Package 'ggfigdone'

September 10, 2024

Title Manage & Modify 'ggplot' Figures using 'ggfigdone'

Version 0.1.2

Description
When you prepare a presentation or a report, you often need to manage a large number of 'ggplot' figures. You need to change the figure size, modify the title, label, themes, etc. It is inconvenient to go back to the original code to make these changes. This package provides a simple way to manage 'ggplot' figures. You can easily add the figure to the database and update them later using CLI (command line interface) or GUI (graphical user interface).

License GPL-3

Encoding UTF-8

Depends R (>= 4.0.0)

Imports ggplot2, httpuv, httr, readr, jsonlite, uuid, filelock, data.table

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), knitr

BugReports https://github.com/wenjie1991/ggfigdone/issues

NeedsCompilation no

Author Wenjie SUN [aut, cre] (<https://orcid.org/0000-0002-3100-2346>)

Maintainer Wenjie SUN <sunwjie@gmail.com>

Repository CRAN

Date/Publication 2024-09-10 10:00:10 UTC

Contents

`fd_add` *Add a ggplot object to the ggfigdone database*

Description

This function adds a ggplot object to the ggfigdone database. It can also be utilized to update an existing figure using its figure ID.

Usage

```
fd_add(
  name,
  g = last_plot(),
  fdObj = fd_get_db(),
  width = 10,
  height = 10,
  units = "cm",
  dpi = 200,
  overwrite = FALSE,
  id = uuid::UUIDgenerate()
)
```

Arguments

`name`	A character string representing the figure name.
`g`	A ggplot object.
`fdObj`	An object of class `fdObj`.
`width`	A numeric value specifying the width of the canvas.
`height`	A numeric value specifying the height of the canvas.
`units`	A character string indicating the units of the canvas.
`dpi`	A numeric value denoting the dpi of the canvas.
`overwrite`	A logical value. If set to TRUE, the function will overwrite the figure if it already exists. If set to FALSE, the function will terminate with an error message.
`id`	A character string representing the figure ID. If not provided, the function will generate a random ID. Alternatively, an existing ID can be provided to update the corresponding figure.

Value

An object of class `fdObj`.

Examples

```
library(ggplot2)

## Initial ggfigdone database using `fd_init`
db_dir = file.path(tempdir(), "fd_add_exp")
fo = fd_init(db_dir, rm_exist = TRUE)

## Draw a ggplot figure
g = ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()

## Add the figure to the database
fd_add(g = g, name  = "fig1", fo)

## Add the same figure with a different name
fd_add(g = g, name  = "fig2", fo)

## Show the updated ggfigdone database
print(fo)
```

`fd_canvas` *Update the figure canvas size*

Description

This function is designed to update the size of the figure canvas.

Usage

```
fd_canvas(
  id,
  fdObj = fd_get_db(),
  width = fdObj$env[[id]]$canvas_options$width,
  height = fdObj$env[[id]]$canvas_options$height,
  units = fdObj$env[[id]]$canvas_options$units,
  dpi = fdObj$env[[id]]$canvas_options$dpi
)
```

Arguments

`id`	A character string representing the figure ID.
`fdObj`	An object of class `fdObj`.
`width`	A numeric value specifying the width of the canvas.

height	A numeric value specifying the height of the canvas.
units	A character string indicating the units of measurement for the canvas, such as "cm", "in", "mm", or "px".
dpi	A numeric value denoting the dots per inch (DPI) of the canvas.

Value

No return value, changes are made directly to the ggfigdone database.

`fd_get_db` *Get the default ggfigdone database*

Description

Get the default ggfigdone database

Usage

```
fd_get_db()
```

Value

An object of class `fdObj` representing the default ggfigdone database.

`fd_init` *Initiates the ggfigdone database*

Description

This function generates a folder that serves as a database for ggfigdone.

Usage

```
fd_init(dir, recursive = TRUE, rm_exist = FALSE, set_default = TRUE, ...)
```

Arguments

dir	A character string specifying the directory path.
recursive	A logical value. If TRUE, the function will create the directory along with any necessary parent directories if they do not already exist. If FALSE, the function will create the directory only if its parent directory already exists.
rm_exist	A logical value. If TRUE, the function will remove the content in the directory if it already exists. If FALSE, the function will ask the user whether to remove the content in the directory.
set_default	A logical value. If TRUE, the function will set the database as the default database.
...	Additional arguments to be passed to `fd_load` function.

Value

An object of class `fdObj`.

Examples

```
library(ggplot2)
## create ggfigdone database in a temporary directory
db_dir = file.path(tempdir(), "fd_init")

## Initate the ggfigdone database
fd_init(db_dir, rm_exist = TRUE)
```

`fd_load` *Load the ggfigdone database*

Description

This function loads the ggfigdone database from the disk.

Usage

```
fd_load(dir, auto_database_upgrade = TRUE, set_default = TRUE)
```

Arguments

`dir`	A character string representing the directory path.
`auto_database_upgrade`	A logical value. If TRUE, the function will automatically upgrade the database to the latest version. If FALSE, you need to manually save the data using the `fd_save` function.
`set_default`	A logical value. If TRUE, the function will set the database as the default database.

Value

An object of class `fdObj`.

Examples

```
library(ggplot2)
## create ggfigdone database in a temporary directory
db_dir = file.path(tempdir(), "fd_load")
fd_init(db_dir, rm_exist = TRUE)

## Load the ggfigdone database
fd_load(db_dir)
```

`fd_ls` *List the figures*

Description

This function provides a List or data.frame of figures along with their associated parameters.

Usage

```
fd_ls(fdObj = fd_get_db())

fd_df(fdObj = fd_get_db())
```

Arguments

`fdObj`	An instance of the `fdObj` class.

Details

The parameters include:

- id: The unique identifier for the figure
- name: The name of the figure
- created_date: The date the figure was created
- updated_date: The date the figure was last updated
- width: The width of the canvas
- height: The height of the canvas
- units: The units of measurement for the canvas
- dpi: The dots per inch (DPI) of the canvas
- file_name: The name of the file
- plot_labels: The labels used in the plot

Value

A List/data.frame containing the figures along with their respective parameters.

fd_merge *Merge ggfigdone databases*

Description

This function merges two ggfigdone databases. The function will update the figures in the 'to' database with the figures in the 'from' database. If there is a figure with the same ID in both databases, the function will keep the figure with the latest updated date or created date.

Usage

```
fd_merge(from, to = fd_get_db(), replace = "updated_date")
```

Arguments

from	An object of class fdObj that will be merged from.
to	An object of class fdObj that will be merged to. The default value is the default ggfigdone database.
replace	A character string specifying the method to keep the figure with the unique ID. It can be either "updated_date" or "created_date".

Value

An object of class fdObj with the merged database.

Examples

```
library(ggplot2)
## create ggfigdone database in a temporary directory
db_dir1 = file.path(tempdir(), "db1")
db_dir2 = file.path(tempdir(), "db2")
fo1 = fd_init(db_dir1, rm_exist = TRUE)
fo2 = fd_init(db_dir2, rm_exist = TRUE)

## Draw a ggplot figure
g = ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()

## Add the figure to the database
fd_add(g = g, name  = "fig1", fdObj = fo1)
fd_add(g = g + theme_classic(), name  = "fig2", fdObj = fo2)

## Merge the databases
fo_merge = fd_merge(from = fo1, to = fo2, replace = "updated_date")

## Show the updated ggfigdone database
print(fo_merge)

##
```

fd_rm *Remove a figure*

Description

This function removes a figure from the ggfigdone database.

Usage

```
fd_rm(id, fdObj = fd_get_db())
```

Arguments

id	A character string representing the figure ID.
fdObj	An object of class fdObj.

Value

No return value, changes are made directly to the ggfigdone database.

fd_save *Update the ggfigdone database changes* **to the disk**

Description

This function saves the ggfigdone data to the disk. By default, when using the fd_load funciton to load the databse, the data will be automatically saved to the disk when changes are made. But if you set the auto_database_upgrade argument to FALSE in fd_load, you need to manually save the data using this function.

Usage

```
fd_save(fdObj = fd_get_db(), do_lock = TRUE)
```

Arguments

fdObj	An object of class fdObj.
do_lock	A logical value. If TRUE, the function will lock the database file when saving the data.

Value

No return value, changes are made directly to the ggfigdone database.

`fd_server` *Initiates a server for ggfigdone*

Description

This function initiates a server for ggfigdone, which can be accessed through a web browser. The web application enables users to manage and modify ggplot figures with ease. Users have the ability to:

- Update the ggplot code by adding new components.
- Adjust the figure size.
- Download the figure as a PDF.
- Download the data used to create the figure.

Usage

```
fd_server(dir, host = "0.0.0.0", port = 8080, auto_open = TRUE)
```

Arguments

`dir`	The directory of the ggfigdone database.
`host`	The host on which the server will run; the default is '0.0.0.0'.
`port`	The port on which the server will run; the default is 8080.
`auto_open`	A logical value indicating whether the server should be opened in a web browser; the default is TRUE.

Details

By default the function will open a web browser to access the server.

You can configure the web browser by setting the options:

```
options(browser = "firefox")  # Set Firefox as the default
```

Value

No return value, the function is called for its side effects.

fd_set_db *Set the default ggfigdone database*

Description

Set the default ggfigdone database

Usage

```
fd_set_db(fdObj)
```

Arguments

fdObj	An object of class fdObj to be set as the default ggfigdone database.

Value

No return value, the default ggfigdone database is set to an environment variable.

fd_unique *Keep figure name unique by removing older figures with the same name*

Description

This function keeps the figure name unique by removing the older figures with the same name. Users can specify whether to keep the figure with the latest updated date or the latest created date. If a figure is created without changing, the created date and updated date are the same.

Usage

```
fd_unique(fdObj = fd_get_db(), by = "updated_date")
```

Arguments

fdObj	An object of class fdObj.
by	A character string specifying the method to keep the figure with the unique name. It can be either "updated_date" or "created_date".

Value

An object of class fdObj.

Examples

```
library(ggplot2)
## create ggfigdone database in a temporary directory
db_dir = file.path(tempdir(), "fd_unique")
fo = fd_init(db_dir, rm_exist = TRUE)

## Draw a ggplot figure
g = ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()

## Add the figure to the database
fd_add(g = g, name  = "fig1", fdObj = fo)

## Add the another figure with the same name
fd_add(g = g + theme_classic(), name  = "fig1", fdObj = fo)

## Keep the figure with the latest created date
fd_unique(fdObj = fo, by = "created_date")

## Show the updated ggfigdone database
print(fo)
```

---

fd_update_fig *Update a figure using ggplot expression*

Description

This function updates a figure using a ggplot expression.

Usage

```
fd_update_fig(id, expr, fdObj = fd_get_db())
```

Arguments

id	A character string of the figure id
expr	A character string of the ggplot expression
fdObj	An object of class fdObj

Value

A character string of the status

ggfigdone *ggfigdone*

Description

ggfigdone: Manage & Modify ggplot figures easily

Details

When preparing a presentation or report, it is often necessary to manage a substantial number of ggplot figures. Adjustments such as changing the figure size, modifying titles, labels, and themes may be required. Returning to the original code to implement these changes can be inconvenient. This package offers a straightforward method for managing ggplot figures. Figures can be easily added to the database and subsequently updated using either a GUI (graphical user interface) and/or CLI (command line interface).

Author(s)

Maintainer: Wenjie SUN <sunwjie@gmail.com> (ORCID)

See Also

Useful links:

- Report bugs at https://github.com/wenjie1991/ggfigdone/issues

Index