2/9/23

Data Exploration Assignment

By-Sagar Darnal

Source Code:

```
#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <math.h>


using namespace std;                    // Use the standard namespace


vector<double> rm;                      // Declare vectors

vector<double> medv;


// Declare functions

double sum(vector<double> v) {

    double s = 0;

    for (int i = 0; i < v.size(); i++) {

        s += v[i];

    }

    return s;

}


double mean(vector<double> v) {

    return sum(v) / v.size();

}


double median(vector<double> v) {

    sort(v.begin(), v.end());

    int n = v.size();
    if (n % 2 == 0) {
        return (v[n / 2 - 1] + v[n / 2]) / 2;
    } else {
        return v[n / 2];
    }
}


double range(vector<double> v) {
    return *max_element(v.begin(), v.end()) - *min_element(v.begin(), v.end());
}
// Covariance and correlation by pg.74
double covariance(vector<double> x, vector<double> y) {
    int n = x.size();
    double mean_x = mean(x);
    double mean_y = mean(y);
    double cov = 0;
    for (int i = 0; i < n; i++) {
        cov += (x[i] - mean_x) * (y[i] - mean_y);
    }
    return cov / (n - 1);
}


double variance(vector<double> x, vector<double> y) {
    return covariance(x, y);
}


double correlation(vector<double> x, vector<double> y) {
    return covariance(x, y) / sqrt(variance(x, x) * variance(y, y));
}


// Main function
```

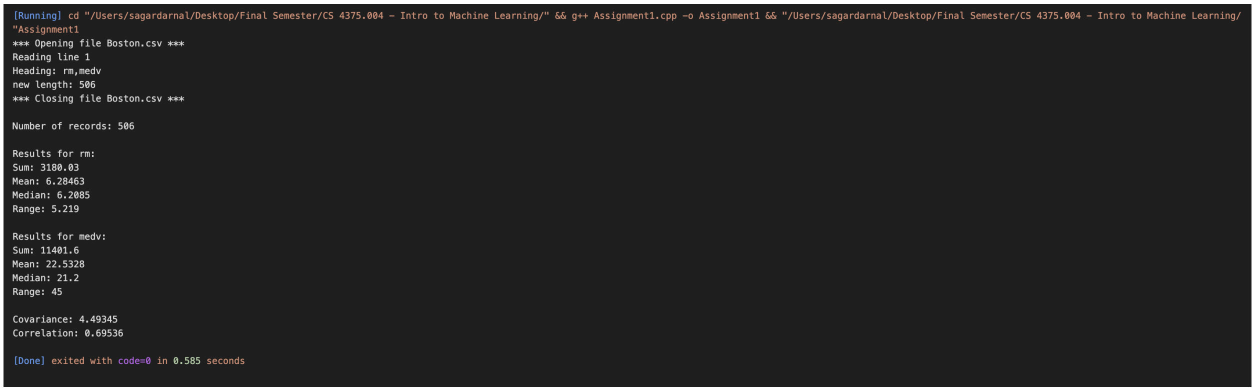
2/9/23

```
int main() {
    // Read data from file
    ifstream inFS;
    string line;
    string rm_in, medv_in;
    const int MAX_SIZE = 1000;
    vector<double> rm(MAX_SIZE);
    vector<double> medv(MAX_SIZE);

    cout << "*** Opening file Boston.csv ***" << endl;

    inFS.open("Boston.csv");
    if (!inFS.is_open()) {
        cout << "!!! Could not open file Boston.csv !!!" << endl;
        return 1;                              // 1 indicates error
    }

    cout << "Reading line 1" << endl;
    getline(inFS, line);

    cout << "Heading: " << line << endl;

    int numObservations = 0;
    while (inFS.good()){
        getline(inFS, rm_in, ',');
        getline(inFS, medv_in, '\n');

        rm[numObservations] = stod(rm_in);                    // Convert string to double
        medv[numObservations] = stod(medv_in);
        numObservations++;
    }
```

```
rm.resize(numObservations);
medv.resize(numObservations);

cout << "new length: " << rm.size() << endl;

cout << "*** Closing file Boston.csv ***" << endl;
inFS.close();

cout << "\nNumber of records: " << numObservations << endl;



// Call functions for rm vector
cout << "\nResults for rm:" << endl;
cout << "Sum: " << sum(rm) << endl;
cout << "Mean: " << mean(rm) << endl;
cout << "Median: " << median(rm) << endl;
cout << "Range: " << range(rm) << endl;

// Call functions for medv vector
cout << "\nResults for medv:" << endl;
cout << "Sum: " << sum(medv) << endl;
cout << "Mean: " << mean(medv) << endl;
cout << "Median: " << median(medv) << endl;
cout << "Range: " << range(medv) << endl;

// Call functions for covariance and correlation
cout << "\nCovariance: " << covariance(rm, medv) << endl;
cout << "Correlation: " << correlation(rm, medv) << endl;

return 0;
```

```
}
```

Output



```
[Running] cd "/Users/sagardarnal/Desktop/Final Semester/CS 4375.004 - Intro to Machine Learning/" && g++ Assignment1.cpp -o Assignment1 && "/Users/sagardarnal/Desktop/Final Semester/CS 4375.004 - Intro to Machine Learning/
"Assignment1
*** Opening file Boston.csv ***
Reading line 1
Heading: rm,medv
new length: 506
*** Closing file Boston.csv ***

Number of records: 506

Results for rm:
Sum: 3180.03
Mean: 6.28463
Median: 6.2085
Range: 5.219

Results for medv:
Sum: 11401.6
Mean: 22.5328
Median: 21.2
Range: 45

Covariance: 4.49345
Correlation: 0.69536

[Done] exited with code=0 in 0.585 seconds
```

b.

R is ideal for quick data analysis and exploration, while C++ is more suitable for performance-critical applications and for fine-tuning the implementation for specific use cases. As we did the data exploration in class, I found R allows a lot of functionality and easiness than C++. Meanwhile, C++ experience is not as bad for this project.

c.

The central tendency and distribution of a dataset are condensed into three statistical measures: mean, median, and range. The average of the data is called the mean, however outliers might affect it. The center number, or median, is unaffected by outliers. Range, which is the difference between the largest and lowest value, is a tool for locating outliers. These measurements are helpful in spotting potential problems with the data, such as outliers, skewness, or missing values, before machine learning. It is easier to get data ready for machine learning when you are aware of its features.

d.

Covariance and correlation are two statistics that describe the relationship between two variables or attributes in a dataset. Covariance is a measure of the strength of the linear relationship between the variables, with positive values indicating they increase together and negative values indicating they vary in opposite directions. Correlation normalizes the covariance to a value between -1 and 1, with -1 indicating a strong negative relationship, 0 indicating no relationship, and 1 indicating a strong positive relationship. This information is useful in machine learning as it can help identify which variables have a strong impact on the target variable, which can inform feature selection and the development of more effective models.