Mason Cushing

CS4375.004

Professor Mazidi

February 4th, 2023

https://github.com/mta825/Cushing_Portfolio_CS4375

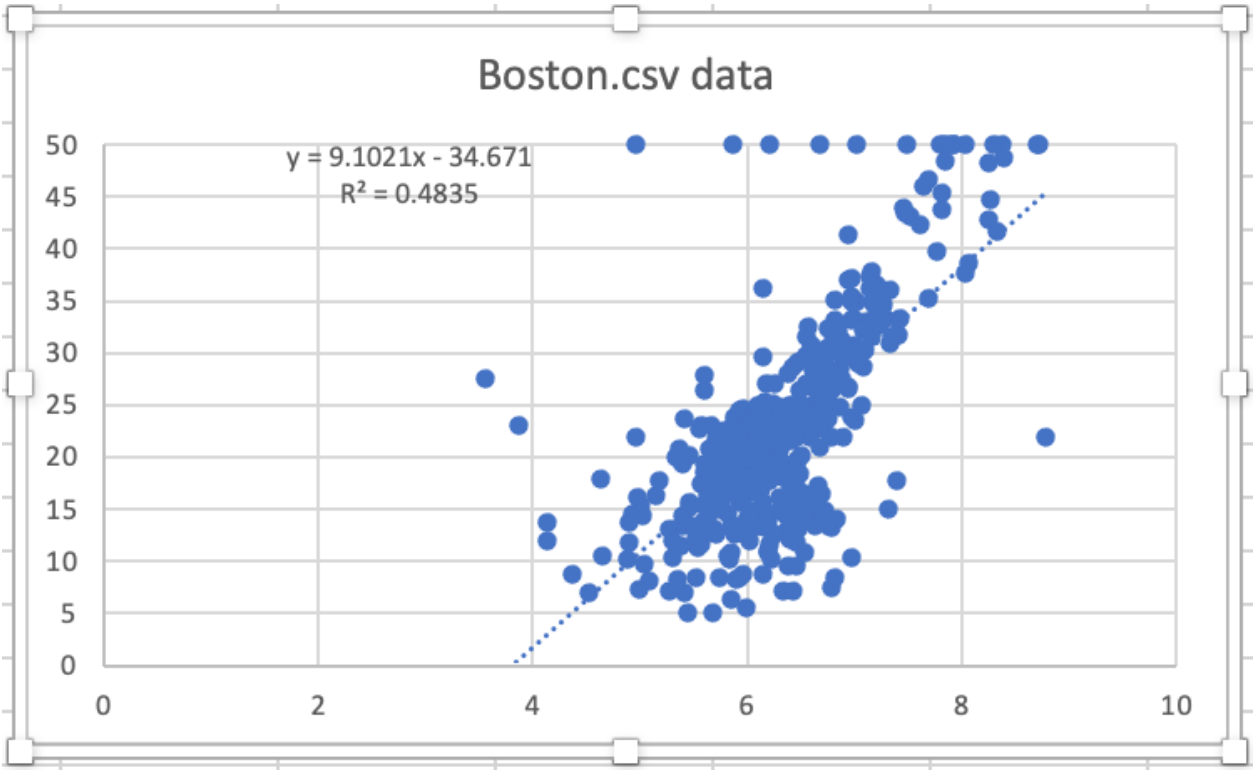
Data Exploration in C++

I am writing here about my data exploration assignment on the data given in "Boston.csv". Above is shown three example runs of the code (on the same data, so the results are the same), and a run of the built-ins in R working on the same data and performing the same function. Writing my own code and using the R built-ins was very different, especially as the C++ versions had to be tested thoroughly before using to create others (to avoid accumulating errors in the design process). Coding my own functions did enhance my understanding of how these statistics are calculated and give me a respect for their value.

The value of these statistics in the context of machine learning is immense. Mean, median, and range are summary statistics that are very easy for a person to understand and can be very valuable to the understanding of the data. In brief, the mean of a set of data is the sum divided by its length. For example, the mean of the following list (1,1,2,3,5,8) is 1+1+2+3+5+8 divided by 6, or 10/3, which is approximately 3.33. The median is the value at the halfway point of the data (so that half of the values are above it and half are below it). If the length is even, it is the average of the middle two values (so for our above example, the median would be 2.5). The range of the data is the difference between the largest value and the smallest value. For our example, the range would be 7. This tells us that the data doesn't ever have a larger difference than 7, and this information can be very helpful in analyzing the data and getting meaning from it

as a human being. In the context of machine learning, looking at these statistics before beginning an extensive, lengthy, and sometimes expensive algorithmic analysis of the data, some information can be determined from just these statistics, and sometimes looking for some correlation that isn't there can be avoided by looking at the statistics beforehand.

Covariance and correlation (grouped together because they are extremely similar in semantics and complexity) are telling statistics about how two sets of data are related. Shown below is the data I was given to explore for this assignment in a scatterplot with a dashed trendline and its equation. Notice on the equation label there is another value $R^2 = 0.4835$. This is an expression of correlation (the value I derived earlier with my code and R is the square root of this value). The closer this value is to 1 (or ±1 in the case of correlation as I derived it), the more closely related the values are. As this correlation is fairly low as far as these things are concerned, I would say that the values being measured likely do not correlate given this data set (with a few reasonable assumptions). Covariance is an unscaled value, while correlation is measured from -1 to 1, which makes it much easier to understand and immensely valuable to humans. In the context of machine learning, having this value is very helpful beforehand. For example, if you are looking for a correlation between two variables and are thinking of beginning a machine learning based analysis, looking at this value first can be very helpful for the analysis, and if it's too low in the first place, it tells you that you may be looking at the problem the wrong way. This concludes my exploration of this data and the corresponding assignment.



My Excel plot of the Boston.csv data. The rm vector is the x axis, and the medv vector is the y axis.

Shown here are screenshots of the runs of my C++ code (left) and my R output (right).

```
HEADING: rm,medv
SUMS: 3180.03 11401.6
MEANS: 6.28463 22.5328
MEDIANS: 6.2085 21.2
RANGES: 5.219 45
COVARIANCE: 4.49345
CORRELATION: 0.69536
```

```
HEADING: rm,medv
SUMS: 3180.03 11401.6
MEANS: 6.28463 22.5328
MEDIANS: 6.2085 21.2
RANGES: 5.219 45
COVARIANCE: 4.49345
CORRELATION: 0.69536
```

```
HEADING: rm,medv
SUMS: 3180.03 11401.6
MEANS: 6.28463 22.5328
MEDIANS: 6.2085 21.2
RANGES: 5.219 45
COVARIANCE: 4.49345
CORRELATION: 0.69536
```

```
> sum(Boston$rm)
[1] 3180.025
> sum(Boston$medv)
[1] 11401.6
> mean(Boston$rm)
[1] 6.284634
> mean(Boston$medv)
[1] 22.53281
> median(Boston$rm)
[1] 6.2085
> median(Boston$medv)
[1] 21.2
> range(Boston$rm)
[1] 3.561 8.780
> range(Boston$medv)
[1]  5 50
> cov(Boston$rm,Boston$medv)
[1] 4.493446
> cor(Boston$rm,Boston$medv)
[1] 0.6953599
```

Included here is my code that I reference in the description above. It is also posted at the portfolio listed in the header of this document.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;

//Calculate the sum of a vector
double mysum(vector<double> inv) {
   double summa = 0.0;
   for(int i=0; i<inv.size(); i++) {
      summa+=inv.at(i);
   }
   return summa;
}

//Calculate the mean of a vector
double mymean(vector<double> inv) {
   double summa = mysum(inv);
   double leng = (double) inv.size();
   return summa/leng;
}

//Calculate the median of a vector
double mymedian(vector<double> inv) {
   sort(inv.begin(),inv.end());
   int s = inv.size();
   int hs = s/2;
   if (s%2 == 0) {
      return ((double) (inv.at(hs-1)+inv.at(hs))) / 2.0;
   } else {
      return inv.at(hs);
   }
}

//Calculate the range of a vector
double myrange(vector<double> inv) {
   sort(inv.begin(),inv.end());
   int s = inv.size()-1;
   return inv.at(s) - inv.at(0);
}

//Calculate the standard deviation of a vector
double mysd(vector<double> inv) {
   double m = mymean(inv);
   double s1 = 0.0;
   double ss1;
   for (int i=0; i<inv.size(); i++) {
      ss1=(inv.at(i) - m);
      s1+=(ss1*ss1);
   }
   s1 = s1/( (double) (inv.size()-1) );
   s1 = sqrt(s1);
   return s1;
}

//Calculate the covariance of two vectors
double mycovariance(vector<double> inv1,vector<double> inv2) {
   double m1 = mymean(inv1);
   double m2 = mymean(inv2);
   double s1 = 0.0;
   double ss1;
```

```
    double ss2;
    int sz = inv1.size();
    for (int i=0; i<inv1.size(); i++) {
        ss1=(inv1.at(i) - m1);
        ss2=(inv2.at(i) - m2);
        s1+=(ss1*ss2);
    }
    return s1/((double) (inv1.size()-1));
}

//Calculate the correlation of two vectors (using functions from earlier)
double mycorrelation(vector<double> inv1,vector<double> inv2) {
    return mycovariance(inv1,inv2)/(mysd(inv1)*mysd(inv2));
}

int main(int argc, char *argv[]) {

    //Initializing reading variables
    ifstream f1;
    string line;
    string rm1, medv1;
    const int MAXLEN = 1000;
    vector<double> rm(MAXLEN);
    vector<double> medv(MAXLEN);

    //Attempting to open file
    f1.open("Boston.csv");
    if (!f1.is_open()) {
        cout << "Open failed" << endl;
        return 1;
    }

    //Read in the header
    getline(f1,line);
    cout << "HEADING: " << line << endl;

    //Read in the rest of the file and close it
    int numObservations = 0;
    while (f1.good()) {
        getline(f1,rm1,',');
        getline(f1,medv1,'\n');

        rm.at(numObservations) = stof(rm1);
        medv.at(numObservations) = stof(medv1);

        numObservations++;
    }
    rm.resize(numObservations);
    medv.resize(numObservations);
    f1.close();

    //Print the statistics using the above functions
    cout << "SUMS: " << mysum(rm) << " " << mysum(medv) << endl;
    cout << "MEANS: " << mymean(rm) << " " << mymean(medv) << endl;
    cout << "MEDIANS: " << mymedian(rm) << " " << mymedian(medv) << endl;
    cout << "RANGES: " << myrange(rm) << " " << myrange(medv) << endl;
    cout << "COVARIANCE: " << mycovariance(rm,medv) << endl;
    cout << "CORRELATION: " << mycorrelation(rm,medv) << endl;

}
```