

R wizardry Homework 5 (week 6), 2017

After doing some analyses, your lab mates keep bugging you to help them do the same thing for their data. Rather than having to do it over and over, you decide it would be easier to make some functions so they can do it themselves. Convert the following code into three functions (only the first 2 are required for this assignment). See the 2nd page for specifics on what you would code if you did this just for your specified data, then write a generalizable function that would work with any dataset.

Question 1 (4 points)

`zScore()`

This function will take in a vector of numbers and converts them to a vector of z-scaled numbers. You should be able to run `boxplot(zScore(data$biomass))` to recreate figure 1.

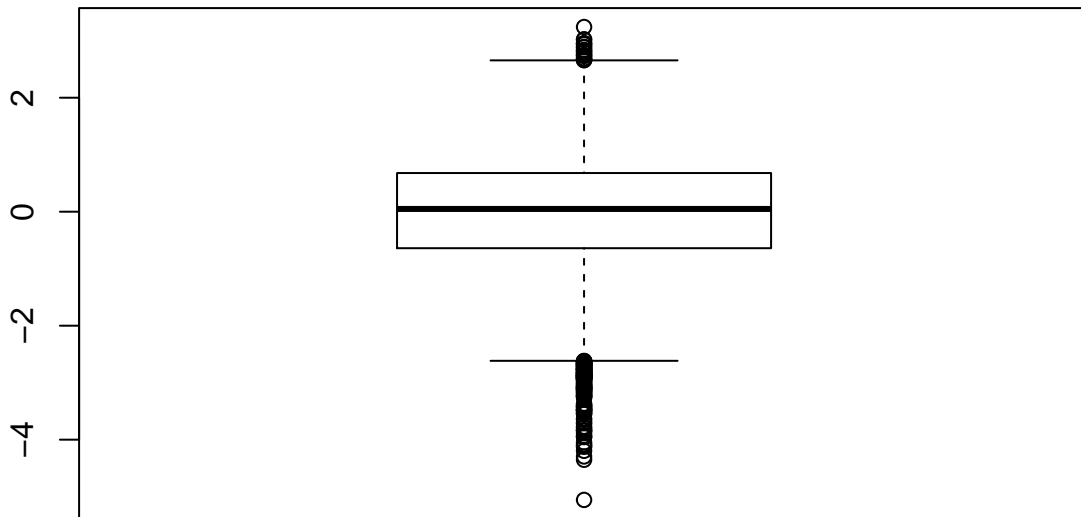


Figure 1: Output of `'boxplot(zScore(data$biomass))'`.

Question 2 (6 points)

`simplePlot()`

This function will take in column names for the x and y axes, a data.frame for the dataset and a value for the colour and width of the trend line. It also takes in, but doesn't require, any other parameters you want to pass to the plot function (remember the parameter `"..."`). It produces a plot of the data with a trend line overlaid.

You should be able to run

`'simplePlot(x = "day", y = "biomass", data = data, lineCol = 2, lwd = 2, pch = 16, col = 3)'` to recreate figure 2, and

`'simplePlot(x = "day", y = "biomass", data = data, lineCol = 2, lwd = 2, pch = 2)'` to recreate figure 3 (provided you make sure the parameter names match what you have used).

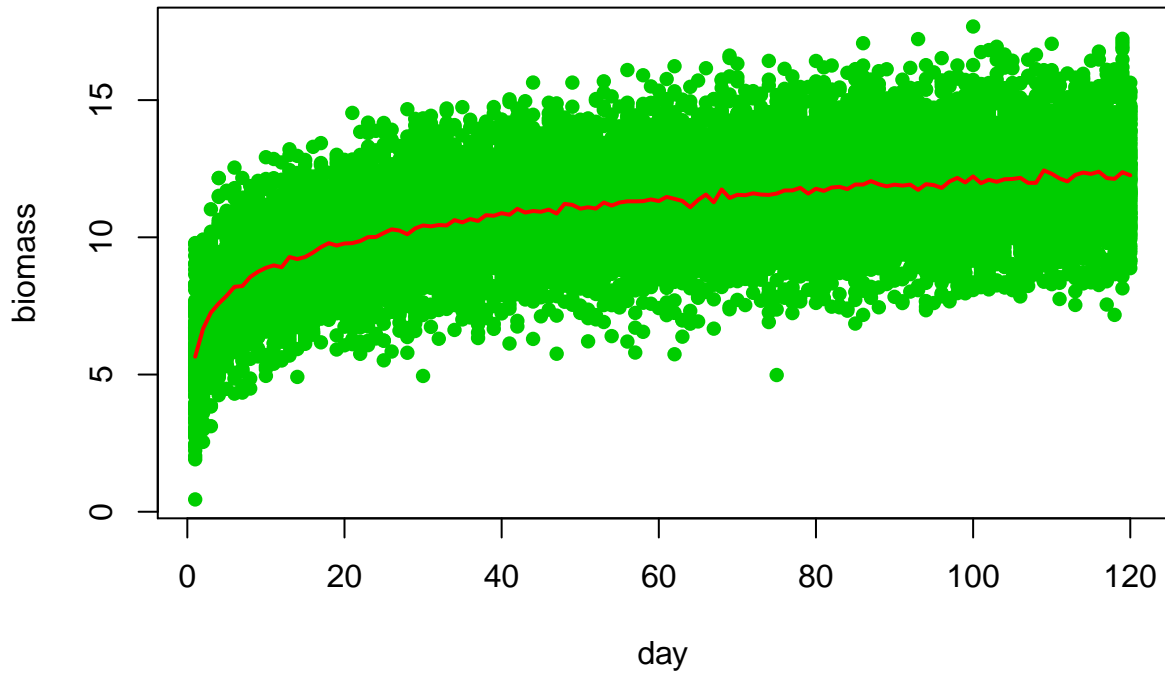


Figure 2: Output of `'simplePlot(x = 'day', y = 'biomass', data = data, lineCol = 2, lwd = 2, pch = 16, col = 3)'`.

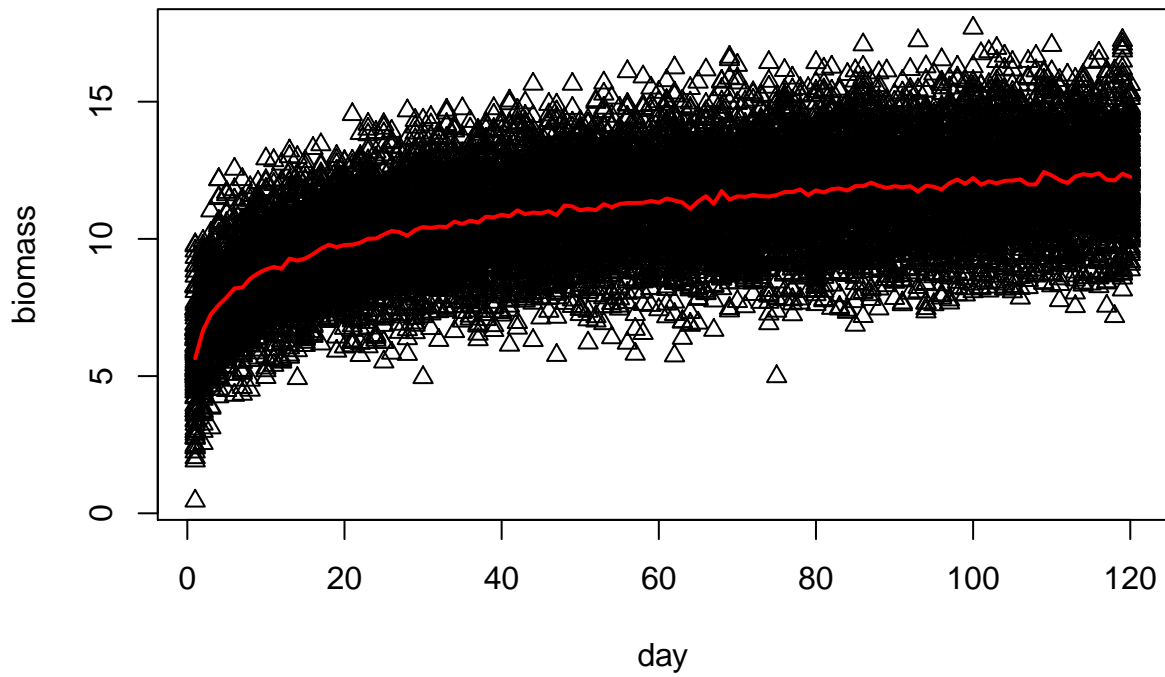


Figure 3: Output of `'simplePlot(x = 'day', y = 'biomass', data = data, lineCol = 2, lwd = 2, pch = 2)'`.

Extra challenge (3 points)

`stackedPlot()`

This function will take all the same parameters as `simplePlot()` except instead of a single vector for `y`, it will take `y1` and `y2`, using `y1` as `y` for the first plot and `y2` as `y` for the second plot.

Hint, using the parameter “...” to its full potential and using `simplePlot()` could make writing this function very simple.

You should be able to run “`stackedPlot(x = "day", y1 = "biomass", y2 = "count", data = data, lineCol = 2, lwd = 2, pch = 16, col = 4)`” to recreate Figure 4.

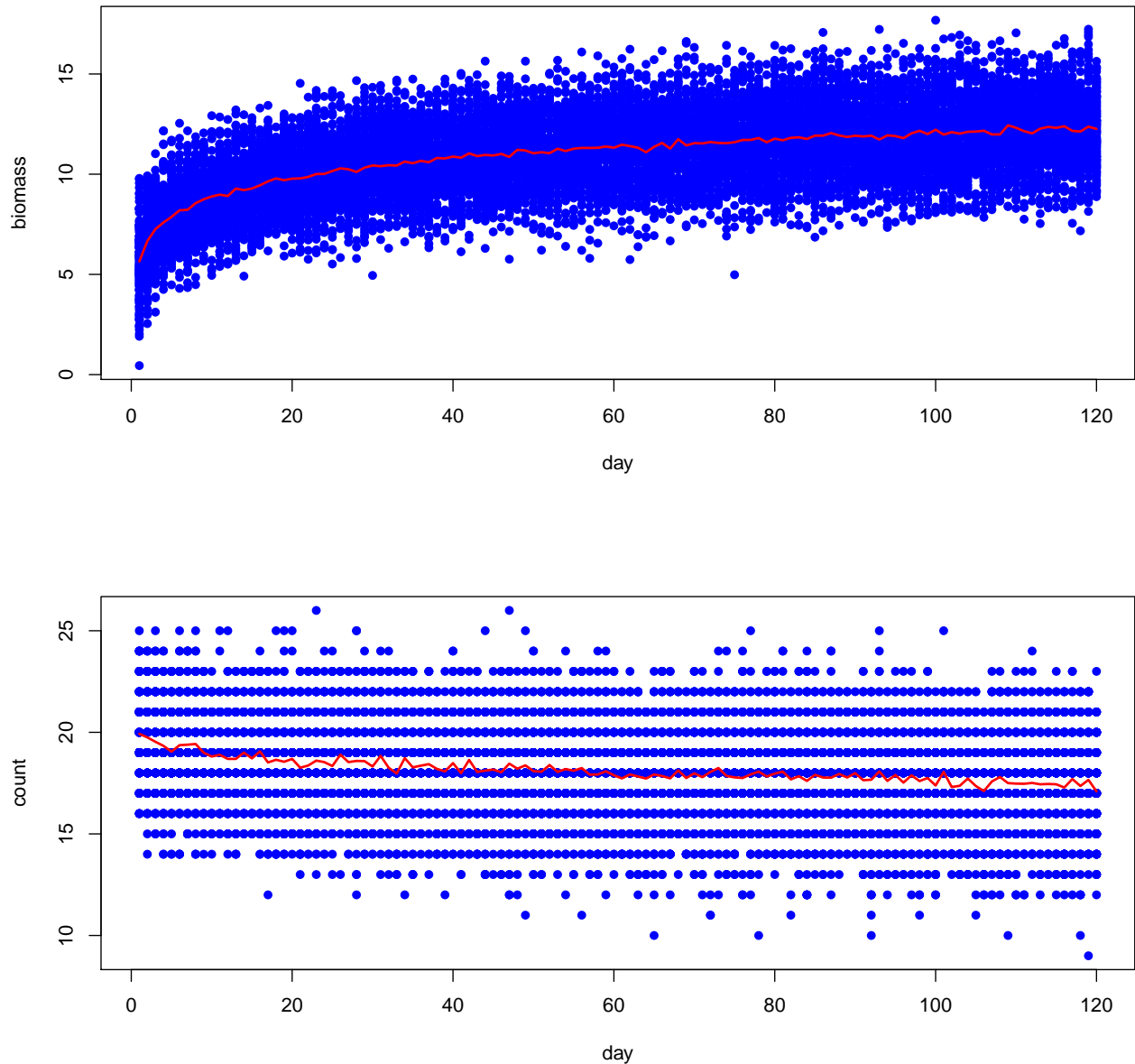


Figure 4: Output of ‘`stackedPlot(x = 'day', y1 = 'biomass', y2 = 'count', data = data, lineCol = 2, lwd = 2, pch = 16, col = 4)`’.

Data analysis

This is the data analysis code which you want to convert into functions:

Question 1. Convert biomass to a z-score

```
setwd(" ")
data = read.csv("SimDataTrend.csv")

biomass.z = (data$biomass - mean(data$biomass))/sd(data$biomass)
```

Question 2. Plot biomass against day with a trendline

```
plot(data[, "biomass"] ~ data[, "day"], pch = 16, xlab = "day", ylab = "biomass")
lines(aggregate(data[, "biomass"], list(data[, "day"]), mean), lwd = 2, col = 2)
```

Question 3. Plot biomass and count against day

```
par(mfrow = c(2,1))

plot(data[, "biomass"] ~ data[, "day"], pch = 16)
lines(aggregate(data[, "biomass"], list(data[, "day"]), mean), lwd = 2, col = 2)

plot(data[, "count"] ~ data[, "day"], pch = 16)
lines(aggregate(data[, "count"], list(data[, "day"]), mean), lwd = 2, col = 2)
```