Project

Manasa

2024-03-17

R Markdown

```
if (!requireNamespace("readr", quietly = TRUE)) install.packages("readr")
if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dplyr")
if (!requireNamespace("tidyr", quietly = TRUE)) install.packages("tidyr")
if (!requireNamespace("lubridate", quietly = TRUE)) install.packages("lubridate")

#Load the libraries
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(readr)
library(ggplot2)
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(tidyr)

crd <- read_csv("/Users/manasavishnumoorthy/Desktop/IDMP/CRD.csv",
                na = c("", "NA", " "), show_col_types = FALSE)

incident <- read_csv("/Users/manasavishnumoorthy/Desktop/IDMP/incidents_data.csv",
```

```
                na = c("", "NA", " "), show_col_types = FALSE)

non_motorists <- read_csv("/Users/manasavishnumoorthy/Desktop/IDMP/non_motorist_data.csv",
                na = c("", "NA", " "), show_col_types = FALSE)

# Convert 'Local Case Number' to character in both data frames
crd$`Local Case Number` <- as.character(crd$`Local Case Number`)
non_motorists$`Local Case Number` <- as.character(non_motorists$`Local Case Number`)
incident$`Local Case Number` <- as.character(incident$`Local Case Number`)

combined_data <- crd %>%
  full_join(non_motorists, by = c("Report Number", "Local Case Number")) %>%
  full_join(incident, by = c("Report Number", "Local Case Number"))

# Check the first few rows of the combined data
#str(combined_data)
```

Data preprocessing Standardize text variables: Convert all character variables to lowercase for consistency.

```
combined_data <- combined_data %>%
  mutate(across(where(is.character), tolower))
```

Handle missing values: Replace missing values in character columns with "unknown" and in numeric columns with 0 (or another appropriate value).

```
combined_data <- combined_data %>%
  mutate(across(where(is.character), ~replace_na(., "unknown"))) %>%
  mutate(across(where(is.numeric), ~replace_na(., 0)))
```

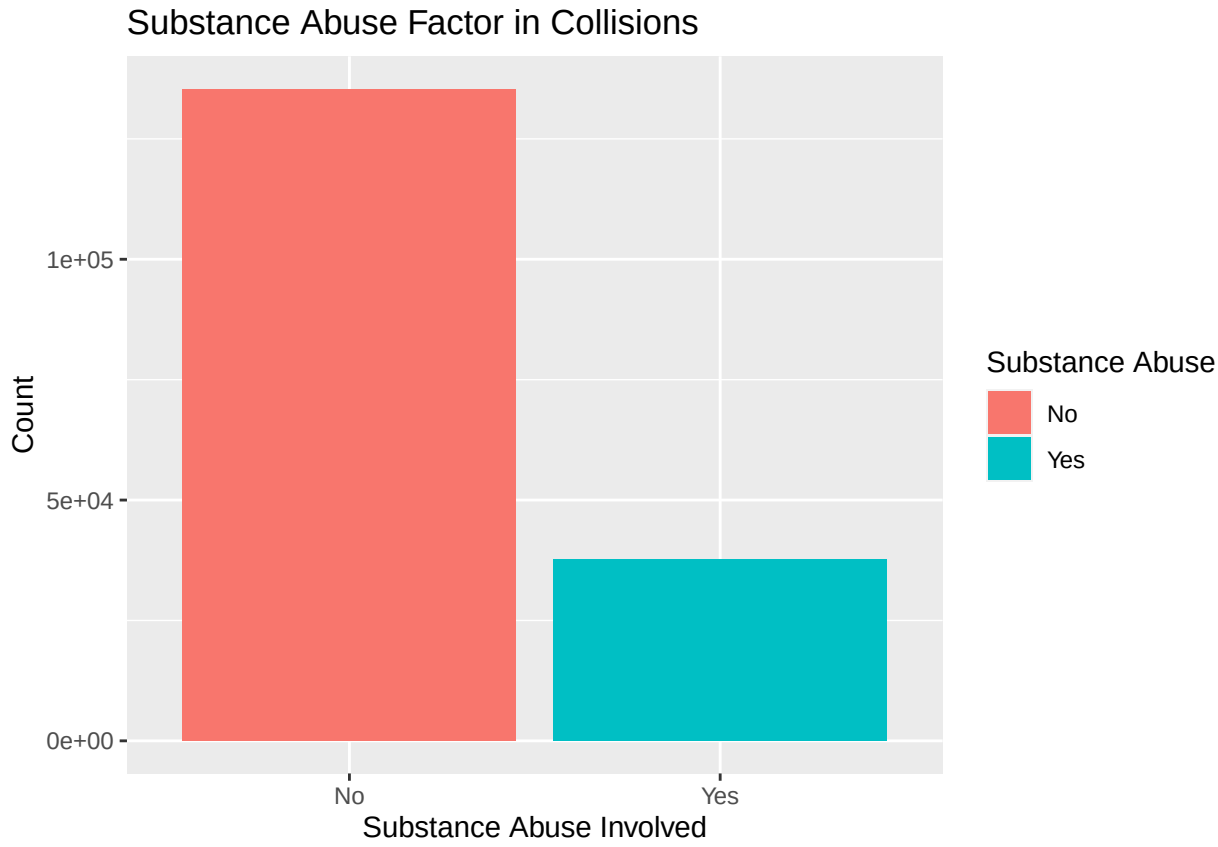
Create new variables from 'Crash Date/Time': Parse the 'Crash Date/Time' and extract useful components such as the date, hour, day of the week, etc.

```
# This assumes that the date/time column is named 'Crash Date/Time.x' in your combined_data
combined_data <- combined_data %>%
  mutate(`Crash Date/Time` = mdy_hms(`Crash Date/Time.x`))  # Convert to date-time object

# Extract components
combined_data <- combined_data %>%
  mutate(
    Date = as.Date(`Crash Date/Time`),  # Extract date
    Hour = hour(`Crash Date/Time`),  # Extract hour
    Weekday = wday(`Crash Date/Time`, label = TRUE),  # Extract day of the week
    Month = month(`Crash Date/Time`, label = TRUE)  # Extract month
  )
```

```
#str(combined_data)
```

Substance Abuse Factor: Investigating the factor of substance abuse in collisions.

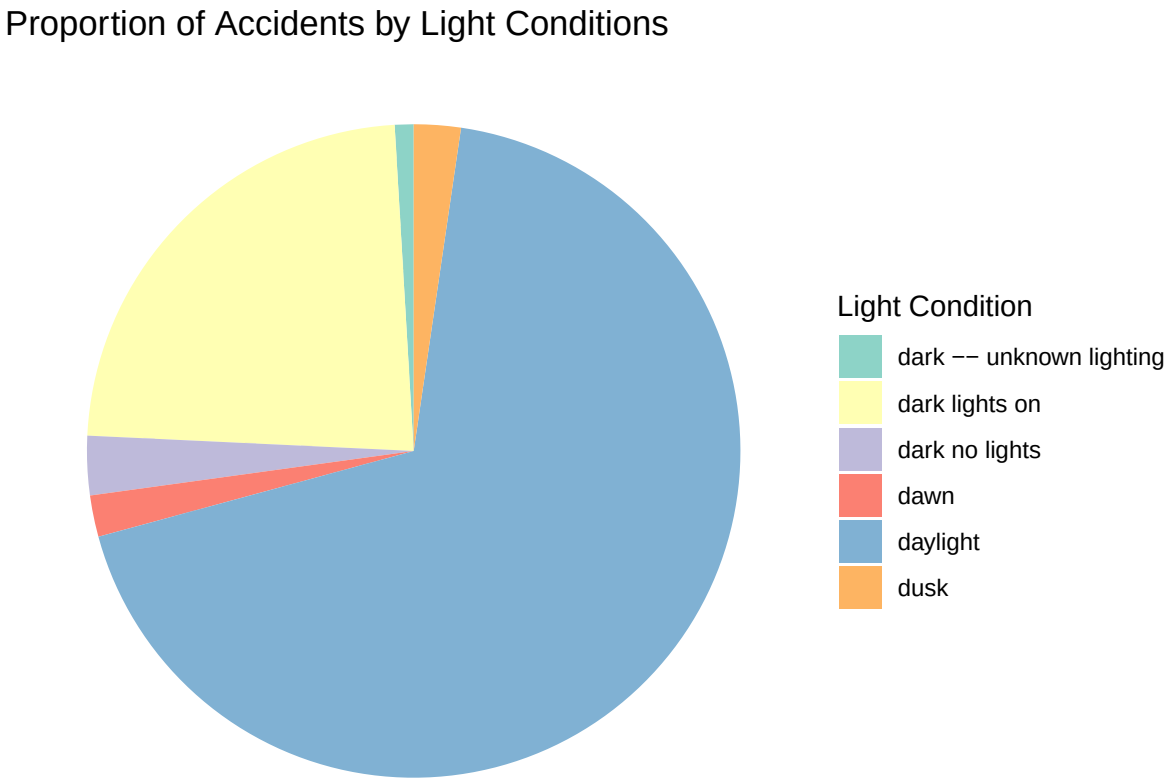
```r
combined_data$`Substance Abuse` <- ifelse(combined_data$`Driver Substance Abuse.x` != "none detected" &
                                          combined_data$`Driver Substance Abuse.x` != "unknown", "Yes",

ggplot(combined_data, aes(x = `Substance Abuse`, fill = `Substance Abuse`)) +
  geom_bar() +
  labs(title = "Substance Abuse Factor in Collisions", x = "Substance Abuse Involved", y = "Count")
```



Accidents by Light Conditions and Time of Day: It might be interesting to see how light conditions impact accidents, and if there are peak times.

```r
library(ggplot2)
library(dplyr)

# Filter out unwanted 'Light.x' categories
filtered_data <- combined_data %>%
  filter(!Light.x %in% c("n/a", "other", "unknown"))

# Count the number of accidents by light condition
light_data <- filtered_data %>%
  count(Light.x) %>%
  rename(Number_of_Accidents = n)

# Create a pie chart
ggplot(light_data, aes(x = "", y = Number_of_Accidents, fill = Light.x)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +  # This makes it a pie chart
```

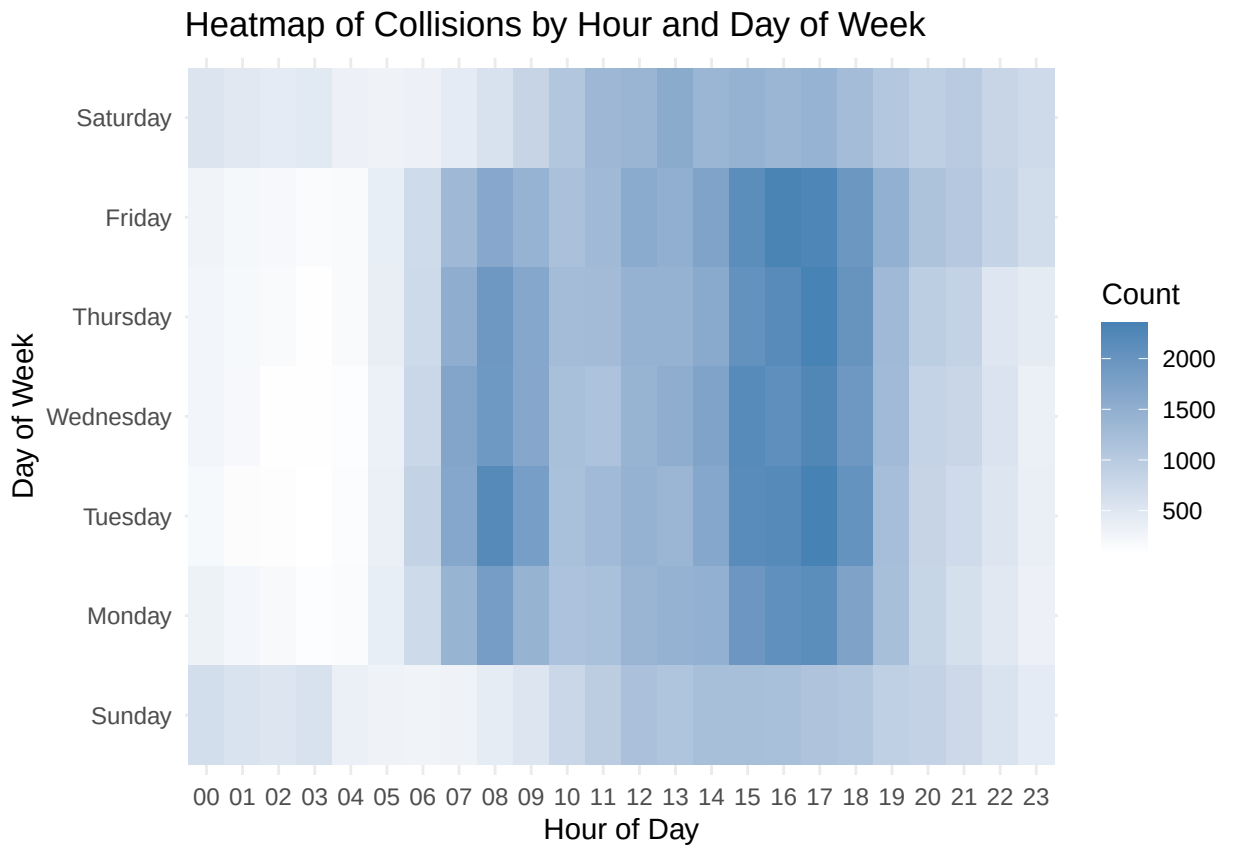
```
  scale_fill_brewer(palette = "Set3") +
  labs(title = "Proportion of Accidents by Light Conditions", x = "", y = "", fill = "Light Condition") +
  theme_void()  # This removes the background, gridlines, and text
```



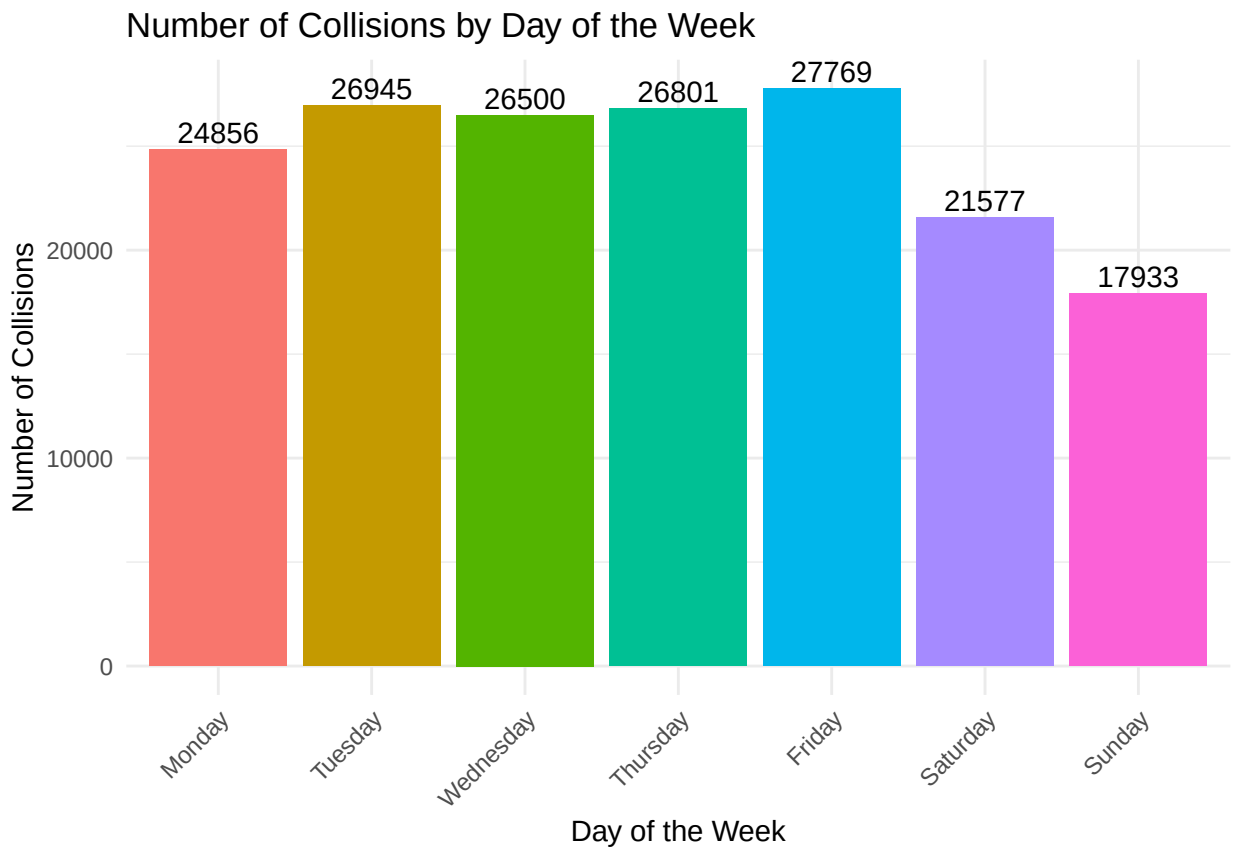
Heatmap of Collisions by Hour and Day of the Week This visualization will help us understand when most collisions occur during the week.

```
library(ggplot2)
library(dplyr)

# Create a column for the day of the week and hour
combined_data$DayOfWeek <- weekdays(as.Date(combined_data$`Crash Date/Time`))
combined_data$Hour <- format(combined_data$`Crash Date/Time`, "%H")

# Prepare the data for the heatmap
heatmap_data <- combined_data %>%
  filter(!is.na(DayOfWeek) & DayOfWeek != "na") %>%  # Filter out the "NA" or "na" entries
  group_by(DayOfWeek, Hour) %>%
  summarize(Count = n(), .groups = 'drop') %>%
  mutate(DayOfWeek = factor(DayOfWeek, levels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday

# Create the heatmap
ggplot(heatmap_data, aes(x = Hour, y = DayOfWeek, fill = Count)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "steelblue") +
  labs(title = "Heatmap of Collisions by Hour and Day of Week", x = "Hour of Day", y = "Day of Week") +
```

```
  theme_minimal()
```




```
library(ggplot2)
library(dplyr)

# Assuming the 'DayOfWeek' variable is already created and 'NA' is one of the levels
collisions_by_day_filtered <- combined_data %>%
  filter(!is.na(DayOfWeek) & DayOfWeek != "na") %>%  # Filter out the "NA" or "na" entries
  group_by(DayOfWeek) %>%
  summarise(NumberOfCollisions = n(), .groups = 'drop') %>%
  mutate(DayOfWeek = factor(DayOfWeek, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday

# Now, create the bar chart without the "NA" category
ggplot(collisions_by_day_filtered, aes(x = DayOfWeek, y = NumberOfCollisions, fill = DayOfWeek)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  geom_text(aes(label = NumberOfCollisions), vjust = -0.3) +
  labs(title = "Number of Collisions by Day of the Week", x = "Day of the Week", y = "Number of Collisi
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



ARIMA

```
library(dplyr)
library(lubridate)
library(forecast)
```

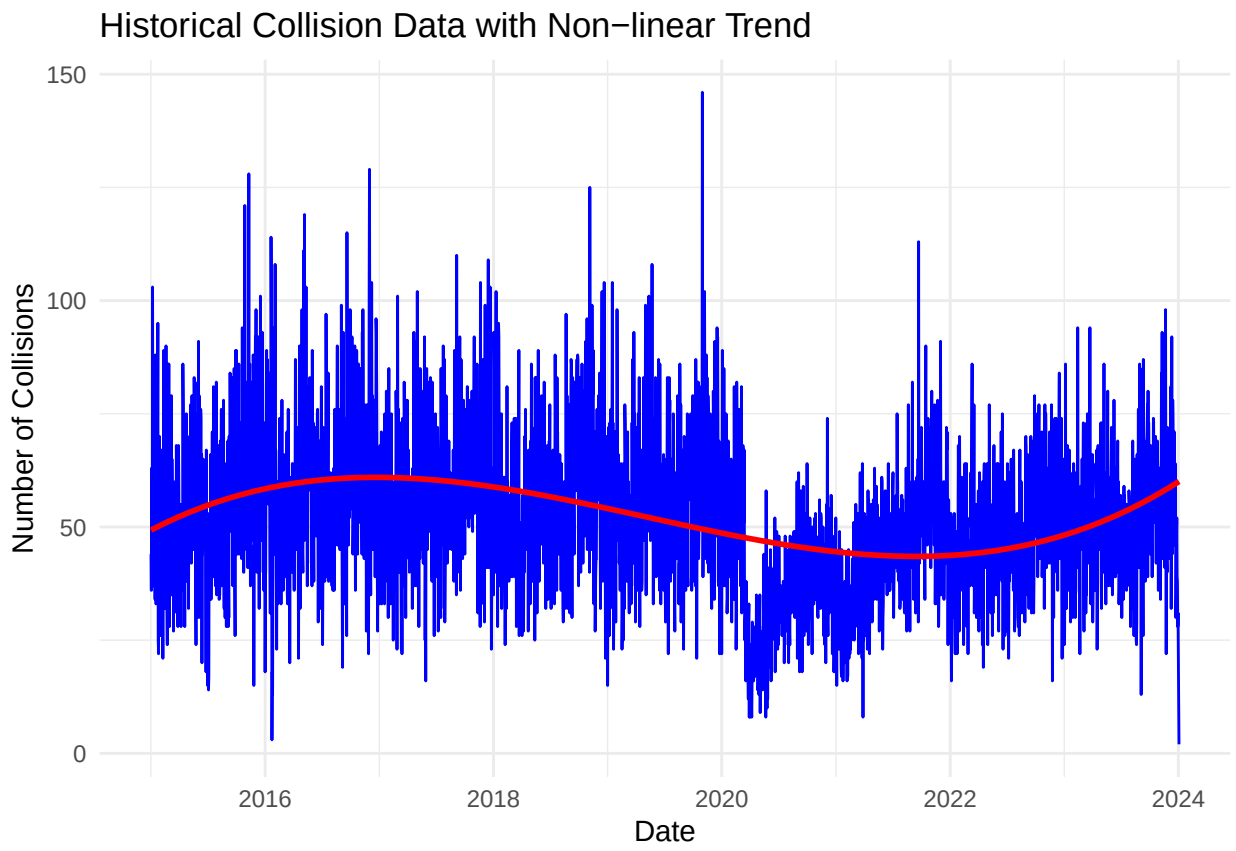
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tibble)
library(ggplot2)


# Convert the Crash Date/Time to a proper Date format (assuming it is in a recognisable format)
combined_data$Crash_Date <- as.Date(mdy_hms(combined_data$`Crash Date/Time.x`))

# Remove rows with NA in Crash_Date (if any conversion failed)
combined_data <- combined_data %>%
  filter(!is.na(Crash_Date))

# Aggregate data by Crash_Date to get the number of collisions per day
daily_collisions <- combined_data %>%
  group_by(Crash_Date) %>%
  summarise(Number_of_Collisions = n(), .groups = 'drop')
```

```r
# Generate the time series object from daily collision counts
collision_ts <- ts(daily_collisions$Number_of_Collisions, frequency = 365)

# Fit the ARIMA model
fit <- auto.arima(collision_ts)

# Check model summary
summary(fit)
```

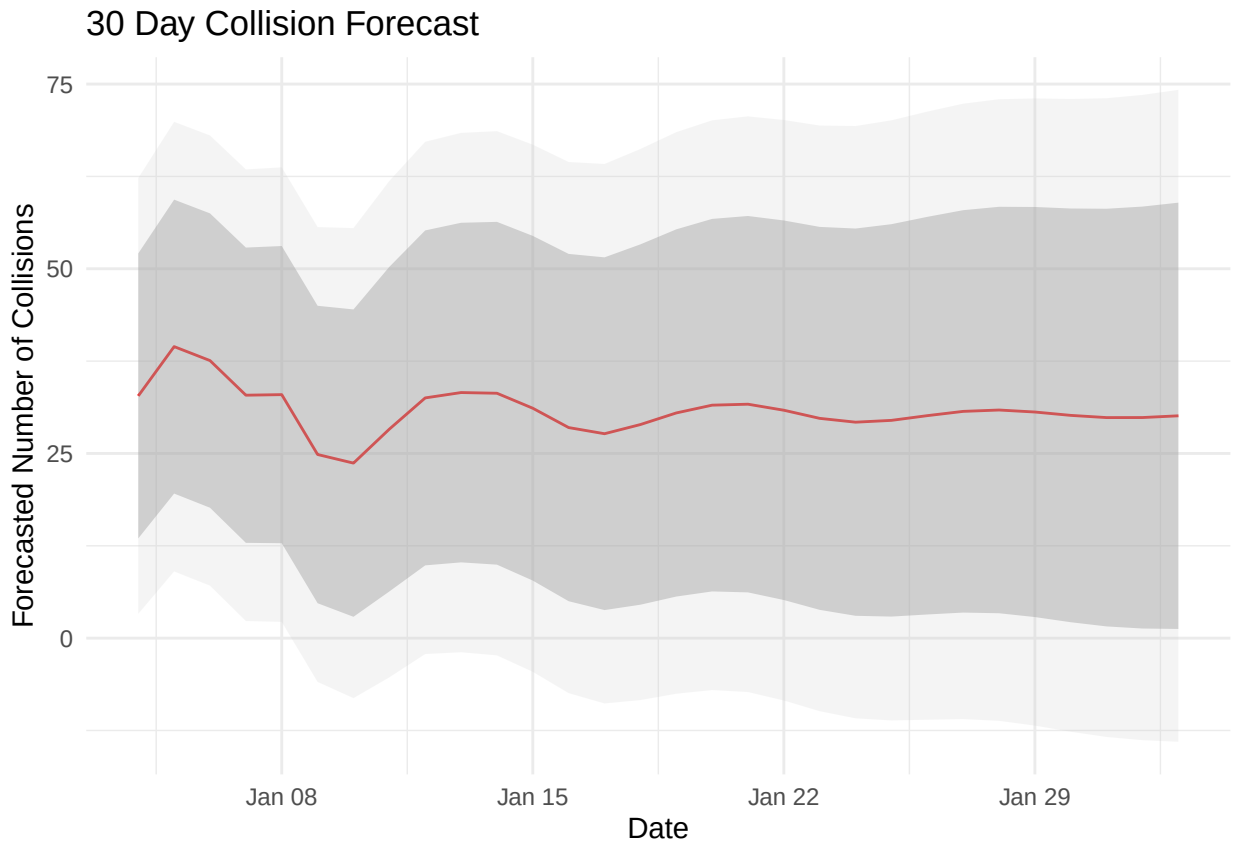
```
## Series: collision_ts
## ARIMA(5,1,2)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5      ma1     ma2
##       0.0974  -0.3863  -0.2462  -0.2249  -0.2886  -0.8447  0.2640
## s.e.  0.0610   0.0696   0.0349   0.0291   0.0324   0.0535  0.1214
##
## sigma^2 = 226.5:  log likelihood = -13573.89
## AIC=27163.78   AICc=27163.82   BIC=27212.56
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.02631048 15.03234 11.64846 -9.676269 26.68143 0.6615329
##                      ACF1
## Training set -0.01223579
```

```r
library(ggplot2)
library(dplyr)
library(lubridate)

# Assuming 'daily_collisions' is your data frame with the 'Crash_Date' and 'Number_of_Collisions' colum

# Ensure that the dates are in the proper Date format
daily_collisions$Crash_Date <- as.Date(daily_collisions$Crash_Date)

#Plot using ggplot2
#ggplot(daily_collisions, aes(x = Crash_Date, y = Number_of_Collisions)) +
# geom_line(color = "blue") +
#  geom_smooth(method = "lm", color = "red", se = FALSE) + # Trend line
#  labs(title = "Historical Collision Data",
#       x = "Date",
#       y = "Number of Collisions") +
#  theme_minimal()
#

ggplot(daily_collisions, aes(x = Crash_Date, y = Number_of_Collisions)) +
  geom_line(color = "blue") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 3), color = "red", se = FALSE) + # Quadratic trend l
  labs(title = "Historical Collision Data with Non-linear Trend",
       x = "Date",
       y = "Number of Collisions") +
  theme_minimal()
```

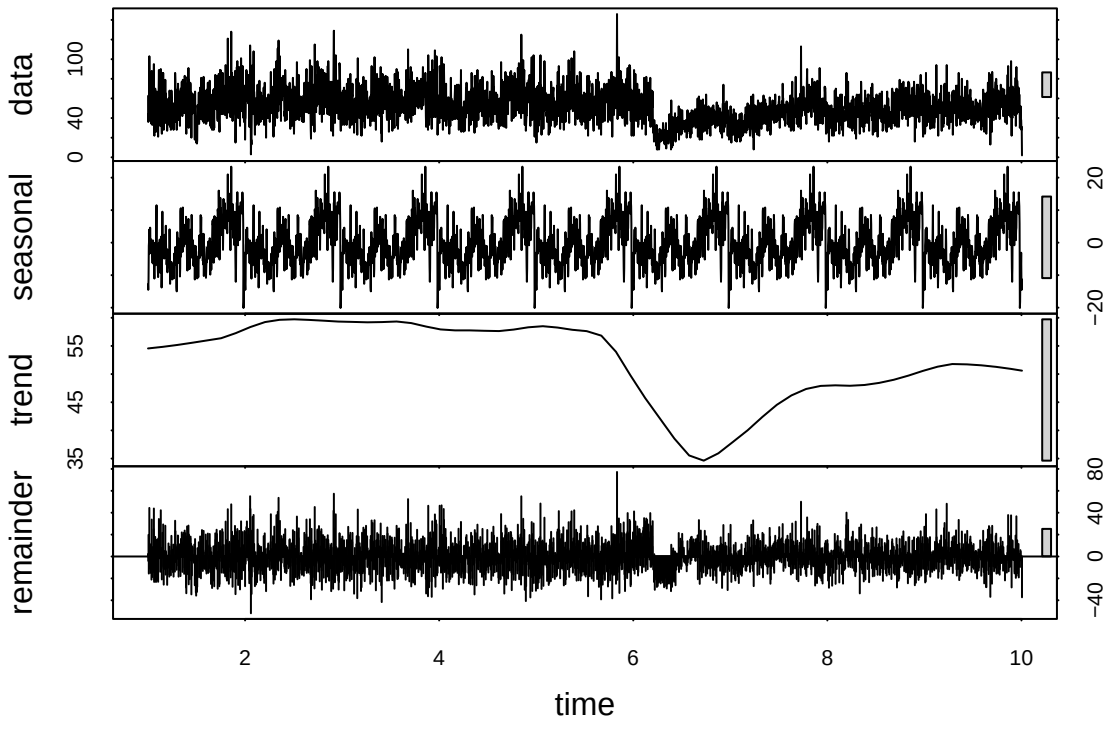



```r
library(forecast)
library(ggplot2)

# Assuming 'fit' is your previously fitted ARIMA model
# and 'daily_collisions' is the data frame with historical collision data

# Forecast the next 30 days using the fitted ARIMA model
forecast_future <- forecast(fit, h = 30)

# Create a data frame for the forecast dates and predictions
last_date <- max(daily_collisions$Crash_Date)  # Last date in your historical data
forecast_dates <- seq.Date(from = last_date + 1, by = "day", length.out = 30)

forecast_data <- data.frame(
  Date = forecast_dates,
  Forecast = as.numeric(forecast_future$mean),
  Lower_80 = forecast_future$lower[, "80%"],
  Upper_80 = forecast_future$upper[, "80%"],
  Lower_95 = forecast_future$lower[, "95%"],
  Upper_95 = forecast_future$upper[, "95%"]
)


# Plot the forecast using ggplot2
ggplot(forecast_data, aes(x = Date)) +
  geom_line(aes(y = Forecast), color = "red") +
```

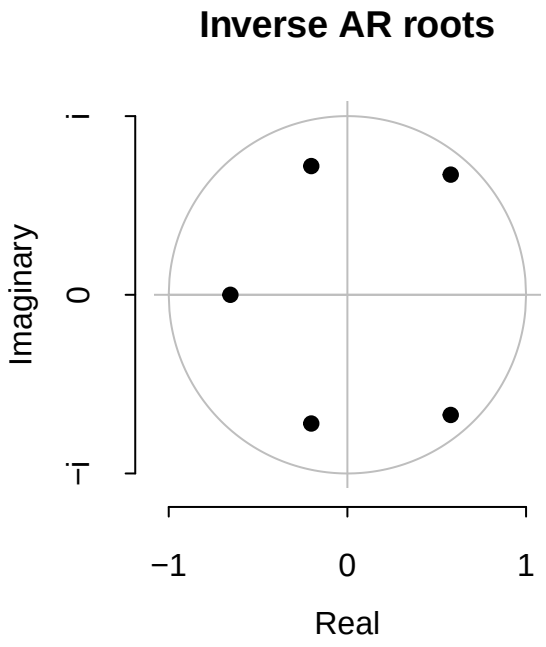
```
  geom_ribbon(aes(ymin = Lower_95, ymax = Upper_95), fill = "grey80", alpha = 0.2) +
    geom_ribbon(aes(ymin = Lower_80, ymax = Upper_80), fill = "grey60", alpha = 0.4) +
  labs(title = "30 Day Collision Forecast",
       x = "Date",
       y = "Forecasted Number of Collisions") +
  theme_minimal()
```

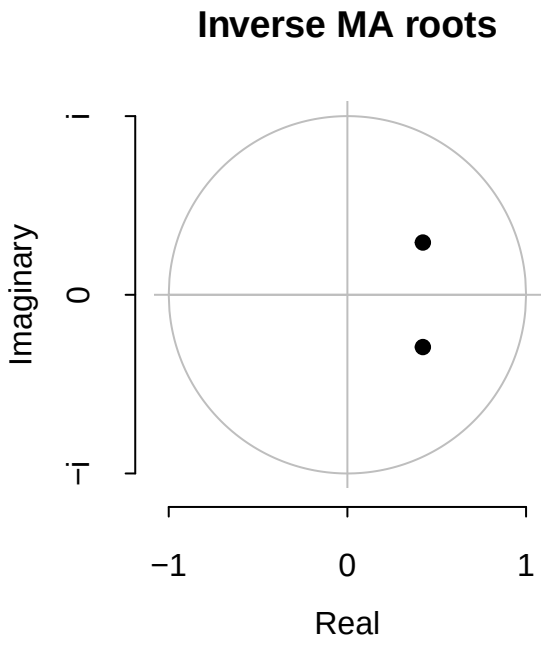



```
# Time series decomposition
decomposition <- stl(collision_ts, s.window = "periodic")

# Plot decomposed components
plot(decomposition)
```




```r
# Model selection
# Example: Fit a SARIMA model
sarima_model <- auto.arima(collision_ts, seasonal = TRUE)

# Model validation
# Diagnostic plots
plot(sarima_model)
```






```
# Forecasting
forecast_values <- forecast(sarima_model, h = 30)

# Evaluate forecast accuracy
accuracy(forecast_values)
```

```
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.02631048 15.03234 11.64846 -9.676269 26.68143 0.6615329
##                      ACF1
## Training set -0.01223579
```

Regression Analysis

```
# Load required libraries
library(ggplot2)
library(dplyr)
library(tidyr)
library(caret)
```

```
## Loading required package: lattice
```

```
# Assuming 'Injury Severity.x' is a factor variable indicating the severity of crashes
# Let's encode it into numeric values for regression analysis
combined_data$Injury_Severity_Num <- as.numeric(factor(combined_data$`Injury Severity.x`))
```

```r
# Selecting different variables for the regression model
model_data <- combined_data %>%
  select(`Injury_Severity_Num`, `Weather`, `Surface Condition`, `Light`, `Vehicle Year`, `Vehicle Make`

# Convert Weather, Surface_Condition, Light, Vehicle_Make, and Vehicle_Model to factors
model_data$Weather <- as.factor(model_data$Weather)
model_data$Surface_Condition <- as.factor(model_data$`Surface Condition`)
model_data$Light <- as.factor(model_data$Light)
model_data$Vehicle_Year <- as.factor(model_data$`Vehicle Year`)
model_data$Vehicle_Make <- as.factor(model_data$`Vehicle Make`)
model_data$Vehicle_Model <- as.factor(model_data$`Vehicle Model`)

# Encoding factors into numeric values for regression analysis
model_data <- model_data %>%
  mutate(Weather_Num = as.numeric(Weather),
         Surface_Condition_Num = as.numeric(Surface_Condition),
         Light_Num = as.numeric(Light),
         Vehicle_Make_Num = as.numeric(Vehicle_Make),
         Vehicle_Model_Num = as.numeric(Vehicle_Model))

# Perform multiple linear regression
model <- lm(Injury_Severity_Num ~ Weather_Num + Surface_Condition_Num + Light_Num + Vehicle_Year + Vehi

# Summary of the regression model
summary(model)
```

```
##
## Call:
## lm(formula = Injury_Severity_Num ~ Weather_Num + Surface_Condition_Num +
##     Light_Num + Vehicle_Year + Vehicle_Make_Num + Vehicle_Model_Num,
##     data = model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5149 -0.2874 -0.2589 -0.1987  3.0065
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.010e+00  1.298e-02 154.778  < 2e-16 ***
## Weather_Num             1.035e-02  9.569e-04  10.814  < 2e-16 ***
## Surface_Condition_Num -7.883e-03  4.148e-04 -19.003  < 2e-16 ***
## Light_Num              -2.456e-03  1.068e-03  -2.300 0.021466 *
## Vehicle_Year1          -5.815e-02  4.360e-01  -0.133 0.893893
## Vehicle_Year2          -7.445e-02  6.165e-01  -0.121 0.903883
## Vehicle_Year3          -3.647e-02  4.360e-01  -0.084 0.933345
## Vehicle_Year4           5.494e-04  6.165e-01   0.001 0.999289
## Vehicle_Year8          -4.870e-02  6.165e-01  -0.079 0.937039
## Vehicle_Year13         -2.182e-02  4.360e-01  -0.050 0.960090
## Vehicle_Year14         -4.403e-02  6.165e-01  -0.071 0.943058
## Vehicle_Year15          2.847e-01  3.560e-01   0.800 0.423836
## Vehicle_Year97          1.003e+00  6.165e-01   1.626 0.103875
## Vehicle_Year99          6.364e-02  1.213e-01   0.525 0.599920
## Vehicle_Year198        -2.987e-02  6.165e-01  -0.048 0.961361
```

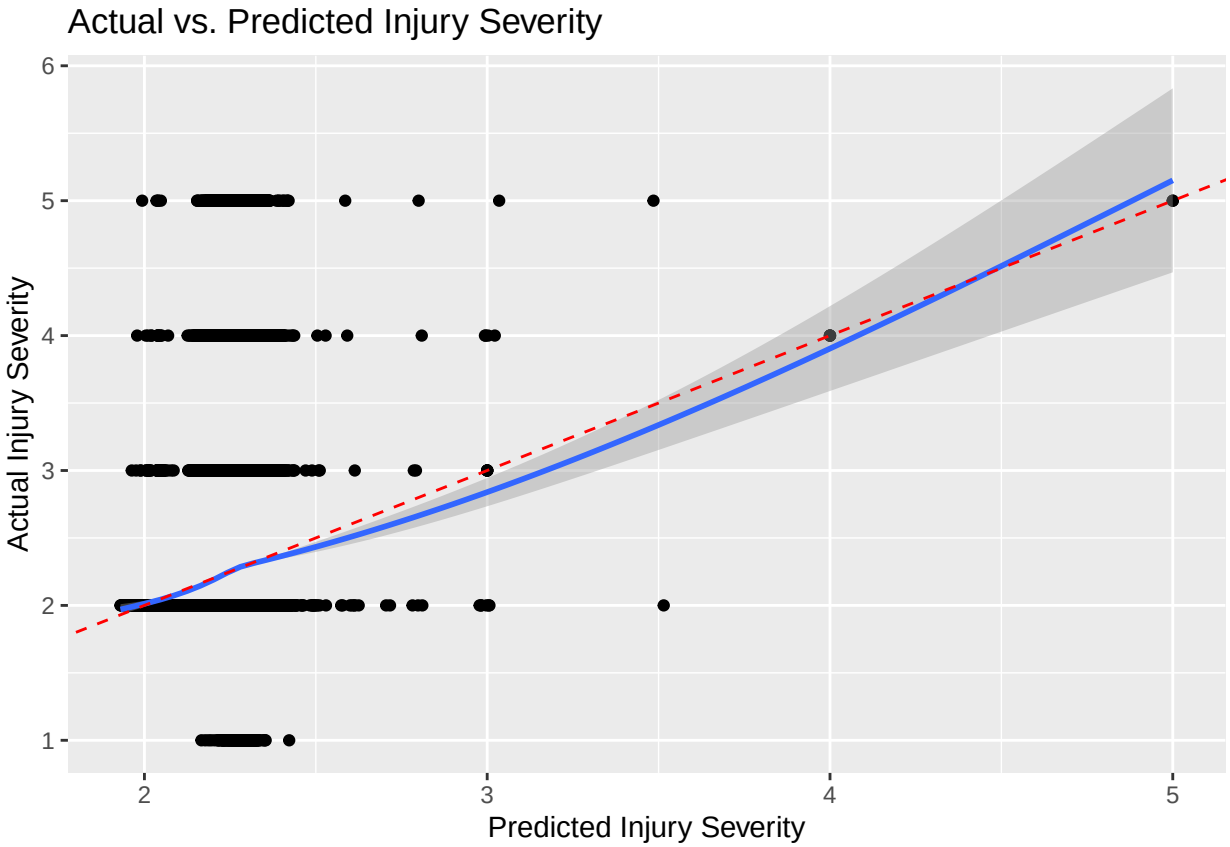
```
## Vehicle_Year200        9.800e-01  4.360e-01   2.248 0.024591 *
## Vehicle_Year201        9.502e-01  4.360e-01   2.179 0.029301 *
## Vehicle_Year202        5.602e-03  6.165e-01   0.009 0.992750
## Vehicle_Year215        1.953e+00  6.165e-01   3.168 0.001533 **
## Vehicle_Year999       -1.779e-02  6.165e-01  -0.029 0.976973
## Vehicle_Year1005      -6.479e-02  6.165e-01  -0.105 0.916303
## Vehicle_Year1008      -3.260e-02  6.165e-01  -0.053 0.957827
## Vehicle_Year1012       4.015e-02  6.165e-01   0.065 0.948070
## Vehicle_Year1014       2.015e-02  4.360e-01   0.046 0.963136
## Vehicle_Year1015       2.004e+00  6.165e-01   3.250 0.001152 **
## Vehicle_Year1025      -1.297e-02  6.165e-01  -0.021 0.983215
## Vehicle_Year1111      -2.662e-02  3.084e-01  -0.086 0.931196
## Vehicle_Year1140      -6.953e-02  6.165e-01  -0.113 0.910197
## Vehicle_Year1234       2.022e-02  4.360e-01   0.046 0.963006
## Vehicle_Year1900       1.591e-01  7.769e-02   2.048 0.040535 *
## Vehicle_Year1901      -1.741e-02  3.560e-01  -0.049 0.961009
## Vehicle_Year1911      -7.048e-02  6.165e-01  -0.114 0.908983
## Vehicle_Year1930       1.863e-03  6.165e-01   0.003 0.997589
## Vehicle_Year1938      -4.583e-03  6.165e-01  -0.007 0.994069
## Vehicle_Year1946       9.262e-01  6.165e-01   1.502 0.133008
## Vehicle_Year1947       3.002e+00  6.165e-01   4.869 1.12e-06 ***
## Vehicle_Year1949       1.997e+00  6.165e-01   3.240 0.001196 **
## Vehicle_Year1955       4.901e-01  4.360e-01   1.124 0.260999
## Vehicle_Year1959       9.509e-01  4.360e-01   2.181 0.029186 *
## Vehicle_Year1960      -3.975e-02  6.165e-01  -0.064 0.948588
## Vehicle_Year1961       7.014e-02  6.165e-01   0.114 0.909419
## Vehicle_Year1963       4.713e-01  4.360e-01   1.081 0.279758
## Vehicle_Year1965      -2.666e-02  4.360e-01  -0.061 0.951250
## Vehicle_Year1966      -2.231e-02  3.560e-01  -0.063 0.950047
## Vehicle_Year1967       5.801e-02  6.165e-01   0.094 0.925036
## Vehicle_Year1968      -6.748e-03  3.560e-01  -0.019 0.984879
## Vehicle_Year1969       1.748e-01  2.758e-01   0.634 0.526359
## Vehicle_Year1970      -1.650e-02  3.560e-01  -0.046 0.963029
## Vehicle_Year1971       2.269e-03  4.360e-01   0.005 0.995847
## Vehicle_Year1972       4.603e-01  3.084e-01   1.493 0.135550
## Vehicle_Year1974      -2.637e-02  2.759e-01  -0.096 0.923848
## Vehicle_Year1975       3.388e-01  3.560e-01   0.952 0.341269
## Vehicle_Year1976       1.485e+00  4.360e-01   3.407 0.000657 ***
## Vehicle_Year1977       9.593e-01  4.360e-01   2.200 0.027779 *
## Vehicle_Year1978       5.829e-01  1.952e-01   2.986 0.002826 **
## Vehicle_Year1979       9.876e-01  3.560e-01   2.774 0.005539 **
## Vehicle_Year1980       2.229e-01  2.057e-01   1.084 0.278477
## Vehicle_Year1981       4.879e-01  3.084e-01   1.582 0.113606
## Vehicle_Year1982       7.589e-01  2.057e-01   3.689 0.000226 ***
## Vehicle_Year1983       3.847e-01  1.595e-01   2.412 0.015850 *
## Vehicle_Year1984       2.458e-01  1.782e-01   1.379 0.167842
## Vehicle_Year1985       2.502e-01  1.078e-01   2.321 0.020288 *
## Vehicle_Year1986       3.165e-01  9.924e-02   3.189 0.001429 **
## Vehicle_Year1987       2.623e-01  7.775e-02   3.374 0.000741 ***
## Vehicle_Year1988       2.585e-01  6.580e-02   3.930 8.51e-05 ***
## Vehicle_Year1989       2.367e-01  5.941e-02   3.985 6.76e-05 ***
## Vehicle_Year1990       2.590e-01  5.152e-02   5.027 5.00e-07 ***
## Vehicle_Year1991       2.370e-01  5.009e-02   4.732 2.23e-06 ***
## Vehicle_Year1992       2.833e-01  4.501e-02   6.295 3.08e-10 ***
```

```
## Vehicle_Year1993       3.363e-01  3.664e-02   9.178  < 2e-16 ***
## Vehicle_Year1994       3.060e-01  3.116e-02   9.818  < 2e-16 ***
## Vehicle_Year1995       2.639e-01  2.673e-02   9.871  < 2e-16 ***
## Vehicle_Year1996       3.248e-01  2.406e-02  13.498  < 2e-16 ***
## Vehicle_Year1997       2.806e-01  2.007e-02  13.982  < 2e-16 ***
## Vehicle_Year1998       2.841e-01  1.835e-02  15.479  < 2e-16 ***
## Vehicle_Year1999       2.547e-01  1.671e-02  15.247  < 2e-16 ***
## Vehicle_Year2000       2.609e-01  1.485e-02  17.565  < 2e-16 ***
## Vehicle_Year2001       2.573e-01  1.461e-02  17.617  < 2e-16 ***
## Vehicle_Year2002       2.646e-01  1.389e-02  19.043  < 2e-16 ***
## Vehicle_Year2003       2.763e-01  1.323e-02  20.886  < 2e-16 ***
## Vehicle_Year2004       2.655e-01  1.285e-02  20.654  < 2e-16 ***
## Vehicle_Year2005       2.676e-01  1.274e-02  21.009  < 2e-16 ***
## Vehicle_Year2006       2.511e-01  1.245e-02  20.171  < 2e-16 ***
## Vehicle_Year2007       2.483e-01  1.231e-02  20.167  < 2e-16 ***
## Vehicle_Year2008       2.356e-01  1.232e-02  19.124  < 2e-16 ***
## Vehicle_Year2009       2.320e-01  1.268e-02  18.304  < 2e-16 ***
## Vehicle_Year2010       2.507e-01  1.245e-02  20.137  < 2e-16 ***
## Vehicle_Year2011       2.356e-01  1.221e-02  19.291  < 2e-16 ***
## Vehicle_Year2012       2.557e-01  1.203e-02  21.253  < 2e-16 ***
## Vehicle_Year2013       2.448e-01  1.178e-02  20.779  < 2e-16 ***
## Vehicle_Year2014       2.383e-01  1.172e-02  20.338  < 2e-16 ***
## Vehicle_Year2015       2.505e-01  1.163e-02  21.545  < 2e-16 ***
## Vehicle_Year2016       2.324e-01  1.174e-02  19.798  < 2e-16 ***
## Vehicle_Year2017       2.322e-01  1.203e-02  19.295  < 2e-16 ***
## Vehicle_Year2018       2.351e-01  1.263e-02  18.621  < 2e-16 ***
## Vehicle_Year2019       2.268e-01  1.312e-02  17.286  < 2e-16 ***
## Vehicle_Year2020       2.184e-01  1.454e-02  15.019  < 2e-16 ***
## Vehicle_Year2021       2.205e-01  1.572e-02  14.025  < 2e-16 ***
## Vehicle_Year2022       2.149e-01  1.814e-02  11.846  < 2e-16 ***
## Vehicle_Year2023       2.500e-01  2.487e-02  10.050  < 2e-16 ***
## Vehicle_Year2024       2.289e-01  7.770e-02   2.946 0.003222 **
## Vehicle_Year2025      -5.511e-02  6.165e-01  -0.089 0.928774
## Vehicle_Year2033      -2.596e-02  3.560e-01  -0.073 0.941863
## Vehicle_Year2040      -4.317e-03  6.165e-01  -0.007 0.994413
## Vehicle_Year2041      -6.380e-02  6.165e-01  -0.103 0.917580
## Vehicle_Year2048      -1.418e-02  6.165e-01  -0.023 0.981644
## Vehicle_Year2055      -6.124e-02  6.165e-01  -0.099 0.920872
## Vehicle_Year2099      -3.137e-02  6.165e-01  -0.051 0.959421
## Vehicle_Year2100      -4.683e-02  6.165e-01  -0.076 0.939448
## Vehicle_Year2101      -5.775e-02  6.165e-01  -0.094 0.925366
## Vehicle_Year2102       4.464e-02  6.165e-01   0.072 0.942282
## Vehicle_Year2103      -1.125e-03  4.360e-01  -0.003 0.997941
## Vehicle_Year2104       5.026e-01  4.360e-01   1.153 0.249033
## Vehicle_Year2105       1.375e-02  6.165e-01   0.022 0.982211
## Vehicle_Year2107      -4.734e-02  6.165e-01  -0.077 0.938794
## Vehicle_Year2109       2.980e+00  6.165e-01   4.834 1.34e-06 ***
## Vehicle_Year2200      -5.122e-02  6.165e-01  -0.083 0.933788
## Vehicle_Year2201      -2.617e-02  6.165e-01  -0.042 0.966139
## Vehicle_Year2204      -3.336e-02  6.165e-01  -0.054 0.956844
## Vehicle_Year2205      -6.765e-03  6.165e-01  -0.011 0.991246
## Vehicle_Year2208       1.960e+00  6.165e-01   3.180 0.001473 **
## Vehicle_Year2911       4.544e-02  6.165e-01   0.074 0.941247
## Vehicle_Year2912       3.775e-03  4.360e-01   0.009 0.993092
```

```
## Vehicle_Year2914      -2.984e-02  3.560e-01  -0.084 0.933205
## Vehicle_Year2915      -9.341e-03  6.165e-01  -0.015 0.987911
## Vehicle_Year2916       1.928e-02  4.360e-01   0.044 0.964736
## Vehicle_Year2917       1.004e+00  6.165e-01   1.628 0.103450
## Vehicle_Year2918      -3.586e-02  6.165e-01  -0.058 0.953609
## Vehicle_Year2919      -4.610e-02  6.165e-01  -0.075 0.940393
## Vehicle_Year2991       9.700e-01  6.165e-01   1.573 0.115613
## Vehicle_Year2996       2.825e-01  3.560e-01   0.793 0.427531
## Vehicle_Year2997       6.581e-04  6.165e-01   0.001 0.999148
## Vehicle_Year2998       9.206e-01  6.165e-01   1.493 0.135345
## Vehicle_Year3003       4.915e-01  4.360e-01   1.127 0.259642
## Vehicle_Year3013       3.649e-02  6.165e-01   0.059 0.952806
## Vehicle_Year3863      -6.948e-02  6.165e-01  -0.113 0.910264
## Vehicle_Year5005      -3.093e-02  6.165e-01  -0.050 0.959979
## Vehicle_Year7817       2.504e-02  6.165e-01   0.041 0.967605
## Vehicle_Year8008      -1.784e-02  6.165e-01  -0.029 0.976910
## Vehicle_Year8888      -1.014e-02  4.360e-01  -0.023 0.981442
## Vehicle_Year9999       2.071e-02  8.088e-02   0.256 0.797940
## Vehicle_Make_Num       3.161e-05  2.773e-06  11.397  < 2e-16 ***
## Vehicle_Model_Num     -5.816e-06  7.438e-07  -7.820 5.31e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6164 on 172238 degrees of freedom
## Multiple R-squared:  0.008243,   Adjusted R-squared:  0.007425
## F-statistic: 10.08 on 142 and 172238 DF,  p-value: < 2.2e-16
```

```r
# Predictions using the model
predictions <- predict(model)

# Visualize the actual vs. predicted values
ggplot(model_data, aes(x = predictions, y = Injury_Severity_Num)) +
  geom_point() +
  geom_smooth() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(x = "Predicted Injury Severity", y = "Actual Injury Severity", title = "Actual vs. Predicted Inj
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```


Actual vs. Predicted Injury Severity
Actual Injury Severity
Predicted Injury Severity
1
2
3
4
5
6
2
3
4
5