

Refugee Pop. 2019 - Climate Morality 2030 3D World Maps

Maxwell Reikosky

12/10/2020

```
# allowing r chunks to knit without printing 3D models that cause error
knitr::opts_chunk$set(eval = F)

# may need to run developer tools as administrator, commenting code below
# install.packages("devtools")
# devtools::install_github("tylormorganwall/rayshader")

# loading relevant packages
library(readxl)
library(dplyr)
library(ggplot2)
library(ggmap)
library(rayshader)
library(tidyverse)
library(rlang)
library(rgl)
require(devtools)
library(tinytex)
```

1. Importing and Cleaning Refugee Pop. Dataset

```
# must set working directory to MaxwellReikosky_CompToolsFinal folder
# loading refugee population by country of asylum dataset
Refugee_totals <- read_excel("Data/WORLD BANKAPI_SM.POP.REFG_DS2_en_excel_v2_1743495.xls")

# renaming relevant column as a workable string
Refugee_totals <- Refugee_totals %>%
  rename("year_2019" = "2019")

# subsetting refugee dataset to most recent measurement
Refugee_totals <- Refugee_totals %>%
  select(Country_name, year_2019) %>%

# trimming white space
mutate(Country_name = str_trim(Country_name),
       year_2019 = str_trim(year_2019)) %>%

# omitting all countries that were not measured
na.omit()

# recoding country names to match map.world df
# needed escape symbol for recoding Cote d'Ivoire
```

```
Refugee_totals$Country_name <- recode(Refugee_totals$Country_name,
  'United States' = 'USA',
  'United Kingdom' = 'UK',
  'Russian Federation' = 'Russia',
  'Congo, Dem. Rep.' = 'Democratic Republic of the Congo',
  'Congo, Rep.' = 'Republic of Congo',
  'Korea, Rep.' = 'North Korea',
  'Cote d'Ivoire' = 'Ivory Coast',
  'Egypt, Arab Rep.' = 'Egypt',
  'Venezuela, RB' = 'Venezuela',
  'Yemen, Rep.' = 'Yemen',
  'Syrian Arab Republic' = 'Syria',
  'Slovak Republic' = 'Slovakia',
  'Kyrgyz Republic' = 'Kyrgyzstan',
  'North Macedonia' = 'Macedonia',
  'Bahamas, The' = 'Bahamas',
  'Gambia, The' = 'Gambia',
  'Iran, Islamic Rep.' = 'Iran',
  'West Bank and Gaza' = 'Gaza Strip')
```

2. Importing World Map Dataset

```
# grabbing world map df
map.world <- map_data("world")

# checking how many countries are listed on world map dataset
map.world %>%
  select(region) %>%
  unique()
```

3. Joining World Map and Refugee Pop. Datasets

```
# merging map and recoded refugee pop datasets
Refugee_map <- left_join(map.world, Refugee_totals, by = c('region' = 'Country_name'))

# converting year_2019 to numeric because some were in character type
Refugee_map$year_2019 <- as.numeric(Refugee_map$year_2019)
```

4. Importing and Cleaning Climate Vulnerability Dataset

```
# loading climate vulnerability excel data set
Climate_totals <- read_excel("Data/Climate_totals.xls")

# Renaming climate mortality column to workable string
Climate_totals <- Climate_totals %>%
  rename("Climate_mortality" = "Mortality_Climate_total_2030 (Number of People)")

# recoding country names to fit map.world dataset
Climate_totals$Country_name <- recode(Climate_totals$Country_name,
  'United States' = 'USA',
  'United Kingdom' = 'UK',
  'Congo' = 'Republic of Congo',
```

```

        'Cote d'Ivoire' = 'Ivory Coast',
        'DR Congo' = 'Democratic Republic of the Congo',
        'Sudan/South Sudan1' = 'South Sudan',
        'Sudan/South Sudan2' = 'Sudan')

# subsetting to select only 2030 climate mortality column
# clearing any potential white space
Climate_totals <- Climate_totals %>%
  select(Country_name, Climate_mortality) %>%
  mutate(Country_name = str_trim(Country_name),
         Climate_mortality = str_trim(Climate_mortality))

```

5. Joining Climate Mortality With Previous Merged Dataset

```

# merging trimmed climate mortality dataset with previously merged dataset
Refclim_mapdf <- left_join(Refugee_map, Climate_totals, by = c('region' = 'Country_name'))

# coercing climate mortality data to numerical because some were characters
Refclim_mapdf$Climate_mortality <- as.numeric(Refclim_mapdf$Climate_mortality)

```

6. Refugee Population by Country 2019 3D Model

```

# plotting refugee population by country
refpop_gg <- ggplot(Refugee_map, aes(long, lat, group = group, fill = year_2019)) +

# allowing flexible plot shape to match country borders
  geom_polygon() +

# coloring countries on a chosen gradient for value range
  scale_fill_gradient(low = "seashell", high = "darkgreen") +

# labeling title and x & y axes
  labs(title = "Refugee Pop. 2019") +
  xlab("Long.") +
  ylab("Lat.") +

# removing legend title and setting background type
  theme(legend.title = element_blank(),
        panel.background = element_rect())

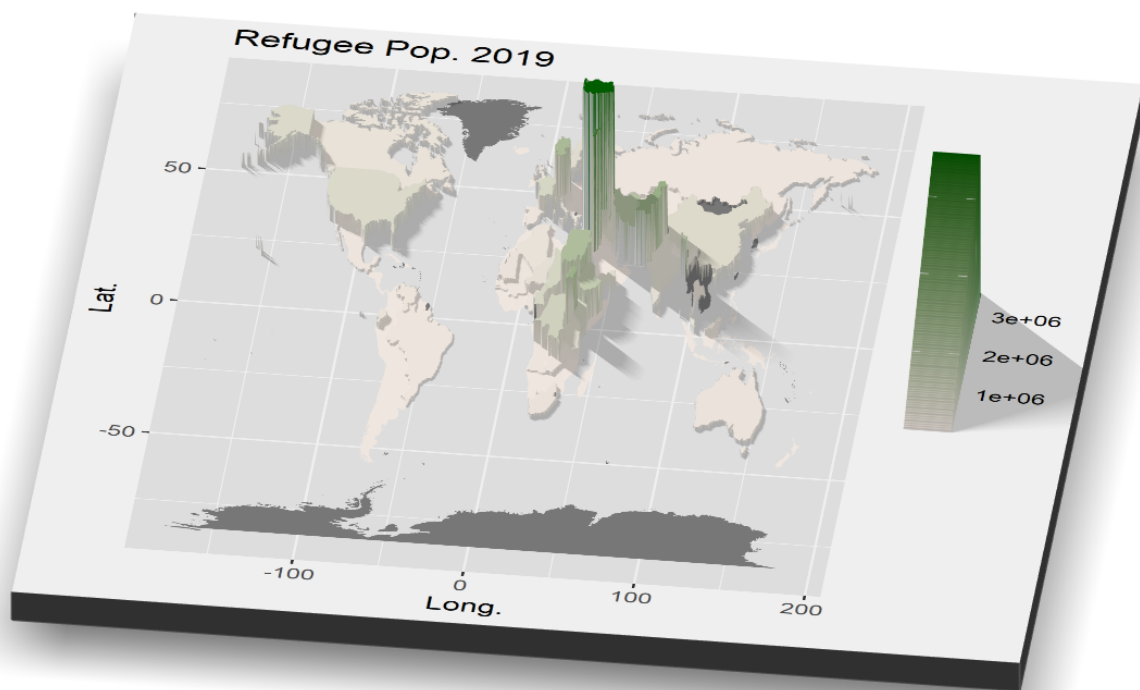
# rendering ggplot as 3D in rayshader, setting parameters and scale
plot_gg(refpop_gg, width = 5, height = 5, raytrace = FALSE, preview = TRUE) +
  plot_gg(refpop_gg, width = 5, height = 5, multicore = TRUE, scale = 250,

# establishing tilt of model and angle of snapshot perspective
        zoom = 0.5, theta = 7, phi = 35, window_size = c(1200, 800)) +

# allowing process launched from R to read input files before execution resumes
  Sys.sleep(0.2) +

# constructing snapshot of final 3D model
  render_snapshot(clear = TRUE)

```



7. Refugee Population by Country 2019 3D Model (sans Turkey & Jordan)

```
# filtering Turkey and Jordan to highlight other standout countries
refpop_sans <- Refugee_map %>%
  filter(region != "Turkey",
         region != "Jordan")

# plotting refugee population by country without Turkey and Jordan
refpop_sansgg <- ggplot(refpop_sans, aes(long, lat, group = group, fill = year_2019)) +

# allowing flexible plot shape to match country borders
  geom_polygon() +

# coloring countries on a chosen gradient for value range
  scale_fill_gradient(low = "seashell", high = "darkgreen") +

# labeling title and x & y axes
  labs(title = "Refugee Pop. 2019 (sans Turkey & Jordan)" +
       xlab("Long.") +
       ylab("Lat.")) +

# removing legend title and setting background type
  theme(legend.title = element_blank(),
        panel.background = element_rect())

# rendering ggplot as 3D in rayshader, setting parameters and scale
plot_gg(refpop_sansgg, width = 5, height = 5, raytrace = FALSE, preview = TRUE) +
plot_gg(refpop_sansgg, width = 5, height = 5, multicore = TRUE, scale = 250,

# establishing angle of snapshot perspective
  zoom = 0.5, theta = 7, phi = 35, windowsize = c(1200, 800)) +

# allowing process launched from R to read input files before execution resumes
  Sys.sleep(0.2) +

# constructing snapshot of final 3D model
  render_snapshot(clear = TRUE)
```

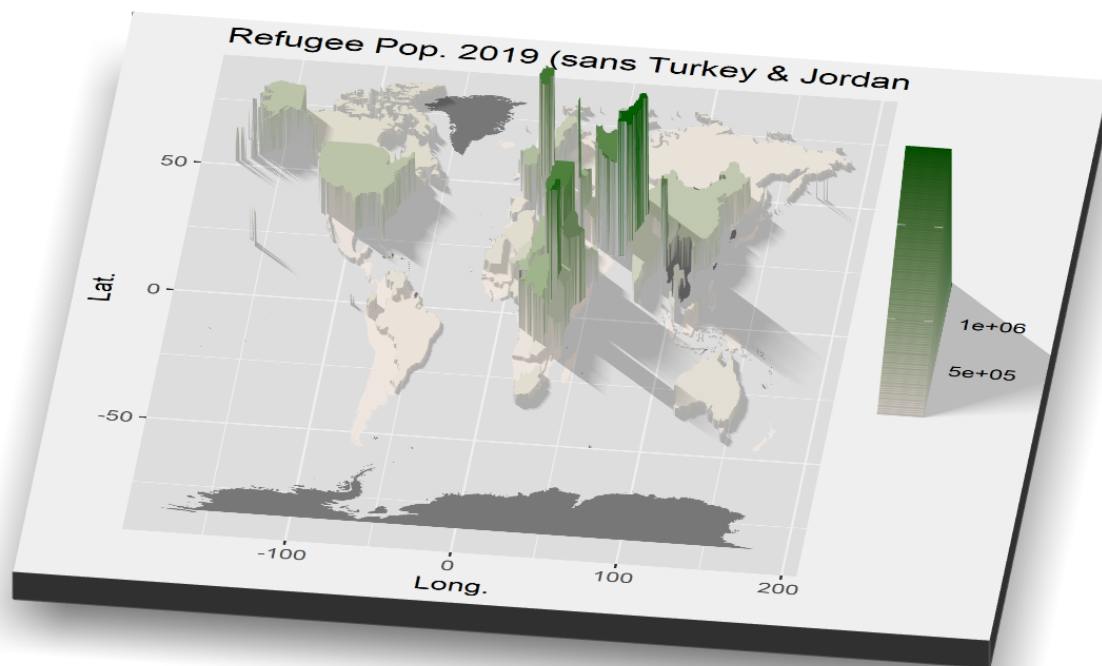


Figure 2: "Refugee Population by Country 2019 (sans Turkey & Jordan)"

8. Climate Mortality 2030 by Country 3D Model

```
# plotting climate mortality in 2030 by country
climmort_gg <- ggplot(Refclim_mapdf, aes(long, lat, group = group, fill = Climate_mortality)) +

# allowing flexible shape to match country borders
  geom_polygon() +

# coloring countries on a chosen gradient for value range
  scale_fill_gradient(low = "seashell", high = "darkred") +

# labeling title and x & y axes
  labs(title = "Climate Mortality 2030") +
  theme() +
  xlab("Long.") +
  ylab("Lat.") +

# removing legend title and setting background type
  theme(legend.title = element_blank(),
        panel.background = element_rect())

# rendering ggplot as 3D in rayshader, setting parameters and scale
plot_gg(climmort_gg, width = 5, height = 5, raytrace = FALSE, preview = TRUE) +
  plot_gg(climmort_gg, width = 5, height = 5, multicore = TRUE, scale = 250,

# establishing angle of snapshot perspective
    zoom = 0.5, theta = 7, phi = 35, windowsize = c(1200, 800)) +

# allowing process launched from R to read input files before execution resumes
  Sys.sleep(0.2) +

# constructing snapshot of final 3D model
  render_snapshot(clear = TRUE)
```

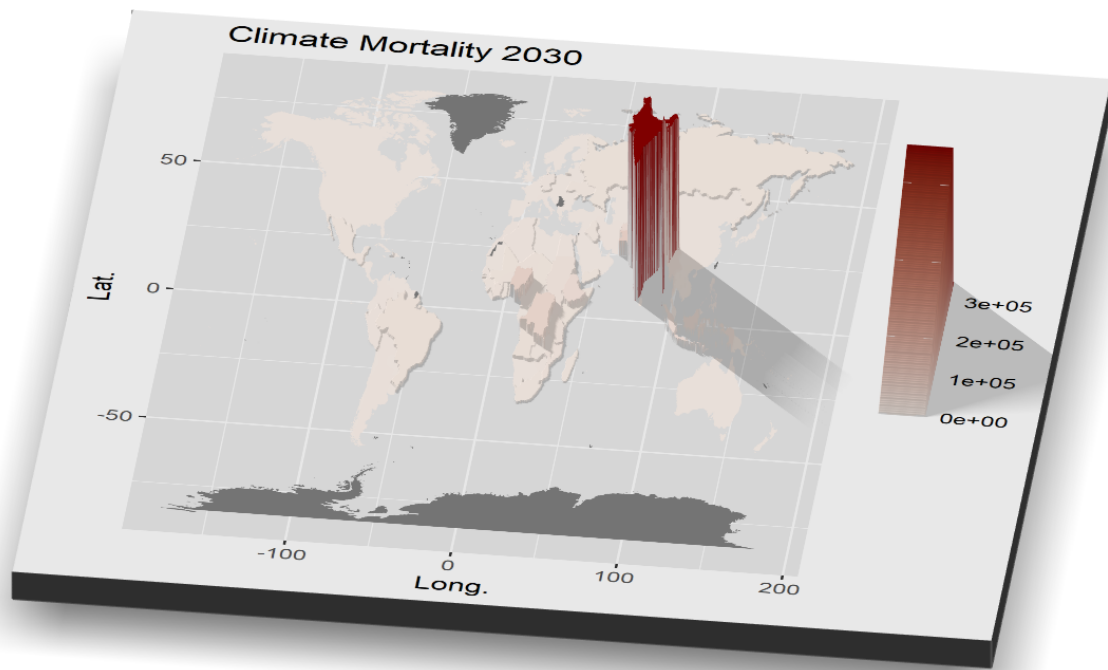


Figure 3: "Climate Mortality 2030 by Country"

9. Climate Mortality 2030 by Country 3D Model (sans India)

```
# filtering out India to calibrate for its disproportionate mortality
climmort_sans <- Refclim_mapdf %>%
  filter(region != "India")

# plotting climate mortality in 2030 by country, excluding India
climmort_sansgg <- ggplot(climmort_sans, aes(long, lat, group = group, fill = Climate_mortality)) +

# allowing flexible plot shape to match country borders
  geom_polygon() +

# coloring countries on a chosen gradient for value range
  scale_fill_gradient(low = "seashell", high = "darkred") +

# labeling title and x & y axes
  labs(title = "Climate Mortality 2030 (sans India)") +
  xlab("Long.") +
  ylab("Lat.") +

# removing legend title and setting background type
  theme(legend.title = element_blank(),
        panel.background = element_rect())

# rendering ggplot as 3D in rayshader, setting parameters and scale
plot_gg(climmort_sansgg, width = 5, height = 5, raytrace = FALSE, preview = TRUE) +
  plot_gg(climmort_sansgg, width = 5, height = 5, multicore = TRUE, scale = 250,

# establishing angle of snapshot perspective
  zoom = 0.5, theta = 7, phi = 35, windowsize = c(1200, 800)) +

# allowing process launched from R to read input files before execution resumes
  Sys.sleep(0.2) +

# constructing snapshot of final 3D model
  render_snapshot(clear = TRUE)
```

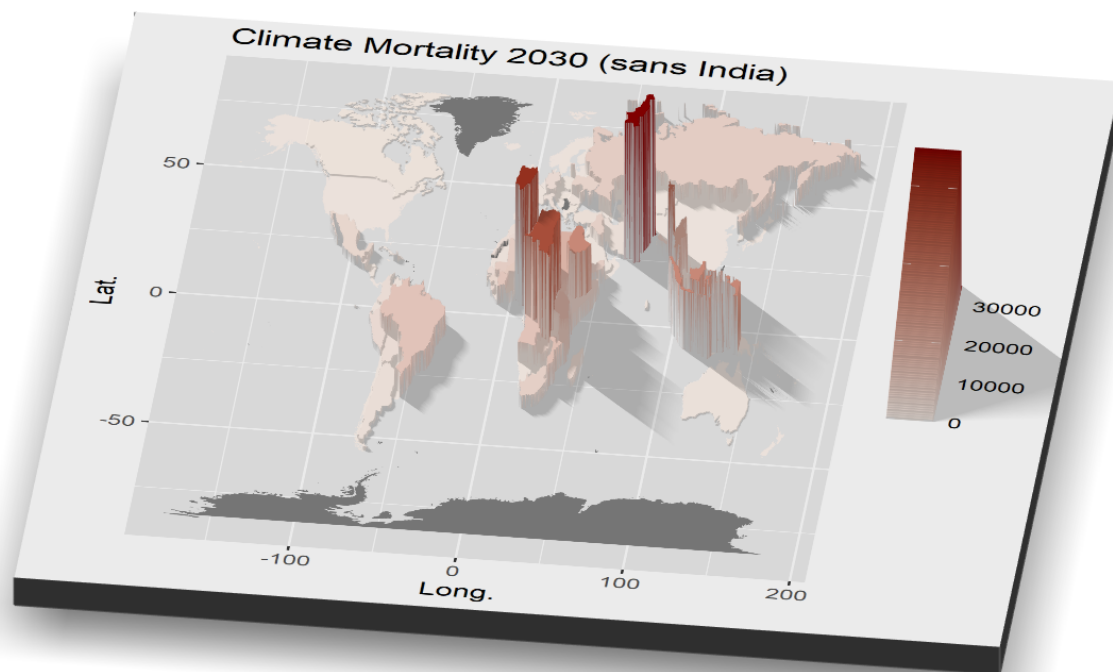


Figure 4: "Climate Mortality 2030 by Country (sans India)"