Tested Studies for Laboratory Teaching
Proceedings of the Association for Biology Laboratory Education
Vol. 36, Article 21, 2015

Integrating the Software Package "R" in Skill-based Introductory Biology Labs to Enhance Student Graphing Skills

Michael S. Berger

Washington State University, School of the Environment, 14204 NE Salmon Creek Ave., Vancouver WA 98686 USA
(**msberger@vancouver.wsu.edu**)

Extended Abstract

Graphs are used in science as a visual mechanism to efficiently and concisely communicate complex information. Undergraduate students frequently lack basic skills involving the visual representation of data, such as graphing. Introductory biology laboratories can be developed to utilize the software package "R" as a learning tool to help students improve their graphing skills. The software package "R" is an open source script-based software package that can be used with a graphical user interface, "R Commander." The use of a script based software removes the "back-box" effect that can limit conceptual understanding of the components involved in the process of visually representing data. Students are introduced to "R" in lab and provided with step-by-step tutorials to guide them through the process of generating graphs. Learning goals can be focused on: (1) understanding components used to construct an informative graph; (2) competency in the visual display of data; (3) learning how to use the software package "R."

To download and install "R", visit the R-Project website at http://cran.r-project.org/ and follow instructions for downloading. Because "R" is command based, you will want to use a Graphical User Interface called "R Commander", which allows you to point and click, compared to typing in commands. To install "R Commander", open "R" and type install.packages() and then press enter. Scroll through the packages until you find a series of packages labeled Rcmdr. Highlight all of these packages and press enter. Choose a site to download them from; I suggest USA (CA1). When you run "R Commander" for the first time, you may be prompted to install more packages; click on yes to install the packages.

How to Construct a Line Plot Using Student Generated Data (See Table 1)

The following set of instructions can be used to develop a step-by-step tutorial to guide students through the process of learning how to graph in "R", using the Graphical User Interface "R Commander".

1A. Enter data into a spreadsheet, such as Microsoft Excel. You will want columns for each pigment absorption spectra (dependent variables) and one column for wavelength (independent variable).

1B. Save the data as a comma delimited file (.csv). File > Save as. In the "Save as type box", scroll down and click on "CSV (Comma delimited) (*.CSV)", name your file, and click on "Save". Be sure to use a name and location you can find!

2. Open R and type library(Rcmdr) at the prompt. Press enter. "R" is syntax specific, so commands must be entered with the correct syntax. For example, if you use a lower case "r" in the command "library(rcmdr)", the command will not work.

3. Import the data file into "R." Data > Import data > "from text file, clipboard, or URL". Enter the name pigment for the data set. Click on "Commas" in the "Field separator" section. Click OK. Open your saved .csv file in the dialog box.

4. Graph the absorption spectra data as a line plot. The script used to construct your line plot is listed below. Each line of script is followed by an explanation. Use the supplemental script text file to copy script into the "R Script" window in "R Commander." All of the variables have been included in the text file. The "#" symbol is used in "R" to denote text that is not part of the script. Text following the '#" symbol is not used by "R" and is included to explain the script components.

When you are ready to make your plot, highlight all of the copied and pasted command script. Click on "Submit". Your graph will be visible in the "R" window.

plot(Y-axis variable~X-axis variable, data=name of data set, type=1"l", lty=1, lwd=2, col="green", xlab="x-axis label", ylab="y-axis label")

plot is the command to make a line or symbol plot based on an x-y coordinate(s)
Y-axis variable~X-axis variable tells "R" which variables to plot. You need to enter the appropriate name from your data set; hint, click on "View data set" in R Commander to view names of data headers.

type="l" specifies a line plot without symbols.
col="green" results in a green line.
lty=1 specifies a solid line.
lwd=2 specifies the line width
xlab and ylab code for x-axis and y-axis labels.

lines(Y-axis variable~X-axis variable, data=name of data set, type="l", lty=1, lwd=2, col="blue")

This script adds another line to your plot. You can use this command to plot the remaining pigment data. In other words, repeat this script for each additional pigment. Make sure you change the y-axis variable name to match the data you want to plot. You can change the color (col="color") to the color of the pigment being graphed.

legend("top", c("Chl-a", "Chl-b", "Xanthophyll", "Carotene", "Total pigment"), lty=c(1,1,1,1,1), lwd=c(2.5,2.5,2.5,2.5,2.5), col=c("green","blue","yellow","orange","black"))
# legend is the command to adds a legend to the plot
# "top" indicates the location ("topright" or "topleft" could also be used) and c() indicates the legend text
# lty=c(1,1,1,1,1) specifies a solid line type for each variable in the plot
# lwd=c(2.5,2.5,2.5,2.5,2.5) specifies the line width for each symbol
# col=c("green","blue","yellow","orange","black") specifies the line color for each symbol

5. Save your graph. In the "R" window where your graph is located, click on your graph. In the upper left of the "R" window, click on File > Save as. Choose a file type and save your graph. I suggest either a PDF or jpeg.

Table 1. Example of absorption spectral data.

Wavelength	Chlorophyll-a	Chlorophyll-b	Xanthophyll	Carotene	Total Pigment
400	1.140	0.348	0.101	0.128	0.332
420	1.500	0.470	0.141	0.146	0.353
440	1.000	0.730	0.154	0.139	0.550
460	0.452	0.965	0.102	0.138	0.764
480	0.241	0.792	0.080	0.111	0.895
500	0.100	0.704	0.016	0.063	0.880
520	0.093	0.640	0.009	0.038	0.794
540	0.102	0.468	0.007	0.041	0.616
560	0.132	0.293	0.008	0.040	0.414
580	0.167	0.150	0.009	0.039	0.244
600	0.206	0.087	0.009	0.036	0.161
620	0.252	0.150	0.010	0.037	0.185
640	0.305	0.340	0.010	0.036	0.322
660	0.760	0.622	0.010	0.045	0.632
680	0.300	0.792	0.008	0.039	0.885
700	0.068	1.055	0.004	0.039	1.250
720	0.101	1.480	0.005	0.042	1.660

Mission, Review Process & Disclaimer

The Association for Biology Laboratory Education (ABLE) was founded in 1979 to promote information exchange among university and college educators actively concerned with teaching biology in a laboratory setting. The focus of ABLE is to improve the undergraduate biology laboratory experience by promoting the development and dissemination of interesting, innovative, and reliable laboratory exercises. For more information about ABLE, please visit **http://www.ableweb.org/**

Papers published in *Tested Studies for Laboratory Teaching: Peer-Reviewed Proceedings of the Conference of the Association for Biology Laboratory Education* are evaluated and selected by a committee prior to presentation at the conference, peer-reviewed by participants at the conference, and edited by members of the ABLE Editorial Board.

Citing This Article

Berger, M. S. 2015. Integrating the Software Package "R" in Skill-based Introductory Biology Labs to Enhance Student Graphing Skills. Pages Article 21 in *Tested Studies for Laboratory Teaching,* Volume 36 (K. McMahon, Editor). Proceedings of the 36th Conference of the Association for Biology Laboratory Education (ABLE). http://www. ableweb.org/volumes/vol-36/?art=21

Appendix

Script to create a line plot with multiple lines

```
#LINEPLOT FUNCTION

plot(Chlorophyll_a~Wavelength, data=pigment, type="l", lty=1, lwd=2,# Plot command to make a line plot for one X-Y variable col="green", xlab="x-axis label", ylab="y-axis label")

#ADD ADDITIONAL LINES

lines(Chlorophyll_b~Wavelength, data=pigment, type="l", lty=1, lwd=2,# Adds a second line col="blue")

lines(Xanthophyll~Wavelength, data=pigment, type="l", lty=1, lwd=2, # Adds a third line col="yellow")

lines(Carotene~Wavelength, data=pigment, type="l", lty=1, lwd=2, # Adds a fourth line col="orange")

lines(Total_pigment~Wavelength, data=pigment, type="l", lty=1, lwd=2, # Adds a fifth line col="black")
```

```
#ADD A LEGEND

legend("top", c("Chl-a","Chl-b","Xanthophyll","Carotene", "Total pigment"),      # Adds a legend to the plot

lty=c(1,1,1,1,1,1), # Assigns a line for the legend symbol

lwd=c(2.5,2.5,2.5,2.5,2.5),col=c("green","blue","yellow","orange","black")) # Assigns line width and color

#Script to create a bar graph

# MEAN VALUES

low = barplot$Low_light # tells "R" that the term low is the data for the low light treatment.

high = barplot$High_light # tells "R" that the term high is the data for the high light treatment.

mean(low) # calculates a mean value for low that is displayed in the output window

mean(high) # calculates a mean value for high that is displayed in the output window

means <- c(mean(low), mean(high)) # designates the term means to represent the average (mean) for low and high

# GRAPH

growth <- barplot(means, # creates a barpot for the mean  values of low and high

xlab="x-axis label", # x-axis label

ylab="y-axis label", # y-axis label

ylim=c(0,100), # sets the y-axis range from 0 to 100

col="gray") # bar color

# X-AXIS

axis(1, labels=c("Low light", "High light"), at = growth)# adds an x-axis with ticks and labels

box() #adds a box around your graph

# ERROR BARS (standard error of the mean (SEM)

SEM <- c(sd(low/sqrt(5)), sd(high/sqrt(5))) # designates the term SEM to represent the standard error of the mean (SEM) for
      low and high

segments(growth, means - SEM, growth, means + SEM, lwd=2) # plots vertical error bars; lwd=2 is the line width

segments(growth - 0.1, means - SEM, growth + 0.1, means - SEM, lwd=2) # plots the lower horizontal bar

segments(growth - 0.1, means + SEM, growth + 0.1, means + SEM, lwd=2) # plots the upper horizontal bar
```