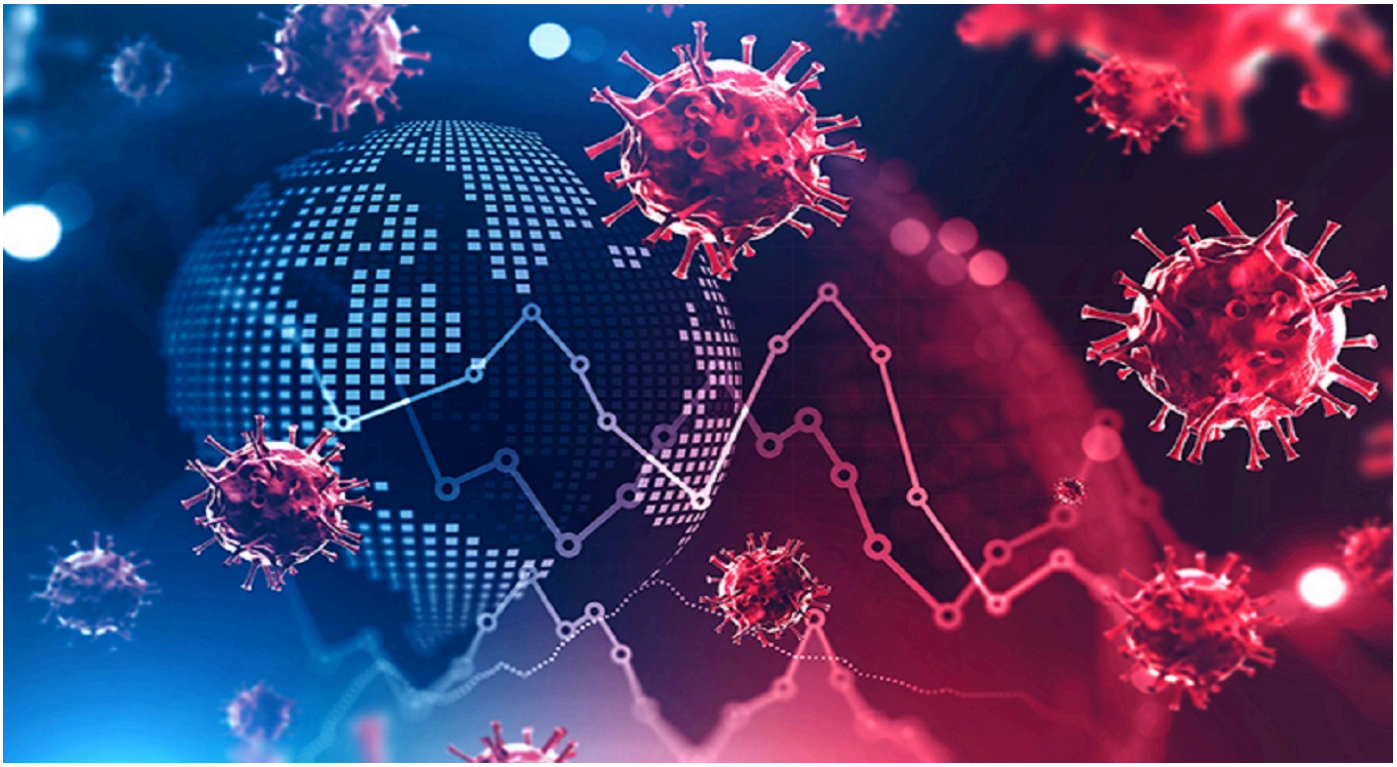


Unmasking the Numbers: COVID-19's Effect on Jobs, Markets, and Growth

Analyzing the Economic Impact of COVID-19 in the US Through Statistics

University of Victoria
Thamy Soares, Kush Manek, Sarah Sandmeier, Xueying Lin
Data Management and Visualization - STAT 321 - Professor Arjun Banik
April 4th, 2025

Index

Introduction and Motivation:

The COVID-19 pandemic had a major impact on the US economy, disrupting jobs, financial markets, and overall economic growth. Unemployment surged, businesses shut down, and stock markets became highly volatile. Understanding these effects is essential for policymakers and businesses to develop recovery strategies and prepare for future crises.

Research Question: *How did COVID-19 impact jobs, financial markets, and economic growth in the US?*

The "Unmasking the Numbers: COVID-19's Impact on Jobs, Markets, and Growth" project systematically examines the economic effects of the pandemic in the USA. The project will use various methods to visually present data, combining information from stock market indices (VGT & VHT), labour market indicators (unemployment rates and initial claims), and COVID-19 case data by state and demographics. The goal is to identify patterns, trends, and key insights to understand how the pandemic has affected different areas of the economy.

Understanding these economic shifts is essential, as the pandemic's long-term consequences are still unfolding. By identifying patterns in job losses, market reactions, and overall economic performance, our research provides valuable insights to inform decision-making in the post-pandemic era.

Data Cleaning and Wrangling

A wide range of data sources was used for the analysis, including monthly price data for the VGT (Tech) and VHT (Health) sectors from Yahoo Finance, dating back to 2015, as well as U.S. unemployment rates and Initial Claims for Unemployment (ICSA) starting from 2015, sourced from FRED. The COVID-19 case and death datasets were collected from the CDC, aggregated by week, region, and demographics.

The datasets were carefully assessed for structure, including variable types, timeframes, and missing values. The stock market dataset included variables like Open, High, Low, Close, Adjusted Close, and Volume. Unemployment data was aggregated by month with variables for Date and Rate. The ICSA data included dates and the number of claims, while the COVID-19 data contained state-level cases and deaths with some demographic details. Some of the COVID-19 data had missing or suppressed values that required further cleaning.

Data cleaning involved handling missing values, such as replacing missing COVID-19 deaths with zeros. Duplication was minimized, and outliers were kept, as they could offer valuable insights. The dates were normalized to the YYYY-MM-DD format for easy time-series analysis. The ICSA dataset was restricted to the last 10 years to match the other datasets' period. Feature selection included the creation of an "average movement" variable from stock data, while eliminating Open, Close, and Adjusted Close. Columns related to COVID-19 jurisdiction and historical case/death data were omitted. Unemployment claims data older than necessary for the current labor market trends was excluded. The information was converted into a numeric format, rounded to two decimal places for consistency. Categorical variables like State in the Covid-19 dataset were coded to simplify analysis and visualization. The cleaned data was then stored in CSV format to support future model-building and prediction tasks.

Data Visuals

The data visuals in this analysis are designed to provide clear, accessible insights into the impact of COVID-19 on the US economy. We created various kinds of plots using ggplot2 package in R to represent key datasets, allowing for an easier interpretation of complex information.

A line graph was created to compare the performance of VGT (Vanguard Growth ETF) and VHT (Vanguard Health Care ETF) over time. The graph highlights the stock value variations throughout the pandemic, with the COVID-19 period highlighted in red. This visualization provides an overview of how the pandemic affected the stock market, particularly in the healthcare and technology sectors.

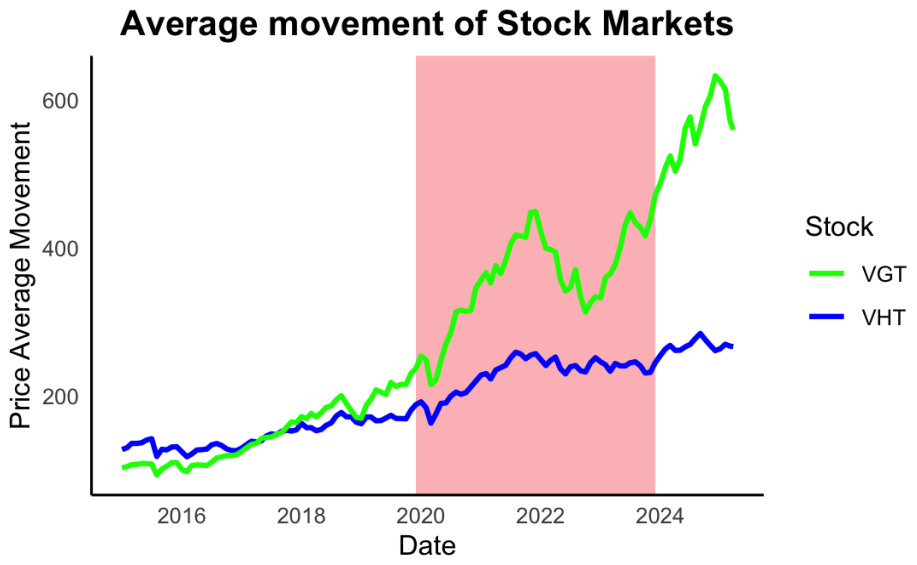


Figure 1: line graphs of Average movement of VGT and VHT over time

We plotted the unemployment claims and unemployment rate, with the COVID-19 period highlighted in red, highlighting the sharp increase during the early months of the COVID-19 pandemic and the subsequent decline as the economy recovered. Both charts show a surge in the rate during the pandemic's early stages, followed by stabilization as recovery efforts took place.

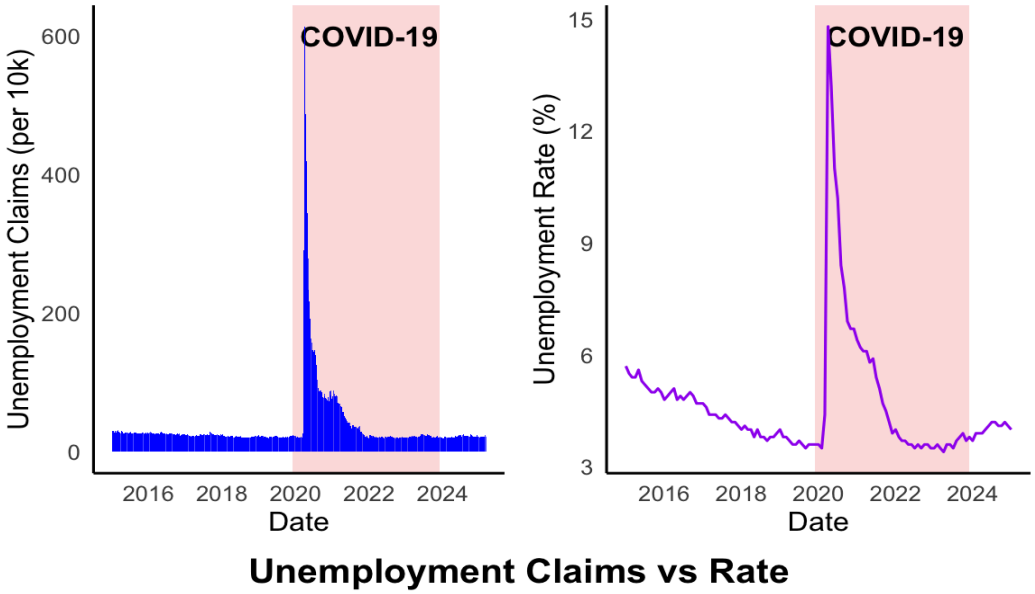
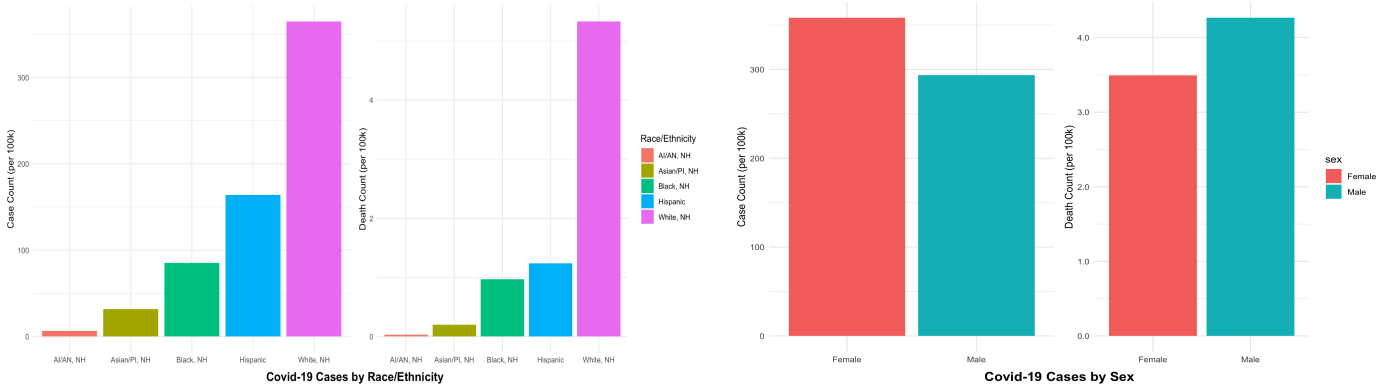
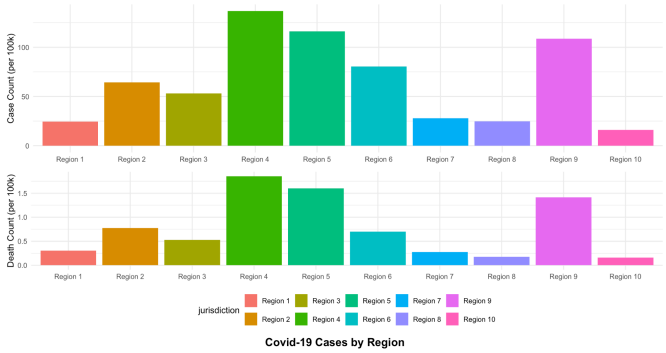


Figure 2: Bar graph of Unemployment Claims and Line graph of Unemployment Rate with red shaded part corresponding to the COVID-19 period.

Bar plots were created to visualize COVID-19 cases and deaths across different demographics, including race/ethnicity, sex, and age. These graphs help highlight the disparities in COVID-19's impact on various demographic groups, using a logarithmic scale for easier interpretation.

We developed separate bar graphs to show infections and deaths by age group, sex, race and region, allowing for a clear comparison of how different factors influenced the spread and severity of COVID-19.





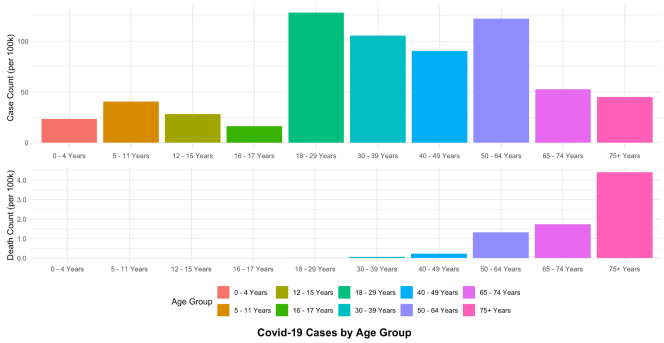


Figure 2: Bar graphs of COVID-19 cases and deaths by Race/Ethnicity, Sex, Region and age group.

Figure 3: Table of states by Region/Jurisdiction

States by Region

Region	States
Region 1	ME, MA, RI, NH, PR
Region 2	VT, NY, CT
Region 3	NJ, DE, PA, VI
Region 4	MD, NC, SC, VA, WV, AL, FL, GA
Region 5	LA, MS, TX
Region 6	KY, MI, OH, TN
Region 7	IL, IN, WI
Region 8	AR, IA, MN, MO, NE, ND, SD
Region 9	AK, AZ, CA, GU, HI, ID, MT, NV, OR, WA
Region 10	CO, KS, NM, OK, UT, WY

The geographic distribution of COVID-19 cases and deaths shows that some areas were more severely affected. Overall, these figures provide clear, data-driven insights into the pandemic's impact on health, the economy, and financial markets, with each visualization offering enough information to understand the key findings.

We created a Shiny app that visualizes COVID-19 data by US state, providing an interactive map where users can explore case and death counts across different states. The app can be accessed through this link: https://ssandmaier.shinyapps.io/app_attempt/. By selecting a specific time period, such as January 23rd, 2020, users can identify the first reported COVID-19 case in the United States, which occurred in Washington State. Furthermore, by adjusting the date range from January 2020 to March 5th, 2020, the app reveals the initial COVID-19-related deaths, with Washington State showing the highest number of early fatalities, followed by California.

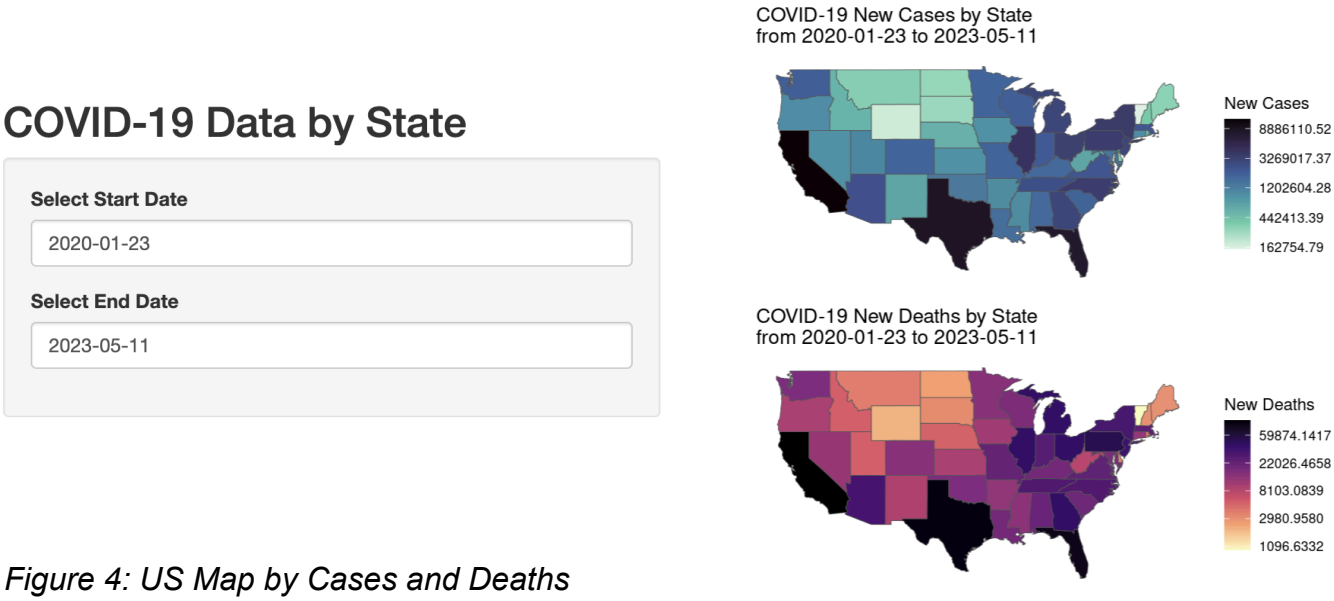


Figure 4: US Map by Cases and Deaths

Statistical Analysis

To better understand the relationship between the spread of the virus and economic fluctuations, we performed linear regression analyses and developed time series models. These methods allowed us to examine trends before, during, and after the pandemic, and to quantify the extent to which COVID-19 influenced the economy over time.

The analysis of VGT (technology) and VHT (healthcare) shows their responses to the COVID-19 pandemic. Before the pandemic (2015 – 2019), both ETFs experienced steady growth with low volatility. During COVID-19 (2020 – 2023), both saw an initial drop but quickly recovered. VGT due to an increased demand for tech caused by remote work policies, and VHT driven by the need for healthcare services and vaccines. Post-pandemic (2024 – now), both sectors continued to grow with reduced volatility, demonstrating long-term strength and resilience in tech and healthcare.

We created a SARIMA model to forecast the average movement of VGT and VHT. For VGT, the model predicted a 99% confidence interval of $492–$569 for May 2025 with the current value at $511. For VHT, the forecasted range for May 2025 was $246–$279, with the current value at $259.

The model performed well, showing strong accuracy in predicting future values.

We performed a simple linear regression to examine the relationship between the unemployment rate and unemployment claims, finding that for every 1% increase in the unemployment rate, unemployment claims rise by about 1,044,116 people. The model explained about 71.7% of the variation in claims (adjusted R-squared = 0.717). So the model shows that higher unemployment is strongly related to higher unemployment claims. We also observed that, before COVID-19, both unemployment claims and unemployment rate were stable. In early 2020, claims spiked and the unemployment rate surged to around 15% due to mass layoffs and business shutdowns. After mid-2020, claims declined but remained higher than pre-pandemic levels. By 2023–2024, the unemployment rate dropped back to around 3.5%–4% as claims began to stabilize, indicating that although the spike was sharp, the recovery was gradual and both indicators eventually returned to levels seen before the pandemic.

We used multiple linear regression to study how age and sex affect COVID-19 death rates, with females aged 0–4 years as the baseline. The results showed that people over 40 had higher death rates, and males had higher death rates than females. For example, individuals aged 75+ had 8.94 more deaths per 100,000 than the baseline, and males had 0.577 more deaths than females. The model explained about 11.3% of the variation in death rates. Older people, especially those 75+, had higher mortality due to weakened immune systems, and males are at higher risk, possibly due to biological factors. Both age and sex are important in understanding the death rate differences.

We conducted a state-level analysis of COVID-19 cases and deaths. California had the highest numbers with over 12 million cases and about 101,000 deaths, while the Republic of the Marshall Islands reported the lowest with 6,000 cases and 9 deaths. Pennsylvania had the highest death rate at 1.430%, compared to the Republic of the Marshall Islands' 0.110%. A simple linear regression revealed a strong relationship ($R^2 = 0.942$, $p < 0.0001$), with every 1,000 cases linked to roughly 10 deaths. Overall, our findings suggest that while case counts vary widely, they are a

strong predictor of the total number of deaths.

We studied COVID-19 cases and deaths by sex. Females had more total cases (360 million), but males experienced more deaths (3.97 million). Additionally, males had a higher average death rate (1.54 per 100,000) and a higher maximum death rate (660.68 per 100,000) compared to females. This suggests that while females had more cases, males suffered more severe outcomes in terms of death rates.

We built a multiple linear regression model to examine how age, sex, and race affect COVID-19 case rates, using children aged 0–4, females, and AI/AN as the baseline. The analysis showed that adults aged 18–39 had the highest increases in case rates, with estimates of +60.246 and +58.051 per 100k, while males had lower rates than females (−13.218). In terms of race, the Hispanic group experienced the highest increase (+16.007), whereas Asian/PI (−45.568), White (−32.024), and Black (−22.084) groups had lower rates compared to the baseline. All effects were statistically significant ($p < 0.001$), although the model only explained about 2.9% of the variation in case rates ($R^2 = 0.029$). Overall, the findings suggest that younger adults and Hispanics have higher case rates, but these factors together account for only a small portion of the variation observed.

Conclusions

The pandemic left long-term impacts on the US work patterns, financial markets, and economic growth. The statistical models and data visualization performed in this study shed light on how all the indicators for the economy performed in response to the crisis.

In conclusion, our study shows that COVID-19 had a strong and complex impact on the economy and public health. We used several methods, such as time series analysis and different types of regression to understand these effects. Our research on technology and healthcare stocks revealed that both sectors suffered during the pandemic but then slowly began to recover. We also looked at unemployment data and found that many people lost their jobs during the crisis, with unemployment claims and rates spiking sharply before gradually returning to more normal levels. In addition, our analysis showed a strong correlation between the number of COVID-19 cases and the number of deaths. We discovered that older people and men had a higher death rate, while younger adults and Hispanic groups were the most infected.

All of these findings help us see the big picture of how the pandemic affected different parts of our society. Understanding these trends is very important to prepare for future circumstances. By studying the effects of COVID-19 carefully, we can better predict and prepare for similar crises, which will help protect our economy and improve public health. This research shows that good analysis can guide policymakers to make better decisions and develop stronger plans to face future challenges.

References(APA)

Centers for Disease Control and Prevention. (n.d.). Weekly United States COVID-19 cases and deaths by state. CDC. https://data.cdc.gov/Case-Surveillance/Weekly-United-States-COVID-19-Cases-and-Deaths-by-/pwn4-m3yp/about_data

Centers for Disease Control and Prevention. (n.d.). COVID-19 weekly cases and deaths by age, race/ethnicity, and sex. CDC. https://data.cdc.gov/Public-Health-Surveillance/COVID-19-Weekly-Cases-and-Deaths-by-Age-Race-Ethni/hrdz-jaxc/about_data

Federal Reserve Bank of St. Louis. (n.d.). Unemployment rate data (UNRATE). FRED. https://fred.stlouisfed.org/series/UNRATE

Federal Reserve Bank of St. Louis. (n.d.). Initial claims (ICSA). FRED. https://fred.stlouisfed.org/series/ICSA

Yahoo Finance. (n.d.). VGT stock - Tech index. Yahoo Finance. https://ca.finance.yahoo.com/quote/VGT/history/?frequency=1mo&period1=1422576000

Yahoo Finance. (n.d.). VHT stock data. Yahoo Finance. https://ca.finance.yahoo.com/quote/VHT/

Walmsley, T., Rose, A., & Wei, D. (2020). The impacts of the coronavirus on the economy of the United States. *Economics of Disasters and Climate Change, 5*(1), 1–52. https://doi.org/10.1007/s41885-020-00080-1

OpenAI. (2023). ChatGPT (Mar 14 version). https://chat.openai.com/chat

Appendix

R-Code Data Cleaning

```r
library(readr)
library(dplyr)
library(ggplot2)
library(tidyverse)
library(lubridate)

library(quantmod)
getSymbols("VGT", src = "yahoo", from = "2015-01-01", periodicity = "monthly")
VGT_rounded <- round(VGT, 2)
head(VGT_rounded)

stock_data <- data.frame(Date = index(VGT_rounded), coredata(VGT_rounded))

# Save as CSV
write.csv(stock_data, "VGT_stock_data.csv", row.names = FALSE)

print("File saved as VGT_stock_data.csv")

getSymbols("VHT", src = "yahoo", from = "2015-01-01", periodicity = "monthly")
VHT_rounded <- round(VHT, 2)
head(VHT_rounded)

stock_data <- data.frame(Date = index(VHT_rounded), coredata(VHT_rounded))

# Save as CSV
write.csv(stock_data, "VHT_stock_data.csv", row.names = FALSE)

print("File saved as VHT_stock_data.csv")
```

For VGT data

```r
library(quantmod)
getSymbols("VGT", src = "yahoo", from = "2015-01-01", periodicity = "monthly")
VGT_rounded <- round(VGT, 2)

stock_data <- data.frame(Date = index(VGT_rounded), coredata(VGT_rounded))

# Save as CSV
write.csv(stock_data, "VGT_stock_data.csv", row.names = FALSE)

print("File saved as VGT_stock_data.csv")

VGT <- read_csv("VGT_stock_data.csv");
VGT

VGT$average_movement <- (VGT$VGT.High+VGT$VGT.Low)/2
VGT
colnames(VGT)
```

```
VGT$VGT.Open <- NULL
VGT$VGT.Close <- NULL
VGT$VGT.Adjusted <- NULL
VGT$VGT.Volume <- NULL
VGT_rounded <- VGT %>%
  mutate(across(where(is.numeric), ~ round(.x, 2)))
write_csv(VGT_rounded,"VGT_stock_data_cleaned.csv")

getSymbols("VHT", src = "yahoo", from = "2015-01-01", periodicity = "monthly")
VHT_rounded <- round(VHT, 2)

stock_data <- data.frame(Date = index(VHT_rounded), coredata(VHT_rounded))

# Save as CSV
write.csv(stock_data, "VHT_stock_data.csv", row.names = FALSE)

print("File saved as VHT_stock_data.csv")
```

For VHT data

```
VHT <- read_csv("VHT_stock_data.csv");

VHT$average_movement <- (VHT$VHT.High+VHT$VHT.Low)/2
VHT

VHT$VHT.Open <- NULL
VHT$VHT.Close <- NULL
VHT$VHT.Adjusted <- NULL
VHT$VHT.Volume <- NULL
VHT_rounded <- VHT %>%
  mutate(across(where(is.numeric), ~ round(.x, 2)))
write_csv(VHT_rounded,"VHT_stock_data_cleaned.csv")
```

For unemployment data

```
unemployment_data <- read_csv("unemployment rate data.csv");
unemployment_data
```

ICSA Unemployment claim data

```
ICSA_data <- read_csv("ICSA_Initial claims.csv")
ICSA_data

ICSA_data <- ICSA_data[-c(1:2505), ]
ICSA_data
ICSA_data <- ICSA_data %>%
  mutate(observation_date = as.Date(observation_date, format = case_when(
  grepl("/", observation_date) ~ "%m/%d/%Y",
  grepl("-", observation_date) ~ "%m-%d-%Y",
  grepl(",", observation_date) ~ "%B %d, %Y",
  TRUE ~ NA_character_
```

```
 )))
ICSA_data
write_csv(ICSA_data,"ICSA_Initial claims_cleaned.csv")
```

removed the previous data as we are only taking data from last 10 years into consideration for the analysis.

|Weekly_United_States_COVID-19|

```
covid_data <-
read_csv("Weekly_United_States_COVID-19_Cases_and_Deaths_by_State_-_ARCHIVED_20250304.csv")
covid_data
covid_data <- covid_data %>%
 mutate(date_updated = as.Date(date_updated, format = case_when(
 grepl("/", date_updated) ~ "%m/%d/%Y",
 grepl("-", date_updated) ~ "%m-%d-%Y",
 grepl(",", date_updated) ~ "%B %d, %Y",
 TRUE ~ NA_character_
 )))

covid_data <- covid_data %>%
 mutate(start_date = as.Date(start_date, format = case_when(
 grepl("/", start_date) ~ "%m/%d/%Y",
 grepl("-", start_date) ~ "%m-%d-%Y",
 grepl(",", start_date) ~ "%B %d, %Y",
 TRUE ~ NA_character_
 )))


covid_data <- covid_data %>%
 mutate(end_date = as.Date(end_date, format = case_when(
 grepl("/", end_date) ~ "%m/%d/%Y",
 grepl("-", end_date) ~ "%m-%d-%Y",
 grepl(",", end_date) ~ "%B %d, %Y",
 TRUE ~ NA_character_
 )))
covid_data

covid_data$state <- as.factor(covid_data$state)
covid_data

covid_data$new_historic_cases <- NULL
covid_data$new_historic_deaths <- NULL
covid_data

covid_data <- covid_data %>%
 mutate(
       circuit = case_when(
       state %in% c("ME", "MA", "NH", "PR", "RI") ~ "Region 1",
       state %in% c("CT", "NY", "VT", "NYC") ~ "Region 2",
       state %in% c("DE", "NJ", "PA", "VI") ~ "Region 3",
       state %in% c("MD", "NC", "SC", "VA", "WV", "AL", "FL", "GA", "DC") ~ "Region 4",
       state %in% c("LA", "MS", "TX","AS") ~ "Region 5",
       state %in% c("KY", "MI", "OH", "TN") ~ "Region 6",
       state %in% c("IL", "IN", "WI") ~ "Region 7",
```

```
        state %in% c("AR", "IA", "MN", "MO", "NE", "ND", "SD") ~ "Region 8",
        state %in% c("AK", "AZ", "CA", "GU", "HI", "ID", "MT", "NV", "OR", "WA","FSM", "MP", "PW", "RMI") ~
"Region 9",
        state %in% c("CO", "KS", "NM", "OK", "UT", "WY") ~ "Region 10"
        )
 )

covid_data$jurisdiction_type <- NULL

covid_data <- covid_data %>%
 rename("Jurisidiction" = circuit)

# View the updated dataset
head(covid_data)

write_csv(covid_data,"Weekly_United_States_COVID-19_Cases_and_Deaths_by_State_-_ARCHIVED_202
50304_cleaned.csv")

covid_19_data <-
read_csv("COVID-19_Weekly_Cases_and_Deaths_by_Age__Race_Ethnicity__and_Sex_-_ARCHIVED_20250
310.csv")
covid_19_data

library(tidyr)

covid_19_data <- covid_19_data %>%
 mutate(
        death_count_suppressed = replace_na(death_count_suppressed, 0),
        death_crude_rate_suppressed_per_100k = replace_na(death_crude_rate_suppressed_per_100k,
0)
 )

covid_19_data <- covid_19_data %>%
 mutate(end_of_week = as.Date(end_of_week, format = case_when(
 grepl("/", end_of_week) ~ "%m/%d/%Y",
 grepl("-", end_of_week) ~ "%m-%d-%Y",
 grepl(",", end_of_week) ~ "%B %d, %Y",
 TRUE ~ NA_character_
 )))

covid_19_data

write.csv(covid_19_data,"COVID-19_Weekly_Cases_and_Deaths_by_Age__Race_Ethnicity__and_Sex_-_A
RCHIVED_20250310_cleaned.csv")
```

R-Code Visualizations

```
library(ggplot2)
library(dplyr)
library(ggtext)
library(tidyr)
```

```
library(flextable)
library(officer)
```

saving and uploading files

```
VGT <- read.csv("VGT_stock_data_cleaned.csv")
VHT <- read.csv("VHT_stock_data_cleaned.csv")
ICSA <- read.csv("ICSA_Initial_claims_cleaned.csv")
Unrate <- read.csv("unemployment_rate_data.csv")
CovState <-
read.csv("Weekly_United_States_COVID-19_Cases_and_Deaths_by_State_-_ARCHIVED_20250304_
cleaned.csv")
CovGroup <-
read.csv("COVID-19_Weekly_Cases_and_Deaths_by_Age__Race_Ethnicity__and_Sex_-_ARCHIVED_
20250310_cleaned.csv"")
```

Graph of VGT vs VHT

```
vgt_vht <- ggplot() +
  geom_rect(aes(xmin = as.Date("2020-03-01"), xmax = as.Date("2023-12-01"), ymin =
-Inf, ymax = Inf),
            fill = "red", alpha = 0.3) +
  geom_text(aes(x = as.Date("2022-01-01"), y = max(VHT$average_movement, na.rm =
TRUE),
                label = "COVID-19", color = "black"),
            size = 4, fontface = "bold", vjust = -15) +
  geom_line(data = VHT, aes(x = Date, y = average_movement, color = "VHT", group = 1),
linewidth = 1) +
  geom_line(data = VGT, aes(x = Date, y = average_movement, color = "VGT", group = 1),
linewidth = 1) +
  labs(title = "Average movement of Stock Markets",
       x = "Date",
       y = "Price Average Movement",
       color = "Stock") +
  scale_color_manual(values = c("VHT" = "blue", "VGT" = "green")) +
  theme_minimal() +
  theme(
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    plot.caption = element_markdown(hjust = 0.5, size = 14, face = "plain"),
    plot.caption.position = "plot",
    plot.margin = margin(t = 10, r = 10, b = 50, l = 10)
  )
```

Sex comparison graphs

```
sexg <- ggplot(CovGroup, aes(x = sex, y = case_count_suppressed, fill= sex)) +
  geom_bar(stat = "identity") +
  labs(title = "Sex", y = "Case Count (per 100k)", x = NULL) +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

sexd <- ggplot(CovGroup, aes(x = sex, y = death_count_suppressed, fill= sex)) +
  geom_bar(stat = "identity") +
  labs(title = "Sex", y = "Death Count (per 100k)", x = NULL) +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))


grid.arrange(
  sexg + theme(legend.position = "none") + labs(title = NULL),
  sexd + labs(title = NULL),
  ncol = 2,
  bottom = textGrob("Covid-19 Cases by Sex", gp = gpar(fontsize = 14, fontface =
"bold"))
)
```

Race comparison graphs

```
raceg <- ggplot(CovGroup, aes(x = race_ethnicity_combined, y = case_count_suppressed,
fill= race_ethnicity_combined)) +
  geom_bar(stat = "identity") +
  labs(title = "Race/Ethnicity", y = "Case Count (per 100k)", x = NULL, fill =
"Race/Ethnicity") +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

raced <- ggplot(CovGroup, aes(x = race_ethnicity_combined, y = death_count_suppressed,
fill= race_ethnicity_combined)) +
  geom_bar(stat = "identity") +
  labs(title = "Race/Ethnicity", y = "Death Count (per 100k)", x = NULL, fill =
"Race/Ethnicity") +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5,face = "bold"))

grid.arrange(
  raceg + theme(legend.position = "none") + labs(title = NULL),
  raced + labs(title = NULL),
  ncol = 2,
  bottom = textGrob("Covid-19 Cases by Race/Ethnicity", gp = gpar(fontsize = 14,
fontface = "bold"))
)
```

Age comparison graphs

```
custom_order <- c("0 - 4 Years","5 - 11 Years","12 - 15 Years","16 - 17 Years",
                  "18 - 29 Years", "30 - 39 Years","40 - 49 Years","50 - 64 Years",
                  "65 - 74 Years", "75+ Years")

CovGroup <- CovGroup %>%
  mutate(age_group = factor(age_group, levels = custom_order))


ageg <- ggplot(CovGroup, aes(x = age_group, y = case_count_suppressed, fill=
age_group)) +
  geom_bar(stat = "identity") +
  labs(title = "Age Group", y = "Case Count (per 100k)", x = NULL, fill = "Age Group")
+
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

aged <- ggplot(CovGroup, aes(x = age_group, y = death_count_suppressed, fill=
age_group)) +
  geom_bar(stat = "identity") +
  labs(title = "Age Group", y = "Death Count (per 100k)", x = NULL, fill = "Age
Group") +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

grid.arrange(
  ageg + theme(legend.position = "none") + labs(title = NULL),
  aged + theme(legend.position = "bottom") + labs(title = NULL),
  nrow = 2,
  bottom = textGrob("Covid-19 Cases by Age Group", gp = gpar(fontsize = 14, fontface =
"bold"))
)
```

Jurisdiction comparison graphs

```
reg_order <- c("Region 1","Region 2","Region 3","Region 4","Region 5","Region 6",
               "Region 7","Region 8","Region 9","Region 10")

CovGroup <- CovGroup %>%
  mutate(jurisdiction = factor(jurisdiction, levels = reg_order))

regg <- ggplot(CovGroup, aes(x = jurisdiction, y = case_count_suppressed, fill=
jurisdiction)) +
  geom_bar(stat = "identity") +
  labs(title = "Region Group", y = "Case Count (per 100k)", x = NULL, fill =
"jurisdiction") +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

```
regd <-ggplot(CovGroup, aes(x = jurisdiction, y = death_count_suppressed, fill=
jurisdiction)) +
  geom_bar(stat = "identity") +
  labs(title = "Region ", y = "Death Count (per 100k)", x = NULL, fill =
"jurisdiction") +
  scale_y_continuous(labels = label_number(scale = 1e-5)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

grid.arrange(
  regg + theme(legend.position = "none") + labs(title = NULL),
  regd + theme(legend.position = "bottom") + labs(title = NULL),
  nrow = 2,
  bottom = textGrob("Covid-19 Cases by Region", gp = gpar(fontsize = 14, fontface =
"bold"))
)
```

ICSA claims vs Unemployment rate graphs

```
unem_bar <- ggplot() +
  geom_rect(aes(xmin = as.Date("2020-03-01"), xmax = as.Date("2023-12-01"), ymin =
-Inf, ymax = Inf),
            fill = "red", alpha = 0.15) +
  geom_text(aes(x = as.Date("2022-01-01"),
                y = max(ICSA$icsa, na.rm = TRUE)),
            label = "COVID-19", color = "black",
            size = 4, fontface = "bold", vjust = 1) +
  geom_col(data = ICSA, aes(x = observation_date, y = icsa), fill = "blue") +
  labs(title = "ICSA Unemployment Claims", x = "Date", y = "Case Amount (in 10k)") +
  scale_y_continuous(labels = label_number(scale = 1e-4)) +
  theme_minimal()


unrate_line <- ggplot() +
  geom_rect(aes(xmin = as.Date("2020-03-01"), xmax = as.Date("2023-12-01"), ymin =
-Inf, ymax = Inf),
            fill = "red", alpha = 0.15) +
  geom_text(aes(x = as.Date("2022-01-01"),
                y = max(Unrate$UNRATE, na.rm = TRUE)),
            label = "COVID-19", color = "black",
            size = 4, fontface = "bold", vjust = 1) +
  geom_line(data = Unrate, aes(x = observation_date, y = UNRATE), color = "purple") +
  labs(title = "Unemployment Rate", x = "Date", y = "Unemployment Rate (%)") +
  theme_minimal()

grid.arrange(
  unem_bar + theme(legend.position = "none") + labs(title = NULL),
  unrate_line + labs(title = NULL),
  nrow = 1,
  bottom = textGrob("Unemployment Claims vs Rate", gp = gpar(fontsize = 14, fontface =
```

```
"bold"))
)
```

Region to State table

```
region_list <- list(
  "Region 1" = c("ME", "MA", "RI", "NH", "PR"),
  "Region 2" = c("VT", "NY", "CT"),
  "Region 3" = c("NJ", "DE", "PA", "VI"),
  "Region 4" = c("MD", "NC", "SC", "VA", "WV", "AL", "FL", "GA"),
  "Region 5" = c("LA", "MS", "TX"),
  "Region 6" = c("KY", "MI", "OH", "TN"),
  "Region 7" = c("IL", "IN", "WI"),
  "Region 8" = c("AR", "IA", "MN", "MO", "NE", "ND", "SD"),
  "Region 9" = c("AK", "AZ", "CA", "GU", "HI", "ID", "MT", "NV", "OR", "WA"),
  "Region 10" = c("CO", "KS", "NM", "OK", "UT", "WY")
)

region_df <- region_list %>%
  tibble::enframe(name = "Region", value = "State") %>%
  unnest(cols = c(State))

region_wide <- region_df %>%
  group_by(Region) %>%
  mutate(row = row_number()) %>%
  pivot_wider(names_from = Region, values_from = State) %>%
  select(-row)
region_table <- flextable(region_wide) %>%
  theme_vanilla() %>%
  set_caption("States by Region") %>%
  border_remove() %>%
  border(i = 1, border.top = fp_border(width = 1)) %>%
  bg(j = seq(1, ncol(region_wide), by = 2), bg = "#e0f7fa") %>%
  bg(j = seq(1, ncol(region_wide), by = 2), part = "header", bg = "#e0f7fa") %>%
  autofit() %>%
  align(align = "center", part = "all")
```

Shiny app code

```
CovState <-
read.csv("Weekly_United_States_COVID-19_Cases_and_Deaths_by_State_-_ARCHIVED_20250304_
cleaned.csv")

#change format to dates
CovState$date_updated <- as.Date(CovState$date_updated, format = "%Y-%m-%d")

#change states to their names
CovState <- CovState %>%
  mutate(state = case_when(
```

```
    state == "AK" ~ "alaska",
    state == "AL" ~ "alabama",
    state == "AR" ~ "arkansas",
    state == "AZ" ~ "arizona",
    state == "CA" ~ "california",
    state == "CO" ~ "colorado",
    state == "CT" ~ "connecticut",
    state == "DE" ~ "delaware",
    state == "FL" ~ "florida",
    state == "GA" ~ "georgia",
    state == "ID" ~ "idaho",
    state == "IL" ~ "illinois",
    state == "IN" ~ "indiana",
    state == "IA" ~ "iowa",
    state == "KS" ~ "kansas",
    state == "KY" ~ "kentucky",
    state == "LA" ~ "louisiana",
    state == "ME" ~ "maine",
    state == "MD" ~ "maryland",
    state == "MA" ~ "massachusetts",
    state == "MI" ~ "michigan",
    state == "MN" ~ "minnesota",
    state == "MS" ~ "mississippi",
    state == "MO" ~ "missouri",
    state == "MT" ~ "montana",
    state == "NE" ~ "nebraska",
    state == "NV" ~ "nevada",
    state == "NH" ~ "new hampshire",
    state == "NJ" ~ "new jersey",
    state == "NM" ~ "new mexico",
    state == "NY" ~ "new york",
    state == "NC" ~ "north carolina",
    state == "ND" ~ "north dakota",
    state == "OH" ~ "ohio",
    state == "OK" ~ "oklahoma",
    state == "OR" ~ "oregon",
    state == "PA" ~ "pennsylvania",
    state == "RI" ~ "rhode island",
    state == "SC" ~ "south carolina",
    state == "SD" ~ "south dakota",
    state == "TN" ~ "tennessee",
    state == "TX" ~ "texas",
    state == "UT" ~ "utah",
    state == "VT" ~ "vermont",
    state == "VA" ~ "virginia",
    state == "WA" ~ "washington",
    state == "WV" ~ "west virginia",
    state == "WI" ~ "wisconsin",
    state == "WY" ~ "wyoming",
    TRUE ~ state
  ))

# delete states that dont appear in US map: AS, DC, FSM, GU, HI, MP, NYC, PR, PW, RMI,
VI
```

```
CovState <- CovState %>%
  filter(!state %in% c("AS", "DC","FSM","GU","HI","MP","NYC","PR","PW","RMI","VI"))

#US map data
us_map <- map_data("state")

#save cleaned data as csv

write.csv(CovState, "/Users/sarita/Documents/UVIC/STAT/STAT 321/visuals/CovState.csv",
row.names = FALSE)
```

Shiny app creation

```
CovState <- read.csv("CovState.csv")

library(leaflet)
library(tidyr)
library(sf)
library(dplyr)
library(maps)
library(shiny)
library(viridis)
library(gridExtra)
library(tmap)
library(rnaturalearth)
library(rnaturalearthdata)

# function for map
covid_den_map <- function(start_date, end_date) {
  start_date <- as.Date(start_date)
  end_date <- as.Date(end_date)
  filtered_data <- CovState[CovState$date_updated >= start_date &
CovState$date_updated <= end_date, ]
  total_new_cases_by_state <- tapply(filtered_data$new_cases, filtered_data$state,
sum, na.rm = TRUE)
  total_new_deaths_by_state <- tapply(filtered_data$new_deaths, filtered_data$state,
sum, na.rm = TRUE)
  us_states <- st_as_sf(maps::map("state", fill = TRUE, plot = FALSE))
  us_states$state <- tolower(us_states$ID)
  us_states$new_cases <- total_new_cases_by_state[us_states$state]
  us_states$new_deaths <- total_new_deaths_by_state[us_states$state]
  map_cases <- ggplot(data = us_states) +
    geom_sf(aes(fill = new_cases)) +
    scale_fill_viridis_c(option = "mako", trans = "log", direction = -1) +
    theme_minimal() +
    labs(title = paste("COVID-19 New Cases by State\nfrom", start_date, "to",
end_date),
         fill = "New Cases") +
    theme(
      axis.text = element_blank(),
      axis.ticks = element_blank(),
      axis.title = element_blank()
    )
  map_deaths <- ggplot(data = us_states) +
```

```
    geom_sf(aes(fill = new_deaths)) +
    scale_fill_viridis_c(option = "magma", trans = "log",direction = -1) +
    theme_minimal() +
    labs(title = paste("COVID-19 New Deaths by State\nfrom", start_date, "to",
end_date),
         fill = "New Deaths") +
    theme(
      axis.text = element_blank(),
      axis.ticks = element_blank(),
      axis.title = element_blank()
    )
  grid.arrange(map_cases, map_deaths, nrow = 2)
}


ui <- fluidPage(
  titlePanel("COVID-19 Data by State"),
  sidebarLayout(
    sidebarPanel(
      dateInput(
        inputId = "start_date",
        label = "Select Start Date",
        value = "2020-01-23",
        min = "2020-01-23",
        max = "2023-05-11"
      ),
      dateInput(
        inputId = "end_date",
        label = "Select End Date",
        value = "2022-01-01",
        min = "2020-01-23",
        max = "2023-05-11"
      )
    ),
    mainPanel(
      plotOutput("covid_map")
    )
  )
)

server <- function(input, output) {
  output$covid_map <- renderPlot({

    covid_den_map(input$start_date, input$end_date)

  })
}

shinyApp(ui = ui, server = server)
```

R-Code Statistical Analysis

```r
suppressMessages(library(tidyverse))
library(readxl)


# 1. COVID case data
covid_data <-
read_csv("COVID-19_Weekly_Cases_and_Deaths_by_Age__Race_Ethnicity__and_Sex_-_ARCHIVED_
20250310_cleaned.csv",show_col_types = FALSE)

covid_data <- covid_data %>%
  select(-`...1`)

covid_data <- covid_data %>%
 rename(
    Jurisdiction = jurisdiction,
    AgeGroup = age_group,
    Sex = sex,
    Race = race_ethnicity_combined,
    CaseSuppressed = case_count_suppressed,
    DeathSuppressed = death_count_suppressed,
    CaseRate = case_crude_rate_suppressed_per_100k,
    DeathRate = death_crude_rate_suppressed_per_100k
  ) %>%
  # Convert end_of_week as a string to a real Date.
  mutate(Date = as.Date(end_of_week, format = "%m/%d/%Y")) %>%
  # Remove raw columns end_of_week
  select(-end_of_week) %>%
  # arrange date
  arrange(Date)

#move date to first
covid_data <- covid_data %>%
select(Date, everything())

covid_data <- covid_data %>%
  mutate(across(where(is.numeric), ~replace_na(., 0)))

# 2. Initial unemployment claims
claims_data <-  read_csv("ICSA_Initial claims_cleaned.csv", show_col_types = FALSE)

#rename
claims_data <- claims_data %>%
  rename(
    Date = observation_date
  )

claims_data <- claims_data %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%Y"))

# 3. Unemployment rate data
unemp_data <- read_csv("unemployment rate data.csv", show_col_types = FALSE)
```

```r
# 4. VGT
vgt_data <- read_csv("VGT_stock_data_cleaned.csv", show_col_types = FALSE)

# 5. VHT
vht_data <- read_csv("VHT_stock_data_cleaned.csv", show_col_types = FALSE)

# state case and death
state_case_death <-
read_csv("Weekly_United_States_COVID-19_Cases_and_Deaths_by_State_-_ARCHIVED_20250304_
cleaned.csv",
                              show_col_types = FALSE)

state_case_death <- state_case_death %>%
  rename(
    DateUpdate = date_updated,
    State = state,
    StartDate = start_date,
    EndDate = end_date,
    TotalCases = tot_cases,
    NewCases = new_cases,
    TotalDeaths = tot_deaths,
    NewDeaths = new_deaths
  )
```

Forecast of the VGT and VHT dataset

```r
library(forecast)
library(tidyverse)

VGT$Date <- as.Date(VGT$Date)
VHT$Date <- as.Date(VHT$Date)

# Create time series objects
vgt_ts <- ts(VGT$average_movement, start = c(year(min(VGT$Date)), month(min(VGT$Date))),
frequency = 12)
vht_ts <- ts(VHT$average_movement, start = c(year(min(VHT$Date)), month(min(VHT$Date))),
frequency = 12)

# Fit ARIMA models
vgt_arima <- auto.arima(vgt_ts)
vht_arima <- auto.arima(vht_ts)

# Forecast for next 12 months
vgt_forecast <- forecast(vgt_arima, h = 12, level = c(95, 99))
vht_forecast <- forecast(vht_arima, h = 12, level = c(95, 99))

# Plot the forecasts
autoplot(vgt_forecast) + ggtitle("VGT Stock Price Forecast") + xlab("Time") + ylab("Average
Movement")
```

```
autoplot(vht_forecast) + ggtitle("VHT Stock Price Forecast") + xlab("Time") + ylab("Average Movement")

# Print forecast summaries
summary(vgt_forecast)
summary(vht_forecast)
```

VGT analysis

```
# Load necessary libraries
library(ggplot2)
library(readr)
library(dplyr)
library(lubridate)

# Load the CSV file
vgt_data <- read_csv("VGT_stock_data_cleaned.csv")

# Convert Date to Date type
vgt_data$Date <- as.Date(vgt_data$Date)

colnames(vgt_data)
# Split into periods
pre_covid <- filter(vgt_data, Date < as.Date("2020-01-01"))
during_covid <- filter(vgt_data, Date >= as.Date("2020-01-01") & Date <= as.Date("2023-12-31"))
post_covid <- filter(vgt_data, Date > as.Date("2021-12-31"))

# Add period labels
vgt_data <- vgt_data %>%
 mutate(Period = case_when(
       Date < as.Date("2020-01-01") ~ "Pre-COVID (2015–2019)",
       Date <= as.Date("2023-12-31") ~ "During COVID (2020–2023)",
       TRUE ~ "Post-COVID (2024–Now)"
 ))

pre_model <- lm(average_movement ~ Date, data = pre_covid)
during_model <- lm(average_movement ~ Date, data = during_covid)
post_model <- lm(average_movement ~ Date, data = post_covid)

summary(pre_model)
summary(during_model)
summary(post_model)

# Plotting
ggplot(vgt_data, aes(x = Date, y = average_movement, color = Period)) +
 geom_line(size = 1) +
 labs(
       title = "VGT Average Monthly Movement: Pre-, During, and Post-COVID",
       x = "Date",
       y = "Average Movement"
 ) +
 scale_color_manual(values = c(
       "Pre-COVID (2015–2019)" = "blue",
```

```
      "During COVID (2020–2023)" = "orange",
      "Post-COVID (2024–Now)" = "green"
  )) +
  theme_minimal()
```

VHT analysis

```
# Load the CSV file
vht_data <- read_csv("VHT_stock_data_cleaned.csv")

# Convert Date to Date type
vht_data$Date <- as.Date(vht_data$Date)

# Split into periods
pre_covid <- filter(vht_data, Date < as.Date("2020-01-01"))
during_covid <- filter(vht_data, Date >= as.Date("2020-01-01") & Date <= as.Date("2023-12-31"))
post_covid <- filter(vht_data, Date > as.Date("2021-12-31"))

# Add period labels
vht_data <- vht_data %>%
  mutate(Period = case_when(
      Date < as.Date("2020-01-01") ~ "Pre-COVID (2015–2019)",
      Date <= as.Date("2023-12-31") ~ "During COVID (2020–2023)",
      TRUE ~ "Post-COVID (2024–Now)"
  ))

pre_model <- lm(average_movement ~ Date, data = pre_covid)
during_model <- lm(average_movement ~ Date, data = during_covid)
post_model <- lm(average_movement ~ Date, data = post_covid)

summary(pre_model)
summary(during_model)
summary(post_model)

# Plotting
ggplot(vht_data, aes(x = Date, y = average_movement, color = Period)) +
  geom_line(size = 1) +
  labs(
      title = "VHT Average Monthly Movement: Pre-, During, and Post-COVID",
      x = "Date",
      y = "Average Movement"
  ) +
  scale_color_manual(values = c(
      "Pre-COVID (2015–2019)" = "blue",
      "During COVID (2020–2023)" = "orange",
      "Post-COVID (2024–Now)" = "green"
  )) +
  theme_minimal()
```

Unemployment rate in different periods

```r
# Rename and add period
unemp_clean <- unemp_data %>%
  rename(Date = observation_date, unemployment_rate = UNRATE) %>%
  mutate(
    Date = as.Date(Date, format = "%m/%d/%Y"),
    period = case_when(
      Date < as.Date("2020-03-01") ~ "pre_covid",
      Date <= as.Date("2023-11-30") ~ "during_covid",
      Date > as.Date("2023-11-01") ~ "post_covid"
    )
  )
```

Unemployment Rate Plotting

```r
# Unemployment rate time trend
# Finding the max, min, start, end
point_start <- unemp_clean %>% filter(Date == as.Date("2020-03-01"))
point_end <- unemp_clean %>% filter(Date == as.Date("2023-12-01"))
point_highest <- unemp_clean %>% filter(unemployment_rate == max(unemployment_rate))
point_lowest <- unemp_clean %>% filter(unemployment_rate == min(unemployment_rate))

# plotting
ggplot(unemp_clean, aes(x = Date, y = unemployment_rate)) +
  geom_line(color = "skyblue", size = 1) +

  #Start
  geom_text(data = point_start,
            aes(label = paste("Start", format(Date, "%b %Y"),sep = "\n")),
            vjust = -1, hjust = 1.2, color = "purple") +

  #End
  geom_text(data = point_end,
            aes(label = paste("End", format(Date, "%b %Y"), sep = "\n")),
            vjust = -1, color = "purple") +

  #Highest
  geom_point(data = point_highest, aes(x = Date, y = unemployment_rate),
             color = "red", size = 3) + geom_text(data = point_highest,
          aes(label = paste0("Highest: ", unemployment_rate,
                             "% (", format(Date, "%b %Y"), ")")),
                  vjust = -1, color = "red", size = 3.5) +

  #Lowest
  geom_point(data = point_lowest, aes(x = Date, y = unemployment_rate),
             color = "purple", size = 3) +geom_text(data = point_lowest,
  aes(label = paste0("Lowest: ", unemployment_rate,
                     "% (", format(Date, "%b %Y"), ")")), vjust = 1.5,
                      color = "blue", size = 3.5) +

  # Vertical lines to indicate period breakpoints
  geom_vline(xintercept = as.Date("2020-03-01"), linetype = "dashed", color =
"tomato") +
```

```
  geom_vline(xintercept = as.Date("2023-12-01"), linetype = "dashed", color =
"mediumseagreen") +

  labs(
    title = "Unemployment Rate Trend (2015–2025)",
    subtitle = "Dashed lines show the beginning and end of the COVID period",
    x = "Date",
    y = "Unemployment Rate (%)"
  ) +
  theme_minimal()
```

Initial Claims for Unemployment Insurance

3 weeks with the lowest and highest unemployment claims

```
top3 <- head(claims_data[order(-claims_data$ICSA), ], 3)
last3 <- tail(claims_data[order(-claims_data$ICSA), ], 3)

cat(paste0("3 weeks with highest unemployment claims:\n",
           "1. ", top3$Date[1], " – ", top3$ICSA[1], " claims\n",
           "2. ", top3$Date[2], " – ", top3$ICSA[2], " claims\n",
           "3. ", top3$Date[3], " – ", top3$ICSA[3], " claims\n\n",
           "3 weeks with lowest unemployment claims:\n",
           "1. ", last3$Date[1], " – ", last3$ICSA[1], " claims\n",
           "2. ", last3$Date[2], " – ", last3$ICSA[2], " claims\n",
           "3. ", last3$Date[3], " – ", last3$ICSA[3], " claims\n"),
    sep = "")
```

```
## 3 weeks with the highest unemployment claims:
## 1. 2020-04-04 – 6137000 claims
## 2. 2020-03-28 – 5946000 claims
## 3. 2020-04-11 – 4869000 claims
##
## 3 weeks with the lowest unemployment claims:
## 1. 2021-12-25 – 195000 claims
## 2. 2024-01-13 – 194000 claims
## 3. 2022-09-24 – 187000 claims
```

Compare the ICSA and the unemployment rate

```
# Change Date Format
# Monthly Caims data set
monthly_claims <- claims_data %>%
  mutate(Month = format(Date, "%Y-%m")) %>%
  group_by(Month) %>%
  summarise(
    TotalClaims = sum(ICSA, na.rm = TRUE),
    .groups = "drop"
  )

# Add month to unemp_clean
unemp_clean <- unemp_clean %>%
  mutate(Month = format(Date, "%Y-%m"))
```

```r
# Combine unemp_clean and Monthly Claims
unemp_claims_data <- unemp_clean %>%
  select(Month, unemployment_rate, period) %>%
  left_join(monthly_claims, by = "Month")
```

Simple linear regression: unemployment_rate ~ TotalClaims

```r
# Simple linear regression: unemployment_rate ~ TotalClaims
model_claims_unemp <- lm(TotalClaims ~ unemployment_rate, data = unemp_claims_data)
summary(model_claims_unemp)
```

COVID-19 case and death analysis

COVID-19 cases and deaths base on Age

```r
# add a column in covid data containing the age group
covid_data <- covid_data %>%
  mutate(
    AgeGroupNum = case_when(
      AgeGroup == "0 - 4 Years"   ~ 1,
      AgeGroup == "5 - 11 Years"  ~ 2,
      AgeGroup == "12 - 15 Years" ~ 3,
      AgeGroup == "16 - 17 Years" ~ 4,
      AgeGroup == "18 - 29 Years" ~ 5,
      AgeGroup == "30 - 39 Years" ~ 6,
      AgeGroup == "40 - 49 Years" ~ 7,
      AgeGroup == "50 - 64 Years" ~ 8,
      AgeGroup == "65 - 74 Years" ~ 9,
      AgeGroup == "75+ Years"     ~ 10,
      AgeGroup == "Overall"       ~ 11
    )
  )

# age summary table (excluding Overall)
age_summary <- covid_data %>%
  filter(AgeGroup != "Overall") %>%
  group_by(AgeGroup, AgeGroupNum) %>%
  summarise(
    TotalCases = sum(CaseSuppressed, na.rm = TRUE),
    TotalDeaths = sum(DeathSuppressed, na.rm = TRUE),
    MeanCaseRate = mean(CaseRate, na.rm = TRUE),
    MeanDeathRate = mean(DeathRate, na.rm = TRUE),
    MaxDeathRate = max(DeathRate, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(AgeGroupNum)

# Total Case
# Highest / Lowest of cases
highest_cases_age <- age_summary[which.max(age_summary$TotalCases), ]

lowest_cases_age <- age_summary[which.min(age_summary$TotalCases), ]

# Mean Case Rate
highest_mean_case_age <- age_summary[which.max(age_summary$MeanCaseRate), ]
```

```r
lowest_mean_case_age <- age_summary[which.min(age_summary$MeanCaseRate), ]

# Total Death
# Highest /Lowest of death
most_deaths_age <- age_summary[which.max(age_summary$TotalDeaths), ]

least_deaths_age <- age_summary[which.min(age_summary$TotalDeaths), ]


# Mean Death Rate
# Highest / Lowest Mean Death Rate
highest_mean_deathrate_age <- age_summary[which.max(age_summary$MeanDeathRate), ]

lowest_mean_deathrate_age <- age_summary[which.min(age_summary$MeanDeathRate), ]

# Max / Min Death Rate
highest_max_deathrate_age <- age_summary[which.max(age_summary$MaxDeathRate), ]

lowest_max_deathrate_age <- age_summary[which.min(age_summary$MaxDeathRate), ]

# Print all
cat(
  " Age Group with the MOST cases:", highest_cases_age$AgeGroup,
  "(", highest_cases_age$TotalCases, "cases)\n",

  "Age Group with the FEWEST cases:", lowest_cases_age$AgeGroup,
  "(", lowest_cases_age$TotalCases, "cases)\n\n",

  "Age Group with the HIGHEST average case rate:", highest_mean_case_age$AgeGroup,
  "(", highest_mean_case_age$MeanCaseRate, "per 100k)\n",

  "Age Group with the LOWEST average case rate:", lowest_mean_case_age$AgeGroup,
  "(", lowest_mean_case_age$MeanCaseRate, "per 100k)\n\n",

  "Age Group with the MOST deaths:", most_deaths_age$AgeGroup,
  "(", most_deaths_age$TotalDeaths, "deaths)\n",

  "Age Group with the FEWEST deaths:", least_deaths_age$AgeGroup,
  "(", least_deaths_age$TotalDeaths, "deaths)\n\n",

  "Age Group with the HIGHEST average death rate:",
highest_mean_deathrate_age$AgeGroup,
  "(", highest_mean_deathrate_age$MeanDeathRate, "per 100k)\n",

  "Age Group with the LOWEST average death rate:", lowest_mean_deathrate_age$AgeGroup,
  "(", lowest_mean_deathrate_age$MeanDeathRate, "per 100k)\n\n",

  "Age Group with the HIGHEST death rate:", highest_max_deathrate_age$AgeGroup,
  "(", highest_max_deathrate_age$MaxDeathRate, "per 100k)\n",

  "Age Group with the LOWEST death rate:", lowest_max_deathrate_age$AgeGroup,
```

```
  "(", lowest_max_deathrate_age$MaxDeathRate, "per 100k)\n"
)
```

```
##  Age Group with the MOST cases: 18 - 29 Years ( 131665357 cases)
##  Age Group with the FEWEST cases: 16 - 17 Years ( 16895805 cases)
##
##  Age Group with the HIGHEST average case rate: 30 - 39 Years ( 138.9509 per 100k)
##  Age Group with the LOWEST average case rate: 0 - 4 Years ( 73.61458 per 100k)
##
##  Age Group with the MOST deaths: 75+ Years ( 3944452 deaths)
##  Age Group with the FEWEST deaths: 16 - 17 Years ( 248 deaths)
##
##  Age Group with the HIGHEST average death rate: 75+ Years ( 9.078197 per 100k)
##  Age Group with the LOWEST average death rate: 16 - 17 Years ( 0.0002006679 per
100k)
##
##  Age Group with the HIGHEST death rate: 75+ Years ( 660.68 per 100k)
##  Age Group with the LOWEST death rate: 5 - 11 Years ( 0.59 per 100k)
```

Age ~ Death Rate: Simple linear regression

```
# simple linear regression age ~ DeathRate
model_deathdata_age <- covid_data %>%
  filter(AgeGroup != "Overall") %>%
  select(AgeGroup, DeathRate) %>%
  mutate(AgeGroup = as.factor(AgeGroup))

model_death_age <- lm(DeathRate ~ AgeGroup, data = model_deathdata_age)
summary(model_death_age)
```

Age ~ Case Rate Simple linear regression

```
# simple linear regression age ~ CaseRate
model_casedata_age <- covid_data %>%
  filter(AgeGroup != "Overall") %>%
  select(AgeGroup, CaseRate) %>%
  mutate(AgeGroup = as.factor(AgeGroup))

model_case_age <- lm(CaseRate ~ AgeGroup, data = model_casedata_age)
summary(model_case_age)
```

COVID-19 cases and deaths based on Sex

```
# sex summary table (excluding Overall)
sex_summary <- covid_data %>%
  filter(Sex != "Overall") %>%
  group_by(Sex) %>%
  summarise(
    TotalCases = sum(CaseSuppressed, na.rm = TRUE),
    TotalDeaths = sum(DeathSuppressed, na.rm = TRUE),
```

```
    MeanCaseRate = mean(CaseRate, na.rm = TRUE),
    MeanDeathRate = mean(DeathRate, na.rm = TRUE),
    MaxDeathRate = max(DeathRate, na.rm = TRUE),
    .groups = "drop"
  )

# Total Case
# Highest of cases
highest_cases_sex <- sex_summary[which.max(sex_summary$TotalCases), ]

# Lowest of cases
lowest_cases_sex <- sex_summary[which.min(sex_summary$TotalCases), ]


# Total Death
# Highest of death
most_deaths_sex <- sex_summary[which.max(sex_summary$TotalDeaths), ]

#Lowest of death
least_deaths_sex <- sex_summary[which.min(sex_summary$TotalDeaths), ]

# Mean Death Rate
# Highest Mean Death Rate
highest_mean_deathrate_sex <- sex_summary[which.max(sex_summary$MeanDeathRate), ]

# Lowest Mean Death Rate
lowest_mean_deathrate_sex <- sex_summary[which.min(sex_summary$MeanDeathRate), ]

# Max Death Rate
highest_max_deathrate_sex <- sex_summary[which.max(sex_summary$MaxDeathRate), ]

# Min Death Rate
lowest_max_deathrate_sex <- sex_summary[which.min(sex_summary$MaxDeathRate), ]


# Total Case
cat(
  "  Sex group with the MOST cases: ", highest_cases_sex$Sex,
  "(", highest_cases_sex$TotalCases, " cases)\n ",

  "Sex group with the FEWEST cases: ", lowest_cases_sex$Sex,
  "(", lowest_cases_sex$TotalCases, " cases)\n\n ",

  # Total Death
  "Sex group with the MOST deaths: ", most_deaths_sex$Sex,
  "(", most_deaths_sex$TotalDeaths, " deaths)\n ",

  "Sex group with the FEWEST deaths: ", least_deaths_sex$Sex,
  "(", least_deaths_sex$TotalDeaths, " deaths)\n\n ",

  # Mean Death Rate
  "Sex group with the HIGHEST average death rate: ", highest_mean_deathrate_sex$Sex,
  "(", highest_mean_deathrate_sex$MeanDeathRate, " per 100k)\n ",
```

```
  "Sex group with the LOWEST average death rate: ", lowest_mean_deathrate_sex$Sex,
  "(", lowest_mean_deathrate_sex$MeanDeathRate, " per 100k)\n\n ",

  # Max Death Rate
  "Sex group with the HIGHEST death rate: ", highest_max_deathrate_sex$Sex,
  "(", highest_max_deathrate_sex$MaxDeathRate, " per 100k)\n ",

  "Sex group with the LOWEST death rate: ", lowest_max_deathrate_sex$Sex,
  "(", lowest_max_deathrate_sex$MaxDeathRate, " per 100k)"
)
```

```
##   Sex group with the MOST cases:  Female ( 360931148  cases)
##   Sex group with the FEWEST cases:  Male ( 302206161  cases)
##
##   Sex group with the MOST deaths:  Male ( 3970788  deaths)
##   Sex group with the FEWEST deaths:  Female ( 3282344  deaths)
##
##   Sex group with the HIGHEST average death rate:  Male ( 1.541264  per 100k)
##   Sex group with the LOWEST average death rate:  Female ( 0.9983938  per 100k)
##
##   Sex group with the HIGHEST death rate:  Male ( 660.68  per 100k)
##   Sex group with the LOWEST death rate:  Female ( 369.58  per 100k)
```

Sex ~ Death Rate T-test & Mann-Whitney U test

```
ttest_deathdata_sex <- covid_data %>%
  filter(Sex != "Overall") %>%
  select(Sex, DeathRate) %>%
  mutate(Sex = as.factor(Sex))

# print Hypothese
cat("Hypotheses for T-test and Mann-Whitney U test:\n",
"H0: There is no difference in death rate between males and females.\n",
"H1: There is a difference in death rate between males and females.\n")
```

```
## Hypotheses for T-test and Mann-Whitney U test:
##  H0: There is no difference in death rate between males and females.
##  H1: There is a difference in death rate between males and females.
```

```
# T- Test
t.test(DeathRate ~ Sex, data = ttest_deathdata_sex)

# Mann-Whitney U test
wilcox.test(DeathRate ~ Sex, data = ttest_deathdata_sex)
```

Sex ~ Case Rate Simple linear regression

```
## simple linear regression Sex ~ CaseRate
model_casedata_sex <- covid_data %>%
  filter(Sex != "Overall") %>%
  select(Sex, CaseRate) %>%
```

```r
  mutate(Sex = as.factor(Sex))

model_case_sex <- lm(CaseRate ~ Sex, data = model_casedata_sex)
summary(model_case_sex)
```

Sex ~ Case Rate T-test & Mann-Whitney U test

```r
ttest_casedata_sex <- covid_data %>%
  filter(Sex != "Overall") %>%
  select(Sex, CaseRate) %>%
  mutate(Sex = as.factor(Sex))

# Step 2: Print the hypotheses
cat("Hypotheses for T-test and Mann-Whitney U test:\n",
"H0: There is no difference in COVID-19 case rate between males and females.\n",
"H1: There is a difference in COVID-19 case rate between males and females.\n\n")
```

```
## Hypotheses for T-test and Mann-Whitney U test:
##  H0: There is no difference in COVID-19 case rate between males and females.
##  H1: There is a difference in COVID-19 case rate between males and females.
```

```r
# T-Test
t.test(CaseRate ~ Sex, data = ttest_casedata_sex)

# Mann-Whitney U test
wilcox.test(CaseRate ~ Sex, data = ttest_casedata_sex)
```

Age + Sex ~ Death Rate: Multiple linear regression

```r
# multiple linear regression Age + Sex ~ DeathRate
model_data_agesex <- covid_data %>%
  filter(Sex != "Overall", AgeGroup != "Overall") %>%
  select(AgeGroup, Sex, DeathRate) %>%
  mutate(
    AgeGroup = as.factor(AgeGroup),
    Sex = as.factor(Sex)
  )

# Multiple linear regression model: Age + Sex ~ DeathRate
model_age_sex <- lm(DeathRate ~ AgeGroup + Sex, data = model_data_agesex)
summary(model_age_sex)
```

COVID-19 cases and deaths based on Race

```r
# Race Summary table (excluding Overall)
race_summary <- covid_data %>%
  filter(Race != "Overall") %>%
  group_by(Race) %>%
  summarise(
    TotalCases = sum(CaseSuppressed, na.rm = TRUE),
    TotalDeaths = sum(DeathSuppressed, na.rm = TRUE),
    MeanCaseRate = mean(CaseRate, na.rm = TRUE),
```

```
    MeanDeathRate = mean(DeathRate, na.rm = TRUE),
    MaxDeathRate = max(DeathRate, na.rm = TRUE),
    .groups = "drop"
  )

# Total Case
# Highest of cases
highest_cases_race <- race_summary[which.max(race_summary$TotalCases), ]

# Lowest of cases
lowest_cases_race <- race_summary[which.min(race_summary$TotalCases), ]

# Mean Case Rate
highest_mean_case_race <- race_summary[which.max(race_summary$MeanCaseRate), ]

lowest_mean_case_race <- race_summary[which.min(race_summary$MeanCaseRate), ]


# Total Death
# Highest of deaths
most_deaths_race <- race_summary[which.max(race_summary$TotalDeaths), ]

# Lowest of deaths
least_deaths_race <- race_summary[which.min(race_summary$TotalDeaths), ]


# Highest Mean Death Rate
highest_mean_deathrate_race <- race_summary[which.max(race_summary$MeanDeathRate), ]

# Lowest Mean Death Rate
lowest_mean_deathrate_race <- race_summary[which.min(race_summary$MeanDeathRate), ]

# Highest Max Death Rate
highest_max_deathrate_race <- race_summary[which.max(race_summary$MaxDeathRate), ]

# Lowest Max Death Rate
lowest_max_deathrate_race <- race_summary[which.min(race_summary$MaxDeathRate), ]

# Total Case
cat(
  " Race group with the MOST cases:", highest_cases_race$Race,
  "(", highest_cases_race$TotalCases, " cases)\n",

  "Race group with the FEWEST cases:", lowest_cases_race$Race,
  "(", lowest_cases_race$TotalCases, " cases)\n\n",

  # Mean Case Rate
  "Race group with the HIGHEST average case rate:", highest_mean_case_race$Race,
  "(", highest_mean_case_race$MeanCaseRate, " per 100k)\n",

  "Race group with the LOWEST average case rate:", lowest_mean_case_race$Race,
  "(", lowest_mean_case_race$MeanCaseRate, " per 100k)\n\n",
```

```
  # Total Death
  "Race group with the MOST deaths:", most_deaths_race$Race,
  "(", most_deaths_race$TotalDeaths, " deaths)\n",

  "Race group with the FEWEST deaths:", least_deaths_race$Race,
  "(", least_deaths_race$TotalDeaths, " deaths)\n\n",

  # Mean Death Rate
  "Race group with the HIGHEST average death rate:", highest_mean_deathrate_race$Race,
  "(", highest_mean_deathrate_race$MeanDeathRate, " per 100k)\n",

  "Race group with the LOWEST average death rate:", lowest_mean_deathrate_race$Race,
  "(", lowest_mean_deathrate_race$MeanDeathRate, " per 100k)\n\n",

  # Max Death Rate
  "Race group with the HIGHEST death rate:", highest_max_deathrate_race$Race,
  "(", highest_max_deathrate_race$MaxDeathRate, " per 100k)\n",

  "Race group with the LOWEST death rate:", lowest_max_deathrate_race$Race,
  "(", lowest_max_deathrate_race$MaxDeathRate, " per 100k)\n"
)
```

```
##  Race group with the MOST cases: White, NH ( 293941659  cases)
##  Race group with the FEWEST cases: AI/AN, NH ( 5333095  cases)
##
##  Race group with the HIGHEST average case rate: Hispanic ( 136.5546  per 100k)
##  Race group with the LOWEST average case rate: Asian/PI, NH ( 75.52885  per 100k)
##
##  Race group with the MOST deaths: White, NH ( 4369003  deaths)
##  Race group with the FEWEST deaths: AI/AN, NH ( 53139  deaths)
##
##  Race group with the HIGHEST average death rate: Hispanic ( 1.814881  per 100k)
##  Race group with the LOWEST average death rate: Asian/PI, NH ( 0.5773722  per 100k)
##
##  Race group with the HIGHEST death rate: Hispanic ( 660.68  per 100k)
##  Race group with the LOWEST death rate: White, NH ( 185.8  per 100k)
```

Death Rate ~ Race Simple linear regression

```
# simple linear regression DeathRate ~ Race
model_data_race <- covid_data %>%
  filter(Race != "Overall") %>%
  select(Race, DeathRate) %>%
  mutate(Race = as.factor(Race))

model_race_simple <- lm(DeathRate ~ Race, data = model_data_race)
summary(model_race_simple)
```

Case Rate ~ Race Simple linear regression

```
## simple linear regression CaseRate ~ Race
model_data_race <- covid_data %>%
```

```
  filter(Race != "Overall") %>%
  select(Race, CaseRate) %>%
  mutate(Race = as.factor(Race))

model_case_race <- lm(CaseRate ~ Race, data = model_data_race)
summary(model_case_race)
```

Age + Sex + Race ~ Death Rate: Multiple linear regression

```
#Multiple linear regression Age + Sex + Race ~ DeathRate
model_deathdata_asr <- covid_data %>%
  filter(AgeGroup != "Overall", Sex != "Overall", Race != "Overall") %>%
  select(AgeGroup, Sex, Race, DeathRate) %>%
  mutate(
    AgeGroup = as.factor(AgeGroup),
    Sex = as.factor(Sex),
    Race = as.factor(Race)
  )

# Multiple linear regression model: Age + Sex + Race ~ DeathRate
model_death_asr <- lm(DeathRate ~ AgeGroup + Sex + Race, data = model_deathdata_asr)
summary(model_death_asr)
```

Age + Sex + Race ~ Case Rate Multiple linear regression

```
# Multiple linear regression: Age + Sex + Race ~ CaseRate
model_casedata_asr <- covid_data %>%
  filter(AgeGroup != "Overall", Sex != "Overall", Race != "Overall") %>%
  select(AgeGroup, Sex, Race, CaseRate) %>%
  mutate(
    AgeGroup = as.factor(AgeGroup),
    Sex = as.factor(Sex),
    Race = as.factor(Race)
  )

model_case_asr <- lm(CaseRate ~ AgeGroup + Sex + Race, data = model_casedata_asr)
summary(model_case_asr)
```

State-Level Analysis

```
# summary tables: total cases, total deaths per state
state_summary <- state_case_death %>%
  group_by(State) %>%
  summarise(
    TotalCases = sum(NewCases, na.rm = TRUE),
    TotalDeaths = sum(NewDeaths, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    Death_Percentage = TotalDeaths / TotalCases * 100  #Calculate Death Percentage
```

```
  )

# Highest total cases
most_cases_state <- state_summary[which.max(state_summary$TotalCases), ]

# Lowest total cases
least_cases_state <- state_summary[which.min(state_summary$TotalCases), ]

# Highest total deaths
most_deaths_state <- state_summary[which.max(state_summary$TotalDeaths), ]

# Lowest total deaths
least_deaths_state <- state_summary[which.min(state_summary$TotalDeaths), ]

# Highest death percentage
highest_deathrate_state <- state_summary[which.max(state_summary$Death_Percentage), ]

# Lowest death percentage
lowest_deathrate_state <- state_summary[which.min(state_summary$Death_Percentage), ]

# print result
cat(
  "State with the MOST total cases:", most_cases_state$State,
  " (", most_cases_state$TotalCases, " cases)\n",
  "State with the FEWEST total cases:", least_cases_state$State,
  " (", least_cases_state$TotalCases, " cases)\n\n",
  "State with the MOST total deaths:", most_deaths_state$State,
  " (", most_deaths_state$TotalDeaths, " deaths)\n",
  "State with the FEWEST total deaths:", least_deaths_state$State,
  " (", least_deaths_state$TotalDeaths, " deaths)\n\n",
  "State with the HIGHEST death rate:", highest_deathrate_state$State,
  " (", round(highest_deathrate_state$Death_Percentage, 2), " %)\n",
  "State with the LOWEST death rate:", lowest_deathrate_state$State,
  " (", round(lowest_deathrate_state$Death_Percentage, 2), " %)\n",
  sep = ""
)
```

```
## State with the MOST total cases:CA (12251820 cases)
## State with the FEWEST total cases:PW (6000 cases)
##
## State with the MOST total deaths:CA (101886 deaths)
## State with the FEWEST total deaths:PW (9 deaths)
##
## State with the HIGHEST death rate:PA (1.43 %)
## State with the LOWEST death rate:RMI (0.11 %)
```

State Simple linear regression

```
model_simple_state <- lm(TotalDeaths ~ TotalCases, data = state_summary)
summary(model_simple_state)
```

Project Contribution

Data Cleaning: Kush Manek

He did an amazing job cleaning the data, ensuring it was accurate and well-structured for analysis. His attention to detail helped eliminate errors and missing values, making the datasets reliable. His teamwork made the data preparation smooth and efficient for the entire group.

Visualization: Sarah Sandmeier

She created clear and insightful visualizations that brought our data to life. Her charts and graphs made complex trends easy to understand, helping connect the analysis with the report and presentation. Her teamwork ensured the visuals matched perfectly with our findings. She promptly made adjustments and modifications as suggested by the team.

Statistical Analysis: Xueying Lin

She played a key role in analyzing our data, applying statistical methods to uncover meaningful insights. She worked closely with the team to ensure accuracy and clarity, explaining complex concepts in a way everyone could understand. Her contributions strengthened our final results. She promptly made adjustments and modifications as suggested by the team.

Report Writing: Thamy Soares & Kush Manek

They worked together to write a clear and well-organized report. They made sure the findings, analysis, and visuals were easy to follow, creating a polished final document. Their teamwork ensured all important details were included.

Debugging & Troubleshooting: Thamy Soares and Kush Manek

They tackled technical issues, reviewing every part in detail: suggesting alterations and fixing errors in data processing, analysis, and visualizations. Their problem-solving skills helped keep the project running smoothly and ensured accurate results.

Slide Preparation: Thamy Soares

She took the lead in preparing a visually engaging and well-organized presentation that effectively demonstrated our findings. She combined data, visuals, and explanations into clear slides showing COVID-19's impact on the US economy. Her ability to structure information logically and present complex data in a concise format significantly contributed to the success of our final presentation. Thamy's careful design made the slides clear and easy to follow during the presentation.