

STAT 408: Final Exam

Name:

Please turn in the exam to D2L. Please include the R Markdown code and a Word or PDF file with output. For any questions completed using SAS turn in the program summary that includes code and output and you can also include the code in your R Markdown code (see examples below for how to do this, but know the code will not be executed). 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 broken code.

1. (18 points - Data Manipulation: R)

Using R, answer the following questions.

```
indian_food <- read_csv('https://raw.githubusercontent.com/stat408/final_exam/master/indian.csv')
```

- a. (3 points) How many different foods are included in this dataset.
- b. (3 points) Compute the reported median `cook_time` for dishes within each course
- c. (3 points) Compute the average total time (`prep_time + cook_time`) required to make dishes within each course
- d. (3 points) How many dishes come from each region?
- e. (3 points) What proportion of the main course meals have `cook_time` of less than 60 minutes?
- f. (3 points) How many meals contain jaggery?

2. (15 points - Data Manipulation in SAS)

Note, the same questions are asked in R for question 2. Using SAS, answer the following questions. For full credit, please submit a SAS file/output and include your SAS code in this document.

The code below will allow you to access the indian dataset in SAS.

```
LIBNAME STAT408 "/courses/d716b355ba27fe300";
```

```
DATA INDIAN;  
    SET STAT408.INDIAN;  
RUN;
```

Recall that SAS also has a procedure called PROC SQL that allows you to use SQL commands to create and modify data sets.

- a. (3 points) How many different foods are included in this dataset.
- b. (3 points) Compute the reported mean `cook_time` for dishes within each course
- c. (3 points) Compute the total time (`prep_time` + `cook_time`) required to make dishes within each course
- d. (3 points) How many dishes come from each region?
- e. (3 points) What proportion of the main course meals have `cook_time` of less than 60 minutes?

3. (19 points - Data Visualization)

Continuing with the Indian food dataset, we are now going to focus on data visualization. These graphs should be compelling and stand-alone with complete titles, labels, axes, and potentially annotations.

a. (8 points)

Create a figure or set of figures in SAS that show the relationship between `prep_time` and `cook_time` as a function of flavor profile and region. You may want to consider using SGPanel <https://documentation.sas.com/?docsetId=grstatproc&docsetTarget=p07rpazajrf2wsn1nzd86x2nw6dv.htm&docsetVersion=9.4&locale=en> to do this.

b. (8 points)

Using R try to recreate (or make a better version) of the figure(s) constructed with SAS.

c. (3 points)

Reflect on making graphics in R and SAS, which do you prefer? Why?

4. (8 points Function)

Using R, write a function called `recipe_finder` that takes the following inputs:

- recipe list: a `data_frame` with columns corresponding to the dish name (called `name`) and a column with ingredients (which are separated by commas) that is called `ingredients`

- ingredient: a character string with the name of an ingredient

the function should return:

- a vector containing the name of recipes that use that ingredient or a string that states “No matches found” if the ingredient is not in any recipes

Don't forget to document your function and include any work, even if incomplete, for partial credit. Demonstrate that your function works by running the three test cases below (you'll need to remove the `eval = F` options from the code chunks below).

```
recipe_finder <- function(){  
  
}
```

```
recipe_list <- indian_food %>% select(name, ingredients)  
recipe_finder(recipe_list = recipe_list, ingredient = 'yogurt')  
  
recipe_finder(recipe_list = recipe_list, ingredient = 'ice cream')  
  
recipe_finder(recipe_list = tibble(name = c('milshake'), ingredients = c('ice cream, milk')),  
              ingredient = 'ice cream')
```