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.

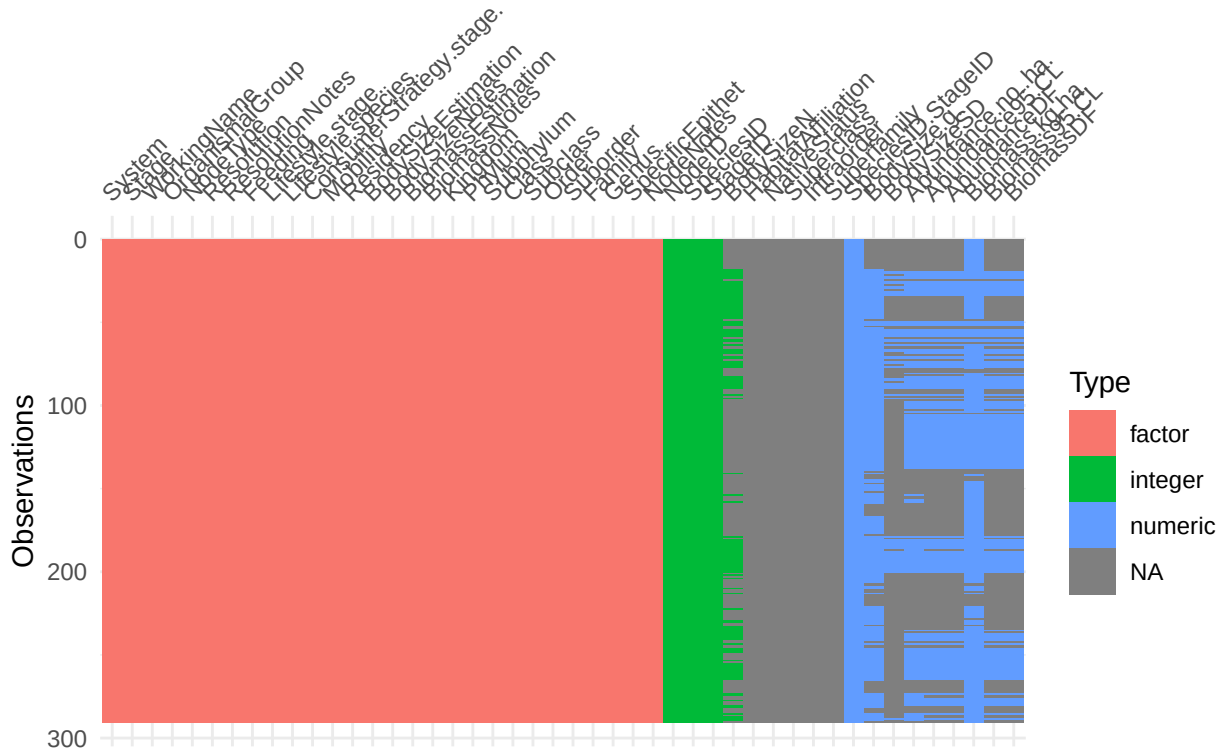
```
library(tidyverse)
library(VIM)
library(naniar)
library(visdat)

# set environment
rm(list=ls())
set.seed(12)
```

Read in the required data. For this we will just use the BSQ subportion as an example. The process is exactly the same for all foodwebs.

```
bsq <- read.csv("https://raw.githubusercontent.com/SmithD19/FoodWeb/master/data/interactionwebdb/Carpin
```

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

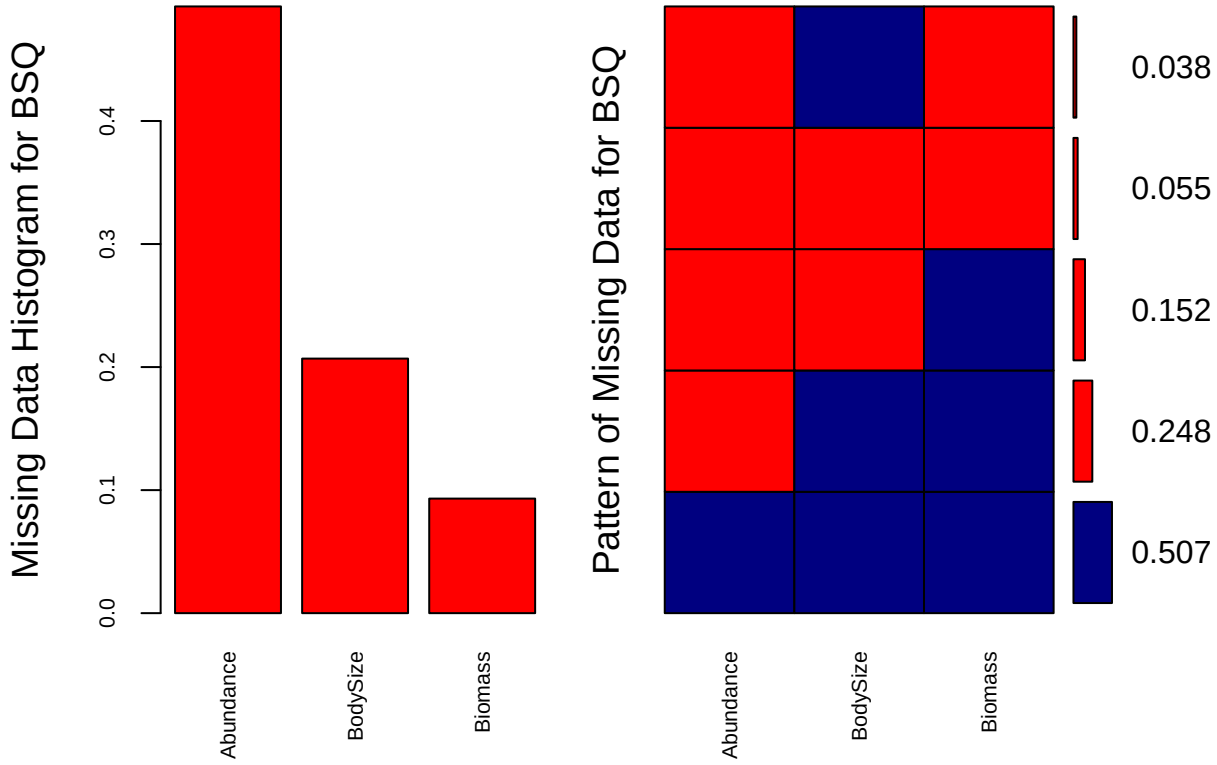
```
# Overall data structure and missing values using visdat library
vis_dat(bsq) +
  labs(title = "Structure and NA values of BSQ data")
```




```
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 f
```

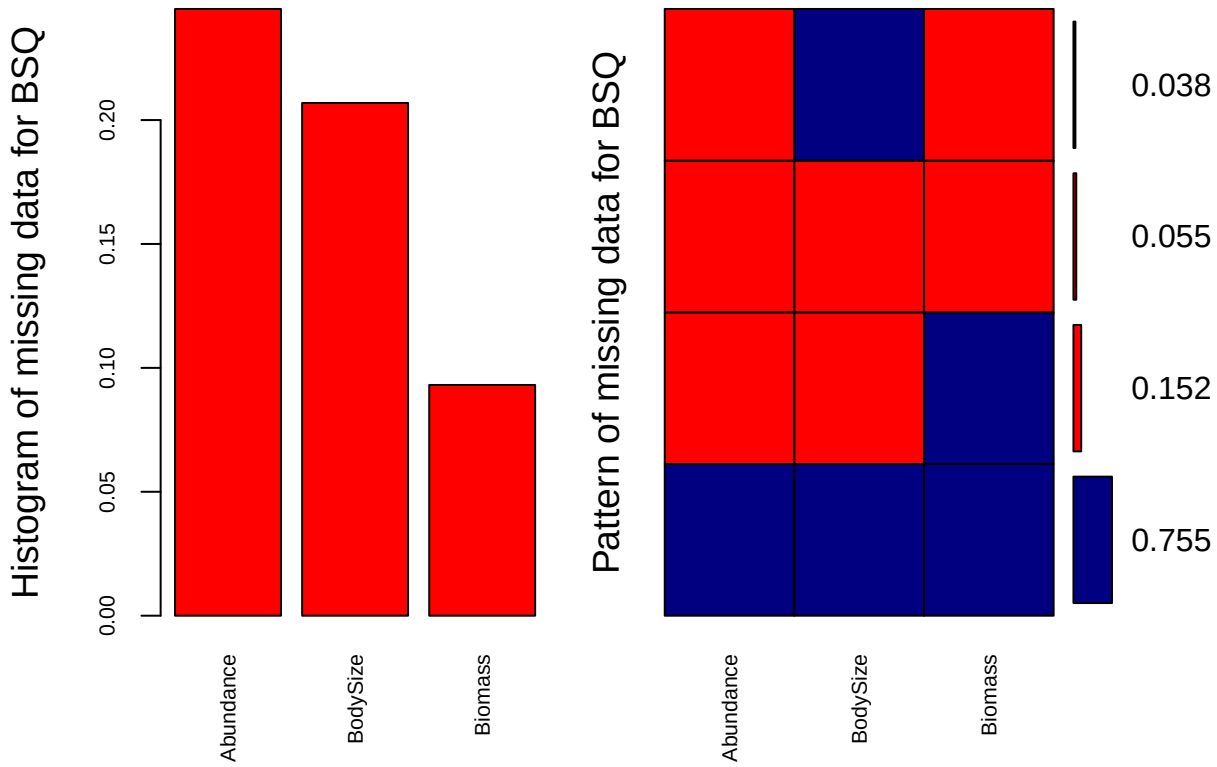


Using the equation $Biomass = BodySize * Abundance$ we can conditionally fill in variables that are missing from data points using `mutate()` and `case_when()`. Below is the workflow used for conditionally replacing this data based on the equation and reassembling the data back to it's original structure 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 th
  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.