Effect of Home Cornhusker Games on Public Safety

Scott Breitbach

8/3/2020

Section 1: Getting Started

Introduction:

Lincoln, Nebraska is the home of the Cornhuskers and every fall, Husker football is a huge draw to the city of Lincoln. When there are home football games, Memorial Stadium, which has a seating capacity of 85,458 and has sold out the last 375 games, essentially becomes the third largest city in Nebraska.

In a city of just under 300,000 people, I would like to see what impact, if any, this huge influx of people has on public safety, including crime and traffic incidents.

Research questions:

1. Does crime increase on Husker game weekends compared to non-game weekends?
2. Do traffic incidents increase on game weekends compared to non-game weekends?

If yes to 1 or 2:

- Which day of the week has the largest change from baseline?
- Does game time have an impact (morning or evening game)?
- Is there a difference in before-game vs after-game?
- Does the opposing team have an impact?
- Is there an opposite effect on away-game weeks when many people travel to watch the game?

Approach:

I plan to look at several years of crime and traffic data (at least 2017-2019) as well as Husker football game date and time information to generate crime and traffic incident baselines for non-Husker home game weeks and compare the baselines to Husker home game weeks to evaluate for any effect.

How this approach addresses the problem:

This approach will be exploratory in nature, in an effort to determine whether there is a corresponding increase in crime and traffic incidences alongside the increase in individuals to the city.

Data:

All datasets were obtained through the Open Data and Performance Management page for the City of Lincoln.

Files:

- LPD_2013_2020_Arrests_and_Citations_De_Coded.csv
- LPD_2017_2020_Incident_Reports.csv
- LPD_Traffic_Crashes_2013_2020.csv

Traffic Stops may also be considered, though those .csv files are compiled by individual year.

All data was created by the Lincoln Police Department and compiled via the LPD Records Management System. The Incident Report data has 25 variables and 116,263 observations. When loading this dataset, there were 50 parsing failures, so I will have to investigate and see if I can solve the issue, or just exclude those data points if I can't. The Traffic Crashes data has 18 variables and 24,500 observations, and the Arrests and Citations data has 366,000 observations of 18 variables.

The meaning of many of the variables is unclear, but I will primarily be looking at total incidences and the dates and times on which they occurred. I will need to find or generate my own dataset with the dates and times of the Cornhusker football games, whether they were home or away, and who they played. I will also need to convert to a common date time format between files in order to perform the analysis.

Required packages:

The following packages will be needed

- readr
- ggplot2
- lubridate
- dplyr
- hmisc
- car

Plots and table needs:

A chart of incidents over time (perhaps by week) could be useful to illustrate whether there are spikes on game weeks. If there is an increase in incidents on game weeks, bar charts showing the average increase by opposing team, day of the week, pre- or post-game, or any other correlations that reveal themselves during analysis.

Questions for future steps:

I will need a refresher on converting dates to a common format as well as a way to compare the date ranges between datasets.

One thing that I would like to learn more about is creating my own functions. Perhaps I will get a chance to do that within the scope of this project.

Section 2: Cleaning Data and Exploratory Data Analysis

```r
# Load game data
Huskers_2019 <- read_csv("data/Huskers_2019.csv")
Huskers_2018 <- read_csv("data/Huskers_2018.csv")
Huskers_2017 <- read_csv("data/Huskers_2017.csv")
Huskers_2016 <- read_csv("data/Huskers_2016.csv")
Huskers_2015 <- read_csv("data/Huskers_2015.csv")
Huskers_2014 <- read_csv("data/Huskers_2014.csv")
Huskers_2013 <- read_csv("data/Huskers_2013.csv")
```

```r
# Combine data
Husker_games <- rbind(Huskers_2013, Huskers_2014, Huskers_2015, Huskers_2016,
                      Huskers_2017, Huskers_2018, Huskers_2019)

# Remove ranking from school names
Husker_games$School <- gsub("[[:digit:]]", "", Husker_games$School)
Husker_games$School <- gsub("[[:punct:]]+[[:punct:]] ", "", Husker_games$School)
Husker_games$Opponent <- gsub("[[:digit:]]", "", Husker_games$Opponent)
Husker_games$Opponent <- gsub("[[:punct:]]+[[:punct:]] ", "",
                              Husker_games$Opponent)

# Rename unnamed columns
names(Husker_games)[6] <- "Location"
names(Husker_games)[9] <- "Win"

# Convert W/L and NA/@ to 1/0
Husker_games$Location[is.na(Husker_games$Location)] <- "Home"
Husker_games$Location[Husker_games$Location=="@"] <- "Away"

# Remove extra columns
Husker_games <- Husker_games[c("Date", "Time", "Day", "School", "Location",
                               "Opponent", "Win")]

# Convert dates
Husker_games$Date <- mdy(Husker_games$Date)
Husker_games$Day <- wday(Husker_games$Date, label = TRUE)

# Keep only months: Sept, Oct, Nov
Husker_games <- Husker_games[month(Husker_games$Date) >= 9 &
                               month(Husker_games$Date) <= 11, ]

# Convert character variables to factors
Husker_games$Day <- as.factor(Husker_games$Day)
Husker_games$Location <- as.factor(Husker_games$Location)
Husker_games$Opponent <- as.factor(Husker_games$Opponent)
Husker_games$Win <- as.factor(Husker_games$Win)

# Add Year and Week columns & Rearrange Columns
Husker_games$Year <- year(Husker_games$Date)
Husker_games$Week <- isoweek(Husker_games$Date)
Husker_games <- Husker_games[c("Date", "Time", "Year", "Week", "Day",
                               "Location", "School", "Opponent", "Win")]

# Cleanup
```

```r
rm(Huskers_2013, Huskers_2014, Huskers_2015, Huskers_2016, Huskers_2017,
   Huskers_2018, Huskers_2019)
```

```r
## Arrests and Citations
arr_cit_13 <- read_csv("data/LPD_arrests_and_citations_2013.csv")
arr_cit_14 <- read_csv("data/LPD_arrests_and_citations_2014.csv")
arr_cit_15 <- read_csv("data/LPD_arrests_and_citations_2015.csv")
arr_cit_16 <- read_csv("data/LPD_arrests_and_citations_2016.csv")
arr_cit_17 <- read_csv("data/LPD_arrests_and_citations_2017.csv")
arr_cit_18 <- read_csv("data/LPD_arrests_and_citations_2018.csv")
arr_cit_19 <- read_csv("data/LPD_arrestd_and_citations_2019.csv")

## Incident Reports
incidents_2017_2020 <- read_csv("data/LPD_2017_2020_Incident_Reports.csv")
incidents_2016 <- read_csv("data/LPD_Incident_Reports_2016.csv")
incidents_2015 <- read_csv("data/LPD_Incident_Reports_2015.csv")
incidents_2014 <- read_csv("data/LPD_Incident_Reports_2014.csv")
incidents_2013 <- read_csv("data/LPD_Incident_Reports_2013.csv")

## Traffic Crashes
Trf_Crash_13 <- read_csv("data/Traffic_Crashes_2013.csv")
Trf_Crash_14 <- read_csv("data/Traffic_Crashes_2014.csv")
Trf_Crash_15 <- read_csv("data/Traffic_Crashes_2015.csv")
Trf_Crash_16 <- read_csv("data/Traffic_Crashes_2016.csv")
Trf_Crash_17 <- read_csv("data/Traffic_Crashes_2017.csv")
Trf_Crash_18 <- read_csv("data/Traffic_Crashes_2018.csv")
Trf_Crash_19 <- read_csv("data/Traffic_Crashes_2019.csv")

## Traffic Stops
Trf_Stop_13 <- read_csv("data/LPD_traffic_stops_2013.csv")
Trf_Stop_14 <- read_csv("data/LPD_traffic_stops_2014.csv")
Trf_Stop_15 <- read_csv("data/LPD_traffic_stops_2015.csv")
Trf_Stop_16 <- read_csv("data/LPD_traffic_stops_2016.csv")
Trf_Stop_17 <- read_csv("data/LPD_traffic_stops_2017.csv")
Trf_Stop_18 <- read_csv("data/LPD_traffic_stops_2018.csv")
Trf_Stop_19 <- read_csv("data/LPD_traffic_stops_2019.csv")
```

```r
## ARRESTS AND CITATIONS
# Remove / rename columns and merge
names(arr_cit_18)[15] <- "FID"
a_c.13_18 <- rbind(arr_cit_13, arr_cit_14, arr_cit_15, arr_cit_16, arr_cit_17,
                   arr_cit_18)
a_c.13_18 <- a_c.13_18[c("CHARGED", "VDAT", "VTIM")]
arr_cit_19 <- arr_cit_19[c("CHARGED", "VDAT", "VTIM")]
a_c.13_19 <- rbind(a_c.13_18, arr_cit_19)
names(a_c.13_19)[1:3] <- c("Charge","Date", "Time")
# Parse dates & times
a_c.13_19$Date <- parse_date(a_c.13_19$Date, "%Y/%m/%d %H:%M:%S+00")
a_c.13_19$Time <- str_pad(a_c.13_19$Time, 4, pad = "0")
a_c.13_19$Time <- parse_time(a_c.13_19$Time, "%H%M")
# Remove all but Sep, Oct, Nov
a_c.13_19 <- a_c.13_19[month(a_c.13_19$Date) >= 9 & month(a_c.13_19$Date) <= 11, ]
# Remove dates before 2013 or after 2019
```

```r
a_c.13_19 <- a_c.13_19[year(a_c.13_19$Date) >= 2013 & year(a_c.13_19$Date) <= 2019, ]
# Remove NAs
a_c.13_19 <- na.omit(a_c.13_19)
# Day of week column
a_c.13_19$Day <- wday(a_c.13_19$Date, label = TRUE)

## INCIDENT REPORTS
# 2017-2020 data set:
#  Remove / Rename columns
inc.17_20 <- incidents_2017_2020[c("CALL_TYPE", "From_Date", "From_Time")]
names(inc.17_20)[1:3] <- c("Type", "Date", "Time")
#  Parse dates
inc.17_20$Date <- ymd(inc.17_20$Date)
# 2015 & 2016 data sets:
inc.15_16 <- rbind(incidents_2015, incidents_2016)
#  Remove / Rename columns
inc.15_16 <- inc.15_16[c("CALL_TYPE", "DATE_FROM", "TIME_FROM")]
names(inc.15_16)[1:3] <- c("Type", "Date", "Time")
#  Parse dates
inc.15_16$Date <- parse_date(inc.15_16$Date, "%Y/%m/%d %H:%M:%S+00")
# 2013 & 2014 data sets:
inc.13_14 <- rbind(incidents_2013, incidents_2014)
#  Remove / Rename columns
inc.13_14 <- inc.13_14[c("CALL_TYPE", "DATE_FROM", "TIME_FROM")]
names(inc.13_14)[1:3] <- c("Type", "Date", "Time")
#  Parse dates
inc.13_14$Date <- mdy(inc.13_14$Date)
# Combine data sets (2013-2020):
inc.13_20 <- rbind(inc.13_14, inc.15_16, inc.17_20)
#  Parse times
inc.13_20$Time <- str_pad(inc.13_20$Time, 4, pad = "0")
inc.13_20$Time <- parse_time(inc.13_20$Time, "%H%M")
#  Remove all but Sep, Oct, Nov
inc.13_20 <- inc.13_20[month(inc.13_20$Date) >= 9 & month(inc.13_20$Date) <= 11, ]
#  Remove dates before 2013 or after 2019
inc.13_19 <- inc.13_20[year(inc.13_20$Date) >= 2013 & year(inc.13_20$Date) <= 2019, ]
#  Remove NAs
inc.13_19 <- na.omit(inc.13_19)
#  Add Day of week column
inc.13_19$Day <- wday(inc.13_19$Date, label = TRUE)

## TRAFFIC CRASHES
#  Rename columns and merge
names(Trf_Crash_18)[16] <- "FID"
names(Trf_Crash_19)[16] <- "FID"
t_c.13_18 <- rbind(Trf_Crash_13, Trf_Crash_14, Trf_Crash_15, Trf_Crash_16,
                   Trf_Crash_17, Trf_Crash_18)
#  Format dates and merge
t_c.13_18$DOA <- parse_date(t_c.13_18$DOA, "%Y/%m/%d %H:%M:%S+00")
Trf_Crash_19$DOA <- as.POSIXct(Trf_Crash_19$DOA/1000, origin = "1970-01-01")
Trf_Crash_19$DOA <- as.character(Trf_Crash_19$DOA)
Trf_Crash_19$DOA <- parse_date(Trf_Crash_19$DOA, "%Y-%m-%d %H:%M:%S")
t_c.13_19 <- rbind(t_c.13_18, Trf_Crash_19)
```

```r
#  Remove / Rename columns
t_c.13_19 <- t_c.13_19[c("TYPE", "ACTION", "PED", "BIKE", "MC", "MOPED", "TRAIN",
                         "TRUCK", "BUS", "DOA", "TOA")]
names(t_c.13_19)[1:11] <- c("Type", "Action", "Pedestrian", "Bike", "Motorcycle",
                            "Moped", "Train", "Truck", "Bus", "Date", "Time")
#  Remove all but Sep, Oct, Nov
t_c.13_19 <- t_c.13_19[month(t_c.13_19$Date) >= 9 & month(t_c.13_19$Date) <= 11, ]
#  Remove dates before 2013 or after 2019
t_c.13_19 <- t_c.13_19[year(t_c.13_19$Date) >= 2013 & year(t_c.13_19$Date) <= 2019, ]
#  Parse times
t_c.13_19$Time <- str_pad(t_c.13_19$Time, 4, pad = "0")
t_c.13_19$Time <- parse_time(t_c.13_19$Time, "%H%M")
#  Remove NAs
t_c.13_19 <- na.omit(t_c.13_19)
#  Add Day of week column
t_c.13_19$Day <- wday(t_c.13_19$Date, label = TRUE)

## TRAFFIC STOPS
#  Rename columns and merge
names(Trf_Stop_14)[4] <- "SEX"
names(Trf_Stop_18)[8] <- "FID"
t_s.13.16 <- rbind(Trf_Stop_13, Trf_Stop_16)
t_s.14_15 <- rbind(Trf_Stop_14, Trf_Stop_15)
t_s.17_19 <- rbind(Trf_Stop_17, Trf_Stop_18, Trf_Stop_19)
# Parse times and merge
t_s.13.16$TIME <- parse_time(t_s.13.16$TIME)
t_s.14_15$TIME <- parse_time(t_s.14_15$TIME, "%Y/%M/%D %H:%M:%S+00")
t_s.17_19$TIME <- gsub(":XX", "", t_s.17_19$TIME)
t_s.17_19$TIME <- parse_time(t_s.17_19$TIME, "%H:%M")
t_s.13_19 <- rbind(t_s.13.16, t_s.14_15, t_s.17_19)
#  Remove / Rename columns
t_s.13_19 <- t_s.13_19[c("REASON", "DATE", "TIME")]
names(t_s.13_19)[1:3] <- c("Reason", "Date", "Time")
#  Parse dates
t_s.13_19$Date <- parse_date(t_s.13_19$Date, "%Y/%m/%d %H:%M:%S+00")
#  Remove all but Sep, Oct, Nov
t_s.13_19 <- t_s.13_19[month(t_s.13_19$Date) >= 9 & month(t_s.13_19$Date) <= 11, ]
#  Remove dates before 2013 or after 2019
t_s.13_19 <- t_s.13_19[year(t_s.13_19$Date) >= 2013 & year(t_s.13_19$Date) <= 2019, ]
#  Remove NAs
t_s.13_19 <- na.omit(t_s.13_19)
#  Add Day of week column
t_s.13_19$Day <- wday(t_s.13_19$Date, label = TRUE)


## CLEANUP
rm(arr_cit_13, arr_cit_14, arr_cit_15, arr_cit_16, arr_cit_17, arr_cit_18, arr_cit_19)
rm(a_c.13_18)
rm(incidents_2017_2020, incidents_2016, incidents_2015, incidents_2014, incidents_2013)
rm(inc.17_20, inc.15_16, inc.13_14, inc.13_20)
rm(Trf_Crash_13, Trf_Crash_14, Trf_Crash_15, Trf_Crash_16, Trf_Crash_17,
   Trf_Crash_18, Trf_Crash_19)
rm(t_c.13_18)
```

```r
rm(Trf_Stop_13, Trf_Stop_14, Trf_Stop_15, Trf_Stop_16, Trf_Stop_17, Trf_Stop_18,
   Trf_Stop_19)
rm(t_s.13.16, t_s.14_15, t_s.17_19)
```

```r
# Add Yr-Week columns to data sets
Husker_games$Yr_Wk <- paste(year(Husker_games$Date), isoweek(Husker_games$Date),
                            sep = "-")
a_c.13_19$Yr_Wk <- paste(year(a_c.13_19$Date), isoweek(a_c.13_19$Date), sep = "-")
inc.13_19$Yr_Wk <- paste(year(inc.13_19$Date), isoweek(inc.13_19$Date), sep = "-")
t_c.13_19$Yr_Wk <- paste(year(t_c.13_19$Date), isoweek(t_c.13_19$Date), sep = "-")
t_s.13_19$Yr_Wk <- paste(year(t_s.13_19$Date), isoweek(t_s.13_19$Date), sep = "-")
# Add total occurrences for Public Safety Incidents by week to Husker data
a_c.occur <- table(unlist(a_c.13_19$Yr_Wk))
Husker_games$A_C <- a_c.occur[Husker_games$Yr_Wk]
Inc.occur <- table(unlist(inc.13_19$Yr_Wk))
Husker_games$Inc <- Inc.occur[Husker_games$Yr_Wk]
t_c.occur <- table(unlist(t_c.13_19$Yr_Wk))
Husker_games$T_C <- t_c.occur[Husker_games$Yr_Wk]
t_s.occur <- table(unlist(t_s.13_19$Yr_Wk))
Husker_games$T_S <- t_s.occur[Husker_games$Yr_Wk]
# Convert new columns to numeric
Husker_games$A_C <- as.numeric(Husker_games$A_C)
Husker_games$Inc <- as.numeric(Husker_games$Inc)
Husker_games$T_C <- as.numeric(Husker_games$T_C)
Husker_games$T_S <- as.numeric(Husker_games$T_S)
# Add column for Total Incidents
Husker_games$Tot_Inc <- rowSums(Husker_games[c("A_C", "Inc", "T_C", "T_S")])
# Cleanup
rm(a_c.occur, Inc.occur, t_c.occur, t_s.occur)
```

```r
"Game Data"
```

```
## [1] "Game Data"
```

```r
head(Husker_games)
```

```
## # A tibble: 6 x 15
##   Date       Time    Year  Week Day   Location School Opponent Win   Yr_Wk   A_C
##   <date>     <time> <dbl> <dbl> <ord> <fct>    <chr>  <fct>    <fct> <chr> <dbl>
## 1 2013-09-07 18:00   2013    36 Sat   Home     Nebra~ Souther~ W     2013~  1404
## 2 2013-09-14 12:00   2013    37 Sat   Home     Nebra~ UCLA     L     2013~  1415
## 3 2013-09-21 15:30   2013    38 Sat   Home     Nebra~ South D~ W     2013~  1290
## 4 2013-10-05 12:00   2013    40 Sat   Home     Nebra~ Illinois W     2013~  1306
## 5 2013-10-12 12:00   2013    41 Sat   Away     Nebra~ Purdue   W     2013~  1216
## 6 2013-10-26 12:00   2013    43 Sat   Away     Nebra~ Minneso~ L     2013~  1122
## # ... with 4 more variables: Inc <dbl>, T_C <dbl>, T_S <dbl>, Tot_Inc <dbl>
```

```r
"Arrests and Citations"
```

```
## [1] "Arrests and Citations"
```

```r
head(a_c.13_19)
```

```
## # A tibble: 6 x 5
##   Charge                                     Date       Time   Day   Yr_Wk
##   <chr>                                      <date>     <time> <ord> <chr>
## 1 DUI-2ND >.15                               2013-09-01 00:47  Sun   2013-35
## 2 NEGLIGENT DRIVING                          2013-09-01 00:47  Sun   2013-35
## 3 DISTURBING THE PEACE                       2013-09-01 02:44  Sun   2013-35
## 4 POSS MARIJ,1 OZ/LESS OR SYNTHETIC MARIJ-1ST 2013-09-01 02:44  Sun   2013-35
## 5 POSS MARIJ,1 OZ/LESS OR SYNTHETIC MARIJ-1ST 2013-09-01 02:50  Sun   2013-35
## 6 ARRESTED ON COUNTY BENCH WARRANT           2013-09-01 08:18  Sun   2013-35
```

```r
"Incident Reports"
```

```
## [1] "Incident Reports"
```

```r
head(inc.13_19)
```

```
## # A tibble: 6 x 5
##   Type       Date       Time   Day   Yr_Wk
##   <chr>      <date>     <time> <ord> <chr>
## 1 SEX OFF    2013-11-01 00:01  Fri   2013-44
## 2 SEX OFF    2013-11-01 00:01  Fri   2013-44
## 3 SUSP ITEM  2013-09-11 08:00  Wed   2013-37
## 4 STALKING   2013-11-12 07:00  Tue   2013-46
## 5 SELL NARCO 2013-09-19 20:35  Thu   2013-38
## 6 SUSP ITEM  2013-10-07 12:00  Mon   2013-41
```

```r
"Traffic Crashes"
```

```
## [1] "Traffic Crashes"
```

```r
head(t_c.13_19)
```

```
## # A tibble: 6 x 13
##   Type   Action      Pedestrian Bike  Motorcycle Moped Train Truck Bus   Date
##   <chr>  <chr>       <chr>      <chr> <chr>      <chr> <chr> <chr> <chr> <date>
## 1 INJURY REAR END    NO         NO    YES        NO    NO    NO    NO    2013-11-16
## 2 INJURY DRIVEWAY    NO         NO    YES        NO    NO    YES   NO    2013-11-29
## 3 INJURY OTHER       NO         YES   YES        NO    NO    NO    NO    2013-10-10
## 4 INJURY RIGHT ANGLE NO         YES   NO         NO    NO    NO    NO    2013-09-02
## 5 INJURY OTHER       YES        NO    YES        NO    NO    NO    NO    2013-10-01
## 6 INJURY DRIVEWAY    NO         YES   NO         NO    NO    NO    NO    2013-09-02
## # ... with 3 more variables: Time <time>, Day <ord>, Yr_Wk <chr>
```

```r
"Traffic Stops"
```

```
## [1] "Traffic Stops"
```

```r
head(t_s.13_19)
```

```
## # A tibble: 6 x 5
##   Reason Date       Time   Day   Yr_Wk
##    <dbl> <date>     <time> <ord> <chr>
## 1      1 2013-09-02 03:07  Mon   2013-36
## 2      1 2013-09-01 02:30  Sun   2013-35
## 3      2 2013-09-01 02:36  Sun   2013-35
## 4      1 2013-09-02 03:50  Mon   2013-36
## 5      1 2013-09-02 18:25  Mon   2013-36
## 6      1 2013-09-03 08:11  Tue   2013-36
```

Section 3: Final Project

NOTE: Variables for the Public Safety data are defined as follows:

- **A_C: Arrests and Citations**
- **Inc: Incident Reports completed**
- **T_C: Traffic Crashes reported**
- **T_S: Traffic (and Pedestrian) Stop records**

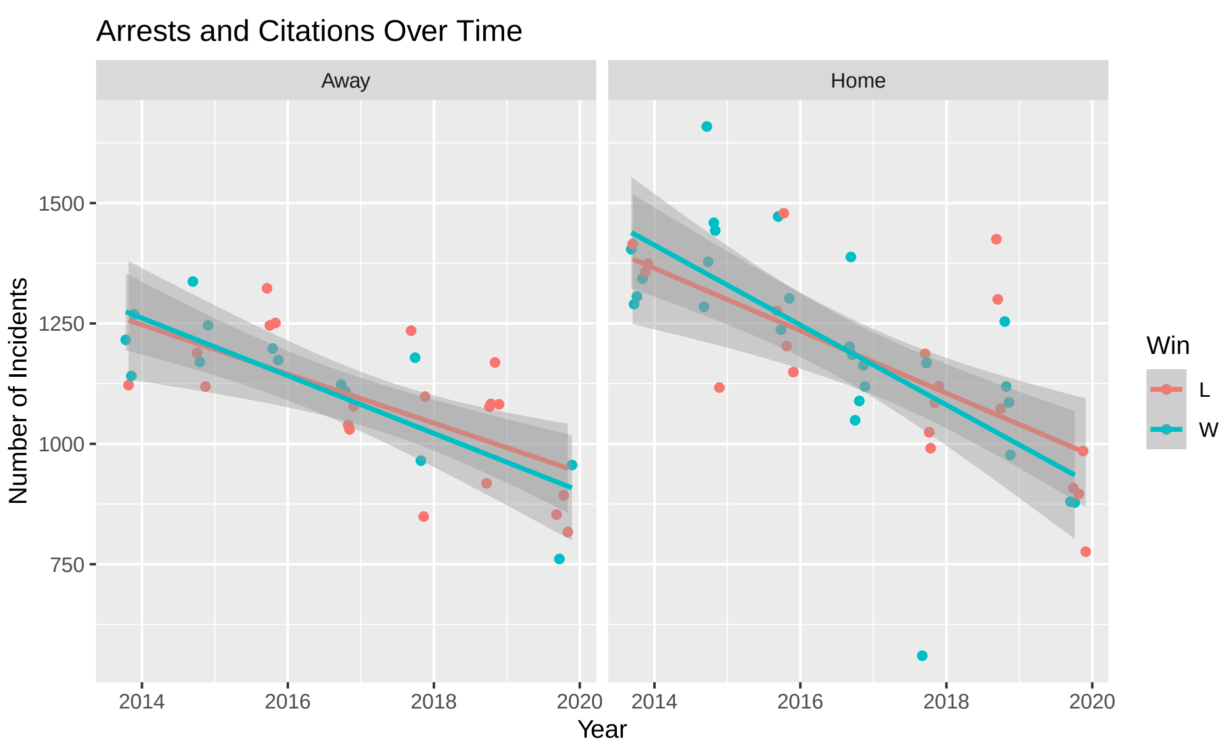
```r
# Melt Police data columns to long format
H_G.long <- melt(Husker_games, id = c("Date", "Year", "Week", "Day", "Time", "School",
                                      "Location", "Opponent", "Win", "Yr_Wk", "Tot_Inc"),
                 measured = c("A_C", "Inc", "T_C", "T_S"))
# Code Win and Location variables to 0s and 1s
H_G.coded <- Husker_games
H_G.coded$Location <- as.numeric(as.factor(H_G.coded$Location))-1
H_G.coded$Win <- as.numeric(as.factor(H_G.coded$Win))-1
```

Once the data was cleaned, I still needed to work with some of it in different ways. To that end, I created a couple additional data frames. In one, *H_G.long*, I took the Public Safety totals that I had added to my Husker Games data set from each category and "melted" them into just two columns, represented by "variable" and "value". This helped in creating charts that included all four categories instead of working with them individually.

The other data set, *H_G.coded*, I created from the Husker Games dataset by converting some of the character variables, such as Location (Home/Away) and Win (W/L) to binary 1's and 0's so that I could run correlation with them. Once set up, I started playing around with different ways to visualize and explore the data:

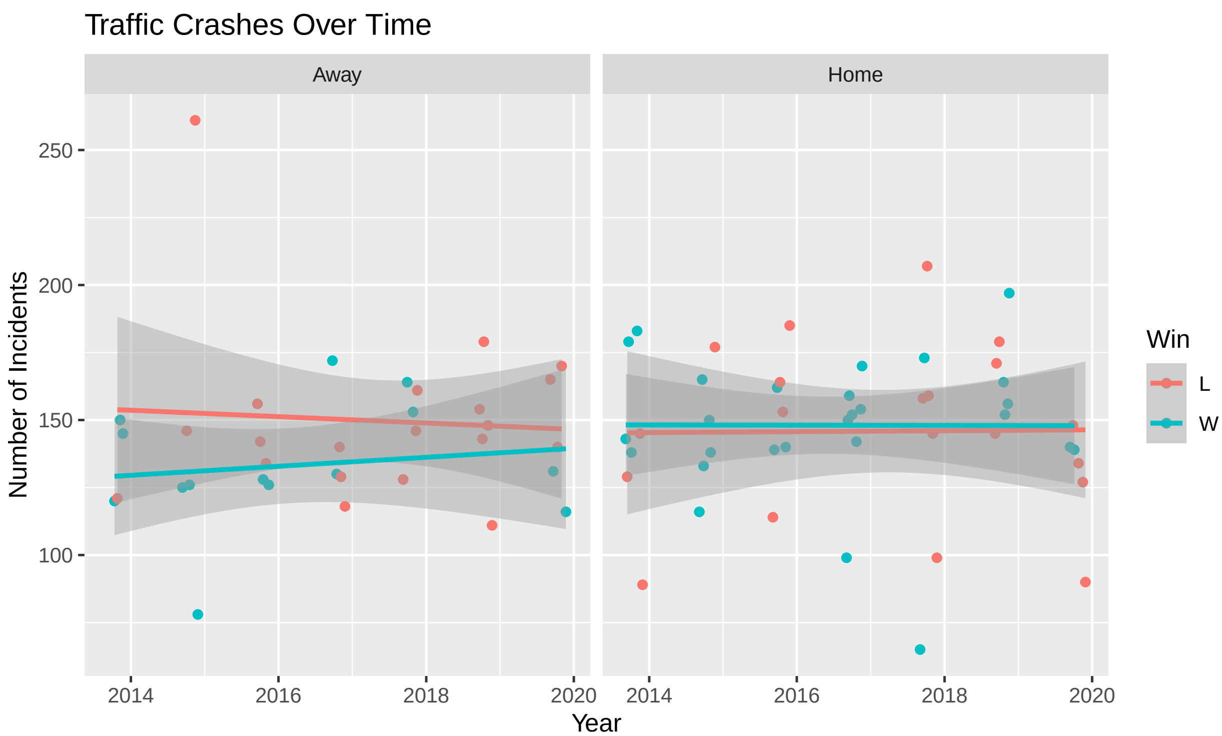
```r
# SCATTERPLOT: Shows change over time for variables, faceted by W/L & Home/Away; shows all steady or de
ggplot(H_G.long, aes(x = Date, y = value, color = variable)) + geom_point() +
  geom_smooth(method = lm) + facet_grid(Location ~ Win) +
  labs(title = "Public Safety Incidents Over Time", x = "Year",
       y = "Number of Incidents", color = "Incident Type")
```


```r
# SCATTERPLOT: Compares Win/Loss over time for each variable, faceted by Home/Away; for the most part W
ggplot(Husker_games, aes(x = Date, y = A_C, color = Win)) + geom_point() +
  geom_smooth(method = lm) + facet_wrap(~ Location) +
  labs(title = "Arrests and Citations Over Time", x = "Year",
       y = "Number of Incidents")
```

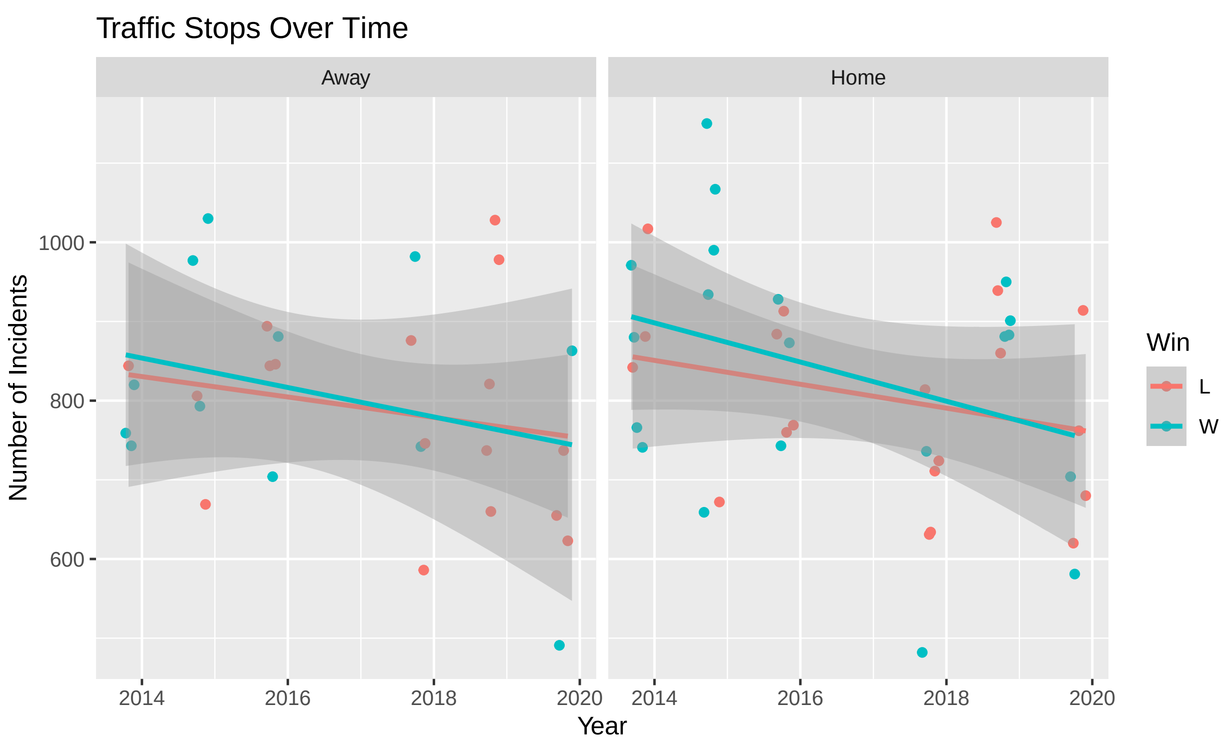


```r
ggplot(Husker_games, aes(x = Date, y = Inc, color = Win)) + geom_point() +
  geom_smooth(method = lm) + facet_wrap(~ Location) +
  labs(title = "Incident Reports Over Time", x = "Year",
       y = "Number of Incidents")
```


```r
ggplot(Husker_games, aes(x = Date, y = T_C, color = Win)) + geom_point() +
  geom_smooth(method = lm) + facet_wrap(~ Location) +
  labs(title = "Traffic Crashes Over Time", x = "Year",
       y = "Number of Incidents")
```



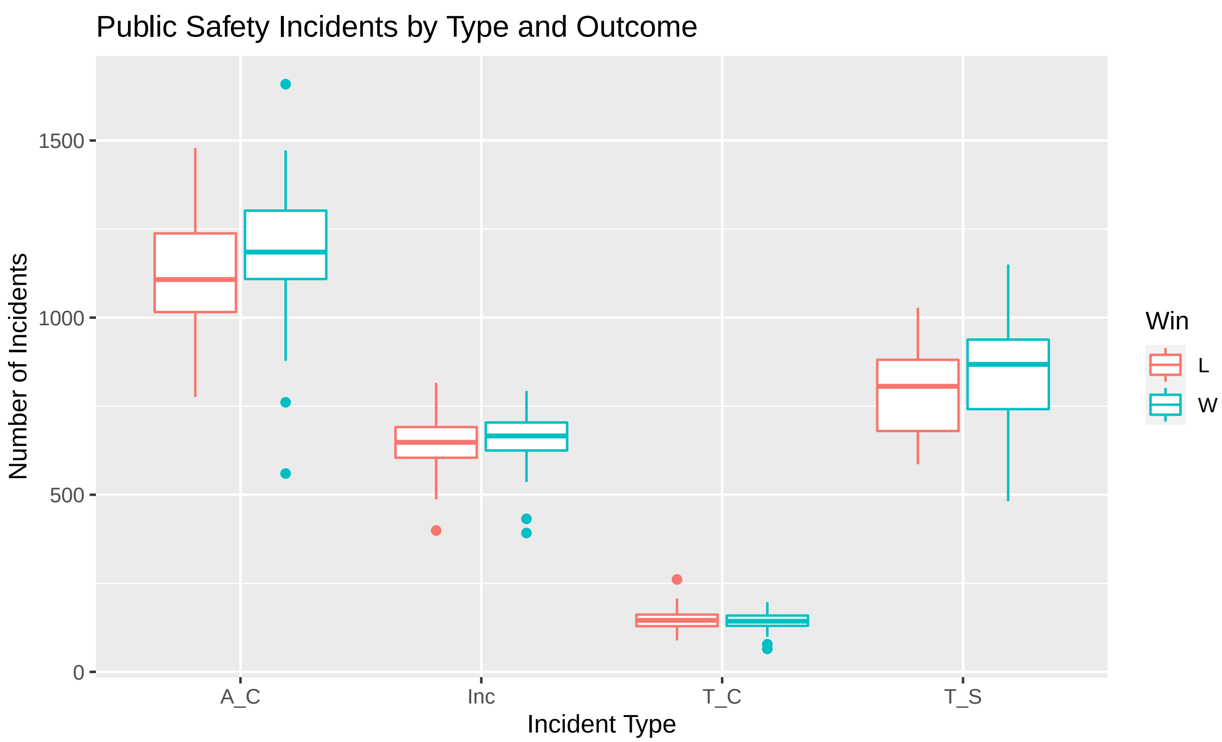

```r
ggplot(Husker_games, aes(x = Date, y = T_S, color = Win)) + geom_point() +
  geom_smooth(method = lm) + facet_wrap(~ Location) +
  labs(title = "Traffic Stops Over Time", x = "Year",
       y = "Number of Incidents")
```



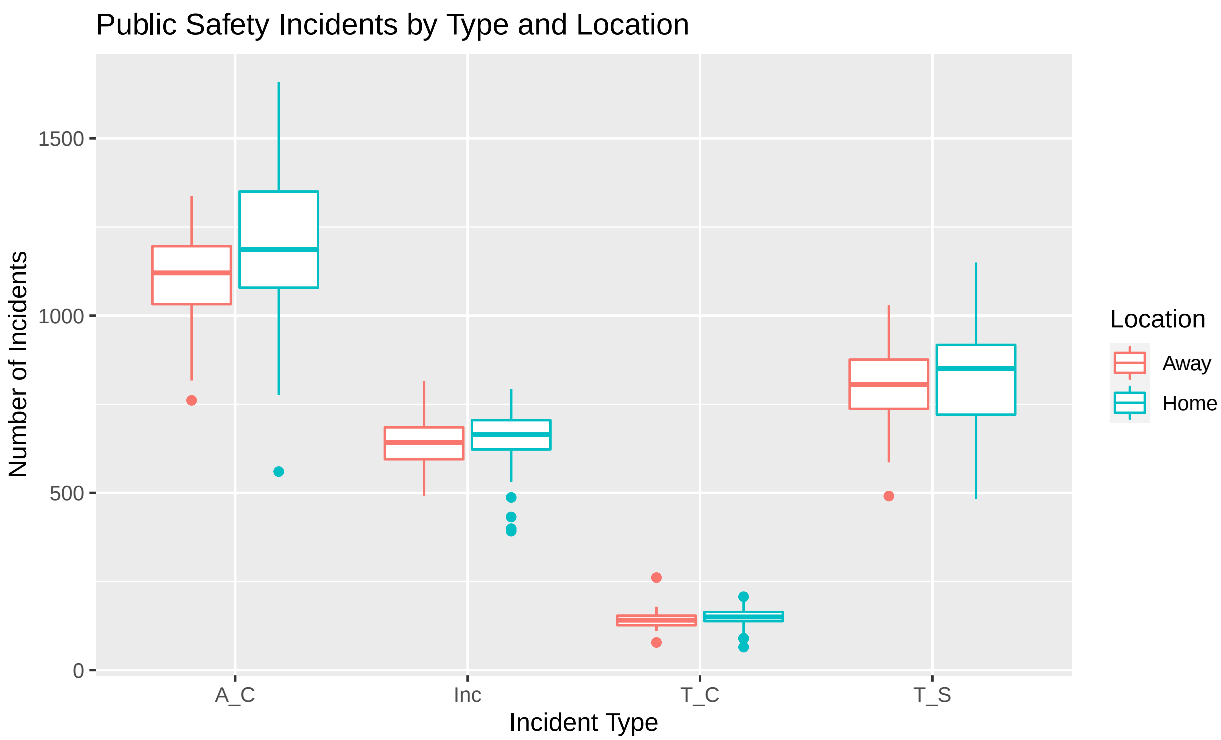
The first chart, with all four variables shows that for both Wins and Losses, Home and Away games, that Public Safety incidents are either generally remaining steady or decreasing over time, which is good news!

The remaining four charts compare Win and Loss data by date, faceted by Location. With the exception of "Incidents" at home games, all of the Win and Loss lines are either crossing or converging, indicating that whether the Huskers win or lose might not have much of an impact on Public Safety.

```
# BOXPLOTS: for all variables, colored by Win or Location
ggplot(H_G.long, aes(variable, value, color = Win)) + geom_boxplot() +
  labs(title = "Public Safety Incidents by Type and Outcome",
       x = "Incident Type", y = "Number of Incidents")
```

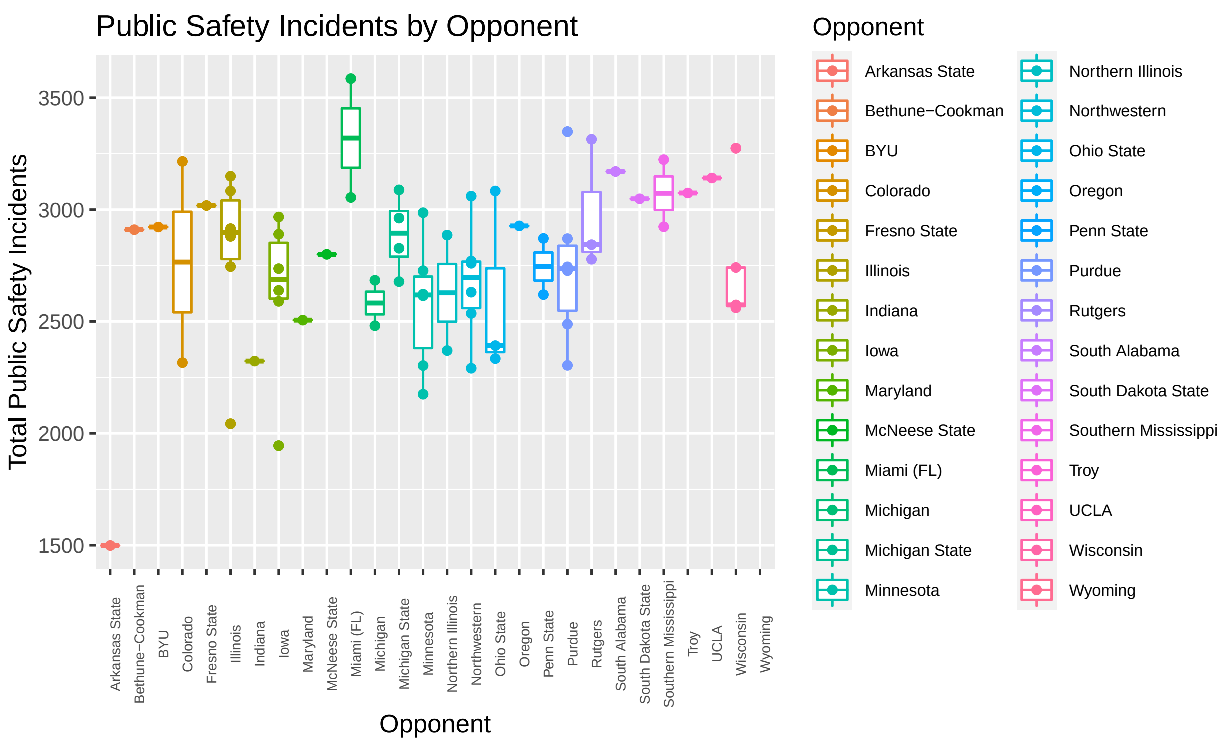



```
ggplot(H_G.long, aes(variable, value, color = Location)) + geom_boxplot() +
  labs(title = "Public Safety Incidents by Type and Location",
       x = "Incident Type", y = "Number of Incidents")
```




```
# BOXPLOT: for each location, colored by Win, faceted by variable (may not need previous two)
ggplot(H_G.long, aes(x = Location, color = Win)) +
  geom_boxplot(aes(y = value)) + facet_grid(~ variable) +
  labs(title = "Public Safety Incidents by Type, Location, and Outcome",
       x = element_blank(), y = "Number of Incidents")
```

```r
# BOXPLOTS: Colored by Opponent, with points
ggplot(Husker_games, aes(Opponent, Tot_Inc, color = Opponent)) +
  geom_boxplot() + geom_point() +
  theme(axis.text.x = element_text(angle = 90, size = 6),
        legend.text = element_text(size = 7)) +
  labs(title = "Public Safety Incidents by Opponent", x = "Opponent",
       y = "Total Public Safety Incidents")
```



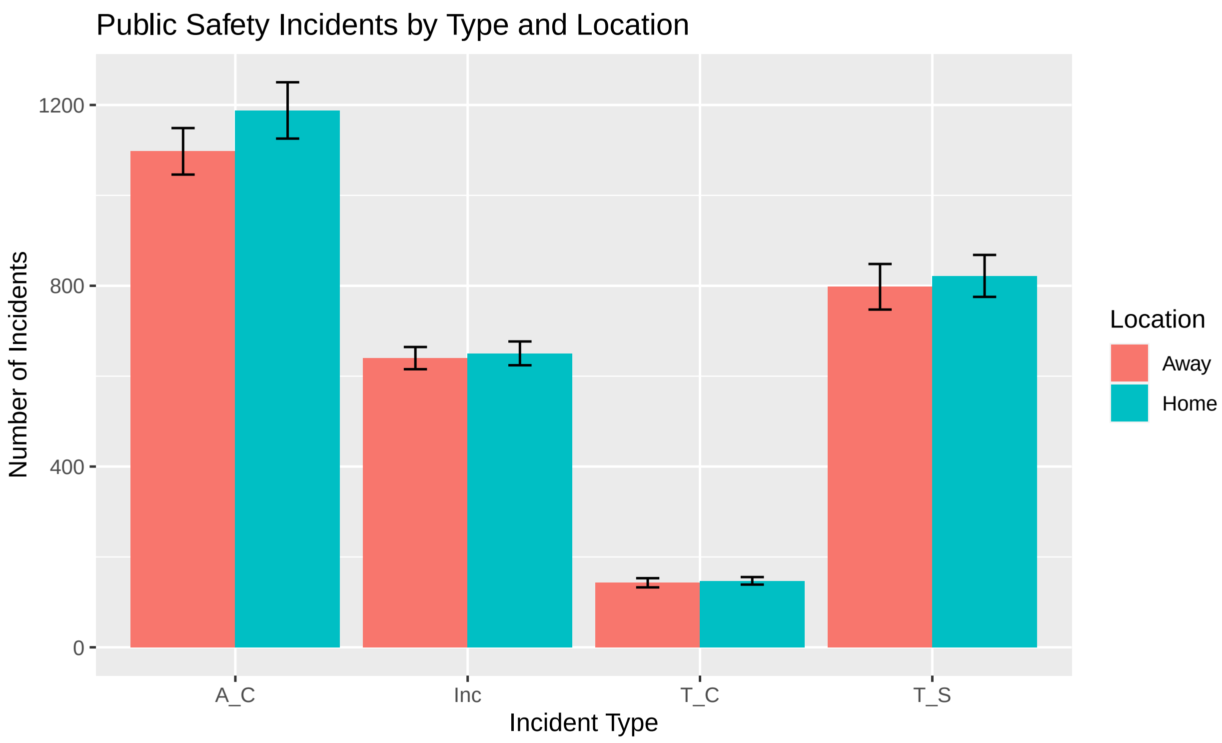

```r
# BOXPLOT: shows Total Incidents, Home and Away, colored by Opponent, with points
ggplot(Husker_games, aes(Opponent, Tot_Inc, color = Opponent)) +
  geom_boxplot() + geom_point(aes(shape = Win)) +
  theme(axis.text.x = element_text(angle = 90, size = 6),
        legend.text = element_text(size = 7)) + facet_wrap(~ Location) +
  labs(title = "Public Safety Incidents by Opponent and Location",
       x = "Opponent", y = "Total Public Safety Incidents")
```

Looking at the boxplots, you can see that for most of the Public Safety variables it doesn't appear to make much difference whether it is a Home or Away game, or whether it is a Win or a Loss. The exception to this appears to be Arrests and Citations, which is slightly higher for both Wins and for Home games.

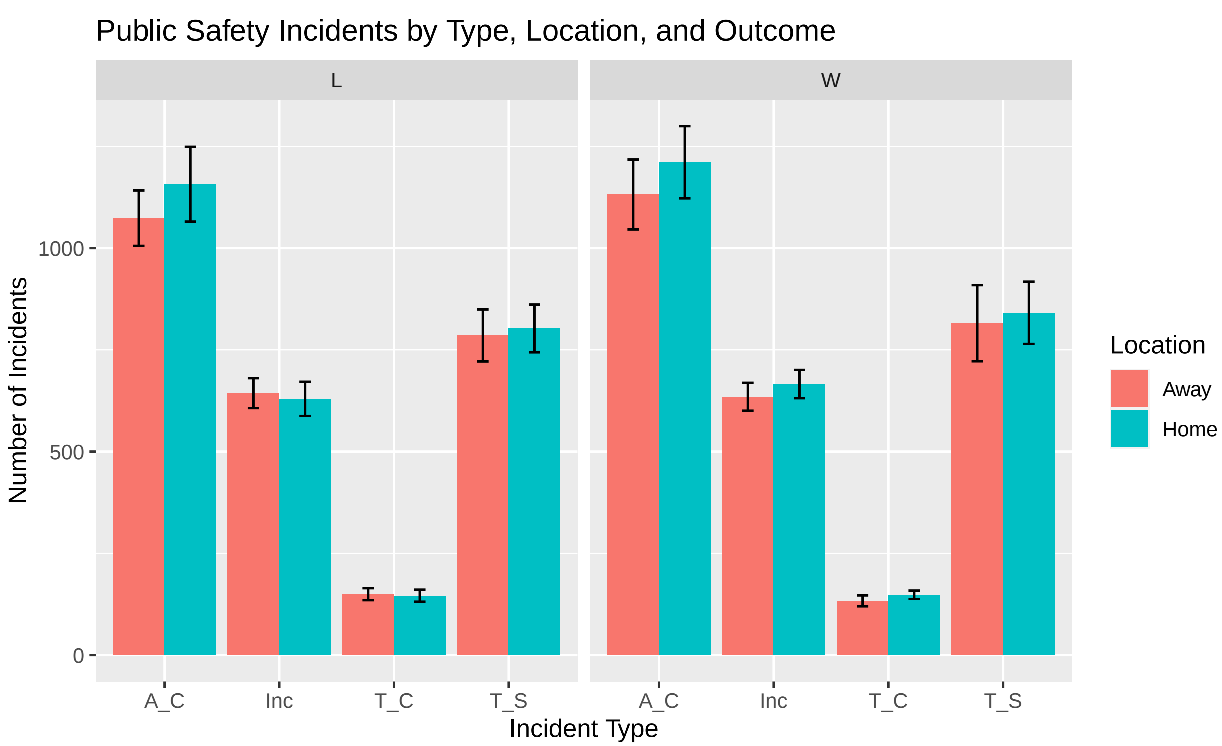
Looking at the different Opponents, it appears that Miami is the clear leader when it comes to total number of Public Safety Incidents. This seems to largely be accounted for in the single data point for Away games, where it lies as a lone point above all of the others. The one Home game against Miami isn't the highest, but does look to be in the top three. There are only two data points, but perhaps this could be something to explore further.

```
# BAR CHART: Comparing variables by w/L & Home/Away; not much difference in Home/Away when losing
ggplot(H_G.long, aes(variable, value, fill = variable)) +
  geom_bar(stat = "identity") + facet_grid(Location ~ Win) +
  labs(title = "Public Safety Incidents by Variable, Location, and Outcome",
       x = element_blank(), y = "Number of Incidents", fill = "Incident Type")
```

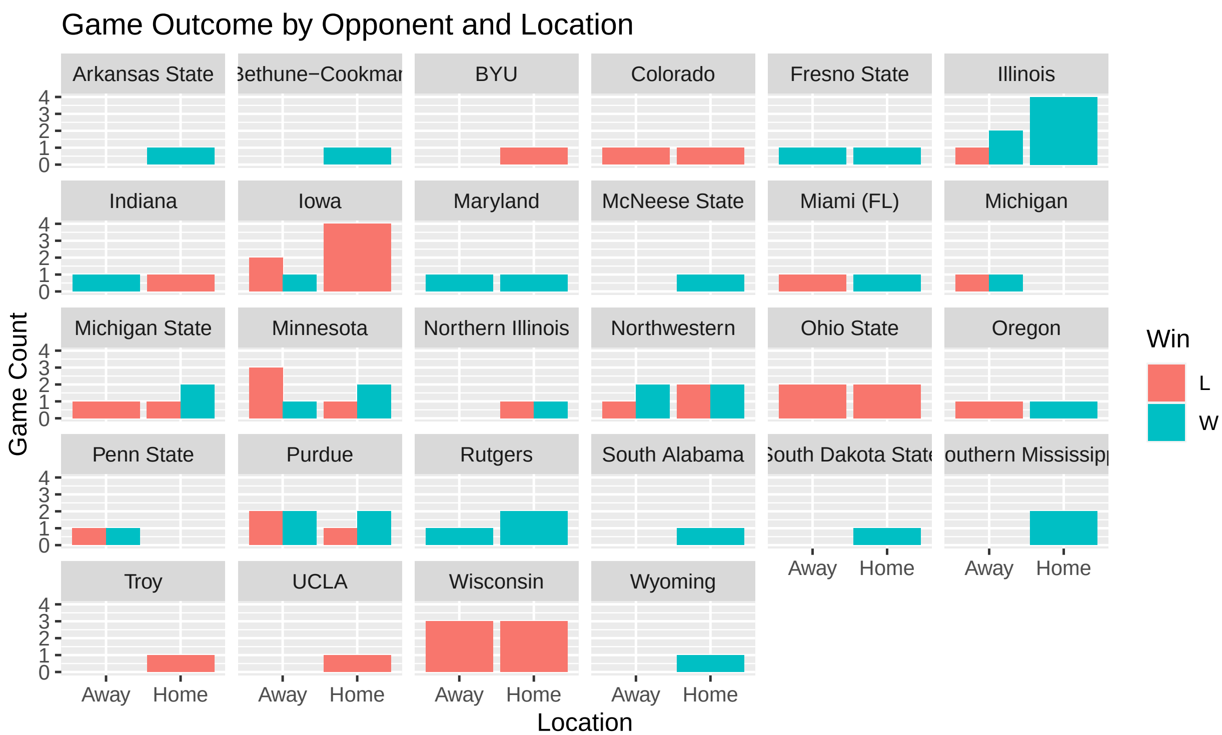

```r
# BAR CHARTS:
ggplot(H_G.long, aes(variable, value, fill = Location)) +
  stat_summary(fun = mean, geom = "bar", position = "dodge") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
               position = position_dodge(width = 0.90), width = 0.2) +
  labs(title = "Public Safety Incidents by Type and Location",
       x = "Incident Type", y = "Number of Incidents")
```




```
ggplot(H_G.long, aes(variable, value, fill = Location)) +
  stat_summary(fun = mean, geom = "bar", position = "dodge") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
               position = position_dodge(width = 0.90), width = 0.2) + facet_wrap(~ Win) +
  labs(title = "Public Safety Incidents by Type, Location, and Outcome", x = "Incident Type", y = "Numbe
```

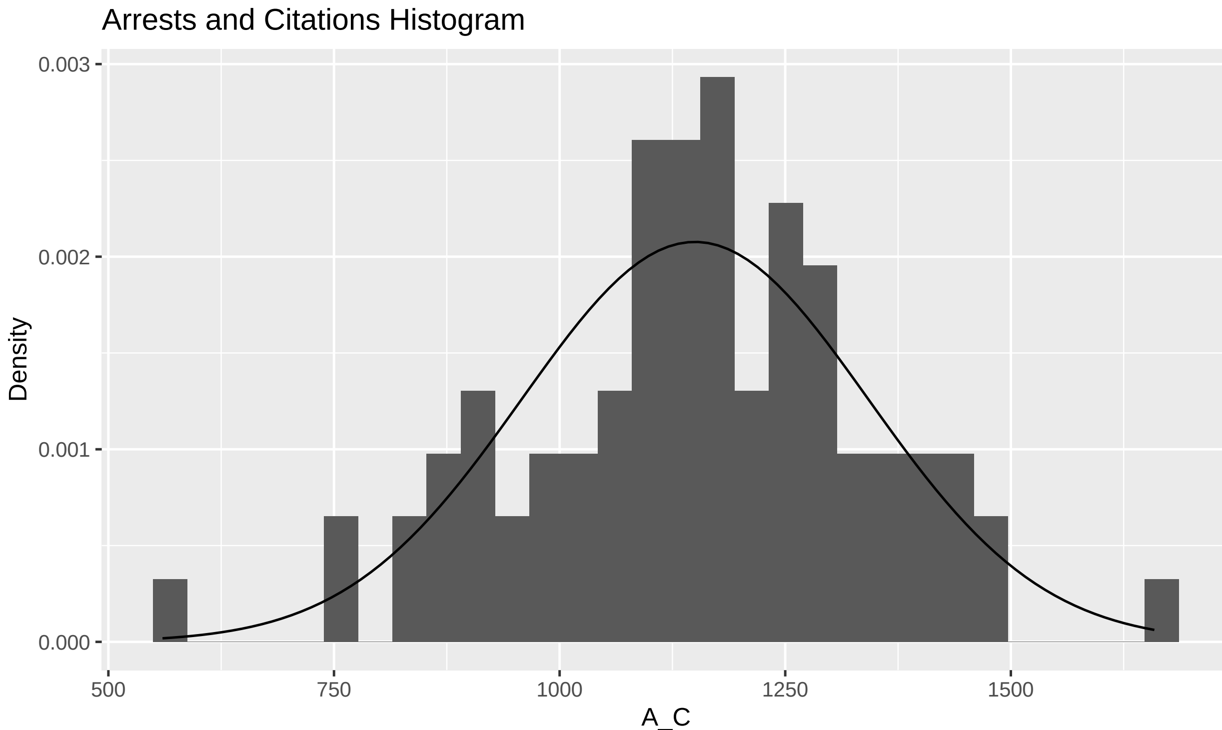


```r
# BAR CHARTS: just for fun, W/L by Location faceted by Opponent
ggplot(Husker_games, aes(Location, fill = Win)) +
  geom_histogram(position = "dodge", stat = "count") + facet_wrap(~ Opponent) +
  labs(title = "Game Outcome by Opponent and Location",
       x = "Location", y = "Game Count")
```

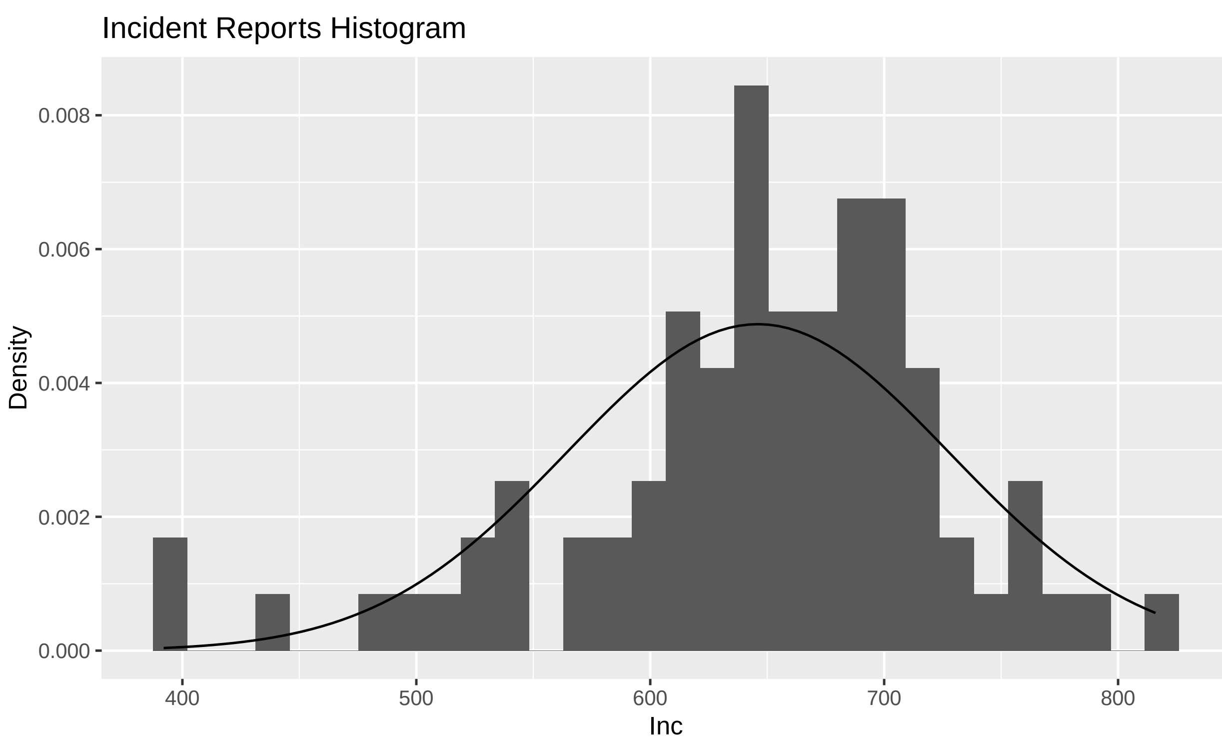


These bar charts provide another way to compare number of Public Safety incidents by type and both Location and Outcome. I have added some error bars and again Arrests and Citations seems to be determined by Location more than the other variables. The last bar chart is just for fun, showing how the Huskers fared against their Opponents in both Home and Away conditions.

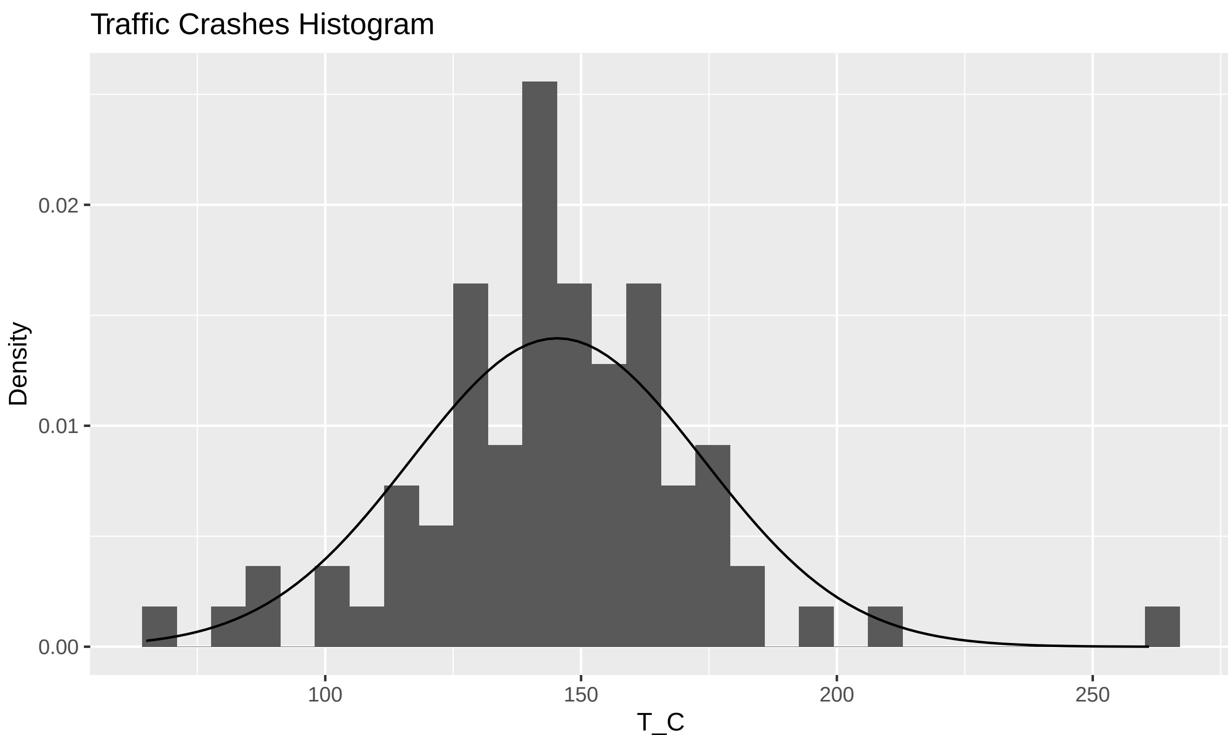
```r
# HISTOGRAMS: for each variable, with normality curve
ggplot(Husker_games, aes(A_C)) + geom_histogram(aes(y = ..density..)) +
  stat_function(fun = dnorm, args = list(mean = mean(Husker_games$A_C),
                                         sd(Husker_games$A_C))) +
  labs(title = "Arrests and Citations Histogram", y = "Density")
```



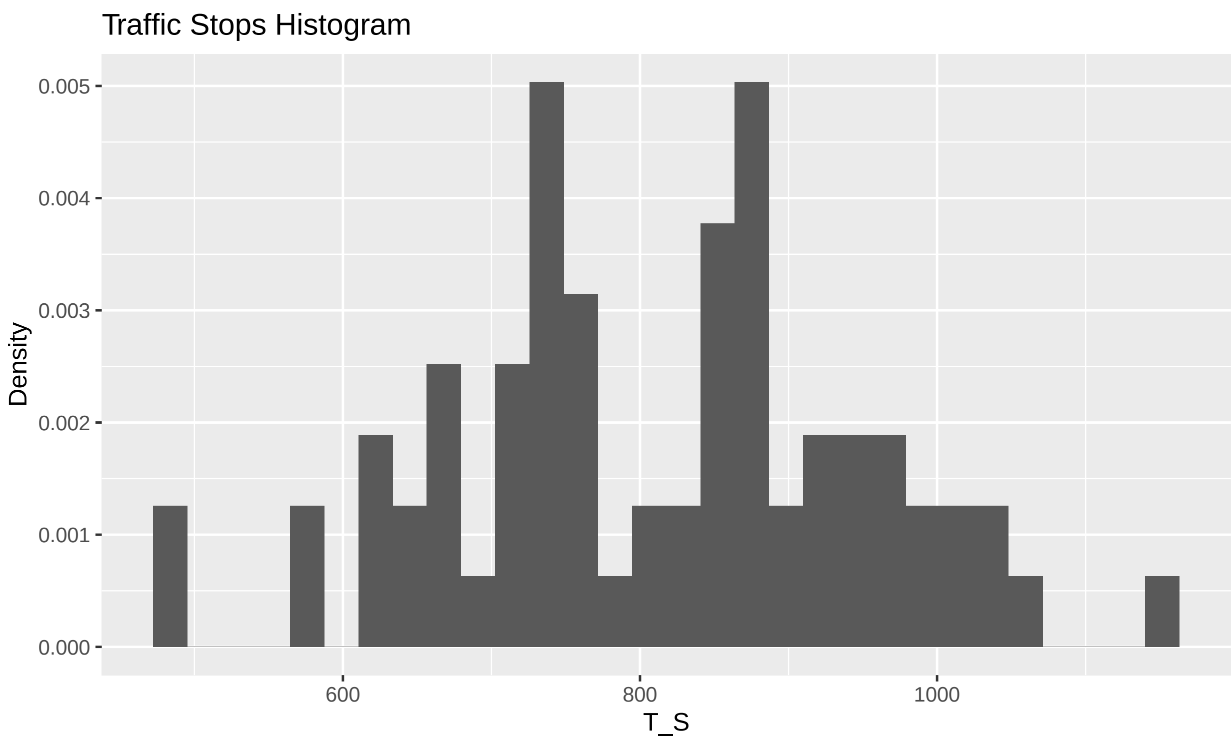

```
ggplot(Husker_games, aes(Inc)) + geom_histogram(aes(y = ..density..)) +
  stat_function(fun = dnorm, args = list(mean = mean(Husker_games$Inc),
                                          sd(Husker_games$Inc))) +
  labs(title = "Incident Reports Histogram", y = "Density")
```



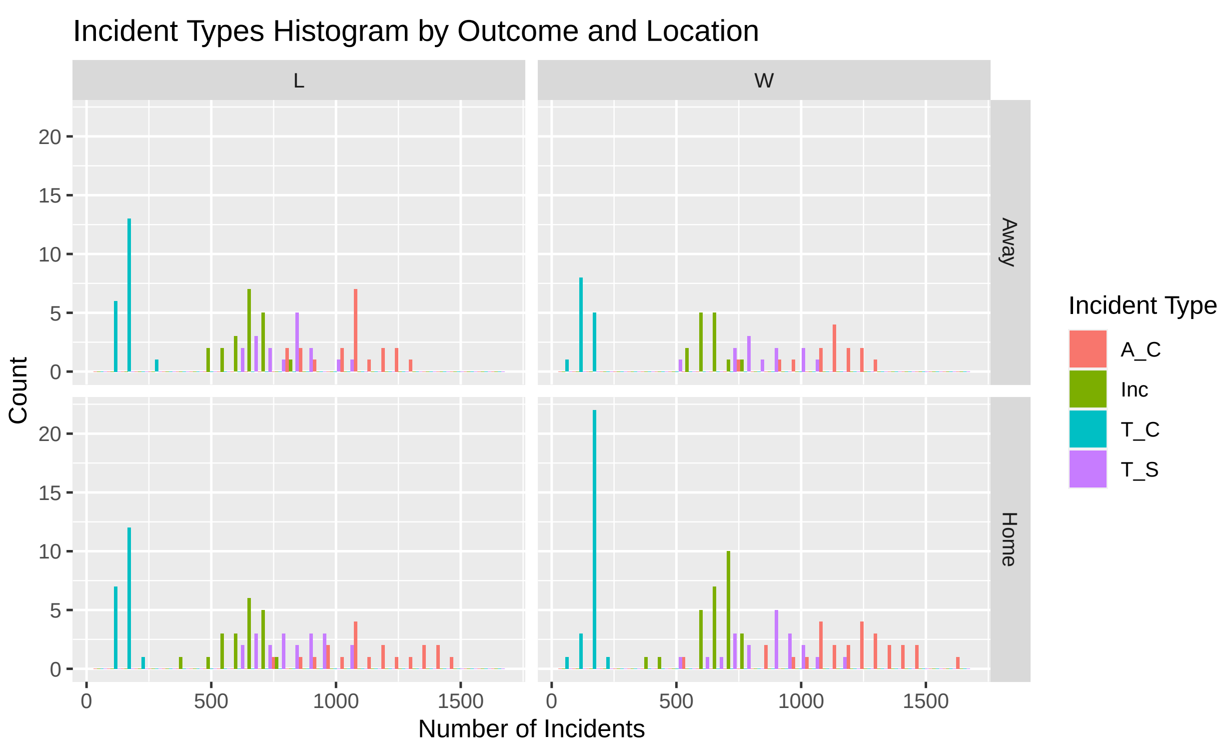
```
ggplot(Husker_games, aes(T_C)) + geom_histogram(aes(y = ..density..)) +
  stat_function(fun = dnorm, args = list(mean = mean(Husker_games$T_C),
                                         sd(Husker_games$T_C))) +
  labs(title = "Traffic Crashes Histogram", y = "Density")
```




```
ggplot(Husker_games, aes(T_S)) + geom_histogram(aes(y = ..density..)) +
  stat_function(fun = dnorm, args = list(mean = mean(Husker_games$T_S),
                                         sd(Husker_games$T_S))) +
  labs(title = "Traffic Stops Histogram", y = "Density")
```




```r
# HISTOGRAMS: colored by variables, faceted by W/L & Home/Away
ggplot (H_G.long, aes(value, fill = variable)) +
  geom_histogram(position = "dodge") + facet_grid(Location ~ Win) +
  labs(title = "Incident Types Histogram by Outcome and Location",
       x = "Number of Incidents",y = "Count", fill = "Incident Type")
```



Overall, the distributions of each Public Safety Incident Type appear fairly normal, which they should, considering each sample set is over 10,000 observations, however the Incident Reports Histogram does look slightly skewed and the Traffic Stops Histogram looks like it might be bimodal. The last histogram shows that the histograms for each Incident Type look pretty similar regardless of Location or Outcome.

```
## Check for normal distribution
shapiro.test(Husker_games$A_C) # 0.8052
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Husker_games$A_C
## W = 0.99026, p-value = 0.8052
```

```
shapiro.test(Husker_games$Inc) # 0.001708 <- not normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Husker_games$Inc
## W = 0.94519, p-value = 0.001708
```

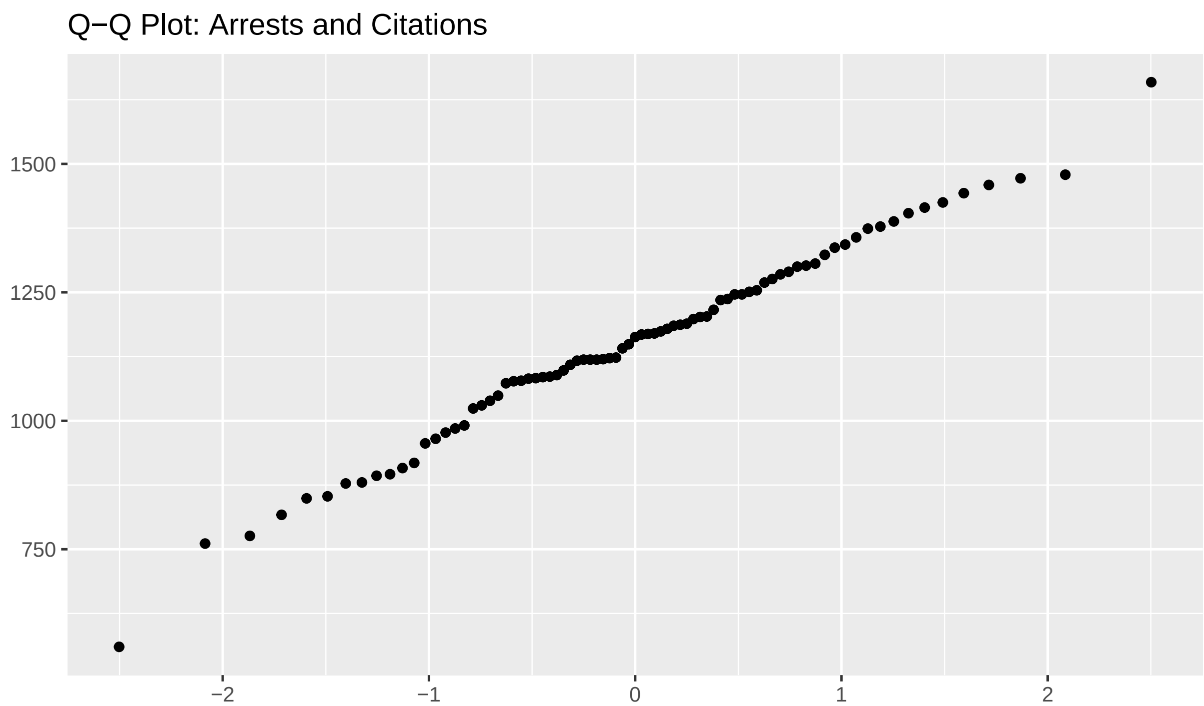
```
shapiro.test(Husker_games$T_C) # 0.005984 <- not normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Husker_games$T_C
## W = 0.95463, p-value = 0.005984
```

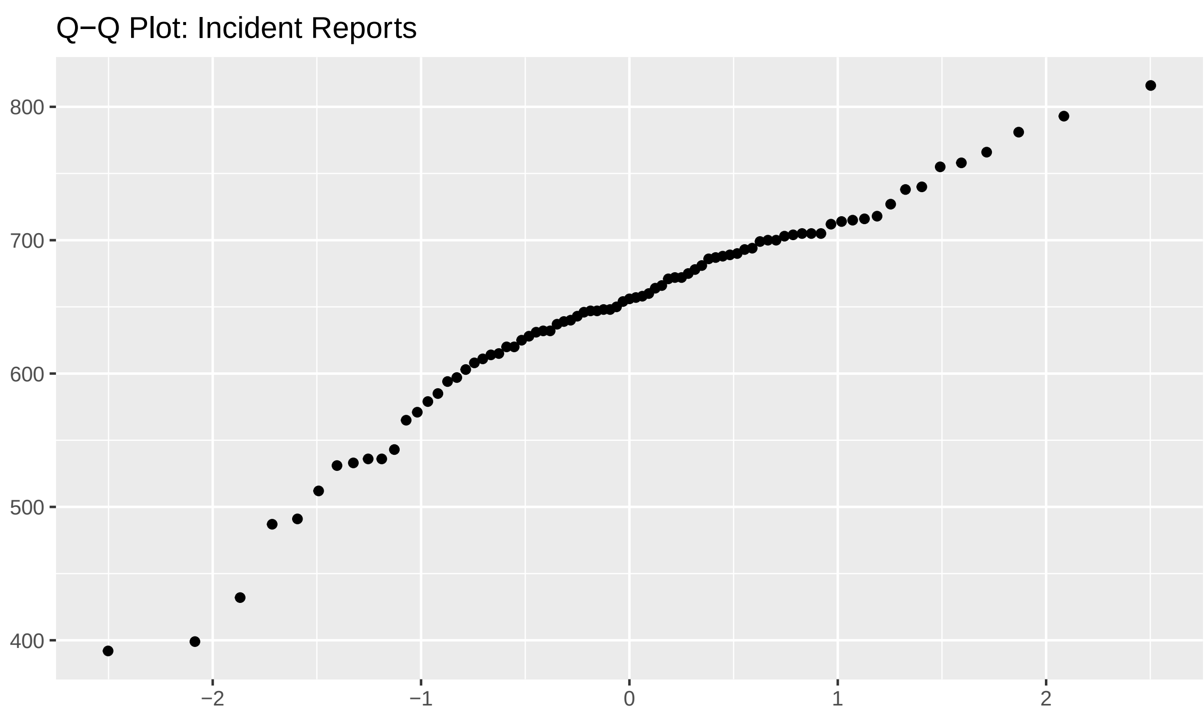
```
shapiro.test(Husker_games$T_S) # 0.8249
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Husker_games$T_S
## W = 0.9908, p-value = 0.8965
```

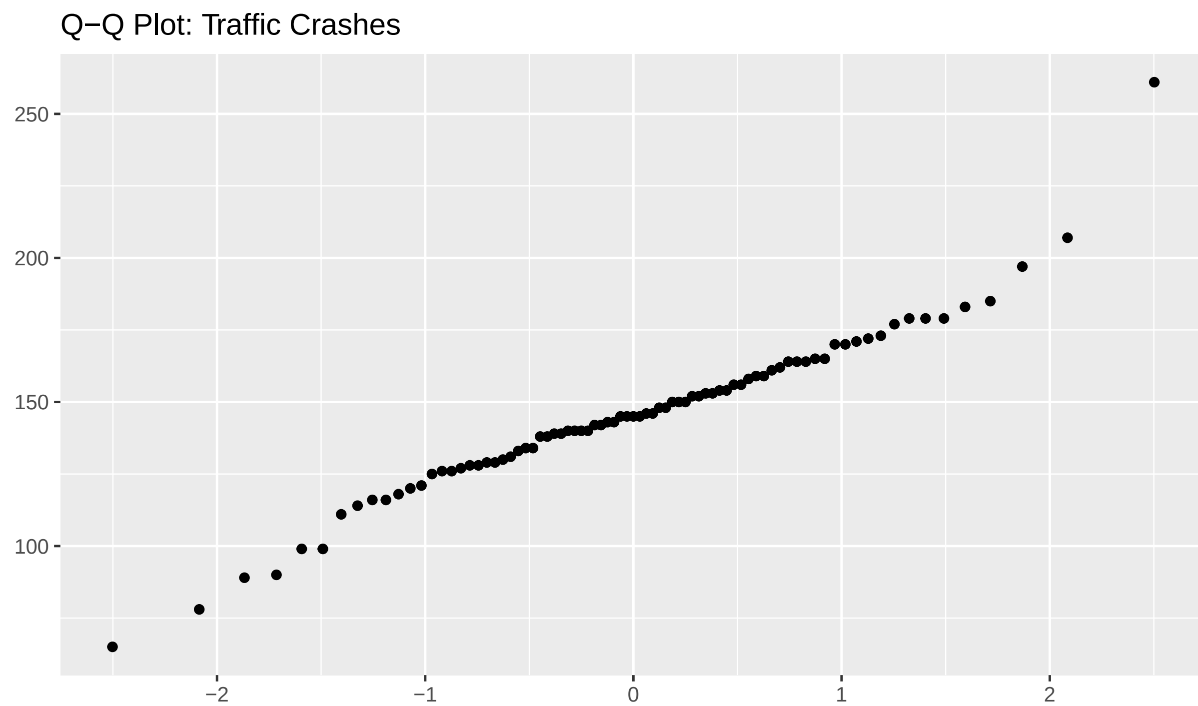
```
# Check for normalcy w/ qq plot
qplot(sample = Husker_games$A_C) + labs(title = "Q-Q Plot: Arrests and Citations")
```



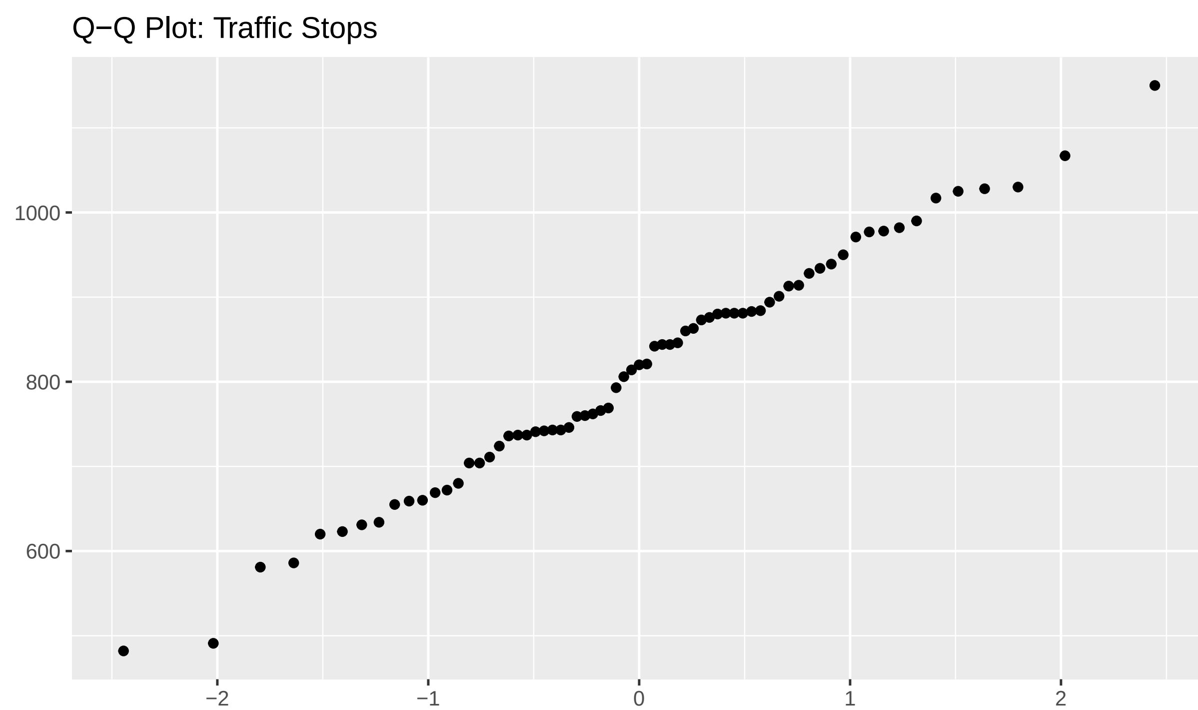

```
qplot(sample = Husker_games$Inc) + labs(title = "Q-Q Plot: Incident Reports")
```




```
qplot(sample = Husker_games$T_C) + labs(title = "Q-Q Plot: Traffic Crashes")
```




```
qplot(sample = Husker_games$T_S) + labs(title = "Q-Q Plot: Traffic Stops")
```



The Shapiro-Wilk normality tests show that the Arrests and Citations and the Traffic Stops Public Safety Incident Types have a fairly normal distribution, while the Incident Reports and the Traffic Crashes have a distribution that is not normally distributed. This is also reflected by the Q-Q plots for each.

```r
# Check (point biserial) correlations
cor.test(H_G.coded$A_C, H_G.coded$Location) # p 0.03558
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$A_C and H_G.coded$Location
## t = 2.1383, df = 79, p-value = 0.03558
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01639473 0.43028032
## sample estimates:
##       cor
## 0.2339068
```

```r
cor.test(H_G.coded$Inc, H_G.coded$Location) # p 0.5723
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$Inc and H_G.coded$Location
## t = 0.56702, df = 79, p-value = 0.5723
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1568651  0.2781477
## sample estimates:
##        cor
## 0.06366492
```

```r
cor.test(H_G.coded$T_C, H_G.coded$Location) # p 0.5048
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$T_C and H_G.coded$Location
## t = 0.66996, df = 79, p-value = 0.5048
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1455757  0.2887730
## sample estimates:
##        cor
## 0.07516284
```

```r
cor.test(H_G.coded$T_S, H_G.coded$Location) # p 0.4838
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$T_S and H_G.coded$Location
## t = 0.70411, df = 67, p-value = 0.4838
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.1541024  0.3159755
## sample estimates:
##        cor
## 0.08570422
```

```
cor.test(H_G.coded$Tot_Inc, H_G.coded$Location) # p 0.09613
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$Tot_Inc and H_G.coded$Location
## t = 1.5888, df = 67, p-value = 0.1168
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0483130  0.4087913
## sample estimates:
##       cor
## 0.1905465
```

```
cor.test(H_G.coded$A_C, H_G.coded$Win) # p 0.1085
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$A_C and H_G.coded$Win
## t = 1.6234, df = 79, p-value = 0.1085
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04025974  0.38299341
## sample estimates:
##       cor
## 0.1796691
```

```
cor.test(H_G.coded$Inc, H_G.coded$Win) # p 0.3060
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$Inc and H_G.coded$Win
## t = 1.0303, df = 79, p-value = 0.306
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1058626  0.3253185
## sample estimates:
##       cor
## 0.1151485
```

```
cor.test(H_G.coded$T_C, H_G.coded$Win) # p 0.4557
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  H_G.coded$T_C and H_G.coded$Win
## t = -0.74969, df = 79, p-value = 0.4557
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2969479  0.1368121
## sample estimates:
##         cor
## -0.08404806
```

```
cor.test(H_G.coded$T_S, H_G.coded$Win) # p 0.2604
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$T_S and H_G.coded$Win
## t = 1.0932, df = 67, p-value = 0.2782
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1076699  0.3578526
## sample estimates:
##       cor
## 0.1323844
```

```
cor.test(H_G.coded$Tot_Inc, H_G.coded$Win) # p 0.1227
```

```
##
##  Pearson's product-moment correlation
##
## data:  H_G.coded$Tot_Inc and H_G.coded$Win
## t = 1.4355, df = 67, p-value = 0.1558
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0666701  0.3933361
## sample estimates:
##       cor
## 0.1727363
```

```
cor(H_G.coded[c("Location", "Win", "A_C", "Inc", "T_C", "T_S", "Tot_Inc")])
```

```
##            Location         Win        A_C        Inc         T_C T_S Tot_Inc
## Location 1.00000000  0.16060647 0.23390680 0.06366492  0.07516284  NA      NA
## Win      0.16060647  1.00000000 0.17966908 0.11514848 -0.08404806  NA      NA
## A_C      0.23390680  0.17966908 1.00000000 0.34380817  0.08629274  NA      NA
## Inc      0.06366492  0.11514848 0.34380817 1.00000000  0.35446651  NA      NA
## T_C      0.07516284 -0.08404806 0.08629274 0.35446651  1.00000000  NA      NA
## T_S              NA          NA         NA         NA          NA   1      NA
## Tot_Inc          NA          NA         NA         NA          NA  NA       1
```

```r
cor(H_G.coded[c("Location", "Win", "A_C", "Inc", "T_C", "T_S", "Tot_Inc")])^2 * 100
```

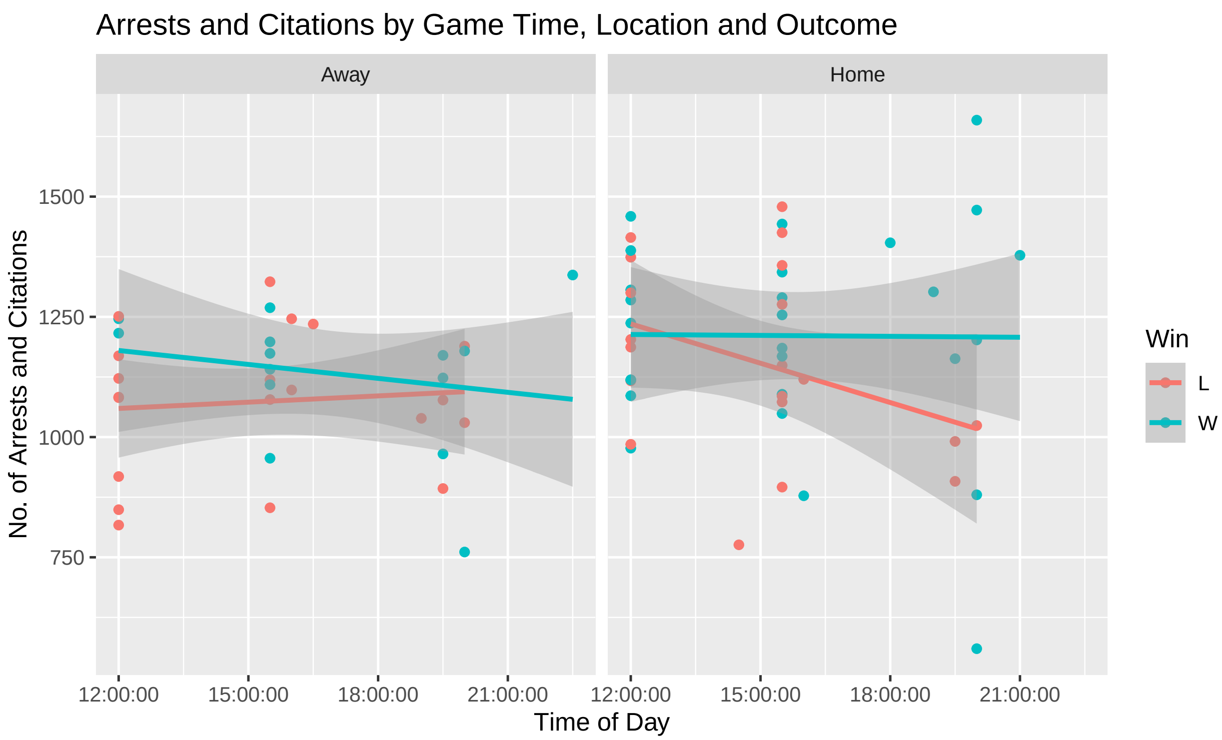
```
##             Location         Win         A_C         Inc         T_C T_S
## Location 100.0000000   2.5794438   5.4712391   0.4053222   0.5649453  NA
## Win        2.5794438 100.0000000   3.2280977   1.3259173   0.7064077  NA
## A_C        5.4712391   3.2280977 100.0000000  11.8204055   0.7446437  NA
## Inc        0.4053222   1.3259173  11.8204055 100.0000000  12.5646509  NA
## T_C        0.5649453   0.7064077   0.7446437  12.5646509 100.0000000  NA
## T_S               NA          NA          NA          NA          NA 100
## Tot_Inc           NA          NA          NA          NA          NA  NA
##          Tot_Inc
## Location      NA
## Win           NA
## A_C           NA
## Inc           NA
## T_C           NA
## T_S           NA
## Tot_Inc      100
```

Checking correlations between each Public Safety Incident Type with both Location and game Outcome, I found that Arrests and Citations was the only Incident Type that had a significant correlation and that was with Location. Arrests and Citations with game Outcome and Total number of Incidents with both Location and Outcome all had a significance around 0.1, but I suspect the correlation of Total Incidents is largely an artifact of the Arrests and Citations data set, which accounts for around 40% of the total data.

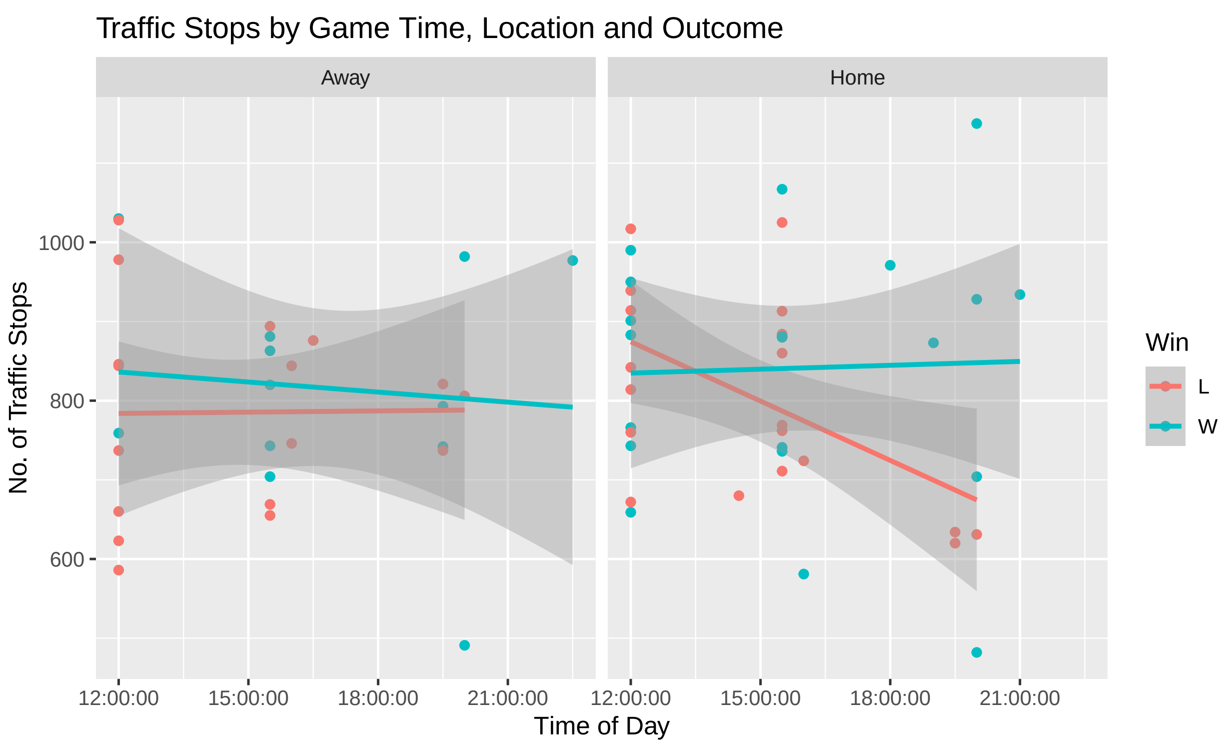
Looking at the R-squared values, we see that the some of the Public Safety Incident Types correlate pretty well with each other. Ignoring Total Incidents for a moment, we see that Arrests and Citations correlates really quite well with Traffic Stops, sharing 53% of variability. Traffic Stops and Incident Reports are the next highest, with 13%. However, we are more interested in how these Incident Types correlate with game Location (Home or Away) and game Outcome (Win or Lose) and unfortunately, while Arrests and Citations comes in at the highest for both, it only accounts for 5.5% and 3.2% of variability with Location and Outcome respectively and as we saw, only the correlation with Location is significant.

Interestingly, Location and Outcome share 2.6% of their variability, so it appears there may be some truth to Home team advantage.

```r
# SCATTERPLOTS: time of day vs A_C & T_S (two normal variables), Colored by Win, Faceted by Location
ggplot(Husker_games, aes(Time, A_C, color = Win)) + geom_point() +
  geom_smooth(method = lm) + facet_wrap(~ Location) +
  labs(title = "Arrests and Citations by Game Time, Location and Outcome",
       x = "Time of Day", y = "No. of Arrests and Citations")
```




```
ggplot(Husker_games, aes(Time, T_S, color = Win)) + geom_point() +
  geom_smooth(method = lm) + facet_wrap(~ Location) +
  labs(title = "Traffic Stops by Game Time, Location and Outcome",
       x = "Time of Day", y = "No. of Traffic Stops")
```



Looking at the two variables that had normal distribution curves, I did notice something interesting and that's that, while the number of incidents for Away games remain fairly constant regardless of game time and also for Home games, if the Huskers Win, but if the Huskers lose, the number of Incidents starts higher in the morning and drops dramatically throughout the day for later games. I suspect it may have something to do with people being tired and ready to go home after later games when the Huskers lose, compared to being in a celebratory mood and staying out if they win.

```
H_G.lm <- lm(A_C ~ Opponent + Location, data = H_G.coded)
summary(H_G.lm)
```

```
##
## Call:
## lm(formula = A_C ~ Opponent + Location, data = H_G.coded)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -373.30  -89.02    0.00  104.19  321.81
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    486.63     180.44   2.697 0.009413 **
## OpponentBethune-Cookman        559.00     247.33   2.260 0.028026 *
## OpponentBYU                    716.00     247.33   2.895 0.005532 **
## OpponentColorado               615.69     215.34   2.859 0.006100 **
## OpponentFresno State           746.19     215.34   3.465 0.001070 **
## OpponentIllinois               615.02     187.93   3.273 0.001897 **
## OpponentIndiana                479.19     215.34   2.225 0.030427 *
## OpponentIowa                   589.30     187.93   3.136 0.002819 **
## OpponentMaryland               514.19     215.34   2.388 0.020619 *
## OpponentMcNeese State          725.00     247.33   2.931 0.005006 **
## OpponentMiami (FL)             967.69     215.34   4.494 3.93e-05 ***
## OpponentMichigan               542.87     218.75   2.482 0.016346 *
## OpponentMichigan State         664.59     195.84   3.393 0.001327 **
## OpponentMinnesota              567.07     188.68   3.005 0.004075 **
## OpponentNorthern Illinois      473.50     214.19   2.211 0.031484 *
## OpponentNorthwestern           597.87     187.93   3.181 0.002473 **
## OpponentOhio State             501.19     196.79   2.547 0.013869 *
## OpponentOregon                 686.69     215.34   3.189 0.002420 **
## OpponentPenn State             696.87     218.75   3.186 0.002442 **
## OpponentPurdue                 603.93     188.68   3.201 0.002338 **
## OpponentRutgers                731.46     202.48   3.612 0.000683 ***
## OpponentSouth Alabama          912.00     247.33   3.687 0.000542 ***
## OpponentSouth Dakota State     730.00     247.33   2.952 0.004734 **
## OpponentSouthern Mississippi   760.50     214.19   3.551 0.000826 ***
## OpponentTroy                   740.00     247.33   2.992 0.004231 **
## OpponentUCLA                   855.00     247.33   3.457 0.001097 **
## OpponentWisconsin              597.19     190.20   3.140 0.002787 **
## OpponentWyoming                828.00     247.33   3.348 0.001520 **
## Location                        73.37      44.44   1.651 0.104746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 174.9 on 52 degrees of freedom
## Multiple R-squared:  0.4612, Adjusted R-squared:  0.1711
```

```
## F-statistic:  1.59 on 28 and 52 DF,  p-value: 0.07361
```

```r
round(tapply(H_G.coded$A_C, H_G.coded[c("Opponent", "Location")], mean, na.rm = TRUE), 2)
```

```
##                        Location
## Opponent                      0       1
##   Arkansas State             NA  560.00
##   Bethune-Cookman            NA 1119.00
##   BYU                        NA 1276.00
##   Colorado               853.00 1425.00
##   Fresno State          1337.00 1202.00
##   Illinois              1062.00 1204.75
##   Indiana               1109.00  896.00
##   Iowa                  1135.33 1104.75
##   Maryland               956.00 1119.00
##   McNeese State              NA 1285.00
##   Miami (FL)            1323.00 1659.00
##   Michigan              1029.50      NA
##   Michigan State        1189.00 1212.00
##   Minnesota             1015.50 1178.00
##   Northern Illinois          NA 1033.50
##   Northwestern          1125.33 1127.25
##   Ohio State            1099.50  949.50
##   Oregon                1235.00 1185.00
##   Penn State            1183.50      NA
##   Purdue                1062.25 1201.67
##   Rutgers               1174.00 1313.50
##   South Alabama              NA 1472.00
##   South Dakota State         NA 1290.00
##   Southern Mississippi       NA 1320.50
##   Troy                       NA 1300.00
##   UCLA                       NA 1415.00
##   Wisconsin             1078.33 1162.67
##   Wyoming                    NA 1388.00
```

Doing some modeling based on Opponents and Location may help to give an indication as to what to expect for different types of incidents. Here I have just used the Arrests and Citations data, but more in-depth analysis could be done for each Opposing team and each Incident Type.

My goal with this project was to determine what effect, if any, the influx of people with a Home Cornhusker football game would have on the overall crime rate in Lincoln and while I found there is a slight increase in Arrests and Citations, overall I found relatively little effect. All the same, the data could be useful (perhaps more so by including additional years of data) in predicting increases in Police coverage, especially when particular Opponents are accounted for.

The most uplifting thing I found was that crime appears to continue on a downward trend year-to-year and I hope that this trend continues for the foreseeable future. Also, Go Big Red!