# STAT 408: Midterm
# Due: March 12 at 10 PM
# Name:

Note: the exam can be turned in as late as March 17th at 8 AM for full credit, but all exams turned in after March 12 at 10 PM require completing an additional problem (# 5).

Please turn in the exam to D2L and include both the R Markdown code *and either* a Word or PDF file. Please verify that all of the code has compiled and the graphics look like you think they should on your 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**. The instructor will answer questions related to expectations or understanding of the exam, but will not fix or troubleshoot broken code.

## 1.

Sudoku is a popular game which players enter numbers into a 9-by-9 matrix, which typically contains some starting digits. To win the game, a player must enter number so each value (between 1 and 9) shows up once and only once in each row, column, and 3-by-3 submatrix. For additional details see the wikipedia page: https://en.wikipedia.org/wiki/Sudoku.

### a. (18 points)

Write a function that:

- Takes a 9-by-9 matrix of numeric values (as a double or integer) as input,
- Returns either 'This matrix is a sucessful Sudoku', or 'This matrix is not a sucessful Sudoku',
- include all necessary documentation and notation for your function and also include errors or warnings for incorrect inputs.

Verify your function by testing it on the following matrices:

```
set.seed(02282017)
mat1 <- matrix(sample(as.character(1:9),81,replace=T),9,9)
mat2 <- replicate(9,sample(9))
mat3 <- matrix(c(1:9,2:9,1,3:9,1:2,4:9,1:3,5:9,1:4,6:9,1:5,7:9,1:6,8:9,1:7,9,1:8),9,9)
mat4 <- matrix(c(5,3,4,6,7,8,9,1,2,
                 6,7,2,1,9,5,3,4,8,
                 1,9,8,3,4,2,5,6,7,
                 8,5,9,7,6,1,4,2,3,
                 4,2,6,8,5,3,7,9,1,
                 7,1,3,9,2,4,8,5,6,
                 9,6,1,5,3,7,2,8,4,
                 2,8,7,4,1,9,6,3,5,
                 3,4,5,2,8,6,1,7,9),
               byrow=T,9,9)
```

### b. (7 points)

Suppose you are given the nearly complete matrix below and want to fill in the remaining nine slots. The matrix does not have any fatal flaws up to this point. Describe a strategy for writing code to complete the soduku for you. Note you do not need to do this, but rather describe the process in detail. Using pseudocode may help describe your thinking.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | | | |
| 2 | 8 | 7 | 4 | 1 | 9 | | | |
| 3 | 4 | 5 | 2 | 8 | 6 | | | |

## 2.

Scrape the data table available at http://www.math.montana.edu/ahoegh/teaching/stat408/course_survey.html. This contains some of the information from the first quiz of the class. For this exercise we are interested in looking at the height of students in the course; however, height is formated differently. All data manipulation should be conducted in R and documented.

### a. (6 points)

Scrape the data table and convert all of the heights to inches. Print a table that shows the heights sorted in decreasing order (tallest to shortest)

### b. (6 points)

Create a data frame that contains counts of students in each of the following height classes.

- less than 65 inches - '< 65 inches'
- greater than equal to 65 and less than 70 inches - '>= 65 inches and < 70 inches'
- greater than equal to 70 inches , '>= 70 inches'

### c. (8 points)

Create a plot of heights of the students in class that follows the design principles for graphics.

## 3.

We have talked about Monte Carlo procedures and the Monty Hall problem throughout the course. This question will expand on those ideas.

### a. (4 points)

Define Monte Carlo procedures.

## b. (4 points)

Describe a process where you might implement a Monte Carlo procedure. This should be an example that has not been covered in class.

## c. (1 point)

Did you enjoy the *Numb3rs* or *21* depiction of the Monty Hall problem more? Why? (see the link on the course webpage)

## d. (16 points)

Consider a generalized case of the Monty Hall problem with N possible doors and M possible cars (this imples N - M goats). Troubleshoot the code below for a Monte Carlo procedure to estimate the probability winning when switching doors after the host opens a door with a goat behind it.

Specifically this function should:

- Take 2 arguments: **num.cars** (numeric or integer) for number of cars and **num.doors** (numeric or integer) for number of doors which must be at least two more than num.cars.
- Return: the winning probability for switching doors and randomly selecting a new door (based on at least 1000 iterations)
- Trip errors for incorrect inputs

```
GenMontyHall <- function(num.cars, num.doors){
  # Function to compute probability of winning a car in generalized
  #    Monty Hall Scenario with options for number of cars and doors
  # ARGS: number of cars (num.cars) as double or integer,
  #       number of doors (num.doors) as double or integer
  # Returns: probability of winning when switching doors
  if (num.cars = num.doors){
    stop('need more doors')
  }
  if (num.goats > num.doors){
    stop('too many goats')
  }
  win.vector <- rep(FALSE,num.iter)
  for (i = 1:num.iter){
    # assign cars to doors
    doors.with.cars <- sample(num.doors, num.cars)
    # assign goats to others
    doors.with.goats <- (1:num.doors)[!(1:num.doors) == doors.with.cars]
    # randomly pick first door
    pick.door <- sample(num.door,1)
    # host reveals a door with goat BUT NOT YOUR DOOR
    goat.door <- sample(doors.with.goats[doors.with.goats !=pick.door],1)
    # pick new door
    new.door <- sample((1:num.doors)[!(1:num.doors) %in% c(pick.door,goat.door)])
    ifelse(new.door == doors.with.cars, win[i] = TRUE, win[i] = False
  }
  return(sum(win.vector))
}
```

```
## Error: <text>:7:16: unexpected '='
```

```
## 6:   # Returns: probability of winning when switching doors
## 7:   if (num.cars =
##                    ^
```

Verify this function works by evaluating the following commands. Note that the option error=T in the R Markdown code allows the document to be compiled, but errors are printed.

```
GenMontyHall(num.cars = 3)
GenMontyHall(num.cars = 3, num.doors = 5)
GenMontyHall(num.cars = 1, num.doors = 10)
GenMontyHall(num.cars = 10, num.doors = 1)
GenMontyHall(num.cars = 1/2, num.doors = 3)
```

# 4.

## a. (4 points)

Find and upload one image that presents data in a misleading way. You may need to revist the R Markdown cheat sheet to upload the image.

Describe the problem with this particular graph.

## b. (4 points)

Find and upload an image that you find compelling. Describe what you like about the image.

## c. (17 points)

Select a data set and create a series of three or more graphs illustrating interesting observations from the data. At least two different types of graphs need to be constructed with a minimum of one graph created using ggplot2. The data set can be one we have used in class or a data set you obtain from elsewhere.

These figures should adhere to the principles we have stated in class and make sure that your graphics can "stand alone". You can add captions via R Markdown if you'd like more than just the title to describe the figures. For full credit your data storytelling results should be *compelling*.

# 5. (5 points)

*If you turn this exam in before 10 PM on March 12, question 5 does not have to be completed and you will receive 5 free points.*

Select 5 functions from the following list and for create a code vignette for each function that:

- Summarizes the function and
- includes examples that demonstrate the various options of the function.

Possible functions include: `seq()`, `rep()`, `lm()`, `t.test()`, `legend()`, `glm()`, `anova()`, `duplicated()`, `apply()`, `replicate()`, `order()`, `array()`, `unlist()`, `which()`, `nrow()`, `dim()`, `geom_ribbon()` `(in ggplot2)`, `geom_rug()` `(in ggplot2)`, `geom_violin()` `(in ggplot2)`