Week 1 Assignment: Basic R

Ashwini Ramesh; Z620: Quantitative Biodiversity, Indiana University

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/DELL/GitHub/QB2017_Ramesh/Week1"
```

```
setwd("C:/Users/DELL/GitHub/QB2017_Ramesh/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.

```
#1 The volume of a cube
l = 5
volofcube = l*l*l
volofcube
```

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

```
#2 The area of a circle with radius 2
r = 2
areaofcircle = pi*r*r
areaofcircle
```

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

```
#3 the length of the opposite side of a right-triangle given that the angle, theta, = pi/4.

hyp = sqrt(2)
theta = pi/4
len = sin(theta)*hyp
len
```

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

```
#4 log base(e) of your fav number
lnoffavnumber = log(1.618) #Golden Ratio!
```

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.

```r
#1) Create a vector `x` consisting of any five numbers.
x <- sample(1:100, 5, replace=TRUE)
x
```

```
## [1] 49 88 42 18 64
```

```r
#2) Create a new vector `w` by multiplying `x` by 14 (i.e., "scalar").
w <- x * 14
w
```

```
## [1]  686 1232  588  252  896
```

```r
#3) Add `x` and `w` and divide by 15.
y <- (x + w)/15
y
```

```
## [1] 49 88 42 18 64
```

Now, do the following: 1) Create another vector (`k`) that is the same length as `w`. 2) Multiply `k` by `x`. 3) Use the combine function to create one more vector, `d` that consists of any three elements from `w` and any four elements of `k`.

```r
#1) Create another vector (`k`) that is the same length as `w`.
k <- sample(1:100, 5, replace=TRUE)
k
```

```
## [1] 65 81 28 62  1
```

```r
#2) Multiply `k` by `x`.
l <- k * x
l
```

```
## [1] 3185 7128 1176 1116   64
```

```r
#3) Use the combine function to create one more vector, `d` that consists of any three elements from `w
d1 <- sample(w,3,replace = FALSE)
d2 <- sample(k,4,replace = FALSE)
d <- c(d1,d2)
d
```

```
## [1]  686 1232  588    1   62   28   65
```

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.

```r
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
```

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

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

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

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

```r
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
a <- c(rnorm(5,mean = 8, sd = 2)) #Approach 1
b <- c(rnorm(5,mean = 25, sd=10))# Approach 1
c <- cbind(a,b)#Approach 1
c
```

```
##             a         b
## [1,] 5.375399 -2.452913
## [2,] 9.364081  7.368015
## [3,] 8.035971 33.237561
## [4,] 7.951436 40.589069
## [5,] 6.656332 26.615692
```

```
l <- matrix(c(a,b), nrow = 5, ncol = 2) # Approach 2
l
```

```
##          [,1]      [,2]
## [1,] 5.375399 -2.452913
## [2,] 9.364081  7.368015
## [3,] 8.035971 33.237561
## [4,] 7.951436 40.589069
## [5,] 6.656332 26.615692
```

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 function is used to specify the parameters drawn from a normal distribution. The arguments in the function denote sample size (n), the mean of the distribution and the standard deviation (sd). By default, the normal distribution of this function, would be mean 0 and stranded deviation 1.

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.

```
#1) Load `matrix.txt` from the Week1 data folder as matrix `m`.
m <- as.matrix(read.table("data/matrix.txt", sep = "\t", header = FALSE))

#2) Transpose this matrix.
n <- t(m)

#3) Determine the dimensions of the transposed matrix.
dim(n)
```

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

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

> Answer 2:(5, 10) 5 columns and 10 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
```

```
##       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
```

```
#1) Index matrix `m` by selecting all but the third column.
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
```

```
#2) Remove the last row of matrix `m`.
m[-10,]
```

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

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

> ***Answer 3***: The matrix is represented as "matrix[rows, columns]". By adding a - sign preceding the index of the row/column, the specified vector is discarded from the matrix

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.

```
setwd("C:/Users/DELL/GitHub/QB2017_Ramesh/Week1/data")
zoo <- read.table("zoop_nuts.txt", sep = "\t", header = TRUE)
str(zoo)
```

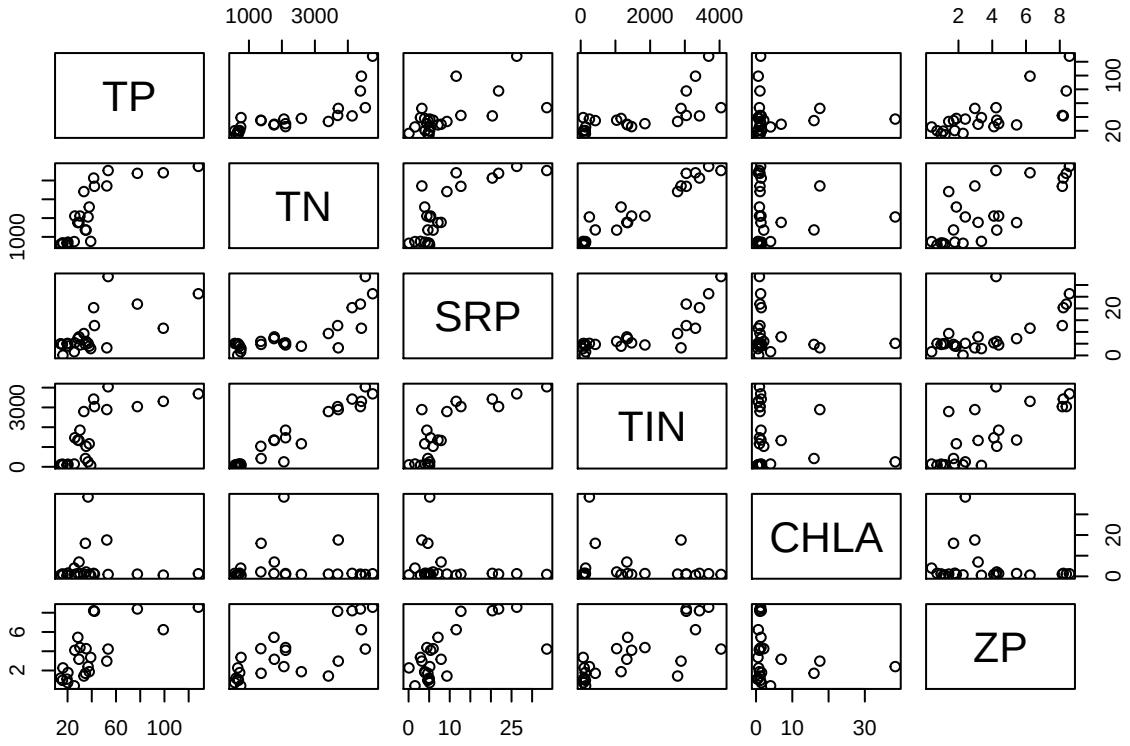
```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: Factor w/ 3 levels "H","L","M": 2 2 2 2 2 2 2 2 3 3 ...
##  $ TP  : num  20.3 25.6 14.2 39.1 20.1 ...
##  $ TN  : num  720 750 610 761 570 ...
##  $ SRP : num  4.02 1.56 4.97 2.89 5.11 4.68 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.

```
#1) Create a matrix with the numerical data in the `meso` dataframe.
meso <- zoo[,3:8]

#2) Visualize the pairwise **bi-plots** of the six numerical variables.
pairs(meso)
```



```
#) Conduct a simple **Pearson's correlation** analysis.
cor1 <- cor(meso)
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: While the scatter plots help us understand the general trend in the relationship between two variables and visually seek out the outliers in the plot or any unusual clumping that may occur in the data, the correlation denotes the strength of relationship between the two variables. For example, total nitrogen seems to be fairly well correlated (>70%) to all other variables except CHLA. CHLA does not appear to be a good predictor (linear) of any of the other variables as the correlation function is close to 0. This is also supported by the scatter-plot indicating a 0 slope value. In fact, it would be interesting to go back and look at the outliers in these plots. This could help one do a 'data-recheck' or in some cases understand the anomalous behaviour of certain data points.

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.

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

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

```
## Warning: package 'psych' was built under R version 3.1.3
```

```
#1) Redo the correlation analysis using the `corr.test()` function in the `psych` package with the foll
cor2 <- corr.test(meso, method = "pearson", adjust = "BH") #Pearson's correlation

#2) Now, redo this correlation analysis using a non-parametric method.

cor3 <- corr.test(meso, method = "kendall", adjust = "BH") #Kendall's correlation

#3) Use the print command from the handout to see the results of each correlation analysis
cor2
```

```
## Call:corr.test(x = meso, 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
```

```
## Call:corr.test(x = meso, 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`. 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***: #Describe what you learned from `corr.test`. Corr.test reports the probability value associated with each of the corresponding values in the correlation matrix i.e. what is the

probability that the observed values of correlation will actually be observed? # #Specifically, are the results sensitive to whether you use parametric (i.e., Pearson's) or non-parametric methods? Yes, the results are sensitive to if parametric(Pearson) or non-parametric(Kendall/Spearman) are used. These are easily observed from the results in the above data. # #When should one use non-parametric methods instead of parametric methods? Non-parametric methods are primarily used when the underlying distribution is unknown, unlike parametric distributions that assume normal distribution of data. Other key reasons to use non-parametric methods is if sample size is low, if the measure of central tendency is better represented by a median than mean(parametric), or if you have ranked data/ outliers that cannot be removed from the dataset. # #With the Pearson's method, is there evidence for false discovery rate due to multiple comparisons? Why is false discovery rate important? With Pearson's method there is evidence for false discovery rate due to multiple comparisons, as it assumes normal distribution as the underlying distribution for the data. If the data is not normally distributed, then there are chances that one incorrectly rejects the true null. False Discovery Rate is especially important as it increases the chances of Type I Error (i.e. incorrectly rejecting a true null hypotheses)

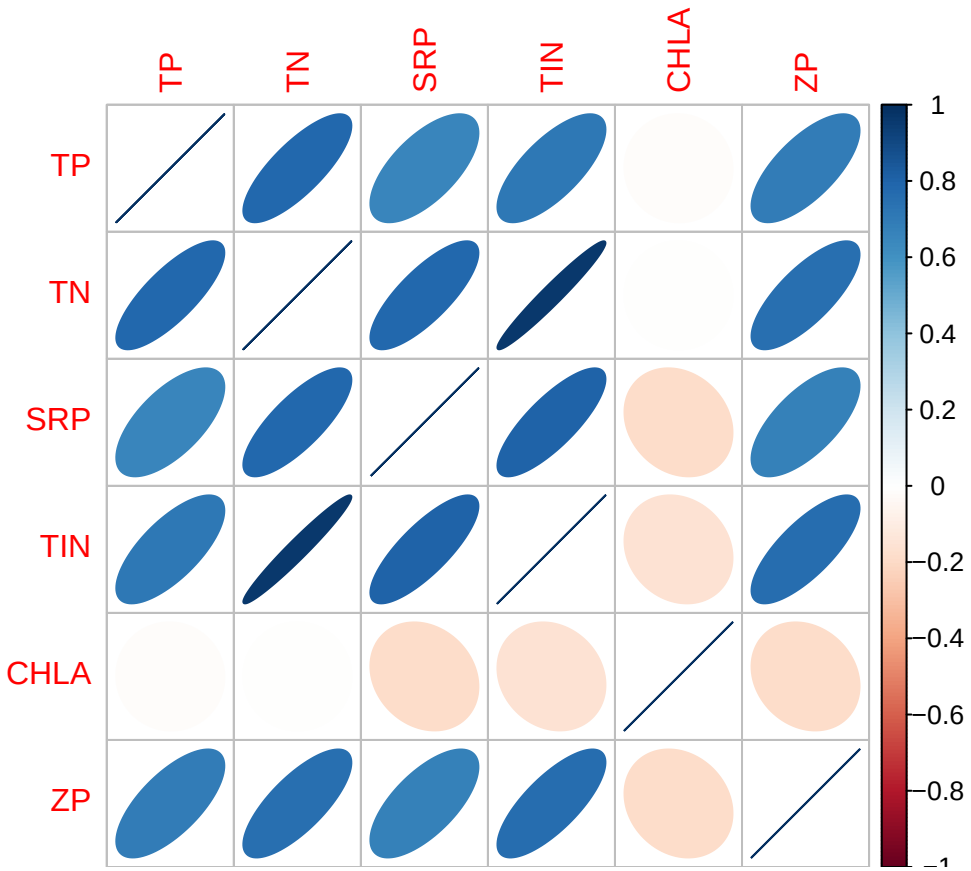
In the R code chunk below, use the `corrplot` function in the *corrplot* package to produce the ellipse correlation plot in the handout.

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

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

```
## Warning: package 'corrplot' was built under R version 3.1.3
```

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



```
# dev.off()
```

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.

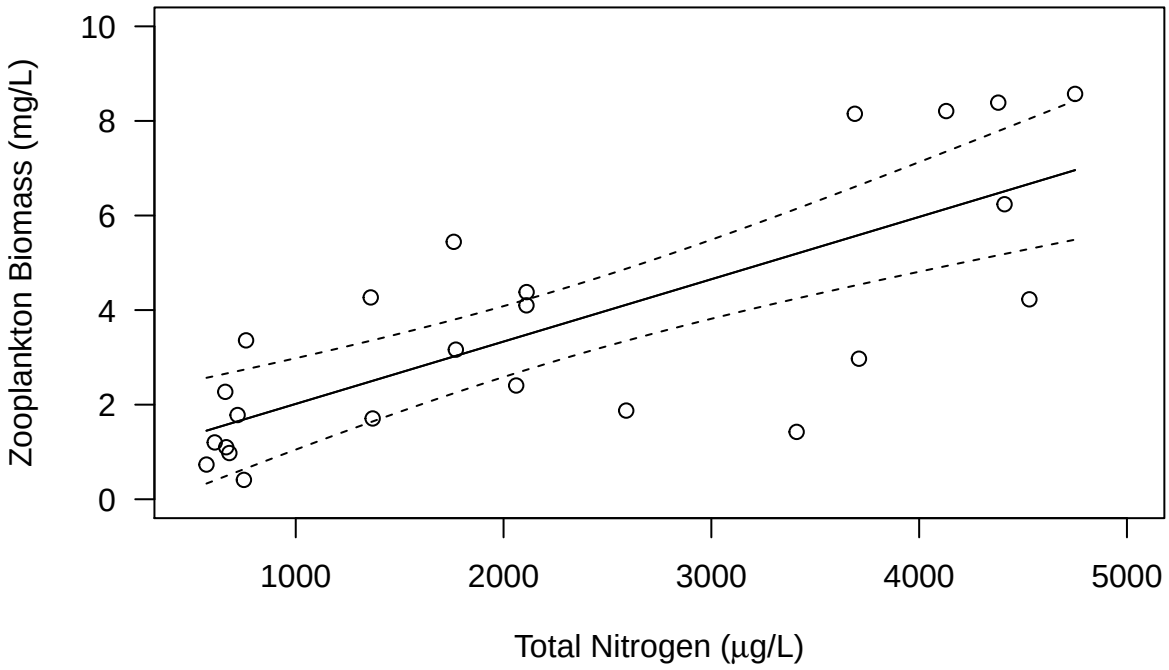
```
#1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zoopl
fitreg <- lm(ZP~TN, data = meso)

#2) Examine the output of the regression analysis.
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
```

```
#3) Produce a plot of this regression analysis including the following: categorically labeled points, t

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$NUTS, pos = 3, cex = 0.8)
text(meso$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)
# the line above calls the previous figure object
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

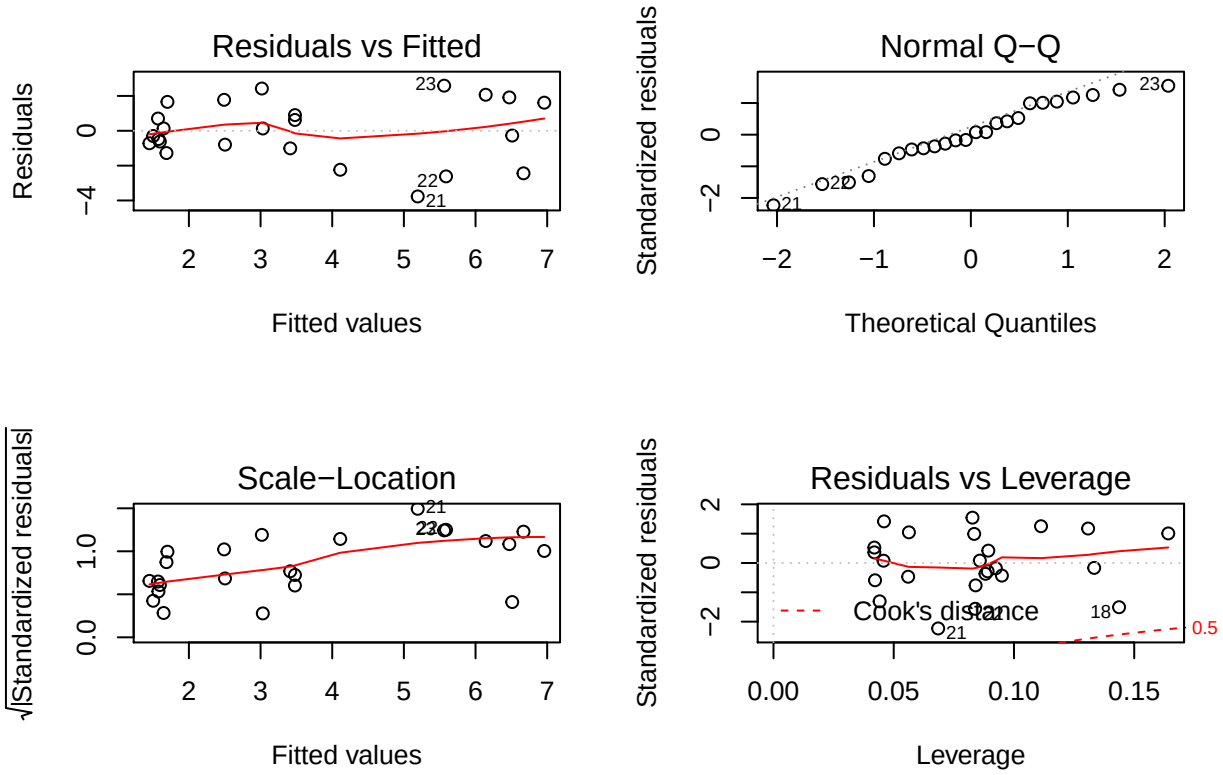
> ***Answer 6***: -Overall, the result from the regression model indicates that total nitrogen content and zooplankton biomass have a positive, linear relationship. -While the slope of this relationship is ~ 0.7 (SE = +/-0.6498643) (0.6977712 + 0.0013181 = 0.6990893) as indicates by the estimate, the strength of the relationship is R^2 ~ 0.55. -The p-values of the intercept indicates that the y-intercept is no different from 0 ($p = 0.294$), while the slope of TN is significantly different from the y-intercept itself ($p < 0.001$)

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

> ***Answer 7***:Predicting the relationship between the two parameters from the regression function that's been fitted and estimating the confidence intervals between them.

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)
```



> ***Answer***:#Both the Residuals vs Fitted and Scale-Location appear to have random distribution of points ('sky at night') #While the Q-Q plot indicates that there is a fairly linear relationship, the residuals at the far end of the plot do not seem to align well #A small proportion of the points lie in the > |1| zone #Overall, there seems to be fair evidence (if not compelling) to indicate that our data fits the assumptions of the linear model

- 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)

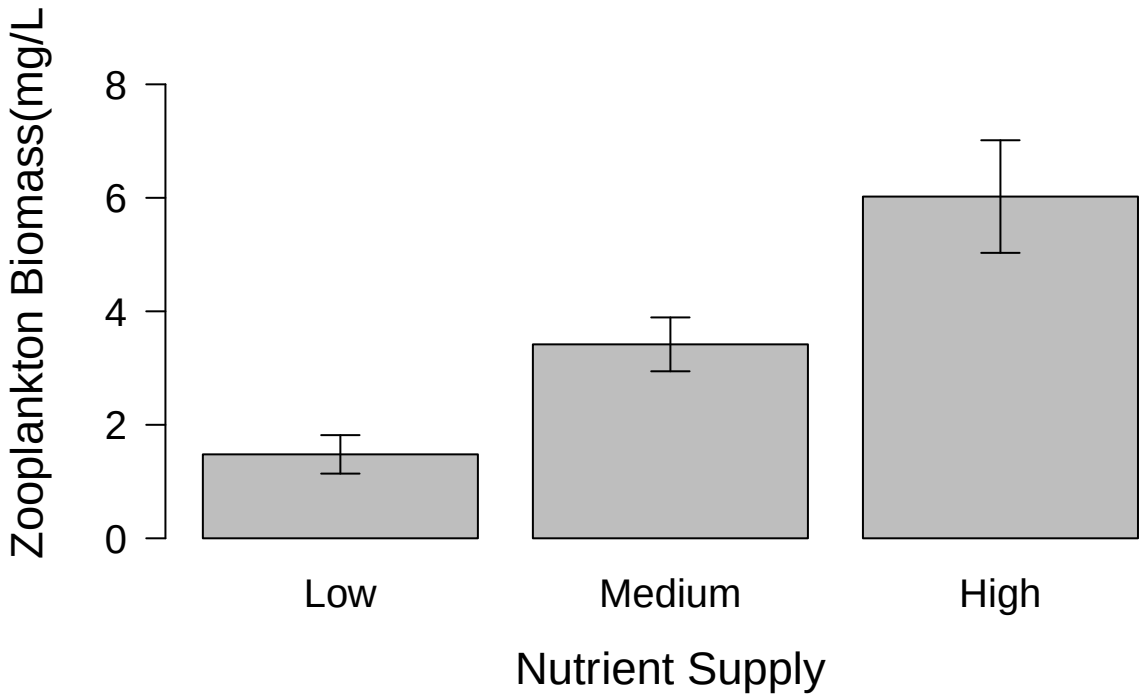
Using the R code chunk below, do the following: 1) Order the nutrient treatments from low to high (see handout). 2) Produce a barplot to visualize zooplankton biomass in each nutrient treatment. 3) Include error bars (+/- 1 sem) on your plot and label the axes appropriately. 4) Use a one-way analysis 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.

```
setwd("C:/Users/DELL/GitHub/QB2017_Ramesh/Week1/data")
#1) Order the nutrient treatments from low to high (see handout).
meso <- read.table("zoop_nuts.txt", sep = "\t", header = TRUE)
NUTS <- factor(meso$NUTS, levels =c('L','M','H'))
head(meso)
```

```
##   TANK NUTS    TP    TN  SRP    TIN CHLA     ZP
## 1   34    L 20.31 720.1 4.02 131.62 1.52 1.7808
## 2   14    L 25.55 750.5 1.56 141.10 4.00 0.4090
## 3   23    L 14.22 610.1 4.97 107.70 0.61 1.2014
## 4   16    L 39.11 760.9 2.89  71.28 0.53 3.3598
## 5   21    L 20.09 570.4 5.11  80.40 1.44 0.7332
## 6    5    L 15.75 680.5 4.68 135.77 1.19 0.9773
```

```
#2) Produce a barplot to visualize zooplankton biomass in each nutrient treatment
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.la

#3) Include error bars (+/- 1 sem) on your plot and label the axes appropriately.
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)
```




```
#4) Use a one-way analysis of variance (ANOVA) to test the null hypothesis that zooplankton biomass is
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
```

```
#5) Use a Tukey's HSD to identify which treatments are different.
TukeyHSD(fitanova)
```

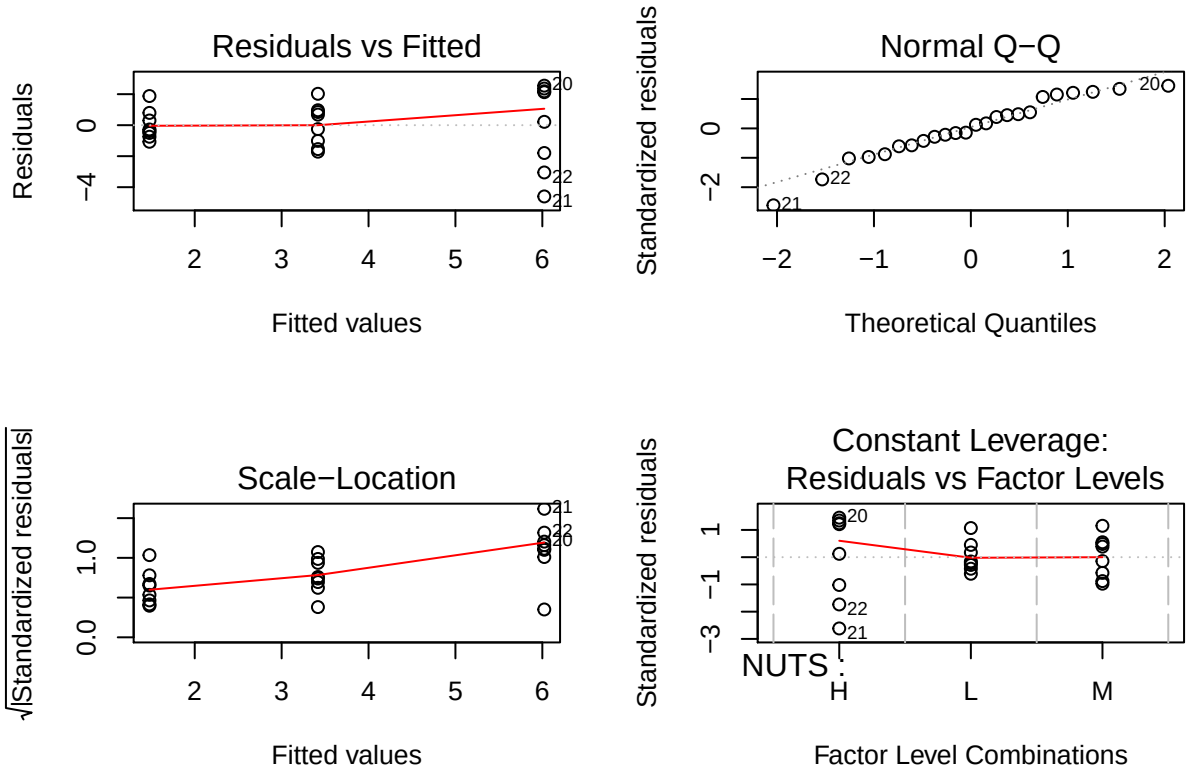
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ZP ~ NUTS, data = meso)
##
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## M-L  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***: The ANOVA results here indicate that changes in nutrient levels does explain the changes in zooplankton biomass ($p < 0.001$). The TukeyHSD shows that zooplankton biomass in the Low and Medium nutrient levels are indeed different from the High nutrient levels. Overall, these results corraborate with the results of the regression, as in the previous analysis with nutrient as a contious predictor also yielded a signficant effect of nutrient content on zoo. biomass. I do not have any pressing concerns about this 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).

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



> ***Answer***: Overall, it does appear that the data meets the assumptions of ANOVA. Since the data is categorical variable, the residuals are clumped to three levels across the graph. However, within these levels, the data points are randomly disperesed from the zero line (Resuduals and Scale-Location graphs). The deviations in the Q-Q plot again could be because of the grouping of data.

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.

```
setwd("C:/Users/DELL/GitHub/QB2017_Ramesh/Week1/data")
dat <- read.table("zoops.txt", sep = "\t", header = TRUE)
dat <- dat[,-c(1,2)]

#Total biomass per site
ZP <- rowSums(dat, na.rm = FALSE, dims = 1)
dat <- cbind(dat, ZP)

mod1 <- lm(ZP ~ CAL, data = dat)
summary(mod1) #Not signficant, R = 0.05
```

```
##
## Call:
## lm(formula = ZP ~ CAL, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3299.6 -2043.0  -445.7  1758.5  4505.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4072.652    591.410   6.886 6.47e-07 ***
## CAL          -13.250      8.771  -1.511    0.145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2545 on 22 degrees of freedom
## Multiple R-squared:  0.09398,    Adjusted R-squared:  0.0528
## F-statistic: 2.282 on 1 and 22 DF,  p-value: 0.1451
```

```
mod2 <- lm(DIAP ~ ZP, data = dat)
summary(mod2) #Not signficant, R = 0.047
```

```
##
## Call:
## lm(formula = DIAP ~ ZP, data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -62.98 -47.65 -31.01  10.65 223.30
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 72.273418  28.673066   2.521   0.0195 *
## ZP          -0.009449   0.006437  -1.468   0.1563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 80.72 on 22 degrees of freedom
## Multiple R-squared:  0.08922,    Adjusted R-squared:  0.04782
## F-statistic: 2.155 on 1 and 22 DF,  p-value: 0.1563
```

```
mod3 <- lm(CYCL ~ ZP, data = dat)
summary(mod3) # p<0.1, R = 0.08
```

```
##
## Call:
## lm(formula = CYCL ~ ZP, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -118.785  -50.221   -5.281   14.068  258.452
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 144.633973  30.745180   4.704 0.000108 ***
## ZP           -0.012307   0.006902  -1.783 0.088365 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86.55 on 22 degrees of freedom
## Multiple R-squared:  0.1263, Adjusted R-squared:  0.08656
## F-statistic:  3.18 on 1 and 22 DF,  p-value: 0.08837
```

```
mod4 <- lm(BOSM ~ ZP, data = dat)
summary(mod4) #Not signficant, R = 0.0025
```

```
##
## Call:
## lm(formula = BOSM ~ ZP, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8467 -1.5205 -0.9622 -0.0339  9.3047
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.9402433  0.9776894   1.985   0.0598 .
## ZP          -0.0002259  0.0002195  -1.029   0.3145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.752 on 22 degrees of freedom
## Multiple R-squared:  0.04594,    Adjusted R-squared:  0.002578
## F-statistic: 1.059 on 1 and 22 DF,  p-value: 0.3145
```

```
mod5 <- lm(SIMO ~ ZP, data = dat)
summary(mod5) # p < 0.01, R = 0.1487
```

```
##
```

```
## Call:
## lm(formula = SIMO ~ ZP, data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -670.5 -332.5 -194.7  231.4 1540.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.78965  183.02233   0.583   0.5655
## ZP            0.09203    0.04109   2.240   0.0355 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 515.2 on 22 degrees of freedom
## Multiple R-squared:  0.1857, Adjusted R-squared:  0.1487
## F-statistic: 5.017 on 1 and 22 DF,  p-value: 0.03552
```

```
mod6 <- lm(CERI ~ ZP, data = dat)
summary(mod6) #Not signficant, R = -0.024
```

```
##
## Call:
## lm(formula = CERI ~ ZP, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -134.62  -62.46  -18.38   22.56  406.39
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 146.858923  40.314798   3.643  0.00144 **
## ZP           -0.006035   0.009050  -0.667  0.51182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 113.5 on 22 degrees of freedom
## Multiple R-squared:  0.01981,    Adjusted R-squared:  -0.02474
## F-statistic: 0.4446 on 1 and 22 DF,  p-value: 0.5118
```

```
mod7 <- lm(NAUP ~ ZP, data = dat)
summary(mod7) #Not signficant, R = 0.01655
```

```
##
## Call:
## lm(formula = NAUP ~ ZP, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8725 -0.6862 -0.2581  0.5743  2.3344
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  9.047e-01  2.945e-01   3.072  0.00558 **
## ZP          -7.787e-05  6.612e-05  -1.178  0.25150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8292 on 22 degrees of freedom
## Multiple R-squared:  0.05931,    Adjusted R-squared:  0.01655
## F-statistic: 1.387 on 1 and 22 DF,  p-value: 0.2515
```

```
mod8 <- lm(DLUM ~ ZP, data = dat)
summary(mod8) #Not significant, R = -0.00078
```

```
##
## Call:
## lm(formula = DLUM ~ ZP, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6192 -0.4749 -0.3152 -0.0617  6.0548
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6632404  0.4787360   1.385    0.180
## ZP          -0.0001065  0.0001075  -0.991    0.332
##
## Residual standard error: 1.348 on 22 degrees of freedom
## Multiple R-squared:  0.04273,    Adjusted R-squared:  -0.0007861
## F-statistic: 0.9819 on 1 and 22 DF,  p-value: 0.3325
```

```
mod9 <- lm(CHYD ~ ZP, data = dat)
summary(mod9) #p-value < 0.001, R = 0.9609
```

```
##
## Call:
## lm(formula = CHYD ~ ZP, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1463.1  -293.2   107.4   334.4   766.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -532.15333  176.62922  -3.013   0.0064 **
## ZP             0.94326    0.03965  23.789   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 497.3 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
```

Answer:The regression of the species taxa with total biomass shows that of all species, only CHYD, CYCL and SIMO are primarily responsible for the total biomass (ZP). Of these, CHYD was the largest predictor of total biomass where R^2 = 0.96.

```

## 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$^{th}$, 2015 at 12:00 PM (noon)**.
```