nypd2

cp santos

2023-06-12

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Load the dataset

```
# readr::read_csv("./Downloads/NYPD_Shooting_Incident_Data__Historic_.csv")
library(readr)
library(ggplot2)
library(dplyr)
```

```
##
## 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(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(maps)
nypd_arrests <-read_csv("./Downloads/NYPD_Shooting_Incident_Data__Historic_.csv")
```

```
## Rows: 27312 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.
```

Data Cleaning and processing

```
nypd_arrests <- na.omit(nypd_arrests)
```

```
nypd_arrests$OCCUR_DATE <- as.Date(nypd_arrests$OCCUR_DATE)
nypd_arrests$OCCUR_TIME <- as.POSIXct(nypd_arrests$OCCUR_TIME, format = "%H:%M:%S")
```

```
nypd_arrests$year <- year(nypd_arrests$OCCUR_DATE)
nypd_arrests$month <- month(nypd_arrests$OCCUR_DATE, label = TRUE)
```

Data Summary

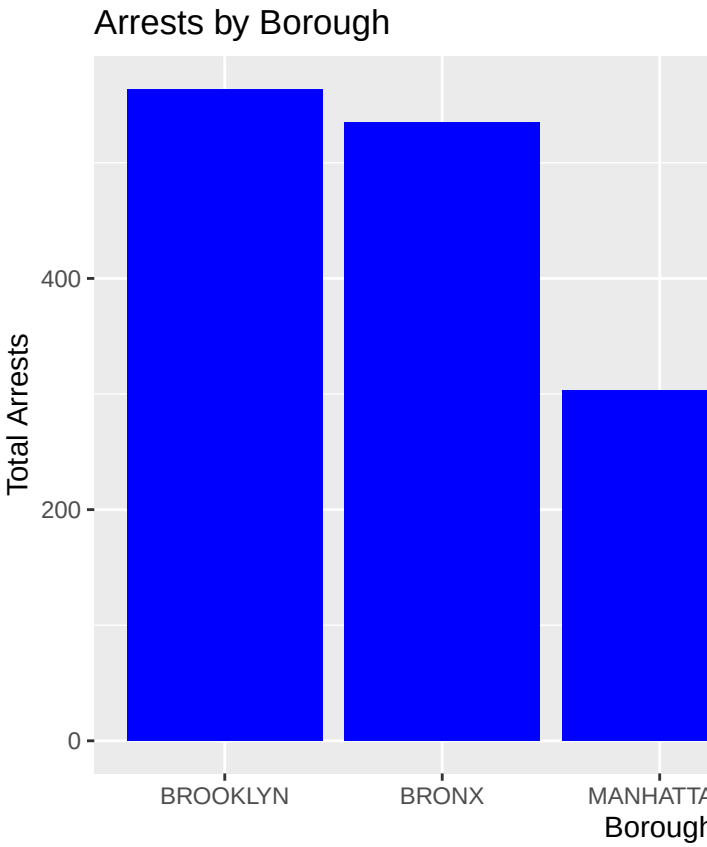
```
summary(nypd_arrests)
```

```
##   INCIDENT_KEY         OCCUR_DATE           OCCUR_TIME
##  Min.   :238531159   Min.   :0001-01-20   Min.   :1970-01-01 00:01:00.0000
##  1st Qu.:243566417   1st Qu.:0004-08-20   1st Qu.:1970-01-01 04:15:00.0000
##  Median :247200780   Median :0006-12-20   Median :1970-01-01 15:30:00.0000
##  Mean   :247286170   Mean   :0006-10-28   Mean   :1970-01-01 13:14:53.9507
##  3rd Qu.:250665115   3rd Qu.:0009-01-20   3rd Qu.:1970-01-01 20:20:00.0000
##  Max.   :261190187   Max.   :0012-12-20   Max.   :1970-01-01 23:58:00.0000
##                      NA's   :1017
##      BORO           LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE
##  Length:1706        Length:1706        Min.   :  1.00   Min.   :0.0000
##  Class :character   Class :character   1st Qu.: 42.00   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median : 60.00   Median :0.0000
##                                        Mean   : 62.14   Mean   :0.2585
##                                        3rd Qu.: 79.00   3rd Qu.:0.0000
##                                        Max.   :123.00   Max.   :2.0000
##
##  LOC_CLASSFCTN_DESC LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Length:1706        Length:1706        Mode :logical
##  Class :character   Class :character   FALSE:1368
##  Mode  :character   Mode  :character   TRUE :338
##
##
##
##
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
```

```
##  Length:1706        Length:1706        Length:1706        Length:1706
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_SEX            VIC_RACE          X_COORD_CD        Y_COORD_CD
##  Length:1706        Length:1706        Min.   : 929510   Min.   :127539
##  Class :character   Class :character   1st Qu.:1000227   1st Qu.:184263
##  Mode  :character   Mode  :character   Median :1008245   Median :210147
##                                        Mean   :1009397   Mean   :211643
##                                        3rd Qu.:1016866   3rd Qu.:241412
##                                        Max.   :1059828   Max.   :269204
##
##     Latitude       Longitude        Lon_Lat               year
##  Min.   :40.52   Min.   :-74.20   Length:1706        Min.   : 1.000
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character   1st Qu.: 4.000
##  Median :40.74   Median :-73.91   Mode  :character   Median : 6.000
##  Mean   :40.75   Mean   :-73.91                      Mean   : 6.309
##  3rd Qu.:40.83   3rd Qu.:-73.88                      3rd Qu.: 9.000
##  Max.   :40.91   Max.   :-73.73                      Max.   :12.000
##                                                      NA's   :1017
##      month
##  Dec    :  79
##  May    :  73
##  Apr    :  66
##  Oct    :  62
##  Jan    :  60
##  (Other): 349
##  NA's   :1017
```

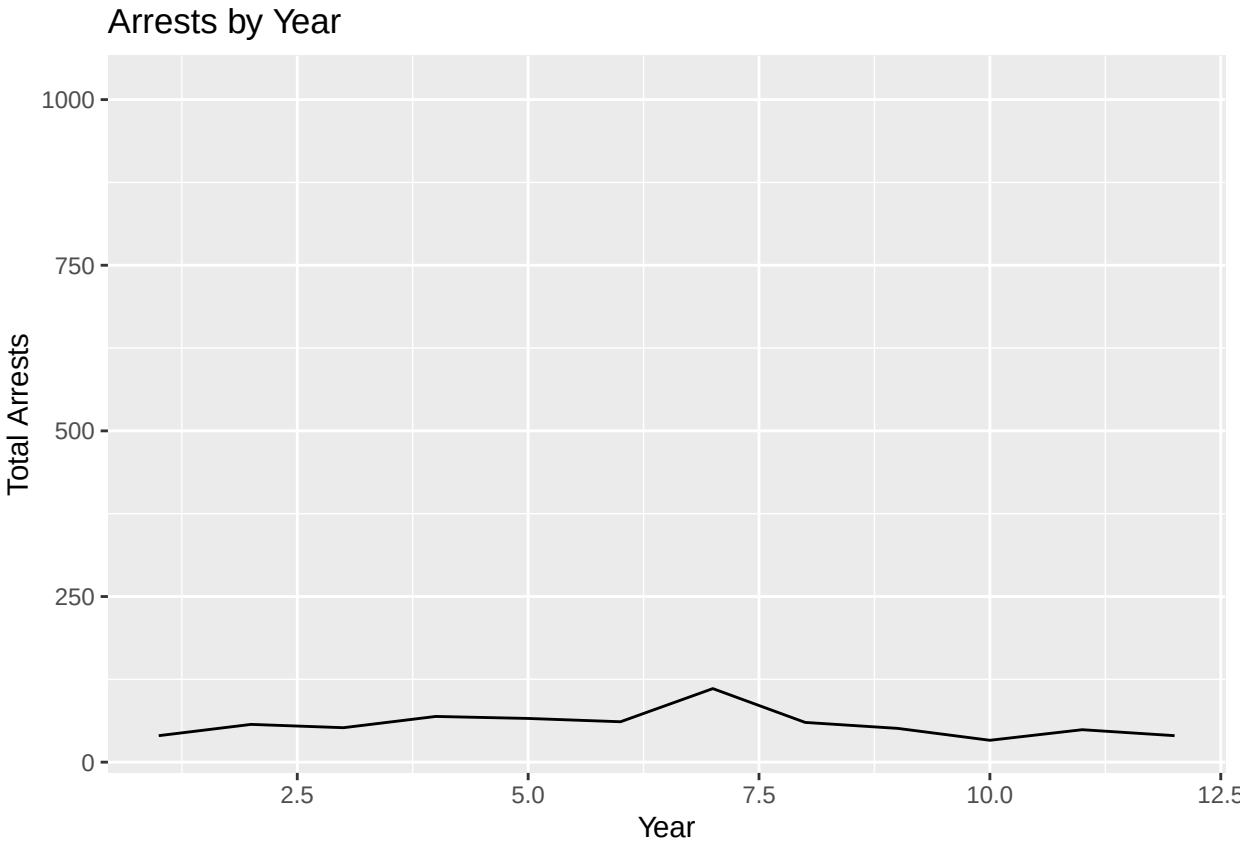
```r
arrests_by_borough <- nypd_arrests %>%
  group_by(BORO) %>%
  summarise(total_arrests = n())
```

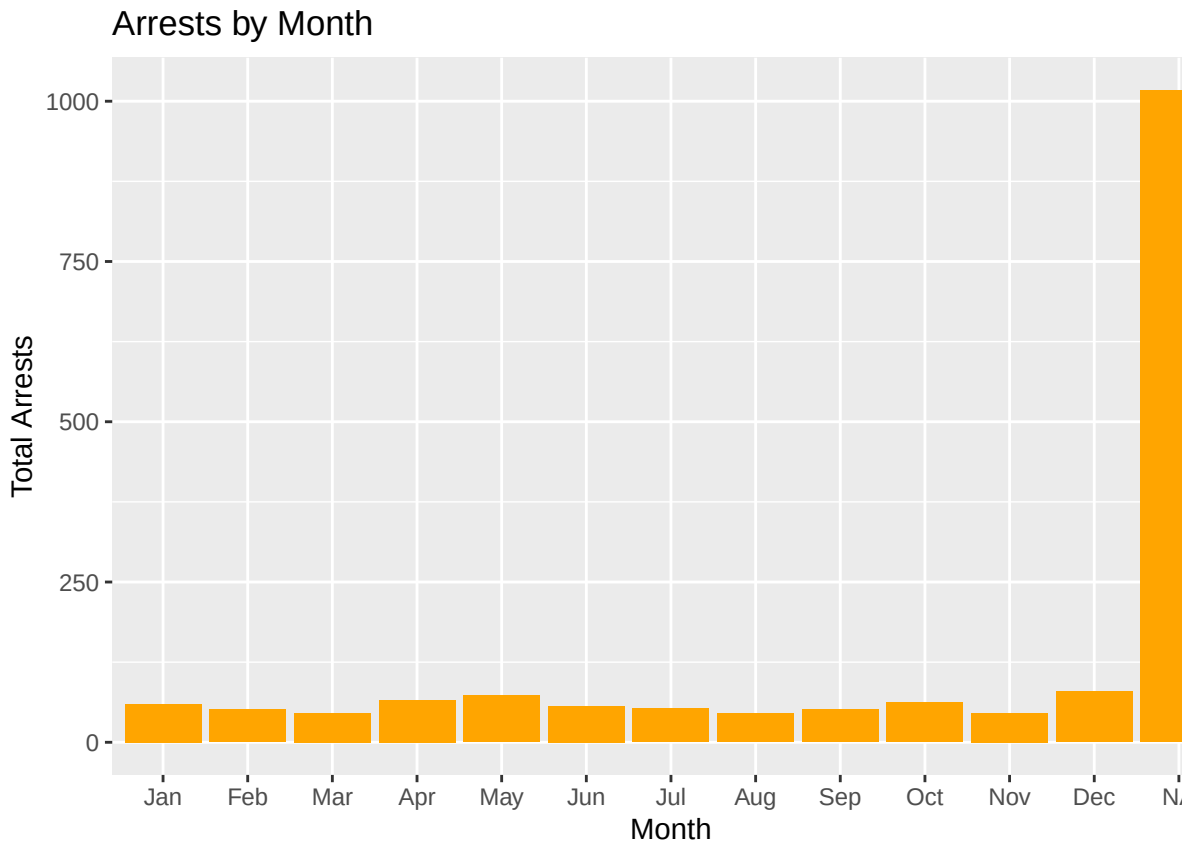
Including Plots



You can also embed plots, for example: ## Arrests by borough
Arrests by year

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```


Arrests by Year
Total Arrests
1000
750
500
250
0
2.5
5.0
7.5
10.0
12.5
Year



Arrests by month

```
# readr::read_csv("./Downloads/NYPD_Shooting_Incident_Data__Historic_.csv")
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
# Data cleaning
nypd_arrests <- na.omit(nypd_arrests)

# Data preprocessing
# Convert occur_date column to date format
nypd_arrests$OCCUR_DATE <- as.Date(nypd_arrests$OCCUR_DATE)

# Select relevant features for the model
selected_features <- c("PRECINCT", "PERP_SEX", "PERP_RACE",
                       "VIC_AGE_GROUP", "VIC_SEX", "VIC_RACE", "Longitude")

# Create a subset of data with selected features
model_data <- nypd_arrests[selected_features]

# Define the target variable
target_variable <- "Longitude"

# Split the data into training and testing sets
set.seed(123)
```

```r
train_index <- createDataPartition(model_data[[target_variable]], p = 0.8, list = FALSE)
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]

# Train the linear regression model
model <- train(
  x = select(train_data, -{{target_variable}}),
  y = train_data[[target_variable]],
  method = "lm",
  trControl = trainControl(method = "cv", number = 5)
)
```

```
## Warning: Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```r
# Make predictions on the test set
predictions <- predict(model, newdata = select(test_data, -{{target_variable}}))

# Evaluate the model
rmse <- sqrt(mean((predictions - test_data[[target_variable]])^2, na.rm = TRUE))
print(paste0("Root Mean Squared Error (RMSE): ", rmse))
```

```
## [1] "Root Mean Squared Error (RMSE): 0.0624689178873866"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.