Data Science Project - Analyzing U.S. Traffic Accident Data

Jatin Suri

1. Introduction

This paper examines a U.S. traffic accident data set (2017 -2020) to explore how the number of accidents differs by state, days of the week, and weather conditions. It also examines whether Covid-19 affected accident numbers and employs a best model to predict accident severity. This paper finds that during the period 2017-2020, California had the most number of accidents, while New York had the greatest number of accidents per capita. Most accidents occurred on Friday, while the least number of accidents occurred over the weekend (specifically on Sunday). The amount of precipitation has a slight positive correlation with the number of accidents and is statistically significant. Additionally, a random forest and linear model both predicted accident severity with an accuracy of above 70%; however, the linear model had a higher accuracy rate (80%). Finally, the number of accidents reduced during the 2020 summer, most likely due to the effects of Covid-19.

2. Data Source

The accident data set covers 48 mainland states of the U.S. along with D.C. and spans from 2016 to 2020. The data has been collected from several data providers including several APIs that stream traffic event data. The API's stream data from the department of transportation, traffic cameras, law enforcement agencies, and traffic sensors. The data set holds over 2 million accident records.

URL: https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents (https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents)

Source: Kaggle

The population data set includes population for 50 U.S. states as well as D.C. for 2019.

URL: https://www.icip.iastate.edu/tables/population/states-estimates (https://www.icip.iastate.edu/tables/population/states-estimates)

Source: U.S. Census Bureau

3. Ethics Reflection

There are both ethical harms and benefits associated with this data. Some ethical benefits include the enhancement of map services and traffic rerouting as a result of predictive algorithms that predict accident delays. As a result, this can save people time when commuting. Another benefit includes useful analytics for automated vehicles. If there are areas where accidents frequently occur, then those analytics can be factored into automated vehicles to have them drive slower in high-risk areas. On the other hand, some ethical harms consist of some locations getting flagged as accident hot spot locations which can potentially create stereotypes for specific cities or zip codes. Additionally, this data set does not include any information about the driver; however, it may be possible to collect that information from external sources, which would be problematic as a privacy issue and could result in an biased algorithm toward certain demographics.

4. Data Import

```
library(tidyverse)
library(party)
library(ranger)
library(modelr)
library(car)
library(lmtest)
```

```
#Reading in CSV File (Accidents)
accidents <- read_csv("US_Accidents_Dec20_updated.csv")

#Reading in CSV File (State Population)
state_pop <- read_csv("2019 population by state.csv")
```

5. Data Cleaning and Tidying

#Explain the variables

The kaggle accident data set includes two columns: Start_Time and End_Time, which each include both a date and a time. During data cleaning, a date column was created by extracting the date from the Start_Time column. Additionally, an accident duration column was created by subtracting End_Time with Start_Time to get the total duration of the accident. Finally, the weekdays() function was used to convert the Date column into a day of the week. These days of the week were stored in a new column: day_week.

```
#Creating a Date Column, Duration Column, and Weekday Column
accidents <- accidents %>%
  mutate(Date = as.Date(accidents$Start_Time), Duration = as.numeric(End_Time - Start_Time),
         day_week = weekdays(Date))
```

The accident data set was then filtered to only include the years 2017-2020 because according to the Kaggle data set author, Sobhan Moosavi, certain traffic API's were adopted at the end of 2016 resulting in greater accuracy for the 2017-2020 years.

```
#Filtering out year 2016 as the data had some inaccuracies (according to the Kaggle Data set Aut
hor)
accidents <- filter(accidents, !str_detect(Date, "2016"))
```

Variables with similar information and variables that could not be effectively analyzed were dropped from the data set. Furthermore, variables were changed to their correct data type and renamed accordingly. Since there were some NaN values in the columns for numeric variables, the NaN values were changed to NA to allow for computations.

```
#Dropping unnecessary columns, changing data types, and renaming variables
accidents <- accidents %>%
  select(-Astronomical_Twilight, -Nautical_Twilight, -Civil_Twilight, -Wind_Direction, -Weather_
Timestamp, -Airport_Code, -Timezone, -Country, -Zipcode,
         -Street, -Number, -Description, -End_Time, -Start_Time, -Amenity, -Bump, -Crossing, -Gi
ve_Way,
         -Junction, -No_Exit, -Railway, -Roundabout, -Station, -Stop, -Traffic_Calming, -Traffic
_Signal,
         -Turning_Loop) %>%
  mutate(Severity = as.integer(Severity), Side = as.factor(Side), State = as.factor(State),
         Weather_Condition = as.factor(Weather_Condition),
         Sunrise_Sunset = as.factor(Sunrise_Sunset)) %>%
  rename(Distance = 'Distance(mi)', Temperature = 'Temperature(F)', Wind_Chill = `Wind_Chill(F)
`,
         Humidity = `Humidity(%)`, Pressure = `Pressure(in)`,
         Visibility = `Visibility(mi)`, Wind_Speed = `Wind_Speed(mph)`, Precipitation =
           `Precipitation(in)`)

#Replacing NaN values with NA
accidents[sapply(accidents, is.nan)] <- NA
```

The state population data set has a column for state names and a column for population; however, the state name column needed to be converted to abbreviations, so that it could be joined to the accident data set. The state.abb() function was used to match the state names to abbreviations. Since D.C. was not part of the state.abb() abbreviation list, it had to be added manually.

```
#Data Cleaning - State Population Data Set
#Changing State Names to State Abbreviations
state_pop <- state_pop %>% mutate(state_ab = state.abb[match(state_pop$State, state.name)])
#Adding in D.C.
state_pop$state_ab[is.na(state_pop$state_ab)] <- "DC"
#Removing extra State Column
state_pop <- state_pop %>% mutate(State = state_ab) %>% select(-state_ab)
```

Finally, a left join was used to merge the accident and state population data sets together. A left join was used because all of the information from the left data set (accidents) was to be kept, while the population needed to be joined by the state name.

```
#Left Join
accidents_merged <- left_join(accidents, state_pop) %>%
  select(ID, State,"Population 2019", everything()) %>%
  rename(Population = 'Population 2019')
```

```
## Joining, by = "State"
```

```
#Merged Data set
head(accidents_merged, 5)
```

```
## # A tibble: 5 x 24
##   ID    State Population Severity Start_Lat Start_Lng End_Lat End_Lng Distance
##   <chr> <chr>      <dbl>    <int>     <dbl>     <dbl>   <dbl>   <dbl>    <dbl>
## 1 A-1   SC       5157702        2      34.8     -82.3    34.8   -82.3      0
## 2 A-2   NC      10501384        2      35.1     -80.7    35.1   -80.7      0
## 3 A-3   CA      39437610        2      37.1    -122.     37.2  -122.       1.4
## 4 A-4   NV       3090771        2      39.1    -120.     39.1  -120.       0
## 5 A-6   NC      10501384        3      35.3     -80.8    35.3   -80.8      0
## # ... with 15 more variables: Side <fct>, City <chr>, County <chr>,
## #   Temperature <dbl>, Wind_Chill <dbl>, Humidity <dbl>, Pressure <dbl>,
## #   Visibility <dbl>, Wind_Speed <dbl>, Precipitation <dbl>,
## #   Weather_Condition <fct>, Sunrise_Sunset <fct>, Date <date>, Duration <dbl>,
## #   day_week <chr>
```

The variable explanations for the merged data set is as follows:

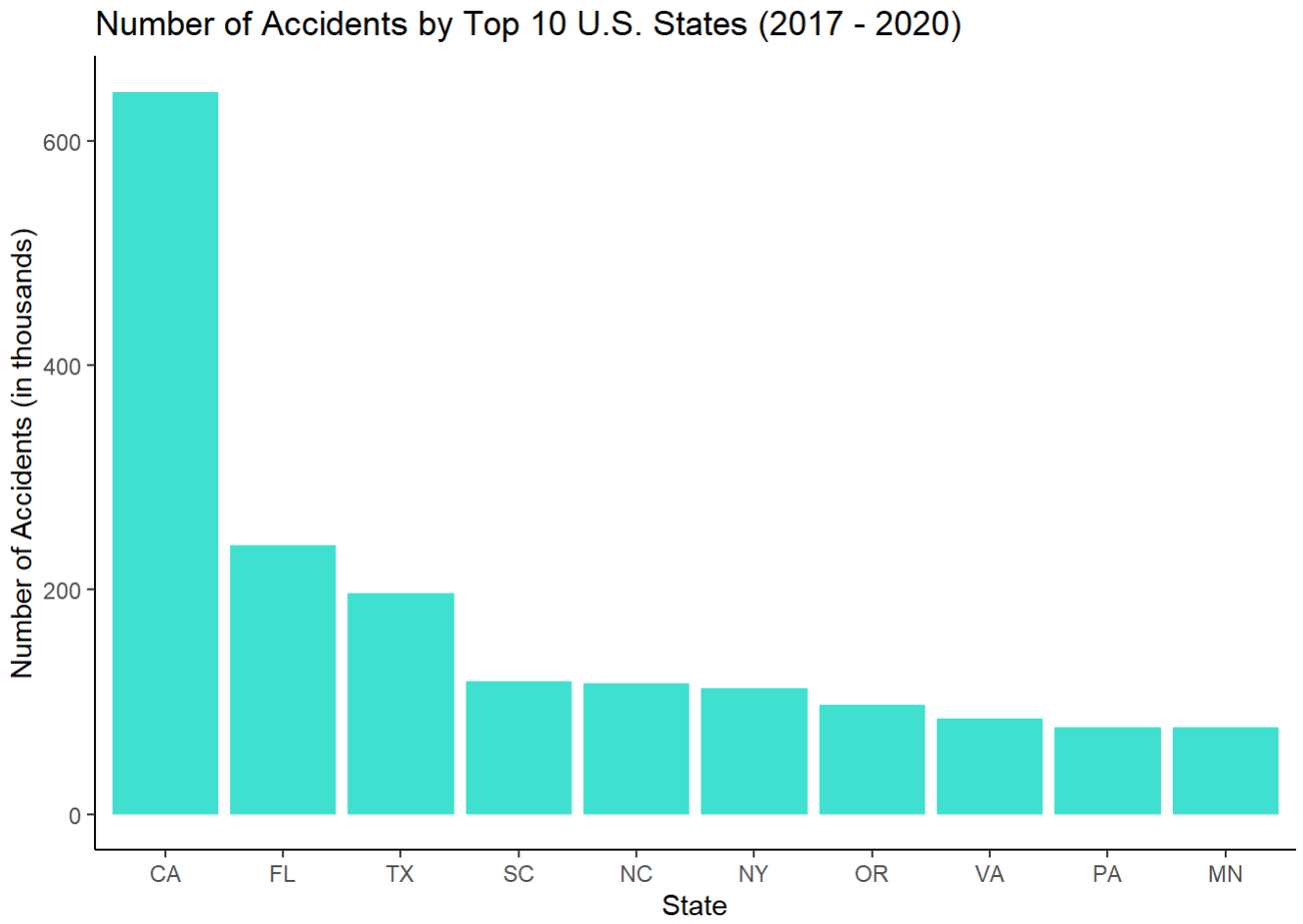
- ID: This is a unique identifier of the accident record
- Date: Date of accident
- week_day: Day of week when accident occurred
- Population: State population number
- Duration: Total time till accident was cleared
- Severity: A number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay)
- Start_Time: Start time of the accident in local time zone
- End_Time: End time of the accident in local time zone. End time here refers to when the impact of accident on traffic flow was dismissed
- Start_Lat: Latitude in GPS coordinate of the start point
- Start_Lng: Longitude in GPS coordinate of the start point
- End_Lat: Latitude in GPS coordinate of the end point
- End_Lng: Longitude in GPS coordinate of the end point
- Distance: The length of the road (mi) extent affected by the accident.
- City: City where accident occurred
- County: County where accident occurred
- State: State where accident occurred
- Temperature: Shows the temperature (in Fahrenheit)
- Wind_Chill: Shows the wind chill (in Fahrenheit)
- Humidity: Shows the humidity (in percentage)
- Pressure: Shows the air pressure (in inches)
- Visibility: Shows visibility (in miles)
- Wind_Speed: Shows wind speed (in miles per hour)
- Precipitation: Shows precipitation amount in inches, if there is any
- Weather_Condition: Shows the weather condition (rain, snow, thunderstorm, fog, etc.)

6. Data Exploration

Visualizing the number of accidents and accidents per capita by state

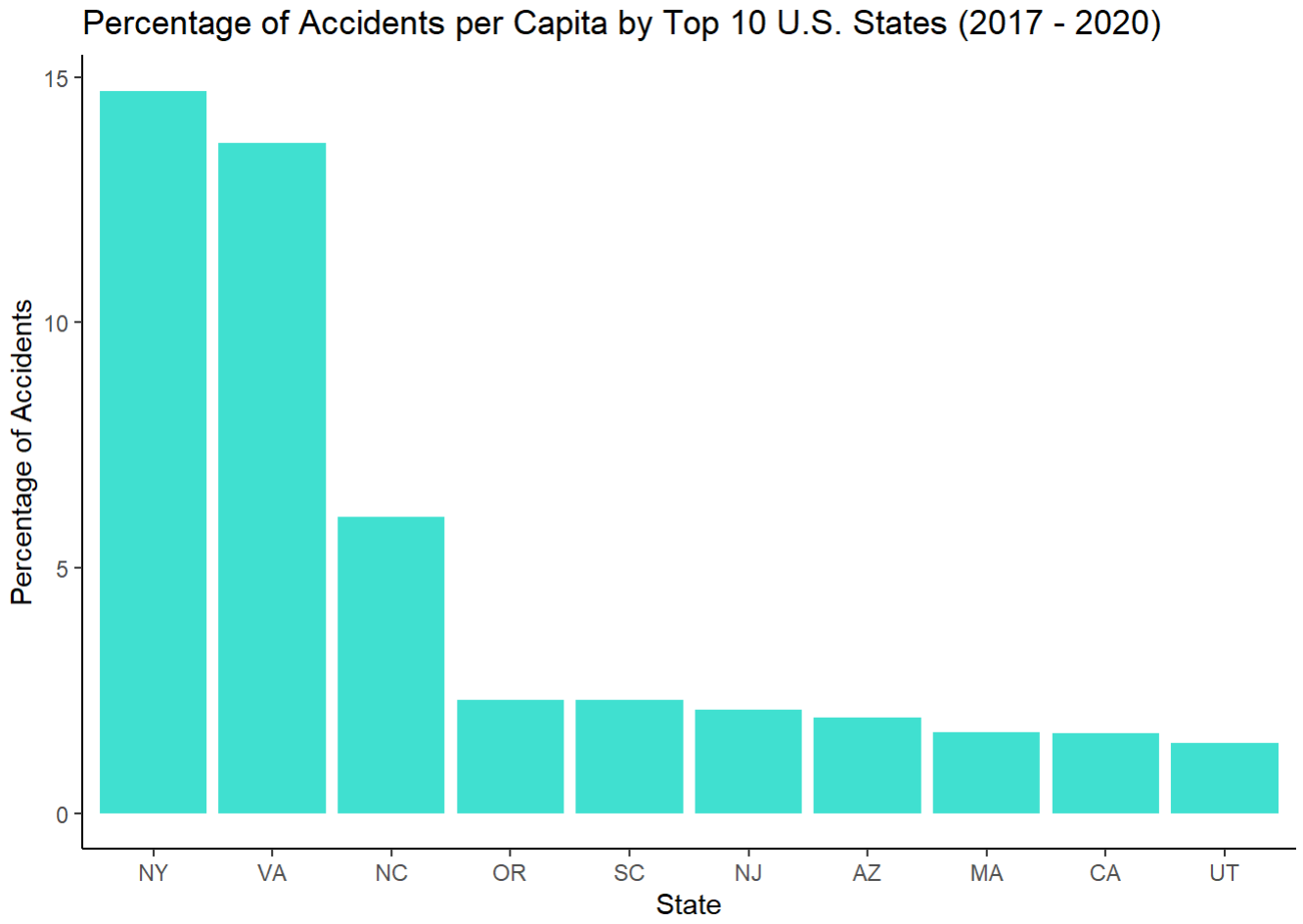
The first step was to create a tibble containing the state name, number of accidents, and accidents per capita (proportion). The accident counts were calculated using a summarize function while grouping by state. The accidents per capita were calculated by dividing the number of accidents per state by the population of that respective state.

```
#Tibble with State, Count, and Proportion (# of accidents in state/state population)
accidents_count <- accidents_merged %>%
  group_by(State) %>%
  summarize(n = n()) %>%
  arrange(State)
accidents_count <- accidents_count %>% mutate(prop = accidents_count$n/state_pop$'Population 201
9'[state_pop$State%in%accidents_count$State])

#Bar Chart - State vs. # of accidents
ggplot(data = accidents_count[order(-accidents_count$n),][1:10,]) + geom_bar(aes(x = reorder(Sta
te, -n), y = n/1000), stat = "identity", fill = "turquoise") + labs(title = "Number of Accidents
by Top 10 U.S. States (2017 - 2020)", x = "State", y = "Number of Accidents (in thousands)") + t
heme_classic()
```



A bar chart illustrating number of accidents by state shows that the state with the most accidents is California followed by Florida and Texas.

```
#Bar Chart - State vs. Prop of accidents
ggplot(data = accidents_count[order(-accidents_count$prop),][1:10,]) + geom_bar(aes(x = reorder
(State, -prop), y = 100*prop), stat = "identity", fill = "turquoise") + labs(title = "Percentage
of Accidents per Capita by Top 10 U.S. States (2017 - 2020)", x = "State", y = "Percentage of Ac
cidents") + theme_classic()
```

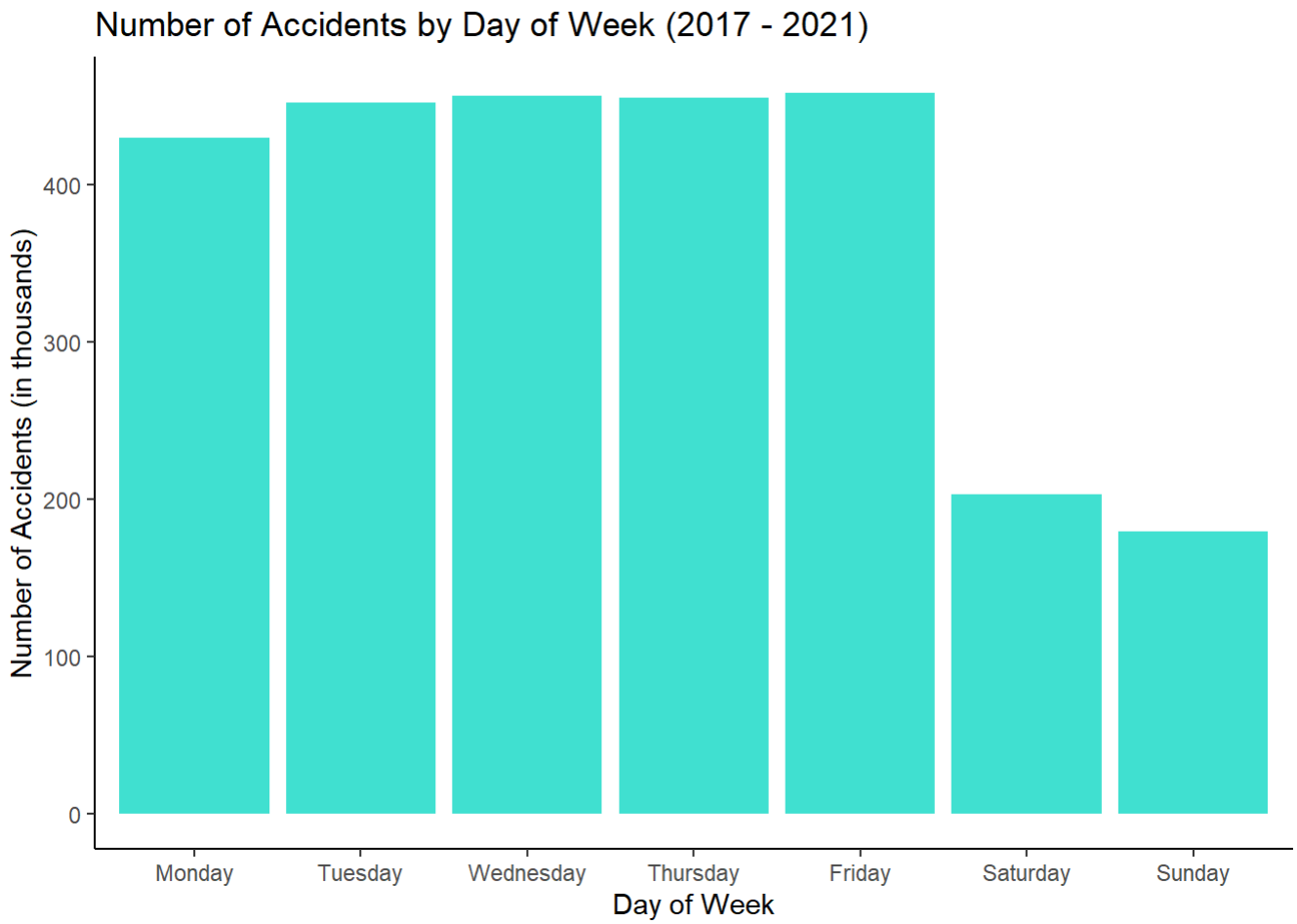


A bar chart illustrating number of accidents per capita by state shows that the state with the most accidents per capita is New York followed by Virginia and North Carolina.

Does day of the week affect the number of accidents?

The first step was to create a tibble with a column for day of week and a column for number of accidents. The number of accidents was counted using a summarize function while grouping my day of week. The day of week was then changed to a factor with the levels Monday through Sunday.

```
#Tibble with day of week (as factors) and # of accidents on that day
accidents_date <- accidents_merged %>%
  group_by(day_week) %>%
  summarize(n = n()) %>%
  arrange(day_week)
accidents_date$day_week = factor(accidents_date$day_week,
                                 levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Frida
y",
                                            "Saturday", "Sunday"))

#Bar Chart - Days of Week vs. # of accidents
ggplot(data = accidents_date) + geom_bar(aes(x = day_week, y = n/1000), stat = "identity", fill
 = "turquoise") + labs(title = "Number of Accidents by Day of Week (2017 - 2021)", x = "Day of W
eek", y = "Number of Accidents (in thousands)") + theme_classic()
```

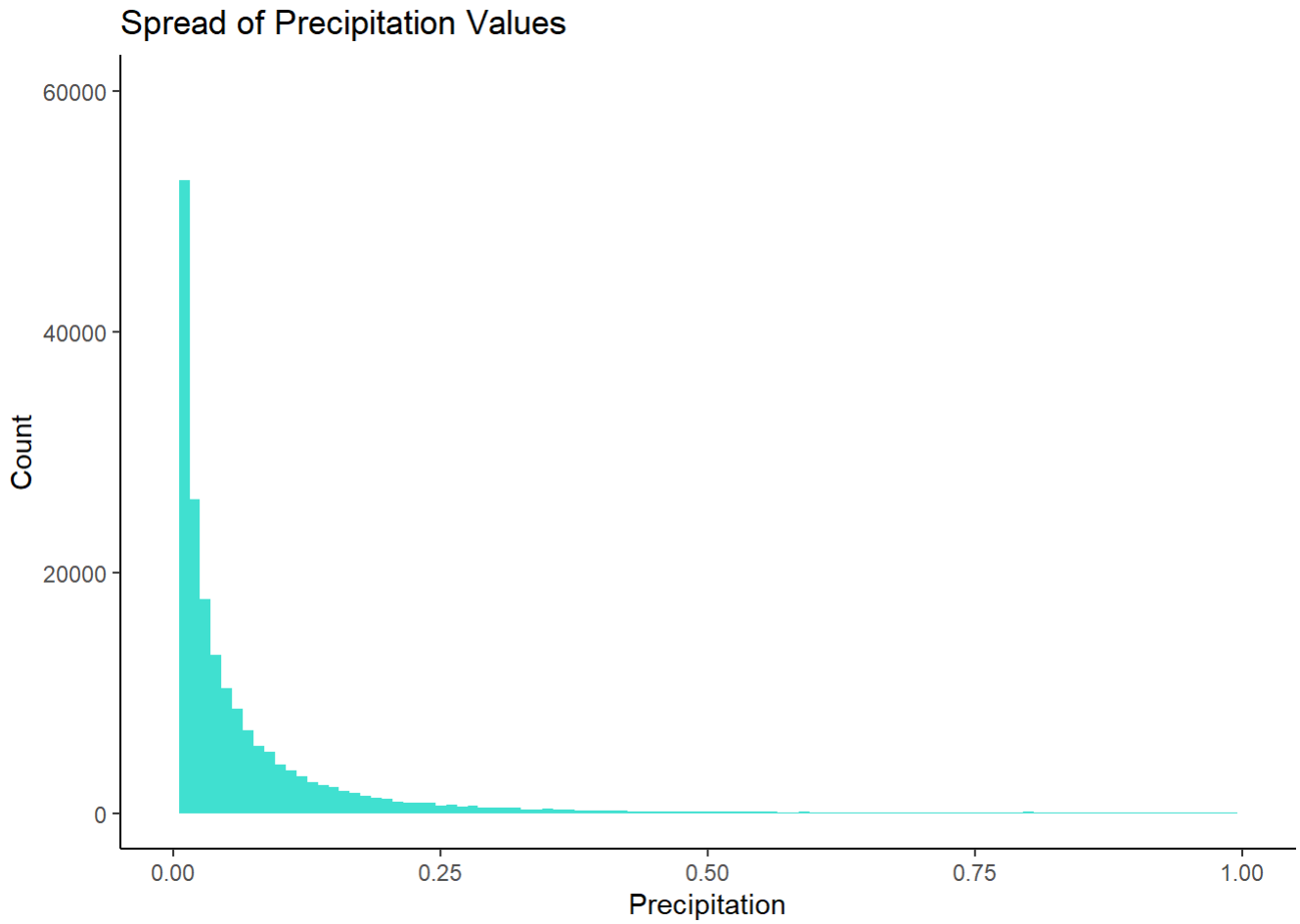


A bar chart illustrating the number of accidents by day of week shows that the most accidents occur on Friday, while substantially less accidents occur over the weekend.

Does the amount of precipitation affect the number of accidents?

First a histogram was created to visualize the spread of precipitation. The histogram shows that the precipitation values are right skewed, with most values being zero.

```
#Histogram to see spread of precipitation data
ggplot(data = accidents_merged) + geom_histogram(aes(x = Precipitation), binwidth = .01, fill =
"turquoise") + xlim(c(0,1)) + ylim(c(0,60000)) + theme_classic() + labs(title = "Spread of Preci
pitation Values", y = "Count")
```



The data was grouped by state to create a tibble with columns for state, number of accidents, and average precipitation by state.

```
#Tibble with State, Number of accidents, and Mean Precipitation
accidents_prec <- accidents_merged[!is.na(accidents_merged$State),] %>%
    group_by(State) %>%
    summarise(n = n(), mean_prec = mean(Precipitation, na.rm = TRUE))
```

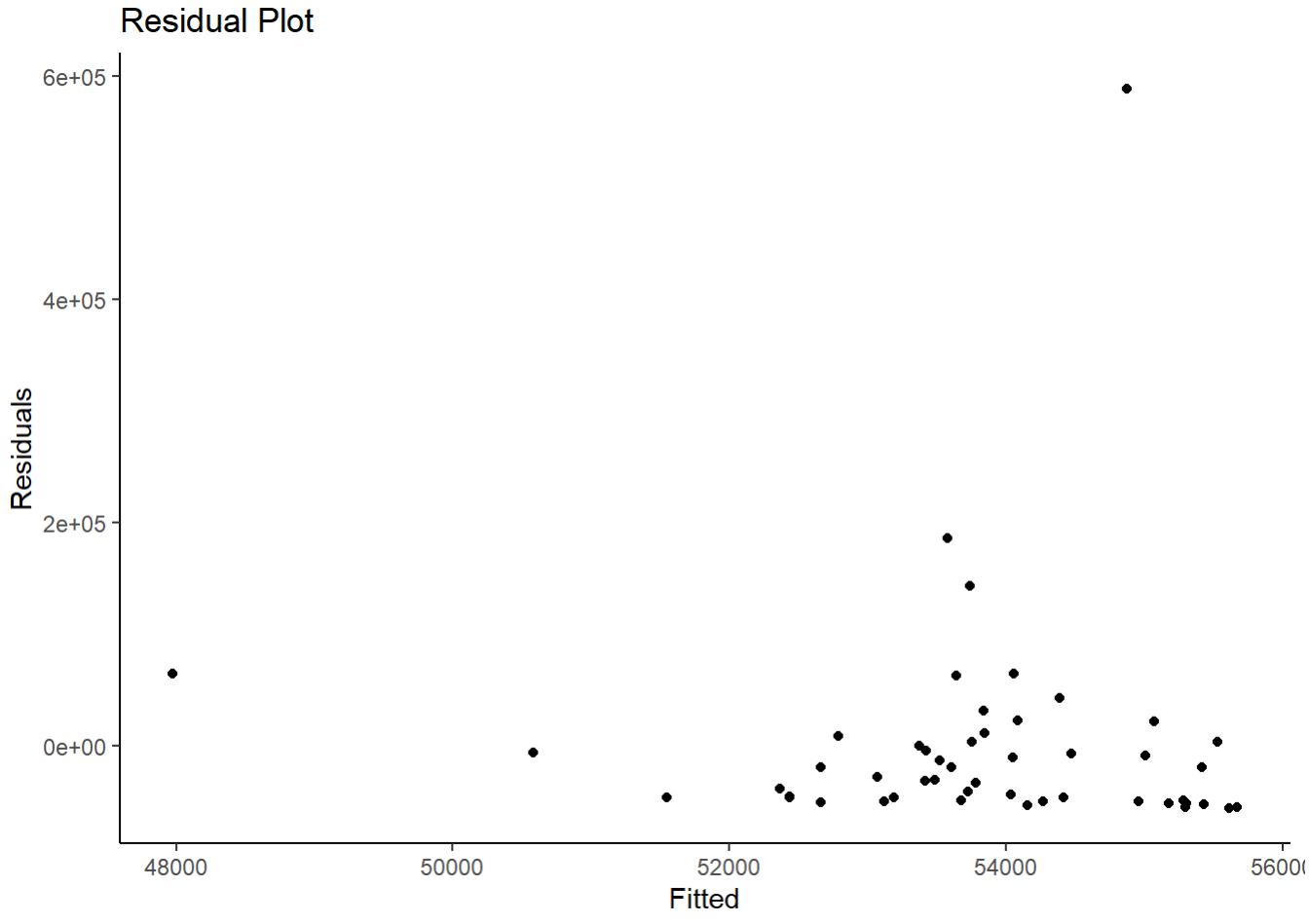
A linear model was fitted to predict the number of accidents by a state's mean precipitation.

```
#Model 1
mod1 <- lm(n ~ mean_prec, data = accidents_prec)
summary(mod1)
```

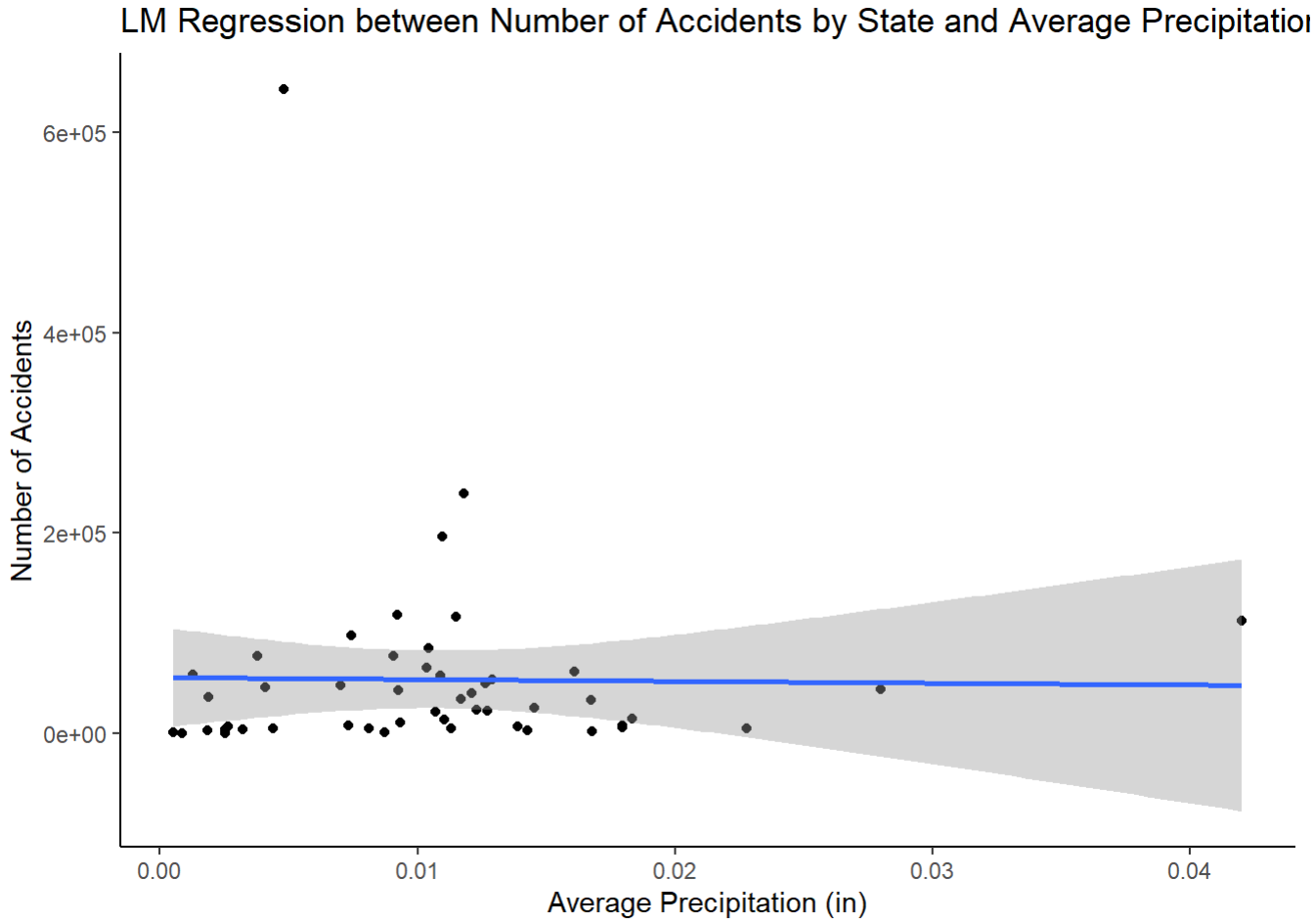
```
##
## Call:
## lm(formula = n ~ mean_prec, data = accidents_prec)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -55401 -48825 -27663   3424 588664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    55769      25200   2.213   0.0318 *
## mean_prec    -185517    1942897  -0.095   0.9243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100400 on 47 degrees of freedom
## Multiple R-squared:  0.0001939,  Adjusted R-squared:  -0.02108
## F-statistic: 0.009117 on 1 and 47 DF,  p-value: 0.9243
```

Model 1 depicts that average precipitation is not statistically significant and results in a R-squared not statistically different from 0.

```
ggplot(data = mod1) + geom_point(aes(x = fitted.values(mod1), y = residuals(mod1))) + labs(title
= "Residual Plot", y = "Residuals", x = "Fitted") + theme_classic()
```



```
ggplot(data = accidents_prec, aes(x = mean_prec, y = n)) + geom_point() +
  stat_smooth(method = "lm", formula = y~x, geom = "smooth") + theme_classic() + labs(title = "L
M Regression between Number of Accidents by State and Average Precipitation", x = "Average Preci
pitation (in)", y = "Number of Accidents")
```

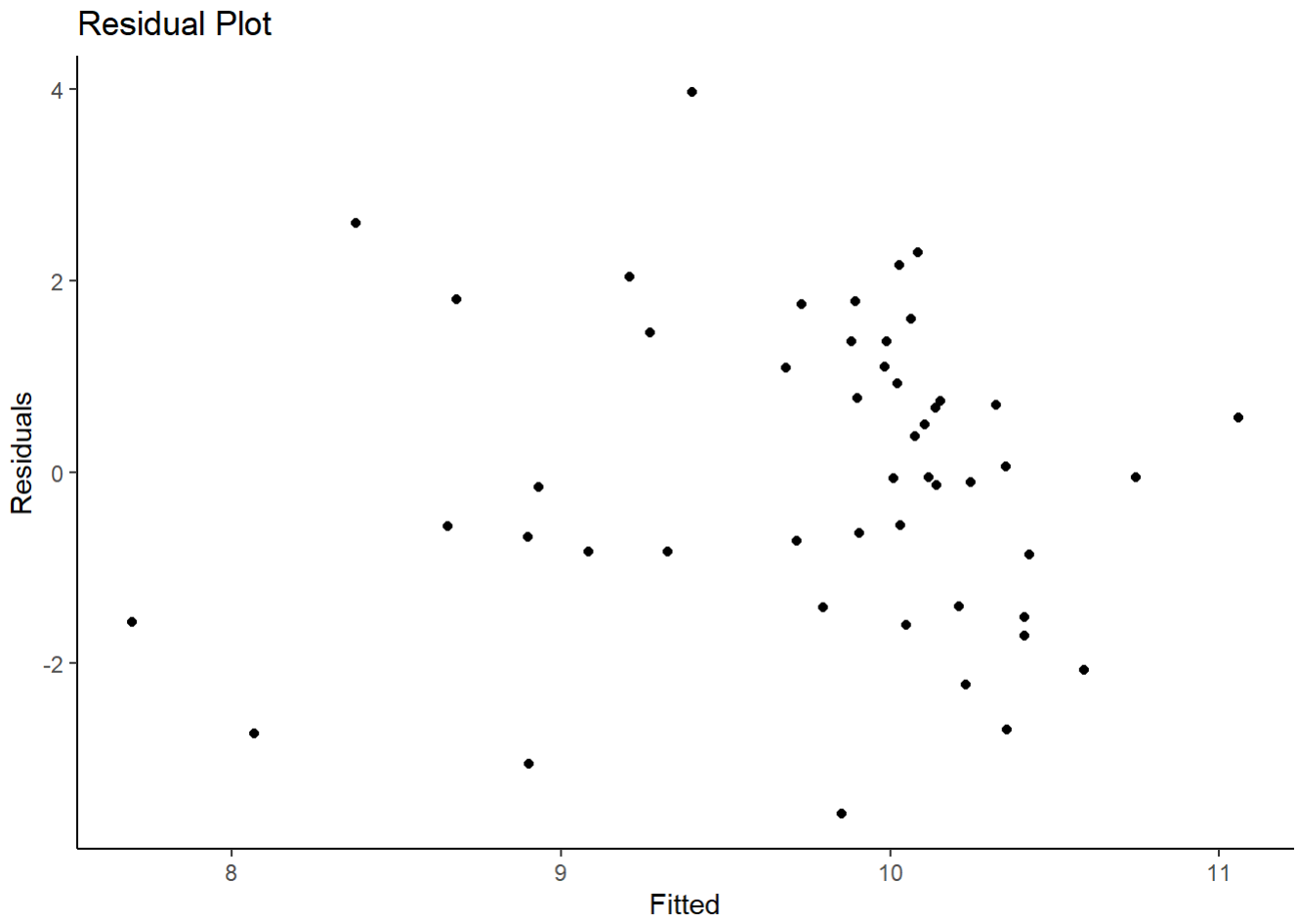


The residual plot for model 1 shows a pattern implying potential heteroskedasticity. A potential transformation may be needed to improve the model.

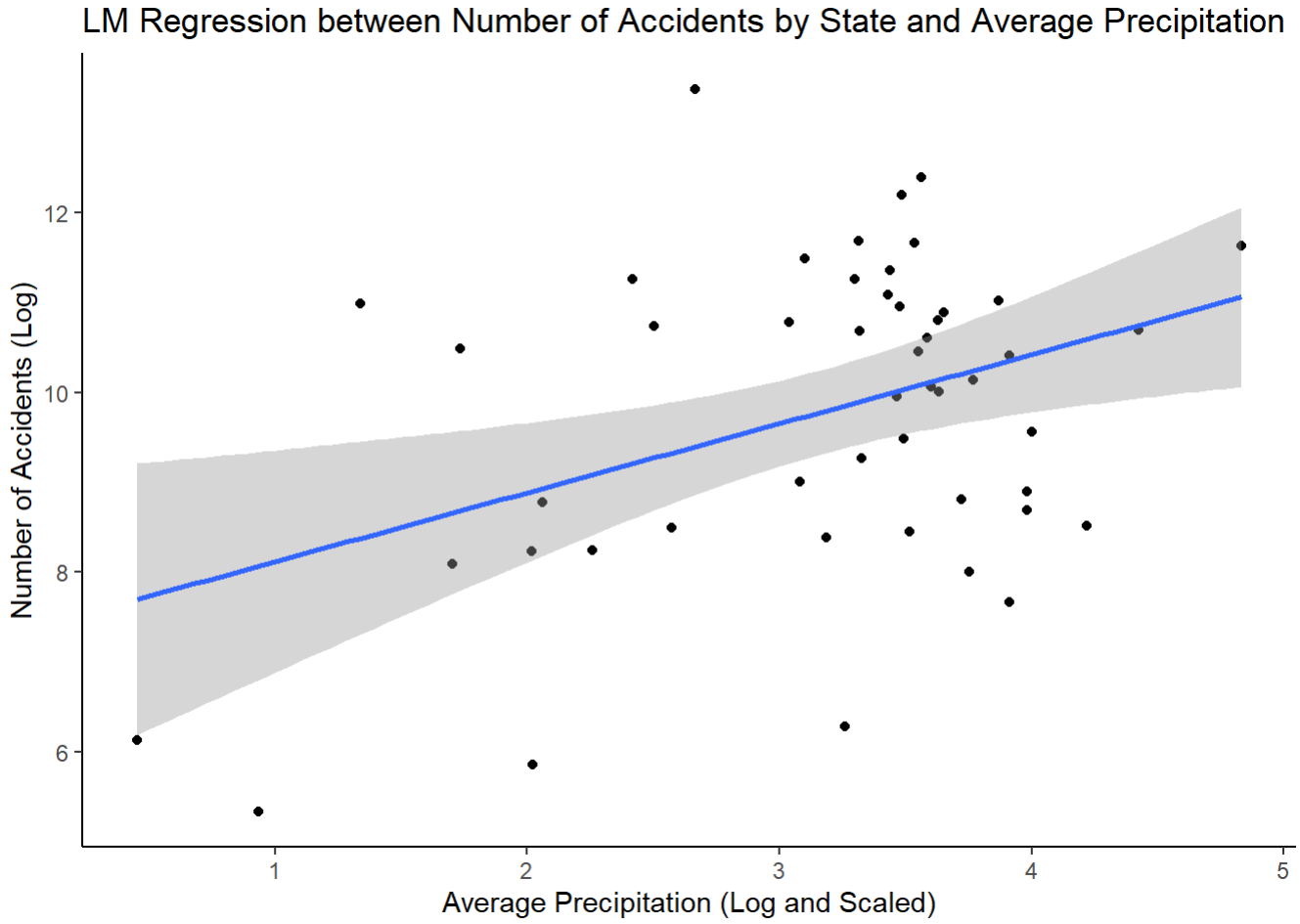
```
#Model 2 - Log Transformation
mod2 <- lm(log(n) ~ log(mean_prec), data = accidents_prec)
summary(mod2)
```

```
##
## Call:
## lm(formula = log(n) ~ log(mean_prec), data = accidents_prec)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5642 -0.8601 -0.0559  1.1002  3.9767
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     13.4919     1.2924   10.44 7.86e-14 ***
## log(mean_prec)   0.7676     0.2629    2.92  0.00536 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.631 on 47 degrees of freedom
## Multiple R-squared:  0.1536, Adjusted R-squared:  0.1356
## F-statistic: 8.527 on 1 and 47 DF,  p-value: 0.005359
```

```
ggplot(data = mod2) + geom_point(aes(x = fitted.values(mod2), y = residuals(mod2))) + labs(title
= "Residual Plot", y = "Residuals", x = "Fitted") + theme_classic()
```



```
#Adding a constant to log tranformation to avoid negative precipitation values
ggplot(data = accidents_prec, aes(x = (log(mean_prec) + 8), y = log(n))) + geom_point() +
  stat_smooth(method = "lm", formula = y~x, geom = "smooth") + theme_classic() + labs(title = "L
M Regression between Number of Accidents by State and Average Precipitation", x = "Average Preci
pitation (Log and Scaled)", y = "Number of Accidents (Log)")
```

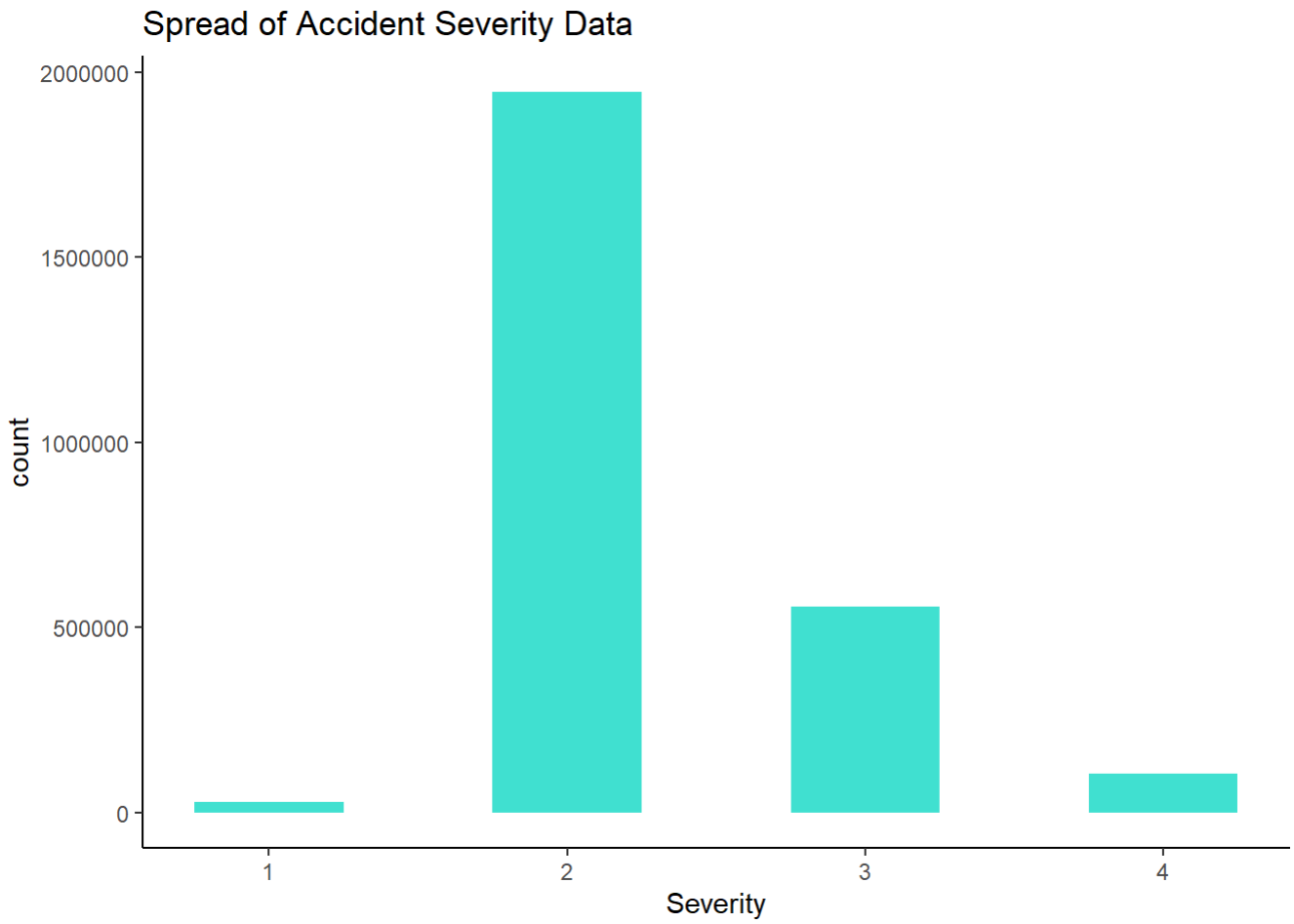


Model 2 is a linear regression model with a log transformation on both the predicted and explanatory variable. This model results in an adjusted R-squared of 0.135. The log of mean precipitation is statistically significant at $p < .01$ and the residual plot no longer shows a pattern implying that the problem of heteroskedasticity may have been resolved.

Can the severity of an accident be modeled?

Accident severity is a number between 1-4 that describes impact on traffic. A value of 1 indicates the least impact on traffic; whereas, a value of 4 indicates a significant impact on traffic. A random forest model was used to predict accident severity using a variety of predictors. As can be seen by the histogram, most of the accidents fall into the level 2 severity.

```
ggplot(accidents_merged)  + geom_histogram(aes(x = Severity), binwidth = .5, fill = "turquoise")
+ theme_classic() + labs(title = "Spread of Accident Severity Data")
```



The data is sorted and then split into training and test data. Since there are many data values, the training and test data is sampled to decrease the computation size. Additionally the Ranger library is used as a faster way to perform a random forest.

The random forest model was then used to create predictions. The predictions were then compared to the original values and if they were equal then a 1 was added to the result column, while a 0 was added if they were not equal. Finally, the mean of the result column was computed to calculate model accuracy.

```
#Creating Predictions
test_sample <- test_sample %>% mutate(predict = round(model_new$predictions))

#Checking if predictions match actual values
test_sample <- test_sample %>% mutate(result = ifelse(test_sample$Severity == test_sample$predic
t, 1, 0))

#Calculating Accuracy: 70.2%
mean(test_sample$result)
```

```
## [1] 0.7052775
```

The random forest model resulted in an accuracy of 70.2%

```
#Calculating Predictor Importance
important <- data.frame(importance(model_new)) %>% rename(importance = importance.model_new.)
important <- important %>% rownames_to_column(var = "variable")
important <- important %>% arrange(desc(importance))
knitr::kable(important[1:10,])
```

variable	importance
Date	0.2597087
Start_Lng	0.2262008
End_Lng	0.2202571
Start_Lat	0.2102728
End_Lat	0.2056943
Distance	0.1716071
Population	0.1113945
State	0.1075078
Duration	0.1013669
County	0.0431691

The most important features included Date, Latitude, Longitude, Duration, etc. A linear model was also created to compare its predictive power to that of the random forest.

```
#Model 2: Linear Model to Predict Severity
model_lm <- lm(formula = Severity ~ Duration  + Distance + Side + Humidity + Temperature + Preci
pitation + Sunrise_Sunset + day_week + State, data = train_sample)

#Model Summary
summary(model_lm)
```

```
##
## Call:
## lm(formula = Severity ~ Duration + Distance + Side + Humidity +
##     Temperature + Precipitation + Sunrise_Sunset + day_week +
##     State, data = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5605 -0.2097 -0.1027 -0.0233  2.2279
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.958e+00  8.632e-03 226.842  < 2e-16 ***
## Duration             1.275e-06  3.189e-07   3.999 6.35e-05 ***
## Distance             2.658e-02  6.030e-04  44.071  < 2e-16 ***
## SideR                8.274e-02  1.994e-03  41.498  < 2e-16 ***
## Humidity             6.232e-04  4.049e-05  15.394  < 2e-16 ***
## Temperature          1.883e-03  5.541e-05  33.988  < 2e-16 ***
## Precipitation        1.739e-01  1.599e-02  10.872  < 2e-16 ***
## Sunrise_Sunsetnan    1.691e-01  1.280e-01   1.321 0.186637
## Sunrise_SunsetNight -3.567e-02  1.724e-03 -20.695  < 2e-16 ***
## day_weekMonday      -7.866e-03  2.647e-03  -2.972 0.002958 **
## day_weekSaturday     3.946e-02  3.092e-03  12.759  < 2e-16 ***
## day_weekSunday       5.500e-02  3.268e-03  16.833  < 2e-16 ***
## day_weekThursday    -1.633e-02  2.598e-03  -6.287 3.24e-10 ***
## day_weekTuesday     -6.481e-03  2.621e-03  -2.473 0.013402 *
## day_weekWednesday   -1.371e-02  2.609e-03  -5.253 1.50e-07 ***
## StateAR              2.809e-02  1.604e-02   1.751 0.079916 .
## StateAZ             -2.359e-01  8.238e-03 -28.642  < 2e-16 ***
## StateCA             -1.118e-01  6.473e-03 -17.267  < 2e-16 ***
## StateCO              4.544e-01  9.196e-03  49.416  < 2e-16 ***
## StateCT              2.021e-01  1.059e-02  19.079  < 2e-16 ***
## StateDC              2.070e-01  1.658e-02  12.488  < 2e-16 ***
## StateDE              2.613e-01  1.892e-02  13.811  < 2e-16 ***
## StateFL             -7.714e-02  6.711e-03 -11.494  < 2e-16 ***
## StateGA              2.888e-01  8.369e-03  34.509  < 2e-16 ***
## StateIA              2.587e-01  1.530e-02  16.909  < 2e-16 ***
## StateID             -3.076e-02  1.760e-02  -1.748 0.080440 .
## StateIL              4.223e-01  8.183e-03  51.613  < 2e-16 ***
## StateIN              3.437e-01  1.160e-02  29.630  < 2e-16 ***
## StateKS              2.632e-01  1.697e-02  15.507  < 2e-16 ***
## StateKY              1.140e-01  1.265e-02   9.013  < 2e-16 ***
## StateLA             -6.984e-02  8.159e-03  -8.560  < 2e-16 ***
## StateMA              2.002e-01  1.079e-02  18.548  < 2e-16 ***
## StateMD              1.787e-01  8.708e-03  20.517  < 2e-16 ***
## StateME             -9.719e-02  2.877e-02  -3.378 0.000731 ***
## StateMI              2.226e-01  8.422e-03  26.425  < 2e-16 ***
## StateMN             -3.630e-02  7.614e-03  -4.767 1.87e-06 ***
## StateMO              3.159e-01  1.026e-02  30.800  < 2e-16 ***
## StateMS              9.291e-02  1.938e-02   4.795 1.63e-06 ***
## StateMT             -8.336e-02  1.779e-02  -4.687 2.77e-06 ***
## StateNC             -4.280e-02  7.275e-03  -5.884 4.01e-09 ***
```

```
## StateND             -1.384e-01  4.342e-02  -3.188 0.001435 **
## StateNE              4.462e-02  1.626e-02   2.745 0.006056 **
## StateNH              3.295e-02  1.877e-02   1.755 0.079220 .
## StateNJ              1.276e-01  8.641e-03  14.771  < 2e-16 ***
## StateNM              2.263e-01  2.396e-02   9.445  < 2e-16 ***
## StateNV              8.568e-02  1.791e-02   4.785 1.71e-06 ***
## StateNY              1.126e-01  7.248e-03  15.534  < 2e-16 ***
## StateOH              1.191e-01  9.196e-03  12.953  < 2e-16 ***
## StateOK             -5.351e-02  9.406e-03  -5.689 1.28e-08 ***
## StateOR             -8.790e-02  7.207e-03 -12.197  < 2e-16 ***
## StatePA              6.992e-02  7.568e-03   9.239  < 2e-16 ***
## StateRI              2.173e-01  1.743e-02  12.469  < 2e-16 ***
## StateSC             -4.784e-02  7.167e-03  -6.675 2.48e-11 ***
## StateSD              5.075e-02  7.074e-02   0.717 0.473142
## StateTN             -3.213e-02  8.093e-03  -3.970 7.18e-05 ***
## StateTX              7.403e-02  6.973e-03  10.616  < 2e-16 ***
## StateUT             -5.159e-02  8.513e-03  -6.059 1.37e-09 ***
## StateVA              4.797e-02  7.477e-03   6.416 1.40e-10 ***
## StateVT              1.488e-01  5.284e-02   2.816 0.004870 **
## StateWA              1.988e-01  9.264e-03  21.462  < 2e-16 ***
## StateWI              4.906e-01  1.374e-02  35.710  < 2e-16 ***
## StateWV              2.618e-02  2.254e-02   1.161 0.245448
## StateWY              7.516e-01  8.003e-02   9.392  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4779 on 399937 degrees of freedom
## Multiple R-squared:  0.09647,    Adjusted R-squared:  0.09633
## F-statistic: 688.7 on 62 and 399937 DF,  p-value: < 2.2e-16
```

The linear model is statistically significant and explains 9% of variance in accident severity. Some explanatory variables were dropped to fix the multicollinearity problem; however, the model still has a heteroskedasticity issue.

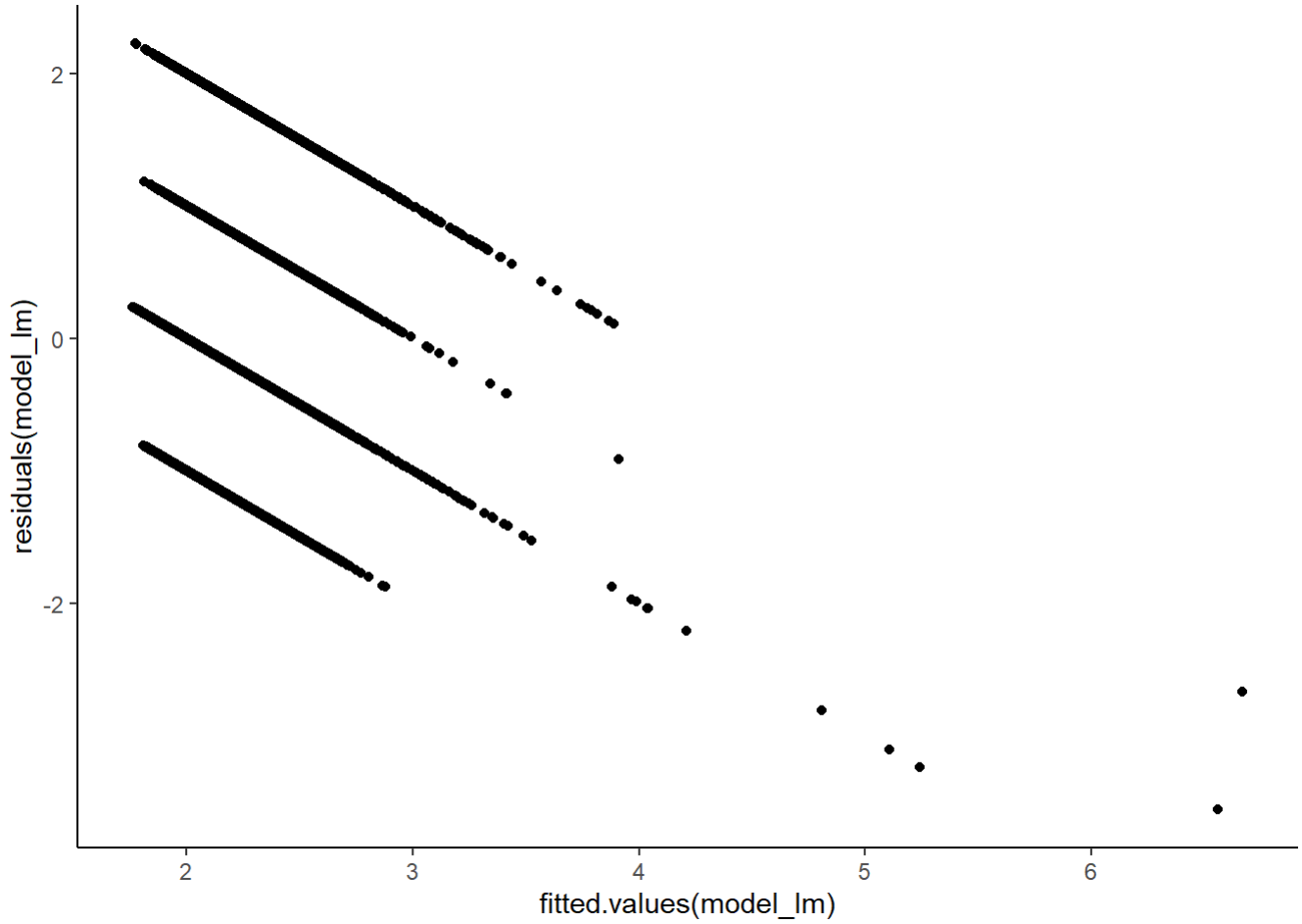
```
#Multicollinearity - fixed VIF < 5
vif(model_lm)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## Duration       1.004809  1        1.002402
## Distance       1.046240  1        1.022859
## Side           1.033923  1        1.016820
## Humidity       1.518001  1        1.232072
## Temperature    1.794157  1        1.339462
## Precipitation  1.024179  1        1.012017
## Sunrise_Sunset 1.245628  2        1.056445
## day_week       1.030207  6        1.002483
## State          1.852902 48        1.006445
```

```
#Heteroskedastic as p-value <.05
bptest(model_lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_lm
## BP = 24406, df = 62, p-value < 2.2e-16
```

```
#Residuals
ggplot(data = model_lm) + geom_point(aes(x = fitted.values(model_lm), y = residuals(model_lm)))
 + theme_classic()
```



There are apparent patterns in the residuals suggesting that the model is not great; however, the model accuracy still needs to be tested. Using a similar approach to the random forest, the linear model is used to create predictions, compare those predictions to the actual values, and then assign 1 or 0 to a result column depending on if the prediction was correct or incorrect respectively. The average of the result column was then computed to calculate model accuracy.

```
#Creating predictions
test_sample2 <- test_sample %>% add_predictions(model_lm) %>% mutate(pred = round(pred))
test_sample2 <- test_sample2 %>% mutate(result = ifelse(test_sample2$Severity == test_sample2$pr
ed, 1, 0))
#Model 2 Accuracy - 80%
mean(test_sample2$result)
```

```
## [1] 0.809025
```

The linear model resulted in a higher predictive accuracy of 80%.

Did Covid-19 have an effect on the number of accidents in 2020?

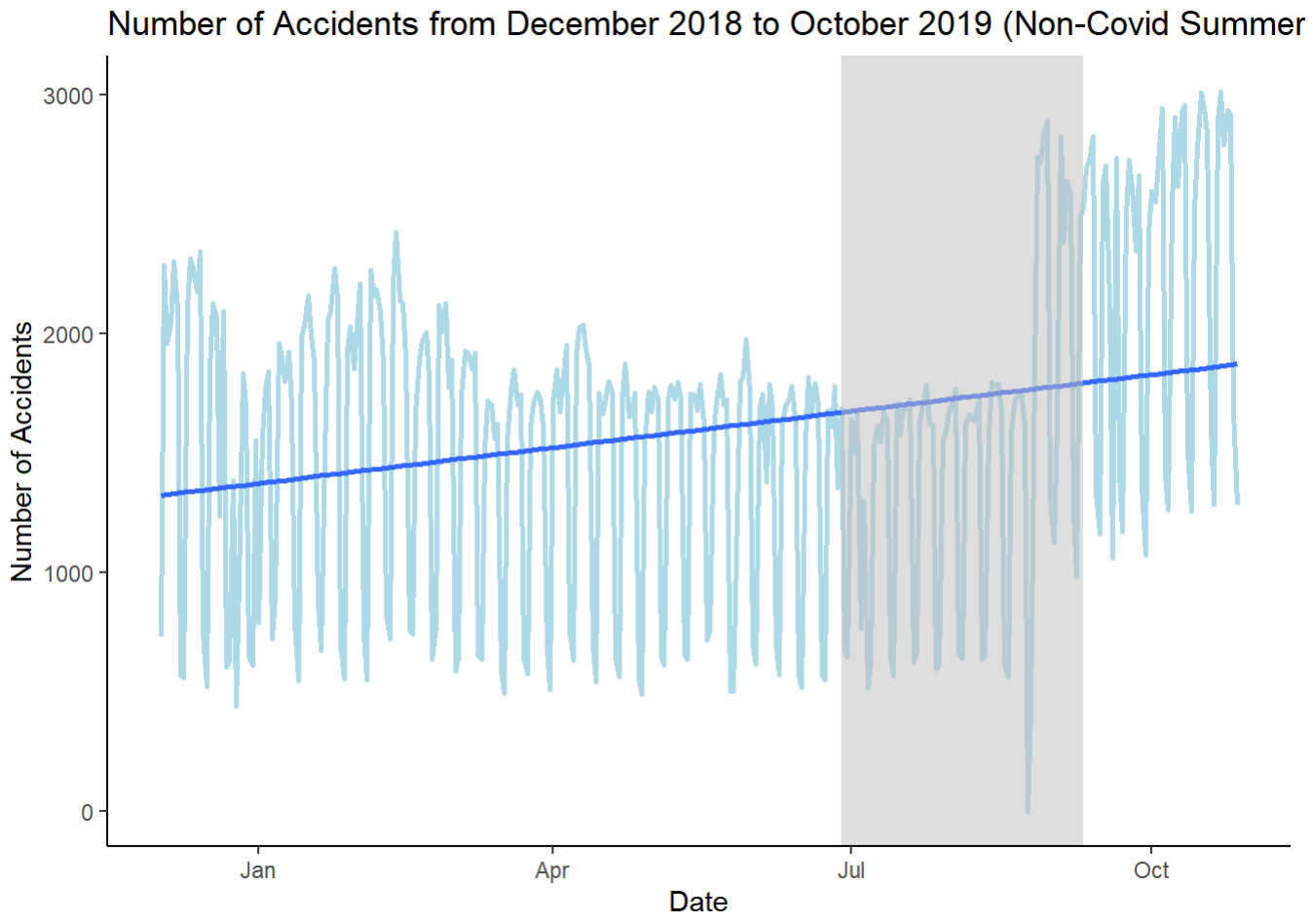
The accident data set was subsetted into two separate date intervals. One representing the summer of “Covid” (Dec 2019 - Oct 2020), while the other (Dec 2018 - Oct 2019) representing a “normal” summer.

```
#Number of Accidents over time - Covid 19
accidents_covid <- accidents_merged %>% subset(Date > "2019-12-01" & Date < "2020-10-28")
accidents_normal <- accidents_merged %>% subset(Date > "2018-12-01" & Date < "2019-10-28")
```

The time-series representing number of accidents over these intervals were then plotted.

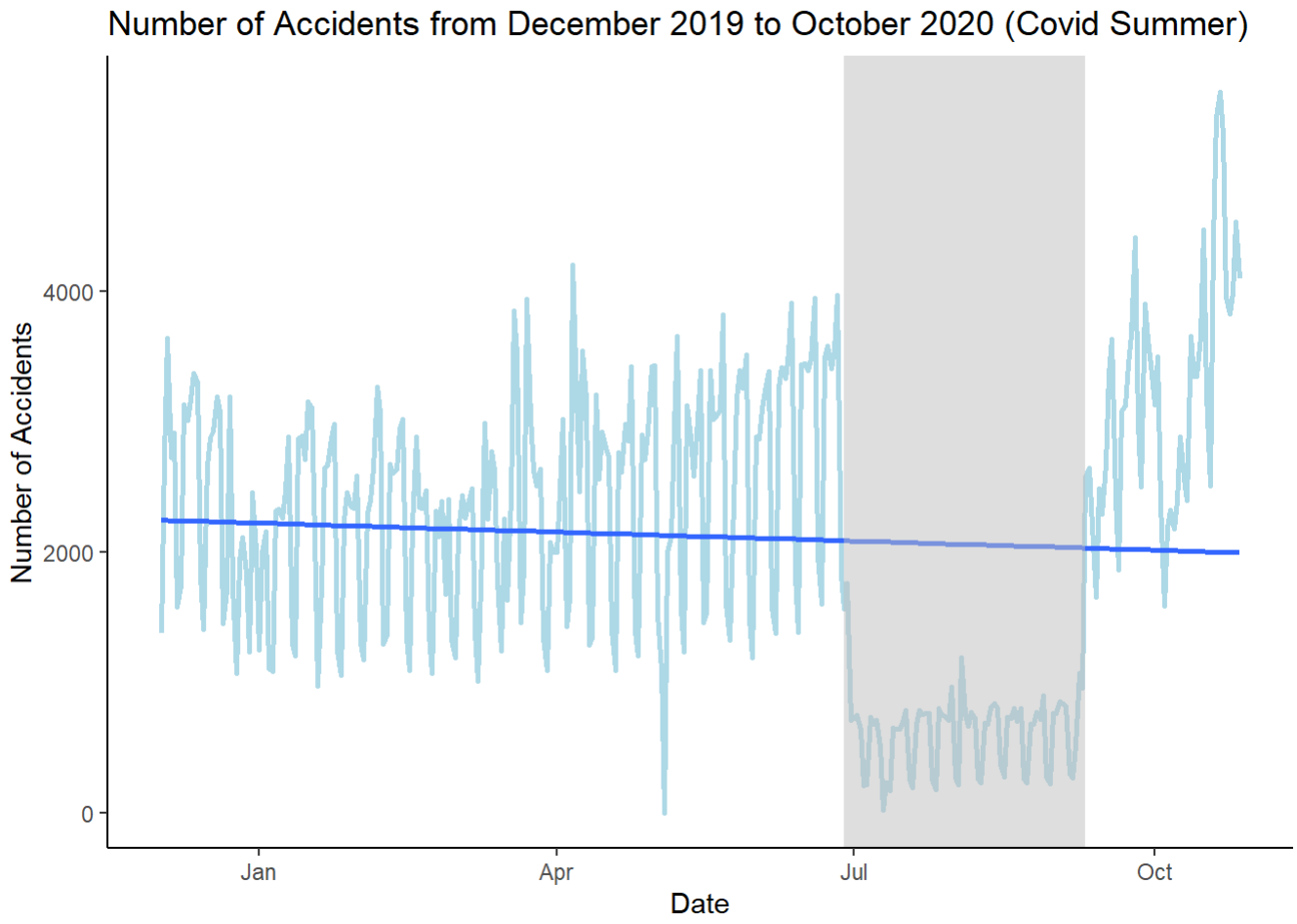
```
#Shaded Rectangles for Plots
rect_shade_covid <- data.frame(xmin =(as.Date(c("2020-06-28"))), xmax = (as.Date(c("2020-09-10"
))), ymin = -Inf, ymax = Inf)
rect_shade_normal <- data.frame(xmin =(as.Date(c("2019-06-28"))), xmax = (as.Date(c("2019-09-10"
))), ymin = -Inf, ymax = Inf)

#Non-Covid Summer Plot
accidents_normal %>%
  group_by(Date) %>%
  summarize(n = n()) %>%
  ggplot() + geom_freqpoly(aes(x = Date, y = n), stat = "identity", color = "light blue", size =
1) +
  theme_classic() +
  labs(title = "Number of Accidents from December 2018 to October 2019 (Non-Covid Summer)",
       x = "Date", y = "Number of Accidents") +
  geom_smooth(aes(x = Date, y = n), method = lm, se = FALSE) +
  geom_rect(data = rect_shade_normal, aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax),
            fill = "grey", alpha = 0.5)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
#Covid Summer Plot
accidents_covid %>%
  group_by(Date) %>%
  summarize(n = n()) %>%
  ggplot() + geom_freqpoly(aes(x = Date, y = n), stat = "identity", color = "light blue", size =
1) +
  theme_classic() + labs(title = "Number of Accidents from December 2019 to October 2020 (Covid
 Summer)",
                         x = "Date", y = "Number of Accidents") +
  geom_smooth(aes(x = Date, y = n), method = lm, se = FALSE) +
  geom_rect(data = rect_shade_covid, aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax),
            fill = "grey", alpha = 0.5)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



As can be seen by the Covid summer time-series plot, the shaded rectangle representing the late summer months has substantially less number of accidents than the surrounding months. Since the number of accidents may decrease during the summer months, an additional time series of a “normal” summer is presented. Since the “normal summer” plot does not show any decrease in the number of accidents during the summer, it can be assumed that the dip is related to the effects of Covid.

7. Conclusion

Three separate Bar charts illustrated how the number of accidents varies by state, how accidents per capita varies by state, and how the number of accidents varies by weekday. California was the state with the greatest number of accidents, while New York had the greatest number of accidents per capita. The most accidents occurred on Friday, while the least number of accidents occurred over the weekend. Additional visualizations could include accidents by time of day and by U.S. region.

A log transformed linear model predicting number of accidents using amount of precipitation resulted in an adjusted R-squared of 0.1356 with amount of precipitation being significant at the $\alpha = 1\%$ level. According to the model, precipitation explains 14% of variance in the number of accidents. Although there is a positive correlation between amount of precipitation and the number of accidents, the correlation is small. It does support the idea that more accidents will occur when the roads are wet; however, the small correlation implies that there may be confounding variables. Most likely there will be more people on the road when the weather is clear and therefore, traffic volume is a confounding variable that should be added to the data set. Moreover, the amount of precipitation from this data set may not be accurate; therefore, it would be prudent to get a data set (from the NOAA) with average precipitation amount by month and state to enhance the model.

A linear model and random forest predicted accident severity using various explanatory variables. The linear model had a greater accuracy then the random forest, which was surprising because there were numerous categorical variables involved in the prediction; however, taking a closer look at the data illustrates that most accidents fell into the Severity $= 2$ category. Since so many accidents fell in this category, the linear model was most likely predicting this level often leading to the high accuracy. The linear model had a good accuracy score, however, the model itself was not great as there was low correlation and heteroskedasticity in the residuals. The variables with the highest importance in the random forest included Date, Latitude, Longitude, Distance, etc. Future models to test may include a regression tree or XGBoost model to compare their respective predictive power with that of the linear model and random forest.

Additionally, a time-series depicting the number of accidents over 2020 illustrates that the number of accidents decreased and remained low during the Covid-19 summer of 2020. The time-series is compared to the earlier summer (2019), where a similar trend is non-existent implying that accidents do not decrease during the summer months. Since there was a substantial decrease during the summer of 2020, this decrease could be attributed to decreased travel as a result of the Covid pandemic. It is interesting to note that the decline in accidents occur suddenly at the end of June and raises the question of why there is a lag considering that lock down policies went into effect in late March of 2020. One possible explanation may be due to the supply shortages, which decreased the number of trucks on the road and hence accidents. Moreover, more people might have adjusted to working remotely from home and thus did not have to travel daily to work. On the contrary, Covid restrictions, and the decline in available flights may have encouraged more people to travel by road for vacations and hence prevented the number of accidents from dipping to much. A future modification to this data set may include joining a data set with traffic volume to uncover how traffic volume changed during the months leading up to the Covid summer of 2020.