Imputation of Missing Body Sizes in Parasites

Daniel Smith

17 April 2019

## Intro

## Missing Data Structure

Before we move further with imputation of missing values it’s important to understand the overall structure of data and the missing values present within that.

### Visualising Missing Data for SSA

Load the required packages.

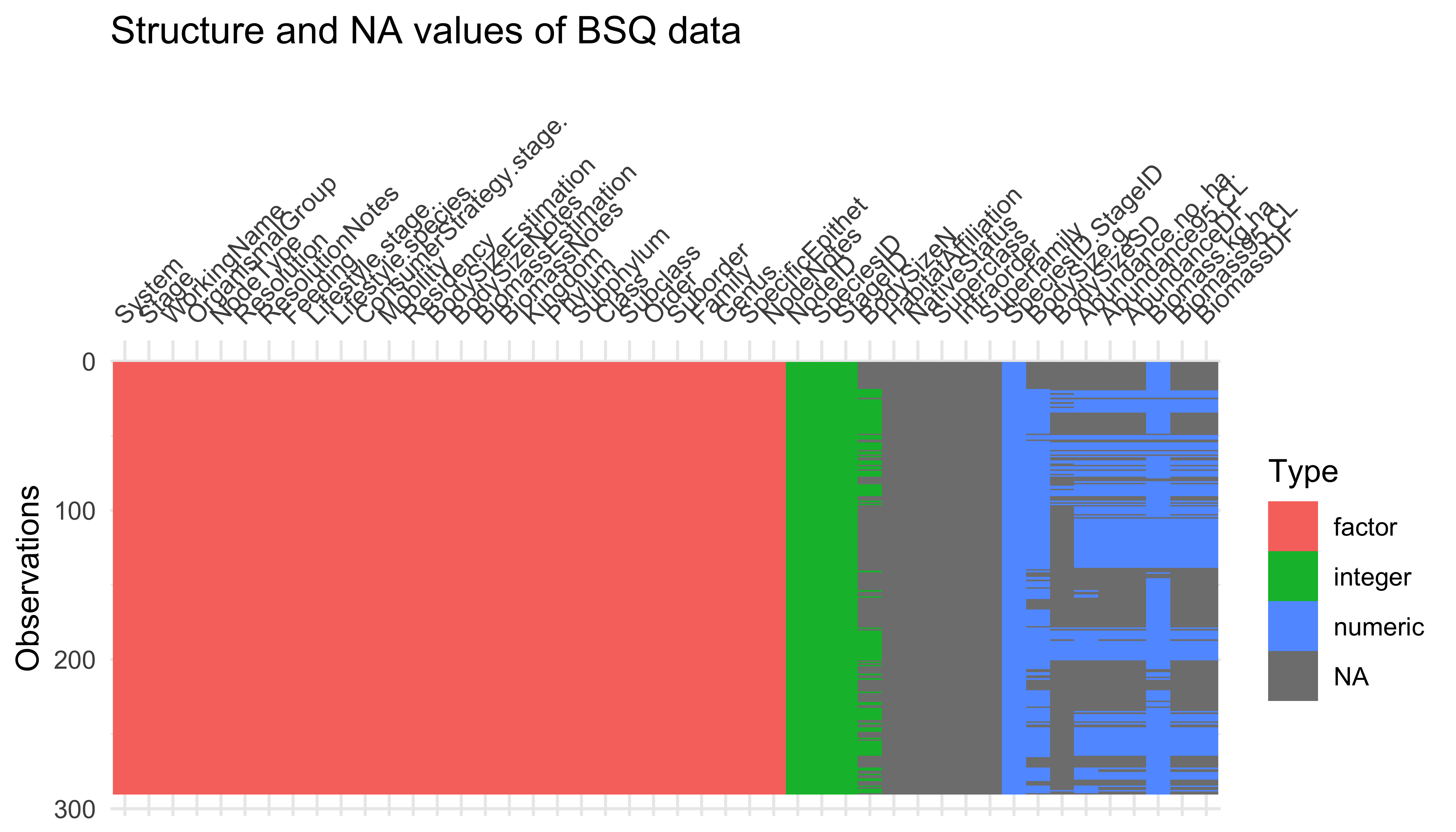
library(tidyverse) # general data wizadry and manipulation  
library(VIM) # visualise missing data patterns  
library(naniar) # data structure visualisation in a 'tidy' manner  
library(visdat) # accompanies the naniar package  
  
# set environment  
rm(list=ls()) # remove all objects  
set.seed(12) # set the random seed to 12 so results are reproducible

First we need to read in the required data from a GitHub repositary. For this methods demonstration we will just use the BSQ subportion of the Carpinteria foodweb as an example. The process is exactly the same for all foodwebs.

bsq <- read.csv("https://raw.githubusercontent.com/SmithD19/FoodWeb/master/data/interactionwebdb/Carpinteria/BSQweb\_Nodes.csv") # load data from GitHub repo - simple

A plot below using the visdat library allows us to visualise this further and gives more information about overall data structure and the type of data.

# Overall data structure and missing values using visdat library  
vis\_dat(bsq) +  
 labs(title = "Structure and NA values of BSQ data")

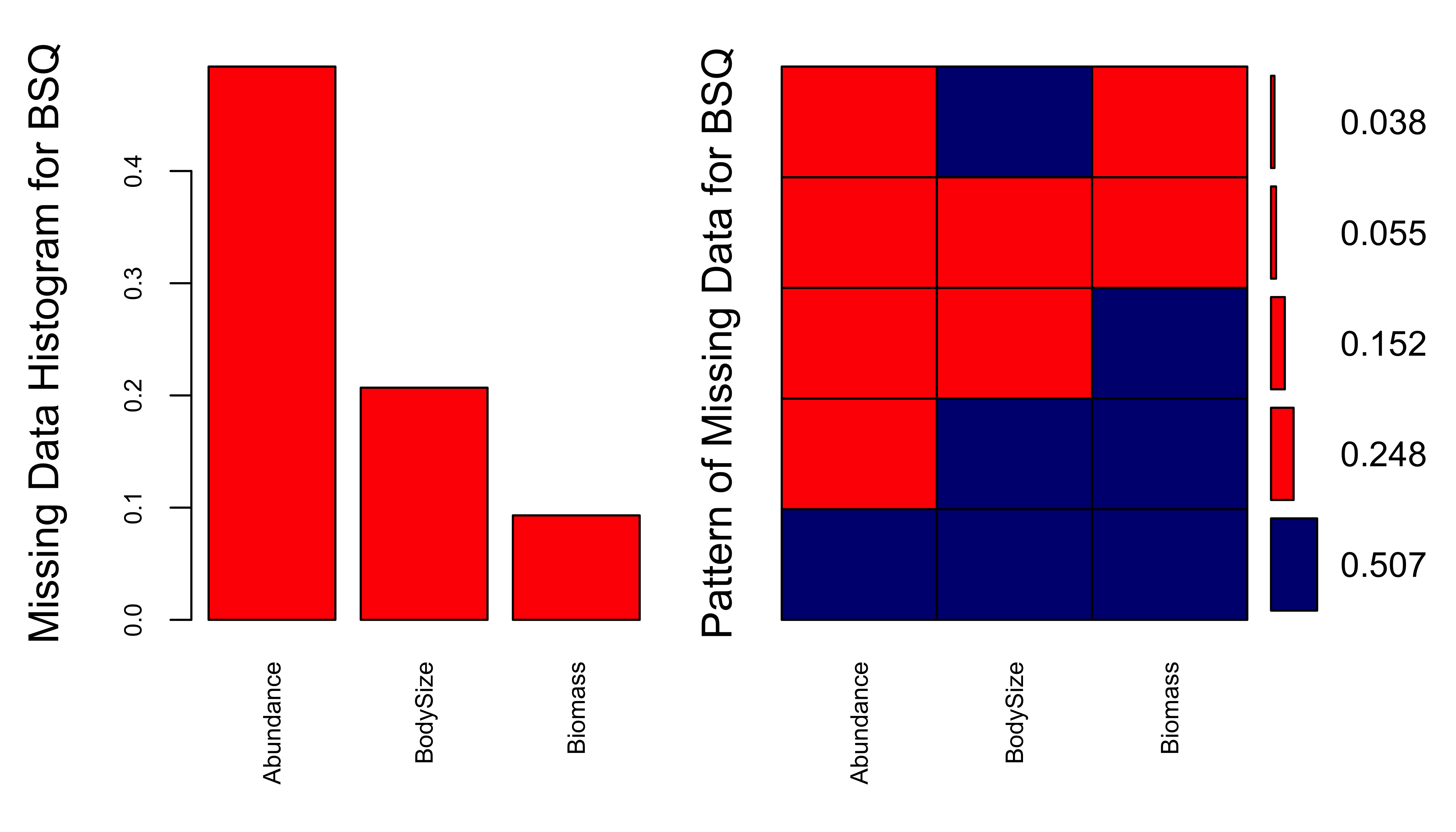


We can clearly see the NA values in the data frame here marked by the grey blocking. This also makes it easy for us to select only the values we care about for imputation, the trait values and give them better, easier to write names in a new and less cluttered data frame.

bsq\_wrk <-   
 bsq %>% # select and rename some variables  
 dplyr::select(FunctionalGroup = ConsumerStrategy.stage., BodySize = BodySize.g.,   
 Biomass = Biomass.kg.ha., Abundance = Abundance.no..ha.) %>%   
 mutate(BodySize = BodySize / 1000) # change BodySize from g to Kg so that units are equal

Using the handy VIM package we can quickly visualise the missing data present in the original data set - in particular the trait values of the nodes are what we’re interested in for now.

aggr(bsq\_wrk[,2:4], col=c('navyblue','red'),   
 numbers=TRUE, sortVars=TRUE, labels=names(bsq\_wrk[, 2:4]),   
 cex.axis=.7, gap=3, ylab=c("Missing Data Histogram for BSQ","Pattern of Missing Data for BSQ"))



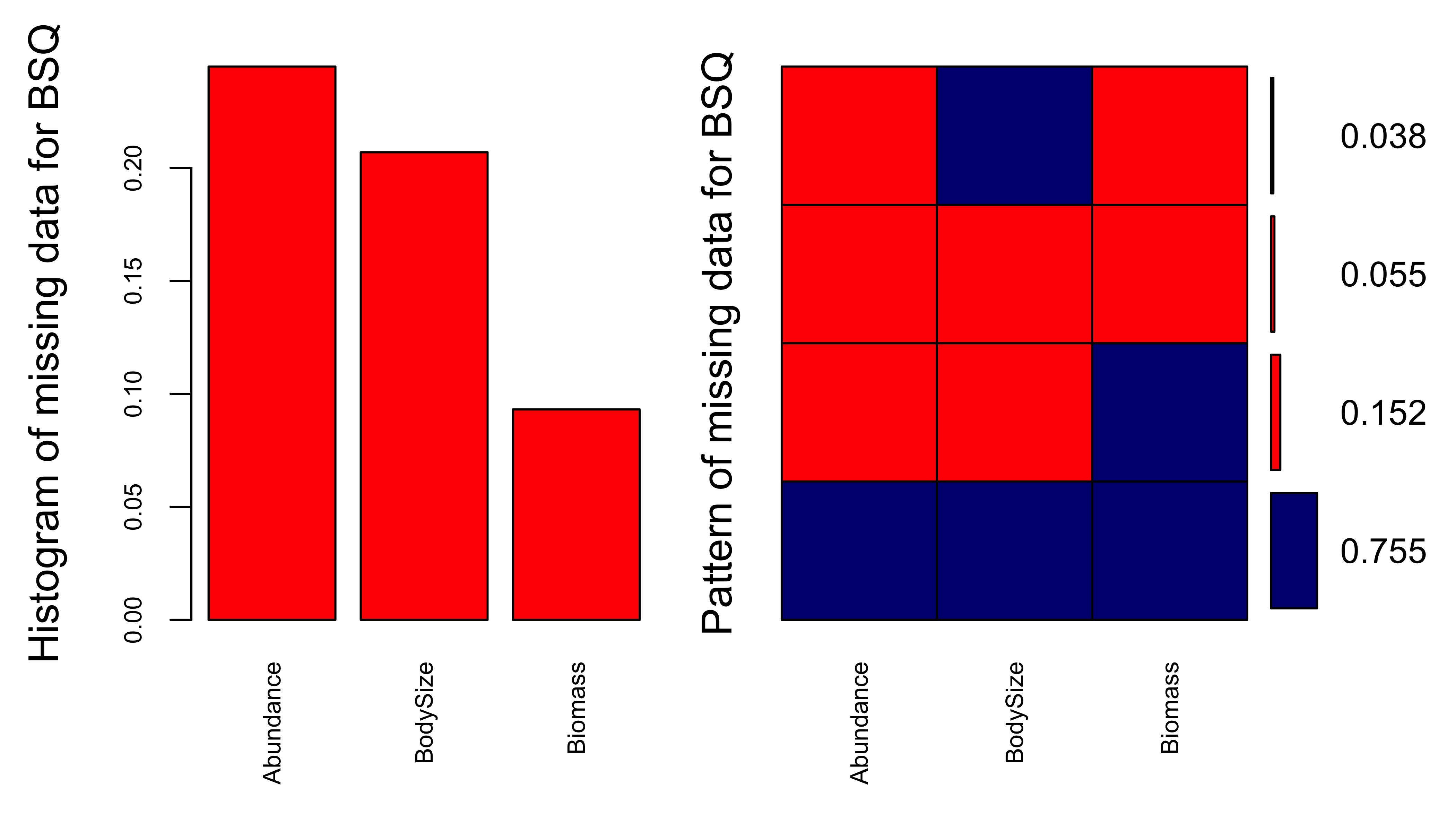
Using the equation we can conditionally fill in variables that are missing from data points using mutate() and case\_when(). Below is the workflow used for conditionally replacing this data based on the equation and reassembling the data back to it’s original structure but containing the new values inferred from the equation above.

bsq\_new <-   
 bsq\_wrk %>%   
 ## if BodySize = NA & if Biomass & Abundance != NA then BodySize = Biomass/Abundance  
 mutate(BodySizeNew = case\_when(!is.na(Abundance & Biomass) ~ Biomass / Abundance),  
 ## if Abundance = NA and BodySize & Biomass != NA then Abundance = Biomass/BodySize   
 AbundanceNew = case\_when(!is.na(BodySize & Biomass) ~ Biomass / BodySize),  
 ## if Biomass = NA and BodySize & Abundance != NA then Biomass = Abundance \* BodySize  
 BiomassNew = case\_when(!is.na(Abundance & BodySize) ~ Abundance \* BodySize)) %>%   
 ## join the columns of new and old together use coalesce -- Biomass + BiomassNew...  
 ## this selects the original value first so only values that are missing from original dataset and then computed are selected   
 mutate(BodySize\_Work = coalesce(BodySize, BodySizeNew),  
 Biomass\_Work = coalesce(Biomass, BiomassNew),  
 Abundance\_Work = coalesce(Abundance, AbundanceNew)) %>%   
 ## now dplyr::select and rename the new working colums to replace the old incomplete data set  
 dplyr::select(FunctionalGroup, BodySize = BodySize\_Work,  
 Biomass = Biomass\_Work, Abundance = Abundance\_Work)

Now with the extra values we added from the inferred relationship between BodySize, Biomass and Abundance we can look at how much data we have to work with when getting ready for the imputation process.

First lets take a look at another plot using the VIM aggr plotting function.

aggr(bsq\_new[,2:4], col=c('navyblue','red'),  
 numbers=TRUE, sortVars=TRUE, labels=names(bsq\_new[, 2:4]),  
 cex.axis=.7, gap=3, ylab=c("Histogram of missing data for BSQ","Pattern of missing data for BSQ"))



Now it looks like we have 75% of the data with no missing values for any of that traits we’re interested in. This is significantly better than the 50% we had earlier and gives us a better starting point for the imputation process using the mice package, which is shorthand for Multivariate Imputation using Chained Equations.

## Imputation

For imputation we have the choice of several packages in the CRAN repository. For now the most appropriate package is that of mice as shown by penone, without phylogenetic data this method provides the best impuation results, particularly for body size data and traits highly correlated to this.

library(mice) # R package for Multivariate Imputation using Chained Equations  
# Be aware that the MASS package attatched to mice masks the select() function by dplyr  
# It is therefore necessry to specifically call dplyr::select() to use this function

Lets create a new data frame for the imputation process. It’s important in the imputation process to log transform values that could be correlated with each otehr, this process reduces the co-linearity of the values and therefore makes imputation more accurate.

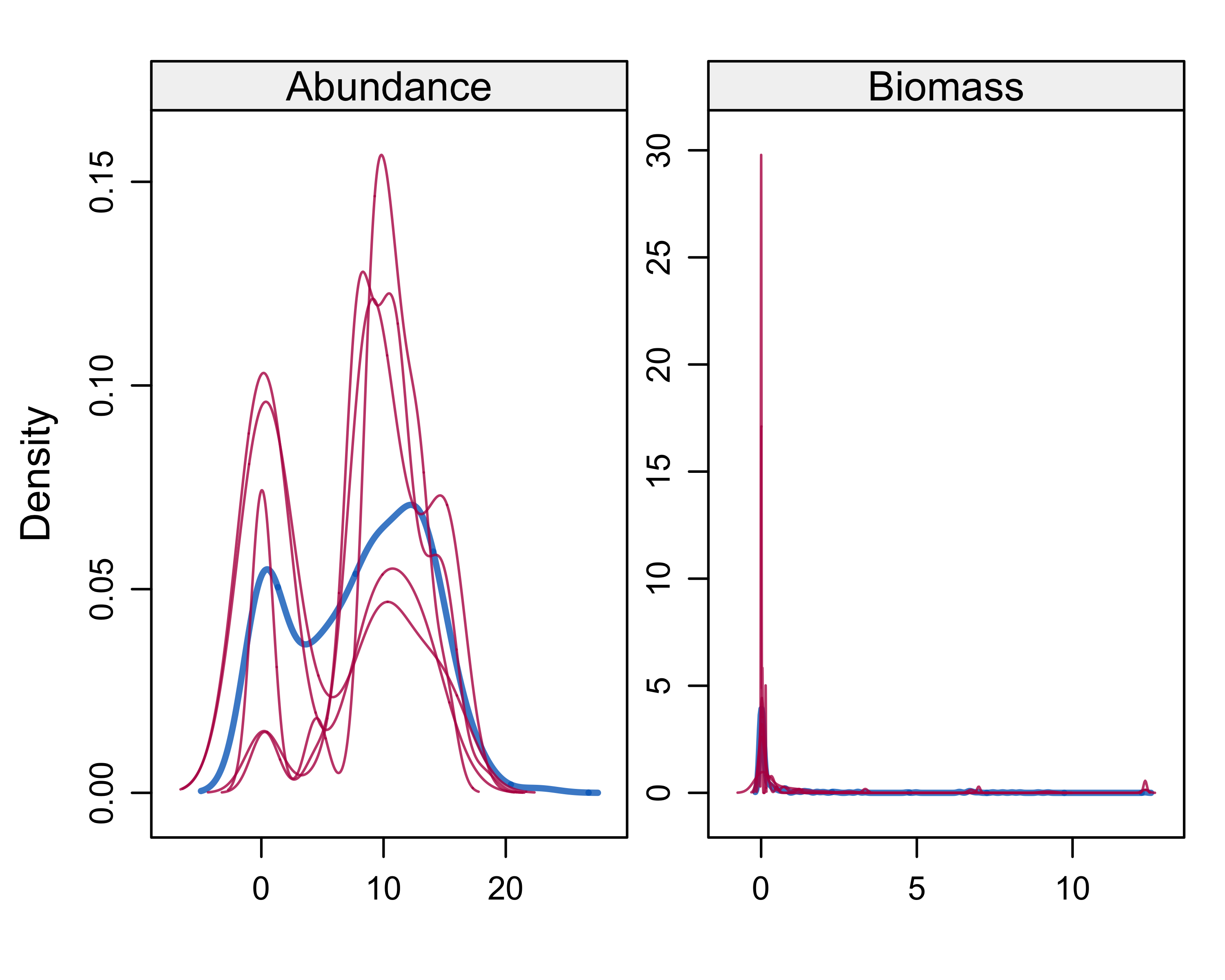
bsq\_imp <- bsq\_new %>% dplyr::select(BodySize, Biomass, Abundance) %>%   
 ## log transform the variables you want to impute, addition of 1 solves problems   
 ## with infinite results due to logging values below 0  
 mutate(BodySize = log(BodySize + 1), Biomass = log(Biomass + 1), Abundance = log(Abundance + 1))

Using the imputation function mice() we can now impute the missing data. the m value corresponds to the number of data sets imputed and for this example we will use 5 as it’s easier to show, but for real analysis more appropriate iterations of imputation are suggested (50+). The data will be imputed using the pmm method whcih is short for predictive mean matching.

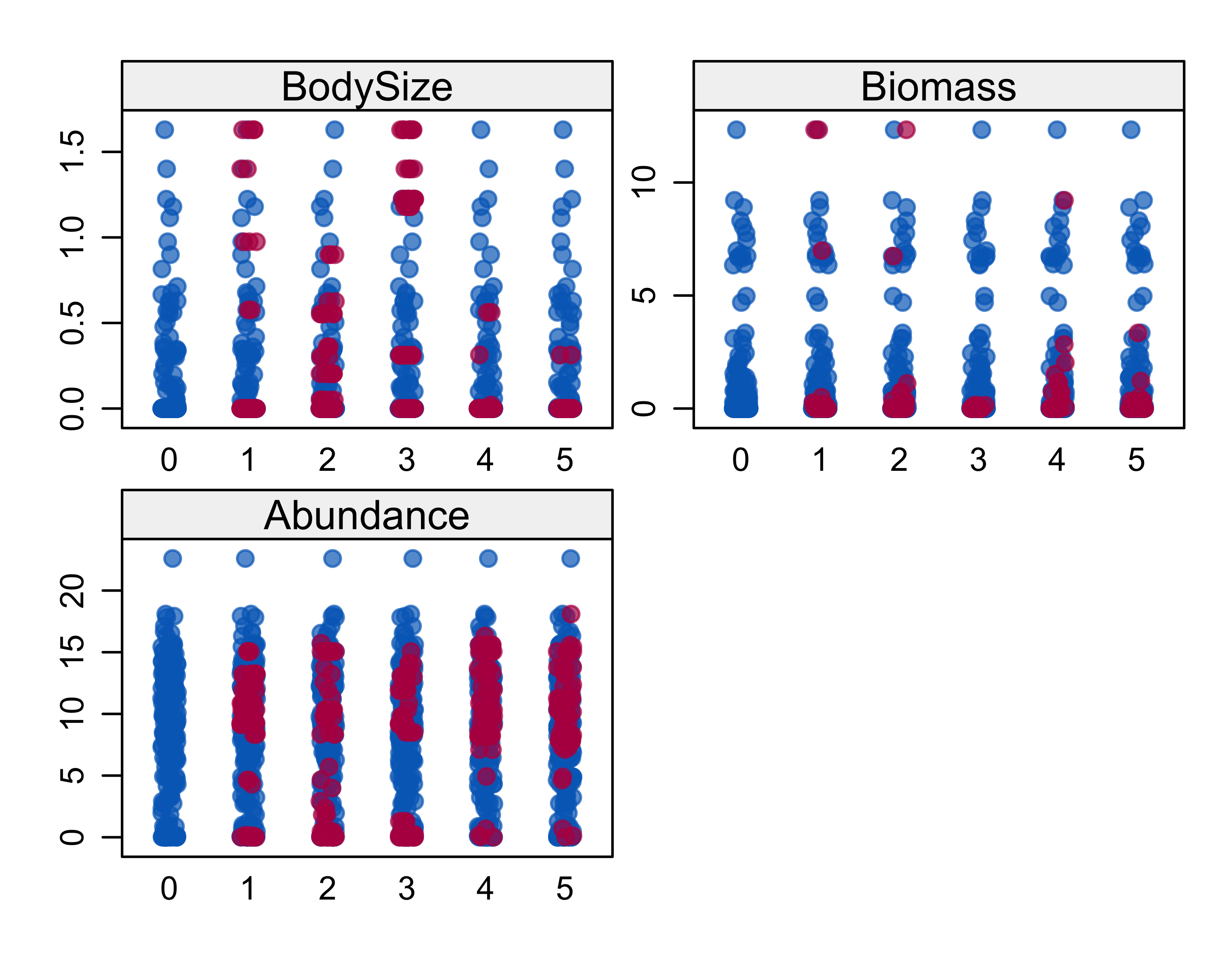
bsq\_imp <- mice(bsq\_imp, m = 5, method = "pmm")

We can look at some of the imputed values and compare them to our original data set using some convenient functions provided in the mice package.

densityplot(bsq\_imp, BodySize ~ Abundance + Biomass)



stripplot(bsq\_imp, pch = 20, cex = 1.2)



bsq\_imp <- complete(bsq\_imp)