

Assignment #1

Output:

```
[Running] cd "c:\Users\amark\Desktop\C++ Programs\" && g++ data_exploration.cpp -o data_exploration && "c:\Users\amark\Desktop\C++ Programs\"data_exploration
data_exploration.cpp: In function 'void print_stats(std::vector<double>)':
data_exploration.cpp:13:33: error: 'rm_medv_sum' was not declared in this scope
    double sum = rm_medv_sum(vec);
                                ^
data_exploration.cpp:17:35: error: 'rm_medv_mean' was not declared in this scope
    double mean = rm_medv_mean(vec);
                                ^
data_exploration.cpp: In function 'double rm_medv_mean(std::vector<double>)':
data_exploration.cpp:38:17: error: 'sum' was not declared in this scope
    mean == sum / 507; // There are 507 values in the data.
                ^
data_exploration.cpp: At global scope:
data_exploration.cpp:44:24: error: 'double' is not a template
    double rm_medv_median (double <vector> vec)
                        ^~~~~~
data_exploration.cpp: In function 'double rm_medv_median(double)':
data_exploration.cpp:47:20: error: request for member 'size' in 'vec', which is of non-class type 'double'
    int size = vec.size();
                   ^
data_exploration.cpp:50:10: error: no matching function for call to 'sort()'
    sort()
    ^
In file included from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\algorithm:62:0,
                 from data_exploration.cpp:4:
c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\stl_algo.h:4697:5: note: candidate: template<class _RAIter> void std::sort(_RAIter, _RAIter)
    sort(_RandomAccessIterator __first, _RandomAccessIterator __last)
    ^
c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\stl_algo.h:4697:5: note: template argument deduction/substitution failed:
data_exploration.cpp:50:10: note: candidate expects 2 arguments, 0 provided
    sort()
    ^
In file included from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\algorithm:62:0,
                 from data_exploration.cpp:4:
c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\stl_algo.h:4727:5: note: candidate: template<class _RAIter, class _Compare> void std::sort(_RAIter, _RAIter, _Compare)
    sort(_RandomAccessIterator __first, _RandomAccessIterator __last,
    ^
c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\stl_algo.h:4727:5: note: template argument deduction/substitution failed:
data_exploration.cpp:50:10: note: candidate expects 3 arguments, 0 provided
    sort()
    ^
```

R vs. C++:

R and C++ have both provided me with coding knowledge that is unique to both languages as the former language's primary IDE, RStudio, is based off of C++. During my time in terms of attempting to code my own functions in C++, I realized that while pondering over algorithms to implement such functions can be beneficial for me as a Data Science major, I value the built-in functions that R has to offer slightly more as statistics is a key component for my major. In essence, I enjoy coding in both languages as they both have functions which provides organization for managing and utilizing a data set, for instance, that can be useful for data exploration and data analysis.

Statistical Measures:

In statistics, the mean is the average of a data set of consisting of values, the median is the middle number within a data set of values, ranging from least to greatest, and the range is the difference that is between the maximum number and minimum number. These measures are quite utilizable for data exploration since they can inform the users of what a data model might look like, the methods/how data could be clean and more organized, and the data representation of what the central tendency is, meaning that it is the number utilized for the middle of an applicable data set.

Covariance & Correlation Statistics:

Covariance is the measurement of how alterations with one variable is directly related to the alterations of another variable. Correlation is the measurement of how closely related two variables are to each other, meaning that the both of them would change, simultaneously. This is quite useful for machine learning as numerous algorithms have these implemented in them to eliminate the emptiness of potential values that are missing.

Reference:

<https://www.upgrad.com/blog/covariance-vs-correlation-everything-you-need-to-know/>