# Covid Data Final Project

Kendall

2025-11-28

**This report will evaluate global Covid 19 data to analyze case trends globally from 2020 to 2023, and how population affects the number of covid cases. The data is from github with links below.**

## Import Data

```
#read in datasets
US_cases = read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_co
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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.
```

```
global_cases = read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/css
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
US_deaths = read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

```
global_deaths = read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/c
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

# Clean Data

## Cleaning up global cases data

```
#remove lat and long columns, pivot date columns to rows
global_cases = global_cases %>%
    pivot_longer(
      cols = -c(`Province/State`, `Country/Region`, Lat, Long),
      names_to = "date",
      values_to = "cases",
      values_transform = list(cases = as.double)
    ) %>%
    mutate(
      date = as.Date(date, format = "%m/%d/%y")
    ) %>%
    select(-c(Lat, Long))
```

## Cleaning up global deaths data

```
#remove lat and long columns, pivot date columns to rows
global_deaths = global_deaths  %>%
    pivot_longer(
      cols = -c(`Province/State`, `Country/Region`, Lat, Long),
      names_to = "date",
      values_to = "deaths",
      values_transform = list(deaths = as.double)
    ) %>%
    mutate(
      date = as.Date(date, format = "%m/%d/%y")
    ) %>%
    select(-c(Lat, Long))
```

## Combining global cases and global death data

```
global = global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region')
```

```
## Joining with ‘by = join_by(‘Province/State‘, ‘Country/Region‘, date)‘
```

### Cleaning up US case data

```
US_cases = US_cases %>%
  #remove lat and long columns, pivot date columns to rows
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
```

### Cleaning up US death data

```
US_deaths = US_deaths %>%
  #remove lat and long columns, pivot date columns to rows
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
```

### Combining US cases and death data

```
US = US_cases %>%
  full_join(US_deaths)
```

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

### Add population to global data

```
#read in file that includes population data
population = read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_

#join the population data column to the global dataset
global = left_join(population, global, by = c("Country_Region"), relationship = "many-to-many")
global = global %>%
  #remove unneeded columns
  select(-c(UID:FIPS)) %>%
  select(-c(Admin2, Lat, Long_))
```

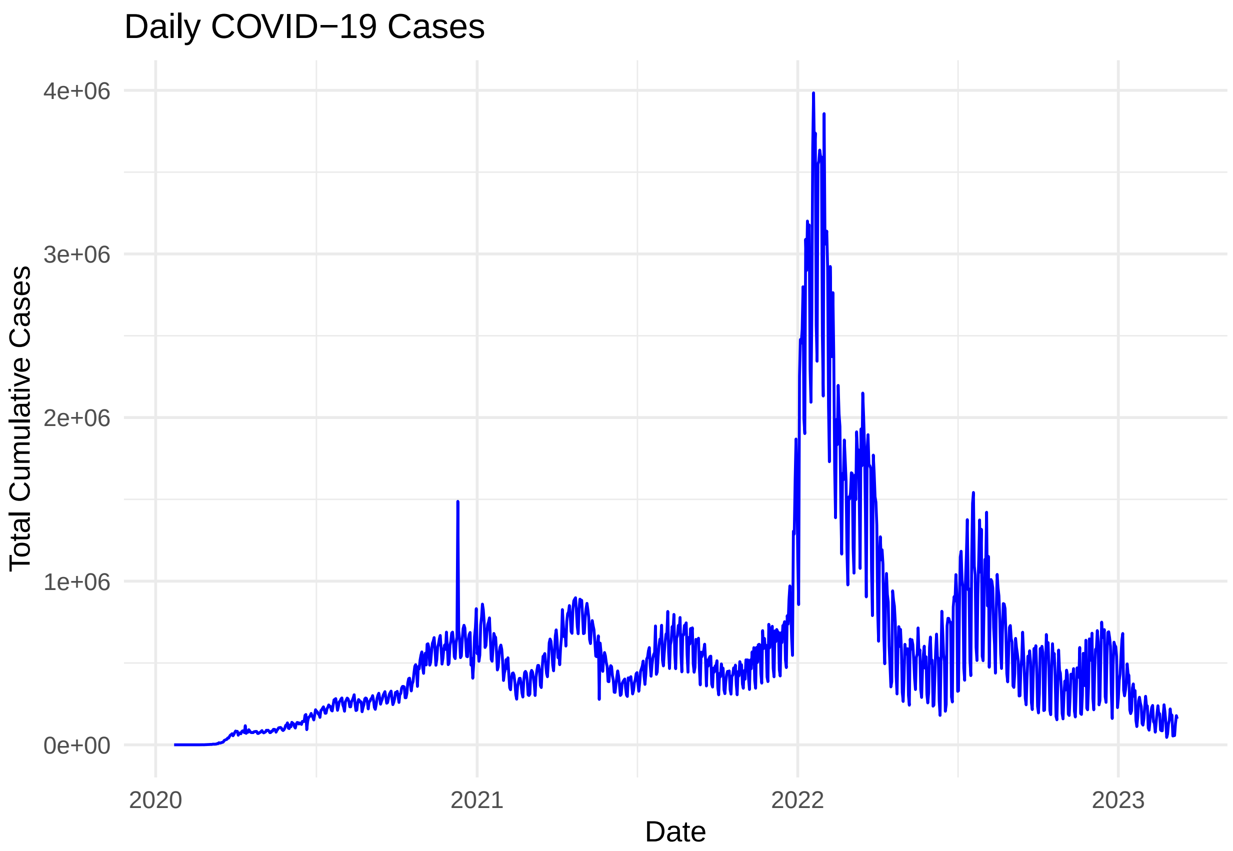
# Visualizing Data

## Global Data

```
#Changing the case data to be new cases, instead of cumulative
global_totals = global %>%
  arrange(Country_Region, date) %>%
  group_by(Country_Region) %>%
  mutate(new_cases = cases - lag(cases, default = 0)) %>%
  ungroup()

#Summarizing the data across all countries by date
global_totals = global_totals %>%
  group_by(date) %>%
  summarise(total_cases = sum(new_cases)) %>%
  select(date, total_cases)

#Plotting the number of new cases per day
ggplot(global_totals, aes(x = date, y = total_cases)) +
  geom_line(na.rm = TRUE, color = "blue") +
  labs(
    title = "Daily COVID-19 Cases",
    x = "Date",
    y = "Total Cumulative Cases"
  ) +
  theme_minimal()
```



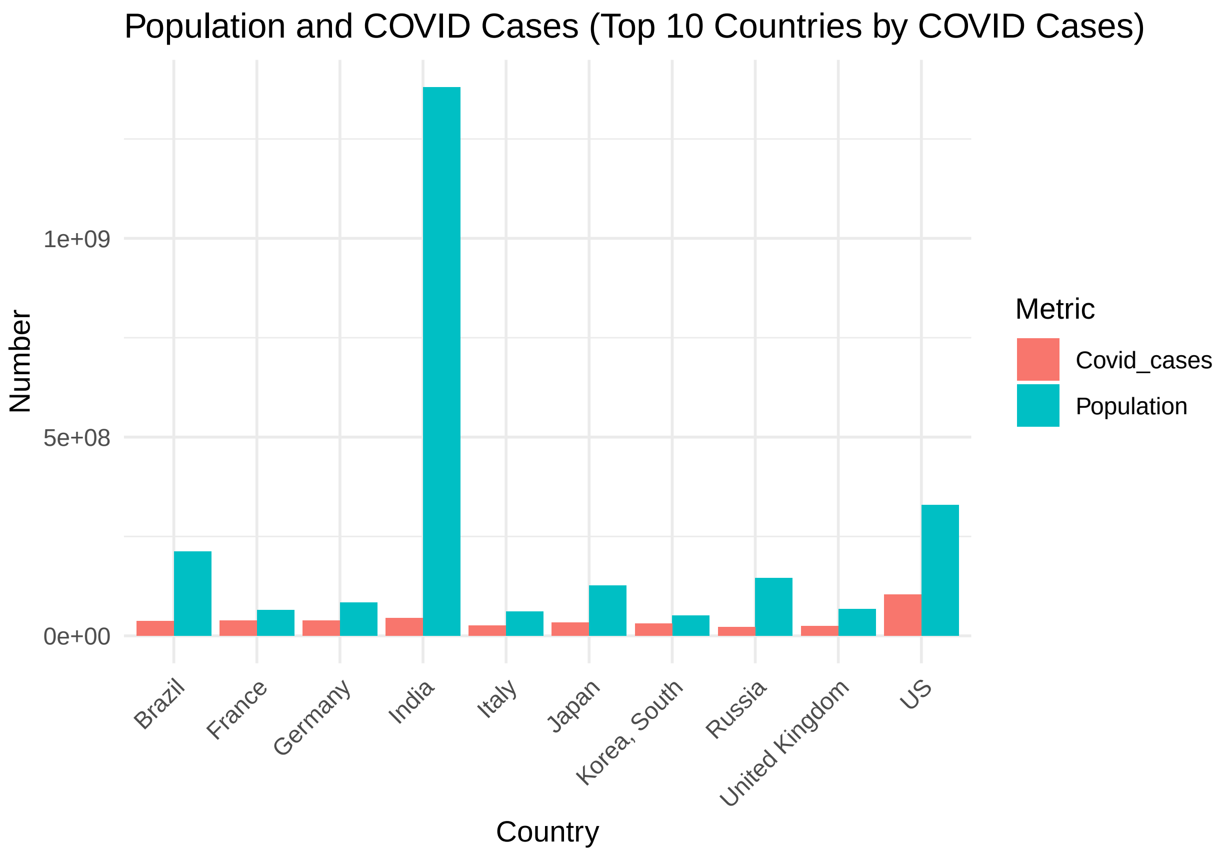
```
#Second visualization - bar chart of population and covid cases
#summarize data by country
by_country <- global %>%
     group_by(Country_Region) %>%
     summarise(
         Covid_cases = if(all(is.na(cases))) NA else max(cases, na.rm = TRUE),
        .groups = "drop"
     )

#Re-add population data to the dataset
by_country = left_join(population, by_country, by = c("Country_Region")) %>%
  # Keep only rows where Province_State is blank or NA
  filter(Province_State == "" | is.na(Province_State))

#Keeping only the top 10 countries in the dataset
top_countries <- by_country %>%
  arrange(desc(Covid_cases)) %>%
  slice_head(n = 10)    # keeps top 10 rows

#Pivot to long format
plot_data <- top_countries %>%
  pivot_longer(
    cols = c(Population, Covid_cases),
    names_to = "Metric",
    values_to = "Value"
  )
```

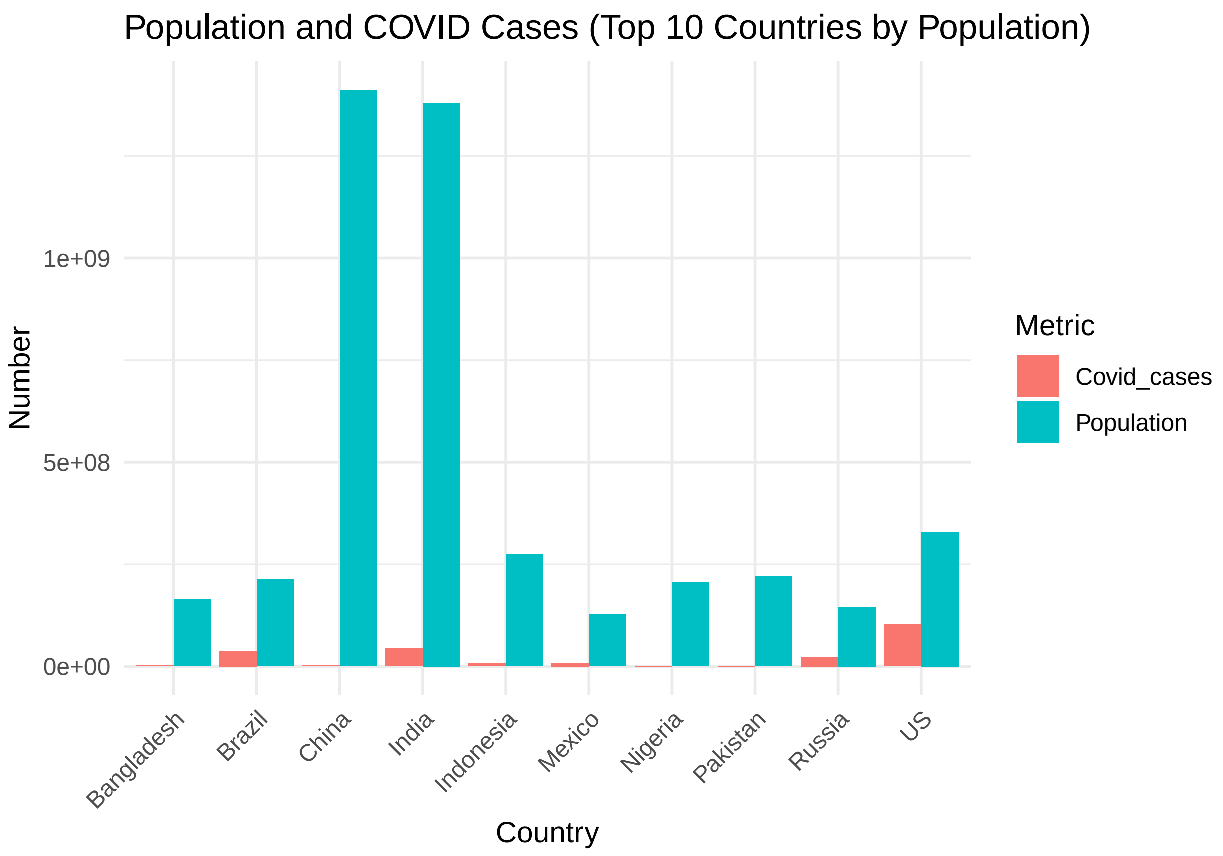
```r
#Create a side-by-side bar chart
ggplot(plot_data, aes(x = Country_Region, y = Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Population and COVID Cases (Top 10 Countries by COVID Cases)",
    x = "Country",
    y = "Number",
    fill = "Metric"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
#Create second dataset by keeping only top 10 countries by population
top_countries_pop <- by_country %>%
  arrange(desc(Population)) %>%
  slice_head(n = 10)    # keeps top 10 rows

#Pivot to long format
plot_data <- top_countries_pop %>%
  pivot_longer(
    cols = c(Population, Covid_cases),
    names_to = "Metric",
    values_to = "Value"
  )
```

```
#Create side-by-side bar chart
ggplot(plot_data, aes(x = Country_Region, y = Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Population and COVID Cases (Top 10 Countries by Population)",
    x = "Country",
    y = "Number",
    fill = "Metric"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



## Analyzing the Data

### I am going to analyze whether the population of a country can predict the number of Covid cases the country had

```
#create linear model of relationship between cases and population
model = lm(log(Covid_cases) ~ log(Population), data = by_country)
#print a summary of the model
summary(model)
```

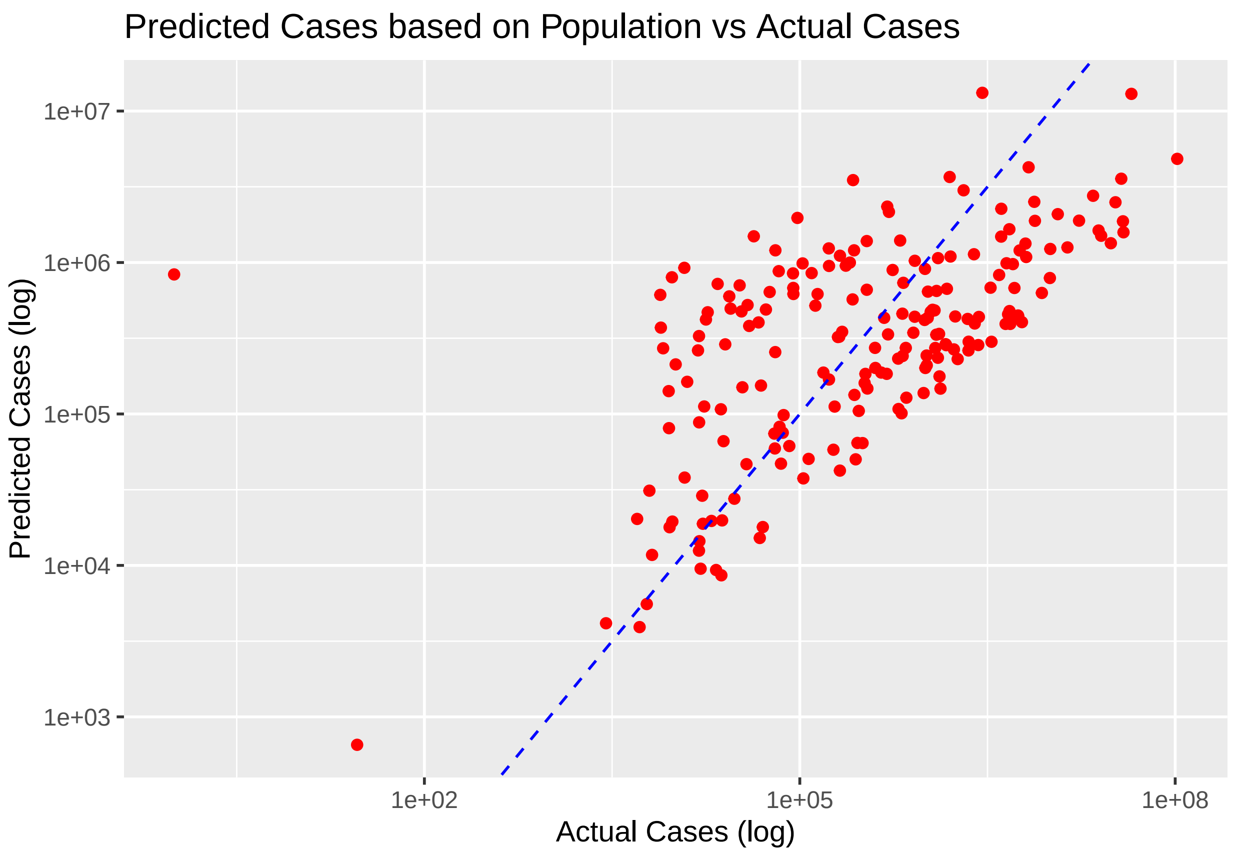
```
##
## Call:
## lm(formula = log(Covid_cases) ~ log(Population), data = by_country)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.634  -1.414   0.412   1.569   3.194
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.86481    1.01704   1.834   0.0683 .
## log(Population)  0.68969    0.06455  10.685   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.076 on 194 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.3705, Adjusted R-squared:  0.3672
## F-statistic: 114.2 on 1 and 194 DF,  p-value: < 2.2e-16
```

## Modeling the Data

```r
#add blank columns to global_totals dataset
by_country = by_country %>%
  mutate(pred_log_cases = NA) %>%
  mutate(pred_cases = NA)

#fill in the blank columns created above with data from the model
by_country$pred_log_cases = predict(model, newdata = by_country)
by_country$pred_cases = exp(by_country$pred_log_cases)

#Plot the predicted cases against the actual cases
ggplot(by_country, aes(x = Covid_cases, y = pred_cases)) +
  geom_point(color = "red", na.rm = TRUE) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "blue") +
  scale_x_log10() +
  scale_y_log10() +
  labs(title = "Predicted Cases based on Population vs Actual Cases",
       x = "Actual Cases (log)",
       y = "Predicted Cases (log)")
```


Predicted Cases based on Population vs Actual Cases
Predicted Cases (log)
1e+07
1e+06
1e+05
1e+04
1e+03
1e+02
1e+05
1e+08
Actual Cases (log)

# Conclusion

**This analysis focused on the global Covid19 dataset. The first visualization created was the total number of covid cases and deaths worldwide from 2020 to 2023. The second visualization charted number of covid cases against the population, first for the top 10 countries based on covid cases, and second for the top 10 countries based on population. A model was then created to determine if the population of a country could predict the number of covid cases that country had. This model had a p-value of 2.2e-16 and F-statistic of 114.2, which means population is a statistically significant indicator of covid cases, which logically makes sense. However, there are sources of bias that can affect the efficacy of the model. Each country may report their data differently, for example, looking at the plot of countries with the highest covid cases versus the plot of countries with the highest populations shows there are several high population countries that don't have corresponding high level of cases. This could be accurate, maybe they had more effective pandemic responses, or it could be due to differing reporting requirements. Understanding the bias around how these metrics are measured will be instrumental in how effective the model is. Looking at the first visualization as well, there seems to be a spike in cases in 2022, however this is also when widespread at-home testing became more available, so there was more data to pull from. This is another potential form of bias, as an increase in testing may look like it leads to an increase in cases, but the number of data points available could be influencing the spike as well. In summary, although population does appear to be a good indicator of the number of covid cases, it's important to remember hidden factors and biases that could be affecting the model and take those into account.**

```r
sessionInfo()
```

```
## R version 4.5.1 (2025-06-13 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
## 
## Matrix products: default
##   LAPACK version 3.12.1
## 
## locale:
## [1] LC_COLLATE=English_United States.utf8 
## [2] LC_CTYPE=English_United States.utf8   
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.utf8    
## 
## time zone: America/Denver
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] lubridate_1.9.4 forcats_1.0.1   stringr_1.6.0   dplyr_1.1.4    
##  [5] purrr_1.2.0     readr_2.1.6     tidyr_1.3.1     tibble_3.3.0   
##  [9] ggplot2_4.0.1   tidyverse_2.0.0
## 
## loaded via a namespace (and not attached):
##  [1] bit_4.6.0          gtable_0.3.6       crayon_1.5.3       compiler_4.5.1
##  [5] tidyselect_1.2.1   parallel_4.5.1     scales_1.4.0       yaml_2.3.10
##  [9] fastmap_1.2.0      R6_2.6.1           labeling_0.4.3     generics_0.1.4
## [13] curl_7.0.0         knitr_1.50         pillar_1.11.1      RColorBrewer_1.1-3
## [17] tzdb_0.5.0         rlang_1.1.6        stringi_1.8.7      xfun_0.54
## [21] S7_0.2.1           bit64_4.6.0-1      timechange_0.3.0   cli_3.6.5
## [25] withr_3.0.2        magrittr_2.0.4     digest_0.6.37      grid_4.5.1
## [29] vroom_1.6.6        rstudioapi_0.17.1  hms_1.1.4          lifecycle_1.0.4
## [33] vctrs_0.6.5        evaluate_1.0.5     glue_1.8.0         farver_2.1.2
## [37] rmarkdown_2.30     tools_4.5.1        pkgconfig_2.0.3    htmltools_0.5.8.1
```