TrafficStopsDraft

2023-11-30

Introduction

The source of the data is from the Burlington Government website and is an observational study. This data was collected from 2012 until October of 2023 and records: “Burlington Police Department Traffic Stops through 10/31/23.” The sample shows data from 2012 until 2023 and the sample was selected through the records kept from the Burlington Police Department. In terms of bias, we found little that was significant, or conveys the message that there may have been some sort of bias. Measurements were taken through recording daily statistics from traffic stops within Burlington. No data is unbiased, but we did not find any meaningful bias in the questions or measurements. This data is interesting because it analyzes the trends that are shown in traffic stops that directly affect citizens and is one of the most common interactions between citizens and the police. The police and citizen interaction in a city like Burlington is important in determining a number of factors like property value and how safe residents feel on a daily basis. Our data cleaning is as shown below.

Color Scheme

This is our the color scheme we used for the whole document

Data Cleaning

From our data we decided to look at factors that affected stop_outcome and stop_type.

Our factors:

Gender: Male, Female, Other

Races: Black, Asian, Hispanic, White

Stop Type: Moving Violation, Investigatory, Susp DUI, Equipment, Other, External

Stop_Outcome: Warning, Ticket, Arrest, No Action

```
traffic_graph <-
  trafficstops %>%
  filter(
    call_type == "Traffic", # only selecting traffic types
    veh_state == "VT", # only chooses Vermont Licenses Plates
    !is.na(stop_type) # This removes NA values within the stop types

  ) %>%

  mutate(call = ymd_hms(call_time),
         year = year(call),
         month = month(call),     # Keep just the month with the name of month (label = T)
         weekday = wday(call, label = TRUE), # Keep the day of the week with name
         hour = hour(call),                  # Keeps just the hour
         minute = minute(call),              # Keeps just the minute
         date = date(call),                  # Keeps just the date - no time
         day = yday(call),                   # Returns the day of the year (Dec 31st = 365)

         time = hour*60 + minute,

         #cleaning name for stop_type
         stop_type = case_when(stop_type == "M = Moving violation" ~ "Moving Violation",
                               stop_type == "I = Investigatory" ~ "Investigatory",
                               stop_type == "D = Susp DUI" ~ "Susp DUI",
                               stop_type == "V = Vehicle Equipment" ~ "Equipment",
                               stop_type == "O = Other" ~ "Other",
                               stop_type == "E = Externally Generated" ~ "External"),

         #cleaning name for stop_outcome
         stop_outcome = case_when(stop_outcome == "W = Warning" ~ "Warning",
                                 stop_outcome == "T = Ticket" ~ "Ticket",
                                 stop_outcome == "A = Arrest for Violation" | stop_outcome == "A
W = Arrest Warrant" ~ "Arrest",
                                 stop_outcome == "N = No action taken" ~ "No Action"),

         #cleaning name for gender
         gender = case_when(gender == "Female - F" ~ "Female",
                            gender == "Male - M" ~ "Male",
                            gender == "Transgender - T" | gender == "Non-Binary/Other - X" |
                              gender == "Unknown - U" ~ "Other")
         ) %>%

  select(
    year, month, weekday, hour, minute, race, gender, age, stop_type, stop_outcome, time
    )

tibble(traffic_graph)
```

```
## # A tibble: 29,583 × 11
##     year month weekday  hour minute race  gender   age stop_type    stop_outcome
##    <dbl> <dbl> <ord>   <int>  <int> <chr> <chr>  <int> <chr>        <chr>
##  1  2012     1 Wed        14     43 White Female    23 Moving Viol… Warning
##  2  2012     1 Sun         0     18 White Female    73 Equipment    Warning
##  3  2012     1 Sun         0     18 White Male      25 Susp DUI     Warning
##  4  2012     1 Sun         8     50 White Female    23 Equipment    Warning
##  5  2012     1 Sun         8     56 White Female    17 Other        Warning
##  6  2012     1 Sun         9      4 White Male      47 Equipment    Warning
##  7  2012     1 Wed        14     50 White Female    29 Equipment    Ticket
##  8  2012     1 Wed        15     13 White Male      28 Moving Viol… Warning
##  9  2012     1 Sun         9     19 White Female    34 Equipment    Warning
## 10  2012     1 Sun        10     27 White Male      23 Moving Viol… Warning
## # ℹ 29,573 more rows
## # ℹ 1 more variable: time <dbl>
```

Breaking Down the Traffic Stop Outcomes

Traffic stops occur all the time, however, how often are people arrested or given a ticket or given a warning?

This graph below compares the number of arrests, tickets, and warnings.

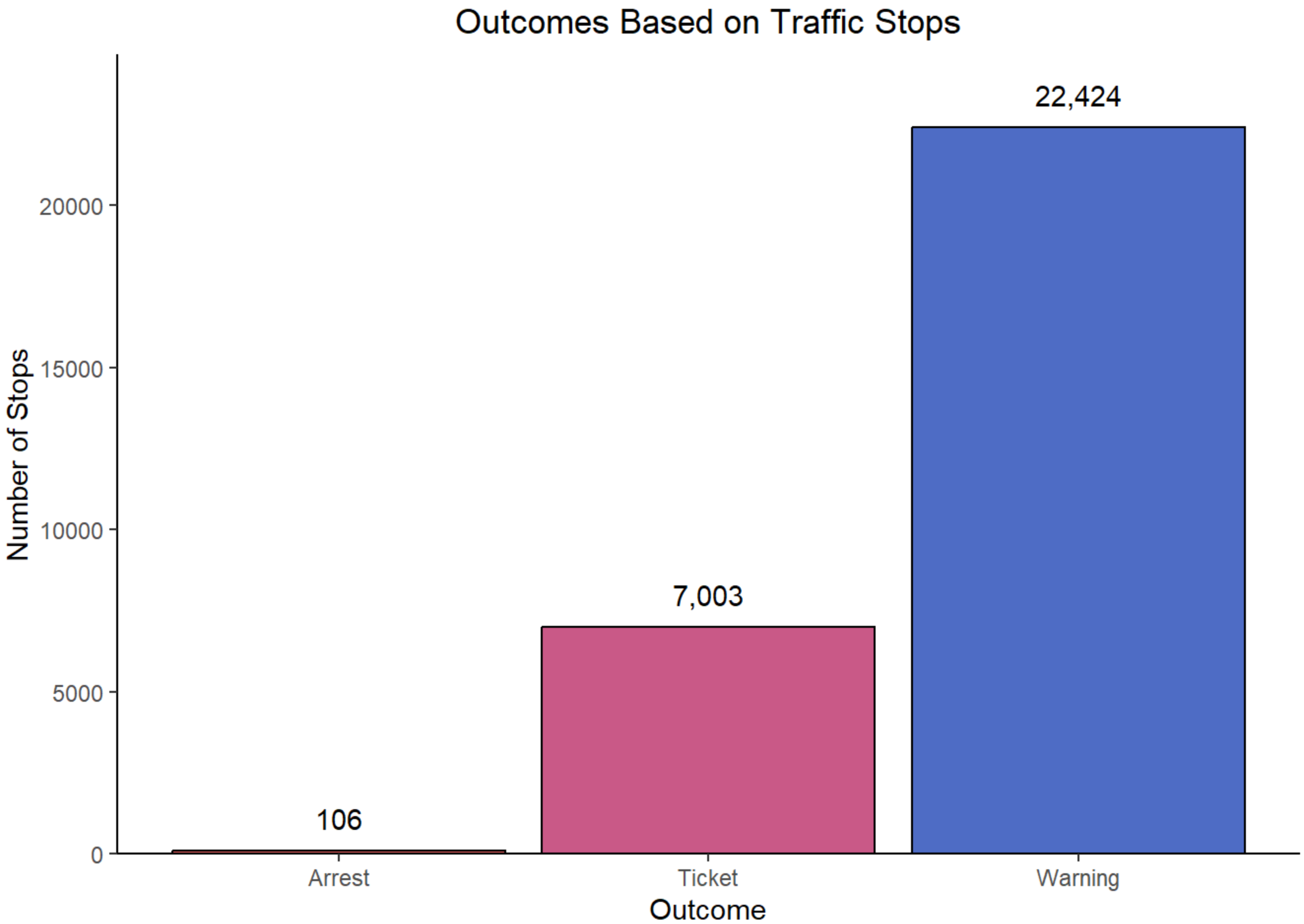
```
traffic_arrests <-
  traffic_graph %>%
  filter(stop_outcome == "Arrest" | stop_outcome == "Ticket" | stop_outcome == "Warning") %>%
  group_by(stop_outcome) %>%
  summarize(total = n())

traffic_arrests %>%
  pivot_wider(names_from = stop_outcome, values_from = total) %>%

  gt() %>%
  tab_header(title = "Total Amount of Each Outcomes")
```

Total Amount of Each Outcomes

Arrest	Ticket	Warning
106	7003	22424

We also graphed the table as a bar chart to visibly observe the difference between the two.

```
ggplot(data = traffic_arrests, mapping = aes(x = stop_outcome, y = total)
) +

  geom_col(
    fill = c("#e86161", "#C95987", "#4e6cc5"),
    color = "black"
  ) +

  scale_y_continuous(
    expand = c(0, 0, .05, 0)
  ) +

  theme_classic() +

  #add labels
  labs(
    x = "Outcome",
    y = "Number of Stops",
    title = "Outcomes Based on Traffic Stops"
  ) +

  theme(
    plot.title = element_text(hjust = 0.5)
        ) +

  geom_text(aes(label = scales::comma(total)), vjust = -1) +

  scale_y_continuous(expand = c(0,0,0.1,0))
```



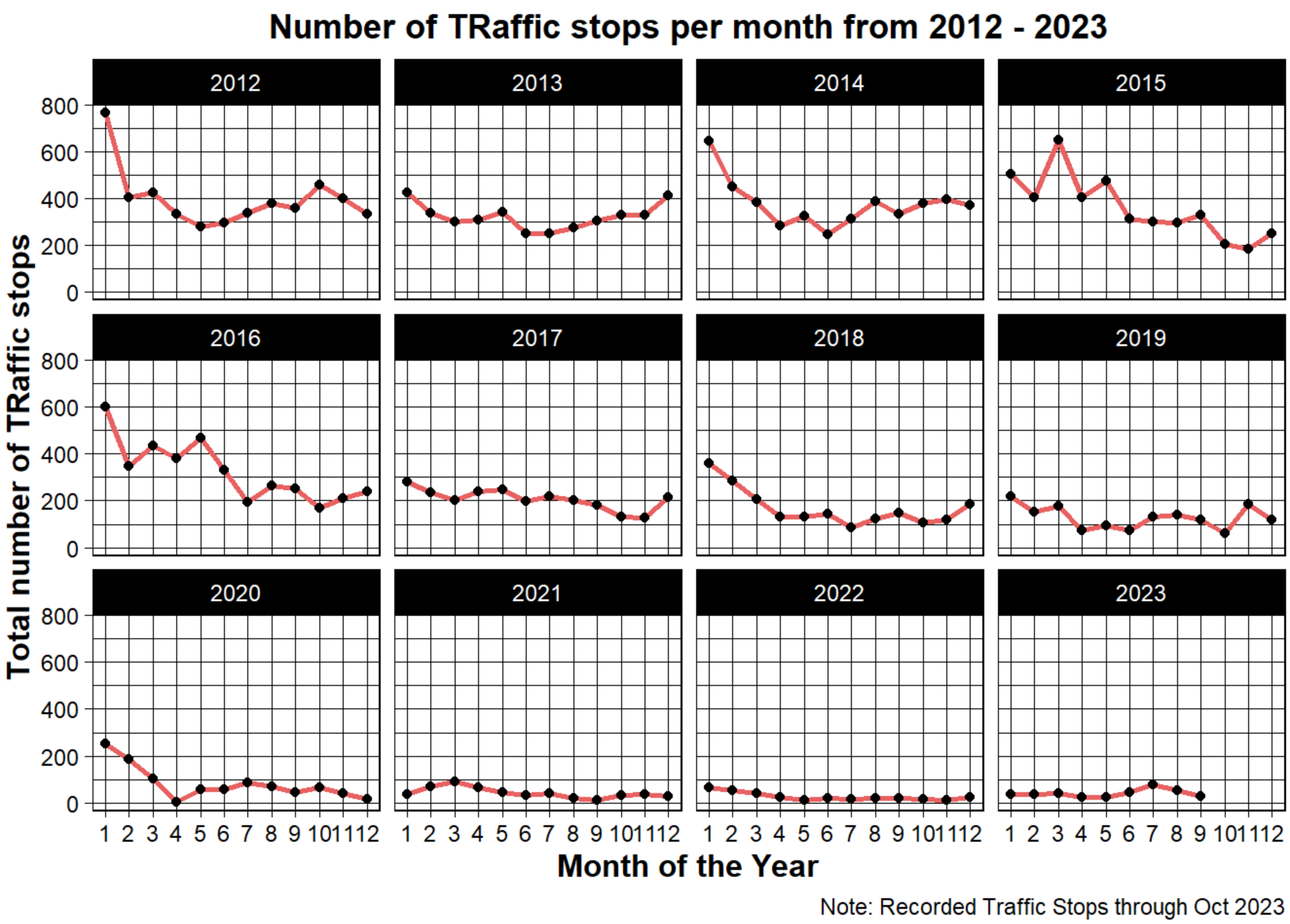
There were significantly more warnings than any other outcome, showing that Vermont police are easy on most people pulled over.

Because most stops are warnings we looked into what demographic was most likely pulled over.

Stops by month

```
traffic_graph %>%
  # Grouping by the relevant times that we are using
  group_by(month, year) %>%
  summarise(
    total = n()
  ) %>%

  # Creating the plot
  ggplot(
    mapping = aes(
      x = month,
      y = total
    )
  ) +

  geom_line(
    color = "#e86161",
    linewidth = 1
  ) +

  geom_point() +

  # Faceting based on year to make visualization easier
  facet_wrap(
    ~ year,
    scales = "fixed"
  ) +

  # Changing the scales for each month
  scale_x_continuous(
    breaks = seq(0, 12, 1),
    minor_breaks = NULL
  ) +

  labs(
    x = "Month of the Year",
    y = "Total number of TRaffic stops",
    title = "Number of TRaffic stops per month from 2012 - 2023",
    caption = "Note: Recorded Traffic Stops through Oct 2023"
  ) +

  theme_linedraw() +

  theme(plot.title = element_text(hjust = .5,
                                  face = "bold"),
        axis.title.x = element_text(face = "bold",
                                    size = 12),
        axis.title.y = element_text(face = "bold",
```

```
                                        size = 12)
        )
```

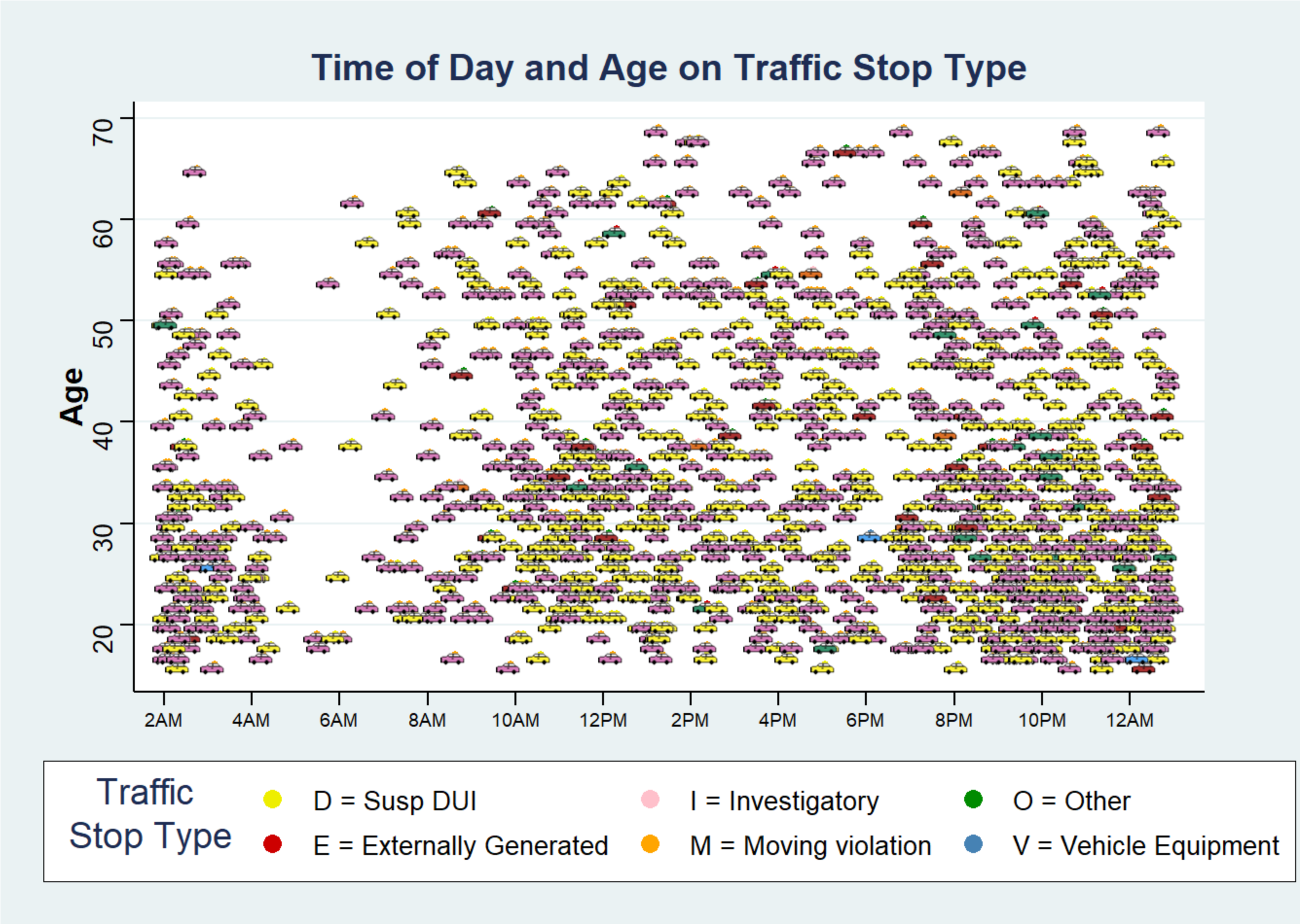


Another factor we were interested in was time. The graph below displays the number of stops by year and month. For almost all years, January had the most stops. There seems to be a steady decline over time on how many people are being pulled over. There are many factors that could be affecting this like the use of technology. While many believe technology to be bad, it is very helpful when driving between navigation, detecting if you're driving over the line or if a car is near you, and for emergencies. Additionally, there are some important events that could have influenced a decline or increase on number of stops per month. One most recently is COVID-19. COVID introduced online learning and working, causing more people to stay home, possibly explaining why after COVID there is a sharp decline in the number of stops from January to March.

Age and Time of Day

```
traffic_graph %>%
  # Setting the time statistic and also mapping each point to be a custom colored car
  mutate(time = hour*60 + minute,
         car_color = case_when(
           stop_type == "Susp DUI" ~ "/Users/ritte/OneDrive/Desktop/DS/FinalProject/cars/blueca
r.png",
           stop_type == "External" ~ "/Users/ritte/OneDrive/Desktop/DS/FinalProject/cars/greenca
r.png",
           stop_type ==  "Investigatory" ~ "/Users/ritte/OneDrive/Desktop/DS/FinalProject/cars/o
rangecar.png",
           stop_type == "Moving Violation" ~ "/Users/ritte/OneDrive/Desktop/DS/FinalProject/car
s/pinkcar.png",
           stop_type == "Other" ~ "/Users/ritte/OneDrive/Desktop/DS/FinalProject/cars/redcar.pn
g",
           stop_type == "Equipment" ~ "/Users/ritte/OneDrive/Desktop/DS/FinalProject/cars/yellow
car.png",
           .default = NA_character_)
  ) %>%

  # Filtering the age to remove outliers, and using only 5% of the data so that the graph is not
too crowded
  filter(age > 15 & age < 70,
         !is.na(stop_type)) %>%
  slice_sample(prop = .05) %>%

  ggplot(
    mapping = aes(
      x = time,
      y = age,
      image = car_color
    )
  ) +

  geom_point(
    size = 1,
    mapping = aes(
      color = stop_type),
    alpha = 1
  ) +

  # Changing the labels on the graph
  scale_color_manual(
    values = c("Yellow2", "Red3", "Pink", "Orange", "Green4", "steelblue"),
    labels = c("D = Susp DUI", "E = Externally Generated", "I = Investigatory", "M = Moving viol
ation", "O = Other", "V = Vehicle Equipment")
  ) +

  # Turning the points into images
  geom_image() +
```

```
  # making the grid of the panel blank
  theme(panel.grid = element_blank()) +

  labs(
    color = "Traffic\n Stop Type",
    x = NULL,
    y = "Age",
    title = "Time of Day and Age on Traffic Stop Type"
  ) +

  # Changing the labels to the different times of day
  scale_x_continuous(
    breaks = seq(0, 1375, 125),
    labels = c("2AM", "4AM", "6AM", "8AM", "10AM", "12PM", "2PM", "4PM", "6PM", "8PM", "10PM",
"12AM"),
    expand = c(0.03, 0, 0.03, 0)
  ) +

  scale_y_continuous(
    expand = c(0.05, 0, 0.05, 0)
  ) +

  theme_stata() +

  # Putting the legend on the bottom of the graph
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(size = 7),
    axis.title.x = element_text(face = "bold",
                                size = 12),
    axis.title.y = element_text(face = "bold",
                                size = 12),
    plot.title = element_text(hjust = .5,
                              face = "bold"),
    plot.margin = margin(t = 20, r = 40, b = 20, l = 20, unit = "pt")
  ) +
```



In this graph, we wanted to see the relationship between age and time of day on the type of traffic stop type that occurred. For this graph we only looked at five percent of the data so that we did not overcrowd the graph. In this data, it was very clear that most of the Traffic stops were due to Moving Violations and Vehicle Equipment. We do not see very many Investigatory reasons or Suspect DUI reasons either. Another one of the main takeaways that we took from this graph was that the time where there is the most activity for Traffic stops is from around 8pm to 4am at night, and we generally see this in the younger demographic. This graph shows clearly that younger people who are out later tend to get pulled over more for a variety of reasons, but mainly the fact that more of them are out on the streets. This data was not very surprising.

Stops by Race

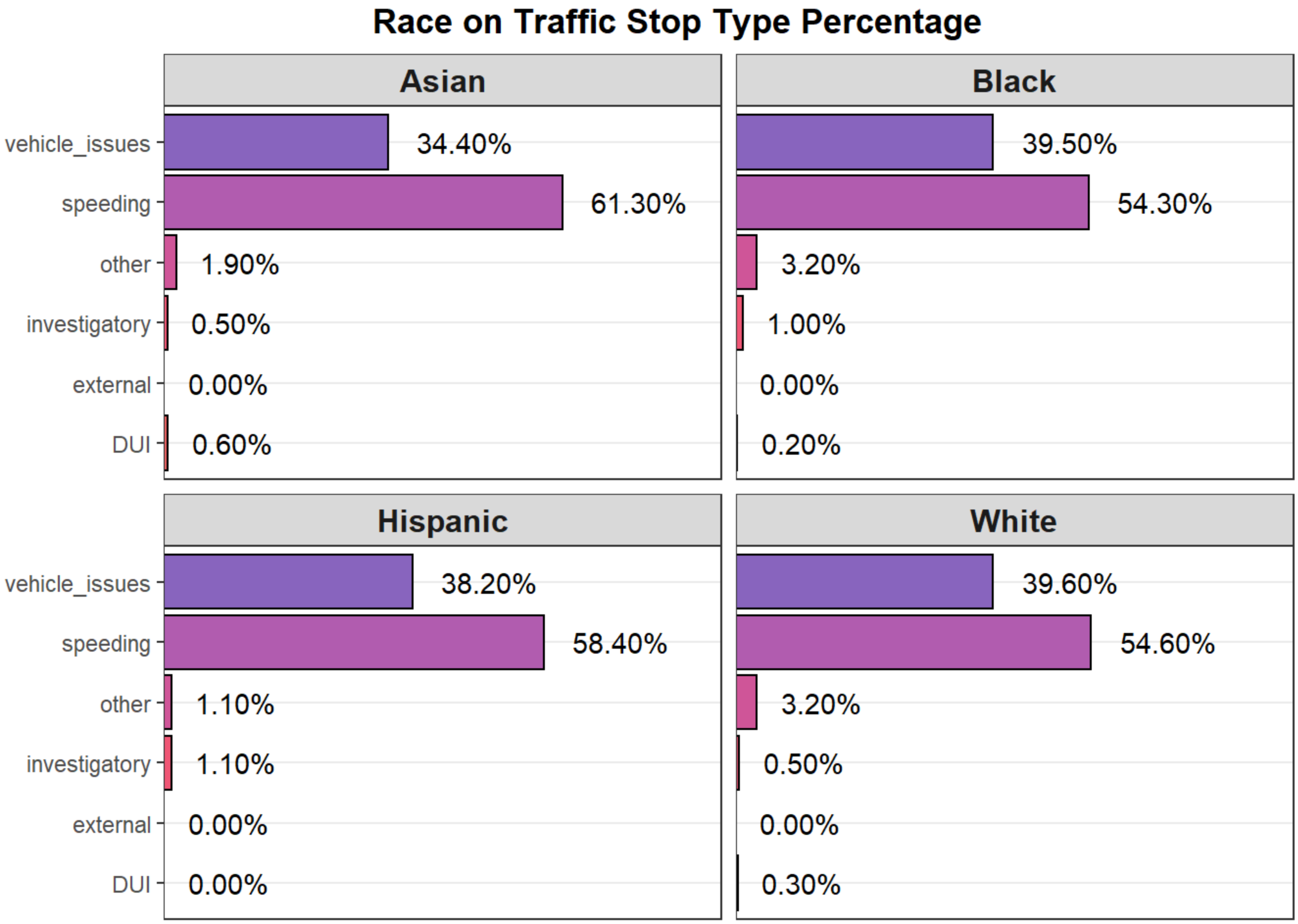
While Vermont is not very racially diverse, we were still interested if this was a factor that was significant on stop type.

To get the data for the graph below we filtered out the missing and other race categories, additionally, we pivoted the data so the type was in a column and the values for the sum was also in a column. This was done to get the total percentage of all races.

```
by_race <- traffic_graph |>
  filter(race != "Missing" & race != "Other") |>
  group_by(race) |>
  summarize(total = n(),
            speeding = sum(stop_type == "Moving Violation"),
            vehicle_issues = sum(stop_type == "Equipment"),
            external = sum(stop_type == "Externally"),
            other = sum(stop_type == "Other"),
            DUI = sum(stop_type == "Susp DUI"),
            investigatory = sum(stop_type == "Investigatory")) |>
  pivot_longer(cols = speeding:investigatory,
               names_to = "type",
               values_to = "value") |>
  mutate(percent = (value/total))

tibble(by_race)
```

```
## # A tibble: 24 × 5
##    race  total type           value percent
##    <chr> <int> <chr>          <int>   <dbl>
##  1 Asian  1278 speeding         783 0.613
##  2 Asian  1278 vehicle_issues   440 0.344
##  3 Asian  1278 external           0 0
##  4 Asian  1278 other             24 0.0188
##  5 Asian  1278 DUI                8 0.00626
##  6 Asian  1278 investigatory      7 0.00548
##  7 Black  2488 speeding        1351 0.543
##  8 Black  2488 vehicle_issues   984 0.395
##  9 Black  2488 external           0 0
## 10 Black  2488 other             79 0.0318
## # ℹ 14 more rows
```

```
ggplot(
  data = by_race,
  mapping = aes(
    x = percent,
    y = type)
  ) +

  geom_col(
    mapping = aes(fill = type),
    color = "black",
    show.legend = F
    ) +

  scale_x_continuous(
    expand = c(0,0,0.4,0),
    breaks = NULL
  ) +

  # Facet wrapping by the different races that we are looking into
  facet_wrap(
    vars(race),
    nrow = 2
  ) +

  # Making sure that the labels are in percentages
  geom_text(
    mapping = aes(
      label = scales::percent(round(percent,
                                    digits = 3))),
    hjust = -0.3, color = "black") +

  labs(
    x = NULL,
    y = NULL,
    fill = "Stop Type",
    title = "Race on Traffic Stop Type Percentage"
  ) +

  theme_bw() +

  theme(
    strip.text.x = element_text(size = 12, face = "bold"),
    plot.title = element_text(hjust = .4,
                              face = "bold")
  ) +

  # Matching our color palette from earlier
  scale_fill_manual(
    values = c("#e86161", "#e35878", "#ef5675", "#cf5598", "#b15caf", "#8864be", "#4e6cc5")
  )
```

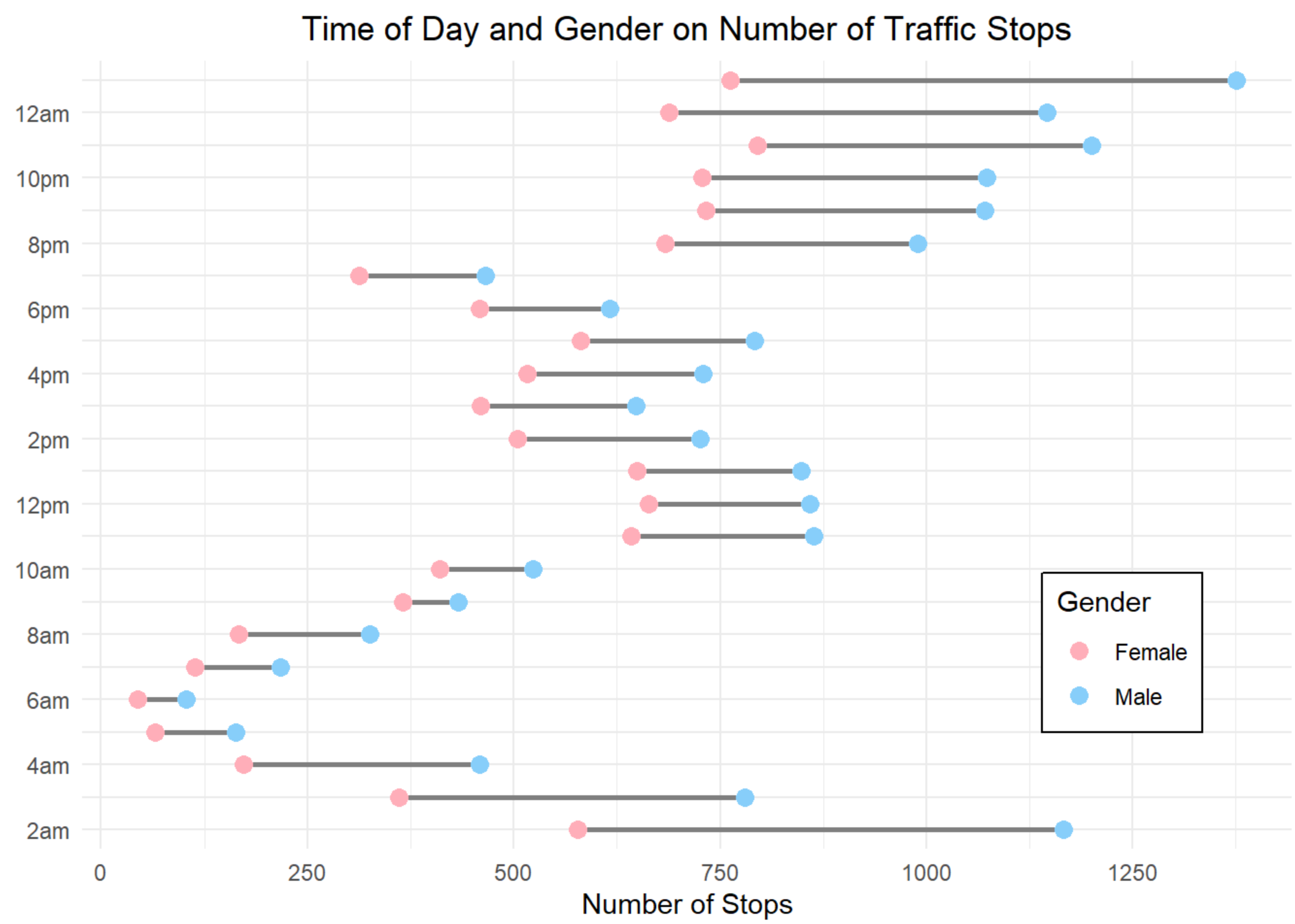


The data is grouped by race and shows the percentage of stop type. For all races most stops are speeding with the second being something wrong with the vehicle. Our data shows that there is no significant difference between race. Vermont police are not typically discriminatory against race for all stop types.

Gender and Number of Stops

```
traffic_graph %>%
  # Pivoting wider and then longer so that the male and female are in the same column for mappin
g purposes
  filter(gender == "Male" | gender == "Female") %>%
  group_by(gender, hour) %>%
  summarise(num_stops = n()) %>%
  pivot_wider(
    names_from = gender,
    values_from = num_stops
  ) %>%
  pivot_longer(
    cols = c(Female, Male),
    names_to = "gender",
    values_to = "population"
  ) %>%


  ggplot(
    mapping = aes(
      x = population,
      y = factor(hour)
    )
  ) +

  geom_line(
    linewidth = 1,
    color = "grey50"
  ) +

  theme_minimal() +


  geom_point(
    mapping = aes(color = gender),
    size = 3,
    shape = 16
  ) +

  scale_color_manual(
    labels = c("Female", "Male"),
    values = c("lightpink1", "lightskyblue")
  ) +

  theme(
    legend.position = c(.86, .25),
    legend.box = "outside",    # Position the legend box outside the plot
    legend.box.background = element_rect(color = "black"),   # Outline color
    legend.background = element_rect(fill = "white"),
    plot.title = element_text(hjust = .5)
```

```
  ) +

  labs(
    y = NULL,
    color = "Gender",
    title = "Time of Day and Gender on Number of Traffic Stops",
    x = "Number of Stops"
  ) +

  scale_x_continuous(
    breaks = seq(0, 1500, 250)
  ) +

  # Setting the time labels while still keeping the minor breaks for easier visualization
  scale_y_discrete(
    breaks = seq(0, 23, 1),
    labels = c("2am", "", "4am", "", "6am", "", "8am", "", "10am", "", "12pm", "", "2pm", "", "4
pm", "", "6pm", "", "8pm", "","10pm", "", "12am", "")
  )
```

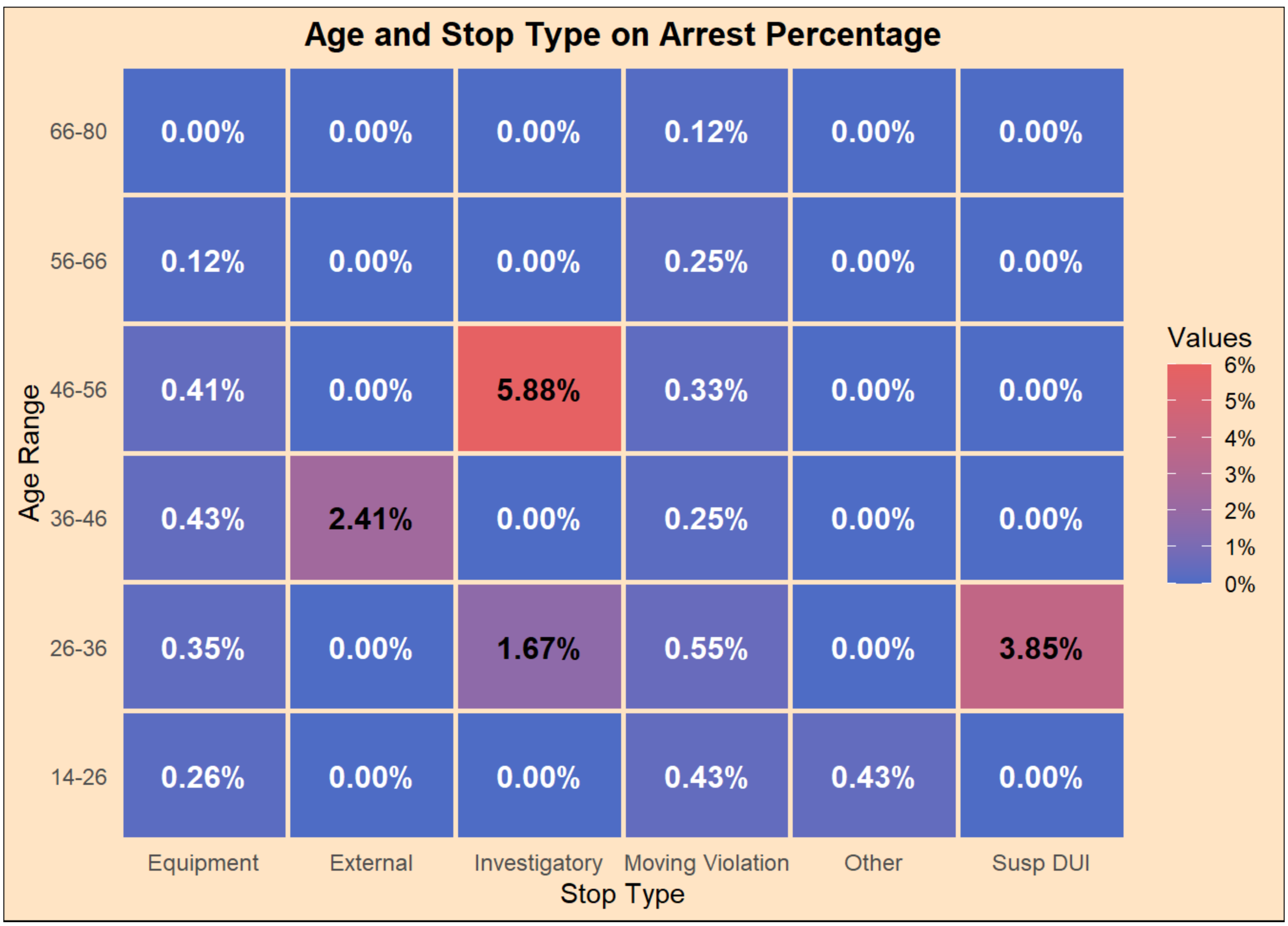


In this graph, we are looking at the time of day and gender on the total number of traffic stops. We can take away that overall, more males are pulled over and that it also aligns generally with our data earlier that most of the traffic stops are from 8pm until 2am. From this graph, we took away that males are pulled over a lot more than females for all hours of the day. This shows that either, male drivers are less responsible than females, or that males in Burlington generally drive more than females and therefore are more likely to have a traffic stop.

Age on stop type

```
traffic_graph %>%

  # Creating the age range
  mutate(age_range = case_when(age >= 14 & age < 26 ~ "14-26",
                               age >= 26 & age < 36 ~ "26-36",
                               age >= 36 & age < 46 ~ "36-46",
                               age >= 46 & age < 56 ~ "46-56",
                               age >= 56 & age < 66 ~ "56-66",
                               age >= 66 & age < 80 ~ "66-80",
                               .default = NA_character_) %>% factor()) %>%
  # For zero percentage
  mutate(stop_outcome = factor(stop_outcome),
         stop_type = factor(stop_type)) %>%
  group_by(age_range, stop_type, stop_outcome, .drop = F) %>%
  summarise(n = n()) %>%
  mutate(percentage = n/sum(n)) %>%
  filter(stop_outcome == "Arrest", !is.na(age_range)) %>%


  ggplot(
    mapping = aes(
      x = stop_type,
      y = age_range,
      fill = percentage
    )
  ) +

  geom_tile(
    linewidth = 1,
    color = "bisque"
  ) +

  theme(axis.text.x = element_text(size = 6)) +

  # Adding the percentages in the square
  geom_fit_text(
    mapping = aes(label = scales::percent(round(percentage, digits = 4))),
    color = "black",
    fontface = "bold",
    reflow = T,
    contrast = T
  ) +

  theme_minimal() +

  labs(
    fill = "Arrest \nPercentage",
    y = "Age Range",
    x = "Stop Type",
```

```
    title = "Age and Stop Type on Arrest Percentage"
  ) +

  # Changing the theme
  theme(
    axis.text.x = element_text(size = 9),
    plot.title = element_text(hjust = .5,
                              face = "bold"),
    plot.background = element_rect(fill = "bisque")
  ) +

  # Creating the gradient scale
  scale_fill_gradient(
    low = "#4e6cc5", high = "#e86161", name = "Values",
    limits = c(0,0.06),
    breaks = seq(0.06,0.00,-0.01),
    labels = c("6%", "5%", "4%", "3%", "2%", "1%", "0%")
  ) +

  coord_cartesian(expand = F)
```



We graphed arrests percent by age and observed the stop type for each age range. Our data shows that those arrested due to investigatory reasons were mostly aged 47 to 56. Additionally those pulled over for a suspected DUI were mostly ages 26 to 36. Though, this data is not very convincing when trying to predict whether someone will be arrested based on age or race. Because the arrests are few and far between comparatively, each individual case needs context for further analysis.

Machine Learning - Classification tree

```
tree_data <-

  # creating an age range instead of using individual ages
  traffic_graph %>%
  filter(race != "White", !is.na(gender)) %>%
  mutate(age_range = case_when(age >= 14 & age < 26 ~ "14-26",
                               age >= 26 & age < 36 ~ "26-36",
                               age >= 36 & age < 46 ~ "36-46",
                               age >= 46 & age < 56 ~ "47-56",
                               age >= 56 & age < 66 ~ "56-66",
                               age >= 66 & age < 76 ~ "66-76",
                               age >= 76 & age < 86 ~ "76-86",
                               .default = NA_character_)) %>%
  select(stop_outcome, stop_type, age_range, race, gender, month)

traffic_full_tree <-
  rpart(
    formula = gender ~ stop_type + race + age_range,
    data = tree_data,
    method = "class",
    parms = list(split = "information"),
    minsplit = 0,
    minbucket = 0,
    cp = -1
  )

traffic_full_tree$cptable
```

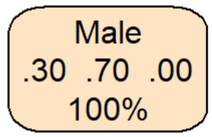
```
##            CP nsplit rel error xerror    xstd
## 1  0.0018899      0    1.0000 1.0000 0.02328
## 2  0.0015564      9    0.9829 1.0000 0.02328
## 3  0.0007782     12    0.9782 0.9946 0.02325
## 4  0.0005188     22    0.9704 1.0086 0.02334
## 5  0.0003891     25    0.9689 1.0101 0.02335
## 6  0.0002594     31    0.9665 1.0125 0.02336
## 7  0.0001556     40    0.9642 1.0171 0.02339
## 8  0.0000000     45    0.9634 1.0179 0.02340
## 9 -1.0000000    104    0.9634 1.0179 0.02340
```

```
traffic_full_tree$cptable %>%
  data.frame() %>%

  # finding the row with the smallest xerror
  slice_min(xerror,
            n = 1,
            with_ties = F) %>%

  #create the xerror_cutoff = xerror + xstd
  mutate(xerror_cutoff = xerror + xstd) %>%

  # picking the xerrorcutoff value
  pull(xerror_cutoff) ->
  xcutoff

traffic_full_tree$cptable %>%
  data.frame() %>%

  # keeping all rows with an xerror below our xcutoff
  filter(xerror < xcutoff) %>%

  # picking the simplest tree based off the location
  slice(1) %>%

  # picking the cp value out of the data frame
  pull(CP) ->
  cp_prune

# printing the important values
c("xerror cutoff" = xcutoff,
  "cp prune value" = cp_prune)
```

```
##  xerror cutoff cp prune value
##        1.01780        0.00189
```

```
prune(
  tree = traffic_full_tree,
  cp = cp_prune
) -> traffic_prune
```

```
rpart.plot(
  x = traffic_prune,
  box.palette = "bisque1"
)
```




```
varImp(traffic_full_tree)
```

```
##            Overall
## age_range   54.35
## race        62.30
## stop_type   62.60
```

```
rpart.rules(
  x = traffic_prune,
  extra = 4
)
```

```
##  gender  Fem Mal Oth
##    Male [.30 .70 .00] null model
```

Summary for Classification Tree: In our attempt to use Machine learning to make predictions about a certain gender based on stop type, age and race, we were unable to find meaningful results that utilized the classification tree. Through our exploration of Machine Learning tools on our data set, we found that there are serious limitations in meaningfully interacting with our data when utilizing machine learning tools. Trying to make predictions about someone's gender based on factors such as age, race, and stop_type does not reveal anything that is meaningful. It would be better just to guess.

Machine Learning - KNN Classification

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

traffic_graph |>
  filter(stop_outcome == "Arrest" | stop_outcome == "Ticket" | stop_outcome == "Warning") |>
  mutate(outcome = as.factor(if_else(stop_outcome == "Arrest" | stop_outcome == "Ticket", 1,
0))) |>
  select(-stop_outcome) -> penalized_data



tibble(penalized_data)
```

```
## # A tibble: 28,992 × 11
##     year month weekday  hour minute race  gender   age stop_type    time outcome
##    <dbl> <dbl> <ord>   <int>  <int> <chr> <chr>  <int> <chr>       <dbl> <fct>
##  1  2012     1 Wed        14     43 White Female    23 Moving Vio…   883 0
##  2  2012     1 Sun         0     18 White Female    73 Equipment      18 0
##  3  2012     1 Sun         0     18 White Male      25 Susp DUI       18 0
##  4  2012     1 Sun         8     50 White Female    23 Equipment     530 0
##  5  2012     1 Sun         8     56 White Female    17 Other         536 0
##  6  2012     1 Sun         9      4 White Male      47 Equipment     544 0
##  7  2012     1 Wed        14     50 White Female    29 Equipment     890 1
##  8  2012     1 Wed        15     13 White Male      28 Moving Vio…   913 0
##  9  2012     1 Sun         9     19 White Female    34 Equipment     559 0
## 10  2012     1 Sun        10     27 White Male      23 Moving Vio…   627 0
## # ℹ 28,982 more rows
```

```
normalize <- function(x){return((x - min(x)) / (max(x) - min(x)))}



traffic_norm <-
  penalized_data %>%
  mutate(across(.cols = c(time, age),
                .fns = normalize))

tibble(traffic_norm)
```

```
## # A tibble: 28,992 × 11
##     year month weekday  hour minute race  gender   age stop_type    time outcome
##    <dbl> <dbl> <ord>   <int>  <int> <chr> <chr>  <dbl> <chr>       <dbl> <fct>
##  1  2012     1 Wed        14     43 White Female 0.204 Moving Vi… 0.614  0
##  2  2012     1 Sun         0     18 White Female 0.337 Equipment  0.0125 0
##  3  2012     1 Sun         0     18 White Male   0.210 Susp DUI   0.0125 0
##  4  2012     1 Sun         8     50 White Female 0.204 Equipment  0.368  0
##  5  2012     1 Sun         8     56 White Female 0.188 Other      0.372  0
##  6  2012     1 Sun         9      4 White Male   0.268 Equipment  0.378  0
##  7  2012     1 Wed        14     50 White Female 0.220 Equipment  0.618  1
##  8  2012     1 Wed        15     13 White Male   0.218 Moving Vi… 0.634  0
##  9  2012     1 Sun         9     19 White Female 0.233 Equipment  0.388  0
## 10  2012     1 Sun        10     27 White Male   0.204 Moving Vi… 0.436  0
## # ℹ 28,982 more rows
```

```
standardize <-
  function(x){return((x- mean(x)) / sd(x))}


# Standardize the traffic data:
traffic_stan <-
  penalized_data |>
  mutate(across(.cols = c(time,age),
              .fns = standardize))

tibble(traffic_stan)
```

```
## # A tibble: 28,992 × 11
##     year month weekday  hour minute race  gender    age stop_type           time
##    <dbl> <dbl> <ord>   <int>  <int> <chr> <chr>   <dbl> <chr>              <dbl>
##  1  2012     1 Wed        14     43 White Female -0.878 Moving Violation  0.0826
##  2  2012     1 Sun         0     18 White Female  2.43  Equipment        -1.98
##  3  2012     1 Sun         0     18 White Male   -0.746 Susp DUI         -1.98
##  4  2012     1 Sun         8     50 White Female -0.878 Equipment        -0.759
##  5  2012     1 Sun         8     56 White Female -1.28  Other            -0.745
##  6  2012     1 Sun         9      4 White Male    0.711 Equipment        -0.726
##  7  2012     1 Wed        14     50 White Female -0.481 Equipment         0.0993
##  8  2012     1 Wed        15     13 White Male   -0.547 Moving Violation  0.154
##  9  2012     1 Sun         9     19 White Female -0.150 Equipment        -0.690
## 10  2012     1 Sun        10     27 White Male   -0.878 Moving Violation -0.528
## # ℹ 28,982 more rows
## # ℹ 1 more variable: outcome <fct>
```

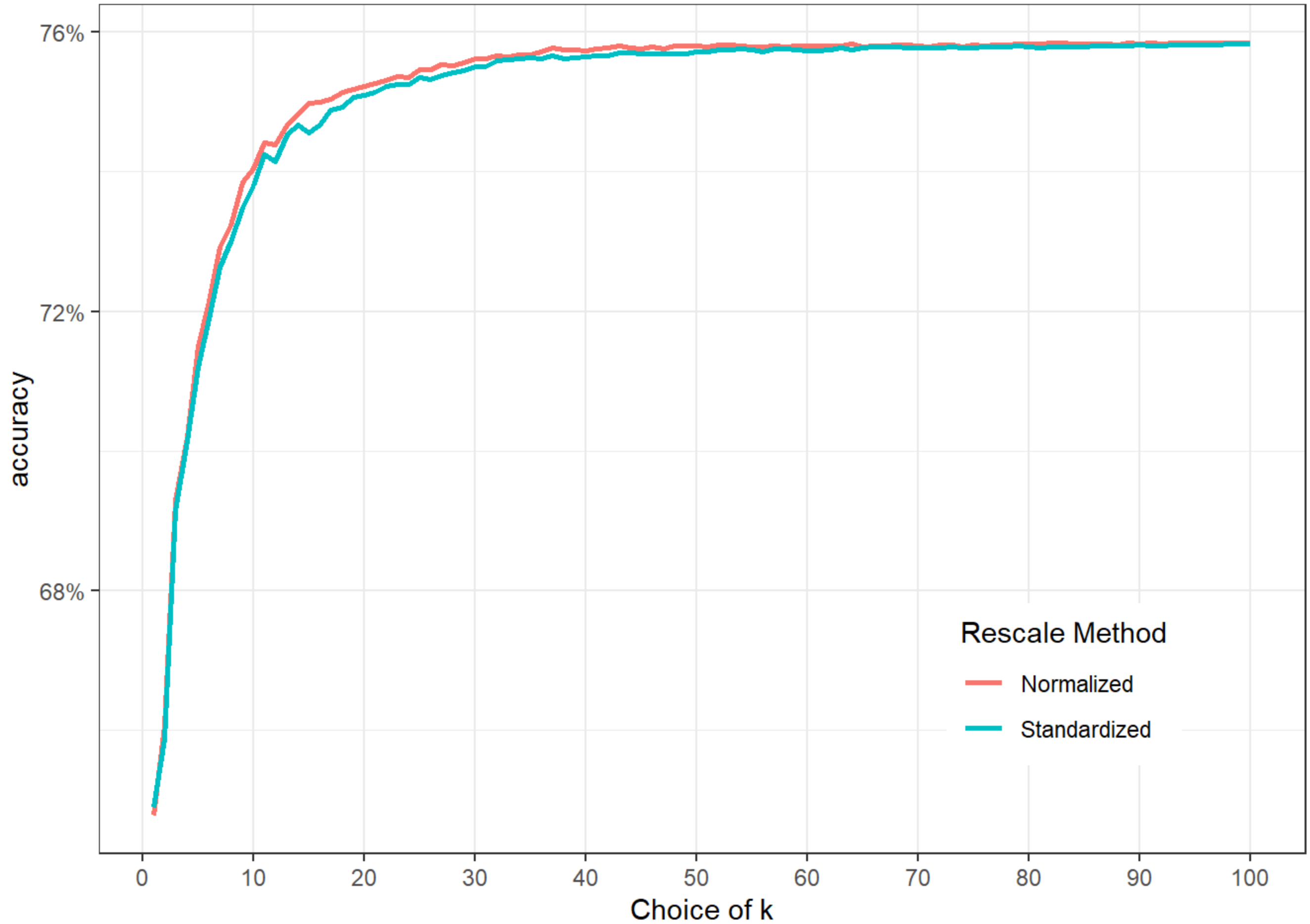
```
size <- 100
# Creating the tibble
knn_results <- data.frame(k = 1:100, norm_acc = rep(-1,size), stan_acc = rep(-1,size))



# Tibble results
tibble(knn_results)
```

```
## # A tibble: 100 × 3
##        k norm_acc stan_acc
##    <int>    <dbl>    <dbl>
##  1     1       -1       -1
##  2     2       -1       -1
##  3     3       -1       -1
##  4     4       -1       -1
##  5     5       -1       -1
##  6     6       -1       -1
##  7     7       -1       -1
##  8     8       -1       -1
##  9     9       -1       -1
## 10    10       -1       -1
## # ℹ 90 more rows
```

```
# Creating the Loop
for(i in 1:size){

  # Gets the knn value for each k based on normalized data
  norm_loop <-
    knn.cv(train = traffic_norm[ , c("time", "age")],
           cl = traffic_norm$outcome,
           k = knn_results$k[i])

  # Creates a confusion Matrix to get overall
  loop_traffic_norm <-
    confusionMatrix(data = norm_loop,
                    reference = traffic_norm$outcome)

  # Puts overall in at k row
  knn_results[i,2] <- loop_traffic_norm$overall[1]

  # Gets knn value for each k based on standardized data
  stan_loop <-
    knn.cv(train = traffic_stan[ , c("time", "age")],
           cl = traffic_stan$outcome,
           k = knn_results$k[i])

  # Creates confusion matrix from stan data to get overall
  loop_traffic_stan <-
    confusionMatrix(data = stan_loop,
                    reference = traffic_stan$outcome)

  # Puts in at k row
  knn_results[i, 3] <- loop_traffic_stan$overall[1]
}


# Displaying the first 10 rows
tibble(knn_results)
```

```
## # A tibble: 100 × 3
##        k norm_acc stan_acc
##    <int>    <dbl>    <dbl>
##  1     1    0.648    0.649
##  2     2    0.661    0.659
##  3     3    0.693    0.692
##  4     4    0.702    0.702
##  5     5    0.715    0.712
##  6     6    0.721    0.719
##  7     7    0.729    0.726
##  8     8    0.732    0.730
##  9     9    0.739    0.735
## 10    10    0.741    0.738
## # ℹ 90 more rows
```

```
knn_results |>

  # Pivot rows to graph
  pivot_longer(cols = -k,
               names_to = "rescale_method",
               values_to = "accuracy") |>

  # Plot knn_results
  ggplot(mapping = aes(x = k, y = accuracy, color = rescale_method)) +

  # Change labels
  labs(x = "Choice of k",
       color = "Rescale Method") +

  # Change y labels to be percentage
  scale_y_continuous(labels = scales::label_percent()) +

  # Change x axis break
  scale_x_continuous(breaks = seq(0,100,10), minor_breaks = NULL) +

  # Add lines
  geom_line(linewidth = 1) +

  # Add theme
  theme_bw() +

  # Update legend postion to be in graph
  theme(legend.position = c(0.8, 0.2)) +

  scale_color_manual(labels = c("Normalized", "Standardized"), values = c("#F8766D","#00BFC4"))
```




```
best_knn <-
  knn.cv(train = traffic_norm[ , c("time", "age")],
         cl = traffic_norm$outcome,
         k = knn_results$k[11])


acc_matrix <-
  confusionMatrix(data = best_knn,
                  reference = traffic_norm$outcome)

acc_matrix
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0 21145  6576
##          1   842   429
##                                           
##                Accuracy : 0.744           
##                  95% CI : (0.739, 0.749)  
##     No Information Rate : 0.758           
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.032           
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.9617          
##             Specificity : 0.0612          
##          Pos Pred Value : 0.7628          
##          Neg Pred Value : 0.3375          
##              Prevalence : 0.7584          
##          Detection Rate : 0.7293          
##    Detection Prevalence : 0.9562          
##       Balanced Accuracy : 0.5115          
##                                           
##        'Positive' Class : 0               
## 
```

```
summary(acc_matrix)
```

```
##          Length Class  Mode     
## positive  1     -none- character
## table     4     table  numeric  
## overall   7     -none- numeric  
## byClass  11     -none- numeric  
## mode      1     -none- character
## dots      0     -none- list     
```

We looked at how age and time of the day (in minutes) affected stop outcome. From our data we found that there was no significance between age and time on stop outcome. We first normalized and standardized the data and then ran KNN. Our confusion matrix shows the normalized data with a no information rate of 0.758 and accuracy 0.744. Since the no information rate is larger than the accuracy this further highlights the fact that our data isn't suited for this machine learning method.

Conclusion

From our data we found that there are a few factors that affect whether someone will be stopped or not by a police. The most common time was from 10pm to 2am. This is probably due to people going home and wanting to get home quick, or drinking and driving. There was no discrimination in race as every race had the same percentage for each stop type, with speeding being the most common. Majority of Vermont's population is white, with 92.93% of Vermont citizens being white. Additionally, we found that gender was another factor with males being pulled over significantly more throughout the whole day. We also looked at age and time. Our data shows that most people stopped are for speeding or suspected DUI. The younger drivers were more likely to be stopped, most likely due to more reckless driving or driving late at night, drinking and driving. Overall, the typical demographic for being pulled over were white males, driving late at night, ages 20 to 30. We were curious as to how arrests factored into this demographic and grouped ages in age ranges of 10 years. Those that were pulled over for investigatory reasons were typically ages 46 to 57 and 6% were arrested, likely due to having a suspicion of a crime. Those who were pulled over for speeding were likely to not get arrested, this is because speeding/moving violation is the most common stop type. All things considered, our data did show gender, age, and time affected stop type and that the factors we originally hypothesized to affect stop type, had little to no impact.

Limitations and Recommendations

There were many limitations with our data as most of it was categorical and a lot of the machine learning we learned was for data sets with more quantitative data. Because of Vermont's population our data leans towards the majority and doesn't show any significance in race. This is why our regression tree only showed one box, because the other nodes were insignificant. Our data is better suited for retrospective data, as our graphs showed trends, but we weren't able to predict trends. For future use we would recommend other machine learning types to show significance and accuracy. Additionally, one could look at counties in Vermont if given them and map it based on different factors. The data is interesting, but very limiting and not very diverse.