Crash course - Decoding Road accidents in the USA

Vedenikova Vitalia / Perez Olusese

2024-06-06

Preface

As a foreword, please be aware that this work was made by a team of two people : Perez Olusese and Vitalia Vedenikova. Our team member Luis Marani quit the Master's in Applied Information and Data Science and has therefore not worked with us on this project.

The context of this report is as follows: we are a team of data scientists mandated by the US Department of Transportation in 2023 to investigate car accidents in the USA for the year 2022.

Introduction

In this report, we conduct a detailed statistical analysis of car accidents in the USA for the year 2022, focusing on the duration, frequency, and severity of accidents. By examining various variables, we aim to identify key predictors for the time it takes to resolve an accident, the occurrence of accidents, and their severity. Our findings reveal significant trends and correlations that can inform more effective road safety regulations and interventions. This analysis provides crucial insights that could help reduce the economic costs borne by taxpayers and enhance overall road safety.

We look forward to answering the following questions in our analysis.

- What variables are associated with longer times to resolve a car accident?
- What variables are associated with the occurrence an accident?
- What variables are the strongest predictors for the severity of an accident? Can it be accurately predicted?

Description of the data used

Two primary datasets serve as the cornerstone for our analysis. The first data can be found on Kaggle through this link sethttps://www.kaggle.com/datasets/sobhanmoosavi/us-accidents. The author provided a sampled data set for easier handling which we will be using here.

Despite this, the data set still had a large size (N = 5*10^5) and 55 predictors. Its data spans from 2016 to 2023 and was collected through the use of real-time data providers who source their information from the US and state departments of transportation as well as various other agencies and sensors. There are multiple continuous variables (temperature, precipitation level, visible distance, etc.) as well as categorical variables with more than two levels (weather conditions, timezone, etc.) which we made sure to keep for this analysis.

Our second dataset is from US Census Bureau with estimated populations per city. (source: https://www.census.gov/data/tables/time-series/demo/popest/2020s-total-cities-and-towns.html (https://www.census.gov/data/tables/time-series/demo/popest/2020s-total-cities-and-towns.html)): We included this data set because we would like to examine the population predictor in our models as well.

This paper does not contain the data transformation process as it is not the subject of our work.

It involved the following steps: 1. As a first step, we briefly examined the data set and filtered for the year 2022. 2. We removed unnecessary variables that were also full of NA values 3. We had to transform all units in the Imperial system of measurement to the metric system and rename some columns 4. We used stratified random sampling per US county to only keep 84% of the data and bring it below the size of N = 10^5 5. We categorized the streets into street types to be able to use them for our models 6. We calculated the duration of each accident,

categorized the times of accidents into time intervals (morning, afternoon, etc.) and kept only the months of occurrence from the dates. 7. We dropped all irrelevant variables 8. We augmented our data frame with cities' populations to take it into account during analysis 9. We handled missing data by replacing them by specifically defined means 10. We added the count of accidents per US state to be used in some models.

Full data preparation code can be read by pressing the button:

▶ Click here to see details

The resulting data set for this report covers all 49 US States and its attributes can be described as follows:

Data Summary

Variable	Description	Type
State	US state where the accident took place	Categorical
County	The type of street where the accident occured	Categorical
City	The city where the accident occured	Categorical
City Population	Population range of the city in which the accident took place	Continuous
Street	The type of street where the accident took place	Categorical
Number of Accidents	The number of car accidents per US state	Continuous
Severity	The severity of the accident with 1 being the lowest and 4 the highest	Categorical
Month	month of the accident	Categorical
Time Interval	Time interval in which the accident took place: early morning, morning, afternoon or night	Categorical
Duration	Duration of the accident in minutes	Continuous
Daytime or Nightime	If it is light or dark outside	Categorical
Weather Condition	The weather condition	Categorical
Temperature	Temperature in Celsius	Continuous
Precipitation	Amount of water on the road after rainfall in mm	Continuous

Data Summary		
Variable	Description	Type
Amenity	Presence of amenity (True or False)	Categorical
Crossing	Presence of crossing (True or False)	Categorical
Junction	Presence of junction (True or False)	Categorical
Station	Presence of station (True or False)	Categorical
Stop Sign	Presence of stop sign (True or False)	Categorical
Traffic Signal	Presence of traffic signal (True or False)	Categorical

Accident Duration

Car accidents involve a considerable amount of logistics: the appropriate amount of personnel and equipment has to be sent at the site of the crash, route deviations are organized and information has to be broadcast to drivers. Understanding what causes longer delays in resolving accidents is therefore crucial to ensure proper planning. For example, equipment could be made more readily available near certain locations at certain times, personnel shifts could be better planned and traffic surcharge in nearby locations could be avoided.

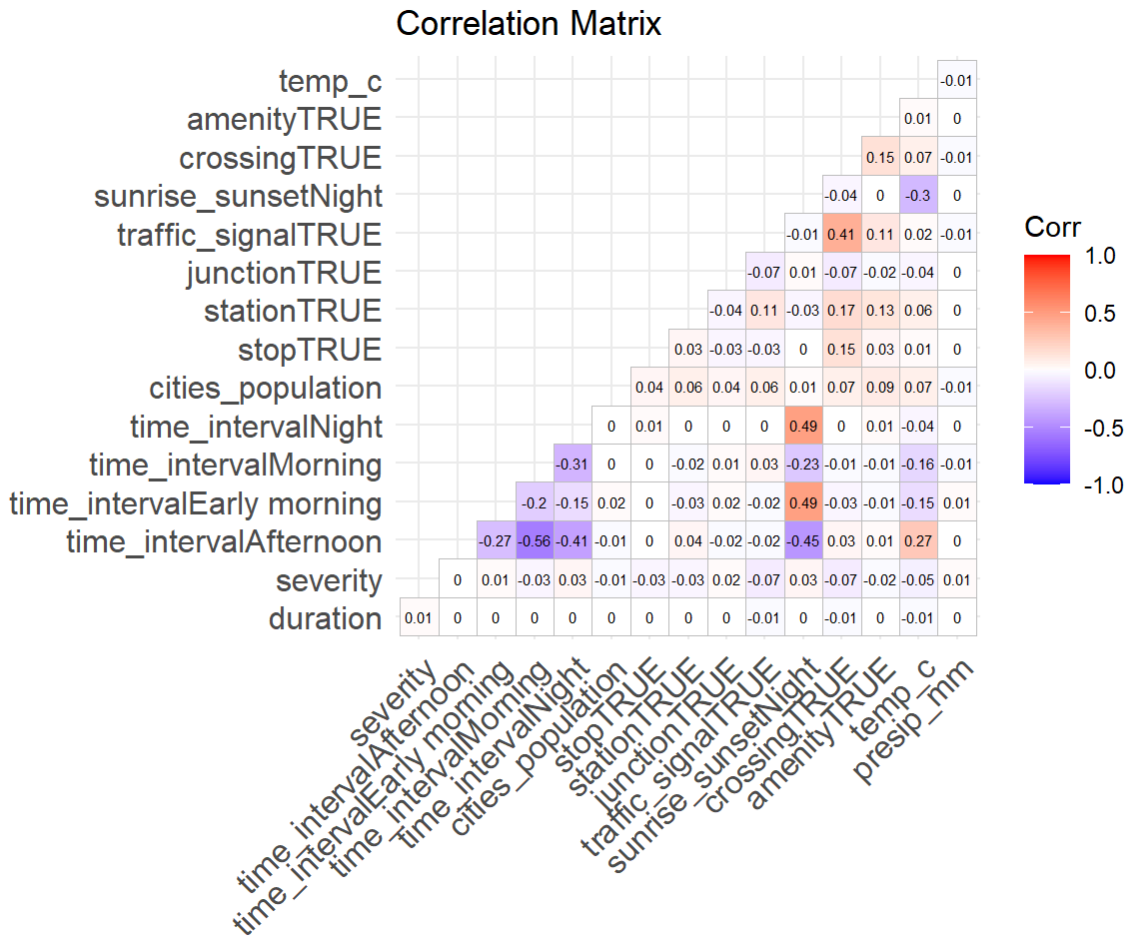
We will first proceed with a visual data analysis to determine which explanatory variables could be included in our Linear Model and the sort of interaction they could have with the duration of an accident.

Linear Model

Vitalia Vedenikova took the lead on the Linear Model sectionn

Exploratory visual analysis

In our data set, we notice that 74,4% of accidents get resolved in hours, while only 24.9% get resolved in minutes and a small percentage of 0.6% of accidents take several days to manage. To bring down the percentage of accidents resolved in hours, we want to first identify possible predictors by building a Correlation Matrix:



We would assume that an increase in population correlates to an increase in accident duration, but it does not appear to be the case as per the Correlation Matrix. The same can be said about the time intervals when accidents took place: they have no impact on duration. Lastly, temperatures did not seem to have an impact and neither did the precipitation level.

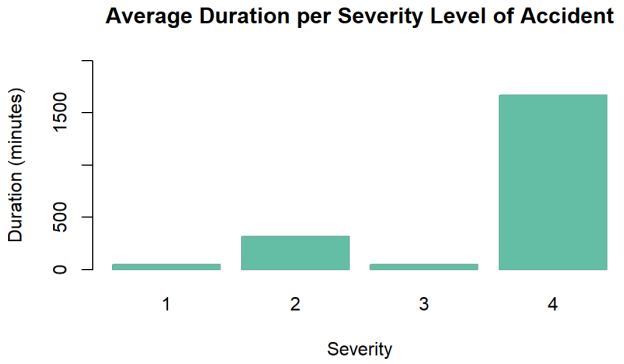
On the other hand, we can see that there is a possible weak negative correlation between duration and the presence of traffic signals and crossings. There might also be a positive correlation with severity. No obvious strong multicollinearity concerns can be observed.

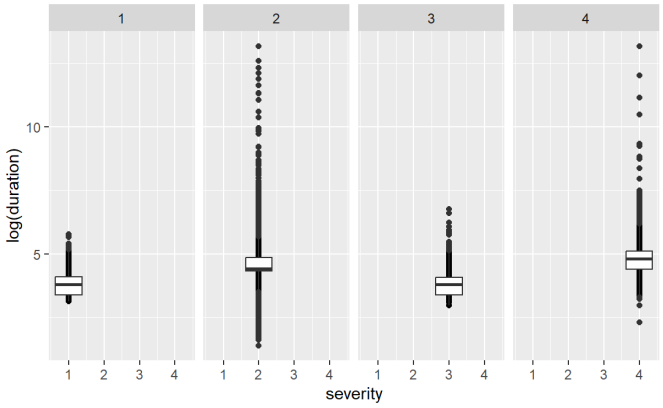
Duration and Severity

By fitting a bar plot showing the average duration per severity of accidents, we find that accidents with a long duration tend to have a severity of 4 out of 4 on average. In fact, some accidents of that severity took more than a day to resolve.

Examining the boxplots for each severity level with log transformed duration (due to the high quantity of outliers), the boxplot for severity 4 is higher than those of all other levels. This means that their duration tends to be higher. Their median is also higher, which is in line with what we evidenced with the barplot.

This is definitely a variable which we want to include in our Linear Model.

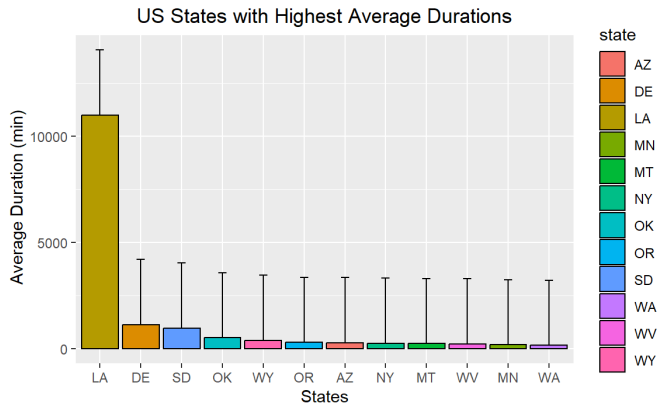




Duration and the Locations of the Accidents

US States

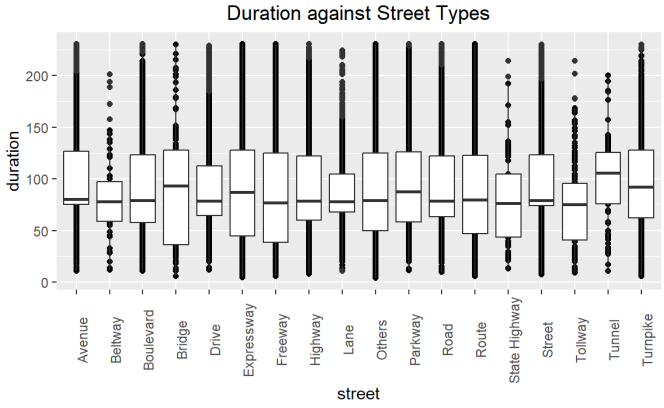
While severity is an obvious factor, the place where the accidents take place could also have an impact. The US States with highest times elapsed are Los Angeles and to a much lesser extent Delaware and South Dakota, so this variable would also be interesting to include in our model.

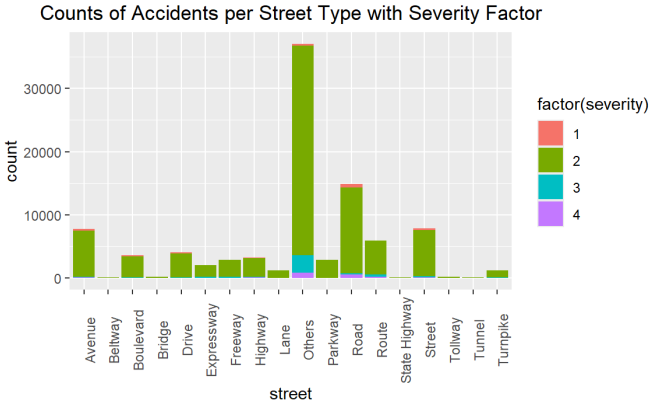


Street types

We notice that there appear to be some differences between the street types and notably that the median duration is highest for tunnels. This makes sense, since accidents in tunnels are more difficult to deal with due to increased danger.

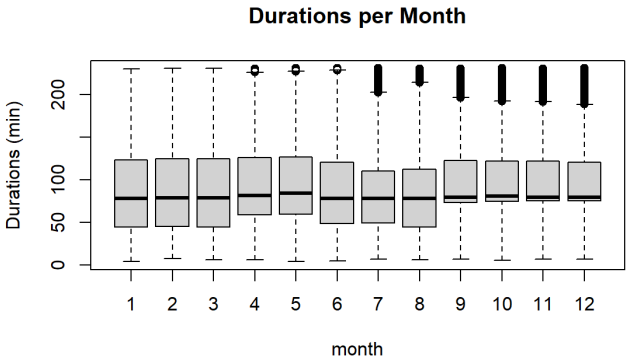
While looking at barplots of counts of accidents for each street type, we also notice that there are more sever accidents happening on “Others”, “Roads” and “Routes”. Since we know that severity is correlated to duration, this is another contributing factor to our decision to keep streets as a possible predictor (perhaps to be used in an interaction term).

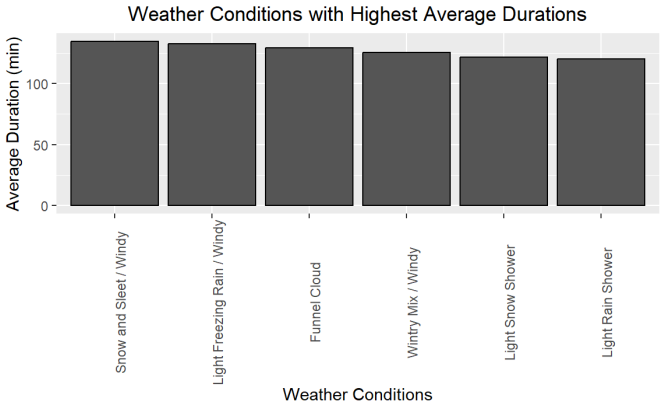




When Accidents Took Place and Weather Conditions

Looking at boxplots for each month, we notice that the boxplots for the months of June, July and August are lower than most of the boxplots from other months. January through March are also somewhat lower which translates to shorter durations. This might be due to fairer weather, and indeed the longest durations appear to coincide with bad weather conditions: snow and wind or rain and wind. We therefore would like to use the months and the weather conditions as predictors in our models.





To conclude our exploratory analysis, we retain the following possible predictors for a linear model to explain the duration: severity, months, US States, street types and weather conditions.

Linear Model Fit

We fit our model on the basis of the above predictors, making sure beforehand that they are all coded as categorical. However, once we examine the summary, we find that the weather_cond predictor does not seem to play an important role as its p-value is above 0.05. We therefore re-fit our model accordingly by removing it.

```
#We make sure that variables are correctly coded as categorical before fitting a mod
el
d.accidents <- d.accidents %>%
  mutate(across(c("severity", "month", "state", "street", "weather_cond"), as.facto
r))

#We fit our model
lm.accidents.1 <-lm(duration~ severity + state + month + street + weather_cond, data
=d.accidents)

#p-value  of weather_cond
summary_lm <- summary(lm.accidents.1)
summary_lm$coefficients["stateCO", "Pr(>|t|)"]
```

```
## [1] 0.7265082
```

```
#We re-fit the model
lm.accidents.2 <- update(lm.accidents.1, .~. - weather_cond)
```

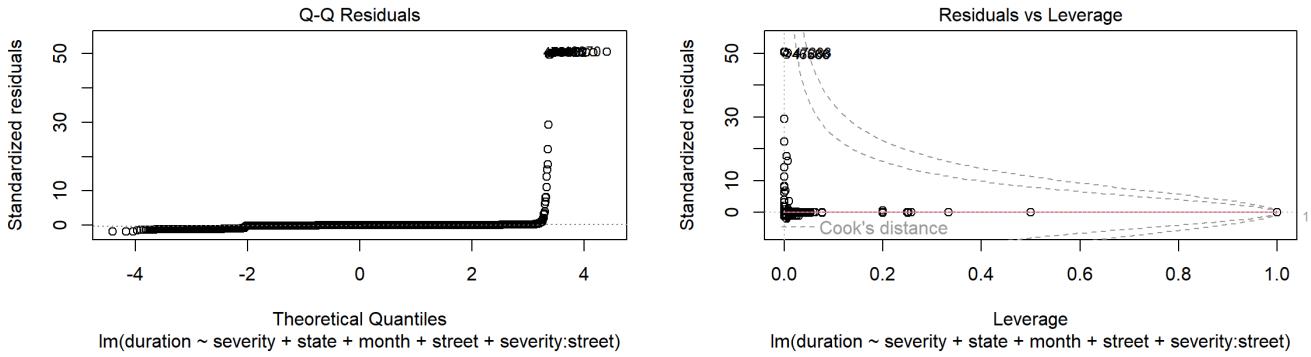
Earlier, we supposed that severity might interact with street types since certain streets appeared to have more severe accidents, so we add an appropriate interaction term which turns out to be significant due to its low p-value:

```
#Adding an interaction term:
lm.accidents.3 <-update(lm.accidents.2, . ~ . + severity*street)

#Single term deletions
drop1(lm.accidents.3, test="F")
```

```
## Single term deletions
##
## Model:
## duration ~ severity + state + month + street + severity:street
##                 Df  Sum of Sq        RSS     AIC F value    Pr(>F)
## <none>                       9.9647e+12 1765604
## state           48 2.0343e+11 1.0168e+13 1767440 40.6147 < 2.2e-16 ***
## month           11 1.1358e+10 9.9761e+12 1765690  9.8951 < 2.2e-16 ***
## severity:street 48 1.2799e+10 9.9775e+12 1765630  2.5554 1.878e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

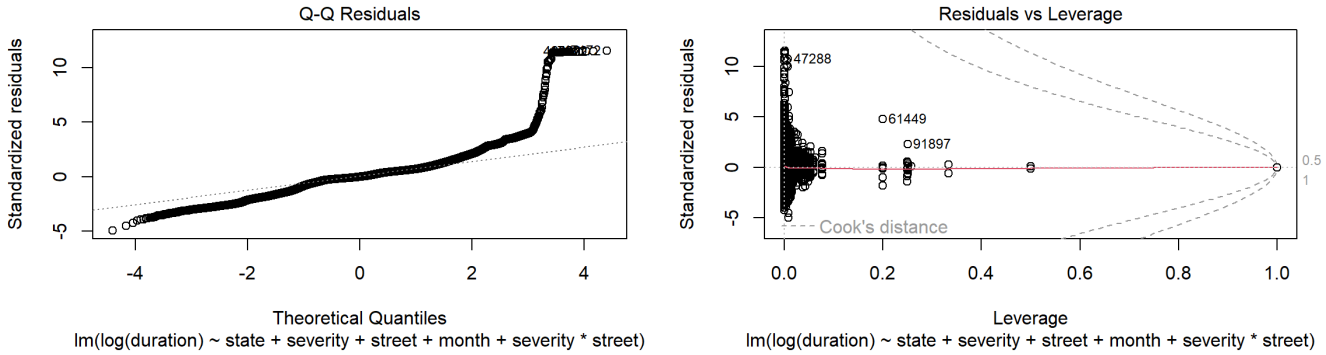
We now take a look at the residuals of our model. In the Residuals vs Leverage plot there is an extreme value above the Cook's Distance line of 1. This means that there is one very influential outlier in the data set. In the Q-Q Residuals plot, there are clear signs in the tails that the data

is not normally distributed, especially in the higher values. Overall, We may say that the amount of outliers is considerable.



For the above reasons, we log transform our explained variable and we re-examine the residuals. In the Q-Q Residuals plot, there are still indications that the data is not normally distributed since it does not fit on the line and is strongly skewed in the tails In the Residuals vs Fitted plot, many observations overlap and the smoother is not on zero indicating that the residuals are not normally distributed. Their variance appears to decrease with the fitted values so there is also probably heteroskedasticity. In the Residuals vs Leverage, we also still have an outlier above the Cook's Distance line of 1.

```
#Re-fit of the model with log transformed duration:
lm.accidents.4 <- lm(log(duration)~ state + severity + street + month + severity*str
eet, data=d.accidents)
```



For this linear model, we notice that the Multiple R-Squared and Adjusted R-squared are very low at 8.8% and 8.7% respectively. Therefore, we may say that the variance of the response variable is poorly explained by its relation to the explanatory variables. This being said, each variable was estimated significant while taking into account the effect of the other variables in the model. Indeed, the p-values of our model are all below 0.05 while using drop1().

R-Squared and Ajusted R-Squared:

```
## [1] 0.08879133
```

```
## [1] 0.08757947
```

```
#Single term deletions
drop1(lm.accidents.4, test="F")
```

```
## Single term deletions
##
## Model:
## log(duration) ~ state + severity + street + month + severity *
##     street
##                  Df Sum of Sq   RSS    AIC F value  Pr(>F)
## <none>                       52779 -56570
## state            48   1391.35 54170 -54178 52.4451 < 2e-16 ***
## month            11    143.78 52923 -56332 23.6485 < 2e-16 ***
## severity:street 48     37.32 52816 -56598  1.4066 0.03306 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Relevant coefficients from the lm.accidents.4 summary:

```
##              Estimate Std. Error    t value     Pr(>|t|)
## severity2  0.66345504 0.04993501 13.2863713 3.015780e-40
## severity3 -0.05294431 0.07927034 -0.6678956 5.042019e-01
## severity4  1.16397219 0.08763939 13.2813819 3.223235e-40
## stateSD    2.11871899 0.43061585  4.9202067 8.659637e-07
## stateMA   -0.17806227 0.04849747 -3.6715788 2.411881e-04
## month7     0.06340158 0.01214437  5.2206572 1.786614e-07
```

In conclusion, US States, months and the interaction term between street types and severity all have a relevant effect on the duration of an accident.

Some important insights we may gather from this model are that, if we take severity 1 as a reference level, severity 2 and 4 are respectively 0.663455 and 1.163972 units higher, meaning that duration is 93% and almost 300% higher in these cases (all else being equal). US States also differ greatly from each other. Taking Alabama as the reference, duration of accidents can be up to 8 times higher in South Dakota while in Massachusetts it is 19% lower (all else being equal). In terms of the month when an accident took place, we can definitely say that duration is not affected by the weather : indeed the reference month for our model is January when weather conditions can be particularly bad, yet accidents in July take 6% longer to resolve.

Generalised Additive Model

Vitalia Vedenikova took the lead on the Generalised Additive Model section

Generalized Additive Model Fit

For our GAM, we decide to add back the continuous variables we had examined in the exploratory visual analysis to see if we had missed a polynomial predictor. We do so using smoothers.

We find that contrary to the linear model, the GAM model identifies the temperature and the cities' populations as significant predictors while there is no strong evidence that the precipitation amounts' effect is different from zero. The edf value of the temperature is 6.3, meaning that there is a non-linear effect. Similarity, the cities' populations edf is of 8.5, meaning that there is a non-linear effect there too.

Overall, the model is barely better than the linear model, with only 9% of the variance of the response variable explained by its relation to the explanatory variable.

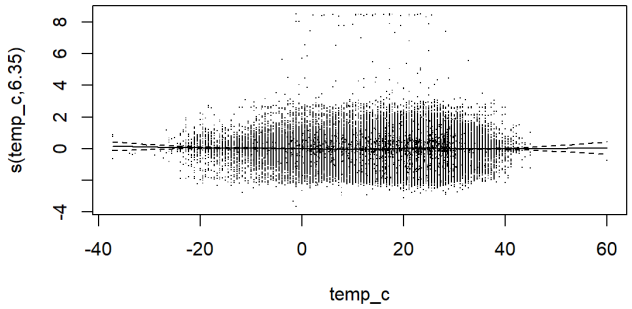
```
#GAM model fit
gam.accidents <-gam(log(duration)~ severity + state + month + street + severity*stre
et + s(temp_c) + s(presip_mm) + s(cities_population), data=d.accidents)

#Approximative significance of smooth terms
gam.summary <- summary(gam.accidents)
gam.summary$s.table
```

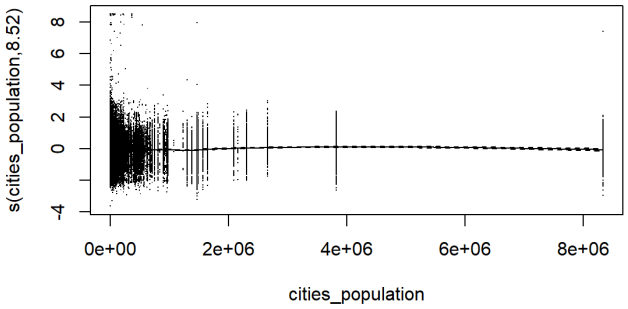
```
##                           edf   Ref.df         F   p-value
## s(temp_c)            6.348075 7.332586  9.522440 0.0000000
## s(presip_mm)         1.000000 1.000000  1.113267 0.2913756
## s(cities_population) 8.515673 8.892562 29.620843 0.0000000
```

```
gam.summary$r.sq #r-squared value
```

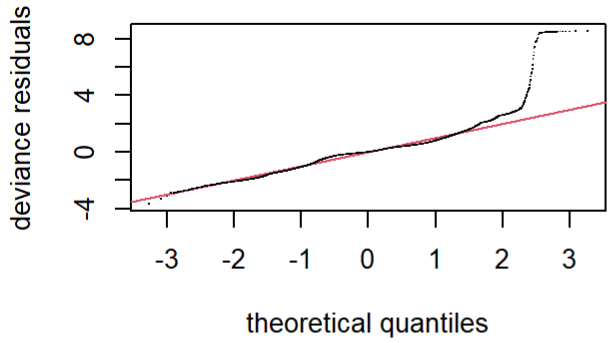
```
## [1] 0.09074888
```

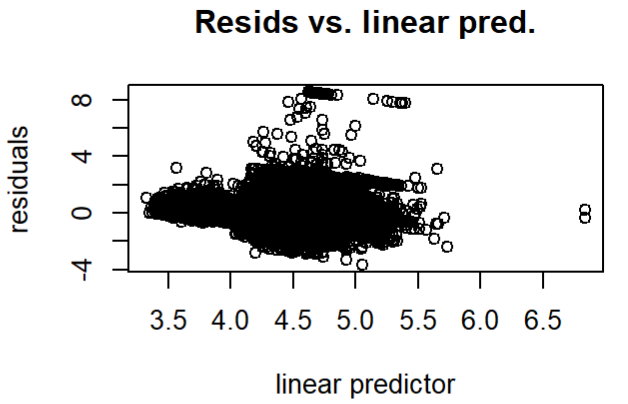
As we already know from the linear model QQ-Plots, our data is very skewed and does not fulfill normality assumptions. This therefore explains why with a GAM we also have a very poorly fitting model.
Looking closer at the estimated smooth functions of the variables with strong evidence of significance, we cannot really see visual evidence of polynomials, so perhaps our model is also overfitted.

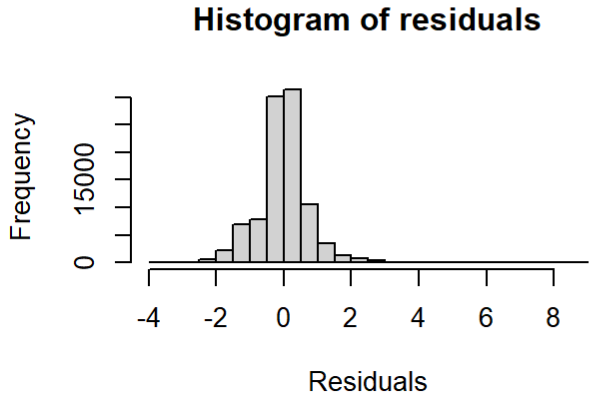


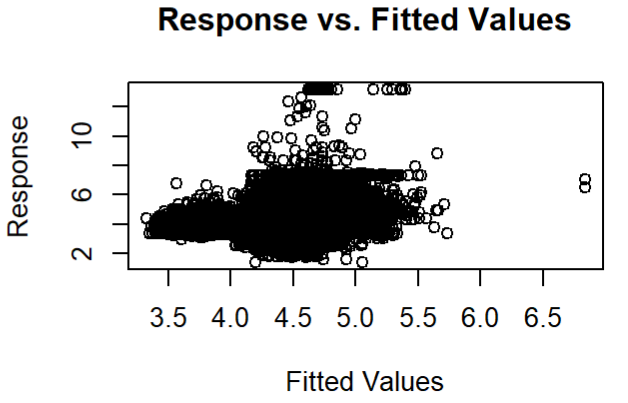


Once we run gam.check(), we see that the k-index is very close to one, so there is no missed pattern left in the residuals. We cannot adjust the k to optimize our model further as this suggests there are no significant patterns in the residuals and there are enough basis functions for all smoother terms. The Response vs Fitted Values plot also clearly suggests that the model is a very bad fit since there is no 1 to 1 line present.









```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 12 iterations.
## The RMS GCV score gradient at convergence was 1.239408e-07 .
## The Hessian was positive definite.
## Model rank =  155 / 158
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                         k'  edf k-index p-value
## s(temp_c)            9.00 6.35    0.99    0.28
## s(presip_mm)         9.00 1.00    0.99    0.26
## s(cities_population) 9.00 8.52    0.99    0.17
```

In conclusion, there is poor evidence that the GAM model is a good fit for modelling the duration of car accidents. It does however suggest that there might be a non-linear effect from cities' populations and temperature which we might have overlooked by only using the linear model.

Number of accidents per state

As we previously noticed, US States were a statistically significant variable in predicting the duration of accidents. It is also easy to observe that the number of accidents varies a lot per State. In this section of our work, we strive to understand why this is the case to be able to address the causes and provide suggestions on how to reduce the amount of accidents in the future.

GLM Models

Perez Olusese took the lead on the Poisson and Binomial models

GLM - Poisson

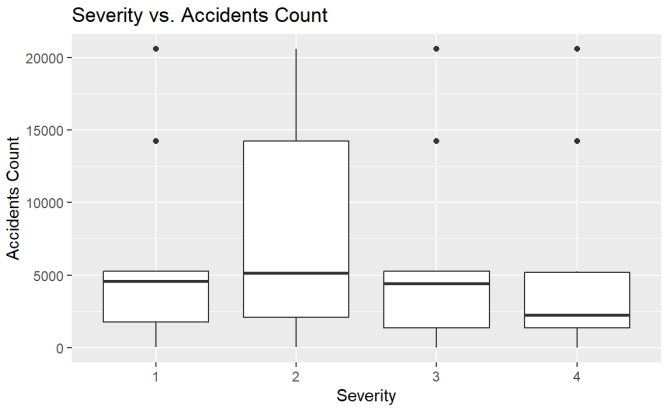
The Generalized Linear Model (GLM) with a Poisson distribution is a powerful tool for count data analysis, ideal for scenarios where the response variable represents non-negative integer counts.The count data in our dataset is the accident count per state. Data compatibility will be considered for this analysis to ensure the independent variables are suitable for this analysis. state, city, and county are not appropriate as independent variables in this Poisson model as they are categorical variables with very many levels.

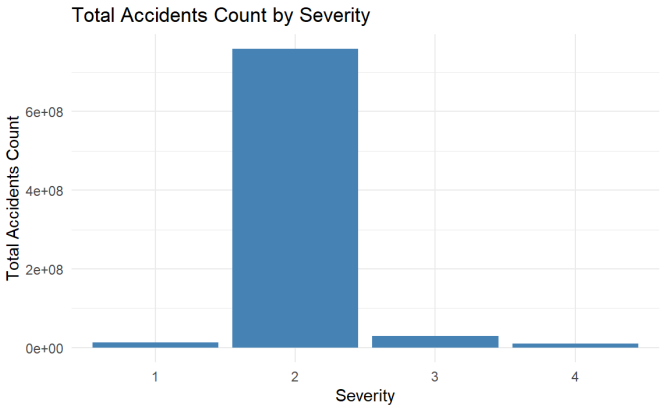
Exploratory data analysis

The independent variables that will look at exploring for this are severity, weather, month and time interval. We will therefore start the exploratory data analysis with these variables.

Severity

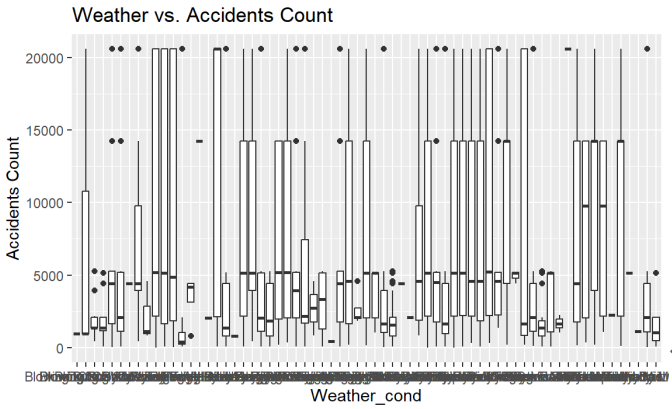
The boxplot and barplot helps visualize the distribution of accident counts across different severity levels. This visualization can reveal whether higher severity levels are associated with higher or accident counts.

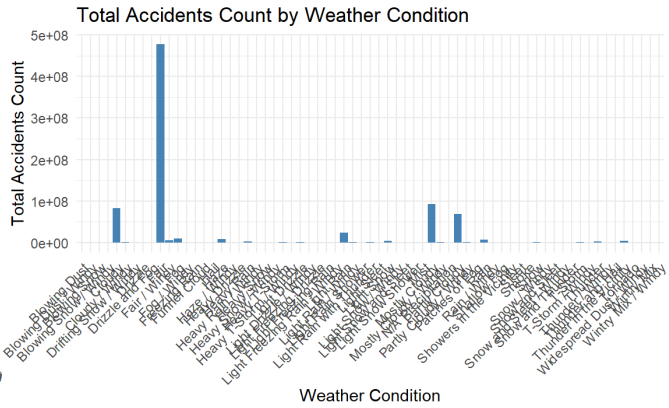




Weather Conditions

Weather conditions can be considered as a factor that could affect the number of accidents count per state. To justify this assumption we will also do an Explanatory data analysis so as to see if of weather conditions on the accident counts per state.



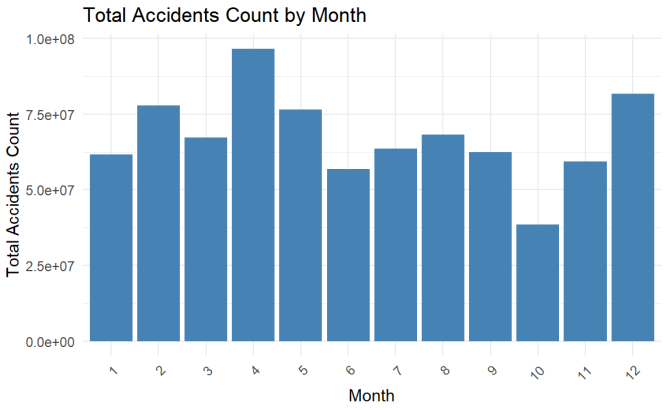


There are some weather conditions that show some evidence that accidents counts increase during those conditions.we shall therefore use variable in our analysis.

Month

We are going to consider months in our study so as to establish whether the month actually has an effect on the accident count per state.

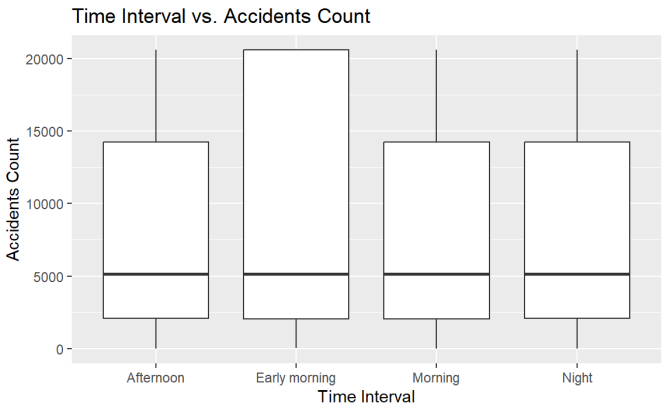
From the Bar plot we can see the variation of accident counts in every month with the highest accidents appearing in month 4.

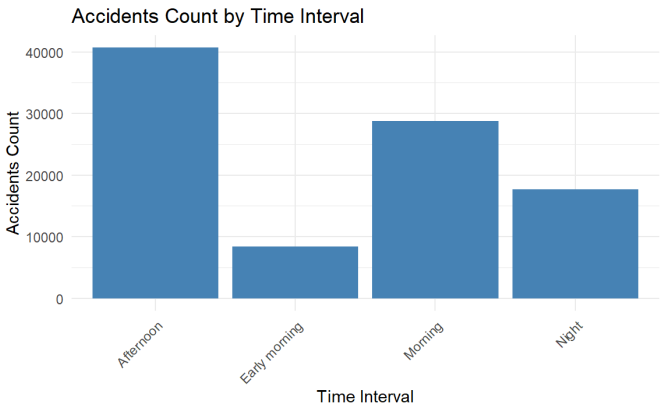


Time interval

The time at which an accident occurs is a crucial factor in our analysis. We aim to investigate whether the timing of accidents significantly impacts the total number of accidents. By examining patterns related to the time of day, we seek to identify peak periods for accidents and understand how temporal factors contribute to accident frequency. This analysis will help us determine if specific times are associated with higher accident counts, providing valuable insights for improving traffic management and road safety interventions.

It is evident that the time when an accident occurs affects the accident count per state. we will therefore use this variable in our analysis.





In conclusion we will use the following predictors for our poisson model. -severity -Weather conditions -month -time interval

Poisson model fitting

```
glm_accident_1 <- glm(accidents_count_per_state ~ severity + weather_cond + month +
time_interval,
                      data = d.accidents,
                      family = poisson)
```

```
## ```
##
## Call:
## glm(formula = accidents_count_per_state ~ severity + weather_cond +
##     month + time_interval, family = poisson, data = d.accidents)
##
## Coefficients:
##                                          Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                             6.506e+00  3.258e-02  199.673  < 2e-16
## severity2                               3.014e-01  2.889e-04 1043.351  < 2e-16
## severity3                               3.533e-02  3.410e-04  103.602  < 2e-16
## severity4                              -3.742e-01  4.222e-04 -886.164  < 2e-16
## weather_condBlowing Dust / Windy        2.134e+00  3.326e-02   64.167  < 2e-16
## weather_condBlowing Snow                7.145e-01  3.303e-02   21.631  < 2e-16
## weather_condBlowing Snow / Windy        6.523e-01  3.282e-02   19.875  < 2e-16
## weather_condCloudy                      2.062e+00  3.258e-02   63.294  < 2e-16
## ...
## month12                                ***
## time_intervalEarly morning             ***
## time_intervalMorning                   ***
## time_intervalNight                     ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 651469040  on 95620  degrees of freedom
## Residual deviance: 602625613  on 95537  degrees of freedom
## AIC: 603613620
##
## Number of Fisher Scoring iterations: 6
##
## ```
```

Upon examining the Poisson model, we observe that the residual deviance is 602,625,613 while the degrees of freedom are 95,537, indicating that our model is over-dispersed. In a well-fitting Poisson model, these two values would be approximately equal. To address this over-dispersion, we will construct a quasi-Poisson model to accurately account for the dispersion parameters and improve the model's fit.

```
quasi_accident_1 <- glm(accidents_count_per_state ~ severity + weather_cond + month
+ time_interval,
                 data = d.accidents,
                 family = quasipoisson)
```

```
## ```
##
## Call:
## glm(formula = accidents_count_per_state ~ severity + weather_cond +
##     month + time_interval, family = quasipoisson, data = d.accidents)
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           6.506026   2.574831   2.527 0.011513
## severity2                             0.301423   0.022830  13.203  < 2e-16
## severity3                             0.035328   0.026946   1.311 0.189846
## severity4                            -0.374177   0.033367 -11.214  < 2e-16
## weather_condBlowing Dust / Windy      2.134082   2.628142   0.812 0.416787
## weather_condBlowing Snow              0.714519   2.610292   0.274 0.784292
## weather_condBlowing Snow / Windy      0.652283   2.593415   0.252 0.801417
## weather_condCloudy                    2.062267   2.574725   0.801 0.423153
## ...
## month12
## time_intervalEarly morning
## time_intervalMorning                 ***
## time_intervalNight                   .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 6244.576)
##
##     Null deviance: 651469040  on 95620  degrees of freedom
## Residual deviance: 602625613  on 95537  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
##
## ```
```

From the quasi-Poisson model, we observe that the dispersion parameter is 6,244.576 compared to the fixed value of 1 in the Poisson model. This indicates that the variance increases faster than linearly.

Interpreting some results

```
## interpreting weather conditions
exp(coef(glm_accident_1)["weather_condDrizzle and Fog"])
```

```
## weather_condDrizzle and Fog
##                    2.450437
```

when the weather condition changes to drizzle and fog we expect the accident count per state to go higher by 24.5%.

Poisson conclusion

When we do a comparison of the estimated model coefficients from the quasi-poisson model we notice that they remain identical to those in the Poisson model, but the standard errors and p-values are adjusted in the quasi-Poisson model. In conclusion.The analysis reveals minimal evidence that factors such as time interval significantly affect the accident count per state. However, certain weather conditions, certain severity levels, and certain months do have a notable impact on the accident count per state.

Severity of Accidents

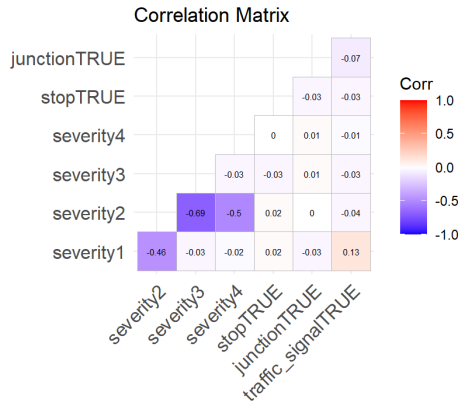
As much as the amount of accidents, the severity of the accidents are an important aspect to consider. Preventing serious accidents could prevent loss of life, serious injury as well as excessive material damage. It is therefore crucial to understand their causes.

GLM -Binomial

The binomial model utilizes binary data for its analysis. To adapt our dataset for this model, we will convert the severity variable into a binary format. The original severity variable has four levels, ranging from one to four. We will combine levels 1 and 2 into a single category, labeled as 0, representing low severity. Similarly, we will merge levels 3 and 4 into another category, labeled as 1, representing high severity. This transformation will allow us to effectively apply the binomial model to our analysis.

```
# Convert severity to binary
d.accidents_binary <- d.accidents %>%
  mutate(severity_binary = ifelse(severity %in% c(1, 2), 0, 1))
```

In this analysis, we are utilizing the boolean variables present in our dataset to gain deeper insights. These boolean variables, which represent binary states such as true/false or yes/no, will help us effectively categorize and analyze different aspects of the data. By leveraging these variables, we can streamline our analysis, enhance the accuracy of our models, and better understand the underlying patterns and trends in our dataset.



We are going to use the junction, stop and traffic_signal to build the glm binomial model since there seems to be some correlation between the severity and these variables. No obvious multicollinearity concerns can be observed.

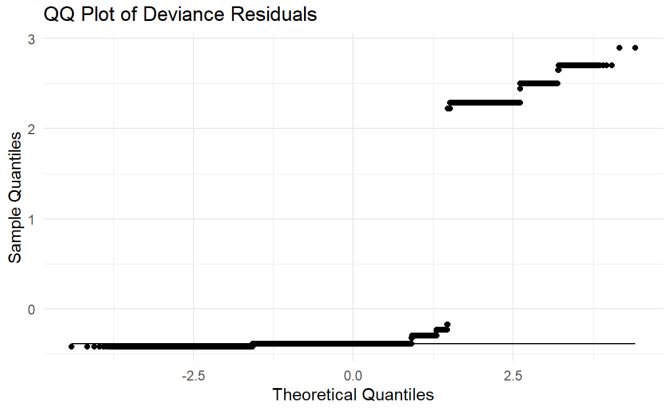
```
##
## Call:
## glm(formula = severity_binary ~ junction + stop + traffic_signal,
##     family = binomial, data = d.accidents_binary)
##
## Coefficients:
##                     Estimate Std. Error  z value Pr(>|z|)
## (Intercept)        -2.54161    0.01371 -185.391  < 2e-16 ***
## junctionTRUE        0.15128    0.04799    3.152  0.00162 **
## stopTRUE           -1.08414    0.12541   -8.645  < 2e-16 ***
## traffic_signalTRUE -0.54911    0.05499   -9.985  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 48416  on 95620  degrees of freedom
## Residual deviance: 48183  on 95617  degrees of freedom
## AIC: 48191
##
## Number of Fisher Scoring iterations: 6
```

Due to the use of a link function, we must transform the coefficients to accurately interpret them. Our analysis reveals little evidence that stops and traffic signals significantly affect the severity of accidents. However, there is some evidence suggesting that junctions have an impact on accident severity. By interpreting the transformed coefficients, we can better understand how the presence of junctions influences the severity of road accidents, potentially indicating that certain junction designs or traffic conditions at junctions may contribute to more severe accidents.

```
exp(coef(glm_binomial_1)["junctionTRUE"])
```

```
## junctionTRUE
##      1.16332
```

We can see that severity of road accidents increase by about 11.6% at junctions.

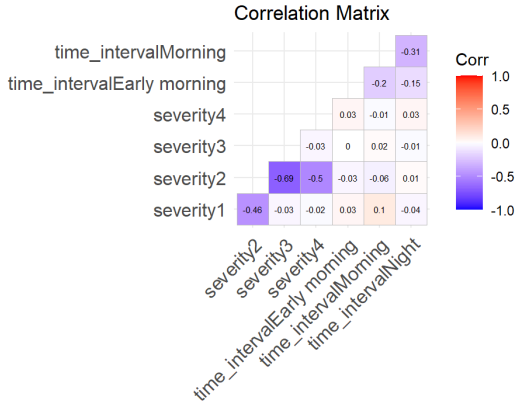


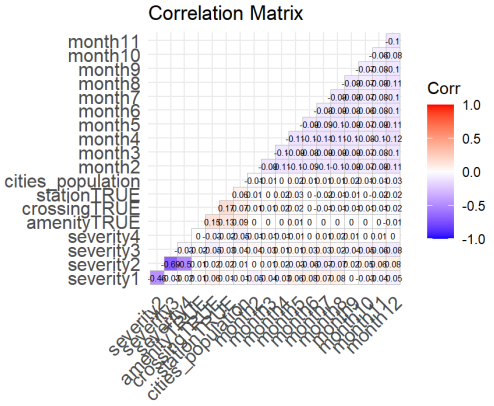
From this analysis we can see that the severity of accidents increases at junctions.This plot shows that this model does not fit our data.

Support Vector Machine

Vitalia Vedenikova and Perez Olusese worked on this model

Now that we examined severity with a binomial model, we would like to increase the amount of predictors we use and check how accurately we can predict the severity of an accident. Support Vector Machines are particularly well adapted to the task as they are robust models capable of classifying accidents into various categories. In order to perfect our work, we will also supplement this method with cross validation techniques.





As a first step, we look at the types of predictors we could use with the help of two correlation matrices. What observe some level of collinearity with traffic sign variables, time intervals, months and cities populations. No obvious multicollinearity concerns can be observed. Street types are also chosen as a predictor since serious accidents often occur on high speed roads.

Our next step is to balance our data set. Indeed, there 13 times more accidents of a low severity than of a high one.

```
##
##     0     1
## 88945  6676
```

We therefore balance these two groups' counts via stratified sampling to obtain equal proportions of minor and major accidents. If the count sizes are not equal, then the model's true positive and true negative rates can be significantly reduced and lead to faulty classifications. The resulting sampled data is as follows:

```
##
##    0    1
## 6676 6676
```

Initial SVM model fitting

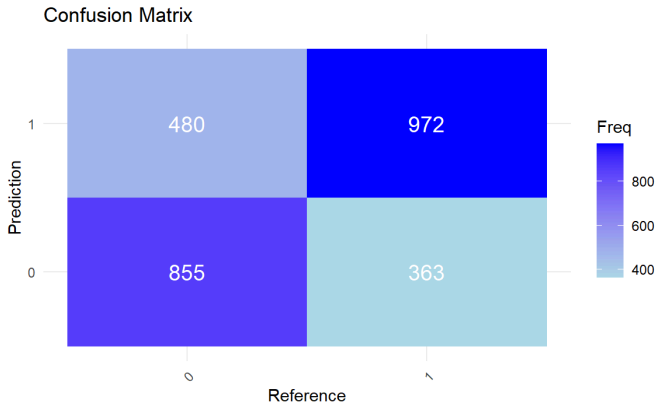
```
# Convert boolean columns to factors
d.accidents_vector[, (bool_cols) := lapply(.SD, as.factor), .SDcols = bool_cols]

# Define the target variable and features
y <- d.accidents_vector$severity
X <- d.accidents_vector[, !("severity"), with = FALSE]

# Split the data into training and testing sets
set.seed(123) # For reproducibility
trainIndex <- createDataPartition(y, p = 0.8, list = FALSE, times = 1)
X_train <- X[trainIndex,]
X_test <- X[-trainIndex,]
y_train <- y[trainIndex]
y_test <- y[-trainIndex]

# Set seed for reproducibility
set.seed(123)

# Train the SVM model with the RBF kernel-linear and cost set to 10
svm_model <- svm(
  as.factor(y_train) ~ .,
  data = X_train,
  kernel = "linear",
  cost = 10,
)
```

Predicting the test set and evaluating the model:

```
##### Make predictions on the test set #####
predictions <- predict(svm_model, X_test)

##### Evaluate the model #####
conf_matrix <- confusionMatrix(predictions, as.factor(y_test))
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 855 363
##          1 480 972
##
##                Accuracy : 0.6843
##                  95% CI : (0.6663, 0.7019)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3685
##
##  Mcnemar's Test P-Value : 6.463e-05
##
##             Sensitivity : 0.6404
##             Specificity : 0.7281
##          Pos Pred Value : 0.7020
##          Neg Pred Value : 0.6694
##              Prevalence : 0.5000
##          Detection Rate : 0.3202
##    Detection Prevalence : 0.4562
##       Balanced Accuracy : 0.6843
##
##        'Positive' Class : 0
##
```

```
##### Defining the Confusion matrix #####
# Convert confusion matrix to a data frame for ggplot2
conf_df <- as.data.frame(conf_matrix$table)
colnames(conf_df) <- c("Prediction", "Reference", "Freq")
```



At this stage, the accuracy of our model is of 68%. It has a sensitivity that is at 64%. It means that it is quite accurate in dividing serious and minor accidents. The specificity, which refers to the true negative rate, is somewhat high at 73%. The Pos Pred Value, which refers to the precision is acceptable at 70%, meaning that there is a high proportion of predicted positive cases that are correctly identified. There is a lower proportion of negative cases that are correctly identified at 67%. Overall, we may say that our model is rather consistent with most indicators being between 64% and 73% but not very accurate in its predictions.

We try to fit a model with RBF kernel-radial to see if we could get better results. However, as the accuracy turns out to be slightly lower and the sensitivity 10% inferior, we discard this model.

```
# Train the SVM model with the RBF kernel-radial and cost set to 100
svm_model <- svm(
  as.factor(y_train) ~ .,
  data = X_train,
  kernel = "radial",
  cost = 100,
)
```

```
##  Accuracy
## 0.6794007
```

```
## Sensitivity
##   0.5550562
```

In this next step, we use different cost parameters to define seven models, then we use cross validation to compare them and return the max accuracy we could find across the models. The best one is 68.2% accurate, so we concede that we cannot find a better SVM model for our accident severity prediction.

```r
# Set seed for reproducibility
set.seed(123)

# Number of folds
k <- 5

# Create folds
folds <- createFolds(y, k = k, list = TRUE)

# Initialize a vector to store accuracy for each fold
accuracy <- numeric(k)

# Perform cross-validation
  for (i in 1:k) {
  # Split the data into training and testing sets
  train_indices <- folds[[i]]
  X_train <- X[-train_indices,]
  y_train <- y[-train_indices]
  X_test <- X[train_indices,]
  y_test <- y[train_indices]

  # Train the SVM model with linear kernel
  svm_model <- svm(as.factor(y_train) ~ ., data = X_train, kernel = "linear",
                   ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))

  summary(svm_model)

  # Make predictions on the test set
  predictions <- predict(svm_model, X_test)
  table(predictions)

  # Calculate accuracy
  accuracy[i] <- mean(predictions == y_test)
}

# Print the average accuracy
max_accuracy <- max(accuracy)
print(max_accuracy)
```

```
## [1] 0.6827715
```

In conclusion, while our binomial model allowed us to understand the types of relations between the severity and some predictors, with out SVM model we tested out their predictive power with measures of accuracy, sensitivity and so on. Although our prediction model was not

the best at around 68% adjusted accuracy, it still tells us that variables related to various traffic signs, the cities' populations, street types and time intervals all have a somewhat strong impact on the severity of an accident.

Artificial Neural Network

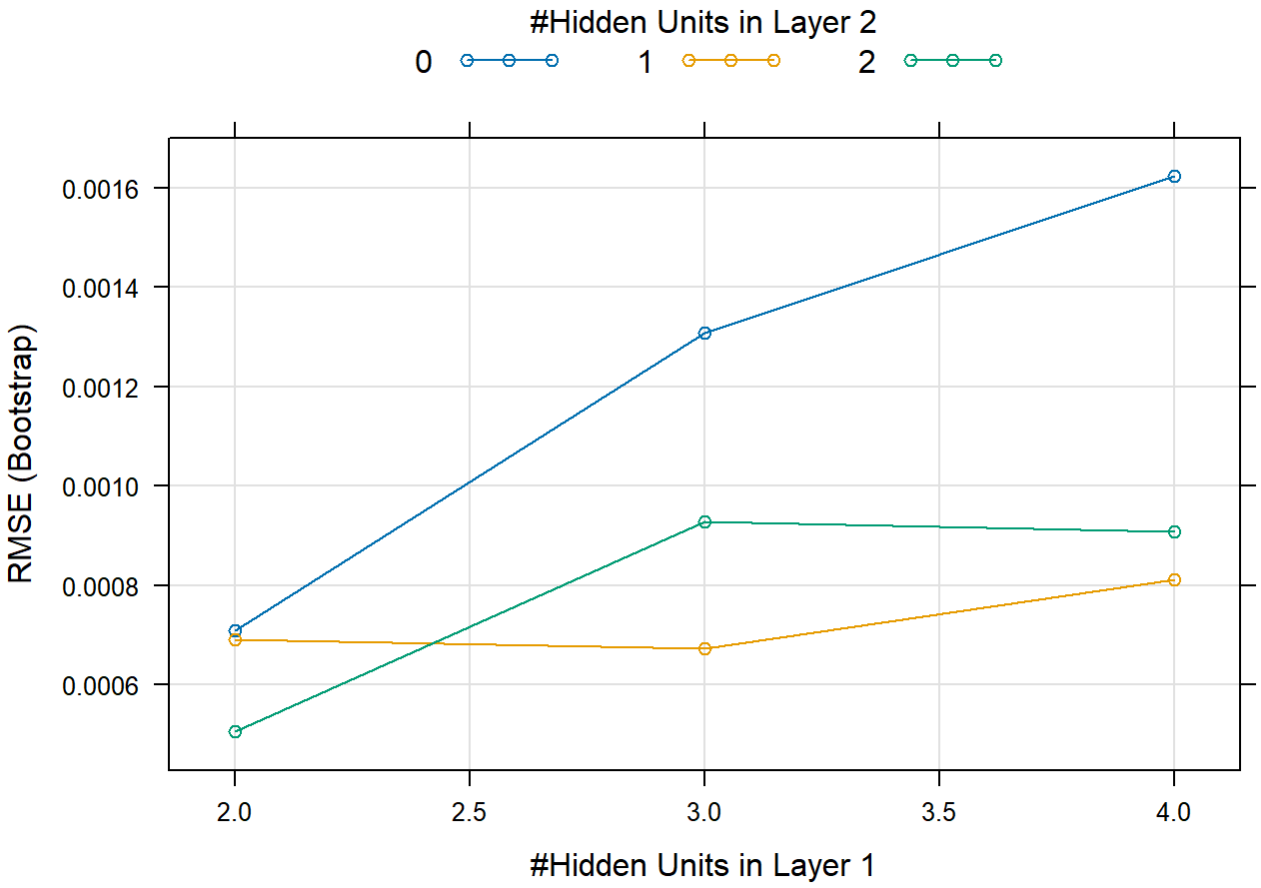
Vitalia Vedenikova and Perez Olusese worked on this model

Another method well adapted in theory to the prediction or classification of severity of accidents are Neural Networks. This method does however involve significantly more computing power than a Support Vector Machine model, which means that its application is more time consuming and prone to errors. Because of this, we had to remove cities populations from our predictors. Same as for the SVM, we used a balanced data set where the count of major and minor accidents is equal.

As further steps, we encoded dummy variables, normalized the numerical variables and lastly partitioned our data for training. We also made sure that the formula for our model was correctly defined in order to avoid using wrong predictors.

Then, we look at the different models:

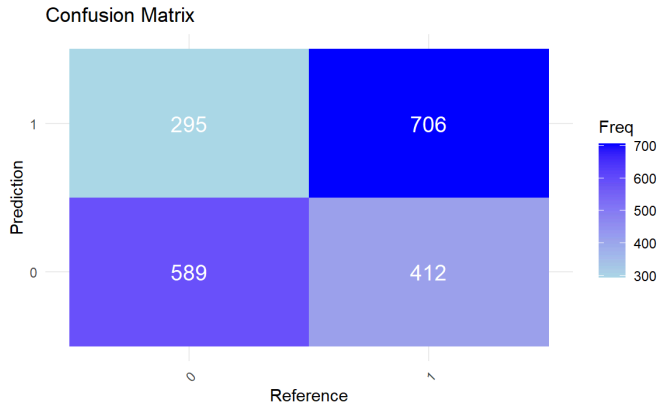
```
#We will implement parallel processing to make the process more efficient:
library(doParallel)
cl <- makePSOCKcluster(detectCores() - 1)
registerDoParallel(cl)

#Then we optimize the network structure
set.seed(142)

models <- train(formula, train_mod,
                method="neuralnet",
                tuneGrid = expand.grid(.layer1=c(2:4), .layer2=c(0:2), .layer3=c(0)),
                learningrate = 0.001,
                threshold = 0.05,
                stepmax = 70000
)

#stop the cluster
stopCluster(cl)
registerDoSEQ()

plot(models)
```



The best model for this data appears to be one with 2 hidden units in layer 1 and 0 hidden units in layer 2. This is the one which we test out now:

```r
#We will implement parallel processing to make the process more efficient:
cl <- makePSOCKcluster(detectCores() - 1)
registerDoParallel(cl)

#We implement the best model:
set.seed(42)
best_model <- neuralnet(
    formula,
    train,
    hidden = c(2),
    learningrate = 0.001,
    threshold = 0.05,
    stepmax = 100000
  )

#stop the cluster
stopCluster(cl)
registerDoSEQ()

#We look at the confusion matrix and evaluate our model:
test_results <- neuralnet::compute(best_model, test_in)
test_pred <- apply(test_results$net.result, 1, which.max)
test_pred <- factor(levels(test_truth)[test_pred], levels = levels(test_truth))
confusionMatrix(test_truth, test_pred)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 589 412
##          1 295 706
##
##                Accuracy : 0.6469
##                  95% CI : (0.6255, 0.6678)
##     No Information Rate : 0.5584
##     P-Value [Acc > NIR] : 5.245e-16
##
##                   Kappa : 0.2937
##
##  Mcnemar's Test P-Value : 1.285e-05
##
##             Sensitivity : 0.6663
##             Specificity : 0.6315
##          Pos Pred Value : 0.5884
##          Neg Pred Value : 0.7053
##              Prevalence : 0.4416
##          Detection Rate : 0.2942
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.6489
##
##        'Positive' Class : 0
##
```



Note that while this is not shown here, we did run tests with many different settings for hidden layers, the learning rate and the stepmax. We are only showing the optimal model in terms of accuracy and other metrics.

The accuracy of our Neural Network Model is of 65%, which is slighlty lower than that of the Support Vector Machine. Its specificity, which refers to the true negative rate, is much lower at 63%. Furthermore, its precision is much lower at 59%, so it does not identify positive cases very well. It does however identify negative cases better with a rate of 71% compared to the rate

of 67% from the SVM. It is also more dispersed, since its confidence interval is larger than that of the SVM model. Surprisingly, it seems to be much worse than the SVM at predicting serious accidents, since a lot of them were misclassified as low severity. This is concerning and would lead us to prefer the SVM model, which does misclassify minor accidents as serious more often, but would prove less risky to use as a prediction model in a real life scenario. Indeed, it would be better to identify accidents as serious than minor in most cases.

Conclusion

After investigating our three main topics of the duration, amount and severity of car accidents in the USA, we draw the following overall conclusions:

We found that US States, months and the interaction term between severity and street type all had a strong association with the duration of an accident. Our GAM model also revealed a possible influence from temperature levels and cities' populations. In particular, a sharp increase in duration coincided with higher severity levels, suggesting a positive correlation. Interestingly, the weather conditions did not appear to have much importance, since it took less time to deal with accidents in January than in warmer months. Perhaps emergency services' response times are already optimized for all weather conditions. Nevertheless, Los Angeles stood out in how long its durations are, so a separate, in-depth study of this State's road infrastructure is advised.

From our models we also found that weather conditions show evidence that they do affect accident occurrence. This is evidenced by the fact that the number of accidents per state are affected by certain weather conditions.

When it comes to measuring severity, it is evident from our models that junctions increase the chances of severe road accidents, perhaps because bad drivers might not slow down at those places. Perhaps accident-prone junctions need to be identified and replaced with roundabouts.

Ultimately, the predictors we identified allowed us to fit Support Vector Machine and Neural Network prediction models that were capable of predicting severity of accidents up to a 68% level of accuracy. To increase their prediction power, more highly correlated variables could be identified and previous years considered in our data set.

Use of Generative AI

We utilized ChatGPT-3.5 to enhance the structure and grammar of our report, ensuring clarity and professionalism. It assisted us in understanding and resolving errors encountered during the coding process. Additionally, ChatGPT-3.5 provided valuable guidance and structure when certain aspects were unclear, helping us to articulate our ideas more effectively and maintain a cohesive flow throughout the report. Since we did not have much experience with functions in R, we also used it to help us define certain variables in the data preparation process and to experiment with sampling methods for the four levels of severity in the context of the SVM and ANN models.

Reflection

This project presented a steep learning curve for us as a team of two people, but it offered invaluable insights into the real-world tasks and responsibilities of data scientists. During the process, we encountered numerous challenges that tested our problem-solving skills and deepened our statistical knowledge. We gained hands-on experience with data collection, cleaning, and preprocessing, which made us realize the importance of data preparation. Analyzing complex datasets and applying advanced statistical methods allowed us to see the practical applications of theoretical concepts we learned. Moreover, we realized how difficult it could be to build machine learning models and draw meaningful conclusions from our findings. Overall, this project truly enhanced our technical skills and adaptability as data scientists.