Contents

[Table of Figures 4](#_Toc135847701)

[Table of Tables 6](#_Toc135847702)

[Data Analysis 7](#_Toc135847703)

[automizePsychometricFunctions.m 7](#_Toc135847704)

[sigmoidAnalysis.m 10](#_Toc135847705)

[createThePlot.m 10](#_Toc135847706)

[Creating Probability Tables 11](#_Toc135847707)

[Recreating Figures 13](#_Toc135847708)

[Creating Baseline Travel Pixel Figures 13](#_Toc135847709)

[travelPixelSigmoidClustering.m ˟ 13](#_Toc135847710)

[travelPixelMaxVsShift.m 14](#_Toc135847711)

[travelPixelMaxVsSteepness.m 14](#_Toc135847712)

[travelPixelShiftVsSteepness.m 14](#_Toc135847713)

[automizeTravelPixelPsychomaticalFunction.m 14](#_Toc135847714)

[travelPixelSigmoidAnalysis.m 15](#_Toc135847715)

[Creating Baseline Stopping Point Figures 16](#_Toc135847716)

[stoppingPointsSigmoidClustering.m ˟ 16](#_Toc135847717)

[stoppingPointsMaxVsShift.m 17](#_Toc135847718)

[stoppingPointsMaxVsSteepness.m 17](#_Toc135847719)

[stoppingPointsShiftVsSteepness.m 17](#_Toc135847720)

[stoppingptsSigmoidAnalysis.m 18](#_Toc135847721)

[Creating Baseline Rotation Points Figures 19](#_Toc135847722)

[RotationPtsSigmoidClustering.m 19](#_Toc135847723)

[rotationPtsMaxVsShift.m 19](#_Toc135847724)

[rotationPtsMaxVsSteepness.m 19](#_Toc135847725)

[rotationPtsShiftVsSteepness.m 19](#_Toc135847726)

[automizeRotationPtsPsychomaticalFunction.m 20](#_Toc135847727)

[RotationPtsSigmoidAnalysis.m 20](#_Toc135847728)

[Creating Baseline Reaction Time Figures 21](#_Toc135847729)

[reactionTime1stSigmoidClustering.m 21](#_Toc135847730)

[reactionTime1stMaxVsShift.m 21](#_Toc135847731)

[reactionTime1stMaxVsSteepness.m 21](#_Toc135847732)

[reactiontime1stShiftVsSteepness.m 21](#_Toc135847733)

[Automizereactiontime1stPsychomaticalFunctions.m 21](#_Toc135847734)

[ReactionTime1stSigmoidAnalysis.m 22](#_Toc135847735)

[Creating Reward Choice Figures 23](#_Toc135847736)

[RewardChoiceSigmoidClustering.m 23](#_Toc135847737)

[rewardChoiceMaxVsShift.m 23](#_Toc135847738)

[rewardChoiceMaxVsSteepness.m 23](#_Toc135847739)

[rewardChoiceShiftVsSteepness.m 23](#_Toc135847740)

[automizePsychomaticalFunction.m 24](#_Toc135847741)

[sigmoidAnalysis.m 24](#_Toc135847742)

[Creating Average Sigmoid Figures for Baseline 25](#_Toc135847743)

[createAverage.m ˟ 25](#_Toc135847744)

[Creating Average Sigmoid For Food Deprivation 26](#_Toc135847745)

[createAverage.m 26](#_Toc135847746)

[Parabola Clustering 27](#_Toc135847747)

[getThreeParametersFromParabolas.m ˟ 27](#_Toc135847748)

[Venn Diagram 27](#_Toc135847749)

[Probability Tables 29](#_Toc135847750)

[Create Rat Probabilities Chart with Probability Labels.ipynb 29](#_Toc135847751)

[Get Rat Probabilities Euclidian Distance.ipynb 29](#_Toc135847752)

[Population Probabilities 30](#_Toc135847753)

[collectAllDataInAMap.m 31](#_Toc135847754)

[get\_oxy\_probabilities.m 31](#_Toc135847755)

[get\_LG\_boost\_probabilities.m 31](#_Toc135847756)

[get\_Lg\_etoh\_probabilities.m 31](#_Toc135847757)

[createBaselineMaps.m 31](#_Toc135847758)

[createFoodDeprivationMaps.m 31](#_Toc135847759)

[Create Population Probability Charts.ipynb 31](#_Toc135847760)

[Create Gender Population Probability Charts.ipynb 31](#_Toc135847761)

[Hierarchal Tree 32](#_Toc135847762)

# Table of Figures

[Figure 1: example of collectAllDataInAMap() being called with the lg\_boost data set 11](#_Toc135847763)

[Figure 2: example of using getSizeOfAllDataSets() function using the lg\_boostData variable 11](#_Toc135847764)

[Figure 3: example of using getSizesOfEachCluster() function using the "lg\_boost clusters" directory 12](#_Toc135847765)

[Figure 4: This code takes the mapOfProbabilities variable gotten from the calculatePopulationProbabilities (mapOfAllClusterSizes, mapOfAllDataSetSizes) function, and turns it into a readable table 12](#_Toc135847766)

[Figure 5: Travel Pixel Max Vs Shift MPC: 0.8925 13](#_Toc135847767)

[Figure 6: Travel Pixel Mac Vs Steepness: 0.8925 13](#_Toc135847768)

[Figure 7: Travel Pixel Shift Vs Steepness, MPC: 0.8628 13](#_Toc135847769)

[Figure 8: example of the variable that has to be modified in travelPixelSigmoidClustering.m 13](#_Toc135847770)

[Figure 9: example of the fullFileName variable which needs to be changed in travelPixelSigmoidClustering.m to work 13](#_Toc135847771)

[Figure 10: Example query that filters out all experiment data. 14](#_Toc135847772)

[Figure 11: Query that will return searchResults in a format that will work for createPsychomaticalFunction(). 15](#_Toc135847773)

[Figure 12: Stopping Points Max Vs Shift MPC: 0.9279 16](#_Toc135847774)

[Figure 13: Stopping Points Max Vs Steepness 0.9279 16](#_Toc135847775)

[Figure 14: Stopping Points Shift Vs Steepness MPC: 0.9279 16](#_Toc135847776)

[Figure 15: example of the variable that has to be modified in stoppingPointsSigmoidClustering.m 16](#_Toc135847777)

[Figure 16: example of the fullFileName variable which needs to be changed in stoppingPointsSigmoidClustering.m to work 16](#_Toc135847778)

[Figure 17: Example query that filters out all experiment data. 17](#_Toc135847779)

[Figure 18: Rotation Points Max Vs Shift MPC 0.9300 19](#_Toc135847780)

[Figure 19: Rotation Points Max Vs Steepness: 0.9422 19](#_Toc135847781)

[Figure 20: Rotation Points Shift Vs Steepness MPC: 0.8946 19](#_Toc135847782)

[Figure 21: Reaction Time Max Vs Shift MPC: 0.8950 21](#_Toc135847783)

[Figure 22: Reaction Time Max VS Steepness MPC: 0.9046 21](#_Toc135847784)

[Figure 23: Reaction Time Shift Vs Steepness, MPC: 0.8575 21](#_Toc135847785)

[Figure 24: Query that will return searchResults in a format that will work for automizereactiontime1stPsychomaticalFunctions.m 22](#_Toc135847786)

[Figure 25: Reward Choice Max Vs Steepness, MPC: 0.9057 23](#_Toc135847787)

[Figure 26: Reward Choice Max Vs Steepness, MPC: 0.9057 23](#_Toc135847788)

[Figure 27: Reward Choice Max Vs Steepness, MPC: 0.9057 23](#_Toc135847789)

[Figure 28: Query that will return searchResults in a format that will work for automizePsychomaticalFunction.m 24](#_Toc135847790)

[Figure 29: Average Sigmoid for Baseline J1, J2, J3, and J4 clusters. 25](#_Toc135847791)

[Figure 30: Average Sigmoid For Food Deprivation J1, J2,J3, and J4 clusters. 26](#_Toc135847792)

[Figure 31: Parabola Clustering 27](#_Toc135847793)

[Figure 32: Lines of code that need to be uncommented for getThreeParametersFromParabolas.m to work. 27](#_Toc135847794)

[Figure 33: Venn Diagrams 27](#_Toc135847795)

[Figure 34: Code 1 that needs to be modified in Create Multiple Venn Diagrams to work. 28](#_Toc135847796)

[Figure 35:Code 2 that needs to be modified in Create Multiple Venn Diagrams to work. 28](#_Toc135847797)

[Figure 36:Code 3 that needs to be modified in Create Multiple Venn Diagrams to work. 28](#_Toc135847798)

[Figure 37: Probability Tables individuals 29](#_Toc135847799)

[Figure 38: Population Probabilities for Oxycodone and Alcohol. 30](#_Toc135847800)

[Figure 39: Population Probabilities for Baseline and Food Deprivation 30](#_Toc135847801)

[Figure 40: The line of code that needs to be modified in collectAllDataInAMap.m 31](#_Toc135847802)

[Figure 41: Code that needs to be modified for createBaselineMaps.m to run. 31](#_Toc135847803)

[Figure 42: Code that needs to be modified for createFoodDeprivationMaps.m to work. 31](#_Toc135847804)

[Figure 43: Hierarchal Clustering Tree 32](#_Toc135847805)

# Table of Tables

[Table 1: Displays the order in which the columns of searchResults must be structured for the CreateRewardChoicePyschometricFunctions function 7](#_Toc135847806)

[Table 2: Displays the order in which the columns of the searchResults Variable must be structured for the createReactionTimePsychometricFunctions function. 8](#_Toc135847807)

[Table 3: Displays the order in which the columns of the searchResults Variable must be structured for the createRotationPointsPsychometricFunctions function. 8](#_Toc135847808)

[Table 4: Displays the order in which the columns of the searchResults Variable must be structured for the createStoppingPointsPsychometricFunction function. 8](#_Toc135847809)

[Table 5: Displays the order in which the columns of the searchResults Variable must be structured for the createTravelPixelPsychometricFunctions function. 9](#_Toc135847810)

[Table 6: Table of date ranges for base data sigmoid figures 14](#_Toc135847811)

[Table 7: Example of how the searchResults table must be formatted. 15](#_Toc135847812)

[Table 8: Example of travelPixelPsychomaticalFunctions table 15](#_Toc135847813)

[Table 9 Example of how the searchResults table must be formatted. 17](#_Toc135847814)

[Table 10: Query that will return searchResults in a format that will be accepted by createPsychomaticalFunction(). 17](#_Toc135847815)

[Table 11: Example of how the searchResults table must be formatted. 20](#_Toc135847816)

[Table 12: Query that will return searchResults in a format that will work for createPsychomaticalFunction. 20](#_Toc135847817)

[Table 13: Example of how the searchResults table must be formatted for reactionTime1st. 22](#_Toc135847818)

[Table 14: List of Cluster Files 32](#_Toc135847819)

[Table 15: Table of Rat Names to Gender. f = female. m = male. 34](#_Toc135847820)

˟ - Functions or files with this symbol MUST be modified before they can be used

# Data Analysis

These functions are used to analyze the raw trial data stored in live\_table located in the database stored in the [Data Base Backup As 04-11-2023.zip.](https://doi.org/10.7910/DVN/QADUKS) From this analysis you can create clusters.

## automizePsychometricFunctions.m

The purpose of this MATLAB file is to automatically create the y coordinates of a psychometric function of sessions performed on rats. All the following functions contribute to the overall goal of this file, and these functions can be modified to make the output of the file match your own goals.

1. createMap: Reads all rows of live\_table and returns a map where the keys are the date a trial was performed, and the values are what rats were tested on that date.
   1. There is a variable “query” that can be altered to affect what data is being mapped, this query can be written in PostgreSQL format and still work.
      1. SELECT \* FROM live\_table;
         1. Gets all columns and all rows from live\_table.
      2. SELECT referencetime, subjectid, mazenumber FROM live\_table WHERE genotype=’lg\_boost’;
         1. Gets the columns reference time, subject id, and maze number from live\_table where the genotype is listed as lg\_boost.
2. formatTheDate: reads the given date and returns it into a MM/DD/YYYY format.
3. CreateRewardChoicePsychometricFunctions(searchResults): Takes the results of PostgreSQL query stored inside of a nx8 MATLAB table where the following columns are used in the SELECT statement of the query. Order must be maintained. Will return an average of approach/avoid for each of the four feeders.

|  |  |
| --- | --- |
| Column1 | subjectid |
| Column 2 | referencetime |
| Column 3 | feeder |
| Column 4 | approachavoid |
| Column 5 | rewardconcentration1 |
| Column 6 | rewardconcentration2 |
| Column 7 | rewardconcentration3 |
| Column 8 | rewardconcentration4 |
| SELECT  subjectid, referencetime, feeder, approachavoid, rewardconcentration1, rewardconcentration2,  rewardconcentration3, rewardconcentration4 FROM live\_table  WHERE SOME CONDITION; | |

Table : Displays the order in which the columns of searchResults must be structured for the CreateRewardChoicePyschometricFunctions function

1. createReactionTimePsychometricFunctions(searchResults): Takes the results of PostgreSQL query stored inside of a nx8 MATLAB table where the following columns are used in the SELECT statement of the query. Returns the average of Reactiontime1st across all four feeders.

|  |  |
| --- | --- |
| Column1 | id |
| Column 2 | referencetime |
| Column 3 | subjectid |
| Column 4 | reactiontime1st |
| SELECT  id, referencetime, subjectid, reactiontime1st FROM featuretable2  WHERE SOME CONDITION; | |

Table : Displays the order in which the columns of the searchResults Variable must be structured for the createReactionTimePsychometricFunctions function.

1. createRotationPointsPsychometricFunctions: Takes the results of a PostgreSQL query stored inside of a nx8 MATLAB table where the following columns are used in the SELECT statement of the query. Returns the average of rotationptsmethod1 across all 4 feeders.

|  |  |
| --- | --- |
| Column1 | id |
| Column 2 | referencetime |
| Column 3 | subjectid |
| Column 4 | rotationptsmethod1 |
| SELECT  id, referencetime, subjectid, rotationptsmethod1 FROM featuretable2  WHERE SOME CONDITION; | |

Table : Displays the order in which the columns of the searchResults Variable must be structured for the createRotationPointsPsychometricFunctions function.

1. createStoppingPointsPsychometricFunctions(searchResults): Takes the results of a PostgreSQL query stored inside of a nx8 MATLAB table where the following columns are used in the SELECT statement of the query. Returns the average stoptimemethod6 across 4 feeders.

|  |  |
| --- | --- |
| Column1 | id |
| Column 2 | referencetime |
| Column 3 | subjectid |
| Column 4 | stoptimemethod6 |
| SELECT  id, referencetime, subjectid, stoptimemethod6 FROM featuretable2  WHERE SOME CONDITION; | |

Table : Displays the order in which the columns of the searchResults Variable must be structured for the createStoppingPointsPsychometricFunction function.

1. createTravelPixelPsychometricFunctions(searchResults): Takes the results of a PostgreSQL query stored inside of a nx8 MATLAB table where the following columns are used in the SELECT statement of the query. Returns the average distanceaftertoneuntillimitingtimestamp across 4 feeders.

|  |  |
| --- | --- |
| Column1 | id |
| Column 2 | referencetime |
| Column 3 | subjectid |
| Column 4 | distanceaftertoneuntillimitingtimestamp |
| SELECT  id, referencetime, subjectid, distanceaftertoneuntillimitingtimestamp FROM featuretable2  WHERE SOME CONDITION; | |

Table : Displays the order in which the columns of the searchResults Variable must be structured for the createTravelPixelPsychometricFunctions function.

1. rewardChoiceLoop(T, tableToWriteIn)
   1. T – the map created by the function createMap.
   2. tableToWriteIn – a string name of the table you want to write in the database, if the table does not exist it will be created automatically.
   3. This function automatically queries data from live\_table, and puts it in a format suitable for CreateRewardChoicePsychometricFunctions, then calls that function. It then writes the results of this function to the table specified in tableToWriteIn.
2. reactionTimeLoop(T, tableToWriteIn)
   1. T – the map created by the function createMap.
   2. tableToWriteIn – a string name of the table you want to write in the database, if the table does not exist it will be created automatically.
   3. This function automatically queries data from live\_table, and puts it in a format suitable for CreateReactionTimePsychometricFunctions, then calls that function. It then writes the results of this function to the table specified in tableToWriteIn.
3. rotationPointsLoop(T, tableToWriteIn).
   1. T – the map created by the function createMap.
   2. tableToWriteIn – a string name of the table you want to write in the database, if the table does not exist it will be created automatically.
   3. This function automatically queries data from live\_table, and puts it in a format suitable for CreateRotationPointsPsychometricFunctions, then calls that function. It then writes the results of this function to the table specified in tableToWriteIn.
4. stoppingPointsLoop(T, tableToWriteIn)
   1. T – the map created by the function createMap.
   2. tableToWriteIn – a string name of the table you want to write in the database, if the table does not exist it will be created automatically.
   3. This function automatically queries data from live\_table, and puts it in a format suitable for CreateStoppingPointsPsychometricFunctions, then calls that function. It then writes the results of this function to the table specified in tableToWriteIn.
5. travelPixelLoop(T, tableToWriteIn)
   1. T – the map created by the function createMap.
   2. tableToWriteIn – a string name of the table you want to write in the database, if the table does not exist it will be created automatically.
   3. This function automatically queries data from live\_table, and puts it in a format suitable for CreateTravelPixelPsychometricFunctions, then calls that function. It then writes the results of this function to the table specified in tableToWriteIn.

## sigmoidAnalysis.m

This MATLAB file will creates folders in the current directory where data will be stored, and then calls the following function 5 times to actually store the data.

1. createSigmoidFigures(results, feature)
   1. results – the results of a querying all data from a table created from any of the following functions: rewardChoiceLoop, reactionTimeLoop, rotationPointsLoop, stoppingPointsLoop, travelPixelLoop.
      1. SELECT \* FROM tableToWriteIn;
   2. feature: one of the following strings: “tp”, ”sp”, ”rp”, ”rc”, “rt”. The string specified tells the function which folder to store its output to. “tp” will make the function store all output to the travel pixel folders, “sp” will make the function store all output to the stopping points folder etc.
   3. This function will take each row of results and fit the (x,y) coordinates stored within that row with a parabola model, a linear model, a 2 parameter sigmoid model, a 3 parameter sigmoid model, and a 4 parameter sigmoid model. It then takes a measure of how well the data fits those models’ using coefficient of determination (R-squared). It then stores both the graph of the best model, and the fitting parameters into the folders indicated by feature

## createThePlot.m

1. createThePlot(param1, param2, param3,param4,param5,param6,param7)
   1. This function servers 3 utilities
      1. It utilizes Fuzzy C-Means clustering to cluster a given nx2 array (param1), with string labels (param2), with a desired number of clusters (param 3) and displays it into a figure.
      2. It will print a 0-1 score of your clustering with 1 being better.
      3. It will create a directory "All Clusters In Dataset" in your current directory and write all of the desired clusters with labels to this dataset
         1. Note that while multiple running of this function might produce the same clusters, different names might be assigned to them.
         2. This is because Fuzzy C-Means clustering picks a random starting point.
         3. You may overwrite old clusters by running this function multiple times.
         4. To avoid this change param7
   2. param1: an n row by 2 column array which contains the x & y data for the clusters you are trying to create.
   3. param2: an n row by 2 column string array that has the session labels for each data point, both columns should be identical.
   4. param3: This is how many clusters you expect in the data.
      1. This number should be altered to maximize the score printed by running this.
   5. param4: A string representing what will go along the x-axis of the figure.
   6. param5: A string representing what will go along the y-axis of the figure.
   7. param6: A string representing what feature is being clustered.
   8. param7: An arbitrary letter used when writing the found clusters to a file.
      1. This letter should be changed for each feature you are analyzing.
      2. The ideal letters to be used for each feature should be A=Travel Pixel, D= Stopping Points, G = Rotation Points, J = Reaction Time, M = Reward Choice
      3. These are ideal because the creation of Probability Tables relies on these names, but if one is inclined, they can rewrite the Probability Tables Code to accept different names.

# Creating Probability Tables

The following files and functions are used to create a table of probabilities that displays what probability of a point belonging to any given cluster in the dataset. To figure out how to create the clusters please reference Data Analysis

1. collectAllDataMap(dataSets): ˟
   1. This function will return a map of all data specified in dataSets.
      1. The keys of the map are specified in the dataSets variable.
      2. The values of the map are MATLAB cluster tables created by the createThePlot () function.
   2. dataSets is a string array specified as follows
      1. dataSets = ["Experiment you're analyzing|Travel Pixel","Experiment you're analyzing|Stopping Points","Experiment you're analyzing|Rotation Points","Experiment you're analyzing|Reward Choice"];
   3. This function has a variable “filePathWithData” which MUST be modified to specify the file path on your local machine which contains the experiment data.



Figure : example of collectAllDataInAMap() being called with the lg\_boost data set

1. getSizeOfAllDataSets(mapOfData)
   1. This function returns a map of sizes of the datasets stored inside the “mapOfData” variable,
   2. Returns the map dataSetToSize, where the keys are the name of the dataset, and the values are the number of data points in the dataset
   3. It requires that that keys of the mapOfData map, take the format of the variable “dataSets” found in collectAllDataMap(dataSets)



Figure : example of using getSizeOfAllDataSets() function using the lg\_boostData variable

1. getSizesOfEachCluster(directoryOfClusters) ˟
   1. This function must be modified before it can be used.
   2. Returns the map clusterSize, where the keys are the name of the cluster, and the value is the number of data points in the cluster.
   3. Will navigate into the specified directory.
   4. It will read all files in the specified folder, which should be cluster files created in the createThePlot (param1, param2, param3,param4,param5,param6,param7) function
   5. By default, it will only read .xlsx files with the following list of names
      1. {'A1','A2','A3','D1','D2','D3','G1','G2','G3','G4', 'M1','M2','M3','M4'}
      2. To change this list go into the file and change the “keyset variable”



Figure : example of using getSizesOfEachCluster() function using the "lg\_boost clusters" directory

1. calculatePopulationProbabilities (mapOfAllClusterSizes, mapOfAllDataSetSizes) ˟
   1. This function has to be modified before it can be used.
   2. Returns the variable: mapOfProbabilities which is a map where the keys is the name of the cluster, and the value is the probability of a cluster existing
      1. Probability is defined as # of data points in the cluster/#of total data points for the set
   3. By default it will only read .xlsx files with the following list of names
      1. {'A1','A2','A3','D1','D2','D3','G1','G2','G3','G4', 'M1','M2','M3','M4'}
      2. To change this list, go into the file and change the “keyset variable”.
   4. The following lines of code can be used to turn the data gotten from these functions into a csv file where it can be transformed to a figure suitable for demonstration purposes.

Text

Description automatically generated

Figure : This code takes the mapOfProbabilities variable gotten from the calculatePopulationProbabilities (mapOfAllClusterSizes, mapOfAllDataSetSizes) function, and turns it into a readable table

# Recreating Figures

## Creating Baseline Travel Pixel Figures

All of the codes mentioned in this section can be found in the following GitHub directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Travel%20Pixel>

Chart, scatter chart

Description automatically generated

Figure : Travel Pixel Max Vs Shift MPC: 0.8925

Chart, scatter chart

Description automatically generated

Figure : Travel Pixel Mac Vs Steepness: 0.8925

Chart, scatter chart

Description automatically generated

Figure : Travel Pixel Shift Vs Steepness, MPC: 0.8628

To recreate the figures exactly: download all the files found at the following link.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Travel%20Pixel/All%20Sigmoids>

Store these files in a folder on your local computer then run the following file.

### travelPixelSigmoidClustering.m ˟

1. Modify the “myDir” variable so it points to the file which houses all the data that you downloaded.



Figure : example of the variable that has to be modified in travelPixelSigmoidClustering.m

1. Modify the fullFileName variable with the same path used in myDir variable.



Figure : example of the fullFileName variable which needs to be changed in travelPixelSigmoidClustering.m to work

1. travelPixelSigmoidClustering.m will create a variable called newTable which contains all the data obtained from fitting the raw data with a sigmoid.

Now run the following files.

### travelPixelMaxVsShift.m

### travelPixelMaxVsSteepness.m

### travelPixelShiftVsSteepness.m

To create an updated figure from our raw data run the following functions. Keep in mind that these figures will likely include the data seen in the figures above, as well as new data. If there is a desire to recreate figure 5, figure 6, and figure 7 directly from raw data you must filter data down to the date ranges found in the file where you downloaded all the data.

|  |  |
| --- | --- |
| Beginning Date Range | End Date Range |
| 11-11-2021 | 05-23-2022 |

Table : Table of date ranges for base data sigmoid figures

automizeTravelPixelPsychomaticalFunction.m ˟

This file will read from featuretable, located in the database backup found in [Data Base Backup As 04-11-2023.zip.](https://dataverse.harvard.edu/file.xhtml?fileId=7060611&version=DRAFT) For each trial in a session that a rat performs it will take the average measured travel pixel, for each of the 4 feeder values. It will write these values to the table found in the database “travelPixelPsychomaticalFunctions”.

1. createMap ()
   1. This function returns a Map T
      1. Keys are dates.
      2. The values are a list of all rats to run a session on this date.
   2. This function must be modified.
      1. The variable “query” located in line 3 of the file must be modified to exclude any experiment data.



Figure : Example query that filters out all experiment data.

* + 1. This must be modified because at the date of this figure creation no experiments were being run and as a result there was no need to filter, but the database now includes experiment data.

1. createPsychomaticalFunction(searchResults)
   1. This function returns xcoordinates, ycoordinates.
   2. xcoordinates is the following array of values.
      1. [0.005, 0.01, 0.02, 0.09]
      2. Each of these values represents the percentage of glucose located in a feeder which is present during the trials.
   3. searchResults is a MATLAB in the following form.

|  |  |
| --- | --- |
| id | travelpixel |
| 1 | 10000 |

Table : Example of how the searchResults table must be formatted.

* 1. You can get a table like this by using the following query
     1. date is a date that is formatted in MM-shortened Month-YYYY format
     2. animalsubjectid is any name of an animal in the table

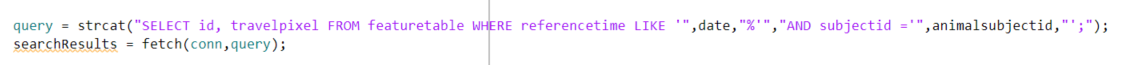


Figure : Query that will return searchResults in a format that will work for createPsychomaticalFunction().

* 1. ycoordinates is the averages of travel pixel recorded during each of these feeder values.
     1. The average is calculated by summing the travel pixel based on which feeder was active during the trial, and dividing by the number of trials that used that feeder.

1. Once the functions above are run they will automatically write the results to a table in the database named “travelPixelPsychomaticalFunctions” in the following format

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| subjectid | date | x1 | x2 | x3 | x4 | y1 | y2 | y3 | y4 |
| example id | example date | 0.005 | 0.01 | 0.02 | 0.09 | average | average | average | average |

Table : Example of travelPixelPsychomaticalFunctions table

### travelPixelSigmoidAnalysis.m

1. This function will read from the “travelPixelPsychomaticalFunctions” table created by automizeTravelPixelPsychomaticalFunction.m and fit the data there with a parabola, a sigmoid, and a line.
2. Coefficient of determination (R-Squared) is used to measure goodness of fit
3. Each row of data in the “travelPixelPsychomaticalFunctions” table will be sorted into folders based on which model fits it better, determined by which produces a higher R-Squared value.

After these functions are run, you can now run travelPixelMaxVsShift.m, travelPixelMaxVsSteepness.m, and travelPixelShiftVsSteepness.m. This should create updated figures.

## Creating Baseline Stopping Point Figures

Chart, scatter chart

Description automatically generated

Figure : Stopping Points Max Vs Shift MPC: 0.9279

Chart, scatter chart

Description automatically generated

Figure : Stopping Points Max Vs Steepness 0.9279

Chart, scatter chart

Description automatically generated

Figure : Stopping Points Shift Vs Steepness MPC: 0.9279

All the codes discussed here can be found in the following GitHub directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Stopping%20Points>

To recreate the figures exactly: download all the files found at the following link.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Stopping%20Points/All%20Sigmoids>

Store the files in a folder on your computer and run the following functions.

### stoppingPointsSigmoidClustering.m ˟

1. Modify the “myDir” variable so it points to the file which houses all the data that you downloaded.



Figure : example of the variable that has to be modified in stoppingPointsSigmoidClustering.m

1. Modify the fullFileName variable with the same path used in myDir variable.



Figure : example of the fullFileName variable which needs to be changed in stoppingPointsSigmoidClustering.m to work

1. stoppingPointsSigmoidClustering.m will create a variable called newTable which contains all the data obtained from fitting the raw data with a sigmoid.

Now run the following files.

### stoppingPointsMaxVsShift.m

### stoppingPointsMaxVsSteepness.m

### stoppingPointsShiftVsSteepness.m

To create an updated figure from our raw data run the following functions. Keep in mind that these figures will likely include the data seen in the figures above, as well as new data. If there is a desire to recreate figure 12, figure 13, and figure 14 directly from raw data you must filter data down to the date ranges found in the file where you downloaded all the data.

automizestoppingptsPsychomaticalFunction.m ˟

This file will read from featuretable, located in the database backup found in [Data Base Backup As 04-11-2023.zip.](https://dataverse.harvard.edu/file.xhtml?fileId=7060611&version=DRAFT)  For each trial in a session that a rat performs it will take the average measured stopping points, for each of the 4 feeder values. It will write these values to the table found in the database “stoppingptsPsychomaticalFunctions”.

1. createMap ()
   1. This function returns a Map T
      1. Keys are dates.
      2. The values are a list of all rats to run a session on this date.
   2. This function must be modified.
      1. The variable “query” located in line 3 of the file must be modified to exclude any experiment data.



Figure : Example query that filters out all experiment data.

1. createPsychomaticalFunction(searchResults)
   1. This function returns xcoordinates, ycoordinates.
   2. xcoordinates is the following array of values. [0.005, 0.01, 0.02, 0.09]
      1. Each of these values represents the percentage of glucose located in a feeder which is present during the trials.
   3. searchResults is a MATLAB in the following form.

|  |  |
| --- | --- |
| id | stoppingpts |
| 1 | 10000 |

Table Example of how the searchResults table must be formatted.

* 1. You can get a table like this by using the following query.
     1. date is a date that is formatted in MM-Shortened Month-YYYY format.
     2. animalsubjectid is any name of an animal in the table.

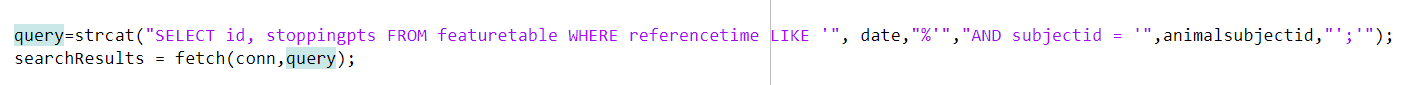


Table : Query that will return searchResults in a format that will be accepted by createPsychomaticalFunction().

1. Once the functions above are run they will automatically write the results to a table in the database named “stoppingptsPsychomaticalFunctions” in the format seen in table 8.

### stoppingptsSigmoidAnalysis.m

1. This function will read from the “stoppingptsPsychomaticalFunctions” table created by automizestoppingptsPsychomaticalFunction.m and fit the data there with a parabola, a sigmoid, and a line.
2. Coefficient of determination (R-Squared) is used to measure goodness of fit
3. Each row of data in the “travelPixelPsychomaticalFunctions” table will be sorted into folders based on which model fits it better, determined by which produces a higher R-Squared value.

After these functions are run, you can now run stoppingPtsMaxVsShift.m,

stoppingPtsMaxVsSteepness.m, and stoppingPtsShiftVsSteepness.m. This should create updated figures.

## Creating Baseline Rotation Points Figures

All of the codes mentioned in this section can be found in the following GitHub directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Rotation%20Points>

Chart, scatter chart

Description automatically generated

Figure : Rotation Points Max Vs Shift MPC 0.9300

Chart, scatter chart

Description automatically generated

Figure : Rotation Points Max Vs Steepness: 0.9422

Chart, scatter chart

Description automatically generated

Figure : Rotation Points Shift Vs Steepness MPC: 0.8946

To recreate the figures exactly: download all the files found at the following link.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Rotation%20Points/All%20Sigmoids>

Store these files in a folder on your local computer then run the following file.

### RotationPtsSigmoidClustering.m

1. Modify the “myDir” variable so it points to the file which houses all the data you downloaded, as shown in figure 8
2. Modify the fullFileName variable with the same path used in myDir variable, see figure 9 for an example
3. travelPixelSigmoidClustering.m will create a variable called newTable which contains all the data obtained from fitting the raw data with a sigmoid

Now run the following files.

### rotationPtsMaxVsShift.m

### rotationPtsMaxVsSteepness.m

### rotationPtsShiftVsSteepness.m

To create an updated figure from our raw data run the following functions. Keep in mind that these figures will likely include the data seen in the figures above, as well as new data. If there is a desire to recreate figure 18, figure 19, and figure 20 directly from raw data you must filter data down to the date ranges found in the file where you downloaded all the data.

### automizeRotationPtsPsychomaticalFunction.m

This file will read from featuretable, located in the database backup found in [Data Base Backup As 04-11-2023.zip.](https://dataverse.harvard.edu/file.xhtml?fileId=7060611&version=DRAFT)  For each trial in a session that a rat performs it will take the average measured rotation points, for each of the 4 feeder values. It will write these values to the table found in the database “rotationPtsPsychomaticalFunctions”.

1. This function works almost identically to automizeTravelPixelPsychomaticalFunction.m, but instead reference the following table instead of table 7 and figure 11.

|  |  |
| --- | --- |
| id | rotationPts |
| 1 | 10000 |

Table : Example of how the searchResults table must be formatted.

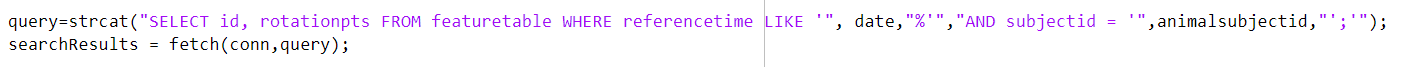


Table : Query that will return searchResults in a format that will work for createPsychomaticalFunction.

1. Once the function runs it will write results to a table in the database named “rotationPtsPsychomaticalFunctions”. See table 8 for the format.

### RotationPtsSigmoidAnalysis.m

1. This function will read from the “rotationPtsPsychomaticalFunctions” table created by automizeRotationPtsPsychomaticalFunction.m and fit the data there with a parabola, a sigmoid, and a line.
2. Coefficient of determination (R-Squared) is used to measure goodness of fit
3. Each row of data in the “rotationPtsPsychomaticalFunctions” table will be sorted into folders based on which model fits it better, determined by which produces a higher R-Squared value.

After these functions are run, you can now run rotationPtsMaxVsShift.m,

rotationPtsMaxVsSteepness.m, and rotationPtsShiftVsSteepness.m. This should create updated figures.

## Creating Baseline Reaction Time Figures

Chart, scatter chart

Description automatically generated

Figure : Reaction Time Max Vs Shift MPC: 0.8950

Chart, scatter chart

Description automatically generated

Figure : Reaction Time Max VS Steepness MPC: 0.9046

Chart, scatter chart

Description automatically generated

Figure : Reaction Time Shift Vs Steepness, MPC: 0.8575

To recreate the figures exactly download all the files found at the following link.

[**https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Reaction%20Time%201st/All%20Sigmoids**](https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Reaction%20Time%201st/All%20Sigmoids)

Store the files in a folder on your computer and run the following functions.

## reactionTime1stSigmoidClustering.m

1. Modify the “myDir” variable so it points to the file which houses all the data you downloaded, as shown in figure 8.
2. Modify the fullFileName variable with the same path used in myDir variable, see figure 9 for an example.
3. Automizereactiontime1stPsychomaticalFunction.m will create a variable called newTable which contains all the data obtained from fitting the raw data with a sigmoid.

Now run the following files.

## reactionTime1stMaxVsShift.m

## reactionTime1stMaxVsSteepness.m

## reactiontime1stShiftVsSteepness.m

To create an updated figure from our raw data run the following functions. These figures will include the data seen in the figures above, as well as new data. If you want to recreate figure 21, 22, 23 directly from raw data you must filter the data down to the date ranges found in the file where you downloaded all the data.

## Automizereactiontime1stPsychomaticalFunctions.m

This file will read from featuretable, located in the database backup found in [Data Base Backup As 04-11-2023.zip.](https://dataverse.harvard.edu/file.xhtml?fileId=7060611&version=DRAFT)  For each trial in a session that a rat performs it will take the average measured rotation points, for each of the 4 feeder values. It will write these values to the table found in the database “basePsychometricFunctions”.

|  |  |
| --- | --- |
| id | reactionTime |
| 1 | 20 |

1. This function works almost identically to automizeTravelPixelPsychomaticalFunction.m, but instead reference the following table instead of table 7 and figure 11.

Table : Example of how the searchResults table must be formatted for reactionTime1st.

A picture containing text, font, screenshot, purple

Description automatically generated

Figure : Query that will return searchResults in a format that will work for automizereactiontime1stPsychomaticalFunctions.m

1. Once the function runs it will write results to a table in the database named “basePsychometricFunctions”. See table 8 for the format.

## ReactionTime1stSigmoidAnalysis.m

1. This function will read from the “basePsychometricFunctions” table created by automizereactiontime1stPsychomaticalFunction.m and fit the data there with a parabola, a sigmoid, and a line.
2. Coefficient of determination (R-Squared) is used to measure goodness of fit.
3. Each row of data in the “basePsychometricFunctions” table will be sorted into folders based on which model fits it better, determined by which produces a higher R-Squared value.

After these functions are run, you can now run reactionTime1stMaxVsShift.m,

reactionTime1stMaxVsSteepness.m, and reactionTime1stShiftVsSteepness.m. This should create updated figures.

## Creating Reward Choice Figures

Chart, scatter chart

Description automatically generated

Figure : Reward Choice Max Vs Steepness, MPC: 0.9057

Chart, scatter chart

Description automatically generated

Figure : Reward Choice Max Vs Steepness, MPC: 0.9057

Chart, scatter chart

Description automatically generated

Figure : Reward Choice Max Vs Steepness, MPC: 0.9057

All the codes mentioned in this section can be found in the following GitHub directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Reward%20Choice>

To recreate the figures exactly, download all the files found at the following link.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Reward%20Choice/All%20Sigmoids>

Store these files in a folder on your local computer then run the following file.

## RewardChoiceSigmoidClustering.m

1. Modify the “myDir” variable so it points to the file which houses all the data that you downloaded. Refer to figure 8 for an example.
2. Modify the “fullFileName” variable with the same path used in myDir variable, refer to figure 9 for an example.
3. RewardChoiceSigmoidClustering.m will create a variable called newTable which contains all the data obtained from fitting the raw data with a sigmoid.

Now run the following files.

## rewardChoiceMaxVsShift.m

## rewardChoiceMaxVsSteepness.m

## rewardChoiceShiftVsSteepness.m

To create an updated figure from our raw data run the following functions, these updated figures will contain the old data and more. To recreate figures 26, 27, and 28 from raw data you must filter data down to the date ranges found in the file where you downloaded all the data.

## automizePsychomaticalFunction.m

This file will read from live\_table, located in the database backup found in [Data Base Backup As 04-11-2023.zip.](https://dataverse.harvard.edu/file.xhtml?fileId=7060611&version=DRAFT) For each trial in a session that a rat performs it will take the average measured reward choice for each of the 4 feeder values. It will write these values to the table found in the database “psychomaticalFunctions”.

This function works almost identically to automizeTravelPixelPsychomaticalFunction.m, but instead reference the following table instead of table 7 and figure 11.

A picture containing text, font, screenshot, line

Description automatically generated

Figure : Query that will return searchResults in a format that will work for automizePsychomaticalFunction.m

Once the function runs it will write results to a table in the database named “psychomaticalFunctions”. See table 8 for the format.

## sigmoidAnalysis.m

1. This function will read from “psychomaticalFunctions” table created by automizePsychomaticalFunction.m and fit the data there with a parabola, a sigmoid, and a line.
2. Coefficient of determination (R-Squared) is used to measure goodness of fit.
3. Each row of data in the “psychomaticalFunctions” table will be sorted into folders based on which model fits it better, determined by which produces a higher R-Squared value.

After these functions are run, you can now run rewardChoiceMaxVsShift.m, rewardChoiceShiftVsSteepness.m, and rewardChoiceMaxVsSteepness.m. This should create updated figures.

## Creating Average Sigmoid Figures for Baseline

A picture containing text, line, font, plot

Description automatically generated

Figure : Average Sigmoid for Baseline J1, J2, J3, and J4 clusters.

Download the following files into a local directory, for our purposes let’s call it directory 1.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Reaction%20Time%20Sigmoid%20Data>

Download the following GitHub directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Gender%20Differences%20Of%20Sigmoids/Old%20Baseline>

Run the following files.

### createAverage.m ˟

1. Modify the “newTable” to point to directory 1.
2. The figure should be automatically created.

## Creating Average Sigmoid For Food Deprivation

A picture containing text, line, font, plot

Description automatically generated

Figure : Average Sigmoid For Food Deprivation J1, J2,J3, and J4 clusters.

Download the following files into a local directory, for our purposes let’s call it directory 2.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Data%20Analysis/Old%20Base%20Data/Reaction%20Time%20Sigmoid%20Data>

Download the following GitHub Directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Gender%20Differences%20Of%20Sigmoids/Food%20Deprivation>

Run the following files.

### createAverage.m

1. Modify the “newTable” to point to directory 2.
2. The figure should be automatically created.

## Parabola Clustering

A picture containing text, line, diagram, plot

Description automatically generated

Figure : Parabola Clustering

To recreate this figure, download the following file.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Parabolas%20Analysis/Baseline>

Run the following file.

### getThreeParametersFromParabolas.m ˟

1. Uncomment the following lines of code.

A picture containing text, font, line, screenshot

Description automatically generated

Figure : Lines of code that need to be uncommented for getThreeParametersFromParabolas.m to work.

1. Run the file, and the figures should automatically be created.

## Venn Diagram

A picture containing text, screenshot, font

Description automatically generated

Figure : Venn Diagrams

To recreate these figures, follow the following steps.

1. Download all the files in the following GitHub directory in a folder, we will call this folder, folder 1.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Find%20Euclidian%20Distances%20From%20Baseline%20to%20Experiments/Create%20Probability%20Tables/Baseline>

1. Download all the files in the following GitHub directory in a folder, we will call this folder, folder 2.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Find%20Euclidian%20Distances%20From%20Baseline%20to%20Experiments/Create%20Probability%20Tables/Food%20Deprivation>

1. Download all the files in the following GitHub directory in a folder, we will call this folder, folder 3.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Find%20Euclidian%20Distances%20From%20Baseline%20to%20Experiments/Create%20Probability%20Tables/Ghrelin>

1. Now download “Create Multiple Venn Diagrams.ipynb” from the following GitHub directory.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Find%20Euclidian%20Distances%20From%20Baseline%20to%20Experiments>

1. Modify the following lines of code in “Create Multiple Venn Diagrams.ipynb” to point towards folder 1.

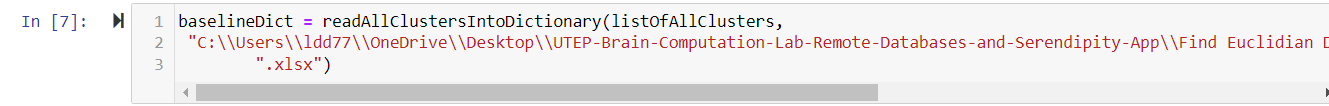


Figure : Code 1 that needs to be modified in Create Multiple Venn Diagrams to work.

1. Modify the following lines of code in “Create Multiple Venn Diagrams.ipynb” to point towards folder 2.

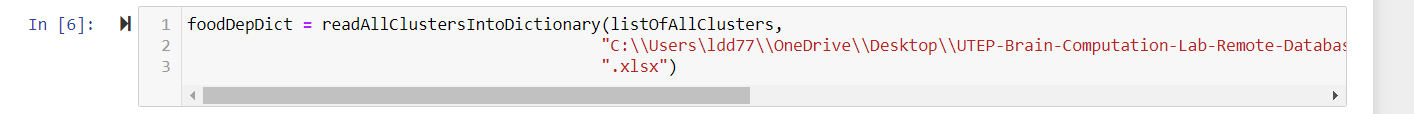


Figure :Code 2 that needs to be modified in Create Multiple Venn Diagrams to work.

1. Modify the following lines of code in “Create Multiple Venn Diagrams.ipynb” to point towards folder 3.

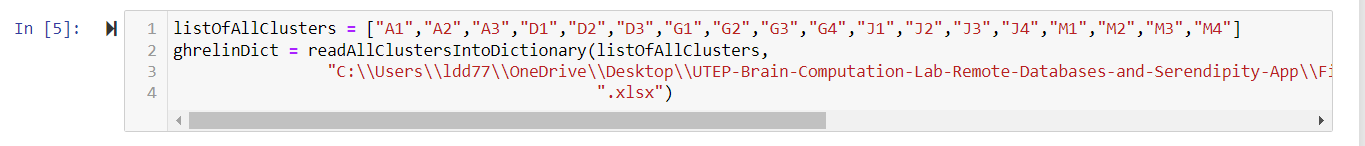


Figure :Code 3 that needs to be modified in Create Multiple Venn Diagrams to work.

1. Now you can run “Create Multiple Venn Diagrams.ipynb” and it will automatically create all the Venn diagrams used in our figures as well as many more which were not used.

## Probability Tables

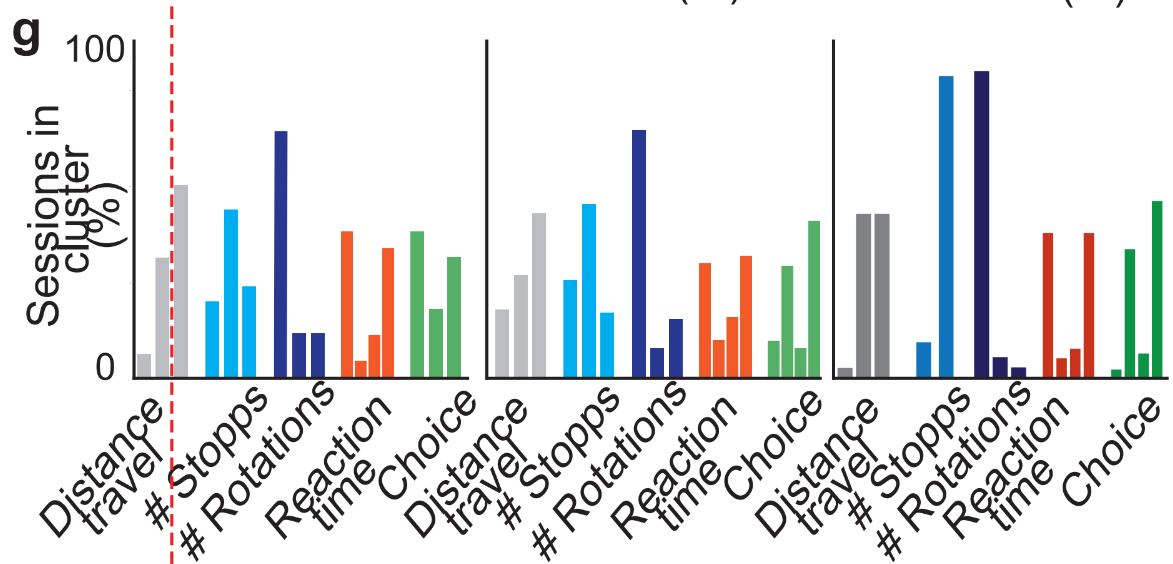


Figure : Probability Tables individuals

1. To recreate these probability charts, download the following file in GitHub.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Raw%20Figures/All%20Clusters>

1. Run the following file

### Create Rat Probabilities Chart with Probability Labels.ipynb

1. To have a measure of which charts are similar and which ones are different run the following file which uses Euclidian distance to determine “similar”.

### Get Rat Probabilities Euclidian Distance.ipynb

## Population Probabilities

A picture containing text, screenshot, font, diagram

Description automatically generated

Figure : Population Probabilities for Oxycodone and Alcohol.

A picture containing text, screenshot, font, diagram

Description automatically generated

Figure : Population Probabilities for Baseline and Food Deprivation

To recreate this figure, follow the proceeding steps.

1. Perform a Git Clone of the following GitHub Repository to your local computer.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App.git>

1. Navigate into the Create Probability Tables folder.
2. Modify the following file so that the “filePathWithData” variable points to the desired file on your local computer.

### collectAllDataInAMap.m

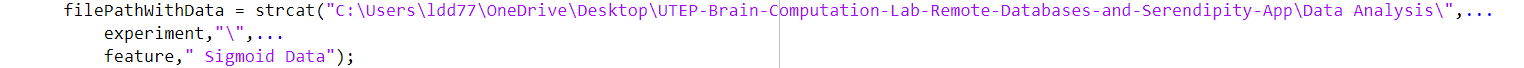


Figure : The line of code that needs to be modified in collectAllDataInAMap.m

1. It’s likely that the only modification you will need to do is to modify “C:\Users\ldd77\OneDrive\” to work with your local computer.
2. Now run the following files.

### get\_oxy\_probabilities.m

### get\_LG\_boost\_probabilities.m

### get\_Lg\_etoh\_probabilities.m

### createBaselineMaps.m

* Modify the following line of code to point to the same directory but stored in your local computer.

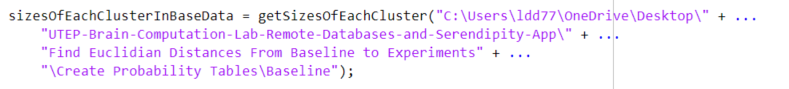


Figure : Code that needs to be modified for createBaselineMaps.m to run.

* The only thing you should need to modify is the “C:\Users\ldd77\OneDrive\Desktop” section to point to the correct directory in your local computer.

### createFoodDeprivationMaps.m

* Modify the following line of code to point to the same directory but stored on your local computer.

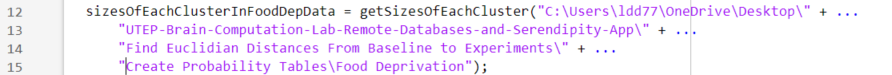


Figure : Code that needs to be modified for createFoodDeprivationMaps.m to work.

* The only thing you should need to modify is the “C:\Users\ldd77\OneDrive\Desktop” section to point to the correct directory in your local computer.

1. Now run the following files.

### Create Population Probability Charts.ipynb

### Create Gender Population Probability Charts.ipynb

## Hierarchal Tree

A picture containing text, screenshot, circle, diagram

Description automatically generated

Figure : Hierarchal Clustering Tree

To recreate this tree, follow the following steps.

1. Download the following files, which are available on GitHub. Each of these files is a cluster in the baseline Clustering. The link to GitHub follows the table below.

|  |
| --- |
| File Name |
| A1.xlsx |
| A2.xlsx |
| A3.xlsx |
| D1.xlsx |
| D2.xlsx |
| D3.xlsx |
| G1.xlsx |
| G2.xlsx |
| G3.xlsx |
| G4.xlsx |
| J1.xlsx |
| J2.xlsx |
| J3.xlsx |
| J4.xlsx |
| M1.xlsx |
| M2.xlsx |
| M3.xlsx |
| M4.xlsx |

Table : List of Cluster Files

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Baseline>

1. Each file has 3 columns of information, clusterLabels, cluserX, and clusterY.
2. To test which clusters overlap with each other simply write code that checks which clusters have the same clusterLabels.
3. Note that Clusters with the same leading letter, for example A1, and A2, will have no overlap.
4. You can save the labels that clusters share and repeat the process with the now smaller set of labels and a new cluster.
5. You can repeat this process until the overlap is none, and you can graph the tree using any tree creating software.
6. You can also check for gender of these overlaps by using the following table.
7. Codes to help with this process can be found at the following Github.

<https://github.com/lddavila/UTEP-Brain-Computation-Lab-Remote-Databases-and-Serendipity-App/tree/main/Raw%20Figures/Cluster%20Tables>

|  |  |
| --- | --- |
| Rat Name | Gender |
| Alexis | f |
| Kryssia | f |
| Harley | f |
| Raissa | f |
| Andrea | f |
| Fiona | f |
| Sully | m |
| Jafar | m |
| Kobe | m |
| jr | m |
| scar | m |
| jimi | m |
| sarah | f |
| raven | f |
| Shakira | f |
| Renata | f |
| Neftali | f |
| Juana | f |
| Mike | m |
| Aladdin | m |
| Carl | m |
| Simba | m |
| Johnny | m |
| Captain | m |
| Pepper | f |
| Buzz | m |
| ken | m |
| woody | m |
| slinky | m |
| rex | m |
| Trixie | f |
| Barbie | f |
| Bopeep | f |
| Wanda | f |
| Vision | f |
| Buttercup | f |
| Monster | f |

Table : Table of Rat Names to Gender. f = female. m = male.