

Package ‘dataversionr’

August 18, 2022

Title Time Versioned Storage of Data Frames

Version 0.9.0

Description Create, update, read and delete data frames in time versioned, parquet-backed datasets. Inspect how your data frame has changed over time. Run your analysis against multiple versions of your data frame.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Imports arrow (>= 8.0.0), dplyr, lubridate, purrr, diffr, tidyr, magrittr, rlang (>= 1.0.2)

Suggests testthat (>= 3.0.0), withr, sys, aws.s3, tzd

Config/testthat/edition 3

NeedsCompilation no

Author Riaz Arbi [aut, cre]

Maintainer Riaz Arbi <dataversionr@arbidata.com>

Repository CRAN

Date/Publication 2022-08-18 08:40:05 UTC

R topics documented:

commit_diff	2
create_dv	3
delete_backup	3
destroy_dv	4
fix_path	5
generate_metadata	6
get_backups	7
get_diffs	7
get_diff_stats	8
get_latest	9
get_metadata	9

make_prefix	10
make_SubTreeFileSystem	11
put_backup	11
put_diff_stats	12
put_latest	13
put_metadata	13
read_dv	14
read_dv_backup	15
read_dv_diff	15
remove_prefix	16
summarise_diffs	17
update_dv	17

Index 19

commit_diff	<i>Commit diff</i>
-------------	--------------------

Description

Write a diff to a versioned dataset destination. Check that it was written correctly, otherwise return an error.

Usage

```
commit_diff(diff_df, destination, verbose = FALSE)
```

Arguments

diff_df	a data frame. Output off diffdfs::diffdfs.
destination	a local directory path or an arrow SubTreeFileSystem
verbose	TRUE /FALSE should the function be chatty?

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
new_df <- data.frame(a = 2:5, b = letters[2:5])
diff <- diffdfs::diffdfs(new_df, df)

commit_diff(diff, temp_dir)

unlink(temp_dir)
```

create_dv	<i>Create dv</i>
-----------	------------------

Description

Create a versioned dataset

Usage

```
create_dv(df, destination, key_cols = NA, diffed = TRUE, backup_count = 0L)
```

Arguments

df	a data frame
destination	a local directory path or an arrow SubTreeFileSystem
key_cols	a character vector of column names that constitute a unique key
diffed	should we store diffs between each dv version?
backup_count	how many backups should we store?

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])

create_dv(df, temp_dir)

unlink(temp_dir)
```

delete_backup	<i>Delete backup</i>
---------------	----------------------

Description

Delete backups after a particular position in a versioned dataset

Usage

```
delete_backup(destination, after_position, verbose = FALSE)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem
 after_position how many backups should we leave in the dv?
 verbose TRUE /FALSE should the function be chatty?

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
for(i in 1:10) {put_backup(df, temp_dir)}

# before
list.files(file.path(temp_dir, "backup"))

delete_backup(temp_dir, 4L)

# after
list.files(file.path(temp_dir, "backup"))
```

destroy_dv

Destroy dv

Description

Destroy dv

Usage

```
destroy_dv(destination, prompt = TRUE)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem
 prompt should we ask for manual confirmation?

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])

create_dv(df, temp_dir)
destroy_dv(temp_dir, prompt = FALSE)

unlink(temp_dir)
```

fix_path

Fix path

Description

Take a prefix and a local file path or SubTreeFileSystem and return the correct SubTreeFileSystem. A bit like file.path for arrow SubTreeFileSystems.

Usage

```
fix_path(path, destination, verbose = FALSE)
```

Arguments

path	a sub prefix of the destination
destination	a local directory path or an arrow SubTreeFileSystem
verbose	TRUE /FALSE should the function be chatty?

Value

an arrow SubTreeFileSystem

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)

fix_path("backup", temp_dir)
```

generate_metadata	<i>Generate metadata</i>
-------------------	--------------------------

Description

Generate a conformant metadata structure for a versioned dataset

Usage

```
generate_metadata(  
  df,  
  destination,  
  key_cols = NA,  
  diffed = TRUE,  
  backup_count = 0L  
)
```

Arguments

df	a data frame to create a dv from
destination	a local directory path or an arrow SubTreeFileSystem
key_cols	a character vector of column names that constitute a unique key
diffed	should we store diffs between each dv version?
backup_count	how many backups should we store?

Value

a list

Examples

```
temp_dir <- tempfile()  
dir.create(temp_dir, recursive = TRUE)  
df <- data.frame(a = 1:5, b = letters[1:5])  
  
generate_metadata(df, temp_dir)  
  
unlink(temp_dir)
```

`get_backups`*Get backups*

Description

Read the backups in a versioned dataset to a data frame or an arrow dataset

Usage

```
get_backups(destination, collect = TRUE)
```

Arguments

`destination` a local directory path or an arrow SubTreeFileSystem
`collect` should we collect the underlying arrow dataset or return just the connection?

Value

an arrow dataset

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
put_backup(df, temp_dir)

get_backups(temp_dir)
```

`get_diffs`*Get diffs*

Description

Read in all the diffs in a versioned dataset to a data frame or arrow dataset

Usage

```
get_diffs(destination, collect = TRUE)
```

Arguments

`destination` a local directory path or an arrow SubTreeFileSystem
`collect` should we collect the underlying arrow dataset or return just the connection?

Value

a data frame or an arrow dataset connection

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
new_df <- data.frame(a = 2:5, b = letters[2:5])
diff <- diffdfs::diffdfs(new_df, df)
commit_diff(diff, temp_dir)

get_diffs(temp_dir)

unlink(temp_dir)
```

get_diff_stats	<i>Get diff stats</i>
----------------	-----------------------

Description

Read a versioned dataset's diff statistics to a data frame

Usage

```
get_diff_stats(destination)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem

Value

a data frame

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
new_df <- data.frame(a = 2:5, b = letters[2:5])
diff <- diffdfs::diffdfs(new_df, df)
commit_diff(diff, temp_dir)
put_diff_stats(temp_dir)

get_diff_stats(temp_dir)

unlink(temp_dir)
```

get_latest	<i>Get latest</i>
------------	-------------------

Description

Read in the latest version of a versioned dataset to a data frame

Usage

```
get_latest(destination, collect = TRUE)
```

Arguments

destination	a local directory path or an arrow SubTreeFileSystem
collect	should we return a data frame (TRUE) or an arrow dataset connection (FALSE)

Value

a data frame or an arrow dataset

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
put_latest(df, temp_dir)

get_latest(temp_dir)

unlink(temp_dir)
```

get_metadata	<i>Get metadata</i>
--------------	---------------------

Description

Read versioned dataset metadata to a list

Usage

```
get_metadata(destination, verbose = FALSE)
```

Arguments

destination	a local directory path or an arrow SubTreeFileSystem
verbose	TRUE /FALSE should the function be chatty?

Value

a list

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
md <- generate_metadata(df, temp_dir)
put_metadata(md, temp_dir)

get_metadata(temp_dir)

unlink(temp_dir)
```

make_prefix

Make prefix

Description

Create the necessary subdirectories to make sure that SubTreeFileSystem methods will work.

Usage

```
make_prefix(destination)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem

Value

silent

Examples

```
temp_dir <- tempfile()
make_prefix(temp_dir)
```

make_SubTreeFileSystem
Make SubTreeFileSystem

Description

Turn a local file path string or a SubTreeFileSystem into a SubTreeFileSystem

Usage

```
make_SubTreeFileSystem(destination, verbose = FALSE)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem
verbose TRUE /FALSE should the function be chatty?

Value

an arrow SubTreeFileSystem

Examples

```
temp_dir <- tempfile()  
make_SubTreeFileSystem(temp_dir)
```

put_backup *Put backup*

Description

Write a data frame to the backup section of a versioned dataset

Usage

```
put_backup(new_df, destination)
```

Arguments

new_df a data frame
destination a local directory path or an arrow SubTreeFileSystem

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])

put_backup(df, temp_dir)
```

put_diff_stats	<i>Put diff stats</i>
----------------	-----------------------

Description

Write diff statistics to a versioned dataset

Usage

```
put_diff_stats(destination)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
new_df <- data.frame(a = 2:5, b = letters[2:5])
diff <- diffdfs::diffdfs(new_df, df)
commit_diff(diff, temp_dir)

put_diff_stats(temp_dir)

unlink(temp_dir)
```

put_latest	<i>Put latest</i>
------------	-------------------

Description

Write a new latest version to a versioned dataset. Does not update diffs or backups.

Usage

```
put_latest(new_df, destination)
```

Arguments

new_df	a dataframe
destination	a local directory path or an arrow SubTreeFileSystem

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])

put_latest(df, temp_dir)

unlink(temp_dir)
```

put_metadata	<i>Put metadata</i>
--------------	---------------------

Description

Write versioned dataset metadata to a destination.

Usage

```
put_metadata(metadata, destination, verbose = FALSE)
```

Arguments

metadata	a metadata list generated by generate_metadata
destination	a local directory path or an arrow SubTreeFileSystem
verbose	TRUE /FALSE should the function be chatty?

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
md <- generate_metadata(df, temp_dir)

put_metadata(md, temp_dir)
unlink(temp_dir)
```

read_dv

*Read dv***Description**

Read a version of a versioned dataset into a data frame

Usage

```
read_dv(destination, as_of = NA, source = "latest")
```

Arguments

destination	a local directory path or an arrow SubTreeFileSystem
as_of	the valid date at which you'd like to read the dv
source	the source of the dv. Options are 'latest', 'diffs' or 'backup'

Value

a data frame

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
create_dv(df, temp_dir, backup_count = 5L)

read_dv(temp_dir)

read_dv(temp_dir, source = "backup")

read_dv(temp_dir, as_of = lubridate::now(), source = "diffs")

unlink(temp_dir)
```

read_dv_backup	<i>Read dv backup</i>
----------------	-----------------------

Description

Read a version of a versioned dataset into a data frame using just the stored backups

Usage

```
read_dv_backup(destination, as_of)
```

Arguments

destination	a local directory path or an arrow SubTreeFileSystem
as_of	the valid date at which you'd like to read the dv

Value

a data frame

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
create_dv(df, temp_dir, backup_count = 5L)

read_dv_backup(temp_dir, as_of = lubridate::now())

unlink(temp_dir)
```

read_dv_diff	<i>Read dv diff</i>
--------------	---------------------

Description

Read a version of a versioned dataset into a data frame using just the stored diffs.

Usage

```
read_dv_diff(destination, as_of, key_cols)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem
 as_of the valid date at which you'd like to read the dv
 key_cols a character vector of column names that constitute a unique key

Value

a data frame

Examples

```

temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
create_dv(df, temp_dir, diffed = TRUE)

read_dv_diff(temp_dir,
             as_of = lubridate::now(),
             key_cols = get_metadata(temp_dir)$key_cols)

unlink(temp_dir)

```

remove_prefix

Remove prefix

Description

Remove all the contents of a subdirectory or SubTreeFileSystem

Usage

```
remove_prefix(destination, prompt = TRUE)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem
 prompt should we ask for user confirmation?

Value

TRUE

Examples

```

temp_dir <- tempfile()
make_prefix(temp_dir)

remove_prefix(temp_dir, prompt = FALSE)

```

summarise_diffs	<i>Summarise diffs</i>
-----------------	------------------------

Description

Create a data frame which summarises the diffs in a versioned dataset

Usage

```
summarise_diffs(destination)
```

Arguments

destination a local directory path or an arrow SubTreeFileSystem

Value

a data frame of diff statistics

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
new_df <- data.frame(a = 2:5, b = letters[2:5])
diff <- diffdfs::diffdfs(new_df, df)
commit_diff(diff, temp_dir)

summarise_diffs(temp_dir)

unlink(temp_dir)
```

update_dv	<i>Update dv</i>
-----------	------------------

Description

Update a versioned dataset with a new version of the data.

Usage

```
update_dv(df, destination)
```

Arguments

df a data frame. Must be structurally identical to the dv.
destination a local directory path or an arrow SubTreeFileSystem

Value

TRUE

Examples

```
temp_dir <- tempfile()
dir.create(temp_dir, recursive = TRUE)
df <- data.frame(a = 1:5, b = letters[1:5])
new_df <- data.frame(a = 2:5, b = letters[2:5])
create_dv(df, temp_dir)

update_dv(new_df, temp_dir)
```

Index

[commit_diff](#), 2
[create_dv](#), 3

[delete_backup](#), 3
[destroy_dv](#), 4

[fix_path](#), 5

[generate_metadata](#), 6
[get_backups](#), 7
[get_diff_stats](#), 8
[get_diffs](#), 7
[get_latest](#), 9
[get_metadata](#), 9

[make_prefix](#), 10
[make_SubTreeFileSystem](#), 11

[put_backup](#), 11
[put_diff_stats](#), 12
[put_latest](#), 13
[put_metadata](#), 13

[read_dv](#), 14
[read_dv_backup](#), 15
[read_dv_diff](#), 15
[remove_prefix](#), 16

[summarise_diffs](#), 17

[update_dv](#), 17