Alper Duru
CS 4375.004

C++ Data Exploration

Introduction
This program is a C++ implementation of statistical analysis of data stored in a CSV file named "Boston.csv". It computes the sum, mean, median, range, covariance, and correlation of two sets of data, "rm" and "medv".

Main Function

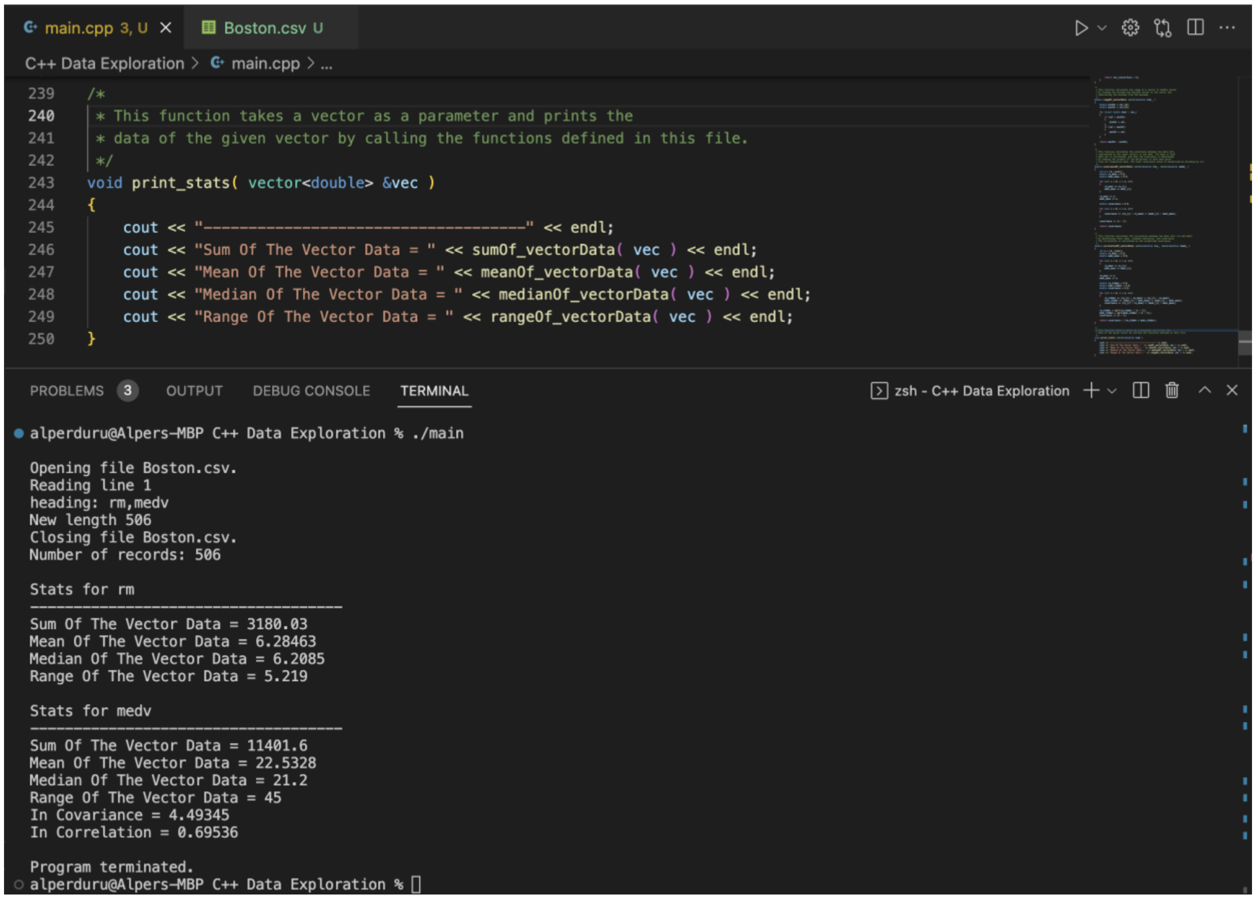
- Opens the "Boston.csv" data file
- Reads the data from the file and stores it in two vectors "rm" and "medv" to analyze the given data
- Computes the number of observations in the data
- Calls the **print_stats** function to display the statistical information for the two sets of data
- Computes the covariance and correlation of the two sets of data
- Closes the file and terminates the program.

Functions

1. **sumOf_vectorData**- computes the sum of a given vector of data
2. **meanOf_vectorData**- computes the mean of a given vector of data
3. **medianOf_vectorData** - computes the median of a given vector of data
4. **rangeOf_vectorData** - computes the range of a given vector of data
5. **covarianceOf_vectorData** - computes the covariance of two given vectors of data
6. **correlationOf_vectorData** - computes the correlation of two given vectors of data
7. **print_stats** - displays the sum, mean, median, and range of a given vector of data

Conclusion
The program reads data from a CSV file, performs statistical analysis on the data, and displays the results on the console. The program can be used to perform statistical analysis on different sets of data by changing the input file and the data contained within it.

Output of the written code:



main.cpp 3, U
Boston.csv U
C++ Data Exploration > main.cpp > ...
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
zsh - C++ Data Exploration

```cpp
/*
 * This function takes a vector as a parameter and prints the
 * data of the given vector by calling the functions defined in this file.
 */
void print_stats( vector<double> &vec )
{
    cout << "--------------------------------------" << endl;
    cout << "Sum Of The Vector Data = " << sumOf_vectorData( vec ) << endl;
    cout << "Mean Of The Vector Data = " << meanOf_vectorData( vec ) << endl;
    cout << "Median Of The Vector Data = " << medianOf_vectorData( vec ) << endl;
    cout << "Range Of The Vector Data = " << rangeOf_vectorData( vec ) << endl;
}
```

```
alperduru@Alpers-MBP C++ Data Exploration % ./main

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 Of The Vector Data = 3180.03
Mean Of The Vector Data = 6.28463
Median Of The Vector Data = 6.2085
Range Of The Vector Data = 5.219

Stats for medv
------------------------------------
Sum Of The Vector Data = 11401.6
Mean Of The Vector Data = 22.5328
Median Of The Vector Data = 21.2
Range Of The Vector Data = 45
In Covariance = 4.49345
In Correlation = 0.69536

Program terminated.
alperduru@Alpers-MBP C++ Data Exploration %
```

My experience using built-in functions in R versus coding my own functions in C++:

Both approaches have advantages and disadvantages when using built-in functions in R versus coding your own in C++. Using built-in functions in R is more convenient and faster as it provides a wide range of ready-made and tested functions for various statistical and data manipulation tasks. This saves the user time and effort.
However, coding your own functions in C++ gives you more control and customization of the process, which may be useful for your specific tasks and needs. Additionally, custom functions in C++ can reduce execution time, making them ideal for performance-critical applications.
Ultimately, the decision to use built-in functions in R or code your own in C++ depends on the task at hand, the level of control and performance you require, and your level of comfort in programming.

Descriptive statistical measures mean, median, and range, and how these values might be useful in data exploration prior to machine learning:

Mean, median, and range are important descriptive statistical measures used to understand the distribution of data. The mean is the mean value of a data set calculated by adding all the values and dividing by the number of values. The median is the median of the dataset and can be found by ordering the values and selecting the median. A range is the difference between the maximum and minimum values in the data set.

These measurements are very useful for data exploration prior to machine learning as they quickly provide insight into the overall distribution of the data.
The mean and median show the central trend of the data, while the range gives information about the spread of the data. You can use this information to identify outliers and skewness's in your data that can affect the results of your machine learning algorithms.
For example, if the data range is very large, this could indicate the presence of outliers that can affect the algorithm's performance. Understanding the descriptive statistics of your data can help you prepare your data and select features to improve the performance of your machine learning models.

Covariance and correlation statistics, and what information they give about two attributes. How might this information be useful in machine learning?

Covariance and correlation are important statistical measures of the relationship between two attributes. Covariance measures the extent to which two variables vary together and can be calculated as the expected value of the product of the deviations of each variable from its respective mean.
Correlation, on the other hand, measures the strength of the linear relationship between two variables and is expressed as a value between -1 and 1. A value of 1 indicates a perfectly positive linear relationship, a value of -1 indicates a perfectly negative linear relationship, and a value of 0 indicates no linear relationship.

Machine learning can use these statistics to understand relationships between different attributes and assist in feature selection.
For example, if two variables are highly correlated, you may not need to include both variables in your machine learning model because they both provide similar information.
Additionally, covariances and correlations can be used to identify non-linear relationships between variables. This can be important in creating accurate machine learning models. By understanding the relationships between various attributes, practitioners can make informed decisions about which features to include in their models and how to develop new features that improve performance.

Link to portfolio: https://alper-enes-duru.github.io/Machine_Learning_Portfolio/