Stats 306 Project - Analysis of Traffic Accidents

University of Michigan, Fall 2021

Team members:

- Andy Lee (sunylee)
- Olivia Lee (oliviale)
- Malachi Graybill (malachig)
- Mahdi Qazwini (mqazwini)
- Vishnu Nair (nairv)

All team members contributed equally in this project.

Introduction

Over 36,000 Americans died from car crash fatalities in the year 2019. In fact, based on the National Highway Traffic Safety Administration's 2019 crash fatality data, the fatality rate per 100 million decreased by 3.5 percent from 1.14 to 1.10. Nevertheless, injury rates per 100 million have stagnated (NHTSA data). As the data suggests, while fatality rates have decreased, albeit minutely, injury rates have failed to improve. This notion drives an impetus for research that strives to decrease the number of car accidents.

The Chicago Dataset offers an insight into the causes behind car fatalities and injuries in 2019. The following investigation will explore four primary concerns: the chief causes of accidents, the impact of the timing of the incident on the frequency of accidents, the conditions in which injuries are most likely to occur, and the types of accidents that are most likely to incur a hit and run situation. We suspect that incidents with no reports to the accident or those with longer response times may correspond to hit and runs. Road conditions such as inclement weather, surface deterioration, defects, and travel across work zones may impact injury and fatality rates by inducing a difficult driving environment. We must then also question if these conditions have a greater impact largely due to higher city speed limits. Moreover, driver awareness can vary across the time of day, week, month, or even year.

With an analysis of these factors, we hope to propose a solution to mitigating the total number of accidents, which will in turn reduce the number of injuries and fatalities. We will proceed with an in-depth review of the primary and secondary causes of accidents after loading and preparing the data.

Packages Required

To run the code from this notebook, these packages must be loaded.

```
In [2]: library(tidyverse)  # Data manipulation
        library(lubridate)  # Dates
        library(cowplot)    # Plot manipulation
        library(janitor)    # Row and column manipulation
        library(ggmap)      # Maps
```

Data Preparation

Importing Data

The data used for this project was collected and published by the City of Chicago. The dataset's page can be found here and the csv file can be found here.

We are using a cleaned subset of this dataset provided by Professor Kean Ming Tan. However, we will also import the full dataset because some variables we are using are missing from the provided subset.

```
In [4]: # Load dataset
        data = read_csv("chicagotraffic2019.csv")

        # Load full dataset
        full_data = read_csv("https://data.cityofchicago.org/api/views/85ca-t3if/rows.csv")
```

Cleaning Data

With our dataset loaded, we now need to transform it to fit our needs. This includes removing variables we won't need or repetative columns and transforming the types of variables (e.g. date strings to date objects).

The data is already tidy so no extra work is needed in that regard. Tidy data has the property that each variable has its own column, each observation has its own row, and each value has its own cell. Because our dataset already follows these three properties, it is tidy. For example, if our dataset had SPEED_LIMIT_30, SPEED_LIMIT_45, etc., with true/false as the values, we would want to combine these into SPEED_LIMIT where the speed limit is the value. The only issue present in our dataset that is somewhat related to that is having repetative columns (there exists CRASH_DATE, CRASH_HOUR, CRASH_DAY_OF_WEEK, and CRASH_MONTH, where the latter three are unnecessary because that information can be extracted from CRASH_DATE alone), which we solved by simply not including the duplicates.

```
In [6]: # Cleaned data
        cdata = data %>%
            # Selecting only variables we want
            select(CRASH_DATE, DATE_POLICE_NOTIFIED,
                   POSTED_SPEED_LIMIT,
                   WEATHER_CONDITION, LIGHTING_CONDITION, ROADWAY_SURFACE_COND,
                   DAMAGE, PRIM_CONTRIBUTORY_CAUSE, SEC_CONTRIBUTORY_CAUSE,
                   INJURIES_TOTAL, INJURIES_FATAL, INJURIES_INCAPACITATING,
                   LATITUDE, LONGITUDE,
                   CRASH_RECORD_ID) %>%
            # Converting string date to datetime object
            mutate(CRASH_DATE = mdy_hms(CRASH_DATE),
                   DATE_POLICE_NOTIFIED = mdy_hms(DATE_POLICE_NOTIFIED))
```

Here is a list of the variables we will be using from the dataset:

- **CRASH_DATE**: Date and time of crash
- **DATE_POLICE_NOTIFIED**: Date and time on which police were notified of the crash
- **WEATHER_CONDITION**: Weather condition at time of crash
- **LIGHTING_CONDITION**: Light condition at time of crash
- **FIRST_CRASH_TYPE**: Type of first collision in crash
- **ROADWAY_SURFACE_COND**:
- **PRIM_CONTRIBUTORY_CAUSE**: The factor which was most significant in causing the crash, as determined by officer judgment
- **SEC_CONTRIBUTORY_CAUSE**: The factor which was second most significant in causing the crash, as determined by officer judgment
- **INJURIES_TOTAL**: Total persons sustaining fatal, incapacitating, non-incapacitating, and possible injuries
- **INJURIES_FATAL**: Total persons sustaining fatal injuries in the crash
- **INJURIES_INCAPACITATING**: Total persons sustaining incapacitating/serious injuries in the crash
- **LATITUDE**: The latitude of the crash location
- **LONGITUDE**: The longitude of the crash location
- **CRASH_RECORD_ID**: An id that uniquely identifies each car crash incident

and from the full dataset:

- **HIT_AND_RUN_I**: Crash did/did not involve a driver who caused the crash and fled the scene without exchanging information and/or rendering aid
- **NOT_RIGHT_OF_WAY_I**: Whether the crash begun or first contact was made outside of the public right-of-way

Analysis

After we have prepared our data and cleaned it, we are ready to begin analyzing it.

Causes of Accidents

Perhaps one of the most important questions related to car accidents is, "What causes them?" If we are able to determine the causes of accidents, we will be able to leverage that information to combat these issues.

In this section, we will examine different causes that the crashes were involved in from a micro level. Then, we will also look into the causes of accidents on a macro scale, such as by looking at the frequency of accidents on different months of the year.

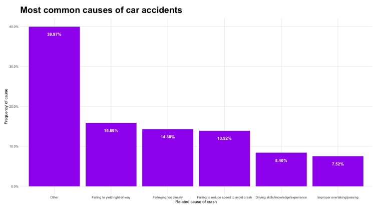
First, empty values for the primary and secondary contributory causes will be filtered out.

```
In [28]: coaData = cdata %>%
             filter(!(PRIM_CONTRIBUTORY_CAUSE %in% c("UNABLE TO DETERMINE", "NOT APPLICABLE")),
                    !(SEC_CONTRIBUTORY_CAUSE %in% c("UNABLE TO DETERMINE", "NOT APPLICABLE")))
```

To get a quick look into the causes accidents, we will plot a histogram of the relative frequencies of the top five causes. We will merge the primary with the secondary causes to see which causes are most crashes involved in.

```
In [32]: causes_data = coaData %>%
             gather("CAUSE_TYPE", "CRASH_CAUSE", c("PRIM_CONTRIBUTORY_CAUSE", "SEC_CONTRIBUTORY_CAUSE")) %>%
             group_by(CRASH_CAUSE) %>%
             count() %>%
             ungroup() %>%
             mutate(freq = n / sum(n))
```

```
In [33]: # Set plot dimensions
         options(repr.plot.width = 14, repr.plot.height = 8)

         # Plot our histogram
         causes_data %>%
             mutate(CRASH_CAUSE = ifelse(rank(desc(freq)) <= 5, str_to_sentence(CRASH_CAUSE), "Other")) %>%
             group_by(CRASH_CAUSE) %>%
             summarize(freq = sum(freq)) %>%
             ungroup() %>%
             ggplot(aes(x = reorder(CRASH_CAUSE, -freq), y = freq)) +
             geom_col(fill = "purple") +
             geom_text(color = "white", size = 4, fontface = "bold", aes(label = scales::percent(freq)), vjust = 3) +
             scale_y_continuous(labels = scales::percent, limits = c(0.0, 0.4)) +
             labs(title = "Most common causes of car accidents",
                  x = "Related cause of crash", y = "Frequency of cause") +
             theme_minimal() +
             theme(panel.grid.minor = element_blank(),
                   plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30)))
```



And here are all of them:

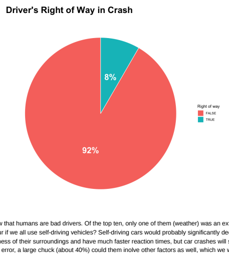
```
In [77]: causes_data %>%
             select(Cause = CRASH_CAUSE, Frequency = freq) %>%
             arrange(desc(Frequency)) %>%
             mutate(Cause = str_to_sentence(Cause), Frequency = scales::percent(Frequency))
```

A tibble: 38 × 2

Cause	Frequency
<chr>	<chr>
Failing to yield right-of-way	15.8853%
Following too closely	14.3002%
Failing to reduce speed to avoid crash	13.9237%
Driving skills/knowledge/experience	8.4014%
Improper overtaking/passing	7.5216%
Improper lane usage	6.1088%
Improper turning/no signal	5.4351%
Improper backing	4.5990%
Weather	4.3869%
Disregarding traffic signals	2.6750%
Operating vehicle in erratic, reckless, careless, negligent or aggressive manner	2.4947%
Disregarding stop sign	1.6644%
Under the influence of alcohol/drugs (use when arrest is effected)	1.3474%
Distraction - from inside vehicle	1.2919%
Vision obscured (signs, tree limbs, buildings, etc.)	1.2523%
Physical condition of driver	1.1175%
Equipment - vehicle condition	0.9313%
Driving on wrong side/wrong way	0.9234%
Exceeding safe speed for conditions	0.7430%
Distraction - from outside vehicle	0.7113%
Road engineering/surface/marking defects	0.6321%
Exceeding authorized speed limit	0.5806%
Road construction/maintenance	0.5073%
Disregarding other traffic signs	0.4359%
Disregarding road markings	0.4280%
Cell phone use other than texting	0.3487%
Had been drinking (use when arrest is not made)	0.2754%
Evasive action due to animal, object, nonmotorist	0.1962%
Animal	0.1625%
Related to bus stop	0.1605%
Turning right on red	0.1486%
Distraction - other electronic device (navigation device, dvd player, etc.)	0.1229%
Texting	0.0832%
Disregarding yield sign	0.0674%
Obstructed crosswalks	0.0555%
Bicycle advancing legally on red light	0.0416%
Passing stopped school bus	0.0218%
Motorcycle advancing legally on red light	0.0178%

It is apparent that the most common causes of car crashes are due to human error and not the cause of some external factor. Most of them are silly human errors. If we look into the right of way data, we find that most of these crashes occur because the driver did not have right of way.

```
In [36]: # Get right of way data
         row_data = full_data %>%
             select(CRASH_RECORD_ID, NOT_RIGHT_OF_WAY_I)

         cdata %>%
             left_join(row_data, by = c("CRASH_RECORD_ID")) %>%
             filter(!is.na(NOT_RIGHT_OF_WAY_I)) %>%
             mutate(RIGHT_OF_WAY_I = NOT_RIGHT_OF_WAY_I == "N") %>%
             group_by(RIGHT_OF_WAY_I) %>%
             count() %>%
             ungroup %>%
             mutate(freq = n/sum(n)) %>%
             ggplot(aes(x = "", y = freq, fill = factor(RIGHT_OF_WAY_I))) +
             geom_col(width = 1, color = "white") +
             coord_polar(theta = "y") +
             labs(title = "Driver's Right of Way in Crash",
                  x = NULL, y = NULL, fill = "Right of way") +
             theme_void() +
             theme(panel.grid.minor = element_blank(),
                   plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30))) +
             geom_text(aes(label = scales::percent(freq)),
                       position = position_stack(vjust = 0.5),
                       size=8, color = "white", fontface = "bold")
```



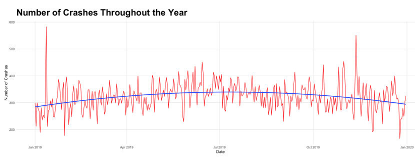
Therefore, our results show that humans are bad drivers. Of the top ten, only one of them (weather) was an external cause. But, does that mean that car crashes will no longer occur if we all use self-driving vehicles? Self-driving cars would probably significantly decrease the rate of car crashes because they in constant, complete awareness of their surroundings and have much faster reaction times, but car crashes will still exist due to external factors. Although the top five causes are human error, a large chuck (about 40%) could them inolve other factors as well, which we will now examine.

Frequency of Accidents in Relation to Time

There are many factors that could affect the frequency of accidents throughout the year. Through some analysis, we can observe weather there are any trends that arise from looking at the number of crashes against the time of year. The day of week is also another interesting factor to look at regarding the frequency of accidents. From the data, we will compute which day of the week has the most number of accidents, and see how much it deviates from the average on any other given day. The time of day the crashes occur is also another variable that could hold significance in the number of crashes. Through models and analysis, we will determine whether these time factors are statiscally signifcant in the frequency of crashes in Chicago throughout 2019.

Firstly, we analyse the frequency of accidents per day throughout the year. There are many things that change throughout the year, such as weather conditions and number of cars on the road due to travelling, such as the Summer or holiday seasons. Our findings below will allow us to see any observable trends and whether the date or time of year is significant in affecting the frequency of accidents.

```
In [38]: # Set plot dimensions
         options(repr.plot.width = 16, repr.plot.height = 6)

         crash_dates = cdata %>%
             mutate(CRASH_DATE = as.Date(CRASH_DATE)) %>%
             select(CRASH_DATE) %>%
             group_by(CRASH_DATE) %>%
             summarize(NUMBER_OF_CRASHES = n())

         ggplot(crash_dates, aes(x = CRASH_DATE, y = NUMBER_OF_CRASHES)) +
             geom_line(color = "red") +
             geom_smooth(method = "lm", se = F, formula = "y ~ poly(x, 2)") +
             labs(title = "Number of Crashes Throughout the Year",
                  y = "Number of Crashes", x = "Date") +
             theme_minimal() +
             theme(panel.grid.minor = element_blank(),
                   plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30)),
                   legend.position = "none")
```



From the graph above, we can see that the number of crashes does not vary significatly throughout the year. There are two major outliers, which happen in January and November, which does not affect the overall estimated trend line. Although the number of crashes are quite evelyn distributed throughout 2019, we can see that the graph peaks during the Summer. Thus, through this investigation, although the number of crashes peak in the Summer, we can hypothesise that the time of year does not have great significance in the outcome of the frequency of crashes.

We will use a linear regression model to determine whether the time of year does have a correlation and effect in the frequency of crashes.

```
In [81]: # Using a linear regression model to determine whether the
         # crash date has a correlation with the frequency of crashes.
         linMod = lm(NUMBER_OF_CRASHES ~ CRASH_DATE, data = crash_dates)
         summary(linMod)
```

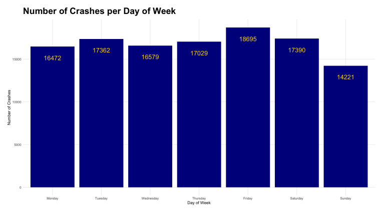
```
Call:
lm(formula = NUMBER_OF_CRASHES ~ CRASH_DATE, data = crash_dates)

Residuals:
     Min       1Q   Median       3Q      Max
-161.409  -31.156   -0.082   29.919  265.078

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -171.65151  461.73220  -0.372    0.710
CRASH_DATE     0.02734    0.02554   1.070    0.285

Residual standard error: 51.41 on 363 degrees of freedom
Multiple R-squared:  0.003147,	Adjusted R-squared:  0.0004005
F-statistic: 1.146 on 1 and 363 DF,  p-value: 0.2851
```

As seen in the linear regression model above, we can see that the crash date has a p-value of 0.2851. Using a critical value of 0.05, since the p-value is less than the critical value, we can conclude that the time of year is not statiscally significant in the outcome of the number of crashes.

In other words, this linear regression model tells us the crash date does not have an affect on the frequency of crashes, which supports our earlier hypothesis.

Next, we will analyse the days of the week in relation to the number of crashes. Throughout the week, there are certainly different factors that could lead to a variance in the frequency of crashes. Some of the outcomes could be that there are more crashes during the weekend due to a larger presence of drinking and driving during or higher traffic during work and school days. Below, we will investigate which day of the week has the highest number of crashes, and compare it to the number of crashes on any given day throughout the year.

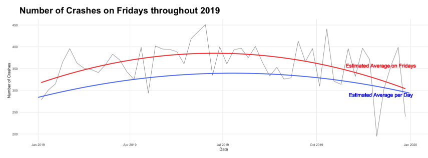
In [40]:
```
# Set plot dimensions
options(repr.plot.width = 14, repr.plot.height = 8)

cdata %>% mutate(CRASH_DAY = as.factor(weekdays(CRASH_DATE))) %>%
    mutate(CRASH_DAY = factor(CRASH_DAY, levels = c("Monday", "Tuesday", "Wednesday", "Thursday",
                                                     "Friday", "Saturday","Sunday"))) %>%
    select(CRASH_DAY) %>%
    group_by(CRASH_DAY) %>%
    summarize(NUMBER_OF_CRASHES = n()) %>%
    ggplot(aes(x = CRASH_DAY, y = NUMBER_OF_CRASHES)) +
    geom_col(fill = "darkblue") +
    labs(title = "Number of Crashes per Day of Week",
         y = "Number of Crashes", x = "Day of Week") +
    geom_text(aes(label = NUMBER_OF_CRASHES), vjust = 3, size = 6, color = "gold") +
    theme_minimal() +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30)),
          legend.position = "none")
```



The graph tells us that the most crashes happen on Friday, out of all seven days in a week, with 18695 crashes in 2019. Although Friday has the highest number, it is seen through the graph that the number of crashes does not vary greatly among the seven days in a week.

We will compare the estimated average of crashes on Friday with the estimated average of crashes on any day throughout 2019. This will let us know whether Fridays consistently have a higher number of crashes throughout the year compared to the overall estimated average.

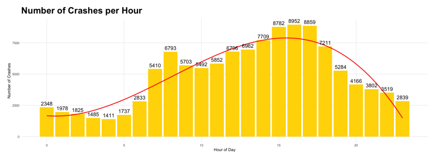
In [41]:
```
# Set plot dimensions
options(repr.plot.width = 16, repr.plot.height = 6)

# Creating a function to calculate the total number of crashes for any weekday of every week in 2019
crashDays_all = function(day) {
    cdata %>%
        arrange(CRASH_DATE) %>%
        mutate(CRASH_DAY = weekdays(CRASH_DATE), CRASH_DATE = as.Date(CRASH_DATE)) %>%
        select(CRASH_DATE, CRASH_DAY) %>%
        filter(CRASH_DAY == day) %>%
        group_by(CRASH_DATE) %>%
        count() %>%
        mutate(NUMBER_OF_CRASHES = n) %>%
        select(CRASH_DATE, NUMBER_OF_CRASHES)
}

# Calculating the total number of crashes for each Friday in 2019
Fridays = crashDays_all("Friday")

ggplot(cdata, aes(x = CRASH_DATE, y = NUMBER_OF_CRASHES)) +
    geom_line(color = "black", data = Fridays, alpha = 0.5) +
    geom_smooth(color = "red", data = Fridays, method = "lm", se = F, formula = "y ~ poly(x, 2)") +
    geom_smooth(data = crash_dates, method = "lm", se = F, formula = "y ~ poly(x, 2)") +
    labs(title = "Number of Crashes on Fridays throughout 2019",
         y = "Number of Crashes", x = "Date") +
    geom_text(aes(y = 290, x = as.Date("2019-12-13") - 10), label = "Estimated Average per Day", data = crash_dates,
              col = "blue", size = 5) +
    geom_text(aes(y = 357, x = as.Date("2019-12-13") - 10), label = "Estimated Average on Fridays", data = Fridays,
              col = "red", size = 5) +
    theme_minimal() +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30)),
          legend.position = "none")
```



The graph above shows that the estimated average of crashes on Fridays are higher than the estimated average of crashes on any given day in 2019. Thus, we can conclude from the graph that Fridays consistently have a higher number of crashes than any other day.

Our last analysis in this section is regarding the hour of the day the crashes occur. There are many reasons why the number of crashes can vary throughout the day, such as the amount of traffic, the weather condition, the presence of day light and the presence of road work amongst many other factors. Our analysis will allow us to determine whether the hour of day the crashes occurs has any significance in the frequency of crashes. We will also be able to observe any trends that arise and which hours of the day produce the most crashes.

In [42]:
```
crash_hours = cdata %>%
    mutate(CRASH_HOUR = hour(CRASH_DATE)) %>%
    select(CRASH_HOUR) %>%
    group_by(CRASH_HOUR) %>%
    summarize(NUMBER_OF_CRASHES = n())

crash_hours %>%
    ggplot(aes(x = CRASH_HOUR, y = NUMBER_OF_CRASHES)) +
    geom_col(fill = "gold") +
    geom_smooth(method = "lm", se = F, formula = "y ~ poly(x, 3)", color = "red") +
    labs(title = "Number of Crashes per Hour",
         y = "Number of Crashes", x = "Hour of Day") +
    geom_text(aes(x = CRASH_HOUR, y = NUMBER_OF_CRASHES, label = NUMBER_OF_CRASHES), vjust = -0.5, size = 5) +
    theme_minimal() +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30)),
          legend.position = "none")
```



The graph above shows us that the number of crashes vary greatly throughout any given day. We can see that the trend is that the number of crashes tend to increase from the morning to afternoon, peak from 3pm - 5pm, then decrease again. We can also see that rush hour times in Chicago tend to have higher frequency of crashes. During the morning rush hour, which is around 7am - 9am, we can see a steep increase in the numbers. It decreases again and increases when the afternoon rush hour starts, which is around 3pm - 6pm. Thus, the frequency of crashes could be directly correlated to the amount of traffic on the roads at that time, and does not necessarily mean that there is a higher probability of getting into a crash during rush hour than other day time of the day.

Since we can observe a clear trend in the graph, we will construct a linear regression model to determine whether the crash hour has an affect on the frequency of crashes.

In [43]:
```
# Using a linear regression model to determine whether the
# crash date has a correlation with the frequency of crashes.
linMod = lm(NUMBER_OF_CRASHES ~ CRASH_HOUR, data = crash_hours)
summary(linMod)
```

```
Call:
lm(formula = NUMBER_OF_CRASHES ~ CRASH_HOUR, data = crash_hours)

Residuals:
    Min      1Q  Median      3Q     Max 
-3984.8 -2024.2    97.7  1805.9  3295.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  2988.56     895.75   3.336  0.00299 **
CRASH_HOUR    166.75      66.73   2.499  0.02043 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2263 on 22 degrees of freedom
Multiple R-squared:  0.2211,	Adjusted R-squared:  0.1857 
F-statistic: 6.243 on 1 and 22 DF,  p-value: 0.02043
```

This linear regression model tells that the crash hour has a p-value of 0.02043. Using a critical value of 0.05, since the p-value is less than the critical value, we can say that the crash hour is statiscally significant in the outcome frequency of crashes.

This linear regression model brings us to the conclusion that the time of the day has an affect in the frequency of crashes.

In this section we have come to three conclusions by looking at the relationship of frequency of crashes with the time of year, week and day. We have firstly concluded that the time of year does not have an impact on the frequency crashes, and that the most crashes in 2019 happened in Summer. Secondly, we discovered that there were more crashes on Fridays than any other day of the week, with the estimated average of crashes every Friday being higher than the estimated average of crashes throughout the year. In our last anaylsis, we concluded that the time of day does have a correlation with the frequency of crashes, with the most number of crashes happening between 3pm - 5pm. The frequency of crashes in relation to different times of the year, week and month were definitely interesting to analyse, and there could be many factors contributing to these outcomes.

Injuries

From scratching a bumper, to totaling a car, the level of damage caused by a car accident can vary tremendously. While one could argue the importance of cars, the value of human life is something that should be held at the uptmost priority. If we were able to learn the conditions in which injury to human life is most likely to occur in an accident, we may be able to limit our driving when those conditions are present, or at least drive with extra caution during them.

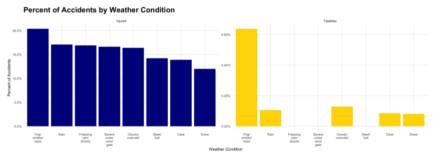
In this section, we will examine how various conditions effect the percentage of accidents with injuries, as well as take a closer look into the percentage of accidents involving fatalities under these conditions. After examining the conditions individual effects, we will conclude by examining combinations of each possible condition to determine the worst driving conditions as a whole.

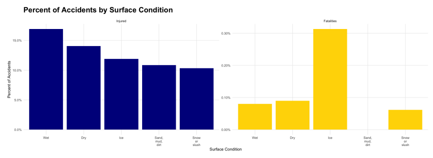
First, we will select the necessary columns from the dataset for our analysis of injuries under various conditions.

In [44]:
```
injury_data = cdata %>%
    select(CRASH_DATE,
           WEATHER_CONDITION, ROADWAY_SURFACE_COND, POSTED_SPEED_LIMIT,
           INJURIES_TOTAL, INJURIES_FATAL)
```

It would seem to be a natural assumption that weather may play a big role in the severity of an accident. As such, we would like to start by examining the role of weather on rates of accidents involving fatal injuries as well as injuries in general. Closely related to weather, we would also like to examine the impact of road surface conditions on each of these rates. For this first analysis, we will be filtering out any unknown weather/road conditions and injury counts for our investigation. We've chosen to filter to only conditions containing at least 30 entries, in hopes to ensure we have sufficent data for statistical analysis of each condition.

In [45]:
```
# Set plot dimensions
options(repr.plot.width = 16, repr.plot.height = 6)

weather_data = injury_data %>%
    filter(!(WEATHER_CONDITION %in% c("OTHER", "UNKNOWN"))) %>%
    group_by(WEATHER_CONDITION) %>%
    filter(n() > 30) %>%
    summarize(PERCENT_INJURED = mean(INJURIES_TOTAL > 0, na.rm = T), PERCENT_FATALITIES = mean(INJURIES_FATAL > 0, n
a.rm = T)) %>%
    rename(Injured = PERCENT_INJURED, Fatalities = PERCENT_FATALITIES) %>%
    gather(key = "type", value = "percent", Injured, Fatalities) %>%
    mutate(type = factor(type, levels = c("Injured", "Fatalities")))

surface_data = injury_data %>%
    filter(!(ROADWAY_SURFACE_COND %in% c("OTHER", "UNKNOWN"))) %>%
    group_by(ROADWAY_SURFACE_COND) %>%
    filter(n() > 30) %>%
    summarize(PERCENT_INJURED = mean(INJURIES_TOTAL > 0, na.rm = T), PERCENT_FATALITIES = mean(INJURIES_FATAL > 0, n
a.rm = T)) %>%
    rename(Injured = PERCENT_INJURED, Fatalities = PERCENT_FATALITIES) %>%
    gather(key = "type", value = "percent", Injured, Fatalities) %>%
    mutate(type = factor(type, levels = c("Injured", "Fatalities")))

ggplot(weather_data, aes(x = reorder(WEATHER_CONDITION, -percent), y = percent)) +
    geom_col(data = subset(weather_data, type == "Injured"), fill = "darkblue") +
    geom_col(data = subset(weather_data, type == "Fatalities"), fill = "gold") +
    facet_wrap(~ type, scales = "free_y") +
    scale_x_discrete(labels = function(x) str_wrap(str_to_sentence(x), width = 5)) +
    scale_y_continuous(labels = scales::percent) +
    labs(title = "Percent of Accidents by Weather Condition",
         x = "Weather Condition", y = "Percent of Accidents") +
    theme_minimal() +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 20, face = "bold", margin = margin(b = 10, t = 30)))

ggplot(surface_data, aes(x = reorder(ROADWAY_SURFACE_COND, -percent), y = percent)) +
    geom_col(data = subset(surface_data, type == "Injured"), fill = "darkblue") +
    geom_col(data = subset(surface_data, type == "Fatalities"), fill = "gold") +
    facet_wrap(~ type, scales = "free_y") +
    scale_x_discrete(labels = function(x) str_wrap(str_to_sentence(x), width = 5)) +
    scale_y_continuous(labels = scales::percent) +
    labs(title = "Percent of Accidents by Surface Condition",
         x = "Surface Condition", y = "Percent of Accidents") +
    theme_minimal() +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 20, face = "bold", margin = margin(b = 10, t = 30)))
```





In order from left to right, it appears in our first plot that fog, smoke, and haze lead to the highest percentage of injuries in accidents. Interestingly, the percentage of accidents involving injuries during clear weather is actually higher than that of snowy weather.

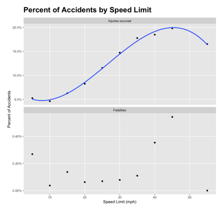
In our second plot, we see that fog, smoke, and haze also lead to the greatest amount of fatalities. While fatalities within this condition are still well below 1% of accidents, that rate is nearly five times greater than the next highest percentage condition:cloudy/overcast. Additionally, it would appear that clear weather has a slightly higher percentage of fatalities than snow.

In our third plot, we find that the highest percentage of accidents occur when the road is wet. Interestingly, we find that dry roads have the second highest percentage of injuries, above both that of ice and slush.

In our forth plot, it would appear to be the case that icy roads by far have the highest percentage of accidents with fatalities, despite having only the third highest percentage of injuries as seen in the third plot. While ice only has just a .3% accident fatality rate, it is over three times the second highest fatality rate, which intriguingly is again dry roads.

We would next like to examine the effect that posted speed limits have on injury percentages, both fatal and in general. We will be again be filtering the data to ensure we have only statistically sound data for our analysis with reasonable amounts of data for conventional speed limits.

In [46]:
```
# Set plot dimensions
options(repr.plot.width = 8, repr.plot.height = 8)

speed_limit_data = injury_data %>%
    filter((POSTED_SPEED_LIMIT %% 5) == 0, POSTED_SPEED_LIMIT > 0, !is.na(INJURIES_FATAL)) %>%
    group_by(POSTED_SPEED_LIMIT) %>%
    filter(n() > 30) %>%
    summarize(PERCENT_INJURIES_OCCURED = mean(INJURIES_TOTAL > 0), PERCENT_FATALITIES = sum(INJURIES_FATAL > 0)/n())
%>%
    rename(Fatalities = PERCENT_FATALITIES, `Injuries occured` = PERCENT_INJURIES_OCCURED) %>%
    gather(key = "type", value = "percent", `Injuries occured`, Fatalities) %>%
    mutate(type = factor(type, levels = c("Injuries occured", "Fatalities")))

ggplot(speed_limit_data, aes(x = POSTED_SPEED_LIMIT, y = percent)) +
    geom_point() +
    geom_smooth(data = subset(speed_limit_data, type == "Injuries occured"), se = F, method = 'lm', formula = 'y ~ p
oly(x, 3)') +
    facet_wrap(~ type, ncol = 1, scales = "free_y") +
    scale_y_continuous(labels = scales::percent) +
    labs(title = "Percent of Accidents by Speed Limit",
         x = "Speed Limit (mph)", y = "Percent of Accidents") +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 20, face = "bold", margin = margin(b = 10, t = 30)))
```

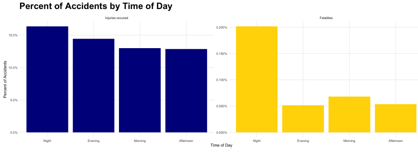


From our first plot, we see that as a general trend, it would appear that the percent of accidents involving injuries increases as speed limit increases, in a sigmoidal fashion. 10 mph falls slightly below the trend, but fifty five mph is definitely a point of interest here, as it has a lower rate of injury than 35-45 mph. Instead, fourty five mph has the highest injury rates.

From our second plot, we find that there does not appear to be a clear correlation between posted speed limit and accident fatality rates, but the highest fatality rate is shown to occur at 45 mph.

The final condition we would like to examine for impacts on accident injury rates, both fatal and in general, is the time of day an accident occurs. We will once again be filtering out data that does not have a known injury total. We will create a new time of day variable to help us place times into specific categories. We will also ensure we have a sufficent amount of data in each time of day category for statistical analysis.

In [47]:
```
# Set plot dimensions
options(repr.plot.width = 16, repr.plot.height = 6)

tod_data = injury_data %>%
    filter(!is.na(INJURIES_TOTAL)) %>%
    mutate(TIME_OF_DAY = case_when(
        between(hour(CRASH_DATE), 20, 23) | between(hour(CRASH_DATE), 0, 4) ~ "Night",
        between(hour(CRASH_DATE), 5, 11) ~ "Morning",
        between(hour(CRASH_DATE), 12, 16) ~ "Afternoon",
        between(hour(CRASH_DATE), 17, 20) ~ "Evening"
    )) %>%
    group_by(TIME_OF_DAY) %>%
    filter(n() > 30) %>%
    summarize(PERCENT_INJURY_OCCURED = sum(INJURIES_TOTAL > 0)/n(), PERCENT_FATALITIES_OCCURED = sum(INJURIES_FATAL
> 0)/n()) %>%
    rename(Fatalities = PERCENT_FATALITIES_OCCURED, `Injuries occured` = PERCENT_INJURY_OCCURED) %>%
    gather(key = "type", value = "percent", `Injuries occured`, Fatalities) %>%
    mutate(type = factor(type, levels = c("Injuries occured", "Fatalities")))

ggplot(tod_data, aes(x = reorder(TIME_OF_DAY, -percent), y = percent)) +
    geom_col(data = subset(tod_data, type == "Injuries occured"), fill = "darkblue") +
    geom_col(data = subset(tod_data, type == "Fatalities"), fill = "gold") +
    facet_wrap(~ type, scales = "free_y") +
    scale_y_continuous(labels = scales::percent) +
    labs(title = "Percent of Accidents by Time of Day",
         x = "Time of Day", y = "Percent of Accidents") +
    theme_minimal() +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 24, face = "bold", margin = margin(b = 10, t = 30)))
```



In our first plot, it appears that night time has the highest percentage of accidents involving injuries. Evening has the second highest injuries, with morning just barely having a higher percentage than the afternoon.

Our second plot reveals that night by far has the highest accident fatality rate. While night only has just around a .2% accident fatality rate, it is almost three times the second highest fatality rate, which was morning.

Now that we have individually investegated several conditions' effects on the rate of accidents involving injuries, we will next be combining them to query under which set of conditions accident involving injuries, both fatal and in general, are most likely to happen. We will again be applying the necessary filters from each previous step to ensure our data is statistically sound, as well as to recreate our time of day variable for our investigation.

In [50]:
```
top_injury_data = injury_data %>%
    filter(
    WEATHER_CONDITION %in% c("RAIN", "SNOW", "CLEAR", "FOG/SMOKE/HAZE", "CLOUDY/OVERCAST", "SEVERE CROSS WIND GATE"
),
    !is.na(INJURIES_TOTAL),
    !(ROADWAY_SURFACE_COND %in% c("OTHER", "UNKNOWN")),
    (POSTED_SPEED_LIMIT %% 5) == 0,
    POSTED_SPEED_LIMIT > 0) %>%
    mutate(TIME_OF_DAY = case_when(
        between(hour(CRASH_DATE), 20,23) | between(hour(CRASH_DATE), 0,4) ~ "Night",
        between(hour(CRASH_DATE), 5,11) ~ "Morning",
        between(hour(CRASH_DATE), 12,16) ~ "Afternoon",
        between(hour(CRASH_DATE), 17,20) ~ "Evening"
    )) %>%
    group_by(WEATHER_CONDITION, ROADWAY_SURFACE_COND, POSTED_SPEED_LIMIT, TIME_OF_DAY) %>%
    filter(n() > 30) %>%
    summarize(PERCENT_INJURY_OCCURED = sum(INJURIES_TOTAL > 0)/n(),
              PERCENT_FATALITY_OCCURED = sum(INJURIES_FATAL > 0)/n(),
              .groups = "drop_last")

top_injury_data %>%
    arrange(-PERCENT_INJURY_OCCURED) %>%
    mutate(PERCENT_INJURY_OCCURED = scales::percent(PERCENT_INJURY_OCCURED)) %>%
    ungroup %>%
    slice_max(1) %>%
    select(`Weather` = WEATHER_CONDITION,
           `Roadway Surface` = ROADWAY_SURFACE_COND,
           `Posted Speed Limit` = POSTED_SPEED_LIMIT,
           `Time of Day` = TIME_OF_DAY,
           `Percent of Accidents with Injuries` = PERCENT_INJURY_OCCURED)

top_injury_data %>%
    arrange(-PERCENT_FATALITY_OCCURED) %>%
    mutate(PERCENT_FATALITY_OCCURED = scales::percent(PERCENT_FATALITY_OCCURED, accuracy = 0.1)) %>%
    ungroup %>%
    slice_max(1) %>%
    select(`Weather` = WEATHER_CONDITION,
           `Roadway Surface` = ROADWAY_SURFACE_COND,
           `Posted Speed Limit` = POSTED_SPEED_LIMIT,
           `Time of Day` = TIME_OF_DAY,
           `Percent of Accidents with Fatalities` = PERCENT_FATALITY_OCCURED)
```

A tibble: 1 × 5

Weather	Roadway Surface	Posted Speed Limit	Time of Day	Percent of Accidents with Injuries
<chr>	<chr>	<dbl>	<chr>	<chr>
CLEAR	DRY	45	Night	26.4%

A tibble: 1 × 5

Weather	Roadway Surface	Posted Speed Limit	Time of Day	Percent of Accidents with Fatalities
<chr>	<chr>	<dbl>	<chr>	<chr>
CLOUDY/OVERCAST	WET	35	Morning	2.7%

From our first query, we have fascinatingly found that, from the combinations of conditions of which we had sufficent data, clear weather on a dry road, with a posted 45mph speed limit during the night has the highest percentage of accident injuries, at around 26.4% of accidents under that combination of conditions involving injuries.

Our second query has revealed that, of the combinations of conditions of which we had sufficent data, cloudy/overcast weather on a wet road with a posted 35mph speed limit in the morning has the highest percent of accidents with fatalities, at around a whopping 2.7%.

Through our analysis, we have learned that the highest percentage of Chicago accidents with injuries in each individual category are fog, wet roads, a 45mph speed limit, and night time driving. The highest individual condition percentage was from foggy weather, at around 20.5%. Combined, the worst conditions were clear weather on a dry road, with a posted 45mph speed limit during the night having the highest percentage of accidents involving injuries, at around 26.4% of accidents.

In terms of fatalities, we found that the highest percentage of accidents with fatalities in each individual category are fog, icy roads, 45mph speed limits, and night time driving. The highest individual condition percentage was also from foggy weather, at around 0.63%. Combined, the worst conditions were cloudy/overcast weather on a wet road with a posted 35mph speed limit in the morning, at 2.7%.

Overall, we have seen that both injuries and fatalites can be more likely based on certain conditions. Wrecking your car is unfortunate, but potentially fatal injuries could affect people for the rest of their lives. With these associatiations in mind, it is our hope that people will be able to make better and safer decisions given whatever driving conditions they may have at hand.

Hit and Runs

In this section, we will answer some intriguing questions about hit and runs. However, because the supplied dataset does not contain hit and run info, we will add it to our dataset using the original dataset from its respective website.

Additionally, we must note that the results of our analysis will not be completely accurate due to the fact that most non-hit-and-runs were not entered in the system as "not a hit and run." Instead, they are left blank with no value. Only a small percentage of non-hit-and-runs are actually entered, as we can see in the table below.

In [51]:
```
full_data %>%
    select(HIT_AND_RUN_I) %>%
    mutate(HIT_AND_RUN_I = HIT_AND_RUN_I == "Y") %>%
    group_by(HIT_AND_RUN_I) %>%
    count() %>%
    ungroup() %>%
    rename(`Hit and run` = HIT_AND_RUN_I, Observations = n)
```

A tibble: 3 × 2

Hit and run	Observations
<lgl>	<int>
FALSE	7425
TRUE	165599
NA	397291

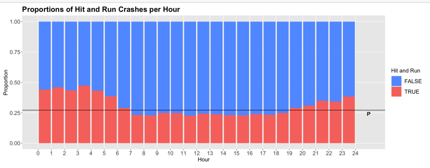
If an accident was actually a hit and run, it most likely would have been documented in the hit and run column, whereas a non-hit and run accident might not be documented as such and the HIT_AND_RUN_I column would have been left blank. Using this reasoning, although it would not lead to completely accurate results, we will assume that all empty values are not hit and runs.

In [52]:
```
# Keeping only useful information in our dataset
full_hit_n_run = full_data %>%
    select(CRASH_RECORD_ID, HIT_AND_RUN_I) %>%
    mutate(HIT_AND_RUN_I = replace_na(HIT_AND_RUN_I == "Y", F))

# Merging our cleaned data with the full dataset (that contains the hit and run data)
hit_n_run = cdata %>%
    left_join(full_hit_n_run, by = c("CRASH_RECORD_ID"))
```

We first observe how the frequency of hit and runs are distributed throughout the day. We wish to depict the frequency of all accidents for each hour of the day, separated by whether the accident is categorized as a Hit and Run or not. At first glimpse, however, this does not give much insight.

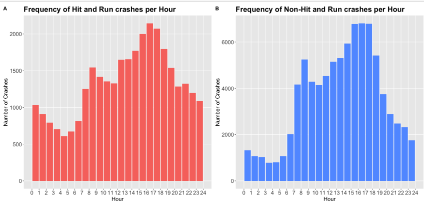
We instead turn to graphing the proportions of hit and run crashes over all crashes in each hour. With proportions, we can see that there is a higher proportion of hit and run crashes in the early and late hours of the day, relative to the proportion of all hit and run crashes over all total crashes. Specifically, the first through sixth hours of the day, as well as the 21st through 24th hours of the day have a higher proportion of hit and run crashes relative to the total proportion of hit and run crashes. This can be seen in the graph below, where the horizontal line P is the total proportion of hit and run crashes over all crashes.

In [53]:
```
# Finding the number of total crashes, total hit and run crashes, and total non-hit and run crashes
num_total_crashes = hit_n_run %>% nrow()
num_hit_n_run = hit_n_run %>% count(HIT_AND_RUN_I) %>% select(n) %>% slice(1) %>% as.integer()
num_not_hit_n_run = num_total_crashes - num_hit_n_run

# Plotting the proportion of hit and run crashes (in blue) over all crashes in each hour
hit_n_run %>%
    mutate(CRASH_HOUR = hour(CRASH_DATE) + .5) %>%
    select(CRASH_HOUR, HIT_AND_RUN_I) %>%
    ggplot() + geom_bar(mapping = aes(x = CRASH_HOUR, fill = HIT_AND_RUN_I), width = .9, position = "fill") +
    geom_hline(yintercept = num_not_hit_n_run/num_total_crashes) +
    scale_fill_manual(values = c("FALSE" =  "#619CFF", "TRUE" = "#F8766D")) +
    geom_text(aes(y = num_not_hit_n_run/num_total_crashes-0.03, x = 25), label = "P", data = hit_n_run, size = 5) +
    labs(title = "Proportions of Hit and Run Crashes per Hour",
         y = "Proportion", x = "Hour", fill = "Hit and Run") +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 20, face = "bold"), axis.title = element_text(size = 15),
          axis.text = element_text(size = 15), legend.title = element_text(size = 15),
          legend.text = element_text(size = 15), legend.key.size = unit(1, 'cm')) +
    scale_x_continuous(breaks = seq(0, 24, by = 1))
```



This observation is further supported by a side by side comparison of the distribution of the frequency of hit and run crashes and non-hit and run crashes per hour, shown by the graph below. The graph depicting the frequency of hit and run accidents have a much higher frequency in the early and late hours of the day relative to the graph depicting non-hit and run accident frequencies per hour.

In [54]:
```
hour_har_only = hit_n_run %>%
    mutate(CRASH_HOUR = hour(CRASH_DATE) + .5) %>%
    select(CRASH_HOUR, HIT_AND_RUN_I) %>%
    filter(HIT_AND_RUN_I == T) %>%
    group_by(CRASH_HOUR) %>%
    summarize(n = n()) %>%
    ggplot() +
    geom_col(mapping = aes(x = CRASH_HOUR, y = n), fill = "#F8766D") +
    scale_x_continuous(breaks = seq(0, 24, by = 1)) +
    labs(title = "Frequency of Hit and Run crashes per Hour",
         y = "Number of Crashes", x = "Hour") +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 20, face = "bold"), axis.title = element_text(size = 15),
          axis.text = element_text(size = 15))

hour_not_har_only = hit_n_run %>%
    mutate(CRASH_HOUR = hour(CRASH_DATE) + .5) %>%
    select(CRASH_HOUR, HIT_AND_RUN_I) %>%
    filter(HIT_AND_RUN_I == F) %>%
    group_by(CRASH_HOUR) %>%
    summarize(n = n()) %>%
    ggplot() +
    geom_col(mapping = aes(x = CRASH_HOUR, y = n), fill = "#619CFF") +
    scale_x_continuous(breaks = seq(0, 24, by = 1)) +
    labs(title = "Frequency of Non-Hit and Run crashes per Hour",
         y = "Number of Crashes", x = "Hour") +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(size = 20, face = "bold"), axis.title = element_text(size = 15),
          axis.text = element_text(size = 15))

options(repr.plot.width=16, repr.plot.height=7.5)
plot_grid(hour_har_only, hour_not_har_only, labels = "AUTO")
```



We are not surprised by the resulting inference that follows: There is a relatively higher proportion of hit and run accidents over all accidents in the early and late hours of the day. This inference makes sense; These hours may impose a plethora of reasons on the offender of the accident to "run" after a "hit". Lack of daylight, sleepiness/lack of desire to deal with repercussions, and lack of police presence may be a few among potentially many of these reasons. This inference suggests that hit and runs may be more likely to occur during the darker hours of the day than during the day's lighter hours.

Next, we take a look at how injuries are related to hit and run accidents. We wish to observe whether people more willing to call police if there are any injuries on the scene of the accident (We also wish to find that people do indeed have motivation to call the police if there are injuries). In order to answer this question, we will run two sets of two proportion z-tests, where the two populations are hit and run accidents and non-hit and run accidents. The numbers for these two proportion z-tests are given below:

In [55]:
```
prop_test_nums = hit_n_run %>%
    select(HIT_AND_RUN_I, starts_with("Injuries")) %>%
    filter(!is.na(HIT_AND_RUN_I),
           !is.na(INJURIES_TOTAL),
           !is.na(INJURIES_FATAL),
           !is.na(INJURIES_INCAPACITATING)) %>%
    mutate(INJURY = ifelse(INJURIES_TOTAL > 0, 1, 0),
           SEVERE_INJURY = ifelse(INJURIES_FATAL > 0 | INJURIES_INCAPACITATING >0, 1, 0),
           HIT_AND_RUN_I = ifelse(HIT_AND_RUN_I == T, "Hit and Run", "Not Hit and Run")) %>%
    group_by(HIT_AND_RUN_I) %>%
    summarize(`Number of Crashes with Injuries` = sum(INJURY),
              `Number of Crashes with Severe Injuries` = sum(SEVERE_INJURY),
              `Number of Crashes` = n()) %>%
    mutate(`Hit and Run?` = HIT_AND_RUN_I) %>%
    select(`Hit and Run?`, !HIT_AND_RUN_I)
prop_test_nums
```

A tibble: 2 × 4

Hit and Run?	Number of Crashes with Injuries	Number of Crashes with Severe Injuries	Number of Crashes
<chr>	<dbl>	<dbl>	<int>
Hit and Run	3153	393	31860
Not Hit and Run	13159	1719	85623

The first two proportion z-test will test whether the proportion of any injury in a hit and run accident is equal to the proportion of any injury in a non-hit and run accident.

Population 1: Hit and Run accidents
$n_1 = 31,860$
$p_1 = \frac{3,153}{31,860} = 0.0990$

Population 2: Non-Hit and Run accidents
$n_2 = 85,623$
$p_2 = \frac{13,159}{85,623} = 0.1537$

Null hypothesis: $H_0 : p_1 = p_2$
Alternative hypothesis: $H_0 : p_1 \leq p_2$
Significance level: $\alpha = 0.05$

The second two proportion z-test will test whether the proportion of a severe injury in a hit and run accident is equal to the proportion of a severe injury in a non-hit and run accident. A severe injury is categorized as a fatal or incapacitating injury at the scene of the accident.

Population 1: Hit and Run accidents
$n_1 = 31,860$
$p_1 = \frac{393}{31,860} = 0.0123$

Population 2: Non-Hit and Run accidents
$n_2 = 85,623$
$p_2 = \frac{1,719}{85,623} = 0.0201$

Null hypothesis: $H_0 : p_1 = p_2$
Alternative hypothesis: $H_0 : p_1 \leq p_2$
Significance level: $\alpha = 0.05$

The results are shown below, along with a table that outlines the proportions of the two sets of two proportion z-tests:

In [57]:
```
# Table outlining proportions
hit_n_run %>%
    select(HIT_AND_RUN_I, starts_with("Injuries")) %>%
    filter(!is.na(HIT_AND_RUN_I),
           !is.na(INJURIES_TOTAL),
           !is.na(INJURIES_FATAL),
           !is.na(INJURIES_INCAPACITATING)) %>%
    mutate(INJURY = ifelse(INJURIES_TOTAL > 0, 1, 0),
           SEVERE_INJURY = ifelse(INJURIES_FATAL > 0 | INJURIES_INCAPACITATING >0, 1, 0),
           HIT_AND_RUN_I = ifelse(HIT_AND_RUN_I == T, "Hit and Run", "Not Hit and Run")) %>%
    group_by(HIT_AND_RUN_I) %>%
    summarize(prop_injuries = sum(INJURY)/n(),
              prop_severe_injuries = sum(SEVERE_INJURY)/n()) %>%
    t() %>%
    as_tibble() %>%
    row_to_names(row_number = 1) %>%
    cbind(c("Proportion of Any Injuries in Crash", "Proportion of Severe Injuries in Crash")) %>%
    mutate(Proportions = `c("Proportion of Any Injuries in Crash", "Proportion of Severe Injuries in Crash")`) %>%
    select(Proportions, `Hit and Run`, `Not Hit and Run`)

# Naming variables for two proportion z-tests
n1 = as.integer(prop_test_nums[1,4])
n2 = as.integer(prop_test_nums[2,4])
x1_any = as.integer(prop_test_nums[1,2])
x2_any = as.integer(prop_test_nums[2,2])
x1_severe = as.integer(prop_test_nums[1,3])
x2_severe = as.integer(prop_test_nums[2,3])

# Running two proportion z-tests
prop.test(x = c(x1_any, x2_any), n = c(n1, n2), alternative = "less")
prop.test(x = c(x1_severe, x2_severe), n = c(n1, n2), alternative = "less")
```

A data.frame: 2 × 3

Proportions	Hit and Run	Not Hit and Run
<chr>	<chr>	<chr>
Proportion of Any Injuries in Crash	0.09896422	0.15368534
Proportion of Severe Injuries in Crash	0.01233522	0.02007638

```
	2-sample test for equality of proportions with continuity correction

data:  c(x1_any, x2_any) out of c(n1, n2)
X-squared = 581.05, df = 1, p-value < 2.2e-16
alternative hypothesis: less
95 percent confidence interval:
 -1.00000000 -0.05128167
sample estimates:
    prop 1     prop 2 
0.09896422 0.15368534 
```

```
	2-sample test for equality of proportions with continuity correction

data:  c(x1_severe, x2_severe) out of c(n1, n2)
X-squared = 78.382, df = 1, p-value < 2.2e-16
alternative hypothesis: less
95 percent confidence interval:
 -1.000000000 -0.006432687
sample estimates:
    prop 1     prop 2 
0.01233522 0.02007638 
```

Let us take a look at the results of both of the two proportion z-tests above. For both tests, the respective p-values are much much less than our prescribed significance level $\alpha = 0.05$, so we can reject their null hypotheses. Thus we can conclude that the proportion of any injury in a hit and run accident is less than the proportion of any injury in a non-hit and run accident in Chicago. We can also conclude that the proportion of a severe injury in a hit and run accident is less than the proportion of a severe injury in a non-hit and run accident in Chicago. While these may seem to imply that people are more likely to call the police if an accident results in an injury of any sort, we must be careful not to extend correlation to causation. The only reasonable inference we can draw from these results is that there is a higher proportion of injuries sustained in non-hit and run accidents than injuries sustained in hit and run accidents. Regardless, this is some insightful result, and may definitely be an uplifting one as well.

We also investigate how the total cost in damage (at the scene of accident) may be distributed given whether the accident was a hit and run. To dissect this, we separate the dollar value of damages into three categories:

```
Small damage: $500 or less
```

Moderate damage: $501 to$1500 Large damage: $1501 or more

The table below displays the proportion of small damage, moderate damage, and large damage given whether the accident was a hit and run:

In [58]:

```r
# Table displaying the proportions of damage given whether hit and run or not
HAR_DAMAGE = hit_n_run %>%
    select(HIT_AND_RUN_I, DAMAGE) %>%
    filter(!is.na(HIT_AND_RUN_I), HIT_AND_RUN_I == T) %>%
    mutate(DAMAGE = as.factor(DAMAGE),
           HIT_AND_RUN = ifelse(HIT_AND_RUN_I == T, "Hit and Run", "Not Hit and Run")) %>%
    mutate(HIT_AND_RUN = as.factor(HIT_AND_RUN),
           sdam_har = ifelse(DAMAGE == "$500 OR LESS", 1, 0),
           mdam_har = ifelse(DAMAGE == "$501 - $1,500", 1, 0),
           ldam_har = ifelse(DAMAGE == "OVER $1,500", 1, 0)) %>%
    group_by(HIT_AND_RUN) %>%
    summarize(`Proportion of Small Damage` = sum(sdam_har)/n(),
              `Proportion of Moderate Damage` = sum(mdam_har)/n(),
              `Proportion of Large Damage` = sum(ldam_har)/n())

NOT_HAR_DAMAGE = hit_n_run %>%
    select(HIT_AND_RUN_I, DAMAGE) %>%
    filter(!is.na(HIT_AND_RUN_I), HIT_AND_RUN_I == F) %>%
    mutate(DAMAGE = as.factor(DAMAGE),
           HIT_AND_RUN = ifelse(HIT_AND_RUN_I == T, "Hit and Run", "Not Hit and Run")) %>%
    mutate(HIT_AND_RUN = as.factor(HIT_AND_RUN),
           sdam_not_har = ifelse(DAMAGE == "$500 OR LESS", 1, 0),
           mdam_not_har = ifelse(DAMAGE == "$501 - $1,500", 1, 0),
           ldam_not_har = ifelse(DAMAGE == "OVER $1,500", 1, 0)) %>%
    group_by(HIT_AND_RUN) %>%
    summarize(`Proportion of Small Damage` = sum(sdam_not_har)/n(),
              `Proportion of Moderate Damage` = sum(mdam_not_har)/n(),
              `Proportion of Large Damage` = sum(ldam_not_har)/n())

DAMAGE_vs_HIT_AND_RUNS = rbind(HAR_DAMAGE, NOT_HAR_DAMAGE) %>%
    t() %>%
    as_tibble() %>%
    row_to_names(row_number = 1) %>%
    cbind(c("Proportion of small damage",
            "Proportion of moderate damage",
            "Proportion of large damage")) %>%
    mutate(Proportions = `c("Proportion of small damage", "Proportion of moderate damage", `) %>%
    select(Proportions, `Hit and Run`, `Not Hit and Run`)

DAMAGE_vs_HIT_AND_RUNS
```

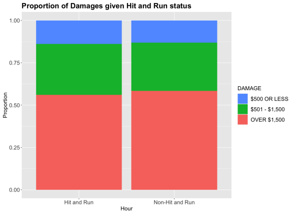
A data.frame: 3 × 3

Proportions	Hit and Run	Not Hit and Run
<chr>	<chr>	<chr>
Proportion of small damage	0.1385333	0.1306659
Proportion of moderate damage	0.3003224	0.2849334
Proportion of large damage	0.5611443	0.5844008

While the previous investigation of injuries and hit and runs resulted in some large differences between pairwise proportions, this table does not give any proportions that seem to statistically differ between hit and runs and non-hit and runs. Indeed, when we visualize the proportions given hit and run status, we see that the proportions seem to be nearly equal to one another. The graph below displays the visualization of the table above:

In [59]:

```r
# Graphing the proportion of damages given whether hit and run or not
options(repr.plot.width = 11, repr.plot.height = 8)
hit_n_run %>%
    select(HIT_AND_RUN_I, DAMAGE) %>%
    filter(!is.na(HIT_AND_RUN_I)) %>%
    mutate(DAMAGE = as.factor(DAMAGE),
           HIT_AND_RUN_I = ifelse(HIT_AND_RUN_I == TRUE, "Hit and Run", "Non-Hit and Run")) %>%
    ggplot() +
    geom_bar(mapping = aes(x = HIT_AND_RUN_I, fill = DAMAGE), position = "fill") +
    scale_fill_manual(values = c(`$500 OR LESS` = "#619CFF", `$501 - $1,500` = "#00BA38",
                                 `OVER $1,500` =  "#F8766D")) +
    labs(title = "Proportion of Damages given Hit and Run status",
         y = "Proportion", x = "Hour") +
    theme(plot.title = element_text(size = 20, face = "bold"), axis.title = element_text(size = 15),
          axis.text = element_text(size = 15), legend.title = element_text(size = 15),
          legend.text = element_text(size = 15), legend.key.size = unit(1, 'cm'))
```



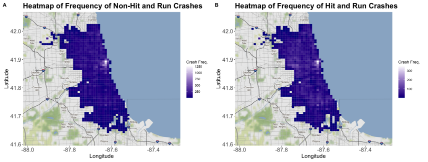
Potential reasons for the lack of difference of proportions between hit and run accidents and non-hit and run accidents can also be explained. For one, at the time of the accident, the offender most likely may not know the exact dollar amount of damage they've caused, and, more often than not, they may not even know whether the damage will be a small dollar amount or large dollar amount (except for a few cases, such as a very small tap on the rear end, or a very destructive t-bone at full speed). Thus the dollar amount of damage most likely may not go into much consideration when someone decides to speed off after causing a crash—If the offender runs after a hit because of the consequences of paying off damages, they may most likely run for any type of damage. Thus, we conclude that damage does not play a large role in whether the accident will be a hit and run or not.

Lastly, we examine how hit and run accidents are distributed throughout the city of Chicago, compared to how non-hit and run accidents are distributed throughout the city of Chicago. To show these two distributions, we use a heatmap overlayed on top of the city of Chicago.

In [62]:

```r
# Plotting heatmaps over map of chicago
chicago = get_stamenmap(bbox = c(left = -88.0225, bottom = 41.5949, right = -87.2713, top = 42.0677), zoom = 11)

# Finding locations of frequencies of accidents
lat_lon_har = hit_n_run %>%
    select(HIT_AND_RUN_I, LATITUDE, LONGITUDE) %>%
    filter(!is.na(LATITUDE), !is.na(LONGITUDE), HIT_AND_RUN_I == T)

lat_lon_not_har = hit_n_run %>%
    select(HIT_AND_RUN_I, LATITUDE, LONGITUDE) %>%
    filter(!is.na(LATITUDE), !is.na(LONGITUDE), HIT_AND_RUN_I == F)

# Setting map of chicago
options(repr.plot.width = 16, repr.plot.height = 6)
heatmap_har = ggmap(chicago) +
    geom_bin2d(mapping = aes(x = LONGITUDE, y = LATITUDE), bins = 70, data = lat_lon_har) +
    scale_fill_gradient(low = "dark blue", high = "white") +
    labs(title = "Heatmap of Frequency of Hit and Run Crashes",
         y = "Latitude", x = "Longitude", fill = "Crash Freq.") +
    theme(plot.title = element_text(size = 20, face = "bold"),
          axis.title = element_text(size = 15),
          axis.text = element_text(size = 15))

heatmap_not_har = ggmap(chicago) +
    geom_bin2d(mapping = aes(x = LONGITUDE, y = LATITUDE), bins = 70, data = lat_lon_not_har) +
    scale_fill_gradient(low = "dark blue", high = "white") +
    labs(title = "Heatmap of Frequency of Non-Hit and Run Crashes",
         y = "Latitude", x = "Longitude", fill = "Crash Freq.") +
    theme(plot.title = element_text(size = 20, face = "bold"),
          axis.title = element_text(size = 15),
          axis.text = element_text(size = 15))

plot_grid(heatmap_not_har, heatmap_har, labels = "AUTO")
```

```
Warning message:
“Removed 9 rows containing non-finite values (stat_bin2d).”
```



With these heatmaps, we can see one rather interesting result: Non-hit and run crashes are more heavily distributed within downtown Chicago than hit and run crashes are. This actually makes sense from several standpoints. Firstly, downtown Chicago may have a higher police presence than the other areas of Chicago, so hit and runs may be less likely to happen in downtown. Secondly, there may most likely be more witnesses at the scene of accidents that occur in downtown, which may deter hit and runs further. Also, due to high vehicle count and traffic in downtown, as well as lower posted speed limits, hitting and running may be more difficult. These are just some of many possible reasons why hit and runs are less concentrated in downtown Chicago than non-hit and run accidents. Nevertheless, it's important to note that both hit and runs and non-hit and run accidents are largely concentrated in downtown.

Summary

We found that the most common cause of car crashes was due to human error with drivers not having the right of way in most situations. Although most crashes do not vary with the time of year, we concluded that the time of day, specifically 3-5 pm, and the day of the week, or Friday's in particular, tended to have far more crashes. The stand alone conditions with the largest number of fatalities were fog, icy roads, 45mph speed limits and night time driving. The worst combination of these conditions for fatalities was cloudy/overcast weather on a wet road with a posted 35mph speed limit in the morning. In contrast, the worst combination of these conditions for injuries was clear weather on a dry road with a posted 45mph speed limit in the night. Furthermore, we also gathered that hit and runs tend to happen at early and late hours and do not vary greatly across the cost of damage. Most hit and runs also happen within downtown Chicago.

Potential solutions include improving automated braking systems and self-driving capabilities to avoid human error. Lowering speed limits on roads with frequent accidents and incentivizing not driving in inclement weather would help lower accident fatalities and injuries on the whole. Alternatively, investing more in city roads to improve conditions in cold weather may reduce the risk of injuries. We believe that if the city council proposes statistically-enforced changes to the law and helps lobby for better roads and vehicles, the number of accidents will reduce drastically.