Assignment 2: Projections and Distortions

Julia Meinhardt

9/19/2020

```
library(ggplot2)
library(sf)
library(rnaturalearth)
library(tidyverse)
library(ggspatial)
library(cartogram)
library(ggthemes)
library(rgeos)
library(dplyr)
library(rnaturalearthhires)
library(proj4)
library(readxl)
theme_set(theme_bw())
```

Process and Thinking

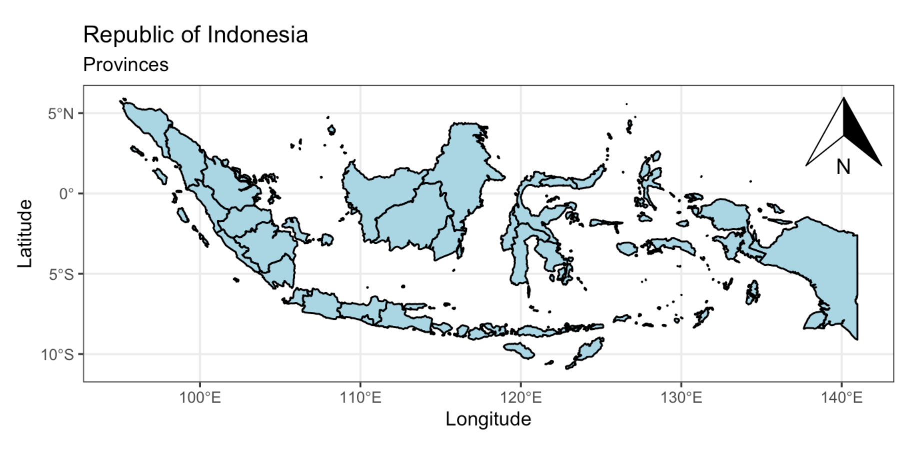
I'm using Carole's uploaded tutorial and Mel Moreno and Mathieu Basille's article "Drawing Beautiful Maps Programmatically with R, sf and ggplot2 — Part 1: Basics" as my fundamental guides to create these projections. I also referred to Taelor's tutorial for ideas. I'll be creating a map of Indonesia and exploring different projections and spatial data visualizations.

I chose Indonesia because my grandparents immigrated from Java and Sumatra, and I was interested in learning more about the country through mapping and data exploration. I chose this before realizing that we were supposed to pick one of six areas provided in the syllabus so that the area as large enough to show unique projections I was having trouble with this, but after speaking with my peers on Monday and hearing from Gianina and Summer during our breakout, I think I managed to find projections that are sufficiently distinct.

Indonesia by Province

```
indonesia_provinces <- ne_states(country = "Indonesia", returnclass = "sf")

ggplot(indonesia_provinces) +
  annotation_north_arrow(location = "tr", which_north = "true") +
  geom_sf(color = "black", fill = "lightblue") +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Republic of Indonesia", subtitle = ("Provinces"))
```



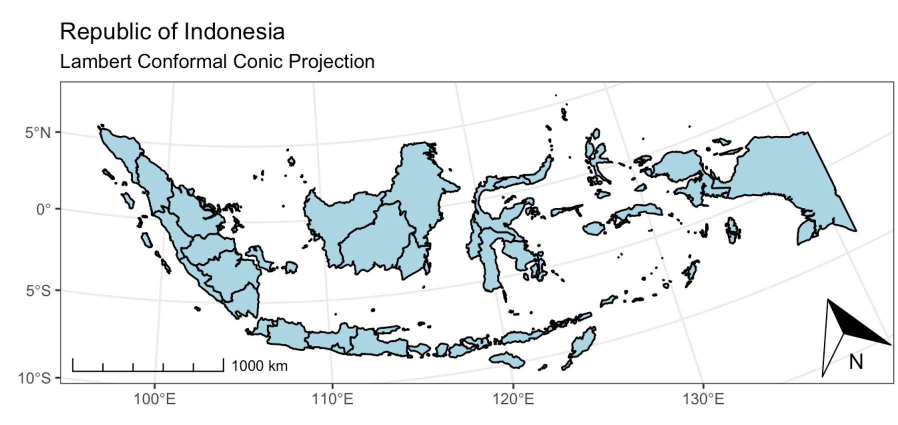
Loading Projections

I loaded four projections next: three for visualizing distinct projections and the fourth (albers) for my map that I'm mapping data on. These are all from spatialreference.org.

```
lambert <- "+proj=lcc +lat_1=30 +lat_2=62 +lat_0=0 +lon_0=105 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +n
o_defs "
equidistant <- "+proj=eqdc +lat_0=0 +lon_0=0 +lat_1=7 +lat_2=-32 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m
+no_defs "
epsg2110 <- "+proj=tmerc +lat_0=-39.51222222222222 +lon_0=175.64 +k=1 +x_0=400000 +y_0=800000 +ellps=GRS80 +towgs
84=0,0,0,0,0,0,0 +units=m +no_defs "
albers <- "+proj=aea +lat_1=7 +lat_2=-32 +lat_0=-15 +lon_0=125 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +
no_defs "
```

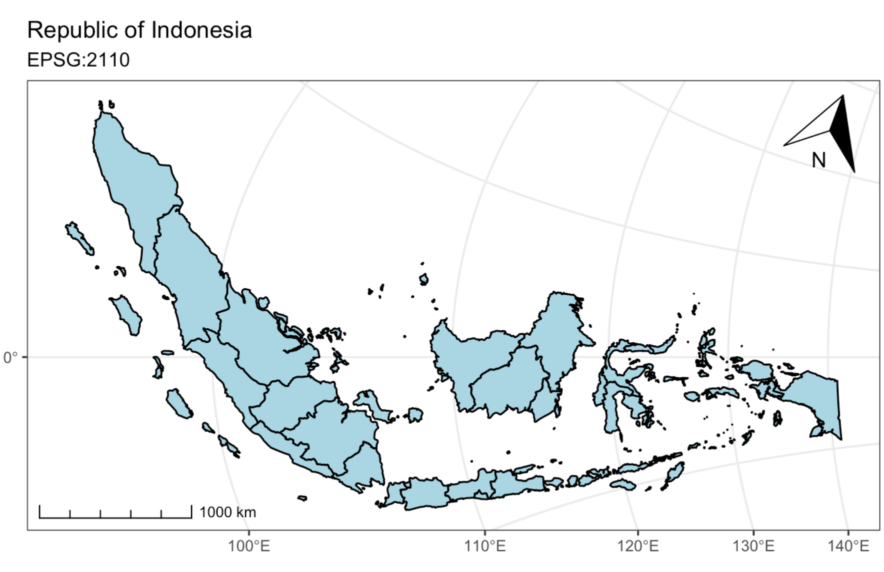
Lambert Conformal Conic Projection ERSI:102012

```
ggplot(indonesia_provinces) +
    geom_sf(color = "black", fill = "lightblue") +
    annotation_scale(style = "ticks") +
    annotation_north_arrow(location = "br", which_north = "true") +
    coord_sf(crs = lambert) +
    ggtitle("Republic of Indonesia", subtitle = ("Lambert Conformal Conic Projection"))
```



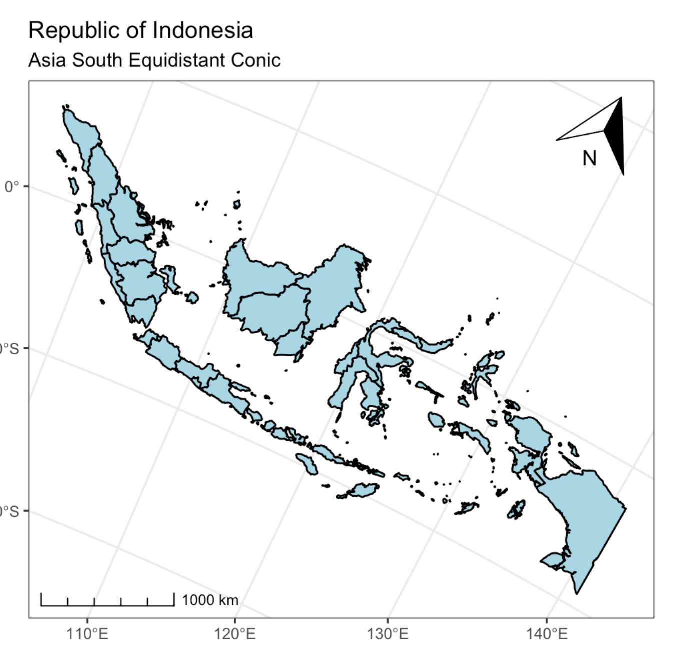
EPSG:2110

```
ggplot(indonesia_provinces) +
    geom_sf(color = "black", fill = "lightblue") +
    annotation_scale(style = "ticks") +
    annotation_north_arrow(location = "tr", which_north = "true") +
    coord_sf(crs = epsg2110 ) +
    ggtitle("Republic of Indonesia", subtitle = ("EPSG:2110"))
```



Asia South Equidistant Conic: ERSI:102029

```
ggplot(indonesia_provinces) +
    geom_sf(color = "black", fill = "lightblue") +
    annotation_scale(style = "ticks") +
    annotation_north_arrow(location = "tr", which_north = "true") +
    coord_sf(crs = equidistant) +
    ggtitle("Republic of Indonesia", subtitle = ("Asia South Equidistant Conic"))
```



Loading, Transforming, and Visualizing Data

I found a dataset on COVID-19 cases (confirmed, recovered, and deaths) by province for Indonesia that was updated in September 2020 (source: https://data.humdata.org/dataset/indonesia-covid-19-cases-recoveries-and-deaths-per-province)

After I visualized the data through the following maps and chart, I became curious about its accuracy. It makes sense that cases would be concentrated in Java, which is a heavily populated island, but I would expect to see more cases overall since Indonesia is the world's fourth most populated country. I did some research and found that, as in many places throughout the world, there are discrepancies in COVID-19 testing, counting, and data management. Thus, I'm including disclaimer that visualizations below do not represent rigorous data.

That said, I think this was a very effective lesson for me in the power and utility of data visualization and mapping! My visualizations made me ask questions and forced me to do further research to better understand my area of interest.

```
covid_indonesia <- read_csv("casesbyprovince.csv")
```

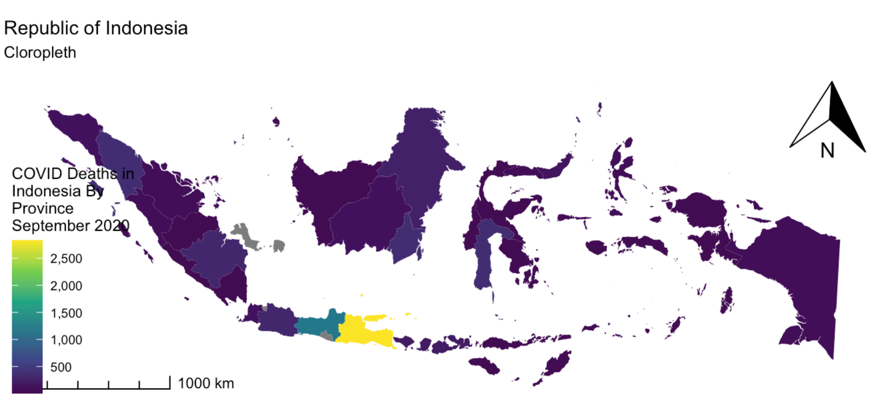
```
indonesia_transformed <- indonesia_provinces %>%
  st_transform(crs = albers)
```

```
indo_map_data <-indonesia_transformed %>%
  left_join(covid_indonesia, by = c("name" = "Province_name")) %>%
  select(Death_cases, Recovered_cases, Confirmed_cases, name)
```

Cloropleth

This following map shows the number of COVID deaths in Indonesia as of September 2020. Deaths in this dataset are concentrated in East Java (yellow), which is showing many more deaths than surrounding islands, or even other provinces on the same island. It makes sense for deaths to be concentrated on Java, which is the most populated island in Indonesia. What isn't clear is why Central Java and Western Java (both of which are on the same island and nearly double the population of East Java) are showing a much lower death count . Again, I'd imagine this has everything to do with gaps in the data.

```
ggplot(indo_map_data, aes(fill = Death_cases)) +
  geom_sf(color = NA) +
  scale_fill_viridis_c(
    name = "COVID Deaths in\nIndonesia By\nProvince\nSeptember 2020",
    breaks = seq(0, 3000, by = 500),
    labels = formatC(seq(0, 3000, by = 500),
                     big.mark = ",", format = "f", digits = 0)) +
    ggtitle("Republic of Indonesia", subtitle = ("Cloropleth")) +
    annotation_scale(style = "ticks") +
    annotation_north_arrow(location = "tr", which_north = "true") +
  theme_map() +
  theme(legend.background = element_blank())
```



Cartogram

The continuous cartogram below uses number of COVID deaths to determine both the size and color of provinces.

The reasons I didn't use population in this graphic as Carole did in the tutorial are twofold. First, I personally have some trouble interpreting these maps which use both size and color attached to different variables. Maybe my difficulty with reading these maps can applies more broadly however, as our assigned readings revealed that hue and saturation are the hardest visual cues for people to interpret correctly, and shape, area, and volume are not far behind. What might help me in this context is to see side-by-side cartograms with one using deaths to distort size and the other using population for comparison. Which leads me to the second reason I didn't use population which is that I had trouble finding data on population by province that would join with my other dataset reasonably easily! Province names vary by datset, with some using English names for provinces, others using Indonesian names, and some with variability within languages themselves.

```
covid_cartogram_cont <- indo_map_data %>%
  cartogram_cont("Death_cases")
```

```
## Warning in cartogram_cont.sf(., "Death_cases"): NA not allowed in weight vector.
## Features will be removed from Shape.
```

```
ggplot(covid_cartogram_cont, aes(fill = Death_cases)) +
  geom_sf(color = NA) +
  ggtitle("Republic of Indonesia", subtitle = ("Cartogram")) +
  scale_fill_viridis_c(
    name = "Covid Deaths in \nIndonesia By \nProvince",
    breaks = seq(0, 3000, by = 500),
    labels = formatC(seq(0, 3000, by = 500),
                     big.mark = ",", format = "f", digits = 0)) +
  theme_map() +
  theme(legend.background = element_blank())
```

Proportional Symbol Map

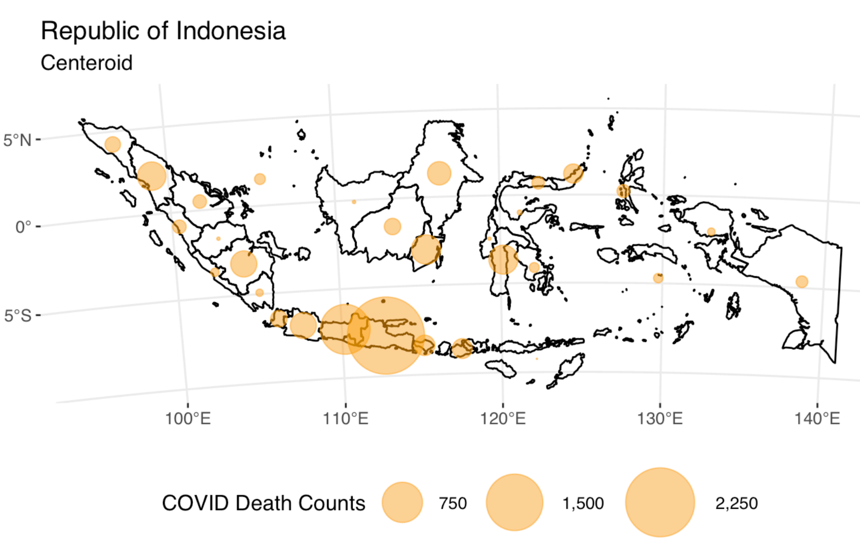
I find proportional symbol maps much more readable than both clorpleths and cartograms in this context. The many islands and small geographic areas can make color harder to distinguish and pull the eye almost exclusively to Java, making the other provinces seem irrelevant. Of course, that might be one's intent for some reason, in which case the above maps might be a good choice. But for a more straightforward visualization of deaths by province, I find this type of map much more effective.

```
indo_centeroids <- indo_map_data %>%
  st_centroid()
```

```
## Warning in st_centroid.sf(.): st_centroid assumes attributes are constant over
## geometries of x
```

```
ggplot(indonesia_transformed) +
  geom_sf(fill = NA, color = "black") +
  geom_sf(data = indo_centeroids,
          aes(size = Death_cases),
          alpha = 0.5, color = "orange") +
  ggtitle("Republic of Indonesia", subtitle = ("Centeroid")) +
  scale_size_continuous(name = "COVID Death Counts",
    breaks = seq(0, 3000, by = 750),
    labels = formatC(seq(0, 3000, by = 750),
                     big.mark = ",", format = "f", digits = 0),
    range = c(0, 20)) +
  theme_hc()
```

```
## Warning: Removed 3 rows containing missing values (geom_sf).
```



Bar Chart

```
ggplot(indo_map_data,
       aes(x = reorder (name, -Death_cases),
           y = Death_cases)) +
  geom_bar(stat = "identity", color = "black", fill = "orange")+
  labs(title = 'Indonesia COVID Deaths by Province',
       subtitle = 'As of September 2020')+
  theme_solarized_2()+
  coord_flip()
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```