secuTrialR package vignette

Patrick R. Wright

2019-05-21

This R package provides functions for handling data from the clinical data management system (CDMS) secuTrial. The most important components are related to loading data exports from secuTrial into R. In brief, the package aims to allow swift execution of repetitive tasks in order to allow spending more time on the unique aspects of a dataset.

For this vignette and also to test your installation of the secuTrialR package some example export data is delivered with the package. The data (*calcium*) was retrieved from the **lava** package and and extended with some made up data (ID 999). Subsequently the dataset was prepared for import into a secuTrial electronic case report form (eCRF). After import, the data was again exported and added to the secuTrialR package. The exports processed in the following are non-rectangular.

Load the secuTrial library

```
library(secuTrialR)
```

Introduction to the dataset

To give you a brief impression of the dataset it will be briefly shown here.

```
bone_mineral_dinsity_loc <- system.file("extdata",
                                        "calcium_secuTrial.csv",
                                        package = "secuTrialR")

bmd_data <- read.table(file = bone_mineral_dinsity_loc, sep = ";", header = TRUE)

head(bmd_data[,c("patid", "visitdate", "bmd.bmd", "bmd.grouping", "bmd.age")])
#>   patid  visitdate bmd.bmd bmd.grouping  bmd.age
#> 1   101 30.04.1930   0.815      Calcium 10.91307
#> 2   101 04.11.1930   0.875      Calcium 11.42779
#> 3   101 23.04.1931   0.911      Calcium 11.89322
#> 4   101 29.10.1931   0.952      Calcium 12.41068
#> 5   101 28.04.1932   0.970      Calcium 12.90897
#> 6   102 30.04.1930   0.813      Placebo 10.91307
tail(bmd_data[,c("patid", "visitdate", "bmd.bmd", "bmd.grouping", "bmd.age")])
#>     patid  visitdate bmd.bmd bmd.grouping  bmd.age
#> 499   430 23.07.1931   1.024      Calcium 12.14237
#> 500   430 25.02.1932   1.054      Calcium 12.73648
#> 501   430 26.07.1932   1.071      Calcium 13.15264
#> 502   999 23.02.2019      NA      Calcium 45.10000
#> 503   999 24.02.2019      NA              45.10000
#> 504   999 25.02.2019   0.872                   NA
table(bmd_data$bmd.grouping)
#>
#>         Calcium Placebo
#>       2     246     256
```

```
unique(length(bmd_data$patid))
#> [1] 504
```

As you can see there are 504 unique patients grouped into "Calcium" and "Placebo". The patient with id 999 was subsequently added to the original dataset. This record has some missing values. These values are missing on purpose, since missing values are needed to demonstrate some downstream functionalities of the package.

The central files this vignette will be using are part of the package. Since the paths end up being a little cryptic we will store them in the following two variables. Both zip archives are secuTrial exports of the data above.

```
export_location_shortnames <- system.file("extdata",
                                          "s_export_CSV-xls_BMD.zip",
                                           package = "secuTrialR")

export_location_longnames <- system.file("extdata",
                                          "s_export_CSV-xls_longnames_BMD.zip",
                                          package = "secuTrialR")
```

Loading an export

The code below shows how to load a secuTrial export. The output of read_secuTrial_export is a list.

```
sT_export_shortnames <- read_secuTrial_export(data_dir = export_location_shortnames)
typeof(sT_export_shortnames)
#> [1] "list"
sT_export_longtnames <- read_secuTrial_export(data_dir = export_location_longnames)
typeof(sT_export_longtnames)
#> [1] "list"
```

The first element of this list is another list containing the options of the export. The export options are followed by data frames containing meta data tables and finally the central study data tables. At this point you can also see how exports with long names and short names differ.

```
names(sT_export_shortnames)
#>  [1] "export_options" "fs"             "cn"             "ctr"
#>  [5] "is"             "qs"             "qac"            "vp"
#>  [9] "vpfs"           "atcn"           "atcvp"          "cts"
#> [13] "bmd"            "atbmd"
names(sT_export_longtnames)
#>  [1] "export_options"   "forms"            "casenodes"
#>  [4] "centres"          "items"            "questions"
#>  [7] "queries"          "visitplan"        "visitplanforms"
#> [10] "atcasenodes"      "atcasevisitplans" "comments"
#> [13] "dem00bmd"         "atmnpdem00bmd"
```

The study data for the bmd secuTrial form is stored in bmd for the short table names and dem00bmd for the long table names. As expected they are the same if compared, since the only difference is the formatting of the table names.

```
head(sT_export_shortnames$bmd[,c("pat_id", "age", "grouping", "bmd")], n = 4)
#>   pat_id      age grouping   bmd
#> 1    101 10.91307  Calcium 0.815
```

```
#> 2    101 11.42779  Calcium 0.875
#> 3    101 11.89322  Calcium 0.911
#> 4    101 12.41068  Calcium 0.952

all.equal(sT_export_shortnames$bmd, sT_export_longtnames$dem00bmd)
#> [1] TRUE
```

Analysing data completeness

For various reasons eCRFs are frequently only partially filled. This can be commonly observed in cohort studies of registries. Thus, before performing more thorough analyses it is sometimes informative to get an overview of data completeness.

To asses data completeness of a secuTrial export you first need to generate the validation overview in secuTrial and export it as an *.xlsx file. Remember to select the "Column" and "Completion status" columns before you export the validation overview. The validation overview for this vignette is also part of this package.

```
val_ovv_location <- system.file("extdata",
                                "bmd_validation_overview.xlsx",
                                package = "secuTrialR")
```

In order to asses variable completeness we first require a loaded secuTrial validation overview and a loaded secuTrial export. Both the validation overview and the data export should be from the same time (not more than minutes apart if possible) to ensure data integrity.

The secuTrial export has already been loaded above. The code below shows how to load the validation overview. The validation overview contains a line for every variable with a missing or erroneous value. As explained earlier, patient 999 was added with missing values. As you can see the validation report only contains entries for "Patient" 999.

```
val_ovv <- read_validation_overview(data_dir = val_ovv_location)
val_ovv[,c("Patient", "Completion status", "Column")]
#> # A tibble: 5 x 3
#>   Patient `Completion status` Column
#>   <chr>   <chr>               <chr>
#> 1 999     partly filled       bmd
#> 2 999     partly filled       bmd
#> 3 999     partly filled       grouping
#> 4 999     partly filled       age
#> 5 999     partly filled       grouping
```

First we will asses the completeness only for "savedforms". Below you see that "age" is missing once while "grouping" and "bmd" are both missing twice just as reported in the validation overview.

```
assess_form_variable_completeness(form = sT_export_longtnames$dem00bmd,
                                  casenodes_table = sT_export_longtnames$casenodes,
                                  validation_overview = val_ovv,
                                  completeness = "savedforms")
#>   variable timesentered timesmissing completeness
#> 1      age          503            1    0.9980159
#> 2      bmd          502            2    0.9960317
#> 3 grouping          502            2    0.9960317
```

If you would like to know the completeness for all forms also taking forms into account which have not been saved, then you need to set the completeness parameter to "allforms". Note that the occ_in_vp parameter has also been set to five. This is because the patients in the study were followed up, up to five times (i.e. five

visits). Thus, if a patient was followed up less then five times his or her data is not complete. To set this parameter correctly, knowledge of the study design is necessary. As expected the completeness is reduced when regarding all forms.

```
assess_form_variable_completeness(form = sT_export_longtnames$dem00bmd,
                                  casenodes_table = sT_export_longtnames$casenodes,
                                  validation_overview = val_ovv,
                                  completeness = "allforms",
                                  occ_in_vp = 5)
#>   variable timesentered timesmissing completeness
#> 1      age          503           62    0.8902655
#> 2      bmd          502           63    0.8884956
#> 3 grouping          502           63    0.8884956
```