

flag\_outliers

Contents

- Repo location
- Intro
- Usage
- Example 1
- Disclaimer
- Post usage recomendations
- Credits

Repo location

This function is contained within basic\_stats [https://gitlab.com/Fair\\_lab/basic\\_stats](https://gitlab.com/Fair_lab/basic_stats)

Intro

This function is designed to take a quick look to data that has been imported and saved in tidyformat using the function import\_tidyData\_with\_Dictionary. Once the data is imported in Matlab as table, this function (flag\_outliers) will identify outliers using matlab built in method to detect them.

Usage

This function takes as input a table (T) and look for outliers adfer grouping data using the columns specified by the second argument. OL detection is made using Matlab's built-in capabilities

```
% Input variables
% T, table with data
%
% columns_to_group_by, indicates the columns with variables to be used to
% group the data by
%
% columns_to_be_tested, indicates the columns with data to be analyzed  columns_to_be_tested, indicates the columns with data to be analyzed
```

Example 1

First load the companion data provided here:

```
load('tidyData.mat')
whos tidyData
```

Name	Size	Bytes	Class	Attributes
tidyData	129x9	55109	table	

Display the first 10 elements:

```
tidyData(1:10,:)
```

ans =

10x9 table

SubjectID	Sex	test	Freq	AMP	Rise_Time	Decay_Time	AREA	Treatment
'31287'	'M'	'PU IPSC'	12.087	51.08	3	5.206	295.07	ISO 3
'31503'	'F'	'PU IPSC'	9.6455	70.082	3.658	5.904	505.77	ISO 3
'32041'	'F'	'PU IPSC'	2.1437	23.422	4.7613	5.5883	157.68	ISO 3
'32050'	'F'	'PU IPSC'	4.996	38.635	5.0917	10.685	419.42	ISO 3
'32228'	'M'	'PU IPSC'	5.6527	42.254	4.4943	8.805	362.39	ISO 3
'32207'	'M'	'PU IPSC'	5.4553	37.677	5.441	10.515	411.63	ISO 3
'32269'	'M'	'PU IPSC'	3.2833	47.737	4.5743	8.9103	405.88	ISO 3
'31418'	'M'	'PU IPSC'	1.4387	28.497	4.8567	3.3103	138.8	Con
'31348'	'F'	'PU IPSC'	4.299	42.017	3.55	6.387	302.44	Con

```
'31970'      'M'      'PU IPSC'      4.5675      28.995      3.9813      7.1467      203.65      Con
```

This table will be the first argument to the function.

Now, let's suppose you want to see if there are outliers for the outcomes in the columns 4 to 8 (ie Freq, AMP, Rise\_Time, Decay\_Time, and Area, respectively) after grouping them by the columns 3 and 9 (ie test and treatment, respectively).

Based on this, the vales for columns\_to\_group\_by and columns\_to\_be\_tested are

```
columns_to_group_by=[3 9];% ie ie test and treatment, respectively
columns_to_be_tested=4:8;% ie Freq, AMP, Rise_Time, Decay_Time, and AREA, respectively
```

Now run the function and show the results

```
outliers_table = flag_outliers(tidyData,columns_to_group_by,columns_to_be_tested)
%
```

outliers\_table =

15x10 table

SubjectID	Sex	test	Freq	AMP	Rise_Time	Decay_Time	AREA	Treatment	outlier_var
'32227'	'M'	'PU EPSC'	1.231	22.284	3.908	3.985	127.79	Con	'Freq'
'32227'	'M'	'V1 IPSC'	8.2975	47.358	2.6685	1820.5	202.3	Con	'Decay_Time'
'32031'	'F'	'PU EPSC'	0.42233	35.591	2.0773	3.0573	86.565	ISO 1	'AMP'
'32031'	'F'	'PU IPSC'	2.4073	55.145	3.0883	5.104	266.8	ISO 1	'AMP'
'31349'	'M'	'SS IPSC'	1.2803	41.307	4.225	6.4637	925.15	ISO 1	'AREA'
'32134'	'M'	'V1 EPSC'	4.1283	18.523	3.0463	7.887	133.27	ISO 1	'Decay_Time'
'31349'	'M'	'V1 IPSC'	4.6985	132.98	4.011	13.793	448.99	ISO 1	'AMP'
'31349'	'M'	'V1 IPSC'	4.6985	132.98	4.011	13.793	448.99	ISO 1	'Decay_Time'
'31349'	'M'	'V1 IPSC'	4.6985	132.98	4.011	13.793	448.99	ISO 1	'AREA'
'31503'	'F'	'PU EPSC'	2.608	22.093	2.0885	0.867	34.443	ISO 3	'Freq'
'32041'	'F'	'SS EPSC'	3.8152	11.725	3.6947	6.9437	84.538	ISO 3	'AMP'
'31503'	'F'	'SS IPSC'	27.225	141.8	3.938	5.746	976.7	ISO 3	'Freq'
'31503'	'F'	'SS IPSC'	27.225	141.8	3.938	5.746	976.7	ISO 3	'AMP'
'31287'	'M'	'SS IPSC'	4.548	37.915	1.751	1.848	86.479	ISO 3	'Rise_Time'
'31287'	'M'	'V1 EPSC'	8.051	28.566	1.739	2.006	63.301	ISO 3	'AMP'

As you can see, the output of the function is a table that only contains the variables that look like outliers. The output table has the same columns as the input table plus an extra column to indicate which outcome (out of the columns\_to\_be\_tested) seems to be outlier

## Disclaimer

I (Oscar) have tested the function using only 2 columns for columns\_to\_group\_by. I have not had the time to test it with 1, 3 or more grouping variables (Jan 2019).

## Post usage recomendations

Always look at the data and determine whether you trust this function or not

## Credits

Code development: Oscar Miranda-Dominguez

First line of code: Jan 2019