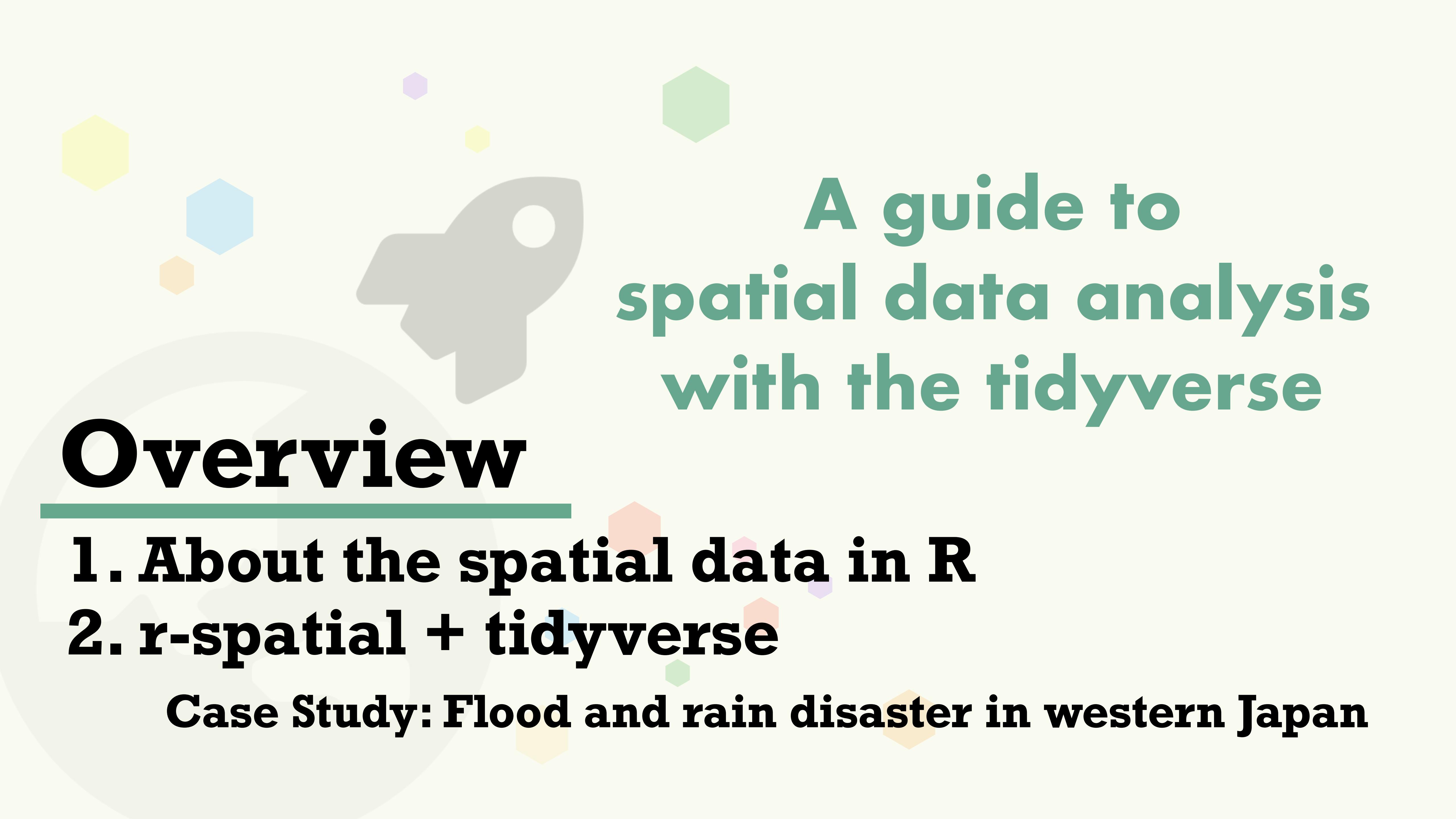




A guide to spatial data analysis with the tidyverse

Shinya Uryu
✉ @u_ribo
🐙 @uribo



A guide to spatial data analysis with the tidyverse

Overview

- 1. About the spatial data in R**
- 2. r-spatial + tidyverse**

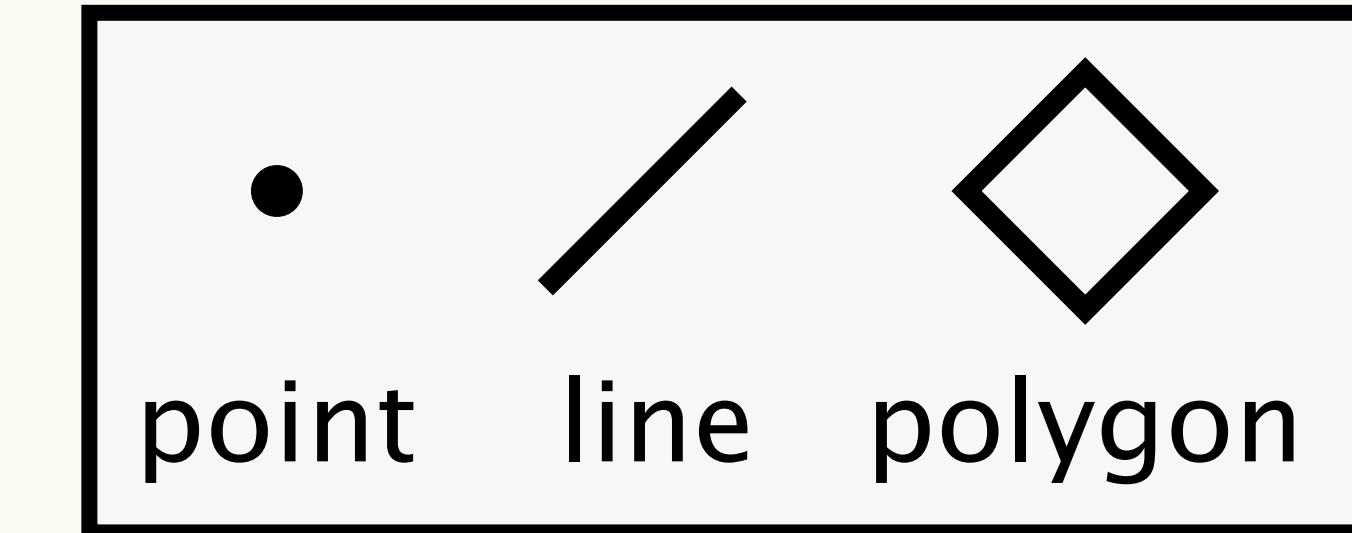
Case Study: Flood and rain disaster in western Japan

Brief history of spatial data in R

2005

sp package rdgal

- Classes and Methods for Spatial Data
- S4 methods
- *Spatial** class (vector)
- lattice based graphics



point line polygon

2010

raster package

ggplot2::geom_raster(),
ggplot2::geom_map() etc.,

maptools

rasterVis package
leaflet package
tmap package

2016

sf package

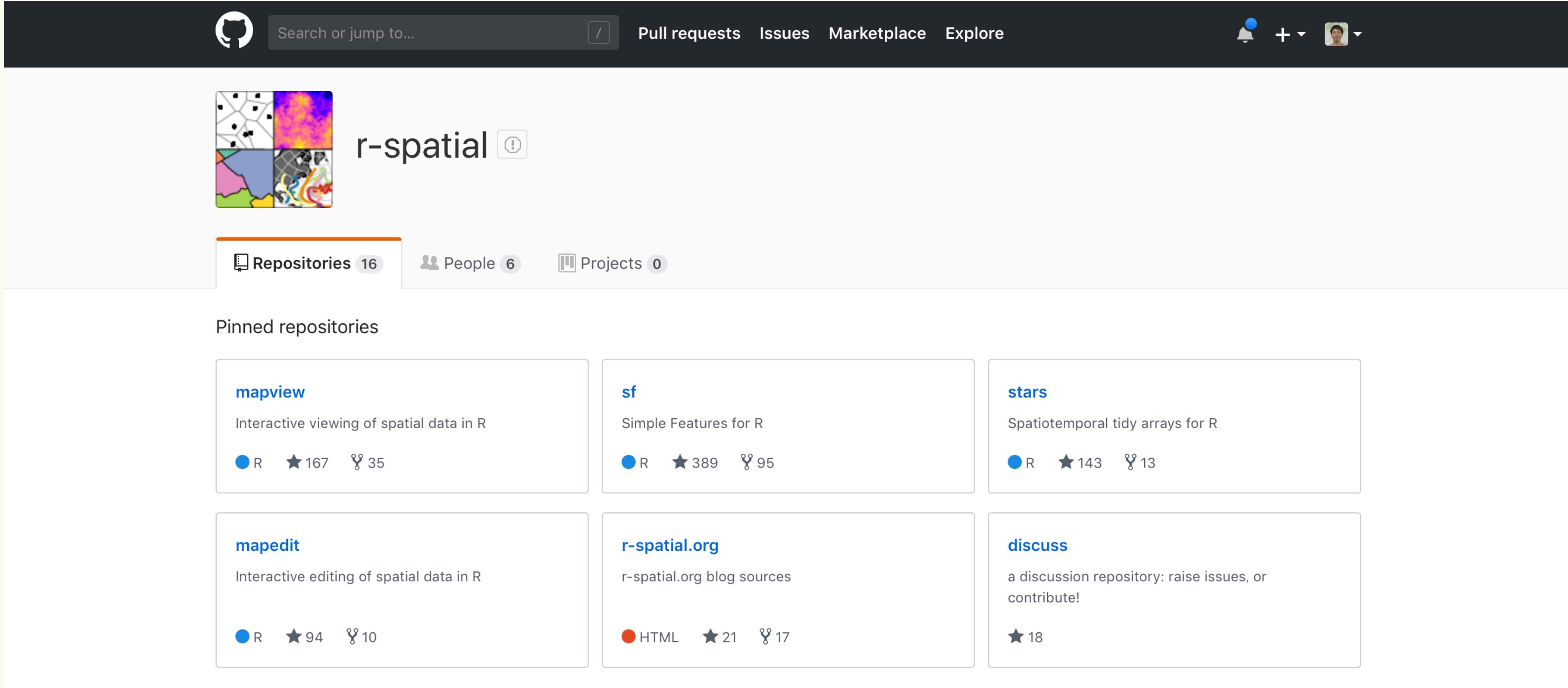
BIG BANG

- mapview package
- stars package
- (...under development)

- tidy manifest
- list-column data frame
- pipe friendly APIs

ggplot2::geom_sf

r-spatial



The screenshot shows the GitHub profile for the 'r-spatial' organization. At the top, there's a dark header bar with the GitHub logo, a search bar containing 'Search or jump to...', and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the far right of the header are icons for notifications, a plus sign, and a user profile. Below the header, the organization's logo is displayed, which is a collage of various spatial data visualizations like maps and heatmaps. The main title 'r-spatial' is next to it. Underneath, there are three tabs: 'Repositories' (16), 'People' (6), and 'Projects' (0). The 'Repositories' tab is selected. Below this, the heading 'Pinned repositories' is followed by six repository cards arranged in two rows of three. Each card includes the repository name, a brief description, the programming language (R), the number of stars, and the number of forks.

Repository	Description	Language	Stars	Forks
mapview	Interactive viewing of spatial data in R	R	167	35
sf	Simple Features for R	R	389	95
stars	Spatiotemporal tidy arrays for R	R	143	13
mapedit	Interactive editing of spatial data in R	R	94	10
r-spatial.org	r-spatial.org blog sources	HTML	21	17
discuss	a discussion repository: raise issues, or contribute!			18

<https://github.com/r-spatial>

sf

```
> library(sf) # version 0.6-3  
> Linking to GEOS 3.6.1, GDAL 2.1.3,  
proj.4 4.9.3
```

Simple feature

To storage and access
model of most geometry types.

In **sf**, represents
simple features as R objects.

```
st_point(c(1, 2)) # X, Y Coordinates
#> POINT (1 2)
st_point(c(1, 2, 3)) # Added Z dimensions
#> POINT Z (1 2 3)

st_linestring(matrix(1:4, ncol = 2))
#> LINESTRING (1 3, 2 4)
```

```
st_polygon(list(
  rbind(
    st_point(c(1, 1)),
    st_point(c(2, 1)),
    st_point(c(2, 2)),
    st_point(c(1, 2)),
    st_point(c(1, 1))))))

#> POLYGON ((1 1, 2 1, 2 2, 1 2, 1 1))
```

*all functions
start with the prefix
“st_”,
(spatial and
temporal)*

3 types of the sf object

```
library(jpndistrict)
#> Loading required package: jpmesh
#> This package provide map data is based on the Digital Map 25000
#> (Map Image) published by Geospatial Information Authority of Japan
#> (Approval No.603FY2017 information usage <http://www.gsi.go.jp>)
```

```
(sf_pref33 <- jpndistrict(33))
#> Simple feature collection with 30 features and 4 fields
```

```
#> geometry type: GEOMETRY
```

```
#> dimens.
```

sfc: Simple feature geometry list-column

```
#> epsg (SRID): 4326
```

```
#> proj4st
```

sf: Simple feature

```
#> # A tibble
```

```
#> #> pref_code prefecture city_code city
#> <chr> <chr> <chr> <chr>
```

```
#> 1 33 岡山県 33101 岡山市 北区
```

geometry
<GEOMETRY [°]>

POLYGON ((133.9098 34.948, ...))

```
#> 2 33 岡山県 33102 岡山市 中区
```

MULTIPOLYGON (((133.9747, ...)))

```
#> 3 33 岡山県 33103 岡山市 南区
```

MULTIPOLYGON (((133.9958, ...)))

```
#> 4 33 岡山県 33202 倉敷市
```

MULTIPOLYGON (((133.6414, ...)))

```
#> 5 33 岡山県 33202 倉敷市
```

sfg: Simple feature geometry

+no_defs

3 types of the sf object

sf > sfc > sfg

```
class(sf_pref33)
#> [1] "sf"           "tbl_df"
     "tbl"          "data.frame"
class(sf_pref33$geometry)
#> [1] "sfc_GEOMETRY" "sfc"
class(sf_pref33$geometry[[1]])
#> [1] "XY"           "POLYGON" "sfg"
```

Seamlessly convert each sf objects.

```
(x <- st_point(c(1, 2)))
#> POINT (1 2)
x %>%
  st_sfc()
#> Geometry set for 1 feature
#> geometry type: POINT
#> dimension: XY
#> bbox: xmin: 1 ymin: 2 xmax: 1 ymax: 2
#> epsg (SRID): NA
#> proj4string: NA
#> POINT (1 2)
x %>%
  st_sfc() %>%
  st_sf()
#> Simple feature collection with 1 feature and 0 fields
#> geometry type: POINT
#> dimension: XY
#> bbox: xmin: 1 ymin: 2 xmax: 1 ymax: 2
#> epsg (SRID): NA
#> proj4string: NA
#>      geometry
#> 1 POINT (1 2)
```

I/O

Standard GIS file formats

- shapefile
- geojson (as strings)
- kml

```
st_read(system.file("shape/nc.shp",
                    package = "sf"))

st_write(nc, "nc.shp")
```

Relational Database

- PostgreSQL

well known-text(WKT),
well known-binary (WKB) formats

Geo-processing functions

Provided like a PostGIS functions.

- *st_distance()*, *st_area()*, *st_make_gird()* etc.,

```
st_area(sf_pref33)
#> Units: m^2
#> [1] 451746760 51433590 161417813 133400415 357919550 507973617 107320002
#> [8] 136991748 245082627 212789693 548694329 794535530 260097992 127276525
#> [15] 210361376 830459441 430381734 67080736 144833382 7786478 12264588
#> [22] 91311276 67191543 420647698 54742098 70131416 58399305 79252153
#> [29] 232806322 269530873
st_bbox(sf_pref33)
#>      xmin      ymin      xmax      ymax
#> 133.26669  34.29839 134.41322  35.35286                      # My Hometown (Tsukuba)
st_distance(st_sfc(st_point(c(140.10, 36.08)), crs = 4326),
            st_sfc(st_point(c(139.74, 35.68)), crs = 4326)) # Here
#> Units: m
#> [,1]
#> [1,] 55014.41
```

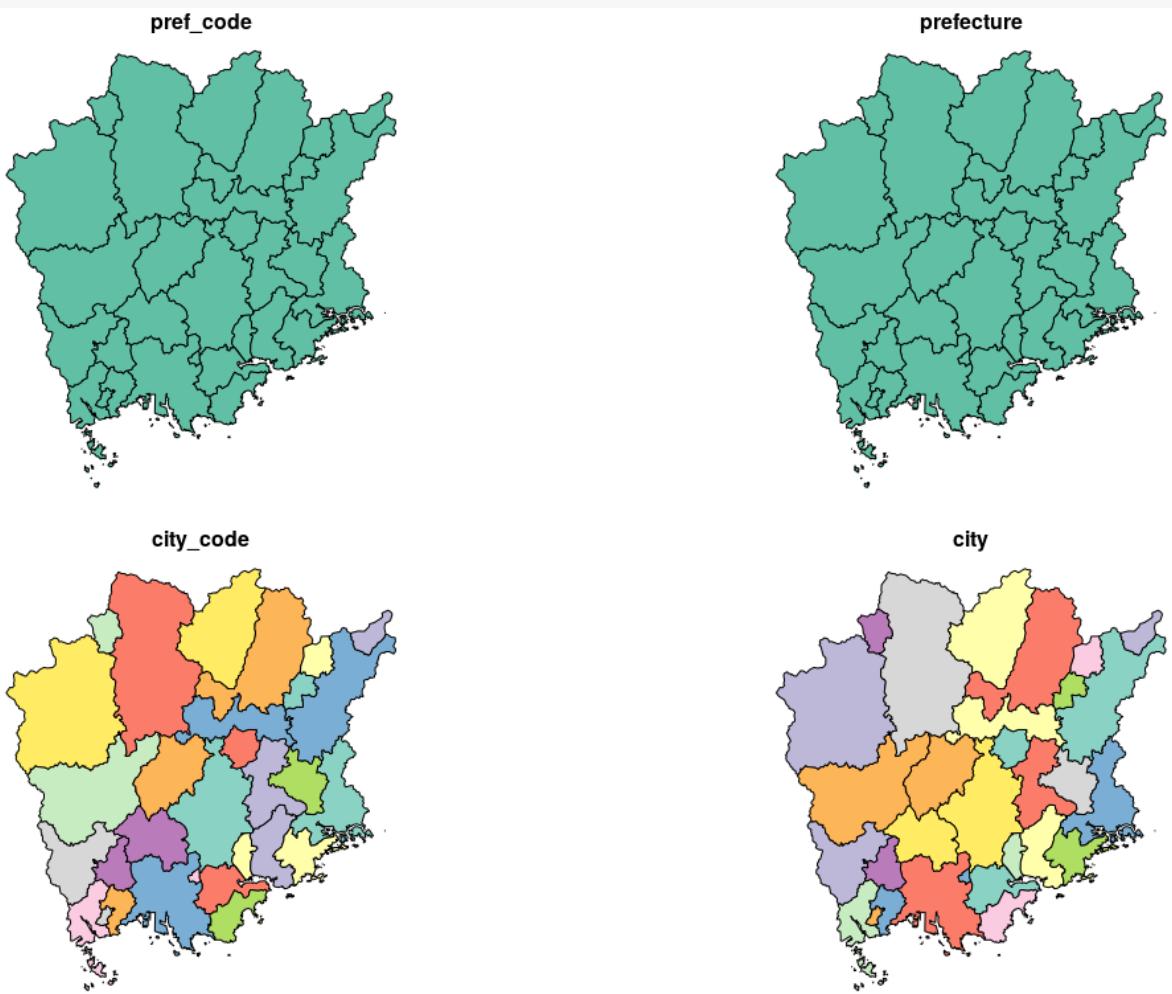
Another support functions found on lwgeom pkgs.

mapping &
visualization

plot.sf*(): S3 method for sf

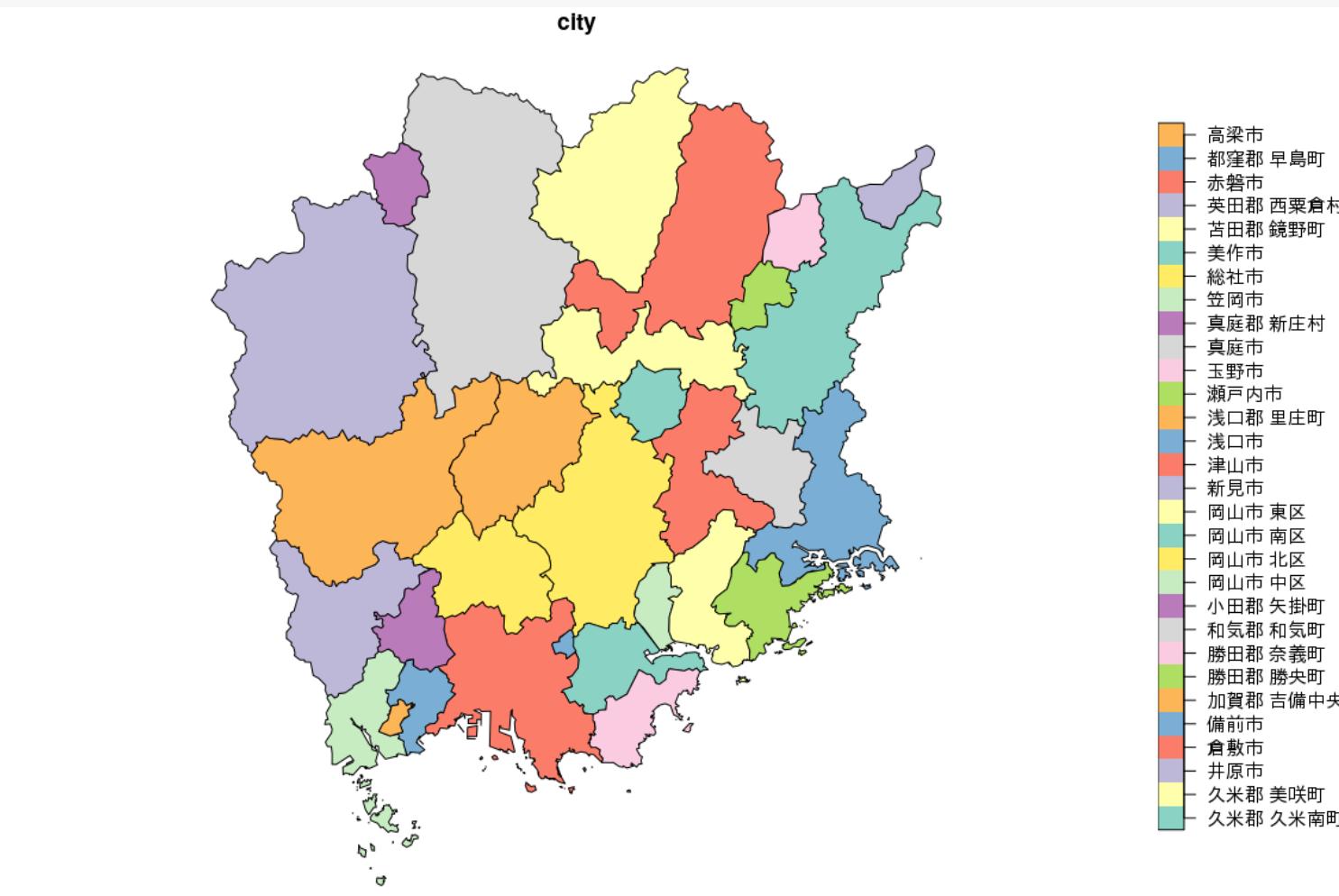
ALL

```
plot(sf_pref33)  
# Drawing the first 10 out  
of all attributes  
# Set by max.plot
```



SELECT

```
plot(sf_pref33[ "city" ],  
      key.pos = 4,  
      key.width = lcm(5),  
      key.length = 0.8)
```



GEOMETRY

```
plot(st_geometry(sf_pref33),  
      col = "white")
```

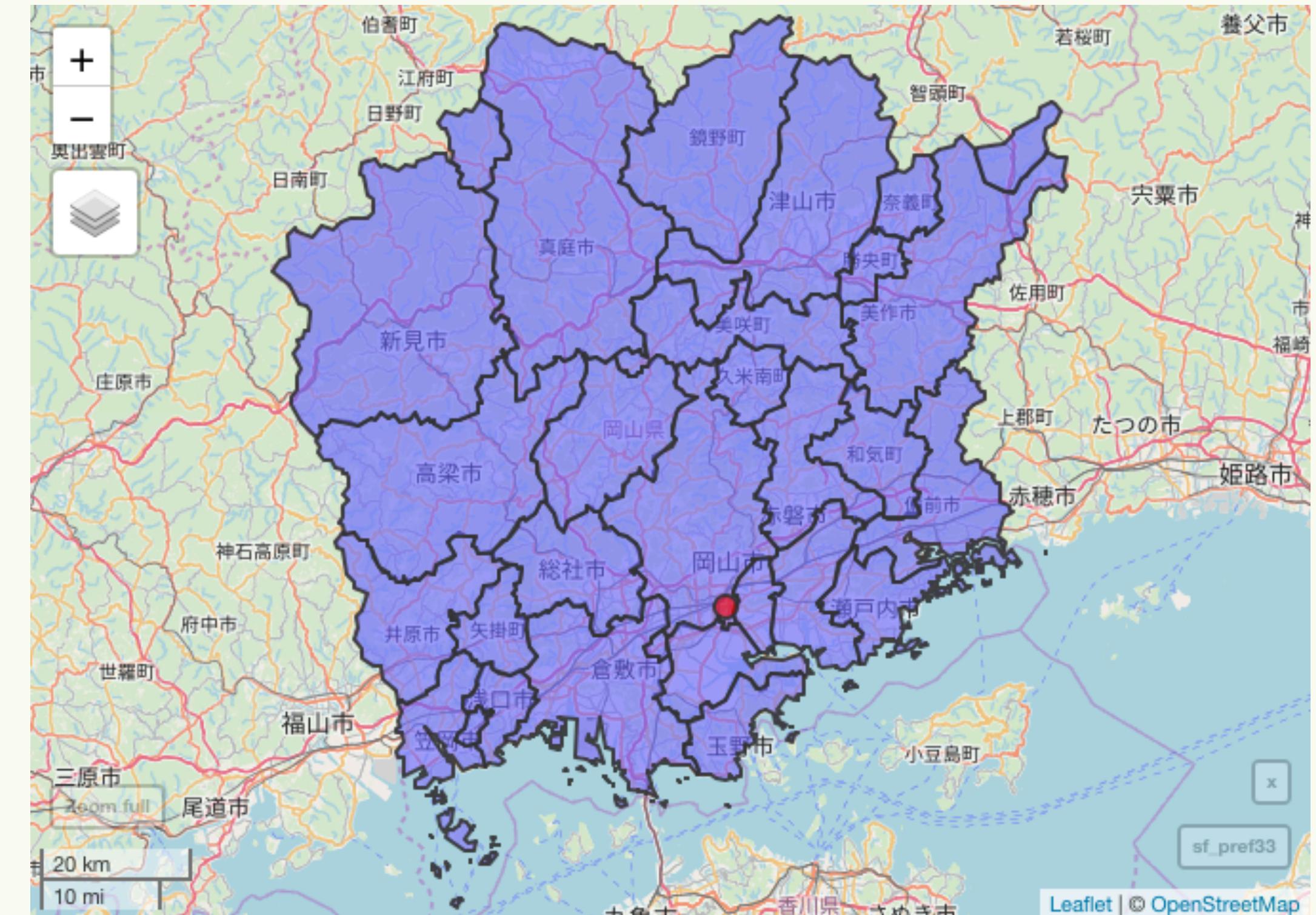


Interactive mapping with mapview

Raster* (raster), Spatial* (sp) and sf

```
library(mapview)
#> Okayama City Administration
st_sfc(
  st_point(c(133.9196,
            34.65511)),
  crs = 4326)

mapview(sf_pref33) +
  mapview(x,
          col.regions = "red")
```



DEMO

r-spatial meets tidyverse

```
> library(tidyverse) # version 1.2.1
```

— Attaching packages —

tidyverse 1.2.1 —

✓ ggplot2 3.0.0	✓ purrr 0.2.5
✓ tibble 1.4.2	✓ dplyr 0.7.6
✓ tidyr 0.8.1	✓ stringr 1.3.1
✓ readr 1.1.1	✓ forcats 0.3.0

— Conflicts —

tidyverse_conflicts() —

- ✗ **dplyr::filter()** masks **stats::filter()**
- ✗ **dplyr::lag()** masks **stats::lag()**

Case Study: Flood and rain disaster in western Japan

The Heavy Rain Event on July 2018 (平成30年7月豪雨).

A total of 11 prefectures that emergency heavy rain warnings areas.

As of 14 July, 197 were dead in various rain-related incidents.

According to the Fire and Disaster Management Agency

Create base map with purrr

```
# Provided by National Spatial Planning and Regional
# Policy Bureau, MILT of Japan
sf_west_japan <-
  26:46 %>%
  map(jpn_pref, district = FALSE) %>%
  # Unused map_dfr() ... bind_rows() not support sf class
  reduce(rbind)

format(object.size(sf_west_japan), units = "MB")
# [1] "18.1 Mb"

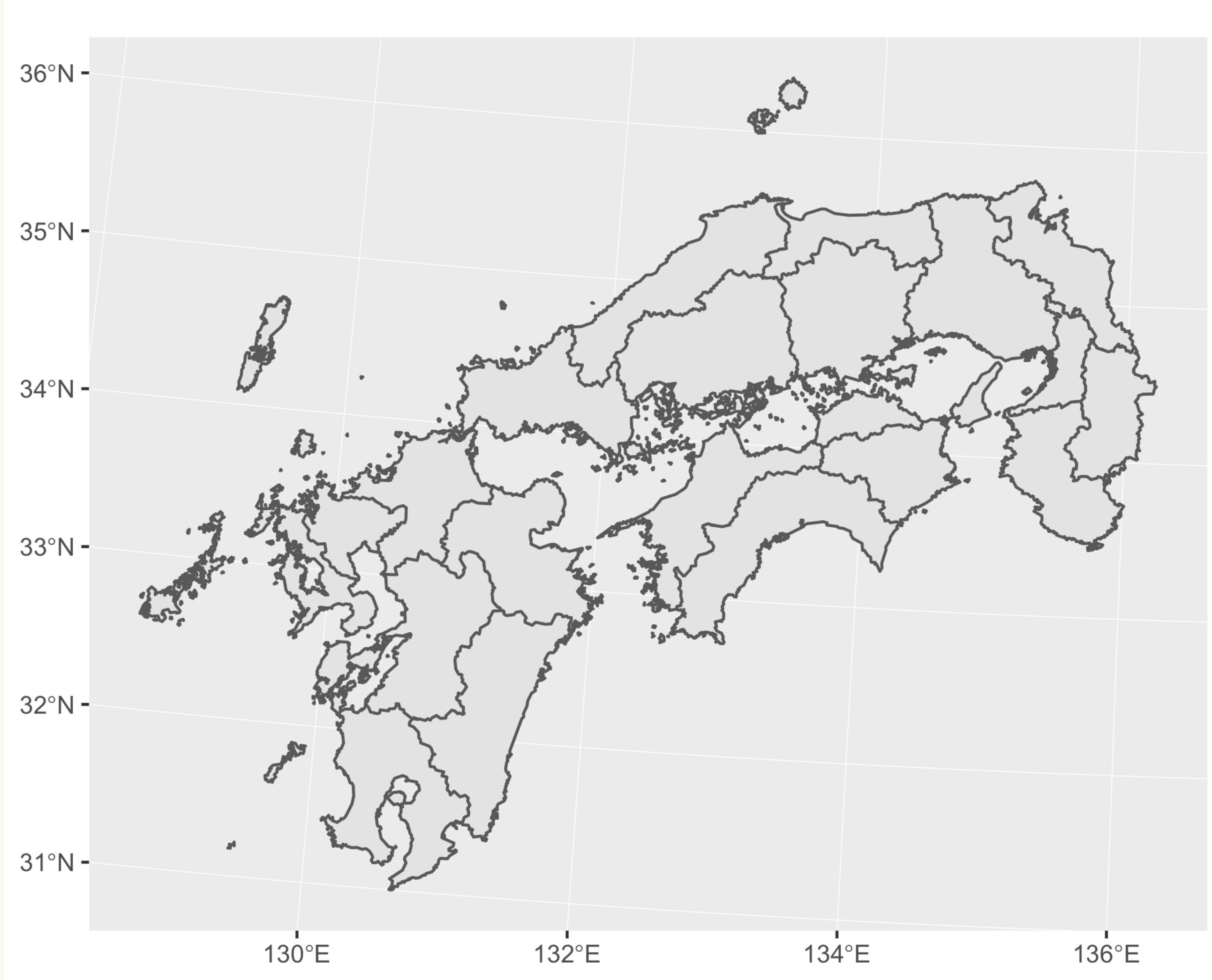
# Area cropped and simplified polygon
sf_west_japan <-
  st_crop(
    sf_west_japan,
    st_bbox(c(xmin = 128.5, ymin = 31.0, xmax = 136.5, ymax = 36.5)))
  ) %>%
  # Reduce object size
  st_simplify(preserveTopology = TRUE, dTolerance = 0.005)

format(object.size(sf_west_japan), units = "MB")
# [1] "1 Mb"
```

Mapping by ggplot2::geom_sf()

```
sf_west_japan <-  
  sf_west_japan %>%  
  st_transform(  
    crs = "+proj=laea  
          +lat_0=30  
          +lon_0=140")
```

```
base_map <-  
  ggplot(sf_west_japan) +  
  geom_sf()
```

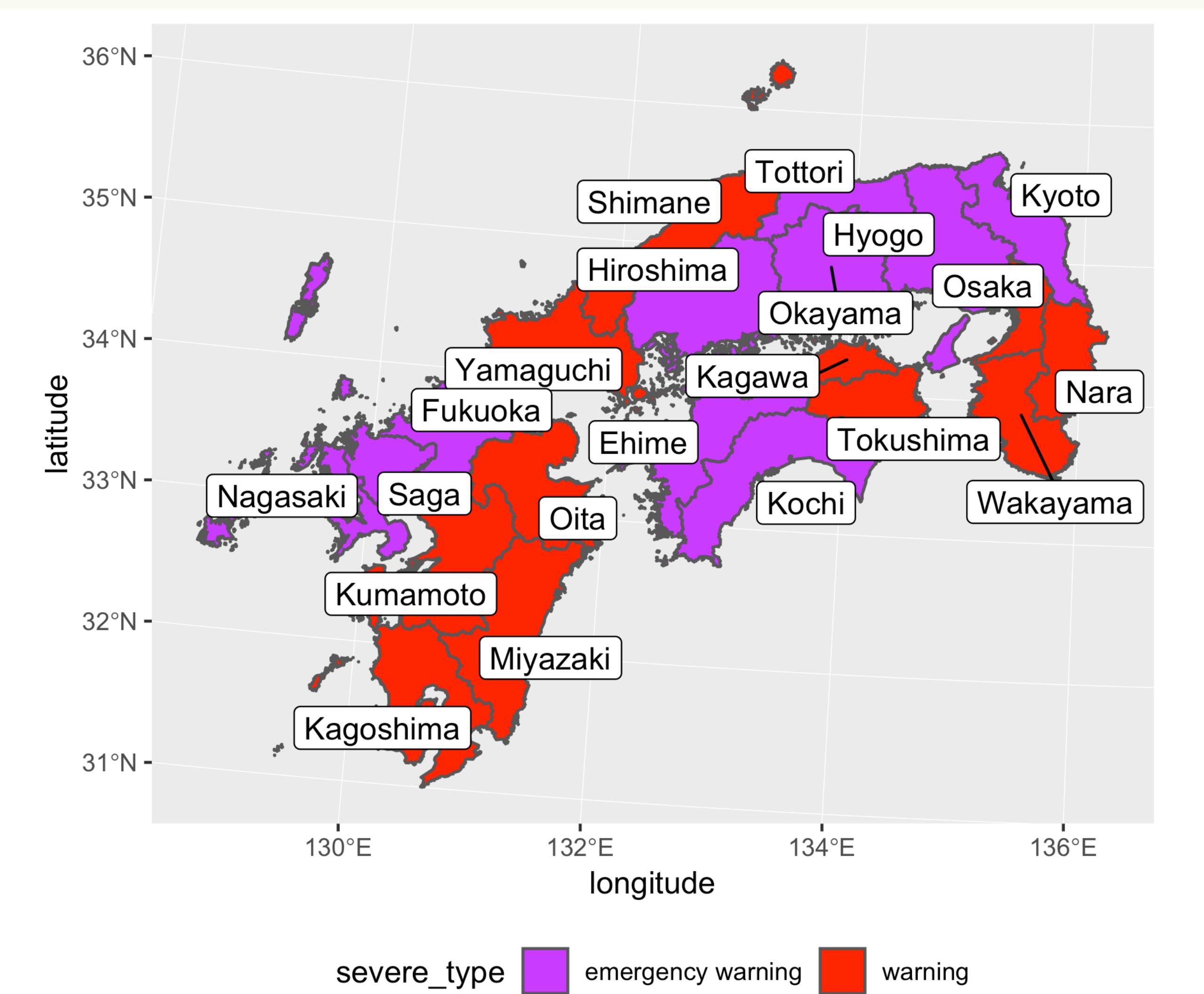


Mapping by ggplot2::geom_sf()

Brush-up 🌈

```
df_west_japan <-
  sf_west_japan %>%
  mutate(
    longitude =
      st_coordinates(st_centroid(geometry))[, 1],
    latitude =
      st_coordinates(st_centroid(geometry))[, 2]) %>%
  st_set_geometry(NULL)

base_map +
  ggrepel::geom_label_repel(
    data = df_west_japan,
    aes(x = longitude,
        y = latitude,
        label = prefecture)) +
  coord_sf()
```



Downloading weather data from JMA

Japan Meteorological Agency

```
> remotes::install_git("https://gitlab.com/uribo/jmastats")
> library(jmastats) # ! Under Development 🚧
```

Access to Japan Meteorological Agency Data

Not provided API



国土交通省 気象庁 Japan Meteorological Agency

各データ・資料 | 知識・解説 | 気象庁について | 案内・申請

過去の気象データ検索

各地の気温、降水量、風など 高層の気温、風など 最新の気象データ 過去の気象データ ダウンロード 地域平均気象データ

年月日の選択 地点の選択

都道府県 都道府県 地点を選択 (未選択)

年月日の選択

1月 1日 16日 2月 2日 17日 3月 3日 18日 4月 4日 19日 5月 5日 20日 6月 6日 21日 7月 7日 22日 8月 8日 23日 9月 9日 24日 10月 10日 25日 11月 11日 26日 12月 12日 27日

13日 28日 14日 29日 15日 30日 31日

データの種類

年ごとの値を表示 (地点を指定してください) 年・月ごとの平年値を表示 (地点を指定してください) 3ヶ月ごとの値を表示 (地点、年を指定してください) 3ヶ月ごとの平年値を表示 (地点を指定してください) 観測開始から月ごとの値を表示 (地点を指定してください) 旬ごとの平年値を表示 (地点を指定してください) 月ごとの値を表示 (地点、年を指定してください) 半旬ごとの平年値を表示 (地点を指定してください) 日ごとの平年値を表示 (地点、月を指定してください) 旬ごとの値を表示 (地点、年を指定してください) 霜・雪・結氷の初終日と初冠雪日の平年値を表示 (気象台、測候所などのみのデータです)

年年値は1981-2010年の30年間の観測値の平均をもとに算出しています。

要素別データの公開期間 (気象台、測候所などのみのデータです) メッシュ平年値図

新着情報

- 気象官署の観測場所の移転に伴い、平年値の一部（「新潟」（新潟県）の気温、蒸気圧、湿度並びに「浜松」（静岡県）及び「山口」（山口県）の気温、蒸気圧、温度、風速及び日照）を更新しました。（2018年5月16日）
- 「洲本」（兵庫県）において、気圧計の設定に誤りがあったため、2017年11月30日から2018年2月19日までの気圧データを修正しました。（2018年4月17日）
- 「南大東（南大東島）」（沖縄県）の「1時間ごとのページ」において、記事欄は、システム上の制限により、記録が11行を超える場合には、それ以降の行が表示されていないことがあります。利用に当たってはご注意ください。（2016年4月27日）

過去のお知らせ

*掲載している気象データは、過去にさかのぼって修正する場合があります。ご利用の際には、最新の掲載データをご確認ください。なお、主な修正については、新着情報の中でお知らせします。

利用される方へ よくある質問 (FAQ) 気象観測統計の解説 年・季節・各月の天候 このページのトップへ

We can extract contents with rvest package.

国土交通省 気象庁 Japan Meteorological Agency

各データ・資料 | 知識・解説 | 気象庁について | 案内・申請

ホーム > 各種データ・資料 > 過去の気象データ検索 > 1時間ごとの値

1時間ごとの値

一覧表 グラフ 見出しの固定 メニューに戻る

前年 前月 前日 翌日 翌月 翌年

日ごとの値 1時間ごとの値 10分ごとの値

本山 2018年7月6日 (1時間ごとの値)

時	降水量 (mm)	気温 (°C)	風速・風向(m/s)	日照時間 (h)	雪(cm)	降雪	積雪
1	2.0	20.9	0.2	静穏	/	/	/
2	2.0	21.0	0.0	静穏	/	/	/
3	3.5	21.2	1.4	南西	/	/	/
4	8.0	21.1	0.0	静穏	/	/	/
5	22.5	21.8	0.6	北東	/	/	/
6	12.5	21.7	0.9	西北西	0.0	/	/
7	9.0	21.5	0.5	北北東	0.0	/	/
8	6.5	21.3	0.3	西北西	0.0	/	/
9	8.0	22.0	0.9	南西	0.0	/	/
10	6.5	22.1	0.7	西南西	0.0	/	/
11	9.5	22.6	0.6	北東	0.0	/	/
12	30.5	22.7	1.2	北東	0.0	/	/
13	17.5	23.4	1.8	北	0.0	/	/
14	36.5	24.5	2.8	西南西	0.0	/	/
15	16.0	23.6	0.7	北西	0.0	/	/
16	25.5	23.6	1.9	南東	0.0	/	/
17	45.5	23.3	1.5	東	0.0	/	/
18	23.0	23.2	1.4	東北東	0.0	/	/
19	28.0	22.7	1.9	東北東	0.0	/	/
20	42.5	23.9	2.3	北北東	0.0	/	/
21	29.5	23.8	2.4	西	/	/	/
22	76.0	23.8	2.2	南	/	/	/
23	21.5	22.4	0.8	北北東	/	/	/
24	28.5	23.1	1.4	北北西	/	/	/

値欄の記号の説明

利用される方へ よくある質問 (FAQ) 気象観測統計の解説 年・季節・各月の天候 このページのトップへ

<http://www.data.jma.go.jp/obd/stats/etrn/index.php>

Collect weather data

```
jma_collect("hourly",
  block_no = tgt_stations$block_no[6], # 舞鶴 (Maiduru, Kyoto)
  year = 2018,
  month = 7,
  day = 5) %>%
  parse_unit()
# A tibble: 24 x 17
  time atmosphere_land_... atmosphere_surf... precipitation_mm temperature dew_point
  <int> <S3: units>          <S3: units>          <S3: units>    <S3: units> <S3: uni>
1     1 1003                  1003.5                 0            25.7      23.2
2     2 1003.1                1003.6                 0            25.4      23.5
3     3 1003.1                1003.6                 0            25.3      23.4
4     4 1003.4                1003.9                 0            25.1      23.5
5     5 1003.9                1004.4                 0            25         23.4
# ... with 19 more rows, and 11 more variables: vapor_h_pa <S3: units>,
#   humidity_percent <S3: units>, wind_speed_m_s <S3: units>,
#   wind_direction_m_s <chr>, daylight_h <S3: units>,
#   solar_irradiance_mj_m_2 <chr>, snow_fall_moment_cm <chr>,
#   snow_fall_period_cm <chr>, weather <chr>, cloud_covering <chr>,
#   visibility_km <S3: units>
```

Collect weather data

1. CRS transformed (To match polygons)
2. Convert MULTIPOINT to (single) POINT
3. Applied dplyr's verbs
 1. Check: Whether points are included in the area?

```
tgt_stations <-  
  stations %>%  
  st_transform(crs = "+proj=laea +lat_0=30 +lon_0=140") %>%  
  st_cast("POINT") %>%  
  select(area, station_type, station_name, elevation, block_no, pref_code) %>%  
  distinct(block_no, .keep_all = TRUE) %>%  
  filter(pref_code %in% 26:46) %>%  
  mutate(in_area = purrr::pmap_lgl(., ~  
    as.logical(sum(st_within(  
      ..7,  
      sf_west_japan,  
      sparse = FALSE)))) %>%  
  filter(in_area == TRUE)
```

Collect weather data

```
df_weather <-
  list(block_no = rep(tgt_stations$block_no, each = 3),
       year = 2018,
       month = 7,
       day = rep(5:7, times = nrow(tgt_stations))) %>%
  pmap(~ jma_collect(item = "hourly",
                      block_no = ..1,
                      year = ..2,
                      month = ..3,
                      day = ..4) %>%
    select(time, `precipitation_(mm)`)) %>%
  parse_unit() %>%
  transmute(block_no = c(..1),
            date = lubridate::make_datetime(..2, ..3, ..4, hour = time),
            precipitation_mm)) %>%
  reduce(rbind) # keep units attributes
```

Collect weather data

```
df_weather
# A tibble: 26,136 x 3
  block_no date           precipitation_mm
  <chr>     <dttm>          <S3: units>
1 1578      2018-07-05 01:00:00 0
2 1578      2018-07-05 02:00:00 1.5
3 1578      2018-07-05 03:00:00 1
# ... with 26,133 more rows
```

```
df_precipitation <-
  df_weather %>%
  group_by(block_no) %>%
  summarise(precipitation = sum(precipitation_mm) )
# A tibble: 363 x 2
  block_no precipitation
  <chr>        <S3: units>
1 0588         355
2 0589         466
3 0593        439.5
# ... with 360 more rows
```

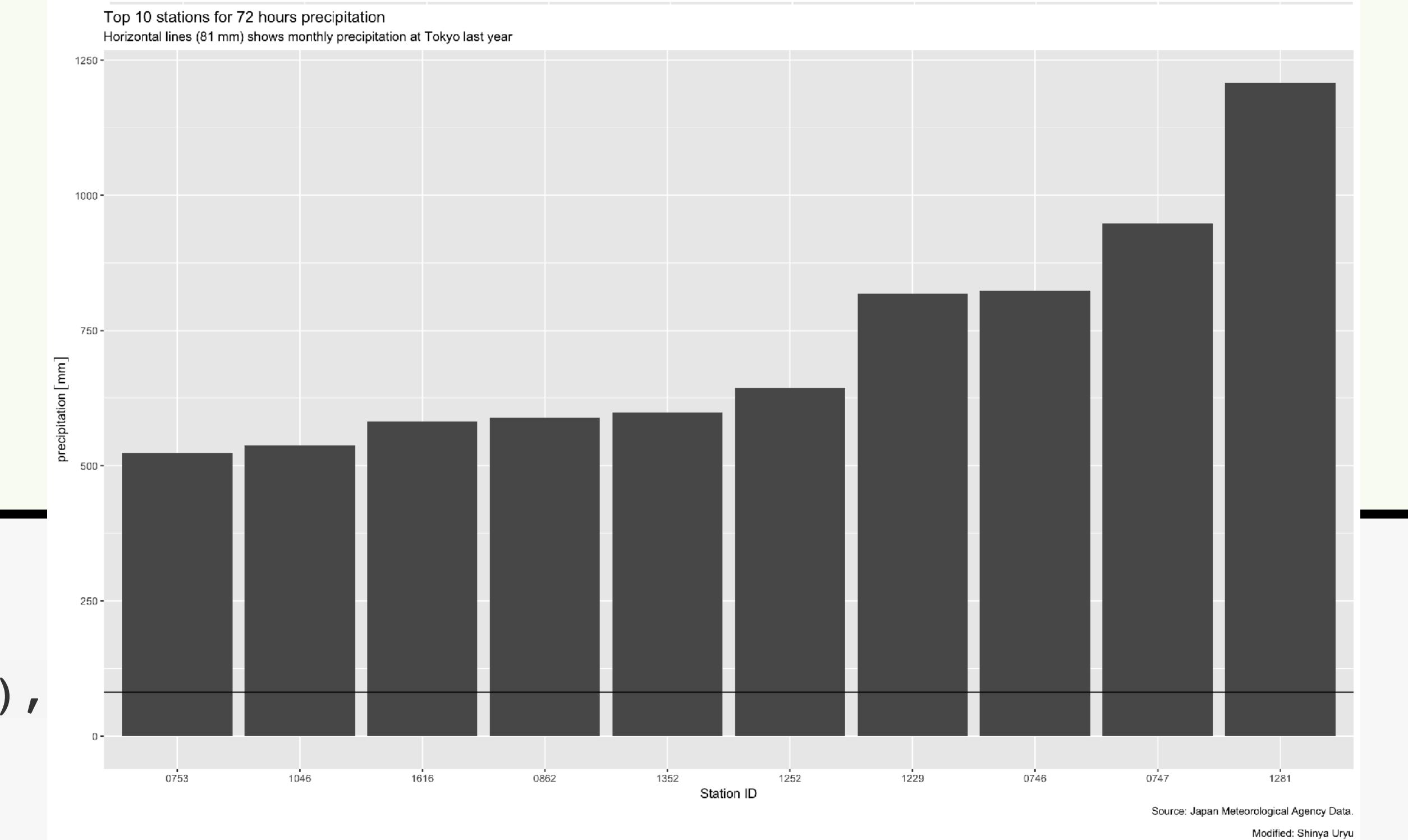
Top 10 stations for 72 hours precipitation

```
df_precipitation_top10 <-
  df_precipitation %>%
  top_n(10, wt = precipitation)
```

🌈 Brush-up 🌈

```
library(ggforce) # For display "units"
library(dplyr)
library(forcats)

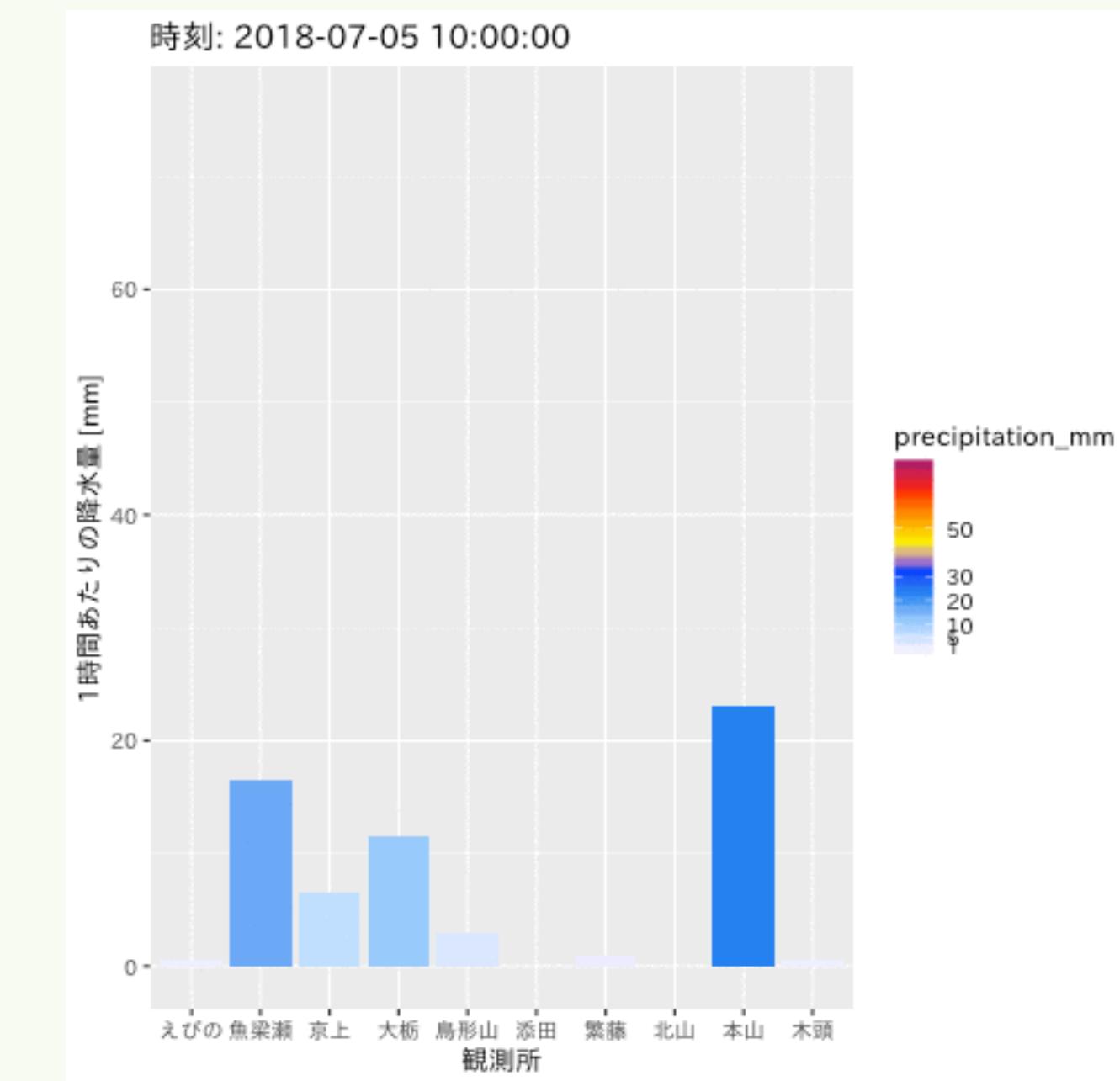
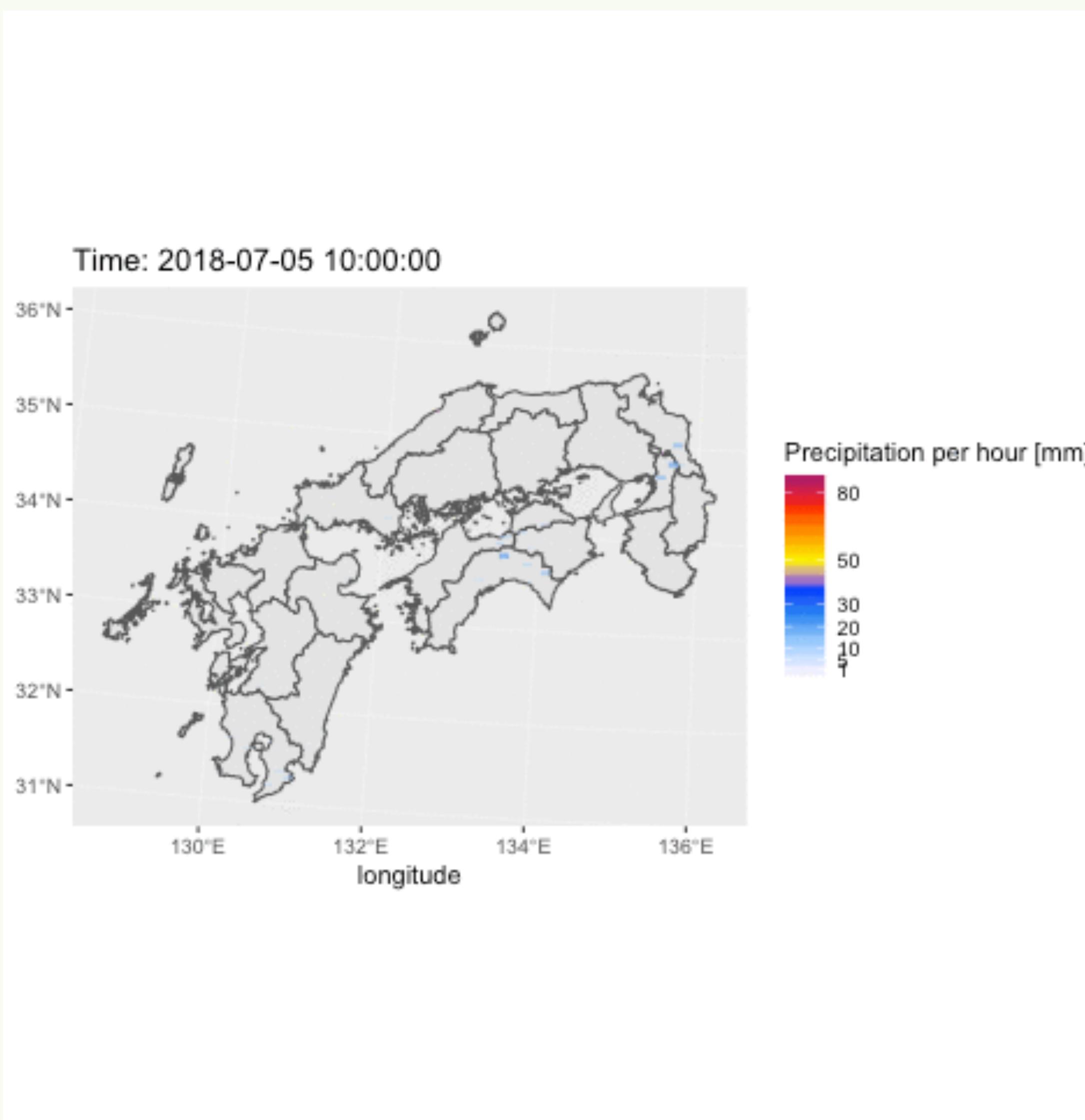
ggplot(df_precipitation_top10,
       aes(forcats::fct_reorder(block_no,
                                 units::drop_units(precipitation)),
            precipitation)) +
  geom_bar(stat = "identity") +
  geom_hline(yintercept = 81) +
  labs(x = "Station ID",
       title = "Top 10 stations for 72 hours precipitation",
       subtitle = "Horizontal lines (81 mm) shows monthly precipitation at Tokyo last year",
       caption = "Source: Japan Meteorological Agency Data.\nModified: Shinya Uryu")
```



mapping plus animation

```
> remotes::install_github("thomasp85/gganimate")
> library(gganimate) # version 0.9.9.9999
```

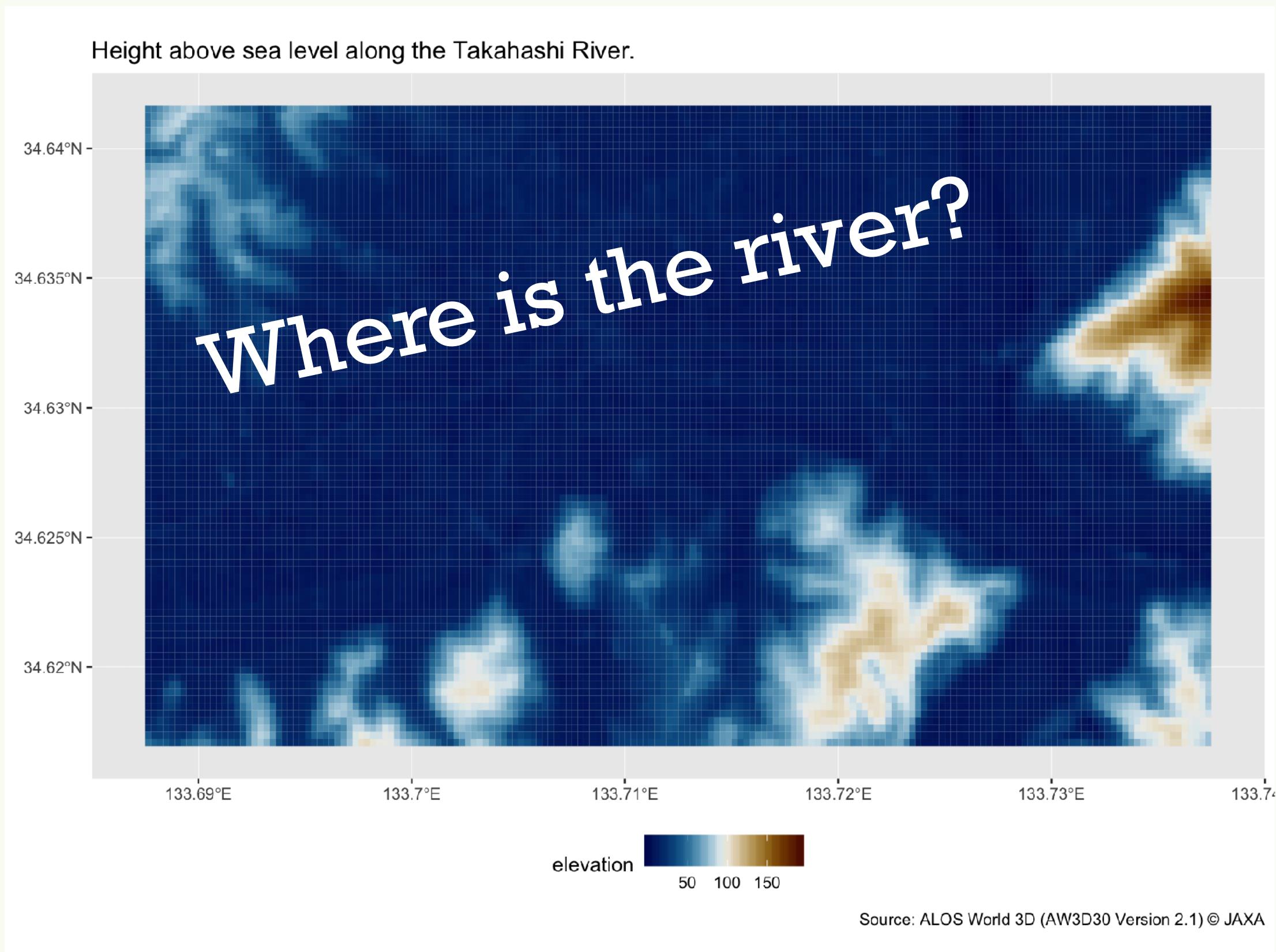
Mapping on precipitation values



Levee failures in Mabi (真備)

Embankment broke, flooding occurred

Water levels reaching 4.8 m

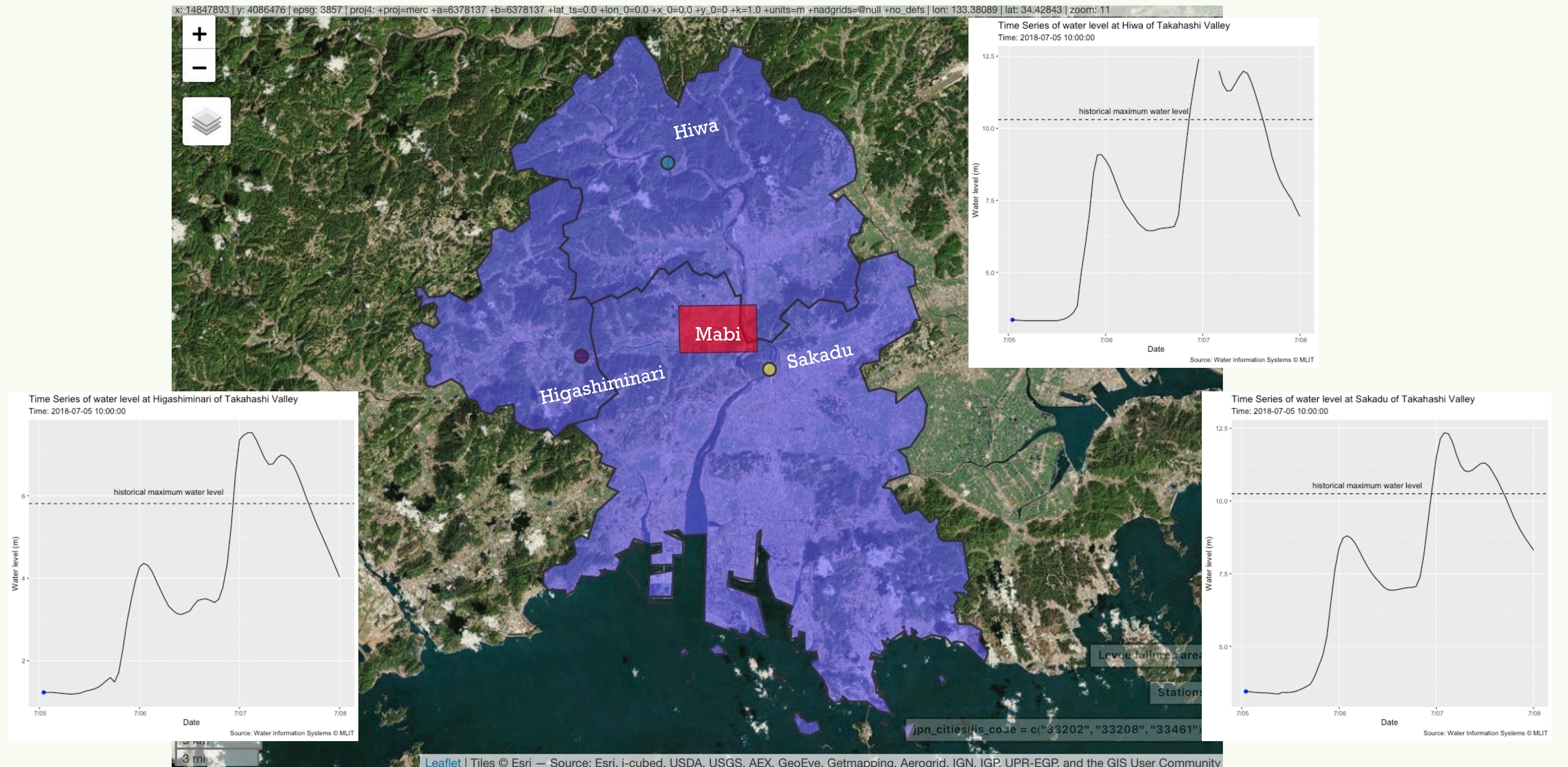


Original Images: Geospatial Information Authority of Japan, composed by 軽快 (CC BY 4.0)

<https://commons.wikimedia.org/wiki/File:平成30年7月豪雨による真備地区の被害.jpg>

<https://creativecommons.org/licenses/by/4.0/>

How changes the river's water level?



Session Information

	package	*	version	date	source
1	abind	*	1.4-5	2016-07-21	CRAN (R 3.5.0)
2	base	*	3.5.0	2018-04-24	local
3	bindrcpp	*	0.2.2	2018-03-29	CRAN (R 3.5.0)
4	datasets	*	3.5.0	2018-04-24	local
5	dplyr	*	0.7.6	2018-06-29	CRAN (R 3.5.0)
6	forcats	*	0.3.0	2018-02-19	CRAN (R 3.5.0)
7	gganimate	*	0.9.9.9999	2018-07-14	Github (thomasp85/gganimate@13a9a29)
8	ggforce	*	0.1.3	2018-07-07	CRAN (R 3.5.0)
9	ggplot2	*	3.0.0	2018-07-03	CRAN (R 3.5.0)
10	graphics	*	3.5.0	2018-04-24	local
11	grDevices	*	3.5.0	2018-04-24	local
12	jmastats	*	0.0.0.9000	2018-07-13	git (@a90e3a3)
13	jpmesh	*	1.1.1.9000	2018-06-26	local (uribo/jpmesh@5054663)
14	jpdistrict	*	0.3.2	2018-06-14	CRAN (R 3.5.0)
15	mapview	*	2.4.0	2018-04-28	CRAN (R 3.5.0)
16	methods	*	3.5.0	2018-04-24	local
17	purrr	*	0.2.5	2018-05-29	CRAN (R 3.5.0)
18	raster	*	2.6-7	2017-11-13	CRAN (R 3.5.0)
19	readr	*	1.1.1	2017-05-16	CRAN (R 3.5.0)
20	sf	*	0.6-3	2018-05-17	CRAN (R 3.5.0)
21	sp	*	1.3-1	2018-06-05	CRAN (R 3.5.0)
22	stars	*	0.1-1	2018-07-13	Github (r-spatial/stars@c9af832)
23	stats	*	3.5.0	2018-04-24	local
24	stringr	*	1.3.1	2018-05-10	CRAN (R 3.5.0)
25	tibble	*	1.4.2	2018-01-22	CRAN (R 3.5.0)
26	tidyverse	*	0.8.1	2018-05-18	CRAN (R 3.5.0)
27	tidyverse	*	1.2.1	2017-11-14	CRAN (R 3.5.0)
28	utils	*	3.5.0	2018-04-24	local

setting	value
version	R version 3.5.0 (2018-04-23)
system	x86_64, darwin15.6.0
ui	RStudio (1.2.826)
language	En
collate	ja_JP.UTF-8
tz	Asia/Tokyo
date	2018-07-18



ENJOY



Shinya Uryu
@u_ribo
@uribo