

# Package ‘tidyrgee’

October 14, 2022

**Title** 'tidyverse' Methods for 'Earth Engine'

**Version** 0.1.0

**Description** Provides 'tidyverse' methods for wrangling and analyzing 'Earth Engine' <<https://earthengine.google.com/>> data. These methods help the user with filtering, joining and summarising 'Earth Engine' image collections.

**License** MIT + file LICENSE

**URL** <https://github.com/r-tidy-remote-sensing/tidyrgee>

**BugReports** <https://github.com/r-tidy-remote-sensing/tidyrgee/issues/>

**Depends** R (>= 4.1)

**Imports** assertthat, crayon, dplyr, glue, lubridate, purrr, readr, reticulate (>= 1.24), rgee, rlang, sf, stringr, tidyr

**Suggests** knitr, rmarkdown, tibble, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Zack Arno [aut, cre, cph],  
Josh Erickson [aut, cph]

**Maintainer** Zack Arno <zackarno@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-09-15 08:00:02 UTC

## R topics documented:

add_date_to_bandname . . . . .	2
as_ee . . . . .	3
as_tidyee . . . . .	3

bgd_msna . . . . .	4
bind_ics . . . . .	5
clip . . . . .	6
create_tidyee . . . . .	7
ee_composite . . . . .	8
ee_extract_tidy . . . . .	8
ee_month_composite . . . . .	10
ee_month_filter . . . . .	11
ee_year_composite . . . . .	11
ee_year_filter . . . . .	12
ee_year_month_composite . . . . .	12
ee_year_month_filter . . . . .	13
filter . . . . .	14
filter_bounds . . . . .	15
group_by . . . . .	16
group_split . . . . .	17
inner_join . . . . .	18
mutate . . . . .	18
print.tidyee . . . . .	19
select . . . . .	19
set_idx . . . . .	20
slice . . . . .	21
summarise . . . . .	22
ungroup . . . . .	23
<b>Index</b>	<b>24</b>

---

add\_date\_to\_bandname    *add\_date\_to\_band\_name*

---

## Description

append date to band name

## Usage

add\_date\_to\_bandname(x)

## Arguments

x                    ee\$ImageCollection or ee\$Image

## Value

a date to band name in x.

---

as_ee	<i>as_ee tidyee to ee\$ImageCollection or ee\$Image</i>
-------	---------------------------------------------------------

---

**Description**

as\_ee tidyee to ee\$ImageCollection or ee\$Image

**Usage**

```
as_ee(x)
```

**Arguments**

x                    tidyee

**Value**

ee\$ImageCollection or ee\$Image

**Examples**

```
## Not run:
library(rgee)
librar(tidyee)

modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")

# create tidyee class
modis_ic_tidy <- as_tidyee(modis_ic)
# convert back to origina ee$ImageCollection class
modis_ic_tidy |>
  as_ee()

## End(Not run)
```

---

as_tidyee	<i>as_tidy_ee</i>
-----------	-------------------

---

**Description**

The function returns a list containing the original object (Image/ImageCollection) as well as a "virtual data.frame (vrt)" which is a data.frame holding key properties of the ee\$Image/ee\$ImageCollection. The returned list has been assigned a new class "tidyee".

**Usage**

```
as_tidyee(x, time_end = FALSE)
```

**Arguments**

`x` ee\$Image or ee\$ImageCollection  
`time_end` logical include time\_end ("system:time\_end") in vrt (default=F)

**Value**

tidyee class object which contains a list with two components ("x","vrt")

**Examples**

```
## Not run:
library(tidyrgree)
library(rgee)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic_tidy <- as_tidyee(modis_ic)

## End(Not run)
```

---

 bgd\_msna

*A subset of question responses from the 2019 Host Community MSNA in Bangladesh*

---

**Description**

Data frame of responses with anonymized coordinates

**Usage**

```
bgd_msna
```

**Format**

A data frame with 1374 rows and 15 variables:

**\_uuid** unique identifier

**informed\_consent** informed consent

**survey\_date** date of survey

**end\_survey** date of end of survey

**electricity\_grid** question about electricity grid

**solar\_light** question about solar light

**illness\_HH\_count** repeat group calculation on # hh members with illness in past x days

**cooking\_fuel/collected\_firewood** select multiple response - did HH collect firewood for cooking fuel

**income\_source/agricultural\_production\_sale** income source question - ariculture  
**agricultural\_land** question on agricultural land  
**employment\_source/agricultural\_casual** employment source - ag  
**employment\_source/non\_agricultural\_casual** employment source - non-ag  
**employment\_source/fishing** employment source - fishing  
**\_gps\_reading\_longitude** longitude - jittered/anonymized  
**\_gps\_reading\_latitude** latitude - jittered/anonymized ...

**Value**

data frame

---

bind_ics	<i>bind ImageCollections</i>
----------	------------------------------

---

**Description**

bind ImageCollections

**Usage**

```
bind_ics(x)
```

**Arguments**

x                    list of tidyee objects

**Value**

tidyee object containing single image collection and vrt

**Examples**

```
## Not run:
library(tidyrgree)
library(rgee)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic_tidy <- as_tidyee(modis_ic)
modis_tidy_list <- modis_tidy |>
  group_split(month)
modis_tidy_list |>
  bind_ics()

## End(Not run)
```

---

clip	<i>clip flexible wrapper for rgee::ee\$Image\$clip()</i>
------	----------------------------------------------------------

---

### Description

allows clipping of tidyee, ee\$Imagecollection, or ee\$Image classes. Also allows objects to be clipped to sf object in addition to ee\$FeatureCollections/ee\$Feature

### Usage

```
clip(x, y, return_tidyee = TRUE)
```

### Arguments

x	object to be clipped (tidyee, ee\$ImageCollection, ee\$Image)
y	geometry object to clip to (sf, ee\$Feature, ee\$FeatureCollections)
return_tidyee	logical return tidyee class (default = TRUE) object or ee\$ImageCollection. Faster performance if F

### Value

x as tidyee or ee\$Image/ee\$ImageCollection depending on return\_tidyee argument.

### Examples

```
## Not run:
library(tidyrgree)
library(tidyverse)
library(rgee)
rgee::ee_initialize()

# create geometry and convert to sf
coord_tibble <- tibble::tribble(
  ~X,          ~Y,
  92.2303683692011, 20.9126490153521,
  92.2311567217866, 20.9127410439304,
  92.2287527311594, 20.9124072954926,
  92.2289221219251, 20.9197352745068,
  92.238724724534, 20.9081803233546
)
sf_ob <- sf::st_as_sf(coord_tibble, coords=c("X", "Y"), crs=4326)

roi <- ee$Geometry$Polygon(list(
  c(-114.275, 45.891),
  c(-108.275, 45.868),
  c(-108.240, 48.868),
  c(-114.240, 48.891)
))
```

```
# load landsat
ls = ee$ImageCollection("LANDSAT/LC08/C01/T1_SR")

# create tidyee class
ls_tidy <- as_tidyee(ls)

# filter_bounds on sf object
# return tidyee object
ls_tidy |>
  filter_bounds(y = roi,return_tidyee = FALSE) |>
  clip(roi,return_tidyee = FALSE)

# pretty instant with return_tidyee=FALSE
ls_clipped_roi_ic <- ls_tidy |>
  filter_bounds(y = roi,return_tidyee = FALSE) |>
  clip(roi,return_tidyee = FALSE)

# takes more time with return_tidyee=T, but you get the vrt
ls_clipped__roi_tidyee <- ls_tidy |>
  filter_bounds(y = roi,return_tidyee = FALSE) |>
  clip(roi,return_tidyee = TRUE)

# demonstrating on sf object
ls_clipped_sf_ob_ic <- ls_tidy |>
  filter_bounds(y = sf_ob,return_tidyee = FALSE) |>
  clip(roi,return_tidyee = FALSE)

ls_clipped_sf_ob_tidyee <- ls_tidy |>
  filter_bounds(y = roi,return_tidyee = FALSE) |>
  clip(roi,return_tidyee = TRUE)

## End(Not run)
```

---

create\_tidyee

*create\_tidyee*

---

## Description

helper function to assign new tidyee when running as\_tidyee

## Usage

```
create_tidyee(x, vrt)
```

## Arguments

x	ee\$ImageCollection
vrt	virtual table

**Value**

tidyee class list object

---

ee_composite	<i>ee_composite</i>
--------------	---------------------

---

**Description**

ee\_composite

**Usage**

```
ee_composite(x, ...)

## S3 method for class 'tidyee'
ee_composite(x, stat, ...)
```

**Arguments**

x	tidyee object containing ee\$ImageCollection
...	other arguments
stat	A character indicating what to reduce the ImageCollection by, e.g. 'median' (default), 'mean', 'max', 'min', 'sum', 'sd', 'first'.

**Value**

tidyee class containing ee\$Image where all images within ee\$ImageCollection have been aggregated based on pixel-level stats

---

ee_extract_tidy	<i>ee_extract_tidy</i>
-----------------	------------------------

---

**Description**

ee\_extract\_tidy



**Usage**

```
ee_extract_tidy(
  x,
  y,
  stat = "mean",
  scale,
  via = "getInfo",
  container = "rgee_backup",
  sf = TRUE,
  lazy = FALSE,
  quiet = FALSE,
  ...
)
```

**Arguments**

x	tidyee, ee\$Image, or ee\$ImageCollection
y	sf or ee\$feature or ee\$FeatureCollection
stat	zonal stat ("mean", "median", "min", "max" etc)
scale	A nominal scale in meters of the Image projection to work in. By default 1000.
via	Character. Method to export the image. Three method are implemented: "getInfo", "drive", "gcs".
container	Character. Name of the folder ('drive') or bucket ('gcs') to be exported into (ignore if via is not defined as "drive" or "gcs").
sf	Logical. Should return an sf object?
lazy	Logical. If TRUE, a future::sequential object is created to evaluate the task in the future. Ignore if via is set as "getInfo". See details.
quiet	Logical. Suppress info message.
...	additional parameters

**Value**

data.frame in long format with point estimates for each time-step and y feature based on statistic provided

**See Also**

[ee\\_extract](#) for information about ee\_extract on ee\$ImageCollections and ee\$Images

**Examples**

```
## Not run:
library(rgee)
library(tidyrg)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
```

```

point_sample_buffered <- tidygee::bgd_msna |>
  sample_n(3) |>
  sf::st_as_sf(coords=c("_gps_reading_longitude",
                       "_gps_reading_latitude"), crs=4326) |>
  sf::st_transform(crs=32646) |>
  sf::st_buffer(dist = 500) |>
  dplyr::select(`_uuid`)
modis_ic_tidy <- as_tidyee(modis_ic)
modis_monthly_baseline_mean <- modis_ic_tidy |>
  select("NDVI") |>
  filter(year %in% 2000:2015) |>
  group_by(month) |>
  summarise(stat="mean")

ndvi_monthly_mean_at_pt<- modis_monthly_baseline_mean |>
  ee_extract(y = point_sample_buffered,
            fun="mean",
            scale = 500)

## End(Not run)

```

---

ee\_month\_composite      *Pixel-level composite by month*

---

## Description

Pixel-level composite by month

## Usage

```

ee_month_composite(x, ...)

## S3 method for class 'ee.imagecollection.ImageCollection'
ee_month_composite(x, stat, months, ...)

## S3 method for class 'tidyee'
ee_month_composite(x, stat, ...)

```

## Arguments

x	An earth engine ImageCollection or tidyee class.
...	extra args to pass on
stat	A character indicating what to reduce the ImageCollection by, e.g. 'median' (default), 'mean', 'max', 'min', 'sum', 'sd', 'first'.
months	A vector of months, e.g. c(1, 12).

## Value

tidyee class containing ee\$Image or ee\$ImageCollection with pixels aggregated by month

---

ee_month_filter	<i>ee_month_filter</i>
-----------------	------------------------

---

**Description**

ee\_month\_filter

**Usage**

```
ee_month_filter(imageCol, month, ...)
```

**Arguments**

imageCol	ee\$ImageCollection
month	numeric vector containing month values (1-12)
...	other arguments

**Value**

ee\$ImageCollection or ee\$Image filtered by month

---

ee_year_composite	<i>Pixel level composite by year</i>
-------------------	--------------------------------------

---

**Description**

Pixel level composite by year

**Usage**

```
ee_year_composite(x, ...)
```

```
## S3 method for class 'ee.imagecollection.ImageCollection'
ee_year_composite(x, stat, year, ...)
```

```
## S3 method for class 'tidyee'
ee_year_composite(x, stat, ...)
```

**Arguments**

x	An earth engine ImageCollection or tidyee class.
...	other arguments
stat	A character indicating what to reduce the ImageCollection by, e.g. 'median' (default), 'mean', 'max', 'min', 'sum', 'sd', 'first'.
year	numeric vector containing years (i.e c(2001,2002,2003))

**Value**

tidyee class containing ee\$Image or ee\$ImageCollection with pixels aggregated by year

---

ee_year_filter	<i>ee_year_filter</i>
----------------	-----------------------

---

**Description**

ee\_year\_filter

**Usage**

```
ee_year_filter(imageCol, year, ...)
```

**Arguments**

imageCol	ee\$ImageCollection
year	numeric vector containing years (i.e c(2001,2002,2003))
...	other arguments

**Value**

ee\$ImageCollection or ee\$Image filtered by year

---

ee_year_month_composite	<i>Pixel-level composite by year and month</i>
-------------------------	------------------------------------------------

---

**Description**

Pixel-level composite by year and month

**Usage**

```
ee_year_month_composite(x, ...)

## S3 method for class 'ee.imagecollection.ImageCollection'
ee_year_month_composite(x, stat, startDate, endDate, months, ...)

## S3 method for class 'tidyee'
ee_year_month_composite(x, stat, ...)
```

**Arguments**

x	An earth engine ImageCollection or tidyee class.
...	args to pass on.
stat	A character indicating what to reduce the ImageCollection by, e.g. 'median' (default), 'mean', 'max', 'min', 'sum', 'sd', 'first'.
startDate	character format date, e.g. "2018-10-23".
endDate	character format date, e.g. "2018-10-23".
months	numeric vector, e.g. c(1,12).

**Value**

tidyee class containing ee\$Image or ee\$ImageCollection with pixels aggregated by year and month

---

ee\_year\_month\_filter    *ee\_year\_month\_filter*

---

**Description**

ee\_year\_month\_filter

**Usage**

```
ee_year_month_filter(imageCol, year, month, ...)
```

**Arguments**

imageCol	ee\$ImageCollection
year	numeric vector contain years to filter
month	numeric vector contain months to filter
...	other arguments

**Value**

ee\$ImageCollection or ee\$Image filtered by year & month

---

filter	<i>filter ee\$ImageCollections or tidyee objects that contain imageCollections</i>
--------	------------------------------------------------------------------------------------

---

### Description

filter ee\$ImageCollections or tidyee objects that contain imageCollections

### Arguments

.data	ImageCollection or tidyee class object
...	other arguments

### Value

filtered image or imageCollection form filtered imageCollection

### See Also

[filter](#) for information about filter on normal data tables.

### Examples

```
## Not run:

library(rgee)
library(tidygee)
ee_initialize()
l8 = ee$ImageCollection('LANDSAT/LC08/C01/T1_SR')
l8 |>
  filter(date>"2016-01-01",date<"2016-03-04")

# example with tidyee class
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic_tidy <- as_tidyee(modis_ic)

# filter by month
modis_march_april <- modis_ic_tidy |>
  filter(month %in% c(3,4))

## End(Not run)
```

---

filter_bounds	<i>filter_bounds a wrapper for rgee::ee\$ImageCollection\$filterBounds</i>
---------------	----------------------------------------------------------------------------

---

**Description**

filter\_bounds a wrapper for rgee::ee\$ImageCollection\$filterBounds

**Usage**

```
filter_bounds(x, y, use_tidyee_index = FALSE, return_tidyee = TRUE)
```

**Arguments**

x	tidyee object containing ee\$ImageCollection or ee\$ImageCollection
y	feature to filter bounds by (sf, ee\$FeatureCollection, ee\$Feature, ee\$Geometry)
use_tidyee_index	filter on tidyee_index (default = F) or system_index (by default)
return_tidyee	logical return tidyee class (default = TRUE) object or ee\$ImageCollection. Faster performance if set to FALSE

**Value**

tidyee class or ee\$ImageCollection class object with scenes filtered to bounding box of y geometry

**Examples**

```
## Not run:

library(tidyrgree)
library(tidyverse)
library(rgee)
rgee::ee_initialize()

# create geometry and convert to sf
coord_tibble <- tibble::tribble(
  ~X,          ~Y,
  92.2303683692011, 20.9126490153521,
  92.2311567217866, 20.9127410439304,
  92.2287527311594, 20.9124072954926,
  92.2289221219251, 20.9197352745068,
  92.238724724534, 20.9081803233546
)
sf_ob <- sf::st_as_sf(coord_tibble, coords=c("X", "Y"), crs=4326)

# load landsat
ls = ee$ImageCollection("LANDSAT/LC08/C01/T1_SR")

#create tidyee class
```

```

ls_tidy <- as_tidyee(ls)

# filter_bounds on sf object
# return tidyee object
ls_tidy |>
  filter_bounds(sf_ob)
# return ee$ImageCollection
ls_tidy |>
  filter_bounds(sf_ob, return_tidyee = FALSE)

# filter_bounds on ee$Geometry object
# return tidyee object
ee_geom_ob <- sf_ob |> rgee::ee_as_sf()
ls_tidy |>
  filter_bounds(ee_geom_ob)

## End(Not run)

```

---

group_by	<i>Group an imageCollection or tidyee object with Imagecollections by a parameter</i>
----------	---------------------------------------------------------------------------------------

---

## Description

Group an imageCollection or tidyee object with Imagecollections by a parameter

## Arguments

.data	ee\$ImageCollection or tidyee object
...	group_by variables
.add	When FALSE, the default, group_by() will override existing groups. To add to the existing groups, use .add = TRUE. This argument was previously called add, but that prevented creating a new grouping variable called add, and conflicts with our naming conventions.
.drop	Drop groups formed by factor levels that don't appear in the data? The default is TRUE except when .data has been previously grouped with .drop = FALSE. See <a href="#">group_by_drop_default()</a> for details.

## Value

ee\$ImageCollection with grouped\_vars attribute

## See Also

[group\\_by](#) for information about group\_by on normal data tables.



**Examples**

```
## Not run:
library(tidyrgree)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic |>
  filter(date>="2016-01-01", date<="2019-12-31") |>
  group_by(year)

## End(Not run)
```

---

group_split	<i>filter ee\$ImageCollections or tidyee objects that contain imageCollections</i>
-------------	------------------------------------------------------------------------------------

---

**Description**

filter ee\$ImageCollections or tidyee objects that contain imageCollections

**Arguments**

.tbl	ImageCollection or tidyee class object
...	other arguments
return_tidyee	logical return tidyee object(default =T), if FALSE - only return ee\$ImageCollection

**Value**

filtered image or imageCollection form filtered imageCollection

**See Also**

[group\\_split](#) for information about filter on normal data tables.

**Examples**

```
## Not run:

library(rgee)
library(tidyrgree)
ee_initialize()
l8 = ee$ImageCollection('LANDSAT/LC08/C01/T1_SR')
l8 |>
  filter(date>"2016-01-01", date<"2016-03-04")

# example with tidyee class
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic_tidy <- as_tidyee(modis_ic)
```

```
# filter by month
modis_march_april <- modis_ic_tidy |>
filter(month %in% c(3,4))

## End(Not run)
```

---

inner_join	<i>inner_join bands from different image/ImageCollections based on shared property</i>
------------	----------------------------------------------------------------------------------------

---

### Description

inner\_join bands from different image/ImageCollections based on shared property

### Arguments

x, y	A pair of tidyee objects containing ee\$ImageCollections
by	A character vector of variables to join by.

### Value

An object of the same type as x. The output has the following properties: Same number of images as x Total number of bands equal the number of bands in x plus the number of bands in y

### See Also

[inner\\_join](#) for information about inner\_join on normal data tables.

---

mutate	<i>mutate columns into tidyee vrt which can later be used to modify tidyee ImageCollection</i>
--------	------------------------------------------------------------------------------------------------

---

### Description

mutate columns into tidyee vrt which can later be used to modify tidyee ImageCollection

### Arguments

.data	tidyee class object (list of ee_ob, vrt)
...	mutate arguments

### Value

return tidyee class object with vrt data.frame mutated.

**See Also**

[mutate](#) for information about mutate on normal data tables.

**Examples**

```
## Not run:
library(tidyrgEE)
library(rgee)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic_tidy <- as_tidyee(modis_ic)

## End(Not run)
```

---

print.tidyee

*print tidyee*


---

**Description**

print tidyee

**Usage**

```
## S3 method for class 'tidyee'
print(x, ...)
```

**Arguments**

x tidyee object  
... additional arguments

**Value**

printed tidyee object

---

select

*Select bands from ee\$Image or ee\$ImageCollection*


---

**Description**

Select bands from ee\$Image or ee\$ImageCollection

**Arguments**

.data tidyee class object containing ee\$ImageCollection or ee\$Image  
... one or more quoted or unquoted expressions separated by commas.

**Value**

tidyee class object with specified (...) bands selected

**See Also**

[select](#) for information about select on normal data tables.

**Examples**

```
## Not run:
library(tidyrgree)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic_tidy <- as_tidyee(modis_ic)

# select NDVI band
modis_ndvi <- modis_ic_tidy |>
  select("NDVI")

# select NDVI band, but change band to new name
modis_ndvi_renamed <- modis_ic_tidy |>
  select(ndvi_new= "NDVI")

## End(Not run)
```

---

set\_idx

*set\_idx*

---

**Description**

set\_idx

**Usage**

```
set_idx(x, idx_name = "tidyee_index")
```

**Arguments**

x                    tidyee or ee\$ImageCollection class object  
 idx\_name            name for index to create (default = "tidyee\_index")

**Value**

tidyee or ee\$ImageCollection class object with new index containing sequential 0-based indexing

**Examples**

```
## Not run:
library(rgee)
library(tidyrgree)
ee_initialize()
modis_link <- "MODIS/006/MOD13Q1"
modisIC <- ee$ImageCollection(modis_link)
modis_ndvi_tidy <- as_tidyee(modisIC) |>
  select("NDVI")
modis_ndvi_tidy |>

## End(Not run)
```

---

slice	<i>slice ee\$ImageCollections or tidyee objects that contain imageCollections</i>
-------	-----------------------------------------------------------------------------------

---

**Description**

slice ee\$ImageCollections or tidyee objects that contain imageCollections

**Arguments**

.data	ImageCollection or tidyee class object
...	other arguments

**Value**

sliced/filtered image or imageCollection form filtered imageCollection

**See Also**

[slice](#) for information about slice on normal data tables.

**Examples**

```
## Not run:

library(rgee)
library(tidyrgree)
ee_initialize()
l8 = ee$ImageCollection('LANDSAT/LC08/C01/T1_SR')
l8 |>
  filter(date>"2016-01-01", date<"2016-03-04")

# example with tidyee class
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
```

```

modis_ic_tidy <- as_tidyee(modis_ic)

# filter by month
modis_march_april <- modis_ic_tidy |>
  filter(month %in% c(3,4))

## End(Not run)

```

---

summarise	<i>Summary pixel-level stats for ee\$ImageCollection or tidygee objects with ImageCollections</i>
-----------	---------------------------------------------------------------------------------------------------

---

### Description

Summary pixel-level stats for ee\$ImageCollection or tidygee objects with ImageCollections

### Usage

```

## S3 method for class 'ee.imagecollection.ImageCollection'
summarise(.data, stat, ...)

## S3 method for class 'tidyee'
summarise(.data, stat, ..., join_bands = TRUE)

```

### Arguments

.data	ee\$Image or ee\$ImageCollection
stat	character stat/function to apply
...	other arguments
join_bands	logical (default= TRUE) if multiple stats selected should bands be joined?

### Value

ee\$Image or ee\$ImageCollection where pixels are summarised by group\_by and stat  
 ee\$Image or ee\$ImageCollection where pixels are summarised by group\_by and stat  
 ee\$Image or ee\$ImageCollection where pixels are summarised by group\_by and stat

### See Also

[summarise](#) for information about summarise on normal data tables.

**Examples**

```
## Not run:
library(tidyrgree)
library(rgee)
ee_initialize()
modis_ic <- ee$ImageCollection("MODIS/006/MOD13Q1")
modis_ic |>
  filter(date>="2016-01-01",date<="2019-12-31") |>
  group_by(year) |>
  summarise(stat="max")

## End(Not run)
```

---

ungroup

*ungroup*

---

**Description**

ungroup

**Arguments**

x	tidyee object
...	ungroup args

**Value**

tidyee class object with vrt ungrouped.

**See Also**

[ungroup](#) for information about ungroup on normal data tables.

# Index

## \* datasets

- bgd\_msna, [4](#)
  
- add\_date\_to\_bandname, [2](#)
- as\_ee, [3](#)
- as\_tidyee, [3](#)
  
- bgd\_msna, [4](#)
- bind\_ics, [5](#)
  
- clip, [6](#)
- create\_tidyee, [7](#)
  
- ee\_composite, [8](#)
- ee\_extract, [9](#)
- ee\_extract\_tidy, [8](#)
- ee\_month\_composite, [10](#)
- ee\_month\_filter, [11](#)
- ee\_year\_composite, [11](#)
- ee\_year\_filter, [12](#)
- ee\_year\_month\_composite, [12](#)
- ee\_year\_month\_filter, [13](#)
  
- filter, [14](#), [14](#)
- filter\_bounds, [15](#)
  
- group\_by, [16](#), [16](#)
- group\_by\_drop\_default(), [16](#)
- group\_split, [17](#), [17](#)
  
- inner\_join, [18](#), [18](#)
  
- mutate, [18](#), [19](#)
  
- print\_tidyee, [19](#)
  
- select, [19](#), [20](#)
- set\_idx, [20](#)
- slice, [21](#), [21](#)
- summarise, [22](#), [22](#)
  
- ungroup, [23](#), [23](#)