Report

Pedro Belém, Rui Fonseca, Tiago Botelho

December 23, 2016

Data Pre processing

We begin by reading the data from the crime.xls file into a data frame

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

Then we removed all the instances with outlier dates. To calculate the outliers we considered a date as the number of days since the beginning of unix time.

```
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, we convert all the missing values to NA.

```
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))
}
```

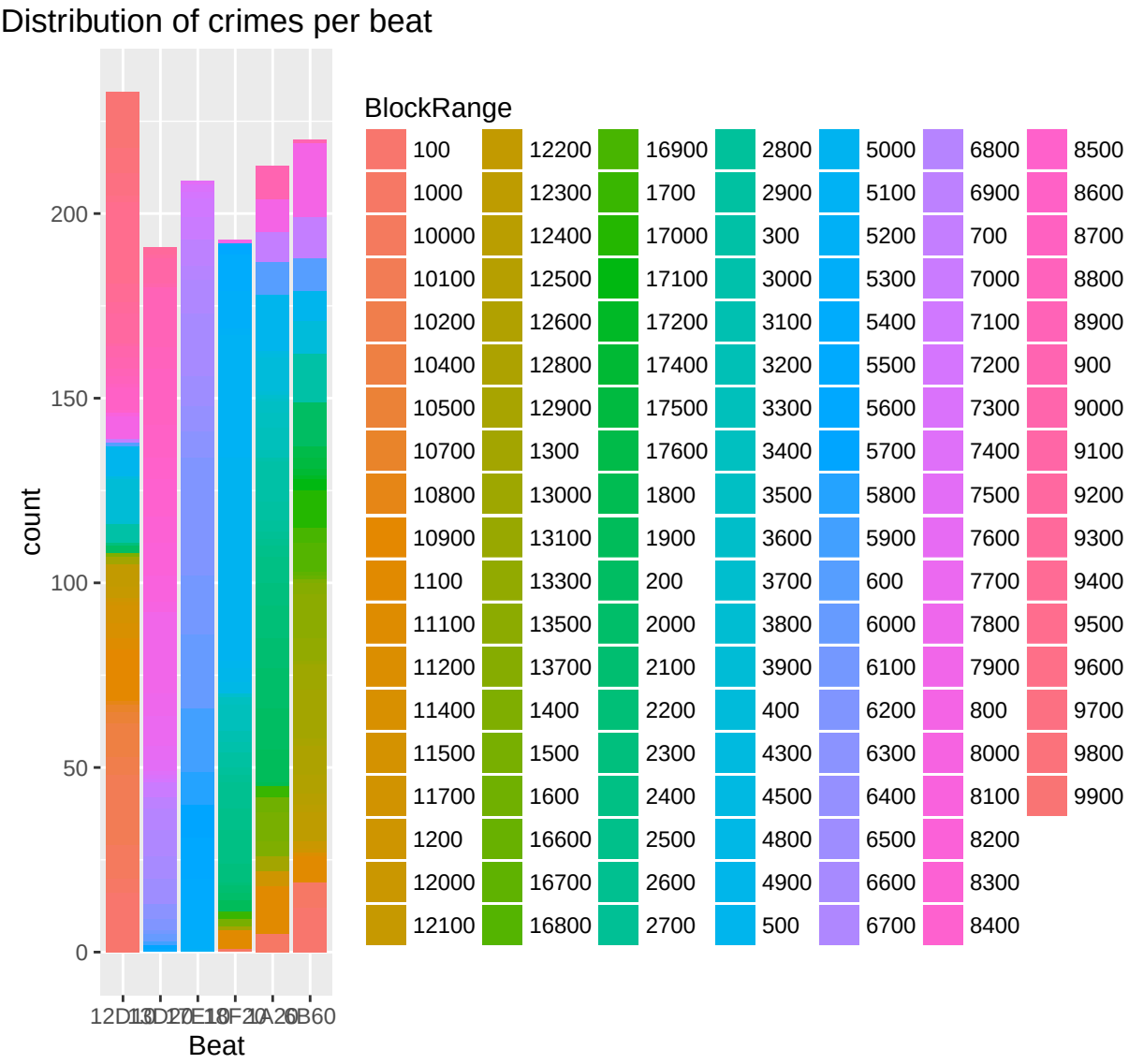
Create all permutations of missing

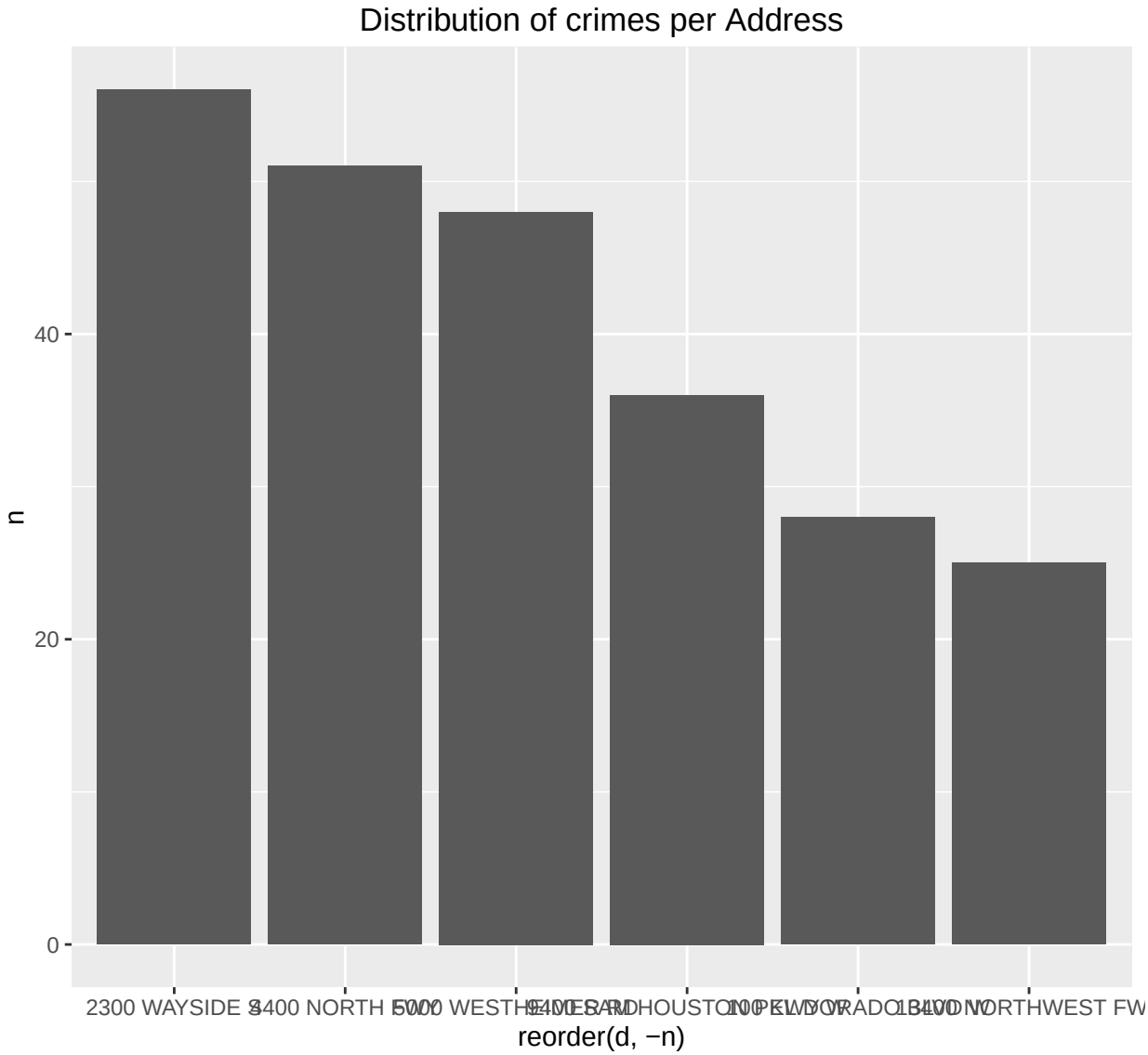
```
create_total_perm <- function(preprocessed, only.week = FALSE) {
  days.between <- get_days_between(info)
  unique_beats <- unique(preprocessed$Beat)
  unique_day_intervals <- unique(preprocessed$DayInterval)

  if(only.week){
    unique_weekdays <- unique(preprocessed$WeekDay)
    all_beats_perm <- data.frame(WeekDay = rep(unique_weekdays, times=length(unique_beats) * length(uni
                                 DayInterval = rep(unique_day_intervals, times=length(unique_beats) * l
                                 Beat = rep(unique_beats, times=length(unique_day_intervals) * length(da
                                 Offenses = rep(0, times=length(unique_day_intervals) * length(days.betw
  } else{
    all_beats_perm <- data.frame(Date = rep(days.between, times=length(unique_beats) * length(unique_da
                                 DayInterval = rep(unique_day_intervals, times=length(unique_beats) * l
                                 Beat = rep(unique_beats, times=length(unique_day_intervals) * length(da
                                 Offenses = rep(0, times=length(unique_day_intervals) * length(days.betw
    all_beats_perm$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)
```

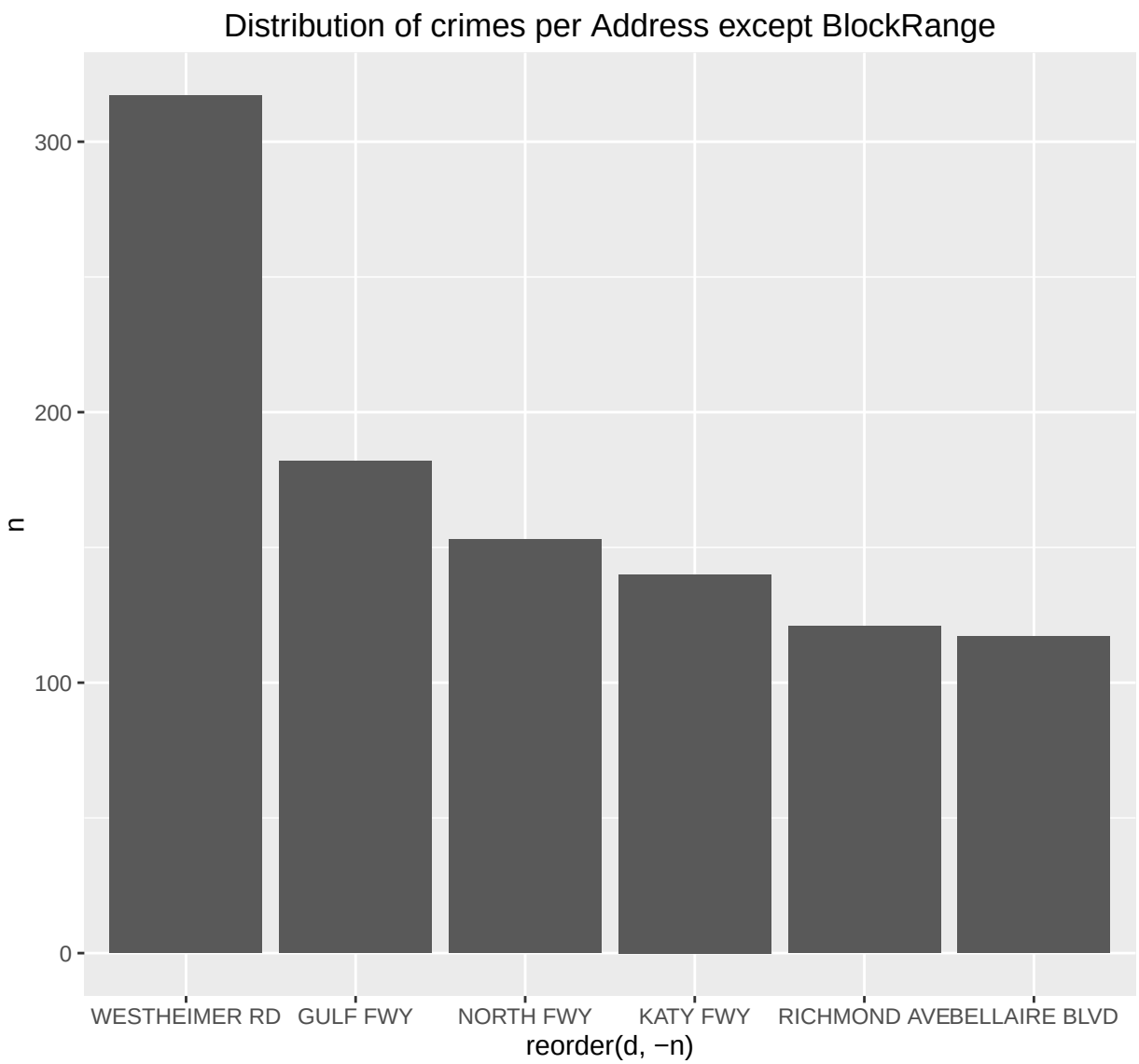
```
    all_beats_perm$WeekDay = as.integer(strftime(all_beats_perm$Date, "%u"))
    all_beats_perm$Day <- day(as.Date(all_beats_perm$Date))
    all_beats_perm$Month <- month(as.Date(all_beats_perm$Date))
    all_beats_perm$Year <- year(as.Date(all_beats_perm$Date))
    all_beats_perm <- all_beats_perm[,!colnames(all_beats_perm) %in% c("Date")]
  }

  return(all_beats_perm)
}
```

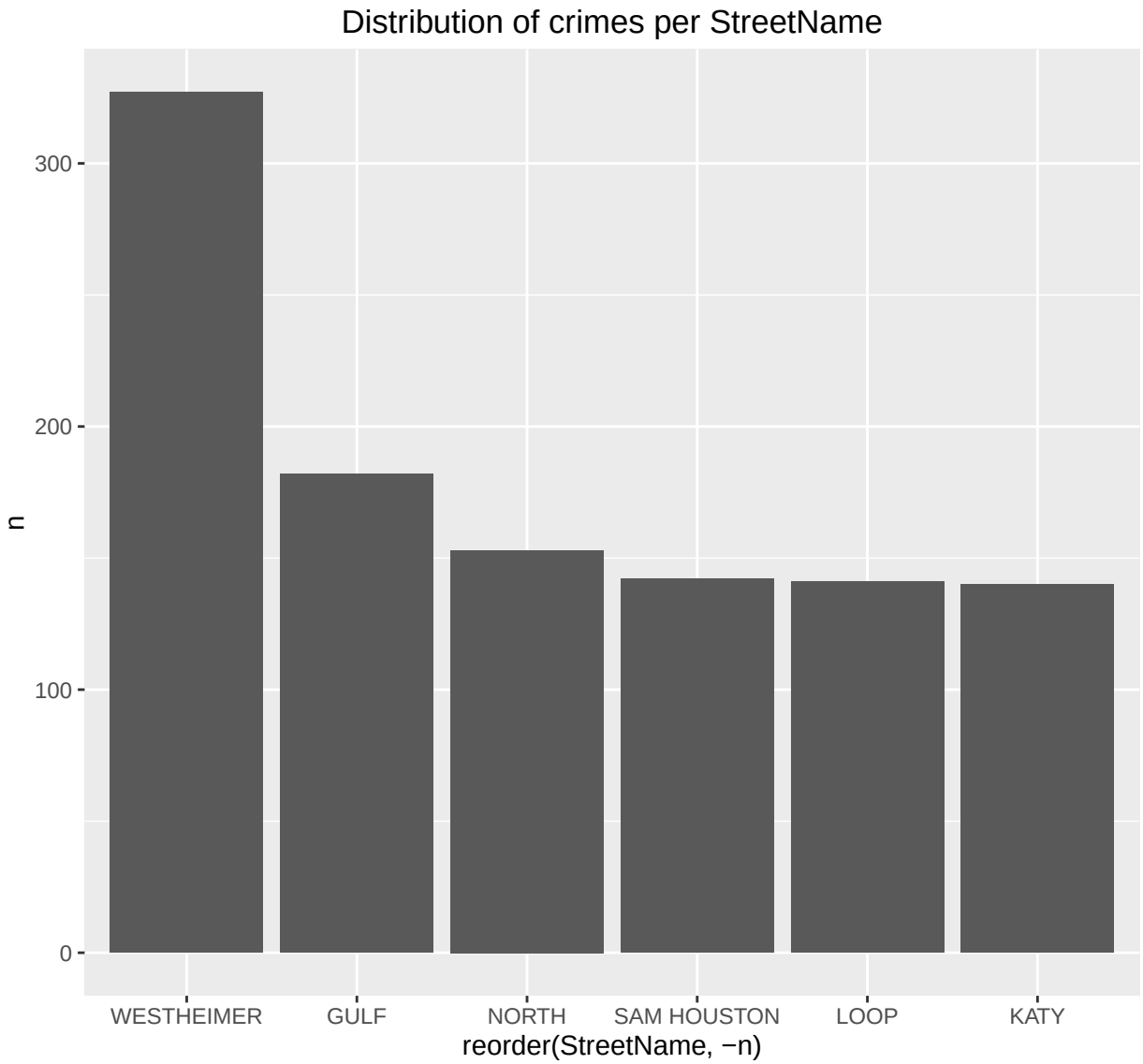
Data Visualisation

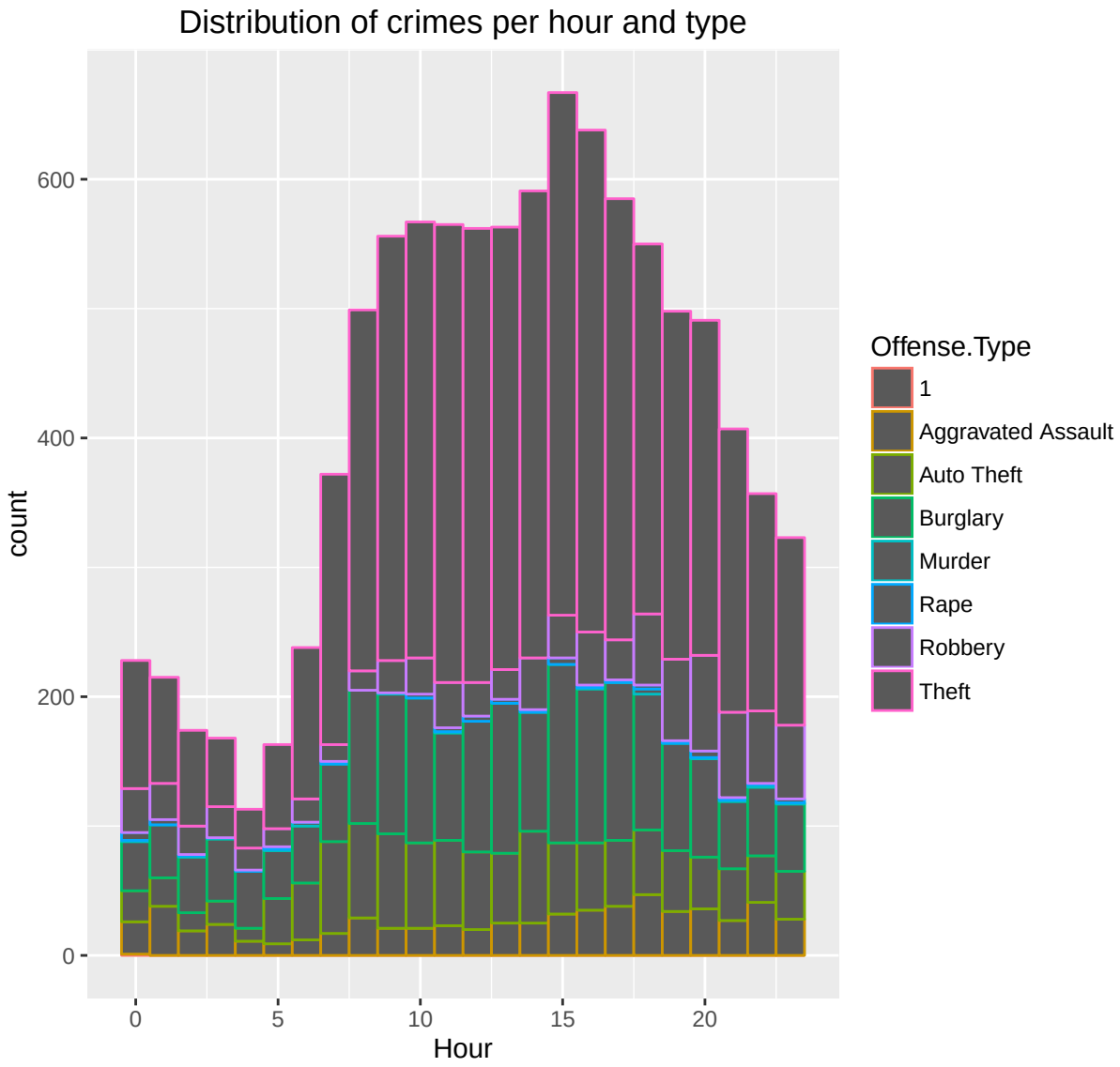
Creating graphics with the data, we can begin to find some usefull patterns in the data.

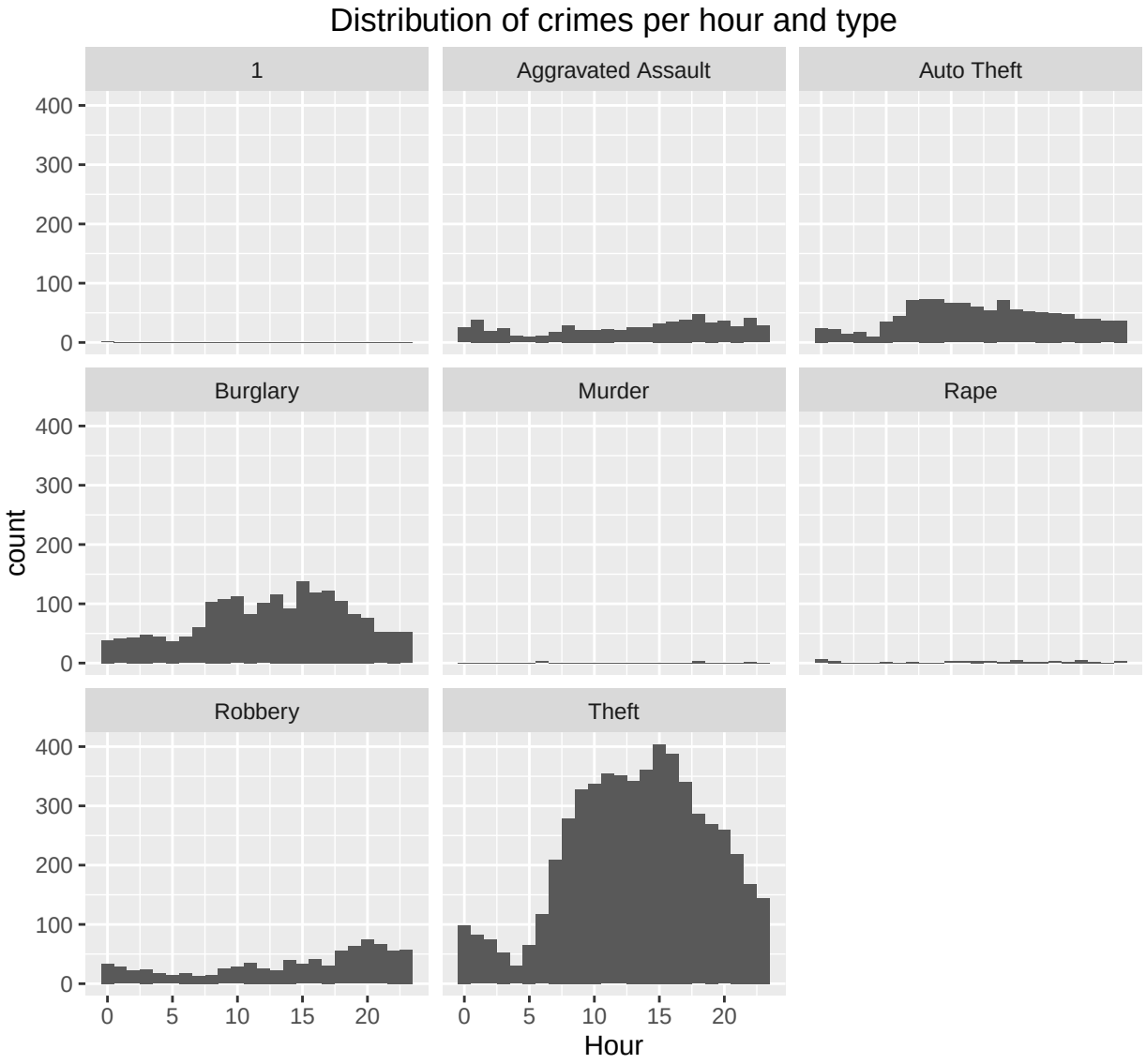



Distribution of crimes per Address
n
40
20
0
2300 WAYSIDE
reorder(d, −n)


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


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


Distribution of crimes per hour and type
600
400
200
0
count
0
5
10
15
20
Hour
Offense.Type
1
Aggravated Assault
Auto Theft
Burglary
Murder
Rape
Robbery
Theft



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. First, we slit the hours of a day into three diferent intervals. The first one is the mourning, starts at 8h(inclusive) until 12h(exclusive). The second one is in the afternoon, starts at 12h (inclusive) until 19h (exclusive). The third interval (night) starts at 19h(exclusive) and ends at 8h (exclusive) of the next day (e.g. if a crime happened at 4th of day 20, it will correspond to the night period of the day 19). We found that some columns of the original data aren't necessary for our predictive model. Only some information for this problem is actually needed so We had to adapt the given data for this problem. First we noticed that the premise of the crimes are irrelevant and that the only necessary attribute to identify an area is the beat. So our final data model for this problem consists of

DayInterval | Beat | Day | Month | Year | Offenses

To convert the given data to this data model, we start by adding the column with the correspondent day interval, and then we create a new data frame only with those column. As there are multiple crimes in the same day interval but we want the number of offenses of the day interval we sum the offenses on the same interval.

```
dataset_prep <- function(x) {
  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

  ret <- 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)
  ret <- group_by(ret, WeekDay, DayInterval, Beat, Day, Month, Year) %>%
    summarize(Offenses = sum(Offenses))

  return(ret)
}
```

At this moment we have the given data adjusted to the problem but we only have cases where was crimes, so we needed to generate all combinations with 0 offenses.

So we generate a new data frame with all combinations of DayInterval, Beat, Day, Month, Year and added to our data only the combinations missing, with Offenses as 0. To do that we merged the 2 data frames and removed the rows with duplicated combination (DayInterval, Beat, Day, Month, Year) and 0 Offenses.

```
get_days_between <- function(info) {
  firstdate <- info$Date[order(format(as.Date(info$Date)))[1]]
  lastdate <- info$Date[tail(order(format(as.Date(info$Date))), n=1)]

  return(seq(as.Date(firstdate), as.Date(lastdate), by="days"))
}

create_total_perm <- function(preprocessed) {
  days.between <- get_days_between(info)
  unique_beats <- unique(preprocessed$Beat)
  unique_day_intervals <- unique(preprocessed$DayInterval)

  all_beats_perm <- data.frame(Date = rep(days.between,
                                          times=length(unique_beats) *
                                            length(unique_day_intervals)),
```

```
                                DayInterval = rep(unique_day_intervals,
                                                  times=length(unique_beats) *
                                                    length(days.between)),
                                Beat = rep(unique_beats,
                                           times=length(unique_day_intervals) *
                                             length(days.between)),
                                Offenses = rep(0,
                                               times=length(unique_day_intervals) *
                                                 length(days.between) *
                                                 length(unique_beats)))
  all_beats_perm$DayInterval <- as.integer(all_beats_perm$DayInterval)
  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)
}

info.preprocessed.total_perm <- create_total_perm(info.preprocessed)
info.preprocessed.joined <- merge(info.preprocessed, info.preprocessed.total_perm,
                                  by=c("WeekDay", "DayInterval", "Beat", "Day", "Month", "Year"),
                                  all=TRUE)
info.preprocessed.joined[is.na(info.preprocessed.joined$Offenses.x),]$Offenses.x <- 0
info.preprocessed.joined$Offenses <- info.preprocessed.joined$Offenses.x
info.preprocessed.joined <- info.preprocessed.joined[,!colnames(info.preprocessed.joined) %in%
                                                       c("Offenses.y", "Offenses.x")]
```

We considered that the given data had all occorrences between the first and the last date of the set.
At this moment we have all the data structured and we can