Covid-19 Project

Emily Agreda

2024-12-10

In this report looking at COVID19 Data provided by John's Hopkins University, we will explore various trends present in the data. The first part of this report was done as a follow along and class. First, we loaded all necessary packages. Next we started by loading in all of the necessary data from JHU.

```
##library in packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(lubridate)
library(dplyr)
library(lmerTest)
```

```
## Loading required package: lme4
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## Attaching package: 'lmerTest'
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
## The following object is masked from 'package:stats':
##
##     step
```

```
library(ggplot2)
```

```
#Get current Data in the four files
url_in <- "https://raw.githubusercontent.com/CSSEGISandDATA/COVID-19/master/csse_covid_19_data/csse_cov
file_names <-
  c("time_series_covid19_confirmed_global.csv",
    "time_series_covid19_deaths_global.csv",
    "time_series_covid19_confirmed_US.csv",
    "time_series_covid19_deaths_US.csv")
urls <- stringr::str_c(url_in, file_names)
```

```
global_cases <- read_csv(urls[1])
global_deaths <- read_csv(urls[2])
US_cases <- read_csv(urls[3])
US_deaths <- read_csv(urls[4])
```

After looking at global_cases and global_death, we will modify those data sets and put each variable (date, cases, deaths) in their own column and rename Region and State so they are easier to work with. We will also be getting rid of the Lat and Long columns as we don't plan to use them for this analysis.

```
#Transform global cases data
global_cases <- global_cases %>%
    pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
                 names_to = "date",
                 values_to = "cases") %>%
    select(-c(Lat,Long)) #remove lat and long columns

#Transform global deaths data
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long))

# Merge cases and deaths
global <- global_cases %>%
  full_join(global_deaths, by = c("Province/State", "Country/Region", "date")) %>%
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%
  mutate(date = mdy(date))
```

Now that we have finished cleaning the global case data sets we will focus on cleaning the US case data sets. Here we got rid of unnecessary columns and renamed others to give them their own columns.

```
#Clean US case data
US_cases %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases")
```

```
## # A tibble: 3,819,906 x 13
##         UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
```

```
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
##  1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  2 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  3 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  4 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  5 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  6 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  7 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  8 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
##  9 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 10 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## # i 3,819,896 more rows
## # i 4 more variables: Long_ <dbl>, Combined_Key <chr>, date <chr>, cases <dbl>
```

```
##Fix columns
US_cases <- US_cases %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
```

```
#Clean US death case data
US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
```

```
#Now join US cases and US deaths together
US <- US_cases %>%
  full_join(US_deaths)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

In class, we were most interested in looking at US data. Here we look at it by state and calculate metrics such as deaths per a million people and cases per a million people.

```
#Focus on analyzing US as a whole
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 /Population) %>%
  select(Province_State, Country_Region, date, cases, deaths,   deaths_per_mill, Population) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

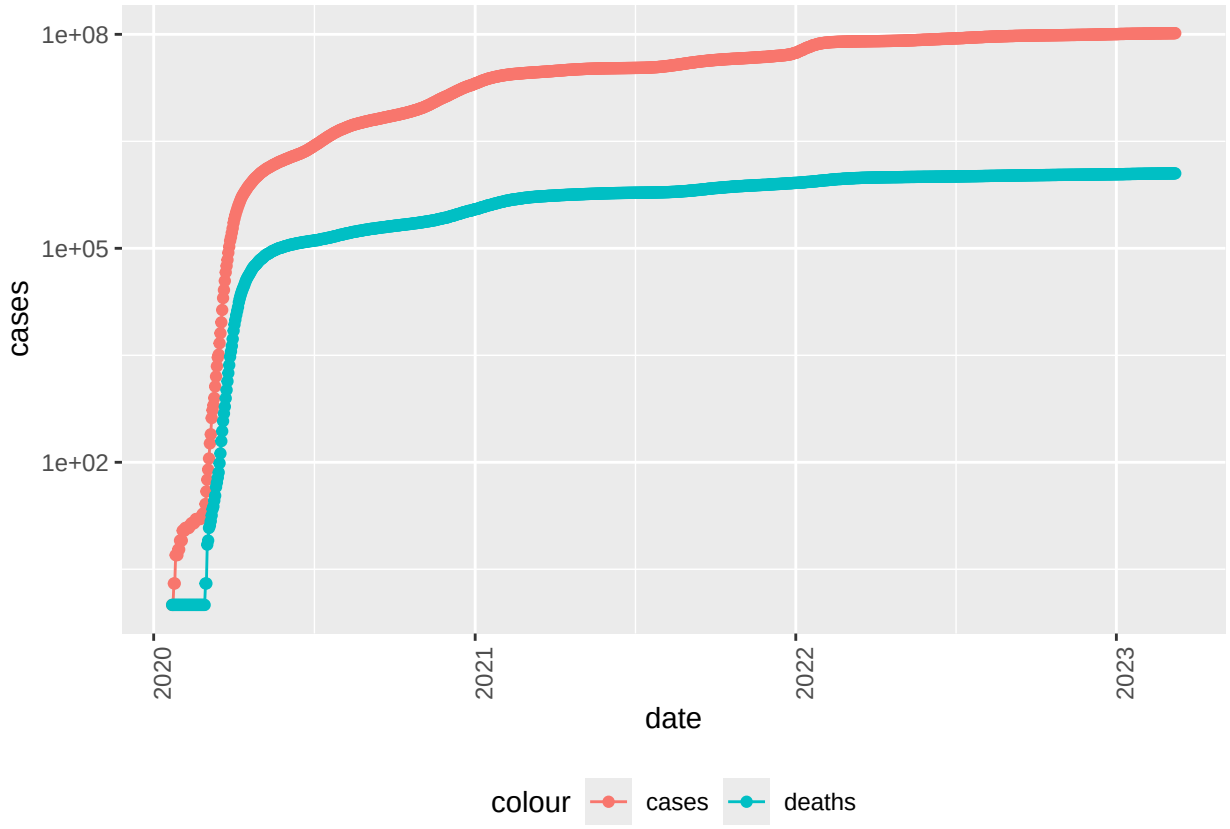
We also calculated the same metrics for the US as a whole by date.

```
#Now look at the total for the US
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>%
  select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

From here, we visualized the data by looking the rate of cases and rate of deaths for the entire US by time. As we can see in early 2020 case and death rates grew quickly and tapered off beginning in 2022.

```
#Now lets visualize the data
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x= date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle =90))
```




```
  labs(title = "COVID19 in US", y =NULL)
```

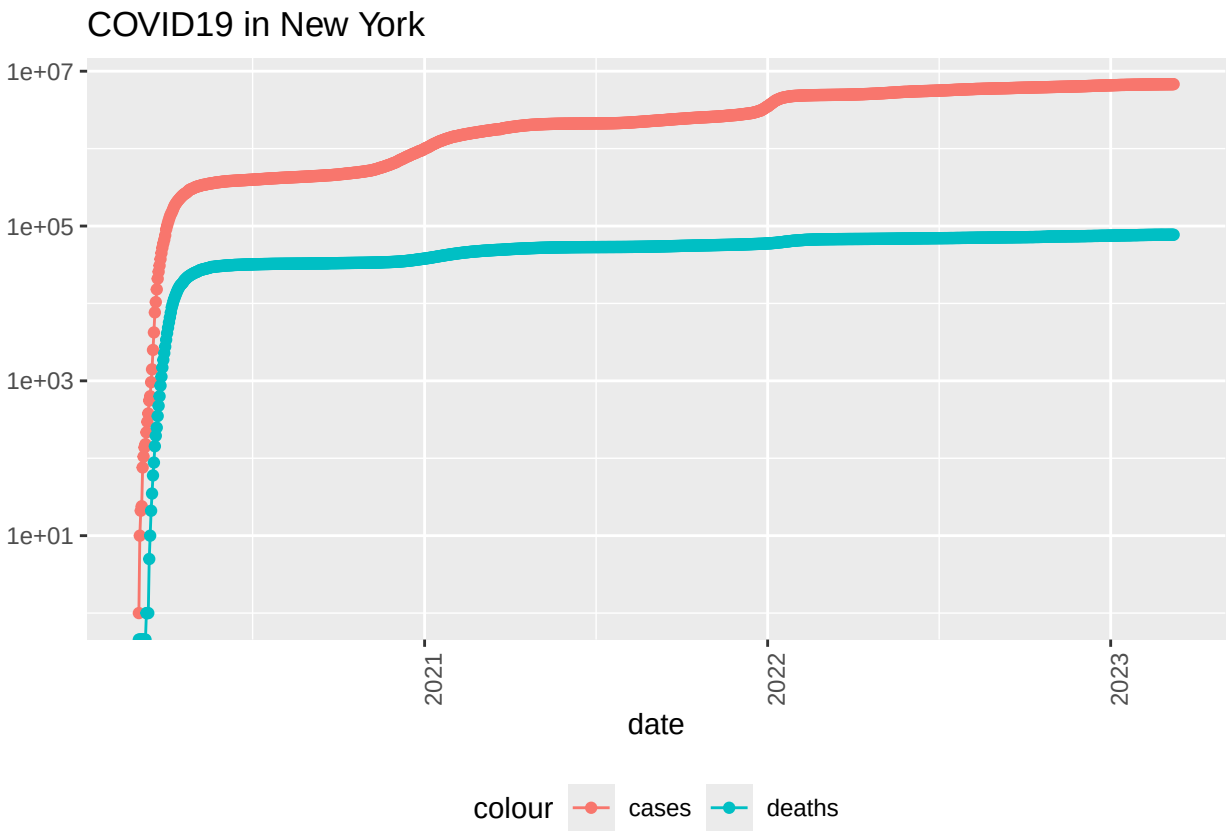
```
## $y
## NULL
##
## $title
## [1] "COVID19 in US"
##
## attr(,"class")
## [1] "labels"
```

After examining the country as a whole, we looked at it by state, specifically New York. New York follows a similar trend to the US as a whole for rates of cases and deaths for COVID19.

```
#Make Visualization by State
state <- "New York"
US_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0) %>%
  ggplot(aes(x= date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
```

```
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y =NULL)
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```



Next, we looked to see if we could predict deaths per thousand using cases per thousand within each state. Please not that some US states and territories in the data had unreported population numbers so for this analysis those provinces or states were removed.

```
##Create data set with state totals
US_state_totals <- US %>% filter(Population > 0) %>%
  group_by(Province_State, Country_Region) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_thou = deaths *1000 / Population) %>%
  mutate(cases_per_thou = cases *1000 / Population) %>%
  select(Province_State, Country_Region, cases, cases_per_thou, deaths, deaths_per_thou, Population) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Province_State'. You can override using
## the `.groups` argument.
```

```
US_state_totals
```

```
## # A tibble: 56 x 7
##    Province_State    Country_Region  cases cases_per_thou deaths deaths_per_thou
##    <chr>             <chr>           <dbl>          <dbl>  <dbl>           <dbl>
##  1 Alabama           US             8.73e8          156.  1.34e7           2.39
##  2 Alaska            US             1.53e8          180.  7.51e5           0.887
##  3 American Samoa    US             2.61e6           41.0 1.08e4           0.170
##  4 Arizona           US             1.33e9          160.  2.08e7           2.50
##  5 Arkansas          US             5.36e8          155.  7.72e6           2.24
##  6 California        US             6.16e9          136.  6.55e7           1.45
##  7 Colorado          US             9.22e8          140.  8.94e6           1.36
##  8 Connecticut       US             5.06e8          124.  8.91e6           2.19
##  9 Delaware          US             1.71e8          154.  2.09e6           1.88
## 10 District of Colu~ US             9.03e7          112.  1.14e6           1.41
## # i 46 more rows
## # i 1 more variable: Population <dbl>
```

To test this prediction hypothesis we used a linear model trying to predict deaths per thousand using cases per thousand by state. This model statistically significantly estimates that for every one-unit increase in cases per thousand that deaths per thousand will increase by 0.0167. The R-squared value, or variance for this model is 0.5066 indicating that 50.66% percent of the variance in deaths per thousand is explained by cases by thousand this is not a strong correlation. This is considered a moderate correlation.

```
#Make a linear model
mod <- lm(deaths_per_thou ~ cases_per_thou, data = US_state_totals)

summary(mod)
```

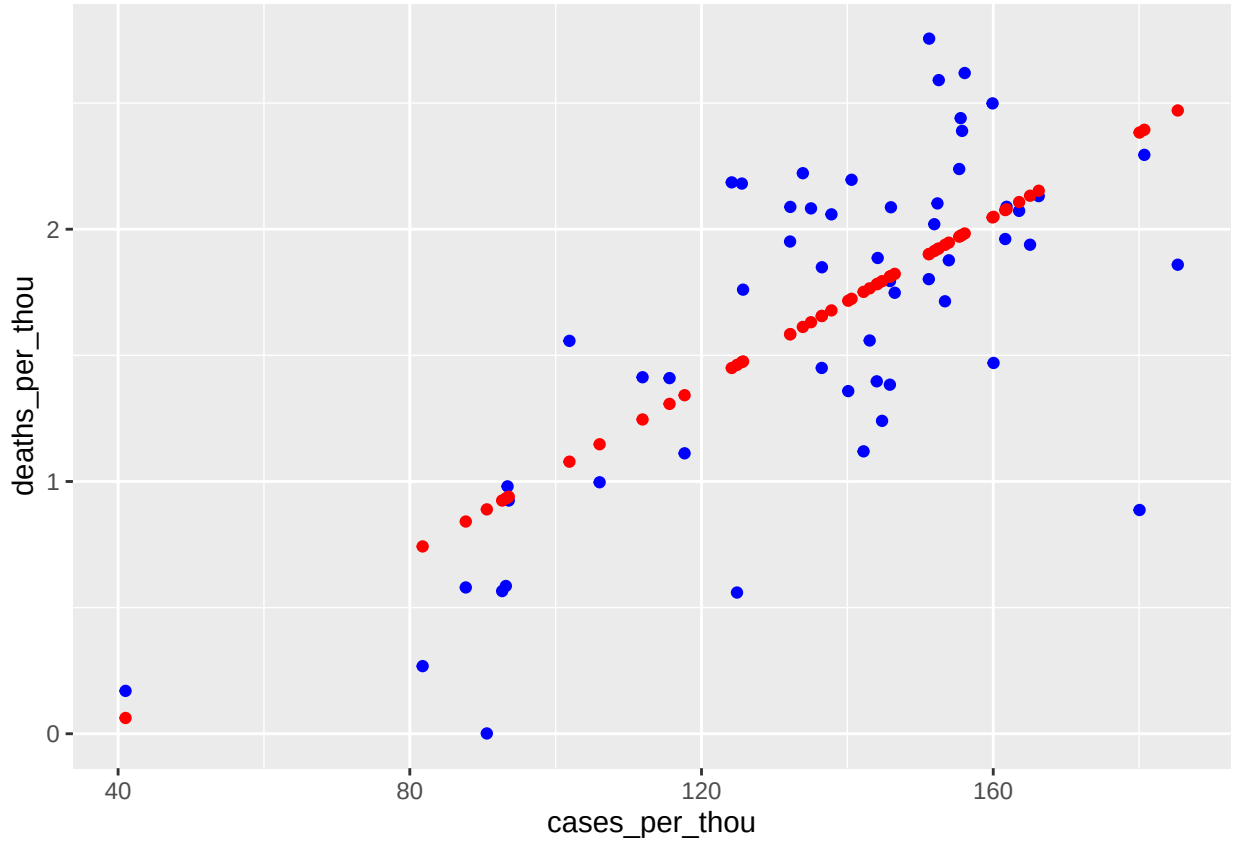
```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49631 -0.23809 -0.01629  0.37105  0.85379
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.622121   0.311577  -1.997   0.0509 .
## cases_per_thou  0.016691   0.002241   7.447 7.76e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4655 on 54 degrees of freedom
## Multiple R-squared:  0.5066, Adjusted R-squared:  0.4975
## F-statistic: 55.45 on 1 and 54 DF,  p-value: 7.76e-10
```

We then demonstrated this correlation graphically. As you can see here that cases per thousand strongly correlates with deaths per thousand at high and low cases per thousand but does not correlation strongly with deaths per thousand in the middle cases per thousand ranges in the graph below.

```r
#make a new data set with prediction values and graph
US_tot_w_pred <- US_state_totals %>% mutate(pred = predict(mod))
US_tot_w_pred
```

```
## # A tibble: 56 x 8
##    Province_State    Country_Region  cases cases_per_thou deaths deaths_per_thou
##    <chr>             <chr>           <dbl>          <dbl>  <dbl>           <dbl>
##  1 Alabama           US             8.73e8          156.  1.34e7           2.39
##  2 Alaska            US             1.53e8          180.  7.51e5           0.887
##  3 American Samoa    US             2.61e6           41.0 1.08e4           0.170
##  4 Arizona           US             1.33e9          160.  2.08e7           2.50
##  5 Arkansas          US             5.36e8          155.  7.72e6           2.24
##  6 California        US             6.16e9          136.  6.55e7           1.45
##  7 Colorado          US             9.22e8          140.  8.94e6           1.36
##  8 Connecticut       US             5.06e8          124.  8.91e6           2.19
##  9 Delaware          US             1.71e8          154.  2.09e6           1.88
## 10 District of Colu~ US             9.03e7          112.  1.14e6           1.41
## # i 46 more rows
## # i 2 more variables: Population <dbl>, pred <dbl>
```

```r
US_tot_w_pred %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



In addition to what we did in class, I was interested in seeing if population density of the state could be

used to predict deaths per thousand. To do this I first used our US data set from class and estimated the population density using the counties found in Admin2 column and assumed that population was consistent per state. For my model to work I had to filter out any rows that had non reported values due to any discrepancies in data collection.

The linear model attempting to predict deaths per thousand by population density per state as a linear function. This model statistically significantly predicts that for every one unit in population increase that deaths per thousand is expected to decrease by approximately 2.95. The R-squared value for this model is 0.0868 indicating that the correlation between deaths per thousand and population density is weak.

I then visualized this model using a scatter plot with a regression line and a box plot. The scatter plot shows that deaths per thousand and population density are negatively correlated with a tight correlation at the higher end of population density and low to no correlation at low population density. The box plot shows that variability in deaths per thousand decreases as population density increases.

In future studies I would be interested in looking at more potential variables that correlate with deaths per thousand. One idea that could be done with the current data set is using latitude and longitude to see if there is a correlation between deaths per thousand and relation to the equator.

This data report could have been effected by a few potential biases including over generalization and confirmation bias. Over generalization could have effected this report by me assuming random variables to be true or have impact on the data. For example, in calculating population density I assumed that all states had roughly the same population. Confirmation bias could also have effected this report by either me or the class focusing on pre-existing ideas for what could effect death cases per million or death cases per thousand and ignoring other potential predictor variables. To help offset these biases pitfalls I have included all data cleaning and analysis done in this report for the viewer to see and make their own conclusions.

```
library(ggplot2)
# Data preprocessing
# Aggregate data to the state level
state_data <- US %>%
  group_by(Province_State) %>%
  summarise(
    total_deaths = sum(deaths, na.rm = TRUE),
    population = first(Population),  # Assuming population is consistent per state
    county_count = n_distinct(Admin2)  # Count unique counties per state
  ) %>%
  mutate(
    deaths_per_thousand = (total_deaths / population) * 1000,
    population_density = population / county_count  # Approximate density per county
  )
# Filter out rows with NA
state_data <- state_data %>%
  filter(
    !is.na(deaths_per_thousand) & !is.infinite(deaths_per_thousand),
    !is.na(population_density) & !is.infinite(population_density)
  )

# Linear Model: Deaths per thousand vs. population density
model <- lm(deaths_per_thousand ~ population_density, data = state_data)
summary(model)
```

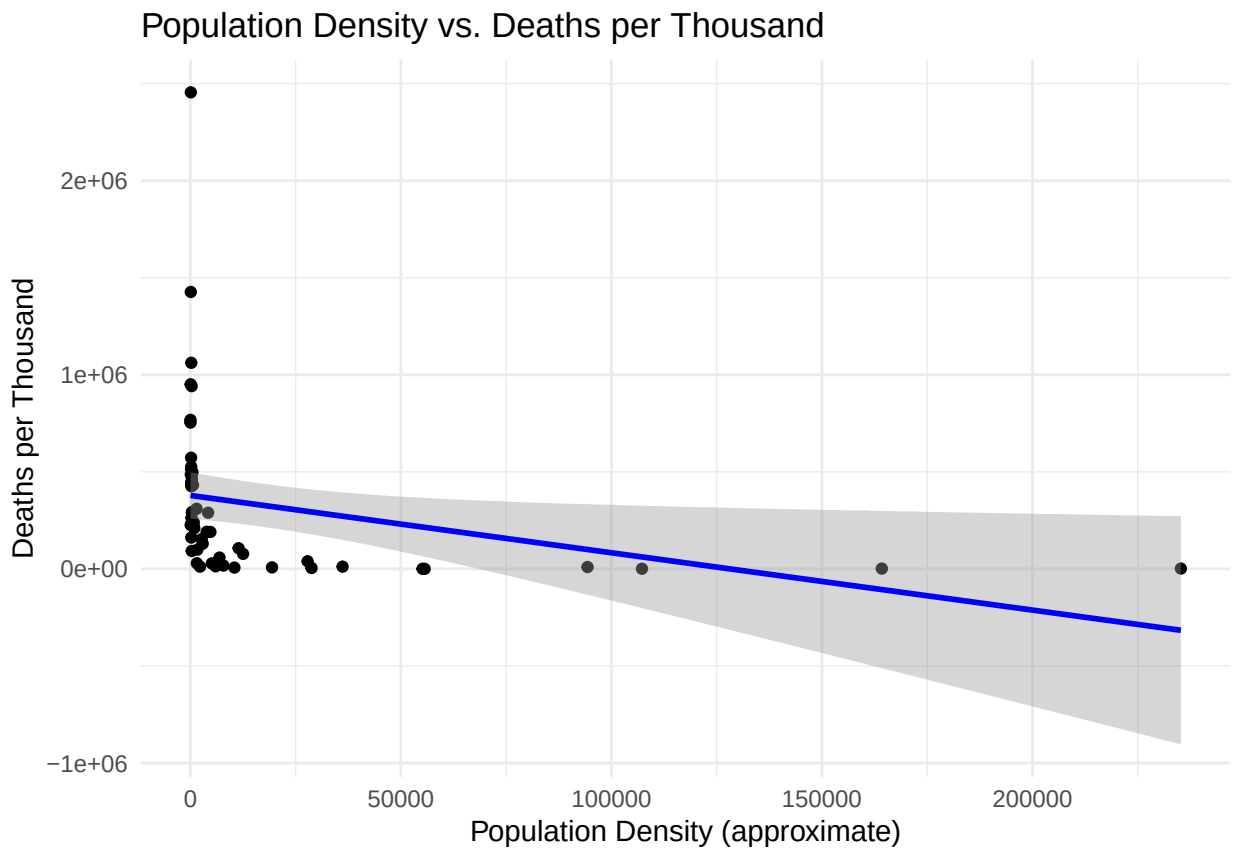
```
##
## Call:
## lm(formula = deaths_per_thousand ~ population_density, data = state_data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -360150 -248503 -115602  107041 2077165
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        378392.023  59599.443   6.349 5.02e-08 ***
## population_density     -2.953      1.316  -2.244    0.029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 410800 on 53 degrees of freedom
## Multiple R-squared:  0.08677,    Adjusted R-squared:  0.06954
## F-statistic: 5.036 on 1 and 53 DF,  p-value: 0.02903
```

```r
# Scatter plot with regression line
scatter_plot <- ggplot(state_data, aes(x = population_density, y = deaths_per_thousand)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(
    title = "Population Density vs. Deaths per Thousand",
    x = "Population Density (approximate)",
    y = "Deaths per Thousand"
  ) +
  theme_minimal()

print(scatter_plot)
```

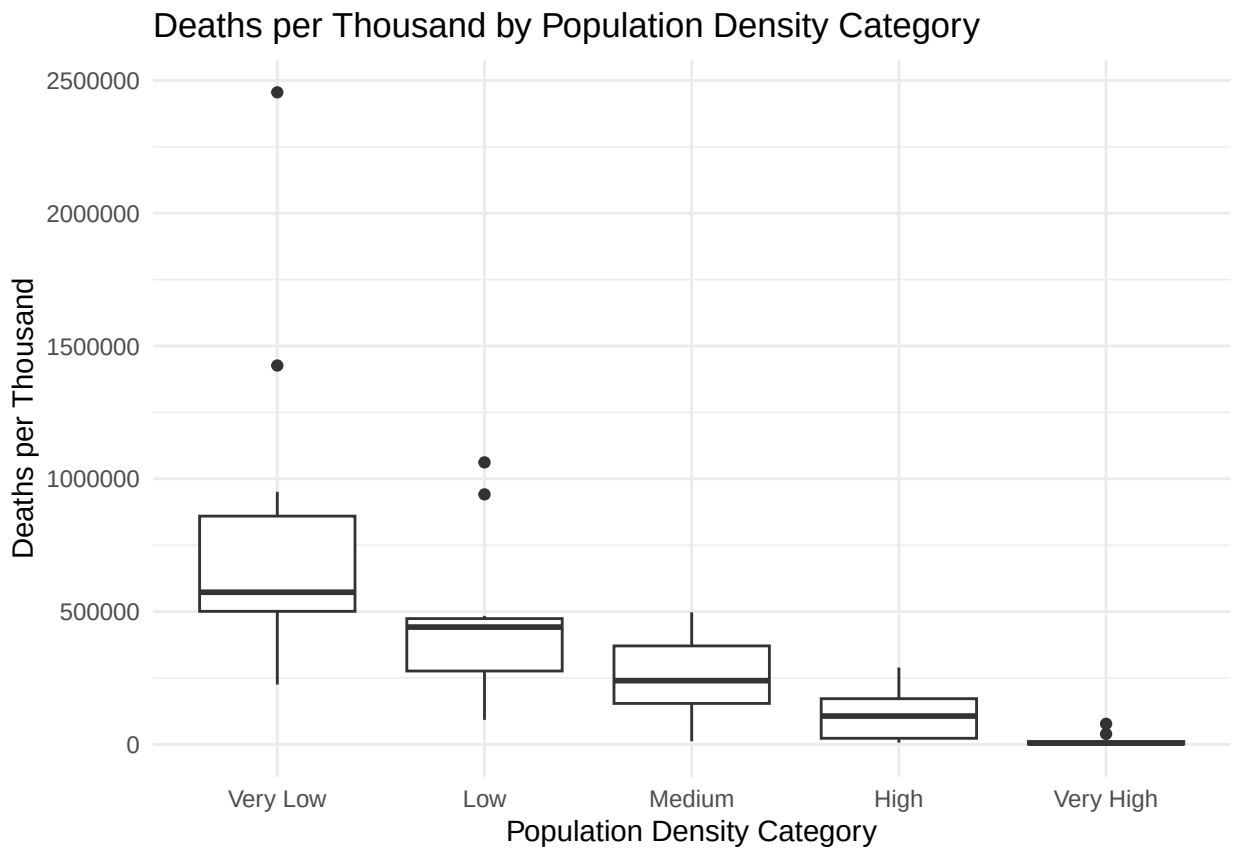
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Categorize states into population density quintiles
state_data <- state_data %>%
  mutate(
    density_category = cut(
      population_density,
      breaks = quantile(population_density, probs = seq(0, 1, 0.2), na.rm = TRUE),
      labels = c("Very Low", "Low", "Medium", "High", "Very High"),
      include.lowest = TRUE
    )
  )

# Boxplot of deaths per thousand by density category
boxplot <- ggplot(state_data, aes(x = density_category, y = deaths_per_thousand)) +
  geom_boxplot() +
  labs(
    title = "Deaths per Thousand by Population Density Category",
    x = "Population Density Category",
    y = "Deaths per Thousand"
  ) +
  theme_minimal()

print(boxplot)
```



Note: For reproducibility, please see the session info below.

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: x86_64-apple-darwin20
## Running under: macOS Ventura 13.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lmerTest_3.1-3  lme4_1.1-35.5   Matrix_1.7-0    lubridate_1.9.3
##  [5] forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2
##  [9] readr_2.1.5     tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.1
## [13] tidyverse_2.0.0
##
```

```
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4          generics_0.1.3      stringi_1.8.4
##  [4] lattice_0.22-6      hms_1.1.3           digest_0.6.37
##  [7] magrittr_2.0.3      evaluate_0.24.0     grid_4.4.1
## [10] timechange_0.3.0    fastmap_1.2.0       mgcv_1.9-1
## [13] fansi_1.0.6         scales_1.3.0        numDeriv_2016.8-1.1
## [16] cli_3.6.3           crayon_1.5.3        rlang_1.1.4
## [19] bit64_4.0.5         munsell_0.5.1       splines_4.4.1
## [22] withr_3.0.1         yaml_2.3.10         parallel_4.4.1
## [25] tools_4.4.1         tzdb_0.4.0          nloptr_2.1.1
## [28] minqa_1.2.8         colorspace_2.1-1    boot_1.3-30
## [31] curl_5.2.2          vctrs_0.6.5         R6_2.5.1
## [34] lifecycle_1.0.4     bit_4.0.5           vroom_1.6.5
## [37] MASS_7.3-60.2       pkgconfig_2.0.3     pillar_1.9.0
## [40] gtable_0.3.5        glue_1.7.0          Rcpp_1.0.13-1
## [43] highr_0.11          xfun_0.47           tidyselect_1.2.1
## [46] rstudioapi_0.16.0   knitr_1.48          farver_2.1.2
## [49] htmltools_0.5.8.1   nlme_3.1-164        labeling_0.4.3
## [52] rmarkdown_2.28      compiler_4.4.1
```