R Programming: Worksheet 3

1. **Basic apply and similar**

   (a) Using `replicate`, generate a list of 50 random datasets, each consisting of 10 independent t_5 distributed random variables. *Have a look at `rt()`*

   (b) Using `mapply`, generate an array with 10 rows and 50 columns, where each column consists of 10 independent t distributed random variables, where column i contains t_i distributed random variables.

   (c) Using `lapply`, generate a list of length 20, where the ith entry in the list is the sequence of numbers $1, \ldots, i$.

   (d) Generate the following random matrix `X`:

   ```
   > set.seed(2023)
   > X <- matrix(rexp(200), 20, 10)
   ```

   Using `apply`, find the smallest entry in each column of `X`.

   (e) Look at the data frame `CO2` (this is preloaded into `R`). How would you determine which columns are numeric? *Check out `is.numeric()`*

2. **GTEx**

   (a) Download the example GTEX gene expression data `GTEx_analysis.txt.gz`. Read it into R using `read.table`, and use `system.time` to record how long it took. Note that this will throw errors with default options, and you'll need to figure out how to modify those to load the document.

   (b) If necessary, install the package `data.table`, and read in `GTEx_analysis.txt.gz` using `data.table::fread`. Again use `system.time` to record how long it took. Which one was faster?

   (c) Make a subset of the GTEX data removing the Bladder column. Compare how long it takes to save this to disk using `save`, versus writing it using `data.table::fwrite` *Use tempfile() to get temporary file names, or just invent your own relevant filenames*

3. **Insect spray experiment**

   I have made available a dataset called `sprays.txt`. The data represent insect counts in agricultural experiments treated with different insecticides. Save the file onto your local drive.

   (a) Read the data into `R`. Take a look at its contents using `head` and `str`.

   (b) Find the mean number of insects for each different experimental unit, first using vector operations, and next using `tapply`

   (c) Use `tapply()` to find the upper and lower quartiles of the counts broken down by spray type. *Check out `quantile()`*

4. **More GTEx**

   Here we are going to look at the GTEx data in more detail, and explore it using some of the `apply` functions.

   This GTEx data, once loaded in, yields a data frame (note, the default when using `fread` is a `data.table`, to make a data frame, use `fread(, data.table = FALSE)`, or `data.frame(fread())`). Each row is the result for a gene. Each column is either the gene name (`Name`, a technical name, and `Description`, a more human readable form), and the subsequent columns list the average gene expression level for a sample of people for that gene in that tissue.

   (a) Read the GTEx data from question 2 back in. Using the `apply` function, calculate the tissue the expression is highest in for each gene. Then summarize these results across all genes. For genes where there is a tie for the highest expression, choose at random. Which tissue most frequently has the highest gene expression for a gene?

   (b) Using the `sapply` function, summarize the data, by calculating for each tissue the mean, standard deviation, median, and 5^{th} and 95^{th} percentile values.

   (c) Here we're going to look for genes that have the same gene expression profile as other genes, *i.e.* they have a similar pattern of expression across multiple tissues. Try calculating the squared distance between the profile for *APOB* and all other genes, using the GTEx data in its current form with the `apply` function, as well as the transposed form using the `sapply` function. Be careful throughout about variable types to minimize type conversion. Which way is faster? In any case, print the human readable names for the top 10 matches. Do any have similar names to *APOB* (suggesting similar function)? *Optional: By plotting, or otherwise, determine the dominant tissue(s) that drives the similarity in expression for these matches*