TidyOsha

Zhenru Han

3/13/2017

Introduction:

First of all, my purpose of this project is to prepare the data for analyze that will be used to report on the most dangerous places to work in Massachusetts. I use the data provided by professor and originally pulled from the website of Occupational Safety and Health Administration to help me reach the goal.

My goal is finding the most dangerous place to work in MA, and I get the brief description from the introduction text of these data as the following:

Osha.dbf - main table with company name, address, date of inspection, etc. If you get the entire country, there's a number after the word OSHA, since it's too big to put on one CD. Viol.dbf - violations from each inspection. If you get the entire country, there's a number after the word VIOL, since it's too big to put on one CD. Accid.dbf - details about accident victims Hazsub.dbf - hazardous substances involved Debt.dbf - status of debt History.dbf - outlines a history of any changes in penalty Admpay.dbf - a record of collecting administrative fees or penalties Prog.dbf - special programs the inspection might be involved in Relact.dbf - whether the inspection is related to another inspection or other action Optinfo.dbf - optional information

Here I discovered that only some of the datasets are useful for reaching my goal.

I only need to consider data Osha, Accid and Hazsub. I need to put these data together and clean and tidy them.

Therefore, I separated my project to two steps. The first step is cleaning and tidying the datasets I selected. I need to create a new dataset that combine all useful information from all there datasets (select useful columns and join them together). The Second part is plotting my overall perspective to this data and how I possibly use the data to find the most dangerous place to work in MA. This is the most important part in my project.

Step 1

First, clean data "Accident"

```
###################### Clean Data "Accident" ########################

library(foreign)
library(lubridate)
```

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

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

```r
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(data.table)
```

```
## ------------------------------------------------------------------------------
## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!
## ------------------------------------------------------------------------------
##
## Attaching package: 'data.table'
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year
```

```r
library(scales)
library(ggplot2)
library(base)

# Read data
accid <- read.dbf("ACCID.dbf")
nrow(accid)
```

```
## [1] 2147
```

```r
ncol(accid)
```

```
## [1] 16
```

```r
acc_1 <- read.dbf("lookups/acc.dbf")
occ_1 <- read.dbf("lookups/occ.dbf")
hzs_1 <- read.dbf("lookups/hzs.dbf")

###############################################

#First, check if there's any NA columns in accident
indi = rep(0, ncol(accid))
for(i in 1:ncol(accid)){indi[i] = sum(!is.na(accid[,i]))}
indi #those columns that retrun 0 are null columns and could be removed, none of them can be removed
```

```
##  [1] 2147 2147  839 2147 1421 2147 2147 2147 2147 2147 2147 2147 2147  442
## [15] 2147 2147
```

```r
#Then remove the duplicate rows
b <- colnames(accid[1:ncol(accid)])

a <- data.table(accid, key= b)
accid <- subset(accid,!duplicated(a))

#Second, check if the column "SITESTATE" all "MA"

if(sum(accid$SITESTATE=="MA") == dim(accid)[1]){accid %<>% select(-SITESTATE)}
dim(acc_1)
```

```
## [1] 153   3
```

```r
#Third, change all numbers in columns "NATURE","BODYPART","ENVIRON","SOURCE","EVENT" and "HUMAN" into n
# 1 "BODYPART"
sum(acc_1$CATEGORY=="PART-BODY")
```

```
## [1] 31
```

```r
parts_1 <- acc_1[(acc_1$CATEGORY== "PART-BODY"),]
dim(parts_1)
```

```
## [1] 31  3
```

```r
parts_1 <- select(parts_1, CODE, VALUE)
head(parts_1)
```

```
##   CODE      VALUE
## 1   01    ABDOMEN
## 2   02   ARM-MULT
## 3   03       BACK
## 4   04 BODYSYSTEM
## 5   05      CHEST
## 6   06     EAR(S)
```

```r
colnames(parts_1) <- c("BODYPART", "BODYPART_1")
str(parts_1)
```

```
## 'data.frame':    31 obs. of  2 variables:
##  $ BODYPART  : Factor w/ 48 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ BODYPART_1: Factor w/ 149 levels "ABDOMEN","ABSORPTION",..: 1 7 9 15 28 40 41 45 46 49 ...
##  - attr(*, "data_types")= chr  "C" "C" "C"
```

```
tidyaccid_1 <- left_join(accid, parts_1, by="BODYPART")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_1$BODYPART = NULL

#2 ENIVIROMENT
sum(acc_1$CATEGORY=="ENVIR-FAC")
```

```
## [1] 18
```

```
parts_2 <- acc_1[(acc_1$CATEGORY == "ENVIR-FAC"),]
dim(parts_2)
```

```
## [1] 18  3
```

```
parts_2 <- select(parts_2, CODE, VALUE)
head(parts_2)
```

```
##    CODE                          VALUE
## 32   01              PINCH POINT ACTION
## 33   02    CATCH POINT/PUNCTURE ACTION
## 34   03              SHEAR POINT ACTION
## 35   04           SQUEEZE POINT ACTION
## 36   05           FLYING OBJECT ACTION
## 37   06 OVERHEAD MOVING/FALLING OBJ AC
```

```
colnames(parts_2) <- c("ENVIRON", "ENVIRON_1")
str(parts_2)
```

```
## 'data.frame':    18 obs. of  2 variables:
##  $ ENVIRON  : Factor w/ 48 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ENVIRON_1: Factor w/ 149 levels "ABDOMEN","ABSORPTION",..: 117 24 129 133 53 112 61 97 27 52 ...
##  - attr(*, "data_types")= chr  "C" "C" "C"
```

```
tidyaccid_2 <- left_join(tidyaccid_1, parts_2, by="ENVIRON")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_2$ENVIRON = NULL

#3 EVENT
sum(acc_1$CATEGORY=="EVENT-TYP")
```

```
## [1] 14
```

```
parts_3 <- acc_1[(acc_1$CATEGORY == "EVENT-TYP"),]
dim(parts_3)
```

```
## [1] 14  3
```

```
parts_3 <- select(parts_3, CODE, VALUE)
head(parts_3)
```

```
##    CODE                VALUE
## 50   01           STRUCK BY
## 51   02 CAUGHT IN OR BETWEEN
## 52   03   BITE/STING/SCRATCH
```

```
## 53   04     FALL(SAME LEVEL)
## 54   05 FALL(FROM ELEVATION)
## 55   06       STRUCK AGAINST
```

```
colnames(parts_3) <- c("EVENT", "EVENT_1")
str(parts_3)
```

```
## 'data.frame':    14 obs. of  2 variables:
##  $ EVENT  : Factor w/ 48 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ EVENT_1: Factor w/ 149 levels "ABDOMEN","ABSORPTION",..: 136 25 11 48 47 135 127 76 75 2 ...
##  - attr(*, "data_types")= chr  "C" "C" "C"
```

```
tidyaccid_3 <- left_join(tidyaccid_2, parts_3, by="EVENT")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_3$EVENT = NULL

#4 HUMAN
sum(acc_1$CATEGORY=="HUMAN-FAC")
```

```
## [1] 20
```

```
parts_4 <- acc_1[(acc_1$CATEGORY == "HUMAN-FAC"),]
dim(parts_4)
```

```
## [1] 20  3
```

```
parts_4 <- select(parts_4, CODE, VALUE)
head(parts_4)
```

```
##    CODE                          VALUE
## 64   01    MISJUDGMENT, HAZ. SITUATION
## 65   02 NO PERSONAL PROTECTIVE EQ USED
## 66   03 NO APPROPR PROTECTIVE CLOTHING
## 67   04 MALFUNC IN SECURING/WARNING OP
## 68   05  DISTRACTING ACTIONS BY OTHERS
## 69   06 EQUIP. INAPPROPR FOR OPERATION
```

```
colnames(parts_4) <- c("HUMAN", "HUMAN_1")
str(parts_4)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ HUMAN  : Factor w/ 48 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ HUMAN_1: Factor w/ 149 levels "ABDOMEN","ABSORPTION",..: 99 108 107 93 37 44 94 114 128 119 ...
##  - attr(*, "data_types")= chr  "C" "C" "C"
```

```
tidyaccid_4 <- left_join(tidyaccid_3, parts_4, by="HUMAN")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_4$HUMAN = NULL

#5 NATURE
sum(acc_1$CATEGORY=="NATUR-INJ")
```

```
## [1] 22
```

```
parts_5 <- acc_1[(acc_1$CATEGORY == "NATUR-INJ"),]
dim(parts_5)
```

```
## [1] 22  3
```

```
parts_5 <- select(parts_5, CODE, VALUE)
head(parts_5)
```

```
##    CODE               VALUE
## 84   01           AMPUTATION
## 85   02             ASPHYXIA
## 86   03 BRUISE/CONTUS/ABRAS
## 87   04       BURN(CHEMICAL)
## 88   05    BURN/SCALD(HEAT)
## 89   06           CONCUSSION
```

```
colnames(parts_5) <- c("NATURE", "NATURE_1")
str(parts_5)
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ NATURE  : Factor w/ 48 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ NATURE_1: Factor w/ 149 levels "ABDOMEN","ABSORPTION",..: 5 8 18 20 21 31 32 34 36 43 ...
##  - attr(*, "data_types")= chr  "C" "C" "C"
```

```
tidyaccid_5 <- left_join(tidyaccid_4, parts_5, by="NATURE")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_5$NATURE = NULL

#6 SOURCE
sum(acc_1$CATEGORY=="SOURC-INJ")
```

```
## [1] 48
```

```
parts_6 <- acc_1[(acc_1$CATEGORY == "SOURC-INJ"),]
dim(parts_6)
```

```
## [1] 48  3
```

```
parts_6 <- select(parts_6, CODE, VALUE)
head(parts_6)
```

```
##     CODE               VALUE
## 106   01             AIRCRAFT
## 107   02         AIR PRESSURE
## 108   03 ANIMAL/INS/REPT/ETC.
## 109   04                BOAT
## 110   05        BODILY MOTION
## 111   06  BOILER/PRESS VESSEL
```

```
colnames(parts_6) <- c("SOURCE", "SOURCE_1")
str(parts_6)
```

```
## 'data.frame':    48 obs. of  2 variables:
##  $ SOURCE  : Factor w/ 48 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ SOURCE_1: Factor w/ 149 levels "ABDOMEN","ABSORPTION",..: 4 3 6 13 14 16 17 19 26 29 ...
##  - attr(*, "data_types")= chr  "C" "C" "C"
```

```
tidyaccid_6 <- left_join(tidyaccid_5, parts_6, by="SOURCE")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_6$SOURCE = NULL

#Fourth, change numbers in column "OCC_CODE" in to names according to data frame occ_1

parts_7 <- occ_1[(occ_1$CODE),]
dim(parts_7)
```

```
## [1] 503   2
```

```
colnames(parts_7) <- c("OCC_CODE", "OCCUPATION")
str(parts_7)
```

```
## 'data.frame':    503 obs. of  2 variables:
##  $ OCC_CODE  : Factor w/ 503 levels "003","004","005",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ OCCUPATION: Factor w/ 503 levels "ACCOUNTANTS AND AUDITORS",..: 227 71 6 8 145 321 367 248 7 249
##  - attr(*, "data_types")= chr  "C" "C"
```

```
tidyaccid_7 <- left_join(tidyaccid_6, parts_7, by="OCC_CODE")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyaccid_7$OCC_CODE = NULL

#Fifth, change numbers in column "HAZSUB" in to names according to data frame hzs_1

parts_8 <- hzs_1[(hzs_1$CODE),]
dim(parts_8)
```

```
## [1] 1777    2
```

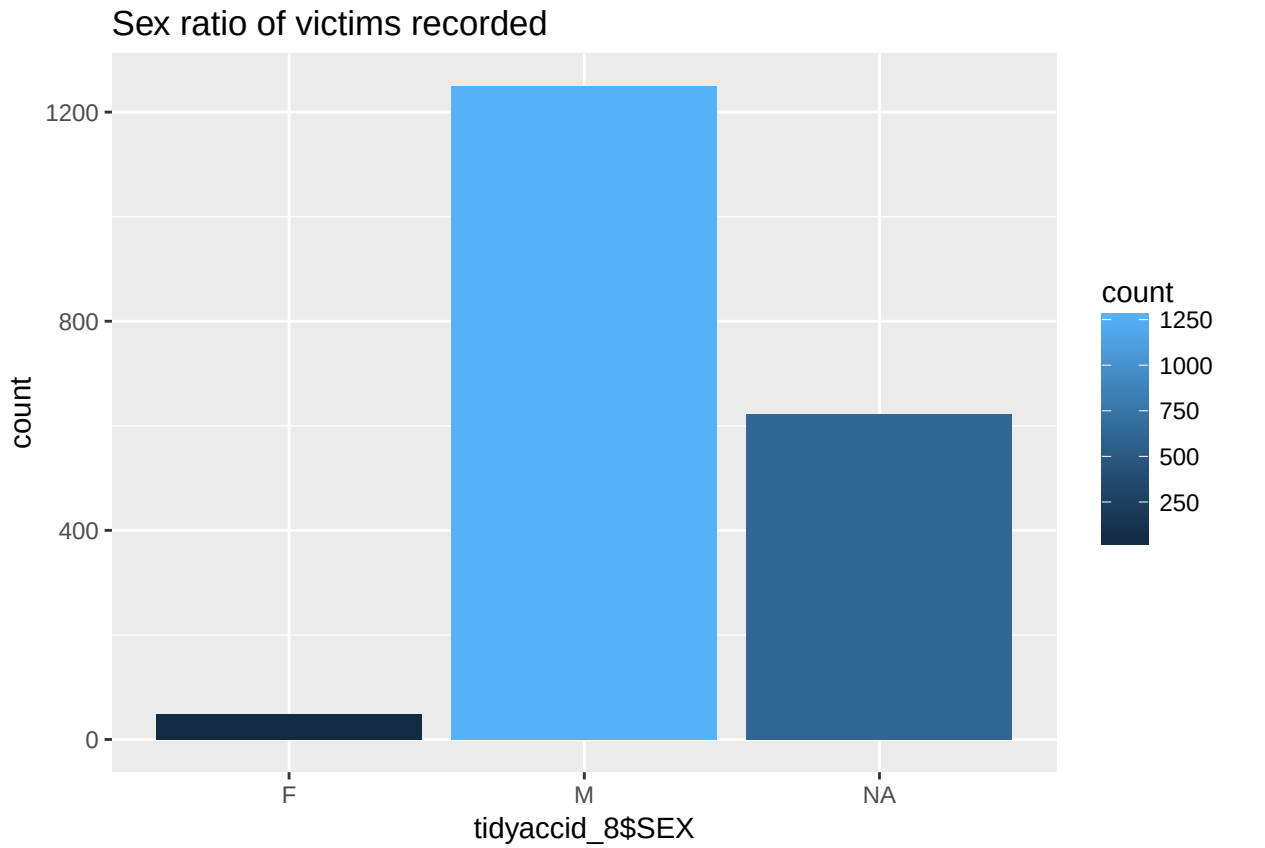
```
colnames(parts_8) <- c("HAZSUB", "HAZSUB_1")
str(parts_8)
```

```
## 'data.frame':    1777 obs. of  2 variables:
##  $ HAZSUB  : Factor w/ 1777 levels "0005","0010",..: 1617 1618 1619 1620 1621 1622 1623 1624 1625 161
##  $ HAZSUB_1: Factor w/ 1771 levels "(DICHLOROMETHYL) BENZENE",..: 1543 1529 1518 1531 1519 1504 1550
##  - attr(*, "data_types")= chr  "C" "C"
```

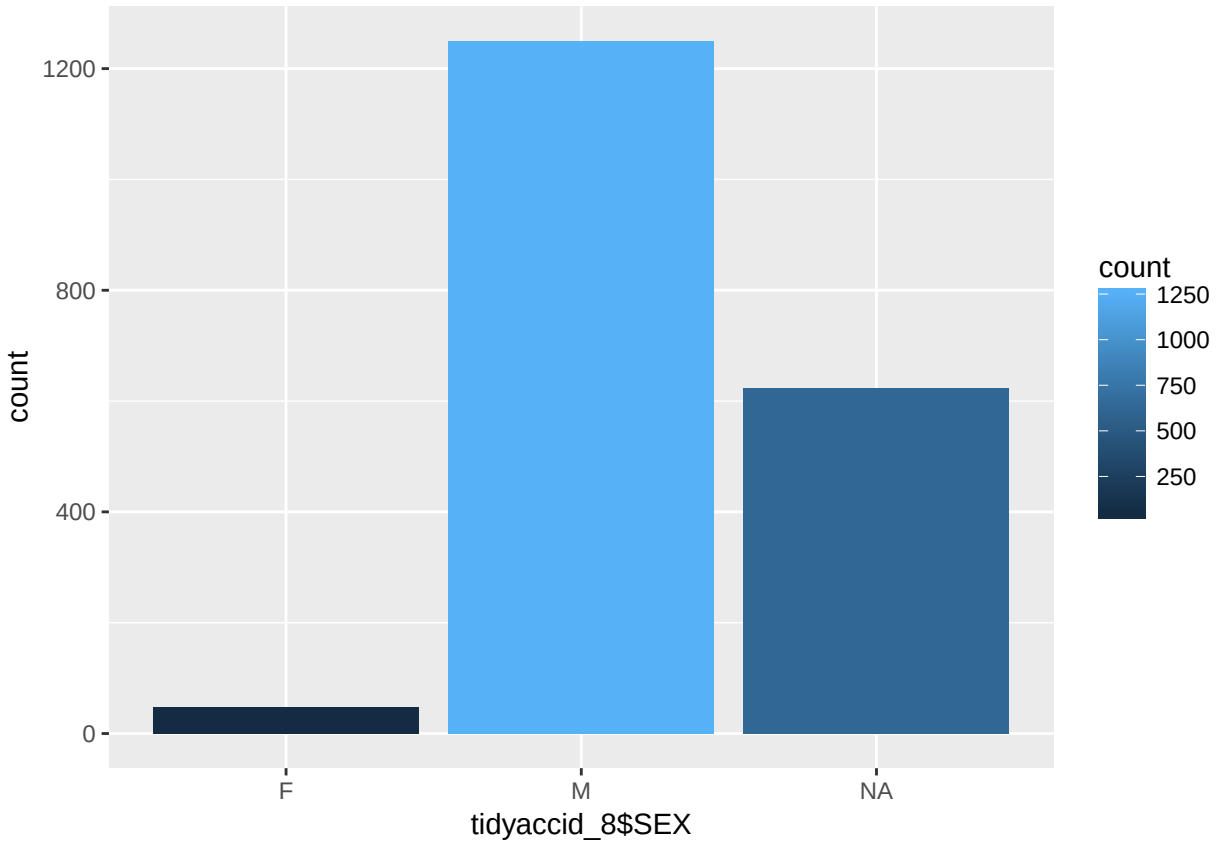
```
tidyaccid_8 <- left_join(tidyaccid_7, parts_8, by="HAZSUB")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

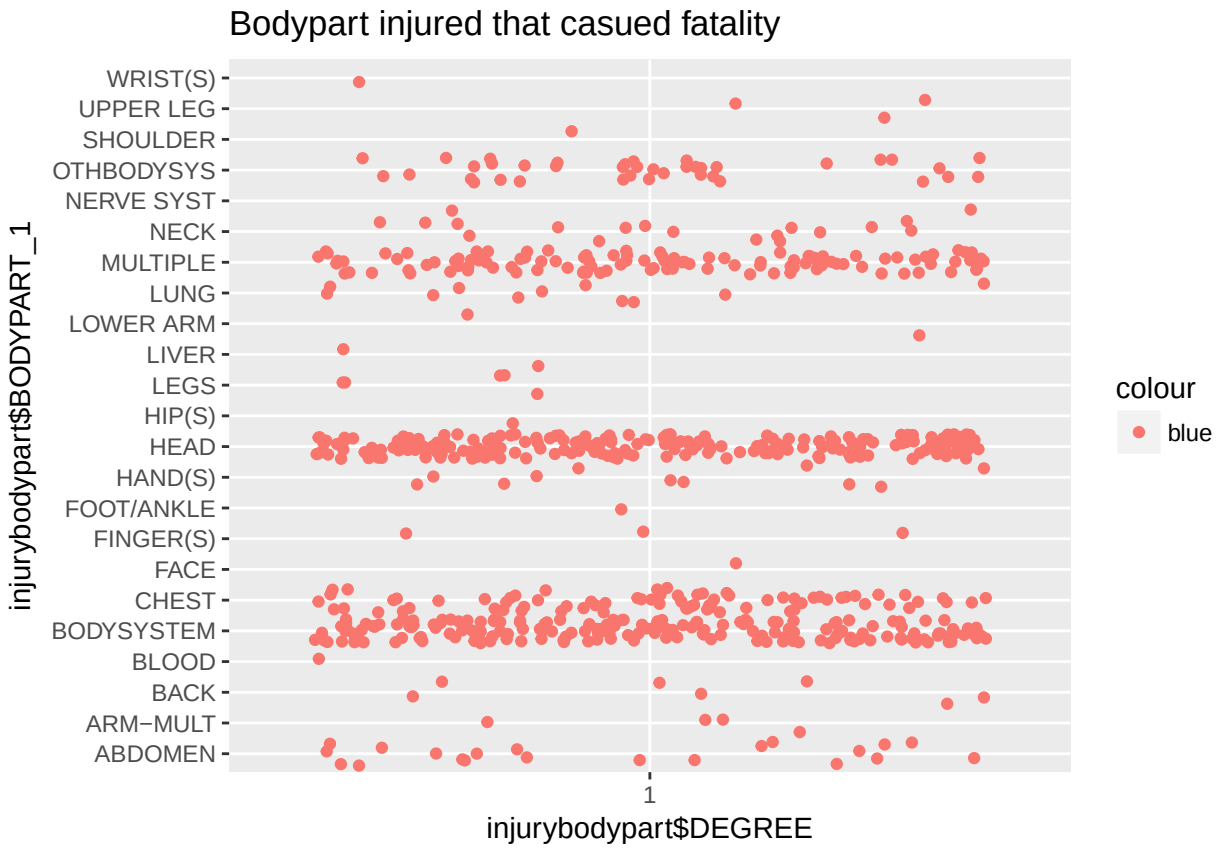
```
tidyaccid_8$HAZSUB = NULL

######################################################################
#Above we get the first tidy data of accidents happened in MA
#We could roughly take a look at this data, do some analysis and see which columns are not useful for f

#First, we could take a look at the sex ratio of victims
sexplot <- ggplot(tidyaccid_8, aes(x = tidyaccid_8$SEX)) + geom_bar(aes(fill = ..count..))
sexplot + ggtitle("Sex ratio of victims recorded")
```


Sex ratio of victims recorded
1200
800
400
0
count
F
M
NA
tidyaccid_8$SEX
count
1250
1000
750
500
250

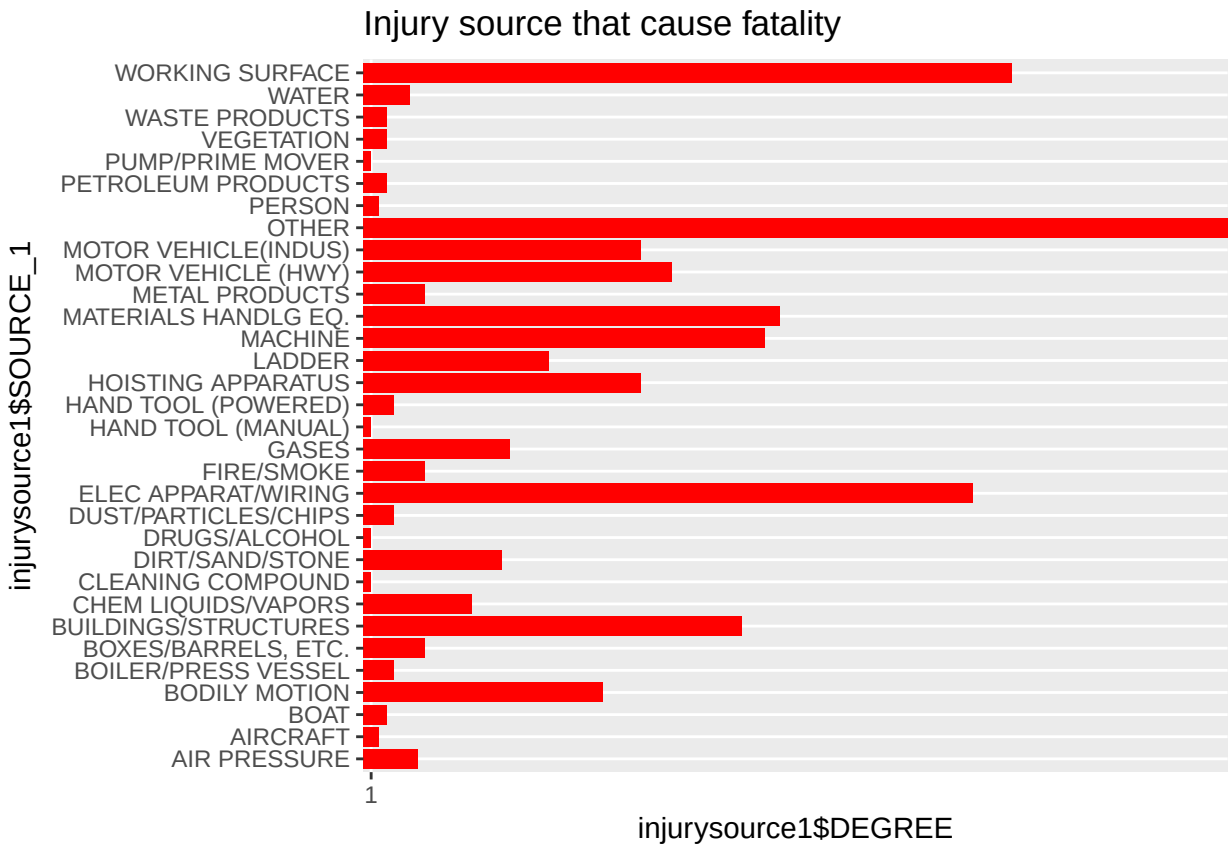
```
sexplot
```



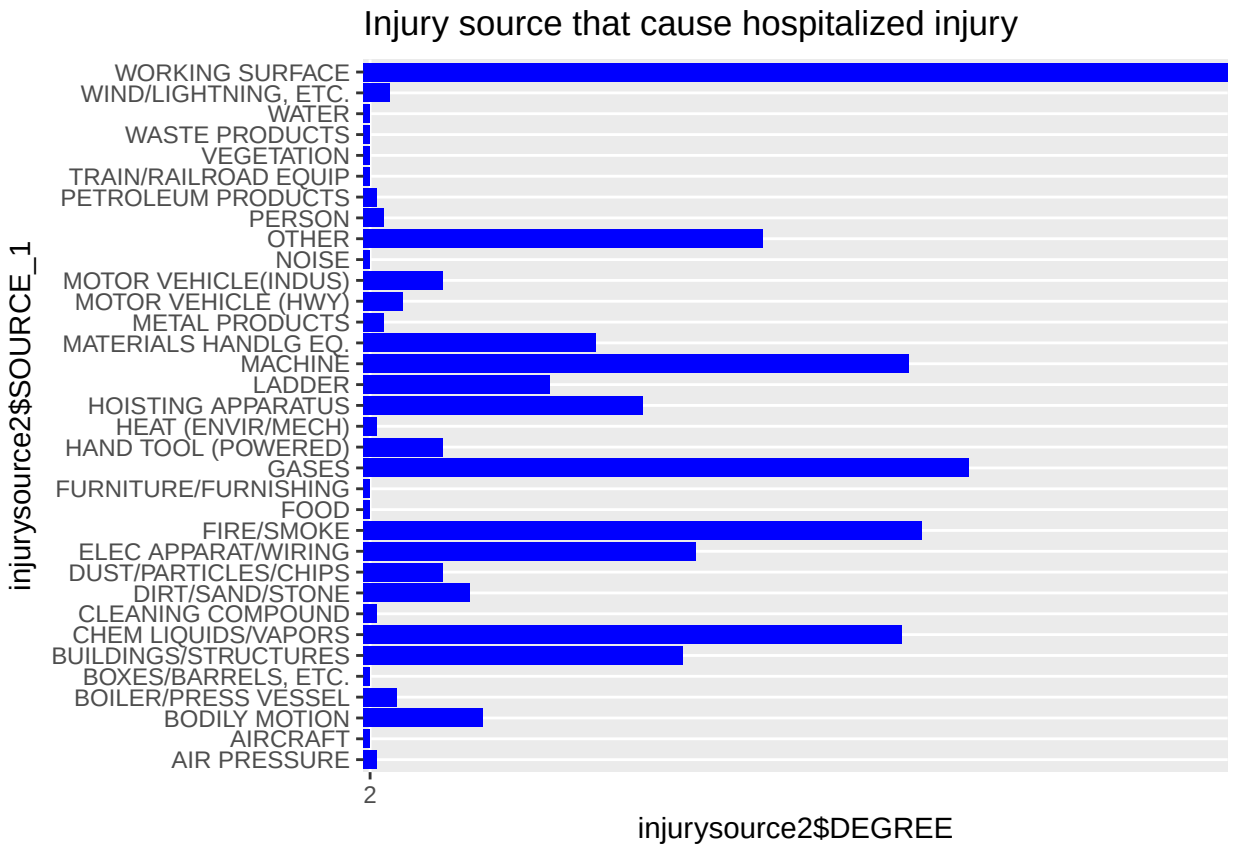

```
#Clearly, we see that most of the victims that have their gender being recorded are males.
#And also, the gender has no influence on finding the "dangerous place", so we can remove SEX column
tidyaccid_9 <- tidyaccid_8
tidyaccid_9$SEX = NULL
#Similarly, the names and ages of victims are also not useful, remove it
tidyaccid_9$NAME = NULL
tidyaccid_9$AGE = NULL

#Second, we could take a look at the bodypart vitims get injured and which part will cause the most sev
injurybodypart <- data.frame(subset(tidyaccid_9, tidyaccid_9$DEGREE ==1))
injurybodyplot <- ggplot(injurybodypart, aes(x=injurybodypart$DEGREE, y=injurybodypart$BODYPART_1)) + gg
injurybodyplot + geom_jitter(aes(x=injurybodypart$DEGREE, y=injurybodypart$BODYPART_1, colour = "blue"))
```



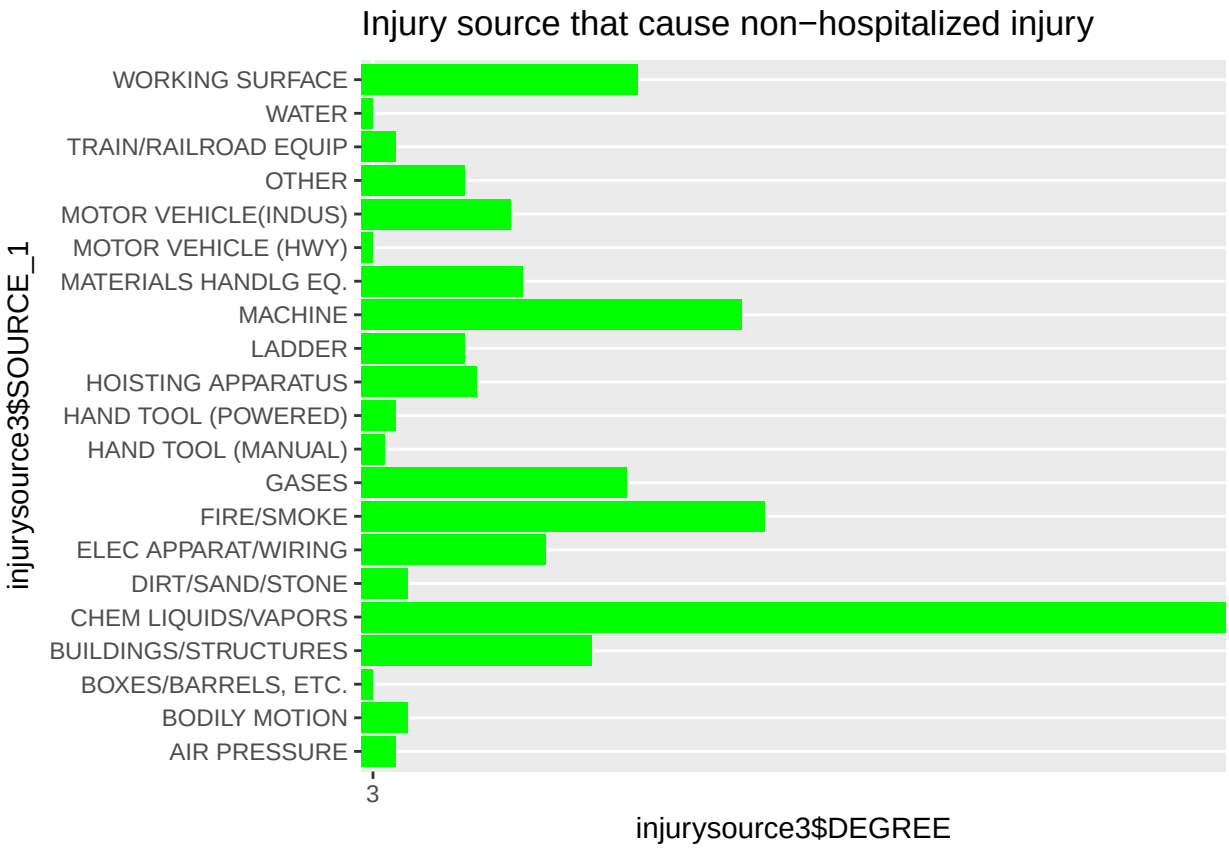

```
#We can clearly see that the victims who get injured in multiple places, head, chest and body system re

#Third, we can try to find the relationship between injury degree and the source

injurysource1 <- data.frame(subset(tidyaccid_9, tidyaccid_9$DEGREE ==1))
injurysource2 <- data.frame(subset(tidyaccid_9, tidyaccid_9$DEGREE ==2))
injurysource3 <- data.frame(subset(tidyaccid_9, tidyaccid_9$DEGREE ==3))

injurysourceplot1 <- ggplot(injurysource1,aes(x=injurysource1$SOURCE_1,y=injurysource1$DEGREE)) + coord
injurysourceplot1 +geom_bar(stat ="identity", fill = "red") + ggtitle("Injury source that cause fatalit
```




```
injurysourceplot2 <- ggplot(injurysource2,aes(x=injurysource2$SOURCE_1,y=injurysource2$DEGREE)) + coord
injurysourceplot2 +geom_bar(stat ="identity", fill = "blue") + ggtitle("Injury source that cause hospit
```




```
injurysourceplot3 <- ggplot(injurysource3,aes(x=injurysource3$SOURCE_1,y=injurysource3$DEGREE)) + coord_
injurysourceplot3 +geom_bar(stat ="identity", fill = "green") + ggtitle("Injury source that cause non-h
```




```
#We can clearly see the difference between these three plots.

#Also I find out that there are some rows that have "0" degree of injury and those observations are use

tidyaccid <- data.frame(subset(tidyaccid_9, tidyaccid_9$DEGREE !=0))
#This tidyaccid is the final cleaned accid data I get.
```

Second, we take a look at data “Hazsub”. We discovered that in this data, the useful column “HAZSUB1” is parially the same as the hazsub column in data table “tidyaccid” we just cleaned. However, this data “hazsub1” is larger, which indicate more activites or places that are hazadous and not yet cause any injury.

```
######################## Clean Data "Hazsub" ##########################

library(foreign)
library(lubridate)
library(tidyr)
library(dplyr)
library(magrittr)
library(data.table)
library(scales)

# Read data

hazsub <- read.dbf("hazsub.dbf")


#remove duplicate rows
bh <- colnames(hazsub[1:ncol(hazsub)])
```

```
ah <- data.table(hazsub, key= bh)
hazsub <- subset(hazsub,!duplicated(ah))

# ACCORDING to the OSHA graph I uploaded to github, we only need to keep the columns ACRIVITYNO and HAZ

tidyhazsub_1 <- data.frame(hazsub$ACTIVITYNO, hazsub$HAZSUB1)
colnames(tidyhazsub_1) <- c("ACTIVITYNO","HAZSUB1")

#Change the code in column HAZSUB1 into names according to hzs_1

parts_hzs <- hzs_1[(hzs_1$CODE),]
dim(parts_hzs)
```

```
## [1] 1777    2
```

```
colnames(parts_hzs) <- c("HAZSUB1", "HAZSUB_1")
str(parts_hzs)
```

```
## 'data.frame':    1777 obs. of  2 variables:
##  $ HAZSUB1 : Factor w/ 1777 levels "0005","0010",..: 1617 1618 1619 1620 1621 1622 1623 1624 1625 16
##  $ HAZSUB_1: Factor w/ 1771 levels "(DICHLOROMETHYL) BENZENE",..: 1543 1529 1518 1531 1519 1504 1550
##  - attr(*, "data_types")= chr  "C" "C"
```

```
tidyhazsub_2 <- left_join(tidyhazsub_1, parts_hzs, by="HAZSUB1")
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
tidyhazsub_2$HAZSUB1 = NULL

#We can get the final cleaned tidyhazsub data now
tidyhazsub <- tidyhazsub_2
```

Third, take a look at the most important dataset "Osha."

```
########## This is the part of data OSHA #################

library(foreign)
library(lubridate)
library(tidyr)
library(dplyr)
library(magrittr)
library(data.table)
library(scales)
library(ggplot2)
library(base)

#read the data frame
osha <- read.dbf("osha.dbf",as.is=FALSE)
scc_1 <- read.dbf("lookups/scc.dbf")

################################################

#First, check if there's any NA columns in osha
indi = rep(0, ncol(osha))
for(i in 1:ncol(osha)){indi[i] = sum(!is.na(osha[,i]))}
```

```
indi #those columns that retrun 0 are null columns and could be removed, they are column 4 and 9
```

```
##  [1]     2 72445 80445     0 11955 80445 80445 80445     0 80397 80445
## [12] 80445 80398 80445 19074 47631 80445 47795 80445 80445 80445 80445
## [23] 80445 80445 80445 80445 80445 80445 80445 13326 10653 77371 79235
## [34]  4548 80445 15203 31022  1276  3128   515    46   212    30 80445
## [45] 80445 80445   497     4 80445 80445 38194 80445 80445 80445 80445
## [56] 80445 80445 80445 80445 80445 80445 80445 80445 80445 80445 80445
## [67] 80445 80445 80445 80445 80445 80445 80445 80445 80445 80445 80445
## [78] 80445 80445 80445 80445 80445 80445   137   402 80420 80445 79361
## [89] 79182 38194   427  3558
```

```
#remove STFLAG and CSHO_ID
osha$STFLAG = NULL
osha$CSHO_ID = NULL

##################################################

#Second, Check the layout and remove useless columns
#According to the layout of Osha, we can pull out several useful columns and make a new data table tidy
#Choose ACTIVITYNO to help join all other datasets, choose JOBTITLE to see which job is the most danger

tidyosha_1 <- data.frame(osha$ACTIVITYNO, osha$JOBTITLE, osha$ESTABNAME, osha$SITEADD, osha$SITEZIP, osh
colnames(tidyosha_1) <- c("ACTIVITYNO","JOBTITLE","ESTABNAME","SITEADD","SITEZIP","SITECITY","SITECNTY"

#add one column that combine the code of columns SITECNTY and SITECITY
tidyosha_1$LOCATIONCODE <- do.call(paste0, tidyosha_1[c("SITESTATE","SITECNTY","SITECITY")])

#Third, we need to change the code in CITECITY and SITECNTY in to names using the dataset scc_1

parts_cty <- data.frame(scc_1$STATE, scc_1$COUNTY, scc_1$CITY, scc_1$NAME)
dim(parts_cty)
```

```
## [1] 43355     4
```

```
colnames(parts_cty) <- c("SITESTATE","SITECNTY", "SITECITY", "LOCATION")
parts_cty$LOCATIONCODE <- do.call(paste0, parts_cty[c("SITESTATE","SITECNTY","SITECITY")])
str(parts_cty)
```

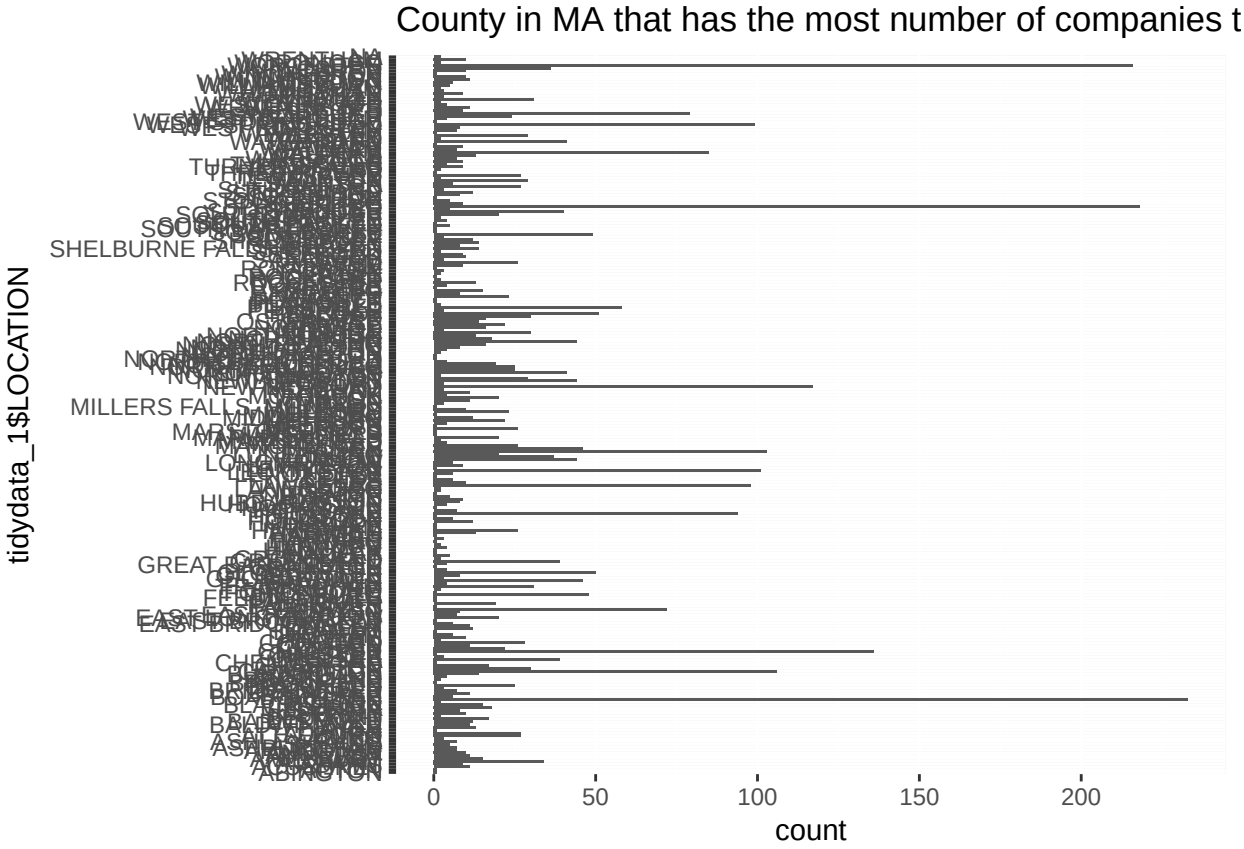
```
## 'data.frame':    43355 obs. of  5 variables:
##  $ SITESTATE   : Factor w/ 71 levels "AK","AL","AR",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SITECNTY    : Factor w/ 360 levels "000","001","002",..: 1 11 14 17 21 51 61 69 71 86 ...
##  $ SITECITY    : Factor w/ 6948 levels "0000","0001",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ LOCATION    : Factor w/ 24874 levels "000TGOMERY","AARONSBURG",..: 193 245 246 247 500 1691 2508 5
##  $ LOCATIONCODE: chr  "AK0000000" "AK0100000" "AK0130000" "AK0160000" ...
```

```
tidyosha_2 <- left_join(tidyosha_1, parts_cty, by="LOCATIONCODE")
tidyosha_2$SITECNTY.x = NULL
tidyosha_2$SITECNTY.y = NULL
tidyosha_2$SITECITY.x = NULL
tidyosha_2$SITECITY.y = NULL
tidyosha_2$SITESTATE.x = NULL
tidyosha_2$SITESTATE.y = NULL

#Join tidyhazsub into tidyosha_1 to get the places that are influenced by hazadous factors
tidyosha_3 <- right_join(tidyosha_2, tidyhazsub, by="ACTIVITYNO")
```

```
#Join tidyaccid into tidyosha_2 to see which places has caused injury under the hazadous factors.
tidyosha_4 <- left_join(tidyosha_3, tidyaccid, by="ACTIVITYNO")

#Therefore we can pick tidyosha_3 as the cleaned tidydata (might need some adjustment later)
tidydata_1 <- tidyosha_4
```

Step 2 Basic overview to the tidydata and plotting

```
#As we see, this tidydata shows the places in MA that are under the hazadous condition and some of them

#We can try to discover which county in MA that has the most number of companies that are exposed to ha

hazadouscountyplot <- ggplot(tidydata_1,aes(tidydata_1$LOCATION)) + coord_flip()
hazadouscountyplot +geom_bar() + ggtitle("County in MA that has the most number of companies that are e
```

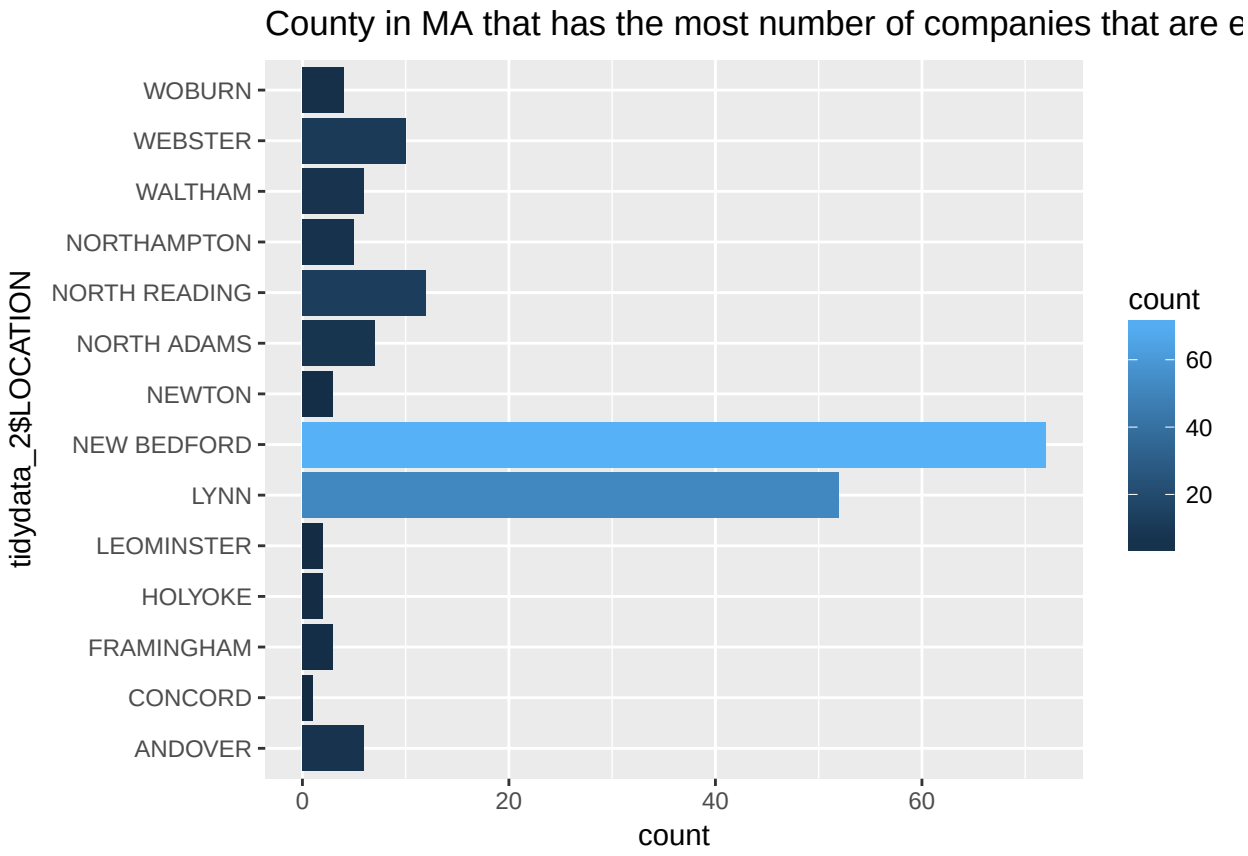



```
#Clearly this plot is way too crowdy and hard to visualize.
#We can still find out the top ten county in MA that has the most number of companies that are exposed
toptencountyhazadous <- tail(names(sort(table(tidydata_1$LOCATION))), 10)
toptencountyhazadous
```

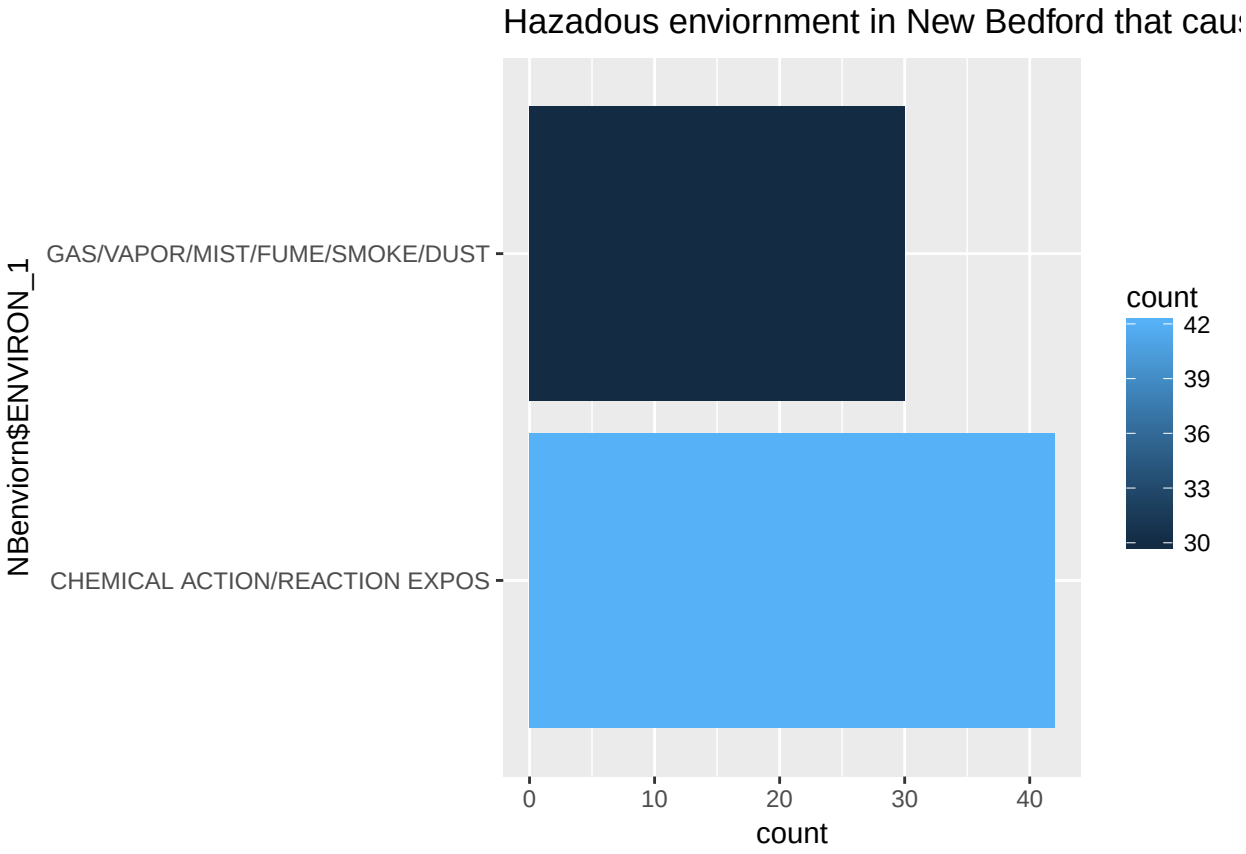
```
##  [1] "LAWRENCE"          "WEST SPRINGFIELD" "LEOMINSTER"
##  [4] "LYNN"              "CAMBRIDGE"        "NEW BEDFORD"
##  [7] "CHICOPEE"          "WORCESTER"        "SPRINGFIELD"
## [10] "BOSTON"
```

```
#However, we could think of narrowing the observations to only the county in MA that has the most numbe
```

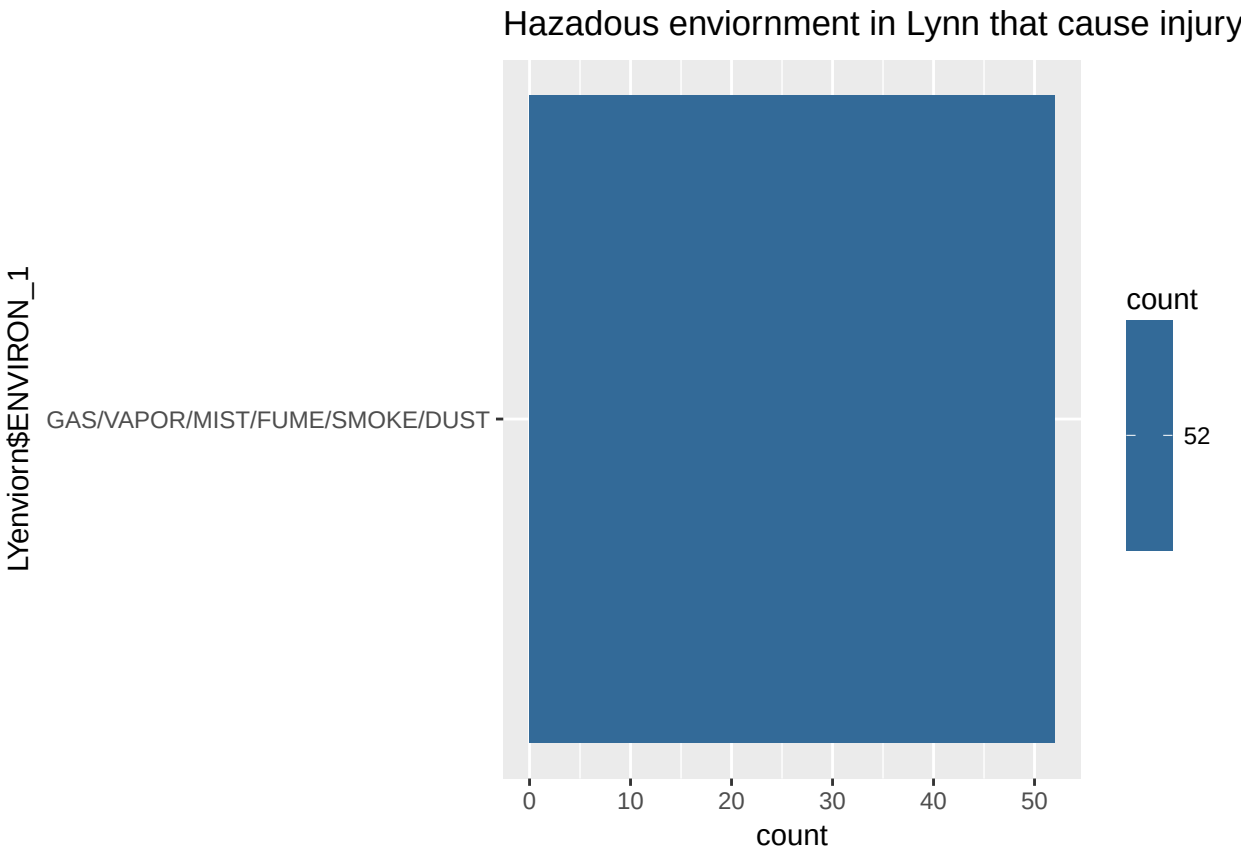
```
tidydata_2 <- filter(tidydata_1, !is.na(tidydata_1$DEGREE))

injurycountyplot <- ggplot(tidydata_2,aes(tidydata_2$LOCATION)) + coord_flip()
injurycountyplot +geom_bar(aes(fill = ..count..)) + ggtitle("County in MA that has the most number of c
```




```
#This is a much better plot. We can easily find out that NEW BEDFORD and LYNN are the two places where

#We can also take a look at what kind of enviornments are New Bedford and Lynn in:

#First, check New Bedford
#Select the subset that only shows data collected in New Bedford:
NBenviorn <- filter(tidydata_2, grepl("NEW BEDFORD", tidydata_2$LOCATION))
nbenviornplot <- ggplot(NBenviorn,aes(NBenviorn$ENVIRON_1)) + coord_flip()
nbenviornplot +geom_bar(aes(fill = ..count..)) + ggtitle("Hazadous enviornment in New Bedford that caus
```




```
#There are only two conditions : Chemical Action/Reaction Expos and GAS/Vapor/Mist/Fume/Smoke/Dust

#Second, check Lynn
#Select the subset that only shows data collected in New Bedford:
LYenviorn <- filter(tidydata_2, grepl("LYNN", tidydata_2$LOCATION))
lyenviornplot <- ggplot(LYenviorn,aes(LYenviorn$ENVIRON_1)) + coord_flip()
lyenviornplot +geom_bar(aes(fill = ..count..)) + ggtitle("Hazadous enviornment in Lynn that cause injury
```




```
#There are only one conditions :GAS/Vapor/Mist/Fume/Smoke/Dust
#And easily we see that in LYenviorn dataset, there are only the non-hospitalized injury happened; whil
#Simply compare these two places, I think that New Bedford is a more dangerous place to work.

#All my work above is not the final conclusion but only a basic view of the tidydata_1 and tidydata_2 I

#The results looks well by using tidydata_2. However, it is too narrow and lack of possibility to do mo

tidydata <- tidydata_1 #tidydata is what I get for my result.
```

Conclusion: I get the dataset "tidydata" that reaches my goal.