

3. Worksheet: Basic R

Jonathan Enriquez Madrid; Z620: Quantitative Biodiversity, Indiana University

18 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. This 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.

```
rm(list = ls())  
getwd()
```

```
## [1] "C:/Users/jonat/GitHub/QB2023_Enriquez_Madrid/2.Worksheets/3.RStudio"
```

```
setwd("C:/Users/jonat/GitHub/QB2023_Enriquez_Madrid/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: $\sin(\theta) = \text{opposite}/\text{hypotenuse}$).
- 4) the log (base e) of your favorite number.

```
5^3      #125
```

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

```
pi*2^2    #12.56637
```

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

```
sqrt(2)*sin(pi/4)    #1
```

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

```
log(7)     #1.94591
```

```
## [1] 1.94591
```

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(1, 2, 3, 4, 5) #Vector made
w<- x*14 #Vector made
y<- x+w #x and w added together
y/15 #1,2,3,4,5
```

```
## [1] 1 2 3 4 5
```

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(5, 10, 15, 20, 25) #vector made
k*x #5, 20, 45, 80, 125
```

```
## [1] 5 20 45 80 125
```

```
d<- c(14, 28, 42, 5, 10, 15, 20) #vector made
```

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)
max(na.omit(v)) #31.4
```

```
## [1] 31.4
```

```
min(na.omit(v)) #10.1
```

```
## [1] 10.1
```

```
sum(na.omit(v)) #292.6
```

```
## [1] 292.6
```

```
mean(na.omit(v)) #20.9
```

```
## [1] 20.9
```

```
median(na.omit(v)) #20.35
```

```
## [1] 20.35
```

```
var(na.omit(v)) #39.44
```

```
## [1] 39.44
```

```
sd(na.omit(v)) #6.280127
```

```
## [1] 6.280127
```

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.

```
j<- c(rnorm(5, mean = 8, sd = 2))
z<- c(rnorm(5, mean = 25, sd = 10))
k2<- cbind(z, j)
k2 #Gives a matrix with two columns and five rows with the required means and standard deviations
```

```
##           z           j
## [1,] 32.490725 8.483207
## [2,] 37.072233 8.358162
## [3,]  5.659473 8.772869
## [4,] 14.269374 4.327088
## [5,] 16.079983 8.737870
```

```
help(rnorm)
```

```
## starting httpd help server ... done
```

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: `rnorm` generates a random distribution of numbers, specifying the mean and standard deviation in the vector of numbers. The arguments in this function specify vector of quantiles, vector of probabilities, number of observations, vector of means, vector of standard deviations, log, and lower.tail ($P[X \leq x]$ as the default and $P[X > x]$ for the nondefault).

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 <- as.matrix(read.table("data/matrix.txt", sep = "\t", header = FALSE))
n <- t(m) #transposed m
dim(n) #Five rows and ten columns. m has ten rows and five columns.
```

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

```
dim(m)
```

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

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

Answer 2: The dimensions of the matrix I just transposed are 5 rows and 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`.

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
m2 <- m[, c(1,2,4,5)] # Indexed matrix to include all columns but the third.
```

```
m3 <- m[1:9,] #Indexed matrix to exclude the last row
```

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.

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

```
str(meso) #Shows structure of data frame, not the same as matrix or vector as matrix and vectors are on
```

```
## '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 ...
```

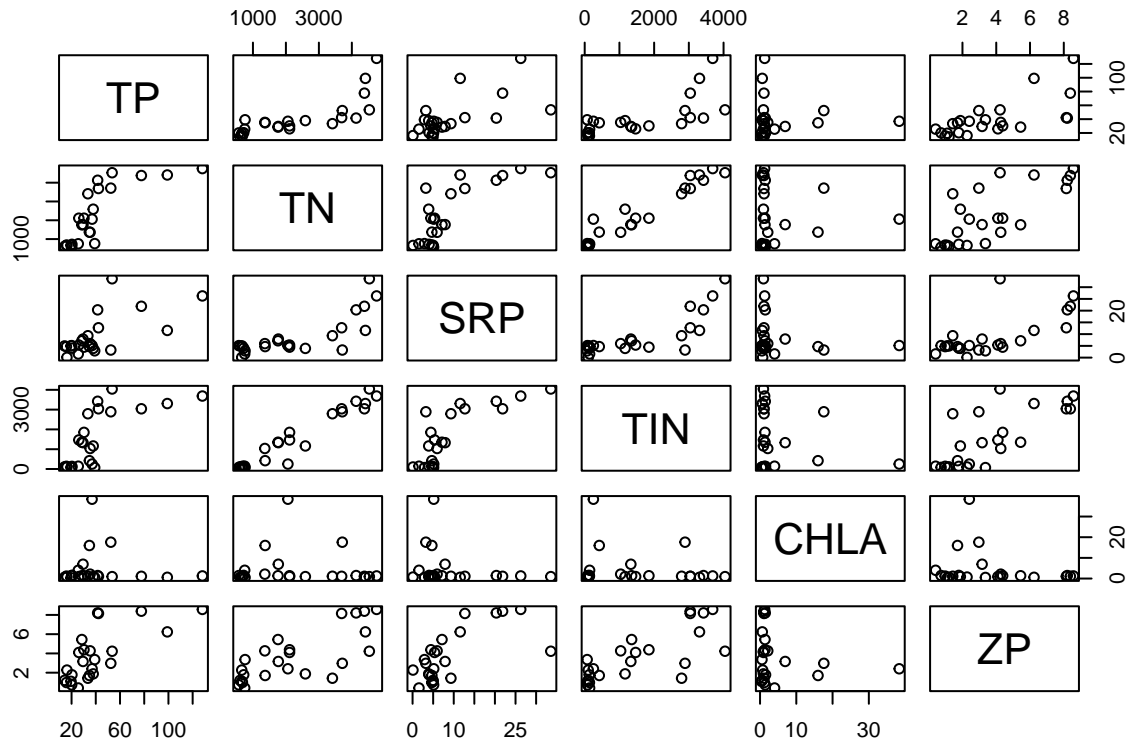
```
## $ CHLA: num 1.52 4 0.61 0.53 1.44 1.19 0.37 0.72 6.93 0.94 ...
```

```
## $ ZP : num 1.781 0.409 1.201 3.36 0.733 ...
```

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.num <- meso[,3:8] #Creates indexed matrix of dataframe. Excludes categorical data
pairs(meso.num) #Gives bi-plots of meso.num matrix
```



```
cor1 <- cor(meso.num)
cor1 #Conducts Pearson's correlation. 1-.5 (strong,positive), .5-.3 (moderate,positive), .3-0 (weak,pos
```

	TP	TN	SRP	TIN	CHLA	ZP
## TP	1.00000000	0.786510407	0.6540957	0.7171143	-0.016659593	0.6974765
## TN	0.78651041	1.00000000	0.7841904	0.9689999	-0.004470263	0.7562474
## SRP	0.65409569	0.784190400	1.0000000	0.8009033	-0.189148017	0.6762947
## TIN	0.71711434	0.968999866	0.8009033	1.0000000	-0.156881463	0.7605629
## CHLA	-0.01665959	-0.004470263	-0.1891480	-0.1568815	1.000000000	-0.1825999
## ZP	0.69747649	0.756247384	0.6762947	0.7605629	-0.182599904	1.0000000

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

Answer 3: General features would include strong and moderate correlations between the variables being studied, in addition to some weak, negative correlations. The data is presented in such a way that one can check a correlation between each of the two variables being looked at. No p-values are provided.

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="http://cran.rstudio.com/")
```

```
## Installing package into 'C:/Users/jonat/AppData/Local/R/win-library/4.2'  
## (as 'lib' is unspecified)
```

```
## package 'psych' successfully unpacked and MD5 sums checked  
##  
## The downloaded binary packages are in  
## C:\Users\jonat\AppData\Local\Temp\Rtmps7aNbl\downloaded_packages
```

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

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

```
cor2 <- corr.test(meso.num, method = "pearson", adjust = "BH")  
print(cor2, digits = 3)#provides P-values for Pearson correlation
```

```
## 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  
##  
## To see confidence intervals of the correlations, print with the short=FALSE option
```

```
cor3 <- corr.test(meso.num, method = "kendal", adjust = "BH")  
cor3
```

```
## Call:corr.test(x = meso.num, method = "kendal", adjust = "BH")  
## Correlation matrix  
##      TP      TN      SRP      TIN      CHLA      ZP  
## TP    1.00  0.74  0.39  0.58  0.04  0.54  
## TN    0.74  1.00  0.48  0.81  0.01  0.55  
## SRP   0.39  0.48  1.00  0.56 -0.07  0.45
```

```
## TIN 0.58 0.81 0.56 1.00 0.04 0.55
## CHLA 0.04 0.01 -0.07 0.04 1.00 -0.05
## ZP 0.54 0.55 0.45 0.55 -0.05 1.00
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##      TP   TN  SRP  TIN  CHLA   ZP
## TP  0.00 0.00 0.09 0.01 0.90 0.01
## TN  0.00 0.00 0.03 0.00 0.95 0.01
## SRP 0.06 0.02 0.00 0.01 0.90 0.05
## TIN 0.00 0.00 0.00 0.00 0.90 0.01
## CHLA 0.84 0.95 0.76 0.84 0.00 0.90
## ZP 0.01 0.01 0.03 0.01 0.81 0.00
##
## To see confidence intervals of the correlations, print with the short=FALSE option

print(cor3, digits = 3)
```

```
## Call:corr.test(x = meso.num, method = "kendal", adjust = "BH")
## Correlation matrix
##      TP   TN  SRP  TIN  CHLA   ZP
## TP  1.000 0.739 0.391 0.577 0.044 0.536
## TN  0.739 1.000 0.478 0.809 0.015 0.551
## SRP 0.391 0.478 1.000 0.563 -0.066 0.449
## TIN 0.577 0.809 0.563 1.000 0.044 0.548
## CHLA 0.044 0.015 -0.066 0.044 1.000 -0.051
## ZP 0.536 0.551 0.449 0.548 -0.051 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 0.088 0.014 0.899 0.015
## TN  0.000 0.000 0.034 0.000 0.946 0.014
## SRP 0.059 0.018 0.000 0.014 0.899 0.046
## TIN 0.003 0.000 0.004 0.000 0.899 0.014
## CHLA 0.839 0.946 0.760 0.839 0.000 0.899
## ZP 0.007 0.005 0.028 0.006 0.813 0.000
##
## To see confidence intervals of the correlations, print with the short=FALSE option
```

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 `corr.test` allows us to see if there is a strong or weak, positive or negative correlation between two variables. The results are sensitive to whether one uses parametric or non-parametric methods. One should use non-parametric methods when the data is not normality distributed, if there is unequal variance, if the sample is too small, or if the data is ordinal or nominal. There is evidence for false discovery rate due to multiple comparisons with the Pearson's method, the adjustment is the Benjamin & Hochberg-corrected P-values. False discovery rate is important because as the number of tests performed increases, the probability of getting a significant P-value just by chance increases (Type-1 error).

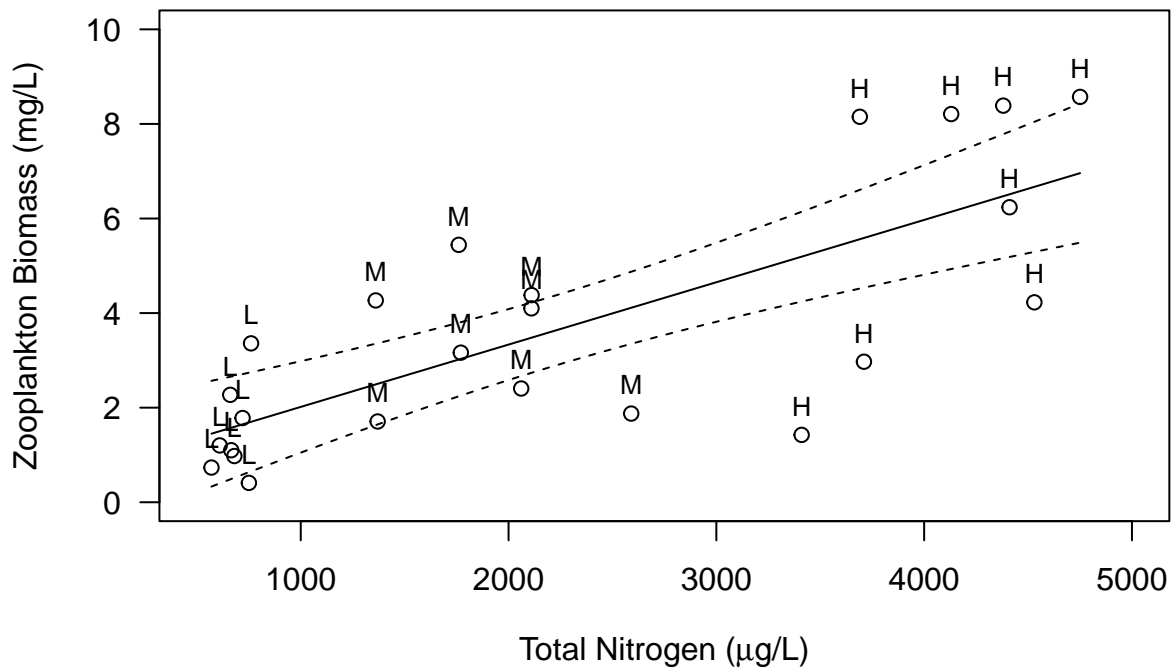
Linear Regression

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) #Gives a sign. P-value, indicating that TN is a good predictor of ZP.
```

```
##
## 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(meso$TN, meso$ZP, ylim = c(0,10), xlim = c(500, 5000), xlab = expression(paste("Total Nitrogen (",
text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)#adds letters to points on plot
newTN <- seq(min(meso$TN), max(meso$TN), 10)#adds regression line
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, t
matlines(newTN, conf95[, c("lwr", "upr")], type = "l", lty = 2, lwd = 1, col = "black")#adds 95% confid
```



Question 5: Interpret the results from the regression model

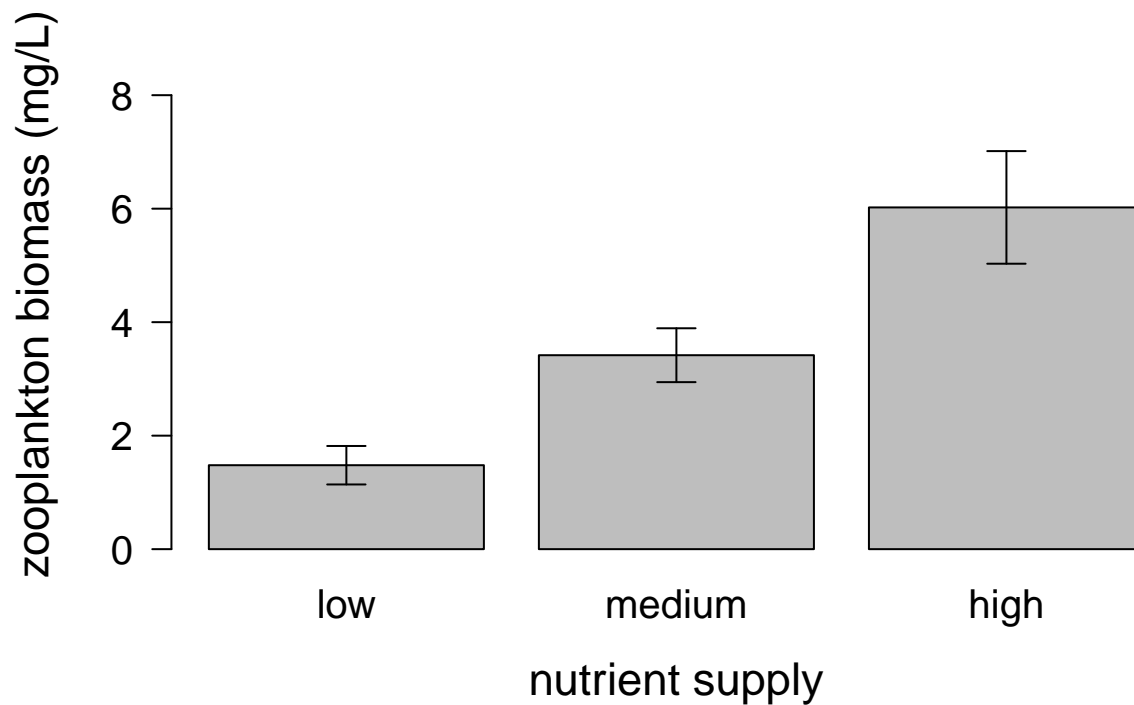
Answer 5: The regression model shows us that Total Nitrogen is indeed a good predictor of zooplankton biomass. We see that as Total Nitrogen increases, so does the zooplankton biomass. The categorically labeled points are especially helpful as we can visually distinguish how high Total Nitrogen concentrations differ from low Total Nitrogen concentrations in terms of how they affect zooplankton biomass.

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)#gives the mean of ZP based on info from other columns (nutrition t

bp <- barplot(zp.means, ylim=c(0, round(max(meso$ZP), digits = 0)), pch = 15, cex = 1.25, las = 1, cex
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)#makes error b
```



```
fitanova <- aov(ZP ~ NUTS, data = meso) #runs anova
summary(fitanova) #Significant p-value. There is a difference in zooplankton biomass between the three d
```

```
##           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) #The differences in zooplankton biomasses are significant between the low and high nu
```

```
##      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
```

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 ($\mu\text{g/L}$) 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.

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

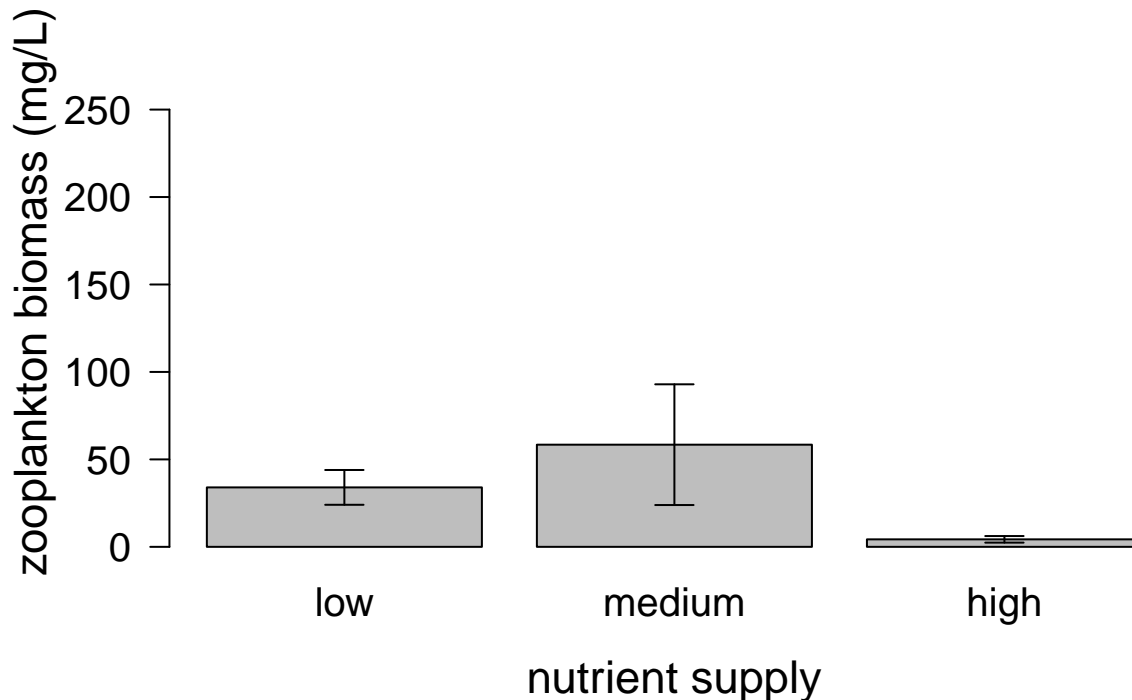
##	TANK	NUTS	CAL	DIAP	CYCL	BOSM	SIMO	CERI	NAUP	DLUM	CHYD
## 1	5	L	70.5	0.0	66.1	2.2	417.8	159.8	0.0	0.0	266.9
## 2	14	L	27.1	19.2	129.6	0.0	0.0	79.4	0.0	0.0	158.7
## 3	16	L	5.3	8.8	12.7	0.0	73.1	107.5	1.2	0.0	3158.2
## 4	21	L	79.2	17.9	141.3	3.4	0.0	199.0	0.0	0.0	298.5
## 5	23	L	31.4	0.0	11.0	0.0	482.0	101.9	0.0	0.0	580.2
## 6	25	L	22.7	285.1	153.0	0.0	241.5	135.5	1.2	6.6	262.4
## 7	27	L	0.0	2.3	11.0	0.0	73.1	185.0	1.6	0.0	2004.4
## 8	34	L	35.7	65.9	102.9	0.0	0.0	318.5	3.1	0.0	1260.7
## 9	12	M	74.8	178.7	266.5	0.0	0.0	1.9	0.0	0.0	1190.9
## 10	15	M	5.3	4.9	87.8	0.0	1099.2	136.4	1.4	0.0	2939.6
## 11	18	M	18.4	2.3	29.4	0.0	393.8	147.6	1.2	0.0	4857.3
## 12	22	M	14.0	2.3	37.7	0.0	1251.5	74.8	0.0	0.0	2725.5
## 13	28	M	14.0	2.3	132.9	0.0	818.6	98.1	1.2	0.0	814.5
## 14	30	M	48.8	2.3	107.9	2.2	9.0	132.7	0.0	0.0	2867.5
## 15	35	M	0.0	0.0	17.7	0.0	145.3	19.7	0.0	0.0	4201.6
## 16	36	M	292.0	269.5	373.4	10.7	0.0	8.5	1.2	0.0	1456.8
## 17	4	H	9.7	0.0	41.1	0.0	2397.8	9.4	0.0	0.0	5697.9
## 18	6	H	0.0	2.3	0.0	0.0	225.5	24.3	0.0	0.0	8323.2
## 19	10	H	5.3	0.0	86.2	0.0	465.9	527.7	1.2	0.0	3146.9
## 20	11	H	14.0	7.5	69.5	0.0	594.2	78.5	0.0	0.0	7629.2
## 21	17	H	0.0	24.4	101.2	0.0	313.6	176.6	0.0	0.0	7597.6
## 22	19	H	0.0	7.5	253.2	8.3	0.0	112.1	1.6	0.0	2594.8
## 23	24	H	5.3	2.3	96.2	0.0	786.6	76.6	0.0	0.0	463.0
## 24	29	H	0.0	2.3	66.1	0.0	826.7	85.1	0.0	0.0	5263.0

```
str(zooptaxa)#shows what is integer, character, and number in data
```

```
## '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 ...
## $ 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 ...
```

```
zooptaxa.num <- zooptaxa[,3:11]#Indexes dataframe so that only numerical data is present. Nutrition (wh
NUTS <- factor(zooptaxa$NUTS, levels = c('L', 'M', 'H'))
#CAL
CAL.means <- tapply(zooptaxa$CAL, NUTS, mean)
CAL.sem <- tapply(zooptaxa$CAL, NUTS, sem)

bpCAL <- barplot(CAL.means, ylim =c(0, round(max(zooptaxa$CAL), digits = 0)), pch = 15, cex = 1.25, las
arrows(x0 = bp, y0 = CAL.means, y1 = CAL.means - CAL.sem, angle = 90, length = 0.1, lwd = 1)
arrows(x0 = bp, y0 = CAL.means, y1 = CAL.means + CAL.sem, angle = 90, length = 0.1, lwd = 1)#More in Me
```

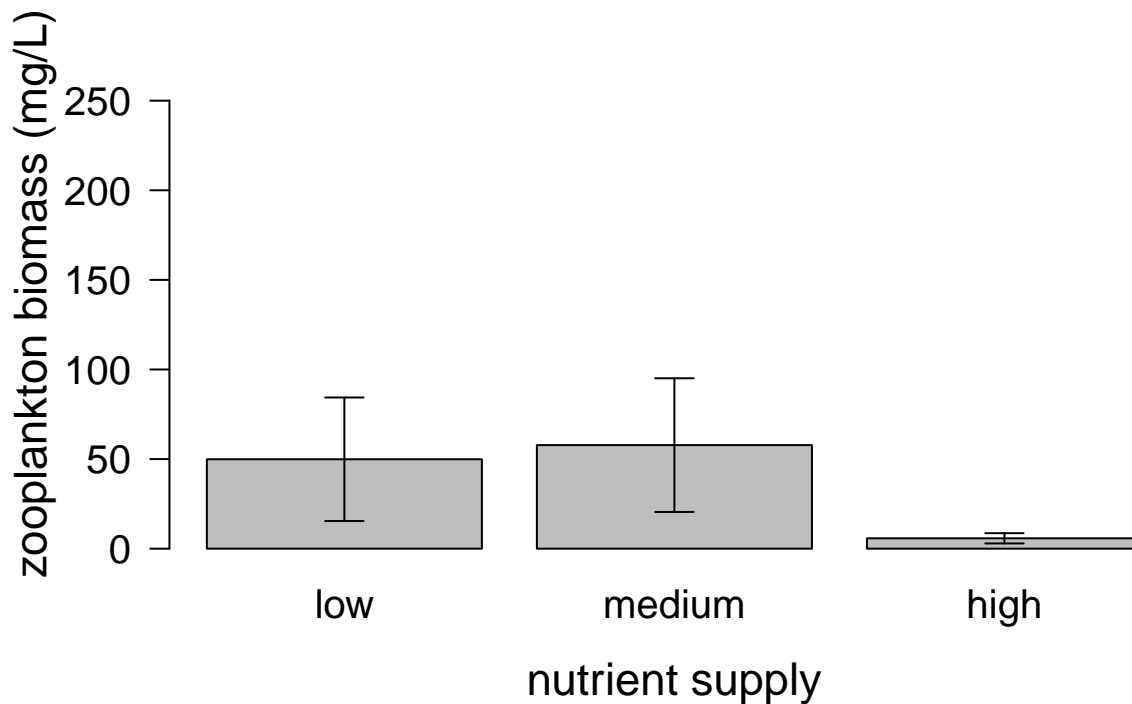


```

#DIAP
DIAP.means <- tapply(zooptaxa$DIAP, NUTS, mean)
DIAP.sem <- tapply(zooptaxa$DIAP, NUTS, sem)

bpDIAP <- barplot(DIAP.means, ylim=c(0, round(max(zooptaxa$DIAP), digits = 0)), pch = 15, cex = 1.25,
arrows(x0 = bp, y0 = DIAP.means, y1 = DIAP.means - DIAP.sem, angle = 90, length = 0.1, lwd = 1)
arrows(x0 = bp, y0 = DIAP.means, y1 = DIAP.means + DIAP.sem, angle = 90, length = 0.1, lwd = 1)#More in

```

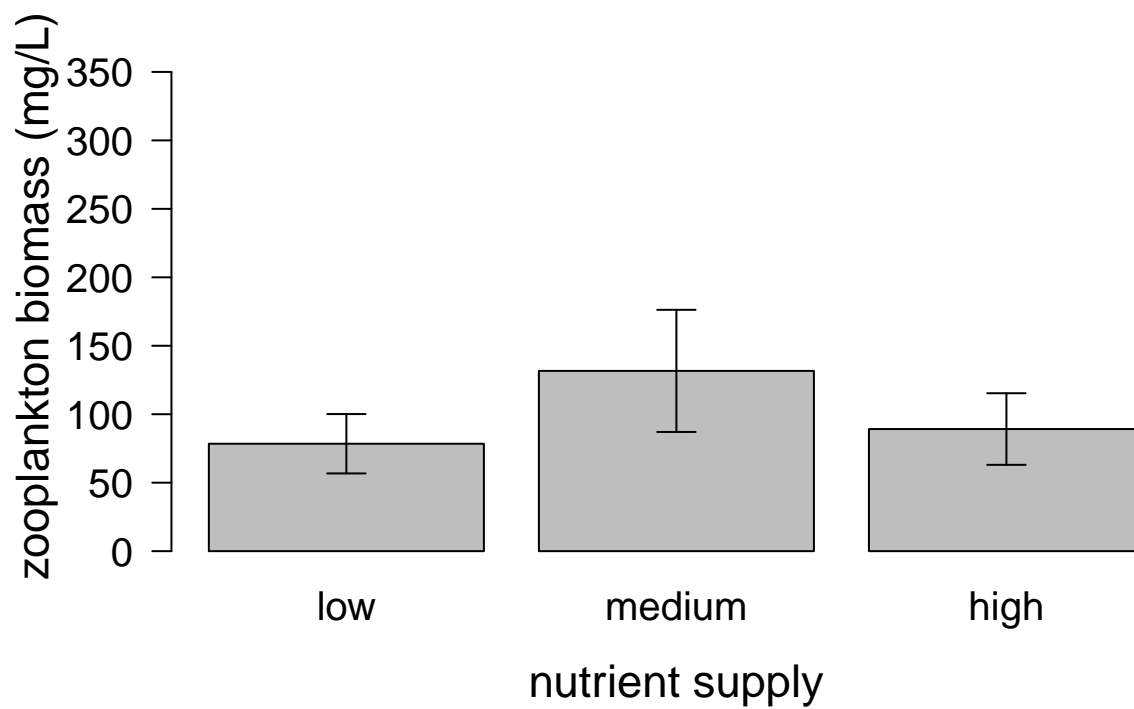


```

#CYCL
CYCL.means <- tapply(zooptaxa$CYCL, NUTS, mean)
CYCL.sem <- tapply(zooptaxa$CYCL, NUTS, sem)

bpCYCL <- barplot(CYCL.means, ylim=c(0, round(max(zooptaxa$CYCL), digits = 0)), pch = 15, cex = 1.25,
arrows(x0 = bp, y0 = CYCL.means, y1 = CYCL.means - CYCL.sem, angle = 90, length = 0.1, lwd = 1)
arrows(x0 = bp, y0 = CYCL.means, y1 = CYCL.means + CYCL.sem, angle = 90, length = 0.1, lwd = 1)#More in

```



```
#BOSM
```

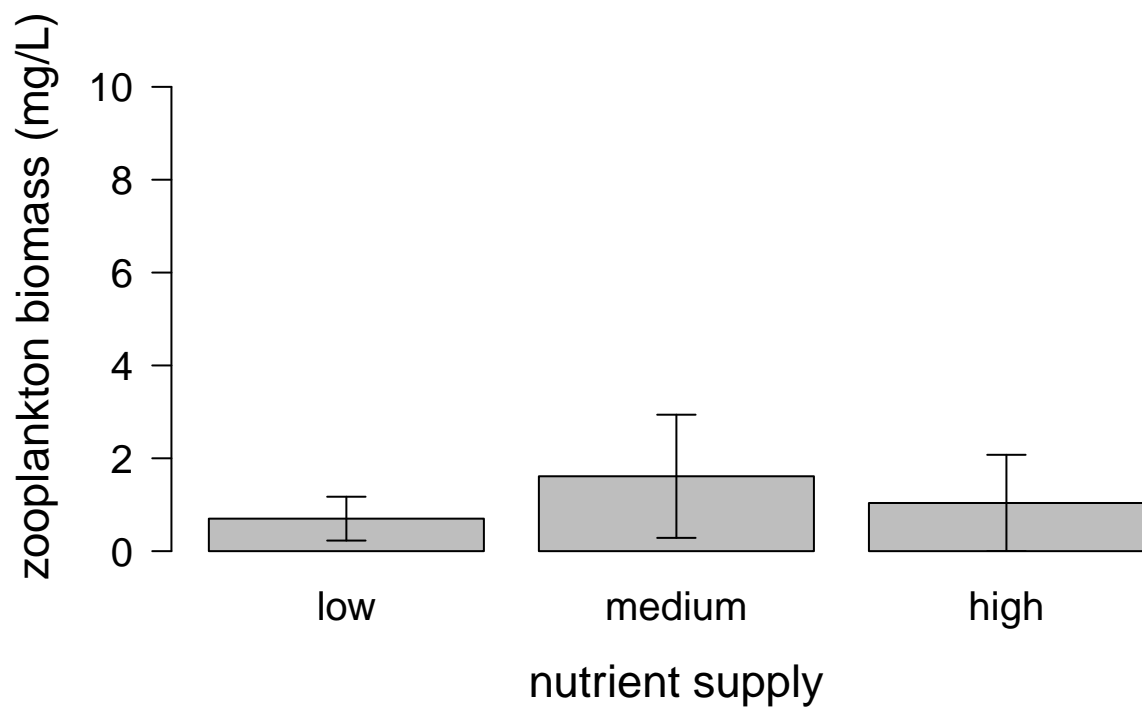
```
BOSM.means <- tapply(zooptaxa$BOSM, NUTS, mean)
```

```
BOSM.sem <- tapply(zooptaxa$BOSM, NUTS, sem)
```

```
bpBOSM <- barplot(BOSM.means, ylim = c(0, round(max(zooptaxa$BOSM), digits = 0)), pch = 15, cex = 1.25, l
```

```
arrows(x0 = bp, y0 = BOSM.means, y1 = BOSM.means - BOSM.sem, angle = 90, length = 0.1, lwd = 1)
```

```
arrows(x0 = bp, y0 = BOSM.means, y1 = BOSM.means + BOSM.sem, angle = 90, length = 0.1, lwd = 1) #More in
```



```
#SIMO
```

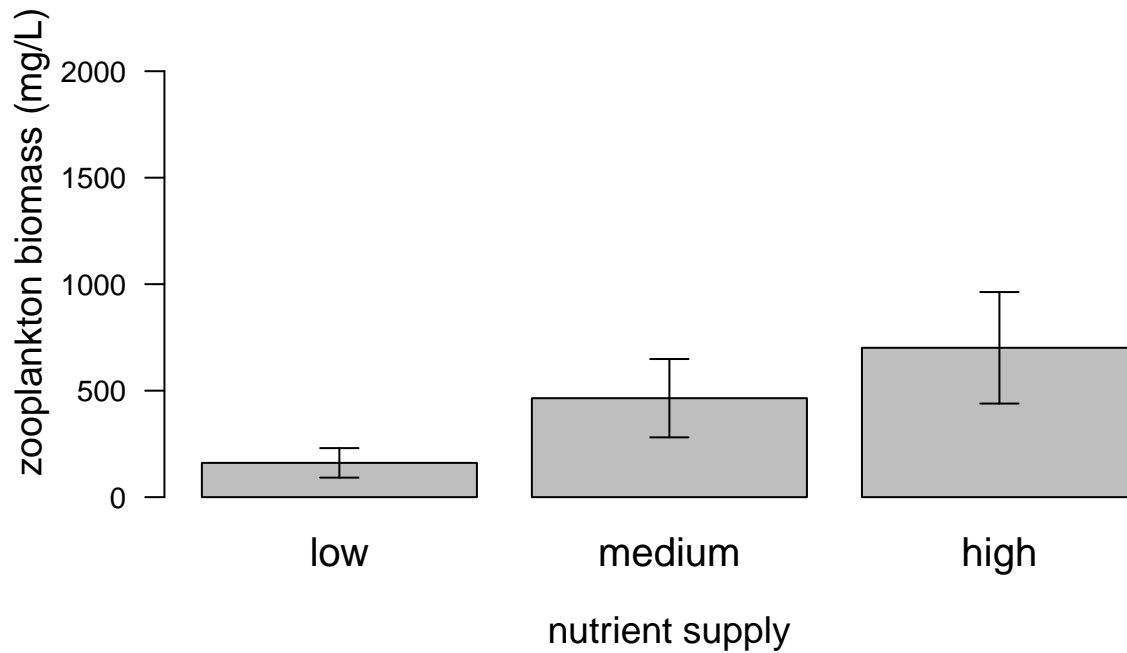
```
SIMO.means <- tapply(zooptaxa$SIMO, NUTS, mean)
```

```
SIMO.sem <- tapply(zooptaxa$SIMO, NUTS, sem)
```

```
bpSIMO <- barplot(SIMO.means, ylim=c(0, round(max(zooptaxa$SIMO), digits = 0)), pch = 15, cex = 1.25, l
```

```
arrows(x0 = bp, y0 = SIMO.means, y1 = SIMO.means - SIMO.sem, angle = 90, length = 0.1, lwd = 1)
```

```
arrows(x0 = bp, y0 = SIMO.means, y1 = SIMO.means + SIMO.sem, angle = 90, length = 0.1, lwd = 1)#More in
```

```
#CERI
```

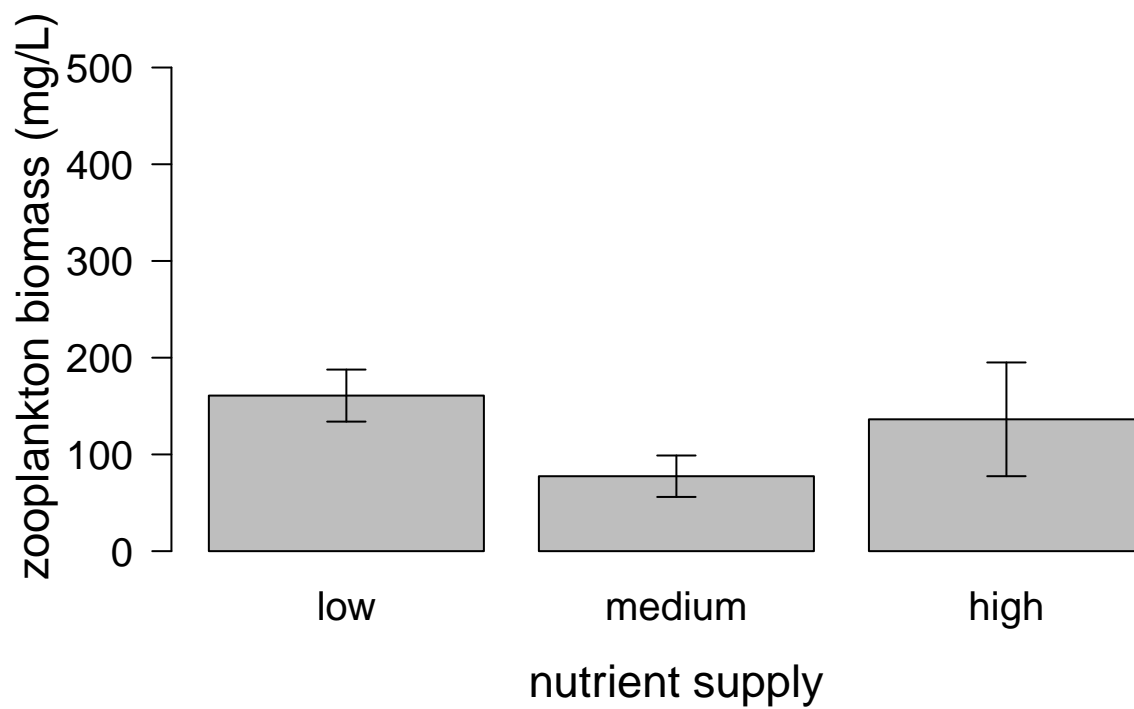
```
CERI.means <- tapply(zooptaxa$CERI, NUTS, mean)
```

```
CERI.sem <- tapply(zooptaxa$CERI, NUTS, sem)
```

```
bpCERI <- barplot(CERI.means, ylim=c(0, round(max(zooptaxa$CERI), digits = 0)), pch = 15, cex = 1.25, l
```

```
arrows(x0 = bp, y0 = CERI.means, y1 = CERI.means - CERI.sem, angle = 90, length = 0.1, lwd = 1)
```

```
arrows(x0 = bp, y0 = CERI.means, y1 = CERI.means + CERI.sem, angle = 90, length = 0.1, lwd = 1) #more in
```



```
#NAUP
```

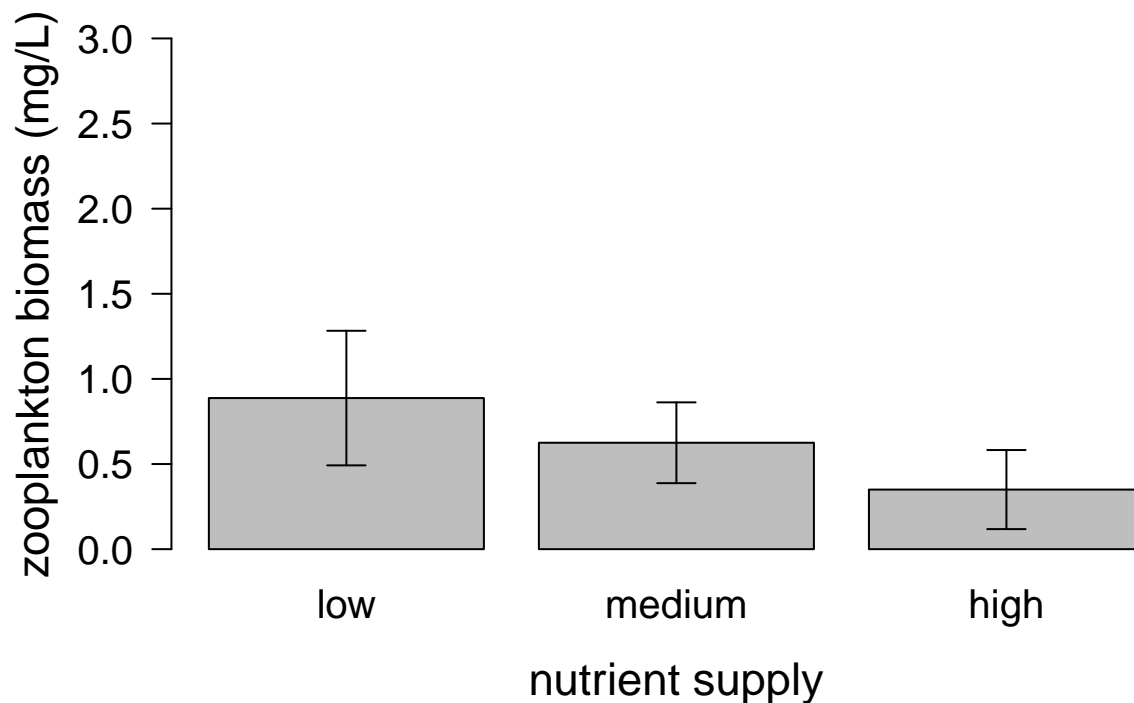
```
NAUP.means <- tapply(zooptaxa$NAUP, NUTS, mean)
```

```
NAUP.sem <- tapply(zooptaxa$NAUP, NUTS, sem)
```

```
bpNAUP <- barplot(NAUP.means, ylim=c(0, round(max(zooptaxa$NAUP), digits = 0)), pch = 15, cex = 1.25, l
```

```
arrows(x0 = bp, y0 = NAUP.means, y1 = NAUP.means - NAUP.sem, angle = 90, length = 0.1, lwd = 1)
```

```
arrows(x0 = bp, y0 = NAUP.means, y1 = NAUP.means + NAUP.sem, angle = 90, length = 0.1, lwd = 1)#More in
```



```
#DLUM
```

```
DLUM.means <- tapply(zooptaxa$DLUM, NUTS, mean)
```

```
DLUM.sem <- tapply(zooptaxa$DLUM, NUTS, sem)
```

```
bpDLUM <- barplot(DLUM.means, ylim=c(0, round(max(zooptaxa$DLUM), digits = 0)), pch = 15, cex = 1.25,
arrows(x0 = bp, y0 = DLUM.means, y1 = DLUM.means - DLUM.sem, angle = 90, length = 0.1, lwd = 1)
```

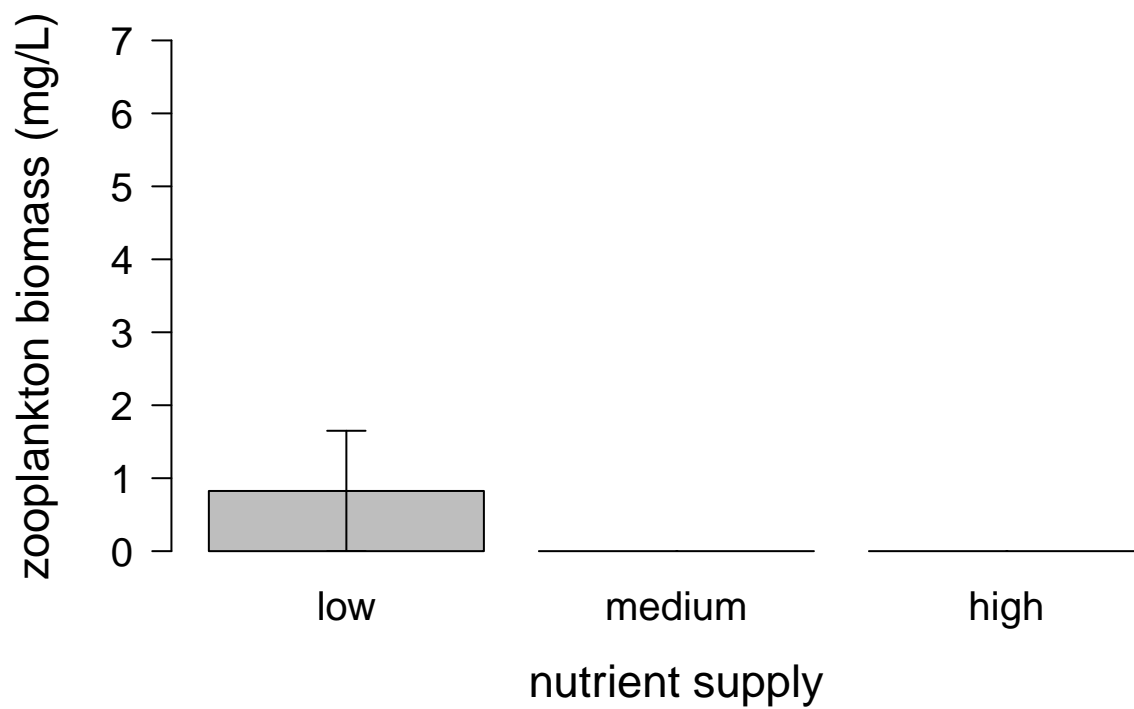
```
## Warning in arrows(x0 = bp, y0 = DLUM.means, y1 = DLUM.means - DLUM.sem, : zero-
## length arrow is of indeterminate angle and so skipped
```

```
## Warning in arrows(x0 = bp, y0 = DLUM.means, y1 = DLUM.means - DLUM.sem, : zero-
## length arrow is of indeterminate angle and so skipped
```

```
arrows(x0 = bp, y0 = DLUM.means, y1 = DLUM.means + DLUM.sem, angle = 90, length = 0.1, lwd = 1)#Only in
```

```
## Warning in arrows(x0 = bp, y0 = DLUM.means, y1 = DLUM.means + DLUM.sem, : zero-
## length arrow is of indeterminate angle and so skipped
```

```
## Warning in arrows(x0 = bp, y0 = DLUM.means, y1 = DLUM.means + DLUM.sem, : zero-
## length arrow is of indeterminate angle and so skipped
```



```
#CHYD
```

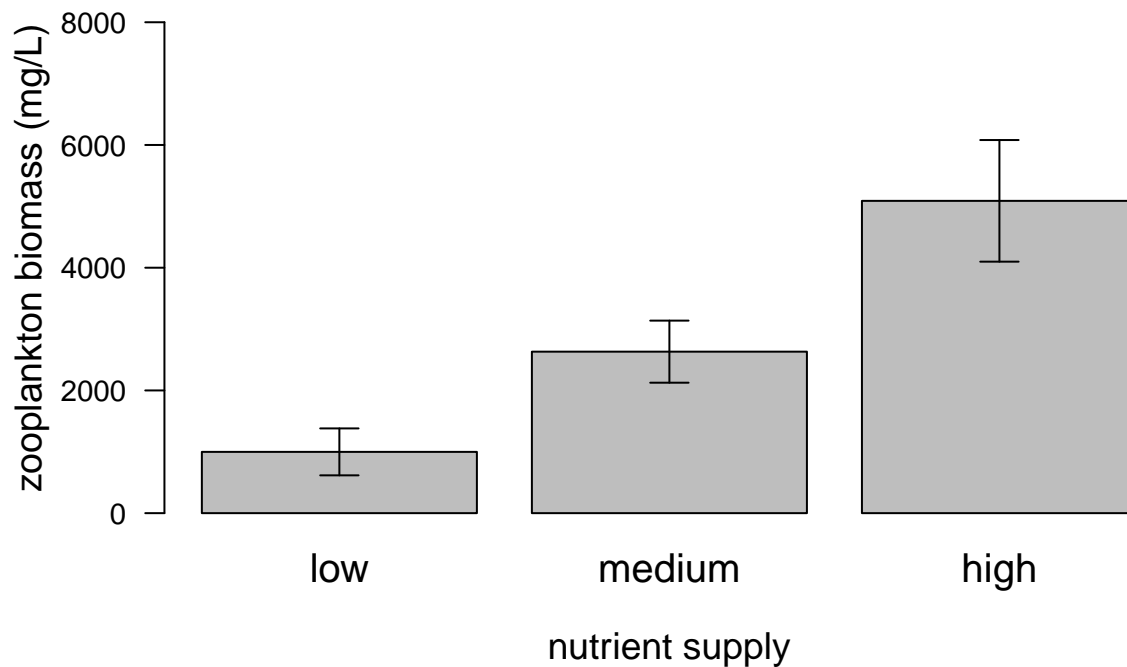
```
CHYD.means <- tapply(zooptaxa$CHYD, NUTS, mean)
```

```
CHYD.sem <- tapply(zooptaxa$CHYD, NUTS, sem)
```

```
bpCHYD <- barplot(CHYD.means, ylim=c(0, round(max(zooptaxa$CHYD), digits = 0)), pch = 15, cex = 1.25, l
```

```
arrows(x0 = bp, y0 = CHYD.means, y1 = CHYD.means - CHYD.sem, angle = 90, length = 0.1, lwd = 1)
```

```
arrows(x0 = bp, y0 = CHYD.means, y1 = CHYD.means + CHYD.sem, angle = 90, length = 0.1, lwd = 1)#More in
```



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)**.

Answer to Question 6: By looking at barplots of different species biomasses at different nutrition levels I can see that different taxa do, visually speaking, differ. I was running anovas on taxa by nutrition level then realized that some of the taxa data might not meet the assumptions of an anova. By looking at the barplots I can say that different taxa differ in how much biomass they contribute to the different nutrition levels. For example, CHYD seem to contribute more biomass at High levels of nutrition compared to DLUM which only contribute biomass at Low levels of nutrition.