Question 7, Lab 3

**read\_csv(...)**

* Reads a CSV file containing county centroids (longitude LON and latitude LAT for each county).
* This dataset contains county fips codes, which are **unique county identifiers**.
* **FIPS (Federal Information Processing Standards) codes** are unique numerical identifiers assigned to geographic areas in the United States. These codes are used by federal agencies to standardize geographic data

**inner\_join(covid\_data)**

* Joins county-centroids data with covid\_data using a common key (fips).
* Keeps only records that **exist in both datasets** (i.e., counties with COVID-19 case data).
* Ensures each COVID-19 case record is linked to the correct county coordinates

**group\_by(date)**

* Groups data by **each unique date** to process daily COVID-19 case distributions separately.

**summarise(…) 🡪** This step calculates the daily "epicenter" of the pandemic in terms of latitude and longitude

* Computes summary statistics for each date:
  + wmX\_c = sum(LON\*cases) / sum(cases)
    - **Weighted mean longitude** of COVID-19 cases.
    - Gives more weight to counties with higher cases.
  + wmY\_c = sum(LAT\*cases) / sum(cases)
    - **Weighted mean latitude** of COVID-19 cases.
    - Shows the central location of the outbreak for each date.

**cases = sum(cases)**

* Computes the **total number of cases** for each date.

**arrange(date)**

* Sorts the dataset in chronological order

**mutate(d = 1:n()) 🡪** Ensures the dataset is in the correct order for **time-based visualizations.**

* Creates a new column (d):
  + Assigns **sequential numbers (1, 2, 3, ...)** to the dates.
  + Helps with **time series visualization** (e.g., animations).

**ggplot(meta)**

* Creates a **ggplot** object using the dataset meta

Meta (the dataset I created) contains:

* wmX\_c → Weighted mean longitude (X-axis).
* wmY\_c → Weighted mean latitude (Y-axis).
* cases → Total number of cases per date.

**borders("state", fill = "gray90", colour = "white")** 🡪 creates a background map for context

* borders(“state”)
  + adds state boundaries pf the US
* fill = “gray90”
  + fills states with light gray
* colour = “white”
  + draws state borders in white

**geom\_point(aes(x = wmX\_c, y = wmY\_c, size = cases), color = "red", alpha = .25)** 🡪 each red dot represents the daily epicenter of COVID cases

* geom\_point(…)
  + plots points on the map
* aes(x = wmX\_c, y = wmY\_c, size = cases)
  + **X-axis:** Weighted mean **longitude** (wmX\_c)
  + **Y-axis:** Weighted mean **latitude** (wmY\_c)
  + **Size:** Proportional to **number of cases**
* **color = “red”**
  + **makes points red**
* **alpha = .25**
  + **adds transparency to precent overcrowding**

**theme\_linedraw()**

* Uses a **minimalistic theme** with black axis lines

**labs(color = "Time", size = "Cases", x = "", y = "", title = "Weighted Center of COVID-19 Cases")** 🡪 helps viewers understand the data

* color = "Time"
  + **( Not used here**, can be removed unless coloring by time)
* size = "Cases"
  + Legend label for point size
* x = "", y = ""
  + Removes **axis labels** for a cleaner map
* title = "Weighted Center of COVID-19 Cases"
  + Sets the **plot title**
* theme(legend.position = "none")
  + hides the legend

**Recap of what the code does:**

 Loads county centroid data, which includes longitude and latitude coordinates for each county.

 Joins this dataset with COVID-19 case data using county FIPS codes to match case data with county locations.

 Groups the data by date and computes the weighted mean center of cases based on county coordinates and reported cases.

 Orders the data chronologically and adds a time index to facilitate time-series visualization.

 Creates a U.S. map as a background for plotting.

 Plots the weighted mean center of COVID-19 cases for each day as a red dot, with larger dots representing higher case counts.

 Uses a minimalistic theme to keep the visualization clean and readable.

 Removes unnecessary legends to avoid clutter.