Package ‘researchcyclematrix’

December 10, 2019

Type Package

Title What the Package Does (Title Case)

Version 0.2.3

Author Martin Barner

Maintainer Chiara Debenedetti `<yourself@somewhere.net>`

Description Internal HQ data unit processes for tracking item validations

License What license is it under?

Encoding UTF-8

LazyData true

Suggests testthat,
xlsx

Depends knitr,
ggplot2,
googlesheets,
httr,
magrittr,
crayon,
utils,
tidyr,
tibble,
dplyr,
ggthemes

RoxygenNote 6.1.1

Imports lubridate,
assertthat,
jsonlite,
rmarkdown,
purrr,
zip,
stringr

VignetteBuilder knitr

R topics documented:

`country_status_markdown` *this is the backend behind rcm_dashboard; it prepares some data and adds them to an environment in which then ./inst/country_status.Rmd is executed*

Description

this is the backend behind rcm_dashboard; it prepares some data and adds them to an environment in which then ./inst/country_status.Rmd is executed

Usage

```
country_status_markdown(rcm, delayed, inconsistencies, path = "./",
  filename = "country_overview.html")
```

Arguments

`rcm`	the researchcyclematrix; should contain validated items too
`delayed`	output from researchcyclematrix::todo_delayed()
`inconsistencies`	output from rcm_check()
`path`	the path to save the html file to
`filename`	the name of the output html file (must end in '.html')

`date_arrived` *is a date today or older?*

Description

is a date today or older?

Usage

```
date_arrived(date)
```

Arguments

`date`	date(s) to check

Value

logical vector if date is today or in the passed

`filter_validated_in_month` *filter rows from rcm that have state 'validated' and date in a certain month / year*

Description

filter rows from rcm that have state 'validated' and date in a certain month / year

Usage

```
filter_validated_in_month(rcm, month, year = 2019)
```

Arguments

`rcm`	the RCM
`month`	the month to keep (single integer)
`year`	the year to keep (single integer; defaults to 2019)

`gdrive_hyperlink_row` *get link to google drive row for a file.id*

Description

get link to google drive row for a file.id

Usage

```
gdrive_hyperlink_row(file.id)
```

`get_gdrive_fileType` *which row on google drive matches a file.id?*

Description

which row on google drive matches a file.id?

Usage

```
get_gdrive_fileType(file.id, rcm = NULL)
```

Arguments

`file.id`	the file id name of the item
`rcm`	you can provide a custom rcm to use for row identification. Not recommended (except for debugging purposes) this will return the wrong row if the this is not a raw rcm (from rcm_download(raw=T)) or if the RCM online changed since it was downloaded

Value

(estimated) google drive row index of file.id

`get_gdrive_relatedFiles` *which row on google drive matches a file.id?*

Description

which row on google drive matches a file.id?

Usage

```
get_gdrive_relatedFiles(file.id, rcm = NULL)
```

Arguments

`file.id`	the file id name of the item
`rcm`	you can provide a custom rcm to use for row identification. Not recommended (except for debugging purposes) this will return the wrong row if the this is not a raw rcm (from rcm_download(raw=T)) or if the RCM online changed since it was downloaded

Value

(estimated) google drive row index of file.id

get_gdrive_row *which row on google drive matches a file.id?*

Description

which row on google drive matches a file.id?

Usage

```
get_gdrive_row(file.id, rcm = NULL)
```

Arguments

file.id	the file id name of the item
rcm	you can provide a custom rcm to use for row identification. Not recommended (except for debugging purposes) this will return the wrong row if the this is not a raw rcm (from rcm_download(raw=T)) or if the RCM online changed since it was downloaded

Value

(estimated) google drive row index of file.id

g_sheets_append_row *wrapper for google sheets api to append a new row*

Description

wrapper for google sheets api to append a new row

Usage

```
g_sheets_append_row(value,
  spreadsheetId = "1iNt__-uMMBTbLEsJkiIXglPJ4GK-9UCVqC7awhMTXF8", ...)
```

Arguments

value	a character vector of values to put in the row

g_sheets_put — *generally we're using googlesheets package for authentification, but then httr package to call the google drive directly. the workhorse is g_sheets_put which is used in most other functions that edit the remote RCM.*

Description

wrapper for google sheets api to change single value

Usage

```
g_sheets_put(row, col, value,
  spreadsheetId = "1Quu2P6z-uA2H64eQENWJkNIOGIfnsdXgKNg4qdiCvXc", ...)
```

Arguments

`row`	the number of the row we want to fill
`col`	the column to fill as a letter character, excel style: ("A", "B", ... , "Z", "BA", ..., "ZZ")
`value`	the value to put into the cell
`spreadsheetId`	we could use this for other spreadsheets, but it's not working at the moment. Throwing an error if this param is changed

g_sheets_update_index — *fills a column on google drive with incremental indices to ensure row matching works as expected*

Description

fills a column on google drive with incremental indices to ensure row matching works as expected

Usage

```
g_sheets_update_index(col = "AR",
  spreadsheetId = "1wX5k3cETrCbnw4vpfY07eSzTyWX6AwmJmxJQwPahrSk", ...)
```

hello — *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

`hq_focal_point` *who is the focal point for the rcid?*

Description

who is the focal point for the rcid?

Usage

```
hq_focal_point(rcid)
```

Arguments

`rcid`	a vector of research cycle ids

Value

a vector of surnames of the HQ data unit focal point for the research cycle

`is_date_format_in_rcm_style` *check if a date string is in the format that matches the research cycle matrix standard for dates*

Description

check if a date string is in the format that matches the research cycle matrix standard for dates

Usage

```
is_date_format_in_rcm_style(date_string)
```

`rcm_add_validation_button` *add validation button html to rcm*

Description

add validation button html to rcm

Usage

```
rcm_add_validation_button(rcm)
```

Value

rcm with button link

`rcm_allocation_table` *create allocation table*

Description

create allocation table

Usage

```
rcm_allocation_table(who = ".", rcm = NULL, rcm_raw = NULL,
  include.file.ids = c(), month, year = 2019)
```

Arguments

`who`	regex pattern matching HQ Focal Point (defaults to "." which matches all)
`rcm`	the research cycle matrix. leave empty to download fresh
`rcm_raw`	the raw research cycle matrix; get with rcm_download(raw = T) or levae empty to download fresh. Need this because "hours" worked column is not included in the cleaned up rcm download (maybe we should include it).
`include.file.ids`	vector of file ids that should be included even if they do not match the focal point pattern in "who" parameter
`month`	integer of the month for which to create the allocation table
`year`	integer of the year for which to create the allocation table (defaults to 2019; that default will likely change, so even if it's 2019, you should provide it in case that stable future behaviour is needed)

`rcm_browse` *open research cycle matrix google sheet in browser*

Description

open research cycle matrix google sheet in browser

Usage

```
rcm_browse()
```

`rcm_change_value` *change a value on google drive based on the file.id*

Description

change a value on google drive based on the file.id

Usage

```
rcm_change_value(file.id, column, value)
```

Arguments

`file.id`	as a string
`column`	best to just put the google drive colum letters (like "A" for the first column, or "AF")
`value`	the value to put in the column, probably safest to give a character string

`rcm_check` *Check inconsistencies in RCM*

Description

Check inconsistencies in RCM

Usage

```
rcm_check(rcm, check.archived = F, check.validated = F)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

data frame with list of inconsistencies

rcm_check_csv *write matrix issues to file for each HQ focal point*

Description

write matrix issues to file for each HQ focal point

Usage

```
rcm_check_csv(rcm = NULL, unit = "data")
```

Arguments

`rcm` use rcm_download(gdrive_links=F). can be ommited (then downloading from gdrive)

Value

writes csv files to current wd, and also opens them in excel directly!

rcm_comment *looking at this function now, it's something really insane. Thank god it's not in use anywhere. Keeping it here to prank the digital archeologists of the future*

Description

looking at this function now, it's something really insane. Thank god it's not in use anywhere. Keeping it here to prank the digital archeologists of the future

Usage

```
rcm_comment(file.id, comment, overwrite = F)
```

Arguments

`file.id` the items file id name as a string

`comment` the comment as a single character string

`overwrite` if TRUE, will completely overwrite the current comment. Otherwise it will append the new comment to the existing one (with a date stamp)

Details

#' add a value on google drive based on the file.id rcm_add_value<-function(file.id,column,value) if(is.numeric(column)) if(column>26)stop('only columns <26 can be addressed with numeric index') column<-LETTERS[column]

row<-get_gdrive_row(file.id) value_in_cell <- get_gdrive_relatedFiles(file.id) # print(file.id) # print(row) # print(value_in_cell) new_value <- paste(value_in_cell, value, sep=" / ") g_sheets_put(row,column,new_value,spreadshe = "1wX5k3cETrCbnw4vpfY07eSzTyWX6AwmJmxJQwPahrSk")

change an items comment on google drive

rcm_dashboard *compile an overview report of each countries status*

Description

compile an overview report of each countries status

Usage

```
rcm_dashboard(rcm = NULL, subs = NULL, update_rcm = T,
  unit = "data")
```

Arguments

`rcm`	can be ommited
`subs`	can be ommited
`update_rcm`	logical: whether to update the rcm from the submissions sheet before producing the report. Slows it down a bit but hey at least your report is up to date afterwards. default is TRUE
`unit`	which unit(s) to include? Defaults to 'data'. uses regex: for all units, enter a dot "."; for multiple units, use | operator (i.e. "data|reporting|research design|gis")

Value

nothing interesting, but writes a html file into the working directory and opens it in the browser

rcm_days_with_hq *how long has each item been with HQ?*

Description

how long has each item been with HQ?

Usage

```
rcm_days_with_hq(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

Date time difference vector

`rcm_deadline_status` *Days until a given date as ordinal categories*

Description

Changes a date of a deadline into words such as "today", "tomorrow", "within 30 days"

Usage

```
rcm_deadline_status(date)
```

Arguments

`date`	a vector of dates

Value

vector with ordinal values "more than 30 days", "within 30 days", "within 14 days", "within 7 days", "tomorrow", "today" and "overdue"

`rcm_delayed_to_google_sheet` *write delays to google sheet*

Description

write delays to google sheet

Usage

```
rcm_delayed_to_google_sheet(rcm)
```

Arguments

`todo`	todo list (see ?todo_download)

See Also

todo_delayed_browse()

rcm_download *download the research cycle matrix*

Description

download the research cycle matrix

Usage

```
rcm_download(include_archived = F, include_validated = F,
  after_year = "2015", main_columns_only = T, fill.dates = T,
  remove_empty = T, gdrive_links = F, raw = F)
```

Arguments

`include_archived`	logical: should archived entries be included?
`include_validated`	logical: should validated entries be included?
`after_year`	excludes all entries from before the given year (defaults to 2015)
`main_columns_only`	logical include only most important columns?
`fill.dates`	logical: should dates be filled from other dates? e.g. "first planned date" copyed to "latest planned date" where that is missing
`remove_empty`	logical: should empty rows be removed (default TRUE)
`gdrive_links`	logical: should a column with links to the google drive row be added?
`raw`	logical: if TRUE ignores all other parameters and returns the raw download (default FALSE)

Value

a data frame with the research cycle matrix

rcm_fill_dates *filling dates (used in rcm_download when fill_dates=T)*

Description

filling dates (used in rcm_download when fill_dates=T)

Usage

```
rcm_fill_dates(rcm)
```

`rcm_find_file.id` *Search RCM for terms and return the file id*

Description

Search RCM for terms and return the file id

Usage

```
rcm_find_file.id(rcm, search, unit = NULL)
```

Arguments

`search`	single string with search terms separated by a simple space. Search is not case sensitive. The best match will be returned. If there are multiple matches with the best match score, the user is prompted to select one.

Value

the id as a string

`rcm_find_issue` *applies an issue checking function to the RCM*

Description

applies an issue checking function to the RCM

Usage

```
rcm_find_issue(rcm, get_issue_ids_fun, issue_name)
```

`rcm_find_row_by_file.id` *Search RCM for terms and return the row*

Description

Search RCM for terms and return the row

Usage

```
rcm_find_row_by_file.id(rcm, search, unit = NULL)
```

Arguments

`search`	single string with search terms separated by a simple space. Search is not case sensitive. The best match will be returned. If there are multiple matches with the best match score, the user is prompted to select one.

Value

the RCM row as a data frame with a single row

---

rcm_gant *Gant chart of all RCM items*

Description

Gant chart of all RCM items

Usage

```
rcm_gant(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

ggplot item

---

rcm_gdrive_links *get google drive row links matching items in a RCM*

Description

get google drive row links matching items in a RCM

Usage

```
rcm_gdrive_links(rcm)
```

---

rcm_has_identified_status *is the RCM status coercible into a standard state?*

Description

is the RCM status coercible into a standard state?

Usage

```
rcm_has_identified_status(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

`rcm_is_data_unit_item` *do RCM rows belong to the data unit?*

Description

do RCM rows belong to the data unit?

Usage

```
rcm_is_data_unit_item(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

`rcm_is_design_unit_item` *do RCM rows belong to the research design unit?*

Description

do RCM rows belong to the research design unit?

Usage

```
rcm_is_design_unit_item(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

rcm_is_gis_unit_item *do RCM rows belong to the GIS unit?*

Description

do RCM rows belong to the GIS unit?

Usage

```
rcm_is_gis_unit_item(rcm)
```

Arguments

rcm	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

rcm_is_reporting_unit_item *do RCM rows belong to the reporting unit?*

Description

do RCM rows belong to the reporting unit?

Usage

```
rcm_is_reporting_unit_item(rcm)
```

Arguments

rcm	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

`rcm_longest_with_hq` *get rcm items that have been with hq for a long time*

Description

get rcm items that have been with hq for a long time

Usage

```
rcm_longest_with_hq(rcm, n = NULL, add.columns = c())
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

a subset of the RCM: only rows that are with HQ; only basic information columns. Sorted by added column "days.with.hq"

`rcm_missing_data_unit_items` *Identify research cycles that do not have data and analysis items*

Description

Identify research cycles that do not have data and analysis items

Usage

```
rcm_missing_data_unit_items(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

data frame with columns.. - data.missing ("exists" or "missing") - analysis.missing ("exists" or "missing") - RCID (the research cycle ID)

rcm_passed_milestone *has an item missed its milestone?*

Description

has an item missed its milestone?

Usage

```
rcm_passed_milestone(rcm)
```

Arguments

rcm	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

rcm_past_planned_submission *has the submission date passed?*

Description

has the submission date passed?

Usage

```
rcm_past_planned_submission(rcm)
```

Arguments

rcm	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

`rcm_prefill_research_tracker` *create a prefilled monthly RC tracker xlsx file*

Description

create a prefilled monthly RC tracker xlsx file

Usage

```
rcm_prefill_research_tracker(output_xlsx_file_name = paste0("tracker_",
  Sys.Date(), ".xlsx"), rcm = NULL, by_country = FALSE,
  use_xlsx_package = TRUE)
```

Arguments

`output_xlsx_file_name`	the path to the newly created file
`the`	rcm; see rcm_download()

Details

only creates an xslx template file if the _xlsx package is installed_. In any case this function will return (invisibly) a data.frame with the rows that need to be put in the tracker. if you want an xlsx template file to be created, you need to install.packages("xlsx "). Careful you must have compatible R and Java versions for this to work (both 32 or both 64 bit)

Value

writes an xlsx template with prefilled file id rows as a side effect (if xlsx package is installed). Returns a data.frame with the rows that were prefilled

`rcm_set_hours_worked` *save the hours worked in the rcm*

Description

save the hours worked in the rcm

Usage

```
rcm_set_hours_worked(file.id, hours)
```

Arguments

`file.id`	the file id
`hours`	the number of hours worked on the file id

Value

the file id

rcm_set_HQ_focal_point *set the name of the person handling the item at HQ in the gdrive RCM*

Description

set the name of the person handling the item at HQ in the gdrive RCM

Usage

```
rcm_set_HQ_focal_point(file.id, name = NULL)
```

Arguments

file.id	the file id of the item to change
name	the name of the person as a character string.

Value

returns nothing important

rcm_set_submission_date *set the submission date in the gdrive RCM*

Description

set the submission date in the gdrive RCM

Usage

```
rcm_set_submission_date(file.id, date)
```

Arguments

file.id	the file id of the item to change
date	the date to set it to. Must be of class "POSIXct".

Value

returns nothing important

`rcm_set_to_validated` *set an item's status to "validated" on google drive*

Description

set an item's status to "validated" on google drive

Usage

```
rcm_set_to_validated(file.id, hours_worked, comment = NULL,
  DDR.received = FALSE, fp)
```

Arguments

`file.id`	the items file id name as a string
`hours_worked`	time spent on the validation in hours; must be numeric or NA
`comment`	(optional) add a comment
`DDR.received`	logical; did we receive the data deletion report? defaults to FALSE
`fp`	the name of the focal point at HQ handling the item (always overwrites)

`rcm_set_to_withField` *set an item's status to "with Field" on google drive*

Description

set an item's status to "with Field" on google drive

Usage

```
rcm_set_to_withField(file.id, comment = NULL, fp)
```

Arguments

`file.id`	the items file id name as a string
`comment`	(optional) add a comment
`fp`	the name of the focal point at HQ handling the item (always overwrites)

`rcm_set_to_withHQ` *set an item's status to "with HQ" on google drive*

Description

set an item's status to "with HQ" on google drive

Usage

```
rcm_set_to_withHQ(file.id, fp, date = NULL)
```

Arguments

`file.id`	the items file id name as a string
`fp`	the name of the focal point at HQ handling the item (always overwrites)

`rcm_set_validation_date` *set the validation date in the gdrive RCM*

Description

set the validation date in the gdrive RCM

Usage

```
rcm_set_validation_date(file.id, date = as.POSIXct(Sys.Date()))
```

Arguments

`file.id`	the file id of the item to change
`date`	the date to set it to. Must be of class "POSIXct".

Value

returns nothing important

`rcm_set_withfield_date` *set the "feedback given" date in the gdrive RCM*

Description

set the "feedback given" date in the gdrive RCM

Usage

```
rcm_set_withfield_date(file.id, date = as.POSIXct(Sys.Date()))
```

Arguments

`file.id`	the file id of the item to change
`date`	the date to set it to. Must be of class "POSIXct".

Value

returns nothing important

`rcm_set_withHQ_date` *set the "actual received" date in the gdrive RCM*

Description

set the "actual received" date in the gdrive RCM

Usage

```
rcm_set_withHQ_date(file.id, date = as.POSIXct(Sys.Date()))
```

Arguments

`file.id`	the file id of the item to change
`date`	the date to set it to. Must be of class "POSIXct".

Value

returns nothing important

`rcm_show` *display a row in the rcm based on the file id*

Description

display a row in the rcm based on the file id

Usage

```
rcm_show(rcm, file.id)
```

Arguments

`rcm`	the RCM
`file.id`	the file id, or a file id search term (see '?rcm_find_row_by_file.id' for details)

Value

no output, just printing to console

`rcm_sort_priority` *sort RCM by priority*

Description

sort RCM by priority

Usage

```
rcm_sort_priority(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

the research cycle matrix sorted by 1. milestone passed, 2. days with HQ

`rcm_standardised_columns` *standardising columns*

Description

Regexing messy values into predefined categories and renames them. Applied by default in 'rcm_download()'

Usage

```
rcm_standardised_columns(rcm)
```

`rcm_status_cols` *Colours for standard item stati*

Description

Colours for standard item stati

Usage

```
rcm_status_cols()
```

Value

vector with six colour hexcodes

`rcm_submission_expected` *Is an item expected for submission?*

Description

Is an item expected for submission?

Usage

```
rcm_submission_expected(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

A subset of the research cycle matrix: all items with planned dates arrived/passed but don't have status HQ, validated, with field or with partner

`rcm_unit` *which unit do RCM rows belong to?*

Description

which unit do RCM rows belong to?

Usage

```
rcm_unit(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

character vector (with values "data", "reporting", "research design" or "GIS")

`rcm_update_from_subs` *Update RCM status based on form validation submissions*

Description

Update RCM status based on form validation submissions

Usage

```
rcm_update_from_subs(subs, rcm)
```

Arguments

`subs`	the submissions as received from subs_download()
`rcm`	the RCM as received by rcm_download()

Value

a data.frame listing which were / were not updated

`status_with_field` *is with field?*

Description

is with field?

Usage

```
status_with_field(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

`status_with_hq` *is with HQ?*

Description

is with HQ?

Usage

```
status_with_hq(rcm)
```

Arguments

`rcm`	the research cycle matrix from rcm_download(raw=F)

Value

logical vector

`subs_browse` *open submission google sheet in browser*

Description

open submission google sheet in browser

Usage

```
subs_browse()
```

subs_download *downloads items submitted for validation through the shiny submission form (stored in gdrive)*

Description

downloads items submitted for validation through the shiny submission form (stored in gdrive)

Usage

```
subs_download()
```

Value

a data frame with all submissions

subs_rcm_rows *find submission rows in the research cyclce matrix*

Description

find submission rows in the research cyclce matrix

Usage

```
subs_rcm_rows(subs, rcm)
```

Arguments

subs	the submissions from subs_download()
rcm	the research cycle matrix from rcm_download()

Value

a vector with row indices

Examples

```
# get the matching rcm rows:
rcm<-rcm_download()
subs<-subs_download()
subs_in_rcm_indices<-subs_rcm_rows(subs,rcm)
# subset rcm
rcm[subs_in_rcm_indices,]
```

`subs_status` *Status of subs (asfound in rcm)*

Description

Status of subs (asfound in rcm)

Usage

```
subs_status(subs, rcm)
```

`subs_with_new_id` *Get only submissions where new ids were suggested*

Description

Get only submissions where new ids were suggested

Usage

```
subs_with_new_id(subs)
```

Arguments

`subs`	the submission list as received by subs_download()

Value

the subset of the submission list

`todo_create` *make ’todo’ item from rcm and subs*

Description

make ’todo’ item from rcm and subs

Usage

```
todo_create(rcm, subs, who = ".")
```

Arguments

`rcm`	RCM from rcm_download(include_validated=T,include_archived=T)
`subs`	Submissions from subs_download()
`who`	your first name, i.e. "Karen". Allows regex (for example put "." to get all items)

Value

the todo item (silently)

See Also

todo_download()

`todo_delayed` *which data unit items should have already been submitted but havent?*

Description

which data unit items should have already been submitted but havent?

Usage

```
todo_delayed(rcm, days_since_planned_submission = 14)
```

Arguments

`rcm`	the research cycle matrix from rcm_download()
`days_since_planned_submission`	number of days grace period; if 14 (default), shows only items that are more than two weeks overdue

`todo_delayed_browse` *open delays google sheet in browser*

Description

open delays google sheet in browser

Usage

```
todo_delayed_browse()
```

`todo_delayed_to_csv` *create per country csv files listing all delayed items*

Description

create per country csv files listing all delayed items

Usage

```
todo_delayed_to_csv(todo, path = "./delayed/")
```

Arguments

`todo`	see ?todo_download

`todo_download` *Download 'todo' item list*

Description

Download 'todo' item list

Usage

```
todo_download(who = ".")
```

Arguments

`who`	your first name, i.e. "Karen". Allows regex (for example put "." to get all items)

Details

actually downloads the research cycle matrix and the submissions sheet and mushes them up into a todo list The list is ordered by priority using 1. flagged as emergency? 2. Days

Value

the todo item (silently)

See Also

todo_download()

`todo_next` *show next on todo list*

Description

show next on todo list

Usage

```
todo_next(todo, n = nrow(todo))
```

Arguments

`todo`	'todo' list element (mix of rcm and subs; created with todo_download())
`n`	number of items to show

Details

will print the highest priority item last

Value

the todo item (silently)

`todo_validate_next` *set the top item in the todo list to validated on the RCM*

Description

set the top item in the todo list to validated on the RCM

Usage

```
todo_validate_next(todo, hours_worked = NA)
```

Arguments

`todo`	’todo’ list element (mix of rcm and subs; created with todo_download())

Details

will prompt for confirmation (type ’y’ in console and hit enter; everything else will abort)

Value

the todo item (silently), without the validated entry

`validation_timeline` *plot validation timeline*

Description

plot validation timeline

Usage

```
validation_timeline(rcm, passed_days_included = 30,
  future_days_included = 30)
```

Arguments

`rcm`	the RCM as given by rcm_download()

Value

ggplot object

Index