jpmesh: Utilities for Japanese Mesh Code

Shinya Uryu

2018/3/8

jpmesh

jpmesh

The **jpmesh** is a package that makes it easy to use "regional mesh (i.e. mesh code *JIS X 0410*)" used in Japan from R. Regional mesh is a code given when subdividing Japanese landscape into rectangular subregions by latitude and longitude. Depending on the accuracy of the code, different regional mesh length. By using the same mesh in statistical survey etc., it will become possible to handle the survey results of a large area in the area mesh unit.

jpmesh

In jpmesh, mesh codes and latitude and longitude coordinates are compatible with mesh codes from the first region mesh, which is the standard region mesh, to the quarter regional mesh of the divided region mesh (from 80 km to 125 m). Features include “conversion from latitude and longitude to regional mesh”, “acquisition of latitude and longitude from regional mesh”, “mapping on prefecture unit and leaflet”.

Usage

Install

Available CRAN (`version 1.1.0`), and olso GitHub develop versions.

```
# CRAN
install.packages("jpmesh")
```

```
# the development version from GitHub:
install.packages("remotes")
remotes::install_github("uribo/jpmesh")
```

```
library(jpmesh)
library(sf)
library(leaflet)
```

Convert mesh code to coordinate and vice versa

Return the *latitude* and *longitude* for specifying the mesh range from the mesh code.

```
mesh_to_coords(5133) # 80km
```

lng_center	lat_center	lng_error	lat_error
133.5	34.33333	0.5	0.3333333

```
mesh_to_coords(513377) # 10km
# ...
mesh_to_coords(51337783123) # 125m
```

Convert mesh code to coordinate and vice versa

Find the mesh code within the range from *latitude* and *longitude*.

```
coords_to_mesh(133, 34) # default as 1km meshcode
```

```
## [1] "51330000"
```

```
coords_to_mesh(133, 34, mesh_size = "80km")
```

```
## [1] "5133"
```

```
coords_to_mesh(133, 34, mesh_size = "125m")
```

```
## [1] "51330000111"
```

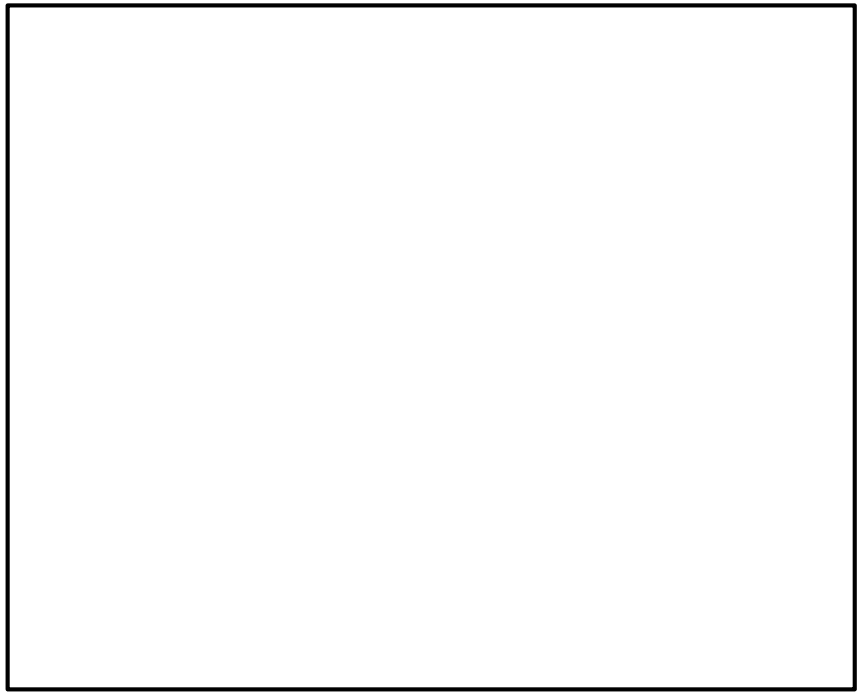
Detect fine and neighborhood mesh codes

```
mesh_80km <- coords_to_mesh(133, 34, "80km")
# Convert to sfc_POLYGON
mesh_polygon <- mesh_80km %>%
  export_mesh()

mesh_polygon
```

```
## Geometry set for 1 feature
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: 133 ymin: 34 xmax: 134 ymax: 34.66667
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs

## POLYGON ((133 34, 134 34, 134 34.66667, 133 34....
```

Detect fine and neighborhood mesh codes

```
mesh_polygon %>%
  st_geometry() %>%
  plot()
```



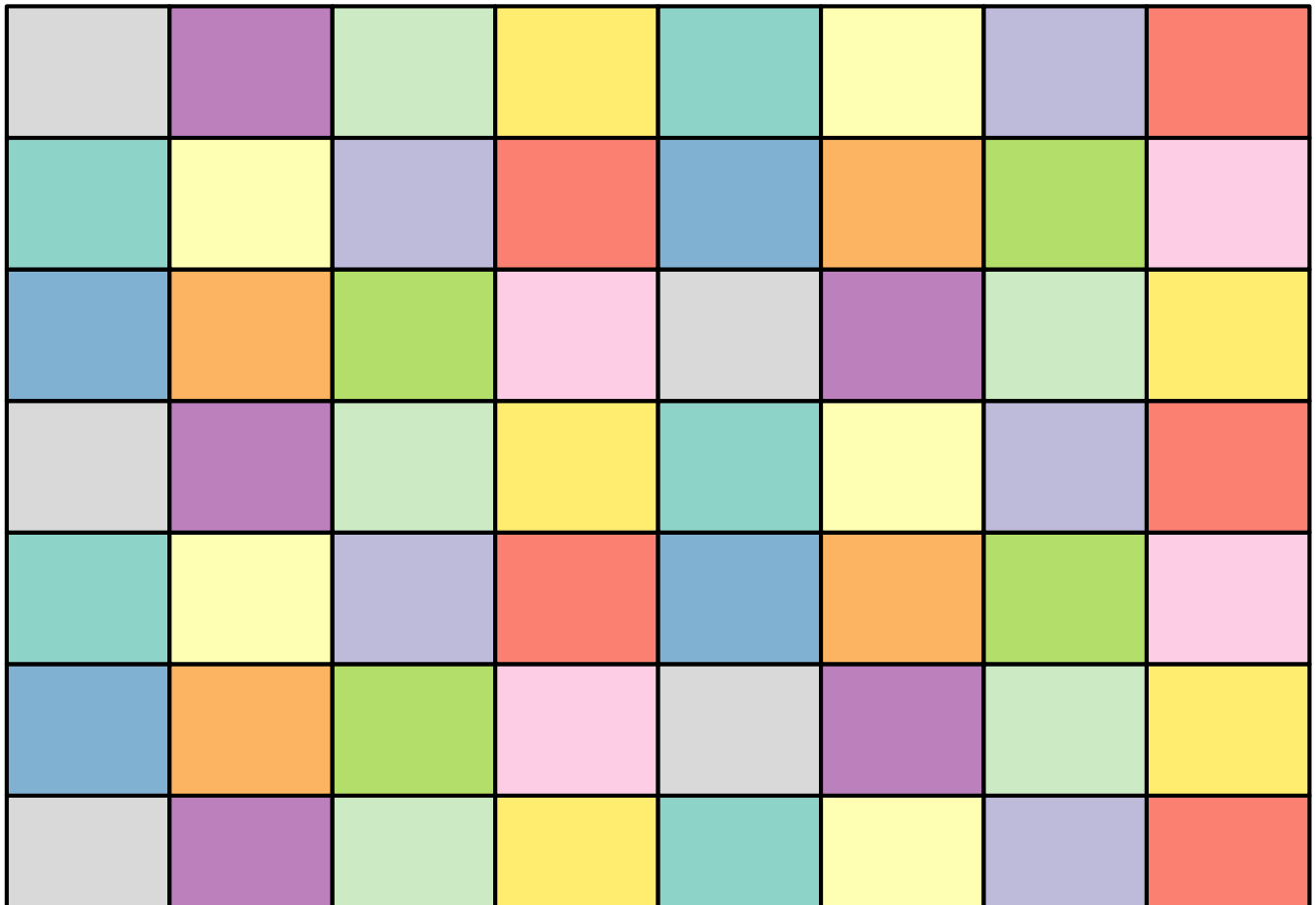
Detect fine and neighborhood mesh codes

```r
# Returns a finer mesh of the area of the mesh codes
# Such as, 80km to 10km mesh codes.
meshes_10km <- mesh_80km %>%
  fine_separate()

meshes_10km
```

```
##  [1] "513300" "513301" "513302" "513303" "513304" "513305" "513306"
##  [8] "513307" "513310" "513311" "513312" "513313" "513314" "513315"
## [15] "513316" "513317" "513320" "513321" "513322" "513323" "513324"
## [22] "513325" "513326" "513327" "513330" "513331" "513332" "513333"
## [29] "513334" "513335" "513336" "513337" "513340" "513341" "513342"
## [36] "513343" "513344" "513345" "513346" "513347" "513350" "513351"
## [43] "513352" "513353" "513354" "513355" "513356" "513357" "513360"
## [50] "513361" "513362" "513363" "513364" "513365" "513366" "513367"
## [57] "513370" "513371" "513372" "513373" "513374" "513375" "513376"
## [64] "513377"
```

Detect fine and neighborhood mesh codes

```
meshes_10km %>%
  export_meshes() %>%
  plot()
```



Detect fine and neighborhood mesh codes

```
# the value of the adjacent mesh codes
coords_to_mesh(133, 34, "80km") %>%
  neighbor_mesh()
```

```
## [1] "5032" "5033" "5034" "5132" "5133" "5134" "5232" "5233" "5234"
```

```
coords_to_mesh(133, 34, "500m") %>%
  neighbor_mesh()
```

Detect fine and neighborhood mesh codes

```
mesh_1km_neighbors <- coords_to_mesh(133, 34, "1km") %>%
  neighbor_mesh()
mesh_1km_neighbors %>%
  export_meshes() %>%
  st_geometry() %>%
  plot()
```

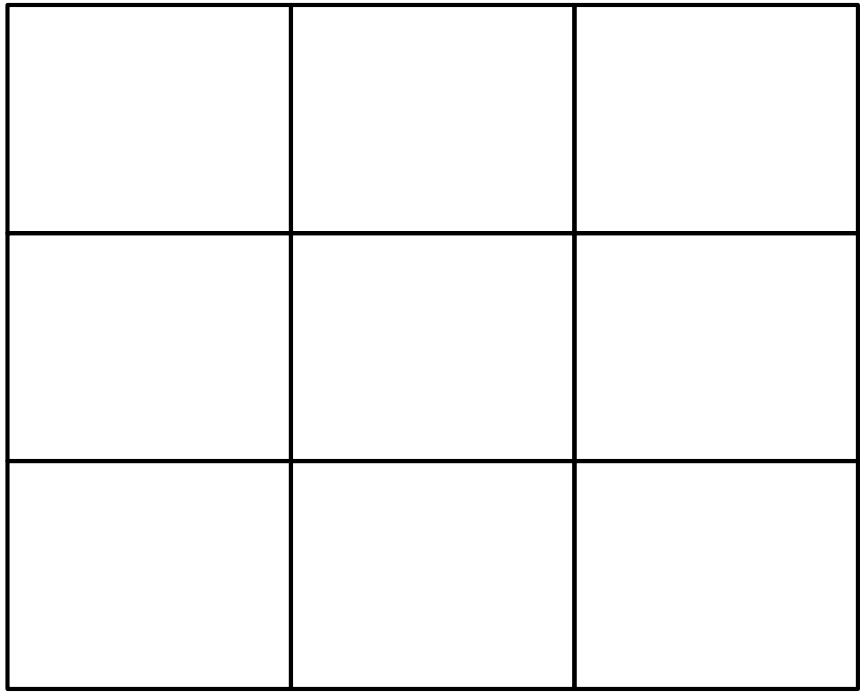


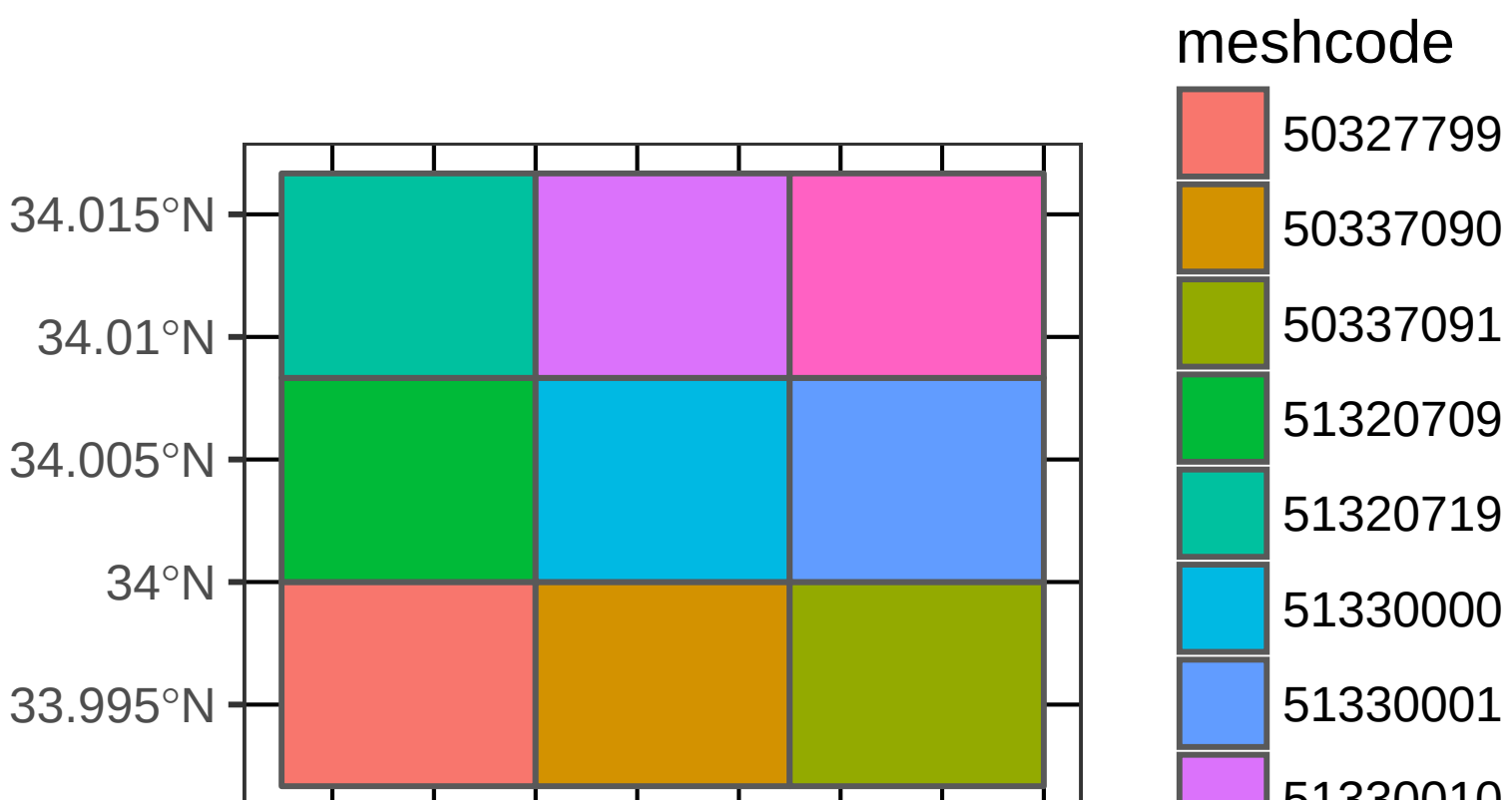
Figure 1: 1km neighborhood meshes

Plots and Visualize

ggplot2

```
remotes::install_github("hadley/ggplot2")

library(ggplot2)
```

```
ggplot() +
  geom_sf(data = mesh_1km_neighbors %>%
  export_meshes(), aes(fill = meshcode)) +
  theme_bw()
```



Administration mesh

```r
set.seed(71)
# Select prefecture or city code
administration_mesh(code = 33, type = "prefecture") %>%
  sample(5)
```

lng_center	lat_center	meshcode	lat_error	geometry
133.8125	34.62500	513376	0.0416667	list(c(133.75, 133.875
133.9375	34.62500	513377	0.0416667	list(c(133.875, 134, 13
133.6875	34.70833	523305	0.0416667	list(c(133.625, 133.75
133.8125	34.70833	523306	0.0416667	list(c(133.75, 133.875
133.9375	34.70833	523307	0.0416667	list(c(133.875, 134, 13
133.6875	34.79167	523315	0.0416667	list(c(133.625, 133.75
133.8125	34.79167	523316	0.0416667	list(c(133.75, 133.875
133.9375	34.79167	523317	0.0416667	list(c(133.875, 134, 13
133.8125	34.87500	523326	0.0416667	list(c(133.75, 133.875
133.9375	34.87500	523327	0.0416667	list(c(133.875, 134, 13
133.8125	34.95833	523336	0.0416667	list(c(133.75, 133.875
133.9375	34.95833	523337	0.0416667	list(c(133.875, 134, 13