MATH2349 Data Wrangling

Assignment 2

Rinaldo Gagiano S3870806

Required packages

```
library(xlsx)
library(readxl)
library(readr)
library(dplyr)
library(Hmisc)
library(lubridate)
library(tidyr)
```

Executive Summary

This assignment involves the preprocessing of two main datasets prior to being merged. The first data set is imported. It has an unused variable removed and another variable renamed. The data set is then parsed for missing values. The identified missing values are replaced or removed using a variety of techniques including mean imputation, ratio replacement, removal, logical assumption replacement and constant value substitution. The second main data set is a binding of two smaller data sets. Both smaller data sets are imported from a large excel document, using specialised import specifications. The data sets are then subsetted to produce the respective desired tables. The subsetted data sets are then cleaned by the removal of blank columns. Once clean the data sets are bound by row. This main dataset then has a variable name changed. Both main data sets have their variable data types scanned and corrected. The two main data sets are then merged to form a grand final data set. The final data set has it's data types double-checked, leading to the factorising and labelling of a variable.

Data I

Data Set 1: BITRE__Roadside__drug__testing__data.csv

This data set contains the statistics of Australian roadside drug tests by jurisdiction, for the years 2008 to 2019. The source of this data set is: https://data.gov.au/data/dataset/australian-roadside-drug-testing/resource/67c577de-7d8f-42fa-8119-87d6bb2d6547

Variable Description of this data set:

- Year: Numeric // A value indicating the year
- State: Character // Name of the respective state
- Road Side Drug Test: Numeric // Count of roadside drug tests
- Positive drug test: Numeric // Count of positive drug tests
- Licences: Numeric // Licence Numbers
- Number of deaths from crashes involving a driver or motorcycle rider who had an illegal drug in their system: Numeric // Count of drug-driving related fatalities

Let's import the data set and take a quick look at the beginning 6 rows:

```
RDT <- read_csv("BITRE_Roadside_drug_testing_data.csv") #Importing CSV to variable name 'RDT'
head(RDT) #Snapshot of data set
```

```
## # A tibble: 6 x 6
##    Year State `Road side drug ~ `Positive drug ~ Licences `Number of deaths fro~
##   <dbl> <chr>             <dbl>            <dbl>    <dbl>                  <dbl>
## 1  2008 NSW               20333              542       NA                     NA
## 2  2009 NSW               24884              613       NA                     NA
## 3  2010 NSW               32455              735  4791490                     53
## 4  2011 NSW               33528              666  4893688                     42
## 5  2012 NSW               31446              705  4984973                     48
## 6  2013 NSW               34280              898  5060762                     52
```

I will not require the 'Licences' variable so this can be removed as such:

```
RDT <- RDT %>% select(-Licences) #Removal of variable 'Licences'
```

From the variable description, we can see that the last variable has an enormously long name. This is not needed and therefore will be renamed using the 'colnames' function:

```
colnames(RDT)[5] <- "Drug Related Crash Fatalities" #Renaming of column name 5
```

Data Set 2 & 3: Road Trauma Australia—Annual Summaries

Data sets 2 and 3 come from the same source: https://www.bitre.gov.au/publications/ongoing/road_deaths_australia_annual_summaries

Data set 2 is the annual summaries of road trauma, within Australia, for the years 2004 to 2013. Data set 3 is the annual summaries of road trauma, within Australia, for the years 2010 to 2019. For both data sets, I will only be using the first table, on the specified sheets. Both have the same variables.

Variable Description of this data set:

- Year: Character // A value indicating the year
- Empty: NAN // Blank Column containing no values
- NSW: Numeric // Count of all road related fatalities for the given year, in the respective state
- Vic: Numeric // Count of all road-related fatalities for the given year, in the respective state
- Qld: Numeric // Count of all road-related fatalities for the given year, in the respective state
- SA: Numeric // Count of all road-related fatalities for the given year, in the respective state
- WA: Numeric // Count of all road-related fatalities for the given year, in the respective state
- Tas: Numeric // Count of all road-related fatalities for the given year, in the respective state
- NT: Numeric // Count of all road-related fatalities for the given year, in the respective state
- ACT: Character // Count of all road-related fatalities for the given year, in the respective state

Before previewing the data sets, specifications need to be made for the import. This includes, sheet specification and skip specification, to ensure the document is read at the appropriate part:

```
early_crash <- read_excel("Road_crash_2013.xls", sheet = 2, skip=5)
late_crash <- read_xlsx("Road_crash_2019.xlsx", sheet = 4, skip = 5)
```

These imported data frames are not ready to be previewed yet as they need to be subsetted first.

Data II

Since I only wish to use the first table of the data sets, I will need to do some subsetting. Further down the track, I will merge data set 1 with data set 2 and 3. This means I only want data that both share. In this instance it will be the years between 2008 to 2019, thus this will be the target of my subsetting.

For the 'early_crash' data set, I will subset the rows starting at the year 2008 and onward, therefore I will subset from row 8 to row 13. The table I wish to use is only contained in the first 10 columns, so I will subset the columns 1 through 10. Also, as noted in the variable description, the second column is a blank and just taking up space, therefore this will be subsetted out:

```
early_crash <- early_crash[8:13,1:10] #Subsetting Rows 8 to 13, Columns 1 to 10
early_crash<- early_crash[,-2] #Removing Column 2
```

Let's preview this data set:

```
early_crash
```

```
## # A tibble: 6 x 9
##   ...1  NSW...3 Vic...4 Qld...5 SA...6 WA...7 Tas...8 NT...9 ACT...10
##   <chr>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>  <dbl> <chr>
## 1 2008      374     303     328     99    205      39     75 14
## 2 2009      454     290     331    119    191      63     31 12
## 3 2010      405     288     249    118    193      31     50 19
## 4 2011      364     287     269    103    179      24     45 6
## 5 2012      369     282     280     94    182      31     49 12
## 6 2013      340     242     271     98    162      36     37 7
```

The 'late_crash' data set contains the remaining years, 2014-2019, of the data I wish to use. The required years are located in rows 7 to 12. I will still use all ten columns to extract just the first table from the large spreadsheet. As seen earlier, the removal of column 2 will also happen for this data set:

```
late_crash <- late_crash[7:12,1:10] #Subsetting Rows 7 to 12, Columns 1 to 10
late_crash<- late_crash[,-2] #Removing Column 2
```

Let's preview this data set:

```
late_crash
```

```
## # A tibble: 6 x 9
##   ...1  NSW...3 Vic...4 Qld...5 SA...6 WA...7 Tas...8 NT...9 ACT...10
##   <chr>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>  <dbl> <chr>
## 1 2014      307     248     223    108    183      33     39 10
## 2 2015      350     252     243    102    159      34     49 15
## 3 2016      380     290     251     86    193      37     45 10
## 4 2017      389     259     247    100    159      31     31 5
## 5 2018      347     213     245     80    158      33     50 9
## 6 2019      355     270     219    114    163      32     36 6
```

Now that these two data sets are ready, I can bind them using the 'bind_rows' function to stack them on top of each other, without repeating the variable names. Let's bind them and have a preview of the final data set:

```
total_crash<- bind_rows(early_crash,late_crash) #Data set bind through rows
total_crash
```

```
## # A tibble: 12 x 9
##    ...1    NSW   Vic   Qld    SA    WA   Tas    NT ACT
##    <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
##  1 2008    374   303   328    99   205    39    75 14
##  2 2009    454   290   331   119   191    63    31 12
##  3 2010    405   288   249   118   193    31    50 19
##  4 2011    364   287   269   103   179    24    45 6
##  5 2012    369   282   280    94   182    31    49 12
##  6 2013    340   242   271    98   162    36    37 7
##  7 2014    307   248   223   108   183    33    39 10
##  8 2015    350   252   243   102   159    34    49 15
##  9 2016    380   290   251    86   193    37    45 10
## 10 2017    389   259   247   100   159    31    31 5
## 11 2018    347   213   245    80   158    33    50 9
## 12 2019    355   270   219   114   163    32    36 6
```

From the preview above, we can see the first column is named correctly. Since the variable contains the years, I shall call the column 'Year' with a simple 'colnames' function change:

```
colnames(total_crash)[1]<- c("Year") #Name Change of first column
```

Tidy & Manipulate Data I

In practice, tidy data is ideal to work with, yet most data scrapped or downloaded from the web is not in a tidy data set format. The data set 'total_crash', is not in a tidy format. As seen in the preview above, the column headers between column 2 to 9, are values, and not variable names. The column names are values of the State variable.

To fix this we can use the 'gather' function. To use this function we need to specify a few things. First the names of all the columns we wish to select. In this case, it will be the name of each State located in the data set. Second, we will specify the 'key', which will be 'State'. Thirdly we will specify the 'value' name which will be 'All Road User Deaths' in this instance:

```
total_crash <- total_crash %>%
  gather(`NSW`, `Vic`, `Qld`,`SA`,`WA`,`Tas`,
         `NT`,`ACT`,key = "State", value = "All Road User Deaths")
```

Let's preview the first 5 rows of the data set:

```
head(total_crash, 5) #Preview of the first 5 rows
```

```
## # A tibble: 5 x 3
##   Year  State `All Road User Deaths`
##   <chr> <chr> <chr>
## 1 2008  NSW   374
## 2 2009  NSW   454
## 3 2010  NSW   405
## 4 2011  NSW   364
## 5 2012  NSW   369
```

Scan I

Many times, data sets obtained online, do not have a value for every observation. This requires the need for re-coding of missing data. In the preview of our data set 'RDT', we could see some 'NA' values displayed. This tells us some data is missing, but how much? Let's perform a quick table calculation using the 'table' function and 'is.na' function to get a general idea of how much data we are missing.

```
colSums(is.na(RDT)) #Tabulation of missing values per variable
```

```
##                         Year                        State
##                            0                            0
##            Road side drug test           Positive drug test
##                           17                            5
## Drug Related Crash Fatalities
##                           36
```

As we can see, quite a few variables missing. To tackle this problem, I will break the data set into parts, containing all data for each state.

ACT

Let's filter the original data set, to subset ACT, and see what is missing within the data set:

```
ACT <- RDT %>% filter(State == "ACT") #Subsetting using filter function
ACT
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 ACT                     NA                 NA                        NA
##  2  2009 ACT                     NA                 NA                        NA
##  3  2010 ACT                     NA                 NA                        NA
##  4  2011 ACT                     NA                 NA                        NA
##  5  2012 ACT                   1733                 37                         1
##  6  2013 ACT                   2429                116                         3
##  7  2014 ACT                   2520                392                         2
##  8  2015 ACT                   2090                258                         4
##  9  2016 ACT                   2721                444                         2
## 10  2017 ACT                   2919                504                         0
## 11  2018 ACT                   3328                877                         2
## 12  2019 ACT                   4128                852                        NA
```

'ACT' is missing data for the first 4 rows, as well as a 'Drug Crash Fatalities' count for the year 2019. Since there is such a large gap of data missing, I will go ahead and remove the first 4 rows. As for the missing count, I will replace this missing value with the mean of the other values within the same category, using the 'impute' function. To use the 'impute' function, the specification 'fun' has to be named. In this case, it will be 'mean:

```
ACT <- ACT[5:12,] #Subsetting rows 5 through 12 (Removing of rows 1 to 4)
ACT$`Drug Related Crash Fatalities` <- impute(ACT$`Drug Related Crash Fatalities`,
                                              fun = mean) #Replacing values with mean
```

Let's preview our 'ACT' data set:

```
ACT
```

```
## # A tibble: 8 x 5
##    Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fatal~
##   <dbl> <chr>                <dbl>              <dbl> <impute>
## 1  2012 ACT                   1733                 37 1
## 2  2013 ACT                   2429                116 3
## 3  2014 ACT                   2520                392 2
## 4  2015 ACT                   2090                258 4
## 5  2016 ACT                   2721                444 2
## 6  2017 ACT                   2919                504 0
## 7  2018 ACT                   3328                877 2
## 8  2019 ACT                   4128                852 2
```

NSW

Let's filter the original data set, to subset NSW, and see what is missing within the data set:

```
NSW <- RDT %>% filter(State == "NSW") #Subsetting using filter function
NSW
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 NSW                  20333                542                        NA
##  2  2009 NSW                  24884                613                        NA
##  3  2010 NSW                  32455                735                        53
##  4  2011 NSW                  33528                666                        42
##  5  2012 NSW                  31446                705                        48
##  6  2013 NSW                  34280                898                        52
##  7  2014 NSW                  38830               2096                        50
##  8  2015 NSW                  62247               9123                        75
##  9  2016 NSW                  89101               8220                        83
## 10  2017 NSW                 111176               9273                        81
## 11  2018 NSW                 115874               9067                        69
## 12  2019 NSW                 166351               9446                        NA
```

Here we only have a few missing values for our 'Drug Crash Fatalities' Variable. A mean imputation will be conducted, as seen earlier:

```
NSW$`Drug Related Crash Fatalities` <- impute(NSW$`Drug Related Crash Fatalities`, fun = mean)
```

NT

Let's filter the original data set, to subset NT, and see what is missing within the data set:

```
NT <- RDT %>% filter(State == "NT") #Subsetting using filter function
NT
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 NT                      NA                  9                         2
##  2  2009 NT                      NA                 63                         5
##  3  2010 NT                      NA                107                        12
##  4  2011 NT                      NA                 92                        11
```

```
##  5  2012 NT                       NA                106                          6
##  6  2013 NT                       NA                 84                          8
##  7  2014 NT                       NA                 90                          5
##  8  2015 NT                       NA                120                          6
##  9  2016 NT                       NA                196                         19
## 10  2017 NT                       NA                329                          6
## 11  2018 NT                       NA                341                          7
## 12  2019 NT                       NA                462                         NA
```

This subsetted data set has values missing for the entire variable 'Road side drug test'. Without more info, these observations can not be predicted or guessed. One speculation that can be made with certainty is that there had to be at least one roadside drug test per positive test. This inference leads us to use the values in our 'Positive drug test' variable for our 'Road side drug test' column. There is also one missing value in our 'Drug Crash Fatalities' variable that will be taken care of through mean imputation:

```
NT$`Road side drug test` <- NT$`Positive drug test` #Value Duplication
NT$`Drug Related Crash Fatalities` <- impute(NT$`Drug Related Crash Fatalities`, fun = mean)
```

Qld

Let's filter the original data set, to subset Qld, and see what is missing within the data set:

```
Qld <- RDT %>% filter(State == "Qld") #Subsetting using filter function
Qld
```

```
## # A tibble: 12 x 5
##      Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##     <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 Qld                  10747                216                          0
##  2  2009 Qld                  12489                254                          0
##  3  2010 Qld                  21655                440                          0
##  4  2011 Qld                  25172                825                          0
##  5  2012 Qld                  19686                937                          0
##  6  2013 Qld                  20787               1300                          0
##  7  2014 Qld                  21225               2208                          0
##  8  2015 Qld                  39950               7446                          0
##  9  2016 Qld                  50812              10663                          0
## 10  2017 Qld                  62098              11697                          4
## 11  2018 Qld                  67784              13975                          1
## 12  2019 Qld                  66851              13264                         NA
```

Here we only have a few missing values for our 'Drug Crash Fatalities' Variable. A mean imputation will be conducted as seen earlier:

```
Qld$`Drug Related Crash Fatalities` <- impute(Qld$`Drug Related Crash Fatalities`, fun = mean)
```

SA

Let's filter the original data set, to subset SA, and see what is missing within the data set:

```
SA <- RDT %>% filter(State == "SA") #Subsetting using filter function
SA
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 SA                   25903                600                        11
##  2  2009 SA                   43681               1179                        20
##  3  2010 SA                   45124               1699                        16
##  4  2011 SA                   44178               2320                        14
##  5  2012 SA                   43569               3237                        14
##  6  2013 SA                   51179               3737                        10
##  7  2014 SA                   49777               4681                        17
##  8  2015 SA                   53691               5239                        16
##  9  2016 SA                   48690               4310                        20
## 10  2017 SA                   49626               4337                        22
## 11  2018 SA                   51382               5141                        18
## 12  2019 SA                   49062               4985                        NA
```

Here we only have a few missing values for our 'Drug Crash Fatalities' Variable. A mean imputation will be conducted as seen earlier:

```
SA$`Drug Related Crash Fatalities` <- impute(SA$`Drug Related Crash Fatalities`, fun = mean)
```

Tas

Let's filter the original data set, to subset Tas, and see what is missing within the data set:

```
Tas <- RDT %>% filter(State == "Tas") #Subsetting using filter function
Tas
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 Tas                    412                211                        17
##  2  2009 Tas                     NA                252                        14
##  3  2010 Tas                   1427                 NA                         3
##  4  2011 Tas                   1678                573                         4
##  5  2012 Tas                   1698                523                         3
##  6  2013 Tas                   1819                639                         4
##  7  2014 Tas                   3431               1969                         8
##  8  2015 Tas                   3738               2318                         1
##  9  2016 Tas                   3722               2154                        11
## 10  2017 Tas                   3730               2152                         6
## 11  2018 Tas                   4005               2408                         7
## 12  2019 Tas                   4826               2487                        NA
```

'Tas' data set has a missing value for each variable. The 'Drug Crash Fatalities' will be taken care of through mean imputation. Since 'Road side drug test' and 'Positive drug test' are in somewhat of a ratio, to replace their respective values, I will use ratio replacement. This involves calculating the ratio of the previous set, and applying this ratio to this missing value:

```
Tas$`Drug Related Crash Fatalities` <- impute(Tas$`Drug Related Crash Fatalities`, fun = mean)
#Ratio Replacement
```

```r
Tas$`Road side drug test` [is.na(Tas$`Road side drug test`)] <- round((412/211)*252)
#Ratio Replacement
Tas$`Positive drug test` [is.na(Tas$`Positive drug test`)] <- round(1427/(1678/573))
```

Vic

Let's filter the original data set, to subset Vic, and see what is missing within the data set:

```r
Vic <- RDT %>% filter(State == "Vic") #Subsetting using filter function
Vic
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 Vic                  25006                438                        NA
##  2  2009 Vic                  28083                323                        NA
##  3  2010 Vic                  41642                741                        NA
##  4  2011 Vic                  25140                760                        NA
##  5  2012 Vic                  47745               2180                        NA
##  6  2013 Vic                  39471               2540                        NA
##  7  2014 Vic                  55908               3749                        NA
##  8  2015 Vic                 106503               7823                        NA
##  9  2016 Vic                  95104               9065                        NA
## 10  2017 Vic                 100475               8252                        NA
## 11  2018 Vic                 109780              11548                        NA
## 12  2019 Vic                 176294              11693                        NA
```

Here we can see that 'Drug Related Crash Fatalities' variable is missing all the values. Without further data, these values can not be replaced. In this instance, there are two options. Remove all the rows containing the missing values, or replace with a constant value. In this case, I will replace with a constant value '0', since no fatality was recorded for the state of Vic.

```r
Vic$`Drug Related Crash Fatalities` <- 0 #Constant Value replacement
```

WA

Let's filter the original data set, to subset WA, and see what is missing within the data set:

```r
WA <- RDT %>% filter(State == "WA") #Subsetting using filter function
WA
```

```
## # A tibble: 12 x 5
##     Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fata~
##    <dbl> <chr>                <dbl>              <dbl>                     <dbl>
##  1  2008 WA                    9823                406                        NA
##  2  2009 WA                    7565                289                        NA
##  3  2010 WA                    9773                418                        NA
##  4  2011 WA                    7637                460                        NA
##  5  2012 WA                    9124                623                        NA
##  6  2013 WA                    7265                539                        NA
##  7  2014 WA                   12099               1104                        NA
##  8  2015 WA                   27899               2803                        NA
##  9  2016 WA                   33525               3651                        NA
## 10  2017 WA                   36916               3311                        NA
## 11  2018 WA                   40291               4787                        NA
## 12  2019 WA                   39695               5174                        NA
```

Here we can see the same problem as before. I will complete the same replacement as seen above:

```
WA$`Drug Related Crash Fatalities` <- 0 #Constant Value replacement
```

Date frame restoration Since all values within the original data set have been replaced in some manner, I will merge the subsetted data sets to reform the original 'RDT' data set. To do so I will use the 'bind_rows' function as seen earlier:

```
RDT <- bind_rows(ACT,NSW,NT,Qld,SA,Tas,Vic,WA) #Data frame binding through rows
```

To check we replaced all the missing values, I will once again perform a missing value tabulation:

```
colSums(is.na(RDT)) #Tabulation of missing values per variable
```

```
##                          Year                         State
##                             0                             0
##             Road side drug test           Positive drug test
##                             0                             0
## Drug Related Crash Fatalities
##                             0
```

Excellent! Let's preview the final version of this data set:

```
head(RDT)
```

```
## # A tibble: 6 x 5
##    Year State `Road side drug tes~ `Positive drug te~ `Drug Related Crash Fatal~
##   <dbl> <chr>                <dbl>              <dbl>                      <dbl>
## 1  2012 ACT                   1733                 37                          1
## 2  2013 ACT                   2429                116                          3
## 3  2014 ACT                   2520                392                          2
## 4  2015 ACT                   2090                258                          4
## 5  2016 ACT                   2721                444                          2
## 6  2017 ACT                   2919                504                          0
```

Understand and Merge

Before I merge my two remaining data sets. I will check to see if their respective variables are in the correct format. To do so I will use the 'apply' function with the specification of the function 'mode'. The 'sapply' function will repeat the specified function across all variables in the respective data set. Let's start with 'RDT':

```
sapply(RDT,mode) #Data type display for each variable
```

```
##                          Year                         State
##                     "numeric"                   "character"
##             Road side drug test           Positive drug test
##                     "numeric"                     "numeric"
## Drug Related Crash Fatalities
##                     "numeric"
```

Everything seems to be in order for now. Let's attempt the same check on out 'total_crash' data set:

```
sapply(total_crash,mode) #Data type display for each variable
```

```
##                 Year                 State All Road User Deaths
##          "character"           "character"          "character"
```

The variables 'Year' and 'All Road User Deaths' seem to be in the wrong format. Currently, they are specified as characters but we wish them to be numeric. This change can be made using the 'as.numeric' function:

```
total_crash$Year <- as.numeric(total_crash$Year) #Data Type Change to Numeric
total_crash$`All Road User Deaths` <- as.numeric(total_crash$`All Road User Deaths`)
```

Now that our data sets contain the correct data types we can merge them. In order to do so, the 'merge' function will be used. This merge will be conducted on two entries to the 'by' specification. The entries will be 'State' and 'Year' in that order. The new data set will be called 'Aus_Road':

```
Aus_Road <- merge(RDT, total_crash, by=c("State","Year")) #Data set merge on certain specifications
```

Let's check the data types of 'Aus_Road' to make sure everything is correct:

```
sapply(Aus_Road,mode) #Data type display for each variable
```

```
##                         State                          Year
##                   "character"                     "numeric"
##           Road side drug test           Positive drug test
##                     "numeric"                     "numeric"
## Drug Related Crash Fatalities          All Road User Deaths
##                     "numeric"                     "numeric"
```

The 'State' Variable is a character data type as seen above. This needs to be factored and given new labels to clean up the data set. This can be done through the function 'factor' with specifications of 'levels' and 'labels' used:

```
Aus_Road$State <- Aus_Road$State %>%
  factor(levels = c("ACT","NSW","NT","Qld","SA","Tas","Vic","WA"),
         labels = c("ACT","NSW","NT","QLD","SA","TAS","VIC","WA"))
#Factoring and labeling of variable
```

Let's have a final preview:

```
head(Aus_Road)
```

```
##   State Year Road side drug test Positive drug test
## 1   ACT 2012                1733                 37
## 2   ACT 2013                2429                116
## 3   ACT 2014                2520                392
## 4   ACT 2015                2090                258
## 5   ACT 2016                2721                444
## 6   ACT 2017                2919                504
##   Drug Related Crash Fatalities All Road User Deaths
## 1                             1                   12
## 2                             3                    7
## 3                             2                   10
## 4                             4                   15
## 5                             2                   10
## 6                             0                    5
```

Tidy & Manipulate Data II

From our new data set 'Aus_Road', we can calculate some interesting statistics. Since we have drug-related fatalities and all road fatalities, we can see the percentage of total drug-related fatalities compared to all road fatalities. To do this we can use the 'mutate' function. From this function, we can specify a new variable with a given name and a given calculation. The calculation will be 'Drug Related Crash Fatalities' divided by 'All Road User Deaths' then multiplied by 100. This calculation will be rounded to two decimal places using the 'round' function:

```
Aus_Road <- Aus_Road %>%
  mutate("Percent of Drug Fatalities" =
           (`Drug Related Crash Fatalities`/`All Road User Deaths`)*100) #Mutation of new variable
Aus_Road$`Percent of Drug Fatalities` <- round(Aus_Road$`Percent of Drug Fatalities`,2) #Rounding of va
```

Let's preview this addition:

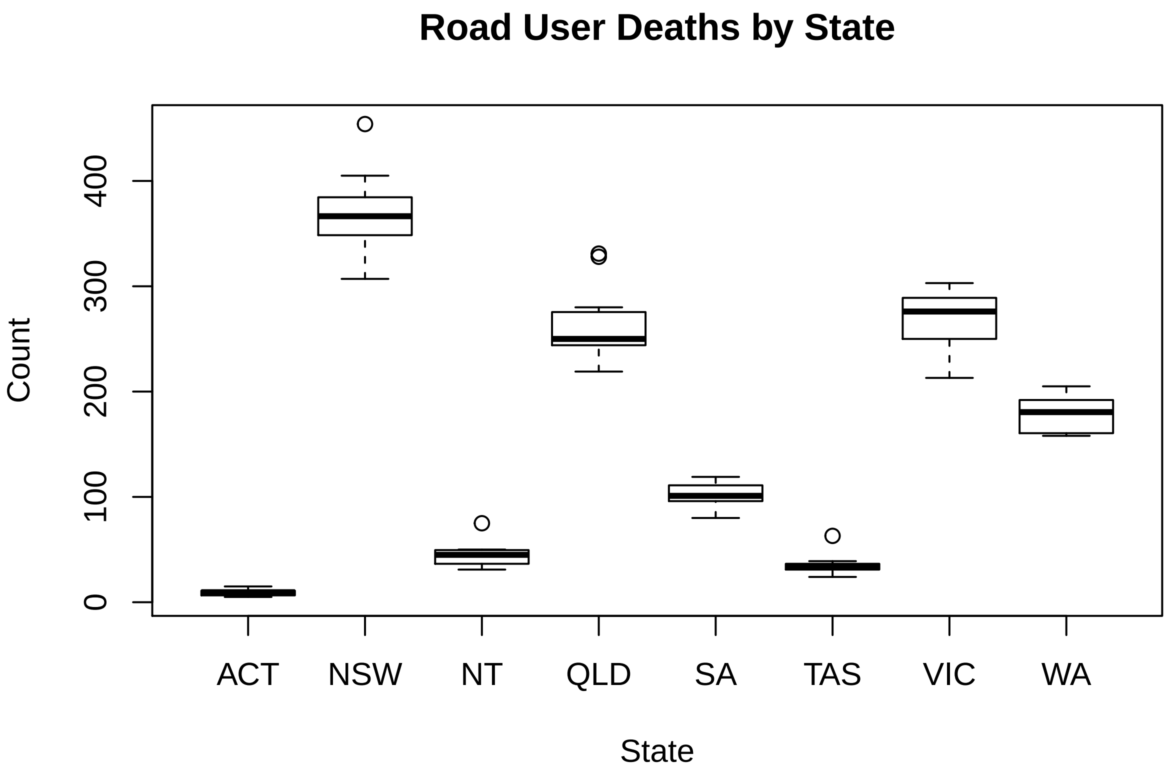
```
head(Aus_Road)
```

```
##   State Year Road side drug test Positive drug test
## 1   ACT 2012                1733                 37
## 2   ACT 2013                2429                116
## 3   ACT 2014                2520                392
## 4   ACT 2015                2090                258
## 5   ACT 2016                2721                444
## 6   ACT 2017                2919                504
##   Drug Related Crash Fatalities All Road User Deaths Percent of Drug Fatalities
## 1                             1                   12                       8.33
## 2                             3                    7                      42.86
## 3                             2                   10                      20.00
## 4                             4                   15                      26.67
## 5                             2                   10                      20.00
## 6                             0                    5                       0.00
```

Scan II

In data, outliers can be present. In order to deal with outliers, one must first locate them within the data set. In the 'Aus_Road' data set, we can have a quick outlier scan using a box plot. For this scan, I will be using the 'All Road User Deaths' variable compared to each 'State' through the function 'boxplot'. I will give the plot a name through the specification 'main', a y-axis name through the specification 'ylab', and a colour to the plot through the specification 'col':

```
boxplot(Aus_Road$`All Road User Deaths` ~ Aus_Road$State,
        main="Road User Deaths by State",
        ylab = "Count", xlab = "State")
```



We can see that State's 'NSW', 'NT', 'QLD', and 'TAS' all have outliers, but which observations are they?

Since each outlier per State is above the max, we will attempt to remove said outliers by filtering through the respective max. These maxes can be found through summary statistics. Using the function 'group_by' and a specification 'State', the data can be grouped without formatting the actual data frame. The 'summarise' function produces a convenient summary according to specifications. In this instance we are looking for the max, so we will input the 'quantile' function, with the specification of 'prob' equalling .75 and multiple this by 1,5. This all together produces the following table:

```
Aus_Road %>%
  group_by(State) %>%
  summarise(MAX =  quantile(`All Road User Deaths`,probs = .75,na.rm
                                = TRUE)*1.5)
```

```
## # A tibble: 8 x 2
##   State   MAX
##   <fct> <dbl>
## 1 ACT    15.8
## 2 NSW   573.
## 3 NT     73.9
## 4 QLD   410.
## 5 SA    164.
## 6 TAS    54.4
## 7 VIC   433.
## 8 WA    287.
```

From this table, we can see the Maxs for the outliers per State discussed earlier. NSW max is 573, NT max is 73.9, QLD max is 410, and TAS max is 54.4. Working with these maxes, we can remove the outliers. Using the 'which' function we can locate the observation that falls within our specification of particular State and max:

```
which(Aus_Road$State == "NSW" & Aus_Road$`All Road User Deaths`>573)
```

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

```
which(Aus_Road$State == "NT" & Aus_Road$`All Road User Deaths`>73.9)
```

```
## [1] 21
```

```
which(Aus_Road$State == "QLD" & Aus_Road$`All Road User Deaths`>410)
```

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

```
which(Aus_Road$State == "TAS" & Aus_Road$`All Road User Deaths`>54.4)
```

```
## [1] 58
```

The output above indicates that we must remove rows 21 and 58. We can do this using the 'slice' function:

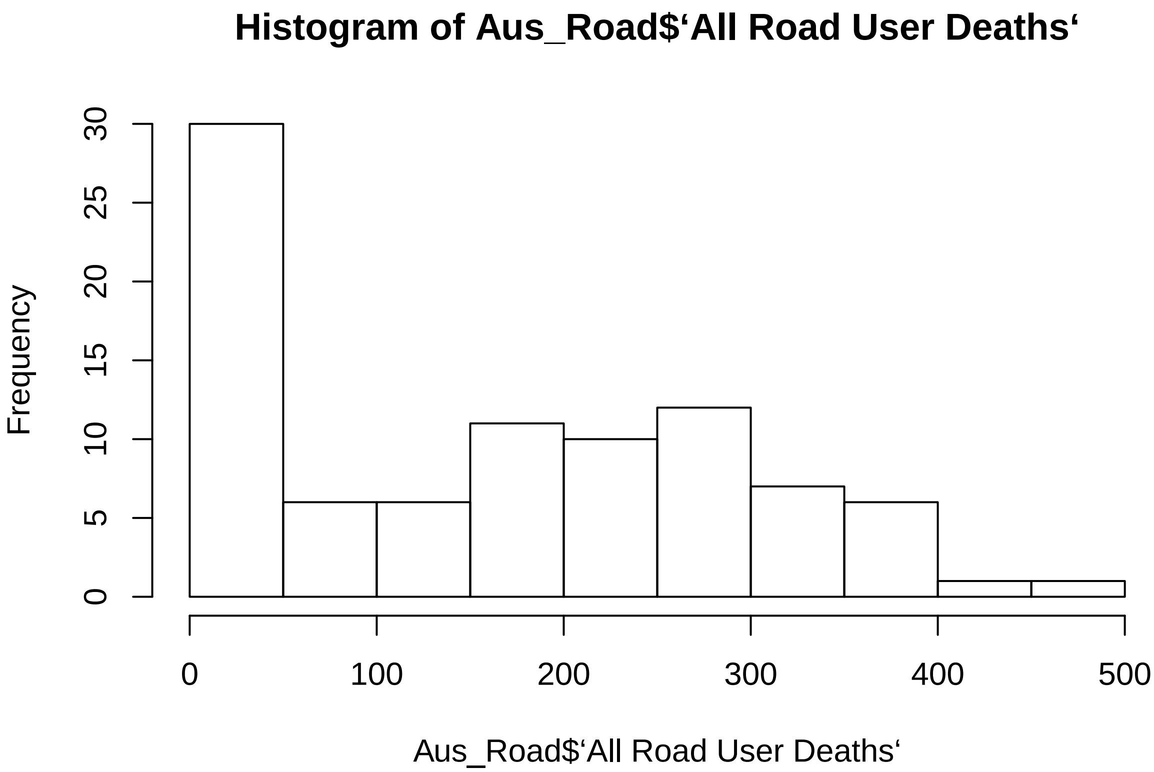
```
Aus_Road <- Aus_Road %>% slice(-c(21,58))
```

Excellent, we have removed all outliers in the variable 'All Road User Deaths'!

Transform

The 'hist' function produces a histogram from values. Let's take a look at a histogram of our variable 'All Road User Deaths', from which we removed the outliers:

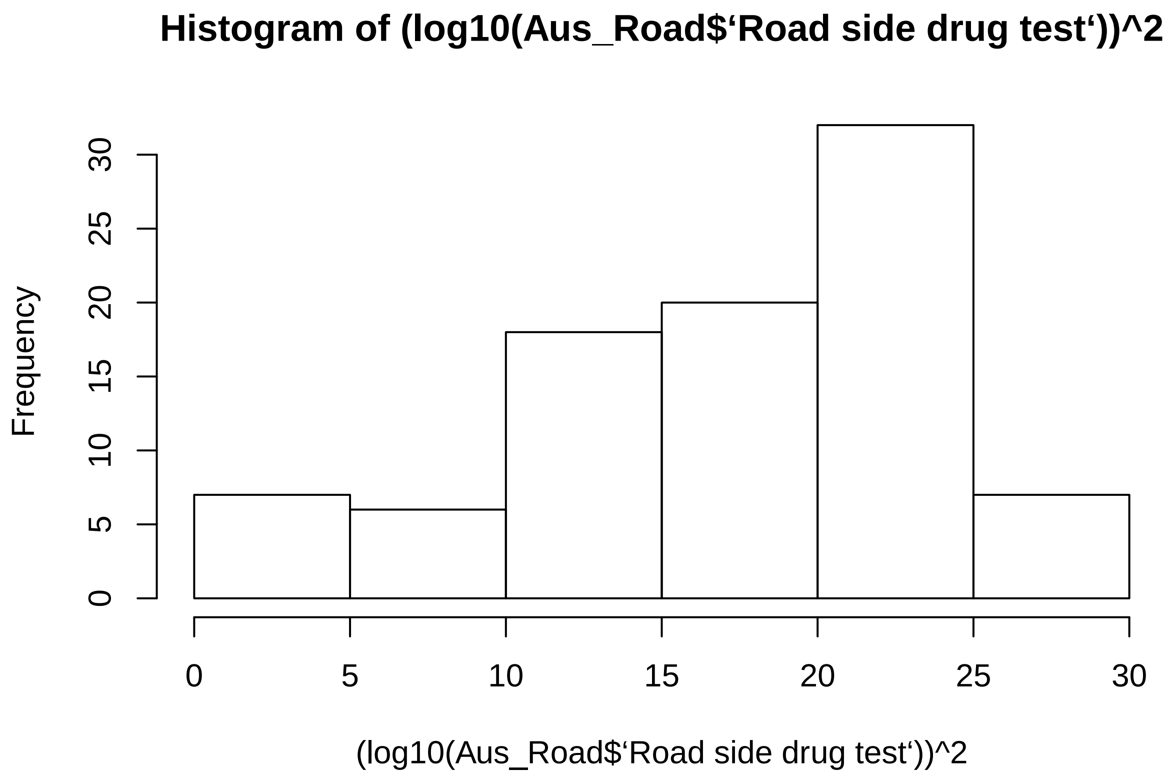
```
hist(Aus_Road$`All Road User Deaths`)
```



We can see that this histogram doesn't follow a normal distribution. In order to obtain a normal distribution, we will perform a transformation.

The transformation I will use to obtain a somewhat normal distribution will be log10 transformation combined with a square transformation. Using the 'log10' function combined with squaring the result within our 'hist' function:

```
hist((log10(Aus_Road$`Road side drug test`))^2)
```



Way Better Looking!

References

- BITRE_Roadside_drug_testing_data.Csv. 22 Sept. 2020, data.gov.au/data/dataset/australian-roadside-drug-testing/resource/67c577de-7d8f-42fa-8119-87d6bb2d6547.
- Bureau of Infrastructure and Transport Research Economics. "Road Trauma Australia-Annual Summaries." Bureau of Infrastructure and Transport Research Economics, Bureau of Infrastructure and Transport Research Economics, 9 July 2020,

www.bitre.gov.au/publications/ongoing/road_deaths_australia_annual_summaries.

- Dolgun, Anil. "One Account. All of Google." Sign in - Google Accounts, 22 June 2020, rare-phoenix-161610.appspot.com/secured/Module_04.html.
- Dolgun, Anil. "One Account. All of Google." Sign in - Google Accounts, 22 June 2020, rare-phoenix-161610.appspot.com/secured/Module_06.html.