Portfolio: C++ Data Exploration

Neo Zhao

CS4375 - Introduction to Machine Learning

- Code Input

```
#include <bits/stdc++.h>
#include <fstream>
#include <iostream>
#include <vector>
using namespace std;

// Sum
double sum(vector<double> v) {
  double total = 0;

  for (int i = 0; i < v.size(); i++) {
    total += v[i];
  }
  return total;
}

// Mean
double mean(vector<double> v) {
  return sum(v) / v.size();
}

// Median
double median(vector<double> v) {
  double median;

  sort(v.begin(), v.end());
  if (v.size() % 2 == 0) {
    median = (v[v.size() / 2] + v[v.size() / 2 - 1]) / 2;
  } else {
    median = v[v.size() / 2];
  }
  return median;
}

// Range
double range(vector<double> v) {
  sort(v.begin(), v.end());
  return v[v.size() - 1] - v[0];
}

// Covariance
double covar(vector<double> v1, vector<double> v2) {
  double sum = 0;
  double mean1 = mean(v1);
  double mean2 = mean(v2);

  for (int i = 0; i < v1.size(); i++) {
    sum = sum + (v1[i] - mean1) * (v2[i] - mean2);
  }
  return sum / (v1.size() - 1);
}

// Correlation
double corr(vector<double> v1, vector<double> v2) {
  double sumv1 = 0, sumv2 = 0;
  double sumTotal = 0;
  double sqSumv1 = 0, sqSumv2 = 0;
  int n = v1.size();

  for (int i = 0; i < n; i++) {
    sumv1 = sumv1 + v1[i];
    sumv2 = sumv2 + v2[i];
    sumTotal = sumTotal + v1[i] * v2[i];

    sqSumv1 = sqSumv1 + v1[i] * v1[i];
    sqSumv2 = sqSumv2 + v2[i] * v2[i];
  }

  double corr = (double)(n * sumTotal - sumv1 * sumv2) / sqrt((n * sqSumv1 - sumv1 * sumv1) *
                             (n * sqSumv2 - sumv2 * sumv2));
  return corr;
}

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

  ifstream inFS; // Input file stream
  string line;
  string rm_in, medv_in;
  const int MAX_LEN = 1000;
  vector<double> rm(MAX_LEN);
  vector<double> medv(MAX_LEN);

  // Try to open file
  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
  }

  // Can now use inFS stream like cin stream
  // Boston.csv should contain two doubles

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

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

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

    rm.at(numObservations) = stof(rm_in);
    medv.at(numObservations) = stof(medv_in);

    numObservations++;
  }

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

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

  cout << "Closing file Boston.csv." << endl;
  inFS.close(); // Done with file, so close it

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

  // rm Stats
  cout << "\nrm Statistics" << endl;
  cout << "Sum: " << sum(rm) << endl;
  cout << "Mean: " << mean(rm) << endl;
  cout << "Median: " << median(rm) << endl;
  cout << "Range: " << range(rm) << endl;

  // medv Stats
  cout << "\nmedv Statistics" << endl;
  cout << "Sum: " << sum(medv) << endl;
  cout << "Mean: " << mean(medv) << endl;
  cout << "Median: " << median(medv) << endl;
  cout << "Range: " << range(medv) << endl;

  // Covariance
  cout << "\nCovariance: " << covar(rm, medv) << endl;

  // Correlation
  cout << "\nCorrelation: " << corr(rm, medv) << endl;

  cout << "\nProgram terminated.";

  return 0;
}
```

- Code Output

```
Opening file Boston.csv.
Reading Line 1
Heading: rm,medv
New length: 506
Closing file Boston csv.

Number of records: 506
```

```
rm Statistics
Sum: 3180.03
Mean: 6.28463
Median: 6.2085
Range: 5.219

medv Statistics
Sum: 11401.6
Mean: 22.5328
Median: 21.2
Range: 45

Covariance: 4.49345

Correlation: 0.69536
```

- With built in R functions you don’t really know what goes into the process; however, with C++ you need to build the function as well as understand every component. We abstract away the complexity when using built in functions.
- The mean or the average is calculated by adding up all the values and dividing the sum by the total amount of values within a column or attribute. It is said the mean is the best, unbiased estimate of the population mean.
- The median is the middle most value within a sorted vector of values within a column or attribute. The median is resistant to outliers and is a good measure of center.
- The range is the largest value minus the smallest value within a column or attribute.
- The covariance is positive; unlike our correlation coefficient, we cannot determine the “strongness” from this value. Our Covariance result was 4.49345.
- The correlation coefficient measures the strongness of the linear relationship between X and Y. In this case, the correlation between **rm** and **medv**. Our Correlation result was 0.696737, which means we have a positive correlation as the value is > 0. We can conclude the correlation is strong.
-