# robustToxicities

This document goes through using this package. The example data is computer generated and does not come from a trial or study. The data for this example is time based data. This means that each toxicity has a start date and end date. The data must also provide start dates for each cycle or time period. This package can also deal with cycle based data. This data simply details what cycles a patient was on trial and what toxicities were present in each cycle (Hopefully I will add another example).

This document is generated directly from r markdown. Markdown is a lightweight markup language which uses plain text formatting syntax designed so that it can be principally converted to HTML. An r markdown file is a combination of markdown and code chunks which generate output to add to the report. Output includes generated markdown, code, tables and plots. This approach allows for a fully automated document whereby if the underlying data is changed the report will generate using this new data. Because markdown is lightweight it is fast and easy to use but comes with some significant limitations for complex output.

We use the knitr package to format our tables in markdown. The knitr package also supports latex (and lyx) which is a heavyweight markup language used to create pdfs. A latex based approach may be considerably more time consuming but is much more flexible.

# Time based data example

The following example is based on time data. The data set consists of toxicities with a start and stop date. Further information is provided about the cycle or time period start dates. This means tables can be produced by cycle and graphs can be produced using the toxicity durations.

Lets load packages we will be using today. knitr is available on CRAN and you can install robustToxicities from github directly: devtools::install\_github("finite2/robustToxicities"). Note than you may need to install the devtools package first. We then load the example data and have a look.

library(robustToxicities)

## Loading required package: stringr

## Loading required package: xtable

library(knitr)  
data = read.csv("Fake\_data\_time.csv", stringsAsFactors = FALSE)  
str(data, vec.len = 2)

## 'data.frame': 29 obs. of 20 variables:  
## $ patientNo : chr "PT01" "PT01" ...  
## $ Treatment : chr "Placebo" "Placebo" ...  
## $ toxicity : chr "Vomiting" "Vomiting" ...  
## $ category : chr "Gastrointestinal disorders" "Gastrointestinal disorders" ...  
## $ grade : int 1 2 1 1 3 ...  
## $ ae\_onset\_date : chr "04/01/2016" "22/01/2016" ...  
## $ ae\_resolve\_date : chr "10/01/2016" "27/01/2016" ...  
## $ ae\_cont\_end : logi FALSE FALSE FALSE ...  
## $ Registration\_date : chr "01/01/2016" "01/01/2016" ...  
## $ Randomisation\_date : chr "15/01/2016" "15/01/2016" ...  
## $ Cycle\_1\_date : chr "22/01/2016" "22/01/2016" ...  
## $ Cycle\_2\_date : chr "29/01/2016" "29/01/2016" ...  
## $ Cycle\_3\_date : chr "" "" ...  
## $ Cycle\_4\_date : chr "" "" ...  
## $ Cycle\_5\_date : chr "" "" ...  
## $ Cycle\_6\_date : chr "" "" ...  
## $ End\_cycle\_6\_date : chr "" "" ...  
## $ end\_of\_treatment\_date : chr "04/02/2016" "04/02/2016" ...  
## $ end\_of\_assessment\_date: chr "05/03/2016" "05/03/2016" ...  
## $ ass\_TRUE : logi TRUE TRUE TRUE ...

## Format data

The nameDatabase function does a simple renaming of the column names for our data to make it compatible for the rest of the package.

data = nameDatabase(  
 data,  
 patid = "patientNo",  
 treatment = "Treatment",  
 ae\_term = "toxicity",  
 ae\_system = "category",  
 ae\_grade = "grade",  
 ae\_start\_date = "ae\_onset\_date",  
 ae\_end\_date = "ae\_resolve\_date",  
 ae\_cont\_end\_study = "ae\_cont\_end",  
 date\_stopped\_treatment = "end\_of\_treatment\_date",  
 date\_end\_assessment = "end\_of\_assessment\_date",  
 dateColumnNames = c("Registration\_date", "Randomisation\_date", "Cycle\_1\_date", "Cycle\_2\_date", "Cycle\_3\_date", "Cycle\_4\_date", "Cycle\_5\_date", "Cycle\_6\_date")  
)

## Options

We need to specify the initial options for our tables. The default function defaultToxicityOptions simply asks for the metadata to be stored. Other options can be changed directly by accessing their slots using the @ operator for s4 classes.

options=defaultToxicityOptions(trialName="FAKEtrial", folderPath="", fileName = "Fake\_data.csv")

## Load data into the robustToxicities s4 class

We now define cycle and treatment labels for the output tables and run the initialize function. This function does considerable data cleaning and checking tasks to make sure the provided data is suitable for the other functions.

cycleLabels = c("Registration", "Randomisation", "Cycle 1", "Cycle 2", "Cycle 3", "Cycle 4", "Cycle 5", "Cycle 6")  
treatmentLabels = c("Placebo", "Fake Drug")  
  
toxDB = robustToxicities(data, cycleLabels, options, treatmentLabels)

## [1] 1 1 1 1 1 1 2 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 1 1 1 1 1 2 2

## Number of patients: 7 in the provided database

## There were 1 patients with no eligible toxicities

## ae\_cycle\_occured not provided, creating column and will populate it

## Patient: PT01 toxicity: Vomiting line: 6 was continuing at end of study, setting this value to date\_end\_assessment

## Patient: PT03 toxicity: Sneezing line: 15 was continuing at end of study, setting this value to date\_end\_assessment

## Patient: PT05 toxicity: Headache line: 26 was continuing at end of study, setting this value to date\_end\_assessment

##   
## #############################################################

## # Summary of preparation

## Number of patients: 7

## Number of patients with no toxicities: 1

## Patients missing date of end of treatment: 0

## Number of notes: 3

## Number of missing data problems: 0

## Number of incorrect data problems: 0

## Queries and notes

Assuming there were no major issues from the initialisation we may want to look at the generated notes and queries to see what changes and problems the data cleaning step picked up on.

kable(toxDB@queries)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| patid | ae | ae\_cycle\_occured | problem\_type | message |
|  |  |  | Affirmation | Number of patients: 7 in the provided database |
|  |  |  | Affirmation | There were 1 patients with no eligible toxicities |
| PT01 | Vomiting | NA | Note | Patient: PT01 toxicity: Vomiting line: 6 was continuing at end of study, setting this value to date\_end\_assessment |
| PT03 | Sneezing | NA | Note | Patient: PT03 toxicity: Sneezing line: 15 was continuing at end of study, setting this value to date\_end\_assessment |
| PT05 | Headache | NA | Note | Patient: PT05 toxicity: Headache line: 26 was continuing at end of study, setting this value to date\_end\_assessment |

# Tables

This package can generate two types of tables:

* A summary table detailing the toxicity grade by cycle (print\_toxTable\_summary)
* A cycle or time period table, detailing what the toxicities were and what grade they were in a specified cycle or cycles (print\_toxTable\_cycle)

You should note that due to the limitations of markdown these tables output via a markdown file do not contain treatment labels. If there is more than one treatment then these will have to be added by hand. Alternatively you can obtain these labels automatically using a latex based solution.

There are a number of options for both tables:

* discardBaseline A logical value used to determine if toxicities reported at baseline should be reported or not. The default is FALSE.
* tabulationMethod One of "worst" or "all" determining if all toxicity changes are counted or only the worst reported grade in a time period. The default is worst.
* tabulationPercent A logical value used to determine if toxicity tables should report counts (FALSE) or percentages (TRUE). The default is FALSE.
* cumulativeGrades A logical value used to determine whether toxicity grades should be reported cumulatively or not. Defaut is TRUE.

There are also some options specific to each table.

### Summary:

* sumCycleMerge Cycles to merge in the print\_toxTable\_summary. Use numeric values with | to divide the merged cycles and , to divide cycles in a merge e.g. "1,2|3,4,5" is two merged time periods with the first 2 time periods and the last 3 time periods.
* sumColumnMerge Grades to merge in the print\_toxTable\_summary. Similar syntax to sumCycleMerge.

### Cycle:

* cycleCycleMerge Cycles to merge in the print\_toxTable\_cycle. Similar syntax to sumCycleMerge.
* cycleColumnMerge Grades to merge in the print\_toxTable\_cycle. Similar syntax to sumCycleMerge.
* cycleCategoryMerge A vector of categories to collapse down to one row in the print\_toxTable\_cycle.

This section shows off some of the table options.

## Summary table by cycle with percentages

toxDB@options@tabulationPercent = TRUE  
kable(print\_toxTable\_summary(toxDB, printMethod = "table"))

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time period | Total | 1+ | 2+ | 3+ | 4+ | Total | 1+ | 2+ | 3+ | 4+ |
| Registration | 4 | 50 | 25 | 0 | 0 | 3 | 33 | 0 | 0 | 0 |
| Randomisation | 4 | 50 | 50 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| Cycle 1 | 4 | 75 | 50 | 25 | 0 | 3 | 100 | 67 | 0 | 0 |
| Cycle 2 | 3 | 67 | 33 | 33 | 0 | 2 | 50 | 50 | 0 | 0 |
| Cycle 3 | 2 | 50 | 0 | 0 | 0 | 2 | 100 | 0 | 0 | 0 |
| Cycle 4 | 2 | 50 | 50 | 0 | 0 | 2 | 50 | 50 | 0 | 0 |
| Cycle 5 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Cycle 6 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

## Summary table by cycle with counts, not cumulative

toxDB@options@tabulationPercent = FALSE  
toxDB@options@cumulativeGrades = FALSE  
kable(print\_toxTable\_summary(toxDB, printMethod = "table"))

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time period | Total | 1 | 2 | 3 | 4 and 5 | Total | 1 | 2 | 3 | 4 and 5 |
| Registration | 4 | 1 | 1 | 0 | 0 | 3 | 1 | 0 | 0 | 0 |
| Randomisation | 4 | 0 | 2 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| Cycle 1 | 4 | 1 | 1 | 1 | 0 | 3 | 1 | 2 | 0 | 0 |
| Cycle 2 | 3 | 1 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 |
| Cycle 3 | 2 | 1 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 |
| Cycle 4 | 2 | 0 | 1 | 0 | 0 | 2 | 0 | 1 | 0 | 0 |
| Cycle 5 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Cycle 6 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

## All toxicities (worst grade by patient)

kable(print\_toxTable\_cycle(toxDB, cycles = "all"))

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1 | 2 | 3 | 4 and 5 | 1 | 2 | 3 | 4 and 5 |
|  | Number of patients | 3 |  |  |  | 4 |  |  |  |
| Gastrointestinal disorders | Vomiting | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| Nervous system disorders | Headache | 0 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | Sore Throat | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |

## Baseline table with counts

toxDB@options@cumulativeGrades = TRUE  
cat("\n## ",toxDB@cycleLabels[1],"\n\n")

##   
## ## Registration

kable(print\_toxTable\_cycle(toxDB, cycles = 1))

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 3 |  |  |  | 4 |  |  |  |
| Gastrointestinal disorders | Vomiting | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Nervous system disorders | Headache | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sore Throat | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## All cycles in a for loop

for(i in 1:length(toxDB@cycleLabels)) {  
 cat("\n## ",toxDB@cycleLabels[i],"\n\n")  
 print(kable(print\_toxTable\_cycle(toxDB, cycles = i)))  
}

## Registration

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 3 |  |  |  | 4 |  |  |  |
| Gastrointestinal disorders | Vomiting | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Nervous system disorders | Headache | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sore Throat | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Randomisation

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 3 |  |  |  | 4 |  |  |  |
| Gastrointestinal disorders | Vomiting | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Nervous system disorders | Headache | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 |

## Cycle 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 3 |  |  |  | 4 |  |  |  |
| Gastrointestinal disorders | Vomiting | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Nervous system disorders | Headache | 3 | 2 | 0 | 0 | 2 | 1 | 1 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | Sore Throat | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |

## Cycle 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 2 |  |  |  | 3 |  |  |  |
| Gastrointestinal disorders | Vomiting | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Nervous system disorders | Headache | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

## Cycle 3

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 2 |  |  |  | 2 |  |  |  |
| Nervous system disorders | Headache | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

## Cycle 4

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 2 |  |  |  | 2 |  |  |  |
| Nervous system disorders | Headache | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

## Cycle 5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 1 |  |  |  | 1 |  |  |  |

## Cycle 6

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 1 |  |  |  | 1 |  |  |  |

## All toxicity changes in all cycles after time period

toxDB@options@discardBaseline = FALSE  
toxDB@options@tabulationMethod = "all"  
toxDB@treatmentLabels

[1] "Placebo" "Fake Drug"

toxDB@options@tabulationMethod = "worst"  
cat("\n## ",toxDB@cycleLabels[3],"\n\n")

## Cycle 1

print(kable(print\_toxTable\_cycle(toxDB, cycles = "all")))

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 2+ | 3+ | 4+ | 1+ | 2+ | 3+ | 4+ |
|  | Number of patients | 3 |  |  |  | 4 |  |  |  |
| Gastrointestinal disorders | Vomiting | 1 | 0 | 0 | 0 | 2 | 1 | 1 | 0 |
| Nervous system disorders | Headache | 3 | 3 | 0 | 0 | 3 | 2 | 1 | 0 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | Sore Throat | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |

## Merge differently

toxDB@options@cycleColumnMerge = "1,2|3,4,5"  
toxDB@options@tabulationMethod = "worst"  
kable(print\_toxTable\_cycle(toxDB, cycles = "all"))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 3+ | 1+ | 3+ |
|  | Number of patients | 3 |  | 4 |  |
| Gastrointestinal disorders | Vomiting | 1 | 0 | 2 | 1 |
| Nervous system disorders | Headache | 3 | 0 | 3 | 1 |
| Respiratory, thoracic and mediastinal disorders | Sneezing | 1 | 0 | 1 | 0 |
|  | Sore Throat | 2 | 0 | 2 | 0 |

## Merge the Respiratory category

toxDB@options@cycleCategoryMerge = "Respiratory, thoracic and mediastinal disorders"  
kable(print\_toxTable\_cycle(toxDB, cycles = "all"))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Category | Event Term | 1+ | 3+ | 1+ | 3+ |
|  | Number of patients | 3 |  | 4 |  |
| Gastrointestinal disorders | Vomiting | 1 | 0 | 2 | 1 |
| Nervous system disorders | Headache | 3 | 0 | 3 | 1 |
| Respiratory, thoracic and mediastinal disorders |  | 3 | 0 | 3 | 0 |

## Latex output

To output to latex simply change the printMethod to latex

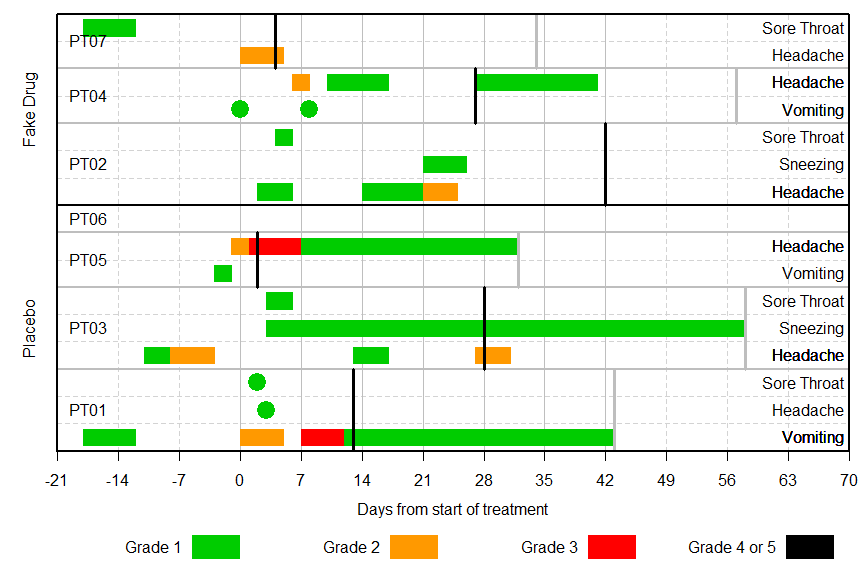
print\_toxTable\_summary(toxDB, printMethod = "latex")

## % latex table generated in R 3.2.4 by xtable 1.8-2 package  
## % Thu Apr 28 13:32:59 2016  
## \begin{table}[ht]  
## \centering  
## \begin{tabular}{|r|ccccc|ccccc|}  
## \hline  
## & \multicolumn{5}{c|}{Placebo}& \multicolumn{5}{c|}{Fake Drug}\\  
## Time period & Total & 1+ & 2+ & 3+ & 4+ & Total & 1+ & 2+ & 3+ & 4+ \\   
## \hline  
## Registration & 4 & 2 & 1 & 0 & 0 & 3 & 1 & 0 & 0 & 0 \\   
## Randomisation & 4 & 2 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\   
## Cycle 1 & 4 & 3 & 2 & 1 & 0 & 3 & 3 & 2 & 0 & 0 \\   
## Cycle 2 & 3 & 2 & 1 & 1 & 0 & 2 & 1 & 1 & 0 & 0 \\   
## Cycle 3 & 2 & 1 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\   
## Cycle 4 & 2 & 1 & 1 & 0 & 0 & 2 & 1 & 1 & 0 & 0 \\   
## Cycle 5 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\   
## Cycle 6 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\   
## \hline  
## \end{tabular}  
## \end{table}

# Plots

Plots can be output directly into your word document using a code chunk like below:

toxDB@options@plotLeftSideOption = "patid"  
toxDB@options@plotStartTreatment = "cycle\_start\_date\_3"  
toxDB@options@plotCycleLength = 7  
toxDB@options@plotxMin = -21  
toxDB@options@plotxMax = 70  
toxDB@options@plotLeftSideOption = "both"  
toxPlot(toxDB)

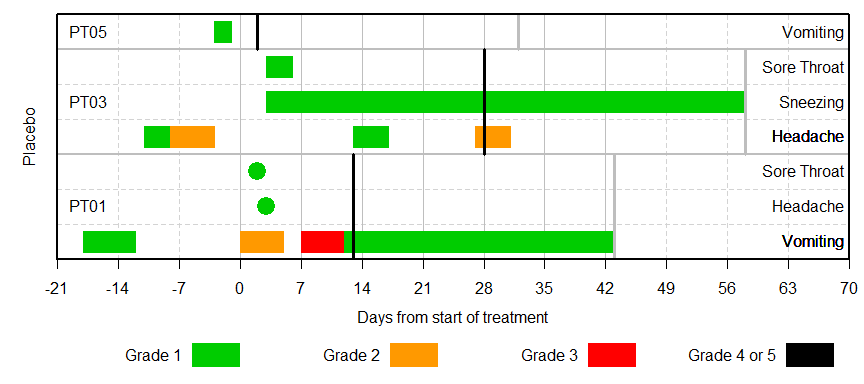


Toxicity data may require a large number of rows. For this reason it may be impractical to have all the data on a single graph output. To aid you here there is a function to output in pages to pdf and png (toxPlot\_output). Additionally if you set the plot option to FALSE then toxPlot will return the number of rows of data. This will allow you to decide how much data to plot on each graph and the rowID\_range allows you to define what to plot on each graph.

toxPlot(toxDB,plot = FALSE)

## [1] 16

toxPlot(toxDB, rowID\_range = c(1,7))



toxPlot(toxDB, rowID\_range = c(8,16))

