STAT 408: Midterm Exam
Due: March 6 at 5:00 PM
Name:

Please turn in the exam to D2L and include the R Markdown code, SAS code *and either* a Word or PDF file with output. You are welcome to turn in code and output for each question separately. Please verify that all of the code has compiled and the graphics look like you think they should on your Word or PDF file, you are welcome to upload image files directly if they look distorted in the Word or PDF file.

While the exam is open book, meaning you are free to use any resources from class, this is strictly an individual endeavor and **you should not discuss the problems with anyone outside the course instructor including group mates or class members.** The instructor will answer questions related to the data, expectations, and understanding of the exam, but will not fix or troubleshoot broken code.

The first two questions are designed to walk you through the start of the data analysis cycle using a subset of data scraped from the OkCupid website http://www.math.montana.edu/ahoegh/teaching/stat408/datasets/OKCupid_profiles_clean.csv. More details on a larger version of this dataset are available at https://github.com/rudeboybert/JSE_OkCupid/blob/master/okcupid_codebook.txt.

1. (22 points - Data manipulation)

This question will focus on data manipulation using the OKCupid dataset, which can be accessed using the link above. For full credit, include your code and create neat output by using options like `kable()` for tables and writing results in line with r commands.

a. (2 points)

Describe the dataset, what do the columns and variables mean?

b. (4 points)

What proportion of the individuals in the dataset are in their 20's?

c. (4 points)

Compute the average height for each of the different self-reported body_types in the dataset.

d. (4 points)

How many respondents report their ethnicity as 'white', including a mix with other ethnicities?

e. (4 points)

What are the average heights of males and females for respondents that report their ethnicity as pacific islander, including a mix with other ethnicities?

f. (4 points)

Note that the complete dataset also contains paragraphs, such as:

- essay2: I'm really good at
- essay5: The six things I could never do without
- essay9: You should message me if…

Assume that the each essay is stored as a really long character string (technically a vector of length one). Now imagine that you are playing matchmaker for Olympic Champion Mikaela Shiffrin and that she is only interested in profiles that explicitly mention ski terms: `"pow"`, `"gnar"`, and `"ripper"` in essay2, essay5, or essay9.

Note I am not asking you to do this, but please detail how you would sort through these profiles (hint: take a look at the stringr package in R).

2. (24 points - Data Visualization)

Continuing with the OkCupid dataset, we are know going to focus on data visualization.

a. (15 points)

Create a set of **three** graphics to illustrate components of the dataset. These graphs should be compelling and stand-alone with complete titles, labels, and axes. At least one of these figures should have annotation and at least one of the figures should be made with `ggplot2`.

b. (9 points)

Write a set of captions for each figure. The captions should be 2 - 3 sentences and fully describe the figures.

3. (14 points - Word Search Function)

For this problem we are going to write a function to automate a word search. *Note:* for simplicity, we will not include diagonal matches, but rather only the four orthongonal up-down and left-right directions are allowed. Even if you are not able to finish this problem, include your results and thoughts for partial credit.

```r
WordSearch <- function(SearchMatrix, StringList){
  # function to automate a word search
  # ARGS:   SearchMatrix - is a N x N matrix with each entry a character letter
  #         StringList - a list of length p containing strings to search in matrix
  # RETURNS:
  #          A list of length p. If a word is found the list will be a vector of
  #          length 4 with numerical values for the position of the word.
  #          (startrow, startcol, endrow, endcol). If the word is not found
  #          that position in the list should contain 'not found' (character string)
  #
  #          If the SearchMatrix does not meet the criteria above, return an error.
}
```

For example, consider:

```r
mat1 <- matrix(c('b','c','q','a','z','d','t','o','p'), nrow = 3, ncol = 3); mat1
```

```
##      [,1] [,2] [,3]
## [1,] "b"  "a"  "t"
## [2,] "c"  "z"  "o"
## [3,] "q"  "d"  "p"
```

```r
words <- list('bat','top','car'); words
```

```
## [[1]]
## [1] "bat"
##
## [[2]]
## [1] "top"
##
## [[3]]
## [1] "car"
```

Then `WordSearch(mat1, words)` should return

```
## [[1]]
## [1] 1 1 1 3
##
## [[2]]
## [1] 1 3 3 3
##
## [[3]]
## [1] "not found"
```

Finally demonstrate that your function works by evaluating the following statements.

```r
# 6 objects to evaluate
mat1 <- matrix(c('b','c','q','a','z','d','t','o','p'), nrow = 3, ncol = 3)
mat2 <- matrix(1:9, 3, 3)
mat3 <- matrix(c("p", "t", "t", "o", "q", "b", "t", 'a', 'b'), 3, 3)
mat4 <- matrix(c("p", "t", "t", "o", "q", "b", "t", 'a', 'b'))
mat5 <- matrix(c("p", "t", "t", "o", "q", "b", "t", 'a', 'help me'), 3, 3)
mat6 <- matrix(c('s','n','o','w','t','x','a','o','a','c','s','l','t','a','r','p'), 4, 4)

# two lists containing search words
new.words <- list('snow','stat','plow','tarp','pump')
words <- list('bat','top','car')

# Run these 6 commands using your function,
# note, placing error = T while allow R Markdown to compile with errors.
WordSearch(mat1, words)
WordSearch(mat2, words)
WordSearch(mat3, words)
WordSearch(mat4, words)
WordSearch(mat5, words)
WordSearch(mat6, new.words)
```