Devin Trejo
ECE 3522: Stochastic Processes in Signals and Systems
Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

## I. PROBLEM STATEMENT

We want to learn the basics of what mean and variance represent for a given dataset. In this, case we will look at an arbitrary speech signal and Google's stock data for the past eleven years (2616 days). First we calculate the mean/variance/min/max/median for the entire set of data values which we then use to compare to the mean/variance of smaller sections of data. We explore how to interpreting the mean/variance values for smaller window sizes allows us to see different overall characteristics. We also learn some basics on reading in data from external files into MatLab to utilize MatLab's built in statistical functions.

## II. APPROACH AND RESULTS

To begin we load our data into MatLab. To better understand what is happening throughout the data we plot the signals.

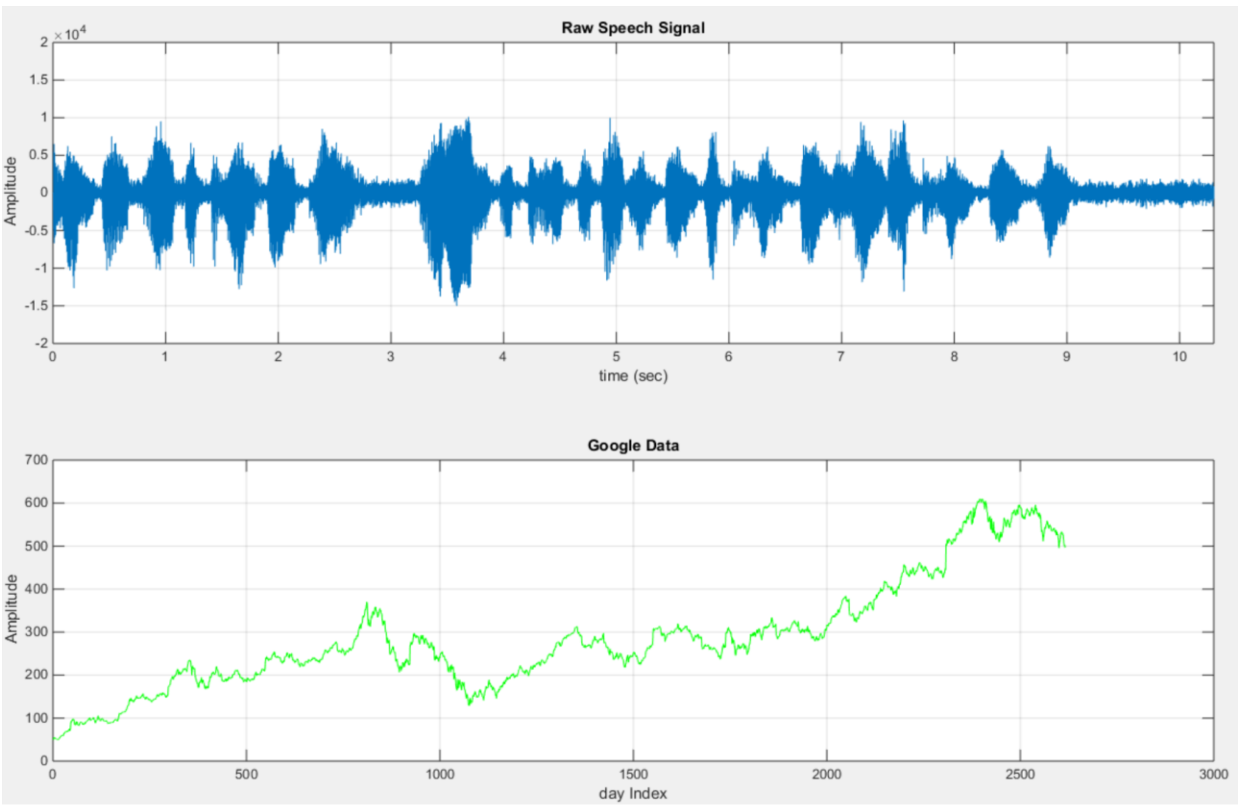


Figure 1: Graphical representation of the speech/Google data

Using MatLab's built in function we find the min/max/mean/median/variance for speech and Google stock data.

```
Google data: min = 49.950000, max = 609.470000, mean = 286.737370, median = 264.825000, variance = 16194.466872

Speech data: min = -14993.000000, max = 10104.000000, mean = -0.389117, median = 83.000000, variance = 4139362.775440
```

Comparing the speech signal to the google data we see different characteristics between the values. The mean value centers around zero because we have an AC signal with zero DC offset. Therefore, we expect the mean of the signal to be zero. Variance is higher for the speech signal because of radically changing sine wave amplitudes that compose the speech signal. The Google stock price does not have a mean of zero since their stock price does not oscillate around zero like the speech signal. Instead, the Google stock mean shows the average price of the stock over the past eleven years. Recalling that the variance is the standard deviation of a set of data squared, we can take the square root of our variance to see a standard deviation of 127 points. Comparing the standard deviation with the mean we see that most of our data falls between one standard deviation from the mean.

Now we will investigate what the mean/variance of the data looks like when we break them up into varying window/frame sizes. First we focus on the speech signal. We break the speech frame sizes of 40, 60, 169 and window sizes of 160 and 240.

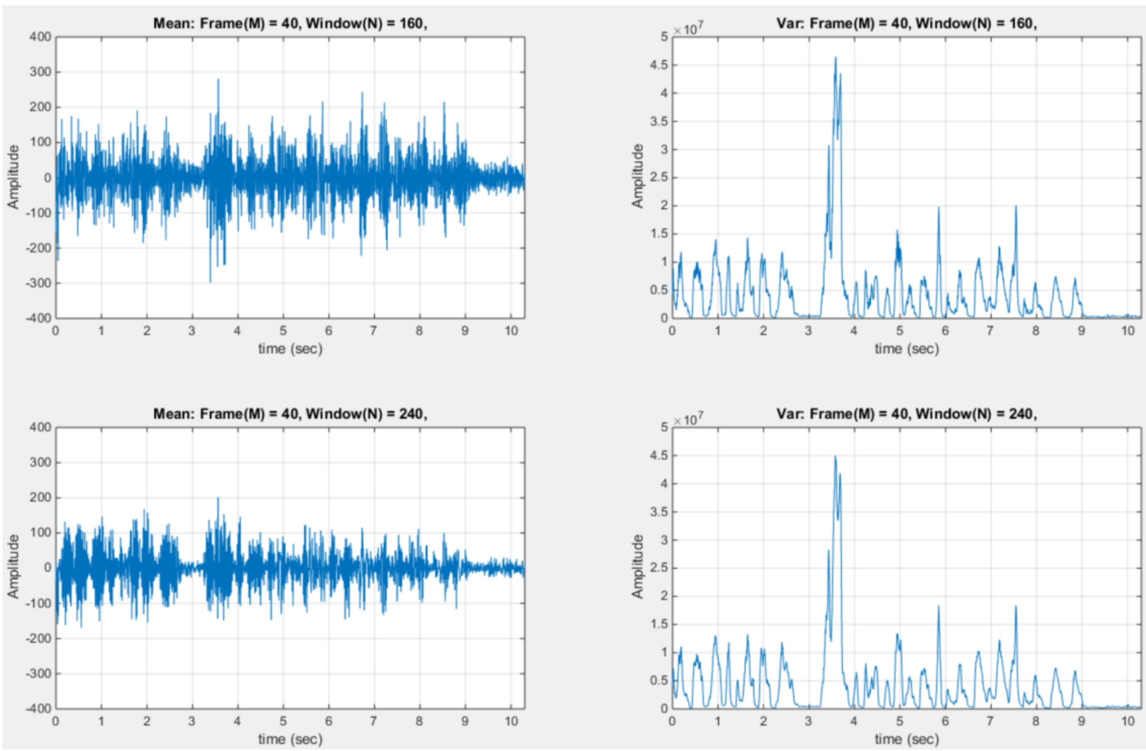


Figure 2: Speech signal mean/variance for all windows when frame = 40

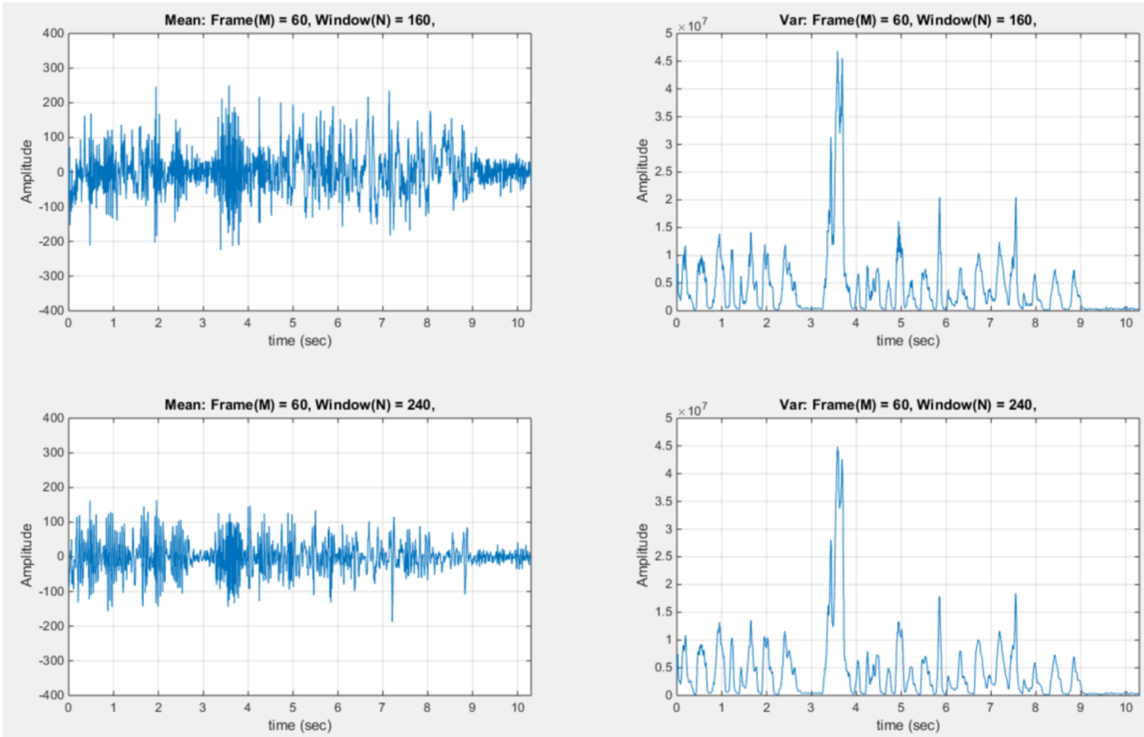


Figure 3: Speech signal mean/variance for all windows when frame = 60

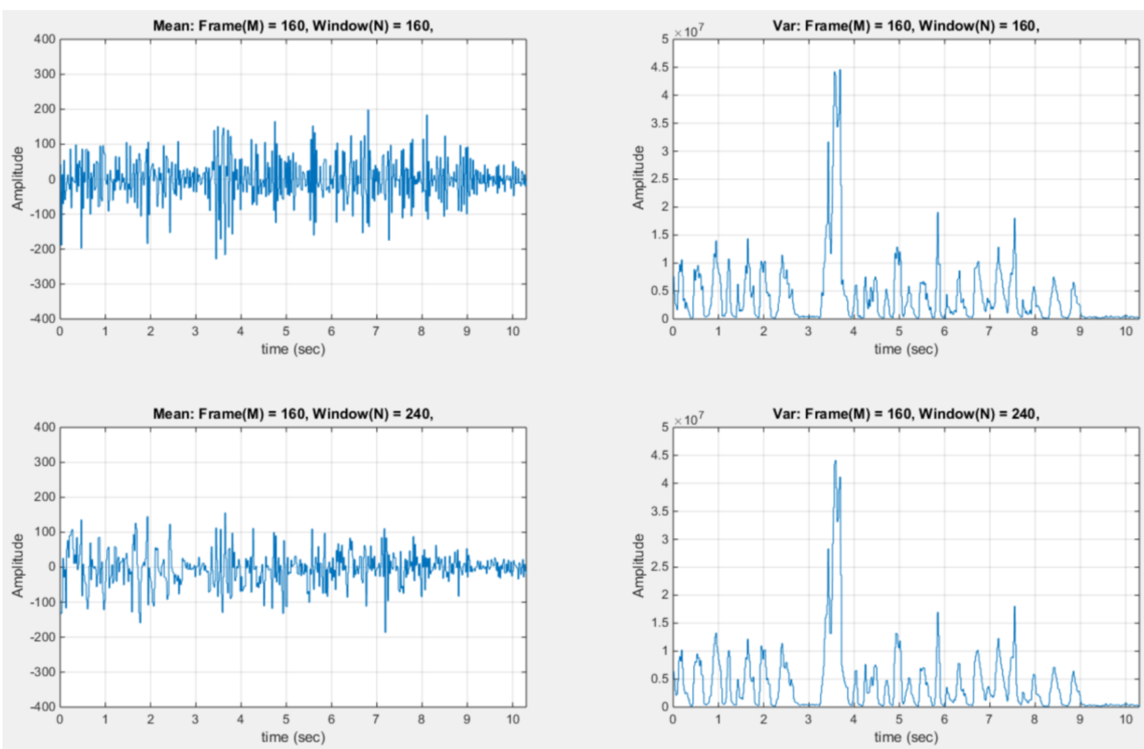


Figure 4: Speech signal mean/variance for all windows when frame = 160

As we break our speech signal into larger frame sizes we see more averaging of the signal. In conjunction with using larger window sizes we can see more easily where the signal changes the most. We have large variance around 3.5

seconds where the speaker says “Hello ECE […]”. For the speech signal, large changes in variance tell us when a person is speaking since our mean is zero.

With smaller window sizes we incorporate more of the data points so we get a closer representation of the original signal. In contrast, larger window/frame sizes make blockier looking plots.

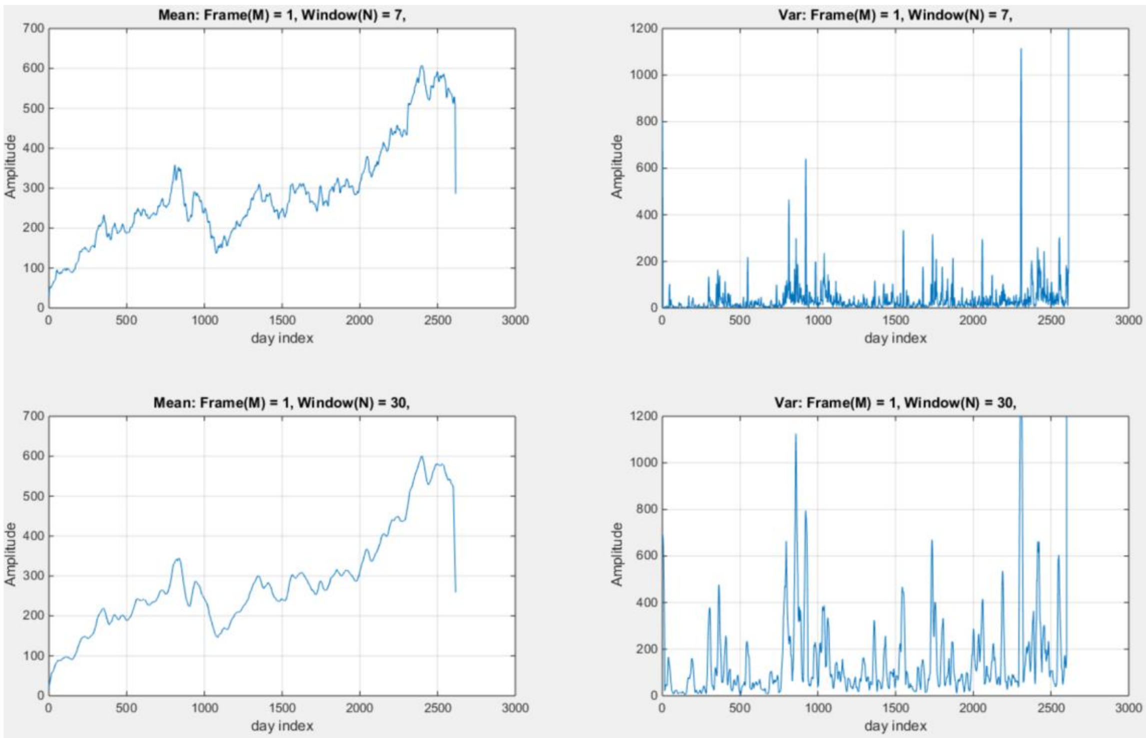


Figure 5: Google Stock mean/variance for all windows when frame = 1

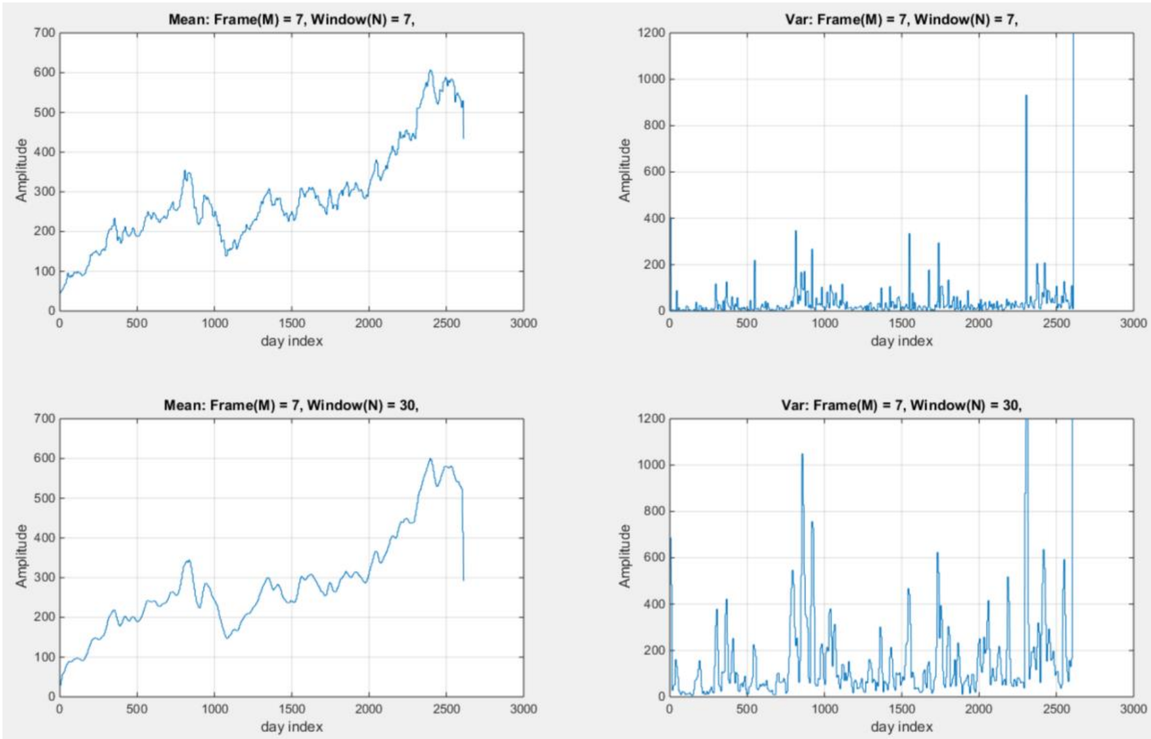


Figure 6: Google Stock mean/variance for all windows when frame = 7

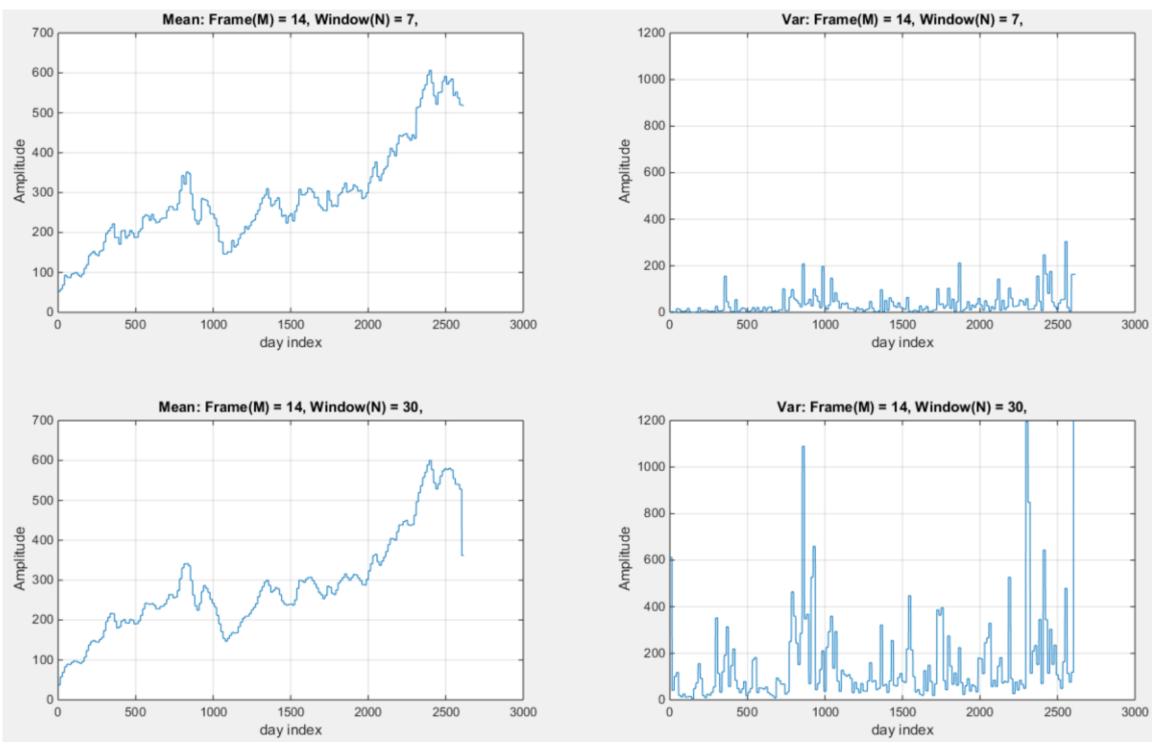


Figure 7: Google Stock mean/variance for all windows when frame = 14

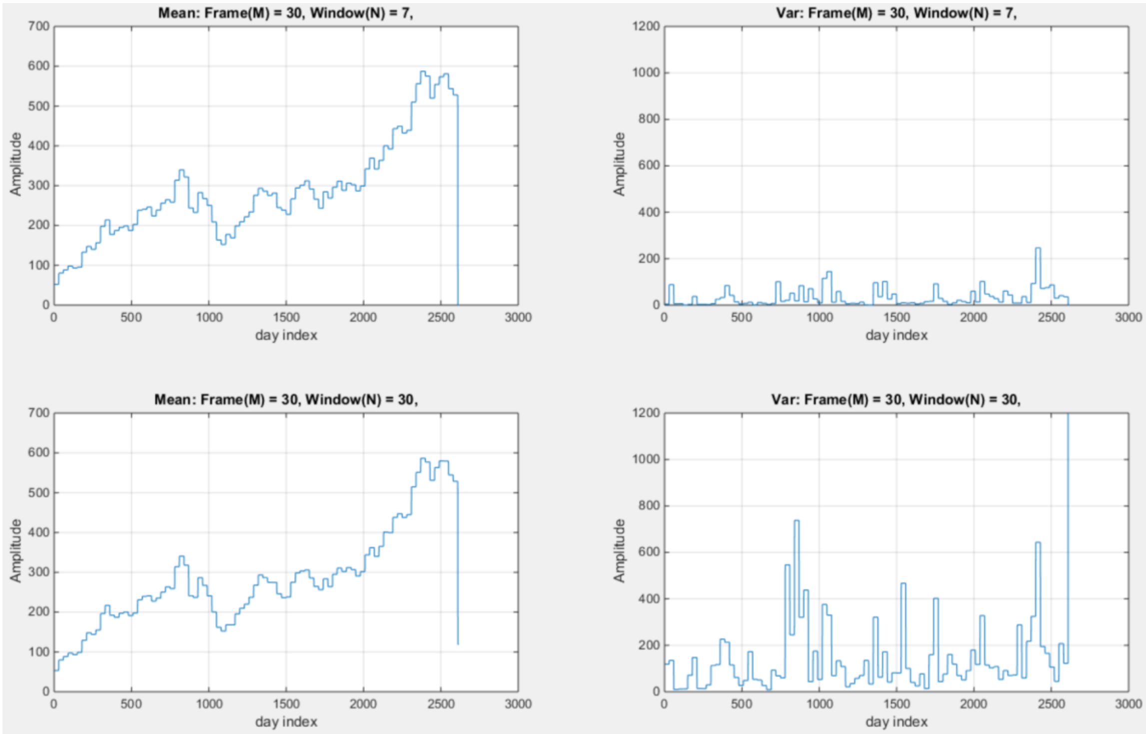


Figure 8: Google Stock mean/variance for all windows when frame = 30

Breaking our signal into larger window sizes averages more data points together, so what results are smoother plots. Also, as we increase our frame sizes our plots become blockier since we are utilizing fewer points. Having a frame size that is larger will show us the general characteristics of Google's stock price for a given range of days. For example, in the case where our frame is 30 we see we are averaging Google's stock price for an entire month at a time.

The variance shows the change in Google's stock for a range (window size) of days. We can see the Google's stock changed gradually in price for the ~750 days. Then we see a large increase in stock price. The change can correspond to a new technology being unveiled such as the android mobile operating system. The peak at around day ~750 is the most substantial. Afterwards we see increasing/decreasing in stock price at varying intervals. Specifically when our frame is 14 and window 7 and when our frame is 30 and window 7 we see smaller amplitudes in variance since our frame is larger to our window. Therefore we are skipping over portions of the stock price and Google's stock does not change too drastically between day to day.

Lastly, as time progress pass day ~1700 Google's stock does not stray below the mean of 286 so our variance never drops back down to zero.

## III. MATLAB CODE

```
% Let's first open the raw speech data file and store its values in a
% vector fn
%
fp=fopen('rec_01_speech.raw', 'r');
% Test Sine Wave
    %fp=fopen('rec_01_sine.raw', 'r');
fn=fread(fp,inf,'int16');
fclose(fp);

% Let's open the xls data file and store its values in a vector fn
% google_v00
%
google_v00 = xlsread('google_v00.xlsx');
% google_open = google_v00(:,1);
% google_high = google_v00(:,2);
% google_low = google_v00(:,3);
google_close = google_v00(:,4);
L_googleClose = length(google_close);

clear google_v00
% We are given a sample frequency of 8 kHz
%
fs = 8000;
L_speech = length(fn);
timeL = L_speech/fs;
% We can find the legnth of our signal given our sample frequency
%
t= linspace(0, timeL,L_speech);
```

1 We load our data into MatLab first. We utilize MatLab's built in xlsread to read in the Google Data which is stored in an excel document. We need to isolate the closing price from the excel document which is stored in the fourth column. We also need to let MatLab know what the sample frequency for our speech signal so that we can properly define our time axis. We can then plot the data into a figure.

```
% Plot our findings
%
figure('name','[ECE 3522] Class Assigment [1]');
subplot(2,2,[1 2]);
plot(t, fn);
grid on
xlabel('time (sec)');
ylabel('Amplitude');
title('Raw Speech Signal');
axis1 = axis;
axis([axis1(1) t(length(t)) axis1(3:4)]);

subplot(2,2,[3 4]);
dDays = 1:1:L_googleClose;
plot(dDays, google_close, 'g');
grid on
xlabel('day Index');
ylabel('Amplitude');
title('Google Data');
```

2 Plotting the data.

```
% Let us find the min/max val, mean, median, and variance
%
google_min = min(google_close);
google_max = max(google_close);
google_mean = mean(google_close);
google_median = median(google_close);
google_var = var(google_close);

fn_min = min(fn);
fn_max = max(fn);
fn_mean = mean(fn);
fn_median = median(fn);
fn_var = var(fn);

% Print our findings
%
out = sprintf('Google data: min = %f, max = %f, mean = %f, median = %f,
variance = %f \n'...
    , google_min, google_max, google_mean, google_median, google_var);
disp(out);
out = sprintf('Speech data: min = %f, max = %f, mean = %f, median = %f,
variance = %f \n'...
    , fn_min, fn_max, fn_mean, fn_median, fn_var);
disp(out);
```

3 Once we have our data read in correctly we can find the min/max/mean/median/variance of our datasets by using MatLab's built in functions. We store these values in variables that correspond to what they contain. We output the findings to the console.

```
% Now we separate into frames and windows.
%
M1 = [40, 60, 160];
N1 = [160, 240];



for M=M1 % frame
    num_frames = 1+round(L_speech / M);
    % Clear our intermediate window Mean/Variance holding variables.
    windowMean = zeros(length(N1),L_speech);
    windowVar = zeros(length(N1),L_speech);

    % index2 indexs the intermediate windows containing the mean/var of the
    % entire signal where the window = N
    index2 = 1;
    for N=N1 % window
        % index3 indexes the intermediate windows containing the mean/var
        % of the portion of the signal where the window = N
        for i=1:num_frames
            sig_wbuf = zeros(1, N);
            Mp = M*(i-1)+M/2;
            windowStart = Mp-N/2;
            windowEnd = windowStart+N-1;

            % transfer the data to this buffer:
            %  note that this is really expensive computationally
            %
            for j = 1:N
                index = windowStart + (j - 1);
                if ((index > 0) && (index <= L_speech))
                    sig_wbuf(j) = fn(index);
                end
            end

            windowMean_t = mean(sig_wbuf);
            windowVar_t = var(sig_wbuf);


           for j = 1:M
                index3 = Mp + (j - 1) - (M/2);
                if ((index3 > 0) && (index3 <= L_speech))
                    windowMean(index2,index3) = windowMean_t;
                    windowVar(index2,index3) = windowVar_t;
                end
           end

        end
        index2 = index2+1;
    end
```

4 The next part of the assignment calls for breaking the signal into smaller windows (N) separated by various frame sizes (M). We define the window/frame sizes we want to loop through. We first determine how many frames fit into our data (num_frames). Since our signal will not separate nicely into our desired window sizes we know there will be parts of the data where we will need to zero stuff. We initialize a buffer (sig_wbuf) of length N with all zeros. We

then load a section of data referenced by the windowStart and windowEnd variables into our buffer. If the section of data is not the same length as the window we know that zeros will fill in the rest of the buffer since we initialized it with zeros. Now that we have our buffer we find the mean/variance of this section of data into temporary variables. We have another loop that will plug the values into a bigger matrix that contains all the mean/variance values for the specified window size (indexed by variable index2).

```
figure('name','[ECE 3522] Class Assigment [1]');
     index3 = 1;
    for index2 = 1:size(windowMean,1)
        subplot(2,2,index3);
        plot(t, windowMean(index2,:));
        title(sprintf('Mean: Frame(M) = %d, Window(N) = %d,',M
,N1(index2)));
        xlabel('time (sec)');
        ylabel('Amplitude');
        grid on
        axis1 = axis;
        axis([axis1(1) t(length(t)) -400 400]);

        index3 = index3+1;

        subplot(2,2,index3);
        plot(t, windowVar(index2,:));
         title(sprintf('Var: Frame(M) = %d, Window(N) = %d,',M
,N1(index2)));
        xlabel('time (sec)');
        ylabel('Amplitude');
        grid on

        axis1 = axis;
        axis([axis1(1) t(length(t)) axis1(3:4)]);

        index3 = index3+1;
    end
end
```

5 To finish we plot the findings of each window/frame size into different figures. To make comparisons easier we keep constant axis throughout our plots. We also plot mean/variance plots of similar frame length into the same figure utilizing subplots. We repeat the same process for the Google Stock data.

## IV. CONCLUSIONS

By the end of the assignment we successfully read in data from external files. The two sources of data contrast the differences that can arise in the statistical values for a given dataset. In our audio signal we saw that our mean was close to zero since audio recordings are centered around zero. In comparison, Google's stock does not center around a price of $0.00. A speech signal changes in amplitude drastically so its variance changes more frequently. We can correlate large changes in variance for a speech recording to when the speaker is talking. Large variance in Google's stock price would be bad as it would signify that Google's stock changes drastically between day to day operations. Therefore, we learn that when performing statistical calculations on data it is important to note what the data represents since sometimes you may be expecting a high variance/mean or not.