

# DV HW2

Kavana Manvi KrishnaMurthy

2024-10-05

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following objects are masked from 'package:base':  
##  
##    date, intersect, setdiff, union
```

```
library(zoo)
```

```
##  
## Attaching package: 'zoo'  
  
## The following objects are masked from 'package:base':  
##  
##    as.Date, as.Date.numeric
```

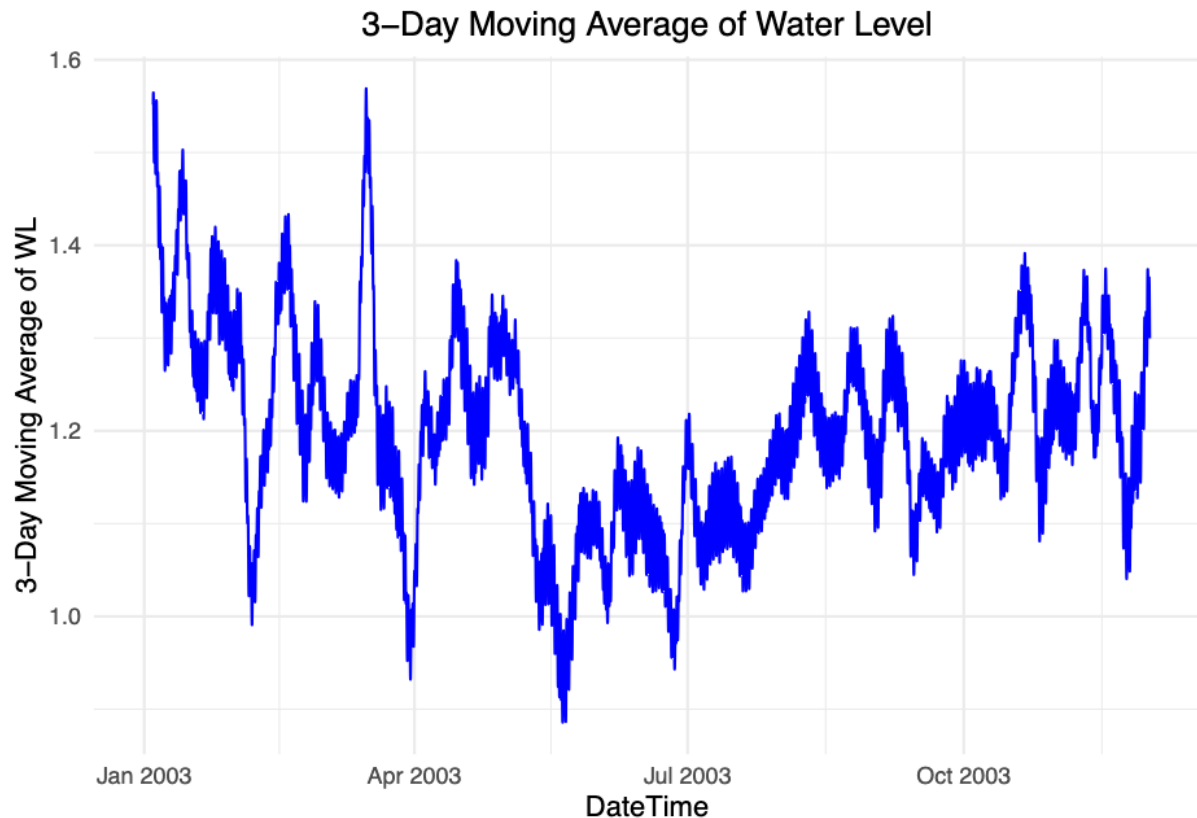
```
library(ggplot2)  
portland_wl <- read.csv("~/Desktop/DV/HW2/PortlandWaterLevel2003.csv", stringsAsFactors=TRUE)
```

```
portland_wl$DateTime <- paste(portland_wl$Date, portland_wl$Time)  
portland_wl$DateTime <- mdy_hm(portland_wl$DateTime)  
  
portland_wl$DateTime <- as.POSIXct(portland_wl$DateTime)
```

```
# Calculate the 3-day (72 hours) moving average  
portland_wl$MA_3days <- rollmean(portland_wl$WL, k = 72, fill = NA, align = "right")
```

```
# Create the plot for the 3-day moving average  
ggplot(portland_wl, aes(x = DateTime, y = MA_3days)) +  
  geom_line(color = "blue", size = 0.5) +  
  labs(  
    title = "3-Day Moving Average of Water Level",  
    x = "DateTime",  
    y = "3-Day Moving Average of WL"  
  ) +  
  theme_minimal() +  
  theme(plot.title = element_text(hjust = 0.5)) # Center the title
```

```
## Warning: Removed 71 rows containing missing values or values outside the scale range
## ('geom_line()').
```



Heatmap

```
portland_wl <- read.csv("~/Desktop/DV/HW2/PortlandWaterLevel2003.csv",
                        stringsAsFactors=TRUE)
```

```
portland_wl$Date <- as.Date(portland_wl$Date, format="%m/%d/%Y")
portland_wl$Time <- as.numeric(substr(portland_wl$Time, 1, 2))
```

```
## Warning: NAs introduced by coercion
```

```
# Create the heatmap
heatmap_plot <- ggplot(portland_wl, aes(x = as.factor(day(Date)), y = Time,
                                         fill = WL)) +
  geom_tile() +
  labs(
    title = "Daily Tidal Cycles (Day vs Hour)",

```

```

    x = "Day",
    y = "Hour of Time",
    fill = "Tide Patterns"
  ) +
  scale_fill_gradient(low = "light blue", high = "dark blue") +
  scale_y_continuous(breaks = seq(0, 23, by = 1)) +
  theme_minimal()

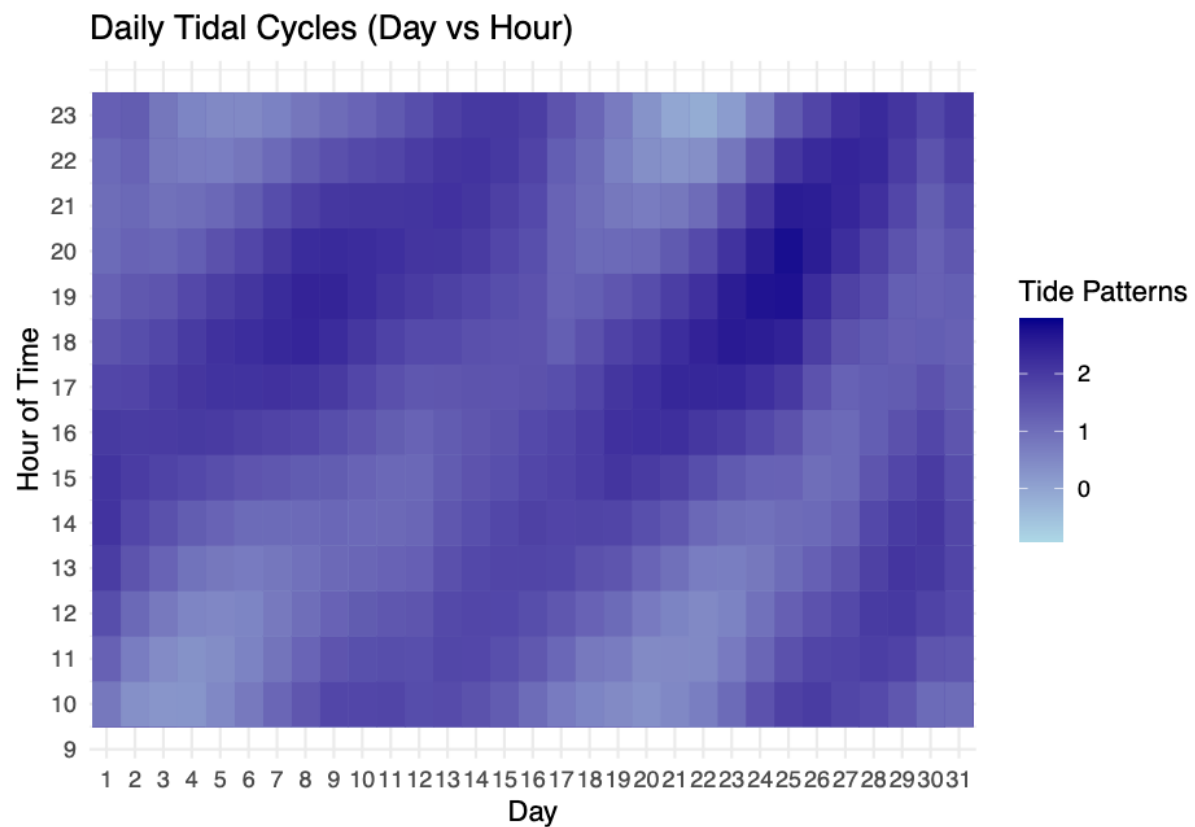
# Print the plot
print(heatmap_plot)

```

```

## Warning: Removed 3350 rows containing missing values or values outside the scale range
## ('geom_tile()').

```



```

heatmap_plot_custom_divergent <- ggplot(portland_wl,
  aes(x = as.factor(day(Date)), y = Time, fill = WL)) +
  geom_tile() +
  labs(
    title = "Daily Tidal Cycles (Day vs Hour)",
    x = "Day",
    y = "Hour of Time",
    fill="Tide Patterns"
  ) +
  scale_fill_gradient2(

```

```

    low = "red", mid = "white", high = "blue",
    midpoint = mean(portland_wl$WL, na.rm = TRUE)
  ) + # Divergent color scale with white in the middle
  theme_minimal()

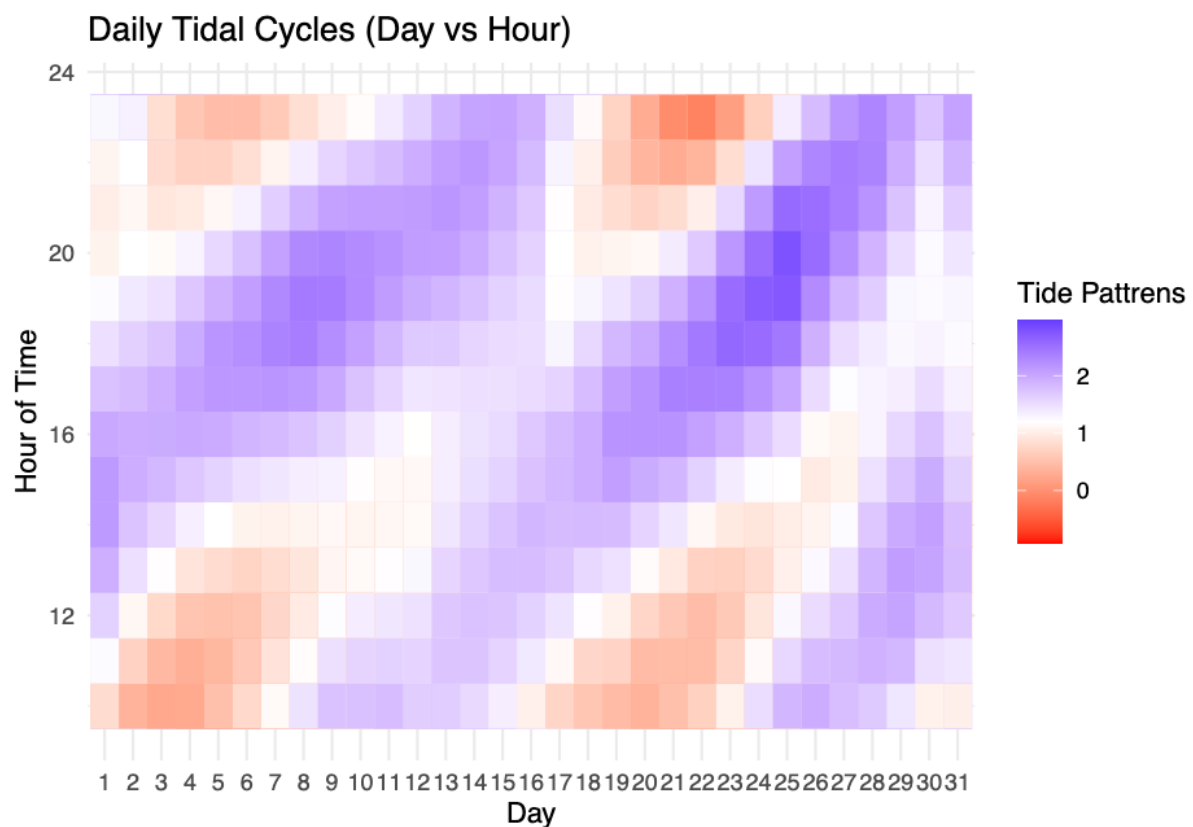
# Print the plot
print(heatmap_plot_custom_divergent)

```

```

## Warning: Removed 3350 rows containing missing values or values outside the scale range
## ('geom_tile()').

```



```

library(cartogram)
library(sf)

```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(tmap)
```

```

## Breaking News: tmap 3.x is retiring. Please test v4, e.g. with
## remotes::install_github('r-tmap/tmap')

```

```
library(tigris)
```

```
## To enable caching of data, set 'options(tigris_use_cache = TRUE)'  
## in your R script or .Rprofile.
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
food_services_data <- read.csv("~/Desktop/DV/HW2/FoodSrvByCounty.csv", header=TRUE)
```

```
# Create FDbyState DataFrame
```

```
FDbyState <- data.frame(  
  state = c("Alabama", "Alaska", "Arizona", "Arkansas", "California",  
            "Colorado", "Connecticut", "Delaware", "District of Columbia",  
            "Florida", "Georgia", "Hawaii", "Idaho", "Illinois",  
            "Indiana", "Iowa", "Kansas", "Kentucky", "Louisiana",  
            "Maine", "Maryland", "Massachusetts", "Michigan",  
            "Minnesota", "Mississippi", "Missouri", "Montana",  
            "Nebraska", "Nevada", "New Hampshire", "New Jersey",  
            "New Mexico", "New York", "North Carolina", "North Dakota",  
            "Ohio", "Oklahoma", "Oregon", "Pennsylvania",  
            "Rhode Island", "South Carolina", "South Dakota",  
            "Tennessee", "Texas", "Utah", "Vermont",  
            "Virginia", "Washington", "West Virginia", "Wisconsin",  
            "Wyoming"),  
  FoodServices_1997 = c(6955, 1763, 9094, 4663, 62629,  
                        10073, 6903, 1605, 1700, 28999,  
                        13829, 3081, 2980, 23984, 11705,  
                        6830, 5677, 6546, 7151, 3716,  
                        9049, 14827, 18958, 9982, 4050,  
                        11150, 3280, 4070, 3633, 3033,  
                        16975, 3827, 38051, 14579, 1827,  
                        22631, 6534, 8371, 24465, 2617,  
                        7775, 2259, 9604, 34160, 3785,  
                        1932, 12343, 13124, 3290, 13253,  
                        1751),  
  FoodServices_2002 = c(7075, 1849, 9944, 4659, 66568,  
                        10799, 7047, 1576, 1799, 30215,  
                        15463, 3138, 3088, 24245, 11788,  
                        6586, 5584, 6660, 7535, 3726,  
                        9406, 15175, 19084, 10232, 4329,
```

```

11280, 3260, 3992, 4252, 3160,
17537, 3756, 39428, 15747, 1765,
22663, 6506, 8816, 24778, 2701,
8135, 2203, 10070, 36591, 4106,
1950, 13305, 13699, 3310, 13268,
1742),
FoodServices_2007 = c(8093, 1996, 11610, 5112, 75989,
12075, 7941, 1850, 2148, 35012,
18640, 3528, 3482, 26774, 12932,
7014, 5866, 7309, 8169, 3938,
10802, 16039, 19678, 11340, 4817,
12261, 3360, 4241, 5570, 3508,
19526, 4090, 43791, 18268, 1840,
23959, 6900, 10241, 26910, 2926,
9291, 2426, 11592, 43509, 4541,
1942, 15765, 15893, 3650, 14439,
1768)
)

# Clean up state names in FDbyState to match the shapefile's state names
FDbyState$state <- tolower(FDbyState$state)

states <- states(cb = TRUE)

## Retrieving data for the year 2021

## |

states$NAME <- tolower(states$NAME)

merged_data <- states %>%
  left_join(FDbyState, by = c("NAME" = "state"))

merged_data[is.na(merged_data)] <- 0

# Transform the projection (to Mercator projection, for example)
merged_data <- st_transform(merged_data, 3395)

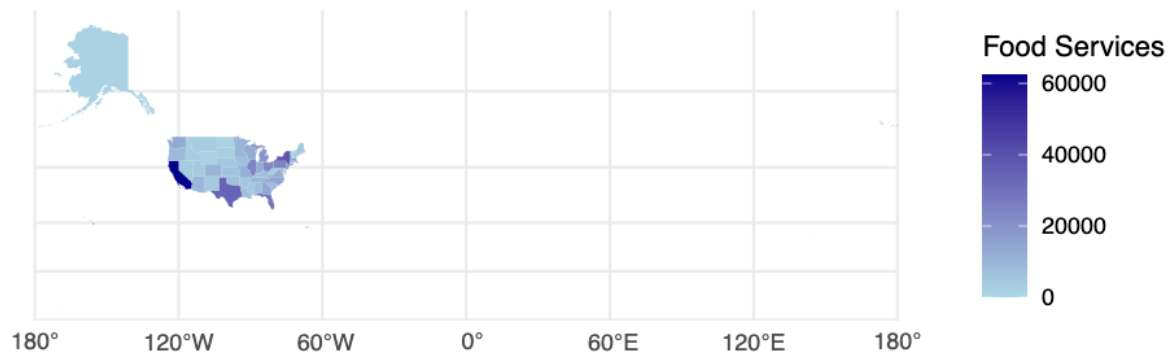
# Construct the cartogram based on a food services variable (e.g., FoodServices-2007)
merged_data <- st_make_valid(merged_data)
merged_data <- st_simplify(merged_data, dTolerance = 0.01)

# Create cartogram
cartogram_data1 <- cartogram_cont(merged_data, "FoodServices_1997", itermax=1)
cartogram_data2 <- cartogram_cont(merged_data, "FoodServices_2002", itermax=1)
cartogram_data3 <- cartogram_cont(merged_data, "FoodServices_2007", itermax=1)

# Plot the cartogram
ggplot(cartogram_data1) +
  geom_sf(aes(fill = FoodServices_1997), color = NA) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "Cartogram of Food Services by State (1997)", fill = "Food Services") + theme_minimal()

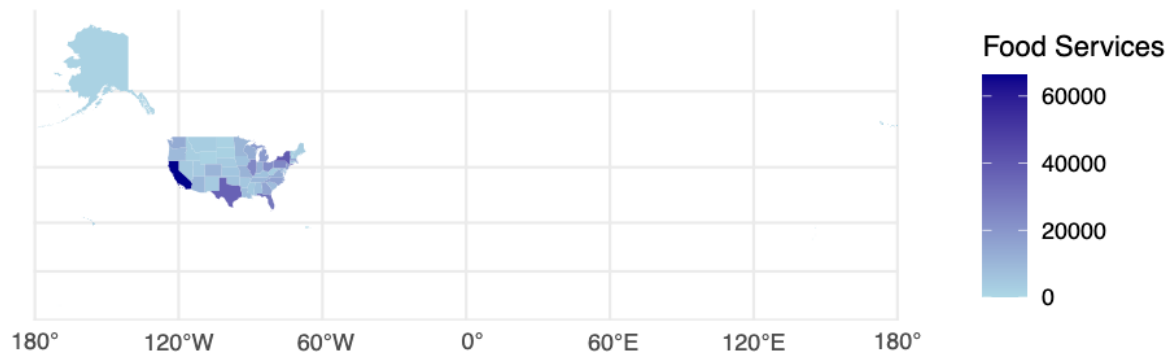
```

Cartogram of Food Services by State (1997)



```
ggplot(cartogram_data2) +  
  geom_sf(aes(fill = FoodServices_2002), color = NA) +  
  scale_fill_gradient(low = "lightblue", high = "darkblue") +  
  labs(title = "Cartogram of Food Services by State (2002)", fill = "Food Services") + theme_minimal()
```

Cartogram of Food Services by State (2002)



```
ggplot(cartogram_data3) +
  geom_sf(aes(fill = FoodServices_2007), color = NA) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "Cartogram of Food Services by State (2007)", fill = "Food Services") + theme_minimal()
```



Cartogram of Food Services by State (2007)

