map.widener

widener

10/15/2020

#loading library

```
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(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.4     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

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

```
library(tidylog)
```

```
##
## Attaching package: 'tidylog'

## The following objects are masked from 'package:tidyr':
##
##     drop_na, fill, gather, pivot_longer, pivot_wider, replace_na,
##     spread, uncount
## The following objects are masked from 'package:dplyr':
##
##     add_count, add_tally, anti_join, count, distinct, distinct_all,
##     distinct_at, distinct_if, filter, filter_all, filter_at, filter_if,
##     full_join, group_by, group_by_all, group_by_at, group_by_if,
##     inner_join, left_join, mutate, mutate_all, mutate_at, mutate_if,
##     relocate, rename, rename_all, rename_at, rename_if, rename_with,
##     right_join, sample_frac, sample_n, select, select_all, select_at,
##     select_if, semi_join, slice, slice_head, slice_max, slice_min,
##     slice_sample, slice_tail, summarise, summarise_all, summarise_at,
##     summarise_if, summarize, summarize_all, summarize_at, summarize_if,
##     tally, top_frac, top_n, transmute, transmute_all, transmute_at,
##     transmute_if, ungroup
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(sf)
```

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

```r
library(raster)
```

```
## Loading required package: sp
```

```
##
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:tidylog':
##
##     select
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
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(tmap)
library(leaflet)
library(cartogram)
library(readr)
```

#load data

```
key_crop_yields <- readr::read_csv(’https://raw.githubusercontent.com/rfordatascience/tidytuesday/maste
```

```
##
## -- Column specification --------------------------------------------------------------------
## cols(
##   Entity = col_character(),
##   Code = col_character(),
##   Year = col_double(),
##   ‘Wheat (tonnes per hectare)‘ = col_double(),
##   ‘Rice (tonnes per hectare)‘ = col_double(),
##   ‘Maize (tonnes per hectare)‘ = col_double(),
##   ‘Soybeans (tonnes per hectare)‘ = col_double(),
##   ‘Potatoes (tonnes per hectare)‘ = col_double(),
##   ‘Beans (tonnes per hectare)‘ = col_double(),
##   ‘Peas (tonnes per hectare)‘ = col_double(),
##   ‘Cassava (tonnes per hectare)‘ = col_double(),
##   ‘Barley (tonnes per hectare)‘ = col_double(),
##   ‘Cocoa beans (tonnes per hectare)‘ = col_double(),
##   ‘Bananas (tonnes per hectare)‘ = col_double()
## )
```

#load world data frame

```
worlddf <- world
```

#clean/ strip down data

```
avoid_entity_names <- key_crop_yields %>%
  filter(is.na(Code) | Entity == ’World’) %>%
  distinct(Entity)
```

```
## filter: removed 11,098 rows (85%), 1,977 rows remaining
```

```
## distinct: removed 1,941 rows (98%), 36 rows remaining
```

```
key_crop_cleaned <- key_crop_yields %>%
  filter(Year == max(Year)) %>%
  anti_join(avoid_entity_names) %>%
  group_by(Entity,
           Code,
           Year) %>%
  pivot_longer(cols = -c(’Entity’,
                         ’Code’,
                         ’Year’),
               names_to = ’Crop’) %>%
    ungroup() %>%
  mutate(Crop = gsub(’ \\(tonnes per hectare\\)’, ’’, Crop)) %>%
  filter(Crop == ’Potatoes’) %>%
  mutate(Entity = case_when(Entity == ’Democratic Republic of Congo’ ~ ’Democratic Republic of the Cong
                            Entity == ’Russia’ ~ ’Russian Federation’,
                            Entity == ’Timor’ ~ ’Timor-Leste’,
                            Entity == "Cote d’Ivoire" ~ "Côte d’Ivoire",
                            Entity == ’Congo’ ~ ’Republic of the Congo’,
                            Entity == ’Swaziland’ ~ ’eSwatini’,
                            Entity == ’Gambia’ ~ ’The Gambia’,
                            Entity == ’Laos’ ~ ’Lao PDR’,
                            Entity == ’North Korea’ ~ ’Dem. Rep. Korea’,
                            Entity == ’South Korea’ ~ ’Republic of Korea’,
                            Entity == ’Brunei’ ~ ’Brunei Darussalam’,
                            TRUE ~ Entity))
```

```
## filter: removed 12,835 rows (98%), 240 rows remaining

## Joining, by = "Entity"

## anti_join: added no columns

##            > rows only in x   208

##            > rows only in y  (  4)

##            > matched rows    ( 32)

##            >                 =====

##            > rows total       208

## group_by: 3 grouping variables (Entity, Code, Year)

## pivot_longer: reorganized (Wheat (tonnes per hectare), Rice (tonnes per hectare), Maize (tonnes per

## ungroup: no grouping variables

## mutate: changed 2,288 values (100%) of ’Crop’ (0 new NA)

## filter: removed 2,080 rows (91%), 208 rows remaining

## mutate: changed 11 values (5%) of ’Entity’ (0 new NA)
```

#Remove countries w/o data for our crops of interest (no potato)

```r
country_names <- dplyr::select(key_crop_cleaned, Entity)

null_countries <- worlddf %>%
  mutate(value = NA,
         Crop_Percentile = 0) %>%
  anti_join(country_names,
            by = c('name_long' = 'Entity')) %>%
  filter(iso_a2 != 'AQ')
```

```
## mutate: new variable 'value' (logical) with one unique value and 100% NA
##         new variable 'Crop_Percentile' (double) with one unique value and 0% NA
## anti_join: added no columns
##            > rows only in x     7
##            > rows only in y  ( 38)
##            > matched rows    (170)
##            >                 =====
##            > rows total         7
## filter: removed 3 rows (43%), 4 rows remaining
```

```r
na_countries <- key_crop_cleaned %>%
  dplyr::select(Entity,
                value) %>%
  filter(is.na(value)) %>%
  mutate(Crop_Percentile = 0) %>%
  inner_join(world,
             by = c('Entity' = 'name_long'))
```

```
## filter: removed 159 rows (76%), 49 rows remaining
## mutate: new variable 'Crop_Percentile' (double) with one unique value and 0% NA
## inner_join: added 10 columns (iso_a2, continent, region_un, subregion, type, ...)
##             > rows only in x  ( 25)
##             > rows only in y  (153)
##             > matched rows      24
##             >                 =====
##             > rows total        24
```

```
#na_null_countries <- ...(null_countries, na_countries)

percentile_countries <- key_crop_cleaned %>%
  dplyr::select(Entity,
                value) %>%
  filter(!is.na(value)) %>%
  mutate(Crop_Percentile = ntile(value, 100) / 100) %>%
  inner_join(world,
             by = c(’Entity’ = ’name_long’))
```

```
## filter: removed 49 rows (24%), 159 rows remaining

## mutate: new variable ’Crop_Percentile’ (double) with 100 unique values and 0% NA

## inner_join: added 10 columns (iso_a2, continent, region_un, subregion, type, ...)

##              > rows only in x  ( 13)

##              > rows only in y  ( 31)

##              > matched rows     146

##              >                 =====

##              > rows total       146
```

```
full_country_df <-
  bind_rows(percentile_countries, null_countries, null_countries)

full_country_buckets <- full_country_df %>%
  mutate(Buckets = ifelse(
    Crop_Percentile == 0,
    ’NA’,
    ifelse(
      Crop_Percentile > 0 & Crop_Percentile <= 0.25,
      ’1-25th’,
      ifelse(
        Crop_Percentile > 0.25 & Crop_Percentile <= 0.5,
        ’25-50th’,
        ifelse(
          Crop_Percentile > 0.5 & Crop_Percentile <= 0.75,
          ’50-75th’,
          ifelse(Crop_Percentile > 0.75 &
                   Crop_Percentile <= 1, ’75-100th’, NA)
        )
      )
    )
  ))
```

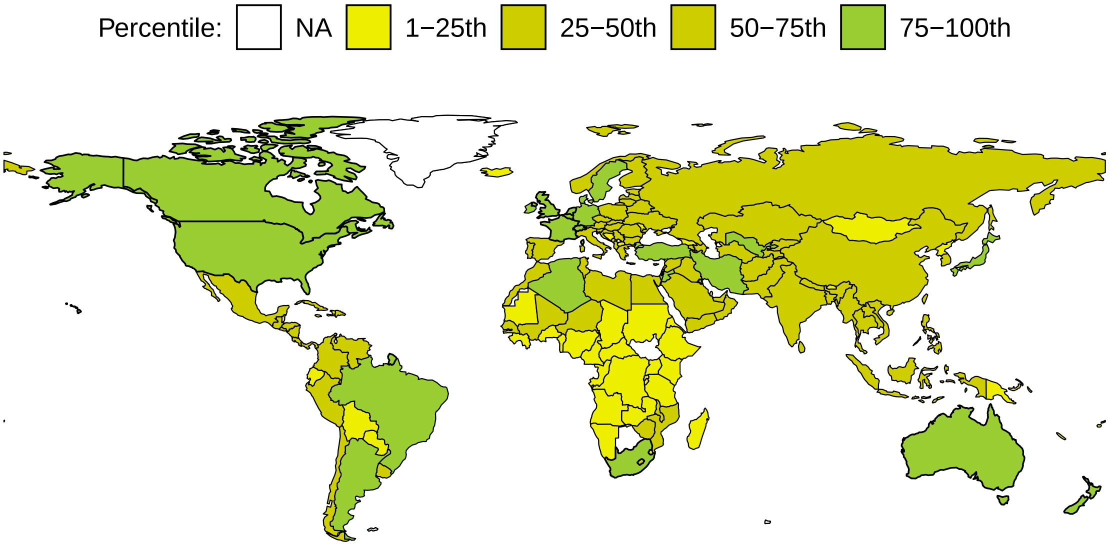
```
## mutate: new variable ’Buckets’ (character) with 5 unique values and 0% NA
```

```
full_country_buckets$Buckets <-
  factor(full_country_buckets$Buckets, levels = unique(c(
    'NA', '1-25th', '25-50th', '50-75th', '75-100th'
  )))
```

make the actual map!!!

```
ggplot(full_country_buckets) +
  geom_sf(aes(geometry = geom,
              fill = Buckets),
          color = 'black',
          size = 0.2) +
  geom_sf(data = subset(full_country_buckets, Crop_Percentile >= 0.9),
          mapping = aes(geometry = geom,
                        fill = Buckets),
          color = 'black',
          size = 0.3) +
  labs(title = 'Potato Yields Around the World',
       caption = 'Souces: Tidytuesday & Our World in Data',
       fill = 'Percentile:') +
  scale_fill_manual(values = c('white', 'yellow2', 'yellow3', 'yellow3', 'yellowgreen')) +
  theme(plot.title = element_text(face = 'bold', size = 25),
        legend.position = 'top',
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 10),
        plot.subtitle = element_text(size = 10),
        plot.caption = element_text(size = 10),
        axis.title = element_blank(),
        axis.ticks = element_blank(),
        axis.text = element_blank(),
        strip.text = ggplot2::element_text(size = 12, hjust = 0, face = 'bold', color = 'black'),
        strip.background = element_rect(fill = NA),
        panel.background = ggplot2::element_blank(),
        axis.line = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.major.x = element_blank()
  )
```

Potato Yields Around the World



Souces: Tidytuesday & Our World in Data