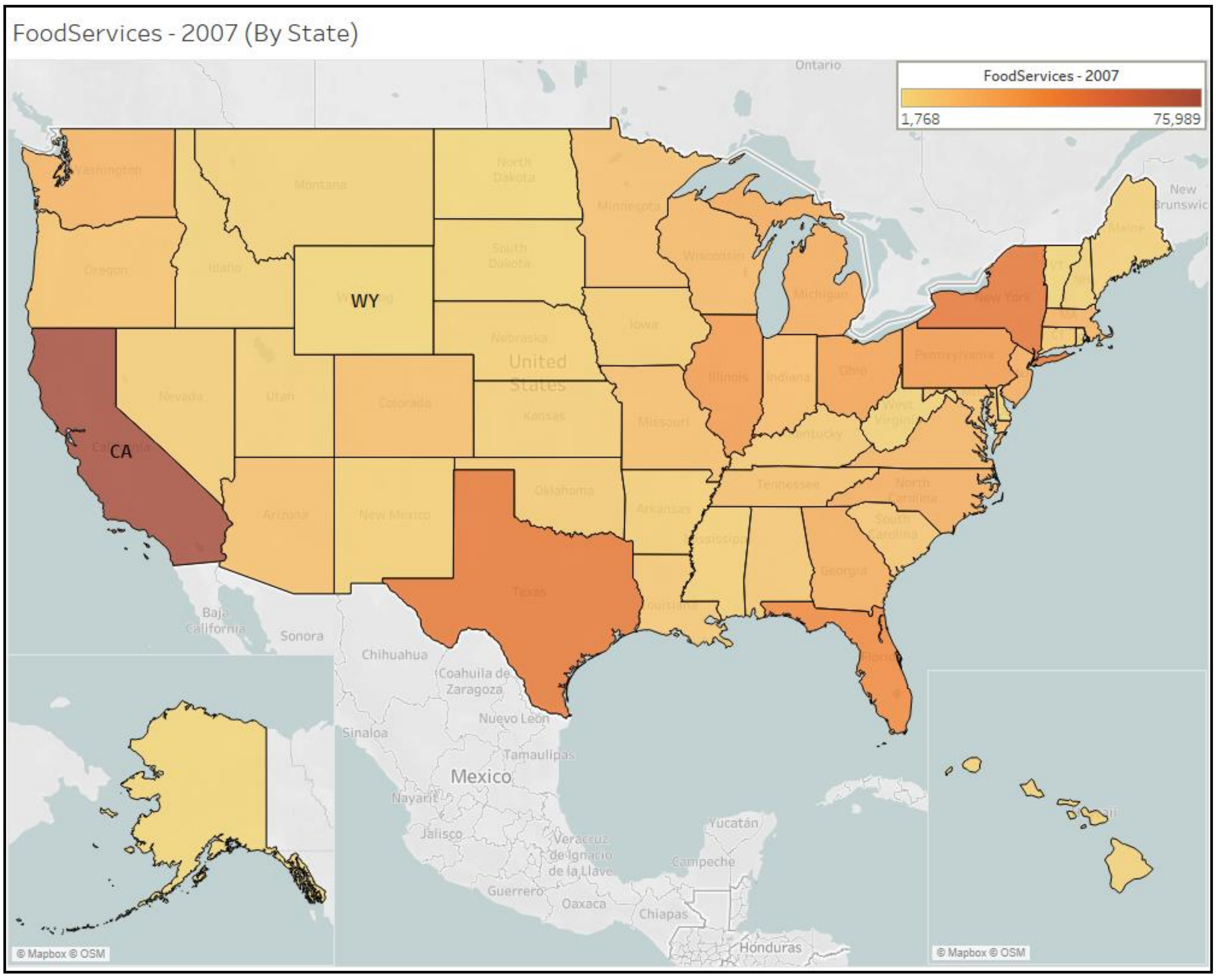
(Tableau)

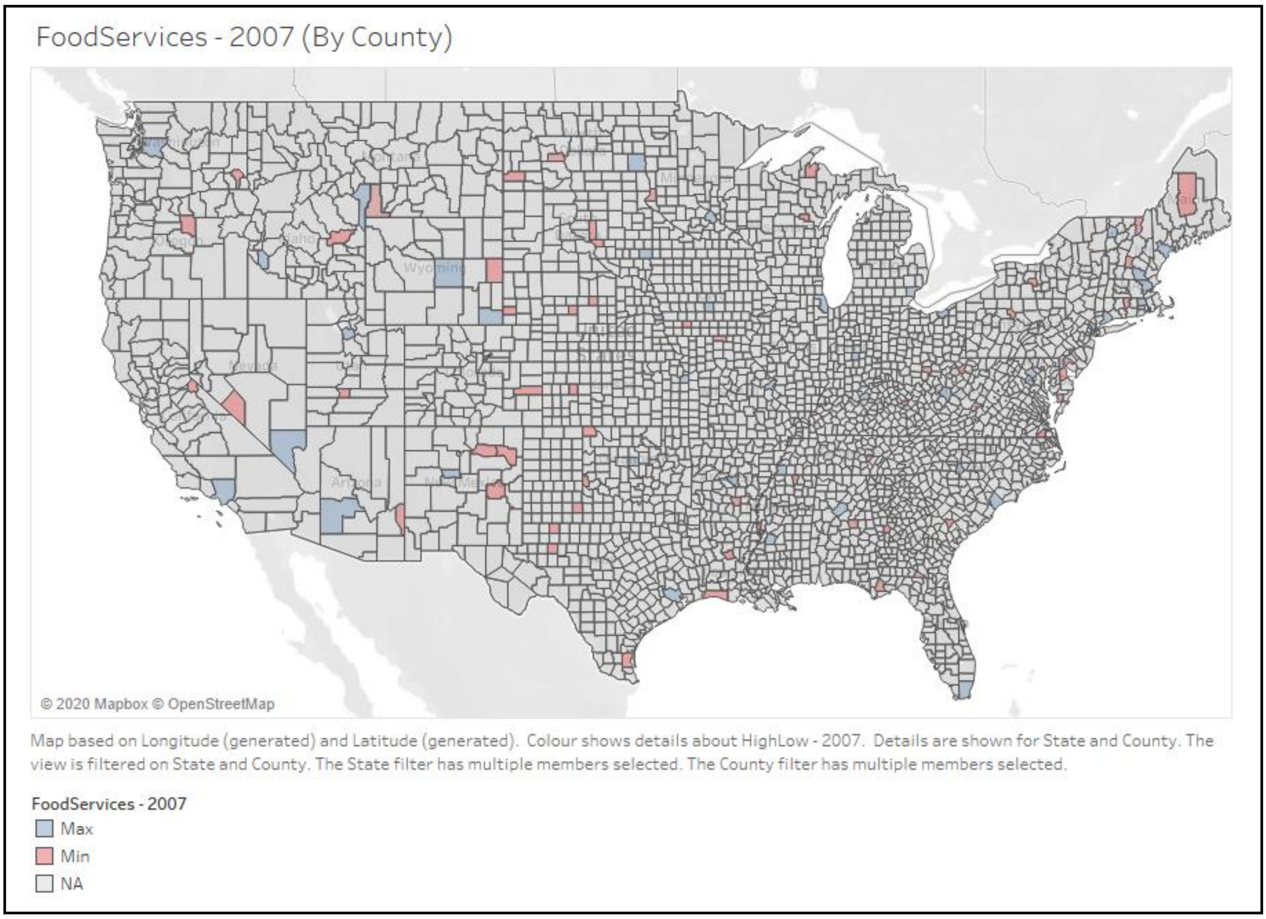
Food services by state



Conclusion: The above United States map shows the Food Services by State for 2007.
The colour scheme highlights the changes in levels. The light orange represents the minimum level while dark gold represents maximum level. Text labels are assigned to minimum and maximum value.
As can be seen in the map, CA has the maximum food services for 2007 while WY has the minimum food services for 2007.

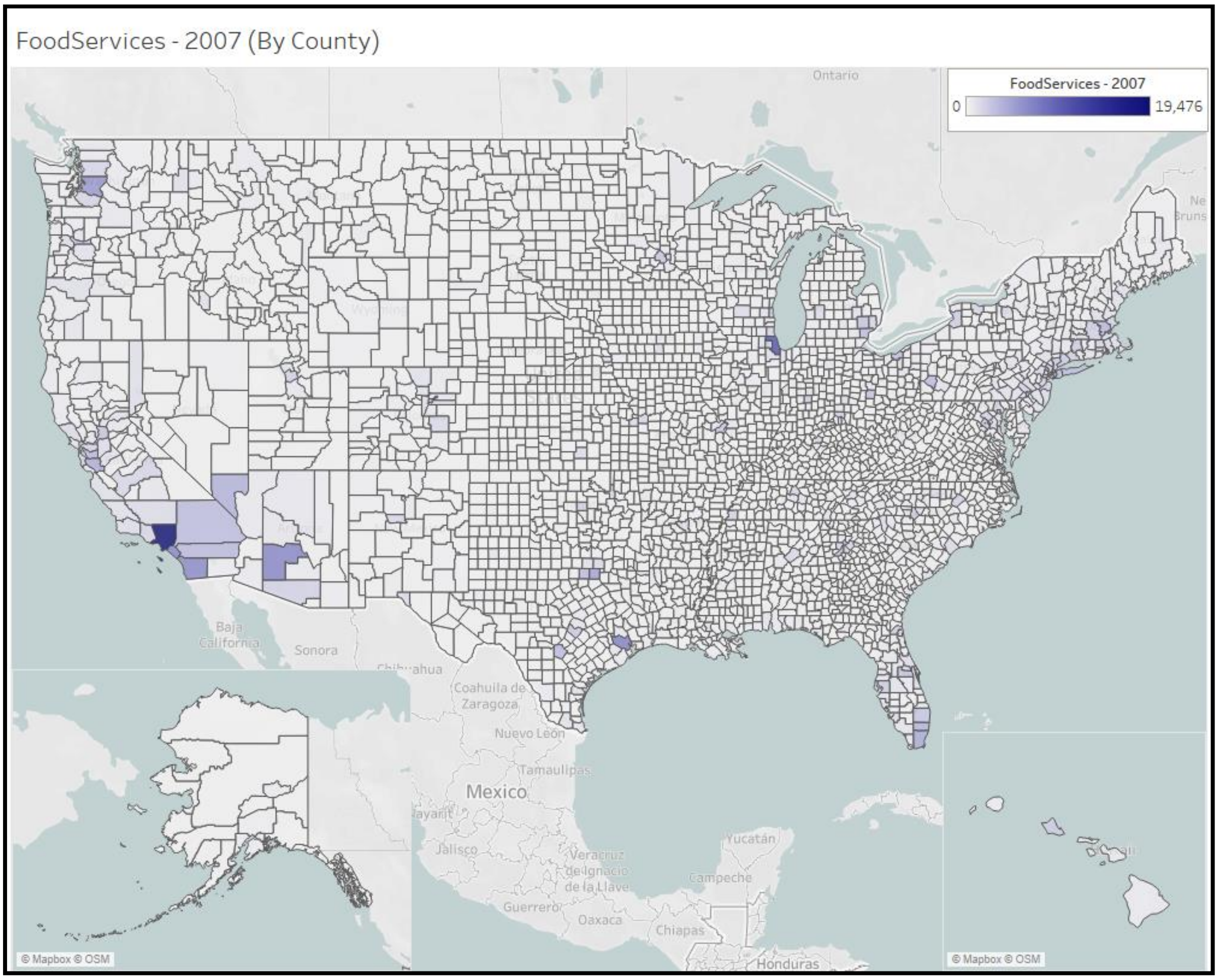
(Tableau)

Food services by county (Min Max in each state)



Conclusion: The above United States map shows the Food Services by County for 2007.
The colour scheme highlights the maximum and minimum level on a county level in each state. The blue colour represents the maximum level while the red colour represents the minimum level.
Example: Los Angeles County in CA has the maximum food services for 2007 while Alpine County in CA has the minimum food services for 2007.

Food services by county (Levels in each county)

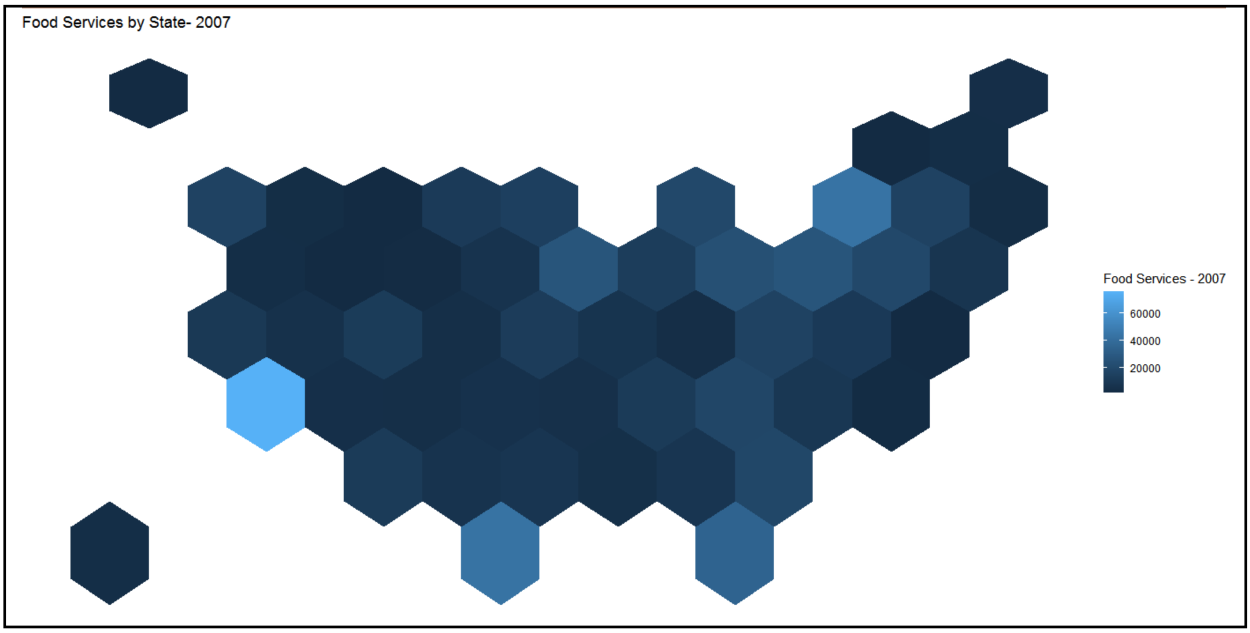


Conclusion: Another version of the United States map shows the Food Services by County for 2007. The colour scheme shows the gradual increase on food level on a county basis in each state. As the food level increases, the saturation of the colour increases.

(RStudio)

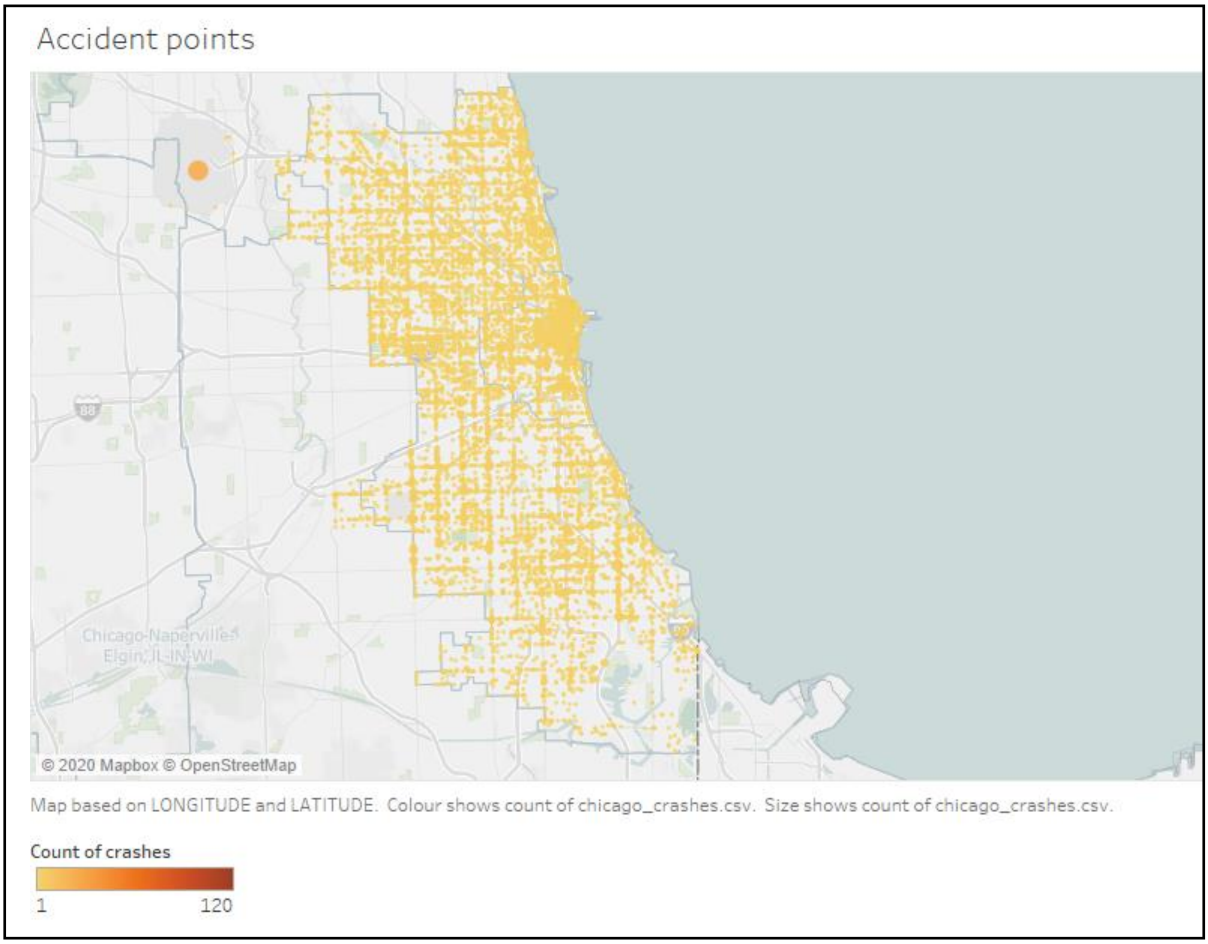
Food services by state

```
> spdf <- geojson_read("us_states_hexgrid.geojson",  what = "sp")

> # Bit of reformating
> spdf@data = spdf@data %>%
+   mutate(google_name = gsub(" \\(United States\\)", "", google_name))

> spdf@data = spdf@data %>% mutate(google_name = gsub(" \\(United States\\
)", "", google_name))

> spdf_fortified <- tidy(spdf, region = "google_name")

> spdf_fortified_name <- tidy(spdf, region = "google_name")

> FoodSrvcByCounty$county = as.character(FoodSrvcByCounty$County)

> spdf_fortified$county = toupper(spdf_fortified$id)

> spdf_fortified <- spdf_fortified %>%
+   left_join(. , FoodSrvcByCounty, by = c("county" = "county"))

> head(spdf_fortified)
# A tibble: 6 x 13
   long   lat order hole  piece group id    county County State
  <dbl> <dbl> <int> <lgl> <fct> <chr> <chr> <chr>  <fct>  <fct>
1 -94.4  35.8     1 FALSE 1     Alab~ Alab~ ALABA~ ALABA~ ""
2 -91.7  34.5     2 FALSE 1     Alab~ Alab~ ALABA~ ALABA~ ""
3 -91.7  31.9     3 FALSE 1     Alab~ Alab~ ALABA~ ALABA~ ""
4 -94.4  30.5     4 FALSE 1     Alab~ Alab~ ALABA~ ALABA~ ""
5 -97.1  31.9     5 FALSE 1     Alab~ Alab~ ALABA~ ALABA~ ""
6 -97.1  34.5     6 FALSE 1     Alab~ Alab~ ALABA~ ALABA~ ""
# ... with 3 more variables: FoodServices.97 <int>, FoodServices.2002 <int
>,
#   FoodServices.2007 <int>

> ggplot() +
+   geom_polygon(data = spdf_fortified,
+                aes(
+                  x = long,
+                  y = lat,
+                  group = group,
+                  fill =  FoodServices.2007
+                )) +
+   theme_void() + labs(fill = 'Food Services - 2007',
+                       title = "Food Services by State- 2007")
```



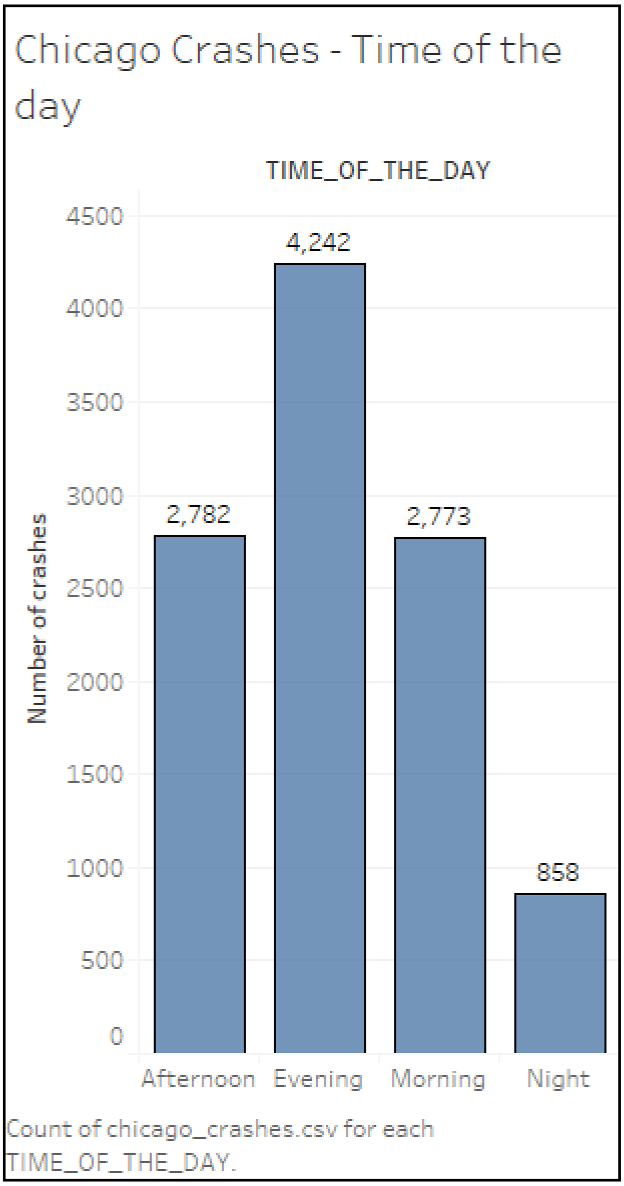
Conclusion: The above United States map shows the Food Services by State for 2007.
The colour scheme highlights the changes in levels. The light blue represents the maximum level while light blue represents maximum level.

(Tableau)



Conclusion: The above map for Chicago state shows the crashes in Chicago.
The dots represent the accident points in June 2019.
As you can see on the map majority of the crashes are in the centre of Chicago city.

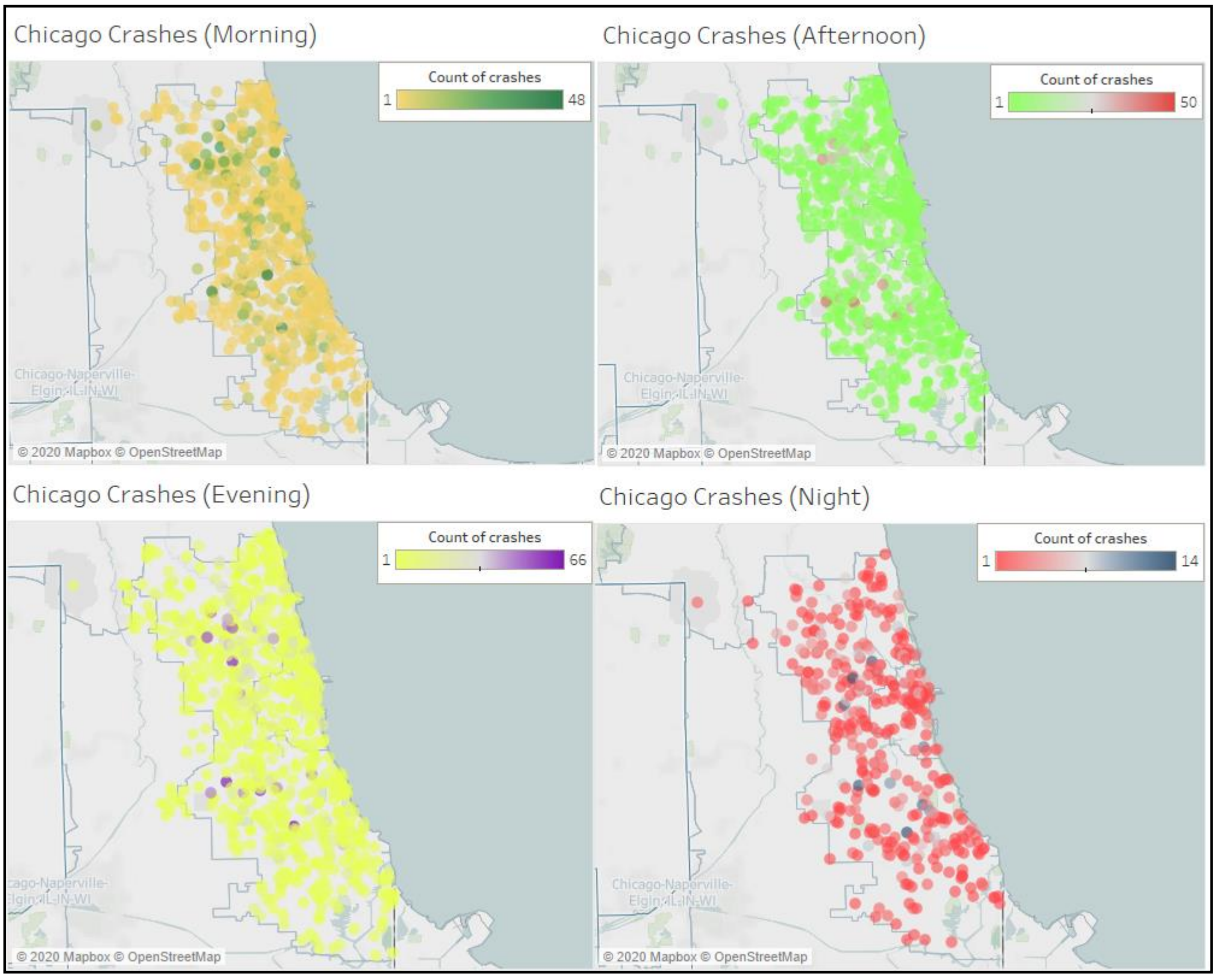
(Tableau)



The above bar chart shows us the number of crashes in Chicago city in June 2019.
The time of the day is calculated field based on the time throughout the day.
Following is the classification to determining the time of the day.

- Morning: 5:00 hours - 11:59 hours
- Afternoon: 12:00 hours – 15:59 hours
- Evening: 16:00 hours – 23:59 hours
- Night: 00:00 hours – 4:49 hours

As can be seen in the graph, there are minimum crashes during the night while maximum number of crashes in the evening. We have almost equal number of crashes in the afternoon and morning.



Conclusion: The above map for Chicago state shows the crashes in Chicago during the different time of the day.

The dots represent the accident points in a specific direction and street over the month of June 2019.

The top left geo plot shows the crashes in the morning. The peak numbers of crashes are all over the city.

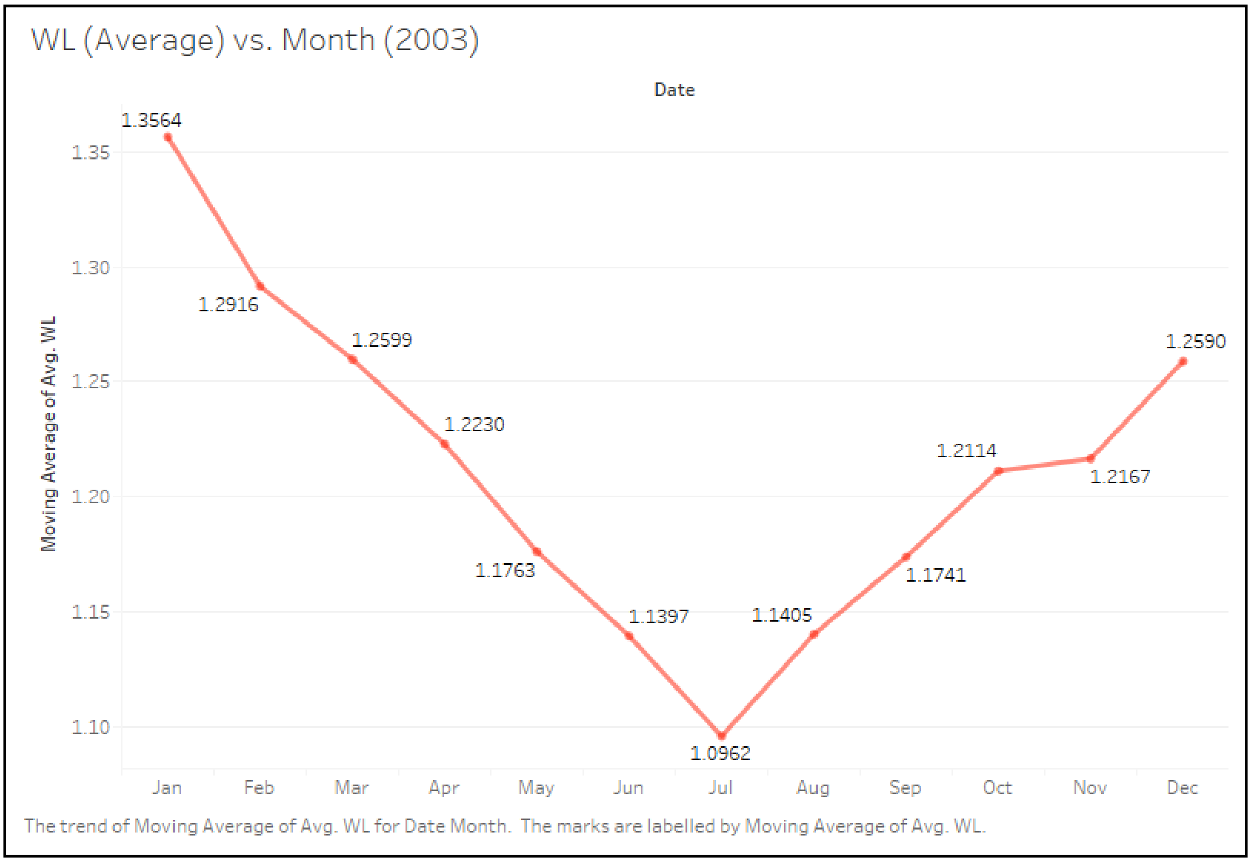
We see a similar geo plot (top right) in the afternoon except the peaks are at few points in the inner parts of the city.

There are maximum numbers of crashes in the evening (bottom left). We see the peak points in the inner parts of the city.

There are minimum numbers of crashes in the night. The maximum crash count during the night is 14 which is almost 1/3 of the crashes across all the plots.

(Tableau)

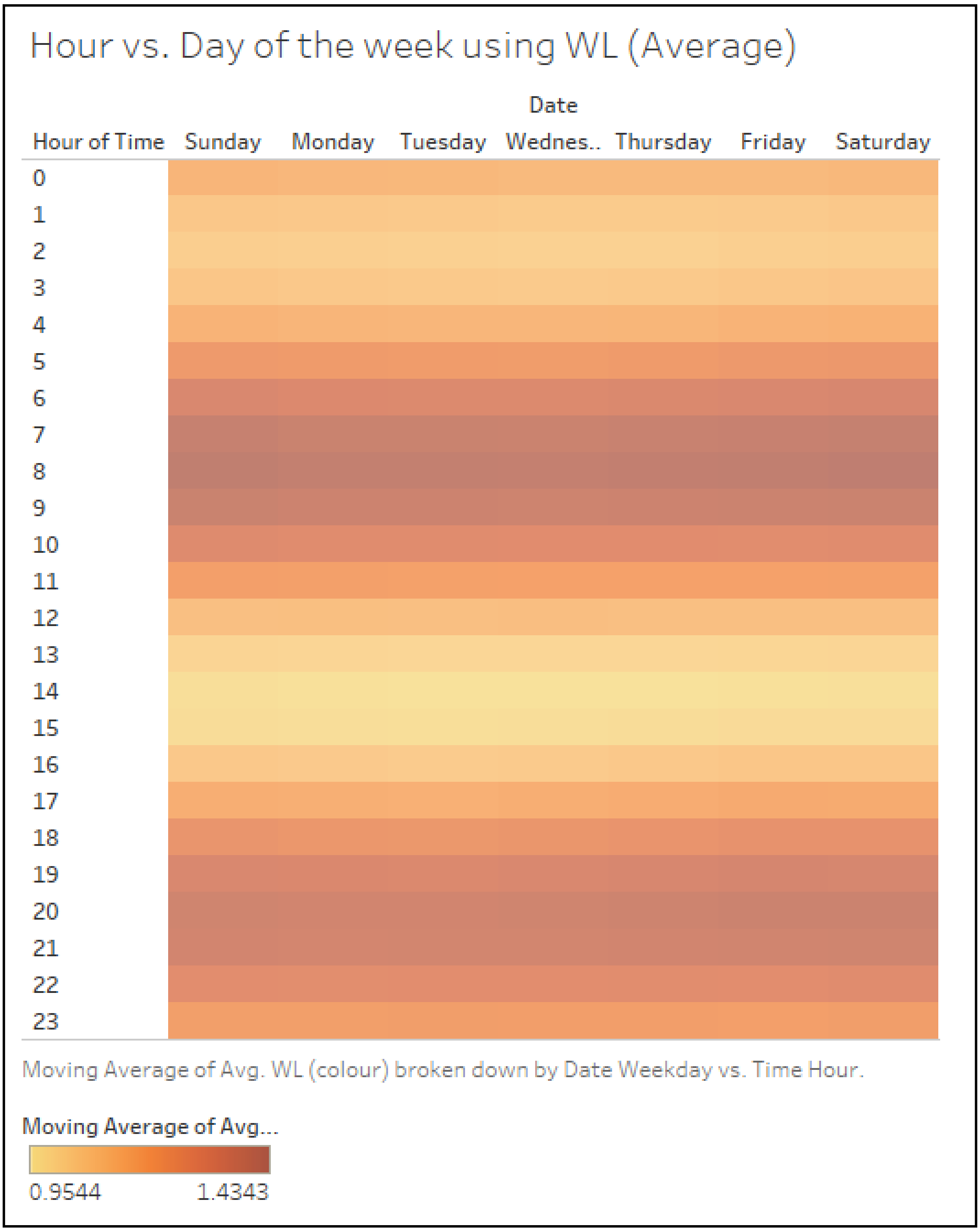
Average WL vs. Month (2003)



Conclusion: The above line graph plots the average value of WL for month on month basis.
The graph shows the maximum peak value at the start of the year while the minimum value during the mid year in the month of July.

(Tableau)

Hour vs. Day of the week using Average WL - (2003)



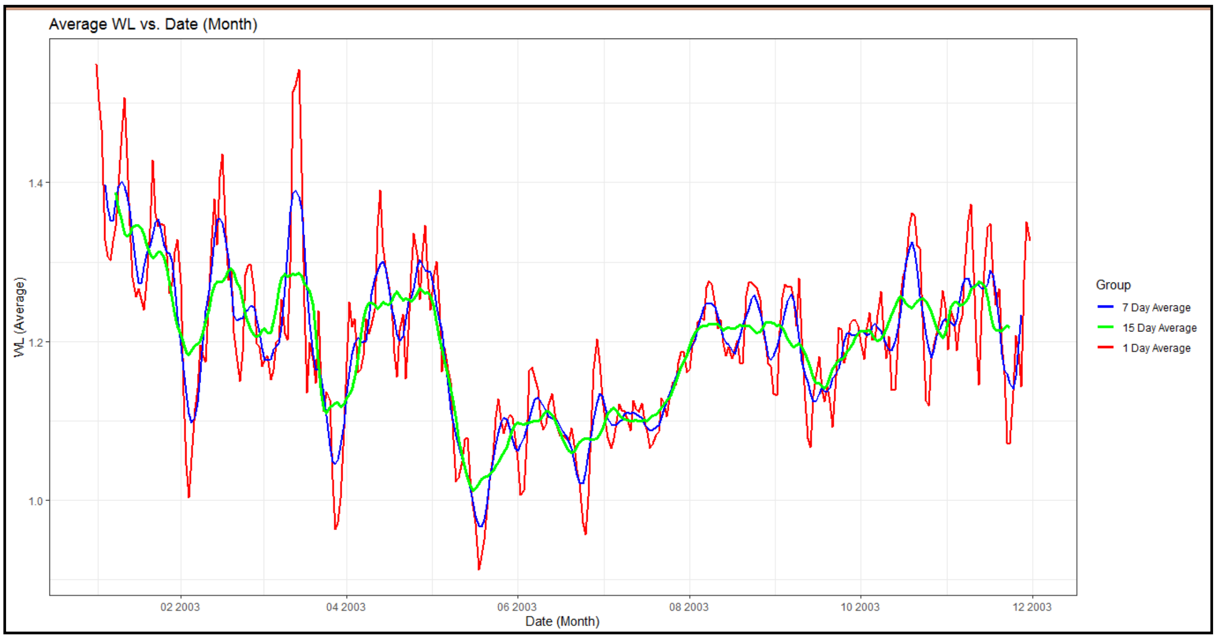
Conclusion: The above heat map plots the hour of time vs day of the week using the average WL value.
The brown tint represents the maximum WL value while the yellowish tint represents the minimum WL value.
The heat map shows a maximum value between 5:00 AM to 10:00 AMD and 5:00 PM to 11:00 PM daily.

(Rstudio)

Average WL vs. Month (2003)

```
> PortlandWaterLevel2003$date <- as_date(as.character(PortlandWaterLevel20
03$Date),format="%m/%d/%Y")

> WaterMA <- PortlandWaterLevel2003 %>%
+   select(date, WLValue = WL) %>%
+   mutate(WLValue_01 = ma(WLValue, order = 24, centre = TRUE),
+          WLValue_02 = ma(WLValue, order = 168, centre = TRUE),
+          WLValue_03 = ma(WLValue, order = 360, centre = TRUE),
+          WLValue_04 = ma(WLValue, order = 720, centre = TRUE),
+          WLValue_05 = ma(WLValue, order = 1080, centre = TRUE))

> head(WaterMA,10)
         date WLValue WLValue_01 WLValue_02 WLValue_03 WLValue_04 WLValue_
05
1  2003-01-01  -0.226         NA         NA         NA         NA
NA
2  2003-01-01  -0.115         NA         NA         NA         NA
NA
3  2003-01-01   0.213         NA         NA         NA         NA
NA
4  2003-01-01   0.741         NA         NA         NA         NA
NA
5  2003-01-01   1.190         NA         NA         NA         NA
NA
6  2003-01-01   1.686         NA         NA         NA         NA
NA
7  2003-01-01   1.942         NA         NA         NA         NA
NA
8  2003-01-01   2.071         NA         NA         NA         NA
NA
9  2003-01-01   1.964         NA         NA         NA         NA
NA
10 2003-01-01   1.718         NA         NA         NA         NA
NA

> WL1DayAvg <- WaterMA %>%
+   group_by(date) %>%
+   summarize(avg_by_date =  sum(WLValue_01)/length(WLValue_01)) %>%
+   arrange((date))

> head(WL1DayAvg)
# A tibble: 6 x 2
  date       avg_by_date
  <date>           <dbl>
1 2003-01-01       NA
2 2003-01-02         1.55
3 2003-01-03         1.50
4 2003-01-04         1.47
5 2003-01-05         1.33
6 2003-01-06         1.31

> WL1WeekAvg <- WaterMA %>%
+   group_by(date) %>%
+   summarize(avg_by_date =  sum(WLValue_02)/length(WLValue_02)) %>%
+   arrange((date))

> head(WL1WeekAvg)
# A tibble: 6 x 2
  date       avg_by_date
  <date>           <dbl>
1 2003-01-01       NA
2 2003-01-02       NA
3 2003-01-03       NA
4 2003-01-04       NA
```

```
5 2003-01-05         1.40
6 2003-01-06         1.37

> WL15DayAVG <- WaterMA %>%
+   group_by(date) %>%
+   summarize(avg_by_date =  sum(WLValue_03)/length(WLValue_03)) %>%
+   arrange((date))

> head(WL15DayAVG,10)
# A tibble: 10 x 2
   date       avg_by_date
   <date>           <dbl>
 1 2003-01-01       NA
 2 2003-01-02       NA
 3 2003-01-03       NA
 4 2003-01-04       NA
 5 2003-01-05       NA
 6 2003-01-06       NA
 7 2003-01-07       NA
 8 2003-01-08       NA
 9 2003-01-09        1.39
10 2003-01-10        1.37


> timeplot <-ggplot() +
+   geom_line(na.omit(WL1DayAvg), mapping =aes(x=date, y=avg_by_date), col
or = "red",size =0.8) +
+   geom_line(na.omit(WL1WeekAvg), mapping =aes(x=date, y=avg_by_date), co
lor = "blue",size =1) +
+   geom_line(na.omit(WL15DayAVG), mapping =aes(x=date, y=avg_by_date), co
lor = "green",size =1.3) +
+   theme_bw() +
+   scale_x_date(breaks = as.Date(c("2003-01-01", "2003-12-01")),
+                date_breaks = "2 months",
+                 date_labels = "%m %Y")

> timeplot + labs(x = "Date (Month)",
+                 y = "WL (Average)",
+                 title = "Average WL vs. Date (Month)") +
+   scale_color_manual(name = "Group",
+                      values = c(green = "green", red = "red", blue = "bl
ue"),
+                      labels = c("7 Day Average", "15 Day Average", "1 Da
y Average"))
```



Conclusion: The above line graph plots the average value of WL for month on month basis.
This particular line graph groups all the values based on date and produces a single record for single date.
The red line represents 1 day average value while the blue and green line represents the 7 day and 15 day average value respectively.
The line graphs are overlapping each other showing a smooth curve based on the change in window.

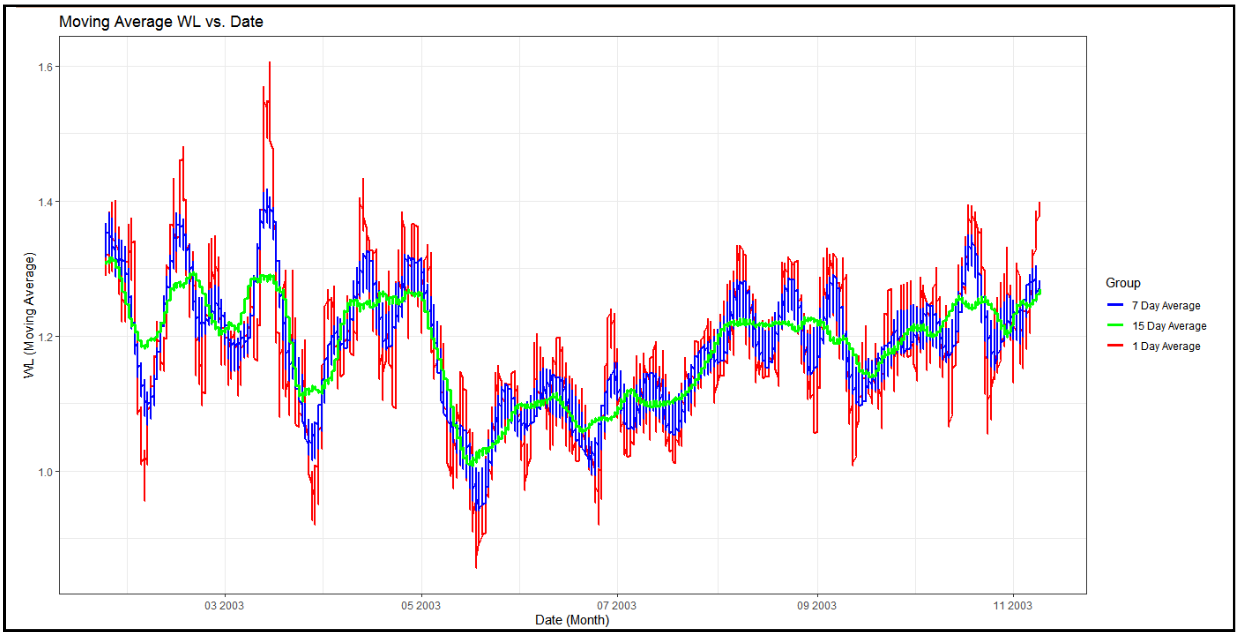
(Rstudio)

Moving Average WL vs. Month (2003)

```
> WaterMA <- PortlandWaterLevel2003 %>%
+   select(date, WLValue = WL) %>%
+   mutate(WLValue_01 = ma(WLValue, order = 24, centre = TRUE),
+          WLValue_02 = ma(WLValue, order = 168, centre = TRUE),
+          WLValue_03 = ma(WLValue, order = 360, centre = TRUE),
+          WLValue_04 = ma(WLValue, order = 720, centre = TRUE),
+          WLValue_05 = ma(WLValue, order = 1080, centre = TRUE))
```

```
> head(WaterMA,10)
         date WLValue WLValue_01 WLValue_02 WLValue_03 WLValue_04 WLValue_
05
1  2003-01-01  -0.226          NA          NA          NA          NA
NA
2  2003-01-01  -0.115          NA          NA          NA          NA
NA
3  2003-01-01   0.213          NA          NA          NA          NA
NA
4  2003-01-01   0.741          NA          NA          NA          NA
NA
5  2003-01-01   1.190          NA          NA          NA          NA
NA
6  2003-01-01   1.686          NA          NA          NA          NA
NA
7  2003-01-01   1.942          NA          NA          NA          NA
NA
8  2003-01-01   2.071          NA          NA          NA          NA
NA
9  2003-01-01   1.964          NA          NA          NA          NA
NA
```

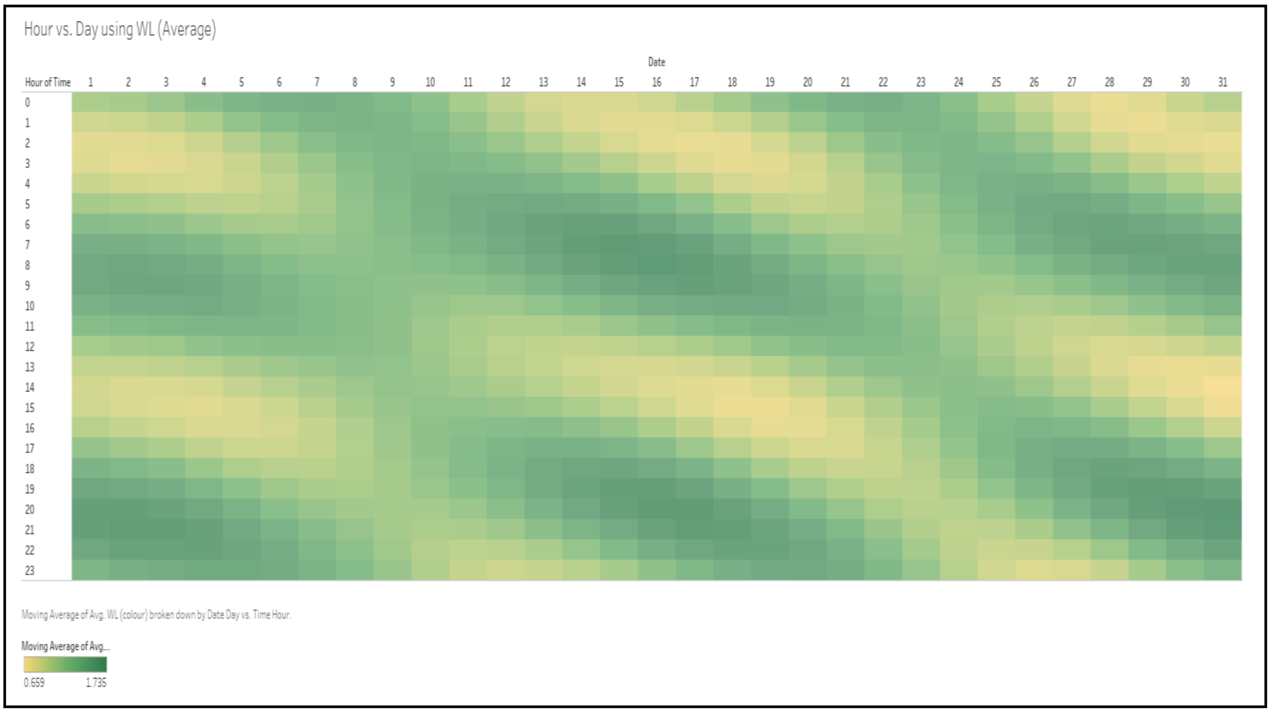
```
10 2003-01-01    1.718          NA          NA          NA          NA
NA

> timeplotma <- ggplot() +
+   geom_line(na.omit(WaterMA), mapping =aes(x=date, y=WLValue_01, color =
"red"), show.legend=TRUE,size =0.8) +
+   geom_line(na.omit(WaterMA), mapping =aes(x=date, y=WLValue_02, color =
"blue"), show.legend=TRUE,size =1) +
+   geom_line(na.omit(WaterMA), mapping =aes(x=date, y=WLValue_03, color =
"green"), show.legend=TRUE,size =1.3) +
+   theme_bw() +
+   scale_x_date(breaks = as.Date(c("2003-01-01", "2003-12-01")),
+                date_breaks = "2 months",
+                date_labels = "%m %Y")

> timeplotma + labs(x = "Date (Month)",
+                   y = "WL (Moving Average)",
+                   title = "Moving Average WL vs. Date") +
+   scale_color_manual(name = "Group",
+                      values = c(green = "green", red = "red", blue = "bl
ue"),
+                      labels = c("7 Day Average", "15 Day Average", "1 Da
y Average"))
```



Conclusion: The above line graph plots the average value of WL for month on month basis.
This particular line graph consists of many records for a single date on a hourly basis.
The line graph is directly created using these points.
The red line represents 1 day average value while the blue and green line represents the 7 day and 15 day average value respectively.
The line graphs are overlapping each other showing a smooth curve based on the change in window.

(Tableau)

Hour vs. Day using Average WL - (2003)



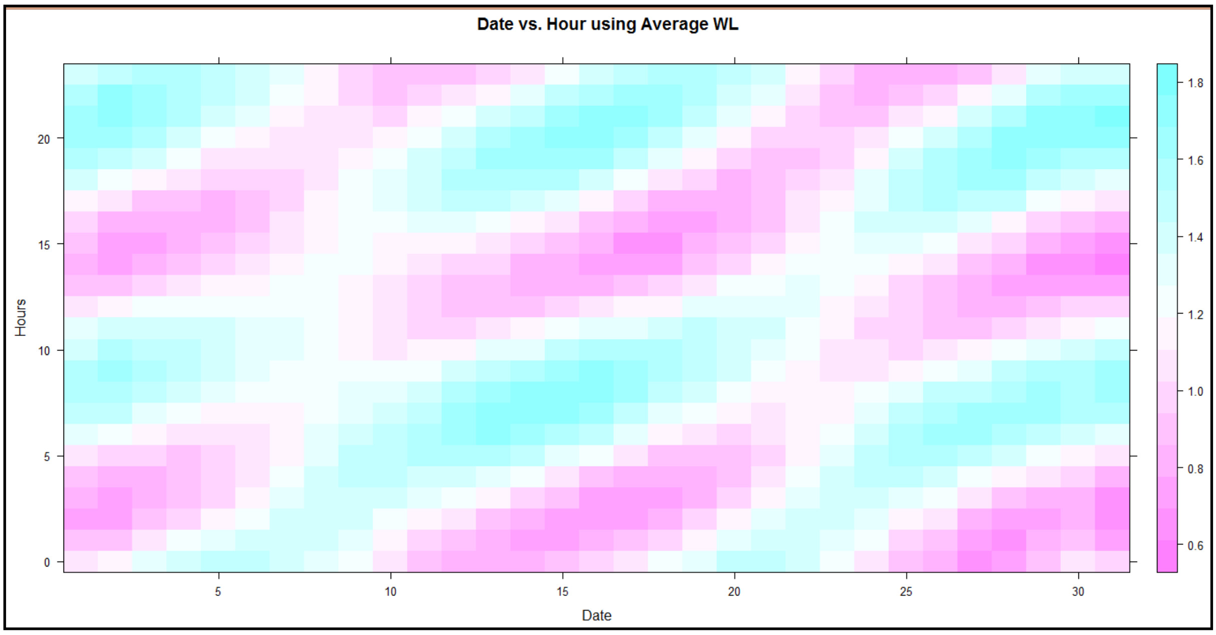
Conclusion: The above heat map plots the hour of time vs date using the average WL value.
The green tint represents the maximum WL value while the yellowish tint represents the minimum WL value.
The heat map shows a wave like pattern on an hourly and daily basis.

(Rstudio)

Hour vs. Day using Average WL - (2003)

```
> levelplot(as.numeric(avg_by_dd) ~ as.numeric(dd)*as.numeric(hh), data=WL
_dd_AVG  ,xlab="Date",
+            ylab = "Hours",
+            region = TRUE,
+            main = "Date vs. Hour using Average WL")
```

```
> PortlandWaterLevel <- PortlandWaterLevel2003 %>%
+   mutate(dd = format(as_date(date),"%d"),
+          hh= format(as.POSIXct(Time,format="%H:%M"),"%H"),
+          wd = wday(date),
+          wday = wday(date, label = TRUE, abbr = TRUE))

> WL_dd_AVG <- PortlandWaterLevel %>%
+   group_by(dd,hh) %>%
+   summarize(avg_by_dd =  sum(WL)/length(WL)) %>%
+   arrange(dd,hh)
```

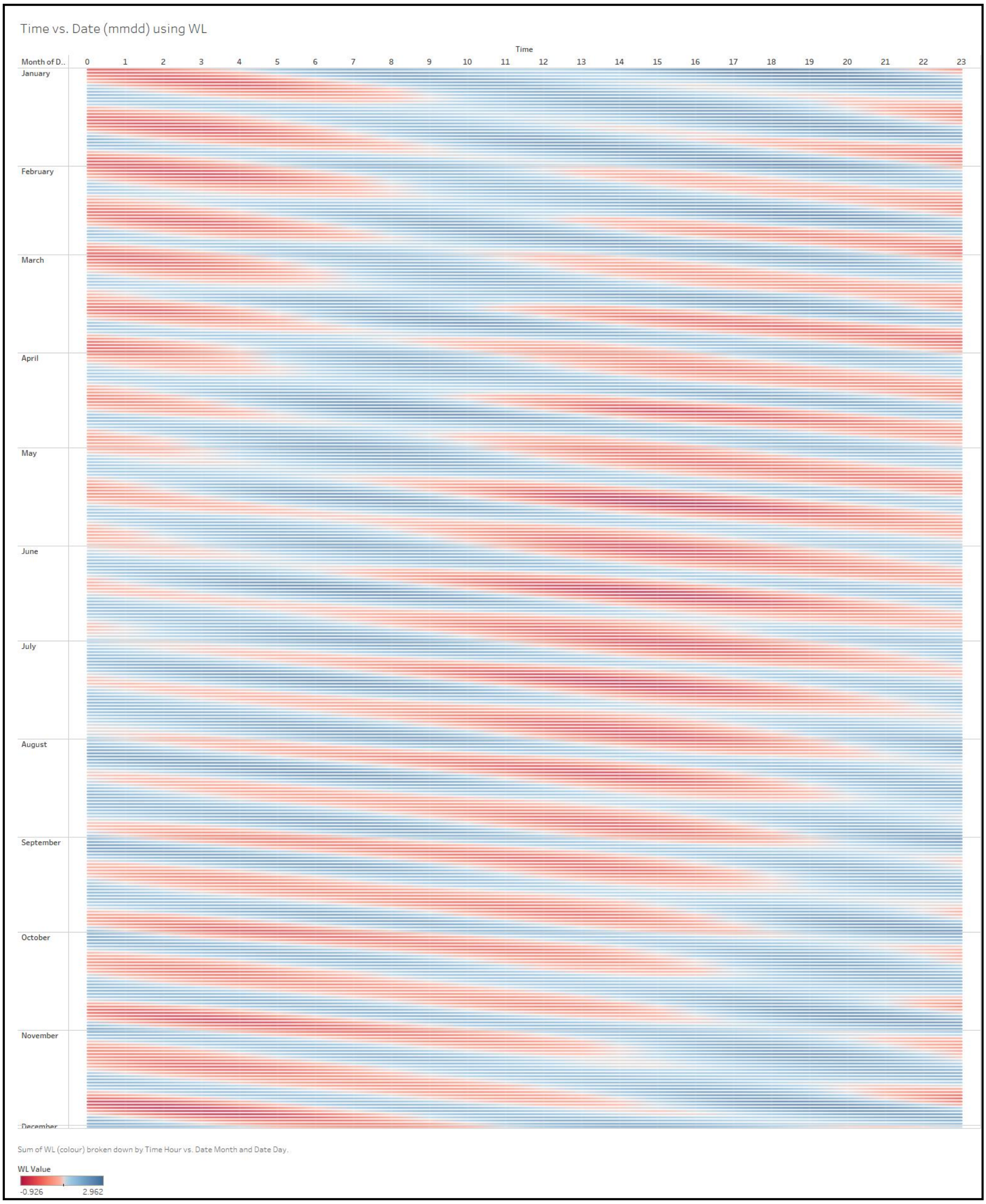


Conclusion: The above level plot the hour of time vs date using the average WL value.
The blue tint represents the maximum WL value while the pink tint represents the minimum WL value.
The level plot shows a wave like pattern on an hourly and daily basis.

Section b)

(Tableau)

Time vs. Date (mmdd) using WL - (2003)



Conclusion: The above line graph plots the value of WL for month and date basis.
The graph shows a wave like pattern over the year.

Extra credit:

Tableau by default assigns 0 has the midpoint value. Considering the data distribution tableau highlights the value below 0 with red colour while the values above 0 are assigned the blue colour. By updating the centre value for better visualization, we are changing the data distribution.
The distribution based on colour follows a power law. The centre values are updated to show a pattern.

Section c)

The various plot presented over Portland water data provides a different picture on a weekly and monthly basis by plotting the data standalone or by calculating the moving average for the date or day of the week or for a combination of date and month.

In order, I have plotted the moving average value for WL based on the day of the week and the hours. This graph shows the peak value during the certain hours of the day.

I have tried plotting the line graph for the moving average for 1 day, 1 week and fortnight basis. The line graphs are overlapping each other showing a smooth curve based on the change in window.

In tableau, I have used the moving average WL value to draw a heatmap based on the date of the month.
The above graph has been replicated in R studio.

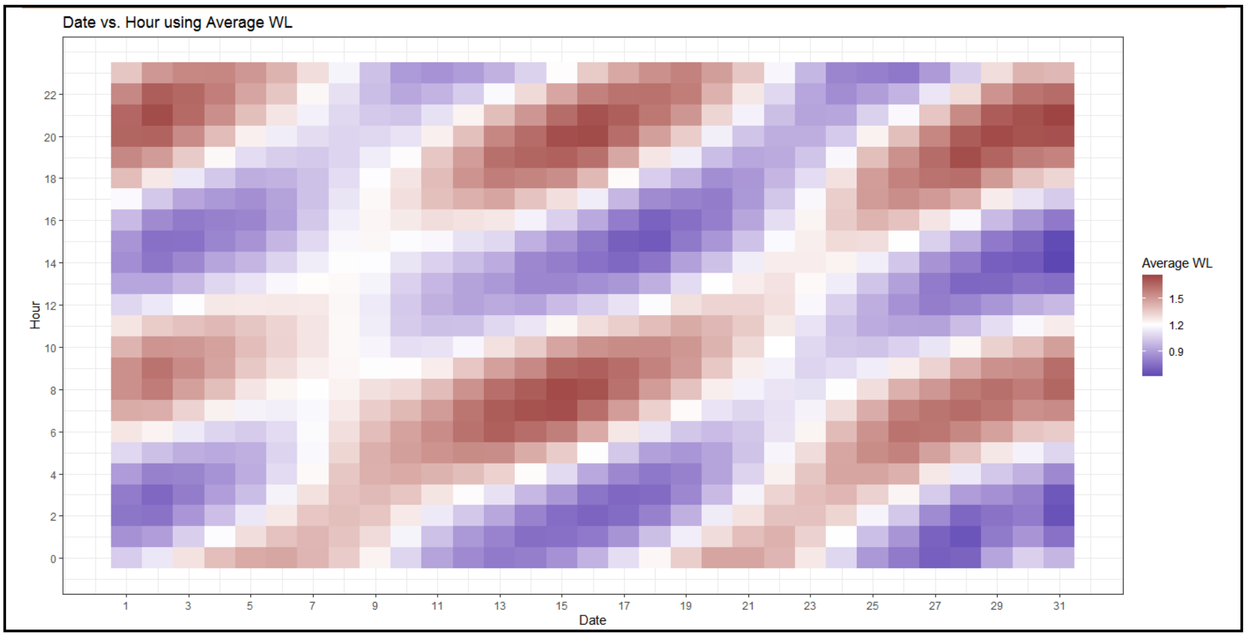
In tableau, I have plotted the line graph to show a wave like pattern for each day of the year.
The plot shows the value of WL increases towards the end of the month.

Problem 4

(Rstudio)

Hour vs. Day using Average WL - (2003)

```
> PortlandWaterLevel <- PortlandWaterLevel2003 %>%
+   mutate(
+     dd = format(as_date(date), "%d"),
+     hh = format(as.POSIXct(Time, format = "%H:%M"), "%H"),
+     wd = wday(date),
+     wday = wday(date, label = TRUE, abbr = TRUE)
+   )

> WL_dd_AVG <- PortlandWaterLevel %>%
+   group_by(dd, hh) %>%
+   summarize(avg_by_dd =  sum(WL) / length(WL)) %>%
+   arrange(dd, hh)

> heatmap <-
+   ggplot(WL_dd_AVG, aes(as.numeric(dd), as.numeric(hh), fill = avg_by_dd
)) + geom_tile() +  scale_fill_gradient2(
+     low = hsv(0.7,0.6,0.7),
+     mid = "white",
+     high = hsv(1,0.6,0.6),
+     midpoint = 1.2,
+     space = "Lab"
+   ) + scale_x_continuous(breaks = seq(01, 31, 2)) + scale_y_continuous(b
reaks = seq(00, 23, 2)) + theme_bw() +
+   labs(x = "Date",
+        y = "Hour",
+        title = "Date vs. Hour using Average WL",
+        fill = 'Average WL')

> heatmap
```



Conclusion: The above level plot the hour of time vs date using the average WL value.
The blue tint represents the minimum WL value while the red tint represents the maximum WL value.
The level plot shows a wave like pattern on an hourly and daily basis.