Crime in Chennai - R Notebook

Jeffrey Strickland

2/23/2022

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

## Introduction

# Objectives

## Install R libraries

if(!require(readr)) install.packages("readr")  
if(!require(dplyr)) install.packages("dplyr")  
if(!require(DT)) install.packages("DT")  
if(!require(ggrepel)) install.packages("ggrepel")  
if(!require(leaflet)) install.packages("leaflet")

# the Crime Data

## About the Data

* IncidntNum (T) Incident number
* Category (T) Crime category, i.e., larceny/theft
* Descript (T)
* DayOfWeek (T)
* Date (D Date: DD/MM/YYYY
* Time (T) Time: 24-hour system
* PdDistrict (T) Police district where incident occured
* Resolution (T) Resolution of the crime
* Address (T) Address of the crime
* X (N) Longitude
* Y (N) Latitude
* Location (T) Lat/long
* PdId (N) Police Department ID

## Read the data

### Load the data using readr and read\_csv().

# path <- "https://github.com/stricje1/VIT\_University/blob/master/Crime\_Analysis\_Mapping/data/Chennai\_crime\_4yr.zip"  
path <- "C:\\Users\\jeff\\Documents\\Books\\Crime Analysis\\India\_data\\chennai\_crimes.csv"  
df <- read\_csv(path)

## Display Data

### Display the data using DT and datatable().

library(DT)  
df\_sub <- df[1:100,] # display the first 100 rows  
df\_sub$Time <- as.character(df\_sub$Time)   
datatable(df\_sub, options = list(pageLength = 5,scrollX='400px'))

sprintf("Number of Rows in Dataframe: %s", format(nrow(df),big.mark = ","))

## [1] "Number of Rows in Dataframe: 499,365"

## Preprocess Data

# The All-Caps text is difficult to read. Let’s force the text in the appropriate columns into proper case.

proper\_case <- function(x) {  
 return (gsub("\\b([A-Z])([A-Z]+)", "\\U\\1\\L\\2" , x, perl=TRUE))  
}  
  
library(dplyr)  
df <- df %>% mutate(Category = proper\_case(Category),  
 Descript = proper\_case(Descript),  
 PdDistrict = proper\_case(PdDistrict),  
 Resolution = proper\_case(Resolution),  
 Time = as.character(Time))  
df\_sub <- df[1:100,] # display the first 100 rows  
datatable(df\_sub, options = list(pageLength = 5,scrollX='400px'))

# Visualize Data

## Crime across space

library(leaflet)  
  
data <- df[1:100000,] # display the first 10,000 rows  
data$popup <- paste("<b>Incident #: </b>", data$IncidntNum, "<br>", "<b>Category: </b>", data$Category,  
 "<br>", "<b>Description: </b>", data$Descript,  
 "<br>", "<b>Day of week: </b>", data$DayOfWeek,  
 "<br>", "<b>Date: </b>", data$Date,  
 "<br>", "<b>Time: </b>", data$Time,  
 "<br>", "<b>PD district: </b>", data$PdDistrict,  
 "<br>", "<b>Resolution: </b>", data$Resolution,  
 "<br>", "<b>Address: </b>", data$Address,  
 "<br>", "<b>Longitude: </b>", data$X,  
 "<br>", "<b>Latitude: </b>", data$Y)

leaflet(data, width = "100%") %>% addTiles() %>%  
 addTiles(group = "OSM (default)") %>%  
 addProviderTiles(provider = "Esri.WorldStreetMap",group = "World StreetMap") %>%  
 addProviderTiles(provider = "Esri.WorldImagery",group = "World Imagery") %>%  
 # addProviderTiles(provider = "NASAGIBS.ViirsEarthAtNight2012",group = "Nighttime Imagery") %>%  
 addMarkers(lng = ~X, lat = ~Y, popup = data$popup, clusterOptions = markerClusterOptions()) %>%  
 addLayersControl(  
 baseGroups = c("OSM (default)","World StreetMap", "World Imagery"),  
 options = layersControlOptions(collapsed = FALSE)  
 )

In this manner, we can click icons on the map to show incident details. We need to set up some generate some parameters that we concatenate or “paste” together to form these incident descriptions. For example, the concatenated strings pdata$popup, provides the content of the second incident as shown here:

You may notice the “%>%” or forward-pipe operator in the leaflet arguments. The operators pipe their left-hand side values forward into expressions that appear on the right-hand side, rather than from the inside and out.

data$popup[1]

## [1] "<b>Incident #: </b> 150098210 <br> <b>Category: </b> Robbery <br> <b>Description: </b> Robbery, Bodily Force <br> <b>Day of week: </b> Sunday <br> <b>Date: </b> 2/1/2015 <br> <b>Time: </b> 15:45:00 <br> <b>PD district: </b> Zone4 <br> <b>Resolution: </b> None <br> <b>Address: </b> None <br> <b>Longitude: </b> 80.26669397 <br> <b>Latitude: </b> 13.09199072"

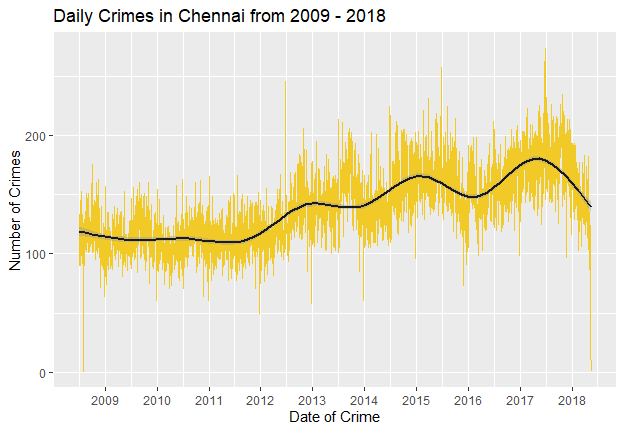
## Crime Over Time

That was not meant to rhyme, but I like it. In this section, we will manipulate the data using the dplyr::mutate function. mutate adds new variables while preserving extisting variables. Below, we used “shades of bue” in the code for our plot, with a dark blue line that smooths the data.

library(dplyr)  
  
df\_crime\_daily <- df %>%  
 mutate(Date = as.Date(Date, "%m/%d/%Y")) %>%  
 group\_by(Date) %>%  
 summarize(count = n()) %>%  
 arrange(Date)

## Daily Crimes Plot with Variance

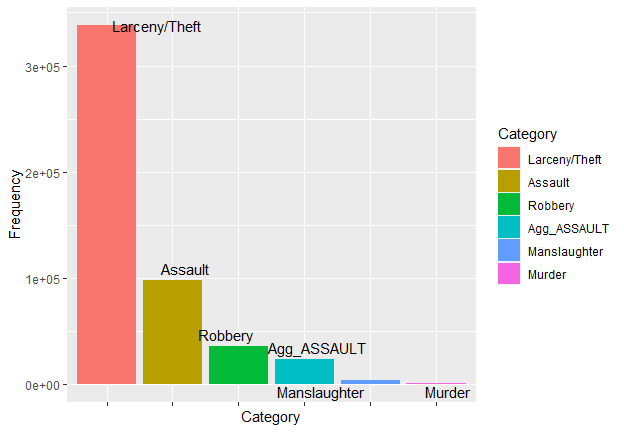
library(ggplot2)  
library(scales)  
plot <- ggplot(df\_crime\_daily, aes(x = Date, y = count)) +  
 geom\_line(color = "#F2CA27", size = 0.1) +  
 geom\_smooth(color = "#1A1A1A") +  
 # fte\_theme() +  
 scale\_x\_date(breaks = date\_breaks("1 year"), labels = date\_format("%Y")) +  
 labs(x = "Date of Crime", y = "Number of Crimes", title = "Daily Crimes in Chennai from 2009 - 2018")  
 plot

 ## Aggregate Data

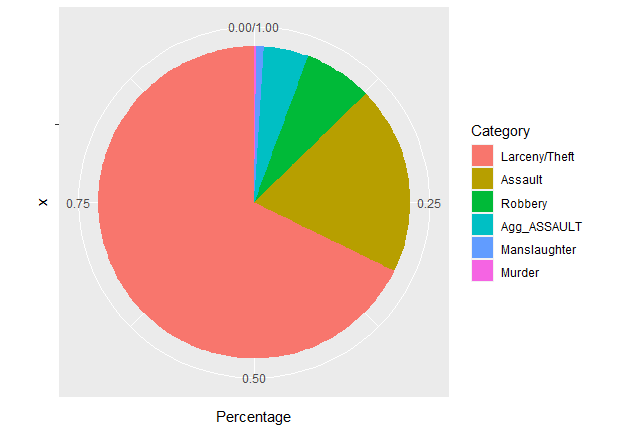
df\_category <- sort(table(df$Category),decreasing = TRUE)  
df\_category <- data.frame(df\_category[df\_category > 1000])  
colnames(df\_category) <- c("Category", "Frequency")  
df\_category$Percentage <- df\_category$Frequency / sum(df\_category$Frequency)  
datatable(df\_category, options = list(scrollX='400px'))

## Create a Bar Chart

library(ggplot2)  
library(ggrepel)  
bp<-ggplot(df\_category, aes(x=Category, y=Frequency, fill=Category)) + geom\_bar(stat="identity") +   
 theme(axis.text.x=element\_blank()) + geom\_text\_repel(data=df\_category, aes(label=Category))  
bp

 ## Create a pie chart based on the incident category.

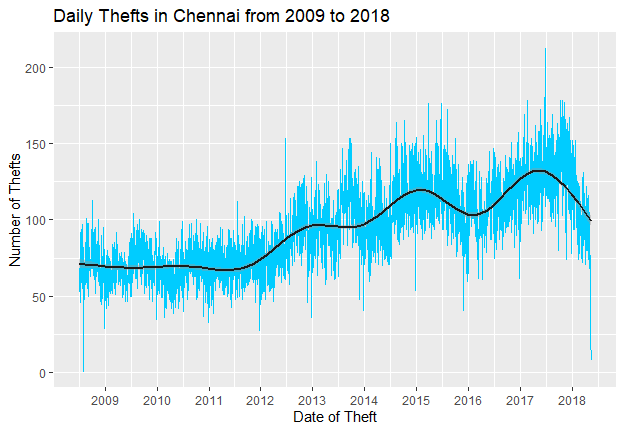
bp<-ggplot(df\_category, aes(x="", y=Percentage, fill=Category)) + geom\_bar(stat="identity")   
pie <- bp + coord\_polar("y")   
pie

 # Temporal Trends ## Theft Over Time

df\_theft <- df %>% filter(grepl("Larceny/Theft", Category))  
  
df\_theft\_daily <- df\_theft %>%  
 mutate(Date = as.Date(Date, "%m/%d/%Y")) %>%  
 group\_by(Date) %>%  
 summarize(count = n()) %>%  
 arrange(Date)

# PlotTemporal Trends

library(ggplot2)  
library(scales)  
plot <- ggplot(df\_theft\_daily, aes(x = Date, y = count)) +  
 geom\_line(color = "#00ccff", size = 0.1) +  
 geom\_smooth(color = "#1A1A1A") +  
 # fte\_theme() +  
 scale\_x\_date(breaks = date\_breaks("1 year"), labels = date\_format("%Y")) +  
 labs(x = "Date of Theft", y = "Number of Thefts", title = "Daily Thefts in Chennai from 2009 to 2018")  
plot

 ## Theft Time Heatmap

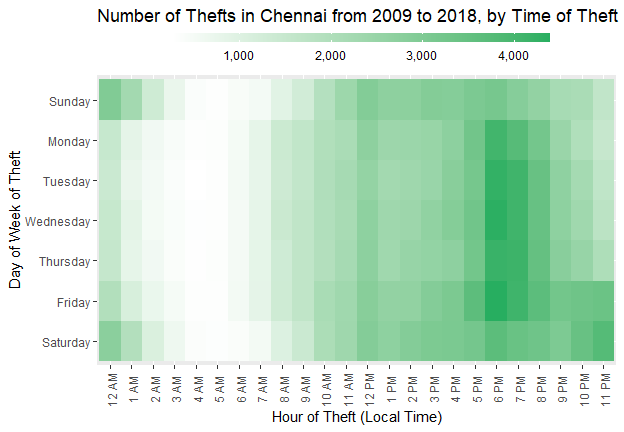
get\_hour <- function(x) {  
 return (as.numeric(strsplit(x,":")[[1]][1]))  
}  
  
df\_theft\_time <- df\_theft %>%  
 mutate(Hour = sapply(Time, get\_hour)) %>%  
 group\_by(DayOfWeek, Hour) %>%  
 summarize(count = n())  
datatable(df\_theft\_time, options = list(scrollX='400px'))

## Reorder and format Factors

dow\_format <- c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")  
hour\_format <- c(paste(c(12,1:11),"AM"), paste(c(12,1:11),"PM"))  
  
df\_theft\_time$DayOfWeek <- factor(df\_theft\_time$DayOfWeek, level = rev(dow\_format))  
df\_theft\_time$Hour <- factor(df\_theft\_time$Hour, level = 0:23, label = hour\_format)  
  
datatable(df\_theft\_time, options = list(scrollX='400px'))

## Create Time Heatmap

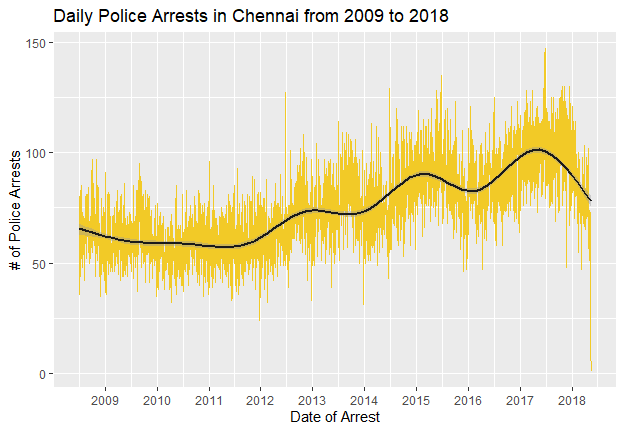
plot <- ggplot(df\_theft\_time, aes(x = Hour, y = DayOfWeek, fill = count)) +  
 geom\_tile() +  
 # fte\_theme() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6), legend.title = element\_blank(), legend.position="top", legend.direction="horizontal", legend.key.width=unit(2, "cm"), legend.key.height=unit(0.25, "cm"), legend.margin=unit(-0.5,"cm"), panel.margin=element\_blank()) +  
 labs(x = "Hour of Theft (Local Time)", y = "Day of Week of Theft", title = "Number of Thefts in Chennai from 2009 to 2018, by Time of Theft") +  
 scale\_fill\_gradient(low = "white", high = "#27AE60", labels = comma)  
plot

 ## Arrest Over Time

df\_arrest <- df %>% filter(grepl("Arrest", Resolution))  
  
df\_arrest\_daily <- df\_arrest %>%  
 mutate(Date = as.Date(Date, "%m/%d/%Y")) %>%  
 group\_by(Date) %>%  
 summarize(count = n()) %>%  
 arrange(Date)

## Daily Arrests

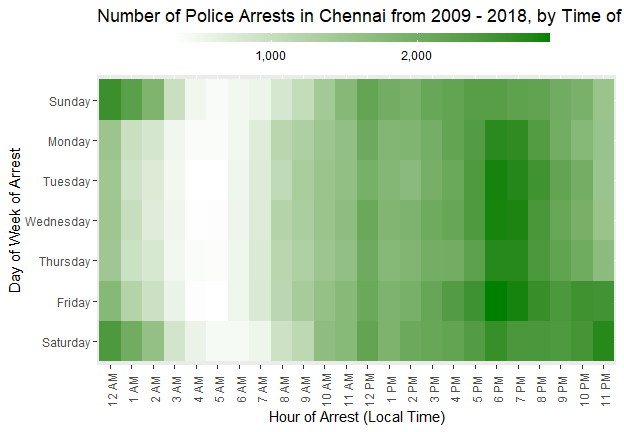
library(ggplot2)  
library(scales)  
plot <- ggplot(df\_arrest\_daily, aes(x = Date, y = count)) +  
 geom\_line(color = "#F2CA27", size = 0.1) +  
 geom\_smooth(color = "#1A1A1A") +  
 # fte\_theme() +  
 scale\_x\_date(breaks = date\_breaks("1 year"), labels = date\_format("%Y")) +  
 labs(x = "Date of Arrest", y = "# of Police Arrests", title = "Daily Police Arrests in Chennai from 2009 to 2018")  
plot

 ## Number of Arrest by Time of Arrest

get\_hour <- function(x) {  
 return (as.numeric(strsplit(x,":")[[1]][1]))  
}  
  
df\_arrest\_time <- df\_arrest %>%  
 mutate(Hour = sapply(Time, get\_hour)) %>%  
 group\_by(DayOfWeek, Hour) %>%  
 summarize(count = n())  
  
dow\_format <- c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")  
hour\_format <- c(paste(c(12,1:11),"AM"), paste(c(12,1:11),"PM"))  
  
df\_arrest\_time$DayOfWeek <- factor(df\_arrest\_time$DayOfWeek, level = rev(dow\_format))  
df\_arrest\_time$Hour <- factor(df\_arrest\_time$Hour, level = 0:23, label = hour\_format)

## Plot

plot <- ggplot(df\_arrest\_time, aes(x = Hour, y = DayOfWeek, fill = count)) +  
 geom\_tile() +  
 # fte\_theme() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6), legend.title = element\_blank(), legend.position="top", legend.direction="horizontal", legend.key.width=unit(2, "cm"), legend.key.height=unit(0.25, "cm"), legend.margin=unit(-0.5,"cm"), panel.margin=element\_blank()) +  
 labs(x = "Hour of Arrest (Local Time)", y = "Day of Week of Arrest", title = "Number of Police Arrests in Chennai from 2009 - 2018, by Time of Arrest") +  
 scale\_fill\_gradient(low = "white", high = "#008000", labels = comma)  
plot

 # Correlation Analysis

## Factor by Crime Category

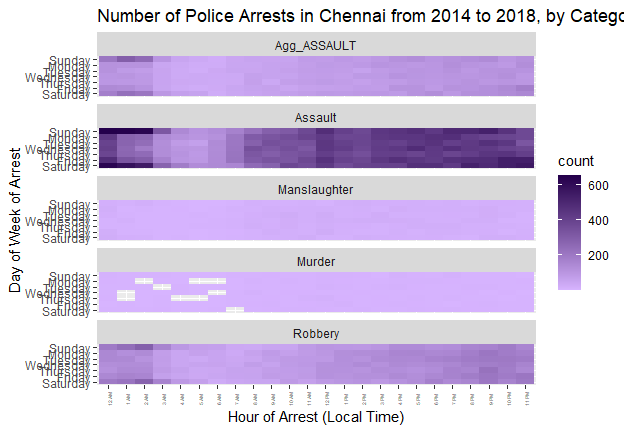
df\_top\_crimes <- df\_arrest %>%  
 group\_by(Category) %>%   
 summarize(count = n()) %>%  
 arrange(desc(count))  
  
datatable(df\_top\_crimes, options = list(pageLength = 10,scrollX='400px'))

## Number of Arrests by Category and time of Arrest

df\_arrest\_time\_crime <- df\_arrest %>%  
 filter(Category %in% df\_top\_crimes$Category[2:19]) %>%  
 mutate(Hour = sapply(Time, get\_hour)) %>%  
 group\_by(Category, DayOfWeek, Hour) %>%   
 summarize(count = n())  
  
df\_arrest\_time\_crime$DayOfWeek <- factor(df\_arrest\_time\_crime$DayOfWeek, level = rev(dow\_format))  
df\_arrest\_time\_crime$Hour <- factor(df\_arrest\_time\_crime$Hour, level = 0:23, label = hour\_format)  
  
datatable(df\_arrest\_time\_crime, options = list(pageLength = 10,scrollX='400px'))

## Number of Arrests by Category and time of Arrest

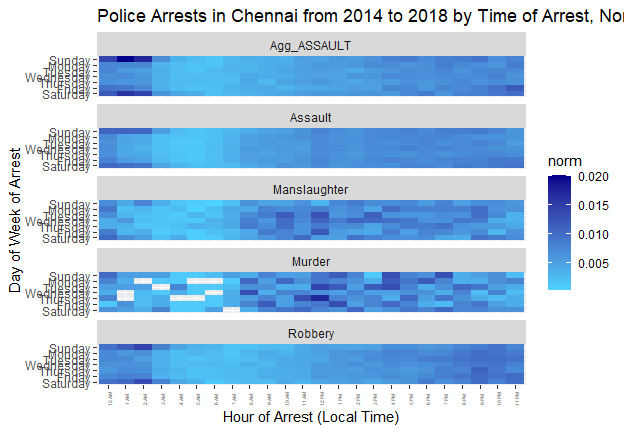
plot <- ggplot(df\_arrest\_time\_crime, aes(x = Hour, y = DayOfWeek, fill = count)) +  
 geom\_tile() +  
 # fte\_theme() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6, size = 4)) +  
 labs(x = "Hour of Arrest (Local Time)", y = "Day of Week of Arrest", title = "Number of Police Arrests in Chennai from 2014 to 2018, by Category and Time of Arrest") +  
 scale\_fill\_gradient(low = "#d7b4ff", high = "#24004b") +  
 facet\_wrap(~ Category, nrow = 6)  
plot

 ## Normailzed Gradients

df\_arrest\_time\_crime <- df\_arrest\_time\_crime %>%  
 group\_by(Category) %>%  
 mutate(norm = count/sum(count))  
  
datatable(df\_arrest\_time\_crime, options = list(pageLength = 10,scrollX='400px'))

## Normalized Number of Arrests by Category and Time of Arrest

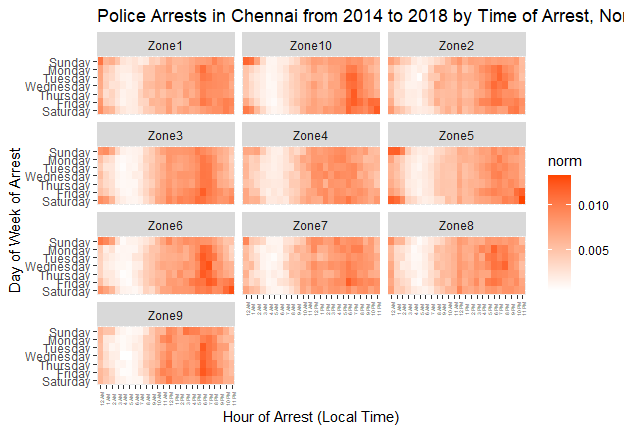
plot <- ggplot(df\_arrest\_time\_crime, aes(x = Hour, y = DayOfWeek, fill = norm)) +  
 geom\_tile() +  
 # fte\_theme() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6, size = 4)) +  
 labs(x = "Hour of Arrest (Local Time)", y = "Day of Week of Arrest", title = "Police Arrests in Chennai from 2014 to 2018 by Time of Arrest, Normalized by Type of Crime") +  
 scale\_fill\_gradient(low = "#4dd2ff", high = "#00008b") +  
 facet\_wrap(~ Category, nrow = 6)  
plot

 ## Factor by Police District

df\_arrest\_time\_district <- df\_arrest %>%  
 mutate(Hour = sapply(Time, get\_hour)) %>%  
 group\_by(PdDistrict, DayOfWeek, Hour) %>%   
 summarize(count = n()) %>%  
 group\_by(PdDistrict) %>%  
 mutate(norm = count/sum(count))  
  
df\_arrest\_time\_district$DayOfWeek <- factor(df\_arrest\_time\_district$DayOfWeek, level = rev(dow\_format))  
df\_arrest\_time\_district$Hour <- factor(df\_arrest\_time\_district$Hour, level = 0:23, label = hour\_format)  
  
datatable(df\_arrest\_time\_district, options = list(pageLength = 10,scrollX='400px'))

## Factor by Police District

plot <- ggplot(df\_arrest\_time\_district, aes(x = Hour, y = DayOfWeek, fill = norm)) +  
 geom\_tile() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6, size = 4)) +  
 labs(x = "Hour of Arrest (Local Time)", y = "Day of Week of Arrest", title = "Police Arrests in Chennai from 2014 to 2018 by Time of Arrest, Normalized by Station") +  
 scale\_fill\_gradient(low = "white", high = "#ff4500") +  
 facet\_wrap(~ PdDistrict, nrow = 4)  
plot

 ## Factor by Month

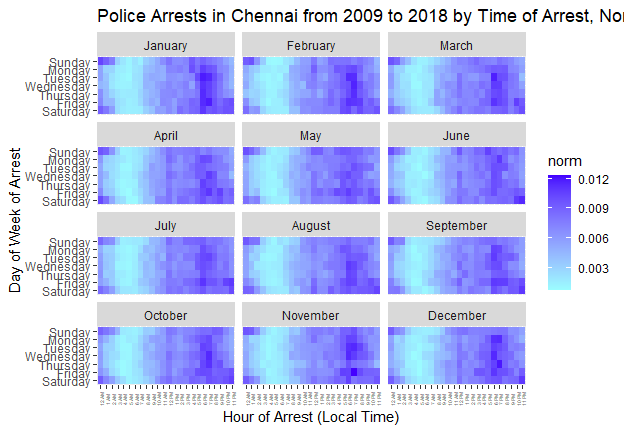
df\_arrest\_time\_month <- df\_arrest %>%  
 mutate(Month = format(as.Date(Date, "%m/%d/%Y"), "%B"), Hour = sapply(Time, get\_hour)) %>%  
 group\_by(Month, DayOfWeek, Hour) %>%   
 summarize(count = n()) %>%  
 group\_by(Month) %>%  
 mutate(norm = count/sum(count))

# Here, we set order of month facets by chronological order instead of alphabetical.

df\_arrest\_time\_month$DayOfWeek <- factor(df\_arrest\_time\_month$DayOfWeek, level = rev(dow\_format))  
df\_arrest\_time\_month$Hour <- factor(df\_arrest\_time\_month$Hour, level = 0:23, label = hour\_format)  
df\_arrest\_time\_month$Month <- factor(df\_arrest\_time\_month$Month,  
 level = c("January","February","March","April","May","June","July","August","September","October","November","December"))

# Plot of Factor by Month

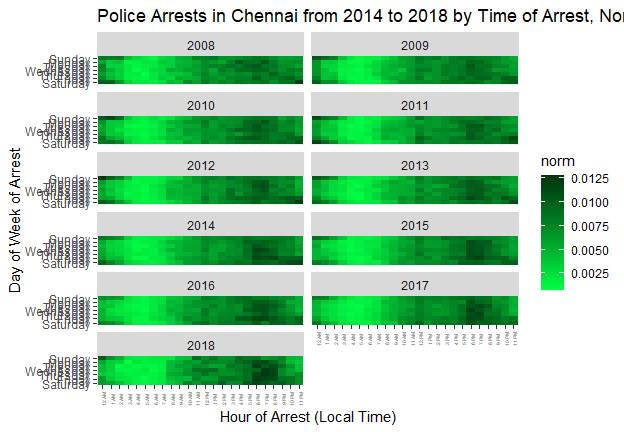
plot <- ggplot(df\_arrest\_time\_month, aes(x = Hour, y = DayOfWeek, fill = norm)) +  
 geom\_tile() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6, size = 4)) +  
 labs(x = "Hour of Arrest (Local Time)", y = "Day of Week of Arrest", title = "Police Arrests in Chennai from 2009 to 2018 by Time of Arrest, Normalized by Month") +  
 scale\_fill\_gradient(low = "#9bfdff", high = "#4401ff") +  
 facet\_wrap(~ Month, nrow = 4)  
plot

 ## Factor By Year #what if things changed overtime?

df\_arrest\_time\_year <- df\_arrest %>%  
 mutate(Year = format(as.Date(Date, "%m/%d/%Y"), "%Y"), Hour = sapply(Time, get\_hour)) %>%  
 group\_by(Year, DayOfWeek, Hour) %>%   
 summarize(count = n()) %>%  
 group\_by(Year) %>%  
 mutate(norm = count/sum(count))  
  
df\_arrest\_time\_year$DayOfWeek <- factor(df\_arrest\_time\_year$DayOfWeek, level = rev(dow\_format))  
df\_arrest\_time\_year$Hour <- factor(df\_arrest\_time\_year$Hour, level = 0:23, label = hour\_format)

## Police Arrest Normalized by YEar

plot <- ggplot(df\_arrest\_time\_year, aes(x = Hour, y = DayOfWeek, fill = norm)) +  
 geom\_tile() +  
 # fte\_theme() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.6, size = 4)) +  
 labs(x = "Hour of Arrest (Local Time)", y = "Day of Week of Arrest", title = "Police Arrests in Chennai from 2014 to 2018 by Time of Arrest, Normalized by Year") +  
 scale\_fill\_gradient(low = "#01ff44", high = "#00340e") +  
 facet\_wrap(~ Year, nrow = 6)  
plot

 ## Works CIted