Week 1 Assignment: Basic R

Alison Partee; Z620: Quantitative Biodiversity, Indiana University

17 January, 2017

OVERVIEW

Week 1 Assignment introduces some of the basic features of the R computing environment (http://www.r-project.org). It is designed to be used along side your Week 1 Handout (hard copy). You will not be able to complete the exercise if you do not have your handout.

Directions:

1. Change "Student Name" on line 3 (above) with your name.
2. Complete as much of the assignment as possible during class; what you do not complete in class will need to be done on your own outside of class.
3. Use the handout as a guide; it contains a more complete description of data sets along with examples of proper scripting needed to carry out the exercise.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. Before you leave the classroom today, it is *imperative* that you **push** this file to your GitHub repo.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file. Basically, just press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Week1 folder.
7. After Knitting, please submit the completed exercise by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (*Week1_Assignment.Rmd*; with all code blocks filled out and questions answered) and the PDF output of `Knitr` (*Week1_Assignment.pdf*).

The completed exercise is due on **Wednesday, January 18$^{\text{th}}$, 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] "/Users/flopsei/GitHub/QB2017_Partee/Week1"
```

```r
setwd("/Users/flopsei/GitHub/QB2017_Partee/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.

```r
# volume of a cube
l <- 5
cubevol <- l^3

# area of a circle
r <- 2
cirarea <- pi * r^2

# length of opposite side of right triangle
theta <- pi/4
hypot <- sqrt(2)
oppside <- hypot * sin(theta)

# natural log of your fav number
favnum <- 2.72
natlog <- log(favnum)
```

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
x <- c(4, 28, 3, -9, 1)
w <- x*14
(x+w)/15
```

```
## [1]  4 28  3 -9  1
```

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, 0, 5, 9, 4)
k*x
```

```
## [1]   4   0  15 -81   4
```

```
d <- c(w[1],w[3],w[4],k[2],k[3],k[1],k[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 of v
vmax <- max(v)

#min of v
vmin <- min(v)

#sum of v
vsum <- sum(v)

#mean of v
vmean <-mean(v)

#median of v
vmed <- median(v)

#variance of v
vvar <- var(v)

#standard deviation of v
vsd <- sd(v)

#standard error of the mean for v
v_sterror <- sd(v)/sqrt(length(v))
```

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
x <- c(rnorm(5,mean = 8, sd = 2))
y <- c(rnorm(5, mean = 25, sd = 10))

xy <- cbind(x,y)
```

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 first argument in the rnorm function specifies the length of the vector you are creating. The second parameter we use, mean, specifies the mean of the values within the vector. The second parameter we're using, sd, specifies the standard deviation of the values within our vector.

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.

```r
m <- as.matrix(read.table("data/matrix.txt", sep = "\t", header = FALSE))
mtransposed <- t(m)
mtdim <- dim(mtransposed)
```

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

> Answer 2: 5 rows by 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`.

```r
#m without the third column
mno3c <- m[, c(1:2,4:dim(m)[2])]

# m without the last row
mnolastr <- m[1:dim(m)[1]-1,]
```

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

> ***Answer 3***:
>
> To index matrix m by selecting all but the third column, I indexed matrix m by all rows and selected the columns by creating a vector with the numbers of the columns I want. In this case, the columns I want are 1,2,and then columns 4 until the last column.
>
> To return a copy of matrix m without the last row, I index matrix m by selecting all but the last row and all columns. This returns a copy of matrix m without the last row.

6) BASIC DATA VISUALIZATION AND STATISTICAL ANALYSIS

Load Zooplankton Dataset

In the R code chunk below, do the following: 1) Load the zooplankton dataset from the Week1 data folder. 2) Display the structure of this data set.

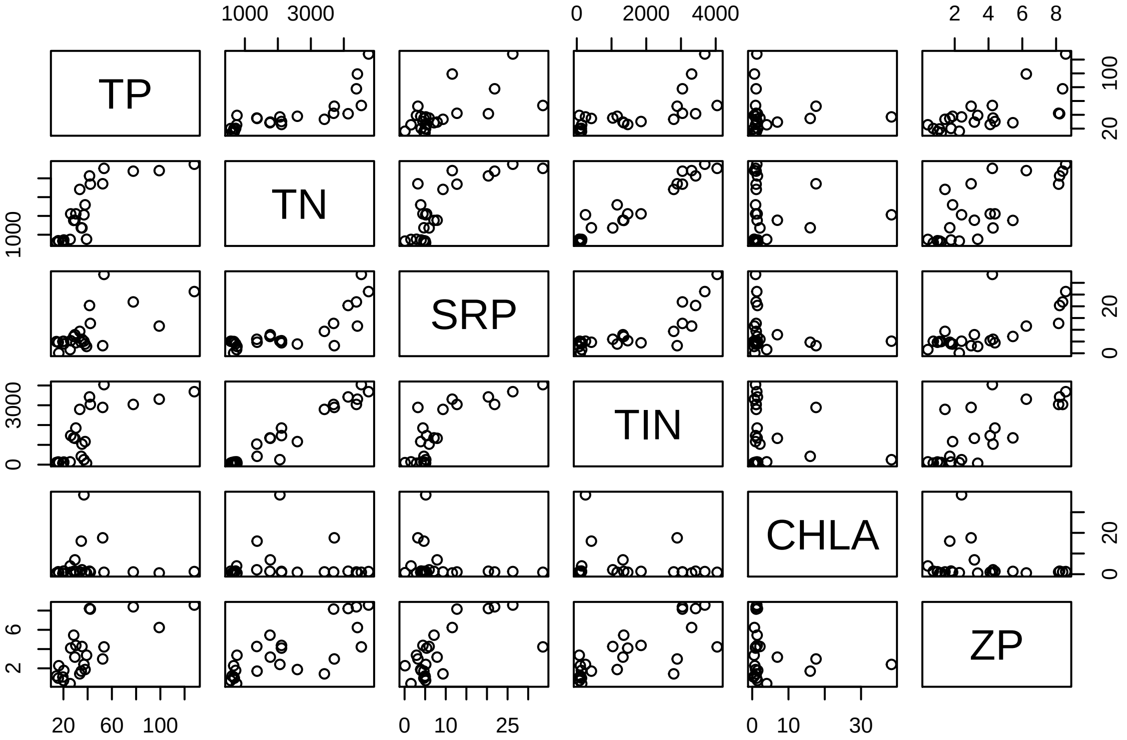
```r
#load the dataset
meso <- read.table("data/zoop_nuts.txt", sep = "\t", header = TRUE)

#display the structure of the dataset
str(meso)
```

```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: Factor w/ 3 levels "H","L","M": 2 2 2 2 2 2 2 2 3 3 ...
##  $ TP  : num  20.3 25.6 14.2 39.1 20.1 ...
##  $ TN  : num  720 750 610 761 570 ...
##  $ SRP : num  4.02 1.56 4.97 2.89 5.11 4.68 5 0.1 7.9 3.92 ...
##  $ TIN : num  131.6 141.1 107.7 71.3 80.4 ...
##  $ CHLA: num  1.52 4 0.61 0.53 1.44 1.19 0.37 0.72 6.93 0.94 ...
##  $ ZP  : num  1.781 0.409 1.201 3.36 0.733 ...
```

Correlation

In the R code chunk below, do the following: 1) Create a matrix with the numerical data in the `meso` dataframe. 2) Visualize the pairwise **bi-plots** of the six numerical variables. 3) Conduct a simple **Pearson's correlation** analysis.

```r
#create a matrix with the numerical meso data
meso.num <- meso[,3:dim(meso)[2]]

#visualize the pairwise bi-plots of the six numerical variables
pairs(meso.num)
```



```
#conduct a simple Pearson's correlation analysis
cor1 <- cor(meso.num)
```

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

> Answer 4: The visualization above shows small plots of each variable combination. The correlation analysis gives correlation coefficients for each respecive plot.

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 the psych package
# install.packages("psych", repos="http://cran.rstudio.com/")

# load the psych package
require("psych")
```

```
## Loading required package: psych

## Warning: package 'psych' was built under R version 3.2.5
```

```
# redo the correlation alalysis using the corr.test() function in the psych package
# with method = "pearson" and adjust = "BH"
cor2 <- corr.test(meso.num, method = "pearson", adjust = "BH")

# redo the correlation analusis using a non-parametric method
```

```r
cor3 <- corr.test(meso.num, method = "kendall", adjust = "BH")

# use the print command to see the correlation results
print(cor2, digits = 3)
```

```
## Call:corr.test(x = meso.num, method = "pearson", adjust = "BH")
## Correlation matrix
##          TP     TN    SRP    TIN   CHLA     ZP
## TP    1.000  0.787  0.654  0.717 -0.017  0.697
## TN    0.787  1.000  0.784  0.969 -0.004  0.756
## SRP   0.654  0.784  1.000  0.801 -0.189  0.676
## TIN   0.717  0.969  0.801  1.000 -0.157  0.761
## CHLA -0.017 -0.004 -0.189 -0.157  1.000 -0.183
## ZP    0.697  0.756  0.676  0.761 -0.183  1.000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         TP    TN   SRP   TIN  CHLA    ZP
## TP   0.000 0.000 0.001 0.000 0.983 0.000
## TN   0.000 0.000 0.000 0.000 0.983 0.000
## SRP  0.001 0.000 0.000 0.000 0.491 0.000
## TIN  0.000 0.000 0.000 0.000 0.536 0.000
## CHLA 0.938 0.983 0.376 0.464 0.000 0.491
## ZP   0.000 0.000 0.000 0.000 0.393 0.000
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```r
print(cor3, digits = 3)
```

```
## Call:corr.test(x = meso.num, method = "kendall", adjust = "BH")
## Correlation matrix
##         TP    TN    SRP   TIN   CHLA     ZP
## TP   1.000 0.739  0.391 0.577  0.044  0.536
## TN   0.739 1.000  0.478 0.809  0.015  0.551
## SRP  0.391 0.478  1.000 0.563 -0.066  0.449
## TIN  0.577 0.809  0.563 1.000  0.044  0.548
## CHLA 0.044 0.015 -0.066 0.044  1.000 -0.051
## ZP   0.536 0.551  0.449 0.548 -0.051  1.000
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         TP    TN   SRP   TIN  CHLA    ZP
## TP   0.000 0.000 0.088 0.014 0.899 0.015
## TN   0.000 0.000 0.034 0.000 0.946 0.014
## SRP  0.059 0.018 0.000 0.014 0.899 0.046
## TIN  0.003 0.000 0.004 0.000 0.899 0.014
## CHLA 0.839 0.946 0.760 0.839 0.000 0.899
## ZP   0.007 0.005 0.028 0.006 0.813 0.000
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Question 5: Describe what you learned from `corr.test`. Describe what you learned from corr.test. Specifically, are the results sensitive to whether you use parametric (i.e., Pearson's) or non-parametric methods? When should one use non-parametric methods instead of parametric methods? With the Pearson's method, is there evidence for false discovery rate due to multiple comparisons? Why is false discovery rate important?

> ***Answer 5***: Yes, the corr.test values were different depending on whether a parametric parameter was given or a non-parametric parameter. Non-parametric methods should be used when sample sizes are small. Since the values in the upper half of the matrix above the diagonal differ from their respective values in the lower triangular part of the matrix, there is evidence for false discovery rate due to multiple comparisons. The false discovery rate is important because it detects the proportion of type I errors when making multiple comparisons.

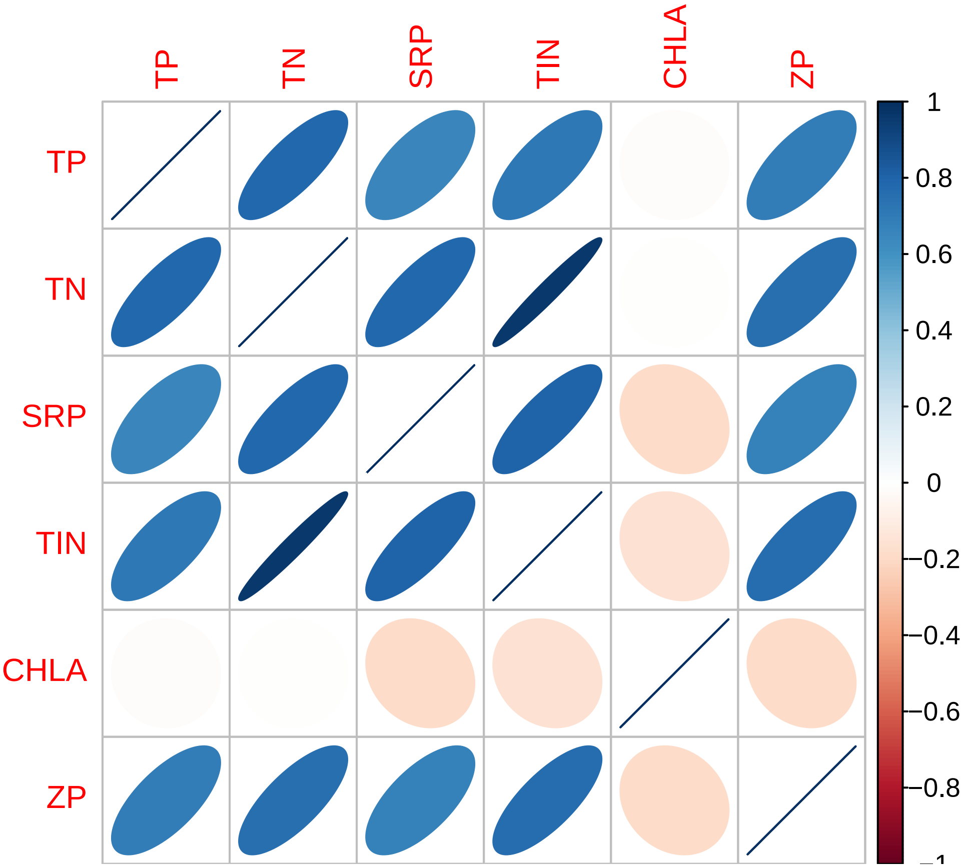
In the R code chunk below, use the `corrplot` function in the *corrplot* package to produce the ellipse correlation plot in the handout.

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

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

```
## Warning: package 'corrplot' was built under R version 3.2.5
```

```
#produce the ellipse correlation plot
corrplot(cor1, method = "ellipse")
```



```r
dev.off()
```

```
## null device
##           1
```

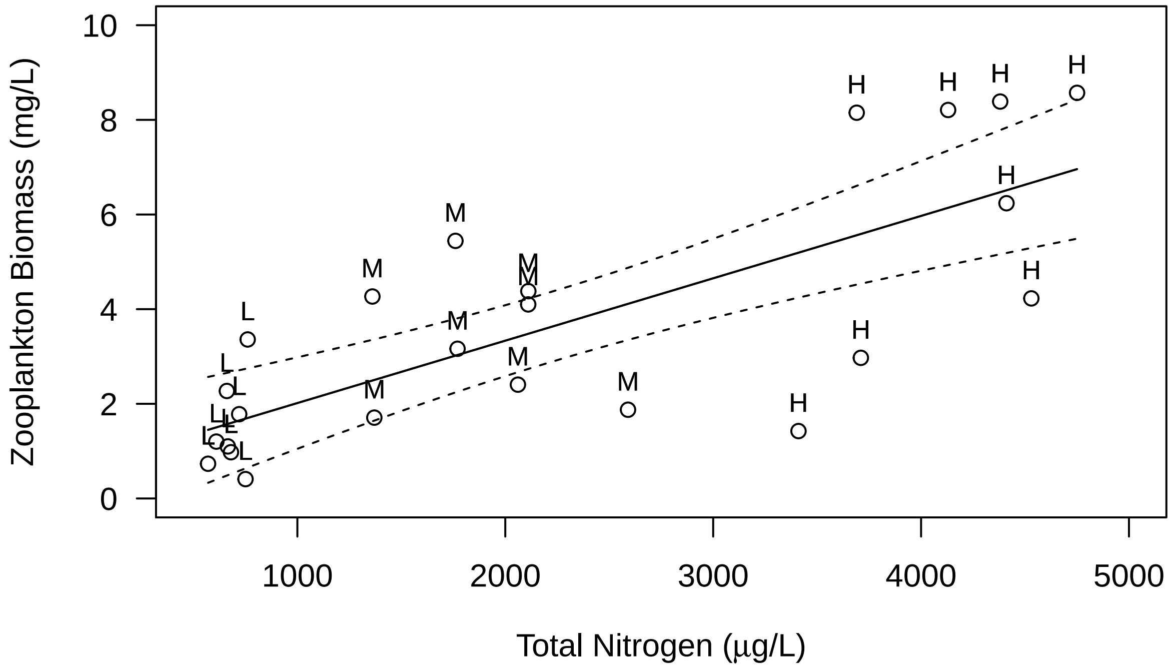
Linear Regression

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.

```r
fitreg <- lm(ZP ~ TN, data = meso)

summary(fitreg)
```

```
##
## Call:
## lm(formula = ZP ~ TN, data = meso)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7690 -0.8491 -0.0709  1.6238  2.5888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6977712  0.6496312   1.074    0.294
## TN          0.0013181  0.0002431   5.421 1.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.75 on 22 degrees of freedom
## Multiple R-squared:  0.5719, Adjusted R-squared:  0.5525
## F-statistic: 29.39 on 1 and 22 DF,  p-value: 1.911e-05
```

```r
plot(meso$TN, meso$ZP, ylim = c(0,10), xlim = c(500,5000), xlab = expression(paste("Total Nitrogen (", 
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)

#create and plot the confidence intervals
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, t
matlines(newTN, conf95[, c("lwr", "upr")], type="l", lty = 2, lwd = 1, col = "black")
```



Question 6: Interpret the results from the regression model

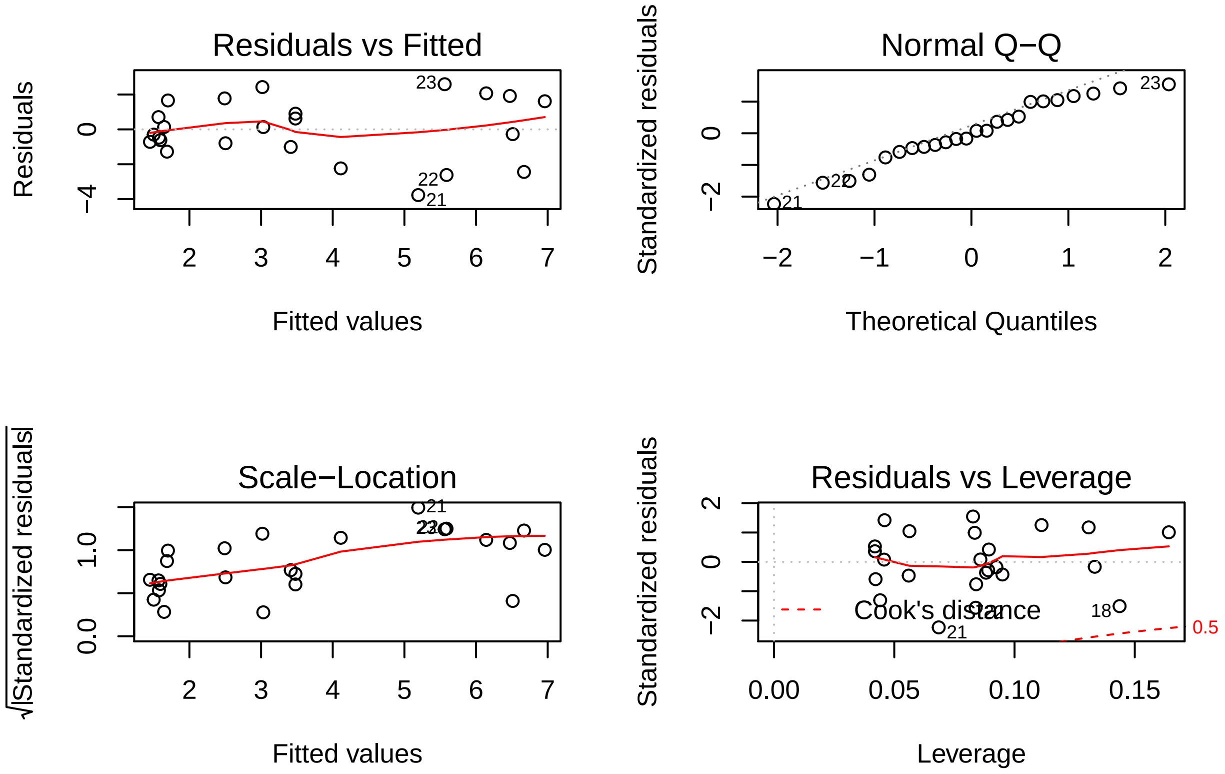
> ***Answer 6***: Total nitrogen and zooplankton biomass are strongly positively correlated in this data set.

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

> ***Answer 7***: The predict() function is predicting the values for our confidence intervals using the model we give it and a given range of 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.

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



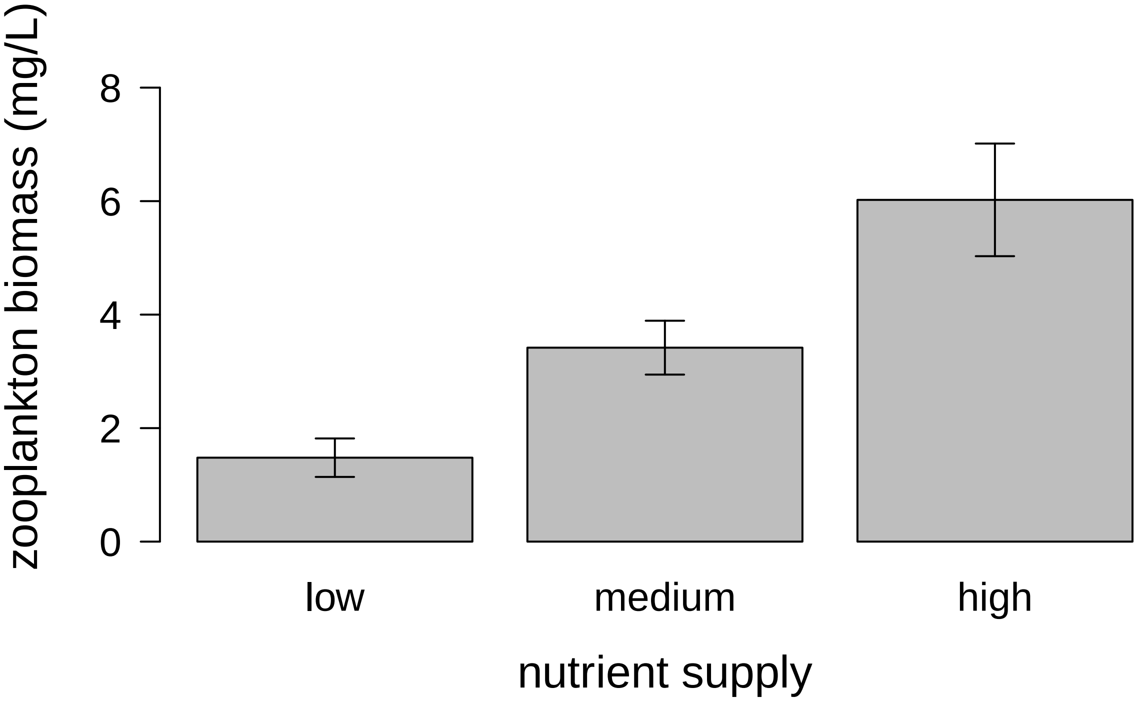
- Upper left: is there a random distribution of the residuals around zero (horizontal line)?
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)`
- Bottom left: again, looking for a random distribution of sqrt(standardized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

Analysis of Variance (ANOVA)

Using the R code chunk below, do the following: 1) Order the nutrient treatments from low to high (see handout). 2) Produce a barplot to visualize zooplankton biomass in each nutrient treatment. 3) Include error bars (+/- 1 sem) on your plot and label the axes appropriately. 4) Use a one-way analysis of variance (ANOVA) to test the null hypothesis that zooplankton biomass is affected by the nutrient treatment. 5) Use a Tukey's HSD to identify which treatments are different.

```
# order the treatments from low to high
NUTS <- factor(meso$NUTS, levels = c('L', 'M', 'H'))

# calculate the means and standard errors for zooplankton biomass
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)

# make the bar plot
```

```r
bp <- barplot(zp.means, ylim = c(0, round(max(meso$ZP), digits = 0)), pch = 15,
              cex =1.25, las = 1, cex.lab = 1.4, cex.axis = 1.25,
              xlab = "nutrient supply", ylab = "zooplankton biomass (mg/L)", names.arg = c("low", "mediu

#add the error bars
arrows(x0 = bp, y0 = zp.means, y1 = zp.means - zp.sem, angle = 90, length = 0.1,
       lwd = 1)
arrows(x0 = bp, y0 = zp.means, y1 = zp.means + zp.sem, angle = 90, length = 0.1,
       lwd = 1)
```



```r
#conduct a one-way analysis of variance (ANOVA)
fitanova <- aov(ZP ~ NUTS, data = meso)

#look at the output with more detail
summary(fitanova)
```

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

```r
#conduct a post-hoc comparison of treatments using Tukey's HSD
# this will tell us whether or not there are differences (alpha = 0.05) among pairs
# of the three nutrient treatments
TukeyHSD(fitanova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
```

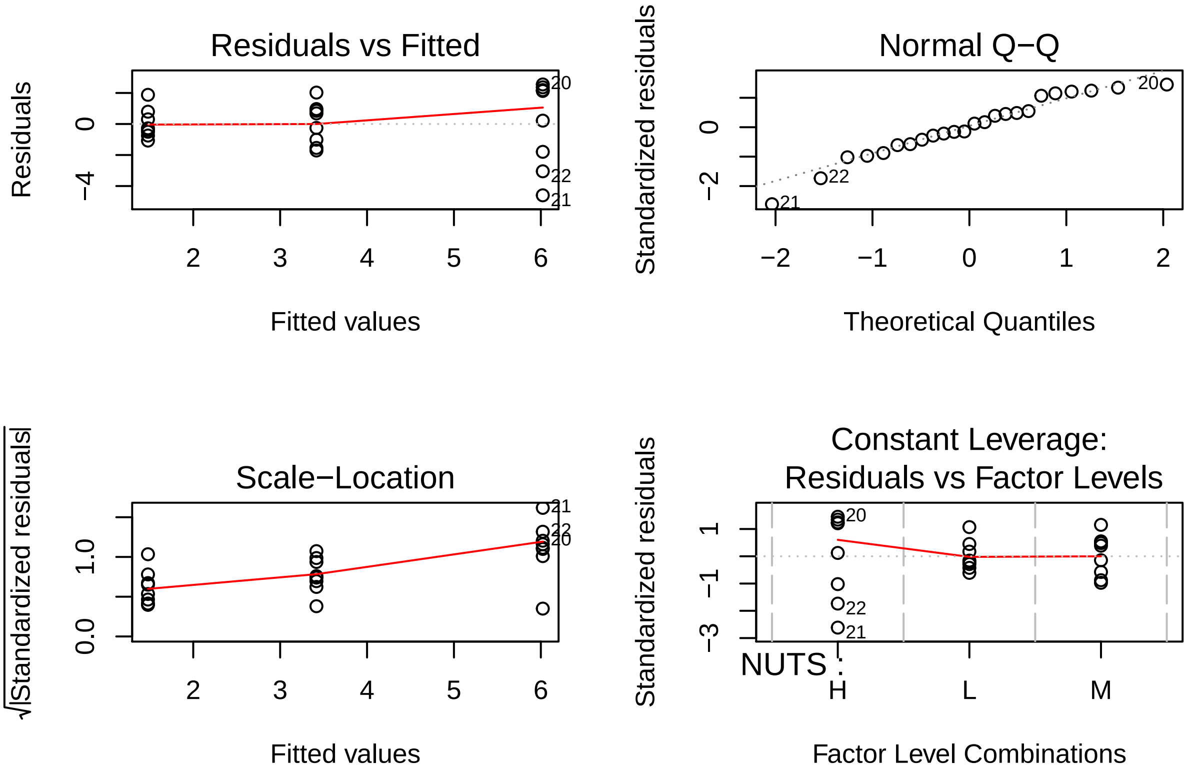
```
##
## Fit: aov(formula = ZP ~ NUTS, data = meso)
##
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## M-L  1.938625 -0.4297094  4.3069594 0.1220246
```

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

> ***Answer 8***: Since the regression results showed a strong linear fit, there appears to be a strong linear relationship between total nitrogen and zooplankton biomass. The Tukey's HSD results showed adjusted p values lower than 0.5 for the Low vs High and Medium vs High comparisons, meaning the high nutrient treatment significantly differed from the other two treatments. However, my concerns about this analysis would center on the lack of difference between the low and medium nutient levels treatments, because the ANOVA and Tukey's HSD results showed there was no significant difference, as the adjusted p value is $>$ 0.05. Given the type of data and ANOVA results, I would suggest the linear regression model is a better fit for this data.

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).

```
#look at the residuals
par(mfrow = c(2,2), mar = c(5.1, 4.1, 4.1, 2.1))
plot(fitanova)
```



SYNTHESIS: SITE-BY-SPECIES MATRIX

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

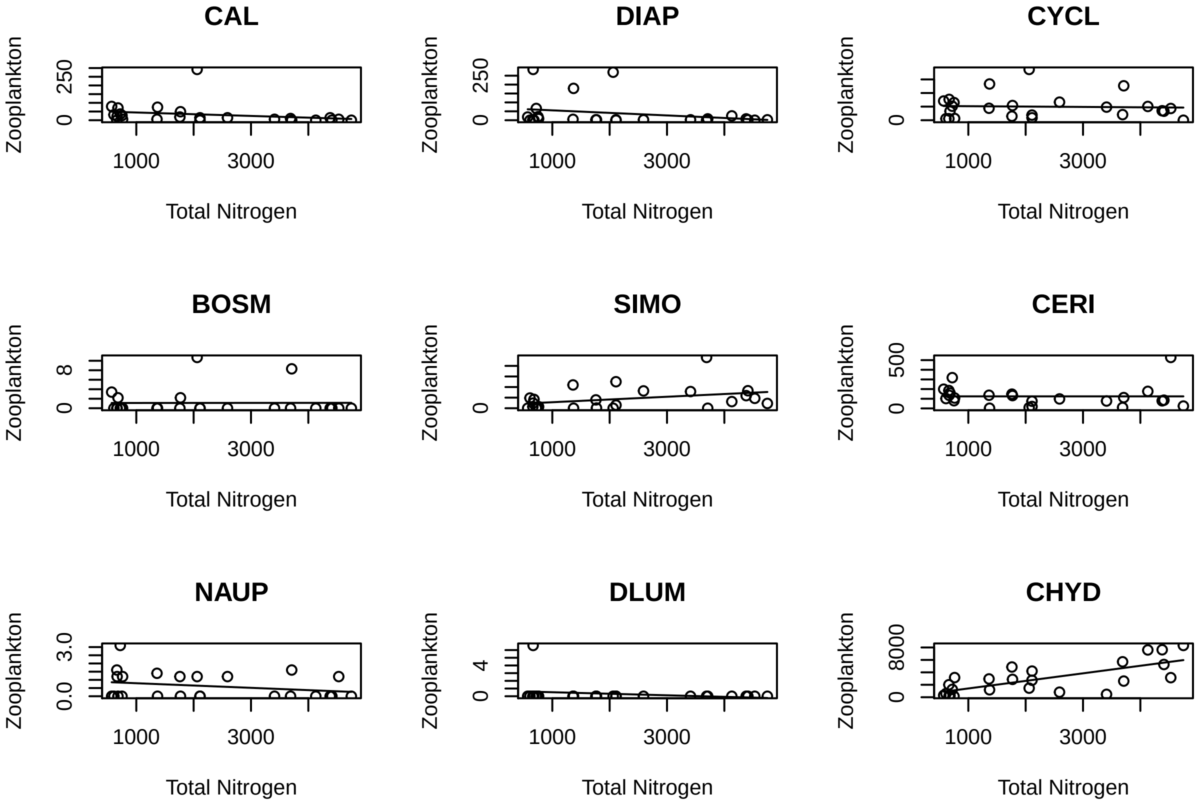
- CAL = calanoid copepods
- DIAP = *Diaphanasoma* sp.
- CYL = cyclopoid copepods
- BOSM = *Bosmina* sp.
- SIMO = *Simocephallus* sp.
- CERI = *Ceriodaphnia* sp.
- NAUP = naupuli (immature copepod)
- DLUM = *Daphnia lumholtzi*
- CHYD = *Chydorus* sp.

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

> ***Answer 9***: Using the code below, I plotted the different species against the total nitrogen values in each tank. The only species that followed the same general trend as the overall data were Simocephallus and Chydorus. They also had by far the largest numbers on the y-axis. Given that they were the only two that followed the same trend and had higher numbers than the other species, I would say that Simocephallus and Chydorus are the most responsible for the total biomass response to nutrient enrichment.

```r
#load zoops.txt data
zoops <- read.table("data/zoops.txt", sep = "\t", header = TRUE)

sortedmeso <- meso[order(meso$TANK), ]
sortedzoops <- zoops[order(zoops$TANK),]

zoopsnums <- cbind(sortedzoops[,3:dim(sortedzoops)[2]], sortedmeso$TN)
names(zoopsnums)[10] <- "TN"

#fit a lin reg model to all species

par(mfrow=c(3,3))

lmCAL <- lm(CAL ~ TN, data = zoopsnums)
lmDIAP <- lm(DIAP ~ TN, data = zoopsnums)
lmCYCL <- lm(CYCL ~ TN, data = zoopsnums)
lmBOSM <- lm(BOSM ~ TN, data = zoopsnums)
lmSIMO <- lm(SIMO ~ TN, data = zoopsnums)
lmCERI <- lm(CERI ~ TN, data = zoopsnums)
lmNAUP <- lm(NAUP ~ TN, data = zoopsnums)
```

```r
lmDLUM <- lm(DLUM ~ TN, data = zoopsnums)
lmCHYD <- lm(CHYD ~ TN, data = zoopsnums)

#get new x values for the model
newTN <- seq(min(zoopsnums$TN), max(zoopsnums$TN), 10)

plot(zoopsnums$TN, zoopsnums$CAL, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "CAL")
CALregline <- predict(lmCAL, newdata = data.frame(TN = newTN))
lines(newTN, CALregline)
plot(zoopsnums$TN, zoopsnums$DIAP, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "DIAP")
DIAPregline <- predict(lmDIAP, newdata = data.frame(TN = newTN))
lines(newTN, DIAPregline)
plot(zoopsnums$TN, zoopsnums$CYCL, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "CYCL")
CYCLregline <- predict(lmCYCL, newdata = data.frame(TN = newTN))
lines(newTN, CYCLregline)
plot(zoopsnums$TN, zoopsnums$BOSM, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "BOSM")
BOSMregline <- predict(lmBOSM, newdata = data.frame(TN = newTN))
lines(newTN, BOSMregline)
plot(zoopsnums$TN, zoopsnums$SIMO, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "SIMO")
SIMOregline <- predict(lmSIMO, newdata = data.frame(TN = newTN))
lines(newTN, SIMOregline)
plot(zoopsnums$TN, zoopsnums$CERI, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "CERI")
CERIregline <- predict(lmCERI, newdata = data.frame(TN = newTN))
lines(newTN, CERIregline)
plot(zoopsnums$TN, zoopsnums$NAUP, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "NAUP")
NAUPregline <- predict(lmNAUP, newdata = data.frame(TN = newTN))
lines(newTN, NAUPregline)
plot(zoopsnums$TN, zoopsnums$DLUM, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "DLUM")
DLUMregline <- predict(lmDLUM, newdata = data.frame(TN = newTN))
lines(newTN, DLUMregline)
plot(zoopsnums$TN, zoopsnums$CHYD, xlab = "Total Nitrogen", ylab = "Zooplankton", main = "CHYD")
CHYDregline <- predict(lmCHYD, newdata = data.frame(TN = newTN))
lines(newTN, CHYDregline)
```



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)**.