Week 1 Assignment: Basic R

Katie Beidler; Z620: Quantitative Biodiversity, Indiana University

15 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.

```
getwd()
```

```
## [1] "/Users/bhbeidler/GitHub/QB2017_Beidler/Week1"
```

```
setwd("/Users/bhbeidler/GitHub/QB2017_Beidler/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. 45°) and with hypotenuse length sqrt(2) (remember: sin(theta) = opposite/hypotenuse).
4) the log (base e) of your favorite number.

```
# Cube Volume l=5
v_cb = 5*5*5
v_cb
```

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

```
# Circle Area r=2
a_cir = pi * 2^2
a_cir
```

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

```
# Length of the opposite side of a right-triangle (l) when theta = pi/4, hypotenuse = sqrt(2)
theta = pi/4
hyp = sqrt(2)
l = hyp * sin(theta)
l
```

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

```
# the log (base e) of 22
log(22)
```

```
## [1] 3.091042
```

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(5, 12, 300, 19, -4)
w = x *14
x+w/15
```

```
## [1]   9.666667  23.200000 580.000000  36.733333  -7.733333
```

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(9,3,-10,4,210)
k*x
```

```
## [1]    45    36 -3000    76  -840
```

```
d = c(w[1],w[2],w[3],k[1],k[2],k[3],k[4])
d
```

```
## [1]   70  168 4200    9    3  -10    4
```

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))
```

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

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

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

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

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

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

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

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

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

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

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

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

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

```r
sem = function(x){
  sd(na.omit(x))/sqrt(length(na.omit(x)))
  }
sem(v)
```

```
## [1] 1.678435
```

5) WORKING WITH MATRICES

In the R code chunk below, do the following: Using a mixture of Approach 1 and 2 from the 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
c1 = c(rnorm(5), mean = 8, sd = 2)
c2 = c(rnorm(5), mean =25, sd = 10)
cbind(c1,c2)
```

```
##              c1         c2
##      -0.7948739 -0.7260713
##       0.3695065 -0.7556572
##      -1.1704805  0.7790272
##       1.5014507 -0.6865337
##       0.7774976  1.5385778
## mean  8.0000000 25.0000000
## sd    2.0000000 10.0000000
```

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 string of random numbers from a normal distribution- the output is a vector- for which you can specfy the length, mean and 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.

```
m = read.table("./data/matrix.txt")
#transposing the matrix
m = t(m)
dim(m)
```

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

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

Answer 2: The transposed matrix has 10 columns and 5 rows.

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(1:2,4:10)]
```

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

```
#removing the last row...2 ways
m[-5, ]
```

```
##    [,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
```

```
m[-nrow(m), ]
```

```
##    [,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
```

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

Answer 3: We retrieved / removed certain portions of matrix (m). In the first indexing step we eliminated column 3 and in the second step we eliminated row 5.

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.

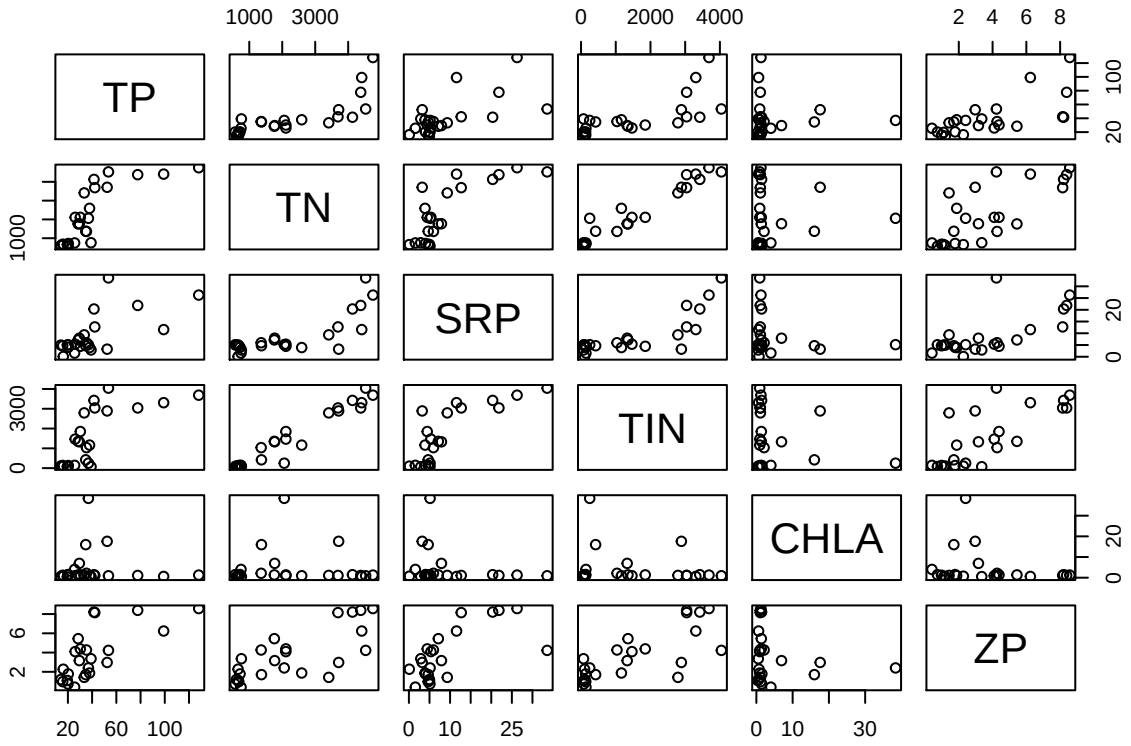
```
zoops = read.table("./data/zoops.txt")
str(zoops)
```

```
## 'data.frame':    25 obs. of  11 variables:
##  $ V1 : Factor w/ 25 levels "10","11","12",..: 25 23 4 6 10 12 14 15 19 3 ...
##  $ V2 : Factor w/ 4 levels "H","L","M","NUTS": 4 2 2 2 2 2 2 2 2 3 ...
##  $ V3 : Factor w/ 15 levels "0.0","14.0","18.4",..: 15 11 5 10 13 7 4 1 8 12 ...
##  $ V4 : Factor w/ 13 levels "0.0","17.9","178.7",..: 13 1 4 12 2 1 8 5 10 3 ...
##  $ V5 : Factor w/ 23 levels "0.0","101.2",..: 23 18 7 6 9 5 10 5 3 13 ...
##  $ V6 : Factor w/ 6 levels "0.0","10.7","2.2",..: 6 3 1 1 4 1 1 1 1 1 ...
##  $ V7 : Factor w/ 19 levels "0.0","1099.2",..: 19 10 1 14 1 12 7 14 1 1 ...
##  $ V8 : Factor w/ 25 levels "1.9","101.9",..: 25 9 20 3 13 2 6 11 15 1 ...
##  $ V9 : Factor w/ 6 levels "0.0","1.2","1.4",..: 6 1 1 2 1 1 2 4 5 1 ...
##  $ V10: Factor w/ 3 levels "0.0","6.6","DLUM": 3 1 1 1 1 1 2 1 1 1 ...
##  $ V11: Factor w/ 25 levels "1190.9","1260.7",..: 25 8 4 14 12 20 7 5 2 1 ...
```

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)
#indexing the numerical data in the 'meso' dataframe
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: Total [N] appears to positively correlated with total [P], soluble reactive [P] (SRP), total inorganic nutrient concentration (TIN) and ZP. There is a strong positive correlation between TIN and total [N] (r=0.96). Correspondingly, total [P] is positively correlated with SRP, TIN and zooplankton biomass (ZP). There is a weak negative correlation between chlorophyll a concentration (CHLA) & total [P], total [N], SRP, TIN, and ZP. Overall zooplankton biomass is positively correlated with the different nutrient concentrations (TN,TP,SRP and TIN).

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

```
##
## The downloaded binary packages are in
##  /var/folders/65/897bx8690x5_qclp18zt0sf80000gn/T//RtmpHMq1NS/downloaded_packages
```

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

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

```
## Warning: package 'psych' was built under R version 3.3.2
```

```
#Parametric Test using Pearson's correlation analysis.
cor2 = corr.test(meso.num, method = "pearson", adjust = "BH")
#Nonparametric Test using Spearman's rho test
cor3 = corr.test(meso.num, method = "spearman", adjust = "BH")

print(cor1, digits = 3)
```

```
##           TP       TN    SRP    TIN     CHLA     ZP
## TP    1.0000  0.78651  0.654  0.717 -0.01666  0.697
## TN    0.7865  1.00000  0.784  0.969 -0.00447  0.756
## SRP   0.6541  0.78419  1.000  0.801 -0.18915  0.676
## TIN   0.7171  0.96900  0.801  1.000 -0.15688  0.761
## CHLA -0.0167 -0.00447 -0.189 -0.157  1.00000 -0.183
## ZP    0.6975  0.75625  0.676  0.761 -0.18260  1.000
```

```
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
```

```
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##           lower      r upper     p
## TP-TN     0.561  0.787 0.903 0.000
## TP-SRP    0.341  0.654 0.837 0.001
## TP-TIN    0.441  0.717 0.869 0.000
## TP-CHLA  -0.417 -0.017 0.389 0.938
## TP-ZP     0.409  0.697 0.859 0.000
## TN-SRP    0.557  0.784 0.902 0.000
## TN-TIN    0.929  0.969 0.987 0.000
## TN-CHLA  -0.407 -0.004 0.400 0.983
## TN-ZP     0.508  0.756 0.889 0.000
## SRP-TIN   0.587  0.801 0.910 0.000
## SRP-CHLA -0.551 -0.189 0.232 0.376
## SRP-ZP    0.375  0.676 0.848 0.000
## TIN-CHLA -0.527 -0.157 0.263 0.464
## TIN-ZP    0.515  0.761 0.891 0.000
## CHLA-ZP  -0.546 -0.183 0.238 0.393
```

```
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
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##           lower      r upper     p
## TP-TN     0.769  0.895 0.954 0.000
## TP-SRP    0.173  0.539 0.774 0.007
## TP-TIN    0.515  0.761 0.891 0.000
## TP-CHLA  -0.369  0.040 0.436 0.853
## TP-ZP     0.481  0.741 0.881 0.000
## TN-SRP    0.330  0.647 0.833 0.001
## TN-TIN    0.870  0.942 0.975 0.000
```

```
## TN-CHLA  -0.386  0.021 0.421 0.923
## TN-ZP     0.493  0.748 0.884 0.000
## SRP-TIN   0.457  0.726 0.874 0.000
## SRP-CHLA -0.456 -0.064 0.348 0.767
## SRP-ZP    0.299  0.627 0.822 0.001
## TIN-CHLA -0.327  0.088 0.474 0.683
## TIN-ZP    0.476  0.738 0.879 0.000
## CHLA-ZP  -0.462 -0.072 0.341 0.737
```

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 corr.test reports the probability values and confidence intervals in addition to the correlation matrix. The Corr.test shows that the negative correlations between CHLA and the other variables are not statistically significant ($P > 0.05$). Values in the correlation matrix and probability values changed slightly when a non-parametric method was used. However, correlations between CHLA and the other variables remained non-significant. Non-parametric tests should be used when when the data doesn't meet the assumptions of the parametric test, most often the assumption about normally distributed data. Multiple tests are adjusted for with Pearson's method. i.e. there is no evidence for false discovery rate due to multiple comparisons. Controlling the false discovery rate reduces the number of incorrect rejections of the null hypothesis (type 1 error)

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

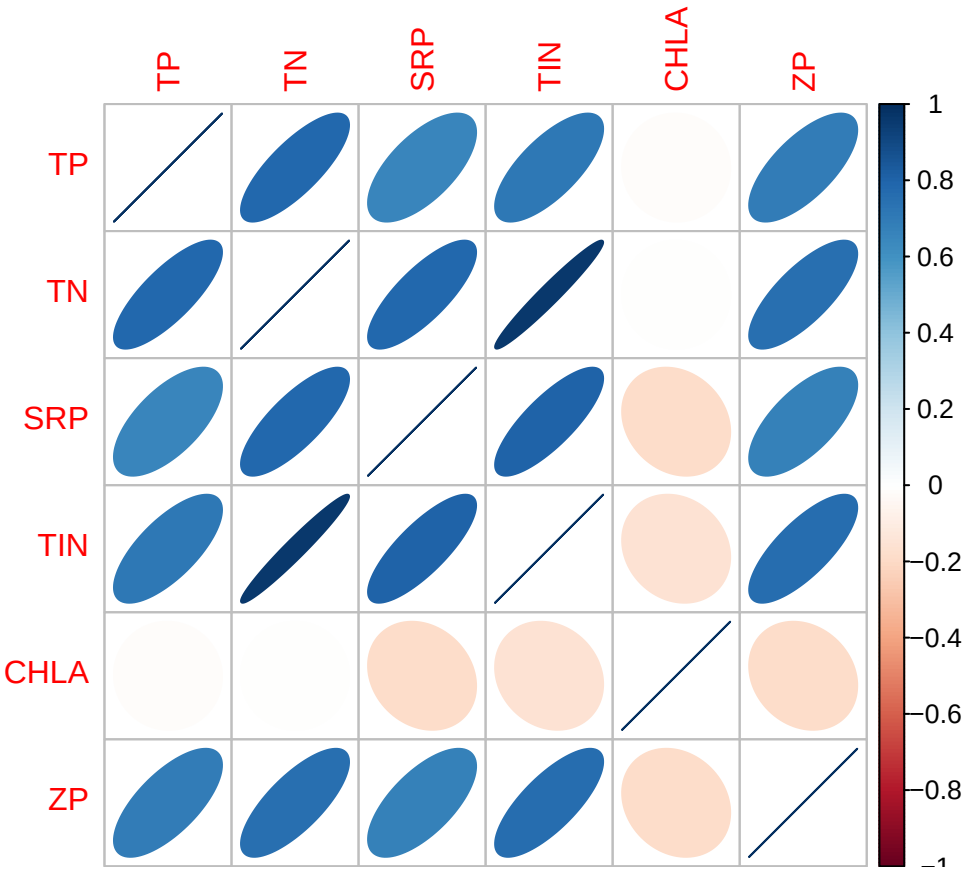
```
install.packages("corrplot", repos = "http://cran.rstudio.com/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/65/897bx8690x5_qclp18zt0sf80000gn/T//RtmpHMq1NS/downloaded_packages
```

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

```
## Loading required package: corrplot
```

```
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.

```
# Linear Regression analysis to test the relationship between TN and ZP
fitreg = lm(ZP ~TN, data = meso)
summary(fitreg)
```

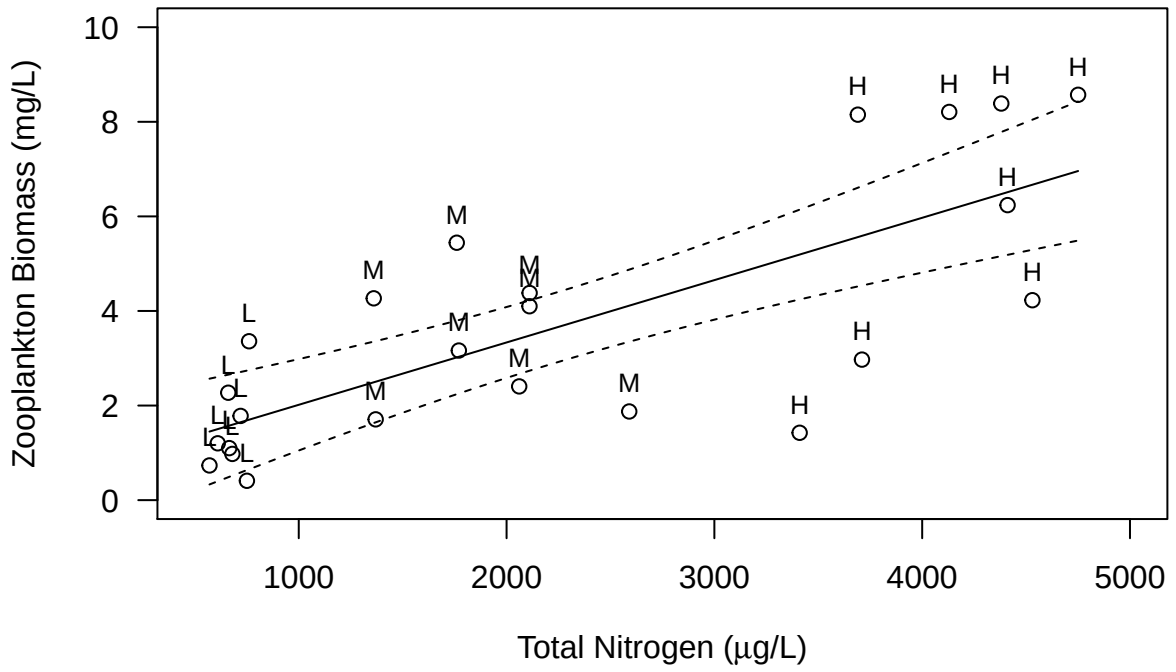
```
##
## Call:
## lm(formula = ZP ~ TN, data = meso)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7690 -0.8491 -0.0709  1.6238  2.5888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6977712  0.6496312   1.074    0.294
## TN          0.0013181  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 of Regression analysis
plot(meso$TN, meso$ZP, ylim=c(0,10), xlim = c(500, 5000),
     xlab = expression(paste("Total Nitrogen (", mu,"g/L)")),
     ylab = "Zooplankton Biomass (mg/L)", las = 1)
text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)

# Identifying a range of x values and generating the corresponding predicted y values from our regressi
newTN = seq(min(meso$TN), max(meso$TN), 10)
regline = predict(fitreg, newdata = data.frame(TN = newTN))
lines(newTN, regline)
# the line above calls the previous figure object

# Creating and plotting the 95% confidence intervals using newTN to generate corresponding confidence i
conf95 =  predict(fitreg, newdata = data.frame(TN = newTN),
                  interval = c("confidence"), level = 0.95, type = "response")
matlines(newTN, conf95[, c("lwr", "upr")], type = "l", lty = 2, lwd = 1, col = "black")
```



Question 6: Interpret the results from the regression model

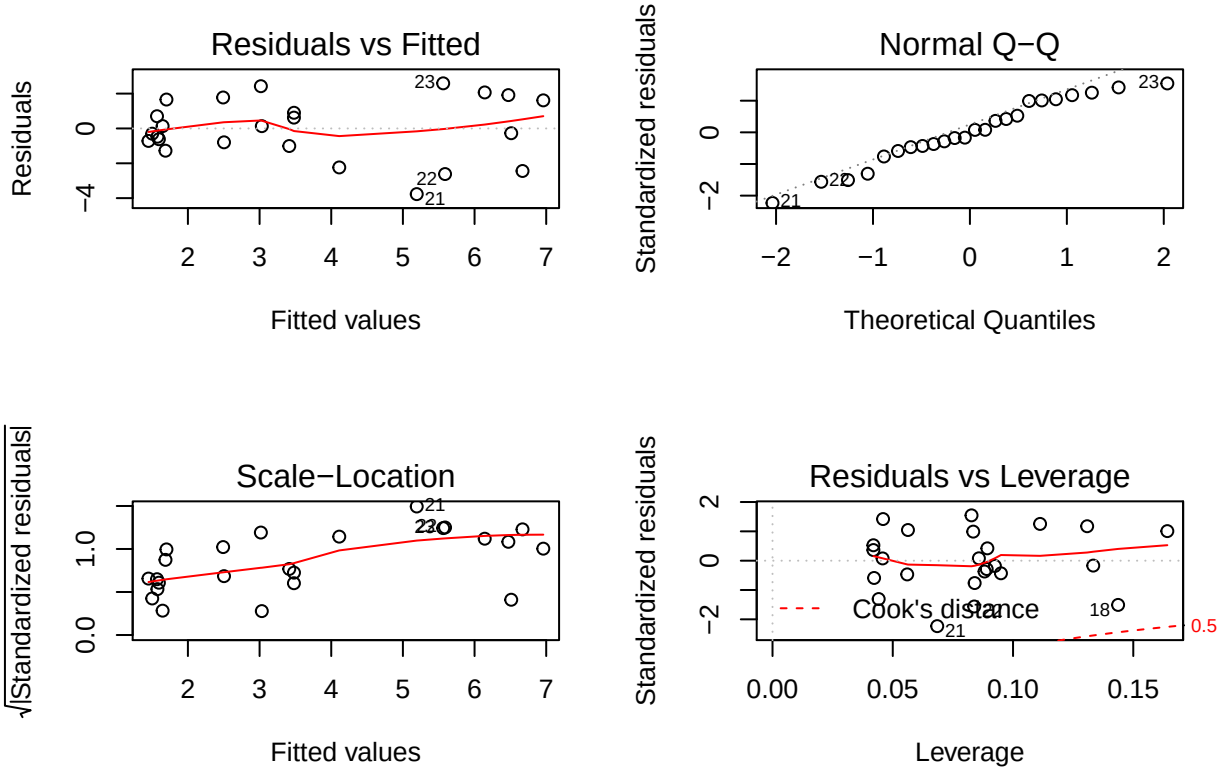
> ***Answer 6***: Zooplankton biomass appears to increase with increasing total N concentrations. The positve linear relationship between TN and ZP is statistically significant ($P < 0.01$; R^2 = 0.55). Total [N] explains ~55% of the variablity in zooplankton biomass.

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

> ***Answer 7***: The predict function is generating the predicted y values for a range of x values, from the regression model 'fitreg'

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(fitreg)
```



- Upper left: is there a random distribution of the residuals around zero (horizontal line)?
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)`
- Bottom left: again, looking for a random distribution of sqrt(standardized residuals)
- Bottom right: leverage indicates the influence of points; contours correspond with Cook's distance, where values $> |1|$ are "suspicious"

Analysis of Variance (ANOVA)

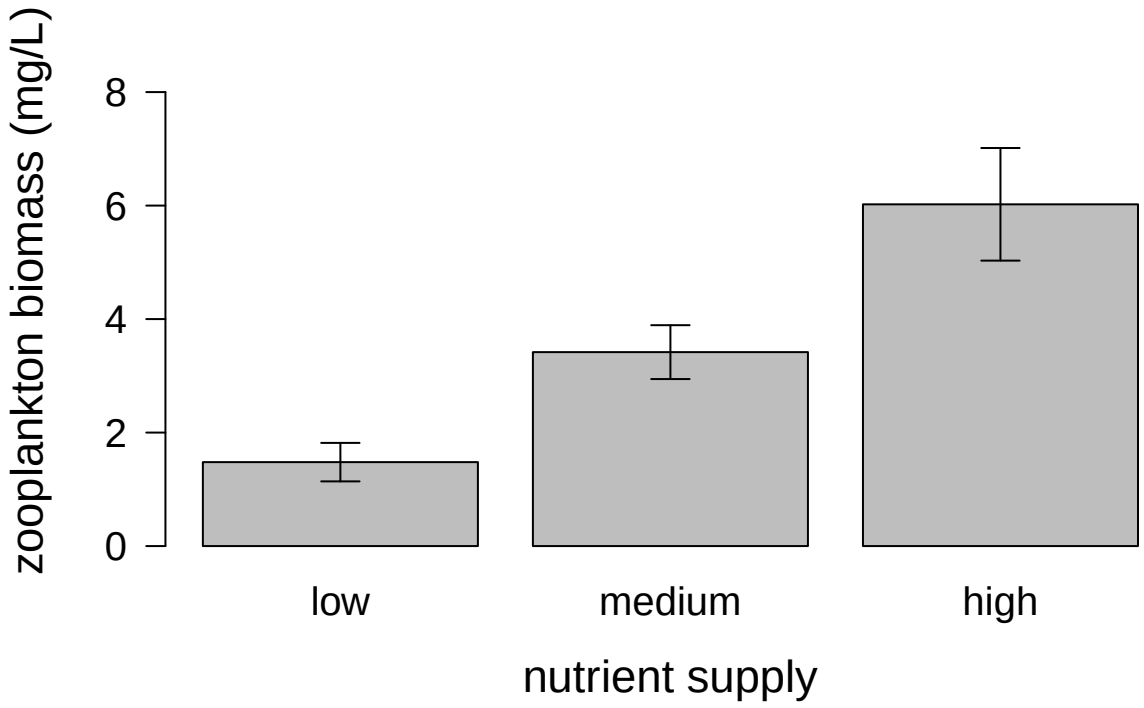
Using the R code chunk below, do the following: 1) Order the nutrient treatments from low to high (see handout). 2) Produce a barplot to visualize zooplankton biomass in each nutrient treatment. 3) Include error bars (+/- 1 sem) on your plot and label the axes appropriately. 4) Use a one-way analysis 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$NUTS, levels = c('L', 'M', 'H'))
# Calculating the means and errors for std errors for zooplankton biomass in the different nutrient tre
zp.means = tapply(meso$ZP, NUTS, mean)
zp.sem = tapply(meso$ZP, NUTS, sem)

# Barplot of ZP in each nutrient treatment
```

```r
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"))
# Adding error bars (sem)
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 to test H0: zooplankton biomass is affected by the nutrient treatment
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
```

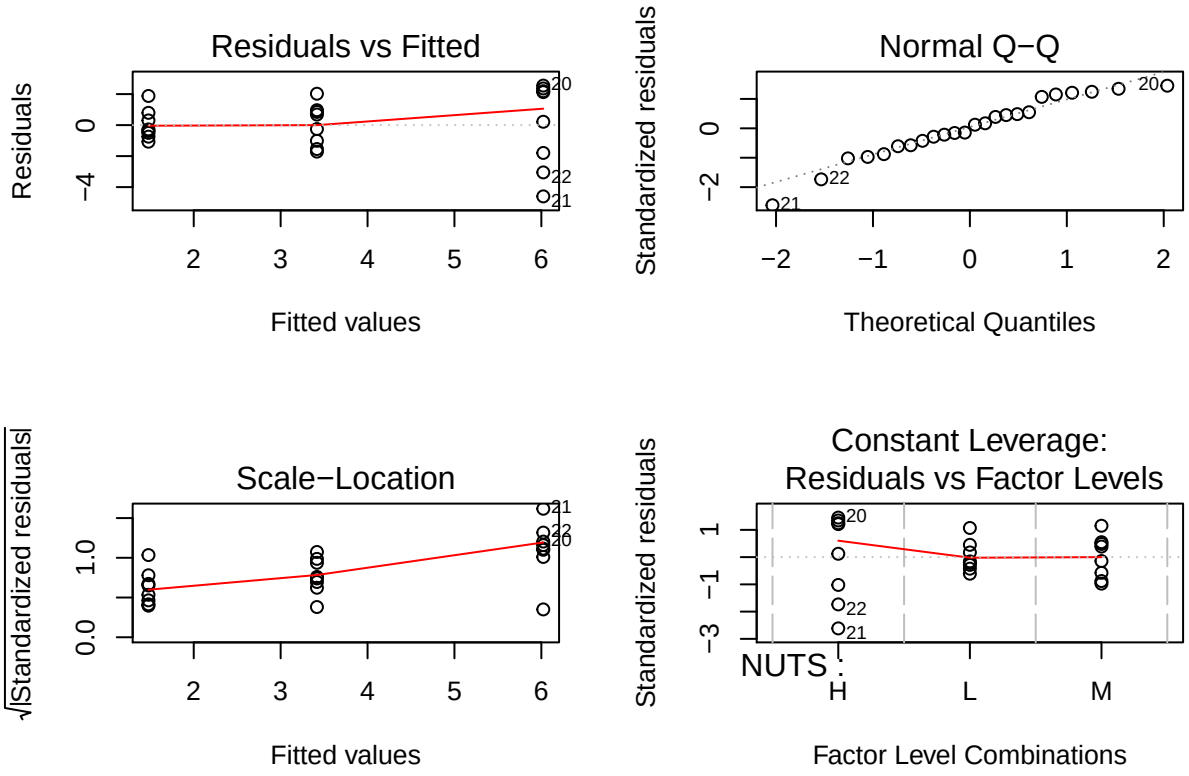
Question 8: How do you interpret the ANOVA results relative to the regression results?

Do you have any concerns about this analysis?

> ***Answer 8***: Zooplankton biomass varies with respect to nutrient treatment level ($P < 0.01$). So the extent to which total [N] predicts ZP defends on the nutrient treatment level. The low and high nutrent treatments differ significantly ($P < 0.01$), in additon to the medium and high treatments ($P < 0.05$). Low and medium leves were not significantly different from one another ($P = 0.12$). To eliminate potential concerns about the suitability of this analysis, we have to check to make sure the data meets the assumptions of an anova.

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).

```
# Checking on the residuals for fitanova
par(mfrow = c(2,2), mar = c(5.1, 4.1, 4.1, 2.1))
plot(fitanova)
```



SYNTHESIS: SITE-BY-SPECIES MATRIX

In the R code chunk below, load the 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 (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.

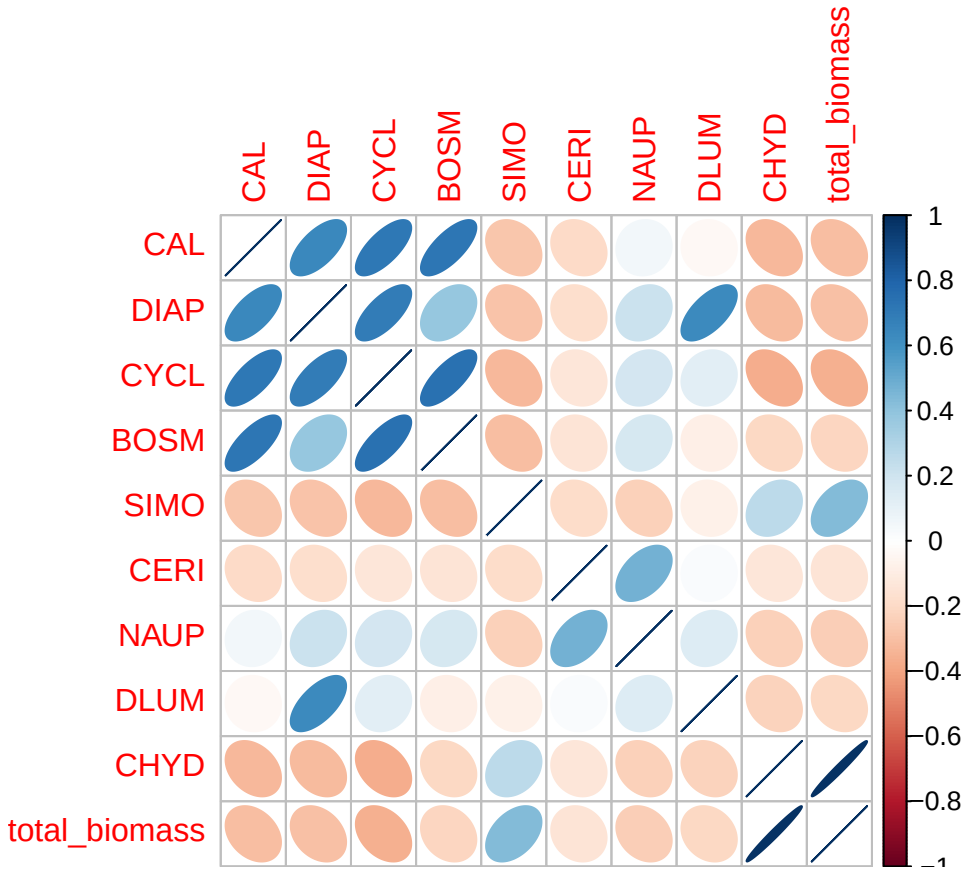
Question 9: With the visualization and statistical tools that we learned about in the Week 1 Handout, use the site-by-species matrix to assess whether and how different zooplankton taxa were responsible for the total biomass (ZP) response to nutrient enrichment. Describe what you learned below in the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

> ***Answer 9***: Total zooplankton biomass response to nutrient enrichment varied among the different zooplankton taxa. With *Chydorus* sp. significantly increasing the biomass response at the different sites ($P < 0.01$). *Chydorus* sp. biomass explained 96% of the variablity in total biomass response. *Simocephallus* sp. biomass was also a significant predictor of total biomass response, but to a lesser extent then *Chydorus* sp ($P = 0.04$). Together *Simocephallus* sp. and *Chydorus* sp biomass explained 99% of the variability in total biomass response.

```
zoop = read.table("./data/zoops.txt", header = TRUE)
# Biomass of different zooplankton taxa (sample by species matrix)
zoop.num = zoop[, 3:11]

# Calculating total zooplankton biomass (µg/L) for each sample
zoop.num$total_biomass = rowSums(zoop.num)

# Correlations among taxa with respect to total biomass response
cor4 = cor(zoop.num)
corrplot(cor4, method = "ellipse")
```




```
cor5 = corr.test(zoop.num, method = "pearson", adjust = "BH")
print(cor5, digits = 3, short = FALSE)
```

```
## Call:corr.test(x = zoop.num, method = "pearson", adjust = "BH")
## Correlation matrix
##                   CAL   DIAP   CYCL   BOSM   SIMO   CERI   NAUP   DLUM
## CAL             1.000  0.643  0.712  0.728 -0.271 -0.191  0.058 -0.034
## DIAP            0.643  1.000  0.694  0.381 -0.287 -0.172  0.217  0.637
## CYCL            0.712  0.694  1.000  0.747 -0.325 -0.132  0.186  0.125
## BOSM            0.728  0.381  0.747  1.000 -0.308 -0.141  0.179 -0.086
## SIMO           -0.271 -0.287 -0.325 -0.308  1.000 -0.183 -0.237 -0.077
## CERI           -0.191 -0.172 -0.132 -0.141 -0.183  1.000  0.475  0.020
## NAUP            0.058  0.217  0.186  0.179 -0.237  0.475  1.000  0.148
## DLUM           -0.034  0.637  0.125 -0.086 -0.077  0.020  0.148  1.000
## CHYD           -0.322 -0.314 -0.369 -0.206  0.262 -0.135 -0.238 -0.224
## total_biomass  -0.307 -0.299 -0.355 -0.214  0.431 -0.141 -0.244 -0.207
##                  CHYD total_biomass
## CAL            -0.322        -0.307
## DIAP           -0.314        -0.299
## CYCL           -0.369        -0.355
## BOSM           -0.206        -0.214
## SIMO            0.262         0.431
## CERI           -0.135        -0.141
## NAUP           -0.238        -0.244
## DLUM           -0.224        -0.207
## CHYD            1.000         0.981
```

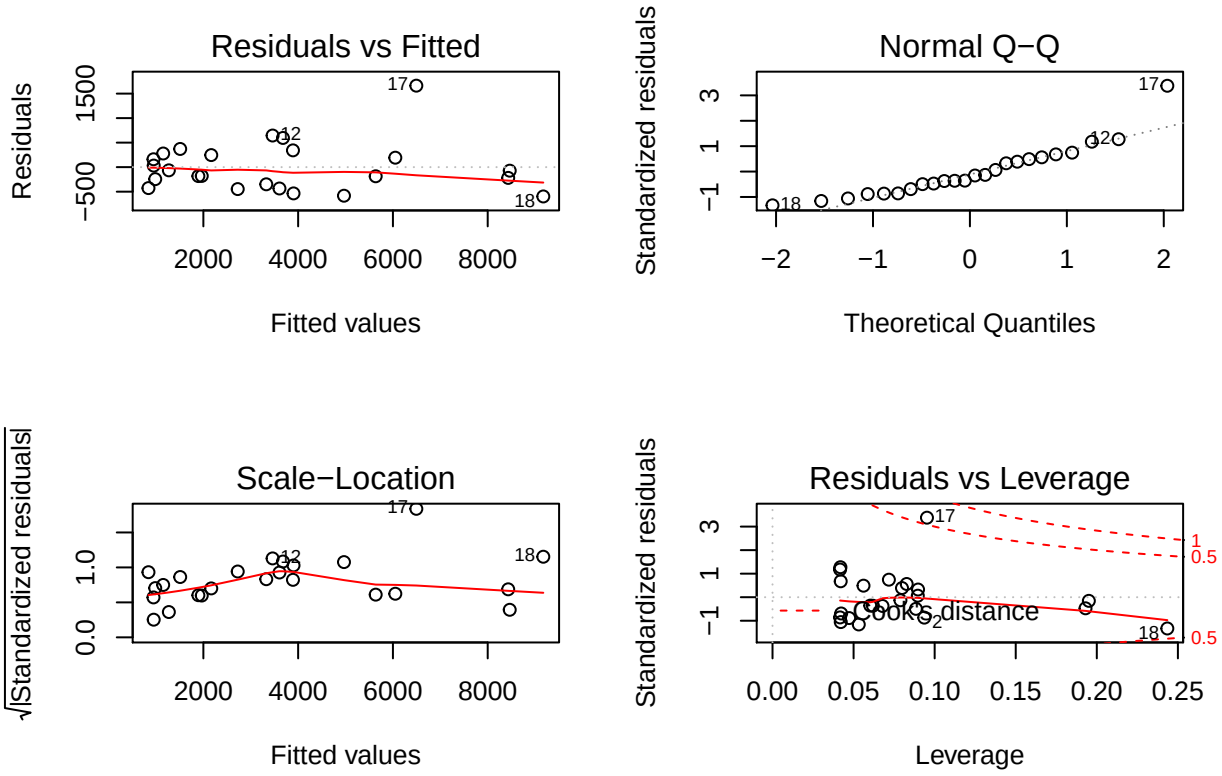
```
## total_biomass  0.981         1.000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##                  CAL  DIAP  CYCL  BOSM  SIMO  CERI  NAUP  DLUM  CHYD
## CAL           0.000 0.005 0.001 0.001 0.449 0.549 0.826 0.896 0.384
## DIAP          0.001 0.000 0.002 0.298 0.414 0.557 0.518 0.005 0.384
## CYCL          0.000 0.000 0.000 0.001 0.384 0.621 0.549 0.630 0.313
## BOSM          0.000 0.066 0.000 0.000 0.384 0.621 0.549 0.756 0.518
## SIMO          0.199 0.175 0.122 0.143 0.000 0.549 0.497 0.774 0.462
## CERI          0.371 0.421 0.538 0.510 0.393 0.000 0.108 0.925 0.621
## NAUP          0.789 0.309 0.385 0.403 0.265 0.019 0.000 0.621 0.497
## DLUM          0.876 0.001 0.560 0.688 0.722 0.925 0.491 0.000 0.518
## CHYD          0.125 0.136 0.076 0.334 0.216 0.528 0.263 0.293 0.000
## total_biomass 0.145 0.156 0.088 0.315 0.036 0.512 0.251 0.332 0.000
##               total_biomass
## CAL                   0.384
## DIAP                  0.391
## CYCL                  0.331
## BOSM                  0.518
## SIMO                  0.178
## CERI                  0.621
## NAUP                  0.497
## DLUM                  0.518
## CHYD                  0.000
## total_biomass         0.000
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##             lower      r upper     p
## CAL-DIAP    0.323  0.643 0.831 0.001
## CAL-CYCL    0.433  0.712 0.866 0.000
## CAL-BOSM    0.460  0.728 0.875 0.000
## CAL-SIMO   -0.608 -0.271 0.148 0.199
## CAL-CERI   -0.552 -0.191 0.230 0.371
## CAL-NAUP   -0.354  0.058 0.451 0.789
## CAL-DLUM   -0.431 -0.034 0.375 0.876
## CAL-CHYD   -0.642 -0.322 0.094 0.125
## CAL-ttl_b  -0.632 -0.307 0.110 0.145
## DIAP-CYCL   0.404  0.694 0.858 0.000
## DIAP-BOSM  -0.026  0.381 0.680 0.066
## DIAP-SIMO  -0.618 -0.287 0.132 0.175
## DIAP-CERI  -0.538 -0.172 0.248 0.421
## DIAP-NAUP  -0.205  0.217 0.570 0.309
## DIAP-DLUM   0.314  0.637 0.828 0.001
## DIAP-CHYD  -0.636 -0.314 0.103 0.136
## DIAP-ttl_b -0.627 -0.299 0.119 0.156
## CYCL-BOSM   0.491  0.747 0.884 0.000
## CYCL-SIMO  -0.644 -0.325 0.090 0.122
## CYCL-CERI  -0.508 -0.132 0.287 0.538
## CYCL-NAUP  -0.235  0.186 0.548 0.385
## CYCL-DLUM  -0.293  0.125 0.503 0.560
## CYCL-CHYD  -0.672 -0.369 0.041 0.076
```

```
## CYCL-ttl_b -0.664 -0.355 0.056 0.088
## BOSM-SIMO  -0.633 -0.308 0.109 0.143
## BOSM-CERI  -0.515 -0.141 0.278 0.510
## BOSM-NAUP  -0.242  0.179 0.543 0.403
## BOSM-DLUM  -0.473 -0.086 0.329 0.688
## BOSM-CHYD  -0.563 -0.206 0.215 0.334
## BOSM-ttl_b -0.569 -0.214 0.207 0.315
## SIMO-CERI  -0.546 -0.183 0.238 0.393
## SIMO-NAUP  -0.584 -0.237 0.184 0.265
## SIMO-DLUM  -0.466 -0.077 0.337 0.722
## SIMO-CHYD  -0.158  0.262 0.602 0.216
## SIMO-ttl_b  0.033  0.431 0.711 0.036
## CERI-NAUP   0.088  0.475 0.737 0.019
## CERI-DLUM  -0.386  0.020 0.420 0.925
## CERI-CHYD  -0.511 -0.135 0.283 0.528
## CERI-ttl_b -0.515 -0.141 0.278 0.512
## NAUP-DLUM  -0.272  0.148 0.520 0.491
## NAUP-CHYD  -0.585 -0.238 0.183 0.263
## NAUP-ttl_b -0.589 -0.244 0.177 0.251
## DLUM-CHYD  -0.575 -0.224 0.197 0.293
## DLUM-ttl_b -0.563 -0.207 0.215 0.332
## CHYD-ttl_b  0.956  0.981 0.992 0.000
```

```
# Total biomass appears to strongly positively correlated with CHYD biomass- testing to see if this rel
fitreg.CHYD = lm(total_biomass ~ CHYD, data = zoop.num)
summary(fitreg.CHYD)
```

```
##
## Call:
## lm(formula = total_biomass ~ CHYD, data = zoop.num)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -597.8 -369.3 -125.6  254.0 1661.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 679.4691   163.3773   4.159 0.000409 ***
## CHYD          1.0205     0.0429  23.789  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 517.2 on 22 degrees of freedom
## Multiple R-squared:  0.9626, Adjusted R-squared:  0.9609
## F-statistic: 565.9 on 1 and 22 DF,  p-value: < 2.2e-16
```

```
# Not sure if the data is normal
par(mfrow = c(2,2), mar = c(5.1, 4.1, 4.1, 2.1))
plot(fitreg.CHYD)
```



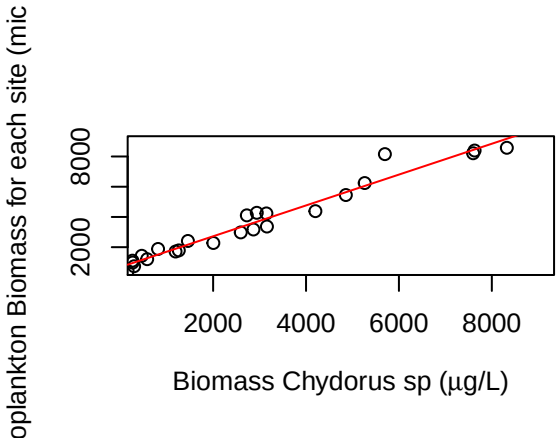

```
# Plot of Regression analysis
plot(zoop.num$CHYD, zoop.num$total_biomass, ylim=c(500, 9000), xlim = c(500, 9000),
     xlab = expression(paste("Biomass Chydorus sp (", mu,"g/L)")),
     ylab = "Total Zooplankton Biomass for each site (micrograms/L)")
newCHYD = seq(min(zoop.num$CHYD), max(zoop.num$CHYD), 10)
regline.CHYD = predict(fitreg.CHYD, newCHYD = data.frame(CHYD = newCHYD))
abline(fitreg.CHYD, col = "red")

# Regression including both SIMO and CHYD biomass
fitreg.taxa = lm(total_biomass ~  SIMO + CHYD, data = zoop.num)
summary(fitreg.taxa)
```

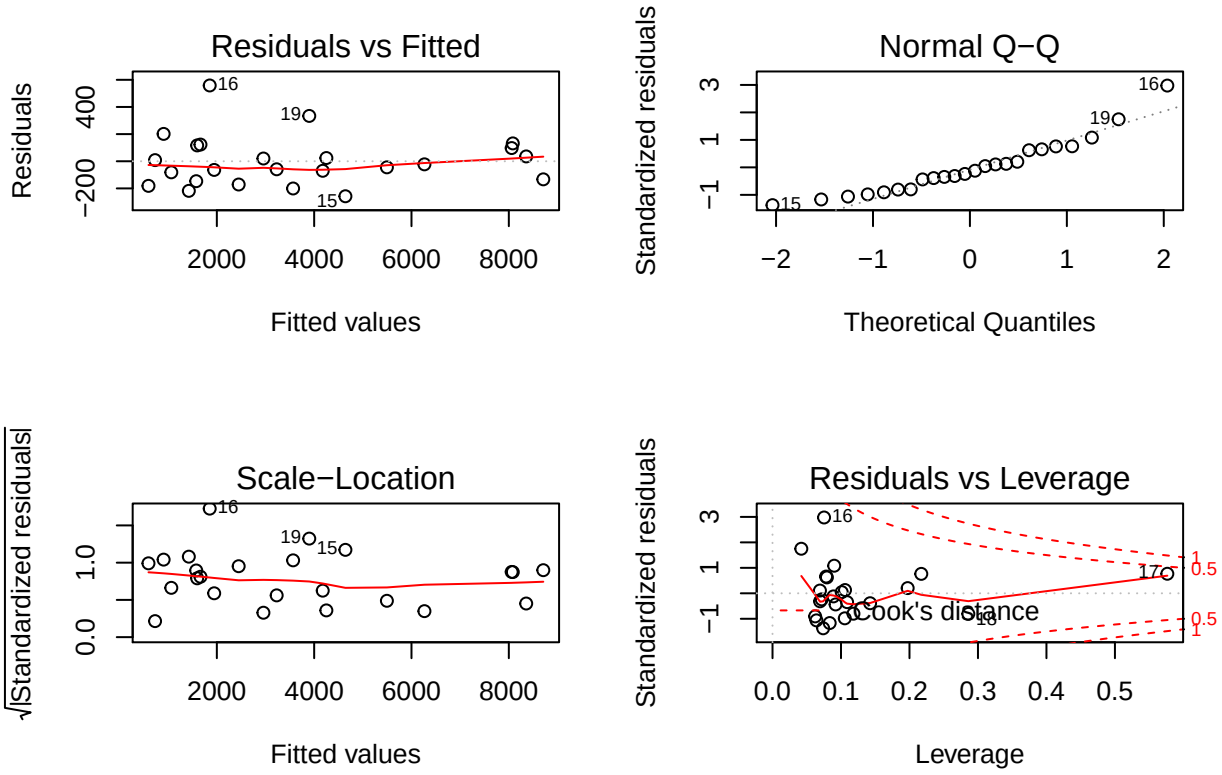
```
##
## Call:
## lm(formula = total_biomass ~ SIMO + CHYD, data = zoop.num)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -257.78 -136.54  -33.68  102.20  558.24
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 441.31874   64.96599   6.793 1.02e-06 ***
## SIMO          0.87264    0.07547  11.562 1.44e-10 ***
## CHYD          0.96962    0.01676  57.837  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.1 on 21 degrees of freedom
```

```
## Multiple R-squared:  0.9949, Adjusted R-squared:  0.9944
## F-statistic:  2056 on 2 and 21 DF,  p-value: < 2.2e-16
```

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




```
plot(fitreg.taxa)
```



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)**.