**Statistical Monitoring Program Instructions.**

**3 Outliers**

The function **outlier\_check** can be used to detect outliers in continuous variables using one of 5 different methods.

Parameters to give the function:

1. **data**:

This should be in the form of a data frame with the participant id (must be numeric) in the first column, the site name/number in the second column and the variables to check in columns 3+

Data frames can be read in with the following code:

**options(stringsAsFactors = FALSE)**

**reg.data<-data.frame(read.table("STUDY12\_REG.txt", row.names=NULL, header=TRUE, sep="\t"))**

(This would read in a text file called *STUDY12\_REG.txt* and store it in the data frame *reg.data*.)

1. **n**

This is the number of standard deviations from the mean (or IQRs from the median) a value should fall to be considered “extreme” and be counted as an outlier.

Where the Mahalanobis distance option has been selected, the outliers are based on a comparison to the chi squared distribution. In this case, the value of n will be used to find the critical value. If outliers are to be the upper 2.5% set n as 0.975.

1. **method**

This is the method that should be used to detect outliers. The options are:

“sd” – The SD method selects any value that falls more than n SDs from the mean as an outlier.

“grubbs” – Grubbs test is similar, however the process is performed iteratively. The most extreme value (that is more than n SDs from the mean) is selected as an outlier and then removed. The mean and SD are then recalculated and the next most extreme value (more than n of the recalculated SDs from the recalculated mean) is removed. This process continues until there are no more extreme values. This may detect smaller outlying points which are not detected using the SD method because they have been masked by more extreme values which falsely inflate the mean and SD.

“Euclid” - finds multivariate outliers using the Normalised Euclidian distance. These distances are summed across all variables to give “d”. Points which fall more than n SDs of d from the mean d are counted as outliers.

"mahal" - finds multivariate outliers using the Mahalanobis distance. Similar to the Euclidian distance but this also takes into account correlations.

The d values are compared to a chi squared distribution to find the extreme points.

"IQR" – to be used with data that is non-normally distributed. Points are counted as outliers if they fall more than n times the interquartile range (IQR) from the median.

1. **trial.name**

The name of the trial. This will be used to label the output files. For example:

**trial.name<- “STUDY12”**

1. **normal.plot**

This declares whether a normal probability plot should be drawn for each variable. It should be set as TRUE if plots are required and FALSE if not.

**Calling the function**

Once the program and the parameters above are stored in R’s memory the program can be run using the following command:

**outlier\_check(data, n, method, trial.name, TRUE)**

Where each parameter is stored as in 1-5

**The output:**

**Plots:**

In the univariate options, each variable will be plotted: the id number on the x-axis and the value on the y-axis. Any points detected as outliers will be coloured red.



Above: plots for the same variable (LDH at randomisation) output using the SD method (left), the Grubbs method (middle) and the IQR method (right). Note the plots are identical but different numbers of outliers (red points) are detected. Plots have a name in the form: *“OUTLIERS\_SD\_METHOD\_study12\_LDHRAND\_2013-11-06.*tif”

Where SD is the method used (could be IQR or Grubbs with they were used), trial.name was set as study12, and LDHRAND was the variable name. The date is set as the date the program is run.

For multivariate methods, only one plot is created; this shows the d values (either Euclidean or Mahalanobis distances) plotted against the id number. Again, all outliers are shown in red. Examples for study 12 are shown below (left: Euclidean, right: Mahalanobis) :



These plots have names of the form:

*“OUTLIERS\_MAHAL\_METHOD\_study12\_2013-11-06.tif”*

The names are created in the same way but no variable name is given as they are multivariate.

Finally, if normal.plot has been set as TRUE the program outputs Normal probability plots for each variable so the Normality assumption can be checked. An example for LDH at randomisation below:

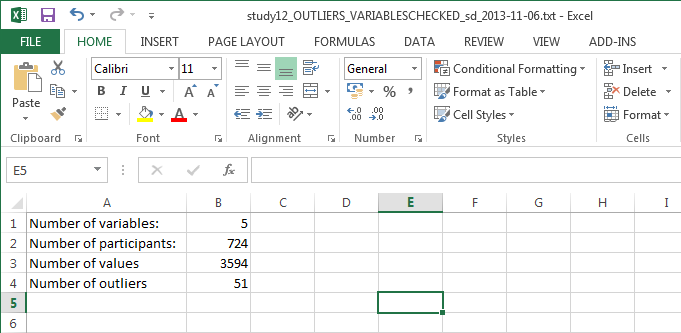


Plots names are all in the form:

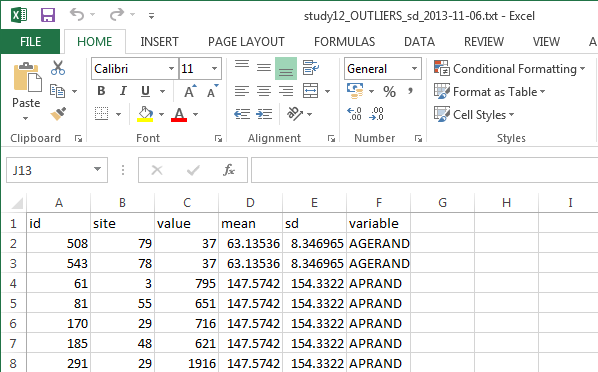
*“OUTLIERS\_SD\_METHOD\_study12\_LDHRAND\_2013-11-06.tif”*

**Text files**

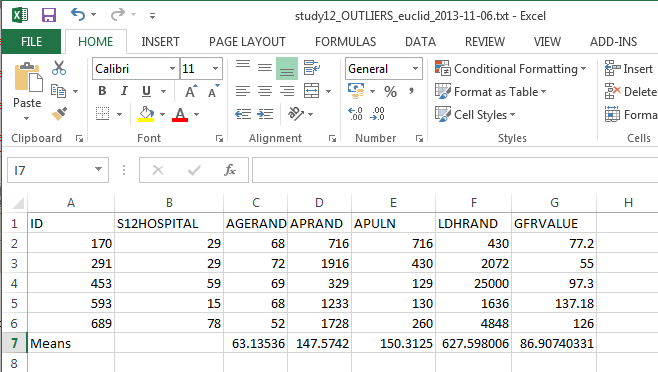
Each time the program is run it will produce a text file giving details on the variables checked and the number of outliers found (study 12 randomisation data using the SD method shown).



If outliers have been found then a second file, giving more details will also be output.



Above: study 12 randomisation data using the SD method. IQR and Grubbs will look similar, however if a multivariate method was used the data file will appear as below:

****

**Warnings:**

There are no error messages coded into the function. If data is not read in as above, the function may not work as it should, or possibly at all. Please take care when creating the parameters from your data.