New York Collision Data Report

Suvethika Kandasamy, Adithi Sumitran, Anirudh Murugesan, Abhinand Eedara

2023-12-10

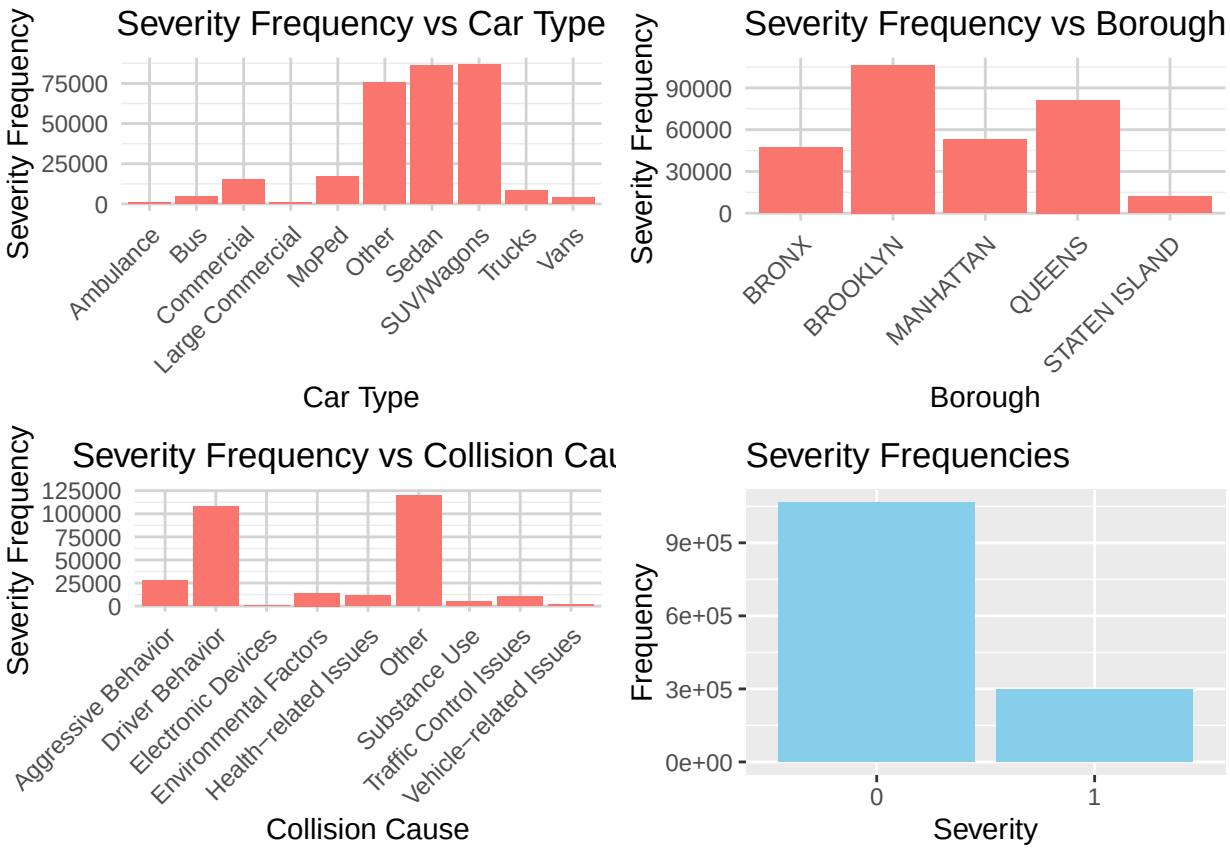
Background

Motor vehicle collisions are a leading cause of injury and death across America. In NYC, collisions pose a serious danger to public life, and the NYPD is forced to deal with vast numbers of these incidents per year. In an effort to promote a safer city for all, we ask ourselves - how can we use the police data at hand to produce actionable insights to create a safer environment on the roads of New York City.

Dataset Background

We are using the New York Motor Vehicles Collisions dataset from Kaggle. This dataset contains details of crash events, based on police reports of crashes. We want to look at the various variables presented within the dataset and analyze how they are correlated with the collisions.

```
## Total Number of Rows: 1364463
```



Methodologies

The data analysis approaches we will be utilizing in this project to get further insight on New York vehicle collisions are logistic regression models and linear discriminant analysis. The question we will be addressing in this project:

1. What kind of model can law enforcement officers utilize to predict which qualitative factors like borough, vehicle type, month, and cause most affect collision severity?
2. What insights can we derive from temporal and demographic information to predict borough-specific patterns in the amount of motor vehicle collisions in New York City?

For the first question, we intend to use the categorical variables of borough, vehicle type, month, and cause of collision to predict a binary variable, severity, which we define as any collision with one or more injury or death. As severity is binary, we will use logistic regression. We expect that this can be utilized to give law enforcement some insight of what to be on the lookout for or what time of year to be more alert while patrolling.

For the second question, we will use LDA to build a model which predicts the borough that an accident is most likely to occur in based on time of day and the accident count per capita per borough. We want to create an LDA model because we hope to aid the NYPD in resource diversion. With our model, they will be able to input a time of day and receive the borough which has the most accidents per Capita during that time, thereby allowing them to make decisions of where to send patrol units more efficiently.

Part I

To predict the severity of collisions given the different predictor variables, we will create a Severity variable. We will assign 1 to any collisions with an injury or death and 0 to the remaining collisions. We subsetted our data set into 50% train and 50% test data to build a logistic regression model with glm() for each predictor.

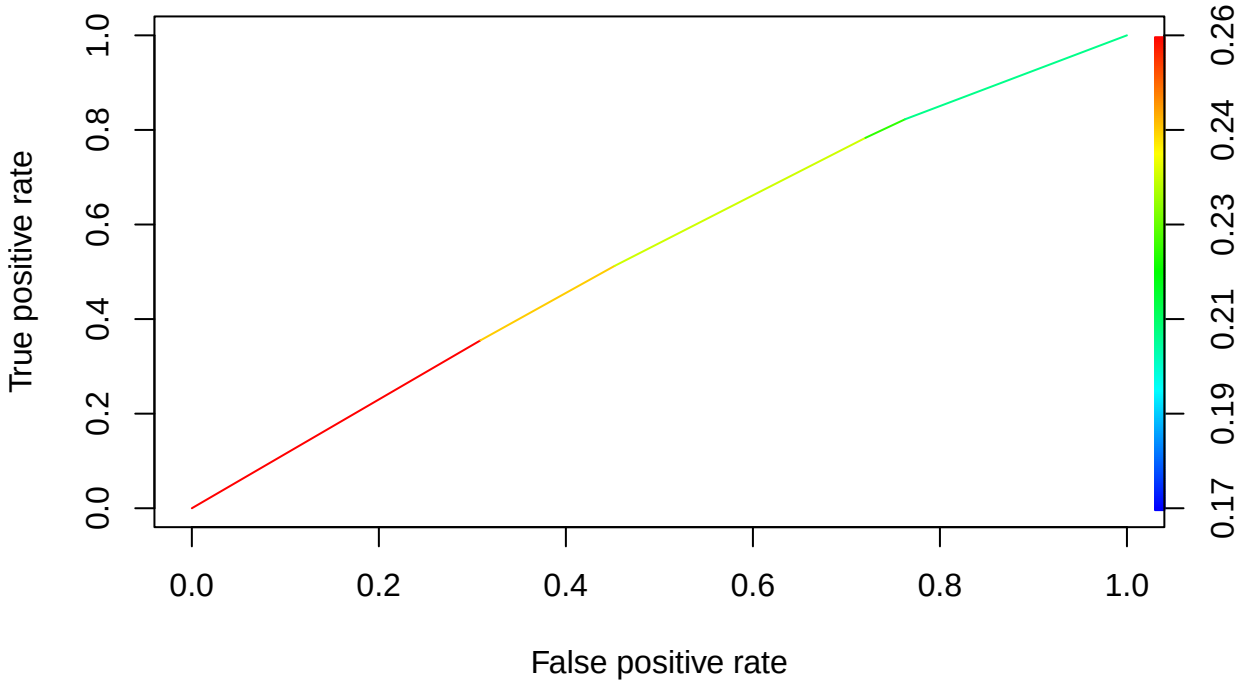
Borough Model

```
##                           Estimate  Std. Error     z value     Pr(>|z|)
## (Intercept)           -1.17457999 0.007437484 -157.927052 0.000000e+00
## BOROUGHBROOKLYN        0.04793878 0.008955053    5.353266 8.638089e-08
## BOROUGHMANHATTAN      -0.38966676 0.010055287  -38.752427 0.000000e+00
## BOROUGHQUEENS         -0.07977927 0.009322288   -8.557906 1.149358e-17
## BOROUGHSTATEN ISLAND -0.17548511 0.016406666  -10.695964 1.063086e-26
```

For the borough logistic model, the coefficients provide insights into how each level of the categorical variable 'BOROUGH' affects the log-odds of the outcome. The significance codes indicate the statistical significance of these effects as they have a p-value of less than 0.05.

Using predict(), we will use the model to predict the severity on the remaining test data with "response" argument type, which will return the probability of severity being 1 (more than one injury or death).

Using the probabilities, we can calculate the actual predictions using a threshold value, 't.' If the probability is less than the threshold, we predict that the collision is not severe. We can calculate the optimal threshold by examining the Receiver Operating Characteristic curve (ROC curve) to identify the best range for the trade-off between True Positive Rate and False Positive Rate which the ROCR package labels with colors. We can then compare the predicted values from the model with the actual data to assess the model's accuracy.



Utilizing information from this graph, we chose a t of 0.24. The package indicates good range between the trade off between TPR and FPR.

```
##     Predicted
## True      0      1
##    0 368173 164336
##    1  96585  53138
```

```
## [1] 0.617548
```

Using a threshold of 0.24, the model predicts severity of a collision with an accuracy of 61.75% which we calculated by (true_positives + true_negatives) / total_observations.

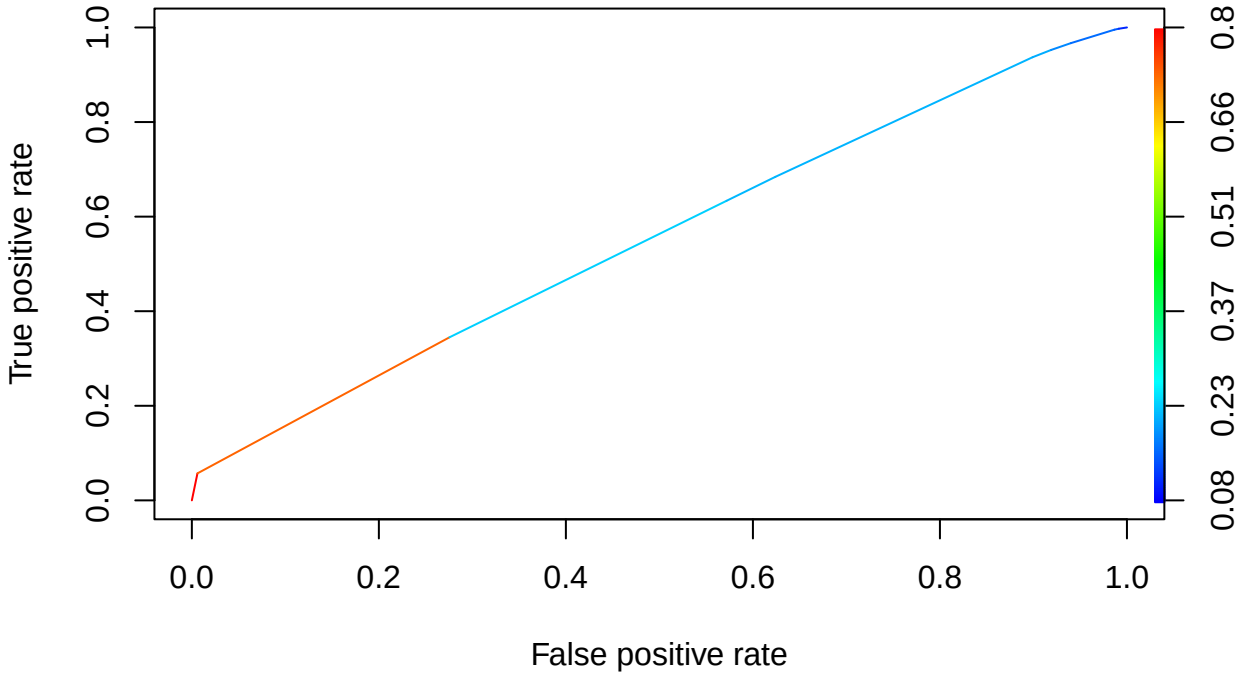
Vehicle Type Model

For the predictor vehicle type, we grouped the various vehicle types listed into ten categories such as Trucks, SUVs, etc.

```
##                                 Estimate Std. Error    z value      Pr(>|z|)
## (Intercept)                   -2.0578024 0.05872478 -35.041463 5.260314e-269
## NewVehicleTypeBus              0.5512419 0.06304186   8.744062  2.248769e-18
## NewVehicleTypeCommercial       0.7448576 0.06013813  12.385780  3.120344e-35
## NewVehicleTypeLarge Commercial -0.3270828 0.07768233  -4.210518  2.547862e-05
## NewVehicleTypeMoPed             3.0487366 0.06226644  48.962762  0.000000e+00
## NewVehicleTypeOther             0.7088700 0.05900678  12.013364  3.022918e-33
```

```
## NewVehicleTypeSedan              0.8504226 0.05898239  14.418245  3.973169e-47
## NewVehicleTypeSUV/Wagons         0.7671176 0.05897423  13.007674  1.106592e-38
## NewVehicleTypeTrucks             0.2914131 0.06103194   4.774764  1.799178e-06
## NewVehicleTypeVans               0.4113682 0.06320978   6.507984  7.616623e-11
```

In the vehicle type model, the coefficients for all of the variables are statistically significant as they have a p-value below 0.05.



Utilizing information from this graph, we chose a t of 0.45. The package indicates good range between the trade off between TPR and FPR.

```
##     Predicted
## True      0      1
##    0 529312   3197
##    1 141180   8543
```

```
## [1] 0.7883755
```

Using a threshold of 0.45, the model predicts severity of a collision with an accuracy of 78.85% which we calculated by (true_positives + true_negatives) / total_observations.

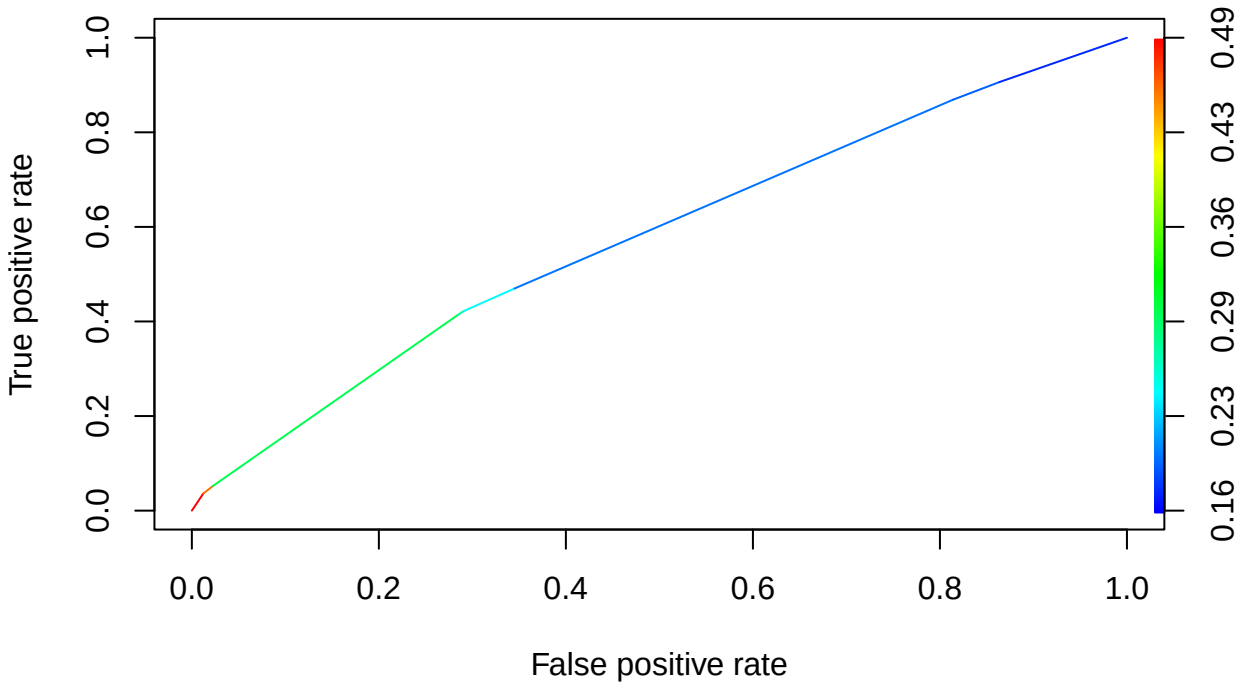
Cause Model

We created a column called Causes where we will categorize the 62 unique vehicle 1 contributing factors into 10 categories as it is too computationally expensive to compute the model with 62 categories. Factors like

glare, road conditions, and others will be categorized into environment factors. Similarly, we created We chose to only look into the vehicle 1 contributing factors as it has the most data. Additionally, a significant portion of the data set involves collisions with only one vehicle, making it a reasonable choice for initial analysis.

```
##                                Estimate  Std. Error    z value      Pr(>|z|)
## (Intercept)                -1.63761861 0.009187974 -178.23501  0.000000e+00
## CauseDriver Behavior        0.68669995 0.010500827   65.39485  0.000000e+00
## CauseElectronic Devices     0.51131361 0.060686583    8.42548  3.592731e-17
## CauseEnvironmental Factors  0.25206688 0.016161635   15.59662  7.675069e-55
## CauseHealth-related Issues  0.09784849 0.017345209    5.64124  1.688299e-08
## CauseOther                  0.21354953 0.010246433   20.84135  1.826057e-96
## CauseSubstance Use          0.79488072 0.026303272   30.21984 1.299795e-200
## CauseTraffic Control Issues 1.45842733 0.020622624   70.71978  0.000000e+00
## CauseVehicle-related Issues 0.60727059 0.038327416   15.84429  1.539561e-56
```

In the Causes model, the coefficients for all of the variables are statistically significant as they have a p-value below 0.05.



Utilizing information from this graph, we chose a t of 0.37. The package indicates good range between the trade off between TPR and FPR.

```
##     Predicted
## True      0      1
##    0 525946   6563
##    1 144291   5432
```

```
## [1] 0.7788817
```

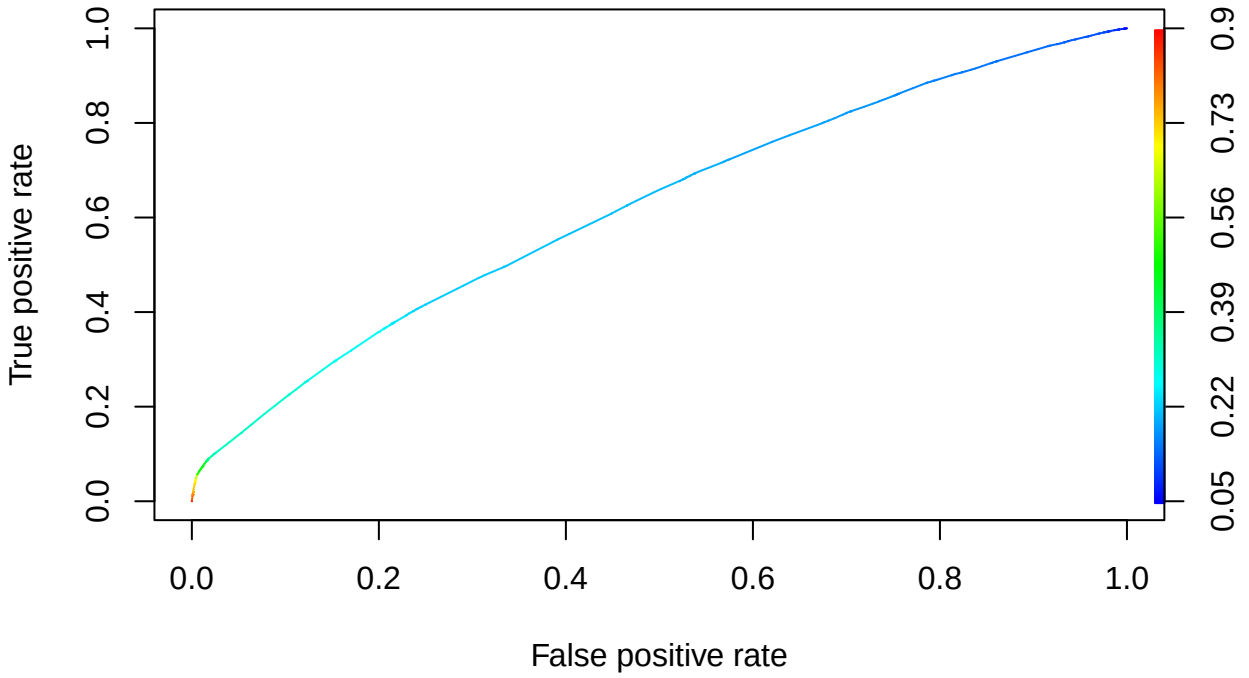
Using a threshold of 0.37, the model predicts severity of a collision with an accuracy of around 77% which we calculated by (true_positives + true_negatives) / total_observations.

Combined Model

We will also look at the logistic model using the three previous predictors combined into one logistic regression model.

```
##                                  Estimate  Std. Error    z value      Pr(>|z|)
## (Intercept)                   -2.29226235 0.060020078 -38.191592  0.000000e+00
## BOROUGHBROOKLYN                0.03086255 0.009174677   3.363884  7.685382e-04
## BOROUGHMANHATTAN              -0.45211757 0.010549219 -42.857919  0.000000e+00
## BOROUGHQUEENS                 -0.12476086 0.009567303 -13.040337  7.213734e-39
## BOROUGHSTATEN ISLAND          -0.19185719 0.016687778 -11.496869  1.367883e-30
## NewVehicleTypeBus              0.59395190 0.063464861   9.358752  8.068398e-21
## NewVehicleTypeCommercial       0.89164915 0.060624985  14.707619  5.759901e-49
## NewVehicleTypeLarge Commercial -0.25294874 0.078112227  -3.238273  1.202555e-03
## NewVehicleTypeMoPed            3.04930120 0.062722299  48.615903  0.000000e+00
## NewVehicleTypeOther            0.73066511 0.059409812  12.298728  9.201189e-35
## NewVehicleTypeSedan            0.78199861 0.059370751  13.171446  1.281207e-39
## NewVehicleTypeSUV/Wagons       0.72084201 0.059361262  12.143307  6.225806e-34
## NewVehicleTypeTrucks           0.30546418 0.061431312   4.972451  6.611174e-07
## NewVehicleTypeVans             0.48047183 0.063648083   7.548881  4.390121e-14
## CauseDriver Behavior           0.69186170 0.010720882  64.534027  0.000000e+00
## CauseElectronic Devices        0.57616062 0.061434035   9.378525  6.690224e-21
## CauseEnvironmental Factors     0.28907106 0.016539073  17.478069  2.105091e-68
## CauseHealth-related Issues     0.17993030 0.018004206   9.993792  1.622527e-23
## CauseOther                     0.21141080 0.010633197  19.882149  5.809124e-88
## CauseSubstance Use             0.78599116 0.026644521  29.499167 2.950695e-191
## CauseTraffic Control Issues    1.40976467 0.021037659  67.011480  0.000000e+00
## CauseVehicle-related Issues    0.61703184 0.038905537  15.859743  1.203849e-56
```

In the month model, the coefficients for all of the variables are statistically significant as they have a p-value below 0.05.



Utilizing information from this graph, we chose a t of 0.56. The package indicates good range between the trade off between TPR and FPR.

```
##      Predicted
## True      0      1
##    0 529312   3197
##    1 141180   8543
```

```
## [1] 0.7883755
```

Using a threshold of 0.56, the model predicts severity of a collision with an accuracy of around 78% which we calculated by (true_positives + true_negatives) / total_observations.

Discussion

The best predictor of severity of accident was the car type model and the combined model, with an accuracy of around 78%. Borough was the worst, which could be explained by the large number of NAs in the original data set. Around 30% of the borough values were missing, which means that there could be some stronger correlation that we are not able to uncover with these logistic models.

In general, there were plenty of missing values in the data set in key areas, such as car type, location, and primary cause of the accident. This could be a possible explanation for why our highest accuracy rate was 78% out of our logistic models.

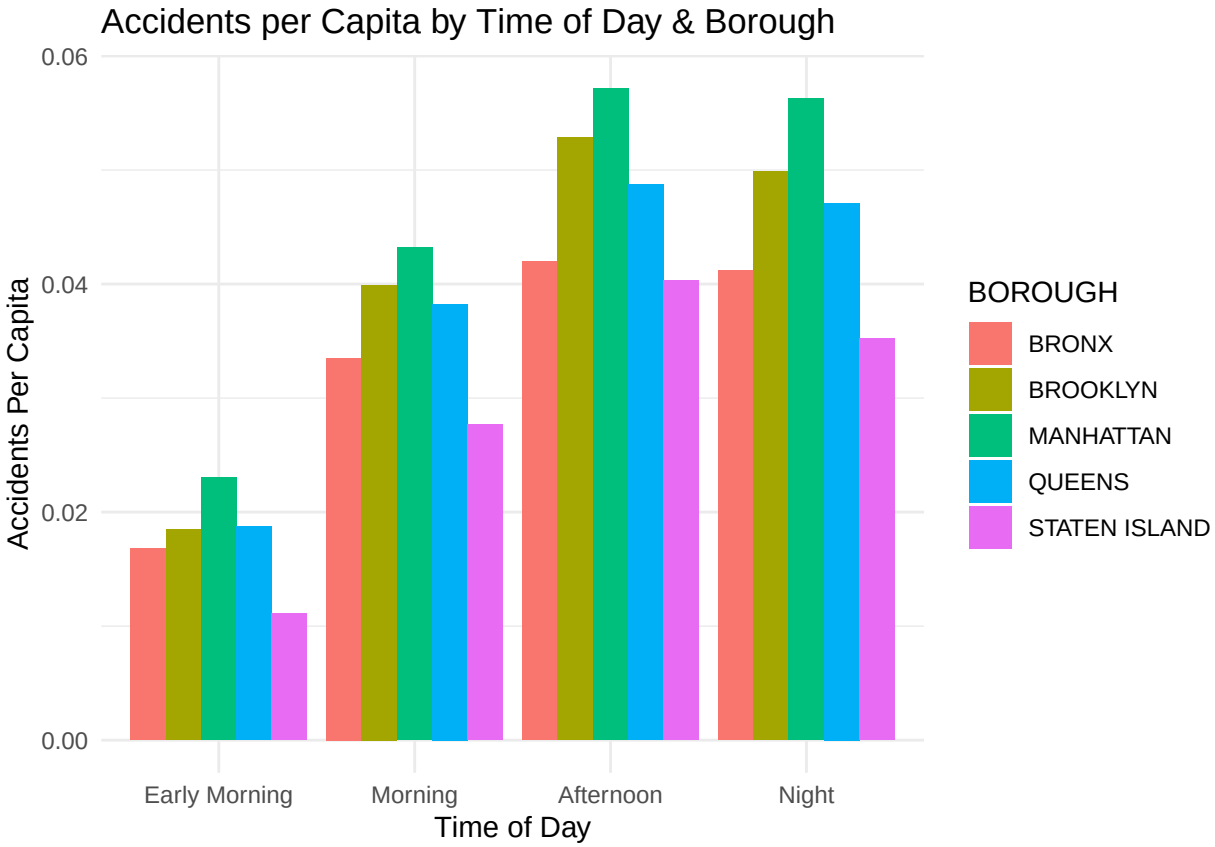
Despite the challenges posed by missing data, the models, especially the car type and combined models, still showcase potential as tools for informing law enforcement about what factors and behaviors may contribute

to the severity of accidents. Law enforcement can leverage the insights provided by these models to be more vigilant and targeted in their efforts, focusing on specific car types or behaviors that the models identify as potential indicators of higher accident severity.

As a next step, addressing data gaps through improved documentation and data collection practices could enhance the accuracy of these predictive models. Additionally, ongoing refinement and validation of the models based on updated and complete datasets would contribute to their reliability and effectiveness in supporting law enforcement efforts to prevent and mitigate severe accidents.

Part II

We suspected that Brooklyn may be more populous than the other boroughs, leading to a greater raw number of collisions. As such, we hypothesized that it would be more effective to incorporate the population of each borough and create a collisions per Capita metric for each time classification and borough. We then presented this relationship using a histogram. We decided to utilize a histogram as opposed to another method because it is effective for visualizing the distribution of the data.



This visualization provides a more accurate depiction of the collision frequencies per time classification for each borough. The addition of per Capita data as a demographic factor gives us a better understanding of the number of collisions in relation to population density, which is a more valuable metric from a city official's standpoint for police resource allocation.

The next step is splitting the data into training and testing data for our linear discriminant analysis. We then create our model using time classification and per capita as our predictor variables and borough as our output.

```
## Call:
```

```
## lda(BOROUGH ~ time_classification + per_capita, data = train_data)
##
## Prior probabilities of groups:
##         BRONX      BROOKLYN     MANHATTAN        QUEENS STATEN ISLAND
##    0.14695169    0.31872929    0.22368963    0.26870517    0.04192422
##
## Group means:
##               time_classificationMorning time_classificationAfternoon
## BRONX                          0.2500723                     0.3138597
## BROOKLYN                       0.2483272                     0.3267049
## MANHATTAN                      0.2405313                     0.3175194
## QUEENS                         0.2494450                     0.3181285
## STATEN ISLAND                  0.2416265                     0.3500804
##               time_classificationNight per_capita
## BRONX                        0.3091616 0.03642371
## BROOKLYN                     0.3104297 0.04476578
## MANHATTAN                    0.3131422 0.04913213
## QUEENS                       0.3089751 0.04190751
## STATEN ISLAND                0.3098734 0.03280248
##
## Coefficients of linear discriminants:
##                                    LD1       LD2       LD3         LD4
## time_classificationMorning   -15.89549 -2.305536 2.5069220  0.66648015
## time_classificationAfternoon -25.42006 -3.313909 0.1099583  0.02150163
## time_classificationNight     -23.86727 -2.373751 0.3793835  2.33847192
## per_capita                   804.72044 -1.711943 1.0915657 -0.32675422
##
## Proportion of trace:
## LD1 LD2 LD3 LD4
##   1   0   0   0

##                Predicted
## True             BRONX BROOKLYN MANHATTAN QUEENS STATEN ISLAND
##   BRONX          87493        0         0  12486             0
##   BROOKLYN           0   192483         0  24991             0
##   MANHATTAN          0        0    152832      0             0
##   QUEENS             0        0         0 183391             0
##   STATEN ISLAND   2749        0         0      0         25807

## [1] 0.9410377
```

We output a confusion matrix of the lda model's performance on our testing data. To calculate the accuracy of our model, we summed the diagonals of the matrix and divided that value by the sum of all values in the matrix. We calculated a 94% accuracy rate for our model.

Discussion

Through comprehensive data preprocessing and the creation of a linear discriminant analysis model, we were able to conclude we could derive predictive, actionable insights for borough-specific collision frequency patterns given temporal and demographic data. We started by creating a qualitative time classification category for each data point. After examining the distribution of the raw collision frequency data, we realized that a per Capita of collision frequency would provide a more detailed look at collision patterns based on population densities in each borough.

The linear discriminant analysis model that we created performed with an accuracy of 94.07% in predicting the borough of an accident given one of the time time classifications as well as a collisions per Capita metric. This signifies a resounding success in our process, as we now shift our intent towards applying this model to increase road safety in the city. Using our model, city officials can input the time of day and previously collected collisions per Capita data to predict accident locations. This would allow for them to effectively plan for the future allocation of police & patrol resources throughout the five boroughs.

Code Appendix

```
Motor_Vehicle_Collisions <- read.csv("/Users/Suvethika/Desktop/Motor_Vehicle_Collisions\ 2.csv")
#Creating Severity Column
Motor_Vehicle_Collisions$Severity <-
  ifelse(Motor_Vehicle_Collisions$NUMBER.OF.PERSONS.INJURED
> 0 | Motor_Vehicle_Collisions$NUMBER.OF.PERSONS.KILLED > 0, 1, 0)
#Cleaning Vehicle Type and Creating New Column
library(dplyr)
categorize_vehicle <- function(vehicle) {
  if (grepl("tru|deliv|dump|garb|tow|pick|flat|fire|fdny|box|dump|USPS|trac",
            vehicle, ignore.case = TRUE)) {
    return("Trucks")
  } else if (grepl("am", vehicle, ignore.case = TRUE) && !grepl("tru", vehicle,
                                                                ignore.case = TRUE)) {
    return("Ambulance")
  } else if (grepl("spor|wag", vehicle, ignore.case = TRUE)) {
    return("SUV/Wagons")
  } else if (grepl("van", vehicle, ignore.case = TRUE) && !grepl("tru", vehicle,
                                                                 ignore.case = TRUE)) {
    return("Vans")
  } else if (grepl("bus|scho", vehicle, ignore.case = TRUE)) {
    return("Bus")
  } else if (grepl("sedan|3-|conv", vehicle, ignore.case = TRUE)) {
    return("Sedan")
  } else if (grepl("com|cab|taxi", vehicle, ignore.case = TRUE) &&
             !grepl("Large", vehicle, ignore.case = TRUE)) {
    return("Commercial")
  } else if (grepl("large", vehicle, ignore.case = TRUE)) {
    return("Large Commercial")
  } else if (grepl("cyc|bik|scoo|moped", vehicle, ignore.case = TRUE)) {
    return("MoPed")
  } else {
    return("Other")
  }
}
Motor_Vehicle_Collisions$NewVehicleType <-
  sapply(Motor_Vehicle_Collisions$VEHICLE.TYPE.CODE.1, categorize_vehicle)

Motor_Vehicle_Collisions$NewVehicleType <- factor(Motor_Vehicle_Collisions$NewVehicleType)
Motor_Vehicle_Collisions <- na.omit(Motor_Vehicle_Collisions, cols = "Severity")
#Creating Cause Variable
library(dplyr)
categorize_cause <- function(cause) {
```

```
  if (grepl("aggr|close|unsafe", cause, ignore.case = TRUE)) {
    return("Aggressive Behavior")
  } else if (grepl("Driver|right|reaction|eating|passenger", cause, ignore.case = TRUE))
    {
    return("Driver Behavior")
  } else if (grepl("pavement|view|glare|obstruction|animals|lighting|outside cars|windshield|
                   shoulders defective|other vehicular", cause, ignore.case = TRUE)) {
    return("Environmental Factors")
  } else if (grepl("steering failure|brakes defective|tires failure|accelerator defective|
                   headlights defective|lane marking|tow hitch|vandalism",
                   cause, ignore.case = TRUE)) {
    return("Vehicle-related Issues")
  } else if (grepl("illness|consciousness|fatigued/drowsy|physical disability|prescription",
                   cause, ignore.case = TRUE)) {
    return("Health-related Issues")
  } else if (grepl("alcohol involvement|drugs", cause, ignore.case = TRUE)) {
    return("Substance Use")
  } else if (grepl("Device|texting|cell phone|headphones", cause, ignore.case = TRUE)) {
    return("Electronic Devices")
  } else if (grepl("Traffic control", cause, ignore.case = TRUE)) {
    return("Traffic Control Issues")
  } else {
    return("Other")
  }
}
Motor_Vehicle_Collisions$Cause <-
  sapply(Motor_Vehicle_Collisions$CONTRIBUTING.FACTOR.VEHICLE.1, categorize_cause)

Motor_Vehicle_Collisions$Cause <- factor(Motor_Vehicle_Collisions$Cause)
library(dplyr)
summary_df <- Motor_Vehicle_Collisions %>%
  group_by(NewVehicleType) %>%
  summarize(Severity_Frequency = sum(Severity))
#Creating dataset visualization of severity frequency vs car type
suppressMessages(library(gridExtra))
library(ggplot2)
cat("Total Number of Rows:", nrow(Motor_Vehicle_Collisions))
Severity_CarType_Graph = ggplot(summary_df, aes(x = NewVehicleType,
                                                y = Severity_Frequency,
                                                fill = "skyblue")) +
 geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Severity Frequency vs Car Type",
       x = "Car Type",
       y = "Severity Frequency") +
  theme_minimal() +
   theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    panel.grid.major = element_line(color = "lightgray", size = 0.5),
    panel.border = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )
```

```r
#Creating dataset visualization of severity frequency vs borough
summary2_df <- Motor_Vehicle_Collisions %>%
  filter(BOROUGH != "") %>%
  group_by(BOROUGH) %>%
  summarize(Severity_Frequency = sum(Severity))

Severity_Borough_Graph = ggplot(summary2_df, aes(x = BOROUGH,
                                                 y = Severity_Frequency,
                                                 fill = "skyblue")) +
 geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Severity Frequency vs Borough",
       x = "Borough",
       y = "Severity Frequency") +
  theme_minimal() +
   theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    panel.grid.major = element_line(color = "lightgray", size = 0.5),
    panel.border = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )

#Creating dataset visualization of severity frequency vs cause
summary3_df <- Motor_Vehicle_Collisions %>%
  group_by(Cause) %>%
  summarize(Severity_Frequency = sum(Severity))

Severity_Cause_Graph = ggplot(summary3_df, aes(x = Cause,
                                               y = Severity_Frequency,
                                               fill = "skyblue")) +
 geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Severity Frequency vs Collision Cause",
       x = "Collision Cause",
       y = "Severity Frequency") +
  theme_minimal() +
   theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    panel.grid.major = element_line(color = "lightgray", linewidth = 0.5),
    panel.border = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )


#Creating dataset visualization of the severity binary variable that we created
summary4_df <- data.frame(Severity = factor(Motor_Vehicle_Collisions$Severity))

Severity_Bar_Graph = ggplot(summary4_df, aes(x = Severity)) +
  geom_bar(fill = "skyblue") +
  labs(title = "Severity Frequencies", x = "Severity",
       y = "Frequency")

#Plotting all 4 graphs in a matrix
```

```r
grid.arrange(Severity_CarType_Graph, Severity_Borough_Graph,
             Severity_Cause_Graph, Severity_Bar_Graph,
             ncol=2)
#Borough Data Cleaning
Borough <- Motor_Vehicle_Collisions[Motor_Vehicle_Collisions$BOROUGH != "",
                                    c("BOROUGH","Severity")]
Borough <- na.omit(Borough, cols = "Severity")
Borough$BOROUGH <- factor(Borough$BOROUGH)
set.seed(123)
train_index_Borough <- sample(seq_len(nrow(Borough)), size = 0.5 *
                                nrow(Motor_Vehicle_Collisions))
train_data_Borough <- Borough[train_index_Borough, ]
test_data_Borough <- Borough[-train_index_Borough, ]
#Borough Logistic Regression Model
logistic_model_Borough <- glm(Severity ~ BOROUGH, data = train_data_Borough,
                              family = "binomial")
predictions_Borough <- predict(logistic_model_Borough, newdata = test_data_Borough,
                               type = "response")
summary(logistic_model_Borough)$coefficients
#Borough Model ROC curve
library(ROCR)
prediction_obj_Borough <- prediction(predictions_Borough, test_data_Borough$Severity)
performance_obj_Borough <- performance(prediction_obj_Borough, measure = "tpr",
                                       x.measure = "fpr")
plot(performance_obj_Borough, colorize = TRUE)
predicted_labels_Borough <- ifelse(predictions_Borough > 0.24, 1, 0)
#Confusion Matrix for Borough Model
conf_matrix_Borough <- table(Actual = test_data_Borough$Severity, Predicted =
                               predicted_labels_Borough, dnn = c("True", "Predicted"))
print(conf_matrix_Borough)
true_positives_Borough <- conf_matrix_Borough[2, 2]  # Number of true positives
true_negatives_Borough <- conf_matrix_Borough[1, 1]  # Number of true negatives
total_observations_Borough <- sum(conf_matrix_Borough)  # Total number of observations
# Accuracy of Borough Model
accuracy_Borough <- (true_positives_Borough + true_negatives_Borough) /
  total_observations_Borough
accuracy_Borough
#Training and Test Data Vehicle Type
set.seed(123)
train_index_Car <- sample(seq_len(nrow(Motor_Vehicle_Collisions)), size = 0.5 *
                            nrow(Motor_Vehicle_Collisions))
train_data_Car <- Motor_Vehicle_Collisions[train_index_Car, ]
test_data_Car <- Motor_Vehicle_Collisions[-train_index_Car, ]
#Vehicle Type Logistic Regression Model
logistic_model_Car <- glm(Severity ~ NewVehicleType, data = train_data_Car,
                          family = "binomial")
predictions_Car <- predict(logistic_model_Car, newdata = test_data_Car, type = "response")
summary(logistic_model_Car)$coefficients
#Vehicle Type ROC Curve
library(ROCR)
prediction_obj_Car <- prediction(predictions_Car, test_data_Car$Severity)
performance_obj_Car <- performance(prediction_obj_Car, measure = "tpr", x.measure = "fpr")
plot(performance_obj_Car, colorize = TRUE)
```

```r
predicted_labels_Car <- ifelse(predictions_Car > 0.45, 1, 0)
#Confusion Matrix for Car Type Model
conf_matrix_Car <- table(Actual = test_data_Car$Severity,
                         Predicted = predicted_labels_Car, dnn = c("True", "Predicted"))
print(conf_matrix_Car)
true_positives_Car <- conf_matrix_Car[2, 2]
true_negatives_Car <- conf_matrix_Car[1, 1]
total_observations_Car <- sum(conf_matrix_Car)
# Accuracy of Car Type Model
accuracy_Car <- (true_positives_Car + true_negatives_Car) / total_observations_Car
accuracy_Car
#Cause Training and Test Data
set.seed(123)
train_index_Cause <- sample(seq_len(nrow(Motor_Vehicle_Collisions)),
                            size = 0.5 * nrow(Motor_Vehicle_Collisions))
train_data_Cause <- Motor_Vehicle_Collisions[train_index_Cause, ]
test_data_Cause <- Motor_Vehicle_Collisions[-train_index_Cause, ]
#Cause Logistic Regression
logistic_model_Cause <- glm(Severity ~ Cause, data = train_data_Cause,
                            family = "binomial")
predictions_Cause <- predict(logistic_model_Cause, newdata = test_data_Cause,
                             type = "response")
summary(logistic_model_Cause)$coefficients
#ROC Curve for Causes
library(ROCR)
prediction_obj_Cause <- prediction(predictions_Cause, test_data_Cause$Severity)
performance_obj_Cause <- performance(prediction_obj_Cause, measure = "tpr",
                                     x.measure = "fpr")
plot(performance_obj_Cause, colorize = TRUE)
predicted_labels_Cause <- ifelse(predictions_Cause > 0.37, 1, 0)
#Confusion Matrix Cause model
conf_matrix_Cause <- table(Actual = test_data_Cause$Severity,
                           Predicted = predicted_labels_Cause, dnn = c("True", "Predicted"))
print(conf_matrix_Cause)
true_positives_Cause <- conf_matrix_Cause[2, 2]
true_negatives_Cause <- conf_matrix_Cause[1, 1]
total_observations_Cause <- sum(conf_matrix_Cause)
#Accuracy of Cause model
accuracy_Cause <- (true_positives_Cause + true_negatives_Cause) / total_observations_Cause
accuracy_Cause
#Combined model
set.seed(123)
train_index <- sample(seq_len(nrow(Motor_Vehicle_Collisions)),
                      size = 0.5 * nrow(Motor_Vehicle_Collisions))
train_data <- Motor_Vehicle_Collisions[train_index, ]
test_data <- Motor_Vehicle_Collisions[-train_index, ]
train_data <- na.omit(train_data[, c("Severity", "BOROUGH", "NewVehicleType", "Cause")])
test_data <- na.omit(test_data[, c("Severity", "BOROUGH", "NewVehicleType", "Cause")])
logistic_model <- glm(Severity ~ BOROUGH + NewVehicleType + Cause,
                      data = train_data,
                      family = "binomial")
predictions <- predict(logistic_model, newdata = test_data, type = "response")
summary(logistic_model)$coefficients
```

```r
#ROC Curve for Combined
library(ROCR)
prediction_obj <- prediction(predictions, test_data$Severity)
performance_obj <- performance(prediction_obj, measure = "tpr", x.measure = "fpr")
plot(performance_obj, colorize = TRUE)
predicted_labels <- ifelse(predictions > 0.56, 1, 0)
#Confusion Matrix Combined model
conf_matrix <- table(Actual = test_data$Severity,
                          Predicted = predicted_labels,
                          dnn = c("True", "Predicted"))
print(conf_matrix)
true_positives <- conf_matrix[2, 2]
true_negatives <- conf_matrix[1, 1]
total_observations <- sum(conf_matrix)
#Accuracy of Combined model
accuracy <- (true_positives + true_negatives) / total_observations
accuracy
library(dplyr)
#Creating a new df that removes all rows with NA values in the BOROUGH column
clean_df = Motor_Vehicle_Collisions
clean_df <- clean_df %>% filter(BOROUGH != "")
#Converting crash time into a new, numeric column of the times in decimals
clean_df[["CRASH.TIME"]] <- substr(clean_df[["CRASH.TIME"]], 1, 5)
clean_df$decimal_time <- gsub("[: -]", "." , clean_df$CRASH.TIME, perl=TRUE)
clean_df$decimal_time = as.numeric(clean_df$decimal_time)
#Using the new decimal times to classify the crash times into four qualitative categories
breaks <- c(0, 7, 12, 17, 24)
labels <- c("Early Morning", "Morning", "Afternoon", "Night")
clean_df$time_classification <- cut(clean_df$decimal_time, breaks = breaks,
                                    labels = labels, include.lowest = TRUE)
library(dplyr)
#Creating a df of accident frequencies per borough for each time classification
accident_freqs <- clean_df %>%
  group_by(BOROUGH, time_classification) %>%
  summarise(Count = n())
#Creating a new df of each borough and its corresponding population data
BOROUGH = c("QUEENS", "BRONX", "BROOKLYN", "MANHATTAN", "STATEN ISLAND")
borough_pops = c(2400000, 1500000, 2700000, 1700000, 500000)
borough_df = data.frame(BOROUGH, borough_pops)
#Merging the accident frequency and borough data frames
#and adding an accidents per capita column
per_capita_df = merge(accident_freqs, borough_df,by = "BOROUGH", all.x = TRUE)
per_capita_df$per_capita <- per_capita_df$Count / per_capita_df$borough_pops
#Visualization of collision frequencies per capita per borough for each time classification
library(ggplot2)
ggplot(per_capita_df, aes(x = time_classification, y = per_capita, fill = BOROUGH)) +
 geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Accidents per Capita by Time of Day & Borough",
       x = "Time of Day",
       y = "Accidents Per Capita") +
  theme_minimal()
#Creating a final df that merges this per capita data with the full, cleaned df
final_df = left_join(clean_df, per_capita_df, by = c("BOROUGH", "time_classification"))
```

```r
#Splitting the data in the final dataframe to create train
#& test data for linear discriminant analysis
library(MASS)
set.seed(123)
train_index <- sample(seq_len(nrow(final_df)), size = 0.5 * nrow(final_df))
train_data <- final_df[train_index, ]
test_data <- final_df[-train_index, ]
#Running linear discriminant analysis using the training & test data
borough_lda = lda(BOROUGH~time_classification + per_capita, data = train_data)
borough_lda
#Creating a confusion matrix of the LDA
predicted <- predict(borough_lda, test_data)
confusion_matrix <- table(test_data$BOROUGH, predicted$class, dnn = c("True", "Predicted"))
confusion_matrix
#Calculating the accuracy of our model
sum(diag(confusion_matrix))/sum(confusion_matrix)
```