NYPD Project

Ryan Ruff

2024-11-13

```r
# Load libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lubridate)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(rsample)
```

```r
# Import the dataset
nypd_data <- read_csv("NYPD_Shooting_Incident_Data__Historic_.csv")
```

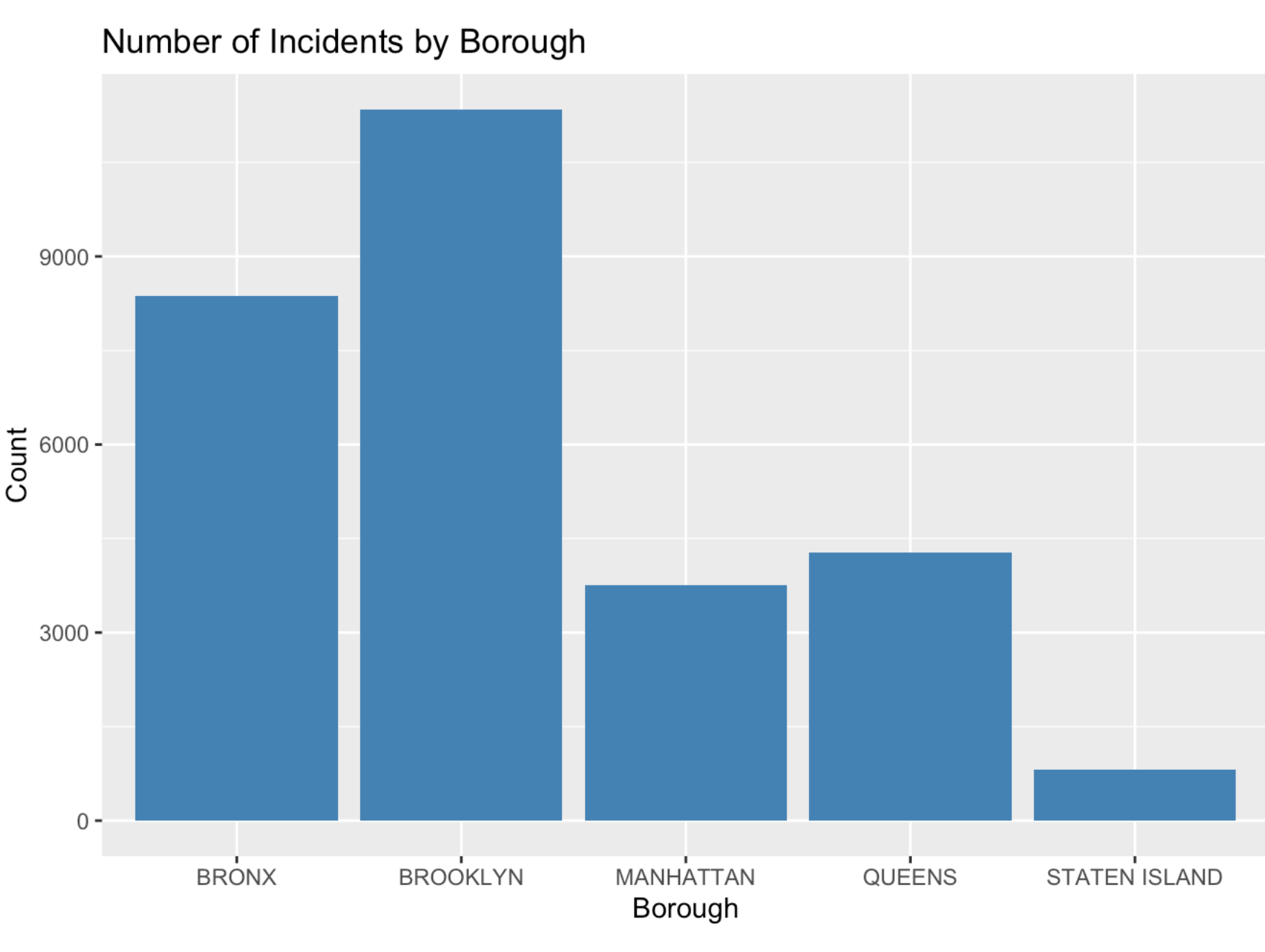
```
## Rows: 28562 Columns: 21
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## 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.
```

```r
# Inspect the data
glimpse(nypd_data)
```

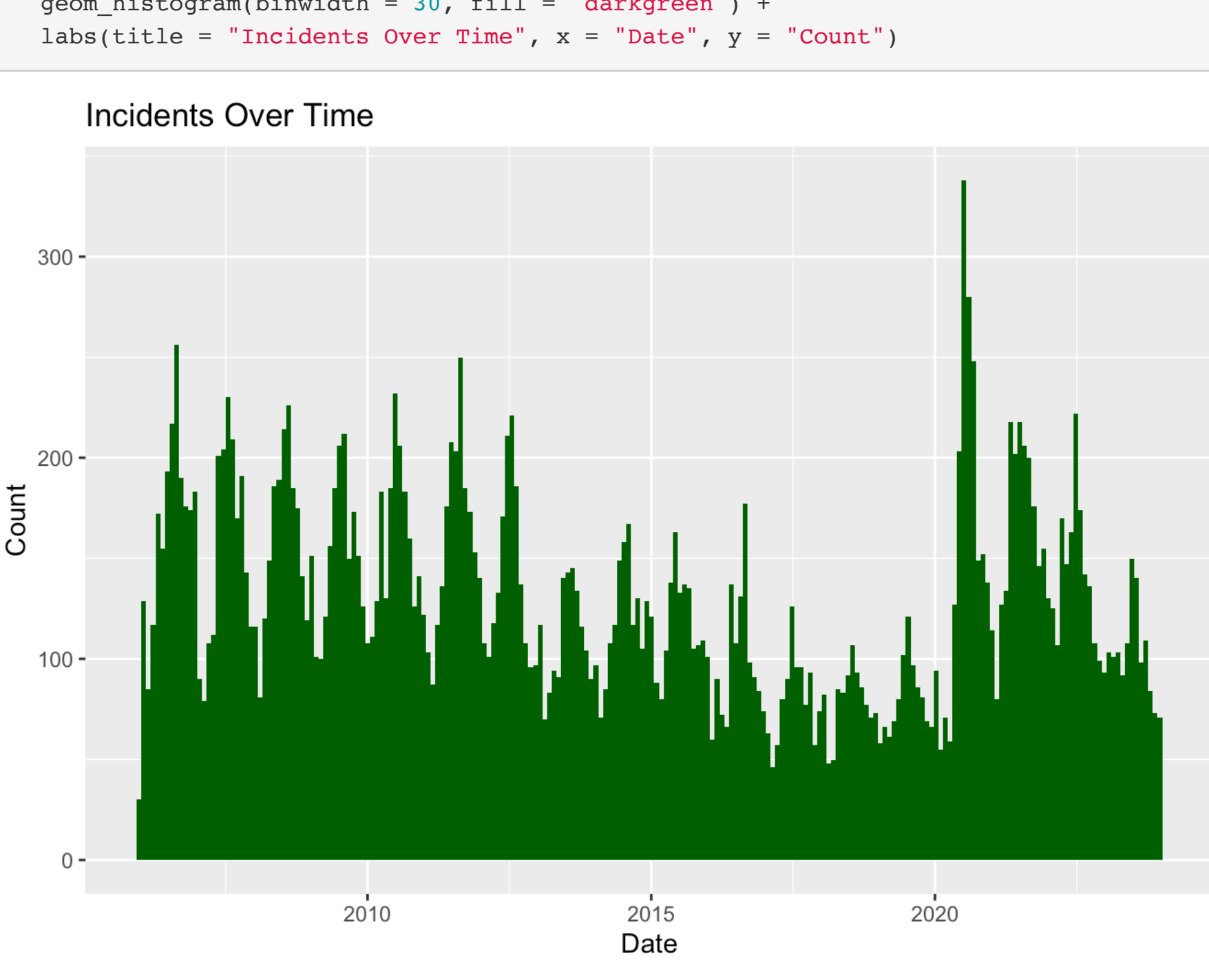
```
## Rows: 28,562
## Columns: 21
## $ INCIDENT_KEY            <dbl> 244608249, 247542571, 84967535, 202853370, 270…
## $ OCCUR_DATE              <chr> "5/5/2022", "7/4/2022", "5/27/2012", "9/24/201…
## $ OCCUR_TIME              <time> 00:10:00, 22:20:00, 19:35:00, 21:00:00, 21:00…
## $ BORO                    <chr> "MANHATTAN", "BRONX", "QUEENS", "BRONX", "BROO…
## $ LOC_OF_OCCUR_DESC       <chr> "INSIDE", "OUTSIDE", NA, NA, NA, NA, NA, NA, N…
## $ PRECINCT                <dbl> 14, 48, 103, 42, 83, 23, 113, 77, 48, 49, 73, …
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ LOC_CLASSFCTN_DESC      <chr> "COMMERCIAL", "STREET", NA, NA, NA, NA, NA, NA…
## $ LOCATION_DESC           <chr> "VIDEO STORE", "(null)", NA, NA, NA, "MULTI DW…
## $ STATISTICAL_MURDER_FLAG <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, TRUE, …
## $ PERP_AGE_GROUP          <chr> "25-44", "(null)", NA, "25-44", "25-44", NA, N…
## $ PERP_SEX                <chr> "M", "(null)", NA, "M", "M", NA, NA, NA, NA, "…
## $ PERP_RACE               <chr> "BLACK", "(null)", NA, "UNKNOWN", "BLACK", NA,…
## $ VIC_AGE_GROUP           <chr> "25-44", "18-24", "18-24", "25-44", "25-44", "…
## $ VIC_SEX                 <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "…
## $ VIC_RACE                <chr> "BLACK", "BLACK", "BLACK", "BLACK", "BLACK", "…
## $ X_COORD_CD              <dbl> 986050, 1016802, 1048632, 1014493, 1009149, 99…
## $ Y_COORD_CD              <dbl> 214231.0, 250581.0, 198262.0, 242565.0, 190104…
## $ Latitude                <dbl> 40.75469, 40.85440, 40.71063, 40.83242, 40.688…
## $ Longitude               <dbl> -73.99350, -73.88233, -73.76777, -73.89071, -7…
## $ Lon_Lat                 <chr> "POINT (-73.9935 40.754692)", "POINT (-73.8823…
```

```r
# Convert date and time columns to appropriate formats
nypd_data <- nypd_data %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE),
         OCCUR_TIME = hms(OCCUR_TIME),
         YEAR = year(OCCUR_DATE))

# Fill missing values, if any
nypd_data <- nypd_data %>%
  fill(PERP_AGE_GROUP, PERP_SEX, PERP_RACE, .direction = "downup") %>%
  fill(VIC_AGE_GROUP, VIC_SEX, VIC_RACE, .direction = "downup")

# Ensure the relevant features are factors with the same levels
nypd_data <- nypd_data %>%
  mutate(PERP_AGE_GROUP = factor(PERP_AGE_GROUP),
         PERP_SEX = factor(PERP_SEX),
         VIC_AGE_GROUP = factor(VIC_AGE_GROUP),
         VIC_SEX = factor(VIC_SEX),
         FATAL = ifelse(STATISTICAL_MURDER_FLAG == "TRUE", 1, 0))
```

```r
# Plot the number of incidents by borough
ggplot(nypd_data, aes(x = BORO)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Number of Incidents by Borough", x = "Borough", y = "Count")
```




```r
# Plot incidents over time
ggplot(nypd_data, aes(x = OCCUR_DATE)) +
  geom_histogram(binwidth = 30, fill = "darkgreen") +
  labs(title = "Incidents Over Time", x = "Date", y = "Count")
```




```r
# Select relevant features and split the data
set.seed(123)
data_split <- initial_split(nypd_data, prop = 0.7)
train_data <- training(data_split)
test_data <- testing(data_split)

# Align factor levels
test_data <- test_data %>%
  mutate(PERP_AGE_GROUP = factor(PERP_AGE_GROUP, levels = levels(train_data$PERP_AGE_GROUP)),
         PERP_SEX = factor(PERP_SEX, levels = levels(train_data$PERP_SEX)),
         VIC_AGE_GROUP = factor(VIC_AGE_GROUP, levels = levels(train_data$VIC_AGE_GROUP)),
         VIC_SEX = factor(VIC_SEX, levels = levels(train_data$VIC_SEX)),
         FATAL = factor(FATAL, levels = c(0, 1)))
```

```r
# Train a logistic regression model
model <- train(FATAL ~ BORO + PERP_AGE_GROUP + PERP_SEX + VIC_AGE_GROUP + VIC_SEX,
               data = train_data, method = "glm", family = "binomial")
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```r
# Model summary
summary(model)
```

```
##
## Call:
## NULL
##
## Coefficients: (2 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.114e+00  1.234e-01 -17.125  < 2e-16 ***
## BOROBROOKLYN            3.167e-03  4.418e-02   0.072 0.942848
## BOROMANHATTAN          -1.320e-01  6.131e-02  -2.154 0.031276 *
## BOROQUEENS              3.957e-03  5.734e-02   0.069 0.944979
## `BOROSTATEN ISLAND`     1.189e-01  1.078e-01   1.104 0.269768
## `PERP_AGE_GROUP<18`     6.927e-01  1.583e-01   4.375 1.22e-05 ***
## PERP_AGE_GROUP1020     -1.044e+01  2.292e+02  -0.046 0.963683
## PERP_AGE_GROUP1028             NA         NA      NA       NA
## `PERP_AGE_GROUP18-24`   7.748e-01  1.481e-01   5.230 1.69e-07 ***
## PERP_AGE_GROUP224      -1.030e+01  3.247e+02  -0.032 0.974699
## `PERP_AGE_GROUP25-44`   9.294e-01  1.479e-01   6.285 3.27e-10 ***
## `PERP_AGE_GROUP45-64`   1.184e+00  1.659e-01   7.140 9.36e-13 ***
## `PERP_AGE_GROUP65+`     1.265e+00  2.872e-01   4.406 1.05e-05 ***
## PERP_AGE_GROUP940      -1.061e+01  1.451e+02  -0.073 0.941698
## PERP_AGE_GROUPUNKNOWN -4.565e-01  1.205e-01  -3.790 0.000151 ***
## PERP_SEXF              -2.220e-01  1.582e-01  -1.403 0.160506
## PERP_SEXM              -3.776e-01  1.161e-01  -3.251 0.001151 **
## PERP_SEXU                      NA         NA      NA       NA
## VIC_AGE_GROUP1022      -1.072e+01  3.247e+02  -0.033 0.973669
## `VIC_AGE_GROUP18-24`    2.234e-01  7.331e-02   3.048 0.002303 **
## `VIC_AGE_GROUP25-44`    4.855e-01  7.160e-02   6.781 1.19e-11 ***
## `VIC_AGE_GROUP45-64`    7.088e-01  9.119e-02   7.774 7.63e-15 ***
## `VIC_AGE_GROUP65+`      1.220e+00  1.951e-01   6.254 3.99e-10 ***
## VIC_AGE_GROUPUNKNOWN    8.657e-01  3.510e-01   2.466 0.013654 *
## VIC_SEXM                9.974e-04  6.158e-02   0.016 0.987078
## VIC_SEXU               -1.274e+00  1.083e+00  -1.176 0.239751
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19679  on 19992  degrees of freedom
## Residual deviance: 19053  on 19969  degrees of freedom
## AIC: 19101
##
## Number of Fisher Scoring iterations: 11
```

```r
# Ensure there are no NA values in the test data
test_data <- test_data %>%
  drop_na(PERP_AGE_GROUP, PERP_SEX, VIC_AGE_GROUP, VIC_SEX)

# Make predictions
predictions <- predict(model, test_data)
```

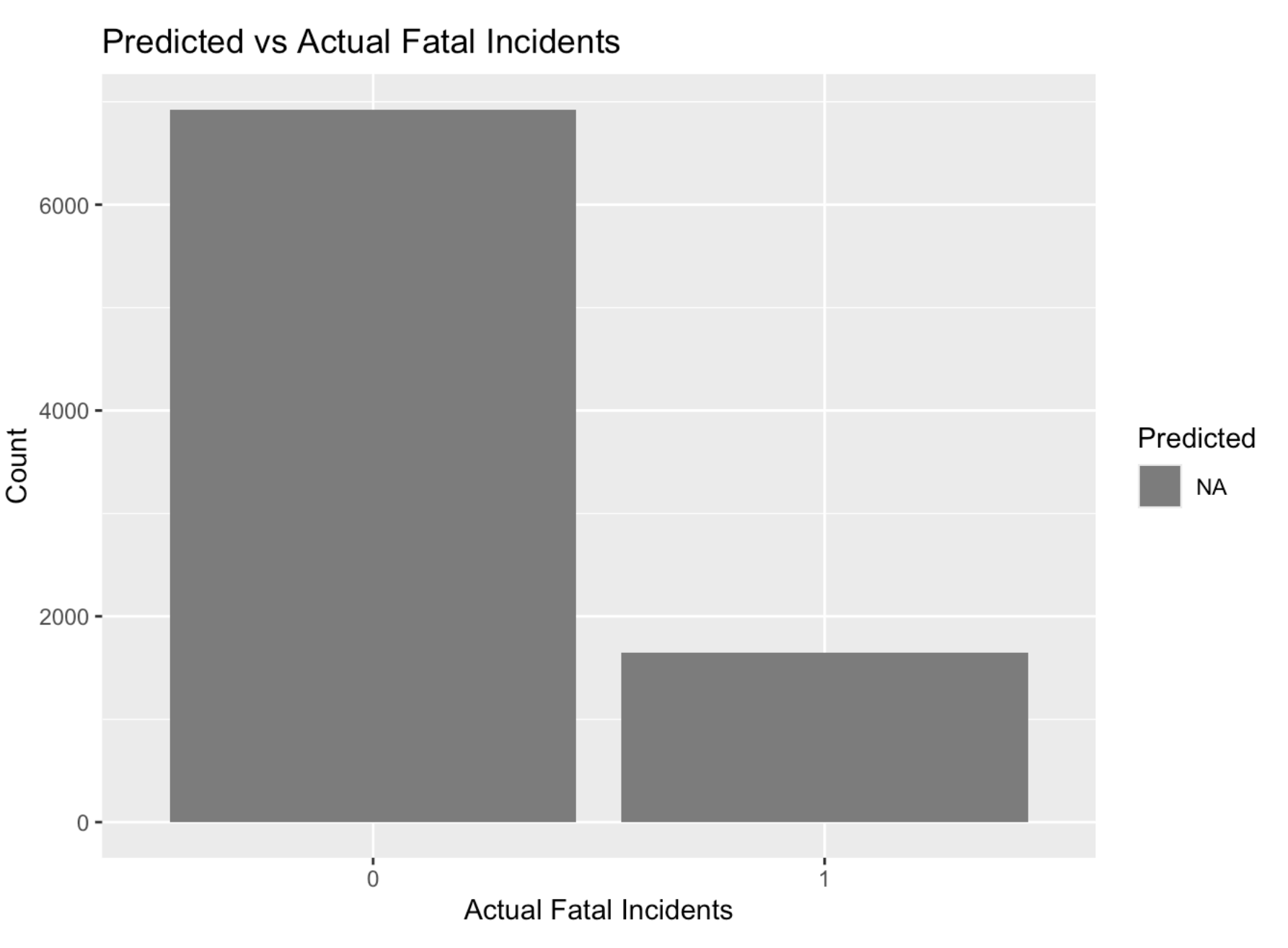
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```r
# Ensure predictions and reference are factors with the same levels
predictions <- factor(predictions, levels = levels(test_data$FATAL))
test_data$FATAL <- factor(test_data$FATAL, levels = levels(predictions))
```

```r
# Evaluate the model
confusionMatrix(predictions, test_data$FATAL)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 0 0
##          1 0 0
##
##                Accuracy : NaN
##                  95% CI : (NA, NA)
##     No Information Rate : NA
##     P-Value [Acc > NIR] : NA
##
##                   Kappa : NaN
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity :  NA
##             Specificity :  NA
##          Pos Pred Value :  NA
##          Neg Pred Value :  NA
##              Prevalence : NaN
##          Detection Rate : NaN
##    Detection Prevalence : NaN
##       Balanced Accuracy :  NA
##
##        'Positive' Class : 0
##
```

```r
# Create a visual model comparison
results <- data.frame(
  Actual = test_data$FATAL,
  Predicted = predictions
)

ggplot(results, aes(x = Actual, fill = Predicted)) +
  geom_bar(position = "dodge") +
  labs(title = "Predicted vs Actual Fatal Incidents", x = "Actual Fatal Incidents", y = "Count", fill = "Predicte
d")
```




```r
# Check for representation bias in the data
table(nypd_data$PERP_RACE)
```

```
##
##                    (null) AMERICAN INDIAN/ALASKAN NATIVE
##                      1467                              9
##       ASIAN / PACIFIC ISLANDER                      BLACK
##                       249                          17876
##            BLACK HISPANIC                        UNKNOWN
##                      2015                           2762
##                     WHITE                 WHITE HISPANIC
##                       453                           3731
```

```r
table(nypd_data$VIC_RACE)
```

```
##
## AMERICAN INDIAN/ALASKAN NATIVE       ASIAN / PACIFIC ISLANDER
##                             11                            440
##                          BLACK                 BLACK HISPANIC
##                          20235                           2795
##                        UNKNOWN                          WHITE
##                             70                            728
##                 WHITE HISPANIC
##                           4283
```

```r
# Discuss potential biases and their impact on the analysis
# The NYPD Shooting Incident dataset may exhibit several inherent
#biases that could affect the analysis. Reporting bias is a primary concern,
#as not all shooting incidents may be reported or recorded accurately,
#potentially leading to underreporting or selective reporting. Additionally,
#data collection bias might arise from the subjective nature of how data is
#categorized, including racial classifications and incident descriptions.
#Survivor bias is also a factor, as the dataset only includes recorded
#incidents, leaving out unreported cases where victims or witnesses did not
#inform authorities. Geographical bias may occur if certain boroughs or
#neighborhoods are overrepresented or underrepresented due to variations
#in policing, socioeconomic factors, or community relations with law
#enforcement. Lastly, temporal bias could influence trends over time,
#reflecting changes in law enforcement practices, policies, and socio-political
#factors that affect the number and type of reported incidents.
#Recognizing these biases is essential for accurately interpreting the
#results and understanding the limitations of the analysis.
```