R Notebook

Exploratory Analysis of the US Opioid Epidemic: Overdoses, Treatment Providers, and Prescribing Rates

Introduction

Opioid addiction is an epidemic in the U.S. with 3 million citizens affected by opioid use disorder. Opioids are often overprescribed in the U.S., leading to an increase in addiction rates 1. Certain providers in the US have enrolled in Medicare under the Opioid Treatment Program, and provide services to help treat opioid use disorder 2.

Data Sources

To learn more about the the opioid epidemic in the U.S., I used three main datasets:

- Medicaid Opioid Prescribing Rates (3)
- Opioid Treatment Program Providers (4)
- VSRR Provisional Drug Overdose Death Counts (5)

I also used the tidycensus library to obtain the population for each state in the 2020 census (6).

Guiding Research Questions:

How do opioid prescribing rates, opioid overdose deaths, and availability of treatment providers vary by state?

Are states with a higher percent increase in opioid prescribing rates during a certain time frame more likely to have an increase in death rate in the last five years?

Do states with a larger opioid problem have more treatment providers or vice-versa?

To complete the analysis, I used the tidycensus (6), tidyverse (7), gt (8), gtsummary (9), usmap (10), and lubridate (11) packages.

```r
#load packages
library(tidycensus)
library(tidyverse)
library(gt)
library(gtsummary)
library(usmap)
library(lubridate)
```

```r
#load datasets
prescribing_rates <- read_csv("Medicaid_Opioid_Prescribing_Rates - Sheet1.csv")
overdose_rates <- read_csv("VSRR_Provisional_Drug_Overdose_Death_Counts.csv")
providers <- read_csv("Opioid_Treatment_Program_Providers - Sheet1.csv")
```

I used 2020 census data so that I could adjust any counts by state population.

```r
#Census data 2020, population by state
total_population_2020 <- get_decennial(
  geography = "state",
  variables = "P1_001N",
  year = 2020
)
```

I created a data frame with state names and abbreviation using the built in vectors, manually adding in DC. I used an inner_join to merge the data frame containing state abbreviations and names with the population data. I used an inner join since I only wanted to include states that are in my state abbreviations data frame and also in the population data frame. I also used an inner join to merge the prescribing rates data frame with the state abbreviations data frame, since I only wanted to include states that had data in both data frames. Since every state in the state abbreviations data frame has a value in the prescribing rates data frame, a right join of the prescribing rates with the state abbreviations would have accomplished the same thing as the inner join.

```
#create df with state abbreviations and names
state_abb <- data.frame(state.abb = state.abb, state.name = state.name)

#add DC
state_abb[nrow(state_abb) + 1,] <- list("DC", "District of Columbia")

#add state abbreviations to population df
total_population_2020 <- inner_join(state_abb, total_population_2020, by =
                                      c("state.name" = "NAME"))

#add state abbreviations to prescribing rates df
prescribing_rates <- inner_join(prescribing_rates, state_abb, by= c("Geo_Desc" =
                                "state.name"))
```

To begin, I wanted to visualize the number of opioid addiction treatment providers per 100,000 people by state. I grouped the providers data frame by state and summarized the number of providers for each state. I then used a right join to merge the providers per state data frame with the population data frame, since I wanted there to be a provider count for each state in the population data frame. If I had used a left join or inner join, then Wyoming would not be included in the final data frame since it did not have any providers.

Now that each state had a population count, I grouped the dataframe by state and summarized, creating a column that listed the number of providers per 100k people. To do this, I divided the number of providers by the population for each state and multiplied by 100000. Finally, since Wyoming did not have any providers, but was listed as NA, I used replace() to change NA to 0.

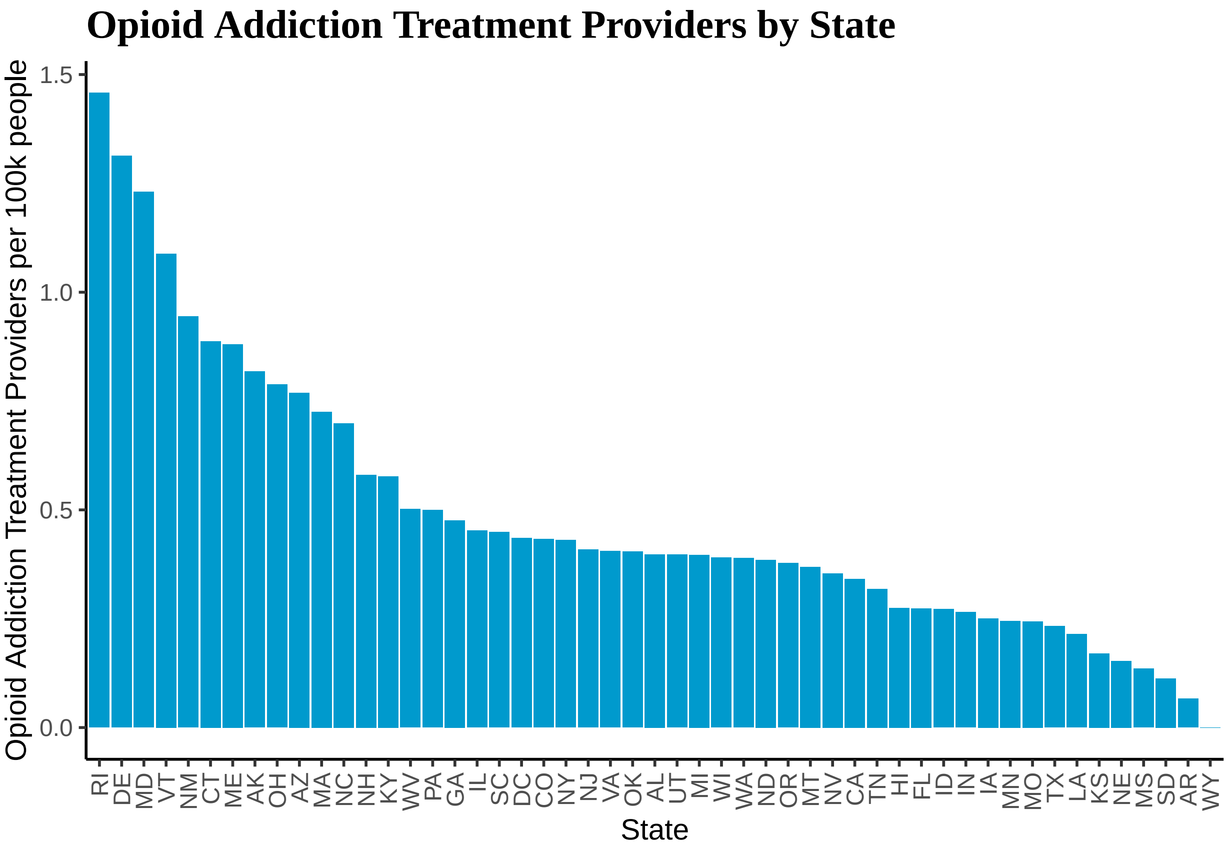
```
#looking at the number of opioid addiction treatment providers by state
num_providers_state <- providers %>% group_by(STATE) %>%
  summarize(total_providers = n())

#merging the total_population_2020 and num_providers_state dfs using a left join
#create a column with the providers per 10000 people for each state
providers_pop <- right_join(num_providers_state, total_population_2020,
                            by = c("STATE" = "state.abb")) %>%
  group_by(STATE) %>%
  summarize(providers_per_100000 = 100000*total_providers/value) %>%
  replace(is.na(.), 0)
```

I created a column chart to display how the number of treatment providers per 100,000 people varied by state. I chose to order the x axis by the number of providers, in order to easily visualize the pattern and see which states have many providers or few providers.

I decided to look at this data in a mapped format as well. This makes it easier for someone to quickly find a certain state or to compare regions of the US. From these figures we see that Rhode Island, Delaware, Maryland, and New Mexico all have a high number of opioid addiction treatment providers per 100k people. Wyoming, Arkansas, and South Dakota have the lowest number of opioid addiction treatment providers per

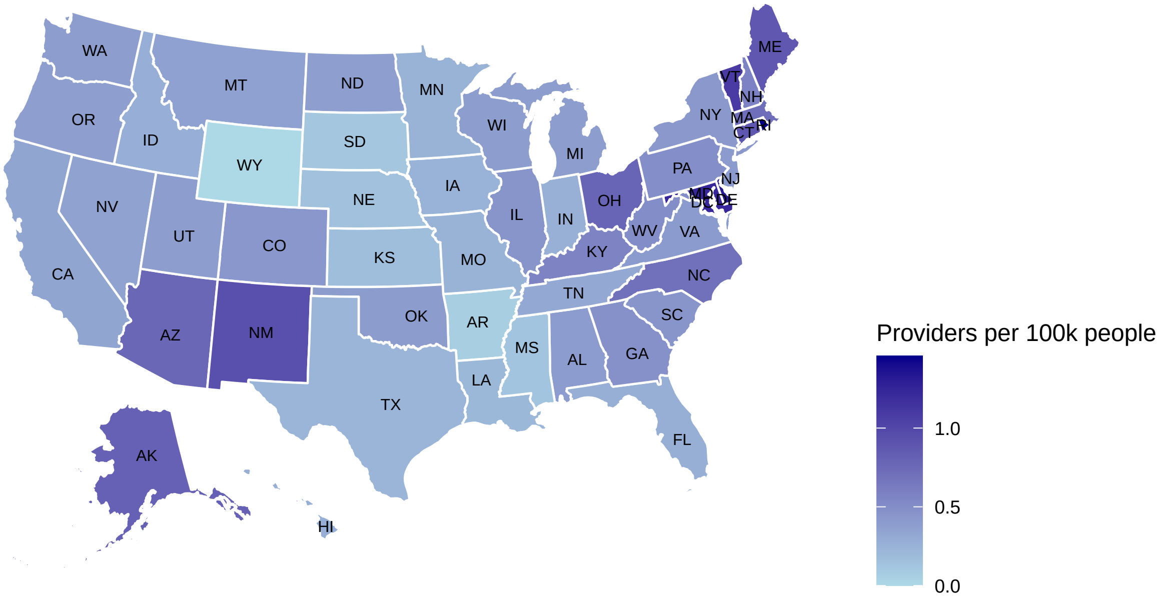
100k people.

```
#plotting providers per 100000 people
ggplot(providers_pop) +
  geom_col(aes(x= reorder(STATE, -providers_per_100000), y=providers_per_100000),
           fill= "deepskyblue3") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(x="State",y="Opioid Addiction Treatment Providers per 100k people"
       ,title="Opioid Addiction Treatment Providers by State") +
  theme(plot.title=element_text(family="Times", face="bold", size=15))
```



```
#change variable name so that it works with the US Map
names(providers_pop)[1] = "state"

#plot providers per 100k on a US map
map <- plot_usmap(data = providers_pop, values = "providers_per_100000",
           color = "white", labels = TRUE) +
  scale_fill_gradient(name = "Providers per 100k people", low = "lightblue",
                      high = "darkblue") +
  theme(legend.position = "right") +
  labs(title = "Opioid Treatment Providers per 100k People") +
  theme(plot.title=element_text(family="Times", face="bold", size=15))

map$layers[[2]]$aes_params$size <- 2
print(map)
```

Opioid Treatment Providers per 100k People



The next visualization that I'm interested in is the rate of opioid overdose deaths by state over time.

The Provisional Drug Overdose Death Counts dataset includes count data for several different cause of death codes. To decide which cause of death I would like to use for my analyses, I created a table of the different cause of death categories and the number of observations present for each category.

```
#look at different values on the Indicator variable of overdose_rates
table(overdose_rates$Indicator)
```

```
##
##                                                             Cocaine (T40.5)
##                                                                        4949
##                                                              Heroin (T40.1)
##                                                                        4949
##                                                           Methadone (T40.3)
##                                                                        4949
##                                    Natural & semi-synthetic opioids (T40.2)
##                                                                        4949
##            Natural & semi-synthetic opioids, incl. methadone (T40.2, T40.3)
##                                                                        4949
## Natural, semi-synthetic, & synthetic opioids, incl. methadone (T40.2-T40.4)
##                                                                        4949
##                                                            Number of Deaths
##                                                                        5353
##                                              Number of Drug Overdose Deaths
##                                                                        5353
##                                                 Opioids (T40.0-T40.4,T40.6)
##                                                                        4949
##                                                Percent with drugs specified
##                                                                        5353
##                               Psychostimulants with abuse potential (T43.6)
##                                                                        4949
##                                  Synthetic opioids, excl. methadone (T40.4)
##                                                                        4949
```

I will only use counts for the cause of death, “Opioids (T40.0-T40.4,T40.6)”. It is important to note that not all states have values for this cause of death for each month/year. Additionally, reporting of specific drugs and drug classes is variable by jurisdiction, and the data provider recommends that this data should not be used to compare death rates involving specific drugs across jurisdictions (5). Since the Opioid category is rather large, is not a specific drug, and this is just a preliminary investigation, I will be comparing death rates among states based off the Opioids category.

I filtered my data to only include the death count data for the opioids category. To make the data set easier to read, I selected only the Indicator, Month, State, Year, and Date Value columns.

In order to observe the death rate over time, I created a column that combines month and year into a “Date” column using the lubridate package.

The column “Data Values” was hard to use with the tidyverse functions since it has a space in the name, so I renamed it to “opioid_overdoses”.

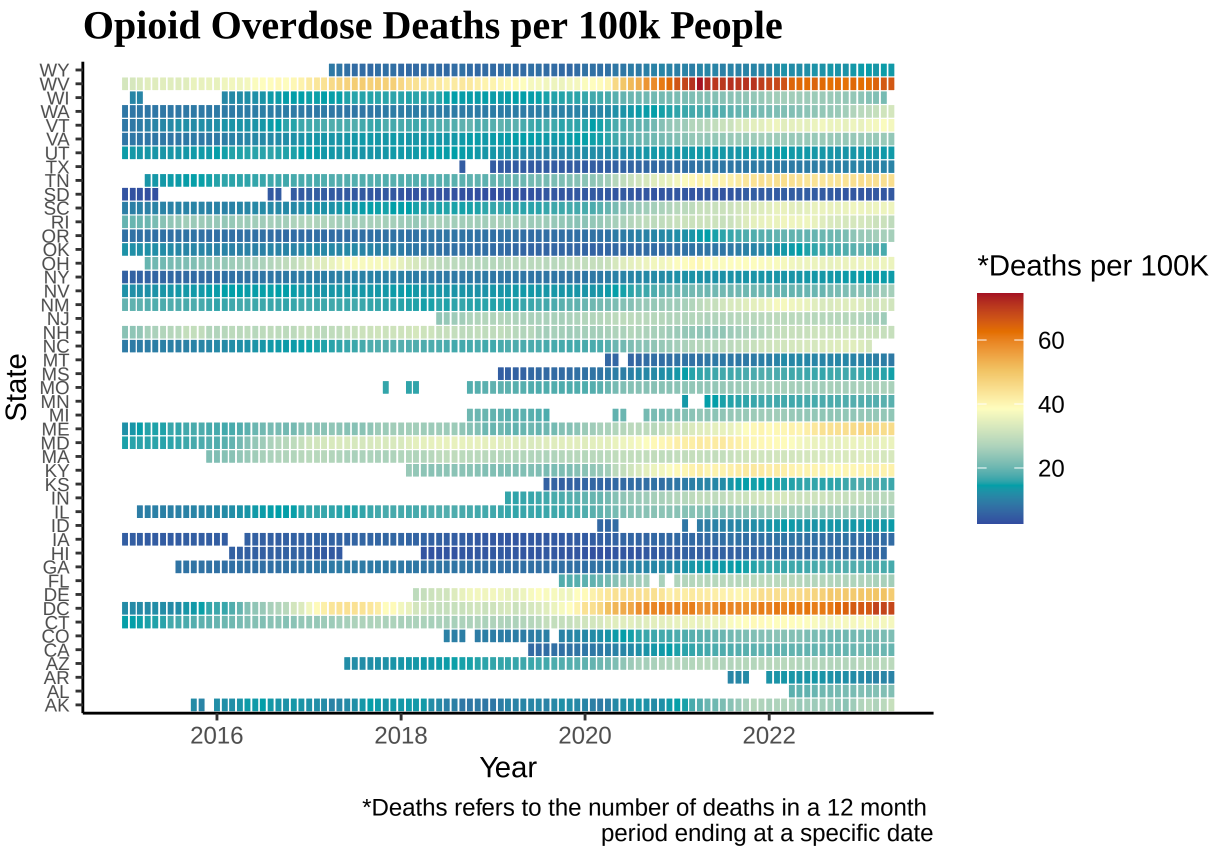
Next, I removed any NAs, filtered out the National data, grouped by State and Date, and then summarized the total number of opioid deaths per state per date. I initially created a visualization with this data as well, but determined it wasn’t very useful compared to the population adjusted data. I used an inner join to merge the opioid overdoses dataset to the population data, since I only wanted to include states that had a count and a population. If I had used a left join, the state with the abbreviation “YC” would have been included, but there would not be a population count for it. I then grouped by State and Date and summarized the number of deaths per 100k people by dividing the state’s overdoses by its population and multiplying by 100000.

```
#filter for only opioids and select only columns im interested in
opioid_overdoses <- overdose_rates %>% filter(Indicator ==
                                              "Opioids (T40.0-T40.4,T40.6)") %>%
  select(Indicator, Month, State, Year, 'Data Value')

#add a column for date that combines the year and month columns
opioid_overdoses$Date <- myd(paste(opioid_overdoses$Month, opioid_overdoses$Year,
                                   "1"))

#rename 'Data Values' to Opioid_overdoses
names(opioid_overdoses)[5] = "Opioid_overdoses"

#remove national data and find opioid overdoses by state and remove NAs
#group by state and date
opioid_overdose_rates_per_100k <- opioid_overdoses %>%
  na.omit(opioid_overdoses) %>%
  filter(State != "US") %>%
  group_by(State, Date) %>%
  summarize(Opioid_overdoses = sum(Opioid_overdoses)) %>%
  inner_join(total_population_2020, by = c("State" = "state.abb")) %>%
  group_by(State, Date) %>%
  summarize(deaths_per_100000 = 100000*Opioid_overdoses/value)
```

I then made a plot using geom_tile with year on the x axis, state on the y axis, and deaths per 100k as the fill. I used the “RdYlBu” palette for the fill, which does a good job differentiating the high death rates from the low death rates, but it’s somewhat difficult to see the more subtle variations. Another possible visualization would be to create a line plot with a line for each state, with the x-axis as date and y-axis as death rate, faceted by region of the US. This would make it harder to compare death rates between states in different regions, but would do a better job visualizing the changes over time.

From the below plot, we see that West Virginia and DC are the states that were the most affected by the opioid epidemic, with especially high death counts in the last 3 years. The white space represents states that did not supply data for those dates.

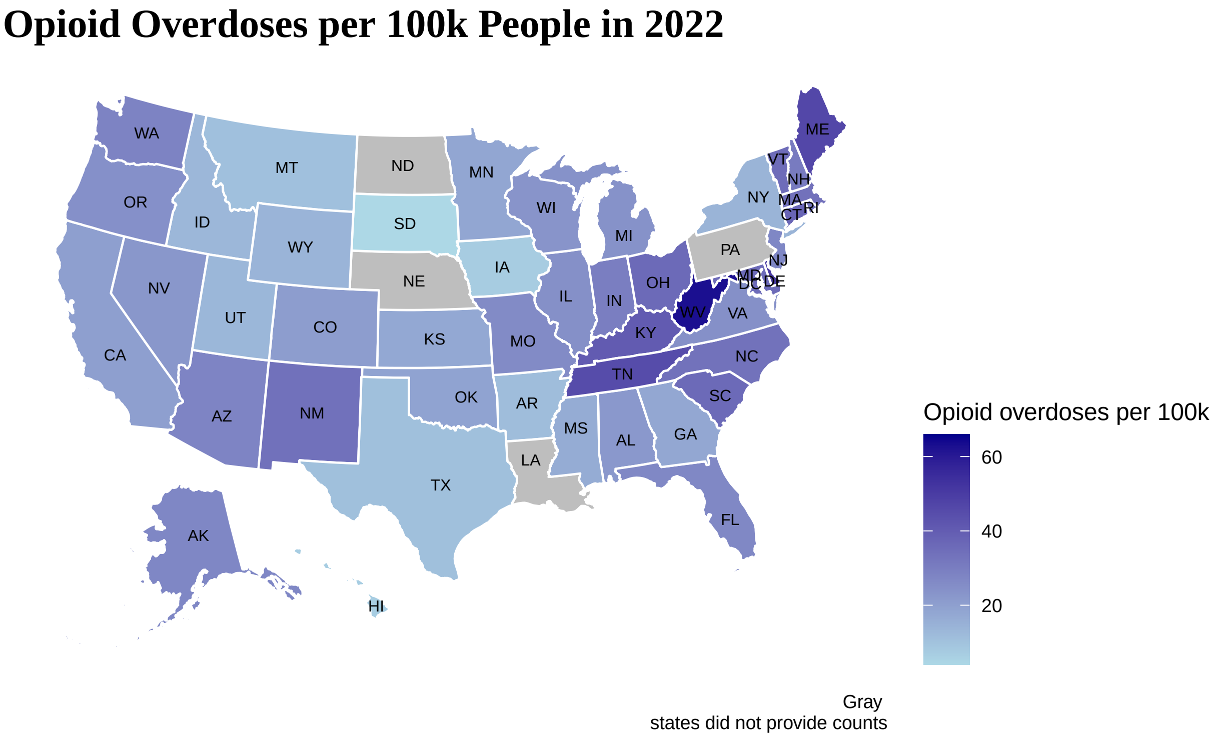
```
#plot opioid overdoses per 100k over time by state
ggplot(opioid_overdose_rates_per_100k) +
  geom_tile(color="white" , size = 0.2 ,aes(x= Date, y= State,
                                            fill = deaths_per_100000)) +
  theme_classic() +
  scale_fill_gradientn(na.value = "white", name = "*Deaths per 100K",
                       breaks = c(0, 20, 40, 60, 80),
                       labels = c("0","20", "40", "60", "80"),
                       colors = hcl.colors(7, palette = "RdYlBu")[7:1]) +
  labs(x="Year",y= "State", title="Opioid Overdose Deaths per 100k People") +
  labs(caption = paste("*Deaths refers to the number of deaths in a 12 month
                       period ending at a specific date" ,sep="\n")) +
  theme(plot.title= element_text(family="Times", face="bold", size=15),
        axis.text.y= element_text(size = 7),
        panel.grid= element_blank(),
        legend.position="right")
```



I also wanted to map the overdose rates just in 2022. To do this, I filtered the opioid overdose rates per 100k so it only included the date 2023-01-01, which would represent the number of deaths from the year 2022. From this map, we also see that West Virginia is the state with the highest death rate, and South Dakota and Iowa have low death rates. States in the west seem to be less impacted by the epidemic compared to states on the east coast.

```
#filtering to only counts for the year 2022
opioid_overdose_rates_per_100k_2022 <- opioid_overdose_rates_per_100k %>%
filter(Date == "2023-01-01")
```

```
#changing column name so it works with the map
names(opioid_overdose_rates_per_100k_2022)[1] = "state"

#mapping deaths per 100k in 2022
map <- plot_usmap(data = opioid_overdose_rates_per_100k_2022,
           values = "deaths_per_100000", color = "white", labels = TRUE) +
  scale_fill_gradient(name = "Opioid overdoses per 100k", low = "lightblue",
                      high = "darkblue", na.value = "gray") +
  labs(title = "Opioid Overdoses per 100k People in 2022", caption = "Gray
      states did not provide counts") +
  theme(plot.title=element_text(family="Times", face="bold", size=15),
        legend.position = "right")

map$layers[[2]]$aes_params$size <- 2
print(map)
```

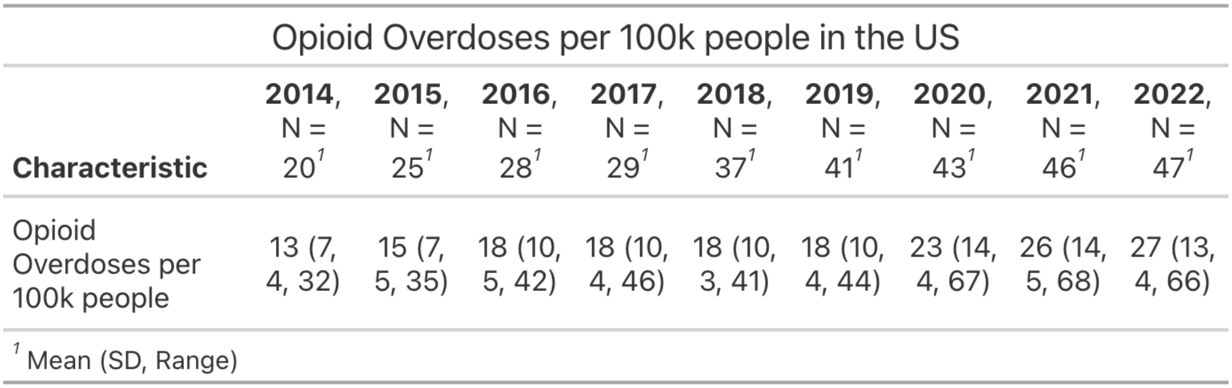


For further investigation of how opioid death rates changed over time, I created a table of the number of opioid overdoses per 100k people per year. To do this, I filtered the overdose data so it only included rows from January 2015-23. These rows represent the number of deaths for the 12 months prior to that month. So the January 2015 data would provide the number of deaths in a state in the year 2014. I used an inner join to merge the count data with population data, and then added a column for the opioid overdoses per 100k people, and adjusted the year column by subtracting one.

The table shows the mean deaths per 100k, using all the states that provided a death count for that year. N represents the number of states that provided a death count. We can see that death rate for each year is increasing over time.

Since we don't have data from each state for every year, if a state with a very high death rate wasn't included in the earlier data, but was included in the later data, it may make it look like the number of overdose deaths per year is increasing more than it actually is.

```
dates <- c("2015-01-01","2016-01-01","2017-01-01","2018-01-01","2019-01-01",
           "2020-01-01","2021-01-01","2022-01-01","2023-01-01")

#Getting mean number of opioid overdoses per 100k people
#Grouping by State and Year rather than State and Date like we did before
opioid_overdose_rates_per_100k2 <- opioid_overdoses %>%
  na.omit(opioid_overdoses) %>%
  filter(Date %in% dates) %>%
  inner_join(total_population_2020, by = c("State" = "state.abb")) %>%
  mutate("Opioid Overdoses per 100k people" = 100000 *Opioid_overdoses/value,
         Year = Year - 1)

#creating a table
table <-tbl_summary(opioid_overdose_rates_per_100k2, include= c("Year",
                    "Opioid Overdoses per 100k people"),
  by = "Year", statistic = list(all_continuous() ~ "{mean} ({sd}, {min},
                                {max})")) %>%
  as_gt() %>%
  tab_header(title = md("Opioid Overdoses per 100k people in the US"))
```



Opioid Overdoses per 100k people in the US

Characteristic	**2014**, N = 20[1]	**2015**, N = 25[1]	**2016**, N = 28[1]	**2017**, N = 29[1]	**2018**, N = 37[1]	**2019**, N = 41[1]	**2020**, N = 43[1]	**2021**, N = 46[1]	**2022**, N = 47[1]
Opioid Overdoses per 100k people	13 (7, 4, 32)	15 (7, 5, 35)	18 (10, 5, 42)	18 (10, 4, 46)	18 (10, 3, 41)	18 (10, 4, 44)	23 (14, 4, 67)	26 (14, 5, 68)	27 (13, 4, 66)

[1] Mean (SD, Range)

Now I would like to compare states that have had an increase in opioid prescribing rates in the last five years to states that have had a decrease in opioid prescribing rates.

First, I filtered to include all plan types, and only the year 2021. I then added a column that describes how prescribing rates have changed in the last five years for each state. I used a case_when() to create the different levels for this column (Greatly decreased, moderately decreased, slightly decreased, and increased. I then used an inner join to merge the prescribing rates data to the overdose rates data, since I wanted to include only the states present in both data frames

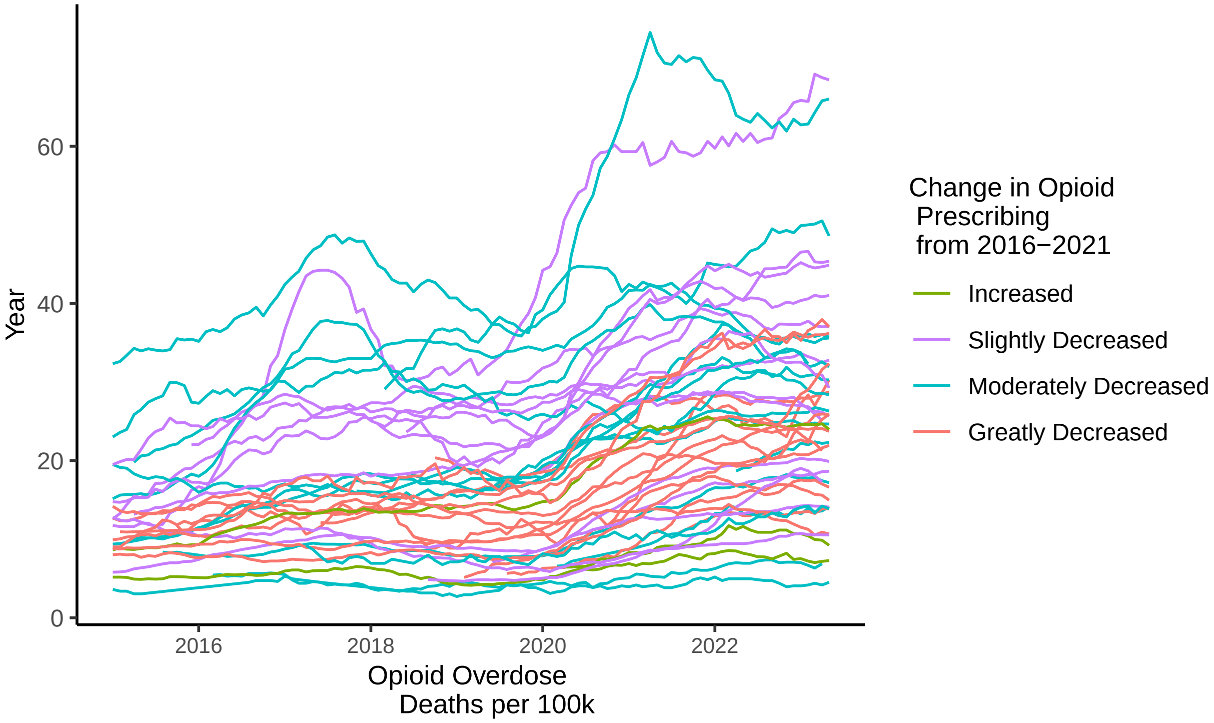
```
#Now I would like to compare states that have had an increase in opioid
#prescribing rate in the last five years to states that have had a decrease
#in opioid prescring rate

#filter to include all plan types and only the year 2021
#add a column to say if a state has increasing or decreasing prescribing rates
#in the last five years
#join with opioid rates
prescribing_overdose <- prescribing_rates %>% filter(Plan_Type == "All",
                                                     Year == "2021") %>%
  mutate(fiveyrchange = case_when(
    Opioid_Prscrbng_Rate_5Y_Chg < -3.2 ~ "Greatly Decreased",
    Opioid_Prscrbng_Rate_5Y_Chg < -2.4 ~ "Moderately Decreased",
    Opioid_Prscrbng_Rate_5Y_Chg < 0 ~ "Slightly Decreased",
    Opioid_Prscrbng_Rate_5Y_Chg >= 0 ~ "Increased")) %>%
  inner_join(opioid_overdose_rates_per_100k, by= c('state.abb'='State'))
```

I then plotted opioid overdose deaths over time for each state, colored by the change in opioid prescribing rate categories.

States that slightly decreased or moderately decreased opioid prescriptions from 2016-2021 seem to be the states that have experienced a higher rate of death from opioid overdoses. States that greatly decreased opioid prescriptions were mostly states that were in the middle in terms of opioid deaths, and these states generally stayed in the middle over time. The states that increased opioid prescriptions did not have a comparatively large opioid problem to begin with.

Although the states that decreased opioid prescriptions still experienced an increase in opioid overdoses, this increase may have been greater if opioid prescriptions had not decreased. It appears that the states that greatly decreased prescribing rates were able to keep their death rates somewhat under control over time.

```
ggplot(prescribing_overdose) +
  geom_line(aes(x=Date, y=deaths_per_100000, group=state.abb,
                color =fiveyrchange)) +
  labs(title = "Opioid Overdose Deaths Over Time per State by Change \n
       in Prescribing Rate of Opioids from 2016-2021", x = "Opioid Overdose
       Deaths per 100k", y = "Year") +
  theme_classic() +
  scale_color_discrete(name = "Change in Opioid \n Prescribing \n from 2016-2021",
    breaks=c('Increased', 'Slightly Decreased', 'Moderately Decreased','Greatly Decreased')) +
  theme(axis.text.x = element_text(size=8),
        plot.title = element_text(family="Times", face="bold", size=13),
        axis.title.y = element_text(size = 10),
        axis.title.x = element_text(size = 10),
        legend.title = element_text(size=10))
```


Opioid Overdose Deaths Over Time per State by Change in Prescribing Rate of Opioids from 2016−2021

I also plotted the percent change in opioid overdose deaths from 2019 to 2022. First I filtered to only include the states that had values for both 2020-01-01 and 2023-01-01. I then grouped by state and summarized the percent change. To summarize the percent change, I divided the opioid overdose rate in 2023 by the rate for 2020, using lag().

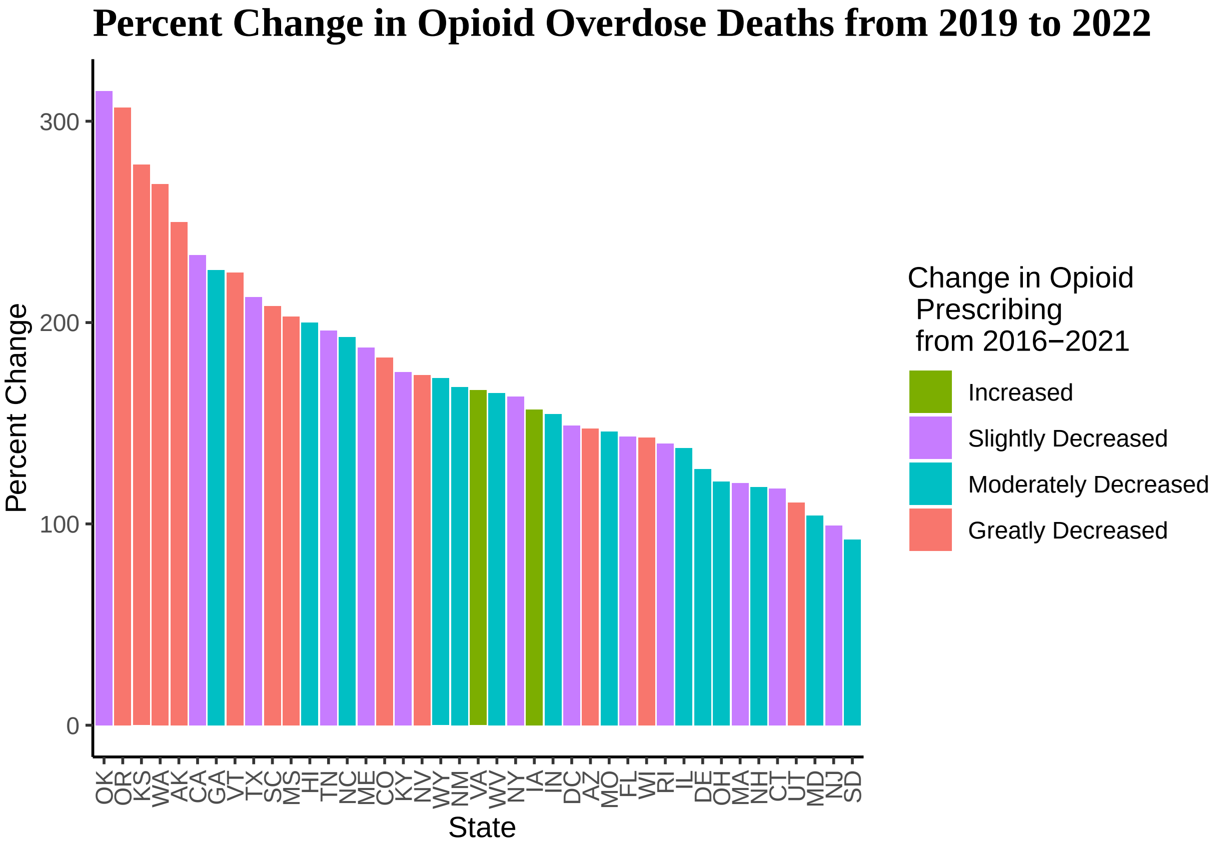
Next, I combined this data with prescribing rates data using an inner join, to get states that are included in both datasets.

I plotted the percent change in opioid deaths per state in a column chart, and filled each column by the prescribing rate change category.

```r
#get states that have values for both jan 2020 and jan 2023
opioid_overdoses_pull_states <- opioid_overdoses %>%
  filter(Date == "2020-01-01" | Date == "2023-01-01") %>%
  group_by(State) %>%
  summarize(number_of_dates = n()) %>%
  filter(number_of_dates == 2) %>%
  pull(State)

#Change column name from 'Date Value' to "Opioid_overdoses
names(opioid_overdoses)[5] = "Opioid_overdoses"

#filter to only rows from 2020-01-01 and 2023-01-01 and states that have
#values for both dates, then group by state and summarize percent change
opioid_overdoses2 <-opioid_overdoses %>%
  filter((Date == "2020-01-01" | Date == "2023-01-01") & State %in%
           opioid_overdoses_pull_states) %>%
  group_by(State) %>%
  summarize(prcnt_chng = (100*Opioid_overdoses/lag(Opioid_overdoses))) %>%
  na.omit()

#filter overdose data to only be for the year 2022. Combine percent change
#in overdose and change in prescribing datasets, omit NAs
prescribing_overdose2 <- prescribing_overdose %>%
  filter(Date == "2023-01-01") %>%
  inner_join(opioid_overdoses2, by = c("state.abb" = "State"))
```

Interestingly, some of the states that had the greatest decrease in opioid prescription rates from 2016 to 2021 had the highest percent increase in opioid overdose deaths from 2019 to 2022 (OR, KS, WA, AK). The states that had an increase in opioid prescriptions did not have a particularly high percent increase comparatively. This is potentially because prescribers in the states where the opioid epidemic hit the hardest did not want to worsen the problem by prescribing more opioids.

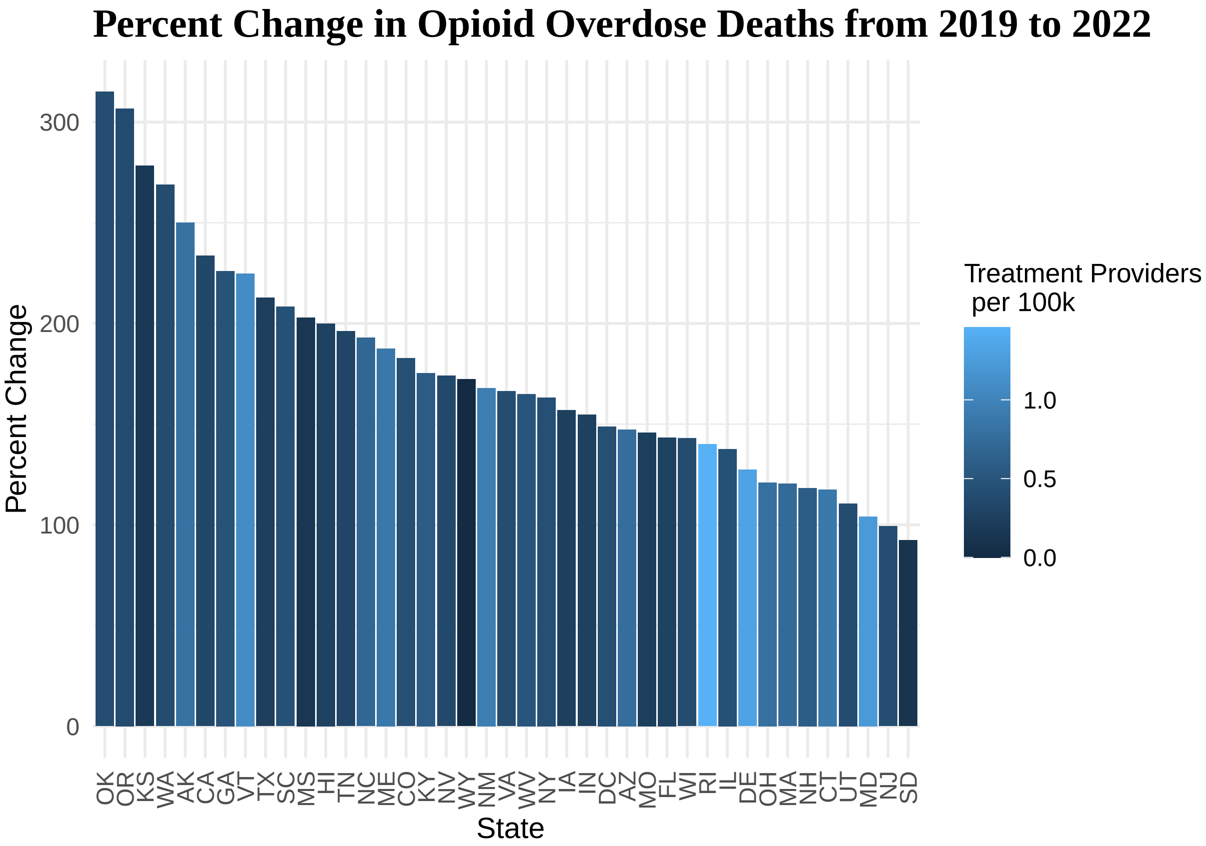
```r
#plot percent change in Opioid Deaths from 2019 to 2022
ggplot(prescribing_overdose2) +
  geom_col(aes(x= reorder(state.abb, -prcnt_chng), y= prcnt_chng,
               fill = fiveyrchange)) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_fill_discrete(name = "Change in Opioid \n Prescribing \n from 2016-2021",
    breaks=c('Increased', 'Slightly Decreased', 'Moderately Decreased','Greatly Decreased')) +
  labs(title = "Percent Change in Opioid Overdose Deaths from 2019 to 2022",
      x= " State", y= "Percent Change") +
  theme(plot.title=element_text(family="Times", face="bold", size=15))
```



My final analysis focused on plotting percent change, but grouped by the number of treatment providers this time. I combined the percent change data with the population data using an inner join, to get the states included in both data frames, and then produced the following column chart.

This plot shows the percent change in opioid overdose deaths from 2019 - 2022. Here we notice that the states with the highest percent change in opioid overdose deaths from 2019 to 2022 have a low number of treatment providers per 100k, and some of the states with a higher number of treatment providers per 100k had a lower percent change in deaths (RI, DE, MD). This could potentially imply that states who invest more in opioid treatment providers will more effectively prevent a large increase in deaths over time.

```
#combining the providers_pop df with the percent change data
overdoses_providers <- inner_join(opioid_overdoses2, providers_pop,
                                  by= c('State'='state'))

#plotting the percent change data again but with number of providers as the fill
ggplot(overdoses_providers) +
  geom_col(aes(x= reorder(State, -prcnt_chng), y= prcnt_chng,
               fill = providers_per_100000)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  theme(plot.title=element_text(family="Times", face="bold", size=15),
        legend.title = element_text(size=10)) +
  labs(title = "Percent Change in Opioid Overdose Deaths from 2019 to 2022",
       x= " State", y= "Percent Change") +
  scale_fill_continuous(name = "Treatment Providers \n per 100k")
```



References:

1. Azadfard M, Huecker MR, Leaming JM. Opioid Addiction. (2023). In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK448203/

2. Opioid use disorder treatment services. (n.d.). Opioid Use Disorder Treatment Coverage. https://www.medicare.gov/coverage/opioid-use-disorder-treatment-services

3. Data.cms.gov. (2021, December 22). Medicaid opioid prescribing rates - by geography. Centers for Medicare & Medicaid Services. HealthData.gov. https://healthdata.gov/dataset/Medicaid-Opioid-Prescribing-Rates-by-Geography/3fp8-zi9z

4. Data.cms.gov. (2021b, December 22). Opioid treatment program providers. HealthData.gov. Centers for Medicare & Medicaid Services. https://healthdata.gov/dataset/Opioid-Treatment-Program-Providers/vm5j-fnkk

5. Data.cdc.gov. (2021, February 25). VSRR provisional drug overdose death counts. HealthData.gov. Centers for Disease Control and Prevention. https://healthdata.gov/dataset/VSRR-Provisional-Drug-Overdose-Death-Counts/and2-axw4

6. Walker K, Herman M. (2023). tidycensus: Load US Census Boundary and Attribute Data as 'tidyverse' and 'sf'-Ready Data Frames. R package version 1.5. https://walker-data.com/tidycensus/.

7. Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, *4*(43), 1686. doi:10.21105/joss.01686 https://doi.org/10.21105/joss.01686.

8. Iannone R, Cheng J, Schloerke B, Hughes E, Lauer A, Seo J (2023). *gt: Easily Create Presentation-Ready Display Tables*. R package version 0.10.0, https://CRAN.R-project.org/package=gt.

9. Sjoberg DD, Whiting K, Curry M, Lavery JA, Larmarange J. Reproducible summary tables with the gtsummary package. The R Journal 2021;13:570–80. https://doi.org/10.32614/RJ-2021-053.

10. Di Lorenzo P (2023). usmap: US Maps Including Alaska and Hawaii. R package version 0.6.2. https://CRAN.R-project.org/package=usmap

11. Grolemund G, Wickham H (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.