jxz200021

Jiarui Groves

CS 4375.004

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <math.h>

using namespace std;

double sum (vector <double> v){

        double s = 0;

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

        {

            s+=v[i];

        }

        return s;

    }

double mean ( int size, double sum){

    double meanVal = sum / size;

    return meanVal;

    }

double median (vector <double> v){

    sort(v.begin(), v.end());

    int n = v.size();

    if(n %2 == 0)

    {

        return v[n/2];

    }

    else

    return (v[(n/2)-1]+v[n/2])/2;

}

 void range (vector <double> v, double\* min, double\* max){

    sort(v.begin(), v.end());

    \*min = v[0];

    \*max = v[v.size()-1];

}

double cov ( double mean1, double mean2, vector <double> v1, vector <double> v2){

        double sum, cov =0;

        for ( int i = 0; i< v1.size(); i++)

        {

            sum+= (v1[i] - mean1)\*(v2[i]- mean2);

        }

        cov = sum / (v1.size()-1);

        return cov;

}

double cor (double cov, vector <double> v1, vector <double> v2, double mean1, double mean2){

    double std1, std2,sum1,sum2;

    for ( int i = 0; i<v1.size();i++)

    {

        sum1+= pow((v1[i] - mean1),2);

    }

    std1 = sqrt(sum1/(v1.size()-1));

     for ( int i = 0; i<v2.size();i++)

    {

        sum2+= pow((v2[i] - mean2),2);

    }

    std2 = sqrt(sum2/(v2.size()-1));

    return cov/std1/std2;

}

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

{

    ifstream inFS;

    string line;

    string rm\_in,medv\_in;

    const int MAX\_LEN= 1000;

    vector <double> rm (MAX\_LEN);

    vector <double> medv(MAX\_LEN);

    double total\_rm,total\_medv, mean\_rm, mean\_medv,median\_rm, median\_medv;

    double min\_rm, min\_medv,max\_rm,max\_medv,covariance,correlatioln;

    cout<<"Openning file Boston,csv. "<<endl;

    inFS.open("Boston.csv");

    if(!inFS.is\_open()) {

        cout<<" Could not open file."<<endl;

        return 1;

    }

    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.at(numObservations) = stof(rm\_in); // take the input put into vector

        medv.at(numObservations) = stof(medv\_in);

        numObservations++;   //cout<<"breakpoint. "<<numObservations<<endl;

        if(numObservations == 506)

        {

            break;

        }

    }

   //while loop is working

    rm.resize(numObservations);

    medv.resize(numObservations);

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

    cout<<"Closing file Boston.csv. "<<endl;

    cout<<"Number of records: " <<numObservations<<endl;

    cout<<"\nStats for rm"<<endl;

    //print\_stats(rm);

    cout<<"\nStats for medv"<<endl;

    // print\_stats(medv);

    //cout<<"\n Covariance = "<< covar(rm,medv)<<endl;

    //cout<<"\n Correlation = " <<cor(rm,medv)<<end;

    cout<<"\nProgram terminaed.";

    //find sum

    total\_rm = sum(rm);

    cout<<" Sum of room is "<<total\_rm<<endl;

    total\_medv = sum(medv);

    cout<<"sum of median home value "<<total\_medv<<endl;

    // find mean

    mean\_rm = mean (rm.size(), total\_rm);

    cout<<" Mean of rm: "<<mean\_rm<<endl;

    mean\_medv = mean (medv.size(), total\_medv);

    cout<<" Mean of medv: "<<mean\_medv<<endl;

    //find median

    median\_rm = median(rm);

    cout<<" median of rm: "<<median\_rm<<endl;

    median\_medv = median(medv);

    cout<<" median of medv: "<<median\_medv<<endl;

//find range by passing two var

range(rm, &min\_rm, &max\_rm);

cout<<" The range of the vetor rm is "<<min\_rm<<" " <<max\_rm<<endl;

range(medv, &min\_medv, &max\_medv);

cout<<" The range of the vetor mdev is "<<min\_medv<<" " <<max\_medv<<endl;

//covariance

covariance = cov(mean\_rm, mean\_medv, rm, medv);

cout<<"Covariance is "<<covariance<<endl;

//correlation

cout<<"Correlation is "<<cor(covariance, rm, medv, mean\_rm,mean\_medv)<<endl;

    return 0;

}

Text

Description automatically generated

R built in function is way more efficient and convenient to use in order to get the statistical number from data such as vectors. In R, I can simply use command such as below Graphical user interface, application

Description automatically generated with medium confidence

Comment:

It takes many lines of code written in C++ in order to achieve the same output.

Mean, median and range of a dataset was mainly used to evaluate input as a standard.

It can give us an idea on how our data fall into the whole in a big picture.

Covariance tells us how change in one variable affact to the other, while correlation tells us how closely are the two variable related where the value apporaches to -1 or 1 indicates close relation.