Principles for missing data exploration

Nicholas Tierney

14/12/2016

Introduction

Missing data is ubiquitous in data analysis, and are often the source of much energy, frustration, and confusion. Since 2014 there has been substantial development in the area of "tidy data" (@wickham), which states the (surprisingly simple!) rule that each row is an observation and each column is a variable, which makes it easy to reason with data. This paper describes approaches for summarising missing data in numerical and graphical forms whilst maintaining a tidy format.

Types of missing data

Three categories of missing data are usually identified: Missing Completeley at Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). MCAR describes where missingness has no association with the observed or unobserved data. For example, assessments of lung function taken at a workplace may be missing for workers who are on vacation. If there is no known or measurable relationship between the timing of the tests and the timing of vacations, and if the other relevant features of the workers who are on vacation at the time of the tests are similar to that of other workers, then these missing data can be considered MCAR. MAR describes cases where missingness depends on data observed, but not data unobserved. For example if the missing lung function data occurs in workers who are being assessed for depression, and if there is no relationship between lung function and depression, then it can be considered as MAR. MNAR is where the missingness of the response is related to an unobserved value relevant to the assessment of interest. For example, if BMI is of interest, but those with especially large BMIs are more likely to have missing BMI data, these data can be considered as MNAR. It is important for researchers to recognise MNAR as it introduces bias into the estimation of associations and parameters of interest. For example, if lung function and BMI are negatively correlated, an estimate of BMI based on the MNAR may be too low.

These three varieties of missing data could be further divided into a knowable structure (MAR) or an unknown structure (MAR or MNAR), where the process driving data becoming missing are either known or unknown, and structure refers to variables and interactions that may influence missingness. Data MCAR are without a missingness structure, as missingness does not have any dependence on other variables. Determining whether this is known or unknown is important for determining whether bias may be introduced into the analysis.

Existing approaches for handling missing data

Tests confirming whether data is MCAR or not are very useful as they open up the doors for the use of standard multiple imputation techniques. As described by Little,who proposed a single test statistic for testing MCAR, involving an evaluation of equality of means between identified missing data groups. Rejection of this test result gives strong evidence that the data are not MCAR. Little's test of MCAR is widely used today, especially in social science8 and medical research.9 Recent research has also provided statistical tests and software that evaluate missing data via patterns, equality of means, and homogeneity of variance, and allow for non-normal data.

This is achieved, for example, in the MissMech package for the R statistical software,10 which uses imputation (from either normal or non-normal distributions) to compare means and covariances. These tests enable the researcher to determine whether or not there is sufficient evidence for data to be declared as MCAR.

However, understanding how and why missingness is being generated can become arduous when handling larger data sets, as they can have many missingness patterns, making inference difficult, as there are many combinations of missingness to explore. Reliance on statistical significance testing to assess whether data are missing may fail to address settings where there may not be significant missingness, but a complete case analysis may still result in bias (11). Approaches that elucidate missingness structure that are simple to understand and implement, are therefore still in demand.

Common methods of handling missing data, such as complete case analysis, missing indicator method, and last case carried forward have been shown to be acceptable when data is MCAR.12 ,13 That being said, most recommendations now are to use multiple imputation, but subject to some care as it only reduces bias from analysis when data are MAR or MCAR; multiple imputation also requires variables that influence missingness to be included in the imputation model.1–4 ,14 When data are MNAR, multiple imputation can be used but requires the MNAR mechanism to be known, which is not often undertaken in practice.3 Improving the understanding of missingness structure in a data set allows for consideration of other appropriate multiple imputation methods, or other methods to incorporate partially observed variables, such as random effect models, Bayesian methods, down-weighting analyses, or pattern mixture models.2 ,15 ,16

Current approaches to evaluating missingness

There are various approaches and packages specifically developed to explore missing data, and resultant imputation methods. These include R packages VIM, Amelia, mi, the MANET program as well as the standalone software, MissingDataGUI (17–21). These packages facilitate the graphical exploration of data prior to and after imputation to evaluate missingness trends and causations, and imputation accuracy, respectively.

Data structures for missing data

There are two main forms for representing missing data:

1. Shadow Matrix
2. Long Form Shadow

The Shadow Matrix is simply the representation of the data where the missing values are regarded as TRUE, and the present values are regarded as FALSE. To visualise this matrix it is useful to gather the data into long form, going from each row being an observation and each column being a variable, to columns.

It creates a tension as there is this extra dimension to the data. Visualising missing data can be very challenging, and on the surface is somewhat paradoxical: how do you visualise things that are missing? One way to keep track of missing data is to consider a "shadow matrix", of the data, where it is 1 if missing, and 0 is present. One can think of the "shadow matrix" sitting under the actual data frame, like an additional row sitting on the top.

One common method for visualising missing data is to display the data in binary form of missing or not-missing. This approach can be very helpful for giving an overview of which variables contain the most missingness, and methods for rearranging the rows and columns (such as in the seriation package), can be applied to find clusters, identifying other interesting features of the data that may have previously been hidden or unclear. This method has been used in other missing data packages such as VIM, mi, Amelia, and MissingDataGUI.

These functions provide information on ... ...

Data Structures

How the data structures facilitates the visualisation and the summaries

There are many methods for imputation such as ...

Mice provides imputation methods, diagnostics, and for user-written imputation functions, among many other features.

We consider how numerical summaries and visualisation can be used to explore missing data structure. In the future we would also like to consider model based missing data structures.

Numerical summaries:

A good starting place for exploring missingness structure it to look at numerical summaries. The `ggmissing` package provides functions for summarising missing data, for example finding the overall proportion of missing values in a dataset is obtained with `percent_missing_df(data)`, in our case of using our dataset, we find that 3.0061141% of the data has missing values. similarly, we can find the proportion of cases that contain a missing value with `percent_missing_case`, giving 23.2336957%, and the proportion of variables that contain a missing value with `percent_missing_var(tao)` = 37.5.

```
# percent_missing_df()
# miss_prop_df(tao)
# prop_na_df(tao)
# mean(is.na(tao))

prop_na <- function(x){

  prop_na_df <- data_frame(
    df = percent_missing_df(x),
    var = percent_missing_var(x),
    case = percent_missing_case(x)
  )

  prop_na_df

}

prop_na(airquality)
```

```
## # A tibble: 1 × 3
##         df      var     case
##      <dbl>    <dbl>    <dbl>
## 1 4.793028 33.33333 27.45098
```

```
tao1 <- tao %>% filter(year == 1997)
tao2 <- tao %>% filter(year == 1993)
#
ggmissing::summary_missing_var(tao)
```

```
## # A tibble: 8 × 3
##           variable n_missing    percent
##              <chr>     <int>      <dbl>
## 1         humidity        93 12.6358696
## 2         air.temp        81 11.0054348
```

```
## 3 sea.surface.temp         3  0.4076087
## 4             year         0  0.0000000
## 5         latitude         0  0.0000000
## 6        longitude         0  0.0000000
## 7            uwind         0  0.0000000
## 8            vwind         0  0.0000000
```

```
ggmissing::summary_missing_var(tao1)
```

```
## # A tibble: 8 × 3
##           variable n_missing  percent
##              <chr>     <int>    <dbl>
## 1         air.temp        77 20.92391
## 2             year         0  0.00000
## 3         latitude         0  0.00000
## 4        longitude         0  0.00000
## 5 sea.surface.temp         0  0.00000
## 6         humidity         0  0.00000
## 7            uwind         0  0.00000
## 8            vwind         0  0.00000
```

```
ggmissing::summary_missing_var(tao2)
```

```
## # A tibble: 8 × 3
##           variable n_missing    percent
##              <chr>     <int>      <dbl>
## 1         humidity        93 25.2717391
## 2         air.temp         4  1.0869565
## 3 sea.surface.temp         3  0.8152174
## 4             year         0  0.0000000
## 5         latitude         0  0.0000000
## 6        longitude         0  0.0000000
## 7            uwind         0  0.0000000
## 8            vwind         0  0.0000000
```

```
#
```

from here

- finish the summary methods
- finish the visualisation methods possible
- read and write again
- write down a list of things that I need to fix with ggmissing etc.

Conditional summaries, or summaries grouped by another variable.

summary windows present - % of values that are missing - % of varaibles containing missings - the percent of cases that have at least one missing value - tabulation of the number of values missing per case.

This study is taken from the R package norm, and MissingDataGUI

it would be really cool if we could implement dplyr `group_by` syntax for the data, to produce summaries of missingness for 1993 and 1997 respectively.

Numerical summaries can occur at a few different levels

Single number summaries:

- The proportion elements in dataset that contains missing values
- The proportion of variables that contain any missing values

- the proportion of cases that contain any missing values

tabular summaries: - The proportion of missings in every column (variable) - the proportion of missings in every row (case)

further summaries that use more steps: table_missing_case(airquality) table_missing_var(airquality)

sumamries that return dataframes:

```
summary_missing_var(airquality)
```

```
## # A tibble: 6 × 3
##   variable n_missing   percent
##      <chr>     <int>     <dbl>
## 1    Ozone        37 24.183007
## 2  Solar.R         7  4.575163
## 3     Wind         0  0.000000
## 4     Temp         0  0.000000
## 5    Month         0  0.000000
## 6      Day         0  0.000000
```

```
summary_missing_case(airquality)
```

```
## # A tibble: 153 × 3
##     case n_missing  percent
##    <int>     <int>    <dbl>
## 1      1         0  0.00000
## 2      2         0  0.00000
## 3      3         0  0.00000
## 4      4         0  0.00000
## 5      5         2 33.33333
## 6      6         1 16.66667
## 7      7         0  0.00000
## 8      8         0  0.00000
## 9      9         0  0.00000
## 10    10         1 16.66667
## # ... with 143 more rows
```

Visual summaries:

Exploring missingness

exploring missinginess in multivariate - geom_missing_point()

naniar helps open the door to a crazy world where there are tools to handle missings (NAs).

The aim of naniar is to:

- facilitate the visualisation of missing data
- exploration of possible causes for missing data
- modelling possible mechanisms for missing data
- exploring missing data imputation methods.

1. Exploring missing data
2. Visualising missing data
3. Modelling mechanisms for missing data
4. Testing mechanisms for missing data
5. Confirming mechanisms for missing data

- Helpers for working with missing data
- Systematic principles for exploring missing data

- Guidance on how to sensibly impute missing data
- Visualisation of missing data imputation

Exploration - vis_miss - sort_cols - cluster Visualisations - Model missing cluster - relate clusters to missing data mechanism - testing/modelling - somehow relate this cluster to missing data mechanism - m-fold, or mcmc)

It is important to remember that there isn't a "solution" to the missing data problem, but that instead there are much better ways of doing missing data analysis.

smaller tasks: - Visualise whole data frames + missingness: visdat - Exploratory graphics for missing data: ggmissing, Geoms, - Model missingness structure: - Decision tree method + hierarchical clustering - Use missingness structure to infer patterns / impute:

`nainr` `narniar` `narnia`: A package to a world where missing data makes more sense.

Other potential names, for historical reasons: `banana`: `narwhal` `nagpie` `nagpipes` `naan`
`nacho`
`nada` `nana` `misstletoe` `natrix`

Other commands that might be useful? `is_na` `fill_na` `drop_na` `is_null`

note on the use of the "upsidedown"/shadowmatrix having its own missing values, where we go from being MISSING / !MISSING to MISSING / MISSING_REASON_#1 / MISSING_REASON_#2 / etc.. - this idea of a "senteinel value", which I read about in this http://www.residentmar.io/2016/06/12/null-and-missing-data-python.html > . . . a sentinel value, a special bit pattern in the data column's native type flagging a missing value. This is a traditional way of indicating missing data—recording unknown income as -99 or an unknown year as 0, for example—which is used in-place (hopefully with documentation!) by many datasets.The trouble is that sentinal values are in no way durable. -99 could be a valid year and 0 could be a valid income! Without a special reserved keyword, a sentinel robs you of a value that you might otherwise want to use. They have to be considered on a case-by-case, column-by-column basis, a burdensome thing.

Sometimes just working out where to start can be difficult and even paralysing.

Defining missing data and its various forms.

It is helpful to first define what missing data is, and what it is not. Missing data is data that we know should exist, but for some particular reason is not recorded. For example, if there is temperature data recorded every hour of every day, and one particular hour is missing on a particular day, this is missing data. This is contrasted to data that does not exist at all, for example, combining person-level data with environment level data - A person would not have an ambient temperature, and an environment does not have a pulse. These data are sometimes referred to as NULL data, or non-data.

One of the motivations for understanding structure of missing data is to understand *why* it is missing in the first place. In the example above for the temperature data, the temperature might not have been recorded due to instrument failure, or system-wide shut down, perhaps it was scheduled for maintenance.

While humans are very good at finding patterns, a model-driven approach provides a more precise and potentially more automatic framework for exploring missing data. We propose the use of decision trees as a complementary tool for doing this.

Interesting, potentially incorrect blog post about this:

This means that when answering the question "is this data entry filled?" one must actually consider three possible answers: "Yes", "No, but it can be", and "No, and it cannot be".

mice features:

- Columnwise specification of the imputation model

- Support for arbitrary patterns of missing data
- Passive imputation techniques that maintain consistency among data transformations
- Subset selection of predictors
- Support of arbitrary complete-data methods
- Support pooling various types of statistics
- Diagnostics for imputations
- Callable user-written imputation functions

Extras

Examples of missingness

Canonical sources of missing data are questionnaires. Data obtained from questionnaires are often subject to both unknown and known missingness structure. For example, MCAR data can arise from respondents accidentally failing to answer questions or inadvertently providing inappropriate answers. On the other hand, MAR data may arise due to the structure of the questionnaire. For example, the first question on a survey might be: 'If YES, skip to question 4', resulting in questions 2 and 3 missing. If the structure of the questionnaire is known, this type of missingness can be evaluated easily. However, if this information is not available, the mechanism responsible for producing missing data must be inferred from the data.

Another common source of known and unknown structured missingness is medical examination data. The results of particular medical tests may be: absent for purely random reasons (MCAR), due to the procedure (MAR), or based on decisions arising from the observed data (MNAR). For example, if a worker is young, they may not be subjected to neurodegenerative tests reserved for older workers, leading to MAR or MNAR data, depending on the aim of the analysis. A final example is dropouts in a longitudinal study, where participants do not return for future testing sessions. In this case, it is difficult, sometimes impossible, to ascertain the reason for the dropouts, and hence, whether the missingness is known or unknown, or MCAR, MAR or MNAR. However, this ascertainment is essential if the estimates based on these data are to be believed as unbiased.5–7