Assignment 17

Anthony J. Kulowski

November 14, 2018

## Load Packages

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.1

library(caret)

## Loading required package: lattice

library(lattice)  
library(RANN)

## Warning: package 'RANN' was built under R version 3.5.1

## Read and Prepare Data

car\_data = read.csv("C:/Users/student/Documents/Bryant University/4 - Senior Year 2018-19/2018 FA/MATH 421/c2015.csv")  
df = read.csv("c2015.csv", header=TRUE)  
  
for(x in 1:ncol(df)) {  
 colnames(df)[x] <- tolower(colnames(df)[x])  
}  
  
df$year <- NULL  
df$st\_case <- NULL  
df$trav\_sp <- NULL  
df$age = as.numeric(df$age)  
df$month = as.factor(df$month)  
df$mod\_year = as.numeric(df$mod\_year)  
  
df = df[df$hour != 99,]  
df = df[df$sex != "Unknown",]  
df = df[df$sex != "Not Rep",]  
df = df[df$mod\_year != "Unknown",]  
df = df[df$mod\_year != "Not Rep",]  
df = df[df$deformed != "Unknown",]  
df = df[df$deformed != "Not Reported",]  
df = df[df$inj\_sev != "Unknown",]  
df = df[df$man\_coll != "Unknown",]  
df = df[df$owner != "Unknown",]  
df = df[df$route != "Unknown",]  
df = df[df$lgt\_cond != "Unknown",]  
df = df[df$lgt\_cond != "Not Reported",]  
df = df[df$weather != "Unkno",]  
df = df[df$weather != "Not R",]  
  
sum(is.na(df))

## [1] 465

## Change Target Variable Levels

levels(df$inj\_sev) = c("survived","died","survived","survived","survived","survived","survived","survived")

## Impute Missing Values

preProcess\_med <- preProcess(df, method='medianImpute')  
MedData <- predict(preProcess\_med, newdata = df)

## Check For Missing Values

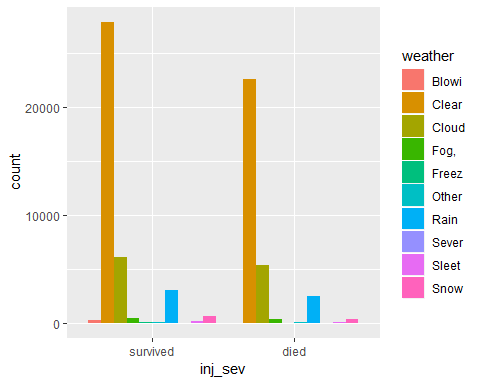
if(sum(is.na(MedData)) == 0) {  
 print('0')  
} else {  
 for(x in 1:ncol(MedData)) {  
 miss1 <- sum(is.na(MedData[x]))  
 print(colnames(MedData)[x])  
 print(miss1)  
 }  
 }

## [1] "0"

## Data Visualization

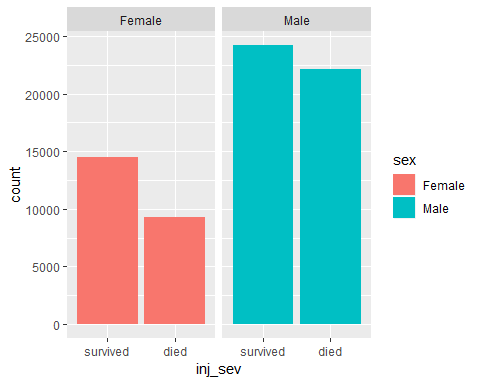
# Injury Severity by Weather

ggplot(data = MedData) + geom\_bar(mapping = aes(x = inj\_sev, fill = weather), position = "dodge")



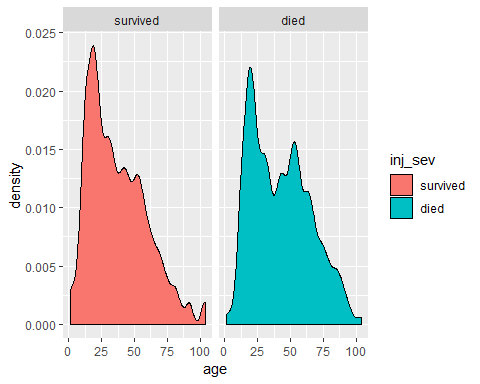
# Injury Severity by Gender

ggplot(data = MedData) + geom\_bar(mapping = aes(x = inj\_sev, fill = sex)) + facet\_wrap(~sex)



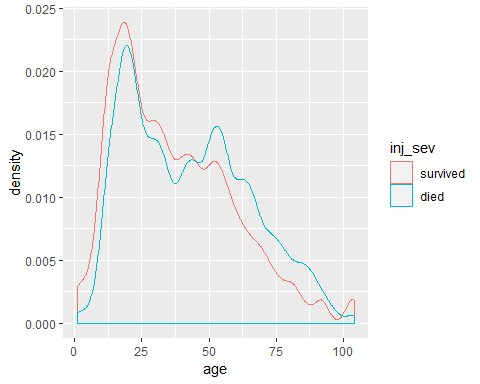
# Density of Injury Severity by Age split by Gender

ggplot(data = MedData) + geom\_density(mapping=aes(x = age, fill = inj\_sev)) + facet\_wrap(~inj\_sev)



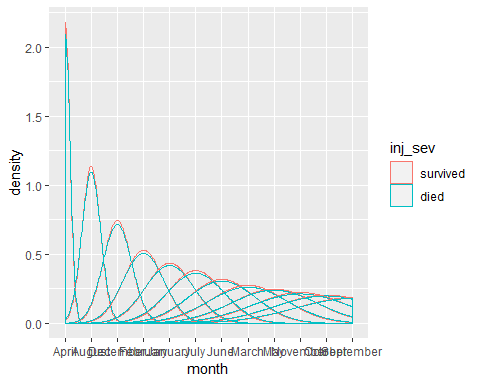
# Density of Injury Severity over Age

ggplot(df) + geom\_density(mapping = aes(x = age, color = inj\_sev))



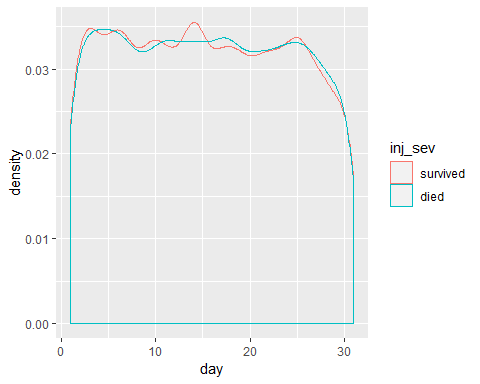
# Density of Injury Severity by Month

ggplot(df) + geom\_density(mapping = aes(x = month, color = inj\_sev))



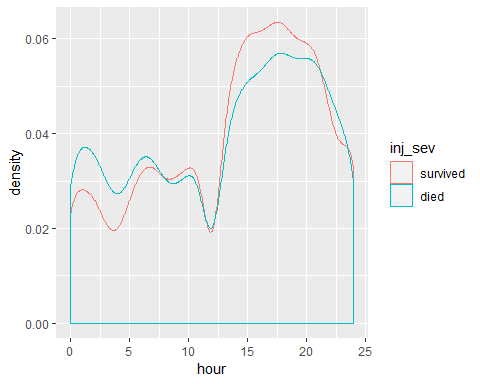
# Density of Injury Severity by Day of Month

ggplot(df) + geom\_density(mapping = aes(x = day, color = inj\_sev))



# Density of Injury Severity by Hour of Occurence

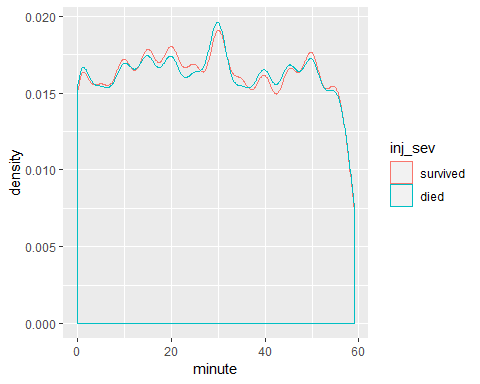
ggplot(df) + geom\_density(mapping = aes(x = hour, color = inj\_sev))



# Density of Injury Severity by Minute of Occurence

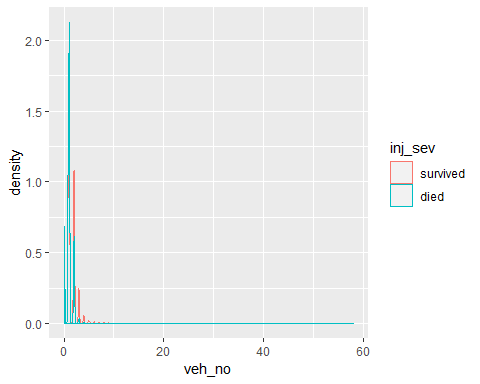
ggplot(df) + geom\_density(mapping = aes(x = minute, color = inj\_sev))

## Warning: Removed 7 rows containing non-finite values (stat\_density).



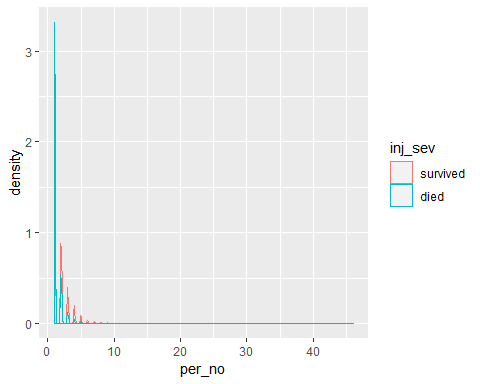
# Density of Injury Severity by Number of Vehicles Involved

ggplot(df) + geom\_density(mapping = aes(x = veh\_no, color = inj\_sev))



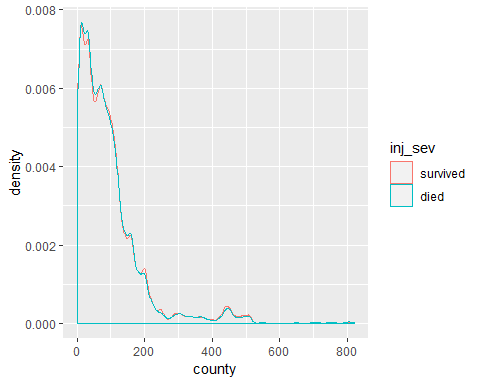
# Density of Injury Severity by Number of People Involved

ggplot(df) + geom\_density(mapping = aes(x = per\_no, color = inj\_sev))



# Density of Injury Severity by County Number

ggplot(df) + geom\_density(mapping = aes(x = county, color = inj\_sev))



### Predictive Models

## Decision Tree

dt1 <- train(inj\_sev ~.,data = MedData, method = "rpart",   
 trControl = trainControl(method ="cv", number = 3, verboseIter = TRUE))

## + Fold1: cp=0.0257   
## - Fold1: cp=0.0257   
## + Fold2: cp=0.0257   
## - Fold2: cp=0.0257   
## + Fold3: cp=0.0257   
## - Fold3: cp=0.0257   
## Aggregating results  
## Selecting tuning parameters  
## Fitting cp = 0.0257 on full training set

## Decision Tree Tuned

myGrid1 = expand.grid(.cp=0.02)  
  
dt2 <- train(inj\_sev ~.,data = MedData, method = "rpart",   
 trControl = trainControl(method ="cv", number = 3, verboseIter = TRUE),  
 tuneGrid = myGrid1)

## + Fold1: cp=0.02   
## - Fold1: cp=0.02   
## + Fold2: cp=0.02   
## - Fold2: cp=0.02   
## + Fold3: cp=0.02   
## - Fold3: cp=0.02   
## Aggregating results  
## Fitting final model on full training set

# Random Forest

myGrid2 = expand.grid(mtry = c(10,25), splitrule = c("gini"), min.node.size = c(1))  
  
rf1 <- train(inj\_sev ~.,data = MedData, method = "ranger", trControl = trainControl(method ="cv", number = 2, verboseIter = TRUE), tuneGrid = myGrid2)

## + Fold1: mtry=10, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 80%. Estimated remaining time: 7 seconds.  
## - Fold1: mtry=10, splitrule=gini, min.node.size=1   
## + Fold1: mtry=25, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 39%. Estimated remaining time: 48 seconds.  
## Growing trees.. Progress: 78%. Estimated remaining time: 17 seconds.  
## - Fold1: mtry=25, splitrule=gini, min.node.size=1   
## + Fold2: mtry=10, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 79%. Estimated remaining time: 8 seconds.  
## - Fold2: mtry=10, splitrule=gini, min.node.size=1   
## + Fold2: mtry=25, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 39%. Estimated remaining time: 49 seconds.  
## Growing trees.. Progress: 78%. Estimated remaining time: 17 seconds.  
## - Fold2: mtry=25, splitrule=gini, min.node.size=1   
## Aggregating results  
## Selecting tuning parameters  
## Fitting mtry = 25, splitrule = gini, min.node.size = 1 on full training set  
## Growing trees.. Progress: 16%. Estimated remaining time: 2 minutes, 40 seconds.  
## Growing trees.. Progress: 33%. Estimated remaining time: 2 minutes, 5 seconds.  
## Growing trees.. Progress: 50%. Estimated remaining time: 1 minute, 35 seconds.  
## Growing trees.. Progress: 66%. Estimated remaining time: 1 minute, 4 seconds.  
## Growing trees.. Progress: 83%. Estimated remaining time: 32 seconds.  
## Growing trees.. Progress: 99%. Estimated remaining time: 1 seconds.

# Random Forest Tuned

myGrid3 = expand.grid(mtry = c(30,50), splitrule = c("gini"), min.node.size = c(1))  
  
rf2 <- train(inj\_sev ~.,data = MedData, method = "ranger", trControl = trainControl(method ="cv", number = 2, verboseIter = TRUE), tuneGrid = myGrid3)

## + Fold1: mtry=30, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 33%. Estimated remaining time: 1 minute, 1 seconds.  
## Growing trees.. Progress: 67%. Estimated remaining time: 30 seconds.  
## - Fold1: mtry=30, splitrule=gini, min.node.size=1   
## + Fold1: mtry=50, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 22%. Estimated remaining time: 1 minute, 51 seconds.  
## Growing trees.. Progress: 44%. Estimated remaining time: 1 minute, 19 seconds.  
## Growing trees.. Progress: 66%. Estimated remaining time: 47 seconds.  
## Growing trees.. Progress: 88%. Estimated remaining time: 16 seconds.  
## - Fold1: mtry=50, splitrule=gini, min.node.size=1   
## + Fold2: mtry=30, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 34%. Estimated remaining time: 1 minute, 0 seconds.  
## Growing trees.. Progress: 68%. Estimated remaining time: 29 seconds.  
## - Fold2: mtry=30, splitrule=gini, min.node.size=1   
## + Fold2: mtry=50, splitrule=gini, min.node.size=1   
## Growing trees.. Progress: 22%. Estimated remaining time: 1 minute, 51 seconds.  
## Growing trees.. Progress: 44%. Estimated remaining time: 1 minute, 18 seconds.  
## Growing trees.. Progress: 67%. Estimated remaining time: 46 seconds.  
## Growing trees.. Progress: 89%. Estimated remaining time: 14 seconds.  
## - Fold2: mtry=50, splitrule=gini, min.node.size=1   
## Aggregating results  
## Selecting tuning parameters  
## Fitting mtry = 50, splitrule = gini, min.node.size = 1 on full training set  
## Growing trees.. Progress: 9%. Estimated remaining time: 5 minutes, 13 seconds.  
## Growing trees.. Progress: 19%. Estimated remaining time: 4 minutes, 35 seconds.  
## Growing trees.. Progress: 28%. Estimated remaining time: 4 minutes, 4 seconds.  
## Growing trees.. Progress: 38%. Estimated remaining time: 3 minutes, 30 seconds.  
## Growing trees.. Progress: 47%. Estimated remaining time: 2 minutes, 57 seconds.  
## Growing trees.. Progress: 57%. Estimated remaining time: 2 minutes, 25 seconds.  
## Growing trees.. Progress: 66%. Estimated remaining time: 1 minute, 53 seconds.  
## Growing trees.. Progress: 75%. Estimated remaining time: 1 minute, 22 seconds.  
## Growing trees.. Progress: 84%. Estimated remaining time: 53 seconds.  
## Growing trees.. Progress: 94%. Estimated remaining time: 21 seconds.

## Predictive Model Accuracy

max(dt1$results$Accuracy)

## [1] 0.7466759

max(dt2$results$Accuracy)

## [1] 0.7474607

max(rf1$results$Accuracy)

## [1] 0.7805741

max(rf2$results$Accuracy)

## [1] 0.7813731