

# Package ‘dmtools’

May 12, 2020

**Title** Tools for Validation the Dataset

**Version** 0.2.4

**Description** For checking the dataset from EDC(Electronic Data Capture) in clinical trials.  
'dmtools' can check laboratory, dates, WBCs(White Blood Cells) count and rename the dataset.  
Laboratory - does the investigator correctly estimate the laboratory analyzes?  
Dates - do all dates correspond to the protocol's timeline?  
WBCs count - do absolute equal (all \* relative) / 100?  
If the clinical trial has different lab reference ranges, 'dmtools' also can help.

**Depends** R (>= 3.6)

**Imports** magrittr (>= 1.5), dplyr (>= 0.8.3), readxl (>= 1.3.1), purrr (>= 0.3.3), lubridate (>= 1.7.4)

**License** MIT + file LICENSE

**URL** <https://github.com/chachaboos/dmtools>

**BugReports** <https://github.com/chachaboos/dmtools/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Konstantin Ryabov [aut, cre]

**Maintainer** Konstantin Ryabov <chachaboos@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-05-12 11:00:06 UTC

## R topics documented:

calc_diff . . . . .	2
check . . . . .	3

check.default	3
check_sites	4
choose_test	5
choose_test.date	6
choose_test.lab	7
choose_test.wbc	8
create_norm	9
create_spec	10
date	10
dmtools	11
find_colnames	11
find_colnames.date	12
find_colnames.default	12
get_result	13
get_result.default	14
lab	15
rename_dataset	16
run_tests	17
run_tests.date	17
run_tests.lab	18
run_tests.short	19
run_tests.wbc	19
short	20
test_sites	20
to_dbl	21
wbc	22

## Index 23

---

calc_diff	<i>Function for calculating the difference between two dates</i>
-----------	--

---

### Description

Function for calculating the difference between two dates

### Usage

```
calc_diff(st_inter, dt_item)
```

### Arguments

st_inter	An interval. An Object of interval.
dt_item	A date item. An Object of date.

### Value

An integer scalar. Differences between the two dates.

---

check	<i>Check</i>
-------	--------------

---

**Description**

Check

**Usage**

```
check(obj, dataset)
```

**Arguments**

obj	An object for check.
dataset	A dataset, a type is a data frame.

**Value**

An object with a check result.

**Examples**

```
id <- c("01", "02", "03")
screen_date_E1 <- c("1991-03-13", "1991-03-07", "1991-03-08")
rand_date_E2 <- c("1991-03-15", "1991-03-11", "1991-03-10")
ph_date_E3 <- c("1991-03-21", "1991-03-16", "1991-03-16")
bio_date_E3 <- c("1991-03-23", "1991-03-16", "1991-03-16")

df <- data.frame(id, screen_date_E1, rand_date_E2, ph_date_E3, bio_date_E3,
stringsAsFactors = FALSE)

timeline <- system.file("dates.xlsx", package = "dmtools")
obj_date <- date(timeline, id, dplyr::contains)

obj_date <- check(obj_date, df)
```

---

check.default	<i>Check</i>
---------------	--------------

---

**Description**

Check

**Usage**

```
## Default S3 method:
check(obj, dataset)
```

**Arguments**

obj                    An object for check.  
 dataset                A dataset, a type is a data frame.

**Value**

An object with a check result.

**Examples**

```
id <- c("01", "02", "03")
screen_date_E1 <- c("1991-03-13", "1991-03-07", "1991-03-08")
rand_date_E2 <- c("1991-03-15", "1991-03-11", "1991-03-10")
ph_date_E3 <- c("1991-03-21", "1991-03-16", "1991-03-16")
bio_date_E3 <- c("1991-03-23", "1991-03-16", "1991-03-16")

df <- data.frame(id, screen_date_E1, rand_date_E2, ph_date_E3, bio_date_E3,
stringsAsFactors = FALSE)

timeline <- system.file("dates.xlsx", package = "dmtools")
obj_date <- date(timeline, id, dplyr::contains)

obj_date <- check(obj_date, df)
```

---

check\_sites

*Check sites*

---

**Description**

Check sites

**Usage**

```
check_sites(objs, dataset, col_site)
```

**Arguments**

objs                    A list of objects.  
 dataset                A dataset, a type is a data frame.  
 col\_site                A column name of a site in the dataset, without quotes.

**Value**

A list of objects with a check result.

**Examples**

```

site <- c("site 01", "site 02")
id <- c("01", "02")
age <- c("19", "20")
sex <- c("f", "m")
gluc_post <- c("5.5", "4.1")
gluc_res_post <- c("norm", "no")
ast_post <- c("30", "48")
ast_res_post <- c(NA, "norm")

df <- data.frame(
  site, id, age, sex,
  gluc_post, gluc_res_post,
  ast_post, ast_res_post,
  stringsAsFactors = FALSE
)

refs_s01 <- system.file("labs_refer_s01.xlsx", package = "dmtools")
refs_s02 <- system.file("labs_refer_s02.xlsx", package = "dmtools")

s01_lab <- lab(refs_s01, id, age, sex, "norm", "no", site = "site 01")
s02_lab <- lab(refs_s02, id, age, sex, "norm", "no", site = "site 02")

labs <- list(s01_lab, s02_lab)
labs <- check_sites(labs, df, site)

```

---

choose_test	<i>Filter final result</i>
-------------	----------------------------

---

**Description**

Filter final result

**Usage**

```
choose_test(obj, test, group_id)
```

**Arguments**

obj	An object for check.
test	Parameters, which use to filter the final dataset.
group_id	A logical scalar, default is TRUE. True is grouped by id, otherwise, it isn't grouped.

**Value**

The filtered dataset.

**Examples**

```

id <- c("01", "02", "03")
screen_date_E1 <- c("1991-03-13", "1991-03-07", "1991-03-08")
rand_date_E2 <- c("1991-03-15", "1991-03-11", "1991-03-10")
ph_date_E3 <- c("1991-03-21", "1991-03-16", "1991-03-16")
bio_date_E3 <- c("1991-03-23", "1991-03-16", "1991-03-16")

df <- data.frame(id, screen_date_E1, rand_date_E2, ph_date_E3, bio_date_E3,
stringsAsFactors = FALSE)

timeline <- system.file("dates.xlsx", package = "dmtools")
obj_date <- date(timeline, id, dplyr::contains)

obj_date <- check(obj_date, df)
choose_test(obj_date, "out")

```

---

choose_test.date	<i>Filter final result</i>
------------------	----------------------------

---

**Description**

Filter final result

**Usage**

```

## S3 method for class 'date'
choose_test(obj, test = "out", group_id = T)

```

**Arguments**

obj	An object for calculation. Class date.
test	A character scalar. Parameters, which use to filter the final dataset, default: "out": "out" - dates, which are out of the protocol's timeline, "uneq" - dates, which are unequal, "ok" - correct dates, "skip" - empty dates.
group_id	A logical scalar, default is TRUE. True is grouped by id, otherwise, it isn't grouped.

**Value**

The dataset by a value of test.

**Examples**

```

id <- c("01", "02", "03")
screen_date_E1 <- c("1991-03-13", "1991-03-07", "1991-03-08")
rand_date_E2 <- c("1991-03-15", "1991-03-11", "1991-03-10")
ph_date_E3 <- c("1991-03-21", "1991-03-16", "1991-03-16")
bio_date_E3 <- c("1991-03-23", "1991-03-16", "1991-03-16")

df <- data.frame(id, screen_date_E1, rand_date_E2, ph_date_E3, bio_date_E3,
stringsAsFactors = FALSE)

timeline <- system.file("dates.xlsx", package = "dmtools")
obj_date <- date(timeline, id, dplyr::contains)

obj_date <- check(obj_date, df)
choose_test(obj_date, "out")

```

---

choose_test.lab	<i>Filter final result</i>
-----------------	----------------------------

---

**Description**

Filter final result

**Usage**

```

## S3 method for class 'lab'
choose_test(obj, test = "mis", group_id = T)

```

**Arguments**

obj	An object. Class lab.
test	A character scalar. Parameters, which use to filter the final dataset, default: "mis": "ok" - analysis, which has a correct estimate of the result, "mis" - analysis, which has an incorrect estimate of the result, "skip" - analysis, which has an empty value of the estimate, "null" - analysis, which has an empty result and value of the estimate.
group_id	A logical scalar, default is TRUE. True is grouped by id, otherwise, it isn't grouped.

**Value**

The filtered dataset by a value of test.

**Examples**

```

id <- c("01", "02", "03")
site <- c("site 01", "site 02", "site 03")
age <- c("19", "20", "22")
sex <- c("f", "m", "f")
gluc_post <- c(5.5, 4.1, 9.7)
gluc_res_post <- c("norm", "no", "c1")
ast_post <- c("30", "48", "31")
ast_res_post <- c(NA, "norm", "norm")

df <- data.frame(
  id, site, age, sex,
  gluc_post, gluc_res_post,
  ast_post, ast_res_post,
  stringsAsFactors = FALSE )

refs <- system.file("labs_refer.xlsx", package = "dmtools")
obj_lab <- lab(refs, id, age, sex, "norm", "no")

obj_lab <- check(obj_lab, df)
choose_test(obj_lab, "mis")

```

---

choose_test.wbc	<i>Filter final result</i>
-----------------	----------------------------

---

**Description**

Filter final result

**Usage**

```

## S3 method for class 'wbc'
choose_test(obj, test = "mis", group_id = T)

```

**Arguments**

obj	An object. Class wbc.
test	A character scalar. Parameters, which use to filter the final dataset, default: "mis": "ok" - wbc, which is calculated correct, "mis" - wbc, which is calculated incorrect.
group_id	A logical scalar, default is TRUE. True is grouped by id, otherwise, it isn't grouped.

**Value**

The dataset by a value of test.



**Examples**

```

id <- c("01", "02", "03")
wbc_post <- c(5.6, 7.8, 8.1)
lym_rel_post <- c(21, 25, 30)
lym_abs_post <- c(1.18, 1.95, 2.13)

df <- data.frame(
  id, wbc_post, lym_rel_post, lym_abs_post,
  stringsAsFactors = FALSE
)

wbcc <- system.file("wbcc.xlsx", package = "dmtools")
obj_wbc <- wbc(wbcc, id)

obj_wbc <- check(obj_wbc, df)
choose_test(obj_wbc, "mis")

```

---

create\_norm

*Estimating laboratory values*


---

**Description**

Estimating laboratory values

**Usage**

```
create_norm(vals, left_bound, right_bound, ds_norm, normal, abnormal, clsign)
```

**Arguments**

vals	A double vector. The laboratory values.
left_bound	A double scalar. The minimum.
right_bound	A double scalar. The maximum.
ds_norm	An estimate of the laboratory values from the dataset.
normal	An option for the normal estimate, for example, "NORMAL".
abnormal	An option for the abnormal estimate, for example, "ABNORMAL".
clsign	An option for the clinical significant estimate, for example, "CLISIG".

**Value**

A vector with the auto estimate.

---

create_spec	<i>For creating part of the specification</i>
-------------	---

---

**Description**

For creating part of the specification

**Usage**

```
create_spec(df_spec, all_colname, part_spec, is_pst)
```

**Arguments**

df_spec	A dataset, a type is a data frame.
all_colname	A character vector with all names in the dataset.
part_spec	A character scalar. Prefixes or postfixes.
is_pst	A logical scalar, default is TRUE. True is postfix, otherwise, prefix.

**Value**

A data frame. Part of the specification.

---

date	<i>Create object date</i>
------	---------------------------

---

**Description**

Create object date

**Usage**

```
date(file, id, get_visit, get_date = dplyr::contains, str_date = "DAT")
```

**Arguments**

file	A character scalar. Path to the date's parameters in the excel table.
id	A column name of the subject id in the dataset, without quotes.
get_visit	A function, which select necessary visit or event e.g. dplyr::start_with, dplyr::contains.
get_date	A function, which select dates from necessary visit e.g. dplyr::matches, dplyr::contains, default: dplyr::contains.
str_date	A date's pattern in column names, default: "DAT".

**Value**

The object date.

**Examples**

```
obj_date <- date("dates.xlsx", id, dplyr::contains)
obj_date <- date("dates.xlsx", id, dplyr::contains, "uneq")
```

---

dmtools

*dmtools: package to validate data*

---

**Description**

for checking the dataset from EDC in clinical trials

---

find\_colnames

*Find column names*

---

**Description**

Find column names

**Usage**

```
find_colnames(obj, dataset, row_file)
```

**Arguments**

obj	An object for check.
dataset	A dataset, a type is a data frame.
row_file	A row of the file.

**Value**

A data frame. Result of run\_tests.

---

find\_colnames.date      *Find column names with dates*

---

**Description**

Find column names with dates

**Usage**

```
## S3 method for class 'date'
find_colnames(obj, dataset, row_file)
```

**Arguments**

obj	An object for validation.
dataset	A data frame. Class date.
row_file	A data frame. A data frame with analysis parameters.

**Value**

A data frame. Visit's dates.

---

find\_colnames.default      *Find column names*

---

**Description**

Find column names

**Usage**

```
## Default S3 method:
find_colnames(obj, dataset, row_file)
```

**Arguments**

obj	An object for validation.
dataset	A dataset, a type is a data frame.
row_file	A row of the file.

**Value**

A data frame. Result of run\_tests.

---

get_result	<i>Get the final result</i>
------------	-----------------------------

---

**Description**

Get the final result

**Usage**

```
get_result(obj, group_id)
```

**Arguments**

obj	An object. Can be all classes: short, lab, wbc, date.
group_id	A logical scalar, default is TRUE. True is grouped by id, otherwise, it isn't grouped.

**Value**

A data frame. The final result.

**Examples**

```
id <- c("01", "02", "03")
site <- c("site 01", "site 02", "site 03")
sex <- c("f", "m", "f")
preg_yn_e2 <- c("y", "y", "y")
preg_res_e2 <- c("neg", "neg", "neg")
preg_yn_e3 <- c("y", "y", "n")
preg_res_e3 <- c("neg", "pos", "unnes")

df <- data.frame(
  id, site, sex,
  preg_yn_e2, preg_res_e2,
  preg_yn_e3, preg_res_e3,
  stringsAsFactors = FALSE )

preg <- system.file("preg.xlsx", package = "dmtools")
obj_short <- short(preg, id, "res", c("site", "sex"))

obj_short <- check(obj_short, df)
get_result(obj_short)
```

---

get\_result.default      *Get final result*

---

## Description

Get final result

## Usage

```
## Default S3 method:  
get_result(obj, group_id = T)
```

## Arguments

`obj`                    An object. Can be all classes: short, lab, wbc, date.  
`group_id`              A logical scalar, default is TRUE. True is grouped by id, otherwise, it isn't grouped.

## Value

A data frame. The final result.

## Examples

```
id <- c("01", "02", "03")  
site <- c("site 01", "site 02", "site 03")  
sex <- c("f", "m", "f")  
preg_yn_e2 <- c("y", "y", "y")  
preg_res_e2 <- c("neg", "neg", "neg")  
preg_yn_e3 <- c("y", "y", "n")  
preg_res_e3 <- c("neg", "pos", "unnes")  
  
df <- data.frame(  
  id, site, sex,  
  preg_yn_e2, preg_res_e2,  
  preg_yn_e3, preg_res_e3,  
  stringsAsFactors = FALSE )  
  
preg <- system.file("preg.xlsx", package = "dmttools")  
obj_short <- short(preg, id, "res", c("site", "sex"))  
  
obj_short <- check(obj_short, df)  
get_result(obj_short)
```

---

lab	<i>Create object lab</i>
-----	--------------------------

---

**Description**

Create object lab

**Usage**

```
lab(  
  file,  
  id,  
  age,  
  sex,  
  norm,  
  no_norm,  
  is_post = T,  
  cl_sign = NA,  
  site = NA,  
  name_to_find = "name_is_norm"  
)
```

**Arguments**

file	A character scalar. Path to the laboratory's reference in the excel table.
id	A column name of the subject id in the dataset, without quotes.
age	A column name of the subject age in the dataset, without quotes.
sex	A column name of the subject sex in the dataset, without quotes.
norm	A normal estimate, for example, "NORMAL".
no_norm	An abnormal estimate, for example, "ABNORMAL".
is_post	A logical scalar, default is TRUE. True is postfix, otherwise, prefix.
cl_sign	A clinical significant estimate, for example, "CLISIG", default: NA.
site	A site number, default: NA.
name_to_find	A character scalar. For search prefixes or postfixes, default is "name_is_norm".

**Value**

The object lab.

**Examples**

```
obj_lab <- lab("lab_refer.xlsx", id, age, sex, 1, 2)  
obj_lab <- lab("lab_refer.xlsx", id, age, sex, "NORMAL", "NOCLISIG", cl_sign = "CLISIG")  
obj_lab <- lab("lab_refer.xlsx", id, age, sex, "norm", "no", FALSE)
```

---

rename_dataset	<i>For rename dataset</i>
----------------	---------------------------

---

### Description

For rename dataset

### Usage

```
rename_dataset(
  dataset,
  path_crfs,
  no_readable_name,
  readable_name,
  num_sheet = 1,
  is_post = T
)
```

### Arguments

dataset	A dataset, a type is a data frame.
path_crfs	A character scalar. Path to the specification files the in excel table.
no_readable_name	A character scalar. A column name of no_readable values.
readable_name	A character scalar. A column name of readable values.
num_sheet	An integer scalar, default is the first sheet. A position of a sheet in the excel document.
is_post	A logical scalar, default is TRUE. True is postfix, otherwise, prefix.

### Value

The list with two values: data - renamed dataset, spec - common specification. The common specification is data frame of two values: no\_readable\_var, readable\_var.

### Examples

```
id <- c("01", "02", "03")
age <- c("19", "20", "22")
sex <- c("f", "m", "f")
bio_date_post <- c("1991-03-23", "1991-03-16", "1991-03-16")
gluc_post <- c("5.5", "4.1", "9.7")
gluc_res_post <- c("norm", "no", "norm")

df <- data.frame(
  id, age, sex,
  bio_date_post,
```



```

    gluc_post, gluc_res_post,
    stringsAsFactors = FALSE
  )

  crfs <- system.file("forms", package = "dmtools")

  result <- rename_dataset(df, crfs, "old_name", "new_name")
  result[["data"]]

```

---

run\_tests

*Run tests*


---

### Description

Run tests

### Usage

```
run_tests(obj, dataset, row_file, part)
```

### Arguments

obj	An object for check.
dataset	A data frame.
row_file	A data frame. A data frame with parameters.
part	A character scalar. Prefixes or postfixes.

### Value

A data frame. The part of final result.

---

run\_tests.date

*Run test*


---

### Description

Run test

### Usage

```

## S3 method for class 'date'
run_tests(obj, dataset, row_file, date)

```

**Arguments**

obj	An object for validation.
dataset	A data frame. Class date.
row_file	A data frame. A data frame with analysis parameters.
date	A column name with dates.

**Value**

A data frame. Result of the date's validation.

---

run_tests.lab	<i>Run tests</i>
---------------	------------------

---

**Description**

Run tests

**Usage**

```
## S3 method for class 'lab'  
run_tests(obj, dataset, row_file, part)
```

**Arguments**

obj	An object. Class lab.
dataset	A data frame.
row_file	A data frame. A data frame with parameters.
part	A character scalar. Prefixes or postfixes.

**Value**

A data frame. The part of the final result.

---

run_tests.short	<i>Run tests</i>
-----------------	------------------

---

**Description**

Run tests

**Usage**

```
## S3 method for class 'short'  
run_tests(obj, dataset, row_file, part)
```

**Arguments**

obj	An object. Class short.
dataset	A data frame.
row_file	A data frame. A data frame with parameters.
part	A character scalar. Prefixes or postfixes.

**Value**

A data frame. The part of the final result.

---

run_tests.wbc	<i>Run tests</i>
---------------	------------------

---

**Description**

Run tests

**Usage**

```
## S3 method for class 'wbc'  
run_tests(obj, dataset, row_file, part)
```

**Arguments**

obj	An object. Class wbc.
dataset	A data frame.
row_file	A data frame. A data frame with parameters.
part	A character scalar. Prefixes or postfixes.

**Value**

A data frame. The part of the final result.

---

short	<i>Create object short</i>
-------	----------------------------

---

**Description**

Create object short

**Usage**

```
short(file, id, name_to_find, common_cols = c(), extra = NA, is_post = T)
```

**Arguments**

file	A character scalar. Path to the excel table.
id	A column name of the subject id in the dataset, without quotes.
name_to_find	A character scalar. For search prefixes or postfixes.
common_cols	A character vector. A column names in the dataset, which common for all events.
extra	A character scalar. For additional information.
is_post	A logical scalar, default is TRUE. True is postfix, otherwise, prefix.

**Value**

The object short.

**Examples**

```
obj_short <- short("preg.xlsx", id,"res", c("site", "sex"))
obj_short <- short("labs.xlsx", id,"name_labs", c("site"), "human_name")
```

---

test_sites	<i>Test sites</i>
------------	-------------------

---

**Description**

Test sites

**Usage**

```
test_sites(objs, func)
```

**Arguments**

objs	A list of objects.
func	A function e.g. choose_test, get_result.

**Value**

A data frame. The dataset.

**Examples**

```
site <- c("site 01", "site 02")
id <- c("01", "02")
age <- c("19", "20")
sex <- c("f", "m")
gluc_post <- c("5.5", "4.1")
gluc_res_post <- c("norm", "no")
ast_post <- c("30", "48")
ast_res_post <- c(NA, "norm")

df <- data.frame(
  site, id, age, sex,
  gluc_post, gluc_res_post,
  ast_post, ast_res_post,
  stringsAsFactors = FALSE
)

refs_s01 <- system.file("labs_refer_s01.xlsx", package = "dmtools")
refs_s02 <- system.file("labs_refer_s02.xlsx", package = "dmtools")

s01_lab <- lab(refs_s01, id, age, sex, "norm", "no", site = "site 01")
s02_lab <- lab(refs_s02, id, age, sex, "norm", "no", site = "site 02")

labs <- list(s01_lab, s02_lab)
labs <- check_sites(labs, df, site)

test_sites(labs, func = function(lab) choose_test(lab, "mis"))
```

---

to\_dbl

*Cast to double type*

---

**Description**

Cast to double type

**Usage**

```
to_dbl(vals)
```

**Arguments**

vals                    A character or double vector.

**Value**

A double vector.

---

wbc	<i>Create object wbc</i>
-----	--------------------------

---

**Description**

Create object wbc

**Usage**

```
wbc(file, id, is_post = T, site = NA, name_to_find = "relative")
```

**Arguments**

file	A character scalar. Path to the laboratory's reference in the excel table.
id	A column name of the subject id in the dataset, without quotes.
is_post	A logical scalar, default is TRUE. True is postfix, otherwise, prefix.
site	A site number, default: NA.
name_to_find	A character scalar. For search prefixes or postfixes, default is "relative".

**Value**

The object wbc.

# Index

calc\_diff, 2  
check, 3  
check.default, 3  
check\_sites, 4  
choose\_test, 5  
choose\_test.date, 6  
choose\_test.lab, 7  
choose\_test.wbc, 8  
create\_norm, 9  
create\_spec, 10

date, 10  
dmtools, 11

find\_colnames, 11  
find\_colnames.date, 12  
find\_colnames.default, 12

get\_result, 13  
get\_result.default, 14

lab, 15

rename\_dataset, 16  
run\_tests, 17  
run\_tests.date, 17  
run\_tests.lab, 18  
run\_tests.short, 19  
run\_tests.wbc, 19

short, 20

test\_sites, 20  
to\_dbl, 21

wbc, 22