US CAR ACCIDENTS Data Analysis

Krashagi Gupta

11 December, 2021

Contents

```
#Please know that you can use a html output but you need to keep the sectioning.

#Please Reference your figures and tables so that it is readable

#Each update is important to keep for grading
```

1 Update 6

- I tried to answer the question : accidents during the day and night.
  - Divided it by time
  - Recognised that peak times, is when accidnets increase
  - During the day, at peak hours accidents spread more than in the night,
  - meaning there are hotzones at night, where accidents are happening, as compared to day
  - divided this plot by months and recognised that accidnets are peaking at night as the year comes to an end, at night, but during the day its fine
  - Saw the hotzones on the map and recognised that none of the pois are present in the hotzones and the only POI that shows up for lesser hotzones is a junction

2 Update 5

I tried to find the relationships between number of accidents and variables. * Number of accidents vs time of twlight * Number of accidents vs POI * Here I looked for major causes for accidents,and found it to be distracted driving among other things, I considered what are the pois that could reduce the speed of the car and others that could cause the driver to change their speed abruptly, and put them in as positive and negative, one that would cause accidents and one that would prevent accidents I saw that when combined the "causers" the number of accidents increased and when i combined the prevents the number of accidents decreased.

To consider the environmental variables * I divided USA into climatic groups, and found thenumber of accidents based on precipitation, visibility, temperature, could not find anything of value here, so i would like some help in this.

3 Update 4

- Please put a bulleted list of things you have accomplished since the last update
  - Update 4 part of the data visualisations i did.

4 Update 3

- Found that my data set was just a sample, and therefore had to change the dataset.

5 Update 2

I plan to take the fast food dataset to answer: 1. What is America's top five restaurants ? 2. Based on The restaurant names, make a wordcloud of what is the food that is generally eaten in america. 3. Use the income data from the census and correlate a county's number and type of restaurants to the per capita income of that county. 4. I had to clean the senus data to sum the income in the county, it had a lot of NA values, and that failed. 5. I tried to make some basic plots to get comfortable with gg plots and also dplyr. 6. I am nt really sure of what my big questions are with respect to the data set and therefore am just exploring the dataset trying to find something interesting to corelate. 7. My plan is vaguely to relate the type of restaurants to the location, and to correlate the location to the people living there. What preferences do people having similar backgrounds have when it comes to food.

6 Update 1

1. What is America's top five restaurants ?
2. Based on The restaurant names, make a wordcloud of what is the food that is generally eaten in america.
3. Use the income data from the census and correlate a county's number and type of restaurants to the per capita income of that county.
4. I had to clean the senus data to sum the income in the county, it had a lot of NA values, and that failed.
5. I tried to make some basic plots to get comfortable with gg plots and also dplyr.
6. I am nt really sure of what my big questions are with respect to the data set and therefore am just exploring the dataset trying to find something interesting to corelate.
7. My plan is vaguely to relate the type of restaurants to the location, and to correlate the location to the people living there. What preferences do people having similar backgrounds have when it comes to food.

7 Executive Summary

- Dataset consists of 43 variables, variables include info about : when and where of car accidents, severity, environmental and weather conditions and Point of interests on the road, illumination based on twilight times
- major data cleaning - renaming variables, extracting time (day, date , month, year), extracting time in hrs
- Findings from EDA
- Accidents peak during peak traffic hours, more accidents happen around accidental hotzones in the evening peak hours than in the morning.
- Accidents come down during the weekends, and peak on thursdays
- Accidents increase incredibly in the last quarter of the year.
- There are significant number of accidents happen in the presence of junction,station, crossing, Traffic signal

- Built a model that predicted severity based on POIs mentioned above and months of the year, only number 2 was predicted correctly for a fraction and the accuracy was 78.29 %
- The dataset was quite data heavy in certain categories- say severity 2 and accidents in California state, therefore unsure, how accurate is the result obtained. However, given the dataset as it is, the results obtained make sense.

8 Abstract

Car accidents are a leading cause of deaths in the USA. Using this dataset trends in car accidents within a day, within a week and within a month are visualised. Hotspots are recognized in the morning peak and evening peak hours, Variation of car accidents hotspots in the day and the night as months pass by are visualised. Recognised that certain Points of interest are present in a significant number of car accidents, and developed a model that used time and POIS as predictors ,which had prediction accuracy of 78.29 %.

9 Introduction

- The motivation of this project was to essentially learn how to do a data- science project with a large dataset with numerical and categorical variables.
- I wanted to use my current knowledge about the road safety protocols and knowledge about the USA, to make predictions and see if it tallies with the data.
- Research in this area has a lot of useful applications
- real-time car accident predictions,
- studying car accidents hotspot locations,
- casualty analysis
- extracting cause and effect rules to predict car accidents studying the impact of precipitation or other environmental stimuli on accident occurrence.
- Explanation of your data
  - This is a countrywide car accident dataset, which covers 49 states of the USA. The accident data are collected from February 2016 to Dec 2020, using multiple APIs that provide streaming traffic incident (or event) data. These APIs broadcast traffic data captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks.
  - The variables can be categorized as responses and predictors, there are environmental condition predictors, road POI (Point of interests) predictors, when and where the accidents occurred with what severity.
- What data would be necessary to improve your analysis?
- Since drunk driving is an essential cause of car accidents, the state of the drivers involved could be of help.

10 Data Science Methods

- Geo-spatial analysis, model building
- packages : ggmap, nnet

11 Exploratory Data Analysis

11.1 Explanation of your data set

- How many variables?
- What are the data classes?
- How many levels of factors for factor variables?
- Is your data suitable for a project analysis?
- Write you databook, defining variables, units and structures
- There were 47 variables and I removed 3 to get 43 variables.
- classes can be seen in the glimpse output
- There are many factor variables :
- Important ones include : severity - 5 levels
- POIs and twilight variables - 2 levels

```
src = "datasets/US_Accidents_Dec20_updated.csv/US_Accidents_Dec20_updated.csv"

orig_df <- read.csv(src)

proj_df <- orig_df %>% # only remove id
    select( -c("ID", "Description","Number", "Street"))



glimpse(proj_df)
```

```
## Rows: 1,516,064
## Columns: 43
## $ Severity              <int> 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ Start_Time            <fct> 2016-02-08 00:37:08, 2016-02-08 05:56:20, 2016-0~
## $ End_Time              <fct> 2016-02-08 06:37:08, 2016-02-08 11:56:20, 2016-0~
## $ Start_Lat             <dbl> 40.10891, 39.86542, 39.10266, 39.10148, 41.06213~
## $ Start_Lng             <dbl> -83.09286, -84.06280, -84.52468, -84.52341, -81.~
## $ End_Lat               <dbl> 40.11206, 39.86501, 39.10209, 39.09841, 41.06217~
## $ End_Lng               <dbl> -83.03187, -84.04873, -84.52396, -84.52241, -81.~
## $ Distance.mi.          <dbl> 3.230, 0.747, 0.055, 0.219, 0.123, 0.500, 1.427,~
## $ Side                  <fct> R, R, R, R, R, R, R, R, R, R, R, R, R, R, R, R, ~
## $ City                  <fct> Dublin, Dayton, Cincinnati, Cincinnati, Akron, C~
```

```
## $ County                <fct> Franklin, Montgomery, Hamilton, Hamilton, Summit~
## $ State                 <fct> OH, OH, OH, OH, OH, OH, OH, OH, OH, OH, OH, OH, ~
## $ Zipcode               <fct> 43017, 45424, 45203, 45202, 44311, 45217, 45176,~
## $ Country               <fct> US, US, US, US, US, US, US, US, US, US, US, US, ~
## $ Timezone              <fct> US/Eastern, US/Eastern, US/Eastern, US/Eastern, ~
## $ Airport_Code          <fct> KOSU, KFFO, KLUK, KLUK, KAKR, KLUK, KI69, KI69, ~
## $ Weather_Timestamp     <fct> 2016-02-08 00:53:00, 2016-02-08 05:58:00, 2016-0~
## $ Temperature.F.        <dbl> 42.1, 36.9, 36.0, 36.0, 39.0, 37.0, 35.6, 35.6, ~
## $ Wind_Chill.F.         <dbl> 36.1, NA, NA, NA, NA, 29.8, 29.2, 29.2, NA, 30.0~
## $ Humidity...           <dbl> 58, 91, 97, 97, 55, 93, 100, 100, 100, 92, 70, 1~
## $ Pressure.in.          <dbl> 29.76, 29.68, 29.70, 29.70, 29.65, 29.69, 29.66,~
## $ Visibility.mi.        <dbl> 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, ~
## $ Wind_Direction        <fct> SW, Calm, Calm, Calm, Calm, WSW, WSW, WSW, SW, S~
## $ Wind_Speed.mph.       <dbl> 10.4, NA, NA, NA, NA, 10.4, 8.1, 8.1, 2.3, 3.5, ~
## $ Precipitation.in.     <dbl> 0.00, 0.02, 0.02, 0.02, NA, 0.01, NA, NA, NA, 0.~
## $ Weather_Condition     <fct> Light Rain, Light Rain, Overcast, Overcast, Over~
## $ Amenity               <fct> False, False, False, False, False, False, False,~
## $ Bump                  <fct> False, False, False, False, False, False, False,~
## $ Crossing              <fct> False, False, False, False, False, False, False,~
## $ Give_Way              <fct> False, False, False, False, False, False, False,~
## $ Junction              <fct> False, False, True, True, False, False, False, F~
## $ No_Exit               <fct> False, False, False, False, False, False, False,~
## $ Railway               <fct> False, False, False, False, False, False, False,~
## $ Roundabout            <fct> False, False, False, False, False, False, False,~
## $ Station               <fct> False, False, False, False, False, False, False,~
## $ Stop                  <fct> False, False, False, False, False, False, False,~
## $ Traffic_Calming       <fct> False, False, False, False, False, False, False,~
## $ Traffic_Signal        <fct> False, False, False, False, False, False, True, ~
## $ Turning_Loop          <fct> False, False, False, False, False, False, False,~
## $ Sunrise_Sunset        <fct> Night, Night, Night, Night, Night, Day, Day, Day~
## $ Civil_Twilight        <fct> Night, Night, Night, Night, Night, Day, Day, Day~
## $ Nautical_Twilight     <fct> Night, Night, Night, Night, Day, Day, Day, Day, ~
## $ Astronomical_Twilight <fct> Night, Night, Day, Day, Day, Day, Day, Day, Day,~
```

11.2 Data Cleaning

- What you had to do to clean your data
- I had to rename the columns
- Extract, month, day and year info and also time info.

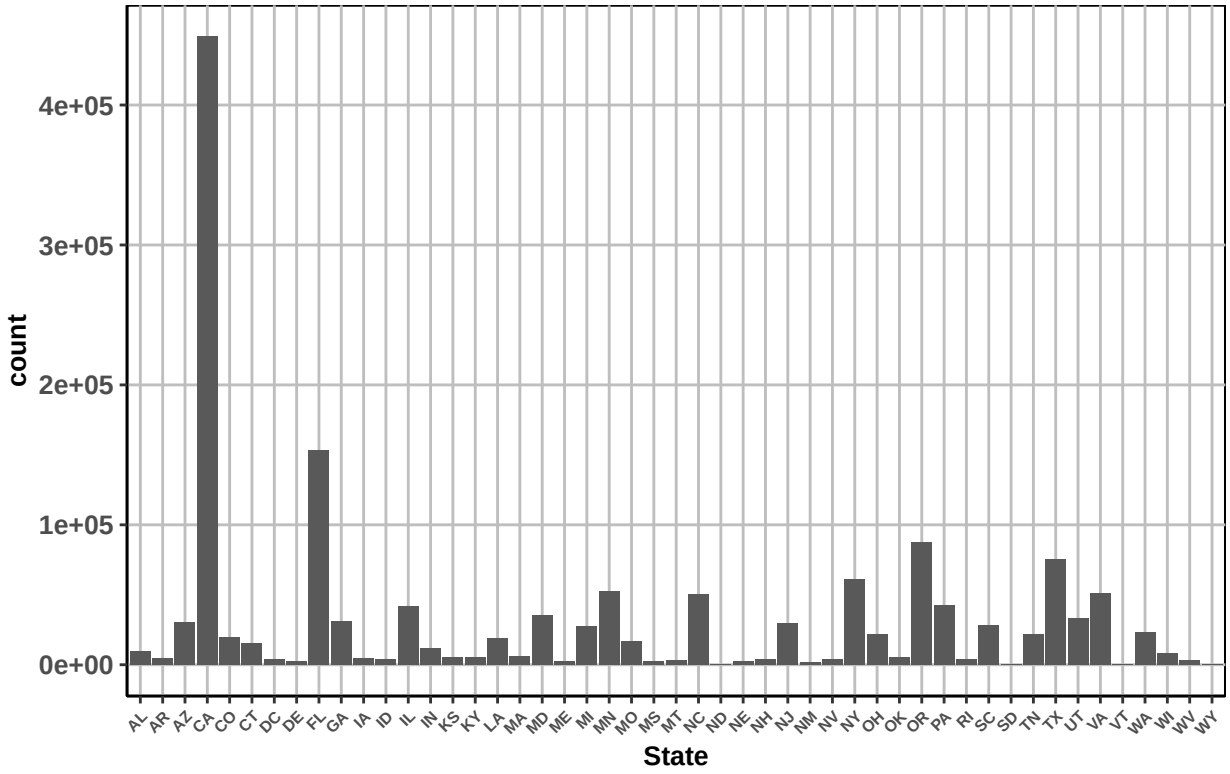
```
## renaming columns
proj_df <- proj_df %>%
  rename(Distance_mi = Distance.mi., Temperature_F = Temperature.F.,
         Wind_Chill_F = Wind_Chill.F., Humidity = Humidity...,
         Pressure_in = Pressure.in., Visibility_mi =Visibility.mi.,
         Wind_Speed_mph = Wind_Speed.mph., Precipitation_in = Precipitation.in.,
         )
```

```
t <- theme(panel.background = element_rect(fill = "white", colour = "black"), panel.grid.major
axis.text = element_text(size = 10, face = "bold"),
axis.text.x = element_text(size = 6, face = "bold", angle = 45, hjust = 1),
axis.title = element_text(size = 10, face = "bold"),
strip.text = element_text(size = 12
, face = "bold"),
text = element_text(size = 18), panel.spacing = unit(1, "lines"),
legend.position = "top")


time_div <- proj_df %>%
  separate(Start_Time, c("start_date", "start_time"), " ", extra = "merge") %>% mutate(day_week
  separate(start_date, c ("year", "month", "day"), "-", extra = "merge")
```

11.3 Data Vizualizations

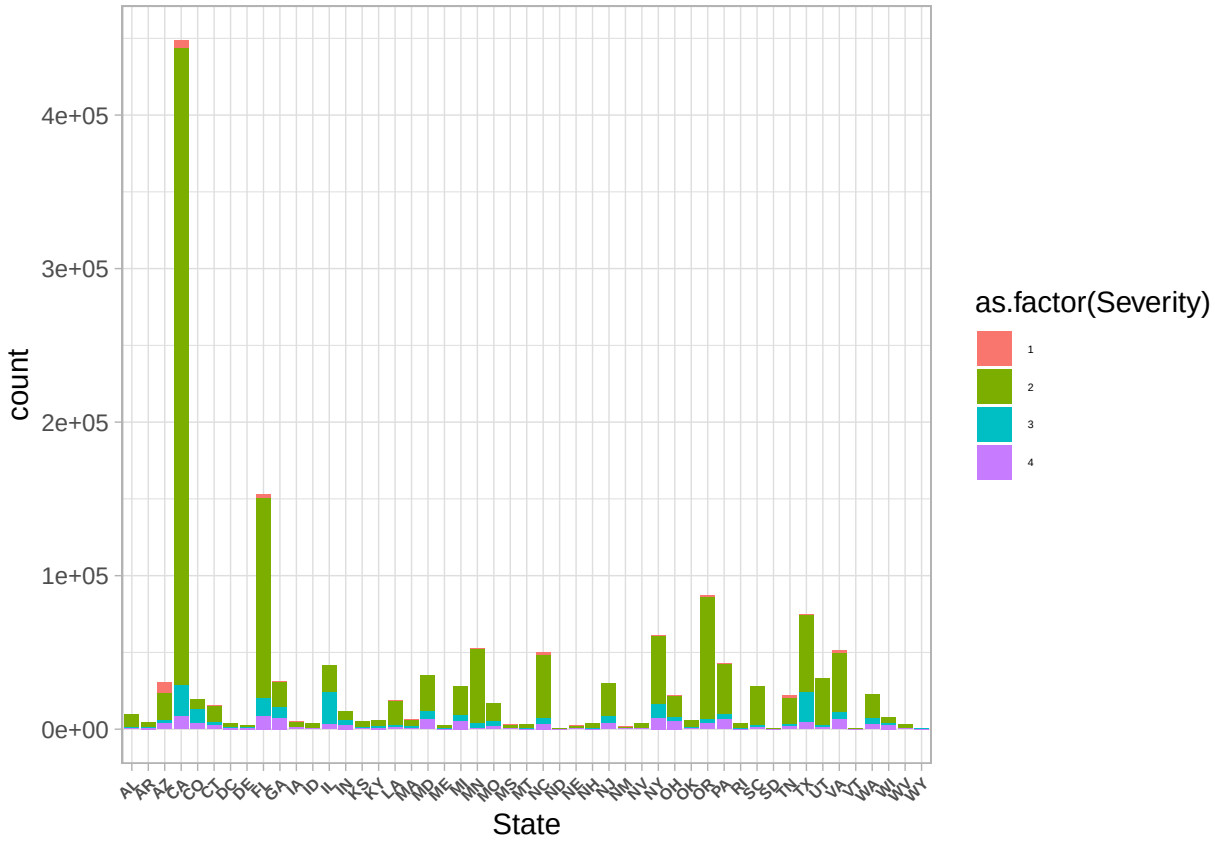
```
## let us look by state, severity the number of accidents


t <- theme(panel.background = element_rect(fill = "white", colour = "black"), panel.grid.major
axis.text = element_text(size = 10, face = "bold"),
axis.text.x = element_text(size = 6, face = "bold", angle = 45, hjust = 1),
axis.title = element_text(size = 10, face = "bold"),
strip.text = element_text(size = 12
, face = "bold"),
text = element_text(size = 18), panel.spacing = unit(1, "lines"),
legend.position = "top")

ggplot(data = proj_df, aes(State)) +
geom_histogram(stat = "count") + theme_classic() + t +
labs(title = "Total accidents by state" )
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
ggplot(data = proj_df, aes(x = State, fill = as.factor(Severity))) +
geom_histogram(stat = "count",show.legend = TRUE) +
theme(legend.text = element_text(size = 4) ) +
theme(legend.key.size = unit(0.5, 'cm') ,
axis.text.x = element_text(size = 6,
face = "bold", angle = 45, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
labs(title = "Total accidents by state " )
```

```
## $title
## [1] "Total accidents by state "
##
## attr(,"class")
## [1] "labels"
```

```
by_state <- proj_df  %>%
  group_by(State) %>%
  tally %>%
  arrange(desc(n))

head(by_state)
```

11.3.0.1 Identifying the top 5 accident state

```
## # A tibble: 6 x 2
##   State      n
##   <fct>  <int>
## 1 CA    448833
## 2 FL    153007
## 3 OR     87484
## 4 TX     75142
## 5 NY     60974
## 6 MN     52345
```

```
tail(by_state) %>%
  arrange(n)
```

```
## # A tibble: 6 x 2
##   State     n
##   <fct> <int>
## 1 SD      213
## 2 WY      330
## 3 VT      352
## 4 ND      455
## 5 NM     1467
## 6 NE     2178
```

```
by_state_sev_4 <- proj_df  %>%
  filter(Severity == 4)%>%
  group_by(State) %>%
  tally %>%
  arrange(desc(n))

head(by_state_sev_4)
```

11.3.0.2 Identifying the top 5 accident state with severity = 4

```
## # A tibble: 6 x 2
##   State     n
##   <fct> <int>
## 1 FL     8566
## 2 CA     8321
## 3 NY     7324
## 4 GA     7299
## 5 MD     6497
## 6 PA     6481
```

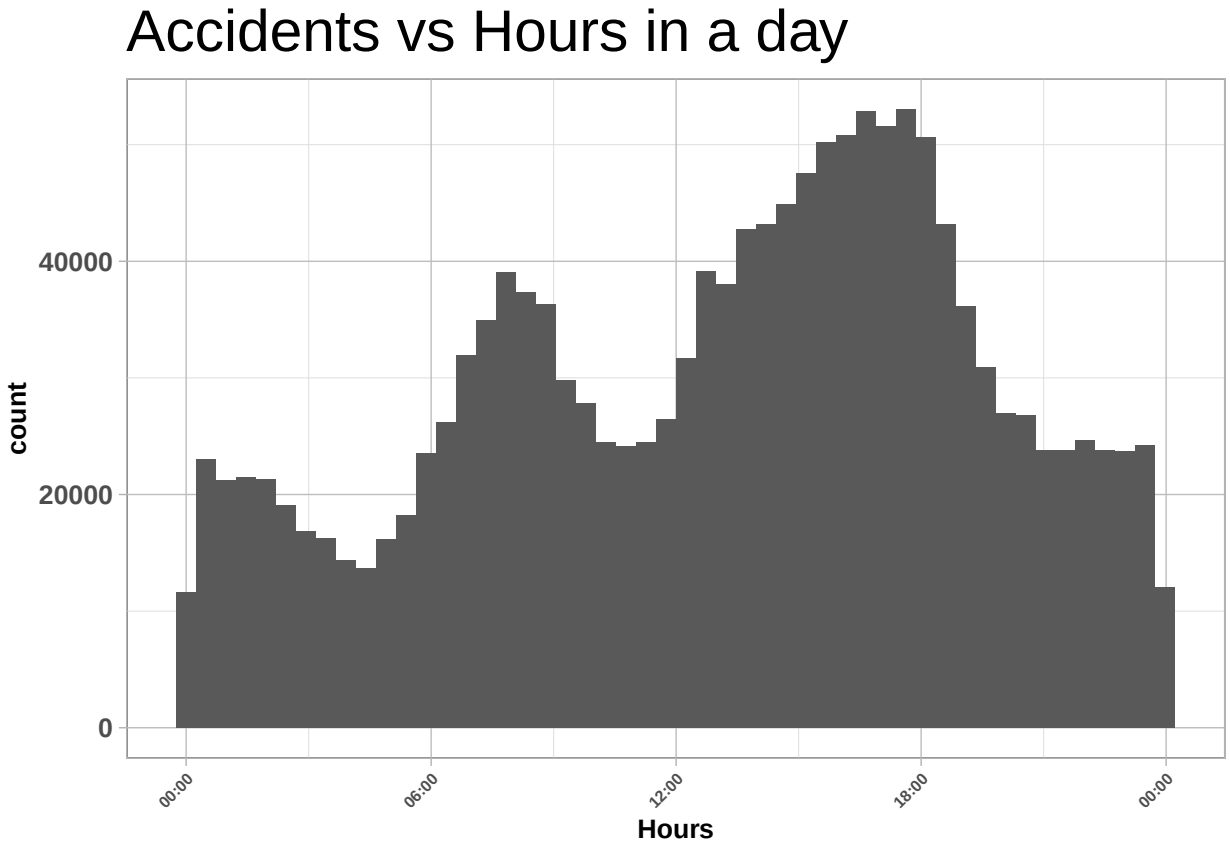
```
tail(by_state_sev_4) %>%
   arrange(n)
```

```
## # A tibble: 6 x 2
##   State     n
##   <fct> <int>
## 1 ND       11
## 2 SD       40
## 3 VT       61
## 4 ME       97
## 5 RI      109
## 6 MT      112
```

```
# Time in a day
```

```
time_div %>%
    mutate(start_time = as.POSIXct(hms::parse_hm(start_time))) %>%
    ggplot(aes(start_time)) +
    geom_histogram(bins = 50) +
    scale_x_datetime(date_labels = "%H:%M") +
    labs(title = "Accidents vs Hours in a day") + xlab("Hours") + t
```

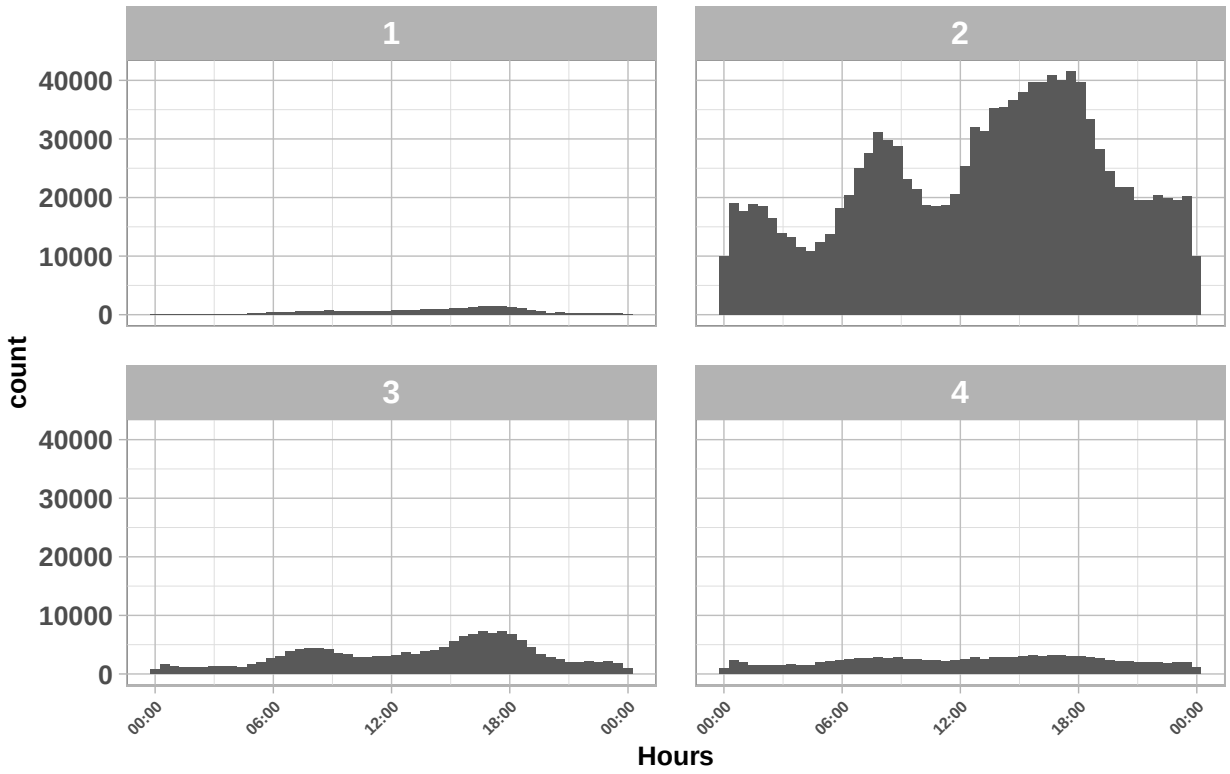
11.3.0.3 Time break down




```
# there are two peaks : 7 am and 6 pm, both the times when most people are
# moving for work
# More accidents happen at night than in the day
```

```
time_div %>%
    mutate(start_time = as.POSIXct(hms::parse_hm(start_time))) %>%
    ggplot(aes(start_time, )) +
    geom_histogram(bins = 50, ) +
    scale_x_datetime(date_labels = "%H:%M") + facet_wrap(~ Severity)+
    labs(title = "Severity of Accidents vs Hours in a day") + xlab("Hours")+ t
```

11.3.0.4 Severity distribution within a day

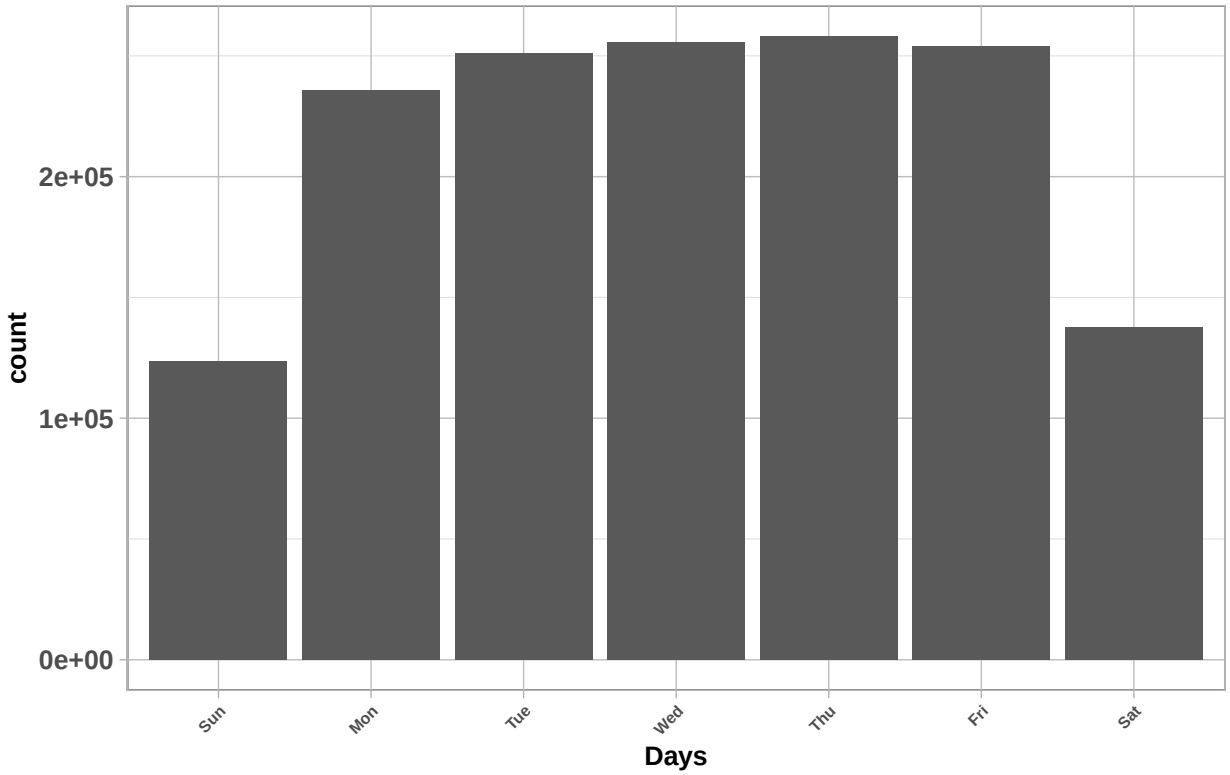



```
### severity of 4, was mostly constant throughout the day, which means people's
### movement was not a factor that determined, peak of accidents of severity 4.
## the most severe accidents are not a function of peak time, but severity of 2
## and 3 are both related to peak hour time, and evening seems to be a popular
## time for it
```

```
proj_df <- proj_df %>%
  separate(Start_Time, c("start_date", "start_time"), " ", extra = "merge")%>%
  mutate(day_week = wday(start_date, label=T) )

proj_df %>%
  ggplot(aes(day_week  )) +
    geom_histogram(bins = 50,stat = "count" ) +
    labs(title = "Accidents vs Days in a week") + xlab("Days") + t
```

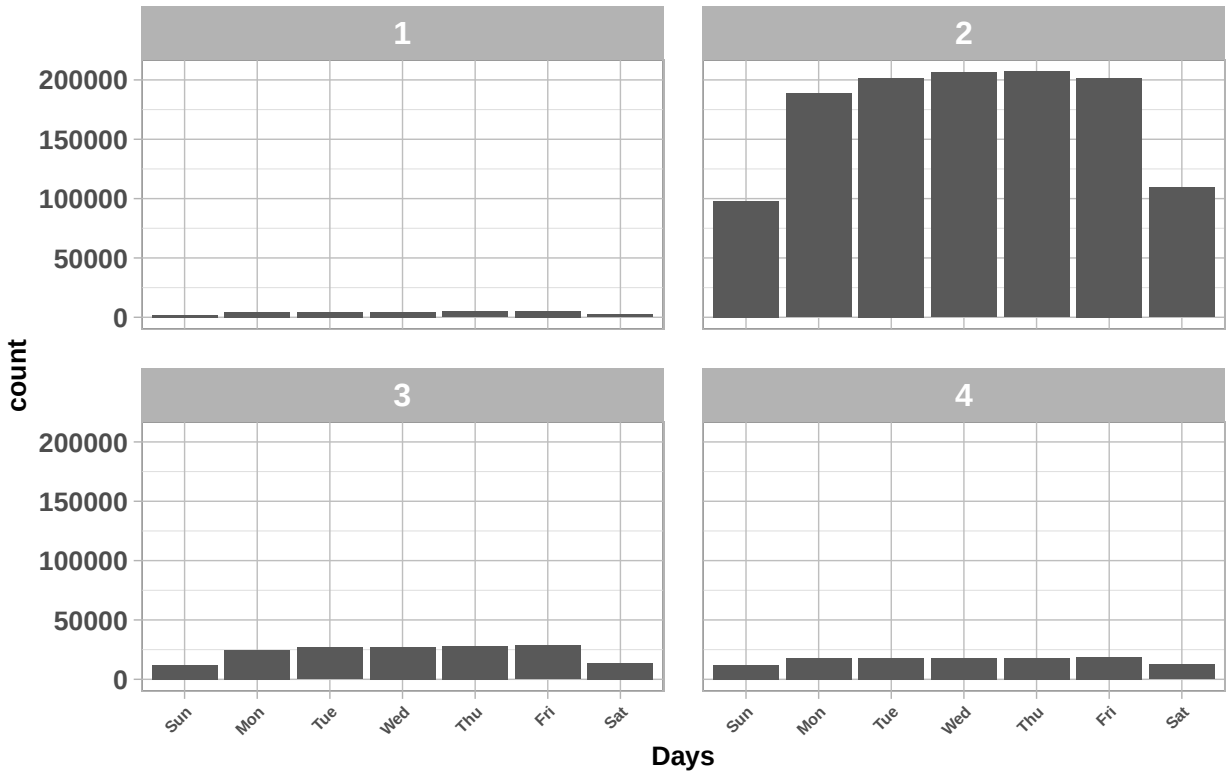
11.3.0.5 time in a week

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
proj_df %>%
  ggplot(aes(day_week)) +
    geom_histogram(bins = 50,stat = "count" ) + facet_wrap( ~ Severity) +
  labs(title = "Severity of Accidents vs Days in a week") + xlab("Days") + t
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

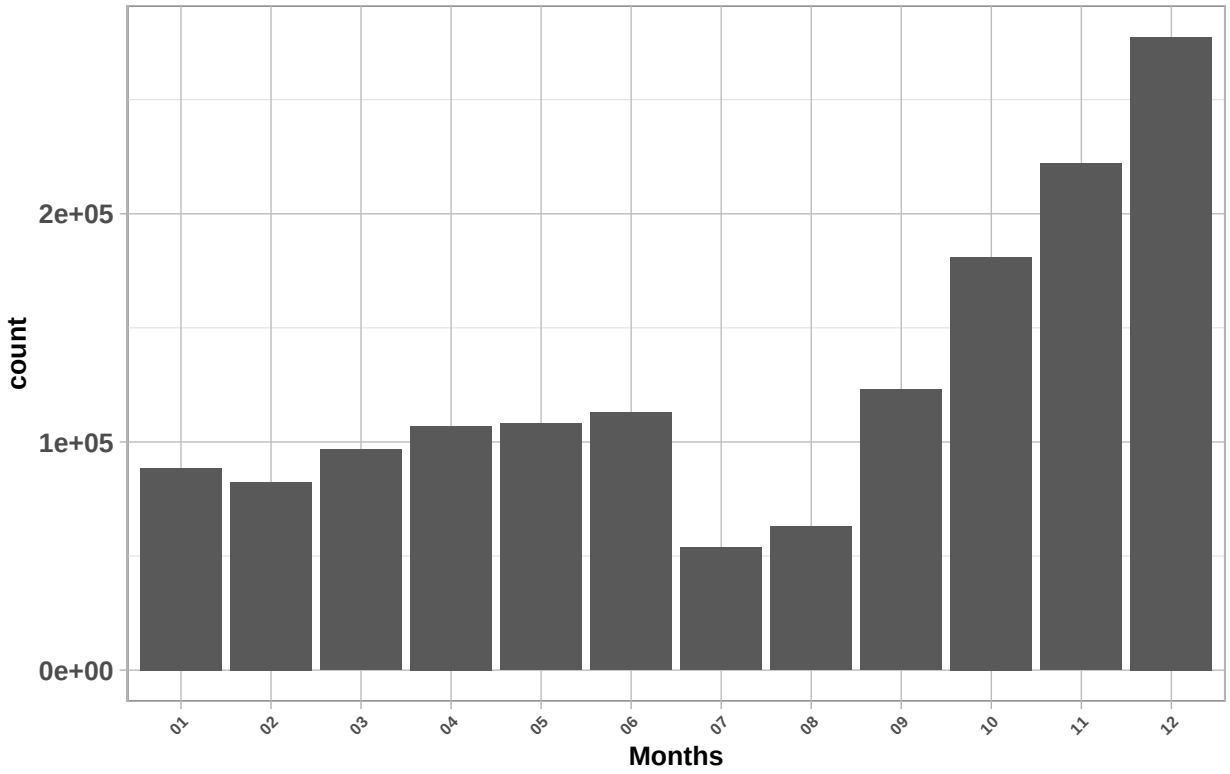



```
## In general accidents drop down in the weekends, and reach a very non-distinct
## peak by thursday.
## The severity of accidents follow a similar trend, though it is more accurate
## to say that severity 1 and 4 are pretty much uniform throughout the week
```

```
time_div %>%
  ggplot(aes(month  )) +
    geom_histogram(bins = 50,stat = "count" )  +
   labs(title = "Accidents vs Months in a year") + xlab("Months") + t
```

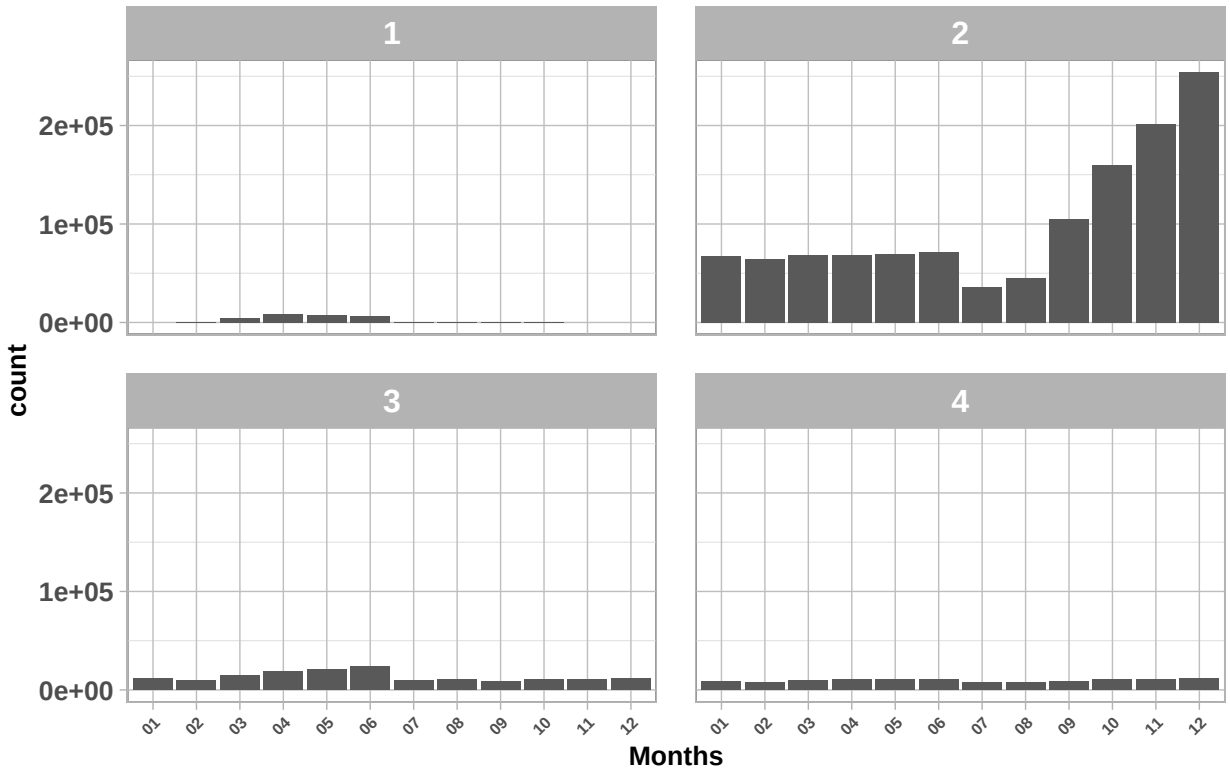
11.3.0.6 time in a year

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
## maybe people's outlook gets better
# time_div %>%
#   ggplot(aes(month, fill = as.factor(day))) +
#     geom_histogram(bins = 50,stat = "count")



time_div %>%
  ggplot(aes(month  )) +
    geom_histogram(bins = 50,stat = "count" ) + facet_wrap( ~ Severity) +
   labs(title = "Severity of Accidents vs Months in a year") + xlab("Months") + t
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
## Accidents drop in july and august and peak in december in the US.
## Severity of 4 remains constant throughout the year
```

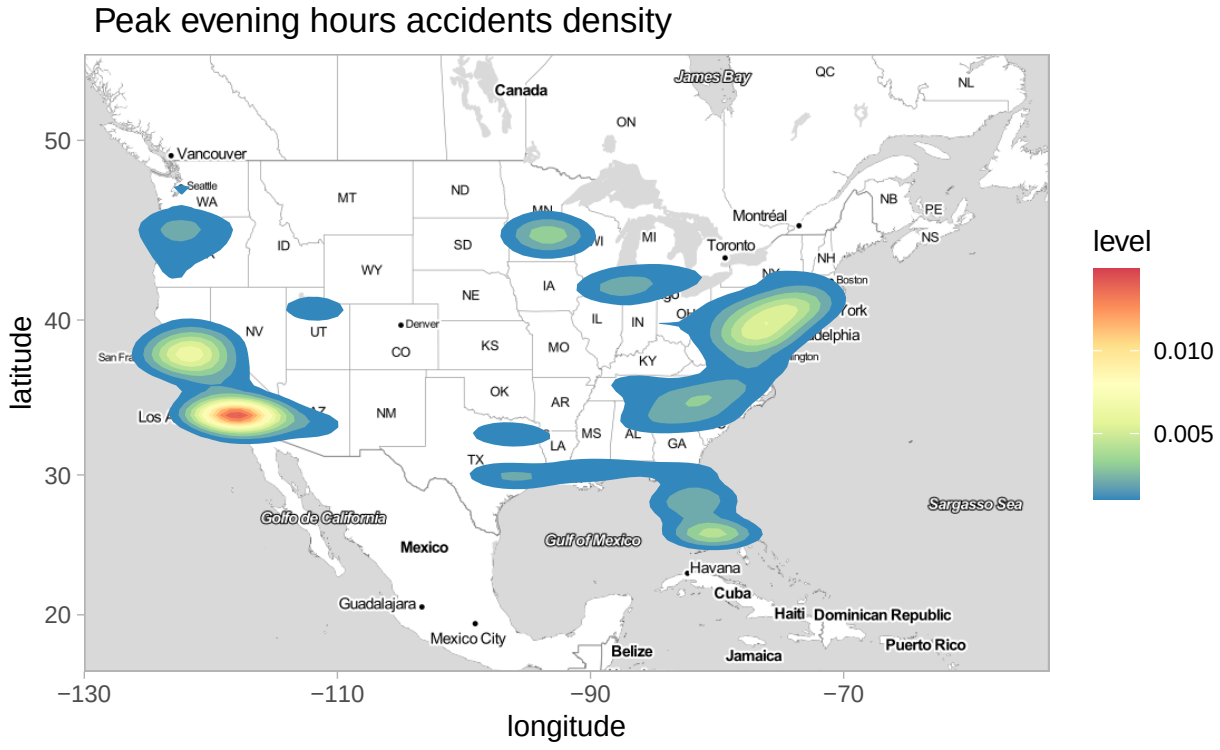
```
unitedStatesmap <- get_stamenmap(
bbox = c(left = -130 , bottom = 15.74, right = -53.76, top = 54.18),
maptype = "toner-lite",
zoom = 4
)
```

11.3.0.7 Peak hours in a day on a map

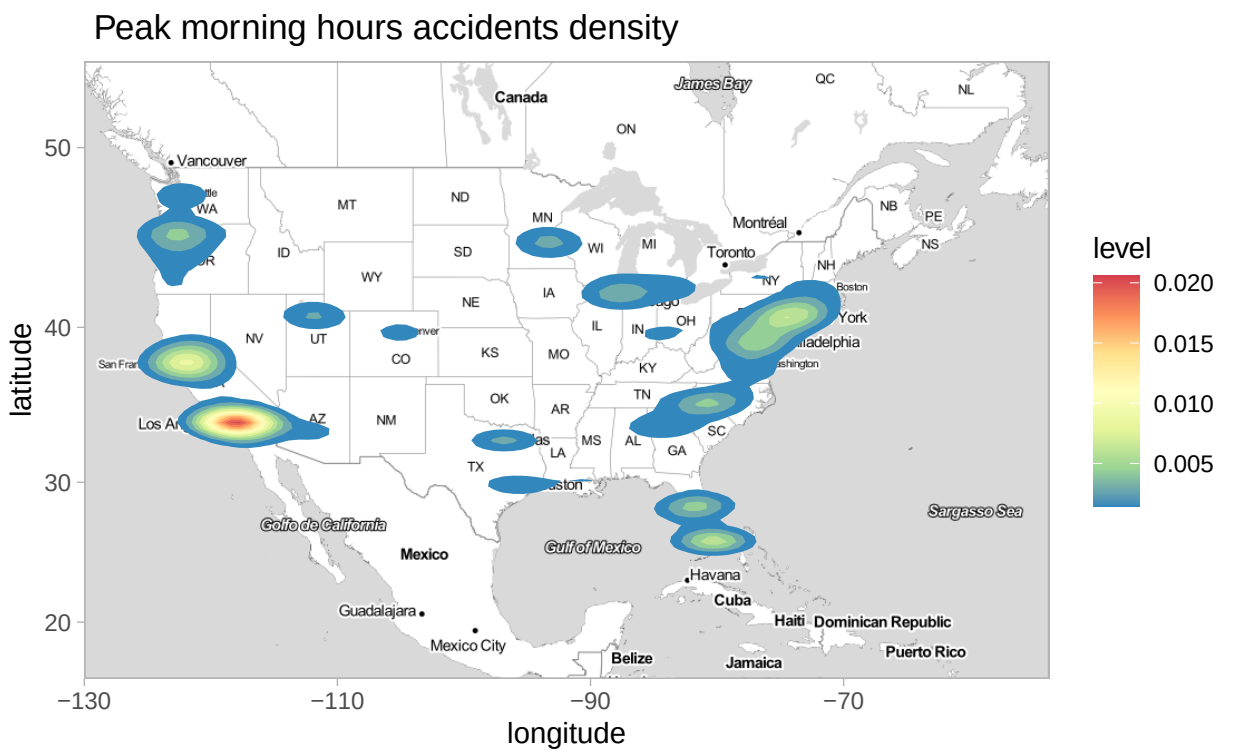
```
## Source : http://tile.stamen.com/toner-lite/4/2/5.png
## Source : http://tile.stamen.com/toner-lite/4/3/5.png
## Source : http://tile.stamen.com/toner-lite/4/4/5.png
## Source : http://tile.stamen.com/toner-lite/4/5/5.png
## Source : http://tile.stamen.com/toner-lite/4/2/6.png
## Source : http://tile.stamen.com/toner-lite/4/3/6.png
## Source : http://tile.stamen.com/toner-lite/4/4/6.png
## Source : http://tile.stamen.com/toner-lite/4/5/6.png
## Source : http://tile.stamen.com/toner-lite/4/2/7.png
## Source : http://tile.stamen.com/toner-lite/4/3/7.png
```

```
## Source : http://tile.stamen.com/toner-lite/4/4/7.png

## Source : http://tile.stamen.com/toner-lite/4/5/7.png
```

```
day <- time_div %>%
    mutate(hour = as.POSIXct(hms::parse_hm(start_time))) %>%
    filter(hour >= as.POSIXct("1970-01-01 05:30:00") &
             hour <= as.POSIXct("1970-01-01 07:30:00"))

    # ggmap(unitedStatesmap) +
    # geom_point(data= day,aes(x = Start_Lng , y = Start_Lat,
    #                          ), size = 0.1 ,alpha = 1 / 3)

night <- time_div %>%
    mutate(hour = as.POSIXct(hms::parse_hm(start_time))) %>%
    filter(hour >= as.POSIXct("1970-01-01 17:30:00") &
             hour <= as.POSIXct("1970-01-01 18:30:00"))

    # ggmap(unitedStatesmap) +
    # geom_point(data= night,aes(x = Start_Lng , y = Start_Lat,
    #                          ), size = 0.1 ,alpha = 1 / 3)

ggmap(unitedStatesmap) +
stat_density2d( aes(x = Start_Lng, y = Start_Lat , fill = ..level..),
size = 0.02, bins = 15, data = night,
geom = "polygon") + scale_fill_gradientn(colours=rev(brewer.pal(7,"Spectral")))+
labs(title = " Peak evening hours accidents density ") + xlab("longitude") +
  ylab("latitude")
```

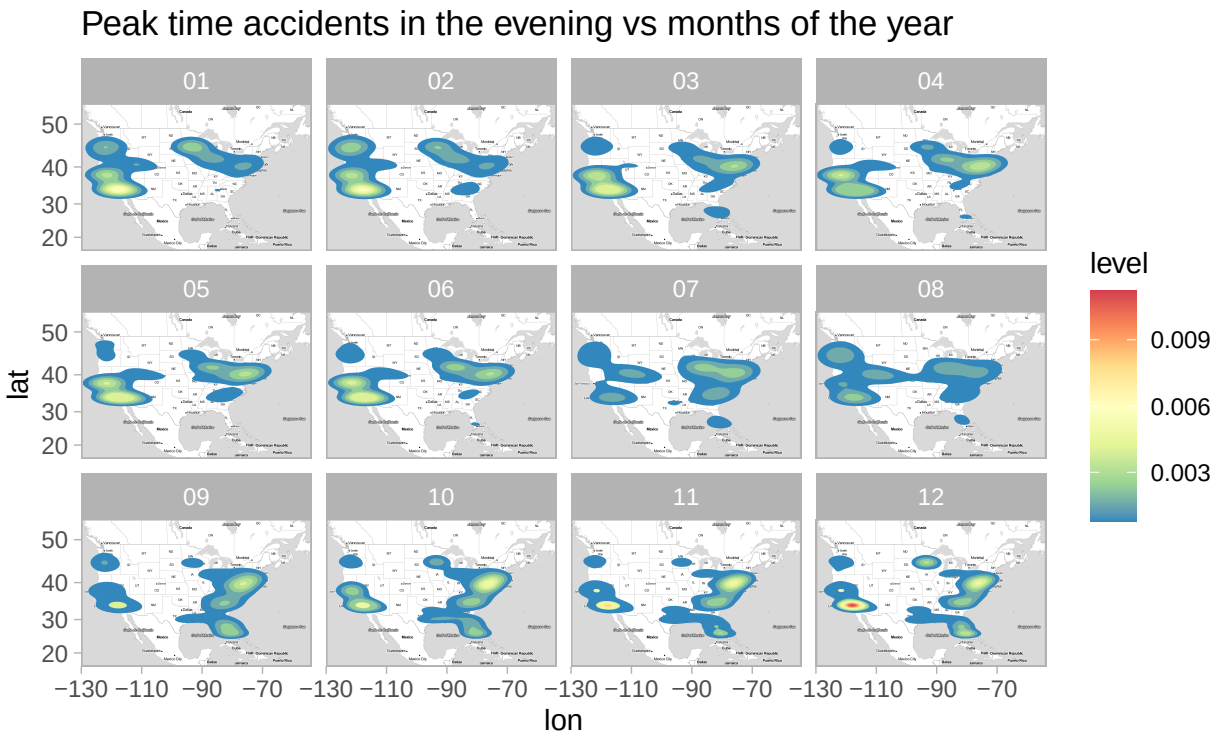


```
ggmap(unitedStatesmap) +
stat_density2d( aes(x = Start_Lng, y = Start_Lat , fill = ..level..),
size = 0.02, bins = 15, data = day,
geom = "polygon") + scale_fill_gradientn(colours=rev(brewer.pal(7,"Spectral")))+
labs(title = " Peak morning hours accidents density") + xlab("longitude") +
  ylab("latitude")
```

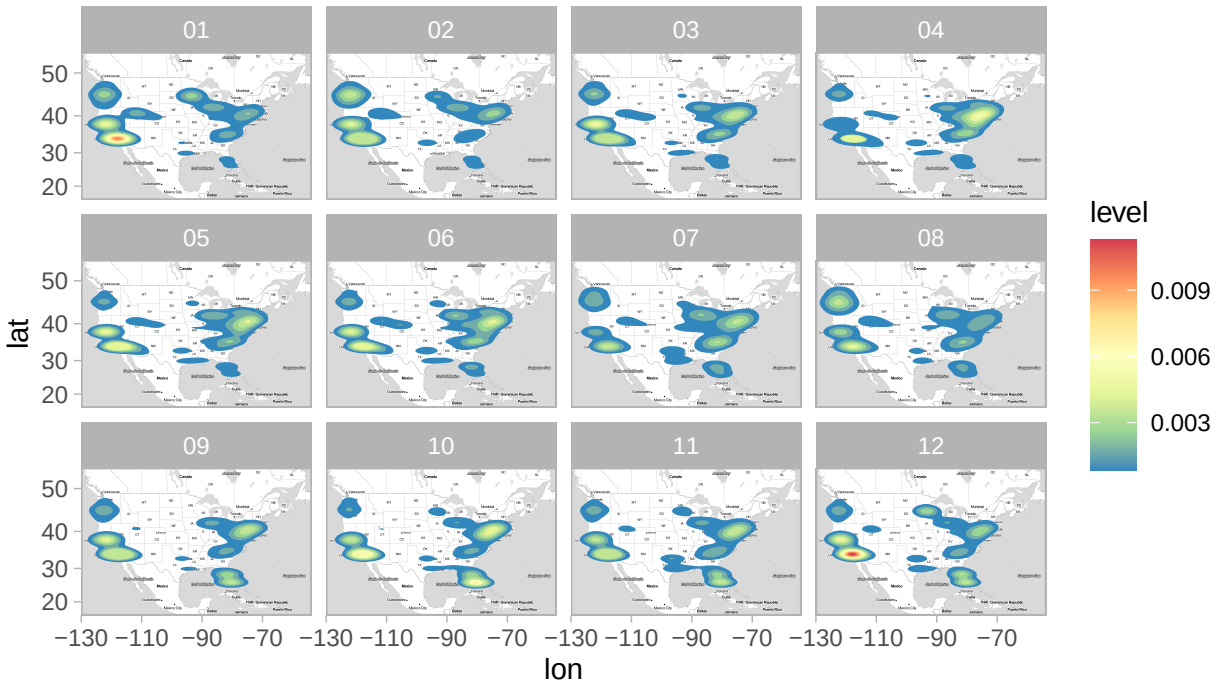



```
# The epicenters for peak time morning and evening accidents are the same, however
# in the night there are more accidents happening around the epicenters than in
# the day

# It means driving at night near the epicenters is more dangerous than in the
# morning
```

```
# ggmap(unitedStatesmap) +
#     geom_point(data = night,aes(x = Start_Lng , y = Start_Lat,
#                                 ), size = 0.001 ,alpha = 1 / 3) +
#   facet_wrap(~ month)
#
#      ggmap(unitedStatesmap) +
#     geom_point(data= day,aes(x = Start_Lng , y = Start_Lat,
#                                 ), size = 0.001 ,alpha = 1 / 3) + facet_wrap(~month)


ggmap(unitedStatesmap) +
stat_density2d( aes(x = Start_Lng, y = Start_Lat , fill = ..level..),
size = 0.02, bins = 15, data = night,
geom = "polygon") + scale_fill_gradientn(colours=rev(brewer.pal(7,"Spectral")))+
labs(title = "Peak time accidents in the evening vs months of the year") + facet_wrap(~ month)
```




```
ggmap(unitedStatesmap) +
stat_density2d( aes(x = Start_Lng, y = Start_Lat , fill = ..level..),
size = 0.02, bins = 15, data = day,
geom = "polygon") + scale_fill_gradientn(colours=rev(brewer.pal(7,"Spectral")))+
labs(title = "Peak time accidents in the morning vs months of the year")+ facet_wrap(~ month)
```



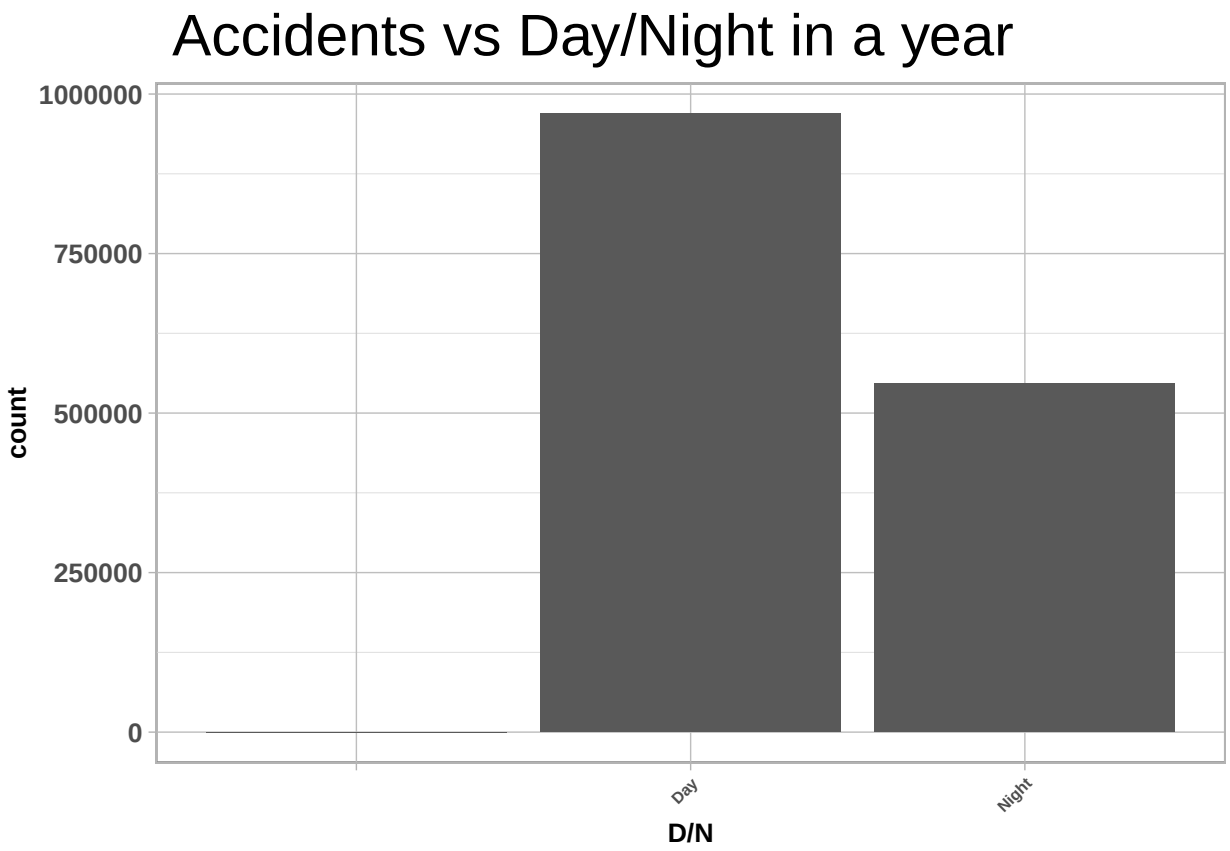

```
##  in the evenings, in the beginning of the year, the accident epicenter is the
# north east and the western states, from July to December it shifts towards
# east and south and California notes the highest number of accidents in December
```

```
## in the mornings, accidents seem to be uniformly distributed in the USA,
## throughout the year
```

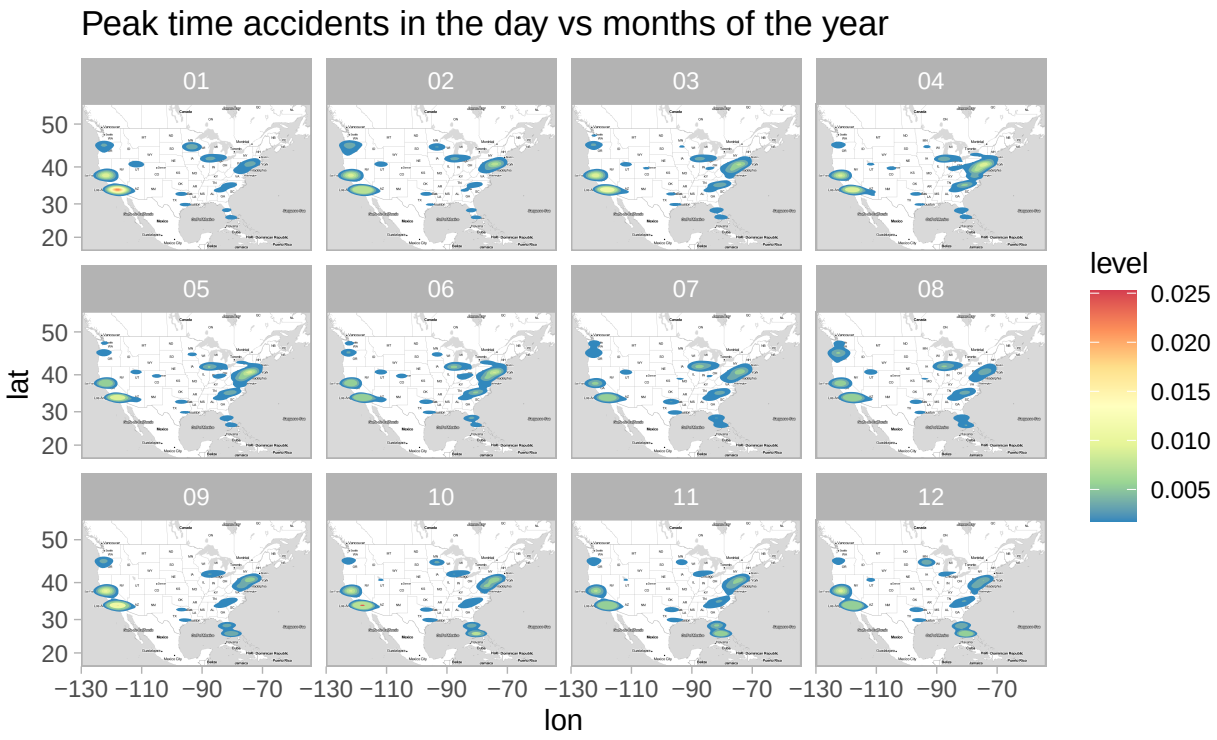
```
time_div %>%
  ggplot(aes(Civil_Twilight  )) +
    geom_histogram(bins = 50,stat = "count" )  +
   labs(title = " Accidents vs Day/Night in a year") + xlab("D/N") + t
```

11.3.0.8 civil twilight study

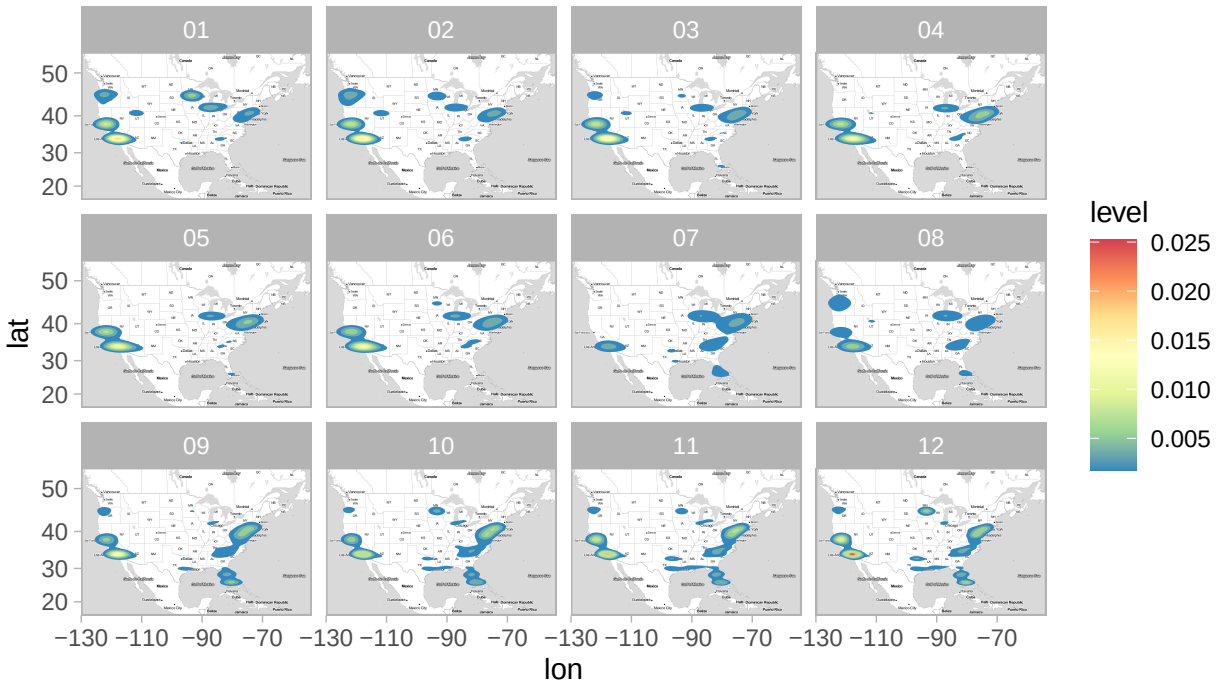
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
daytime <- time_div %>%
    filter(Civil_Twilight == "Day")

nightime <- time_div %>%
    filter(Civil_Twilight == "Night")


ggmap(unitedStatesmap) +
stat_density2d( aes(x = Start_Lng, y = Start_Lat , fill = ..level..),
size = 0.02, bins = 15, data = daytime,
geom = "polygon") + scale_fill_gradientn(colours=rev(brewer.pal(7,"Spectral")))+
labs(title = "Peak time accidents in the day vs months of the year")+ facet_wrap(~ month)
```




```
ggmap(unitedStatesmap) +
stat_density2d( aes(x = Start_Lng, y = Start_Lat , fill = ..level..),
size = 0.02, bins = 15, data = nightime,
geom = "polygon") + scale_fill_gradientn(colours=rev(brewer.pal(7,"Spectral")))+
labs(title = "Peak time accidents in the night vs months of the year")+ facet_wrap(~ month)
```



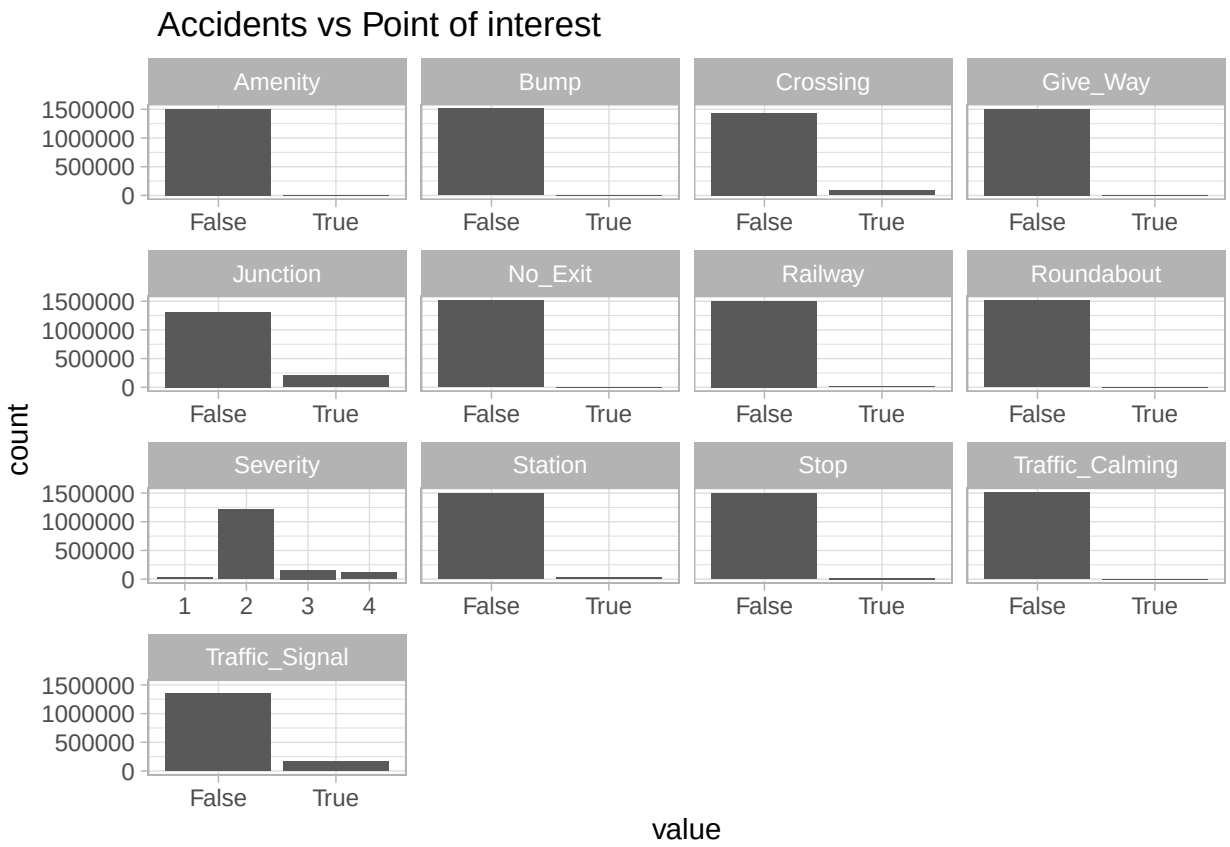

```
#### Based on civil twilight, the environmental illumination in USA cannot be redicted using
```

```
POIs <- proj_df %>%
  select("Severity", "Amenity" ,"Bump", "Crossing", "Give_Way",
         "Junction","No_Exit", "Railway", "Roundabout" , "Station" ,
         "Stop", "Traffic_Calming" , "Traffic_Signal")

ggplot(gather(POIs), aes(value)) +
    geom_histogram(bins = 10,stat = "count") +
    facet_wrap(~ key, scales = 'free_x') +
  labs(title = " Accidents vs Point of interest")
```

11.3.0.9 POIS and accidents

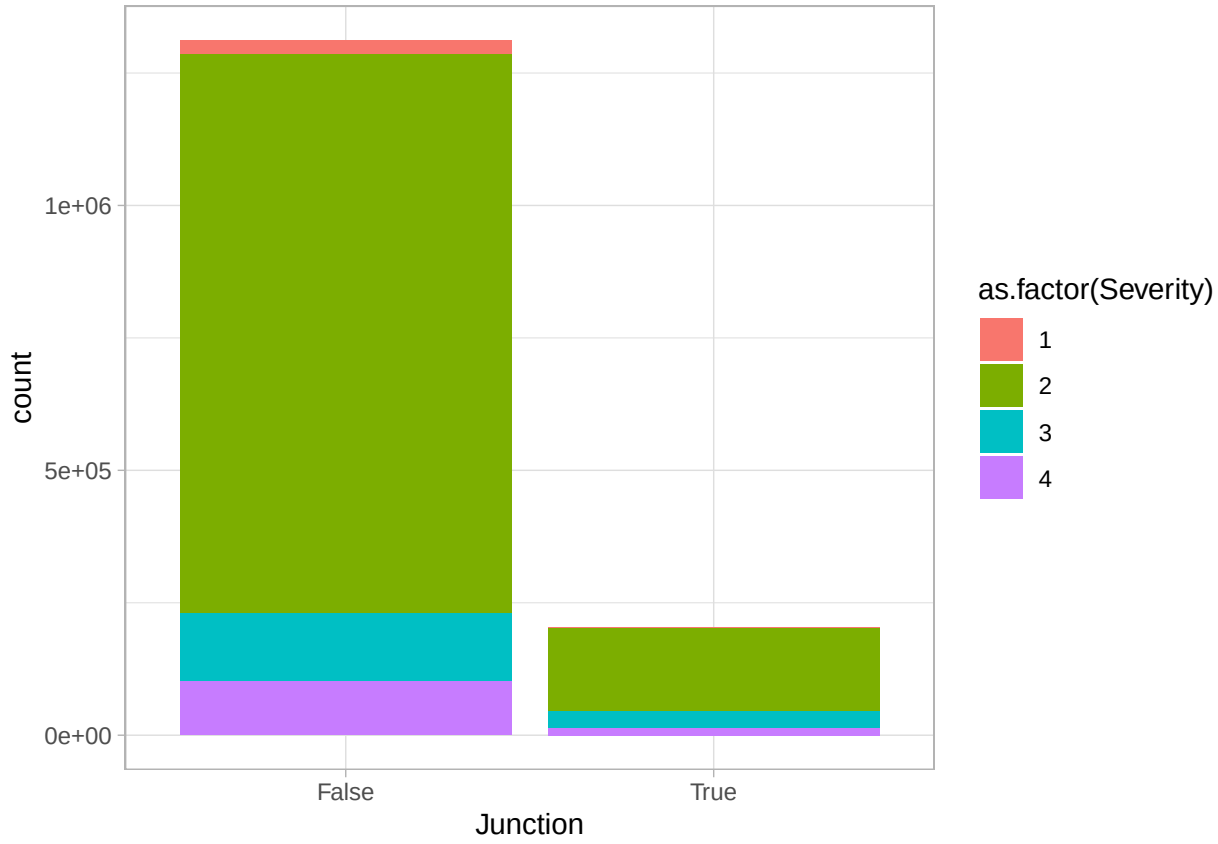
```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



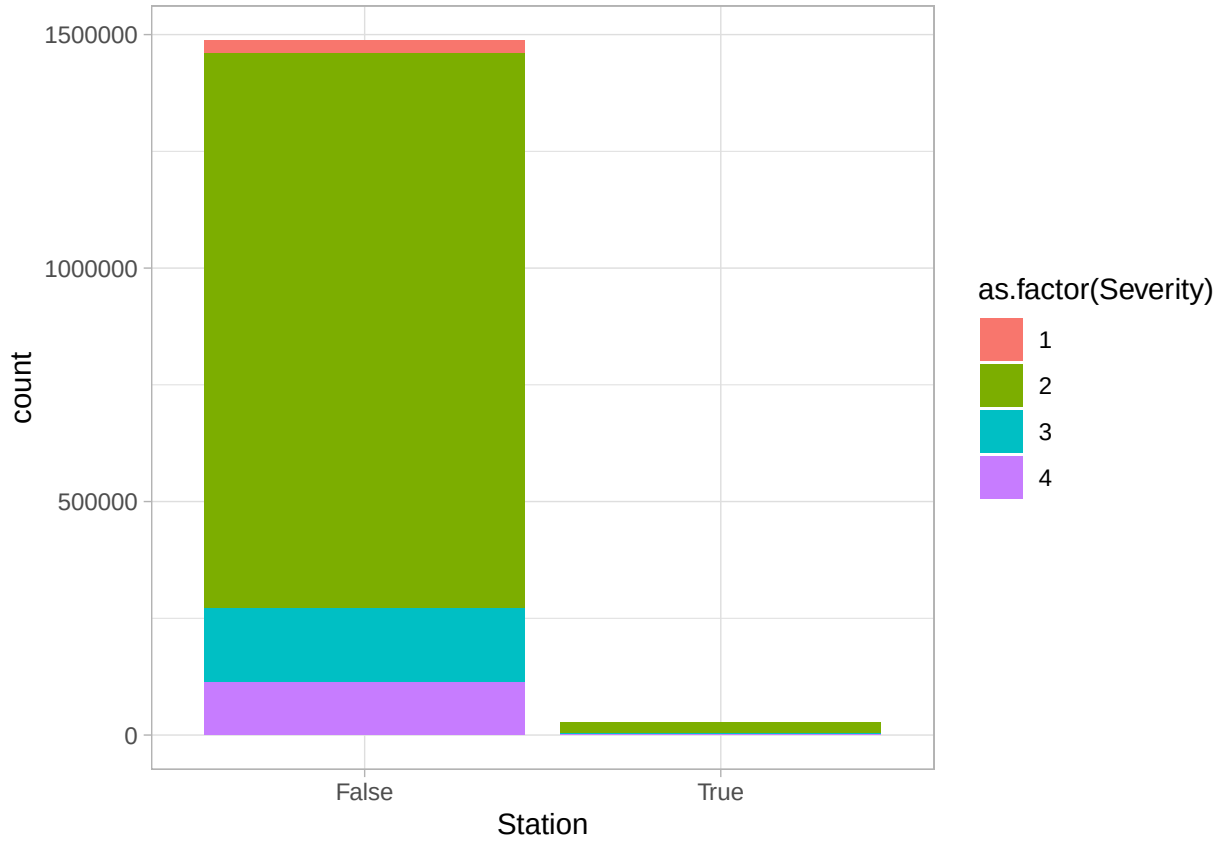

```
## For most accidents, no POIS are present, however,
## substantial accidents happen in
# the presence of crossing, junction , traffic signal and station

# severity ratio of all, versus the ones with these cases

ggplot(POIs, aes(x = Junction, fill = as.factor(Severity))) +
    geom_histogram(bins = 10, stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



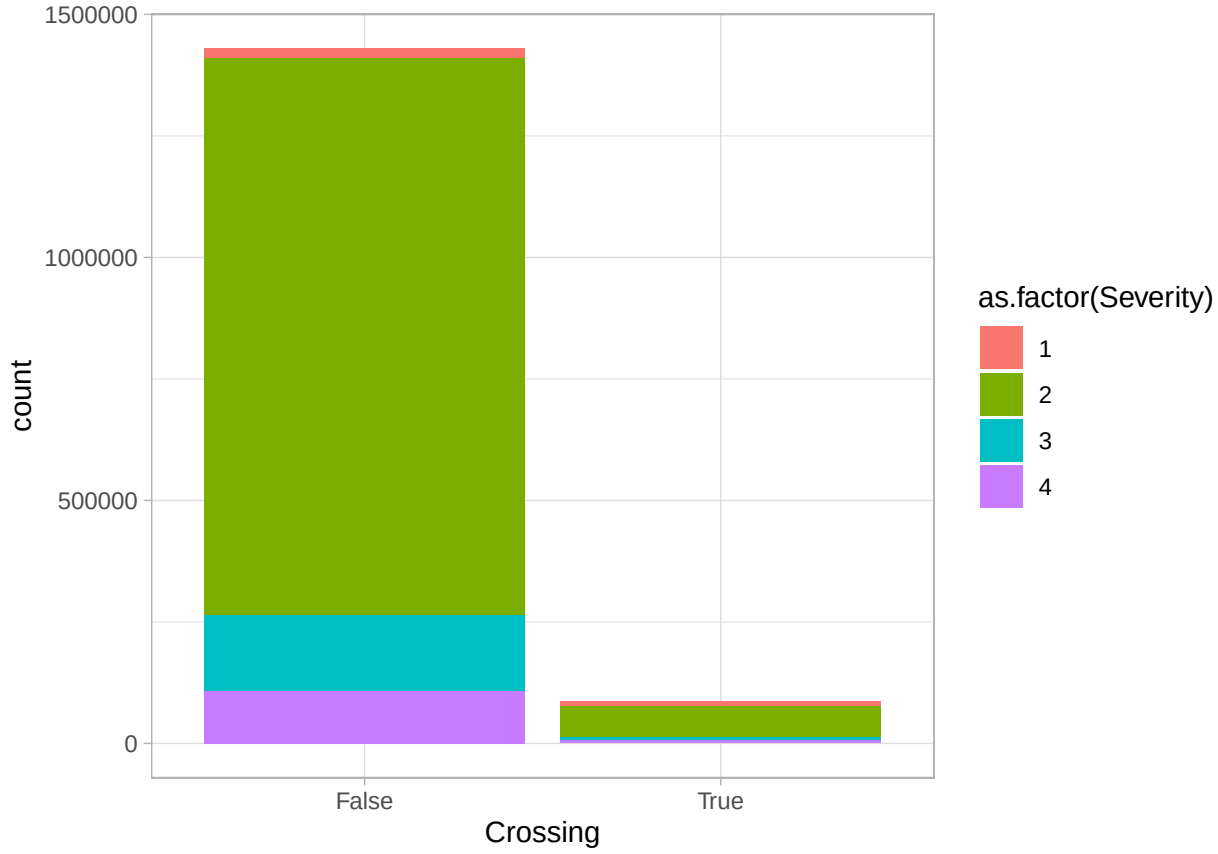

```
ggplot(POIs, aes(x = Station, fill = as.factor(Severity))) +
    geom_histogram(bins = 10, stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



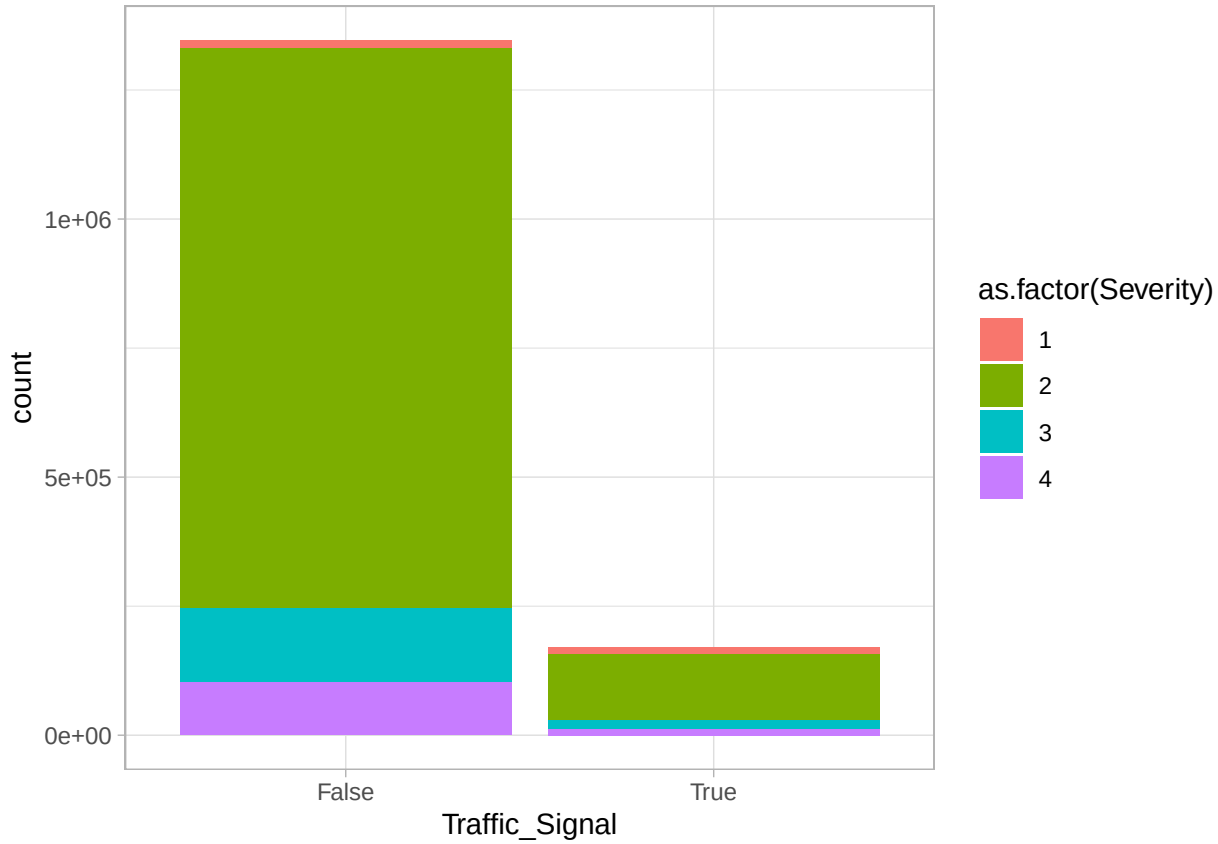
```
ggplot(POIs, aes(x = Crossing, fill = as.factor(Severity))) +
    geom_histogram(bins = 10, stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
ggplot(POIs, aes(x = Traffic_Signal, fill = as.factor(Severity))) +
    geom_histogram(bins = 10, stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
## what proportion of severity 4 accidents happened with these conditions.

POIs %>% filter(Severity == 4) %>%
  group_by(Junction)%>%
  tally()
```

```
## # A tibble: 2 x 2
##   Junction      n
##   <fct>     <int>
## 1 False    101027
## 2 True      13425
```

```
POIs %>% filter(Severity == 4) %>%
  group_by(Station)%>%
  tally()
```

```
## # A tibble: 2 x 2
##   Station      n
##   <fct>    <int>
## 1 False   112777
## 2 True      1675
```

```
POIs %>% filter(Severity == 4) %>%
  group_by(Crossing)%>%
  tally()
```

```
## # A tibble: 2 x 2
##   Crossing      n
```

```
##   <fct>     <int>
## 1 False    108506
## 2 True       5946
```

```
POIs %>% filter(Severity == 4) %>%
  group_by(Traffic_Signal)%>%
  tally()
```

```
## # A tibble: 2 x 2
##   Traffic_Signal      n
##   <fct>           <int>
## 1 False          102649
## 2 True            11803
```

12 Statistical Learning: Modeling & Prediction

```
set.seed(0)
## Predicting the severity of accidents based on POIs combination, that were
## present in accidents
mdl_poi_a <- multinom(Severity ~ Junction + Traffic_Signal + Station +
                  Crossing + Stop , data = proj_df)
```

```
## # weights:  28 (18 variable)
## initial  value 2101710.974353
## iter  10 value 1070841.439228
## iter  20 value 1034767.870428
## iter  30 value 1024017.277781
## final  value 1024017.087195
## converged
```

```
# fit2 <- multinom(Severity ~ Junction + Traffic_Signal + Station +
#                   Crossing + Stop + Bump + Traffic_Calming , data = proj_df)
#
# fit3 <- multinom(Severity ~ Junction * Traffic_Signal * Station *
#                   Crossing * Stop  , data = proj_df)

idx <- createDataPartition(time_div$Severity, p = 0.8, list = FALSE,
)
#Subset the data
#Create training and testing dfs
training <- time_div[idx, ] # use the indices to obtain the ful rows
testing <- anti_join(time_div, training)
```

```
## Joining, by = c("Severity", "year", "month", "day", "start_time", "End_Time", "Start_Lat", "
```

```
confusionMatrix(predict(mdl_poi_a, testing),as.factor(testing$Severity))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction      1      2      3      4
##          1      0      0      0      0
##          2   5026 207552  30877  21607
##          3      0      0      0      0
##          4      0      0      0      0
##
## Overall Statistics
##
##                Accuracy : 0.783
##                  95% CI : (0.7815, 0.7846)
##     No Information Rate : 0.783
##     P-Value [Acc > NIR] : 0.5011
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity           0.00000    1.000   0.0000  0.00000
## Specificity           1.00000    0.000   1.0000  1.00000
## Pos Pred Value            NaN    0.783      NaN      NaN
## Neg Pred Value        0.98104      NaN   0.8835  0.91848
## Prevalence            0.01896    0.783   0.1165  0.08152
## Detection Rate        0.00000    0.783   0.0000  0.00000
## Detection Prevalence  0.00000    1.000   0.0000  0.00000
## Balanced Accuracy     0.50000    0.500   0.5000  0.50000
```

```
## Predicting the severity of accidents based on days of the week, months of the year
idx <- createDataPartition(time_div$Severity, p = 0.8, list = FALSE,
)
#Subset the data
#Create training and testing dfs
training <- time_div[idx, ] # use the indices to obtain the ful rows
testing <- anti_join(time_div, training)
```

```
## Joining, by = c("Severity", "year", "month", "day", "start_time", "End_Time", "Start_Lat", "
```

```
mdl_time_a <- multinom(Severity ~ as.factor(month) , data = time_div)
```

```
## # weights:  52 (36 variable)
## initial  value 2101710.974353
## iter  10 value 1026090.156209
## iter  20 value 1024284.899833
## iter  30 value 997266.568280
## iter  40 value 964054.152474
## iter  50 value 957336.004210
## iter  60 value 957180.815705
```

```
## iter  70 value 957157.010303
## final  value 957156.359221
## converged
```

```
confusionMatrix(predict(mdl_time_a, testing),as.factor(testing$Severity))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      1      2      3      4
##          1      0      0      0      0
##          2   4973 207276  30821  21701
##          3      0      0      0      0
##          4      0      0      0      0
##
## Overall Statistics
##
##                Accuracy : 0.7829
##                  95% CI : (0.7813, 0.7844)
##     No Information Rate : 0.7829
##     P-Value [Acc > NIR] : 0.5011
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity           0.00000   1.0000   0.0000  0.00000
## Specificity           1.00000   0.0000   1.0000  1.00000
## Pos Pred Value            NaN   0.7829      NaN      NaN
## Neg Pred Value        0.98122      NaN   0.8836  0.91804
## Prevalence            0.01878   0.7829   0.1164  0.08196
## Detection Rate        0.00000   0.7829   0.0000  0.00000
## Detection Prevalence  0.00000   1.0000   0.0000  0.00000
## Balanced Accuracy     0.50000   0.5000   0.5000  0.50000
```

```
# mdl_time_b <- multinom(Severity ~ as.factor(month) + as.factor(day) , data = time_div)
# summary(mdl_time_b)
# confusionMatrix(predict(mdl_time_b, testing),as.factor(testing$Severity))


# mdl_time_poi <- multinom(Severity ~ as.factor(month) + as.factor(day) +Junction + Traffic_S
# summary(mdl_time_poi)
#
# confusionMatrix(predict(mdl_time_poi, testing),as.factor(testing$Severity))
```

- DSCI 451 will accomplish at least 1 simple linear model (or simple logistic model)
- DSCI 352/352M/452 requires the appropriate modeling for your data set including machine learning

- Types of modeling to try
- Statistical prediction/modeling
- Model selection
- Cross-validation, Predictive R2
- Interpret results
- Challenge results

13 Discussion

Discussion of the answers to the data science questions framed in the introduction

- The dataset was not perfect, there wasn't much data from the center part of the USA, and most of the data came from the the state of California in 2020.
- Many of the results were predictable
- Car accidents peaked during peak hours.
- More accidents happened in the evenings during peak hours than in the morning.
- Car accidents increased at the end of the year.
- Car accidnets reduced in the weekends
- Some results were new
- At the half way point of the year, the epicenter of peak accidnets in the evening shifts from north east to south east.
- For a significant number of accidents Junctions, Stations, Railways and traffic signals were present.
- Could not use the weather conditions to predict, because i was not sure how to use the information available.
- The model we used to predict severity
- could only predict severity 2 because most ofthe data was severity 2.
- we made models using both important POIS and time parameters
- the results were the same, 78.29 % for each case, only severity 2 was classified correctly.
- Since this is a multifactor problem, it was very difficult to find a direct correlation between predictor and severity, also the data was pretty skewed.

14 Conclusions

We were succefully able to use the data science techniques learnt in class to wrangle and visualise thislarge dataset, and also build a primitive logistic model from it.

15 Acknowledgments

16 References

- Kaggle : https://www.kaggle.com/sobhanmoosavi/us-accidents/tasks?taskId=189