Crimes in San Francisco

Maike Heinrich

14 10 2021

Predicting crimes in San Francisco using machine learning

Overview

1. Introduction and summary
2. Pre-Processing

- Downloading the data and libraries
- Excursion robberies with observation and visualization
- Create the final data set with observation and visualization

3. Methods and modelling
4. Analysis and results of the training
5. Final test and conclusion

1. Introduction and summary

The "Crime Incident Reporting System" of the San Francisco Police Department provides a data set of crimes which took place in the different Police Districts of the city of San Francisco from July to December 2012. The data set is available on the GeoDa Data and Lab platform. Content of this data set is among other variables the kind of crime, the date, the police district, the location and the kind of resolution.

In the first part, a deeper observation of the robbery data set will be shown, with some visualizations to get some insights.

After that, all four crime sets will be combined to get the final set for predictions. Target of this report is to wrangle the data, visualize some interesting facts, clean and pre-process the data and find the best performing model to predict the number of crime incidents per day using machine learning.

The models used here are besides just the average, the Bias Effect and repeated Cross Validation to compare the following models: Regression Trees, Linear Regression, Support Vector Machine with radial basis function and K-Nearest-Neighbors. Further an Ensemble prediction will be tested and last but not least a Random Forest model.

Outcome will be the common used RMSE (root mean squared error), which shows the mean difference between the predicted values and the actual ones. For this report the count of crimes per day and police district will be predicted.

2. Pre-processing

Downloading the data and libraries

The San Francisco crime data set includes several files, which won't be used for this prediction. It is going to focus on four dbf-files, which include the values of four different kind of crimes. "Cartheft", "Drugs", "Vandalism" and "Robbery". Additionally a shape file including a street map of San Francisco will be used to visualize the location of crimes on a specific day. To read that shape file the function "shapefile" contained in the "foreign" package is used.

```
# loading and installing the libraries needed to read and wrangle the data
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(caret)) install.packages("caret")
if(!require(raster)) install.packages("raster")
if(!require(foreign)) install.packages("foreign")
if(!require(lubridate)) install.packages("lubridate")
if(!require(dplyr)) install.packages("dplyr")
if(!require(bigreadr)) install.packages("bigreadr")
if(!require(ggplot2)) install.packages("ggplot2")


library(raster)
library(foreign)
library(tidyverse)
library(lubridate)
library(dplyr)
library(caret)
library(bigreadr)
library(ggplot2)



# Download the file from the GeoDa Data and Lab platform. The source of the data is
# the San Francisco Police Department Crime Incident Reporting System.
sf <- tempfile()
download.file("https://geodacenter.github.io/data-and-lab//data/SFCrime_July_Dec2012.zip", sf)

# unzip and observe the containing data to choose the files needed
unzipped <- unzip(sf)

# Store the data frames for the general blocks data and the four different kind of crimes
cartheft <- read.dbf("./SFCrime_July_Dec2012 2/Crime Events/sf_cartheft.dbf")
drugs <- read.dbf("./SFCrime_July_Dec2012 2/Crime Events/sf_drugs.dbf")
vandalism <- read.dbf("./SFCrime_July_Dec2012 2/Crime Events/sf_vandalism.dbf")
robbery <- read.dbf("./SFCrime_July_Dec2012 2/Crime Events/sf_robbery.dbf")

# Store the map of San Francisco
block_map <- shapefile("./SFCrime_July_Dec2012 2/SF PD Plots/SFCrime_blocks.shp")
```

Excursion robberies

The four single crime tables show the distribution of crimes over date and police districts of San Francisco with coordinates. To get an overview we first take a closer look to one of these tables, the robberies. It

contains 2761 observations of 13 variables. There are 45 unique descriptions of crimes. Looking at the ten most and ten less happened crimes, we see robbery on the street with bodily force as most happened incident while bank robbery with a gun happened only very rarely.

```
# Observe the robbery data, 2761 incidents of 45 different categories of description
head(robbery)
```

```
##   IncidntNum    X_pr    Y_pr Category                          Descript
## 1  120516979 5996647 2088056  ROBBERY  ROBBERY ON THE STREET, STRONGARM
## 2  120517096 6018850 2094117  ROBBERY ROBBERY OF A RESIDENCE WITH A GUN
## 3  120517096 6018850 2094117  ROBBERY ROBBERY OF A RESIDENCE WITH A GUN
## 4  120517096 6018850 2094117  ROBBERY ROBBERY OF A RESIDENCE WITH A GUN
## 5  120519058 6009324 2114646  ROBBERY             ROBBERY, BODILY FORCE
## 6  120519064 6010023 2115050  ROBBERY  ROBBERY ON THE STREET, STRONGARM
##   DayOfWeek       Date       Time PdDistrict Resolution              Location
## 1    Sunday 2012-07-01 1899-12-30    TARAVAL       NONE BROAD ST / SAN JOSE AV
## 2    Sunday 2012-07-01 1899-12-30    BAYVIEW       NONE   0 Block of HARBOR RD
## 3    Sunday 2012-07-01 1899-12-30    BAYVIEW       NONE   0 Block of HARBOR RD
## 4    Sunday 2012-07-01 1899-12-30    BAYVIEW       NONE   0 Block of HARBOR RD
## 5    Sunday 2012-07-01 1899-12-30 TENDERLOIN       NONE   GEARY ST / TAYLOR ST
## 6    Sunday 2012-07-01 1899-12-30    CENTRAL       NONE   400 Block of POST ST
##           X        Y
## 1 -122.4535 37.71321
## 2 -122.3771 37.73110
## 3 -122.3771 37.73110
## 4 -122.3771 37.73110
## 5 -122.4115 37.78694
## 6 -122.4091 37.78809
```

```
dim(robbery)
```

```
## [1] 2761   13
```

```
length(unique(robbery$Descript))
```

```
## [1] 45
```

```
# Sort by number of most happened incidents
rob_arranged <- robbery %>%
  group_by(Descript) %>%
  summarise(Count = length(Descript)) %>%
  arrange(desc(Count))
head(rob_arranged, 10)
```

```
## # A tibble: 10 x 2
##    Descript                                  Count
##    <fct>                                     <int>
##  1 ROBBERY ON THE STREET, STRONGARM            863
##  2 ROBBERY, BODILY FORCE                       580
##  3 ROBBERY ON THE STREET WITH A GUN            259
##  4 ROBBERY, ARMED WITH A GUN                   171
```

```
##  5 ATTEMPTED ROBBERY ON THE STREET WITH BODILY FORCE   149
##  6 ROBBERY ON THE STREET WITH A DANGEROUS WEAPON        81
##  7 ROBBERY ON THE STREET WITH A KNIFE                   77
##  8 ROBBERY, ARMED WITH A KNIFE                          76
##  9 ATTEMPTED ROBBERY WITH BODILY FORCE                  65
## 10 ROBBERY OF A CHAIN STORE WITH BODILY FORCE           50
```

```
# See which incidents happens only less. Gun robberies of banks or in houses
rob_less <- robbery %>%
  group_by(Descript) %>%
  summarise(Count = length(Descript)) %>%
  arrange(Count)
head(rob_less, 10)
```

```
## # A tibble: 10 x 2
##    Descript                                          Count
##    <fct>                                             <int>
##  1 ATTEMPTED ROBBERY OF A BANK WITH A GUN                1
##  2 ATTEMPTED ROBBERY RESIDENCE WITH A GUN                1
##  3 ROBBERY OF A BANK WITH A DANGEROUS WEAPON             1
##  4 ROBBERY OF A BANK WITH A GUN                          1
##  5 ROBBERY OF A CHAIN STORE WITH A KNIFE                 1
##  6 ROBBERY OF A SERVICE STATION WITH BODILY FORCE        1
##  7 ATTEMPTED ROBBERY COMM. ESTABLISHMENT WITH A GUN      2
##  8 ATTEMPTED ROBBERY OF A BANK WITH BODILY FORCE         2
##  9 ATTEMPTED ROBBERY RESIDENCE WITH BODILY FORCE         2
## 10 ROBBERY OF A BANK WITH BODILY FORCE                   2
```

In 20 percent of the robberies guns were involved.

```
# Filter for all incidents with guns using the string detect function "grep"
guns <- rob_arranged$Descript[grep("GUN", rob_arranged$Descript)]
gun_rob <- rob_arranged %>% filter(Descript %in% guns)


# In 20 percent of the cases of robbery a gun was involved
sum((gun_rob$Count) / sum(rob_arranged$Count))*100
```

```
## [1] 19.95654
```

Now we take a look at the five most dangerous and the five safest place regarding robberies. When we plot the coordinates of the 100 most happened robberies we can easily see a large accumulation in the North East of San Francisco. We will see this region later on the downloaded map when we plot all crimes of one day.

```
# 5 Locations of the most happening robberies in descending order
rob_location <- robbery %>%
  group_by (Location) %>%
  summarise(Count = length(Location)) %>%
  arrange(desc(Count))
head(rob_location, 5)
```

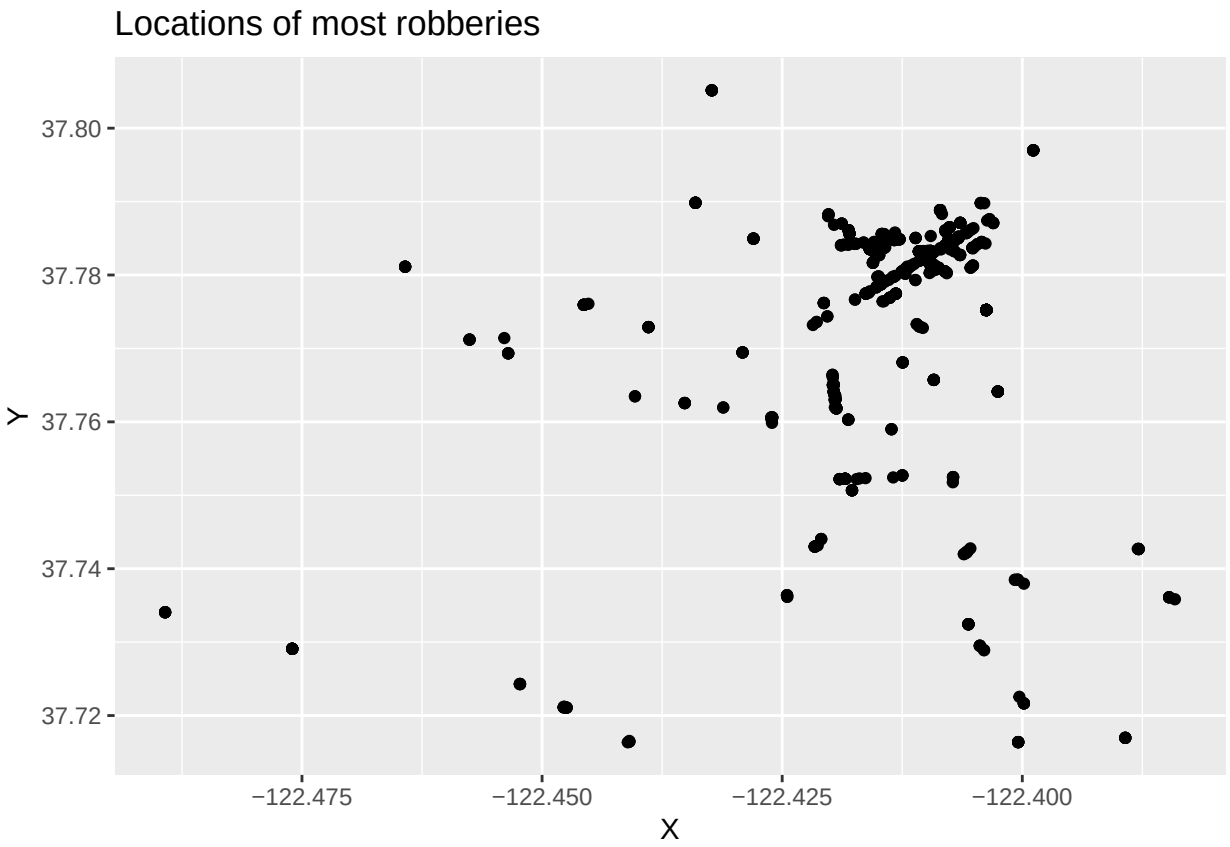
```
## # A tibble: 5 x 2
```

```
##   Location                 Count
##   <fct>                    <int>
## 1 800 Block of BRYANT ST      76
## 2 800 Block of MARKET ST      33
## 3 2000 Block of MISSION ST    21
## 4 400 Block of EDDY ST        19
## 5 900 Block of MARKET ST      16
```

```
# 5 Safest places in San Francisco regarding robberies
rob_location_safe <- robbery %>%
  group_by (Location) %>%
  summarise(Count = length(Location)) %>%
  arrange(Count)
head(rob_location_safe, 5)
```

```
## # A tibble: 5 x 2
##   Location               Count
##   <fct>                  <int>
## 1 0 Block of 4TH ST          1
## 2 0 Block of 8TH ST          1
## 3 0 Block of ALHAMBRA ST     1
## 4 0 Block of BALDWIN CT      1
## 5 0 Block of BAYVIEW ST      1
```

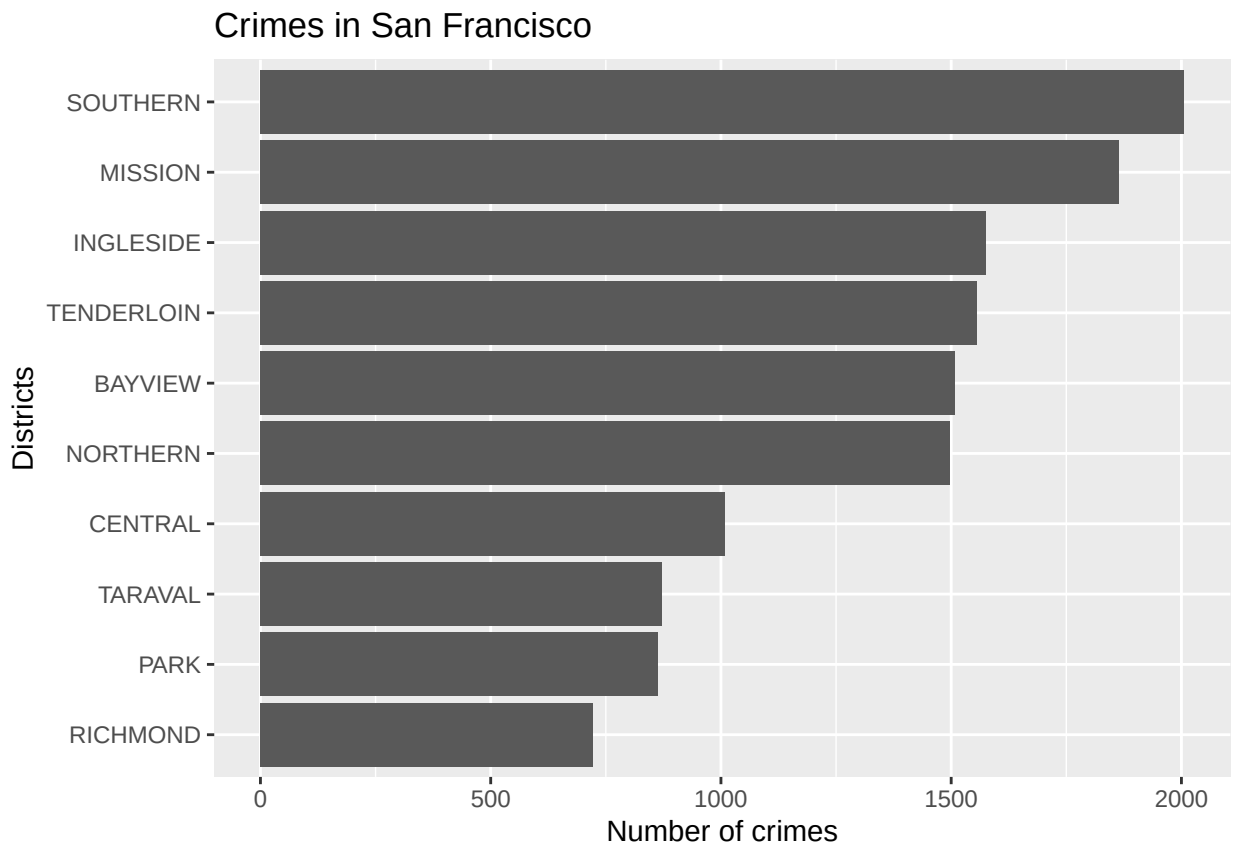
```
# Plotting the coordinates of the 100 most happened robberies - the unsafest places
most_robberies <- head(rob_location, 100) %>%
  left_join(robbery, by = "Location")

most_robberies %>% ggplot(aes(X, Y))+
  geom_point() +
  labs(title = "Locations of most robberies")
```



Create the final data set by combining all four tables

Now we join all crime tables in one and look at the number of general crimes in each Police District. "Southern" and "Mission" are the most dangerous districts in San Francisco.

```
# Joining all crimes in one table using full_join
crimes <- robbery %>%
  full_join(cartheft)
crimes <- crimes %>%
  full_join(drugs)
crimes <- crimes %>%
  full_join(vandalism)

# Plot all districts with their number of crimes
crimes %>% group_by(PdDistrict) %>%
  summarize(Count = length(PdDistrict)) %>%
  arrange(desc(Count)) %>%
  ggplot(aes(x = reorder(PdDistrict, Count), y = Count)) +
  geom_bar(stat = "identity") +
  labs(x ="Districts", y = "Number of crimes", title = "Crimes in San Francisco") +
  coord_flip()
```



When we want to observe the description of crimes we notice, that there's a very large number of unique ones, 106. We're gonna clean this data and combine these 106 into 14 to get a clearer view, and plot the new distribution of crimes to see which incidents happened the most. To replace the old descriptions we use the function "fct_recode" within mutate.

Yes, this is a lot of work and needs a lot of space but it's worth it, as we will be able to use it for some interesting visualizations later.

```
# Very large number of different types of crimes
length(unique(crimes$Descript))
```

```
## [1] 106
```

```
# Cleaning the Descripts of crimes
crime_clean<-crimes %>%
  mutate(Crime = fct_recode(Descript,
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, VANDALISM",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, GRAFFITI",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, VANDALISM OF VEHICLES",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, TIRE SLASHING",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, BREAKING WINDOWS",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, BREAKING WINDOWS WITH BB GUN",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, STREET CARS/BUSES",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, FICTITIOUS PHONE CALLS",
                      "Malicious Mischief" = "MALICIOUS MISCHIEF, BUILDING UNDER CONSTRUCTION",
                      "Robbery with bodily force" = "ROBBERY ON THE STREET, STRONGARM",
```

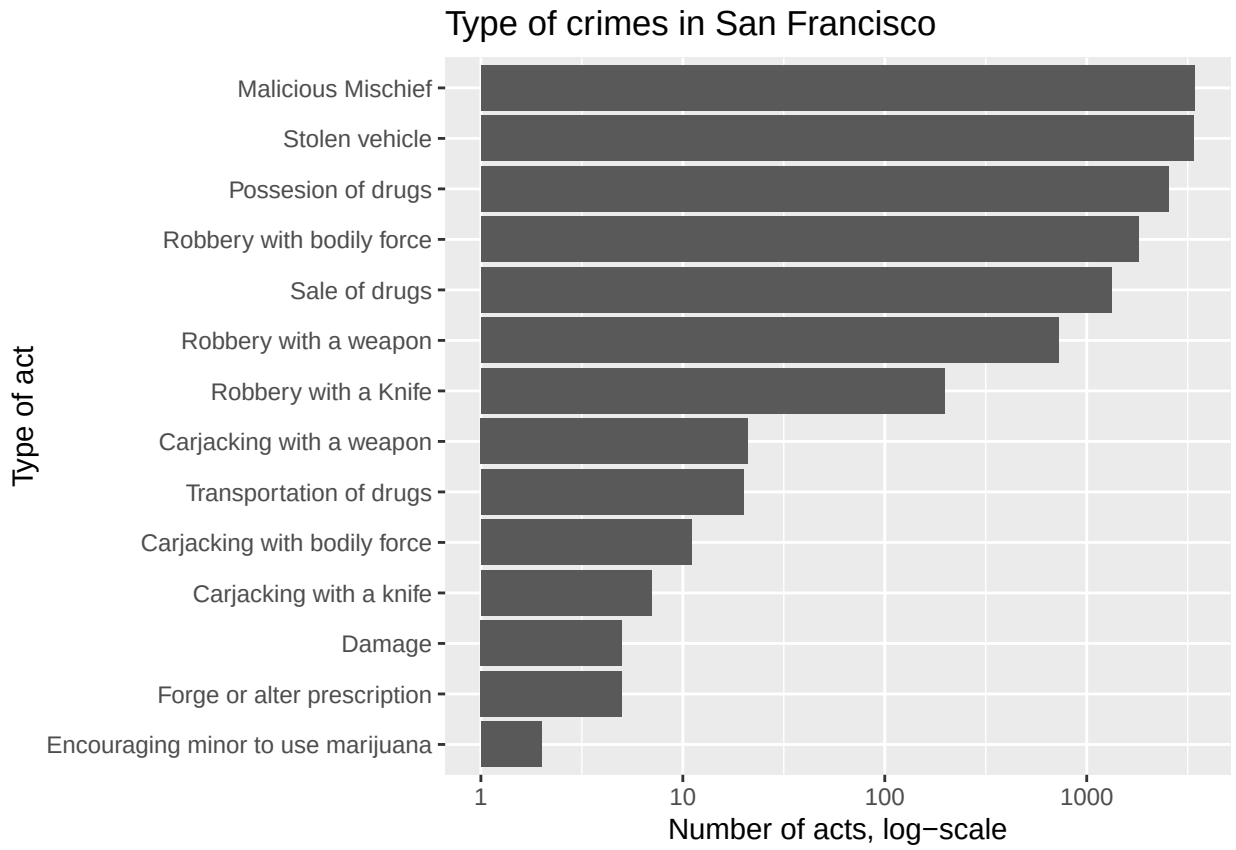
```
                    "Robbery with bodily force" = "ROBBERY, BODILY FORCE",
                    "Robbery with bodily force" = "ROBBERY OF A RESIDENCE WITH BODILY FORCE",
                    "Robbery with bodily force" = "ATTEMPTED ROBBERY CHAIN STORE WITH BODILY FORCE",
                    "Robbery with bodily force" = "ATTEMPTED ROBBERY COMM. ESTAB. WITH BODILY FORCE",
                    "Robbery with bodily force" = "ROBBERY OF A COMMERCIAL ESTABLISHMENT, STRONGARM",
                    "Robbery with bodily force" = "ATTEMPTED ROBBERY WITH BODILY FORCE",
                    "Robbery with bodily force" = "ROBBERY OF A CHAIN STORE WITH BODILY FORCE",
                    "Robbery with bodily force" = "ATTEMPTED ROBBERY OF A BANK WITH BODILY FORCE",
                    "Robbery with bodily force" = "ATTEMPTED ROBBERY RESIDENCE WITH BODILY FORCE",
                    "Robbery with bodily force" = "ROBBERY OF A SERVICE STATION WITH BODILY FORCE",
                    "Robbery with bodily force" = "ROBBERY OF A BANK WITH BODILY FORCE",
                    "Robbery with bodily force" = "ATTEMPTED ROBBERY ON THE STREET WITH BODILY FORCE",
                    "Robbery with a Knife" = "ROBBERY, ARMED WITH A KNIFE",
                    "Robbery with a Knife" = "ROBBERY OF A RESIDENCE WITH A KNIFE",
                    "Robbery with a Knife" = "ROBBERY ON THE STREET WITH A KNIFE",
                    "Robbery with a Knife" = "ROBBERY OF A COMMERCIAL ESTABLISHMENT W/ A KNIFE",
                    "Robbery with a Knife" = "ATTEMPTED ROBBERY ON THE STREET WITH A KNIFE",
                    "Robbery with a Knife" = "ATTEMPTED ROBBERY WITH A KNIFE",
                    "Robbery with a Knife" = "ROBBERY OF A CHAIN STORE WITH A KNIFE",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY RESIDENCE WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY OF A RESIDENCE WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY, ARMED WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY ON THE STREET WITH A GUN",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY ON THE STREET W/DEADLY WEAPON",
                    "Robbery with a weapon" = "ROBBERY, ARMED WITH A DANGEROUS WEAPON",
                    "Robbery with a weapon" = "ROBBERY ON THE STREET WITH A DANGEROUS WEAPON",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY WITH A DEADLY WEAPON",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY ON THE STREET WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY OF A CHAIN STORE WITH A DANGEROUS WEAPON",
                    "Robbery with a weapon" = "ROBBERY OF A COMMERCIAL ESTABLISHMENT WITH A GUN",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY OF A SERVICE STATION WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY OF A CHAIN STORE WITH A GUN",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY CHAIN STORE WITH DEADLY WEAPON",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY COMM. ESTABLISHMENT WITH A GUN",
                    "Robbery with a weapon" = "ROBBERY OF A RESIDENCE WITH A DANGEROUS WEAPON",
                    "Robbery with a weapon" = "ROBBERY OF A COMMERCIAL ESTABLISHMENT W/ WEAPON",
                    "Robbery with a weapon" = "ROBBERY OF A BANK WITH A DANGEROUS WEAPON",
                    "Robbery with a weapon" = "ROBBERY OF A BANK WITH A GUN",
                    "Robbery with a weapon" = "ATTEMPTED ROBBERY OF A BANK WITH A GUN",
                    "Carjacking with bodily force" = "CARJACKING WITH BODILY FORCE",
                    "Carjacking with a knife" = "CARJACKING WITH A KNIFE",
                    "Carjacking with a weapon" = "CARJACKING WITH A DANGEROUS WEAPON",
                    "Carjacking with a weapon" = "CARJACKING WITH A GUN",
                    "Stolen vehicle" = "ATTEMPTED STOLEN VEHICLE",
                    "Stolen vehicle" = "STOLEN TRUCK",
                    "Stolen vehicle" = "STOLEN AND RECOVERED VEHICLE",
                    "Stolen vehicle" = "STOLEN AUTOMOBILE",
                    "Stolen vehicle" = "STOLEN MOTORCYCLE",
                    "Stolen vehicle" = "STOLEN MISCELLANEOUS VEHICLE",
                    "Stolen vehicle" = "STOLEN TRAILER",
                    "Possesion of drugs" = "POSSESSION OF MARIJUANA",
                    "Possesion of drugs" = "POSSESSION OF COCAINE",
```

```
                    "Possesion of drugs" = "POSSESSION OF HEROIN",
                    "Possesion of drugs" = "POSSESSION OF CONTROLLED SUBSTANCE",
                    "Possesion of drugs" = "UNDER INFLUENCE OF DRUGS IN A PUBLIC PLACE",
                    "Possesion of drugs" = "POSSESSION OF BASE/ROCK COCAINE",
                    "Possesion of drugs" = "POSSESSION OF NARCOTICS PARAPHERNALIA",
                    "Possesion of drugs" = "POSSESSION OF METH-AMPHETAMINE",
                    "Possesion of drugs" = "POSSESSION OF METHADONE",
                    "Possesion of drugs" = "POSSESSION OF OPIATES",
                    "Possesion of drugs" = "POSSESSION OF AMPHETAMINE",
                    "Possesion of drugs" = "POSSESSION OF BARBITUATES",
                    "Possesion of drugs" = "POSSESSION OF HALLUCINOGENIC",
                    "Possesion of drugs" = "POSSESSION OF OPIUM DERIVATIVE",
                    "Possesion of drugs" = "FURNISHING MARIJUANA",
                    "Possesion of drugs" = "PLANTING/CULTIVATING MARIJUANA",
                    "Possesion of drugs" = "POSSESSION OF OPIUM",
                    "Encouraging minor to use marijuana" = "ENCOURAGING MINOR TO USE MARIJUANA",
                    "Sale of drugs" = "POSSESSION OF METHADONE FOR SALES",
                    "Sale of drugs" = "POSSESSION OF METH-AMPHETAMINE FOR SALE",
                    "Sale of drugs" = "POSSESSION OF HALLUCINOGENIC FOR SALES",
                    "Sale of drugs" = "POSSESSION OF CONTROLLED SUBSTANCE FOR SALE",
                    "Sale of drugs" = "POSSESSION OF BASE/ROCK COCAINE FOR SALE",
                    "Sale of drugs" = "POSSESSION OF MARIJUANA FOR SALES",
                    "Sale of drugs" = "SALE OF METH-AMPHETAMINE",
                    "Sale of drugs" = "POSSESSION OF COCAINE FOR SALES",
                    "Sale of drugs" = "POSSESSION OF OPIATES FOR SALES",
                    "Sale of drugs" = "POSSESSION OF HEROIN FOR SALES",
                    "Sale of drugs" = "SALE OF MARIJUANA",
                    "Sale of drugs" = "POSSESSION OF AMPHETAMINE FOR SALES",
                    "Sale of drugs" = "SALE OF HEROIN",
                    "Sale of drugs" = "SALE OF METHADONE",
                    "Sale of drugs" = "SALE OF OPIATES",
                    "Sale of drugs" = "SALE OF METHADONE",
                    "Sale of drugs" = "SALE OF OPIATES",
                    "Sale of drugs" = "SALE OF COCAINE",
                    "Sale of drugs" = "SALE OF BASE/ROCK COCAINE",
                    "Sale of drugs" = "SALE OF CONTROLLED SUBSTANCE",
                    "Transportation of drugs" = "TRANSPORTATION OF COCAINE",
                    "Transportation of drugs" = "TRANSPORTATION OF MARIJUANA",
                    "Transportation of drugs" = "TRANSPORTATION OF METH-AMPHETAMINE",
                    "Transportation of drugs" = "TRANSPORTATION OF OPIATES",
                    "Transportation of drugs" = "TRANSPORTAION OF CONTROLLED SUBSTANCE",
                    "Damage" = "DAMAGE TO FIRE ALARM APPARATUS",
                    "Damage" = "DAMAGE/DESTRUCTION OF MAIL",
                    "Damage" = "DAMAGE TO MAIL BOX",
                    "Forge or alter prescription" = "FORGE OR ALTER PRESCRIPTION"))

unique(crime_clean$Crime)
```

```
##  [1] Robbery with bodily force          Robbery with a weapon
##  [3] Carjacking with a weapon           Robbery with a Knife
##  [5] Carjacking with bodily force       Carjacking with a knife
##  [7] Stolen vehicle                     Possesion of drugs
##  [9] Sale of drugs                      Transportation of drugs
```

```
## [11] Forge or alter prescription        Encouraging minor to use marijuana
## [13] Malicious Mischief                 Damage
## 14 Levels: Robbery with bodily force ... Malicious Mischief
```

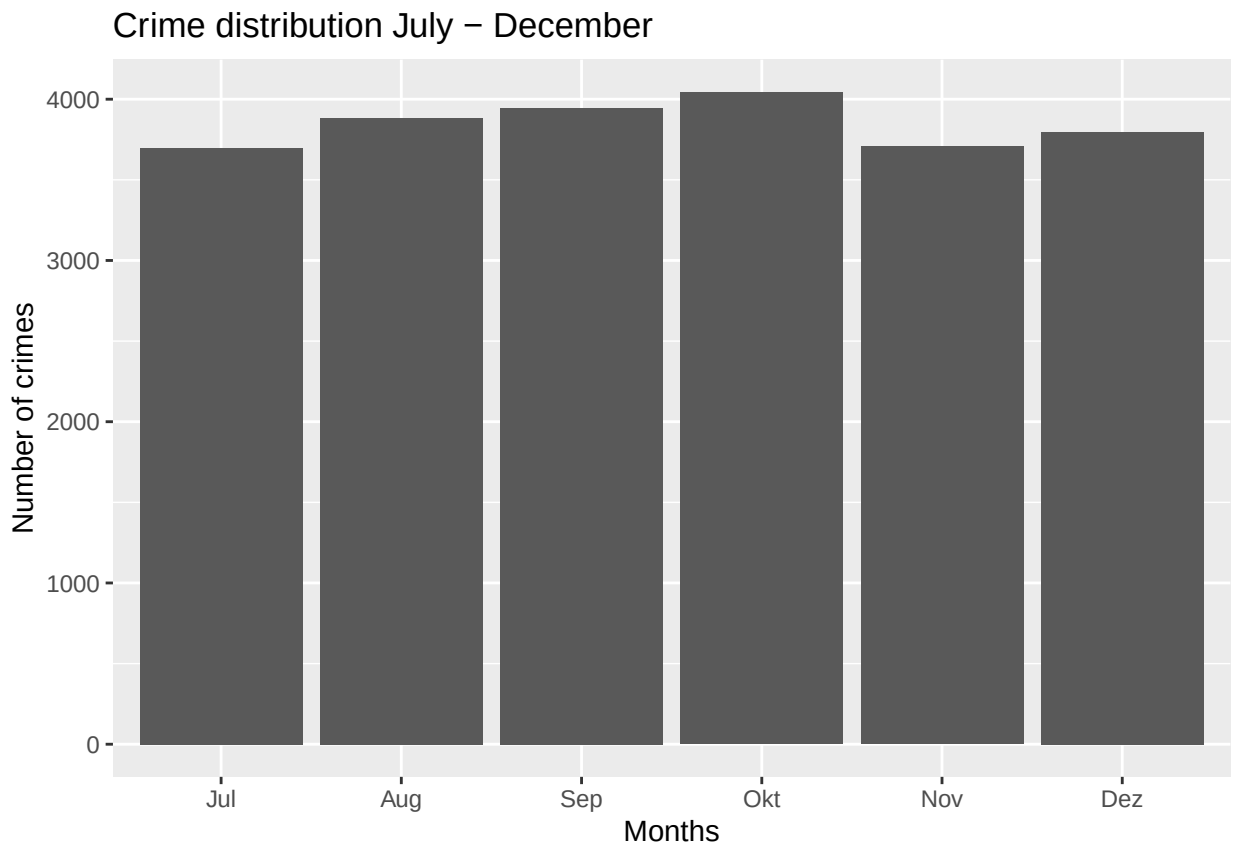
```
# Plot the crimes
crime_clean %>%
  group_by(Crime) %>%
  summarize(Acts = length(Crime)) %>%
  arrange(desc(Acts)) %>%
  ggplot(aes(x = reorder(Crime, Acts), y = Acts)) +
  geom_bar(stat = "identity") +
  scale_y_log10() +
  labs(x ="Type of act", y ="Number of acts, log-scale", title = "Type of crimes in San Francisco") +
  coord_flip()
```



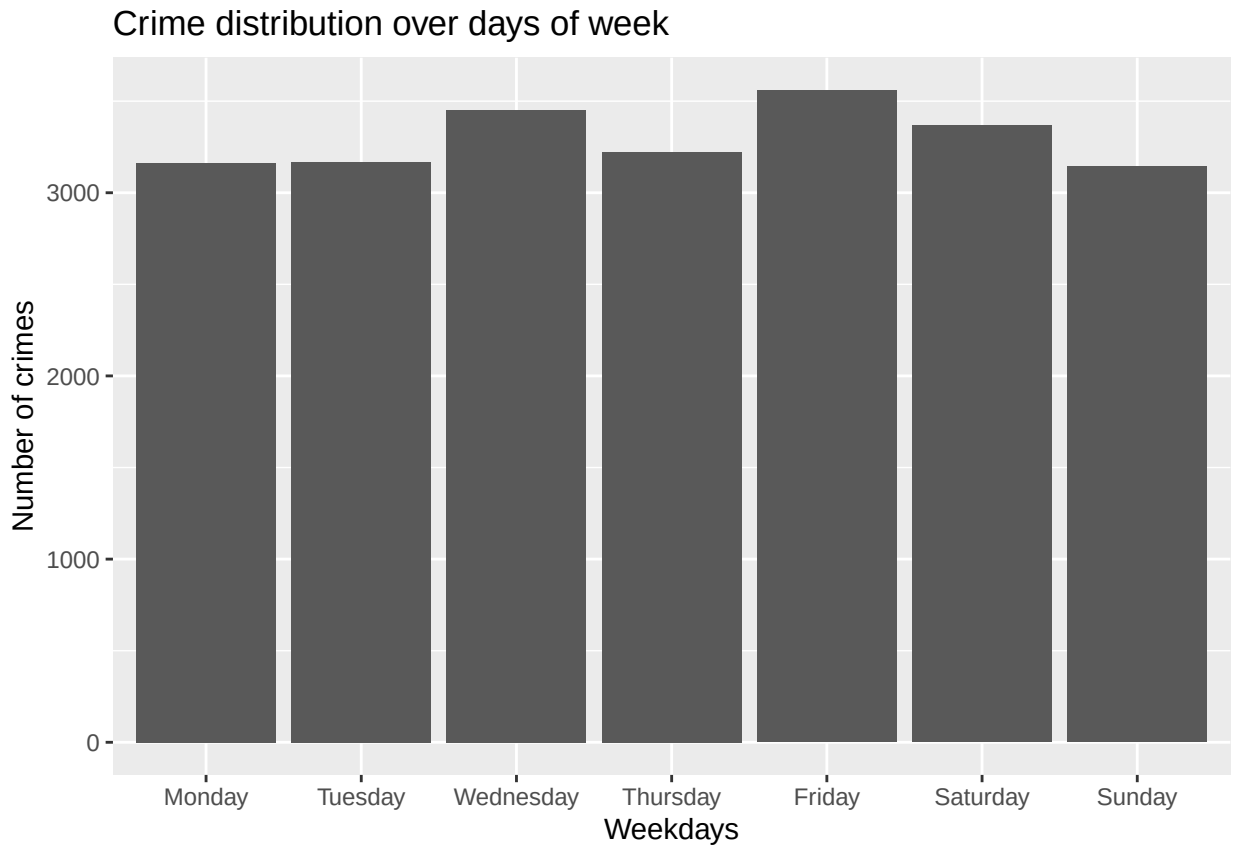
Now the data set will be further optimized by first rejoining with the crimes table to get all columns back. Then we use the month-function of the lubridate package to create a column for the month (July to December), and take a look at the monthly distribution of crimes. After that we look at the crime distribution over different days of the week. An image of crimes vs. weekdays will give an interesting insight of how the different types of crimes appear during the weekdays.

```
# Reunite with the crimes table to get all columns back
sf_crimes <- crime_clean %>%
  left_join(crimes)

# Distribution months
```

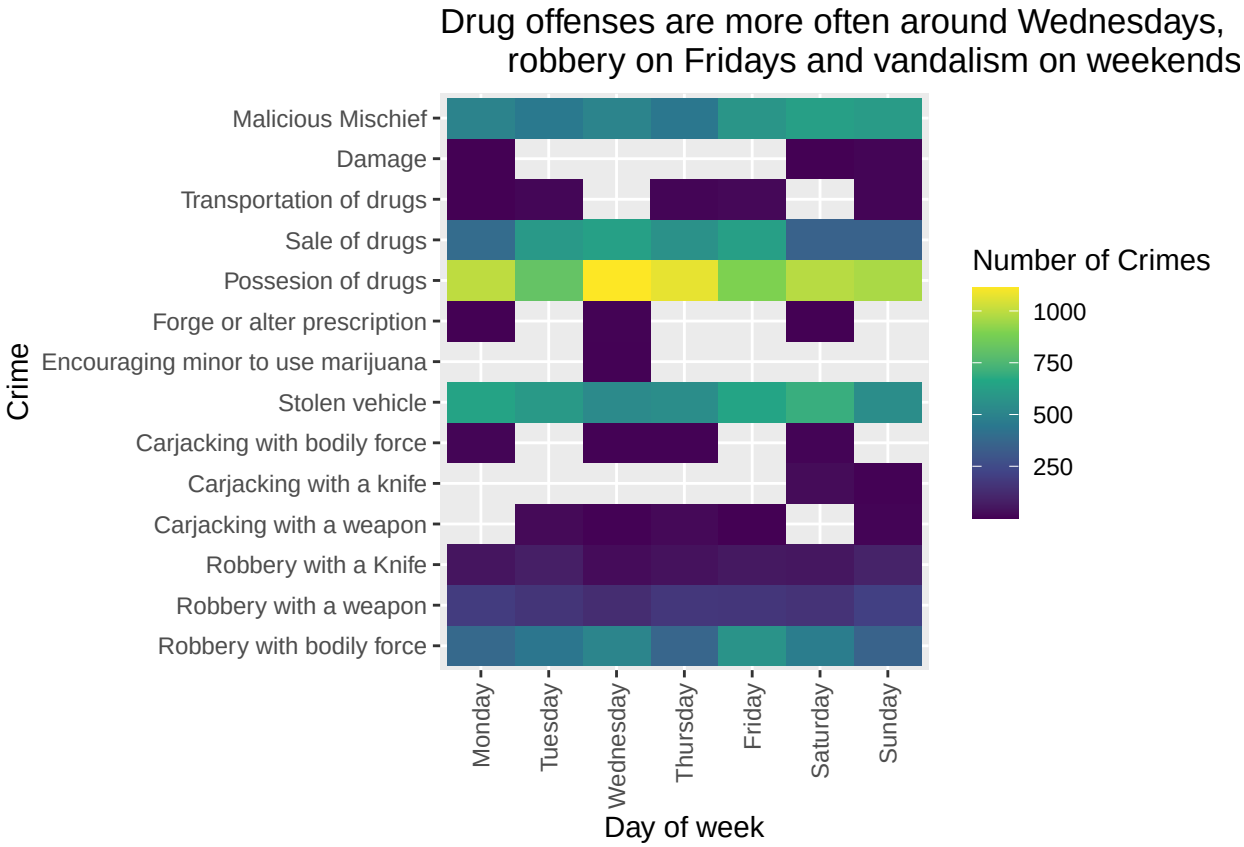
```r
sf_crimes <- sf_crimes %>%
  mutate(Month = month(Date, label=TRUE, abbr=TRUE))

sf_crimes %>% group_by(Month) %>%
  summarize(Count = length(Month)) %>%
  ggplot(aes(Month, Count)) +
  geom_bar(stat = "identity") +
  labs(x = "Months", y = "Number of crimes", title = "Crime distribution July - December")
```




```r
# Distribution weekdays with ordered days
sf_crimes %>%
  group_by(DayOfWeek) %>%
  summarize(Count = length(DayOfWeek)) %>%
  ggplot(aes(ordered(DayOfWeek,
                     levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
                                "Saturday", "Sunday")), Count)) +
  geom_bar(stat = "identity") +
  labs(x = "Weekdays", y = "Number of crimes",
       title = "Crime distribution over days of week")
```



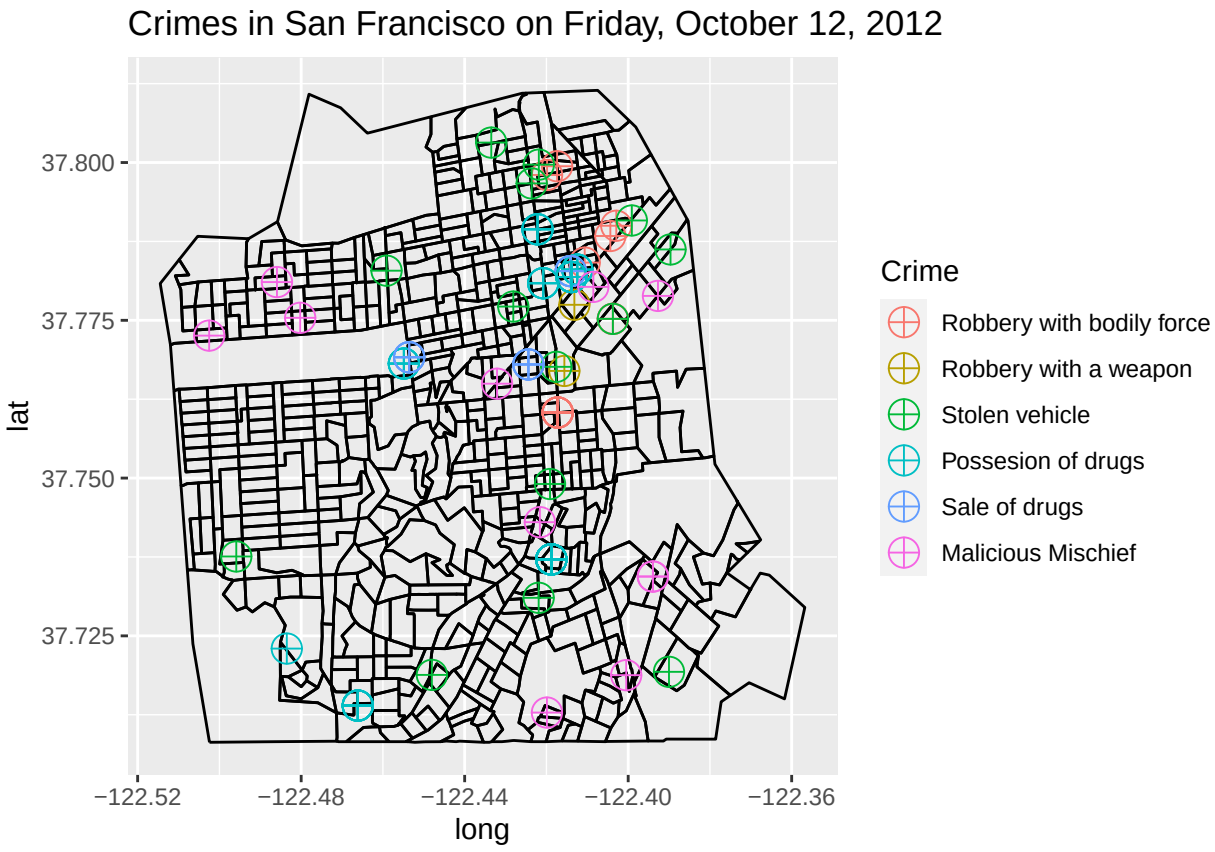

```
# Crimes per day
sf_crimes %>%
  group_by(Crime, DayOfWeek) %>%
  summarise(count=n()) %>%
  ggplot(aes(x = Crime,
             y = ordered(DayOfWeek, levels = c("Monday", "Tuesday", "Wednesday",
                                               "Thursday", "Friday",
                                               "Saturday", "Sunday")))) +
  geom_tile(aes(fill = count)) +
  labs(x = "Crime", y = "Day of week", title = "Drug offenses are more often around Wednesdays,
       robbery on Fridays and vandalism on weekends") +
  scale_fill_viridis_c("Number of Crimes") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  coord_flip()
```



As we saw that more crimes happened on Fridays and in October, we're gonna filter the data for one random Friday in October and plot the coordinates of these crimes onto the map. To join the coordinates of the Friday data frame to the shape file, we first have to convert the map, which is in SpatialPoints class into a data.frame. Spatial Points means a matrix with two columns for coordinates, long and lat. The function fortify.SpatialPolygons converts this to a data frame.

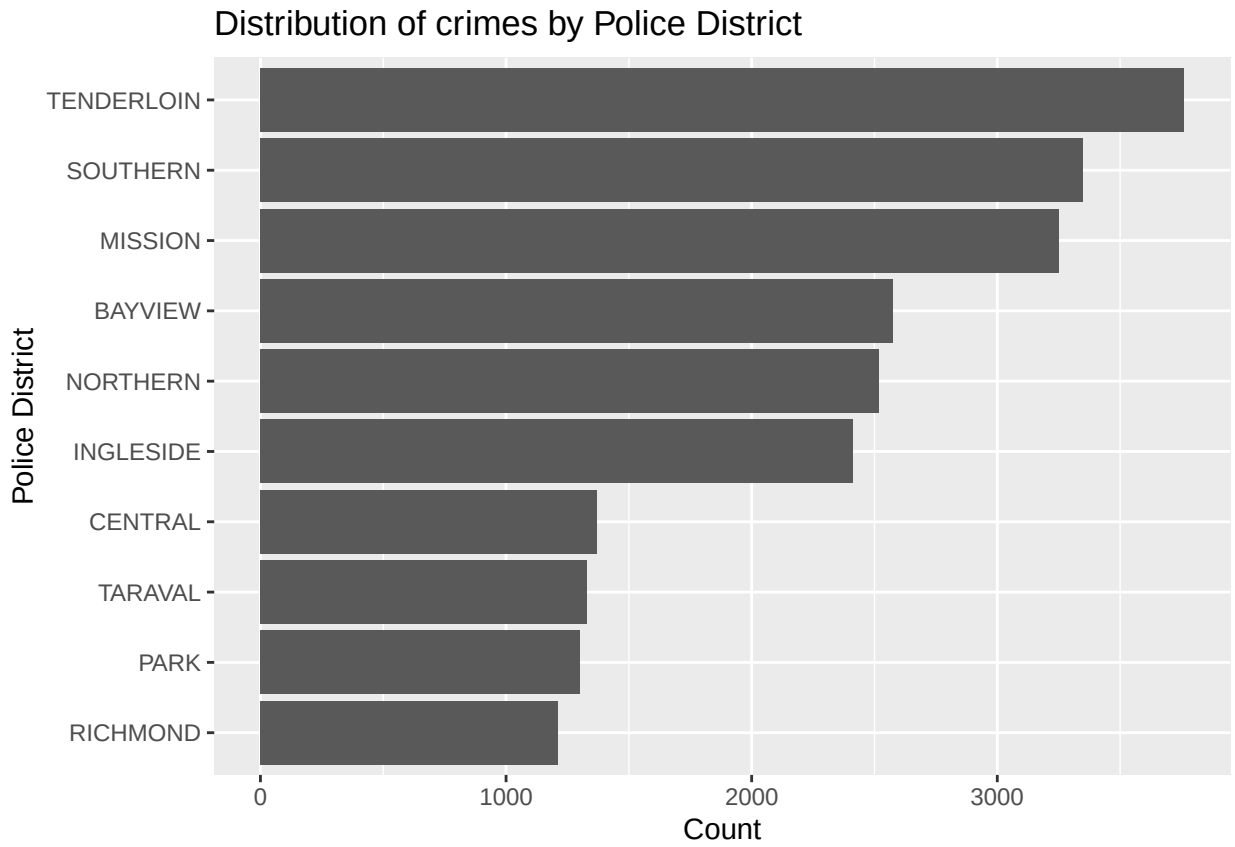
```
# San Francisco map
plot(block_map)
```

```r
# Filter only crimes on Friday, 12th of October
Friday <- sf_crimes %>%
  filter(Date == "2012-10-12")

# Convert the spatial points matrix of the map into a data frame to use it with ggplot
# Source of the function("https://raw.githubusercontent.com/tidyverse/ggplot2/master/R/fortify-spatial.
fortify.SpatialPolygons <- function(model, data, ...) {rbind_df(lapply(model@polygons, fortify))}

block_map_df<-fortify.SpatialPolygons(block_map)

# Show in all crimes on the map that happened on Friday, 12th of October
ggplot(block_map_df, aes(x = long, y = lat))+
  geom_path(aes(group = group))+
  geom_point(data = Friday, aes(x = X, y = Y, color = Crime), pch = 10, size = 5, show.legend = TRUE)+
  labs(title = "Crimes in San Francisco on Friday, October 12, 2012")
```



The most crimes happen in the police districts “Tenderloin”, “Southern” and “Mission”.

```
# In some Police Districts happen more crimes than in others
sf_crimes %>% group_by(PdDistrict) %>%
  summarise(Count = length(PdDistrict)) %>%
  ggplot(aes(x = reorder(PdDistrict, Count), Count))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(x = "Police District", title = "Distribution of crimes by Police District")
```



Now we are curious to know the resolution rate of the San Francisco Police Department. To calculate it, and to finally get our data set for the prediction we're gonna do several steps.

First we replace the resolutions by simply the factor "yes" or "no", so we can calculate the rate later. Then we will group the data to create a column which counts the crimes. For a reasonable prediction of crimes, we choose to predict by day and police district. We create a data frame "Dist_Date" with all possible combinations of districts and dates. Then we use the grouped crime data (with Counts) to group by District and Date to find the total number of crimes per day and District and finally put both together to our final set.

After removing empty rows we are able to calculate the resolution rate of about 36%.

```
# For being able to count the resolution as simply resolved yes or no we modify the 13 levels of Resolu
sf_crimes <- sf_crimes %>%
  mutate(Resolved = fct_recode(Resolution,
                               "N" = "NONE",
                               "Y" = "ARREST, BOOKED",
                               "Y" = "JUVENILE BOOKED",
                               "N" = "UNFOUNDED",
                               "Y" = "JUVENILE ADMONISHED",
                               "N" = "DISTRICT ATTORNEY REFUSES TO PROSECUTE",
                               "Y" = "ARREST, CITED",
                               "Y" = "JUVENILE CITED",
                               "Y" = "EXCEPTIONAL CLEARANCE",
                               "Y" = "PSYCHOPATHIC CASE",
                               "N" = "NOT PROSECUTED",
                               "N" = "COMPLAINANT REFUSES TO PROSECUTE",
```

```
                              "Y" = "CLEARED-CONTACT JUVENILE FOR MORE INFO"))

# Group data to count the crimes
grouped_crimes <- group_by(sf_crimes, Descript, Resolved, PdDistrict,
                           Date, X ,Y, Time, DayOfWeek, Month, Crime) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))

# Unique Police Districts
PdDis <- sf_crimes %>%
  distinct(PdDistrict) %>%
  arrange(PdDistrict)

# Unique dates
dates <- sf_crimes %>%
  distinct(Date) %>%
  arrange(Date)

# Create a data frame with every possible combination of Districts and Dates
Dist_Date <- expand_grid(PdDis, dates) %>%
  arrange(PdDistrict)
head(Dist_Date)
```

```
## # A tibble: 6 x 2
##   PdDistrict Date
##   <fct>      <date>
## 1 BAYVIEW    2012-07-01
## 2 BAYVIEW    2012-07-02
## 3 BAYVIEW    2012-07-03
## 4 BAYVIEW    2012-07-04
## 5 BAYVIEW    2012-07-05
## 6 BAYVIEW    2012-07-06
```

```
# Use the grouped crime data (with Counts) to group by District and Date
# to find the total number of crimes per day and District
crimes_Dist_Date <- grouped_crimes %>%
  group_by(PdDistrict, Date) %>%
  summarize(Count = sum(Count, na.rm = TRUE),
            Resolved = sum(as.numeric(Resolved)-1, na.rm = TRUE),
            Month = Month, DayOfWeek = DayOfWeek, Crime = Crime)

# Put both data sets together to get the modeling data set
dataset <- Dist_Date %>%
  left_join(crimes_Dist_Date, keep=TRUE)

# Remove double columns
dataset <- dataset %>%
  mutate(Date = Date.y, PdDistrict = PdDistrict.y) %>%
  dplyr::select(-PdDistrict.x, -PdDistrict.y, -Date.x, -Date.y)


# 34 empty lines were produced
sum(is.na(dataset$DayOfWeek))
```

```
## [1] 34
```

```
which(is.na(dataset$Month))
```

```
##  [1]  316 1279 2013 4606 5258 5783 6075 6087 6317 6369 6454 6517 6605 6671 6692
## [16] 6702 6761 6767 6781 6951 7000 7001 7046 8638 9144 9184 9223 9230 9265 9281
## [31] 9300 9410 9464 9959
```

```
Show_empty_rows <- dataset[c(316, 1279, 2013, 4606, 5258, 5783, 6075, 6087, 6317, 6369, 6454, 6517,
                  6605, 6671, 6692, 6702, 6761, 6767, 6781, 6951, 7000, 7001, 7046, 8638,
                  9144, 9184, 9223, 9230, 9265, 9281, 9300, 9410, 9464, 9959),]
Show_empty_rows
```

```
## # A tibble: 34 x 7
##    Count Resolved Month DayOfWeek Crime Date   PdDistrict
##    <int>    <dbl> <ord> <fct>     <fct> <date> <fct>
##  1    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  2    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  3    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  4    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  5    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  6    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  7    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  8    NA       NA <NA>  <NA>      <NA>  NA     <NA>
##  9    NA       NA <NA>  <NA>      <NA>  NA     <NA>
## 10    NA       NA <NA>  <NA>      <NA>  NA     <NA>
## # ... with 24 more rows
```

```
#Remove these empty rows from the dataset
dataset <- dataset[-c(316, 1279, 2013, 4606, 5258, 5783, 6075, 6087, 6317, 6369, 6454, 6517,
                      6605, 6671, 6692, 6702, 6761, 6767, 6781, 6951, 7000, 7001, 7046, 8638,
                      9144, 9184, 9223, 9230, 9265, 9281, 9300, 9410, 9464, 9959),]

# The resolution rate of San Francisco police it at around 36%
(sum(dataset$Resolved)/sum(dataset$Count))*100
```

```
## [1] 36.0164
```

To start the training the data set will now be separated into a training and a validation set, which will be 10% of the data set and only be used with the final best performing model to get the final RMSE.

To test the training set, this will further be separated into a train (80%) and a test set (20%).

```
# Make a data partition for validation set (10%)
set.seed(7, sample.kind = "Rounding")

validation_index <- createDataPartition(dataset$Count, times = 1, p = 0.1, list = FALSE)
training_set <- dataset %>% slice(-validation_index)
validation_set <- dataset %>% slice(validation_index)

# Splitting a test set from the training_set (20%)
set.seed(7, sample.kind = "Rounding")
```

```
test_index <- createDataPartition(training_set$Count, times = 1, p = 0.2, list = FALSE)
test_set <- training_set %>% slice(test_index)
train_set <- training_set %>% slice(-test_index)
```

3. Methods and modelling (Training section)

This section contains 5 different approaches to train 8 different models. These are first just the "Average", then the "Bias" effect, means the effect the different variables like Police District, Month or Day have on the Count of crimes. This is because we saw earlier, that for example in different districts are more or less crimes happening. Taken these variations into account we will be able to improve the prediction.

Next repeated Cross Validation will be used to compare the following four models: "Regression trees", "Linear regression", "Svm", which is Support Vector Machine with, in this case, radial basis function and fourth, "K-Nearest-Neighbors".

After that we will try an "Ensemble" prediction of the four models above.

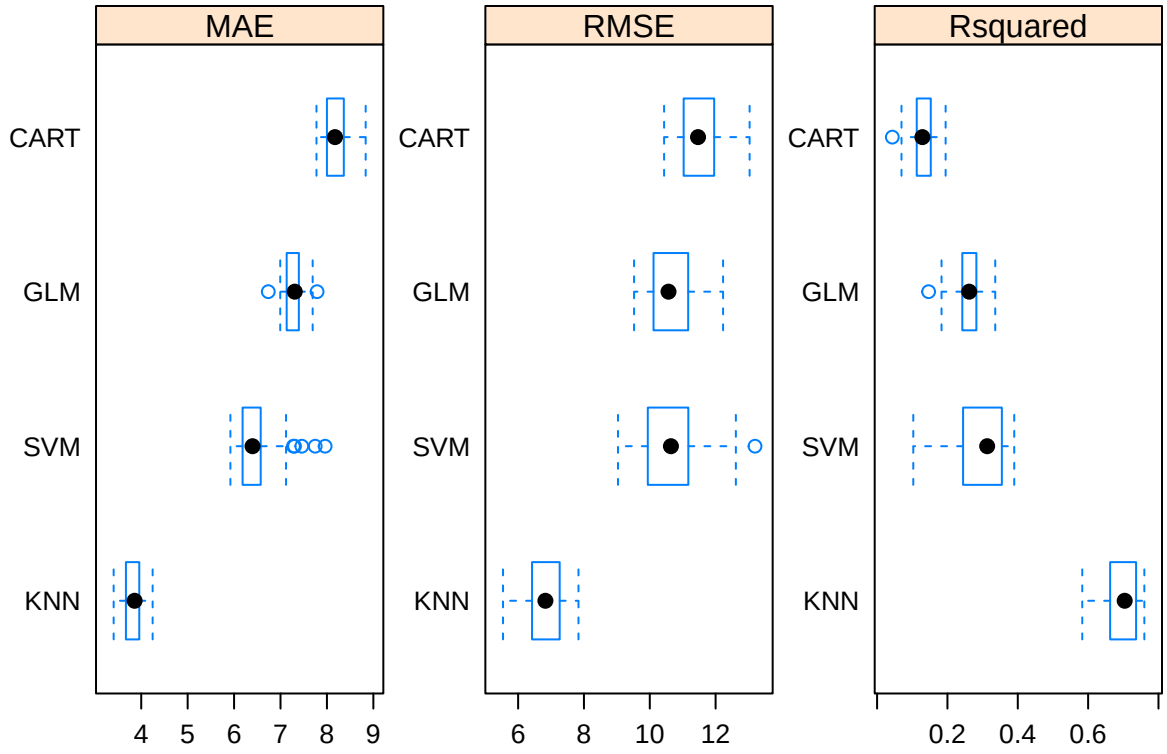
Last but not least we use "Random Forest". This model operates by constructing a multitude of decision trees. It is very large and takes about ten minutes to train but is successful and can be improved a lot by adjusting the tuning parameters. To find the best working parameters we could use train control, but using this function exceeds the aimed temporal scope, hence after testing several values, I chose a number of 80 trees and an mtry value of 100 which show a good performance. The mtry value sets the number of variable splits within each decision tree.

```
# 3. Methods and TRAINING SECTION (5 approaches)

# 3.1. First model: Just the average number of crimes happening in San Francisco per day
Count_avg <- mean(train_set$Count)


# 3.2. Second model: Vector bias which has an effect on the count of crimes
# Effect of police districts
Pd_effect <- train_set %>% group_by(PdDistrict) %>%
  summarize(Pd_effect = mean(Count - Count_avg))

# Effect of weekdays
Day_effect <- train_set %>% group_by(DayOfWeek) %>%
  summarize(Day_effect = mean(Count - Count_avg))

#Effect of crimes as some crimes happen more often than others
Crime_effect <- train_set %>% group_by(Crime) %>%
  summarize(Crime_effect = mean(Count - Count_avg))

# Month effect
Month_effect <- train_set %>% group_by(Month) %>%
  summarize(Month_effect = mean(Count - Count_avg))

# Predict the count of crimes regarding the bias effects on the test set
predictions <- test_set %>%
  left_join(Pd_effect, by = "PdDistrict") %>%
  left_join(Day_effect, by = "DayOfWeek") %>%
  left_join(Crime_effect, by = "Crime") %>%
  left_join(Month_effect, by = "Month") %>%
```

```
  mutate(predictions = Count_avg + Pd_effect + Day_effect + Crime_effect + Month_effect)
fit_bias <- predictions$predictions


# 3.3. Cross validation on 4 different machine learning models with 10 folds and 3 repeats as configura

# prepare training scheme
control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

# GLM
set.seed(7, sample.kind = "Rounding")
fit_glm <- train(Count ~ ., method = "glm", data = train_set, trControl = control)

# CART
set.seed(7, sample.kind = "Rounding")
fit_cart <- train(Count ~ ., method = "rpart", data = train_set, trControl = control)

# SVM (takes several minutes to run, 4.2 MB)
set.seed(7, sample.kind = "Rounding")
fit_svm <- train(Count ~ ., method = "svmRadial", data = train_set, trControl = control)

# kNN
set.seed(7, sample.kind = "Rounding")
fit_knn <- train(Count ~ ., method = "knn", data = train_set, trControl = control)

# collect resamples
results <- resamples(list(GLM = fit_glm,
                          CART = fit_cart, SVM = fit_svm,
                          KNN = fit_knn))
results
```

```
##
## Call:
## resamples.default(x = list(GLM = fit_glm, CART = fit_cart, SVM = fit_svm, KNN
##  = fit_knn))
##
## Models: GLM, CART, SVM, KNN
## Number of resamples: 30
## Performance metrics: MAE, RMSE, Rsquared
## Time estimates for: everything, final model fit
```

```
# box and whisker plots to compare models
scales <- list(x=list(relation="free"), y=list(relation="free"))
bwplot(results, scales=scales)
```




```
# 3.4. An ensemble model
models <- c("glm", "rpart", "svmRadial", "knn")

set.seed(1, sample.kind = "Rounding")
fit_ensemble <- lapply(models, function(model){
  print(model)
  train(Count ~ ., method = model, data = train_set)
})
```

```
## [1] "glm"
## [1] "rpart"
## [1] "svmRadial"
## [1] "knn"
```

```
names(fit_ensemble) <- models


# 3.5. Random Forest with tuning values of 80 trees and an mtry of 100
set.seed(1, sample.kind = "Rounding")
# This codes takes about 10 minutes
fit_rf <- train(Count ~ .,
                  method = "rf",
                  data = train_set,
                  ntree = 80,
                  tuneGrid = expand.grid(.mtry = 100))
```

4. Analysis and results

Now we're gonna see how good the trained models work by calculating the RMSE for each and show it in a table

```
# Making predictions with the trained models

# Average
rmse_avg <- RMSE(Count_avg, test_set$Count)

# Bias
rmse_bias <- RMSE(test_set$Count, predictions$predictions)

# GLM
pred_glm <- predict(fit_glm, test_set)
rmse_glm <- RMSE(test_set$Count, pred_glm)

# CART
pred_cart <- predict(fit_cart, test_set)
rmse_cart <- RMSE(test_set$Count, pred_cart)

# SVM
pred_svm <- predict(fit_svm, test_set)
rmse_svm <- RMSE(test_set$Count, pred_svm)

# KNN
pred_knn <- predict(fit_knn, test_set)
rmse_knn <- RMSE(test_set$Count, pred_knn)

# Ensemble
pred_ensemble <- sapply(fit_ensemble, function(object)
  predict(object, newdata = test_set))
rmse_ensemble <- RMSE(test_set$Count, pred_ensemble)

# RF
pred_rf <- predict(fit_rf, test_set)
rmse_rf <- RMSE(test_set$Count, pred_rf)



# result table for RMSEs
RMSE_results <- data_frame(method = c("Average", "Bias", "Regression trees", "Linear regression",
                                      "Svm", "K-Nearest-Neighbors", "Ensemble", "Random Forest"),
                           RMSE = c(rmse_avg, rmse_bias, rmse_cart, rmse_glm, rmse_svm, rmse_knn,
                                    rmse_ensemble, rmse_rf))
RMSE_results
```

```
## # A tibble: 8 x 2
##   method               RMSE
##   <chr>               <dbl>
## 1 Average             11.3
## 2 Bias                10.0
## 3 Regression trees    10.3
## 4 Linear regression    9.48
```

```
## 5 Svm                   9.05
## 6 K-Nearest-Neighbors  6.54
## 7 Ensemble             8.96
## 8 Random Forest        3.72
```

Final test and conclusion

We see, that the Random Forest is the best performing model, so now we will test it on the validation set to see the achieved RMSE.

```
# Test the best model on the validation set
Validation_pred <- predict(fit_rf, validation_set)
Final_result <- RMSE(validation_set$Count, Validation_pred)
Final_result
```

```
## [1] 4.091835
```

```
# See the maximum count of crimes per day and district
max <- max(validation_set$Count)
max
```

```
## [1] 78
```

Our model reaches a RMSE of about 4.09, which means an error prediction of only 4 crimes per day and police district, which can be rated as a good performing model, as in every district happened up to 78 crimes per day. Regarding the small set with only data of 6 months, this is a satisfying prediction, as less values lead to larger errors.

Crime forecasting using predictive machine learning algorithms is nowadays used by Police Departments all over the world to assist in preventing and solving criminal cases. With a large number of methods for predicting this section of data analysis is permanent increasing and improving.

Scenarios like in movies, where perpetrators can be arrested even before they commit the crime will surely remain part of some producers' fantasy, because even best performing machine learning models are no prophets and can only show tendencies and probabilities but no facts.

Nonetheless these predictions are and will always be a very useful assist to global crime preventing and will probably further improve their performance.