Lab 7

STL Algorithms

In this workshop, you use the Standard Template Library's algorithms to evaluate sample data.

**LEARNING OUTCOMES**

Upon successful completion of this workshop, you will have demonstrated the abilities to

* copy data from a file into a sequential container
* accumulate data values using the STL's numeric library
* specify an operation on each value in a data set using a lambda expression
* sort the data values in a data set using the STL's algorithm

**INTRODUCTION TO STATISTICAL ANALYSIS**

Statistical analysis uses standard measures to make predictions based on a small sample of the actual data:

* sample mean - the average of all values in the sample
* sample standard deviation - the spread of the numbers away from their mean
* sample median - the middle number in the sorted set of the values (that is, the value separating the lower and upper halves of the data in a sorted set)

The formula for sample mean is

**zmean = ( Σizi ) / n**

The symbol **Σ** denotes 'sum of', **i** refers to an element in the set, and **n** refers to the number of elements in the set.

The formula for sample standard deviation (**ssd**) is

**ssd = √{ [ Σi(zi - zmean)2 ] / (n - 1) }**

Regression Line

A regression line relates a set of independent (x) values to a corresponding set of dependent (y) values.  The number of values in each set is the same in both sets.  Each value in the independent set has one corresponding value in the dependent set.

The regression line best fits the pairs of data values.  It is the line that passes through the data points drawn on a two-dimensional (x,y) system of coordinates as close as possible to the data points.  The line's coefficients are:

* **slope** - the slope of the line in the x-y plane
* **y\_intercept** - the y value of the line where it crosses the y-axis

The formulas for these two coefficients are:

**slope = [ n ( Σixiyi ) - Σixi Σiyi ] / [ n (Σixi2) - (Σixi)2 ]**

**y\_intercept = [ Σiyi - slope \* Σixi ] / n**

You can find a linear regression calculator [here](https://www.easycalculation.com/statistics/regression.php).

**SPECIFICATIONS**

Code a class template named **DataTable** for performing statistical analysis on data stored in text files.  This lab is to be completed using STL Algorithm calls. Failure to use STL algorithm function will result in loss of marks. Mean, sigma, median, and regression functions should not contain any manual loops.

Main Program

The **main()** program provided analyzes data using your **DataTable** template.

The first command-line argument is the name of the file to analyze.  The second argument, if present, requests a listing of all data values retrieved by the object.

Data Files

The data for this workshop is stored in the following three files:

* [***Simple.dat***](https://scs.senecac.on.ca/~btp305/pages/workshops/Simple.dat)***– TO BE SUBMITTED FOR MARKS***
* [**Flat.dat**](https://scs.senecac.on.ca/~btp305/pages/workshops/Flat.dat) - test 2
* [**HS\_College\_GPA.dat**](https://scs.senecac.on.ca/~btp305/pages/workshops/HS_College_GPA.dat) - high school and college GPA comparisons

Each record in each file contains two fields.  The first field holds the independent value (x coordinate).  The second field holds the dependent value (y coordinate).  For example, the file named **Simple.dat** contains

|  |
| --- |
| **2.1 8**  **2.5 12**  **4.0 14**  **3.6 10** |

DataTable Class Template

Upon instantiation, a **DataTable** object receives a reference to the file stream that holds the data values, the field width for displaying the data and the number of decimals to display.  The object retrieves the data values from the file and stores them in its instance variables.

Your design includes the following queries:

* **T mean() const** - returns the mean value of the dependent coordinate
* **T sigma() const** - returns the standard deviation of the dependent coordinates
* **T median() const** - returns the median value of the dependent coordinate
* **void regression(T& slope, T& y\_intercept) const** - returns the slope and intercept for the data set
* **void display(std::ostream&) const** - displays the data pairs as shown below

Your design also overloads the insertion operator as a helper for the **DataTable** class:

* **std::ostream& operator<<(std::ostream&, const DataTable&)** - inserts the data generated by **display()** into the output stream

Output

The results of your test runs on the data files should be similar to those listed below.

Simple.dat

Data Values

===========

x y

2.10 8.00

2.50 12.00

4.00 14.00

3.60 10.00

Statistics

==========

y mean = 11.00

y sigma = 2.58

y median = 12.00

slope = 1.91

intercept = 5.18

Flat.dat

Data Values

===========

x y

1.00 1.00

2.00 1.00

3.00 1.00

4.00 1.00

Statistics

==========

y mean = 1.00

y sigma = 0.00

y median = 1.00

slope = 0.00

intercept = 1.00

HS\_College\_GPA

Statistics

==========

y mean = 3.12

y sigma = 0.51

y median = 3.21

slope = 0.78

intercept = 0.73

Workshop 7

* Wrapper Class Templates
  + Reference\_wrapper: references to value in original wrapper, any change in original will affect this wrapper
  + Bind(&function call, 1st argument, 2nd argument), can be pass later using \_x, use when there is a fixed argument for most of the time
  + Use ref(var) function if you want to change the val in ref
* Functions
* Operator Class
* Algorithm
  + Count(begin(), end(), checkVar) 🡪 return how many times checkVar appear
  + Transform(begin(), end(), var.begin(), case) 🡪 do case with original and pass result to var, regular value remain unchanged