Background

We're going to explore a large data set or traffic crashes to learn about what factors are connected with injuries. We will use data from the city of Chicago's open data portal. (This activity is derived from a blog post by Julia Silge)

```
years_ago <- mdy("01/01/2022") # data from last 2 years. May take time to load!
crash_url <- glue::glue("https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3i
crash_raw <- as_tibble(read.socrata(crash_url)) # a new way to read in data, don't worry about it!
```

This dataset is pretty crazy! Take a look at it in the viewer, and then let's do some data munging to get it into a nicer form.

-create a variable called `injuries` which indicates if the crash involved injuries or not. -create an unknown category for missing `report_types` -decide which other variables to keep

```
crash <- crash_raw %>%
  arrange(desc(crash_date)) %>%
  transmute(
    injuries = as.factor(if_else(injuries_total > 0, "injuries", "none")),
    crash_record_id, crash_date, weather_condition, lighting_condition, first_crash_type, roadway_surfa
    latitude, longitude
  )
crash_raw <- crash_raw %>% mutate(injuries = ifelse(injuries_total > 0, "injuries", "none"))
```

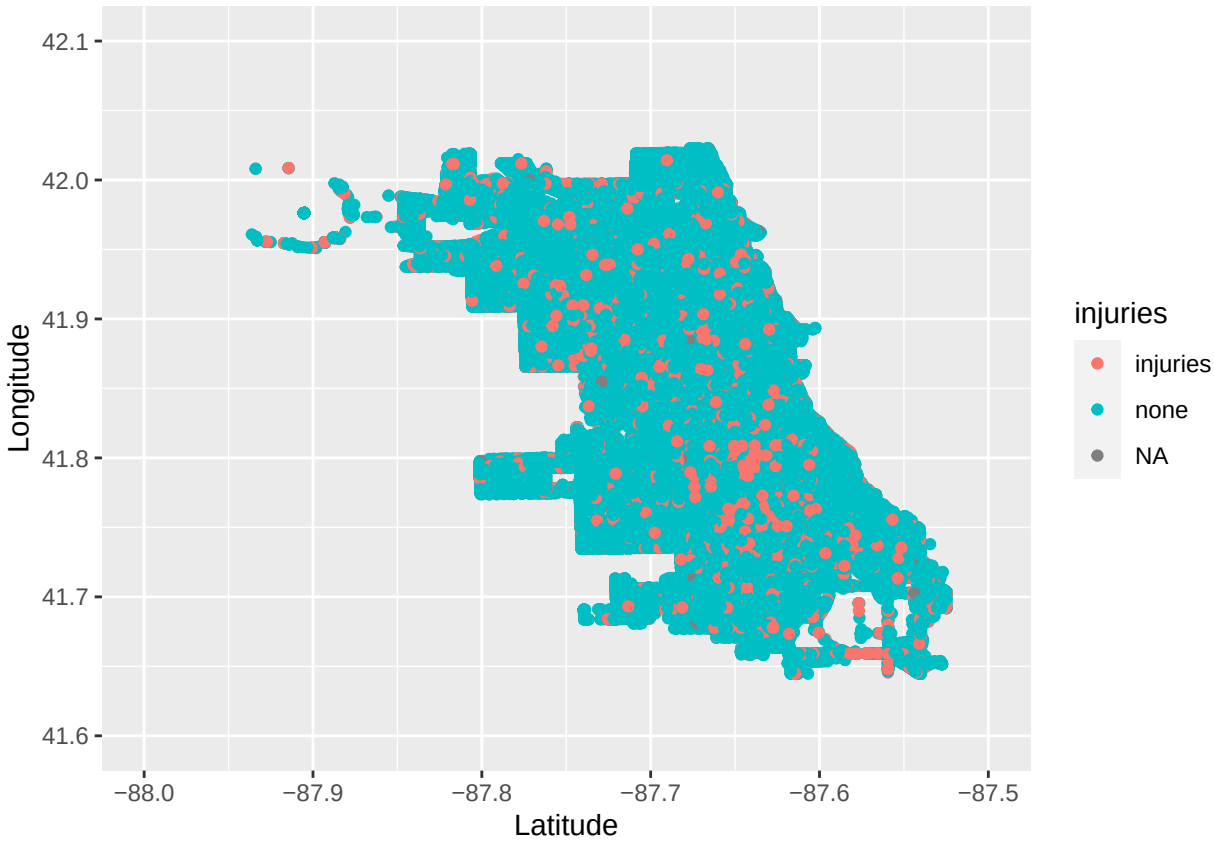
Exploratory Data Analysis

Here's a few questions to get you started.

1. Take a look at crashes by latitude and longitude, colored by injuries. What do you notice?
2. What are the most common contributing factors to a crash?
3. How do crashes vary month by month? Compare crashes by month in 2022 to 2023.
4. Are crashes more likely to cause injuries when it is rainy and dark? Use the variables `weather_condition` and `lighting_condition` to explore.
5. Choose a question you want to explore, and create an appropriate visual.

```
ggplot(crash, aes(x = longitude, y = latitude, color = injuries)) +
  geom_point() +
  xlab("Latitude") +
  ylab("Longitude") +
  xlim(-88, -87.5) +
  ylim(41.6, 42.1)
```

```
## Warning: Removed 2132 rows containing missing values (`geom_point()`).
```




```
  ggtitle("Longitude vs. Latitude")
```

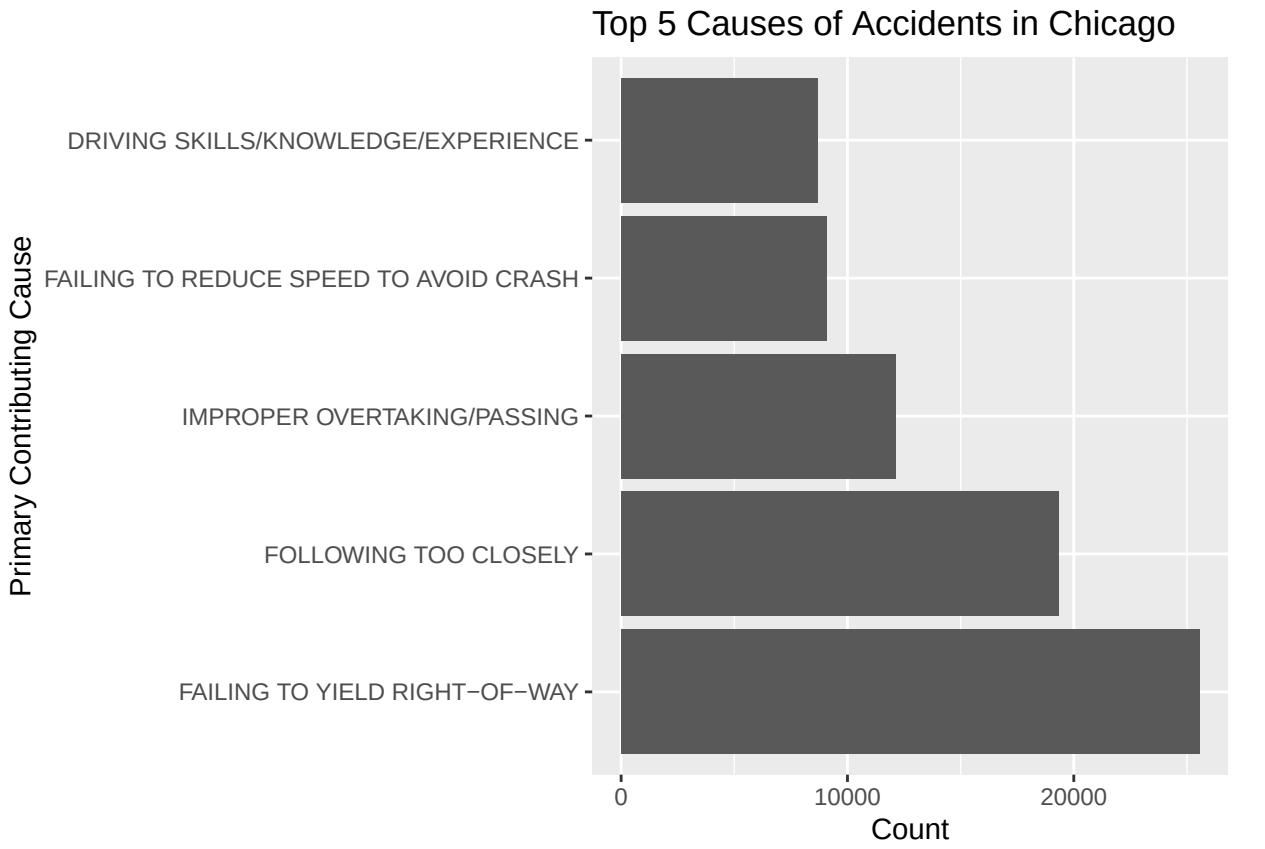
```
## $title
## [1] "Longitude vs. Latitude"
##
## attr(,"class")
## [1] "labels"
```

```
crash_filtered <- crash %>% filter(!(prim_contributory_cause %in% c('NOT APPLICABLE', 'UNABLE TO DETERM
frequency_table <- table(crash_filtered$prim_contributory_cause)
top_5_items <- names(sort(frequency_table, decreasing = TRUE)[1:5])
top5_df <- data.frame(top_5_items)

crash_filtered <- crash_filtered %>% filter(prim_contributory_cause %in% c('FAILING TO YIELD RIGHT-OF-W

#don't know what's happening right now... is.factor(prim_contributory_cause) is returning false, and it
#this is making me lose my mind it's just not right
crash_filtered$prim_contributory_cause <- factor(
  crash_filtered$prim_contributory_cause,
    levels = c(
      'FAILING TO YIELD RIGHT-OF-WAY',
      'FOLLOWING TOO CLOSELY',
      'IMPROPER OVERTAKING/PASSING',
      'FAILING TO REDUCE SPEED TO AVOID CRASH',
      'DRIVING SKILLS/KNOWLEDGE/EXPERIENCE'
    ),
  ordered = TRUE
  )
```

```
ggplot(crash_filtered, aes(y = prim_contributory_cause)) +
  geom_bar() +
  ylab("Primary Contributing Cause") +
  xlab("Count") +
  ggtitle("Top 5 Causes of Accidents in Chicago")
```




```
year = year(crash$crash_date)
crash = mutate(crash, year = year(crash_date))
crash_graph <- filter(crash, year == 2023 | year == 2022)
```

```
crash_graph <- crash_graph %>%
  mutate(
    crash_year = as.factor(year(crash_date)),
    month_name = month(crash_month, label = TRUE, abbr = TRUE)
  )
crash_graph <- crash_graph %>% drop_na(crash_year)
crash_graph_df <- crash_graph %>%
  group_by(crash_year, month_name) %>%
  summarize(Injury_count = sum(injuries == 'injuries'))
```

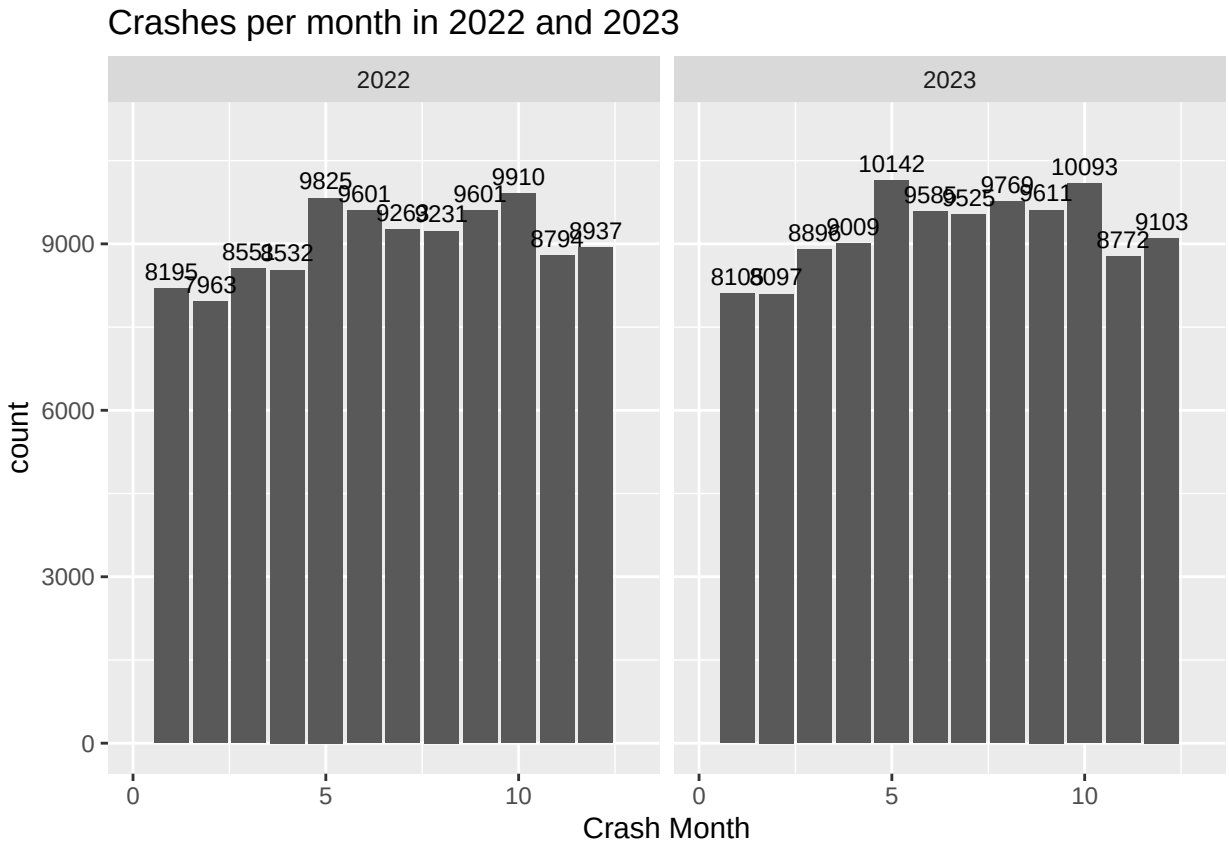
```
## `summarise()` has grouped output by 'crash_year'. You can override using the
## `.groups` argument.
```

```
ggplot(crash_graph %>% filter(year == 2022 | year == 2023), aes(x = crash_month)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  xlab("Crash Month") +
  facet_wrap(~year) +
```

```
  scale_x_continuous(breaks = 1:12, labels = month.abb) +
  xlim(0, 13) +
  ylim(0, 11000) +
  ggtitle("Crashes per month in 2022 and 2023")
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



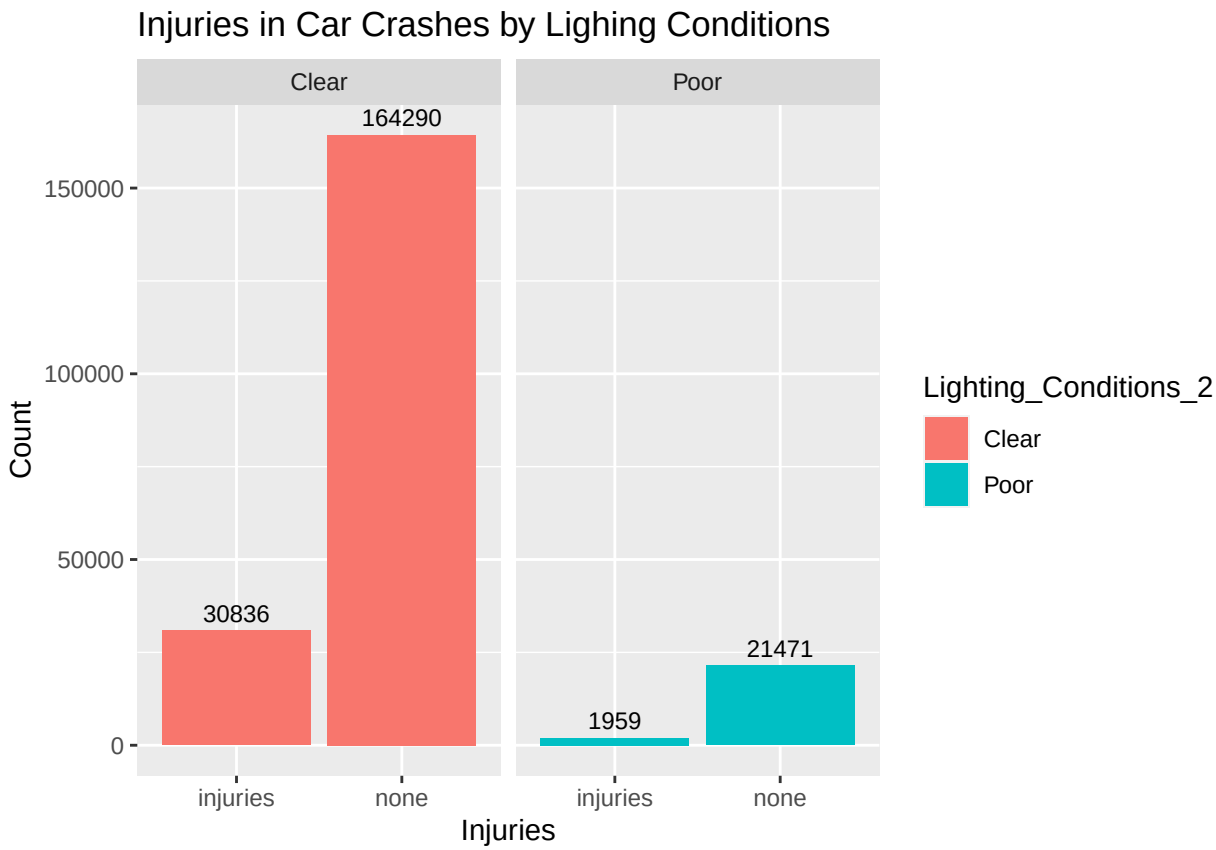

```
crash_22 <- crash_graph %>% filter(year == 2022)
Count_2022 <- crash_22 %>%
  group_by(crash_month) %>%
  summarise(Count_2022 = n())
crash_23 <- crash_graph %>% filter(year == 2023)
Count_2023 <- crash_23 %>%
  group_by(crash_month) %>%
  summarise(Count2023 = n())
g3_df <- merge(Count_2022, Count_2023, by = 'crash_month', all = FALSE)
```

##second attempt at question 4

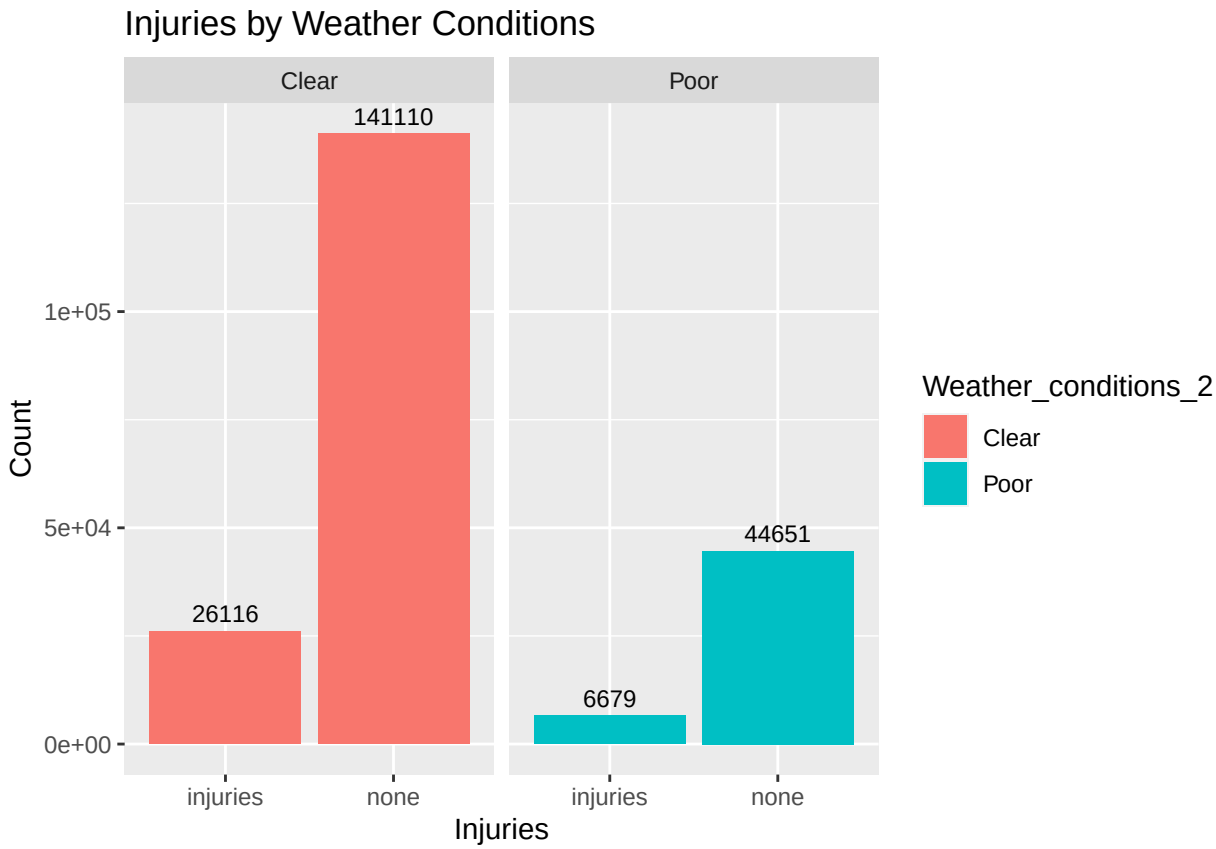
```
crash_graph <- crash_graph %>% mutate(Weather_conditions_2 = ifelse(weather_condition == 'CLEAR', 'Clea
crash_graph <- crash_graph %>% mutate(Lighting_Conditions_2 = ifelse(lighting_condition %in% c('DAWN',
```

##by lighting conditions

```
crash_graph <- crash_graph %>%
  drop_na(injuries)
ggplot(crash_graph, aes(x = injuries, fill = Lighting_Conditions_2)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  xlab("Injuries") +
  ylab("Count") +
  facet_wrap(~Lighting_Conditions_2) +
  ggtitle("Injuries in Car Crashes by Lighing Conditions")
```



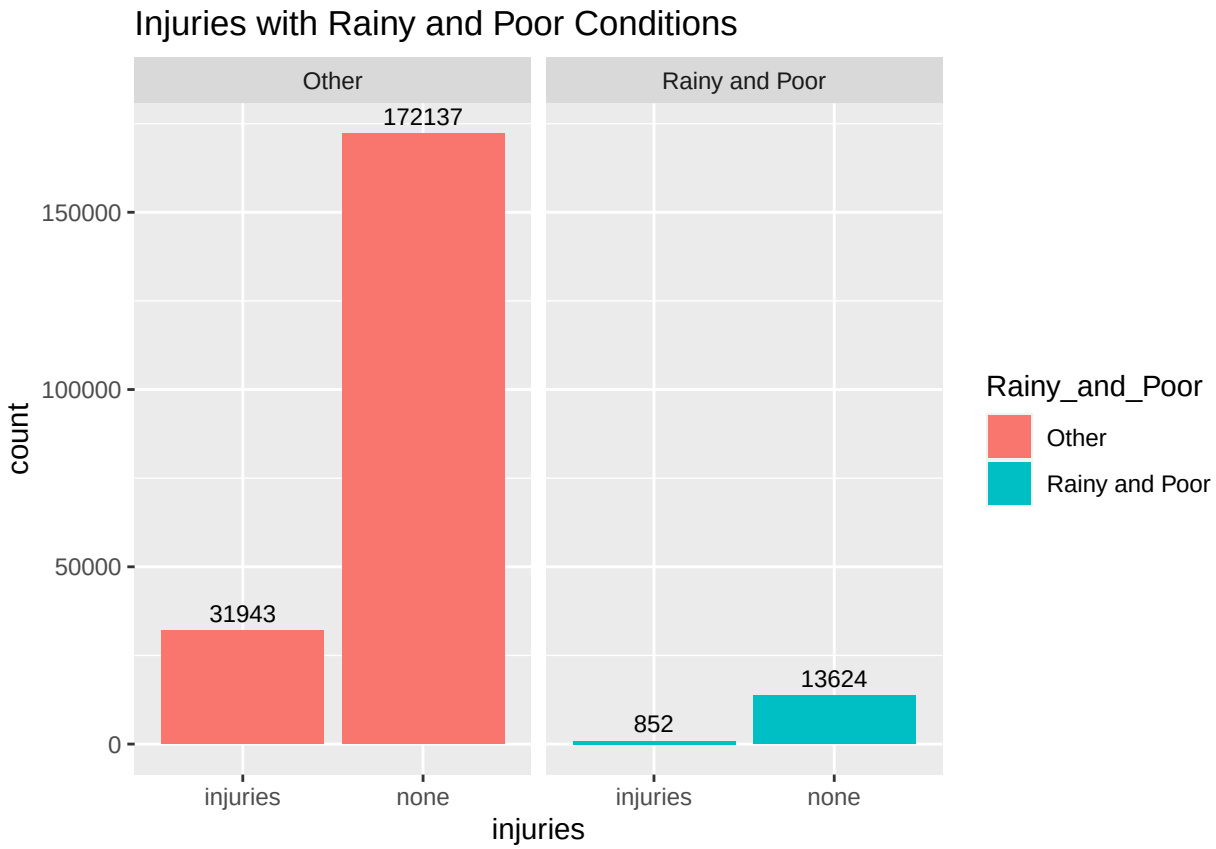

```
ggplot(crash_graph, aes(x = injuries, fill = Weather_conditions_2)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  xlab("Injuries") +
  ylab("Count") +
  facet_wrap(~Weather_conditions_2) +
  ggtitle("Injuries by Weather Conditions")
```



##graph of conditions by both

```
crash_graph <- crash_graph %>%
  mutate(Rainy_and_Poor = ifelse((Weather_conditions_2 == 'Poor' & Lighting_Conditions_2 == 'Poor'), 'Ra
```

```
ggplot(crash_graph, aes(x = injuries, fill = Rainy_and_Poor)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  facet_wrap(~Rainy_and_Poor) +
  ggtitle("Injuries with Rainy and Poor Conditions")
```



##exploring my own question What were the most common types of crashes, and did those lead to injuries often?

```
frequency_table_2 <- table(crash_filtered$first_crash_type)
print(frequency_table_2)
```

```
##
##                          ANGLE                         ANIMAL
##                           9522                              8
##                  FIXED OBJECT                       HEAD ON
##                           1641                            339
##            OTHER NONCOLLISION                   OTHER OBJECT
##                             60                            262
##                     OVERTURNED          PARKED MOTOR VEHICLE
##                             18                           7822
##                   PEDALCYCLIST                     PEDESTRIAN
##                           1291                           1998
##                       REAR END                  REAR TO FRONT
##                          23131                            407
##                   REAR TO REAR                   REAR TO SIDE
##                             94                            463
## SIDESWIPE OPPOSITE DIRECTION     SIDESWIPE SAME DIRECTION
##                            660                          14074
##                        TURNING
##                          12975
```

```
graph5_df <- crash_graph %>% filter(first_crash_type %in% c('SIDESWIPE SAME DIRECTION', 'TURNING', 'ANG
```

```r
graph5_df <- graph5_df %>%
  drop_na(injuries)
ggplot(graph5_df, aes(x = injuries, fill = first_crash_type)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  facet_wrap(~first_crash_type) +
  ylim(0, 45000) +
  ggtitle("Injuries by crash type")
```