Business Analytics 1 – Assignment 2 (50 points)

Submission Deadline: Sunday, the 4th of March, at midnight

The first 6 questions relate to the `FlagsData.txt` dataset from the UCI Machine Learning Repository. It contains details of various nations and their flags. More information can be found here: http://archive.ics.uci.edu/ml/datasets/Flags

1) First, read in `FlagData.txt` and store it as `flags`.
2) The "`landmass`" variable in our dataset takes on integer values between 1 and 6, each of which represents a different part of the world. Use the `table()` function to see how many flags/countries fall into each group. Please also apply the same procedure to variable "`religion`". (2 points)
3) Use the `split()` function to partition the variable "`name`" into subgroups, broken down by "`landmass`" and "`religion`"; save the output in "`landmass_religion_subgroups`". (2 points)

   According to the output:

   - How many subgroups have been generated? (2 points)
   - Which country (or countries) in South America are labelled "Other Christian"? (3 points)
   - Which subgroup (or subgroups) have the most countries? (3 points)
   - How many countries are there in the largest subgroup (or subgroups) you have just identified? (2 points)

4) Please find the minimum, 50% cut point and maximum population values (in round millions) for countries within each landmass. (2 points)
5) Variable "`zone`" indicates the geographic quadrant, based on Greenwich and the Equator. Please find out which landmass has all its countries in one zone, and which landmass has countries in every zone. You are not allowed to use the table() or CrossTable() function. (4 points)

6) Please create a function, `findcountry()`, which takes five arguments:

   - Number of bars
   - Number of stripes
   - Number of circles
   - Number of crosses
   - Number of quarters

   and returns the names of countries whose flags meet the given criteria. If there is no flag satisfying the given criteria, your function should print out a warning message. (5 points)

To answer questions 1 to 6 listed above, please first create an R script in RStudio and name it `Question1to6.R`, and then do the following:

- Write down your answers in the R script as comments (*i.e.*, all textual explanation should be prefaced with ##).
- For each question, write down the R code or function you have used or created (not as comments), and make sure that every piece of conclusion you have made can be verified by the code you have provided.

There are three more questions on the next page.

The next three questions relate to the `outcome-of-care-measures.csv` dataset, which contains information about 30-day mortality and readmission rates for heart attacks, heart failure, and pneumonia for over 4,000 hospitals. A description of the variables in the above files is in the included PDF file named `description-for-outcome-of-care-measures.pdf`.

To answer questions 7 to 9 listed below, please first create an R script in RStudio and name it `Question7to9.R`, and then input the following codes first:

- Please first read in `outcome-of-care-measures.csv` and store it as `data`:

```
data <- read.csv("outcome-of-care-measures.csv",
        header = TRUE, stringsAsFactors = FALSE)
```

- Because we originally read the data in as character (by specifying `stringsAsFactors = FALSE`) we need to coerce the following three columns to be numeric (with `as.numeric()` function). Please code in the following commands:

```
data[, 11] <- as.numeric(data[, 11])
data[, 17] <- as.numeric(data[, 17])
data[, 23] <- as.numeric(data[, 23])
```

  You may get a warning message saying

```
NAs introduced by coercion
```

  This is OK.

Having input the above codes, please answer questions 7, 8 and 9 listed on the next page.

7) Write a function called `find_best_rate()` that takes two arguments: the 2-character abbreviated name of a state and an outcome name. The outcome name can be one of "heart attack", "heart failure" or "pneumonia". This function should return the best (*i.e.*, lowest) 30-Day Death (Mortality) Rate for the specified outcome in the specified state.

   The function should check the validity of its arguments. If an invalid state name is passed to it, the function should return the exact message "invalid state". If an invalid outcome name is passed to it, the function should return the exact message "invalid outcome". (10 points)

8) Write another function called `find_best_hospital()` that takes the same two arguments as above: the 2-character abbreviated name of a state and an outcome name. The outcome name can be one of "heart attack", "heart failure" or "pneumonia". This function should return the name(s) of the hospital that has the best (i.e., lowest) 30-Day Death (Mortality) Rate for the specified outcome in the specified state.

   If there is a tie for the best hospital for the specified outcome in the specified state, then all their names should be returned by this function.

   This function should also check the validity of its arguments. If an invalid state name is passed to it, the function should return the exact message "invalid state". If an invalid outcome name is passed to it, the function should return the exact message "invalid outcome". (10 points)

9) By using your `find_best_hospital()` function, for each of the three outcomes, please find the hospitals' names and the state name if they tie for the best hospital in that state. (5 points)