Week 1 Assignment: Basic R

Savannah Bennett; Z620: Quantitative Biodiversity, Indiana University

19 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. I## 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. 8. 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.gThis 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/Savannah/GitHub/QB2017_Bennett/Week1"
```

```
setwd("C:/Users/Savannah/GitHub/QB2017_Bennett/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.

```
5^3
```

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

```
pi*2^2
```

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

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

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

```
log(200)
```

```
## [1] 5.298317
```

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(2,3,4,5,6)
w <- x * 14
ww <- (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(2,4,6,8,10)
xx <- k * x
d1 <- sample(w,3,replace=FALSE)
d2 <- sample(k,4,replace=FALSE)
d <- c(d1,d2)
```

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(v){sd(na.omit(v))/sqrt(length(na.omit(v)))}
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
?rnorm
```

```
## starting httpd help server ...
```

```
##  done
```

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

```
##             j        l
## [1,] 7.892525 44.48480
## [2,] 6.128935 20.03858
## [3,] 7.760202 22.91606
## [4,] 8.354352 17.55580
## [5,] 9.737878 25.27132
```

```r
mm <- matrix(c(j,l), nrow=5, ncol=2, byrow=FALSE) #Approach 2
mm
```

```
##          [,1]     [,2]
## [1,] 7.892525 44.48480
## [2,] 6.128935 20.03858
## [3,] 7.760202 22.91606
## [4,] 8.354352 17.55580
## [5,] 9.737878 25.27132
```

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 is used to create random numbers with a specific mean and standard deviation that exhibit a normal distribution.The arguments in this function specify that n is the number of observations (if length (n) >1, the length is taken to be the number required), mean is the vector of means, and sd is the vector of standard deviations.

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

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

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

> Answer 2:The dimensions of this matrix are now 5 X 10 (5 columns, 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`.

```
o <- m[, c(1:2, 4:5)]
o <- m[1:9, ]
```

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

> ***Answer 3***:In the last few steps, we specified which rows and columns would be included and removed 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.

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

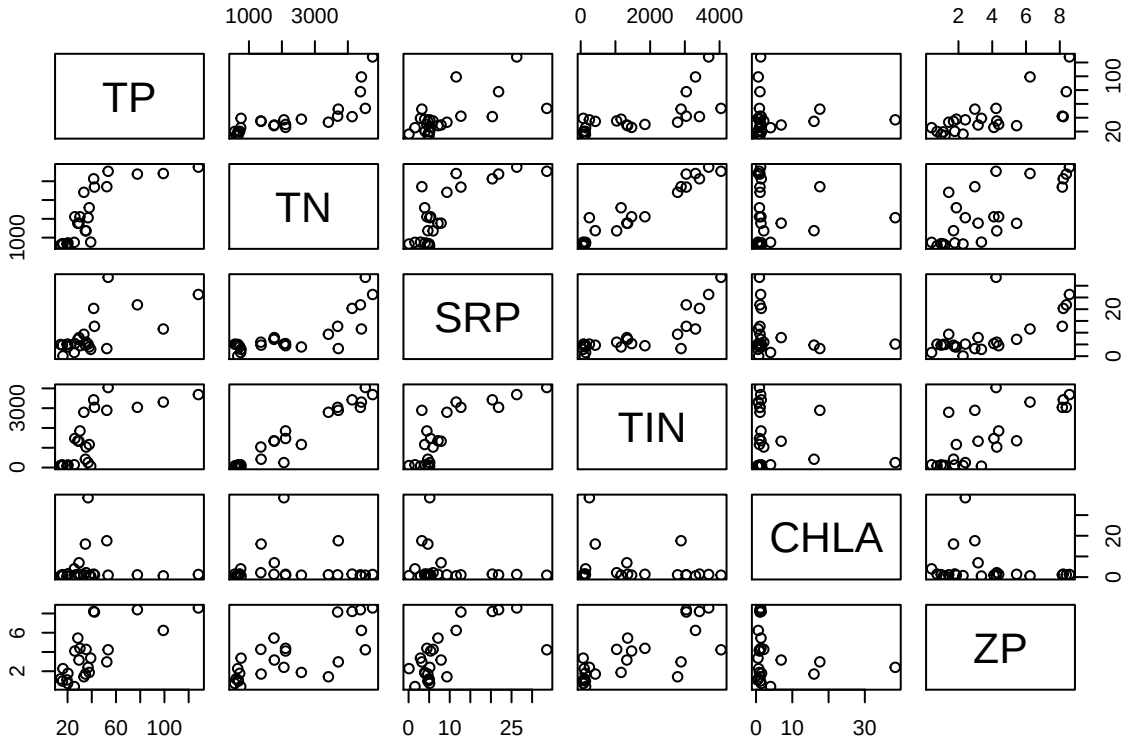
```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: Factor w/ 3 levels "H","L","M": 2 2 2 2 2 2 2 2 3 3 ...
##  $ TP  : num  20.3 25.6 14.2 39.1 20.1 ...
##  $ TN  : num  720 750 610 761 570 ...
##  $ SRP : num  4.02 1.56 4.97 2.89 5.11 4.68 5 0.1 7.9 3.92 ...
##  $ TIN : num  131.6 141.1 107.7 71.3 80.4 ...
```

```
##  $ CHLA: num  1.52 4 0.61 0.53 1.44 1.19 0.37 0.72 6.93 0.94 ...
##  $ ZP  : num  1.781 0.409 1.201 3.36 0.733 ...
```

Correlation

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

```
meso.num <- meso[,3:8]
pairs(meso.num)
```




```
cor1 <- cor(meso.num)
cor1
```

```
##               TP           TN         SRP        TIN         CHLA
## TP    1.00000000  0.786510407  0.6540957  0.7171143 -0.016659593
## TN    0.78651041  1.000000000  0.7841904  0.9689999 -0.004470263
## SRP   0.65409569  0.784190400  1.0000000  0.8009033 -0.189148017
## TIN   0.71711434  0.968999866  0.8009033  1.0000000 -0.156881463
## CHLA -0.01665959 -0.004470263 -0.1891480 -0.1568815  1.000000000
## ZP    0.69747649  0.756247384  0.6762947  0.7605629 -0.182599904
##               ZP
## TP    0.6974765
## TN    0.7562474
## SRP   0.6762947
## TIN   0.7605629
```

```
## CHLA -0.1825999
## ZP    1.0000000
```

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

> Answer 4: There are relatively weak negative relationships between chlorophyll concentrations and nutrient concentrations. However, there are relatively strong positive correlations between zooplankton biomass and nutrient concentrations such as total nitrogen concentrations and total inorganic nutrient concentrations. There are also relatively strong positive correlations between different nutrient measurements such as TP and TN, and TN and TIN.

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
```

```
cor2 <- corr.test(meso.num, method = "pearson", adjust = "BH")
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
```

```
cor2 <- corr.test(meso.num, method = "spearman", adjust = "BH")
print(cor2, digits = 3)
```

```
## Call:corr.test(x = meso.num, method = "spearman", adjust = "BH")
## Correlation matrix
##         TP    TN    SRP   TIN   CHLA     ZP
## TP   1.000 0.895  0.539 0.761  0.040  0.741
## TN   0.895 1.000  0.647 0.942  0.021  0.748
## SRP  0.539 0.647  1.000 0.726 -0.064  0.627
## TIN  0.761 0.942  0.726 1.000  0.088  0.738
## CHLA 0.040 0.021 -0.064 0.088  1.000 -0.072
## ZP   0.741 0.748  0.627 0.738 -0.072  1.000
## Sample Size
```

```
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         TP    TN   SRP   TIN  CHLA    ZP
## TP   0.000 0.000 0.010 0.000 0.914 0.000
## TN   0.000 0.000 0.001 0.000 0.923 0.000
## SRP  0.007 0.001 0.000 0.000 0.884 0.002
## TIN  0.000 0.000 0.000 0.000 0.884 0.000
## CHLA 0.853 0.923 0.767 0.683 0.000 0.884
## ZP   0.000 0.000 0.001 0.000 0.737 0.000
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

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

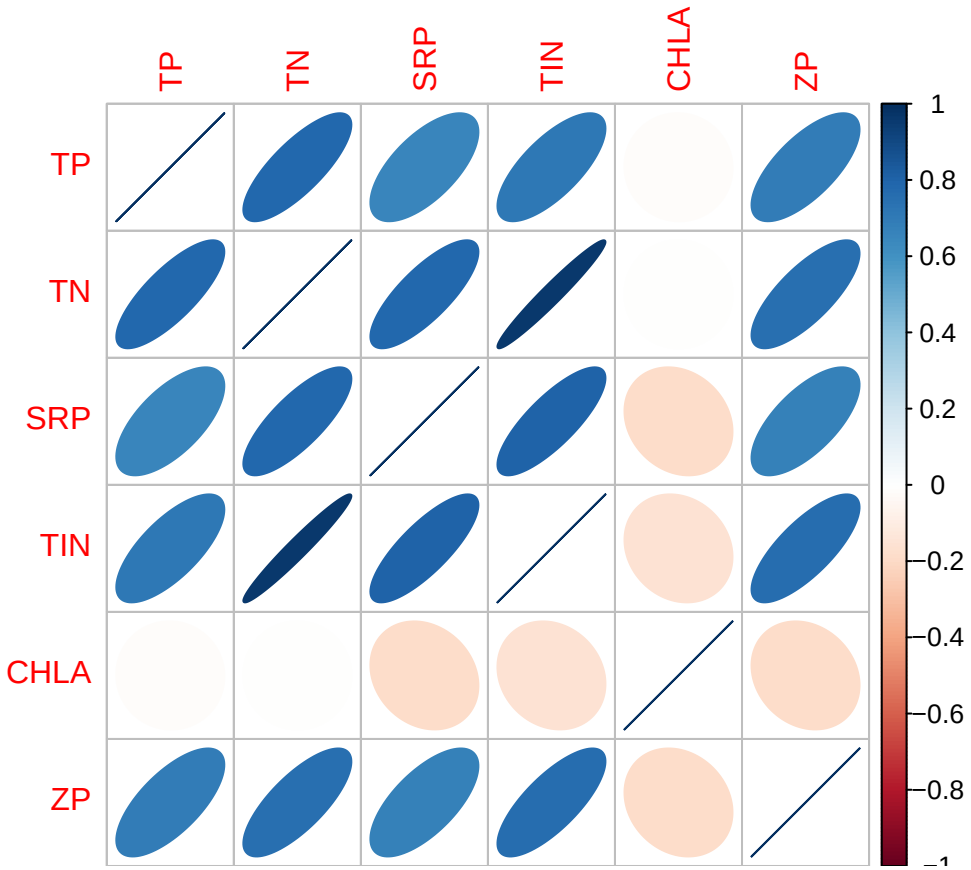
> ***Answer 5***: Yes, the results of the correlation analyses differed depending on whether a parametric or non-parametric test was performed. Non-parametric tests should be used when a data set does not exhibit a normal distribution, or when the sample size is very low. Several of the correlation matrix and probability values were higher under the Pearson's test compared to the Spearman's test. It is important to recognize these differences because they can lead to type I error, where a true null hypothesis is rejected. When the Spearman test is conducted, the adjust statement, BH, corrects for false discovery rate.

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
```

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



Linear Regression

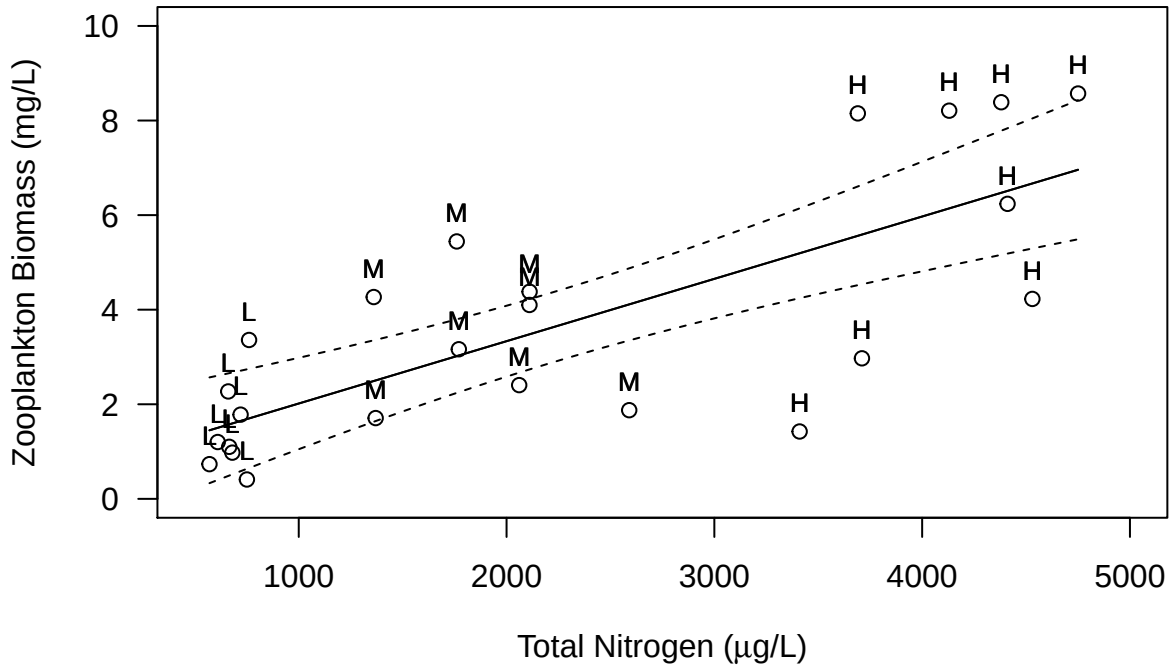
In the R code chunk below, do the following: 1) Conduct a linear regression analysis to test the relationship between total nitrogen (TN) and zooplankton biomass (ZP). 2) Examine the output of the regression analysis. 3) Produce a plot of this regression analysis including the following: categorically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

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

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

```
## Multiple R-squared:  0.5719, Adjusted R-squared:  0.5525
## F-statistic: 29.39 on 1 and 22 DF,  p-value: 1.911e-05
```

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




```
?predict()
```

Question 6: Interpret the results from the regression model

> ***Answer 6***:There is a significant positive relationship between total nitrogen and zooplankton biomass (p=1.91e-0.05, R-squared= 0.5525).

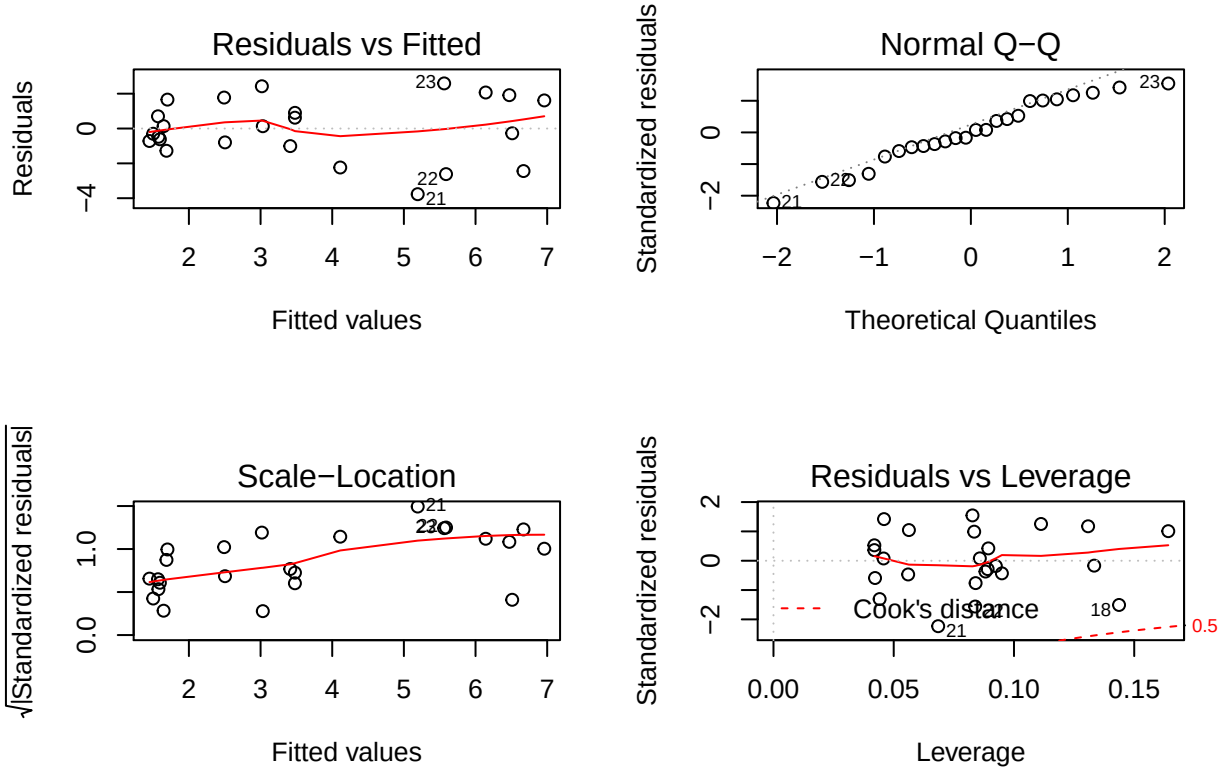
Question 7: Explain what the `predict()` function is doing in our analyses.

> ***Answer 7***: In general, the predict() function is used to predict values for new data. In our

analyses, this function predicts or creates values for the regression line. The predict() function is also used to create values for the 95% confidence intervals. It is essentially used to fit the regression an dcreate confidence intervals.

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)
```




```
help(qqplot)
#The data meet the assumptions of the linear regression analysis because there is a random distribution
```

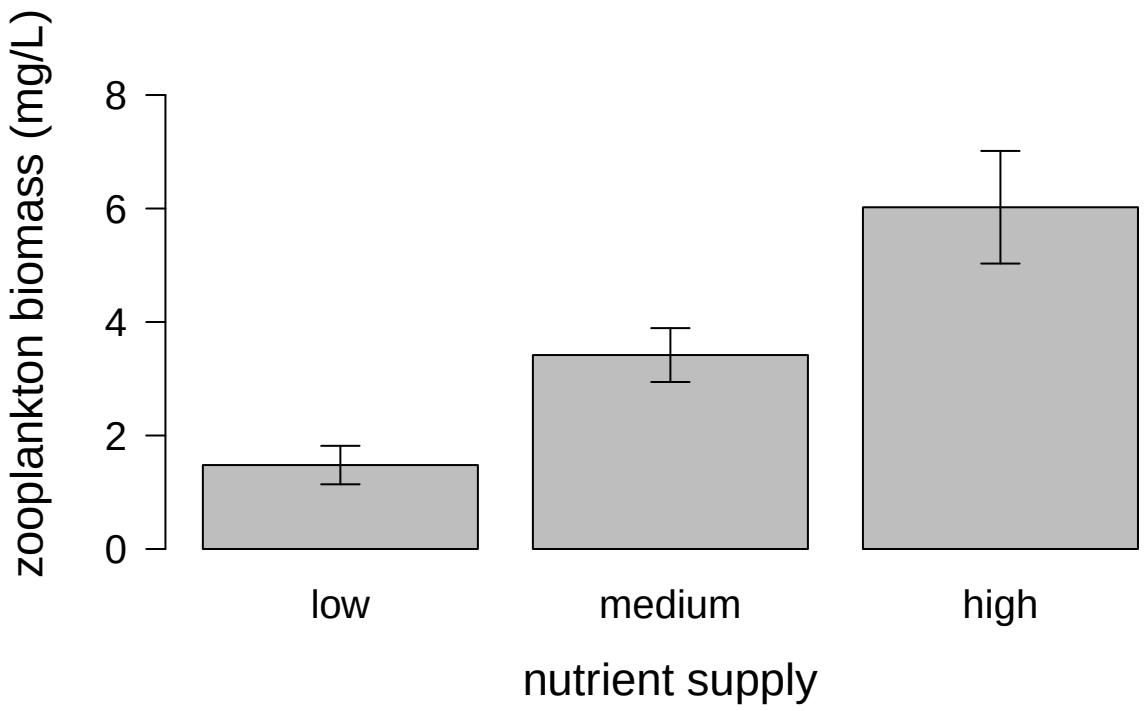
- Upper left: is there a random distribution of the residuals around zero (horizontal line)? >Yes, there seems to be a random distribution of residuals around zero.
- Upper right: is there a reasonably linear relationship between standardized residuals and theoretical quantiles? Try `help(qqplot)` >Yes, there does seem to be a reasonably linear relationship between standarized residuals and theoretical quantiles.
- Bottom left: again, looking for a random distribution of sqrt(standardized residuals) >Yes, there seems to be a random distribution of residuals around zero.
- 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.

```
NUTS <- factor(meso$NUTS, levels = c('L', 'M', 'H'))
zp.means <- tapply(meso$ZP, NUTS, mean)
zp.sem <- tapply(meso$ZP, NUTS, sem)
bp <- barplot(zp.means, ylim =c(0, round(max(meso$ZP), digits = 0)),
pch = 15, cex = 1.25, las = 1, cex.lab = 1.4, cex.axis = 1.25,
xlab = "nutrient supply",
ylab = "zooplankton biomass (mg/L)",
names.arg = c("low", "medium", "high"))
arrows(x0 = bp, y0 = zp.means, y1 = zp.means - zp.sem, angle = 90,
length=0.1, lwd = 1)
arrows(x0 = bp, y0 = zp.means, y1 = zp.means + zp.sem, angle = 90,
length=0.1, lwd = 1)
```




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

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

```
TukeyHSD(fitanova)
```

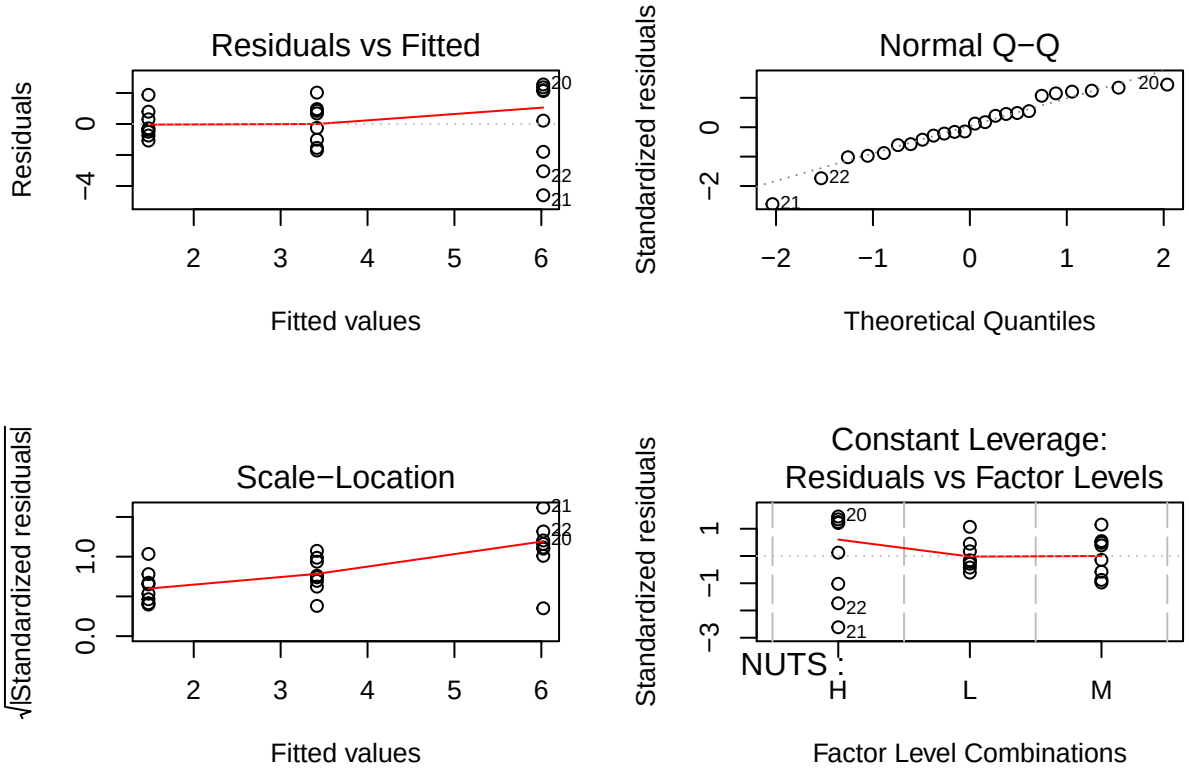
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ZP ~ NUTS, data = meso)
##
## $NUTS
##          diff        lwr        upr     p adj
## L-H -4.543175 -6.9115094 -2.1748406 0.0002512
## M-H -2.604550 -4.9728844 -0.2362156 0.0294932
## M-L 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

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

> ***Answer 8***:The anova results indicate that water with higher nutrient concentrations has significantly more zooplankton than water with a low nutrient supply. Similarly, the high nutrient concentration treatment has significantly more zooplankton than the medium nutrient supply treatment. However, the residuals plot below shows that the residuals are not randomly distributed around the zero. Therefore, this is not the proper/best analysis for these 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).

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



SYNTHESIS: SITE-BY-SPECIES MATRIX

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

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

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

```
zoo <- read.table("data/zoops.txt", sep = "\t", header = TRUE)
dat1 <- zoo[,c(3:11)]
ZP <- rowSums (dat1, na.rm = FALSE, dims = 1)
dat3 <- cbind(ZP,dat1)

reg1 <- lm(CAL ~ ZP, data=dat3)
reg2 <- lm(DIAP ~ ZP, data=dat3)
reg3 <- lm(CYCL ~ ZP, data=dat3)
reg4 <- lm(BOSM ~ ZP, data=dat3)
reg5 <- lm(SIMO ~ ZP, data=dat3)
reg6  <- lm(CERI ~ ZP, data=dat3)
reg7 <- lm(NAUP ~ ZP, data=dat3)
reg8 <- lm(DLUM ~ ZP, data=dat3)
reg9 <- lm(CHYD ~ ZP, data=dat3)

summary(reg1)
```

```
##
## Call:
## lm(formula = CAL ~ ZP, data = dat3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42.65 -27.66 -14.38  10.40 251.02
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 58.089171  20.916671   2.777    0.011 *
## ZP          -0.007093   0.004696  -1.511    0.145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 58.89 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
```

```
summary(reg2)
```

```
##
## Call:
## lm(formula = DIAP ~ ZP, data = dat3)
##
## 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
```

```
summary(reg3)
```

```
##
## Call:
## lm(formula = CYCL ~ ZP, data = dat3)
##
## 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
```

```
summary(reg4)
```

```
##
## Call:
## lm(formula = BOSM ~ ZP, data = dat3)
##
## 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
```

```
summary(reg5)
```

```
##
## Call:
## lm(formula = SIMO ~ ZP, data = dat3)
##
## 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
```

```
summary(reg6)
```

```
##
## Call:
## lm(formula = CERI ~ ZP, data = dat3)
##
## 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
```

```
summary(reg7)
```

```
##
## Call:
## lm(formula = NAUP ~ ZP, data = dat3)
##
## 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
```

```
summary(reg8)
```

```
##
## Call:
## lm(formula = DLUM ~ ZP, data = dat3)
##
## 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
```

```
summary(reg9)
```

```
##
## Call:
## lm(formula = CHYD ~ ZP, data = dat3)
##
## 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
```

Cylopoid copepods (p=0.08, R-squared=0.086), Simocephallus (p=0.03, R-squared=0.18), and Chydorus (p=e2-16, R-squared= 0.96) are primarily responsible for the total zooplankton biomass. Chydorus explains a large amount of variation in the total zooplankton biomass.

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)**.