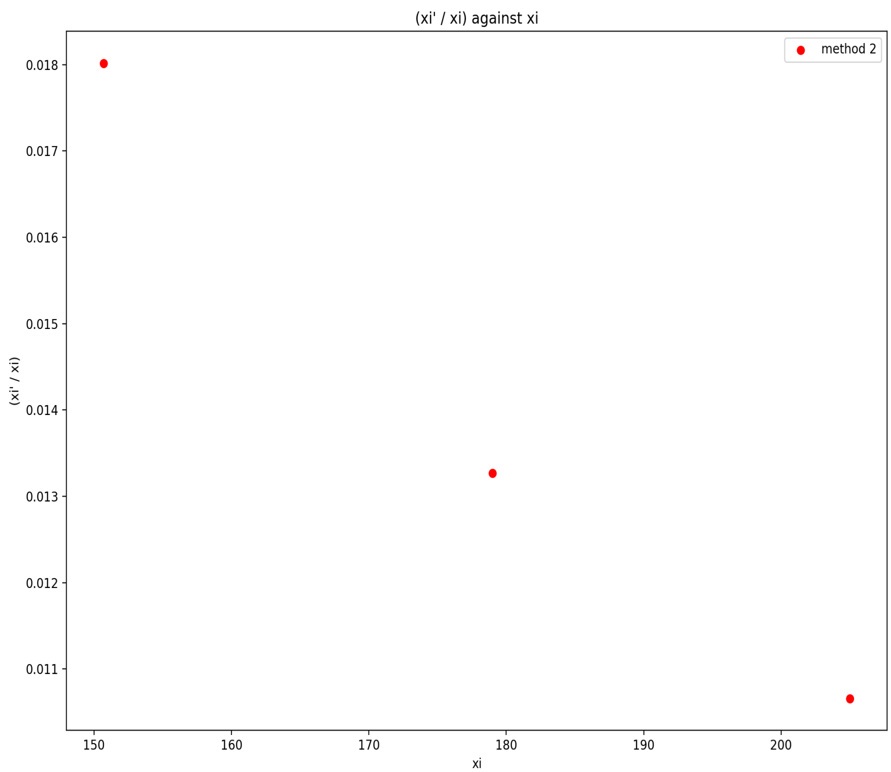

Figure 7: Plot $\frac{x_i'}{x}$ against x_i using method 2

The same process and equations used to estimate r, K, and the population in Year 2000 can be referred to in Problem 1.9.6.

The appendix will provide the code used to estimate the following solutions:
<u>For Method 1</u> : The r-value is: 0.026934364223571844
The k-value is: 401.90507869616
The estimated population for Year 2000 is: 399.5767101634981
<u>For Method 2</u> : The r-value is: 0.020788611516317987
The k-value is: 616.780988325134
The estimated population for Year 2000 is: 593.4582953900559 millions

Problem 1.9.8: What reasons might there be for the apparent jump in carrying capacity between 1940 and 1950?

Over the past 150 years, immigration to the U. S. A. has varied considerably from year to year but has averaged more than 150,000 per year. The logistic model does not include any immigration.

Solution.

The apparent jump in carrying capacity between 1940 and 1950 can most likely be explained by the impacts that World War 2 had on the country's population and economy. After soldiers returned home following the end of the war in 1945, a large baby boom took place as young adults of the time sought to settle down and return to normalcy. Families began to move out of inner cities into newly built affordable housing in the suburbs, increasing the number of white-collar workers, and so in conjunction with the postwar confidence, an era of prosperity led the U.S. to become a New Global Power.

From a mathematical viewpoint, carrying capacity can increase when there are enough resources to support a larger population. Thus, we can assume that the aforementioned postwar population boom along with the economic expansion between the 1940's and 1950's can be exemplified by the sudden inflation in the carrying capacity of the time.

Problem 1.9.9: Suggest a modification of the logistic model which would include immigration. What would you expect to be the effect of this modification on the estimate of carrying capacity which would be obtained using the same data?

Solution.

Below is the logistic equation.

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

If you consider immigration (I) you can modify the equation as such:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K}) + I$$

Immigration, I can be constant or it can vary by time. If we want to show the logistic equation to include immigration that varies by time it would like this:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K}) + I(t)$$

Immigration in the logistic growth model could increase the estimate of K. There are 2 cases in which this would happen if immigration is continuous and considerable.

Problem 1.9.10: Taking into account all that you have done in this project, what is your best guess for the result of the Year 2000 census?

Solution.

Our best guess for the result of the year 2000 census takes into account all four results of problems 1.9.6 and 1.9.7 according to our perceived accuracy of each. We gave more weight to the results of 1.9.6 than 1.9.7 as we believe that since problem 1.9.7 only uses the population values from 1950 on, it would be overestimating the future population growth as the population boom of the 1950s could not be sustained for later years. Thus, we assigned problem 1.9.6 a collective weight of .7 and 1.9.7 a weight of .3. As for the two methods of estimating x', we focused on method 2 the most as it produced the smallest values of $\frac{x_i'}{x_i}$, proving to be more accurate than method 1, resulting in a collective weight of .45 for method 1 and .55 for method 2.

$$297.193(.315) + 339.753(.385) + 399.577(.135) + 593.458(.165) = 376.284. \tag{7}$$

Using equation 7, our estimated population for the year 2000 is 376 million. Although higher than realistically possible compared to the given population data of 1990, this is our best answer using our project's results.

Conclusion

Throughout this project, our team approached the matter of population growth through a mathematical lens. We utilized the logistic population model along with historical population data of the U.S. from 1790 to 1990 in order to predict future trends in population growth. Throughout it, we focused on different relevant carrying capacities, examining their connection to the population size sustainable by available resources, alongside the corresponding intrinsic growth rate and per capita growth rate values.

We conclude that although the mathematical models used are valid in their approach to estimate population growth, there are many limitations that prevent them from being accurate predictors of population. Although we tried our best to give the most conscientious guess for the result of the Year 2000 census based on our results, we acknowledge that compared to the population data given from 1990, our estimation is strikingly inflated. Even our smallest estimated population, 297.193 million calculated in Problem 1.9.6 with Method II, is still exceptionally high.

Indeed, it is important to recognize the limitations of mathematical models when modeling dynamic sciences such as population. For example, we see that the logistic population model provides a simplified representation of populations disregarding changing mortality rates, population densities, and technological innovations such as carbon emissions, war, and the economy. As previously mentioned, World War II and immigration have had great impacts on the U.S. population over the past decades that cannot be represented with such a model.

As we continue to study mathematical models, we accept their importance considering their various factors and characteristics, as well as acknowledge their faults and how it is up to us to utilize their results in the most responsible and appropriate way.

Appendix

Mia Marche

October 24, 2023

Contents

A Python Code

A.1 Question 1.9.2

```
from google.colab import drive
drive.mount('/content/drive')
!ls '/content/drive/My Drive/MATH 442/Project_1'
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

#read in data
file_path = '/content/drive/My Drive/MATH 442/Project_1/censusdata.txt'
dataset= pd.read_csv(file_path, delimiter='\t')

# Extract t and x(t) from the dataset
t = dataset['t']
x = dataset['x(t) short']

# Values of K and corresponding colors
K_values = [200, 250, 300]
colors = ['b', 'g', 'r']

# Create separate graphs for each value of K with different colors
for K, color in zip(K_values, colors):
    function = np.log(abs((K - x)) / x)
    plt.figure()  # Create a new figure for each plot
    plt.plot(t, function, label=f'K={K}', color=color)
    plt.xlabel('t')
    plt.ylabel('log(|K - x|/x)')
    plt.legend()
    plt.title(f'log(|K - x|/x) as a Function of t for K={K}')
    plt.grid(True)

# Show the plots
plt.show();
```

A.2 Question 1.9.3

```
import math
"""
```

```
estimate_slopes_and_intercepts(t, x, K_values) estimates
    slope and
intercept takes in parameters year, population, and the
    list of K_values
"""

def estimate_slopes_and_intercepts(t, x, K_values):
    slope_r_estimates = []
    intercept_estimates = []

    for K in K_values:
        Y = np.log(abs((K - x)) / x)
        mean_t = np.mean(t)
        mean_Y = np.mean(Y)
        numerator = np.sum((t - mean_t) * (Y - mean_Y))
        denominator = np.sum((t - mean_t) ** 2)
        slope = numerator / denominator
        x0 = x.iloc[0]
        intercept = math.log((K - x0) / x0)

        slope_r_estimates.append(-slope)
        intercept_estimates.append(intercept)

    return slope_r_estimates, intercept_estimates


slope_r_estimates, intercept_estimates =
    estimate_slopes_and_intercepts(t, x, K_values)

for i, K in enumerate(K_values):
    print(f"For K = {K}:")
    print(f"Estimated slope (-r):
        {slope_r_estimates[i]}")
    print(f"Estimated intercept (log((K - x_0) / x_0)):
        {intercept_estimates[i]}")
    print()
```

A.3 Question 1.9.4

```
# making x(t)=Kx_0e^rt/K + x_0 + x_0e^rt function

def calculate_x_t(K_values, x_0, slope_r_estimates,
    time=210):
    x_t_values = []
```

```
    for K, r in zip(K_values, slope_r_estimates):
        x_t = (K * x_0 * math.exp(r * time) )/ (K + x_0
            + x_0 * math.exp(r * time))
        x_t_values.append(x_t)

    return x_t_values

results = calculate_x_t(K_values, x[0],
    slope_r_estimates)

print("So the solutions are as follows:\n")
for K, x_t, r in zip(K_values, results,
    slope_r_estimates):
    print(f"For K = {K}:")
    print(f"For r = {r}")
    print(f"x(t=210) = {x_t}\n")
```

A.4 Question 1.9.5

Method 1

```
# Initialize a list to store x' estimates
x_prime_estimates_method1 = []

# Time interval 'h'
h = 10  # You can adjust this based on your data

# Calculate x' for each data point using Method 1
for i in range(len(x)):
    if i == 0:
        x_prime_estimate = (x[i + 1] - x[i]) / h
    elif i == len(x) - 1:
        x_prime_estimate = (x[i] - x[i - 1]) / h
    else:
        x_prime_estimate = (x[i + 1] - x[i]) / h
    x_prime_estimates_method1.append(x_prime_estimate)

# The list x_prime_estimates_method1 now contains x'
    estimates for each data point
print(len(x_prime_estimates_method1))
```

Method 2

```
# Initialize a list to store x' estimates
```

```
x_prime_estimates_method2 = []

# Time interval 'h'
h = 10  # time between each year

# Calculate x' for each data point using Method 2
for i in range(len(x)):
    if i == 0:
        x_prime_estimate = (x[i + 1] - x[i]) / h
    elif i == len(x) - 1:
        x_prime_estimate = (x[i] - x[i - 1]) / h
    else:
        x_prime_estimate = (x[i + 1] - x[i - 1]) / (2 *
            h)
    x_prime_estimates_method2.append(x_prime_estimate)

# The list x_prime_estimates_method2 now contains x'
    estimates for each data point
print(len(x_prime_estimates_method2))
```

Creating the Plot

```
# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Plot x'_i / x_i against x_i for Method 1 in the first
    subplot, including the first and last data points
ax1.scatter(x, [x_prime / xi for xi, x_prime in zip(x,
    x_prime_estimates_method1)], label='Method 1', s=30,
    marker='o', c='b')
ax1.set_xlabel('x_i')
ax1.set_ylabel('x\'_i / x_i')
ax1.set_title('Method 1: x\'_i / x_i vs. x_i')
ax1.grid(True)

# Plot x'_i / x_i against x_i for Method 2 in the second
    subplot, including the first and last data points
ax2.scatter(x, [x_prime / xi for xi, x_prime in zip(x,
    x_prime_estimates_method2)], label='Method 2', s=30,
    marker='o', c='r')
ax2.set_xlabel('x_i')
ax2.set_ylabel('x\'_i / x_i')
ax2.set_title('Method 2: x\'_i / x_i vs. x_i')
ax2.grid(True)
```

```
# Display legends for both subplots
ax1.legend()
ax2.legend()

plt.tight_layout()
plt.show()

x_prime_over_x_method1 = [x_prime / xi for x_prime, xi
    in zip(x_prime_estimates_method1, x)]
x_prime_over_x_method2 = [x_prime / xi for x_prime, xi
    in zip(x_prime_estimates_method2, x)]
```

Note: From this point forward the variable names and the code has changed.

A.5 Question 1.9.6

```
def p_estimation(yval, x1, x2):
     xbar = sum(yval)/len(yval)
     xminus_xbar = []

     #for loop to create x-xbar data set
     for j in range (0,len(yval)):
          xminus = yval[j]-xbar
          xminus_xbar.append(xminus)

     #average (x'/x)
     avg_xp1 = sum(x1)/len(x1)
     avg_xp2 = sum(x2)/len(x2)

     xp1_minus_avg = []
     xp2_minus_avg = []

     #for loop to create (x'/x) - (x'/x)_bar
     for j in range (0,len(x1)):
          xp1bar = x1[j] - avg_xp1
          xp1_minus_avg.append(xp1bar)

     for j in range(0,len(x2)):
          xp2bar = x2[j] - avg_xp2
          xp2_minus_avg.append(xp2bar)

     num_xp1 = []
     num_xp2 = []
```

```
#for loop to get the numerator of the sum of least
    squares for slope
for j in range (0,len(xp1_minus_avg)):
     numxp1 = xminus_xbar[j]*xp1_minus_avg[j]
     num_xp1.append(numxp1)

for j in range(0,len(xp2_minus_avg)):
     numxp2 = xminus_xbar[j]*xp2_minus_avg[j]
     num_xp2.append(numxp2)

den_xminus_xbar = []

#for loop to get the denominator of the sum of least
    squares for the slope
for j in range (0,len(xminus_xbar)):
     den1 = math.pow(xminus_xbar[j],2)
     den_xminus_xbar.append(den1)

# solve for slope depending on the method used, in
    this case, the slope is (-r/k)
s_xp1 = sum(num_xp1)/sum(den_xminus_xbar)
s_xp2 = sum(num_xp2)/sum(den_xminus_xbar)

#solve for the intercept depending on the method, in
    this case, the intercept is r-value
b_xp1 =  avg_xp1- (s_xp1*xbar)
b_xp2 =  avg_xp2- (s_xp2*xbar)

print("The intercept(r-value) based for method 1 is:
    ",b_xp1)
print("The intercept(r-value) based for method 2 is:
    ",b_xp2)

k_xp1 = -(b_xp1/s_xp1)
k_xp2 = -(b_xp2/s_xp2)

print("The k-value based on method 1 is: ", k_xp1)
print("The k-value based on method 2 is: ", k_xp2)

print(" ")
p2000_rmethod1 =
    (k_xp1*yval[0]*exp(b_xp1*210))/(k_xp1 - yval[0] +
    yval[0]*exp(b_xp1*210))
p2000_rmethod2 =
    (k_xp2*yval[0]*exp(b_xp2*210))/(k_xp2 - yval[0] +
    yval[0]*exp(b_xp2*210))
```

```
    print(" ")
    print("The estimated population for Year 2000 using
        method 1 is: ",p2000_rmethod1, "millions" )
    print("The estimated population for Year 2000 using
        method 2 is: ",p2000_rmethod2, "millions" )


p_estimation(pop,xp1_function(pop),xp2_function(pop))
```

A.6 Question 1.9.7

```
def p_estimation(yval, x1, x2):
    xbar = sum(yval)/len(yval)
    xminus_xbar = []

    #for loop to create x-xbar data set
    for j in range (0,len(yval)):
        xminus = yval[j]-xbar
        xminus_xbar.append(xminus)

    #average (x'/x)
    avg_xp1 = sum(x1)/len(x1)
    avg_xp2 = sum(x2)/len(x2)

    xp1_minus_avg = []
    xp2_minus_avg = []

    #for loop to create (x'/x) - (x'/x)_bar
    for j in range (0,len(x1)):
        xp1bar = x1[j] - avg_xp1
        xp1_minus_avg.append(xp1bar)

    for j in range(0,len(x2)):
        xp2bar = x2[j] - avg_xp2
        xp2_minus_avg.append(xp2bar)

    num_xp1 = []
    num_xp2 = []

    #for loop to get the numerator of the sum of least
        squares for slope
    for j in range (0,len(xp1_minus_avg)):
        numxp1 = xminus_xbar[j]*xp1_minus_avg[j]
```

```
    num_xp1.append(numxp1)

for j in range(0,len(xp2_minus_avg)):
    numxp2 = xminus_xbar[j]*xp2_minus_avg[j]
    num_xp2.append(numxp2)

den_xminus_xbar = []

#for loop to get the denominator of the sum of least
    squares for the slope
for j in range (0,len(xminus_xbar)):
    den1 = math.pow(xminus_xbar[j],2)
    den_xminus_xbar.append(den1)

# solve for slope depending on the method used, in
    this case, the slope is (-r/k)
s_xp1 = sum(num_xp1)/sum(den_xminus_xbar)
s_xp2 = sum(num_xp2)/sum(den_xminus_xbar)

#solve for the intercept depending on the method, in
    this case, the intercept is r-value
b_xp1 =  avg_xp1- (s_xp1*xbar)
b_xp2 =  avg_xp2- (s_xp2*xbar)

print("The intercept(r-value) based for method 1 is:
    ",b_xp1)
print("The intercept(r-value) based for method 2 is:
    ",b_xp2)

k_xp1 = -(b_xp1/s_xp1)
k_xp2 = -(b_xp2/s_xp2)

print("The k-value based on method 1 is: ", k_xp1)
print("The k-value based on method 2 is: ", k_xp2)

print(" ")
p2000_rmethod1 =
    (k_xp1*yval[0]*exp(b_xp1*210))/(k_xp1 - yval[0] +
    yval[0]*exp(b_xp1*210))
p2000_rmethod2 =
    (k_xp2*yval[0]*exp(b_xp2*210))/(k_xp2 - yval[0] +
    yval[0]*exp(b_xp2*210))

print(" ")
print("The estimated population for Year 2000 using
    method 1 is: ",p2000_rmethod1, "millions" )
```

```
    print("The estimated population for Year 2000 using
      method 2 is: ",p2000_rmethod2, "millions" )

p_estimation(pop,xp1_function(pop),xp2_function(pop))
```