Package 'datacleanr'

July 22, 2025

Title Interactive and Reproducible Data Cleaning

Version 1.0.5

Description Flexible and efficient cleaning of data with interactivity.
'datacleanr' facilitates best practices in data analyses and reproducibility with built-in features and by translating interactive/manual operations to code.
The package is designed for interoperability, and so seamlessly fits into reproducible analyses pipelines in 'R'.

License GPL-3

Suggests testthat (>= 2.1.0)

Encoding UTF-8

RoxygenNote 7.3.1

URL https://github.com/the-Hull/datacleanr

BugReports https://github.com/the-Hull/datacleanr/issues

Imports shiny (>= 1.5.0), summarytools (>= 0.9.6), dplyr (>= 1.0.2), rlang (>= 0.4.9), DT (>= 0.16), magrittr (>= 2.0.1), plotly (>= 4.9.2.1), grDevices, stats, purrr (>= 0.3.4), glue (>= 1.4.2), formatR (>= 1.7), RColorBrewer (>= 1.1.2), clipr (>= 0.7.1), rstudioapi (>= 0.13), utils, lubridate (>= 1.7.9.2), shinyWidgets (>= 0.5.4), htmlwidgets (>= 1.5.3), tools, fs (>= 1.5.0), shinyFiles (>= 0.8.0), bslib

Depends R (>= 3.6)

NeedsCompilation no

Author Alexander Hurley [cre, aut, cph] (ORCID: <https://orcid.org/0000-0002-9641-2805>), Richard Peters [ctb] (ORCID: <https://orcid.org/0000-0002-7441-1297>), Christoforos Pappas [ctb] (ORCID: <https://orcid.org/0000-0001-5721-557X>)

Maintainer Alexander Hurley <agl.hurley@gmail.com>

Repository CRAN

Date/Publication 2025-05-10 18:10:05 UTC

Contents

`apply_data_set_up` *Applies grouping to data set conditionally*

Description

Applies grouping to data set conditionally

Usage

```
apply_data_set_up(df, group)
```

Arguments

`df`	data frame
`group`	supply reactive output from group selector

Value

returns df either grouped or not

`calc_limits_per_groups`
Return x and y limits of "group-subsetted" dframe

Description

Used for adjusting layout of plotly plot based on selected groups in `group_selector_table`; currently used in viz tab

Usage

```
calc_limits_per_groups(dframe, group_index, xvar, yvar, scaling = 0.02)
```

Arguments

dframe	dataframe/tibble, grouped/ungrouped
group_index	numeric, group indices for which to return lims
xvar	character, name of x var for plot (must exist in dframe)
yvar	character, name of y var for plot (must exist in dframe)
scaling	numeric, 1 +/- scaling times limits

Value

list with xlim and ylim

---

can_internet *Check for internet connection*

Description

Check for internet connection

Usage

```
can_internet(url = "http://www.google.com")
```

Arguments

url	character, valid path to url - user responsible

Value

logical - TRUE or FALSE

---

check_individual_statement *check if a filter statement is valid*

Description

check if a filter statement is valid

Usage

```
check_individual_statement(df, statement)
```

Arguments

df	data frame / tibble to be filtered
statement	character string,

Value

logical, did filter statement work?

---

datacleanr_server *datacleanr server function*

Description

datacleanr server function

Usage

```
datacleanr_server(input, output, session, dataset, df_name, is_on_disk)
```

Arguments

input, output, session	standard shiny boilerplate
dataset	data.frame, tibble or data.table that needs cleaning
df_name	character, name of dataset or file_path passed into shiny app
is_on_disk	logical, whether df was read from file

---

dcr_app *Interactive and reproducible data cleaning*

Description

Launches the datacleanr app for interactive and reproducible cleaning. See Details for more information.

Usage

```
dcr_app(dframe, browser = TRUE)
```

Arguments

dframe	Character, a string naming a data.frame, tbl or data.table in the environment or a path to a .Rds file. **Note, that** data.table**s are converted to tibbles internally.**‘
browser	logical, should app start in OS’s default browser? (default TRUE)

Details

datacleanr provides an interactive data overview, and allows reproducible filtering and (manual, interactive) visual outlier detection and annotation across multiple app tabs:

- **Overview and Set-up**: set groups (see below) and generate a exploratory summary of dframe
- **Filtering**: Provide and apply filter statements (groupwise, see below and filter_scoped_df)
- **Visualization and Annotating**: interactive visualization allowing outlier highlighting, annotating and before/after histograms of displayed (numeric) variables
- **Extraction**: generates *Reproducible Recipe* and outputs

For data sets exceeding 1.5 million rows, we suggest splitting the data, if possible, by a grouping factor. This is because at this volume interactive visualizations using plotly stretch the limits of what modern web browsers can handle. A simple example using iris is:

```
iris_split <- split(iris, iris$Species)
dcr_app(iris_split[[1]])
# or
lapply(iris_split, dcr_app)
```

Extensive documentation is provided on each of the tabs for individual procedures in help links. datacleanr relies on 1) generating a column of unique IDs (.dcrkey) and subsetting dframe into sub-groups (generated in-app, added as column .dcrindex) for filtering and visualization. These groups are composed of unique combinations of columns in the data set (must be factor) and are passed to group_by, and are carried through the app for exploratory analyses (tab **Overview and Set-up**), filtering (tab **Filtering**) and plotting (tab **Visualization**). These groups should ideally be chosen to facilitate a convenient filtering and viewing/cleaning process. For example, a data set with time series of multiple sensors could be grouped by sensor and/or additional columns, such that periods of interest can be visualized and cleaned simultaneously in the interactive plot.

Filtering is achieved by providing expressions that evaluate to TRUE \ FALSE, and can be applied to the entire data set, or individual/all groups via scoped filtering (see filter_scoped_df).

The interactive visualization allows selecting and deselecting points with lasso and box select tools, as well as interactive zooming (toolbar or clicking on legend items or group overview table, see tab in-app) as well as panning (toolbar and hover over plot's axes). Data formats supported are

1. Observational (numeric), timeseries (POSIXct) and categorical data in x and y dimensions/axis
2. Observational (numeric) data in z dimension (point size)
3. Spatial data, when lon and lat in decimal degrees are present in x and y.

Displaying spatial data requires a Mapbox account, from which an access token needs to be copied into your .Renviron (e.g. MAPBOX_TOKEN=your_copied_token).

Note, that when a column .dcrflag (logical, TRUE \ FALSE) is present in dframe, respective observations are given contrasting symbols (FALSE = circle, TRUE = star-triangle). This column is employed as a cross-referencing tool for e.g.other outlier detection or data-processing algorithms that were applied prior.

The tab **Extraction** provides code to reproduce the entire procedure (a *Reproducible Recipe*), which

1. can be copied, or sent directly to an active RStudio script when used interactively (i.e. when dframe is an object in R's environment),
2. can be saved to disk with intermediate outputs (filter statements and selected outliers), where file names are based on the input file and configurable suffixes when dframe is a path.

Value

When datacleanr is ended by clicking on Close in the app's navigation bar, a list is **invisibly** returned with the following items:

1. **df_name**: character, object name/file path passed into dcr_app
2. **dcr_df**: tibble, filtered data set **with** additional columns .dcrkey, .dcrindex, .annotation - the latter is NA for non-outliers, an empty string for outliers without annotation, and a custom string for annotated outliers
3. **dcr_selected_outliers**: data.frame, contains the outlier .dcrkey, the .annotation and a selection_count (integer, count incrementer) column
4. **dcr_groups**: character, a vector defining the groups (via group_by) used throughout datacleanr
5. **dcr_condition_df**: tibble, with columns filter (character, statement used for filtering) and group (list, of integers), defining groups that correspond to .dcrindex
6. **dcr_code**: character string, containing *Reproducible Recipe*

---

dcr_checks *Initial checks for data set*

Description

Initial checks for data set

Usage

```
dcr_checks(dframe)
```

Arguments

dframe	dframe supplied to dcr_app

extend_palette *extend brewer palette*

Description

extend brewer palette

Usage

```
extend_palette(n)
```

Arguments

n	numeric, number of colors

Value

color vector of length n

filter_scoped *Apply filter based on a statement, scoped to* dplyr *groups*

Description

Apply filter based on a statement, scoped to dplyr groups

Usage

```
filter_scoped(dframe, statement, scope_at = NULL)
```

Arguments

dframe	data.frame/tbl, grouped or ungrouped
statement	character, statement for filtering (only VALID expressions; use check_individual_statement to grab only valid.
scope_at	numeric, group indices to apply filter statements to

Value

List, containing item filtered_df, a data.frame filtered based on statements and scope.

filter_scoped_df *Filter / Subset data* dplyr*-groupwise*

Description

filter_scoped_df subsets rows of a data frame based on grouping structure (see group_by). Filtering statements are provided in a separate tibble where each row represents a combination of a logical expression and a list of groups to which the expression should be applied to corresponding to see indices from cur_group_id).

Usage

```
filter_scoped_df(dframe, condition_df)
```

Arguments

dframe	A grouped or ungrouped tibble or data.frame
condition_df	A tibble with two columns; condition_df[,1] with character strings which evaluate to valid logical expressions applicable in subset or filter, and condition_df[,2], a list-column with group scoping levels (numeric) or NULL for unscoped filtering. If all groups are given for a statement, the operation is the same as for a grouped data.frame in filter.

Details

This function is applied in the "Filtering" tab of the datacleanr app, and applied in the reproducible code recipe in the "Extract" tab. Note, that multiple checks for valid statements are performed in the app (and only valid operations printed in the "Extract" tab). It is therefore not advisable to manually alter this code or use this function interactively.

Value

An object of the same type as dframe. The output is a subset of the input, with groups and rows appearing in the same order, and an additional column .dcrindex representing the group indices. The output may have less groups as the input, depending on subsetting.

Examples

```
# set-up condition_df
cdf <- dplyr::tibble(
  statement = c(
    "Sepal.Width > quantile(Sepal.Width, 0.1)",
    "Petal.Width > quantile(Petal.Width, 0.1)",
    "Petal.Length > quantile(Petal.Length, 0.8)"
  ),
  scope_at = list(NULL, NULL, c(1, 2))
)
```

```
fdf <- filter_scoped_df(
  dplyr::group_by(
    iris,
    Species
  ),
  condition_df = cdf
)

# Example of invalid expression:
# column 'Spec' does not exist in iris
# "Spec == 'setosa'"
```

`get_factor_cols_idx` *Identify columns carrying non-numeric values*

Description

Identify columns carrying non-numeric values

Usage

```
get_factor_cols_idx(x)
```

Arguments

x	data.frame

Value

logical, is column in x non-numeric?

`handle_add_outlier_trace`
Handle outlier trace

Description

Single outlier trace is added to plotly; interactive select/deselect was implemented by adjusting `selected_points`, and subsequently adding, or deleting+adding the (modified) trace at the end of the existing JS data array. Requires tracemap with trace names and corresponding indices. Simple check for re-execution was implemented by passing on the selection keys to compare against on pertinent `plotly_event`.

Usage

```
handle_add_outlier_trace(
  sp,
  dframe,
  ok,
  selectors,
  trace_map,
  source = "scatterselect",
  session
)
```

Arguments

sp	selected points
dframe	plot data
ok	reactive, old keys
selectors	reactive input selectors
trace_map	numeric, max trace id
source	plotly source
session	active session

---

handle_restyle_traces *Wrapper for adjusting axis lims and hiding traces*

Description

Wrapper for adjusting axis lims and hiding traces

Usage

```
handle_restyle_traces(
  source_id,
  session,
  dframe,
  scaling = 0.05,
  xvar,
  yvar,
  trace_map,
  max_id_group_trace,
  input_sel_rows,
  flush = TRUE
)
```

Arguments

source_id	character, plotly source id
session	session object
dframe	data frame/tibble (grouped/ungrouped)
scaling	numeric, 1 +/- scaling applied to x lims for xvar and yvar
xvar	character, name of xvar, must be in dframe
yvar	character, name of yvar, must be in dframe
trace_map	matrix, with columns for trace name (col 1) and trace id (col 2)
max_id_group_trace	numeric, max id of plotly trace from original data (not outlier traces)
input_sel_rows	numeric, input from DT grouptable
flush	character, plotlyProxy settings

Value

Used for it's side effect - no return

---

handle_sel_outliers *Handle selection of outliers (with select - unselect capacity)*

Description

Handle selection of outliers (with select - unselect capacity)

Usage

```
handle_sel_outliers(sel_old_df, sel_new)
```

Arguments

sel_old_df	data.frame of selection info
sel_new	data.frame, event data from plotly, must have column customdata

Value

updated selection data frame

hide_trace_idx *Provide trace ids to set to invisible*

Description

Provide trace ids to set to invisible

Usage

```
hide_trace_idx(trace_map, max_groups, selected_groups)
```

Arguments

trace_map	matrix, with cols trace name (col 1), trace id (col 2)
max_groups	numeric, number of groups in grouptable
selected_groups	groups highlighted in grouptable

Details

Provides the indices (JS notation, starting at 0) for indices that are set to `visible = 'legendonly'` through `plotly.restyle`

make_group_table *Make grouping overview table*

Description

Make grouping overview table

Usage

```
make_group_table(dframe)
```

Arguments

dframe	data.frame

Value

tibble with one row per group

make_save_filepath *Wrapper for saving files*

Description

Wrapper for saving files

Usage

```
make_save_filepath(save_dir, input_filepath, suffix, ext)
```

Arguments

save_dir	character, selected save dir
input_filepath	character, original file path to folder
suffix	character, e.g. 'CLEAN' or 'cleaning_script'
ext	character, file extension, no dot!!

Value

OS-conform file path for saving

module_server_apply_reset *Server Module: apply / reset filter*

Description

Server Module: apply / reset filter

Usage

```
module_server_apply_reset(input, output, session, df_filtered, df_original)
```

Arguments

input, output, session	standard
df_filtered	reactive, filtered df
df_original	reactive, original df

module_server_box_str_filter

Server Module: box for str filter condition

Description

Server Module: box for str filter condition

Usage

```
module_server_box_str_filter(input, output, session, selector, actionbtn)
```

Arguments

input, output, session	standard
selector	character, html selector for placement
actionbtn	reactive, action button counter

module_server_checkbox

Server Module: checkbox rendering

Description

Server Module: checkbox rendering

Usage

```
module_server_checkbox(input, output, session, text)
```

Arguments

input, output, session	standard shiny boilerplate
text	Character, appears next to checkbox (or coerced)

`module_server_df_filter`
Server Module: filter info text and filtered df output

Description

Server Module: filter info text and filtered df output

Usage

```
module_server_df_filter(input, output, session, dframe, condition_df)
```

Arguments

`input`, `output`, `session`	standard `shiny` boilerplate
`dframe`	data frame/tibble for filtering
`condition_df`	data frame/tibble with filtering conditions and grouping scope

Value

df, either filtered or original, based on validity of `statements` in `condition_df`

`module_server_extract_code`
Server Module: Selection Annotator

Description

Server Module: Selection Annotator

Usage

```
module_server_extract_code(
  input,
  output,
  session,
  df_label,
  filter_df,
  gvar,
  statements,
  sel_points,
  overwrite,
  is_on_disk,
  out_path
)
```

Arguments

input, output, session	standard shiny boilerplate
df_label	string, name of original df input
filter_df	reactiveValue data frame with filter statements and scoping lvl
gvar	reactive character, grouping vars for dplyr::group_by
statements	reactive, lgl, vector of working statements
sel_points	reactiveValue, data frame with selected point keys, annotations, and selection count
overwrite	reacive value, TRUE/FALSE from checkbox input
is_on_disk	Logical, whether df represented by df_label was on disk or from interactive R use
out_path	reactive, List, with character strings providing directory paths and file names for saving/reading in code output

---

module_server_extract_code_fileconfig

Server Module: Extraction File selection menu

Description

Server Module: Extraction File selection menu

Usage

```
module_server_extract_code_fileconfig(
  input,
  output,
  session,
  df_label,
  is_on_disk,
  has_processed
)
```

Arguments

input, output, session	standard shiny boilerplate
df_label	character, name of original df input
is_on_disk	Logical, whether df represented by df_label was on disk or from interactive R use
has_processed	reactive, logical, TRUE if filtered / selected points

module_server_filter_str

Server Module: box for str filter condition

Description

Server Module: box for str filter condition

Usage

```
module_server_filter_str(input, output, session, dframe)
```

Arguments

input, output, session	standard shiny boilerplate
dframe	data frame passed into dcr app

Details

provides UI text box element

module_server_group_relayout_buttons

Server Module: Selection Annotator

Description

Server Module: Selection Annotator

Usage

```
module_server_group_relayout_buttons(input, output, session, startscatter)
```

Arguments

input, output, session	standard shiny boilerplate
startscatter	reactive, actionbutton value

Details

provides UI text box element

Value

reactive values with input xvar, yvar and actionbutton counter

module_server_group_select
Server Module: group selection

Description

Server Module: group selection

Usage

```
module_server_group_select(input, output, session, dframe)
```

Arguments

input, output, session	standard
dframe	data frame for filtering

module_server_group_selector_table
Server Module: box for str filter condition

Description

Server Module: box for str filter condition

Usage

```
module_server_group_selector_table(input, output, session, df, df_label, ...)
```

Arguments

input, output, session	standard shiny boilerplate
df	data frame (either from overview or filtering tab)
df_label	character, original input data frame
...	arguments passed to datatable()

Details

provides UI text box element

`module_server_histograms` *Server Module: dynamic histogram output for n vars str filter condition*

Description

Server Module: dynamic histogram output for n vars str filter condition

Usage

```
module_server_histograms(
  input,
  output,
  session,
  dframe,
  selector_inputs,
  sel_points
)
```

Arguments

`input`, `output`, `session`	standard `shiny` boilerplate
`dframe`	df
`selector_inputs`	reactive vals from above-plot controls,
`sel_points`	reactive, provides .dcrkey of selected points

Details

provides UI buttons for deleting last / entire outlier selection

Value

reactive values with input xvar, yvar and actionbutton counter

module_server_lowercontrol_btn

Server Module: box for str filter condition

Description

Server Module: box for str filter condition

Usage

```
module_server_lowercontrol_btn(
  input,
  output,
  session,
  selector_inputs,
  action_track
)
```

Arguments

input, output, session	standard shiny boilerplate
selector_inputs	reactive vals from above-plot controls, used to determine if plot is a map (lon/lat)
action_track	reactive, logical - has plot been pressed?

Details

provides UI buttons for deleting last / entire outlier selection

Value

reactive values with input xvar, yvar and actionbutton counter

module_server_plot_annotation_table

Server Module: DT for annotation

Description

Server Module: DT for annotation

Usage

```
module_server_plot_annotation_table(input, output, session, dframe, sel_points)
```

Arguments

input, output, session	standard shiny boilerplate
dframe	df used for plotting
sel_points	numeric, vector of .dcrkeys selected in plot

Value

df with .dcrkeys and annotations

---

module_server_plot_selectable

Server Module: box for str filter condition

Description

Server Module: box for str filter condition

Usage

```
module_server_plot_selectable(
  input,
  output,
  session,
  selector_inputs,
  df,
  sel_points,
  mapstyle
)
```

Arguments

input, output, session	standard shiny boilerplate
selector_inputs	reactive, output from module_plot_selectorcontrols
df	reactive df
sel_points	reactive, provides .dcrkey of selected points
mapstyle	reactive, selected mapstyle from below-plot controls

Details

provides plot, note, that data set needs a column .dcrkey, added in initial processing step

module_server_plot_selectorcontrols *Server Module: box for str filter condition*

Description

Server Module: box for str filter condition

Usage

```
module_server_plot_selectorcontrols(input, output, session, df)
```

Arguments

input, output, session	standard shiny boilerplate
df	df (not reactive - prevent re-execution of observer)

Details

provides UI text box element

Value

reactive values with input xvar, yvar and actionbutton counter

module_server_summary *Server Module: data summary*

Description

Server Module: data summary

Usage

```
module_server_summary(
  input,
  output,
  session,
  dframe,
  df_label,
  start_clicked,
  group_var_check
)
```

Arguments

input, output, session	standard shiny boilerplate
dframe	reactive, input data frame
df_label	character, name of initial data set
start_clicked	reactive holding start action button
group_var_check	reactive holding group check output

---

module_server_text_annotator *Server Module: Selection Annotator*

Description

Server Module: Selection Annotator

Usage

```
module_server_text_annotator(input, output, session, sel_data)
```

Arguments

input, output, session	standard shiny boilerplate
sel_data	reactive df

Details

provides UI text box element

Value

reactive values with input xvar, yvar and actionbutton counter

module_ui_apply_reset *UI Module: Apply/Reset Filtering*

Description

UI Module: Apply/Reset Filtering

Usage

```
module_ui_apply_reset(id)
```

Arguments

id	Character, identifier for variable selection

module_ui_box_str_filter *UI Module: box for str filter condition*

Description

UI Module: box for str filter condition

Usage

```
module_ui_box_str_filter(id, actionbtn)
```

Arguments

id	Character, identifier for variable selection
actionbtn	reactive, action button counter

module_ui_checkbox *UI Module: data summary*

Description

UI Module: data summary

Usage

```
module_ui_checkbox(id, cond_id)
```

Arguments

id	shiny standard
cond_id	character,

module_ui_df_filter *UI Module: filter info text output*

Description

UI Module: filter info text output

Usage

```
module_ui_df_filter(id)
```

Arguments

id	character, shiny namespacing

Value

UI text element giving number of failed filters and percent of filtered rows

module_ui_extract_code *UI Module: Extraction Text output*

Description

UI Module: Extraction Text output

Usage

```
module_ui_extract_code(id)
```

Arguments

id	Character string

module_ui_extract_code_fileconfig *UI Module: Extraction File selection menu*

Description

UI Module: Extraction File selection menu

Usage

```
module_ui_extract_code_fileconfig(id)
```

Arguments

id	Character string

module_ui_filter_str *UI Module: box for str filter condition*

Description

UI Module: box for str filter condition

Usage

```
module_ui_filter_str(id)
```

Arguments

id	Character string

module_ui_group_relayout_buttons *UI Module: Grouptable Relayout Buttons*

Description

UI Module: Grouptable Relayout Buttons

Usage

```
module_ui_group_relayout_buttons(id)
```

Arguments

id	Character string

module_ui_group_select *UI Module: group selection*

Description

UI Module: group selection

Usage

```
module_ui_group_select(id)
```

Arguments

id	Character, identifier for variable selection

module_ui_group_selector_table *UI Module: box for str filter condition*

Description

UI Module: box for str filter condition

Usage

```
module_ui_group_selector_table(id)
```

Arguments

id	Character string

module_ui_histograms *UI Module: dynamic histogram output for n vars*

Description

UI Module: dynamic histogram output for n vars

Usage

```
module_ui_histograms(id)
```

Arguments

id	Character string

`module_ui_lowercontrol_btn` *UI Module: Delete selection buttons*

Description

UI Module: Delete selection buttons

Usage

```
module_ui_lowercontrol_btn(id)
```

Arguments

`id`	Character string

`module_ui_plot_annotation_table` *UI Module: DT for annotation*

Description

UI Module: DT for annotation

Usage

```
module_ui_plot_annotation_table(id)
```

Arguments

`id`	Character string

`module_ui_plot_selectable` *UI Module: plotly plot*

Description

UI Module: plotly plot

Usage

```
module_ui_plot_selectable(id)
```

Arguments

`id`	Character string

module_ui_plot_selectorcontrols *UI Module: selector controls*

Description

UI Module: selector controls

Usage

```
module_ui_plot_selectorcontrols(id)
```

Arguments

id	Character string

module_ui_summary *UI Module: data summary*

Description

UI Module: data summary

Usage

```
module_ui_summary(id)
```

Arguments

id	shiny standard

module_ui_text_annotator *UI Module: Selection Annotator*

Description

UI Module: Selection Annotator

Usage

```
module_ui_text_annotator(id)
```

Arguments

id	Character string

print.dcr_code *Method for printing dcr_code output*

Description

Method for printing dcr_code output

Usage

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

Arguments

x	character, code output from dcr_app
...	additional arguments passed to cat

split_groups *Split data.frame/tibble based on grouping*

Description

Split data.frame/tibble based on grouping

Usage

```
split_groups(dframe)
```

Arguments

dframe	data.frame

Value

list of data frames

Index