Homework #4

For all questions, please submit your R code, C++ code, and a doc file with a copy of the screen output and plots (where applicable) to [smtowers@asu.edu](mailto:smtowers@asu.edu). Due Wed April 3rd at 6pm.

The code in your R and C++ files should exhibit all of the good coding practices mentioned in <http://sherrytowers.com/2012/12/14/good-programming-practices-in-any-language/>

All plots should exhibit all of the good practices in creating figures, mentioned in

<http://sherrytowers.com/2013/01/04/good-practices-in-producing-plots/>

Please submit your files in a format hwk4\_<first name>\_<initial of last name>.R , hwk4\_<first name>\_<initial of last name>.cpp and .h, and hwk4\_<first name>\_<initial of last name>.doc

Question 1:

As discovered in class, it turns out there are bugs in the weighted variance calculation currently implemented in the UsefulUtils::WeightedVar() method. Correct this using the following steps:

1. In the program example\_useful\_utils.cpp, the WeightedVar method is used to calculate the variance of myvector weighted with myweight. In R, fill vectors myvector and myweight, in concordance with those filled in example\_useful\_utils.cpp.
2. To calculate the true variance of myvector weighted by myweight, use the R method cov.wt(as.matrix(myvector),myweight) This will return a 1x1 covariance matrix, which contains the weighted covariance of myvector). What value is returned? Note that the cov.wt method in R is bug free, and this is the correct value.
3. Now, using the formulae provided here: <http://en.wikipedia.org/wiki/Weighted_mean#Weighted_sample_variance> fix the C++ code in UsefulUtils.cpp such that the WeightedVar method correctly calculates the weighted variance. Use the formula for s^2 on that page, not the formula for sigma^2.

In past classes we discussed operations in C++ on STL vectors, but if you need to refresh yourself, you can find information on C++ vector operation methods like transform and accumulate here <http://www.cplusplus.com/reference/algorithm/transform/>

and here <http://www.cplusplus.com/reference/numeric/accumulate/>

(along with lots of other places on the internet).

1. Write a function in R called weighted.var that implements the same algorithm.

Question 2:

The UsefulUtils class is currently missing the code to calculate the mean and variance of a vector in the Mean() and Variance() methods, respectively. Fill in this code, and write a simple C++ program (similar to example\_useful\_utils.cpp) that will allow you to test these methods compared to the output of mean() and var() in R.

Question 3:

The UsefulUtils class is currently missing the code to calculate the covariance and correlation between two vectors. Fill in this code, and write a simple C++ program that will allow you to test these methods compared to the output of cov() and cor() in R.

Question 4

A) Write the R code to read in the data in www.sherrytowers.com/hwk4\_data.txt

The file contains 100 different simulations, where the data are simulated at x=0,1,2,…,24 with the function y\_obs = A\*x+B+epsilon, where epsilon is a random number drawn from some probability distribution that is unknown to you, and A=0 and B=100. A and B are the same for all the samples, but the underlying probability distributions used for the random smearing are different.

B) Explain why the Pearson chi-squared method is an appropriate method for fitting these data sets.

C) For each of the data sets, assume that the data are linearly distributed with y\_pred = A\*x+B.

Calculate the Pearson chi-squared statistic comparing the data to the model for many different hypotheses of A and B; do this by uniformly randomly sampling many potential values of B, and note that you can obtain A for each sample and B hypothesis by requiring that sum(y\_pred)=sum(y\_obs), and calculating the A that achieves this.

Justify the number of randomly sampled hypotheses of B you examine, and the range you consider.

For each sample, then estimate the best fit values of A and B, mu\_A and mu\_B, and estimate the uncertainty on A and B, delta\_A and delta\_B, using the methods:

1. Estimate the best fit A and B from the values of the parameter hypothesis at the minimum value of the Pearson chi-squared statistic, Tmin. Estimate the one standard deviation parameter uncertainties from the range of values that result in a Pearson chi-squared statistic that are at most one from the best fit minimum value Tmin
2. For each parameter hypothesis, calculate the weight dnorm(sqrt(T-Tmin)). Estimate the best fit A and B by taking mean of the parameter hypotheses weighted by these weights (you can use the weighted.mean function in R). Estimate the one-standard deviation uncertainties by taking the sqrt of the weighted variance of the A and B hypotheses. Use the R function you wrote in Question 1d). For each sample, estimate the correlation between the A and B parameter estimates, and comment on the results.

Calculate the mean and standard deviation of N\_A = (mu\_A-0)/delta\_A and N\_B=(mu\_B-100)/delta\_B for each of the two methods. What should the means and standard deviations of N\_A and N\_B be if the data are not over- or under- dispersed? Test whether or not your results significantly deviate from the expected mean and standard deviation under the null hypothesis that the data are not over- or under-dispersed.

Histogram N\_A and N\_B for each of the two methods. Your results should look something like this:



D) Test the three data sets for over- or under- dispersion when compared to the best fit linear model. Which samples, if any, appear to be significantly over-dispersed when compared to the linear model? Which, if any, appear to be significantly under-dispersed when compared to the linear model?

E) For the data samples that are significantly over- or under-dispersed, correct the Pearson chi-squared statistics for the dispersion. Then repeat all the parameter estimation steps in C for all of the data samples (both corrected for dispersion, and the uncorrected samples). Calculate the mean and standard deviation of N\_A = (mu\_A-0)/delta\_A and N\_B=(mu\_B-100)/delta\_B for each of the two methods, and test whether or not they are consistent with expected. Does the dispersion correction method applied to the Pearson chi-squared appear to work? Histogram N\_A and N\_B for each of the two methods. Your results should look something like this:

