

# Package: rgee2 (via r-universe)

May 27, 2026

**Title** Extra functions for R Google Earth Engine

**Version** 0.1.2

**Description** Google Earth Engine Extensions in R.

**License** MIT + file LICENSE

**Depends** R (>= 3.6)

**Imports** rgee, dplyr, tidyr, glue, magrittr, data.table, stringr,  
crayon, lubridate, reticulate

**Suggests** sf, testthat (>= 3.0.0)

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libcu-  
dev libpng-dev libuv1-dev libssl-dev libproj-dev python3 libsqlite3-dev libudunits2-dev

**Repository** <https://rpkgs.r-universe.dev>

**Date/Publication** 2025-10-29 06:13:49 UTC

**RemoteUrl** <https://github.com/rpkgs/rgee2>

**RemoteRef** HEAD

**RemoteSha** 3c0398060de8e68733883e2703fab36fb679dc5b

## Contents

add_dn_date . . . . .	2
col2hex . . . . .	2
ee_aggregate . . . . .	3
ee_date . . . . .	3
ee_extract_clean . . . . .	4
ee_extract2 . . . . .	4
ee_get_proj . . . . .	5

ee_gif . . . . .	5
ee_Init . . . . .	6
ee_tools . . . . .	7
export_Img . . . . .	7
st_point_buffer . . . . .	8

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

---

add_dn_date	<i>add_dn_date</i>
-------------	--------------------

---

### Description

add\_dn\_date

### Usage

```
add_dn_date(img, date_begin = NULL, include.year = TRUE, chunksize = 8)
```

```
cal_di(date_begin, chunksize = 8)
```

```
dndate_start(year, di, chunksize = 8)
```

### Examples

```
## Not run:
dndate_start(2020, 46) %>% cal_di(include.year = FALSE)

## End(Not run)
```

---

col2hex	<i>colors to hex</i>
---------	----------------------

---

### Description

colors to hex

### Usage

```
col2hex(cname)
```

### Arguments

cname            color names

### Examples

```
col2hex("grey60")
```

---

ee_aggregate	<i>ee_aggregate</i>
--------------	---------------------

---

### Description

ee\_aggregate

### Usage

```
ee_aggregate(col, prop, reducerList = "mean", bandList = 0)
```

```
ee_aggregate_list(col, prop, reducerList = "mean", bandList = 0)
```

---

ee_date	<i>ee_date</i>
---------	----------------

---

### Description

Get imgcol or img "system:time\_start" or "system:time\_end"

### Usage

```
as_date_img(x, prop = "system:time_start")
```

```
as_date_millis(x)
```

```
str_date(date)
```

```
ee_systemtime(x, prop = "system:time_start")
```

```
ee_yday(x, ...)
```

```
ee_timestart(x)
```

```
ee_timeend(x)
```

### Arguments

prop "system:time\_start" or "system:time\_end"

### Value

- ee\_yday: ee.Number

---

ee_extract_clean	<i>ee_extract_clean</i>
------------------	-------------------------

---

### Description

ee\_extract\_clean

### Usage

```
ee_extract_clean(x)
```

### Author(s)

Song HeYang, Kong Dongdong

---

ee_extract2	<i>ee_extract2</i>
-------------	--------------------

---

### Description

ee\_extract2

### Usage

```
ee_extract2(
  imgcol,
  y,
  fun = ee$Reducer$mean(),
  scale = NULL,
  prefix = "",
  outfile = NULL,
  lazy = FALSE,
  ...
)
```

### Arguments

<b>y</b>	ee\$Geometry\$, ee\$Feature, ee\$FeatureCollection, sfc or sf objects.
<b>fun</b>	ee\$Reducer object. Function to summarize the values. The function must take a single numeric value as an argument and return a single value. See details.
<b>scale</b>	A nominal scale given in meters of the Image projection to work in. By default 1000.
<b>lazy</b>	Logical. If TRUE, a <code>future::sequential</code> object is created to evaluate the task in the future. Ignore if <code>via</code> is set as "getInfo". See details.
<b>...</b>	ee\$Image\$reduceRegions additional parameters. See <code>ee_help(ee\$Image\$reduceRegions)</code> for more details.

---

ee_get_proj	<i>Get projection of ee.Image or ee.ImageCollection</i>
-------------	---

---

**Description**

Get projection of ee.Image or ee.ImageCollection

**Usage**

```
ee_get_proj(imgcol)
```

**Arguments**

imgcol	ee.Image or ee.ImageCollection
--------	--------------------------------

**Value**

- proj
- scale
- crs
- transform

**Note**

After map operation, scale and transform information will disappear.

---

ee_gif	<i>ee_gif</i>
--------	---------------

---

**Description**

ee\_gif

**Usage**

```
ee_gif(  
  imgcol,  
  vis = NULL,  
  mask = NULL,  
  reigon = NULL,  
  dimensions = 600,  
  crs = "EPSG:3857",  
  framesPerSecond = 2  
)
```

**Arguments**

<code>mask</code>	(Feature Geometry Object) The Geometry or Feature to clip to.
<code>dimensions</code>	A number or pair of numbers in format <code>c(WIDTH,HEIGHT)</code> . Max dimensions of the thumbnail to render, in pixels. If only one number is passed, it is used as the maximum, and the other dimension is computed by proportional scaling.
<code>crs</code>	A CRS string specifying the projection of the output. "EPSG:4326" and "EPSG:3857" (Web Mercator projection) are the most frequent used.
<code>framesPerSecond</code>	String. Animation speed.
<code>region</code>	(optional) <code>ee\$Geometry\$Polygon</code> , GeoJSON or <code>c(E,S,W,N)</code> . Geospatial region of the result. By default, the whole image.
<code>crs_transform</code>	The affine transform to use for the output pixel grid.
<code>scale</code>	A scale to determine the output pixel grid; ignored if both <code>crs</code> and <code>crs_transform</code> are specified.
<code>format</code>	String. The output format (only 'gif' is currently supported).

---

<code>ee_Init</code>	<code>ee_init</code>
----------------------	----------------------

---

**Description**

`ee_init`

**Usage**

```
ee_Init(drive = FALSE, ...)
```

**Arguments**

<code>drive</code>	ignored
<code>...</code>	others to <a href="#">rgee::ee_Initialize()</a>

**See Also**

[rgee::ee\\_Initialize\(\)](#)

**Examples**

```
## Not run:
# ee_Initialize(user = "cuijian426", drive = TRUE)
ee_init()

## End(Not run)
```

---

ee_tools	<i>rgee imgcol tools collection</i>
----------	-------------------------------------

---

### Description

rgee imgcol tools collection

### Usage

```
ee_getInfo(x)
ee_num2str(x)
getInfo(x)
getInfo2(value)
ee_aggregate_array(col, prop = "system:time_start")
ee_systemIndex(col)
ee_first(col)
ee_last(col)
ee_bandNames(x)
ee_propertyNames(col)
ee_properties(col, verbose = TRUE)
ee_filterDate(col, date_start, date_end = NULL)
ee_filterBounds(col, geom)
image_size(x)
```

---

export_Img	<i>export_Img</i>
------------	-------------------

---

### Description

export\_Img

**Usage**

```
export_Img(img, task, options)
```

**Some options to clip regional data**

- 1. crs + region + crsTransform |  $\sqrt{\quad}$
- 1. crs + region + dimensions |  $\sqrt{\quad}$  (resample)
- 1. crs + region + scale |  $\sqrt{\quad}$

---

```
st_point_buffer      ee_point_buffer
```

---

**Description**

```
ee_point_buffer
```

**Usage**

```
st_point_buffer(st, scale = 463.3127, half_win = 1, cellsize = NULL)
```

**Arguments**

<code>st</code>	A data.frame with the columns at least of <code>site</code> , <code>lon</code> , and <code>lat</code> .
<code>scale</code>	in the unit of m
<code>half_win</code>	half window of the buffer (in grids). window size equals to $2 * \text{halfwin} + 1$ .

**Examples**

```
## Not run:
st <- st_212[, .(site, lon, lat, IGBP)]
st_point_buffer_3by3(st, 500)

## End(Not run)
```

# Index

future::sequential, 4

add\_dn\_date, 2

as\_date\_img (*ee\_date*), 3

as\_date\_millis (*ee\_date*), 3

cal\_di (*add\_dn\_date*), 2

col2hex, 2

dndate\_start (*add\_dn\_date*), 2

ee\_aggregate, 3

ee\_aggregate\_array (*ee\_tools*), 7

ee\_aggregate\_list (*ee\_aggregate*), 3

ee\_bandNames (*ee\_tools*), 7

ee\_date, 3

ee\_extract2, 4

ee\_extract\_clean, 4

ee\_filterBounds (*ee\_tools*), 7

ee\_filterDate (*ee\_tools*), 7

ee\_first (*ee\_tools*), 7

ee\_get\_proj, 5

ee\_getInfo (*ee\_tools*), 7

ee\_gif, 5

ee\_Init, 6

ee\_last (*ee\_tools*), 7

ee\_num2str (*ee\_tools*), 7

ee\_properties (*ee\_tools*), 7

ee\_propertyNames (*ee\_tools*), 7

ee\_systemIndex (*ee\_tools*), 7

ee\_systemtime (*ee\_date*), 3

ee\_timeend (*ee\_date*), 3

ee\_timestart (*ee\_date*), 3

ee\_tools, 7

ee\_yday (*ee\_date*), 3

export\_Img, 7

getInfo (*ee\_tools*), 7

getInfo2 (*ee\_tools*), 7

image\_size (*ee\_tools*), 7

rgee::ee\_Initialize(), 6

st\_point\_buffer, 8

str\_date (*ee\_date*), 3