MTH208: Midsem Exam - 2024

Overall Instructions

1. Follow the instructions for each question **exactly**.
2. If your code is not properly commented or horizontal/vertical spacing is missing, points will be deducted.
3. There are three questions with overall 100 marks. Make sure every question's answer is pasted in their respected files in your GitHub.
4. You are only allowed to use the libraries that we have discussed in the course, and no other libraries are allowed to be called.

Questions

1. **Instructions:**
   - Copy and paste your final function for this whole question in the `ans1.R` file in your exam repository.
   - Make sure you DO NOT have command `rm(list = ls())` in the R Script
   - DO NOT put anything else other than the function in the script.

   It's such a small world! (or is it): (30 points)
   It's 2035 and you move to Mumbai for a new job (Senior Data Scientist at Falaana Dhimkaana). Very soon after moving to Mumbai you run into a long forgotten friend from your good old IIT Kanpur days! Your friend says "It's such a small world!". You are no fool. You are the senior data scientist at Falaana Dhimkaana. You know exactly what's going on. You go back to your swanky new apartment and write the code below to prove to your old friend why it is not such a small world.

   First, let's assume the following:

   i. **Locality Size:**
   Let's assume your new locality has a population of 10^5 people.

   ii. **Daily Encounters:**
   Each day, you encounter roughly 500 people while walking around the area (a small percentage of the total population). These encounters are random.

   iii. **Social Circle:**
   You have a network of 150 former IITK friends who live in Mumbai. (Notice that meeting any of these friends would have led one to conclude the same "It's such a small world!" feeling).

   Task:
   Assume equal probability of running into any person in the population. Write an R function that simulates

each day and returns the number of days it would take to meet any old friend among your 150 friends. Call this function `smallWorld()`. Your code should look like:

```
# function has no inputs
smallWorld <- function()
{
  n.population <- 1e5
  n.friends <- 150
  n.encounters <- 500
  ...
  ...
  days <- ...
  ...
  return(days)
}
```

2. **Instructions:**

   - Copy and paste **ONLY** your final function `hsv_convert` into the `ans2.R` file in your repository.
   - Make sure you DO NOT have command `rm(list = ls())` in the R Script.
   - An image of a boat is provided to you for you to test your function on.

   Image and Histograms: (35 points)

   You are given a color image and need to process it to analyze its color components in the HSV (Hue, Saturation, Value) color space. Your task is to write a function that inputs an `imager` image and performs this conversion in R, producing appropriate visualizations.

   i. **Convert the Image to HSV Color Space:**

   - Let the color of every pixel be $[R, G, B]$. For every pixel, find the corresponding $[H, S, V]$ using the formulas below:
     - **Calculate Value (V):** $V = \max(R, G, B)$
     - **Calculate Saturation (S):**

$$S = \frac{V - \min(R, G, B)}{V} \quad \text{if } V \neq 0 \text{ and } S = 0 \text{ if } V = 0$$

     - **Calculate Hue (H):**
       * If $V = R$:

$$H = 60 \times \left(\frac{G - B}{V - \min(R, G, B)} \mod 6 \right)$$

       * If $V = G$:

$$H = 60 \times \left(\frac{B - R}{V - \min(R, G, B)} + 2 \right)$$

       * If $V = B$:

$$H = 60 \times \left(\frac{R - G}{V - \min(R, G, B)} + 4 \right)$$

       * Adjust H to ensure it falls within the range $[0, 360]$ degrees: If $H < 0$, then $H = H + 360$ and If $H \geq 360$, then $H = H - 360$.

   ii. The function should ONLY plot three histograms (in a row), one for each Hue, Saturation, and Value, with all axes labeled and appropriate titles. The function shouldn't return anything. The function should be called `hsv_convert()` and should look something like this:

```
# img is an imager image
hsv_convert <- function(img)
{
  ...
  par(mfrow = c(1,3))
  ...
}
```

3. **Instructions:**

   - Copy all relevant code and paste in the file `ans3.R`. Make sure to include all lines of code that are required to produce the tibble exactly.
   - Make sure you DO NOT have command `rm(list = ls())` in the R Script.

   Web Scraping: (35 points) Below is a link that has the list of the most coveted Statistics and Probability Journals. For those interested in research in Statistics, such a list is very useful to have

   https://www.scimagojr.com/journalrank.php?area=2600&category=2613

   Scrape and clean the complete table of the 268 journals and gather the following information.

   - The Serial Number (int or dbl) (1:268)
   - Title (chr): the name of the journal
   - Type (chr): the type of journal
   - SJR (int or dbl): the SJR score
   - Q1-Q4: The Quality index (in green, yellow, red, on the website)
   - H-index (int or dbl)
   - Total Docs (int or dbl)
   - Total Docs 3 years (int or dbl)
   - Total Refs (int or dbl)
   - Total Cites (int or dbl)
   - Citable Docs (int or dbl)
   - Cites/Doc (dbl)
   - Ref/doc (dbl)
   - % Female (dbl)
   - Country (chr)

   Save the required tibble in `stat_journals`; this should be a 268×15 tibble. Your last line of the code should be:

```
stat_journals <- ...
```