Introduction to Missing data

Table of contents

Introduction

Steps for dealing with missing data

Figure 1: reference: https://fastercapital.com/topics/dealing-with-outliers-and-missing-data.html

The five steps to ensuring missing data are correctly identified and appropriately dealt with:

1. Ensure your data are coded correctly.
2. Identify missing values within each variable.
3. Look for patterns of missingness.
4. Check for associations between missing and observed data.
5. Decide how to handle missing data.

Reference: https://argoshare.is.ed.ac.uk/healthyr_book/identification-of-missing-data.html

What is missing data ?

> **Note**
>
> In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a effect on the conclusions that can be drawn from the data

Why are missing data a problem ?

- Most data analysis procedures and statistical software were not designed to handle missing data.
- Ignoring missing data or editing lend an appearance of completeness, but may lead to serious problems:
  - loss of information/power
  - biased results
  - unreliable results

Example: We want to investigate the relationship between the time of dreaming and the lifespan as well as the gestation period of mammals based on `sleep` data.

```
data(sleep, package="VIM")
#help(sleep, package="VIM")
str(sleep)
```

```
'data.frame':   62 obs. of  10 variables:
 $ BodyWgt : num  6654 1 3.38 0.92 2547 ...
```

```
 $ BrainWgt: num  5712 6.6 44.5 5.7 4603 ...
 $ NonD    : num  NA 6.3 NA NA 2.1 9.1 15.8 5.2 10.9 8.3 ...
 $ Dream   : num  NA 2 NA NA 1.8 0.7 3.9 1 3.6 1.4 ...
 $ Sleep   : num  3.3 8.3 12.5 16.5 3.9 9.8 19.7 6.2 14.5 9.7 ...
 $ Span    : num  38.6 4.5 14 NA 69 27 19 30.4 28 50 ...
 $ Gest    : num  645 42 60 25 624 180 35 392 63 230 ...
 $ Pred    : int  3 3 1 5 3 4 1 4 1 1 ...
 $ Exp     : int  5 1 1 2 5 4 1 5 2 1 ...
 $ Danger  : int  3 3 1 3 4 4 1 4 1 1 ...
```

The first six observations in the dataset.

```
head(sleep)
```

```
   BodyWgt BrainWgt NonD Dream Sleep Span Gest Pred Exp Danger
1 6654.000   5712.0   NA    NA   3.3 38.6  645    3   5      3
2    1.000      6.6  6.3   2.0   8.3  4.5   42    3   1      3
3    3.385     44.5   NA    NA  12.5 14.0   60    1   1      1
4    0.920      5.7   NA    NA  16.5   NA   25    5   2      3
5 2547.000   4603.0  2.1   1.8   3.9 69.0  624    3   5      4
6   10.550    179.5  9.1   0.7   9.8 27.0  180    4   4      4
```

Goal

Physiological variables and ecological variables are predictor variables.

- physiological variables include body weight (BodyWgt, in kg), brain weight (BrainWgt, in g), lifespan (Span, in years), gestation period (Gest, in days),
- ecological variables include the degree of predation suffered by the species (Pred), the extent of exposure during sleep (Exp), and the overall danger score faced (Danger).

Missing data patterns

- Univariate pattern: missing values occur on one item that are either entirely observed or missing
- Multivariate pattern: missing values occur on group of items that are either entirely observed or missing

- Monotone pattern: items are ordered such that if item p is missing, then items p+1; ... ; k are also missing
- Arbitrary pattern: random scatter of missing data

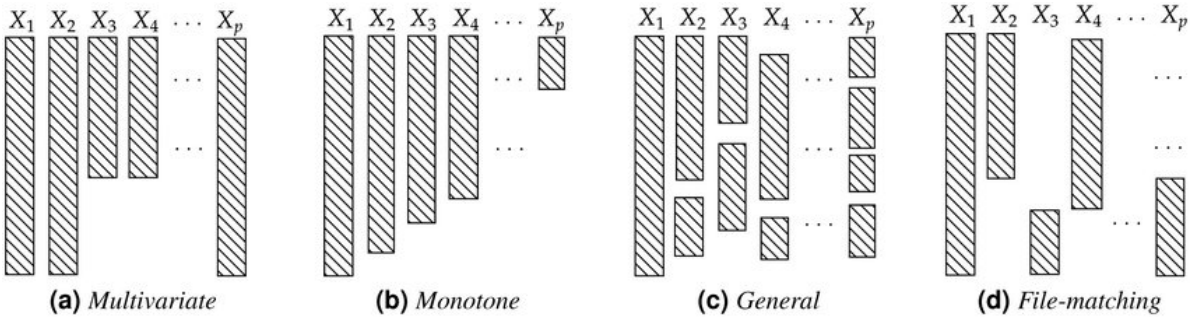


Figure 2: reference: https://www.researchgate.net/publication/356181953/figure/fig1/AS:108951481461 of-missing-data-patterns-a-Multivariate-b-Monotone-C-General-d.jpg

missing data Mechanism

Missing Completely at Random (MCAR):

- A variable is missing completely at random
- Missing data values do not relate to any other data in the dataset and there is no pattern to the actual values of the missing data themselves.
- For example, the smoking status is missing from a random subset of male and female patients
  - This may have the effect of making our population smaller, but the complete case population has the same characteristics as the missing data population.
  - This is easy to handle, but unfortunately, data are almost never missing completely at random.

Missing at Random (MAR):

- A variable is said to be missing at random if other variables (but not the variable itself) in the dataset can be used to predict missingness on a given variable.
- Missingness in a particular variable has an association with one or more other variables in the dataset.
- It would be better to name it **missing conditionally at random**.
- For example, the smoking status is missing for some female patients but not for male patients.

  - data is missing from the same number of female smokers as female non-smokers.
  - the complete case female patients has the same characteristics as the missing data female patients.

Missing Not at Random (MNAR):

- Data are said to be missing not at random if the value of the unobserved variable itself predicts missingness, also called non-ignorable (informative, systematic bias).
- The pattern of missingness is associated with other variables in the dataset, and the actual values of the missing data are not random.
- Data missing not at random are important, as they can alter your conclusions and are the most difficult to diagnose and handle.
- For example, the smoking status is missing in female patients who are more likely to smoke, but not for male patients.
  - the complete case female patients have different characteristics to the missing data female patients.

> **!** Important
>
> How you deal with missing data depends on the type of missingness; once identified, you can start addressing it.

Survivorship bias ( )

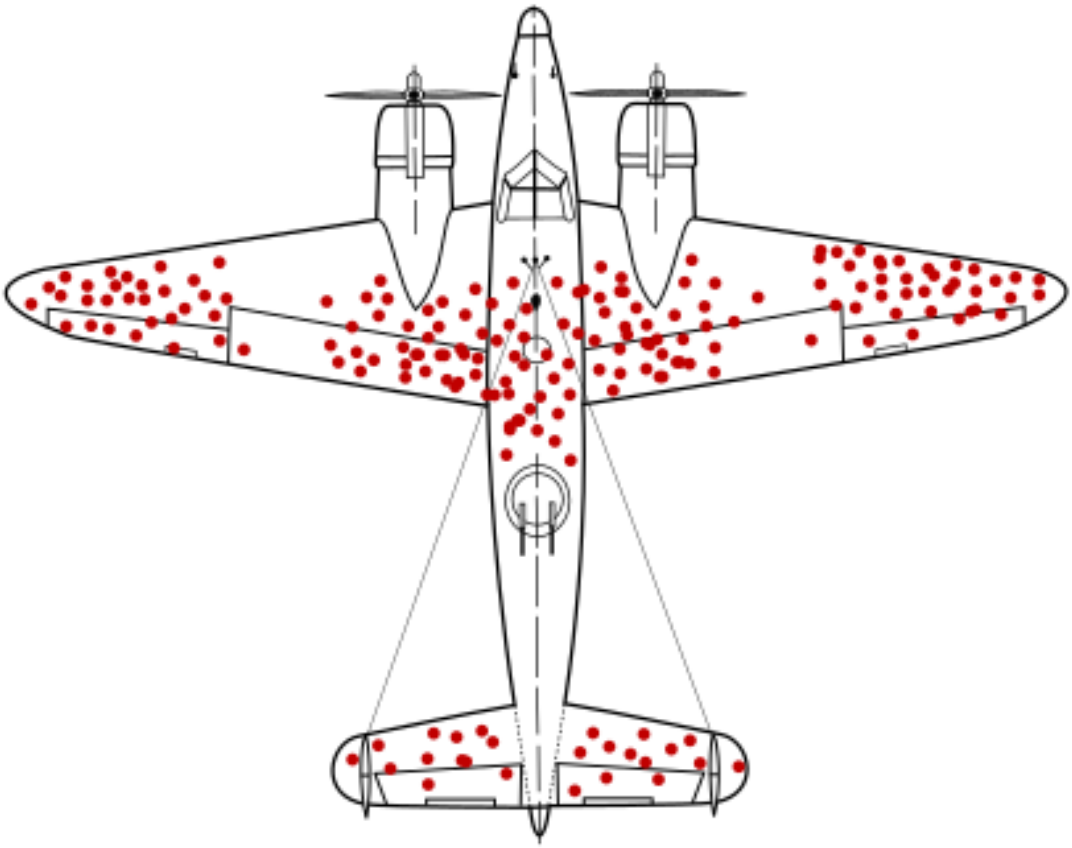


Figure 3: reference: https://en.wikipedia.org/wiki/Survivorship_bias

https://kknews.cc/military/egn6jky.html

reference paper: https://people.ucsc.edu/~msmangel/Wald.pdf

Comparison

Missing Completely at Random (MCAR) - The missing values of a certain variable are completely random and do not depend on any other cause Missing at Random (MAR) - the missing of a certain variable is related to other variables Missing Not at Random (MNAR) - The missing of a certain variable is related to the values of that variable itself, e.g., the instrument's detection limit (left-censored missing)

Conventional Methods

Case deletion

- Complete-case analysis, listwise deletion:
  - Analyze units that are completely observed w.r.t. all variables under consideration
    - Advantage: easy
    - Disadvantage: generally valid only under MCAR, inefficient.
- Available-case analysis, pairwise deletion:
  - Use different sets of sample units for different parameters
    - Advantage: use all available data
    - Disadvantage: difficult to compute SEs because different sets of units are used

Weighting

- Adjustment weighting is one of the most important correction techniques. Every observed unit is assigned a weight, and estimates are based on weighted observations.
  - Weights are derived from probabilities of response
  - Auxiliary information (completely observed and available) is used to make the sample representative for the population
  - Only for MCAR and MAR data

Single Imputation

reference: Schafer and Graham (2002) - unconditional means: filling in means

- unconditional distributions: filling in draws from the observed scores
- conditional means: filling in (model) predictions
  (e.g. with a regression model)
- conditional distributions: filling in draws from the distribution filling in (regression) predictions plus random error

Imputation

- Last observation carried forward, LOCF
- Baseline observation carried forward, BOCF

> **Note**
>
> Imputation and other modern methods such as direct maximum likelihood generally assumes that the data are at least MAR

Advantages:

- More efficient than analyzing complete cases
- Completed data can be analyzed using standard procedures and software

Disadvantages:

- Sometimes hard to implement, especially multivariate cases
- SEs, p values, and other uncertainty measures are misleading because they do not take into account the extra uncertainty caused by missing values

Multiple Imputation

Before applying multiple imputation, we must meet the following assumptions:

- Missing data should be at least missing at random (MAR).
- The missingness of the data is only related to the observed values; missing values are independent and do not affect each other.
- The data follows a multivariate normal distribution (or approaches normality asymptotically)

> **Note**
>
> - Under this approach, the imputation process is repeated several times.
> - Each completed data set is analyzed to obtain estimates of the parameters
>   - their completed-data standard errors and test statistics.
> - The variation across the completed data quantities is used to capture the additional uncertainty due to imputation.

- The completed data estimates, standard errors, test statistics, etc. are combined to form a single inference.

Fraction of missing information, FMI

- These measures are the Fraction of Missing information (FMI), the relative increase in variance due to nonresponse and the Relative Efficiency.
- They are derived from values of the between, and within imputation variance and the total variance.
- There exist two versions of the FMI, which are referred to as lambda and FMI.

Some Practical Issues

- Number of Imputations, M
  - The choice of the number of imputations, M, depends on the fraction of missing information (FMI).
  - Generally, ten imputations may suffice if the fraction of missing information is about 20% and five may be sufficient for a smaller fraction of missing information.

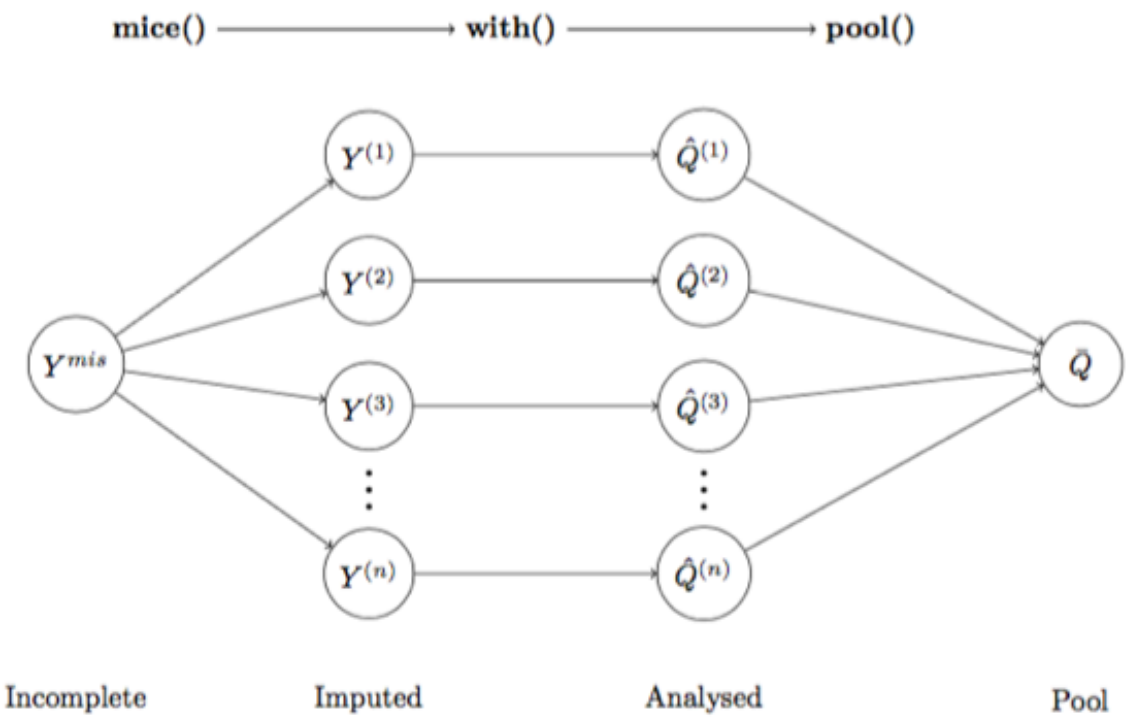
Pos and Cons

- Pos:
  - When the data is missing at random (MAR), correctly applying multiple imputation will produce consistent, asymptotically efficient, and asymptotically normal estimates.
  - It can be employed with almost any type of data or model, and the analysis can be conducted using conventional software.
- Cons:
  - Its implementation can be cumbersome,
  - The most significant drawback is that each time multiple imputation is used, different estimates are generated.

Multivariate Imputation by Chained Equations

- R package: MICE
  - Each variable has its own imputation model. Built-in imputation models are provided for continuous data (predictive mean matching, normal), binary data (logistic regression), unordered categorical data (polytomous logistic regression) and ordered categorical data (proportional odds).
  - MICE can also impute continuous two-level data (normal model, pan, second-level variables). Passive imputation can be used to maintain consistency between variables.
  - Various diagnostic plots are available to inspect the quality of the imputations.

Basic idea




```
library(mice)
imp <- mice(mydata, m)
fit <- with(imp, analysis)
pooled <- pool(fit)
summary(pooled)
```

sleep dataset

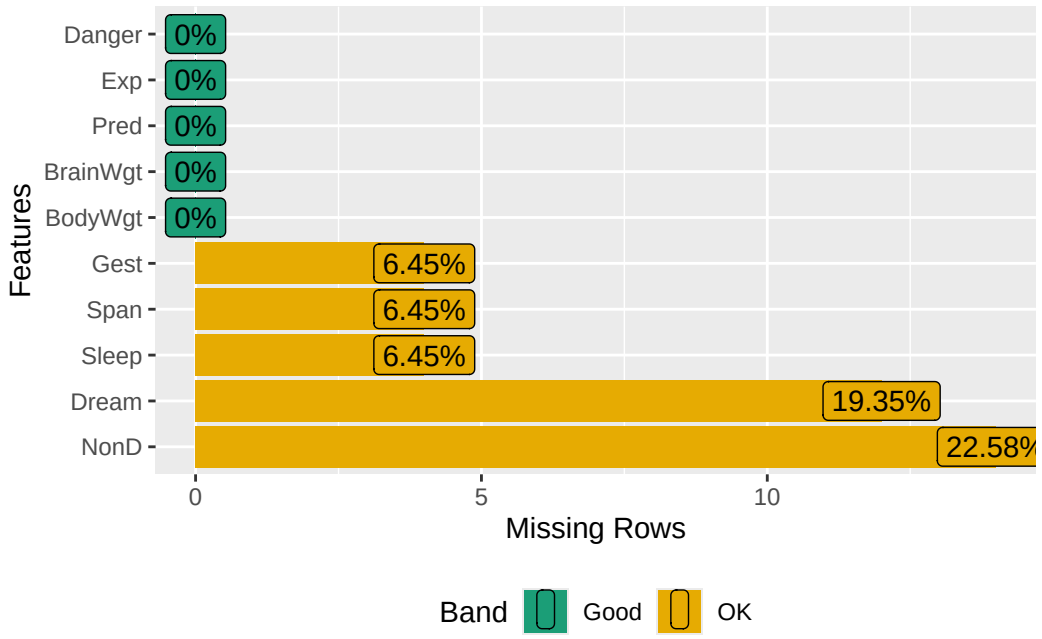
```
library(VIM)
```

```
Warning: package 'VIM' was built under R version 4.3.2
```

```
data(sleep, package="VIM")
head(sleep)
```

```
   BodyWgt BrainWgt NonD Dream Sleep Span Gest Pred Exp Danger
1 6654.000   5712.0   NA    NA   3.3 38.6  645    3   5      3
2    1.000      6.6  6.3   2.0   8.3  4.5   42    3   1      3
3    3.385     44.5   NA    NA  12.5 14.0   60    1   1      1
4    0.920      5.7   NA    NA  16.5   NA   25    5   2      3
5 2547.000   4603.0  2.1   1.8   3.9 69.0  624    3   5      4
6   10.550    179.5  9.1   0.7   9.8 27.0  180    4   4      4
```

```
DataExplorer::plot_missing(sleep)
```



```r
library(mice)
```

```
Warning: package 'mice' was built under R version 4.3.2
```

```r
imp <- mice(sleep, seed=1234)
```

```
 iter imp variable
  1   1  NonD  Dream  Sleep  Span  Gest
  1   2  NonD  Dream  Sleep  Span  Gest
  1   3  NonD  Dream  Sleep  Span  Gest
  1   4  NonD  Dream  Sleep  Span  Gest
  1   5  NonD  Dream  Sleep  Span  Gest
  2   1  NonD  Dream  Sleep  Span  Gest
  2   2  NonD  Dream  Sleep  Span  Gest
  2   3  NonD  Dream  Sleep  Span  Gest
  2   4  NonD  Dream  Sleep  Span  Gest
  2   5  NonD  Dream  Sleep  Span  Gest
  3   1  NonD  Dream  Sleep  Span  Gest
  3   2  NonD  Dream  Sleep  Span  Gest
  3   3  NonD  Dream  Sleep  Span  Gest
  3   4  NonD  Dream  Sleep  Span  Gest
  3   5  NonD  Dream  Sleep  Span  Gest
  4   1  NonD  Dream  Sleep  Span  Gest
  4   2  NonD  Dream  Sleep  Span  Gest
  4   3  NonD  Dream  Sleep  Span  Gest
  4   4  NonD  Dream  Sleep  Span  Gest
  4   5  NonD  Dream  Sleep  Span  Gest
  5   1  NonD  Dream  Sleep  Span  Gest
  5   2  NonD  Dream  Sleep  Span  Gest
  5   3  NonD  Dream  Sleep  Span  Gest
  5   4  NonD  Dream  Sleep  Span  Gest
  5   5  NonD  Dream  Sleep  Span  Gest
```

```
Warning: Number of logged events: 2
```

```r
# The default number of multiple imputations is set to m=5
fit <- with(imp, lm(Dream ~ Span + Gest))
pooled <- pool(fit)
summary(pooled)
```

```
         term     estimate   std.error statistic       df      p.value
1 (Intercept)  2.598553331 0.247119369 10.515377 51.61960 1.949165e-14
2        Span -0.005256987 0.011726809 -0.448288 53.36003 6.557604e-01
3        Gest -0.004050236 0.001495123 -2.708965 48.20381 9.316284e-03
```

```
imp
```

```
Class: mids
Number of multiple imputations:  5
Imputation methods:
 BodyWgt BrainWgt     NonD    Dream    Sleep     Span     Gest     Pred
      ""       ""    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"       ""
     Exp   Danger
      ""       ""
PredictorMatrix:
         BodyWgt BrainWgt NonD Dream Sleep Span Gest Pred Exp Danger
BodyWgt        0        1    1     1     1    1    1    1   1      1
BrainWgt       1        0    1     1     1    1    1    1   1      1
NonD           1        1    0     1     1    1    1    1   1      1
Dream          1        1    1     0     1    1    1    1   1      1
Sleep          1        1    1     1     0    1    1    1   1      1
Span           1        1    1     1     1    0    1    1   1      1
Number of logged events:  2
  it im  dep meth   out
1  3  2 Span  pmm Sleep
2  4  2 Span  pmm Sleep
```

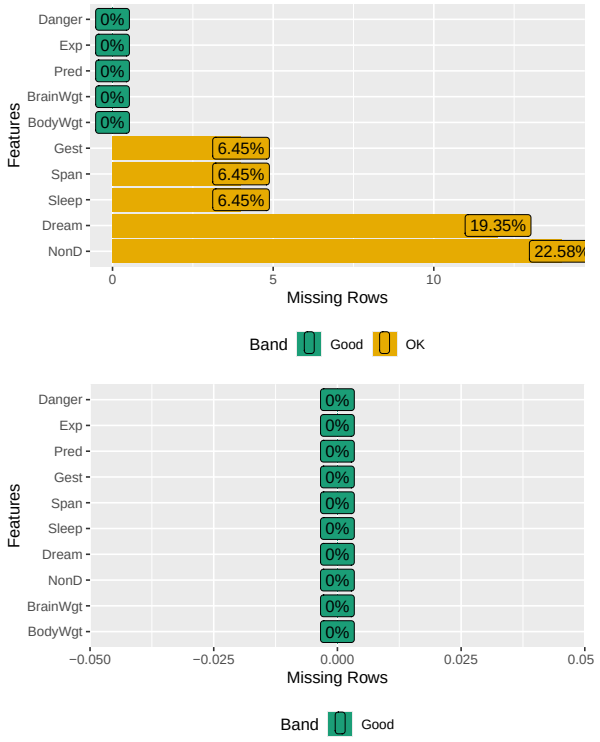
```
# return the original data
mice::complete(imp, action = 0)[1:5,]
```

```
   BodyWgt BrainWgt NonD Dream Sleep Span Gest Pred Exp Danger
1 6654.000   5712.0   NA    NA   3.3 38.6  645    3   5      3
2    1.000      6.6  6.3   2.0   8.3  4.5   42    3   1      3
3    3.385     44.5   NA    NA  12.5 14.0   60    1   1      1
4    0.920      5.7   NA    NA  16.5   NA   25    5   2      3
5 2547.000   4603.0  2.1   1.8   3.9 69.0  624    3   5      4
```

```
# returns the first imputed data set
mice::complete(imp, action = 1)[1:5,]
```

```
   BodyWgt BrainWgt NonD Dream Sleep Span Gest Pred Exp Danger
1 6654.000   5712.0  2.1   0.5   3.3 38.6  645    3   5      3
2    1.000      6.6  6.3   2.0   8.3  4.5   42    3   1      3
3    3.385     44.5 10.9   1.4  12.5 14.0   60    1   1      1
4    0.920      5.7 13.8   2.7  16.5  3.0   25    5   2      3
5 2547.000   4603.0  2.1   1.8   3.9 69.0  624    3   5      4
```

```
DataExplorer::plot_missing(sleep)
DataExplorer::plot_missing(complete(imp))
```



Fraction of Missing Information

- Proportion of the total sampling variance that is due to missing data. So an FMI of 0.08 for write means that 8% of the total sampling variance is attributable to missing data.
- The interpretation is similar to an R-squared. The accuracy of the estimate of FMI increases as the number imputation increases because variance estimates stabilize with larger numbers imputations.

- A high FMI can indicate a problematic variable.
- Bottom line: If FMI is high for any particular variable(s) then consider increasing the number of imputations. In general, the estimation of FMI improves.

```
pooled
```

```
Class: mipo    m = 5
         term m     estimate         ubar            b            t dfcom
1 (Intercept) 5  2.598553331 5.760302e-02 2.887473e-03 6.106798e-02    59
2        Span 5 -0.005256987 1.316819e-04 4.863448e-06 1.375181e-04    59
3        Gest 5 -0.004050236 2.052609e-06 1.523196e-07 2.235393e-06    59
        df        riv     lambda        fmi
1 51.61960 0.06015254 0.05673951 0.09127878
2 53.36003 0.04431996 0.04243906 0.07641920
3 48.20381 0.08904936 0.08176797 0.11763374
```

Predictive Mean Matching

- Predictive Mean Matching (PMM) is a technique of imputation that estimates the likely values of missing data by matching to the observed values/data.
- PMM is a widely used statistical imputation method for missing values, first proposed by Donald B. Rubin in 1986 and R.J.A Little in 1988.
  - Statistical matching using file concatenation with adjusted weights and multiple imputations, Rubin (1986)
  - Missing-data adjustments in large surveys, Little (1988)

> **Note**
>
> The underlying principles of the Predictive Mean Matching (PMM) method:
>
> - Prediction Based on Observed Data:
>   - PMM starts by predicting the missing values using a statistical model. This model leverages the observed data to estimate what the missing values might be.
> - Matching to Observed Values:
>   - Rather than relying solely on the predicted values, PMM places significant emphasis on matching these predictions to the observed values.
>   - The imputation process involves selecting the observed value that is closest to the predicted value.

- Preservation of Distributional Characteristics:
  - One key strength of PMM is its commitment to preserving the distributional characteristics of the observed dataset.
  - By matching imputed values to observed values, PMM ensures that the imputed dataset closely mirrors the original data's patterns and variability.
- Handling Multivariate Relationships:
  - PMM is well-suited for situations where missing data are not only univariate but also involve multivariate relationships.
  - It can be applied to datasets with multiple variables, allowing for a comprehensive approach to imputing missing values.
- Robustness in Various Missing Data Patterns:
  - PMM exhibits robustness in handling different missing data patterns, making it versatile in real-world scenarios.
  - Whether data is missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR), PMM can be applied effectively.
- Applicability in Various Statistical Analyses

! What should I report

- Which statistical program was used to conduct the imputation.
- The type of imputation algorithm used (i.e. MICE).
- Some justification for choosing a particular imputation method.
- The number of imputed datasets (m) created.
- The proportion of missing observations for each imputed variable.
- The variables used in the imputation model and why so your audience will know if you used a more inclusive strategy. This is particularly important when using auxiliary variables.

A Case study

Most materials were adopted from https://argoshare.is.ed.ac.uk/healthyr_book/chap11-h1.html

> **Note**
>
> The R `finalfit` package provides functions that help you quickly create elegant final results tables and plots when modelling in R.

```
library(finalfit)
```

```
Warning: package 'finalfit' was built under R version 4.3.2
```

Ensure your data are coded correctly

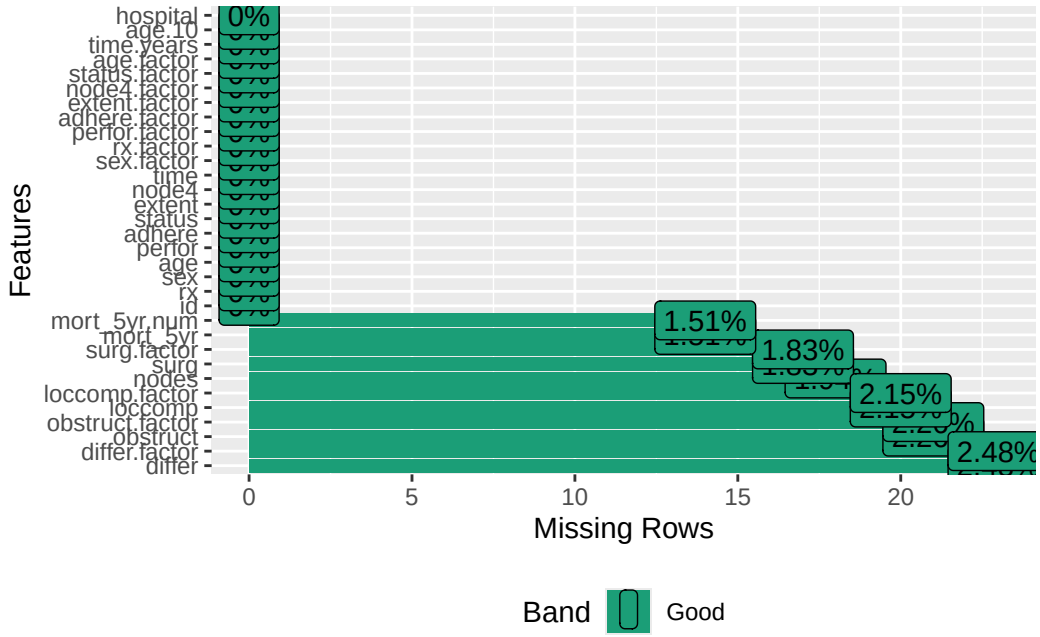
> **Important**
>
> The first step in any analysis is robust data cleaning and coding.
>
> - Ensure all variables are of the type you expect them to be.
>   - Numbers should be numeric, categorical variables should be characters or factors, and dates should be dates
> - Ensure you know which variables have missing data.
> - Ensure factor levels and variable labels are assigned correctly.

load the dataset `colon_s`

```
dim(colon_s) # Chemotherapy for Stage B/C colon cancer
```

```
[1] 929  32
```

```
DataExplorer::plot_missing(colon_s)
```




```
str(colon_s) # Display the Structure
```

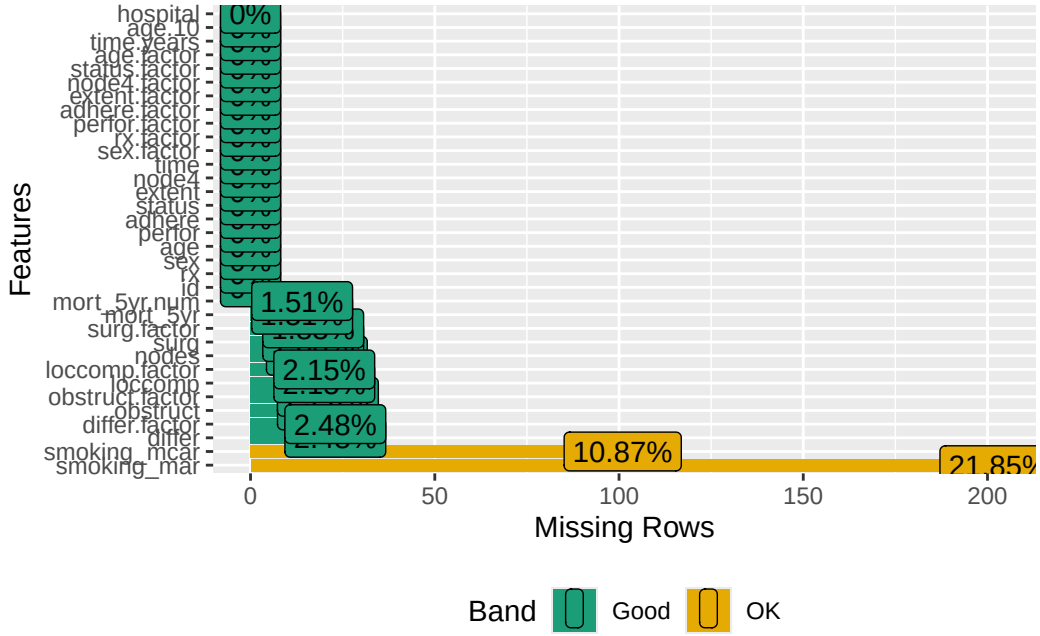
```
# Create some extra missing data
library(finalfit)
library(dplyr)
set.seed(1)
colon_s <- colon_s %>%
  mutate(
    ## Smoking missing
    ### completely at random
    smoking_mcar = sample(c("Smoker", "Non-smoker", NA),
                          n(), replace = TRUE,
                          prob = c(0.2, 0.7, 0.1)) %>%
      factor() %>%
      ff_label("Smoking (MCAR)"),

    ## Smoking missing
    ## conditional on patient sex
    smoking_mar = ifelse(sex.factor == "Female",
                         sample(c("Smoker", "Non-smoker", NA),
                                sum(sex.factor == "Female"),
                                replace = TRUE,
                                prob = c(0.1, 0.5, 0.4)),
```

```
                        sample(c("Smoker", "Non-smoker", NA),
                               sum(sex.factor == "Male"),
                               replace=TRUE, prob = c(0.15, 0.75, 0.1))
    ) %>%
      factor() %>%
      ff_label("Smoking (MAR)")
  )
```

```
dim(colon_s) # Chemotherapy for Stage B/C colon cancer
```

```
[1] 929  34
```

```
DataExplorer::plot_missing(colon_s)
```



Use the `ff_glimpse()` function to examine the variables of interest.

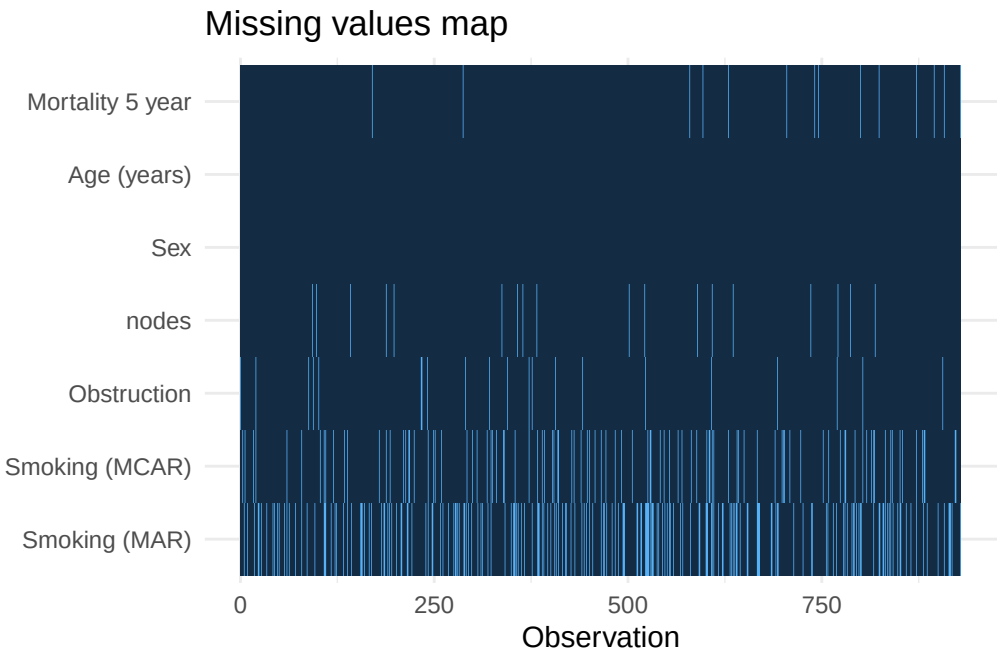
```
explanatory <- c("age", "sex.factor", "nodes", "obstruct.factor",
                 "smoking_mcar", "smoking_mar")
dependent <- "mort_5yr"
```

> **Note**
>
> - nodes: number of lymph nodes with detectable cancer
> - obstruct: obstruction of colon by tumour

```
colon_s %>%
  ff_glimpse(dependent, explanatory)
```

```
$Continuous
            label var_type   n missing_n missing_percent mean   sd  min
age   Age (years)    <dbl> 929         0             0.0 59.8 11.9 18.0
nodes       nodes    <dbl> 911        18             1.9  3.7  3.6  0.0
      quartile_25 median quartile_75  max
age          53.0   61.0        69.0 85.0
nodes         1.0    2.0         5.0 33.0

$Categorical
                          label var_type   n missing_n missing_percent
mort_5yr        Mortality 5 year    <fct> 915        14             1.5
sex.factor                  Sex    <fct> 929         0             0.0
obstruct.factor      Obstruction    <fct> 908        21             2.3
smoking_mcar      Smoking (MCAR)    <fct> 828       101            10.9
smoking_mar        Smoking (MAR)    <fct> 726       203            21.9
                levels_n                                levels  levels_count
mort_5yr               2        "Alive", "Died", "(Missing)"  511, 404, 14
sex.factor             2       "Female", "Male", "(Missing)"      445, 484
obstruct.factor        2            "No", "Yes", "(Missing)"  732, 176, 21
smoking_mcar           2 "Non-smoker", "Smoker", "(Missing)" 645, 183, 101
smoking_mar            2 "Non-smoker", "Smoker", "(Missing)" 585, 141, 203
                  levels_percent
mort_5yr        55.0, 43.5,  1.5
sex.factor                48, 52
obstruct.factor 78.8, 18.9,  2.3
smoking_mcar          69, 20, 11
smoking_mar           63, 15, 22
```

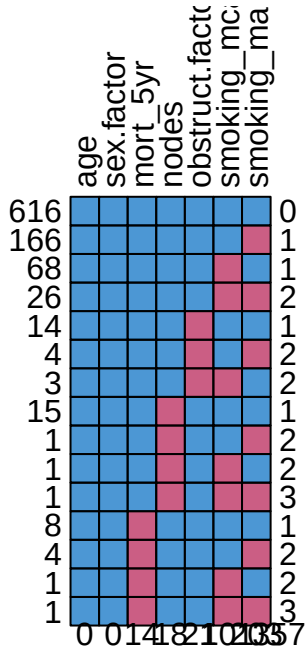
```
colon_s %>%
  finalfit::missing_plot(dependent, explanatory)
```



Look for ptterns of missingness

> **Note**
>
> The `missing_pattern()` function wraps the function `md.pattern()` in the `mice` package, which is used to show the pattern of missingness between variables.

```
colon_s %>%
  finalfit::missing_pattern(dependent, explanatory)
```



	age	sex.factor	mort_5yr	nodes	obstruct.factor	smoking_mcar	smoking_mar	
616	1	1	1	1	1	1	1	0
166	1	1	1	1	1	1	0	1
68	1	1	1	1	1	0	1	1
26	1	1	1	1	1	0	0	2
14	1	1	1	1	0	1	1	1
4	1	1	1	1	0	1	0	2
3	1	1	1	1	0	0	1	2
15	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	0	2
1	1	1	1	0	1	0	1	2
1	1	1	1	0	1	0	0	3
8	1	1	0	1	1	1	1	1
4	1	1	0	1	1	1	0	2
1	1	1	0	1	1	0	1	2
1	1	1	0	1	1	0	0	3
	0	0	14	18	21	101	203	357

Tip

- A matrix with ncol(x)+1 columns, in which each row corresponds to a missing data pattern (1=observed, 0=missing).

- Rows and columns are sorted in increasing amounts of missing information.
- The last column and row contain row and column counts, respectively.

- There are 11 patterns in these data.
- The number and pattern of missingness help us to determine the likelihood of it being random rather than systematic.

Including missing data in demographics tables

Tip

Table 1 in a healthcare study is often a demographics table of an **explanatory variable of interest** against other explanatory variables.

The function `summary_factorlist()` provides a useful summary of a dependent variable against explanatory variables.

```
table1 <- colon_s %>%
  summary_factorlist(dependent, explanatory,
                     na_include=TRUE, na_include_dependent = TRUE,
                     total_col = TRUE, add_col_totals = TRUE, p=TRUE,
                     p_cont_para = "aov",
                     p_cat = "chisq")
knitr::kable(table1,
             caption = "Simulated missing completely
             at random (MCAR) and missing at random (MAR) dataset.")
```

Table 1: Simulated missing completely at random (MCAR) and missing at random (MAR) dataset.

label	levels	Alive	Died	(Missing)	Total	p
Total N (%)		511 (55.0)	404 (43.5)	14 (1.5)	929	
Age (years)	Mean (SD)	59.8 (11.4)	59.9 (12.5)	53.9 (12.7)	59.8 (11.9)	0.986
Sex	Female	243 (47.6)	194 (48.0)	8 (57.1)	445 (47.9)	0.941
	Male	268 (52.4)	210 (52.0)	6 (42.9)	484 (52.1)	
	(Missing)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	

label	levels	Alive	Died	(Missing)	Total	p
nodes	Mean (SD)	2.7 (2.4)	4.9 (4.4)	2.9 (2.8)	3.7 (3.6)	<0.001
Obstruction	No	408 (79.8)	312 (77.2)	12 (85.7)	732 (78.8)	0.219
	Yes	89 (17.4)	85 (21.0)	2 (14.3)	176 (18.9)	
	(Missing)	14 (2.7)	7 (1.7)	0 (0.0)	21 (2.3)	
Smoking (MCAR)	Non-smoker	358 (70.1)	277 (68.6)	10 (71.4)	645 (69.4)	0.133
	Smoker	90 (17.6)	91 (22.5)	2 (14.3)	183 (19.7)	
	(Missing)	63 (12.3)	36 (8.9)	2 (14.3)	101 (10.9)	
Smoking (MAR)	Non-smoker	312 (61.1)	266 (65.8)	7 (50.0)	585 (63.0)	0.082
	Smoker	87 (17.0)	52 (12.9)	2 (14.3)	141 (15.2)	
	(Missing)	112 (21.9)	86 (21.3)	5 (35.7)	203 (21.9)	

- `na_include=TRUE` ensures missing data from the explanatory variables (but not dependent) are included.
- `na_include_dependent = TRUE` ensures missing data from the dependent.
- `total_col = TRUE` include a total column

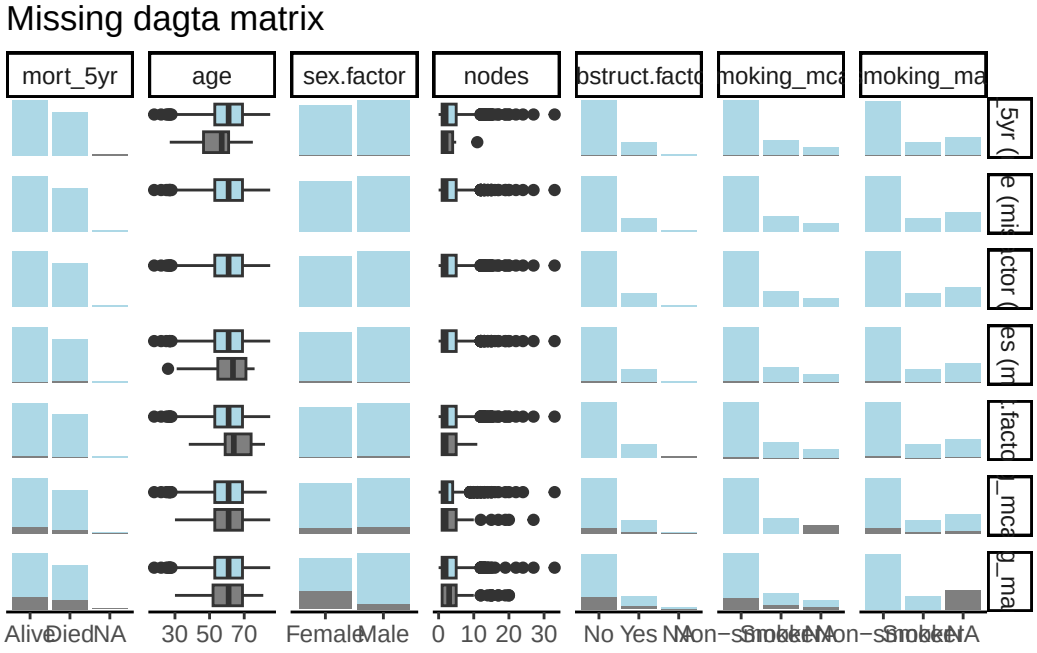
check for associations bewtwwen missing and observed data

> Tip
>
> - To assess whether the data follows MCAR or MAR, one can examine missingness patterns across different levels of included variables.
> - This holds particular significance when dealing with a primary outcome measure or dependent variable.

`missing_pairs()` uses functions from the `GGally` package. It produces pairs plots to show relationships between missing values and observed values in all variables.

> Tip
>
> 1. Suggest limit the number of variables to a maximum of around six.
> 2. For continuous variables, the distributions of observed and missing data can be visually compared.
> 3. For categorical data, the comparisons are presented as counts or proportions.

```
dependent <- "mort_5yr"
explanatory <- c("age", "sex.factor", "nodes", "obstruct.factor",
                 "smoking_mcar", "smoking_mar")
colon_s %>%
  missing_pairs(dependent, explanatory,
                title = "Missing dagta matrix",
                use_labels = FALSE,
                showYAxisPlotLabels = FALSE)
```



Findings:

The two sets of bar plots that show the proportion of missing smoking data for sex.

- Missingness in Smoking (MCAR) does not relate to sex - females and males have the same proportion of missing
- Missingness in Smoking (MAR), does differ by sex as females have more missing data than men here.

One can use the `missing_compare()` function to confirm the result.

Tip

`missing_compare()` uses an F-test test for continuous variables and chi-squared for categorical variables.

```
explanatory <- c("age", "sex.factor",
                 "nodes", "obstruct.factor")
dependent <- "smoking_mcar"
missing_mcar <- colon_s %>%
  missing_compare(dependent, explanatory)
knitr::kable(missing_mcar)
```

Missing data analysis: Smoking (MCAR)		Not missing	Missing	p
Age (years)	Mean (SD)	59.7 (11.9)	59.9 (12.6)	0.882
Sex	Female	399 (89.7)	46 (10.3)	0.692
	Male	429 (88.6)	55 (11.4)	
nodes	Mean (SD)	3.6 (3.4)	4.0 (4.5)	0.302
Obstruction	No	654 (89.3)	78 (10.7)	0.891
	Yes	156 (88.6)	20 (11.4)	

```
explanatory <- c("age", "sex.factor",
                 "nodes", "obstruct.factor")
dependent <- "smoking_mar"
missing_mcar <- colon_s %>%
  missing_compare(dependent, explanatory)
knitr::kable(missing_mcar)
```

Missing data analysis: Smoking (MAR)		Not missing	Missing	p
Age (years)	Mean (SD)	59.9 (11.8)	59.4 (12.6)	0.632
Sex	Female	288 (64.7)	157 (35.3)	<0.001
	Male	438 (90.5)	46 (9.5)	
nodes	Mean (SD)	3.6 (3.5)	3.9 (3.9)	0.321

Missing data analysis: Smoking (MAR)		Not missing	Missing	p
Obstruction	No	568 (77.6)	164 (22.4)	0.533
	Yes	141 (80.1)	35 (19.9)	

Tip

1. It takes dependent and explanatory variables, and in this context "dependent" refers to the variable being tested for missingness against the explanatory variables.
2. A relationship is seen between sex and smoking (MAR) but not smoking (MCAR).

Handling missing data

Prior to a standard regression analysis, we can either:

1. Delete the cases with the missing data (row-wise deletion)
2. Delete the variable with the missing data
3. Impute (fill in) the missing data
4. Model the missing data

List-wise deletion

Tip

Depending on the number of data points that are missing, we may have sufficient power with complete cases to examine the relationships of interest.

```
explanatory <- c("age", "sex.factor",
                 "nodes", "obstruct.factor",
                 "smoking_mcar")
dependent <- "mort_5yr"
fit = colon_s %>%
  finalfit(dependent, explanatory)
```

```
Note: dependent includes missing data. These are dropped.
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
```

```
knitr::kable(fit, caption = "Regression analysis with missing data: List-wise deletion")
```

Table 4: Regression analysis with missing data: List-wise deletion

	Dependent: Mortality 5 year		Alive	Died	OR (univariable)	OR (multivariable)
1	Age (years)	Mean (SD)	59.8 (11.4)	59.9 (12.5)	1.00 (0.99-1.01, p=0.986)	1.01 (1.00-1.02, p=0.200)
5	Sex	Female	243 (55.6)	194 (44.4)	-	-
6		Male	268 (56.1)	210 (43.9)	0.98 (0.76-1.27, p=0.889)	1.02 (0.76-1.38, p=0.872)
2	nodes	Mean (SD)	2.7 (2.4)	4.9 (4.4)	1.24 (1.18-1.30, p<0.001)	1.25 (1.18-1.33, p<0.001)
3	Obstruction	No	408 (56.7)	312 (43.3)	-	-
4		Yes	89 (51.1)	85 (48.9)	1.25 (0.90-1.74, p=0.189)	1.53 (1.05-2.22, p=0.027)
7	Smoking (MCAR)	Non-smoker	358 (56.4)	277 (43.6)	-	-
8		Smoker	90 (49.7)	91 (50.3)	1.31 (0.94-1.82, p=0.113)	1.37 (0.96-1.96, p=0.083)

Omit the variable

> **Note**
>
> If the variable does not appear to be important, it may be omitted from the analysis.

Model the missing data for the categorical variable

There is an alternative method to model missing data for the **categorical variable**: simply consider the missing data as a factor level.

> **Note**
>
> This has the advantage of simplicity, with the disadvantage of increasing the number of terms in the model.

```
library(dplyr)
explanatory = c("age", "sex.factor",
                "nodes", "obstruct.factor", "smoking_mar")
fit_explicit_na = colon_s %>%
  mutate(
    smoking_mar = forcats::fct_na_value_to_level(smoking_mar)) %>%
  finalfit(dependent, explanatory)
```

```
Note: dependent includes missing data. These are dropped.
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
```

```
knitr::kable(fit_explicit_na, row.names = FALSE)
```

Dependent: Mortality 5 year		Alive	Died	OR (univariable)	OR (multivariable)
Age (years)	Mean (SD)	59.8 (11.4)	59.9 (12.5)	1.00 (0.99-1.01, p=0.986)	1.01 (1.00-1.02, p=0.114)
Sex	Female	243 (55.6)	194 (44.4)	-	-
	Male	268 (56.1)	210 (43.9)	0.98 (0.76-1.27, p=0.889)	0.95 (0.71-1.28, p=0.743)
nodes	Mean (SD)	2.7 (2.4)	4.9 (4.4)	1.24 (1.18-1.30, p<0.001)	1.25 (1.19-1.32, p<0.001)

Dependent: Mortality 5 year		Alive	Died	OR (univariable)	OR (multivariable)
Obstruction	No	408 (56.7)	312 (43.3)	-	-
	Yes	89 (51.1)	85 (48.9)	1.25 (0.90-1.74, p=0.189)	1.35 (0.95-1.92, p=0.099)
Smoking (MAR)	Non-smoker	312 (54.0)	266 (46.0)	-	-
	Smoker	87 (62.6)	52 (37.4)	0.70 (0.48-1.02, p=0.067)	0.78 (0.52-1.17, p=0.233)
		112 (56.6)	86 (43.4)	0.90 (0.65-1.25, p=0.528)	0.85 (0.59-1.23, p=0.390)

Multivariate Imputation

> **!** Important
>
> - If we simply drop all the patients for whom smoking is missing (list-wise deletion), then we drop relatively more females than men.
> - This may have consequences for our conclusions if sex is associated with our explanatory variable of interest or outcome.
> - Imputation is not usually appropriate for the outcome variable.

The process of multiple imputation involves:

- Impute missing data m times, which results in m complete datasets
- Diagnose the quality of the imputed values
- Analyse each completed dataset
- Pool the results of the repeated analyses

```
library(finalfit)
library(dplyr)
library(mice)
explanatory <- c("age", "sex.factor",
                 "nodes", "obstruct.factor", "smoking_mar")
dependent <- "mort_5yr"
# Choose which variable to input missing values
colon_s %>%
  select(dependent, explanatory) %>%
  missing_predictorMatrix(
```

```
    drop_from_imputed = c("mort_5yr")
  ) -> predM
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(dependent)

  # Now:
  data %>% select(all_of(dependent))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(explanatory)

  # Now:
  data %>% select(all_of(explanatory))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

Make 10 imputed datasets and run our logistic regression analysis on each set.

```
fits <- colon_s %>%
  select(dependent, explanatory) %>%
  # run imputation with 10 imputed sets
  mice(m = 10, predictorMatrix = predM) %>%
  # Run logistic regression on each imputed set
  with(glm( formula(ff_formula(dependent, explanatory)),
            family="binomial"))
```

```
 iter imp variable
  1   1  mort_5yr  nodes  obstruct.factor  smoking_mar
  1   2  mort_5yr  nodes  obstruct.factor  smoking_mar
  1   3  mort_5yr  nodes  obstruct.factor  smoking_mar
  1   4  mort_5yr  nodes  obstruct.factor  smoking_mar
  1   5  mort_5yr  nodes  obstruct.factor  smoking_mar
  1   6  mort_5yr  nodes  obstruct.factor  smoking_mar
```

```
1   7  mort_5yr  nodes  obstruct.factor  smoking_mar
1   8  mort_5yr  nodes  obstruct.factor  smoking_mar
1   9  mort_5yr  nodes  obstruct.factor  smoking_mar
1   10  mort_5yr  nodes  obstruct.factor  smoking_mar
2   1  mort_5yr  nodes  obstruct.factor  smoking_mar
2   2  mort_5yr  nodes  obstruct.factor  smoking_mar
2   3  mort_5yr  nodes  obstruct.factor  smoking_mar
2   4  mort_5yr  nodes  obstruct.factor  smoking_mar
2   5  mort_5yr  nodes  obstruct.factor  smoking_mar
2   6  mort_5yr  nodes  obstruct.factor  smoking_mar
2   7  mort_5yr  nodes  obstruct.factor  smoking_mar
2   8  mort_5yr  nodes  obstruct.factor  smoking_mar
2   9  mort_5yr  nodes  obstruct.factor  smoking_mar
2   10  mort_5yr  nodes  obstruct.factor  smoking_mar
3   1  mort_5yr  nodes  obstruct.factor  smoking_mar
3   2  mort_5yr  nodes  obstruct.factor  smoking_mar
3   3  mort_5yr  nodes  obstruct.factor  smoking_mar
3   4  mort_5yr  nodes  obstruct.factor  smoking_mar
3   5  mort_5yr  nodes  obstruct.factor  smoking_mar
3   6  mort_5yr  nodes  obstruct.factor  smoking_mar
3   7  mort_5yr  nodes  obstruct.factor  smoking_mar
3   8  mort_5yr  nodes  obstruct.factor  smoking_mar
3   9  mort_5yr  nodes  obstruct.factor  smoking_mar
3   10  mort_5yr  nodes  obstruct.factor  smoking_mar
4   1  mort_5yr  nodes  obstruct.factor  smoking_mar
4   2  mort_5yr  nodes  obstruct.factor  smoking_mar
4   3  mort_5yr  nodes  obstruct.factor  smoking_mar
4   4  mort_5yr  nodes  obstruct.factor  smoking_mar
4   5  mort_5yr  nodes  obstruct.factor  smoking_mar
4   6  mort_5yr  nodes  obstruct.factor  smoking_mar
4   7  mort_5yr  nodes  obstruct.factor  smoking_mar
4   8  mort_5yr  nodes  obstruct.factor  smoking_mar
4   9  mort_5yr  nodes  obstruct.factor  smoking_mar
4   10  mort_5yr  nodes  obstruct.factor  smoking_mar
5   1  mort_5yr  nodes  obstruct.factor  smoking_mar
5   2  mort_5yr  nodes  obstruct.factor  smoking_mar
5   3  mort_5yr  nodes  obstruct.factor  smoking_mar
5   4  mort_5yr  nodes  obstruct.factor  smoking_mar
5   5  mort_5yr  nodes  obstruct.factor  smoking_mar
5   6  mort_5yr  nodes  obstruct.factor  smoking_mar
5   7  mort_5yr  nodes  obstruct.factor  smoking_mar
5   8  mort_5yr  nodes  obstruct.factor  smoking_mar
5   9  mort_5yr  nodes  obstruct.factor  smoking_mar
```

```
  5   10  mort_5yr  nodes  obstruct.factor  smoking_mar
```

Extract metrics from each model

```
# Examples of extracting metrics from fits and taking the mean
## AICs
fits %>%
  getfit() %>%
  purrr::map(AIC) %>%
  unlist() %>%
  mean()
```

```
[1] 1193.216
```

> Akaike information criterion, AIC (Source: Wike)
>
> - AIC is an estimator of prediction error and thereby relative quality of statistical models for a given set of data.Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection.
> - AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model.

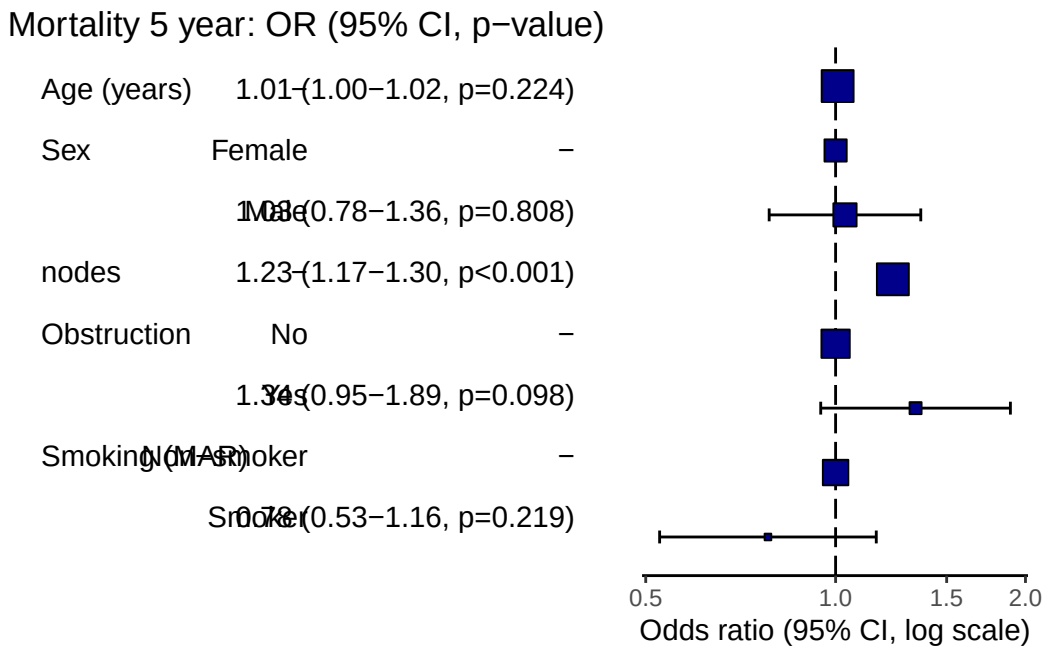
Pool models together

```
# Pool  results
fits_pool <- fits %>%
  pool()
knitr::kable(fits_pool$pooled)
```

term	m	estimate	ubar	b	t	dfcom	df	riv	lambda	fmi
(Intercept)	10	-1.4501982	0.1566905	0.0095674	0.1672147	923	625.4584	0.0671653	0.0629380	0.0659201
age	10	0.0073772	0.0000355	0.0000010	0.0000368	923	789.0676	0.0369910	0.0356710	0.0381067
sex.factorMale	10	0.0342079	0.0194230	0.0004106	0.0198746	923	855.8753	0.0232530	0.0227246	0.0250003
nodes	10	0.2089765	0.0006534	0.0000270	0.0006832	923	743.4049	0.0454546	0.0434783	0.0460413
obstruct.factorYes	10	0.2919570	0.0304573	0.0005987	0.0311159	923	862.7930	0.0216240	0.0211663	0.0234274
smoking_marSmoker	10	-0.2469609	0.0073386	0.0040136	0.0117522	923	170.8304	4.2517621	0.8011261	0.8210317

```
## Can be passed to or_plot
colon_s %>%
  or_plot(dependent, explanatory, glmfit = fits_pool, table_text_size=4)
```

```
Warning: Removed 3 rows containing missing values or values outside the scale range
(`geom_errorbarh()`).
```



Summarise and put in table

```
fit_imputed <- fits_pool %>%
  fit2df(estimate_name = "OR (multiple imputation)", exp = TRUE)

# Use finalfit merge methods to create and compare results
explanatory <- c("age", "sex.factor",
                 "nodes", "obstruct.factor", "smoking_mar")

table_uni_multi <- colon_s %>%
  finalfit(dependent, explanatory, keep_fit_id = TRUE)
```

```
Note: dependent includes missing data. These are dropped.
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
```

```
explanatory = c("age", "sex.factor",
                "nodes", "obstruct.factor")

fit_multi_no_smoking <- colon_s %>%
  glmmulti(dependent, explanatory) %>%
  fit2df(estimate_suffix = " (multivariable without smoking)")
```

```
Waiting for profiling to be done...
```

```
# Combine to final table
table_imputed <-
  table_uni_multi %>%
  ff_merge(fit_multi_no_smoking) %>%
  ff_merge(fit_imputed, last_merge = TRUE)
knitr::kable(table_imputed, row.names = FALSE)
```

Dependent: Mortality 5 year		Alive	Died	OR (univariable)	OR (multivariable)	OR (multivariable without smoking)	OR (multiple imputation)
Age (years)	Mean (SD)	59.8 (11.4)	59.9 (12.5)	1.00 (0.99-1.01, p=0.986)	1.02 (1.01-1.04, p=0.004)	1.01 (1.00-1.02, p=0.122)	1.01 (1.00-1.02, p=0.224)
Sex	Female	243 (55.6)	194 (44.4)	-	-	-	-
	Male	268 (56.1)	210 (43.9)	0.98 (0.76-1.27, p=0.889)	0.97 (0.69-1.34, p=0.836)	0.98 (0.74-1.30, p=0.890)	1.03 (0.78-1.36, p=0.808)
nodes	Mean (SD)	2.7 (2.4)	4.9 (4.4)	1.24 (1.18-1.30, p<0.001)	1.28 (1.21-1.37, p<0.001)	1.25 (1.19-1.32, p<0.001)	1.23 (1.17-1.30, p<0.001)

Dependent: Mortality 5 year		Alive	Died	OR (univariable)	OR (multivariable)	OR (multivariable without smoking)	OR (multiple imputation)
Obstruction	No	408 (56.7)	312 (43.3)	-	-	-	-
	Yes	89 (51.1)	85 (48.9)	1.25 (0.90-1.74, p=0.189)	1.49 (1.00-2.22, p=0.052)	1.36 (0.95-1.93, p=0.089)	1.34 (0.95-1.89, p=0.098)
Smoking (MAR)	Non-smoker	312 (54.0)	266 (46.0)	-	-	-	-
	Smoker	87 (62.6)	52 (37.4)	0.70 (0.48-1.02, p=0.067)	0.77 (0.51-1.16, p=0.221)	-	0.78 (0.53-1.16, p=0.219)

Little, Roderick JA. 1988. "Missing-Data Adjustments in Large Surveys." Journal of Business & Economic Statistics 6 (3): 287–96.

Rubin, Donald B. 1986. "Statistical Matching Using File Concatenation with Adjusted Weights and Multiple Imputations." Journal of Business & Economic Statistics 4 (1): 87–94.

Schafer, Joseph L, and John W Graham. 2002. "Missing Data: Our View of the State of the Art." Psychological Methods 7 (2): 147.