Week 1 Assignment: Basic R

Erik Parker; 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.

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

```
## [1] "/var/host/media/removable/USB Drive/GitHub/QB2017_Parker/Week1"
```

```
setwd("/var/host/media/removable/USB Drive/GitHub/QB2017_Parker/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.

```
#Volume of a cube with length=5?
l <- 5
Volume <- l^3
Volume
```

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

```
# Area of a circle with radius r=2?
radius <- 2
Area <- pi*radius^2
Area
```

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

```
# Length of opposite side of a right-triangle?
theta <- pi/4
hypotenuse <- sqrt(2)
opposite.length <- sin(theta)*hypotenuse
opposite.length
```

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

```
# log(e) of 7?
log(7)
```

```
## [1] 1.94591
```

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)
w <- x*14
(x+w)/15
```

```
## [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(15,25,46,12,89)
k*x
```

```
## [1]  15  50 138  48 445
```

```
d <- c(w[2:5], k[1:4])
d
```

```
## [1] 28 42 56 70 15 25 46 12
```

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)

summary.v <- summary(v)
summary.v
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   10.10   16.50   20.35   20.90   24.95   31.40       1
```

```
sum.v <- sum(v, na.rm = TRUE)
sum.v
```

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

```
var.v <- var(v, na.rm = TRUE)
var.v
```

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

```
sd.v <- sd(v, na.rm = TRUE)
sd.v
```

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

```
standard.error.v <- sd(v, na.rm = TRUE)/sqrt(length(v))
standard.error.v
```

```
## [1] 1.621522
```

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.

```
first <- c(rnorm(5, mean = 8, sd = 2))
second <- c(rnorm(5, mean = 25, sd = 10))
matrix.q5 <- matrix(c(first,second),nrow = 5, ncol = 2)
matrix.q5
```

```
##           [,1]      [,2]
## [1,] 7.631278 27.944427
## [2,] 4.608784 12.411565
## [3,] 7.589052  0.947674
## [4,] 8.036372 30.685193
## [5,] 2.110638 24.473120
```

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 "rnorm" function generates a random set of numbers of length "n" which fit the normal distribution with a certain mean and standard deviation specified by the arguments "mean" and "sd" respectively.

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")
m.transposed <- t(m)
dim(m.transposed)
```

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

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

> Answer 2: The transposed matrix is 5X10. It has 5 rows and 10 columns.

Indexing a Matrix

In the R code chunk below, do the following: 1) Index matrix `m` by selecting all but the third column. 2) Remove the last row of matrix `m`.

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

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

```
m.nolast <- m[1:9,]
m.nolast
```

```
##   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***: In the first step we wanted to retreive all columns aside from the third one. To do this we needed to use square brackets to return data from the matrix "m", and then additionally specify which columns we were interested in by using "c()" to combine the columns of interest into a single vector which was placed to the right of the comma within the square brackets used as this position corresponds to columns. For the second step I used a very similar process, this time creating a new matrix called "m.nolast" which contained only the first 9 rows of matrix "m". The main difference here being that I was interested in retreiving the rows, so my entry was to the left of the comma, and the rows of interest were sequential so I could use a colon instead of the "c()" command.

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.

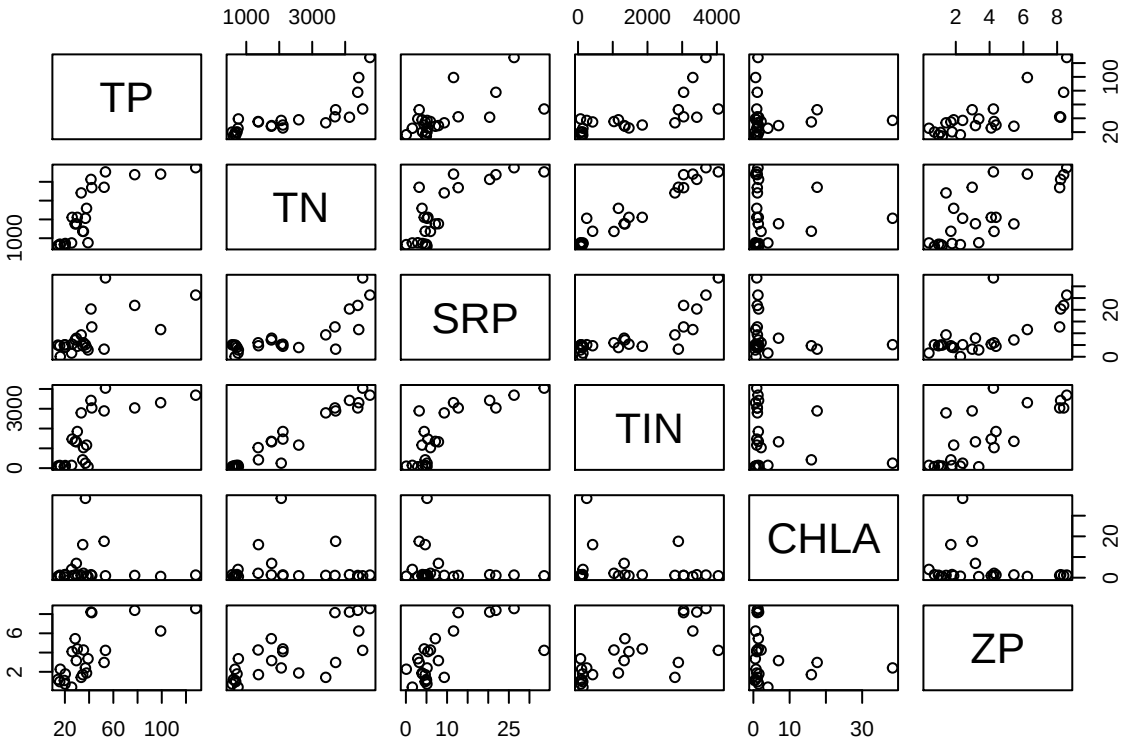
```
zoo <- read.table("data/zoop_nuts.txt", sep = "\t", header = TRUE)
str(zoo)
```

```
## '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.

```
zoo.m <- as.matrix(zoo[,c(3:8)])
pairs(zoo.m)
```




```
cor1.zoo.m <- cor(zoo.m)
cor1.zoo.m
```

```
##                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: Probably as expected, many of the inorganic nutrient concentration variables are very highly correlated with the total inorganic nutrient concentration variable. More interesting are the findings that zooplankton biomass is pretty highly correlated with all of the inorganic nutrient levels (especially Nitrogen and total concentration), and that it is (very) slightly inversely correlated wit chlorophyll a concentrations.

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")
require(psych)
```

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

```
cor2.zoo.m <- corr.test(zoo.m, method = "pearson", adjust = "BH")
cor.nonpar.zoo.m <- corr.test(zoo.m, method = "kendall", adjust = "BH")
print(cor2.zoo.m)
```

```
## Call:corr.test(x = zoo.m, 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
```

```
print(cor.nonpar.zoo.m)
```

```
## Call:corr.test(x = zoo.m, method = "kendall", adjust = "BH")
## Correlation matrix
##        TP   TN   SRP  TIN  CHLA    ZP
## TP   1.00 0.74  0.39 0.58  0.04  0.54
## TN   0.74 1.00  0.48 0.81  0.01  0.55
## SRP  0.39 0.48  1.00 0.56 -0.07  0.45
## TIN  0.58 0.81  0.56 1.00  0.04  0.55
## CHLA 0.04 0.01 -0.07 0.04  1.00 -0.05
## ZP   0.54 0.55  0.45 0.55 -0.05  1.00
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##        TP   TN  SRP  TIN CHLA   ZP
## TP   0.00 0.00 0.09 0.01 0.90 0.01
## TN   0.00 0.00 0.03 0.00 0.95 0.01
## SRP  0.06 0.02 0.00 0.01 0.90 0.05
## TIN  0.00 0.00 0.00 0.00 0.90 0.01
## CHLA 0.84 0.95 0.76 0.84 0.00 0.90
## ZP   0.01 0.01 0.03 0.01 0.81 0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Question 5: Describe what you learned from `corr.test`. Describe what you learned from corr.test. Specifically, are the results sensitive to whether you use parametric (i.e., Pearson's) or non-parametric

methods? When should one use non-parametric methods instead of parametric methods? With the Pearson's method, is there evidence for false discovery rate due to multiple comparisons? Why is false discovery rate important?

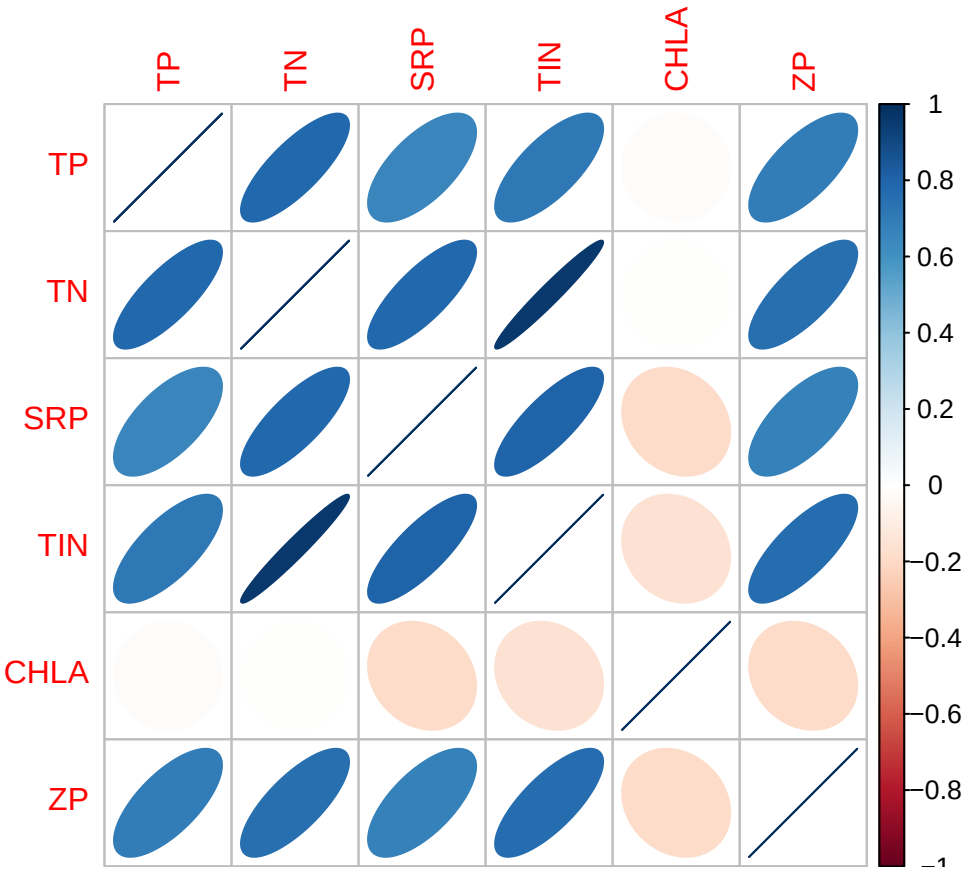
> Answer 5: From corr.test I learned that the results of a correlation test are quite different between parametric and non-parametric methods. Here the parametric Pearson's test returned much higher correlation values across the board than did the non-parametric Kendall's test. Additionally, In general, it seems best to rely on non-parametric methods when the data being analyzed doesn't clearly come from a normal distribution, when dealing with ordinal or ranked data, when sample sizes are small (and don't adaquately reflect a normal population) and when there are a significant number of outliers present in the dataset. There does seem to be evidence for false discovery with Pearson's as the raw probability values reported below the diagonal are in some cases lower (more significant) than the probabilities above the diagonal which have been adjusted for multiple comparisons. In this particular case it doesn't seem to be of too much concern though as the majority of the probabilities are apparently < 0.00 either raw or adjusted. In general though, false discovery rate is important when multiple comparisons are performed, especially when only testing at an arbitrary confidence level such as 0.05, because as the number of comparisons increases it becomes more and more likely of finding a result of $p < 0.05$ due just to chance alone. As an example, if we are running 1000 comparisons, we might expect to find 50 results as significant in error if we use a standard cutoff of 0.05

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")
require("corrplot")
```

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

```
corrplot(cor1.zoo.m, 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.

```
ZTreg <- lm(ZP ~ TN, data = zoo)
ZTreg
```

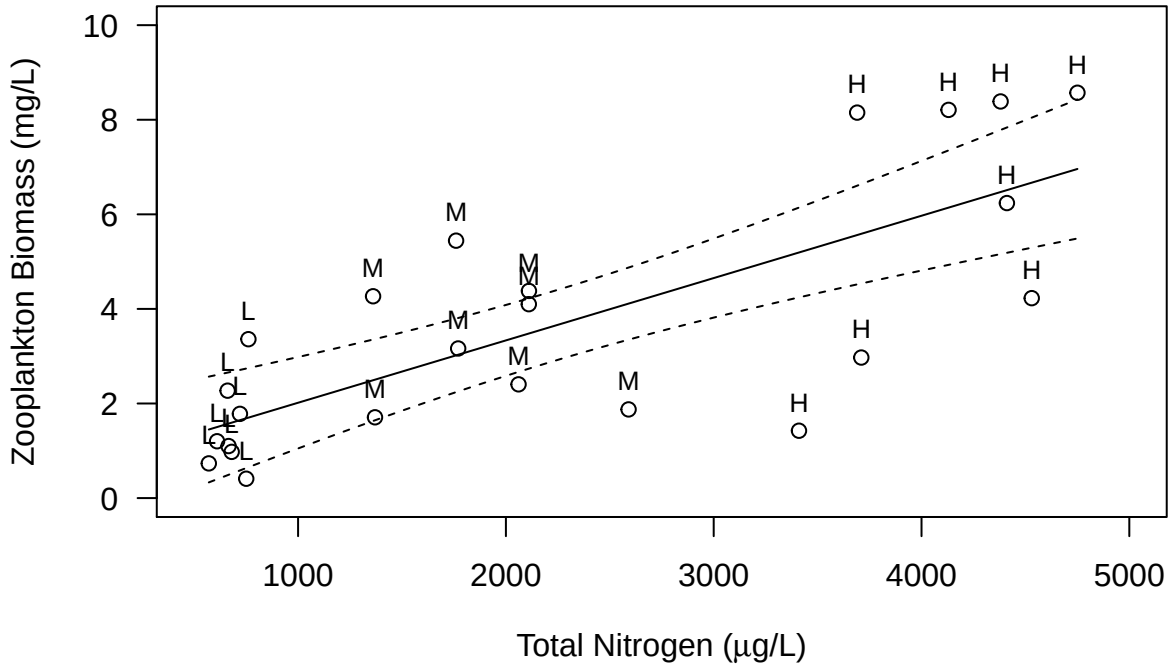
```
##
## Call:
## lm(formula = ZP ~ TN, data = zoo)
##
## Coefficients:
## (Intercept)           TN
##    0.697771     0.001318
```

```
summary(ZTreg)
```

```
##
## Call:
## lm(formula = ZP ~ TN, data = zoo)
##
## 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(zoo$TN, zoo$ZP, ylim = c(0,10), xlim = c(500,5000), xlab = expression(paste("Total Nitrogen (", mu
text(zoo$TN, zoo$ZP, zoo$NUTS, pos = 3, cex =0.8)
newTN <- seq(min(zoo$TN), max(zoo$TN), 10)
regline <- predict(ZTreg, newdata = data.frame(TN = newTN))
lines(newTN, regline)
conf95 <- predict(ZTreg, newdata = data.frame(TN = newTN), interval = c("confidence"), level = 0.95, ty
matlines(newTN, conf95[, c("lwr", "upr")], type = "l", lty = 2, lwd = 1, col = "black")
```



Question 6: Interpret the results from the regression model

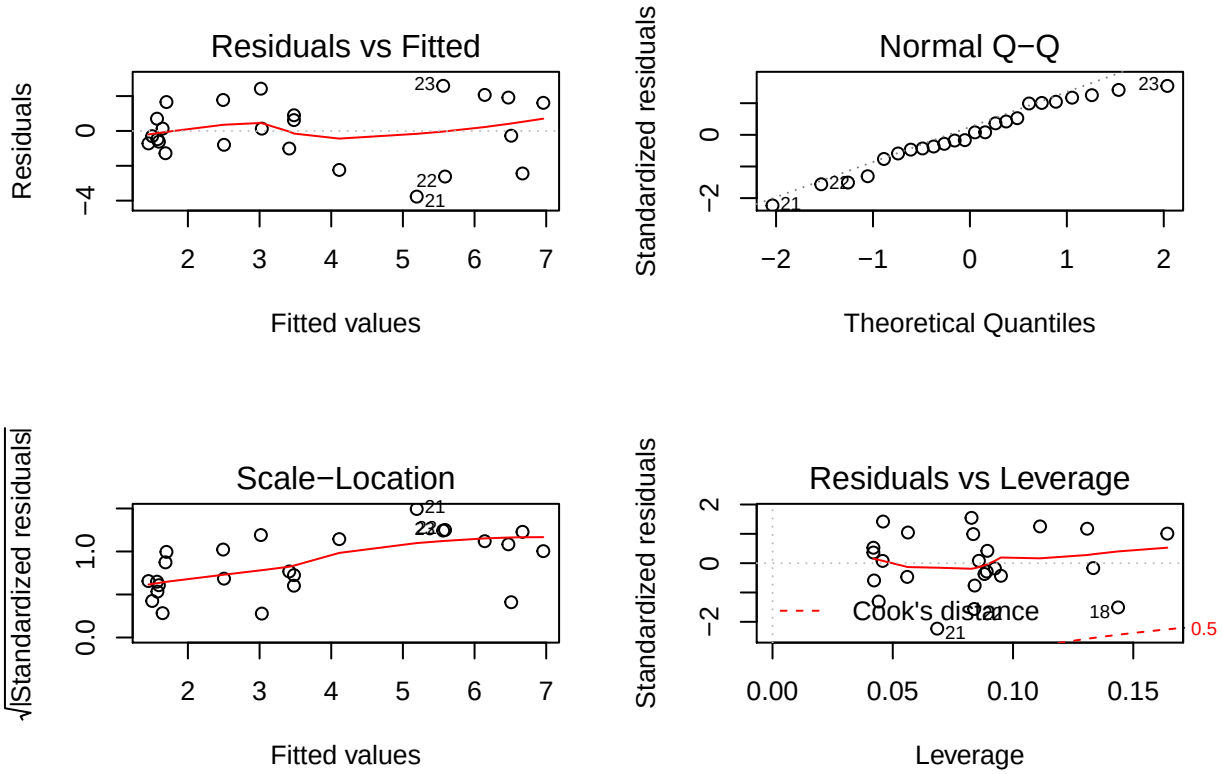
> ***Answer 6***: The above linear regresion seems to show a pretty good predictive ability of Total Nitrogen concentration for overall Zooplankton Biomass across all treatments. When we look more closely at individual tratments though, it seems as though this predictive model performs best at low and medium nutrient levels (many L and M points falling within the 95% confidence interval), and not so well at high condition (most H points falling outside the 95% CI).

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

> ***Answer 7***: In the above analyses, the predict() function was used to generate predicted values for Zooplnakton Biomass for particular Total Nitrogen levels based on the results of the linear regression for ZP on TN performed earlier. These predicted values were then used to generate the regression line and 95% CI curves displayed on the graph above.

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(ZTreg)
```



- Upper left: is there a random distribution of the residuals around zero (horizontal line)? > Not perfect, but it looks pretty good. I think that the resulting line is close enough to horizontal for us to conclude that the distribution of the residuals is close to random.
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)` > Certainly not the worst QQ plot I've ever seen. This relationship appears to be quite close to normal, not perfect by any means but very good for real data.
- Bottom left: again, looking for a random distribution of sqrt(standardized residuals) > This distribution appears to be less random than the upper left plot without the sqrt transformation applied to the residuals. I don't think this is neccessarily anything to be concerned with though, as the non-transformed values seemed to fit the assumption of random distribution just fine.
- Bottom right: leverage indicates the influence of points; contours correspond with Cook's distance, where values > |1| are "suspicious" > There seem to be a fair number of datapoints with values for cooks distance >|1|, with three points in particular (18, 21, and 22) being close to or greater than |2|. It would be interesting to run the analyses again without these outliers and see if the results are greatly affected with these highly influential points removed.

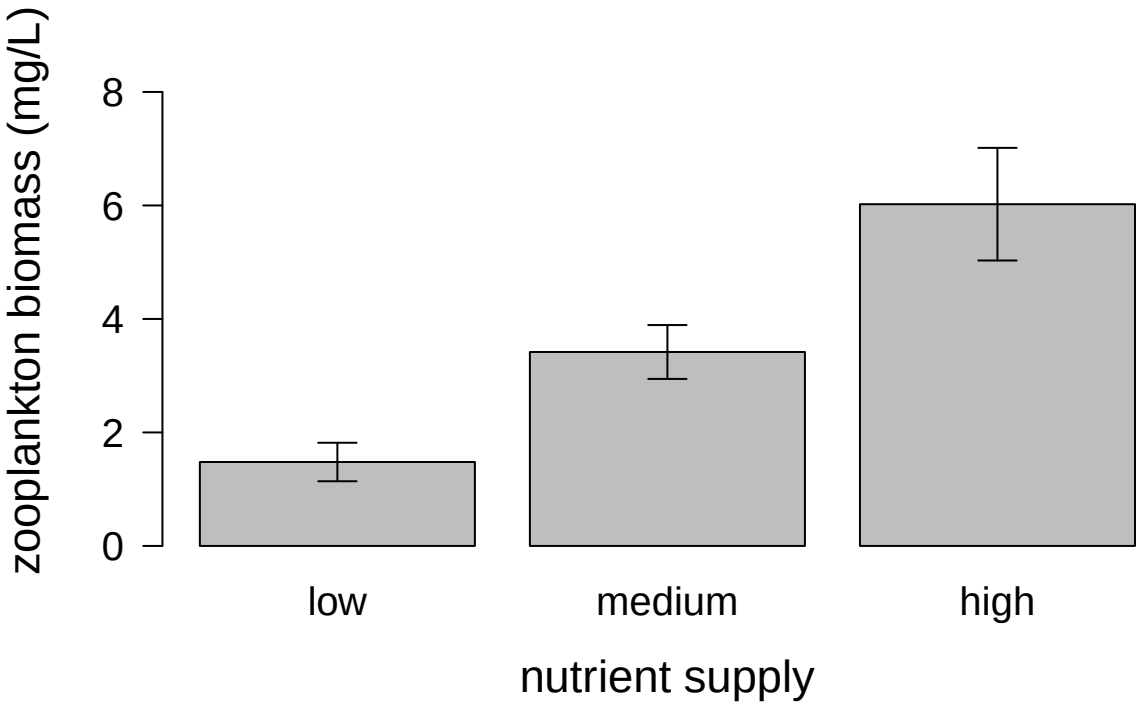
Analysis of Variance (ANOVA)

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(zoo$NUTS, levels = c('L', 'M', 'H'))
zp.means <- tapply(zoo$ZP, Nuts, mean)
sem <- function(x){sd(na.omit(x))/sqrt(length(na.omit(x)))}
zp.sem <- tapply(zoo$ZP, Nuts, sem)

bp <- barplot(zp.means, ylim = c(0, round(max(zoo$ZP), digits = 0)), pch = 15, cex =1.25, las = 1, cex.
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 = zoo)
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)
```

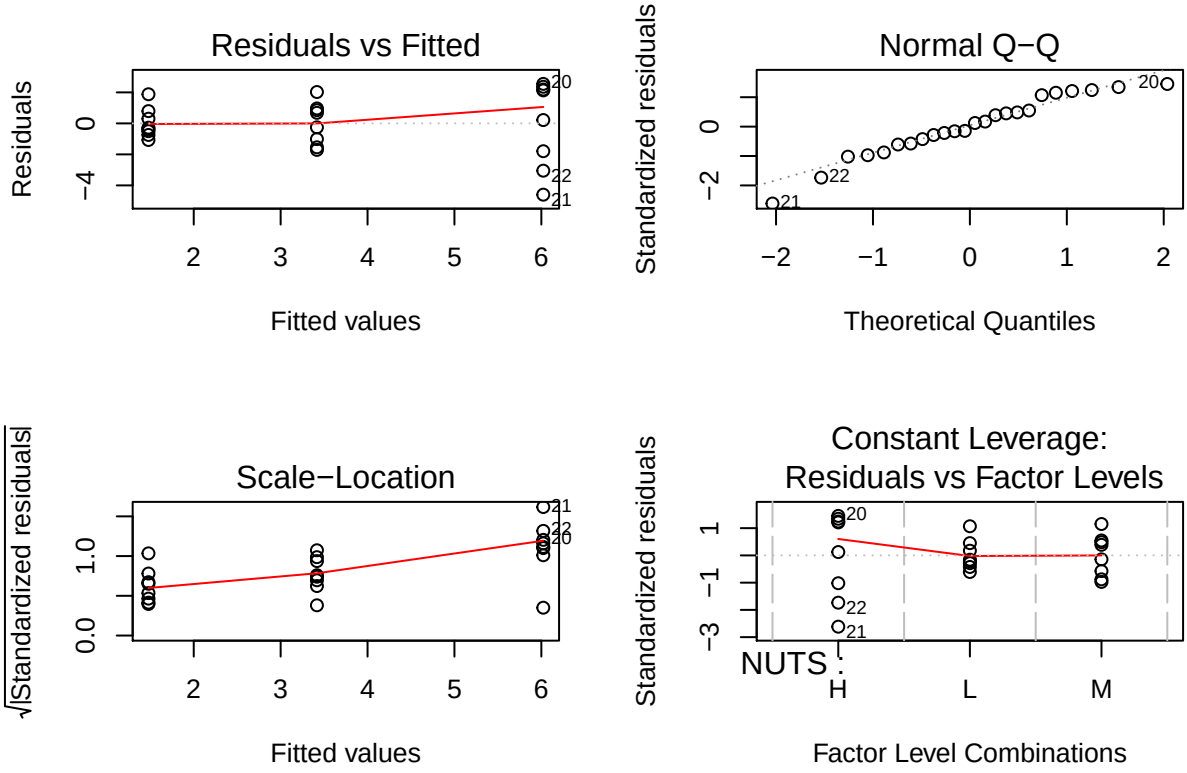
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ZP ~ NUTS, data = zoo)
##
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## M-L  1.938625 -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***: These overall ANOVA results seem to be saying that there is a real difference between the three nutrient groups being analyzed in this dataset, and the post-hoc Tukey's test tells us that this overall difference is really coming from differences between the low and high, and medium and high samples - but not the medium and low samples. This causes me some concern when compared to the regression results, as there we saw that the medium and low treatments fit the analysis quite well and fell along the regression line whereas the high treatment as a whole was by and large not well represented by the predicted regression line or 95% confidence intervals. I am not completely confident in my analysis here, but I think that this means that we might just be seeing a significant difference between the medium/low and high treatments (but not medium and low) just because the high treatment samples did not fit the predicted relationship between TN and ZP which was more generated thanks to the weight of the agreement between the large number of low and medium samples outweighing the disagreement from the high samples.

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: Aside from the constant leverage analysis which contains many points of concern in the high nutritional treatment group (consistent, I believe, with my concerns expressed above in answer 8) the other three graphs seem to show that the data meet the assumptions of normality and homoscedasticity.

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.

```
zoop <- read.table("data/zoops.txt", sep = "\t", header = TRUE)
zoop.m <- as.matrix(zoop[,3:11])
mean.zoop.Low <- zoop.m[mean(1:8),]
mean.zoop.Medium <- zoop.m[mean(9:16),]
mean.zoop.High <- zoop.m[mean(17:24),]
mean.zoop.L <- zoop[mean(1:8),2:11]
mean.zoop.M <- zoop[mean(9:16),2:11]
mean.zoop.H <- zoop[mean(17:24),2:11]
mean.zoop.L
```

```
##   NUTS  CAL DIAP  CYCL BOSM SIMO CERI NAUP DLUM  CHYD
## 4    L 79.2 17.9 141.3  3.4    0  199    0    0 298.5
```

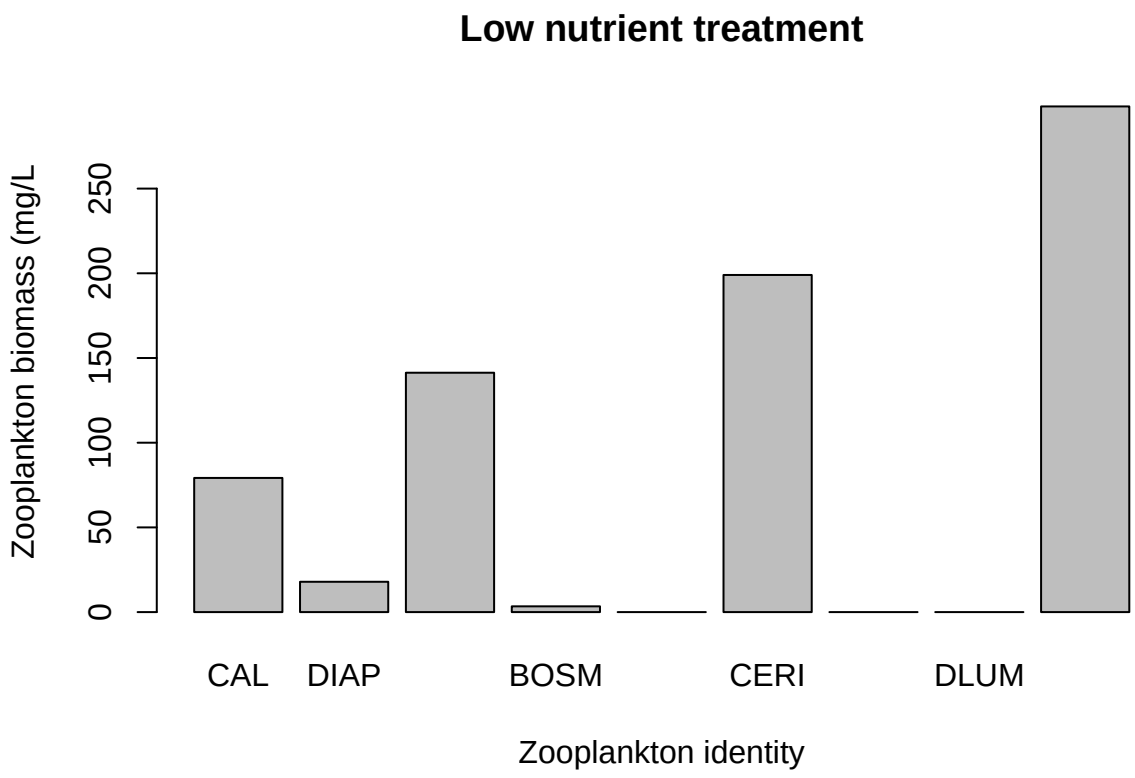
```
mean.zoop.M
```

```
##    NUTS CAL DIAP CYCL BOSM   SIMO CERI NAUP DLUM   CHYD
## 12    M  14  2.3 37.7    0 1251.5 74.8    0    0 2725.5
```

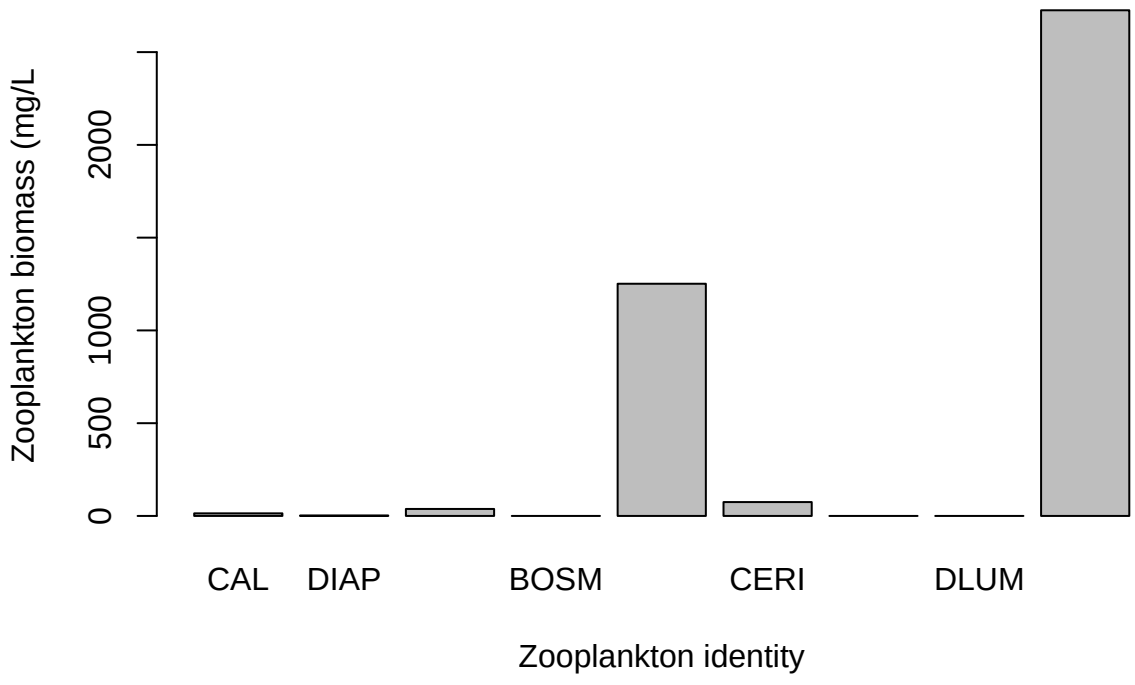
```
mean.zoop.H
```

```
##    NUTS CAL DIAP CYCL BOSM  SIMO CERI NAUP DLUM   CHYD
## 20    H  14  7.5 69.5    0 594.2 78.5    0    0 7629.2
```

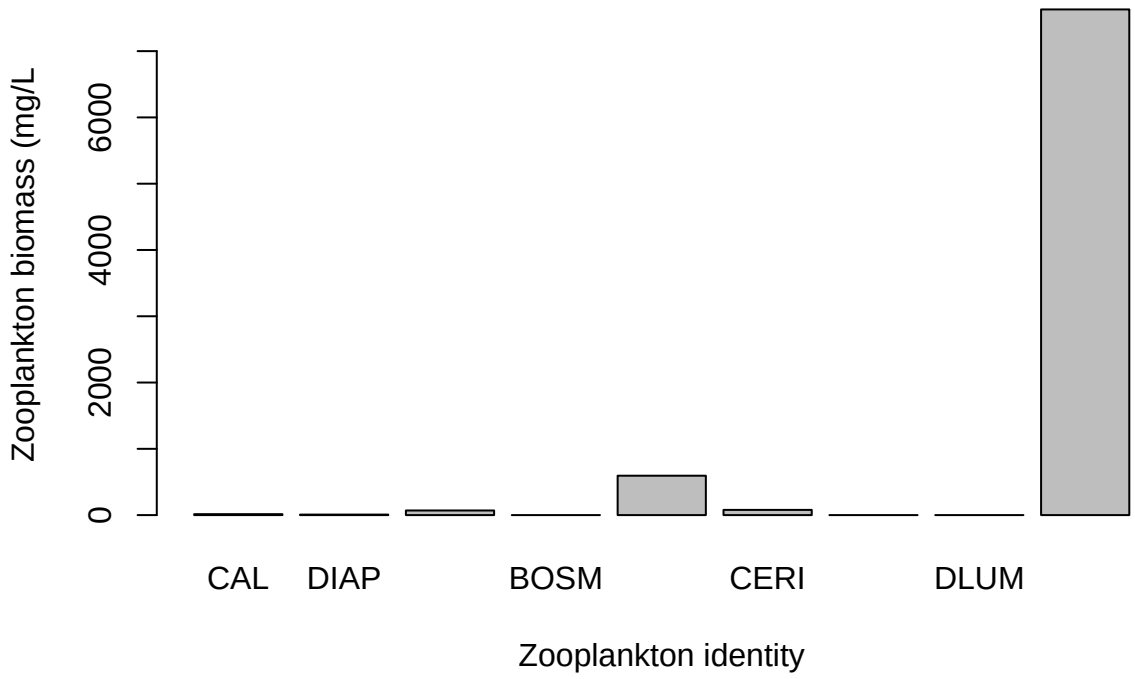
```
barplot(mean.zoop.Low, xlab = "Zooplankton identity", ylab = "Zooplankton biomass (mg/L", main = "Low nu
```




```
barplot(mean.zoop.Medium, xlab = "Zooplankton identity", ylab = "Zooplankton biomass (mg/L", main = "Me
```




```
barplot(mean.zoop.High, xlab = "Zooplankton identity", ylab = "Zooplankton biomass (mg/L", main = "High
```



Answer: Above, we learned that as nutrient enrichment increases so does total zooplankton biomass. Here in this analysis we saw that at both medium and high nutrient levels the most abundant taxa by total biomass were the Chydorus. This was especially true at high nutrient condition where the closest taxa by biomass, the Simocephallus, were almost 13 times less abundant. At medium nutrient levels, the relationship between Chydorus and Simocephallus was less extreme, with Chydorus responsible for a little over twice as much biomass. At low nutrient conditions, the picture is a little different, though Chydorus does still represent the single largest taxa in terms of biomass. Here though, both the Ceriodaphnia species and cyclopoid copepods contribute about half of the biomass generated by Chydorus. So while one taxa still reigns as king of the biomass at low nutrient conditions, the difference between it and other community members is not nearly as great as it is at higher nutrient conditions.

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)**.