SS2864B, 2020

Assignment #2 due to February 12, 2020

Instructions Submit an electronic version (pdf, words, etc) of your solutions (appropriately annotated with comments, plots, and explanations) to owl.

1. One of the most important tasks in using R is to read data from external sources into R objects, so that you can perform calculations and produce graphics. If you have a data set that you're interested in, try reading it into R using the **scan** and **read.table** functions. If you don't have a file of data to read, you can use the **write** and **write.table** (opposite of **scan** and **read.table** respectively) to produce one. Before you start, use ¿ **help(scan)** to find the detailed usage of the function **scan**, the same for other functions. Once you know how to use these functions, try

   (a) Create a vector and a data.frame objects in R. Make sure to have floating or real numbers in your objects.

   (b) Use **write** and **write.table** to export the two objects into two files.

   (c) Use **scan** and **read.table** to import data sets in the two files back into R but two different object names.

   (d) Verify that they are the same.

2. Use R to identify the elements as well as their index(s) of the sequence $\{2^1, 2^2, \ldots, 2^{15}\}$ that exceed the corresponding elements of the sequence $\{1^3, 2^3, \ldots, 15^3\}$.

3. R functions **dump** and **save** can be used to save any R objects into files. Use ?dump and ?save to find out how to use them. Please list the similarity and difference of **dump** and **save** functions. Create a few R objects (more than one object) and dump (and save) them into (two separate) file. Find a way to rename dumped (saved) objects. Then use R function **source** (and **load**) to source (load) the file you just created. Again use ?source and ?load to find out how to use them. Check if you get the same R objects back (need to compare renamed R objects with ones sourced (loaded) from a file.

4. The R function **cut** can be used to cut numerical values into many levels. Please investigate its usage and use it to cut bmi in the dataset **Pima.te** (from library(MASS)) into many categories. First use the values in http://en.wikipedia.org/wiki/Body_mass_index to do the cut and the R function **table** to find their frequencies and plot it. Then cut again with only three categories underweight, normal, overweight and redo the frequency and plot.

5. Consider the **pressure** data frame. There are two columns - temperature and pressure.

   (a) Construct a scatterplot with pressure on the vertical axis and temperature on the horizontal axis. Are the variables related linearly or nonlinear?

   (b) The graph of the following function passes through the plotted points reasonably well:

   $$y = (0.168 + 0.007x)^{20/3}.$$

The differences between the pressure values predicted by the curve and the observed pressure vales are called residuals. Here is a way to calculate them:

```
residuals <- with(pressure, pressure-(0.168+0.007*temperature)^(20/3))
```

Construct a normal QQ plot (adding a qqline) of these residuals and decide whether they normally distributed or weather they follow a skewed distribution.

(c) Now, apply the power transformation $y^{3/20}$ to the pressure data values. Plot these transformed values against temperature. Is a linear or nonlinear relationship evident now?

(d) Calculate residuals for the difference between transformed pressure values and those predicted by the straight line. Obtain a normal QQ plot (adding a qqline), and decide whether the residuals follow a normal distribution or not.

6. Consider the **pressure** data frame again.

(a) Plot pressure against temperature, and use the following command to pass a curve through these data:

```
curve((0.168+0.007*x)^(20/3), from = 0, to = 400, add = TRUE)
```

Note: You can use **lines** instead of **curve**.

(b) Now, apply the power transformation $y^{3/20}$ to the pressure data values. Plot these transformed values against temperature. Use the **abline** function to pass a straight line through the points.

(c) Add a suitable title to the graph. Also add a legend to indicate the straight line.

(d) Re-do the above plots, but use the **mfrow** function to display them in a 2x1 layout on the graphics page. Repeat once again using a 1x2 layout.