# Computer Assignment 01: Simple Statistics

Tyler Berezowsky

January 22, 2015

## 1 Problem Statement

The purpose of this assignment was to develop proficiency loading and processing various types of time series data in MATLAB. Specifically, an audio signal and a stock price trend were loaded and processed. Processing consisted of calculating global statistics (minimum, maximum, mean, median, and variance), filtering the signal via the frame and window approach, and recalculating mean and variance for each window.

## 2 Approach and Results

### 2.1 The Signal Abstraction

While a Google stock trend and an audio signal qualitatively are completely separate identities, quantitatively they are described the same. Both are time dependent signals described by sample points of varying amplitude values and indexed by either the sample rate or a time vector. Therefore, once the signal is translated into a matrix or vector, the procedure for processing is identical in MATLAB.

The Google stock data set contained five vectors; the stock's opening price, closing price, minimum price, peak price and the date at which the data was collected. It is assumed the date is independent and all other vectors are dependent. The data was delivered through an excel spreadsheet.

The audio data set consisted of a single vector. The points were plotted in time by generating a time vector based on the sample rate of the audio signal. The data was deliver through a .raw file consisting of 16 bit signed integers for each sample point.

### 2.2 Loading the Data Set

The Google stock data was loaded via the `xlsread` command. The command transferrs each column of a spreadsheet into a vector of a matrix.

The audio signal was transfer to MATLAB through the command `fread` which places the sample points in the .raw file into a vector.

### 2.3 Plotting

The entire Google stock data set was plotted utilizing the `highlow` command which plots a vertical bar that spans the high and low prices of the stocks value on a specific day. In addition, tick marks are plotted on the bar representing the opening and closing prices. The plot's abscissa is time given as dates. The closing price of the stock and date was also plotted as requested. The plots are displayed in figure 1.

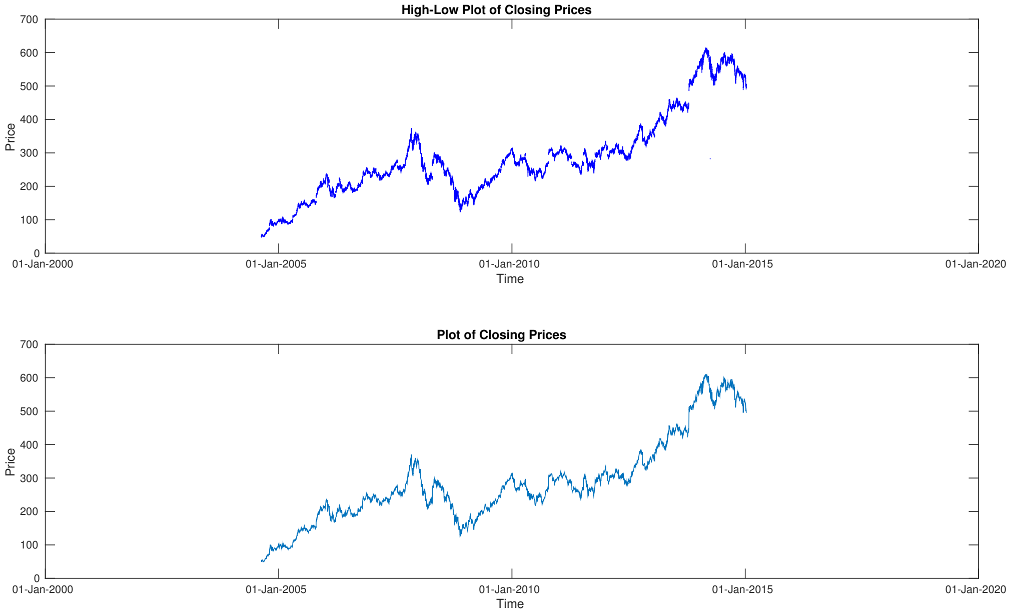

Figure 1: Plot of Google stock prices via a "High-Low" plot with displays the range of prices for each day, and a normal plot.

The overall trend of the stock for both plots is identical. The `highlow` plot appears nosier due its data points representing the range of the stock price for a day instead of a single point.

Plotting the audio signal was identical to plotting the closing stock prices. A time vector was calculated for the signal utilizing the sample rate and the number of samples in the signal. The plot can be seen in figure 2.

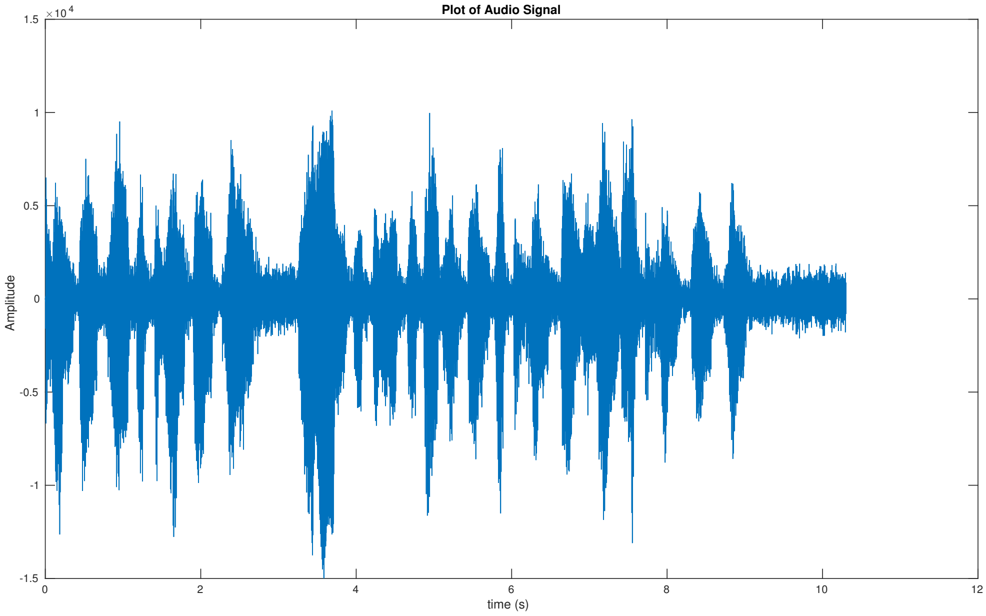

Figure 2: Plot of audio signal sampled at 8kHz.

### 2.3.1 Calculating Global Statistics

Calculation of global statistics for both data sets was trivial utilizing built in MATLAB commands. The Google stock trend contain multiple vectors of data, therefore the metrics were computed for each vector. The global statistics for the Google stock trend and the audio signal can be seen in table 1 and 2 below.

| Vector | Min | Max | Median | Mean | Variance |
|---|---|---|---|---|---|
| Open | 49.5 | 612.79 | 265.06 | 286.79 | 16198 |
| High | 50.82 | 613.83 | 267.98 | 289.65 | 16345 |
| Low | 47.93 | 608.69 | 262.31 | 283.82 | 16030 |
| Close | 49.95 | 609.47 | 264.83 | 286.74 | 16194 |

Table 1: Table of global statistics for Google stock trend

The global statistics between vectors varied less than ten dollars for minimum, maximum, median, and mean. The variance for stock prices displayed drastic change in comparison. Specifically, the change in variance between the high and low stock prices is 315.

| Min | Max | Median | Mean | Variance |
|---|---|---|---|---|
| -14993 | 10104 | 83 | -0.38912 | 4.14E+06 |

Table 2: Table of global statistics for audio signal

Interesting to note is magnitude and polarity of the mean for the audio signal. The existence of a non-zero mean suggests a DC bias, in this case negative, but its relevance is diminished by the range of values the minimum and maximum describe. In comparison to range of the signal's medium ($\pm 32,767$) or the actual range of the signal's amplitudes ($10,104 - (-14,993)$), the mean is essential zero. This is due to an audio signals equal distribution of sample points on the negative and positive axis canceling the sum of amplitudes for the mean calculation. The large swings in amplitude along with the near zero mean, explains the high level of variance in the audio signal compared to the Google stock trend. The calculation for variance is listed below:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{1}$$

### 2.3.2 Frame and Windowing

The frame and windowing technique, covered extensively last semester, consists of splitting the signal into segments called frames. These frames are sized by the number of data points or samples they contain. The number of samples the frame contains is denoted by M. The frame is the increment in which the signal is "stepped" through. For each frame in the signal, a window of data is extracted from the signal. A window is a chunk of data points or samples, similar to the frame, centered at the middle of the frame. The number of samples in a window is denoted by N. For this assignment, the mean and variance of each window is taken to represent a frame of data.

The Google stock prices were run with the following frame and window combinations for the closing prices: N = [7 30] and M = [1 7 14 30]. The plots generated can be seen in figure 3 below. The frame and window configuration is listed above each subplot as the title. The blue signal is the mean of the window and the red signal is the variance of the window. As the plots describe, the mean of the google stock price is increasing with time. Also can be seen is the variance spiking proportionally with the range between the mean of a window and it's samples.

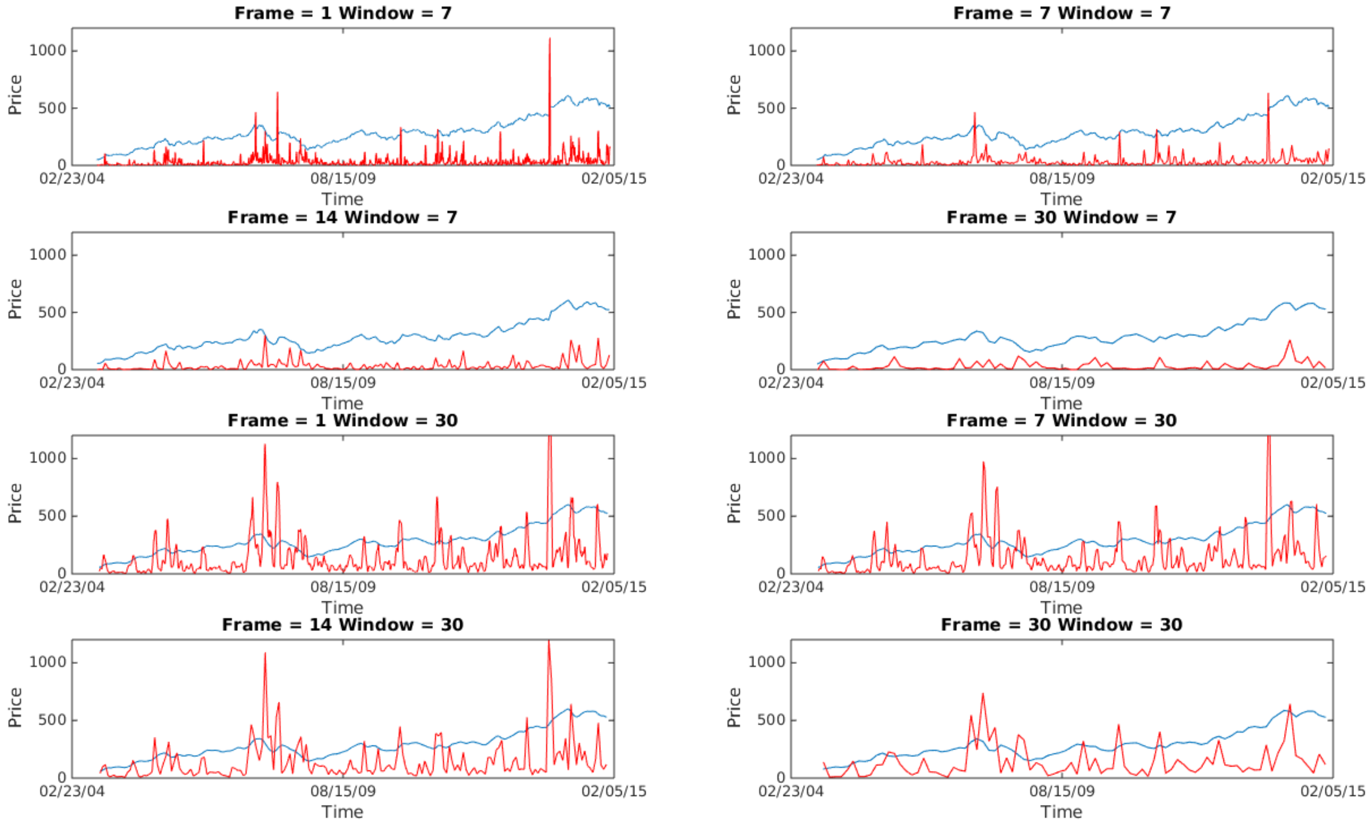

Figure 3: Plot of Google stock prices filtered via frame and windowing. The blue signal is the mean of the windows. The red signal is the variance of the windows.

The same process was repeated for the audio signal with the following parameters: M = [40 80 160] N = [160 240]. The plot below in figure 4 display the results. The blue signal represents the mean of the windows and the black signal represents the variance of the windows. As the plots display, the mean of the windows is magnitudes smaller than the range of the signal's amplitudes calculated from the global statistics. In addition, variance appears to follow the amplitude of the signal which is likely due its similarities to the power calculation of a signal with a near zero mean.

$$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{n} x_i} \tag{2}$$

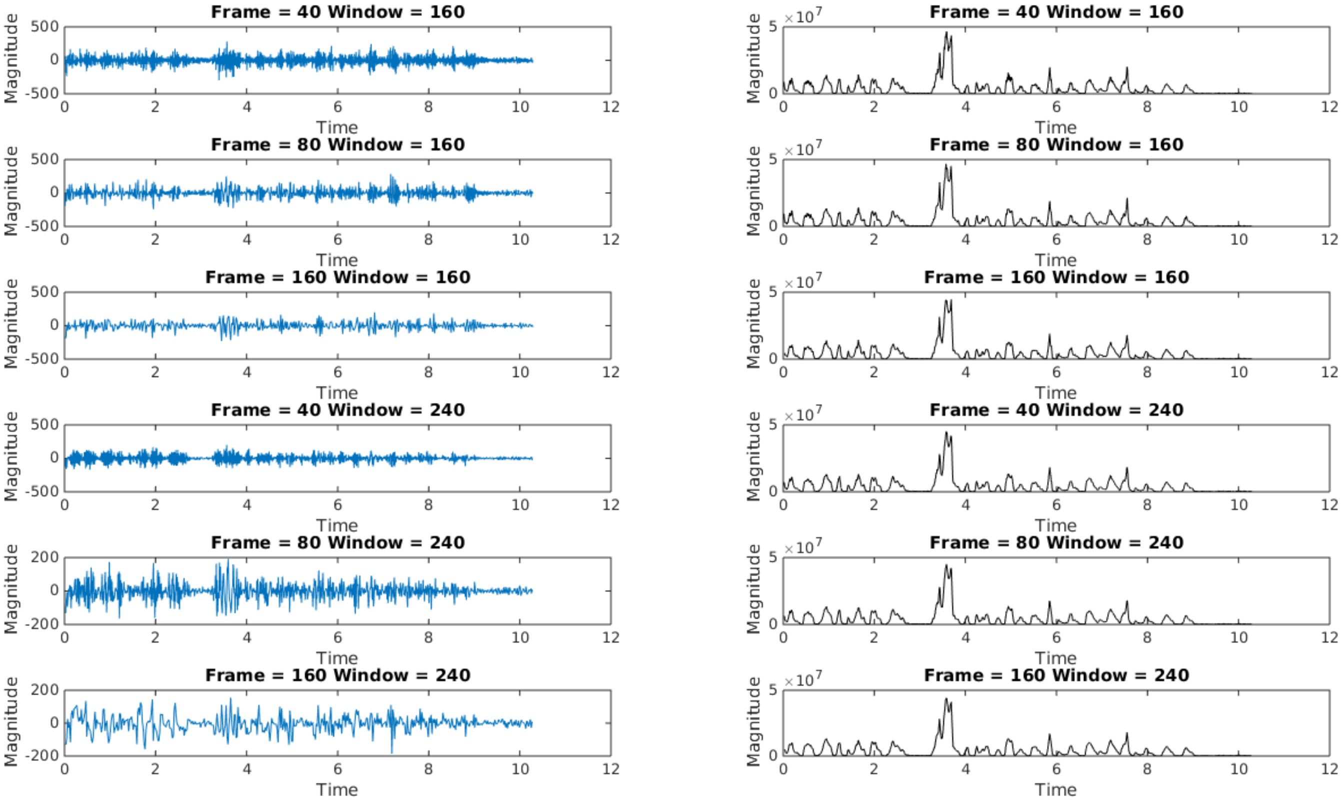

Figure 4: Plot of audio signal filtered via various frame and window combinations.

Utilizing the frame and windowing technique permitted the signal to be examined within segments of time, instead of the entire length of the signals existence as with global statistics. For example, the global mean for the Google stock's closing prices was $286.74, but as all the plots in figure 3 display, the mean is steadily increasing with time. The global overall mean does not convey if the stock's prices were increasing or decreasing with time. Through the frame and windowed data, the mean was examined through chucks permitting the mean to be displayed for a segment which in turn describes the direction or rate of the signal. This information could be used to predict the future trend of the stock price.

Frame and windowing the audio signal did not reveal any trends in the mean. The mean did vary through time, but the mean of the windows remained magnitudes smaller compared to the range the signal's amplitudes displayed. This is concurrent with the zero-cross nature of audio signals as previously discussed. Further evidence of this is displayed when the frame size or window size increased to include more sample points, and the mean decreases for each frame. As the number of sample points increase, the mean converges further.

# 3 MATLAB Code

The MATLAB code is displayed in listing 1 below. The only peculiarity from the frame and window technique established last semester is the absence of zero stuffing. Zero stuffing would artificially change the mean and variance of a window and therefore any frames with a window which extended past the signal were disregarded. This permitted the windows to be concatenated into a single matrix.

Listing 1: MATLAB code for ca_01

```
%% Import Google Stock Prices
% Import the excel spreadsheet into an array

clear; clc; clear all;

filename = 'google_v00.xlsx'
[data, header, raw] = xlsread(filename);

%% Plot the Stock Data with Variance
% Plot data utilizing the highlow function which  plots the high, low,
% opening, and closing prices of an asset. Plots are vertical lines whose
% top is the high, bottom is the low, open is a short horizontal tick to
% the left, and close is a short horizontal tick to the right.

% ease data manipulation with names
High = data(:,3);
Low = data(:,4);
Close = data(:,5);
Open = data(:,2);
Dates = x2mdate(data(:,1), 1);

subplot(211);
highlow(High, Low, Close, Open, 'b', Dates, 1);
datetick('x', 1, 'keeplimits')
xlabel('Time')
ylabel('Price')
title('High-Low Plot of Closing Prices')

subplot(212);
plot(Dates, Close);
xlabel('Time')
ylabel('Price')
datetick('x', 1)
title('Plot of Closing Prices')


%% Calaculate Global Statistics of Stock Prices
% The first column is the date therefore it is excluded when
% global statistics are calculated

Min = min(data);
Max = max(data);
Mean = mean(data);
Median = median(data);
Variance = var(data);

googleStats = table(Min, Max, Mean, Median, Variance);

%% Frame and Window Stock Data
% Windows which exceed frames will be trunkcated instead of zero-stuffed.
% This will "hopefully" prevent skewing the mean and variance.
% Actually no. MATLAB will not permit irregular array sizes to be
% concatenated. There maybe a work around, but I believe Python would be a
% better environment. Frames will be disregarded.
% Okay, well to plot ... how do I plot with a time-series.
% The time values corresponding to the window values selected where
% averaged to reflect the data point calculated from the window.

% N (Window Size) = 7, 30
% M (Frame Size) = 1, 7, 14, 30

clear windows
clear windowDates

N = [7 30];
M = [1 7 14 30];
googleFW = table()
fwMean = []
plotIndex = 1;
figure()

% Raster Through Windows and Frames Dimensions
%
for x = 1:length(N)
    for y = 1: length(M)

        sigLength = length(data);

        frameSize = M(y);
        windowSize = N(x);

        % initialize arrays for windows or dates
        windows = []
        windowDates = []

        for z = 1:frameSize:sigLength
          % calculate the frame center, and then the right and left window indexes
          frameCenter = floor( z + frameSize/2 ) ;
          windowLeft = floor( (frameCenter - 1) - 0.5*windowSize );
          windowRight = windowLeft + windowSize - 1;

          % insure the window never exceeds signal
          if (windowLeft >= 1) && (windowRight <= sigLength)
            windows = [windows; data(windowLeft : windowRight, 5)'];
            windowDates = [windowDates; Dates(windowLeft: windowRight)'];
          end
        end

        centerDates = mean(windowDates, 2);

        windowsMean = mean(windows, 2);
        windowsVariance = var(windows, 0, 2);

        subplot(4, 2, plotIndex);
        plot(centerDates, windowsMean, centerDates, windowsVariance, 'r')
        ylim([0 1200]);
        datetick('x',2 ,'keeplimits', 'keepticks');
        xlabel('Time')
        ylabel('Price')
        plotIndex = plotIndex + 1
        titleStr = sprintf('Frame = %d Window = %d',...
            frameSize, windowSize);
        title(titleStr);
        fwMean = [fwMean ; mean(windowsMean)]
    end
end

%% Import Audio File
% Import the .raw file into an array

filename = 'rec_01_speech.raw';
file = fopen(filename, 'r');
audio = fread(file, inf,'short');

%% Plot the Audio Signal
```

```
%

sampleRate = 8e3
time = (0:length(audio)-1)*1/(sampleRate);

figure();
plot(time, audio);
title('Plot of Audio Signal')
xlabel('time (s)')
ylabel('Amplitude')


%% Calculate Global Statistics of Audio Signal
% The stats are stored in the table signalStats

Min = min(audio);
Max = max(audio);
Mean = mean(audio);
Median = median(audio);
Variance = var(audio);

signalStats = table(Min, Max, Mean, Median, Variance);

%% Frame and Window Audio Signal

% N (window size)
% M (frame size)
M = [40, 80, 160];
N = [160 240];

plotIndex = 1;
figure();

% Raster Through Windows and Frames Dimensions
%
for x = 1:length(N)
    for y = 1: length(M)

        sigLength = length(audio);

        frameSize = M(y);
        windowSize = N(x);

        % initialize arrays for windows or dates
        windows = []
        windowTimes = []

        for z = 1:frameSize:sigLength
          % calculate the frame center, and then the right and left window indexes
          frameCenter = floor( z + frameSize/2 ) ;
          windowLeft = floor( (frameCenter - 1) - 0.5*windowSize );
          windowRight = windowLeft + windowSize - 1;

          % insure the window never exceeds signal
          if (windowLeft >= 1) && (windowRight <= sigLength)
            windows = [windows; audio(windowLeft : windowRight)'];
            windowTimes = [windowTimes; time(windowLeft : windowRight)];
          end
        end

        meanTimes = mean(windowTimes, 2);
        windowsMean = mean(windows, 2);
        windowsVariance = var(windows, 0, 2);
        subplot(6, 2, plotIndex);
        plot(meanTimes, windowsMean);
        xlabel('Time')
        ylabel('Magnitude')
        titleStr = sprintf('Frame = %d Window = %d',...
            frameSize, windowSize);
        title(titleStr);
        plotIndex = plotIndex + 1;
        subplot(6, 2, plotIndex);
        plot(meanTimes, abs(windowsVariance), 'k');
        xlabel('Time')
        ylabel('Magnitude')
        plotIndex = plotIndex + 1
        titleStr = sprintf('Frame = %d Window = %d',...
            frameSize, windowSize);
        title(titleStr);
    end
end
```

## 4 Conclusions

Frame and windowing data can introduce another level of information depending upon the signal. Global statistics provided only a singular value to represent the signal for all time while frame and windowing delivers multiple views through segments of time in which rates could be calculated for future prediction.

How a signal is evaluated is not uniform process. For example, frame and windowing the mean of the audio signal delivered little information due to nature of the medium, but illuminated the Google's average stock price was increasing with time.