

Package ‘rollup’

July 23, 2025

Title A Tidy Grouping Set Aggregation

Version 0.1.0

Description A Tidy implementation of 'grouping sets', 'rollup' and 'cube' - extensions of the 'group_by' clause that allow for computing multiple 'group_by' clauses in a single statement. For more detailed information on these functions, please refer to ``Enhanced Aggregation, Cube, Grouping and Rollup" <<https://cwiki.apache.org/confluence/display/Hive/Enhanced+Aggregation%2C+Cube%2C+Grouping+and+Rollup>>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.0

Depends R (>= 2.10), dplyr, tidyr

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports rlang, methods, utils, magrittr, sparklyr

URL <https://juyoungahn.github.io/rollup/>

NeedsCompilation no

Author Ju Young Ahn [aut, cre] (ORCID:
<<https://orcid.org/0009-0008-4613-3438>>)

Maintainer Ju Young Ahn <juyoung.ahn@snu.ac.kr>

Repository CRAN

Date/Publication 2024-08-29 11:10:04 UTC

Contents

grouped_df_list-class	2
grouping_sets	2
summarise	3
summarise,ANY-method	3
summarise,grouped_df_list-method	4

summarize	4
summarize,ANY-method	5
summarize,grouped_df_list-method	5
summarize_rollup	6
web_service_data	6
with_cube	7
with_rollup	7
Index	9

grouped_df_list-class	<i>grouped_df_list class definition</i>
-----------------------	---

Description

A class to represent a list of grouped data frames.

grouping_sets	<i>grouping_sets</i>
---------------	----------------------

Description

Compute total amounts at different group levels, producing multiple subtotals. With the 'grouping_sets' clause following 'group_by', you can aggregate multiple grouping variables in one operation. This reflects the 'GROUPING SETS' operations in 'SQL'.

Usage

```
grouping_sets(df, ...)
```

Arguments

- | | |
|-----|-------------------------|
| df | dataframe or grouped df |
| ... | grouping variables |

Value

A list of 'grouped_df' class. each 'grouped_df' object has a different grouping level.

Examples

```
mtcars %>% group_by(vs, am) %>% grouping_sets("vs","am",c("vs","am"))
mtcars %>% group_by(vs, am) %>% with_rollup()
mtcars %>% group_by(vs, am) %>% with_cube()
```

summarise	<i>Generic summarise function</i>
-----------	-----------------------------------

Description

Generic summarise function

Usage

```
summarise(object, ...)
```

Arguments

object	Object to be summarized.
...	Additional arguments.

Value

An object of the same class as `.data`. One grouping level will be dropped.

summarise, ANY-method	<i>Default method for summarise</i>
-----------------------	-------------------------------------

Description

Default method for summarise

Usage

```
## S4 method for signature 'ANY'
summarise(object, ...)
```

Arguments

object	An object
...	Additional arguments.

Value

An object of the same class as `.data`. One grouping level will be dropped.

summarise, grouped_df_list-method

Method for summarise on grouped_df_list

Description

Method for summarise on grouped_df_list

Usage

```
## S4 method for signature 'grouped_df_list'  
summarise(object, ...)
```

Arguments

object	A grouped_df_list object.
...	Additional arguments.

Value

An object of the same class as .data. One grouping level will be dropped.

summarize

Generic summarize function

Description

Generic summarize function

Usage

```
summarize(object, ...)
```

Arguments

object	Object to be summarized.
...	Additional arguments.

Value

An object of the same class as .data. One grouping level will be dropped.

summarize,ANY-method *Default method for summarize*

Description

Default method for summarize

Usage

```
## S4 method for signature 'ANY'  
summarize(object, ...)
```

Arguments

object	An object.
...	Additional arguments.

Value

An object of the same class as `.data`. One grouping level will be dropped.

summarize,grouped_df_list-method
Method for summarize on grouped_df_list

Description

Method for summarize on grouped_df_list

Usage

```
## S4 method for signature 'grouped_df_list'  
summarize(object, ...)
```

Arguments

object	A grouped_df_list object.
...	Additional arguments.

Value

An object of the same class as `.data`. One grouping level will be dropped.

summarize_rollup	<i>summarize_rollup</i>
------------------	-------------------------

Description

'summarize_rollup' aggregates each 'grouped_df' in the 'grouped_df_list' class and return the unioned aggregated results.

Usage

```
summarize_rollup(df_list, ...)
```

Arguments

df_list	'grouped_df_list' class
...	functions for 'summarize'

Value

An object of the same class as .data. The unioned aggregated result of multiple grouping levels will be dropped.

web_service_data	<i>Web Service Data</i>
------------------	-------------------------

Description

A dataset containing information about various web services.

Usage

```
web_service_data
```

Format

A data frame with 30,000 rows and 6 variables:

date_id date id
id user id
gender gender
age age band
page_view_cnt pageview count
product_view_cnt_cat product view count (category)

Source

Generated for example purposes

`with_cube`*with_cube*

Description

Compute total amounts at different group levels, producing multiple subtotals. With the 'with_cube' clause following 'group_by', you can aggregate multiple grouping variables in one operation. This reflects the 'WITH CUBE' operations in 'SQL'.

Usage

```
with_cube(grouped_df)
```

Arguments

```
grouped_df      'grouped_df' class
```

Value

A list of 'grouped_df' class. each 'grouped_df' object has a different grouping level.

Examples

```
mtcars %>% group_by(vs, am) %>% grouping_sets("vs", "am", c("vs", "am"))
mtcars %>% group_by(vs, am) %>% with_rollup()
mtcars %>% group_by(vs, am) %>% with_cube()
```

`with_rollup`*with_rollup*

Description

Compute total amounts at different group levels, producing multiple subtotals. With the 'with_rollup' clause following 'group_by', you can aggregate multiple grouping variables in one operation. This reflects the 'WITH ROLLUP' operations in 'SQL'.

Usage

```
with_rollup(grouped_df)
```

Arguments

```
grouped_df      'grouped_df' class
```

Value

A list of 'grouped_df' class. each 'grouped_df' object has a different grouping level.

Examples

```
mtcars %>% group_by(vs, am) %>% grouping_sets("vs", "am", c("vs", "am"))
mtcars %>% group_by(vs, am) %>% with_rollup()
mtcars %>% group_by(vs, am) %>% with_cube()
```

Index

* **datasets**

web_service_data, [6](#)

grouped_df_list-class, [2](#)

grouping_sets, [2](#)

summarise, [3](#)

summarise, ANY-method, [3](#)

summarise, grouped_df_list-method, [4](#)

summarize, [4](#)

summarize, ANY-method, [5](#)

summarize, grouped_df_list-method, [5](#)

summarize_rollup, [6](#)

web_service_data, [6](#)

with_cube, [7](#)

with_rollup, [7](#)