3. Worksheet: Basic R

Anna Werkowski; Z620: Quantitative Biodiversity, Indiana University

17 January, 2023

OVERVIEW

This worksheet introduces some of the basic features of the R computing environment (http://www.r-project.org). It is designed to be used along side the **3. RStudio** handout in your binder. You will not be able to complete the exercises without the corresponding handout.

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) with your name.
2. Complete as much of the worksheet as possible during class.
3. Use the handout as a guide; it contains a more complete description of data sets along with examples of proper scripting needed to carry out the exercises.
4. Answer questions in the worksheet. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio (color may vary if you changed the editor theme).
5. Before you leave the classroom today, it is *imperative* that you **push** this file to your GitHub repo, at whatever stage you are. Ths will enable you to pull your work onto your own computer.
6. When you have completed the worksheet, **Knit** the text and code into a single PDF file by pressing the `Knit` button in the RStudio scripting panel. This will save the PDF output in your '3.RStudio' folder.
7. After Knitting, please submit the worksheet by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (**3.RStudio_Worksheet.Rmd**) with all code blocks filled out and questions answered) and the PDF output of `Knitr` (**3.RStudio_Worksheet.pdf**).

The completed exercise is due on **Wednesday, January 18th, 2023 before 12:00 PM (noon)**.

1) HOW WE WILL BE USING R AND OTHER TOOLS

You are working in an RMarkdown (.Rmd) file. This allows you to integrate text and R code into a single document. There are two major features to this document: 1) Markdown formatted text and 2) "chunks" of R code. Anything in an R code chunk will be interpreted by R when you *Knit* the document.

When you are done, you will *knit* your document together. However, if there are errors in the R code contained in your Markdown document, you will not be able to knit a PDF file. If this happens, you will need to review your code, locate the source of the error(s), and make the appropriate changes. Even if you are able to knit without issue, you should review the knitted document for correctness and completeness before you submit the Worksheet. Next to the `Knit` button in the RStudio scripting panel there is a spell checker button (`ABC`) button.

2) SETTING YOUR WORKING DIRECTORY

In the R code chunk below, please provide the code to: 1) clear your R environment, 2) print your current working directory, and 3) set your working directory to your '3.RStudio' folder.

```
#to clear the R environment, you would type rm(list = ls())
#to print your current working directory, you would type "pwd" in the terminal
#to set the current working directory to the 3.RStudio folder, I would type the following code --> setw
```

3) USING R AS A CALCULATOR

To follow up on the pre-class exercises, please calculate the following in the R code chunk below. Feel free to reference the **1. Introduction to version control and computing tools** handout.

1) the volume of a cube with length, l, = 5 (volume = l^3)
2) the area of a circle with radius, r, = 2 (area = pi * r^2).
3) the length of the opposite side of a right-triangle given that the angle, theta, = pi/4. (radians, a.k.a. 45°) and with hypotenuse length sqrt(2) (remember: sin(theta) = opposite/hypotenuse).
4) the log (base e) of your favorite number.

```
#5^3 = 125
#pi * 2^2 = 12.56637
#sin(pi/4) = x/sqrt(2); 0.7071068 = x/1.414214; x=1
#log10(8) = 0.90309
```

4) WORKING WITH VECTORS

To follow up on the pre-class exercises, please perform the requested operations in the R-code chunks below.

Basic Features Of Vectors

In the R-code chunk below, do the following: 1) Create a vector `x` consisting of any five numbers. 2) Create a new vector `w` by multiplying `x` by 14 (i.e., "scalar"). 3) Add `x` and `w` and divide by 15.

```
#x <- c(5,10,15,20,25)
#w <- 14*x = c(70,140,210,280,350)
#x + w = c(75,150,225,300,375); x+w/15 = c(9.67,19.33,29,38.67,48.33)
```

Now, do the following: 1) Create another vector (`k`) that is the same length as `w`. 2) Multiply `k` by `x`. 3) Use the combine function to create one more vector, `d` that consists of any three elements from `w` and any four elements of `k`.

```
#k <- c(3,7,12,18,22)
#k*x = c(15,70,180,360,550)
#To tackle this question, I would first create new vectors with the specified number of variables. The
```

Summary Statistics of Vectors

In the R-code chunk below, calculate the **summary statistics** (i.e., maximum, minimum, sum, mean, median, variance, standard deviation, and standard error of the mean) for the vector (`v`) provided.

```
#The vector v originally had 15 variables, including an NA value. However, if you try to run the summary
vectorV <- c(16.4, 16.0, 10.1, 16.8, 20.5, 20.2, 13.1, 24.8, 20.2, 25.0, 20.5, 30.5, 31.4, 27.1)
#max(vectorV) = 31.4
#min(vectorV) = 10.1
#sum(vectorV) = 292.6
#mean(vectorV) = 20.9
#median(vectorV) = 20.35
#var(vectorV) = 39.44
#sd(vectorV) = 6.280127
#MeanSE(vectorV) = 1.678435
```

5) WORKING WITH MATRICES

In the R-code chunk below, do the following: Using a mixture of Approach 1 and 2 from the **3. RStudio** handout, create a matrix with two columns and five rows. Both columns should consist of random numbers. Make the mean of the first column equal to 8 with a standard deviation of 2 and the mean of the second column equal to 25 with a standard deviation of 10.

```
groupA <- c(rnorm(5, mean = 8, sd=2))
groupB <- c(rnorm(5, mean = 25, sd=10))
#groupA = c(8.84, 8.83, 10.14, 8.82, 10.12)
#groupB = c(16.6, 29.7, 18.4, 19.6, 19.4)
combinedmatrix <- matrix(c(8.84, 8.83, 10.14, 8.82, 10.12, 16.6, 29.7, 18.4, 19.6, 19.4), nrow = 5, nco
```

Question 1: What does the `rnorm` function do? What do the arguments in this function specify? Remember to use `help()` or type `?rnorm`.

> Answer 1: The rnorm function creates random variables with a normal distribution.The arguments in the function can specify the mean and standard deviation of the randomly generated variables.

In the R code chunk below, do the following: 1) Load `matrix.txt` from the **3.RStudio** data folder as matrix `m`. 2) Transpose this matrix. 3) Determine the dimensions of the transposed matrix.

```
#matrixM <- read.delim("~/Documents/Github/QB2023_Werkowski/2.Worksheets/3.RStudio/data/matrix.txt", he
#t(matrixM)
#The transposed matrix should have dimensions that are the flipped version of the original dimensions.
```

Question 2: What are the dimensions of the matrix you just transposed?

> Answer 2: The dimensions would be 5 rows by 10 columns.

###Indexing a Matrix

In the R code chunk below, do the following: 1) Index matrix `m` by selecting all but the third column. 2) Remove the last row of matrix `m`.

```
#indexedmatrix <- transposed[1:4, c(1,2,4:10)]
```

6) BASIC DATA VISUALIZATION AND STATISTICAL ANALYSIS

Load Zooplankton Data Set

In the R code chunk below, do the following: 1) Load the zooplankton data set from the **3.RStudio** data folder. 2) Display the structure of this data set.

```
#mesodata <- read.table("data/zoop_nuts.txt", sep = "\t", header = TRUE)
#str(mesodata)
```

Correlation

In the R-code chunk below, do the following: 1) Create a matrix with the numerical data in the `meso` dataframe. 2) Visualize the pairwise **bi-plots** of the six numerical variables. 3) Conduct a simple **Pearson's correlation** analysis.

```
#mesonumericals <- mesodata[ ,3:8]
#pairs(mesonumericals)
#analysis <- cor(mesonumericals)


#Question 3***: Describe some of the general features based on the visualization and correlation analys

#Answer 3: If the values of the correlation analysis were negative, then that was reflected in the bi-p


#In the R code chunk below, do the following:
#1)Redo the correlation analysis using the `corr.test()` function in the `psych` package with the follo
#2) Now, redo this correlation analysis using a non-parametric method.
#3)Use the print command from the handout to see the results of each correlation analysis.

#analysis2 <- corr.test(mesonumericals, method = "pearson", adjust = "BH")
#print(analysis2, digits = 3)
```

```
#analysis3 <- corr.test(mesonumericals, method = "kendall", adjust = "BH")
#print(analysis3, digits = 3)
```

Question 4: Describe what you learned from `corr.test`. Specifically, are the results sensitive to whether you use parametric (i.e., Pearson's) or non-parametric methods? When should one use non-parametric methods instead of parametric methods? With the Pearson's method, is there evidence for false discovery rate due to multiple comparisons? Why is false discovery rate important?

> Answer 4: The correlation analysis test provided different results based upon whether or not the method being used was parametric vs nonparametric. The probability values were higher when nonparametric statistical methods were used. You should use nonparametric methods when there are outliers in the data, as the nonparametric methods are less likely to be affected by them. Parametric methods use continuous data and work better with data that is normally distributed. There is evidence for the false discovery rate with the Pearson's method. False discovery rate is important because it helps prevent you from creating Type-1 errors.

Linear Regression

In the R code chunk below, do the following: 1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zooplankton biomass (ZP). 2) Examine the output of the regression analysis. 3) Produce a plot of this regression analysis including the following: categorically labeled points, the predicted regression line with 95% confidence intervals, and the appropriate axis labels.

```
#reganalysis <-lm(ZP ~ TN, data= meso)
#summary(reganalysis)
```

```
#plot(meso$TN, meso$ZP, ylim = c(0,10), xlim = c(500, 5000), xlab = expression(paste("Total Nitrogen ("
#text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)
#newTN <- seq(min(meso$TN), max(meso$TN), 10)
#regline <- predict(reganalysis, newdata = data.frame(TN = newTN))
#lines(newTN, regline)
#conf95 <- predict(reganalysis, newdata = data.frame(TN = newTN), interval = c("confidence"), level = 0
#matlines(newTN, conf95[, c("lwr", "upr")], type = "l", lty = 2, lwd = 1, col = "black")
```

Question 5: Interpret the results from the regression model

> Answer 5: The regression model shows that the amount of nitrogen and the amount of zooplankton biomass are linked. As the total nitrogen increased, so did the biomass of zooplankton. If the nitrogen total was low, the amount of zooplankton was as well.

Analysis of Variance (ANOVA)

Using the R code chunk below, do the following: 1) Order the nutrient treatments from low to high (see handout). 2) Produce a barplot to visualize zooplankton biomass in each nutrient treatment. 3) Include error bars (+/- 1 sem) on your plot and label the axes appropriately. 4) Use a one-way analysis of variance (ANOVA) to test the null hypothesis that zooplankton biomass is affected by the nutrient treatment.

```
#NUTS <- factor(meso$NUTS, levels = c('L', 'M', 'H'))
#zp.means <- tapply(meso$ZP, NUTS, mean)
#sem <- function(x){sd(na.omit(x))/sqrt(length(na.omit(x)))}
#zp.sem <- tapply(meso$ZP, NUTS, sem)
#bp <- barplot(zp.means, ylim = c(0, round(max(meso$ZP), digits = 0)), pch = 15, cex = 1.25, las = 1, c
#arrows(x0 = bp, y0 = zp.means, y1 = zp.means - zp.sem, angle = 90, length = 0.1, lwd = 1)
#arrows(x0 = bp, y0 = zp.means, y1 = zp.means + zp.sem, angle = 90, length = 0.1, lwd = 1)
```

```
#fitanova <- aov(ZP ~ NUTS, data = meso)
#summary(fitanova)
```

```
#TukeyHSD(fitanova)
```

```
#par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
#plot(fitanova)
```

SYNTHESIS: SITE-BY-SPECIES MATRIX

In the R code chunk below, load the zoops.txt data set in your **3.RStudio** data folder. Create a site-by-species matrix (or dataframe) that does *not* include TANK or NUTS. The remaining columns of data refer to the biomass (µg/L) of different zooplankton taxa:

- CAL = calanoid copepods
- DIAP = *Diaphanasoma* sp.
- CYL = cyclopoid copepods
- BOSM = *Bosmina* sp.
- SIMO = *Simocephallus* sp.
- CERI = *Ceriodaphnia* sp.
- NAUP = naupuli (immature copepod)
- DLUM = *Daphnia lumholtzi*
- CHYD = *Chydorus* sp.

```
#sitebyspecies <- zoops[1:24, c(3:11)]
```

Question 6: With the visualization and statistical tools that we learned about in the **3. RStudio** handout, use the site-by-species matrix to assess whether and how different zooplankton taxa were responsible for the total biomass (ZP) response to nutrient enrichment. Describe what you learned below in the "Answer" section and include appropriate code in the R chunk.

```
#by looking at the Pearson's correlation, we can see that CYCL seems to be the only non-zooplankton tha
#Pcorrelation <- corr.test(sitebyspecies, method = "pearson", adjust = "BH")
#print(Pcorrelation, digits = 3)
```

```
#PcorrelationNP <- corr.test(sitebyspecies, method = "kendall", adjust = "BH")
#print(PcorrelationNP, digits = 3)
```

```
#zooreg <- lm(BOSM ~ CYCL, data = sitebyspecies)
#summary(zooreg)
```

```
#ANSWER: Bosmina showed a significant p-value in the summary of the linear regression analysis. In the
#plot(sitebyspecies$CYCL, sitebyspecies$BOSM, ylim = c(0,11), xlim = c(5, 400), xlab = expression(paste
```

SUBMITTING YOUR WORKSHEET

Use Knitr to create a PDF of your completed **3.RStudio_Worksheet.Rmd** document, push the repo to GitHub, and create a pull request. Please make sure your updated repo include both the PDF and RMarkdown files.

This assignment is due on **Wednesday, January 18th, 2021 at 12:00 PM (noon)**.