MA415 Midterm Project

Rachel Feingold

3/7/2017

Instructions:

Prepare the OSHA data and create a dataset for analysis.

Reading Data:

Libraries Needed:

```
require(foreign) # the require functions loads and attached the add on packages needed for the rest of
```

```
## Loading required package: foreign
```

```
require(dplyr)
```

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

```
require(tidyr)
```

```
## Loading required package: tidyr
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(lubridate)
```

```
## Loading required package: lubridate
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##     date
```

```
require(stringr)
```

```
## Loading required package: stringr
```

```
require(magrittr)
```

```
## Loading required package: magrittr

##
## Attaching package: 'magrittr'

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

```
require(data.table)
```

```
## Loading required package: data.table

## -------------------------------------------------------------------------

## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!

## -------------------------------------------------------------------------

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

## 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:dplyr':
##
##     between, first, last
```

I only read the data that was neccessary to the analysis of the most dangerous places to work in Massachusetts. The data tables I choose to include in the final dataset were:

Osha.dbf- I chose this data table because it was the main table with the companys names, addresses, inspection dates and other descriptions information about the comapnies. Accid.dbf- I chose this data table because it had details about the accidents, which is clearly related to the dangerous levels of workplaces in Massachusetts. I extracted the variables: ACTIVITYNO DEGREE, NATURE, SOURCE, EVENT, ENVIRON, HUMAN, and HAZSUB. Viol.dbf- I chose this data table because it had details on the violations of each inspection. If a company has more violations during an OSHA inspection, then a workplace may be more dangerous to work at. I extracted the variables: ACTIVITYNO, EMPHASIS, PENCURRENT, VIOLTYPE, INSTANCES, and NUMEXPOSED. History.dbf- I chose this data table because it had details on the history of the penalties imposed on the company and other violations, as well as the amount and reason for charge. If a company has many penalties, this may be related to how it complies to safety and health regualtions. I extracted the variables: ACTIVITYNO, HISTTYPE, HISTEVENT, HISTVTYPE, and FTAEVENT. Relact.dbf- I chose this data table because it had details on whether the inspection was related to another inspection or other action. If the inspection is realted to another inspection, then the ACTIVITYNO of that inspection is listed. This can give more information on whether a company is compliant with OSHA guidelines. I extracted the variables: ACTIVITYNO, RELTYPE, SAFETY, and HEALTH.

```
osha <- read.dbf("osha.dbf")
accidents <- read.dbf("accid.dbf")
violations <- read.dbf("viol.dbf")
history <- read.dbf("history.dbf")
relact <- read.dbf("relact.dbf")
```

I also needed to read in the data from lookups, in order to retrieve informations on the coding and different character levels of the various variables.

```r
acc <- read.dbf("lookups/acc.dbf")
hzs <- read.dbf("lookups/hzs.dbf")
sic <- read.dbf("lookups/sic.dbf")
scc <- read.dbf("lookups/scc.dbf")
```

Cleaning Data

First, I cleaned the OSHA table. I selected the variables (columns) I deemed most important to the analysis and I discarded the rest. Kept variables included: PREVCTTYP- type of most recent OSHA activity (accident, none, complaint, inspection, referral) ACTIVITYNO- unique identifier for each inspection record REPORTID- unique identification code, one for each federal and state office submitting compliance data ESTABNAME- name of company/establishment SITECITY- Depertment of Commerce city code establishment is in SITECNTY- Department of Commerce county code establishment is in SITEZIP- zipcode of the establishment SITEADD- address of establishment OWNERTYPE- type of owner of company (private, local government, state government, federal government) ADVNOTICE- whether an advanced notice of the inspection of given CAT_SH- safety or health codes INSPTYPE- type of inspection (unprogrammed: fatality/catashrophe, complaint, referral, monitoring, follow-up, etc; Programmed: planned, related, etc) INSPSCOPE- how indepth the inspection was (comprehensive, partial, records only, none) WHYNOINSPC- if there was no inspection, reason for such (estb. not found, denied entry, employer out of business, etc.) PASUMHOURS- sum of all inspection activity hours reported TOTPENLTY- total current penalities issued, may be a result of negotiations between the company and OSHA TOTALFTA- total failure to abate (FTA) penalties issued, may also bea result of negotiations TOTALVIOLS- total number of violations issued TOTSERIOUS- total number of serious, willful, and repeat violations issued for inspection OPENDATE- date of opening conference or entry or attempted entry into the site of the inspection CLOSEDATE- date of last closing conference held before issuing citations or exit from the inspection site.

I also turned the character variables that had coding levels, and assigned names to these levels based on the layout text files. In other instances I used the the layout file with descriptions of the variables usually references the lookups dbf's. I would then use the .dbf to help name the levels, and eventually delete the redundant columns.

```r
# Create new data set, using dplyr select(), to keep certain variables
tidyosha <- select(osha, PREVCTTYP, ACTIVITYNO, REPORTID, ESTABNAME, SITECITY, SITECNTY, SITEZIP, OWNER

# Remove old osha data table to clean environment
rm(osha)

# Assign names to the levels, based on layouts
levels(tidyosha$PREVCTTYP) <- c("Accident", "Complaint", "Inspection", "Referral")
levels(tidyosha$OWNERTYPE) <- c("Private", "Local Govt.", "State Govt.", "Federal Govt.")
levels(tidyosha$ADVNOTICE) <- c("No", "Yes")
levels(tidyosha$CAT_SH) <- c("Health", "Safety")
levels(tidyosha$INSPTYPE) <- c("Fatality/Catastrophe","Complaint", "Referral", "Monitoring","Variance",
levels(tidyosha$INSPSCOPE) <- c("Comprehensive", "Partial", "Records Only", "No Inspection")
levels(tidyosha$WHYNOINSP) <- c("Est. Not Found", "Employer Out of Bus.", "Unactive Process", "<10 Empl

# Wanted to change the format of the dates and make them more sensible to read.
tidyosha$OPENDATE <- ymd(tidyosha$OPENDATE)
tidyosha$CLOSEDATE <- ymd(tidyosha$CLOSEDATE)
```

```
## Warning: 1084 failed to parse.
```

```r
# Wanted to create a new variable, DURATION, that calculates how long a site was inspected for
duration <- interval(tidyosha$OPENDATE,tidyosha$CLOSEDATE)
```

```
duration <- as.duration(duration)
duration <- as.period(duration, days)
tidyosha <- mutate(tidyosha, INSDURATION = duration)
rm(duration)

# Want industry names for the companies. This will give a better description and background of the type
tidyosha <- left_join(tidyosha, sic, by = "SIC")
```

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

```
tidyosha <- select(tidyosha, -SIC)
rm(sic)

# Wanted to add names of the towns from the scc.dbf lookups
# First need to eliminate states other than Massachusetts
scc <- scc[(scc$STATE=="MA"),]
# Since each town/ city has two codes- CNTY and CITY, need to combine them
scc$TOWN <- with(scc, interaction(scc$CITY,scc$COUNTY))
scc <- select(scc, TOWN, NAME)
# Need to mimic the form in the tidyosha table
tidyosha$TOWN <- with(tidyosha, interaction(tidyosha$SITECITY,tidyosha$SITECNTY))
# Now that they match, we can join them
tidyosha <- left_join(tidyosha, scc, by="TOWN")
```

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

```
# Can remove old variables for city, county, linked coding
tidyosha <- select(tidyosha, -SITECITY, -SITECNTY,-TOWN)
colnames(tidyosha)[22] <- "CITY"
rm(scc)
```

Next, I cleaned the data from the accidents table. I kept the variables:

ACTIVITYNO- unique identifier for each inspection record DEGREE- extent of injury (fatal, hospoitalized, nonhispitalized) NATURE- nature of the injury (physical characteristics of the injury) SOURCE- cause/source of injury EVENT- event/action that caused the injury ENVIRON- contributing factor of the injury HUMAN- any human factory involved in the injury HAZSUB- any hazardous substances contributing to the incident

```
tidyaccid <- select(accidents, ACTIVITYNO, DEGREE, NATURE, SOURCE, EVENT, ENVIRON, HUMAN, HAZSUB)
rm(accidents)
```

Then, I recoded and renamed the character variables in a similar fashion to the OSHA data table.

```
levels(tidyaccid$DEGREE) <- c("no injury", "fatality", "hospitalized", "nonhospitalized")

natureinj <- acc[(acc$CATEGORY=="NATUR-INJ"),]
natureinj <- select(natureinj, CODE, VALUE)
colnames(natureinj) <- c("NATURE", "VALUE")
tidyaccid <- left_join(tidyaccid, natureinj, 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 <- select(tidyaccid, -NATURE)
names(tidyaccid)[8] <- "NATURE-INJ"
rm(natureinj)
```

```r
injsource <- acc[(acc$CATEGORY=="SOURC-INJ"),]
injsource <- select(injsource, CODE, VALUE)
colnames(injsource) <- c("SOURCE", "VALUE")
tidyaccid <- left_join(tidyaccid, injsource, 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
```

```r
tidyaccid <- select(tidyaccid, -SOURCE)
names(tidyaccid)[8] <- "INJ-SOURCE"
rm(injsource)

etype <- acc[(acc$CATEGORY=="EVENT-TYP"),]
etype <- select(etype, CODE, VALUE)
colnames(etype) <- c("EVENT", "VALUE")
tidyaccid <- left_join(tidyaccid, etype, 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
```

```r
tidyaccid <- select(tidyaccid, -EVENT)
names(tidyaccid)[8] <- "TYPE-EVENT"
rm(etype)

envfac <- acc[(acc$CATEGORY=="ENVIR-FAC"),]
envfac <- select(envfac, CODE, VALUE)
colnames(envfac) <- c("ENVIRON", "VALUE")
tidyaccid <- left_join(tidyaccid, envfac, 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
```

```r
tidyaccid <- select(tidyaccid, -ENVIRON)
names(tidyaccid)[8] <- "ENVIR-FAC"
rm(envfac)

humfac <- acc[(acc$CATEGORY=="HUMAN-FAC"),]
humfac <- select(humfac, CODE, VALUE)
colnames(humfac) <- c("HUMAN", "VALUE")
tidyaccid <- left_join(tidyaccid, humfac, 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
```

```r
tidyaccid <- select(tidyaccid, -HUMAN)
names(tidyaccid)[8] <- "HUMAN-FAC"
rm(humfac)

names(tidyaccid)[3] <- "CODE"        # Needed to rename the column to do a join with the lookups code db
tidyaccid <- left_join(tidyaccid, hzs, by = "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
```

```r
tidyaccid <- select(tidyaccid, -CODE)
names(tidyaccid)[8] <- "HAZ-SUB"
```

Next, I cleaned the violations table. I kept the variables:

ACTIVITYNO- unique identifier for each inspetion record EMPHASIS- whether violation was egregious (outstandingly bad) GRAVITY- level of potential harm to worker (scale of 1-10) PENCURRENT- total amount OSHA collected or expects to collect from violations VIOLTYPE- type of violation (repeat, serious, unclassified, willful, other) INSTANCES- number of instances of violation of standard related event NUMEXPOSED- number of employees exposed to hazard violates

I also changed the character variables that were encoded with codes instead of their actual names.

```
tidyviol <- select(violations, ACTIVITYNO, EMPHASIS, GRAVITY, PENCURRENT, VIOLTYPE, INSTANCES,NUMEXPOSED
rm(violations)

levels(tidyviol$EMPHASIS) <- c("Egregious")
levels(tidyviol$VIOLTYPE) <- c("Other", "Repeat", "Serious", "Unclassified", "Willful")
```

Next, I cleaned the history table. I kept the variables:

ACTIVITYNO- unique identifier for each inspection record HISTTYPE- identifies whether date relates to a penalty or FTA HISTEVENT- identifies source or cause of activity spplied against violation/penalty record (ex. state decision, review commission, amendment) HISTVTYPE- current violation type (serious, repeat, willfull, unclassified) FTAEVENT- idenitifies source or cause of activity applied against FTA (ex. appeals court, government dismissed, formal settlement)

I also changed the character variables that were encoded with codes instead of their actual names.

```
tidyhist <- select(history, ACTIVITYNO, HISTTYPE, HISTEVENT, HISTVTYPE, FTAEVENT)
rm(history)

levels(tidyhist$HISTTYPE) <- c("Failure-To-Abate", "Penalty")
levels(tidyhist$HISTEVENT) <- c("Appeals Court", "Supreme Court", "Amendment", "Govt Dismissed", "Formal
levels(tidyhist$HISTVTYPE) <- c("Other", "Repeat", "Serious", "Unclassfied", "Willful")
levels(tidyhist$FTAEVENT)  <- c("Appeals Court", "Govt Dismissed", "Formal Settlement", "Informal Settle
```

Next, I cleaned the relact table. I kept the variables:

ACTIVITYNO- iunique identifier for each inspection record RELTYPE- type of related activity (accident, complaint, inspections, referral) SAFETY- whether all safety hazards for complaint/ referral are satisfied HEALTH- whether all health hazards for complaint/ referral are satisfied

I made changes to any character variables that were unclearing with the coding given to us.

```
tidyrelact <- select(relact, ACTIVITYNO, RELTYPE, SAFETY, HEALTH)
rm(relact)

levels(tidyrelact$RELTYPE) <- c("Accident", "Complaint", "Inspection", "Referral")
levels(tidyrelact$SAFETY) <- c("Satisfied")
levels(tidyrelact$HEALTH) <- c("Satisfied")
```

Organizing Data

After cleaning the most useful databases, I needed to join the tables together, using OSHA as the base. The common variable throughout all the tables is ACTIVITYNO, which I used to join the columns. Some numbers in the ACTIVITYNO field appear more than once because an inspection may have more violations than one record can contain. This is more evident of a dangerous place to work because it means a lot of violated activity happened at this company.

```r
master <- full_join(tidyosha, tidyaccid, by ="ACTIVITYNO")
master <- full_join(master, tidyviol, by = "ACTIVITYNO")
master <- full_join(master, tidyhist, by = "ACTIVITYNO")
master <- full_join(master, tidyrelact, by = "ACTIVITYNO")

# Wanted to reorder the columns in the most sensible way
master <- master[c("ESTABNAME", "SITEADD", "CITY", "SITEZIP", "INDUSTRY", "OWNERTYPE", "ACTIVITYNO", "RE

# Wanted to eliminate duplicate rows
unimaster <- unique(master)
rm(master)
```

Next, I removed the cleaned data tables, just to clean up the environment.

```r
rm(acc,hzs,tidyaccid,tidyhist,tidyosha,tidyrelact,tidyviol)
```

Next, I wanted to check that the NA data was coded correctly. For this I looked at the quantitative variables and their distributions on graphs with ggplot.

```r
# If the sum of the NA values is equal to 0, then the qualitative variables were coded correctly.
# If the majority of the values are 0's it is okay. This does not mean that the activity for these reco
sum(unimaster$PASUMHOURS.has.na)
```

```
## [1] 0
```

```r
sum(unimaster$TOTALVIOLS.has.na)
```

```
## [1] 0
```

```r
sum(unimaster$TOTSERIOUS.has.na)
```

```
## [1] 0
```

```r
sum(unimaster$GRAVITY.has.na)
```

```
## [1] 0
```

```r
sum(unimaster$PENCURRENT.has.na)
```

```
## [1] 0
```

```r
sum(unimaster$TOTALFTA.has.na)
```

```
## [1] 0
```

```r
sum(unimaster$TOTPENLTY.has.na)
```

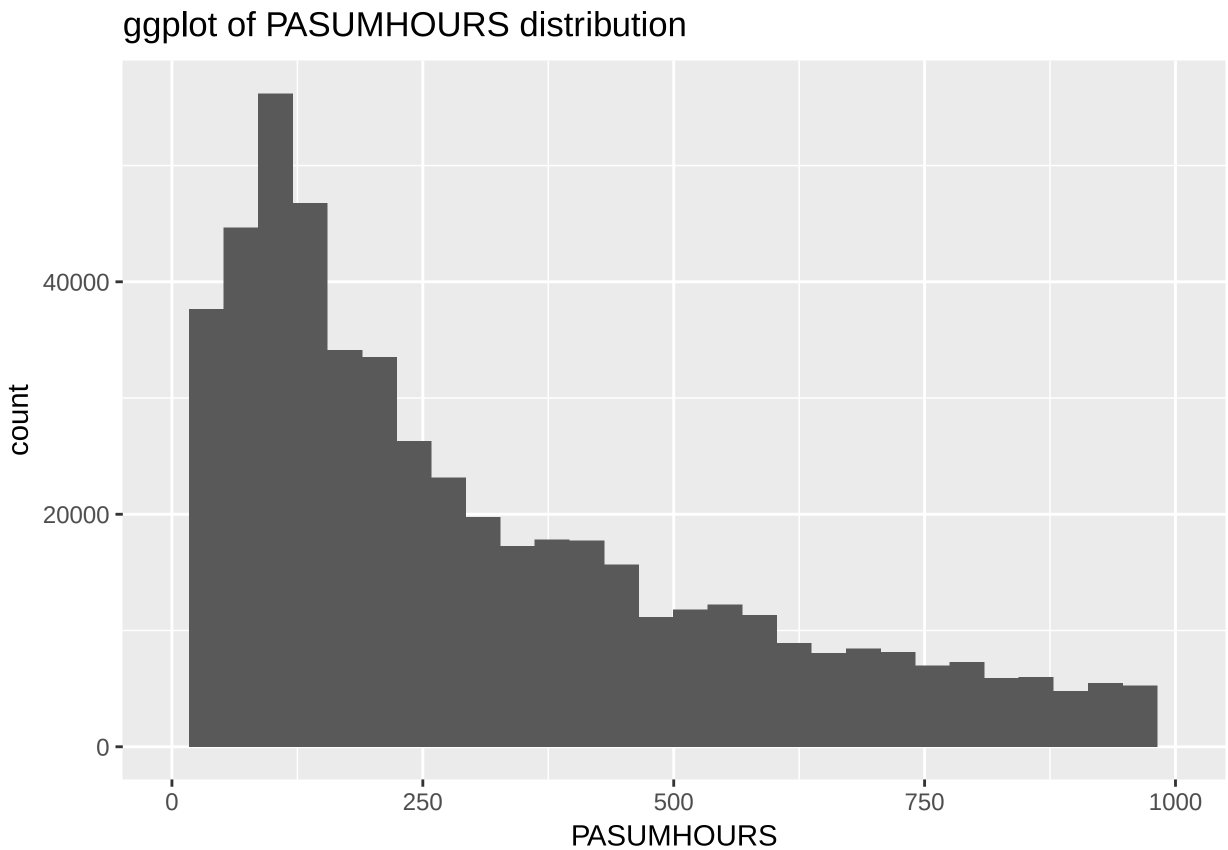
```
## [1] 0
```

```r
# Wanted to create a subset of the unimaster data frame, easier to work with
quant <- select(unimaster, PASUMHOURS, TOTALVIOLS, TOTSERIOUS, GRAVITY, PENCURRENT, TOTALFTA, TOTPENLTY)
# Histograms are used because it shwoed the distributions the best
# Used the xlim to limit the x,axis values shown, for most values, I wanted to not picture 0 values bec
ggplot(data= quant, aes(x = PASUMHOURS)) + geom_histogram() + ggtitle("ggplot of PASUMHOURS distributio
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 314255 rows containing non-finite values (stat_bin).

## Warning: Removed 1 rows containing missing values (geom_bar).
```




```
mean(quant$PASUMHOURS) # 607.5 , why we made the x-axis show values [1:1000]
```

```
## [1] 607.5038
```

```
sd(quant$PASUMHOURS) # 1218.05
```

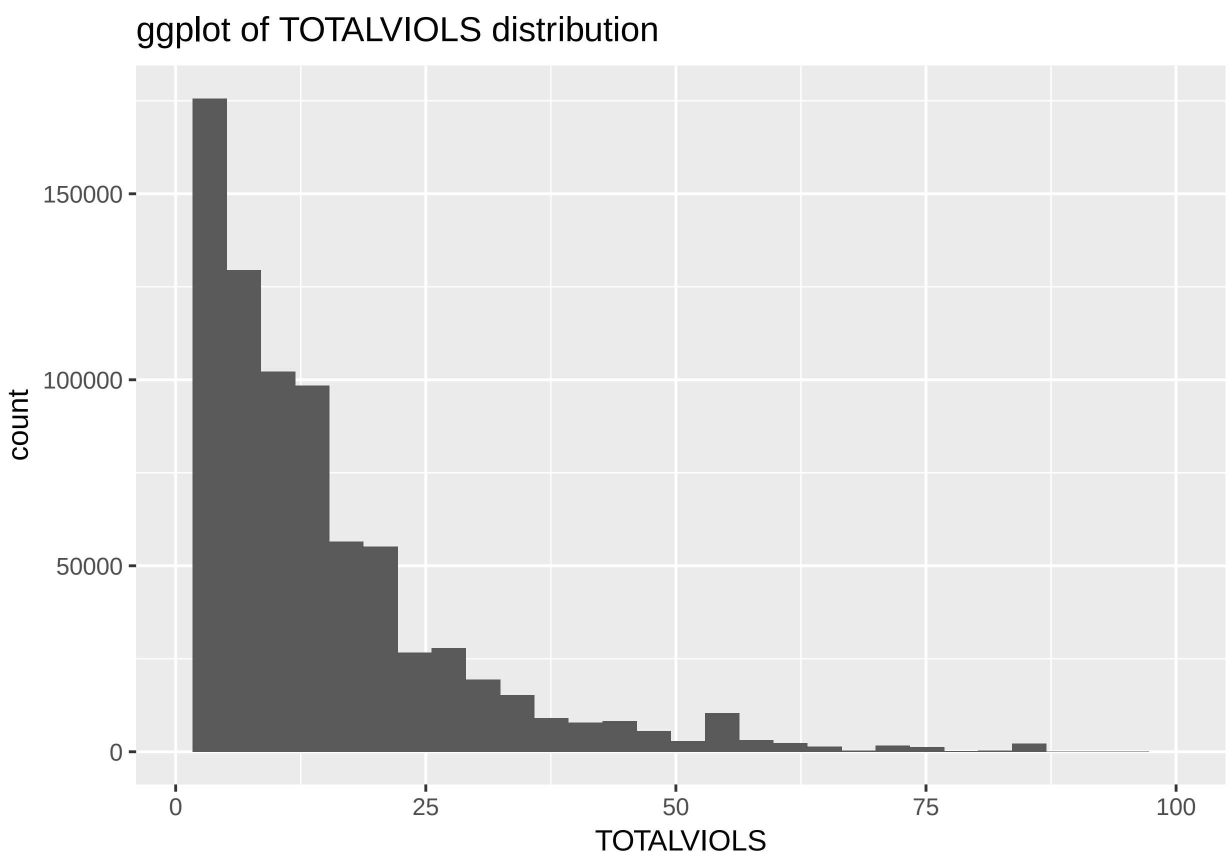
```
## [1] 1218.054
```

```
ggplot(data= quant, aes(x = TOTALVIOLS)) + geom_histogram() + ggtitle("ggplot of TOTALVIOLS distributio
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 41362 rows containing non-finite values (stat_bin).

## Warning: Removed 1 rows containing missing values (geom_bar).
```




```
mean(quant$TOTALVIOLS) # 14.76 , why we made the x-axis show values [1:100]
```

```
## [1] 14.76576
```

```
sd(quant$TOTALVIOLS) # 16.15
```

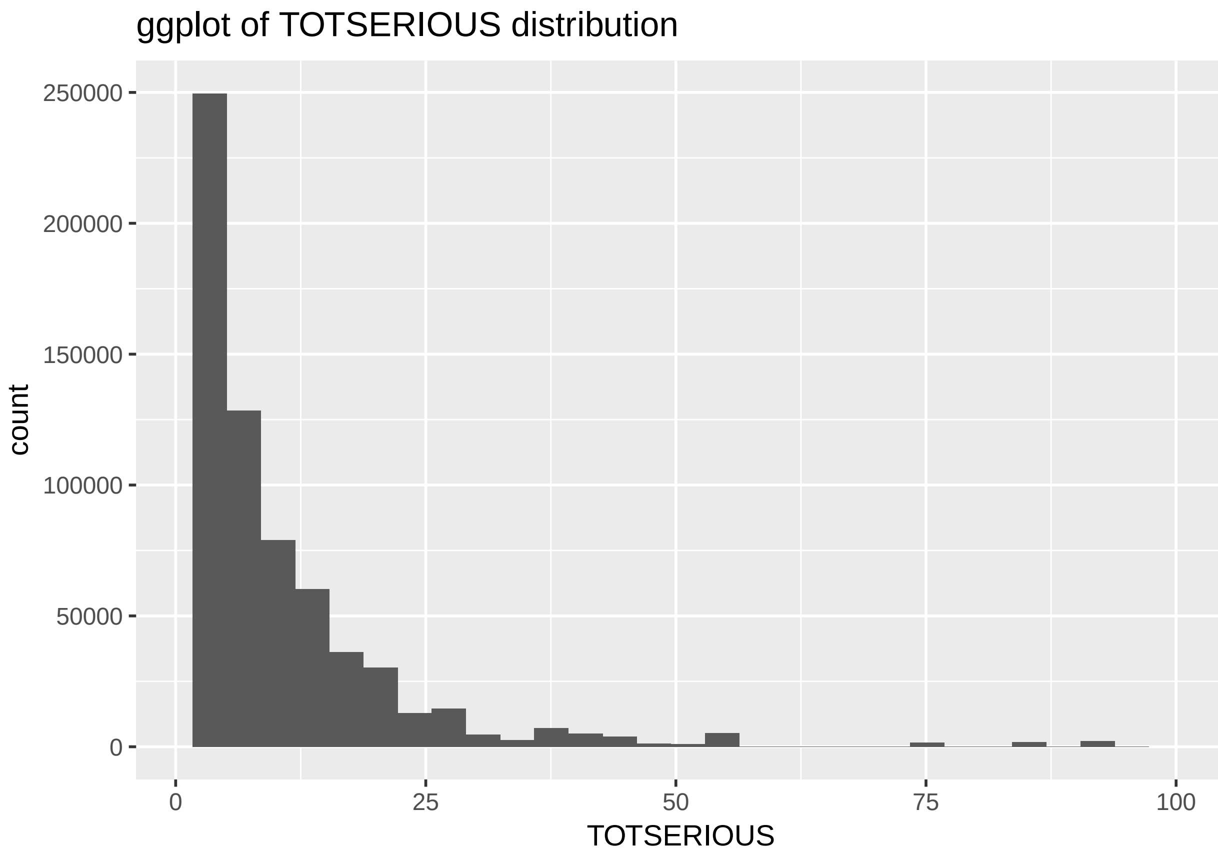
```
## [1] 16.14752
```

```
ggplot(data= quant, aes(x = TOTSERIOUS)) + geom_histogram() + ggtitle("ggplot of TOTSERIOUS distributio
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125206 rows containing non-finite values (stat_bin).

## Warning: Removed 1 rows containing missing values (geom_bar).
```



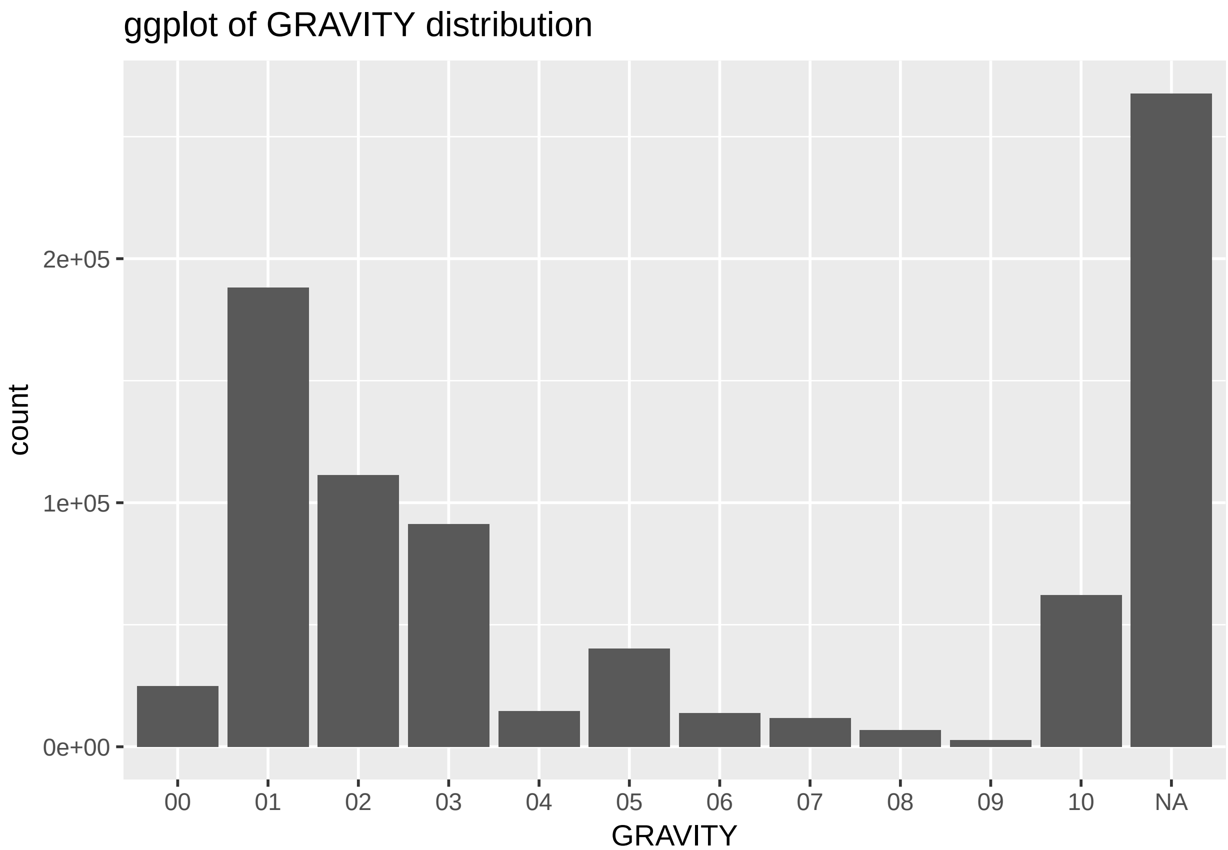

```
mean(quant$TOTSERIOUS) # 14.76 , why we made the x-axis show values [1:100]

## [1] 8.458653

sd(quant$TOTSERIOUS) # 16.15

## [1] 11.19196

ggplot(data= quant, aes(x = GRAVITY)) + geom_bar() + ggtitle("ggplot of GRAVITY distribution")
```



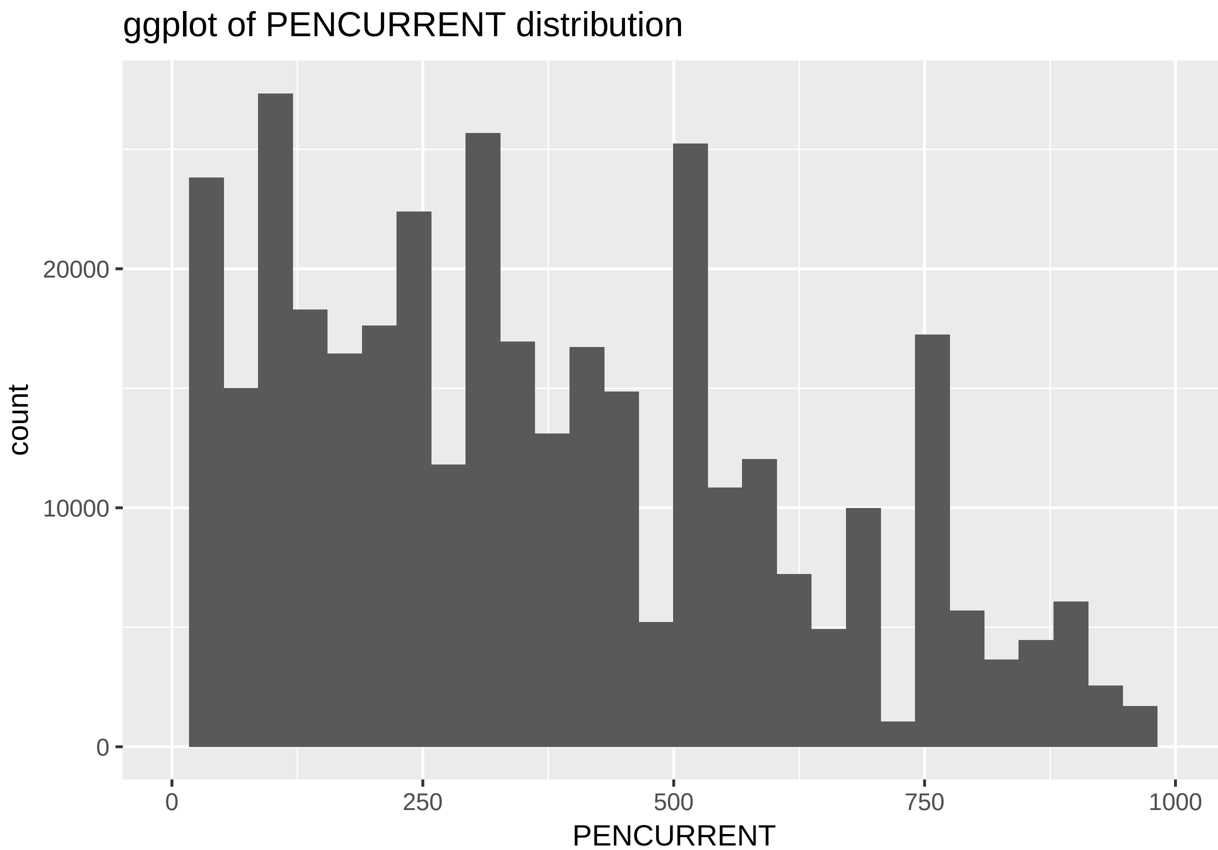

```
# A bar graph was needed because GRAVITY was determined by a scale

ggplot(data= quant, aes(x = PENCURRENT, na.rm = T)) + geom_histogram() + ggtitle("ggplot of PENCURRENT d
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 462436 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```




```
mean(quant$PENCURRENT,na.rm = T) # 513.98 , why we made the x-axis show values [1:1000]
```

```
## [1] 513.9809
```

```
sd(quant$PENCURRENT, na.rm = T) # 1897.68
```

```
## [1] 1897.682
```

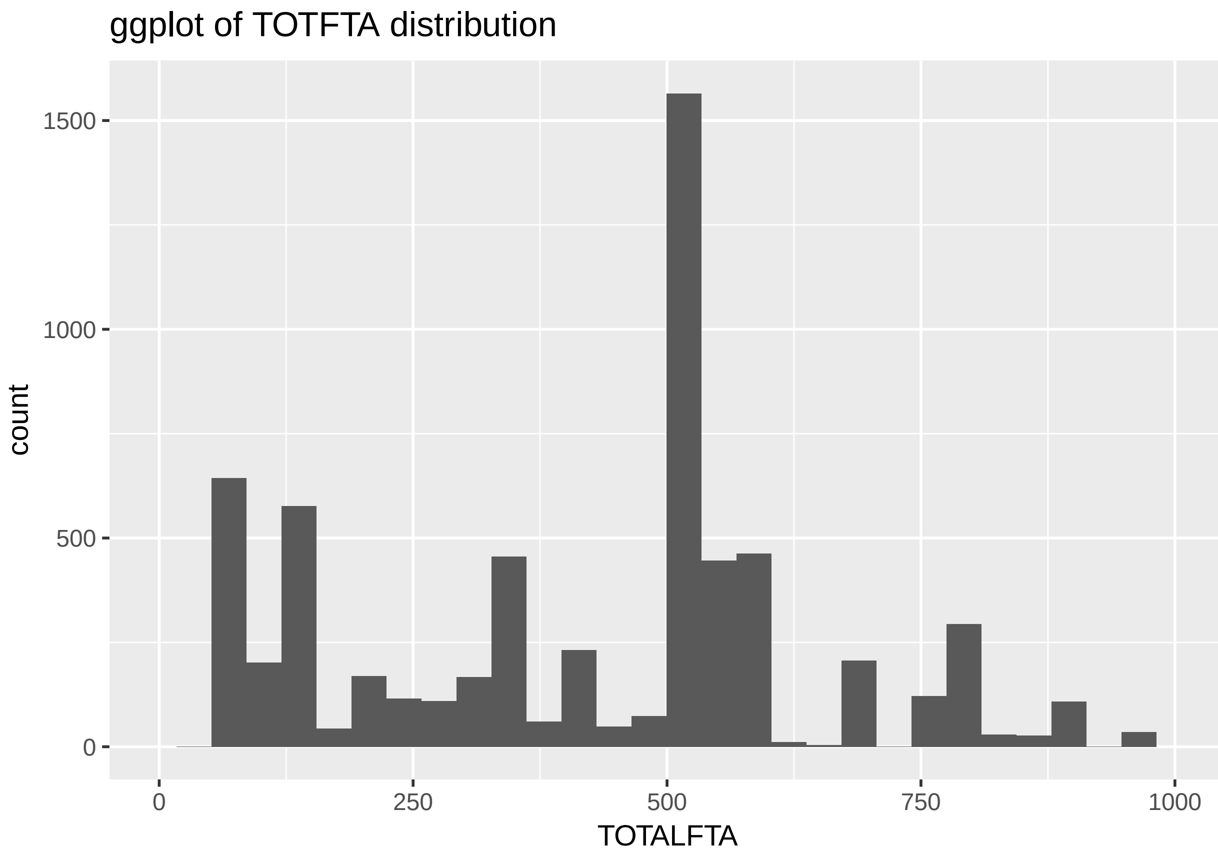
```
# Needed to omit NA values to get statistical data

ggplot(data= quant, aes(x = TOTALFTA)) + geom_histogram() + ggtitle("ggplot of TOTFTA distribution")  +
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 829297 rows containing non-finite values (stat_bin).

## Warning: Removed 1 rows containing missing values (geom_bar).
```




```
mean(quant$TOTALFTA) # 245.95 , why we made the x-axis show values [1:1000]
```

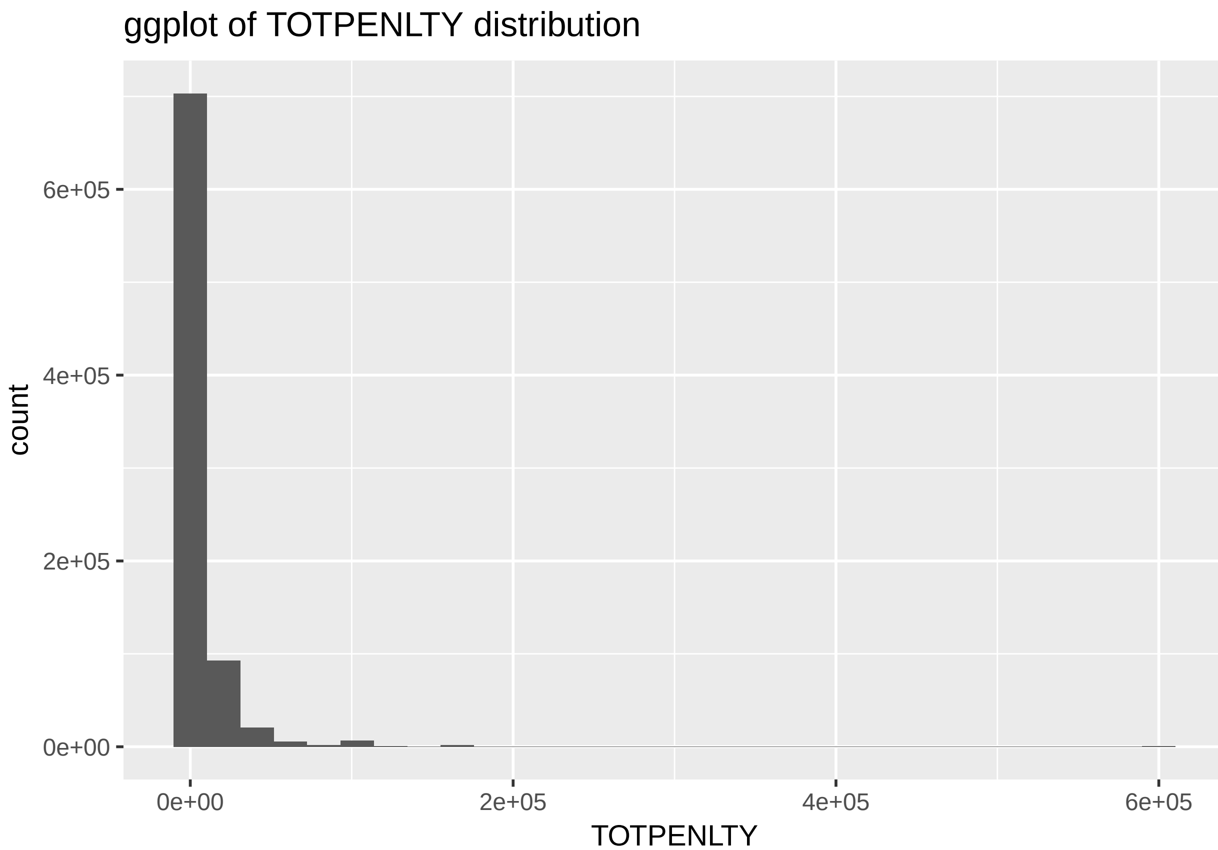
```
## [1] 245.9499
```

```
sd(quant$TOTALFTA) # 2326.74
```

```
## [1] 2326.737
```

```
ggplot(data= quant, aes(x = TOTPENLTY)) + geom_histogram() + ggtitle("ggplot of TOTPENLTY distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

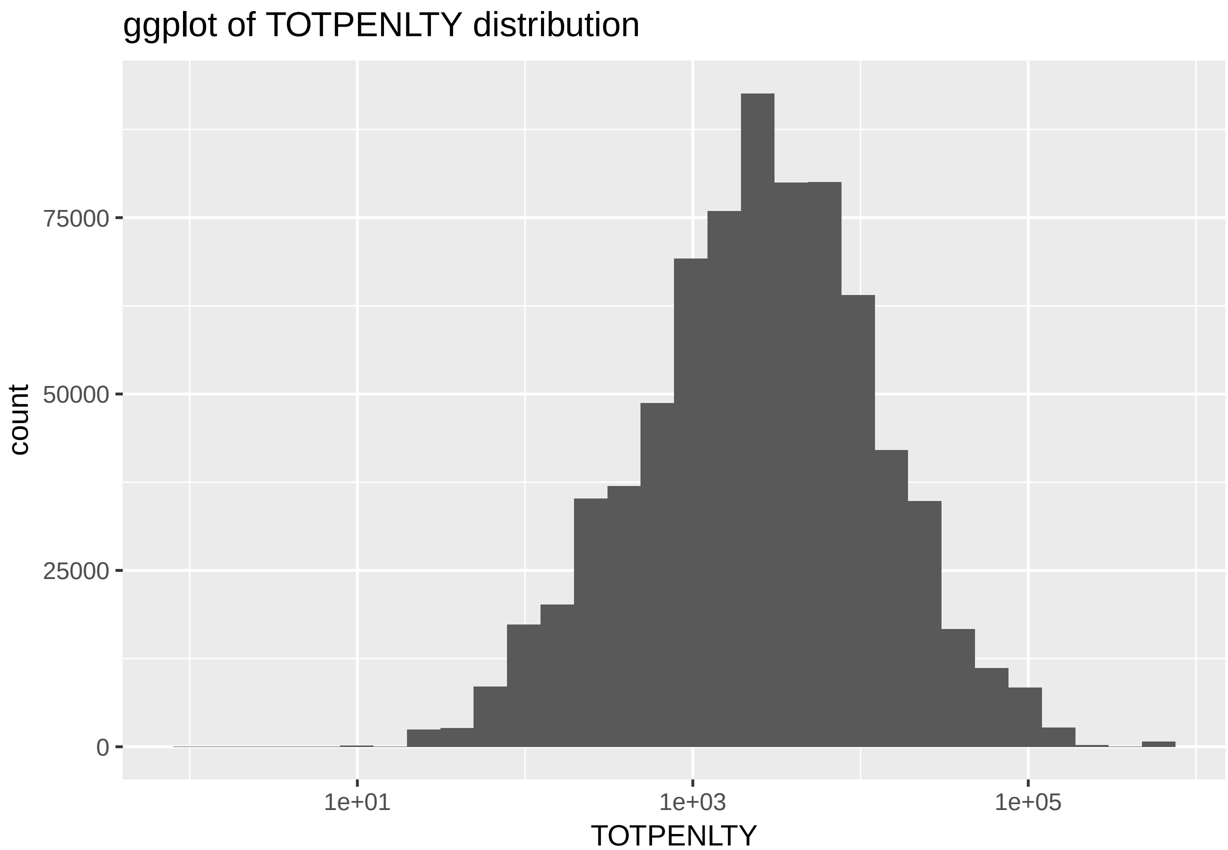



```
# it seems like the distribution is logarithmic, put the x on a log10 scale
ggplot(data= quant, aes(x = TOTPENLTY)) + geom_histogram() + ggtitle("ggplot of TOTPENLTY distribution")
```

```
## Warning: Transformation introduced infinite values in continuous x-axis

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 84805 rows containing non-finite values (stat_bin).
```




```r
# the data now looks like a normal distribution
mean(quant$TOTPENLTY) # 7660.28 , why we made the x-axis show values [1:1000]
```

```
## [1] 7660.279
```

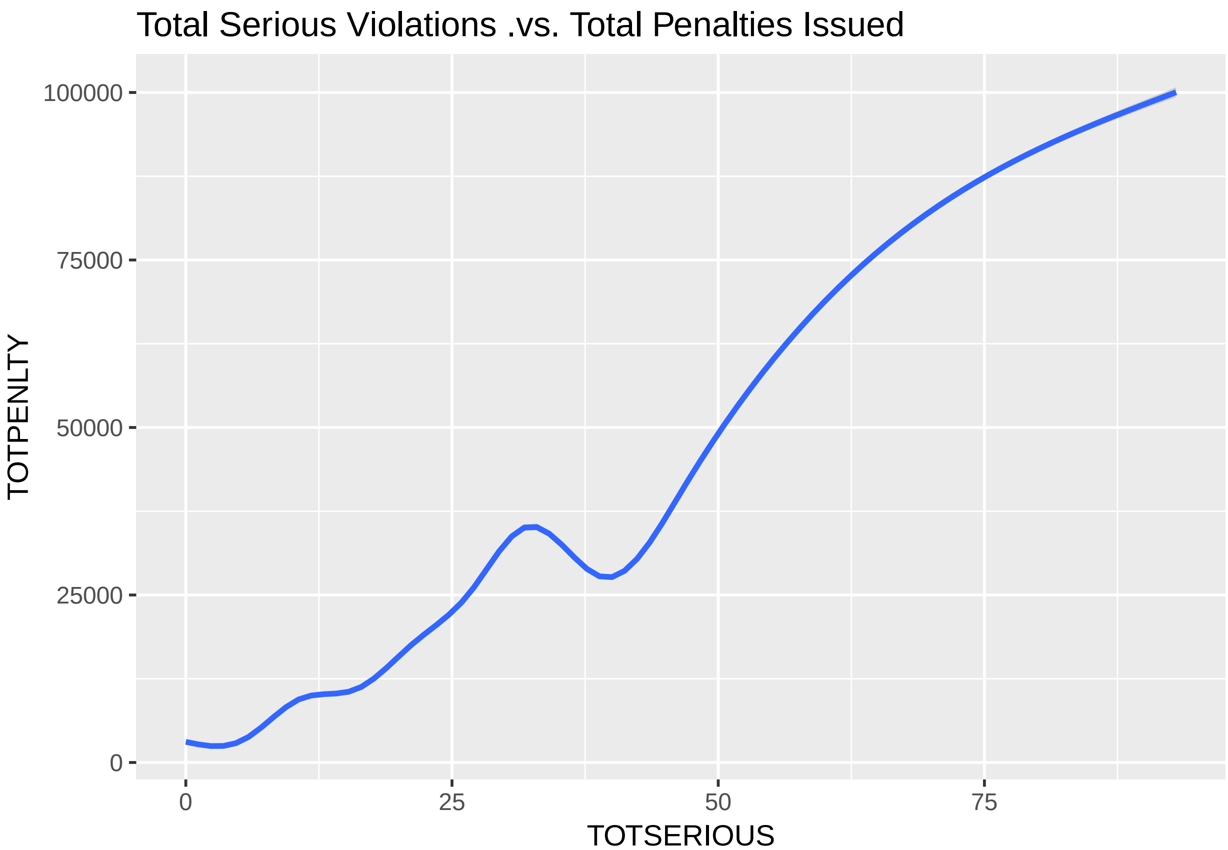
```r
sd(quant$TOTPENLTY) # 23811
```

```
## [1] 23811.12
```

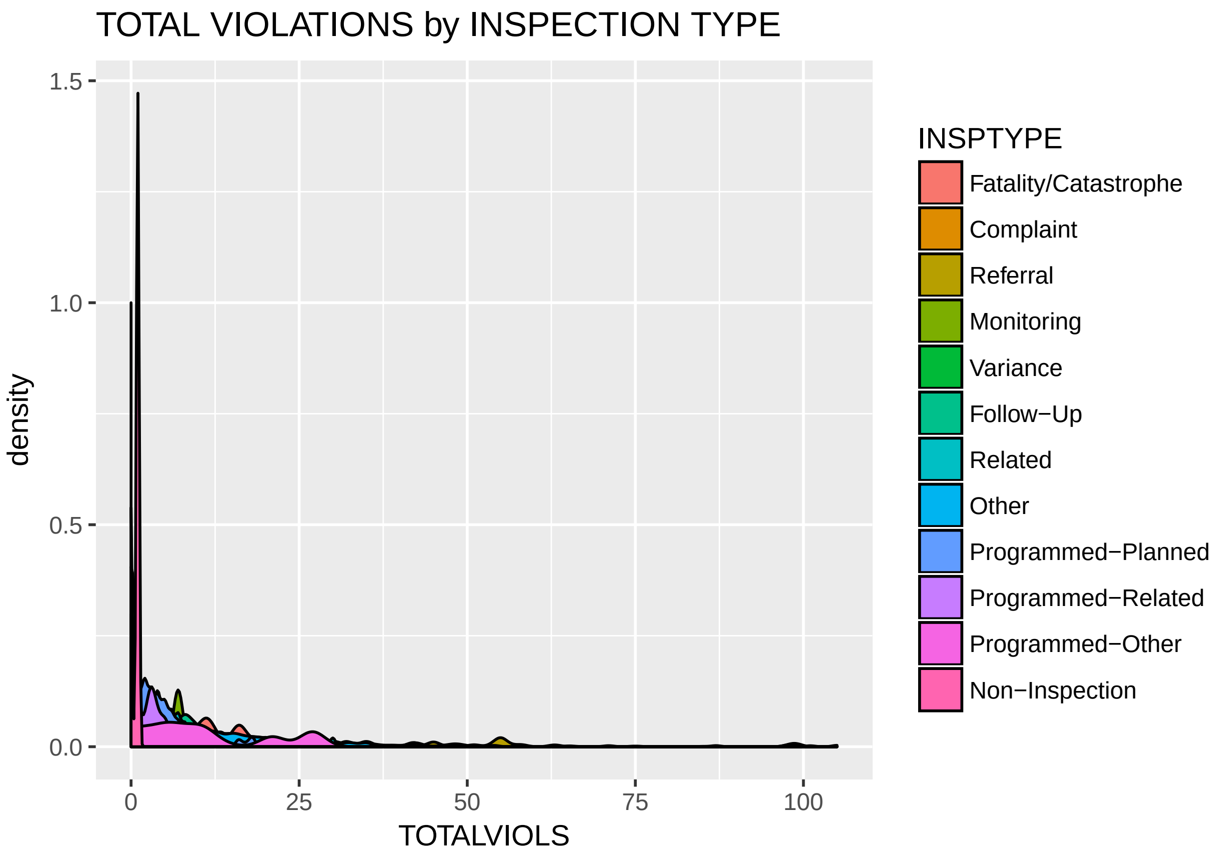
```r
# the statistical evidence confirms the need for a x log10 scaled

ggplot(data=quant) + geom_smooth(aes(x = TOTSERIOUS, y = TOTPENLTY)) + ggtitle("Total Serious Violations
```

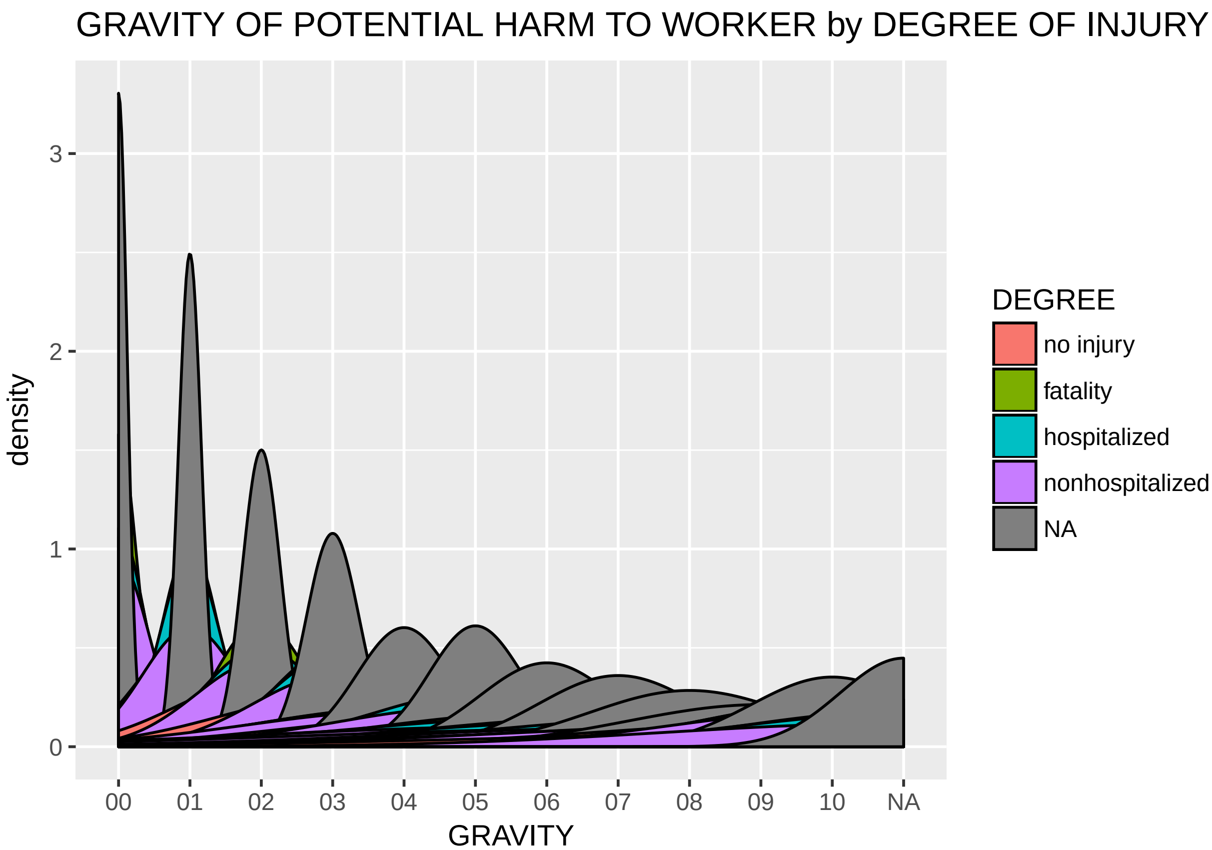
```
## `geom_smooth()` using method = 'gam'
```




```
# The increasing curve (with a positive slope) makes sense. the more serious violations a company has a
ggplot(unimaster, aes(x = TOTALVIOLS, fill = INSPTYPE)) + geom_density() + ggtitle("TOTAL VIOLATIONS by
```




```
ggplot(unimaster, aes(x = GRAVITY, fill = DEGREE )) + geom_density() + ggtitle("GRAVITY OF POTENTIAL HA
```




```
# This plot looks messy because of the NA data. We need to find a way to remove it.
```

Now I would like to look at the qualitative variables. They are coded by categories. I will make a table with counts to see if the sample size of each category is similar or if the counts make sense.

```
# Want to look at Cities
cities <- table(unimaster$CITY)
cities <- sort(cities,decreasing = T)
head(cities)
```

```
##
##      BOSTON    WORCESTER SPRINGFIELD NEW BEDFORD  FALL RIVER      QUINCY
##       90936        33076       25923       18759       17876       17369
```

```
# It makes sense that Boston, Worcster, and Springfield are the top three cities with the most OSHA Ins
```

City	BOSTON WORCESTER SPRINGFIELD NEW BEDFORD FALL RIVER QUINCY
Frequency of Inspections	90936 33076 25923 18759 17876 17369

```
# Want to look at INSPTYPE
inspect_type <- table(unimaster$INSPTYPE)
inspect_type <- sort(inspect_type, decreasing = T)
head(inspect_type)
```

```
##
##                 Other              Complaint               Referral
##                378779                 221481                 124333
##               Related Fatality/Catastrophe   Programmed-Planned
```

```
##                    45463                   24846                   23419
#These are the top 5 types of OSHA inspections in Massachusetts
```

Type of Inspection	Other Complaint Referral Related Fatality/Catastrophe Programmed-Planned
Count	378779 221481 124333 45463 24846 23419

In closing, the data frame unimaster should be handed off to someone doing the analysis. Upon exploring the different quatitative and qualitative variables, it would be most useful to look at the table from left to right. First, there is information about the different establishment- such as the address and type of company. Then there are details about the OSHA inspections. Next comes accidents, violations, and penalties. I chose this order because it decreases from most pertinant/ most dangerous factors of a workplace to least dangerous factors.