<u>HOMEWORK – 2</u>

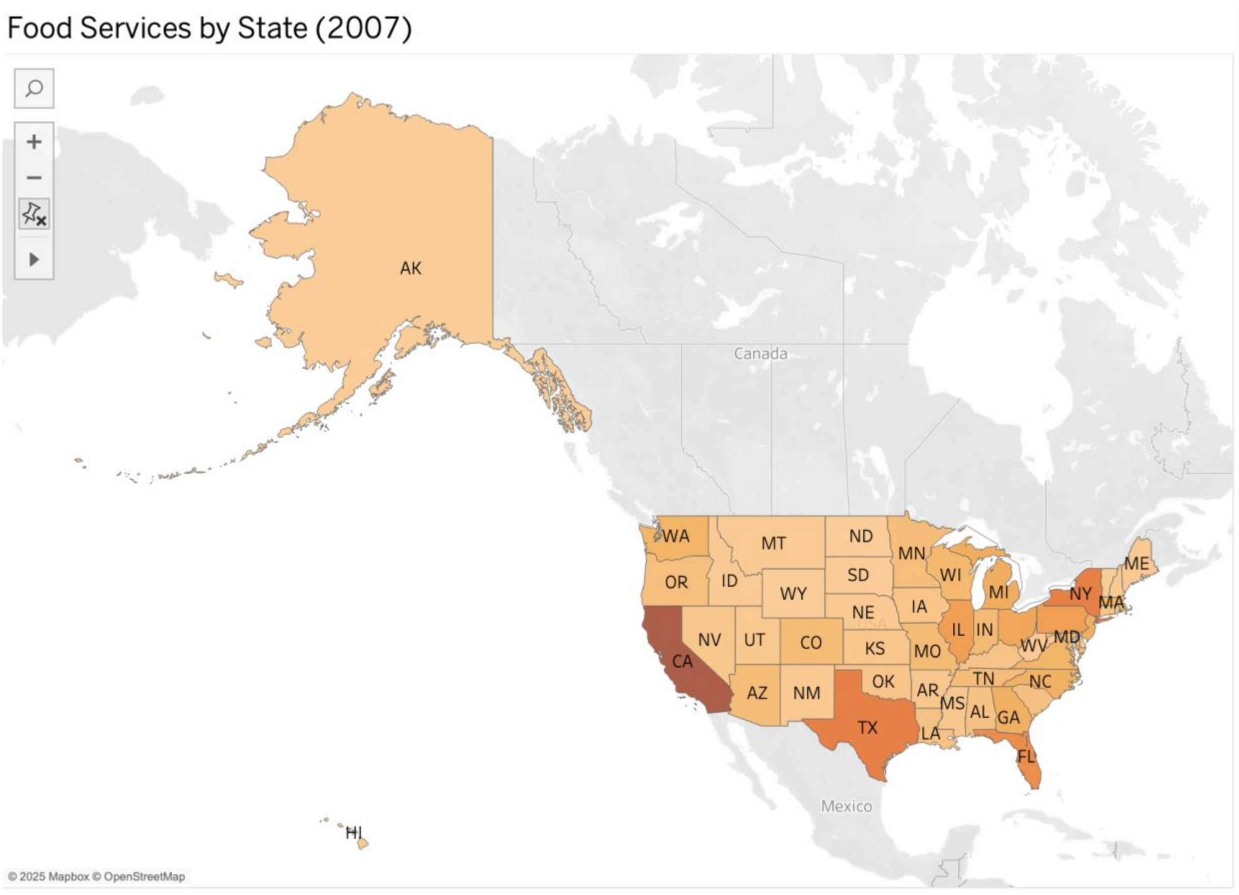
1) DownloadtheFoodSrvcByCounty.txtfileandcreatethefollowingvisualizationsforthis geographical data. The data is for the availability of food services by county in the U.S. It also has data by state (in the county field, some of them have the state names, and those rows hold the state totals, or you can aggregate by state)

a) Graph food services by state with an appropriate geographic visualization. Note any patterns that arise. Your visualization should clearly display states that have high levels or low levels of food service availability, so think carefully about the color scheme.

Steps :

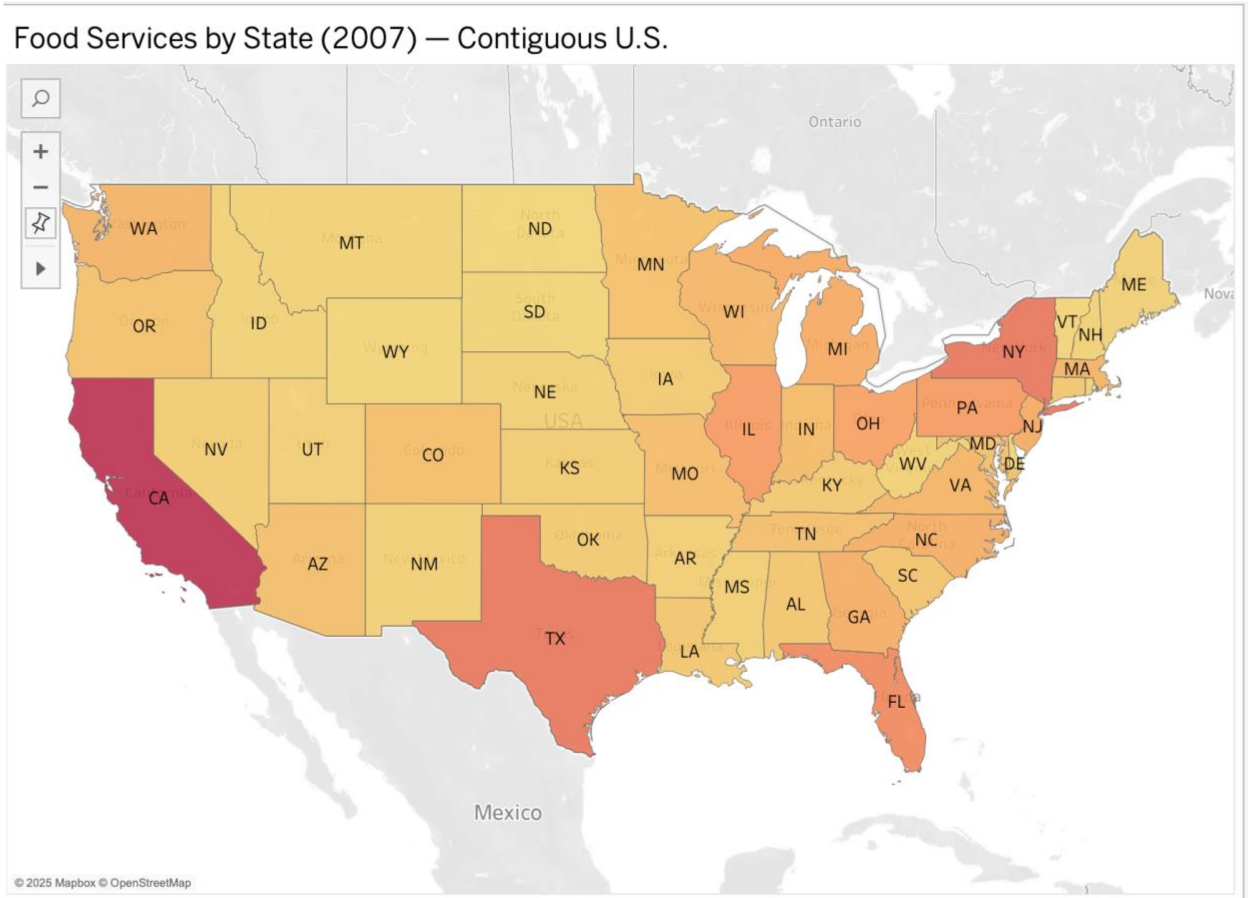
Open Tableau - Connect - Text file - select FoodSrvcByCounty.txt.

On a new worksheet, I dropped **State** onto the view and set the **Marks** type to **Map**, then dragged **FoodServices-2007** to **Color** and set it to **Measure → Sum** so I'm showing total food-service locations per state. I edited the colors to a clear, single-hue sequential palette (darker = more), added **State** to **Label** for easy reading, and filtered the data so states with **null** values don't appear. Finally, I gave it a clear title **'Food Services by State (2007)'** . kept the legend visible, and exported the image for my PDF.



Another way :

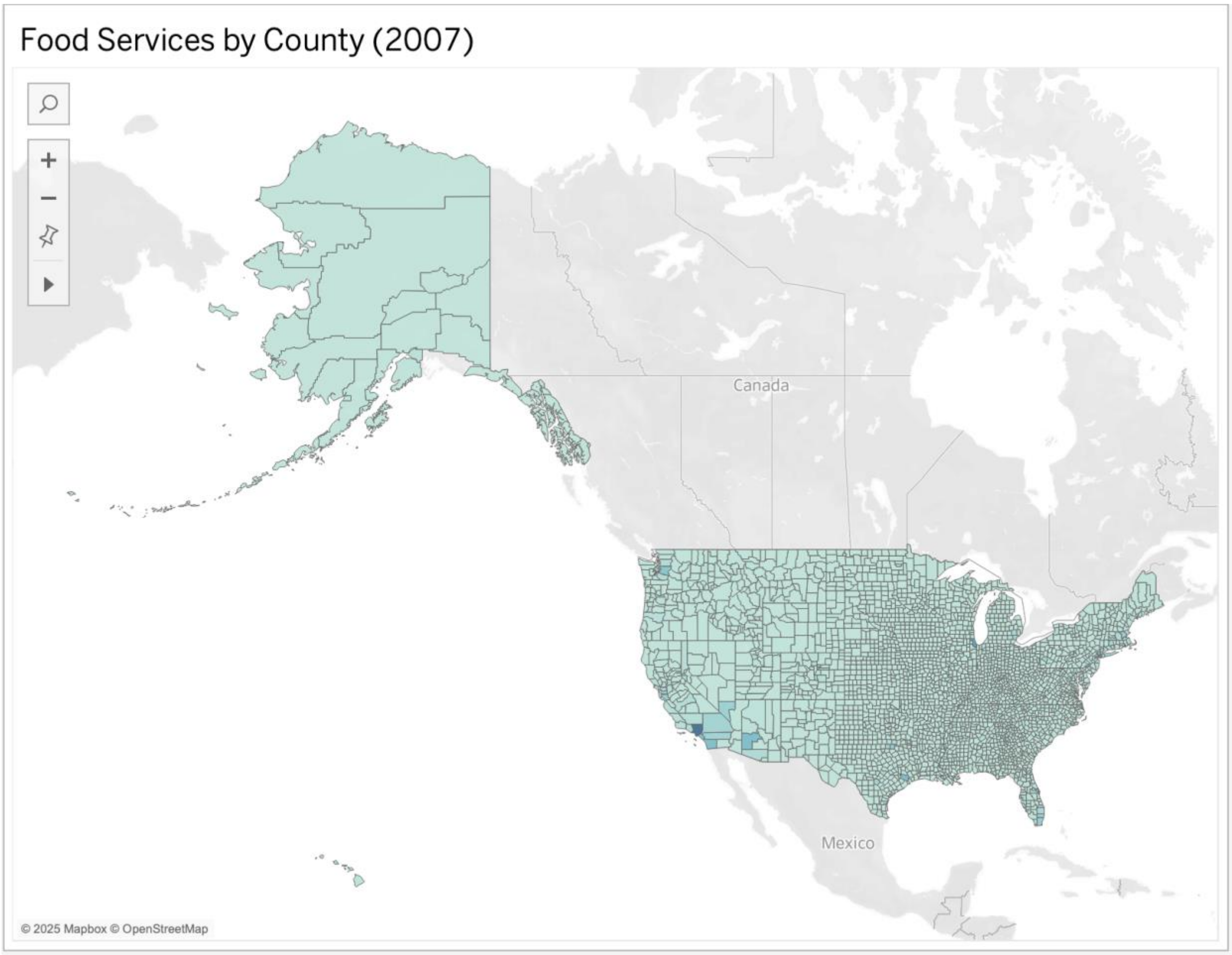
All steps are same but applied additional filter to exclude state Alaska AK and Hawai HI for better readability .



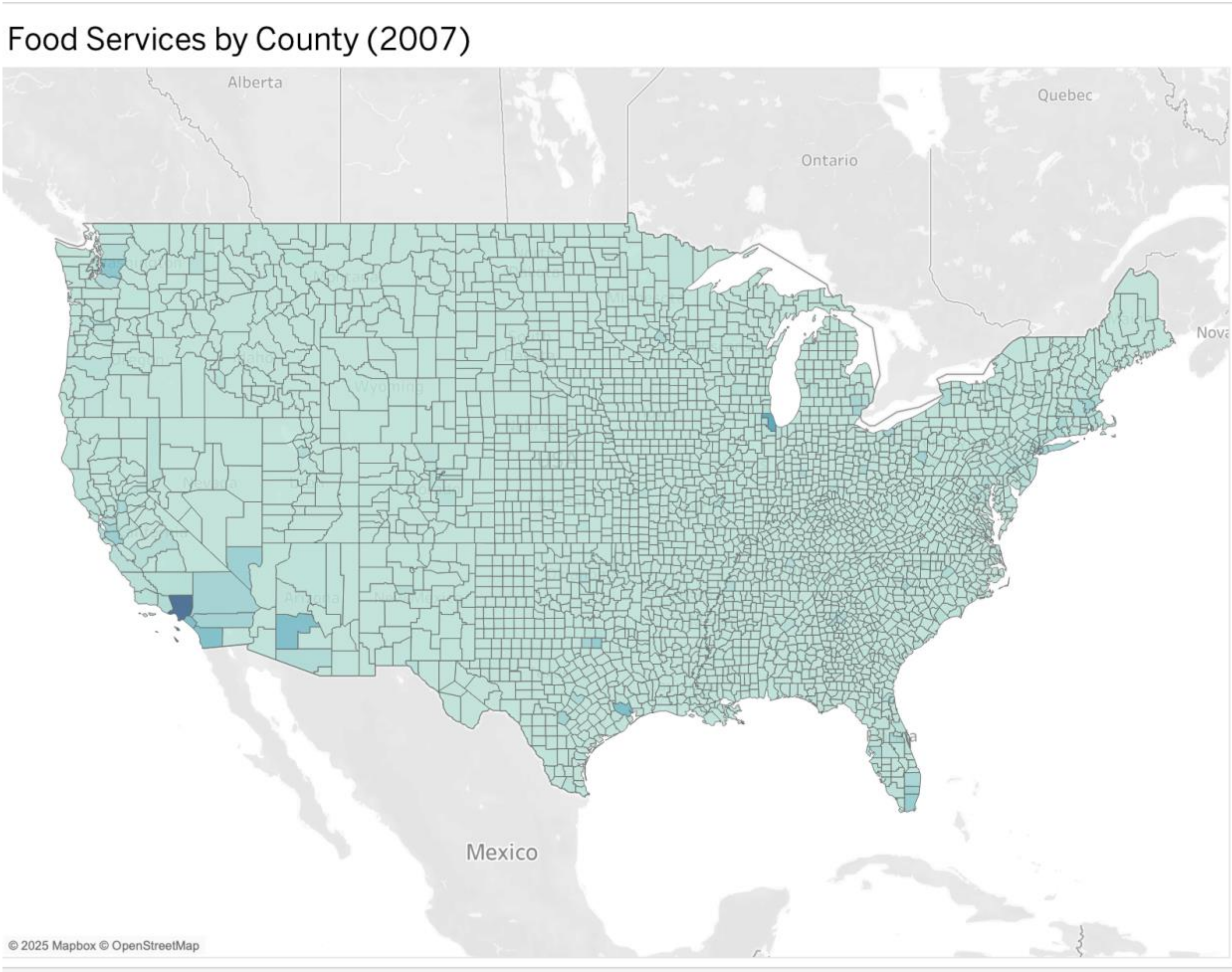
Food Services by State (2007) — Contiguous U.S.

b.Graph food services by county with the same type of visualization. Again, think carefully about the color scheme.

Imported FoodSrvcByCounty.txt text file , set State to *State/Province* and County to *County*, and filtered out the state-total rows by keeping only records with a non-null State. Then I built a Filled Map by placing State on the view and County on Detail, colored the counties with SUM(FoodServices-2007), and cleaned the few “unknown” spots via Map → Edit Locations… (or by filtering the nulls). I applied a clear single-hue sequential palette, added optional extracted borders for contrast, left labels off in favor of tooltips (county, state, total), optionally in another visualization I excluded AK/HI for a contiguous view, titled the chart, and exported it.



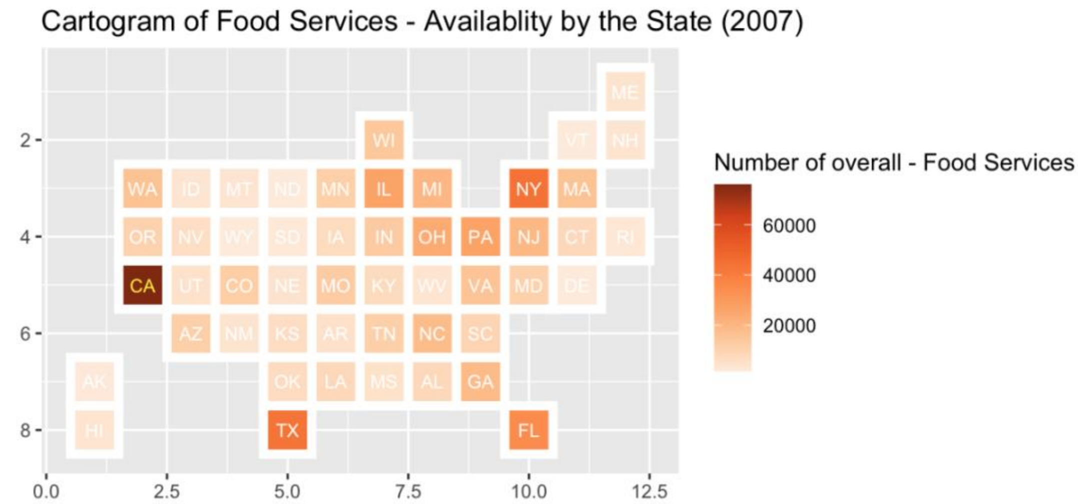
By applying filter over State – Null , AK and HI for better readability



c.(Extra credit) Research how to do a diffusion or tile cartogram in R or D3 and create a cartogram of the state data from this dataset.

Explanation :

I created a tile cartogram in R using statebins after aggregating the dataset to state totals for 2007 (sum of FoodServices-2007). Each state is a fixed-size square in a U.S.–like grid, so area is constant and color encodes the data (sequential orange: darker = more establishments). This avoids large land-area states dominating the view and makes value comparisons straightforward. The pattern is clear: California, New York, Texas, Florida, and Pennsylvania stand out, while many Mountain–Plains states are lighter. Alaska and Hawaii are included as tiles for a consistent national layout. The full R code to build this cartogram is included in the attached .Rmd file.



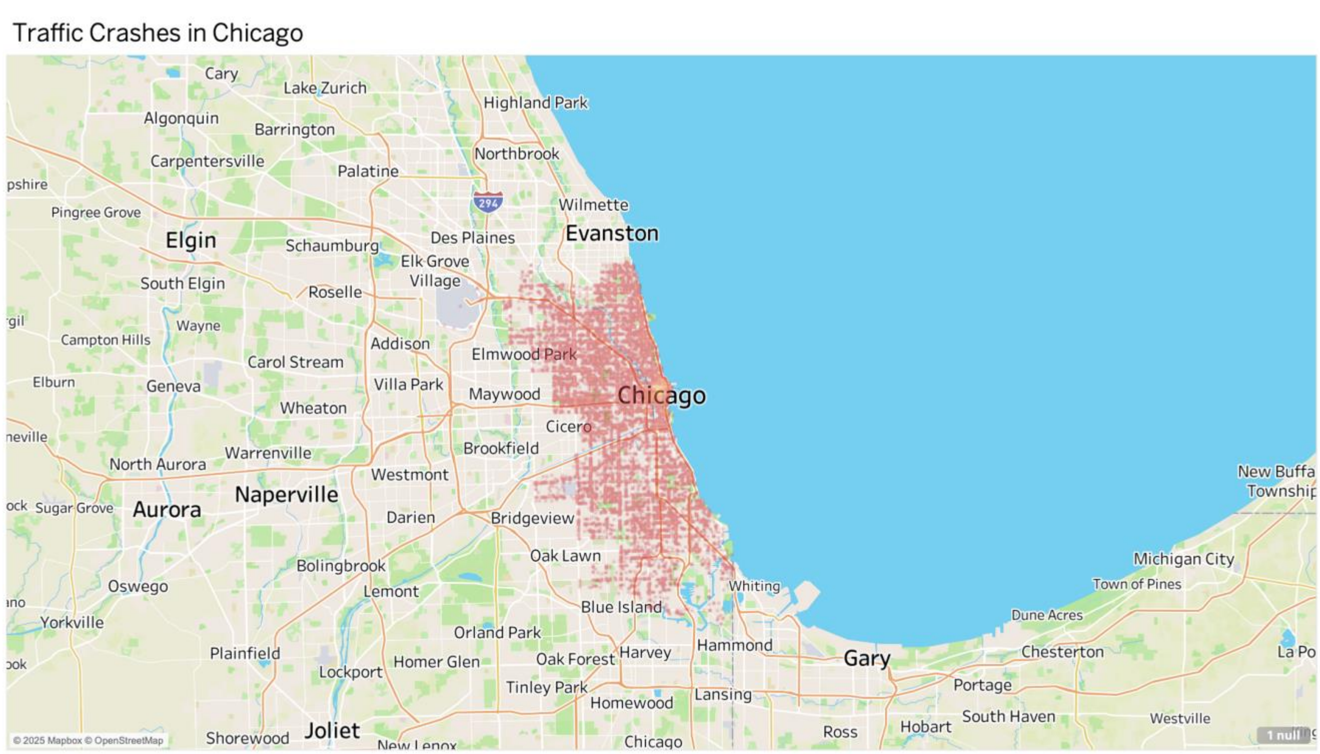
2
TheChicago_crashes.csvfilecontainsinformationoneverycrashrecordedinChicagoinJune 2019 (see Chicago's portal at https://data.cityofchicago.org/Transportation/Traffic-Crashes- Crashes/85ca-t3if for the latest data. I chose a random month because the data get dense quickly).

a)Create an appropriate type of geographic plot to show where all the accidents in this data occur.

I built this in Tableau as a density (heat) map of Chicago crash locations for June 2019. I put **Longitude** on Columns and **Latitude** on Rows, then switched the Marks card to **Density** to turn overlapping points into a smooth hotspot map. I used a red→pink gradient: deep red = highest crash concentration, light pink = lower density.

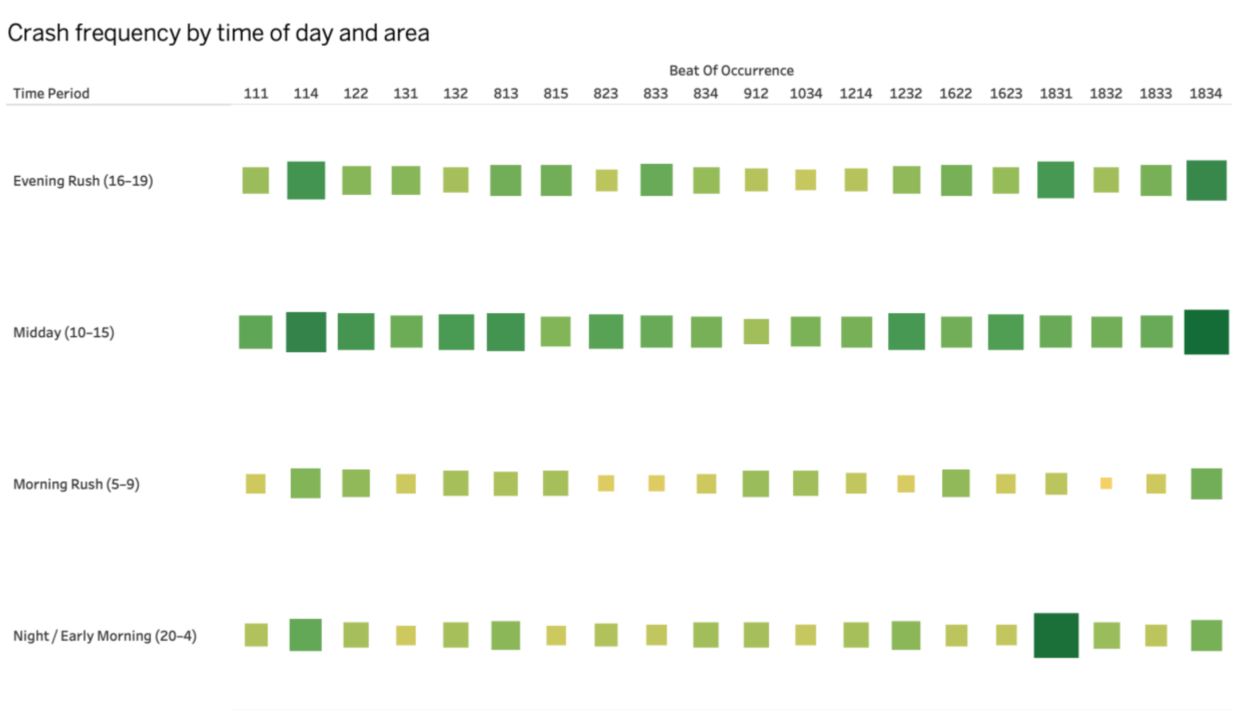
What it shows is pretty clear: crashes cluster heavily inside the city limits, with the strongest hotspot downtown and stretching up along the lakefront. The pattern follows the dense street grid city neighborhoods light up, while the outer suburbs stay mostly pale. Western suburbs like Naperville/Aurora/Elgin and areas south toward Joliet show much lighter activity. Compared with a raw scatter of points, this density view makes the spatial pattern instantly readable: Chicago's urban core saw far more incidents during that month.

I chose Map type as density to avoid more clutters.



b) Create a visualization that shows how common crashes are in different parts of the city based on time of day. There are multiple approaches to this. Explain your approach and what you can see in your graph.

I built the heatmap by putting Beat of Occurrence on Columns and Time Period on Rows, then placing COUNT(Number of Records) on Color with the Square mark type. I switched the palette to a yellow-to-green gradient (dark green = more crashes, light yellow = fewer) and sorted the beats to make the pattern easier to read. I placed Beat Of Occurrence on Columns (20 different police beats) and Time Period on Rows (four time windows: Night/Early Morning, Morning Rush, Midday, Evening Rush), using a yellow-to-green gradient where dark green = high crash concentration. I finished with a clear, descriptive title.

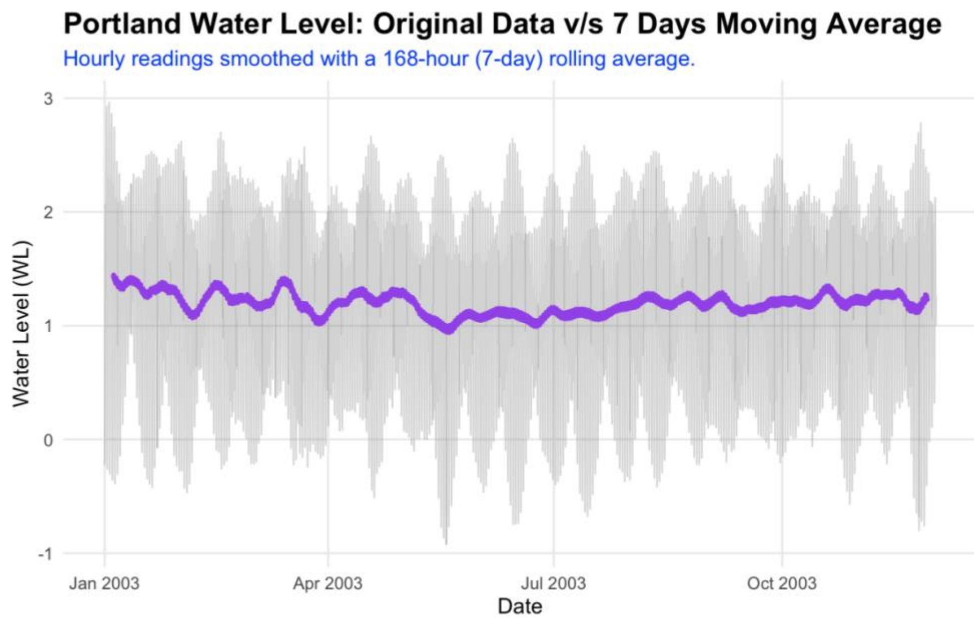


3.DownloadthePortlandWaterLeveldatasetandexploreitbycreatingthefollowing visualizations of the time series from the techniques described in lecture. Use both R and Tableau for at least one question part. They should, of course, adhere to the design criteria that we’ve learned, and should clearly display the information described in each part.

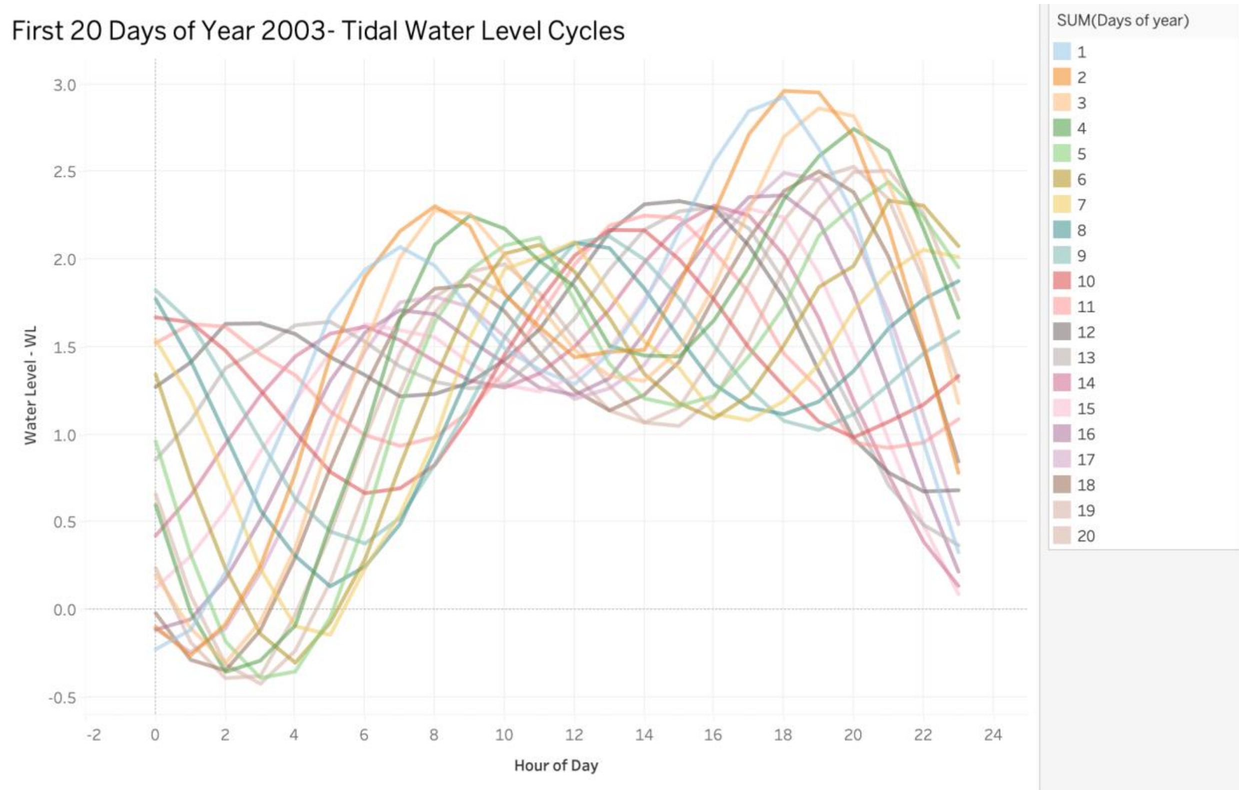
a. This data contains a year of data with water level (WL) measurements every hour as a function of Time (i.e. 365 x 24 data points!). Since there is a lot of data, clean it up by smoothing the data by calculating a moving average. Use a window approach with window size that covers a range of days (remember, the data is hourly) and graph the smoothed result. Work with the window to see what size window gives you the best view of the changes in the data while still smoothing the noise well. Remember that the moving average is in the Quick-Table calculations inside of the right click menu on the data item in Tableau, and we can compute it in R quite easily as shown in the tutorial.
Explanation:

I did this in **R (RMarkdown)** by reading the PortlandWaterLevel CSV, parsing the **Time** column to POSIXct, and using the **WL** (water level) column to compute a **centered 7-day (168-hour) moving average** with zoo::rollmean. I then **joined the smoothed series back by Time**, so the working table contains Time, WL, and WL_MA_7d. With **ggplot2** I overlaid a faint line for the original **WL** and a bold line for **WL_MA_7d**, added titles/labels, and saved the figure. All code and output are included in the attached. Rmd for full reproducibility.

My csv file path: ~/Desktop/Fall Quarter/Data Visualization/HW-2/DATA VISUALIZATION (Tue) - DSC.465 F.25 - 1062025 - 1023 PM/PortlandWaterLevel2003.csv



b. Graph the cycles that happen each day (because of tides). You might try overlapping many days' data as separate overlapping time series, using a level plot, a horizon graph, etc. The point of this exercise is to try to come up with a way of showing the progression of the tides over some period of time that is rich and detailed, and which shows the pattern, but which is still readable and which doesn't clutter the graph.



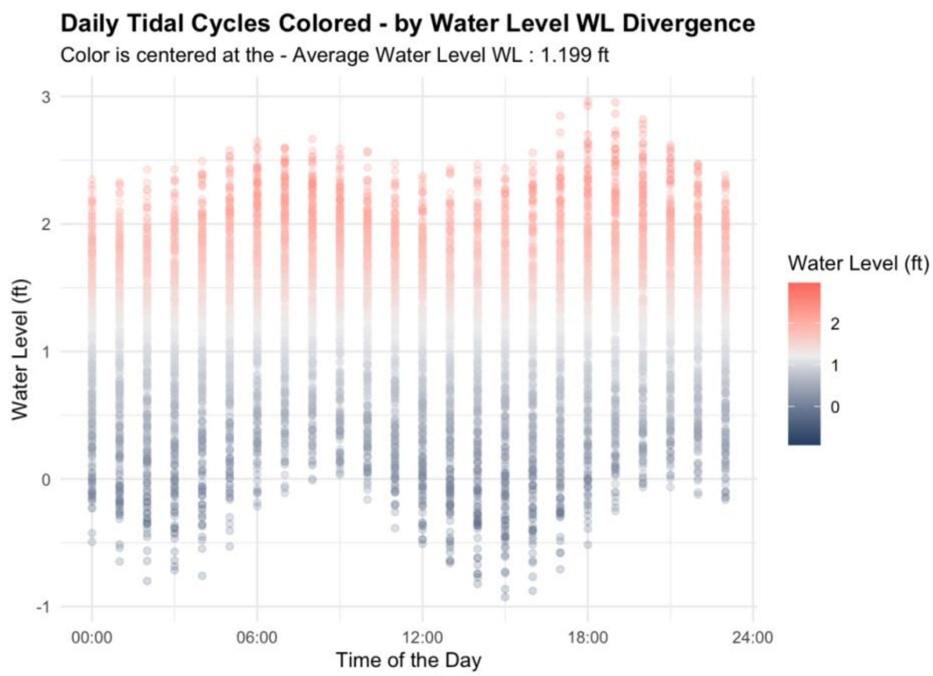
Explanation:

I built the daily tidal-cycle view in Tableau by adding two calcs: Hour of Day (DATEPART('hour',[Time])) and Day of Year (via DATEDIFF). Then I put Hour of Day (as a *dimension*) on Columns and WL on Rows, set Marks to Line, and dropped Date on *Detail* so each day draws its own line. I colored by Day of Year (set to *discrete*) and filtered it to 1–20 to keep only the first 20 days. With line opacity around 50–60%, the overlapping curves are easy to read, clearly showing the semi-diurnal pattern two highs and two lows per day throughout early January 2003.

c. *Then write a single paragraph outlining the differences between the information that each graph communicates.*

These two charts work together to show different aspects of the water level data. The 7 days moving average removes the daily tidal ups and downs, letting us focus on broader patterns like seasonal shifts, gradual changes over weeks, and overall trends throughout the year. This view makes it easy to compare different time periods and spot any unusual long term behavior. Meanwhile, the overlapping lines preserve all the tidal detail, revealing the natural rhythm of two high tides and two low tides each day. We can see how the timing of tides shifts gradually and how the tide heights vary from one day to the next. This perspective is perfect for studying the actual tidal cycles and their day-to-day variations. In practice, the smoothed chart sacrifices tidal timing to highlight overall water level changes, while the daily chart maintains all the tidal information but obscures any long-term drift. Using both visualizations gives us a comprehensive view :- one reveals the forest i.e. long-term trends , while the other shows the trees i.e. individual tidal patterns .

4) ReturntothePortlandWaterLeveldataset.RecreateoneofyourplotsfromQuestion3witha custom color scale. Specifically, create a divergent color scale with the average water level at the midpoint and two separate colors used to show when the water is getting very high and very low. The point of this exercise is to experiment with creating a color scale, so choose your own distinctive colors to use for the endpoints and center. Make sure that they are reasonable choices given what you know about color scales. Use HSV space to choose the colors and explain how you made your decision. In Tutorial 4, you can see how to create color scale in ggplot that is interpolated in Lab space.



Explanation :

I graphed every hourly water-level reading from 2003 by time of day (x-axis) with water level (ft) on the y-axis. Points use a diverging color scale centered on the yearly mean (~1.199 ft): cool greens mark below-average levels, warm yellow/orange mark above-average. Because all days are overlaid on the same 0–24h frame, the semi-diurnal pattern jumps out—two highs and two lows most days—with gentle phase shifts as the bands drift across the day. Warmer bands cluster around morning/late afternoon, cooler bands around late night/early afternoon, making typical high/low windows obvious. Transparency reduces overplotting so the density pattern stays readable.