Week 1 Assignment: Basic R

Matthew Gibson; Z620: Quantitative Biodiversity, Indiana University

16 January, 2017

OVERVIEW

Week 1 Assignment introduces some of the basic features of the R computing environment (http://www.r-project.org). It is designed to be used along side your Week 1 Handout (hard copy). You will not be able to complete the exercise if you do not have your handout.

Directions:

1. Change "Student Name" on line 3 (above) with your name.
2. Complete as much of the assignment as possible during class; what you do not complete in class will need to be done on your own outside of 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 exercise.
4. Be sure to **answer the questions** in this assignment document. 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 assignment, **Knit** the text and code into a single PDF file. Basically, just press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Week1 folder.
7. After Knitting, please submit the completed exercise by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (*Week1_Assignment.Rmd*; with all code blocks filled out and questions answered) and the PDF output of `Knitr` (*Week1_Assignment.pdf*).

The completed exercise is due on **Wednesday, January 18th, 2017 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 assignment.

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 Week1 folder.

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

```
## [1] "C:/Users/matth/Documents/bin/QB2017_Gibson/Week1"
```

```
setwd("c:/Users/matth/Documents/bin/QB2017_Gibson/Week1")
```

3) USING R AS A CALCULATOR

To follow up on the Week 0 exercises, please calculate the following in the R code chunk below. Feel free to reference the Week 0 handout.

1) the volume of a cube with length, l, = 5.
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.

```
5^3
```

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

```
pi*2^2
```

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

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

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

```
log(36)
```

```
## [1] 3.583519
```

4) WORKING WITH VECTORS

To follow up on the Week 0 exercises, please perform the requested operations in the Rcode chunks below. Feel free to reference the Week 0 handout.

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(0,1,2,3,4)

w <- 14*x

(x+w)/15
```

```
## [1] 0 1 2 3 4
```

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(5,6,7,8,9)
k*x
```

```
## [1]  0  6 14 24 36
```

```
d <- c(w[0],w[1],w[2],k[0],k[1],k[2],k[3])
d
```

```
## [1]  0 14  5  6  7
```

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)
#Omitting the missing value
max(na.omit(v))
```

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

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

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

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

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

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

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

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

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

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

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

```r
sd(na.omit(v))
```

```
## [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 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.

```r
col_1 <- c(rnorm(5, mean=8, sd=2))
col_2 <- c(rnorm(5, mean=25, sd=10))
newMatrix <- cbind(col_1, col_2)
newMatrix
```

```
##         col_1    col_2
## [1,] 7.018020 18.77459
## [2,] 7.602695 12.56137
## [3,] 6.598404 20.05851
## [4,] 7.740623 20.25662
## [5,] 9.165185 29.26588
```

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 vector of random numbers of length n from a normal distribution with a specific mean and standard deviation. The first argument specifies the length of the vector to generate. The second argument specifies the mean. The third argument specifies the standard deviation.

In the R code chunk below, do the following: 1) Load `matrix.txt` from the Week1 data folder as matrix `m`. 2) Transpose this matrix. 3) Determine the dimensions of the transposed matrix.

```r
m <- read.table("data/matrix.txt", sep = "\t", header = F)
dim(t(m))
```

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

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

> Answer 2: 5x10. 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`.

```
m[ ,c(1:2, 4:5)]
```

```
##    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
```

```
m[,-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
```

Question 3: Describe what we just did in the last series of indexing steps.

> ***Answer 3***: We first selected only select columns of the matrix `m`. Second, we displayed the matrix `m` missing its third column. These two steps did the exact same thing...except one is shorter.

6) BASIC DATA VISUALIZATION AND STATISTICAL ANALYSIS

Load Zooplankton Dataset

In the R code chunk below, do the following: 1) Load the zooplankton dataset from the Week1 data folder. 2) Display the structure of this data set.

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

str(meso)
```

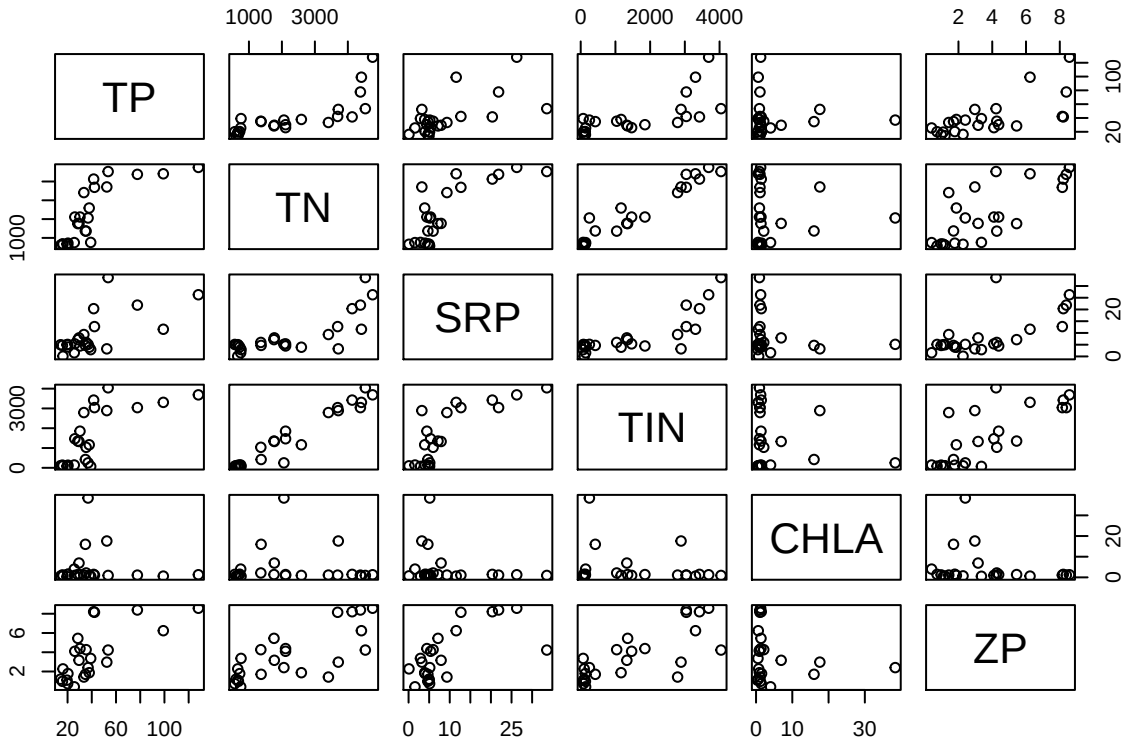
```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: Factor w/ 3 levels "H","L","M": 2 2 2 2 2 2 2 2 3 3 ...
##  $ TP  : num  20.3 25.6 14.2 39.1 20.1 ...
##  $ TN  : num  720 750 610 761 570 ...
```

```
##  $ 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)
cor1
```

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

```
##              ZP
## TP    0.6974765
## TN    0.7562474
## SRP   0.6762947
## TIN   0.7605629
## CHLA -0.1825999
## ZP    1.0000000
```

Question 4: Describe some of the general features based on the visualization and correlation analysis above?

> Answer 4: The variables `TN` and `TIN` show a strong positive linear relationship as seen in both the scatterplot matrix and the correlation analysis. `TIN` is also positively correlated with `SRP` though this relationship does not appear strictly linear. `TN` and `TP` have a positive, though not linear, relationship.

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.

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.2.5
```

```
cor2 <- corr.test(meso.num, method="pearson", adjust="BH")
cor2
```

```
## Call:corr.test(x = meso.num, method = "pearson", adjust = "BH")
## Correlation matrix
##         TP   TN   SRP   TIN  CHLA    ZP
## TP    1.00 0.79  0.65  0.72 -0.02  0.70
## TN    0.79 1.00  0.78  0.97  0.00  0.76
## SRP   0.65 0.78  1.00  0.80 -0.19  0.68
## TIN   0.72 0.97  0.80  1.00 -0.16  0.76
## CHLA -0.02 0.00 -0.19 -0.16  1.00 -0.18
## ZP    0.70 0.76  0.68  0.76 -0.18  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 0.00 0.00 0.98 0.00
## TN   0.00 0.00 0.00 0.00 0.98 0.00
## SRP  0.00 0.00 0.00 0.00 0.49 0.00
## TIN  0.00 0.00 0.00 0.00 0.54 0.00
## CHLA 0.94 0.98 0.38 0.46 0.00 0.49
## ZP   0.00 0.00 0.00 0.00 0.39 0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

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

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

Question 5: Describe what you learned from `corr.test`. 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?

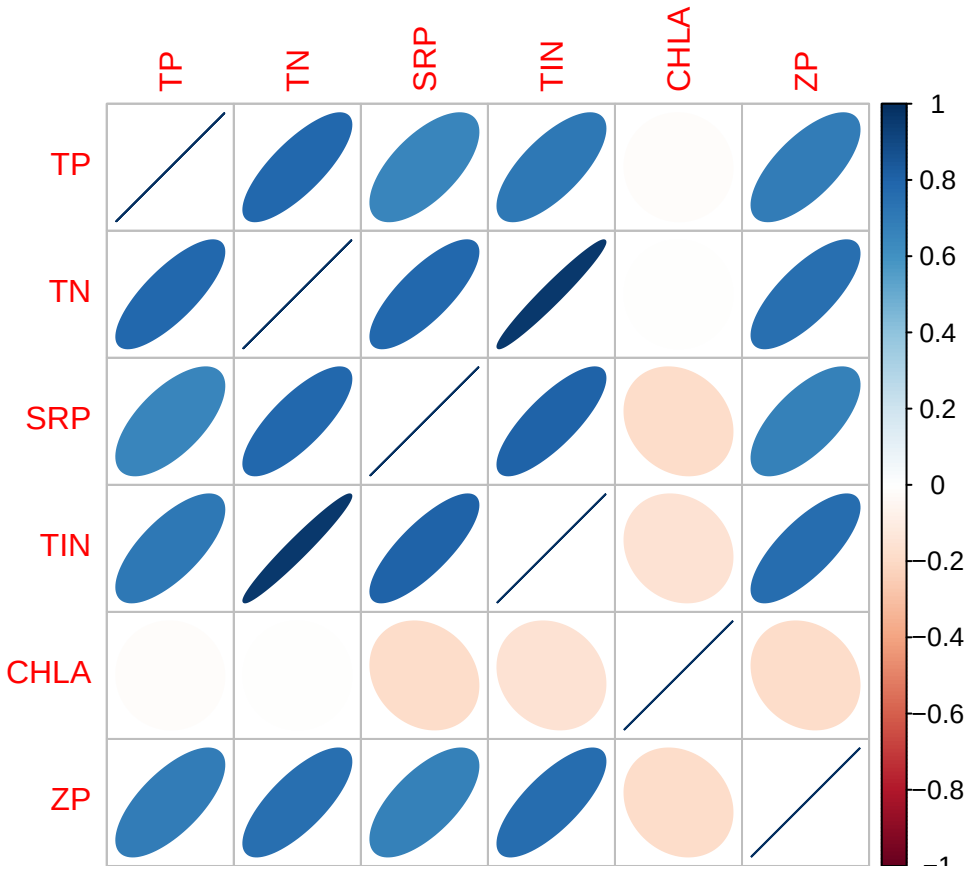
> ***Answer 5***: The results of these tests are sensitive to whether we use parametric or non-parametric methods. Correlation coefficients vary between the two methods as well as p-values for the coefficient being equal to 0. We should use non-parametric methods when you cant assume a normal distribution or constant variance. Non-parametric tests offer more freedom but cannot draw as strong of conclusions. P-values above the diagonal in the Pearson test are higher than the corresponding p-values below the diagonal, but these corrected values do not change any of our conclusions from the hypothesis tests.

In the R code chunk below, use the `corrplot` function in the *corrplot* package to produce the ellipse correlation plot in the handout.

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.2.5
```

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



Linear Regression

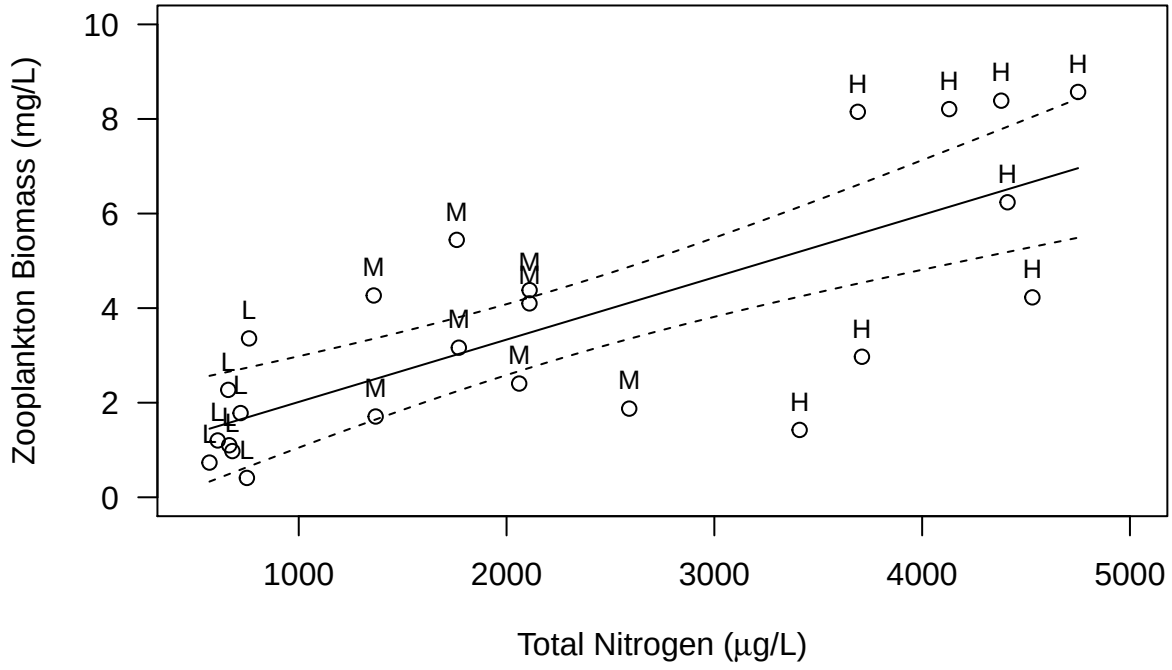
In the R code chunk below, do the following: 1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zooplankton biomass (ZP). 2) Examine the output of the regression analysis. 3) Produce a plot of this regression analysis including the following: categorically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

```
model1 <- lm(ZP ~ TN, data=meso)
summary(model1)
```

```
##
## 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 (", mu,"g/
     ylab = "Zooplankton Biomass (mg/L)", las = 1)

text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)
newTN <- seq(min(meso$TN), max(meso$TN), 10)
regline <- predict(model1, newdata=data.frame(TN=newTN))
lines(newTN, regline)
conf95 <- predict(model1, newdata=data.frame(TN = newTN),
                  interval = c("confidence"), level = 0.95, type="response")
matlines(newTN, conf95[, c("lwr", "upr")], lty=2, lwd=1, col="black")
```



Question 6: Interpret the results from the regression model

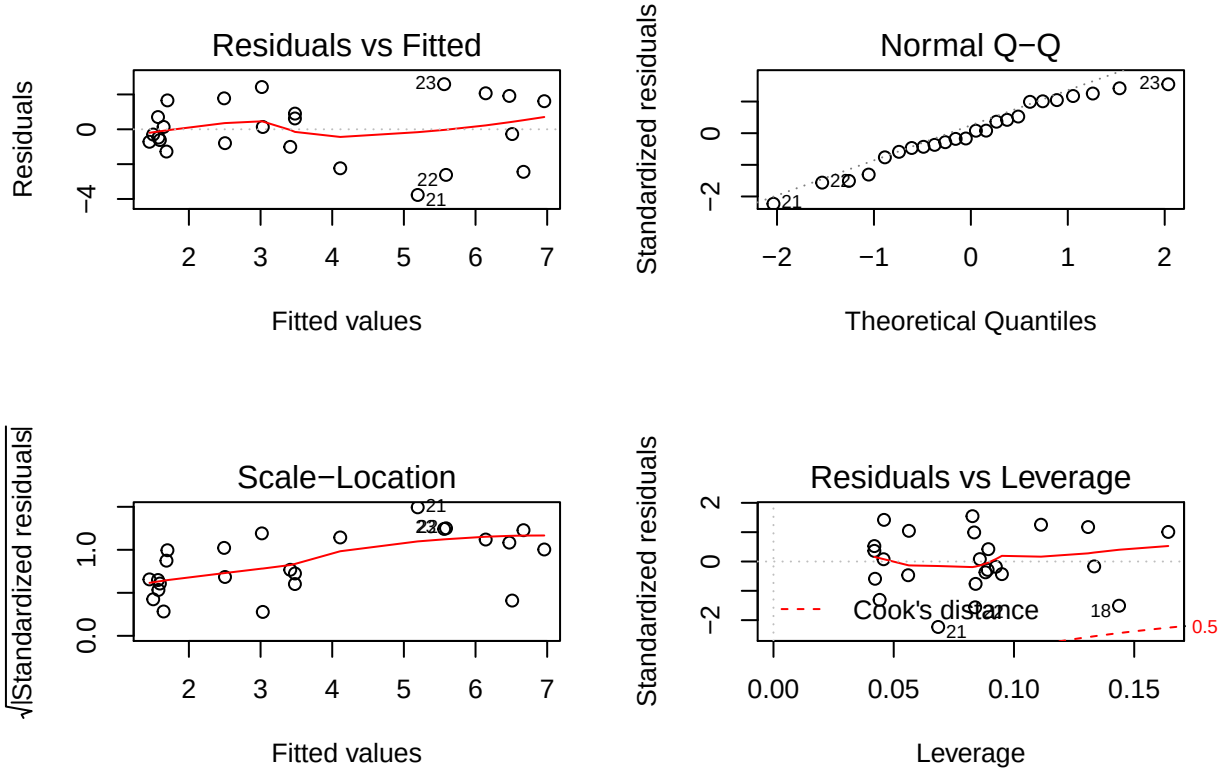
> ***Answer 6***: Zooplankton biomass increases with total nitrogen content. Total nitrogen content is a highly significant predictor of zooplankton biomass (P = 1.91e-5) based on a t-test of the regression coefficient but this simple linear model explains only 57% of the variance in `ZP`. It may be possible to fit a better model.

Question 7: Explain what the `predict()` function is doing in our analyses.

> ***Answer 7***: `predict()` first is used to generated the fitted values to plot our OLS regression line. It is then used to generate 95% confidence intervals for the fitted values across all x.

Using the R code chunk below, use the code provided in the handout to determine if our data meet the assumptions of the linear regression analysis.

```
par(mfrow = c(2,2), mar=c(5.1, 4.1, 4.1, 2.1))
plot(model1)
```



- Upper left: is there a random distribution of the residuals around zero (horizontal line)?
  > No, there is not. Looks like non-constant variance and deviation from normality
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)`

> Looks pretty good. But there may a slight left skew to the distribution.

- Bottom left: again, looking for a random distribution of sqrt(standardized residuals)

> Not quite random. Shows a similar pattern to the top-left plot.

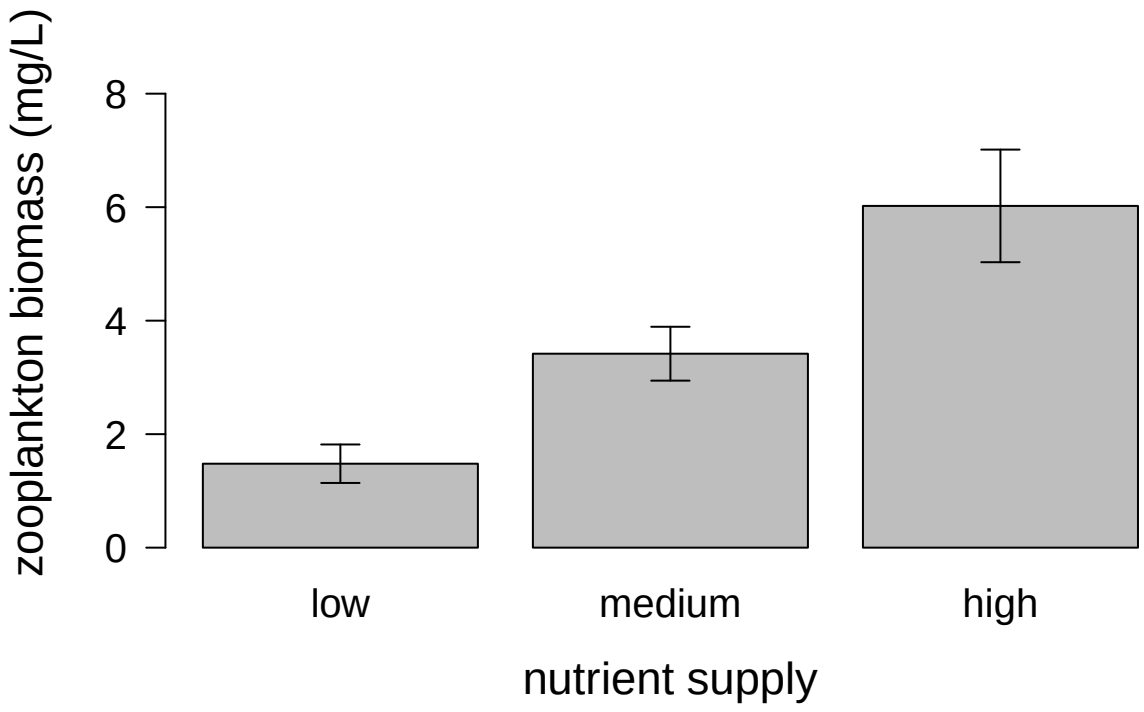
- Bottom right: leverage indicates the influence of points; contours correspond with Cook's distance, where values > |1| are "suspicious"

> Several points are greater than |1|.

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. 5) Use a Tukey's HSD to identify which treatments are different.

```
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, cex.lab = 1.4, cex.axis = 1.25,
              xlab = "nutrient supply", ylab = "zooplankton biomass (mg/L)",
              names.arg = c("low", "medium", "high"))

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
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)
```

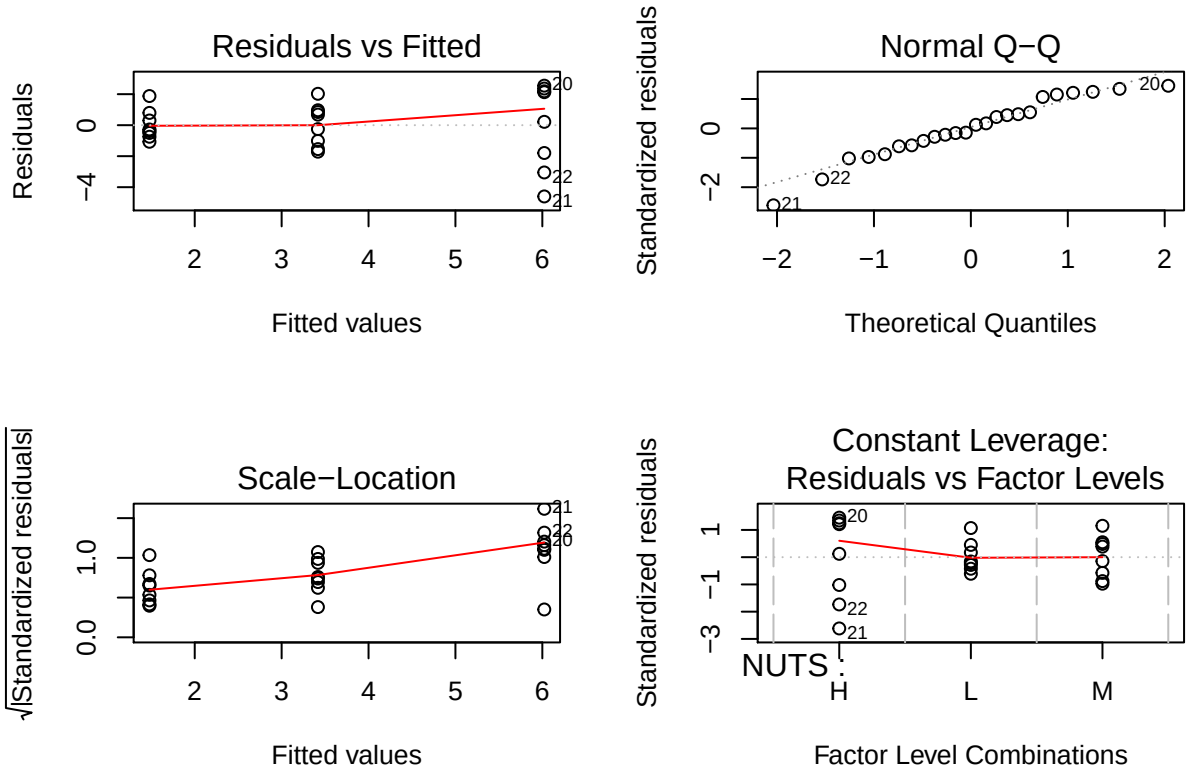
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ZP ~ NUTS, data = meso)
##
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## M-L 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

Question 8: How do you interpret the ANOVA results relative to the regression results? Do you have any concerns about this analysis?

> ***Answer 8***: Based on the barplot, the ANOVA, and the Tukey post-hoc tests, the biomass level for the high treatment is significantly higher than the medium and low treatments. In other words, this analysis is consistent with our regression results that biomass increased with total nitrogen content. Concerns might include the normality of the data or constant variance across treatment groups. We address those concerns below...

Using the R code chunk below, use the diagnostic code provided in the handout to determine if our data meet the assumptions of ANVOA (similar to regression).

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



Answer 8 cont.: Variance does not appear equal across all treatment groups. Seems to be higher in the 'high' nutrient group.

SYNTHESIS: SITE-BY-SPECIES MATRIX

In the R code chunk below, load the zoop.txt dataset in your Week1 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.

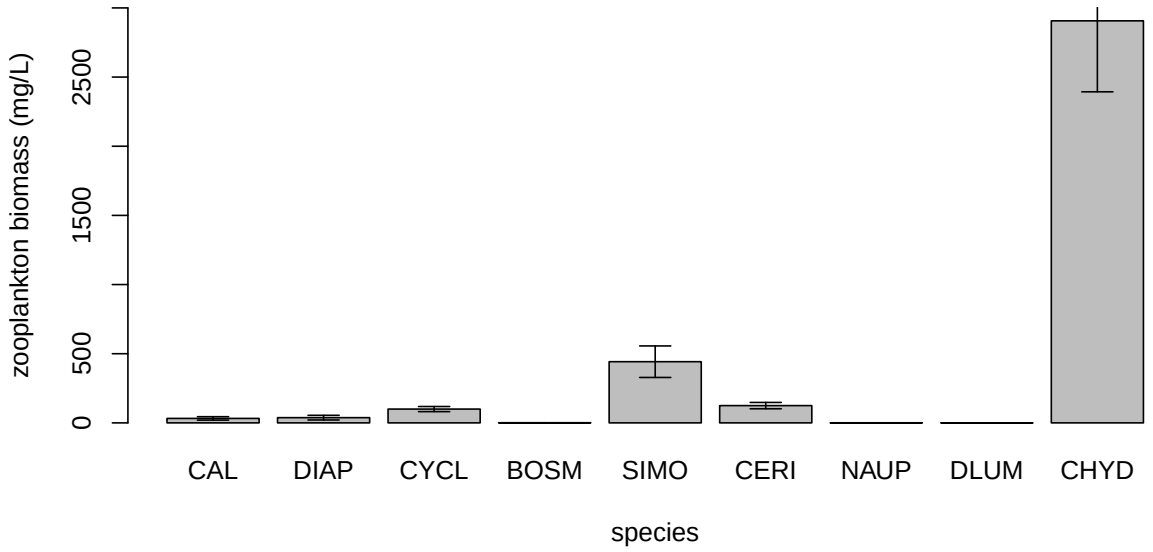
Question 9: With the visualization and statistical tools that we learned about in the Week 1 Handout, use the site-by-species matrix to assess whether and how different zooplankton taxa were responsible for the total biomass (ZP) response to nutrient enrichment. Describe what you learned below in the "Answer" section and include appropriate code in the R chunk.

```
zoop <- read.table("data/zoops.txt", sep = "\t", header = T)
zoop <- zoop[,-1]
zoop <- zoop[,-1]

#Means of each species
sp.means <- colMeans(zoop)

sem <- function(x){
  sd(na.omit(x))/sqrt(length(na.omit(x)))
}

#SEs of each species
sp.sem <- apply(zoop, 2, sem)

#Barplot
bp <- barplot(sp.means, ylim = c(0, 3000),
              xlab = "species", ylab = "zooplankton biomass (mg/L)",
              names.arg = c("CAL", "DIAP", "CYCL", "BOSM", "SIMO", "CERI",
                            "NAUP", "DLUM", "CHYD"))

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

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

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

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

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

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

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

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



Based on this barplot, it seems pretty clear that *Chydorus* is responsible for the total biomass response. Total biomass in *Chydorus* is more than 5X higher than the next highest, *Simocephallus*. We can do an ANOVA and Tukey post-hoc tests to formally test this...

```
#We "stack" the data to make doing an ANOVA with `aov()` easier.
stacked <- stack(zoop)
test <- aov(values~ind, data=stacked)
summary(test)
```

```
##              Df    Sum Sq  Mean Sq F value Pr(>F)
## ind           8 172690808 21586351   29.16 <2e-16 ***
## Residuals   207 153261728   740395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(test)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = values ~ ind, data = stacked)
##
## $ind
##                     diff        lwr        upr     p adj
## CAL-BOSM      31.1125000  -747.5841   809.8091 1.0000000
## CERI-BOSM    123.7416667  -654.9549   902.4383 0.9998995
## CHYD-BOSM   2905.5208333  2126.8242  3684.2174 0.0000000
## CYCL-BOSM     98.6500000  -680.0466   877.3466 0.9999823
## DIAP-BOSM     36.7083333  -741.9883   815.4049 1.0000000
```

```
## DLUM-BOSM    -0.8416667  -779.5383   777.8549 1.0000000
## NAUP-BOSM    -0.4958333  -779.1924   778.2008 1.0000000
## SIMO-BOSM   441.1833333  -337.5133  1219.8799 0.6978319
## CERI-CAL     92.6291667  -686.0674   871.3258 0.9999892
## CHYD-CAL   2874.4083333  2095.7117  3653.1049 0.0000000
## CYCL-CAL     67.5375000  -711.1591   846.2341 0.9999991
## DIAP-CAL      5.5958333  -773.1008   784.2924 1.0000000
## DLUM-CAL    -31.9541667  -810.6508   746.7424 1.0000000
## NAUP-CAL    -31.6083333  -810.3049   747.0883 1.0000000
## SIMO-CAL    410.0708333  -368.6258  1188.7674 0.7752715
## CHYD-CERI  2781.7791667  2003.0826  3560.4758 0.0000000
## CYCL-CERI   -25.0916667  -803.7883   753.6049 1.0000000
## DIAP-CERI   -87.0333333  -865.7299   691.6633 0.9999933
## DLUM-CERI  -124.5833333  -903.2799   654.1133 0.9998942
## NAUP-CERI  -124.2375000  -902.9341   654.4591 0.9998964
## SIMO-CERI   317.4416667  -461.2549  1096.1383 0.9367071
## CYCL-CHYD -2806.8708333 -3585.5674 -2028.1742 0.0000000
## DIAP-CHYD -2868.8125000 -3647.5091 -2090.1159 0.0000000
## DLUM-CHYD -2906.3625000 -3685.0591 -2127.6659 0.0000000
## NAUP-CHYD -2906.0166667 -3684.7133 -2127.3201 0.0000000
## SIMO-CHYD -2464.3375000 -3243.0341 -1685.6409 0.0000000
## DIAP-CYCL   -61.9416667  -840.6383   716.7549 0.9999995
## DLUM-CYCL   -99.4916667  -878.1883   679.2049 0.9999811
## NAUP-CYCL   -99.1458333  -877.8424   679.5508 0.9999816
## SIMO-CYCL   342.5333333  -436.1633  1121.2299 0.9045938
## DLUM-DIAP   -37.5500000  -816.2466   741.1466 1.0000000
## NAUP-DIAP   -37.2041667  -815.9008   741.4924 1.0000000
## SIMO-DIAP   404.4750000  -374.2216  1183.1716 0.7881858
## NAUP-DLUM     0.3458333  -778.3508   779.0424 1.0000000
## SIMO-DLUM   442.0250000  -336.6716  1220.7216 0.6956242
## SIMO-NAUP   441.6791667  -337.0174  1220.3758 0.6965319
```

> We see based on the barplot, ANOVA, and Tukey tests that the species CHYD has a significantly high mean biomass (mg/L) and is therefore likely contributing the most to the diferential biomass response to nutrient levels. The SIMO species also showed a high mean biomass (compared to the other species) but this difference was not significant based on any of the post-hoc tests. We could perform a more thorough analysis of this data looking at individual species responses to nutrient levels.

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed Week1_Assignment.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.

Unless otherwise noted, this assignment is due on **Wednesday, January 18th, 2015 at 12:00 PM (noon)**.