

project

MA Zhuoheng 1930026096

2022/4/27

Installing packages

```
# Necessary packages
pkgs <- c("sf", "dplyr", "magrittr", "gganimate", "lubridate", "magick", "nCov2019", "RColorBrewer", "devtools")

#Installing packages
install.packages(pkgs, lib = "D:/R-4.1.3/library")
devtools::install_github("jl5000/barRacer")
```

Loading packages

```
library(sf)
library(ggplot2)
library(dplyr)
library(magrittr)
library(gganimate)
library(lubridate)
library(magick)
library(nCov2019)
library(RColorBrewer)
library(barRacer)
```

Query data from nCov2019

NOTICE: Using UIC WIFI (any campus WIFI) to run this block would fail. Please run query() function without UIC WIFI (Hotspot is OK!)

```
# Query data from nCov2019 by query()
x=query()
```

```
## last update: 2022-05-18
## Gloabl total 524317031 cases; and 6293182 deaths
## Gloabl total affect country or areas: 229
## Gloabl total recovered cases: 537804
## last update: 2022-05-18
## Total Candidates Programs : 51
## Total Candidates Programs : 84
```

1. Chinese province confirmed case dynamic map

```
# Get the table that contains historical data
his_data <- x$historical$province

#Select province in China and modify data mistake
data <- filter(his_data, country == 'China')
data$province[which(data$province == 'unknown')] <- 'taiwan'
```

```
# Get the date, province name, and confirm case
data_province = data.frame(time=data$date, Yname=data$province, cum_confirm = data$cases)

# Transfer time data type to date
data_province$time = ymd(data_province$time)
```

```
# Load province geometry data
province = read_sf("E:/uic/Y3_0_xia/dataVis/project_dataset/rmap/省面.shp")

# Transfer province name to lower capital
province$Yname <- tolower(province$Yname)

# Modify province name to keep consistency
province$Yname[which(province$Yname == 'neimenggu')] <- 'inner mongolia'
province$Yname[which(province$Yname == 'macao')] <- 'macau'
province$Yname[which(province$Yname == 'hongkong')] <- 'hong kong'
province$Yname[which(province$Yname == 'xizang')] <- 'tibet'
province$Yname[which(province$Yname == 'shangdong')] <- 'shandong'

province
```

```
## Simple feature collection with 34 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:   xmin: -2578822 ymin: 2367106 xmax: 2092054 ymax: 6385320
## Projected CRS: China_Lambert_Conformal_Conic
## # A tibble: 34 x 6
##   DZM   NAME   Yname      省全名      分色      geometry
## * <chr> <chr> <chr>    <chr>    <int>    <MULTIPOLYGON [m]>
## 1 230000 黑龙江 heilongjiang 黑龙江省      1 (((1329152 5619034, 132332~
## 2 650000 新疆   xinjiang   新疆维吾尔自治区      3 (((-1227177 5904428, -1224~
## 3 140000 山西   shanxi     山西省        3 (((761692.1 4443125, 76099~
## 4 640000 宁夏   ningxia    宁夏回族自治区      3 (((146556.3 4728376, 14772~
## 5 540000 西藏   tibet      西藏自治区      1 (((-2189253 4611401, -2187~
## 6 370000 山东   shandong   山东省          0 (((915805.7 4438425, 91755~
## 7 410000 河南   henan      河南省          1 (((915805.7 4438425, 91387~
## 8 320000 江苏   jiangsu     江苏省          1 (((1261146 4381810, 126454~
## 9 340000 安徽   anhui      安徽省          2 (((1016688 4289115, 101885~
## 10 420000 湖北   hubei      湖北省          3 (((547929.6 4087822, 55153~
## # ... with 24 more rows
```

```
data_province$cum_confirm = cut(data_province$cum_confirm, breaks=c(0,1,10,50,100,500,1000,5000,100000),labels=c("0","1-9","10-49","50-99","100-499","500-999","1000-4999",">=5000"),order = TRUE,include.lowest = TRUE, right = TRUE)
```

```
# Generate time array
startTime <- ymd("2020-01-22")
nowTime <- Sys.time()
endTime <- date(nowTime) - ddays(2)
timeLength <- interval(startTime, endTime) %>% time_length("day")
mytime <- startTime + ddays(0:timeLength)
```

```
# Define map theme
mytheme= theme(
plot.title = element_text(face = "bold", hjust = 0.5, color = "black"),
plot.subtitle = element_text(face = "bold", hjust = 0.5, size = 20, color = "red"),
plot.caption = element_text(face = "bold", hjust = 1, color = "black"),
legend.title = element_text(face = "bold", color = "black"),
legend.text = element_text(face = "bold",color = "black"),
legend.background = element_rect(colour = "black"),
legend.key = element_rect(fill = NA),
legend.position = "right",
axis.ticks = element_blank(),
axis.text = element_blank(),
panel.background = element_blank(),
panel.border = element_rect(color = "black", linetype = "solid", size = 1, fill = NA)
)
```

```
# Generate map images in a rate of 30 days
subtime <- c()
for (i in seq(1,length(mytime),30)) {
subtime <- append(subtime,mytime[i])
# Select subset by time array
data_time=subset(data_province,time==mytime[i])
# Merge data by key 'Yname'
province_data=province%>%left_join(data_time,by='Yname')

# Generate map images and save
p=ggplot(province_data)+
geom_sf(aes(fill=cum_confirm))+
coord_sf()+
scale_fill_brewer(palette = "OrRd",direction = 1)+
guides(fill=guide_legend(title = "Confirmed Number",reverse = 1))+
labs(title = "cov-19 Data Visualization",
subtitle=mytime[i],
caption="Data Souce: package nCov2019")+
mytheme
ggsave(filename = paste0(date(mytime[i]),".png"),
plot = p, path = "E:/uic/Y3_0_xia/dataVis/project_dataset/pic",
width = 20, height = 20, units = "cm")
}
```

```
# Generate dynamic map
animate_p=image_animate(image=image_read(path=paste0("E:/uic/Y3_0_xia/dataVis/project_dataset/pic","/",date(subtime),".png")))
anim_save(filename = "疫情地图可视化动态图.gif",animation = animate_p,path="E:/uic/Y3_0_xia/dataVis/project_dataset/animate",fps=0.0001)
```

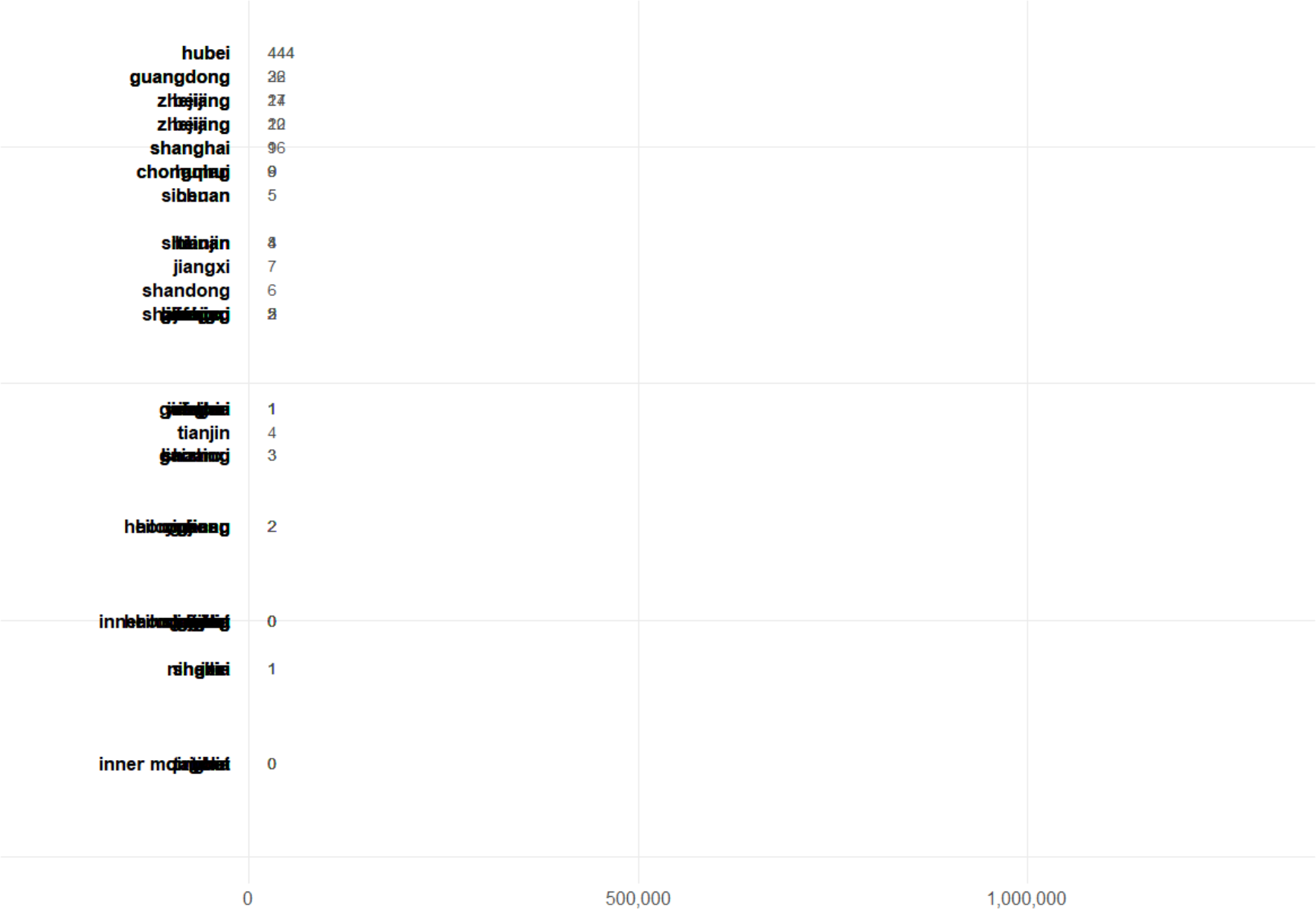
2. Chinese Province Confirmed Case Dynamic Horizontal Bar Chart

```
# Get the plot data
data_province = data.frame(time=data$date, Yname=data$province, cum_confirm = data$cases)
data_province$time = ymd(data_province$time)
```

```
# plot bar chart and save as .gif
bar_chart_race(data_province,
               cat_col = Yname,
               val_col = cum_confirm,
               time_col = time,
               maxBars = 34,
               duration = 30,
               title = "China COVID19 Data")
```

China COVID19 Data

2020-01-22



```
gganimate::anim_save("COVID19_Bar.gif")
```