STA 445 HW3

Breelyn Cocke

February 29th, 2023

```
library(tidyverse)
library(readr)
library(readxl)
```

Problem 1

Download from GitHub the data file Example_5.xls. Open it in Excel and figure out which sheet of data we should import into R. At the same time figure out how many initial rows need to be skipped. Import the data set into a data frame and show the structure of the imported data using the `str()` command. Make sure that your data has $n = 31$ observations and the three columns are appropriately named. If you make any modifications to the data file, comment on those modifications.

```
treedata <- read_excel('Example_5.xls', sheet='RawData', range ='A5:C36')
str(treedata)
```

```
## tibble [31 x 3] (S3: tbl_df/tbl/data.frame)
##  $ Girth : num [1:31] 8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
##  $ Height: num [1:31] 70 65 63 72 81 83 66 75 80 75 ...
##  $ Volume: num [1:31] 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

The imported data takes the structure of a tibble since I used the read_excel command vs the read.excel command. There are 31 observations in the 3 labeled columns.

Problem 2

Download from GitHub the data file Example_3.xls. Import the data set into a data frame and show the structure of the imported data using the `tail()` command which shows the last few rows of a data table. Make sure the Tesla values are `NA` where appropriate and that both `-9999` and `NA` are imported as NA values. If you make any modifications to the data file, comment on those modifications.

```
cardata <- read_excel('Example_3.xls', sheet='data', range='A1:L34', na= c('NA', '-9999'))
tail(cardata)
```

```
## # A tibble: 6 x 12
##   model          mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##   <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Lotus Europa  30.4     4  95.1   113  3.77  1.51  16.9     1     1     5     2
## 2 Ford Panter~  15.8     8 351     264  4.22  3.17  14.5     0     1     5     4
## 3 Ferrari Dino  19.7     6 145     175  3.62  2.77  15.5     0     1     5     6
## 4 Maserati Bo~  15       8 301     335  3.54  3.57  14.6     0     1     5     8
## 5 Volvo 142E    21.4     4 121     109  4.11  2.78  18.6     1     1     4     2
## 6 Tesla Model~  98      NA  NA     778 NA     4.94  10.4    NA     0     1    NA
```

Problem 3

Download all of the files from GitHub `data-raw/InsectSurveys` directory here. Each month's file contains a sheet that contains site level information about each of the sites that were surveyed. The second sheet contains information about the number of each species that was observed at each site. Import the data for each month and create a single `site` data frame with information from each month. Do the same for the `observations`. Document any modifications you make to the data files. Comment on the importance of consistency of your data input sheets.

```
read_excel('May.xlsx', sheet=1, range ='A1:F10', na=c('NA'))
read_excel('June.xlsx', sheet=1, range ='A1:F10', na=c('NA'))
read_excel('July.xlsx', sheet=1, range ='A1:F10', na=c('NA'))
read_excel('August.xlsx', sheet=1, range ='A1:F10', na=c('NA'))
read_excel('September.xlsx', sheet=1, range ='A1:F10', na=c('NA'))
read_excel('October.xlsx', sheet=1, range ='A1:F10', na=c('NA'))
```

```
files <- c('May.xlsx', 'June.xlsx', 'July.xlsx', 'August.xlsx', 'September.xlsx', 'October.xlsx')
data <- NULL
for(file in files){
  temp.data <- read_excel(file, sheet=1, range ='A1:F10', na=c('NA'))
  data <- rbind(data, temp.data)}
data
```

```
## # A tibble: 54 x 6
##    `Site Name`      `Pond Area` `Water Depth`    ph Date                Observer
##    <chr>                  <dbl>         <dbl> <dbl> <dttm>              <chr>
##  1 Araphahoe Road            34             3   6.2  2020-05-15 00:00:00 Bob
##  2 Bridger Valley           240             6   6.5  2020-05-16 00:00:00 Bob
##  3 Calculus Vector          321            13   6.4  2020-05-17 00:00:00 Bob
##  4 Deer Valley               74             4.4 6.9  2020-05-18 00:00:00 Bob
##  5 Ephemeral Stream          28             2   7.1  2020-05-15 00:00:00 Charlie
##  6 Fennel Gardens            62             3.6 7    2020-05-16 00:00:00 Charlie
##  7 Gigantic Pain            489             4   7.1  2020-05-17 00:00:00 Charlie
##  8 Happy Feet               398            10   6.8  2020-05-18 00:00:00 Charlie
##  9 Indigo Flats             126             9   6.75 2020-05-19 00:00:00 Charlie
## 10 Araphahoe Road            34             3   6.2  2020-06-15 00:00:00 Bob
## # i 44 more rows
```

```
read_excel('May.xlsx', sheet=2, range ='A1:C37', na=c('NA'))
read_excel('June.xlsx', sheet=2, range ='A1:C37', na=c('NA'))
read_excel('July.xlsx', sheet=2, range ='A1:C37', na=c('NA'))
read_excel('August.xlsx', sheet=2, range ='A1:C37', na=c('NA'))
read_excel('September.xlsx', sheet=2, range ='A1:C37', na=c('NA'))
read_excel('October.xlsx', sheet=2, range ='A1:C37', na=c('NA'))
```

```
files1 <- c('May.xlsx', 'June.xlsx', 'July.xlsx', 'August.xlsx', 'September.xlsx', 'October.xlsx')
data1 <- NULL
for(file in files1){
  temp.data1 <- read_excel(file, sheet=2, range ='A1:C37', na=c('NA'))
  data1 <- rbind(data1, temp.data1)}
data1
```

```
## # A tibble: 216 x 3
##    Site            Species    Count
##    <chr>           <chr>      <dbl>
##  1 Araphahoe Road  Caddis Fly     2
##  2 <NA>            May Fly        4
```

```
##  3 <NA>            Stone Fly      8
##  4 <NA>            Dragon Fly     7
##  5 Bridger Valley  Caddis Fly     2
##  6 <NA>            May Fly        4
##  7 <NA>            Stone Fly      8
##  8 <NA>            Dragon Fly     7
##  9 Calculus Vector Caddis Fly     2
## 10 <NA>            May Fly        4
## # i 206 more rows
```

The data files in excel required a lot of changes to generate consistency between the sheets. In order to be able to use the for loop and rbind in the manner that I did, the sheet, row, and column names had to be perfectly aligned without error in spacing, capitalization, and order. For example, there were many times in which I had to change capitalization, such as "sites" to "Sites". I also had to make sure each date was coded the same way, and get rid of unwanted character strings by changing to "NA". It took a bit of time to correct the data within the excel sheets to make sure each of my sheets matched correctly. This time spent could have been avoided if consistency had been kept when making the original files. There is more room for error without consistency, and essentially we were on a goose chase trying to track down and eliminate discrepancies. Going forward, I will note the importance of maintaining consistency in data labeling and organization to save future time constraints.