**Project Code – R Vinay Gandra**

1. **RSI for Massachusetts data – mass.R on github repo**

#Loading Massachusetts fatal acccidents data for 2014-2015

data1 <- read.csv("capstone\_proj/FileName1.csv")

data2 <- read.csv("capstone\_proj/FileName2.csv")

#Cleaning data to remove unknown values

data <- rbind(data1,data2)

data <- data[!data$acchr == 99,]

#Grouping accidents by time groups

timegroup <- vector()

for(i in 1:nrow(data)){

if(data[i, 3] == 7 | data[i, 3] == 8 | data[i, 3] == 9 | data[i, 3] == 10 | data[i, 3] == 11){

timegroup[[i]] = 7

}

else if (data[i, 3] == 12 | data[i, 3] == 13 | data[i, 3] == 14){

timegroup[[i]] = 11

}

else if (data[i, 3] == 15 | data[i, 3] == 16){

timegroup[[i]] <- 14

}

else if (data[i, 3] == 17 | data[i, 3] == 18 | data[i, 3] == 19 | data[i, 3] == 20 | data[i, 3] == 21){

timegroup[[i]] <- 16

}

else if (data[i, 3] == 22 | data[i, 3] == 23 | data[i, 3] == 0){

timegroup[[i]] = 21

}

else {timegroup[[i]] = 1}

}

#Assigning each accidents its corresponding time group

data <- cbind(data, timegroup)

#Assigning some points to two arbitrary routes from data

r1 <- rbind(data[1:10,], data[18:19,], data[30:31,])

r2 <- rbind(data[1:3,], data[15:19,], data[21:23,])

r1 <- r1[c(4, 5, 8)]

r2 <- r2[c(4, 5, 8)]

#Finding RSI for each route during time group 16

RSI1\_16 <- nrow(r1[r1$timegroup == 16,])/nrow(r1)

RSI2\_16 <- nrow(r2[r2$timegroup == 16,])/nrow(r2)

RSI1\_16

RSI2\_16

1. **RSI for NYC data – logistic.R on github repo**

Logistic regression to identify significant factor levels within vehicle types and reason for crash.

#Loading crashes data for NYC 2014-2015

crashes <- read.csv("capstone\_proj/NYPD\_Motor\_Vehicle\_Collisions.csv")

#Identiying if crash involved has injuries/ fatalities or none.

for(i in 1:nrow(crashes)){

crashes[i,10] <- sum(crashes[i, 10:11])

}

#Grouping crashes with injuries/fatalities and crashes without injuries/ fatalities

injury\_killed <- vector()

for(i in 1:nrow(crashes)){

if(crashes[i,10] == 0) injury\_killed[[i]] = 0

else injury\_killed[[i]] = 1

}

rm(i)

accidents\_data <- cbind(crashes[, 1:10], injury\_killed, crashes[, c(14, 17)])

accidents\_data <- accidents\_data[!accidents\_data$VEHICLE.TYPE.CODE.1 == "",]

#Logitic regression on fatality/injury depending on vehicle type and reason for crash individually

vehicle\_log <- glm(accidents\_data$injury\_killed~accidents\_data$VEHICLE.TYPE.CODE.1, family = "binomial")

reason\_log <- glm(accidents\_data$injury\_killed~accidents\_data$CONTRIBUTING.FACTOR.VEHICLE.1, family = "binomial")

#Logistic regression with both independent variables together

logistic\_model <- glm(injury\_killed~CONTRIBUTING.FACTOR.VEHICLE.1+VEHICLE.TYPE.CODE.1, data = accidents\_data, family = "binomial")

1. **Coefficients for volume by time - volume.R on github repo**

Principal component analysis

#Loading volume data

volume <- read.csv("capstone\_proj/Volume.csv")

#Grouping hourly volume in timeperiods

tbins <- vector("list", 7)

for(i in 1:nrow(volume)){

tbins[[1]][i] <- sum(volume[i, 15:17])

tbins[[2]][i] <- sum(volume[i, 18:21])

tbins[[3]][i] <- sum(volume[i, 22:23])

tbins[[4]][i] <- sum(volume[i, 24:25])

tbins[[5]][i] <- sum(volume[i, 26:28])

tbins[[6]][i] <- sum(volume[i, 29:31], volume[i, 8])

tbins[[7]][i] <- sum(volume[i, 9:14])}rm(i)

tbins <- as.data.frame(tbins)

cnames <- c(7,10,14,16,18,21,1)

names(tbins) <- cnames

rm(cnames)

tbins <- cbind(volume[,c(3)], tbins)

colnames(tbins)[1] <- "STREETNAME"

#Extracting two routes

route\_a <- subset(tbins, STREETNAME == "7 AVENUE")

route\_b <- subset(tbins, STREETNAME == "BROADWAY")

#Normalizing data

for(i in 1:nrow(route\_a)){

for(j in 2:8){

route\_a[i, j] <- (route\_a[i, j] - min(route\_a[, j])) / (max(route\_a[,j]) - min(route\_a[,j]))

}}

for(i in 1:nrow(route\_b)){

for(j in 2:8){

route\_b[i, j] <- (route\_b[i, j] - min(route\_b[, j])) / (max(route\_b[,j]) - min(route\_b[,j]))

}}rm(i,j)

#Principal component analysis to get volume coeffiecient by timeperiod

pca\_a <- princomp(route\_a[, 2:8])

pca\_b <- princomp(route\_b[, 2:8])

coef\_rta <- abs(pca\_a$loadings[,1])

coef\_rtb <- abs(pca\_b$loadings[,1])

#Volume coefficients by time period

volCoef\_a <- as.data.frame(cbind(coef\_rta, names(coef\_rta)))

volCoef\_b <- as.data.frame(cbind(coef\_rtb, names(coef\_rtb)))

colnames(volCoef\_a) <- c("volCoef", "Hour")

colnames(volCoef\_b) <- c("volCoef", "Hour")

1. **Coefficients for crashes by type & RSI – crashes.R on github repo**

#Loading crashes data for NYC 2014-2015

crashes <- read.csv("capstone\_proj/NYPD\_Motor\_Vehicle\_Collisions.csv")

crashes <- crashes[, c(2,4:6, 8, 10, 11)]

#Identiying if crash involved has injuries/ fatalities or none.

#Also, extracating hour of crash

FATALITIES\_INJURIES = CRASH\_HOUR = vector()

for(i in 1:nrow(crashes)){

if(sum(crashes[i, 6:7]) == 0) {FATALITIES\_INJURIES[[i]] = 0}

else {FATALITIES\_INJURIES[[i]] = 1}

CRASH\_HOUR[[i]] <- format(as.POSIXct(crashes[i, 1], format = "%H:%M"), "%H")

}

data <- cbind(CRASH\_HOUR, crashes[,1:5], FATALITIES\_INJURIES)

colnames(data)[5] <- "STREETNAME"

#Grouping accidents by timeperiods

TIMEPERIOD <- vector()

for(i in 1:nrow(data)){

if(data[i,1] %in% c(07:09)){

TIMEPERIOD[[i]] <- 7}

else if(data[i,1] %in% c(10:13)){

TIMEPERIOD[[i]] <- 10}

else if(data[i,1] %in% c(14:15)){

TIMEPERIOD[[i]] <- 14}

else if(data[i,1] %in% c(16:17)){

TIMEPERIOD[[i]] <- 16}

else if(data[i,1] %in% c(18:20)){

TIMEPERIOD[[i]] <- 18}

else if(data[i,1] %in% c(21:24, 00)){

TIMEPERIOD[[i]] <- 21}

else {TIMEPERIOD[[i]] <- 1}}

data <- cbind(TIMEPERIOD, data[, 3:7])

#Extracting data for two routes

crashes\_a <- subset(data, STREETNAME == "7 AVENUE")

crashes\_b <- subset(data, STREETNAME == "BROADWAY")

#Group by to get counts of crash types by time period

library(plyr)

crashCount\_a <- ddply(crashes\_a, .(crashes\_a$TIMEPERIOD, crashes\_a$FATALITIES\_INJURIES), nrow)

crashCount\_b <- ddply(crashes\_b, .(crashes\_b$TIMEPERIOD, crashes\_b$FATALITIES\_INJURIES), nrow)

names(crashCount\_a) <- c("Hour","F\_I","Count")

names(crashCount\_b) <- c("Hour","F\_I","Count")

#Crash coefficient by crash type

crash\_coef <- vector()

for(i in 1:nrow(crashCount\_a)){

if(crashCount\_a[i,2] == 0)

crash\_coef[[i]] = crashCount\_a[i,3]\*0.20

else crash\_coef[[i]] = crashCount\_a[i,3]\*0.80}

crashCount\_a <- cbind(crashCount\_a, crash\_coef) rm(i)

crash\_coef <- vector()

for(i in 1:nrow(crashCount\_b)){

if(crashCount\_b[i,2] == 0)

crash\_coef[[i]] = crashCount\_b[i,3]\*0.20

else

crash\_coef[[i]] = crashCount\_b[i,3]\*0.80}

crashCount\_b <- cbind(crashCount\_b, crash\_coef) rm(i)

#Crash coefficient by time period

crashCoef\_a <- ddply(crashCount\_a,.(crashCount\_a$Hour), sum)

crashCoef\_b <- ddply(crashCount\_b,.(crashCount\_b$Hour), sum)

colnames(crashCoef\_a) <- c("Hour", "crashCoef")

colnames(crashCoef\_b) <- c("Hour", "crashCoef")

#Merging crash and volume data

data\_a <- merge(crashCoef\_a, volCoef\_a, by = "Hour")

data\_b <- merge(crashCoef\_b, volCoef\_a, by = "Hour")

#Initial individual time period rsi values

values\_a = values\_b = vector()

for(i in 1:nrow(data\_a)){

values\_a[[i]] <- (data\_a[i,2]\*as.numeric(data\_a[i,3]))/100}

for(i in 1:nrow(data\_b)){

values\_b[[i]] <- (data\_b[i,2]\*as.numeric(data\_b[i,3]))/100}

#RSI by route

rsi\_a <- sum(values\_a)/10

rsi\_b <- sum(values\_b)/10