# Jianqi Liu

# Dylan Patel

# Aver McKay

# US\_ACCIDENT

一 Data preprocessing <https://www.kaggle.com/sobhanmoosavi/us-accidents#US_Accidents_Dec19.csv> The data came from the website of Kaggle, which covers 49 states of the United States. The data is collected from February 2016 to December 2019. We selected the last 15000 observations from the dataset for data analysis. 1.1 Var selection

library(MASS)

library(readr)

library(rpart)

library(kknn)

library(caret)

library(dplyr)

library(knitr)

setwd('~/Desktop/US\_ACCIDENT')

train <- readRDS('train\_.rds')

dim(train)#49 vars

## [1] 2243939 49

train1 = train[sample(nrow(train),15000),]

#1.Data preprocessing

#1.delete variables with 0 variance

train1 = train1[,!nearZeroVar(train1,saveMetrics = T)$nzv]

ncol(train1)#35

## [1] 35

#2. delete insignificant vars

xxnonsignificant <- c('ID','Source','TMC','Description','City',

'Country','Street')

train2 <- train1[,!names(train1) %in% xxnonsignificant]

#The summary of the dataset is shown below:

summary(train2)

## Severity Start\_Time End\_Time

## 0: 0 Min. :2016-02-09 05:27:20 Min. :2016-02-09 05:57:20

## 1: 4 1st Qu.:2017-03-24 16:40:40 1st Qu.:2017-03-24 21:04:36

## 2:9699 Median :2018-01-02 07:54:43 Median :2018-01-02 08:39:05

## 3:4780 Mean :2017-12-05 00:59:56 Mean :2017-12-05 03:23:28

## 4: 517 3rd Qu.:2018-08-21 13:11:06 3rd Qu.:2018-08-21 13:42:24

## Max. :2019-03-31 18:43:15 Max. :2019-03-31 20:01:06

##

## Start\_Lat Start\_Lng Distance(mi) Number

## Min. :24.93 Min. :-124.42 Min. : 0.0000 Min. : 1

## 1st Qu.:33.49 1st Qu.:-117.18 1st Qu.: 0.0000 1st Qu.: 838

## Median :35.82 Median : -88.15 Median : 0.0000 Median : 2697

## Mean :36.45 Mean : -94.90 Mean : 0.2874 Mean : 5706

## 3rd Qu.:40.53 3rd Qu.: -80.85 3rd Qu.: 0.0130 3rd Qu.: 6999

## Max. :48.96 Max. : -68.37 Max. :106.1400 Max. :240010

## NA's :9651

## Side County State

## Length:15000 Length:15000 Length:15000

## Class :character Class :character Class :character

## Mode :character Mode :character Mode :character

##

##

##

##

## Zipcode Timezone Airport\_Code

## Length:15000 Length:15000 Length:15000

## Class :character Class :character Class :character

## Mode :character Mode :character Mode :character

##

##

##

##

## Weather\_Timestamp Temperature(F) Wind\_Chill(F)

## Min. :2016-02-09 05:27:00 Min. :-27.90 Min. :-55.10

## 1st Qu.:2017-03-23 21:01:30 1st Qu.: 48.90 1st Qu.: 19.10

## Median :2017-12-29 20:48:00 Median : 64.00 Median : 28.50

## Mean :2017-12-04 04:16:03 Mean : 61.43 Mean : 26.03

## 3rd Qu.:2018-08-21 18:36:15 3rd Qu.: 75.90 3rd Qu.: 36.60

## Max. :2019-03-31 18:53:00 Max. :114.10 Max. : 45.20

## NA's :314 NA's :402 NA's :12429

## Humidity(%) Pressure(in) Wind\_Direction Wind\_Speed(mph)

## Min. : 4.00 Min. :29.03 Length:15000 Min. : 1.200

## 1st Qu.: 49.00 1st Qu.:29.92 Class :character 1st Qu.: 5.800

## Median : 68.00 Median :30.03 Mode :character Median : 8.100

## Mean : 65.89 Mean :30.04 Mean : 8.792

## 3rd Qu.: 85.00 3rd Qu.:30.15 3rd Qu.:11.500

## Max. :100.00 Max. :30.92 Max. :99.000

## NA's :417 NA's :377 NA's :2960

## Precipitation(in) Weather\_Condition Crossing Junction

## Min. :0.000 Length:15000 Mode :logical Mode :logical

## 1st Qu.:0.000 Class :character FALSE:14171 FALSE:13767

## Median :0.010 Mode :character TRUE :829 TRUE :1233

## Mean :0.056

## 3rd Qu.:0.040

## Max. :9.950

## NA's :13264

## Traffic\_Signal Sunrise\_Sunset Civil\_Twilight Nautical\_Twilight

## Mode :logical Length:15000 Length:15000 Length:15000

## FALSE:12579 Class :character Class :character Class :character

## TRUE :2421 Mode :character Mode :character Mode :character

##

##

##

##

## Astronomical\_Twilight

## Length:15000

## Class :character

## Mode :character

##

##

##

##

#2.turn into factor

train2$Timezone = as.factor(train2$Timezone)

train2$Wind\_Direction = as.factor(train2$Wind\_Direction)

train2$Weather\_Condition = as.factor(train2$Weather\_Condition)

train2$Civil\_Twilight = as.factor(train2$Civil\_Twilight)

train2$Sunrise\_Sunset = as.factor(train2$Sunrise\_Sunset)

train2$Nautical\_Twilight = as.factor(train2$Nautical\_Twilight)

train2$Astronomical\_Twilight=as.factor(train2$Astronomical\_Twilight)

1.2 Missing data

#missing num

sum(is.na(train2))

## [1] 40785

train2.na <- train2[!complete.cases(train2),]

head(train2.na,2)

## # A tibble: 2 x 29

## Severity Start\_Time End\_Time Start\_Lat Start\_Lng

## <fct> <dttm> <dttm> <dbl> <dbl>

## 1 3 2016-11-10 18:51:48 2016-11-10 19:36:48 28.2 -82.4

## 2 2 2017-10-11 07:30:56 2017-10-11 08:15:46 35.2 -80.9

## # … with 24 more variables: `Distance(mi)` <dbl>, Number <dbl>,

## # Side <chr>, County <chr>, State <chr>, Zipcode <chr>, Timezone <fct>,

## # Airport\_Code <chr>, Weather\_Timestamp <dttm>, `Temperature(F)` <dbl>,

## # `Wind\_Chill(F)` <dbl>, `Humidity(%)` <dbl>, `Pressure(in)` <dbl>,

## # Wind\_Direction <fct>, `Wind\_Speed(mph)` <dbl>,

## # `Precipitation(in)` <dbl>, Weather\_Condition <fct>, Crossing <lgl>,

## # Junction <lgl>, Traffic\_Signal <lgl>, Sunrise\_Sunset <fct>,

## # Civil\_Twilight <fct>, Nautical\_Twilight <fct>,

## # Astronomical\_Twilight <fct>

Only two our of the all observations contain missing data.

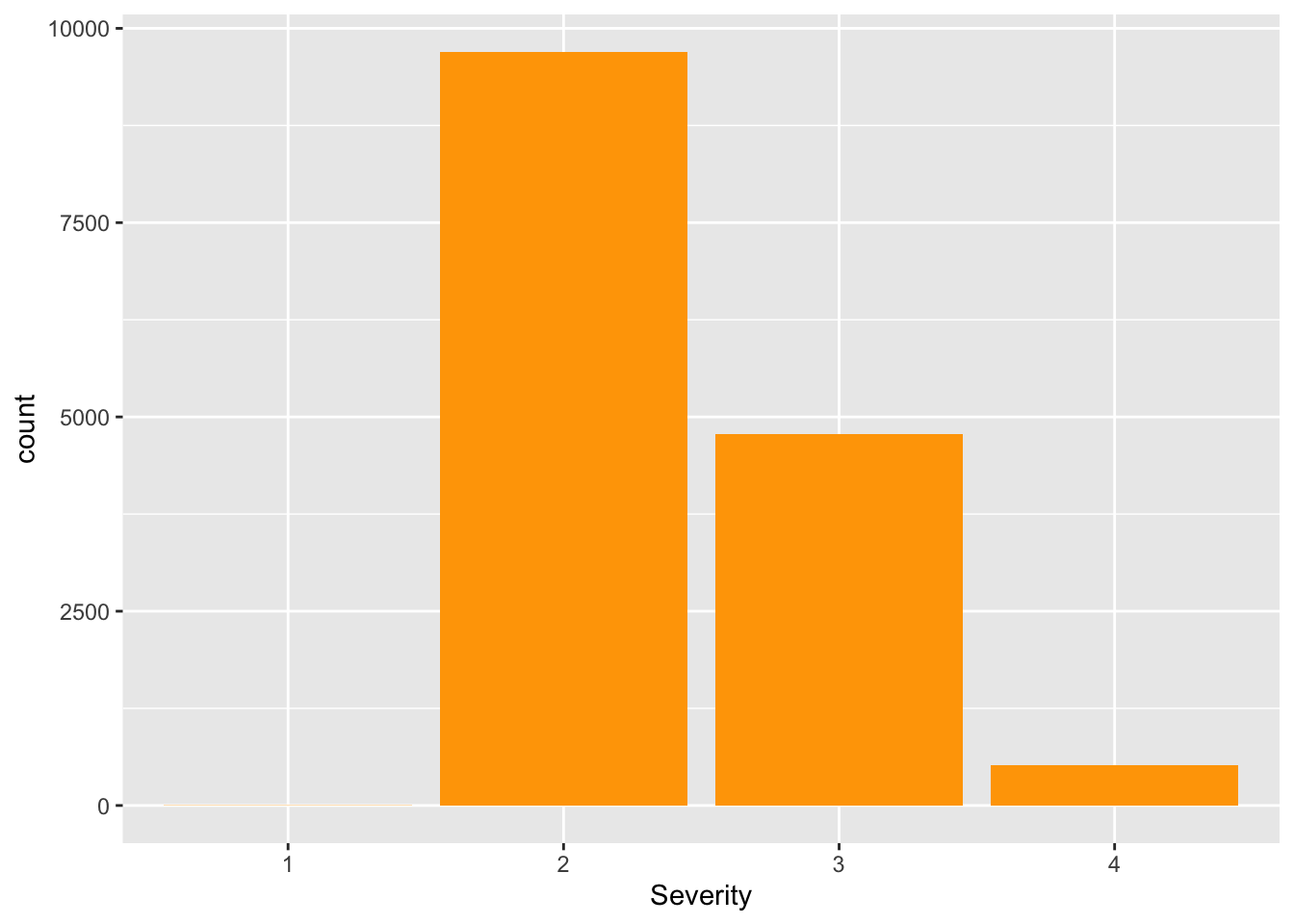
II visualization

we want to explore which variables will affect the severity of the accident. Firstly, the histogram of accident severity is made.

train$Severity <- as.factor(train$Severity)

v1 <- ggplot(train2,aes(Severity))+geom\_bar(position="identity",fill="#FFA500")

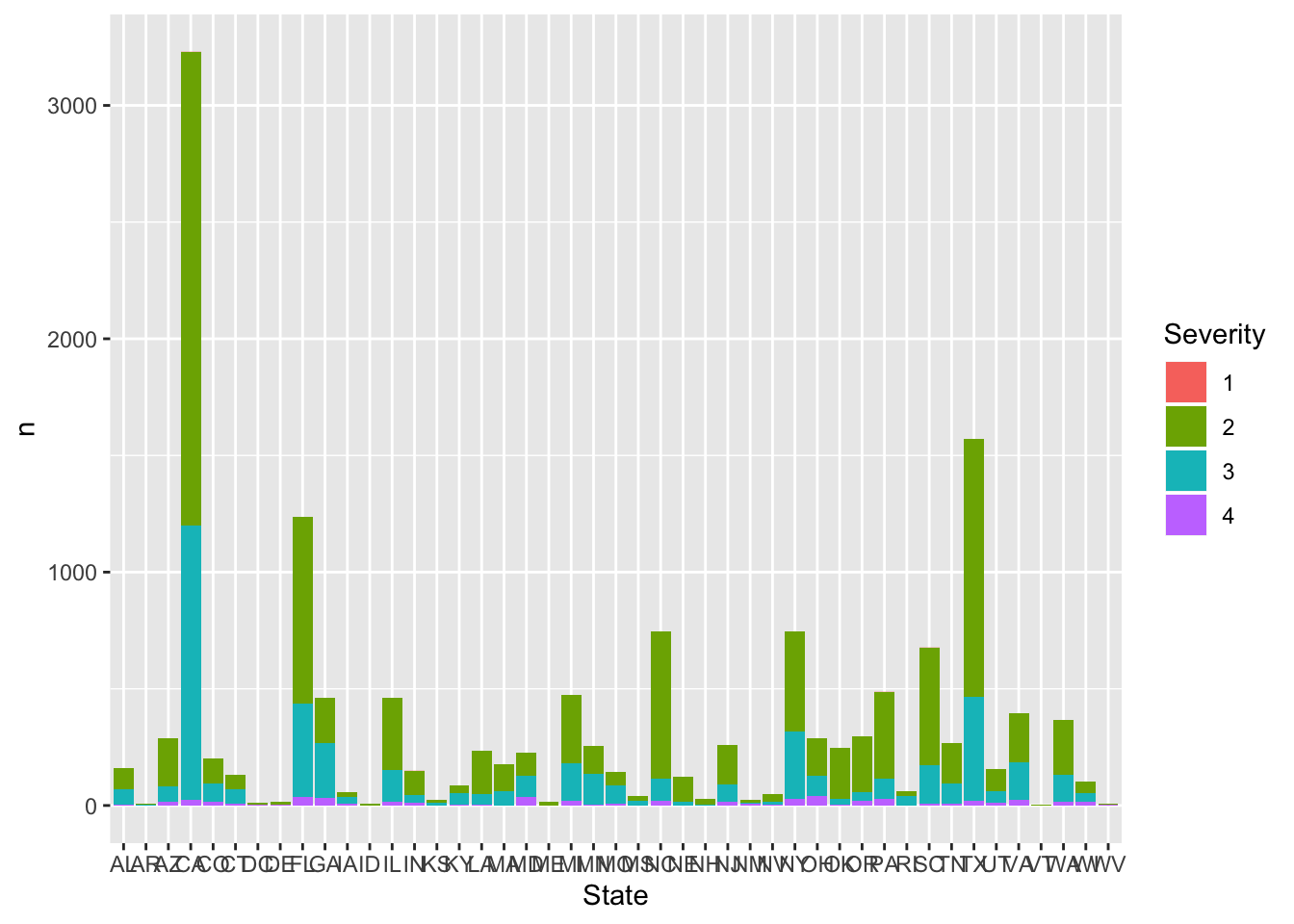
v1



It can be seen that the severity of level 2 accidents is the most frequent. Then, other factors are added to see which variables affect the severity of the accident from the visualization. We then take a look at the relationship with each state.

v2 <- train2 %>% group\_by(Severity,State) %>% count() %>% arrange(desc(n))

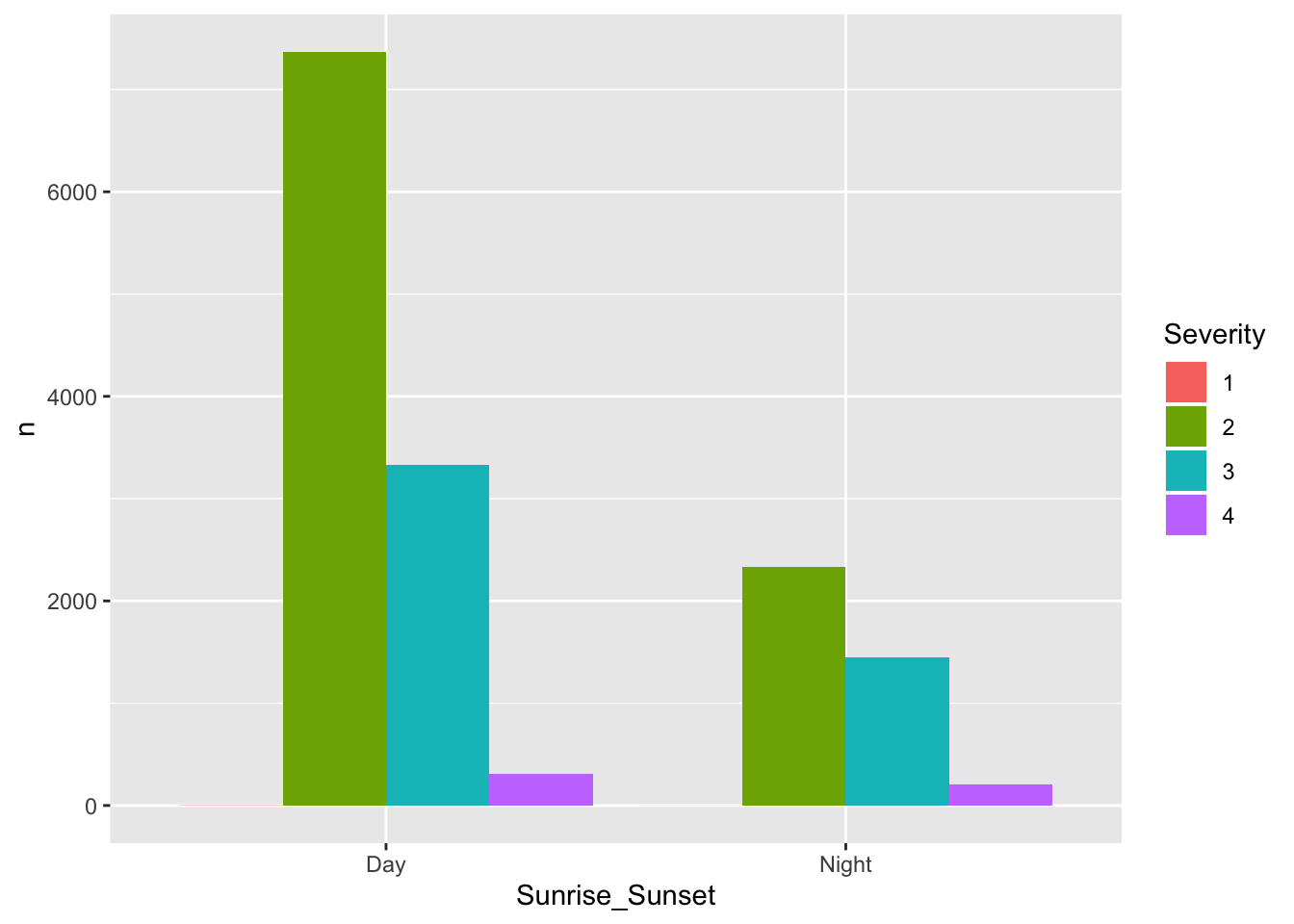
ggplot(v2, aes(fill=Severity,y=n, x=State))+ geom\_bar(stat = "identity")



Take a look at the Sunrise Sunset and Nautical Twilight columns of the dataset.

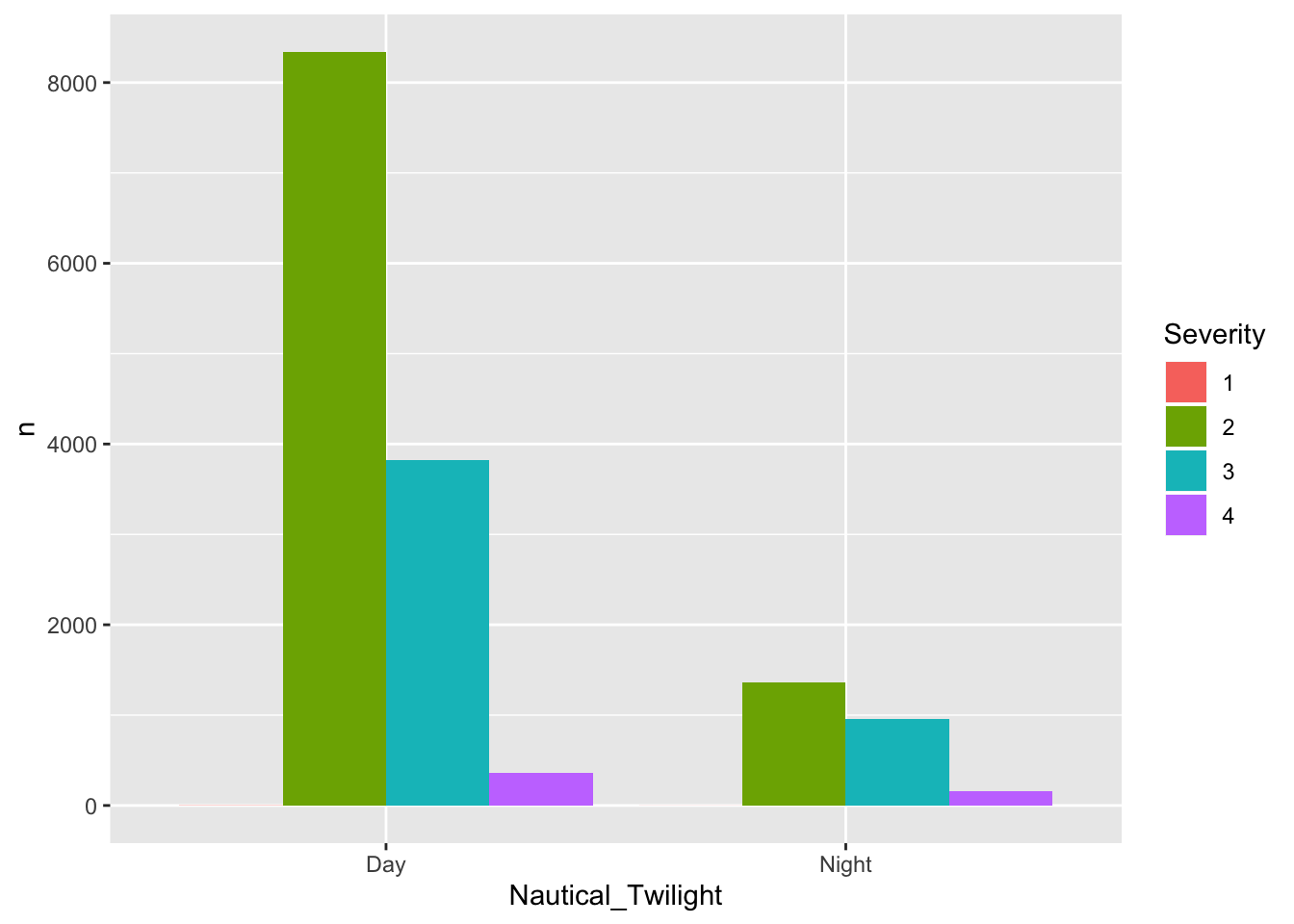
v3 <- train2 %>% group\_by(Severity,Sunrise\_Sunset) %>% count() %>% arrange(desc(n))

ggplot(v3, aes(fill=Severity,y=n, x=Sunrise\_Sunset))+ geom\_bar(position="dodge",stat = "identity")



v4 <- train2 %>% group\_by(Severity,Nautical\_Twilight) %>% count() %>% arrange(desc(n))

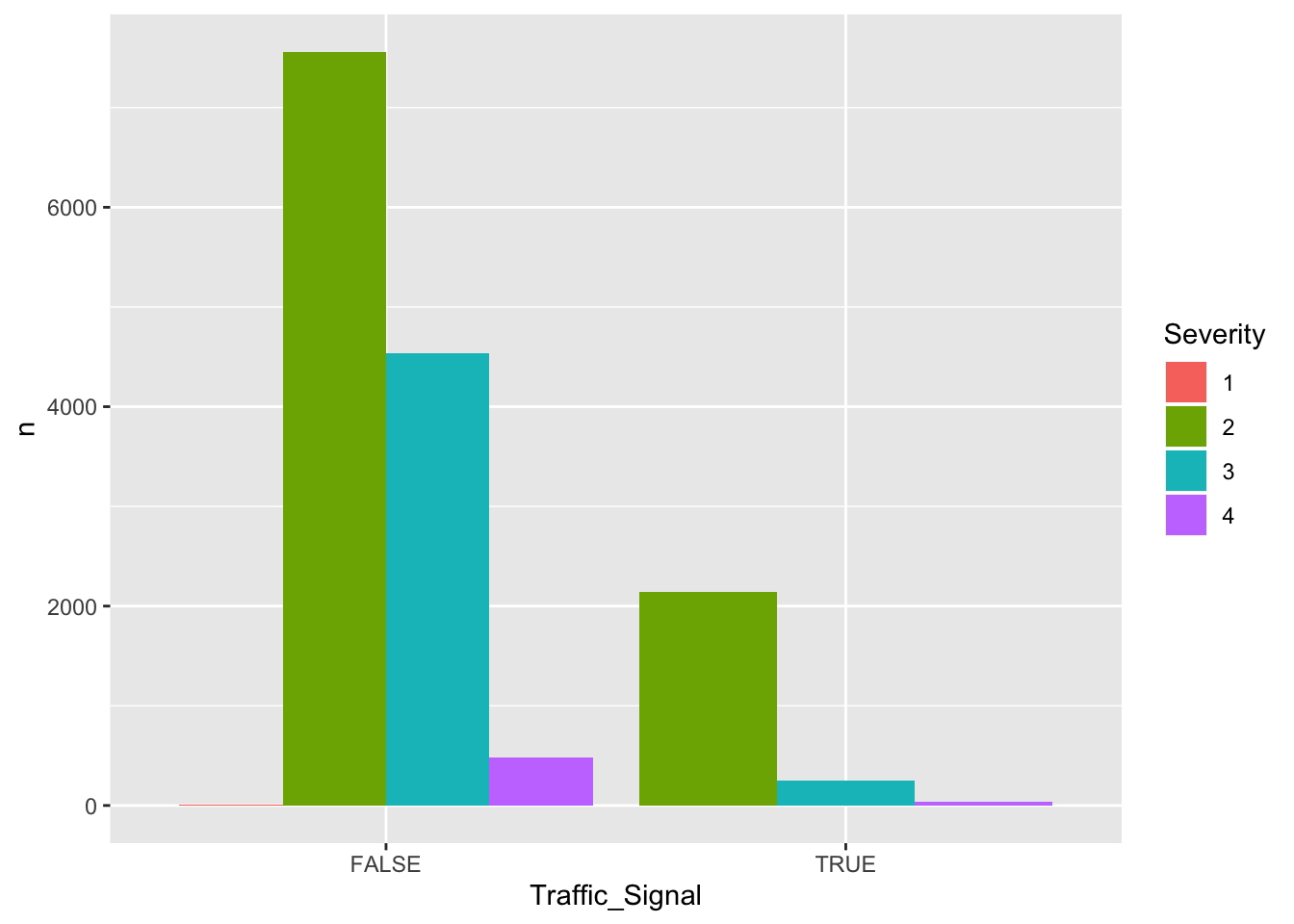
ggplot(v4, aes(fill=Severity,y=n, x=Nautical\_Twilight))+ geom\_bar(position="dodge",stat = "identity")



It can be seen from the visualization that the directivity of these two variables is similar. Thus we just include one of them into the model.

v5 <- train2 %>% group\_by(Severity,Traffic\_Signal) %>% count() %>% arrange(desc(n))

ggplot(v5, aes(fill=Severity,y=n, x=Traffic\_Signal))+ geom\_bar(position="dodge",stat = "identity")+theme()



Draw some time series plots

library(lubridate)

train2$Start\_Time <- as\_datetime(train2$Start\_Time)

train2$End\_Time <- as\_datetime(train2$End\_Time)

train2$accident\_time <- round(abs((train2$Start\_Time-train2$End\_Time)/60))

train2$Year <- as.numeric(format(train2$Start\_Time,format="%Y"))

train2$month <- as.numeric(format(train2$Start\_Time,format="%m"))

train2$date <- as.numeric(format(train2$Start\_Time,format="%d"))

data\_time <- train2 %>% dplyr::select(Year) %>% group\_by(Year) %>% summarise(num=n()) %>% arrange(num)

head(data\_time)

## # A tibble: 4 x 2

## Year num

## <dbl> <int>

## 1 2019 1490

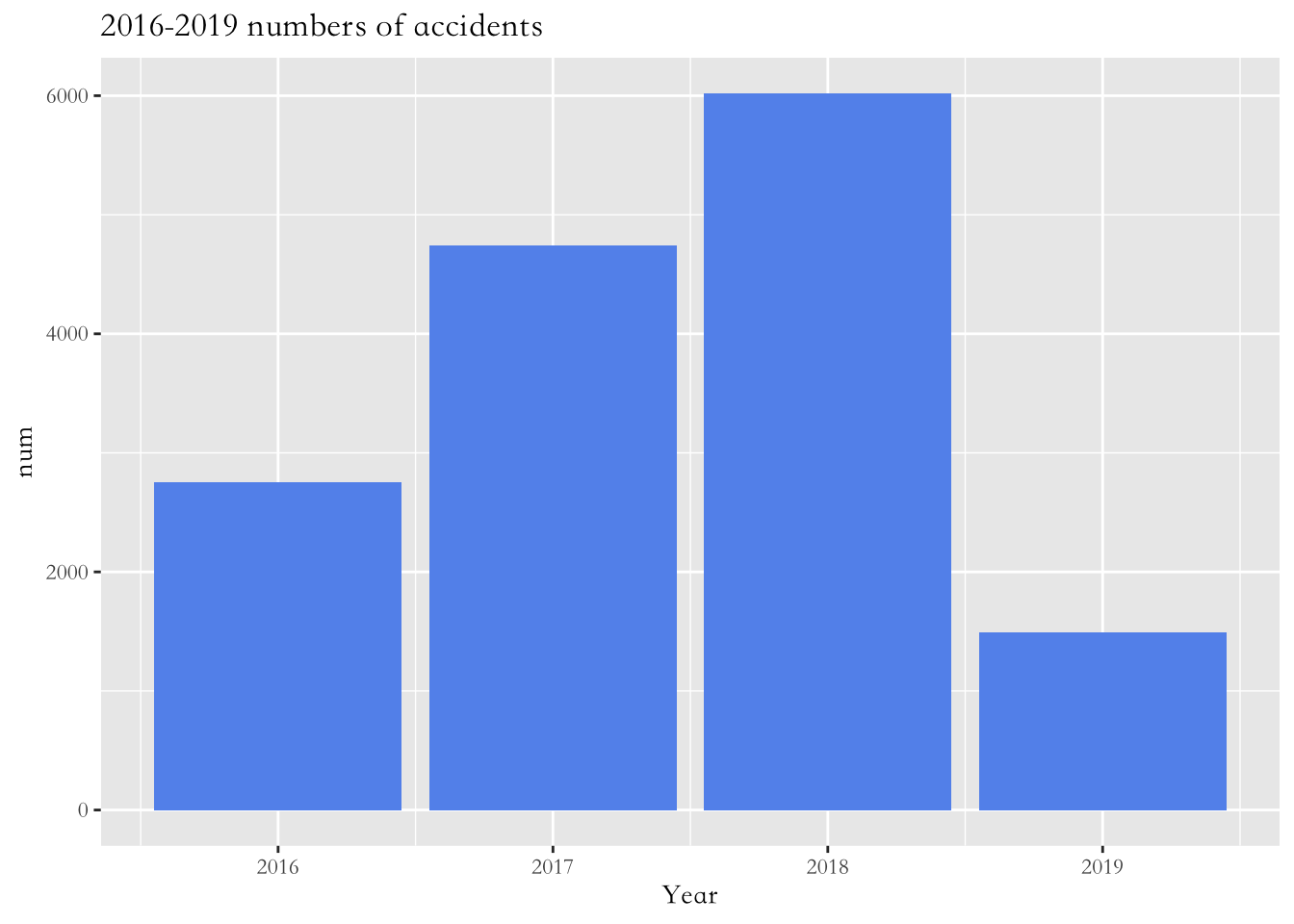
## 2 2016 2752

## 3 2017 4740

## 4 2018 6018

ggplot(data\_time,aes(x=Year, y=num)) + geom\_bar(stat="identity", fill = "#6495ED")+

ggtitle('2016-2019 numbers of accidents')+ theme\_grey(base\_family = "STKaiti")



accurate\_data\_time\_2018 <- train2 %>% group\_by(month,Year) %>% filter(Year == 2018)%>%summarise(total.count=n())

# Plot

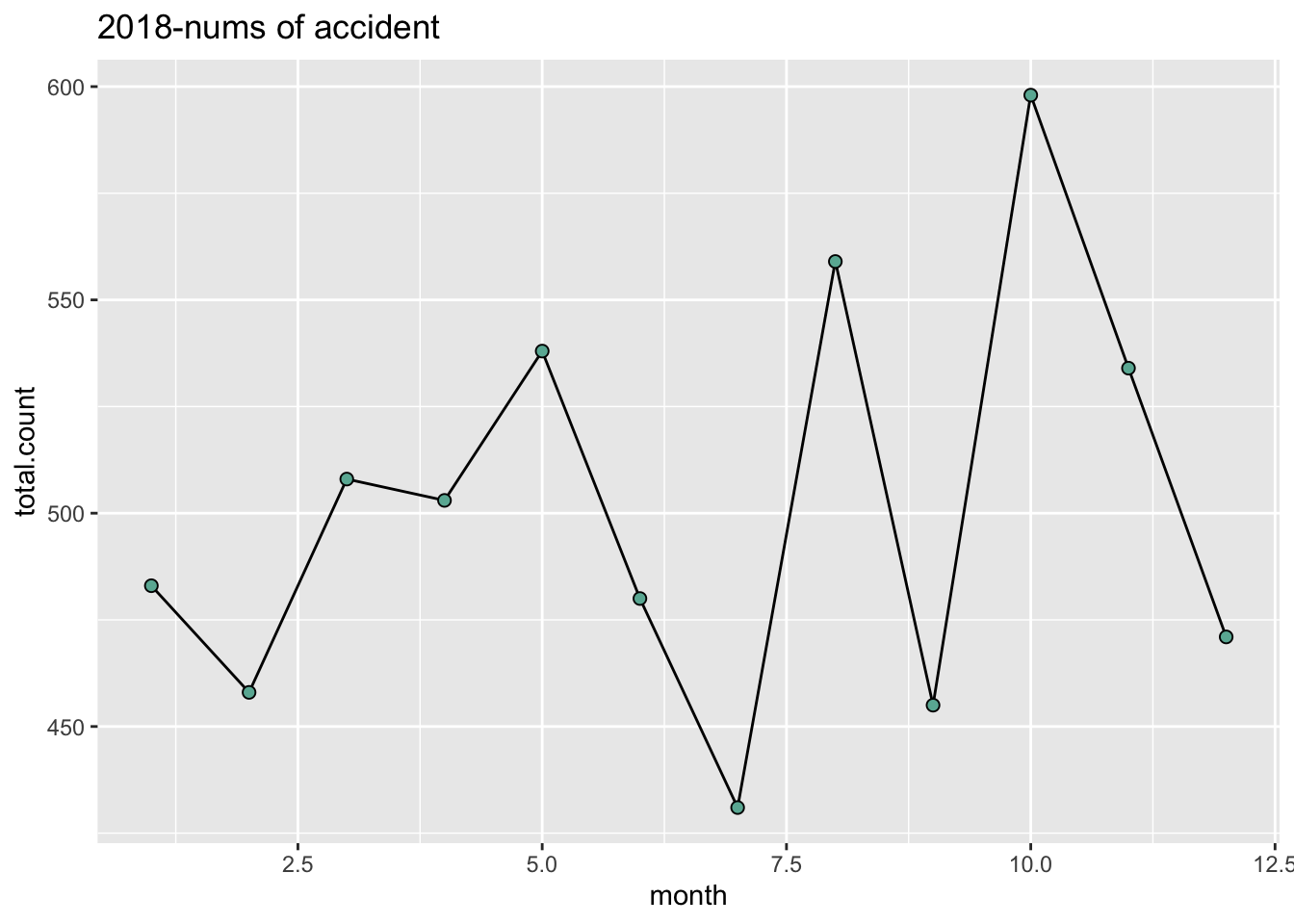
accurate\_data\_time\_2018 %>%

ggplot( aes(x=month, y=total.count)) +

geom\_line() +

geom\_point(shape=21, color="black", fill="#69b3a2", size=2) +

ggtitle("2018-nums of accident")

The number of accidents arrived the peak in October in 2018.

To figure out what factors influence the serivity of the accident, we conduct some data modelling. # Model(RF & KNN) ### 1. deal with the data

set.seed(1)

train2$County=as.factor(train2$County)

kernel.type=c("rectangular","triangular","epanechnikov","biweight","triweight","cos","inv","gaussian","rank","optimal")

model\_var = c('Severity','Start\_Lat','Start\_Lng','Distance(mi)','Side',

'Temperature(F)','Humidity(%)','Pressure(in)','Wind\_Direction',

'Wind\_Speed(mph)','Precipitation(in)','Crossing','Junction',

'Traffic\_Signal','Sunrise\_Sunset','accident\_time','Year','month','date')

model\_data = train2[,names(train2) %in% model\_var]

model\_data$Severity = model\_data$Severity %>% factor()

model\_data$Side = model\_data$Side %>% factor()

summary(model\_data)

## Severity Start\_Lat Start\_Lng Distance(mi) Side

## 1: 4 Min. :24.93 Min. :-124.42 Min. : 0.0000 L: 2735

## 2:9699 1st Qu.:33.49 1st Qu.:-117.18 1st Qu.: 0.0000 R:12265

## 3:4780 Median :35.82 Median : -88.15 Median : 0.0000

## 4: 517 Mean :36.45 Mean : -94.90 Mean : 0.2874

## 3rd Qu.:40.53 3rd Qu.: -80.85 3rd Qu.: 0.0130

## Max. :48.96 Max. : -68.37 Max. :106.1400

##

## Temperature(F) Humidity(%) Pressure(in) Wind\_Direction

## Min. :-27.90 Min. : 4.00 Min. :29.03 Calm :2470

## 1st Qu.: 48.90 1st Qu.: 49.00 1st Qu.:29.92 South :1134

## Median : 64.00 Median : 68.00 Median :30.03 West :1105

## Mean : 61.43 Mean : 65.89 Mean :30.04 North :1041

## 3rd Qu.: 75.90 3rd Qu.: 85.00 3rd Qu.:30.15 Variable: 788

## Max. :114.10 Max. :100.00 Max. :30.92 (Other) :8148

## NA's :402 NA's :417 NA's :377 NA's : 314

## Wind\_Speed(mph) Precipitation(in) Crossing Junction

## Min. : 1.200 Min. :0.000 Mode :logical Mode :logical

## 1st Qu.: 5.800 1st Qu.:0.000 FALSE:14171 FALSE:13767

## Median : 8.100 Median :0.010 TRUE :829 TRUE :1233

## Mean : 8.792 Mean :0.056

## 3rd Qu.:11.500 3rd Qu.:0.040

## Max. :99.000 Max. :9.950

## NA's :2960 NA's :13264

## Traffic\_Signal Sunrise\_Sunset accident\_time Year

## Mode :logical Day :11013 Length:15000 Min. :2016

## FALSE:12579 Night: 3987 Class :difftime 1st Qu.:2017

## TRUE :2421 Mode :numeric Median :2018

## Mean :2017

## 3rd Qu.:2018

## Max. :2019

##

## month date

## Min. : 1.000 Min. : 1.00

## 1st Qu.: 3.000 1st Qu.: 8.00

## Median : 7.000 Median :16.00

## Mean : 6.642 Mean :15.67

## 3rd Qu.:10.000 3rd Qu.:23.00

## Max. :12.000 Max. :31.00

##

model\_data = model\_data[complete.cases(model\_data),]

colnames(model\_data)=c('Severity','Start\_Lat','Start\_Lng','Distance','Side',

'Temperature','Humidity','Pressure','Wind\_Direction',

'Wind\_Speed','Precipitation','Crossing','Junction',

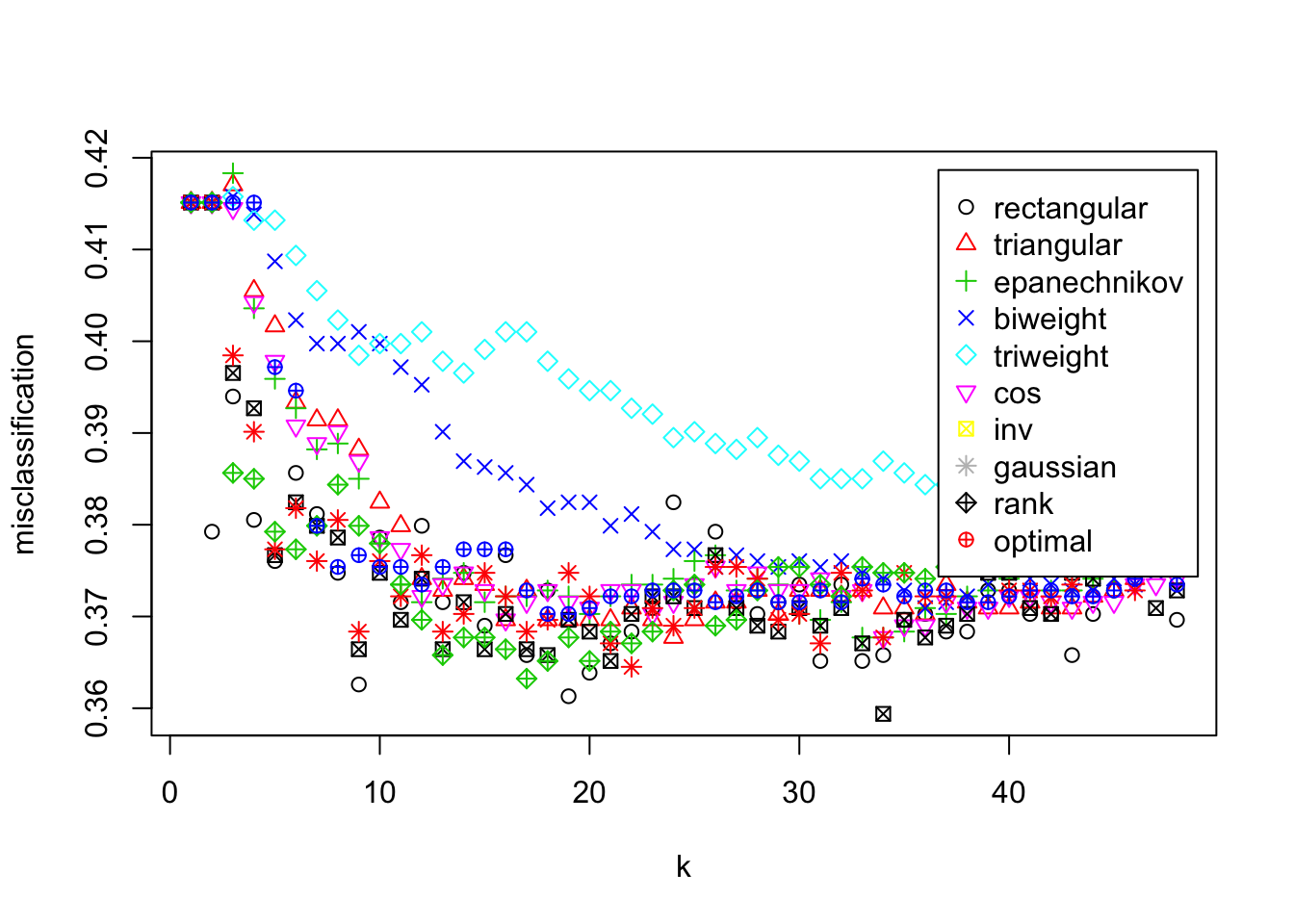
'Traffic\_Signal','Sunrise\_Sunset','accident\_time','Year','month','date')

## Data modeling

**1.knn** Description for knn:\_\_.

model.tkknn <- train.kknn(Severity~.,model\_data,kmax=48,kernel = kernel.type,distance=2,scale=T)

plot(model.tkknn)



model.tkknn #print out the best parameters

##

## Call:

## train.kknn(formula = Severity ~ ., data = model\_data, kmax = 48, distance = 2, kernel = kernel.type, scale = T)

##

## Type of response variable: nominal

## Minimal misclassification: 0.359385

## Best kernel: inv

## Best k: 34

#predict with the best parameter

set.seed(123)

testx <- model\_data[,-1]

train\_data = model\_data[sample(nrow(model\_data),nrow(model\_data)/3),]

test\_data = model\_data[-sample(nrow(model\_data),nrow(model\_data)/3),]

#split train and test dataset

model.kknn <- kknn(Severity~.,train\_data,test\_data,k=model.tkknn$best.parameters$k,scale=T,distance=2,kernel=model.tkknn$best.parameters$kernel)

confusionMatrix(model.kknn$fitted.values,test\_data$Severity)# Accuracy : 0.7347

## Confusion Matrix and Statistics

##

## Reference

## Prediction 1 2 3 4

## 1 0 0 0 0

## 2 0 562 134 32

## 3 0 88 207 3

## 4 0 0 0 15

##

## Overall Statistics

##

## Accuracy : 0.7531

## 95% CI : (0.7258, 0.7791)

## No Information Rate : 0.6244

## P-Value [Acc > NIR] : < 2.2e-16

##

## Kappa : 0.4735

##

## Mcnemar's Test P-Value : NA

##

## Statistics by Class:

##

## Class: 1 Class: 2 Class: 3 Class: 4

## Sensitivity NA 0.8646 0.6070 0.30000

## Specificity 1 0.5754 0.8700 1.00000

## Pos Pred Value NA 0.7720 0.6946 1.00000

## Neg Pred Value NA 0.7188 0.8197 0.96589

## Prevalence 0 0.6244 0.3276 0.04803

## Detection Rate 0 0.5399 0.1988 0.01441

## Detection Prevalence 0 0.6993 0.2863 0.01441

## Balanced Accuracy NA 0.7200 0.7385 0.65000

The best parameter is ：Best kernel: triangular；Best k: 15 The prediction performance of the model on the training set is accuracy: 0.7347, kappa coefficient: 0.4586

**2.RandomForest** Description for rf:\_\_.

library(randomForest)

train\_data$Severity=factor(train\_data$Severity)

rf.model <- randomForest(Severity~., data=train\_data)

rf.model

##

## Call:

## randomForest(formula = Severity ~ ., data = train\_data)

## Type of random forest: classification

## Number of trees: 500

## No. of variables tried at each split: 4

##

## OOB estimate of error rate: 38.85%

## Confusion matrix:

## 2 3 4 class.error

## 2 253 60 0 0.1916933

## 3 124 64 0 0.6595745

## 4 17 1 1 0.9473684

summary(train\_data)

## Severity Start\_Lat Start\_Lng Distance Side

## 2:313 Min. :25.68 Min. :-124.03 Min. : 0.0000 L: 81

## 3:188 1st Qu.:33.96 1st Qu.: -96.73 1st Qu.: 0.0000 R:439

## 4: 19 Median :38.71 Median : -83.61 Median : 0.0000

## Mean :37.78 Mean : -90.03 Mean : 0.4651

## 3rd Qu.:41.96 3rd Qu.: -79.11 3rd Qu.: 0.1143

## Max. :48.96 Max. : -70.24 Max. :30.9500

##

## Temperature Humidity Pressure Wind\_Direction

## Min. :-11.90 Min. : 28 Min. :29.03 North : 68

## 1st Qu.: 40.35 1st Qu.: 83 1st Qu.:29.85 South : 48

## Median : 53.10 Median : 89 Median :29.98 ESE : 37

## Mean : 52.91 Mean : 87 Mean :29.97 East : 36

## 3rd Qu.: 68.28 3rd Qu.: 94 3rd Qu.:30.09 NE : 30

## Max. : 89.10 Max. :100 Max. :30.76 NNE : 30

## (Other):271

## Wind\_Speed Precipitation Crossing Junction

## Min. : 3.50 Min. :0.00000 Mode :logical Mode :logical

## 1st Qu.: 5.80 1st Qu.:0.00000 FALSE:505 FALSE:476

## Median : 8.10 Median :0.01000 TRUE :15 TRUE :44

## Mean : 9.77 Mean :0.04796

## 3rd Qu.:12.70 3rd Qu.:0.04000

## Max. :34.50 Max. :1.68000

##

## Traffic\_Signal Sunrise\_Sunset accident\_time Year

## Mode :logical Day :376 Length:520 Min. :2016

## FALSE:455 Night:144 Class :difftime 1st Qu.:2017

## TRUE :65 Mode :numeric Median :2018

## Mean :2018

## 3rd Qu.:2018

## Max. :2019

##

## month date

## Min. : 1.000 Min. : 1.00

## 1st Qu.: 3.000 1st Qu.: 8.00

## Median : 6.000 Median :15.00

## Mean : 6.115 Mean :15.59

## 3rd Qu.:10.000 3rd Qu.:23.00

## Max. :12.000 Max. :31.00

##

pred.rf = predict(rf.model,test\_data)

confusionMatrix(pred.rf,test\_data$Severity)

## Confusion Matrix and Statistics

##

## Reference

## Prediction 1 2 3 4

## 1 0 0 0 0

## 2 0 570 134 32

## 3 0 80 207 2

## 4 0 0 0 16

##

## Overall Statistics

##

## Accuracy : 0.7618

## 95% CI : (0.7347, 0.7874)

## No Information Rate : 0.6244

## P-Value [Acc > NIR] : < 2.2e-16

##

## Kappa : 0.4897

##

## Mcnemar's Test P-Value : NA

##

## Statistics by Class:

##

## Class: 1 Class: 2 Class: 3 Class: 4

## Sensitivity NA 0.8769 0.6070 0.32000

## Specificity 1 0.5754 0.8829 1.00000

## Pos Pred Value NA 0.7745 0.7163 1.00000

## Neg Pred Value NA 0.7377 0.8218 0.96683

## Prevalence 0 0.6244 0.3276 0.04803

## Detection Rate 0 0.5476 0.1988 0.01537

## Detection Prevalence 0 0.7070 0.2776 0.01537

## Balanced Accuracy NA 0.7262 0.7449 0.66000

varimp <- varImp(rf.model) %>% as.data.frame()

varimp <- cbind(varimp,rownames(varimp))

varimp <- varimp %>% arrange(desc(Overall))

varimp

## Overall rownames(varimp)

## 1 37.2941413 Wind\_Direction

## 2 23.9117531 Start\_Lat

## 3 22.7286444 Start\_Lng

## 4 22.2868497 Pressure

## 5 20.3528316 Temperature

## 6 18.3583745 Humidity

## 7 17.4288999 date

## 8 16.3325629 Distance

## 9 15.9054018 Wind\_Speed

## 10 12.7537380 Precipitation

## 11 12.5968274 month

## 12 11.6579777 Side

## 13 11.5110293 accident\_time

## 14 6.5971769 Year

## 15 6.3275222 Traffic\_Signal

## 16 3.0796325 Sunrise\_Sunset

## 17 2.1422223 Junction

## 18 0.7572261 Crossing

The importance table of model variables is as above. Importance rate: wind\_Direction > start? Lat >… Randomforest performs better than knn.