MY AMAZING FANTASTICAL PROJECT UPDATE

The Final Chapter

Project Aim

Main Question:

Given a time and information about the location, can I predict what type of crime is most likely occurring?

Cleaning

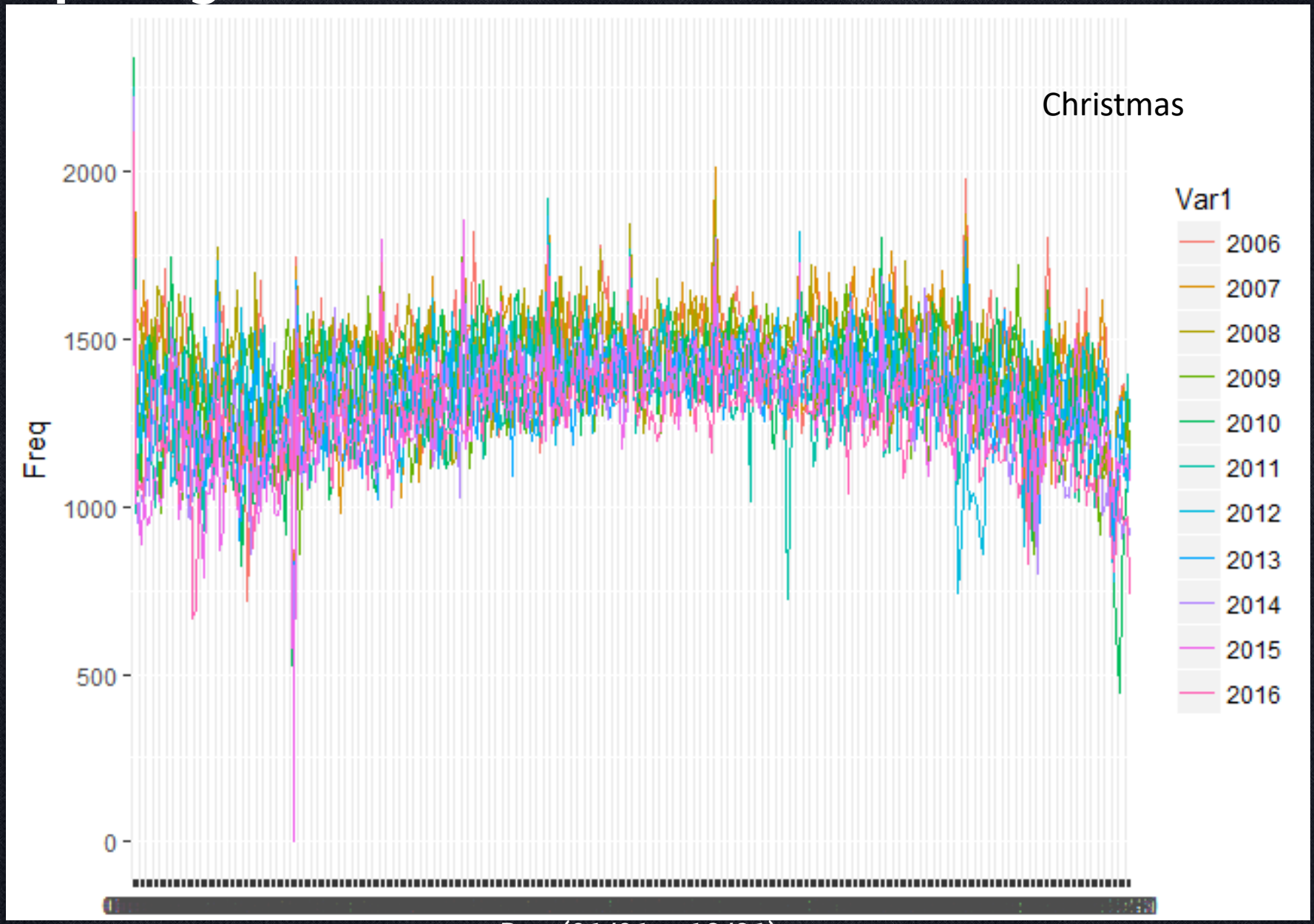
- Selected only for successful crimes
- Deleted unwanted variables
  - “CRM_ATPT_CPTD_CD", "ADDR_PCT_CD", "CMPLNT_NUM", "CMPLNT_TO_DT", "CMPLNT_TO_TM", "RPT_DT", "X_COORD_CD", "Y_COORD_CD", "Lat_Lon“
- Transformed PARK_NM, HADEVELOPT, and JURIS_DESC into indicator variables
- Set LOC_OF_OCCUR_DESC to indicator
  - Inside <- 1, Outside <- 0
  - Filled in missing info using logical rules based on PREM_DESC
- Selected for wanted years (2006-2016)
- Parsed Dates and Times into multi-variable columns
- Created Weekend and Holiday indicator variables
- Cleaned response variable data
  - Deleted points where both OFNS_DESC and LAW_CAT_CD were NA
  - Imputed NA OFNS_DESC points using PD_DESC values
- Reduced OFNS_DESC from 72 classifiers to 26
  - Deleted crimes committed at extremely low frequencies
  - Combined crimes of similar natures

Cleaning

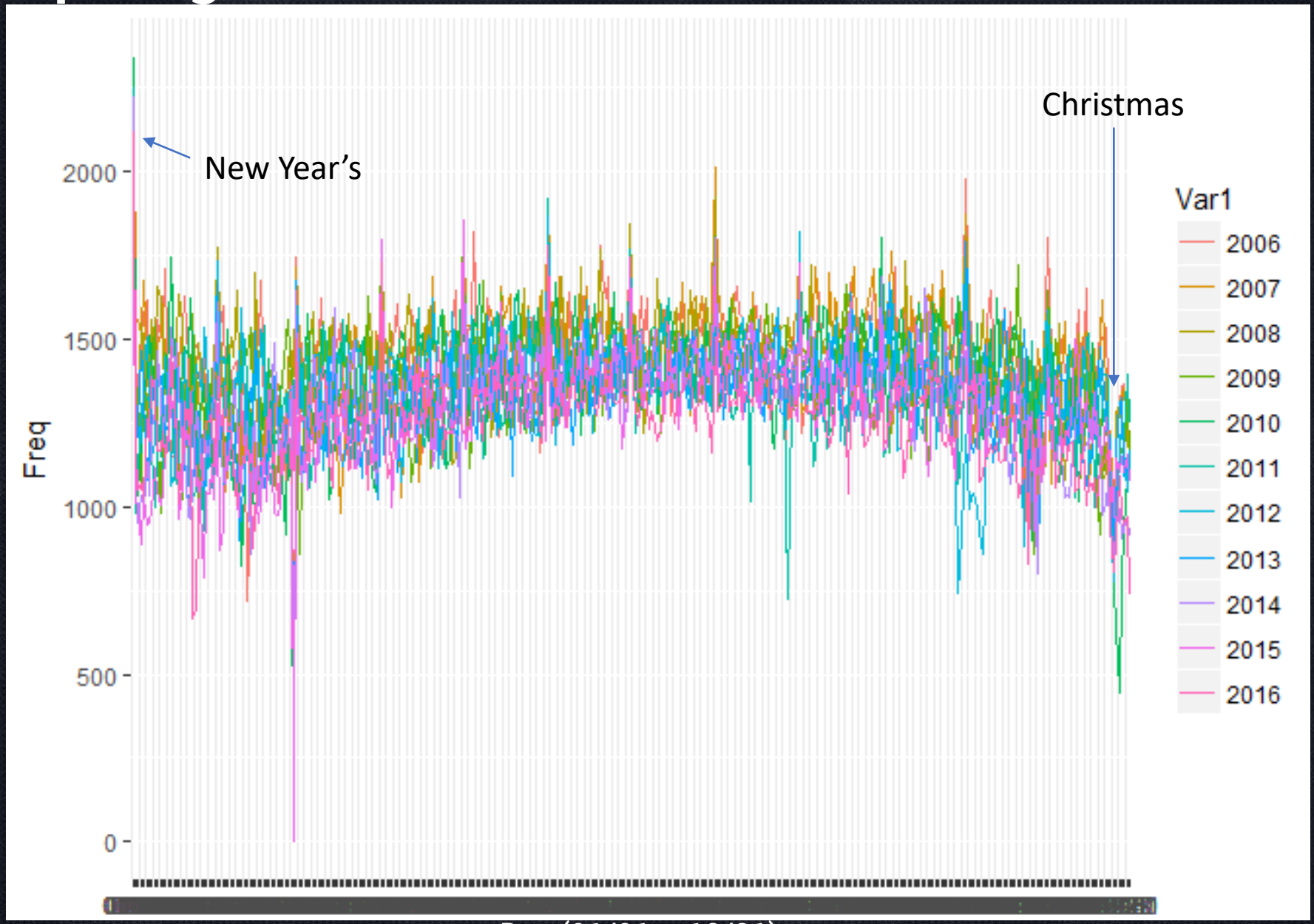
5,580,035 observations, 22 variables, 1.36Gb

5,217,799, observations, 20 variables, 760Mb

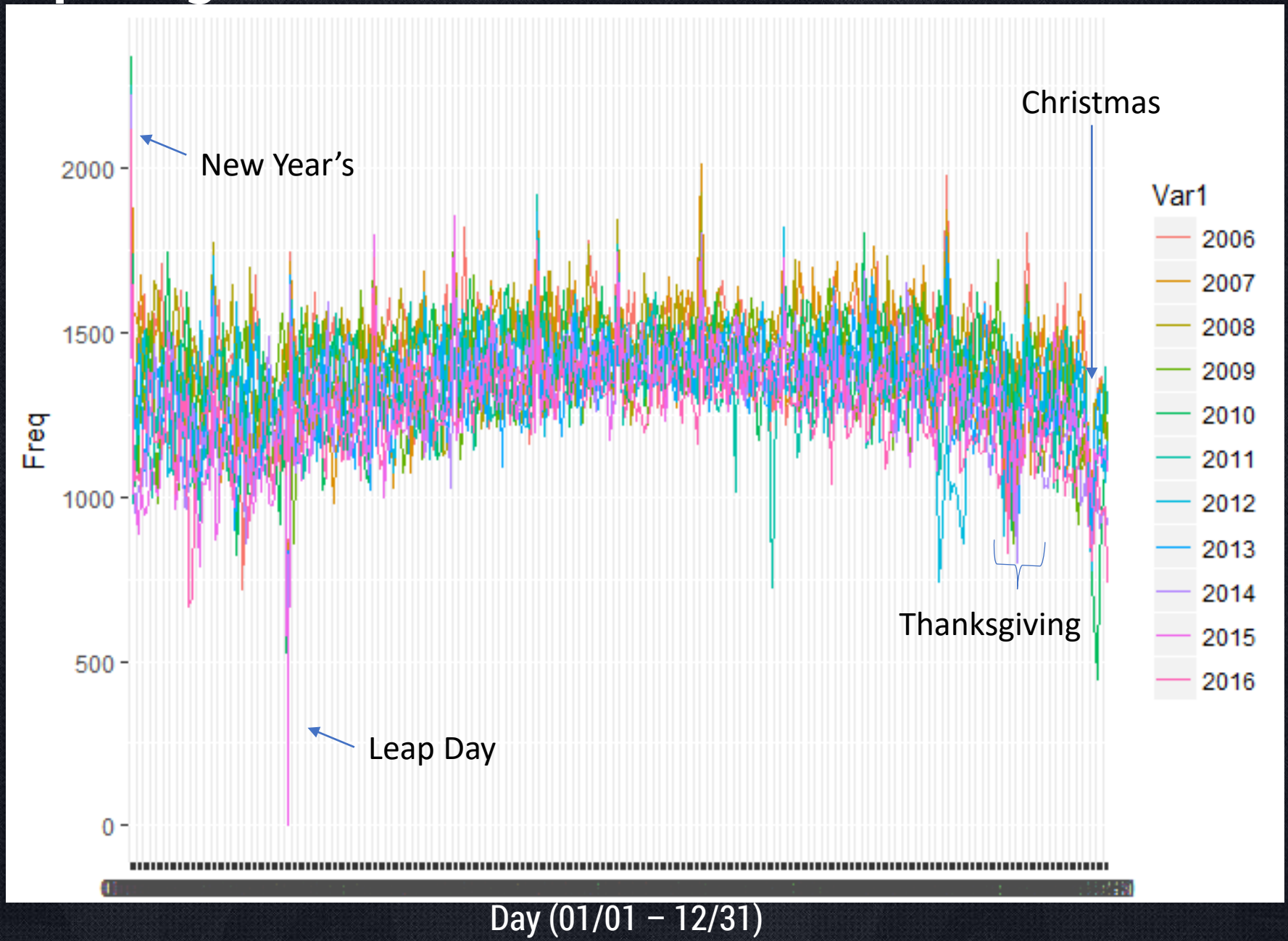
Graphing


Christmas
2000
1500
1000
500
0
Freq
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Day (01/01 – 12/31)

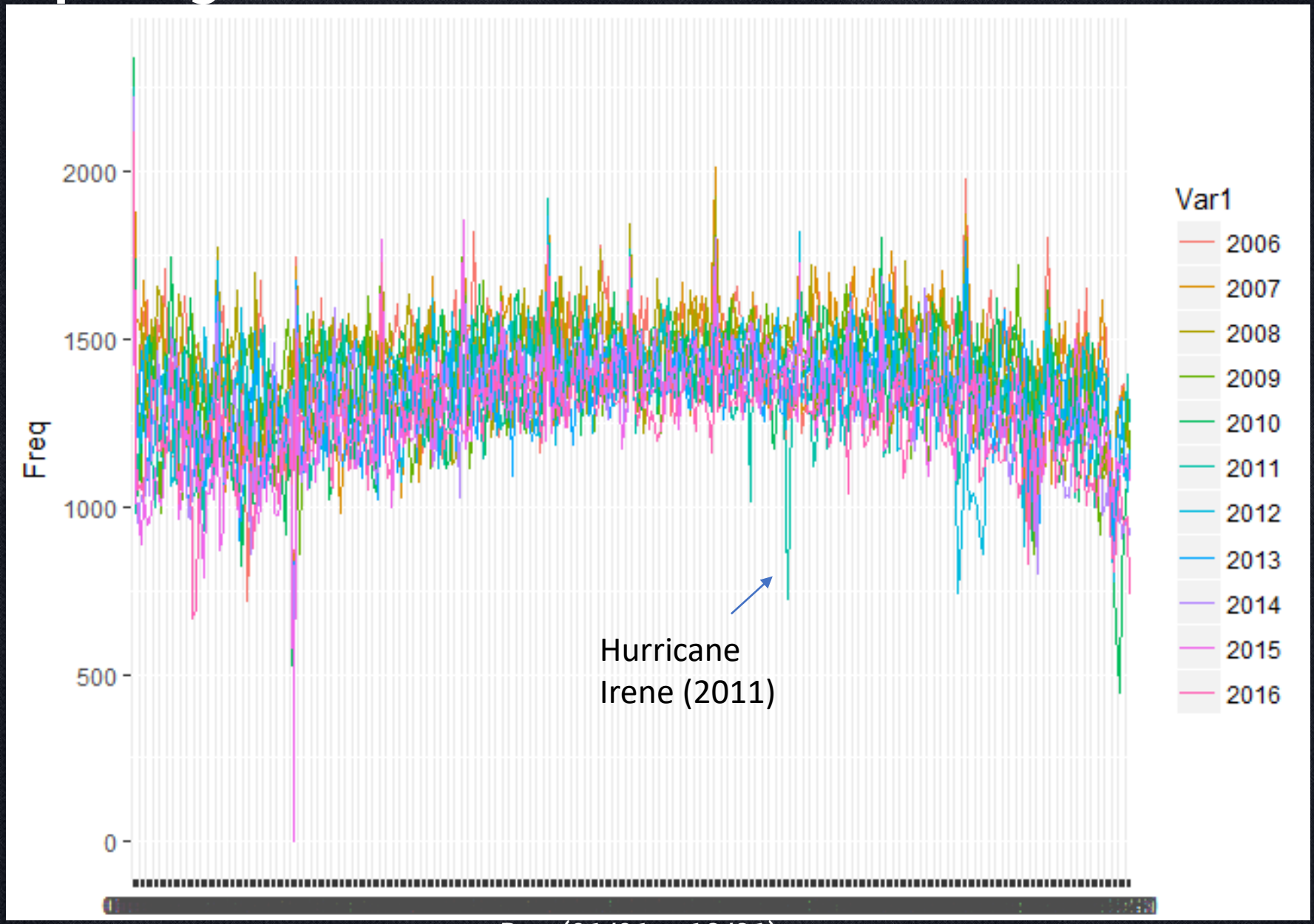
Graphing


Christmas
New Year's
2000
1500
1000
500
0
Freq
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Day (01/01 – 12/31)

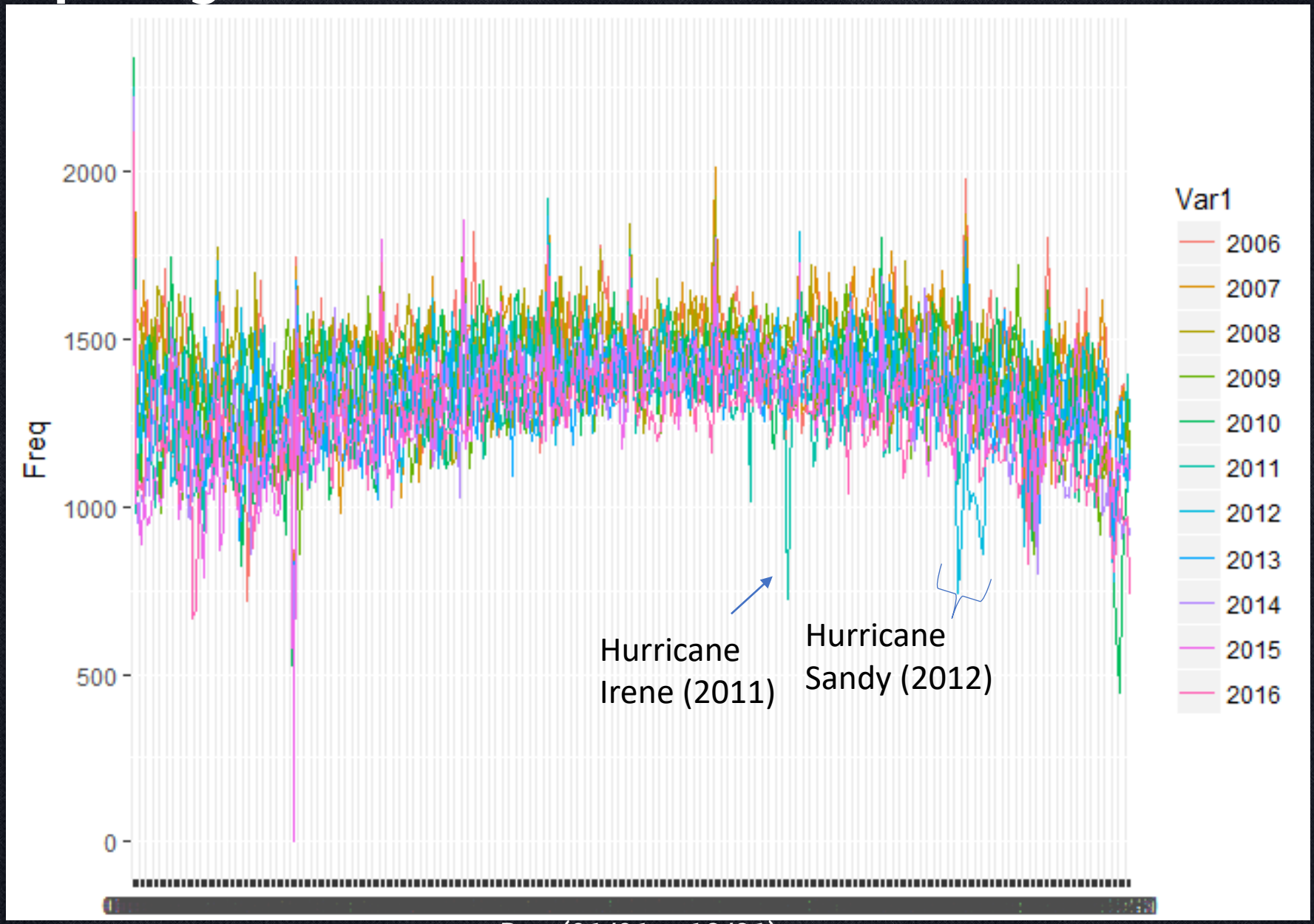
Graphing


Christmas
New Year's
2000
1500
1000
500
0
Freq
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Thanksgiving
Leap Day
Day (01/01 – 12/31)

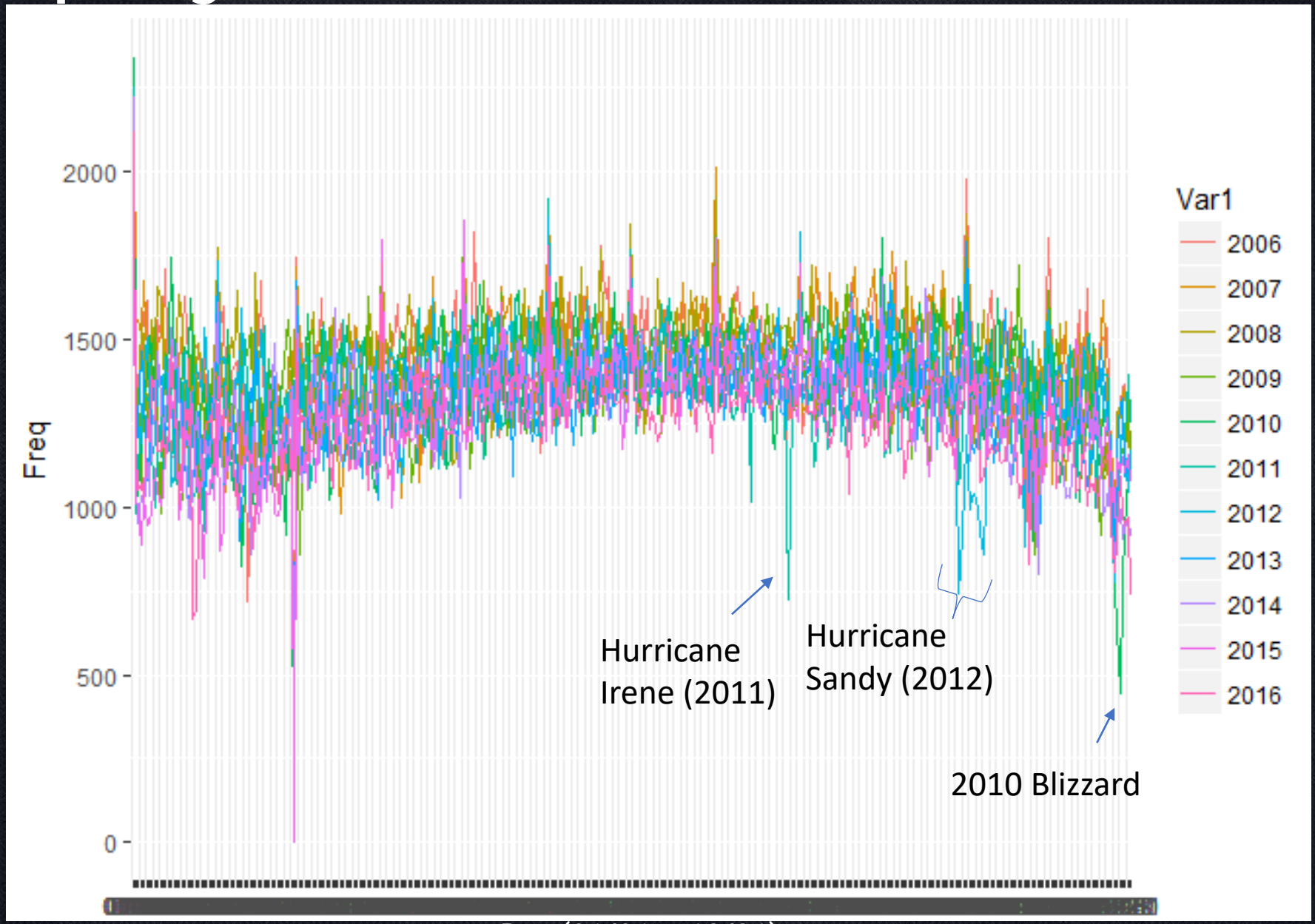
Graphing


2000
1500
1000
500
0
Freq
Hurricane
Irene (2011)
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Day (01/01 – 12/31)

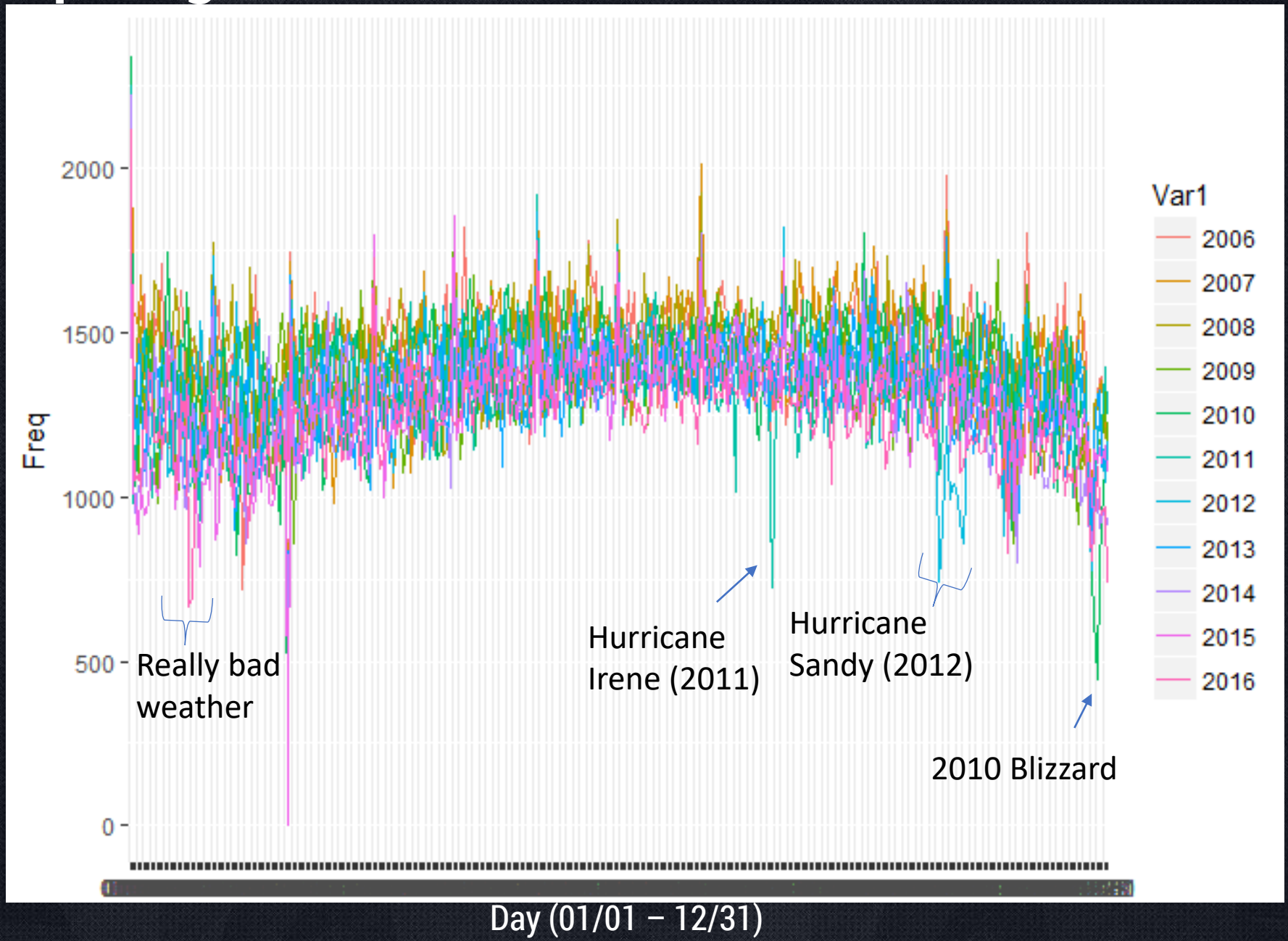
Graphing


2000
1500
1000
500
0
Freq
Hurricane
Irene (2011)
Hurricane
Sandy (2012)
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Day (01/01 – 12/31)

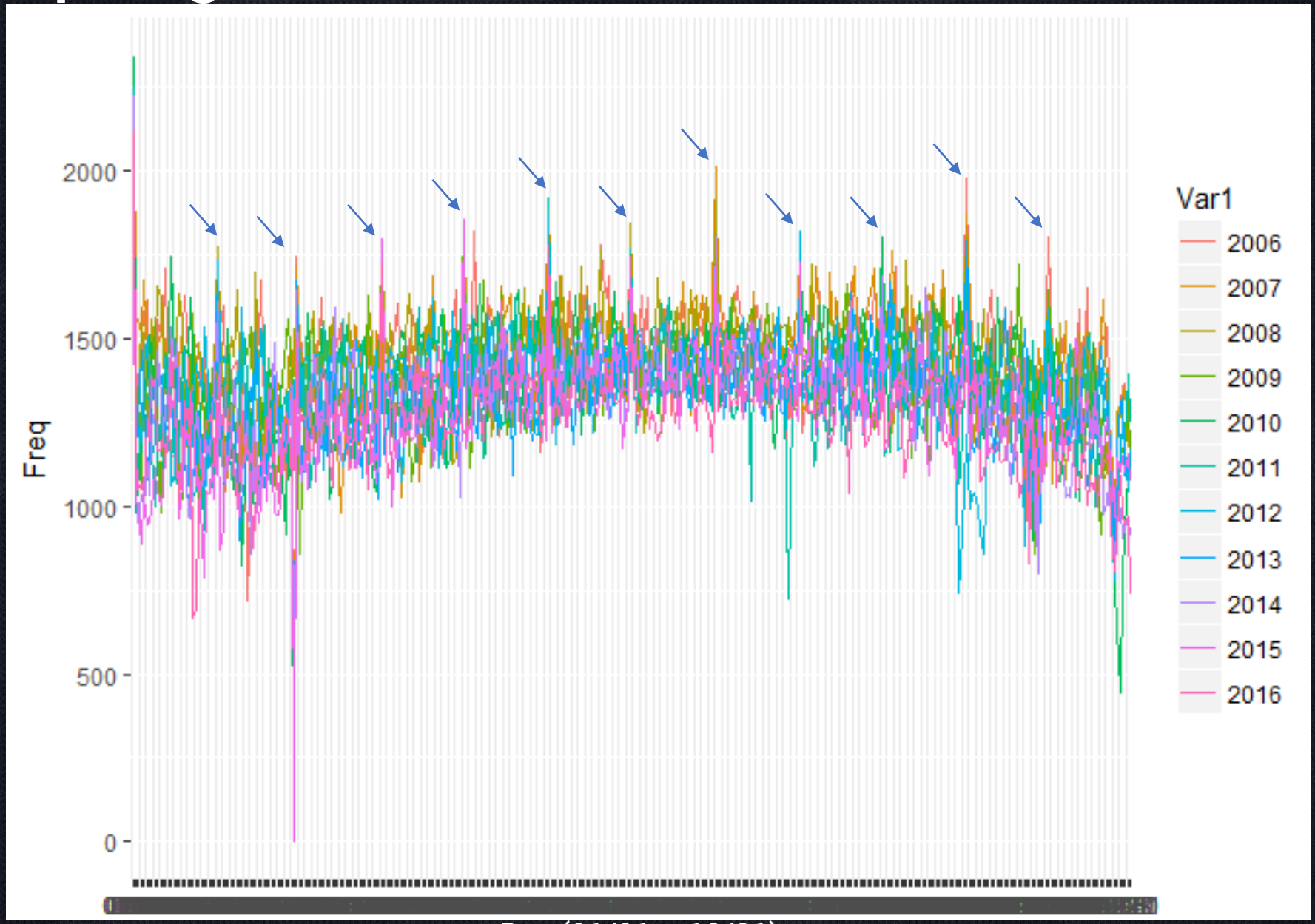
Graphing


Freq
2000
1500
1000
500
0
Hurricane Irene (2011)
Hurricane Sandy (2012)
2010 Blizzard
Day (01/01 – 12/31)
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016

Graphing

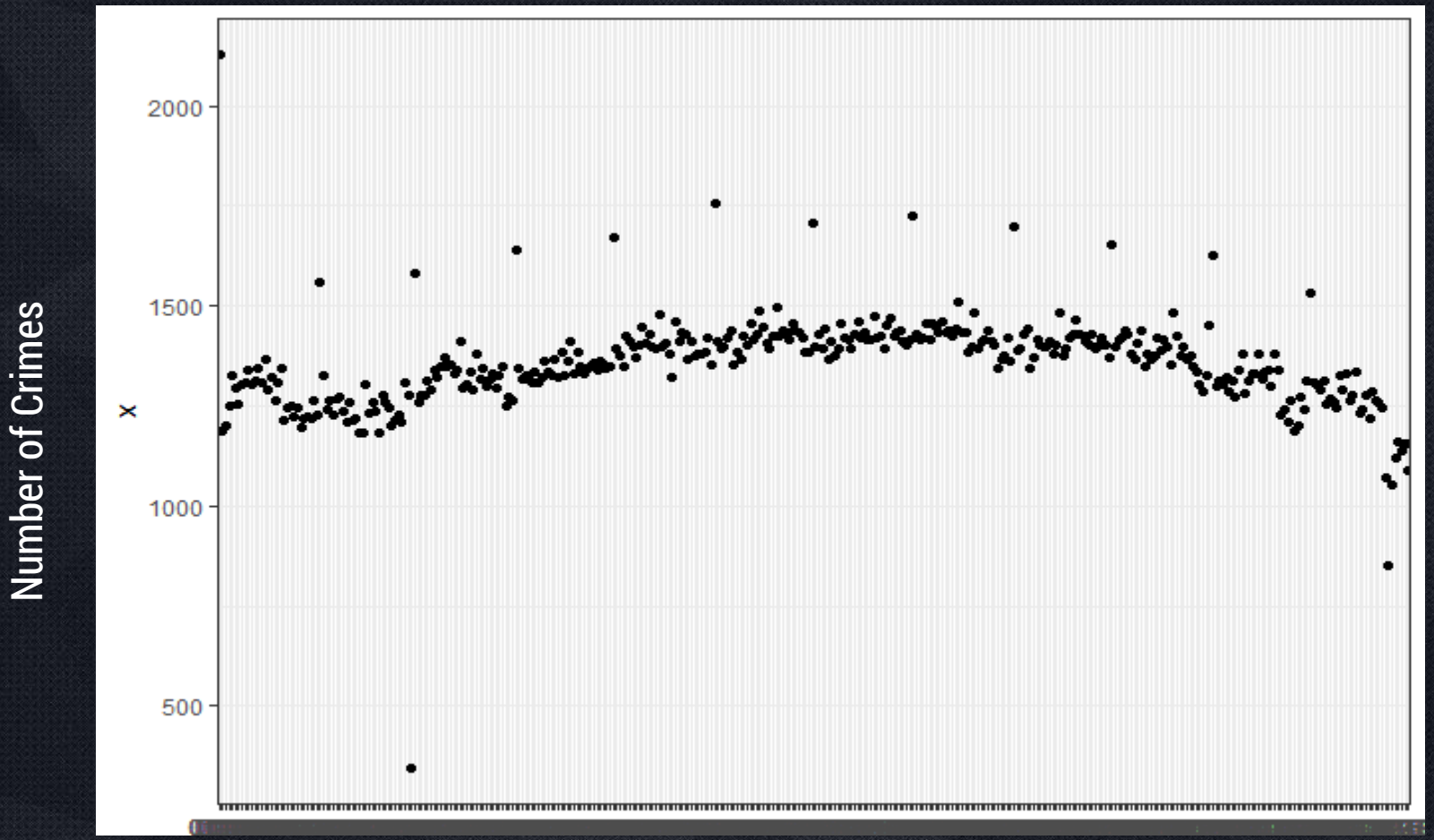

2000
1500
1000
500
0
Freq
Really bad weather
Hurricane Irene (2011)
Hurricane Sandy (2012)
2010 Blizzard
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Day (01/01 – 12/31)

Graphing

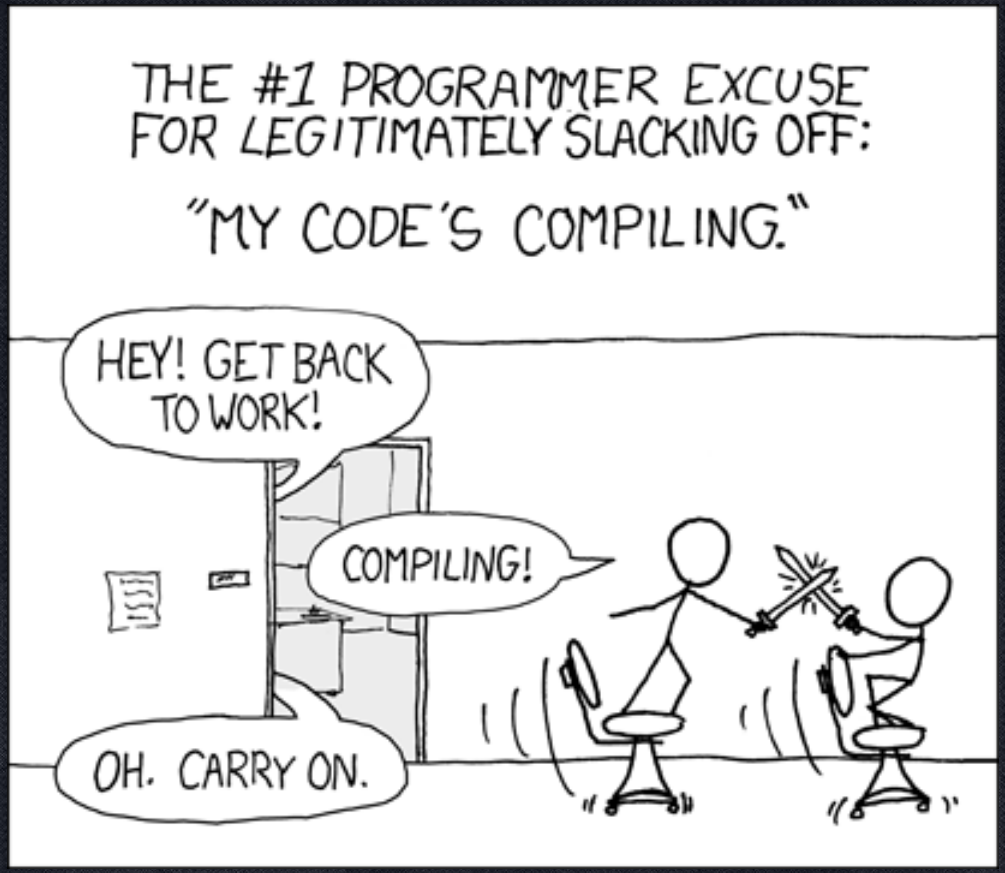

2000
1500
1000
500
0
Freq
Var1
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Day (01/01 – 12/31)

Cleaning—``The First Effect"

Average #Crimes/Day (2006-2016)



Difficulties



Project Aim, v2.

Main Question:

Given a time and information about the location, can I predict what type of crime is most likely occurring?

Given the time and location of the event, can I determine if Criminal Mischief or Miscellaneous Offenses is being committed?

Cleaning Summary

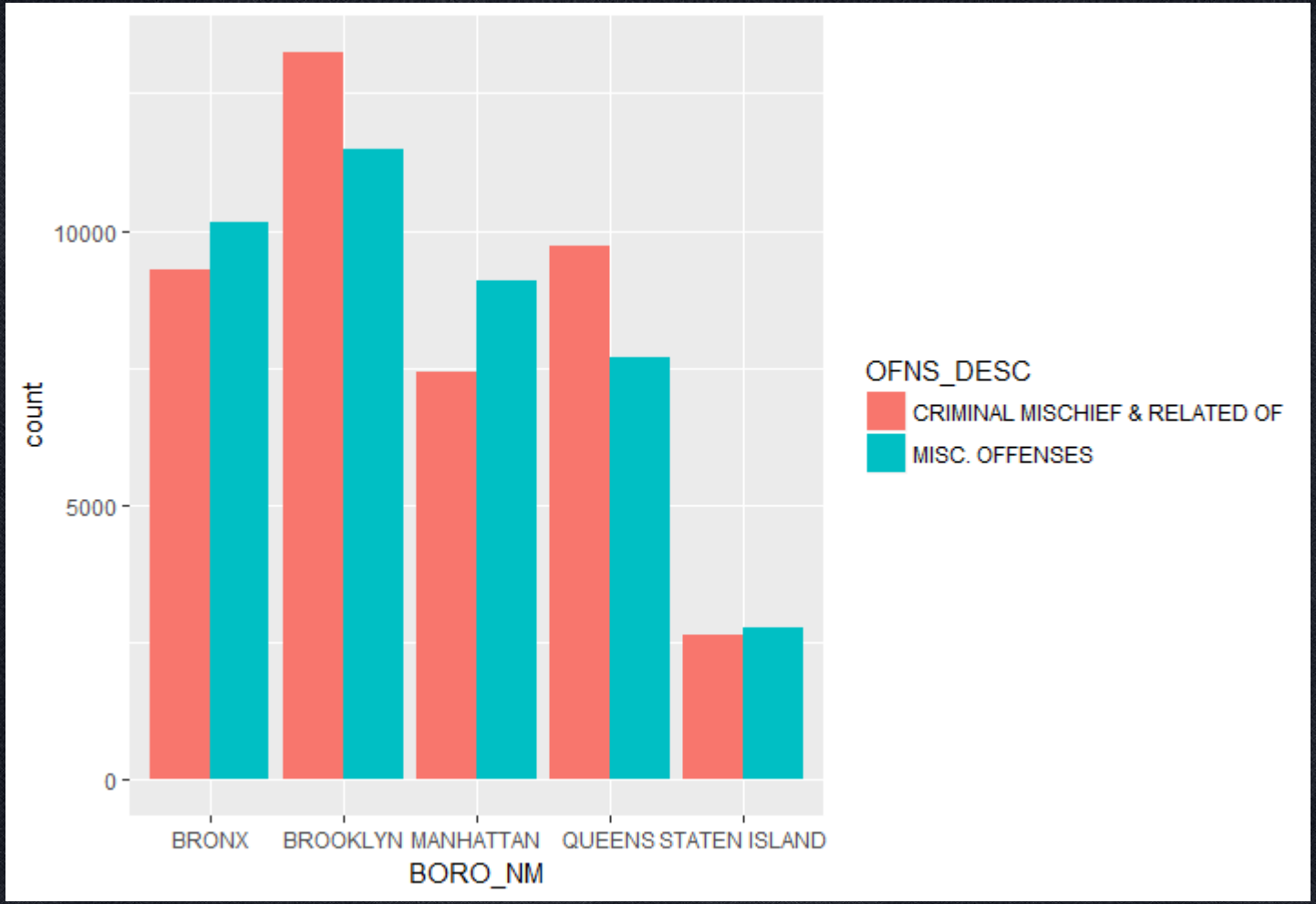
5,217,799, observations, 20 variables, 760Mb
(26 crimes, 10 years)

83,479 observations, 20 predictors, 13Mb
(2 crimes, 1 year)

10,000, 3 predictors, XXKb?
(Randomly sampled)

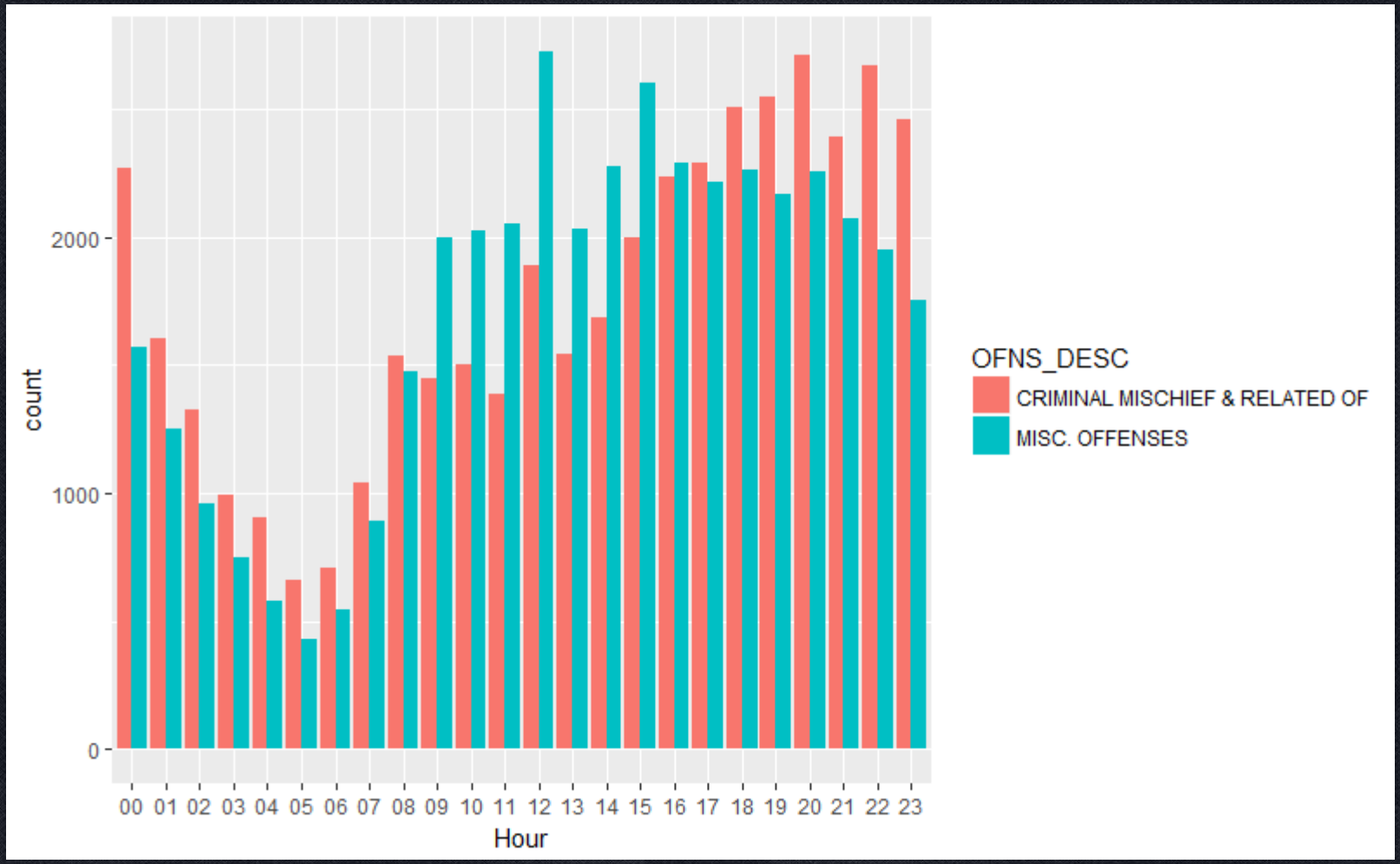
Preliminary Analysis

Boro



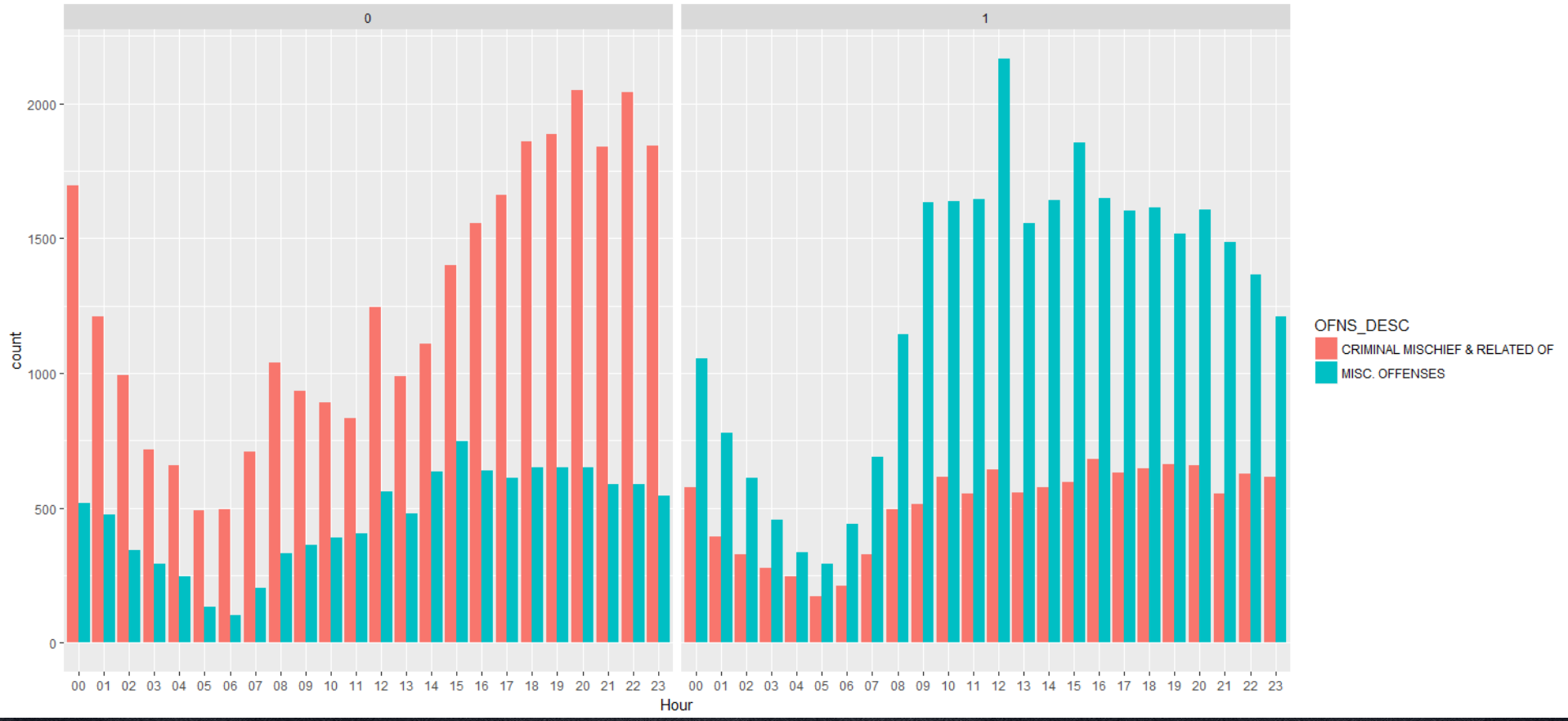
Preliminary Analysis

Hour



Preliminary Analysis

Location

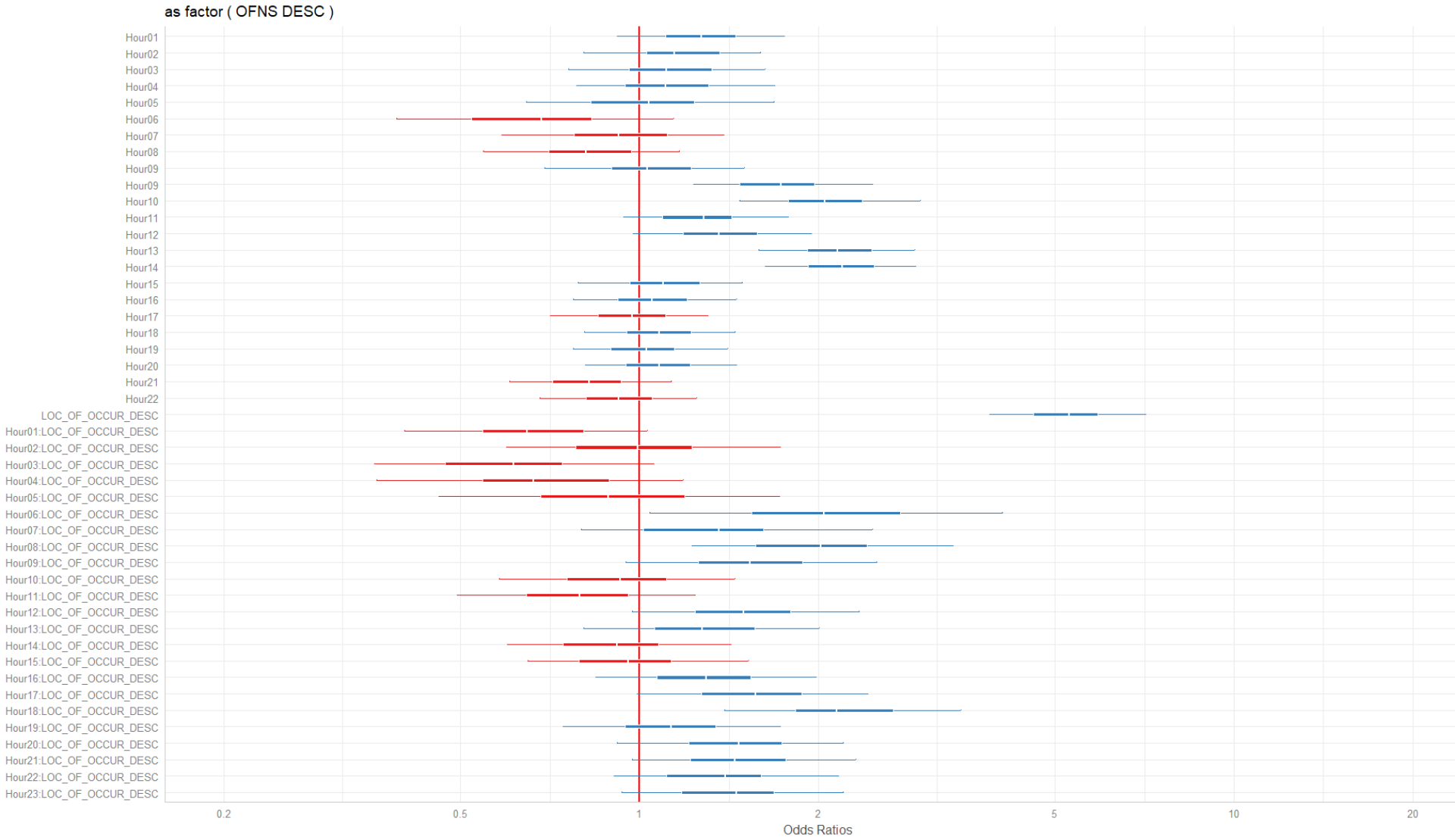


Model

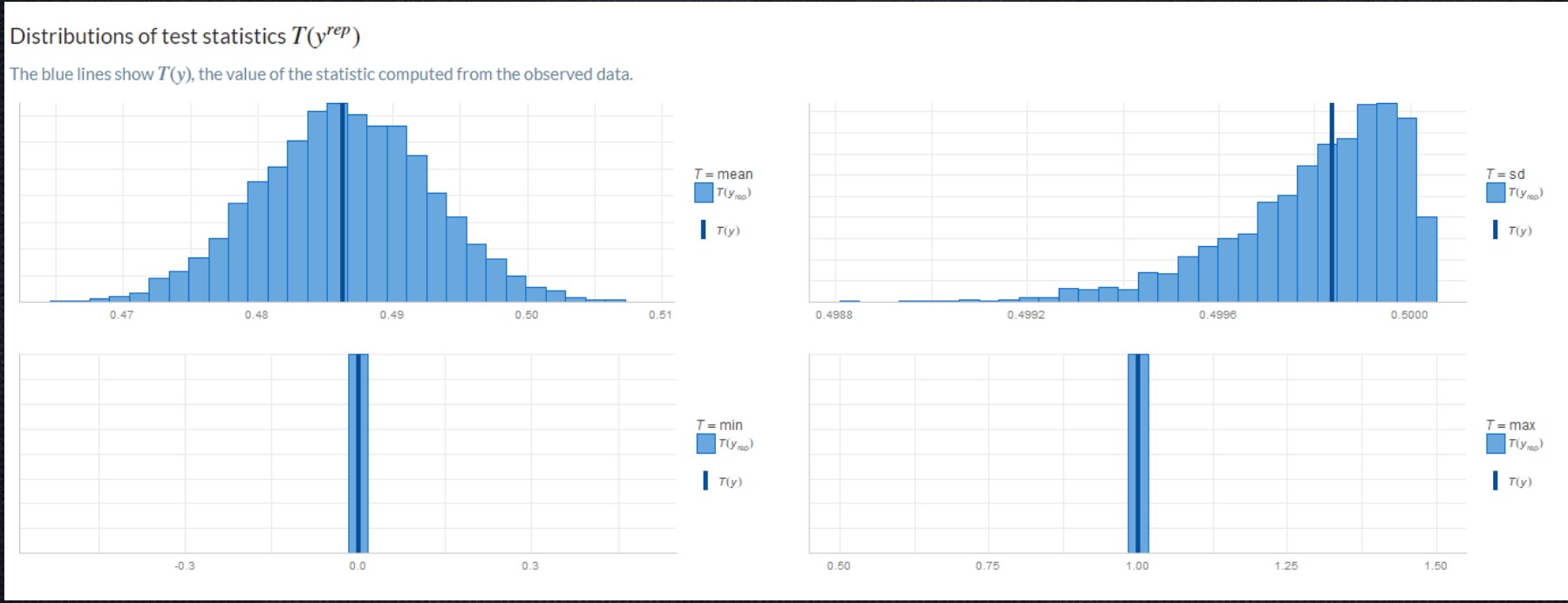
$$Crime \sim Hour + Location + Hour : Location + (1|Boro)$$

```
stan_glmer(as.factor(Crimes) ~ Location+Hour+Hour:Location+(1|BoroName), data=fake_data_df_2,
family=binomial(link="logit"))
```

Model Results



PP Checks – Real Data



Fake it 'til you make it

- Create empty matrix, rename column names
- Choose select variables to have value 1, creating the data

```r
#make an empty matrix
colnum <- 32
nsims <- 5000

fake_data1 <- matrix(0, nsims, colnum)

#change the column names to be more helpful
my_names <- c('Crimes', 'Location', 'Queens', 'Manhattan', 'StatenIsland', 'Bronx', 'Brooklyn', 'H1', 'H2', 'H3', 'H4', 'H5', 'H6', 'H7'
, 'H8', 'H9', 'H10', 'H11', 'H12', 'H13', 'H14', 'H15', 'H16', 'H17', 'H18', 'H19', 'H20', 'H21', 'H22', 'H23','BoroName', 'Hour')

colnames(fake_data1) <- my_names

#create lists of these column names, separated by type (location, neighborhood, hour)
loc_values <- c(0, 1)
boro_names <- c("Bronx", "Brooklyn", "Manhattan", "Queens", "StatenIsland")
hour_names <- c('H0','H1', 'H2', 'H3', 'H4', 'H5', 'H6', 'H7', 'H8', 'H9', 'H10', 'H11', 'H12', 'H13', 'H14', 'H15', 'H16', 'H17', 'H18'
, 'H19', 'H20', 'H21', 'H22', 'H23')

#for each row in the matrix, choose one column from each list and set the value in that row:column to 1. This ensures that only column
of each type will have a value.
for (i in 1:nsims){
  random_boro <- sample(x=boro_names, size=1, replace=TRUE, prob=hood_prob)
  random_loc <- sample(x=loc_values, size=1, replace=TRUE, prob=rep(1/2,2))
  random_hour <- sample(x=hour_names, size=1, replace=TRUE, prob=hour_prob)

  fake_data1[i, random_boro] <- 1
  fake_data1[i, "Location"] <- random_loc
  if (random_hour!="H0"){
    fake_data1[i, random_hour] <- 1
  }
}
```

Fake it ‘til you make it

```
> head(fake_data3)
     Crimes Location Queens Manhattan StatenIsland Bronx Brooklyn H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 H13 H14 H15 H16 H17 H18 H19 H20 H21 H22 H23 BoroName Hour
[1,]      0        1      0         1           0     0        0  0  0  0  0  0  0  0  0  0   1   0   0   0   0   0   0   0   0   0   0   0   0        0    0
[2,]      0        1      1         0           0     0        0  0  0  1  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0   0        0    0
[3,]      0        0      1         0           0     0        0  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   1   0   0   0   0   0   0        0    0
[4,]      0        0      0         0           0     1        0  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   1   0   0   0   0        0    0
[5,]      0        0      0         0           0     0        1  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0   0        0    0
[6,]      0        1      0         0           1     0        0  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   1   0   0   0   0   0   0        0    0
>
```

Location

Boro

Hour

Categorical Variables made based on dummy variable values

Fake it ‘til you make it

- Make a few coefficients

```
#coefficients for hours
mu_in <- 1.2
mu_1 <- 0.2
mu_2 <- 0.1
mu_3 <- 0.1
mu_4 <- 0.1
mu_5  <- 0.0
mu_6  <- -0.4
mu_7 <- -0.1
mu_8 <- -0.2
mu_9 <- 0.7
mu_10 <- 0.7
mu_11 <- 0.9
mu_12 <- 0.8
mu_13 <- 0.8
mu_14  <- 0.9
mu_15 <- 0.9
mu_16 <- 0.1
mu_17 <- 0.0
mu_18 <- 0.0
mu_19 <- 0.1
mu_20 <- 0.0
mu_21 <- 0.1
mu_22  <- -0.2
mu_23 <- -0.1
```

```
#coefficients for interaction terms
mu_loc_1 <- -0.4
mu_loc_2 <- 0.0
mu_loc_3 <- -0.5
mu_loc_4 <- -0.4
mu_loc_5 <- -0.1
mu_loc_6 <- 0.7
mu_loc_7 <- 0.3
mu_loc_8 <- 0.7
mu_loc_9 <- 0.4
mu_loc_10 <- -0.1
mu_loc_11 <- -0.2
mu_loc_12 <- 0.4
mu_loc_13 <- 0.2
mu_loc_14 <- -0.1
mu_loc_15 <- 0.0
mu_loc_16 <- 0.3
mu_loc_17 <- 0.4
mu_loc_18 <- 0.8
mu_loc_19 <- 0.1
mu_loc_20 <- 0.4
mu_loc_21 <- 0.4
mu_loc_22 <- 0.3
mu_loc_23 <- 0.4
```

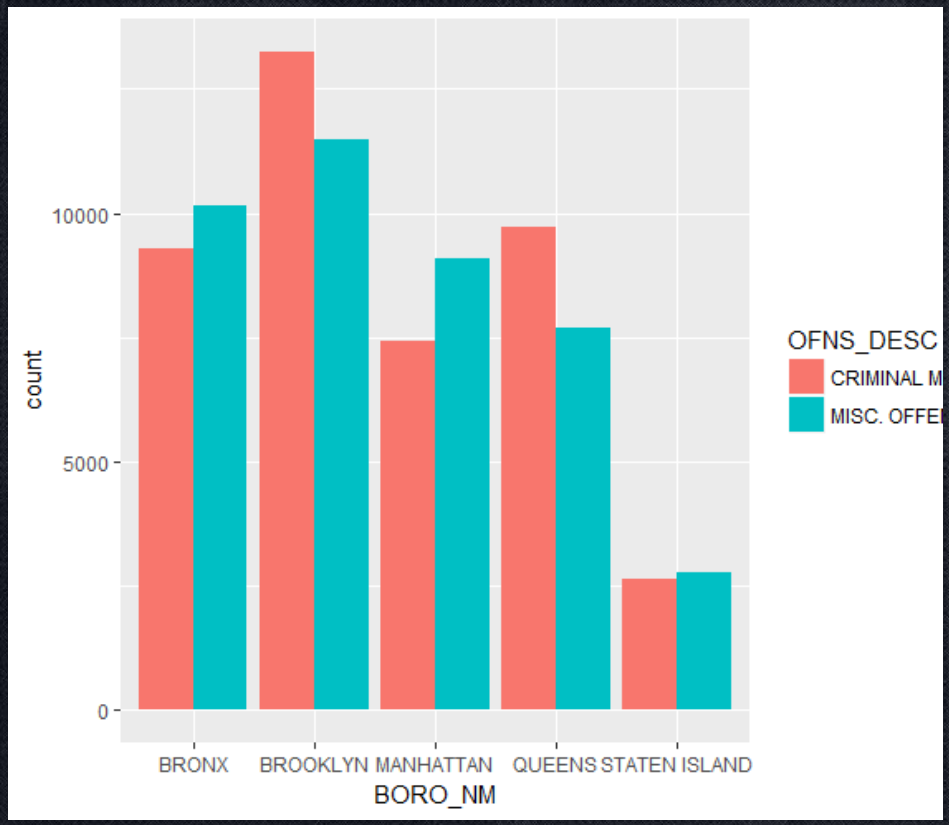
Fake it 'til you make it

- Create the response variable
- Y = Hour + Location + Hour : Location + Boro

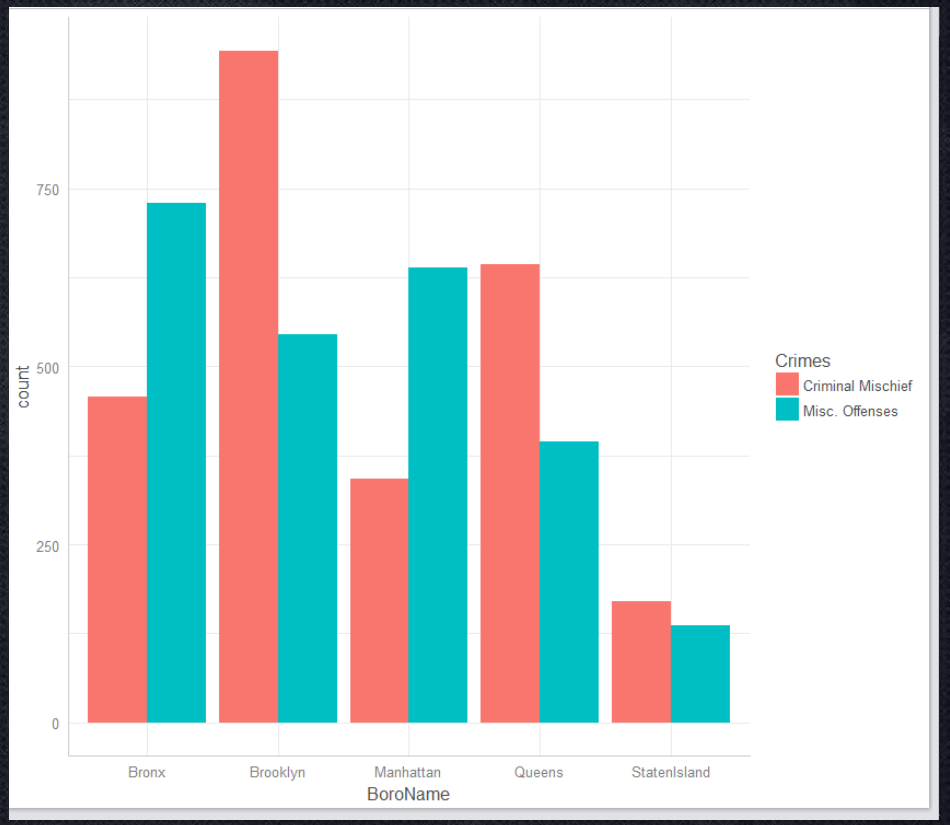
```
for (i in 1:nsims){
  y <- mu_in*as.numeric(fake_data3a[[i, 'Location']])+mu_1*as.numeric(fake_data3a[[i, 'H1']])+mu_2*as.numeric(fake_data3a[[i, 'H2']])+mu_3*as.numeric(fake_data3a[[i, 'H3']])+mu_4*as.numeric(fake_data3a[[i, 'H4']])+mu_5
*as.numeric(fake_data3a[[i, 'H5']])+mu_6*as.numeric(fake_data3a[[i, 'H6']])+mu_7*as.numeric(fake_data3a[[i, 'H7']])+mu_8*as.numeric(fake_data3a[[i, 'H8']])+mu_9*as.numeric(fake_data3a[[i, 'H9']])+mu_10*as.numeric
(fake_data3a[[i, 'H10']])+mu_11*as.numeric(fake_data3a[[i, 'H11']])+mu_12*as.numeric(fake_data3a[[i, 'H12']])+mu_13*as.numeric(fake_data3a[[i, 'H13']])+mu_14*as.numeric(fake_data3a[[i, 'H14']])+mu_15*as.numeric(fake_dat
a3a[[i, 'H15']])+mu_16*as.numeric(fake_data3a[[i, 'H16']])+mu_17*as.numeric(fake_data3a[[i, 'H17']])+mu_18*as.numeric(fake_data3a[[i, 'H18']])+mu_19*as.numeric(fake_data3a[[i, 'H19']])+mu_20*as.numeric(fake_data3a[[i,
'H20']])+mu_21*as.numeric(fake_data3a[[i, 'H21']])+mu_22*as.numeric(fake_data3a[[i, 'H22']])+mu_23*as.numeric(fake_data3a[[i, 'H23']]) +mu_loc_1*as.numeric(fake_data3a[[i, 'H1']])*as.numeric(fake_data3a[[i, 'Location']]
)+mu_loc_2*as.numeric(fake_data3a[[i, 'H2']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_3*as.numeric(fake_data3a[[i, 'H3']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_4*as.numeric(fake_data3a[[i, 'H4']])*as
.numeric(fake_data3a[[i, 'Location']])+mu_loc_5*as.numeric(fake_data3a[[i, 'H5']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_6*as.numeric(fake_data3a[[i, 'H6']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_7
*as.numeric(fake_data3a[[i, 'H7']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_8*as.numeric(fake_data3a[[i, 'H8']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_9*as.numeric(fake_data3a[[i, 'H9']])*as.numeric
(fake_data3a[[i, 'Location']])+mu_loc_10*as.numeric(fake_data3a[[i, 'H10']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_11*as.numeric(fake_data3a[[i, 'H11']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_12*as
.numeric(fake_data3a[[i, 'H12']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_13*as.numeric(fake_data3a[[i, 'H13']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_14*as.numeric(fake_data3a[[i, 'H14']])*as.numeric
(fake_data3a[[i, 'Location']])+mu_loc_15*as.numeric(fake_data3a[[i, 'H15']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_16*as.numeric(fake_data3a[[i, 'H16']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_17*as
.numeric(fake_data3a[[i, 'H17']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_18*as.numeric(fake_data3a[[i, 'H18']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_19*as.numeric(fake_data3a[[i, 'H19']])*as.numeric
(fake_data3a[[i, 'Location']])+mu_loc_20*as.numeric(fake_data3a[[i, 'H20']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_21*as.numeric(fake_data3a[[i, 'H21']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_22*as
.numeric(fake_data3a[[i, 'H22']])*as.numeric(fake_data3a[[i, 'Location']])+mu_loc_23*as.numeric(fake_data3a[[i, 'H23']])*as.numeric(fake_data3a[[i, 'Location']]) +mu_Queen*as.numeric(fake_data3a[[i, 'Queens']])+mu_Man
*as.numeric(fake_data3a[[i, 'Manhattan']])+mu_Bronx*as.numeric(fake_data3a[[i, 'Bronx']])+mu_Brook*as.numeric(fake_data3a[[i, 'Brooklyn']])+mu_SI*as.numeric(fake_data3a[[i, 'StatenIsland']])

  if (y<1.0){
    my_crime='Criminal Mischief'
  }
  else if (y>=1.0){
    my_crime='Misc. Offenses'
  }
  fake_data1a[[i, 'Crimes']] <- my_crime
}
```

Close Enough?

- Create some coefficients, taken from real data model



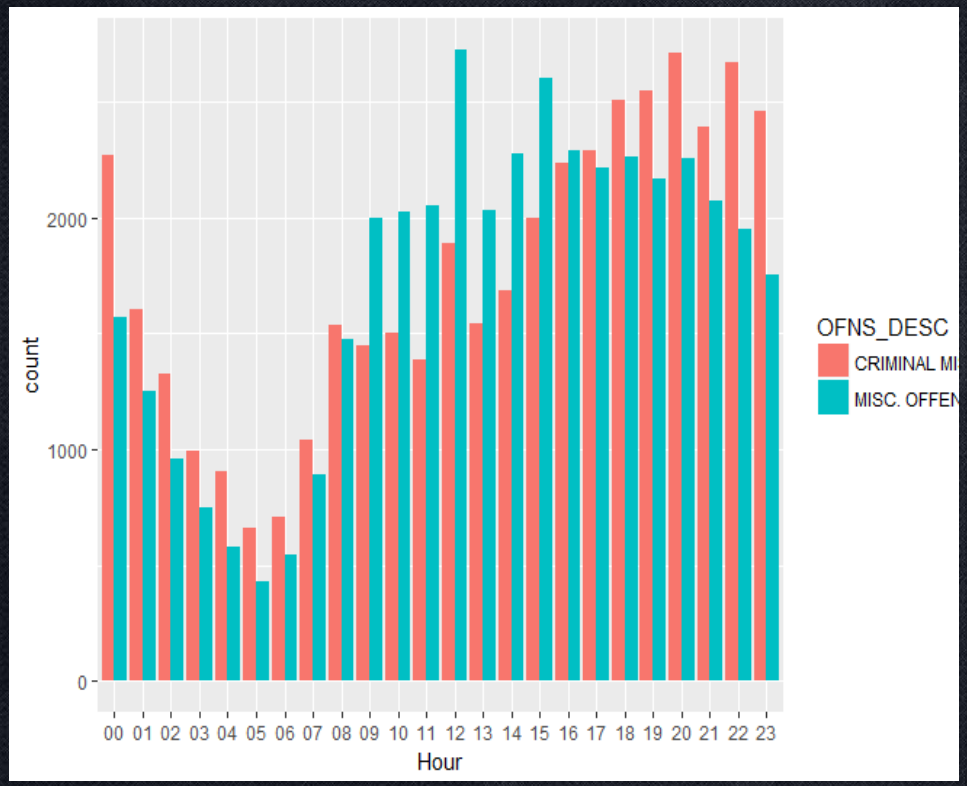
REAL



FAKE

Close Enough?

- Create some coefficients, taken from real data model



REAL

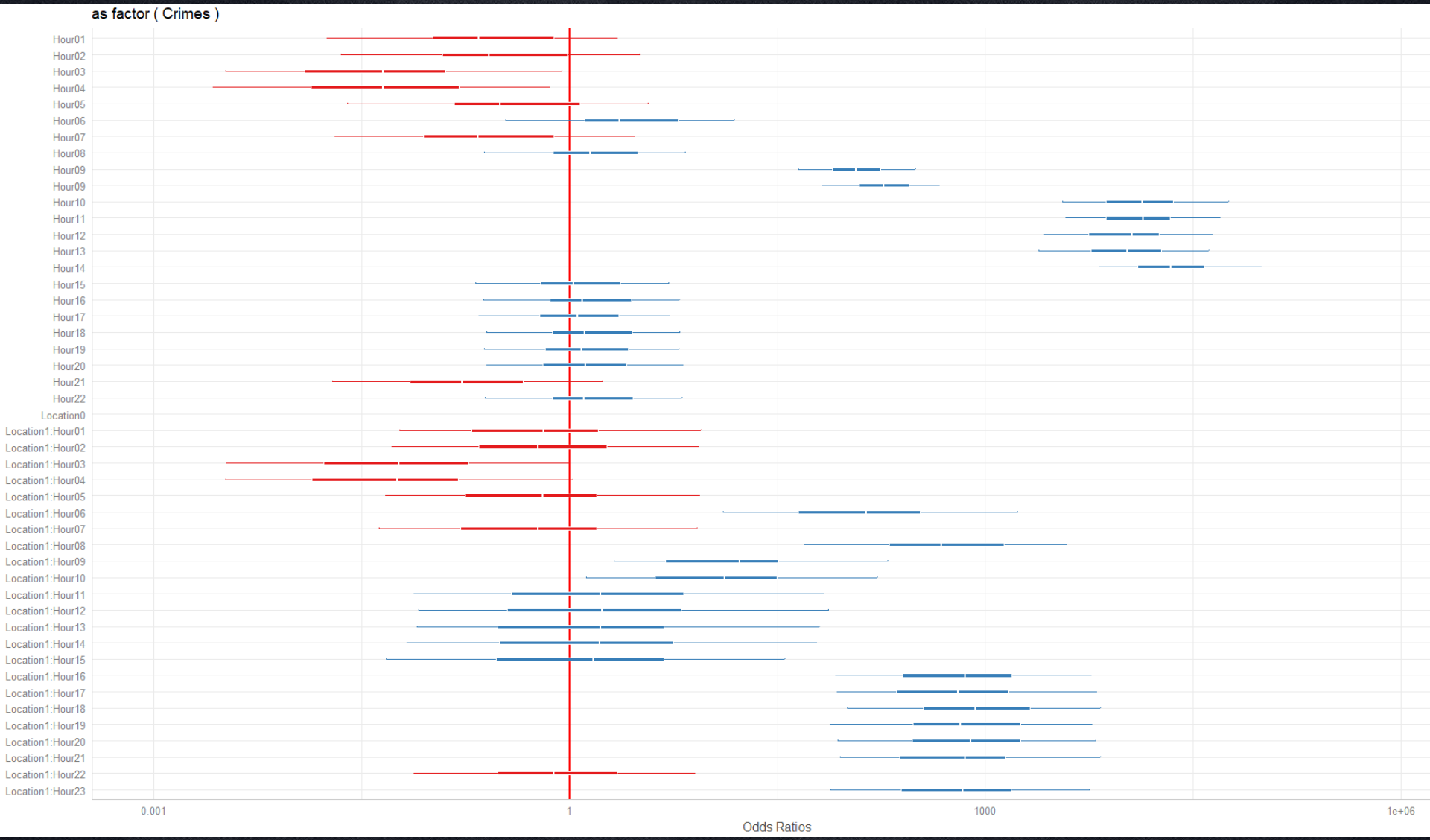
FAKE

Fake Model

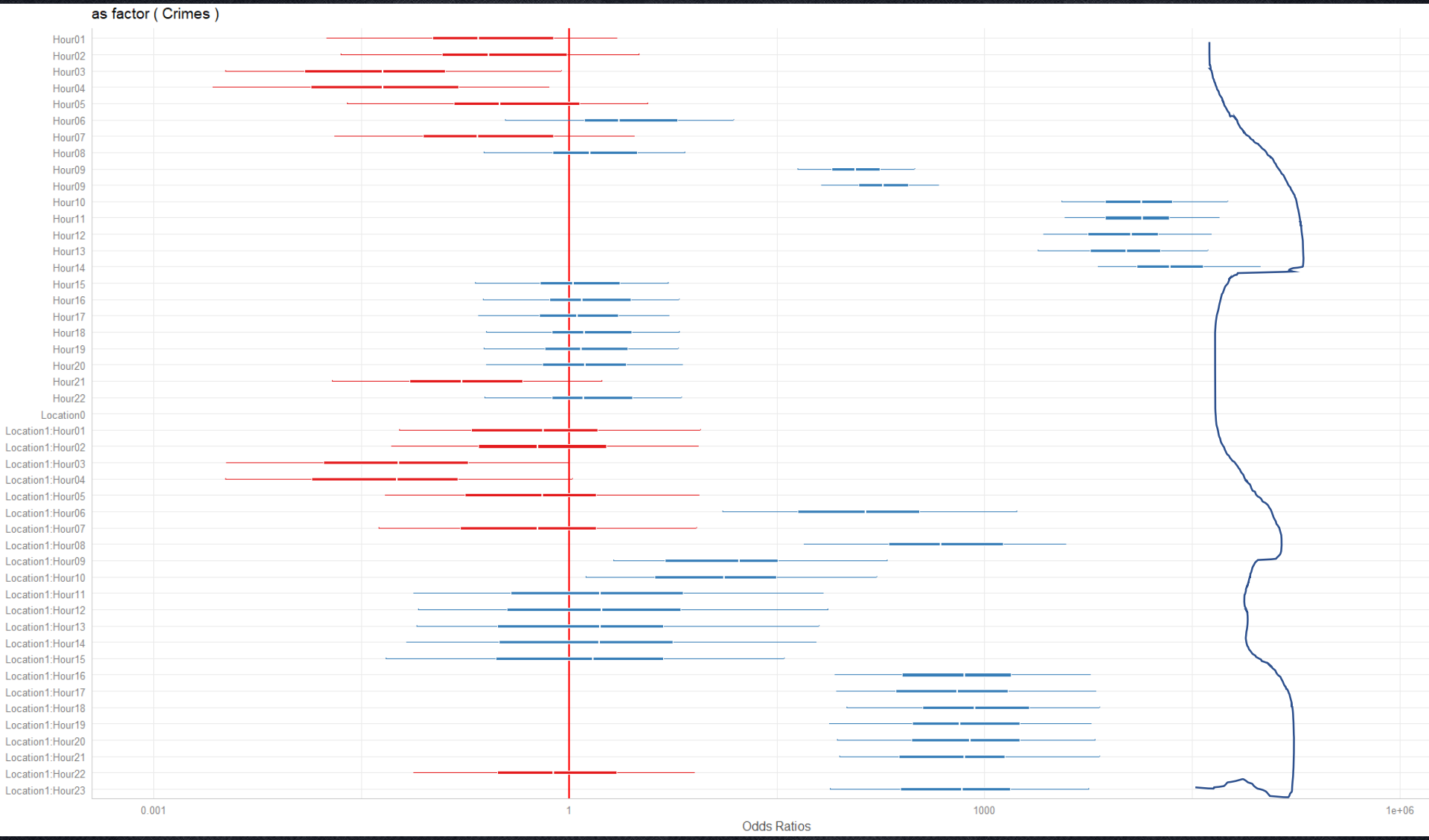
Crime ~ Hour + Location + Hour : Location+(1|Boro)

```
m_fake_loc_3 <- stan_glmer(as.factor(Crimes) ~ Location+Hour+Hour:Location+(1|BoroName),
data=fake_data_df_2, family=binomial(link="logit"))
```

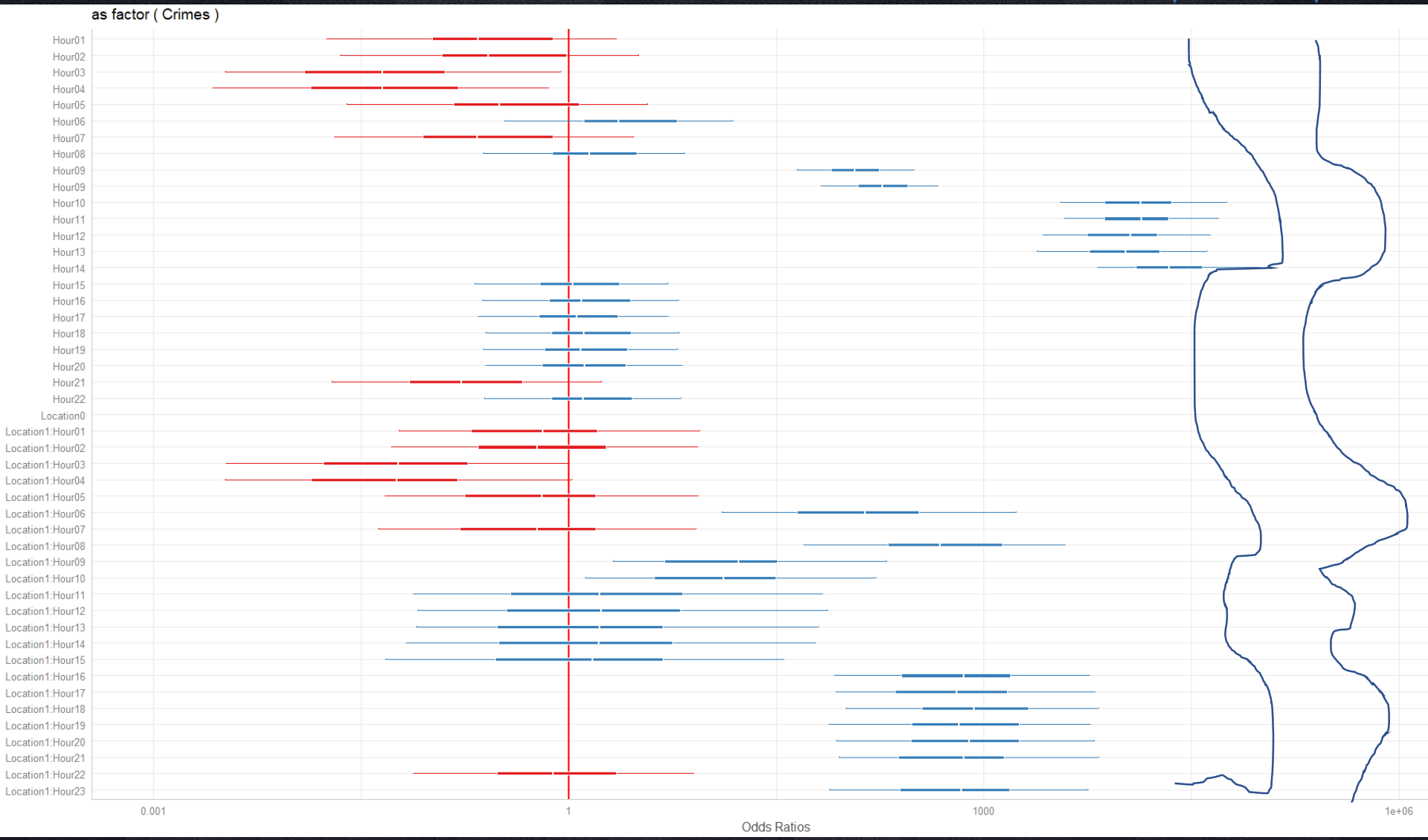
Mischief Managed



Mischief Managed



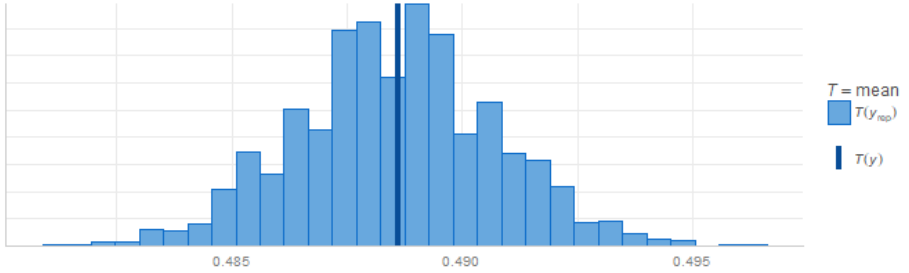
Mischief Managed

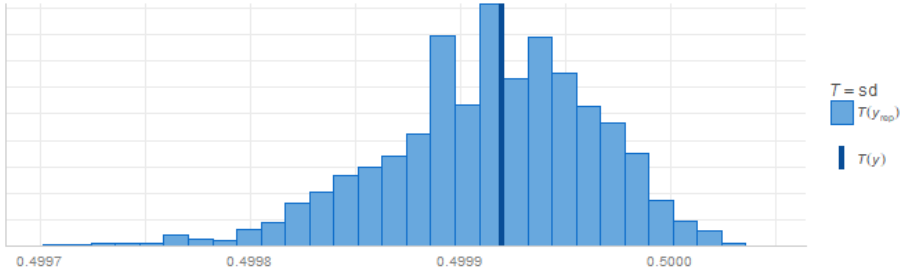


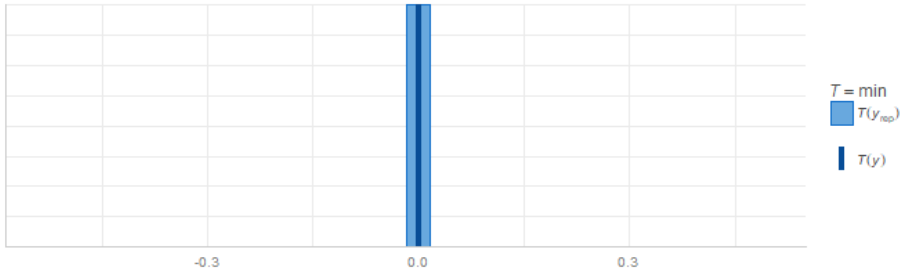
PP Checks – Fake Data

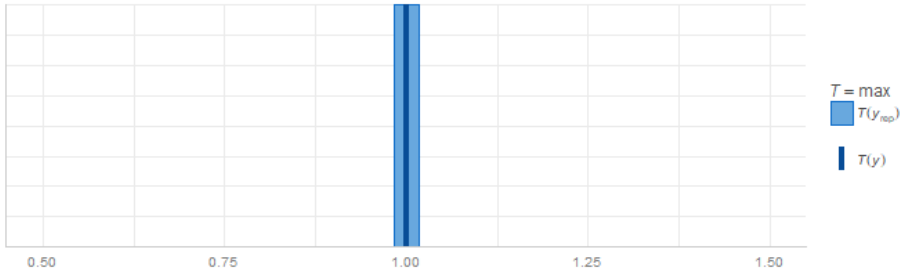
Distributions of test statistics $T(y^{rep})$

The blue lines show $T(y)$, the value of the statistic computed from the observed data.









Potential Next Steps

- Compare the model against subsets from other years
- Expand the model to include a third crime type?
- Investigate if the pattern of inside vs outside matters in other crimes?

Extra Slides

Data Description

- NYC Crime Data from 2006-2016
- 22 variables
  - 5 ~ Date/Time
  - 5 ~ Type of Crime
  - 6 ~ Info about Location
  - 4 ~ Exact Location

Cleaning

- Selected only for successful crimes
- Deleted unwanted variables

```
#########################################################################################################################################

#                                                    DATA CLEANING

#########################################################################################################################################


#select for Completed Crimes
as.data.frame(table(nyc$CRM_ATPT_CPTD_CD)) #find out just how many were attempted....negligible, ~90,000 out of 5.5million
unique(nyc$CRM_ATPT_CPTD_CD) #Completed, attempted, NA
nyc <- subset(nyc, CRM_ATPT_CPTD_CD=="COMPLETED")

# delete the following columns: CMPLNT_TO_DT, CMPLNT_TO_TM, RPT_DT, X_COORD_CD, Y_COORD_CD,
# and Lat_Lon

bad_vars <- names(nyc) %in% c("CRM_ATPT_CPTD_CD", "ADDR_PCT_CD", "CMPLNT_NUM", "CMPLNT_TO_DT", "CMPLNT_TO_TM", "RPT_DT", "X_COORD_CD", "Y_COORD_CD", "Lat_Lon")
nyc <- nyc[!bad_vars]
```

Cleaning

- Transformed PARK_NM, HADEVELOPT, and JURIS_DESC into indicator variables

```
##########################################################     INDICATOR VARIABLES    ########################################
#Reduce parks to binary
  #if it occurred in a park, doesn't matter which park, -> 1
  #if it did not occur in a park -> 0
nyc$PARKS_NM[is.na(nyc$PARKS_NM)] <- 0
nyc$PARKS_NM[nyc$PARKS_NM!=0] <- 1

#Reduce housing developments to binary
  #if it occurred in a housing development -> 1
  #if it did not -> 0
nyc$HADEVELOPT[is.na(nyc$HADEVELOPT)] <- 0
nyc$HADEVELOPT[nyc$HADEVELOPT!=0] <- 1

#Reduce jurisdiction to binary
  #if NY Police Department -> 1
  #if any other department -> 0
      # unique(nyc$JURIS_DESC)
      # [1] "1"                                "N.Y. HOUSING POLICE"              "N.Y. TRANSIT POLICE"
      # [4] "N.Y. STATE POLICE"                "DEPT OF CORRECTIONS"              "TRI-BORO BRDG TUNNL"
      # [7] "OTHER"                            "PORT AUTHORITY"                   "NYC PARKS"
      # [10] "HEALTH & HOSP CORP"               "METRO NORTH"                       "LONG ISLAND RAILRD"
      # [13] "STATN IS RAPID TRANS"             "N.Y. STATE PARKS"                  "U.S. PARK POLICE"
      # [16] "NEW YORK CITY SHERIFF OFFICE"     "NYS DEPT TAX AND FINANCE"          "AMTRACK"
      # [19] "CONRAIL"                          "POLICE DEPT NYC"                   "FIRE DEPT (FIRE MARSHAL)"
      # [22] "NYC DEPT ENVIRONMENTAL PROTECTION"  "SEA GATE POLICE DEPT"               "DISTRICT ATTORNEY OFFICE"
      # [25] "NYS DEPT ENVIRONMENTAL CONSERVATION"
nyc$JURIS_DESC[nyc$JURIS_DESC=='N.Y. POLICE DEPT'] <- 1
nyc$JURIS_DESC[nyc$JURIS_DESC!=1] <- 0
```

Cleaning

- Set LOC_OF_OCCUR_DESC to indicator
  - Inside <- 1, Outside <- 0
- Filled in missing info using logical rules based on PREM_DESC

```
na_premises <- unique(nyc$PREM_TYP_DESC[is.na(nyc$LOC_OF_OCCUR_DESC)])
# [1] "OTHER"                      "TRANSIT - NYC SUBWAY"        "RESIDENCE - APT. HOUSE"      "BUS STOP"
# [5] "GROCERY/BODEGA"             "TUNNEL"                      "RESIDENCE-HOUSE"             "BRIDGE"
# [9] "AIRPORT TERMINAL"           "PUBLIC BUILDING"             "FOOD SUPERMARKET"            "BUS (NYC TRANSIT)"
# [13] "OPEN AREAS (OPEN LOTS)"     "PARKING LOT/GARAGE (PUBLIC)" "PARKING LOT/GARAGE (PRIVATE)" "HIGHWAY/PARKWAY"
# [17] "DRY CLEANER/LAUNDRY"        "HOTEL/MOTEL"                 "CLOTHING/BOUTIQUE"           "STORAGE FACILITY"
# [21] "COMMERCIAL BUILDING"        "BAR/NIGHT CLUB"              "CONSTRUCTION SITE"           "FAST FOOD"
# [25] "BANK"                       "CHAIN STORE"                 "TAXI (LIVERY LICENSED)"      "HOSPITAL"
# [29] "SMALL MERCHANT"             "TAXI (YELLOW LICENSED)"      "TAXI/LIVERY (UNLICENSED)"    "TRANSIT FACILITY (OTHER)"
# [33] "BUS TERMINAL"               "PUBLIC SCHOOL"               "BUS (OTHER)"                 "RESTAURANT/DINER"
# [37] "BEAUTY & NAIL SALON"        "MARINA/PIER"                 NA                            "RESIDENCE - PUBLIC HOUSING"
# [41] "DEPARTMENT STORE"           "CANDY STORE"                 "TELECOMM. STORE"             "STORE UNCLASSIFIED"
# [45] "DRUG STORE"                 "GYM/FITNESS FACILITY"        "CHURCH"                      "BOOK/CARD"
# [49] "CHECK CASHING BUSINESS"     "ABANDONED BUILDING"          "SYNAGOGUE"                   "LIQUOR STORE"
# [53] "OTHER HOUSE OF WORSHIP"     "DOCTOR/DENTIST OFFICE"       "FACTORY/WAREHOUSE"           "ATM"
# [57] "PRIVATE/PAROCHIAL SCHOOL"   "CEMETERY"                    "JEWELRY"                     "SOCIAL CLUB/POLICY"
# [61] "VARIETY STORE"              "TRAMWAY"                     "FERRY/FERRY TERMINAL"        "PHOTO/COPY"
# [65] "VIDEO STORE"                "SHOE"                        "MOSQUE"                      "LOAN COMPANY"



#going to set the following as outside....
na_prem_outside <- na_premises[c(2,4,6,8,12:16,23,27,29:33,35,38,55,57,61,62)]
na_premises[c(2,4,6,8,12:16,23,27,29:33,35,38,55,57,61,62)]

nyc$LOC_OF_OCCUR_DESC[with(nyc, nyc$PREM_TYP_DESC %in% na_prem_outside & is.na(nyc$LOC_OF_OCCUR_DESC))] <- 0
sum(is.na(nyc$LOC_OF_OCCUR_DESC))
# #set these as the inside...
na_prem_inside <-  na_premises[c(3,5,7,9:11,17:22,24:26,28,34,36,37,39:54,56,58:60,63,64,66,67)]
na_prem_inside
nyc$LOC_OF_OCCUR_DESC[with(nyc, nyc$PREM_TYP_DESC %in% na_prem_inside & is.na(nyc$LOC_OF_OCCUR_DESC))] <- 1
```

Cleaning

- Parsed Dates and Times into multi-variable columns
- Selected for wanted years

```
#Time for Time
#want to bin time into 1-hour segments
#divide Hour:Minute:Second into three separate columns
nyc <- separate(nyc, CMPLNT_FR_TM, sep= ":", into=c("Hour", "Minute", "Second"), fill='right', remove=FALSE)

#separate date, similarly, into Month, Day, Year
nyc <- separate(nyc, CMPLNT_FR_DT, sep= "/", into=c("Month", "Day", "Year"), fill='right', remove=FALSE)

#DATA DELETION
#this database was supposed to be 2006-2016, but there are years here from 1905 and 1015 (prob a typo). Gonna delete the few thousand from before 2006
#2005 is also a little skewed, though. Even though it has 10,000+ events, all the other years have nearly half a million data points.
yearsIWant <- c("2006","2007","2008","2009","2010","2011","2012","2013","2014","2015","2016")
nyc <- subset(nyc, nyc$Year %in% yearsIWant)
```

Cleaning

```
> as.data.frame(table(nyc$Year))
   Var1   Freq
1  1015      9
2  1016     15
3  1026      5
4  1900      5
5  1905      2
6  1906      1
7  1908      3
8  1909      3
9  1910      9
10 1911      7
11 1912     10
12 1913      9
13 1914     11
14 1915      8
15 1916      6
16 1919      1
17 1920      6
18 1922      1
19 1929      1
20 1930      1
21 1938      1
22 1940      1
23 1941      2
24 1942      2
25 1945      2
26 1946      2
27 1948      1
28 1950      3
29 1954      2
30 1955      3
31 1956      1
32 1958      1
33 1959      2
34 1960     10
35 1961      1
36 1962      3
37 1964      1
38 1965      5
39 1966     27
40 1967     13
41 1968     10
42 1969      7
43 1970      6
44 1971      3
45 1972      6
46 1973      7
47 1974      9
48 1975      8
49 1976      5
50 1977      9
51 1978      6
52 1979      9
53 1980     14
54 1981      8
55 1982      7
56 1983      6
57 1984      8
58 1985     21
59 1986     30
60 1987     16
61 1988     17
62 1989     25
63 1990     35
64 1991     29
65 1992     45
66 1993     46
67 1994     64
68 1995     75
69 1996    122
70 1997    134
71 1998    224
72 1999    342
73 2000    908
74 2001   1008
75 2002   1047
76 2003   1547
77 2004   2116
78 2005  10797
79 2006 539084
80 2007 537242
81 2008 528744
82 2009 511014
83 2010 509853
84 2011 498381
85 2012 504334
86 2013 495304
87 2014 491131
88 2015 477031
89 2016 468290
```

Cleaning

- Found out the day of the week for each date -> created a Weekend indicator variable
- Made lists of major holidays -> created a Holiday indicator variable

```r
#deal with date...convert to standard format
nyc$Date <- as.Date(nyc$CMPLNT_FR_DT, "%m/%d/%Y")

#find out day of week
nyc$DayName <-  weekdays(as.Date(nyc$Date))

#find out weekday or weekend
daysoftheweek <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
daysoftheweekend <- c("Saturday", "Sunday")

#create indicator variable, where 0 is a weekday and 1 is a weekend.
nyc$Weekend <- as.integer(nyc$DayName %in% daysoftheweekend)

#some holidays are always the same day
holidays_list <- c("01-01", "02-14", "07-04", "09-04", "10-31", "12-25" )
easter_list <- c("2006-04-23", "2007-04-08", "2008-03-27", "2009-04-19", "2010-04-04", "2011-04-24", "2012-04-15", "2013-05-05", "2014-04-20", "2015-04-12")
as.data.frame(table(nyc$Year))
thanksgiving_list <- c("2005-11-24", "2006-11-23", "2007-11-22", "2008-11-27", "2009-11-26", "2010-11-25", "2011-11-24", "2012-11-22", "2013-11-28", "2014-11-27", "2

#create Holiday indicator variable, 0 if not holiday, 1 if it matches any of the holidays specified above
nyc$Holiday <- as.integer(nyc$MonthDay %in% holidays_list, nyc$CMPLNT_FR_DT %in% easter_list, nyc$CMPLNT_FR_DT %in% thanksgiving_list)
    # > as.data.frame(table(nyc$Holiday))
    # Var1    Freq
    # 1    0 5375860
    # 2    1   93413
```

Cleaning

```
unique(nyc$OFNS_DESC) #...71 different classifiers
unique(nyc$PD_DESC) #...410 different classifiers
unique(nyc$LAW_CAT_CD) #...3 different classifiers

#well, 71 is a lot better than 410...
#I'm not sure there's anything between 3 and 71 without loosing a lot of data. 71 will have to do.
# so I guess I'm looking at a hierarchical (between boros) multinomial (unordered categorial crime type) model?
```

Cleaning

- Imputed missing response variables

```r
#oh look...NAs. Yay.
# sum(is.na(nyc$OFNS_DESC))
# as.data.frame(table(nyc$PD_DESC[is.na(nyc$OFNS_DESC)]))
#for the 56 cases where the OFNS_DESC is NA but the PD_DESC is not, I could substitute the PD_DESC for the OFNS_DESC
#...or I could go through and find previous incidences where the PD_DESC is the same OFNS_DESC is not NA, and substitute that OFNS_DESC for the NA
#probably the better way to go....but probs a lot more involved. Ugh.

#solution....blimey this takes forever. Just FYI.
#Feel free to delete the message(i) lines if you don't want to see a bazilion numbers on your screen
for (i in 1:nrow(nyc)){
  if (is.na(nyc$OFNS_DESC[i])){
    crimetype=nyc$PD_DESC[i]
    othercrimetypes=unique(nyc$OFNS_DESC[nyc$PD_DESC==crimetype])
    if (length(othercrimetypes)==2){
      nyc$OFNS_DESC[i] <- othercrimetypes[2]
      message(i)
    }
    else if (is.na(othercrimetypes)){
      nyc$OFNS_DESC[i] <- nyc$PD_DESC[i]
      message(i)
    }
  }
}
```

Cleaning

```r
#Last Step
#reorder the columns to put similar variables together
#don't care about Minute or Second, so not including those
#nixing PD_DESC since I'm going to use OFNS_DESC as my response variable
nyc_clean <- nyc[,c("CMPLNT_FR_DT", "Date", "Month", "Day", "Year", "MonthDay",
      "Holiday", "DayName", "Weekend", "CMPLNT_FR_TM", "Hour", "OFNS_DESC", "LAW_CAT_CD",
      "JURIS_DESC", "BORO_NM", "LOC_OF_OCCUR_DESC", "PARKS_NM", "HADEVELOPT")]


# sum(is.na(nyc_clean$LAW_CAT_CD))
#save cleaned data so I don't have to do all this again
write.csv(nyc_clean, "NYPD_Crime_Data_CLEAN.csv")
```

Cleaning

- Reduced OFNS_DESC from 72 classifiers to 26
  - Deleted crimes committed at extremely-low frequencies
  - Combined crimes of similar natures

```
#get rid of boring crimes
useless_crimes <- c("ABORTION", "AGRICULTURE & MRKTS LAW-UNCLASSIFIED", "ALCOHOLIC BEVERAGE CONTROL LAW",
"ANTICIPATORY OFFENSES", "CHILD ABANDONMENT/NON SUPPORT", "DISORDERLY CONDUCT", "DISRUPTION OF A RELIGIOUS
SERV", "ENDAN WELFARE INCOMP", "ESCAPE 3", "FORTUNE TELLING", "GAMBLING", "JOSTLING", "NEW YORK CITY HEALTH
CODE", "NYS LAWS-UNCLASSIFIED FELONY", "NYS LAWS-UNCLASSIFIED VIOLATION", "OTHER STATE LAWS", "OTHER STATE LAWS
(NON PENAL LA", "OTHER STATE LAWS (NON PENAL LAW)", "OTHER TRAFFIC INFRACTION", "PROSTITUTION & RELATED
OFFENSES", "THEFT,RELATED OFFENSES,UNCLASS", "UNDER THE INFLUENCE OF DRUGS", "UNLAWFUL POSS. WEAP. ON SCHOOL")

nyc_2 <- subset(nyc_clean, !nyc_clean$OFNS_DESC %in% useless_crimes)
```

Cleaning

- Reduced OFNS_DESC from 72 classifiers to 26
  - Deleted crimes committed at extremely-low frequencies
  - Combined crimes of similar natures

```
var <- "OFNS_DESC"

cd_old <- c("UNAUTHORIZED USE OF A VEHICLE", "VEHICLE AND TRAFFIC LAWS")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old, "VEHICLE/TRAFFIC LAWS RELATED",x))

cd_old_2 <- c("HOMICIDE-NEGLIGENT,UNCLASSIFIE", "HOMICIDE-NEGLIGENT-VEHICLE", "MURDER & NON-NEGL. MANSLAUGHTER"
)
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_2, "MURDER",x))

cd_old_3 <- c("OFFENSES AGAINST PUBLIC ADMINI", "OFF. AGNST PUB ORD SENSBLTY &", "OFFENSES AGAINST MARRIAGE
UNCL", "OFFENSES AGAINST PUBLIC SAFETY")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_3, "MISC. OFFENSES",x))

cd_old_4 <- c("RAPE", "SEX CRIMES")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_4, "RAPE OR SEX CRIME",x))

cd_old_5 <- c("ADMINISTRATIVE CODES", "ADMINISTRATIVE CODES")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_5, "MISCELLANEOUS PENAL LAW",x))

cd_old_6 <- c("BURGLAR'S TOOLS", "BURGLARY")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_6, "BURGLARY RELATED",x))

cd_old_7 <- c("FRAUDS", "FRAUDULENT ACCOSTING", "OFFENSES INVOLVING FRAUD")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_7, "FRAUD RELATED",x))

cd_old_8 <- c("GRAND LARCENY", "PETIT LARCENY")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_8, "GRAND/PETIT LARCENY",x))

cd_old_9 <- c("GRAND LARCENY OF MOTOR VEHICLE", "PETIT LARCENY OF MOTOR VEHICLE")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_9, "VEHICULAR GRAND/PETIT LARCENY",x))

cd_old_10 <- c("INTOXICATED & IMPAIRED DRIVING", "INTOXICATED/IMPAIRED DRIVING")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_10, "DUI",x))

cd_old_11 <- c("KIDNAPPING", "KIDNAPPING AND RELATED OFFENSES", "KIDNAPPING & RELATED OFFENSES")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_11, "KIDNAPPING RELATED",x))

cd_old_12 <- c("LOITERING", "LOITERING FOR DRUG PURPOSES", "LOITERING FOR PROSTITUTION OR", "LOITERING/DEVIATE
SEX", "LOITERING/GAMBLING (CARDS, DIC")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_12, "LOITERING RELATED",x))

cd_old_13 <- c("OFFENSES AGAINST THE PERSON", "OFFENSES RELATED TO CHILDREN")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_13, "OFFENSES AGAINST HUMANS",x))

cd_old_14 <- c("OTHER OFFENSES RELATED TO THEF", "THEFT-FRAUD", "THEFT OF SERVICES")
nyc_2[,var] <- sapply(nyc_2[,var],function(x) ifelse(x %in% cd_old_14, "THEFT RELATED",x))
```

Fake it ‘til you make it

- Fill in categorical variables based on binary dummy variables

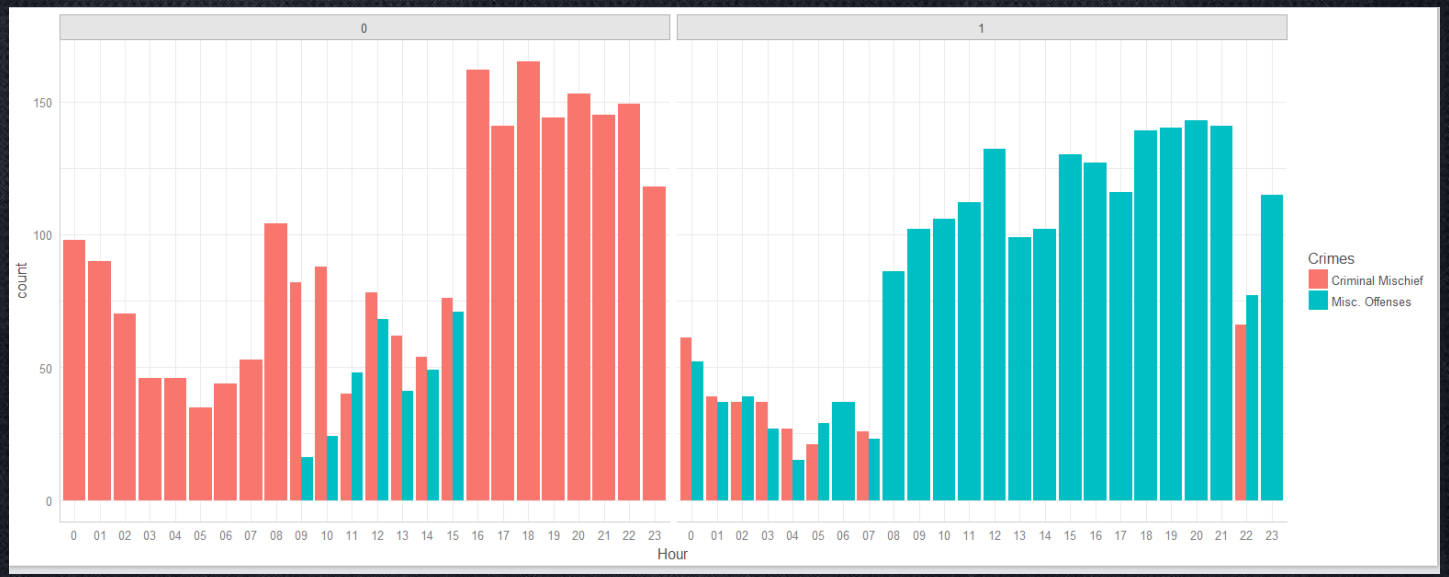
```
for (i in 1:nsims){
  if (fake_data3[[i,"Queens"]]==1){
    fake_data3[[i,"BoroName"]] <- "Queens"
  }
  else if (fake_data3[[i,"Bronx"]]==1){
    fake_data3[[i,"BoroName"]] <- "Bronx"
  }
  else if (fake_data3[[i,"Brooklyn"]]==1){
    fake_data3[[i,"BoroName"]] <- "Brooklyn"
  }
  else if (fake_data3[[i,"Manhattan"]]==1){
    fake_data3[[i,"BoroName"]] <- "Manhattan"
  }
  else if (fake_data3[[i,"StatenIsland"]]==1){
    fake_data3[[i,"BoroName"]] <- "StatenIsland"
  }

  if(fake_data3[[i,"H1"]]==1){
    fake_data3[[i, "Hour"]] <- "01"
  }
  else if(fake_data3[[i,"H2"]]==1){
    fake_data3[[i, "Hour"]] <- "02"
  }
  else if(fake_data3[[i,"H3"]]==1){
    fake_data3[[i, "Hour"]] <- "03"
  }
  else if(fake_data3[[i,"H4"]]==1){
    fake_data3[[i, "Hour"]] <- "04"
  }
  else if(fake_data3[[i,"H5"]]==1){
    fake_data3[[i, "Hour"]] <- "05"
  }
  else if(fake_data3[[i,"H6"]]==1){
    fake_data3[[i, "Hour"]] <- "06"
  }
  else if(fake_data3[[i,"H7"]]==1){
    fake_data3[[i, "Hour"]] <- "07"
  }
  else if(fake_data3[[i,"H8"]]==1){
    fake_data3[[i, "Hour"]] <- "08"
  }
  else if(fake_data3[[i,"H9"]]==1){
    fake_data3[[i, "Hour"]] <- "09"
  }
  else if(fake_data3[[i,"H10"]]==1){
    fake_data3[[i, "Hour"]] <- "10"
  }
  else if(fake_data3[[i,"H11"]]==1){
    fake_data3[[i, "Hour"]] <- "11"
  }
  else if(fake_data3[[i,"H12"]]==1){
    fake_data3[[i, "Hour"]] <- "12"
  }
  else if(fake_data3[[i,"H13"]]==1){
    fake_data3[[i, "Hour"]] <- "13"
  }
  else if(fake_data3[[i,"H14"]]==1){
    fake_data3[[i, "Hour"]] <- "14"
  }
  else if(fake_data3[[i,"H15"]]==1){
    fake_data3[[i, "Hour"]] <- "15"
  }
  else if(fake_data3[[i,"H16"]]==1){
    fake_data3[[i, "Hour"]] <- "16"
  }
  else if(fake_data3[[i,"H17"]]==1){
    fake_data3[[i, "Hour"]] <- "17"
  }
  else if(fake_data3[[i,"H18"]]==1){
    fake_data3[[i, "Hour"]] <- "18"
  }
  else if(fake_data3[[i,"H19"]]==1){
    fake_data3[[i, "Hour"]] <- "19"
  }
  else if(fake_data3[[i,"H20"]]==1){
    fake_data3[[i, "Hour"]] <- "20"
  }
  else if(fake_data3[[i,"H21"]]==1){
    fake_data3[[i, "Hour"]] <- "21"
  }
 else if(fake_data3[[i,"H22"]]==1){
    fake_data3[[i, "Hour"]] <- "22"
  }
  else if(fake_data3[[i,"H23"]]==1){
    fake_data3[[i, "Hour"]] <- "23"
  }
  else if(any(as.numeric(fake_data3[i,8:30])==1)){
  fake_data3[[i, "Hour"]] <- "00"
  }
}
```

Close Enough?

- Create some coefficients, taken from real data model

REAL

FAKE



NYC Crime Project

Zane Wolf

Project Aim, v2.

Main Question:

Given a time and information about the location, can I predict what type of crime is most likely occurring?

Given a time and information about the location, can I which of the 26 crime types I had defined were more likely?