

Sample_Maps

Vaibhav Agarwal

2024-10-31

Making Maps

Here is a Rmnd file you can use to make maps!

- sf is the main package we use to make maps
- sf accepts data in many formats – mainly shapefiles
- shape files are downloaded as zipfiles and you must open the entire zip file for it to work :)
- For this example, only .shp for finland is enough, in the future it would be required to use zipfile or it is also possible to save all the files in the same folder

```
# Sf is the main file
library(sf)

## Linking to GEOS 3.11.2, GDAL 3.7.2, PROJ 9.3.0; sf_use_s2() is TRUE

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##     filter, lag

## The following objects are masked from 'package:base':
##     intersect, setdiff, setequal, union

library(ggthemes)

# this is to set the path of your shapefile (must have all other files in the same folder as well)
fin_mun <- read_sf("C:/Users/User/OneDrive - University of Helsinki/Desktop/FIN_adm4.shp")
```

```

#Legend as provided by you; however, very overlapping.
# The normal index (those used in the df are mutually exclusive, and perhaps better)
#Capital region
# Ussimam
# southern finland
# western finland
#northern and eastern finaland
# aland islans

#Feeding all the names in NAMES_2 to chatGPT and getting the classification for the categories you have

# Based on your requirements, you can change this

category_lookup <- list(
  `Capital Region` = c("Eastern Uusimaa"),
  `Uusimaa` = c("Uusimaa"),
  `Southern Finland` = c("Kymenlaakso", "Päijänne Tavastia", "Tavastia Proper", "South Karelia"),
  `Western Finland` = c("Finland Proper", "Satakunta", "Pirkanmaa", "Southern Ostrobothnia", "Ostrobothnia"),
  `Northern and Eastern Finland` = c("Central Finland", "North Karelia", "North Savonia", "Northern Ostrobothnia"),
  `Åland Islands` = c()
)

# Using the stack fucntion to attach the list to make the dictionary into a sort of df
# How stack works:
## example we have
#### column names a,b
#### row1 - 1,2
#### row2 -4,5
#### row3 - 9,3

# stacking will make this
## row1 - 1,a
## row2 - 4,a
## row3 - 9,a
## row4 - 2,b
## row5 - 5,b
## row6 -3,b

#it is the same as converting from wide to long.

category_df <- stack(category_lookup)

## Warning in stack.default(category_lookup): non-vector elements will be ignored

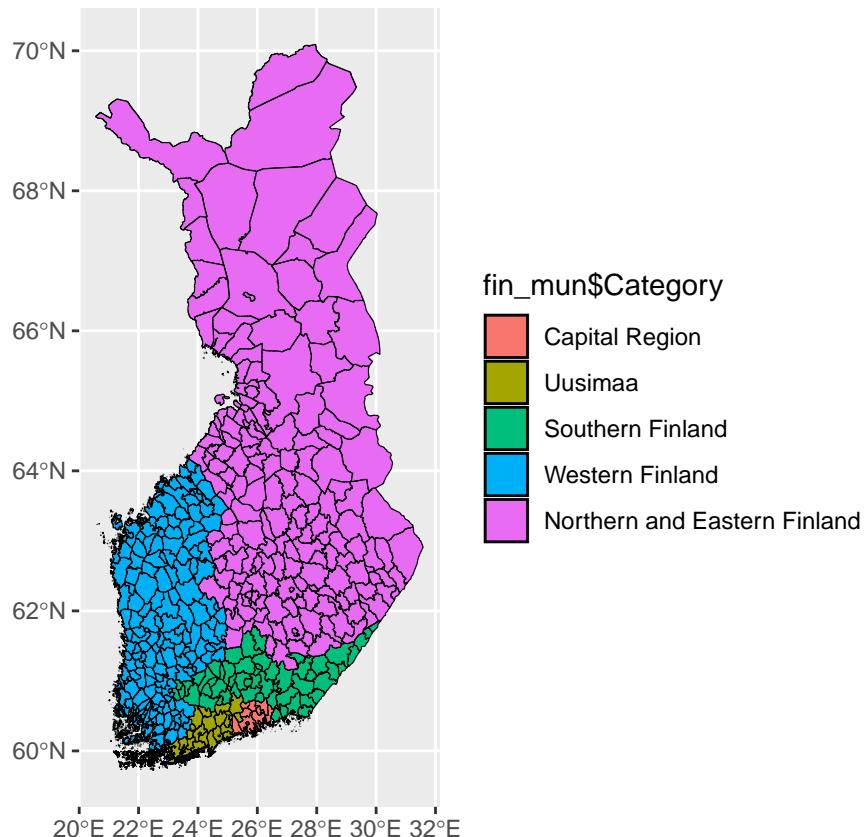
colnames(category_df) <- c("Region", "Category") #renaming

# lets add the column.
# we will do so by merging
# the common column is Name_2 and Region (the rows have same info)
# We merge them together and want only those which merge to be in df, si using all.x = TRUE
fin_mun <- merge(fin_mun, category_df, by.x = "NAME_2", by.y = "Region", all.x = TRUE)

```

```
#Quick Plotting
ggplot() +
  geom_sf(aes(fill = fin_mun$Category),color = 'black',data = fin_mun)
```

Warning: Use of 'fin_mun\$Category' is discouraged.
i Use 'Category' instead.

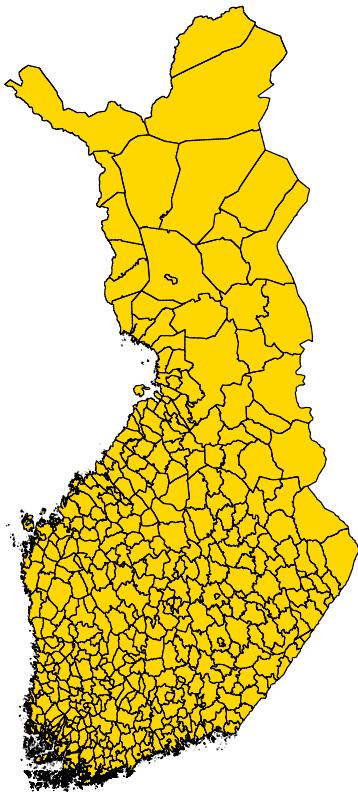


Some quick map making help

Shading only some subsets

*If you want for some instance only to shade southern finland and capital region and leave the rest blank
chose the color of the empty df. by default it is grey. If you dont want that pass this line of code*

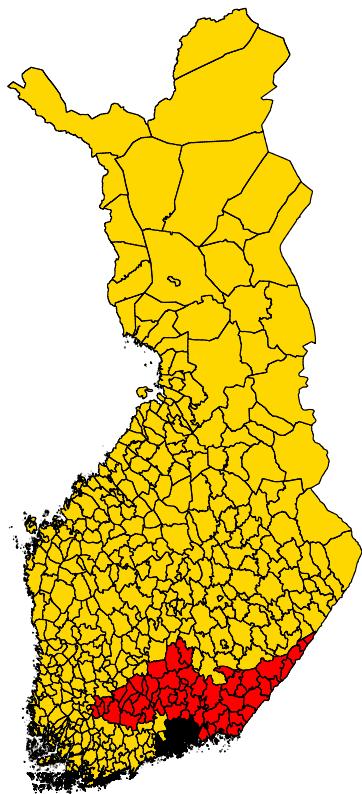
```
ggplot() +
  geom_sf(fill = 'gold',color = 'black',data = fin_mun)+
```



```
# lets say you want to shade capital region and southern finland
# step 1 - make subsets
cap_region <- fin_mun %>%
  filter(Category == 'Capital Region')

south_FI <- fin_mun %>%
  filter(Category == 'Southern Finland')

# After this, overlay these graphs one over the other
# Next, coloring always have sequentially. If you pass the entire df at the end, the whole map will be y
# like this
ggplot() +
  geom_sf(fill = 'gold',color = 'black',data = fin_mun)+#
  geom_sf(fill = 'red',color = 'black',data = south_FI)+#
  geom_sf(fill = 'black',color = 'black',data = cap_region)+#
  theme_void()
```

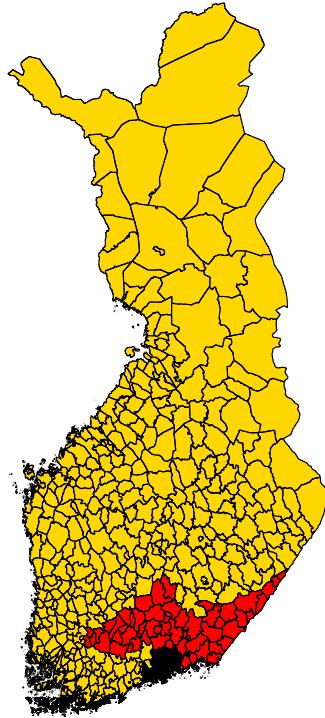


```
# You can also add some title, fommatting and labels to your graph to make it more beautiful!

ggplot() +
  geom_sf(fill = 'gold', color = 'black', data = fin_mun) +
  geom_sf(fill = 'red', color = 'black', data = south_FI) +
  geom_sf(fill = 'black', color = 'black', data = cap_region) +
  theme_void() +
  labs(title = 'Adding colors and shading subsets on maps in R',
       subtitle = 'Entire Finland, Southern Finland, Capital Region') +
  theme(
    plot.title = element_text(color = "black", size = 12, hjust = 0.5),
    plot.subtitle = element_text(color = 'black', size = 8, hjust = 0.5),
    plot.caption = element_text(face = "italic"),
    legend.position = 'none'
  )
```

Adding colors and shading subsets on maps in R

Entire Finland, Southern Finland, Capital Region



Merging this data with a random column and plotting these graphs on a color scale

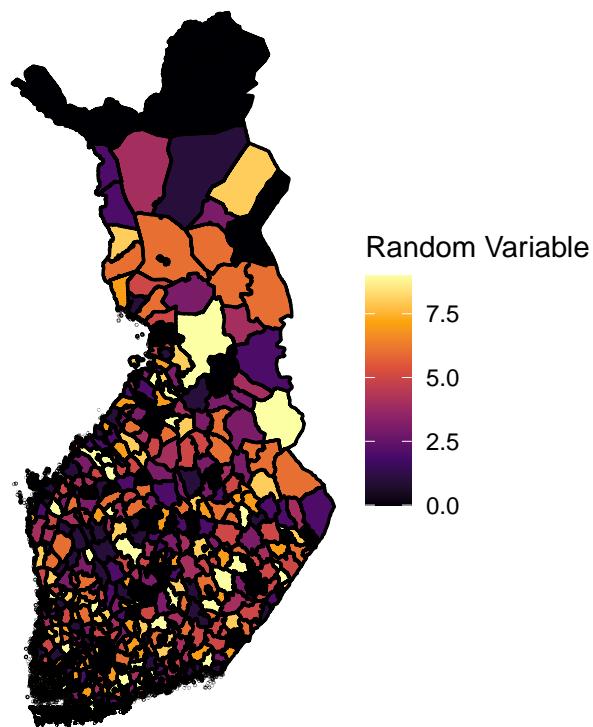
```
# Merge the column based on common place name.  
# For example, if NAME_2 and your df both have 'uusimaa' exactly as it is, then you can merge the 2 add  
# If they are not same 'uusimaa' vs 'Uusimaa', then they will not merge.  
# It is best to first convert to lowercase and check for similarity  
# Then you can do left merge, i.e., everything in df1 will be considered important and those rows wont  
# I will look something like this  
  
###  
#fin_mun <- merge(fin_mun, 'your df', by.x = "your column in geo df", by.y = "freq column", all.x = TRUE)  
###  
  
#For now, I will directly create  
fin_mun$value <- as.integer(runif(nrow(fin_mun), 0, 10))  
  
ggplot() +  
  geom_sf(aes(fill = value), color = 'black', linewidth = 0.6, data = fin_mun) +  
  theme_void() +  
  theme(legend.position = 'right') +  
  scale_fill_viridis_c(option = "inferno") +  
  labs(title = 'Adding color based on frequency',  
       subtitle = '',
```

```

    fill = 'Random Variable')+
theme(
  plot.title = element_text(color="black", size=12, hjust = 0.5),
  plot.subtitle = element_text(color = 'black',size = 8, hjust = 0.5),
  plot.caption = element_text(face="italic"),
  legend.position = 'right')

```

Adding color based on frequency



Combining 1 and 2. Making a zoomed in map with frequency

```

west_FI <- fin_mun %>%
  filter(Category == 'Western Finland')

ggplot() +
  #geom_sf(fill = 'white',color = 'black', linewidth = 0.6,data = fin_mun)+ #adding defualt color to re
  geom_sf(aes(fill = value),color = 'black',linewidth = 0.6,data = west_FI)+ #using south FO file we m
  theme_void()+
  scale_fill_viridis_b(option = 'turbo')+
  theme(legend.position = 'right') +
  labs(title = 'Adding color based on frequency',
       subtitle = 'This also includes subsets',
       fill = 'Random Variable')+
  theme(
    plot.title = element_text(color="black", size=12, hjust = 0.5),

```

```
plot.subtitle = element_text(color = 'black',size = 8, hjust = 0.5),  
plot.caption = element_text(face="italic"),  
legend.position = 'right')
```

Adding color based on frequency

This also includes subsets

