STAT40730/STAT40620
Data Programming with R.
Assignment 1.

Instructions

- This assignment is due at the end of week 6, and is worth 5% of your final grade.
- You should submit your assignment to the 'Assignment 1' assignment object in Blackboard.
- You should submit two files only:
  1. a 2-page (maximum) document (in pdf format only) which should contain answers to the questions below shown in bold
  2. a single R script file detailing the commented code you used to obtain your answers.
- The marks available for each question are shown in brackets.
- Assignment 1 is broken up into 3 tasks: data manipulation, analysis, and creativity.
- You will have to learn and discover some new functions. Use `help()` and `help.search()` to find what you need. There are some hints at the end of this document.
- Make sure your R script file contains all the commands used to complete the tasks, and that it includes comments (starting with #) so that the file is readable.

Assignment 1 analyses US baby names from 1880-2010 to determine patterns in naming conventions. The data are available in the assessment object in a file called 'names.zip'. Download the file to a specific folder on your machine, called Assignment1, say. If you unzip the file you'll see that each year is a comma-separated file with 3 columns – name, sex, and number of births.

Task 1: manipulation

1. Set your working directory to the Assignment1 folder where you saved the data (use the function `setwd()` or in R studio go to the 'Session' menu item and choose 'Set working directory' and then 'Choose directory' – more on this in week 4 lecture materials). Load in the file from 1950 using `read.table` or `read.csv`. (2 marks)

2. Produce appropriate commands to answer the following questions:

   (a) **According to these data, how many children were born in 1950?** (2 marks)

   (b) **Which were the 10 most popular names for each sex?** (2 marks)

   (c) **Are there any names in the data set that were only given once?** (Note: the readme file suggests names given less than 5 times should have been removed, but you should check this.) (2 marks)

3. Load in the file from 2010 in a separate data frame. **Which names had the biggest rise/fall compared to 1950?** (7 marks)

4. Write some R code which loads in all of the files for each year and merges them into a single data frame with 4 columns: year, name, sex, and number of births. (10 marks)

Task 2: analysis

1. Produce a table of the popularity of your name over each year. (If your name is not in the data set choose a similar name which is in the data set.) **What year was the maximum for your name?** (2 marks)
2. Create a table showing the total births by sex and year. **Do males or females tend to have higher birth rates?** (6 marks)
3. Create a table of the frequency of different last letters in names for years 1910, 1960 and 2010 for males and females. **Which last letter(s) stand out as having the biggest increase/decrease?** (10 marks)
4. **Which are the most popular palindromic names?** Calculate the proportion of palindromic names per year. **Are such names on the increase?** (12 marks)

Task 3: creativity

Do something interesting with these data! Create a table (or even a plot if you have got as far as Lecture 5) which shows something we have not discovered above already. Make sure to include all R code in your script file and outline your findings in your pdf document. (15 marks)

Hints

- If you find your computer is too slow at doing some of the calculations in tasks 1 and 2 then try running every 10th year instead of every year.
- A non-exhaustive list of some useful functions is given below.

Function	Details
`read.csv`	Reads in a comma separated values file (note default is `header=TRUE`) similar to `read.table`.
`subset`	Subsets a data frame
`order`	Finds the rank order of a vector
`merge`	Merges together two data frames
`unlist`	Turns a list into a data frame
`lapply`	Applies a function to the individual tags in a list
`do.call`	Applies a function to all the tags in a list
`rbind/cbind`	Binds together columns or rows of a matrix
`aggregate`	Performs a function broken down by a list in a data frame
`strsplit`	Divides up a string into parts
`tolower`	Converts a string into lower case
`rev`	Reverses the order of a vector
`Vectorize`	Takes a non-vectorised function and vectorises it
`unique`	Finds the number of unique values in a vector
`head/tail`	Prints out the top or bottom 6 rows of a matrix