3. Worksheet: Basic R

Danny Peltier; Z620: Quantitative Biodiversity, Indiana University

26 March, 2021

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, March 24th, 2021 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.

```
rm(list = ls()) #clears environment
getwd() #current wd
```

```
## [1] "C:/Users/Danny/Desktop/GitHub/QB2021_Peltier-Thompson/2.Worksheets/3.RStudio"
```

```
setwd("C:/Users/Danny/Desktop/GitHub/QB2021_Peltier-Thompson/2.Worksheets/3.RStudio") #set new wd
getwd() #double check it worked
```

```
## [1] "C:/Users/Danny/Desktop/GitHub/QB2021_Peltier-Thompson/2.Worksheets/3.RStudio"
```

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.

```
l <- 5
l^3 #1
```

```
## [1] 125
```

```
r <- 2
pi * r^2 #2
```

```
## [1] 12.56637
```

```
theta <- pi / 4
hypot <- sqrt(2)
sintheta <- sin(theta)
sintheta / hypot #3
```

```
## [1] 0.5
```

```
favlog <- log(23) #4
```

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(3, 8, 11, 23, 27) #1
w <- x * 14 #2
(x + w) / 15 #3
```

```
## [1]  3  8 11 23 27
```

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(9, 8, 2, 1, 2) #1
k * x #2
```

```
## [1] 27 64 22 23 54
```

```
d <- c(w[2:4], k[2:5]) #3
```

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.

```
v <- c(16.4, 16.0, 10.1, 16.8, 20.5, NA, 20.2, 13.1, 24.8, 20.2, 25.0, 20.5, 30.5, 31.4, 27.1)
maximum <- max(na.omit(v))
minimum <- min(na.omit(v))
totalsum <- sum(na.omit(v))
average <- mean(na.omit(v))
middle <- median(na.omit(v))
variance <- var(na.omit(v))
standdev <- sd(na.omit(v))
sem <- function(v) {standdev / sqrt(length(na.omit(v)))}
stander <- sem(v)
```

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.

```
col1 <- rnorm(5, mean = 8, sd = 2)
col2 <- rnorm(5, mean = 25, sd = 10)
rm <- cbind(col1,col2)
```

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: for the rnorm function you enter a value of numbers you want, a mean, and a standard deviation. The function returns a vector of random numbers that fit a normal distribution of the variables you entered. So if i put in x <- rnorm(10, mean = 20, sd = 3) I will get a vector of 10 random numbers that have a mean of 20 and a standard deviation of 3

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.

```
m <- read.delim("data/matrix.txt",header = FALSE) #loaded the matrix
tm <- t(m) #tm is the transposed matrix
dim(tm) #dimensions of tm = 5x10
```

```
## [1]  5 10
```

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

Answer 2: 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`.

```
im <- m[-3] #all but the third column
m <- read.delim("data/matrix.txt",header = FALSE) #reload original m
lm <- m[-5]
```

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.

```
zoop <- read.delim("data/zoops.txt") #load zooplapnkton dataset as zoopa
str(zoop) #structure of zoop
```

```
## 'data.frame':    24 obs. of  11 variables:
##  $ TANK: int  5 14 16 21 23 25 27 34 12 15 ...
##  $ NUTS: chr  "L" "L" "L" "L" ...
##  $ CAL : num  70.5 27.1 5.3 79.2 31.4 22.7 0 35.7 74.8 5.3 ...
##  $ DIAP: num  0 19.2 8.8 17.9 0 ...
##  $ CYCL: num  66.1 129.6 12.7 141.3 11 ...
##  $ BOSM: num  2.2 0 0 3.4 0 0 0 0 0 0 ...
##  $ SIMO: num  417.8 0 73.1 0 482 ...
##  $ CERI: num  159.8 79.4 107.5 199 101.9 ...
##  $ NAUP: num  0 0 1.2 0 0 1.2 1.6 3.1 0 1.4 ...
##  $ DLUM: num  0 0 0 0 0 6.6 0 0 0 0 ...
##  $ CHYD: num  267 159 3158 298 580 ...
```

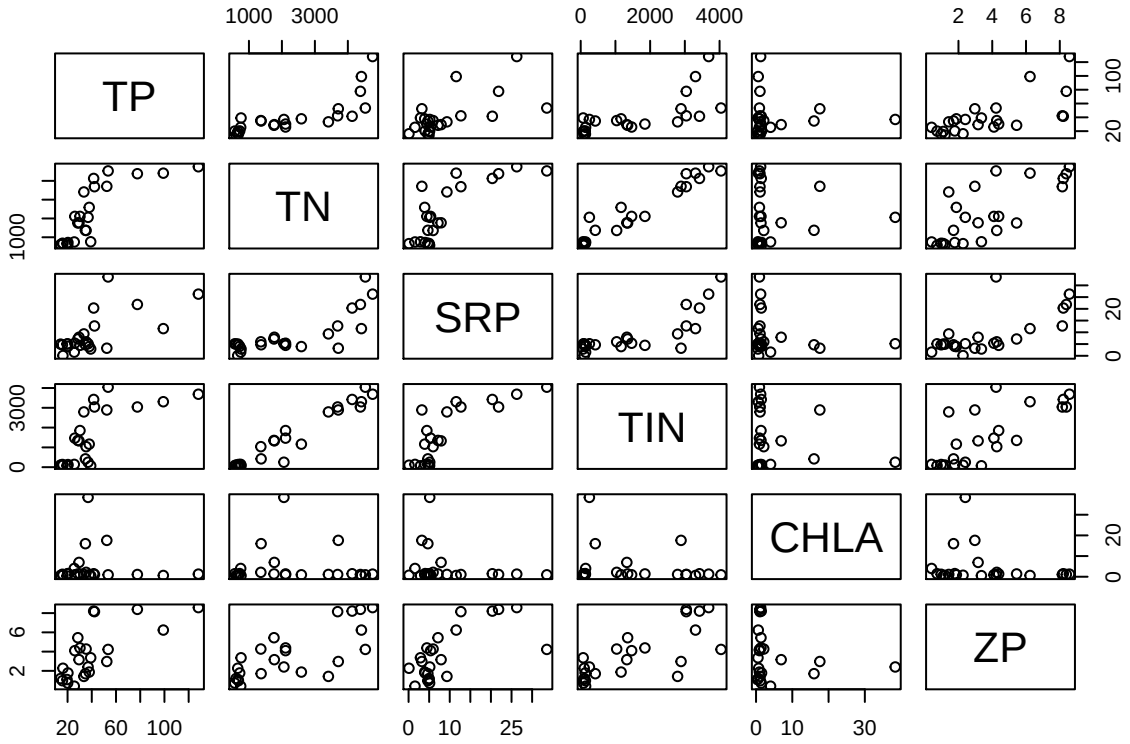
Correlation

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.

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

```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: chr  "L" "L" "L" "L" ...
##  $ TP  : num  20.3 25.6 14.2 39.1 20.1 ...
##  $ TN  : num  720 750 610 761 570 ...
##  $ SRP : num  4.02 1.56 4.97 2.89 5.11 4.68 5 0.1 7.9 3.92 ...
##  $ TIN : num  131.6 141.1 107.7 71.3 80.4 ...
##  $ CHLA: num  1.52 4 0.61 0.53 1.44 1.19 0.37 0.72 6.93 0.94 ...
##  $ ZP  : num  1.781 0.409 1.201 3.36 0.733 ...
```

```
meso.num <- meso[,3:8]
pairs(meso.num) #bi-plots
```




```
cor1 <- cor(meso.num) #pearsons correlation analysis
```

Question 3: Describe some of the general features based on the visualization and correlation analysis above?

Answer 3: zooplankton biomass is most correlated to the total inorganic nutrient concentration, followed in descending order by total nitrogen concentration, total phosphorus concentration, and soluble reactive phosphorus concentration. Zooplankton biomass is not correlated with chlorophyll a concentration. Total inorganic nutrient concentration, total nitrogen concentration, total phosphorus concentration, and soluble reactive phosphorus concentration are all coorelated with eachother which could explain why they are all fairly correlated with plankton biomass. Nothing is correlated to the chlorophyll a concentration suggesting that it doesn't have a major impact on the zooplankton biomass or the factors actually impact the zooplankton biomass. The reason total inorganic nutrient concentration is the most correlated to zooplankton biomass is because it encompasses the other non-zooplankton factors, aside from that, total Nitrogen concentration would be the nutrient zooplankton biomass is most related to.

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 following options: method = "pearson", adjust = "BH". 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.

```
install.packages("psych", repos="https://cran.rstudio.com")
```

```
## Installing package into 'C:/Users/Danny/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)
```

```
## package 'psych' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Danny\AppData\Local\Temp\RtmpuGlNa2\downloaded_packages
```

```
require("psych")
```

```
## Loading required package: psych
```

```
cor2 <- corr.test(meso.num, method = "pearson", adjust = "BH")
print(cor2, digits = 3, short = FALSE)
```

```
## Call:corr.test(x = meso.num, method = "pearson", adjust = "BH")
## Correlation matrix
##          TP     TN    SRP    TIN   CHLA     ZP
## TP    1.000  0.787  0.654  0.717 -0.017  0.697
## TN    0.787  1.000  0.784  0.969 -0.004  0.756
## SRP   0.654  0.784  1.000  0.801 -0.189  0.676
## TIN   0.717  0.969  0.801  1.000 -0.157  0.761
## CHLA -0.017 -0.004 -0.189 -0.157  1.000 -0.183
## ZP    0.697  0.756  0.676  0.761 -0.183  1.000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         TP    TN   SRP   TIN  CHLA    ZP
## TP   0.000 0.000 0.001 0.000 0.983 0.000
## TN   0.000 0.000 0.000 0.000 0.983 0.000
## SRP  0.001 0.000 0.000 0.000 0.491 0.000
## TIN  0.000 0.000 0.000 0.000 0.536 0.000
```

```
## CHLA 0.938 0.983 0.376 0.464 0.000 0.491
## ZP   0.000 0.000 0.000 0.000 0.393 0.000
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##          raw.lower  raw.r raw.upper raw.p lower.adj upper.adj
## TP-TN        0.561  0.787     0.903 0.000     0.398     0.936
## TP-SRP       0.341  0.654     0.837 0.001     0.141     0.890
## TP-TIN       0.441  0.717     0.869 0.000     0.255     0.912
## TP-CHLA     -0.417 -0.017     0.389 0.938    -0.576     0.554
## TP-ZP        0.409  0.697     0.859 0.000     0.218     0.906
## TN-SRP       0.557  0.784     0.902 0.000     0.393     0.935
## TN-TIN       0.929  0.969     0.987 0.000     0.893     0.991
## TN-CHLA     -0.407 -0.004     0.400 0.983    -0.568     0.562
## TN-ZP        0.508  0.756     0.889 0.000     0.334     0.926
## SRP-TIN      0.587  0.801     0.910 0.000     0.431     0.940
## SRP-CHLA    -0.551 -0.189     0.232 0.376    -0.682     0.421
## SRP-ZP       0.375  0.676     0.848 0.000     0.180     0.898
## TIN-CHLA    -0.527 -0.157     0.263 0.464    -0.663     0.448
## TIN-ZP       0.515  0.761     0.891 0.000     0.343     0.927
## CHLA-ZP     -0.546 -0.183     0.238 0.393    -0.678     0.427
```

```
cor3 <- corr.test(meso.num, method = "spearman", adjust = "BH")
print(cor3, digits = 3, short = FALSE)
```

```
## Call:corr.test(x = meso.num, method = "spearman", adjust = "BH")
## Correlation matrix
##         TP    TN    SRP   TIN   CHLA     ZP
## TP   1.000 0.895  0.539 0.761  0.040  0.741
## TN   0.895 1.000  0.647 0.942  0.021  0.748
## SRP  0.539 0.647  1.000 0.726 -0.064  0.627
## TIN  0.761 0.942  0.726 1.000  0.088  0.738
## CHLA 0.040 0.021 -0.064 0.088  1.000 -0.072
## ZP   0.741 0.748  0.627 0.738 -0.072  1.000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         TP    TN   SRP   TIN  CHLA    ZP
## TP   0.000 0.000 0.010 0.000 0.914 0.000
## TN   0.000 0.000 0.001 0.000 0.923 0.000
## SRP  0.007 0.001 0.000 0.000 0.884 0.002
## TIN  0.000 0.000 0.000 0.000 0.884 0.000
## CHLA 0.853 0.923 0.767 0.683 0.000 0.884
## ZP   0.000 0.000 0.001 0.000 0.737 0.000
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##          raw.lower  raw.r raw.upper raw.p lower.adj upper.adj
## TP-TN        0.769  0.895     0.954 0.000     0.667     0.970
## TP-SRP       0.173  0.539     0.774 0.007    -0.038     0.846
## TP-TIN       0.515  0.761     0.891 0.000     0.343     0.927
## TP-CHLA     -0.369  0.040     0.436 0.853    -0.537     0.592
## TP-ZP        0.481  0.741     0.881 0.000     0.302     0.921
## TN-SRP       0.330  0.647     0.833 0.001     0.129     0.888
## TN-TIN       0.870  0.942     0.975 0.000     0.807     0.984
## TN-CHLA     -0.386  0.021     0.421 0.923    -0.551     0.579
```

```
## TN-ZP        0.493  0.748     0.884 0.000     0.316     0.923
## SRP-TIN      0.457  0.726     0.874 0.000     0.273     0.916
## SRP-CHLA    -0.456 -0.064     0.348 0.767    -0.607     0.520
## SRP-ZP       0.299  0.627     0.822 0.001     0.096     0.880
## TIN-CHLA    -0.327  0.088     0.474 0.683    -0.502     0.622
## TIN-ZP       0.476  0.738     0.879 0.000     0.295     0.919
## CHLA-ZP     -0.462 -0.072     0.341 0.737    -0.612     0.514
```

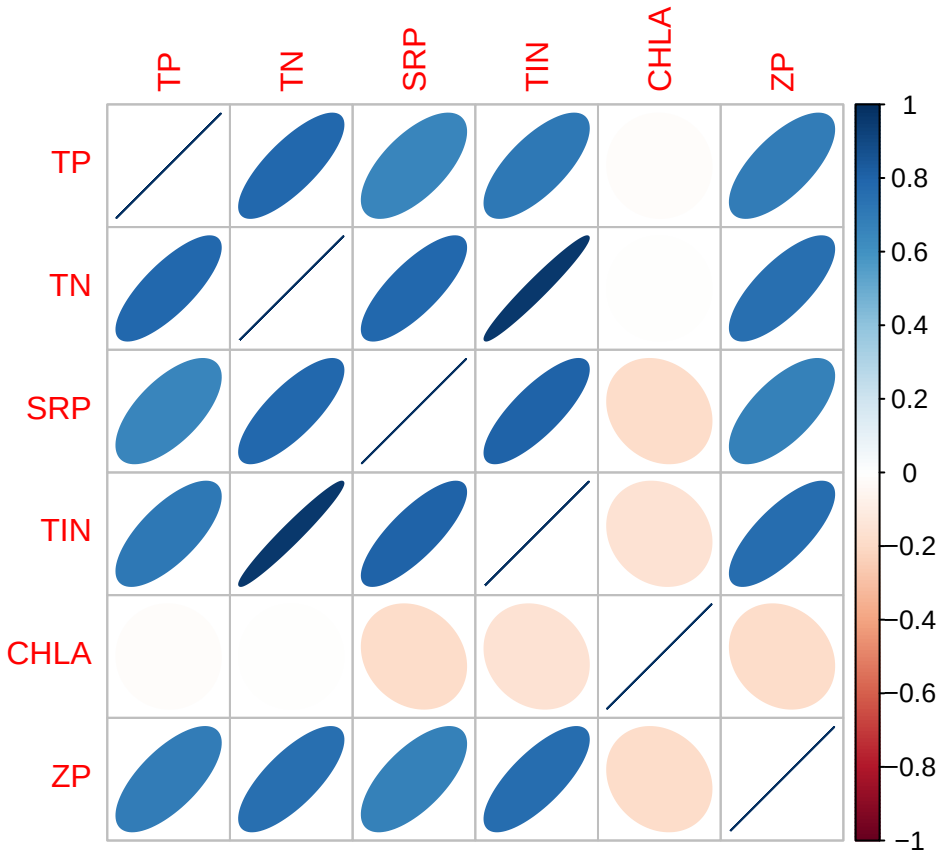
```
install.packages("corrplot", repos="http://cran.rstudio.com")
```

```
## Installing package into 'C:/Users/Danny/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)

## package 'corrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Danny\AppData\Local\Temp\RtmpuGlNa2\downloaded_packages
```

```
require(corrplot)
```

```
## Loading required package: corrplot

## corrplot 0.84 loaded
```

```
corrplot(cor1, method = "ellipse")
```



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: corr.test does the correlation analysis you tell it to do but it also shows the p-values to each correlation. I've never taken a statistics class so I'm not 100% sure about my answers to this and would definitly like feedback. Parametric methods analyze the means and assumes that the data follows a normal distribution; however, the data doesnt have to have a normal distribution if its a large data set.Parametric methods can be heavily affected if your data has outliers because it analyzes means. Non-parametric methods analyze the medians and theres no assumption that the data have normal distributions. Non-parametric methods are good to use if you have a small sample size, the data is better represented by the median, or if the data has major outliers. The results are sensitive to the different methods. Both analyses had low correlation values and high P-values for CHLA comparisons, which were drastically different than other variables, but parametric methods had larger ranges and more extreme values (correlation range: P = -0.183 - -0.004, NP = -0.072 - 0.088; P-value range: P = 0.376 - 0.983 NP = 0.683 - 0.923). The other variables were similar to each other with high correlation values and low P-values (correlation range: P = 0.654 - 0.969, NP = 0.539 - 0.895; P-value range: P = 0.000 - 0.001 NP = 0.001 - 0.010). Parametric methods returned really high correlation values, lowest (most significant) P-values, and the correlation with zooplankton body mass ranked from highest to lowest is TIN, TN, TP, SRP, CHLA. Non-parametric methods had more moderate values but the p-values were still significant and the correlation with zooplankton body mass ranked from highest to lowest TN, TP, TIN, SRP, CHLA. The Pearson's method has evidence for false discovery rate but only in the CHLA P-Values (I actually see more evidence for false discovery rate in the Spearman method and I'm not sure if that was supposed to happen). False discovery rate is important because they will skew the significance because it increases the chance to reject the null-hypothesis with an increase in tests done.

Linear Regression

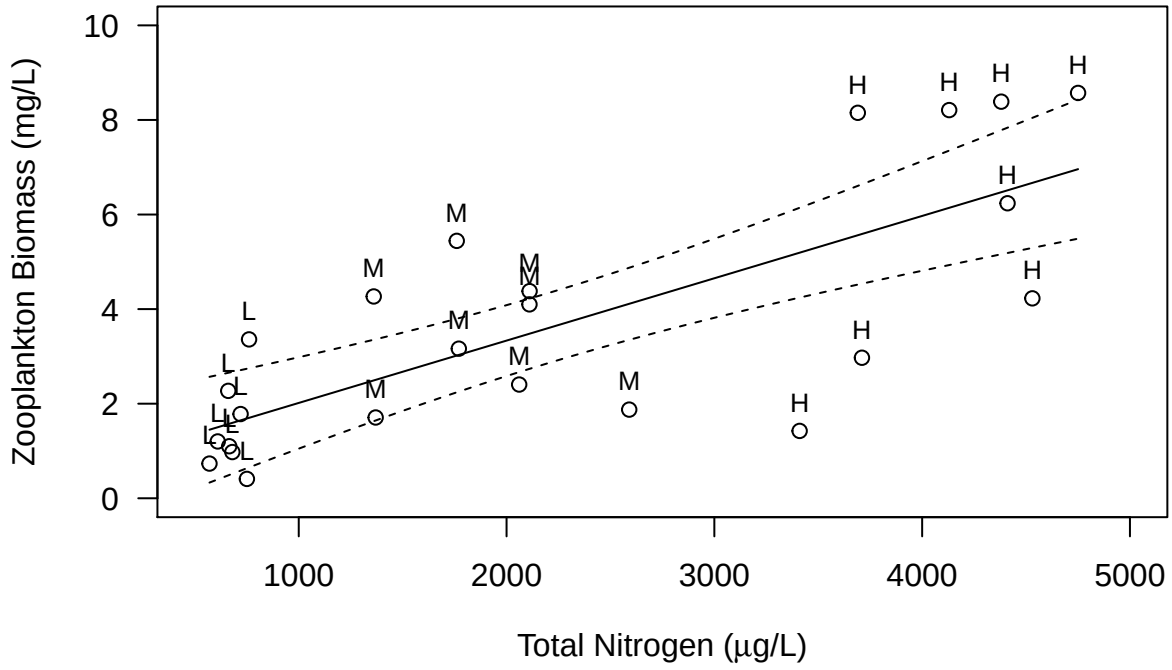
In the R code chunk below, do the following: 1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zooplankton biomass (ZP). 2) Examine the output of the regression analysis. 3) Produce a plot of this regression analysis including the following: categorically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

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

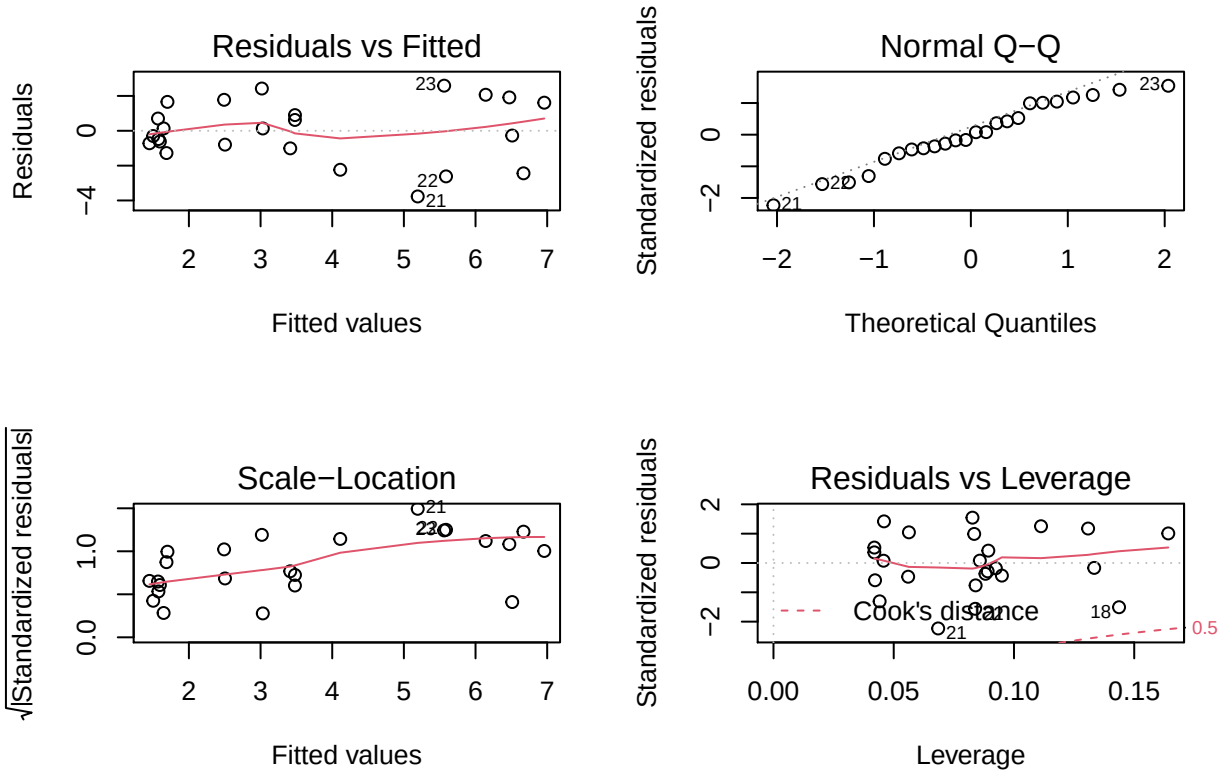
```
##
## Call:
## lm(formula = ZP ~ TN, data = meso)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7690 -0.8491 -0.0709  1.6238  2.5888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6977712  0.6496312   1.074    0.294
## TN          0.0013181  0.0002431   5.421 1.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.75 on 22 degrees of freedom
## Multiple R-squared:  0.5719, Adjusted R-squared:  0.5525
## F-statistic: 29.39 on 1 and 22 DF,  p-value: 1.911e-05
```

```
plot.new()
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(fitreg, newdata = data.frame(TN = newTN))
lines(newTN, regline)
conf95 <- predict(fitreg, newdata = data.frame(TN = newTN), interval = c("confidence"), level = 0.95, ty
matlines(newTN, conf95[, c("lwr", "upr")], type ="l", lty = 2, lwd = 1, col = "black")
```




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




```
dev.off()
```

```
## null device
##           1
```

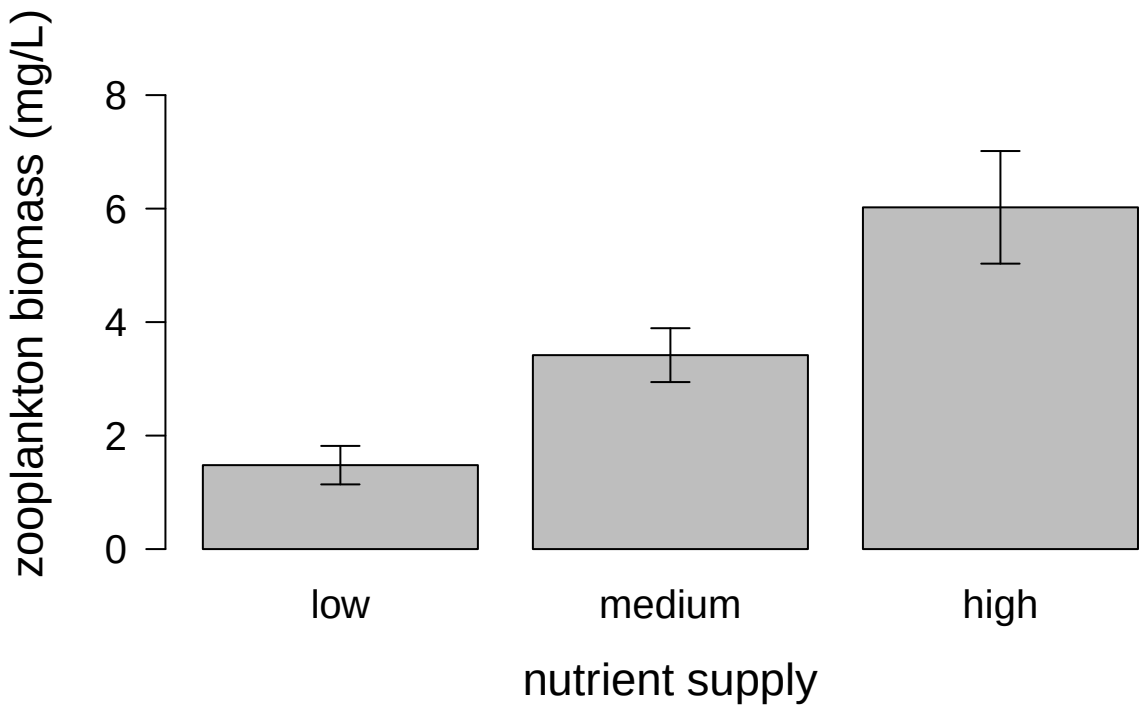
Question 5: Interpret the results from the regression model

> Answer 5: The P-value is very low so the results are statistically significant, but the adjusted R-squared value is ~0.55 so about half of the variation can be explained by the regression model so zooplankton body mass is related to total Nitrogen, but total Nitrogen wont be the most accurate in predicting zooplankton body mass. The residuals aren't randomly distributed around zero so TN doesn't fully explain ZP. In the QQ plot, the relationship is generally linear but there are outliers and most values fall below the line. The sqrt of the residuals are not linerally related to the fitted values. It doesn't look like values fall outside of the Cook's distance line but there are many values >|1|. In general, the values aren't normally distributed and aren't homoscedastic.

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.

```r
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$ZP, NUTS, sem)
plot.new()
bp <- barplot(zp.means, ylim = c(0, round(max(meso$ZP), digits = 0)), pch = 15, cex = 1.25, las = 1, ce
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)
```




```r
dev.off()
```

```
## null device
##           1
```

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

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## NUTS         2  83.15   41.58   11.77 0.000372 ***
## Residuals   21  74.16    3.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(fitanova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ZP ~ NUTS, data = meso)
##
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## M-L  1.938625 -0.4297094  4.3069594 0.1220246
```

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

```
## hat values (leverages) are all = 0.125
##  and there are no factor predictors; no plot no. 5
```

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.

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.

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

```
## 'data.frame':    24 obs. of  11 variables:
##  $ TANK: int  5 14 16 21 23 25 27 34 12 15 ...
##  $ NUTS: chr  "L" "L" "L" "L" ...
##  $ CAL : num  70.5 27.1 5.3 79.2 31.4 22.7 0 35.7 74.8 5.3 ...
##  $ DIAP: num  0 19.2 8.8 17.9 0 ...
##  $ CYCL: num  66.1 129.6 12.7 141.3 11 ...
##  $ BOSM: num  2.2 0 0 3.4 0 0 0 0 0 0 ...
##  $ SIMO: num  417.8 0 73.1 0 482 ...
##  $ CERI: num  159.8 79.4 107.5 199 101.9 ...
##  $ NAUP: num  0 0 1.2 0 0 1.2 1.6 3.1 0 1.4 ...
##  $ DLUM: num  0 0 0 0 0 6.6 0 0 0 0 ...
##  $ CHYD: num  267 159 3158 298 580 ...
```

```
nonut <- zoops[,3:11]
str(nonut)
```

```
## 'data.frame':    24 obs. of  9 variables:
##  $ CAL : num  70.5 27.1 5.3 79.2 31.4 22.7 0 35.7 74.8 5.3 ...
##  $ DIAP: num  0 19.2 8.8 17.9 0 ...
##  $ CYCL: num  66.1 129.6 12.7 141.3 11 ...
##  $ BOSM: num  2.2 0 0 3.4 0 0 0 0 0 0 ...
##  $ SIMO: num  417.8 0 73.1 0 482 ...
##  $ CERI: num  159.8 79.4 107.5 199 101.9 ...
##  $ NAUP: num  0 0 1.2 0 0 1.2 1.6 3.1 0 1.4 ...
##  $ DLUM: num  0 0 0 0 0 6.6 0 0 0 0 ...
##  $ CHYD: num  267 159 3158 298 580 ...
```

```
bigZP <- c(meso.num[,6]*1000) #convert ZP into micrograms
nonutZP <- cbind(nonut,bigZP) #add converted ZP to zooplankton set
#I thought that the sum of the biomass for each species of zooplankton would sum up to ZP but it didn't
avgnonut <- apply(nonut, 2, mean)
sdnonut <- apply(nonut,2,sd)
rank(avgnonut) #this returned the order of each species average biomass, from largest to smallest: CHYD
```

```
##  CAL DIAP CYCL BOSM SIMO CERI NAUP DLUM CHYD
##    4    5    6    3    8    7    2    1    9
```

```
range(avgnonut) #the averages had a range of 0.275 micrograms to 2906.638 micrograms
```

```
## [1]    0.275 2906.638
```

```
#because the range is so drastic, the species with the larger biomass will have more impact on the tota
```

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 24th, 2021 at 12:00 PM (noon)**.