Week 1 Assignment: Basic R

Michelle Benavidez; Z620: Quantitative Biodiversity, Indiana University

17 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/Michelle/GitHub/QB2017_Benavidez/Week1"
```

```
setwd("C:/Users/Michelle/GitHub/QB2017_Benavidez/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.

```
cubevolume=5^3
cubevolume
```

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

```
circleradius=pi*2^2
circleradius
```

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

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

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

```
favelog=log(3)
favelog
```

```
## [1] 1.098612
```

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(1,2,3,4,5)
x
```

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

```
w=x*14
w
```

```
## [1] 14 28 42 56 70
```

```
number3=(x+w)/15
number3
```

```
## [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(2,4,6,8,10)
k
```

```
## [1]  2  4  6  8 10
```

```
number2=k*x
number2
```

```
## [1]  2  8 18 32 50
```

```
d=c(14,28,42,2,4,6,8)
d
```

```
## [1] 14 28 42  2  4  6  8
```

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(v,na.rm=T)
```

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

```
min(v,na.rm=T)
```

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

```
sum(v,na.rm=T)
```

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

```
mean(v,na.rm=T)
```

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

```
median(v,na.rm=T)
```

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

```
var(v,na.rm=T)
```

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

```
sd(v,na.rm=T)
```

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

```
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.

```
one=c(rnorm(5,mean=8,sd=2))
two=c(rnorm(5,mean=25,sd=10))
one
```

```
## [1] 10.989801  7.001467  8.778146 12.468262  8.814717
```

```
two
```

```
## [1] 14.97444 37.65469 37.77051 45.58992 24.72494
```

```
onetwo=cbind(one,two)
onetwomatrix=matrix(onetwo, nrow=5,ncol=2,byrow=F)
onetwomatrix
```

```
##           [,1]     [,2]
## [1,] 10.989801 14.97444
## [2,]  7.001467 37.65469
## [3,]  8.778146 37.77051
## [4,] 12.468262 45.58992
## [5,]  8.814717 24.72494
```

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:The command rnorm draws random numbers from a normal distribution based on user-defined parameters. Arguements: rnorm (number of numbers to draw, average of the numbers drawn, standard deviation of the numbers drawn)

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=as.matrix(read.table("data/matrix.txt",sep="\t",header=F))
m
```

```
##       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
```

```
transposem=t(m)
transposem
```

```
##    [,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
```

```r
dim(transposem)
```

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

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

***Answer 2: 5X10

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`.

```r
indexm=transposem[,c(1:2,4:10)]
indexm
```

```
##    [,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
```

```r
indexm2=indexm[c(1:4),c(1:9)]
indexm2
```

```
##    [,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
```

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

***Answer 3: In the last series of steps we excluded specified rows and columns by concatenating the row/columns we wanted to include in the matrices. In the first step, columns 1 through 4 were concatenated with columns 4 through 10; thereby excluding column three. The comma included before the column concatenation indicates that all rows should be included. For step two, the modified matrix was used and rows 1 through 4 were concatenated and grouped with columns 1 through 9; thereby excluding row 5 (i.e. the last row).

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.

```r
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 ...
```

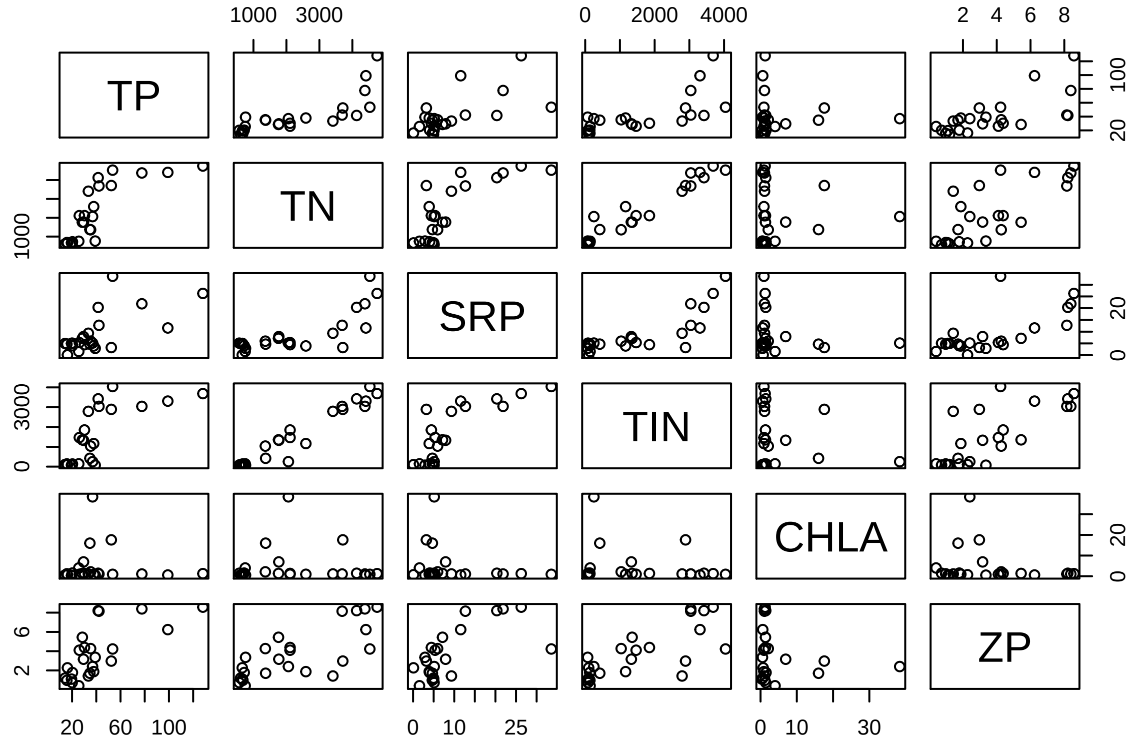
Correlation

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
```

```
pairs(meso.num)
```



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

> ***Answer 4: TN/TIN has a very strong positive correlation. TN/SRP, TP/TN, ZP/TN, TP/TRN, TIN/ZP, and TIN/SRP have strong to moderately strong positive correlations. ZP/TP, TP/SRP, ZP/SRP have moderate positive correlations. CHLA lacks noticable correlation with any other factor. Overall, total nitrogen and total inorganic nutrient concentration have the strongest correlation with total biomass.

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", lib.loc="~/R/win-library/3.1")
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
```

```
cor2non=corr.test(meso.num,method="spearman",adjust="BH")
cor2non
```

```
## Call:corr.test(x = meso.num, method = "spearman", adjust = "BH")
## Correlation matrix
##        TP   TN   SRP  TIN  CHLA    ZP
## TP   1.00 0.89  0.54 0.76  0.04  0.74
## TN   0.89 1.00  0.65 0.94  0.02  0.75
## SRP  0.54 0.65  1.00 0.73 -0.06  0.63
## TIN  0.76 0.94  0.73 1.00  0.09  0.74
## CHLA 0.04 0.02 -0.06 0.09  1.00 -0.07
## ZP   0.74 0.75  0.63 0.74 -0.07  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 0.01 0.00 0.91 0.00
## TN   0.00 0.00 0.00 0.00 0.92 0.00
## SRP  0.01 0.00 0.00 0.00 0.88 0.00
## TIN  0.00 0.00 0.00 0.00 0.88 0.00
## CHLA 0.85 0.92 0.77 0.68 0.00 0.88
## ZP   0.00 0.00 0.00 0.00 0.74 0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
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(cor2non,digits = 3)
```

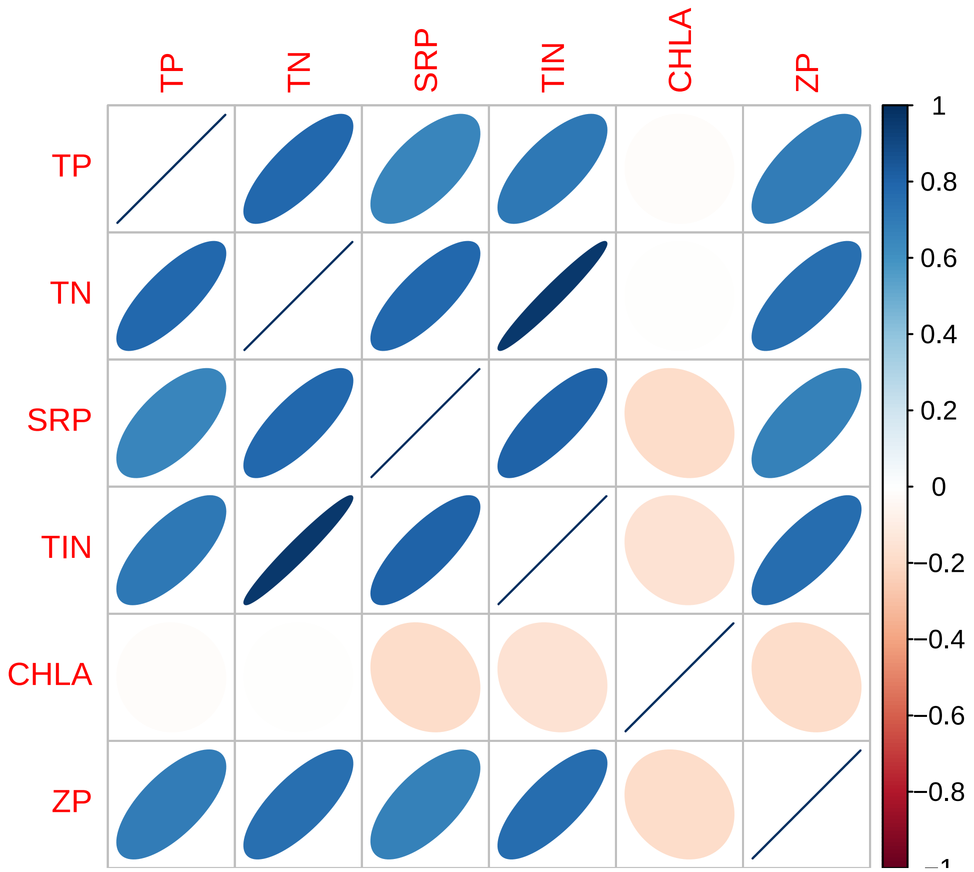
```
## Call:corr.test(x = meso.num, method = "spearman", adjust = "BH")
## Correlation matrix
##         TP    TN    SRP   TIN   CHLA     ZP
## TP   1.000 0.895  0.539 0.761  0.040  0.741
## TN   0.895 1.000  0.647 0.942  0.021  0.748
## SRP  0.539 0.647  1.000 0.726 -0.064  0.627
## TIN  0.761 0.942  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
```

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: In the case of this dataset, the results were somewhat sensitive to parametric vs nonparametric tests. One should use non-parametric tests when the data does not meet the assumptions of the test (i.e. normality of data distrubtion, independence of variables, homoscedasticity). Using the Pearson method, there is minimal evidence that suggests a false discovery rate due to mulitple comparisons. Assessing false discovery rates is important for determining how susceptible your data/test are to Type I and II errors.

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

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



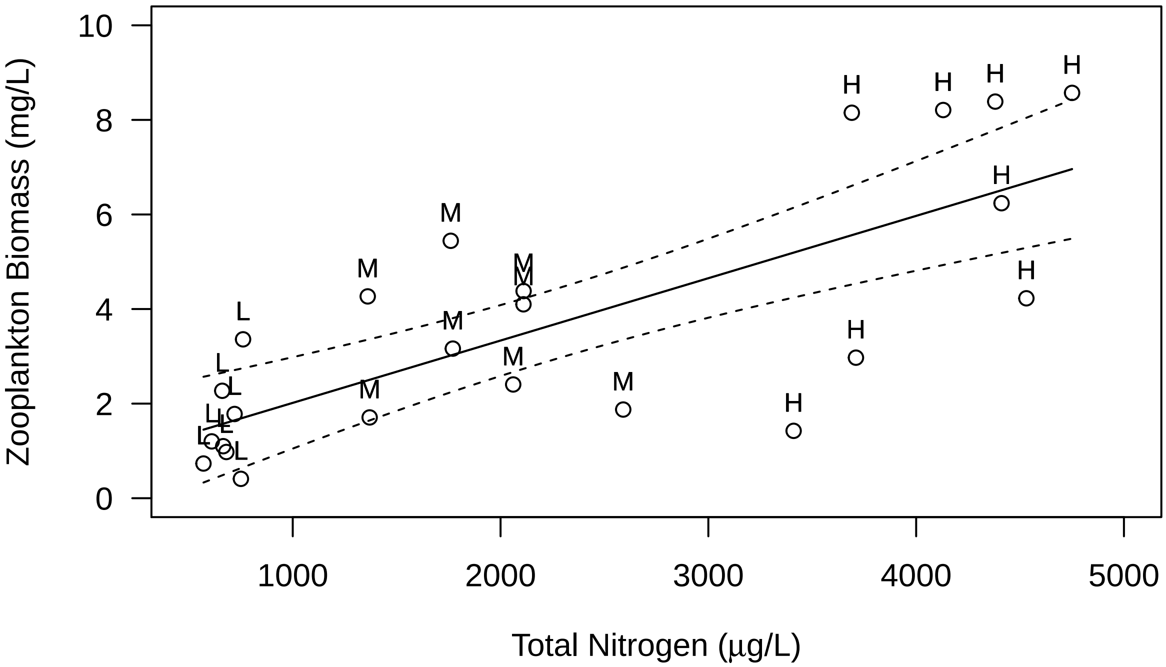
Linear Regression

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.

```
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(meso$TN, meso$ZP, ylim=c(0,10),xlim=c(500,5000),xlab=expression(paste("Total Nitrogen (",mu,"g/L)"
text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)
text(meso$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)
text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)
newTN=seq(min(meso$TN), max(meso$TN), 10)
regline=predict(fitreg, newdata = data.frame(TN = newTN))
lines(newTN, regline)
conf95=predict(fitreg, newdata = data.frame(TN = newTN),interval = c("confidence"), level = 0.95, type
matlines(newTN, conf95[, c("lwr", "upr")], type="l", lty = 2, lwd = 1, col = "black")
```



Question 6: Interpret the results from the regression model

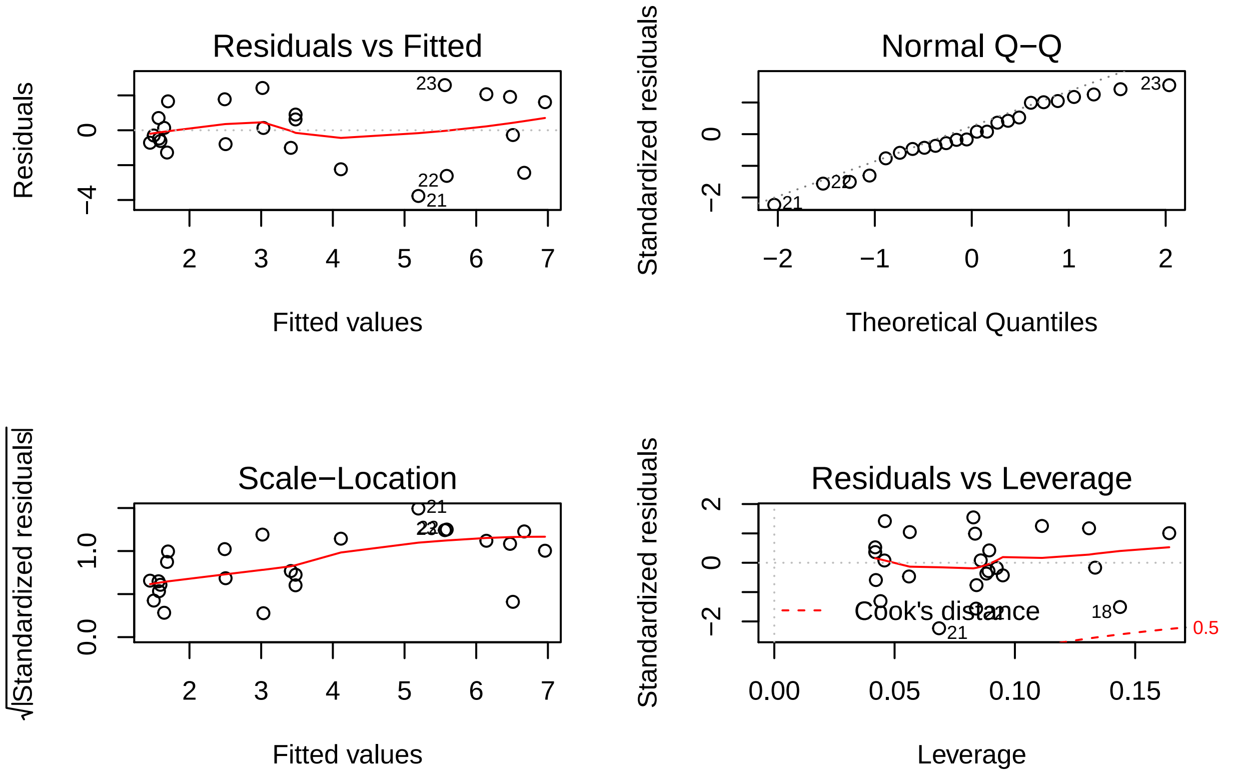
***Answer 6: TN has a significant effect on the reponse variable (F = 20.29, df=1/22, p < .001). For each unit of change in TN the reponse variable changes by a factor of 0.001.

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

***Answer 7: The predict function is generating the regression line, which in a sense generates an expected trajectory line that provides a visualiztion of the direction and strength of the effect of predictor variables on the reponse variable based on the specified model parameters.

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)
```

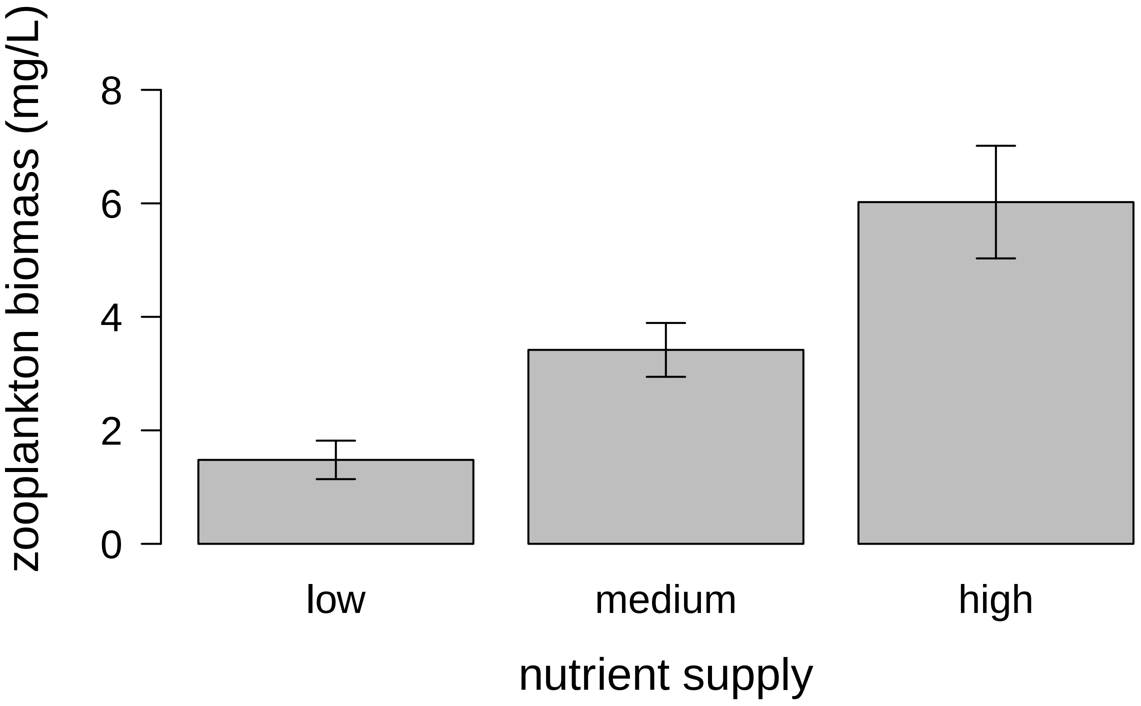


- Upper left: is there a random distribution of the residuals around zero (horizontal line)? > ***There does appear to be a randon distribution because no specific patterns or clustering are present in relation to zero.
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)` > ***There is a reasonably linear relation between standardized residuals and the theoretical quantiles. The devition of the perfect linear relationsip are miniscule.
- Bottom left: again, looking for a random distribution of sqrt(standardized residuals) > ***Again, the distribution appears random because there is not a distinct pattern in the distribution of data points.
- Bottom right: leverage indicates the influence of points; contours correspond with Cook's distance, where values > |1| are "suspicious" > ***There does not appear to be influential points when comparing residuals and leverage. All points appear to be within Cook's line.

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$NUTS, levels = c('L', 'M', 'H'))
zp.sem <- tapply(meso$ZP, NUTS, sem)
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
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)
```



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

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

```
TukeyHSD(fitanova)
```

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

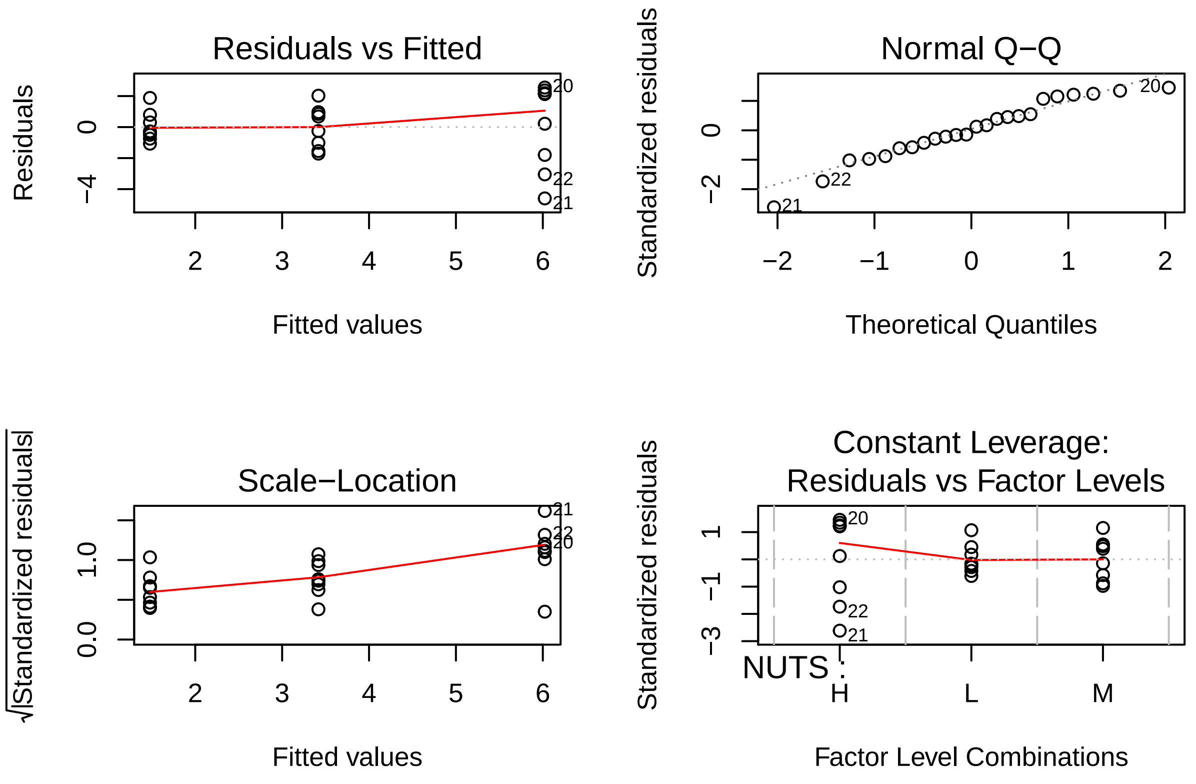
Question 8: How do you interpret the ANOVA results relative to the regression results? Do you have any concerns about this analysis?

> ***Answer 8: There appear to be significant differences between at least one pair of treatment levels (F=11.77, df=2/22, p < 0.001). The post hoc test identified a significant difference between the Low and High groups and the Medium and High groups. Since CI overlap zero for the Medium and Low group, I conclude that they are not statistically different. At this point, assumptions have not been checked, but otherwise I have no particualr concerns about the analysis.

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).

> Data may be heteroskedastic because there appears to be a football-shaped pattern among group variances; therefore the data does not meet the assumptions.

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



SYNTHESIS: SITE-BY-SPECIES MATRIX

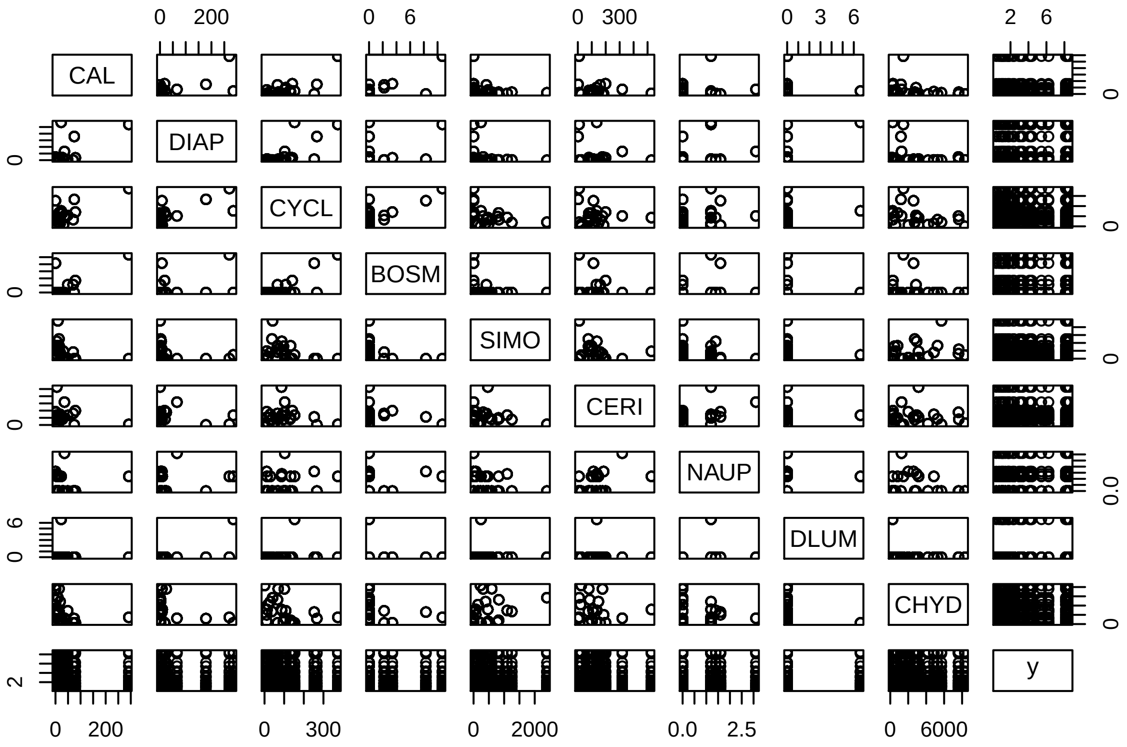
In the R code chunk below, load the zoop.txt dataset in your Week1 data folder. Create a site-by-species matrix (or dataframe) that does not include TANK or NUTS. The remaining columns of data refer to the biomass (Âµg/L) of different zooplankton taxa:

- 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.

> ***Answer 9: Zooplankton species are not a good indicator of biomass change in reference to total biomass reponse to nutrient enrichment.

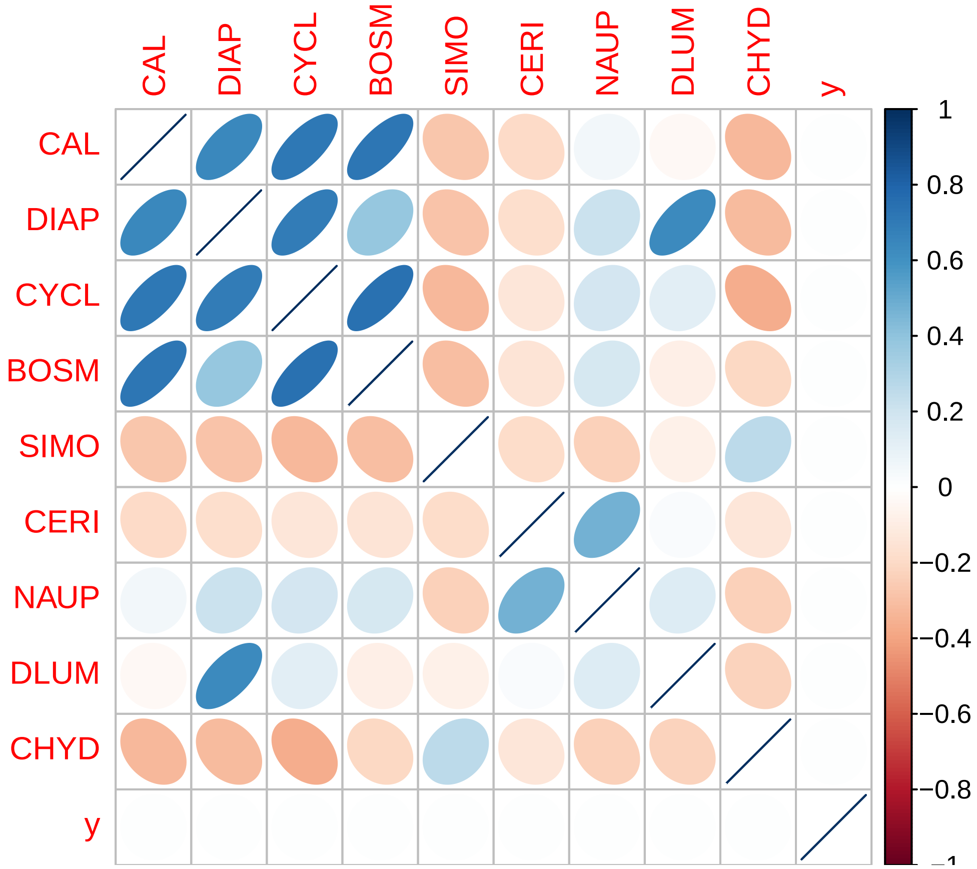
```
zoop <-read.table("data/zoops.txt", sep = "\t", header = TRUE)
data=merge(zoop,meso$ZP)
data.num=data[,3:12]
pairs(data.num)
```



```r
zoocor1=cor(data.num)
zoocor2=corr.test(data.num,method="pearson",adjust="BH")
print(zoocor2,digits=3)
```

```
## Call:corr.test(x = data.num, method = "pearson", adjust = "BH")
## Correlation matrix
##         CAL   DIAP   CYCL   BOSM   SIMO   CERI   NAUP   DLUM   CHYD y
## CAL   1.000  0.643  0.712  0.728 -0.271 -0.191  0.058 -0.034 -0.322 0
## DIAP  0.643  1.000  0.694  0.381 -0.287 -0.172  0.217  0.637 -0.314 0
## CYCL  0.712  0.694  1.000  0.747 -0.325 -0.132  0.186  0.125 -0.369 0
## BOSM  0.728  0.381  0.747  1.000 -0.308 -0.141  0.179 -0.086 -0.206 0
## SIMO -0.271 -0.287 -0.325 -0.308  1.000 -0.183 -0.237 -0.077  0.262 0
## CERI -0.191 -0.172 -0.132 -0.141 -0.183  1.000  0.475  0.020 -0.135 0
## NAUP  0.058  0.217  0.186  0.179 -0.237  0.475  1.000  0.148 -0.238 0
## DLUM -0.034  0.637  0.125 -0.086 -0.077  0.020  0.148  1.000 -0.224 0
## CHYD -0.322 -0.314 -0.369 -0.206  0.262 -0.135 -0.238 -0.224  1.000 0
## y     0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 1
## Sample Size
## [1] 576
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##        CAL DIAP  CYCL  BOSM  SIMO  CERI  NAUP  DLUM  CHYD y
## CAL  0.000    0 0.000 0.000 0.000 0.000 0.221 0.542 0.000 1
## DIAP 0.000    0 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1
## CYCL 0.000    0 0.000 0.000 0.000 0.002 0.000 0.004 0.000 1
## BOSM 0.000    0 0.000 0.000 0.000 0.001 0.000 0.054 0.000 1
## SIMO 0.000    0 0.000 0.000 0.000 0.000 0.000 0.090 0.000 1
## CERI 0.000    0 0.001 0.001 0.000 0.000 0.000 0.785 0.002 1
## NAUP 0.167    0 0.000 0.000 0.000 0.000 0.000 0.001 0.000 1
## DLUM 0.422    0 0.003 0.038 0.066 0.628 0.000 0.000 0.000 1
## CHYD 0.000    0 0.000 0.000 0.000 0.001 0.000 0.000 0.000 1
## y    1.000    1 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```r
corrplot::corrplot(zoocor1,method="ellipse")
```



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 18$^{\text{th}}$, 2015 at 12:00 PM (noon)**.