Package 'secuTrialR'

July 23, 2025

Type Package

Title Handling of Data from the Clinical Data Management System 'secuTrial'

Version 1.3.3

Maintainer Alan G. Haynes <alan.haynes@unibe.ch>

Description Seamless and standardized interaction with data exported from the clinical data management system (CDMS) 'secuTrial'<https://www.secutrial.com>. The primary data export the package works with is a standard non-rectangular export.

Depends R (>= 3.5)

Imports haven (>= 2.2.0), readr, readxl, stringr, tibble, magrittr, purrr, tidyr, dplyr, rlang, lubridate

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Suggests knitr, lintr, rmarkdown, testthat, tufte, tcltk, igraph

VignetteBuilder knitr

URL https://github.com/SwissClinicalTrialOrganisation/secuTrialR

BugReports https://github.com/SwissClinicalTrialOrganisation/secuTrialR/issues

NeedsCompilation no

Author Alan G. Haynes [cre, aut, cph], Patrick R. Wright [aut, cph], Milica Markovic [aut, cph], Gilles Dutilh [ctb], Armando Lenz [ctb], Pascal Benkert [com], Silvia Grieder [ctb], Pia Neuschwander [ctb], Constantin Sluka [ctb]

Repository CRAN

Date/Publication 2024-05-30 15:20:02 UTC

Contents

annual_recruitment . . . 2
as.data.frame.secuTrialdata . . . 3
assess_form_variable_completeness . . . 5
build_secuTrial_url . . . 6
check_export_options . . . 8
dates_secuTrial . . . 9
dictionary_secuTrial . . . 10
diff_secuTrial . . . 11
export_datetime . . . 12
factorize . . . 12
form_status_counts . . . 13
form_status_summary . . . 14
get_participants . . . 15
labels_secuTrial . . . 15
links_secuTrial . . . 17
plot_recruitment . . . 19
read_secuTrial . . . 20
read_validation_overview . . . 21
return_hidden_items . . . 21
return_random_participants . . . 22
return_scores . . . 23
secuTrialdata . . . 24
subset_secuTrial . . . 25
visit_structure . . . 26
write_secuTrial . . . 27

Index **29**

`annual_recruitment` *Returns a data.frame showing the annual recruitment per center*

Description

secuTrial exports inherently contain the information on which participant was registered at which point in time. This function makes use of this property to show annual recruitment.

Usage

```
annual_recruitment(x, rm_regex = "")
```

Arguments

`x`	a `secuTrialdata` object
`rm_regex`	character - specifies a regular expression to be removed from the centre names in the legend. e.g. rm_regex = "\\(.*\\)$" will remove trailing brackets and their contents.

Value

a data.frame showing the annual recruitment counts per center

Note

This function wraps plot_recruitment to retrieve the data.

Examples

```
# export location
expot_loc <- system.file("extdata", "sT_exports", "lnames",
                         "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                         package = "secuTrialR")
# read export
sT_export <- read_secuTrial(expot_loc)

# plot recruitment
annual_recruitment(sT_export)

# show without trailing bracket
annual_recruitment(sT_export, rm_regex = "\\(.*\\)$")
```

`as.data.frame.secuTrialdata`

as.data.frame method for secuTrialdata objects Make the data from the exports more easily accessible by placing them in another environment (e.g. place them in the global environment (`.GlobalEnv`*) and you can reference them without referring to the* `secuTrialdata` *object anymore. Ie. they become regular* `data.frame`*s).*

Description

as.data.frame method for secuTrialdata objects Make the data from the exports more easily accessible by placing them in another environment (e.g. place them in the global environment (`.GlobalEnv`) and you can reference them without referring to the `secuTrialdata` object anymore. Ie. they become regular `data.frame`s).

Usage

```
## S3 method for class 'secuTrialdata'
as.data.frame(
  x,
  ...,
  envir,
  data.frames = NULL,
  meta = FALSE,
  regex = NULL,
```

```
  rep = ""
)
```

Arguments

x	secuTrialdata object
...	further parameters
envir	environment in which to put the data (e.g. .GlobalEnv)
data.frames	character vector of data.frame names to turn into data.frames. If the vector is named,data.frames will be renamed into the non-empty names(data.frames)
meta	logical should metadata be returned
regex	regex syntax to remove from names
rep	replacement for regex

Details

envir must be specifically defined. For simplicity, .GlobalEnv would probably be the easiest (assigning it to another environment would still entail referring to that environment).

Value

each data.frame in the secuTrialdata object is saved to it's own data.frame in the designated environment

Examples

```
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)
# add files to a new environment called env1
env1 <- new.env()
as.data.frame(sT_export, envir = env1)
# add files to a new environment called env2, removing the project name from
# the file names
env2 <- new.env()
as.data.frame(sT_export, regex = "ctu05", envir = env2)
# add files to a new environment called env3, renaming files in data.frames
env3 <- new.env()
as.data.frame(sT_export, data.frames = c("allmedications" = "ctu05allmedi", "ctu05baseline"),
              envir = env3)
```

`assess_form_variable_completeness` *Asses completeness of data for each variable in a secuTrial export*

Description

NOTE: This is not exported currently since it is not generic enough. It can still be used but in depth knowledge of the function is required to evaluate if the result is correct. As a rule of thumb: rigid CDMA setups with low complexity will likely work as expected.

Variable completeness is defined as percentage of data entered for a variable when considering all occurrences of the variable throughout the visit plan and all participants registered in the secuTrial data base. The completeness can be assessed for 'allforms', which will also take unsaved forms into account, and for 'savedforms' which will only consider forms that have been actively saved. In order to asses variable completeness the function requires a loaded secuTrial validation overview and a loaded secuTrial standard 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. Variable completeness is considered based on rules in the secuTrial setup. Thus, if a variable is not marked as necessary for completeness, this variable will always be considered as 100 percent complete if only 'savedforms' are assessed. Please note that variable completeness for repetition forms should only be assessed for saved forms.

Usage

```
assess_form_variable_completeness(
  form,
  casenodes_table,
  validation_overview,
  completeness = "allforms",
  occ_in_vp = 1,
  omit_mnp = TRUE
)
```

Arguments

`form`	data.frame Form for which to assess variable completeness (i.e. a list element returned by read_secuTrial_raw).
`casenodes_table`	data.frame The central casenodes record file (i.e. 'casenodes' from the list returned by read_secuTrial_raw).
`validation_overview`	tibble returned by read_validation_overview.
`completeness`	string Specifies if completeness is assessed for all forms ('allforms') or only for saved forms ('savedforms').
`occ_in_vp`	integer Specifies how often the form occurs in the visit plan (only relevant if completeness = 'allforms' is set).
`omit_mnp`	boolean Removes variable names from the result table that start with mnp.

Value

data.frame showing percent completeness per variable.

See Also

read_validation_overview, read_secuTrial_raw

Examples

```
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "BMD",
                               "s_export_CSV-xls_BMD_short_en_utf8.zip",
                               package = "secuTrialR")
# read all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)

# read validation overview
val_ovv_location <- system.file("extdata", "sT_exports", "BMD",
                                "bmd_validation_overview.xlsx",
                                package = "secuTrialR")
val_ovv <- read_validation_overview(data_dir = val_ovv_location)

secuTrialR:::assess_form_variable_completeness(form = sT_export$bmd,
                                               casenodes_table = sT_export$cn,
                                               validation_overview = val_ovv,
                                               completeness = "allforms",
                                               occ_in_vp = 5)
```

---

build_secuTrial_url *Compose a secuTrial URL*

Description

Given a secuTrial server URL, and optionally instance, customer, project id and document id, this function composes a URL to a specific secuTrial instance, customer or form.

Usage

```
build_secuTrial_url(
  server,
  instance = NA,
  customer = NA,
  projid = NA,
  docid = NA,
  prefix = "apps/WebObjects/"
)
```

Arguments

`server`	string containing a server URL
`instance`	(optional) string containing secuTrial instance name
`customer`	(optional) string containing secuTrial customer label
`projid`	(optional) string containing secuTrial project identifier
`docid`	(optional) secuTrial document/form identifer
`prefix`	(optional) string containing the URL prefix to the WebObjects application (default: "apps/WebObjects/")

Details

To find the server and instance of a secuTrial database, simply extract the information from the URL that you usually use to log in. For example in:

https://server.secutrial.com/apps/WebObjects/ST21-setup-DataCapture.woa/wa/choose?customer=TES

- server id is: **server.secutrial.com**
- instance id is: **ST21-setup-DataCapture**
- url prefix is: **apps/WebObjects/**
- you can find the customer id at the end of the link i.e **TES**
  Alternatively it can be found in the CustomerAdminTool -> click "Kunden" -> Value in DB column of table on login page.
- you can find the project id in the AdminTool -> Projects -> left side ID column in the table.
- you can find docids in secuTrial exports as values in the "mnpdocid" column.
  Alternatively they are noted in the form footers of the DataCapture.

Also note that only the server address has to be provided, the other arguments are optional. Thus, there are different scenarios:

- if only server address is provided, the output will point to the secuTrial server
- if secuTrial server and instance are provided, the output will point to the secuTrial instance.
- if secutrial server, instance and customer id are provided, the output will point to the customer page.
- if secuTrial server, instance, customer, project and document id are provided, the output will point to a specific secuTrial form.

Value

string containing a URL to desired secuTrial page. Currently we provide no guarantee that the returned URL is valid.

Examples

```
# This example, builds pseudo-urls that do not point to an active secuTrial instance.

server <- "server.secutrial.com"
instance <- "ST21-setup-DataCapture"
customer <- "TES"
project <- "7036"
docid <- "181"

build_secuTrial_url(server)
build_secuTrial_url(server, instance)
build_secuTrial_url(server, instance, customer)
build_secuTrial_url(server, instance, customer, project)
build_secuTrial_url(server, instance, customer, project, docid)

# examples of docids (mnpdocid)
path <- system.file("extdata", "sT_exports", "lnames",
                    "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                    package = "secuTrialR")
sT_export <- read_secuTrial(path)

# return docids
docids <- sT_export$ctu05baseline$mnpdocid

# make several links with all docids
build_secuTrial_url(server, instance, customer, project, docids)
```

---

check_export_options *Returns deviations from suggested export options*

Description

Given a secuTrialdata object, this function returns information on deviations from suggested export options.

Usage

```
check_export_options(dat)
```

Arguments

dat	secuTrialdata object

Details

While the package strives to allow loading of as many types of secuTrial data exports as possible, there are certain export options which are less likely to cause issues. If possible it is suggested to export data which adheres to a suggested option set. This function points out deviations from the

suggested set of options which are:
is_zip == TRUE
refvals_separate == TRUE
add_id == TRUE
duplicate_meta == FALSE
encoding == "UTF-8"
form_status == TRUE
centre_info == TRUE
proj_setup == TRUE
dict_items$lang == "en"
hidden_fields == FALSE
structure == TRUE

Examples

```
path <- system.file("extdata", "sT_exports", "exp_opt",
                    "s_export_CSV-xls_CTU05_only_column_names.zip",
                    package = "secuTrialR")
sT_export <- read_secuTrial_raw(path)

secuTrialR:::check_export_options(sT_export)
```

---

dates_secuTrial *Methods to handle date(times)s in secuTrial exports*

Description

Converts dates and datetime variables to `Date` or `POSIXct` class, as appropriate.

Usage

```
dates_secuTrial(object, ...)

## S3 method for class 'secuTrialdata'
dates_secuTrial(object, ...)
```

Arguments

`object`	`secuTrialdata` object
`...`	further parameters

Details

New variables are created appended with `.date` or `.datetime`. This is a safety mechanism in case NAs are inadvertently introduced.

Value

same as the original object with date variables converted to `Dates`.

Examples

```
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)
# prepare dates
sT_export_dates <- dates_secuTrial(sT_export)

# show parsed datetime example
sT_export_dates$ctu05baseline$hiv_date.datetime[1]
# [1] "2019-03-05 23:56:00 CET"
# show parsed date example
sT_export_dates$ctu05baseline$paracetamol_start.date[1]
# [1] "2019-03-05"
```

---

dictionary_secuTrial *Data dictionary for secuTrialdata objects*

Description

Data dictionary for secuTrialdata objects

Usage

```
dictionary_secuTrial(x)
```

Arguments

x	secuTrialdata object

Value

data frame with info on variables

Examples

```
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# read all export data
sT_export <- read_secuTrial(data_dir = export_location)
# get dictionary
dictionary_secuTrial(sT_export)
```

diff_secuTrial *Returns differences in the setup of two* secuTrialdata *objects*

Description

During ongoing studies it is possible that changes to the DataCapture interface are made. Sometimes these changes may call for adjustments in analysis code. It is considered good practice to run `diff_secuTrial()` on the last export and the current export of a project to at least make yourself aware of potential changes to the setup. If there are differences, the results of this function should be interpreted as a first indicator since they may not cover all alterations. Information is returned on new forms and variables. A detailed list of changes can be produced in the FormBuilder with "Compare project setup".

Usage

```
diff_secuTrial(x, y)
```

Arguments

x	a secuTrialdata object (the older export)
y	a secuTrialdata object (the newer export)

Value

If there are differences, `diff_secuTrial()` will produce a list of vectors. The fist vector informs about new forms and the second vector informs about new variables.

Examples

```
# read exports

# v1 is essentially a clone of the CTU05 setup
ctu06_v1 <- read_secuTrial(system.file("extdata", "sT_exports", "change_tracking",
                                       "s_export_CSV-xls_CTU06_version1.zip",
                                       package = "secuTrialR"))
# v2 contains 2 additional forms (mnpctu06anewform, mnpctu06anothernewform) and
# 2 additional variables (new_item_in_fu, new_item_in_new_form)
ctu06_v2 <- read_secuTrial(system.file("extdata", "sT_exports", "change_tracking",
                                       "s_export_CSV-xls_CTU06_version2.zip",
                                       package = "secuTrialR"))
# return diff
diff_secuTrial(ctu06_v1, ctu06_v2)
```

export_datetime *Return the date and time that an export was performed*

Description

Return the date and time that an export was performed

Usage

```
export_datetime(x)

export_date(x)
```

Arguments

x	secuTrial export

Value

`export_date` returns a date object, `export_datetime` returns a string

Examples

```
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# read all export data
sT_export <- read_secuTrial(data_dir = export_location)
# get date/time
export_datetime(sT_export)
export_date(sT_export)
```

factorize *Add factors to* `secuTrialdata` *objects*

Description

secuTrial can return a codebook of codes and labels for categorical variables, including lookup type variables, if this option is selected in the export tool ('reference values as separate table'). This allows factors to be easily created. Factorize methods exist for `secuTrialdata` objects, `data.frames`, `integers` and `logicals`, but the intent is that only the former be used by users. The other methods could be used with customized codebooks.

Usage

```
factorize_secuTrial(object, ...)

## S3 method for class 'secuTrialdata'
factorize_secuTrial(object, ...)
```

Arguments

object	secuTrialdata object with additional factor variables in study forms containing categorical data
...	further parameters

Details

factorize_secuTrial will return an error if the appropriate codebook is not available.

Value

factorized secuTrialdata object

Examples

```
# load secuTrial export with separate reference table
sT_export <- read_secuTrial_raw(system.file("extdata", "sT_exports", "lnames",
                                    "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                                           package = "secuTrialR"))
# factorize the secuTrialdata object
sT_export_factorized <- factorize_secuTrial(sT_export)
```

---

form_status_counts *A function to assess the status of forms*

Description

This function returns a data.frame informing on the status of saved forms per participant. There is a line for every combination of form type and participant id. The numbers are occurrence counts.

Usage

```
form_status_counts(object, ...)
```

Arguments

object	secuTrialdata object
...	further parameters

Value

data.frame informing on the status of saved forms per participant

Examples

```
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "snames",
                               "s_export_CSV-xls_CTU05_short_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial(data_dir = export_location)

# get form status
form_status_counts(sT_export)
```

`form_status_summary` *A function to show summary statistics for form statuses*

Description

This function warps form_status_counts and returns a data.frame summarizing the statuses for each form. Only saved forms are considered for the statistic.

Usage

```
form_status_summary(object, ...)
```

Arguments

`object`	`secuTrialdata` object
`...`	further parameters

Value

data.frame summarizing the statuses for each form

Examples

```
# prepare path to example export
export_location <- system.file("extdata","sT_exports", "snames",
                               "s_export_CSV-xls_CTU05_short_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial(data_dir = export_location)

# get form status
form_status_summary(sT_export)
```

get_participants *Retrieves participants present in* secuTrialdata

Description

Given a secuTrialdata object, this function simply returns a list of participants. Information included are participant IDs and corresponding study centre information, if available.

Usage

```
get_participants(dat)
```

Arguments

dat	secuTrialdata object containing participant IDs and centre information

Value

data.frame containing participants present in dat

Examples

```
path <- system.file("extdata", "sT_exports", "exp_opt",
                    "s_export_CSV-xls_CTU05_all_info.zip",
                    package = "secuTrialR")
sT_export <- read_secuTrial(path)

# show participants
participants <- get_participants(sT_export)
```

labels_secuTrial *Get variable labels for secuTrialdata objects*

Description

Variable labels are important for understanding the contents of a variable. secuTrialR offers two main methods to get those labels. labels_secuTrial returns a named list of labels. label_secuTrial adds labels and units to variables (and data.frames) which can then be queried via label or units.

Usage

```
labels_secuTrial(object, form = NULL)

label_secuTrial(object, ...)

label(x)

units(x)

label(x) <- value

units(x) <- value
```

Arguments

object	a secuTrialdata object
form	which form (string)
...	further parameters
x	any object
value	any object

Details

For labels_secuTrial, regular expressions are used with form (specifically, it is inserted between (and)$ to identify the form). Consequently, if form matches multiple forms (because the beginning is different), multiple forms may be returned. You could be more specific with the regular expression, remembering that it is inserted between (and)$.

Value

labels_secuTrial returns a named vector label_secuTrial returns the same object as object, but with labels added to variables and data.frames label and units return strings with the appropriate labels

secuTrialdata object with labels applied to each variable

Note

The label_secuTrial/label syntax is similar to that used in Hmisc, with the advantage that it does not change data types (Hmisc coerces everything to labelled integer). Similar to Hmisc, however, most operations will remove the labels.

Examples

```
# APPROACH 1: labels_secuTrial
# ex. 1
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "BMD",
                               "s_export_CSV-xls_BMD_short_en_utf8.zip",
```

```
                                  package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)
# get all labels
labels <- labels_secuTrial(sT_export)
labels[["age"]]

# ex. 2
# load export
sT_export <- read_secuTrial_raw(system.file("extdata", "sT_exports", "lnames",
                                            "s_export_CSV-xls_CTU05_long_miss_en_utf8.zip",
                                               package = "secuTrialR"))

# get labels for sae, treatment and surgeries forms
labels <- labels_secuTrial(sT_export, form = c("sae", "treatment", "surgeries"))


# APPROACH 2: label_secuTrial
# load secuTrial export with separate reference table
sT_export <- read_secuTrial_raw(system.file("extdata", "sT_exports", "lnames",
                                        "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                                               package = "secuTrialR"))
# label the secuTrialdata object
sT_export_labelled <- label_secuTrial(sT_export)
# form label
label(sT_export_labelled$ctu05baseline)
# variable label
label(sT_export_labelled$ctu05baseline$visit_date)
# sampling units
units(sT_export_labelled$ctu05baseline$height)
```

`links_secuTrial` *Show links between forms*

Description

secuTrial creates a large number of files and identifiers with which to link files together. Understanding the links can be difficult. This function produces a map linking the forms with common variables.

Usage

```
links_secuTrial(
  object,
  forms = NULL,
  formcol = "#d8b365",
  varcol = "#e5f5f9",
  plot = TRUE
)
```

Arguments

object	secuTrialdata object
forms	a regular expression for which forms should be included
formcol	color for form name circles
varcol	color for variable name circles
plot	boolean specifies if the plot should be shown

Details

We recommend to resize the tcltk window and and click view/"fit to screen" to improve readability. Forms are colored dull orange, variables are colored light blue.

Value

a tcltk plot window.

Note

Note that where a form name is also a variable name, it is appended by _form (igraph requires uniquely named nodes).

Examples

```
# ex. 1
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "BMD",
                               "s_export_CSV-xls_BMD_short_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)
# plot links
links_secuTrial(sT_export)

# ex. 2
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)
# plot links for form names starting with "ctu05"
links_secuTrial(sT_export, forms = "^ctu05")
```

plot_recruitment *Plots the recruitment over time for* secuTrialdata *objects*

Description

secuTrial exports inherently contain the information on which participant was registered at which point in time. This function makes use of this property to plot recruitment over time. Centers indicated with a black line in the legend did not recruit at all.

Usage

```
plot_recruitment(
  x,
  return_data = FALSE,
  show_centres = TRUE,
  cex = 1,
  rm_regex = ""
)
```

Arguments

x	a secuTrialdata object
return_data	logical - return the data used to produce the plot instead of the plot
show_centres	logical - subset the data into centres
cex	double - specifies font size in legend
rm_regex	character - specifies a regular expression to be removed from the centre names in the legend. e.g. rm_regex = "\\(.*\\)$" will remove trailing brackets and their contents.

Value

a simple line plot showing recruitment over time or a list of data.frames if return_data is set to TRUE

Examples

```
# export location
expot_loc <- system.file("extdata", "sT_exports", "lnames",
                         "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                         package = "secuTrialR")
# read export
sT_export <- read_secuTrial(expot_loc)

# plot recruitment
plot_recruitment(sT_export)

# plot without trailing bracket
```

```
plot_recruitment(sT_export, rm_regex = "\\(.*\\)$")
```

`read_secuTrial` *Read secuTrial export*

Description

Convenience wrapper for `read_secuTrial_raw`, `label_secuTrial`, `factorize_secuTrial` and `dates_secuTrial`.

Usage

```
read_secuTrial(data_dir, labels = TRUE, factor = TRUE, dates = TRUE, ...)
```

Arguments

`data_dir`	string - location of the export
`labels`	logical - add labels to variables and table
`factor`	logical - convert categorical variables to factor variables (ignored when reference values are not in a separate table)
`dates`	logical - convert date variables
`...`	options passed to `read_secuTrial_raw`. Can be used for passing options to read.table

Value

`secuTrialdata` object - a list with one data.frame for each file on the export and a list containing the export options

Examples

```
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# read all export data
sT_export <- read_secuTrial(data_dir = export_location)
```

`read_validation_overview` *This function loads a multi-page secuTrial 'Validation Overview' report into an R tibble.*

Description

This function loads a multi-page secuTrial 'Validation Overview' report into an R tibble.

Usage

```
read_validation_overview(data_dir, skip = 7)
```

Arguments

`data_dir`	Path to the Validation Overview (must be an *.xlsx file).
`skip`	Equivalent parameter in read_excel(). The validation overview xlsx files contain some information in the first few lines of each sheet which need to be skipped in order to produce the correct header in R. Prior to reading the validation overview with read_validation_overview() it is likely a good idea to check how many lines need to be skipped. This parameter has been added because the amount of lines can differ between different versions of secuTrial.

Value

tibble with the 'Validation Overview' data

Examples

```
val_ovv_location <- system.file("extdata", "sT_exports", "BMD",
                                "bmd_validation_overview.xlsx",
                                package = "secuTrialR")
val_ovv <- read_validation_overview(data_dir = val_ovv_location)
```

`return_hidden_items` *Returns hidden items (variables) from* `secuTrialdata` *objects*

Description

Sometimes, during a study, certain fields may be hidden because data should no longer be entered into them. If this is the case and the data of these fields is part of your export is likely good to know about it.

Usage

```
return_hidden_items(x)
```

Arguments

x	a secuTrialdata object

Value

a data.frame (columns: name, itemtype, label) that pinpoints which items are hidden

Examples

```
# export location
expot_loc <- system.file("extdata", "sT_exports", "lnames",
                         "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                         package = "secuTrialR")
# read export
sT_export <- read_secuTrial(expot_loc)

# return scores
return_hidden_items(sT_export)
```

---

return_random_participants

Returns the random participants from a secuTrial export

Description

There are situations (e.g. randomized monitoring) in which you may want to return a list of random participants from a secuTrial export.

Usage

```
return_random_participants(
  x,
  centres = "all",
  percent = 0.1,
  date = "1900-01-01",
  seed = 1
)
```

Arguments

x	a secuTrialdata object
centres	A character vector of centres for which participants should be returned. If left unspecified it will return participants for every study centre.
percent	A number greater than 0 and smaller than 1 specifying the approximate percentage of participants to be returned per centre.

date	If only participants after a specific date should be considered this can be entered here. Format should be "YYYY-MM-DD" (e.g. "2011-03-26" for March 26th 2011). This date is checked against mnpvisstartdate in the casenodes table.
seed	Allows to configure a seed for id sampling. Every centre will use a small variation of this seed.

Value

list of two elements. First, a data.frame that contains the random participants from each specified centre. This is performed based on a specified seed to retain reproducibilty. Second, the configuration of the randomization (i.e. result of `RNGkind()`). If the percentage does not yield an integer for a centre the number is tranformed into an integer under application of the `ceiling()` function (i.e. it is rounded up).

Examples

```
# export location
expot_loc <- system.file("extdata", "sT_exports", "lnames",
                         "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                         package = "secuTrialR")
# read export
sT_export <- read_secuTrial(expot_loc)

# return random participants
return_random_participants(sT_export, percent = 0.25, seed = 1337, date = "2019-03-18",
                       centres = c("Inselspital Bern (RPACK)", "Charité Berlin (RPACK)"))
```

---

return_scores *Returns the score (calculated) items from* secuTrialdata *objects*

Description

secuTrial allows to set up calculated fields (i.e. scores) that depend on other items. It is not suggested to use the scores calculated by secuTrial to perform reliable analyses. To this end, calling `return_scores` will return all items in the secuTrial export which are scores and should be manually recalculated before data analysis.

Usage

```
return_scores(x)
```

Arguments

x	a secuTrialdata object

Value

a data.frame (columns: name, itemtype, label) that pinpoints which items are scores/calculated.

Examples

```
# export location
expot_loc <- system.file("extdata", "sT_exports", "lnames",
                         "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                         package = "secuTrialR")
# read export
sT_export <- read_secuTrial(expot_loc)

# return scores
return_scores(sT_export)
```

`secuTrialdata` *This function loads a secuTrial export*

Description

This function will always load the full set of meta and data tables. The export options are also loaded and written into export_options.

Usage

```
read_secuTrial_raw(data_dir, ...)

## S3 method for class 'secuTrialdata'
print(x, ...)
```

Arguments

`data_dir`	string The data_dir specifies the path to the secuTrial data export.
`...`	further parameters
`x`	secuTrialdata object as returned by `read_secuTrial_raw`

Value

`secuTrialdata` object containing a list of export options and data.frames with all the data loaded from the secuTrial export. The list will contain at least the metadata data.frames and export_options list.

data.frame with a row for each table in the export. For each table it contains the name, number of rows and columns, an indicator for whether the table is a metadata table and the files original name.

Examples

```
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "BMD",
                               "s_export_CSV-xls_BMD_short_en_utf8.zip",
                               package = "secuTrialR")
# read all export data
sT_export <- read_secuTrial_raw(data_dir = export_location)

# Print method
print(sT_export)
# or
sT_export
```

---

subset_secuTrial *Subsets a* secuTrialdata *object*

Description

Given a secuTrialdata object, and subsetting parameters, this function filters the data object to include only the desired study participants. Subsetting is possible based on participants and based on centres. In order to subset based on participants, participant IDs (mnpaid) musst be present in the export. In order to subset based on centres, centre information must be included in the export.

Usage

```
subset_secuTrial(dat, participant = NULL, centre = NULL, exclude = FALSE)
```

Arguments

dat	secuTrialdata object containing participant IDs and centre information
participant	character vector with a selection of participant IDs (mnpaid) used for subsetting
centre	character vector with a selection of centre names (mnpctrname) used for subsetting
exclude	boolean which if true excludes participants and centres from dat

Details

Subsetting based on participants only, centers only, or based on both is possible. The value of parameter exclude determines whether the output will include participants that meet selection criteria (when exclude = FALSE), or exclude them (when exclude = TRUE). When selecting based on both participants and centres, exclude = FALSE will include the intersection of participants meeting the selection criteria. If exclude = TRUE, a complement of union of participant and centre sets is returned.

Value

secuTrialdata object containing only those participants that meet the selection criteria.

Examples

```
path <- system.file("extdata", "sT_exports", "exp_opt",
                    "s_export_CSV-xls_CTU05_all_info.zip",
                    package = "secuTrialR")
sT <- read_secuTrial(path)
participants <- c("RPACK-INS-011", "RPACK-INS-014", "RPACK-INS-015")
centres <- c("Inselspital Bern (RPACK)", "Universitätsspital Basel (RPACK)")

# show all participants
get_participants(sT)

# subset sT_export
sT_subset1 <- subset_secuTrial(dat = sT, participant = participants)
get_participants(sT_subset1)
sT_subset2 <- subset_secuTrial(dat = sT, participant = participants, exclude = TRUE)
get_participants(sT_subset2)
sT_subset3 <- subset_secuTrial(dat = sT, centre = centres, exclude = TRUE)
get_participants(sT_subset3)
sT_subset4 <- subset_secuTrial(dat = sT, participant = participants,
                               centre = centres, exclude = FALSE)
get_participants(sT_subset4)
sT_subset5 <- subset_secuTrial(dat = sT, participant = participants,
                               centre = centres[2], exclude = FALSE)
get_participants(sT_subset5)
```

---

visit_structure *Get the visit structure of* secuTrialdata *objects*

Description

Get the visit structure of secuTrialdata objects

Usage

```
visit_structure(x, sorted = TRUE)

## S3 method for class 'secuTrialvisit'
plot(x, ...)
```

Arguments

x	a secuTrialdata object
sorted	logical if TRUE sorted by first visit
...	further parameters

Value

data.frame with 1 for whether a form (rows) was collected during a particular visit (columns)

plot of the visit plan

Note

Requires a fixed visit structure - an error will be returned for projects without a visit structure or one with flexible visits

Examples

```
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# read all export data
sT_export <- read_secuTrial(data_dir = export_location)
# get visit structure
vs <- visit_structure(sT_export)
# plot
plot(vs)
```

---

write_secuTrial *Write secuTrial exports to other formats*

Description

Convert the export prepared in R and export it to SPSS (sav), Stata (dta) or SAS (xpt version 8) using the haven package.

Usage

```
write_secuTrial(object, ...)

## S3 method for class 'secuTrialdata'
write_secuTrial(object, format = "dta", metadata = FALSE, ...)

## S3 method for class 'data.frame'
write_secuTrial(object, filename, path = "", format = "dta", ...)
```

Arguments

object	secuTrialdata object
...	further parameters
format	format in which to save the export (one of "dta", "sav", "xpt")
metadata	if TRUE then metadate files will also be written
filename	file name
path	directory where the files should be saved

Details

Due to variable naming limitations in other packages, date variables are appended with _d (rather than _date), datetime/POSIX variables are appended with _dt (rather than _datetime) and factors with _f (rather than _factor). Further variable names may be altered in the conversion process. For details please refer to the haven documentation.

Value

a list of filenames

Examples

```
# prepare path to example export
export_location <- system.file("extdata", "sT_exports", "lnames",
                               "s_export_CSV-xls_CTU05_long_ref_miss_en_utf8.zip",
                               package = "secuTrialR")
# load all export data
sT_export <- read_secuTrial(data_dir = export_location)
tdir <- tempdir()
write_secuTrial(sT_export, format = "dta", path = tdir)
list.files(tdir)
```

Index

∗ **completeness**
    form_status_counts, 13
    form_status_summary, 14
∗ **form**
    form_status_counts, 13
    form_status_summary, 14
∗ **status**
    form_status_counts, 13
    form_status_summary, 14

annual_recruitment, 2
as.data.frame.secuTrialdata, 3
assess_form_variable_completeness, 5

build_secuTrial_url, 6

check_export_options, 8

dates_secuTrial, 9
dictionary_secuTrial, 10
diff_secuTrial, 11

export_date *(*export_datetime*)*, 12
export_datetime, 12

factorize, 12
factorize_secuTrial *(*factorize*)*, 12
form_status_counts, 13
form_status_summary, 14

get_participants, 15

label *(*labels_secuTrial*)*, 15
label<- *(*labels_secuTrial*)*, 15
label_secuTrial *(*labels_secuTrial*)*, 15
labels_secuTrial, 15
links_secuTrial, 17

plot.secuTrialvisit *(*visit_structure*)*, 26
plot_recruitment, 19
print.secuTrialdata *(*secuTrialdata*)*, 24

read_secuTrial, 20
read_secuTrial_raw *(*secuTrialdata*)*, 24
read_validation_overview, 21
return_hidden_items, 21
return_random_participants, 22
return_scores, 23

secuTrialdata, 24
subset_secuTrial, 25

units *(*labels_secuTrial*)*, 15
units<- *(*labels_secuTrial*)*, 15

visit_structure, 26

write_secuTrial, 27