Project - NYS-Motor-Vehicle-Crashes-And-Insurance-Reduction

March 3, 2019

```
library(ggplot2)
library(ggrepel)
library(reshape2)
library(stringr)
library(scales)
library(plyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:plyr':
##
##     here
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(readxl)
library(zipcode)
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(stringi)
library(proto)
library(gsubfn)
library(RSQLite)
library(sqldf)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------
- tidyverse 1.2.1 --
```

```
## v tibble  2.0.1       v purrr   0.3.0
## v tidyr   0.8.2       v dplyr   0.8.0.1
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ---------------------------------------------------------------------------- tidy
verse_conflicts() --
## x dplyr::arrange()         masks plyr::arrange()
## x lubridate::as.difftime() masks base::as.difftime()
## x readr::col_factor()      masks scales::col_factor()
## x purrr::compact()         masks plyr::compact()
## x dplyr::count()           masks plyr::count()
## x lubridate::date()        masks base::date()
## x purrr::discard()         masks scales::discard()
## x dplyr::failwith()        masks plyr::failwith()
## x dplyr::filter()          masks stats::filter()
## x lubridate::here()        masks plyr::here()
## x dplyr::id()              masks plyr::id()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x dplyr::mutate()          masks plyr::mutate()
## x dplyr::rename()          masks plyr::rename()
## x lubridate::setdiff()     masks base::setdiff()
## x dplyr::summarise()       masks plyr::summarise()
## x dplyr::summarize()       masks plyr::summarize()
## x lubridate::union()       masks base::union()
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

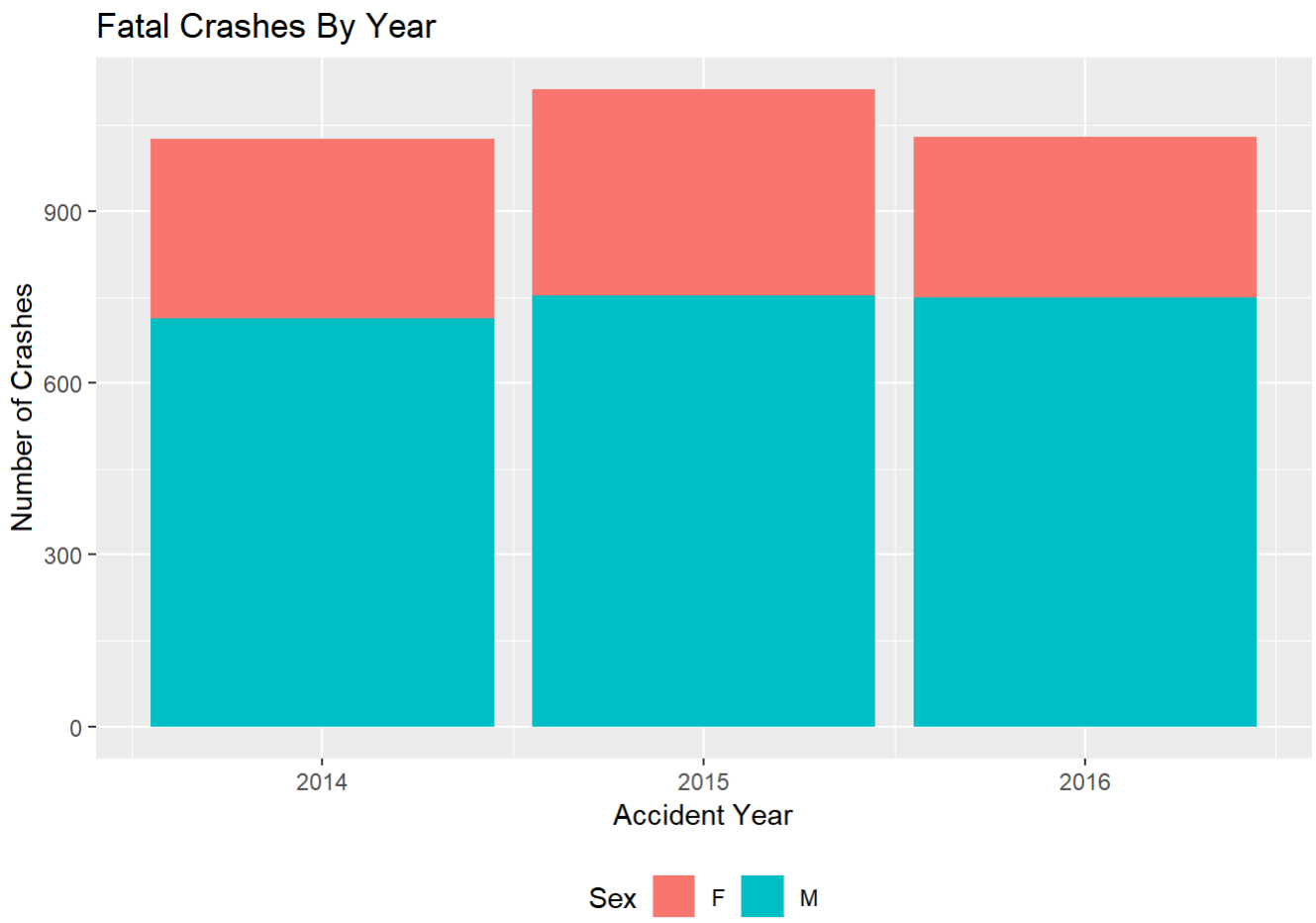
```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

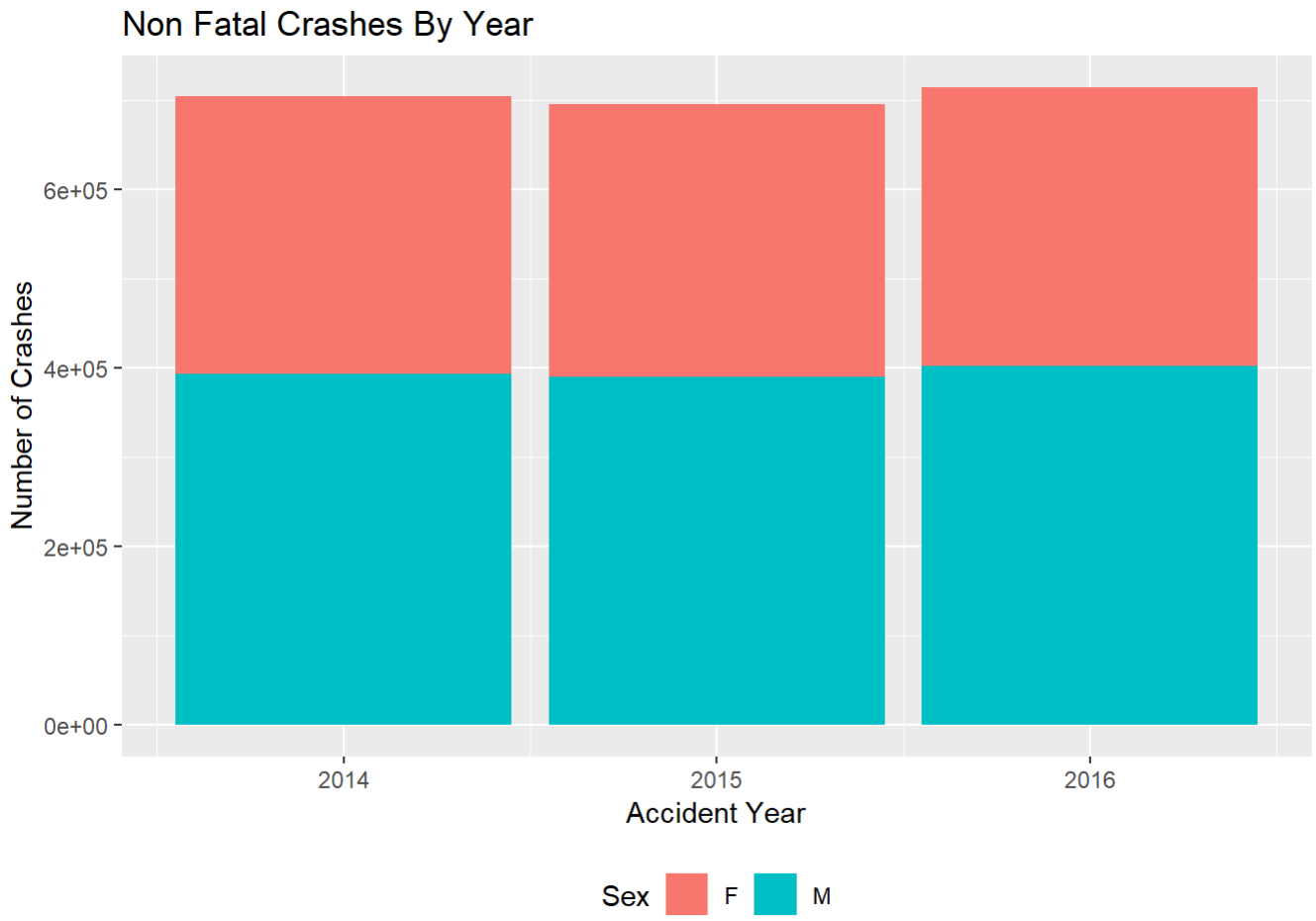
```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# Clear objects
rm(list=ls())

# Load nys-motor-vehicle-crashes-and-insurance-reduction  from kaggle datasets
# https://www.kaggle.com/new-york-state/nys-motor-vehicle-crashes-and-insurance-reduction

mv.crashes.by.facility <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes
-and-insurance-reduction/motor-vehicle-crashes-by-facility-port-authority-of-ny-nj-beginning-200
0.csv")
mv.crashes.case.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-a
nd-insurance-reduction/motor-vehicle-crashes-case-information-three-year-window.csv",stringsAsFa
ctors = FALSE)
mv.crashes.individual.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-cra
shes-and-insurance-reduction/motor-vehicle-crashes-individual-information-three-year-window.csv"
)
mv.crashes.vehicle.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashe
s-and-insurance-reduction/motor-vehicle-crashes-vehicle-information-three-year-window.csv")
mv.crashes.violation.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-cras
hes-and-insurance-reduction/motor-vehicle-crashes-violation-information-three-year-window.csv")
mv.pirp.participation <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-
and-insurance-reduction/motor-vehicle-point-insurance-reduction-program-pirp-participation-five-
year-window.csv")
```

```
#detach("package:RMySQL", unload=TRUE)

# Load Individual and Vehicle Data
#mv.crashes.individual.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-and-insurance-reduction/motor-vehicle-crashes-individual-information-three-year-window.csv")
#mv.crashes.vehicle.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-and-insurance-reduction/motor-vehicle-crashes-vehicle-information-three-year-window.csv")


# Merge into one dataset
mv.crashes.individual.vehicle.info <- merge(mv.crashes.individual.info,mv.crashes.vehicle.info,by.x = c("Case.Vehicle.ID"),by.y =c("Case.Vehicle.ID"))

# Response variable is Injury Severity and flagging fatal and non-fatal crashes
Ref.Injury.Severity <- unique(mv.crashes.individual.vehicle.info$Injury.Severity)
Ref.Injury.Severity <-data.frame(Ref.Injury.Severity.id=as.character(seq_len(length(Ref.Injury.Severity))),Injury.Severity=Ref.Injury.Severity)
Ref.Injury.Severity <-cbind(Ref.Injury.Severity,Fatal.Ind=ifelse(as.numeric(Ref.Injury.Severity$Ref.Injury.Severity.id=="6")==1,"Y","N"))

# Link Fatal/nonFatal flag to the main dataset
mv.crashes.individual.vehicle.info <- merge(mv.crashes.individual.vehicle.info,Ref.Injury.Severity,by.x = c("Injury.Severity"),by.y = c("Injury.Severity"))


my_veh_dist<-sqldf(' select `Year.x`,`Sex`,`Age`,`Fatal.Ind`,`Injury.Severity`,`Injury.Descriptor`,`Action.Prior.to.Accident`,count(1) cnt from `mv.crashes.individual.vehicle.info` group by `Year.x`,`Sex`,`Age`,`Fatal.Ind`,`Injury.Severity`,`Injury.Descriptor`,`Action.Prior.to.Accident`
 order by `Year.x`,`Sex`,`Age`,`Fatal.Ind`,`Injury.Severity`,`Injury.Descriptor`,`Action.Prior.to.Accident` ')

my_veh_dist <- my_veh_dist[which(my_veh_dist$Sex=="F" | my_veh_dist$Sex=="M"),]

my_veh_dist.f <-my_veh_dist[which(my_veh_dist$Fatal.Ind=="Y"),]
my_veh_dist.nf <-my_veh_dist[which(my_veh_dist$Fatal.Ind=="N"),]

ggplot() + geom_bar(data = my_veh_dist.f,aes(x=my_veh_dist.f$Year.x,y=my_veh_dist.f$cnt,fill=my_veh_dist.f$Sex),stat="identity")+labs (x="Accident Year",y="Number of Crashes",title = "Fatal Crashes By Year",fill="Sex") + theme(legend.position = "bottom")
```

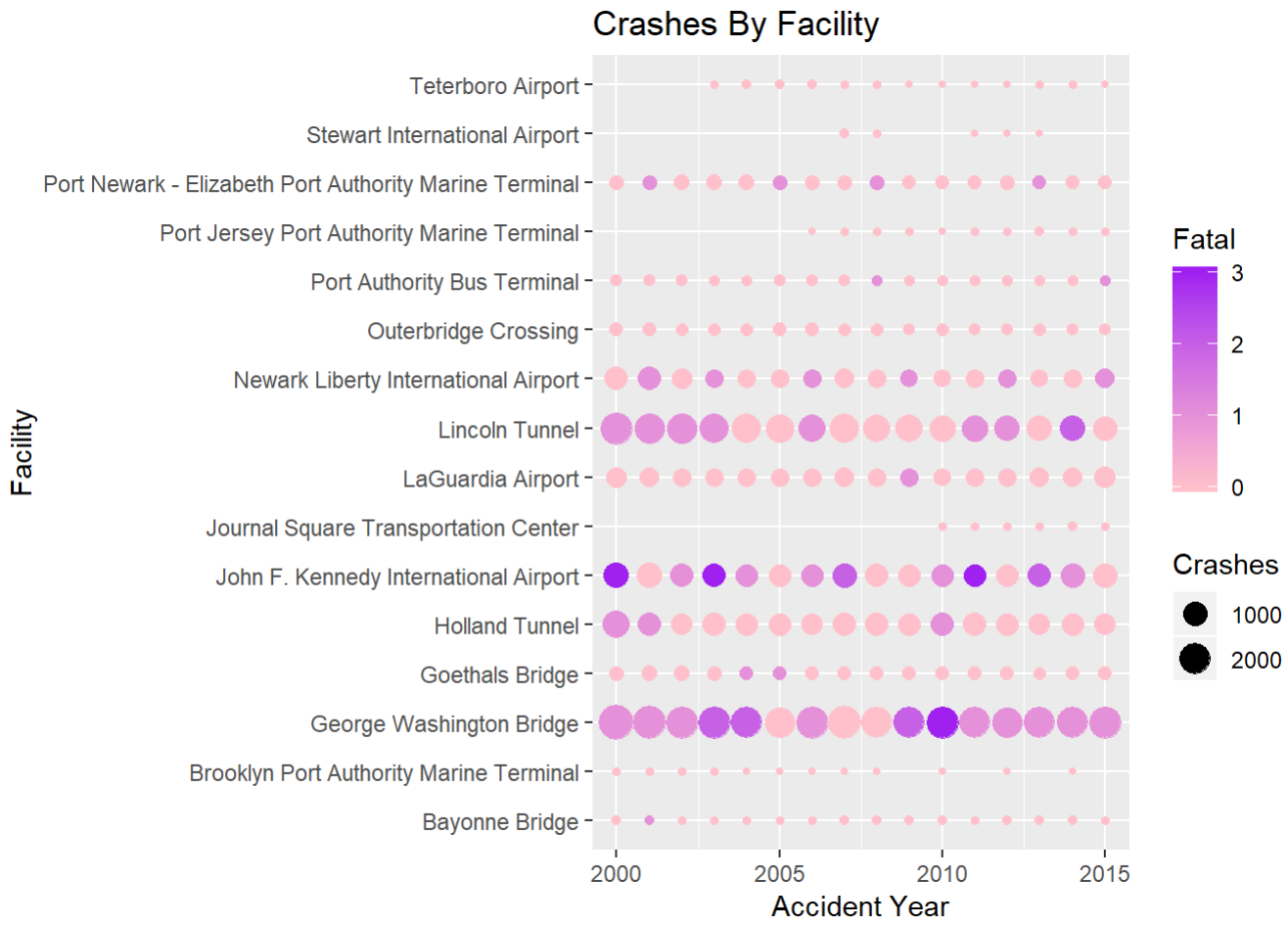



```
ggplot() + geom_bar(data = my_veh_dist.nf,aes(x=my_veh_dist.nf$Year.x,y=my_veh_dist.nf$cnt,fill=
my_veh_dist.nf$Sex),stat="identity")+labs (x="Accident Year",y="Number of Crashes",title = "Non
 Fatal Crashes By Year",fill="Sex") + theme(legend.position = "bottom")
```

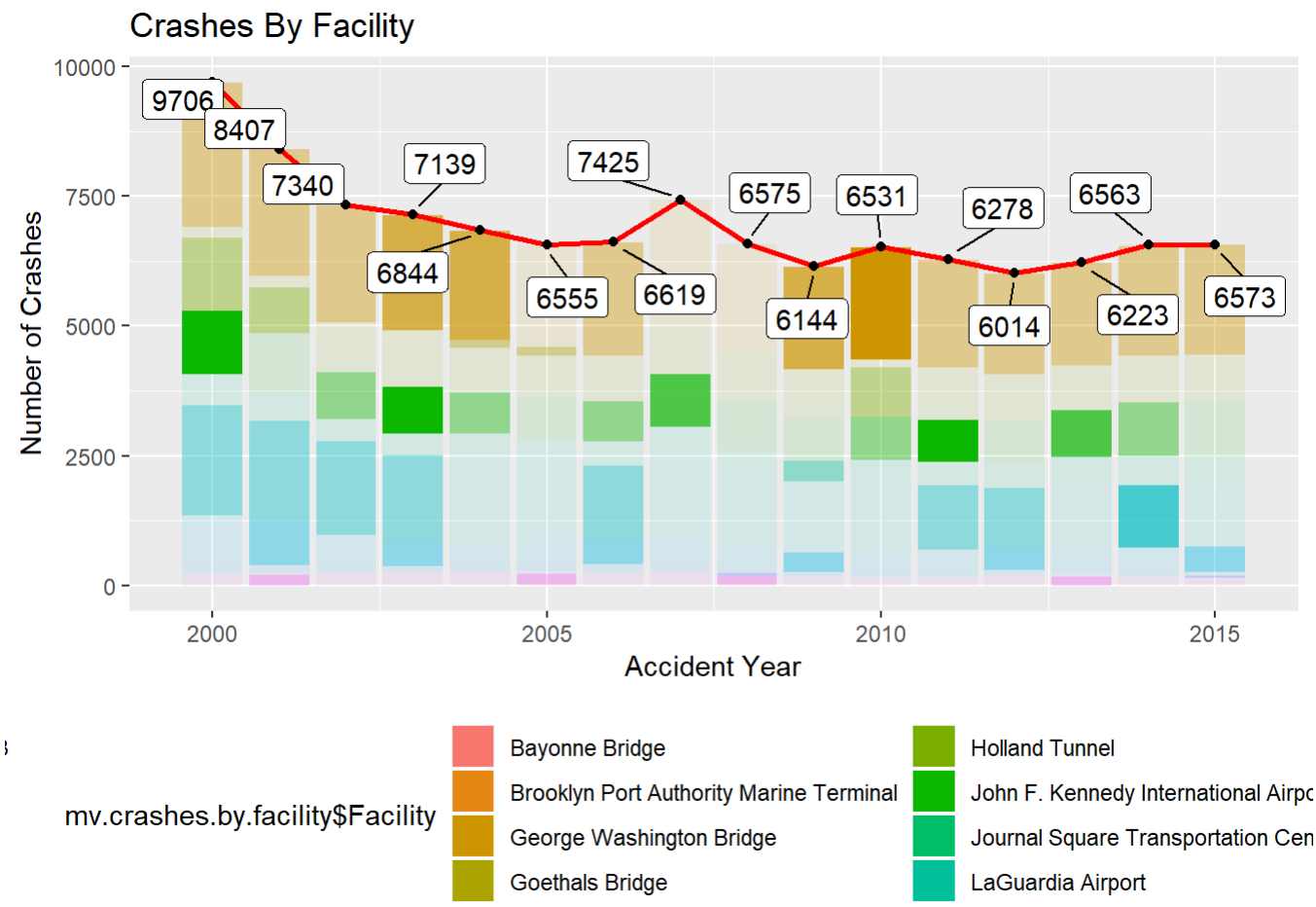



```
# Load he Facility Data
#mv.crashes.by.facility <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-and-insurance-reduction/motor-vehicle-crashes-by-facility-port-authority-of-ny-nj-beginning-2000.csv")

# Crashes By Facility
ggplot(data = mv.crashes.by.facility) + geom_point(aes(x=mv.crashes.by.facility$Year , y=mv.crashes.by.facility$Facility,size=ifelse(mv.crashes.by.facility$Total..Number.of.Motor.Vehicle.Crashes==0, NA, mv.crashes.by.facility$Total..Number.of.Motor.Vehicle.Crashes),color=mv.crashes.by.facility$Number.of.Fatal.Crashes)) +labs (x="Accident Year",y="Facility",title = "Crashes By Facility",color="Fatal",size="Crashes") +scale_color_continuous(low = "pink",high = "purple")
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```




```
# Get the total crashes in a da dataset
crashes <-tapply(mv.crashes.by.facility$Total..Number.of.Motor.Vehicle.Crashes , mv.crashes.by.facility$Year, sum)
crashes <-data.frame(year=as.numeric(names(crashes)),total.crashes=crashes)
rownames(crashes)<-NULL

# Crashes by Facility  in bar chart
ggplot() + geom_bar(data = mv.crashes.by.facility,aes(x=mv.crashes.by.facility$Year,y=mv.crashes.by.facility$Total..Number.of.Motor.Vehicle.Crashes,fill=mv.crashes.by.facility$Facility,alpha=mv.crashes.by.facility$Number.of.Fatal.Crashes),stat="identity")+labs (x="Accident Year",y="Number of Crashes",title = "Crashes By Facility") + theme(legend.position = "bottom") + geom_line(data=crashes,aes(x=year,y=total.crashes),size=1,color="red") +geom_point(data =crashes,aes(x=year,y=total.crashes)) + geom_label_repel(data =crashes,aes(x=year,y=total.crashes,label=crashes$total.crashes),box.padding = 0.35,point.padding = 0.5)
```




```
library(RMySQL)
```

```
## Loading required package: DBI
```

```
##
## Attaching package: 'RMySQL'
```

```
## The following object is masked from 'package:RSQLite':
##
##     isIdCurrent
```

```
library(DBI)
library(NLP)
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(tm)
library(RColorBrewer)
library(wordcloud)

# Load crashes by individual data
#mv.crashes.individual.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-and-insurance-reduction/motor-vehicle-crashes-individual-information-three-year-window.csv")

#  Apply Text mining on Injury Descriptor and get the most frequequent injuries
mv.crashes.individual.info$Injury.Descriptor<-tolower(mv.crashes.individual.info$Injury.Descriptor)
inj.vec <- VectorSource(mv.crashes.individual.info$Injury.Descriptor)
inj.corpus <- Corpus(inj.vec)
inj.corpus
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 2221172
```

```
inj.tdm <- TermDocumentMatrix(inj.corpus)
str(inj.tdm)
```

```
## List of 6
##  $ i       : int [1:4569518] 1 2 3 4 3 4 5 4 6 7 ...
##  $ j       : int [1:4569518] 1 1 2 2 3 3 5 5 6 6 ...
##  $ v       : num [1:4569518] 1 1 1 1 1 1 1 1 1 1 ...
##  $ nrow    : int 55
##  $ ncol    : int 2221172
##  $ dimnames:List of 2
##   ..$ Terms: chr [1:55] "complaint" "pain" "applicable" "not" ...
##   ..$ Docs : chr [1:2221172] "1" "2" "3" "4" ...
##  - attr(*, "class")= chr [1:2] "TermDocumentMatrix" "simple_triplet_matrix"
##  - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
inspect(inj.tdm[1:20,1:20])
```

```
## <<TermDocumentMatrix (terms: 20, documents: 20)>>
## Non-/sparse entries: 34/366
## Sparsity           : 92%
## Maximal term length: 16
## Weighting          : term frequency (tf)
## Sample             :
##              Docs
## Terms         1 10 11 12 2 3 5 6 7 9
##   (no         0  0  0  0 0 0 0 0 0 0
##   applicable  0  1  0  1 1 1 0 0 0 1
##   complaint   1  0  0  0 0 0 0 0 0 0
##   dislocation 0  0  0  0 0 0 0 1 0 0
##   entered     0  0  1  0 0 0 1 0 1 0
##   fracture    0  0  0  0 0 0 0 1 0 0
##   none        0  0  0  0 0 0 0 0 0 0
##   not         0  1  1  1 1 1 1 0 1 1
##   pain        1  0  0  0 0 0 0 0 0 0
##   visible     0  0  0  0 0 0 0 0 0 0
```

```
inj.m <- as.matrix(inj.tdm)
str(inj.m)
```

```
##  num [1:55, 1:2221172] 1 1 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Terms: chr [1:55] "complaint" "pain" "applicable" "not" ...
##   ..$ Docs : chr [1:2221172] "1" "2" "3" "4" ...
```

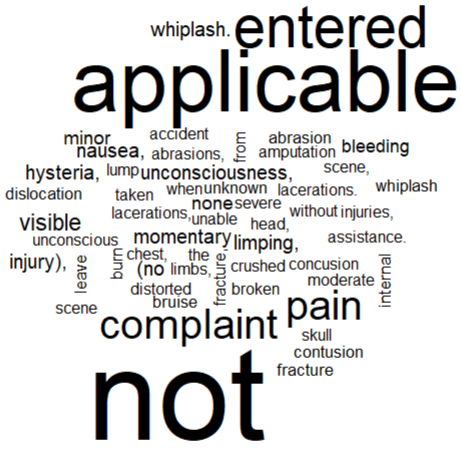
```
inj.word.cnts <- rowSums(inj.m)
head(inj.word.cnts)
```

```
##   complaint        pain  applicable         not     entered dislocation
##      342529      342529      959613     1529962      570349        8227
```

```
inj.word.cnts <- sort(inj.word.cnts,decreasing = TRUE)
head(inj.word.cnts)
```

```
##        not applicable    entered  complaint       pain    visible
##    1529962     959613     570349     342529     342529      97648
```

```
wordcloud(names(inj.word.cnts),inj.word.cnts)
```




```
# Remove unwanted words
inj.corpus <- tm_map(inj.corpus,removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(inj.corpus, removePunctuation):
## transformation drops documents
```

```
inj.corpus <- tm_map(inj.corpus,removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(inj.corpus, removeNumbers): transformation
## drops documents
```

```
inj.corpus <- tm_map(inj.corpus,removeWords,stopwords("english"))
```

```
## Warning in tm_map.SimpleCorpus(inj.corpus, removeWords,
## stopwords("english")): transformation drops documents
```

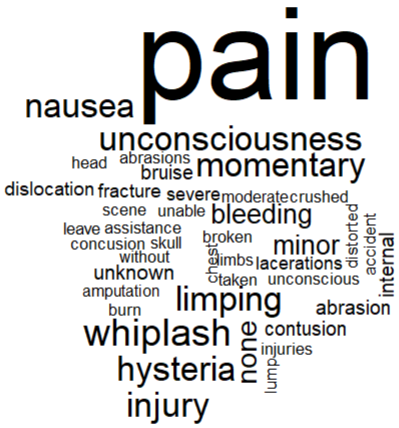
```
ExcludeWords <- c("applicable", "entered", "complaint","visible")
inj.corpus <- tm_map(inj.corpus,removeWords,ExcludeWords)
```

```
## Warning in tm_map.SimpleCorpus(inj.corpus, removeWords, ExcludeWords):
## transformation drops documents
```

```
# Recreate the TDM
inj.tdm <- TermDocumentMatrix(inj.corpus)
inj.m <- as.matrix(inj.tdm)
inj.word.cnts <- rowSums(inj.m)
inj.word.cnts <- sort(inj.word.cnts,decreasing = TRUE)
head(inj.word.cnts)
```

```
##      pain  whiplash  hysteria    injury   limping momentary
##    342529     68213     57031     57031     57031     57031
```

```
wordcloud(names(inj.word.cnts),inj.word.cnts)
```




```
# Creating image of the word cloud and storing locally
color_theme <- brewer.pal(8,"Dark2")
png("Injury_Descriptor.png", width=12,height=8, units='in', res=300)
wordcloud(names(inj.word.cnts),inj.word.cnts,scale=c(5,.3),min.freq =500 ,max.words =100,colors
 = color_theme )
dev.off()
```

```
## png
##   2
```

```
wordcloud(names(inj.word.cnts),inj.word.cnts,min.freq =1000 ,max.words =50 ,rot.per =10 )
```


```
# Load crashes by Vehicle data
#mv.crashes.vehicle.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crash
es-and-insurance-reduction/motor-vehicle-crashes-vehicle-information-three-year-window.csv")
#mv.crashes.violation.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-cra
shes-and-insurance-reduction/motor-vehicle-crashes-violation-information-three-year-window.csv")

# Text Mining on Contibution Factor 1 and 2 to see the leading contibutions for accidents
mv.crashes.vehicle.info$Contributing.Factor.1.Description<-tolower(mv.crashes.vehicle.info$Contr
ibuting.Factor.1.Description)
mv.crashes.vehicle.info$Contributing.Factor.2.Description<-tolower(mv.crashes.vehicle.info$Contr
ibuting.Factor.2.Description)
Contributing.Factor <-rbind(mv.crashes.vehicle.info$Contributing.Factor.1.Description,mv.crashe
s.vehicle.info$Contributing.Factor.2.Description)

cf.vec <- VectorSource(Contributing.Factor)
cf.corpus <- Corpus(cf.vec)
cf.corpus
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 3308964
```

```
cf.tdm <- TermDocumentMatrix(cf.corpus)
str(cf.tdm)
```

```
## List of 6
##  $ i       : int [1:7011638] 1 2 1 2 3 1 2 1 2 1 ...
##  $ j       : int [1:7011638] 1 1 2 2 3 4 4 5 5 6 ...
##  $ v       : num [1:7011638] 1 1 1 1 1 1 1 1 1 1 ...
##  $ nrow    : int 105
##  $ ncol    : int 3308964
##  $ dimnames:List of 2
##   ..$ Terms: chr [1:105] "entered" "not" "unknown" "driver" ...
##   ..$ Docs : chr [1:3308964] "1" "2" "3" "4" ...
##  - attr(*, "class")= chr [1:2] "TermDocumentMatrix" "simple_triplet_matrix"
##  - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
inspect(cf.tdm[1:20,1:20])
```

```
## <<TermDocumentMatrix (terms: 20, documents: 20)>>
## Non-/sparse entries: 38/362
## Sparsity           : 90%
## Maximal term length: 24
## Weighting          : term frequency (tf)
## Sample             :
##                         Docs
## Terms                    1 10 11 12 2 4 5 6 8 9
##   alcohol                0  0  0  0 0 0 0 0 0 0
##   applicable             0  0  0  0 0 0 0 0 0 0
##   closely                0  0  0  0 0 0 0 0 0 0
##   driver                 0  0  0  0 0 0 0 0 0 0
##   entered                1  1  1  1 1 1 1 1 1 1
##   following              0  0  0  0 0 0 0 0 0 0
##   inattention/distraction* 0  0  0  0 0 0 0 0 0 0
##   involvement            0  0  0  0 0 0 0 0 0 0
##   not                    1  1  1  1 1 1 1 1 1 1
##   unknown                0  0  0  0 0 0 0 0 0 0
```

```
cf.m <- as.matrix(cf.tdm)
str(cf.m)
```

```
##  num [1:105, 1:3308964] 1 1 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Terms: chr [1:105] "entered" "not" "unknown" "driver" ...
##   ..$ Docs : chr [1:3308964] "1" "2" "3" "4" ...
```

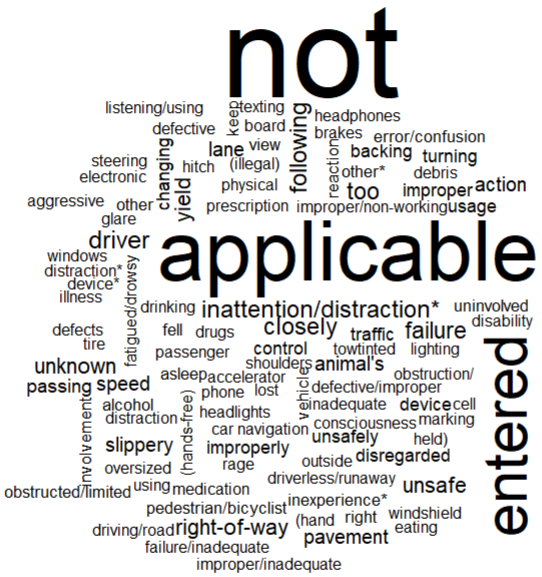
```
cf.word.cnts <- rowSums(cf.m)
head(cf.word.cnts)
```

```
##                entered                    not                unknown
##                 852283                2151584                 119351
##                 driver inattention/distraction*              applicable
##                 177455                 160037                1299301
```

```
cf.word.cnts <- sort(cf.word.cnts,decreasing = TRUE)
head(cf.word.cnts)
```

```
##                    not              applicable                 entered
##                2151584                 1299301                  852283
##                 driver inattention/distraction*                 closely
##                 177455                  160037                  149673
```

```
wordcloud(names(cf.word.cnts),cf.word.cnts)
```




```
# Remove unwanted words
cf.corpus <- tm_map(cf.corpus,removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(cf.corpus, removePunctuation):
## transformation drops documents
```

```
cf.corpus <- tm_map(cf.corpus,removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(cf.corpus, removeNumbers): transformation
## drops documents
```

```
cf.corpus <- tm_map(cf.corpus,removeWords,stopwords("english"))
```

```
## Warning in tm_map.SimpleCorpus(cf.corpus, removeWords,
## stopwords("english")): transformation drops documents
```

```
ExcludeWords <- c("applicable", "entered", "complaint","visible")
cf.corpus <- tm_map(cf.corpus,removeWords,ExcludeWords)
```

```
## Warning in tm_map.SimpleCorpus(cf.corpus, removeWords, ExcludeWords):
## transformation drops documents
```

```
# Recreate the TDM
cf.tdm <- TermDocumentMatrix(cf.corpus)
cf.m <- as.matrix(cf.tdm)
cf.word.cnts <- rowSums(cf.m)
cf.word.cnts <- sort(cf.word.cnts,decreasing = TRUE)
head(cf.word.cnts)
```

```
##                driver inattentiondistraction               closely
##                177455                160037                149673
##             following               failure            rightofway
##                144638                140807                124333
```

```
wordcloud(names(cf.word.cnts),cf.word.cnts)
```

```
## Warning in wordcloud(names(cf.word.cnts), cf.word.cnts):
## inattentiondistraction could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(names(cf.word.cnts), cf.word.cnts): rightofway could
## not be fit on page. It will not be plotted.
```


```
# Creating image of the word cloud and storing locally

color_theme <- brewer.pal(8,"Dark2")
png("contributing_factor.png", width=12,height=8, units='in', res=300)
wordcloud(names(cf.word.cnts),cf.word.cnts,scale=c(5,.3),min.freq =500 ,max.words =100,colors =
 color_theme )
dev.off()
```

```
## png
##   2
```

```
wordcloud(names(cf.word.cnts),cf.word.cnts,min.freq =1000 ,max.words =50 ,rot.per =10 )
```

```
## Warning in wordcloud(names(cf.word.cnts), cf.word.cnts, min.freq = 1000, :
## inattentiondistraction could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(names(cf.word.cnts), cf.word.cnts, min.freq = 1000, :
## following could not be fit on page. It will not be plotted.
```

unknown
slippery
driver
lane
rightofway
uninvolved
disregarded
involvement
brakes
devices
drivingroad
pedestrianbicyclist
unsafe
traffic
obstruction
erlychanging
improper
asleep
ssing
turning
usage
rage
fell
unsafely
aggressive
obstructedlimited
debris
view
right
alcohol
distraction
defective
errorconfusion
control
vehicle
speed
closely
backing
keep
reaction
nexperience
animals
glare
ment
action
yield
failure

```
#  Focusing only on the mv.crashes.case.info dataset to to some predictions on given the conditi
on how likely the accident will result in a fatal one

#mv.crashes.case.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-
and-insurance-reduction/motor-vehicle-crashes-case-information-three-year-window.csv",stringsAsF
actors = FALSE)

# Data preparation to convert all categorical values into a numeric string to avoid memory issu
e. when i ran the model with the description I had memory issue cannot allocate vector of size
 6.7 Gb

Ref.Crash.Descriptor <- unique(mv.crashes.case.info$Crash.Descriptor)
Ref.Crash.Descriptor <-data.frame(Crash.Descriptor.id=as.character(seq_len(length(Ref.Crash.Desc
riptor))),Crash.Descriptor=Ref.Crash.Descriptor)
Ref.Crash.Descriptor <-cbind(Ref.Crash.Descriptor,Fatal.Ind=ifelse(as.numeric(Ref.Crash.Descript
or$Crash.Descriptor.id=="4")==1,"Y","N"))
Ref.Crash.Descriptor$Crash.Descriptor <- as.character(Ref.Crash.Descriptor$Crash.Descriptor)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Crash.Descriptor,by.x = c("Crash.Descript
or"),by.y = c("Crash.Descriptor"),all.x = TRUE)


Ref.Lighting.Conditions <- unique(mv.crashes.case.info$Lighting.Conditions)
Ref.Lighting.Conditions <-data.frame(Lighting.Conditions.id=as.character(seq_len(length(Ref.Ligh
ting.Conditions))),Lighting.Conditions=Ref.Lighting.Conditions)
Ref.Lighting.Conditions$Lighting.Conditions <- as.character(Ref.Lighting.Conditions$Lighting.Con
ditions)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Lighting.Conditions,by.x = c("Lighting.Co
nditions"),by.y = c("Lighting.Conditions"),all.x = TRUE)

Ref.Collision.Type.Descriptor <- unique(mv.crashes.case.info$Collision.Type.Descriptor)
Ref.Collision.Type.Descriptor <-data.frame(Collision.Type.Descriptor.id=as.character(seq_len(len
gth(Ref.Collision.Type.Descriptor))),Collision.Type.Descriptor=Ref.Collision.Type.Descriptor)
Ref.Collision.Type.Descriptor$Collision.Type.Descriptor <- as.character(Ref.Collision.Type.Descr
iptor$Collision.Type.Descriptor)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Collision.Type.Descriptor,by.x = c("Colli
sion.Type.Descriptor"),by.y = c("Collision.Type.Descriptor"),all.x = TRUE)

Ref.Road.Descriptor <- unique(mv.crashes.case.info$Road.Descriptor)
Ref.Road.Descriptor <-data.frame(Road.Descriptor.id=as.character(seq_len(length(Ref.Road.Descrip
tor))),Road.Descriptor=Ref.Road.Descriptor)
Ref.Road.Descriptor$Road.Descriptor <- as.character(Ref.Road.Descriptor$Road.Descriptor)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Road.Descriptor,by.x = c("Road.Descripto
r"),by.y = c("Road.Descriptor"),all.x = TRUE)

Ref.Weather.Conditions <- unique(mv.crashes.case.info$Weather.Conditions)
Ref.Weather.Conditions <-data.frame(Weather.Conditions.id=as.character(seq_len(length(Ref.Weathe
r.Conditions))),Weather.Conditions=Ref.Weather.Conditions)
Ref.Weather.Conditions$Weather.Conditions <- as.character(Ref.Weather.Conditions$Weather.Conditi
ons)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Weather.Conditions,by.x = c("Weather.Cond
itions"),by.y = c("Weather.Conditions"),all.x = TRUE)

Ref.Traffic.Control.Device <- unique(mv.crashes.case.info$Traffic.Control.Device)
```

```
Ref.Traffic.Control.Device <-data.frame(Traffic.Control.Device.id=as.character(seq_len(length(Re
f.Traffic.Control.Device))),Traffic.Control.Device=Ref.Traffic.Control.Device)
Ref.Traffic.Control.Device$Traffic.Control.Device <- as.character(Ref.Traffic.Control.Device$Tra
ffic.Control.Device)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Traffic.Control.Device,by.x = c("Traffic.
Control.Device"),by.y = c("Traffic.Control.Device"),all.x = TRUE)

Ref.Road.Surface.Conditions <- unique(mv.crashes.case.info$Road.Surface.Conditions)
Ref.Road.Surface.Conditions <-data.frame(Road.Surface.Conditions.id=as.character(seq_len(length
(Ref.Road.Surface.Conditions))),Road.Surface.Conditions=Ref.Road.Surface.Conditions)
Ref.Road.Surface.Conditions$Road.Surface.Conditions <- as.character(Ref.Road.Surface.Conditions
$Road.Surface.Conditions)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Road.Surface.Conditions,by.x = c("Road.Su
rface.Conditions"),by.y = c("Road.Surface.Conditions"),all.x = TRUE)

Ref.Pedestrian.Bicyclist.Action <- unique(mv.crashes.case.info$Pedestrian.Bicyclist.Action)
Ref.Pedestrian.Bicyclist.Action <-data.frame(Pedestrian.Bicyclist.Action.id=as.character(seq_len
(length(Ref.Pedestrian.Bicyclist.Action))),Pedestrian.Bicyclist.Action=Ref.Pedestrian.Bicyclist.
Action)
Ref.Pedestrian.Bicyclist.Action$Pedestrian.Bicyclist.Action <- as.character(Ref.Pedestrian.Bicyc
list.Action$Pedestrian.Bicyclist.Action)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Pedestrian.Bicyclist.Action,by.x = c("Ped
estrian.Bicyclist.Action"),by.y = c("Pedestrian.Bicyclist.Action"),all.x = TRUE)

Ref.Event.Descriptor <- unique(mv.crashes.case.info$Event.Descriptor)
Ref.Event.Descriptor <-data.frame(Event.Descriptor.id=seq_len(length(Ref.Event.Descriptor)),Even
t.Descriptor=Ref.Event.Descriptor)
Ref.Event.Descriptor$Event.Descriptor <- as.character(Ref.Event.Descriptor$Event.Descriptor)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Event.Descriptor,by.x = c("Event.Descript
or"),by.y = c("Event.Descriptor"),all.x = TRUE)

mv.crashes.case.info.map <- mv.crashes.case.info[,10:28]
mv.crashes.case.info.map <- mv.crashes.case.info.map[,-8]
mv.crashes.case.info.map$Time <- as.factor(mv.crashes.case.info.map$Time)
mv.crashes.case.info.map$Day.of.Week <-as.factor(mv.crashes.case.info.map$Day.of.Week)
mv.crashes.case.info.map$Police.Report <-as.factor(mv.crashes.case.info.map$Police.Report)
mv.crashes.case.info.map$Municipality <-as.factor(mv.crashes.case.info.map$Municipality)
mv.crashes.case.info.map$County.Name <-as.factor(mv.crashes.case.info.map$County.Name)
mv.crashes.case.info.map$Date <- substring(mv.crashes.case.info.map$Date,0, regexpr("T", mv.cras
hes.case.info.map$Date)-1)
mv.crashes.case.info.map$Date <-as.factor(mv.crashes.case.info.map$Date)
mv.crashes.case.info.map$TimeHr <- substring(mv.crashes.case.info.map$Time,0, regexpr(":", mv.cr
ashes.case.info.map$Time)-1)
mv.crashes.case.info.map$TimeMin <- substring(mv.crashes.case.info.map$Time, regexpr(":", mv.cra
shes.case.info.map$Time)+1)
mv.crashes.case.info.map$Month <-substring(mv.crashes.case.info.map$Date,6,7)
mv.crashes.case.info.map$Dt <-substring(mv.crashes.case.info.map$Date,9,10)



# remove unwanted columns from the model dataset
mv.crashes.case.info.map <- mv.crashes.case.info.map [,-(6:7)]
mv.crashes.case.info.map <- mv.crashes.case.info.map [,-(2)]
mv.crashes.case.info.map <- mv.crashes.case.info.map [,-2]
```

```
mv.crashes.case.info.map <- mv.crashes.case.info.map [,-5]


# train the model with 100000 records
mv.crashes.case.info.map <- mv.crashes.case.info.map[1:100000,]

model.mv <- randomForest(mv.crashes.case.info.map$Fatal.Ind ~mv.crashes.case.info.map$Year + mv.
crashes.case.info.map$Day.of.Week +mv.crashes.case.info.map$Month + mv.crashes.case.info.map$Tim
eHr +mv.crashes.case.info.map$Number.of.Vehicles.Involved + mv.crashes.case.info.map$Lighting.Co
nditions.id+ mv.crashes.case.info.map$Collision.Type.Descriptor.id +mv.crashes.case.info.map$Roa
d.Descriptor.id +mv.crashes.case.info.map$Weather.Conditions.id +mv.crashes.case.info.map$Traffi
c.Control.Device.id +mv.crashes.case.info.map$Road.Surface.Conditions.id +mv.crashes.case.info.m
ap$Pedestrian.Bicyclist.Action.id +mv.crashes.case.info.map$Event.Descriptor.id ,mv.crashes.cas
e.info.map,ntree=250)

# Summary of the Random Forest results
summary(model.mv)
```

```
##                 Length Class  Mode
## call                 4 -none- call
## type                 1 -none- character
## predicted       100000 factor numeric
## err.rate           750 -none- numeric
## confusion            6 -none- numeric
## votes           200000 matrix numeric
## oob.times       100000 -none- numeric
## classes              2 -none- character
## importance          13 -none- numeric
## importanceSD         0 -none- NULL
## localImportance      0 -none- NULL
## proximity            0 -none- NULL
## ntree                1 -none- numeric
## mtry                 1 -none- numeric
## forest              14 -none- list
## y               100000 factor numeric
## test                 0 -none- NULL
## inbag                0 -none- NULL
## terms                3 terms  call
```

```
# Confusion Matrix
model.mv$confusion
```

```
##       N Y  class.error
## N 99757 1 1.002426e-05
## Y   242 0 1.000000e+00
```

```
# importance matrix
importance(model.mv)
```

```
##                                                        MeanDecreaseGini
## mv.crashes.case.info.map$Year                                 22.505111
## mv.crashes.case.info.map$Day.of.Week                          44.404701
## mv.crashes.case.info.map$Month                                48.974295
## mv.crashes.case.info.map$TimeHr                               56.490251
## mv.crashes.case.info.map$Number.of.Vehicles.Involved           8.853824
## mv.crashes.case.info.map$Lighting.Conditions.id               15.991343
## mv.crashes.case.info.map$Collision.Type.Descriptor.id          1.828480
## mv.crashes.case.info.map$Road.Descriptor.id                   23.347154
## mv.crashes.case.info.map$Weather.Conditions.id                15.727324
## mv.crashes.case.info.map$Traffic.Control.Device.id            23.879064
## mv.crashes.case.info.map$Road.Surface.Conditions.id            9.380486
## mv.crashes.case.info.map$Pedestrian.Bicyclist.Action.id       24.088079
## mv.crashes.case.info.map$Event.Descriptor.id                  20.909605
```

```
# Number of decision trees used by the Random Forest
model.mv$ntree
```

```
## [1] 250
```

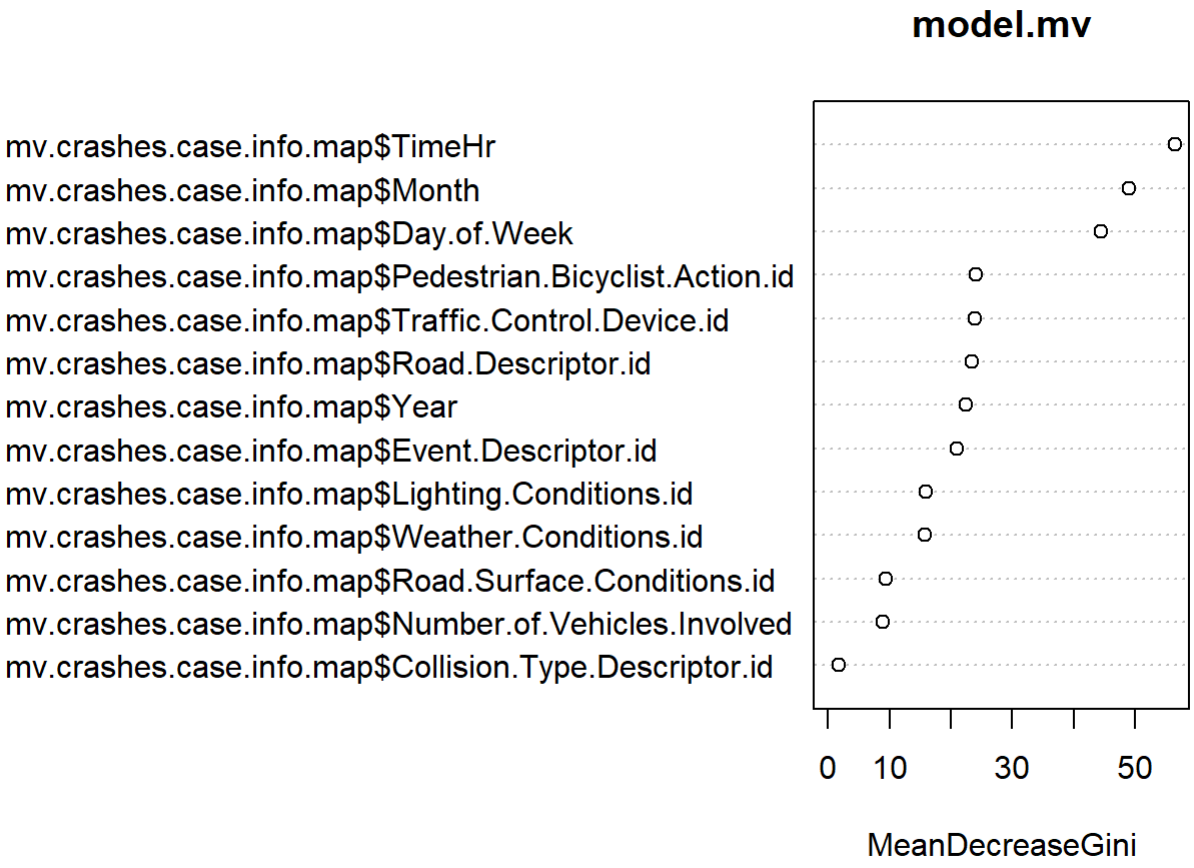
```
# Type of algorithm used by the Random Forest
model.mv$type
```

```
## [1] "classification"
```

```
# Class Error
model.mv$confusion[,'class.error']
```

```
##           N           Y
## 1.002426e-05 1.000000e+00
```

```
# Variable importance plot
varImpPlot(model.mv)
```


model.mv
mv.crashes.case.info.map$TimeHr
mv.crashes.case.info.map$Month
mv.crashes.case.info.map$Day.of.Week
mv.crashes.case.info.map$Pedestrian.Bicyclist.Action.id
mv.crashes.case.info.map$Traffic.Control.Device.id
mv.crashes.case.info.map$Road.Descriptor.id
mv.crashes.case.info.map$Year
mv.crashes.case.info.map$Event.Descriptor.id
mv.crashes.case.info.map$Lighting.Conditions.id
mv.crashes.case.info.map$Weather.Conditions.id
mv.crashes.case.info.map$Road.Surface.Conditions.id
mv.crashes.case.info.map$Number.of.Vehicles.Involved
mv.crashes.case.info.map$Collision.Type.Descriptor.id
0
10
30
50
MeanDecreaseGini

```
# clear some memory to traing the model in R , Keep only the model variable dataset and remove a
ll others
#rm(list = ls())
#gc()

mv.crashes.case.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes-a
nd-insurance-reduction/motor-vehicle-crashes-case-information-three-year-window.csv",stringsAsFa
ctors = FALSE)



# Data preparation to convert all categorical values into a numeric string to avoid memory issu
e. when i ran the model with the description I had memory issue cannot allocate vector of size
 6.7 Gb

Ref.Crash.Descriptor <- unique(mv.crashes.case.info$Crash.Descriptor)
Ref.Crash.Descriptor <-data.frame(Crash.Descriptor.id=as.character(seq_len(length(Ref.Crash.Desc
riptor))),Crash.Descriptor=Ref.Crash.Descriptor)
Ref.Crash.Descriptor <-cbind(Ref.Crash.Descriptor,Fatal.Ind=ifelse(as.numeric(Ref.Crash.Descript
or$Crash.Descriptor.id=="4")==1,"Y","N"))
Ref.Crash.Descriptor$Crash.Descriptor <- as.character(Ref.Crash.Descriptor$Crash.Descriptor)
mv.crashes.case.info <- merge(mv.crashes.case.info,Ref.Crash.Descriptor,by.x = c("Crash.Descript
or"),by.y = c("Crash.Descriptor"),all.x = TRUE)

mv.crashes.case.info.map <- mv.crashes.case.info
mv.crashes.case.info.map$Time <- as.factor(mv.crashes.case.info.map$Time)
mv.crashes.case.info.map$Day.of.Week <-as.factor(mv.crashes.case.info.map$Day.of.Week)
mv.crashes.case.info.map$Police.Report <-as.factor(mv.crashes.case.info.map$Police.Report)
mv.crashes.case.info.map$Municipality <-as.factor(mv.crashes.case.info.map$Municipality)
mv.crashes.case.info.map$County.Name <-as.factor(mv.crashes.case.info.map$County.Name)
mv.crashes.case.info.map$Date <- substring(mv.crashes.case.info.map$Date,0, regexpr("T", mv.cras
hes.case.info.map$Date)-1)
mv.crashes.case.info.map$Date <-as.factor(mv.crashes.case.info.map$Date)
mv.crashes.case.info.map$TimeHr <- substring(mv.crashes.case.info.map$Time,0, regexpr(":", mv.cr
ashes.case.info.map$Time)-1)
mv.crashes.case.info.map$TimeMin <- substring(mv.crashes.case.info.map$Time, regexpr(":", mv.cra
shes.case.info.map$Time)+1)
mv.crashes.case.info.map$Month <-substring(mv.crashes.case.info.map$Date,6,7)
mv.crashes.case.info.map$Dt <-substring(mv.crashes.case.info.map$Date,9,10)

mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)=
="Municipality")]
mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)=
="DOT.Reference.Marker.Location")]
mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)=
="Date")]
mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)=
="Crash.Descriptor")]
#mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)
=="County.Name")]
mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)=
="Time")]
mv.crashes.case.info.map <- mv.crashes.case.info.map[,-which(colnames(mv.crashes.case.info.map)=
```

```
="Police.Report")]
character_vars <- lapply(mv.crashes.case.info.map, class) == "character"
mv.crashes.case.info.map[, character_vars] <- lapply(mv.crashes.case.info.map[, character_vars],
as.factor)

# train the model with 1st 100,000 records

mv.crashes.case.info.map <- mv.crashes.case.info.map[1:100000,]
#mv.crashes.case.info.map <- mv.crashes.case.info.map[which(mv.crashes.case.info.map$County.Name
=="NEW YORK"),]
model.mv <- randomForest(mv.crashes.case.info.map$Fatal.Ind ~ mv.crashes.case.info.map$Day.of.We
ek +mv.crashes.case.info.map$Month + mv.crashes.case.info.map$TimeHr +mv.crashes.case.info.map$N
umber.of.Vehicles.Involved + mv.crashes.case.info.map$Lighting.Conditions+ mv.crashes.case.info.
map$Collision.Type.Descriptor +mv.crashes.case.info.map$Road.Descriptor +mv.crashes.case.info.ma
p$Weather.Conditions +mv.crashes.case.info.map$Traffic.Control.Device +mv.crashes.case.info.map
$Road.Surface.Conditions +mv.crashes.case.info.map$Pedestrian.Bicyclist.Action +mv.crashes.case.
info.map$Event.Descriptor ,mv.crashes.case.info.map,ntree=500)


# Summary of the Random Forest results
summary(model.mv)
```

```
##                 Length Class  Mode
## call                 4 -none- call
## type                 1 -none- character
## predicted       100000 factor numeric
## err.rate          1500 -none- numeric
## confusion            6 -none- numeric
## votes           200000 matrix numeric
## oob.times       100000 -none- numeric
## classes              2 -none- character
## importance          12 -none- numeric
## importanceSD         0 -none- NULL
## localImportance      0 -none- NULL
## proximity            0 -none- NULL
## ntree                1 -none- numeric
## mtry                 1 -none- numeric
## forest              14 -none- list
## y               100000 factor numeric
## test                 0 -none- NULL
## inbag                0 -none- NULL
## terms                3 terms  call
```

```
# Confusion Matrix
model.mv$confusion
```

```
##       N   Y class.error
## N 96781 239  0.00246341
## Y  2729 251  0.91577181
```

```
# importance matrix
importance(model.mv)
```

```
##                                                    MeanDecreaseGini
## mv.crashes.case.info.map$Day.of.Week                       537.2864
## mv.crashes.case.info.map$Month                             683.7916
## mv.crashes.case.info.map$TimeHr                            836.4434
## mv.crashes.case.info.map$Number.of.Vehicles.Involved       136.3837
## mv.crashes.case.info.map$Lighting.Conditions               228.5311
## mv.crashes.case.info.map$Collision.Type.Descriptor         254.2744
## mv.crashes.case.info.map$Road.Descriptor                   240.3749
## mv.crashes.case.info.map$Weather.Conditions                225.4552
## mv.crashes.case.info.map$Traffic.Control.Device            304.3023
## mv.crashes.case.info.map$Road.Surface.Conditions           145.5570
## mv.crashes.case.info.map$Pedestrian.Bicyclist.Action       227.9216
## mv.crashes.case.info.map$Event.Descriptor                  477.2179
```

```
# Number of decision trees used by the Random Forest
model.mv$ntree
```

```
## [1] 500
```

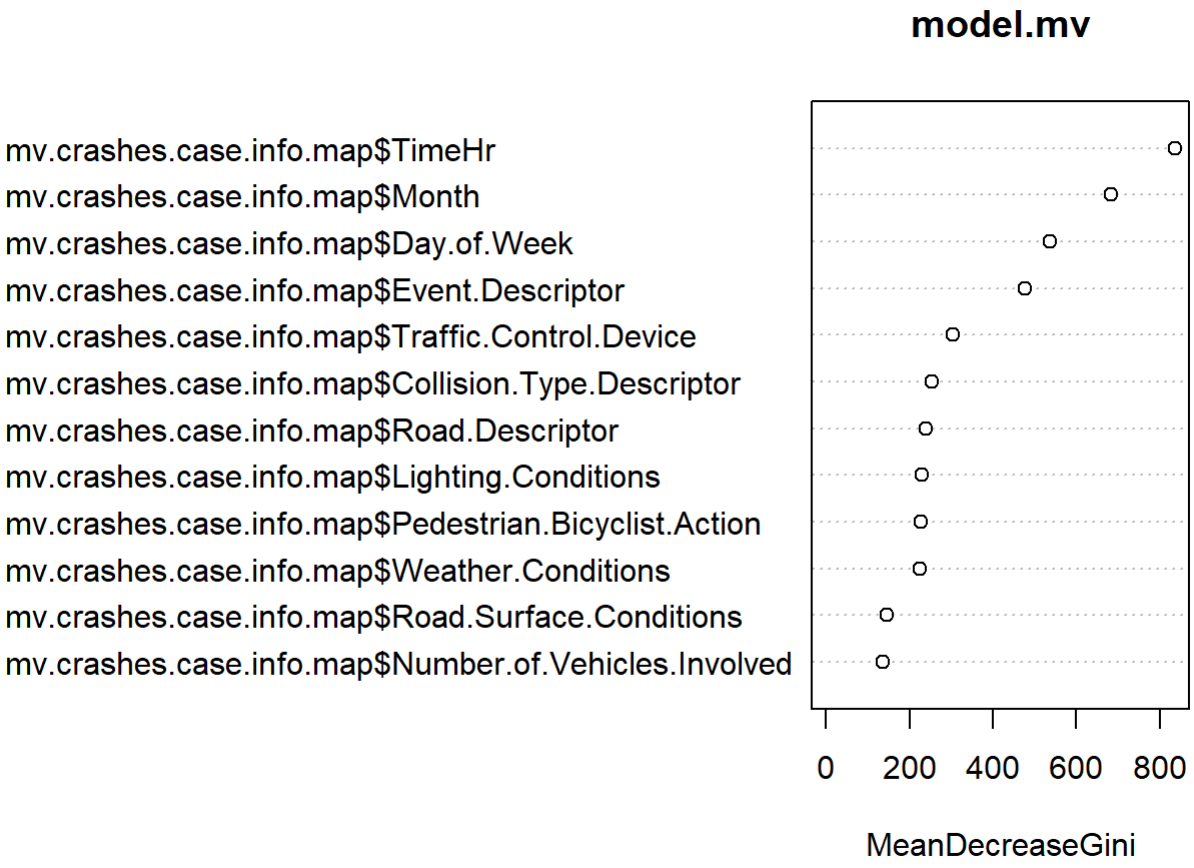
```
# Type of algorithm used by the Random Forest
model.mv$type
```

```
## [1] "classification"
```

```
# Class Error
model.mv$confusion[,'class.error']
```

```
##          N          Y
## 0.00246341 0.91577181
```

```
# Variable importance plot
varImpPlot(model.mv)
```


model.mv
mv.crashes.case.info.map$TimeHr
mv.crashes.case.info.map$Month
mv.crashes.case.info.map$Day.of.Week
mv.crashes.case.info.map$Event.Descriptor
mv.crashes.case.info.map$Traffic.Control.Device
mv.crashes.case.info.map$Collision.Type.Descriptor
mv.crashes.case.info.map$Road.Descriptor
mv.crashes.case.info.map$Lighting.Conditions
mv.crashes.case.info.map$Pedestrian.Bicyclist.Action
mv.crashes.case.info.map$Weather.Conditions
mv.crashes.case.info.map$Road.Surface.Conditions
mv.crashes.case.info.map$Number.of.Vehicles.Involved
0
200
400
600
800
MeanDecreaseGini

```
#Machine Learning on Individual and Vehicle Information using Random Forest

# Load Individual and Vehicle Data
mv.crashes.individual.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-cra
shes-and-insurance-reduction/motor-vehicle-crashes-individual-information-three-year-window.csv"
)
mv.crashes.vehicle.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashe
s-and-insurance-reduction/motor-vehicle-crashes-vehicle-information-three-year-window.csv")


# Merge into one dataset
mv.crashes.individual.vehicle.info <- merge(mv.crashes.individual.info,mv.crashes.vehicle.info,b
y.x = c("Case.Vehicle.ID"),by.y =c("Case.Vehicle.ID"))

# Response variable is Injury Severity and flagging fatal and non-fatal crashes
Ref.Injury.Severity <- unique(mv.crashes.individual.vehicle.info$Injury.Severity)
Ref.Injury.Severity <-data.frame(Ref.Injury.Severity.id=as.character(seq_len(length(Ref.Injury.S
everity))),Injury.Severity=Ref.Injury.Severity)
Ref.Injury.Severity <-cbind(Ref.Injury.Severity,Fatal.Ind=ifelse(as.numeric(Ref.Injury.Severity
$Ref.Injury.Severity.id=="6")==1,"Y","N"))

# Limiting catagory variable with less than 53 values as RF can not handle more than 53
Ref.Contributing.Factor.1.Description <- data.frame(table(mv.crashes.individual.vehicle.info$Con
tributing.Factor.1.Description))
Ref.Contributing.Factor.1.Description <- Ref.Contributing.Factor.1.Description[order(Ref.Contrib
uting.Factor.1.Description$Freq,decreasing = TRUE),]
rownames(Ref.Contributing.Factor.1.Description) <- NULL
Ref.Contributing.Factor.1.Description$Id <-rownames(Ref.Contributing.Factor.1.Description)
Ref.Contributing.Factor.1.Description <- Ref.Contributing.Factor.1.Description[which(Ref.Contrib
uting.Factor.1.Description$Id<53),]
# Limiting catagory variable with less than 53 values as RF can not handle more than 53
Ref.Contributing.Factor.2.Description <- data.frame(table(mv.crashes.individual.vehicle.info$Con
tributing.Factor.2.Description))
Ref.Contributing.Factor.2.Description <- Ref.Contributing.Factor.2.Description[order(Ref.Contrib
uting.Factor.2.Description$Freq,decreasing = TRUE),]
rownames(Ref.Contributing.Factor.2.Description) <- NULL
Ref.Contributing.Factor.2.Description$Id <-rownames(Ref.Contributing.Factor.2.Description)
Ref.Contributing.Factor.2.Description <- Ref.Contributing.Factor.2.Description[which(Ref.Contrib
uting.Factor.2.Description$Id<53),]


# Link Fatal/nonFatal flag to the main dataset
mv.crashes.individual.vehicle.info <- merge(mv.crashes.individual.vehicle.info,Ref.Injury.Severi
ty,by.x = c("Injury.Severity"),by.y = c("Injury.Severity"))

# join on top 53 contributing factors in order avoid more categorical variable limitation in RF
mv.crashes.individual.vehicle.info <- merge(mv.crashes.individual.vehicle.info,Ref.Contributing.
Factor.1.Description,by.x = c("Contributing.Factor.1.Description"),by.y = c("Var1"))
mv.crashes.individual.vehicle.info <- merge(mv.crashes.individual.vehicle.info,Ref.Contributing.
Factor.2.Description,by.x = c("Contributing.Factor.2.Description"),by.y = c("Var1"))
```

```
# Get the columns required for modelling

mv.crashes.individual.vehicle.info.map <-mv.crashes.individual.vehicle.info[,c(which(colnames(m
v.crashes.individual.vehicle.info)=="Case.Individual.ID"),which(colnames(mv.crashes.individual.v
ehicle.info)=="Victim.Status") , which(colnames(mv.crashes.individual.vehicle.info)=="Ejection")
, which(colnames(mv.crashes.individual.vehicle.info)=="Sex") , which(colnames(mv.crashes.individ
ual.vehicle.info)=="Injury.Descriptor") , which(colnames(mv.crashes.individual.vehicle.info)=="I
njury.Location") , which(colnames(mv.crashes.individual.vehicle.info)=="Age") , which(colnames(m
v.crashes.individual.vehicle.info)=="Action.Prior.to.Accident") , which(colnames(mv.crashes.indi
vidual.vehicle.info)=="Number.of.Occupants") , which(colnames(mv.crashes.individual.vehicle.inf
o)=="Engine.Cylinders") , which(colnames(mv.crashes.individual.vehicle.info)=="Contributing.Fact
or.1.Description") , which(colnames(mv.crashes.individual.vehicle.info)=="Contributing.Factor.2.
Description") , which(colnames(mv.crashes.individual.vehicle.info)=="Event.Type"),
which(colnames(mv.crashes.individual.vehicle.info)=="Fatal.Ind"))]

# since the dataset is very large train the model with 1500 fatal case and 2000  non fatal cases
mv.crashes.individual.vehicle.info.map.f <- mv.crashes.individual.vehicle.info.map[which(mv.cras
hes.individual.vehicle.info.map$Fatal.Ind=="Y"),]

mv.crashes.individual.vehicle.info.map.f.train <- mv.crashes.individual.vehicle.info.map.f[1:150
0,]
mv.crashes.individual.vehicle.info.map.f.test <- mv.crashes.individual.vehicle.info.map.f[1501:2
000,]

mv.crashes.individual.vehicle.info.map.nf <- mv.crashes.individual.vehicle.info.map[which(mv.cra
shes.individual.vehicle.info.map$Fatal.Ind=="N"),]

mv.crashes.individual.vehicle.info.map.nf.train <- mv.crashes.individual.vehicle.info.map.nf[1:2
000,]
mv.crashes.individual.vehicle.info.map.nf.test <- mv.crashes.individual.vehicle.info.map.nf[2001
:2500,]

# create training dataset
mv.crashes.individual.vehicle.info.map.m <- rbind(mv.crashes.individual.vehicle.info.map.f.trai
n,mv.crashes.individual.vehicle.info.map.nf.train)
mv.crashes.individual.vehicle.info.map.m.test <- rbind(mv.crashes.individual.vehicle.info.map.f.
test,mv.crashes.individual.vehicle.info.map.nf.test)


# convert all the characters to factors
character_vars <- lapply(mv.crashes.individual.vehicle.info.map.m, class) == "character"
mv.crashes.individual.vehicle.info.map.m[, character_vars] <- lapply(mv.crashes.individual.vehic
le.info.map.m[, character_vars], as.factor)
mv.crashes.individual.vehicle.info.map.m.test[, character_vars] <- lapply(mv.crashes.individual.
vehicle.info.map.m.test[, character_vars], as.factor)

# Exclude all NAs from the model dataset
mv.crashes.individual.vehicle.info.map.m <- na.omit(mv.crashes.individual.vehicle.info.map.m)
mv.crashes.individual.vehicle.info.map.m.test <- na.omit(mv.crashes.individual.vehicle.info.map.
m.test)

#traing the model using random forest
model.mv.civ <- randomForest(mv.crashes.individual.vehicle.info.map.m$Fatal.Ind ~mv.crashes.indi
vidual.vehicle.info.map.m$Ejection + mv.crashes.individual.vehicle.info.map.m$Sex +mv.crashes.in
```

```
dividual.vehicle.info.map.m$Injury.Descriptor + mv.crashes.individual.vehicle.info.map.m$Injury.
Location +mv.crashes.individual.vehicle.info.map.m$Action.Prior.to.Accident +mv.crashes.individu
al.vehicle.info.map.m$Number.of.Occupants +mv.crashes.individual.vehicle.info.map.m$Engine.Cylin
ders  +mv.crashes.individual.vehicle.info.map.m$Event.Type+mv.crashes.individual.vehicle.info.ma
p.m$Age  ,mv.crashes.individual.vehicle.info.map.m,ntree=500)


# Summary of the Random Forest results
summary(model.mv.civ)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted       1981   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes           3962   matrix numeric
## oob.times       1981   -none- numeric
## classes            2   -none- character
## importance         9   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               1981   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
# Confusion Matrix
model.mv.civ$confusion
```

```
##      N   Y class.error
## N 1229  68  0.05242868
## Y   40 644  0.05847953
```

```
# importance matrix
importance(model.mv.civ)
```

```
##                                                                MeanDecreaseGini
## mv.crashes.individual.vehicle.info.map.m$Ejection                      24.025558
## mv.crashes.individual.vehicle.info.map.m$Sex                            4.758861
## mv.crashes.individual.vehicle.info.map.m$Injury.Descriptor            382.630517
## mv.crashes.individual.vehicle.info.map.m$Injury.Location              274.446095
## mv.crashes.individual.vehicle.info.map.m$Action.Prior.to.Accident      52.607191
## mv.crashes.individual.vehicle.info.map.m$Number.of.Occupants           12.255979
## mv.crashes.individual.vehicle.info.map.m$Engine.Cylinders              13.983774
## mv.crashes.individual.vehicle.info.map.m$Event.Type                    43.719433
## mv.crashes.individual.vehicle.info.map.m$Age                           81.770241
```

```
# Number of decision trees used by the Random Forest
model.mv.civ$ntree
```

```
## [1] 500
```

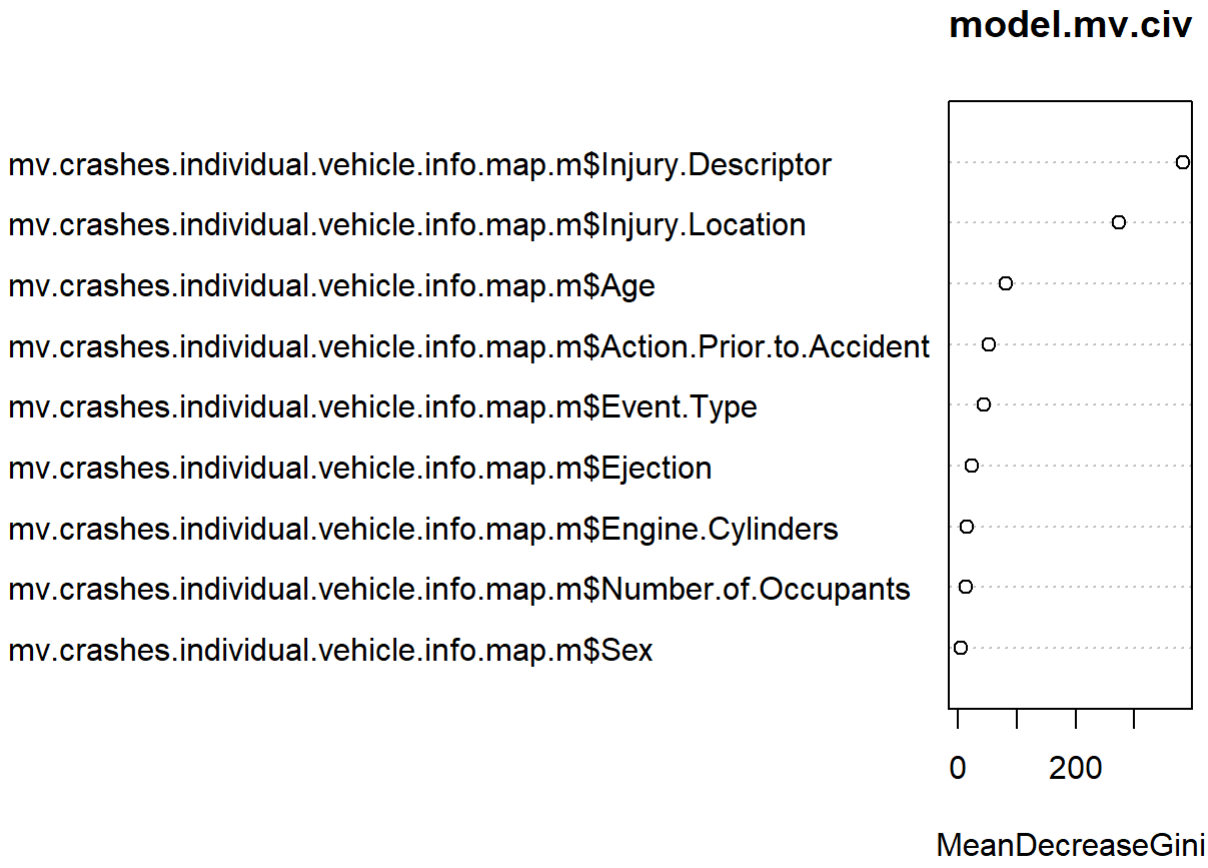
```
# Type of algorithm used by the Random Forest
model.mv.civ$type
```

```
## [1] "classification"
```

```
# Class Error
model.mv.civ$confusion[,'class.error']
```

```
##          N          Y
## 0.05242868 0.05847953
```

```
# Variable importance plot
varImpPlot(model.mv.civ)
```




```
# Predict the test dataset using the model created
rf.pred<-predict(model.mv.civ,mv.crashes.individual.vehicle.info.map.m[,-c(1,2,11,12,14)])
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 3.5.3
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(Matrix)
```

```
## Warning: package 'Matrix' was built under R version 3.5.3
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
library(lattice)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

```
##
## Attaching package: 'caret'
```

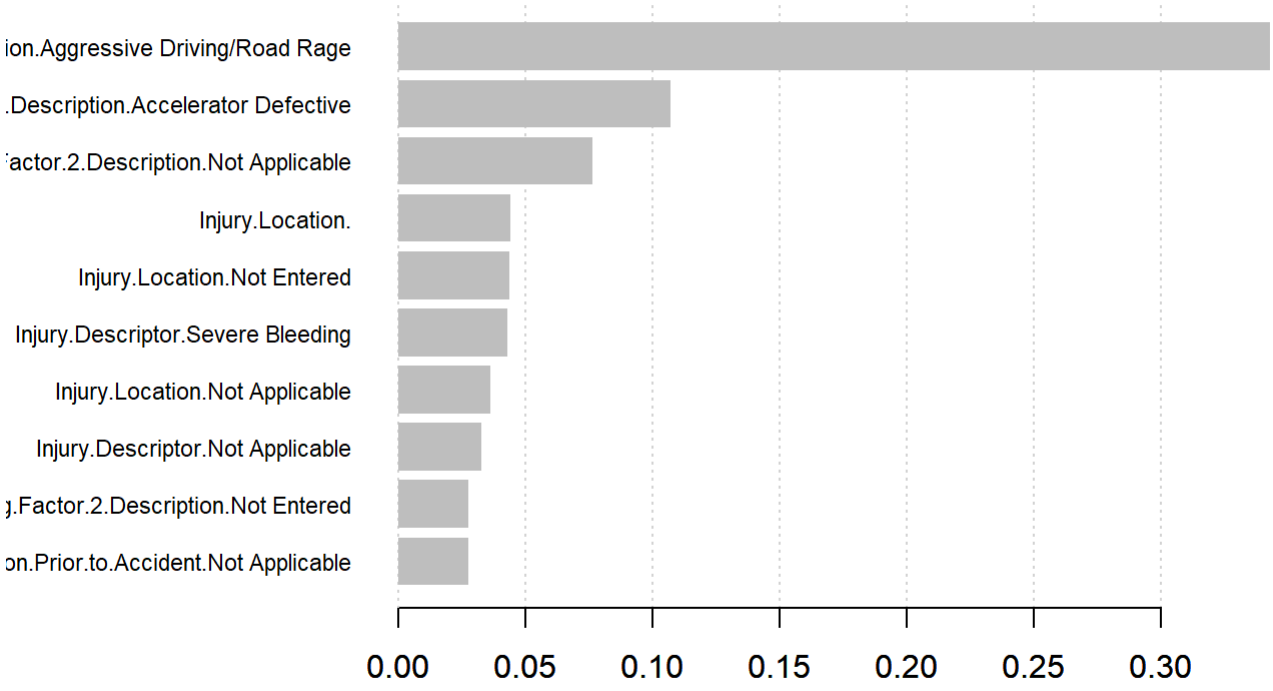
```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
mv.crashes.individual.vehicle.info.map.m <- rbind(mv.crashes.individual.vehicle.info.map.f.trai
n,mv.crashes.individual.vehicle.info.map.nf.train,mv.crashes.individual.vehicle.info.map.f.test,
mv.crashes.individual.vehicle.info.map.nf.test)

mv.crashes.individual.vehicle.info.map.train <- rbind(mv.crashes.individual.vehicle.info.map.f.t
rain,mv.crashes.individual.vehicle.info.map.nf.train)
mv.crashes.individual.vehicle.info.map.test <- rbind(mv.crashes.individual.vehicle.info.map.f.te
st,mv.crashes.individual.vehicle.info.map.nf.test)

sparse_matrix <- sparse.model.matrix(Fatal.Ind ~ .-1, data = mv.crashes.individual.vehicle.info.
map.m)

mv.crashes.individual.vehicle.info.map.test <- na.omit(mv.crashes.individual.vehicle.info.map.te
st)
output_vector = mv.crashes.individual.vehicle.info.map.train[,"Fatal.Ind"] == "Y"
output_vector.test = mv.crashes.individual.vehicle.info.map.test[,"Fatal.Ind"] == "Y"

ohe <- c('Ejection','Sex' , 'Injury.Descriptor' ,'Injury.Location' ,'Action.Prior.to.Accident' ,
'Number.of.Occupants' ,'Engine.Cylinders' ,'Event.Type' , 'Contributing.Factor.1.Description' ,
'Contributing.Factor.2.Description' )

dummies <- dummyVars(~ Ejection +  Sex + Injury.Descriptor +Injury.Location +Action.Prior.to.Acc
ident +Number.of.Occupants +Engine.Cylinders +Event.Type + Contributing.Factor.1.Description + C
ontributing.Factor.2.Description  , data = mv.crashes.individual.vehicle.info.map.m)

mv.crashes.individual.vehicle.info.map.m.ohe <- as.data.frame(predict(dummies, newdata = mv.cras
hes.individual.vehicle.info.map.m))

mv.crashes.individual.vehicle.info.map.m.ohe.mix <- cbind(mv.crashes.individual.vehicle.info.ma
p.m[,-c(which(colnames(mv.crashes.individual.vehicle.info.map.m) %in% ohe))],mv.crashes.individu
al.vehicle.info.map.m.ohe)

#mv.crashes.individual.vehicle.info.map.m.ohe.mix <-  mv.crashes.individual.vehicle.info.map.m.o
he.mix[]

mv.train = mv.crashes.individual.vehicle.info.map.m.ohe.mix[mv.crashes.individual.vehicle.info.m
ap.m.ohe.mix$Case.Individual.ID %in% mv.crashes.individual.vehicle.info.map.train$Case.Individua
l.ID,]

mv.test = mv.crashes.individual.vehicle.info.map.m.ohe.mix[mv.crashes.individual.vehicle.info.ma
p.m.ohe.mix$Case.Individual.ID %in% mv.crashes.individual.vehicle.info.map.test$Case.Individual.
ID,]



#labels = mv.train['Fatal.Ind'] == "Y"

xgb <- xgboost(data = data.matrix(mv.train[,-c(1,2,4)]),
               label = output_vector,
  eta = 0.1,
 max_depth = 15,
 nround=25,
 subsample = 0.5,
```

```
 colsample_bytree = 0.5,
 seed = 1,
 eval_metric = "merror",
 objective = "multi:softprob",
 num_class = 12,
 nthread = 3
)
```

```
## [1]  train-merror:0.022000
## [2]  train-merror:0.007714
## [3]  train-merror:0.005143
## [4]  train-merror:0.005429
## [5]  train-merror:0.006286
## [6]  train-merror:0.005143
## [7]  train-merror:0.003714
## [8]  train-merror:0.003714
## [9]  train-merror:0.003143
## [10] train-merror:0.002571
## [11] train-merror:0.002571
## [12] train-merror:0.002571
## [13] train-merror:0.002571
## [14] train-merror:0.002571
## [15] train-merror:0.002571
## [16] train-merror:0.002571
## [17] train-merror:0.002286
## [18] train-merror:0.002286
## [19] train-merror:0.002286
## [20] train-merror:0.002286
## [21] train-merror:0.002286
## [22] train-merror:0.002000
## [23] train-merror:0.002000
## [24] train-merror:0.001714
## [25] train-merror:0.001714
```

```
y_pred <- predict(xgb, data.matrix(mv.test[,-c(1,2,4)]))
model <- xgb.dump(xgb, with_stats = T)
names <- dimnames(data.matrix(mv.train[,-c(1,2,4)]))[[2]]

importance_matrix <- xgb.importance(names, model = xgb)
xgb.plot.importance(importance_matrix[1:10,])
```




```
test <- chisq.test(mv.train$Age, output_vector)
```

```
## Warning in chisq.test(mv.train$Age, output_vector): Chi-squared
## approximation may be incorrect
```

```
print(test)
```

```
##
##  Pearson's Chi-squared test
##
## data:  mv.train$Age and output_vector
## X-squared = 502.88, df = 95, p-value < 2.2e-16
```

```
bst.cv <- xgb.cv( data=data.matrix(mv.train[,-c(1,2,4)]), label=output_vector,
                      nfold=10, nrounds=25, prediction=TRUE, verbose=FALSE)

bst.cv.test <- xgb.cv( data=data.matrix(mv.test[,-c(1,2,4)]), label=output_vector.test,
                      nfold=10, nrounds=25, prediction=TRUE, verbose=FALSE)

pred.cv = matrix(bst.cv$pred,nrow =length(bst.cv$pred) , ncol = 2)
pred.cv[ pred.cv[,1] < 0.5 ,2] <- -1
pred.cv[ pred.cv[,1] >= 0.5 ,2] <- 1
pred.cv = max.col(pred.cv,"last")
confusionMatrix(factor(output_vector+1),factor(pred.cv))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2
##          1 1998    2
##          2   56 1444
##
##                Accuracy : 0.9834
##                  95% CI : (0.9786, 0.9874)
##     No Information Rate : 0.5869
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.966
##  Mcnemar's Test P-Value : 3.421e-12
##
##             Sensitivity : 0.9727
##             Specificity : 0.9986
##          Pos Pred Value : 0.9990
##          Neg Pred Value : 0.9627
##              Prevalence : 0.5869
##          Detection Rate : 0.5709
##    Detection Prevalence : 0.5714
##       Balanced Accuracy : 0.9857
##
##        'Positive' Class : 1
##
```

```
pred.cv.test = matrix(bst.cv.test$pred,nrow =length(bst.cv.test$pred) , ncol = 2)
pred.cv.test[ pred.cv.test[,1] < 0.5 ,2] <- -1
pred.cv.test[ pred.cv.test[,1] >= 0.5 ,2] <- 1
pred.cv.test = max.col(pred.cv.test,"last")
confusionMatrix(factor(output_vector.test+1),factor(pred.cv.test))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2
##          1 316   0
##          2   0 162
##
##                Accuracy : 1
##                  95% CI : (0.9923, 1)
##     No Information Rate : 0.6611
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6611
##          Detection Rate : 0.6611
##    Detection Prevalence : 0.6611
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 1
##
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
## The following object is masked from 'package:plyr':
##
##     ozone
```

```
library(ggmap)
library("data.table")
```

```
## Warning: package 'data.table' was built under R version 3.5.3
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

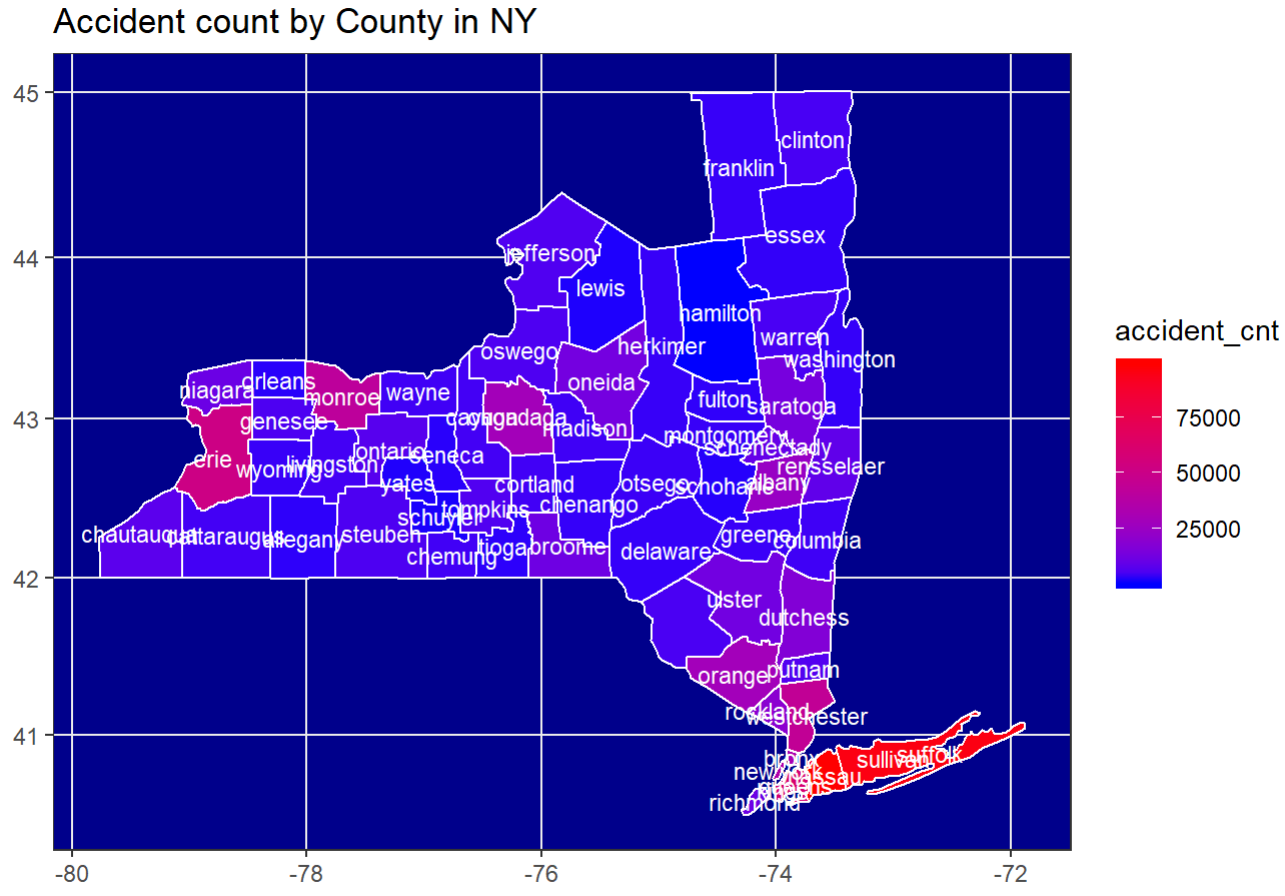
```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year
```

```
## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

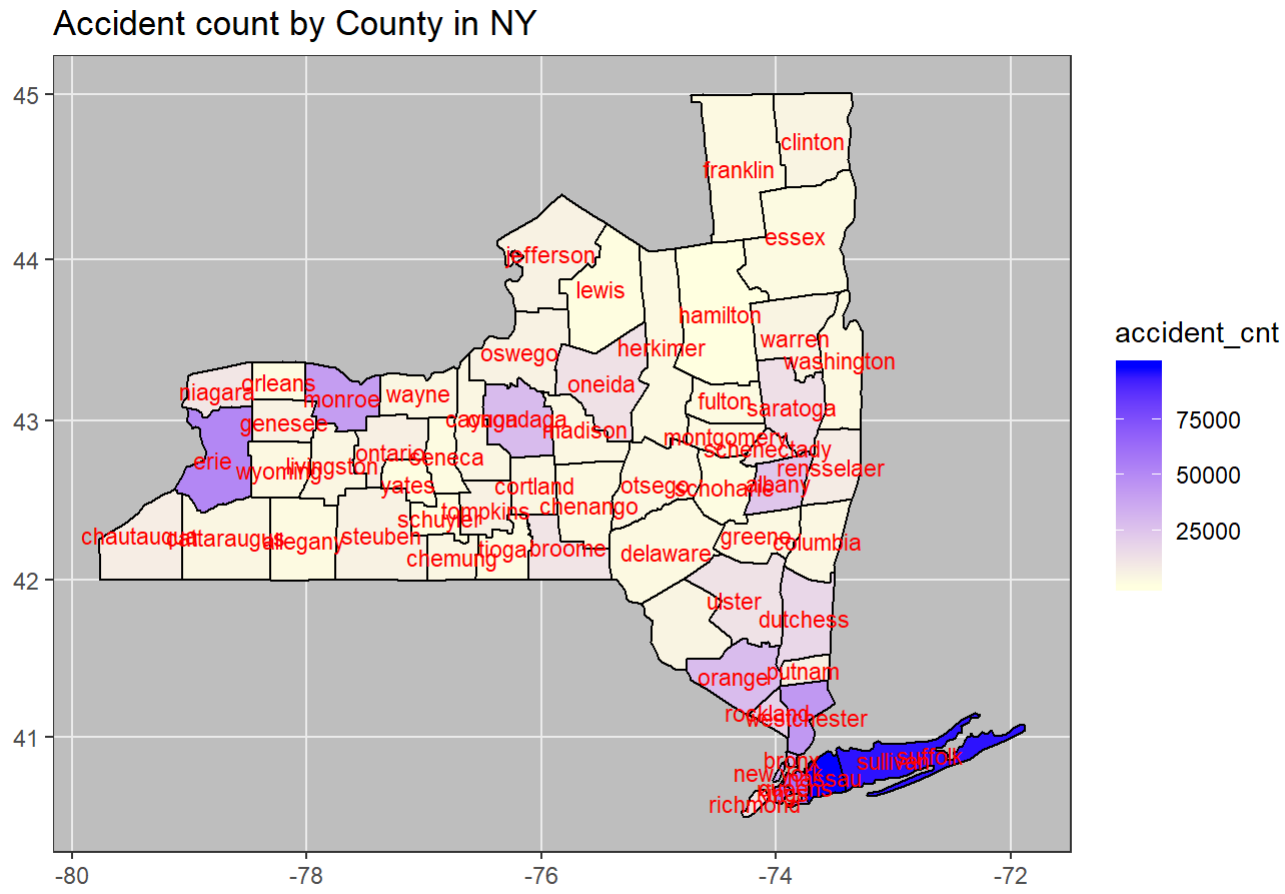
```
library(knitr)
library(dplyr)

# merger crash data with zip code and county data to get the lat long
map.county <- map_data('county')
crashes.map <- data.frame(table(mv.crashes.case.info$County.Name))
crashes.map$county_names <-tolower(crashes.map$Var1)
crashes.map$state_names <- "new york"
crashes.map$accident_cnt <- crashes.map$Freq
crashes.map <- crashes.map[,3:5]
map.county <- data.table(map_data('county'))
setkey(map.county,region,subregion)
crashes.map <- data.table(crashes.map)
setkey(crashes.map,state_names,county_names)
map.df      <- map.county[crashes.map]

# process county name to fit into the map
cnames1 <- aggregate(cbind(long, lat) ~ subregion, data=map.df,
                     FUN=function(x)mean(range(x)))
cnames2 <- map.df %>% group_by(subregion) %>%
    summarize_at(vars(long, lat), ~ mean(range(.)))

all.equal(cnames1, as.data.frame(cnames2))
```

```
##  [1] "Attributes: < Names: 2 string mismatches >"
##  [2] "Attributes: < Length mismatch: comparison on first 2 components >"
##  [3] "Attributes: < Component 1: Modes: character, externalptr >"
##  [4] "Attributes: < Component 1: target is character, current is externalptr >"
##  [5] "Attributes: < Component 2: Modes: numeric, character >"
##  [6] "Attributes: < Component 2: Lengths: 61, 1 >"
##  [7] "Attributes: < Component 2: target is numeric, current is character >"
##  [8] "Component \"subregion\": Lengths (61, 63) differ (string compare on first 61)"
##  [9] "Component \"subregion\": 12 string mismatches"
## [10] "Component \"long\": Numeric: lengths (61, 63) differ"
## [11] "Component \"lat\": Numeric: lengths (61, 63) differ"
```

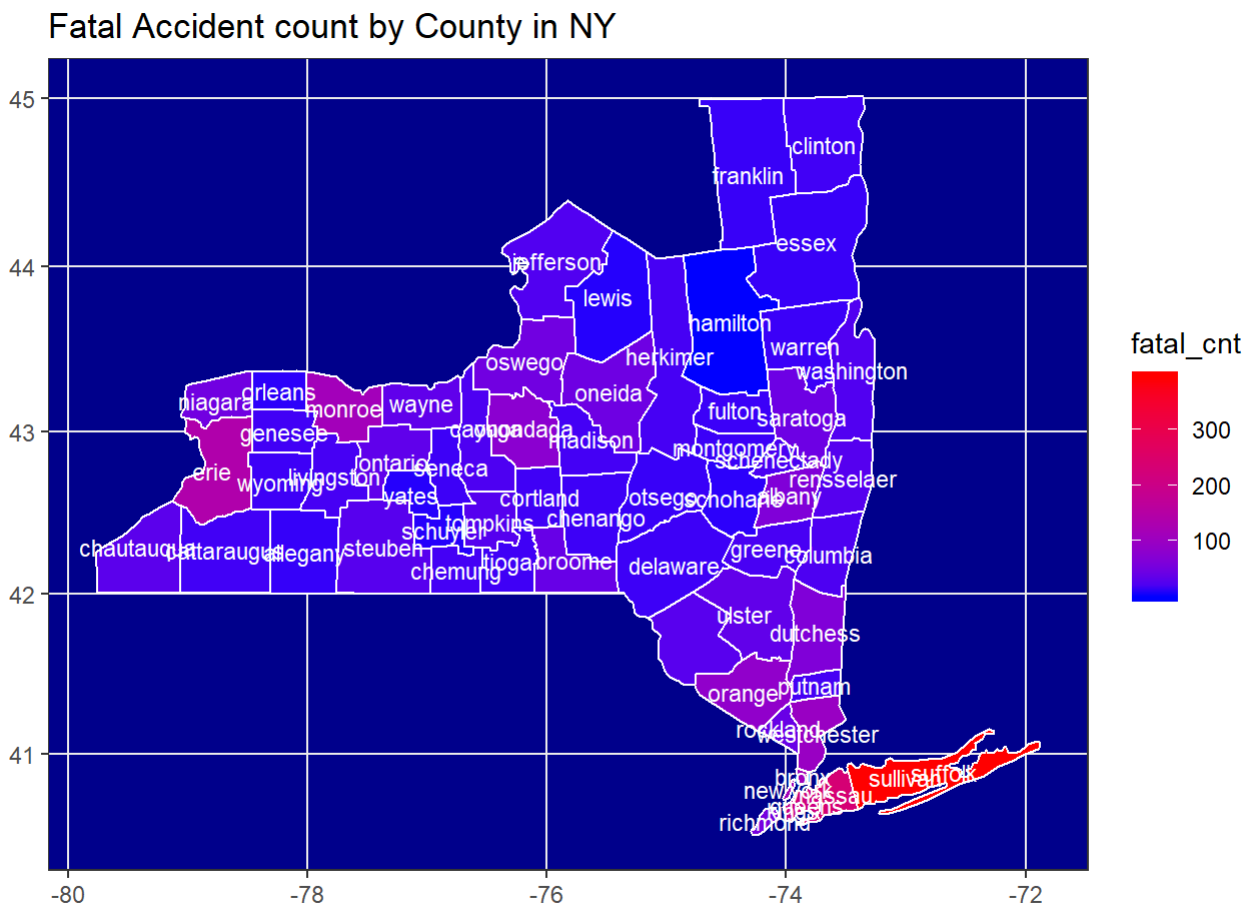
```
cnames <- aggregate(cbind(long, lat) ~ subregion, data=map.df,
                    FUN=function(x)mean(range(x)))

cnames[52, 2:3] <- c(-73, 40.855)  #adjust the long and lat of poorly centered names
cnames$angle <- rep(0, nrow(cnames)) #create an angle column
cnames[22, 4] <- -90    #adjust the angle of atypically shaped

# now plot the accident count on geomaps
ggplot(map.df, aes(x=long, y=lat)) + geom_polygon(aes(group=group,  fill=accident_cnt),color="wh
ite")+coord_map() + ggtitle("Accident count by County in NY") + labs(x = "", y = "",color ="Acci
dent Count") +scale_fill_continuous(low = "blue", high = "red") +geom_text(data=cnames, aes( lon
g, lat, label = subregion), size=3,color="white") +  theme_bw() +
  theme(panel.background = element_rect(fill = "blue4"))
```



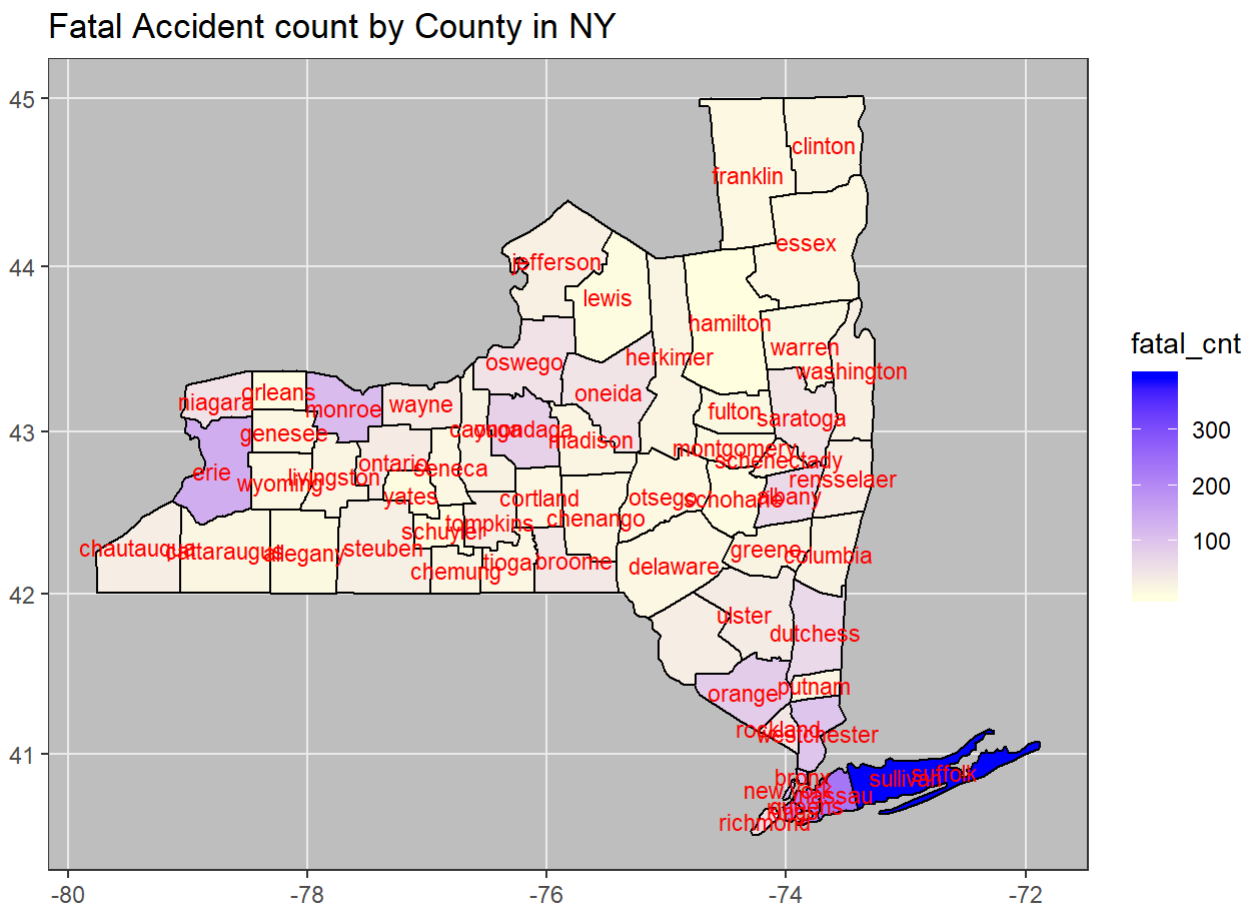
```
ggplot(map.df, aes(x=long, y=lat), color="black") + geom_polygon(aes(group=group,  fill=accident_cnt),color="black")+coord_map() + ggtitle("Accident count by County in NY") + labs(x = "", y = "",color ="Accident Count") +scale_fill_continuous(low = "lightyellow", high = "blue") +geom_text(data=cnames, aes( long, lat, label = subregion), size=3,color="red") +  theme_bw() + theme(panel.background = element_rect(fill = "gray"))
```



```
# mv.crashes.case.info <- read.csv("~/01 Personal/MS/IST 687/Project 2/nys-motor-vehicle-crashes
-and-insurance-reduction/motor-vehicle-crashes-case-information-three-year-window.csv",stringsAs
Factors = FALSE)
map.county <- map_data('county')


# merger crash data with zip code and county data to get the lat long
crashes.map <- data.frame(table(mv.crashes.case.info[which(mv.crashes.case.info$Crash.Descriptor
=="Fatal Accident"),10]))
crashes.map$county_names <-tolower(crashes.map$Var1)
crashes.map$state_names <- "new york"
crashes.map$fatal_cnt <- crashes.map$Freq
crashes.map <- crashes.map[,3:5]
map.county <- data.table(map.county)
setkey(map.county,region,subregion)
crashes.map <- data.table(crashes.map)
setkey(crashes.map,state_names,county_names)
map.df      <- map.county[crashes.map]

# Process the county name to fit into the map
cnames1 <- aggregate(cbind(long, lat) ~ subregion, data=map.df,
                     FUN=function(x)mean(range(x)))
cnames2 <- map.df %>% group_by(subregion) %>%
    summarize_at(vars(long, lat), ~ mean(range(.)))

#all.equal(cnames1, as.data.frame(cnames2))

cnames <- aggregate(cbind(long, lat) ~ subregion, data=map.df,
                    FUN=function(x)mean(range(x)))

cnames[52, 2:3] <- c(-73, 40.855)  #adjust the long and lat of poorly centered names
cnames$angle <- rep(0, nrow(cnames)) #create an angle column
cnames[22, 4] <- -90    #adjust the angle of atypically shaped


#Now plot the data in geomap
ggplot(map.df, aes(x=long, y=lat)) + geom_polygon(aes(group=group,  fill=fatal_cnt),color="whit
e")+coord_map() + ggtitle("Fatal Accident count by County in NY") + labs(x = "", y = "",color =
"Fatal Count") +scale_fill_continuous(low = "blue", high = "red") +geom_text(data=cnames, aes( l
ong, lat, label = subregion), size=3,color="white") +  theme_bw() +  theme(panel.background = el
ement_rect(fill = "blue4"))
```




```
ggplot(map.df, aes(x=long, y=lat), color="black") + geom_polygon(aes(group=group,  fill=fatal_cnt),color="black")+coord_map() + ggtitle("Fatal Accident count by County in NY") + labs(x = "", y = "",color ="Fatal Count") +scale_fill_continuous(low = "lightyellow", high = "blue") +geom_text(data=cnames, aes( long, lat, label = subregion), size=3,color="red") +  theme_bw() + theme(panel.background = element_rect(fill = "gray"))
```


Fatal Accident count by County in NY
45
44
43
42
41
-80
-78
-76
-74
-72
clinton
franklin
essex
jefferson
lewis
hamilton
warren
washington
herkimer
oswego
oneida
fulton
saratoga
niagara
orleans
monroe
wayne
genesee
madison
montgomery
schenectady
rensselaer
erie
wyoming
livingston
ontario
seneca
yates
cortland
otsego
schoharie
albany
tompkins
chenango
schuyler
chautauqua
cattaraugus
allegany
steuben
chemung
tioga
broome
delaware
greene
columbia
ulster
dutchess
orange
putnam
rockland
westchester
bronx
sullivan
suffolk
new york
nassau
richmond
fatal_cnt
300
200
100