MAP

ryang273

25/11/2020

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(sf)
```

```
## Linking to GEOS 3.7.2, GDAL 2.4.2, PROJ 5.2.0
```

```r
library(tmap)
library(grid)
data("World")
Deaths <- read_csv("/Users/yangruiqin/Desktop/2864/time_series_covid19_deaths_global.csv")
```

```
##
## -- Column specification -------------------------------------------------------------------------------
## cols(
##   .default = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```r
Confirmed <- read_csv("/Users/yangruiqin/Desktop/2864/time_series_covid19_confirmed_global.csv")
```

```
##
## -- Column specification -------------------------------------------------------------------------------
## cols(
##   .default = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```r
Recovered <- read_csv("/Users/yangruiqin/Desktop/2864/time_series_covid19_recovered_global.csv")
```

```
##
## -- Column specification -------------------------------------------------------------------------------
## cols(
##   .default = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
Deaths_data <- Deaths %>%
  group_by(`Country/Region`) %>%
  summarise(Deaths = sum(`11/22/20`, na.rm =TRUE)) %>%
  rename(Region = `Country/Region`)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Confirmed_data <- Confirmed %>%
  group_by(`Country/Region`) %>%
  summarise(Confirmed = sum(`11/22/20`, na.rm = TRUE)) %>%
  rename(Region = `Country/Region`)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Recovered_data <- Recovered %>%
  group_by(`Country/Region`) %>%
  summarise(Recovered = sum(`11/22/20`, na.rm = TRUE)) %>%
  rename(Region = `Country/Region`)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Comb_deaths <- Deaths_data %>%
  mutate(Region = replace(Region, Region == "US", "United States"))%>%
  group_by(Region) %>%
  summarise(Deaths = sum(Deaths, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Comb_confirmed <- Confirmed_data %>%
  mutate(Region = replace(Region, Region == "US", "United States"))%>%
  group_by(Region) %>%
  summarise(Confirmed = sum(Confirmed, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

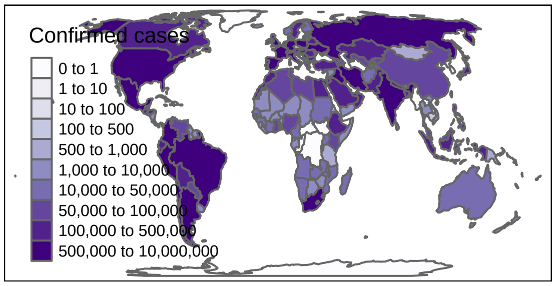
```
Comb_recovered <- Recovered_data %>%
  mutate(Region = replace(Region, Region == "US", "United States"))%>%
  group_by(Region) %>%
  summarise(Recovered = sum(Recovered, na.rm = TRUE)) %>%
  ungroup()
```

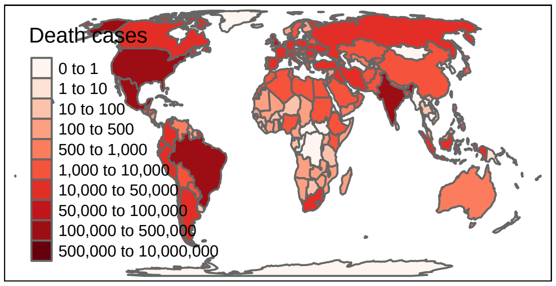
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

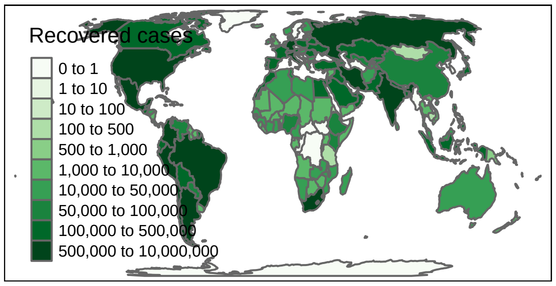
```
comb <-Comb_confirmed %>%
  left_join(Comb_deaths, by = "Region")%>%
  left_join(Comb_recovered, by ="Region")
cov_world <- left_join(World, comb, by = c("name" = "Region")) %>%
  replace_na(list(Confirmed=0, Deaths=0, Recovered=0))
mybreaks<- c(0,1, 10,100, 500, 1000,10000,50000,100000,500000,10000000)
p1 <- tm_shape(cov_world) +
```

```r
  tm_polygons(col="Deaths", breaks=mybreaks, title="Death cases", palette="Reds") +
  tm_legend(position=c("left", "centre"))
mybreaks<- c(0,1, 10,100, 500, 1000,10000,50000,100000,500000,10000000)
p2 <- tm_shape(cov_world) +
  tm_polygons(col="Confirmed", breaks=mybreaks, title="Confirmed cases", palette="Purples") +
  tm_legend(position=c("left", "centre"))
mybreaks<- c(0,1, 10,100, 500, 1000,10000,50000,100000,500000,10000000)
p3 <- tm_shape(cov_world) +
  tm_polygons(col="Recovered", breaks=mybreaks, title="Recovered cases", palette="Greens") +
  tm_legend(position=c("left", "centre"))
multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL) {
  require(grid)

  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  if (is.null(layout)) {
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                 ncol = cols, nrow = ceiling(numPlots/cols))
}

if (numPlots == 1) {
print(plots[[1]])

} else {
grid.newpage()
pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

for (i in 1:numPlots) {
  matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

  print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                  layout.pos.col = matchidx$col))
 }
}
 }
multiplot(p2, p1,p3, cols = 1)
```

```
## Warning: Values have found that are higher than the highest break
```


Confirmed cases
0 to 1
1 to 10
10 to 100
100 to 500
500 to 1,000
1,000 to 10,000
10,000 to 50,000
50,000 to 100,000
100,000 to 500,000
500,000 to 10,000,000


Death cases
0 to 1
1 to 10
10 to 100
100 to 500
500 to 1,000
1,000 to 10,000
10,000 to 50,000
50,000 to 100,000
100,000 to 500,000
500,000 to 10,000,000


Recovered cases
0 to 1
1 to 10
10 to 100
100 to 500
500 to 1,000
1,000 to 10,000
10,000 to 50,000
50,000 to 100,000
100,000 to 500,000
500,000 to 10,000,000