Highway Hustle - Unraveling Metro Interstate Traffic Trends

Group Members:

Abraar Patel (2069545)

Talha Sohail (2092858)

Ujwal Joshi (2018577)

Kevin Zheng (2022183)

Alton Phan (2020230)

INTRODUCTION

In the ever-evolving world of urban landscapes, we find inspiration in the vibrant heartbeat of cities and the endless potential for progress. As population growth and urbanization continue their inexorable march, our motivation to decode the intricate patterns of urban life intensifies. Urban transportation networks face increasing challenges due to population growth and urbanization. Understanding traffic patterns and congestion on metro interstate highways is crucial for efficient urban planning and management. By unraveling the complexities of traffic patterns, we aim to provide valuable insights for city planners, policymakers, and researchers. We plan to embark on a detailed analysis of the current state of metro interstate traffic, examining congestion hotspots, peak hours, and the impact on overall urban mobility.

ABOUT THE DATA

We chose a comprehensive **metro interstate traffic dataset** sourced from Kaggle to analyze and uncover valuable insights into traffic flow, congestion and accurately forecast traffic volume. The dataset consists of **48,205 observations** and following **9 variables**:

- ***holiday***: a categorical variable representing a national or regional holiday.
- ***temp***: a numeric variable that represents the average temperature in kelvin.
- ***rain_1h***: a numeric variable that shows the amount of rain occurring in an hour in mm.
- ***snow_1h***: a numeric variable that shows the amount of snow occurring in an hour in mm.
- ***clouds_all***: numeric variable representing the percentage of cloud cover.
- ***weather_main***: categorical variable containing a short textual description of the current weather.
- ***weather_description***: categorical variable providing longer textual description and detail for the current weather.
- ***date_time***: datetime variable that shows the hour of the data collected in local CST time.
- ***traffic_volume***: numeric variable that shows the hourly I-94 reported westbound traffic volume.

The **main question** we want to answer is: **Which variables or factors have the most significant impact on traffic volume and how accurate are the models in predicting traffic volume ?**

```r
# Importing the dataset and displaying the summary
traffic = read.csv("C:/Users/hp/Downloads/Metro_Interstate_Traffic_Volume.csv")
summary(traffic)
```

```
traffic_volume   holiday              temp          rain_1h
Min.   :   0   Length:48204       Min.   :  0.0   Min.   :   0.000
1st Qu.:1193   Class :character   1st Qu.:272.2   1st Qu.:   0.000
Median :3380   Mode  :character   Median :282.4   Median :   0.000
Mean   :3260                      Mean   :281.2   Mean   :   0.334
3rd Qu.:4933                      3rd Qu.:291.8   3rd Qu.:   0.000
Max.   :7280                      Max.   :310.1   Max.   :9831.300
   snow_1h            clouds_all     weather_main       weather_description
Min.   :0.0000000   Min.   :  0.00   Length:48204       Length:48204
1st Qu.:0.0000000   1st Qu.:  1.00   Class :character   Class :character
Median :0.0000000   Median : 64.00   Mode  :character   Mode  :character
Mean   :0.0002224   Mean   : 49.36
3rd Qu.:0.0000000   3rd Qu.: 90.00
Max.   :0.5100000   Max.   :100.00
 date_time
Length:48204
Class :character
Mode  :character
```

```r
# checking if there are any missing values in the entire dataset
# (returns TRUE if there are missing values , otherwise FALSE)
any(is.na(traffic))
```

```
[1] FALSE
```

There are no missing values in the dataset.

```r
# Converting the temperatures from Kelvin to Farenheit
traffic$temp <- (traffic$temp - 273.15) * 9/5 + 32
```

After conducting some data visualizations, it became evident that data cleaning needed. Implementing data cleaning is an important step in the data analytics process.

DATA CLEANING (Abraar and Talha)

During data cleaning, we formatted the date_time column, extracted time-related features, and categorized hours into distinct periods. We converted the holiday variable to a binary format, where 0 represents no holiday, and 1 represents a holiday. The outliers in temp and rain_1h column were removed. We also simplified weather conditions into 'thunderstorm,' 'mist,' 'fog,' and 'haze,' since they are the most distinct weather conditions and then created dummy variables for weather_description column. Additionally, we separated the snow_1h variable into two categories: "snow" and "no_snow" and then created a new binary variable, snow_present, which indicates the presence of snow (1) or the absence of snow (0) in the dataset. The negative and zero values in the traffic_volume column were also removed.The unnecessary columns were then dropped. The cleaned dataset has 15 variables: ***traffic_volume***, ***holiday***, ***temp***, ***rain_1h***, ***clouds_all***, ***date***, ***weekday***, ***hour***, ***month***, ***year***, ***fog***, ***haze***, ***mist***, ***thunderstorm***, ***snow_present.***

```
# Removing the outlier in temp variable and rain_1h variable
traffic <- traffic[traffic$temp > -400, ]
traffic <- traffic[traffic$rain_1h <2500, ]

traffic$date_time <- strptime(traffic$date_time, format = "%d-%m-%Y %H:%M")
# Formatting the date_time column in the desired format (%Y-%m-%d %H:%M)
traffic$date_time <- strftime(traffic$date_time, format = "%Y-%m-%d %H:%M")

# Extracting additional features from date_time variable
#(For weekdays, Monday is 0 and Sunday is 6)
traffic$date_time = as.POSIXct(traffic$date_time)
traffic$date = as.Date(traffic$date_time)
traffic$weekday = as.numeric(format(traffic$date_time, "%w"))
traffic$hour = as.numeric(format(traffic$date_time, "%H"))
traffic$month = as.numeric(format(traffic$date_time, "%m"))
traffic$year = as.numeric(format(traffic$date_time, "%Y"))
```

The full data cleaning code can be found in the R source code file.

```
# Separating snow_1h into categories such as "snow" and "no_snow"
traffic$snow_1h <- ifelse(traffic$snow_1h > 0, "snow", "no_snow")

# creating new column snow_present (binary variable) which specifies
# if there is a snow or not
traffic$snow_present <- ifelse(traffic$snow_1h == "snow", 1, 0)

# Dropping the unnecessary or not required columns
```

```r
# (date_time, weather_descriptionother column, weather_description
# and weather_main column)
# Dropped the date_time column because we already extracted the
# features from the date_time column.
# Dropped snow_1h since we already one-hot encoded these columns.
traffic <- traffic[, !colnames(traffic) %in%
                     c("date_time", "snow_1h", "weather_description",
                       "weather_descriptionother",
                       "weather_main")]

# Checking if there are negative or zero values in traffic_volume column
if (any(traffic$traffic_volume < 0)) {
  cat("There are negative values in traffic_volume column.\n")
} else if (any(traffic$traffic_volume == 0)) {
  cat("There are zero values in traffic_volume column.\n")
} else {
  cat("There are no negative or zero values in traffic_volume column.\n")
}
```

```
There are zero values in traffic_volume column.
```

```r
# Excluding rows with zero or negative traffic_volume
traffic <- traffic[traffic$traffic_volume > 0, ]
```

MULTIPLE LINEAR REGRESSION MODEL (Abraar and Talha)

In this section, we focus on utilizing a multiple linear regression model to see which predictors are significant in impacting and predicting the interstate highway traffic volume. Multiple linear Regression model tries to find the direct correlation between the response variable, **traffic_volume**, against the other potential predictors. We are using multiple linear regression model because the response variable is a continuous or quantitative variable. The biggest advantage of this model is the ability to clearly understand how the response variable is changing with a one unit increase/decrease in each of the predictors. Multiple linear Regression model is also easy to understand and interpret and is also quick to train and make predictions.The disadvantage of multiple linear regression model is that it does not capture complex and non-linear relationships in the data. Furthermore, multiple linear regression models exhibit sensitivity to outliers, leading to potential distortions in regression coefficients and predictive outcomes.

The multiple linear regression formula would be as follows:

$$\begin{aligned}\text{traffic\_volume} =& \beta_0 + \beta_1 * \text{holiday} + \beta_2 * \text{temp} + \beta_3 * \text{rain\_1h} \\ &+ \beta_4 * \text{clouds\_all} + \beta_5 * \text{date} + \beta_6 * \text{weekday} + \beta_7 * \text{hour} \\ &+ \beta_8 * \text{month} + \beta_9 * \text{year} + \beta_{10} * \text{fog} + \beta_{11} * \text{haze} \\ &+ \beta_{12} * \text{mist} + \beta_{13} * \text{thunderstorm} + \beta_{14} * \text{snow\_present} + \epsilon\end{aligned}$$

```r
# creating the multiple linear regression model
traffic.lm = lm(traffic_volume ~ ., data=traffic)
summary(traffic.lm)
```

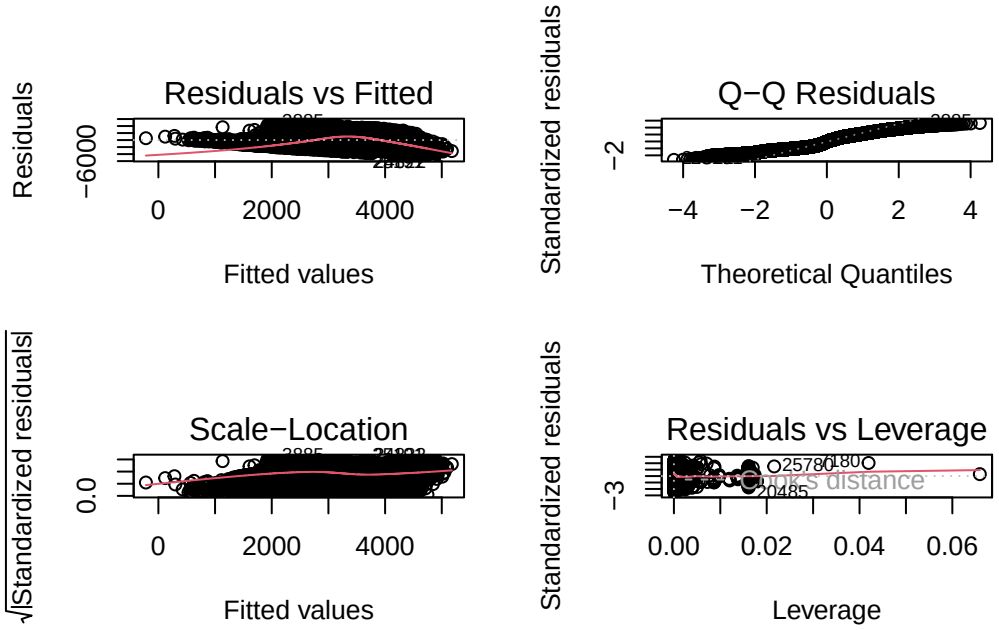
```
Call:
lm(formula = traffic_volume ~ ., data = traffic)

Residuals:
    Min      1Q  Median      3Q     Max
-4862.3 -1650.1   -63.3  1533.2  4874.0

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.765e+06  6.893e+05  -6.913 4.82e-12 ***
holiday      -1.190e+03  2.354e+02  -5.055 4.32e-07 ***
temp          1.013e+01  3.928e-01  25.790  < 2e-16 ***
rain_1h      -5.214e+01  8.497e+00  -6.136 8.53e-10 ***
clouds_all    3.776e+00  2.218e-01  17.022  < 2e-16 ***
date         -6.642e+00  9.576e-01  -6.936 4.10e-12 ***
weekday       7.569e+01  4.183e+00  18.095  < 2e-16 ***
hour          9.441e+01  1.229e+00  76.821  < 2e-16 ***
month         1.866e+02  2.930e+01   6.368 1.93e-10 ***
year          2.419e+03  3.499e+02   6.915 4.73e-12 ***
fog          -2.421e+02  6.168e+01  -3.925 8.67e-05 ***
haze          2.508e+02  5.085e+01   4.933 8.14e-07 ***
mist         -1.945e+02  2.615e+01  -7.436 1.05e-13 ***
thunderstorm -4.758e+02  5.924e+01  -8.032 9.81e-16 ***
snow_present -1.614e+02  2.316e+02  -0.697    0.486
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1834 on 48176 degrees of freedom
Multiple R-squared:  0.1481,    Adjusted R-squared:  0.1479
```

```
F-statistic: 598.2 on 14 and 48176 DF,  p-value: < 2.2e-16
```

From the above summary of our multiple linear regression model, it is evident that almost all predictors are statistically significant in predicting the traffic volume except `snow present` variable. The p-value for snow_present variable is greater than 0.05 (0.486). Hence, snow_present variable is not statistically significant in predicting the traffic volume and we would exclude it from consideration. Rest of the predictors have a p-value less than 0.05 and are considered statistically significant in predicting the traffic volume. Therefore, we would retain these predictors in our model.

With an F-statistic p-value of less than 2.2×10−16, we can confidently reject the null hypothesis that
$\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = \beta_7 = \beta_8 = \beta_9 = \beta_10 = \beta_11 = \beta_12 = \beta_13 = \beta_14 = 0$. In other words, at least one predictor demonstrates a statistically significant relationship with `traffic volume`. A multiple R^2 value of **0.1481** informs us that approximately 14.81% of the variance is explained by the model, which means that we have a **very poor fit**.We also notice that variables `year`, `temp`, `hour`, `month`, `weekday`, `haze` and `clouds_all` have positive relationship with response variable `traffic volume` and the rest of the variables have a negative effect on `traffic volume`. The lack of statistical significance for `snow\_present` variable appears noteworthy, given the typical influence of snow on traffic volume. However, a closer examination of the dataset summary shows that the variable's maximum value is only 0.51mm of snow, which is a minimal amount. Consequently, it seems reasonable that the snow_present" variable is not significant, as the overall dataset suggests that the limited quantity of snow had negligible impact on traffic conditions.The multiple linear model obtained a R^2 value of **0.1481** which indicates a relatively weak fit for the model as **only 14%** of the variability in the dependent variable is explained by the independent variables. Hence, the model was not able to adequately capture the complexities of the relationship between the variables.

```
par(mfrow=c(2,2))
plot(traffic.lm)
```



From the Residuals vs Fitted,Normal Q-Q,and Scale-Location plots,we see that the assumptions of linearity, normality,and homoscedasticity(or constant variance) **do not** hold for the multiple linear model. From the Residuals vs Leverage plot, we notice there are some observations outside of Cook's distance as well as some observations which have slightly high leverage. Hence, we decide to normalize the data by **scaling the numeric variables** through min-max scaling to have a zero mean and unit variance. Scaling the data might improve the performance of the model, remove noise on the training data and also prevent possible overfitting of the data.

Now, we proceed to train the multiple linear regression model and evaluate the prediction accuracy. We will perform a randomized **80:20** (training:testing) split on our dataset, scale the numeric variables, calculate the mean squared error (MSE) and then cross-validate 10 times. Subsequently, we will determine the average MSE across these 10 iterations.

```
MSE = rep(0,10)

# Loop for 10 iterations
for (i in 1:10) {
  set.seed(i)
  sample <- sample(1:nrow(traffic), 0.8 * nrow(traffic))
  train_data <- traffic[sample, ]
  test_data <- traffic[-sample, ]
```

```r
  #scaling numeric variables to have zero mean and unit variance (min-max scaling)
  numeric_cols <- sapply(train_data, is.numeric)
  train_data_scaled <- as.data.frame(scale(train_data[, numeric_cols]))
  test_data_scaled <- as.data.frame(scale(test_data[, numeric_cols]))

  traffic.lm <- lm(traffic_volume ~ ., data = train_data_scaled)

  yhat <- predict(traffic.lm, newdata = test_data_scaled)
  MSE[i] = mean((yhat - test_data_scaled$traffic_volume)^2)
}

cat("MSE Values:", MSE, "\n")
```

```
MSE Values: 0.8524005 0.8562625 0.8568919 0.8525875 0.8535871 0.8528314 0.8537008 0.859995 0
```

```r
cat("Average Test MSE:", mean(MSE), "\n")
```

```
Average Test MSE: 0.8556354
```

The average test Mean Squared Error (MSE) obtained across all 10 training and testing iterations is **0.8556**. MSE quantifies the average squared difference between the predicted values from our model, and the actual observed values in the test sample. A more precise model tends to have an MSE closer to 0, while a less accurate model will result in a higher MSE. An MSE (Mean Squared Error) value of 0.8556 suggests that the model's predictions, on average, have a squared difference of approximately 0.8556 from the actual observed values in the test sample. Considering the average test MSE achieved by the multiple linear model is 0.8556, we conclude that the quality of our model, **traffic.lm**, is somewhat accurate in predicting traffic volume with slight discrepancies between predicted values and observed values.

RANDOM FOREST MODEL (Ujwal, Kevin and Alton) In this section, we focus on utilizing a random forest model to see if it was a better fit in predicting the interstate highway traffic volume.

```
# DATA MODELING: Random Forest
library(randomForest)
```

```
randomForest 4.7-1.1
```

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

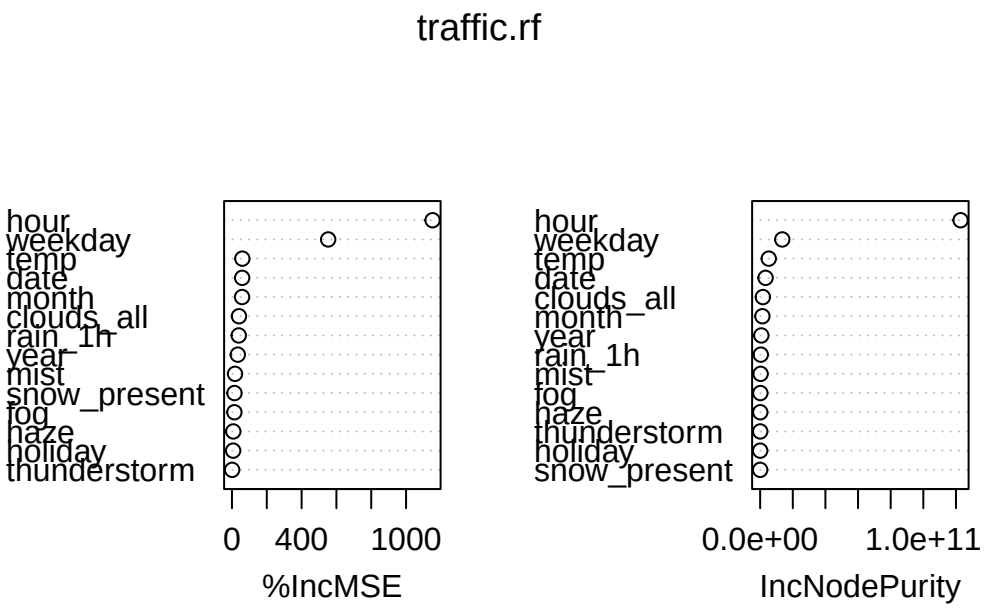
```
#split into train and test 80/20
set.seed(123)
train <- sample(1:nrow(traffic), 0.8*nrow(traffic))
traffic.train <- traffic[train, ]
traffic.test <- traffic[-train, ]
#random forest model
traffic.rf <- randomForest(traffic_volume ~., data = traffic, subset = train,
                           mtry = (ncol(traffic)-1)/ 3,
                           importance = TRUE)
traffic.rf
```

```
Call:
 randomForest(formula = traffic_volume ~ ., data = traffic, mtry = (ncol(traffic) -      1)/3
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 5

          Mean of squared residuals: 128018.8
                    % Var explained: 96.75
```

In this section, our focus turns to utilizing a random forest model to see which predictors are significant in impacting and predicting the interstate highway traffic volume. Unlike multiple linear regression, random forests excel in capturing complex, non-linear relationships within the data. The decision to opt for a random forest model is driven by its robustness in handling intricate patterns and resilience against outliers compared to linear regression. The model considers interactions among predictors, providing a more comprehensive understanding of their collective impact on traffic volume. While sacrificing some interpretability compared to multiple linear regression, random forests remain relatively

interpretable and compensate with efficient training and predictive capabilities. This approach aims to strike a balance between model complexity and predictive performance, offering insights into the nuanced relationships shaping highway traffic volume.

```
#Plot
varImpPlot(traffic.rf, sort=TRUE)
```



The most important predictors for the random forest model include **hour and weekday** since they have the highest %IncMSE and highest IncNodePurity.

```
#Calculating Test MSE
yhat_test = predict(traffic.rf, newdata = traffic.test)
mse = mean((yhat_test - traffic.test$traffic_volume)^2)
cat("MSE Value:", mse, "\n")
```

```
MSE Value: 122013.5
```

The evaluation of the random forest model's performance reveals a concerning Mean Squared Error (MSE) value of **122013.5**. This substantial MSE implies a considerable discrepancy between the predicted values and the actual observed values in the test dataset, indicating a less-than-optimal fit of the model. Moreover, the observed high MSE suggests the presence of

overfitting issues. Overfitting occurs when a model captures noise in the training data as if it were a genuine pattern, leading to poor generalization to new, unseen data.

```
# 10 iterations of randomForest model
rf_MSE = rep(0,10)

for (i in 1:10) {
  set.seed(i)
  sample <- sample(1:nrow(traffic), 0.8 * nrow(traffic))
  train_data <- traffic[sample, ]
  test_data <- traffic[-sample, ]

  #scaling numeric variables to have zero mean and unit variance (min-max scaling)
  numeric_cols <- sapply(train_data, is.numeric)
  train_data_scaled <- as.data.frame(scale(train_data[, numeric_cols]))
  test_data_scaled <- as.data.frame(scale(test_data[, numeric_cols]))

  traffic.rf = randomForest(traffic_volume ~.,
                            data = train_data_scaled,
                            mtry = (ncol(traffic)-1) / 3, importance = TRUE)

  yhat.rf = predict(traffic.rf, newdata = test_data_scaled)
  rf_MSE[i] = mean((yhat.rf - test_data_scaled$traffic_volume)^2)
}

cat("MSE Values:", rf_MSE, "\n")
```

```
MSE Values: 0.04543639 0.03951885 0.04215601 0.04040025 0.03922676 0.03978334 0.03808645 0.04
```

```
cat("Average Test MSE:", mean(rf_MSE), "\n")
```

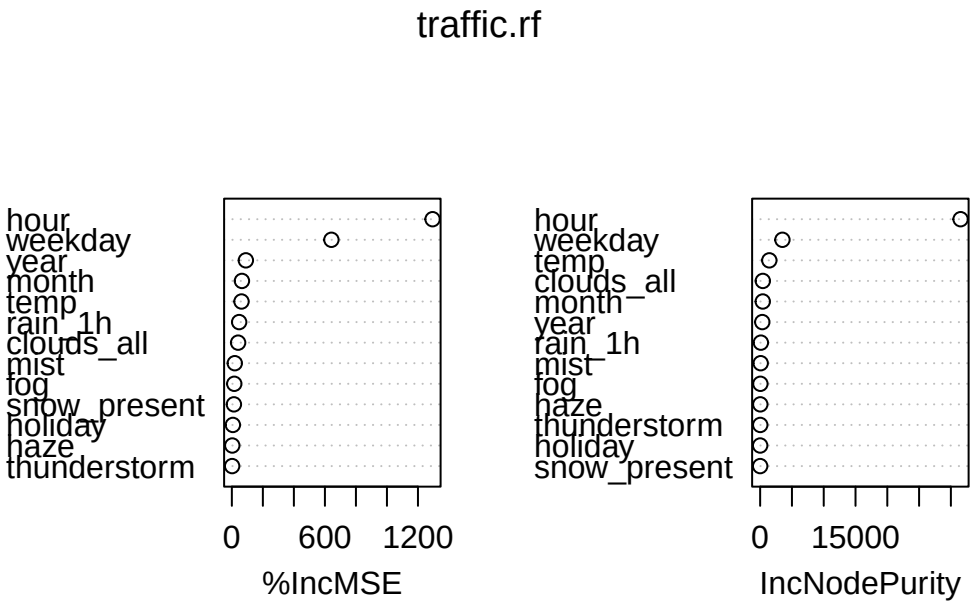
```
Average Test MSE: 0.0410031
```

In an effort to enhance the performance of our model and mitigate potential overfitting, we opted to normalize the data by employing min-max scaling on the numeric variables. This process aims to achieve a zero mean and one variance for the variables, contributing to improved model performance. The evaluation metric utilized for assessing the model's predictive accuracy is the Mean Squared Error (MSE). Across 10 iterations of training and testing, the average test MSE is calculated to be **0.0410031**. We can conclude that the test error rate of our model has improved drastically compared to the original run of the random

forest, though there still exists some very slight discrepancies between the predicted values and the actual observed values in the context of traffic volume prediction.

Now, we will display the plot of the 10 iteration random forest model.

```
# variable importance plot of the 10 iteration random forest model
varImpPlot(traffic.rf, sort=TRUE)
```




```
importance(traffic.rf)
```

```
                %IncMSE IncNodePurity
holiday        6.601517      7.026954
temp          62.206387   1442.435311
rain_1h       47.120026    113.791473
clouds_all    40.495912    418.870147
weekday      641.854480   3503.202729
hour        1294.193354  31528.421645
month         65.110667    411.123514
year          90.347472    337.302533
fog           15.649723     40.455094
haze           2.825259     27.202549
mist          18.953332     83.268206
```

```
thunderstorm    1.870367     16.759412
snow_present   13.351261      4.138492
```

The most important predictors for the 10 iteration random forest model include **hour and temp** since they have the highest %IncMSE and highest IncNodePurity. Given a regression random forest, we place more emphasis on %IncMSE than IncNodePurity and hence we will not include `temp` as an important predictor since it has a low %IncMSE and high IncNodePurity.

In summary, the refined 10-iteration random forest model, with data normalization, demonstrated a significant improvement over the initial random forest. The addition of normalization addressed overfitting issues, contributing to enhanced predictive accuracy. While some slight discrepancies still existed, the refined model struck a better balance between complexity and performance, offering valuable insights into the relationships governing traffic volume. The comparison highlights the iterative nature of model development and the importance of addressing overfitting and normalizing data to refine predictive accuracy in complex scenarios such as highway traffic volume prediction.

CONCLUSION

Upon comparing the multiple linear regression model with the random forest model, it is evident that the latter is more suitable for inferring the impact of various factors on interstate highway traffic volume. Through 10 iterations of the random forest model, it was consistently observed that the most influential variables were `hour` and `temp`. Interestingly, the finding that `holiday` did not emerge as an important variable in influencing traffic volume contrasts with common expectations given the known effects of holidays on traffic patterns. In the multiple linear regression model, the analysis revealed that variables such as year, average `temperature` in Celcius, `hour`, `month`, `weekday`, `haze`, and `clouds_all` have a positive effect on `traffic volume`. Conversely, variables like `holiday`, `amount of rain`, `date`, `fog`, `mist`, `thunderstorm`, and `snow_present` exhibit a negative effect on `traffic volume`. This aligns with the expectation that certain weather conditions, time-related factors, and seasonal variations play a role in traffic volume fluctuations. For predictive purposes, the random forest model consistently demonstrated lower test error rates compared to the multiple linear regression model. In fact, the test error rates for the cross-validated multiple linear regression model were slightly higher than those observed in the 10-iteration random forest model. This underscores the **superiority of the random forest model** in terms of predictive accuracy, making it a more effective model for understanding and forecasting interstate highway traffic volume based on the given variables.

REFERENCES

- **Interstate Traffic Dataset**. Available at https://www.kaggle.com/datasets/anshtanwar/metro-interstate-traffic-volume
- **Data Cleaning in R** https://www.geeksforgeeks.org/data-cleaning-in-r/
- **James, G, Witten, D, Hastie, T, and Tibshirani, R** *An Introduction to Statistical Learning.* Springer New York. DOI: https://doi.org/10.1007/978-1-4614-7138-7