KC_analysis

2024-03-20

Load in Data

KC School Data

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.3.2
```

```
# Specify the file path
file_path <- "C:/Users/jans7/OneDrive - Marquette University/SP24/COSC 6510 - Data Intelligence/KC_Schoo

# Read in the Excel file
school_data <- read_excel(file_path)

# View the first few rows of the data
(school_data)
```

```
## # A tibble: 36 x 8
##    Level     SchoolName           Address City  State ZipCode Latitude Longitude
##    <chr>     <chr>                <chr>   <chr> <chr>   <dbl>    <dbl>     <dbl>
##  1 Secondary African-Centered Co~ 3500 E~ Kans~ MO      64132     39.0     -94.5
##  2 Primary   African-Centered Pr~ 6410 S~ Kans~ MO      64130     39.0     -94.6
##  3 Primary   Benjamin Banneker E~ 7050 A~ Kans~ MO      64132     39.0     -94.5
##  4 Primary   Border Star Montess~ 6321 W~ Kans~ MO      64113     39.0     -94.6
##  5 Secondary Central High School  3221 I~ Kans~ MO      64128     39.1     -94.5
##  6 Secondary Central Middle Scho~ 3611 L~ Kans~ MO      64128     39.1     -94.5
##  7 Secondary East High School     1924 V~ Kans~ MO      64127     39.1     -94.5
##  8 Primary   Faxon Elementary Sc~ 1320 E~ Kans~ MO      64109     39.1     -94.6
##  9 Primary   Foreign Language Ac~ 3450 W~ Kans~ MO      64111     39.1     -94.6
## 10 Primary   Garfield Elementary~ 436 Pr~ Kans~ MO      64124     39.1     -94.6
## # i 26 more rows
```

```
summary(school_data)
```

```
##     Level            SchoolName          Address              City
##  Length:36          Length:36          Length:36          Length:36
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##     State              ZipCode         Latitude       Longitude
##  Length:36          Min.   :64106   Min.   :38.98   Min.   :-94.60
##  Class :character   1st Qu.:64111   1st Qu.:39.04   1st Qu.:-94.57
##  Mode  :character   Median :64125   Median :39.07   Median :-94.55
##                     Mean   :64121   Mean   :39.06   Mean   :-94.55
##                     3rd Qu.:64129   3rd Qu.:39.09   3rd Qu.:-94.53
##                     Max.   :64133   Max.   :39.12   Max.   :-94.46
```

KC Crime Data

```
# Specify the file path
file_path <- "C:/Users/jans7/OneDrive - Marquette University/SP24/COSC 6510 - Data Intelligence/crimeda

crime_data_clean <- read.csv(file_path)

# View the first few rows of the data
head(crime_data_clean)
```

```
##   X Reported_Date          Description                      Address        City
## 1 1    03/06/2015       Misc Violation    BROADWAY and WESTPORT RD KANSAS CITY
## 2 2    09/21/2015 Aggravated Assault (                        <NA>        <NA>
## 3 3    09/21/2015       Family Offense                        <NA>        <NA>
## 4 4    09/08/2015           Auto Theft PROSPECT AV and E TRUMAN RD KANSAS CITY
## 5 5    05/19/2015 Possession/Sale/Dist    VICTOR ST and WALROND AV KANSAS CITY
## 6 6    08/31/2015 Non Aggravated Assau       PASEO and E TRUMAN RD KANSAS CITY
##   Zip.Code Rep_Dist Area Age Latitude Longitude       Date
## 1    64131   PJ3229  CPD  NA  38.9767  -94.5767 2015-03-06
## 2    99999     <NA> <NA>  NA       NA        NA 2015-09-21
## 3    99999     <NA> <NA>  NA       NA        NA 2015-09-21
## 4    64126   PJ7474  EPD  NA  39.0947  -94.5516 2015-09-08
## 5    64128   PJ2340  EPD  NA  39.0735  -94.5461 2015-05-19
## 6    61109   PJ1326  CPD  29       NA        NA 2015-08-31
```

Analyzing Crime Data

```
summary(crime_data_clean)
```

```
##        X           Reported_Date      Description          Address
##  Min.   :      1   Length:1039773     Length:1039773     Length:1039773
##  1st Qu.: 259944   Class :character   Class :character   Class :character
##  Median : 519887   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 519910
##  3rd Qu.: 779830
##  Max.   :1039901
##
##      City              Zip.Code          Rep_Dist             Area
##  Length:1039773     Min.   :     5301   Length:1039773     Length:1039773
##  Class :character   1st Qu.:    64112   Class :character   Class :character
```

```
##  Mode  :character   Median :    64127   Mode  :character   Mode  :character
##                     Mean   :    67239
##                     3rd Qu.:    64133
##                     Max.   :641303016
##                     NA's   :44615
##       Age            Latitude        Longitude          Date
##  Min.   : 17.0    Min.   :38.65    Min.   :-94.94   Length:1039773
##  1st Qu.: 26.0    1st Qu.:39.02    1st Qu.:-94.58   Class :character
##  Median : 35.0    Median :39.07    Median :-94.56   Mode  :character
##  Mean   : 37.7    Mean   :39.07    Mean   :-94.55
##  3rd Qu.: 47.0    3rd Qu.:39.11    3rd Qu.:-94.52
##  Max.   :100.0    Max.   :39.89    Max.   :-94.07
##  NA's   :372410   NA's   :141833   NA's   :141833
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.2
```

```
## Linking to GEOS 3.11.2, GDAL 3.7.2, PROJ 9.3.0; sf_use_s2() is TRUE
```

```
library(ggplot2)
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.3.3
```

```
library(sp)
```

```
## Warning: package 'sp' was built under R version 4.3.2
```

```r
library(geosphere)
```

```
## Warning: package 'geosphere' was built under R version 4.3.2
```

Has Crime Increased or Decreased?

First seeing if total crime has increased or decreased from year to year

```r
crime_data <- crime_data_clean

# Extract year from the Date column
crime_data$Year <- as.integer(format(as.Date(crime_data$Date), "%Y"))

# Calculate the increase in crime over the years
crime_increase <- crime_data %>%
  group_by(Year) %>%
  summarise(total_crime = n()) %>%
  mutate(crime_increase = total_crime - lag(total_crime))

# Print the increase in crime over the years
print(crime_increase)
```

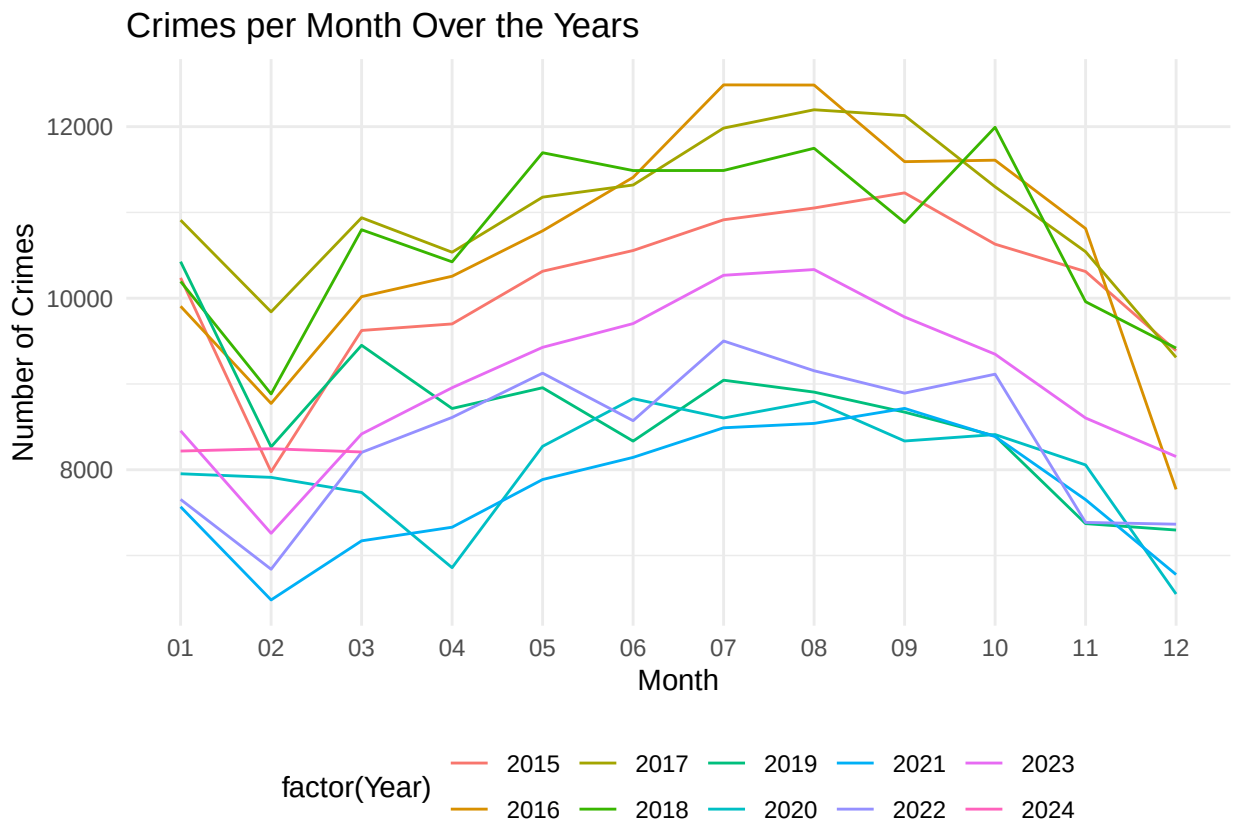
```
## # A tibble: 10 x 3
##     Year total_crime crime_increase
##    <int>       <int>          <int>
##  1  2015      121931             NA
##  2  2016      127903           5972
##  3  2017      132183           4280
##  4  2018      128974          -3209
##  5  2019      103832         -25142
##  6  2020       96312          -7520
##  7  2021       93141          -3171
##  8  2022      100417           7276
##  9  2023      108703           8286
## 10  2024       26377         -82326
```

Note 2024 is not complete

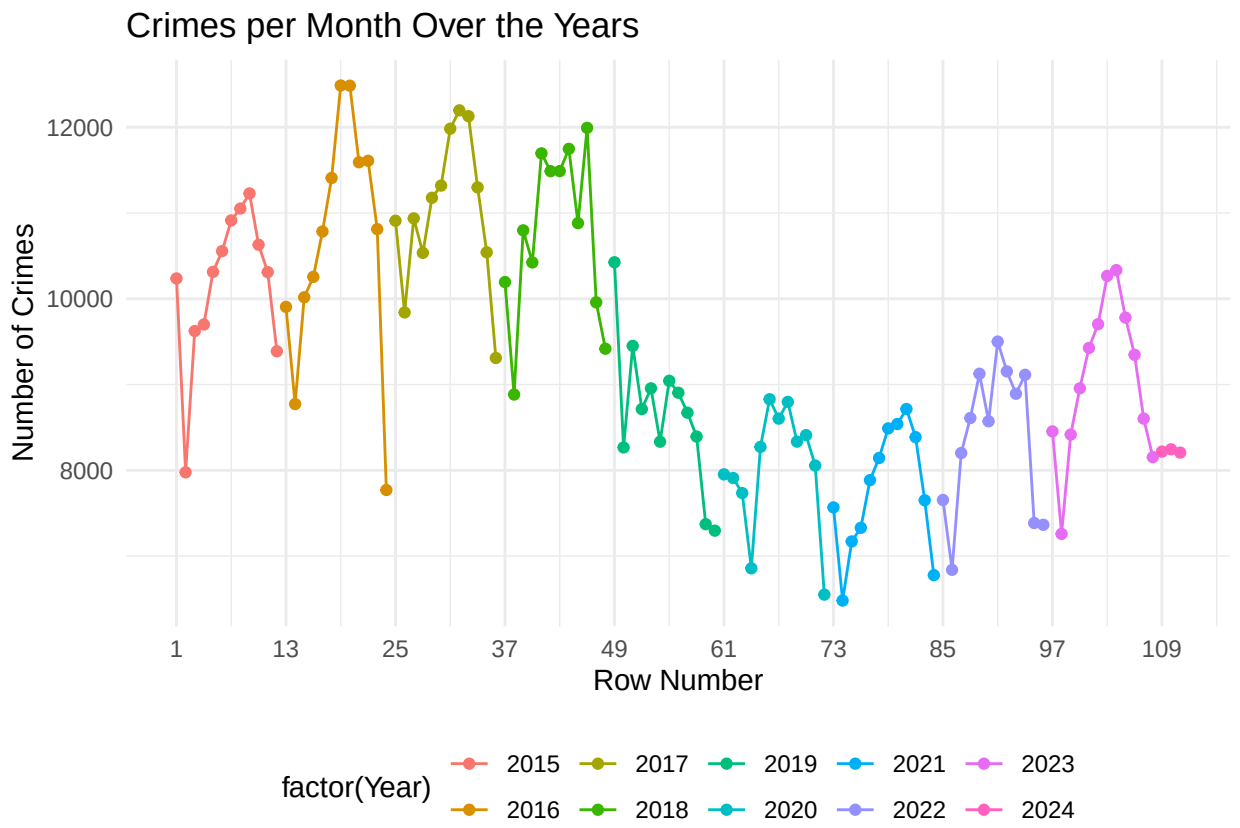
```r
# Plot the amount of crimes per month over the years
crime_data$Month <- format(as.Date(crime_data$Date), "%m")
crime_data$Month <- factor(crime_data$Month, levels = sprintf("%02d", 1:12))

crime_per_month <- crime_data %>%
  group_by(Year, Month) %>%
  summarise(total_crime = n())
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```r
# to remove april of 2024 because that data was not complete when I downloaded it
crime_per_month <- head(crime_per_month, -1)

ggplot(crime_per_month, aes(x = Month, y = total_crime, group = Year, color = factor(Year))) +
  geom_line() +
  labs(x = "Month", y = "Number of Crimes", title = "Crimes per Month Over the Years") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



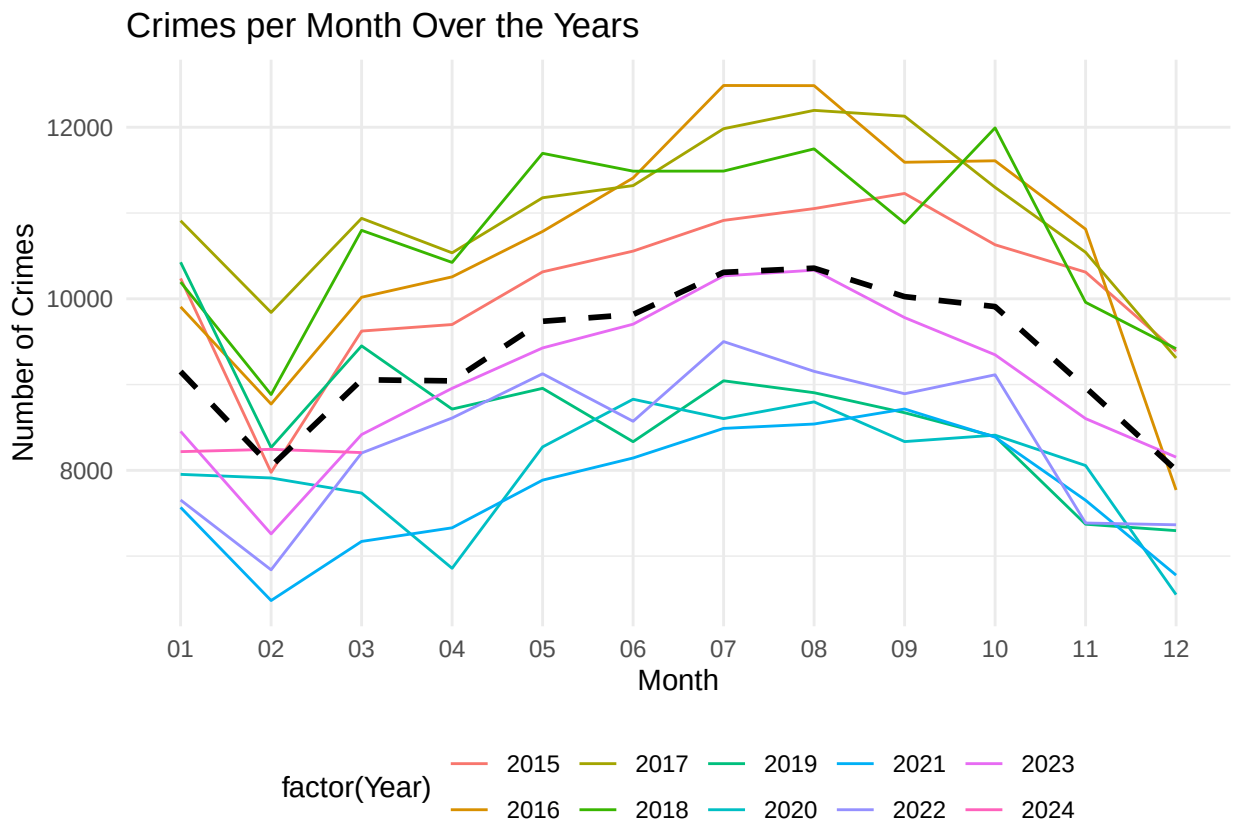

```r
crime_per_month$row_number <- 1:nrow(crime_per_month)

# Plot the amount of crimes per month over the years
ggplot(crime_per_month, aes(x = row_number, y = total_crime, color = factor(Year))) +
  geom_point() +
  geom_line() +
  labs(x = "Row Number", y = "Number of Crimes", title = "Crimes per Month Over the Years") +
  scale_x_continuous(breaks = seq(1, nrow(crime_per_month), by = 12)) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



adding the average on there for easy comparison

```
average_crime_per_month <- crime_per_month %>%
  group_by(Month) %>%
  summarise(average_crime = mean(total_crime))

average_crime_per_month$Year <- "Average"

#crime_per_month
#average_crime_per_month

# Plot the amount of crimes per month over the years with average line
ggplot(crime_per_month, aes(x = Month, y = total_crime, group = Year, color = factor(Year))) +
  geom_line() +
  geom_line(data = average_crime_per_month, aes(x = Month, y = average_crime, group = Year),
            color = "black", linetype = "dashed", size = 1) +
  labs(x = "Month", y = "Number of Crimes", title = "Crimes per Month Over the Years") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Where is crime prevelent?

```
# Identify areas with the most crime
# only printing top 25
top_areas <- crime_data %>%
  group_by(Latitude, Longitude) %>%
  summarise(total_crime = n()) %>%
  arrange(desc(total_crime)) %>%
  head(25)
```

```
## `summarise()` has grouped output by 'Latitude'. You can override using the
## `.groups` argument.
```

```
print(top_areas)
```

```
## # A tibble: 25 x 3
## # Groups:   Latitude [25]
##    Latitude Longitude total_crime
##       <dbl>     <dbl>       <int>
##  1     NA        NA         141833
##  2     39.0     -94.5         3904
##  3     39.1     -94.5         2850
##  4     39.0     -94.4         2562
```

```
##  5     39.1     -94.6        2436
##  6     39.1     -94.6        2424
##  7     39.1     -94.6        2387
##  8     39.0     -94.4        2245
##  9     39.1     -94.6        2164
## 10     39.1     -94.6        2158
## # i 15 more rows
```

this isn't very useful without a visualization

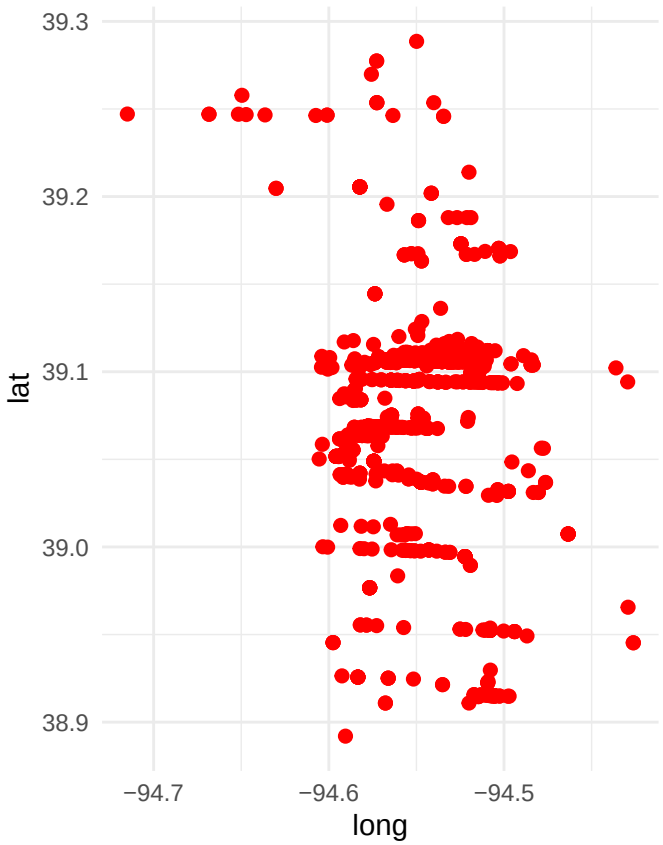
Make a map

Attempted making a map of the crime in R. However, I did not have a shape file for Kansas City and did not get the google API to work. So I will do this in Tableau

For an example here is one of my attempts only looking at 1000 points

```
world_map <- map_data("world")
cropped_map <- subset(world_map, lat >= 37 & lat <= 40 & long >= -95 & long <= -93)

cropped_data <- head(crime_data, 1000)

ggplot() +
  geom_polygon(data = cropped_map, aes(x = long, y = lat, group = group), fill = "white", color = "blac
  geom_point(data = cropped_data, aes(x = Longitude, y = Latitude), color = "red", size = 2) +
  coord_fixed() +
  theme_minimal()
```

```
## Warning: Removed 5 rows containing missing values (`geom_point()`).
```



Creation of Dataframe for Tableau

Making the Dataframe for the plot

```
plot_df <- mutate(crime_data, Description = paste("Crime", Year, sep = " "))
plot_df2 <- mutate(school_data, Description = paste(Level, "School", sep = " "))

plotdf <- plot_df[, c("Description", "Latitude", "Longitude")]
plotdf2 <- plot_df2[, c("Description", "Latitude", "Longitude")]
```

```
plotDF <- rbind(plotdf, plotdf2)

head(plotDF)
```

```
##   Description Latitude Longitude
## 1  Crime 2015  38.9767  -94.5767
## 2  Crime 2015       NA        NA
## 3  Crime 2015       NA        NA
## 4  Crime 2015  39.0947  -94.5516
## 5  Crime 2015  39.0735  -94.5461
## 6  Crime 2015       NA        NA
```

```r
tail(plotDF)
```

```
##              Description Latitude Longitude
## 1039804 Secondary School 39.09316 -94.56823
## 1039805   Primary School 38.99405 -94.53743
## 1039806   Primary School 39.08912 -94.50965
## 1039807   Primary School 39.08260 -94.55236
## 1039808   Primary School 39.10006 -94.54163
## 1039809   Primary School 39.10435 -94.55984
```

```r
write.csv(plotDF, "C:/Users/jans7/OneDrive - Marquette University/SP24/COSC 6510 - Data Intelligence/LL
```

How Many Crimes are Near Each School

```r
# remove na
crime_data <- crime_data[complete.cases(crime_data$Longitude, crime_data$Latitude), ]

# getting the data in the right format for the function
crime_sp <- crime_data %>%
  select(Longitude, Latitude) %>%
  SpatialPointsDataFrame(coords = ., data = crime_data)

school_sp <- school_data %>%
  select(Longitude, Latitude) %>%
  SpatialPointsDataFrame(coords = ., data = school_data)
```

```r
# the buffer is in meters thus this is about 1 mile away
buffer_dist <- 1609

calculate_monthly_distances <- function(crime_data, school_data, buffer_dist) {
  # empty matrix to store counts for each school, month, and year
  num_schools <- nrow(school_data)
  num_months <- 12
  num_years <- length(unique(crime_data$Year)) - 1 #don't do 2024 because incomplete year
  counts <- array(0, dim = c(num_schools, num_months, num_years))

  # each combination of year and month
  for (year in unique(crime_data$Year)[1:(num_years)]) {
    for (month in sprintf("%02d", 1:12)) {
      # Subset crime_data for the current year and month
      subset_crime <- subset(crime_data, Year == year & Month == month)

      # Calculate distances for the subset
      distances <- sapply(1:nrow(school_data), function(i) {
        dist <- distm(school_data[i, c("Longitude", "Latitude")],
                      subset_crime[, c("Longitude", "Latitude")],
                      fun = distGeo)

        sum(dist <= buffer_dist)
      })
```

```
      # Store
      counts[, as.integer(month), year - min(unique(crime_data$Year)) + 1] <- distances
    }
    # print year so we can see that we are making progress
    print(year)
  }

  # Return the counts matrix
  return(counts)
}

distances_monthly <- calculate_monthly_distances(crime_data, school_data, buffer_dist)
```

```
## [1] 2015
## [1] 2016
## [1] 2017
## [1] 2018
## [1] 2019
## [1] 2020
## [1] 2021
## [1] 2022
## [1] 2023
```

now store into a dataframe

```
years <- min(unique(crime_data$Year)) + 0:(dim(distances_monthly)[3] - 1)
months <- sprintf("%02d", 1:12)

result_df <- data.frame(school_data)

# Loop over years and months
for (year in years) {
  for (month in months) {
    # Extract counts for the current year and month
    counts <- as.vector(distances_monthly[, as.integer(month), year - min(years) + 1])

    # Add counts as a new column to the data frame
    col_name <- paste(month, "/", year, sep = "")
    result_df[[col_name]] <- counts
  }
}

head(result_df,1)
```

```
##       Level                                    SchoolName
## 1 Secondary African-Centered College Preparatory Academy
##                      Address        City State ZipCode Latitude Longitude
## 1 3500 East Meyer Boulevard Kansas City    MO   64132 39.04205 -94.53179
##   01/2015 02/2015 03/2015 04/2015 05/2015 06/2015 07/2015 08/2015 09/2015
## 1     297     285     349     328     391     370     367     450     384
##   10/2015 11/2015 12/2015 01/2016 02/2016 03/2016 04/2016 05/2016 06/2016
## 1     354     320     309     337     376     375     429     381     328
```

```
##   07/2016 08/2016 09/2016 10/2016 11/2016 12/2016 01/2017 02/2017 03/2017
## 1     456     443     450     337     397     245     181     155     195
##   04/2017 05/2017 06/2017 07/2017 08/2017 09/2017 10/2017 11/2017 12/2017
## 1     163     180     187     231     171     235     180     169     176
##   01/2018 02/2018 03/2018 04/2018 05/2018 06/2018 07/2018 08/2018 09/2018
## 1     189     103     235     239     229     214     220     196     169
##   10/2018 11/2018 12/2018 01/2019 02/2019 03/2019 04/2019 05/2019 06/2019
## 1     218     145     210     162     133     108     161     149     117
##   07/2019 08/2019 09/2019 10/2019 11/2019 12/2019 01/2020 02/2020 03/2020
## 1     145     134     147     133     118     104     158     119     125
##   04/2020 05/2020 06/2020 07/2020 08/2020 09/2020 10/2020 11/2020 12/2020
## 1     147     170     170     159     189     177     140     125     123
##   01/2021 02/2021 03/2021 04/2021 05/2021 06/2021 07/2021 08/2021 09/2021
## 1     117      93     137     156     154     120     107     152     128
##   10/2021 11/2021 12/2021 01/2022 02/2022 03/2022 04/2022 05/2022 06/2022
## 1     101     101     108     114     132     112     177     163     156
##   07/2022 08/2022 09/2022 10/2022 11/2022 12/2022 01/2023 02/2023 03/2023
## 1     166     179     138     174     120     110     112     142     131
##   04/2023 05/2023 06/2023 07/2023 08/2023 09/2023 10/2023 11/2023 12/2023
## 1     175     145     145     213     191     213     139     126     145
```

Save Dataframe

```
write.csv(result_df, "C:/Users/jans7/OneDrive - Marquette University/SP24/COSC 6510 - Data Intelligence/
```

Regression Analysis!

since, Time Series is out of the scope of this course I really wanted to look at some kind of regression on the data. My thoughts are if we look back at the graph made "Crimes per Month over the Years" there is a really consistent trend / shape of the line. It's a very strong pattern. I seams as though that if I can find an equation of that line then once you know the y-intercept or the number of crimes for January you then can essentially have a good prediction for the rest of the year. So this is my attempt to fit a regression on the average crime counts per year.

```
average_crime_per_month
```

```
## # A tibble: 12 x 3
##    Month average_crime Year
##    <fct>         <dbl> <chr>
##  1 01            9152. Average
##  2 02            8048. Average
##  3 03            9056. Average
##  4 04            9043. Average
##  5 05            9738. Average
##  6 06            9817. Average
##  7 07           10309. Average
##  8 08           10357  Average
##  9 09           10025. Average
## 10 10            9909. Average
## 11 11            8966. Average
## 12 12            8004. Average
```

Monthly

Lets just show what linear would look like

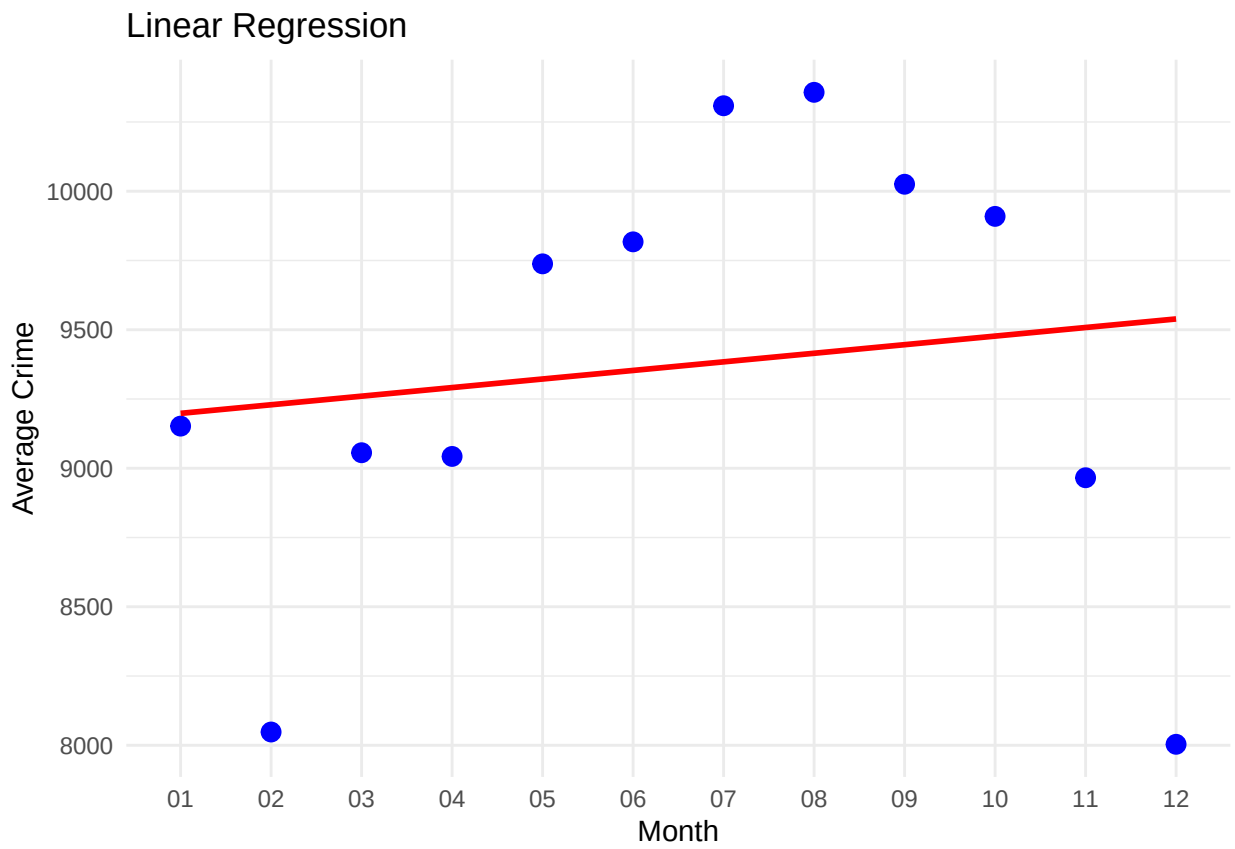
```
# Fit a second-degree polynomial regression
model <- lm(average_crime ~ poly(Month, 1, raw = TRUE), data = average_crime_per_month)

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = average_crime ~ poly(Month, 1, raw = TRUE), data = average_crime_per_month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1535.4  -321.9   184.8   492.9   941.9
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 9167.24     507.71  18.056 5.82e-09 ***
## poly(Month, 1, raw = TRUE)    30.98      68.98   0.449    0.663
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 824.9 on 10 degrees of freedom
## Multiple R-squared:  0.01977,    Adjusted R-squared:  -0.07825
## F-statistic: 0.2017 on 1 and 10 DF,  p-value: 0.6629
```

```
predicted_values <- predict(model, newdata = average_crime_per_month)

# Calculate MSE
residuals <- average_crime_per_month$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 567082.8
```

```
ggplot(average_crime_per_month, aes(x = Month, y = average_crime)) +
  geom_point(color = "blue", size = 3) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Month", y = "Average Crime", title = "Linear Regression") +
  theme_minimal()
```



As you can see it does not fit the data very well so lets try a second degree polynomial regression

```
# Fit a second-degree polynomial regression
model <- lm(average_crime ~ poly(Month, 2, raw = TRUE), data = average_crime_per_month)

# Summary of the model
summary(model)
```

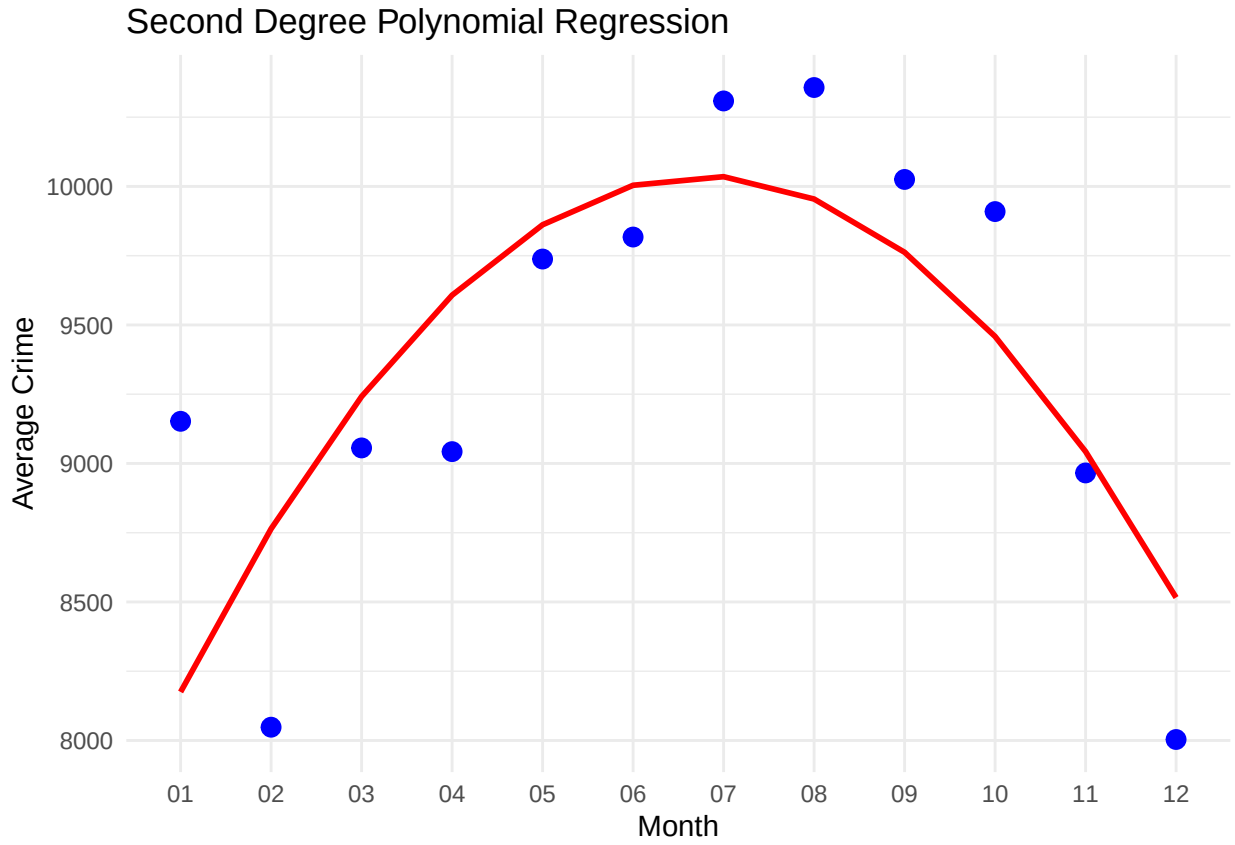
```
##
## Call:
## lm(formula = average_crime ~ poly(Month, 2, raw = TRUE), data = average_crime_per_month)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -716.3 -268.2 -100.5  305.7  977.0
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7474.49     560.67  13.331 3.13e-07 ***
## poly(Month, 2, raw = TRUE)1   756.45     198.30   3.815  0.00412 **
## poly(Month, 2, raw = TRUE)2   -55.81      14.85  -3.758  0.00450 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 542.5 on 9 degrees of freedom
## Multiple R-squared:  0.6185, Adjusted R-squared:  0.5337
```

```
## F-statistic: 7.295 on 2 and 9 DF,  p-value: 0.01309
```

```
predicted_values <- predict(model, newdata = average_crime_per_month)

# Calculate MSE
residuals <- average_crime_per_month$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 220713.3
```

```
ggplot(average_crime_per_month, aes(x = Month, y = average_crime)) +
  geom_point(color = "blue", size = 3) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Month", y = "Average Crime", title = "Second Degree Polynomial Regression") +
  theme_minimal()
```



This is better but the degree that gives the shape that I am desiring is degree 6

```
# Fit a second-degree polynomial regression
model <- lm(average_crime ~ poly(Month, 5, raw = TRUE), data = average_crime_per_month)

# Summary of the model
summary(model)
```

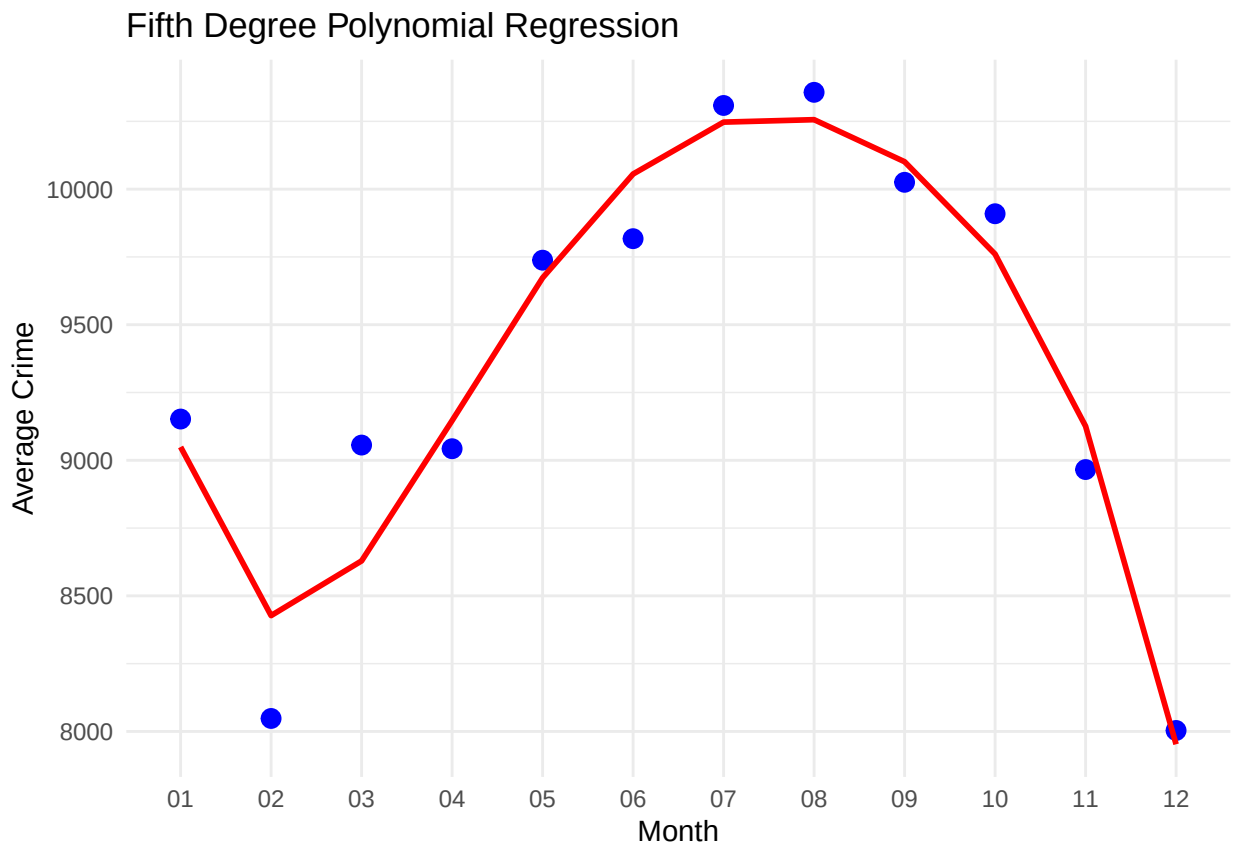
```
##
```

```
## Call:
## lm(formula = average_crime ~ poly(Month, 5, raw = TRUE), data = average_crime_per_month)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -379.4 -117.7   56.2  101.3  427.2
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  11252.1146  1234.2790   9.116 9.79e-05 ***
## poly(Month, 5, raw = TRUE)1 -3318.1331  1636.0450  -2.028   0.0889 .
## poly(Month, 5, raw = TRUE)2  1303.8562   705.8958   1.847   0.1142
## poly(Month, 5, raw = TRUE)3  -202.7322   131.3667  -1.543   0.1737
## poly(Month, 5, raw = TRUE)4    14.4173    10.9560   1.316   0.2362
## poly(Month, 5, raw = TRUE)5    -0.4014     0.3359  -1.195   0.2772
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 282.5 on 6 degrees of freedom
## Multiple R-squared:  0.931,  Adjusted R-squared:  0.8736
## F-statistic:  16.2 on 5 and 6 DF,  p-value: 0.001988
```

```
predicted_values <- predict(model, newdata = average_crime_per_month)

# Calculate MSE
residuals <- average_crime_per_month$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 39899.74
```

```
ggplot(average_crime_per_month, aes(x = Month, y = average_crime)) +
  geom_point(color = "blue", size = 3) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Month", y = "Average Crime", title = "Fifth Degree Polynomial Regression") +
  theme_minimal()
```



Weekly

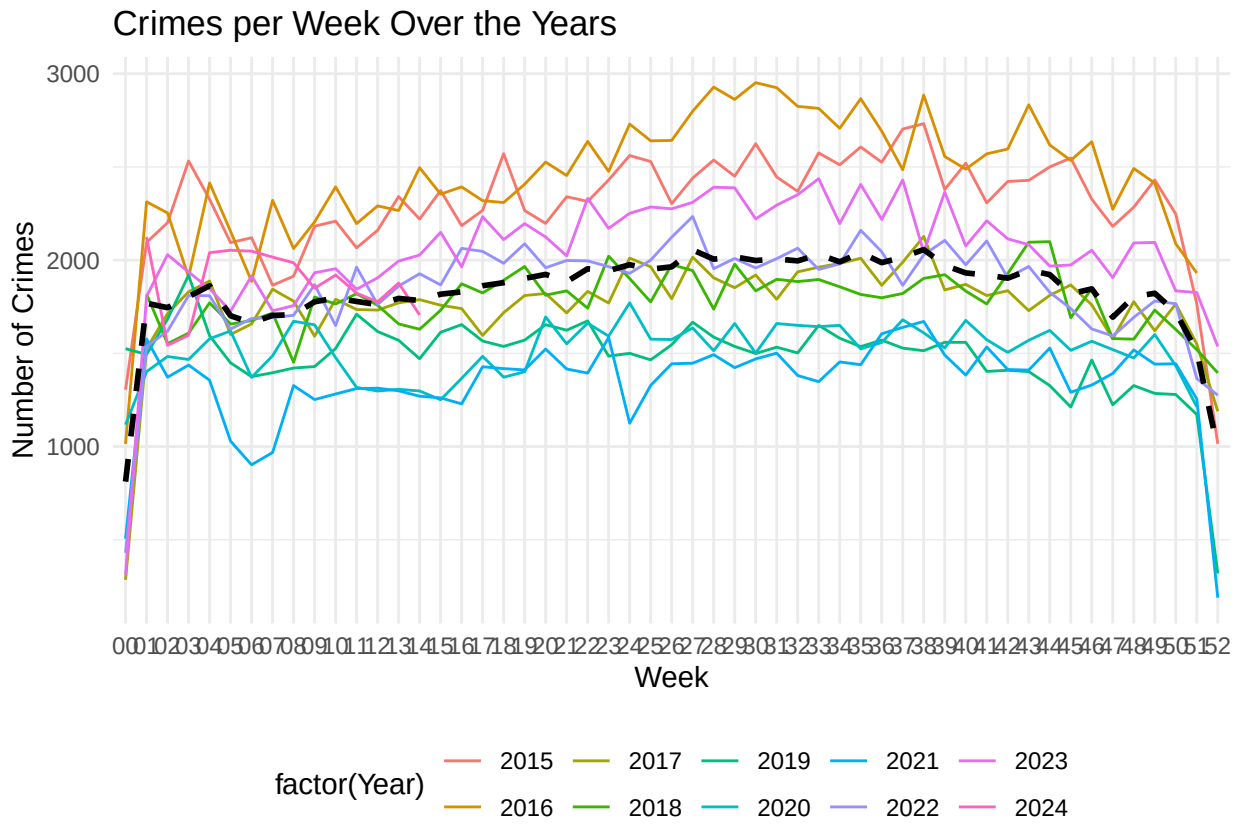
I feel like this isn't enough data with only 12 points so I want to update this to weekly?

```
# Extract week and year from the date
crime_data$Week <- format(as.Date(crime_data$Date), "%W")
crime_data$Year <- format(as.Date(crime_data$Date), "%Y")

# Group by year and week
crime_per_week <- crime_data %>%
  group_by(Year, Week) %>%
  summarise(total_crime = n())
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

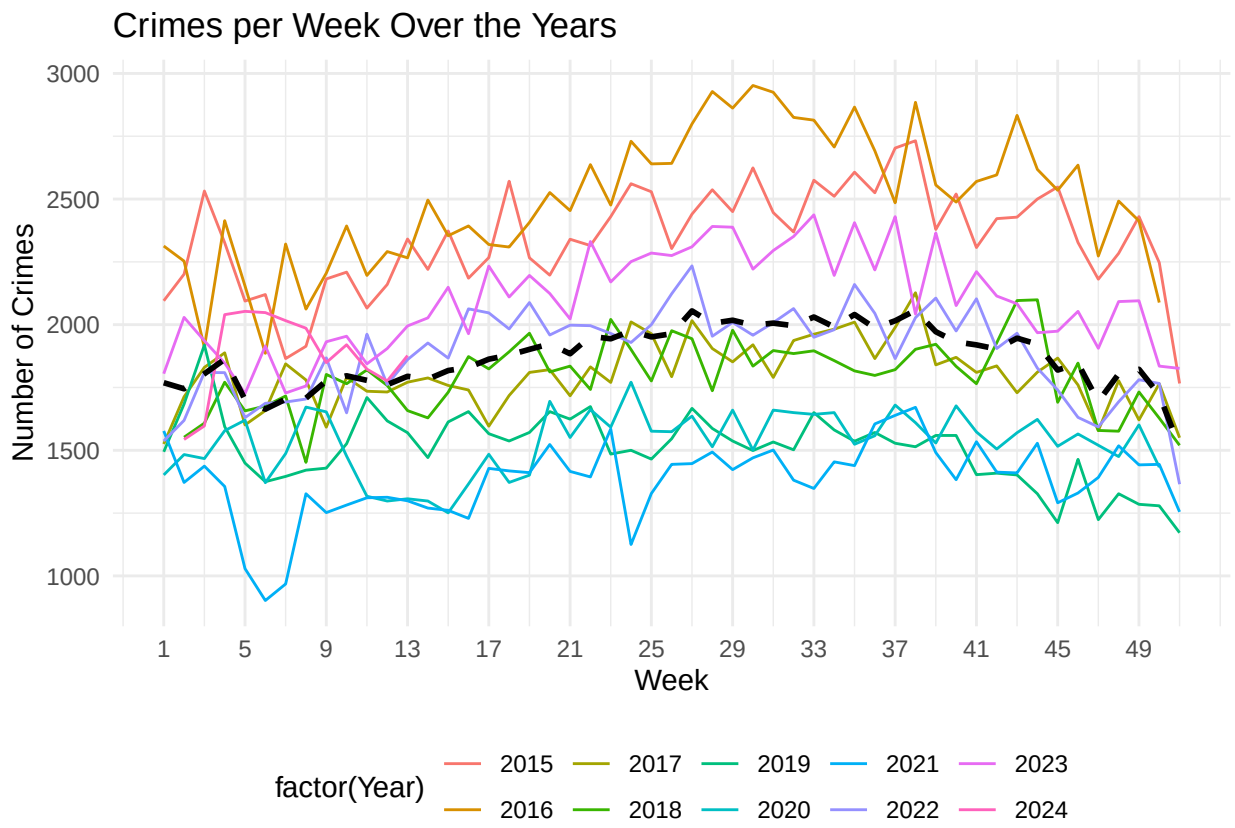
```
# Calculate average crime per week
average_crime_per_week <- crime_per_week %>%
  group_by(Week) %>%
  summarise(average_crime = mean(total_crime))

average_crime_per_week$Year <- "Average"

# Plot crimes per week over the years with average line
```

```r
ggplot(crime_per_week, aes(x = Week, y = total_crime, group = Year, color = factor(Year))) +
  geom_line() +
  geom_line(data = average_crime_per_week, aes(x = Week, y = average_crime, group = Year), color = "bla
  labs(x = "Week", y = "Number of Crimes", title = "Crimes per Week Over the Years") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Well because the first and last week are not always a complete week lets remove those from our data.

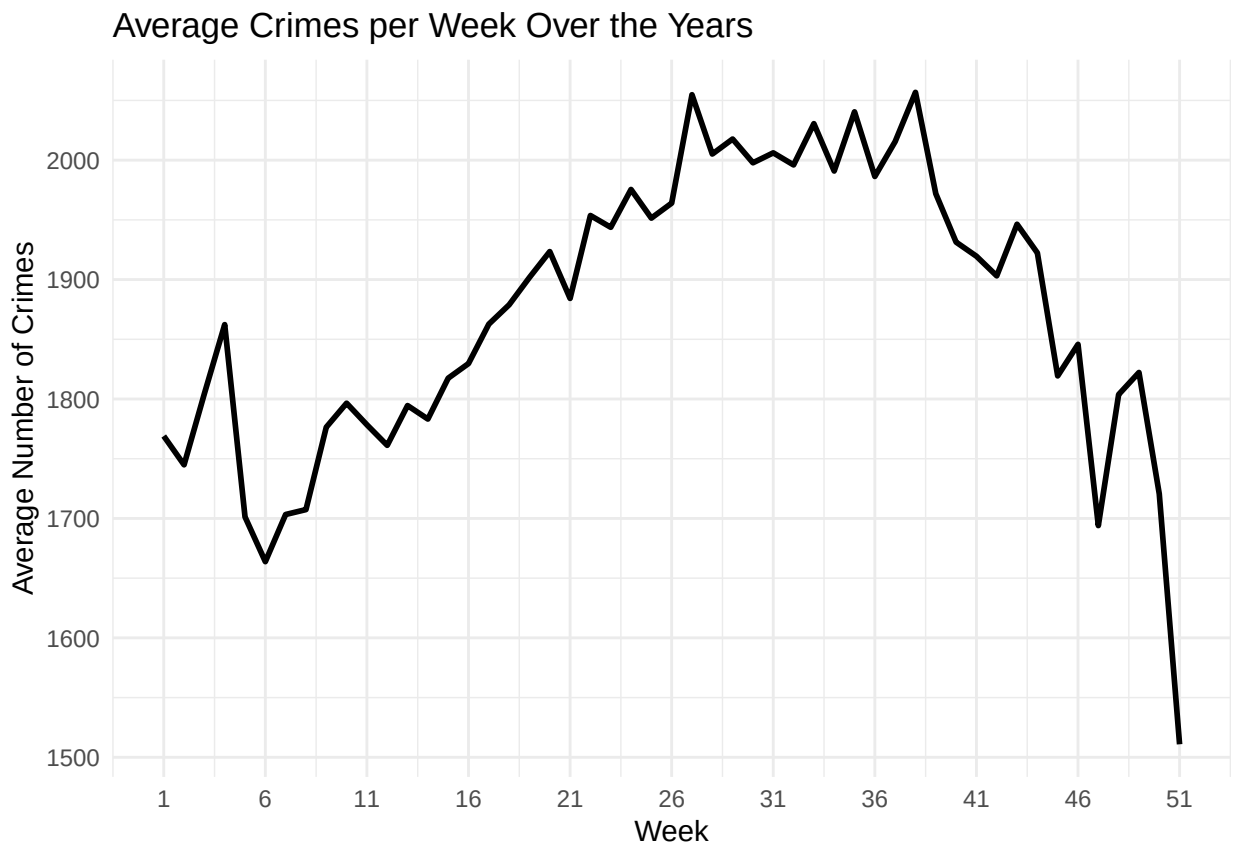
```r
# Remove first and last week of data
average_crime_per_week_c <- average_crime_per_week %>%
  filter(Week != min(Week), Week != max(Week))
crime_per_week_c <- crime_per_week %>%
  filter(Week != min(Week), Week != max(Week))

# Plot crimes per week over the years with adjusted x-axis
ggplot(crime_per_week_c, aes(x = as.numeric(Week), y = total_crime, group = Year, color = factor(Year))
  geom_line() +
  geom_line(data = average_crime_per_week_c, aes(x = as.numeric(Week), y = average_crime, group = Year)
  labs(x = "Week", y = "Number of Crimes", title = "Crimes per Week Over the Years") +
  scale_x_continuous(breaks = seq(min(as.numeric(crime_per_week_c$Week)), max(as.numeric(crime_per_week
  theme_minimal() +
  theme(legend.position = "bottom")
```



Now because this is very variable you can see how the average line is pretty smooth. It also interesting to me because when looking month to month we saw alot more variation. Also note how large the scale is

Lets look at just the average

```
ggplot(average_crime_per_week_c, aes(x = as.numeric(Week), y = average_crime)) +
  geom_line(color = "black", size = 1) + # Add average line
  labs(x = "Week", y = "Average Number of Crimes", title = "Average Crimes per Week Over the Years") +
  scale_x_continuous(breaks = seq(min(as.numeric(average_crime_per_week_c$Week)), max(as.numeric(averag
  theme_minimal()
```



First notice the scale change Second this is definitely not linear. Lets run the same models on it

```
# needs to be a factor not a character
average_crime_per_week_c$Week <- factor(average_crime_per_week_c$Week)


# Fit a second-degree polynomial regression
model <- lm(average_crime ~ poly(Week, 1, raw = TRUE), data = average_crime_per_week_c)

# Summary of the model
summary(model)
```

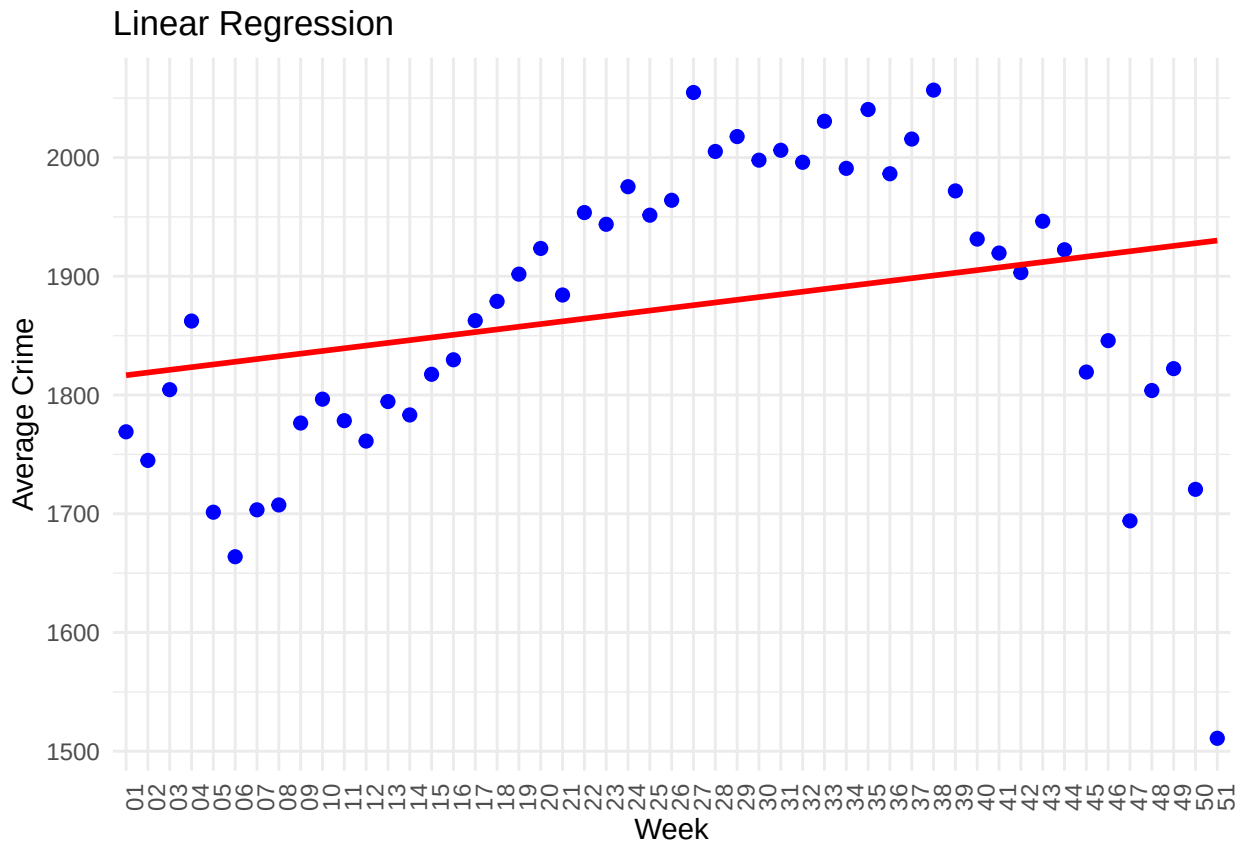
```
##
## Call:
## lm(formula = average_crime ~ poly(Week, 1, raw = TRUE), data = average_crime_per_week_c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -419.22  -67.97   12.14   90.44  179.13
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1814.379     33.372  54.368   <2e-16 ***
## poly(Week, 1, raw = TRUE)    2.269      1.117   2.032   0.0476 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 117.4 on 49 degrees of freedom
## Multiple R-squared:  0.07769,    Adjusted R-squared:  0.05886
## F-statistic: 4.127 on 1 and 49 DF,  p-value: 0.04764
```

```
predicted_values <- predict(model, newdata = average_crime_per_week_c)

# Calculate MSE
residuals <- average_crime_per_week_c$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 13245.25
```

```
ggplot(average_crime_per_week_c, aes(x = Week, y = average_crime)) +
  geom_point(color = "blue", size = 2) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Week", y = "Average Crime", title = "Linear Regression") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90,hjust = 1))
```




```
# needs to be a factor not a character
average_crime_per_week_c$Week <- factor(average_crime_per_week_c$Week)


# Fit a second-degree polynomial regression
```

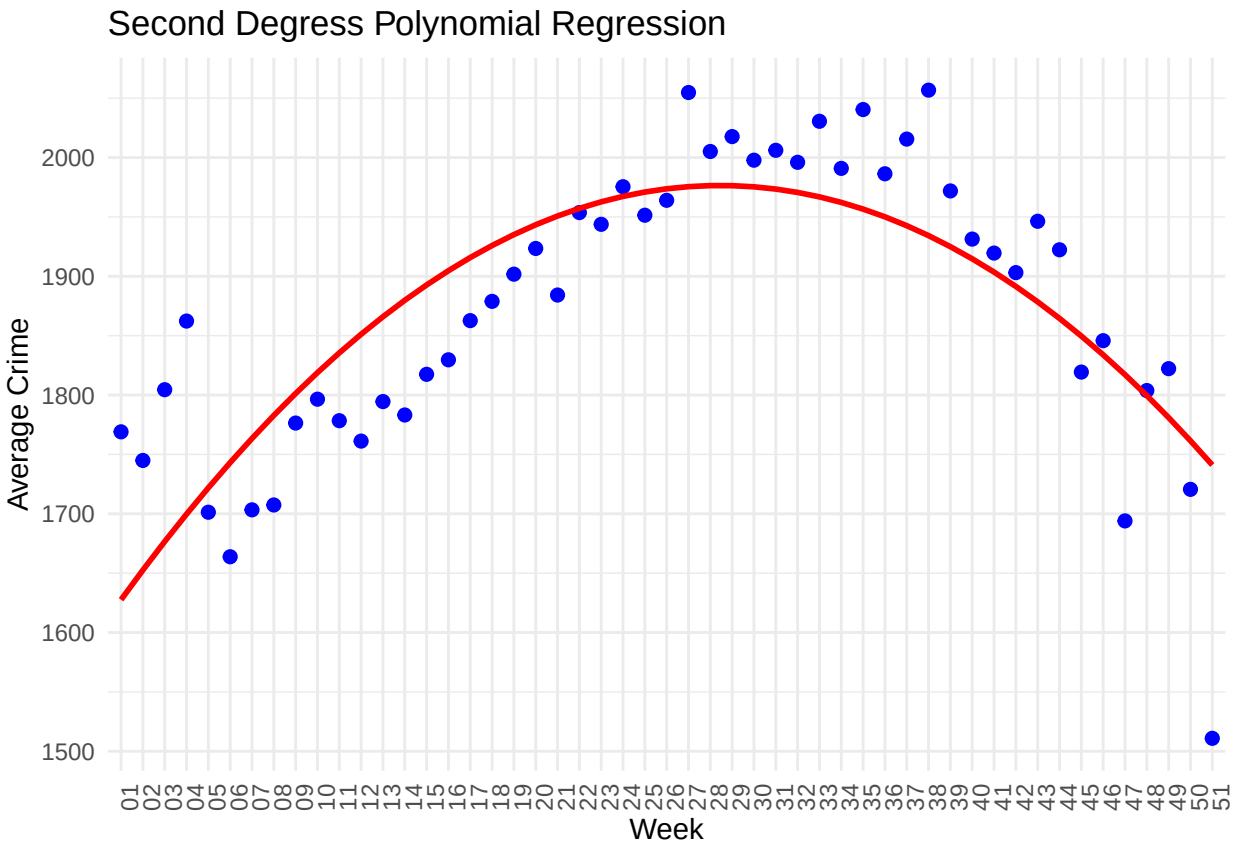
```r
model <- lm(average_crime ~ poly(Week, 2, raw = TRUE), data = average_crime_per_week_c)

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = average_crime ~ poly(Week, 2, raw = TRUE), data = average_crime_per_week_c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -230.170  -50.057    4.247   41.296  162.613
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1601.71915   32.50543  49.275  < 2e-16 ***
## poly(Week, 2, raw = TRUE)1    26.34386    2.88383   9.135 4.49e-12 ***
## poly(Week, 2, raw = TRUE)2    -0.46297    0.05376  -8.611 2.67e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.36 on 48 degrees of freedom
## Multiple R-squared:  0.6376, Adjusted R-squared:  0.6225
## F-statistic: 42.22 on 2 and 48 DF,  p-value: 2.637e-11
```

```r
predicted_values <- predict(model, newdata = average_crime_per_week_c)

# Calculate MSE
residuals <- average_crime_per_week_c$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 5204.679
```

```r
ggplot(average_crime_per_week_c, aes(x = Week, y = average_crime)) +
  geom_point(color = "blue", size = 2) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Week", y = "Average Crime", title = "Second Degress Polynomial Regression") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90,hjust = 1))
```




```r
# needs to be a factor not a character
average_crime_per_week_c$Week <- factor(average_crime_per_week_c$Week)


# Fit a second-degree polynomial regression
model <- lm(average_crime ~ poly(Week, 5, raw = TRUE), data = average_crime_per_week_c)

# Summary of the model
summary(model)
```

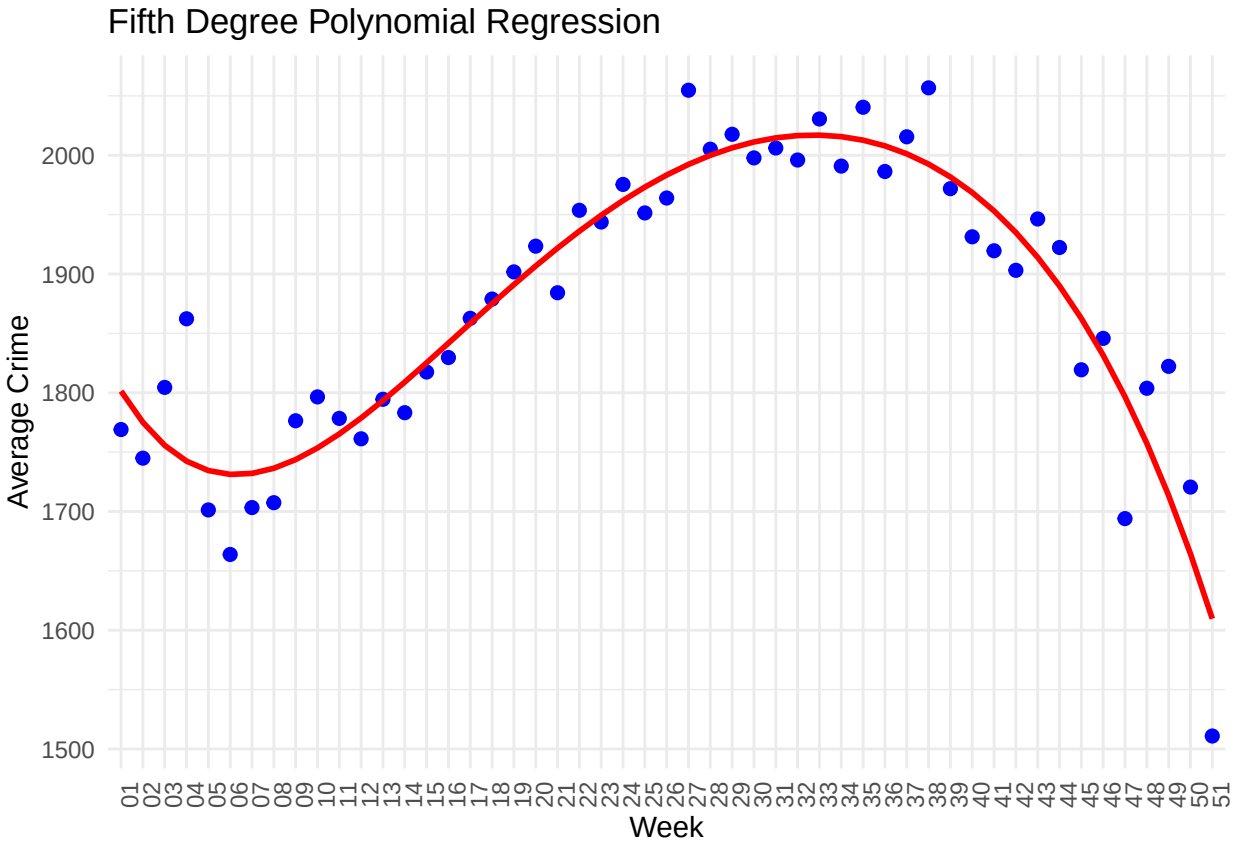
```
##
## Call:
## lm(formula = average_crime ~ poly(Week, 5, raw = TRUE), data = average_crime_per_week_c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.639  -27.284   -5.702   17.128  119.909
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.835e+03  4.502e+01  40.769   <2e-16 ***
## poly(Week, 5, raw = TRUE)1 -3.810e+01  1.689e+01  -2.255   0.0290 *
## poly(Week, 5, raw = TRUE)2  4.291e+00  1.965e+00   2.184   0.0342 *
## poly(Week, 5, raw = TRUE)3 -1.528e-01  9.475e-02  -1.613   0.1137
## poly(Week, 5, raw = TRUE)4  2.462e-03  2.001e-03   1.230   0.2250
## poly(Week, 5, raw = TRUE)5 -1.687e-05  1.532e-05  -1.102   0.2765
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.68 on 45 degrees of freedom
## Multiple R-squared:  0.8773, Adjusted R-squared:  0.8637
## F-statistic: 64.36 on 5 and 45 DF,  p-value: < 2.2e-16
```

```r
predicted_values <- predict(model, newdata = average_crime_per_week_c)

# Calculate MSE
residuals <- average_crime_per_week_c$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 1761.761
```

```r
ggplot(average_crime_per_week_c, aes(x = Week, y = average_crime)) +
  geom_point(color = "blue", size = 2) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Week", y = "Average Crime", title = "Fifth Degree Polynomial Regression") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90,hjust = 1))
```

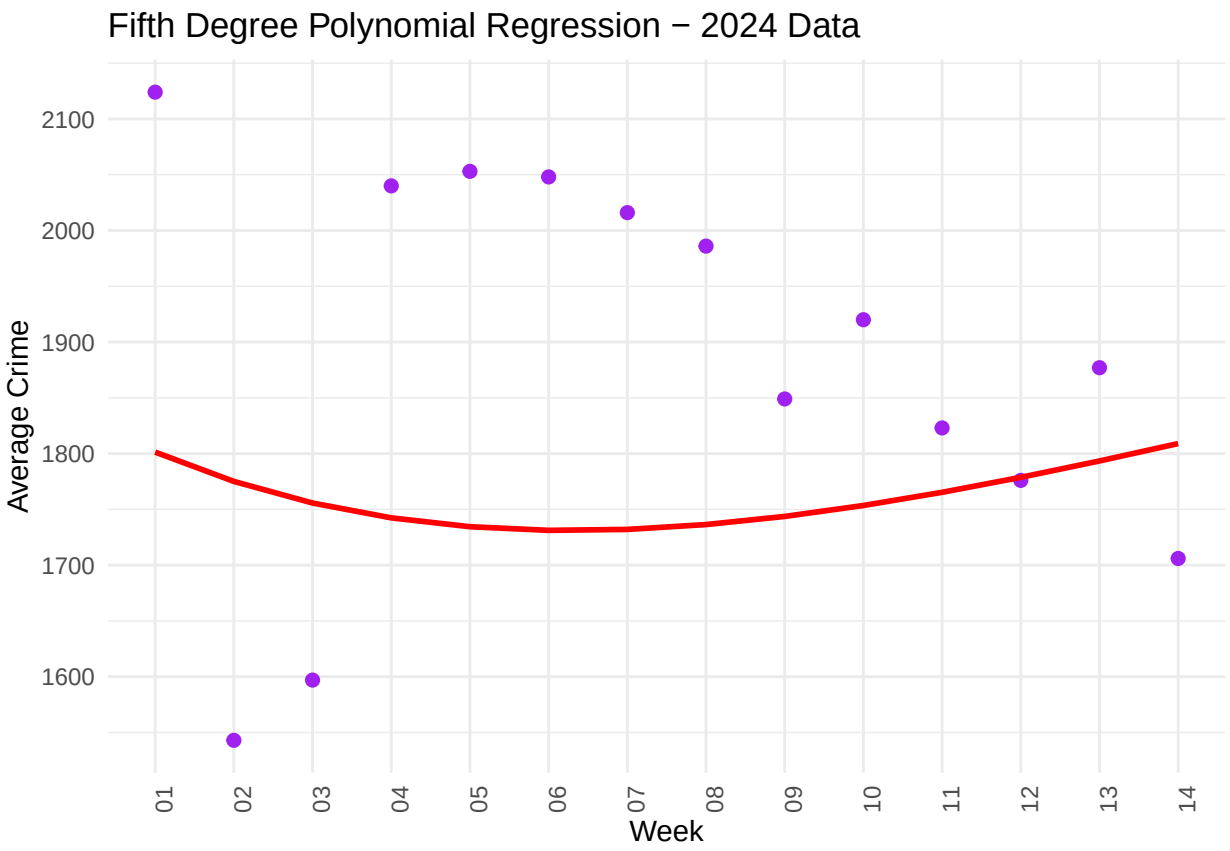


Early Predictions

```
data2024 <- subset(crime_per_week, Year == "2024" )

data2024$Week <- factor(data2024$Week)
data2024$total_crime <- as.double(data2024$total_crime)
data2024 <- data2024 %>%
  rename(average_crime = total_crime)

predicted_values <- predict(model, newdata = data2024)

# Calculate MSE
residuals <- data2024$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 48300.94
```

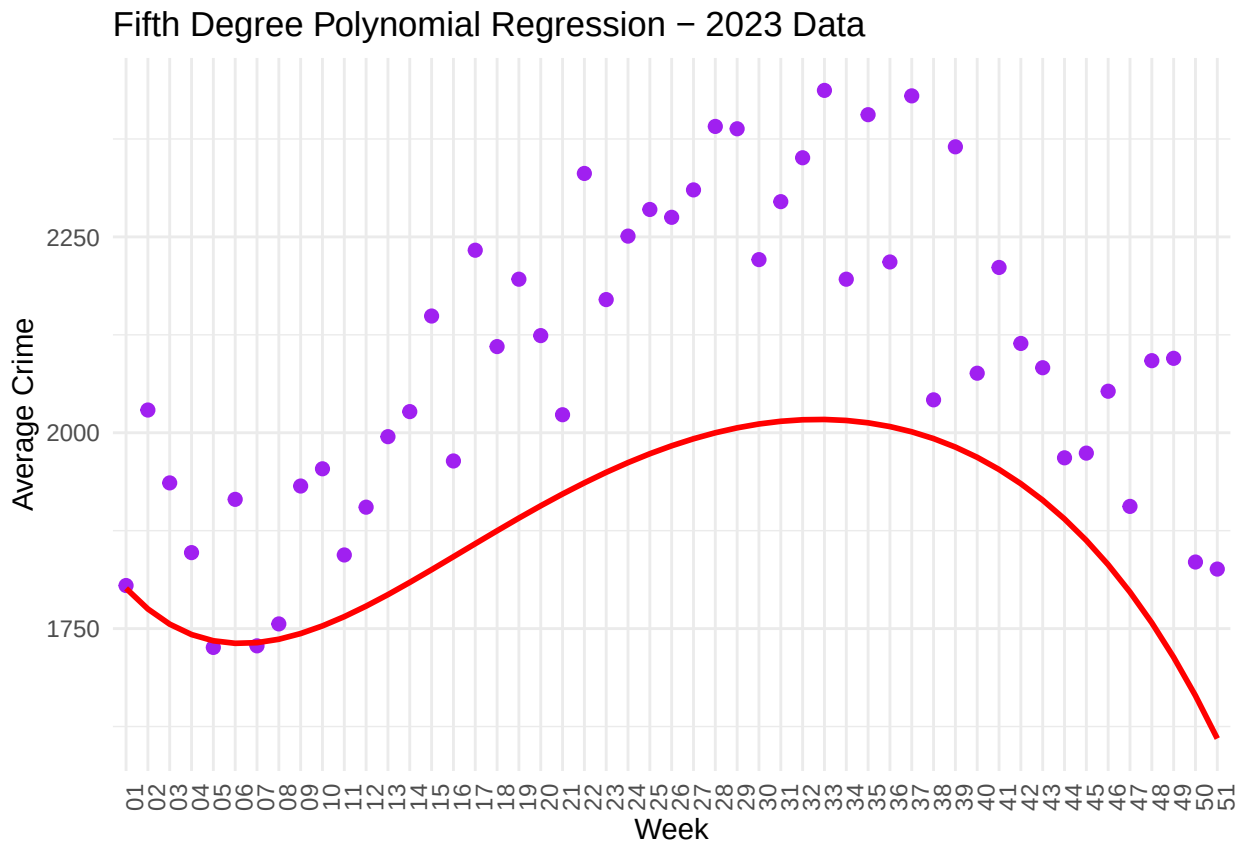
```
ggplot(data2024, aes(x = Week, y = average_crime)) +
  geom_point(color = "purple", size = 2) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Week", y = "Average Crime", title = "Fifth Degree Polynomial Regression - 2024 Data") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90,hjust = 1))
```



```r
# Extracting data for the first week of 2023
first_week_2023 <- subset(crime_per_week_c, Year == "2023" & Week == "01")
data2023 <- subset(crime_per_week_c, Year == "2023" )

data2023$Week <- factor(data2023$Week)
data2023$total_crime <- as.double(data2023$total_crime)
data2023 <- data2023 %>%
  rename(average_crime = total_crime)

predicted_values <- predict(model, newdata = data2023)

# Calculate MSE
residuals <- data2023$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 62567.64
```

```r
ggplot(data2023, aes(x = Week, y = average_crime)) +
  geom_point(color = "purple", size = 2) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Week", y = "Average Crime", title = "Fifth Degree Polynomial Regression - 2023 Data") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90,hjust = 1))
```



```r
data2022 <- subset(crime_per_week_c, Year == "2022" )

data2022$Week <- factor(data2022$Week)
data2022$total_crime <- as.double(data2022$total_crime)
data2022 <- data2022 %>%
  rename(average_crime = total_crime)

predicted_values <- predict(model, newdata = data2022)

# Calculate MSE
residuals <- data2022$average_crime - predicted_values
mse <- mean(residuals^2)
mse
```

```
## [1] 13859.19
```

```r
ggplot(data2022, aes(x = Week, y = average_crime)) +
  geom_point(color = "purple", size = 2) +  # Plot data points as blue dots
  geom_line(aes(y = predicted_values, group = 1), color = "red", size = 1) +  # Add regression line
  labs(x = "Week", y = "Average Crime", title = "Fifth Degree Polynomial Regression - 2022 Data") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90,hjust = 1))
```