Table of Contents


```
% Alexander Hay
% Final

fprintf('\n');
fprintf('Alexander Hay\n');
fprintf('NUIN 408\n');
fprintf('Final\n');

% # # # µ #
```

Alexander Hay
NUIN 408
Final

Problem Set 1

********** 1a

```
fprintf('********** Problem Set 1 **********\n');
fprintf('\n');
```

```
fprintf('1a ********** \n');
fprintf('Having too many parameters to fit the data may reintroduce
 noise into the data youre trying to filter.\n');
fprintf('Too many parameters may also make it difficult to visualize
 any data clusters.\n');
fprintf('\n');

% ********************************************
% 1b

fprintf('1b ********** \n');
fprintf(['With the residuals you can calculate the coefficient of
 determination.\n',...
'That tells you how much variance is explained by your model.\n',...
'\n']);
% ********************************************
% 1c

fprintf('1c ********** \n');
fprintf(['Binary Random Variable: An action potential crossing (or not
\n',...
'\tcrossing) the excitation threshold\n',...
'Discrete Random Variable: Population of cells\n',...
'Continuous Random Variable: Cell voltage\n',...
'\n']);

% ********************************************
% 1d

fprintf('1d ********** \n');
fprintf('Standard deviation measures the variation of the data.\n');
fprintf('Standard error measures how far the sample mean is from the
 true mean.\n');
fprintf('\n');


% ********************************************
% 1e

fprintf('1e ********** \n');
fprintf('Assumption 1a: Data is a normal distribution\n');
fprintf('Counter scenario: flu disproportionally affects children and
 elderly, creating two population means\n');
fprintf('Test: Jarque-Bera test\n');
fprintf('\n');
fprintf('Assumption 1b: Reasonably large population (there enough data
 to plot a normal distribution)\n');
fprintf('Counter scenario: Sample size of one from a population of one
\n');
fprintf('\n');
fprintf('Assumption 2: Populations tested have the same variance\n');
fprintf('Counter scenario: Men and women have same average
 intelligence, men have more vairance in intelligence than women\n');
```

```
fprintf('Test: 2-sample F test\n');
fprintf('\n');
fprintf('Assumption 3: Data sampled is representative of the total
 population\n');
fprintf('Counter scenario: Testing quality of grain where the sample
 tested was a pebble that got mixed in with the grain\n');
fprintf('Test: Z-test\n')
fprintf('\n');

% ********************************************
% 1f

fprintf('1f ********** \n');
fprintf('95%% confidence interval means that there is 95%% certainty
 that the range of values contain the true mean\n');
fprintf('\n');

% ********************************************
% 1g

fprintf('1g ********** \n');
fprintf('p-values indicate the strength of the evidence against the
 null-hypothesis. A p-value of 0.05 shows a 5% probability\n');
fprintf('that the null hypothesis is correct\n');
fprintf('\n');

********** Problem Set 1 **********

1a **********
Having too many parameters to fit the data may reintroduce noise into
 the data youre trying to filter.
Too many parameters may also make it difficult to visualize any data
 clusters.

1b **********
With the residuals you can calculate the coefficient of determination.
That tells you how much variance is explained by your model.

1c **********
Binary Random Variable: An action potential crossing (or not
 crossing) the excitation threshold
Discrete Random Variable: Population of cells
Continuous Random Variable: Cell voltage

1d **********
Standard deviation measures the variation of the data.
Standard error measures how far the sample mean is from the true mean.

1e **********
Assumption 1a: Data is a normal distribution
Counter scenario: flu disproportionally affects children and elderly,
 creating two population means
Test: Jarque-Bera test
```

```
Assumption 1b: Reasonably large population (there enough data to plot
 a normal distribution)
Counter scenario: Sample size of one from a population of one

Assumption 2: Populations tested have the same variance
Counter scenario: Men and women have same average intelligence, men
 have more vairance in intelligence than women
Test: 2-sample F test

Assumption 3: Data sampled is representative of the total population
Counter scenario: Testing quality of grain where the sample tested was
 a pebble that got mixed in with the grain
Test: Z-test

1f **********
95% confidence interval means that there is 95% certainty that the
 range of values contain the true mean

1g **********
p-values indicate the strength of the evidence against the null-
hypothesis. A p-value of 0.05 shows a 5that the null hypothesis is
 correct
```

Problem Set 2

********** 2a

```
fprintf('\n');
fprintf('********** Problem Set 2 **********\n');
fprintf('\n');

rat = linspace(1,15,15);
lo_anx = [3,6,3,7,7,2,3,5,8,10,2,5,7,8,11];
hi_anx = [5,6,7,11,8,4,8,9,12,17,15,3,16,14,9];

fprintf('2a ********** \n');
fprintf('Low anxiety\n');
fprintf('mean:\t%.2f \t sigma:\t%.2f\n',mean(lo_anx),std(lo_anx));
fprintf('\n');
fprintf('High anxiety\n');
fprintf('mean:\t%.2f \t sigma:\t%.2f\n',mean(hi_anx),std(hi_anx));
fprintf('\n');
```

```
********** Problem Set 2 **********

2a **********
Low anxiety
mean: 5.80   sigma: 2.83

High anxiety
mean: 9.60   sigma: 4.42
```

****

2b

```
n = length(rat);
mean_lo_anx = mean(lo_anx);
mean_hi_anx = mean(hi_anx);
std_lo_anx = std(lo_anx);
std_hi_anx = std(hi_anx);

% standard error is sigma/root(N-1) (per hw2)
sem_lo_anx = std_lo_anx/sqrt(n-1);
sem_hi_anx = std_hi_anx/sqrt(n-1);

lo_anx_int_lo = mean_lo_anx - (1.96 * sem_lo_anx);
lo_anx_int_hi = mean_lo_anx + (1.96 * sem_hi_anx);

hi_anx_int_lo = mean_hi_anx - (1.96 * sem_lo_anx);
hi_anx_int_hi = mean_hi_anx + (1.96 * sem_hi_anx);

fprintf('2b ********** \n');
fprintf('Low anxiety\n');
fprintf('95%% interval: %.2f - %.2f\n', lo_anx_int_lo, lo_anx_int_hi);
fprintf('\n');
fprintf('High anxiety\n');
fprintf('95%% interval: %.2f - %.2f\n', hi_anx_int_lo, hi_anx_int_hi);
fprintf('\n');
```

```
2b **********
Low anxiety
95% interval: 4.32 - 8.12

High anxiety
95% interval: 8.12 - 11.92
```

****

2c

```
fprintf('2c ********** \n');
fprintf('A two sample t-test would determine if the data comes from
 different populations or not.\n');
fprintf('In this context it would determine if each condition have
 different means (effects).\n');
fprintf('\n');
```

```
2c **********
A two sample t-test would determine if the data comes from different
 populations or not.
In this context it would determine if each condition have different
 means (effects).
```

****

2d

```
fprintf('2d ********** \n');
fprintf('Null hypothesis: that both conditions have the same means
\n');
fprintf('μ_lo_anx = μ_hi_anx\n')
fprintf('\n');
fprintf('Alt hypothesis: that both conditions have the different means
\n');
fprintf('μ_lo_anx # μ_hi_anx\n')
fprintf('\n');
```

```
2d **********
Null hypothesis: that both conditions have the same means
μ_lo_anx = μ_hi_anx

Alt hypothesis: that both conditions have the different means
μ_lo_anx # μ_hi_anx
```

****

2e

```
[h p ci stats] = ttest2(lo_anx, hi_anx);

fprintf('2e ********** \n');
fprintf('Two-sample t-test results:\n');
fprintf('h:\t%.f\n',h);
fprintf('DoF:\t%.f\n',stats.df);
fprintf('p:\t%.4f\n',p);
fprintf('\n');
```

```
2e **********
Two-sample t-test results:
h: 1
DoF: 28
p: 0.0091
```

****

2f

```
fprintf('2f ********** \n');
fprintf('The results of the two-sample t-test indicate that the data
 from each condition represent different effects\n');
fprintf('\n');
```

```
2f **********
The results of the two-sample t-test indicate that the data from each
 condition represent different effects
```

****

2g

```
fprintf('2g ********** \n');
fprintf('Statistical power:\t%.2f\n',sampsizepwr('t2',[mean_lo_anx
 std_lo_anx],mean_hi_anx,[],n));
fprintf('Rats needed for power of 0.95:\t%.f\n',sampsizepwr('t2',
[mean_lo_anx std_lo_anx],mean_hi_anx,.95,[]));
fprintf('Statistical power is the likelihood of the test rejecting the
 hypothesis\n');
fprintf('A power of 0.95 demonstrates a 95%% confidence of rejecting
 the null hypothesis\n');
fprintf('\n');
```

```
2g **********
Statistical power: 0.94
Rats needed for power of 0.95: 16
Statistical power is the likelihood of the test rejecting the
 hypothesis
A power of 0.95 demonstrates a 95% confidence of rejecting the null
 hypothesis
```

****

2h

```
fprintf('2h ********** \n');
fprintf('Using the same rats would mean using a paired t-test\n');
fprintf('Statistical power:\t%.2f\n',sampsizepwr('t',[mean_lo_anx
 std_lo_anx],mean_hi_anx,[],n));
fprintf('Advantages:\trequires fewer rats\n');
fprintf('\t\teliminates variations between rats\n');
```

```
2h **********
Using the same rats would mean using a paired t-test
Statistical power: 1.00
Advantages: requires fewer rats
  eliminates variations between rats
```

Problem Set 3

********** 3a

```
fprintf('\n');
fprintf('********** Problem Set 3 **********\n');
fprintf('\n');

% Protein rate
P_a = 0.75;
P_b = 0.20;
```

```
P_c = 0.05;

% Protein misfold rate
P_ma = 0.5/100;
P_mb = 3.5/100;
P_mc = 5.0/100;

% Total misfold
P_y = (P_a * P_ma) + (P_b * P_mb) + (P_c * P_mc);

% Prob misfold from B
P_yb = (P_b * P_mb)/P_y;

fprintf('3 ********** \n');
fprintf('Probability that the source of misfolding came from
 population B:\n');
fprintf('%2.2f %%\n',P_yb*100);
fprintf('\n');
fprintf('See code for work\n');
fprintf('\n');
```

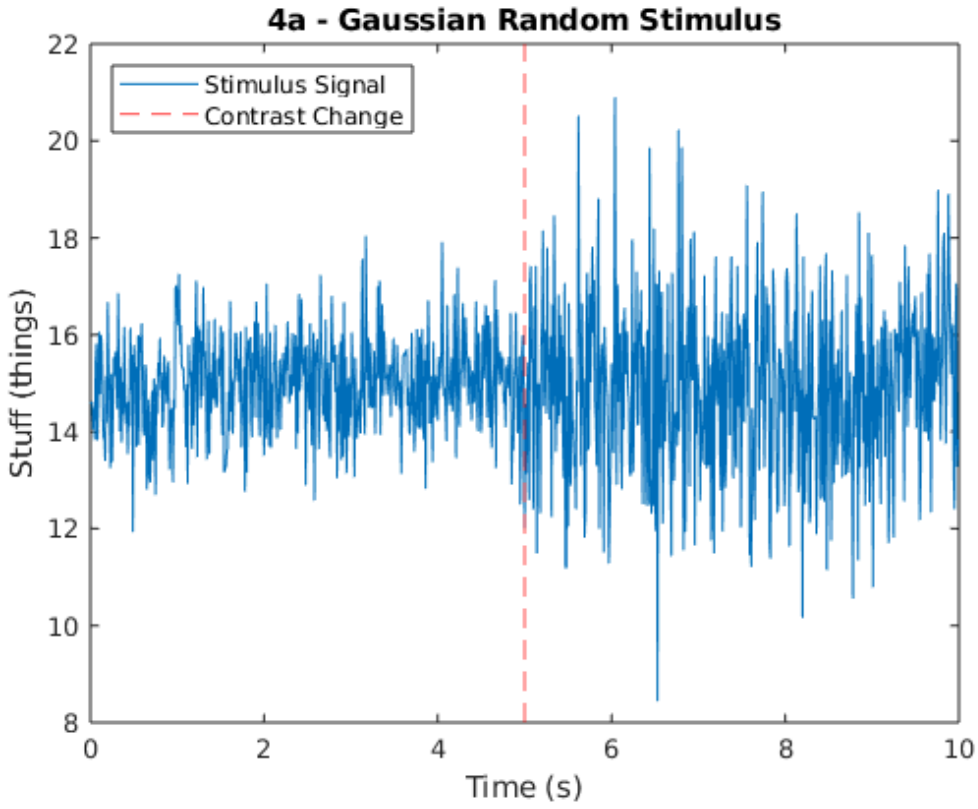
```
********** Problem Set 3 **********

3 **********
Probability that the source of misfolding came from population B:
52.83 %

See code for work
```

Problem Set 4

```
****************************************** 4a

fprintf('\n');
fprintf('********** Problem Set 4 **********\n');
fprintf('\n');

% coefficient of variation = #/μ

% define time axis
fs = 100;   % Sampling frequency
A = 10;     % Amplitude

t = 5;
timeaxis = 0:1/fs:((fs*t)-1)/fs;

% define mu
mu = 5;
sigma = mu*0.20;
y1 = A + sigma*randn(size(timeaxis)) + mu;

sigma = mu*0.36;
```

```
y2 = A + sigma*randn(size(timeaxis)) + mu;

% redefine time axis to plot
t = 10;
timeaxis = 0:1/fs:((fs*t)-1)/fs;
y = [y1 y2];

figure_4a = figure;
plot(timeaxis, y,'DisplayName','Stimulus Signal');
hold on;
xline(5,'--','Color','r','DisplayName','Contrast Change');
title('4a - Gaussian Random Stimulus');
xlabel('Time (s)');
ylabel('Stuff (things)');
legend('Location','northwest');
hold off;

fprintf('4a ********** \n');
fprintf('see figure 4a\n');
fprintf('\n');
```

```
********** Problem Set 4 **********

4a **********
see figure 4a
```



****

4b

```
fprintf('4b ********** \n');
fprintf('A two-sample F-test for equal variances would be the
 statistical test. It tests\n');
fprintf('if two samples come from normal distributions with the same
 variance (contrast)\n');
fprintf('versus the alternative that they dont\n');
fprintf('\n');
```

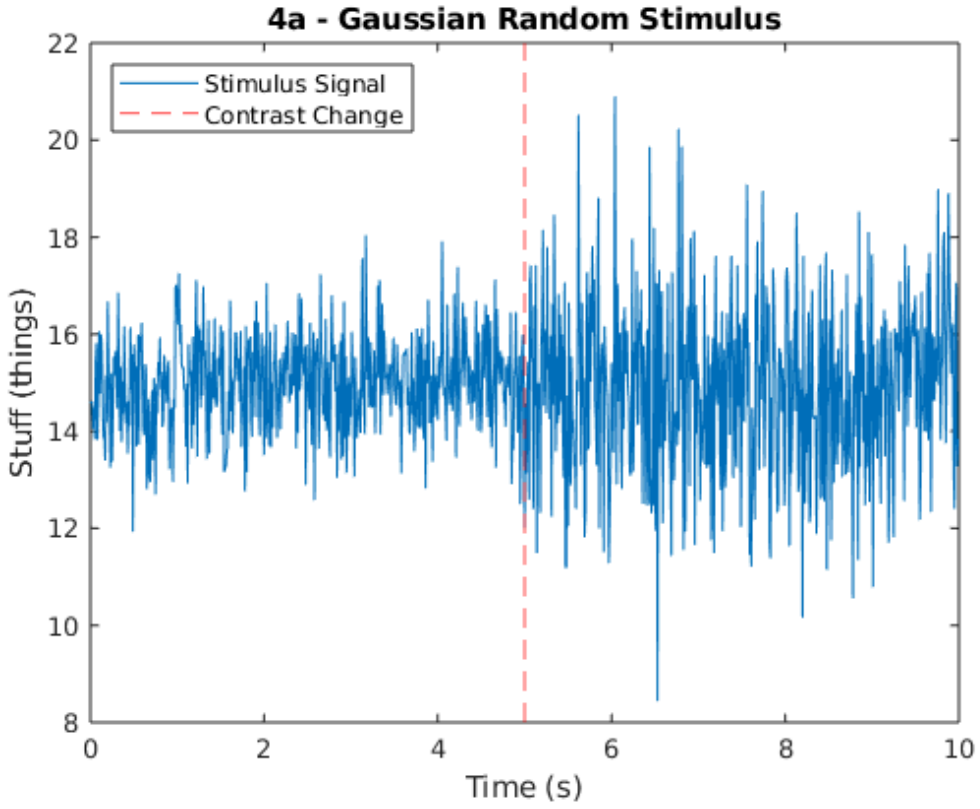
```
4b **********
A two-sample F-test for equal variances would be the statistical test.
 It tests
if two samples come from normal distributions with the same variance
 (contrast)
versus the alternative that they dont
```

****

4c

```
h = zeros(1000,1);
p = zeros(1000,1);

% iterate through data, offeset because matlab indexing starts at 1
for i = 101:length(y);
    a = y(i-100:i-51);
    b = y(i-50:i-1);
    [h(i),p(i)] = vartest2(a,b);    % Two-sample F-test
end

figure_4c = figure;
semilogy(p,'DisplayName','p value');
hold on;
title('Figure 4c - Two-Sample F-Test P Values');
ylabel('p values');
xlabel('time (ms)');
xlim([0 1000]);
xline(101,'--','Color','k','DisplayName','1.00 second');
legend('Location','southeast');
hold off;

fprintf('4c ********** \n');
fprintf('see figure 4c\n');
fprintf('\n');
```

```
4c **********
see figure 4c
```


4a - Gaussian Random Stimulus
Stimulus Signal
Contrast Change
Stuff (things)
Time (s)
22
20
18
16
14
12
10
8
0
2
4
6
8
10

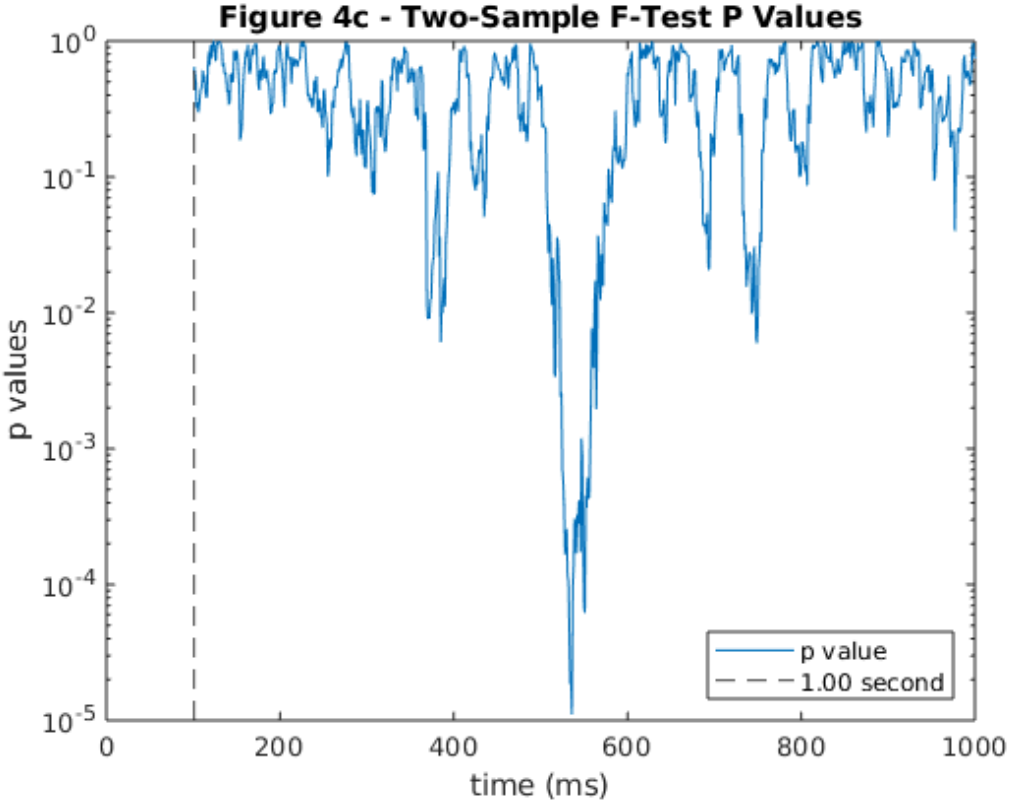


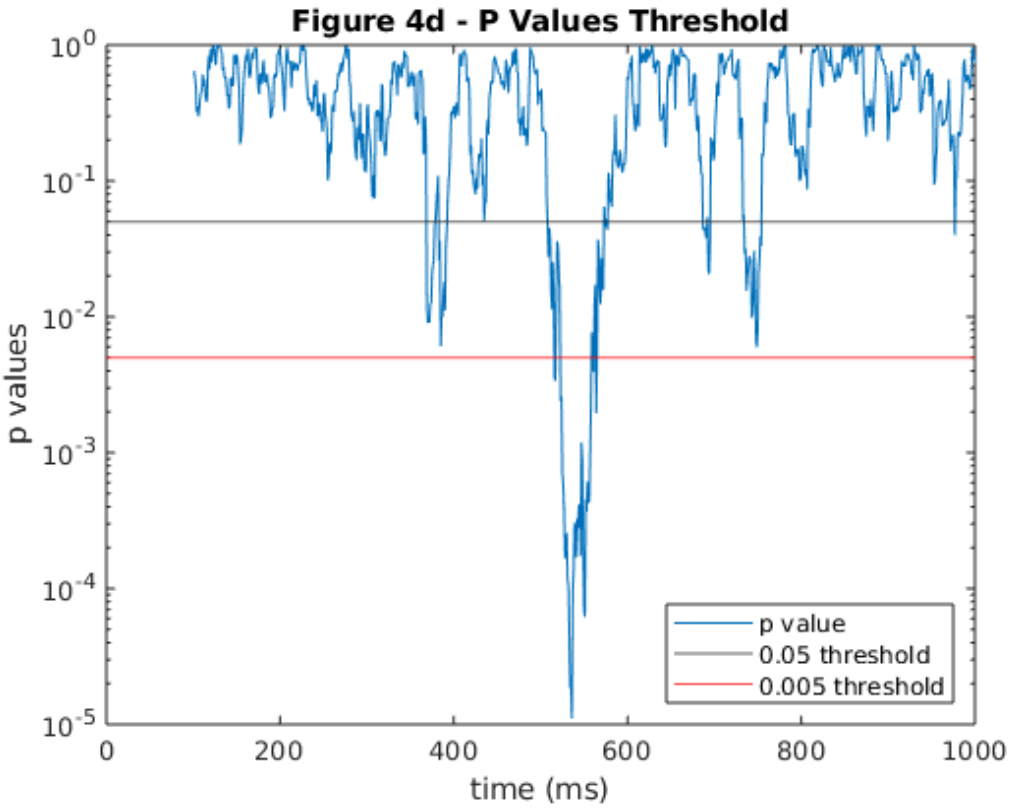
Figure 4c - Two-Sample F-Test P Values
p values
time (ms)
p value
1.00 second
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0
200
400
600
800
1000

****

4d

```
figure_4d = figure;
semilogy(p,'DisplayName','p value');
hold on;
title('Figure 4d - P Values Threshold');
ylabel('p values');
xlabel('time (ms)');
xlim([0 1000]);
% xline(101,'--','Color','k','DisplayName','1.00 second');
yline(0.05,'DisplayName','0.05 threshold');
yline(0.005,'Color','r','DisplayName','0.005 threshold');
% yline(0.001,'Color','r');
legend('Location','southeast');
hold off;

fprintf('4d ********** \n');
fprintf('see figure 4d\n');
fprintf('\n');
fprintf('0.05 is not small enough to filter the noise. 0.005 is better
 and filters out the noise\n');
fprintf('\n');
```

```
4d **********
see figure 4d

0.05 is not small enough to filter the noise. 0.005 is better and
 filters out the noise
```



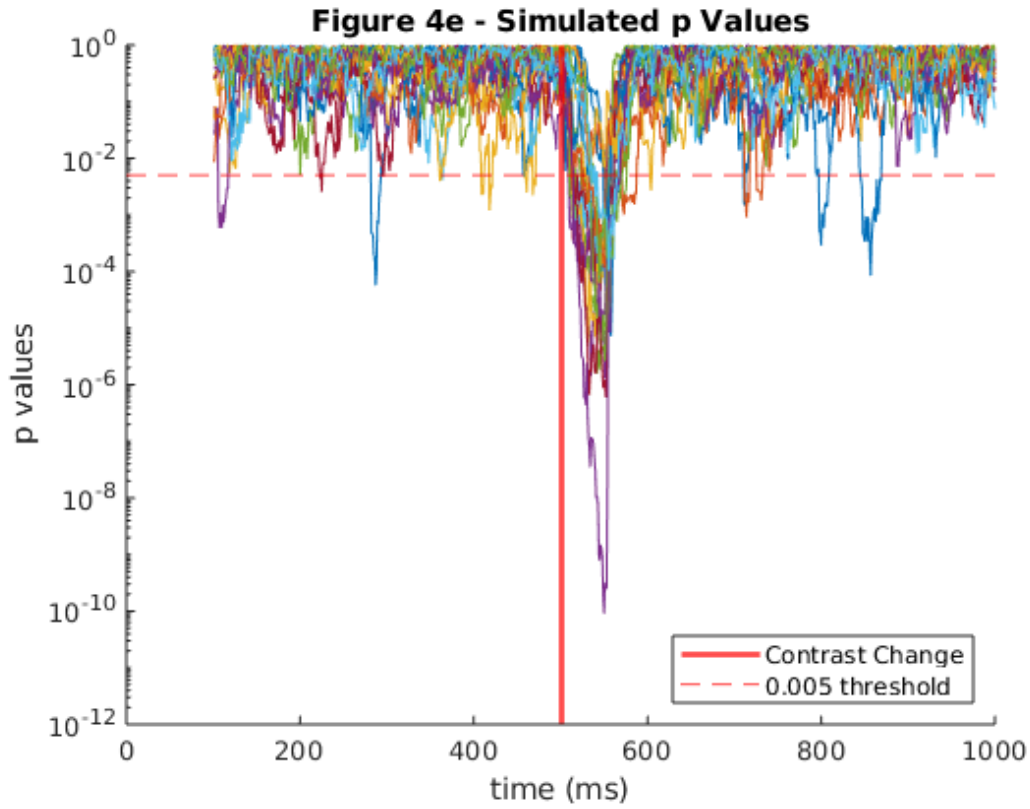
****

4e

```
% initialize seed
rng('default');

sim = zeros(20,1);   % each row is a simulation
p = zeros(1000,1);

figure_4e = figure;
axes_4e = axes('Parent',figure_4e);
hold on;

for i = 1:20

    % set seed
    rng(i);

    % find_lat performs steps a-c, returns the latency
    [sim(i), p] = find_lat();
    plot(p);

end

title('Figure 4e - Simulated p Values');
```

```
ylabel('p values');
xlabel('time (ms)');
ylabel('p values');
xlim([0 1000]);
leg(1) = xline(500,'LineWidth',2,'Color','r','DisplayName','Contrast
 Change');
leg(2) = yline(0.005,'--','Color','r','DisplayName','0.005
 threshold');
set(axes_4e,'YScale','log');
legend(leg,'Location','southeast');
hold off;

mean_sim = mean(sim);
se_sim= std(sim)/sqrt(length(sim));

fprintf('4e ********** \n');
fprintf('see figure 4e\n');
fprintf('\n');
fprintf('Latency Mean:\t%.2f ms\n', mean_sim);
fprintf('Latency Error:\t %.2f ms\n', se_sim);
fprintf('\n');
```

```
4e **********
see figure 4e

Latency Mean: 538.15 ms
Latency Error:  10.35 ms
```



****

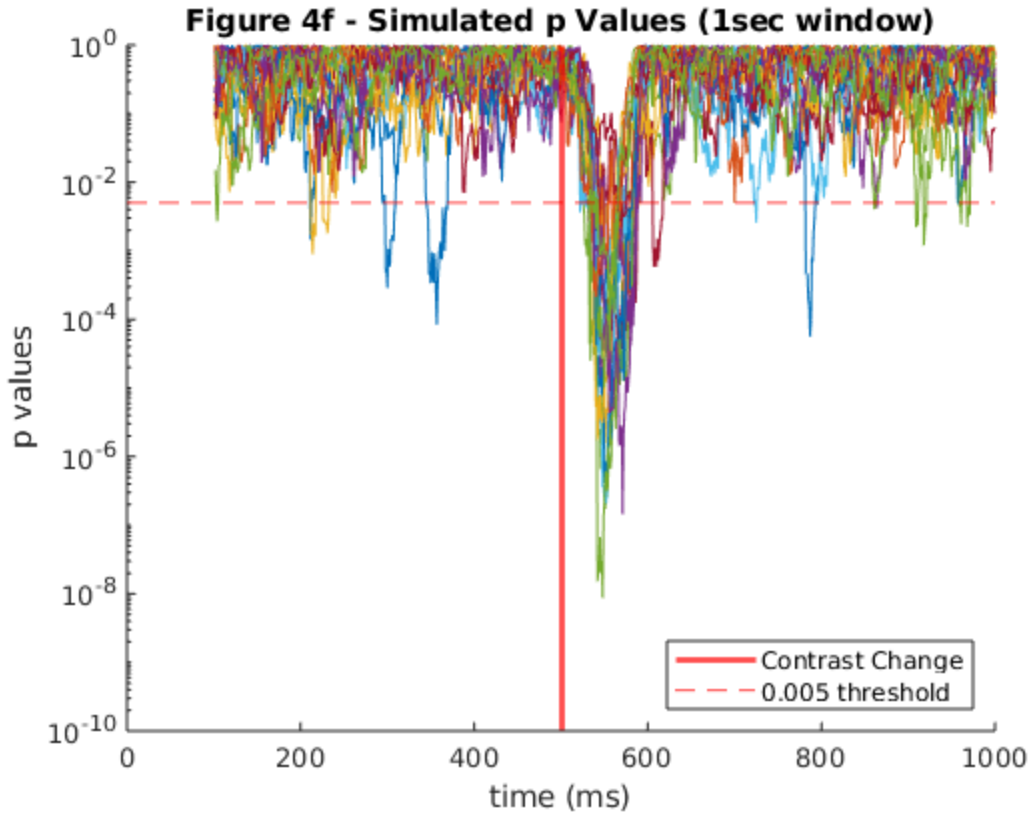
4f

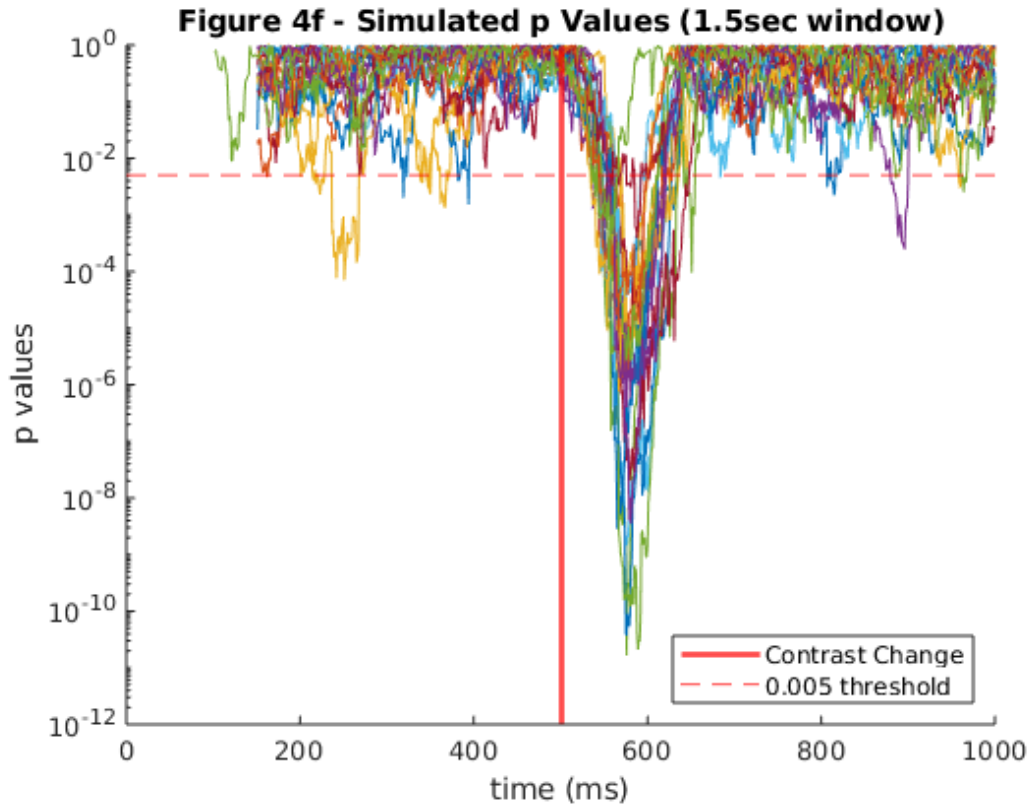
```
sim = zeros(20,1);   % each row is a simulation

% Sample interval 1s
figure_4f = figure;
axes_4f = axes('Parent',figure_4f);
hold on;

for i = 1:20

    % set seed
    rng(i);

    % find_lat performs steps a-c, returns the latency
    sim(i) = find_lat_rev();
    plot(p);

end

mean_sim_rev = mean(sim);
se_sim_rev= std(sim)/sqrt(length(sim));

title('Figure 4f - Simulated p Values (1sec window)');
```

```
ylabel('p values');
xlabel('time (ms)');
ylabel('p values');
xlim([0 1000]);
leg(1) = xline(500,'LineWidth',2,'Color','r','DisplayName','Contrast
 Change');
leg(2) = yline(0.005,'--','Color','r','DisplayName','0.005
 threshold');
set(axes_4f,'YScale','log');
legend(leg,'Location','southeast');
hold off;



% Sample interval 1.5s
figure_4fa = figure;
axes_4fa = axes('Parent',figure_4fa);
hold on;

for i = 1:20

    % set seed
    rng(i);

    % find_lat performs steps a-c, returns the latency
    sim(i) = find_lat_rev_long();
    plot(p);

end

mean_sim_rev_long = mean(sim);
se_sim_rev_long = std(sim)/sqrt(length(sim));

title('Figure 4f - Simulated p Values (1.5sec window)');
ylabel('p values');
xlabel('time (ms)');
ylabel('p values');
xlim([0 1000]);
leg(1) = xline(500,'LineWidth',2,'Color','r','DisplayName','Contrast
 Change');
leg(2) = yline(0.005,'--','Color','r','DisplayName','0.005
 threshold');
set(axes_4fa,'YScale','log');
legend(leg,'Location','southeast');
hold off;

fprintf('4f ********** \n');
fprintf('see figures 4f\n');
fprintf('\n');
fprintf('Latency (reversed) Mean:\t%.2f ms\n', mean_sim_rev);
fprintf('Latency (reversed) Error:\t %.2f ms\n', se_sim_rev);
fprintf('\n');
fprintf('Latency (reversed, long) Mean:\t%.2f ms\n',
 mean_sim_rev_long);
```

```
fprintf('Latency (reversed, long) Error:\t %.2f ms\n',
 se_sim_rev_long);
fprintf('\n');

fprintf('There seems to be no significant change in p-values with the
 contrasts reversed.\n');
fprintf('Changing the sample interval did change the P-values, with a
 lower latency and error\n');
fprintf('\n');
```

```
4f **********
see figures 4f

Latency (reversed) Mean: 539.45 ms
Latency (reversed) Error:  2.07 ms

Latency (reversed, long) Mean: 505.60 ms
Latency (reversed, long) Error:  0.93 ms

There seems to be no significant change in p-values with the contrasts
 reversed.
Changing the sample interval did change the P-values, with a lower
 latency and error
```





****

4g

```
fprintf('4g ********** \n');
fprintf('Theres no significant change in p-values in regards to the
 contrast order\n');
fprintf('However, changing the window did (and lowered the latency and
 error).\n');
fprintf('This suggests that the signal is a bit noisy for the sample
 interval\n');
fprintf('and that there is probably an optimal speed. Too large of a
 sample window\n');
fprintf('and you dont notice the change. Too small of a window makes
 it too sensitive\n');
fprintf('to noise.\n');
fprintf('ie. climate vs weather\n');
fprintf('\n');
```

```
4g **********
Theres no significant change in p-values in regards to the contrast
 order
However, changing the window did (and lowered the latency and error).
This suggests that the signal is a bit noisy for the sample interval
and that there is probably an optimal speed. Too large of a sample
 window
```

and you dont notice the change. Too small of a window makes it too sensitive
to noise.
ie. climate vs weather

Problem Set 5

```
****************************************** 5a

fprintf('\n');
fprintf('********** Problem Set 5 **********\n');
fprintf('\n');

fprintf('5a ********** \n');

% I just learned that this is a thing..
fprintf(['1) Threshold image\n',...
'\tValue: 1160\n',...
'\n',...
'2) Watershed image\n',...
'3) Analyze Image\n',...
'\tSize: 10-infinity\n',...
'\tCircularity: 0.00-1.00\n',...
'\t[x]Display Results\n',...
'\t[x]Summarize\n',...
'\t[x]Add to Manager\n',...
'\t[x]Exclude Edges\n',...
'\n',...
'Number of Cells: 112\n',...
'\n']);
```

```
********** Problem Set 5 **********

5a **********
1) Threshold image
 Value: 1160

2) Watershed image
3) Analyze Image
 Size: 10-infinity
 Circularity: 0.00-1.00
 [x]Display Results
 [x]Summarize
 [x]Add to Manager
 [x]Exclude Edges

Number of Cells: 112
```

****

5b

```
fprintf('5b ********** \n');
fprintf(['\n',...
'see 5b.csv and 5b.png\n',...
'\n']);
```

```
5b **********

see 5b.csv and 5b.png
```

****

5c

```
% #, Area, Mean, Min, Max, X, Y
data_sacs = readtable('5b.csv','HeaderLines',1);
data_sacs = table2array(data_sacs);

mean_sacs = mean(data_sacs(:,2));
sigma_sacs = std(data_sacs(:,2));

jb_result = jbtest(data_sacs(:,2));

fprintf('5c ********** \n');
fprintf(['\n',...
'The cells on the edge do not count towards a valid area estimate.
\n',...
'Being cut off, they dont represent their true shape.\n',...
'\n',...
'Mean Area (microns):\t%.2f\n',...
'Std Dev (microns):\t%.2f\n',...
'\n',...
'Jarque-Bera Test tests normality of a dataset.\n',...
'Result:\t%.0f\n',...
'The test rejects the hypothesis that the data is normally
 distributed.\n'],...
mean_sacs,sigma_sacs,jb_result);
fprintf('\n');
```

```
5c **********

The cells on the edge do not count towards a valid area estimate.
Being cut off, they dont represent their true shape.

Mean Area (microns): 78.52
Std Dev (microns): 11.98

Jarque-Bera Test tests normality of a dataset.
```

Result: 1
The test rejects the hypothesis that the data is normally distributed.

****

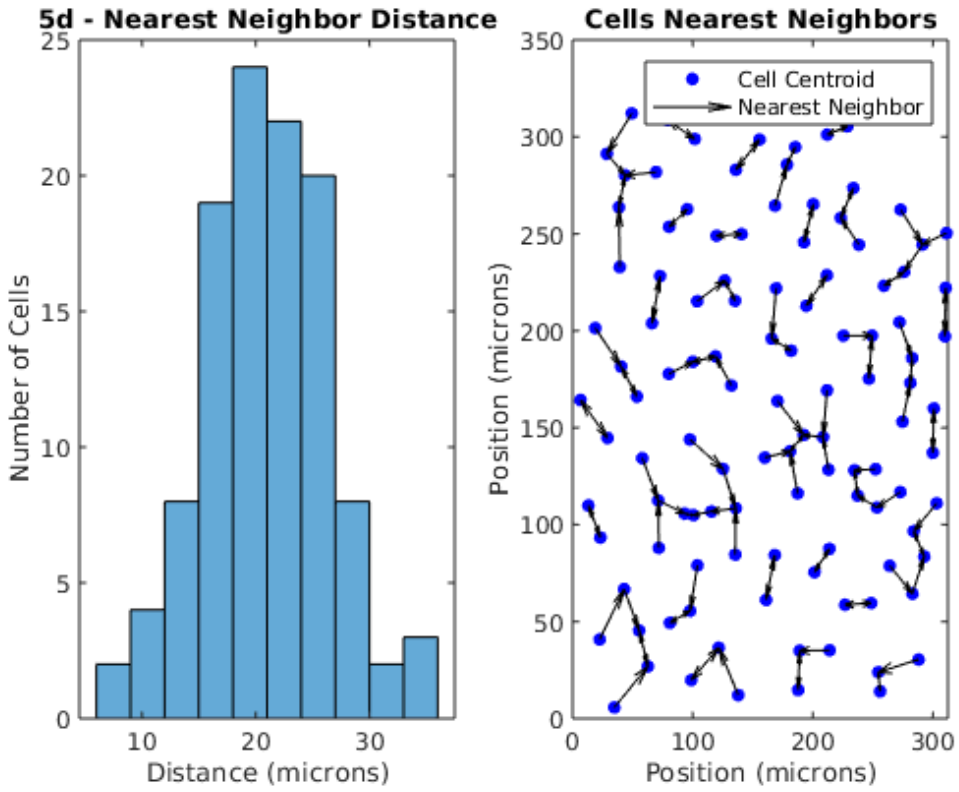
5d

```
X = [data_sacs(:,6) data_sacs(:,7)];

for i = 1:length(data_sacs)
    % The nearest neighbor of each point is itself
    % The 2nd nearest neighbor is the closest point that isn't itself

    Y = [data_sacs(i,6) data_sacs(i,7)];
    [idx(i,:), d(i,:)] = knnsearch(X, Y, 'K', 2);
end

figure_5d = figure;
subplot(1,2,1,'Parent',figure_5d);
histogram(d(:,2));
title('5d - Nearest Neighbor Distance');
xlabel('Distance (microns)');
ylabel('Number of Cells');

% This plots the centroid of each cell and points to its nearest
 neighbor
% Double arrows incate that both points are each other's nearest
 neighbor
X = [data_sacs(:,6), data_sacs(:,7)];
[i d]=knnsearch(X,X,'K',2);
i = i(:,2);

subplot(1,2,2,'Parent',figure_5d);
plot(X(:,1), X(:,2), 'b.','MarkerSize', 15,'DisplayName','Cell
 Centroid');
hold on;
quiver(X(:,1)', X(:,2)', X(i,1)' - X(:,1)', X(i,2)' - X(:,2)',
 0, 'k','DisplayName','Nearest Neighbor');
title('Cells Nearest Neighbors');
xlabel('Position (microns)');
ylabel('Position (microns)');
legend;
hold off;

co_var = std(d(:,2))/mean(d(:,2));

fprintf('5d ********** \n');
fprintf(['\n',...
'see figure 5d\n',...
'\n',...
'Coefficient of Variation:\t%.2f\n'],co_var);
fprintf('\n');
```

*5d ***********

see figure 5d

Coefficient of Variation: 0.26



****

5e

```
co_var = mean(d(:,2))^(-1/2);

fprintf('5e ********** \n');
fprintf(['\n',...
'This sort of spatial (cellular) development follows a Poisson
 distribution.\n',...
'The coefficient of variation is defined as #^(-1/2). If we define #
 to be the\n',...
'the mean of the distance, the coefficient of variation is %.2f, very
 close to\n',...
'the coefficient found in part d. This suggests the default assumption
 was wrong.\n'],co_var);
fprintf('\n');
```

*5e ***********

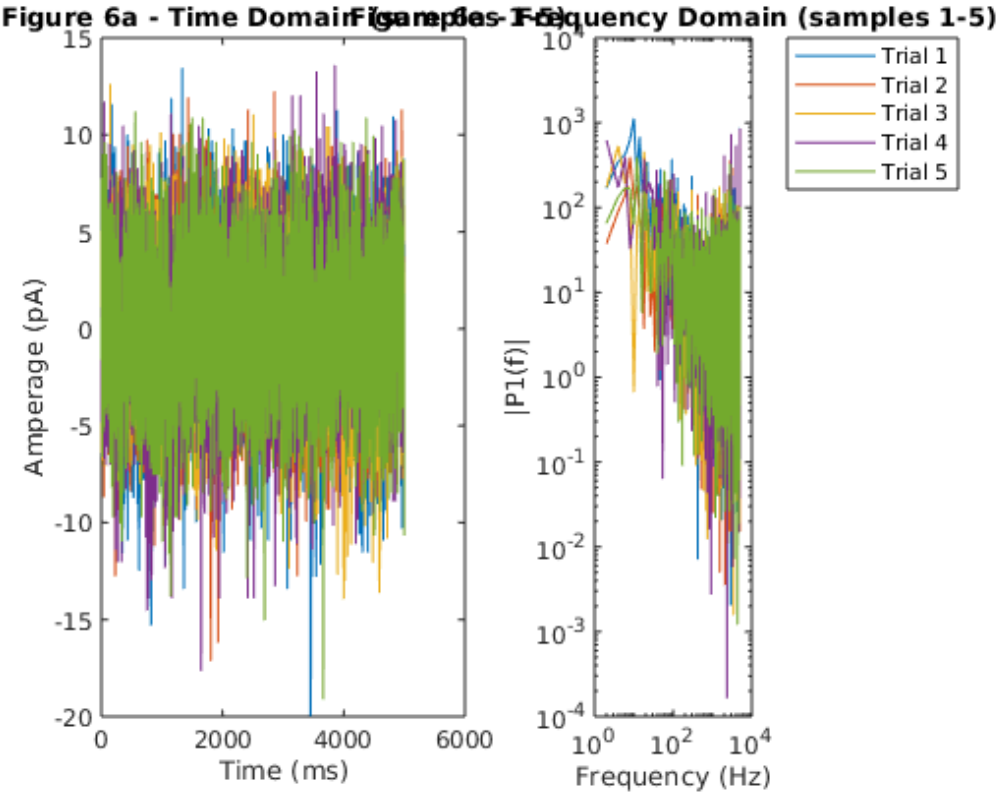
```
This sort of spatial (cellular) development follows a Poisson
 distribution.
The coefficient of variation is defined as #^(-1/2). If we define # to
 be the
the mean of the distance, the coefficient of variation is 0.22, very
 close to
the coefficient found in part d. This suggests the default assumption
 was wrong.
```

Problem Set 6

```
****************************************** 6a

fprintf('\n');
fprintf('********** Problem Set 6 **********\n');
fprintf('\n');

load('synapticNoiseData.mat');

fs = 10000;
t = 0.5;
timeaxis = 0:1/fs:((fs*t)-1)/fs;

[freqAxis, power] = powerSpectrum(dataMatrix(:,1:5), fs);

figure_6a = figure;
subplot(1,2,1,'Parent',figure_6a);
plot(dataMatrix(:,1:5));
hold on;
title('Figure 6a - Time Domain (samples 1-5)');
xlabel('Time (ms)');
ylabel('Amperage (pA)');

subplot(1,2,2,'Parent',figure_6a);
for i = 1:5
    [freqAxis, power] = powerSpectrum(dataMatrix(:,i), fs);
    txt = ['Trial ',num2str(i)];
    loglog(freqAxis, power, 'DisplayName',txt);
    hold on;
end
title('Figure 6a - Frequency Domain (samples 1-5)');
xlabel('Frequency (Hz)');
ylabel('|P1(f)|');

% theres a better way to do this legend, I want to work the rest of
 the
% final before addressing this.
legend('Location','bestoutside');

fprintf('6a ********** \n');
fprintf('\n');
```

```
fprintf('see figure 6a\n');
fprintf('\n');
fprintf('The Nyquist frequency is %0.f Hz\n',fs/2);
fprintf('Theres a peak at 54 Hz, which seems appropriate\n');
fprintf('\n');
```

```
********** Problem Set 6 **********

6a **********

see figure 6a

The Nyquist frequency is 5000 Hz
Theres a peak at 54 Hz, which seems appropriate
```



****

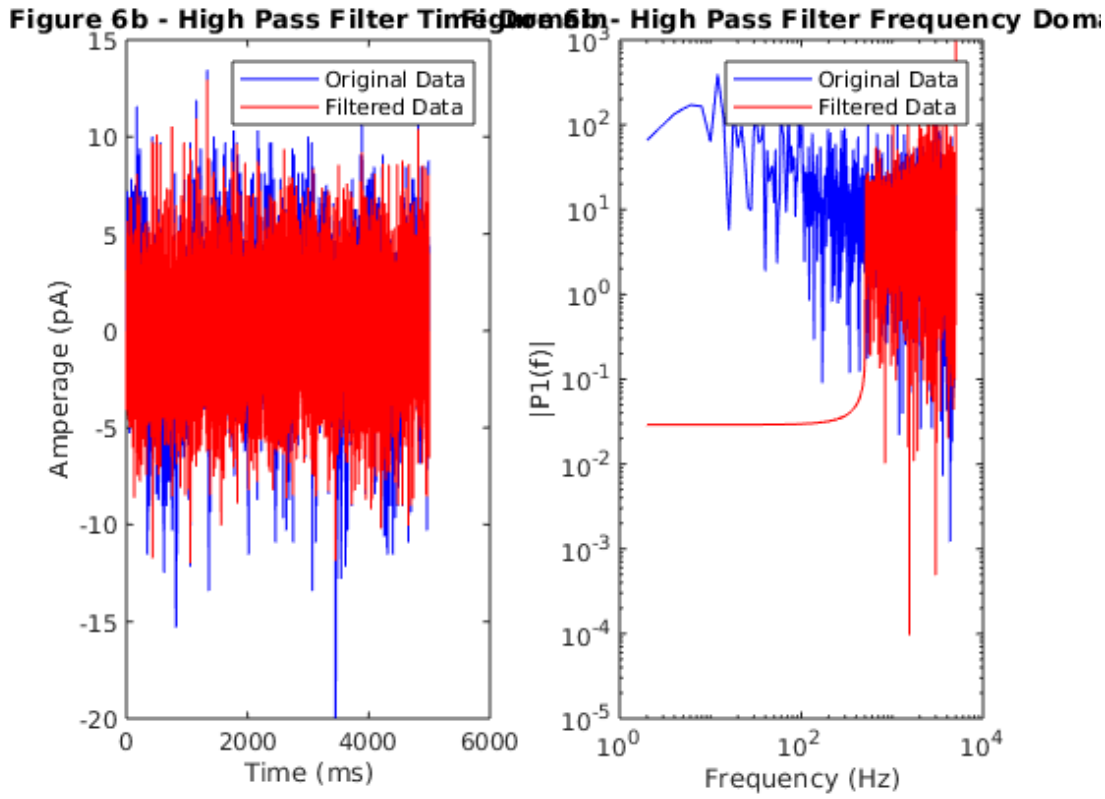
6b

```
load('filt_500.mat');

% -------------------------+
% Filter Parameters        |
%                          |
```

```
% Response Type              |
%   Highpass                 |
%                            |
% Design Method              |
%   FIR: Equiripple          |
%                            |
% Filter Order               |
%   Minimum order            |
%                            |
% Options                    |
%   Density Factor: 20       |
%                            |
% Frequency                  |
%   Units:  Hz               |
%   Fs:     10000            |
%   Fstop:  490              |
%   Fpass:  510              |
%                            |
% Magnitude                  |
%   Units: dB                |
%   Astop: 80                |
%   Apass: 1                 |
%                            |
% ---------------------------+

data_filt_b = filtfilt(filt_500.Numerator,1,dataMatrix);
[freqAxis_filt, power_filt] = powerSpectrum(data_filt_b, fs);

figure_6b = figure;
subplot(1,2,1,'Parent',figure_6b);
plot(dataMatrix(:,1),'b','DisplayName','Original Data');
hold on;
plot(data_filt_b(:,1),'r','DisplayName','Filtered Data');
title('Figure 6b - High Pass Filter Time Domain');
xlabel('Time (ms)');
ylabel('Amperage (pA)');
legend;

subplot(1,2,2,'Parent',figure_6b);
loglog(freqAxis, power,'b','DisplayName','Original Data');
hold on;
loglog(freqAxis_filt, power_filt,'r','DisplayName','Filtered Data');
title('Figure 6b - High Pass Filter Frequency Domain');
xlabel('Frequency (Hz)');
ylabel('|P1(f)|');
legend;

fprintf('6b ********** \n');
fprintf('\n');
fprintf('see figure 6b\n');
fprintf('\n');
```

```
6b **********
```

see figure 6b



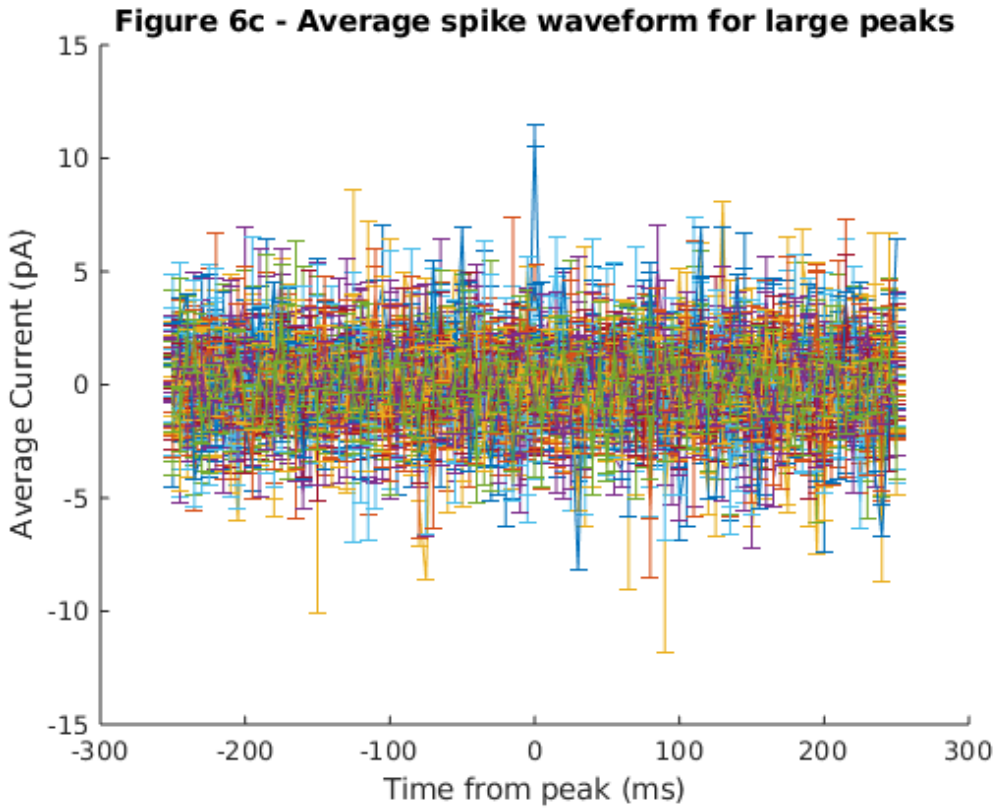
****

6c

```
T = 0.005;  % period
fs = 1/T;   % Sample Freq
window = -50:50;
window_length = length(window);
timeAxis_waveform = 1E3*window*T;
trials = 40;

count = 0;

figure_6c = figure;
hold on;
for i = 1:trials

    % Get the mean and s.e.m. of the spikeWaveformMatrix and plot them
    [pks,locs] = findpeaks(data_filt_b(1:5000,i),'MinPeakHeight',10);
    n = length(pks);
    spikeWaveformMatrix = zeros(n, window_length);

    for j=1:n
```

```
        % window buffer
        if (locs(j) > 51) && (locs(j) < 4950)
            spikeWaveformMatrix(j,:) = data_filt_b(locs(j)+window);
            count = count + 1;
        end
    end

    spikeWaveform_mean = mean(spikeWaveformMatrix, 1);
    spikeWaveform_sem = std(spikeWaveformMatrix, [], 1) ./ sqrt(n);
    errorbar(timeAxis_waveform, spikeWaveform_mean,
 spikeWaveform_sem);

end

xlabel('Time from peak (ms)');
ylabel('Average Current (pA)');
title('Figure 6c - Average spike waveform for large peaks');
hold off;

fprintf('6c ********** \n');
fprintf('\n');
fprintf('see figure 6c\n');
fprintf('\n');
fprintf('In %0.0f trials there were %0.f events\n',trials,count);
fprintf('\n');
```

```
6c **********

see figure 6c

In 40 trials there were 208 events
```



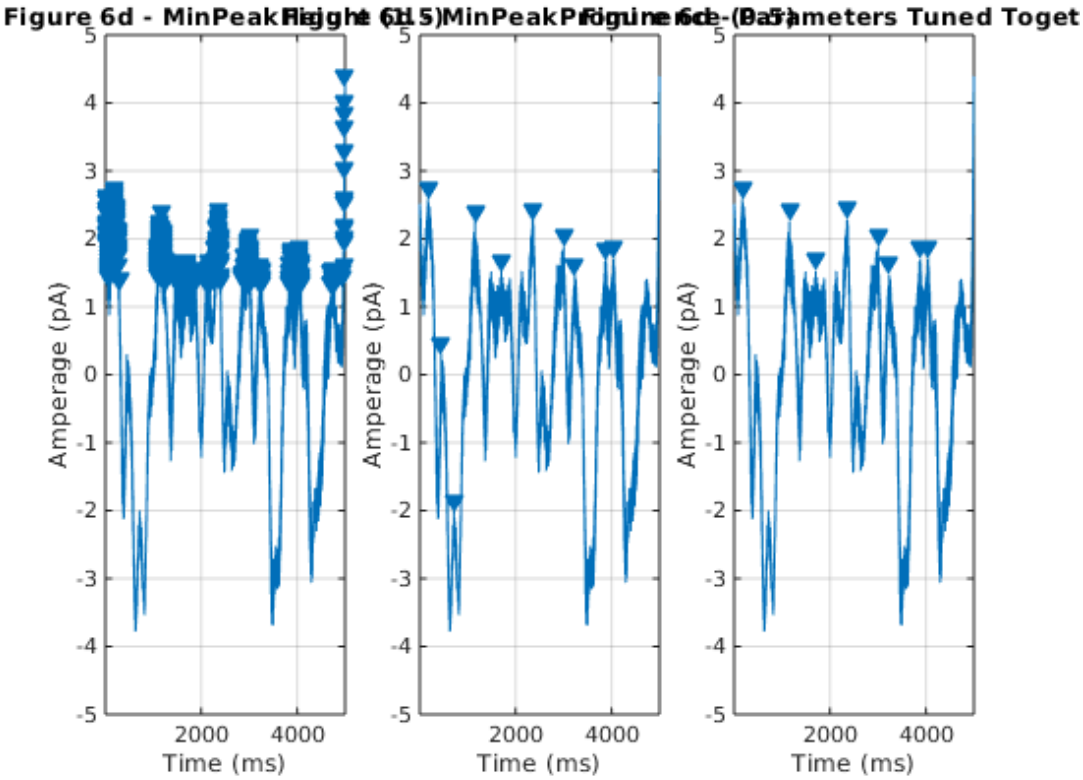
****

6d

```
% used trial 1, had a good spike, not too big.

trial = 1;
min_peak_height = 1.2;
min_peak_prom = 1.5;
count2 = 0;

dataTrace = -dataMatrix(:,trial);
[pks,locs] =
 findpeaks(dataTrace,'MinPeakHeight',min_peak_height,'MinPeakProminence',min_peak_
n = length(locs);
spikeWaveformMatrix = zeros(n, window_length);

for i=1:n
    % window buffer
    if (locs(i) > 51) && (locs(i) < 4950)
        spikeWaveformMatrix(i,:) = dataTrace(locs(i)+window);
        count2 = count2 + 1;
    end
end

spikeWaveform_mean = mean(spikeWaveformMatrix, 1);
```

```
spikeWaveform_Filter = spikeWaveform_mean./sum(spikeWaveform_mean);
dataTrace_matchFiltered = filtfilt(spikeWaveform_Filter, 1,
 dataMatrix);

% min_peak_height = 1;
% min_peak_prom = .5;

figure_6d = figure;
subplot(1,3,1,'Parent',figure_6d);
findpeaks(dataTrace_matchFiltered(1:5000,trial),'MinPeakHeight',min_peak_height);
hold on;
title('Figure 6d - MinPeakHeight (1.5)');
xlabel('Time (ms)');
ylabel('Amperage (pA)');

subplot(1,3,2,'Parent',figure_6d);
findpeaks(dataTrace_matchFiltered(1:5000,trial),'MinPeakProminence',min_peak_prom)
hold on;
title('Figure 6d - MinPeakProminence (0.5)');
xlabel('Time (ms)');
ylabel('Amperage (pA)');

subplot(1,3,3,'Parent',figure_6d);
findpeaks(dataTrace_matchFiltered(1:5000,trial),'MinPeakHeight',min_peak_height,'M
hold on;
title('Figure 6d - Parameters Tuned Together');
xlabel('Time (ms)');
ylabel('Amperage (pA)');

fprintf('6d ********** \n');
fprintf('\n');
fprintf('see figure 6d\n');
fprintf('\n');
fprintf('For this trial, a MinPeakHeight of %0.1f seemed reasonable.
 It captured\n',min_peak_height);
fprintf('significant spikes above some of the noise. Likewise, a
 MinPeakProminence\n');
fprintf('of %0.1f filtered smaller local spikes from the greater
 signal spike.\n',min_peak_prom);
fprintf('\n');
```

```
6d **********

see figure 6d

For this trial, a MinPeakHeight of 1.2 seemed reasonable. It captured
significant spikes above some of the noise. Likewise, a
 MinPeakProminence
of 1.5 filtered smaller local spikes from the greater signal spike.
```



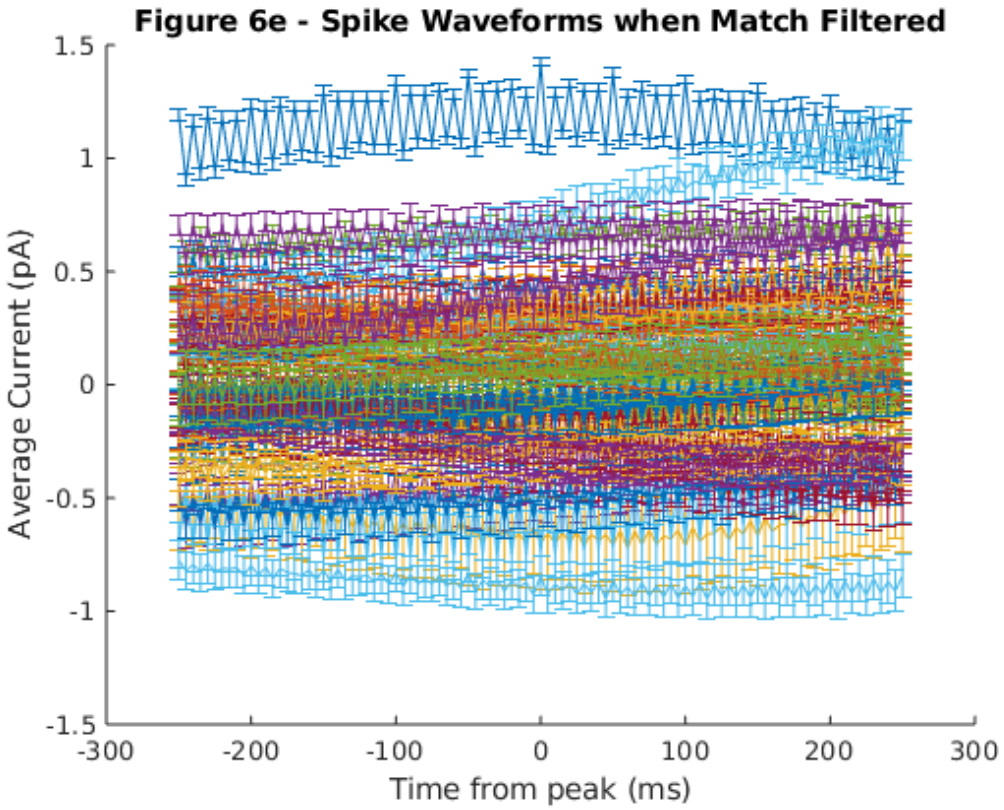
****

6e

```
pks_tot = 0;
min_peak_height = 1;
min_peak_prom = 0.2;

figure_6e = figure;
hold on;
for i = 1:trials

    [pks,locs] =
 findpeaks(dataTrace_matchFiltered(1:5000,i),'MinPeakHeight',min_peak_height,'MinP

    n = length(pks);
    spikeWaveformMatrix = zeros(n, window_length);

    for j=1:n
        % window buffer
        if (locs(j) > 51) && (locs(j) < 4950)
            spikeWaveformMatrix(j,:) =
 dataTrace_matchFiltered(locs(j)+window);
            pks_tot = pks_tot + 1;
        end
    end
```

```
    spikeWaveform_mean = mean(spikeWaveformMatrix, 1);
    spikeWaveform_sem = std(spikeWaveformMatrix, [], 1) ./ sqrt(n);
    errorbar(timeAxis_waveform, spikeWaveform_mean,
 spikeWaveform_sem);

end

title('Figure 6e - Spike Waveforms when Match Filtered');
xlabel('Time from peak (ms)');
ylabel('Average Current (pA)');

fprintf('6e ********** \n');
fprintf('\n');
fprintf('see figure 6e\n');
fprintf('\n');
fprintf(['In %0.0f trials there were %0.f events. In part c there were
 %0.f events\n',...
'See figures 6c and 6e to compare the plots. Figure 6c is noisy and
 erratic,\n',...
'an indication that peak heights arent a good way of filtering the
 signal.\n',...
'Figure 6e shows clearer signal traces for each trial.
\n'],trials,pks_tot,count);
```

```
6e **********

see figure 6e

In 40 trials there were 12104 events. In part c there were 208 events
See figures 6c and 6e to compare the plots. Figure 6c is noisy and
 erratic,
an indication that peak heights arent a good way of filtering the
 signal.
Figure 6e shows clearer signal traces for each trial.
```



Published with MATLAB® R2019b