EDA – Vehicle Vulnerability

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
opts_chunk$set(comment = NA)
```

Analyzing patterns related to vehicle attributes such as type, make, model, body type, etc. combined with type of damage, rollover, fire/explosion, and frequency of incidents can reveal vehicle vulnerabilities. Employ an exploratory approach similar to what is discussed in the previous question to hypothesize and validate vehicle vulnerabilities.

```
library(MASS)
library(plyr)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:plyr':

    arrange, count, desc, failwith, id, mutate, rename, summarise,
    summarize

The following object is masked from 'package:MASS':

    select

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(caret)
```

```
Loading required package: lattice
```

```
setwd("/Users/jackfrancis/Dropbox/Heuristics_Optimization/IISE_Data_Analytics/Jack_Code")
```

Initial thoughts:

- Jeep/ SUV are probably more likely to rollover compared to other vehicle types, this may be an interesting relationship to investigate
- Frequency of incidents should be interesting, but care will need to be taken that the general observations are not correlated with demand
- Fire/explosion seem exceedingly rare, so investigating the types of cars that fall in this category would be interesting. It would also be interesting to see if there is a common sequence of events that leads to fires, or if the car itself is the important covariate.

- An interesting point to consider is that this data is survivorship biased heavily. Finding the safest car based on the data is not the same as finding the safest car overall, because the safest car may have never been in an accident and thus not in this dataset.
- Are certain types/ make/model more likely to be in speeding related accidents?
- Get Reliability Data for each make / model
- Get Safety Data for various body types

Code Information:

Table/Column	CSV Name	FARS Data Dictionary Location	Other
Damaged Areas	DAMAGE	371	NA
Vehicle/MAKE	VEHICLE	189	NA
Vehicle/MODEL	VEHICLE	192	NA
Vehicle/BODY_TYP	VEHICLE	298	NA
Vehicle/MOD_YEAR	VEHICLE	316	4 digit model year
Vehicle/ROLLOVER	VEHICLE	367	Rollover Type (0: None, 1:Tripped, 2:Untripped, 9:Unknown)
Vehicle/ROLINLOC	VEHICLE	369	Rollover Location
Vehicle/IMPACT1	VEHICLE	371	Initial Impact
Vehicle/DEFORMED	VEHICLE	380	Extent of Damage
Vehicle/FIRE_EXP	VEHICLE	420	Fire/Explosion Occurence (0: No, 1: Yes)
Vehicle/SPEEDREL	VEHICLE	464	Speeding Related

ROLINLOC Codes	Attributes
0	None
1	Roadway
2	Shoulder
3	Median/Separator
4	In Gore
5	On Roadside
6	Outside of Trafficway
7	In Parking Lane/Zone
9	Unknown

IMPACT 1 Codes	Attributes
00	Non-Collision
01-12	Clock Points
13	Top
14	Undercarriage
61	Left
62	Left-Front Side
63	Left-Back Side
81	Right
82	Right-Front Side
83	Right-Back Side
18	Cargo/Vehicle Parts Set-In-Motion
19	Other Objects Set-In-Motion

IMPACT 1 Codes	Attributes
20	Object Set in Motion, Unknown if Cargo/Vehicle Parts or Other
98	Not Reported
99	Reported as Unknown

MDAREAS Codes	Attributes
01-12	Clock Values
13	Top
14	Undercarriage
15	No Damage
99	Damage Areas Unknown

DEFORMED Codes	Attributes
0	None
2	Minor Damage
4	Functional Damage
6	Disabling Damage
8	Not Reported
9	Unknown

SPEEDREL Codes	Attributes
0	No
2	Yes, Racing
3	Yes, Exceeded Speed Limit
4	Yes, Too Fast for Conditions
5	Yes, Specifics Unknown
9	Unknown

```
vehicle_df = read.csv("../FARS_Data/FARS2018NationalCSV/VEHICLE.csv")
damages_df = read.csv("../FARS_Data/FARS2018NationalCSV/DAMAGE.csv")
accident_df = read.csv("../FARS_Data/FARS2018NationalCSV/ACCIDENT.csv")
```

The first item to investigate is if the type of car is an important factor in crashes/fatalities. Let's first get all of the types of cars in the dataset.

```
vehicle_body_types = vehicle_df$BODY_TYP
print(sort(unique(vehicle_body_types)))
```

```
 [1]  1  2  3  4  5  6  8  9 10 11 12 13 14 15 16 17 19 20 21 22 28 29 34 39 40
[26] 45 48 49 50 51 52 55 58 59 60 61 62 63 64 65 66 67 71 72 73 78 79 80 81 82
[51] 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
```

Lots of vehicle types. These are grouped into 9 categories (plus an other category) in the Data Dictionary. I am grouping them based on these predefined groupings.

Category	Group
1	Automobile

Category	Group
2	Automobile Derivative
3	Utility Vehicle
4	Van
5	Light Truck
6	Bus
7	Heavy Truck
8	Motor Home
9	Motorcycle/Moped
10	Other

```
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 17),
                                to = rep(1, 10))
```

```
The following `from` values were not present in `x`: 7
```

```
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(10, 11, 12, 13),
                                to = rep(2, 4))
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(14, 15, 16, 19),
                                to = rep(3, 4))
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(20, 21, 22, 28, 29),
                                to = rep(4, 5))
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(33, 34, 39, 40, 41, 45, 48, 49),
                                to = rep(5, 8))
```

```
The following `from` values were not present in `x`: 33, 41
```

```
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(50, 51, 52, 55, 58, 59),
                                to = rep(6, 6))
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(60, 61, 62, 63, 64, 66, 67, 71, 72, 78, 79),
                                to = rep(7, 11))
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(42, 65, 73),
                                to = rep(8, 3))
```

```
The following `from` values were not present in `x`: 42
```

```
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90),
                                to = rep(9, 11))
vehicle_body_types <- mapvalues(vehicle_body_types,
                                from = c(91, 92, 93, 94, 95, 96, 97, 98, 99),
                                to = rep(10, 9))
```

Next, lets see how the various groups compare on number of incidents, extent of damage, rollover percentage, fire/explosion percentage, and speed related percentage. Before this, some data preparation is needed.

```r
# Extent of Damage, assuming that not reported and unknown
# are same category to be removed
damage_extent <- vehicle_df$DEFORMED
damage_extent <- mapvalues(damage_extent, from = c(0, 2, 4, 6,
    8, 9), to = c(1, 2, 3, 4, 5, 5))

# Number of Missing/Not Reported Values (4455)
length(damage_extent[damage_extent == 5])
```

```
[1] 4455
```

```r
# Rollover Percentage
rollover <- vehicle_df$ROLLOVER
rollover <- mapvalues(rollover, from = c(0, 1, 2, 9), to = c(0,
    1, 1, 2))

# Some missing values can be collected by examining the
# ROLINLOC column. If this value is not 0 or 9, then the
# vehicle did rollover. (430 -> 29)
for (i in 1:length(rollover)) {
    if (rollover[i] == 2) {
        rollover[i] = ifelse(vehicle_df$ROLINLOC[i] != 9 & vehicle_df$ROLINLOC[i] !=
            0, 1, 2)
    }
}

# Number of Missing Values (29)
length(rollover[rollover == 2])
```

```
[1] 29
```

```r
# Fire/Explosion
fire_exp <- vehicle_df$FIRE_EXP

# Speed Related Incidents
speed_rel <- vehicle_df$SPEEDREL
speed_rel <- mapvalues(speed_rel, from = c(0, 2, 3, 4, 5, 8,
    9), to = c(0, 1, 1, 1, 1, 2, 2))

# Some missing values can be collected by comparing the
# travel speed to the speed limit (2077 -> 1596)
for (i in 1:length(speed_rel)) {
    if (speed_rel[i] == 2) {
        if (vehicle_df$TRAV_SP[i] < 151) {
            speed_rel[i] = ifelse((vehicle_df$TRAV_SP[i] - vehicle_df$VSPD_LIM[i]) >
                0, 1, 0)
        } else {
            speed_rel[i] == 2
        }
    }
}

# Number of Missing Values (1596)
length(speed_rel[speed_rel == 2])
```

```
[1] 1596
```

Let's get the number of occurences for each vehicle type and the percentage of rollover, fire/exp, speed rel, and the 4 types of damage

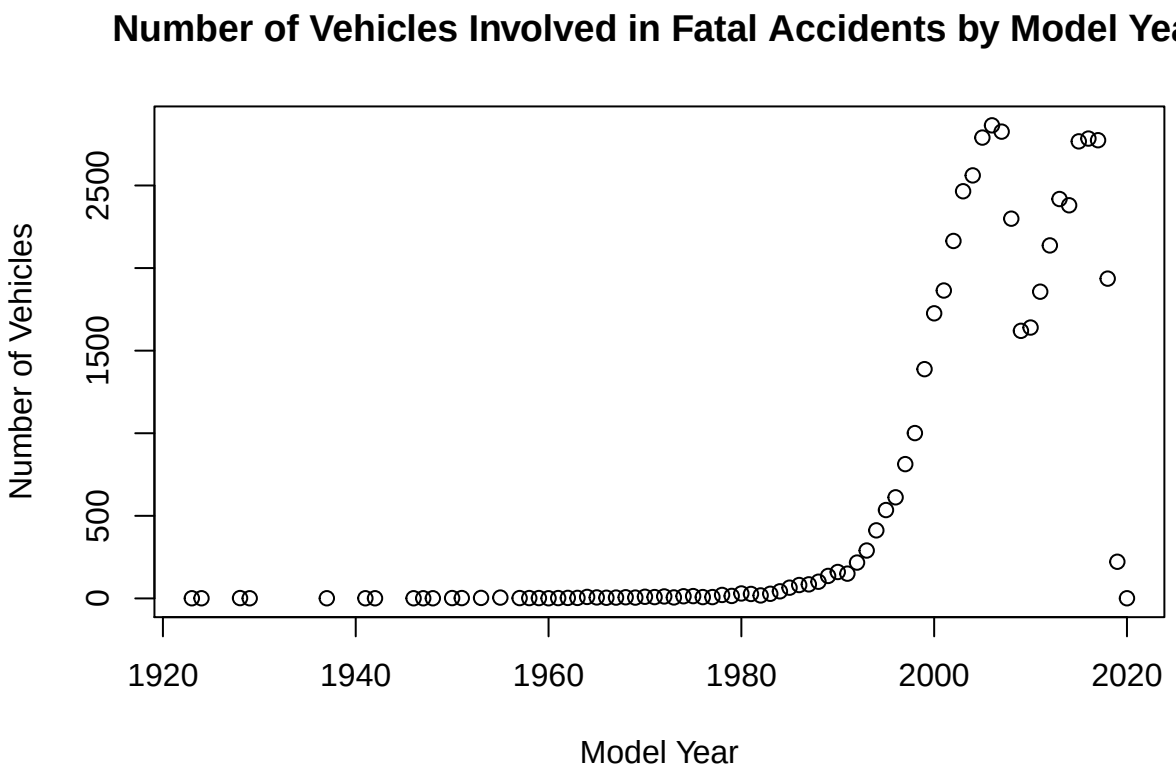
```
body_type_vulnerability <- data.frame(matrix(ncol = 9, nrow = 10))
colnames(body_type_vulnerability) <- c("Group", "Accidents", "Rollover Percentage",
                                       "Fire/Exp Percentage", "Speed Rel Percentage",
                                       "No Damage Percentage", "Minor Damage Percentage",
                                       "Functional Damage Percentage",
                                       "Disabling Damage Percentage")
vehicle_combined = data.frame(vehicle_body_types, rollover, fire_exp, speed_rel, damage_extent)
for (i in 1:10) {
    vehicle_rollover = vehicle_combined[vehicle_combined$vehicle_body_types == i &
                                            vehicle_combined$rollover != 2, ]
    vehicle_fire_exp = vehicle_combined[vehicle_combined$vehicle_body_types == i,]
    vehicle_speedrel = vehicle_combined[vehicle_combined$vehicle_body_types == i &
                                            vehicle_combined$speed_rel != 2, ]
    vehicle_damage = vehicle_combined[vehicle_combined$vehicle_body_types == i &
                                        vehicle_combined$damage_extent != 5,]
    body_type_vulnerability[i, 1] = i
    body_type_vulnerability[i, 2] = length(vehicle_body_types[vehicle_body_types == i])
    body_type_vulnerability[i, 3] = round(sum(vehicle_rollover$rollover)
                                            /nrow(vehicle_rollover) * 100, digits = 3)
    body_type_vulnerability[i, 4] = round(sum(vehicle_fire_exp$fire_exp)
                                            /nrow(vehicle_fire_exp) * 100, digits = 3)
    body_type_vulnerability[i, 5] = round(sum(vehicle_speedrel$speed_rel)
                                            /nrow(vehicle_speedrel) * 100, digits = 3)
    body_type_vulnerability[i, 6] = round(sum(vehicle_damage$damage_extent == 1)
                                            /nrow(vehicle_damage) * 100, digits = 3)
    body_type_vulnerability[i, 7] = round(sum(vehicle_damage$damage_extent == 2)
                                            /nrow(vehicle_damage) * 100, digits = 3)
    body_type_vulnerability[i, 8] = round(sum(vehicle_damage$damage_extent == 3)
                                            /nrow(vehicle_damage) * 100, digits = 3)
    body_type_vulnerability[i, 9] = round(sum(vehicle_damage$damage_extent == 4)
                                            /nrow(vehicle_damage) * 100, digits = 3)
}
```

Can turn the vulnerability calculation into a function for body_type, make, model etc.

```
get_vulnerability <- function(x) {
    vulnerability <- data.frame(matrix(ncol = 9, nrow = length(sort(unique(x)))))
    colnames(vulnerability) <- c("Group", "Accidents", "Rollover Percentage",
                                 "Fire/Exp Percentage", "Speed Rel Percentage",
                                 "No Damage Percentage", "Minor Damage Percentage",
                                 "Functional Damage Percentage",
                                 "Disabling Damage Percentage")
    vehicle_combined = data.frame(x, rollover, fire_exp, speed_rel, damage_extent)
    # Not all categories before preprocessing start from 1
    counter = 1
    for (i in sort(unique(x))) {
        vehicle_rollover = vehicle_combined[vehicle_combined$x == i &
                                                vehicle_combined$rollover != 2, ]
        vehicle_fire_exp = vehicle_combined[vehicle_combined$x == i, ]
        vehicle_speedrel = vehicle_combined[vehicle_combined$x == i &
                                                vehicle_combined$speed_rel != 2, ]
```

```
        vehicle_damage = vehicle_combined[vehicle_combined$x == i &
                                            vehicle_combined$damage_extent != 5, ]
        vulnerability[counter, 1] = i
        vulnerability[counter, 2] = length(x[x == i])
        vulnerability[counter, 3] = round(sum(vehicle_rollover$rollover)
                                          /nrow(vehicle_rollover) * 100, digits = 3)
        vulnerability[counter, 4] = round(sum(vehicle_fire_exp$fire_exp)
                                          /nrow(vehicle_fire_exp) * 100, digits = 3)
        vulnerability[counter, 5] = round(sum(vehicle_speedrel$speed_rel)
                                          /nrow(vehicle_speedrel) * 100, digits = 3)
        vulnerability[counter, 6] = round(sum(vehicle_damage$damage_extent == 1)
                                          /nrow(vehicle_damage) * 100, digits = 3)
        vulnerability[counter, 7] = round(sum(vehicle_damage$damage_extent == 2)
                                          /nrow(vehicle_damage) * 100, digits = 3)
        vulnerability[counter, 8] = round(sum(vehicle_damage$damage_extent == 3)
                                          /nrow(vehicle_damage) * 100, digits = 3)
        vulnerability[counter, 9] = round(sum(vehicle_damage$damage_extent == 4)
                                          /nrow(vehicle_damage) * 100, digits = 3)
        counter = counter + 1
    }
  return(vulnerability)
}
```

```
# Body Type
body_type_vulnerability = get_vulnerability(vehicle_body_types)

# Make There are a lot of makes, so need a better way to
# preprocess this data.
make_vulnerability = get_vulnerability(vehicle_df$MAKE)

# Model Sorting by model leads to the same observation as
# body type.
model_vulnerability = get_vulnerability(vehicle_df$MODEL)

# Model Year
model_yr_vulnerability = get_vulnerability(vehicle_df$MOD_YEAR)
plot(model_yr_vulnerability[1:78, 1], model_yr_vulnerability[1:78,
    2], main = "Number of Vehicles Involved in Fatal Accidents by Model Year",
    xlab = "Model Year", ylab = "Number of Vehicles")
```

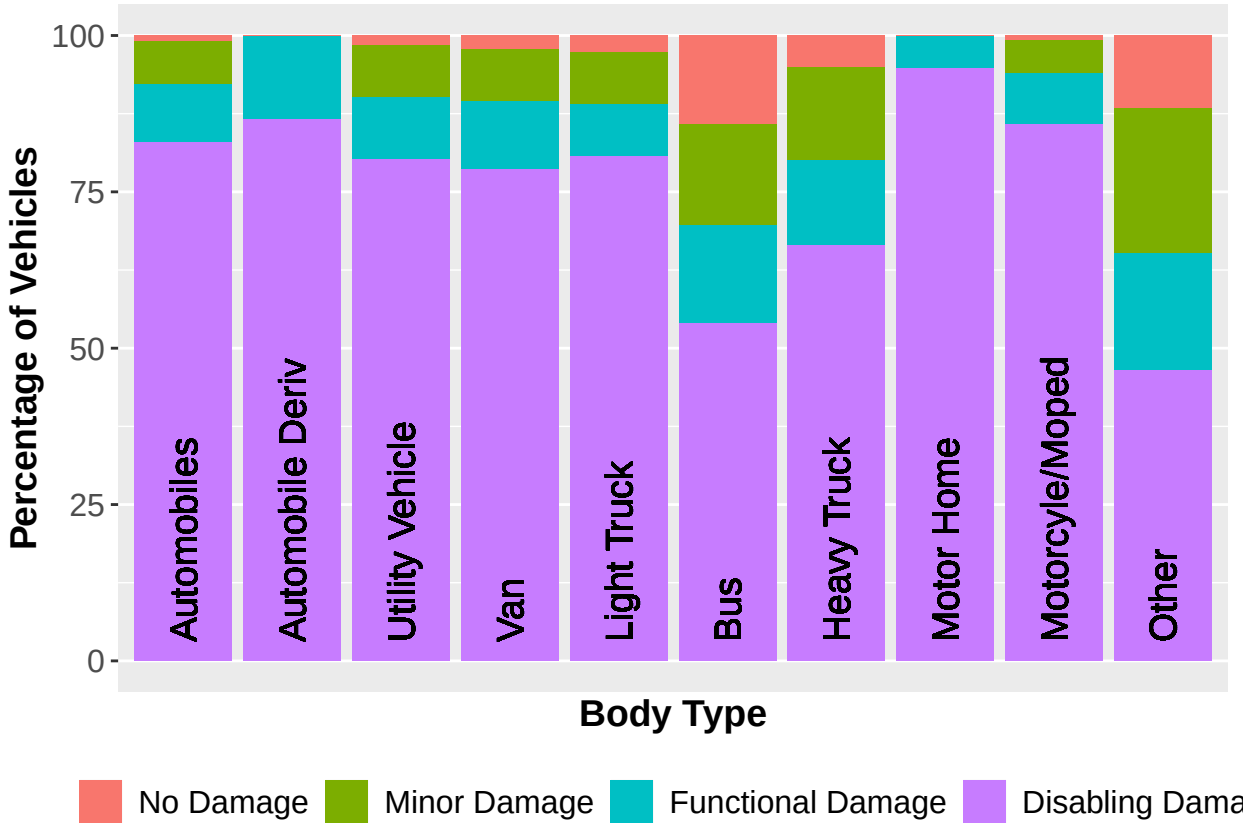


Interestingly, there is a sharp downturn in the number of vehicles in the mid 2000s.

This is due to the economic recession. Additionally, there were multiple vehicles from the 1920's involed in fatal accidents.

```
# Now turn the above into a table and then a stacked barplot
summary_info_body_type = c(0, 0, 0)
named_columns = c("Automobile", "Automobile Derivative", "Utility Vehicle",
    "Van", "Light Truck", "Bus", "Heavy Truck", "Motor Home",
    "Motorcycle/Moped", "Other")
named_rows = c(0, 0, 0, 0, 0, "No Damage", "Minor Damage", "Functional Damage",
    "Disabling Damage")

for (i in 1:nrow(body_type_vulnerability)) {
    for (j in 6:ncol(body_type_vulnerability)) {
        summary_info_body_type = rbind(summary_info_body_type,
            c(named_columns[i], named_rows[j], body_type_vulnerability[i,
                j]))
    }
}

summary_info_body_type = data.frame(summary_info_body_type)
summary_info_body_type$X3 = as.numeric(as.matrix(summary_info_body_type$X3))
summary_info_body_type = summary_info_body_type[-1, ]
row.names(summary_info_body_type) <- 1:nrow(summary_info_body_type)

summary_plot = ggplot(summary_info_body_type, aes(fill = factor(X2,
    levels = c("No Damage", "Minor Damage", "Functional Damage",
        "Disabling Damage")), y = as.numeric(X3), x = factor(X1,
    levels = c("Automobile", "Automobile Derivative", "Utility Vehicle",
        "Van", "Light Truck", "Bus", "Heavy Truck", "Motor Home",
        "Motorcycle/Moped", "Other")))) + geom_bar(position = "stack",
    stat = "identity") + xlab("Body Type") + # names(c('Sunday', 'Monday', 'Tuesday', 'Wed.', 'Thursday
# 'Friday', 'Saturday')) +
ylab("Percentage of Vehicles") + # ggtitle('Percentage of Accidents Where Each Cause was
# Related') +
scale_x_discrete(breaks = 1:10, labels = c("Automobile", "Automobile Derivative",
    "Utility Vehicle", "Van", "Light Truck", "Bus", "Heavy Truck",
    "Motor Home", "Motorcycle/Moped", "Other")) + theme(legend.position = "bottom",
    legend.direction = "horizontal", legend.title = element_blank(),
    plot.title = element_text(size = 16, hjust = 0.5), axis.text = element_text(size = 12),
    axis.title = element_text(size = 14, face = "bold"), legend.text = element_text(size = 12),
    axis.text.x = element_text(angle = 45, hjust = 1)) + geom_text(x = 1,
    y = 3, label = "Automobiles", angle = 90, hjust = 0, size = 5) +
    geom_text(x = 2, y = 3, label = "Automobile Deriv", angle = 90,
        hjust = 0, size = 5) + geom_text(x = 3, y = 3, label = "Utility Vehicle",
    angle = 90, hjust = 0, size = 5) + geom_text(x = 4, y = 3,
    label = "Van", angle = 90, hjust = 0, size = 5) + geom_text(x = 5,
    y = 3, label = "Light Truck", angle = 90, hjust = 0, size = 5) +
    geom_text(x = 6, y = 3, label = "Bus", angle = 90, hjust = 0,
        size = 5) + geom_text(x = 7, y = 3, label = "Heavy Truck",
    angle = 90, hjust = 0, size = 5) + geom_text(x = 8, y = 3,
    label = "Motor Home", angle = 90, hjust = 0, size = 5) +
    geom_text(x = 9, y = 3, label = "Motorcyle/Moped", angle = 90,
        hjust = 0, size = 5) + geom_text(x = 10, y = 3, label = "Other",
    angle = 90, hjust = 0, size = 5)
```

```
summary_plot
```

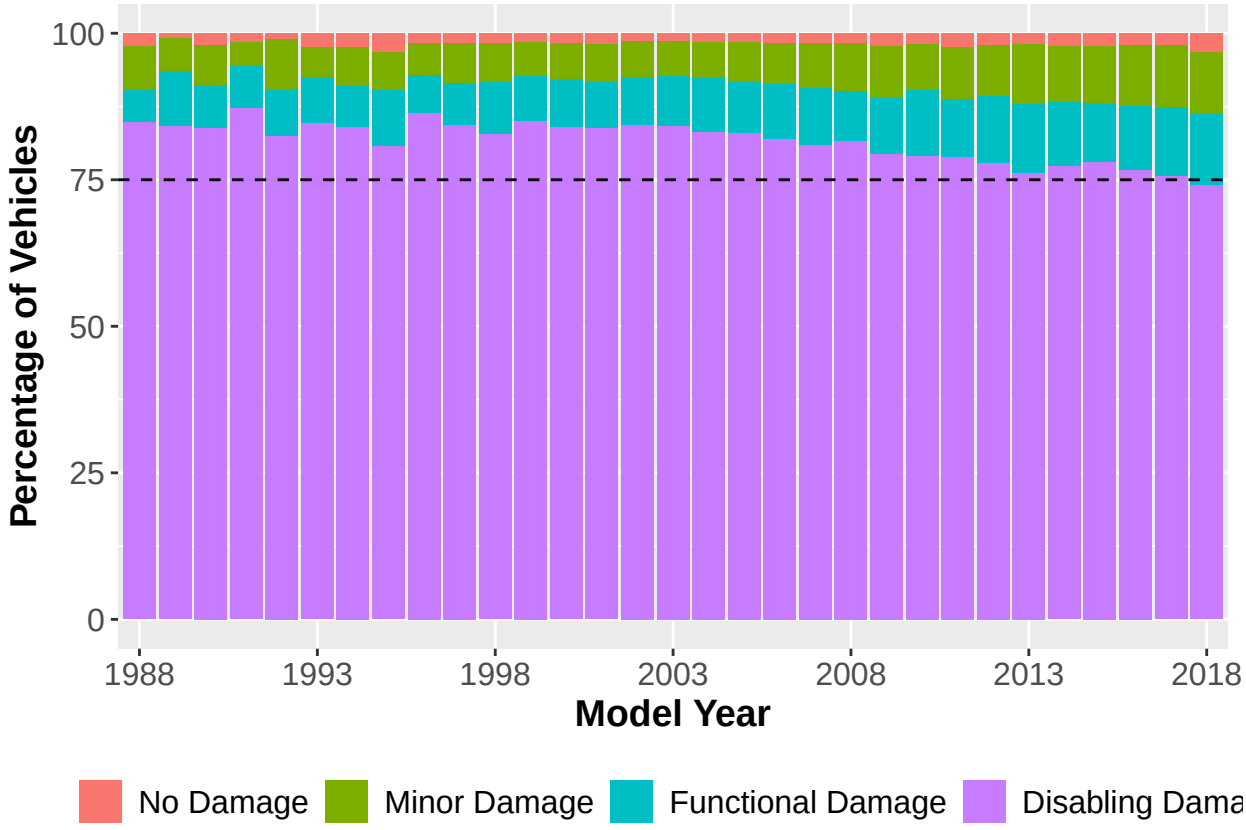



```
png("Q5_Body_Type_Damage_Summary.png")
summary_plot
dev.off()
```

```
pdf
  2
```

```
# Now turn the above into a table and then a stacked barplot
model_yr_summary = model_yr_vulnerability[model_yr_vulnerability$Group >
    1987, ]
model_yr_summary = model_yr_summary[model_yr_summary$Group <
    2019, ]

summary_info_model_year = c(0, 0, 0)
named_columns = c(seq(1988, 2018, 1))
named_rows = c(0, 0, 0, 0, 0, "No Damage", "Minor Damage", "Functional Damage",
    "Disabling Damage")

for (i in 1:nrow(model_yr_summary)) {
    for (j in 6:ncol(model_yr_summary)) {
        summary_info_model_year = rbind(summary_info_model_year,
            c(named_columns[i], named_rows[j], model_yr_summary[i,
                j]))
    }
}
```

```r
summary_info_model_year = data.frame(summary_info_model_year)
summary_info_model_year$X3 = as.numeric(as.matrix(summary_info_model_year$X3))
summary_info_model_year = summary_info_model_year[-1, ]
row.names(summary_info_model_year) <- 1:nrow(summary_info_model_year)

every_nth = function(n) {
    return(function(x) {
        x[c(TRUE, rep(FALSE, n - 1))]
    })
}

summary_plot = ggplot(summary_info_model_year, aes(fill = factor(X2,
    levels = c("No Damage", "Minor Damage", "Functional Damage",
        "Disabling Damage")), y = as.numeric(X3), x = factor(X1,
    levels = c(seq(1988, 2018, 1))))) + geom_bar(position = "stack",
    stat = "identity") + xlab("Model Year") + ylab("Percentage of Vehicles") +
    geom_hline(yintercept = 75, linetype = "dashed") + scale_x_discrete(breaks = every_nth(n = 5)) +
    theme(legend.position = "bottom", legend.direction = "horizontal",
        legend.title = element_blank(), plot.title = element_text(size = 16,
            hjust = 0.5), axis.text = element_text(size = 12),
        axis.title = element_text(size = 14, face = "bold"),
        legend.text = element_text(size = 12), axis.text.x = element_text(angle = 0,
            hjust = 0.5))

summary_plot
```



```r
png("Q5_Model_Year_Damage_Summary.png")
summary_plot
dev.off()
```

```
pdf
  2
```