**EEG Database Data Set Analysis**

Date :14th August 2017  
Name : Saumya Sharma

**Abstract :**

This report analyses two subjects, one alcoholic and the other control. Each subject was exposed to either a single stimulus (S1) or two stimuli (S1 and S2).  For each of the 3 matching paradigms, c\_1 (one presentation only), c\_m (match to previous presentation) and c\_n (no-match to previous presentation), 10 runs were shown. We investigate the various techniques of visualising and understanding the relationships between the two groups. This report samples the small dataset, but the same principle and methodology can be extended for a larger data set.

**Table of contents :**

|  |  |  |
| --- | --- | --- |
| **Abstract** | |  |
| **1.0** | **Introduction** | |  |
| **2.0** | **Parsing and storing data set** | |  |
|  | **2.1 Unzipping the data**  **2.2 Creating columns**  **2.3 Creating pickle files** | |  |
| **3.0** | **Visualising the data** | |  |
|  | **3.1 Visualising the single channel of recording**  **3.2 Visualising all channels of a recording in a single graph**  **3.3 Visualising all channels of a recording in multiple plots**  **3.4 Recording a heatmap for average of each channel**  **3.5 Cubehelix pallete**  **3.6 Bar plots between frequency and voltage** | |  |
| **4.0** | **Recording cross correlations** | |  |
|  | **4.1 Finding the mean of data across all channels in a recording.**  **4.2 Finding the standard deviation of data across all channels in a recording**  **4.3 Pearson correlation**  **4.4 Plotting highest correlating nodes using networkx**  **4.5 Box plot representation**  **4.6 Comparing means between two subjects using lmplot**  **4.7 Performing t-test on both subjects** | |  |
| **5.0** | **Reference list** | |  |

1. Introduction

The purpose of this report is to compare two groups, alcoholic and control to draw out the similarities and differences between the two groups using the visualisation techniques in Python.

Each subject is recorded 10 times over 64 channels with 3 types of stimulus. The goal is to parse this data, and perform the analysis as per the given tasks.

2. Parsing and storing data set

2.1 Unzipping the data

The python code named as "gzip1.py" unzips all the files at once and reveals the data files . This data is parsed and stored in .pkl files within the same directory.

2.2 Creating columns

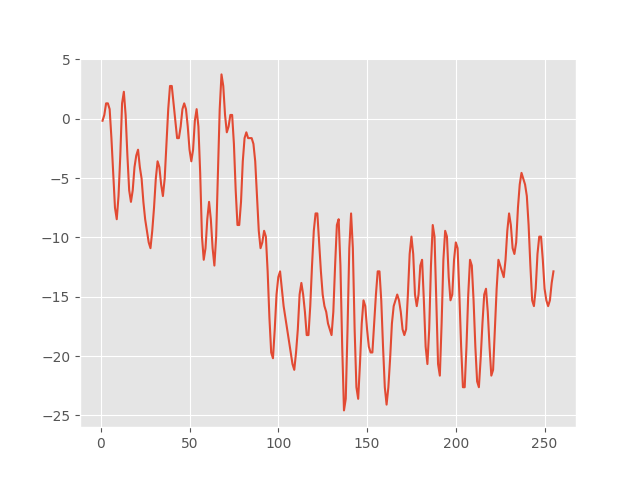
The python code named as "panda1.py" creates 4 columns as per the given data set. This snippet is also responsible for dynamically creating and writing the data to csv files.

2.3 Creating pickle files

This code creates pickle of files by reading the csv files iteratively. I created another class Columns\_seaborn so it returns the individual line of data as a single string. This data is finally dumped to the pickle files.

3. Visualising data

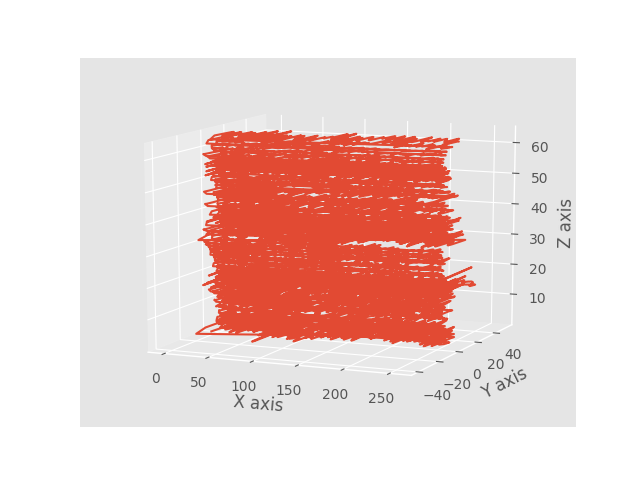
3.1 Visualising a single channel of a recording



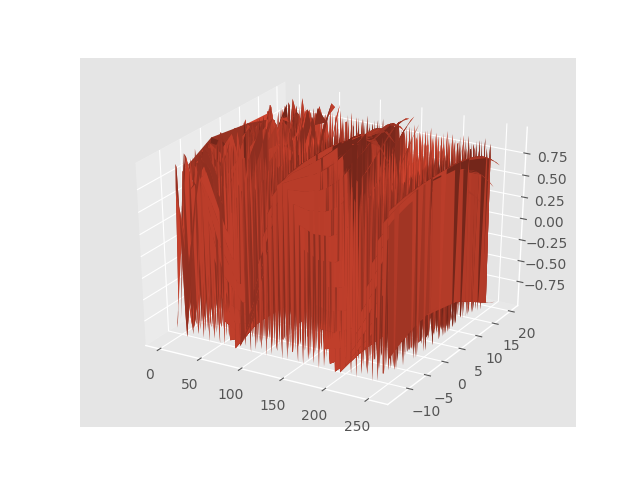
This is the plot of a single channel of a recording. Recording 1: test\_data\_0.txt, channel 1, a\_m\_co2a0000364.

3.2 Visualising all channels of a recording in a single graph

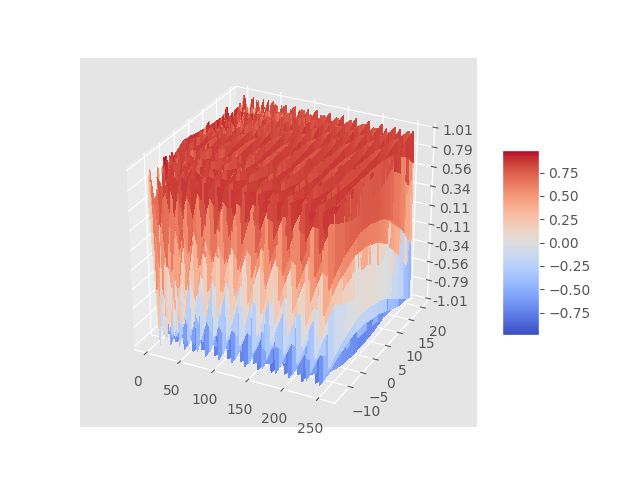
There are several ways to visualise 3D data using wire plots, scatter plots etc. The following image represents the variations of voltage readings over 64 channels. Recording 1 : test\_data\_0.txt, a\_1\_co2a0000364



This wireframe represents the graphs of all 64 channels at once. By rotating the graph and examining through all axes, we can find the connections among 2 axes.



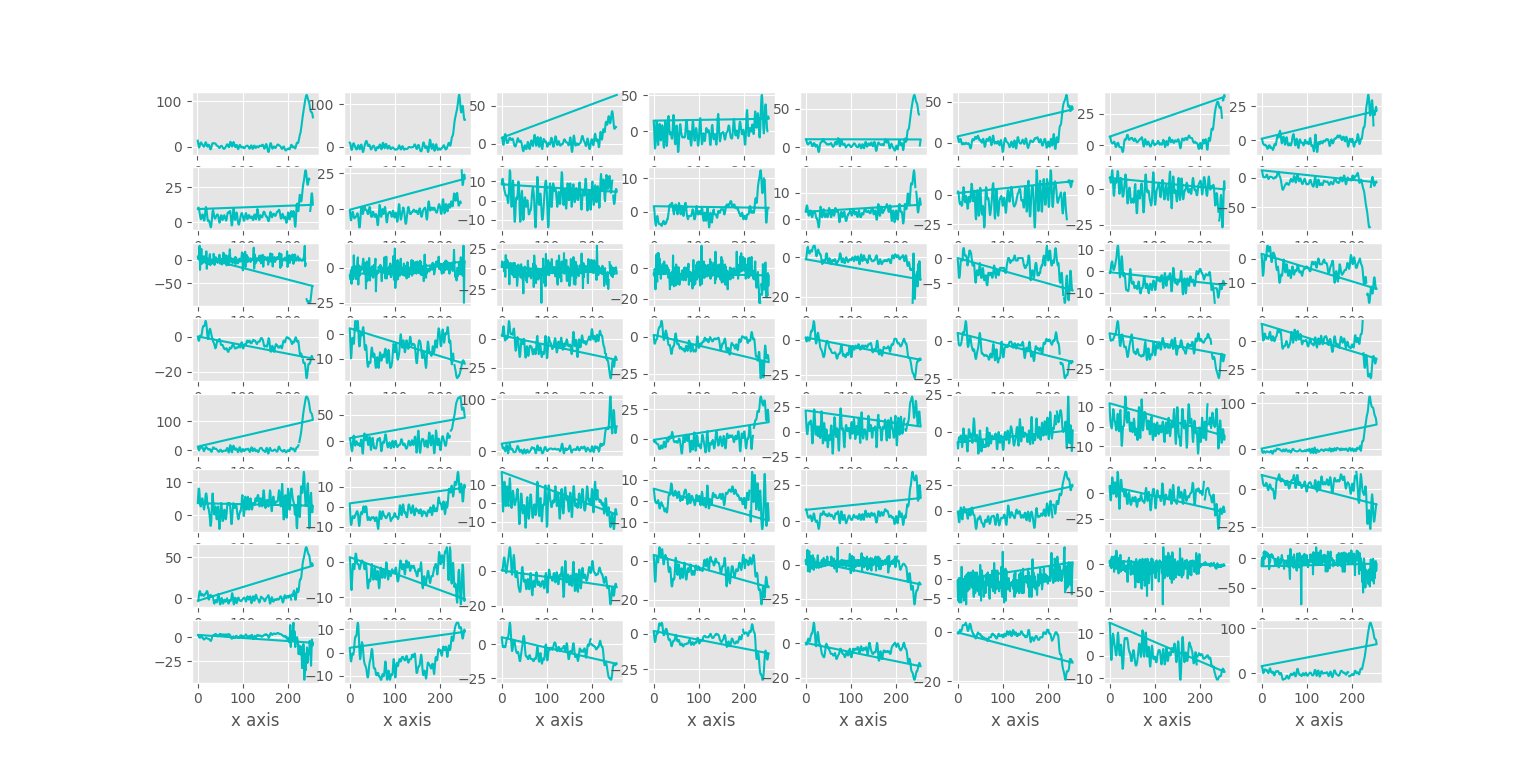
This is a mesh grid representation



The surface plot represents the relationship between the dependent "Y axis" variables, i.e. voltage and independent "X axis" and "Z axis" variables. The above graph shows that X and Y are dependent on each other and Z is a monotonically increasing variable.

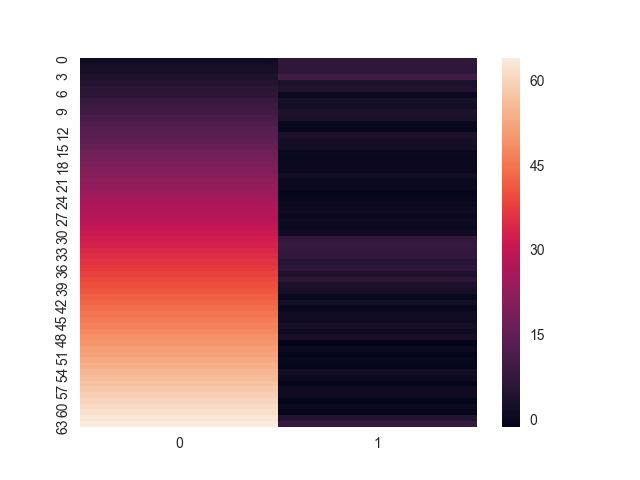
3.3 Visualising all channels of a recording in multiple plots

This plot displays all the variations in voltage as per frequency across all 64 channels individually in a 8x8 grid.

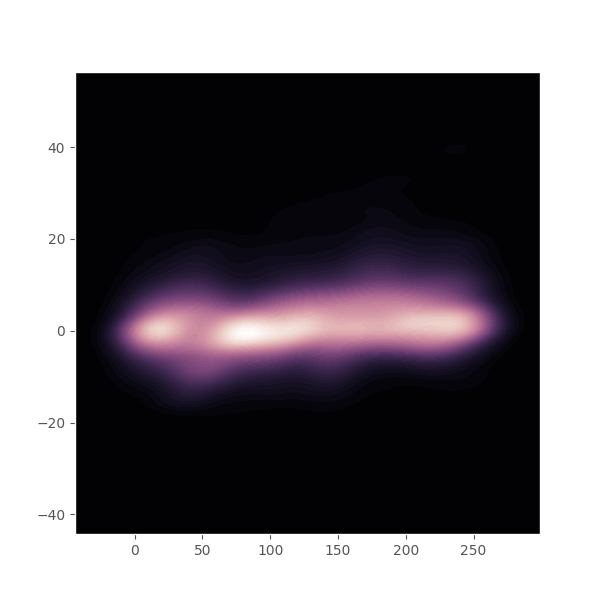


3.4 Recording a heatmap for average of each channel

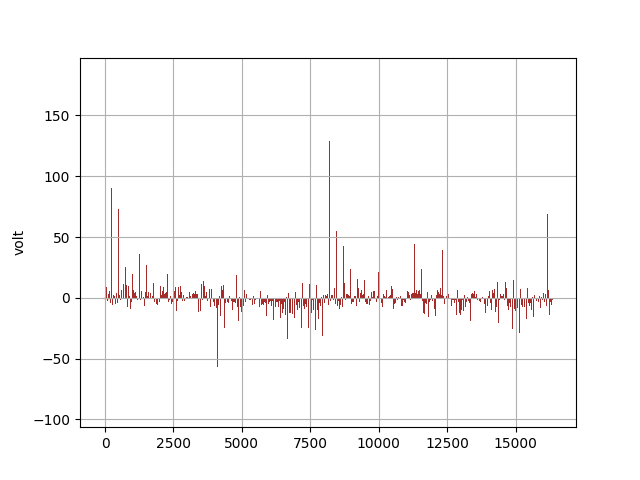
The heatmap is a good representation of the variations of averages of each channel as it displays the increasing averages with the intensity of each recording. Here's a representation of test\_data\_7.txt of c\_1\_co2c0000337.



3.5 Cubehelix palette



3.6 Bar plot between frequency and voltage



4 Recording cross-correlations

Using Pearson correlation, we can find the pairs of channel that have the highest correlation of their averages.

4.1 Finding the mean of data across all channels in a recording.

Using "arr\_hist.py" we create a new file called mean\_file.txt and keep the array of averages across all recordings. Thus, there are 6 rows of 10 elements in mean\_file.txt . These are the readings:

[ 2.10562009 2.29609704 0.32164538 1.04972613 -0.19978566 0.5747783

-1.99107771 1.16958567 -3.21792801 0.61148176]

[-6.68528718 0.18572024 -2.01833169 -2.31188654 -2.6887746 -2.65099137

-0.95367141 -4.19052523 -5.24838027 -4.42598224]

[ 2.00887359 -1.32247118 -2.84299939 -2.26290119 -3.40233236 -2.66604208

-3.27599544 -3.87930433 -4.43998796 -2.0551646 ]

[-2.23398006 -1.55343536 -1.00424328 0.75614508 2.74500261 -1.66698498

3.53602341 2.31842582 3.82492448 2.10852554]

[ 0.10376833 -4.01058726 -5.17869981 -6.91027253 -1.32858634 -7.17423945

-4.24552815 -0.86084939 -1.23227484 -6.93129655]

[-3.7931981 -8.32937243 -2.4994021 -7.11710495 -0.70963085 -4.59666308

-4.35177052 -5.25894582 -5.35001709 -6.28746534]

4.2 Finding the standard deviation of data across all channels in a recording.

Similar to the mean we found above, we use a similar procedure to find out the standard deviation across all recordings.

[ 7.79801199 19.76601472 15.82705349 6.91415758 7.09870293

6.92569651 6.09902497 7.01470074 6.34255776 6.65101916]

4.3 Pearson Correlation

By running "pearson.py", the highest correlating channels can be easily found. Here are the readings of all recordings for different types of stimulus.

highest correlation recordings - a\_l\_co2a0000364

1)29,30

2)31,62

3)31,62

4)57,58

5)57,59

6)29,58

7)57,58

8)57,59

9)57,59

10)29,58

highest correlation recorings - a\_m\_co2a0000364

1)44,47

2)31,62

3)38,62

4)29,58

5)31,62

6)31,62

7)4,6

8)29,58

9)4,6

10)59,60

highest correlation recorings - a\_n\_co2a0000364

1)31,62

2)31,62

3)57,58

4)44,47

5)57,58

6)31,62

7)29,58

8)31,62

9)29,58

10)31,62

highest correlation recordings - c\_1\_co2c0000337

1)22,28

2)20,48

3)23,27

4)1,5

5)1,5

6)1,5

7)1,5

8)21,23

9)29,30

10)23,51

highest correlation recorings - c\_m\_co2c0000337

1)1,31

2)1,5

3)1,5

4)50,60

5)1,33

6)43,47

7)1,33

8)4,5

9)22,28

10)1,5

highest correlation recoring - c\_n\_co2c0000337

1)1,33

2)1,5

3)22,28

4)22,28

5)1,33

6)29,58

7)1,5

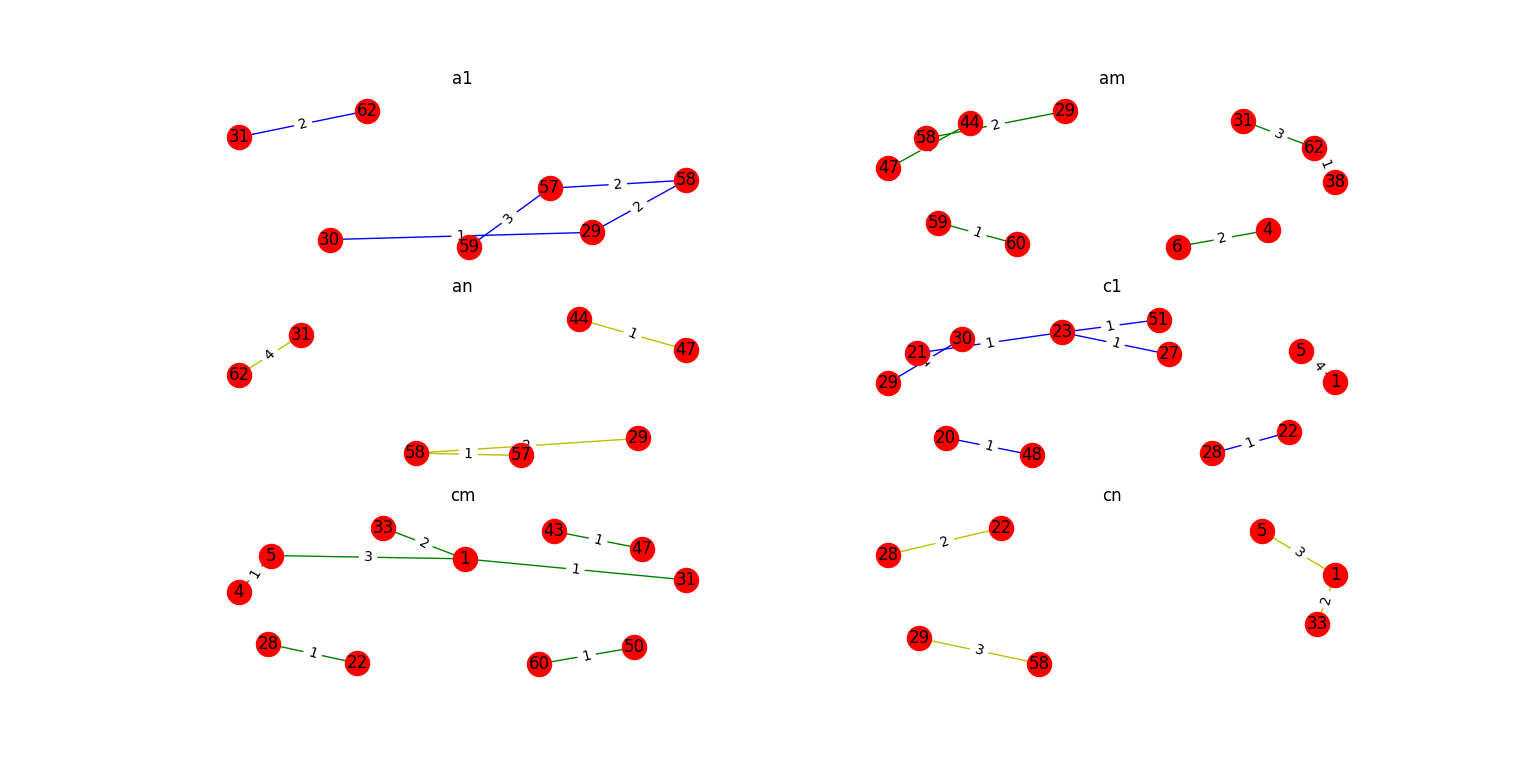
8)29,58

9)1,5

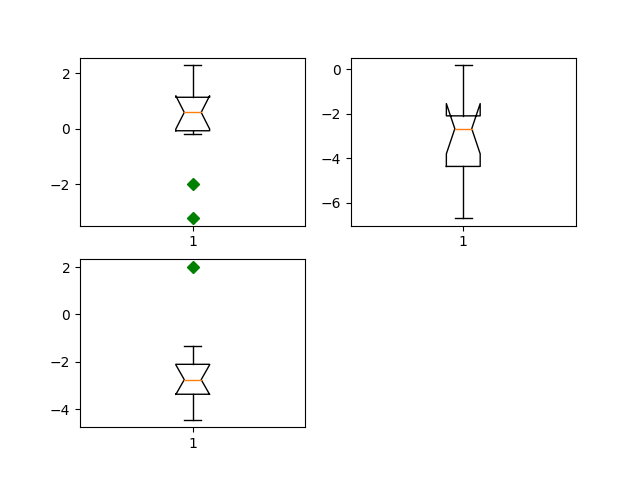
10)29,58

4.4 Plotting highest correlating nodes using networkx

By running "networkx1.py" , the highest correlating channels can be visualised together for both the subjects. The number on the edges represents the number of times these highest correlations occur.



4.5 Box plot representation

We represent the means calculated as a two dimensional matrix. The we move on the plot a box plot comparing the two subjects based on single stimulus, matched and unmatched. And below is the result.

4.6 Comparing means between two subjects using lmplot

This graph is a regression plot between alcoholic and control subjects using lmplot from seaborn.

alc\_l alc\_m alc\_n contr\_l contr\_m contr\_n

0 -2.233980 0.103768 -3.793198 2.105620 -6.685287 2.008874

1 -1.553435 -4.010587 -8.329372 2.296097 0.185720 -1.322471

2 -1.004243 -5.178700 -2.499402 0.321645 -2.018332 -2.842999

3 0.756145 -6.910273 -7.117105 1.049726 -2.311887 -2.262901

4 2.745003 -1.328586 -0.709631 -0.199786 -2.688775 -3.402332

5 -1.666985 -7.174239 -4.596663 0.574778 -2.650991 -2.666042

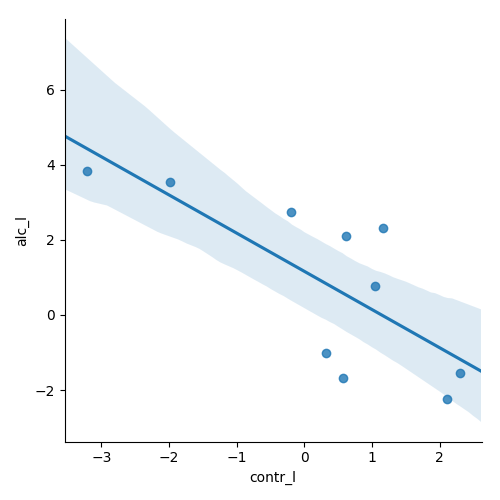
6 3.536023 -4.245528 -4.351771 -1.991078 -0.953671 -3.275995

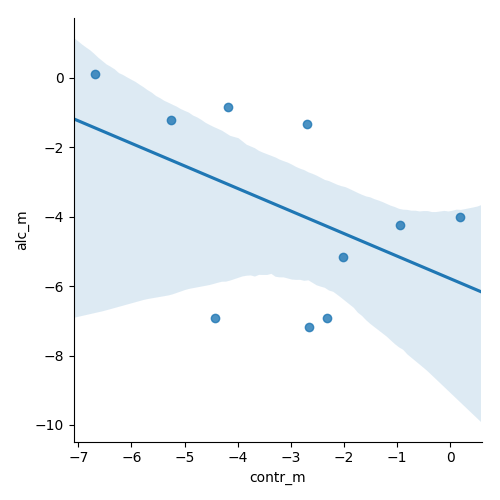
7 2.318426 -0.860849 -5.258946 1.169586 -4.190525 -3.879304

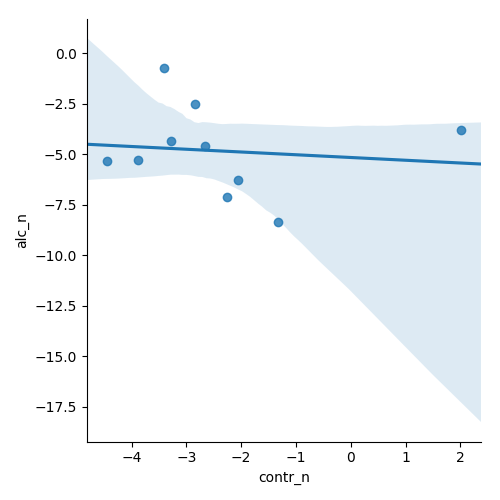
8 3.824924 -1.232275 -5.350017 -3.217928 -5.248380 -4.439988

9 2.108526 -6.931297 -6.287465 0.611482 -4.425982 -2.055165

The data frame created for comparison. Run "tester.py".

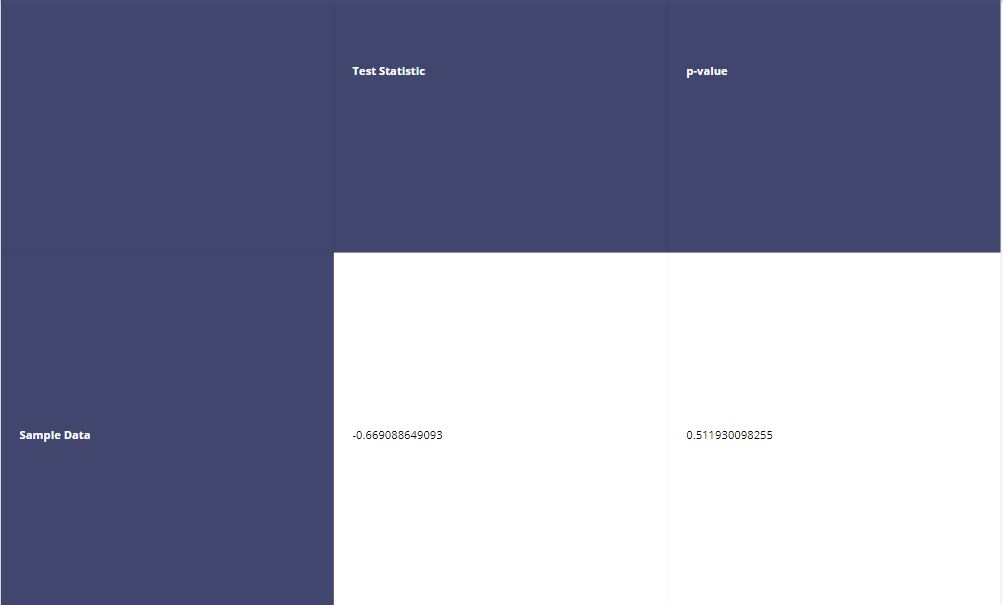






4.7 Performing t-test on both subjects

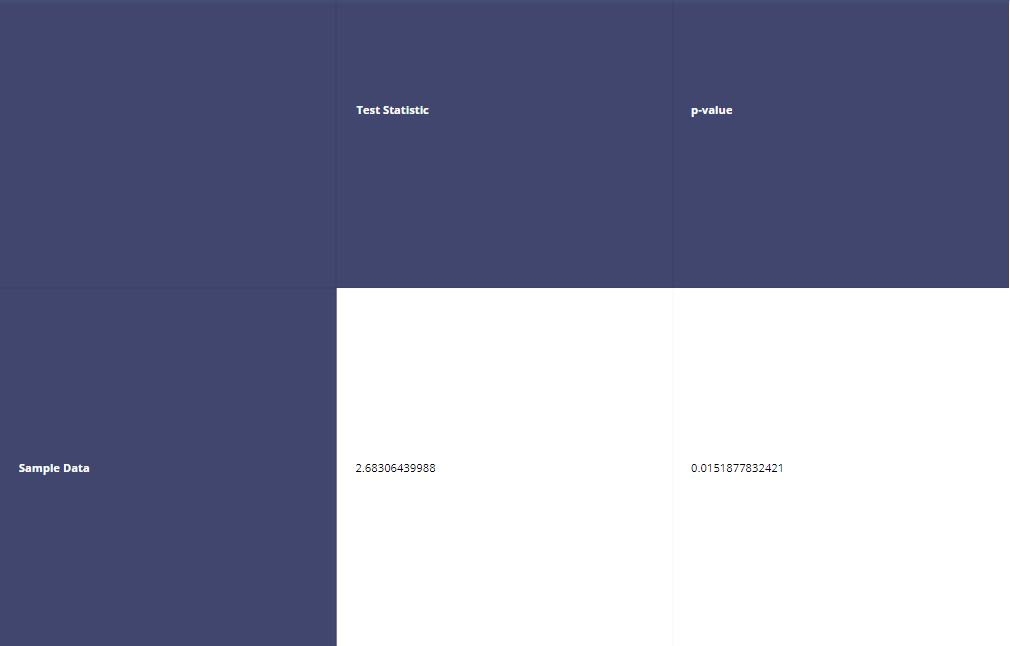
alcoholic and control - 1



alcoholic and control - matched



alcoholic and control - unmatched



5. Reference List

https://matplotlib.org/mpl\_toolkits/mplot3d/tutorial.html#contour-plots

https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.multivariate\_normal.html

http://www.scipy-lectures.org/packages/statistics/auto\_examples/plot\_regression\_3d.html

http://www.scipy-lectures.org/packages/statistics/index.html#linear-models-multiple-factors-and-analysis-of-variance

http://seaborn.pydata.org/generated/seaborn.lmplot.html