

Assignment 2

Liam

9/16/2019

1. Download the c2015 dataset to your computer. Use function `getwd()` to check the current working directory. Use `setwd()` to change the current directory to the c2015 file.

```
setwd('C:\\Users\\student\\Desktop\\Fall2019\\R')
```

2. We need to install a package to read the xlsx file. (Let's not change the xlsx to csv here) There are a few packages for this. I recommend to use the `readxl` package. This package is contained in the `tidyverse` package so if you already installed `tidyverse`, you should have it already. If not, install and load the `readxl` package by

```
#install.packages('readxl') # install the library
library(readxl) # load the library
```

3. Use `read_excel()` to read the c2015 dataset. Use function `class()` to check the type of data you just read in. You will notice that the data now is not just a data frame, it is also a `tibble`. A `tibble` is a generalization of a data frame, so you can still use all the functions and syntax for data frame with `tibble`.

```
c2015 <- read_excel('c2015.xlsx')
class(c2015)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

4. Use `dim` function to check the dimension of the data. Since this data is quite big, a common practice is to randomly subset the data to analyze. Use `sample` function to create a new dataset that has a random 1000 observations from the original data. Use `set.seed(2019)` before using the `sample` function to set the seed for the randomness so that everyone in class is working with the same random subset of the data.

```
dim(c2015)
```

```
## [1] 80587    28
```

```
set.seed(2019)
c2015Sample <- c2015[sample(nrow(c2015), 1000),]
```

5. Use `summary` function to have a quick look at the data. You will notice there is one variable is actually a constant. Remove that variable from the data.

```
summary(c2015Sample)
```

```

##      STATE          ST_CASE          VEH_NO          PER_NO
## Length:1000      Min.    : 10020      Min.    : 0.000      Min.    : 1.000
## Class :character  1st Qu.:122408      1st Qu.: 1.000      1st Qu.: 1.000
## Mode  :character  Median :270249      Median : 1.000      Median : 1.000
##                               Mean  :276444      Mean  : 1.385      Mean  : 1.697
##                               3rd Qu.:420726      3rd Qu.: 2.000      3rd Qu.: 2.000
##                               Max.   :560071      Max.   :13.000      Max.   :48.000
##
##      COUNTY          DAY          MONTH          HOUR
## Min.    : 1.00      Min.    : 1.00      Length:1000      Min.    : 0.00
## 1st Qu.: 32.50      1st Qu.: 8.00      Class :character  1st Qu.: 8.00
## Median : 71.00      Median :16.00      Mode  :character  Median :16.00
## Mean   : 93.05      Mean   :15.89                               Mean   :14.26
## 3rd Qu.:117.00      3rd Qu.:24.00                               3rd Qu.:20.00
## Max.   :810.00      Max.   :31.00                               Max.   :99.00
##
##      MINUTE          AGE          SEX          PER_TYP
## Min.    : 0.00      Length:1000      Length:1000      Length:1000
## 1st Qu.:14.00      Class :character  Class :character  Class :character
## Median :27.00      Mode  :character  Mode  :character  Mode  :character
## Mean   :27.76
## 3rd Qu.:43.00
## Max.   :59.00
## NA's    :5
##      INJ_SEV          SEAT_POS          DRINKING          YEAR
## Length:1000      Length:1000      Length:1000      Min.    :2015
## Class :character  Class :character  Class :character  1st Qu.:2015
## Mode  :character  Mode  :character  Mode  :character  Median :2015
##                               Mean   :2015
##                               3rd Qu.:2015
##                               Max.   :2015
##
##      MAN_COLL          OWNER          MOD_YEAR
## Length:1000      Length:1000      Length:1000
## Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character
##
##
##
##      TRAV_SP          DEFORMED          DAY_WEEK
## Length:1000      Length:1000      Length:1000
## Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character
##
##
##
##      ROUTE          LATITUDE          LONGITUD          HARM_EV
## Length:1000      Min.    :21.30      Min.    : -160.34      Length:1000
## Class :character  1st Qu.:33.48      1st Qu.: -97.59      Class :character
## Mode  :character  Median :36.42      Median : -87.43      Mode  :character
##                               Mean   :36.72      Mean   : -91.83
##                               3rd Qu.:40.40      3rd Qu.: -81.41

```

```
##           Max.      :61.54   Max.      : -67.72
##           NA's      :7       NA's      :7
##   LGT_COND          WEATHER
## Length:1000        Length:1000
## Class :character   Class :character
## Mode  :character   Mode  :character
##
##
##
##
```

```
c2015Sample <- subset(c2015Sample, select = -c(YEAR))
```

6. Check the number of missing values (NA) in each column.

```
colSums(is.na(c2015Sample))
```

```
##   STATE ST_CASE  VEH_NO  PER_NO  COUNTY    DAY    MONTH    HOUR
##     0      0      0      0      0      0      0      0
## MINUTE    AGE    SEX PER_TYP INJ_SEV SEAT_POS DRINKING MAN_COLL
##     5      0      0      0      0      0      0      95
##   OWNER MOD_YEAR TRAV_SP DEFORMED DAY_WEEK    ROUTE LATITUDE LONGITUD
##     95      95      95      95      0      0      7      7
## HARM_EV LGT_COND WEATHER
##     0      0      0
```

7. There are missing values in this data that are not NAs. Identify the form of these missing values. Check the number of these missing values in each column. Notice that you may want to use `na.rm = TRUE` when counting these missing values.

```
c2015SampleB <-c2015Sample
c2015SampleB[c2015Sample=="Unknown" | c2015SampleB=="Unkno" | c2015SampleB=="Unknown (Police Reported)"]
colSums(is.na(c2015SampleB))
```

```
##   STATE ST_CASE  VEH_NO  PER_NO  COUNTY    DAY    MONTH    HOUR
##     0      0      0      0      0      0      0      0
## MINUTE    AGE    SEX PER_TYP INJ_SEV SEAT_POS DRINKING MAN_COLL
##     5      16     11      2      8      11     496      99
##   OWNER MOD_YEAR TRAV_SP DEFORMED DAY_WEEK    ROUTE LATITUDE LONGITUD
##    118      111      629      159      0      36      7      7
## HARM_EV LGT_COND WEATHER
##     0      7      14
```

8. Change the missing values in SEX variable to “Female”

```
unique(c2015Sample$SEX)
```

```
## [1] "Unknown" "Female"  "Male"    "Not Rep"
```

```
c2015Sample$SEX[is.na(c2015Sample$SEX) | c2015Sample$SEX=="Not Rep" | c2015Sample$SEX=="Unknown"] <- "F"
```

9. Fix the AGE variable so that it is in the right form and has no missing values. **Hint:**

- Change the value `Less than 1` to 0 (string 0, not a number 0)
- Change the type of the variable to numeric using `as.numeric` function
- Change the missing values to the average of the age.

```
c2015Sample$AGE[c2015Sample$AGE=="Less than 1"] <- '0'
c2015Sample$AGE <- as.numeric(c2015Sample$AGE)
```

```
## Warning: NAs introduced by coercion
```

```
c2015Sample$AGE[is.na(c2015Sample$AGE)] <- colMeans(c2015Sample['AGE'], na.rm=TRUE)
```

10. Put the TRAV_SP(Travel Speed) variable in the right form (type) and remove all missing values. Calculate the average speed. You can use a non-base R function for this question. **Hint:** check out the function `str_replace`

```
c2015Sample$TRAV_SP[c2015Sample$TRAV_SP=="Stopped"]<-'0'
c2015Sample$TRAV_SP<-stringr::str_replace(c2015Sample$TRAV_SP, ' MPH', '')
c2015Sample$TRAV_SP[c2015Sample$TRAV_SP=="Unknown" | c2015Sample$TRAV_SP=="Not Rep"] <- NA
c2015Sample$TRAV_SP <-as.numeric(c2015Sample$TRAV_SP)
c2015Sample <- c2015Sample[!is.na(c2015Sample$TRAV_SP),]
mean(c2015Sample$TRAV_SP)
```

```
## [1] 43.79245
```

11. Compare the average speed of those who had "No Apparent Injury" and the rest. What do you observe?

```
no_injury<-mean(c2015Sample$TRAV_SP[c2015Sample$INJ_SEV=='No Apparent Injury (0)'])
all_others<-mean(c2015Sample$TRAV_SP[!c2015Sample$INJ_SEV=='No Apparent Injury (0)'])
c(no_injury,all_others)
```

```
## [1] 33.57265 48.50000
```

No injury has a lower travel speed

12. Use the SEAT_POS variable to filter the data so that there is only **drivers** in the dataset. Compare the average speed of man drivers and woman drivers. Comment on the results.

```
male <-mean(c2015Sample$TRAV_SP[c2015Sample$SEAT_POS=="Front Seat, Left Side" & c2015Sample$SEX == "Male"])
female <-mean(c2015Sample$TRAV_SP[c2015Sample$SEAT_POS=="Front Seat, Left Side" & c2015Sample$SEX == "Female"])
c(male, female)
```

```
## [1] 45.57647 37.11429
```

13. Compare the average speed of drivers who drink and those who do not. Comment on the results. **Hint:** This calculation can be done manually or by using the `aggregate` function or by function in base R. For example:

```
drink<-mean(c2015Sample$TRAV_SP[c2015Sample$DRINKING=='Yes (Alcohol Involved)'])
notdrink<-mean(c2015Sample$TRAV_SP[c2015Sample$DRINKING!='Yes (Alcohol Involved)'])
c(drink,notdrink)
```

Interesting to see that drunk drivers travel almost 25 mph higher than sober drivers.

14. Hypothesize about the age range of drivers who may drive more aggressively. Test your hypothesis by comparing the average speed of those in this age range and the rest. Comment on the results.

I'd imagine young drivers (<25) drive more aggressively than older drivers

```
young<-mean(c2015Sample$TRAV_SP[c2015Sample$AGE<25])
notyoung<-mean(c2015Sample$TRAV_SP[c2015Sample$AGE>24])
c(young,notyoung)
```

```
## [1] 46.39450 42.70992
```

Younger drivers drive around 3-4 mph faster than older drivers

15. If the data did not confirm your hypothesis in 14. Could you identify an age group of drivers who may drive more aggressively?

The age group I found to drive more aggressively were those under the age of 25.