Stat 604
Assignment 02 R

Prerequisite:

This assignment does not require any prior knowledge or experience with the R language. The only requirement is that R be installed on your computer.

Specific Instructions for this Assignment:

For this assignment you will be typing various commands into the R console, as directed below, to become familiar with the R language syntax. You will be asked to save the plots that are produced and submit them to WebAssign. At the end of the exercise you will also save the contents of the console for submission to WebAssign. This is NOT the same as saving the workspace. You will need to save or convert all files to PDF before submitting them to WebAssign. Use your Net ID (not a number) in your file names according to the following pattern: FKincheloe_HW02_console.pdf, replacing FKincheloe with your own Net ID. Many word processing softwares, including Microsoft Word, have the capability of saving files to PDF. If you do not have a way to convert documents to PDF, you can download the free Nitro PDF Reader from download.cnet.com. There are separate "questions" on WebAssign; one for each of the files you must download for this assignment. Make sure each file is properly uploaded to the appropriate question. If the upload is successful, you should see the name of the file listed as a hyperlink below the browse box. You must SUBMIT the assignment after you have finished the download in order for the files to be available for the grader.

1. Open a blank Word document or use your own word processing software. You can also use WordPad on Windows or Pages on a Mac. In the steps below, you will be executing a series of commands that were taken from "Appendix A – A sample session" in the document ***R-intro.pdf*** (An Introduction to R) from R manuals that are included with your R installation (also on eCampus) Many of these commands will cause a graphics(Quartz on Mac) window to open. You will be instructed to copy the graphics and paste it into your blank Word document. If you are using Windows, click on the R Graphics Device window then click **File** -> **Copy to the Clipboard** -> **as a Bitmap…** to copy the graphics. If you are using a Mac, click on the Quartz window then click **Edit** -> **Copy** or press the Command and C keys to copy the graphics. Close the graphics window after you have pasted the graphics in to your Word document.
2. (It is recommended that you read over the entire assignment first before you begin.) Since this is not intended to be an exercise in typing, you may copy the commands from this PDF document and paste them into the R console. You are not expected to understand the commands that you are entering. Some of them are beyond the scope of this course. The comments included beside or below the commands will give you a brief overview of what each command does. This exercise is only intended to introduce you to the syntax and behavior of the R system. However, you are expected to pay close attention to the response from R after you execute each command and examine the output. Feel free to experiment with and investigate any of the commands that you find interesting. Please use a separate instance of R for your experimentation so your homework submission will be consistent with what the grader is using as a key.

3. Launch R on your computer. In the R console type a command similar to the one shown below replacing the name shown with your own name. Include the parenthesis. (Press the Enter key after typing each command in the assignment to execute the command.)
   (Name <- "Faron Kincheloe")
4. Enter and execute the command: **Sys.time()**
5. Enter or paste into the R console and execute the commands listed below. The R commands will be in **bold** font. (Do not include the text in normal font.):

   help.start() Start the HTML interface to on-line help (using a web browser available at your machine). You should briefly explore the features of this facility with the mouse.

   x <- rnorm(50) Generate two pseudo-random normal vectors of x- and y-coordinates.
   y <- rnorm(x)

   plot(x, y) Plot the points in the plane. A graphics window will appear automatically.
6. Copy and paste the graphics into your Word document using the method explained in step 3.
7. Enter and execute the next series of commands. NOTE: When you execute the attach function it is normal to receive the message that the object is masked. This occurs because the workspace and dummy data frame both contain an object named x.
   ls() See which R objects are now in the R workspace.

   rm(x, y) Remove objects no longer needed. (Clean up).

   x <- 1:20 Make x = (1; 2; : : : ; 20).

   w <- 1 + sqrt(x)/2 A 'weight' vector of standard deviations.

   dummy <- data.frame(x=x, y= x + rnorm(x)*w)
   dummy Make a data frame of two columns, x and y, and look at it.

   fm <- lm(y ~ x, data=dummy)
   summary(fm) Fit a simple linear regression and look at the analysis. With y to the left of the tilde, we are modelling y dependent on x.

   fm1 <- lm(y ~ x, data=dummy, weight=1/w^2)
   summary(fm1) Since we know the standard deviations, we can do a weighted regression.

   attach(dummy) Make the columns in the data frame visible as variables.

   lrf <- lowess(x, y) Make a nonparametric local regression function.

   plot(x, y) Standard point plot. (DO NOT copy yet. Watch additional objects be added by commands below.

   lines(x, lrf$y) Add in the local regression.

   abline(0, 1, lty=3) The true regression line: (intercept 0, slope 1).

abline(coef(fm)) Unweighted regression line.

abline(coef(fm1), col = "red") Weighted regression line.

8. Now that you have executed the lines command and the three abline commands copy the second graphics image to your Word document. Close the graphics window after you have pasted it.
9. Execute the commands below. Copy and paste the results of the third plot command and close the graphics window:
   detach() Remove data frame from the search path.

   plot(fitted(fm), resid(fm), xlab="Fitted values",ylab="Residuals",main="Residuals vs Fitted")
   A standard regression diagnostic plot to check for heteroscedasticity. Can you see it?
10. Copy and paste the results of the qqnorm command below then close the graphics window:
    qqnorm(resid(fm), main="Residuals Rankit Plot") A normal scores plot to check for skewness, kurtosis and outliers. (Not very useful here.)
11. **rm(fm, fm1, lrf, x, dummy)** Clean up again.
12. The next section will look at data from the classical experiment of Michaelson and Morley to measure the speed of light. This dataset is available in the morley object, but we will read it to illustrate the read.table function:
    filepath <- system.file("data", "morley.tab" , package="datasets")
    filepath Get the path to the data file.

    file.show(filepath) Look at the file.

    mm <- read.table(filepath)
    mm Read in the Michaelson and Morley data as a data frame, and look at it. There are five experiments (column Expt) and each has 20 runs (column Run) and sl is the recorded speed of light, suitably coded.

    mm$Expt <- factor(mm$Expt) Change Expt and Run into factors.
    mm$Run <- factor(mm$Run)

    attach(mm) Make the data frame visible at position 3 (the default).

    plot(Expt, Speed, main="Speed of Light Data", xlab="Experiment No.") Compare the five experiments with simple boxplots.
13. Copy and paste the box plots then close the graphics window.
14. Execute the commands below and both of the two contour functions before pasting the graphics then close the graphics window:
    fm <- aov(Speed ~ Run + Expt, data=mm)
    summary(fm) Analyze as a randomized block, with 'runs' and 'experiments' as factors.

    fm0 <- update(fm, . ~ . - Run)
    anova(fm0, fm) Fit the sub-model omitting 'runs', and compare using a formal analysis of variance.

    detach()

rm(fm, fm0) Clean up before moving on.

We now look at some more graphical features: contour and image plots.
x <- seq(-pi, pi, len=50)
y <- x x is a vector of 50 equally spaced values. y is the same.

f <- outer(x, y, function(x, y) cos(y)/(1 + x^2)) f is a square matrix, with rows and columns indexed by x and y respectively, of values of the function cos(y)=(1 + x2).

oldpar <- par(no.readonly = TRUE)
par(pty="s") Save the plotting parameters and set the plotting region to "square".

contour(x, y, f)
contour(x, y, f, nlevels=15, add=TRUE) Make a contour map of f; add in more lines for more detail.

15. Execute the commands below. Copy and paste the output of the contour function then close the graphics window:
fa <- (f-t(f))/2 fa is the "asymmetric part" of f. (t() is transpose).

contour(x, y, fa, nlevels=15) Make a contour plot
16. Copy and paste the results of each of the two image functions below separately.
par(oldpar) restore the old graphics parameters.

image(x, y, f) Make some high density image plots

image(x, y, fa)
17. Copy and paste the plot of the circle:
objects(); rm(x, y, f, fa) clean up before moving on.

R can do complex arithmetic, also.
th <- seq(-pi, pi, len=100)

z <- exp(1i*th) 1i is used for the complex number i.

par(pty="s")
plot(z, type="l") Plotting complex arguments means plot imaginary versus real parts. This should be a circle.
18. Execute both the plot and the lines functions before pasting the results to your Word document:
w <- rnorm(100) + rnorm(100)*1i Suppose we want to sample points within the unit circle. One method would be to take complex numbers with standard normal real and imaginary parts . . .
w <- ifelse(Mod(w) > 1, 1/w, w). . . and to map any outside the circle onto their reciprocal.

plot(w, xlim=c(-1,1), ylim=c(-1,1), pch="+",xlab="x", ylab="y")
lines(z) All points are inside the unit circle, but the distribution is not uniform.
19. After executing the final plot and lines functions, paste the results to your document:
w <- sqrt(runif(100))*exp(2*pi*runif(100)*1i)

plot(w, xlim=c(-1,1), ylim=c(-1,1), pch="+", xlab="x", ylab="y")

lines(z) The second method uses the uniform distribution. The points should now look more evenly spaced over the disc. If you have completed this exercise successfully, you should have 12 different graphic images.

20. Save the Word document with the naming pattern FKincheloe_HW02_graphics.pdf replacing FKincheloe with your Net ID.
21. With the console window selected, use **File** -> **Save to File** (**Save As** on a Mac) from the R GUI menu to save the contents of the console to a text file. Convert the console document to PDF for submission to WebAssign. Use the name described at the beginning of this assignment.
22. Upload each of the files to the appropriate question on WebAssign and submit the assignment.