DSC630 Final Project - Crime Analysis - Part6

Amie Davis

22 July, 2020

Data Sources:

Uniform Crime Reporting Program Data: National Incident-Based Reporting System, [United States], 2016; United States Federal Bureau of Investigation; Inter-university Consortium for Political and Social Research (ICPSR), University of Michigan; https://www.icpsr.umich.edu/icpsrweb/NACJD/NIBRS/
Geodetic Data for US Cities: https://simplemaps.com/data/us-cities

References:

https://www.latlong.net D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

Load Libraries

```
library(readr)
library(dplyr)
library(gdata)
library(caTools)
library(class)
library(tidyr)
library(ggplot2)
```

1. Prepare Data

a) Import the Data

```
# Load Cleaned data from Part1
crime_data <- read_csv("Data/crime_offenses_top6.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Warning: Duplicated column names deduplicated: 'X1' => 'X1_1' [2]

## Warning: 27870 parsing failures.
##  row            col           expected actual
file
```

```
## 1250 ACT_TYPE_OFFC  1/0/T/F/TRUE/FALSE      6
'Data/crime_offenses_top6.csv'
## 1250 ASSG_TYPE_OFFC 1/0/T/F/TRUE/FALSE      L
'Data/crime_offenses_top6.csv'
## 1251 ACT_TYPE_OFFC  1/0/T/F/TRUE/FALSE      4
'Data/crime_offenses_top6.csv'
## 1251 ASSG_TYPE_OFFC 1/0/T/F/TRUE/FALSE      L
'Data/crime_offenses_top6.csv'
## 1253 ACT_TYPE_OFFC  1/0/T/F/TRUE/FALSE      6
'Data/crime_offenses_top6.csv'
## .... .............. .................. ......
.............................
## See problems(...) for more details.

str(crime_data)

## tibble [3,945,595 x 59] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ X1                 : num [1:3945595] 1 2 3 4 5 6 7 8 9 10 ...
##  $ X1_1               : num [1:3945595] 1 2 3 4 5 6 7 8 9 6 ...
##  $ ORI                : chr [1:3945595] "MA0010100" "MA0010100"
"MA0010100" "MA0010100" "MA0010100" ...
##  $ INC_NUM            : chr [1:3945595] "83- X9Y 728N" "83-1X9Y 728N" "83-
1XZ8 728N" "83-AXZ8 728N" ...
##  $ VIC_INC_DATE       : Date[1:3945595], format: "2016-01-01" "2016-01-01"
...
##  $ VICTIM_TYPE        : chr [1:3945595] "I" "I" "I" "B" ...
##  $ ACT_TYPE_OFFC      : logi [1:3945595] NA NA NA NA NA NA ...
##  $ ASSG_TYPE_OFFC     : logi [1:3945595] NA NA NA NA NA NA ...
##  $ AGE_OF_VICTIM      : num [1:3945595] 27 25 51 36 50 36 36 36 36 36 ...
##  $ SEX_OF_VICTIM      : chr [1:3945595] "F" "F" "M" NA ...
##  $ RACE_OF_VICTIM     : chr [1:3945595] "W" "W" "B" NA ...
##  $ ETHNIC_OF_VIC      : chr [1:3945595] "U" "N" "N" NA ...
##  $ VIC_RESIDENT       : chr [1:3945595] "N" "R" "R" NA ...
##  $ ASSAULT_CIRC1      : num [1:3945595] NA NA NA NA NA NA NA NA NA NA ...
##  $ ASSAULT_CIRC2      : logi [1:3945595] NA NA NA NA NA NA ...
##  $ JUST_HOM_CIRC      : logi [1:3945595] NA NA NA NA NA NA ...
##  $ INJURY_TYPE1       : chr [1:3945595] NA NA NA NA ...
##  $ INJURY_TYPE2       : chr [1:3945595] NA NA NA NA ...
##  $ INJURY_TYPE3       : chr [1:3945595] NA NA NA NA ...
##  $ INJURY_TYPE4       : logi [1:3945595] NA NA NA NA NA NA ...
##  $ INJURY_TYPE5       : logi [1:3945595] NA NA NA NA NA NA ...
##  $ NUM_RECS_PER_VICTIM: num [1:3945595] 1 1 1 1 1 4 NA NA NA 4 ...
##  $ VIC_INC_YEAR       : num [1:3945595] 2016 2016 2016 2016 2016 ...
##  $ VIC_INC_MONTH      : num [1:3945595] 1 1 1 1 1 1 1 1 1 1 ...
##  $ VIC_INC_DAY        : num [1:3945595] 1 1 1 2 2 2 2 2 2 2 ...
##  $ VIC_INC_DOW        : chr [1:3945595] "Friday" "Friday" "Friday"
"Saturday" ...
##  $ NUM_STATE_CODE     : num [1:3945595] 20 20 20 20 20 20 20 20 20 20 ...
##  $ CITY               : chr [1:3945595] "BARNSTABLE" "BARNSTABLE"
"BARNSTABLE" "BARNSTABLE" ...
```

```
##  $ STATE              : chr [1:3945595] "MA" "MA" "MA" "MA" ...
##  $ POP_GROUP          : num [1:3945595] 4 4 4 4 4 4 4 4 4 4 ...
##  $ CTRY_DIVISION      : num [1:3945595] 1 1 1 1 1 1 1 1 1 1 ...
##  $ CTRY_REGION        : num [1:3945595] 1 1 1 1 1 1 1 1 1 1 ...
##  $ AGENCY_IND         : num [1:3945595] 1 1 1 1 1 1 1 1 1 1 ...
##  $ CORE_CITY          : chr [1:3945595] "Y" "Y" "Y" "Y" ...
##  $ FBI_OFFICE         : num [1:3945595] 3090 3090 3090 3090 3090 3090 3090
3090 3090 3090 ...
##  $ JUDICIAL_DIST      : chr [1:3945595] "195A" "195A" "195A" "195A" ...
##  $ CURRENT_POP1       : num [1:3945595] 43974 43974 43974 43974 43974 ...
##  $ UCR_COUNTY_CD1     : num [1:3945595] 1 1 1 1 1 1 1 1 1 1 ...
##  $ MSA_CD1            : num [1:3945595] 76 76 76 76 76 76 76 76 76 76 ...
##  $ LAST_POP1          : num [1:3945595] 44392 44392 44392 44392 44392 ...
##  $ FIPS_COUNTY1       : chr [1:3945595] "001" "001" "001" "001" ...
##  $ city_ascii         : chr [1:3945595] "Barnstable" "Barnstable"
"Barnstable" "Barnstable" ...
##  $ state_name         : chr [1:3945595] "Massachusetts" "Massachusetts"
"Massachusetts" "Massachusetts" ...
##  $ county_fips        : num [1:3945595] 25001 25001 25001 25001 25001 ...
##  $ county_name        : chr [1:3945595] "Barnstable" "Barnstable"
"Barnstable" "Barnstable" ...
##  $ county_fips_all    : num [1:3945595] 25001 25001 25001 25001 25001 ...
##  $ county_name_all    : chr [1:3945595] "Barnstable" "Barnstable"
"Barnstable" "Barnstable" ...
##  $ lat                : num [1:3945595] 41.7 41.7 41.7 41.7 41.7 ...
##  $ lng                : num [1:3945595] -70.4 -70.4 -70.4 -70.4 -70.4 ...
##  $ population         : num [1:3945595] 241132 241132 241132 241132 241132
...
##  $ density            : num [1:3945595] 284 284 284 284 284 284 284 284
284 284 ...
##  $ source             : chr [1:3945595] "polygon" "polygon" "polygon"
"polygon" ...
##  $ military           : logi [1:3945595] FALSE FALSE FALSE FALSE FALSE
FALSE ...
##  $ incorporated       : logi [1:3945595] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ timezone           : chr [1:3945595] "America/New_York"
"America/New_York" "America/New_York" "America/New_York" ...
##  $ ranking            : num [1:3945595] 2 2 2 2 2 2 2 2 2 2 ...
##  $ zips               : chr [1:3945595] "02635 02637 02630 02632 02601
02655 02672 02675 02647 02648 02668 02634" "02635 02637 02630 02632 02601
02655 02672 02675 02647 02648 02668 02634" "02635 02637 02630 02632 02601
02655 02672 02675 02647 02648 02668 02634" "02635 02637 02630 02632 02601
02655 02672 02675 02647 02648 02668 02634" ...
##  $ id                 : num [1:3945595] 1.84e+09 1.84e+09 1.84e+09
1.84e+09 1.84e+09 ...
##  $ OFF_CODE           : chr [1:3945595] "26A" "23G" "13C" "23H" ...
##  - attr(*, "problems")= tibble [27,870 x 5] (S3: tbl_df/tbl/data.frame)
##   ..$ row     : int [1:27870] 1250 1250 1251 1251 1253 1253 1254 1254 1256
1256 ...
##   ..$ col     : chr [1:27870] "ACT_TYPE_OFFC" "ASSG_TYPE_OFFC"
```

```
"ACT_TYPE_OFFC" "ASSG_TYPE_OFFC" ...
##   ..$ expected: chr [1:27870] "1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE"
"1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE" ...
##   ..$ actual  : chr [1:27870] "6" "L" "4" "L" ...
##   ..$ file    : chr [1:27870] "'Data/crime_offenses_top6.csv'"
"'Data/crime_offenses_top6.csv'" "'Data/crime_offenses_top6.csv'"
"'Data/crime_offenses_top6.csv'" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   X1 = col_double(),
##   ..   X1_1 = col_double(),
##   ..   ORI = col_character(),
##   ..   INC_NUM = col_character(),
##   ..   VIC_INC_DATE = col_date(format = ""),
##   ..   VICTIM_TYPE = col_character(),
##   ..   ACT_TYPE_OFFC = col_logical(),
##   ..   ASSG_TYPE_OFFC = col_logical(),
##   ..   AGE_OF_VICTIM = col_double(),
##   ..   SEX_OF_VICTIM = col_character(),
##   ..   RACE_OF_VICTIM = col_character(),
##   ..   ETHNIC_OF_VIC = col_character(),
##   ..   VIC_RESIDENT = col_character(),
##   ..   ASSAULT_CIRC1 = col_double(),
##   ..   ASSAULT_CIRC2 = col_logical(),
##   ..   JUST_HOM_CIRC = col_logical(),
##   ..   INJURY_TYPE1 = col_character(),
##   ..   INJURY_TYPE2 = col_character(),
##   ..   INJURY_TYPE3 = col_character(),
##   ..   INJURY_TYPE4 = col_logical(),
##   ..   INJURY_TYPE5 = col_logical(),
##   ..   NUM_RECS_PER_VICTIM = col_double(),
##   ..   VIC_INC_YEAR = col_double(),
##   ..   VIC_INC_MONTH = col_double(),
##   ..   VIC_INC_DAY = col_double(),
##   ..   VIC_INC_DOW = col_character(),
##   ..   NUM_STATE_CODE = col_double(),
##   ..   CITY = col_character(),
##   ..   STATE = col_character(),
##   ..   POP_GROUP = col_double(),
##   ..   CTRY_DIVISION = col_double(),
##   ..   CTRY_REGION = col_double(),
##   ..   AGENCY_IND = col_double(),
##   ..   CORE_CITY = col_character(),
##   ..   FBI_OFFICE = col_double(),
##   ..   JUDICIAL_DIST = col_character(),
##   ..   CURRENT_POP1 = col_double(),
##   ..   UCR_COUNTY_CD1 = col_double(),
##   ..   MSA_CD1 = col_double(),
##   ..   LAST_POP1 = col_double(),
##   ..   FIPS_COUNTY1 = col_character(),
```

```
##   ..   city_ascii = col_character(),
##   ..   state_name = col_character(),
##   ..   county_fips = col_double(),
##   ..   county_name = col_character(),
##   ..   county_fips_all = col_double(),
##   ..   county_name_all = col_character(),
##   ..   lat = col_double(),
##   ..   lng = col_double(),
##   ..   population = col_double(),
##   ..   density = col_double(),
##   ..   source = col_character(),
##   ..   military = col_logical(),
##   ..   incorporated = col_logical(),
##   ..   timezone = col_character(),
##   ..   ranking = col_double(),
##   ..   zips = col_character(),
##   ..   id = col_double(),
##   ..   OFF_CODE = col_character()
##   .. )
```

b) Remove Unnecessary Columns

```
# The following fields are not needed for visualizations and will be removed.

  crime_data[ ,c(
"X1",
"X1_1",
"ORI",    #character
"INC_NUM",
"VIC_INC_DATE",
#"VICTIM_TYPE",
"ACT_TYPE_OFFC",
"ASSG_TYPE_OFFC",
#"AGE_OF_VICTIM",
#"SEX_OF_VICTIM",
#"RACE_OF_VICTIM",
#"ETHNIC_OF_VIC",
#"VIC_RESIDENT",
"ASSAULT_CIRC1",
"ASSAULT_CIRC2",
"JUST_HOM_CIRC",
"INJURY_TYPE1",
"INJURY_TYPE2",
"INJURY_TYPE3",
"INJURY_TYPE4",
"INJURY_TYPE5",
"NUM_RECS_PER_VICTIM",
"VIC_INC_YEAR",
#"VIC_INC_MONTH",
"VIC_INC_DAY",
```

```
#"VIC_INC_DOW",
"NUM_STATE_CODE",
"CITY",    #character
#"STATE",
#"POP_GROUP",
#"CTRY_DIVISION",
#"CTRY_REGION",
#"AGENCY_IND",
#"CORE_CITY",
"FBI_OFFICE",
"JUDICIAL_DIST",    #character
#"CURRENT_POP1",
"UCR_COUNTY_CD1",
"MSA_CD1",
"LAST_POP1",
"FIPS_COUNTY1",    #character
"city_ascii",
"state_name",
"county_fips",
"county_name",
"county_fips_all",
"county_name_all",
#"lat",
#"lng",
#"population",
#"density",
"source",
"military",
#"incorporated",
#"timezone",
#"ranking",
"zips",
"id"
#"OFF_CODE"
)] <- list(NULL)

head(crime_data)
```

```
## # A tibble: 6 x 23
##   VICTIM_TYPE AGE_OF_VICTIM SEX_OF_VICTIM RACE_OF_VICTIM ETHNIC_OF_VIC
##   <chr>               <dbl> <chr>         <chr>          <chr>
## 1 I                      27 F             W              U
## 2 I                      25 F             W              N
## 3 I                      51 M             B              N
## 4 B                      36 <NA>          <NA>           <NA>
## 5 I                      50 M             W              U
## 6 B                      36 <NA>          <NA>           <NA>
## # ... with 18 more variables: VIC_RESIDENT <chr>, VIC_INC_MONTH <dbl>,
## #   VIC_INC_DOW <chr>, STATE <chr>, POP_GROUP <dbl>, CTRY_DIVISION <dbl>,
## #   CTRY_REGION <dbl>, AGENCY_IND <dbl>, CORE_CITY <chr>, CURRENT_POP1
```

```
<dbl>,
## #   lat <dbl>, lng <dbl>, population <dbl>, density <dbl>, incorporated
<lgl>,
## #   timezone <chr>, ranking <dbl>, OFF_CODE <chr>
```

c) Convert NA Data to Unknown Category where applicable

```
crime_data$SEX_OF_VICTIM[is.na(crime_data$SEX_OF_VICTIM)] <- 'U'
crime_data$RACE_OF_VICTIM[is.na(crime_data$RACE_OF_VICTIM)] <- 'U'
crime_data$ETHNIC_OF_VIC[is.na(crime_data$ETHNIC_OF_VIC)] <- 'U'
crime_data$VIC_RESIDENT[is.na(crime_data$VIC_RESIDENT)] <- 'U'
crime_data$POP_GROUP[is.na(crime_data$POP_GROUP)] <- 0
```

d) Limit to records with geodetic (lat/long) coordinates

```
crime_data <- filter(crime_data, !is.na(lat))
```

e) Exclude remaining records with NA values

```
#crime_data %>% drop_na()
#crime_data <- na.omit(crime_data)
#summary(crime_data)
```

f) Convert categorical variables to factors and then numeric

```
crime_data$VICTIM_TYPE <- factor(crime_data$VICTIM_TYPE)
crime_data$SEX_OF_VICTIM <- factor(crime_data$SEX_OF_VICTIM)
crime_data$RACE_OF_VICTIM <- factor(crime_data$RACE_OF_VICTIM)
crime_data$ETHNIC_OF_VIC <- factor(crime_data$ETHNIC_OF_VIC)
crime_data$VIC_RESIDENT <- factor(crime_data$VIC_RESIDENT)
crime_data$VIC_INC_DOW <- factor(crime_data$VIC_INC_DOW)
crime_data$STATE <- factor(crime_data$STATE)
crime_data$POP_GROUP <- factor(crime_data$POP_GROUP)
crime_data$CTRY_DIVISION <- factor(crime_data$CTRY_DIVISION)
crime_data$CTRY_REGION <- factor(crime_data$CTRY_REGION)
crime_data$AGENCY_IND <- factor(crime_data$AGENCY_IND)
crime_data$CORE_CITY <- factor(crime_data$CORE_CITY)
crime_data$incorporated <- factor(crime_data$incorporated)
crime_data$timezone <- factor(crime_data$timezone)
crime_data$OFF_CODE <- factor(crime_data$OFF_CODE)

# Convert to numeric to compute distances
# Add small amount of noise to reduce ties
crime_data$VICTIM_TYPE <- jitter(as.numeric(crime_data$VICTIM_TYPE))
crime_data$SEX_OF_VICTIM <- jitter(as.numeric(crime_data$SEX_OF_VICTIM))
crime_data$RACE_OF_VICTIM <- jitter(as.numeric(crime_data$RACE_OF_VICTIM))
crime_data$ETHNIC_OF_VIC <- jitter(as.numeric(crime_data$ETHNIC_OF_VIC))
crime_data$VIC_RESIDENT <- jitter(as.numeric(crime_data$VIC_RESIDENT))
crime_data$VIC_INC_DOW <- jitter(as.numeric(crime_data$VIC_INC_DOW))
crime_data$STATE <- jitter(as.numeric(crime_data$STATE))
crime_data$POP_GROUP <- jitter(as.numeric(crime_data$POP_GROUP))
crime_data$CTRY_DIVISION <- jitter(as.numeric(crime_data$CTRY_DIVISION))
crime_data$CTRY_REGION <- jitter(as.numeric(crime_data$CTRY_REGION))
```

```r
crime_data$AGENCY_IND <- jitter(as.numeric(crime_data$AGENCY_IND))
crime_data$CORE_CITY <- jitter(as.numeric(crime_data$CORE_CITY))
crime_data$incorporated <- jitter(as.numeric(crime_data$incorporated))
crime_data$timezone <- jitter(as.numeric(crime_data$timezone))
crime_data$OFF_CODE <- jitter(as.numeric(crime_data$OFF_CODE))

summary(crime_data)
```

```
##   VICTIM_TYPE    AGE_OF_VICTIM   SEX_OF_VICTIM   RACE_OF_VICTIM
##  Min.   :0.800   Min.   : 0.00   Min.   :0.800   Min.   :0.800
##  1st Qu.:3.855   1st Qu.:27.00   1st Qu.:1.062   1st Qu.:4.833
##  Median :3.988   Median :36.00   Median :1.930   Median :5.154
##  Mean   :3.968   Mean   :36.93   Mean   :1.875   Mean   :4.780
##  3rd Qu.:4.121   3rd Qu.:45.00   3rd Qu.:2.812   3rd Qu.:5.985
##  Max.   :9.200   Max.   :99.00   Max.   :3.200   Max.   :6.200
##  ETHNIC_OF_VIC    VIC_RESIDENT   VIC_INC_MONTH     VIC_INC_DOW
##  Min.   :0.800   Min.   :0.800   Min.   : 1.000   Min.   :0.800
##  1st Qu.:1.989   1st Qu.:1.899   1st Qu.: 4.000   1st Qu.:2.049
##  Median :2.812   Median :2.086   Median : 7.000   Median :3.961
##  Mean   :2.476   Mean   :2.229   Mean   : 6.522   Mean   :3.928
##  3rd Qu.:3.006   3rd Qu.:2.912   3rd Qu.: 9.000   3rd Qu.:5.881
##  Max.   :3.200   Max.   :3.200   Max.   :12.000   Max.   :7.200
##      STATE         POP_GROUP     CTRY_DIVISION    CTRY_REGION
##  Min.   :0.800   Min.   :0.800   Min.   :0.800   Min.   :0.800
##  1st Qu.:2.849   1st Qu.:1.077   1st Qu.:2.001   1st Qu.:2.001
##  Median :3.976   Median :2.144   Median :2.977   Median :2.878
##  Mean   :3.899   Mean   :2.739   Mean   :3.124   Mean   :2.713
##  3rd Qu.:5.122   3rd Qu.:4.035   3rd Qu.:4.122   3rd Qu.:3.156
##  Max.   :6.200   Max.   :7.200   Max.   :5.200   Max.   :4.200
##    AGENCY_IND       CORE_CITY      CURRENT_POP1         lat
##  Min.   :0.8005   Min.   :0.800   Min.   :   108   Min.   :32.21
##  1st Qu.:1.9160   1st Qu.:1.010   1st Qu.: 31864   1st Qu.:35.82
##  Median :2.0318   Median :1.818   Median : 88371   Median :41.08
##  Mean   :2.1390   Mean   :1.524   Mean   :206601   Mean   :40.42
##  3rd Qu.:2.1473   3rd Qu.:2.009   3rd Qu.:239885   3rd Qu.:42.84
##  Max.   :5.2000   Max.   :2.200   Max.   :844206   Max.   :49.00
##       lng            population         density      incorporated
##  Min.   :-124.39   Min.   :    106   Min.   :  23   Min.   :0.8001
##  1st Qu.: -89.98   1st Qu.:  24958   1st Qu.: 552   1st Qu.:1.8983
##  Median : -83.73   Median :  94491   Median : 902   Median :1.9989
##  Mean   : -90.98   Mean   : 548652   Mean   :1125   Mean   :1.9942
##  3rd Qu.: -81.86   3rd Qu.: 602694   3rd Qu.:1505   3rd Qu.:2.0993
##  Max.   : -70.36   Max.   :3643765   Max.   :7616   Max.   :2.2000
##     timezone         ranking         OFF_CODE
##  Min.   :0.800   Min.   :1.000   Min.   : 0.8001
##  1st Qu.:2.057   1st Qu.:2.000   1st Qu.:11.9564
##  Median :3.169   Median :2.000   Median :20.9862
##  Mean   :3.511   Mean   :2.264   Mean   :21.6582
##  3rd Qu.:4.993   3rd Qu.:3.000   3rd Qu.:28.1503
##  Max.   :6.200   Max.   :3.000   Max.   :52.1991
```

```r
str(crime_data)
```

```
## tibble [3,104,581 x 23] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ VICTIM_TYPE   : num [1:3104581] 3.963 3.965 3.945 0.838 4.11 ...
##  $ AGE_OF_VICTIM : num [1:3104581] 27 25 51 36 50 36 36 36 36 36 ...
##  $ SEX_OF_VICTIM : num [1:3104581] 0.963 1.024 1.809 2.894 1.935 ...
##  $ RACE_OF_VICTIM: num [1:3104581] 5.97 6.06 1.95 4.83 5.89 ...
##  $ ETHNIC_OF_VIC : num [1:3104581] 3.03 2.01 2.17 2.81 3.05 ...
##  $ VIC_RESIDENT  : num [1:3104581] 0.835 2.165 2.177 3.041 0.991 ...
##  $ VIC_INC_MONTH : num [1:3104581] 1 1 1 1 1 1 1 1 1 1 ...
##  $ VIC_INC_DOW   : num [1:3104581] 1.092 0.887 0.815 3.09 3.128 ...
##  $ STATE         : num [1:3104581] 1.109 0.924 1.154 0.902 1.142 ...
##  $ POP_GROUP     : num [1:3104581] 4.14 4.05 3.85 4.08 4.09 ...
##  $ CTRY_DIVISION : num [1:3104581] 1.016 0.816 1.156 1.178 1.183 ...
##  $ CTRY_REGION   : num [1:3104581] 1.166 0.94 0.962 1.081 1.031 ...
##  $ AGENCY_IND    : num [1:3104581] 2.18 2.12 1.81 2.11 2.01 ...
##  $ CORE_CITY     : num [1:3104581] 2.18 2.07 1.97 1.97 2.02 ...
##  $ CURRENT_POP1  : num [1:3104581] 43974 43974 43974 43974 43974 ...
##  $ lat           : num [1:3104581] 41.7 41.7 41.7 41.7 41.7 ...
##  $ lng           : num [1:3104581] -70.4 -70.4 -70.4 -70.4 -70.4 ...
##  $ population    : num [1:3104581] 241132 241132 241132 241132 241132 ...
##  $ density       : num [1:3104581] 284 284 284 284 284 284 284 284 284 284
...
##  $ incorporated  : num [1:3104581] 2 2.13 1.92 1.97 2.11 ...
##  $ timezone      : num [1:3104581] 5.05 5.18 4.95 4.95 5.01 ...
##  $ ranking       : num [1:3104581] 2 2 2 2 2 2 2 2 2 2 ...
##  $ OFF_CODE      : num [1:3104581] 26 21.8 11.8 23 35.2 ...
##  - attr(*, "problems")= tibble [27,870 x 5] (S3: tbl_df/tbl/data.frame)
##   ..$ row     : int [1:27870] 1250 1250 1251 1251 1253 1253 1254 1254 1256
1256 ...
##   ..$ col     : chr [1:27870] "ACT_TYPE_OFFC" "ASSG_TYPE_OFFC"
"ACT_TYPE_OFFC" "ASSG_TYPE_OFFC" ...
##   ..$ expected: chr [1:27870] "1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE"
"1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE" ...
##   ..$ actual  : chr [1:27870] "6" "L" "4" "L" ...
##   ..$ file    : chr [1:27870] "'Data/crime_offenses_top6.csv'"
"'Data/crime_offenses_top6.csv'" "'Data/crime_offenses_top6.csv'"
"'Data/crime_offenses_top6.csv'" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   X1 = col_double(),
##   ..   X1_1 = col_double(),
##   ..   ORI = col_character(),
##   ..   INC_NUM = col_character(),
##   ..   VIC_INC_DATE = col_date(format = ""),
##   ..   VICTIM_TYPE = col_character(),
##   ..   ACT_TYPE_OFFC = col_logical(),
##   ..   ASSG_TYPE_OFFC = col_logical(),
##   ..   AGE_OF_VICTIM = col_double(),
##   ..   SEX_OF_VICTIM = col_character(),
```

```
##   ..   RACE_OF_VICTIM = col_character(),
##   ..   ETHNIC_OF_VIC = col_character(),
##   ..   VIC_RESIDENT = col_character(),
##   ..   ASSAULT_CIRC1 = col_double(),
##   ..   ASSAULT_CIRC2 = col_logical(),
##   ..   JUST_HOM_CIRC = col_logical(),
##   ..   INJURY_TYPE1 = col_character(),
##   ..   INJURY_TYPE2 = col_character(),
##   ..   INJURY_TYPE3 = col_character(),
##   ..   INJURY_TYPE4 = col_logical(),
##   ..   INJURY_TYPE5 = col_logical(),
##   ..   NUM_RECS_PER_VICTIM = col_double(),
##   ..   VIC_INC_YEAR = col_double(),
##   ..   VIC_INC_MONTH = col_double(),
##   ..   VIC_INC_DAY = col_double(),
##   ..   VIC_INC_DOW = col_character(),
##   ..   NUM_STATE_CODE = col_double(),
##   ..   CITY = col_character(),
##   ..   STATE = col_character(),
##   ..   POP_GROUP = col_double(),
##   ..   CTRY_DIVISION = col_double(),
##   ..   CTRY_REGION = col_double(),
##   ..   AGENCY_IND = col_double(),
##   ..   CORE_CITY = col_character(),
##   ..   FBI_OFFICE = col_double(),
##   ..   JUDICIAL_DIST = col_character(),
##   ..   CURRENT_POP1 = col_double(),
##   ..   UCR_COUNTY_CD1 = col_double(),
##   ..   MSA_CD1 = col_double(),
##   ..   LAST_POP1 = col_double(),
##   ..   FIPS_COUNTY1 = col_character(),
##   ..   city_ascii = col_character(),
##   ..   state_name = col_character(),
##   ..   county_fips = col_double(),
##   ..   county_name = col_character(),
##   ..   county_fips_all = col_double(),
##   ..   county_name_all = col_character(),
##   ..   lat = col_double(),
##   ..   lng = col_double(),
##   ..   population = col_double(),
##   ..   density = col_double(),
##   ..   source = col_character(),
##   ..   military = col_logical(),
##   ..   incorporated = col_logical(),
##   ..   timezone = col_character(),
##   ..   ranking = col_double(),
##   ..   zips = col_character(),
##   ..   id = col_double(),
##   ..   OFF_CODE = col_character()
##   .. )
```

d) Split the data set, randomly into test and train sets.

```
split_off_set <- sample.split(crime_data$OFF_CODE,SplitRatio=0.7)
train_off_set <- subset(crime_data, split_off_set=="TRUE")
test_off_set <- subset(crime_data, split_off_set=="FALSE")
```

Separate Labels

Before running the data through a nearest neighbor model, we need to separate the labels from the data.

```
train_off_labels <- train_off_set[,1, drop=TRUE]
test_off_labels <- test_off_set[,1, drop=TRUE]
train_off_data <- train_off_set[,3:4]
test_off_data <- test_off_set[,3:4]
```

d) Build kNN models with training dataset

Now, we can build the models with the training sets, using a variety of k values.

```
knn_off.3<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=3)
knn_off.5<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=5)
knn_off.10<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=10)
```

e) Test kNN model with test dataset

```
# Accuracy for offense model
ACC_off.3 <- 100 * sum(round(test_off_labels,0) ==
round(as.numeric(as.character(knn_off.3,0)),0))/NROW(round(test_off_labels,0)
)
ACC_off.5 <- 100 * sum(round(test_off_labels,0) ==
round(as.numeric(as.character(knn_off.5,0)),0))/NROW(round(test_off_labels,0)
)
ACC_off.10 <- 100 * sum(round(test_off_labels,0) ==
round(as.numeric(as.character(knn_off.10,0)),0))/NROW(round(test_off_labels,0
))

# Add accuracy values to a new data frame
k <- c(3,5,10)
ACC <- c(ACC_off.3, ACC_off.5, ACC_off.10)
ACC_df <- data.frame(k, ACC, stringsAsFactors=FALSE)

ACC_off.3

## [1] 84.37063

ACC_off.5

## [1] 84.31126
```

```
ACC_off.10
```

```
## [1] 84.33499
```

Plot accuracy values

```
# Convert data types for data frame
ACC_df$k <- as.numeric(ACC_df$k)
ACC_df$ACC <- as.numeric(ACC_df$ACC)

ggplot(ACC_df, aes(x=k, y=ACC, col="light orange")) +
  geom_point() +
  geom_smooth() +
  labs(title="kNN Model Accuracy Values", y="Accuracy") +
    theme(legend.position = "none")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 2.965
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2.035
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 25.351
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : span too small.
fewer
## data values than degrees of freedom.
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used
at
## 2.965
```

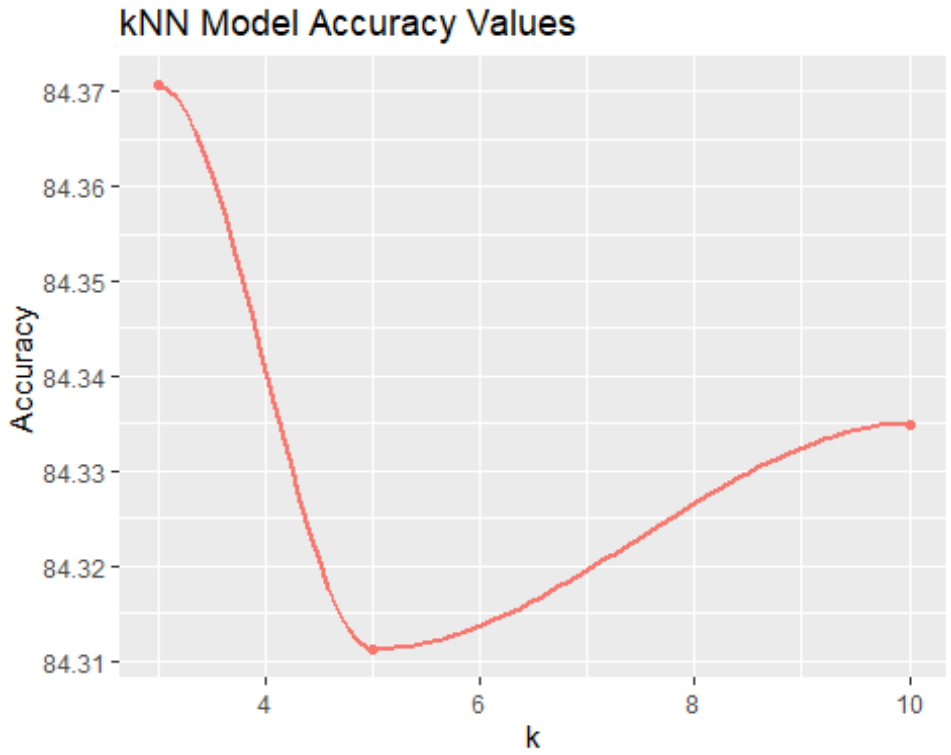
```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood
radius
## 2.035
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
```

```
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal
condition
## number 0

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other
near
## singularities as well. 25.351

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max;
returning -
## Inf
```



The best I will get with this model is around 84% accuracy with k=3 clusters.