3. Worksheet: Basic R

Trang Nguyen; Z620: Quantitative Biodiversity, Indiana University

21 janvier, 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 22nd, 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()) # clear working directory

print(getwd()) # print current working directory
```

```
## [1] "C:/Users/ttran/OneDrive - Indiana University/SP25 - Quantitative Biodiversity/QB2025_Nguyen/Weel
```

```
setwd(getwd())
# setwd("./3.RStudio") # set current workind to #3.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.

```
# Q1
l = 5
paste0("Volume of the cube:", l ^ 3)
```

```
## [1] "Volume of the cube:125"
```

```
# Q2
r = 2
paste0("Area of the circle:", pi * r ^ 2)
```

```
## [1] "Area of the circle:12.5663706143592"
```

```
# Q3
h_length = sqrt(2)
theta = pi / 4
paste0("Length of the opposite side of right triangle:", theta * h_length)
```

```
## [1] "Length of the opposite side of right triangle:1.11072073453959"
```

```
# Q4
paste0("Natural log of 1 is", log(1))
```

```
## [1] "Natural log of 1 is0"
```

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.

```
# 1) Create a vector `x` consisting of any five numbers.
x = c(1,2,3,4,5)
# 2) Create a new vector `w` by multiplying `x` by 14 (i.e., "scalar").
w = 14 * x
w
```

```
## [1] 14 28 42 56 70
```

```
# 3) Add `x` and `w` and divide by 15.
y = (x + w) / 15
y
```

```
## [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`.

```
# 1) Create another vector (`k`) that is the same length as `w`.
k = c(2,2,2,2,2)
# 2) Multiply `k` by `x`.
k * x
```

```
## [1]  2  4  6  8 10
```

```
# 3) Use the combine function to create one more vector, `d` that consists of any three elements from `
d = c(sample(w, size=3), sample(k, size=4))
d
```

```
## [1] 56 14 42  2  2  2  2
```

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)
paste0("maximum:", max(v, na.rm = TRUE))
```

```
## [1] "maximum:31.4"
```

```r
paste0("minimum:", min(v, na.rm = TRUE))
```

```
## [1] "minimum:10.1"
```

```r
paste0("sum:", sum(v, na.rm = TRUE))
```

```
## [1] "sum:292.6"
```

```r
paste0("mean:", mean(v, na.rm = TRUE))
```

```
## [1] "mean:20.9"
```

```r
paste0("median:", median(v, na.rm = TRUE))
```

```
## [1] "median:20.35"
```

```r
paste0("variance:", var(v, na.rm = TRUE))
```

```
## [1] "variance:39.44"
```

```r
paste0("std:", sd(v, na.rm = TRUE))
```

```
## [1] "std:6.28012738724303"
```

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.

```r
# Approach 1
c1 = rnorm(n=5, mean=8, sd=2)
c2 = rnorm(n=5, mean=25, sd=10)
m1 = cbind(c1, c2)
m1
```

```
##             c1       c2
## [1,]  7.803528 28.52973
## [2,]  7.109295 17.06242
## [3,] 10.480842 40.33958
## [4,]  3.678635 36.45980
## [5,]  8.205899 38.65559
```

```r
# Approach 2
m2 = matrix(c(c1, c2), nrow = 5, ncol = 2)
m2
```

```
##           [,1]     [,2]
## [1,]  7.803528 28.52973
## [2,]  7.109295 17.06242
## [3,] 10.480842 40.33958
## [4,]  3.678635 36.45980
## [5,]  8.205899 38.65559
```

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 generates a vector or a matrix of random number drawn from the normal distribution with 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.

```
# load data
m = read.table(paste0(getwd(), "/data/matrix.txt"))

# transpose
t(m)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## V1    8    5    2    3    9   11    2    3    5     6
## V2    1    5    5    2    9    8    2    3    5     5
## V3    7    2    4    5    1    1    5    6    1     9
## V4    6    4    3    1    1    8    8    7    3     2
## V5    1    1    3    4    2    8    5    6    6     2
```

```
# check dimension
dim(t(m))
```

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

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

> Answer 2: The matrix has 10 rows and 5 columns originally (before transposition).

###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`.

```
# rename everything but the third col
colnames(m)[c(1, 2, 4, 5)] = c("N1", "N2", "N4", "N5")
m
```

```
##    N1 N2 V3 N4 N5
## 1   8  1  7  6  1
## 2   5  5  2  4  1
## 3   2  5  4  3  3
## 4   3  2  5  1  4
```

```
## 5   9  9  1  1  2
## 6  11  8  1  8  8
## 7   2  2  5  8  5
## 8   3  3  6  7  6
## 9   5  5  1  3  6
## 10  6  5  9  2  2
```

```
# remove last row
m_new = m[-c(dim(m)),]
m_new
```

```
##   N1 N2 V3 N4 N5
## 1  8  1  7  6  1
## 2  5  5  2  4  1
## 3  2  5  4  3  3
## 4  3  2  5  1  4
## 6 11  8  1  8  8
## 7  2  2  5  8  5
## 8  3  3  6  7  6
## 9  5  5  1  3  6
```

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(paste0(getwd(), "/data/zoop_nuts.txt"), sep="\t", header = TRUE)
df_zoop = as.data.frame(meso)
summary(df_zoop)
```

```
##       TANK           NUTS                TP              TN
##  Min.   : 4.00   Length:24          Min.   : 14.22   Min.   : 570.4
##  1st Qu.:13.50   Class :character   1st Qu.: 24.24   1st Qu.: 742.9
##  Median :20.00   Mode  :character   Median : 34.10   Median :1915.7
##  Mean   :20.04                      Mean   : 39.91   Mean   :2231.9
##  3rd Qu.:27.25                      3rd Qu.: 41.72   3rd Qu.:3695.4
##  Max.   :36.00                      Max.   :128.04   Max.   :4750.4
##       SRP              TIN              CHLA              ZP
##  Min.   : 0.100   Min.   :  71.28   Min.   : 0.3700   Min.   :0.409
##  1st Qu.: 4.343   1st Qu.: 134.73   1st Qu.: 0.9375   1st Qu.:1.637
##  Median : 5.125   Median :1246.45   Median : 1.2300   Median :3.067
##  Mean   : 8.823   Mean   :1497.44   Mean   : 4.3137   Mean   :3.640
##  3rd Qu.: 9.883   3rd Qu.:2928.49   3rd Qu.: 1.6725   3rd Qu.:4.646
##  Max.   :33.570   Max.   :4042.10   Max.   :38.3800   Max.   :8.571
```

```
str(meso)
```

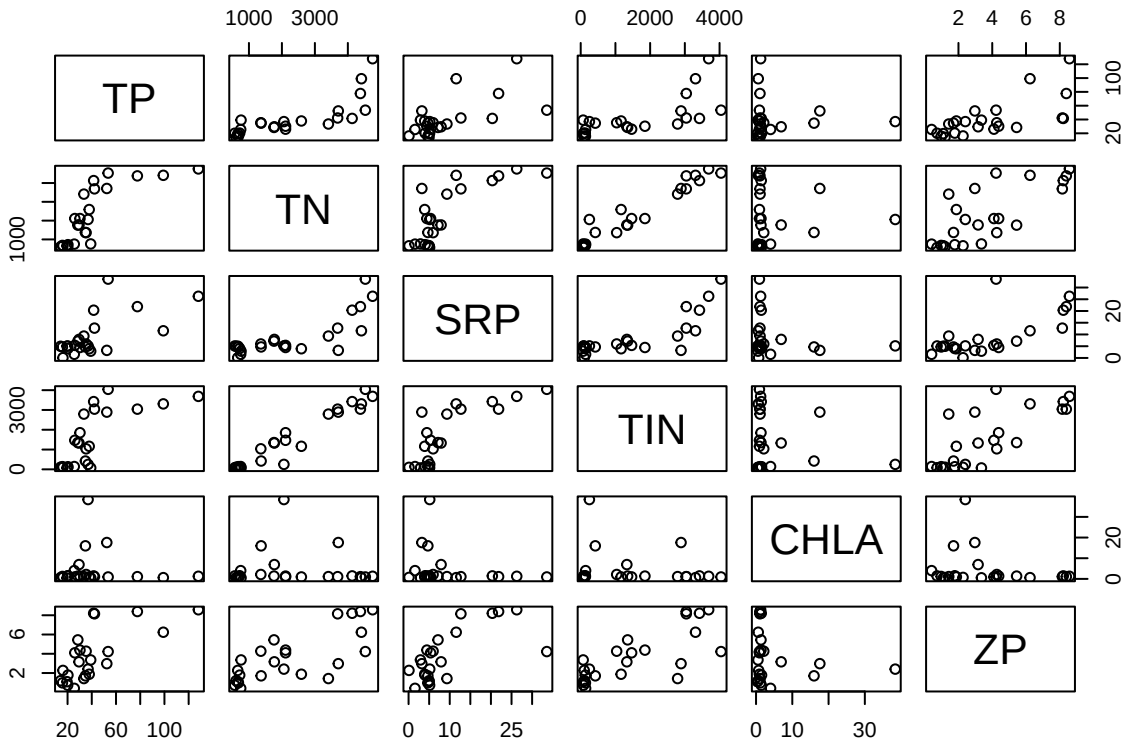
```
## 'data.frame':    24 obs. of  8 variables:
##  $ TANK: int  34 14 23 16 21 5 25 27 30 28 ...
```

```
##  $ 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]
# 2. Pairwise bi plots
pairs(meso.num)
```




```
# 3. Pearson's corr
cor1 = cor(meso.num)
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: Based on the pairwise plots and the Pearson correlation matrix, we see: Highly Correlated Variables: Variables with high positive correlations (e.g., TN and TIN, corr=0.96) show nearly diagonal patterns in their scatter plots. Low Correlation: Variables with low correlation (corr $\sim$ 0) no discernible trends in their scatter plots, appearing scattered randomly. Negative Correlation: Variables with negative correlations (e.g., SRP vs CHLA) exhibit a slight downward slope in their scatter plots.

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
# install.packages("psych")
require(psych)
```

```
## Le chargement a nécessité le package : psych
```

```
## Warning: le package 'psych' a été compilé avec la version R 4.4.2
```

```r
cor2 = corr.test(meso.num, method = "pearson")
print(cor2, digits=3)
```

```
## Call:corr.test(x = meso.num, method = "pearson")
## 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 0.003 0.001 1.000 0.001
## TN   0.000 0.000 0.000 0.000 1.000 0.000
## SRP  0.001 0.000 0.000 0.000 1.000 0.002
## TIN  0.000 0.000 0.000 0.000 1.000 0.000
## CHLA 0.938 0.983 0.376 0.464 0.000 1.000
## 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 = "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
```

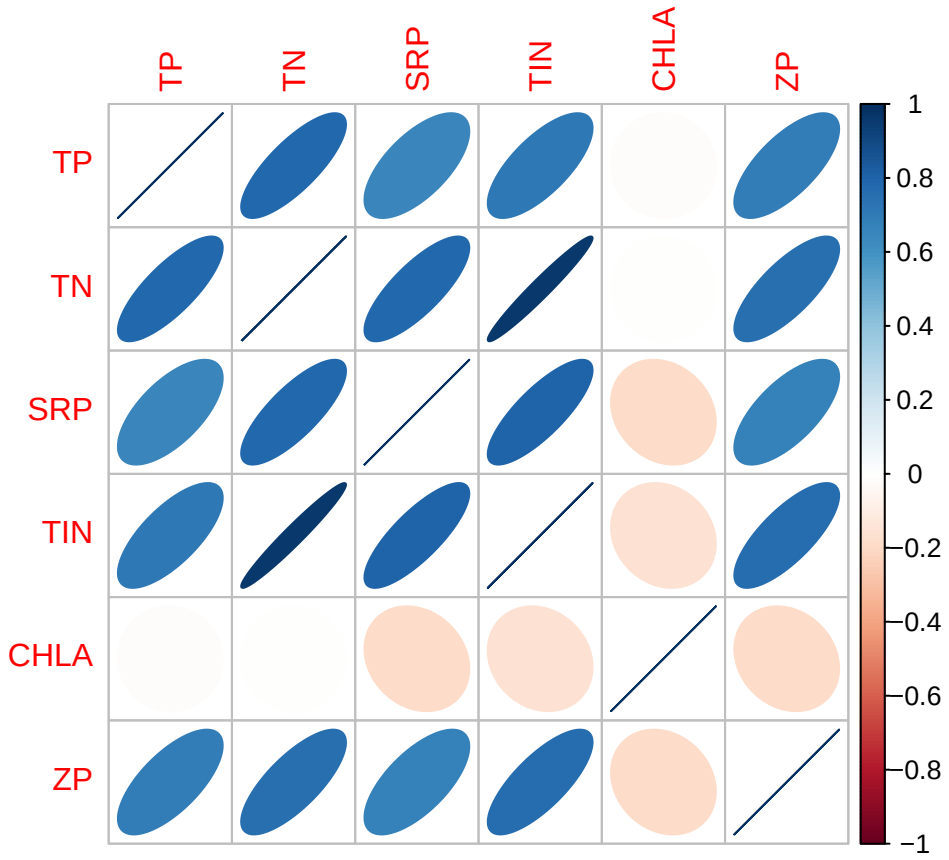
```r
# install.packages("corrplot")
require(corrplot)
```

```
## Le chargement a nécessité le package : corrplot
```

```
## Warning: le package 'corrplot' a été compilé avec la version R 4.4.2
```

```
## corrplot 0.95 loaded
```

```r
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: From the reference of RDocumentation, corr.test uses the cor function to find the correlations, and then applies a t-test to the individual correlations. The results of the outputs are sensitive to the method used (parametric vs. non-parametric). Parametric methods assume that the data are normally distributed and measure linear relationships. Non-parametric methods (e.g., Spearman's correlation) do not assume normality and are more robust for non-linear relationships or data with outliers. One should use non-parametric methods when the data are not normally distributed or when the relationship between variables is not linear. With the Pearson's method, there is evidence for false discovery rate due to multiple comparisons. The corr.test function adjusts for this using methods using BH correction. p-values are significant before adjustment but become non-significant after FDR correction, it indicates evidence of a false discovery rate due to multiple testing. **False discovery rate** is important because it helps to control the proportion of false positives (Type I errors) when multiple hypothesis tests are conducted simultaneously.

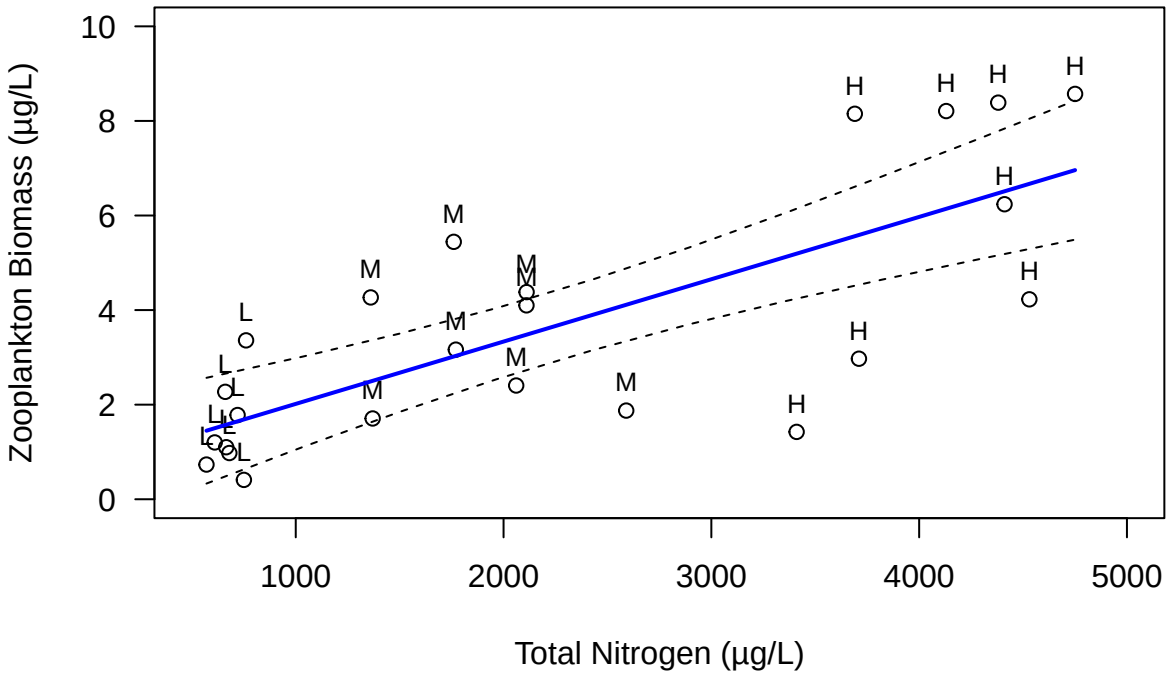
Linear Regression

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
# Linear regression
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

# scatter plot
plot(meso$TN, meso$ZP,
  ylim=c(0, 10),
  xlim=c(500, 5000),
  xlab="Total Nitrogen (µg/L)",
  ylab="Zooplankton Biomass (µg/L)", las=1)

# Add point labels
text(meso$TN, meso$ZP, labels=meso$NUTS, pos=3, cex=0.8)

# create variables for x-axis
newTN = seq(min(meso$TN), max(meso$TN), length.out=10)

# predict values of the variables
regline = predict(fitreg, newdata=data.frame(TN=newTN))


# Make a line that map the values with the regressed line
lines(newTN, regline, lwd=2, col="blue")

# Add confidence intervals
conf95 = predict(fitreg, newdata=data.frame(TN=newTN),
    interval="confidence", level=0.95, type="response")
# print(conf95)
matlines(newTN, conf95[,c("lwr", "upr")], type="l", lty=2, lwd=1, col="black")
```



Question 5: Interpret the results from the regression model

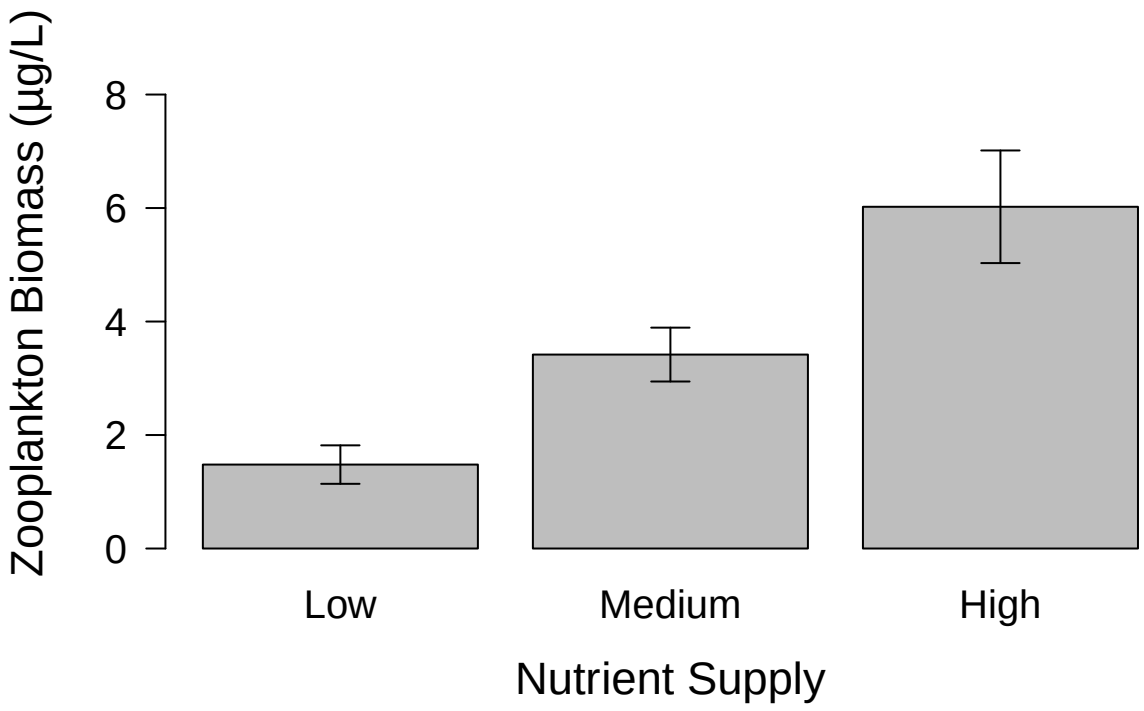
> Answer 5: There is a significant positive relationship between total nitrogen (TN) and zooplankton biomass (ZP) ($p < 0.05$). Also, the model explains 55% of the variance in zooplankton biomass (adjusted R^2 = 0.55). I believe that this relationship is not entirely accurate. If we split data by nutrient *NUTS* variable, there hardly any trend between TN and ZP.

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.

```
NUTS = factor(meso$NUTS, levels=c("L", "M", "H"))
# calculate the mean of zooplancton biomass in each nutrient treatment
zp.means = tapply(meso$ZP, NUTS, mean)

# Function to calculate the sd removing the NA values
sem = function(x) {
  sd(na.omit(x)/sqrt(length(na.omit(x))))
}

# calculate the standard error
zp.sem = tapply(meso$ZP, NUTS, sem)
```

```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 (µg/L)",
  names.arg=c("Low", "Medium", "High"))

# Add error bars
arrows(x0=bp, y0=zp.means-zp.sem,
      x1=bp, y1=zp.means+zp.sem, angle=90, code=3, length=0.1)
```




```r
# ANOVA
anova1 = aov(ZP ~ NUTS, data=meso)
summary(anova1)
```

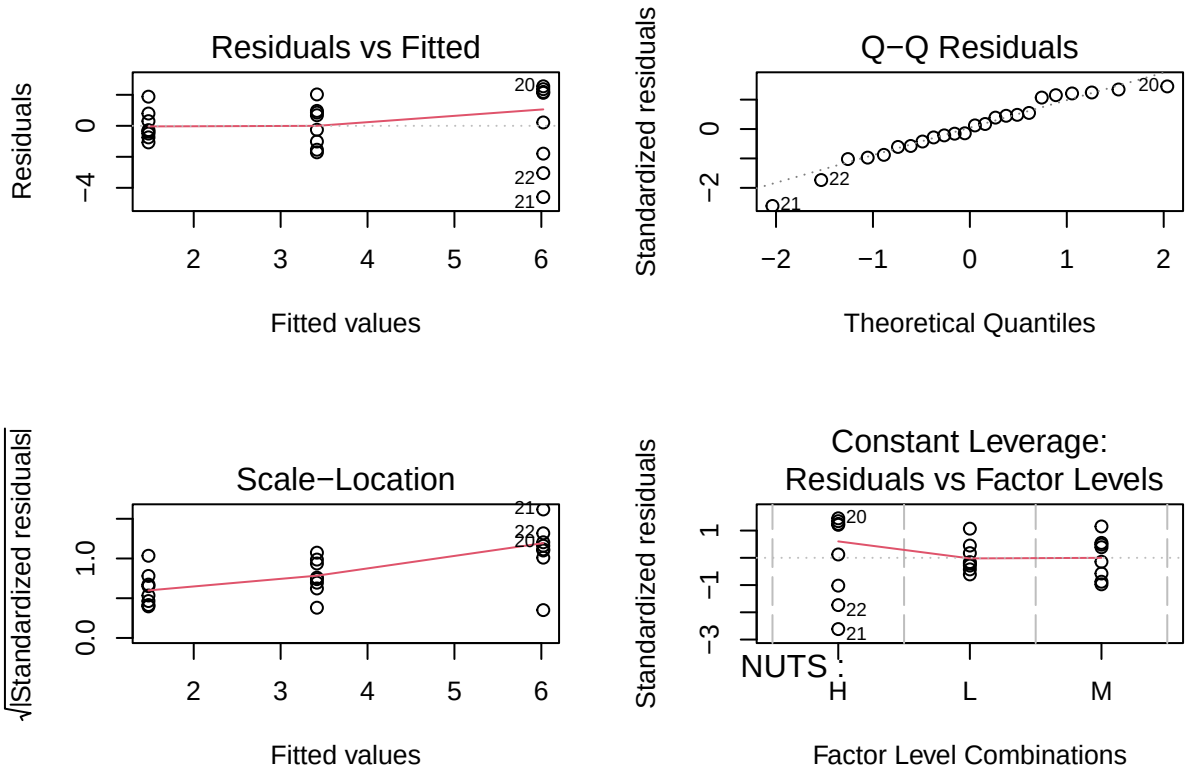
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## NUTS         2  83.15   41.58   11.77 0.000372 ***
## Residuals   21  74.16    3.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Tukey's HSD test
TukeyHSD(anova1)
```

```
##   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
```

```
# Plot the residuals of the ANOVA
par(mfrow=c(2,2), mar=c(5.1, 4.1, 4.1, 2.1))
plot(anova1)
```



SYNTHESIS: SITE-BY-SPECIES MATRIX

In the R code chunk below, load the zoops.txt data set in your **3.RStudio** data folder.

```
# load data
zoops = read.table(paste0(getwd(), "/data/zoops.txt"), sep="\t", header = TRUE)
str(zoops)
```

```
## 'data.frame':    24 obs. of  11 variables:
##  $ TANK: int  5 14 16 21 23 25 27 34 12 15 ...
```

```
##  $ NUTS: chr  "L" "L" "L" "L" ...
##  $ CAL : num  70.5 27.1 5.3 79.2 31.4 22.7 0 35.7 74.8 5.3 ...
##  $ DIAP: num  0 19.2 8.8 17.9 0 ...
##  $ CYCL: num  66.1 129.6 12.7 141.3 11 ...
##  $ BOSM: num  2.2 0 0 3.4 0 0 0 0 0 0 ...
##  $ SIMO: num  417.8 0 73.1 0 482 ...
##  $ CERI: num  159.8 79.4 107.5 199 101.9 ...
##  $ NAUP: num  0 0 1.2 0 0 1.2 1.6 3.1 0 1.4 ...
##  $ DLUM: num  0 0 0 0 0 6.6 0 0 0 0 ...
##  $ CHYD: num  267 159 3158 298 580 ...
```

```
# based on the outputs of str, we see that the data is a data frame with 24 variables and 11 variables

# Check number of unique tanks ( sites)
print(paste("Number of unique tanks:", length(unique(zoops$TANK))))
```

```
## [1] "Number of unique tanks: 24"
```

Create a site-by-species matrix (or dataframe) that does *not* include TANK or NUTS. > Answer: > I'm not sure whether site-by-species matrix means a presence/absence matrix or a matrix with biomass values, so I will create both.

```
# Now create a site-by-species matrix :
# Set 'TANK' as the row names and remove it as a column (after checking that there are exactly 24 tanks
row.names(zoops) <- zoops$TANK
zoops.site_by_species = zoops[, !(names(zoops) %in% c("TANK", "NUTS"))]
zoops.site_by_species
```

```
##      CAL  DIAP  CYCL BOSM   SIMO  CERI NAUP DLUM   CHYD
## 5   70.5   0.0  66.1  2.2  417.8 159.8  0.0  0.0  266.9
## 14  27.1  19.2 129.6  0.0    0.0  79.4  0.0  0.0  158.7
## 16   5.3   8.8  12.7  0.0   73.1 107.5  1.2  0.0 3158.2
## 21  79.2  17.9 141.3  3.4    0.0 199.0  0.0  0.0  298.5
## 23  31.4   0.0  11.0  0.0  482.0 101.9  0.0  0.0  580.2
## 25  22.7 285.1 153.0  0.0  241.5 135.5  1.2  6.6  262.4
## 27   0.0   2.3  11.0  0.0   73.1 185.0  1.6  0.0 2004.4
## 34  35.7  65.9 102.9  0.0    0.0 318.5  3.1  0.0 1260.7
## 12  74.8 178.7 266.5  0.0    0.0   1.9  0.0  0.0 1190.9
## 15   5.3   4.9  87.8  0.0 1099.2 136.4  1.4  0.0 2939.6
## 18  18.4   2.3  29.4  0.0  393.8 147.6  1.2  0.0 4857.3
## 22  14.0   2.3  37.7  0.0 1251.5  74.8  0.0  0.0 2725.5
## 28  14.0   2.3 132.9  0.0  818.6  98.1  1.2  0.0  814.5
## 30  48.8   2.3 107.9  2.2    9.0 132.7  0.0  0.0 2867.5
## 35   0.0   0.0  17.7  0.0  145.3  19.7  0.0  0.0 4201.6
## 36 292.0 269.5 373.4 10.7    0.0   8.5  1.2  0.0 1456.8
## 4    9.7   0.0  41.1  0.0 2397.8   9.4  0.0  0.0 5697.9
## 6    0.0   2.3   0.0  0.0  225.5  24.3  0.0  0.0 8323.2
## 10   5.3   0.0  86.2  0.0  465.9 527.7  1.2  0.0 3146.9
## 11  14.0   7.5  69.5  0.0  594.2  78.5  0.0  0.0 7629.2
## 17   0.0  24.4 101.2  0.0  313.6 176.6  0.0  0.0 7597.6
## 19   0.0   7.5 253.2  8.3    0.0 112.1  1.6  0.0 2594.8
## 24   5.3   2.3  96.2  0.0  786.6  76.6  0.0  0.0  463.0
## 29   0.0   2.3  66.1  0.0  826.7  85.1  0.0  0.0 5263.0
```

```r
# Presence / Absence matrix
zoops.binary = zoops.site_by_species
zoops.binary[zoops.binary  > 0] = 1
zoops.binary
```

```
##    CAL DIAP CYCL BOSM SIMO CERI NAUP DLUM CHYD
## 5    1    0    1    1    1    1    0    0    1
## 14   1    1    1    0    0    1    0    0    1
## 16   1    1    1    0    1    1    1    0    1
## 21   1    1    1    1    0    1    0    0    1
## 23   1    0    1    0    1    1    0    0    1
## 25   1    1    1    0    1    1    1    1    1
## 27   0    1    1    0    1    1    1    0    1
## 34   1    1    1    0    0    1    1    0    1
## 12   1    1    1    0    0    1    0    0    1
## 15   1    1    1    0    1    1    1    0    1
## 18   1    1    1    0    1    1    1    0    1
## 22   1    1    1    0    1    1    0    0    1
## 28   1    1    1    0    1    1    1    0    1
## 30   1    1    1    1    1    1    0    0    1
## 35   0    0    1    0    1    1    0    0    1
## 36   1    1    1    1    0    1    1    0    1
## 4    1    0    1    0    1    1    0    0    1
## 6    0    1    0    0    1    1    0    0    1
## 10   1    0    1    0    1    1    1    0    1
## 11   1    1    1    0    1    1    0    0    1
## 17   0    1    1    0    1    1    0    0    1
## 19   0    1    1    1    0    1    1    0    1
## 24   1    1    1    0    1    1    0    0    1
## 29   0    1    1    0    1    1    0    0    1
```

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.

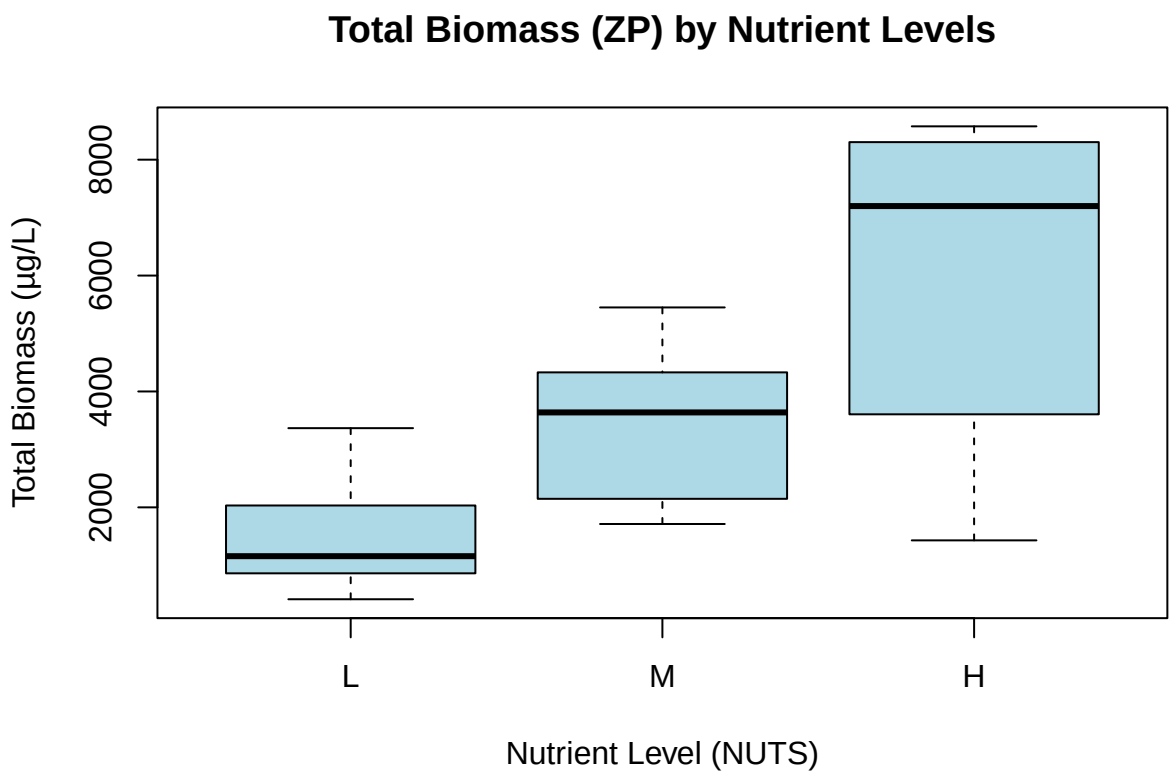
Question 6: With the visualization and statistical tools that we learned about in the **3. RStudio** handout, use the site-by-species matrix to assess whether and how different zooplankton taxa were responsible for the total biomass (ZP) response to nutrient enrichment. Describe what you learned below in the "Answer" section and include appropriate code in the R chunk.

Pre-Answer: Here is how I understood the question and how I solve the question Previously, through ANOVA, we saw how different nutrient enrichments influence the total biomass, but we didnt take into account the fact that each tank has different taxa distribution. In this question, we want to know how each taxa contribute to the total biomass

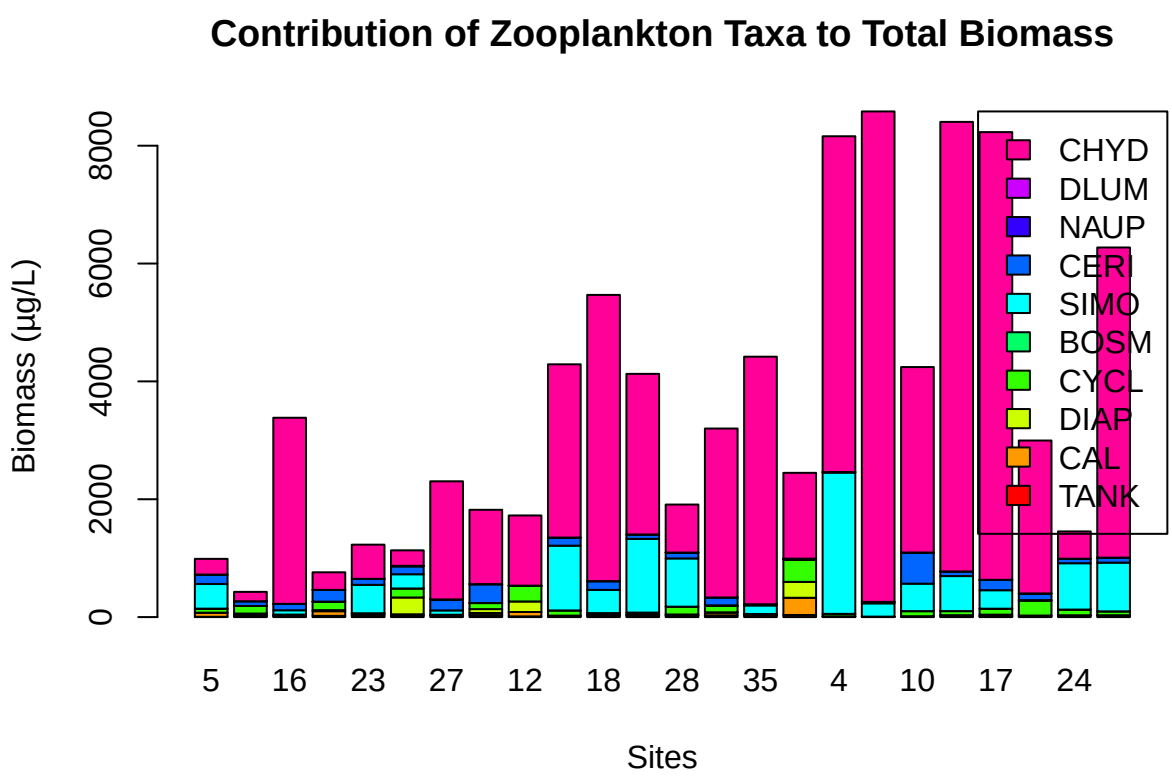
```
# First, let's calculate the total biomass for each tank
zoops$Total_Biomass = rowSums(zoops[, !(names(zoops) %in% c("TANK", "NUTS"))])

# Order category : Low, Med, High nutrient level
zoops$NUTS = factor(zoops$NUTS, levels=c("L", "M", "H"))

# Summarize the data to check differences in Total_Biomass by nutrient levels
# We did this for zoop_nuts but instead of aggregate, we did anova
aggregate(Total_Biomass ~ NUTS, data = zoops, summary)
```

```
##   NUTS Total_Biomass.Min. Total_Biomass.1st Qu. Total_Biomass.Median
## 1    L            414.000               922.300             1157.250
## 2    M           1712.800              2279.475             3638.100
## 3    H           1430.000              3919.275             7199.550
##   Total_Biomass.Mean Total_Biomass.3rd Qu. Total_Biomass.Max.
## 1           1485.263              1909.450           3366.800
## 2           3423.950              4302.025           5450.000
## 3           6027.675              8258.275           8575.300
```

```
# Now, I create a boxplot of total biomass by nutrient level
boxplot(Total_Biomass ~ NUTS, data = zoops,
        main = "Total Biomass (ZP) by Nutrient Levels",
        xlab = "Nutrient Level (NUTS)",
        ylab = "Total Biomass (µg/L)",
        col = "lightblue")
```




```r
# We get what we expected to see, higher nutrient level means higher total biomass.
# ----------------------------------------------------------------------------------

# Now I'll subset the matrix to include only biomass data of taxa
zoops_taxa <- zoops[, !(names(zoops) %in% c("NUTS", "TANKS", "Total_Biomass"))]

# Transpose the data for stacked barplot (taxa as rows) - visually, it gives us an intuition
taxa_matrix <- t(as.matrix(zoops_taxa))

# Colors for taxa
taxa_colors <- rainbow(nrow(taxa_matrix))  # Generate unique colors for each taxon

# Create the stacked bar plot
barplot(
  taxa_matrix,
  beside = FALSE,                    # Stacked bars
  col = taxa_colors,                 # Colors for each taxon
  legend.text = rownames(taxa_matrix),# Add legend for taxa
  args.legend = list(x = "topright"), # Legend position
  xlab = "Sites",                    # X-axis label
  ylab = "Biomass (µg/L)",           # Y-axis label
  main = "Contribution of Zooplankton Taxa to Total Biomass"
)
```




```
## Now, I tackle the question : what is the contribution of each species to the total biomass
# ANOVA to test if total biomass differs by nutrient level
zp_anova <- aov(Total_Biomass ~ NUTS, data = zoops) # copied from previous dataset meso
summary(zp_anova)
```

```
##             Df   Sum Sq  Mean Sq F value   Pr(>F)
## NUTS         2 83123745 41561873   11.77 0.000373 ***
## Residuals   21 74145529  3530739
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# The output shows that the total biomass significantly differs by nutrient level (p < 0.05).
# Linear regression to determine how each taxa contributes to total biomass
lm_model <- lm(Total_Biomass ~ CAL + DIAP + CYCL + BOSM + SIMO + CERI + NAUP + DLUM + CHYD, data = zoops
summary(lm_model)
```

```
## Warning in summary.lm(lm_model): ajustement pratiquement parfait : le résumé
## n'est peut-être pas fiable
```

```
##
## Call:
## lm(formula = Total_Biomass ~ CAL + DIAP + CYCL + BOSM + SIMO +
##     CERI + NAUP + DLUM + CHYD, data = zoops)
##
```

```
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.508e-12 -3.867e-13  2.128e-13  6.789e-13  1.188e-12
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 7.136e-13  9.887e-13 7.220e-01    0.482
## CAL         1.000e+00  1.523e-14 6.566e+13   <2e-16 ***
## DIAP        1.000e+00  1.613e-14 6.198e+13   <2e-16 ***
## CYCL        1.000e+00  8.541e-15 1.171e+14   <2e-16 ***
## BOSM        1.000e+00  2.582e-13 3.873e+12   <2e-16 ***
## SIMO        1.000e+00  5.361e-16 1.865e+15   <2e-16 ***
## CERI        1.000e+00  3.194e-15 3.131e+14   <2e-16 ***
## NAUP        1.000e+00  4.550e-13 2.198e+12   <2e-16 ***
## DLUM        1.000e+00  5.976e-13 1.673e+12   <2e-16 ***
## CHYD        1.000e+00  1.391e-16 7.190e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.224e-12 on 14 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.167e+31 on 9 and 14 DF,  p-value: < 2.2e-16
```

> Answer : For the ANOVA model of effect of nutrient enrichment on total biomass. The p-value of F test is $0.000373 < 0.05$, meaning that nutrient levels (NUTS) significantly affect total biomass.
>
> For the linear regression model of different taxa on total biomass: if we look at the intercept, which is the fixed effect contribution beyond the taxa. The p-value is 0.482, so not significant. Then, when we look and the coefficients of all taxa, we see that all of them have coefficients around 1, which means that their biomass contributes equally and linearly, and all the coefficients are significant. Finally, if we look at the explanatory power of this current model: we can see that R square is 1.0, which means that the model perfectly explains the total biomass, this was accompagnied with a low p value of F stats, which indicates that the model is significant.

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 22nd, 2025 at 12:00 PM (noon)**.