William O’Mullan

ECE 3522: Stochastic Processing

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

For this assignment we evaluated basic statistics for two sets of time series data using MATLAB. The first data set was Google’s closing stock prices (NASDAQ abbreviation: GOOG) for each day from its inception on August 19th 2004 to January 9th 2015. The second data set was the same speech recording ‘rec\_01\_speech.raw’ we used in recitations for our Signals: Continuous and Discrete course. For each data set we first found the global statistics for the minimum value, the maximum value, the mean, the median, and the variance. We then evaluated the mean and the variance for each data set using a frame-window analysis method with different frame and window durations for each data set. We evaluated the Google stock prices for this part with window durations (N) of seven and thirty days and frame durations (M) of one, seven, fourteen, and thirty days. The sampled audio file was evaluated with window durations (N) of twenty and thirty milliseconds and frame durations (M) of five, ten, and twenty milliseconds; converted to numbers of samples using the 8 kHz sampling frequency.

# Approach and Results

I split my evaluation into two sections in MATLAB with each section dedicated to a particular data set. For the first data set, I used the MATLAB function ‘xlsread’ to split the Microsoft Excel file that the Google stock prices were on into it’s numeric, textual, and raw data. Using a ‘for’ loop, I separated the closing prices from the opening, high, and low prices. I then plotted the closing prices as a line graph with respect to the days since Google stock inception. MATLAB includes functions to find the minimum, maximum, mean, median, and variance for any time series set of data so finding the global values for those statistics was as simple as feeding the entire set of closing prices into those functions. The next task was more complicated, we needed to find the mean and variance over a window of data that moved a certain number, or frame, of samples each loop. To do this I set up a series of nested ‘for’ loops that looped through the different values for window and frame duration as well as cycling the window through the data set according to the frame duration. This was to separate data points from the whole set to evaluate and store in arrays for the means and variances. These were dynamically plotted for each different window and frame duration. The second speech signal data set was evaluated fairly similarly to the first data set. There were differences in how the file was opened and closed as well as the fact that I found the root mean square for the data rather than just the mean since the data set contained negative values which would throw off meaningful evaluation.

The results I gained were linked to how mean and variance are linked to concepts we learned in our Signals: Continuous and Discrete course. The mean is defined to be the sum of all the values in a data set divided by the number of values contained in the data set itself:

The mean tells us the average or central value of the data set. The root mean square differs from the mean in that it tells us the magnitude of a set of varying values rather than just the central value. We used the RMS in Signals and Circuits to find the DC signal that would deliver the same average power as the AC signal we were evaluating. In mathematical terms:

The variance is the mean’s accompanying value, it states how far from the average value any number in the data set is expected to be. Noted below, the equation to find variance is:

This is interesting because for a mean of zero the variance becomes the RMS which shows one of the ways in which statistics has been used in signal processing applications.

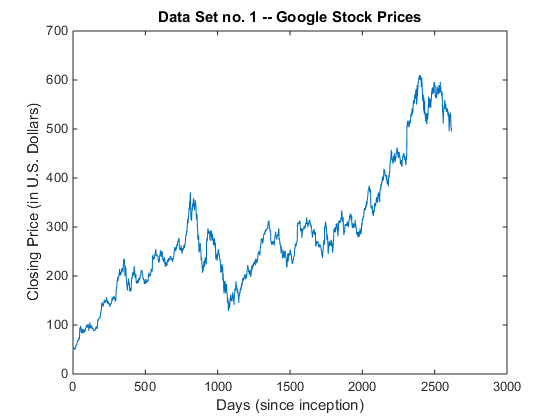
# MATLAB Code

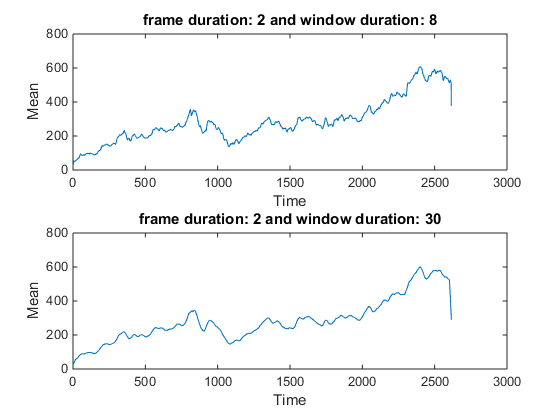
% Computer Assignment no. 1  
% Stochastic Processing  
% Bill O'Mullan

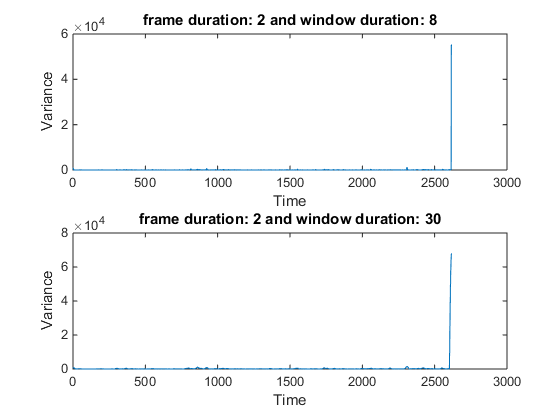
## Data Set no. 1 -- Google Stock Prices

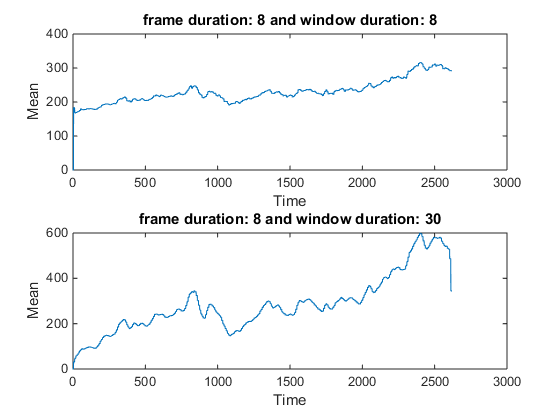
clear;close all;clc;  
  
% load .xlsx file -- full sheet = raw, text = txt,  
% numeric data = stock\_prices  
[stock\_prices,txt,raw] = xlsread('google\_v00.xlsx');  
  
% just evaluate the closing prices for each day  
closing\_prices = zeros(length(stock\_prices),1);  
for a = 1:length(stock\_prices)  
 closing\_prices(a,1) = stock\_prices(a,4);  
end  
  
% generate x axis of line graph for closing price data  
days = 1:length(stock\_prices);  
  
% plot closing prices with respect to days since Google stock inception  
figure(9);clf;  
plot(days,closing\_prices);  
title('Data Set no. 1 -- Google Stock Prices');  
xlabel('Days (since inception)');  
ylabel('Closing Price (in U.S. Dollars)');  
  
% find the global statistics of minimum, maximum, mean, median, and  
% variance for the data  
global\_min\_01 = min(closing\_prices);  
str = sprintf('global minimum: %d',global\_min\_01);  
disp(str);  
global\_max\_01 = max(closing\_prices);  
str = sprintf('global maximum: %d',global\_max\_01);  
disp(str);  
global\_mean\_01 = mean(closing\_prices);  
str = sprintf('global mean: %d',global\_mean\_01);  
disp(str);  
global\_median\_01 = median(closing\_prices);  
str = sprintf('global median: %d',global\_median\_01);  
disp(str);  
global\_variance\_01 = var(closing\_prices);  
str = sprintf('global variance: %d',global\_variance\_01);  
disp(str);  
  
% find the mean and variance over different window and frame durations for  
% the data set  
  
% set up window (N) and frame (M) duration (in days)  
M = [2, 8, 14, 30];  
N = [8, 30];  
  
% loop through the window and frame durations  
for m = 1:length(M)  
 for n = 1:length(N)  
  
 % assign window and frame duration to separate variables  
 window\_dur = N(n);  
 frame\_dur = M(m);  
  
 % determine the number of prices, the number of frames, and set up  
 % answer arrays for population  
 tot\_prices = length(closing\_prices);  
 tot\_frames = round(tot\_prices/frame\_dur);  
 prices\_mean = zeros(tot\_prices,1);  
 prices\_variance = zeros(tot\_prices,1);  
  
 % loop through the prices using specific window and frame sizes  
 for a = 1:tot\_frames  
  
 % determine the size of the window about the center point of  
 % the frame  
 center\_pt = ((a-1)\*frame\_dur) + (frame\_dur/2);  
 left\_bound = center\_pt - (window\_dur/2);  
 right\_bound = center\_pt + (window\_dur/2);  
  
 % zero stuff the variable for mean and variance calculation in  
 % case it exceeds the size of the input data set  
 if ((left\_bound < 0) || (right\_bound > tot\_prices))  
 input\_prices = zeros(window\_dur,1);  
 end  
  
 % transfer closing prices data in window to input\_prices  
 % variable for calculations  
 for b = 1:window\_dur  
 index = left\_bound + (b-1);  
 if((index > 0) && (index <= tot\_prices))  
 input\_prices(b,1) = closing\_prices(index,1);  
 end  
 end  
  
 % perform computations on input\_prices  
 mean\_input = mean(input\_prices);  
 variance\_input = var(input\_prices);  
  
 % populate total answer array with input\_prices computations  
 for c = 1:frame\_dur  
 index = left\_bound + (c - 1) + (window\_dur/2);  
 if((index > 0) && (index <= tot\_prices))  
 prices\_mean(index,1) = mean\_input;  
 prices\_variance(index,1) = variance\_input;  
 end  
 end  
  
 end  
  
 % plot the mean for each M and N  
 figure(2\*m);  
 str\_01 = sprintf('frame duration: %d and window duration: %d', M(m), N(n));  
 subplot(length(N),1,n);  
 plot(prices\_mean);  
 title(str\_01);  
 xlabel('Time');  
 ylabel('Mean');  
  
 % plot the variance for each M and N  
 figure((2\*m)-1);  
 str\_02 = sprintf('frame duration: %d and window duration: %d', M(m), N(n));  
 subplot(length(N),1,n);  
 plot(prices\_variance);  
 title(str\_02);  
 xlabel('Time');  
 ylabel('Variance');  
  
 end  
end

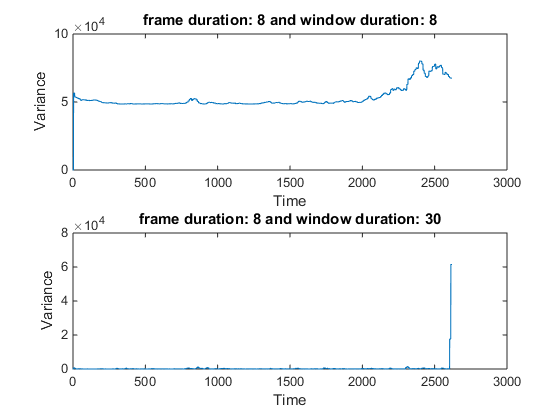
global minimum: 4.995000e+01  
global maximum: 6.094700e+02  
global mean: 2.867374e+02  
global median: 2.648250e+02  
global variance: 1.619447e+04

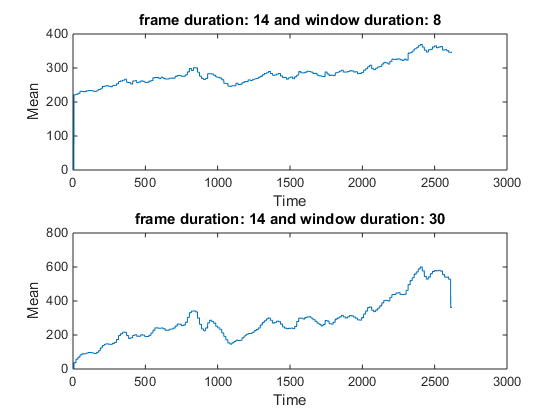


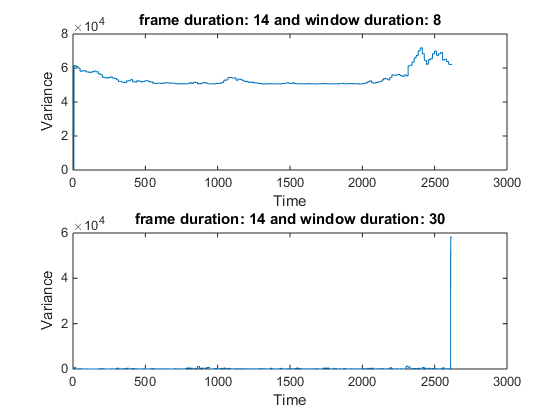


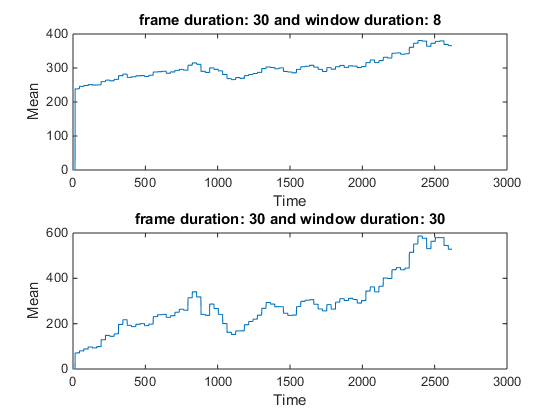


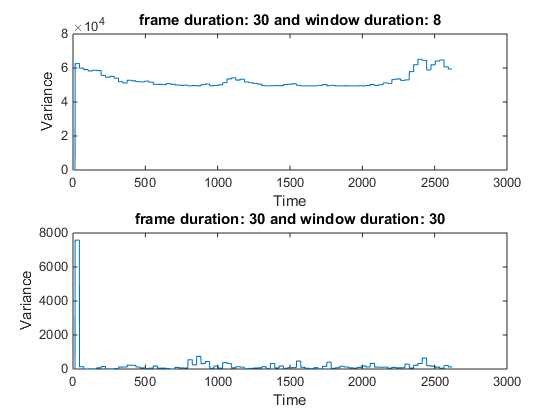








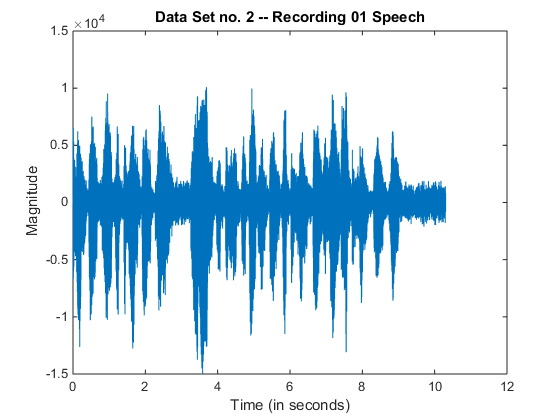


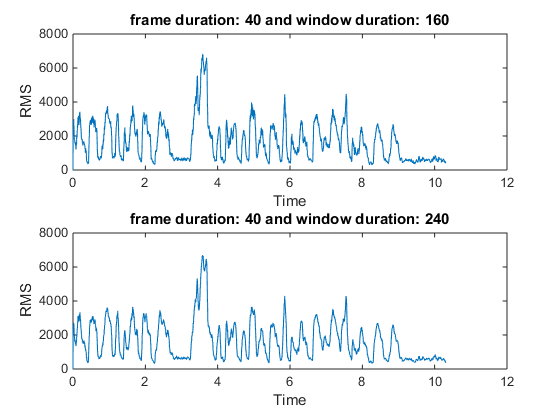


## Data Set no. 2 -- 'rec\_01\_speech.raw'

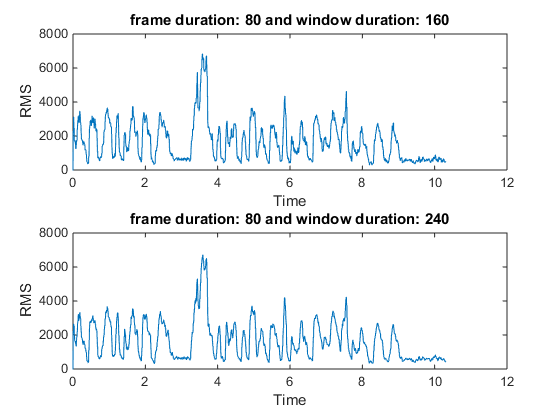
clear;close all;clc;  
  
% open .raw file, load binary data as 16 bit integers, then close the  
% file to avoid corruption  
file = fopen('rec\_01\_speech.raw','r');  
speech\_data = fread(file,inf,'int16');  
fclose(file);  
  
% generate x axis of graph for speech data, correcting for  
% 8 kHz F\_sampling  
Fs = 8000;  
width = 0:length(speech\_data)-1;  
time = zeros(length(width),1);  
for a = 1:length(speech\_data)  
 time(a,1) = width(1,a);  
end  
time = time/Fs;  
  
% plot the speech signal  
figure(7);clf;  
plot(time,speech\_data);  
title('Data Set no. 2 -- Recording 01 Speech');  
xlabel('Time (in seconds)');  
ylabel('Magnitude');  
  
% find the global statistics of minimum, maximum, mean, median, and  
% variance for the data  
global\_min\_02 = min(speech\_data);  
str = sprintf('global minimum: %d',global\_min\_02);  
disp(str);  
global\_max\_02 = max(speech\_data);  
str = sprintf('global maximum: %d',global\_max\_02);  
disp(str);  
global\_rms\_02 = rms(speech\_data);  
str = sprintf('global RMS: %d',global\_rms\_02);  
disp(str);  
global\_median\_02 = median(speech\_data);  
str = sprintf('global median: %d',global\_median\_02);  
disp(str);  
global\_variance\_02 = var(speech\_data);  
str = sprintf('global variance: %d',global\_variance\_02);  
disp(str);  
  
% find the mean and variance over different window and frame durations for  
% the data set  
  
% set up window (N) and frame (M) duration (convert to msec from 8 kHz Fs)  
M = [40, 80, 160];  
N = [160, 240];  
  
% loop through the window and frame durations  
for m = 1:length(M)  
 for n = 1:length(N)  
  
 % assign window and frame duration to separate variables  
 window\_dur = N(n);  
 frame\_dur = M(m);  
  
 % determine the number of prices, the number of frames, and set up  
 % answer arrays for population  
 tot\_samples = length(speech\_data);  
 tot\_frames = round(tot\_samples/frame\_dur);  
 speech\_rms = zeros(tot\_samples,1);  
 speech\_variance = zeros(tot\_samples,1);  
  
 % loop through the prices using specific window and frame sizes  
 for a = 1:tot\_frames  
  
 % determine the size of the window about the center point of  
 % the frame  
 center\_pt = ((a-1)\*frame\_dur) + (frame\_dur/2);  
 left\_bound = center\_pt - (window\_dur/2);  
 right\_bound = center\_pt + (window\_dur/2);  
  
 % zero stuff the variable for mean and variance calculation in  
 % case it exceeds the size of the input data set  
 if ((left\_bound < 0) || (right\_bound > tot\_samples))  
 input\_samples = zeros(window\_dur,1);  
 end  
  
 % transfer closing prices data in window to input\_prices  
 % variable for calculations  
 for b = 1:window\_dur  
 index = left\_bound + (b-1);  
 if((index > 0) && (index <= tot\_samples))  
 input\_samples(b,1) = speech\_data(index,1);  
 end  
 end  
  
 % perform computations on input\_prices  
 rms\_input = rms(input\_samples);  
 variance\_input = var(input\_samples);  
  
 % populate total answer array with input\_prices computations  
 for c = 1:frame\_dur  
 index = left\_bound + (c - 1) + (window\_dur/2);  
 if((index > 0) && (index <= tot\_samples))  
 speech\_rms(index,1) = rms\_input;  
 speech\_variance(index,1) = variance\_input;  
 end  
 end  
  
 end  
  
 % plot the mean for each M and N  
 figure(2\*m);  
 str\_01 = sprintf('frame duration: %d and window duration: %d', M(m), N(n));  
 subplot(length(N),1,n);  
 plot(time,speech\_rms);  
 title(str\_01);  
 xlabel('Time');  
 ylabel('RMS');  
  
 % plot the variance for each M and N  
 figure((2\*m)-1);  
 str\_02 = sprintf('frame duration: %d and window duration: %d', M(m), N(n));  
 subplot(length(N),1,n);  
 plot(time,speech\_variance);  
 title(str\_02);  
 xlabel('Time');  
 ylabel('Variance');  
  
 end  
end

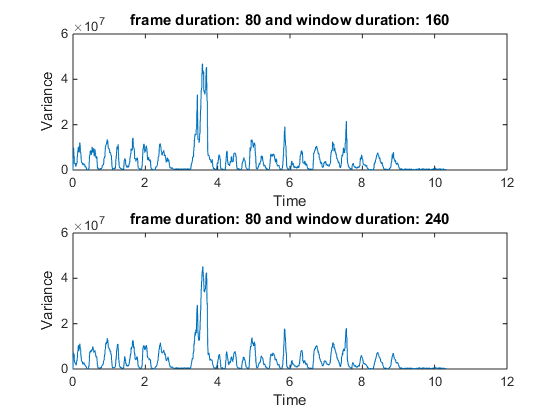
global minimum: -14993  
global maximum: 10104  
global RMS: 2.034530e+03  
global median: 83  
global variance: 4.139363e+06

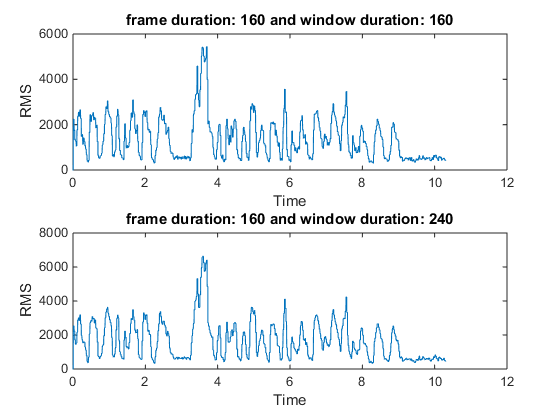


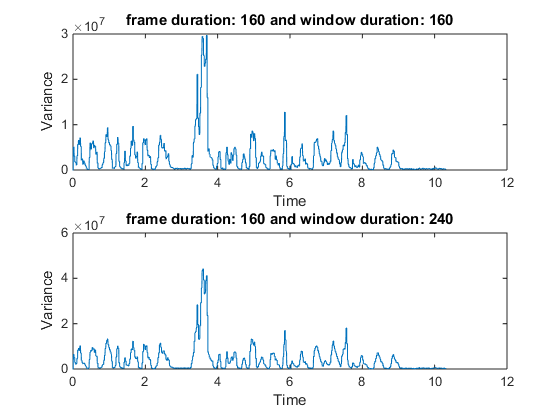












[*Published with MATLAB® R2014b*](http://www.mathworks.com/products/matlab)

# Conclusions

Taking into account some sharp peaks in the variance graphs caused by zero stuffing the window extending past the range of the data set, you can see how the connections we drew between statistics and signal processing hold true. The variance is often zero for the Google closing prices which reflects how close the prices are to each other in the window we choose. There are times where the variance is substantial but this happens for window and frame durations where there are gaps in our evaluation of the data set or the overlap between windows is extremely small. This is exemplary of how faulty window and frame durations in our analysis can skew results and how we need to use our common sense and engineering intuition when utilizing computer programs such as MATLAB when solving problems.

The second conclusion we drew relating variance and the RMS is extremely clear in the evaluation of the speech signal. Both the variance and RMS line graphs mirror each other in shape, even if they differ substantially in magnitude. I hypothesize that this is the influence of the mean on the variance calculations, at many points in the speech signal the magnitude of the data points is negative which can result in a negative mean. When input to the variance calculation any data point will have the negative mean subtracted from it which really results in addition since the negatives cancel out. Exacerbated by squaring the quantity, the magnitudes can quickly rise quite high whilst still maintaining the same general behaviour as the RMS.