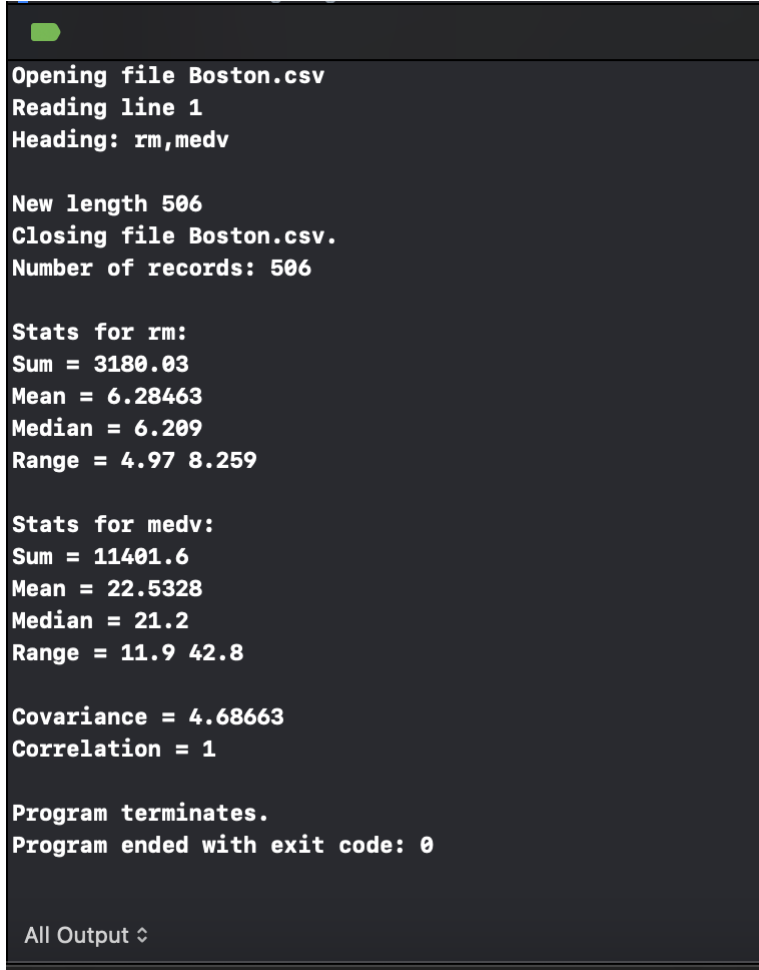


Sharmin Gaziani
CS 4375.004
February 4, 2023

Portfolio Component 1: Data Exploration

Runs of code:

A screenshot of a terminal window with a dark background and light green text. The output shows the execution of a C++ program that reads a CSV file named 'Boston.csv'. It reports the number of records (506) and calculates various statistics for two variables: 'rm' (average number of rooms) and 'medv' (median value). The statistics for 'rm' include sum, mean, median, and range. The statistics for 'medv' include sum, mean, median, and range. It also calculates the covariance and correlation between the two variables. The program ends with a message indicating it terminated successfully.

```
Opening file Boston.csv
Reading line 1
Heading: rm,medv

New length 506
Closing file Boston.csv.
Number of records: 506

Stats for rm:
Sum = 3180.03
Mean = 6.28463
Median = 6.209
Range = 4.97 8.259

Stats for medv:
Sum = 11401.6
Mean = 22.5328
Median = 21.2
Range = 11.9 42.8

Covariance = 4.68663
Correlation = 1

Program terminates.
Program ended with exit code: 0

All Output ↕
```

After writing this C++ program and having previous experience with using the built in R functions, I noticed that R is much easier to use. When it comes to importing the dataset as well as not having to write your own functions for calculations. It can be tedious at times to write out each formula you would like to use in C++ and during times where you need many calculations done all at once, R is much more efficient language for that. Especially if I want to see visuals of my data and don't have time to make my own charts, R does it all for me.

After writing out my own formulas, I have a much better understanding of different statistical measures, like mean for example which is another word for the average of a set of values. I was able to find the mean of rm and medv by taking the sum of all the values and dividing by the total, or numObservations. Median was a bit trickier to find because I had to sort the values from least to greatest to find the middle value in the data set. However, I was able to use a built-in sorting function in C++ which made it much easier to find the median. And for calculating the

range I was able to do this by subtracting the maximum value from the minimum value from the dataset. These statistical measures are important because it gives us an overview of what the “typical” data looks like. Often times data scientists are having to look at mass amounts of data all at once and its inefficient to view each entry of data when an overview will give all the information we need.

I was able to calculate my covariance by taking each entry of data for rm first and subtracting it from the rm mean I calculated earlier. I then did the same thing for medv and found the product of both these values and divided it by the numObservations - 1. I found when coding this in my C++ program, creating a for loop to do this task was much simpler. For the correlation, I used a similar formula where I took each entry of data from rm and medv and subtracted it from the mean of the following variables. I then found the product of the values, using a for loop as well, and divided it by the square root of both values to the second power. After finding the covariance and correlation, we are able to have a better understanding of how these two variables relate to one another. Covariance shows us whether or not the values move in the same direction and correlation shows us if change in one value will affect the other value, and if so by how much. These formulas allow us to better understand our data which is important for machine learning because many times we are working with large amounts of data and at times need to understand it quickly.