

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.

Setup

Lets load packages we will be using today:

1. If necessary install devtools from CRAN `install.packages('devtools')`. This allows you to install packages from non-CRAN sources such as github.
2. Install the package from github `devtools::install_github("finite2/robustToxicities")`.
3. load the packages we will be using, `robustToxicities` `library(robustToxicities)` and `knitr` `library(knitr)` (to install from CRAN `install.packages('knitr')`).

```
devtools::install_github("finite2/robustToxicities")
library(robustToxicities)
library(knitr) # for outputting tables to word or latex
```

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.

```
## The standard way to read in data
## data = read.csv("data.csv", stringsAsFactors = FALSE)

# Data from this package
data(toxicityDataTime)
data = toxicityDataTime
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. We need to provide the following data for time based data:

- `patid`
- `treatment` (unless only one treatment arm)
- `ae_term`, name of the toxicity
- `ae_system`, category the toxicity belongs
- `ae_grade`, the toxicities grade
- `ae_start_date`, the date of first onset of this grade of the toxicity
- `ae_end_date`, the date this toxicity changed grade
- `ae_cont_end_study`, if the end date was after the patient is no longer assessed as part of the study
- `date_stopped_treatment`, the date a patient withdraw from treatment
- `date_end_assessment`, the date the patient is no longer assessable
- start dates, the starting dates for time periods or cycles for this patient on trial. E.g. registration date, cycle 1 start date, etc.

```
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",  
treatmentLabels = c("Placebo", "Fake Drug")
```

```
toxDB = robustToxicities(data, cycleLabels, options, treatmentLabels)
```

```
## ae_cycle_occured not provided, creating column and will populate it
```

```
## Patient PT06 is missing toxicity grade for line 27 (currently set to zero)
```

```
## Patient: PT01 toxicity: Vomiting line: 6 was continuing at end of study, setting this value to date_
```

```
## Patient: PT03 toxicity: Sneezing line: 15 was continuing at end of study, setting this value to date_
```

```
## Patient: PT05 toxicity: Headache line: 26 was continuing at end of study, setting this value to date_
```

```
## Number of patients: 7 in the provided database
```

```
## There were 1 patients with no eligible toxicities
```

```
##
```

```
## #####
```

```
## # 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: 4
```

```
## 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	problem_type	message
PT06		Note	Patient PT06 is missing toxicity grade for line 27 (currently set to zero)
PT01	Vomiting	Note	Patient: PT01 toxicity: Vomiting line: 6 was continuing at end of study, setting this v
PT03	Sneezing	Note	Patient: PT03 toxicity: Sneezing line: 15 was continuing at end of study, setting this v
PT05	Headache	Note	Patient: PT05 toxicity: Headache line: 26 was continuing at end of study, setting this
		Affirmation	Number of patients: 7 in the provided database
		Affirmation	There were 1 patients with no eligible toxicities

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. Default 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

Time period	Total	1+	2+	3+	4+	Total	1+	2+	3+	4+
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
print_toxTable_summary(toxDB, printMethod = "latex")
```

% latex table generated in R 3.3.0 by xtable 1.8-2 package % Tue May 10 11:16:42 2016

Time period	Placebo					Fake Drug				
	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	4				3			
Gastrointestinal disorders	Vomiting	1	0	1	0	1	0	0	0
Nervous system disorders	Headache	1	1	1	0	0	3	0	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	4				3			
Gastrointestinal disorders	Vomiting	1	0	0	0	0	0	0	0

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

Randomisation

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

Cycle 1

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

Cycle 2

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

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

Latex output

To output to latex simply change the printMethod to latex

```
print_toxTable_summary(toxDB, printMethod = "latex")
```

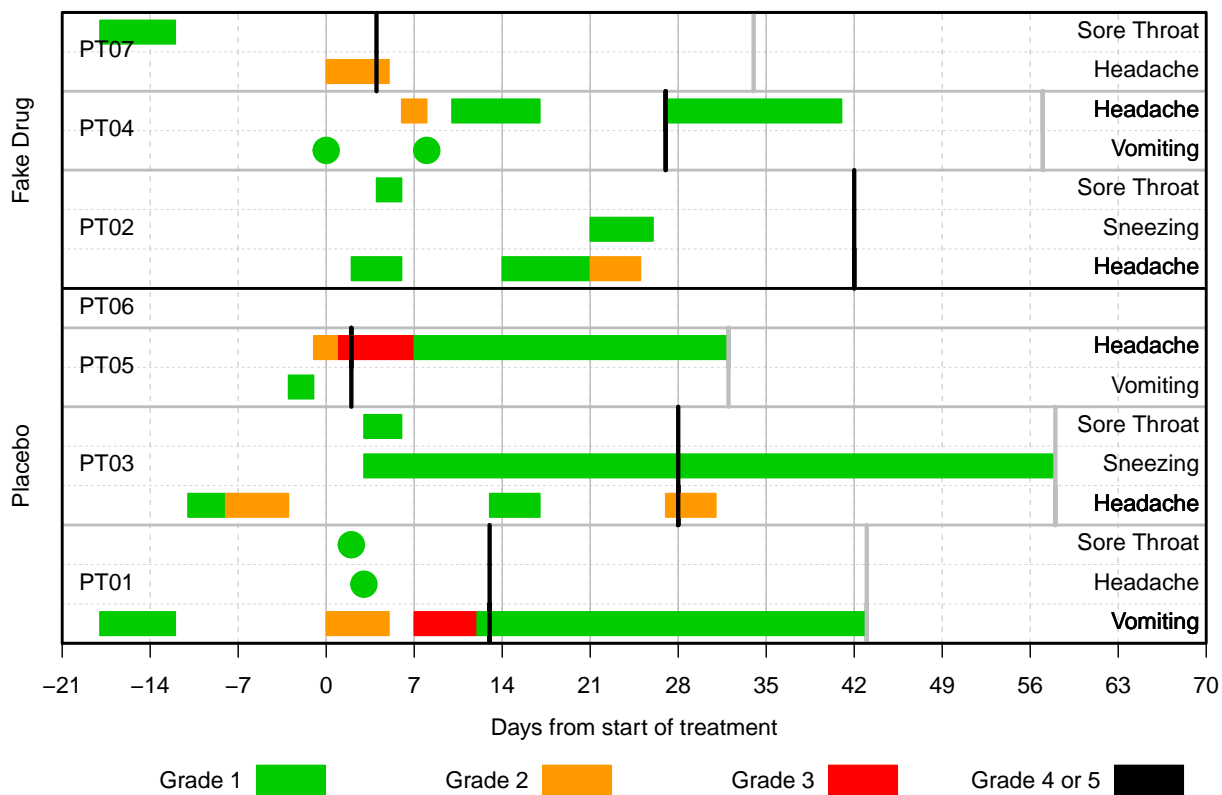
% latex table generated in R 3.3.0 by xtable 1.8-2 package % Tue May 10 11:16:42 2016

Time period	Placebo					Fake Drug				
	Total	1+	2+	3+	4+	Total	1+	2+	3+	4+
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

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)
```

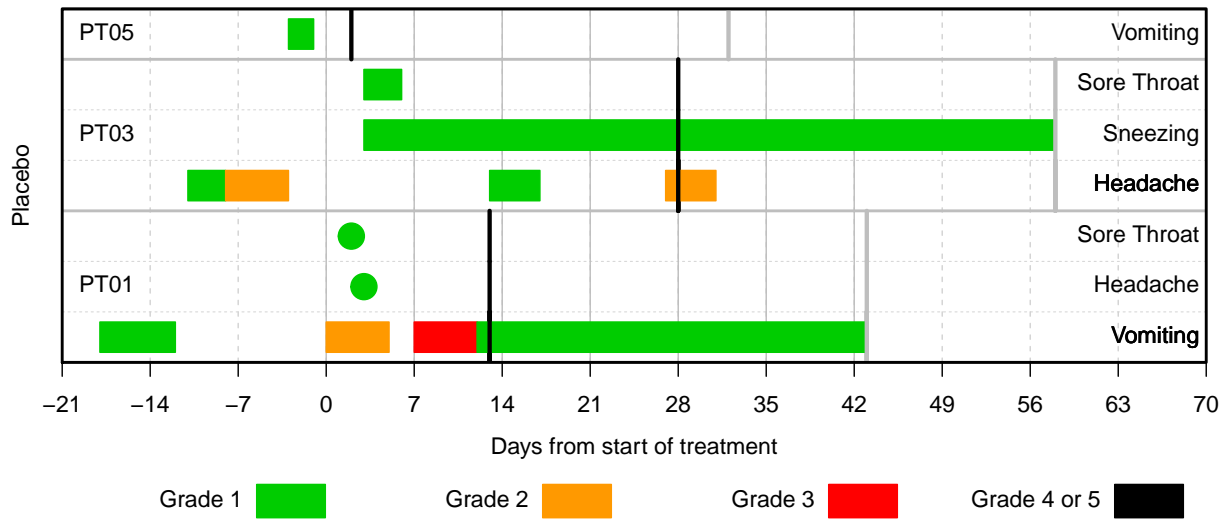


Toxicity data may require a large number of rows. For this reason it may be impractical to have all the data on a single graph output. To aid you here there is a function to output in pages to pdf and png (`toxPlot_output`). Additionally if you set the plot option to `FALSE` then `toxPlot` will return 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.

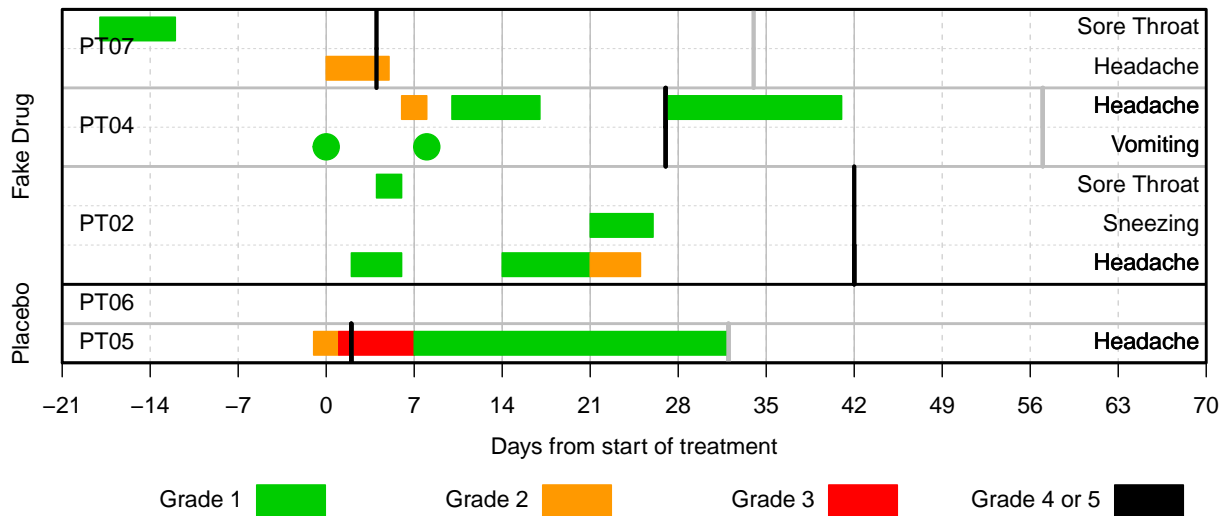
```
toxPlot(toxDB, plot = FALSE)
```

```
## [1] 16
```

```
toxPlot(toxDB, rowID_range = c(1,7))
```



```
toxPlot(toxDB, rowID_range = c(8,16))
```



Cycle based data example

For cycle based data slightly different data is required. The data should be in a wide format by cycle / time period. We need to provide the following data for cycle based data:

- patid
- treatment (unless only one treatment arm)
- ae_term, name of the toxicity
- ae_system, category the toxicity belongs
- patientInCycle, whether a patients was on trial for this cycle
- occurInCycle, the grade a patient experienced for this cycle. Should be zero if they didn't experience this toxicity.

Because some of the names are already correct we don't need `nameDatabase` to rename these for us.

```
data(toxicityDataCycle)
str(data)
```

```
## 'data.frame': 29 obs. of 20 variables:
## $ patid : chr "PT01" "PT01" "PT01" "PT01" ...
## $ treatment : chr "Placebo" "Placebo" "Placebo" "Placebo" ...
## $ ae_term : chr "Vomiting" "Vomiting" "Sore Throat" "Headache" ...
## $ ae_system : chr "Gastrointestinal disorders" "Gastrointestinal disorders" "Respirator" ...
## $ ae_grade : int 1 2 1 1 3 1 1 1 1 1 ...
## $ ae_start_date : chr "04/01/2016" "22/01/2016" "24/01/2016" "25/01/2016" ...
## $ ae_end_date : chr "10/01/2016" "27/01/2016" "24/01/2016" "25/01/2016" ...
## $ ae_cont_end_study : logi FALSE FALSE FALSE FALSE FALSE TRUE ...
## $ cycle_start_date_1 : chr "01/01/2016" "01/01/2016" "01/01/2016" "01/01/2016" ...
## $ cycle_start_date_2 : chr "15/01/2016" "15/01/2016" "15/01/2016" "15/01/2016" ...
## $ cycle_start_date_3 : chr "22/01/2016" "22/01/2016" "22/01/2016" "22/01/2016" ...
## $ cycle_start_date_4 : chr "29/01/2016" "29/01/2016" "29/01/2016" "29/01/2016" ...
## $ cycle_start_date_5 : chr "" "" "" "" ...
## $ cycle_start_date_6 : chr "" "" "" "" ...
## $ cycle_start_date_7 : chr "" "" "" "" ...
## $ cycle_start_date_8 : chr "" "" "" "" ...
## $ End_cycle_6_date : chr "" "" "" "" ...
## $ date_stopped_treatment : chr "04/02/2016" "04/02/2016" "04/02/2016" "04/02/2016" ...
## $ date_end_assessment : chr "05/03/2016" "05/03/2016" "05/03/2016" "05/03/2016" ...
## $ ass_TRUE : logi TRUE TRUE TRUE TRUE TRUE TRUE ...
```

```
data = nameDatabase(
  data = toxicityDataCycle,
  ae_term = "toxicity",
  ae_system = "category",
  patientInCycle = c("active_registration", "active_randomisation", "active_cycle_1", "active_cycle_2",
  occurInCycle = c("tox_registration", "tox_randomisation", "tox_cycle_1", "tox_cycle_2", "tox_cycle_3"
)
```

```
options=defaultToxicityOptions(trialName="FAKEtrial", folderPath="", fileName = "Fake_data_cycle.csv",
```

```
cycleLabels = c("Registration", "Randomisation", "Cycle 1", "Cycle 2", "Cycle 3", "Cycle 4", "Cycle 5",
treatmentLabels = c("Placebo", "Fake Drug")
```

```
toxDB = robustToxicities(data, cycleLabels, options, treatmentLabels)
```

```
## Number of patients: 7 in the provided database
```

```
## There were 1 patients with no eligible toxicities
```

```
##
```

```
## #####
```

```
## # Summary of preparation
```

```
## Number of patients: 7
```

```
## Number of patients with no toxicities: 1
```

```
## Number of notes: 0
```

```
## Number of missing data problems: 0
```

```
## Number of incorrect data problems: 0
```

Tables and plots

The table outputs rely on identical code for both time and cycle based data. This is because for time based data the `patientInCycle` and `occurInCycle` columns are generated from the date data. The plots do have independent code but share plot options.

```
cat("\n\n## Toxicity summary")
```

Toxicity summary

```
kable(print_toxTable_summary(toxDB))
```

Time period	Total	1+	2+	3+	4+	Total	1+	2+	3+	4+
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
End of Cycle 6	1	0	0	0	0	1	0	0	0	0
30 days	4	2	0	0	0	3	1	0	0	0

```
for(i in 1:length(toxDB@cycleLabels)) {  
  cat(paste0("\n\n## ",toxDB@cycleLabels[i]))  
  print(kable(print_toxTable_cycle(toxDB, cycles = i)))  
}
```

Registration

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

Randomisation

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

Cycle 1

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

Cycle 2

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

End of Cycle 6

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

30 days

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

```
toxDB@options@plotxMin = 0
toxDB@options@plotxMax = 10
toxDB@options@plotCycles = 0
toxDB@options@plotLeftSideOption = "both"
toxPlot(toxDB)
```

