

# Data Visualization with R

## Workshop Day 2



# Tuberculosis incidence

The TB data is from the WHO.

```
## # A tibble: 40,800 x 5
##   country     year age_group sex  count
##   <chr>       <dbl> <fct>    <chr> <dbl>
## 1 Afghanistan 1997 15-24     m      10
## 2 Afghanistan 1997 25-34     m      6
## 3 Afghanistan 1997 35-44     m      3
## 4 Afghanistan 1997 45-54     m      5
## 5 Afghanistan 1997 55-64     m      2
## 6 Afghanistan 1997 65-      m      0
## 7 Afghanistan 1997 15-24     f      38
## 8 Afghanistan 1997 25-34     f      36
## 9 Afghanistan 1997 35-44     f      14
## 10 Afghanistan 1997 45-54    f      8
## # ... with 40,790 more rows
```

# What is a choropleth map?

## Why use a choropleth map?



# How do we get a map?

A polygon map of the world can be extracted from the maps package.

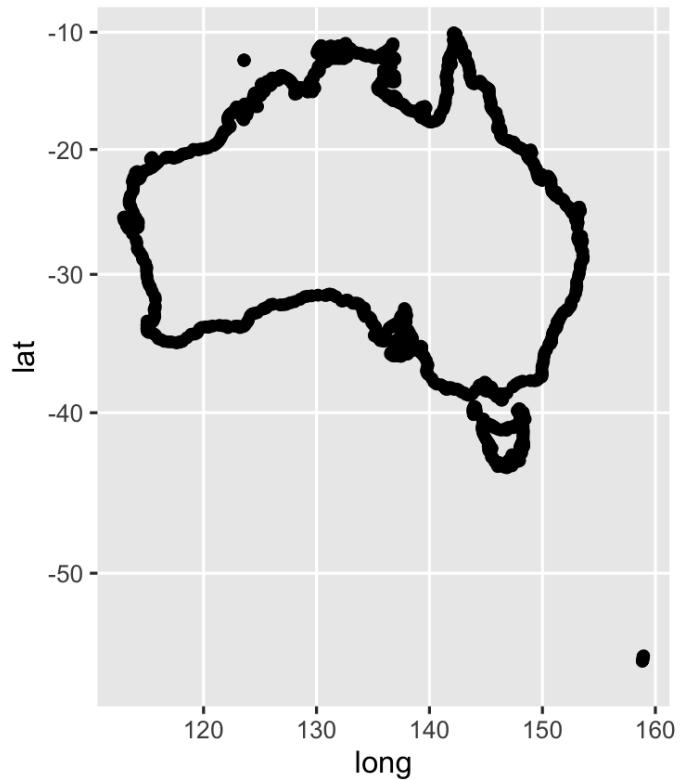
```
world_map <- map_data("world")
world_map %>%
  filter(region %in% c("Australia", "New Zealand")) %>%
  DT::datatable(width=1150, height=100)
```

	long	lat	group	order	region	subregion
1	123.594528198242	-12.4256830215454	133	7115	Australia	Ashmore and Cartier Islands
2	123.595207214355	-12.4359369277954	133	7116	Australia	Ashmore and Cartier Islands
3	123.573150634766	-12.4341802597046	133	7117	Australia	Ashmore and Cartier Islands
4	123.572463989258	-12.4239253997803	133	7118	Australia	Ashmore and Cartier Islands
5	123.594528198242	-12.4256830215454	133	7119	Australia	Ashmore and Cartier Islands
6	158.878799438477	-54.7097625732422	139	7267	Australia	Macquarie Island
7	158.84521484375	-54.7492179870605	139	7268	Australia	Macquarie Island

# Maps are basically groups of connected dots

These are the points, defining the country boundary for Australia

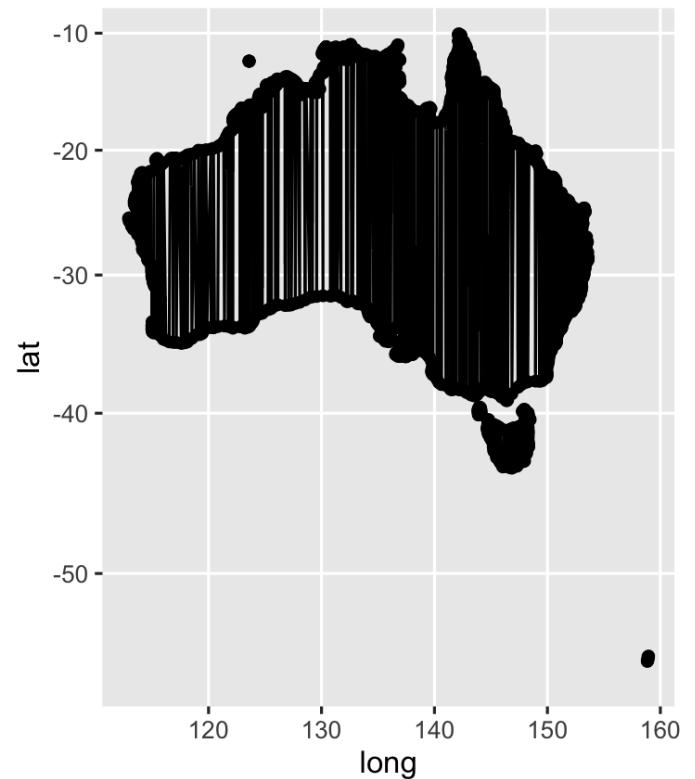
```
oz <- world_map %>%
  filter(region == "Australia")
ggplot(oz, aes(x = long, y = lat)) +
  geom_point() +
  coord_map()
```



## Connect the dots

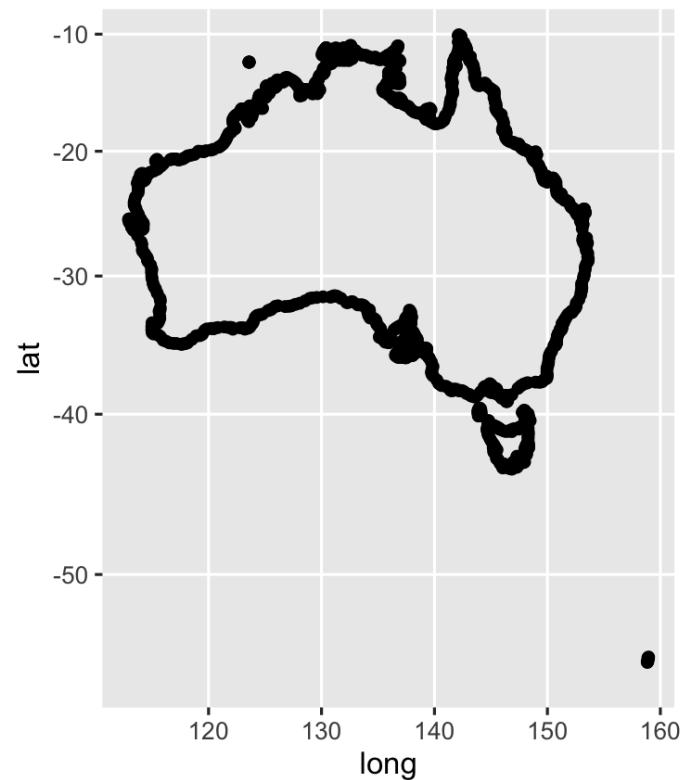
```
ggplot(oz, aes(x = long, y = lat,  
               group = group)) +  
  geom_point() +  
  geom_line() +  
  coord_map()
```

What happened?



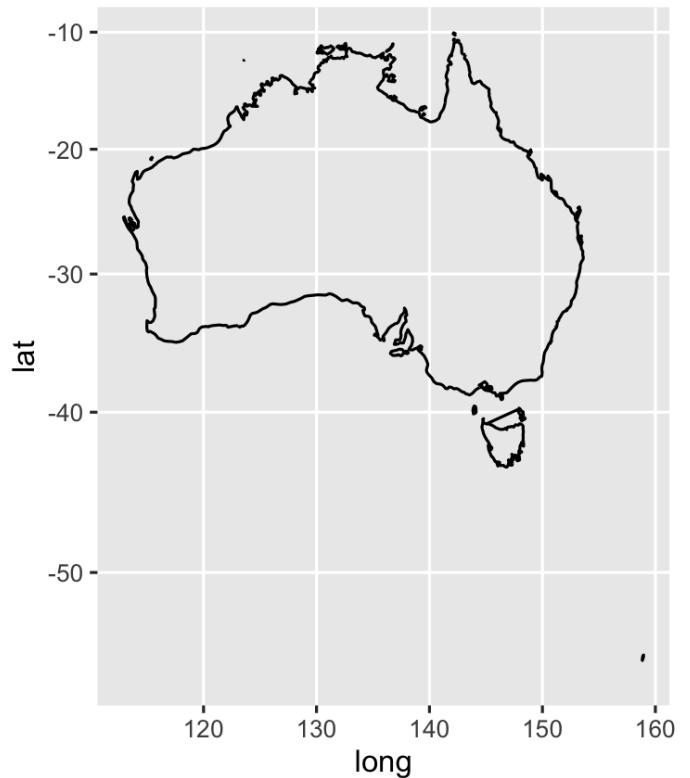
## Connect the dots

```
ggplot(oz, aes(x = long, y = lat,  
               group = group)) +  
  geom_point() +  
  geom_path() +  
  coord_map()
```



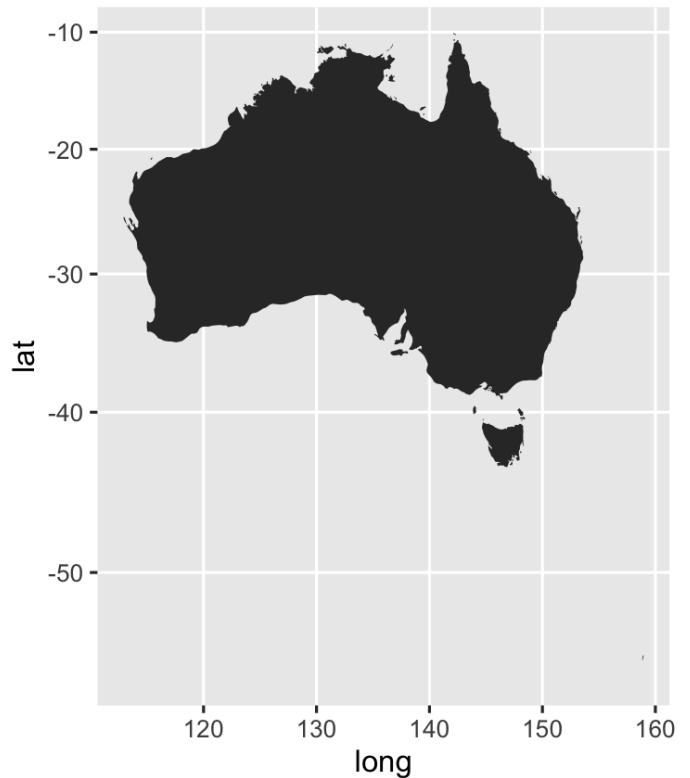
This map doesn't have states and territory connections.

```
ggplot(oz, aes(x = long, y = lat,  
               group = subregion)) +  
  geom_path() +  
  coord_map()
```



We can also plot the map using  
geom\_polygon, and fill with colour.

```
ggplot(oz, aes(x = long, y = lat,  
               group = group)) +  
  geom_polygon() +  
  coord_map()
```



Using a map theme makes the result look more map like

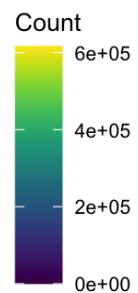
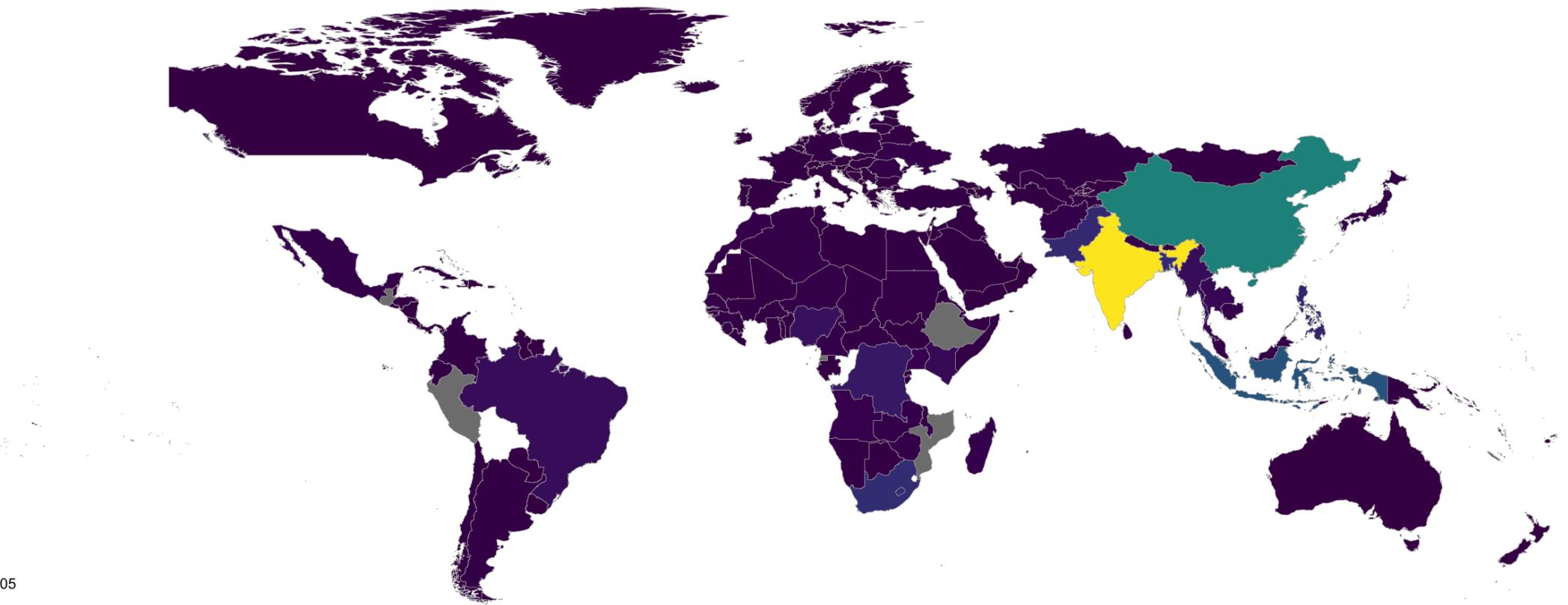
```
ggplot(oz, aes(x = long, y = lat,  
               group = group)) +  
  geom_polygon() +  
  coord_map() +  
  theme_map()
```





## Aggregate counts across sex and age group for 2012

```
tb_2012 <- tb %>%
  filter(year == 2012) %>%
  rename(region = country) %>%
  group_by(region) %>%
  summarise(count = sum(count))
ggplot(tb_2012, aes(map_id = region)) +
  geom_map(aes(fill = count), map = world_map,
           color="grey70", size = 0.1, na.rm = TRUE) +
  expand_limits(x = world_map$long, y = world_map$lat) +
  scale_fill_viridis("Count") +
  theme_map()
```



# What happened to the USA? UK?



# Check the name matching

```
wm_names <- world_map %>%  
  select(region) %>%  
  distinct()  
  
tb_names <- tb %>%  
  filter(year == 2012) %>%  
  select(country) %>%  
  distinct()  
  
tb_miss_from_wm <- anti_join(tb_names, wm_names,  
                                by=c("country" = "region"))  
  
DT::datatable(tb_miss_from_wm, width = 1150, height = 100)
```

Show 10 entries

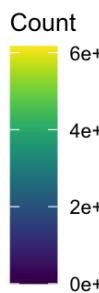
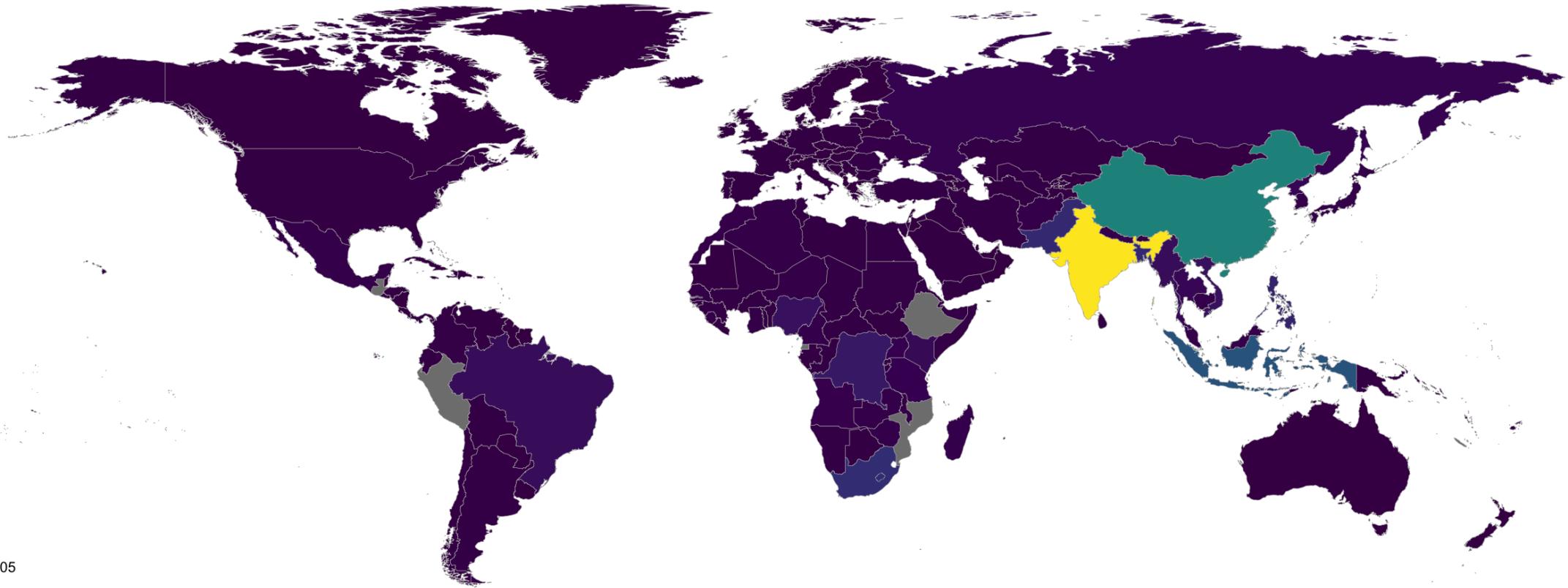
Search:

	country
1	Antigua and Barbuda
2	Bolivia (Plurinational State of)
3	British Virgin Islands
4	Brunei Darussalam

```
tb_fixed <- tb %>%
  mutate(region=recode(country,
    "United States of America" = "USA",
    "United Kingdom of Great Britain and Northern Ireland" = "UK",
    "Russian Federation" = "Russia",
    "Viet Nam" = "Vietnam",
    "Venezuela (Bolivarian Republic of)" = "Venezuela",
    "Bolivia (Plurinational State of)" = "Bolivia",
    "Czechia" = "Czech Republic",
    "Iran (Islamic Republic of)" = "Iran",
    "Iran (Islamic Republic of)" = "Laos",
    "Democratic People's Republic of Korea" = "North Korea",
    "Republic of Korea" = "South Korea",
    "United Republic of Tanzania" = "Tanzania",
    "Congo" = "Republic of Congo"))
```

😊 Try again!

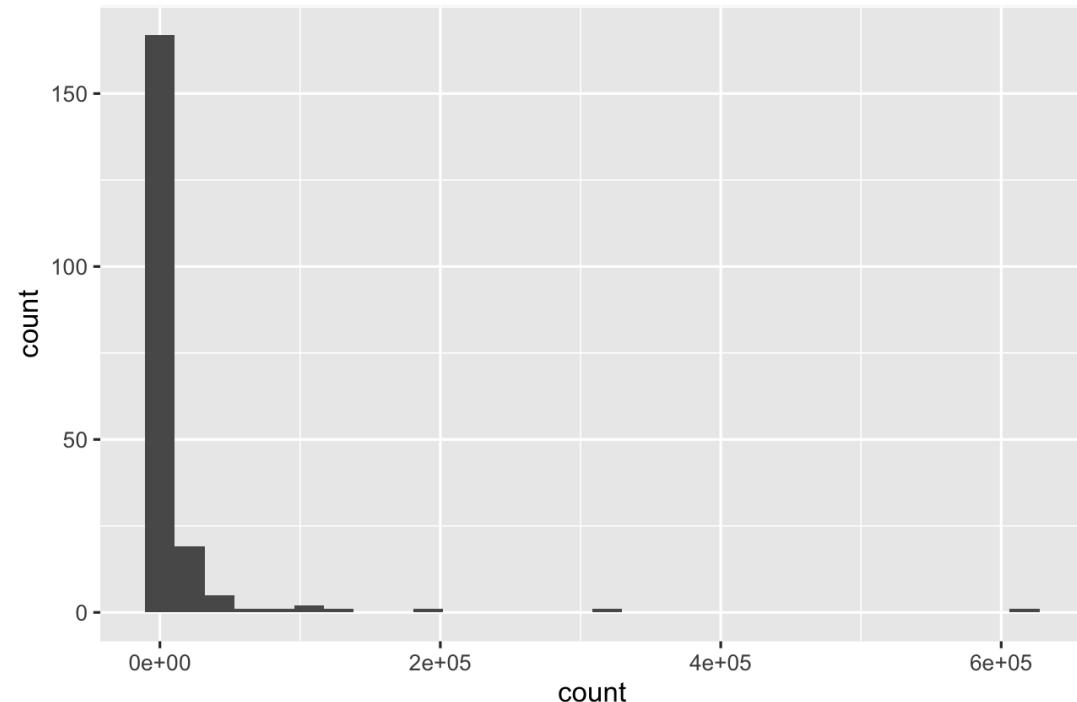
```
tb_2012 <- tb_fixed %>%  
  filter(year == 2012) %>%  
  group_by(region) %>%  
  summarise(count = sum(count))  
ggplot(tb_2012, aes(map_id = region)) +  
  geom_map(aes(fill = count), map = world_map,  
           color = "grey70", size = 0.1, na.rm = TRUE) +  
  expand_limits(x = world_map$long, y = world_map$lat) +  
  scale_fill_viridis("Count") +  
  theme_map()
```



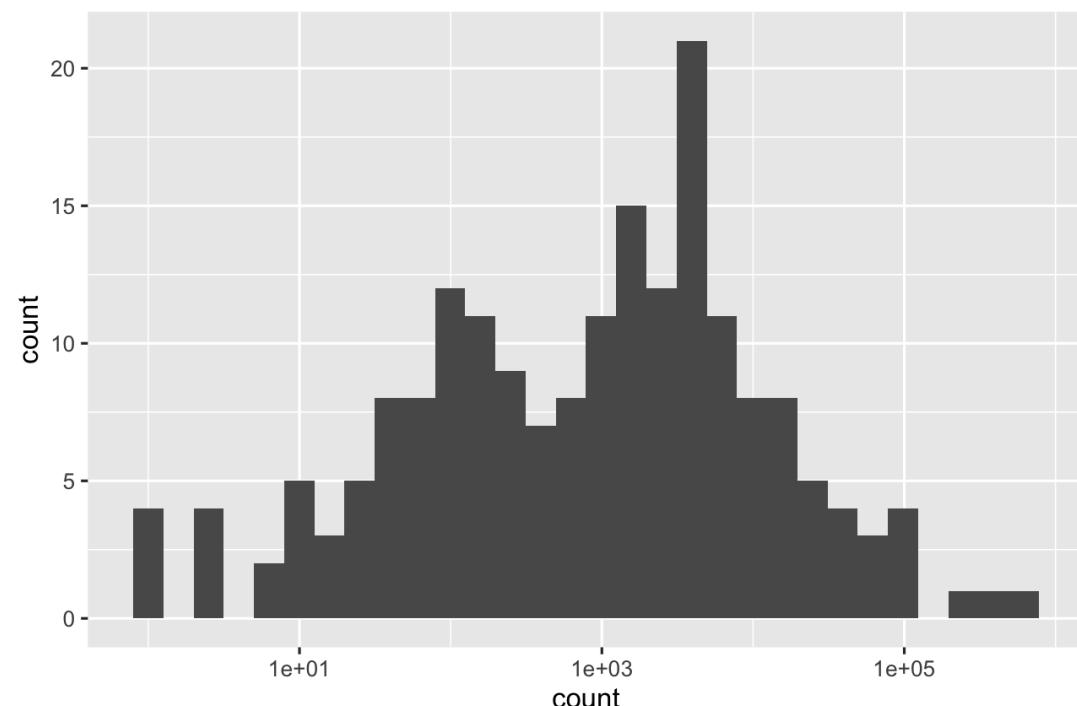
# Counts are typically skewed

may be best to symmetrise

```
ggplot(tb_2012, aes(x = count)) +  
  geom_histogram()
```

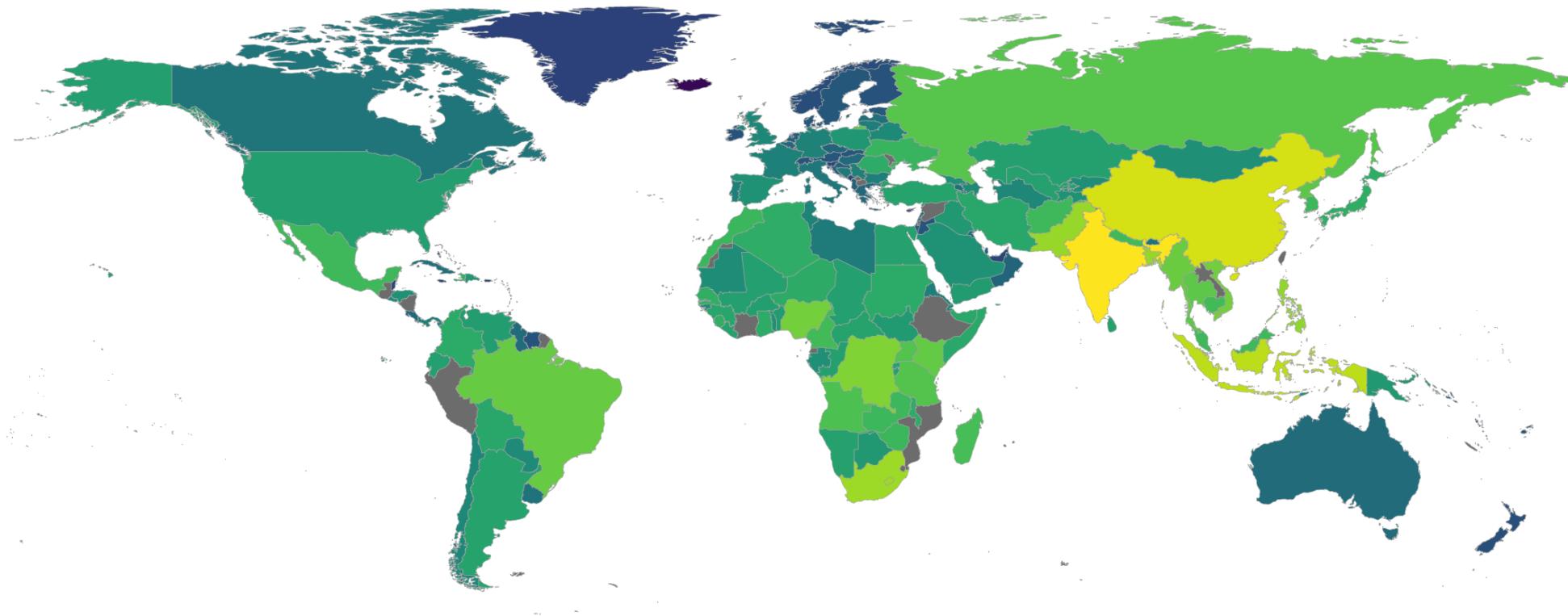


```
ggplot(tb_2012, aes(x = count)) +  
  geom_histogram() +  
  scale_x_log10()
```



# geom\_polygon can also be used

```
tb_2012_map <- world_map %>% left_join(tb_2012)
ggplot(tb_2012_map, aes(x = long, y = lat, group=group)) +
  geom_polygon(aes(fill = count),
               color="grey70", size = 0.1, na.rm = TRUE) +
  expand_limits(x = world_map$long*1.1, y = world_map$lat*1.1) +
  scale_fill_viridis("Count", trans = "log10") +
  theme_map()
```



Count

Color	Count Range
Dark Purple	1
Dark Teal	~100
Medium Teal	~1000
Light Green	~10,000



# Get the data

This is extracted from <https://www.covid19data.com.au/victoria>

data R

```
# Read the data
# Replace null with 0, for three LGAs
# Convert to long form to join with polygons
# Make the date variables a proper date
# Set NAs to 0, this is a reasonable assumption
covid <- read_csv(here::here("data/melb_lga_covid.csv")) %>%
  mutate(Buloke = as.numeric(ifelse(Buloke == "null", "0", Buloke))) %>%
  mutate(Hindmarsh = as.numeric(ifelse(Hindmarsh == "null", "0", Hindmarsh))) %>%
  mutate(Towong = as.numeric(ifelse(Towong == "null", "0", Towong))) %>%
  pivot_longer(cols = Alpine:Yarriambiack, names_to="NAME", values_to="cases") %>%
  mutate(Date = ydm(paste0("2020/",Date))) %>%
  mutate(cases=replace_na(cases, 0))
DT::datatable(covid, width=1150, height=100)
```

# Get a map from ozmaps

data R

```
data("abs_lga")
vic_lga <- abs_lga %>%
  mutate(NAME = ifelse(NAME == "Latrobe (M) (Tas.)", "LatrobeM", NAME)) %>%
  mutate(NAME = ifelse(NAME == "Kingston (DC) (SA)", "KingstonSA", NAME)) %>%
  mutate(NAME = ifelse(NAME == "Bayside (A)", "BaysideA", NAME)) %>%
  mutate(NAME = str_replace(NAME, "\\\(.+\\)", "")) %>%
  mutate(NAME = ifelse(NAME == "Colac-Otway", "Colac Otway", NAME))
vic_lga <- st_transform(vic_lga, 3395)
# 3395 is EPSG CRS, equiv to WGS84 mercator,
# see https://spatialreference.org/ref/epsg/?page=28
# cartogram() needs this to be set
```

# Join and colour the map

plot R

```
# Filter to final day, which is cumulative count
covid_cum <- covid %>%
  filter(Date == max(Date))

# Join covid data to polygon data, remove LGAs with
# missing values which should leave just Vic LGAs
vic_lga_covid <- vic_lga %>%
  left_join(covid_cum, by = "NAME") %>%
  filter(!is.na(cases))

# Make choropleth map, with appropriate colour palette
p <- ggplot(vic_lga_covid) +
  geom_sf(aes(fill = cases, label=NAME), colour="white") +
  scale_fill_distiller("Cases", palette = "YlOrRd",
                      direction=1) +
  theme_map() +
```

# Get population data

data R

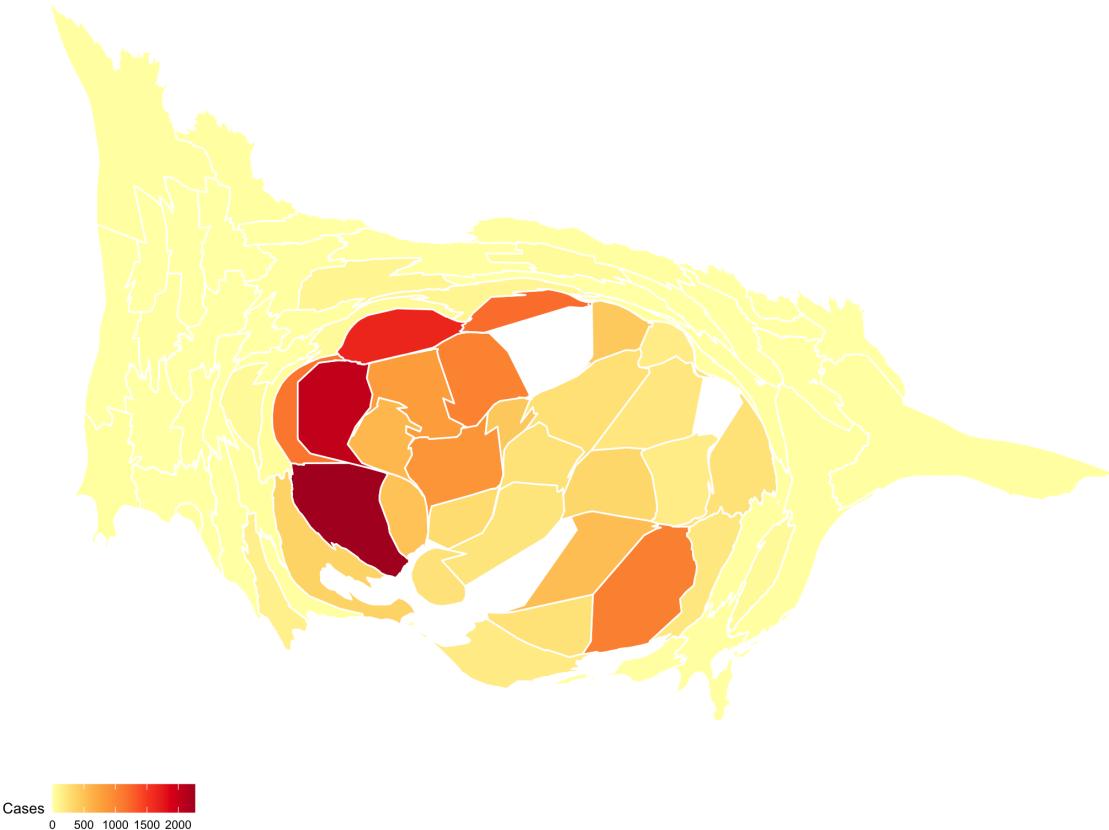
- Incorporate population data to make cartogram
- Population from <https://www.planning.vic.gov.au/land-use-and-population-research/victoria-in-future/tabc-pages/victoria-in-future-data-tables>
- Polygons are transformed so that area matches, as best possible, to the population

Show 10 entries

Search:

	NAME	pop
1	Alpine	12578
2	Ararat	11746
3	Ballarat	103500
4	Banyule	127447
5	Bass Coast	33465
6	Baw Baw	49296
7	Bayside	102912
8	Benalla	13981

# Make a cartogram



# Resources

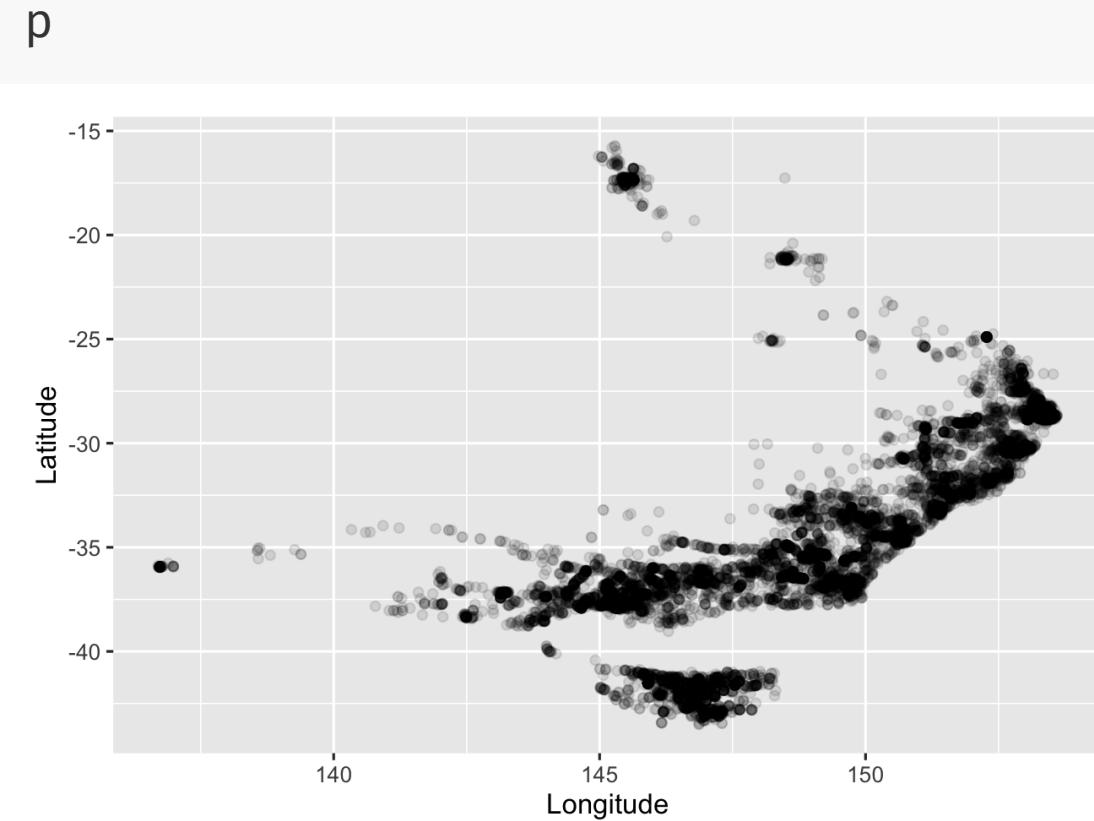
Where to go to get more help on maps

- ozmaps package: <https://github.com/mdsumner/ozmaps>,  
<https://mdsumner.github.io/ozmaps/>
- eechidna package <https://docs.ropensci.org/eechidna/>
- <https://www.littlemissdata.com/blog/maps>
- <https://www.r-spatial.org/r/2018/10/25/ggplot2-sf.html>
- <https://www.paulamoraga.com/book-geospatial/sec-spatialdataandCRS.html>

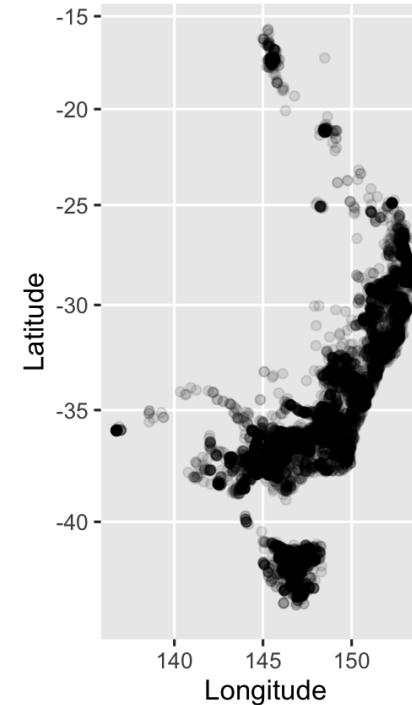
Note: It is important when converting spatial objects from a mapping software to a data analysis project is "thinning" the map to make it smaller and efficient to work with.



```
load(here:::here("data/platypus.rda"))
p <- ggplot(platypus) +
  geom_point(aes(x = Longitude,
                 y = Latitude),
              alpha = 0.1)
```



```
p + coord_map()
```



# Extract Open Street Map using ggmap

```
oz_bbox <- c(112.9, # min long  
           -45, # min lat  
           159, # max long  
           -10) # max lat  
  
oz_map <- get_map(location = oz_bbox, source = "osm")  
save(oz_map, file=here::here("data/oz_map.rda"))
```

```
load(here::here("data/oz_map.rda"))
ggmap(oz_map) +
  geom_point(data = platypus,
             aes(x = Longitude,
                  y = Latitude),
             alpha = 0.1,
             colour = "orange") +
  theme_map()
```





</> Open day2-exercise-02.Rmd

15:00

# Session Information

```
devtools::session_info()
```

```
## - Session info --
##   setting  value
##   version  R version 4.0.1 (2020-06-06)
##   os        macOS Catalina 10.15.7
##   system   x86_64, darwin17.0
##   ui        X11
##   language (EN)
##   collate  en_AU.UTF-8
##   ctype    en_AU.UTF-8
##   tz       Australia/Sydney
##   date     2020-11-08
##
## - Packages --
##   package      * version    date lib

```

These slides are licensed under

