Question 6, Lab 3

state\_covid = covid\_data %>% group\_by(date, state) %>% summarise(cases = sum(cases)) 🡪 A dataset where each row contains the **total COVID-19 cases per state per date**.

group\_by(date, state)

* Groups data by **date and state** so that calculations are performed separately for each state on each date.

summarise(cases = sum(cases))

* Sums the total number of cases reported per **state** per **day**.

filter(state %in% c('New York', "Ohio", 'Colorado', "Alabama"))

* keeps only data for New York, Ohio, Colorado, and Alabama while removing records for all other states

group\_by(state)

* ensures that subsequent calculations happen separately for each state.

mutate(newCases = cases - lag(cases))

* computes the number of new cases each day by subtracting the previous day’s total cases from the current day's total cases.

mutate(roll = zoo::rollmean(newCases, k = 7, align = "right", fill = NA))

* calculates a 7-day rolling average of new cases to smooth daily fluctuations
  + The k = 7 argument specifies that the average includes the current day and the six previous days
  + The align = "right" setting ensures that each computed value represents the average of the previous seven days
  + The fill = NA argument fills the first six rows with NA since there aren’t enough prior values to calculate the average

ungroup()

* removes the grouping structure so that further operations do not inherit the group\_by(state) setting

**The final dataset, state\_covid, contains the total cases per state per day, the number of new cases per day, and the 7-day rolling average of new cases. This dataset can be used for visualization or further analysis.**

ggplot(state\_covid, aes(x = date))

* initializes a ggplot object using the state\_covid dataset. The x-axis represents the date

geom\_col(aes(y = newCases), fill = "pink", col = NA)

* adds bar charts to represent the daily new COVID-19 cases
* The height of each bar is determined by the newCases variable
* The bars are filled with pink, and col = NA removes any border color from the bars.

geom\_line(aes(y = roll), col = "darkred", size = 1)

* overlays a line plot on the bars, representing the 7-day rolling average of new cases.
* The line is colored dark red and has a width of 1 for better visibility.

theme\_linedraw()

* applies a clean, simple theme with black axis lines and minimal distractions.

facet\_wrap(~state, nrow = 2, scales = "free\_y")

* creates separate plots for each state using facet wrapping
* The nrow = 2 argument arranges the plots in two rows
* The scales = "free\_y" setting allows each state's y-axis to be independently scaled, preventing small-case states from being overshadowed by larger-case states

labs(title = "Cumulative COVID-19 Cases", x = "Date", y = "Case Count")

* adds a title and labels for the x and y axes
* The title describes the plot, while the x-axis represents the date, and the y-axis represents the number of cases

**The final visualization displays new daily COVID-19 cases as pink bars, with a dark red line showing the 7-day rolling average. Each state has its own plot, arranged in two rows, making it easier to compare trends across states.**

group\_by(STNAME)

* groups the population dataset by state name (STNAME), ensuring that calculations are performed separately for each state

summarise(state\_pop = sum(POPESTIMATE2021))

* computes the total estimated population for each state in 2021 by summing up population estimates

inner\_join(state\_covid, by = c("STNAME"="state"))

* merges the population data with the COVID-19 dataset
* The STNAME column from the population data is matched with the state column from state\_covid, ensuring each state’s COVID-19 data is linked to its population

mutate(perCap = newCases / state\_pop)

* calculates per capita new cases by dividing the daily new cases by the total state population.

group\_by(STNAME)

* ensures that subsequent calculations are performed separately for each state

mutate(roll = zoo::rollmean(perCap, k = 7, align = "right", fill = NA))

* computes a 7-day rolling average of per capita new cases
* The k = 7 argument specifies that the average includes the current day and the six previous days
* The align = "right" setting ensures that each computed value represents the average of the previous seven days
* The fill = NA argument fills the first six rows with NA since there aren’t enough prior values to calculate the rolling average

ungroup()

* removes the grouping structure so that further operations do not inherit the group\_by(STNAME) setting

**The final dataset, pp, contains the total state population, per capita daily new COVID-19 cases, and the 7-day rolling average of per capita cases. This dataset allows for comparison of COVID-19 spread across states while accounting for population differences.**

ggplot(pp, aes(x = date))

* initializes a ggplot object using the dataset pp. The x-axis represents the date.

geom\_line(aes(y = roll, col = STNAME), size = 1)

* adds a line plot to represent the 7-day rolling average of per capita new cases
* The y-axis value is determined by the roll column
* The col = STNAME argument assigns a different color to each state’s line, allowing for easy comparison
* The size = 1 setting makes the lines slightly thicker for better visibility.

theme\_linedraw()

* applies a clean, simple theme with black axis lines and minimal distractions.

labs(title = "Cumulative COVID-19 Cases", x = "Date", y = "Case Count")

* adds a title and labels for the x and y axes
* The title describes the plot, while the x-axis represents the date, and the y-axis represents the number of per capita cases.

**The final visualization displays the 7-day rolling average of per capita new cases for each state, with different colors distinguishing different states. This allows for a clear comparison of trends over time across multiple states.**