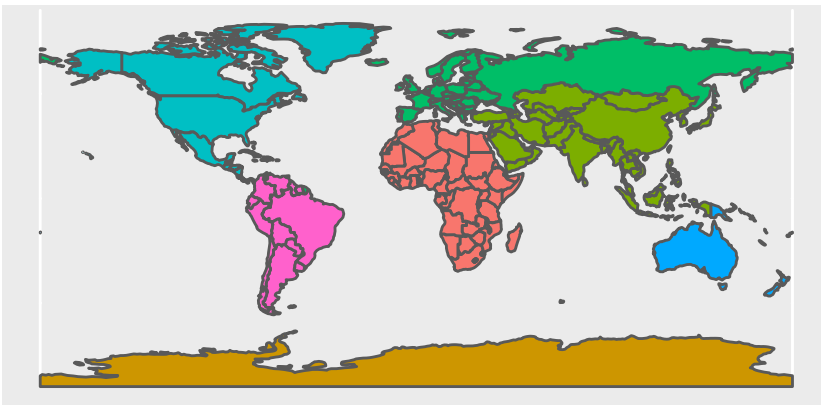


tallest-and-widest.R

cschmertmann

2021-11-15

```
#.....  
# Carl Schmertmann  
# 15 Nov 2021  
#  
# find N-S and E-W extremes for countries  
# .....  
  
library(spData)  
  
## To access larger datasets in this package, install the spDataLarge  
## package with: `install.packages('spDataLarge',  
## repos='https://nowosad.github.io/drat/', type='source')`  
  
library(sf)  
  
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1  
  
library(tidyverse)  
  
## -- Attaching packages ----- tidyverse 1.3.1 --  
## v ggplot2 3.3.5      v purrr  0.3.4  
## v tibble  3.1.4      v dplyr  1.0.7  
## v tidyr   1.1.4      v stringr 1.4.0  
## v readr   2.0.2      v forcats 0.5.1  
  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()  
  
# plot world map just to verify data  
ggplot(data=world) +  
  geom_sf(aes(fill=continent))
```



continent



```
# break multi-polygon countries into polygons. For the "clump"
# countries we'll paste the single polygons back together.
# For all others we'll keep the LARGEST contiguous
# polygon only. This will
# chop off Svalbard, Alaska+Hawaii, Argentine Antarctica, etc...
```

```
clump = c('Chile', 'Japan', 'New Zealand', 'Australia', 'Indonesia', 'Philippines')
```

```
world2 = world %>%
  filter(type %in% c('Country', 'Sovereign country')) %>%
  st_cast(., 'POLYGON') %>%
  mutate(subarea = st_area(geom)) %>%
  arrange(name_long, -subarea) %>%
  group_by(name_long) %>%
  mutate(i=seq(name_long)) %>%
  filter( (i==1) | (name_long %in% clump)) %>%
  summarize(glued_back_together = TRUE)
```

```
## Warning in st_cast.sf(., "POLYGON"): repeating attributes for all sub-geometries
## for which they may not be constant
```

```
# calculate the bounding box for each (possibly chopped) country
box = sapply(world2$geom, st_bbox) %>%
  t() %>%
  as_tibble() %>%
  add_column(name = world2$name_long, .before = 1)
```

```

this_crs = st_crs(world)

# add the corners of the bounding boxes as variables
box$NW_corner = st_as_sf(box, coords=c('xmin', 'ymax'))$geometry
box$SW_corner = st_as_sf(box, coords=c('xmin', 'ymin'))$geometry
box$SE_corner = st_as_sf(box, coords=c('xmax', 'ymin'))$geometry
box$NE_corner = st_as_sf(box, coords=c('xmax', 'ymax'))$geometry

st_crs(box$NW_corner) = this_crs
st_crs(box$SW_corner) = this_crs
st_crs(box$SE_corner) = this_crs
st_crs(box$NE_corner) = this_crs

# calculate the "height" of each country in km
box_height = sapply(1:nrow(box), function(k) {
  st_distance(box$SW_corner[k], box$NW_corner[k]) / 1000
})

names(box_height) = world2$name_long

# 20 tallest

tmp = box_height %>% sort() %>% rev() %>% head(20)

tibble( name=names(tmp), height_km = round(tmp))

## # A tibble: 20 x 2
##   name                      height_km
##   <chr>                     <dbl>
## 1 Brazil                    4338
## 2 Chile                     4229
## 3 Russian Federation       4064
## 4 China                     3689
## 5 Australia                3666
## 6 Argentina                3393
## 7 Canada                   3363
## 8 India                     3061
## 9 United States            2703
## 10 Greenland                2625
## 11 Democratic Republic of the Congo 2059
## 12 Myanmar                  2046
## 13 Peru                     2034
## 14 Mexico                   2022
## 15 Algeria                  2008
## 16 Colombia                 1861
## 17 Mozambique               1826
## 18 Chad                     1778
## 19 Indonesia                1761
## 20 Saudi Arabia            1758

# Sweden, Chile, and Norway
tmp = box_height[c('Sweden', 'Norway', 'Chile', 'Finland')] %>% sort() %>% rev()

```

```

tibble( name=names(tmp), height_km = round(tmp))

## # A tibble: 4 x 2
##   name      height_km
##   <chr>         <dbl>
## 1 Chile         4229
## 2 Sweden        1528
## 3 Norway        1457
## 4 Finland       1147

# "WIDTH" still needs work... depends on latitudes
# calculate the "width" of each country in km
if (FALSE) {

box_width = sapply(1:nrow(box), function(k) {
  st_distance(box$SW_corner[k], box$SE_corner[k]) / 1000
})

names(box_width) = world2$name_long

# 20 widest

tmp = box_width %>% sort() %>% rev() %>% head(20)

tibble( name=names(tmp), width_km = round(tmp))

}

```