Homework 9 – Due November 11, 12 am

The total points on this homework is 250. Points are reserved for readability of code, punctuation and clarity of presentation.

1. Write a function which takes 2 arguments n and k which are positive integers. It should return the $n \times n$ matrix:

$$\begin{pmatrix} k & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & k & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & k & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & k & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & k & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & k \end{pmatrix}$$

   Call the function defined above for $n = 6$ and $k = 5$, and provide the matrix you obtain. [*15 points*]

2. Consider the continuous function

$$f(x) = \begin{cases} x^2 + 2x + 3 & if\ x < 0 \\ x + 3 & if\ 0 \le x < 2 \\ x^2 + 4x - 7 & if\ 2 \le x \end{cases}$$

   Write a function tmpFn which takes a single argument xVec. The function should return the vector of values of the function $f(x)$ evaluated at the values xVec. Plot the function $f(x)$ for $-3 < x < 3$. [*15 points*]

3. *Greatest common divisor of two integers* The greatest common divisor (gcd) of two integers m and n can be calculated using Euclid's Algorithm: Divide m by n. If the remainder is zero, the gcd is n. If not, divide n by the remainder. If the remainder is zero, then the previous remainder is the gcd. If not, continue dividing the remainder into previous remainder until a remainder of zero is obtained. The gcd is the value of the last nonzero remainder. Write a function `gcd(m,n)` using a while loop to find the gcd of two integers m and n. [*20 points*]

4. *eQTL mapping.* (The following problem was suggested by Professor Dan Nettleton.) Write a function `order.matrix` which takes in a matrix `x` and returns a matrix containing the row and column indices of the sorted values of `x`. Test this function on a 4×3 matrix of independent χ^2_1 pseudo-random deviates. [*10 points*]

5. *Polar representation of a number.* Let $\boldsymbol{x} \in I\!R^p$. The polar representation of $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$ is given by $(R, \theta_1, \theta_2, \ldots, \theta_{p-1})$, where

$$\begin{aligned} x_1 &= R\cos\theta_1 \\ x_2 &= R\sin\theta_1\cos\theta_2 \\ x_3 &= R\sin\theta_1\sin\theta_2\cos\theta_3 \\ \ldots &= \ldots \\ x_{p-1} &= R\prod_{i=1}^{p-2}\sin\theta_i\cos\theta_{p-1} \\ x_p &= R\prod_{i=1}^{p-1}\sin\theta_i, \end{aligned}$$

   where $0 \le R < \infty, 0 \le \theta_1 < 2\pi$ and $0 \le \theta_i < \pi$ for $i = 2, 3, \ldots, p-1$.

(a) Write a function `polaroid` which takes in an arbitrary p-dimensional vector $\boldsymbol{x}$ and provides its polar representation as a vector, with the first element as R and the remainder being $\theta_1, \theta_2, \ldots, \theta_{p-1}$. [*20 points*]

(b) Write a function `normalize` which takes in a matrix and returns its normalized form: *i.e.* the matrix with rows scaled such that the sum of squares of each row is equal to 1. [*10 points*]

(c) Obtain a 1000×5 matrix y of $N(0,1)$ pseudo-random deviates. Use apply and `normalize` to obtain the normalized values. Call this matrix `z`. We test whether the `columns of z` are uniform on $U(-1,1)$. One may test whether a sample `x` $\sim U(-1,1)$ using `ks.test(x,"punif",min=-1,max=1)` where `punif` represents the cumulative distribution function of the uniform over range (-1,1). Summarize your results. [*20 points*]

(d) Obtain polar representations of y using your function `polaroid` and test whether $R^2 \sim \chi^2_5$ distribution. Provide a page of histograms or boxplots of $\theta_1, \theta_2, \theta_3, \theta_4$. Test whether these are from the uniform distributions on their respective ranges *i.e.*, $[0, 2\pi)$ for θ_1 and $[0, \pi)$ for $\theta_2, \theta_3, \theta_4$. [*20 points*]

6. The following distribution has the density function:

$$f(x;\theta) = \frac{1-\cos(x-\theta)}{2\pi} \quad \text{for} \quad 0 \le x \le 2\pi, \quad -\pi < \theta < \pi$$

For an observed random sample $x_1, x_2, \ldots, x_n$ from this distribution, the log likelihood is seen to be

$$\ell(\theta) = -n\log 2\pi + \sum_{i=1}^{n} \log\{1-\cos(x_i-\theta)\}$$

Suppose that $(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96, 2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52, 2.50)$ is an observed random sample from the above distribution.

(a) Plot the log likelihood $\ell(\theta)$ in the range $-\pi < \theta < \pi$. [*15 points*]

(b) Use the R function `optimize()` to find the maximum likelihood estimate of θ. [*10 points*]

(c) Use function `newton()` in the class handout to find the maximum likelihood estimate of θ by solving $\ell'(\theta) = 0$ using the starting value of $\theta_{(0)} = 0$. [*15 points*]

(d) What happens if you use $\theta_{(0)} = -2.0$ and -2.7, respectively, as starting values? Explain why. [*5 points*]

Turn in the plot, any functions you write, function calls, and the results.

P.S. To help you out with the necessary derivatives, I derived them below but you need to check them out!

$$\frac{\partial \ell}{\partial \theta} = -\sum_{i=1}^{n} \frac{\sin(x_i-\theta)}{1-\cos(x_i-\theta)}$$

$$\frac{\partial^2 \ell}{\partial \theta^2} = -\sum_{i=1}^{n} \frac{1}{1-\cos(x_i-\theta)}$$

7. *Regression to the mean.* Consider the following very simple genetic model in which a population consist of equal number of two sexes: male and female. At each generation mean and women are paired at random, and each pair produces exactly two offspring, one male and one female. We are interested in the distribution of height from one generation to the next. Supposing that the height of both the children is just the average of the heights of their parents, how will the distribution of height change across generations?

(a) Represent the heights of the current generation as a dataframe with two variables, M and F for the two sexes. Randomly generate a population at generation 1, as for males with $X_1, X_2, \ldots, X_{100} \sim N(125, 25^2)$, and for females $X_1, X_2, \ldots, X_{100} \sim N(125, 15^2)$. [*10 points*]

(b) Take the dataframe from (a) and randomly permute the ordering of men. Men and women are then paired according to rows, and heights for the next generation are calculated by taking the mean of each row. The function should return a dataframe with the same structure, giving the heights of the next generation. You will need to use the `sample(x, size = n)` that will return a random sample of size n, from the vector x. You will also need to use the `apply()` function. [*10 points*]

(c) Use the above function to generate nine generations, then use `ggplot2` to facet histograms to plot the distribution of male heights in each generation. This is called regression to the mean. (*Hint: Instead of using* `facet_grid`*, you will need to use* `facet_wrap(,nrow = 3)`*, in order to create* 3×3 *histograms.*) [*10 points*]

8. The program `mmclustering` available at http://math.univ-lille1.fr/ wicker/softwares.html (*but note: you do not need to download or run any such program*) provides a partitioning of observations into different groups. However, the output is provided in a form which makes it difficult to easily do further analysis in R. In this exercise, we will write a function that will take the output (from a given file) and write out the classification for each observation as a vector.

   The files provided in `Iris1.out` and `Iris2.out` contains results from grouping the iris dataset into a certain number of categories. The file has the following lines:

   - The first line in the file contains the number of clusters which is given by the phrase `Number of clusters` followed by a space, a colon `:`, a space, and then an integer (indicating the number of clusters). For example: `Number of clusters :  4`
   - The second line is a blank.
   - The third line contains the id of the first cluster (always designated by `0`) and designated by `Cluster` followed by a space, then `0`, followed by a space, semi-colon, space and the character string `size` followed by an equality and an integer which denotes the size of the cluster). For example, `Cluster 0; size=37`
   - The next number of lines (which should be the same as the one given in the cluster size in the third line) contain the observation indices which belong to that group. This group of indices ends with a blank line.
   - The following lines repeat the same for the second cluster (given by `Cluster 1`). The process continues till all the cluster memberships are listed.

   (a) The objective here is to write a function which will read in one of the files above and provide a vector of length equal to the sum of the cluster sizes which will contain the group indicators of the observations in the total population. To do this, we can use the `readLines` function to read the file line-by-line (note that in that case, each line will be read in as a character string). Then, we can use (multiple) string-matching methods to parse the first line to obtain the number of clusters. The second line is a blank line. Indeed, the first line after every blank line contains the size of the cluster (with memberships following the next line). Again, we will use string-splitting and matching techniques (*e.g.* `strsplit`) to obtain the cluster size. The next lines are to be converted (from the character string) to integers. Write the above function. [*35 points*]

   (b) Cross-tabulate the results of a call to the above function on the file `Iris1.out` and the file `Iris2.out`. [*10 point*]