Package: repana (via r-universe)

October 17, 2024

Type Package

Title Repeatable Analysis in R

Version 2.2.0.9000

URL https://github.com/johnaponte/repana,

https://johnaponte.github.io/repana/

Description Set of utilities to facilitate the reproduction of analysis in R. It allow to make_structure(), clean_structure(), and run and log programs in a predefined order to allow secondary files, analysis and reports be constructed in an ordered and reproducible form.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Depends DBI (>= 1.0)

Suggests spelling, testthat, knitr, RSQLite, RPostgres, duckdb, pacman, targets

Imports config, dplyr, magrittr, lubridate, rmarkdown, processx, readr, rstudioapi, pool, openxlsx, tools, yaml, digest

Language en-US

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Repository https://johnaponte.r-universe.dev

RemoteUrl https://github.com/johnaponte/repana

RemoteRef HEAD

RemoteSha 163d220bbcfbaf3b888a1dd3e9d9c53151343a35

Contents

`addDataTable` *A wrap to* `writeDataTable`

Description

Include a `data.frame` into a workbook applying a tableStyle and an auto width to the column. For better results you could setup `options("openxlsx.minWidth" = 6)`

Usage

```
addDataTable(
  wb,
  df,
  sheet,
  tableName,
  tableStyle = "TableStyleMedium1",
  withFilter = TRUE,
  firstActiveCol = NULL,
  ...
)
```

Arguments

`wb`	a workbook object
`df`	a data.frame
`sheet`	the sheet name. If missing the name of the data.frame
`tableName`	a name for the table in the excel document. If missing the name of the data.frame
`tableStyle`	a tableStyle name

withFilter	if TRUE the filter is included
firstActiveCol	First column active on the the freeze panel
...	other parameters for the writeDataTable

Value

silently the wb

Examples

```
## Not run:
library(openxlxs)
options("openxlsx.minWidth" = 6)
wb <- createWorkbook(title = "Test addDataTable")
addDataTable(wb,iris)
saveWorkbook(wb, "test_addDataTable.xlsx")

## End(Not run)
```

---

clean_database *Helper function to drop all tables from a database*

Description

Helper function to drop all tables from a database

Usage

```
clean_database(con)
```

Arguments

con	DBI connection

Value

invisible, a lit with result from DBI

clean_structure *Clean the secondary files of the project*

Description

Delete and make new `database`, `logs` and `reports` directory

Usage

```
clean_structure()
```

Value

Invisible, the directories defined by the clean_before_new_analysis entry in the config.yml file.

Author(s)

John J. Aponte

get_con *Get a DBI connection reading a configuration file*

Description

This function get a DBI connection to a database reading the parameters from a `config.yml` file using the `get` function. See the vignette for details.

Usage

```
get_con(configname = "defaultdb", file = "config.yml")
```

Arguments

`configname`	a string with the name of the configuration
`file`	name of the configuration to read from.

Value

a `DBI` connection

Author(s)

John J. Aponte

get_dirs *Get the* dirs *section of the* config.yml *file*

Description

It is a wrap of `config::get("dirs")`.

Usage

```
get_dirs(file = "config.yml")
```

Arguments

file	by default the config.yml file

Value

a list with the directory entries

get_pool *Get a pool connection reading a configuration file*

Description

This function get a pool connection to a database reading the parameters from a config.yml file using the get function. See the vignette for details.

Usage

```
get_pool(configname = "defaultdb", file = "config.yml")
```

Arguments

configname	a string with the name of the value
file	name of the configuration to read from.

Value

a dbPool connection object

Author(s)

John J. Aponte

`insert_template` *RStudio addin app to insert a template to the heading part of a snip code*

Description

RStudio addin app to insert a template to the heading part of a snip code

Usage

```
insert_template()
```

Value

The template to insert

`make_structure` *Make the structure for a new project*

Description

Make the following directories

- data to keep the data necessary for the project
- database to keep the secondary, modified dataset and objects
- handmade to keep reports and dataset modified by hand or not make by the automatic stream
- logs to keep logs of the automatic stream
- reports to keep the automatic reports

The `data`, `handmade` are not clean. The rest are clean as they should be reproduced by the automatic stream. Do not forget to include them in `.gitignore` if you use `git`

Usage

```
make_structure()
```

Value

the `dir` structure

Author(s)

John J. Aponte

`master` *Render programs*

Description

By default, all programs with the pattern "nn_" will be executed in order. The 'Start' and 'Stop' parameters can be used to modify the files to start or stop at a different number.

Usage

```
master(
  start = 0,
  stop = Inf,
  format = "html",
  logdir = config::get("dirs")$logs
)
```

Arguments

`start`	program to start
`stop`	program to stop
`format`	Format to render the programs values accepted are "pdf", "html" and "word"
`logdir`	directory to keep the logs of the files. By default the entry on the `config.yml` `dirs:logs`

Details

The files are treated as *snip* files for rmarkdown and render in the log directory, with the format specified in the *format* parameter.

The default the log directory is configured in the config.yml file.

Value

a data.frame with the files run, running time and exit status

`master_txt` *Runs the programs*

Description

Run the programs specified by the pattern, in the order as the pattern select the files. Keep a log of the results. By default the pattern starting with two numbers and ending with .R is selected. They are run in order

Usage

```
master_txt(
  pattern = "^[0-9][0-9].*\\.R$",
  start = 1,
  stop = Inf,
  logdir = config::get("dirs")$logs,
  rscript_path
)
```

Arguments

pattern	Regular expression to select the files to run
start	index of the program to start
stop	index of the program to stop
logdir	directory to keep the logs of the files. By default
rscript_path	path to the Rscript file the entry on the config.yml dirs:logs

Details

The program add (or create if not exists) the files run, time of execution and exit status in the file master.log of the directory logs.

WARNING: This is a legacy program. Use master instead.

Value

a data.frame with the files run, running time and exit status

---

render_report *A wrap to render a markdown report*

Description

Render the report and copy it to the outputdir directory. More formats are available but only three are included here ‘

Usage

```
render_report(report, format = "pdf", outputdir = get_dirs()$reports, ...)
```

Arguments

report	filename of the report
format	output format. "pdf","html","word" are valid entries
outputdir	directory to save the report
...	other parameters for render function

Value

No return value, called for side effects to render the reports

See Also

render

Examples

```
## Not run:
render_report(myreport.rmd,"pdf")

## End(Not run)
```

`targets_structure` *Create configuration to use targets*

Description

Create of they do not exist the following directories

- R
- dat
- out

Usage

```
targets_structure()
```

Details

If does not exists, create a simplify version of the `_targets.R` script, modify the `.gitignore`. Also add a `config.yml` and a `_template.txt` files.

Examples

```
 ## Not run:
targets_structure()

## End(Not run)
```

update_table — *Helper function to include a data.frame in the database and update the log*

Description

Helper function to include a data.frame in the database and update the log

Usage

```
update_table(con, table, source, tablename)
```

Arguments

con	DBI connection
table	the data.frame to be included in the database
source	a manual comment to identify the source of the table
tablename	if present, the data frame will be saved with this name.

Value

the result from DBI otherwise the name of the data.frame

write_log — *Make a log of the updates on the database*

Description

Make a log of the database updates. If the log table does not exists it creates it Make a new entry with the timestamp of the update

Usage

```
write_log(con, tablename, source)
```

Arguments

con	DBI connection
tablename	the name of the table
source	a manual comment to identify the source of the table

Value

the result from DBI

Index