US Gun Violence and Census Data from California 2015

Assignment 3, MATH2349, Semester 1, 2018

Alistair Grevis-James (s3644119) and Christopher Kaias (s3090801)

Executive Summary

The required libraries were imported into the RStudio interactive development environment, followed by the .csv files “gun-violence-data” and “American Census Data 2015” as `gvd` and `acs2015`, respectively [1][2]. These data sets were then checked for tidiness (`gvd` untidy, `acs2015` tidy), dimensions (`gvd` 239677 obs. of 29 variables, `acs2015` 3220 obs. of 37 variables), NA count and attribute classes (see summary tables below). The `gvd` dataset failed to conform to the “tidy data” principles. Most notably, the attributes prefixed “participant_” contained multiple values per cell, delimited by a combination of [:digit:], :: and/or ||. In order to be computationally efficient, it was decided to first subset `gvd` before cleaning the aforementioned “participant_” attributes. As the `acs2015` dataset contained values from only the 2015 American (US) Census, the first subset involved filtering values of the year 2015 from the `gvd` dataset. This was achieved using the `as.logical()` and `year()` functions, creating and applying a boolean vector to `gvd` to create the dataset `gvd2015`. This methodology enabled us to retain the native date format YYYY-MM-DD. The `gvd2015` was then filtered for values which satisfied the condition `state == “California”`, giving `gvd2015_keep_cali`. A custom function was written and employed to scour through the `participant_age` column of `gvd2015`. This resulted in capturing age counts of incident participants by binning them into approximately 5-years bin intervals. The `participant, _gender, _status, _type` columns were then also all stripped of string prefix values and binned. The resultant `killed_count` and `injured_count` matched perfectly the original `n_killed` and `n_injured` attributes, indicating an extremely precise processing methodology. It was determined that the `incident_url` column of `gvd2015_keep_cali` could be dropped, as the URL prefix in every case matched the unique incident ID. The `acs2015` dataset was then filtered for values from the state of California.

Required packages

```
#The following packages were used whilst completing the report (with annotation)
library(readr) # Used to import .csv files
library(dplyr) # Section Y
library(stringr) # Section Z
library(knitr) # Section X
library(tidyr)
library(lubridate)
library(kableExtra)
library(outliers)
library(ggplot2)
#xlsx #readxl #foreign #gdata #rvest #dplyr  #deductive #validate #Hmisc
#MVN #infotheo #MASS #caret #MLR #ggplot #base R functions
```

Importing the Data

The first dataset, ‘Gun Violence Data - A Comprehensive Record of Over 260k US Gun Violence Incidents from 2013-2018’, was compiled by James Ko and made available on kaggle.com. This dataset contains all recorded gun violence incidents in the US between January 2013 and March 2018 (inclusive).
The second dataset, ‘US Census Demographic Data Demographic and Economic Data for Tracts and Counties’, was collected by the US Census Bureau and made available on kaggle.com. This dataset contains all census data from the 2015 Federal US census.
The `readr` function was used to import the data as shown below.

```
gvdZip <- "gun-violence-data.csv.zip"
outDir<-"/Users/alistairgj/Documents/GitHub" # Change output directory for unzipped .csv file as needed
unzip(gvdZip, exdir=outDir)
setwd("/Users/alistairgj/Documents/GitHub")
gvd <- read.csv("gun-violence-data.csv") # Importing Gun Violence Data
setwd("/Users/alistairgj/Documents/GitHub/MATH2349_Assignment3V2")
acs2015 <- read.csv("acs2015_county_data.csv") # Importing American (US) Census Data 2015
```

Checking the Data

Two custom output table was created to check `gvd` & `acs2015` . These results are summarized below.

Feature	gvd	acs2015
Observations	239677	3220
Attributes	29	37
Tidy	No	Yes
Format	Wide	Wide
Unique Ref	Yes - IncidentID	Yes - CensusID
Incorrect Class	Attribute 1, 5, 8:14, 16, 19	Attribute 1
Contains NA	Attribute 11, 15, 17, 18, 28, 29	Attribute 14, 15, 19

```
data.frame(AttNo = c(1:29), Attribute = names(gvd), Class = sapply(gvd, class),
           gvd_incident486623 = sapply(gvd[84,], function(x) paste0(head(x),  collapse = ", ")),
           NA_Count = sapply(gvd, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),  font_size = 11) %>% scroll_box(height = "200px")
```

AttNo	Attribute	Class	gvd_incident486623	NA_Count
1	incident_id	integer	486623	0
2	date	factor	2013-05-23	0
3	state	factor	Tennessee	0
4	city_or_county	factor	Bean Station	0
5	address	factor	1034 Main Street	0
6	n_killed	integer	2	0

```
data.frame(AttNo = c(1:37), Attribute = names(acs2015), Class = sapply(acs2015, class),
           acs2015 = sapply(acs2015, function(x) paste(head(x),  collapse = ", ")),
           NA_Count = sapply(acs2015, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),
                                                           font_size = 11) %>% scroll_box(height = "200px")
```

AttNo	Attribute	Class	acs2015	NA_Count
1	CensusId	integer	1001, 1003, 1005, 1007, 1009, 1011	0
2	State	factor	Alabama, Alabama, Alabama, Alabama, Alabama, Alabama	0
3	County	factor	Autauga, Baldwin, Barbour, Bibb, Blount, Bullock	0
4	TotalPop	integer	55221, 195121, 26932, 22604, 57710, 10678	0
5	Men	integer	26745, 95314, 14497, 12073, 28512, 5660	0
6	Women	integer	28476, 99807, 12435, 10531, 29198, 5018	0

The `length()` & `unique()` functions were used to determine if `gvd$incident_id` and `acs2015$CensusId` contained only unique values. In each case the number of values was equal to the number of observations. Thus `incident_id` and `CensusId` may be used as observation identifiers if required.

```
length(unique(gvd$incident_id)) # Checking the number of unique values for incident_id
```

```
## [1] 239677
```

```
length(unique(acs2015$CensusId)) # Checking the number of unique values for CensusId
```

```
## [1] 3220
```

Based on the output of the above `gvd` summary table (`data.frame(...)`), the `participant_age` and `participant_gender` were explored to better understand their values. It was noted that these cells store multiple, discrete instances of data. The [:digit:] at the start of each value is presumably used to track the values accross from one `participant_` attribute to the next. The first observation below contains only one age value, with four participant values. One avenue for addressing these inconsistencies could involve performing counts for each delimited value. However this is beyond the scope of the current analysis and so this particulat inconsistency will be ignored.

```
gvd %>% select(participant_age, participant_gender) %>% head(n=8) %>%
  kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"),  font_size = 11)
```

participant_age	participant_gender
0::20	0::Male||1::Male||3::Male||4::Female
0::20	0::Male
0::25||1::31||2::33||3::34||4::33	0::Male||1::Male||2::Male||3::Male||4::Male
0::29||1::33||2::56||3::33	0::Female||1::Male||2::Male||3::Male
0::18||1::46||2::14||3::47	0::Female||1::Male||2::Male||3::Female
0::23||1::23||2::33||3::55	0::Female||1::Female||2::Female||3::Female||4::Male||5::Male
0::51||1::40||2::9||3::5||4::2||5::15	0::Male||1::Female||2::Male||3::Female||4::Female||5::Male
	0::Male||1::Male||2::Male||3::Male||4::Male

Filtering GVD for Year 2015

The Gun Violence Dataset contains data from 2013 till 2015, while the American Census Data is from 2015. We will therefore create a data subset of the Gun Violence Data from the year 2015 only (as we will later be joining these datasets). The attribute `gvd$date` was converted to date format (YYYY-MM-DD) using `as.Date` and checked with `str()`, below. A Boolean vector was then created for 2015-MM-DD values. This was applied as a mask to `gvd`, and extracted as the subset `gvd2015`, which was then checked.

```
gvd$date <- as.Date(gvd$date) # Converting the gvd$date column and applying to the gvd dataset
str(gvd$date) # Checking the conversion of gvd$date
```

```
##  Date[1:239677], format: "2013-01-01" "2013-01-01" "2013-01-01" "2013-01-05" "2013-01-07" ...
```

```
year2015_boolean <- as.logical(year(gvd$date) == "2015") #Creating a year 2015 true/false vector
summary(year2015_boolean) # Checking the vector results
```

```
##    Mode   FALSE    TRUE
## logical  186098   53579
```

```
gvd2015 <- gvd[year2015_boolean,] # Applying the True/False Vector to filter gvd
```

```
summary(gvd2015$date) # Confirming only 2015 data is present
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.
## "2015-01-01" "2015-04-14" "2015-07-09" "2015-07-06" "2015-09-29"
##         Max.
## "2015-12-31"
```

The DS `gvd2015` is now prepared. As an aside, the first methodology used for extracting YYYY = 2015 value from `gvd$date` relied on conversion of `gvd$date` to contain year only (`gvd$date <- format.Date(gvd$date, format="%Y")`), this was then filtered for values of 2015 (`gvd$date_year <- filter(gvd, date_year == "2015")`). This methodolgy was abandoned, as it unnecessarily stripped out the month and day data from `gvd$date`. The applied method is no more complex but retains more information.

Filtering for California

The values of `gvd2015$state` were aggregated for comparison (below). The output enables us to determine the number of `city_or_county` in each State, as defined during the composition of the Gun Violence Dataset. For the subsequent analysis, the State of California was selected. This was based on an assumption it would have a large range of discrete and continuous values in the corresponding Census dataset (for example both extremely poor and rich socioeconomic areas). Also note that the methodology outlined below could be applied to any of the states in the `gvd` dataset, so California will act as a proof of concept.

```
aggregate(city_or_county ~ state, gvd2015, function(x) length(unique(x))) %>%
  kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"),  font_size = 11) %>%
  row_spec(5, bold = T, background = "lightskyblue") %>%
  scroll_box(height = "200px")
```

state	city_or_county
Alabama	212
Alaska	63
Arizona	67
Arkansas	116
California	**502**
Colorado	105

```
gvd2015_keep_cali <- filter(gvd2015, state == "California") # Creating the new dataset
```

```
aggregate(city_or_county ~ state, gvd2015_keep_cali, function(x) length(unique(x))) %>%
  kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"),  font_size = 11) %>%
  row_spec(1, bold = T, background = "lightskyblue") # Check the new dataset
```

state	city_or_county
California	502

GVD 2015 California DS

During data exploration it was determined the following attributes will add no futher value to this analysis and will be dropped.

Attribute	Explanation / Justification
address	This level of extremely fine detail is not required
source_url	This can be found online the other URL incident_url
incident_url_fields_missing	This is a boolean attribute which can be checked with filtering
incident_characteristics	These strings are extremely varied in content
location_description	These strings are extremely varied in content
notes	These strings are extremely varied in content & contain no information of value for us
participant_age_group	We can determine this from participant_age
participant_name	This level of extremely fine detail is not required
sources	We have source information from incident_url

The desired columns were selected back into gvd2015_keep_cali, which was then checked in the table below. The attributes congressional_district, state_house_district & state_sentate_district were converted into factors and the dimensions of the new gvd2015_keep_cali were checked.

```
gvd2015_keep_cali <- select(gvd2015_keep_cali, incident_id, date, state, city_or_county, n_killed, n_injured, inc
ident_url, congressional_district, latitude, longitude, n_guns_involved, participant_age, participant_gender, par
ticipant_status, participant_type, state_house_district, state_senate_district)
```

```
data.frame(Attribute = names(gvd2015_keep_cali), Class = sapply(gvd2015_keep_cali, class),
           gvd_incident274168 = sapply(gvd2015_keep_cali[53,], function(x) paste0(head(x),  collapse = ", ")),
           NA_Count = sapply(gvd2015_keep_cali, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),
                                                           font_size = 11) %>% scroll_box(height = "200px")
```

Attribute	Class	gvd_incident274168	NA_Count
incident_id	integer	274386	0
date	Date	2015-01-05	0
state	factor	California	0
city_or_county	factor	Selma	0
n_killed	integer	3	0
n_injured	integer	0	0

```
gvd2015_keep_cali$congressional_district <- as.factor(gvd2015_keep_cali$congressional_district)
gvd2015_keep_cali$state_house_district <- as.factor(gvd2015_keep_cali$state_house_district)
gvd2015_keep_cali$state_senate_district <- as.factor(gvd2015_keep_cali$state_senate_district)
```

```
dim(gvd2015_keep_cali)
```

```
## [1] 3234   17
```

Incident_URL Attribute

The attribute incident_url was compared with incident_id. Using string manipulation, the suffix http://www.gunviolencearchive.org/incident/ was removed from all values in the incident_url. The resultant _url column and the original _id column were converted into vectors (_id = a, _url = b). The vectors a and b were compared, with a delta value of 0. The _url attribute is therefore deemed unnecessary and will be dropped.

```
gvd2015_keep_cali$incident_url <-  gvd2015_keep_cali$incident_url %>%
  str_replace_all("http://www.gunviolencearchive.org/incident/", "") # Cleaving off the suffix
gvd2015_keep_cali$incident_url <- as.integer(gvd2015_keep_cali$incident_url)
```

```
# Create two vectors
a <- as.vector(gvd2015_keep_cali$incident_id)
b <- as.vector(gvd2015_keep_cali$incident_url)
a[!(a %in% b)]
```

```
## integer(0)
```

n_gun_involved Attribute

This attribute contained approximately 60% NA values (Var1(1) = NA) and was dropped.

```
table(sapply(gvd2015_keep_cali$n_guns_involved, function(y) sum(length(which(is.na(y)))))) %>% kable() %>% kable_
styling(bootstrap_options = c("striped", "hover", "condensed"),  font_size = 11)
```

Var1	Freq
0	1258
1	1976

participant_age Attribute

The following function was constructed to iterate through the integer values of the participant_age column and bin the values using a +1 count. The resultant binned counts were spread over 15 new attributes.

```
gvd2015_keep_cali$matches <- str_match_all(gvd2015_keep_cali$participant_age, "\\d+::(\\d+)")
ages = c(c(0,15)) # First age range is under 15s
for (i in seq(16, 76, 5)) # The other age ranges are every 5 years
{
  ages <- c(ages, c(i, i+4))
}
ages <- c(ages, c(81,100)) # The last age range is 81 plus
for (i in seq(1,length(ages)-1, 2)) # Each pair of ages in the age ranges
{
  # Construct the column using apply
  # Paste here concatenates the string for the column name
  gvd2015_keep_cali[paste("age", ages[i],ages[i+1], sep="_")] <- apply(gvd2015_keep_cali, 1, function(row) {
    cnt <- 0 # Count of ages in current range starts at 0
    for (x in row$matches) # Iterate through the matches we parsed out
    { if (!grepl(":", x, fixed = TRUE)) # If the match does not contain a : then it is the match group rather tha
n the match
      { age <- as.integer(x) # so convert age as string to integer
          if (age >= ages[i] && age <= ages[i+1]) # and check if age falls within age range for the current colum
n
        { cnt <- cnt + 1 }
      }
    }
    as.integer(cnt)
  })
}
```

participant_gender Attribute

The characters `,[:space:]`, `::`, `[:digit:]` & `||` were stripped from the `participant_gender` as seen below. A new char value was subsequently created, `participant_gender_unique_check`. This value was then checked for unique characters, which yielded only blank (“”), Female and Male. Two `str_count` functions were then used to generate count columns for gender, these were named `male_count` and `female_count`. The column `participant_gender` will be later dropped.

```
gvd2015_keep_cali$participant_gender <-  gvd2015_keep_cali$participant_gender %>%
  str_replace_all(",[:space:]", "") %>% #Remove comma & whitespace
  str_replace_all("::", "") %>%
  str_replace_all("[:digit:]", " ") %>%
  str_replace_all("\\||", "")
```

```
participant_gender_unique_check <- as.character(gvd2015_keep_cali[,c("participant_gender")])
ul <- unlist(strsplit(participant_gender_unique_check, " "))
unique(ul)
```

```
## [1] ""       "Male"   "Female"
```

```
gvd2015_keep_cali$male_count <- str_count(gvd2015_keep_cali$participant_gender, pattern = "Male") # Stringr
gvd2015_keep_cali$female_count <- str_count(gvd2015_keep_cali$participant_gender, pattern = "Female") # Stringr
```

Participant_status Attribute

The characters `,[:space:]`, `::`, `[:digit:]` & `||` were stripped from the `participant_status` in the same manner as for `participant_gender`. The same methodology was then used to generate counts (`unlist`, `strsplit`, `unique`) for the `participant_status` values, giving the following:

```
participant_status_unique_check <- as.character(gvd2015_keep_cali[,c("participant_status")])
ul <- unlist(strsplit(participant_status_unique_check, " "))
unique(ul)
```

```
##  [1] ""                          "Killed"
##  [3] "Injured"                   "UnharmedArrested"
##  [5] "InjuredUnharmedArrested" "Arrested"
##  [7] "Unharmed"                  "InjuredArrested"
##  [9] "KilledArrested"            "KilledUnharmed"
## [11] "KilledUnharmedArrested"
```

Several anomalous values which contain mutually exclusive states of being, were identified as; “InjuredUnharmedArrested”, “KilledInjured”, “KilledArrested”, “KilledUnharmed”, “InjuredUnharmed” and “KilledUnharmedArrested”. The `grepl` function was then used to create a boolean vector for in each case to search for the source of the value in `gvd2015_keep_cali`. Once the `incident_id` was identified, the URL “http://www.gunviolencearchive.org/incident/`incident_id (http://www.gunviolencearchive.org/incident/%60incident_id) `" was checked. Each instance was then rationalized and given a more appropriate v` function.

```
InjuredUnharmedArrested <- grepl("InjuredUnharmedArrested", gvd2015_keep_cali$participant_status)
gvd2015_keep_cali[InjuredUnharmedArrested,c(1,14)]
```

	incident_id	participant_status
	<int>	<chr>
7	273316	Killed UnharmedArrested UnharmedArrested InjuredUnharmedArrested

1 row

The values were all checked as described above and updated as follows:

OriginalValue	Explanation / Justification	UpdatedValue
`InjuredUnharmedArrested`	This value was created as the reporting of an incident is updated	`InjuredArrested`
`KilledInjured`	This value was created when a person who was initially injured later dies	`Killed`
`KilledArrested`	This value was created when a person who was mortally wounded, arrested and later dies	`Killed`
`KilledUnharmed`	This value was created as the reporting of an incident is updated	`Killed`
`InjuredUnharmed`	This value was created as the reporting of an incident is updated	`Injured`
`KilledUnharmedArrested`	This value was created as the reporting of an incident is updated	`Killed`

```
gvd2015_keep_cali$participant_status <- str_replace_all(gvd2015_keep_cali$participant_status, "InjuredUnharmedArr
ested", "InjuredArrested")
gvd2015_keep_cali$participant_status <- str_replace_all(gvd2015_keep_cali$participant_status, "KilledInjured", "K
illed")
gvd2015_keep_cali$participant_status  <-  str_replace_all(gvd2015_keep_cali$participant_status,  "KilledArrested",
"Killed")
gvd2015_keep_cali$participant_status  <-  str_replace_all(gvd2015_keep_cali$participant_status,  "KilledUnharmed",
"Killed")
gvd2015_keep_cali$participant_status  <-  str_replace_all(gvd2015_keep_cali$participant_status,  "InjuredUnharmed",
"Injured")
gvd2015_keep_cali$participant_status <- str_replace_all(gvd2015_keep_cali$participant_status, "KilledUnharmedArre
sted", "Killed")
```

```
participant_status_unique_check <- as.character(gvd2015_keep_cali[,c("participant_status")])
ul <- unlist(strsplit(participant_status_unique_check, " "))
unique(ul)
```

```
## [1] ""                 "Killed"           "Injured"
## [4] "UnharmedArrested" "InjuredArrested"  "Arrested"
## [7] "Unharmed"
```

```
gvd2015_keep_cali$injured_count <- str_count(gvd2015_keep_cali$participant_status, pattern = "Injured")
gvd2015_keep_cali$unharmed_count <- str_count(gvd2015_keep_cali$participant_status, pattern = "Unharmed")
gvd2015_keep_cali$killed_count <- str_count(gvd2015_keep_cali$participant_status, pattern = "Killed")
gvd2015_keep_cali$unharmedArrested_count <- str_count(gvd2015_keep_cali$participant_status, pattern = "UnharmedAr
rested")
gvd2015_keep_cali$arrested_count <- str_count(gvd2015_keep_cali$participant_status, pattern = "Arrested")
gvd2015_keep_cali$injuredArrested_count <- str_count(gvd2015_keep_cali$participant_status, pattern = "InjuredArre
sted")
```

Participant_Type Attribute

The characters `,[:space:]`, `::`, `[:digit:]` & `||` were stripped from the `participant_type` in the same manner as for `participant_gender`. The same methodology was then used to generate counts (`unlist`, `strsplit`, `unique`) for the `participant_type` values, giving the following;

```
participant_type_unique_check <- as.character(gvd2015_keep_cali[,c("participant_type")])
ul <- unlist(strsplit(participant_type_unique_check, " "))
unique(ul)
```

```
## [1] ""                "Victim"          "Subject-Suspect"
```

```
gvd2015_keep_cali$victim_count <- str_count(gvd2015_keep_cali$participant_type, pattern = "Victim")
gvd2015_keep_cali$subjectSuspect_count <- str_count(gvd2015_keep_cali$participant_type, pattern = "Subject-Suspec
t")
```

The n_killed & n_injured Attributes

The native `n_killed` attribute was compared with the generated `killed_count` attribute - these columns were found to be equal. The same was true for the native `n_injured` and generated `injured_count`. The `n_killed` and `n_injured` columns were both dropped.

```
a <- as.vector(gvd2015_keep_cali$n_killed) # Creating a vector
b <- as.vector(gvd2015_keep_cali$killed_count) # Creating a vector
a[!(a %in% b)] # Comparing (differential)
```

```
## integer(0)
```

```
a <- as.vector(gvd2015_keep_cali$n_injured) # Creating a vector
b <- as.vector(gvd2015_keep_cali$injured_count) # Creating a vector
a[!(a %in% b)] # Comparing (differential)
```

```
## integer(0)
```

Checking and Subsetting

The `gvd2015_keep_cali` dataset was checked for attribute names and filtered for the the desired attributes to create `gvd2015_pt2`.

```
colnames(gvd2015_keep_cali)
```

```
##  [1] "incident_id"            "date"
##  [3] "state"                  "city_or_county"
##  [5] "n_killed"               "n_injured"
##  [7] "incident_url"           "congressional_district"
##  [9] "latitude"               "longitude"
## [11] "n_guns_involved"        "participant_age"
## [13] "participant_gender"     "participant_status"
## [15] "participant_type"       "state_house_district"
## [17] "state_senate_district"  "matches"
## [19] "age_0_15"               "age_16_20"
## [21] "age_21_25"              "age_26_30"
## [23] "age_31_35"              "age_36_40"
## [25] "age_41_45"              "age_46_50"
## [27] "age_51_55"              "age_56_60"
## [29] "age_61_65"              "age_66_70"
## [31] "age_71_75"              "age_76_80"
## [33] "age_81_100"             "male_count"
## [35] "female_count"           "injured_count"
## [37] "unharmed_count"         "killed_count"
## [39] "unharmedArrested_count" "arrested_count"
## [41] "injuredArrested_count"  "victim_count"
## [43] "subjectSuspect_count"
```

```
gvd2015_pt2 <- select(gvd2015_keep_cali, incident_id, date, state, city_or_county, congressional_district, state_
house_district, state_senate_district, latitude, longitude, age_0_15, age_16_20, age_21_25, age_26_30, age_31_35,
age_36_40, age_41_45, age_46_50, age_51_55, age_56_60, age_61_65, age_66_70, age_71_75, age_76_80, age_81_100, ma
le_count, female_count, injured_count, unharmed_count, killed_count, unharmedArrested_count, arrested_count, inju
redArrested_count, victim_count, subjectSuspect_count)
```

```
data.frame(AttNo = c(1:34), Attribute = names(gvd2015_pt2), Class = sapply(gvd2015_pt2, class),
           gvd2015_pt2 = sapply(gvd2015_pt2, function(x) paste(head(x),  collapse = ", ")),
           NA_Count = sapply(gvd2015_pt2, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),  font_size = 11)
```

AttNo	Attribute	Class	gvd2015_pt2	NA_Count
1	incident_id	integer	271979, 272551, 272555, 271970, 271975, 275619	0
2	date	Date	2015-01-01, 2015-01-01, 2015-01-01, 2015-01-01, 2015-01-01, 2015-01-01	0
3	state	factor	California, California, California, California, California, California	0
4	city_or_county	factor	Grand Terrace, Merced, Seaside, Downey, San Jose, Bermuda Dunes	0
5	congressional_district	factor	31, 16, 20, 40, 19, 36	23
6	state_house_district	factor	47, 21, 29, 58, 27, 56	24
7	state_senate_district	factor	20, 12, 17, 32, 15, 28	23
8	latitude	numeric	34.0322, 37.3006, 36.6121, 33.9327, 37.3339, 33.7429	19
9	longitude	numeric	-117.326, -120.498, -121.836, -118.102, -121.855, -116.274	19
10	age_0_15	integer	0, 0, 0, 0, 0, 0	0
11	age_16_20	integer	0, 0, 0, 0, 0, 0	0
12	age_21_25	integer	1, 0, 0, 0, 0, 0	0
13	age_26_30	integer	1, 0, 0, 0, 0, 0	0
14	age_31_35	integer	0, 0, 0, 0, 0, 0	0
15	age_36_40	integer	0, 0, 0, 0, 0, 0	0
16	age_41_45	integer	0, 0, 0, 0, 0, 0	0

AttNo	Attribute	Class	gvd2015_pt2	NA_Count
17	age_46_50	integer	1, 0, 0, 0, 0, 0	0
18	age_51_55	integer	0, 1, 0, 0, 0, 0	0
19	age_56_60	integer	0, 0, 0, 0, 0, 0	0
20	age_61_65	integer	0, 0, 0, 0, 0, 0	0
21	age_66_70	integer	0, 0, 0, 0, 0, 0	0
22	age_71_75	integer	0, 0, 0, 0, 0, 2	0
23	age_76_80	integer	0, 0, 0, 0, 0, 0	0
24	age_81_100	integer	0, 0, 0, 0, 0, 0	0
25	male_count	integer	2, 1, 0, 0, 2, 1	0
26	female_count	integer	0, 0, 0, 0, 0, 1	0
27	injured_count	integer	2, 0, 0, 0, 2, 1	0
28	unharmed_count	integer	0, 0, 0, 0, 0, 0	0
29	killed_count	integer	1, 1, 0, 0, 0, 1	0
30	unharmedArrested_count	integer	0, 0, 0, 0, 0, 0	0
31	arrested_count	integer	0, 0, 0, 0, 0, 0	0
32	injuredArrested_count	integer	0, 0, 0, 0, 0, 0	0
33	victim_count	integer	3, 1, 0, 0, 2, 1	0
34	subjectSuspect_count	integer	0, 0, 0, 0, 0, 1	0

American Census Data

The previously imported American (US) Census Data was in the table summarized below.

```
data.frame(AttNo = c(1:37), Attribute = names(acs2015), Class = sapply(acs2015, class),
           acs2015 = sapply(acs2015, function(x) paste(head(x),  collapse = ", ")),
           NA_Count = sapply(acs2015, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),  font_size = 11) #%>% scroll_box(height = "200px")
```

AttNo	Attribute	Class	acs2015	NA_Count
1	CensusId	integer	1001, 1003, 1005, 1007, 1009, 1011	0
2	State	factor	Alabama, Alabama, Alabama, Alabama, Alabama, Alabama	0
3	County	factor	Autauga, Baldwin, Barbour, Bibb, Blount, Bullock	0
4	TotalPop	integer	55221, 195121, 26932, 22604, 57710, 10678	0
5	Men	integer	26745, 95314, 14497, 12073, 28512, 5660	0
6	Women	integer	28476, 99807, 12435, 10531, 29198, 5018	0
7	Hispanic	numeric	2.6, 4.5, 4.6, 2.2, 8.6, 4.4	0
8	White	numeric	75.8, 83.1, 46.2, 74.5, 87.9, 22.2	0
9	Black	numeric	18.5, 9.5, 46.7, 21.4, 1.5, 70.7	0
10	Native	numeric	0.4, 0.6, 0.2, 0.4, 0.3, 1.2	0
11	Asian	numeric	1, 0.7, 0.4, 0.1, 0.1, 0.2	0
12	Pacific	numeric	0, 0, 0, 0, 0, 0	0
13	Citizen	integer	40725, 147695, 20714, 17495, 42345, 8057	0
14	Income	numeric	51281, 50254, 32964, 38678, 45813, 31938	1
15	IncomeErr	numeric	2391, 1263, 2973, 3995, 3141, 5884	1
16	IncomePerCap	integer	24974, 27317, 16824, 18431, 20532, 17580	0
17	IncomePerCapErr	integer	1080, 711, 798, 1618, 708, 2055	0
18	Poverty	numeric	12.9, 13.4, 26.7, 16.8, 16.7, 24.6	0
19	ChildPoverty	numeric	18.6, 19.2, 45.3, 27.9, 27.2, 38.4	1
20	Professional	numeric	33.2, 33.1, 26.8, 21.5, 28.5, 18.8	0
21	Service	numeric	17, 17.7, 16.1, 17.9, 14.1, 15	0
22	Office	numeric	24.2, 27.1, 23.1, 17.8, 23.9, 19.7	0
23	Construction	numeric	8.6, 10.8, 10.8, 19, 13.5, 20.1	0
24	Production	numeric	17.1, 11.2, 23.1, 23.7, 19.9, 26.4	0
25	Drive	numeric	87.5, 84.7, 83.8, 83.2, 84.9, 74.9	0
26	Carpool	numeric	8.8, 8.8, 10.9, 13.5, 11.2, 14.9	0
27	Transit	numeric	0.1, 0.1, 0.4, 0.5, 0.4, 0.7	0
28	Walk	numeric	0.5, 1, 1.8, 0.6, 0.9, 5	0
29	OtherTransp	numeric	1.3, 1.4, 1.5, 1.5, 0.4, 1.7	0
30	WorkAtHome	numeric	1.8, 3.9, 1.6, 0.7, 2.3, 2.8	0
31	MeanCommute	numeric	26.5, 26.4, 24.1, 28.8, 34.9, 27.5	0
32	Employed	integer	23986, 85953, 8597, 8294, 22189, 3865	0

AttNo	Attribute	Class	acs2015	NA_Count
33	PrivateWork	numeric	73.6, 81.5, 71.8, 76.8, 82, 79.5	0
34	PublicWork	numeric	20.9, 12.3, 20.8, 16.1, 13.5, 15.1	0
35	SelfEmployed	numeric	5.5, 5.8, 7.3, 6.7, 4.2, 5.4	0
36	FamilyWork	numeric	0, 0.4, 0.1, 0.4, 0.4, 0	0
37	Unemployment	numeric	7.6, 7.5, 17.6, 8.3, 7.7, 18	0

Subsetting Census Data

For the following analysis, only economic and population data were carried over. The `acs2015` dataset was subset into `acs2015_keep` , below. The subset data contained only 1 NA value. As the GVD dataset was filtered for the State of California, so to will be the `acs2015_keep` dataset. The California county count was noted to be 58 (as opposed to 502 for GVD). This was retionalized by the data collection process, which was initially generated in police reports. The classification system for county areas used by law enforcement is highly unlikely to match what is used to collect census data. We can consider county classification in the GVD to perhaps be more generalized, whereas in the American Census Data classification to be more empirical.

```
acs2015_keep <- select(acs2015, State, County, TotalPop, Men, Women, Income, IncomePerCap, Poverty, Unemployment)
```

```
data.frame(AttNo = c(1:9), Attribute = names(acs2015_keep), Class = sapply(acs2015_keep, class),
           acs2015_keep = sapply(acs2015_keep, function(x) paste(head(x),  collapse = ", ")),
           NA_Count = sapply(acs2015_keep, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),  font_size = 11) #%>% scroll_box(height = "200px")
```

AttNo	Attribute	Class	acs2015_keep	NA_Count
1	State	factor	Alabama, Alabama, Alabama, Alabama, Alabama, Alabama	0
2	County	factor	Autauga, Baldwin, Barbour, Bibb, Blount, Bullock	0
3	TotalPop	integer	55221, 195121, 26932, 22604, 57710, 10678	0
4	Men	integer	26745, 95314, 14497, 12073, 28512, 5660	0
5	Women	integer	28476, 99807, 12435, 10531, 29198, 5018	0
6	Income	numeric	51281, 50254, 32964, 38678, 45813, 31938	1
7	IncomePerCap	integer	24974, 27317, 16824, 18431, 20532, 17580	0
8	Poverty	numeric	12.9, 13.4, 26.7, 16.8, 16.7, 24.6	0
9	Unemployment	numeric	7.6, 7.5, 17.6, 8.3, 7.7, 18	0

```
aggregate(County ~ State, acs2015_keep, function(x) length(unique(x))) %>% kable() %>% kable_styling(bootstrap_op
tions = c("striped", "hover", "condensed"),  font_size = 11) %>% row_spec(5, bold = T, background = "lightskyblu
e") %>% scroll_box(height = "200px")
```

State	County
Alabama	67
Alaska	29
Arizona	15
Arkansas	75
California	**58**
Colorado	64

```
acs2015_keep_cali <- filter(acs2015_keep, State == "California") # Subsetting acs for California
```

```
aggregate(County ~ State, acs2015_keep_cali, function(x) length(unique(x))) %>% kable() %>% kable_styling(bootstr
ap_options = c("striped", "hover", "condensed"),  font_size = 11) %>% row_spec(1, bold = T, background = "lightsk
yblue")
```

State	County
California	**58**

Joining the Data Sets

The `gvd2015_pt2` and `acs2015_keep_cali` datasets were joined by County. As an inner join was used, this will retain the County data found only in both sets.

```
gvd2015_pt2 <- rename(gvd2015_pt2, County = city_or_county) # Renaming
gvd_acs_join <- inner_join(gvd2015_pt2, acs2015_keep_cali, by = "County")
gvd_acs_join <- subset(gvd_acs_join, select = -c(State, state)) # State attribute dropped (all values = Californi
a)
```

```
data.frame(Attribute = names(gvd_acs_join), Class = sapply(gvd_acs_join, class),
           gvd_acs_join = sapply(gvd_acs_join, function(x) paste0(head(x),  collapse = ", ")),
           NA_count = sapply(gvd_acs_join, function(y) sum(length(which(is.na(y))))),
            row.names = NULL) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"
),  font_size = 11) %>% scroll_box(height = "200px")
```

Attribute	Class	gvd_acs_join	NA_count
incident_id	integer	272551, 272526, 272165, 273497, 273249, 273495	0
date	Date	2015-01-01, 2015-01-01, 2015-01-01, 2015-01-01, 2015-01-02, 2015-01-03	0
County	character	Merced, Fresno, San Francisco, Merced, Fresno, Fresno	0
congressional_district	factor	16, 16, 12, 16, 16, 16	5
state_house_district	factor	21, 31, 17, 21, 23, 31	5
state_senate_district	factor	12, 14, 11, 12, 8, 14	5
latitude	numeric	37.8000, 38.7907, 37.7081, 37.8102, 38.7591, 38.7594	5

NA Values in Joined Dataset

The following table shows Counties with corresponding NA values. The number of unique values in all six of the attributes do not match (a match example would be four Counties to 4 congressional_district values). As these columns represent all the of geographical information for each instance, it was determined that the NA values could not be imputed or determined, and were omitted. This reduced our total number of instances by only five.

```
gvd_acs_join %>% filter(is.na(congressional_district)) %>%
  select(County, congressional_district, state_house_district, state_senate_district, latitude, longitude) %>%
  kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"),  font_size = 11)
```

County	congressional_district	state_house_district	state_senate_district	latitude	longitude
Fresno	NA	NA	NA	NA	NA
San Bernardino	NA	NA	NA	NA	NA
Fresno	NA	NA	NA	NA	NA
Los Angeles	NA	NA	NA	NA	NA
Sacramento	NA	NA	NA	NA	NA

```
target <- c("Fresno", "San Bernardino", "Los Angeles", "Sacramento")
gvd_select_county <- filter(gvd_acs_join, County %in% target)
gvd_select_county <- gvd_select_county %>% select(County, congressional_district, state_house_district, state_sen
ate_district, latitude, longitude)
UniqueCount <- lengths(lapply(gvd_select_county, unique))
as.data.frame(UniqueCount) %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
 font_size = 11)
```

	UniqueCount
County	4
congressional_district	23
state_house_district	26
state_senate_district	19
latitude	501
longitude	346

```
gvd_acs_join <- na.omit(gvd_acs_join)
```

County	congressional_district	state_house_district	state_senate_district	latitude	longitude

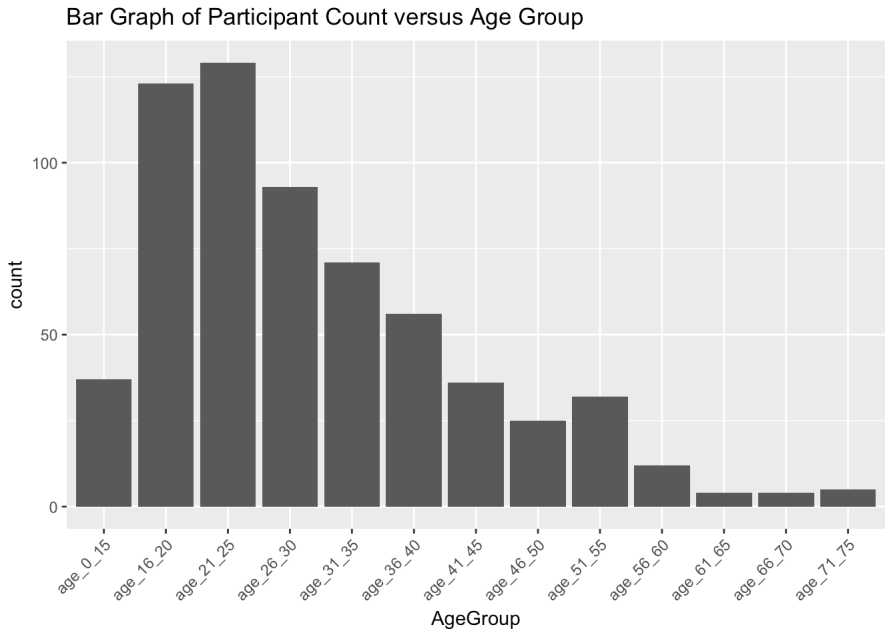
Wide to Long Format

The age_count section of the dataset was very sparse. For this reason the dataset was gathered accross the age bin values. This dataset was then filtered to remove AgeGroupCounts equal to zero.

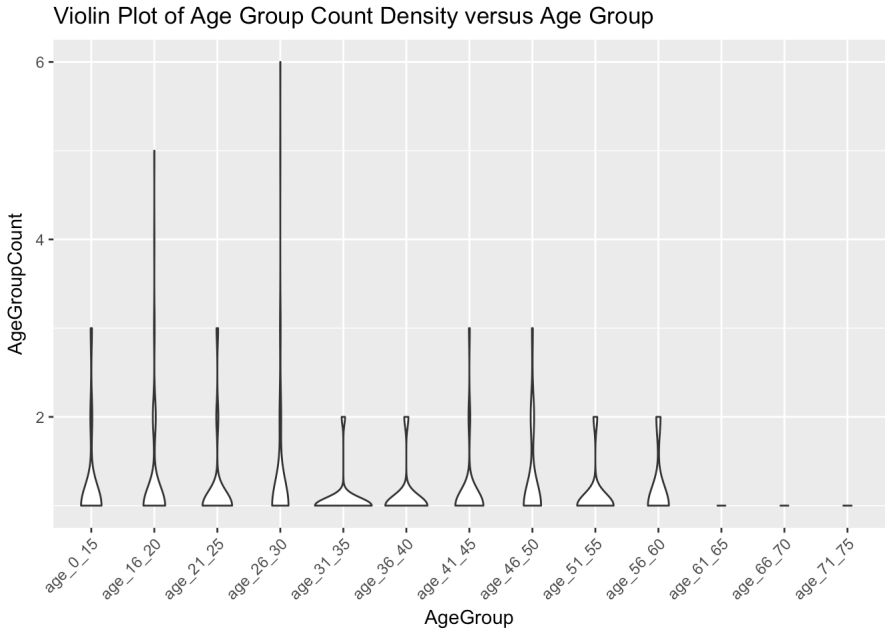
```
gvd_acs_join_gather <- gvd_acs_join %>% gather(age_0_15:age_81_100, key = "AgeGroup", value = "AgeGroupCount")
```

```
gvd_acs_age_long <- filter(gvd_acs_join_gather, AgeGroupCount != 0)
```

```
bar1 <- ggplot(gvd_acs_age_long, aes(x=AgeGroup)) +
  geom_bar() +
  ggtitle("Bar Graph of Participant Count versus Age Group") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
bar1
```




```
violin1 <- ggplot(gvd_acs_age_long, aes(x=AgeGroup, y=AgeGroupCount)) +
  geom_violin() +
  ggtitle("Violin Plot of Age Group Count Density versus Age Group") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
violin1
```

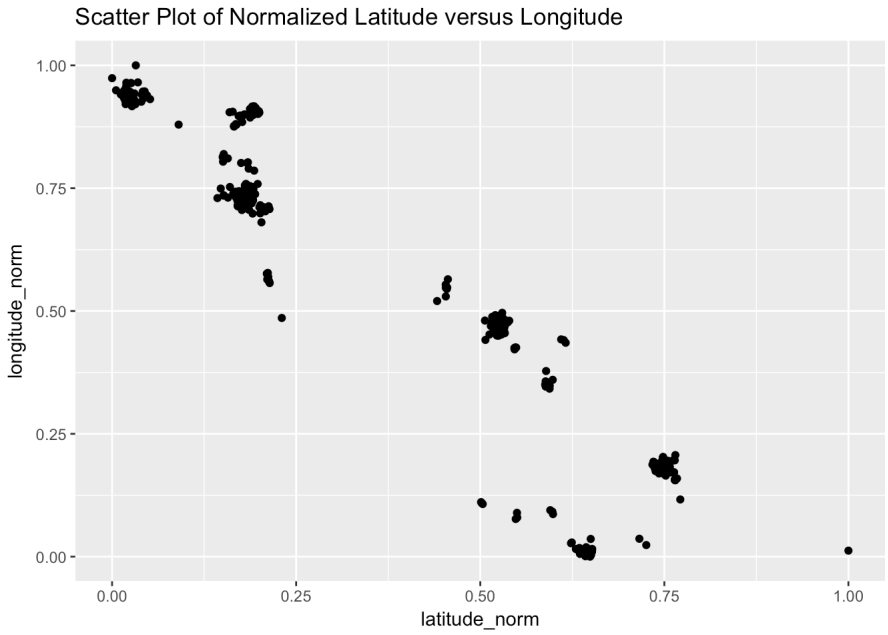


Data Transformation

The latitude and longitude attributes were normalized from 0 till 1 and plotted with an xy scatter.

```
range01 <- function(x){(x-min(x))/(max(x)-min(x))}
gvd_acs_join$latitude_norm <- range01(gvd_acs_join$latitude)
gvd_acs_join$longitude_norm <- range01(gvd_acs_join$longitude)
```

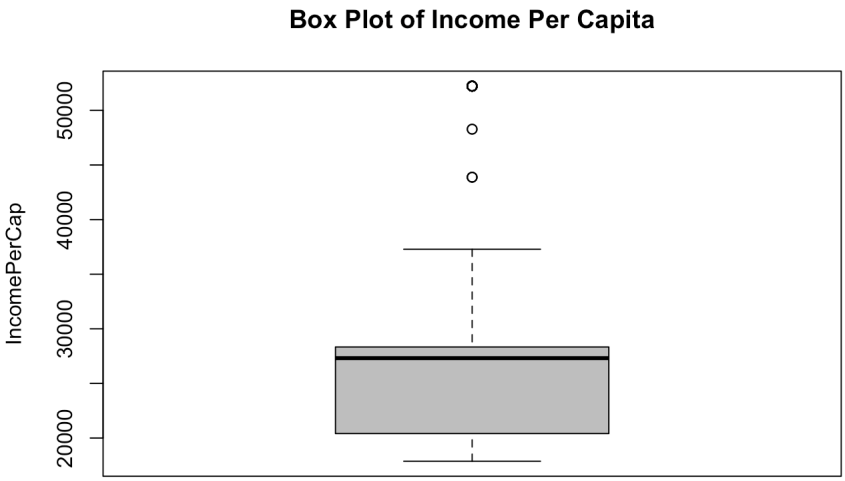
```
scatter3 <- ggplot(gvd_acs_join, aes(latitude_norm, longitude_norm)) +
  geom_point(aes()) +
  ggtitle("Scatter Plot of Normalized Latitude versus Longitude")
scatter3
```



Outliers

Outliers were detected using a boxplot of Income per capita.

```
gvd_acs_join$IncomePerCap %>%  boxplot(main="Box Plot of Income Per Capita", ylab="IncomePerCap", col = "grey")
```




```
# Checking the summary stats
summary(gvd_acs_join$IncomePerCap)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17876   20408   27315   27079   28337   52220
```

```
# Checking the z score summary stats
z.scores <- gvd_acs_join$IncomePerCap %>% scores(type = "z")
z.scores %>% summary()
```

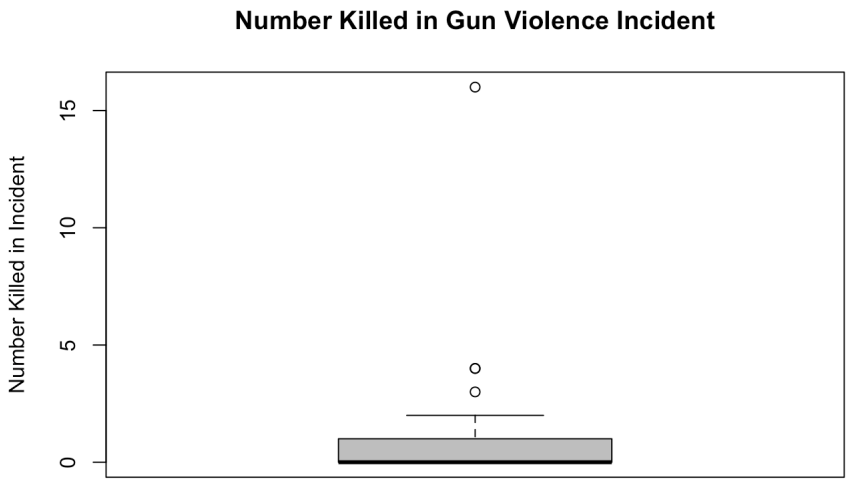
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.08393 -0.78571  0.02783  0.00000  0.14820  2.96123
```

```
# Checking the z score values
gvd_acs_join$IncomePerCap[which(abs(z.scores) > 3)]
```

```
## integer(0)
```

These values could not be classified as outliers according to the z-score method. Outliers were then checked for `gvd_acs_join$killed_count`. These outliers were also evaluated with the z-score method, and subsequently removed from the dataset.

```
gvd_acs_join$killed_count %>% boxplot(main="Number Killed in Gun Violence Incident", ylab="Number Killed in Incid
ent", col = "grey")
```




```
# Checking the summary stats
summary(gvd_acs_join$killed_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3438  1.0000 16.0000
```

```
# Checking the z score summary stats
z.scores <- gvd_acs_join$killed_count %>% scores(type = "z")
z.scores %>% summary()
```

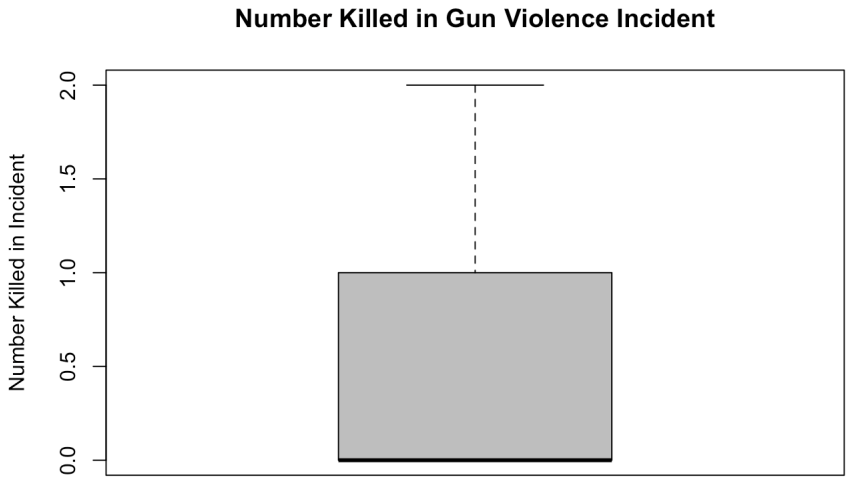
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.4558 -0.4558 -0.4558  0.0000  0.8698 20.7551
```

```
# Checking the z score values
gvd_acs_join$killed_count[which(abs(z.scores) > 3)]
```

```
## [1]  4  3 16  4
```

```
# Imputing the outliers
killed_count_clean <- gvd_acs_join$killed_count[ - which(abs(z.scores) > 3)]
```

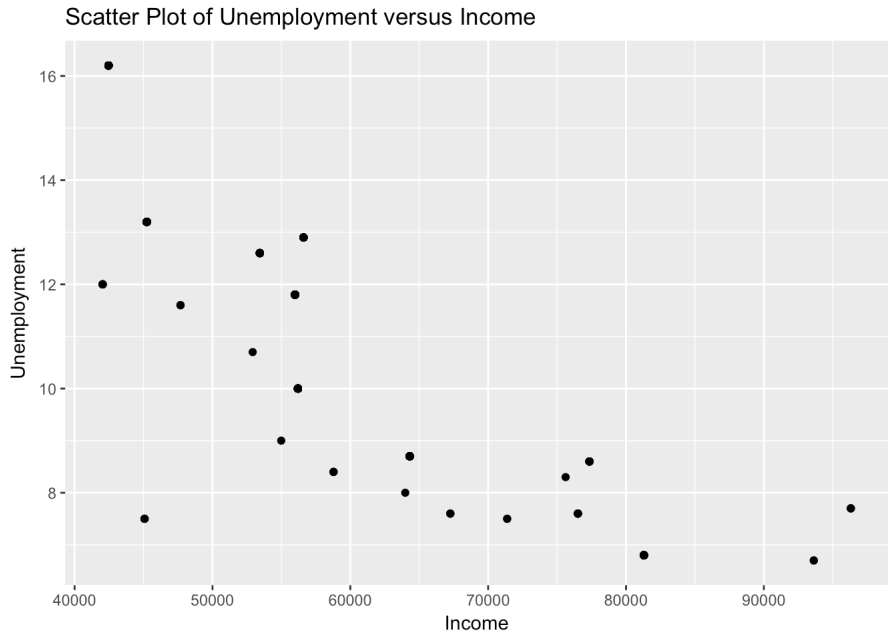
```
killed_count_clean %>% boxplot(main="Number Killed in Gun Violence Incident", ylab="Number Killed in Incident", c
ol = "grey")
```



Other

The following scatter plot shows the relationship between income and unemployment, however could not be transformed with log10 as it contains bivariate data.

```
scatter5 <- ggplot(gvd_acs_join, aes(Income, Unemployment)) +
  geom_point(aes()) +
  ggtitle("Scatter Plot of Unemployment versus Income")
scatter5
```



References

1. https://www.kaggle.com/jameslko/gun-violence-data/data (https://www.kaggle.com/jameslko/gun-violence-data/data)↩
2. https://www.kaggle.com/zimeiyang/2015-us-census-demographic-data/data (https://www.kaggle.com/zimeiyang/2015-us-census-demographic-data/data)↩