Data Exploration with C++

Bridgette Bryant

About data_exploration.cpp:

Purpose:

The purpose of the program was to review C++ and create a few simple functions. It processes a file named ‘Boston.csv’. It has 506 entries by default and only two columns, RM for average number of Rooms per Dwelling and MEDV for Median Value of Owner-Occupied Homes in $1000’s. It was creating on Windows using VS Code and MinGW-w64 g++ compiler. It displays the sum, mean, median, range, covariance, and correlation of the data. You run the program using the terminal ‘./data_exploration’ after building it.

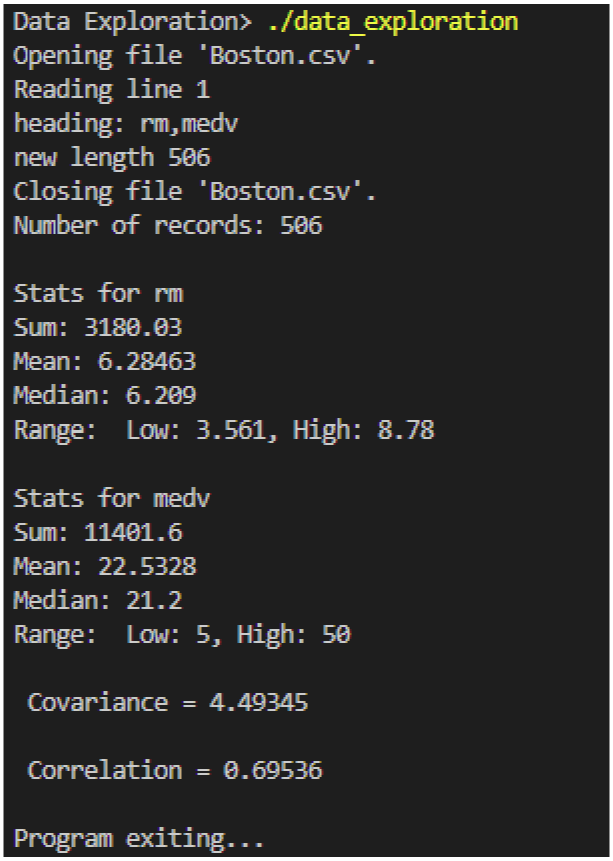
Input:

The Boston.csv, a data file with 2 columns the first is RM for average number of Rooms per Dwelling and MEDV for Median Value of Owner-Occupied Homes in $1000’s. It takes no other input, the spreadsheet may be adjusted, however, if changed layout/context wise it will make the program fail or become useless with its output.

rm,	medv
6.575,	24
6.421,	21.6
7.185,	34.7
6.998,	33.4
7.147,	36.2
6.43,	28.7
6.012,	22.9
6.172,	21.1
6.172,	27.1

Output:

When you run the program in the terminal the output will look like this:



```
Data Exploration> ./data_exploration
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:  Low: 3.561, High: 8.78

Stats for medv
Sum: 11401.6
Mean: 22.5328
Median: 21.2
Range:  Low: 5, High: 50

 Covariance = 4.49345

 Correlation = 0.69536

Program exiting...
```

Functions:

main: This function takes the basic arguments from the terminal but doesn't do anything with them. It uses an input file stream to read the 'Boston.csv' line by line, filling up two double type vectors for RM and MEDV. It verifies the file can be opened and will continue reading until the last line, if it cannot be opened the program will print an error message and return -1. Then it will close the file. It will tell the user when it is reading, the heading of the given file, the number of observations read, and when it is closing the file. It will then give the display shown above for the stats by calling the print_stats function for RM and MEDV. Then it will print the covariance with the covar function and print the correlation with the cor function. Then it will exit the program and return 0 on a successful end.

find_sum: This function takes a double type vector and adds all of the values together and then returns the sum as a double type.

find_mean: This function takes a double type vector and calls the find_sum function, divides it by two, and returns the mean as a double type.

find_median: This function takes a double type vector, then sorts it in increasing order, then it will take the middle element, subtracting 1 from the vector's size is odd, and returns the median as a double type.

find_range: This function takes a double type vector, it creates another double type vector with size 2. It will hold the minimum and maximum from the given vector. It finds the minimum and sets it as the first value of the range vector, then it finds the maximum and sets it as the second value of the range vector. The function then returns the range vector.

covar: This function takes the RM vector and MEDV vector, then calculates the covariance between the RM and MEDV data using the find_mean function. It then returns the calculated covariance as a double.

cor: This function takes the RM vector and MEDV vector, then calculates the correlation between the RM and MEDV data using the find_mean function. It then returns the calculated correlation as a double.

print_stats: This function takes a double type vector and prints the sum, using the find_sum function. Prints the mean, using the find_mean function. Prints the median, using the find_median function. Prints the range, low then high, using the find_range function. This function doesn't return anything.

My Experience using Built in functions in R vs. coding my own in C++

Writing my own functions in C++ was much more challenging and less functional compared to the built-in R functions. The R functions are very robust, probably faster, and very easy to use in the R terminal. My C++ program is more tedious to use as to use a different data set, you would have to change some of the code in the program to adjust to the new data set. It also doesn't give as much information as the build in R functions in terms of statistics such as p-value, etc. Overall, I think coding my own was a great C++ refresher and opportunity to in-depth learn the algorithm for the covariance and correlation calculations. However, the build in R functions are more practical and convenient to use.

Descriptive Statistical Measures

Mean

The mean gives an overall average value for the data set, it could be useful for knowing the general middle ground for values. For example, the average for a large data set of adult women of a given area can be compared to other averages of areas. This could be used to evaluate health of women in different areas, or help women compare themselves such as if they are significantly over or under weight.

Median

The median is very similar to the mean; however, it can be a little accurate because it isn't affected by outliers in the data. Because it is purely from the middle and not calculated it can possibly give you a more accurate way to compare data sets if there is a smaller range. However, if the range is large, it may be an inaccurate/limited representation of the data by only showing the most middle values.

Range

The range can be used to see how far apart the values are, this could be useful for seeing how much outliers affected the mean or how inaccurate the median is. It can also be looked at in comparison of other stats, for example the range for MEDV is much bigger than the range for RM, which could affect other statistics going forward and how correlated they are.

Covariance and Correlation statistics

Covariance is a measurement that shows how much one variable change are associated with how much the other variable changes. It is heavily affected by how much the numbers can range, and sometimes can be difficult to interpret. A positive covariance will show that both variables tend to be high or low at the same time. Whereas a negative covariance will show that when one variable tends to be high, the other tends to be low, vice versa. This is useful in machine learning because it shows the direction of the relationship between two variables. Correlation is a scaled version of covariance that ranges between -1 and 1, where the closer to 0 the value is, the less correlated the variables are. Whereas the closer to 1 shows they are very strongly positive correlated, the closer to -1 shows they are very strongly negatively correlated. The correlation value is very useful in machine learning because it gives us an easy-to-read measurement to compare how accurate models are.