

```

# EXPLORATORY ANALYSIS OF U.S. FATALITY ANALYSIS REPORTING SYSTEM DATA 2010-2014
# Jonathan Owen, April 18, 2016
# Data source http://www-fars.nhtsa.dot.gov/QueryTool/QuerySection/SelectYear.aspx
# Above URL is beginning of data selection for query
# FARS code lookup tables manually created from selection criteria
# GLC codes obtained from http://www.gsa.gov/portal/content/102761
#
# The following criteria were selected for the FARS queries for the years
# 2010-2014
# (Obs. = Sequential observation number within each year)
# statenum = integer representing GLC code for state
# casenum = sequential case number within each state and each year
# vnumber = sequential vehicle number within each case
# pnumber = person number within each vehicle
# atmcond & atmcond2 = integer code for atmospheric conditions
# city = integer representing GLC code for city
# county = integer representing GLC code for county
# acchr = integer representing accident hour, e.g. 0 for 0:00 to 0:59am
# accmon = integer representing accident month, e.g. 1 for January
# caseyear = integer for year of accident in YYYY format
# dayofweek = integer representing day of week, e.g. 1 for Sunday
# fhevent & mhevent = integer codes for first & most harmful events in accident sequence
# latitude & longitude = numeric for decimal form of location
# numfatal = integer number of fatalities in accident
# rfun = integer code for road function type, e.g. rural interstate
# injury = integer for severity of injury to person on a 0-8 scale
# vfatcount = integer number of fatalities in vehicle
# drfl = integer code for primary driver related factor
# crshavoid = integer code for maneuvers made by driver to avoid crash
# contribcirc = integer code for contributing circumstances
# crashtype = integer code for vehicle movement during crash
# criticalevent cat & criticalevent = integer codes for critical events prior to crash and their category
# devfunc = integer code for functional state of traffic control devices
# dridistract = integer code for causes of driver distraction
# driavoidmanvr = integer code for driver trying to avoid something as cause of crash
# drivisobs = integer code for visual obstruction for driver
# preimploc = integer code for pre-impact location
# preimpstab = integer code for pre-impact stability
# vehmanvr = integer code for vehicle movement prior to event
# roadalgn = integer code for road alignment, e.g. curve, left
# roadprof = integer code for roadway grade
# surfcond = integer code for road surface condition
# spdlim = integer of posted speed limit in miles per hour
# traflane = integer code for traffic lane arrangement
# contdev = integer code for types of traffic control device present
# trafflow = integer code for traffic description

library(dplyr)
library(ggplot2)

# READ GLC CODES
GSAloc <- read.table("GLCs_for_the_USA_and_DC.txt", sep = "\t", skip = 1, header = TRUE)
CEtxt <- read.table("Critical_Event_Codes.txt", sep = "\t")
RFTxt <- read.table("Road_Function_Codes.txt", sep = "\t")

# READ 2014 TABLE
ColClass <- rep("character", 42)
ColNames <- read.table("fars_nhtsa_2010.txt", sep = "\t", nrows = 1, colClasses = ColClass)
ColClass[c(6, 9:14, 19:21, 25:28, 32, 37, 40:41)] <- "numeric"
Fars5 <- read.table("fars_nhtsa_2014.txt", sep = "\t", skip = 2, colClasses = ColClass)

# FIX USE OF PERIOD IN 2014 TABLE
Fars5[grepl("^\\.$", Fars5[,7], perl = TRUE), 7] <- "99" #FARS code for Atmospheric Condition 2 UNKNOWN = 99
Fars5[grepl("^\\.$", Fars5[,8], perl = TRUE), 8] <- "9999" #GLC code for city UNKNOWN = 9999
Fars5[grepl("^\\.$", Fars5[,15], perl = TRUE), 15:16] <- "0.0"
Fars5[grepl("^\\.$", Fars5[,18], perl = TRUE), 18] <- "99" #FARS code for Roadway Function Class UNKNOWN = 99
Fars5[grepl("^\\.$", Fars5[,23], perl = TRUE), 23] <- "99" #FARS code for Attempted Avoidance Maneuver UNKNOWN = 99
Fars5[grepl("^\\.$", Fars5[,33], perl = TRUE), 33] <- "9" #FARS code for Pre-Impact Stability UNKNOWN = 9
Fars5[grepl("^\\.$", Fars5[,34], perl = TRUE), 34] <- "99" #FARS code for Pre-Event Movement UNKNOWN = 99
Fars5[grepl("^\\.$", Fars5[,35], perl = TRUE), 35] <- "9" #FARS code for Road Alignment UNKNOWN = 9
Fars5[grepl("^\\.$", Fars5[,36], perl = TRUE), 36] <- "9" #FARS code for Roadway Grade UNKNOWN = 9
Fars5[grepl("^\\.$", Fars5[,38], perl = TRUE), 38] <- "9" #FARS code for Surface Type UNKNOWN = 9
Fars5[grepl("^\\.$", Fars5[,39], perl = TRUE), 39] <- "99" #FARS code for Speed Limit UNKNOWN = 99
Fars5[grepl("^\\.$", Fars5[,42], perl = TRUE), 42] <- "9" #FARS code for Trafficway Description UNKNOWN = 9
for (i in c(7:8, 15:18, 23, 33:36, 38:39, 42)) {Fars5[,i] <- as.numeric(Fars5[,i])}

# READ TABLES FOR OTHER YEARS, COMBINE, CLEAN UP
ColClass[c(7:8, 15:18, 23, 33:36, 38:39, 42)] <- "numeric"
Fars1 <- read.table("fars_nhtsa_2010.txt", sep = "\t", skip = 2, colClasses = ColClass)
Fars2 <- read.table("fars_nhtsa_2011.txt", sep = "\t", skip = 2, colClasses = ColClass)
Fars3 <- read.table("fars_nhtsa_2012.txt", sep = "\t", skip = 2, colClasses = ColClass)
Fars4 <- read.table("fars_nhtsa_2013.txt", sep = "\t", skip = 2, colClasses = ColClass)
Fars <- rbind(Fars1, Fars2, Fars3, Fars4, Fars5)
Fars <- Fars[,1:42]

```

```

names(Fars) <- ColNames[1:42]
rm(Fars1); rm(Fars2); rm(Fars3); rm(Fars4); rm(Fars5)
Fars[grepl("^\$.", Fars[,22], perl = TRUE), 22] <- "99" #FARS code for Driver Related Factor 1 UNKNOWN = 99
Fars[grepl("^\$.", Fars[,29], perl = TRUE), 29] <- "99" #FARS code for Driver Distracted By UNKNOWN = 99
Fars[,22] <- as.numeric(Fars[,22])

# CREATE UNIQUE OBSERVATION AND CASE ID
Fars$"Obs." <- paste0(Fars$caseyear, "-", Fars$"Obs.")
Fars$casenum <- paste0(Fars$caseyear, "-", Fars$statenum, "-", Fars$casenum)
names(Fars)[1] <- "obsid"

# CONVERT TO FACTORS
# NOTE: prior conversion to numeric was used to check for valid coding
for (i in c(2, 6:14, 18:20, 22:42)) {Fars[,i] <- as.factor(Fars[,i])}

#
tCriticalEvents <- Fars %>%
  group_by(criticalevent, casenum) %>%
  summarise(CaseFatal = sum(numfatal)/n()) %>%
  arrange(desc(CaseFatal))

tCriticalEvents2 <- tCriticalEvents %>%
  group_by(criticalevent) %>%
  summarise(EventFatal = sum(CaseFatal)) %>%
  arrange(criticalevent)

tCriticalEvents2$CriticalEvent <- CEtxt[tCriticalEvents2$criticalevent,2]

tCriticalEvents <- arrange(tCriticalEvents2, desc(EventFatal))

pEvent <- ggplot(data = subset(tCriticalEvents, EventFatal > 10000),
  aes(x = CriticalEvent, y = EventFatal, label = CriticalEvent)) +
  theme_bw(base_size = 10)
pEvent <- pEvent + geom_point() + geom_text(vjust = -0.5, size = 3) +
  theme(axis.ticks.x = element_blank(), axis.text.x = element_blank()) +
  ggtitle("U.S. Vehicle Accident Fatalities by Critical Events Prior to Crash, 2010-2014") +
  labs(y = "Total Fatalities 2010-2014")
pdf("critical_event_plot.pdf", width = 11, height = 8)
pEvent
dev.off()

#
tRoadFn <- Fars %>%
  group_by(rfun, spdlim, casenum) %>%
  summarise(CaseFatal = sum(numfatal)/n()) %>%
  arrange(desc(CaseFatal))

tRoadFn2 <- tRoadFn %>%
  group_by(rfun, spdlim) %>%
  summarise(RoadFatal = sum(CaseFatal)) %>%
  arrange(rfun)

tRoadFn2$RoadFunction <- Rftxt[tRoadFn2$rfun,2]

SpeedRange <- c("0-25", "30-35", "40-45", "50-55", "60-65", "70+")
lSpdlim <- levels(Fars$spdlim)
lSpdRng <- c(rep(1, 6), rep(2, 2), rep(4, 2), rep(5, 2), rep(6, 5))
tSpeeds <- data.frame(cbind(lSpdlim, lSpdRng))
tRoadFn2$SpeedLimit <- tSpeeds[tRoadFn2$spdlim,2]
tRoadFn2$SpeedLimit <- SpeedRange[tRoadFn2$SpeedLimit]
tRoadFn <- arrange(tRoadFn2, desc(RoadFatal))
tRoadFn3 <- tRoadFn %>%
  filter(spdlim != "98" & spdlim != "99" ) %>%
  group_by(rfun, RoadFunction, SpeedLimit) %>%
  summarise(RoadFatal = sum(RoadFatal))
tRoadFn3$SpeedLimit <- as.factor(tRoadFn3$SpeedLimit)
tRoadFn3 <- arrange(tRoadFn3, desc(RoadFatal))
subRoadFn <- tRoadFn3[tRoadFn3$rfun %in% as.character(c(1:5)), ]

pRoadFn <- ggplot(data = subRoadFn, aes(x = SpeedLimit, y = RoadFatal)) + theme_bw(base_size = 10)
pRoadFn <- pRoadFn + geom_point(aes(color = RoadFunction), alpha = 0.5, size = 3) +
  facet_grid(. ~ RoadFunction)
pRoadFn <- pRoadFn + theme(legend.position = "none") +
  ggtitle("U.S. Vehicle Accident Fatalities by Speed Limit and Road Function, 2010-2014") +
  labs(x = "Posted Speed Limit, mph", y = "Total Fatalities 2010-2014")
pdf("road_function_plot.pdf", width = 11, height = 8)
pRoadFn
dev.off()

```