

Project

Namitha Kuthani

2025-05-01

```
# -----  
# Accident Severity Analysis in the U.S.  
# Goal: Predict and understand accident severity based on weather, traffic, and road-related factors  
# -----  
  
# Load necessary libraries (install only if not already installed)  
  
# Load libraries  
library(dplyr)  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union  
  
library(lubridate)  
  
## Warning: package 'lubridate' was built under R version 4.4.3  
  
##  
## Attaching package: 'lubridate'  
  
## The following objects are masked from 'package:base':  
##  
##   date, intersect, setdiff, union  
  
library(car)  
  
## Loading required package: carData  
  
##  
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':  
##  
##      recode
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.3
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##  
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':  
##  
##      combine
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v forcats 1.0.0      v stringr 1.5.1  
## v ggplot2 3.5.1      v tibble 3.2.1  
## v purrr 1.0.2        v tidyr 1.3.1  
## v readr 2.1.5
```

```
## -- Conflicts ----- tidyverse_conflicts() --  
## x randomForest::combine() masks dplyr::combine()  
## x dplyr::filter()         masks stats::filter()  
## x dplyr::lag()            masks stats::lag()  
## x ggplot2::margin()       masks randomForest::margin()  
## x car::recode()           masks dplyr::recode()  
## x purrr::some()           masks car::some()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(usmap)
```

```
## Warning: package 'usmap' was built under R version 4.4.3
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.4.3
```

```
##  
## Attaching package: 'reshape2'  
##  
## The following object is masked from 'package:tidyr':  
##  
## smiths
```

```
# -----  
# Step 1: Load and sample data (50,000 rows for faster analysis)  
set.seed(123)  
# Load sampled data for analysis  
small_data <- read.csv("small_data_1M.csv")  
  
# -----  
# Step 2-3: Select and clean relevant columns  
small_data_selected <- small_data %>%  
  select(Severity, Distance.mi., Temperature.F., Humidity...,  
         Visibility.mi., Wind_Speed.mph., Pressure.in., Precipitation.in.,  
         Weather_Condition, Start_Time, Sunrise_Sunset, Traffic_Signal, State,  
         Bump, Crossing, Give_Way, Junction, No_Exit, Railway,  
         Roundabout, Station, Stop, Traffic_Calming, Turning_Loop) %>%  
  drop_na()  
  
# -----  
# Step 4: Rename columns for clarity  
small_data_clean <- small_data_selected %>%  
  rename(  
    Distance = Distance.mi.,  
    Temperature = Temperature.F.,  
    Humidity = Humidity...,  
    Visibility = Visibility.mi.,  
    Wind_Speed = Wind_Speed.mph.,  
    Pressure = Pressure.in.,  
    Precipitation = Precipitation.in.  
  )  
  
# -----  
# Step 5: Feature engineering (create derived variables)  
small_data_clean <- small_data_clean %>%  
  mutate(  
    Start_Hour = hour(ymd_hms(Start_Time)),  
    Rush_Hour = ifelse(Start_Hour %in% c(7:9, 16:18), 1, 0),  
    Weekend = ifelse(weekdays(ymd_hms(Start_Time)) %in% c("Saturday", "Sunday"), 1, 0),  
    Weather_Simple = case_when(  
      Weather_Condition %in% c("Clear", "Fair", "Fair / Windy", "Scattered Clouds") ~ "Clear",  
      Weather_Condition %in% c("Cloudy", "Mostly Cloudy", "Overcast", "Partly Cloudy", "Partly Cloudy /  
      Weather_Condition %in% c("Rain", "Light Rain", "Heavy Rain", "Drizzle", "Light Drizzle", "Light R  
      Weather_Condition %in% c("Snow", "Light Snow", "Heavy Snow", "Snow / Windy", "Blowing Snow / Windy
```

```

    Weather_Condition %in% c("T-Storm", "Thunder", "Thunderstorm", "Heavy Thunderstorms and Rain") ~
    Weather_Condition %in% c("Fog", "Shallow Fog", "Patches of Fog", "Haze", "Haze / Windy", "Smoke",
    Weather_Condition %in% c("Light Freezing Rain", "Light Freezing Drizzle") ~ "Freezing",
    TRUE ~ "Other"
  ),
  Is_Daytime = ifelse(Sunrise_Sunset == "Day", 1, 0),
  Traffic_Signal_Flag = ifelse(Traffic_Signal == "True", 1, 0)
)

# -----
# Step 6: Create Road_Features (composite variable from multiple indicators)
road_cols <- c("Bump", "Crossing", "Give_Way", "Junction", "No_Exit", "Railway",
              "Roundabout", "Station", "Stop", "Turning_Loop")

small_data_clean[road_cols] <- lapply(small_data_clean[road_cols], function(x) x == "True")
small_data_clean$Road_Features <- as.integer(rowSums(small_data_clean[road_cols]) > 0)

# -----
# Step 7: Filter cleaned dataset
accident_data <- small_data_clean %>%
  filter(Weather_Simple %in% c("Clear", "Cloudy", "Rainy", "Snowy", "Freezing")) %>%
  mutate(
    Weather_Simple = as.factor(Weather_Simple),
    Severity = as.numeric(Severity),
    State = as.factor(State)
  )

# -----
# Step 8A: Fit full multiple linear regression model
model_mlr_full <- lm(Severity ~ Distance + Temperature + Humidity + Visibility +
  Wind_Speed + Pressure + Precipitation + Weather_Simple +
  Rush_Hour + Weekend + Is_Daytime + Traffic_Signal_Flag +
  Road_Features + State,
  data = accident_data)

summary(model_mlr_full)

```

```

##
## Call:
## lm(formula = Severity ~ Distance + Temperature + Humidity + Visibility +
##     Wind_Speed + Pressure + Precipitation + Weather_Simple +
##     Rush_Hour + Weekend + Is_Daytime + Traffic_Signal_Flag +
##     Road_Features + State, data = accident_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47278 -0.19881 -0.11054 -0.05027  2.11565
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.5950093   0.1498403   10.645 < 2e-16 ***
## Distance       0.0053316   0.0020688    2.577 0.009969 **
## Temperature    0.0008871   0.0002134    4.157 3.23e-05 ***

```

## Humidity	0.0001366	0.0001885	0.725	0.468617	
## Visibility	-0.0005111	0.0017401	-0.294	0.768963	
## Wind_Speed	0.0014019	0.0006153	2.278	0.022713	*
## Pressure	0.0185696	0.0049255	3.770	0.000164	***
## Precipitation	0.0977542	0.0393247	2.486	0.012933	*
## Weather_SimpleCloudy	0.0165064	0.0072149	2.288	0.022158	*
## Weather_SimpleFreezing	0.1252862	0.1017392	1.231	0.218171	
## Weather_SimpleRainy	0.0699659	0.0142645	4.905	9.42e-07	***
## Weather_SimpleSnowy	0.0636951	0.0229935	2.770	0.005609	**
## Rush_Hour	-0.0123014	0.0067573	-1.820	0.068705	.
## Weekend	0.0394273	0.0081614	4.831	1.37e-06	***
## Is_Daytime	-0.0098625	0.0077350	-1.275	0.202306	
## Traffic_Signal_Flag	-0.1132145	0.0098002	-11.552	< 2e-16	***
## Road_Features	-0.0418940	0.0080230	-5.222	1.79e-07	***
## StateAR	-0.0371110	0.0569372	-0.652	0.514545	
## StateAZ	-0.1127937	0.0343822	-3.281	0.001038	**
## StateCA	-0.1132593	0.0263194	-4.303	1.69e-05	***
## StateCO	0.3210967	0.0461210	6.962	3.46e-12	***
## StateCT	0.0248723	0.0396204	0.628	0.530166	
## StateDC	-0.1063095	0.0658876	-1.613	0.106652	
## StateDE	-0.0165674	0.0681561	-0.243	0.807946	
## StateFL	-0.1292317	0.0270397	-4.779	1.77e-06	***
## StateGA	0.1993725	0.0344507	5.787	7.26e-09	***
## StateIA	0.1951767	0.0605729	3.222	0.001274	**
## StateID	-0.0392758	0.0826562	-0.475	0.634670	
## StateIL	0.2119052	0.0338397	6.262	3.88e-10	***
## StateIN	0.2326870	0.0439717	5.292	1.22e-07	***
## StateKS	-0.0439919	0.0639898	-0.687	0.491786	
## StateKY	0.1549229	0.0540908	2.864	0.004186	**
## StateLA	-0.1191245	0.0328719	-3.624	0.000291	***
## StateMA	0.0546571	0.0477450	1.145	0.252318	
## StateMD	0.0246739	0.0365204	0.676	0.499289	
## StateME	-0.1049074	0.1482851	-0.707	0.479282	
## StateMI	0.0242993	0.0339721	0.715	0.474449	
## StateMN	-0.1037448	0.0323346	-3.208	0.001336	**
## StateMO	0.1907787	0.0404981	4.711	2.48e-06	***
## StateMS	-0.0558715	0.0772944	-0.723	0.469787	
## StateMT	-0.0827102	0.0556104	-1.487	0.136947	
## StateNC	-0.0364693	0.0294316	-1.239	0.215315	
## StateND	-0.1796208	0.1411770	-1.272	0.203278	
## StateNE	-0.0070656	0.0853331	-0.083	0.934011	
## StateNH	-0.0893140	0.1160126	-0.770	0.441389	
## StateNJ	-0.0187534	0.0336919	-0.557	0.577797	
## StateNM	0.2998625	0.0824248	3.638	0.000275	***
## StateNV	-0.0262233	0.0681543	-0.385	0.700417	
## StateNY	-0.0275833	0.0292835	-0.942	0.346234	
## StateOH	0.0593809	0.0364860	1.627	0.103648	
## StateOK	-0.1073748	0.0414171	-2.593	0.009534	**
## StateOR	-0.1095176	0.0323535	-3.385	0.000713	***
## StatePA	0.0369648	0.0296500	1.247	0.212521	
## StateRI	0.2413933	0.0670878	3.598	0.000321	***
## StateSC	-0.1305450	0.0286541	-4.556	5.25e-06	***
## StateSD	-0.2075373	0.3108463	-0.668	0.504363	
## StateTN	-0.0665447	0.0316899	-2.100	0.035753	*

```
## StateTX          0.0093323  0.0281660  0.331 0.740398
## StateUT          0.0241681  0.0421779  0.573 0.566648
## StateVA          0.0061746  0.0294629  0.210 0.834002
## StateVT          0.1167497  0.2541519  0.459 0.645974
## StateWA          0.1087387  0.0374923  2.900 0.003732 **
## StateWI          0.2031722  0.0500819  4.057 4.99e-05 ***
## StateWV          -0.1312204  0.0723243 -1.814 0.069641 .
## StateWY          0.1549013  0.1430400  1.083 0.278856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4379 on 19943 degrees of freedom
## Multiple R-squared:  0.06336,    Adjusted R-squared:  0.06035
## F-statistic: 21.08 on 64 and 19943 DF,  p-value: < 2.2e-16
```

```
vif(model_mlr_full)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## Distance      1.052512  1      1.025920
## Temperature    1.713777  1      1.309113
## Humidity        1.915107  1      1.383874
## Visibility      1.838687  1      1.355982
## Wind_Speed     1.199920  1      1.095409
## Pressure       3.010455  1      1.735066
## Precipitation  1.043549  1      1.021542
## Weather_Simple  2.568294  4      1.125137
## Rush_Hour      1.133390  1      1.064608
## Weekend        1.028167  1      1.013986
## Is_Daytime     1.356326  1      1.164614
## Traffic_Signal_Flag 1.144877  1      1.069989
## Road_Features  1.133217  1      1.064526
## State         5.551919 48      1.018016
```

```
# Plot 1: Bar plot of model coefficients (sorted by magnitude)
```

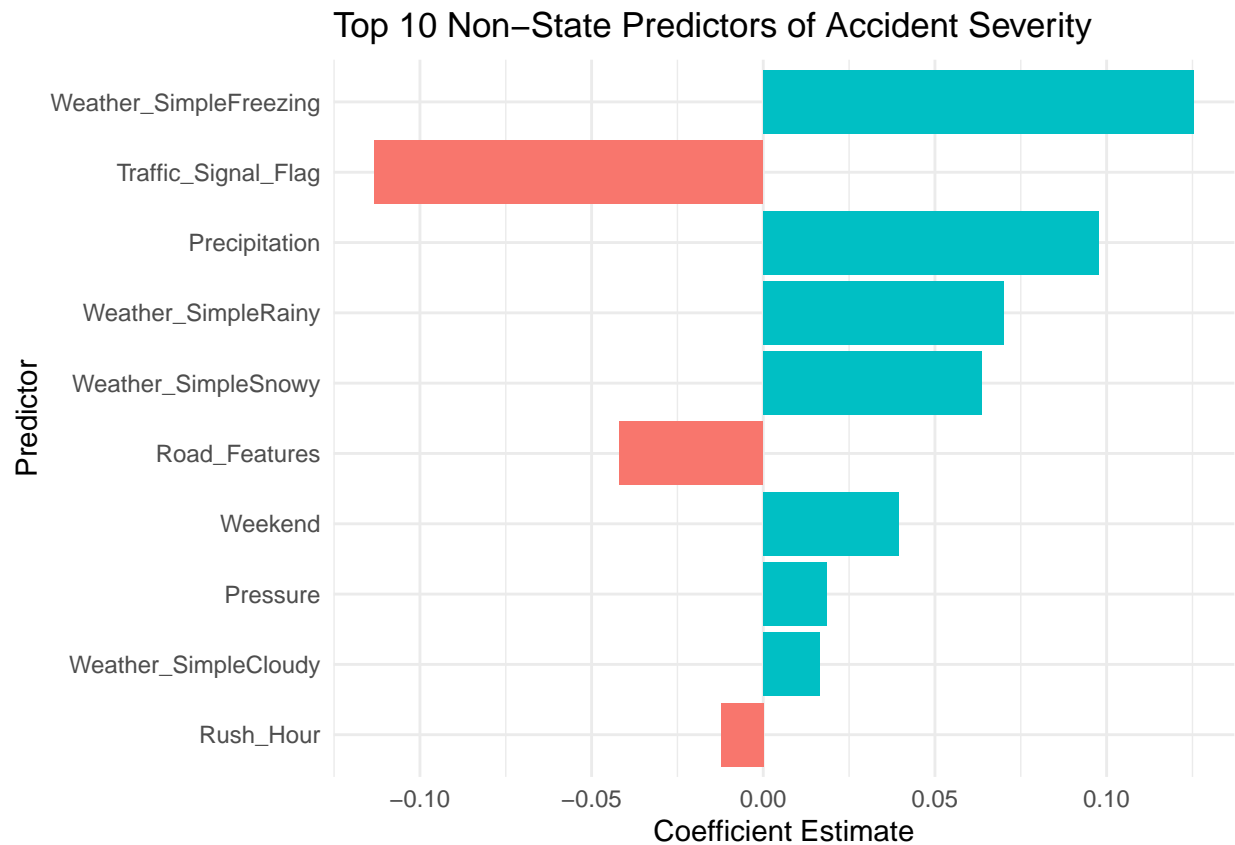
```
library(broom)
library(forcats)
library(dplyr)
library(ggplot2)
```

```
# Get top 10 predictors by effect size
```

```
library(broom)
library(forcats)
```

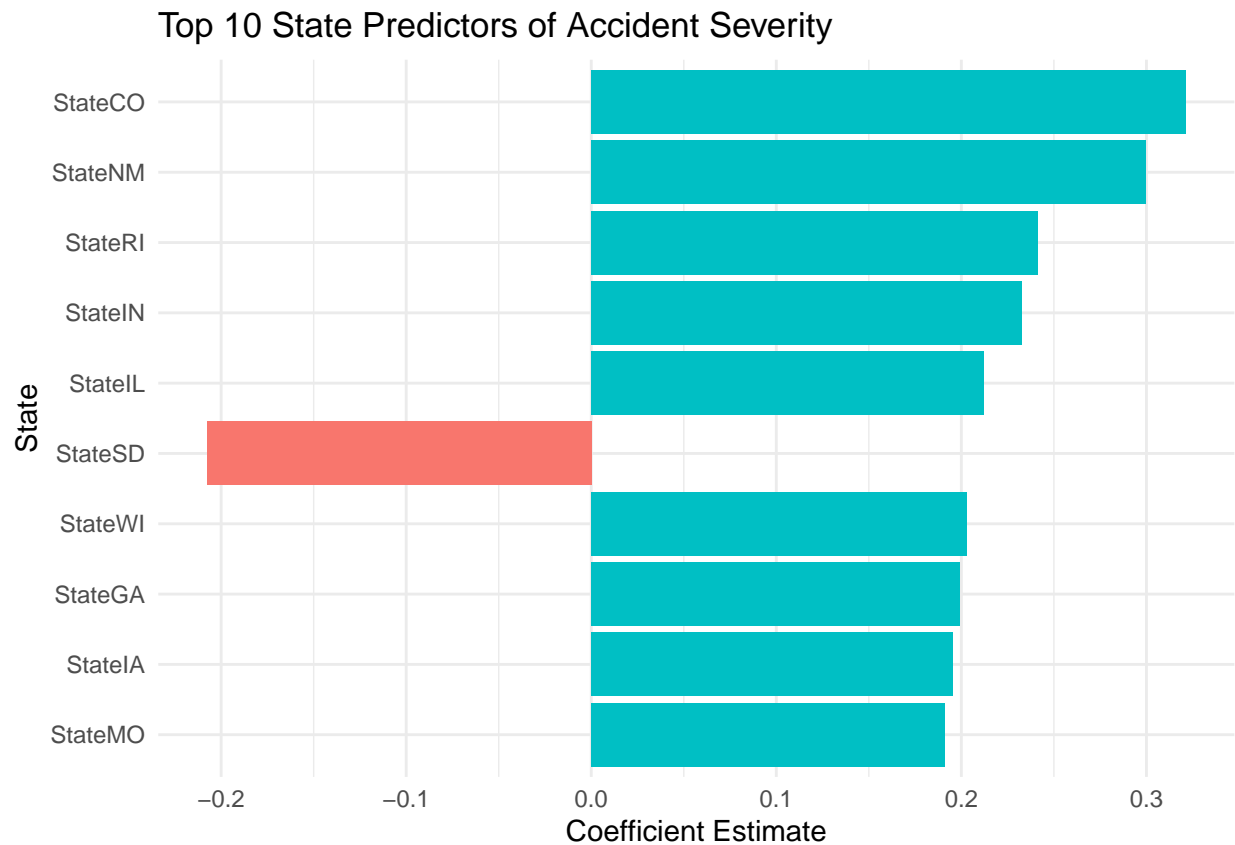
```
non_state_coefs <- tidy(model_mlr_full) %>%
  filter(!grepl("^State", term) & term != "(Intercept)") %>%
  mutate(term = fct_reorder(term, abs(estimate))) %>%
  slice_max(abs(estimate), n = 10)

ggplot(non_state_coefs, aes(x = term, y = estimate, fill = estimate > 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(title = "Top 10 Non-State Predictors of Accident Severity",
       x = "Predictor", y = "Coefficient Estimate") +
  theme_minimal()
```



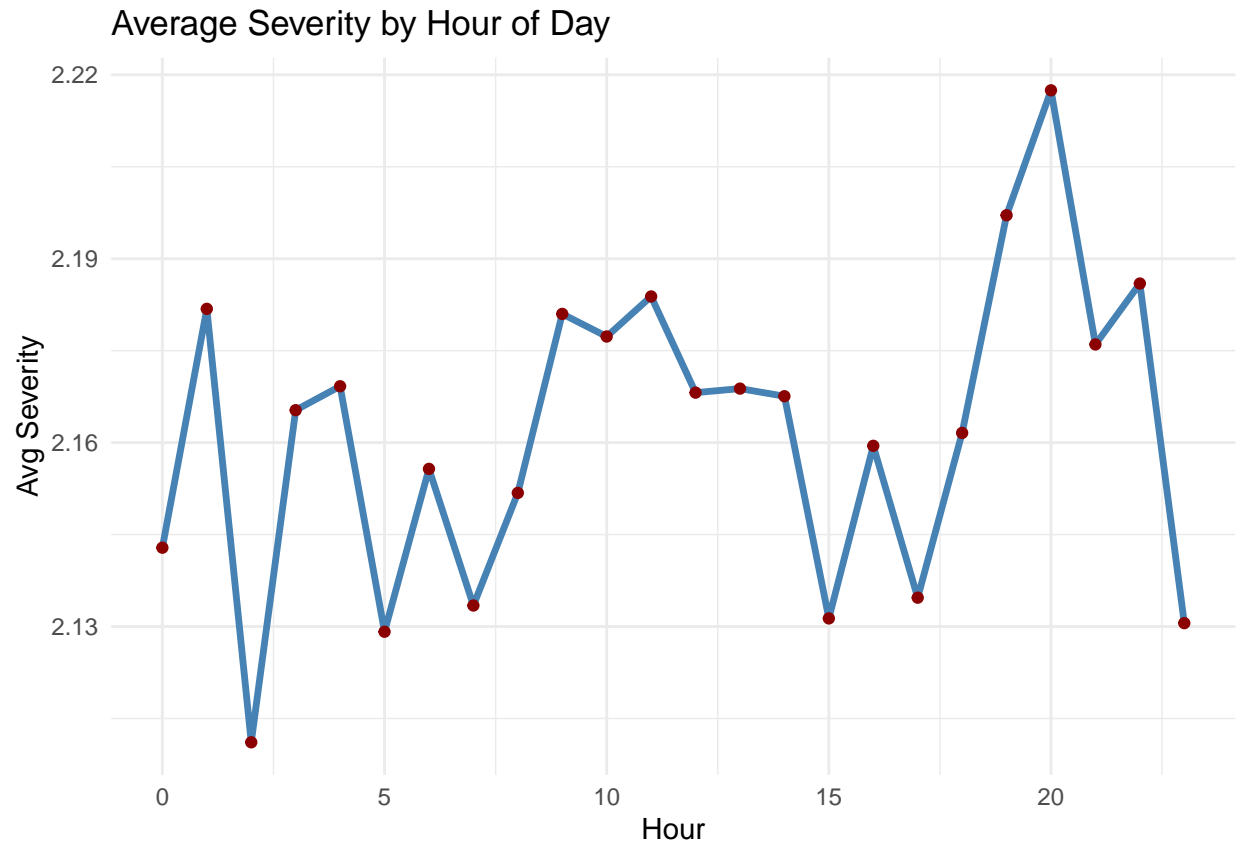
```
state_coefs <- tidy(model_mlr_full) %>%
  filter(grepl("^State", term)) %>%
  mutate(term = fct_reorder(term, abs(estimate))) %>%
  slice_max(abs(estimate), n = 10)

ggplot(state_coefs, aes(x = term, y = estimate, fill = estimate > 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(title = "Top 10 State Predictors of Accident Severity",
       x = "State", y = "Coefficient Estimate") +
  theme_minimal()
```



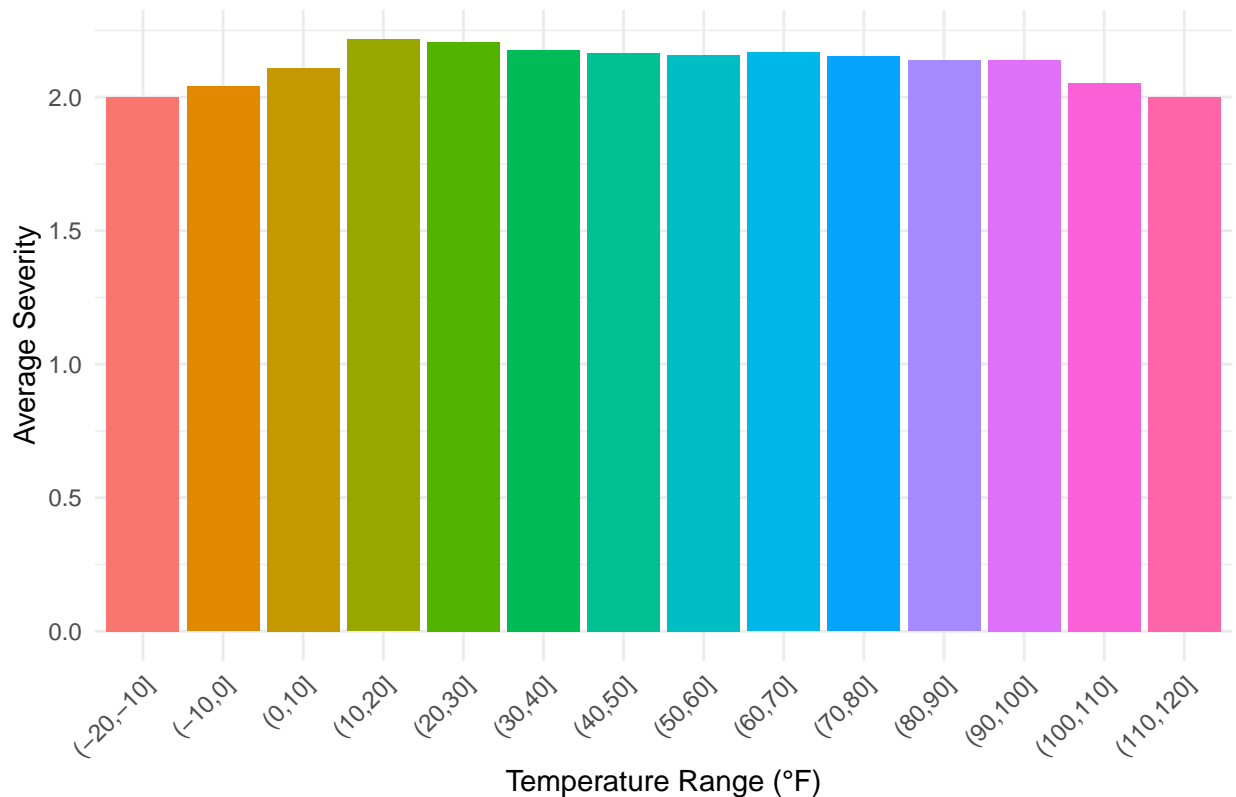
```
# Plot 2: Trend of average severity by hour
accident_data %>%
  group_by(Start_Hour) %>%
  summarise(avg_severity = mean(Severity)) %>%
  ggplot(aes(x = Start_Hour, y = avg_severity)) +
  geom_line(color = "steelblue", size = 1.2) +
  geom_point(color = "darkred") +
  labs(title = "Average Severity by Hour of Day", x = "Hour", y = "Avg Severity") +
  theme_minimal()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



```
# Plot 3: Create temperature bins
accident_data %>%
  mutate(temp_bin = cut(Temperature, breaks = seq(-20, 120, by = 10))) %>%
  group_by(temp_bin) %>%
  summarise(avg_severity = mean(Severity)) %>%
  ggplot(aes(x = temp_bin, y = avg_severity, fill = temp_bin)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Average Severity by Temperature Range",
       x = "Temperature Range (°F)", y = "Average Severity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Average Severity by Temperature Range



```
# -----
# Step 8B: Fit reduced model using only significant predictors and states
significant_states <- c("AZ", "CA", "CO", "FL", "GA", "IA", "IL", "IN", "KY", "LA", "MA", "ME", "MI", "MN", "MO", "NE", "NH", "NJ", "NY", "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN", "TX", "UT", "VA", "VT", "WA", "WI", "WY")

accident_data_reduced <- accident_data %>%
  filter(State %in% significant_states & Weather_Simple %in% c("Clear", "Cloudy", "Freezing", "Rainy"))
  mutate(
    Weather_Simple = factor(Weather_Simple),
    State = factor(State)
  )

model_mlr_reduced <- lm(Severity ~ Distance + Temperature + Humidity + Wind_Speed +
  Weather_Simple + Weekend + Traffic_Signal_Flag + Road_Features + State,
  data = accident_data_reduced)

summary(model_mlr_reduced)

##
## Call:
## lm(formula = Severity ~ Distance + Temperature + Humidity + Wind_Speed +
##     Weather_Simple + Weekend + Traffic_Signal_Flag + Road_Features +
##     State, data = accident_data_reduced)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46909 -0.13608 -0.09560 -0.04699  2.09108
```

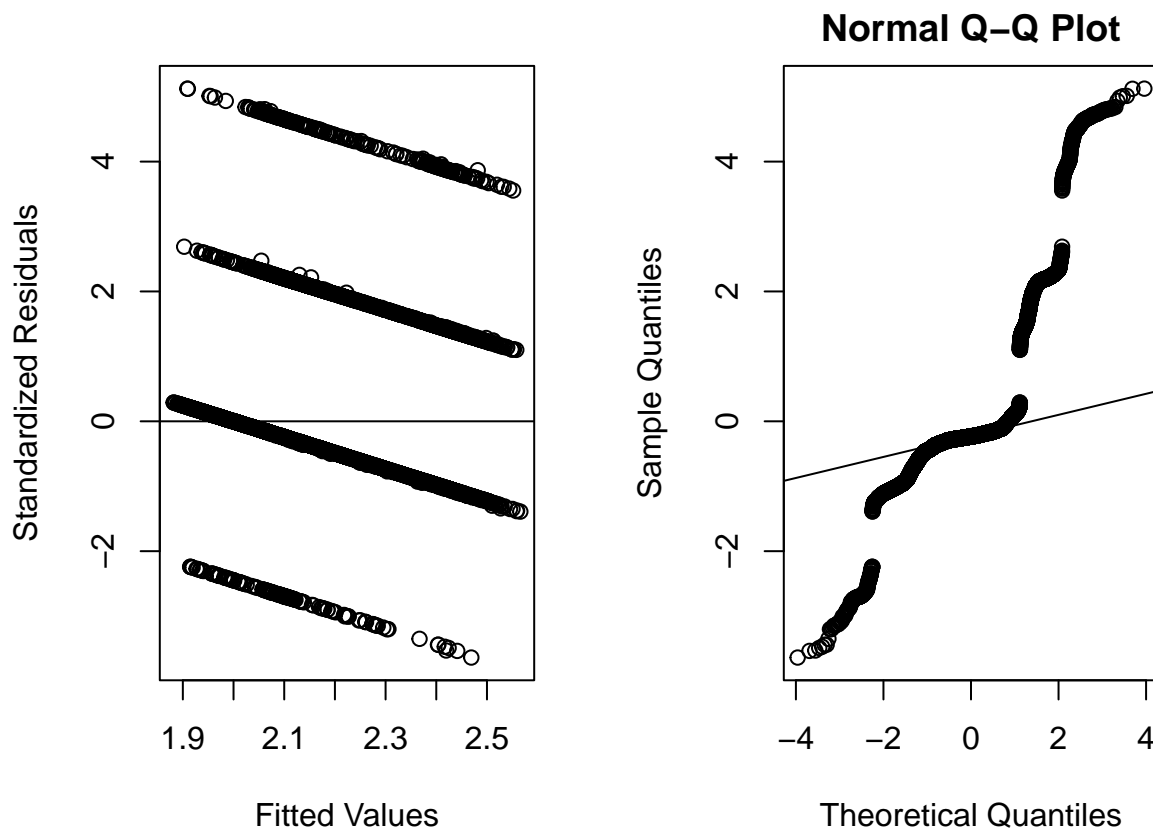
```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.9660903   0.0297308  66.130 < 2e-16 ***
## Distance         -0.0031206   0.0026210  -1.191 0.233822
## Temperature       0.0011373   0.0002505   4.540 5.68e-06 ***
## Humidity          0.0004753   0.0002080   2.286 0.022294 *
## Wind_Speed        0.0001377   0.0007332   0.188 0.851003
## Weather_SimpleCloudy 0.0090643   0.0081249   1.116 0.264603
## Weather_SimpleFreezing 0.1208959   0.1103757   1.095 0.273399
## Weather_SimpleRainy 0.0786488   0.0147898   5.318 1.07e-07 ***
## Weekend           0.0376573   0.0092539   4.069 4.74e-05 ***
## Traffic_Signal_Flag -0.1089508   0.0113773  -9.576 < 2e-16 ***
## Road_Features     -0.0393543   0.0090081  -4.369 1.26e-05 ***
## StateCA           0.0279006   0.0211715   1.318 0.187580
## StateCO           0.3702451   0.0341941  10.828 < 2e-16 ***
## StateFL           0.0146901   0.0227501   0.646 0.518474
## StateGA           0.3315951   0.0301842  10.986 < 2e-16 ***
## StateIA           0.3494500   0.0569610   6.135 8.76e-10 ***
## StateIL           0.3486123   0.0299254  11.649 < 2e-16 ***
## StateIN           0.3549783   0.0405513   8.754 < 2e-16 ***
## StateKY           0.2894248   0.0494526   5.853 4.95e-09 ***
## StateLA           0.0257535   0.0287378   0.896 0.370188
## StateMA           0.2094755   0.0439204   4.769 1.87e-06 ***
## StateME           0.0631046   0.1460898   0.432 0.665779
## StateMI           0.1395208   0.0311093   4.485 7.36e-06 ***
## StateMN           0.0424529   0.0289212   1.468 0.142160
## StateMO           0.3225876   0.0366880   8.793 < 2e-16 ***
## StateMT           0.0164094   0.0543041   0.302 0.762522
## StateNC           0.0953699   0.0247546   3.853 0.000117 ***
## StateNH           0.0964838   0.1154056   0.836 0.403147
## StateNM           0.3646688   0.0749817   4.863 1.17e-06 ***
## StateOH           0.1875112   0.0328704   5.705 1.19e-08 ***
## StateOK           0.0117528   0.0367951   0.319 0.749419
## StateOR           0.0354479   0.0281422   1.260 0.207835
## StateRI           0.3924769   0.0658177   5.963 2.54e-09 ***
## StateSC           0.0036847   0.0242074   0.152 0.879021
## StateWA           0.2502880   0.0333549   7.504 6.59e-14 ***
## StateWI           0.3794455   0.0471177   8.053 8.75e-16 ***
## StateWV           0.0079154   0.0687329   0.115 0.908318
## StateWY           0.0705615   0.1461711   0.483 0.629294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4087 on 13294 degrees of freedom
## Multiple R-squared:  0.08029,    Adjusted R-squared:  0.07773
## F-statistic: 31.37 on 37 and 13294 DF,  p-value: < 2.2e-16
```

```
vif(model_mlr_reduced)
```

```
##               GVIF Df GVIF^(1/(2*Df))
## Distance      1.040726 1      1.020160
## Temperature    1.592593 1      1.261980
## Humidity       1.847477 1      1.359219
```

```
## Wind_Speed      1.198307  1      1.094672
## Weather_Simple  1.419640  3      1.060140
## Weekend         1.016709  1      1.008320
## Traffic_Signal_Flag 1.155687  1      1.075029
## Road_Features   1.143975  1      1.069567
## State           2.046410 27      1.013349
```

```
# -----
# Step 9: Diagnostics for reduced model
par(mfrow = c(1,2), mar = c(4.5,4.5,2,2))
plot(predict(model_mlr_reduced), rstandard(model_mlr_reduced),
     xlab = "Fitted Values", ylab = "Standardized Residuals")
abline(h = 0)
qqnorm(rstandard(model_mlr_reduced))
qqline(rstandard(model_mlr_reduced))
```



```
# -----
# Step 10: Random Forest preparation
rf_data <- accident_data %>% mutate(Severity = as.factor(Severity))

# -----
# Step 11: Fit Random Forest model
set.seed(123)
rf_model <- randomForest(
  Severity ~ Distance + Temperature + Humidity + Visibility +
```

```

    Wind_Speed + Pressure + Precipitation + Weather_Simple +
    Rush_Hour + Weekend + Is_Daytime + Traffic_Signal_Flag + Road_Features + State,
data = rf_data,
ntree = 500,
importance = TRUE
)

print(rf_model)

```

```

##
## Call:
## randomForest(formula = Severity ~ Distance + Temperature + Humidity +      Visibility + Wind_Speed +
##               Type of random forest: classification
##               Number of trees: 500
## No. of variables tried at each split: 3
##
##           OOB estimate of  error rate: 14.65%
## Confusion matrix:
##   1      2      3 4 class.error
## 1 0      208      5 0  1.00000000
## 2 0 16596 270 3   0.01618353
## 3 0  1962 479 0   0.80376895
## 4 0   481   2 2   0.99587629

```

```

conf_matrix <- rf_model$confusion
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Overall Accuracy:", round(accuracy * 100, 2), "%"))

```

```

## [1] "Overall Accuracy: 85.34 %"

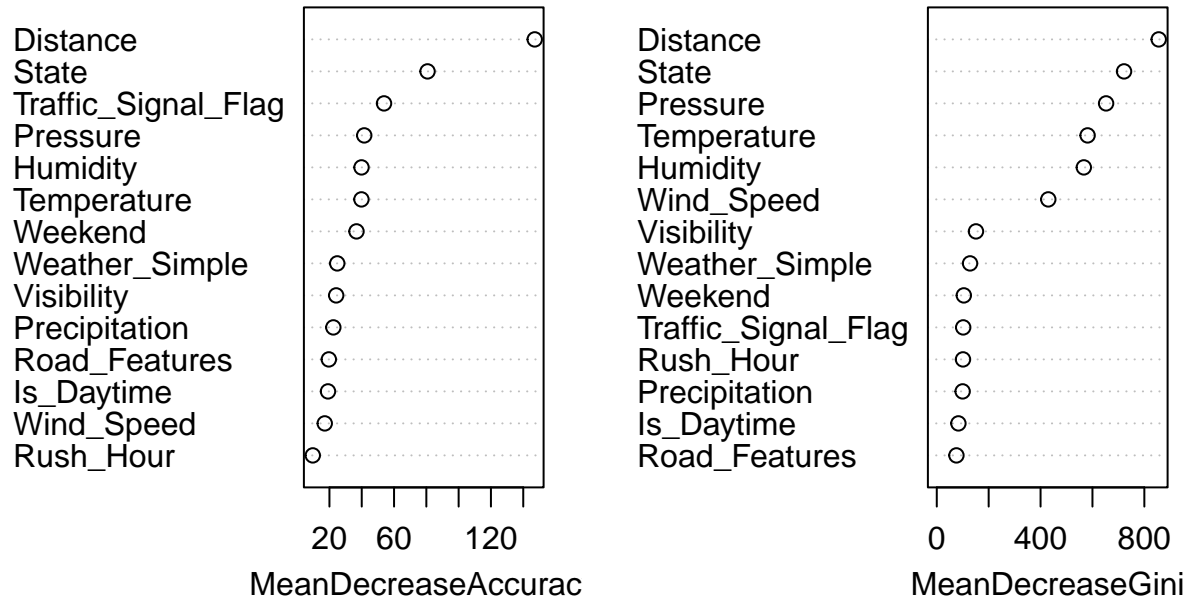
```

```

# -----
# Step 12: Variable importance from Random Forest
varImpPlot(rf_model, main = "Variable Importance (Random Forest)")

```

Variable Importance (Random Forest)



```
importance(rf_model)
```

```
##              1              2              3              4
## Distance      29.97529626 114.735606 140.034386457 33.2769203
## Temperature    10.65736469  39.483822   1.330601033 -4.3229044
## Humidity        4.53014272  37.255512   2.665205855 -3.1217376
## Visibility      0.05109941  22.133430   6.982706424 -5.1341221
## Wind_Speed     -0.17739367  13.646346   9.996835278 -0.8427502
## Pressure        6.62247648  40.118107   9.282228286  0.4032321
## Precipitation   1.32351587  19.447365   9.429188124 -3.5402430
## Weather_Simple  1.49073499  20.742266  10.329174017 -2.3944112
## Rush_Hour       4.73898938  11.528393  -2.630383088  0.8010128
## Weekend         3.17650078  37.796972   8.387639382  0.3381352
## Is_Daytime      1.68026817  18.967037  -0.001800464  1.9187038
## Traffic_Signal_Flag 13.66553986  46.878537  34.089764638  2.4439861
## Road_Features    5.31989213   9.175162  23.730107051  6.4589175
## State           6.63445128  68.543027  36.261972801 20.5727576
##
##              MeanDecreaseAccuracy MeanDecreaseGini
## Distance           147.107993         855.38300
## Temperature         39.806114         581.24600
## Humidity             39.861129         566.73947
## Visibility           24.177355         151.14723
## Wind_Speed           17.088288         429.75833
## Pressure             41.542530         652.59272
## Precipitation        22.413666          99.69144
```

## Weather_Simple	24.843148	128.30324
## Rush_Hour	9.673048	101.09305
## Weekend	36.754244	104.28470
## Is_Daytime	19.109202	83.40098
## Traffic_Signal_Flag	53.792748	101.88243
## Road_Features	19.624896	75.75111
## State	80.750599	721.73812

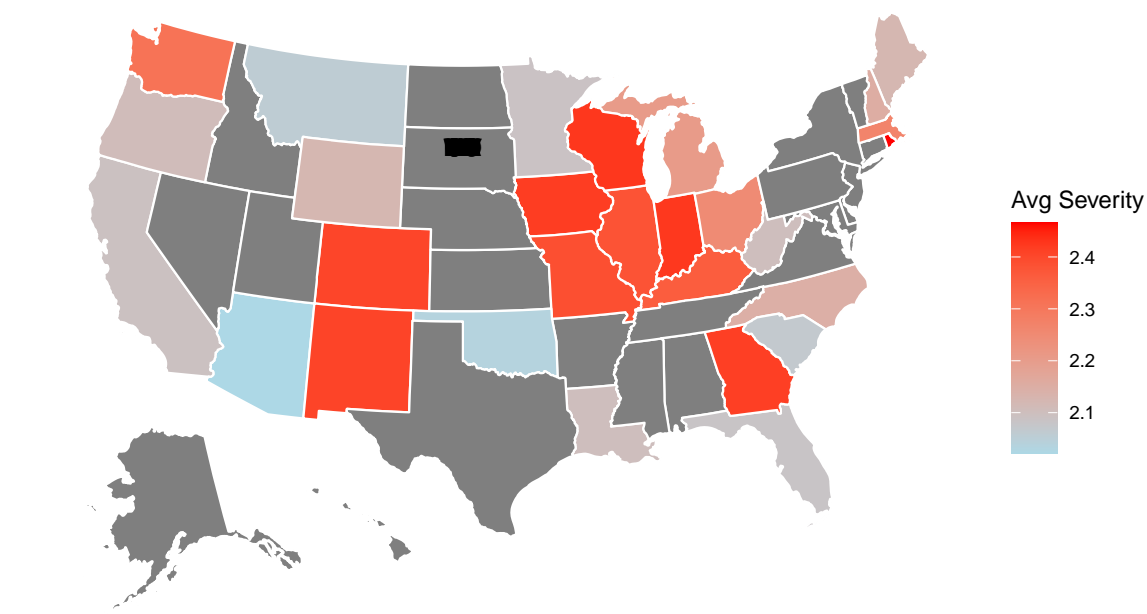
```
# -----
# Create summary of mean severity per state
state_avg <- accident_data_reduced %>%
  group_by(State) %>%
  summarise(mean_severity = mean(Severity)) %>%
  rename(state = State) # required for plot_usmap

# Built-in centroids for label positions
state_centroids <- data.frame(
  state = state.abb,
  x = state.center$x,
  y = state.center$y
)

# Keep only states that are in your data
state_labels <- state_centroids %>%
  filter(state %in% state_avg$state)

# Plot US map with severity and state labels
plot_usmap(data = state_avg, values = "mean_severity", color = "white") +
  geom_text(data = state_labels,
    aes(x = x, y = y, label = state),
    size = 3, color = "black") +
  scale_fill_continuous(name = "Avg Severity", low = "lightblue", high = "red") +
  labs(title = "Average Accident Severity by State") +
  theme(legend.position = "right")
```

Average Accident Severity by State



```
# -----  
# Step 14: Confusion matrix heatmap (Random Forest)  
cm <- as.data.frame(rf_model$confusion)  
cm$Actual <- rownames(cm)  
cm_long <- melt(cm, id.vars = "Actual", variable.name = "Predicted", value.name = "Count")  
  
ggplot(cm_long, aes(x = Predicted, y = Actual, fill = Count)) +  
  geom_tile() +  
  geom_text(aes(label = Count), color = "white") +  
  scale_fill_gradient(low = "pink", high = "red") +  
  labs(title = "Confusion Matrix Heatmap", x = "Predicted", y = "Actual") +  
  theme_minimal()
```

