STA 141C - Final Project

Metro Interstate Traffic Volume Report

March 20TH, 2020

Group Members:
Yutian Yang (915327218)
Xin Ye (915263534)
Xinyue Wang (915059201)
Falak Shah (914663151)

Ⅰ Introduction

Our analysis utilizes a dataset of "Metro Interstate Traffic Volume" consisting of parameters for hourly Minneapolis-St Paul, MN traffic volume for westbound I-94 from 2012-2018. In this project, we are interested in building a predictive model to predict Traffic Volume with the most significant variables. Different techniques would be applied, under a computationally efficient procedure that performs simultaneous variable and model selection.

For our project, we took on the primary task of using the bag of little bootstraps to predict the confidence interval for the correlation between traffic volume and other numerical parameters via parallel processing. In order to accomplish this goal, we, preliminarily, needed to create a model of deterministic and significant predictors that contributed to the general explanation of the generation of instate traffic volume on I-94 from Minneapolis-St.Paul during 2012-2018. Furthermore, we can use this model to verify its certain characteristics such as differences in traffic volume at different times or on holidays along with using it to predict the confidence interval for the traffic volume using BLB.

Ⅱ Data Analysis Plan

The primary interest in this project is to build a predictive model for Traffic Volume. Therefore, determining which several factors will have the most significant impact on Traffic Volume becomes our first task. Moreover, we are interested in if there are interactions between our variables. To solve these problems, we manually implemented the model selection methods, such as Forward and Backward Selection by recording the Akaike Information Criteria(AIC) and Bayesian Information Criteria(BIC) results to avoid overfitting or underfitting. We finally choose the best linear regression model with the reasonable adjusted R2 and least AIC and BIC results to ensure our model includes the most significant variables. After identifying these attributes and selecting the best model, we observe the distribution of the data and decide whether to use the bag of little bootstraps to construct a 95% confidence interval for predicting traffic volume.

2.1 Data Summary

This dataset includes the hourly measurement of Interstate 94 Westbound traffic volume for MN DoT ATR station 301, a station that is roughly midway between Minneapolis and St Paul, MN. The dataset contains 48,204 observations with 9 different attributes.

Attributes**:**

Inputs	Description
Holiday (Categorical)	US National holidays plus regional holiday, Minnesota State Fair
Temp (Numeric)	Average temp in kelvin
Rain_1h (Numeric)	Amount in mm of rain that occurred in the hour
Snow_1h (Numeric)	Amount in mm of snow that occurred in the hour
Clouds_all (Numeric)	Percentage of cloud cover
Weather_main (Categorical)	Short textual description of the current weather
Weather_description (Categorical)	Longer textual description of the current weather
Date_time (Categorical)	DateTime Hour of the data collected in local CST time

Table 1: Attributes of Metro Interstate Traffic Volume

To train our model, we split the attribute Date_time into three sub-features. Those features are year, month and day time (called "Time" in the code and figures).

Output**:**

Traffic_volume (Numeric): Hourly I-94 ATR 301 reported westbound traffic volume

2.2 Data Visualization

We collect and visualize some statistics of our dataset. Based on our visual analysis, we make the following observations about our dataset:

We first looked at the pairwise correlations between each variable. According to Figure 1, we make an initial prognosis about the relationship among various numeric variables through the correlogram. Initial observation of the correlogram suggests that traffic volume has a significant relationship only with the date_time variable.

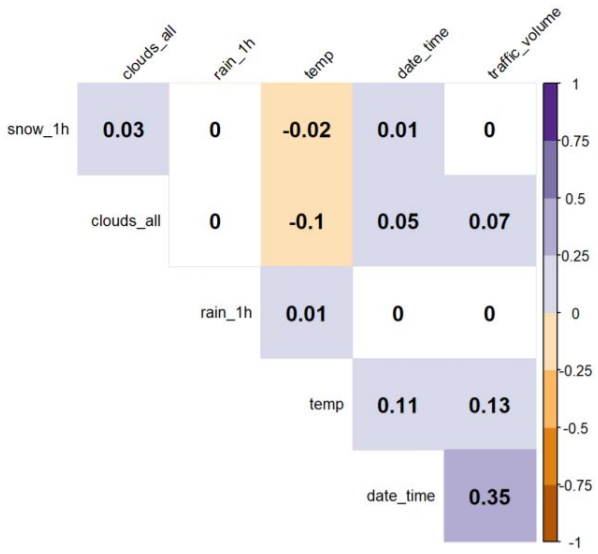

Figure 1: Corrogram

Figure 2 depicts the distribution of our variable of interest - Traffic volume. We can observe from the histogram that the distribution of traffic volume is not a normal-like distribution. It is not perfectly symmetric and slightly skewed to the left.

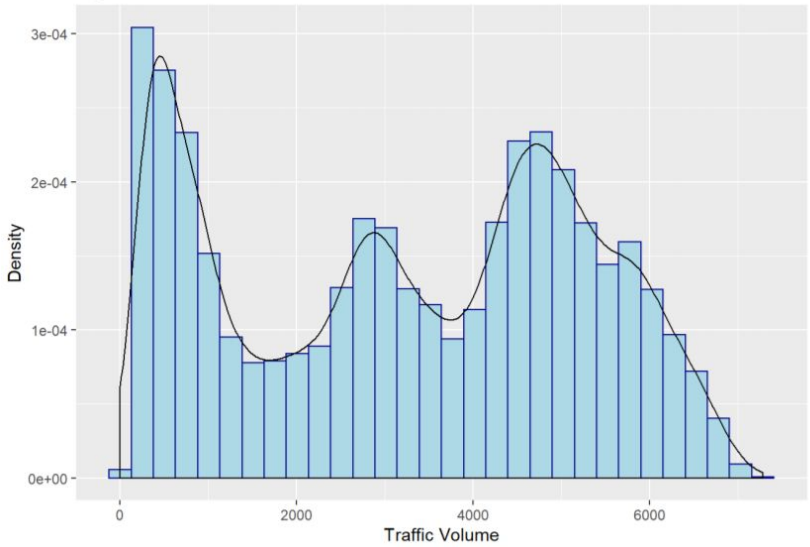

Figure 2: Distribution of Traffic Volume

Figure 3 depicts the qq plot of traffic volume, from which we can observe that it does not follow normality.

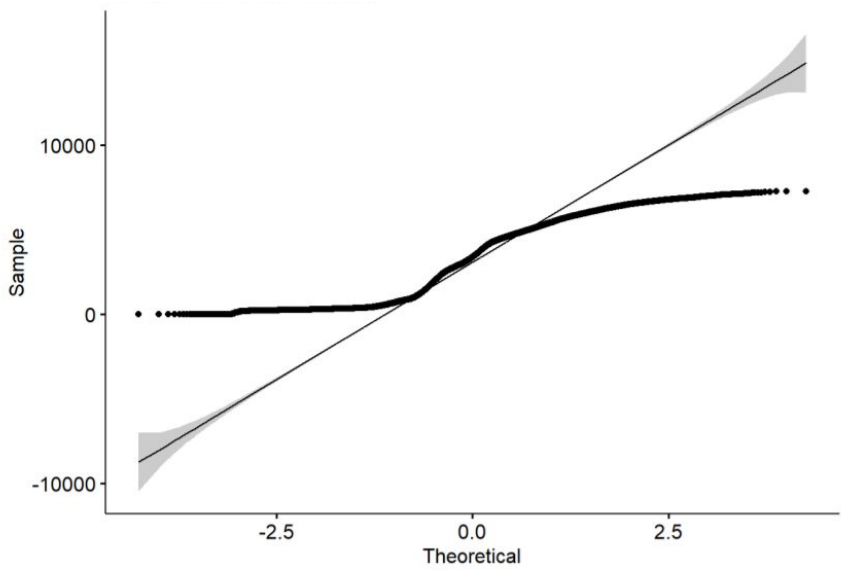

Figure 3: Q-Q Plot for Traffic Volume

Figure 4 checks the variation of traffic volume on various holidays, different weather conditions, hours and years respectively, through conditional boxplots. The most traffic volume on holidays is observed on New Years Day and the least volume is observed on Columbus Day, whereas on all the other days, the traffic volume is in mid-range. Traffic volume does not vary much with the type of weather conditions and it is similar throughout all the six years. As far as traffic volume across various hours is concerned, as expected, it increases as we move from morning to afternoon and decreases as we proceed from late evening to night.

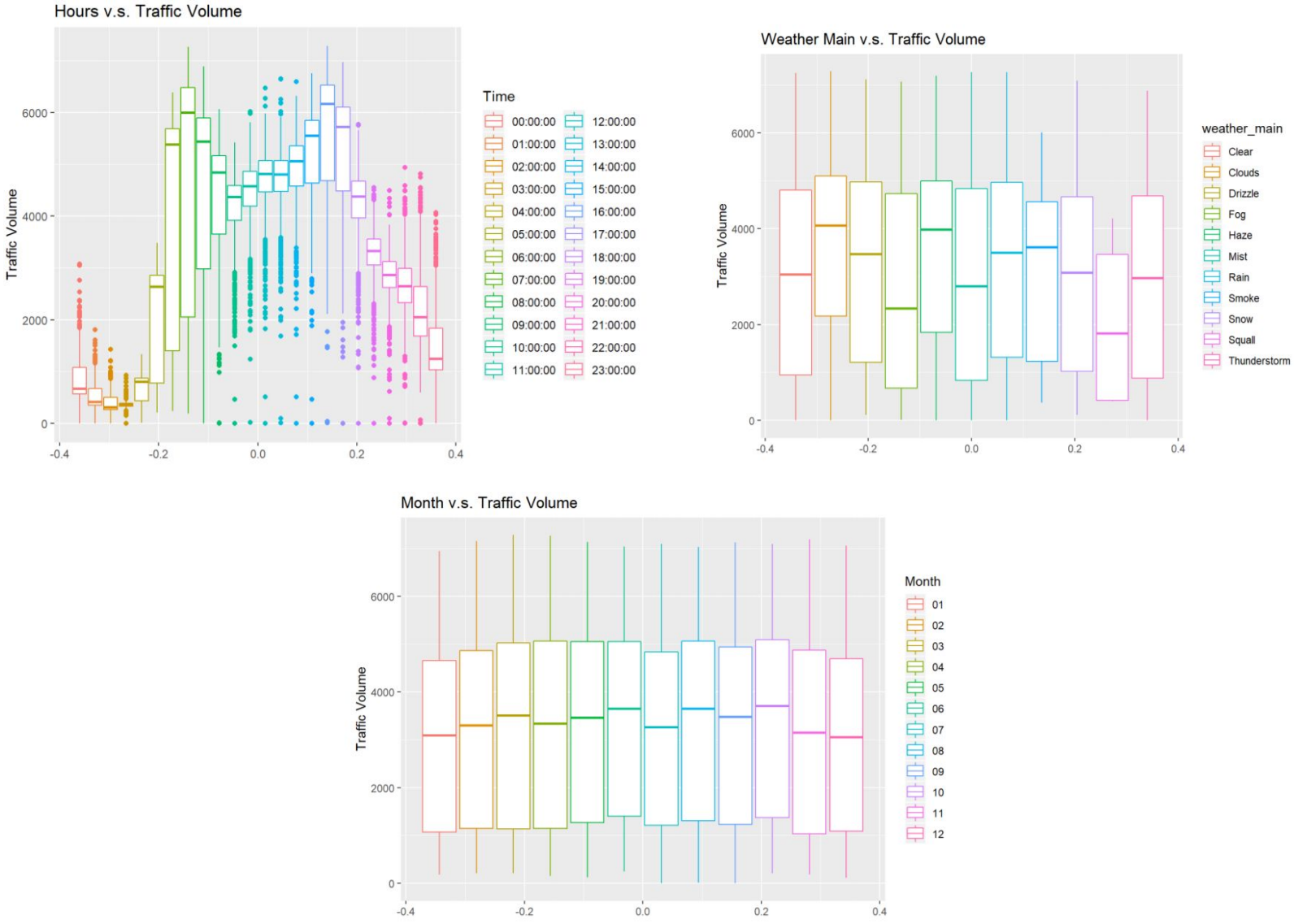

Figure 4: Boxplots of 5 Features v.s. Traffic Volume

Ⅲ Main Analysis

3.1 Regression and Model Selection

3.1.1 Regression Model with Individual Variable

In this part, we analyze and find the most important variable for traffic volume. We first applied each variable individually into the linear regression model to find their relationship with traffic volume. From linear regression models with a single variable and the correlation plot, we

conclude that hour(variable separated from date) is the most important variable for traffic volume because it gives the highest r square (0.77).

Based on that, we applied all variables without adding interaction into the model, we found out that holiday, rain_1h, snow_1h, weather description, year and months are not significant, so we eliminated them from the regression model.

3.1.2 Model Adjustion and Selection

Model:

lm1 = lm(traffic_volume~ temp + clouds_all + as.factor(weather_main))

lm2 = lm(traffic_volume~ temp + clouds_all + as.factor(time))

lm3 = lm(traffic_volume~ temp + clouds_all + as.factor(time) + as.factor(weather_main))

lm4 = lm(new_traffic_volume~ temp + clouds_all + as.factor(weather_main),data = Metro_without_time)

lm5 = lm(new_traffic_volume2~ temp + clouds_all,data = Metro_without_time_weather)

During the model selection procedure, we first constructed three linear models with different combinations of categorical variables, hour and weather_main. After the first round of model selection, we observe the dominance of the categorical variable, hour. In order to zoom out the effects of other numerical variables, we demean the impact of hour by deducting the mean of traffic volume by groups of hours from the traffic volume. Then, we do the same to another categorical variable, weather_main.

We manually implemented the Forward and Backward Selection. We start with an empty model by adding one variable to the model and record the Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) results. We then keep adding until the AIC and BIC results no longer decrease and keep the model. Then we start with a full model by deleting one variable from the model and also stop until the AIC and BIC results no longer decrease and keep the model. Finally, we choose the same linear regression model drawn from both model selection results.

$ModelNumber$	AIC	BIC	R^2
1	867035.7	867158.7	0.0392
2	796897.4	797134.6	0.7758
3	796749	797074	0.7766
4	796530.2	796653.1	0.01385
5	796575.6	796610.7	0.006542

3.1.4 Diagnostics for the final model

Model 3 is the final prediction model we drew from our analysis. Therefore, we use the Non-constant Variance test and find that both of the assumptions of normality are violated. Unfortunately, after trying both box-cox transformations, we still got the same results. Therefore, we continue to use bootstrap to get the confidence interval for predicting the traffic volume. In order to better predict the effect of the rain_1h and cloud coverage, we use Model 5 for bootstrap since the hour and weather factors yield different linear models based on categories beyond expectation.

3.2 BLB and Bootstrap Confidence Interval

BLB is a procedure that incorporates features of both the bootstrap and subsampling to yield a robust, computationally efficient means of assessing the quality of estimators. The bag of little bootstraps = subsample + bootstrap. For each bootstrap, we sample from with replacement instead of the sample from as in ordinary bootstrap.

3.2.1 Bootstrap Confidence Interval

Since our data does not follow the Gaussian distribution based on the previous distribution density plot and normal QQ plot, and as the assumptions of the Central Limit Theorem are violated i.e. the data points do not exhibit the same variance, we cannot use the classical method to calculate the confidence interval. Therefore, we implement the bag of little bootstraps to compute the Confidence Intervals.

Based on the model we previously built, we implemented the above-mentioned procedure to calculate the 95% confidence interval to predict the traffic volume, the intercept, coefficient of temperature, and coefficient of cloud coverage. In the meantime, in order to decrease the time for permuting the samples and calculating the confidence intervals, we use a multi-core, parallel implementation in which we read in the data to each worker to speed up the process and make it more efficient as compared to a single-core implementation.

We consider B = 1000 in each of the four cases to compute 95% CI using BLB and the obtained 95% confidence intervals are summarized below:

$Parameter$	$Upper\ Bound$	$Lower\ Bound$	$Confidence\ Interval$
$Traffic\ Volume$	2526.835	3750.232	$(2526.835, 3750.232)$
$Intercept$	-3540.370	-2805.921	$(-3540.370, -2805.921)$
$Tempreature$	20.81797	23.40296	$(20.81797, 23.40296)$
$Cloud\ Coverage$	3.745826	4.676337	$(3.745826, 4.676337)$

3.2.2 Interpretation for Bootstrap CI

We constructed four confidence intervals to predict the traffic volume and each coefficient in the previous model.

For the traffic volume prediction, we have a 95% confidence interval from 2526.835 to 3750.232, which means we are 95% confident that the predictive traffic volume will be between 2526.835 and 3750.232 given temperature and clouds coverage as the factors.

We also have the 95% bootstrap confidence interval for the intercept of the model to be from -3540.370 to -2805.921 which means we are 95% confident that the traffic volume should be between -3540.370 and -2805.921 if there is no temperature effect and clouds coverage effect in this model. However, as we know, traffic volume cannot be smaller than 0, so this interpretation is meaningless.

Then, we constructed the bootstrap confidence interval for each coefficient within the model. For the coefficient of temperature, we have a 95% confidence interval between 20.81797 and 23.40296, which means for each unit increase of temperature, we are 95% confident that the traffic volume will increase 20.81797 units to 23.40296 units. For the coefficient of cloud coverage, we have a 95% confidence interval from 3.745826 to 4.676337, which means for each unit increase of cloud coverage, we are 95% confident that the traffic volume will increase 3.745826 units to 4.676337 units.

Ⅳ Conclusion

Our study provides an effective predictive model with multiple important variables to predict the traffic volume for MN DoT ATR station 301. To analyze the most important variable that affects the quality of traffic volume, we concluded that time is the most important variable in the model. By adding the rest of the variables into the model, we found out that holiday, rain_1h, snow_1h, weather description, year and months are not significant, so we removed them from the regression model. In addition, the confidence intervals help us to determine the estimators of our model and make our model more reliable.

Since our data does not follow the normality assumptions, we applied bootstrap to calculate confidence intervals in order to predict traffic volume, intercept, coefficient of temperature, and coefficient of cloud coverage from our previously selected model. To decrease the processing time of computing, we also applied parallelization to assign tasks to many workers. In this way, the process is more efficient compared to a single-core implementation.

The results of our study may offer a practical inference for predicting traffic volumes. Nowadays, traffic jam is a problem all over the world and is hard to prevent. Our model reflects the changing tendency of traffic volumes. With more of the sample data to be analyzed, our model provides the possibility to predict traffic volumes ahead of departure, which can help experts to further analyze and solve the traffic jam in the future.

V References

John Hogue,Traffic data from MN Department of Transportation, Weather data from OpenWeatherMap,
UCI Machine Learning Repository, University of California, Irvine (2018)
https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume#

\
Statistical tools for high-throughput data analysis,
Visualize correlation matrix using correlogram
http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram

Statistical tools for high-throughput data analysis,
ggplot2 histogram plot : Quick start guide - R software and data visualization
http://www.sthda.com/english/wiki/ggplot2-histogram-plot-quick-start-guide-r-software-and-data-visualization

VI R Appendix

```
library(car)
library(GGally)
library(dplyr)
library(ggpubr)
library(tidyr)
library(tidyverse)
library(ggplot2)
##Import dataset and split date into month, day, and time
Metro <- read_csv("Metro_Interstate_Traffic_Volume.csv")
New_Metro = separate(Metro, date_time, c("date","time"), sep = " ", remove = TRUE,
  convert = FALSE, extra = "warn", fill = "warn")
New_Metro = separate(New_Metro, date, c("year","month","day"), sep = "-", remove = TRUE,
  convert = FALSE, extra = "warn", fill = "warn")
summary(New_Metro)
## plot the correlation bewteen variables to see if there are interactions.
#ggcorr(New_Metro)

library(corrplot)
library(RColorBrewer)
library(lubridate)
data = Metro
#extract hour from date_time
data$date_time <- hour(data$date_time)

#correlation coefficients
M <-cor(data[,-c(1,6,7)])
```

```
corrplot(M, type="upper",method="color",
         addCoef.col = "black", order="hclust", number.cex= 7/ncol(data[,-c(1,6,7)]),
         tl.col="black", tl.srt=45,tl.cex = .85,
         p.mat = cor.mtest(data[,-c(1,6,7)])$p, sig.level = 0.1, insig = "blank",
         diag=FALSE, col=brewer.pal(n=8, name="PuOr"))

#distribution of y
ggplot(data= New_Metro, aes(x = traffic_volume)) +
  geom_histogram(aes(y=..density..),color="darkblue", fill="lightblue") +
  geom_density()+
  labs(title="Figure 2: Distribution of Traffic Volume",x="Traffic Volume", y = "Density")
ggqqplot(Metro$traffic_volume,main = "QQ plot for traffic volumn")

#how holiday influence y
ggplot(data = New_Metro[New_Metro$holiday!='None',], aes(group = holiday, y=traffic_volume,color=holiday )) +
  geom_boxplot()+
  labs(title= "Figure 3:Holiday v.s. Traffic Volume",y="Traffic Volume", color = "Holiday")

#how weather influence y
  labs(title= "Figure 3:Holiday v.s. Traffic Volume",y="Traffic Volume", color = "Holiday")

#how weather influence y
ggplot(data = New_Metro, aes(group = weather_main, y=traffic_volume,color=weather_main )) +
  geom_boxplot()+
  labs(title="Weather Main v.s. Traffic Volume",y="Traffic Volume", color = "weather_main")

#how hour influence y
ggplot(data = New_Metro, aes(group = time, y=traffic_volume,color=time )) +
  geom_boxplot()+
  labs(title="Hours v.s. Traffic Volume",y="Traffic Volume", color = "Time")

#how year influence y
ggplot(data = New_Metro, aes(group = year, y=traffic_volume,color=year )) +
  geom_boxplot()+
  labs(title="Year v.s. Traffic Volume",y="Traffic Volume", color = "Year")

#how month influence y
ggplot(data = New_Metro, aes(group = month, y=traffic_volume,color= month )) +
  geom_boxplot()+
  labs(title="Month v.s. Traffic Volume",y="Traffic Volume", color = "Month")

##Linear Model Selection
lm1 = lm(traffic_volume~ temp + clouds_all + as.factor(weather_main),data = New_Metro)
### Combining AIC, BIC and linear model selection to select the prediction model.
AIC(lm1)
BIC(lm1)
summary(lm1)

###Factor Time is significant
lm2 = lm(traffic_volume~ temp + clouds_all + as.factor(time),data = New_Metro)
```

```r
summary(lm2)
AIC(lm2)
BIC(lm2)
#plot(lm2)
#hist(lm2$residuals)
### Another possible model
lm3 = lm(traffic_volume~ temp + clouds_all + as.factor(time) + as.factor(weather_main),data = New_Metro)
AIC(lm3)
BIC(lm3)
summary(lm3)

## Demean the traffic volume since it is the dominant factor
Metro_without_time = New_Metro %>%
  group_by(time)%>%
  filter(temp != 0.00) %>%
  mutate( new_traffic_volume = traffic_volume - mean(traffic_volume))

## New linear Model
lm4 = lm(new_traffic_volume~ temp + clouds_all + as.factor(weather_main),data = Metro_without_time)
AIC(lm4)
BIC(lm4)
summary(lm4)
# lm4$coefficients
# plot(lm4)

## Demean the weather factor
Metro_without_time_weather = Metro_without_time%>%
  group_by(weather_main)%>%
  mutate(new_traffic_volume2 = new_traffic_volume - mean(new_traffic_volume))
lm5 = lm(new_traffic_volume2~ temp + clouds_all,data = Metro_without_time_weather)
summary(lm5)
AIC(lm5)
BIC(lm5)
#plot(lm5)
#hist(lm5$residuals)

write_csv(Metro_without_time_weather,"Metro_without_time_weather.csv")

## Using parallelization to bootstrap for CI of traffic volume
library(parallel)
cl = makeCluster(4)
B = 1000
predict_newdata <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients
  y = coefficient[[1]] + coefficient[[2]]* data$temp + coefficient[[3]]* data$clouds_all
}
singleBoots <- function(i){
  index = sample(x = seq_len(n), size = n, replace = TRUE)
  data_star = data[index,]
```

```
  predict_newdata(data_star)
}
clusterEvalQ(cl,{
  library(tidyverse)
  data <- read_csv("Metro_without_time_weather.csv")
  n <- length(data$traffic_volume)
  predict_newdata <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients
  y = coefficient[[1]] + coefficient[[2]]* data$temp + coefficient[[3]]* data$clouds_all
}
})
predict_newdata = parSapply(cl, seq_len(B),singleBoots)
predict_newdata %>% quantile(c(0.025,0.975))
stopCluster(cl)


## Using parallelization to bootstrap for CI of intercept
cl = makeCluster(4)
B =1000
coef_beta0 <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients[[1]]
}
singleBoots <- function(i){
  index = sample(x = seq_len(n), size = n, replace = TRUE)
  data_star = data[index,]
  coef_beta0(data_star)
}
clusterEvalQ(cl,{
  library(tidyverse)
  data <- read_csv("Metro_without_time_weather.csv")
  n <- length(data$traffic_volume)
  coef_beta0 <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients[[1]]
}
})
beta0_ci = parSapply(cl, seq_len(B),singleBoots)
beta0_ci %>% quantile(c(0.025,0.975))
stopCluster(cl)

## Using parallelization to bootstrap for CI of coefficient of temperature
cl = makeCluster(4)
B =1000
coef_beta1 <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients[[2]]
}
singleBoots <- function(i){
```

```
  index = sample(x = seq_len(n), size = n, replace = TRUE)
  data_star = data[index,]
  coef_beta1(data_star)
}
clusterEvalQ(cl,{
  library(tidyverse)
  data <- read_csv("Metro_without_time_weather.csv")
  n <- length(data$traffic_volume)
  coef_beta1 <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients[[2]]
}
})
beta1_ci = parSapply(cl, seq_len(B),singleBoots)
beta1_ci %>% quantile(c(0.025,0.975))
stopCluster(cl)

## Using parallelization to bootstrap for CI of coefficient of clouds coverage.
cl = makeCluster(4)
B = 1000
coef_beta2 <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients[[3]]
}
singleBoots <- function(i){
  index = sample(x = seq_len(n), size = n, replace = TRUE)
  data_star = data[index,]
  coef_beta2(data_star)
}
clusterEvalQ(cl,{
  library(tidyverse)
  data <- read_csv("Metro_without_time_weather.csv")
  n <- length(data$traffic_volume)
  coef_beta2 <- function(data){
  model = lm(data$traffic_volume~data$temp+data$clouds_all)
  coefficient = model$coefficients[[3]]
}
})
beta1_ci = parSapply(cl, seq_len(B),singleBoots)
beta1_ci %>% quantile(c(0.025,0.975))
stopCluster(cl)
```