Week 3 assignment_NYPD Shooting Incident Data

2022-12-04

Step 1 - Identify and import the data

I will first start by reading in the data from the NYPD Shooting Incident Data csv file.

```
## Prepare the necessary libraries for the analysis
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Get the NYDP shooring incident data from the csv file
url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
```

Now, let's read in the data and see what we have

```
## Read csv file to R
NYDP <- read_csv(url)
```

```
## Rows: 25596 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): OCCUR_DATE, BORO, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_R...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## First preview the original version of the data
NYDP
```

```
## # A tibble: 25,596 x 19
##    INCID~1 OCCUR~2 OCCUR~3 BORO  PRECI~4 JURIS~5 LOCAT~6 STATI~7 PERP_~8 PERP_~9
##      <dbl> <chr>   <time>  <chr>   <dbl>   <dbl> <chr>   <lgl>   <chr>   <chr>
##  1  2.36e8 11/11/~ 15:04   BROO~      79       0 <NA>    FALSE   <NA>    <NA>
##  2  2.31e8 07/16/~ 22:05   BROO~      72       0 <NA>    FALSE   45-64   M
##  3  2.31e8 07/11/~ 01:09   BROO~      79       0 <NA>    FALSE   <18     M
##  4  2.38e8 12/11/~ 13:42   BROO~      81       0 <NA>    FALSE   <NA>    <NA>
##  5  2.24e8 02/16/~ 20:00   QUEE~     113       0 <NA>    FALSE   <NA>    <NA>
##  6  2.28e8 05/15/~ 04:13   QUEE~     113       0 <NA>    TRUE    <NA>    <NA>
##  7  2.27e8 04/14/~ 21:08   BRONX      42       0 COMMER~ TRUE    <NA>    <NA>
##  8  2.38e8 12/10/~ 19:30   BRONX      52       0 <NA>    FALSE   <NA>    <NA>
##  9  2.25e8 02/22/~ 00:18   MANH~      34       0 <NA>    FALSE   <NA>    <NA>
## 10  2.25e8 03/07/~ 06:15   BROO~      75       0 <NA>    TRUE    25-44   M
## # ... with 25,586 more rows, 9 more variables: PERP_RACE <chr>,
## #   VIC_AGE_GROUP <chr>, VIC_SEX <chr>, VIC_RACE <chr>, X_COORD_CD <dbl>,
## #   Y_COORD_CD <dbl>, Latitude <dbl>, Longitude <dbl>, Lon_Lat <chr>, and
## #   abbreviated variable names 1: INCIDENT_KEY, 2: OCCUR_DATE, 3: OCCUR_TIME,
## #   4: PRECINCT, 5: JURISDICTION_CODE, 6: LOCATION_DESC,
## #   7: STATISTICAL_MURDER_FLAG, 8: PERP_AGE_GROUP, 9: PERP_SEX
```

Step 2 - Tidy and Transform Data

After looking at the NYDP data file, I would like to tidy the dataset and put each variable in their own column. Also, I don't need the other columns for my coming analysis,so I will remove those columns and keep only the columns that I need: OCCUR_DATE, BORO,VIC_AGE_GROUP,VIC_SEX,VIC_RACE.

```
NYDP <- NYDP %>%
  select(c(OCCUR_DATE, BORO,VIC_AGE_GROUP,VIC_SEX,VIC_RACE))
```

Now, I would like to reformat my OCCUR_DATE to a date format column instead of being a character column like in the original format.

```
NYDP2 <- NYDP %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE))
NYDP2
```

```
## # A tibble: 25,596 x 5
##    OCCUR_DATE BORO      VIC_AGE_GROUP VIC_SEX VIC_RACE
##    <date>     <chr>     <chr>         <chr>   <chr>
##  1 2021-11-11 BROOKLYN  18-24         M       BLACK
##  2 2021-07-16 BROOKLYN  25-44         M       ASIAN / PACIFIC ISLANDER
##  3 2021-07-11 BROOKLYN  25-44         M       BLACK
##  4 2021-12-11 BROOKLYN  25-44         M       BLACK
##  5 2021-02-16 QUEENS    25-44         M       BLACK
##  6 2021-05-15 QUEENS    25-44         M       BLACK
##  7 2021-04-14 BRONX     18-24         M       BLACK
##  8 2021-12-10 BRONX     25-44         M       BLACK
##  9 2021-02-22 MANHATTAN 25-44         M       BLACK HISPANIC
## 10 2021-03-07 BROOKLYN  25-44         M       WHITE HISPANIC
## # ... with 25,586 more rows
```

Do a summary of NYDP data

```
summary(NYDP2)
```

```
##    OCCUR_DATE             BORO           VIC_AGE_GROUP        VIC_SEX
##  Min.   :2006-01-01   Length:25596       Length:25596       Length:25596
##  1st Qu.:2009-05-10   Class :character   Class :character   Class :character
##  Median :2012-08-26   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2013-06-13
##  3rd Qu.:2017-07-01
##  Max.   :2021-12-31
##    VIC_RACE
##  Length:25596
##  Class :character
##  Mode  :character
##
##
##
```

Step 3 - Add Visualizations and Analysis

Visualization 1

Now, I would like to group by data for my first analysis. I would like to see the number of victim by gender for the year of 2021. So I will summarize and group my table to form a data set with OCCUR_DATE, Victim_Sex and number of victim cases.

```
#group and summarize the data for Viz_1
viz_1 <- NYDP2 %>%
  group_by(OCCUR_DATE,VIC_SEX) %>%
  summarize(number_of_case = n()) %>%
  select(OCCUR_DATE,VIC_SEX,number_of_case) %>%
  ungroup()
```

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

```
## Preview the grouped and summarize data for Viz_1
viz_1
```

```
## # A tibble: 6,903 x 3
##    OCCUR_DATE VIC_SEX number_of_case
##    <date>     <chr>            <int>
##  1 2006-01-01 M                    8
##  2 2006-01-02 M                    4
##  3 2006-01-03 M                    4
##  4 2006-01-04 M                    4
##  5 2006-01-05 M                    4
##  6 2006-01-06 M                    4
##  7 2006-01-07 M                    2
##  8 2006-01-08 M                    4
```

```
##  9 2006-01-09 M                    9
## 10 2006-01-10 M                    5
## # ... with 6,893 more rows
```

```
tail(viz_1)
```

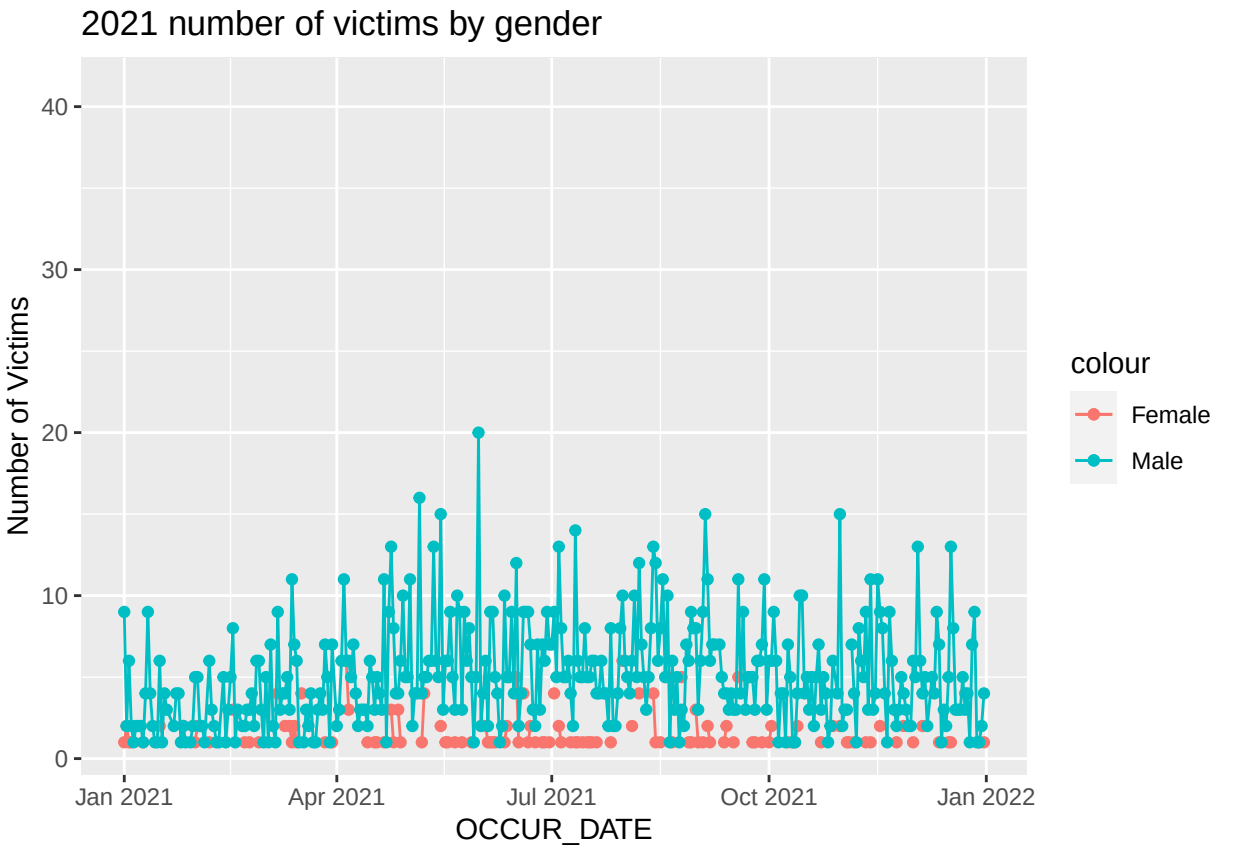
```
## # A tibble: 6 x 3
##   OCCUR_DATE VIC_SEX number_of_case
##   <date>     <chr>            <int>
## 1 2021-12-28 M                    1
## 2 2021-12-29 F                    1
## 3 2021-12-29 M                    1
## 4 2021-12-30 M                    2
## 5 2021-12-31 F                    1
## 6 2021-12-31 M                    4
```

```
## Pivot VIC_SEX from rows to columns
viz_1_pivot <- viz_1 %>%
    tidyr::spread(key = VIC_SEX,value = number_of_case)

##rename the columns
colnames(viz_1_pivot)[2] = "Female"
colnames(viz_1_pivot)[3] = "Male"
colnames(viz_1_pivot)[4] = "Undefined"

plot_1 <- viz_1_pivot %>%
  ggplot(aes(x=OCCUR_DATE, y = Female)) +
  geom_line(aes(color = "Female")) +
  geom_point(aes(color = "Female")) +
  geom_line(aes(y = Male, color = "Male")) +
  geom_point(aes(y = Male, color = "Male")) +
  scale_x_date(limits = as.Date(c('2021-01-01','2021-12-31')))+
  labs(title =str_c("2021 number of victims by gender"), y= "Number of Victims")

suppressWarnings(print(plot_1))
```



From the above plot of my first visualization (viz_1_pivot), I see the following observations:

- The number of Males being a victim is larger than Females.
- Almost at each single day during the year 2021, there are Males being victims of a crime. But this is not the case for Females, as we see the orange line for Females is not continuous throughout the year.
- More Female victims are observed from approximately June to August.
- The day with the highest number of Male victims occurred in May.

Visualization 2

For the second visualization, I would like to do an analysis for the number of victims by Boros.

```
#group and summarize the data for Viz_2
viz_2 <- NYDP2 %>%
  group_by(OCCUR_DATE,BORO) %>%
  summarize(boro_case = n()) %>%
  select(OCCUR_DATE,BORO,boro_case) %>%
  mutate(Year = year(OCCUR_DATE)) %>%
  ungroup()
```

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

```
## Preview the grouped and summarize data for Viz_1
viz_2_pivot <- viz_2 %>%
  group_by(Year,BORO) %>%
  select(Year,BORO,boro_case) %>%
  summarize(boro_case = sum(boro_case)) %>%
  ungroup()
```

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

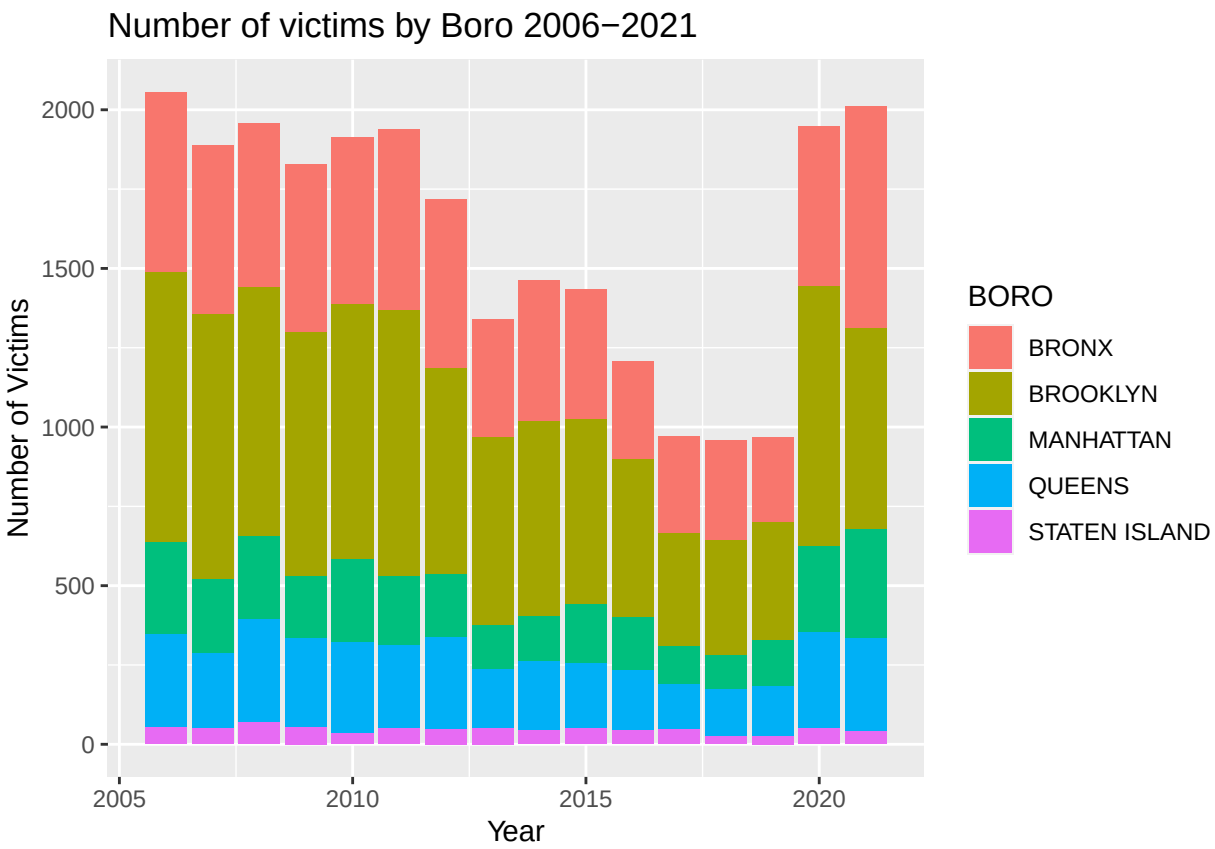
```
viz_2_pivot
```

```
## # A tibble: 80 x 3
##     Year BORO          boro_case
##    <dbl> <chr>             <int>
##  1  2006 BRONX               568
##  2  2006 BROOKLYN            850
##  3  2006 MANHATTAN           288
##  4  2006 QUEENS              296
##  5  2006 STATEN ISLAND        53
##  6  2007 BRONX               533
##  7  2007 BROOKLYN            833
##  8  2007 MANHATTAN           233
##  9  2007 QUEENS              238
## 10  2007 STATEN ISLAND        50
## # ... with 70 more rows
```

```
tail(viz_2_pivot)
```

```
## # A tibble: 6 x 3
##    Year BORO          boro_case
##   <dbl> <chr>             <int>
## 1  2020 STATEN ISLAND        50
## 2  2021 BRONX               701
## 3  2021 BROOKLYN            631
## 4  2021 MANHATTAN           343
## 5  2021 QUEENS              296
## 6  2021 STATEN ISLAND        40
```

```
plot_2 <- ggplot(data=viz_2_pivot, aes(x=Year, y=boro_case, fill=BORO)) +
  geom_bar(stat="identity") +
  labs(title =str_c("Number of victims by Boro 2006-2021"), y= "Number of Victims")
plot_2
```



From my second visualization created above, I see the following observations:

- In the period from 2006 to 2021, the total numbers of victims for all five Boros reach its minimum at the year of 2017, 2018, 2019.
- Throughout the years from 2006 to 2021, Brooklyn is the Boro that has the most number of victims.
- The Boro with the least total number of victims is Staten Island from every year of 2006-2021.
- During this time series of 16 years from 2006-2021, Bronx is the second highest boro in terms of the number of victims.
- During the time series, Manhattan and Queens Boros are having relatively close number of total victims for each of the Boros, with Queens slightly higher than Manhattan.

Create data model

```
mod_data <- viz_2_pivot %>%
  group_by(Year) %>%
  summarize(total_vic_cases = sum(boro_case)) %>%
  select(Year,total_vic_cases) %>%
  ungroup()
mod_data
```

```
## # A tibble: 16 x 2
##     Year total_vic_cases
##    <dbl>           <int>
##  1  2006            2055
```

```
##  2  2007            1887
##  3  2008            1959
##  4  2009            1828
##  5  2010            1912
##  6  2011            1939
##  7  2012            1717
##  8  2013            1339
##  9  2014            1464
## 10  2015            1434
## 11  2016            1208
## 12  2017             970
## 13  2018             958
## 14  2019             967
## 15  2020            1948
## 16  2021            2011
```

```
mod <- lm(total_vic_cases~Year,data=mod_data)
summary(mod)
```

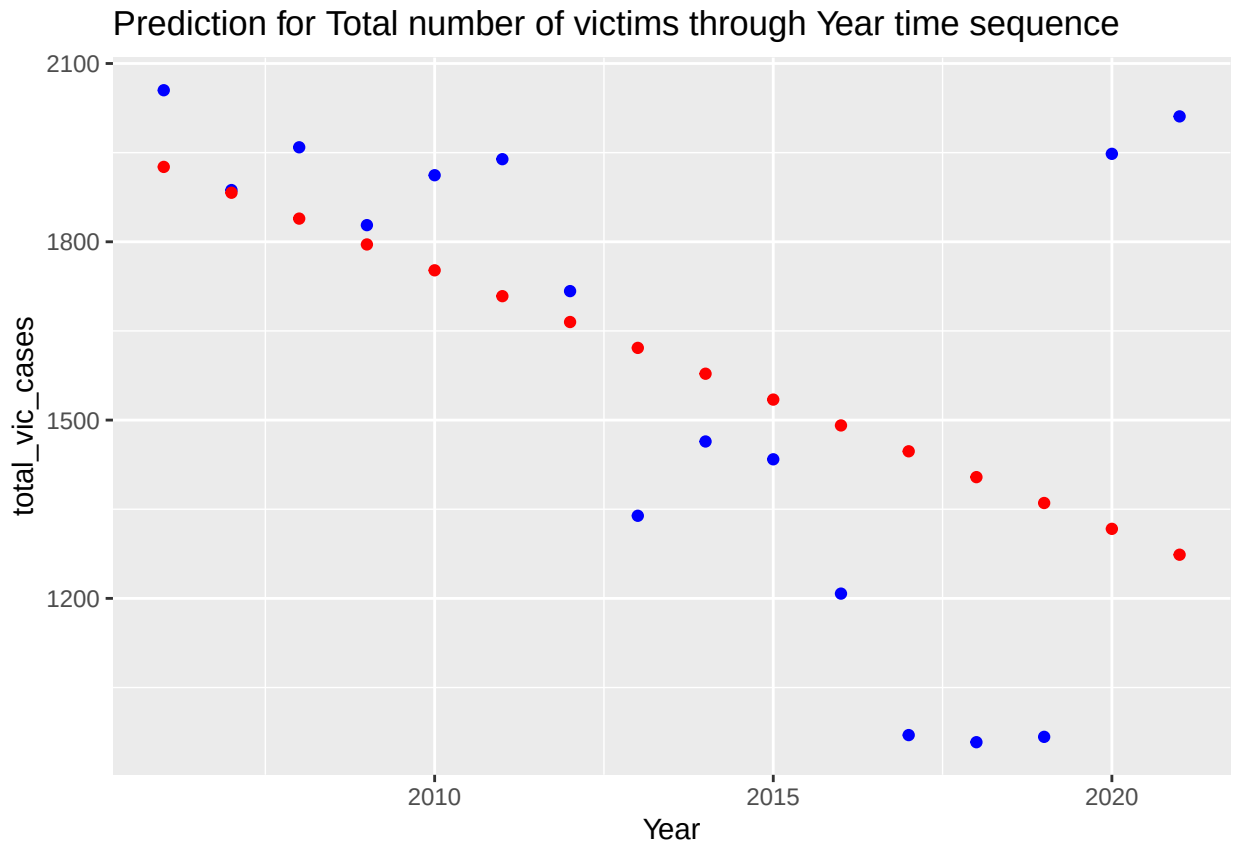
```
##
## Call:
## lm(formula = total_vic_cases ~ Year, data = mod_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -477.49 -282.62   18.48  136.73  737.52
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 89192.92   39375.83   2.265   0.0399 *
## Year          -43.50      19.56  -2.225   0.0431 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 360.6 on 14 degrees of freedom
## Multiple R-squared:  0.2612, Adjusted R-squared:  0.2084
## F-statistic: 4.949 on 1 and 14 DF,  p-value: 0.04307
```

```
x_grid <- seq(2006,2021)
new_df <- tibble(Year = x_grid)
mod_est <- mod_data %>% mutate(pred=predict(mod))
mod_est
```

```
## # A tibble: 16 x 3
##     Year total_vic_cases  pred
##    <dbl>           <int> <dbl>
##  1  2006            2055 1926.
##  2  2007            1887 1883.
##  3  2008            1959 1839.
##  4  2009            1828 1796.
##  5  2010            1912 1752.
##  6  2011            1939 1709.
##  7  2012            1717 1665.
```

```
##  8  2013            1339 1622.
##  9  2014            1464 1578.
## 10  2015            1434 1534.
## 11  2016            1208 1491.
## 12  2017             970 1447.
## 13  2018             958 1404.
## 14  2019             967 1360.
## 15  2020            1948 1317.
## 16  2021            2011 1273.
```

```
mod_est %>% ggplot() +
  geom_point(aes(x=Year, y=total_vic_cases),color="blue")+
  geom_point(aes(x=Year, y=pred),color="red") +
  labs(title =str_c("Prediction for Total number of victims through Year time sequence"))
```



From the above plot, we see that the red dots represent the predicted model and the blue dots represent the actual numbers of victims. I am trying to predict the number of total victims in time sequence of the years. This is a straight-forward and simple model. I see that this model somehow predicted the decrease from 2006 to 2015, but it seems that it cannot reflect the increase after 2020. So definitely, the year factor is not sufficient to construct a complete model as we see that the p-value for year is not significantly small. So, to improve this model, I would suggest to add more parameters in the factors for this model and need to investigate further, whether this model would be a linear model or maybe a quadratic model would perform better for the prediction.

Step 4 - Add Bias Identification

```
viz_3_pivot <- viz_1_pivot %>%
  group_by(OCCUR_DATE,Undefined) %>%
  select(OCCUR_DATE,Undefined) %>%
  summarize(Undefined = sum(Undefined)) %>%
  ungroup()
```

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

```
viz_4_pivot <- viz_3_pivot %>%
  group_by(Undefined) %>%
  select(Undefined) %>%
  summarize(abc=sum(!is.na(Undefined))) %>%
  ungroup()
viz_4_pivot
```

```
## # A tibble: 3 x 2
##   Undefined   abc
##       <int> <int>
## 1         1     7
## 2         2     2
## 3        NA     0
```

For my analysis, I think the bias would come from the data source level. I observed that in the Vic_sex column, there is a category "Undefined". From the above queries, I see there are 11 victims that are classified neither to be female nor male, but they fall into the category of Undefined. I am not sure what is the reason for these 11 people to be undefined (maybe for any political or humanity issues for not disclose victim's detail information). So this makes me thinking and questioning about the precision of my study if I do my analysis based on Gender. Another of my question is, I noticed there are many fields that are NA in the columns for perpetrator's related information. If I wish to do an analysis to see the detection rate (for how many crimes the police has successfully detected a perpetrator and how many the police did not find), this data brings me the doubt, whether the rows without perpetrator are really unsolved cases or because the perpetrator's information cannot be disclosed to the public due to data confidentiality. If this is the case, then using this data set to conduct an analysis on detective rate would be biased.

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.utf8  LC_CTYPE=English_Canada.utf8
## [3] LC_MONETARY=English_Canada.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.utf8
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.0  timechange_0.1.1 forcats_0.5.2    stringr_1.4.1
##  [5] dplyr_1.0.10     purrr_0.3.5      readr_2.1.3      tidyr_1.2.1
##  [9] tibble_3.1.8     ggplot2_3.4.0    tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.30       utf8_1.2.2
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.18       highr_0.9
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.0.6
## [13] googlesheets4_1.0.1 curl_4.3.3          readxl_1.4.1
## [16] rstudioapi_0.14     rmarkdown_2.18      labeling_0.4.2
## [19] googledrive_2.0.0   bit_4.0.5           munsell_0.5.0
## [22] broom_1.0.1         compiler_4.2.2      modelr_0.1.10
## [25] xfun_0.35           pkgconfig_2.0.3     htmltools_0.5.3
## [28] tidyselect_1.2.0    fansi_1.0.3         crayon_1.5.2
## [31] tzdb_0.3.0          dbplyr_2.2.1        withr_2.5.0
## [34] grid_4.2.2          jsonlite_1.8.3      gtable_0.3.1
## [37] lifecycle_1.0.3     DBI_1.1.3           magrittr_2.0.3
## [40] scales_1.2.1        cli_3.4.1           stringi_1.7.8
## [43] vroom_1.6.0         farver_2.1.1        fs_1.5.2
## [46] xml2_1.3.3          ellipsis_0.3.2      generics_0.1.3
## [49] vctrs_0.5.1         tools_4.2.2         bit64_4.0.5
## [52] glue_1.6.2          hms_1.1.2           parallel_4.2.2
## [55] fastmap_1.1.0       yaml_2.3.6          colorspace_2.0-3
## [58] gargle_1.2.1        rvest_1.0.3         knitr_1.41
## [61] haven_2.5.1
```