Report

Pedro Belém, Rui Fonseca, Tiago Botelho

December 23, 2016

Data Pre processing

The data is read from the crime.xls file into a data frame

```
data_path <- "./crime.xls"
info <- read.xls(data_path, sheet=1)
```

Next we remove outliers by unix time days

```
remove_outliers <- function(info) {
  n_days <- day(days(ymd(info$Date)))
  return(info[n_days >= quantile(n_days, .25) - 1.5*IQR(n_days) & n_days <= quantile(n_days, .75) + 1.5
}
```

Then, the unknown values (NA) are treated

```
na_handler <- function(info) {
  info$Beat[info$Beat == 'UNK'] <- NA
  info <- knnImputation(info, k=3)
  info$BlockRange[info$BlockRange=='UNK'] <- NA
  info$Type[info$Type == '-'] <- NA
  info$Suffix[info$Suffix == '-'] <- NA
  info$Beat <- as.character(info$Beat)

  return(info)
}
```

Tthe block range variable is modified. Since it is always a string like "X-Y", with X being a multiple of 100 and Y=X+99, we keep only X/100

```
split <- strsplit(as.character(info$BlockRange), "-")
info$BlockRange <- order(sapply(split, "[", 1))
```

We will also create a data frame that contains for each crime:

- WeekDay: the day of the week when the crime happene
- Beat: the police beat
- DayInterval: the interval of the day in which the crime happened, following this criteria:
    1. "Mourning" represented as "1", is from the hour interval 8 >= h > 12
    2. "Afternoon" represented as "2", is from the hour interval 12 >= h > 19
    3. "Night"represented as "3", is from the hour interval 19 >= h <= 23 of the same day, plus the hour interval 0 >= h > 8 of the following day
- Year: the year in which the crime happened
- Month: the month in which the crime occoured

- Day: the day of the month in which the crime happened

Create the data frame with the new collumns

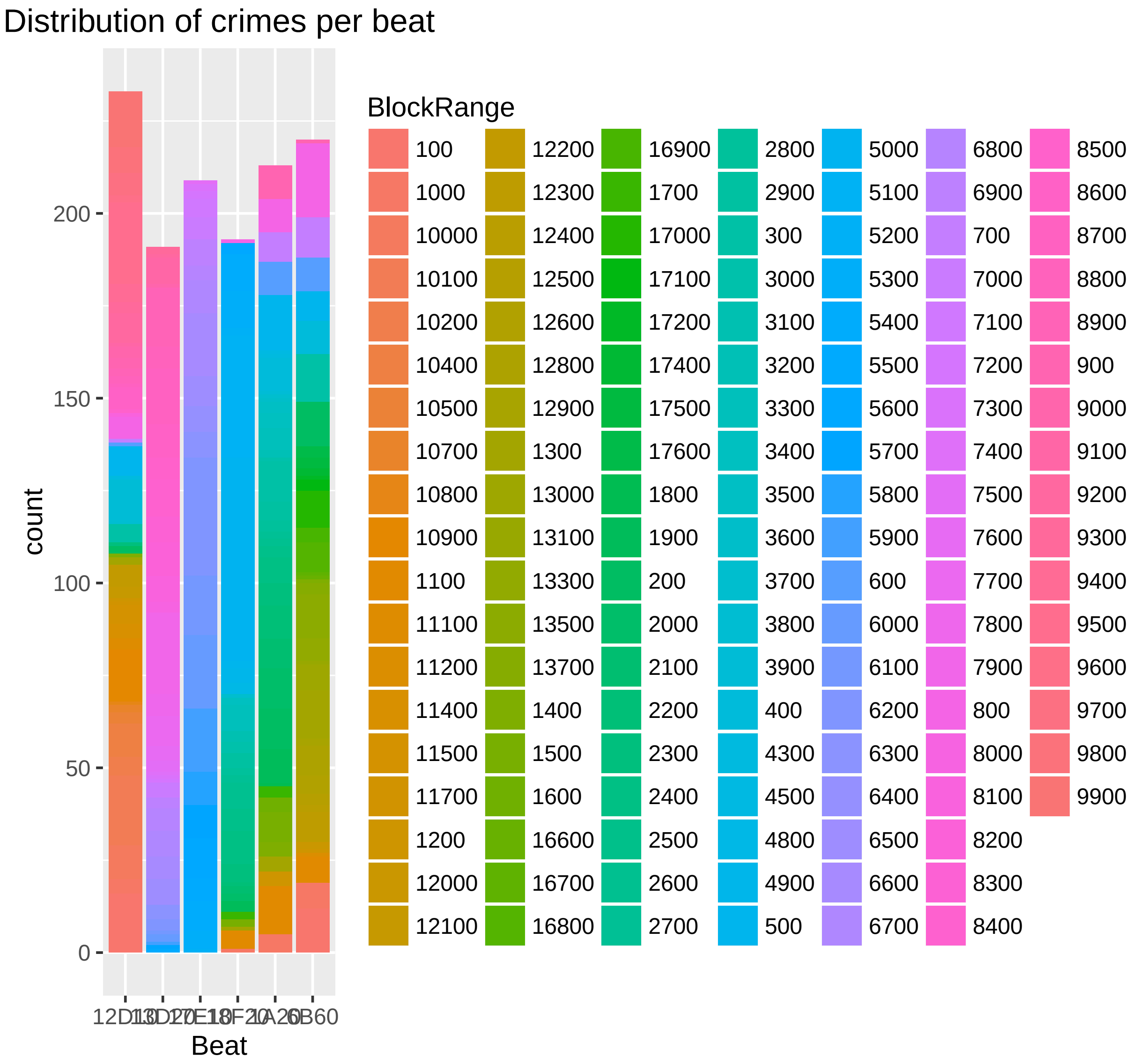
```
dataset_prep <- function(x, only.week=FALSE) {
  x$Date <- ifelse(as.integer(x$Hour) < 8, as.character(as.Date(x$Date) - 1), as.character(x$Date))

  # Split info in time intervals
  x$DayInterval <- 0
  x[as.integer(x$Hour) < 8 | as.integer(x$Hour) >= 19,]$DayInterval <- 3
  x[as.integer(x$Hour) >= 12 & as.integer(x$Hour) < 19,]$DayInterval <- 2
  x[as.integer(x$Hour) >= 8 & as.integer(x$Hour) < 12,]$DayInterval <- 1

  if(only.week){
    return(data.frame(WeekDay = as.integer(strftime(x$Date, "%u")),
                      DayInterval = x$DayInterval,
                      Beat = x$Beat,
                      Offenses = x$X..offenses,
                      stringsAsFactors = FALSE))
  }

  return(data.frame(WeekDay = as.integer(strftime(x$Date, "%u")),
                    DayInterval = x$DayInterval,
                    Beat = x$Beat,
                    Offenses = x$X..offenses,
                    Day = day(x$Date),
                    Month = month(x$Date),
                    Year = year(x$Date),
                    stringsAsFactors = FALSE))
}
```

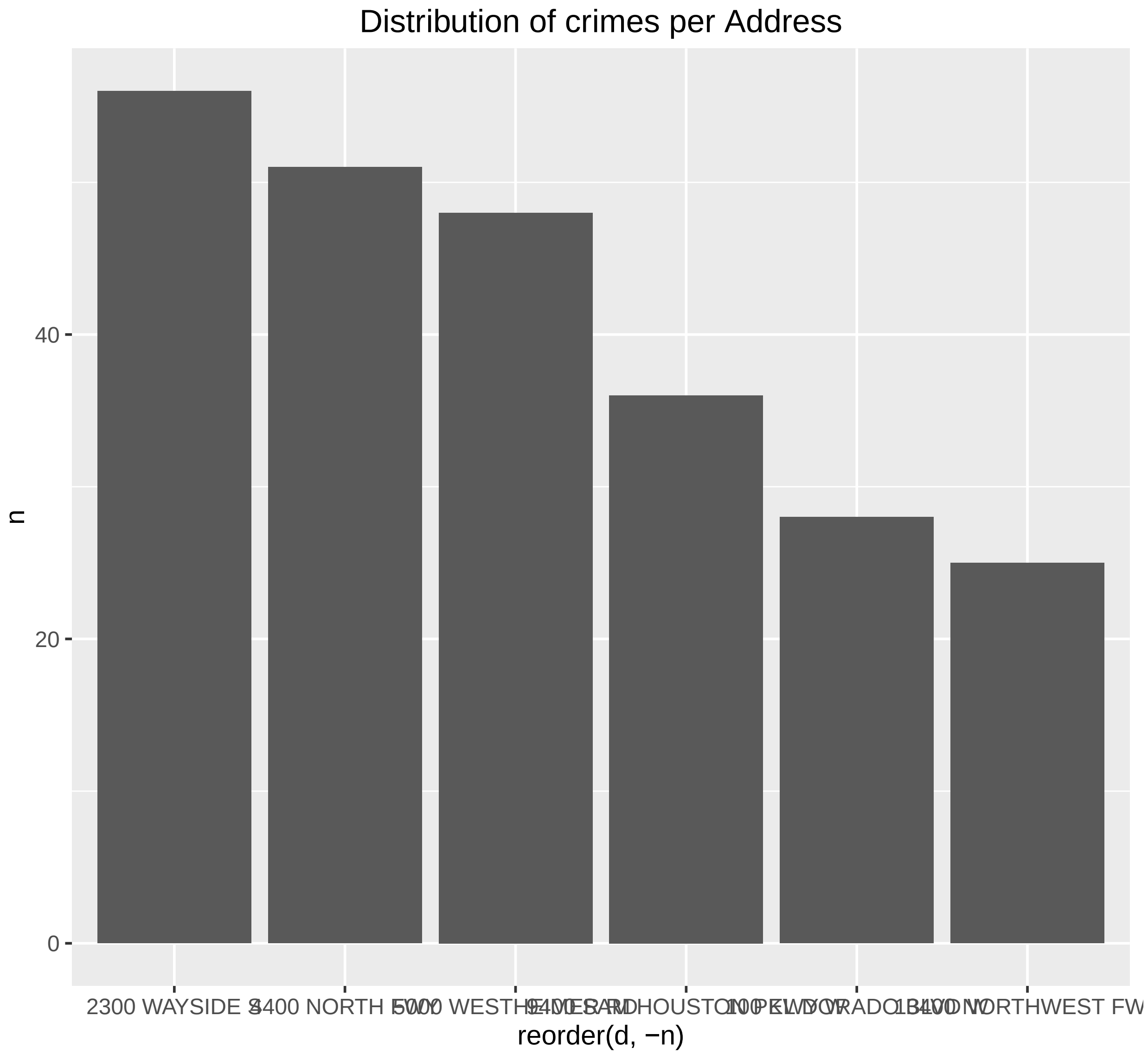
Create all permutations of missing

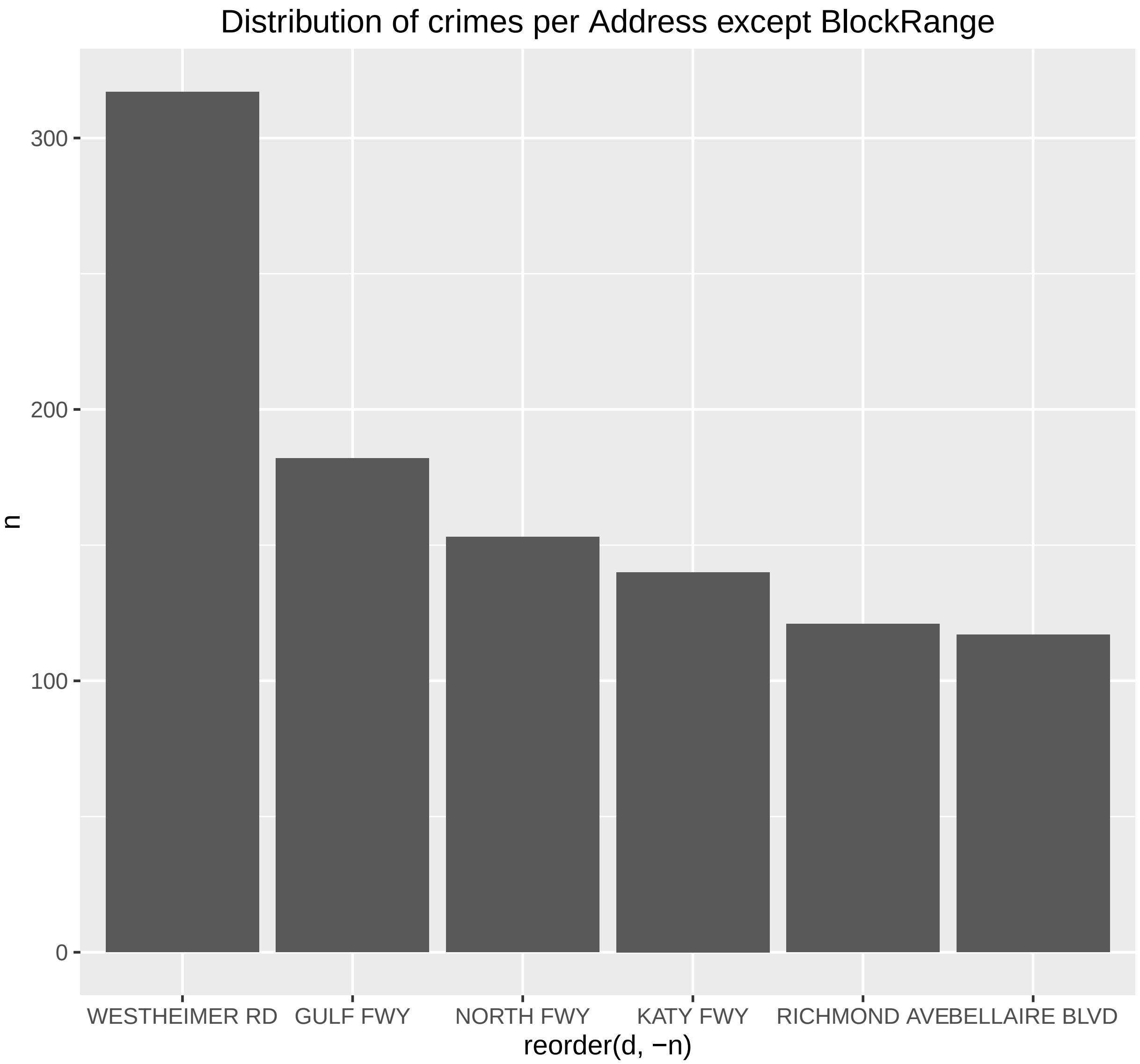
```
create_total_perm <- function(preprocessed, only.week = FALSE) {
  days.between <- get_days_between(info)
  unique_beats <- unique(preprocessed$Beat)
  unique_day_intervals <- unique(preprocessed$DayInterval)

  if(only.week){
    unique_weekdays <- unique(preprocessed$WeekDay)
    all_beats_perm <- data.frame(WeekDay = rep(unique_weekdays, times=length(unique_beats) * length(uni
                                 DayInterval = rep(unique_day_intervals, times=length(unique_beats) * le
                                 Beat = rep(unique_beats, times=length(unique_day_intervals) * length(da
                                 Offenses = rep(0, times=length(unique_day_intervals) * length(days.betw
  } else{
    all_beats_perm <- data.frame(Date = rep(days.between, times=length(unique_beats) * length(unique_day
                                 DayInterval = rep(unique_day_intervals, times=length(unique_beats) * le
                                 Beat = rep(unique_beats, times=length(unique_day_intervals) * length(da
                                 Offenses = rep(0, times=length(unique_day_intervals) * length(days.betw
    all_beats_perm$DayInterval <- as.integer(all_beats_perm$DayInterval)
    all_beats_perm$Beat <- as.numeric(all_beats_perm$Beat)
    all_beats_perm$Offenses <- as.character(all_beats_perm$Offenses)
    all_beats_perm$WeekDay = as.integer(strftime(all_beats_perm$Date, "%u"))
    all_beats_perm$Day <- day(as.Date(all_beats_perm$Date))
    all_beats_perm$Month <- month(as.Date(all_beats_perm$Date))
    all_beats_perm$Year <- year(as.Date(all_beats_perm$Date))
    all_beats_perm <- all_beats_perm[,!colnames(all_beats_perm) %in% c("Date")]
  }

  return(all_beats_perm)
}
```

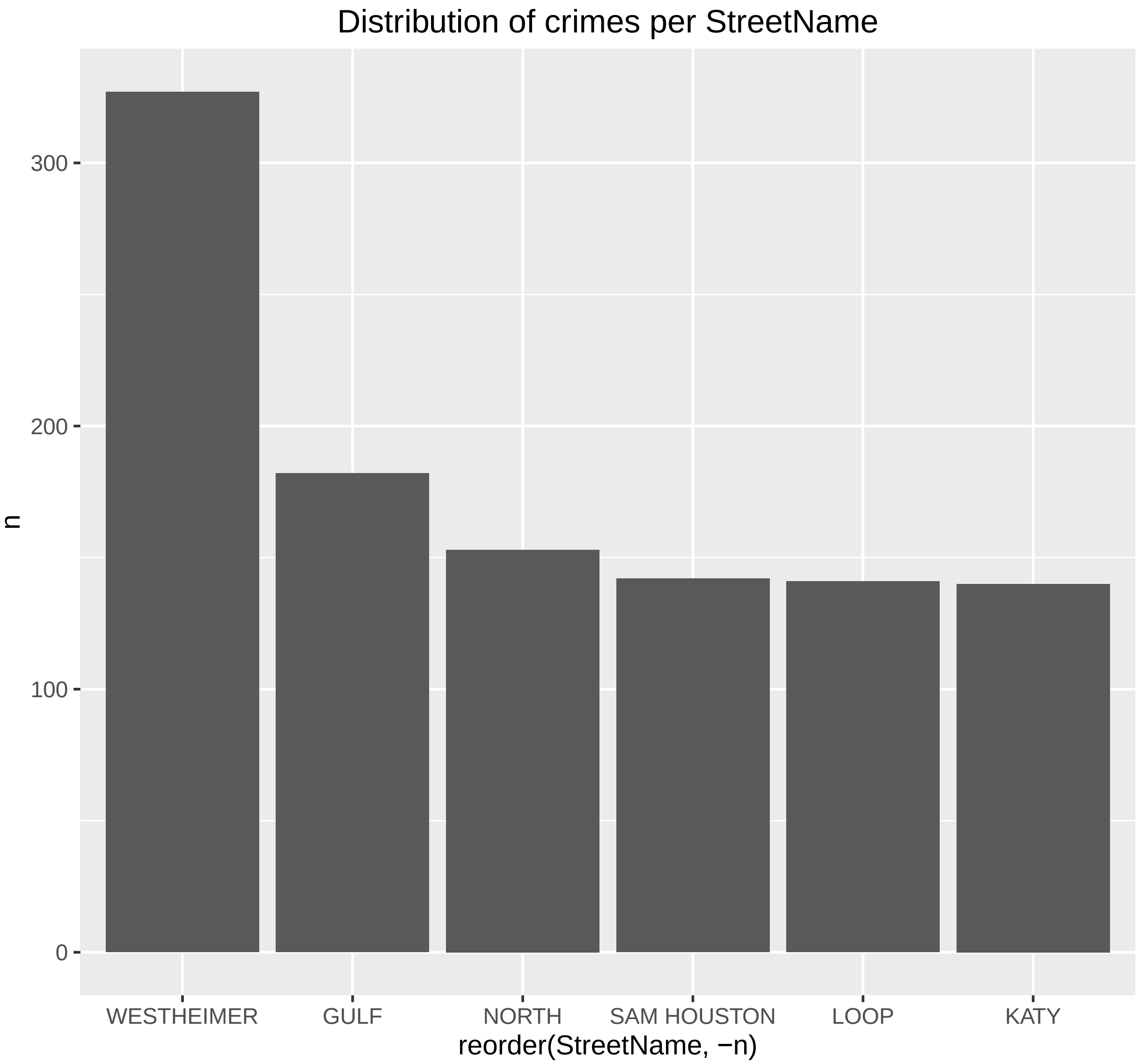
Data Visualisation

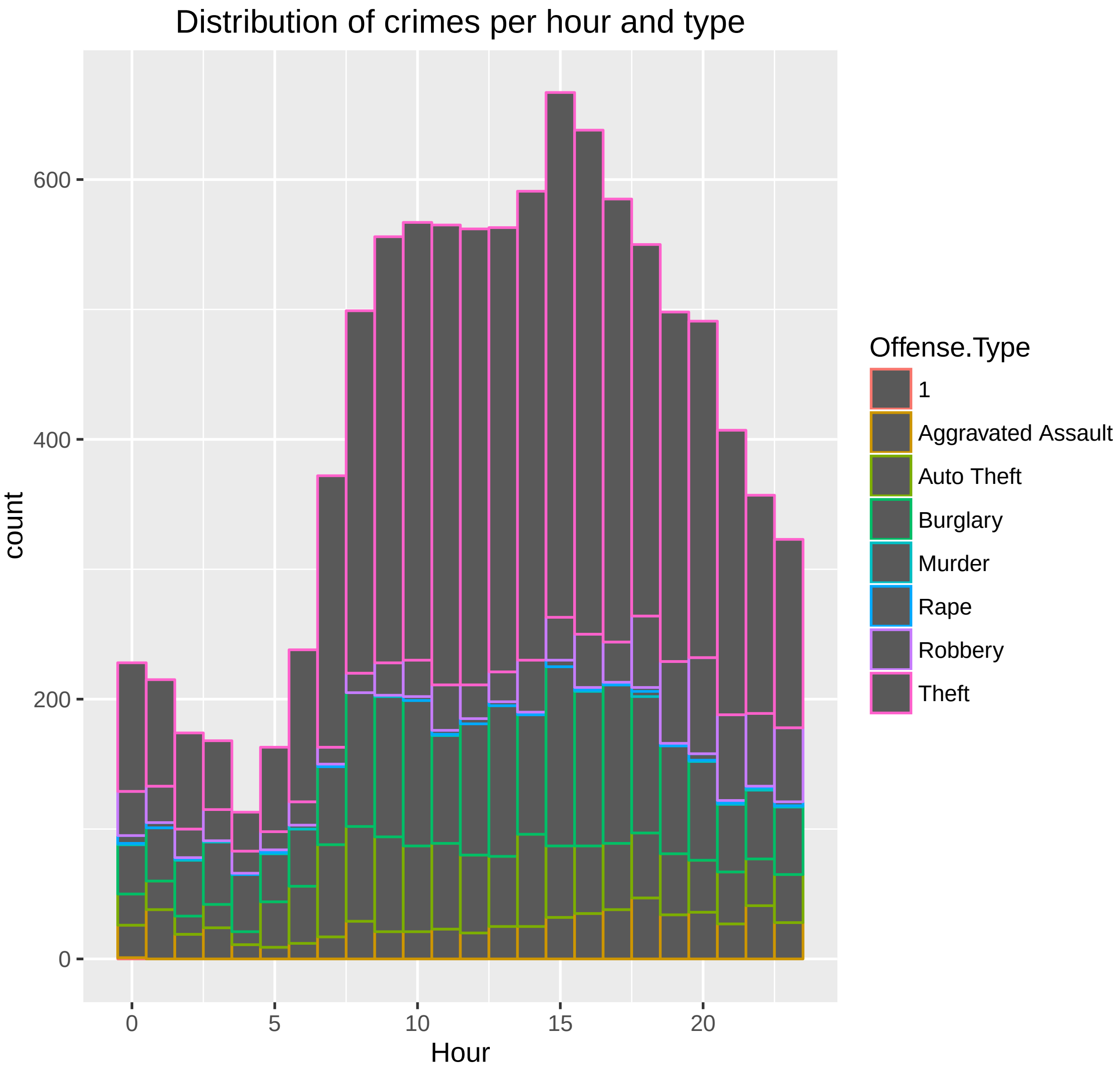
Creating graphics with the data, we can begin to find some usefull patterns in the data.



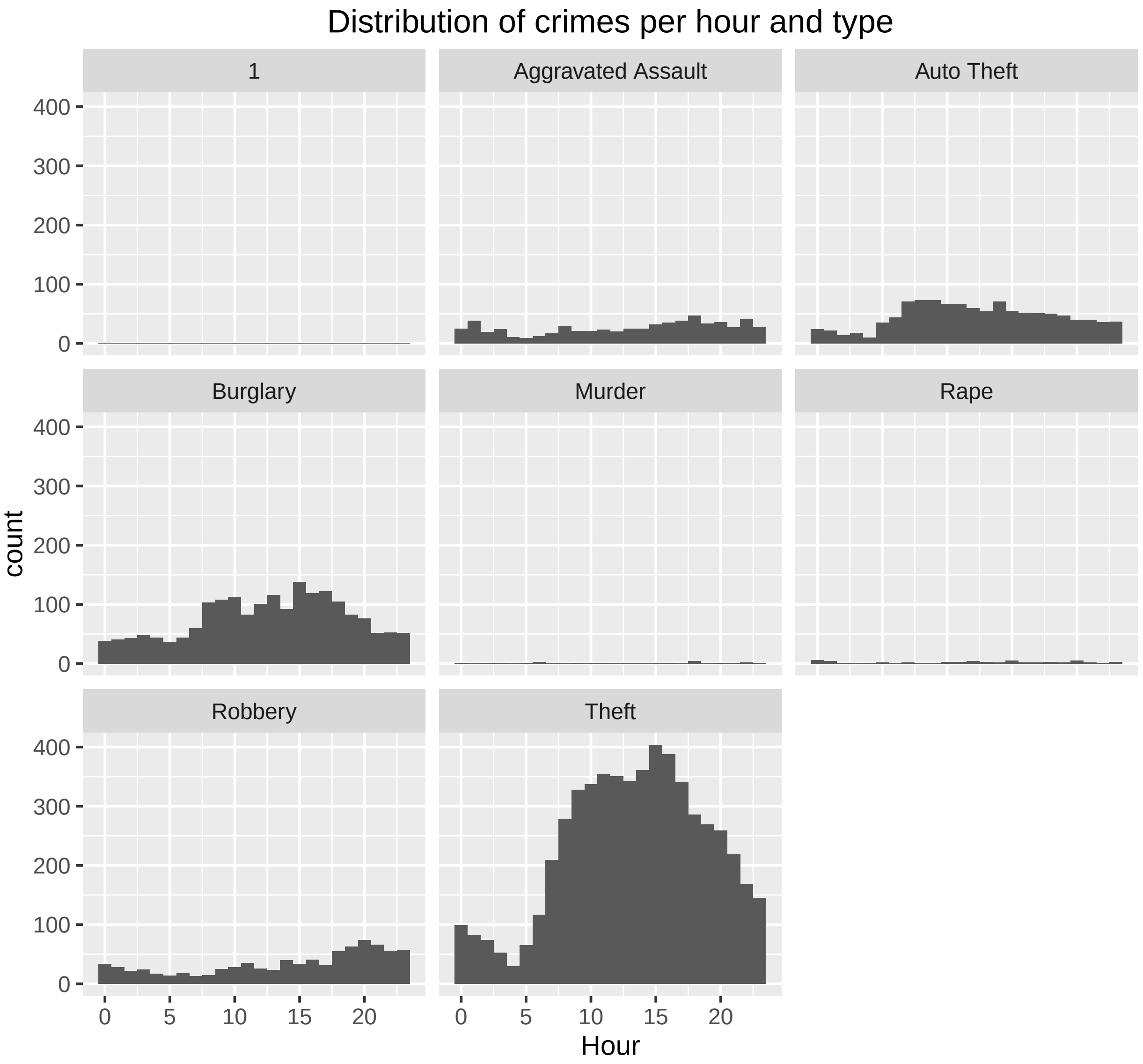

Distribution of crimes per Address
n
40
20
0
2300 WAYSIDE S
8400 NORTH FWY
5000 WESTHEIMER RD
9400 SAM HOUSTON PKWY
100 EL DORADO BLVD
13400 NORTHWEST FW
reorder(d, −n)


Distribution of crimes per Address except BlockRange
n
300
200
100
0
WESTHEIMER RD
GULF FWY
NORTH FWY
KATY FWY
RICHMOND AVE
BELLAIRE BLVD
reorder(d, −n)


Distribution of crimes per StreetName
n
300
200
100
0
WESTHEIMER
GULF
NORTH
SAM HOUSTON
LOOP
KATY
reorder(StreetName, −n)



```
#number of crimes per hour and type
ggplot(info, aes(x=Hour)) + geom_histogram(binwidth = 1) + facet_wrap(~ Offense.Type) + ggtitle("Distril
```



Data Prediction

The question that we're going to answer is: since we know all these crimes that happen, how many offenses will occur, in a given day interval of a day, in a certain police beat.