Some ideas for exploring missing data

Nicholas Tierney

Infectious Disease Ecology and Modelling group

Telethon Kids Institute, Perth, WA, Australia

library(tidyverse)  
library(visdat)  
library(naniar)

# Introduction

When you do data analysis, you come across missing data. Because I felt so frustrated by how hard it was to handle and wrangle missing data, I wanted to make it easier. In my endeavours I have written two R packages, visdat and naniar, for exploring missing data, and several papers on the topic.

The present goal is to share some ideas on exploring missing data, using naniar, and visdat. To that end, we will focus on four questions.

1. How do we start looking at missing data?
2. How do we explore missingness in variables?
3. How do we explore missingness relationships?
4. How do we explore imputed values?

We will explore questions 1 and 2 in this article. Questions 3 and 4 will be explored in the next issue.

But first, let’s introduce ourselves to the data.

# The data

The data used for this paper is measurements of rodents in Kansas, from Hope (2023). The use of this data is inspired by Allison Horst’s “Exploring missing values in naniar” [shiny application](https://allisonhorst.shinyapps.io/missingexplorer/). In this paper we use a different, larger set of the data. For information on the metadata of the paper see [here](http://lter.konza.ksu.edu/content/csm08-small-mammal-host-parasite-sampling-data-16-linear-trapping-transects-located-8-lter). The data set provides various biometric length and weight measurements, for four species of rodents: the Eastern woodrat, Prairie vole, Western harvest mouse, and Deer mouse. Table [Table 1](#tbl-rodents-slice) shows a snapshot of 6 selected rows of the data, which shows some of the missingness.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1: A subset of the rodents dataset, containing measurements of various rodents, including: total length - from tail to nose, tail length, hind foot length, ear length, and weight. Each row represents a measurement of a given species of roden at a particular date. There are missing values represented by ‘NA’.   | date | species | total\_length | tail\_length | hind\_foot\_length | ear\_length | weight | sex | age | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 2016-07-21 | prairie vole | NA | 32 | 20 | NA | 30.0 | F | A | | 2016-07-21 | deer mouse | NA | 62 | 19 | NA | 17.5 | F | A | | 2016-07-21 | western harvest mouse | NA | NA | NA | NA | NA | N | NA | | 2016-07-21 | western harvest mouse | NA | NA | NA | NA | NA | N | NA | | 2016-07-21 | prairie vole | NA | NA | NA | NA | NA | N | NA | | 2016-07-21 | prairie vole | NA | 26 | 18 | NA | 18.0 | M | Juv | |

# How do we start looking at missing data?

To get an overview of the missing data, we can use the visdat package (Tierney 2017), which was inspired by the work in [csv-fingerprint](https://setosa.io/blog/2014/08/03/csv-fingerprints/), and functions like missmap, from Amelia (Honaker, King, and Blackwell 2011). The key function for exploring missingness is vis\_miss(), which visualises the missingness of a whole dataframe. [Figure 1](#fig-vis-miss) gives an example where it displays the data as missing, or not missing, and provides information on the amount of missings in each column.

vis\_miss(rodents)

|  |
| --- |
| Figure 1: An overview of missing data in the rodents dataset. The x axis shows the variables of the data, along with the amount of missingness in that variable, and the y axis shows the rows. Each cell represents the missingness of a datum. The overall missingness is given in a percentage below in the legend. We learn that there is nearly 29% missing data overall, the missing data occurs in total\_length, tail\_length, hind\_foot\_length, ear\_length, weight, and age. |

We learn there is nearly 29% missing data overall, the missing data occurs in total\_length, tail\_length, hind\_foot\_length, ear\_length, weight, and age, and mostly in total\_length and ear\_length.

## Exploring subgroups: Using facetting in visdat

To see this plot split up by species, we can split up the vis\_miss plots into several facetted plots via the facet argument. For example, in [Figure 2](#fig-vis-miss-facet) we facet by the species variable. Visually, it appears that the missingness occurs in each species at roughly the same rate.

vis\_miss(rodents, facet = species)

|  |
| --- |
| Figure 2: An further overview of missing data in the rodents dataset. Similar to the previous graphic, we now present a facetted series of sub plots, one for each species. We learn from this that the missingness seems to be the same across each species |

There are other functions in the visdat package that focus on other types of data, for example, vis\_value(), vis\_binary(), and vis\_compare(). To read more about the functions available in visdat see the vignette [“Using visdat”](https://CRAN.R-project.org/package=visdat/vignettes/using_visdat.html).

# How do we explore missingness in variables?

The visdat package provides overviews of data, whereas naniar provides a more comprehensive set of tools for missing data.

## Numerical summaries of missing values

Two convenient counters of complete values and missings are n\_miss() and n\_complete(). These work on both data frames and vectors, similar to dplyr::n\_distinct()

dplyr::n\_distinct(rodents)

[1] 617

dplyr::n\_distinct(rodents$tail\_length)

[1] 96

n\_miss(rodents)

[1] 2013

n\_miss(rodents$tail\_length)

[1] 245

n\_complete(rodents)

[1] 4944

n\_complete(rodents$tail\_length)

[1] 528

The functions prop\_miss\_case() and pct\_miss\_case() return numeric value describing the proportion or percent of missing values in the dataframe.

prop\_miss\_case(rodents)

[1] 0.7477361

pct\_miss\_case(rodents)

[1] 74.77361

Similar to pct\_miss\_case(), prop\_miss\_case(), pct\_miss\_var() and prop\_miss\_var() returns the percent and proportion of variables that contain a missing value.

prop\_miss\_var(rodents)

[1] 0.6666667

pct\_miss\_var(rodents)

[1] 66.66667

The syntax for the other numerical summaries in naniar are miss\_, and then case, or var to refer to cases or variables. There are then summary, table suffixes. For example, the miss\_case\_summary() function returns a numeric value that describes the number of missings in a given case (aka row), the percent of missings in that row.

miss\_case\_summary(rodents) |> head()

# A tibble: 6 × 3  
 case n\_miss pct\_miss  
 <int> <int> <dbl>  
1 63 6 66.7  
2 74 6 66.7  
3 77 6 66.7  
4 78 6 66.7  
5 79 6 66.7  
6 91 6 66.7

The miss\_case\_table() function tabulates the number of missing values in a case / row. Below, this shows the number of missings in a case:

miss\_case\_table(rodents)

# A tibble: 7 × 3  
 n\_miss\_in\_case n\_cases pct\_cases  
 <int> <int> <dbl>  
1 0 195 25.2   
2 1 48 6.21  
3 2 277 35.8   
4 3 10 1.29  
5 4 14 1.81  
6 5 49 6.34  
7 6 180 23.3

We can interpret this output as follows:

* 195 cases with 0 missings, which comprises about 25% of the data,
* 48 cases with 1 missing, these make up 6% of the data,
* 277 cases with 2 missing, these make up 35% of the data,
* and so on.

miss\_var\_summary() returns the number of missing values in a variable, and the percent missing in that variable.

miss\_var\_summary(rodents)

# A tibble: 9 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <num>  
1 total\_length 569 73.6  
2 ear\_length 530 68.6  
3 tail\_length 245 31.7  
4 hind\_foot\_length 243 31.4  
5 weight 237 30.7  
6 age 189 24.5  
7 date 0 0   
8 species 0 0   
9 sex 0 0

Finally, the miss\_var\_table() function describes the number of missings in a variable:

miss\_var\_table(rodents)

# A tibble: 7 × 3  
 n\_miss\_in\_var n\_vars pct\_vars  
 <int> <int> <dbl>  
1 0 3 33.3  
2 189 1 11.1  
3 237 1 11.1  
4 243 1 11.1  
5 245 1 11.1  
6 530 1 11.1  
7 569 1 11.1

We can interpret this as there being:

* 3 variables with 0 missings, comprising 33% of variables in the dataset, and
* in the remaining variables similar patterns of missings, but not the exact same number of missing values.

## Visualise missingness in variables

To specifically focus on the number or proportion of missings in each variable, we can use gg\_miss\_var(), as seen in [Figure 3](#fig-gg-miss-var).

gg\_miss\_var(rodents)

|  |
| --- |
| Figure 3: Number of missing values for each variable. The x axis shows the number of missings, and the y axis shows each variable. We learn total length, and ear length have the most missing values, followed by tail length, hind foot length, weight, and age. |

This displays the number of missing values in each variable. We learn similar information to [Figure 1](#fig-vis-miss): total length, and ear length have the most missing values, followed by tail length, hind foot length, weight, and age. Just like with vis\_miss(), we can add in facets in these plots, via the facet argument, see [Figure 4](#fig-gg-miss-var-facet). We learn again, that the species have similar amounts of missing data in their variables. Sometimes it is useful to confirms the same piece of information!

gg\_miss\_var(rodents, facet = species, show\_pct = TRUE)

|  |
| --- |
| Figure 4: Similar to the above plot but one subplot for each species. We learn that the missingness pattern is pretty similar for each species. We normalise using show\_pct = TRUE as there are different numbers of observations in each species. |

It feels like there are several patterns with the missingness - some variables tend to go missing at the same time. To explore these patterns we can use gg\_miss\_upset(), which produces an “upset” plot of the intersecting sets of missingness (Conway, Lex, and Gehlenborg 2017). This can be thought of as a generalised way to visualise intersecting Venn diagrams.

gg\_miss\_upset(rodents)

|  |
| --- |
| Figure 5: An upset plot of 7 sets of missingness in the rodents data displaying a more nuanced depiction of the patterns of missingness in the data. The number of missings in each variable is seen on the bottom left bar plot. The bottom panel shows the different sets of co-occuring missingness. For example, the bottom left two dots show ear length and total length going missing together - corresponding to the bar plot above it, showing 278 missings. We learn that there are two dominant sets of missingness, where ear length and total length go missing, and then weight hind foot length, tail length, ear length and total length being missing. |

There are more visualisations available in naniar (each starting with gg\_miss\_) - you can see these in the [“Gallery of Missing Data Visualisations” vignette](https://cran.r-project.org/package=naniar/vignettes/naniar-visualisation.html). Most plots created with the gg\_miss family all have a basic theme (except for gg\_miss\_upset()), but you can customise them by adding components like a standard ggplot object:

gg\_miss\_var(rodents) +   
 theme\_bw() +   
 labs(y = "Number of missing observations")

It is also worth noting that for every visualisation of missing data in naniar, there is an accompanying function to extract the data used in the plot. This is important as the plot should not return a dataframe - but we want to make the data available for use by the user so it isn’t locked into a plot.

For example, miss\_var\_summary() provides the dataframe that gg\_miss\_var() is based on.

miss\_var\_summary(rodents)

# A tibble: 9 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <num>  
1 total\_length 569 73.6  
2 ear\_length 530 68.6  
3 tail\_length 245 31.7  
4 hind\_foot\_length 243 31.4  
5 weight 237 30.7  
6 age 189 24.5  
7 date 0 0   
8 species 0 0   
9 sex 0 0

Which also works with group\_by():

rodents %>%   
 group\_by(species) %>%   
 miss\_var\_summary()

Similarly, there is a data\_vis\_miss() function in the visdat package, which returns the data in the format that this visualisation requires.

data\_vis\_miss(rodents)

The aim of these is to provide the data required to make these visualisations, so if people want to create their own more customised versions of vis\_miss() or gg\_miss\_var() then they can do that.

## Conslusion

This article has explored how to generate summaries and visualisations to help understand the missingness in your data set with the visdat and naniar packages. In the next issue, we will extend this to exploring and visualising missingness in multivariate settings, including bivariate missingness plots, identifying clusters of missingness and visualising imputed data.

# References

Conway, Jake R., Alexander Lex, and Nils Gehlenborg. 2017. “UpSetR: An R Package for the Visualization of Intersecting Sets and Their Properties.” *Bioinformatics (Oxford, England)* 33 (18): 2938–40. <https://doi.org/10.1093/bioinformatics/btx364>.

Honaker, James, Gary King, and Matthew Blackwell. 2011. “Amelia II: A Program for Missing Data.” *Journal of Statistical Software* 45 (1): 1–47. <https://doi.org/10.18637/jss.v045.i07>.

Hope, Andrew. 2023. “CSM08 Small Mammal Host-Parasite Sampling Data for 16 Linear Trapping Transects Located in 8 LTER Burn Treatment Watersheds at Konza Prairie.” Environmental Data Initiative. <https://doi.org/10.6073/PASTA/F7BFB7226B093C817DC391C34A88B514>.

Tierney, Nicholas. 2017. “visdat: Visualising Whole Data Frames.” *The Journal of Open Source Software* 2 (16): 355. <https://doi.org/10.21105/joss.00355>.