3. Worksheet: Basic R

Mark Hibbins; Z620: Quantitative Biodiversity, Indiana University

12 January 2019

OVERVIEW

This worksheet introduces some of the basic features of the R computing environment (http://www.r-project.org). It is designed to be used along side the **3. RStudio** handout in your binder. You will not be able to complete the exercises without the corresponding handout.

Directions:

1. Change "Student Name" on line 3 (above) with your name.
2. Complete as much of the worksheet as possible during 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 exercises.
4. Answer questions in the worksheet. 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 worksheet, **Knit** the text and code into a single PDF file by pressing the `Knit` button in the RStudio scripting panel. This will save the PDF output in your '3.RStudio' folder.
7. After Knitting, please submit the worksheet by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (**3.RStudio_Worksheet.Rmd**) with all code blocks filled out and questions answered) and the PDF output of `Knitr` (**3.RStudio_Worksheet.pdf**).

The completed exercise is due on **Wednesday, January 16$^{\text{th}}$, 2019 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 Worksheet.

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 '3.RStudio' folder.

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

```
## [1] "C:/Users/Marcus/Box Sync/Courses/Quantitative Biodiversity/QB2019_Hibbins/2.Worksheets/3.RStudi
```

```
setwd('C:/Users/Marcus/Box\ Sync/Courses/Quantitative\ Biodiversity/QB2019_Hibbins/2.Worksheets/3.RStudi
```

3) USING R AS A CALCULATOR

To follow up on the pre-class exercises, please calculate the following in the R code chunk below. Feel free to reference the **1. Introduction to version control and computing tools** handout.

1) the volume of a cube with length, l, = 5 (volume = l^3)
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] 0.5
```

```
log(9)
```

```
## [1] 2.197225
```

4) WORKING WITH VECTORS

To follow up on the pre-class exercises, please perform the requested operations in the R-code chunks below.

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(9, 99, 999, 9999, 99999)
w <- x*14
(x + w)/14
```

```
## [1] 9.642857e+00 1.060714e+02 1.070357e+03 1.071321e+04 1.071418e+05
```

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(4, 7, 10, 13, 16)
k*w
```

```
## [1]      504     9702   139860  1819818 22399776
```

```
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)
v <- na.omit(v)
max(v)
```

```
## [1] 31.4
```

```
min(v)
```

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

```
sum(v)
```

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

```
mean(v)
```

```
## [1] 20.9
```

```
median(v)
```

```
## [1] 20.35
```

```
var(v)
```

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

```
sd(v)
```

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

```
sd(v)/sqrt(length(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 **3. RStudio** 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.

```
column1 <- c(rnorm(5, mean = 8, sd = 2))
column2 <- c(rnorm(5, mean = 25, sd = 10))
columns <- c(column1, column2)
matrix(data = columns, nrow = 5, ncol = 2, byrow = FALSE)
```

```
##           [,1]     [,2]
## [1,]  6.837668 13.65257
## [2,]  7.067790 14.04646
## [3,]  6.846778 24.91450
## [4,]  4.843873 41.13677
## [5,] 10.339204 27.45730
```

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: This function randomly samples the specified number of observations from a normal distribution with the specified mean and standard deviation.

In the R code chunk below, do the following: 1) Load `matrix.txt` from the **3.RStudio** data folder as matrix `m`. 2) Transpose this matrix. 3) Determine the dimensions of the transposed matrix.

```
m <- read.table('./data/matrix.txt')
n <- t(m)
dim(n)
```

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

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

> Answer 2: 10 rows by 5 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 <- m[c(1:9),]
```

6) BASIC DATA VISUALIZATION AND STATISTICAL ANALYSIS

Load Zooplankton Data Set

In the R code chunk below, do the following: 1) Load the zooplankton data set from the **3.RStudio** data folder. 2) Display the structure of this data set.

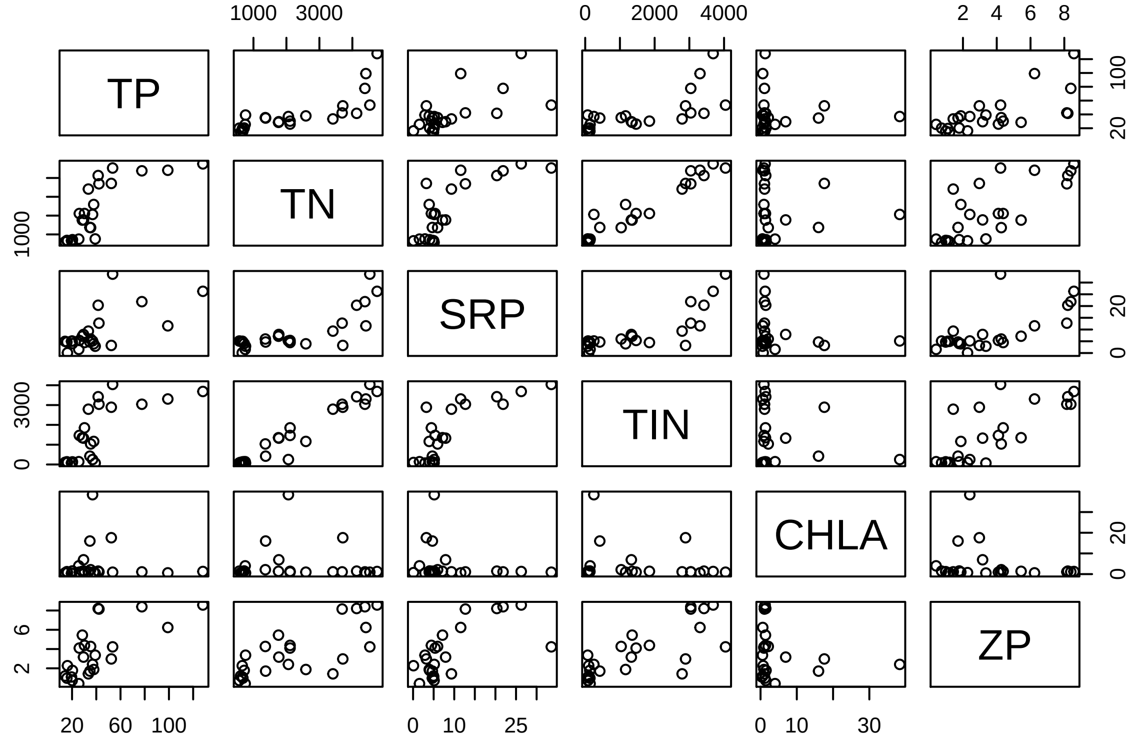
```
meso <- read.table('./data/zoop_nuts.txt', sep = '\t', header=TRUE)
str(meso)
```

```
## '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.

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



```r
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 3: Describe some of the general features based on the visualization and correlation analysis above?

> Answer 3: Many variables appear to be highly correlated, with coefficients in the 60-95% range. There are also quite a few appreciable, but not huge, coefficients in the 10-30$ range. Some particular variable combinations have a low correlation coefficient. As far as signs go, the variable CHLA appears to be negatively correlated with all the other variables (to varying degrees), while all the other coefficients are positive.

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.

```r
require('psych')
```

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

```r
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
```

```r
cor3 <- corr.test(meso.num, method = 'spearman', adjust = 'BH')
print(cor3, 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 4: 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 4: Using a non-parametric method does result in different coefficient estimates and p-values, but in this particular case, this effect does not appear to change our conclusions about the data. For Pearson's method, there is evidence for a false discovery rate, because the adjusted p-values above the diagonal are higher than the uncorrected p-values. P-values represent the probability that the data were generated under a given null hypothesis. Typically we reject the null for p-values below 0.05, but under this standard we will reject 5% of all datasets generated under the null hypothesis - these are false positives. For large datasets, 5% of results can represent a large number of p-values, so adjusting for this false positive rate is crucial.

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

```
##
## 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
require(ggplot2)
```

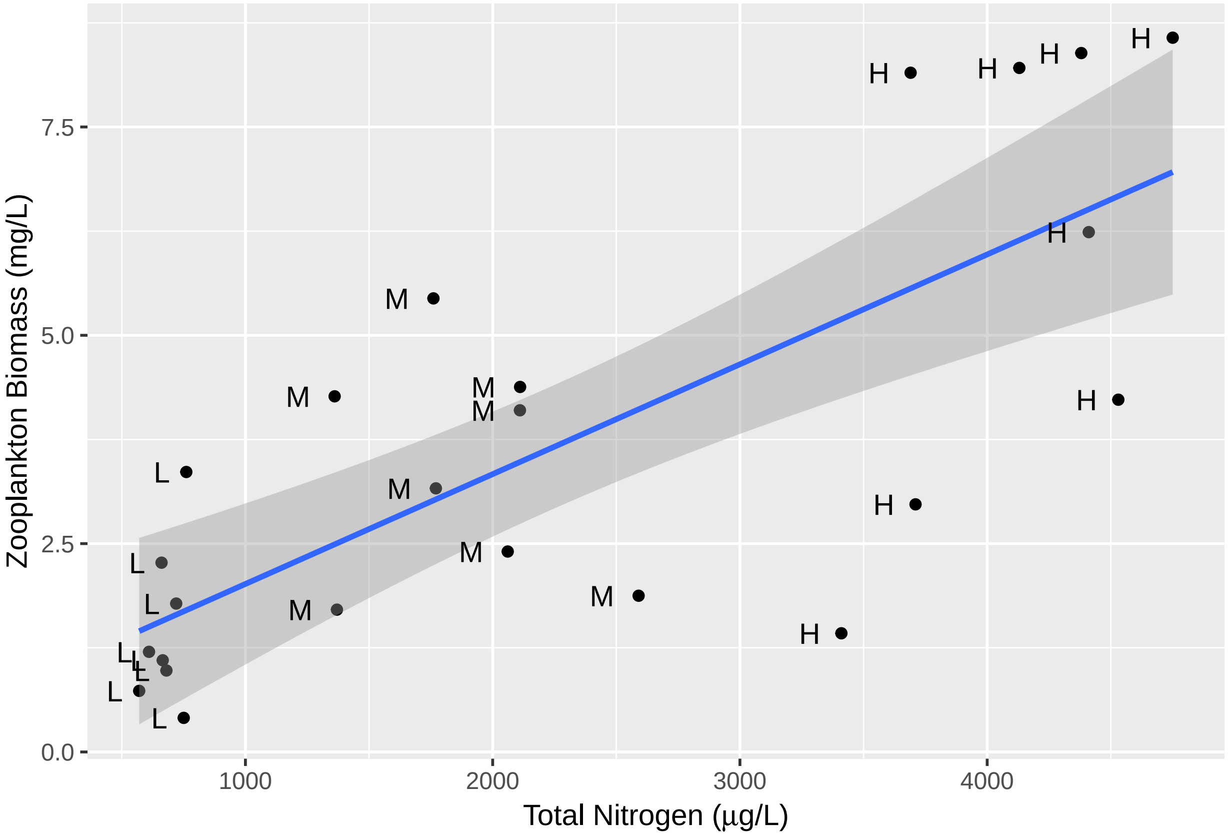
```
## Loading required package: ggplot2
```

```
##
```

```
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

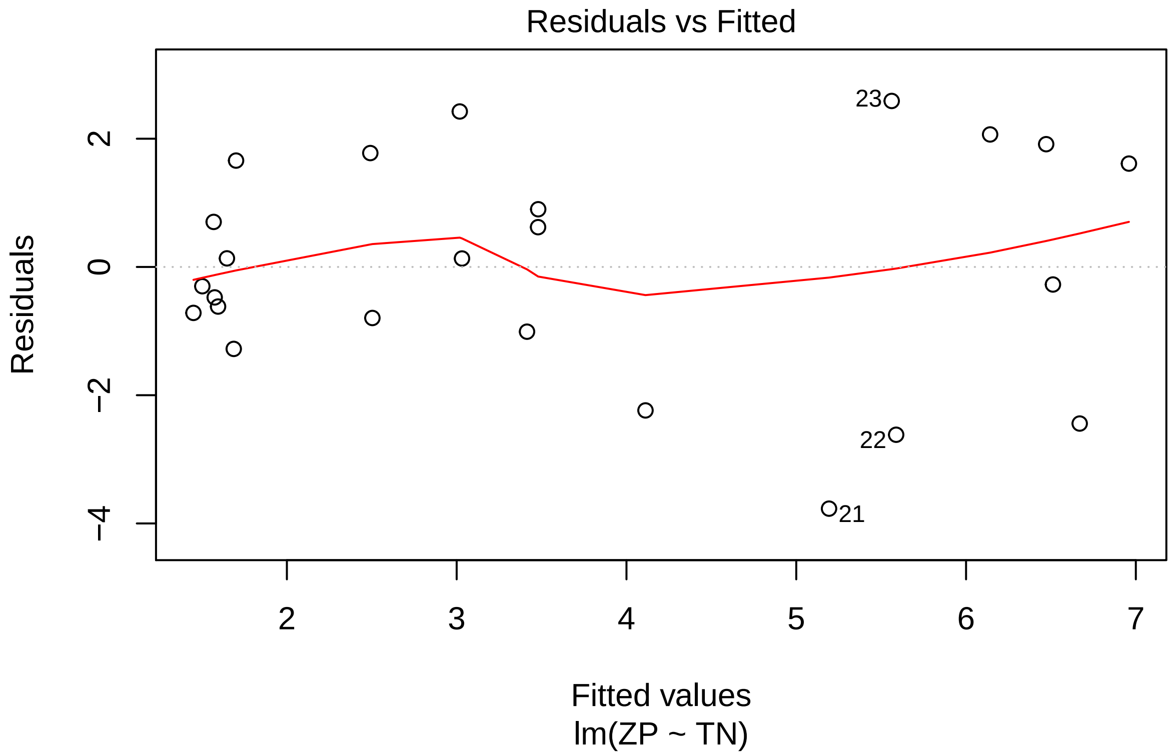
```
lm_plot <- ggplot(meso, aes(x = TN, y = ZP))
lm_plot <- lm_plot + geom_point()
lm_plot <- lm_plot + geom_smooth(method=lm)
lm_plot <- lm_plot + geom_text(aes(label=NUTS), hjust=2)
lm_plot <- lm_plot + labs(x = expression(paste('Total Nitrogen (', mu, 'g/L)')),
                          y = 'Zooplankton Biomass (mg/L)')
lm_plot
```

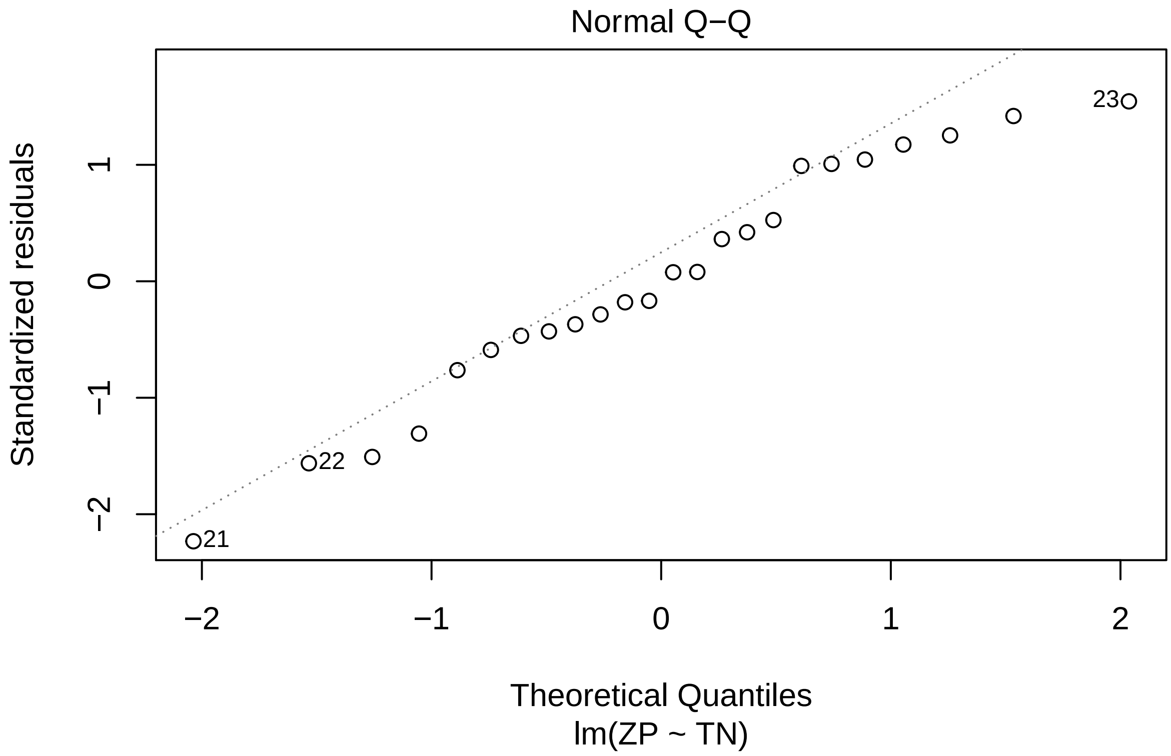


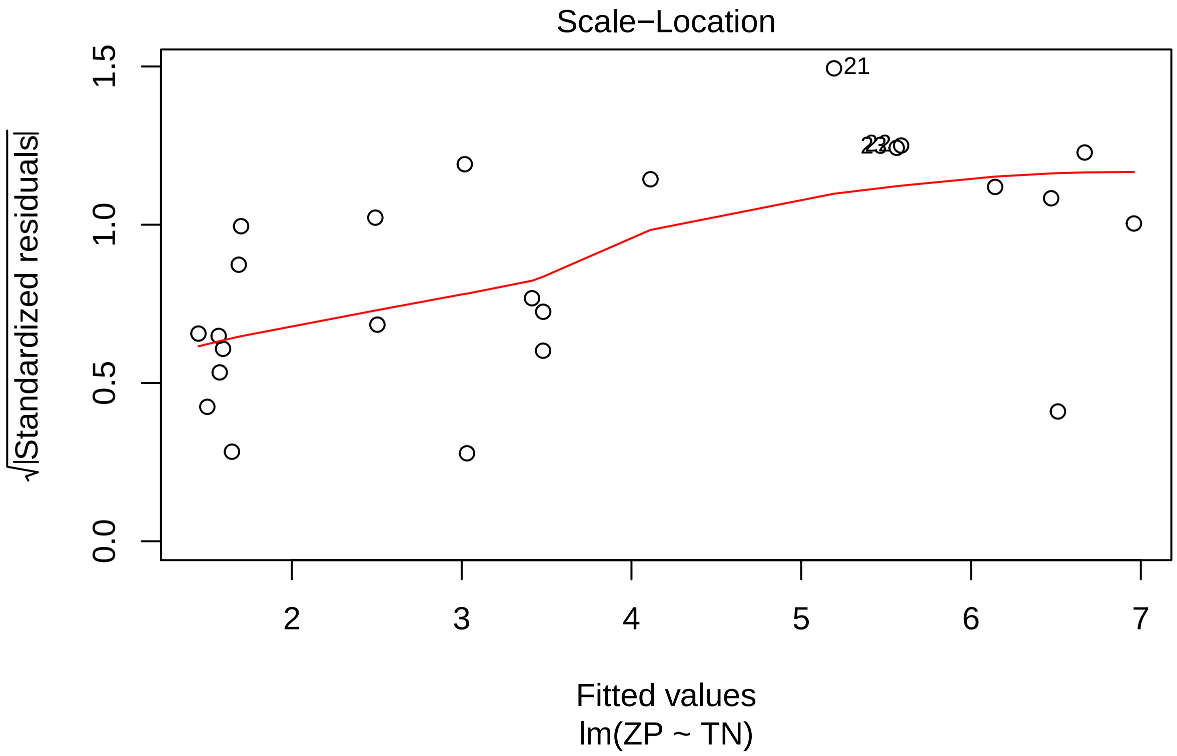
Question 5: Interpret the results from the regression model

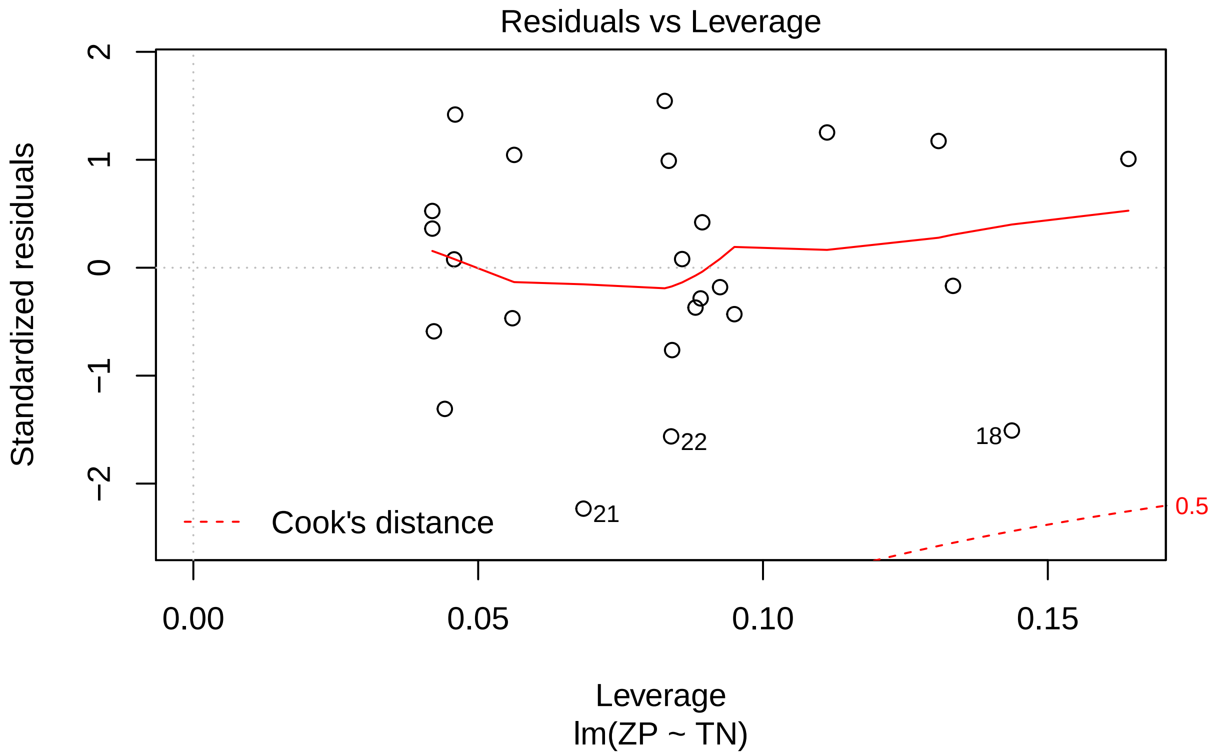
Answer 5: The slope of the relationship between TN and ZP is significantly (p = 0.00001911) different from 0. The estimate of 0.0013 means that for every 1 ug/L increase in total nitrogen, the zooplankton biomass increases by 0.001 mg/L. Based on the plots below, there do not appear to be any serious violations of the assumptions of linear regression.

```
plot(zoo_reg)
```


Residuals vs Fitted
Residuals
2
0
−2
−4
23
22
21
2
3
4
5
6
7
Fitted values
lm(ZP ~ TN)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(ZP ~ TN)
−2
−1
0
1
2
23
22
21


Scale−Location
1.5
1.0
0.5
0.0
√|Standardized residuals|
21
23
22
2
3
4
5
6
7
Fitted values
lm(ZP ~ TN)

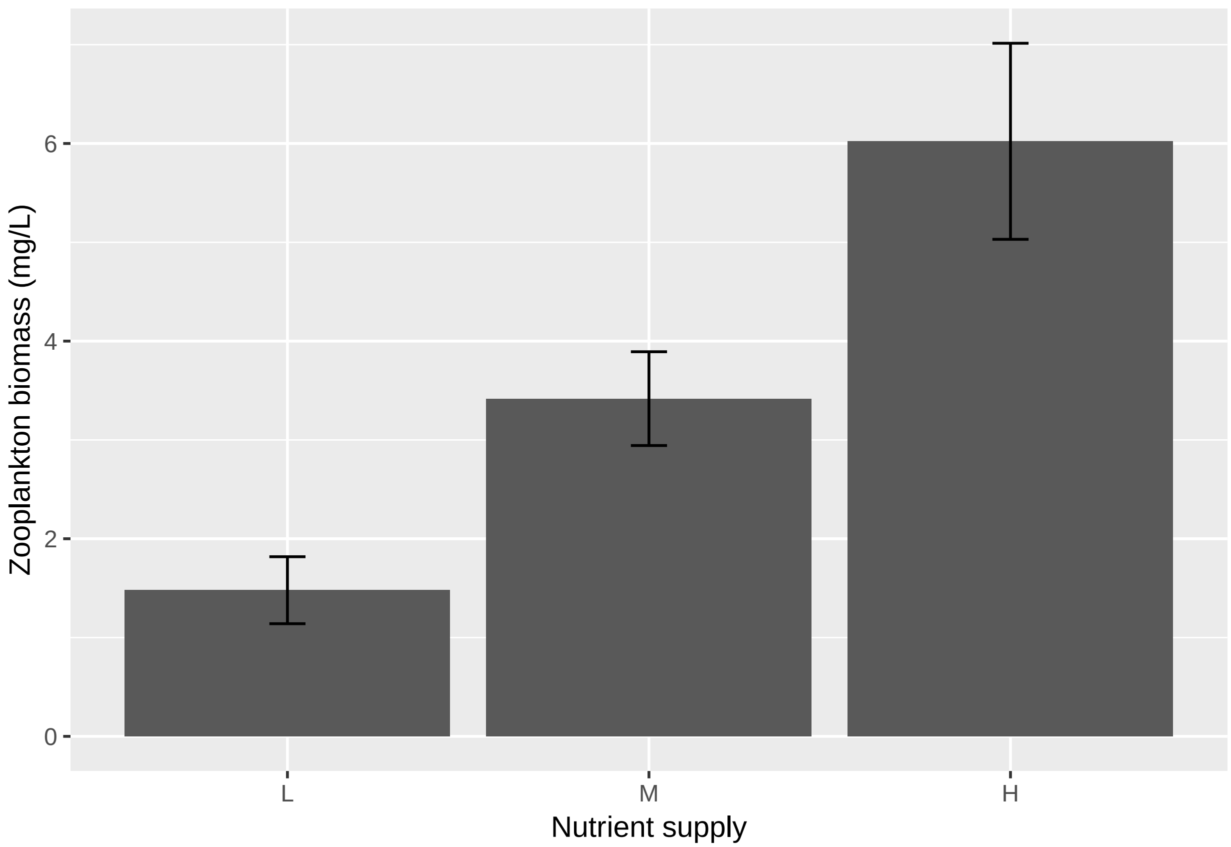


Analysis of Variance (ANOVA)

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.

```r
meso$NUTS <- factor(meso$NUTS, levels = c('L', 'M', 'H'))

sem <- function(y) {
  se <- sd(y)/sqrt(length(y))
  mu <- mean(y)
  c(ymin = mu-se, ymax = mu+se)
}
nut_barplot <- ggplot(meso, aes(x = NUTS, y = ZP))
nut_barplot <- nut_barplot + stat_summary(fun.y = mean, geom = 'bar')
nut_barplot <- nut_barplot + stat_summary(fun.data = sem,
                                          geom = 'errorbar',
                                          width = 0.1)
nut_barplot <- nut_barplot + labs(x = 'Nutrient supply',
                                  y = 'Zooplankton biomass (mg/L)')
nut_barplot
```



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

```
##             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
```

SYNTHESIS: SITE-BY-SPECIES MATRIX

In the R code chunk below, load the zoop.txt data set in your **3.RStudio** 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 6: With the visualization and statistical tools that we learned about in the **3. RStudio** 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 6: Based on the barplot and ANOVA analysis presented below, it seems likely that the trends in total biomass are driven mostly by the CHYD variable, whuch are Chydorus zooplankton. In fact, the mean biomass of this taxonomic group is not significantly different (p = 0.457) from the mean total biomass across all species.

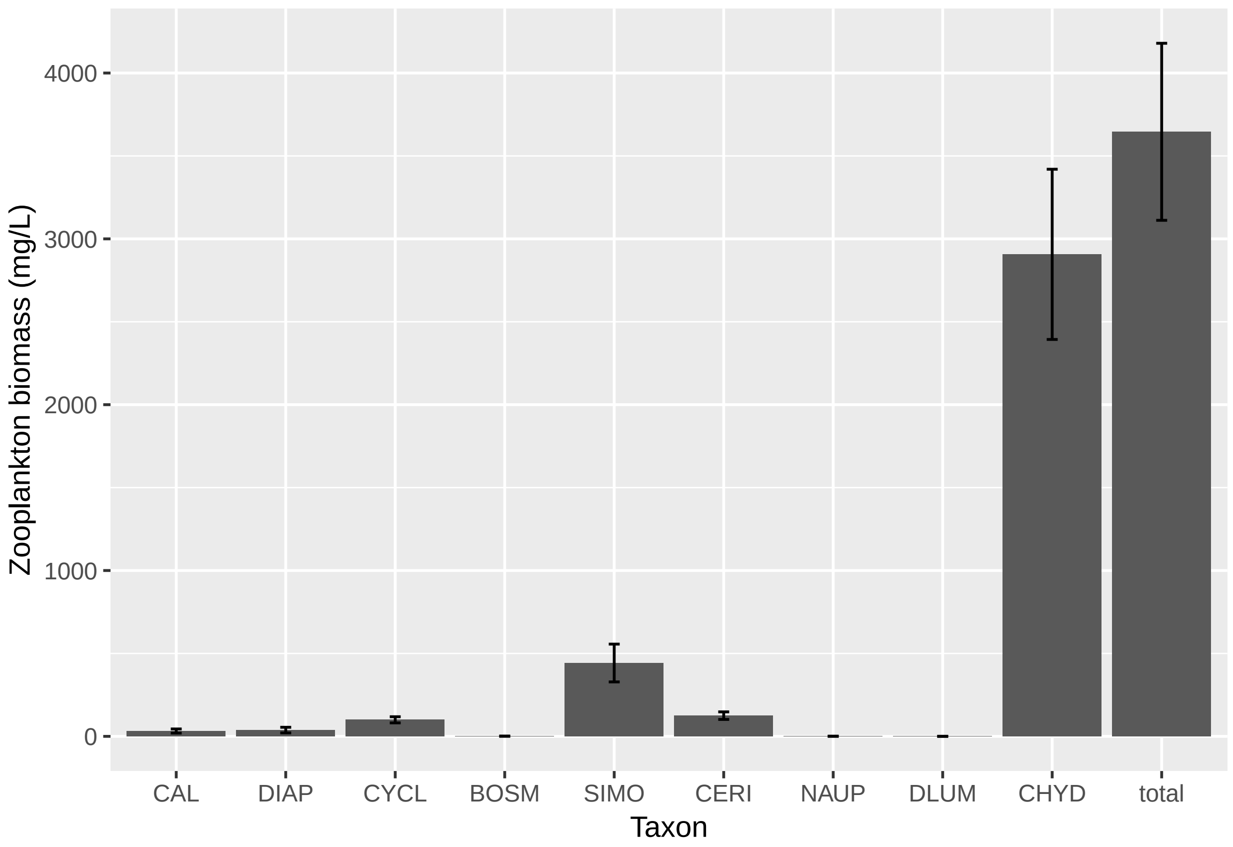
```r
zoops <- read.table('./data/zoops.txt', header = TRUE)
zoops <- zoops [,c(3:11)]
zoops$total <- rowSums(zoops)
require(reshape2)
```

```
## Loading required package: reshape2
```

```r
zoops.long <- melt(zoops)
```

```
## No id variables; using all as measure variables
```

```r
zoops_barplot <- ggplot(zoops.long, aes(x = variable, y = value))
zoops_barplot <- zoops_barplot + stat_summary(fun.y = mean, geom = 'bar')
zoops_barplot <- zoops_barplot + stat_summary(fun.data = sem,
                                          geom = 'errorbar',
                                          width = 0.1)
zoops_barplot <- zoops_barplot + labs(x = 'Taxon',
                                  y = 'Zooplankton biomass (mg/L)')
zoops_barplot
```



```
zoopsanova <- aov(value ~ variable, data = zoops.long)
summary(zoopsanova)
```

```
##              Df    Sum Sq  Mean Sq F value Pr(>F)    
## variable      9 399517253 44390806   32.88 <2e-16 ***
## Residuals   230 310531002  1350135                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(zoopsanova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ variable, data = zoops.long)
##
## $variable
##                    diff         lwr       upr     p adj
## DIAP-CAL      5.5958333 -1066.1184 1077.3101 1.0000000
## CYCL-CAL     67.5375000 -1004.1767 1139.2517 1.0000000
## BOSM-CAL    -31.1125000 -1102.8267 1040.6017 1.0000000
## SIMO-CAL    410.0708333  -661.6434 1481.7851 0.9681112
## CERI-CAL     92.6291667  -979.0851 1164.3434 0.9999998
## NAUP-CAL    -31.6083333 -1103.3226 1040.1059 1.0000000
## DLUM-CAL    -31.9541667 -1103.6684 1039.7601 1.0000000
## CHYD-CAL   2874.4083333  1802.6941 3946.1226 0.0000000
## total-CAL  3613.4000000  2541.6858 4685.1142 0.0000000
## CYCL-DIAP    61.9416667 -1009.7726 1133.6559 1.0000000
## BOSM-DIAP   -36.7083333 -1108.4226 1035.0059 1.0000000
## SIMO-DIAP   404.4750000  -667.2392 1476.1892 0.9708496
## CERI-DIAP    87.0333333  -984.6809 1158.7476 0.9999999
## NAUP-DIAP   -37.2041667 -1108.9184 1034.5101 1.0000000
## DLUM-DIAP   -37.5500000 -1109.2642 1034.1642 1.0000000
## CHYD-DIAP  2868.8125000  1797.0983 3940.5267 0.0000000
## total-DIAP 3607.8041667  2536.0899 4679.5184 0.0000000
## BOSM-CYCL   -98.6500000 -1170.3642  973.0642 0.9999997
## SIMO-CYCL   342.5333333  -729.1809 1414.2476 0.9907341
## CERI-CYCL    25.0916667 -1046.6226 1096.8059 1.0000000
## NAUP-CYCL   -99.1458333 -1170.8601  972.5684 0.9999997
## DLUM-CYCL   -99.4916667 -1171.2059  972.2226 0.9999997
## CHYD-CYCL  2806.8708333  1735.1566 3878.5851 0.0000000
## total-CYCL 3545.8625000  2474.1483 4617.5767 0.0000000
## SIMO-BOSM   441.1833333  -630.5309 1512.8976 0.9493122
## CERI-BOSM   123.7416667  -947.9726 1195.4559 0.9999978
## NAUP-BOSM    -0.4958333 -1072.2101 1071.2184 1.0000000
## DLUM-BOSM    -0.8416667 -1072.5559 1070.8726 1.0000000
## CHYD-BOSM  2905.5208333  1833.8066 3977.2351 0.0000000
## total-BOSM 3644.5125000  2572.7983 4716.2267 0.0000000
## CERI-SIMO  -317.4416667 -1389.1559  754.2726 0.9946901
## NAUP-SIMO  -441.6791667 -1513.3934  630.0351 0.9489603
## DLUM-SIMO  -442.0250000 -1513.7392  629.6892 0.9487137
## CHYD-SIMO  2464.3375000  1392.6233 3536.0517 0.0000000
## total-SIMO 3203.3291667  2131.6149 4275.0434 0.0000000
## NAUP-CERI  -124.2375000 -1195.9517  947.4767 0.9999977
## DLUM-CERI  -124.5833333 -1196.2976  947.1309 0.9999977
## CHYD-CERI  2781.7791667  1710.0649 3853.4934 0.0000000
## total-CERI 3520.7708333  2449.0566 4592.4851 0.0000000
## DLUM-NAUP    -0.3458333 -1072.0601 1071.3684 1.0000000
## CHYD-NAUP  2906.0166667  1834.3024 3977.7309 0.0000000
## total-NAUP 3645.0083333  2573.2941 4716.7226 0.0000000
## CHYD-DLUM  2906.3625000  1834.6483 3978.0767 0.0000000
## total-DLUM 3645.3541667  2573.6399 4717.0684 0.0000000
## total-CHYD  738.9916667  -332.7226 1810.7059 0.4576624
```

SUBMITTING YOUR WORKSHEET

Use Knitr to create a PDF of your completed **3.RStudio_Worksheet.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.

This assignment is due on **Wednesday, January 16$^{\text{th}}$, 2015 at 12:00 PM (noon)**.