Week 1 Assignment: Basic R

Roy Z Moger-Reischer; 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.

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

```
## [1] "C:/Users/rmoge/GitHub/QB2017_Moger-Reischer/Week1"
```

```
#setwd("~/GitHub/QB2017_Moger-Reischer/Week1")
#later, on my own computer
setwd("C:\\Users\\rmoge\\GitHub\\QB2017_Moger-Reischer\\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.

```
l<-5
V1<-l*l*l

r<-2
A2<-pi*r*r

theta<-pi/4
H<-sqrt(2)
O<-sin(theta)*H

log(127)
```

```
## [1] 4.844187
```

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(rnorm(5, mean=0, sd=1))
w<-x*14
number3<-(x+w)/15
```

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(1,1,1,1,1)
thuggy2<-k*x
d<-c(w[1:3],k[1:4])
```

Summary Statistics of Vectors

In the R code chunk below, calculate the **summary statistics** (i.e., maximum, minimum, sum, mean, median, variance, standard deviation, and standard error of the mean) for the vector (`v`) provided.

```
v <- c(16.4, 16.0, 10.1, 16.8, 20.5, NA, 20.2, 13.1, 24.8, 20.2, 25.0, 20.5, 30.5, 31.4, 27.1)
(max(na.omit((v)))); min(na.omit(v)); sum(na.omit(v)); mean(na.omit(v))
```

```
## [1] 31.4

## [1] 10.1

## [1] 292.6

## [1] 20.9
```

```
median(na.omit(v)); var(na.omit(v)); sd(na.omit(v))
```

```
## [1] 20.35

## [1] 39.44

## [1] 6.280127
```

```
SEm<-function(a){
  sd((a))/sqrt(length((a)))
}
SEm(na.omit(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.

```
j<-c(rnorm(5, mean=8, sd=2))
k<-c(rnorm(5,mean=25,sd=10))
mguy<-matrix(c(j,k),nrow=5,ncol=2,byrow=FALSE)
```

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 randomly choose numbers based on a given distribution. specifically, a normal distribution, which mean and stddev specified by the user, or default values of 0,1,respectively (isn't that a Z distrn?)

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=FALSE))
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
```

```
xpose_mguy2<-t(m)
dim(xpose_mguy2)
```

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

```
xpose_mguy2
```

```
##    [,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
```

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

> Answer 2: 5 rows, 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`.

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

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

> ***Answer 3***: I kept all rows, and selected slices of columns. When asked to modify the matrix itself, I set m equal to an updated version of m.

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.

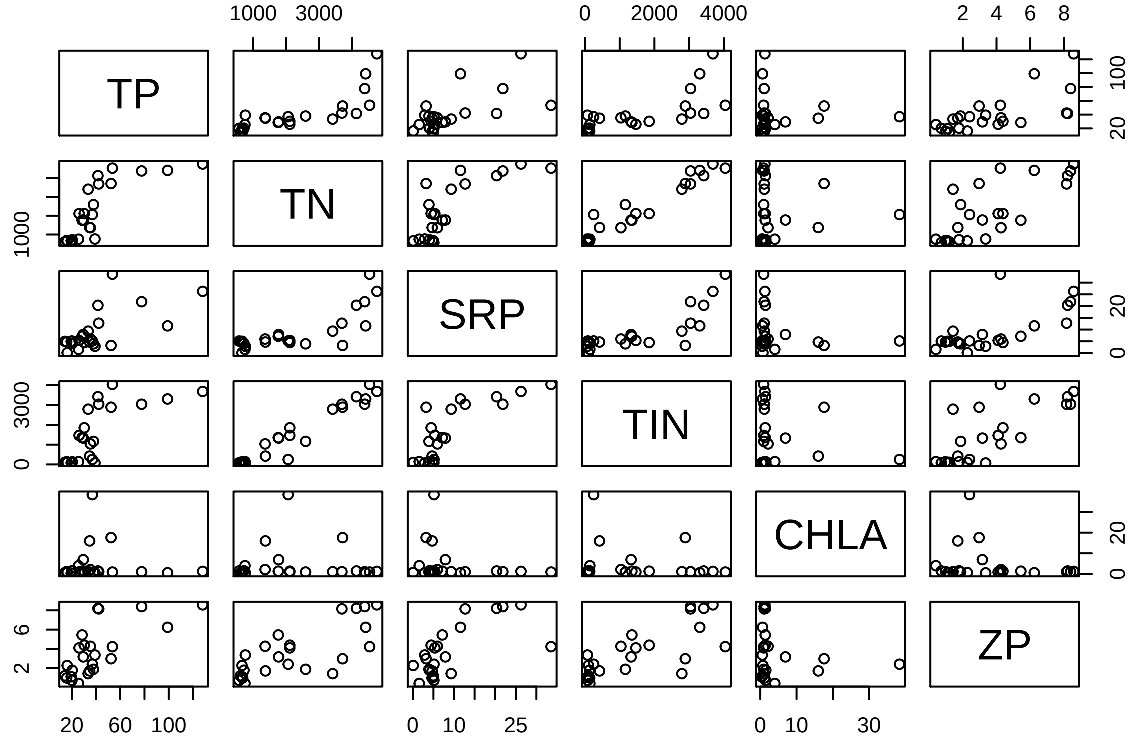
```
ROYZOOP<-read.table("data\\zoop_nuts.txt",sep="\t",header=TRUE)
str(ROYZOOP)
```

```
## '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.

```
NUMZOOP<-ROYZOOP[,3:8]
pairs(NUMZOOP)
```



```
cor3_8<-cor(NUMZOOP)
cor3_8
```

```
##               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: N and P in all their forms are positively correlated. Zooplankton are positively correlated with the nutrients. Chl a is negatively correlated with all of the above, but the correlation is less strong. The is a very large amount of tanks with ~0 chl a, which might be confusing for the Pearson correlation (the data are not normally distributed). Maybe Spearman rank correlation would be more appropriate?

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)
cor3_8P<-corr.test(NUMZOOP,method="pearson",adjust="BH")
cor3_8P
```

```
## Call:corr.test(x = NUMZOOP, 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_8kendall<-corr.test(NUMZOOP,method="kendall",adjust="BH")
print(cor3_8kendall, digits=4,short=FALSE)
```

```
## Call:corr.test(x = NUMZOOP, method = "kendall", adjust = "BH")
## Correlation matrix
##          TP     TN     SRP    TIN    CHLA      ZP
## TP   1.0000 0.7391  0.3913 0.5771  0.0438  0.5362
## TN   0.7391 1.0000  0.4783 0.8094  0.0146  0.5507
## SRP  0.3913 0.4783  1.0000 0.5626 -0.0657  0.4493
## TIN  0.5771 0.8094  0.5626 1.0000  0.0439  0.5481
## CHLA 0.0438 0.0146 -0.0657 0.0439  1.0000 -0.0511
## ZP   0.5362 0.5507  0.4493 0.5481 -0.0511  1.0000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##          TP     TN    SRP    TIN   CHLA     ZP
## TP   0.0000 0.0003 0.0880 0.0139 0.8989 0.0148
## TN   0.0000 0.0000 0.0339 0.0000 0.9460 0.0139
## SRP  0.0586 0.0181 0.0000 0.0139 0.8989 0.0461
## TIN  0.0031 0.0000 0.0042 0.0000 0.8989 0.0139
## CHLA 0.8390 0.9460 0.7604 0.8387 0.0000 0.8989
## ZP   0.0069 0.0053 0.0276 0.0056 0.8126 0.0000
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##            lower       r  upper      p
## TP-TN      0.4784  0.7391 0.8801 0.0000
## TP-SRP    -0.0144  0.3913 0.6864 0.0586
## TP-TIN     0.2265  0.5771 0.7954 0.0031
## TP-CHLA   -0.3661  0.0438 0.4394 0.8390
## TP-ZP      0.1695  0.5362 0.7725 0.0069
## TN-SRP     0.0928  0.4783 0.7391 0.0181
## TN-TIN     0.6029  0.8094 0.9143 0.0000
## TN-CHLA   -0.3911  0.0146 0.4155 0.9460
## TN-ZP      0.1894  0.5507 0.7807 0.0053
## SRP-TIN    0.2060  0.5626 0.7873 0.0042
## SRP-CHLA  -0.4570 -0.0657 0.3469 0.7604
## SRP-ZP     0.0560  0.4493 0.7218 0.0276
## TIN-CHLA  -0.3660  0.0439 0.4395 0.8387
## TIN-ZP     0.1858  0.5481 0.7792 0.0056
## CHLA-ZP   -0.4453 -0.0511 0.3597 0.8126
```

```r
print(cor3_8P,digits=4,short=FALSE)
```

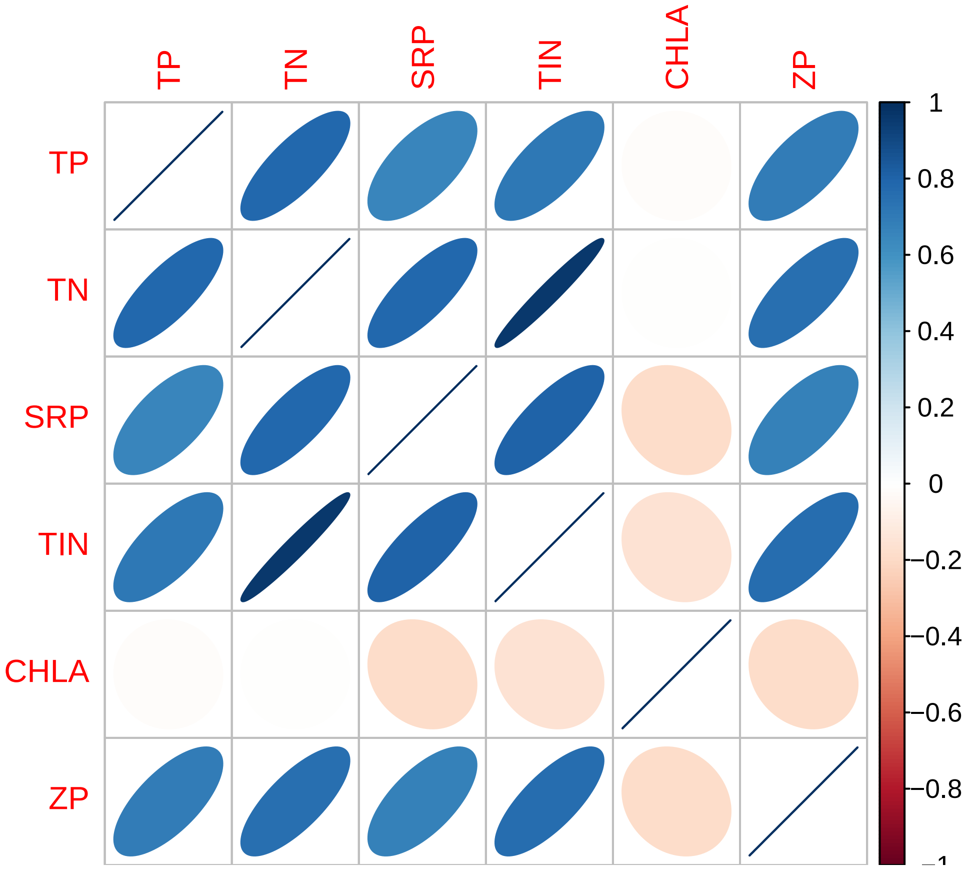
```
## Call:corr.test(x = NUMZOOP, method = "pearson", adjust = "BH")
## Correlation matrix
##           TP      TN     SRP     TIN    CHLA      ZP
## TP    1.0000  0.7865  0.6541  0.7171 -0.0167  0.6975
## TN    0.7865  1.0000  0.7842  0.9690 -0.0045  0.7562
## SRP   0.6541  0.7842  1.0000  0.8009 -0.1891  0.6763
## TIN   0.7171  0.9690  0.8009  1.0000 -0.1569  0.7606
## CHLA -0.0167 -0.0045 -0.1891 -0.1569  1.0000 -0.1826
## ZP    0.6975  0.7562  0.6763  0.7606 -0.1826  1.0000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##          TP     TN    SRP    TIN   CHLA     ZP
## TP   0.0000 0.0000 0.0008 0.0002 0.9835 0.0003
## TN   0.0000 0.0000 0.0000 0.0000 0.9835 0.0000
## SRP  0.0005 0.0000 0.0000 0.0000 0.4914 0.0005
## TIN  0.0001 0.0000 0.0000 0.0000 0.5355 0.0000
## CHLA 0.9384 0.9835 0.3761 0.4641 0.0000 0.4914
## ZP   0.0002 0.0000 0.0003 0.0000 0.3931 0.0000
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##            lower       r  upper      p
## TP-TN      0.5612  0.7865 0.9033 0.0000
## TP-SRP     0.3406  0.6541 0.8367 0.0005
## TP-TIN     0.4414  0.7171 0.8691 0.0001
## TP-CHLA   -0.4173 -0.0167 0.3894 0.9384
## TP-ZP      0.4092  0.6975 0.8591 0.0002
## TN-SRP    0.5570  0.7842 0.9022 0.0000
## TN-TIN    0.9286  0.9690 0.9867 0.0000
## TN-CHLA  -0.4071 -0.0045 0.3996 0.9835
## TN-ZP     0.5077  0.7562 0.8886 0.0000
## SRP-TIN   0.5872  0.8009 0.9102 0.0000
## SRP-CHLA -0.5505 -0.1891 0.2319 0.3761
## SRP-ZP    0.3753  0.6763 0.8483 0.0003
## TIN-CHLA -0.5269 -0.1569 0.2632 0.4641
## TIN-ZP    0.5152  0.7606 0.8907 0.0000
## CHLA-ZP  -0.5458 -0.1826 0.2384 0.3931
```

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***: 1> Not very sensitive. The results were not qualitatively different between the two types of tests. I did notice that P-values tended to be larger in the non-parametric test. IDK if my reasoning is ironclad, but it makes subjective sense bc I would expect the nonparametric test to have less power. . . 2> I would use the nonparametric methods for chl a corrlns, bc the chl a datapoints were positively skewed. 3> A little, but the FDR-adjusted P-values aren't much larger than unadjusted values. 4> It's useful when you're running a LOT of tests, e.g. evaluating SNP data such that you have ~1,000,000 variable sites and are testing for significance at each. You'd expect 50,000 false positive SNPs due to chance. So your chance of having ONE false positive is clearly NOT 5% once you're running multiple tests. So FDR more or less divides how many significant datapoints you expect-due-to-chance-alone by how many significant datapoints were observed. The MOST-significant datapoint doesn't get adjusted; the second-most signif value gets multiplied by (number_of_tests)/(number_of_tests - 1). If it is still < 0.05, then it is stiff signif after the FDR adjustment. <– or something like that.

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

```
library(corrplot)
corrplot(cor3_8,method="ellipse")
```



Linear Regression

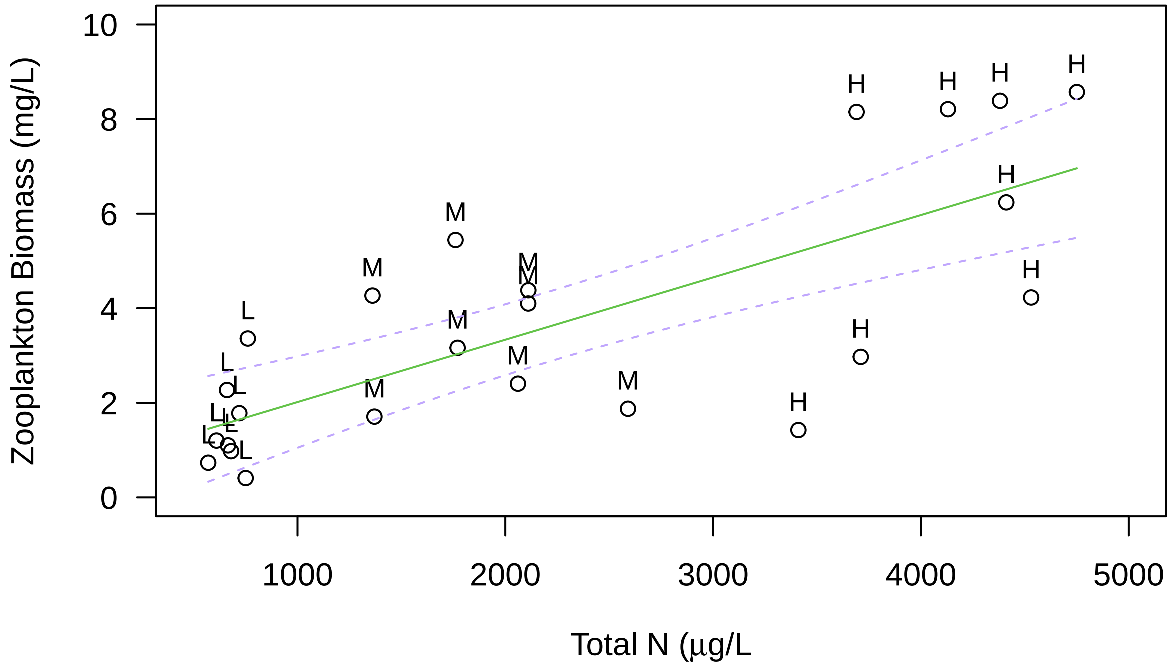
In the R code chunk below, do the following: 1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zooplankton biomass (ZP). 2) Examine the output of the regression analysis. 3) Produce a plot of this regression analysis including the following: categorically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

```
fit8_4<-lm(ZP ~ TN,data=ROYZOOP)
fit8_4
```

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

```
plot(ROYZOOP$TN, ROYZOOP$ZP, ylim=c(0,10), xlim=c(500,5000),xlab=expression(paste("Total N (", mu, "g/L
text(ROYZOOP$TN,ROYZOOP$ZP,ROYZOOP$NUTS, pos=3,cex=0.8)

endsTN<-seq(min(ROYZOOP$TN),max(ROYZOOP$TN),10)

regline<-predict(fit8_4,newdata=data.frame(TN=endsTN))
lines(endsTN,regline,col="#63c349")
```

```
conf95<-predict(fit8_4,newdata=data.frame(TN = endsTN),interval=c("confidence"),level=0.95,type="respons
matlines(endsTN,conf95[,c("lwr","upr")],lty=2,lwd=1,col="#c0a6fd")
```



Question 6: Interpret the results from the regression model

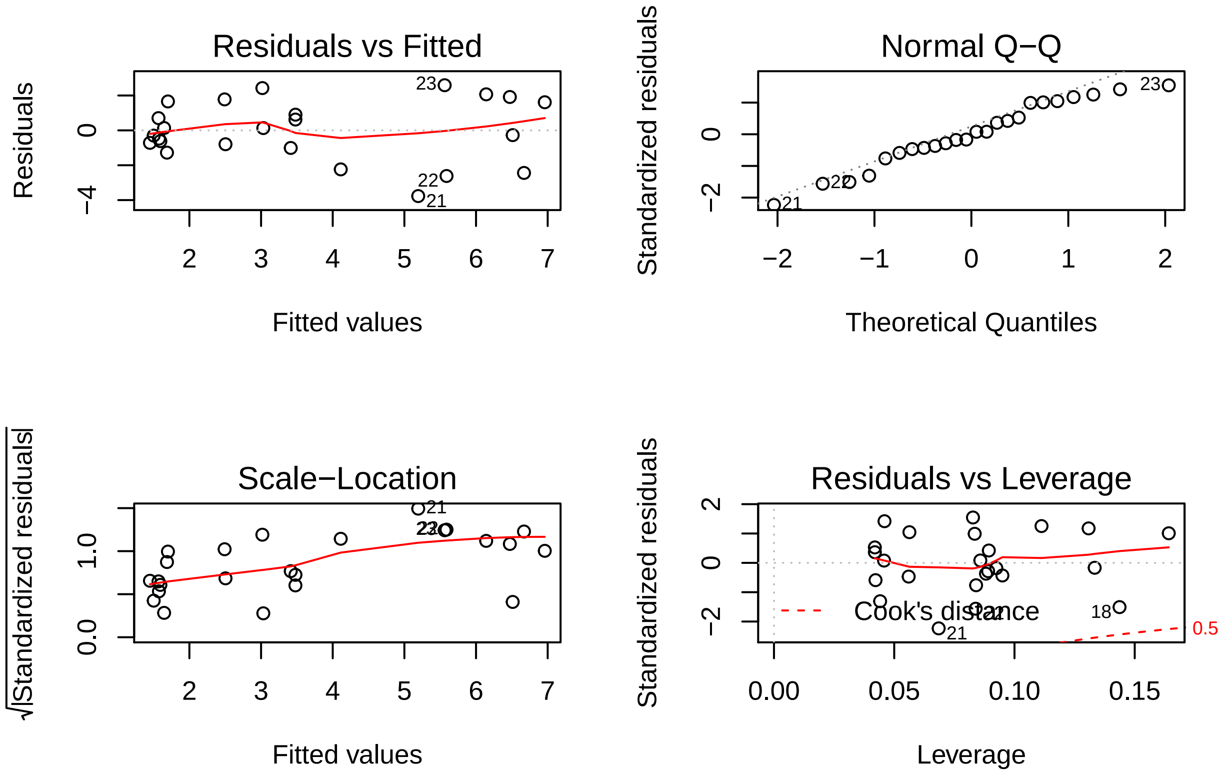
> ***Answer 6***: As [N] increases, [zooplankton] increases also. for each unit increase in [N], [zooplankton] increases by about the same amount (1:1 relationship) However, I don't think it is a perfect model. MANY datapoints fall outside the CIs. It would be useful to calculate an R-square value…

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

> ***Answer 7***: I think it uses the y=mx+b equation of the fit8_4 regression to generate y-values from x-values

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(fit8_4)
```

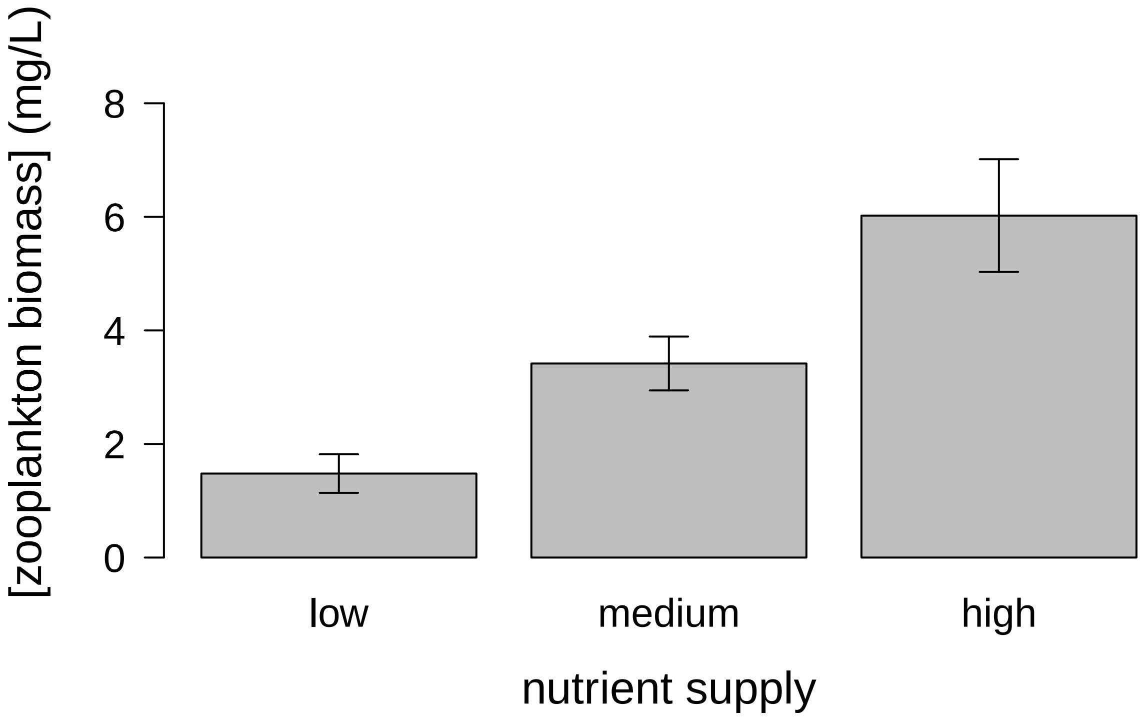


- Upper left: is there a random distribution of the residuals around zero (horizontal line)?
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)`
- Bottom left: again, looking for a random distribution of sqrt(standardized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

No, I don't think the data meet the assumptions. There is a systematic deviation from normality in the QQ plot. The scale-location is also systematically weird. And I am still not sure I understand how to interpret the leverage plot—right now it seems to indicate that not of the points are "weird" because they're all closer to the horizontal axis than the 0.5 Cook's contour… so maybe the data are normal? ### 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.

```
nutr<-factor(ROYZOOP$NUTS,levels=c("L","M","H"))
zp.means<-tapply(ROYZOOP$ZP,nutr,mean)
#wow, that's really useful
SEm<-function(a){
  sd(na.omit(a))/sqrt(length(na.omit(a)))
}
zp.sem<-tapply(ROYZOOP$ZP,nutr,SEm)
zpbp<-barplot(zp.means,ylim=c(0,round(max(ROYZOOP$ZP),digits=0)),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.
arrows(x0=zpbp,y0=zp.means,y1=zp.means-zp.sem,angle=90,length=0.1,lwd=1)
arrows(x0=zpbp,y0=zp.means,y1=zp.means+zp.sem,angle=90,length=0.1,lwd=1)
```



```
nutrANOVA<-aov(ZP~NUTS,data=ROYZOOP)
summary(nutrANOVA)
```

```
##             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(nutrANOVA)
```

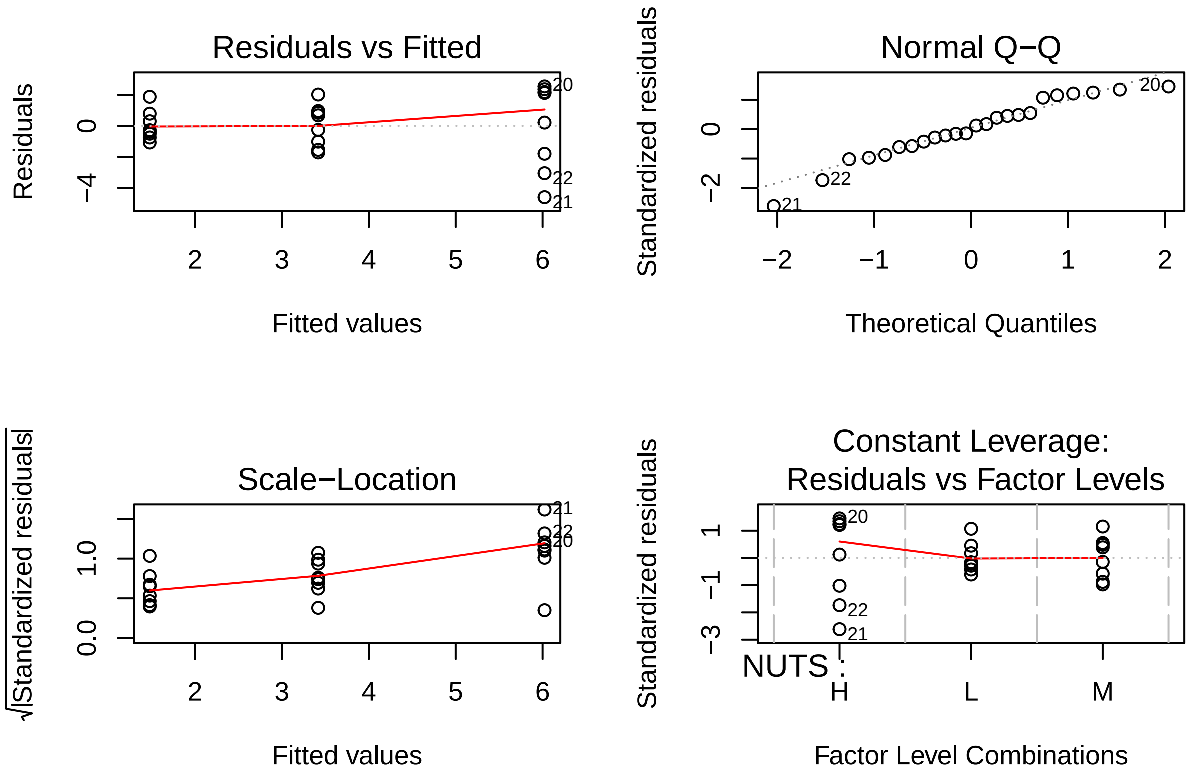
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = ZP ~ NUTS, data = ROYZOOP)
## 
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## 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: Okay, now we're doing a categorical analysis. Using chosen breakpoints, can we detect categories of nutrient amt that are associated with different [zooplankton]? Yes, there is a difference between the high and low nutr treatments. There is a difference between high and medium nutrient treatments. We could not detect a difference in [zooplankton] between medium and low nutr trmtnts. Concerns: Welp, the residuals look a bit better than for the regression, but not perfect. More concerning to me is that the L, M, H cutoffs are arbitrary. It is possible that the scientists had good justification for selecting the cutoffs that they did. Perhaps this was done by looking for "natural breaks" in the dataset. Still, I would prefer my categories to be @truly@ @categorical@.

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(nutrANOVA)
```



```
#as I mentioned above, the resids look a little better than for the rgrssn
```

SYNTHESIS: SITE-BY-SPECIES MATRIX

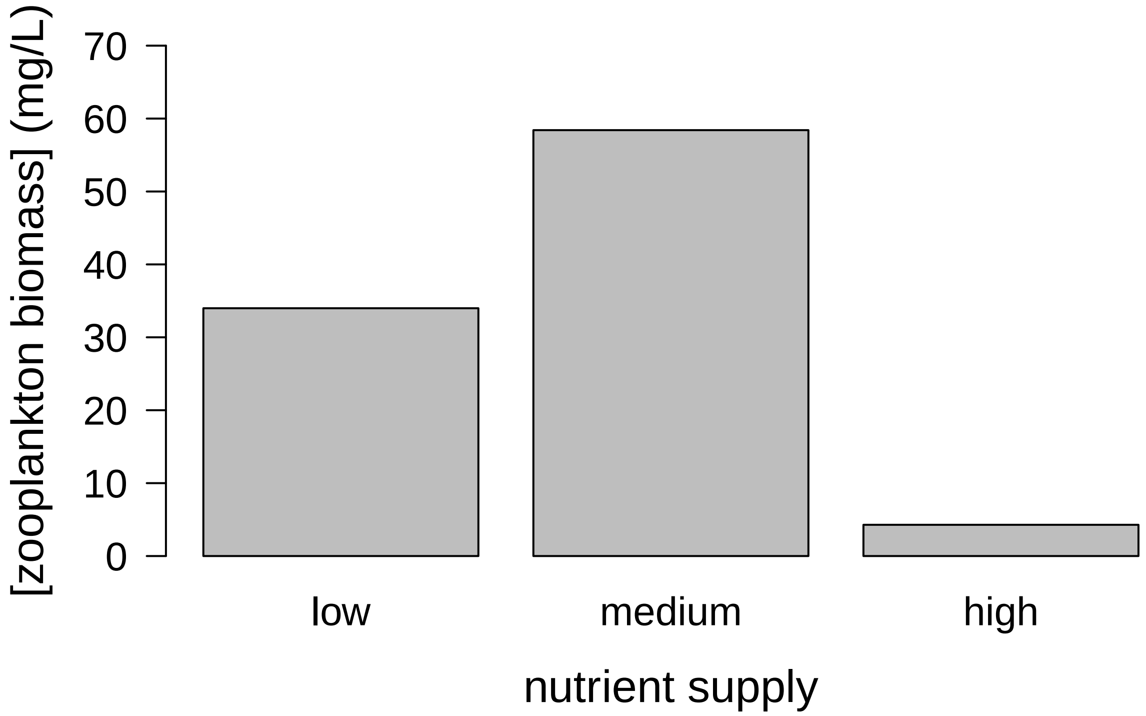
In the R code chunk below, load the zoop.txt dataset in your Week1 data folder. Create a site-by-species matrix (or dataframe) that does not include TANK or NUTS. The remaining columns of data refer to the biomass (Âµg/L) of different zooplankton taxa:

- CAL = calanoid copepods
- DIAP = *Diaphanasoma* sp.
- CYL = cyclopoid copepods

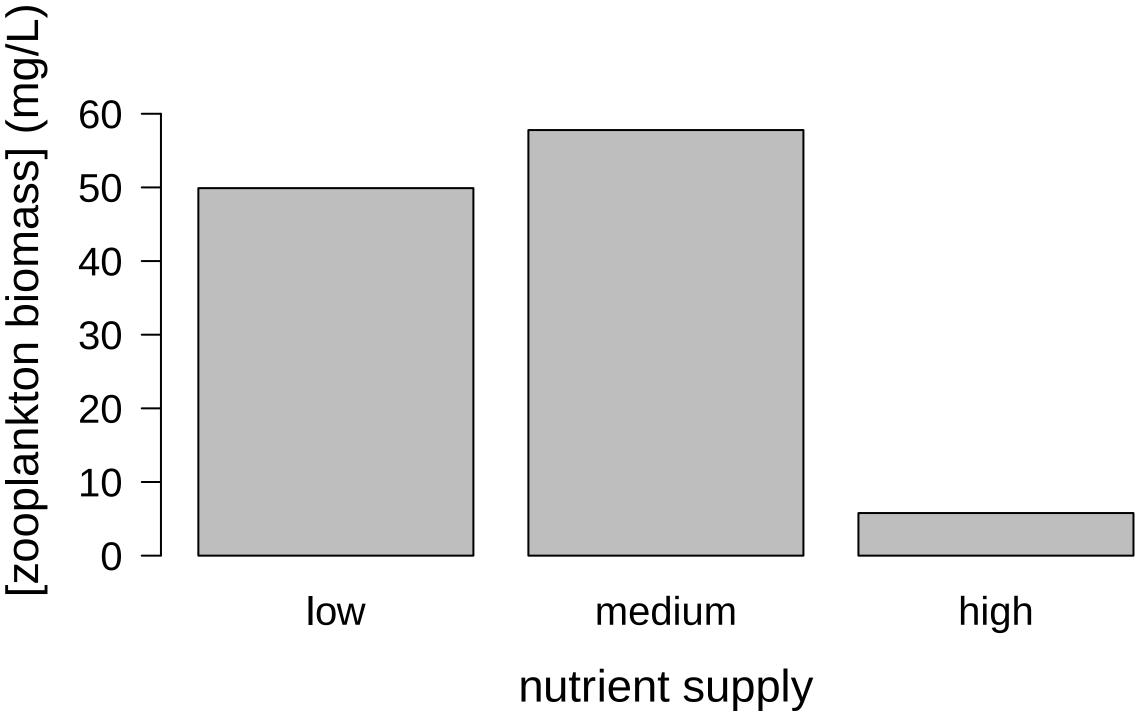
- BOSM = *Bosmina* sp.
- SIMO = *Simocephallus* sp.
- CERI = *Ceriodaphnia* sp.
- NAUP = naupuli (immature copepod)
- DLUM = *Daphnia lumholtzi*
- CHYD = *Chydorus* sp.

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.

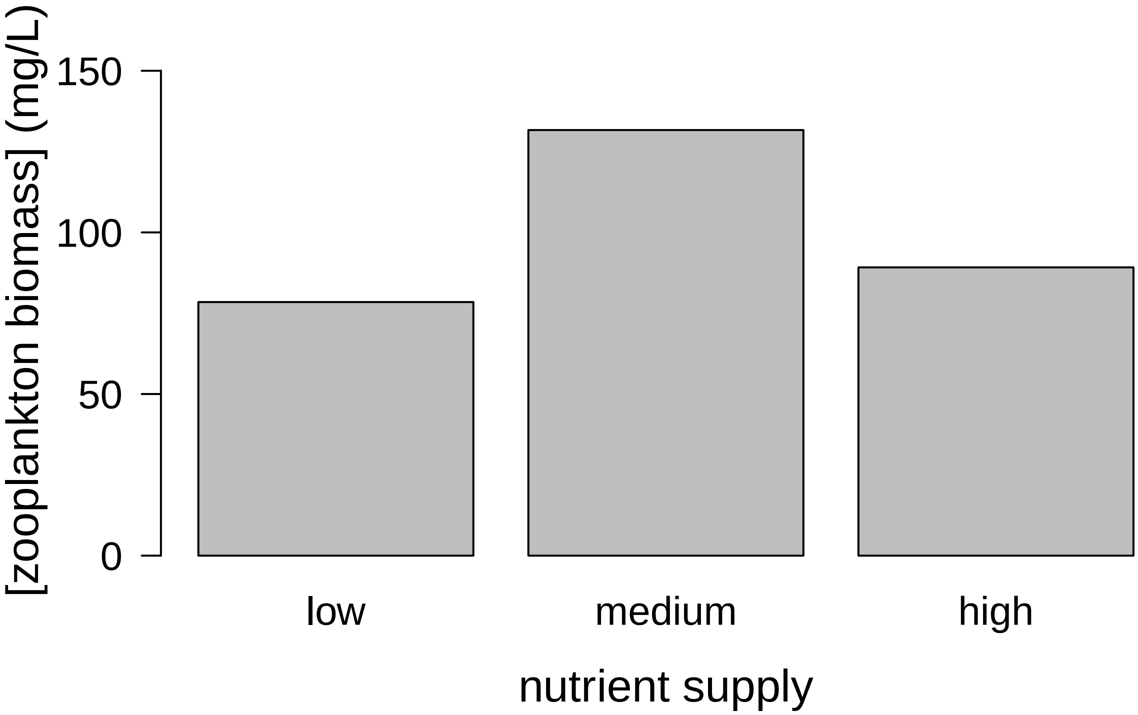
```
#m9<-as.matrix(read.table("data\\zoops.txt",sep="\t",header=TRUE))

m9<-(read.table("data\\zoops.txt",sep="\t",header=TRUE))
m92<-factor(m9$NUTS,levels=c("L","M","H"))
m9num<-m9[,3:11]
zp.means.vect<-c(seq(1,1,1))
#for (i in seq(3,11,1)){
#  zp.means.vect[i]<-tapply(m9[,i],m9[,2],mean)
#}
#for (i in names(m9)){
#  append(zp.means.vect, tapply(m9$i,m9$NUTS,mean))
#}
#for (i in seq(1,9,1)){
#  temp<-tapply(m9[,i+2],m9[,2],mean)
#  print(temp)
#  append(zp.means.vect, c(temp))
#}

zpsp3<-tapply(m9[,3],m92,mean)
zpsp4<-tapply(m9[,4],m92,mean)
zpsp5<-tapply(m9[,5],m92,mean)
zpsp6<-tapply(m9[,6],m92,mean)
zpsp7<-tapply(m9[,7],m92,mean)
zpsp8<-tapply(m9[,8],m92,mean)
zpsp9<-tapply(m9[,9],m92,mean)
zpsp10<-tapply(m9[,10],m92,mean)
zpsp11<-tapply(m9[,11],m92,mean)
zpbp3<-barplot(zpsp3,ylim=c(0,(max(zpsp3*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



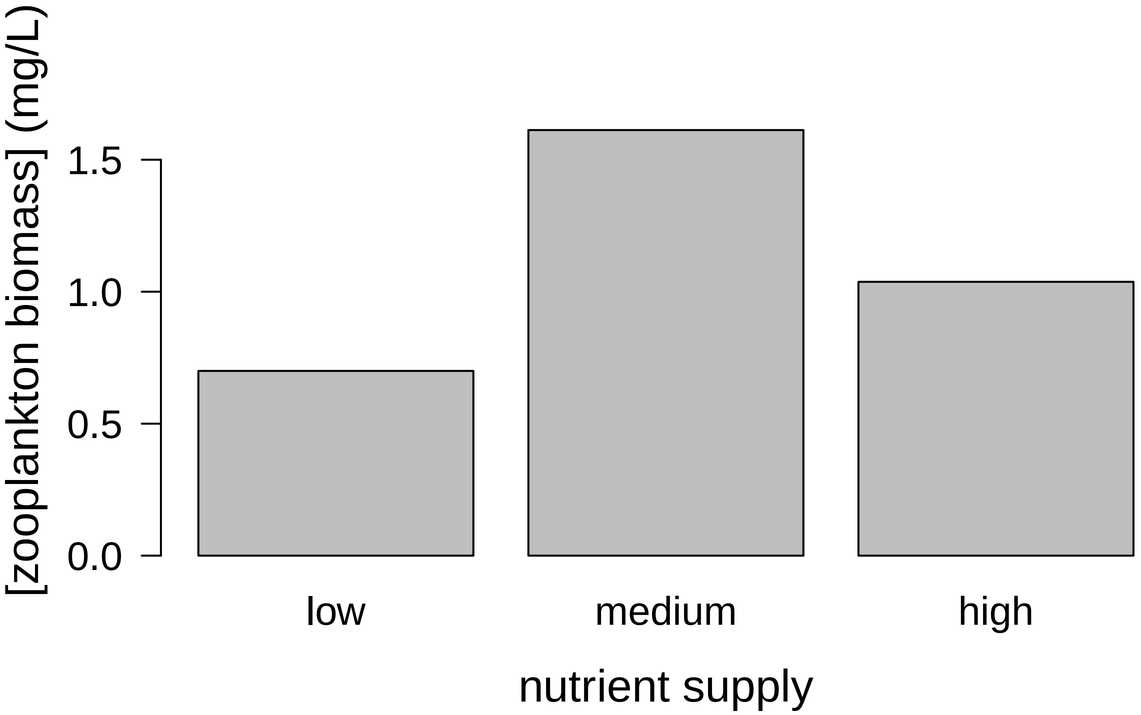
```
zpbp4<-barplot(zpsp4,ylim=c(0,(max(zpsp4*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



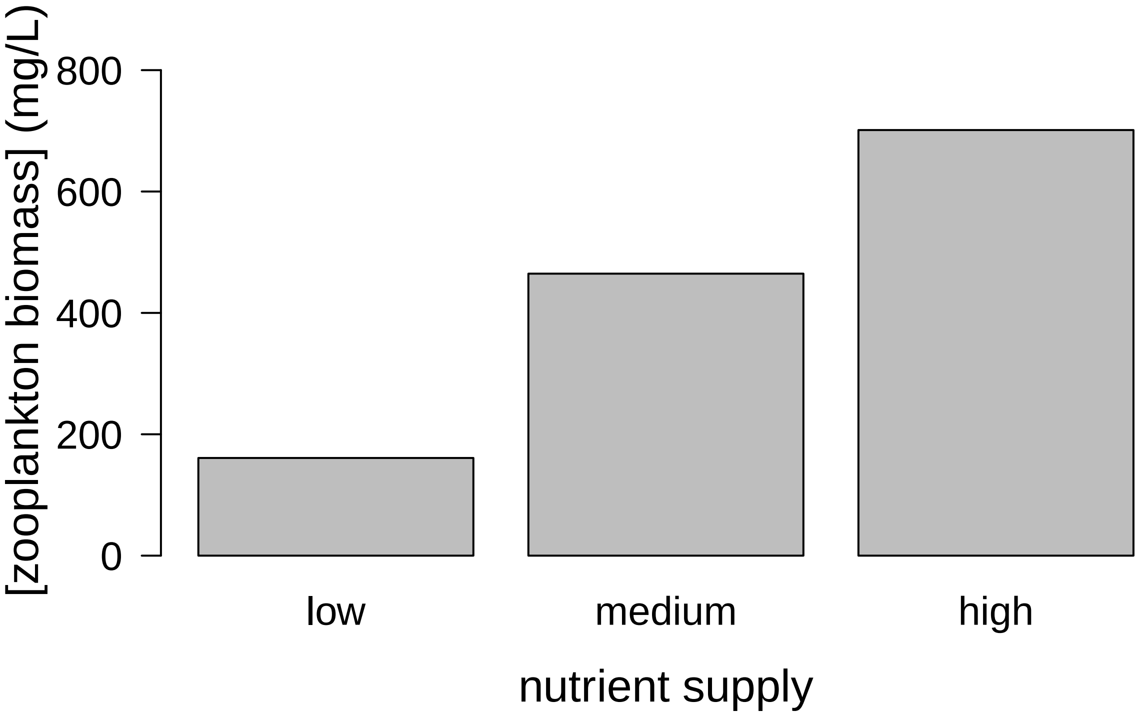
```
zpbp5<-barplot(zpsp5,ylim=c(0,(max(zpsp5*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



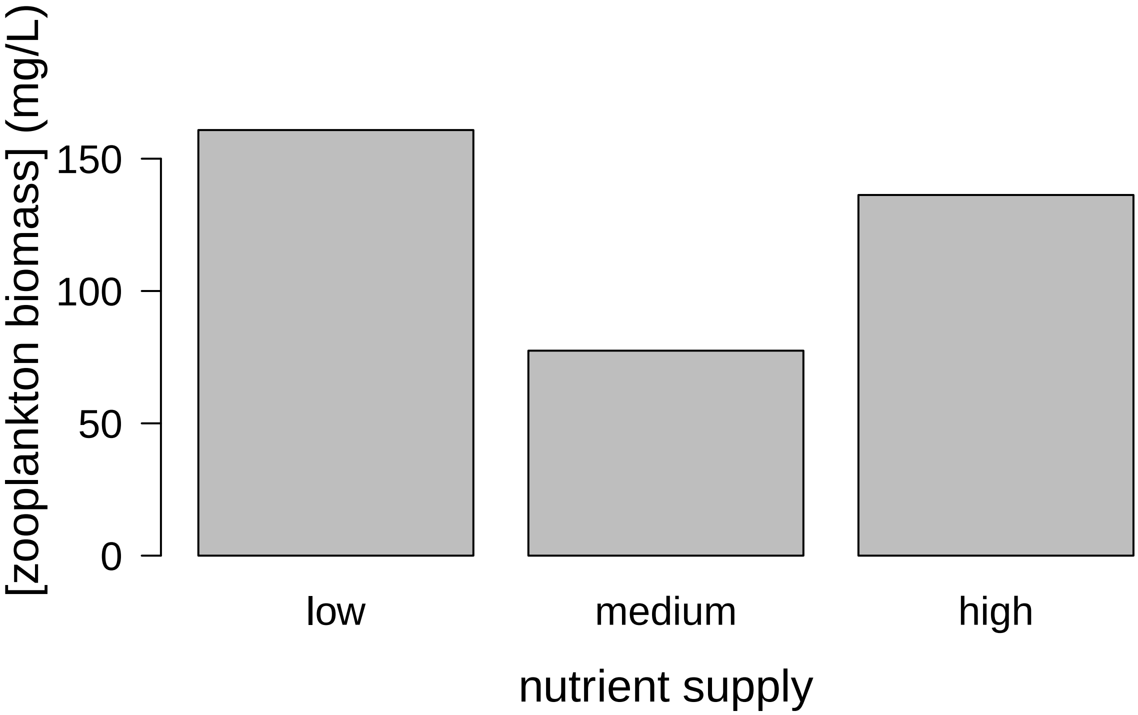
```
zpbp6<-barplot(zpsp6,ylim=c(0,(max(zpsp6*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



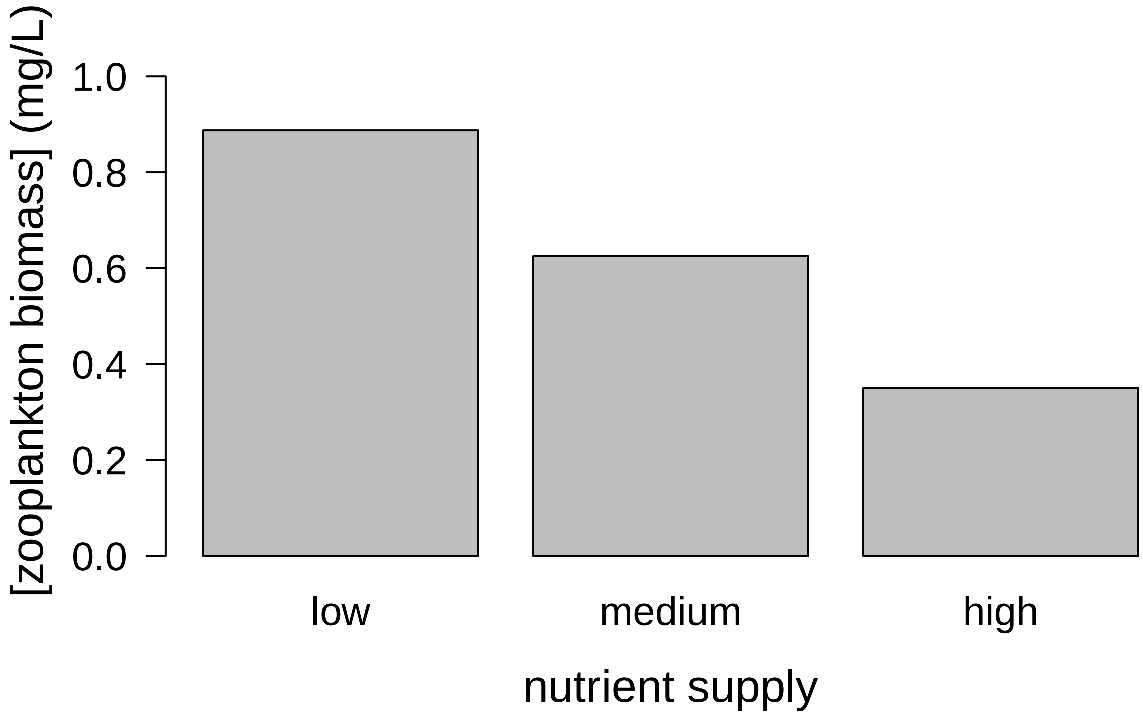
```
zpbp7<-barplot(zpsp7,ylim=c(0,(max(zpsp7*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



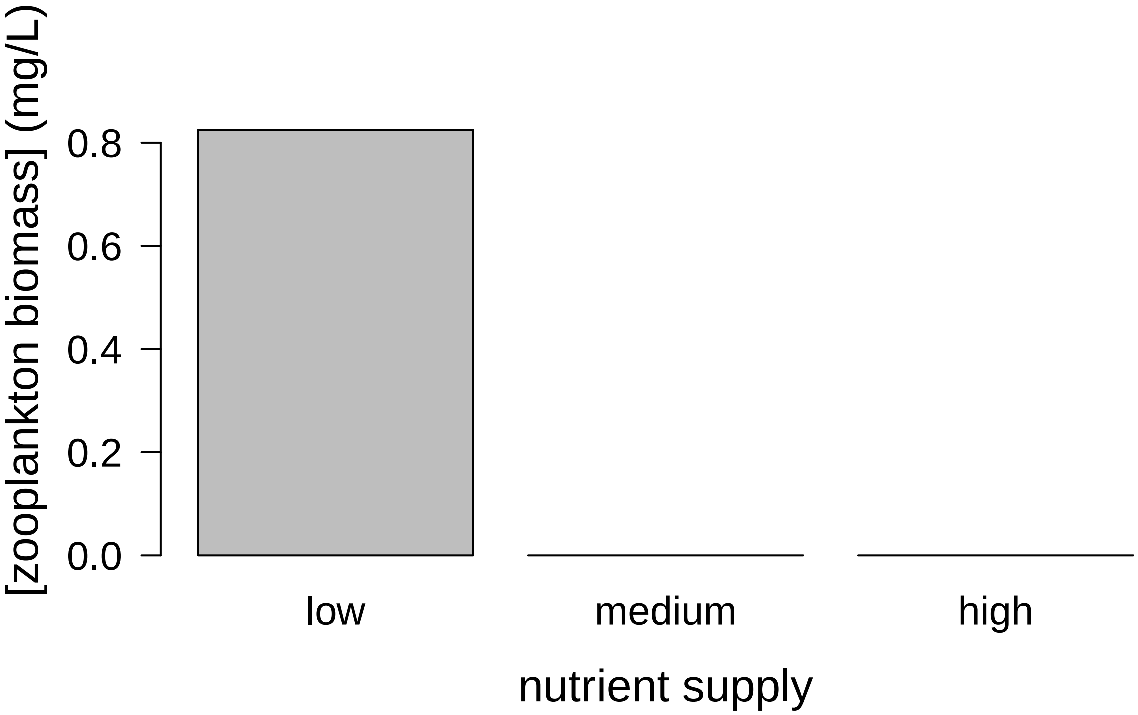
```
zpbp8<-barplot(zpsp8,ylim=c(0,(max(zpsp8*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



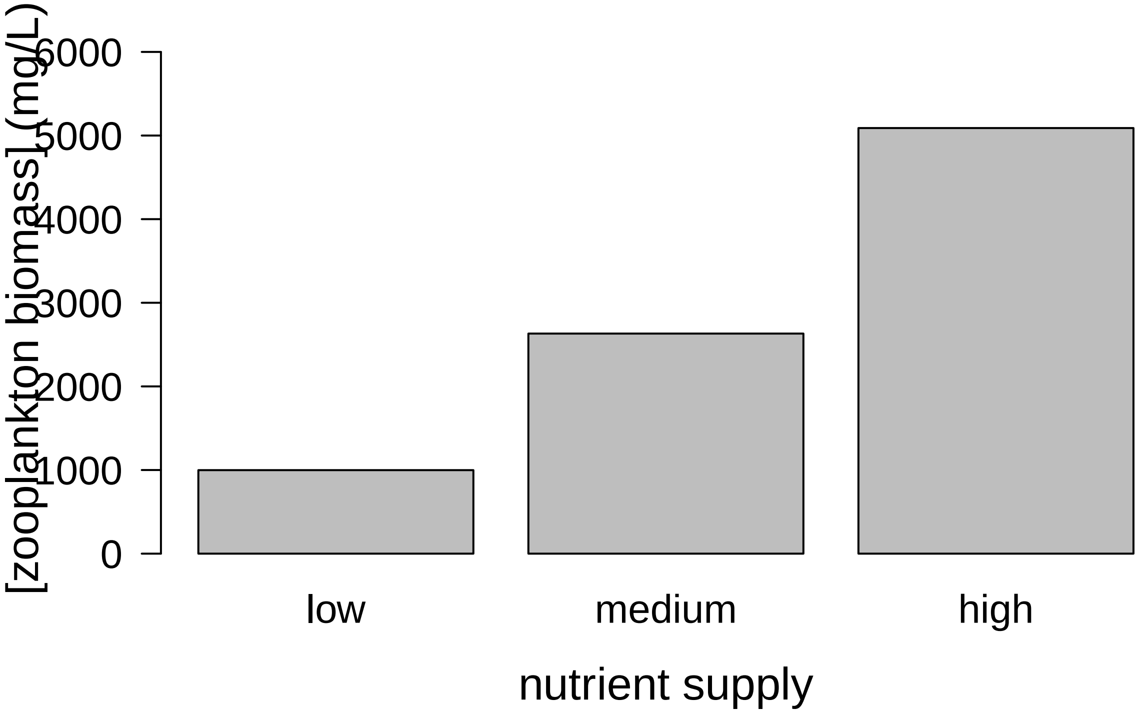
```
zpbp9<-barplot(zpsp9,ylim=c(0,(max(zpsp9*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab="nu
```



```
zpbp10<-barplot(zpsp10,ylim=c(0,(max(zpsp10*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab
```



```
zpbp11<-barplot(zpsp11,ylim=c(0,(max(zpsp11*1.2))),pch=15,cex=1.25,las=1,cex.lab=1.4,cex.axis=1.25,xlab=
```



> I learned that for loops in R make me cry. > I learned that I do not know how to append to a vector in R and I still couldn't figure it out after a couple hours on stackoverflow. > I learned that brute force works with only 9 taxa. > [CAL] didn't have a clear pattern. > Similar pattern for DIAP. in both these clades the total [] isn't very high, <100 mg/L > For CYCL there was no clear visually-detectable [zoop] response to [N] > Similarly for BOSM; moreover, its extremely low []s implies that it did not affect the overall pattern very much > SIMO the expected trend, with lowest [zoop] for low [N], and high for high. Too there is are up to ~700 mg/L SIMO, so this guy is substantive contributor to the obsvd pattern. > CERI also fits the expectation pattern, though its []s are ~25% of SIMO's > [NAUP] is too low for me to care > ditto for DLUM > [CHYD]s are as much as 5000 mg/L, so it should be driving the overall pattern we saw in the correlation. Its []s do indeed corroborate the pattern, a positive correlation.

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)**.