# STA130 Rstudio Homework

## Problem Set 1

Josh Speagle & Scott Schwartz

## Instructions

Complete the exercises in this `.Rmd` file and submit your `.Rmd` and knitted `.pdf` output through Quercus by 11:59 pm E.T. on Sunday, January 22.

# Part 1: R Coding Practice

## Question 1

For this question we will work with data related to the old TV show *Avatar: The Last Airbender*.

- The data is stored in the file `avatar.csv` in the same directory as this file (HW1).

  This data was posted on github by user averyrobbins1 and subsequently featured on Tidy Tuesday. For more information see the above links; or, install the package with `devtools::install_github("averyrobbins1/appa")` and then type `help(appa)`.

### (a) Load the data set from the file `avatar.csv` using `read_csv` and save it as an object named "avatar".

```r
# Write your answer below
# Don't forget to put quote marks around the data set name in the function
install.packages("tidyverse")
library(tidyverse)
avatar <- read_csv(file = 'avatar.csv')
```

### Hints to help fix common "gotchas"

- `Error in read_csv(avatar.csv) : could not find function "read_csv"`
  - *Have you loaded the appropriate libraries? I.e., `library(tidyverse)`?*
- `Error in standardise_path(file) : object 'avatar.csv' not found`
  - *Do you have quotes around the file name?*
- `Error: 'avatar.csv' does not exist in current working directory (...).`
  - *Are you running code as `<ctrl-shift-end>` (PC) or `<cmd-shift-enter>` (Mac)?*

**(b) We learned about two functions in class that let us quickly get an idea of our data: `glimpse()` and `head()`. Using `%>%`, "pipe" the `avatar` object you created into each of these functions. (See HW0 for some additional examples of using the "pipe".)**

```
# Write your answer below
avatar %>% glimpse()
```

```
## Rows: 9,992
## Columns: 10
## $ book            <chr> "Water", "Water", "Water", "Water", "Water", "Water", ~
## $ book_num        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ chapter         <chr> "The Boy in the Iceberg", "The Boy in the Iceberg", "T~
## $ chapter_num     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ character       <chr> "Katara", "Sokka", "Katara", "Sokka", "Katara", "Katar~
## $ full_text       <chr> "Water. Earth. Fire. Air. My grandmother used to tell ~
## $ character_words <chr> "Water. Earth. Fire. Air. My grandmother used to tell ~
## $ mention_appa    <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE~
## $ director        <chr> "Dave Filoni", "Dave Filoni", "Dave Filoni", "Dave Fil~
## $ imdb_rating     <dbl> 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1,~
```

```
# Write your answer below
avatar %>% head()
```

```
## # A tibble: 6 x 10
##   book  book_num chapter chapt~1 chara~2 full_~3 chara~4 menti~5 direc~6 imdb_~7
##   <chr>    <dbl> <chr>     <dbl> <chr>   <chr>   <chr>   <lgl>   <chr>     <dbl>
## 1 Water        1 The Bo~       1 Katara  Water.~ Water.~ FALSE   Dave F~     8.1
## 2 Water        1 The Bo~       1 Sokka   It's n~ It's n~ FALSE   Dave F~     8.1
## 3 Water        1 The Bo~       1 Katara  [Happi~ Sokka,~ FALSE   Dave F~     8.1
## 4 Water        1 The Bo~       1 Sokka   [Close~ Sshh! ~ FALSE   Dave F~     8.1
## 5 Water        1 The Bo~       1 Katara  [Strug~ But, S~ FALSE   Dave F~     8.1
## 6 Water        1 The Bo~       1 Katara  [Excla~ Hey!    FALSE   Dave F~     8.1
## # ... with abbreviated variable names 1: chapter_num, 2: character,
## #   3: full_text, 4: character_words, 5: mention_appa, 6: director,
## #   7: imdb_rating
```

**(c) Run the two code chunks below using (PC) or (MAC) or the "play" button, and then compare their output to the output of the `glimpse()` and `head()` functions above.**

```
avatar
```

```
avatar %>% head(12) # <- try another number instead of 3... maybe 12?
```

- Is the `glimpse()` output or the `head()` output a `tibble`?

*both are tibbles according to is_tibble()*

- Which function allows you to look at the first `n` rows of a data set?

*head()*

- Which function lists data set columns vertically rather than horizontally so you can immediately see them all?

*glimpse()*

- How many observations does the `avatar` data frame include?

*9992*

- How many variables are measured for each observation?

*10*

- How many rows and columns does the `avatar` data frame have?

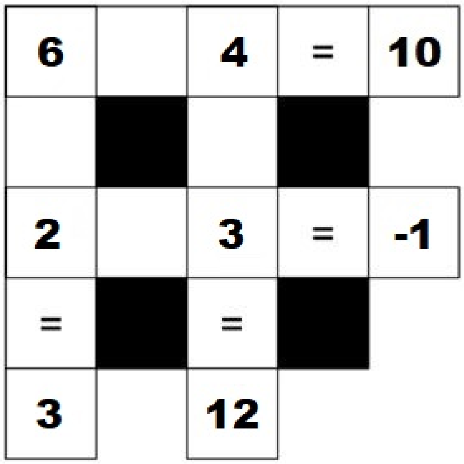
*rows: 9992 columns: 10*

- Is the information for the three previous questions available from the `glimpse()` function or the `head()` function?

*glimpse() has information for all 3, while head() only has information for variables*

## Question 2

Below is a 'math square puzzle'. The value for each row and column is shown after the equals signs, but the operations (`+`, `-`, `*`, `/`) producing the resuts are missing. For example, a row with "2 [blank] 7 = 14" is missing a multiplication (`*`) operation.



| | | | | |
|---|---|---|---|---|
| 6 | | 4 | = | 10 |
| | ■ | | ■ | |
| 2 | | 3 | = | -1 |
| = | ■ | = | ■ | |
| 3 | | 12 | | |

Figure 1: A math square puzzle

**(a) Write out the full correct equations below and assign them to the appropriate names. The first rows has been completed as an example.**

```
# Row 1 (r1)
r1 <- 6 + 4

# Row 2 (r2)
r2 <- -1

# Column 1 (c1)
c1 <- 6 / 2

# Column 2 (c2)
c2 <- 4 * 3
```

**(b) Now, let's check each of your answers individually with the logical `==` operation.**

```
r1 == 10
```

```
## [1] TRUE
```

```
r2 == -1
```

```
## [1] TRUE
```

```
c1 == 3
```

```
## [1] TRUE
```

```
c2 == 12
```

```
## [1] TRUE
```

**(c) Now, let's check each of your answers at the same time with logical `&` operations.**

```
(r1 == 10) & (r2 == -1) & (c1 == 3) & (c2 == 12)
```

```
## [1] TRUE
```

```

my_answers <- c(r1,r2,c1,c2)
square_answers <- c(10,-1,3,12)
my_answers == square_answers
```

**Consider the code above relative to the code below using the `c()` "concatentation" function which "combines" objects into a *vector*.**

```
## [1] TRUE TRUE TRUE TRUE
```

```

correctness <- my_answers == square_answers
all(correctness)
```

**Consider the code above relative to the code below using the `all()` function which checks if every element of a vector is `TRUE`.**

```
## [1] TRUE
```

### (d) What is the benefit of using the `c()` and `all()` functions compared to just writing everything out with logical `==` and `&` operators?

*its less efficient and takes longer time to run due to R being a vectorized computer program*

**Hints**

- Right now we just have `r1`, `r2`, `c1` and `c2`. But what if we had a bigger math square that went all the way up to, say, `r100` and `c100`?
- "Vectorized" computer operations do a series of individual observations in parallel, rather than sequentially. So just like writing out things sequentially takes a long time, doing operations sequentially with a computer also takes more time than just computing them in parallel.

### (e) What is the difference between the code below and the `all(correctness)` code above?

```
sum(correctness)
```

```
## [1] 4
```

*all(correctness) checked if each value inside the boolean vector "correctness" was true while sum added the number of TRUE bools and returned it*

# Part 2: TUT communication/writing exercises *Primer Questions*

You are expected to be efficient with your time in this section, and should ***spend no more than 30 minutes*** on this section.

## Question 1

### (a) How is it that you have come to take STA130?

*STA130 is the first class in my pursuit of my passion in data science and statistcs*

### (b) Do you currently have a sense of the kind of career (e.g., industry, company, type of work) you think you might want to pursue? Please describe your current thinking on this aspect of your university experience.

*I am very much leaning towards a data analyst job either in finance or health. I've always enjoyed applied math such as statistics, I enjoy critical thinking and problem solving and I think my communication skills are decent so I will do my best in this course to pursue my dream*

## Question 2

Suppose you ask your friends to name 10 songs produced prior to Dec 31, 1999 and 10 songs produced after Jan 1, 2000. Then, suppose you check the song statistics on Spotify.

### (a) If the total number of times the older songs have been listened to is greater than the newer songs, would this confirm that music from earlier periods is better than music now?

*Not necessarily, in STA199 and with the survivorship bias airplane problem we're taught to look at the full picture before drawing any conclusions. If I were to name one issue at the moment is that Spotify is a modern app and most of the old "classics" were listened to by a generation that most likely doesn't use this app; alongside that most of my friends are of my age and probably are biased to argue our generation is "better" and will name hit songs that have access to a much larger listener/streaming base.*

**(b) If the average number of times (per user and per year) the older songs have been listened to is greater than the newer songs, would this confirm that music from earlier periods is better than music now?**

*An issue with this question is the wording "better". "Better" music is very subjective and people will always disagree on what better music is. I would argue that there is very little to no metric that can accurately determine which era of music is "better".*

**(c) Could there be a systematic reason that the 10 songs produced prior to Dec 31, 1999 that your friend selected might be expected to have a higher number of listens?**

*Perhaps this friend is old or enjoys music from the 1900s("I was born in the wrong generation, this is real music" under any Beatles or Queen song) and this would lead him to skew data as they probably have a bias towards believing their preferred music era is better. It is also likely that they know very few modern songs; perhaps they're covers of old songs leaving us with a survivorshipesque type of data.*

### Hints

- Are the 10 songs produced prior to Dec 31, 1999 that your friend selected fairly representative of all songs produced prior to Dec 31, 1999?
- This question is addressing ***survivorship bias***, which will be considered further in the first Tutorial class.