Title: Data Challenge #1

Authors: Brandon Wolff, Zachary Heinemann, and Stephanie Langeland

Due Date: 02/22/2017

Cleaning the data:

```
rm(list = ls(all = TRUE))   # cleans everything in the workspace

library(readr)          # easier reading of flat files
library(readxl)         # easier reading of excel files
library(dplyr)          # data manipulation functions
```

```
##
## 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
```

```
library(tidyr)          # tools for tidy datasets
```

```
## Warning: package 'tidyr' was built under R version 3.3.2
```

```
library(magrittr)       # this is not a pipe
```

```
##
## Attaching package: 'magrittr'
```

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

```
library(lubridate)      # easier manipulation of time objects
```

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

```
library(stringr)        # easier manipulation of strings
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.3.2
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: purrr
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## as.difftime(): lubridate, base
## date():        lubridate, base
## filter():      dplyr, stats
## intersect():   lubridate, base
## lag():         dplyr, stats
## setdiff():     lubridate, base
## union():       lubridate, base
```

```
path <- "/Users/StephanieLangeland/Desktop/Columbia/Applied Data Science/Git/QMSS_G5069_
Applied_D_S/Data Challenges"

#path <- "C:\\Users\\Brandon\\Documents\\GitHub\\QMSS_G5069_Applied_D_S\\Data Challenge
s"

# define additional paths for files you will use. In each case, determine
# appropriate additions to the path:
inFileName1  <- "A-E.xlsx"           # raw data on confrontations
inFileName2  <- "ARCH535.csv"   # name equivalence tables
outFileName1 <- "ConfrontationsData_170209.csv" # output file name

# set your path to that defined above, and confirm it
setwd(path)
getwd()
```

```
## [1] "/Users/StephanieLangeland/Desktop/Columbia/Applied Data Science/Git/QMSS_G5069_A
pplied_D_S/Data Challenges"
```

```
# LOADING RAW DATA
# the original file uses 9999 as a sentinel value for missing values changing
# back to null upon loading

library(readxl)
Confrontations <- read_excel(inFileName1,
                             sheet = 1,
                             na = "9999"   # converting sentinel value to null
)

# rough validations that data was correctly loaded
names(Confrontations)
```

```
##  [1] "ID"        "TIMESTAMP" "DIA"       "MES"       "AÑO"      
##  [6] "ESTADO"    "Municipio" "DE"        "PF"        "MIF"      
## [11] "MAF"       "PFF"       "AFIF"      "PEF"       "PMF"      
## [16] "PMUF"      "AMPF"      "DOF"       "CIF"       "PL"       
## [21] "MIL"       "MAL"       "PFL"       "AFIFL"     "PEL"      
## [26] "PML"       "PMUL"      "AMPL"      "DOL"       "CIL"      
## [31] "ARL"       "ARC"       "CARG"      "CART"      "VE"       
## [36] "AC"        "AP"        "DEL"       "TOR"       "DTRA"     
## [41] "PRE"       "FCRU"      "ELE"       "TAX"       "DRO"      
## [46] "VEH"       "VAL"
```

```
nrow(Confrontations)
```

```
## [1] 3835
```

```
summary(Confrontations)
```

```
##        ID           TIMESTAMP              DIA             MES
##  Min.   :   1.0   Min.   :1.169e+09   Min.   : 1.00   Min.   : 1.000
##  1st Qu.: 959.5   1st Qu.:1.255e+09   1st Qu.: 8.00   1st Qu.: 4.000
##  Median :1918.0   Median :1.285e+09   Median :16.00   Median : 7.000
##  Mean   :1918.0   Mean   :1.276e+09   Mean   :15.81   Mean   : 6.488
##  3rd Qu.:2876.5   3rd Qu.:1.304e+09   3rd Qu.:23.00   3rd Qu.: 9.000
##  Max.   :3835.0   Max.   :1.322e+09   Max.   :31.00   Max.   :12.000
##
##       AÑO           ESTADO        Municipio           DE
##  Min.   :2007   Min.   : 1.00   Min.   :  1.0   Min.   : 0.000
##  1st Qu.:2009   1st Qu.:12.00   1st Qu.: 13.0   1st Qu.: 1.000
##  Median :2010   Median :19.00   Median : 27.0   Median : 2.000
##  Mean   :2010   Mean   :18.95   Mean   : 35.3   Mean   : 3.563
##  3rd Qu.:2011   3rd Qu.:28.00   3rd Qu.: 39.0   3rd Qu.: 4.000
##  Max.   :2011   Max.   :32.00   Max.   :469.0   Max.   :40.000
##                                 NA's   :1       NA's   :2388
##        PF              MIF            MAF             PFF
##  Min.   : 0.000   Min.   :1.00   Min.   :1.000   Min.   :1.000
##  1st Qu.: 1.000   1st Qu.:1.00   1st Qu.:1.000   1st Qu.:1.000
##  Median : 2.000   Median :1.00   Median :1.000   Median :1.000
##  Mean   : 2.509   Mean   :1.31   Mean   :1.357   Mean   :1.723
##  3rd Qu.: 3.000   3rd Qu.:1.00   3rd Qu.:1.000   3rd Qu.:2.000
##  Max.   :29.000   Max.   :6.00   Max.   :3.000   Max.   :8.000
##  NA's   :1669     NA's   :3748   NA's   :3821    NA's   :3788
##       AFIF           PEF             PMF             PMUF
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.25   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.00   Median :1.000   Median :1.000   Median :1.000
##  Mean   :2.50   Mean   :1.667   Mean   :1.667   Mean   :1.609
##  3rd Qu.:2.75   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :6.00   Max.   :6.000   Max.   :7.000   Max.   :7.000
##  NA's   :3829   NA's   :3787    NA's   :3790    NA's   :3748
##       AMPF           DOF               CIF              PL
##  Min.   : NA    Min.   : 0.000   Min.   : 0.000   Min.   : 1.000
##  1st Qu.: NA    1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : NA    Median : 2.000   Median : 1.000   Median : 2.000
##  Mean   :NaN    Mean   : 2.459   Mean   : 1.679   Mean   : 2.272
##  3rd Qu.: NA    3rd Qu.: 3.000   3rd Qu.: 2.000   3rd Qu.: 3.000
##  Max.   : NA    Max.   :29.000   Max.   :10.000   Max.   :30.000
##  NA's   :3835   NA's   :1991     NA's   :3611     NA's   :2172
##       MIL             MAL            PFL             AFIFL
##  Min.   :1.000   Min.   :1.00   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:1.000   1st Qu.:1.00   1st Qu.: 1.000   1st Qu.: 1.000
##  Median :1.000   Median :2.00   Median : 2.000   Median : 1.000
##  Mean   :2.003   Mean   :2.48   Mean   : 2.405   Mean   : 2.615
##  3rd Qu.:3.000   3rd Qu.:3.00   3rd Qu.: 3.000   3rd Qu.: 3.000
##  Max.   :9.000   Max.   :9.00   Max.   :16.000   Max.   :15.000
##  NA's   :3516    NA's   :3810   NA's   :3724     NA's   :3822
##       PEL             PML             PMUL            AMPL
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.944   Mean   :1.908   Mean   :1.834   Mean   :1.333
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:1.500
##  Max.   :8.000   Max.   :7.000   Max.   :8.000   Max.   :2.000
##  NA's   :3746    NA's   :3748    NA's   :3660    NA's   :3832
##       DOL              CIL              ARL               ARC
##  Min.   : 1.000   Min.   : 1.000   Min.   :  1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:  2.000   1st Qu.: 1.000
##  Median : 1.000   Median : 1.000   Median :  3.000   Median : 2.000
##  Mean   : 1.881   Mean   : 1.943   Mean   :  5.175   Mean   : 2.436
##  3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.:  6.000   3rd Qu.: 3.000
##  Max.   :30.000   Max.   :27.000   Max.   :144.000   Max.   :34.000
##  NA's   :3052     NA's   :3499     NA's   :2139      NA's   :2781
##       CARG              CART             VE                AC
##  Min.   :   1.00   Min.   :    1   Min.   :  1.000   Min.   :0.00000
##  1st Qu.:   5.00   1st Qu.:   79   1st Qu.:  1.000   1st Qu.:0.00000
##  Median :  19.00   Median :  402   Median :  1.000   Median :0.00000
##  Mean   :  46.26   Mean   : 1171   Mean   :  2.779   Mean   :0.01904
##  3rd Qu.:  45.00   3rd Qu.: 1180   3rd Qu.:  3.000   3rd Qu.:0.00000
##  Max.   :4000.00   Max.   :86365   Max.   :354.000   Max.   :1.00000
##  NA's   :2493      NA's   :2612    NA's   :1990
##        AP             DEL               TOR                DTRA
##  Min.   :0.000   Min.   :0.00000   Min.   :0.000000   Min.   :   0.0
##  1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.: 999.5
##  Median :0.000   Median :0.00000   Median :0.000000   Median :1342.0
##  Mean   :0.261   Mean   :0.07458   Mean   :0.002086   Mean   :1239.9
##  3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:1567.0
##  Max.   :1.000   Max.   :1.00000   Max.   :1.000000   Max.   :1776.0
##
##       PRE                 FCRU             ELE
##  Min.   :0.0000000   Min.   :0.0000   Min.   :0.0000000
##  1st Qu.:0.0000000   1st Qu.:0.0000   1st Qu.:0.0000000
##  Median :0.0000000   Median :0.0000   Median :0.0000000
##  Mean   :0.0007823   Mean   :0.4931   Mean   :0.0002608
##  3rd Qu.:0.0000000   3rd Qu.:1.0000   3rd Qu.:0.0000000
##  Max.   :1.0000000   Max.   :1.0000   Max.   :1.0000000
##
##       TAX               DRO               VEH              VAL
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :1.0000   Median :0.0000
##  Mean   :0.01095   Mean   :0.03051   Mean   :0.5129   Mean   :0.2334
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##
```

```
# :::::: LOADING NAME CONVERSION TABLE
# the original file treats numeric codes as strings, must convert to integers
# upon loading. Also, names of municipalities are in Spanish, so must specify
# the encoding as the file is read

NameTable <- read_csv(inFileName2,
                      col_types = cols(
                          CVE_ENT = col_integer(),    # must convert to integer
                          NOM_ENT = col_character(),
                          NOM_ABR = col_character(),
                          CVE_MUN = col_integer(),    # must convert to integer
                          NOM_MUN = col_character()
                          ),
                      locale = locale(encoding = "ISO-8859-1") # to read accents properl
y
                      )

# rough validations that data was correctly loaded
names(NameTable)
```

```
## [1] "CVE_ENT" "NOM_ENT" "NOM_ABR" "CVE_MUN" "NOM_MUN"
```

```
nrow(NameTable)
```

```
## [1] 2458
```

```
summary(NameTable)
```

```
##     CVE_ENT        NOM_ENT            NOM_ABR             CVE_MUN     
##  Min.   : 1.00   Length:2458        Length:2458        Min.   :  1.0  
##  1st Qu.:14.00   Class :character   Class :character   1st Qu.: 23.0  
##  Median :20.00   Mode  :character   Mode  :character   Median : 56.0  
##  Mean   :19.26                                         Mean   :108.8  
##  3rd Qu.:24.00                                         3rd Qu.:128.8  
##  Max.   :32.00                                         Max.   :570.0  
##    NOM_MUN         
##  Length:2458       
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
```

```
# SOME DATA PROCESSING
# as released, the database is not immediately usable, so some data processing
# is needed to start exploring the data

# 1. add actual names of states and municipalities from a Census table;
#    currently the database only has their numeric codes
# 2. rename columns from Spanish to English (not everyone speaks both languages)
# 3. convert UNIX timestamp variable to a time object; this will be useful to
#    seamlessly create a date variable, and extract month names for graphing
# 4. some additional string changes in state abbreviations that will be useful
#    when graphing
# 5. replace all missing values with 0; this will come in handy as we start to
#    explore the data futher


fullData <-
    Confrontations %>%
        # adding State and Municipality names to dataframe
        left_join(., NameTable,
                  by = c("ESTADO" = "CVE_ENT",
                         "Municipio" = "CVE_MUN")
        ) %>%
        # renaming variables to intelligible English
        rename(day.orig = DIA,
               month.orig = MES,
               #year.orig = AÃ‘O,#had to change this part
               #to run the code on windows
               year.orig = AÑO,
               state_code = ESTADO,
               mun_code = Municipio,
               state = NOM_ENT,
               state.abbr = NOM_ABR,
               municipality = NOM_MUN,
               event.id = ID,
               unix.timestamp = TIMESTAMP,
               detained = DE,
               total.people.dead = PF,
               military.dead = MIF,
               navy.dead = MAF,
               federal.police.dead = PFF,
               afi.dead = AFIF,
               state.police.dead = PEF,
               ministerial.police.dead = PMF,
               municipal.police.dead = PMUF,
               public.prosecutor.dead = AMPF,
               organized.crime.dead = DOF,
               civilian.dead = CIF,
               total.people.wounded = PL,
               military.wounded = MIL,
               navy.wounded = MAL,
               federal.police.wounded = PFL,
               afi.wounded = AFIFL,
               state.police.wounded = PEL,
               ministerial.police.wounded = PML,
               municipal.police.wounded = PMUL,
               public.prosecutor.wounded = AMPL,
               organized.crime.wounded = DOL,
               civilian.wounded = CIL,
               long.guns.seized = ARL,
               small.arms.seized = ARC,
               cartridge.sezied = CART,
               clips.seized = CARG,
               vehicles.seized = VE
        ) %>%
        # creating date by converting unix timestamp, other time-related information
        # can later be extracted from this variable
        # also modifying state abbreviations by capitalizing and droping period
        # to "beautify" graph labels later on
        mutate(date = as.Date(as.POSIXct(unix.timestamp, origin="1970-01-01")),
               state.abbr = str_to_upper(str_replace_all(state.abbr, "[[:punct:]]", ""))

        ) %>%
        # keeping only necessary variables
        select(event.id, unix.timestamp, date,
               state_code, state, state.abbr, mun_code, municipality,
               detained, total.people.dead, military.dead, navy.dead,
               federal.police.dead, afi.dead, state.police.dead,
ministerial.police.dead,
               municipal.police.dead, public.prosecutor.dead, organized.crime.dead,
               civilian.dead, total.people.wounded, military.wounded, navy.wounded,
               federal.police.wounded, afi.wounded, state.police.wounded,
               ministerial.police.wounded, municipal.police.wounded,
               public.prosecutor.wounded, organized.crime.wounded, civilian.wounded,
               long.guns.seized, small.arms.seized, cartridge.sezied, clips.seized,
               vehicles.seized
        ) %>%
        # filling in NAs with zeros, to facilitate graphing and basic computations
        # replace_na() requires a list of columns and rules to apply. Code below
        # provides that
        replace_na(
            setNames(            # creates an object with numeric column names
                lapply(          # applies a function that links numeric column names
                                 # with the asignment of 0
                    vector("list", length(select_if(., is.numeric))), # creates a list l
ength 25
                            function(x) x <- 0),  # defines assignment of 0 to numeric c
ol names
                names(select_if(., is.numeric)))  # provides numeric column names
        )
```

1) Can you replicate the 86.1% number? The overall lethality ratio?

The ratios for the Federal Police, Navy and Army?

- These figures cannot be reproduced because the dataset does not include civilians who were involved in these events who were neither wounded nor killed.
  This makes it impossible to reproduce the overall lethality figure. Additionally, the dataset does not

distinguish between civilians killed or wounded by federal police, army, or navy personnel making it impossible to reproduce the 86.1% figure and lethality ratios for the navy, army, and federal police.

1a) Provide a visualization that presents this information neatly.

- Not applicable - see response to #1 above.

1b) Please show the exact computations you used to calculate them

(most likely than not, you’ll need to do some additional munging

in the data to get there).

- Not applicable - see response to #1 above.

1c) If you could not replicate them, please show why and the difference

relative to your own computations (also, include a neat graph that summarizes

this).

```
#Group Calculations:
#civilian lethality%
fullData$Total.Civilian.Conf <- fullData$civilian.dead + fullData$civilian.wounded
civilian_lethality <- (sum(fullData$civilian.dead))/sum((fullData$Total.Civilian.Conf))
civilian_lethality
```

```
## [1] 0.3654033
```

```
fullData$Civilian.lethality <- (fullData$civilian.dead)/(fullData$Total.Civilian.Conf)
valid.cases <- 3835-sum(is.na(fullData$Civilian.lethality))
valid.cases
```

```
## [1] 495
```

```
civ_leth_by_case <- sum(fullData$Civilian.lethality, na.rm = TRUE)/495
civ_leth_by_case
```

```
## [1] 0.37937
```

```
#Total Lethality%
fullData$Total.Conf <- fullData$total.people.dead + fullData$total.people.wounded
Total_lethality <- (sum(fullData$total.people.dead))/sum((fullData$Total.Conf))
Total_lethality
```

```
## [1] 0.5898633
```

```
#organized crime lethality%
fullData$Total.organized.crime.Conf <- fullData$organized.crime.dead + fullData$organized.crime.wounded
organized_crime_lethality <- (sum(fullData$organized.crime.dead))/sum((fullData$Total.organized.crime.Conf))
organized_crime_lethality
```

```
## [1] 0.7548269
```

```
#Federal Police lethality%
fullData$Total.Federal.Police.Conf <- fullData$federal.police.dead + fullData$federal.police.wounded
Federal_Police_lethality <- (sum(fullData$federal.police.dead))/sum((fullData$Total.Federal.Police.Conf))
Federal_Police_lethality
```

```
## [1] 0.2327586
```

```
#Federal Police deaths per 1 wounded
Federal_Police_lethality2 <- (sum(fullData$federal.police.dead))/sum((fullData$federal.police.wounded))
Federal_Police_lethality2
```

```
## [1] 0.3033708
```

```
#Navy Lethality%
fullData$Total.Navy.Conf <- fullData$navy.dead + fullData$navy.wounded
Navy_lethality <- (sum(fullData$navy.dead))/sum((fullData$Total.Navy.Conf))
Navy_lethality
```

```
## [1] 0.2345679
```

```
#ARMY deaths per 1 wounded
Navy_lethality2 <- (sum(fullData$navy.dead))/sum((fullData$navy.wounded))
Navy_lethality2
```

```
## [1] 0.3064516
```

```
#Army Lethality%
fullData$Total.military.Conf <- fullData$military.dead + fullData$military.wounded
Military_lethality <- (sum(fullData$military.dead))/sum((fullData$Total.military.Conf))
Military_lethality
```

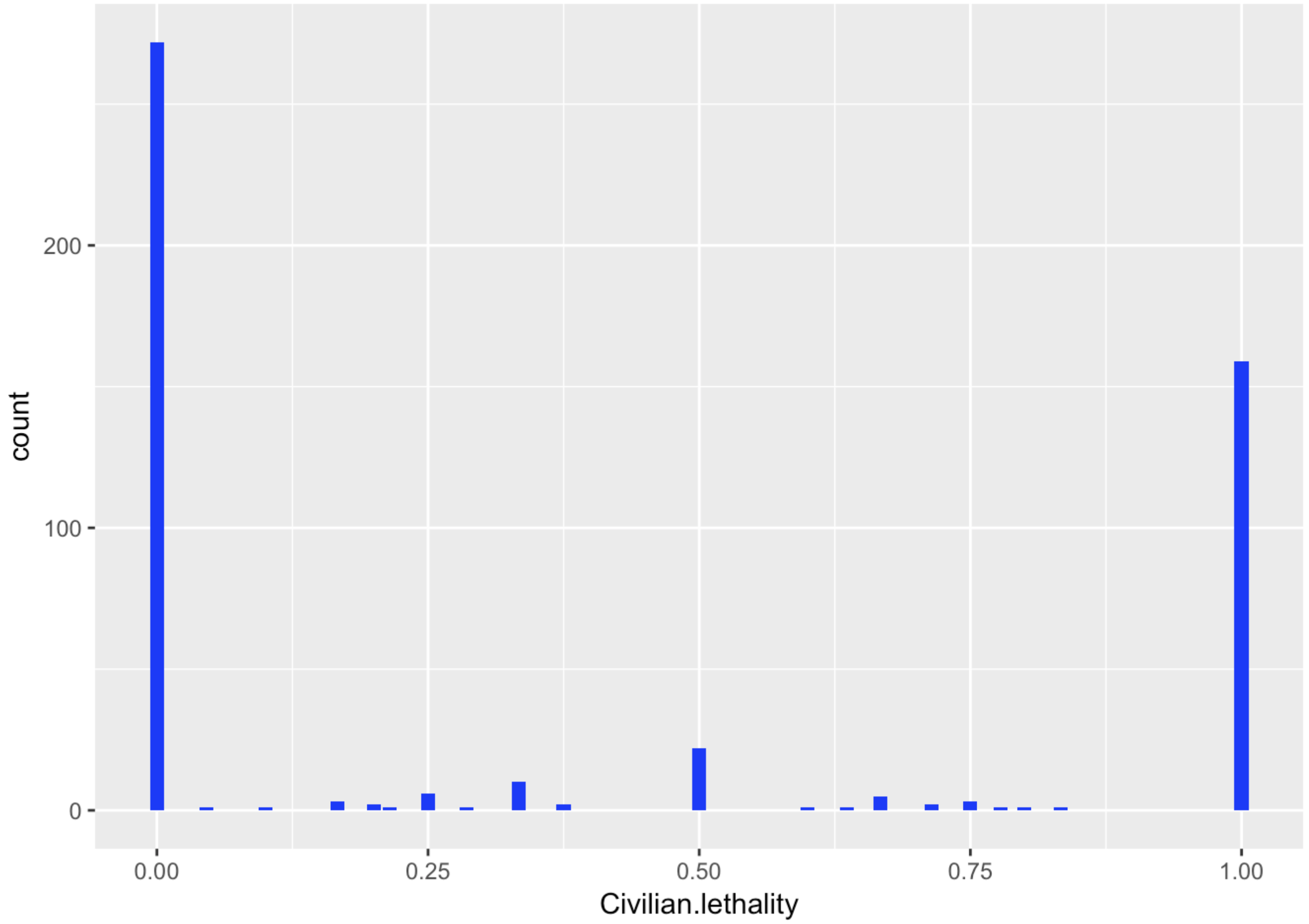
```
## [1] 0.1513944
```

```
#ARMY deaths per 1 wounded
Military_lethality2 <- (sum(fullData$military.dead))/sum((fullData$military.wounded))
Military_lethality2
```

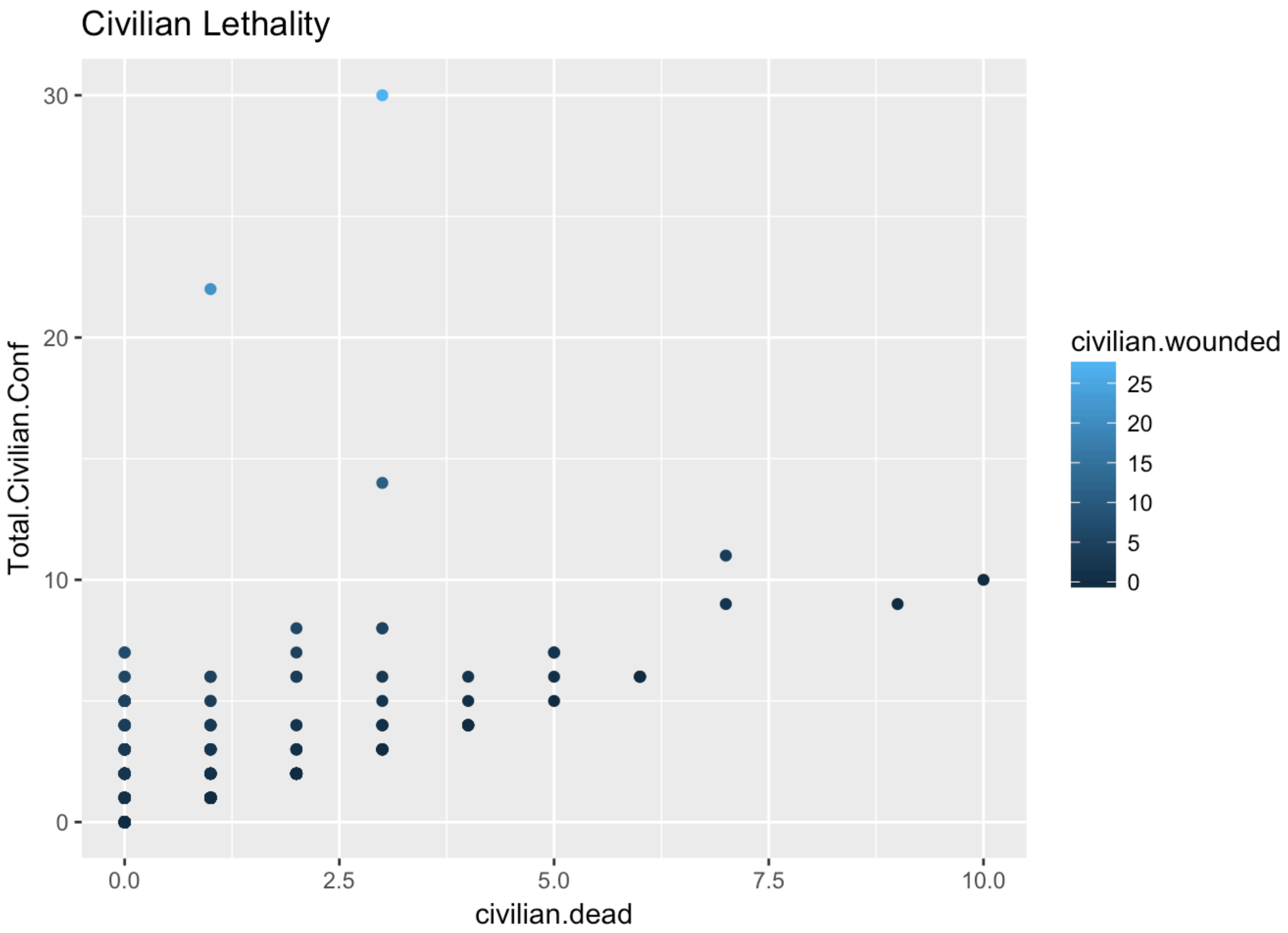
```
## [1] 0.1784038
```

```
#Visualizations:
b <- ggplot(fullData)
b <- b + geom_bar(mapping = aes(Civilian.lethality), fill = "blue")
b
```

```
## Warning: Removed 3340 rows containing non-finite values (stat_count).
```

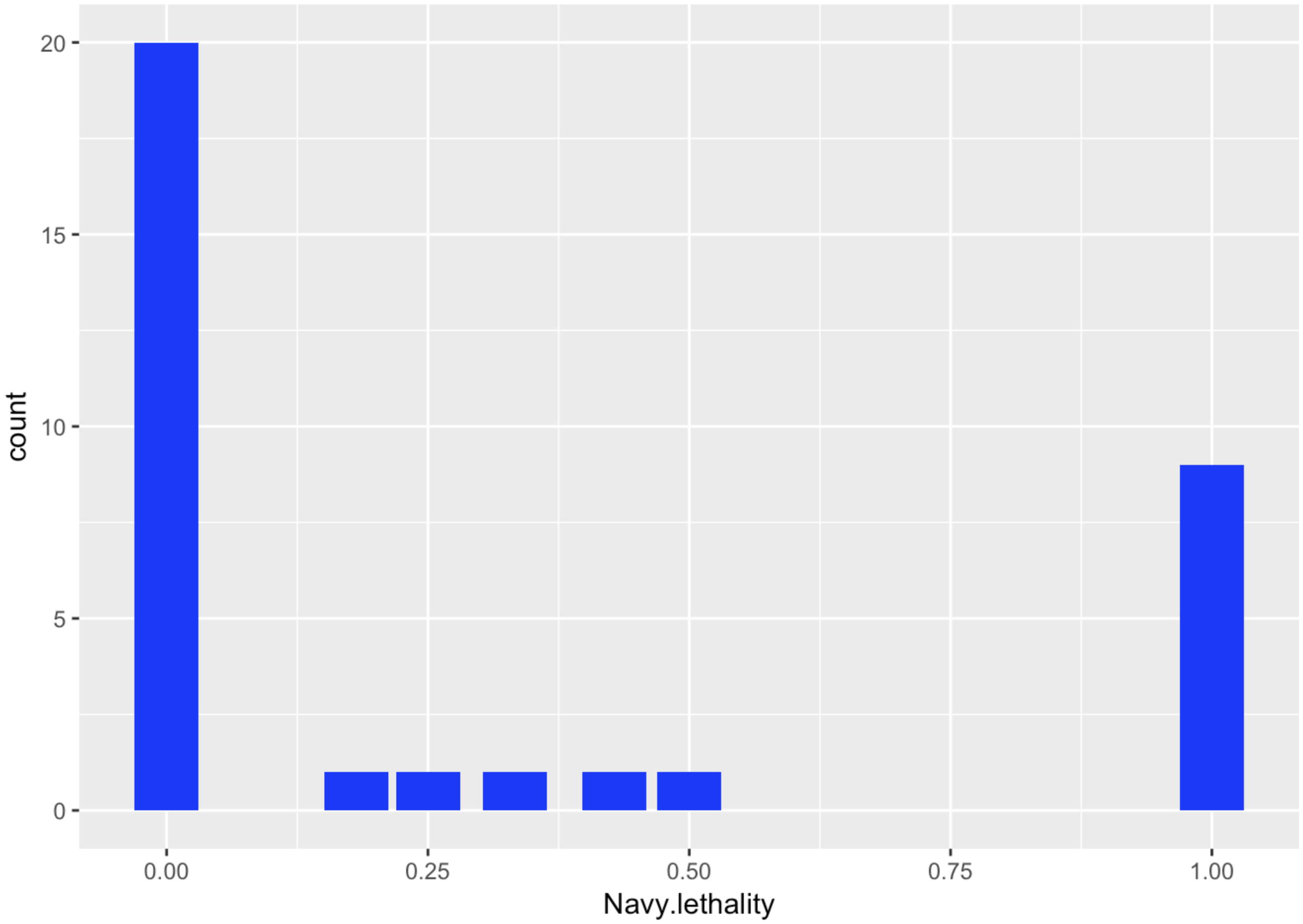


```
#in this graph 0 = wounded with no deaths and 1 = deaths with no wounded
#put this graph here to show the exterme difference of the results we came to from the 8
6.1%

#We could not replicate the results, this may be due to using different data
#or becuase we used a different method which made more logical sense to us.

B <- ggplot(fullData, aes(x =civilian.dead, y = Total.Civilian.Conf))

B + geom_point(aes(color = civilian.wounded)) +
  ggtitle("Civilian Lethality")
```



```
#Graph Navy.Lethality
fullData$Navy.lethality <- (fullData$navy.dead)/(fullData$Total.Navy.Conf)
n <- ggplot(fullData)
n <- n + geom_bar(mapping = aes(Navy.lethality), fill = "blue")
n
```

```
## Warning: Removed 3801 rows containing non-finite values (stat_count).
```



```
#Graph Fed.Police.Lethality
fullData$Federal.Police.lethality <- (fullData$federal.police.dead)/(fullData$Total.Fede
ral.Police.Conf)
fp <- ggplot(fullData)
fp <- fp + geom_bar(mapping = aes(Federal.Police.lethality), fill = "blue")
fp
```

```
## Warning: Removed 3709 rows containing non-finite values (stat_count).
```