

Package ‘cdashQC’

April 24, 2017

Title Quality control for TFLs

Version 0.1.6

Depends R(>= 3.1.1)

Imports lubridate(>= 1.6.0),
reshape2(>= 1.4.1),
dplyr(>= 0.5.0),
haven(>= 0.2.1),
knitr(>= 1.14)

Author Bin Zhuo

Maintainer Bin Zhuo <bin.zhuo@celerion.com>

Description This package creates table and listings for QC of cdash format data.

License What license is it under?

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

assign_ae_trt	2
clean_lab	3
codes	3
count_percent	4
create_aet	4
create_baseline	5
create_dem	5
create_eg	6
create_included	6
create_lab	7
create_phour	7
create_seq	8
create_vs	8
eg_rm	9
ex_start	9
format_time	10
gcv	10
get_summary_stats	11

guess_base_phour	11
guess_race_age	12
guess_reps	12
guess_test	13
lab_oor	14
listing_ae	14
listing_concom	15
listing_dem	15
normal_range	16
replicate_average	16
replicate_check	17
replicate_clean	17
replicate_data	18
round_df	19
sdtm_pkpd	19
summary_ae_rel	20
summary_ae_sev	20
summary_dem	21
summary_lab	22
summary_labshift	22
summary_pkpd	23
summary_teae	24
summary_vs_eg	24
weight_height_bmi	25
Index	26

assign_ae_trt	<i>Assing AE to treatment</i>
---------------	-------------------------------

Description

Assign ae to a treatment.

Usage

```
assign_ae_trt(ae, ex)
```

Arguments

ae	the dataset ae read from sas
ex	the dataset ex read from sas

Value

a data frame with one more column specifying which treatment the AE should be in.

clean_lab	<i>Clean the lab test data set.</i>
-----------	-------------------------------------

Description

clean the lab data, handling rechecks/unscheduled included.

Usage

```
clean_lab(lab, return_prob = FALSE)
```

Arguments

lab	the dataset returned by create_lab
return_prob	should the problematic observations (e.g, unscheduled, early terminatio, and those supposed to be done predose, but actually done postdose) be shown? FALSE by default.

Value

lab data with necessary variables kept for further summarization.

See Also

[create_lab](#)

codes	<i>the data set codes.</i>
-------	----------------------------

Description

A dataset containing the lab test code and its corresponding full name

Usage

```
codes
```

Format

a data frame containing all possible LB_TESTC and the corresponding full names.

TESTNUM Maybe the number code?

LB_TESTC the code for laboratory test

TTL the full name

COLWIDE The column width? ...

Source

<https://github.com/statswork/cdashQC/tree/master/data-raw/>

count_percent	<i>add percentage to a frequency table</i>
---------------	--

Description

count to percent

Usage

```
count_percent(data, var1, var2, digit_keep = 3)
```

Arguments

data	the data should contain at least 2 columns
var1	which column should be the group
var2	which columns are used to calculate percentage (could be a vector)
digit_keep	how many digits should be kept.

Value

a data frame

Examples

```
trait = rep(c("A", "B", "C"), 3); subtype = paste(trait, rep(1:3, each=3), sep = "")
data <- data.frame(trait, subtype, count1 = rpois(9, 5), count2 = rpois(9, 10))
count_percent(data, var1= 1, var2 = 3:4)

r1 <- get_summary_stats(eg, group = "PERIOD", var = "EG_TEST")
count_percent(r1, var1 = 1, var2 = 3:5)
```

create_aet	<i>do what createaet does.</i>
------------	--------------------------------

Description

createaet.

Usage

```
create_aet(ae, ex)
```

Arguments

ae	the dataset ae read from sas
ex	the dataset ex read from sas

Value

aet the data

create_baseline	<i>create baseline indicator.</i>
-----------------	-----------------------------------

Description

create an extra column indicating whether the current row should be considered as baseline.

Usage

```
create_baseline(data, var_identifier = "_TEST")
```

Arguments

data currently support vs, eg and lb_cq.

var_identifier a string that can be used to identify the variable name e.g.(VS_TEST , EG_TEST , LB_TEST)

Value

a data frame with an extra column status whose value could be one of BASELINE, POSTDOSE and PREDOSE (NOT BASELINE)

See Also

[guess_base_phour](#)

create_dem	<i>create the demographics data.</i>
------------	--------------------------------------

Usage

```
create_dem(dm, vs, included)
```

Arguments

dm the dm data set

vs the vs data set

included the included data set created by create_included

Value

a data frame

See Also

[create_included](#)

create_eg	<i>create variable columns for the ECGs parameters.</i>
-----------	---

Description

dcast the ecg variables

Usage

```
create_eg(eg, included)
```

Arguments

eg	the dataset eg read from sas
included	the included data set created by create_included

Value

egl data with necessary variables kept for further summarization.

create_included	<i>an integrated data set.</i>
-----------------	--------------------------------

Description

create_included.

Usage

```
create_included(ex, cr, ds)
```

Arguments

ex	the dataset ex read from sas
cr	the dataset cr read from sas
ds	the dataset ds read from sas

Value

the included data set which contains the following information

CLIENTID	the patient ID
SCRID	the corresponding screening ID
SEQ	the treatment sequence, from ex data
PERIOD	CRF study period, from ex data
EX_ROUTE	Route
EX_FORM	Formulation
EX_STDAT	Date of first dosing

EX_STTIM	Time of first dosing
EX_DOSE	Dose Amount
EX_LASTDOSEDAT	Date of last dosing (Created in this function)
EX_LASTDOSETIM	Time of last dosing
CHK_IN_DAT	date of check-in for each period
CHK_IN_TIM	time of check in for each period
RETURN_DAT	date of return for each period
RETURN_TIM	time of return for each period
DS_TRM_D	Primary reason for discontinuation
DS_SPEC	specify
DS_STDAT	Date of completion or discontinuation per subject
DS_LSDAT	Date of last contact per subject

create_lab	<i>create the lab test data set.</i>
------------	--------------------------------------

Description

create the lab data

Usage

```
create_lab(lb_cq, included)
```

Arguments

lb_cq	the dataset lb_cq read from sas
included	the included data set created by create_included

Value

lab data with necessary variables kept for further summarization.

create_phour	<i>Create protocol hour</i>
--------------	-----------------------------

Description

Create protocol hour variable if it's not already in the data set.

Usage

```
create_phour(data, digit = 2)
```

Arguments

data	either eg, vs or lb_cq
digit	round the hour by digit, default set to be 2.

Value

the same data set with one more column if Phour is created, otherwise returns the input data.

create_seq	<i>Create SEQ</i>
------------	-------------------

Description

Create SEQ variable (treatment) if it's not already in the data set.

Usage

```
create_seq(data, included)
```

Arguments

data	either eg, vs or lb_cq
included	the included data set created by create_included

Value

the same data set with one more column if SEQ is created, otherwise returns the input data.

create_vs	<i>create the vital sign data set.</i>
-----------	--

Description

create the vs data

Usage

```
create_vs(vs, included)
```

Arguments

vs	the dataset dm read from sas
included	the included data set created by create_included

Value

vs data with necessary variables kept for further summarization.

eg_rm	<i>remove the unwanted ecgs and return the "good" ones</i>
-------	--

Description

clean the dirty ecgs by looking at the comments. This function finds the ecgs to be removed identified by exact event time.

Usage

```
eg_rm(eg_prob, rm_row)
```

Arguments

eg_prob	the ("dirty") data set from replicate_data
rm_row	A vector specifying the rows to be removed.

Value

the pruned ecg data

ex_start	<i>Get the start time of each treatment</i>
----------	---

Description

Get the start time of each treatment

Usage

```
ex_start(ex)
```

Arguments

ex	the dataset ex read from sas
----	------------------------------

Value

a data frame specifying the start time of each treatment

format_time	<i>Format the time variable in a data set</i>
-------------	---

Description

Format the time variable in a data set

Usage

```
format_time(data, date = "AE_STDT", time = "AE_STTM",  
            newname = paste(date, time, sep = "_"))
```

Arguments

data	the data which contains the time variables
date	the date variable
time	the time (in seconds)
newname	what should the new variable that concatnates date and time be named?

Value

the same data set but time has been changed to "hh:mm:ss" format.

gcv	<i>calculate geometric CV</i>
-----	-------------------------------

Description

Calculate geometric CV

Usage

```
gcv(v1, log = FALSE)
```

Arguments

v1	a vector
----	----------

Value

the geometric CV

get_summary_stats	<i>get summary statistics</i>
-------------------	-------------------------------

Usage

```
get_summary_stats(data, group = "EX_TRT_C", var = "race", na.rm = TRUE)
```

Arguments

data	the data
group	which column should be the group
var	which columns are used to summarize
na.rm	should missing value be removed? TRUE by default.

Value

a data frame

Examples

```
SEQ = rep(c("A", "B", "C"), 3); subtype = sample(c("ONE", "TWO", "THREE"), 9, replace = TRUE)
data <- data.frame(SEQ, subtype, BMI = rnorm(9, 25, 4), HEIGHT = rnorm(9, 175, 3))
get_summary_stats(data, group = "SEQ", var = "subtype")
get_summary_stats(data, group = "SEQ", var = "BMI")
```

guess_base_phour	<i>find the baseline hour for each test</i>
------------------	---

Description

find the baseline hour for each test category

Usage

```
guess_base_phour(data, var_identifier = "_TEST")
```

Arguments

data	currently support vs, eg and lb_cq.
var_identifier	a string that can be used to identify the variable name e.g.(VS_TEST, EG_TEST, LB_TEST)

Value

a data frame where by-subject baseline hour is determined.

Examples

```
# the following two will give you exactly the same result
r1 <- guess_base_phour(vs)
r2 <- guess_base_phour(vs, var_identifier = "VS_TEST")

# If you want to find baseline hours for eg, the following two methods are equivalent
r3 <- guess_base_phour(eg)
r4 <- guess_base_phour(eg, var_identifier = "EG_TEST")
```

guess_race_age	<i>race and ethnicity indicator, age.</i>
----------------	---

Description

create necessary variable for the demographic summary tables and listing.

Usage

```
guess_race_age(dm, included)
```

Arguments

dm	the dataset dm read from sas
ex	the dataset ex read from sas

Value

a data frame with additional columns listed as follows

race	the race of the subject
ethnic	has two levels, "NOT HISPANIC OR LATINO" and "HISPANIC OR LATINO".
EX_TRT_C	the treatment groups
ptno	convert CLIENTID to numerical values of subject number
age	Age calculated from start date of treatment

guess_reps	<i>guess the number of replicates</i>
------------	---------------------------------------

Description

guess the number of replicates for each SEQ/PERIOD/PHOUR/TESTCODE

Usage

```
guess_reps(data, var_identifier = "_TEST")
```

Arguments

data data of eg (created by [create_eg](#)) or vs (created by [create_vs](#))
var_identifier,
 which columns is the variable for test code names.

Value

a data frame containing number of reps per PERIOD/PHOUR/TESTCODE

Examples

```
eg0 <- create(eg, included)  
d1 <- guess_reps(eg0)
```

guess_test	<i>guess the name of the test</i>
------------	-----------------------------------

Description

guess the name of the test code: it should end with var_identifier.

Usage

```
guess_test(data, var_identifier = "_TEST")
```

Arguments

data currently support eg and vs
var_identifier,
 which columns is the variable for test code names.

Value

the test code

Examples

```
guess_test(eg)
```

lab_oor	<i>Find OOR</i>
---------	-----------------

Description

Find the out-of-range values

Usage

lab_oor(lab)

Arguments

lab the data returned by [create_lab](#).

Value

a list containg OOR.

listing_ae	<i>list Adverse Envent</i>
------------	----------------------------

Description

list the ae

Usage

listing_ae(aet, type = 1)

Arguments

aet created by create_aet
type an indicator. Should I list ae1 (type = 1), ae2 (type = 2) or ae3 (type = 3)?

Value

a data frame

See Also

[create_aet](#)

listing_concom	<i>list concomitant.</i>
----------------	--------------------------

Description

list concomitant medication data

Usage

```
listing_concom(cm)
```

Arguments

cm the dataset cm read from sas

Value

the data with related columns

listing_dem	<i>demographics listing.</i>
-------------	------------------------------

Description

Summarize the demographic data

Usage

```
listing_dem(dmt)
```

Arguments

dmt the data set created by create_dem.

Value

a data frame

See Also

[create_included](#) and [create_dem](#)

normal_range	<i>Find normal range for lab test</i>
--------------	---------------------------------------

Description

Find the normal range of each test code

Usage

```
normal_range(lab)
```

Arguments

lab	the dataset generated by create_lab
-----	-------------------------------------

Value

a data frame containing test codes and their corresponding normal range

replicate_average	<i>get the averanges of the replicates for vs or eg data.</i>
-------------------	---

Description

Get the averages of the replicates

Usage

```
replicate_average(data_clean, digits = NULL, na_rm = TRUE)
```

Arguments

data_clean	an object returned from replicate_clean
digits	should the averages be rounded? Default NO.
na_rm	should missing values be exclude? Default TRUE.

Value

the averages

See Also

[replicate_clean](#)

Examples

```
eg2 <- replicate_data(eg) # find the triplicates
```

replicate_check	<i>check the replicates and issue a message if there's issue</i>
-----------------	--

Description

check data qualities, reporting missing and RECHECKS

Usage

```
replicate_check(data, reps = NULL, printed = TRUE)
```

Arguments

data	data of eg (created by create_eg) or vs (created by create_vs)
reps	the number of replicates planned. (should be tabled by TESTCODE, SEQ, PERIOD and PHOUR). If NULL, guess_reps will be invoked to guess the number of replicates.
printed	whether to print the observations having issues at the console.

Value

a data frame containing subject and issues at given period and protocol hours

See Also

[guess_reps](#).

Examples

```
eg0 <- create_eg(eg, included)  
d1 <- replicate_check(eg1)
```

replicate_clean	<i>get the cleaned replicates by combining the "clean" data and "dirty" data</i>
-----------------	--

Description

get cleaned replicates

Usage

```
replicate_clean(data, rm_row = NULL)
```

Arguments

data	an object returned from replicate_data (works for vs and eg)
rm_row	a vector of integers specifying which rows should be removed from the dirty data.

Value

the cleaned replicates

See Also

[replicate_data](#)

Examples

```
eg2 <- replicate_data(eg) # step 1: find the triplicates
eg_prob <- eg2$data_dirty # need manual check
# specifies the rows that should be removed
rows_removed <- c(8, 31:33, 67, 68, 95, 119, 144, 145, 176, 200, 221, 245, 272, 296)
eg3 <- replicate_clean(eg2, rows_removed)
```

replicate_data	<i>get the replicates for eg or vs data.</i>
----------------	--

Description

get the replicates for each protocol hour (PHOUR)

Usage

```
replicate_data(data, reps = NULL)
```

Arguments

data	the data set created by create_eg (for ecg data) or by create_vs (for vital signs data).
reps	specifies the structures of reps. If not specified, guess_resp will be invoked. Default set to be NULL.

Value

a list	
data_clean	the subjects containing correct number of resplicates
data_dirty	the subjects that have different number of replicates than desired

See Also

[create_eg](#), [guess_resp](#) and [replicate_check](#)

round_df	<i>round data frame</i>
----------	-------------------------

Description

round a data frame if the column is numerical

Usage

```
round_df(df, digits = 3)
```

Arguments

df	the data frame
digits	how many digits you want to keep

Value

a data frame with numerical columns rounded.

sdtm_pkpd	<i>prepare PK/PD data</i>
-----------	---------------------------

Description

create the pc, pd and pp data

Usage

```
sdtm_pkpd(data, ex)
```

Arguments

data	the SDTM format data of either pc, pd or pp.
ex	the SDTM data ex where treatment information is stored.

Value

the data set with necessary variables selected for analysis

summary_ae_rel	<i>Adverse event by relationship to study drug</i>
----------------	--

Description

summarize adverse events by relationship to study drug

Usage

```
summary_ae_rel(data, group = "STUDYNO", var = "EX_TRT_C", by = "subject")
```

Arguments

data	the data set created by create_aet
group	choose a variable that has all values identical (e.g., "STUDYNO").
var	the treatment variable name, "EX_TRT_C" for example
by	has two options, either "event" or "subject". If by = "subject", that means When a subject experienced the same AE at more than one level of drug relationship during a treatment period, each AE was counted separately

Value

a list	
pt	frequency table by preferred term
total	frequency table by treatment

summary_ae_sev	<i>Adverse event by severity</i>
----------------	----------------------------------

Description

summarize adverse events by severity

Usage

```
summary_ae_sev(data, group = "STUDYNO", var = "EX_TRT_C", by = "subject")
```

Arguments

data	the data set created by create_aet
group	choose a variable that has all values identical (e.g., "STUDYNO").
var	the treatment variable name, "EX_TRT_C" for example
by	has two options, either "event" or "subject". If by = "subject", the subject will be counted under the highest severity level.

Value

- a list
- pt frequency table by preferred term
- total frequency table by treatment

summary_dem	<i>demographic summary.</i>
-------------	-----------------------------

Description

Summarize the demographic data

Usage

```
summary_dem(dmt, group = "SEQ", na.rm = TRUE)
```

Arguments

- dmt the data set created by create_dem.
- group which variable name would be used to calculate summary statistics?
- na.rm should missing values be included? TRUE by default.

Value

a data frame

See Also

[create_included](#) and [create_dem](#)

Examples

```
included <- create_included(ex, dm, cr, ds)
dmt <- create_dem(dm, ex, vs, included)
summary_dem(dmt, group = "SEQ")    # the summary by group
summary_dem(dmt, group = "SPONSOR") # to get the overall summary
```

summary_lab	<i>summary statistics for lab shift.</i>
-------------	--

Description

Summarize the lab statistics

Usage

```
summary_lab(lb_clean, digits = 3)
```

Arguments

lb_clean	the dataset returned by clean_lab.
digits	how many digits should be kept for the final data.

Value

the summary statistics by test code, time point and by treatment.

See Also

[clean_lab](#)

Examples

```
included <- create_included(ex, dm, cr, ds)
lab <- create_lab(lb_cq)
lb_clean <- clean_lab(lab, included= included, ex= ex)
s1 <- summary_lab(clean_lab) #not urinalysis
```

summary_labshift	<i>create the lab shift table.</i>
------------------	------------------------------------

Description

Create the lab shift table.

Usage

```
summary_labshift(lb_clean)
```

Arguments

lb_clean	the dataset returned by clean_lab.
----------	------------------------------------

Value

the shift table

See Also

[clean_lab](#)

Examples

```
included <- create_included(ex, dm, cr, ds)
lab <- create_lab(lb_cq)
lb_clean <- clean_lab(lab, included, ex)
lbshift <- summary_labshift(lb_clean)
```

summary_pkpd	<i>calculate the 10-point statistics for PK/PD</i>
--------------	--

Description

Summary statistics for the PK/PD data. Celerion standard: When calculating Geom Mean and GCV, SAS code takes the concentration values of 0.00 that would go into the calculation and sets them to missing and then calculates the geom mean and gcv with those values removed. Those 0.00 values are used when calculating all other summary statistics.

Usage

```
summary_pkpd(data, na_rm = TRUE)
```

Arguments

data	the object returned from sdtm_pkpd
na_rm	should NA be removed? Default TRUE

Value

the statistics	
n	number of observations
mean	the mean
sd	standard deviation
cv	coefficient of variation
sem	standard error of the mean
min	minimum
median	the median
max	the maximum
geom	geometric mean
gcv	geometric standard deviation $GCV = \sqrt{\exp(\text{var}(\log(x))) - 1}$

summary_teae	<i>teae by number of subjects or by number of events</i>
--------------	--

Description

summarize treatment-emergent adverse event

Usage

```
summary_teae(data, group = "STUDYNO", var = "EX_TRT_C", by = "subject")
```

Arguments

data	the data set created by create_aet
group	choose a variable that has all values identical (e.g., "STUDYNO").
var	the treatment variable name, "EX_TRT_C" for example
by	has two options, either "event" or "subject"

Value

a list	
ae	frequency table for teae
soc	summarization by system organ class and by preferred term

summary_vs_eg	<i>summary statistics</i>
---------------	---------------------------

Description

Summary statistics for vital signs or eg

Usage

```
summary_vs_eg(data_clean, inter_digit = NULL, final_digits = 3,  
  na_rm = TRUE, ischangefrombase = FALSE)
```

Arguments

data_clean	the clean data set returned by replicate_clean .
inter_digit	if rounding happens for the intermediate averages, what digits should be kept.
na_rm	should missing values be excluded? Default set to be TRUE.
ischangefrombase	Is this summary for change from baseline? Default set to be FALSE
final_digit	what is the digit for final summary?

Value

a data frame with summary statistics by test, time point and treatment.

weight_height_bmi	<i>read BMI Weight and Height info from Screening stage.</i>
-------------------	--

Description

extract BMI, weight and height from vs.sas7bat.

Usage

```
weight_height_bmi(vs, period = "SCREEN")
```

Arguments

vs the vs sas data

Value

a data frame containing BMI, WEIGHT and HEIGHT from admission stage

Index

*Topic **datasets**

codes, [3](#)

assign_ae_trt, [2](#)

clean_lab, [3](#), [22](#), [23](#)

codes, [3](#)

count_percent, [4](#)

create_aet, [4](#), [14](#), [20](#), [24](#)

create_baseline, [5](#)

create_dem, [5](#), [15](#), [21](#)

create_eg, [6](#), [13](#), [17](#), [18](#)

create_included, [5](#), [6](#), [6](#), [7](#), [8](#), [15](#), [21](#)

create_lab, [3](#), [7](#), [14](#)

create_phour, [7](#)

create_seq, [8](#)

create_vs, [8](#), [13](#), [17](#), [18](#)

eg_rm, [9](#)

ex_start, [9](#)

format_time, [10](#)

gcv, [10](#)

get_summary_stats, [11](#)

guess_base_phour, [5](#), [11](#)

guess_race_age, [12](#)

guess_reps, [12](#)

guess_resp, [18](#)

guess_test, [13](#)

lab_oor, [14](#)

listing_ae, [14](#)

listing_concom, [15](#)

listing_dem, [15](#)

normal_range, [16](#)

replicate_average, [16](#)

replicate_check, [17](#), [18](#)

replicate_clean, [16](#), [17](#), [24](#)

replicate_data, [9](#), [18](#), [18](#)

round_df, [19](#)

sdtm_pkpd, [19](#), [23](#)

summary_ae_rel, [20](#)

summary_ae_sev, [20](#)

summary_dem, [21](#)

summary_lab, [22](#)

summary_labshift, [22](#)

summary_pkpd, [23](#)

summary_teae, [24](#)

summary_vs_eg, [24](#)

weight_height_bmi, [25](#)