```
library(readr)
library(lubridate)
```

```
##
## Attaching package: ’lubridate’

## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
library(janitor)
```

```
##
## Attaching package: ’janitor’

## The following objects are masked from ’package:stats’:
##
##     chisq.test, fisher.test
```

```
library(tidyr)
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(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v ggplot2 3.4.0     v stringr 1.5.0
## v tibble  3.1.8     v forcats 1.0.0
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : ’length(x) = 2 > 1’ in coercion to ’logical(1)’
```

```
##
## Attaching package: ’kableExtra’
##
## The following object is masked from ’package:dplyr’:
##
##     group_rows
```

```
library(tidyverse)
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(viridis)
```

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

```
library(ggplot2)
library(leaflet)
library(maps)
```

```
##
## Attaching package: ’maps’
##
## The following object is masked from ’package:viridis’:
##
##     unemp
##
## The following object is masked from ’package:purrr’:
##
##     map
```

```
gdp <- read_csv("gdp_1960_2020.csv")
```

```
## Rows: 10134 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country, state
## dbl (4): year, rank, gdp, gdp_percent
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
population <- read_csv("population_total_long.csv")
```

```
## Rows: 12595 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country Name
## dbl (2): Year, Count
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
names <- read_csv("noc_regions.csv")
```

```
## Rows: 230 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): NOC, region, notes
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
athletes <- read_csv("athlete_events.csv")
```

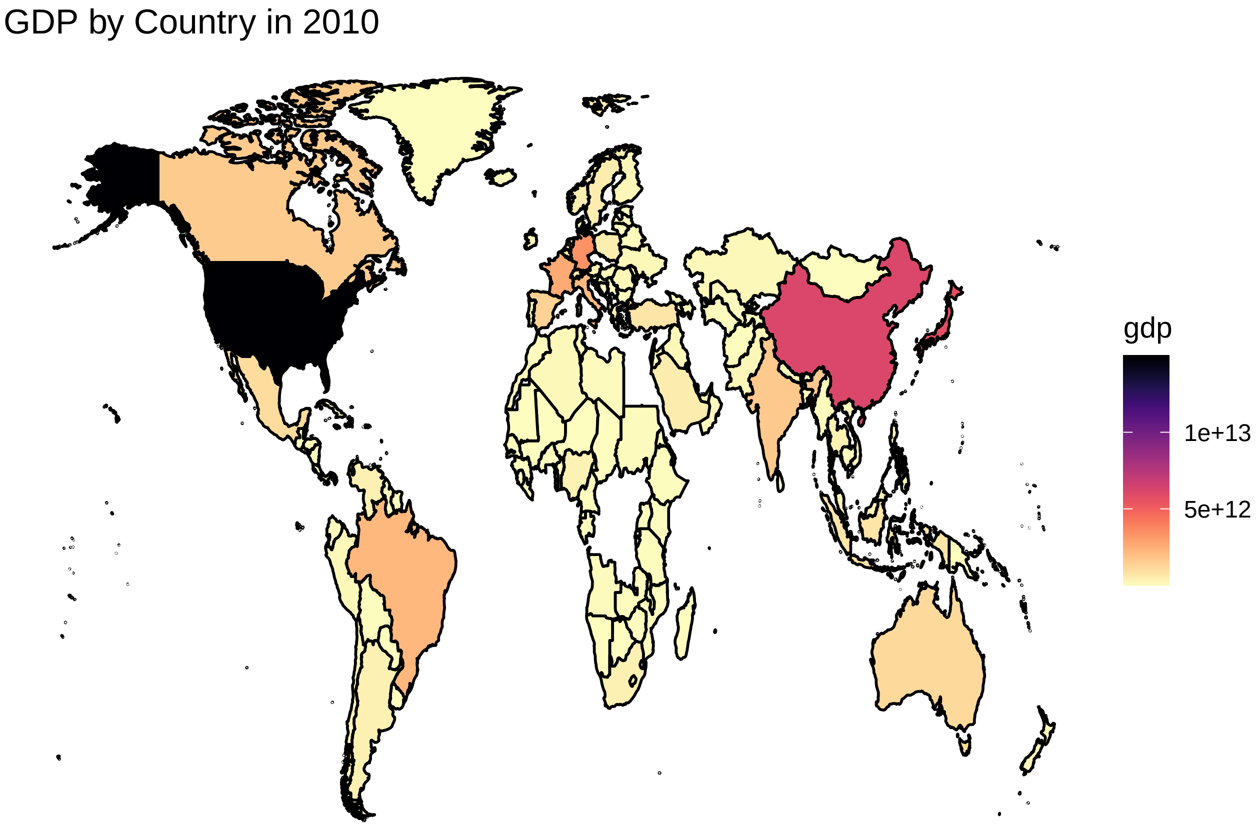
```
## Rows: 271116 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): Name, Sex, Team, NOC, Games, Season, City, Sport, Event, Medal
## dbl  (5): ID, Age, Height, Weight, Year
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
gdp <- rename(gdp, Country = country, Year = year)
population <- rename(population, Country = 'Country Name')

gdp <- gdp %>%
  mutate(Country = ifelse(Country == "the United States", "USA", Country)) #%>%
 # filter(Year >= 1960 & Year <= 2017)

population <- population %>%
  mutate(Country = ifelse(Country == "United States", "USA", Country)) %>%
  mutate(Country = ifelse(Country == "Venezuela, RB", "Venezuela", Country))

#unique(gdp$Country)
#unique(population$Country)

#show which countries are in gdp that are not in pop
gdp_not_in_pop <- anti_join(gdp, population, by = "Country")

#unique(gdp_not_in_pop$Country)
```

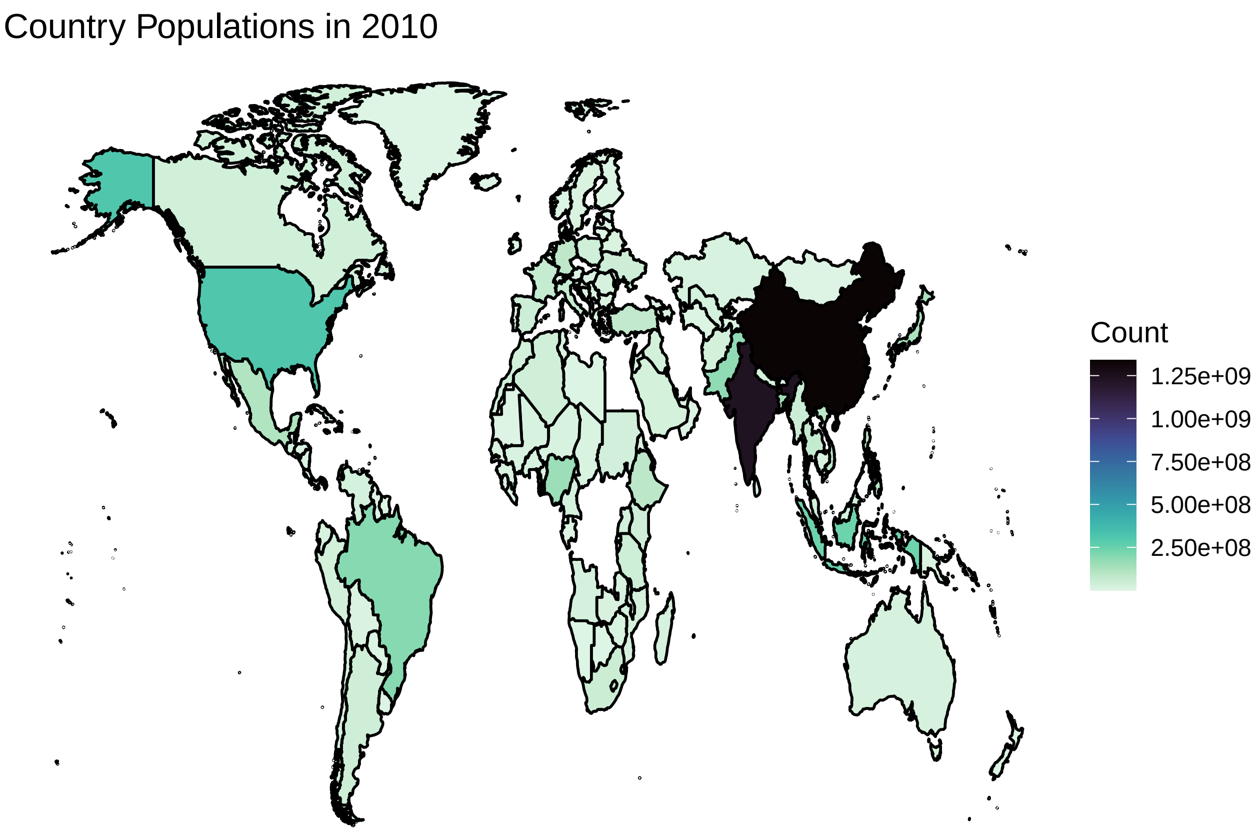
```r
#show which countries are in pop that are not in gdp
pop_not_in_gdp <- anti_join(population, gdp, by = "Country")

#unique(pop_not_in_gdp$Country)

#get rid of countries in the gdp dataset that are not in pop dataset
gdp <- gdp %>%
  filter(!(Country %in% gdp_not_in_pop$Country))


#get rid of countries in the pop dataset that are not in gdp dataset
population <- population %>%
  filter(!(Country %in% pop_not_in_gdp$Country))

#only look at 2010 data, the most recent year possible in the conjoined dataset
gdp_year <- gdp %>%
  filter(Year == 2010)

pop2010 <- population %>%
  filter(Year == 2010)

#join gdp and pop datasets by Country and Year
gdp_pop <- inner_join(gdp_year, population, by = c("Country", "Year"))

world_map <- map_data("world")

map_data_gdp <- inner_join(world_map, gdp_pop, by = c("region" = "Country"))
```

```
## Warning in inner_join(world_map, gdp_pop, by = c(region = "Country")): Each row in ‘x‘ is expected to
## i Row 34717 of ‘x‘ matches multiple rows.
## i If multiple matches are expected, set ‘multiple = "all"‘ to silence this
##   warning.
```

```r
ggplot(map_data_gdp, aes(x = long, y = lat, group = group, fill = gdp)) +
  geom_polygon(color = "black") +
  scale_fill_viridis(option = "magma", direction = -1) +
  ggtitle("GDP by Country in 2010") +
  theme_void()
```

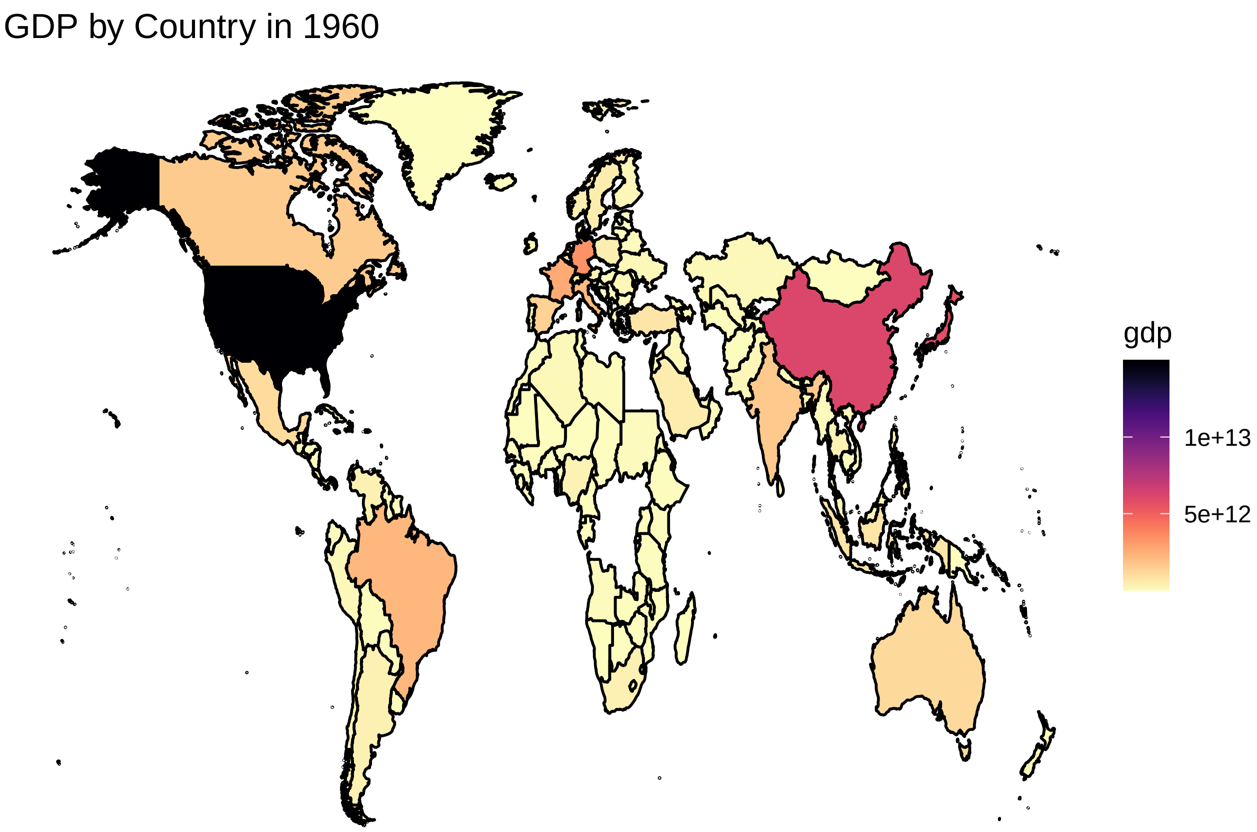


```
ggplot(map_data_gdp, aes(x = long, y = lat, group = group, fill = Count)) +
  geom_polygon(color = "black") +
  scale_fill_viridis(option = "mako", direction = -1) +
  ggtitle("Country Populations in 2010") +
  theme_void()
```



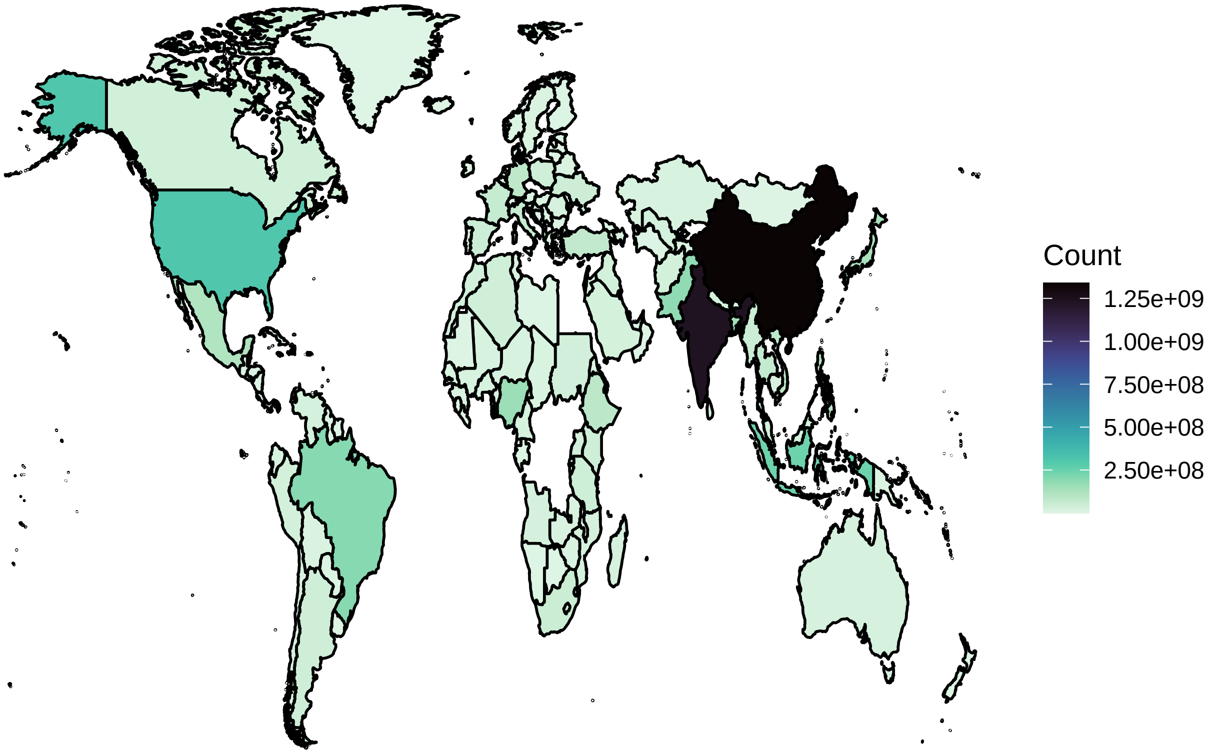
```
#need to work on showing GDP values for a given year and make it interactive? Or just show data for a g
# need to make one for population
```

```
gdp_1960 <- gdp %>%
  filter(Year == 1960)

pop1960 <- population %>%
  filter(Year == 1960)

#join gdp and pop datasets by Country and Year
gdp_pop2 <- inner_join(gdp_1960, pop1960, by = c("Country", "Year"))

map_data_gdp2 <- inner_join(world_map, gdp_pop2, by = c("region" = "Country"))

ggplot(map_data_gdp, aes(x = long, y = lat, group = group, fill = gdp)) +
  geom_polygon(color = "black") +
  scale_fill_viridis(option = "magma", direction = -1) +
  ggtitle("GDP by Country in 1960") +
  theme_void()
```



```
ggplot(map_data_gdp, aes(x = long, y = lat, group = group, fill = Count)) +
  geom_polygon(color = "black") +
  scale_fill_viridis(option = "mako", direction = -1) +
  ggtitle("Country Populations in 1960") +
  theme_void()
```

Country Populations in 1960


Count
1.25e+09
1.00e+09
7.50e+08
5.00e+08
2.50e+08