Package: plnr (via r-universe)

September 14, 2024

Title A System for Planing Analyses

Version 2022.11.23

Description A system to plan analyses within the mental model where you have one (or more) datasets and want to run either A) the same function multiple times with different arguments, or B) multiple functions. This is appropriate when you have multiple strata (e.g. locations, age groups) that you want to apply the same function to, or you have multiple variables (e.g. exposures) that you want to apply the same statistical method to, or when you are creating the output for a report and you need multiple different tables or graphs.

License MIT + file LICENSE

URL https://www.csids.no/plnr/, https://github.com/csids/plnr

BugReports https://github.com/csids/plnr/issues

Encoding UTF-8

LazyData true

Depends R (>= 3.3.0)

Imports data.table, digest, fs, foreach, glue, pbmcapply, purrr, R6, stats, usethis, utils, uuid

Suggests testthat, knitr, rmarkdown, progressr, ggplot2, readxl, magrittr

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

VignetteBuilder knitr

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

RemoteUrl https://github.com/csids/plnr

RemoteRef HEAD

RemoteSha 292c17ede611f32ed247e9816dbf152fb7f349d7

Contents

`create_rmarkdown` *Create example rmarkdown project*

Description

Create example rmarkdown project

Usage

```
create_rmarkdown(home)
```

Arguments

`home`	Location of the 'home' directory

`example_action_fn` *An example action_fn for an analysis*

Description

An example action_fn for an analysis

Usage

```
example_action_fn(data, argset)
```

Arguments

`data`	Named list.
`argset`	Named list.

Examples

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argsets_as_dt()
p$run_one("analysis_1")
```

---

example_data_fn_nor_covid19_cases_by_time_location

An example data_fn that returns a data set

Description

An example data_fn that returns a data set

Usage

```
example_data_fn_nor_covid19_cases_by_time_location()
```

---

expand_list

Expand (cross) lists

Description

The same as purrr::cross, but doesn't require an extra list()

Usage

```
expand_list(...)
```

Arguments

...	Dots

Examples

```
plnr::expand_list(
  a = 1:2,
  b = c("a", "b")
)
```

get_anything *Gets anything (including with package scoping)*

Description

base::get does not work with package scoping (e.g. get("pkg::var")). plnr::get_anything works with package scoping.

Usage

```
get_anything(x)
```

Arguments

x	the string that we are getting

Examples

```
plnr::get_anything("plnr::nor_covid19_cases_by_time_location")
```

is_run_directly *Is this code run directly?*

Description

This function determines if it is being called from within a function or if it is being run directly

Usage

```
is_run_directly()
```

nor_covid19_cases_by_time_location *Covid-19 data for PCR-confirmed cases in Norway (nation and county)*

Description

This data comes from the Norwegian Surveillance System for Communicable Diseases (MSIS). The date corresponds to when the PCR-test was taken.

Usage

```
nor_covid19_cases_by_time_location
```

Format

A csfmt_rts_data_v1 with 11028 rows and 18 variables:

granularity_time day/isoweek

granularity_geo nation, county

country_iso3 nor

location_code norge, 11 counties

border 2020

age total

isoyear Isoyear of event

isoweek Isoweek of event

isoyearweek Isoyearweek of event

season Season of event

seasonweek Seasonweek of event

calyear Calyear of event

calmonth Calmonth of event

calyearmonth Calyearmonth of event

date Date of event

covid19_cases_testdate_n Number of confirmed covid19 cases

covid19_cases_testdate_pr100000 Number of confirmed covid19 cases per 100.000 population

Details

The raw number of cases and cases per 100.000 population are recorded.

This data was extracted on 2022-05-04.

Source

https://github.com/folkehelseinstituttet/surveillance_data/blob/master/covid19/_DOCUMENTATION_data_covid19_msis_by_time_location.txt

---

`Plan` *R6 Class representing a Plan*

Description

We work within the mental model where we have one (or more) datasets and we want to run multiple analyses on these datasets.

By demanding that all analyses use the same data sources we can:

- Be efficient with requiring the minimal amount of data-pulling (this only happens once at the start).
- Better enforce the concept that data-cleaning and analysis should be completely separate.

By demanding that all analysis functions only use two arguments (data and argset) we can:

- Reduce mental fatigue by working within the same mental model for each analysis.
- Make it easier for analyses to be exchanged with each other and iterated on.
- Easily schedule the running of each analysis.

By including all of this in one `Plan` class, we can easily maintain a good overview of all the analyses (i.e. outputs) that need to be run.

Details

An argset is:

- a set of arguments

An analysis is:

- one argset
- one (action) function

A plan is:

- one data pull
- a list of analyses

Public fields

`analyses` List of analyses.

Methods

Public methods:

- `Plan$new()`
- `Plan$add_data()`
- `Plan$add_argset()`
- `Plan$add_argset_from_df()`
- `Plan$add_argset_from_list()`
- `Plan$add_analysis()`
- `Plan$add_analysis_from_df()`
- `Plan$add_analysis_from_list()`

- Plan$apply_action_fn_to_all_argsets()
- Plan$apply_analysis_fn_to_all()
- Plan$x_length()
- Plan$x_seq_along()
- Plan$set_progress()
- Plan$set_progressor()
- Plan$set_verbose()
- Plan$set_use_foreach()
- Plan$get_data()
- Plan$get_analysis()
- Plan$get_argset()
- Plan$get_argsets_as_dt()
- Plan$run_one_with_data()
- Plan$run_one()
- Plan$run_all_with_data()
- Plan$run_all()
- Plan$run_all_progress()
- Plan$run_all_parallel()
- Plan$clone()

Method `new()`**:** Create a new Plan instance.

Usage:

```
Plan$new(verbose = interactive() | config$force_verbose, use_foreach = FALSE)
```

Arguments:

`verbose` Should this plan be verbose?

`use_foreach` ???

Method `add_data()`**:** Add a new data set.

Usage:

```
Plan$add_data(name, fn = NULL, fn_name = NULL, direct = NULL)
```

Arguments:

`name` Name of the data set.

`fn` A function that returns the data set.

`fn_name` A character string containing the name of a function that returns the data set.

`direct` A direct data set.

Examples:

```
p <- plnr::Plan$new()
data_fn <- function(){return(plnr::nor_covid19_cases_by_time_location)}
p$add_data("data_1", fn = data_fn)
p$add_data("data_2", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
p$add_data("data_3", direct = plnr::nor_covid19_cases_by_time_location)
p$get_data()
```

Method `add_argset()`**:** Add a new argset.

Usage:

```
Plan$add_argset(name = uuid::UUIDgenerate(), ...)
```

Arguments:

`name` Name of the (eventual) analysis that this argset will be connected to.

`...` Named arguments that will comprise the argset.

Examples:

```
p <- plnr::Plan$new()
p$add_argset("argset_1", var_1 = 3, var_b = "hello")
p$add_argset("argset_2", var_1 = 8, var_c = "hello2")
p$get_argsets_as_dt()
```

Method `add_argset_from_df()`**:** Add a batch of argsets from a data.frame.

Usage:

```
Plan$add_argset_from_df(df)
```

Arguments:

`df` A data.frame where each row is a new argset, and each column will be a named element in the argset.

Examples:

```
p <- plnr::Plan$new()
batch_argset_df <- data.frame(name = c("a", "b", "c"), var_1 = c(1, 2, 3), var_2 = c("i", "j", "k"))
p$add_argset_from_df(batch_argset_df)
p$get_argsets_as_dt()
```

Method `add_argset_from_list()`**:** Add a batch of argsets from a list.

Usage:

```
Plan$add_argset_from_list(l)
```

Arguments:

`l` A list of lists with named elements where each outermost element is a new argset, and each internal named element named element in the argset.

Examples:

```
p <- plnr::Plan$new()
batch_argset_list <- list(
  list(name = "a", var_1 = 1, var_2 = "i"),
  list(name = "b", var_1 = 2, var_2 = "j"),
  list(name = "c", var_1 = 3, var_2 = "k")
)
p$add_argset_from_list(batch_argset_list)
p$get_argsets_as_dt()
```

Method `add_analysis()`**:** Add a new analysis.

Usage:

```
Plan$add_analysis(name = uuid::UUIDgenerate(), fn = NULL, fn_name = NULL, ...)
```

Arguments:

`name` Name of the analysis.

`fn` Action function.

`fn_name` Action function name.

`...` Named arguments to be added to the argset.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
p$add_analysis(
  name = "analysis_1",
  fn_name = "plnr::example_action_fn"
)
p$get_argsets_as_dt()
p$run_one("analysis_1")
```

Method `add_analysis_from_df()`**:** Add a batch of analyses from a data.frame.

Usage:

```
Plan$add_analysis_from_df(fn = NULL, fn_name = NULL, df)
```

Arguments:

`fn` Action function.

`fn_name` Action function name.

`df` A data.frame where each row is a new argset, and each column will be a named element in the argset.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_df <- data.frame(name = c("a", "b", "c"), var_1 = c(1, 2, 3), var_2 = c("i", "j", "k"))
p$add_analysis_from_df(
  fn_name = "plnr::example_action_fn",
  df = batch_argset_df
 )
p$get_argsets_as_dt()
p$run_one(1)
```

Method `add_analysis_from_list()`**:** Add a batch of argsets from a list.

Usage:

```
Plan$add_analysis_from_list(fn = NULL, fn_name = NULL, l)
```

Arguments:

`fn` Action function.

`fn_name` Action function name.

`l` A list of lists with named elements where each outermost element is a new argset, and each internal named element named element in the argset.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argsets_as_dt()
p$run_one("analysis_1")
```

Method `apply_action_fn_to_all_argsets()`**:** Applies an action function to all the argsets

Usage:

```
Plan$apply_action_fn_to_all_argsets(fn = NULL, fn_name = NULL)
```

Arguments:

`fn` Action function.

`fn_name` Action function name. p <- plnr::Plan$new() p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location") batch_argset_list <- list(list(name = "analysis_1", var_1 = 1, var_2 = "i"), list(name = "analysis_2", var_1 = 2, var_2 = "j"), list(name = "analysis_3", var_1 = 3, var_2 = "k")) p$add_argset_from_list(fn_name = "plnr::example_action_fn", l = batch_argset_list) p$get_argsets_as_dt() p$apply_action_fn_to_all_argsets(fn_name = "plnr::example_action_fn") p$run_one("analysis_1")

Method `apply_analysis_fn_to_all()`**:** Deprecated. Use apply_action_fn_to_all_argsets.

Usage:

```
Plan$apply_analysis_fn_to_all(fn = NULL, fn_name = NULL)
```

Arguments:

`fn` Action function.

`fn_name` Action function name.

Method `x_length()`**:** Number of analyses in the plan.

Usage:

```
Plan$x_length()
```

Method `x_seq_along()`**:** Generate a regular sequence from 1 to the length of the analyses in the plan.

Usage:

```
Plan$x_seq_along()
```

Method `set_progress()`**:** Set an internal progress bar

Usage:

```
Plan$set_progress(pb)
```

Arguments:

`pb` Progress bar.

Method `set_progressor()`**:** Set an internal progressor progress bar

Usage:

```
Plan$set_progressor(pb)
```

Arguments:

`pb` progressor progress bar.

Method `set_verbose()`**:** Set verbose flag

Usage:

```
Plan$set_verbose(x)
```

Arguments:

`x` Boolean.

Method `set_use_foreach()`**:** Set use_foreach flag

Usage:

```
Plan$set_use_foreach(x)
```

Arguments:

`x` Boolean.

Method `get_data()`**:** Extracts the data provided via 'add_data' and returns it as a named list.

Usage:

```
Plan$get_data()
```

Returns: Named list, where most elements have been added via `add_data`.
One extra named element is called 'hash'. 'hash' contains the data hashes of particular datasets/variables, as calculated using the 'spookyhash' algorithm via digest::digest. 'hash' contains two named elements:

- current (the hash of the entire named list)
- current_elements (the hash of the named elements within the named list)

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
p$get_data()
```

Method `get_analysis()`**:** Extracts an analysis from the plan.

Usage:

```
Plan$get_analysis(index_analysis)
```

Arguments:

`index_analysis` Either an integer (1:length(analyses)) or a character string representing the name of the analysis.

Returns: An analysis.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_analysis("analysis_1")
```

Method `get_argset()`**:** Extracts an argset from the plan.

Usage:

```
Plan$get_argset(index_analysis)
```

Arguments:

`index_analysis` Either an integer (1:length(analyses)) or a character string representing the name of the analysis.

Returns: An argset

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argset("analysis_1")
```

Method `get_argsets_as_dt()`**:** Gets all argsets and presents them as a data.table.

Usage:

```
Plan$get_argsets_as_dt()
```

Returns: Data.table that contains all the argsets within a plan.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
```

```
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argsets_as_dt()
```

Method `run_one_with_data()`**:** Run one analysis (data is provided by user).

Usage:

```
Plan$run_one_with_data(index_analysis, data, ...)
```

Arguments:

`index_analysis` Either an integer (1:length(analyses)) or a character string representing the name of the analysis.

`data` Named list (generally obtained from p$get_data()).

`...` Not used.

Returns: Returned value from the action function.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
data <- p$get_data()
p$run_one_with_data("analysis_1", data)
```

Method `run_one()`**:** Run one analysis (data is obtained automatically from self$get_data()).

Usage:

```
Plan$run_one(index_analysis, ...)
```

Arguments:

`index_analysis` Either an integer (1:length(analyses)) or a character string representing the name of the analysis.

`...` Not used.

Returns: Returned value from the action function.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
```

```
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$run_one("analysis_1")
```

Method `run_all_with_data()`**:** Run all analyses (data is provided by user).

Usage:

```
Plan$run_all_with_data(data, ...)
```

Arguments:

`data` Named list (generally obtained from p$get_data()).

`...` Not used.

Returns: List where each element contains the returned value from the action function.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
data <- p$get_data()
p$run_all_with_data(data)
```

Method `run_all()`**:** Run all analyses (data is obtained automatically from self$get_data()).

Usage:

```
Plan$run_all(...)
```

Arguments:

`...` Not used.

Returns: List where each element contains the returned value from the action function.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
```

```
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$run_all()
```

Method `run_all_progress()`**:** Run all analyses with a progress bar (data is obtained automatically from self$get_data()).

Usage:

```
Plan$run_all_progress(...)
```

Arguments:

`...` Not used.

Returns: List where each element contains the returned value from the action function.

Examples:

```
p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$run_all_progress()
```

Method `run_all_parallel()`**:** Run all analyses in parallel (data is obtained automatically from self$get_data()).

This function only works on linux computers and uses `pbmcapply` as the parallel backend.

Usage:

```
Plan$run_all_parallel(mc.cores = getOption("mc.cores", 2L), ...)
```

Arguments:

`mc.cores` Number of cores to be used.

`...` Not used.

Returns: List where each element contains the returned value from the action function.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
Plan$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## ------------------------------------------------
## Method `Plan$add_data`
## ------------------------------------------------

p <- plnr::Plan$new()
data_fn <- function(){return(plnr::nor_covid19_cases_by_time_location)}
p$add_data("data_1", fn = data_fn)
p$add_data("data_2", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
p$add_data("data_3", direct = plnr::nor_covid19_cases_by_time_location)
p$get_data()

## ------------------------------------------------
## Method `Plan$add_argset`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_argset("argset_1", var_1 = 3, var_b = "hello")
p$add_argset("argset_2", var_1 = 8, var_c = "hello2")
p$get_argsets_as_dt()

## ------------------------------------------------
## Method `Plan$add_argset_from_df`
## ------------------------------------------------

p <- plnr::Plan$new()
batch_argset_df <- data.frame(name = c("a", "b", "c"), var_1 = c(1, 2, 3), var_2 = c("i", "j", "k"))
p$add_argset_from_df(batch_argset_df)
p$get_argsets_as_dt()

## ------------------------------------------------
## Method `Plan$add_argset_from_list`
## ------------------------------------------------

p <- plnr::Plan$new()
batch_argset_list <- list(
  list(name = "a", var_1 = 1, var_2 = "i"),
  list(name = "b", var_1 = 2, var_2 = "j"),
  list(name = "c", var_1 = 3, var_2 = "k")
)
p$add_argset_from_list(batch_argset_list)
p$get_argsets_as_dt()

## ------------------------------------------------
## Method `Plan$add_analysis`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
p$add_analysis(
  name = "analysis_1",
  fn_name = "plnr::example_action_fn"
```

```
)
p$get_argsets_as_dt()
p$run_one("analysis_1")

## ------------------------------------------------
## Method `Plan$add_analysis_from_df`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_df <- data.frame(name = c("a", "b", "c"), var_1 = c(1, 2, 3), var_2 = c("i", "j", "k"))
p$add_analysis_from_df(
  fn_name = "plnr::example_action_fn",
  df = batch_argset_df
 )
p$get_argsets_as_dt()
p$run_one(1)

## ------------------------------------------------
## Method `Plan$add_analysis_from_list`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argsets_as_dt()
p$run_one("analysis_1")

## ------------------------------------------------
## Method `Plan$get_data`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
p$get_data()

## ------------------------------------------------
## Method `Plan$get_analysis`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
```

```
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_analysis("analysis_1")

## ------------------------------------------------
## Method `Plan$get_argset`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argset("analysis_1")

## ------------------------------------------------
## Method `Plan$get_argsets_as_dt`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$get_argsets_as_dt()

## ------------------------------------------------
## Method `Plan$run_one_with_data`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
```

```
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
data <- p$get_data()
p$run_one_with_data("analysis_1", data)

## ------------------------------------------------
## Method `Plan$run_one`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$run_one("analysis_1")

## ------------------------------------------------
## Method `Plan$run_all_with_data`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
data <- p$get_data()
p$run_all_with_data(data)

## ------------------------------------------------
## Method `Plan$run_all`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
```

```
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$run_all()

## ------------------------------------------------
## Method `Plan$run_all_progress`
## ------------------------------------------------

p <- plnr::Plan$new()
p$add_data("covid_data", fn_name = "plnr::example_data_fn_nor_covid19_cases_by_time_location")
batch_argset_list <- list(
  list(name = "analysis_1", var_1 = 1, var_2 = "i"),
  list(name = "analysis_2", var_1 = 2, var_2 = "j"),
  list(name = "analysis_3", var_1 = 3, var_2 = "k")
)
p$add_analysis_from_list(
  fn_name = "plnr::example_action_fn",
  l = batch_argset_list
)
p$run_all_progress()
```

`set_opts` *set_opts*

Description

set_opts

Usage

```
set_opts(force_verbose = FALSE)
```

Arguments

`force_verbose`	Force verbose

`test_action_fn` *A test action_fn for an analysis that returns the value 1*

Description

A test action_fn for an analysis that returns the value 1

Usage

```
test_action_fn(data, argset)
```

Arguments

data	Data
argset	argset

`try_again` *Try a code snipped multiple times.*

Description

Adapted from function try_again from package testthat.

Usage

```
try_again(
  x,
  times = 2,
  delay_seconds_min = 5,
  delay_seconds_max = 10,
  verbose = FALSE
)
```

Arguments

x	code
times	Number of times to try
delay_seconds_min	Number of seconds to delay on failure
delay_seconds_max	Number of seconds to delay on failure
verbose	Boolean. Do you want information?

Index