3. Worksheet: Basic R

Maddy Spencer; Z620: Quantitative Biodiversity, Indiana University

22 January, 2025

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. In the Markdown version of this document in your cloned repo, 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 (color may vary if you changed the editor theme).
5. Before you leave the classroom today, you must **push** this file to your GitHub repo, at whatever stage you are. This will enable you to pull your work onto your own computer.
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 22$^{\text{nd}}$, 2025 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. Next to the `Knit` button in the RStudio scripting panel there is a spell checker button (`ABC`) button.

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.

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

```
## [1] "/cloud/project/QB2025_Spencer/Week1-RStudio"
```

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. 45°) 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(25)
```

```
## [1] 3.218876
```

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(1, 2, 3, 4, 5)
w <- x * 14
(x + w) / 15
```

```
## [1] 1 2 3 4 5
```

Now, do the following: 1) Create another vector (`k`) that is the same length as `w`. 2) Multiply `k` by `x`. 3) Use the combine function to create one more vector, `d` that consists of any three elements from `w` and any four elements of `k`.

```
x <- c(1, 2, 3, 4, 5)
w <- x * 14
k <- c(2, 4, 6, 8, 10)
y <- k * x
```

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

```
## [1] 14 28 42  2  4  6  8
```

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, 20.2, 13.1, 24.8, 20.2, 25.0, 20.5, 30.5, 31.4, 27.1)
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
```

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.

```
a <- c(rnorm(5, mean = 8, sd = 2))
b <- c(rnorm(5, mean = 25, sd = 10))
c <- matrix(c(a, b), nrow = 5, ncol = 2, byrow = FALSE)
print(c)
```

```
##           [,1]      [,2]
## [1,]  8.396336 30.971539
## [2,]  4.881731 44.091098
## [3,]  6.456635 31.456943
## [4,] 10.071893 19.847710
## [5,] 11.690502  9.244894
```

```
help(rnorm)
```

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 generates data that follows a normal distribution. The arguments in this specifies the number of elements/numbers generated (n), the mean of the data generated (mean), and the standard deviation of the numbers generated (sd).

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 <- 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 matrix is 5 columns and 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`.

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

```
## [1] 9 4
```

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.

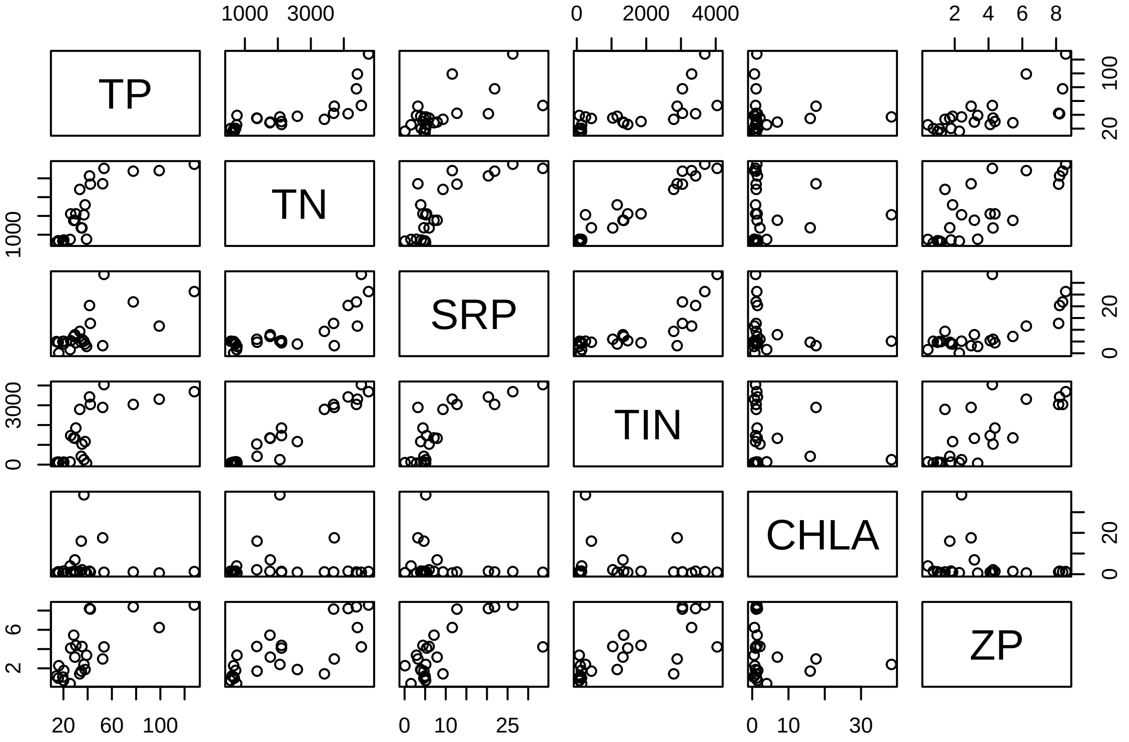
```
meso <- read.table("data/zoop_nuts.txt", sep = "\t", header = TRUE)
str(meso)
```

```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
##  $ NUTS: chr  "L" "L" "L" "L" ...
##  $ 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)
```



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

```
##              TP           TN        SRP        TIN         CHLA         ZP
## TP    1.00000000  0.786510407  0.6540957  0.7171143 -0.016659593  0.6974765
## TN    0.78651041  1.000000000  0.7841904  0.9689999 -0.004470263  0.7562474
## SRP   0.65409569  0.784190400  1.0000000  0.8009033 -0.189148017  0.6762947
## TIN   0.71711434  0.968999866  0.8009033  1.0000000 -0.156881463  0.7605629
## CHLA -0.01665959 -0.004470263 -0.1891480 -0.1568815  1.000000000 -0.1825999
## ZP    0.69747649  0.756247384  0.6762947  0.7605629 -0.182599904  1.0000000
```

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

> Answer 3: Soluble reactive phosphorus and total phosphorus have a positive linear correlation, as do total inorganic nitrogen and total nitrogen - these serve as good sanity checks as the two should be correlated in both cases. Zooplankton biomass is positively correlated with both nitrogen and phosphorus concentrations. Based on the correlations alone, chlorophyll alpha concentration is slightly negatively correlated with N and P concentration, but looking at the bi-plots themselves it appears as though there is an optimal N or P concentration for max chorolophyll alpha concentration.

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
require("corrplot")
```

```
## Loading required package: corrplot
## corrplot 0.95 loaded
```

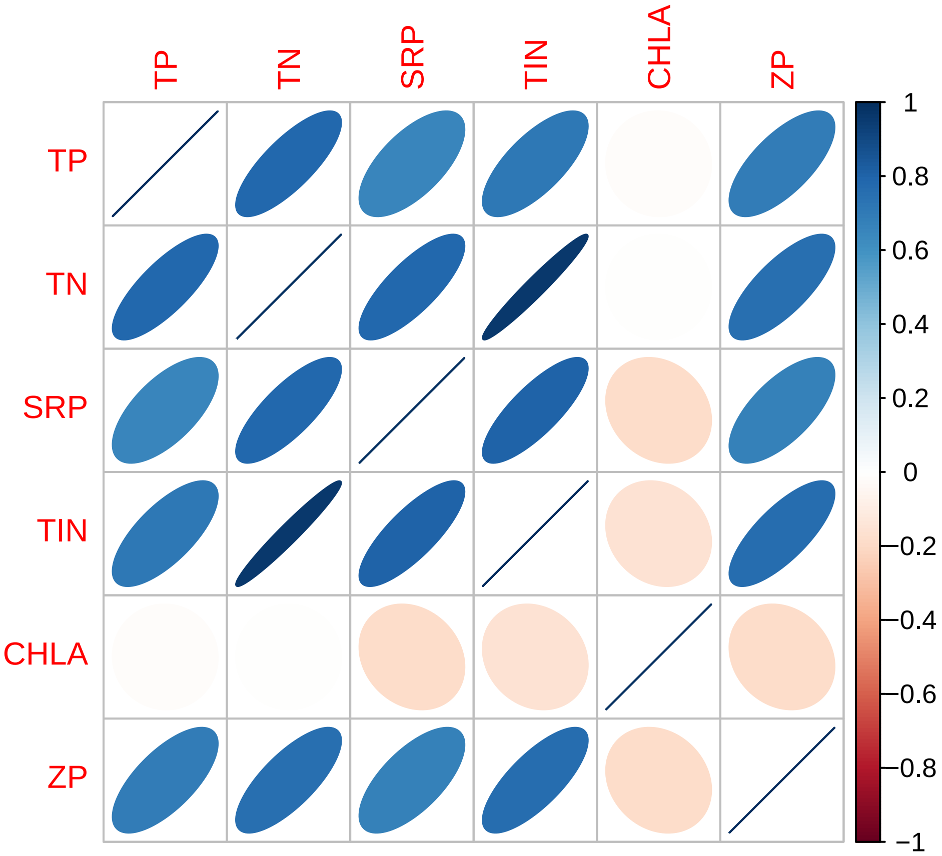
```
cor2 <- corr.test(meso.num, method = "pearson", adjust = "BH")
print(cor2, digits=3)
```

```
## 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
```

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

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



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: The results are sensitive to whether parametric or non-parametric methods are used. It seems that generally, correlations that are not significant in the parametric are not significant in the non-parametric test as well. For example, chlorophyll alpha concentration is not correlated to any other variable in both methods. Some p-values, however, became insignificant in the non-parametric test compared to the parametric test. For example, many of the zooplankton biomass correlations are significant using the parametric test, but not the non-parametric. Parametric tests operate on the assumption that the data input is normally distributed, while this is not assumed for non-parametric tests. Parametric testd are typically work better with larger sample sizes, whereas non-parametric is better suited to smaller data sets. With our data, there does not appear to be a large effect of false discovery rate using the Pearson's method. False discovery rate is important to factor into interpreting a Pearson's correlation test because the chances of a p-value being signficantly are increased with larger sample sizes.

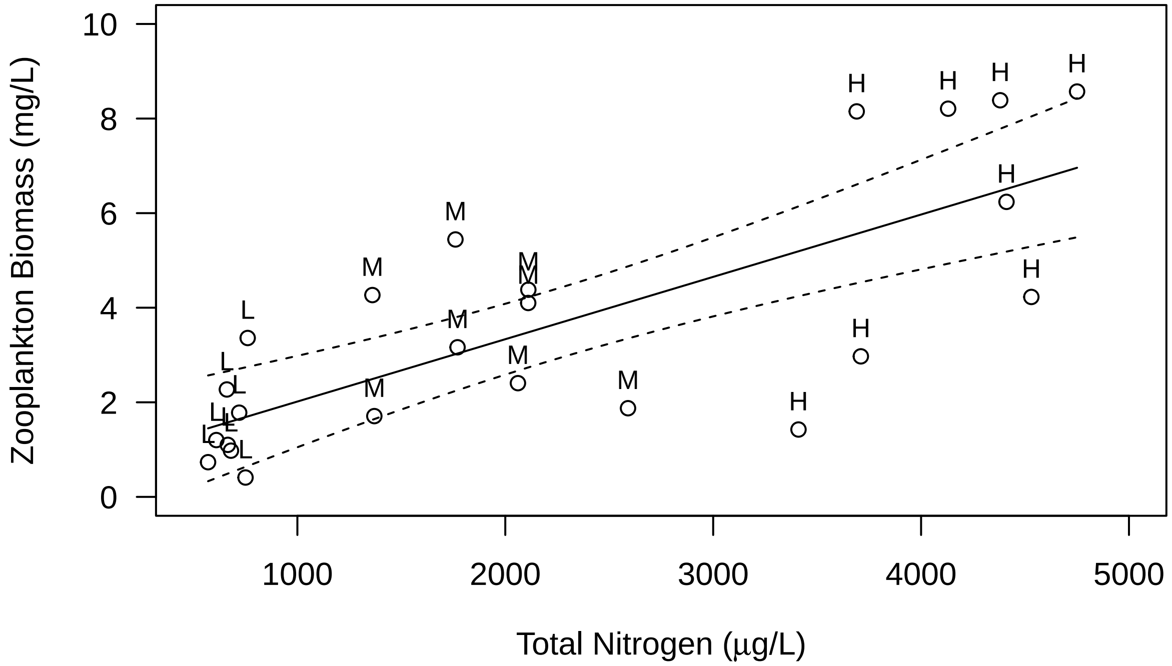
Linear Regression

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
```

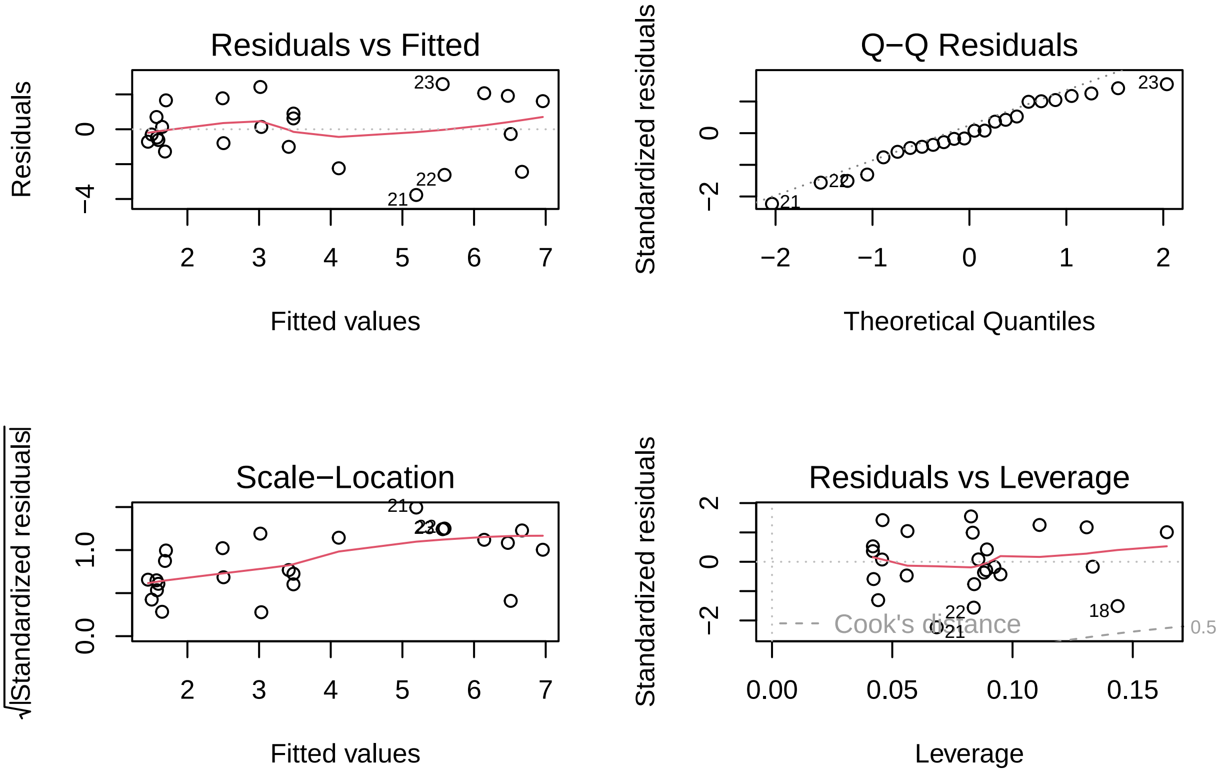
```
plot(meso$TN, meso$ZP, ylim = c(0,10), xlim = c(500, 5000),
    xlab = expression(paste("Total Nitrogen (", mu,"g/L)")),
    ylab = "Zooplankton Biomass (mg/L)", las = 1)
text(meso$TN, meso$ZP, meso$NUTS, pos = 3, cex = 0.8)
newTN <- seq(min(meso$TN), max(meso$TN), 10)
regline <- predict(fitreg, 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")
```



Question 5: Interpret the results from the regression model

Answer 5: Zooplankton biomass is positvely correlated with increasing levels of total nitrogen, meaning that zooplankton biomass increases with increasing input of nitrogen.

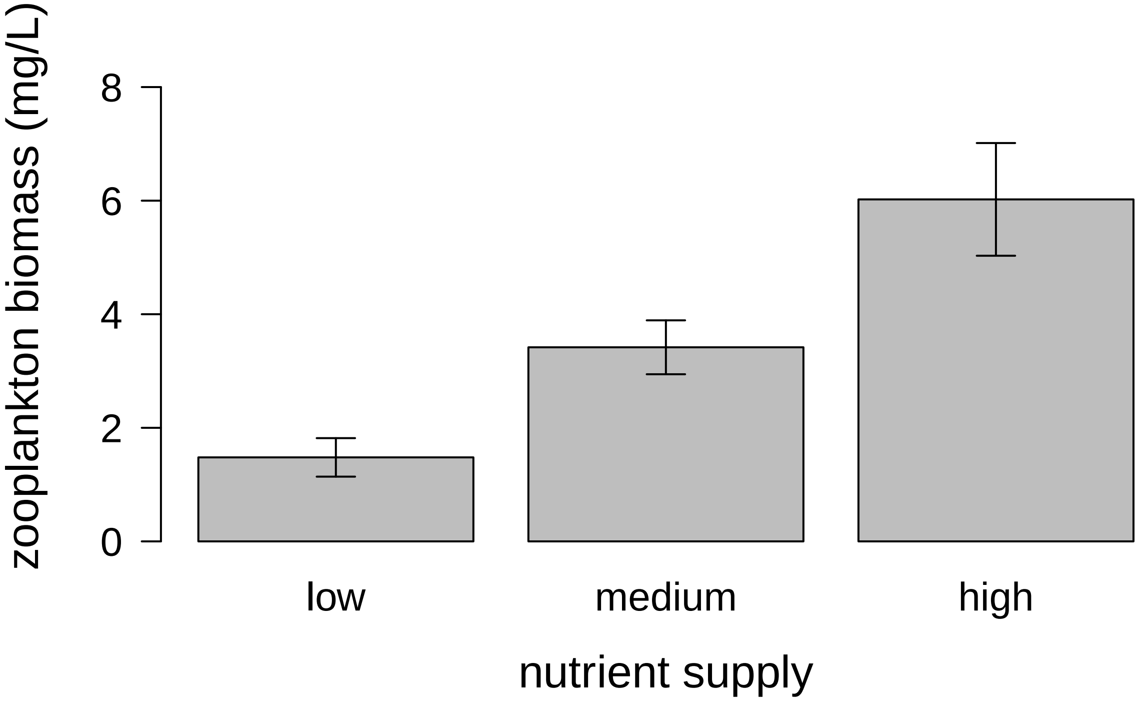
```r
par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
plot(fitreg)
```



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
NUTS <- factor(meso$NUTS, levels = c('L', 'M', 'H'))
zp.means <- tapply(meso$ZP, NUTS, mean)
sem <- function(x){
  sd(na.omit(x)/sqrt(length(na.omit(x))))
}
zp.sem <- tapply(meso$ZP, NUTS, sem)
bp <- barplot(zp.means, ylim =c(0, round(max(meso$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)
```



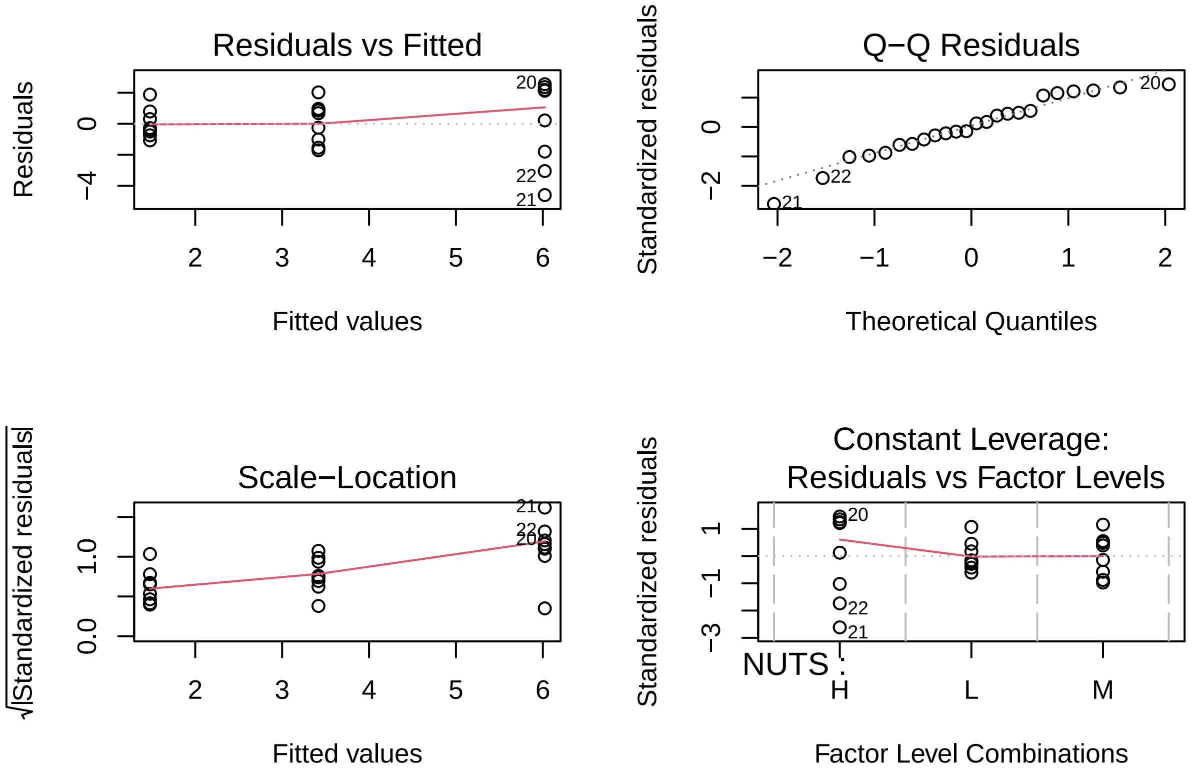
```r
fitanova <- aov(ZP ~ NUTS, data = meso)
summary(fitanova)
```

```
##             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
```

```r
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
```

```r
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 zoops.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 (µg/L) of different zooplankton taxa:

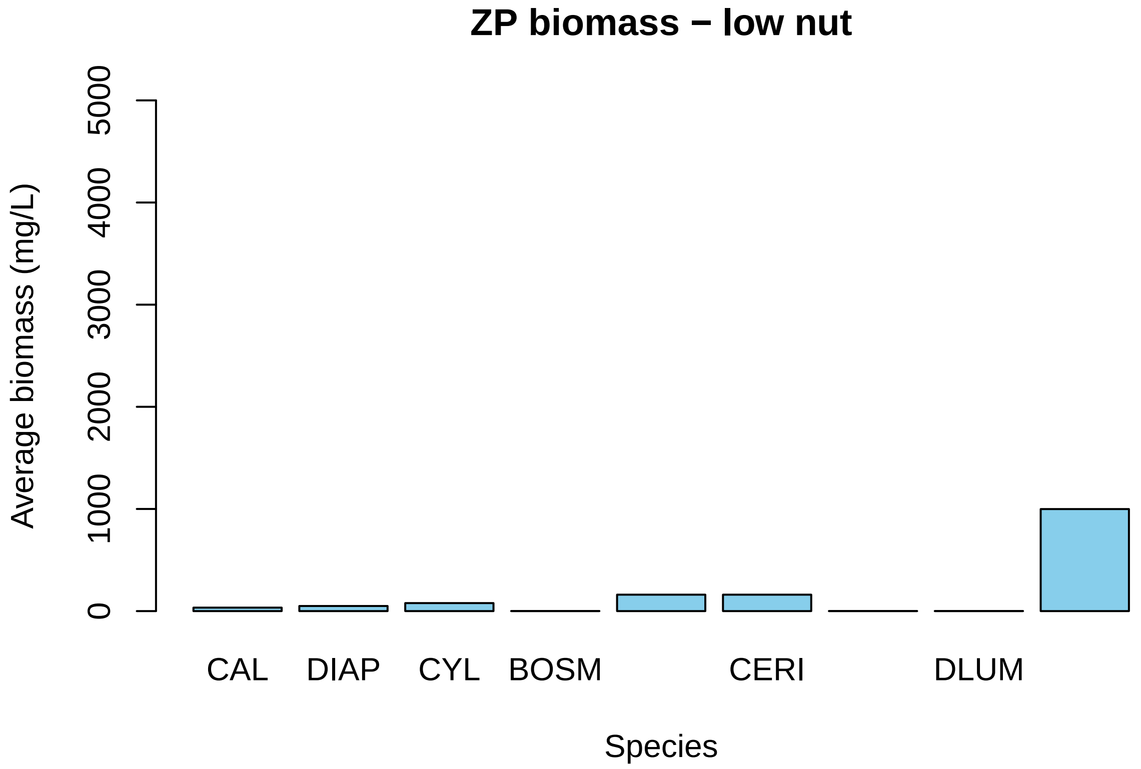
- CAL = calanoid copepods
- DIAP = *Diaphanasoma* sp.
- CYL = cyclopoid copepods
- BOSM = *Bosmina* sp.
- SIMO = *Simocephallus* sp.
- CERI = *Ceriodaphnia* sp.
- NAUP = naupuli (immature copepod)
- DLUM = *Daphnia lumholtzi*
- CHYD = *Chydorus* sp.

Question 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.

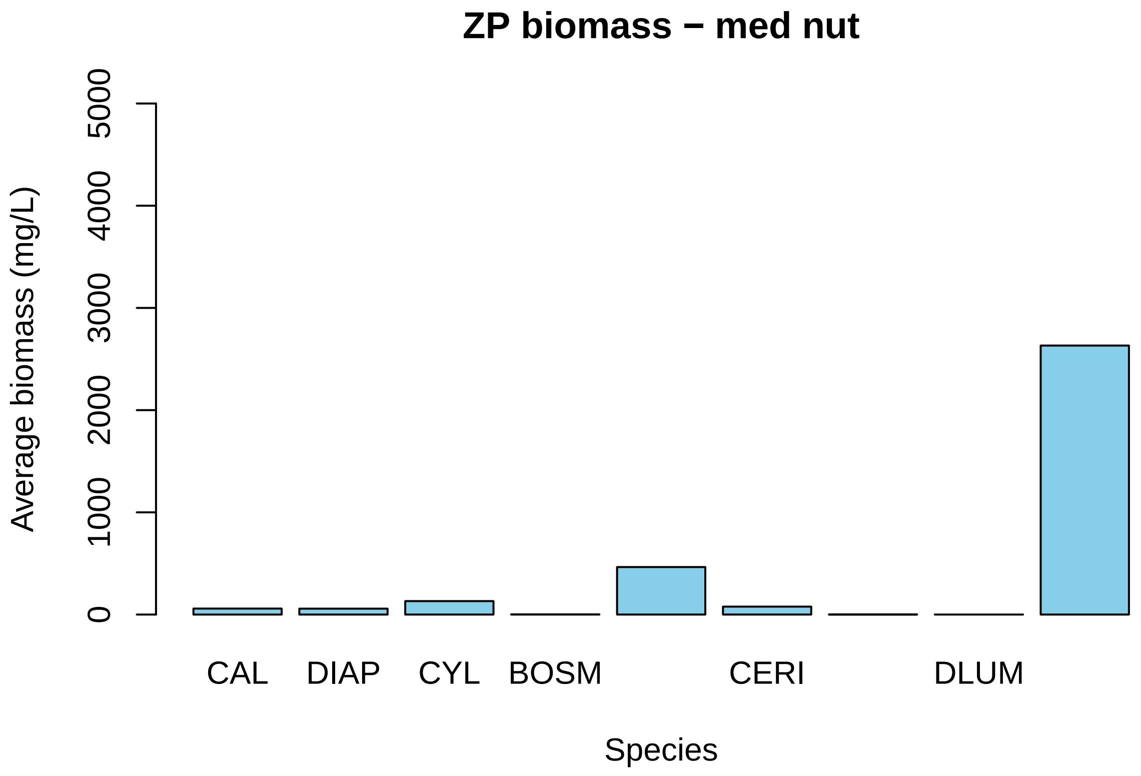
```r
z.m. <- as.matrix(read.table("data/zoops.txt", sep = "\t", header = FALSE))

str(z.m.)
```

```
##  chr [1:25, 1:11] "TANK" "5" "14" "16" "21" "23" "25" "27" "34" "12" "15" ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:11] "V1" "V2" "V3" "V4" ...
```

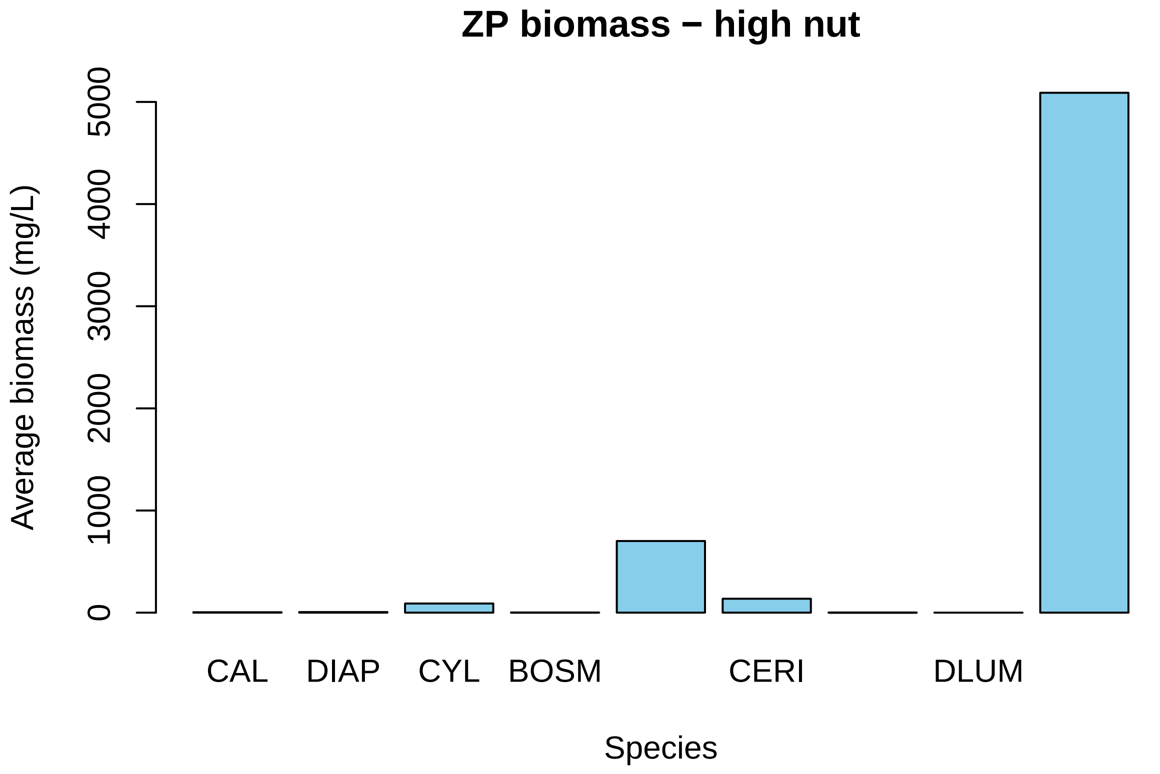
```
ZP.L <- z.m.[2:9,3:11]
ZP.L <- apply(ZP.L, 2, as.numeric)
bar_names <- c("CAL", "DIAP", "CYL", "BOSM", "SIMO", "CERI", "NAUP", "DLUM", "CHYD")
column_means <- colMeans(ZP.L, na.rm = FALSE)
barplot(column_means,  ylim = c(0, 5000), main = "ZP biomass - low nut",
        col = "skyblue",
        ylab = "Average biomass (mg/L)", xlab = "Species",
        names.arg = bar_names)
```



```
ZP.M <- z.m.[10:17,3:11]
ZP.M <- apply(ZP.M, 2, as.numeric)
bar_names <- c("CAL", "DIAP", "CYL", "BOSM", "SIMO", "CERI", "NAUP", "DLUM", "CHYD")
column_means <- colMeans(ZP.M, na.rm = FALSE)
barplot(column_means,  ylim = c(0, 5000), main = "ZP biomass - med nut",
        col = "skyblue",
        ylab = "Average biomass (mg/L)", xlab = "Species",
        names.arg = bar_names)
```



```r
ZP.H <- z.m.[18:25,3:11]
ZP.H <- apply(ZP.H, 2, as.numeric)
bar_names <- c("CAL", "DIAP", "CYL", "BOSM", "SIMO", "CERI", "NAUP", "DLUM", "CHYD")
column_means <- colMeans(ZP.H, na.rm = FALSE)
barplot(column_means,  ylim = c(0, 5000), main = "ZP biomass - high nut",
        col = "skyblue",
        ylab = "Average biomass (mg/L)", xlab = "Species",
        names.arg = bar_names)
```



Answer: It seems as though the *Chydorus* sp. has the strongest change in abundance due to nutrient concentration. *Chydorus* sp. goes from about 1000 mg/L in low nutrient concentrations to 5000 mg/L in high nutrient concentrations. *Simocephallus* sp. also responds to nutrient concentration, although at a much smaller scale.

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 22$^{\text{nd}}$, 2025 at 12:00 PM (noon)**.