j— title: '3. Worksheet: Basic R' author: "Delaney Miller; Z620: Quantitative Biodiversity, Indiana University" date: "16 January, 2019" output: pdf_document geometry: margin=2.54cm —

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. 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.
5. Before you leave the classroom today, it is *imperative* that you **push** this file to your GitHub repo.
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 16th, 2019 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.

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/15053/Desktop"
```

```
setwd("~/QB2019_Miller/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. 45Â°) and with hypotenuse length sqrt(2) (remember: sin(theta) = opposite/hypotenuse).
4) the log (base e) of your favorite number.

```
#1
l<-5
volume <- l^3
#2
r<-2
area=pi*r^2
#3
theta<-pi/4
hypotenuse=sqrt(2)
```

```r
opposite=sin(theta)/hypotenuse
#4
log(12)
```

```
## [1] 2.484907
```

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.

```r
#1
x<-c(12,17,21,52,77)
#2
w<-x*14
#3
(x+w)/15
```

```
## [1] 12 17 21 52 77
```

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`.

```r
#1
x<-c(12,17,21,52,77)
#2
w<-x*14
#3
(x+w)/15
```

```
## [1] 12 17 21 52 77
```

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.
max(v)
```

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

```
min(v)
```

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

```
sum(v)
```

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

```
mean(v)
```

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

```
median(v)
```

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

```
var(v)
```

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

```
sd(v)
```

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

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.

```
m1 <- c(rnorm(5, mean = 8, sd=2))
m2 <- c(rnorm(5, mean=25, sd=10))
```

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:It randomly generates a normal distribution with the designated mean and standard deviation (sd) specified as arguments of the fucntion rnorm()

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.

```
setwd("~/QB2019_Miller/2.Worksheets/3.RStudio/data")
data=read.table("matrix.txt", header=FALSE)
data
```

```
##    V1 V2 V3 V4 V5
## 1   8  1  7  6  1
## 2   5  5  2  4  1
## 3   2  5  4  3  3
## 4   3  2  5  1  4
## 5   9  9  1  1  2
## 6  11  8  1  8  8
## 7   2  2  5  8  5
## 8   3  3  6  7  6
## 9   5  5  1  3  6
## 10  6  5  9  2  2
```

```
t(data)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## V1    8    5    2    3    9   11    2    3    5     6
## V2    1    5    5    2    9    8    2    3    5     5
## V3    7    2    4    5    1    1    5    6    1     9
## V4    6    4    3    1    1    8    8    7    3     2
## V5    1    1    3    4    2    8    5    6    6     2
```

```
dim(data)
```

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

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

> Answer 2: 10 rows, 5 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`.

```
data[,-3]
```

```
##    V1 V2 V4 V5
## 1   8  1  6  1
## 2   5  5  4  1
## 3   2  5  3  3
## 4   3  2  1  4
## 5   9  9  1  2
## 6  11  8  8  8
## 7   2  2  8  5
## 8   3  3  7  6
## 9   5  5  3  6
## 10  6  5  2  2
```

```
data[-10,]
```

```
##   V1 V2 V3 V4 V5
## 1  8  1  7  6  1
## 2  5  5  2  4  1
## 3  2  5  4  3  3
## 4  3  2  5  1  4
## 5  9  9  1  1  2
## 6 11  8  1  8  8
## 7  2  2  5  8  5
## 8  3  3  6  7  6
## 9  5  5  1  3  6
```

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.

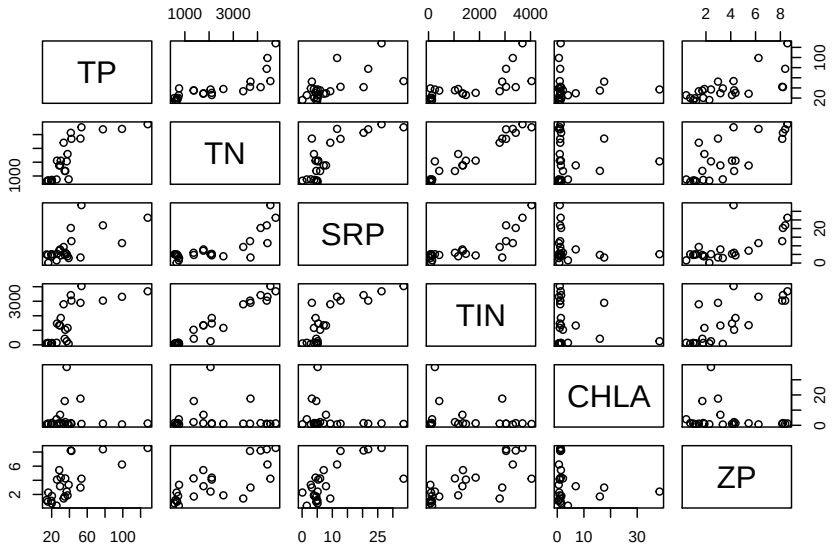
```
setwd("~/QB2019_Miller/2.Worksheets/3.RStudio/data")
meso <- read.table("zoop_nuts.txt", header=TRUE)
str(meso)
```

```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: Factor w/ 3 levels "H","L","M": 2 2 2 2 2 2 2 2 3 3 ...
##  $ 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]
pairs(meso.num)
```




```
cor1 <- cor(meso.num)
```

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

> Answer 3: Some positive correlations include TP and TN, TN and SRP, and TN and TIN. Factors such as TN appear to explain a lot of variation in the other categories, whereas one like CHLA does not show a correlation with other variables.

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.

```
r = getOption("repos")
r["CRAN"] = "http://cran.us.r-project.org"
options(repos = r)
install.packages("psych")
```

```
## Installing package into 'C:/Users/15053/Documents/R/win-library/3.5'
## (as 'lib' is unspecified)
```

```
## package 'psych' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\15053\AppData\Local\Temp\Rtmp29dcEX\downloaded_packages
```

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

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

```
## Warning: package 'psych' was built under R version 3.5.2
```

```
#pearson's correlation test
cor2<- corr.test(meso.num, method="pearson", adjust = "BH")
#kendall's correlation test
cor3 <- corr.test(meso.num, method="kendall", adjust="BH")
#print correlation coefficients
print(cor2, digits=3)
```

```
## 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
```

```
print(cor3, digits=3)
```

```
## Call:corr.test(x = meso.num, method = "kendall", 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
```

```
#visualize data
install.packages("corrplot") #install package
```

```
## Installing package into 'C:/Users/15053/Documents/R/win-library/3.5'
## (as 'lib' is unspecified)
```

```
## package 'corrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\15053\AppData\Local\Temp\Rtmp29dcEX\downloaded_packages
```

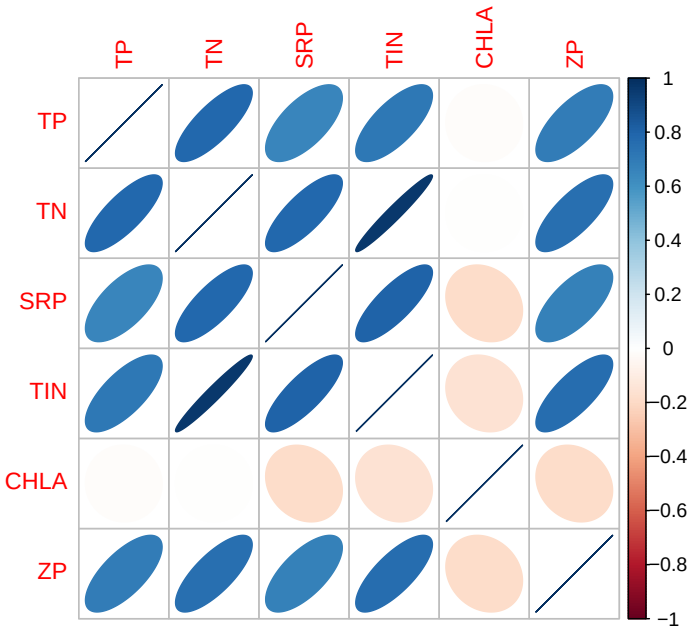
```
require("corrplot")
```

```
## Loading required package: corrplot

## Warning: package 'corrplot' was built under R version 3.5.2

## 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: Yes, using parametric vs non-parametric methods will skew the magnitude of the correlations. Non-parametric methods

should be used if you have a small sample size or if your data doesn't follow a normal distribution (i.e. has a long tail or is heavily skewed by outliers) Any test has the risk of some false discovery rate, but this is hopefully mitigated by adjusting for multiple tests (adjust="BH") False discovery rates are important because they assign signficance to correlations or interactions which are not.

Linear Regression

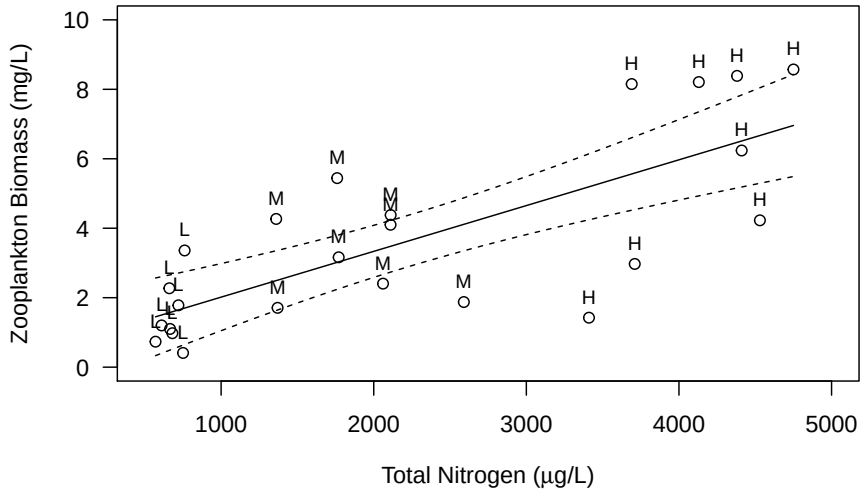
In the R code chunk below, do the following: 1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zooplankton biomass (ZP). 2) Examine the output of the regression analysis. 3) Produce a plot of this regression analysis including the following: categorically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

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

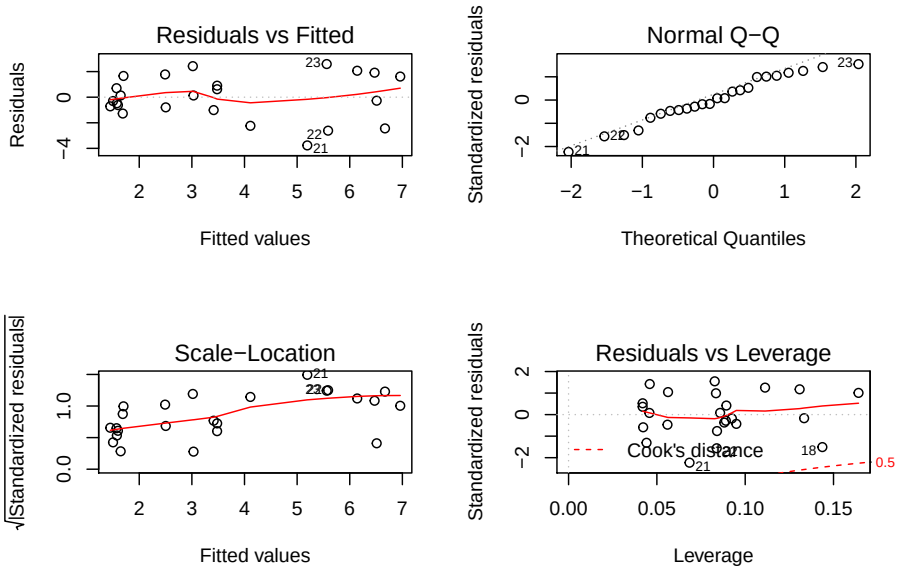
```
##
## Call:
## lm(formula = ZP ~ TN, data = meso)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7690 -0.8491 -0.0709  1.6238  2.5888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6977712  0.6496312   1.074    0.294
## TN          0.0013181  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
plot (meso$TN, meso$ZP, ylim=c(0,10), xlim=c(500,5000), xlab=expression(paste("Total Nitroge
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=0.9
matlines(newTN, conf95[,c("lwr", "upr")], type="l", lty=2, lwd=1, col="black")
```




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



Question 5: Interpret the results from the regression model

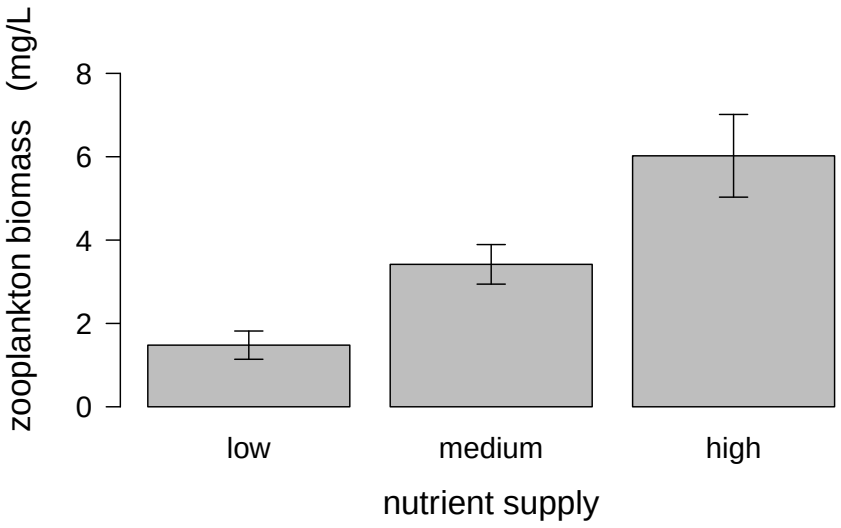
> Answer 5: There is a positive correlation between total nitrogen and zooplankton mass.

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.

```
#order nutrient treatments
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$ZP, NUTS, sem)
#plot
bp<-barplot(zp.means, ylim=c(0, round(max(meso$ZP), digits=0)),
            pch=15, cex=1.25, las=1, cex.lab=1.4, cex.axis=1.25, xlab="nutrient supply", yla
```

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

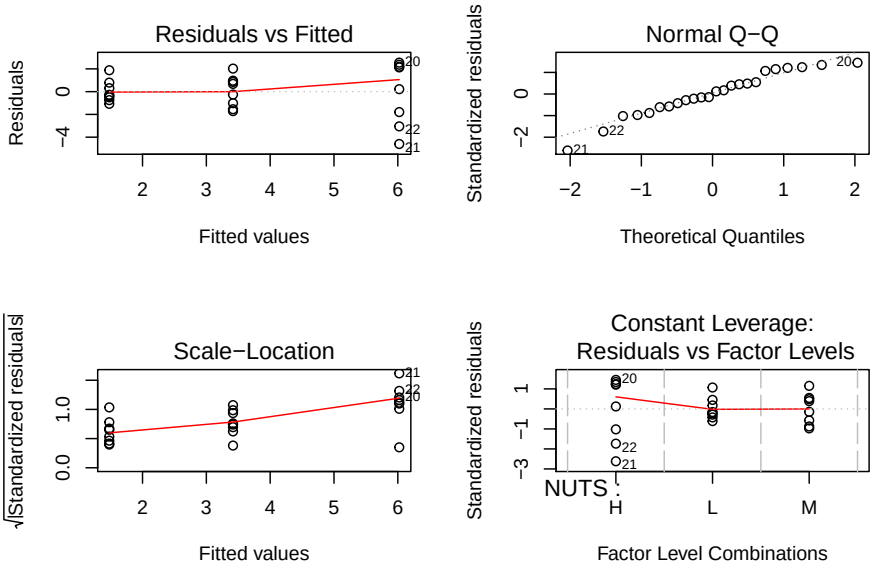
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## NUTS         2  83.15   41.58   11.77 0.000372 ***
## Residuals   21  74.16    3.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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
```

```
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 zoop.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 (Âµg/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.

```
setwd("~/QB2019_Miller/2.Worksheets/3.RStudio/data")
data2 <-read.table("zoops.txt", header=TRUE)
str(data2)
```

```
## 'data.frame':    24 obs. of  11 variables:
##  $ TANK: int  5 14 16 21 23 25 27 34 12 15 ...
##  $ NUTS: Factor w/ 3 levels "H","L","M": 2 2 2 2 2 2 2 2 3 3 ...
##  $ 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 ...
```

```
data3=data2[,3:9]
#plot
data3$totalAb = rowSums(data3)

install.packages("reshape2")
```

```
## Installing package into 'C:/Users/15053/Documents/R/win-library/3.5'
## (as 'lib' is unspecified)
```

```
## package 'reshape2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\15053\AppData\Local\Temp\Rtmp29dcEX\downloaded_packages
```

```
library("reshape2")
```

```
## Warning: package 'reshape2' was built under R version 3.5.2
```

```
require(reshape2)
data3.long= melt(data3)
```

```
## No id variables; using all as measure variables
```

```
data3.long
```

```
##     variable  value
## 1        CAL   70.5
## 2        CAL   27.1
## 3        CAL    5.3
## 4        CAL   79.2
## 5        CAL   31.4
## 6        CAL   22.7
## 7        CAL    0.0
## 8        CAL   35.7
## 9        CAL   74.8
## 10       CAL    5.3
## 11       CAL   18.4
## 12       CAL   14.0
## 13       CAL   14.0
## 14       CAL   48.8
## 15       CAL    0.0
## 16       CAL  292.0
## 17       CAL    9.7
## 18       CAL    0.0
## 19       CAL    5.3
## 20       CAL   14.0
## 21       CAL    0.0
## 22       CAL    0.0
## 23       CAL    5.3
## 24       CAL    0.0
## 25      DIAP    0.0
## 26      DIAP   19.2
## 27      DIAP    8.8
## 28      DIAP   17.9
## 29      DIAP    0.0
## 30      DIAP  285.1
## 31      DIAP    2.3
```

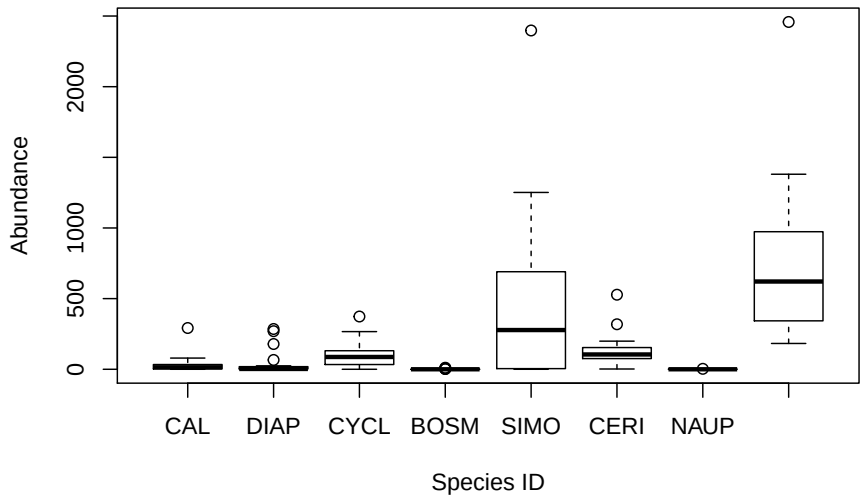
```
## 32      DIAP   65.9
## 33      DIAP  178.7
## 34      DIAP    4.9
## 35      DIAP    2.3
## 36      DIAP    2.3
## 37      DIAP    2.3
## 38      DIAP    2.3
## 39      DIAP    0.0
## 40      DIAP  269.5
## 41      DIAP    0.0
## 42      DIAP    2.3
## 43      DIAP    0.0
## 44      DIAP    7.5
## 45      DIAP   24.4
## 46      DIAP    7.5
## 47      DIAP    2.3
## 48      DIAP    2.3
## 49      CYCL   66.1
## 50      CYCL  129.6
## 51      CYCL   12.7
## 52      CYCL  141.3
## 53      CYCL   11.0
## 54      CYCL  153.0
## 55      CYCL   11.0
## 56      CYCL  102.9
## 57      CYCL  266.5
## 58      CYCL   87.8
## 59      CYCL   29.4
## 60      CYCL   37.7
## 61      CYCL  132.9
## 62      CYCL  107.9
## 63      CYCL   17.7
## 64      CYCL  373.4
## 65      CYCL   41.1
## 66      CYCL    0.0
## 67      CYCL   86.2
## 68      CYCL   69.5
## 69      CYCL  101.2
## 70      CYCL  253.2
## 71      CYCL   96.2
## 72      CYCL   66.1
## 73      BOSM    2.2
## 74      BOSM    0.0
## 75      BOSM    0.0
## 76      BOSM    3.4
## 77      BOSM    0.0
```

```
## 78      BOSM    0.0
## 79      BOSM    0.0
## 80      BOSM    0.0
## 81      BOSM    0.0
## 82      BOSM    0.0
## 83      BOSM    0.0
## 84      BOSM    0.0
## 85      BOSM    0.0
## 86      BOSM    2.2
## 87      BOSM    0.0
## 88      BOSM   10.7
## 89      BOSM    0.0
## 90      BOSM    0.0
## 91      BOSM    0.0
## 92      BOSM    0.0
## 93      BOSM    0.0
## 94      BOSM    8.3
## 95      BOSM    0.0
## 96      BOSM    0.0
## 97      SIMO  417.8
## 98      SIMO    0.0
## 99      SIMO   73.1
## 100     SIMO    0.0
## 101     SIMO  482.0
## 102     SIMO  241.5
## 103     SIMO   73.1
## 104     SIMO    0.0
## 105     SIMO    0.0
## 106     SIMO 1099.2
## 107     SIMO  393.8
## 108     SIMO 1251.5
## 109     SIMO  818.6
## 110     SIMO    9.0
## 111     SIMO  145.3
## 112     SIMO    0.0
## 113     SIMO 2397.8
## 114     SIMO  225.5
## 115     SIMO  465.9
## 116     SIMO  594.2
## 117     SIMO  313.6
## 118     SIMO    0.0
## 119     SIMO  786.6
## 120     SIMO  826.7
## 121     CERI  159.8
## 122     CERI   79.4
## 123     CERI  107.5
```

```
## 124     CERI  199.0
## 125     CERI  101.9
## 126     CERI  135.5
## 127     CERI  185.0
## 128     CERI  318.5
## 129     CERI    1.9
## 130     CERI  136.4
## 131     CERI  147.6
## 132     CERI   74.8
## 133     CERI   98.1
## 134     CERI  132.7
## 135     CERI   19.7
## 136     CERI    8.5
## 137     CERI    9.4
## 138     CERI   24.3
## 139     CERI  527.7
## 140     CERI   78.5
## 141     CERI  176.6
## 142     CERI  112.1
## 143     CERI   76.6
## 144     CERI   85.1
## 145     NAUP    0.0
## 146     NAUP    0.0
## 147     NAUP    1.2
## 148     NAUP    0.0
## 149     NAUP    0.0
## 150     NAUP    1.2
## 151     NAUP    1.6
## 152     NAUP    3.1
## 153     NAUP    0.0
## 154     NAUP    1.4
## 155     NAUP    1.2
## 156     NAUP    0.0
## 157     NAUP    1.2
## 158     NAUP    0.0
## 159     NAUP    0.0
## 160     NAUP    1.2
## 161     NAUP    0.0
## 162     NAUP    0.0
## 163     NAUP    1.2
## 164     NAUP    0.0
## 165     NAUP    0.0
## 166     NAUP    1.6
## 167     NAUP    0.0
## 168     NAUP    0.0
## 169  totalAb  716.4
```

```
## 170  totalAb  255.3
## 171  totalAb  208.6
## 172  totalAb  440.8
## 173  totalAb  626.3
## 174  totalAb  839.0
## 175  totalAb  273.0
## 176  totalAb  526.1
## 177  totalAb  521.9
## 178  totalAb 1335.0
## 179  totalAb  592.7
## 180  totalAb 1380.3
## 181  totalAb 1067.1
## 182  totalAb  302.9
## 183  totalAb  182.7
## 184  totalAb  955.3
## 185  totalAb 2458.0
## 186  totalAb  252.1
## 187  totalAb 1086.3
## 188  totalAb  763.7
## 189  totalAb  615.8
## 190  totalAb  382.7
## 191  totalAb  967.0
## 192  totalAb  980.2
```

```
plot(data3.long, xlab= "Species ID", ylab= "Abundance")
```



```
#The genus Simocephallus explains most of the biomass response to nutrient enrichment.
```

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 16th, 2015 at 12:00 PM (noon)**.