Investigating brain activity in a model with a specific gene removed.

In this project, we will investigate the brain activity from a model with a specific gene removed. The purpose of this scientific research was to gain insight into the role of the gene in brain functions. We will investigate the characteristics of brain activity by analysing the data we extracted from video files, described in part 1. This project forms a part of my PhD thesis, in which the complete content can be found in the University of Auckland Doctoral Thesis repository.

In this project, I want to answer the question: *Did the characteristics*[1] *of brain activity (frequency, amplitude and duration) differ when the specific gene was removed*?

The data we will analyse was extracted from video files that were described in part 1. In brief, the data were video files of brain activity from a model where a specific gene important for brain cells communicating with each other was removed. This investigation will give an insight into the role of the gene in brain functions.

To assess if differences were observed in the model with the gene removed, we need to compare it to a control (normal/unaltered) model. In biology, we call the model without a gene a knockout (KO) and the control a wildtype (WT). We will compare the signal characteristics between the knockout and the wildtype to see if the gene affects brain activity.

All processing and analyses were done on MATLAB. In MATLAB, the core Python libraries (numpy, pandas and matplotlib) are built into the environment, and therefore we do not need to import these libraries. However, we will be needing specialised functions from the *Math, Statistics and Optimisation* toolbox.

[1]In this scientific study, the term *frequency* is defined as how often a brain signal occurs over a defined period; *amplitude* is how intense the brain signal was and *duration* is how long the brain signal lasted.

Data analysis:

Load the data set (saved as a CSV file) into MATLAB and remove all the NaNs in the table. We will then separate the data into two groups: WT and KO.

[1]

```
>> signal_table = readtable(['Experiment/SignalCharacteristics.csv']);
>>
>> % data cleaning: remove all rows with NaNs
>> no_data_row = isnan(signal_table.Frequency);
>> signal_table(no_data_row,:) = [];
>>
>> % separate into WT and KO
>> WT_row = signal_table.Genotype == "WT";
>> WT = signal_table(WT_row,:);
>> KO = signal_table(WT_row==0,:);
>>
>> WT
>> KO
```

```
WT =

  64×5 table

      ID          Genotype    Frequency    Amplitude    Duration
    ________    ________    _________    _________    ________

    {'AC12'}    {'WT'}       0.19375       9.9501       1.9345
    {'AC12'}    {'WT'}       0.24375       6.5289       1.6714
    {'AC12'}    {'WT'}       0.23333       6.8631       1.7027
    {'AC12'}    {'WT'}       0.25417       6.4024       1.6696
    {'AC12'}    {'WT'}       0.23542       6.4834       1.6124
    {'AC12'}    {'WT'}       0.22917       7.2542       1.7614
    {'AC17'}    {'WT'}        0.2875       1.9522       1.1845
    {'AC17'}    {'WT'}           0.3        2.129       1.0955
    {'AC17'}    {'WT'}       0.29583       1.8156       1.1544
    {'AC17'}    {'WT'}       0.30833       1.3622        1.007
    {'AC17'}    {'WT'}       0.31667       1.3631      0.90533
    {'AC22'}    {'WT'}       0.09375       2.3779       1.7285
    {'AC22'}    {'WT'}       0.11667        2.685        1.721

       :           :            :            :            :

    {'AC49'}    {'WT'}       0.34375      0.59987      0.87158
    {'AC49'}    {'WT'}       0.34167      0.59358      0.94856
    {'AC57'}    {'WT'}       0.32917      0.39205       1.0858
    {'AC57'}    {'WT'}       0.34375       0.4215       1.0968
    {'AC57'}    {'WT'}       0.34583      0.36538       1.0626
    {'AC58'}    {'WT'}       0.25833       1.5732       1.4416
    {'AC58'}    {'WT'}       0.26875       1.2857       1.3212
    {'AC58'}    {'WT'}       0.30417      0.90166       1.2017
    {'AC58'}    {'WT'}       0.30625      0.94523       1.3319
    {'AC63'}    {'WT'}          0.25       1.3891       1.1837
    {'AC63'}    {'WT'}       0.23333       1.9103       1.3868
    {'AC63'}    {'WT'}       0.25625       1.3527       1.2848
    {'AC63'}    {'WT'}       0.21458       1.5679       1.3412

  Display all 64 rows.
```

KO =

59×5 table

ID	Genotype	Frequency	Amplitude	Duration
{'AC15'}	{'KO'}	0.24792	1.7429	1.3862
{'AC15'}	{'KO'}	0.22292	1.6338	1.8623
{'AC15'}	{'KO'}	0.24375	1.1604	1.5373
{'AC15'}	{'KO'}	0.23125	1.2076	1.7371
{'AC29'}	{'KO'}	0.19792	1.9175	1.5043
{'AC29'}	{'KO'}	0.1875	1.3353	1.5607
{'AC29'}	{'KO'}	0.17292	1.5578	1.6518
{'AC31'}	{'KO'}	0.10417	8.307	1.8468
{'AC31'}	{'KO'}	0.083333	7.0701	2.224
{'AC31'}	{'KO'}	0.1375	4.6403	1.742
{'AC31'}	{'KO'}	0.16667	4.2291	1.5483
{'AC31'}	{'KO'}	0.16875	6.0357	1.7693
{'AC31'}	{'KO'}	0.14167	9.9359	2.3054
:	:	:	:	:
{'AC61'}	{'KO'}	0.24375	0.40196	1.1165
{'AC61'}	{'KO'}	0.24375	0.53004	1.2786
{'AC61'}	{'KO'}	0.24583	0.53286	1.3233
{'AC61'}	{'KO'}	0.24792	0.49692	1.4064
{'AC61'}	{'KO'}	0.36667	0.84229	0.96841
{'AC61'}	{'KO'}	0.36875	0.76245	0.99239
{'AC61'}	{'KO'}	0.35417	0.58968	0.95255
{'AC61'}	{'KO'}	0.31875	0.5832	1.0496
{'AC62'}	{'KO'}	0.19583	3.2981	1.5778
{'AC62'}	{'KO'}	0.27917	2.7879	1.2004
{'AC62'}	{'KO'}	0.27292	3.4613	1.2519
{'AC62'}	{'KO'}	0.3	2.7499	1.0659
{'AC62'}	{'KO'}	0.30833	2.1683	1.0623

Display all 59 rows.

We will now make a comparison between the two groups. We will first test the assumption that the data are normally distributed using the Shapiro-Wilk test[2].

[2]

```
>> % Create a table to store the outcome
>> h_swtest = array2table(zeros(2,4));
>> h_swtest.Properties.VariableNames = {'Group','h_Frequency',...
    'h_amplitude','h_duration'};
>>
>> h_swtest = convertvars(h_swtest,'Group','cell');
>>
>> h_swtest{1,'Group'} = {'h_WT'};
>> h_swtest{2,'Group'} = {'h_KO'}
>>
>> % Shapiro-Wilk test
>> h_swtest{1,2} = swtest(WT{:,'Frequency'},0.05);
>> h_swtest{1,3} = swtest(WT{:,'Amplitude'},0.05);
>> h_swtest{1,4} = swtest(WT{:,'Duration'},0.05);
>>
>> h_swtest{2,2}= swtest(KO{:,'Frequency'},0.05);
>> h_swtest{2,3}= swtest(KO{:,'Amplitude'},0.05);
>> h_swtest{2,4}= swtest(KO{:,'Duration'},0.05);
>>
>> h_swtest
```

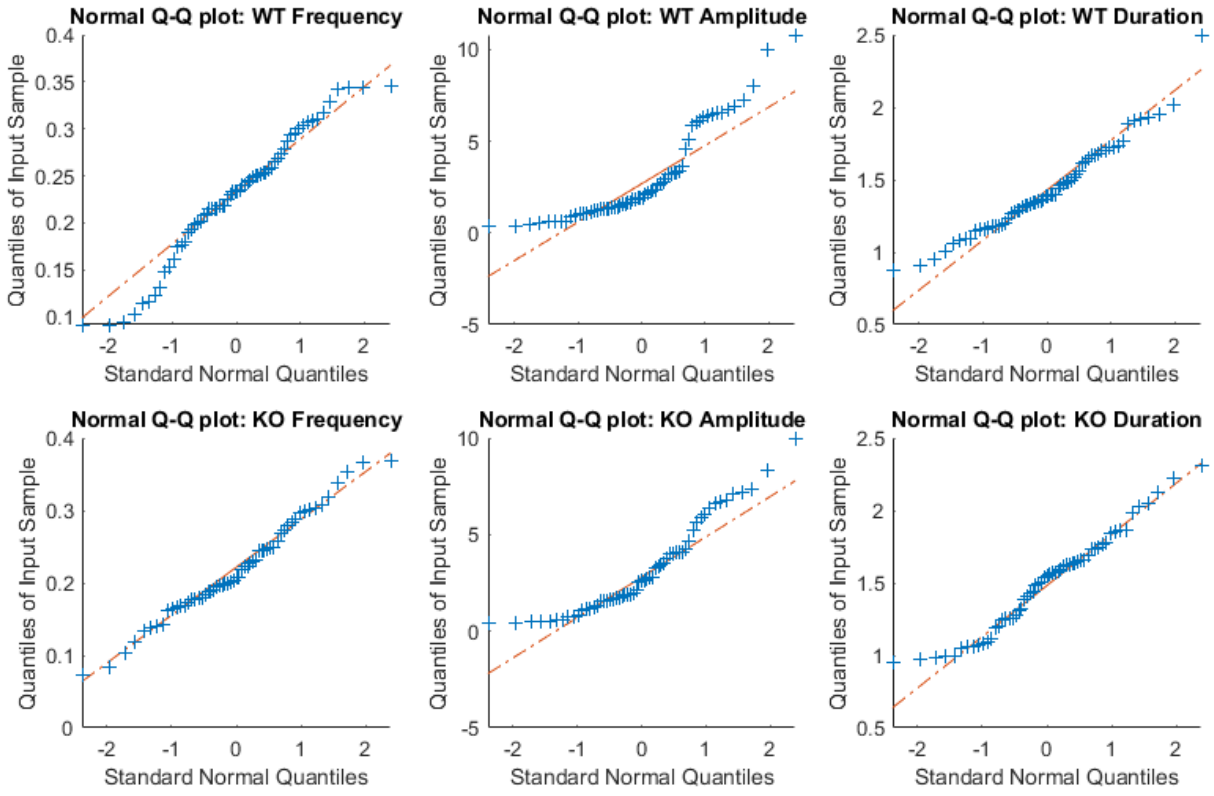
```
h_swtest =

  2×4 table

    Group       h_Frequency    h_amplitude    h_duration
    _________   ____________   ____________   __________

    {'h_WT'}          0              1             1
    {'h_KO'}          0              1             0
```

The output of 0 indicates that the null hypothesis, that the data are in a normal distribution, was not rejected. For the amplitude data, both groups showed an output of 1. For the duration data, only the KO group seems to follow a normal distribution. In conjunction with the Shapiro-Wilk test, we will also do a visual inspection of the data using a Q-Q plot to determine what statistical test to use for comparison.

[2] From a third-party script: Ahmed BenSaïda (2026). Shapiro-Wilk and Shapiro-Francia normality tests. https://au.mathworks.com/matlabcentral/fileexchange/13964-shapiro-wilk-and-shapiro-francia-normality-tests MATLAB Central File Exchange. Retrieved January 12, 2026.

[3]

```
>> figure
>> subplot(2,3,1)
>> qqplot(WT{:,'Frequency'})
>> title ('Normal Q-Q plot: WT Frequency')
>> subplot(2,3,2)
>> qqplot(WT{:,'Amplitude'})
>> title ('Normal Q-Q plot: WT Amplitude')
>> subplot(2,3,3)
>> qqplot(WT{:,'Duration'})
>> title ('Normal Q-Q plot: WT Duration')
>> subplot(2,3,4)
>> qqplot(KO{:,'Frequency'})
>> title ('Normal Q-Q plot: KO Frequency')
>> subplot(2,3,5)
>> qqplot(KO{:,'Amplitude'})
>> title ('Normal Q-Q plot: KO Amplitude')
>> subplot(2,3,6)
>> qqplot(KO{:,'Duration'})
>> title ('Normal Q-Q plot: KO Duration')
>> set(gcf,'color','white')
```



Based on the results from the QQ-plot and Shapiro-Wilk test, we will use a parametric test for the frequency data and a non-parametric test for the amplitude data. For the duration data, we will use a parametric test because the data for both the WT and KO group falls closely along the straight diagonal line with slight deviations at the tail end.

Descriptive statistics and comparison tests:

[4]

```
>> %% descriptive statistics
>> % Make a table to hold results
>> descrip_stats = array2table(zeros(6,6));
>> descrip_stats.Properties.VariableNames = {'Data','Genotype',...
       'n','mean','SD','SEM'};
>>
>> descrip_stats = convertvars(descrip_stats,{'Data','Genotype'},...
       'cell');
>>
>> descrip_stats{:,'Data'} = {'Frequency';'Frequency';...
       'Amplitude';'Amplitude';'Duration';'Duration'};
>> descrip_stats{:,'Genotype'} = repmat({'WT';'KO'},3,1);
>>
>> for vv = 1:size(descrip_stats,1);
       data_var = char(descrip_stats{vv,'Data'});
       group = char(descrip_stats{vv,'Genotype'});
       if group == "WT";
           descrip_stats{vv,'n'} = size(WT,1);
           descrip_stats{vv,'mean'} = mean(WT{:,data_var});
           descrip_stats{vv,'SD'} = std(WT{:,data_var});
           descrip_stats{vv,'SEM'} = (descrip_stats{vv, ...
               'SD'})/sqrt(descrip_stats{vv,'n'});
       else group == "KO";
           descrip_stats{vv,'n'} = size(KO,1);
           descrip_stats{vv,'mean'} = mean(KO{:,data_var});
           descrip_stats{vv,'SD'} = std(KO{:,data_var});
           descrip_stats{vv,'SEM'} = (descrip_stats{vv, ...
               'SD'})/sqrt(descrip_stats{vv,'n'});
       end
   end
>>
>> descrip_stats
```

```
descrip_stats =

  6×6 table
```

Data	Genotype	n	mean	SD	SEM
{'Frequency'}	{'WT'}	64	0.22917	0.06619	0.0082737
{'Frequency'}	{'KO'}	59	0.21917	0.067532	0.0087919
{'Amplitude'}	{'WT'}	64	2.9902	2.4984	0.3123
{'Amplitude'}	{'KO'}	59	3.1624	2.3346	0.30393
{'Duration' }	{'WT'}	64	1.4341	0.30413	0.038017
{'Duration' }	{'KO'}	59	1.5005	0.3381	0.044017

[5]

```
>> %% Statistical analysis
>> % Make a table to hold statistical results
>> stat_tests = array2table(zeros(3,7));
>> stat_tests.Properties.VariableNames = {'Data','Group',...
       'Test_name','p-value','sig','stats_type','stats'};
>>
>> stat_tests = convertvars(stat_tests,{'Data','Group', ...
       'Test_name','sig','stats_type'},'cell');
>>
>> stat_tests{:,'Data'} = {'Frequency';'Amplitude';'Duration'};
>> stat_tests{:,'Group'} = {'KO vs WT'};
>> stat_tests{:,'Test_name'} = {'T-test';'Mann-Whit';'T-test'};
>> stat_tests{:,'stats_type'} = {'T';'Z';'T'};
>>
>> %%%%% Two-way student T-test
>> [~,p(1,1),~,stats] = ttest2(WT{:,'Frequency'},KO{:,'Frequency'});
>> stat_tests{1,'stats'} = stats.tstat;
>>
>> %%%%% Wilcoxon rank sum test
>> [p(2,1),~,stats] = ranksum(WT{:,'Amplitude'},KO{:,'Amplitude'});
>> stat_tests{2,'stats'} = stats.zval;
>>
>> %%%%% Two-way student T-test
>> [~,p(3,1),~,stats] = ttest2(WT{:,'Duration'},KO{:,'Duration'});
>> stat_tests{3,'stats'} = stats.tstat;
>>
>> for ii = 1:3;
       if p(ii,1) <= 0.0001;
           stat_tests{ii,'sig'} = {'****'};
           stat_tests{ii,'p-value'} = p(ii,1);
       elseif p(ii,1) <= 0.001;
           stat_tests{ii,'sig'} = {'***'};
           stat_tests{ii,'p-value'} = p(ii,1);
       elseif p(ii,1) <= 0.01;
           stat_tests{ii,'sig'} = {'**'};
           stat_tests{ii,'p-value'} = p(ii,1);
       elseif p(ii,1) <= 0.05;
           stat_tests{ii,'sig'} = {'*'};
           stat_tests{ii,'p-value'} = p(ii,1);
       else p(ii,1) > 0.05;
           stat_tests{ii,'sig'} = {'ns'};
           stat_tests{ii,'p-value'} = p(ii,1);
       end
   end
>>
>> stat_tests
```

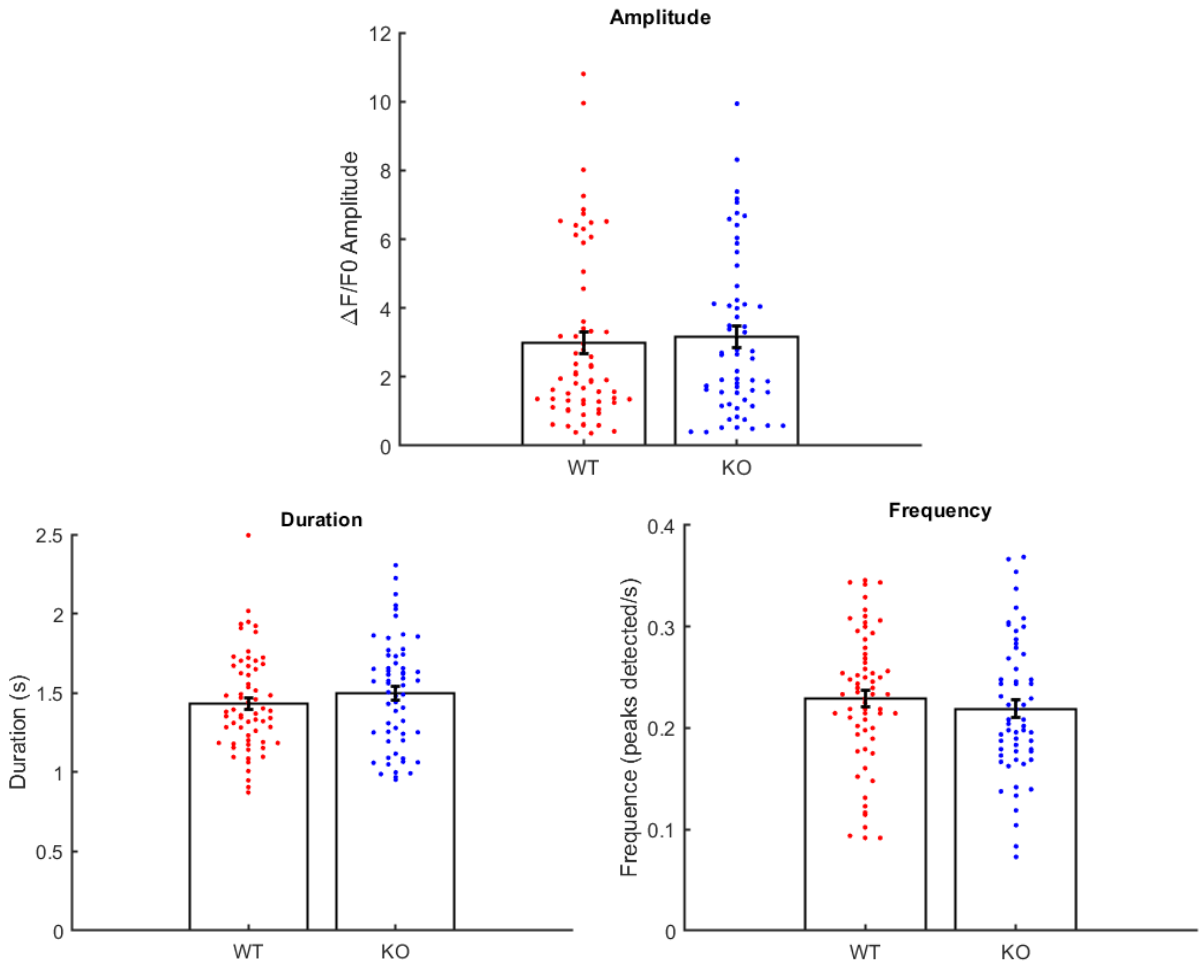
```
stat_tests =

  3×7 table
```

Data	Group	Test_name	p-value	sig	stats_type	stats
{'Frequency'}	{'KO vs WT'}	{'T-test' }	0.40907	{'ns'}	{'T'}	0.82841
{'Amplitude'}	{'KO vs WT'}	{'Wilcoxon'}	0.48005	{'ns'}	{'Z'}	-0.70622
{'Duration' }	{'KO vs WT'}	{'T-test' }	0.25386	{'ns'}	{'T'}	-1.1465

Now, let's get some insight into the role of the gene and brain activity.
[6]

```
>> % Making graphs and a table displaying data
>> datapoint = size(WT,1);
>> nbars = size(unique(descrip_stats.Genotype),1); % number of bars
>> ndata = unique(descrip_stats.Data)
>>
>> even_spaces = 0.1; % the width of bars in bar graph
>> sz = 10;
>>
>> for bb = 1:size(ndata,1);
       clearvars avg_data
       data_var = string(ndata{bb,1});

       data_row_WT = descrip_stats.Data == data_var & descrip_stats.Genotype == "WT";
       data_row_KO = descrip_stats.Data == data_var & descrip_stats.Genotype == "KO";

       avg_data(1,1) = descrip_stats(data_row_WT,'mean');
       avg_data(1,2) = descrip_stats(data_row_KO,'mean');
       avg_data(2,1) = descrip_stats(data_row_WT,'SEM');
       avg_data(2,2) = descrip_stats(data_row_KO,'SEM');
       avg_data = table2array(avg_data);

       figure
       b = bar(avg_data(1,:),'grouped','white','LineWidth',1.5);
       xticks([1,2]);
       xticklabels({'WT','KO'})

       % Overlaying each data point over the bar graph
       hold on
       for ii = 1:nbars;
           if ii == 1;
               d = discretize(table2array(WT(:,data_var)),50);
           else ii == 2;
               d = discretize(table2array(KO(:,data_var)),50);
           end

           x_points = NaN([size(d,1),size(d,2)]);

           for nn = 1:50;
               g = find(d == nn);
               amount_found = length(g);
               for xx = 1:amount_found;
                   lowest_limit = ii-((amount_found/2)+ 0.5)*even_spaces;
                   x_points(g(xx,1),1) = lowest_limit+(xx*even_spaces);
               end
           end

           if ii == 1;
               s = scatter(x_points,(table2array(WT(:,data_var))),sz,'red','filled');
           else ii == 2;
               s = scatter(x_points,(table2array(KO(:,data_var))),sz,'blue','filled');
           end
       end
       box off
       set(gcf, 'color','white')
       ax= gca;
       ax.LineWidth = 1.5;
       ax.FontSize = 13;
       title(data_var, 'FontSize', 13)
       hold on
```

```
        %%% error bars (Standard Error of the Mean; SEM)
        err_x = [1:1:nbars];
        e = errorbar(err_x,avg_data(1,:), avg_data(2,:), avg_data(2,:), ...
            'LineStyle','none','LineWidth',2,'Color',[0 0 0]);
    end
>>
>> figure(1)
>> ylabel('\DeltaF/F0 Amplitude')
>>
>> figure(2)
>> ylabel('Duration (s)')
>>
>> figure(3)
>> ylabel('Frequence (peaks detected/s)')
>>
>> %% Display data
>> % Create a table
>> display_data = array2table(zeros(3,4));
>> display_data.Properties.VariableNames = {' ','WT','KO','significance'}
>> display_data = convertvars(display_data,{' ','WT','KO','significance'},'cell');
>> display_data{:,' '} = {'Frequency';'Amplitude';'Duration'};
>>
>> ndata = unique(descrip_stats.Data);
>>
>> for bb = 1:size(ndata,1)
        data_var = string(ndata{bb,1});
        row_loc = find(display_data{:,1} == data_var);
        display_data(row_loc,'significance') = stat_tests((find(display_data{:,1}...
            == data_var)),'sig');
        for ii = 1:2
            if ii == 1;
                data_row_WT = descrip_stats.Data == data_var & ...
                    descrip_stats.Genotype == "WT";
                a_WT = num2str(round(table2array(descrip_stats(data_row_WT,'mean')),2));
                b_WT = num2str(round(table2array(descrip_stats(data_row_WT,'SEM')),2));
                c_WT = {[a_WT,' ',char(177),' ',b_WT]}
                display_data(row_loc,'WT')= c_WT;
            else ii == 2;
                data_row_KO = descrip_stats.Data == data_var & ...
                    descrip_stats.Genotype == "KO"
                a_KO = num2str(round(table2array(descrip_stats(data_row_KO,'mean')),2));
                b_KO = num2str(round(table2array(descrip_stats(data_row_KO,'SEM')),2));
                c_KO = {[a_KO,' ', char(177),' ',b_KO]};
                display_data(row_loc,'KO')= c_KO;
            end
        end
    end
>>
```



	WT	KO	significance
{'Frequency'}	{'0.23 ± 0.01'}	{'0.22 ± 0.01'}	{'ns'}
{'Amplitude'}	{'2.99 ± 0.31'}	{'3.16 ± 0.3' }	{'ns'}
{'Duration' }	{'1.43 ± 0.04'}	{'1.5 ± 0.04' }	{'ns'}

Data displayed as mean ± SEM.

Conclusion:

From our data, the gene of interest appears to have no effect in brain activity. The characteristics (frequency, amplitude and duration) of the brain signals that were investigated were similar between the model without the gene and the control (the intact model). This was supported by the absence of statistical significance between the two groups. As expected, there was great variability in our data, which highlights the diversity in biological samples. This reinforces the importance of robust statistical analysis when working with complex biological datasets.