Traffic

Zhe Zhao(433707)

2022/5/24

## Introduction

The main target of this project is using regression methods of machine learning to build a model predicting the traffic on one of the highways for one-hourly intervals based on the training sample and generate predictions for all observations from the test sample.

## import   
library(dplyr)

## Warning: łĚĽ­°ü'dplyr'ĘÇÓĂR°ć±ľ4.1.3 Ŕ´˝¨ÔěµÄ

##   
## 载入程辑包：'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(ggplot2)

## Warning: łĚĽ­°ü'ggplot2'ĘÇÓĂR°ć±ľ4.1.3 Ŕ´˝¨ÔěµÄ

library(caret)

## 载入需要的程辑包：lattice

library(lattice)  
library(corrplot)

## corrplot 0.92 loaded

## Import data

Import data and check detailed information.

## read data  
traffic <- read.csv("C:/Users/zhaoz/Desktop/jiqixuexi/pro/traffic\_train.csv")  
  
## check detailed information  
head(traffic)

## date\_time weather\_general weather\_detailed clouds\_coverage\_pct temperature  
## 1 0 Clear sky is clear 1 11.5  
## 2 1 Clear sky is clear 1 10.3  
## 3 2 Clear sky is clear 1 8.0  
## 4 3 Clear sky is clear 1 7.9  
## 5 4 Clear sky is clear 1 6.4  
## 6 5 Clear sky is clear 1 5.5  
## rain\_mm snow\_mm traffic  
## 1 0 0 508  
## 2 0 0 323  
## 3 0 0 274  
## 4 0 0 372  
## 5 0 0 812  
## 6 0 0 2720

str(traffic)

## 'data.frame': 29701 obs. of 8 variables:  
## $ date\_time : int 0 1 2 3 4 5 6 8 9 12 ...  
## $ weather\_general : chr "Clear" "Clear" "Clear" "Clear" ...  
## $ weather\_detailed : chr "sky is clear" "sky is clear" "sky is clear" "sky is clear" ...  
## $ clouds\_coverage\_pct: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ temperature : num 11.5 10.3 8 7.9 6.4 5.5 5.1 5 9.3 18.8 ...  
## $ rain\_mm : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ snow\_mm : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ traffic : int 508 323 274 372 812 2720 5674 6512 5473 5096 ...

## Data Cleaning

“Traffic” is the dependent variable, so we need to remove the value equal to 0 to get the MAPE value, firstly we check if there are “0”s.

## check if there is any "0"s in "traffic"   
any(traffic$traffic=="0")

## [1] TRUE

Yes, there are “0” values, so We have to remove these rows with “traffic” equal to 0.

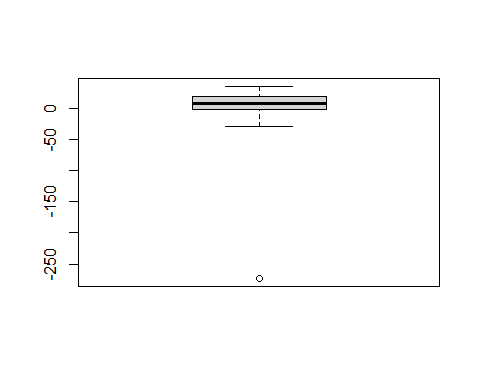
## 0s transform into NA  
traffic$traffic[which(traffic$traffic=="0")] <- NA  
  
## omit NA value  
traffic <- na.omit(traffic)  
  
##check again  
any(is.na(traffic))

## [1] FALSE

Now there is no rows with “traffic” equal to 0.

Check outliers of variable temperature.

## Find out if there are outliers  
boxplot(traffic$temperature)



I also find there are some with temperature at absolute zero, we need to remove outliers.

We have to remove these rows with “temperature” equal to -273.1.

## 0s transform into NA  
traffic$temperature[which(traffic$temperature=="-273.1")] <- NA  
  
## omit NA value  
traffic <- na.omit(traffic)  
  
##check again  
any(is.na(traffic))

## [1] FALSE

Now there are no outliers in variable temperature.

Lests set training sample as 70% and test sample as 30%.

## set training sample and test sample   
  
set.seed(20030329) #set Random seed  
traffic\_which\_training <- createDataPartition(traffic$traffic,  
 p = 0.7, list = FALSE) #set 70% of data as training data  
   
traffic\_train <-traffic[c(traffic\_which\_training),] #training data  
traffic\_test <-traffic[-c(traffic\_which\_training),] #test data  
  
## save data  
save(list = c("traffic\_train",  
 "traffic\_test"),  
 file = "C:/Users/zhaoz/Desktop/jiqixuexi/pro/traffic\_train\_test.RData")

Check again if there are any NA values in training sample.

## check if there are any NA values in training sample   
any(is.na(traffic\_train))

## [1] FALSE

No Na value in training sample.

Have a look at basic statistics of training sample and testing sample, We prefer that the results are similar.

summary(traffic\_train$traffic)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1 1167 3313 3230 4919 7256

summary(traffic\_test$traffic)

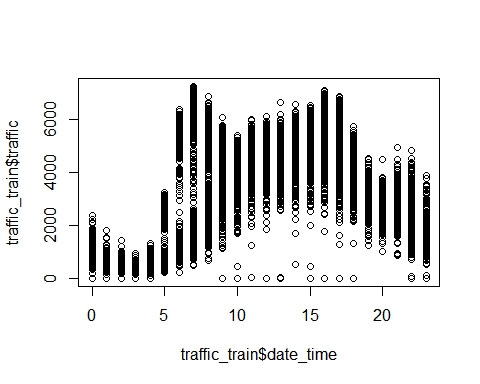
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2 1164 3313 3233 4919 7263

All these statistics are similar, training sample and testing sample could be acceptable.

## Data Exploration

In our experience, traffic is affected by time and we can visualize the result visually.

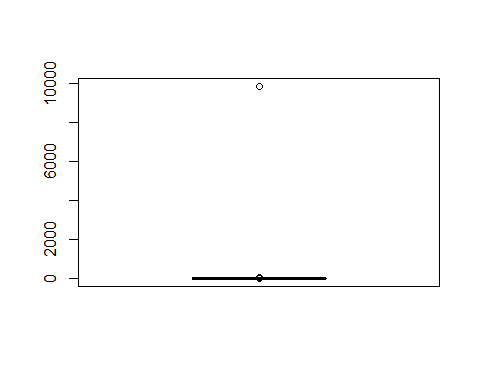
plot(traffic\_train$date\_time,traffic\_train$traffic )



We can see high traffic volume during the day and low traffic volume at night. p.s. date\_time(hour)

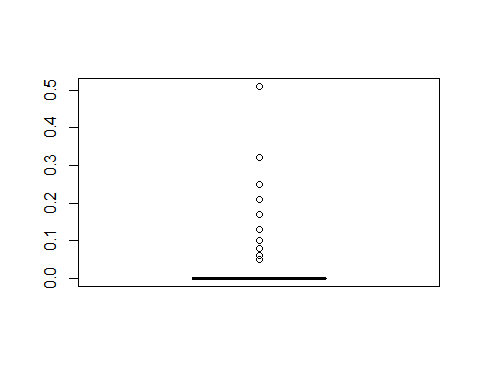
Check outliers of variable rain\_mm, .

## Find out if there are outliers  
boxplot(traffic\_train$rain\_mm)



Check outliers of variable snow\_mm.

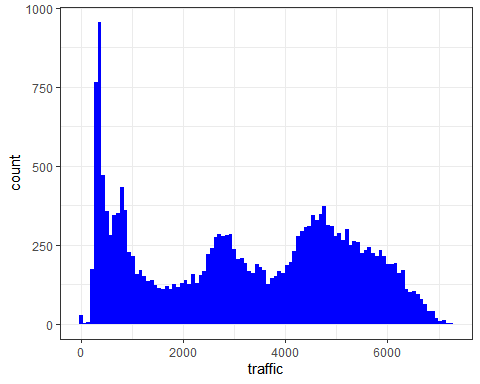
# Find out if there are outliers  
boxplot(traffic\_train$snow\_mm)



## Data Transform

Check the distribution of the dependent variable “traffic”.

# check the distribution (log transformation) of the dependent variable "traffic"  
ggplot(traffic\_train,  
 aes(x = traffic)) +  
 geom\_histogram(fill = "blue",  
 bins = 100) +  
 theme\_bw()



Find character variables.

## check character variables  
traffic\_train\_character\_vars <-   
 sapply(traffic\_train, is.character) %>%   
 which() %>%   
 names()  
  
## result  
traffic\_train\_character\_vars

## [1] "weather\_general" "weather\_detailed"

There are two character variables, sort them.

## sort 2 character variables  
sapply(traffic\_train[, traffic\_train\_character\_vars],   
 function(x)   
 unique(x) %>%   
 length()) %>%   
 sort()

## weather\_general weather\_detailed   
## 11 36

Transfer character variables to factor, find the levels of them.

## transfer character variables to factor  
traffic\_train$weather\_general <- factor(traffic\_train$weather\_general)  
traffic\_train$weather\_detailed <- factor(traffic\_train$weather\_detailed)  
  
## check levels of weather general and weather detailed  
levels(traffic\_train$weather\_general)

## [1] "Clear" "Clouds" "Drizzle" "Fog" "Haze"   
## [6] "Mist" "Rain" "Smoke" "Snow" "Squall"   
## [11] "Thunderstorm"

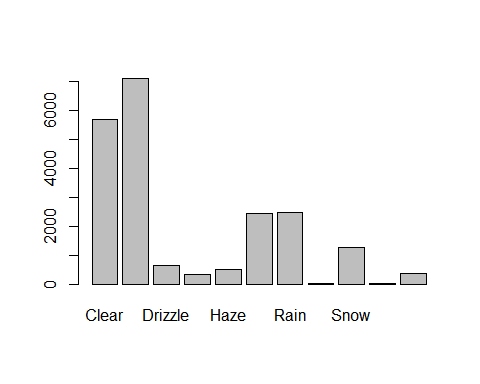
levels(traffic\_train$weather\_detailed)

## [1] "broken clouds" "drizzle"   
## [3] "few clouds" "fog"   
## [5] "freezing rain" "haze"   
## [7] "heavy intensity drizzle" "heavy intensity rain"   
## [9] "heavy snow" "light intensity drizzle"   
## [11] "light intensity shower rain" "light rain"   
## [13] "light rain and snow" "light shower snow"   
## [15] "light snow" "mist"   
## [17] "moderate rain" "overcast clouds"   
## [19] "proximity shower rain" "proximity thunderstorm"   
## [21] "proximity thunderstorm with drizzle" "proximity thunderstorm with rain"   
## [23] "scattered clouds" "shower drizzle"   
## [25] "sky is clear" "sleet"   
## [27] "smoke" "snow"   
## [29] "squalls" "thunderstorm"   
## [31] "thunderstorm with drizzle" "thunderstorm with heavy rain"   
## [33] "thunderstorm with light drizzle" "thunderstorm with light rain"   
## [35] "thunderstorm with rain" "very heavy rain"

## detailed information  
table(traffic\_train$weather\_general)

##   
## Clear Clouds Drizzle Fog Haze Mist   
## 5661 7081 647 339 515 2429   
## Rain Smoke Snow Squall Thunderstorm   
## 2467 10 1272 4 349

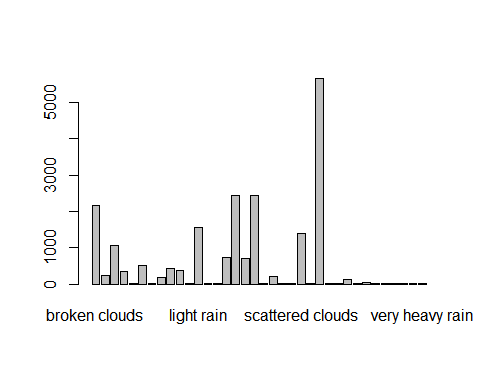
plot(traffic\_train$weather\_general)



table(traffic\_train$weather\_detailed)

##   
## broken clouds drizzle   
## 2162 241   
## few clouds fog   
## 1069 339   
## freezing rain haze   
## 1 515   
## heavy intensity drizzle heavy intensity rain   
## 24 184   
## heavy snow light intensity drizzle   
## 415 381   
## light intensity shower rain light rain   
## 3 1551   
## light rain and snow light shower snow   
## 1 2   
## light snow mist   
## 737 2429   
## moderate rain overcast clouds   
## 694 2451   
## proximity shower rain proximity thunderstorm   
## 24 216   
## proximity thunderstorm with drizzle proximity thunderstorm with rain   
## 8 13   
## scattered clouds shower drizzle   
## 1399 1   
## sky is clear sleet   
## 5661 1   
## smoke snow   
## 10 116   
## squalls thunderstorm   
## 4 37   
## thunderstorm with drizzle thunderstorm with heavy rain   
## 2 28   
## thunderstorm with light drizzle thunderstorm with light rain   
## 1 24   
## thunderstorm with rain very heavy rain   
## 20 10

plot(traffic\_train$weather\_detailed)



## transfer int variables to num  
traffic\_train$date\_time <- as.numeric(traffic\_train$date\_time)  
traffic\_train$clouds\_coverage\_pct <- as.numeric(traffic\_train$clouds\_coverage\_pct)

Label variables weather\_general and weather\_detailed.

## label weather\_general  
traffic\_train$weather\_general<-factor(traffic\_train$weather\_general,  
 level=c("Clear","Clouds","Squall","Smoke","Fog","Thunderstorm","Haze",   
 "Drizzle","Snow", "Mist", "Rain"),  
 labels = c(1:11))

## label weather\_detailed  
traffic\_train$weather\_detailed<-factor(traffic\_train$weather\_detailed,  
 level=c("broken clouds", "drizzle", "few clouds",  
 "Mist", "Rain","fog" ,"freezing rain", "haze" , "heavy intensity drizzle", "heavy intensity rain",   
 "heavy snow" , "light intensity drizzle"   
 , "light intensity shower rain" , "light rain"   
 , "light rain and snow", "light shower snow"   
 ,"light snow"   
 , "moderate rain", "overcast clouds"   
 , "proximity shower rain" , "proximity thunderstorm"   
 , "proximity thunderstorm with drizzle", "proximity thunderstorm with rain"   
 , "scattered clouds", "shower drizzle"   
 , "sky is clear" , "smoke"   
 , "snow" , "squalls"   
 , "thunderstorm" , "thunderstorm with drizzle"   
 , "thunderstorm with heavy rain" , "thunderstorm with light drizzle"   
 , "thunderstorm with light rain" , "thunderstorm with rain"   
 , "very heavy rain" ),  
 labels = c(1:36))

str(traffic\_train)

## 'data.frame': 20774 obs. of 8 variables:  
## $ date\_time : num 0 1 2 3 4 5 6 9 12 14 ...  
## $ weather\_general : Factor w/ 11 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weather\_detailed : Factor w/ 36 levels "1","2","3","4",..: 26 26 26 26 26 26 26 26 26 26 ...  
## $ clouds\_coverage\_pct: num 1 1 1 1 1 1 1 1 1 1 ...  
## $ temperature : num 11.5 10.3 8 7.9 6.4 5.5 5.1 9.3 18.8 21.2 ...  
## $ rain\_mm : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ snow\_mm : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ traffic : int 508 323 274 372 812 2720 5674 5473 5096 5335 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:10] 11884 11885 11886 11887 11932 11933 11934 11935 11936 11937  
## ..- attr(\*, "names")= chr [1:10] "11884" "11885" "11886" "11887" ...

Transfer variable weather\_general and weather\_detailed into numeric variables, it will be easy to do correlation test.

traffic\_train$weather\_detailed <- as.numeric(traffic\_train$weather\_detailed)  
traffic\_train$weather\_general <- as.numeric(traffic\_train$weather\_detailed)  
  
str(traffic\_train)

## 'data.frame': 20774 obs. of 8 variables:  
## $ date\_time : num 0 1 2 3 4 5 6 9 12 14 ...  
## $ weather\_general : num 26 26 26 26 26 26 26 26 26 26 ...  
## $ weather\_detailed : num 26 26 26 26 26 26 26 26 26 26 ...  
## $ clouds\_coverage\_pct: num 1 1 1 1 1 1 1 1 1 1 ...  
## $ temperature : num 11.5 10.3 8 7.9 6.4 5.5 5.1 9.3 18.8 21.2 ...  
## $ rain\_mm : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ snow\_mm : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ traffic : int 508 323 274 372 812 2720 5674 5473 5096 5335 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:10] 11884 11885 11886 11887 11932 11933 11934 11935 11936 11937  
## ..- attr(\*, "names")= chr [1:10] "11884" "11885" "11886" "11887" ...

Find numeric variables for correlation test.

## find numeric variables  
traffic\_numeric\_vars <-   
 # check if variable is numeric  
 sapply(traffic\_train, is.numeric) %>%   
 # select those which are  
 which() %>%   
 # and keep just their names  
 names()  
## result  
traffic\_numeric\_vars

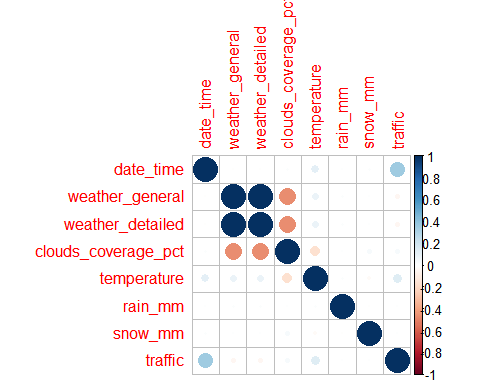
## [1] "date\_time" "weather\_general" "weather\_detailed"   
## [4] "clouds\_coverage\_pct" "temperature" "rain\_mm"   
## [7] "snow\_mm" "traffic"

Check correlations between variables, and draw correlation plot.

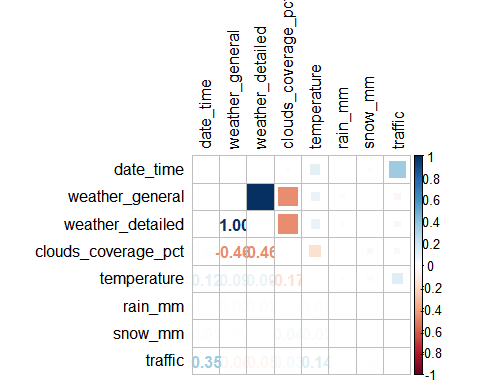
# mutually correlated (irrelevant) variables  
# calculate correlations between variables to identify redundant features  
  
traffic\_train\_correlations <-   
 cor(traffic\_train[, traffic\_numeric\_vars],  
 use = "pairwise.complete.obs")  
traffic\_train\_correlations

## date\_time weather\_general weather\_detailed  
## date\_time 1.000000000 -0.005314650 -0.005314650  
## weather\_general -0.005314650 1.000000000 1.000000000  
## weather\_detailed -0.005314650 1.000000000 1.000000000  
## clouds\_coverage\_pct 0.016931794 -0.460456189 -0.460456189  
## temperature 0.116174956 0.089308001 0.089308001  
## rain\_mm 0.005131944 0.015634544 0.015634544  
## snow\_mm 0.012255935 0.008636438 0.008636438  
## traffic 0.350768119 -0.045461236 -0.045461236  
## clouds\_coverage\_pct temperature rain\_mm snow\_mm  
## date\_time 0.016931794 0.11617496 0.0051319440 0.0122559353  
## weather\_general -0.460456189 0.08930800 0.0156345444 0.0086364376  
## weather\_detailed -0.460456189 0.08930800 0.0156345444 0.0086364376  
## clouds\_coverage\_pct 1.000000000 -0.16699062 0.0060702053 0.0351714756  
## temperature -0.166990618 1.00000000 0.0134931717 -0.0247742579  
## rain\_mm 0.006070205 0.01349317 1.0000000000 -0.0002920342  
## snow\_mm 0.035171476 -0.02477426 -0.0002920342 1.0000000000  
## traffic 0.033635174 0.13955528 0.0074336748 0.0004101275  
## traffic  
## date\_time 0.3507681193  
## weather\_general -0.0454612356  
## weather\_detailed -0.0454612356  
## clouds\_coverage\_pct 0.0336351738  
## temperature 0.1395552756  
## rain\_mm 0.0074336748  
## snow\_mm 0.0004101275  
## traffic 1.0000000000

## result  
corrplot(traffic\_train\_correlations)



## correlation plot  
corrplot.mixed(traffic\_train\_correlations,  
 upper = "square",  
 lower = "number",  
 tl.col = "black", # color of labels (variable names)  
 tl.pos = "lt") # position of labels (lt = left and top)



Chose only variables “date\_time”, “traffic” and “temperature”

Creat new training sample with variables “date\_time”, “traffic” and “temperature” for next steps.

traffic\_train\_pro <- traffic\_train[,c("traffic", "date\_time", "temperature")]  
  
traffic\_train\_pro$date\_time <- as.factor(traffic\_train\_pro$date\_time)

## Linear regression model

## Linear regression model   
traffic\_lm1 <- lm(traffic ~ date\_time + temperature, # formula  
 data = traffic\_train\_pro)  
## output  
summary(traffic\_lm1)

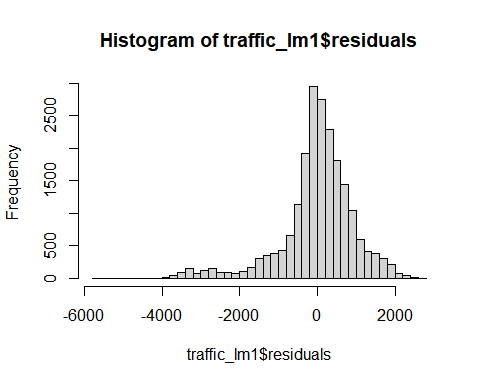
##   
## Call:  
## lm(formula = traffic ~ date\_time + temperature, data = traffic\_train\_pro)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5686.0 -310.3 75.9 523.0 2713.7   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 761.2706 32.6075 23.346 < 2e-16 \*\*\*  
## date\_time1 -284.3174 45.8355 -6.203 5.64e-10 \*\*\*  
## date\_time2 -409.8861 45.8888 -8.932 < 2e-16 \*\*\*  
## date\_time3 -432.7888 45.8082 -9.448 < 2e-16 \*\*\*  
## date\_time4 -113.4122 45.1660 -2.511 0.012 \*   
## date\_time5 1192.8951 45.5414 26.194 < 2e-16 \*\*\*  
## date\_time6 3295.9770 45.4655 72.494 < 2e-16 \*\*\*  
## date\_time7 3848.3285 45.7584 84.101 < 2e-16 \*\*\*  
## date\_time8 3738.5066 45.3694 82.401 < 2e-16 \*\*\*  
## date\_time9 3530.5393 46.5569 75.833 < 2e-16 \*\*\*  
## date\_time10 3342.5743 45.2687 73.839 < 2e-16 \*\*\*  
## date\_time11 3629.5708 46.3902 78.240 < 2e-16 \*\*\*  
## date\_time12 3865.5062 46.2813 83.522 < 2e-16 \*\*\*  
## date\_time13 3879.9570 46.5716 83.312 < 2e-16 \*\*\*  
## date\_time14 4099.2993 45.7524 89.597 < 2e-16 \*\*\*  
## date\_time15 4379.2829 46.1968 94.796 < 2e-16 \*\*\*  
## date\_time16 4769.6657 45.8757 103.969 < 2e-16 \*\*\*  
## date\_time17 4436.7097 46.1674 96.101 < 2e-16 \*\*\*  
## date\_time18 3403.3182 46.1795 73.698 < 2e-16 \*\*\*  
## date\_time19 2413.0120 46.2104 52.218 < 2e-16 \*\*\*  
## date\_time20 1966.5167 45.9738 42.775 < 2e-16 \*\*\*  
## date\_time21 1827.5231 45.9942 39.734 < 2e-16 \*\*\*  
## date\_time22 1346.2881 46.0256 29.251 < 2e-16 \*\*\*  
## date\_time23 630.1230 45.4021 13.879 < 2e-16 \*\*\*  
## temperature 7.1527 0.5285 13.534 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 953.7 on 20749 degrees of freedom  
## Multiple R-squared: 0.7701, Adjusted R-squared: 0.7698   
## F-statistic: 2896 on 24 and 20749 DF, p-value: < 2.2e-16

Residuals

head(traffic\_lm1$residuals)

## 1 2 3 4 5 6   
## -335.52618 -227.62553 -134.60581 -12.98779 118.36458 726.49467

hist(traffic\_lm1$residuals, breaks = 30)



# MSE is basically the average square residual  
  
mean(traffic\_lm1$residuals^2)

## [1] 908531.5

# MAE is the average absolute residual  
  
mean(abs(traffic\_lm1$residuals))

## [1] 655.0371

# similarly we can calculate Median Absolute Error  
  
median(abs(traffic\_lm1$residuals))

## [1] 428.1202

Write a simple function to summarize MSE, RMSE, MAE, MAPE, MedAE, MSLE and R2.

# Write a simple function to summarize all the errors and R2  
  
regressionMetrics <- function(real, predicted) {  
 # Mean Square Error  
 MSE <- mean((real - predicted)^2)  
 # Root Mean Square Error  
 RMSE <- sqrt(MSE)  
 # Mean Absolute Error  
 MAE <- mean(abs(real - predicted))  
 # Mean Absolute Percentage Error  
 MAPE <- mean(abs(real - predicted)/real)  
 # Median Absolute Error  
 MedAE <- median(abs(real - predicted))  
 # Mean Logarithmic Absolute Error  
 MSLE <- mean((log(1 + real) - log(1 + predicted))^2)  
 # R2  
 R2 <- cor(predicted, real)^2  
  
 result <- data.frame(MSE, RMSE, MAE, MAPE, MedAE, MSLE, R2)  
 return(result)  
}  
  
# lets apply it to our model  
regressionMetrics(real = traffic\_train\_pro$traffic,  
 predicted = predict(traffic\_lm1))

## MSE RMSE MAE MAPE MedAE MSLE R2  
## 1 908531.5 953.1692 655.0371 1.164449 428.1202 0.1945309 0.7701107

R^2 = 0.7701107 and MAPE = 1.164449, seems okay, so lets check how it’s performance in testing sample. Testing sample is names “traffic\_test”, separated from “traffic\_train”.

## Linear regression model for testing data  
traffic\_test$date\_time <- as.factor(traffic\_test$date\_time)  
traffic\_test$predicted <- traffic\_lm1 %>% predict(traffic\_test)  
  
## output  
print(bquote(R^2==.(R2(traffic\_test$predicted,traffic\_test$traffic))))

## R^2 == 0.767948620226271

We can check the MAPE value, it is even lower. We can also have a look at the value of R^2 = 0.767948620226271, still high!

## Conslusion

I built SVR and KNN model also, but the results are similar or worse, so i just chose Linear Regression model. The value of R square and MAPE Linear Regression model are not very good, but they are still better than the other models I used.