Assignment 10.1

Scott Breitbach

8/3/2020

### Title: DSC520 Final Project: Effect of Home Cornhusker Games on Public Safety

## 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 your approach addresses (fully or partially) 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 of my datasets were obtained through the [Open Data and Performance Management](http://opendata.lincoln.ne.gov/) page for the city of Lincoln.

Files:

* [LPD\_2013\_2020\_Arrests\_and\_Citations\_De\_Coded.csv](https://opendata.arcgis.com/datasets/69363105cc3f4f73a3318cafed030dfa_0.csv)
* [LPD\_2017\_2020\_Incident\_Reports.csv](https://opendata.arcgis.com/datasets/dc814856aa6645879c3a0aa7e7d527e0_0.csv?outSR=%7B%22latestWkid%22%3A4326%2C%22wkid%22%3A4326%7D)
* [LPD\_Traffic\_Crashes\_2013\_2020.csv](https://opendata.arcgis.com/datasets/fd7f05be61da45b3b14be8780f7685b2_0.csv)

I may also look at Traffic Stops, 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.

Many of the variables are meaningless to me, but I will primarily be looking at total incidences and the dates and times on which they occurred. I will also 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:

I will need the following packages

* readr
* ggplot2
* lubridate
* dplyr
* hmisc
* car

I will probably add more as I need them.

### Plots and table needs:

I will need 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 probably have all of the tools I need, though I’m sure I will need a refresher on some of the steps along the way, especially when it comes to 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, so perhaps I will get a chance to do that within the scope of this project.

## Section 2: Cleaning Your Data and Exploratory Data Analysis

# Load game data  
Huskers\_2019 <- read\_csv("completed/FinalProject/data/Huskers\_2019.csv")  
Huskers\_2018 <- read\_csv("completed/FinalProject/data/Huskers\_2018.csv")  
Huskers\_2017 <- read\_csv("completed/FinalProject/data/Huskers\_2017.csv")  
Huskers\_2016 <- read\_csv("completed/FinalProject/data/Huskers\_2016.csv")  
Huskers\_2015 <- read\_csv("completed/FinalProject/data/Huskers\_2015.csv")  
Huskers\_2014 <- read\_csv("completed/FinalProject/data/Huskers\_2014.csv")  
Huskers\_2013 <- read\_csv("completed/FinalProject/data/Huskers\_2013.csv")

# 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  
rm(Huskers\_2013, Huskers\_2014, Huskers\_2015, Huskers\_2016, Huskers\_2017,   
 Huskers\_2018, Huskers\_2019)

## Arrests and Citations  
arr\_cit\_13 <- read\_csv("completed/FinalProject/data/LPD\_arrests\_and\_citations\_2013.csv")  
arr\_cit\_14 <- read\_csv("completed/FinalProject/data/LPD\_arrests\_and\_citations\_2014.csv")  
arr\_cit\_15 <- read\_csv("completed/FinalProject/data/LPD\_arrests\_and\_citations\_2015.csv")  
arr\_cit\_16 <- read\_csv("completed/FinalProject/data/LPD\_arrests\_and\_citations\_2016.csv")  
arr\_cit\_17 <- read\_csv("completed/FinalProject/data/LPD\_arrests\_and\_citations\_2017.csv")  
arr\_cit\_18 <- read\_csv("completed/FinalProject/data/LPD\_arrests\_and\_citations\_2018.csv")  
arr\_cit\_19 <- read\_csv("completed/FinalProject/data/LPD\_arrestd\_and\_citations\_2019.csv")  
  
## Incident Reports  
incidents\_2017\_2020 <- read\_csv("completed/FinalProject/data/LPD\_2017\_2020\_Incident\_Reports.csv")  
incidents\_2016 <- read\_csv("completed/FinalProject/data/LPD\_Incident\_Reports\_2016.csv")  
incidents\_2015 <- read\_csv("completed/FinalProject/data/LPD\_Incident\_Reports\_2015.csv")  
incidents\_2014 <- read\_csv("completed/FinalProject/data/LPD\_Incident\_Reports\_2014.csv")  
incidents\_2013 <- read\_csv("completed/FinalProject/data/LPD\_Incident\_Reports\_2013.csv")  
  
## Traffic Crashes  
Trf\_Crash\_13 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2013.csv")  
Trf\_Crash\_14 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2014.csv")  
Trf\_Crash\_15 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2015.csv")  
Trf\_Crash\_16 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2016.csv")  
Trf\_Crash\_17 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2017.csv")  
Trf\_Crash\_18 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2018.csv")  
Trf\_Crash\_19 <- read\_csv("completed/FinalProject/data/Traffic\_Crashes\_2019.csv")  
  
## Traffic Stops  
Trf\_Stop\_13 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2013.csv")  
Trf\_Stop\_14 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2014.csv")  
Trf\_Stop\_15 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2015.csv")  
Trf\_Stop\_16 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2016.csv")  
Trf\_Stop\_17 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2017.csv")  
Trf\_Stop\_18 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2018.csv")  
Trf\_Stop\_19 <- read\_csv("completed/FinalProject/data/LPD\_traffic\_stops\_2019.csv")

## 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  
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)  
# 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)  
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)

# 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)

"Game Data"

## [1] "Game Data"

head(Husker\_games)

## # A tibble: 6 x 15  
## Date Time Year Week Day Location School Opponent Win Yr\_Wk A\_C  
## <date> <tim> <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>

"Arrests and Citations"

## [1] "Arrests and Citations"

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

"Incident Reports"

## [1] "Incident Reports"

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

"Traffic Crashes"

## [1] "Traffic Crashes"

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 INJU~ REAR ~ NO NO YES NO NO NO NO 2013-11-16  
## 2 INJU~ DRIVE~ NO NO YES NO NO YES NO 2013-11-29  
## 3 INJU~ OTHER NO YES YES NO NO NO NO 2013-10-10  
## 4 INJU~ RIGHT~ NO YES NO NO NO NO NO 2013-09-02  
## 5 INJU~ OTHER YES NO YES NO NO NO NO 2013-10-01  
## 6 INJU~ DRIVE~ NO YES NO NO NO NO NO 2013-09-02  
## # ... with 3 more variables: Time <time>, Day <ord>, Yr\_Wk <chr>

"Traffic Stops"

## [1] "Traffic Stops"

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 Submission

**NOTE: the variables for the Public Safety data are defined as follows:**

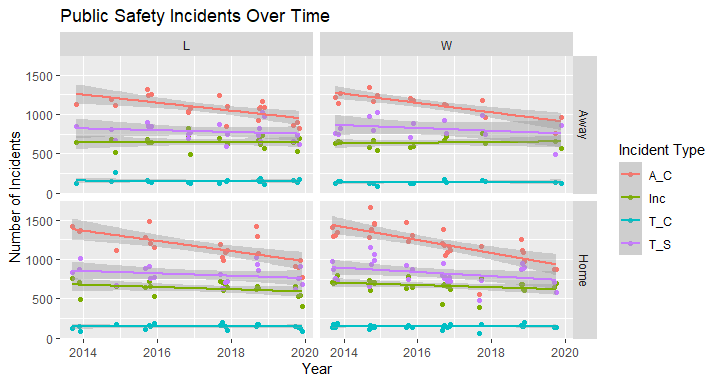
* **A\_C: Arrests and Citations**
* **Inc: Incident Reports completed**
* **T\_C: Traffic Crashes reported**
* **T\_S: Traffic (and Pedestrian) Stop records**

# 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

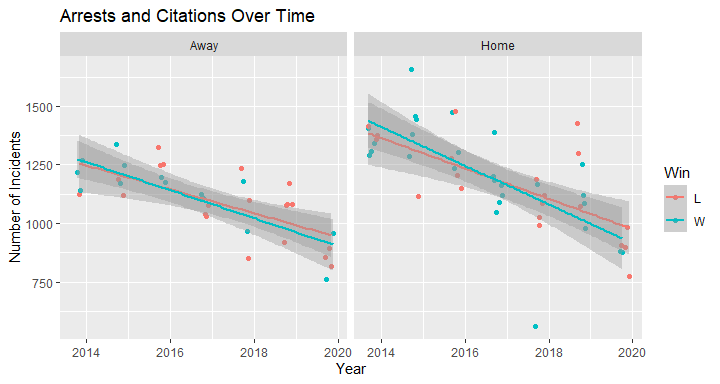
Once I had my data cleaned, I still needed to work with some of the data in different ways. To that end, I made a couple of other data frames. In one, H\_G.long, I took the Public Safety totals that I had added to my Husker Games data set from each category and “melted” them into just two columns, represented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

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

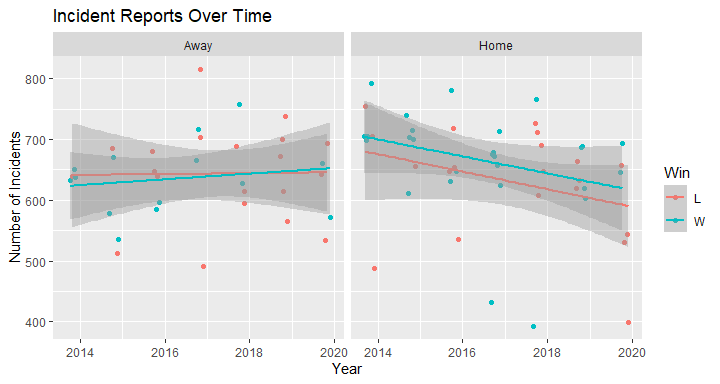
# SCATTERPLOT: Shows change over time for variables, faceted by W/L & Home/Away; shows all steady or decreasing over time  
ggplot(H\_G.long, aes(x = Date, y = value, color = variable)) + geom\_point() +   
 geom\_smooth(method = lm) + facet\_grid(Location ~ Win) +   
 labs(title = "Public Safety Incidents Over Time", x = "Year",   
 y = "Number of Incidents", color = "Incident Type")



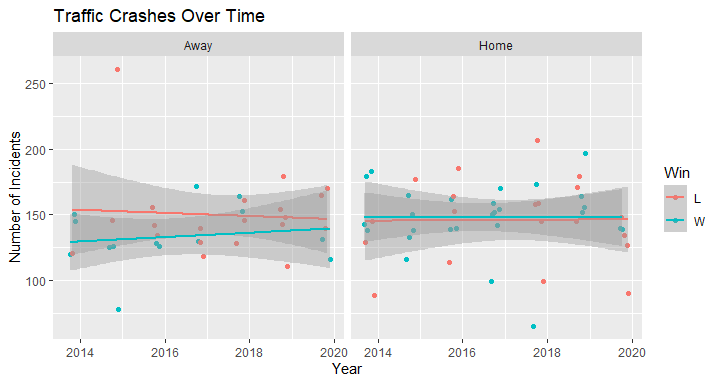
# SCATTERPLOT: Compares Win/Loss over time for each variable, faceted by Home/Away; for the most part Win/Losslines cross or converge, indicating not much effect there, but Home is generally higher than Away  
ggplot(Husker\_games, aes(x = Date, y = A\_C, color = Win)) + geom\_point() +   
 geom\_smooth(method = lm) + facet\_wrap(~ Location) +   
 labs(title = "Arrests and Citations Over Time", x = "Year",   
 y = "Number of Incidents")



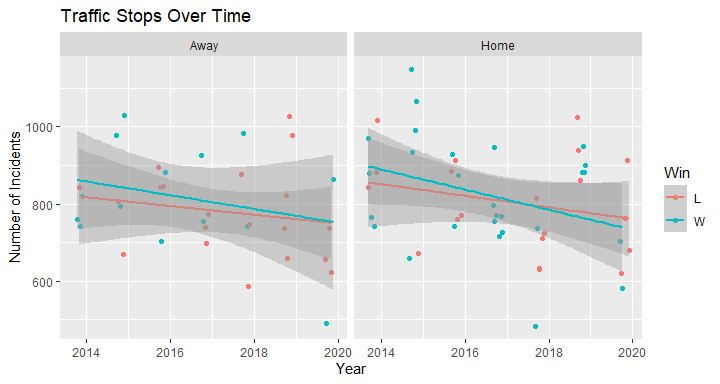
ggplot(Husker\_games, aes(x = Date, y = Inc, color = Win)) + geom\_point() +   
 geom\_smooth(method = lm) + facet\_wrap(~ Location) +   
 labs(title = "Incident Reports Over Time", x = "Year",   
 y = "Number of Incidents")



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")



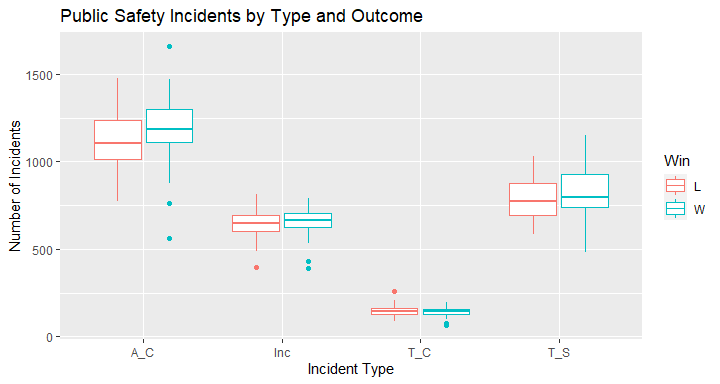
ggplot(Husker\_games, aes(x = Date, y = T\_S, color = Win)) + geom\_point() +   
 geom\_smooth(method = lm) + facet\_wrap(~ Location) +   
 labs(title = "Traffic Stops Over Time", x = "Year",   
 y = "Number of Incidents")



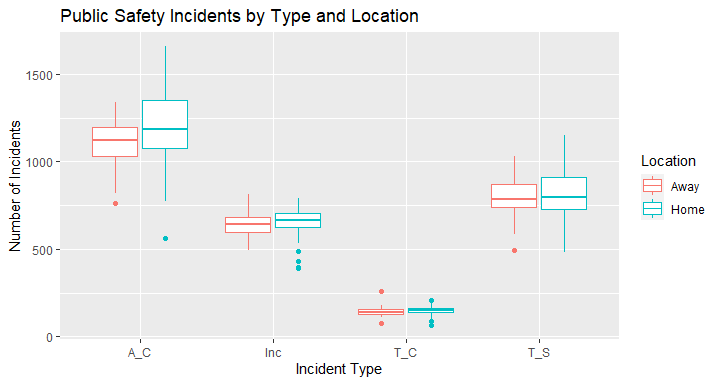
The first chart, with all four variables shows that for both Wins and Losses, Home and Away games, that Public Safety incidents are either generally remaining steady or decreasing over time, which is good news!

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.

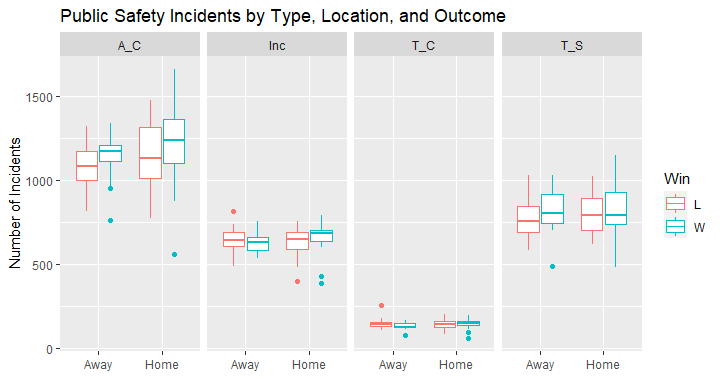
# BOXPLOTS: for all variables, colored by Win or Location  
ggplot(H\_G.long, aes(variable, value, color = Win)) + geom\_boxplot() +   
 labs(title = "Public Safety Incidents by Type and Outcome",   
 x = "Incident Type", y = "Number of Incidents")



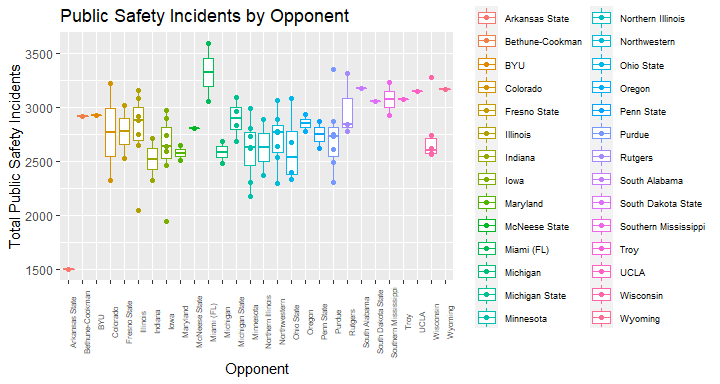
ggplot(H\_G.long, aes(variable, value, color = Location)) + geom\_boxplot() +   
 labs(title = "Public Safety Incidents by Type and Location",   
 x = "Incident Type", y = "Number of Incidents")



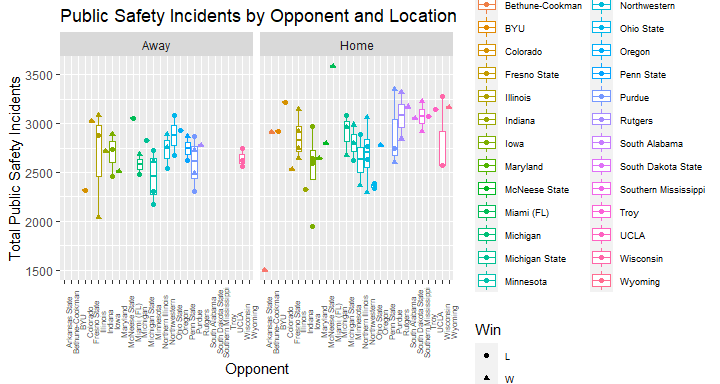
# BOXPLOT: for each location, colored by Win, faceted by variable (may not need previous two)  
ggplot(H\_G.long, aes(x = Location, color = Win)) +   
 geom\_boxplot(aes(y = value)) + facet\_grid(~ variable) +   
 labs(title = "Public Safety Incidents by Type, Location, and Outcome",   
 x = element\_blank(), y = "Number of Incidents")



# 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")



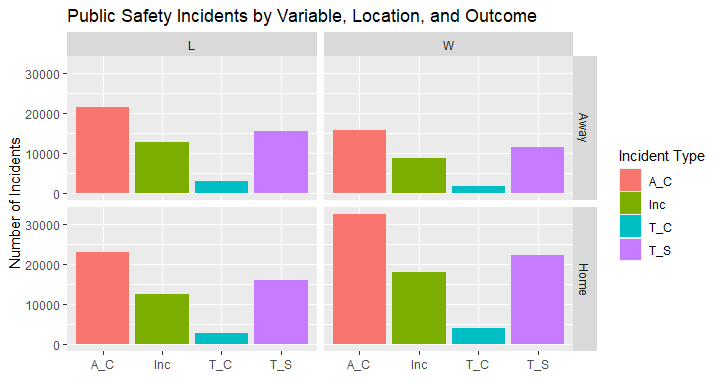
# BOXPLOT: shows Total Incidents, Home and Away, colored by Opponent, with points  
ggplot(Husker\_games, aes(Opponent, Tot\_Inc, color = Opponent)) +   
 geom\_boxplot() + geom\_point(aes(shape = Win)) +   
 theme(axis.text.x = element\_text(angle = 90, size = 6),   
 legend.text = element\_text(size = 7)) + facet\_wrap(~ Location) +   
 labs(title = "Public Safety Incidents by Opponent and Location",   
 x = "Opponent", y = "Total Public Safety Incidents")



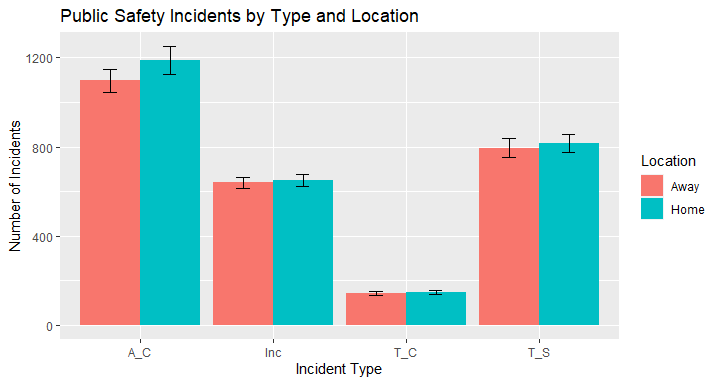
Looking at the boxplots, you can see that for most of the Public Safety variables it doesn’t appear to make much difference whether it is a Home or Away game, or whether it is a Win or a Loss. The exception to this appears to be Arrests and Citations, which is slightly higher for both Wins and Home games.

Looking at the different Opponents, it appears that Miami is the clear leader when it comes to total number of Public Safety Incidents. This appears 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 ponts, but perhaps this could be something to look into further.

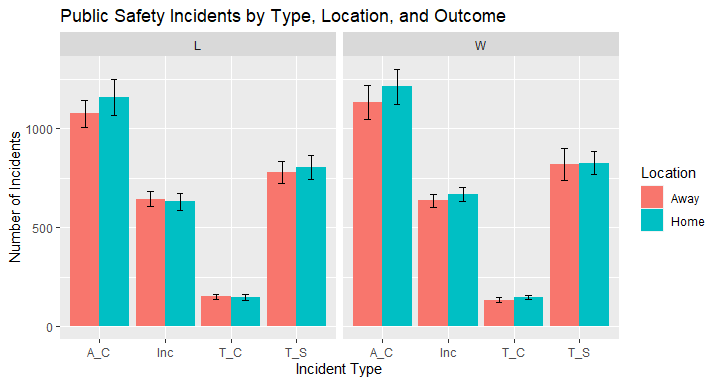
# BAR CHART: Comparing variables by w/L & Home/Away; not much difference in Home/Away when losing  
ggplot(H\_G.long, aes(variable, value, fill = variable)) +   
 geom\_bar(stat = "identity") + facet\_grid(Location ~ Win) +   
 labs(title = "Public Safety Incidents by Variable, Location, and Outcome",   
 x = element\_blank(), y = "Number of Incidents", fill = "Incident Type")



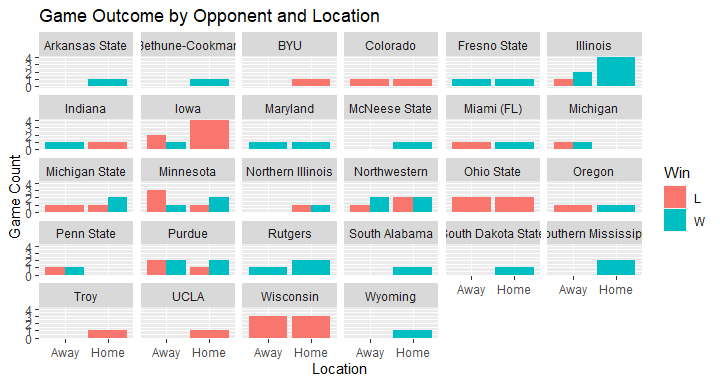
# BAR CHARTS:  
ggplot(H\_G.long, aes(variable, value, fill = Location)) +   
 stat\_summary(fun = mean, geom = "bar", position = "dodge") +   
 stat\_summary(fun.data = mean\_cl\_normal, geom = "errorbar",   
 position = position\_dodge(width = 0.90), width = 0.2) +   
 labs(title = "Public Safety Incidents 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 = "Number of Incidents")

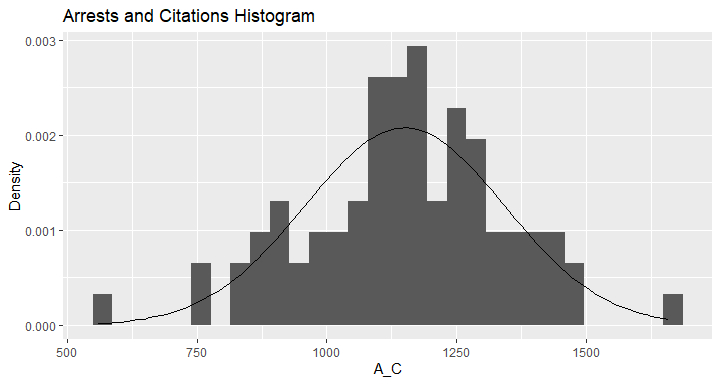


# BAR CHARTS: just for fun, W/L by Location faceted by Opponent  
ggplot(Husker\_games, aes(Location, fill = Win)) +   
 geom\_histogram(position = "dodge", stat = "count") + facet\_wrap(~ Opponent) +   
 labs(title = "Game Outcome by Opponent and Location",   
 x = "Location", y = "Game Count")

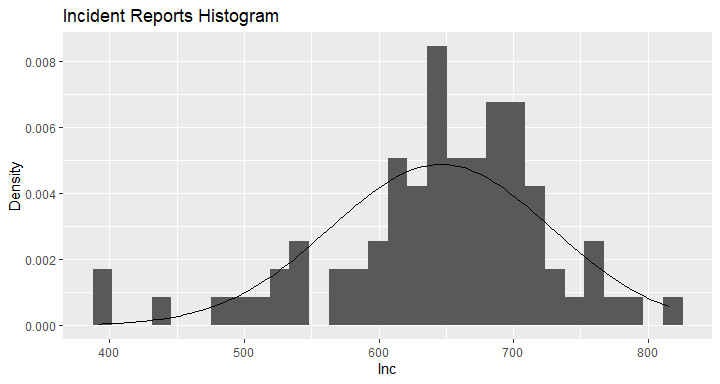


These bar charts provide another way to compare number of Public Safety incidents by type and both Location and Outcome. I have added some error bars and again Arrests and Citations seems to be determined by Location more than the other variables. The last bar chart is just for fun, showing how the Huskers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

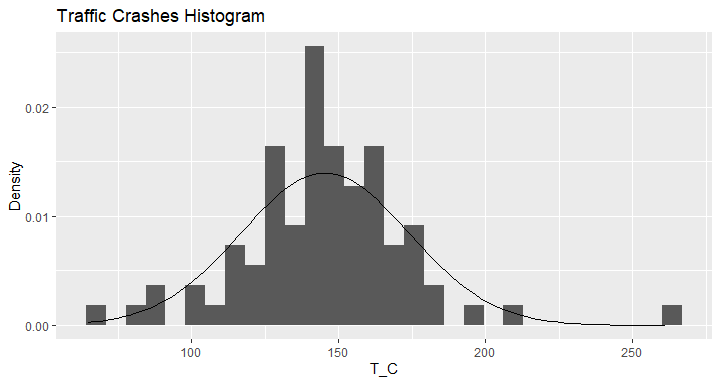
# HISTOGRAMS: for each variable, with normality curve  
ggplot(Husker\_games, aes(A\_C)) + geom\_histogram(aes(y = ..density..)) +  
 stat\_function(fun = dnorm, args = list(mean = mean(Husker\_games$A\_C),   
 sd(Husker\_games$A\_C))) +   
 labs(title = "Arrests and Citations Histogram", y = "Density")



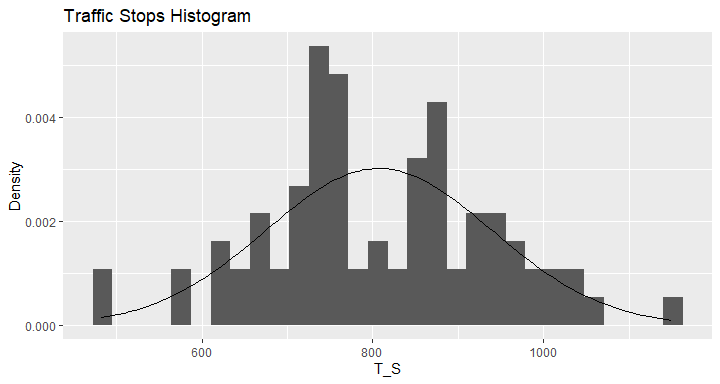
ggplot(Husker\_games, aes(Inc)) + geom\_histogram(aes(y = ..density..)) +  
 stat\_function(fun = dnorm, args = list(mean = mean(Husker\_games$Inc),   
 sd(Husker\_games$Inc))) +   
 labs(title = "Incident Reports Histogram", y = "Density")



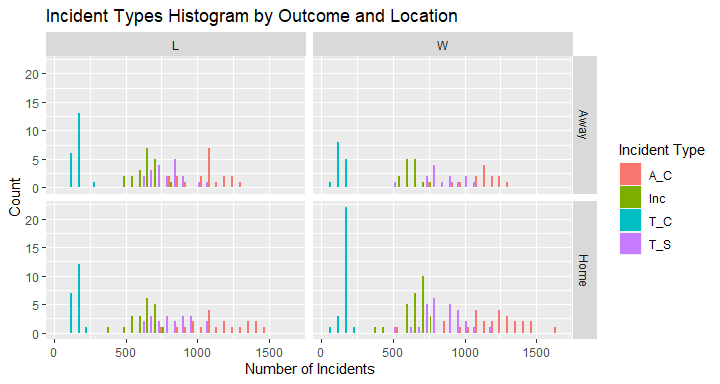
ggplot(Husker\_games, aes(T\_C)) + geom\_histogram(aes(y = ..density..)) +  
 stat\_function(fun = dnorm, args = list(mean = mean(Husker\_games$T\_C),   
 sd(Husker\_games$T\_C))) +   
 labs(title = "Traffic Crashes Histogram", y = "Density")



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")



# 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

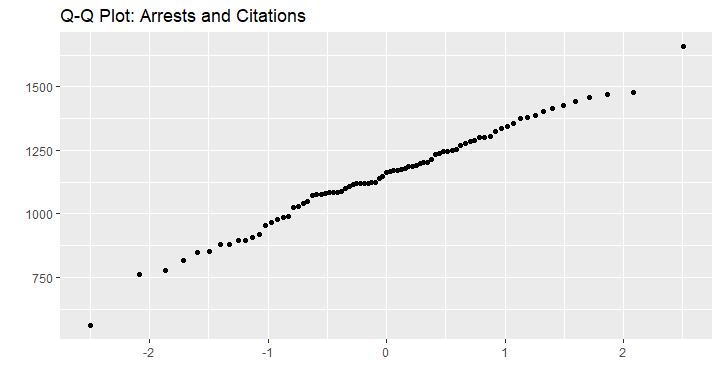
shapiro.test(Husker\_games$T\_C) # 0.005984 <- not normal

##   
## Shapiro-Wilk normality test  
##   
## data: Husker\_games$T\_C  
## W = 0.95463, p-value = 0.005984

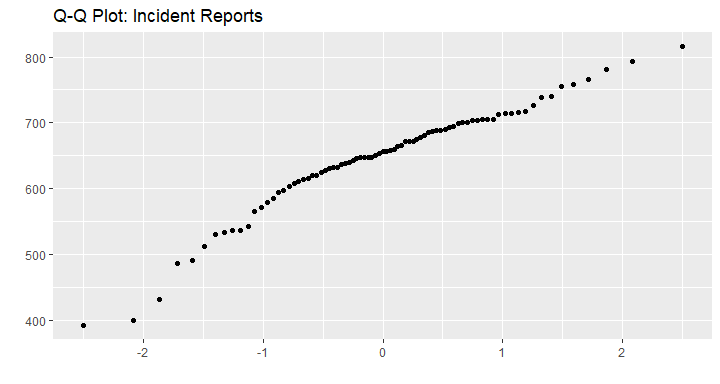
shapiro.test(Husker\_games$T\_S) # 0.8249

##   
## Shapiro-Wilk normality test  
##   
## data: Husker\_games$T\_S  
## W = 0.99058, p-value = 0.8249

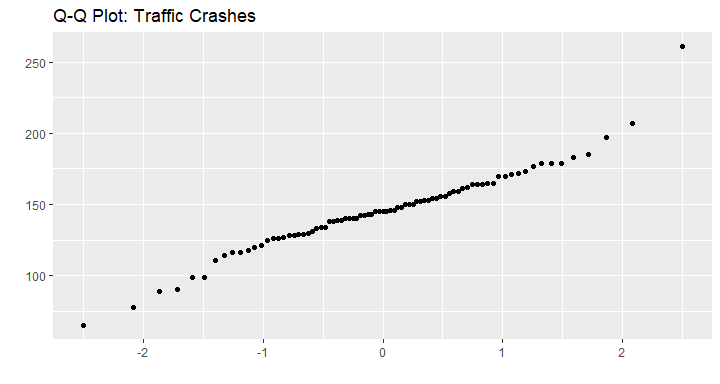
# Check for normalcy w/ qq plot  
qplot(sample = Husker\_games$A\_C) + labs(title = "Q-Q Plot: Arrests and Citations")



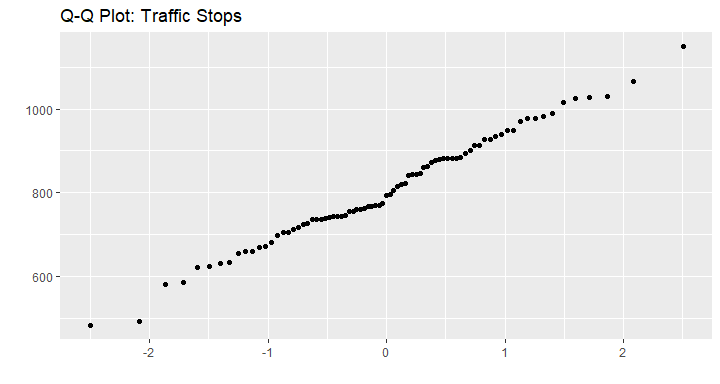
qplot(sample = Husker\_games$Inc) + labs(title = "Q-Q Plot: Incident Reports")



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.

# 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

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

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

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 = 0.70354, df = 79, p-value = 0.4838  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1418864 0.2922222  
## sample estimates:  
## cor   
## 0.07890761

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 = 1.684, df = 79, p-value = 0.09613  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.03356249 0.38870135  
## sample estimates:  
## cor   
## 0.1861511

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 = 1.1336, df = 79, p-value = 0.2604  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.09444637 0.33559110  
## sample estimates:  
## cor   
## 0.126512

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 = 1.5604, df = 79, p-value = 0.1227  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.04722325 0.37702406  
## sample estimates:  
## cor   
## 0.172909

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  
## Location 1.00000000 0.16060647 0.23390680 0.06366492 0.07516284 0.07890761  
## Win 0.16060647 1.00000000 0.17966908 0.11514848 -0.08404806 0.12651201  
## A\_C 0.23390680 0.17966908 1.00000000 0.34380817 0.08629274 0.72762685  
## Inc 0.06366492 0.11514848 0.34380817 1.00000000 0.35446651 0.08179958  
## T\_C 0.07516284 -0.08404806 0.08629274 0.35446651 1.00000000 -0.06809876  
## T\_S 0.07890761 0.12651201 0.72762685 0.08179958 -0.06809876 1.00000000  
## Tot\_Inc 0.18615110 0.17290905 0.94648432 0.50088760 0.19331084 0.82143529  
## Tot\_Inc  
## Location 0.1861511  
## Win 0.1729090  
## A\_C 0.9464843  
## Inc 0.5008876  
## T\_C 0.1933108  
## T\_S 0.8214353  
## Tot\_Inc 1.0000000

cor(H\_G.coded[c("Location", "Win", "A\_C", "Inc", "T\_C", "T\_S", "Tot\_Inc")])^2 \* 100

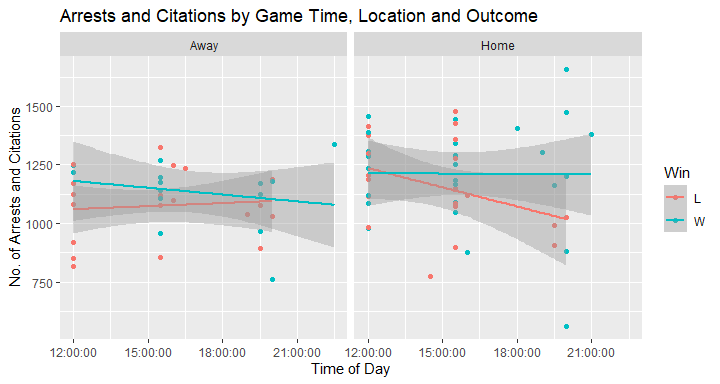
## Location Win A\_C Inc T\_C  
## Location 100.0000000 2.5794438 5.4712391 0.4053222 0.5649453  
## Win 2.5794438 100.0000000 3.2280977 1.3259173 0.7064077  
## A\_C 5.4712391 3.2280977 100.0000000 11.8204055 0.7446437  
## Inc 0.4053222 1.3259173 11.8204055 100.0000000 12.5646509  
## T\_C 0.5649453 0.7064077 0.7446437 12.5646509 100.0000000  
## T\_S 0.6226410 1.6005288 52.9440835 0.6691172 0.4637442  
## Tot\_Inc 3.4652231 2.9897539 89.5832566 25.0888389 3.7369080  
## T\_S Tot\_Inc  
## Location 0.6226410 3.465223  
## Win 1.6005288 2.989754  
## A\_C 52.9440835 89.583257  
## Inc 0.6691172 25.088839  
## T\_C 0.4637442 3.736908  
## T\_S 100.0000000 67.475593  
## Tot\_Inc 67.4755932 100.000000

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.

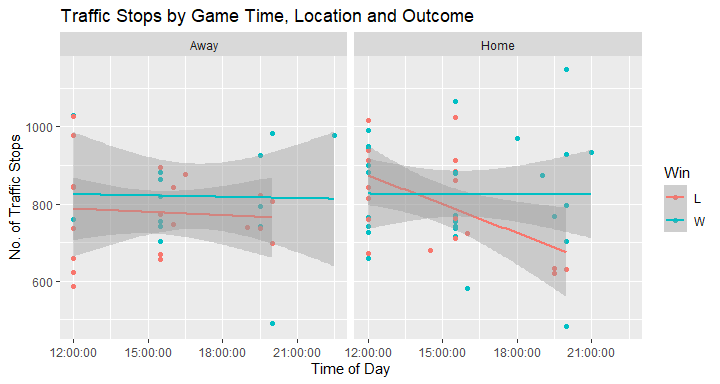
Looking at the R-squared values, we see that the some of the Public Safety Incident Types correlate pretty well with each other. Ignoring Total Incidents for a moment, we see that Arrests and Citations correlates really quite well with Traffic Stops, sharing 53% of variability. Traffic Stops and Incident 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.

# 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

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 moreso 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 forseeable future. Also, Go Big Red!