

# Package ‘rollup’

May 9, 2026

**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://wiki.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

<b>Index</b>	<b>9</b>
--------------	----------

---

grouped\_df\_list-class *grouped\_df\_list class definition*

---

## 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:

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

**Source**

Generated for example purposes

---

with_cube	<i>with_cube</i>
-----------	------------------

---

**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	<i>with_rollup</i>
-------------	--------------------

---

**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](#)