3. Worksheet: Basic R

Carter K Stancil; Z620: Quantitative Biodiversity, Indiana University

16 January, 2026

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. We recommend that you work through this Worksheet on the computer with your Handoutopen in your binder.

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 may want to **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 Posit.cloud workspace: `/cloud/project/QB-2025/Week1-RStudio/`
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 `Week1-RStudio/` folder.

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

```
## [1] "C:/github/QB2026_Stancil/QB2026/Week1-RStudio"
```

```
setwd("../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.

```
#volume of a cube
l=5
cube_volume = l^3
#area of a circle
r=2
circle_area = pi*(r^2)
#length of the opposite side of a right-triangle
theta = pi/4
hypotenuse = sqrt(2)
opposite = sin(theta)*hypotenuse
#log (base e) of my favorite number
fave = 21
log(fave)
```

```
## [1] 3.044522
```

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(6, 5, 4, 2.5, 2.5)
w = x*14
(x + w)/15
```

```
## [1] 6.0 5.0 4.0 2.5 2.5
```

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`.

```
k = c(57, 54, 26, 24, 18)
k*x
```

```
## [1] 342 270 104  60  45
```

```
d = c(w[c(1,3,4)], k[c(1,2,4,5)])
```

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_noNA <- na.omit(v)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
summary(v_noNA)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.10   16.50   20.35   20.90   24.95   31.40
```

```r
#statistics not covered by dplyr::summary
sum(v_noNA)
```

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

```r
var(v_noNA)
```

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

```r
sd(v_noNA)
```

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

```r
#write function for standard error of the mean
sem <- function(x){
  sd(x)/sqrt(length(x))
}

sem(v_noNA)
```

```
## [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.

```r
col_1 <- c(rnorm(5, mean = 8, sd = 2))
col_2 <- c(rnorm(5, mean=25, sd = 10))
mat <- matrix(c(col_1,col_2), nrow=5, ncol=2, byrow=FALSE)
```

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 generates a set of random numbers that are normally distributed according to the mean and standard deviation that the user designates in the parameters.

In the R code chunk below, do the following: 1) Load `matrix.txt` from the **Week1-RStudio** data folder as matrix `m`. 2) Transpose this matrix. 3) Determine the dimensions of the transposed matrix.

```r
m <- read.table("matrix.txt", sep = "\t", header = FALSE)
m <- t(m)
str(m)
```

```
##  int [1:5, 1:10] 8 1 7 6 1 5 5 2 4 1 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:5] "V1" "V2" "V3" "V4" ...
##   ..$ : NULL
```

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

> Answer 2: The dimensions of the matrix are 5 by 10 (rows by 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(1:2,4:10)]
```

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

```
m[,1:9]
```

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

6) BASIC DATA VISUALIZATION AND STATISTICAL ANALYSIS

Load Zooplankton Data Set

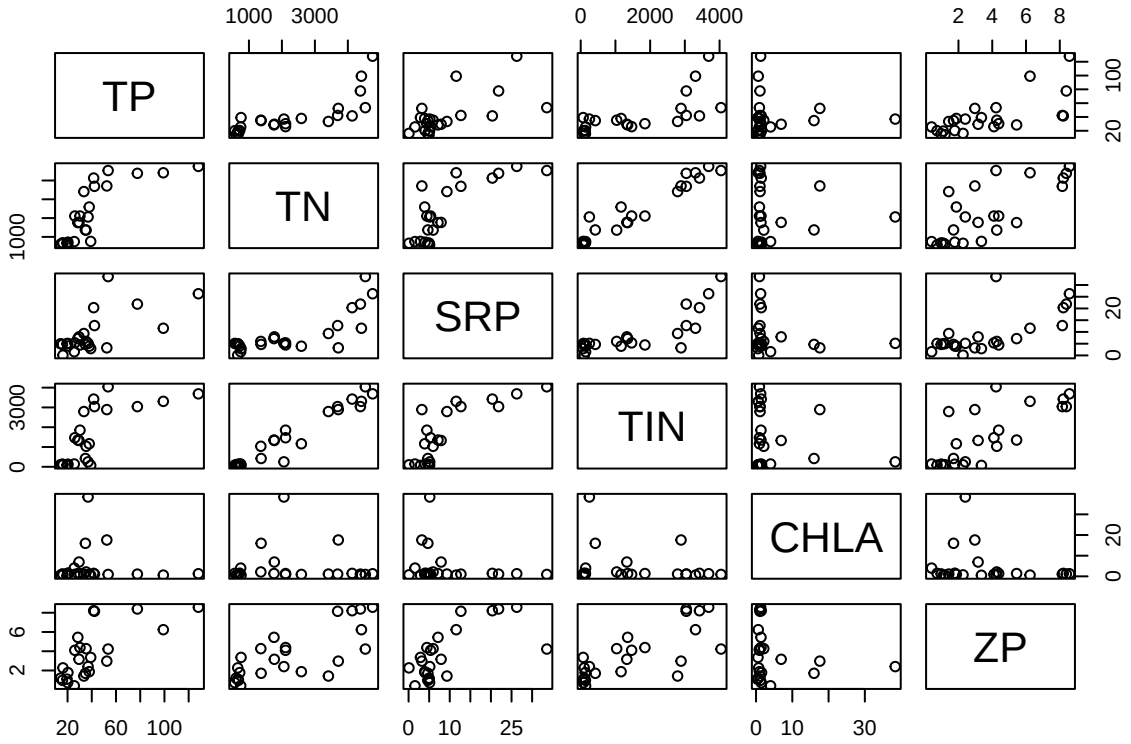
In the R code chunk below, do the following: 1) Load the zooplankton data set from the **Week1-RStudio/** data folder. 2) Display the structure of this data set.

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

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)
```

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

> Answer 3: Total nitrogen (TN) seems to be strongly positively correlated with the total inorganic nutrient concentration (TIN). Chlorophyll A concentration appears to be weakly negatively correlated with all other meatured variables. Generally, zooplankton seems to be strongly correlated with total phosphorous concentration, nitrogen concentration, soluble reactive phosphorous concentration, and total inorganic nutrient concentration (all positively).

In the R code chunk below, do the following: 1) Redo the correlation analysis using the `corr.test()` function in the `psych` package with the following options: method = "pearson", adjust = "BH". 2) Now, redo this correlation analysis using a non-parametric method. 3) Use the print command from the handout to see the results of each correlation analysis.

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

```
## Warning: package 'psych' was built under R version 4.5.2
```

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

```
## Call:corr.test(x = meso.num, method = "pearson", adjust = "BH")
```

```
## Correlation matrix
##         TP   TN   SRP   TIN  CHLA    ZP
## TP    1.00 0.79  0.65  0.72 -0.02  0.70
## TN    0.79 1.00  0.78  0.97  0.00  0.76
## SRP   0.65 0.78  1.00  0.80 -0.19  0.68
## TIN   0.72 0.97  0.80  1.00 -0.16  0.76
## CHLA -0.02 0.00 -0.19 -0.16  1.00 -0.18
## ZP    0.70 0.76  0.68  0.76 -0.18  1.00
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##        TP   TN  SRP  TIN CHLA   ZP
## TP   0.00 0.00 0.00 0.00 0.98 0.00
## TN   0.00 0.00 0.00 0.00 0.98 0.00
## SRP  0.00 0.00 0.00 0.00 0.49 0.00
## TIN  0.00 0.00 0.00 0.00 0.54 0.00
## CHLA 0.94 0.98 0.38 0.46 0.00 0.49
## ZP   0.00 0.00 0.00 0.00 0.39 0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
corr.test(meso.num, method = "spearman")
```

```
## Call:corr.test(x = meso.num, method = "spearman")
## Correlation matrix
##        TP   TN   SRP  TIN  CHLA    ZP
## TP   1.00 0.89  0.54 0.76  0.04  0.74
## TN   0.89 1.00  0.65 0.94  0.02  0.75
## SRP  0.54 0.65  1.00 0.73 -0.06  0.63
## TIN  0.76 0.94  0.73 1.00  0.09  0.74
## CHLA 0.04 0.02 -0.06 0.09  1.00 -0.07
## ZP   0.74 0.75  0.63 0.74 -0.07  1.00
## Sample Size
## [1] 24
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##        TP   TN  SRP  TIN CHLA   ZP
## TP   0.00 0.00 0.04 0.00 1.00 0.00
## TN   0.00 0.00 0.01 0.00 1.00 0.00
## SRP  0.01 0.00 0.00 0.00 1.00 0.01
## TIN  0.00 0.00 0.00 0.00 1.00 0.00
## CHLA 0.85 0.92 0.77 0.68 0.00 1.00
## ZP   0.00 0.00 0.00 0.00 0.74 0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
cor2 <- corr.test(meso.num, method = "pearson", adjust = "BH")
cor3 <- corr.test(meso.num, method = "spearman")
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
```

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

```
## Call:corr.test(x = meso.num, method = "spearman")
## 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 0.039 0.000 1.000 0.000
## TN   0.000 0.000 0.005 0.000 1.000 0.000
## SRP  0.007 0.001 0.000 0.001 1.000 0.007
## TIN  0.000 0.000 0.000 0.000 1.000 0.000
## CHLA 0.853 0.923 0.767 0.683 0.000 1.000
## 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: Because the values are quite similar between `cor2` and `co3`, the data do not appear to be sensitive to parametric vs. non-parametric methods.
> Parametric methods must make assumptions about the distribution of data from the population, most commonly that the population is normally distributed. Non-parametric tests do not assume the shape of distribution of values in the population. However, non-parametric tests are often less statistically powerful than parametric methods when analyzing normally distributed data. If the

population varies signficantly from the assumptions made by the parametric method, the data should be analyzed using a non-parametric method, so that we do not draw incorrect or false conclusions. With the Pearson's method, there is evidence for false discovery rate (FDR) due to multiple comparisions. Due to the nature of the variables being compared to one another, some are likely already correlated with one another. This is common in ecological studies. For example, running a Pearson's test while including variables such as Temperature, Day of Year, Average Sunlight, etc. may provide many false positives because those factors themselves are strongly correlated. False discovery rate is important because it means that more discover significant results, while assuming that a small percentage of them will likely be false.

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.

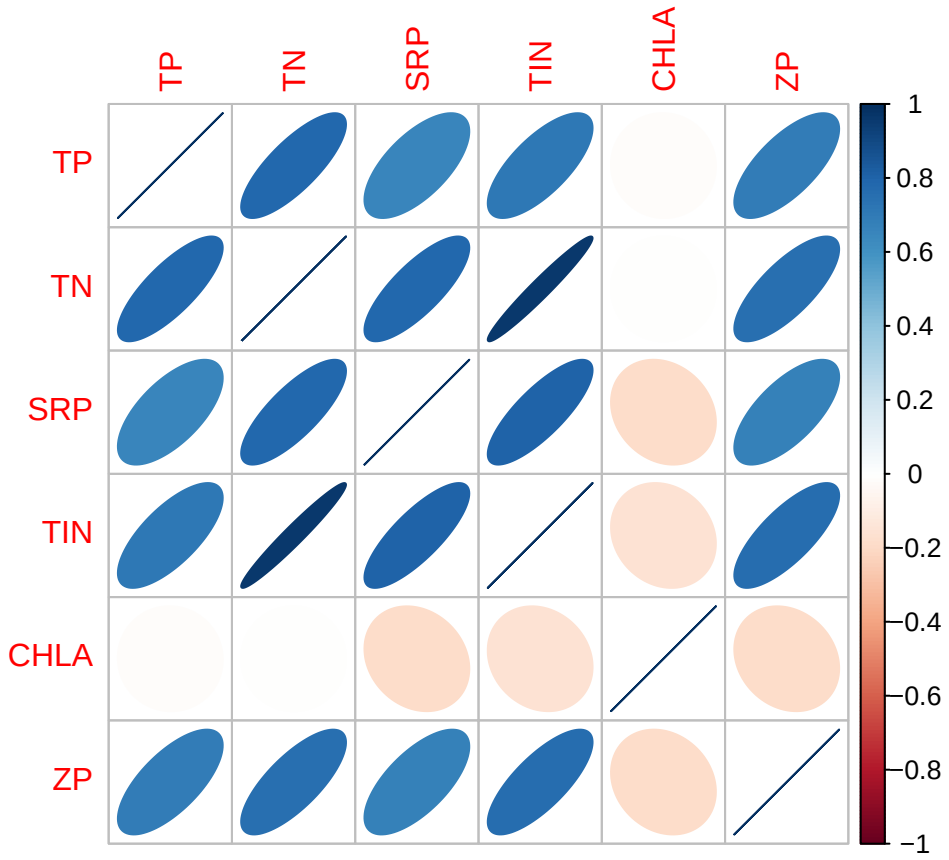
```
#install.packages("corrplot", repos="http://cran.rstudio.com/")
require("corrplot")
```

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

```
## Warning: package 'corrplot' was built under R version 4.5.2
```

```
## corrplot 0.95 loaded
```

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




```
dev.off()
```

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

```
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
```

```
newTN <- seq(min(meso$TN), max(meso$TN), 10)
regline <- predict(fitreg, newdata = data.frame(TN = newTN))
conf95 <- predict(fitreg, newdata = data.frame(TN = newTN),
                  interval = c("confidence"), level = 0.95, type = "response")

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)
```

```
## integer(0)
```

Question 5: Interpret the results from the regression model

> Answer 5: The p-value is <0.05, which indicates a significant correlation between the total nitrogen concentration and zooplankton biomass. Because the coefficients are positive (and by looking at the graph), we know that this correlation is positive.

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.

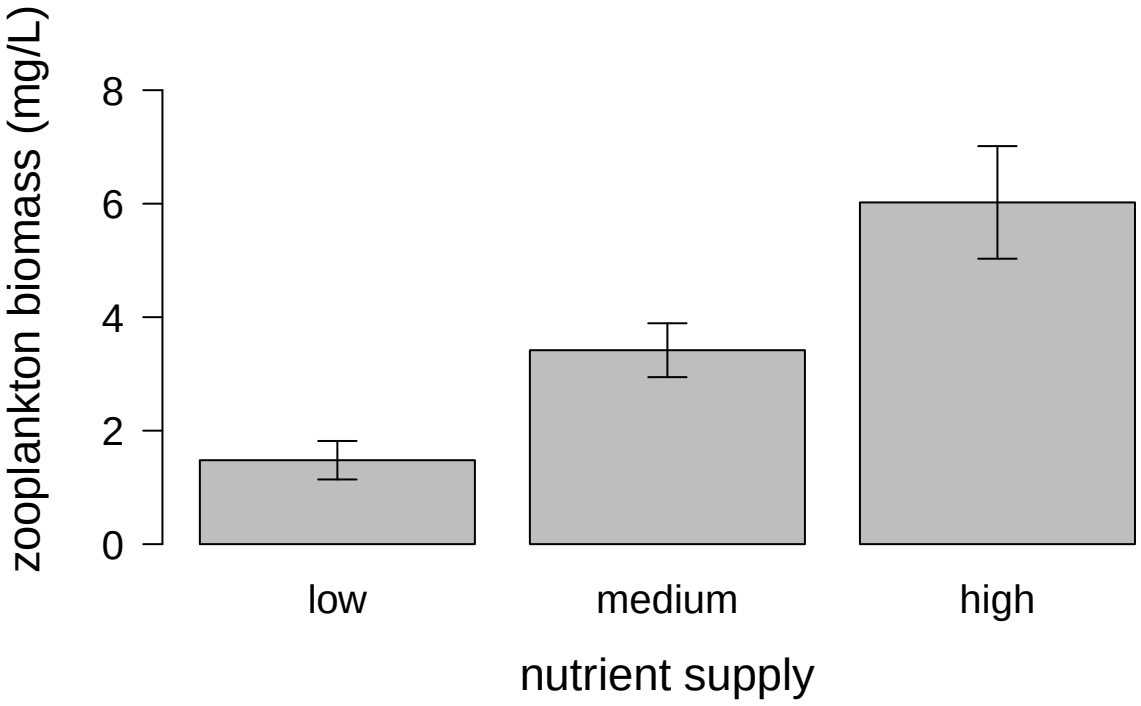
```
NUTS <- factor(meso$NUTS, levels = c('L', 'M', 'H'))

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)
```




```
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
```

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.

> Answer 6: The barplot that I made is a bit difficult to follow, but the *Ceriodaphnia* sp. are commonly present and seem to be very abundant at each site. *Bosmina* sp. were more rare, and were only found at a handful of sites. Calnoid copepods are also common and were present at almost every site.

```
zoops <- read.table("zoops.txt")
zoops_trim <- zoops[,-c(1,2)]
zoops_trim_df <- as.data.frame(zoops_trim)
rownames(zoops_trim[1,]) <- rownames(zoops_trim_df[1,])
colnames(zoops_trim_df) <- zoops_trim_df[1, ]
# Remove the first row (since it's now column names)
zoops_trim_df <- zoops_trim_df[-1, ]

zoops_matrix <- as.matrix(zoops_trim_df)

library(dplyr)
zoops_trim_df <- zoops_trim_df %>%
  mutate(Site = c(1:24))

library(ggplot2)
```

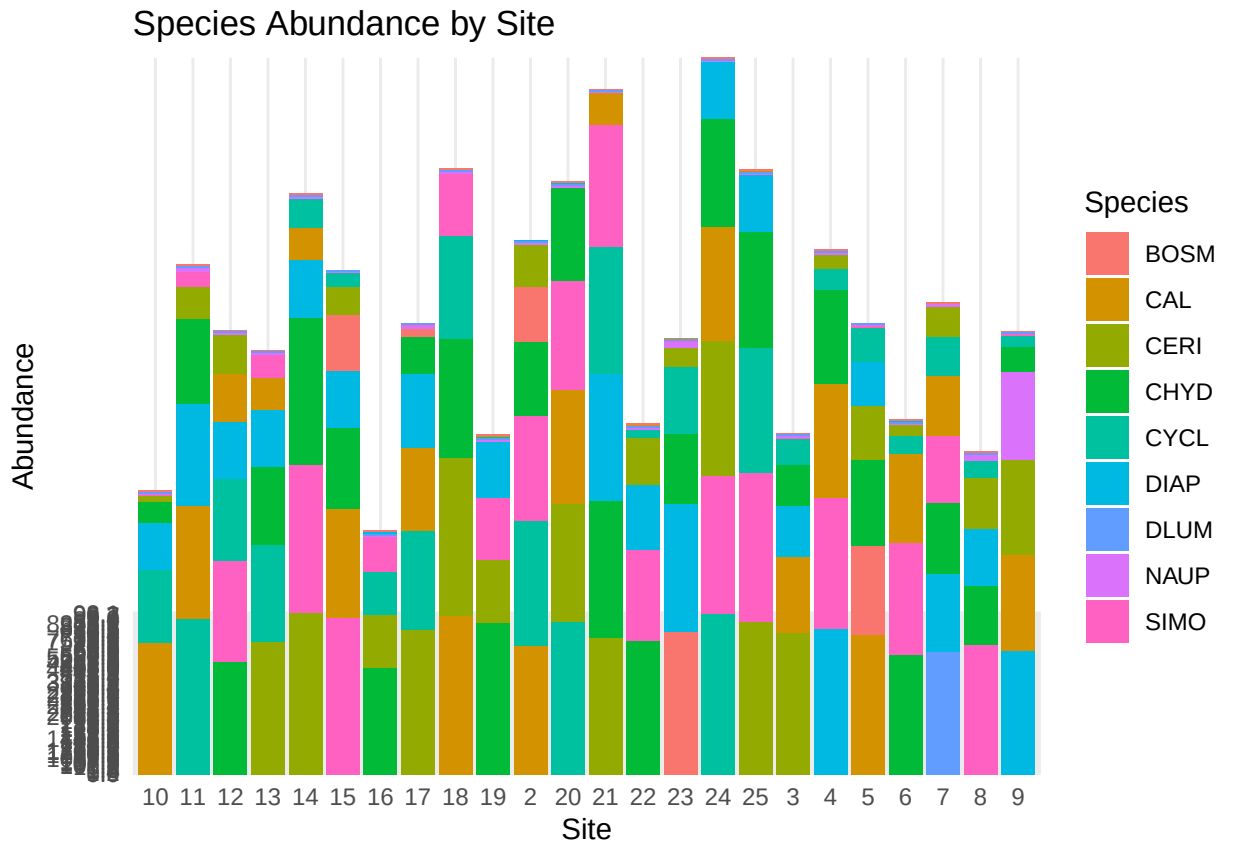
```
##
## Attaching package: 'ggplot2'
```

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

```
library(tidyr)

df_long <- as.data.frame(zoops_trim_df) %>%
  mutate(Site = rownames(.)) %>%
  pivot_longer(-Site, names_to = "Species", values_to = "Abundance")

ggplot(df_long, aes(x = Site, y = Abundance, fill = Species)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Species Abundance by Site")
```



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, but *do not* self-merge it. In a collaboration when using github, you will often assign a pull request to *another* person. This alerts your collaborator to the proposed changes and allows them to look them over before accepting them. To formally **assign** a pull request to someone, you will see an option on the right hand side to choose a reviewer or a assignee. If you click on the gear icon, a list of collaborators will come up. Your primary instructor (`jaytlennon`) should be one of those on the QBstudent organization. Click on his username to assign before you finalize the 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)**.