Summary of "A Comparison of 12 Algorithms for Matching on the Propensity Score"

RE6124019_ 吳明軒

2024-04-30

Table of contents

1 Introduction

1.1 Steps of dealing with missing data

1. Understanding the reasons for non-responding
2. Implemnting effective follow-up procedures
3. Utilizing imputation techniques
4. Analyzing patterns of missing data
5. Sensitivity analysis
6. Reporting limitations

1.2 Missing Data Patterns

1.2.1 1. Univariate Pattern

- **Description**: Missing values occur on a single item. This item is either completely observed or missing across all observations.
- **Example**: In a dataset with multiple variables, only one variable, such as 'Age', might have missing values, while all other variables are fully observed.

1.2.2 2. Multivariate Pattern

- **Description**: Missing values occur across a group of items, and these items are either completely observed or missing together.
- **Example**: If data for 'Blood Pressure' and 'Cholesterol Levels' are missing, they are missing together for some observations but fully observed for others.

1.2.3 3. Monotone Pattern

- **Description**: Items are ordered such that if a particular item (p) is missing, then all subsequent items (p+1 to k) are also missing.
- **Example**: In longitudinal studies, if a participant misses a follow-up visit (time point p), all subsequent follow-up data (time points p+1 to k) will also be missing.

1.2.4 4. Arbitrary Pattern

- **Description**: Missing data are randomly scattered throughout the dataset without any systematic pattern or order.
- **Example**: Missing values occur unpredictably across various variables and observations with no discernible pattern.

1.3 Missing Data Mechanisms

1.3.1 1. Missing Completely at Random (MCAR)

- **Definition**: Missing data does not depend on any observed or unobserved data within the dataset.
- **Example**: Smoking status is randomly missing among male and female patients.
- **Characteristics**: Handling MCAR is straightforward because the missingness introduces no bias related to the data's observed or missing values.

1.3.2 2. Missing at Random (MAR)

- **Definition**: The missingness of a variable is related to other observed variables in the dataset, but not to the values of the variable itself.
- **Example**: Smoking status is missing more frequently for female patients but is not dependent on whether the females are smokers or non-smokers.
- **Characteristics**: MAR requires statistical techniques that use the relationships among variables to handle the missing data.

1.3.3 3. Missing Not at Random (MNAR)

- **Definition**: The probability of a data point being missing is related to its value, representing a systematic loss of information.
- **Example**: Smoking status is more likely to be missing for female patients who are smokers.
- **Characteristics**: MNAR is the most challenging to address as it can bias the study results and requires sophisticated statistical methods to manage.

1.4 Conventional Methods for Handling Missing Data

1.4.1 Complete Case Analysis (Listwise Deletion)

- **Description**: This method involves removing any cases (rows) that have missing values in any field, using only complete cases for analysis.
- **Advantages**:
  - Simple and easy to implement.
- **Disadvantages**:
  - Can significantly reduce the amount of data available.
  - May introduce bias if the missing data are not Missing Completely at Random (MCAR).

1.4.2 Available Case Analysis (Pairwise Deletion)

- **Description**: Utilizes observations that have recorded values for the variables required in specific statistical analyses, even if other variables in the same observation are missing.
- **Advantages**:
  - Allows the use of more data by not completely excluding observations that have partial missing data.
- **Disadvantages**:
  - Can lead to inaccurate statistical estimates when the pattern of missingness varies across variables, increasing computational complexity.

1.4.3 Weighting

- **Description**: Adjusts for the impact of missing data by assigning different weights to the observed cases, aiming to compensate for the information loss due to non-response.
- **Advantages**:
  - Allows for some inference from non-responded data.
  - Weights are usually calculated based on response probabilities or other relevant information.
- **Disadvantages**:
  - Requires a good understanding of the missing data mechanism.
  - Relies on accurate estimation of how weights should be calculated based on the mechanism.

2 Imputation Methods

2.1 Single Imputation

Single imputation involves using a single value to substitute each missing data point in the dataset. Here are the primary types used:

- **Unconditional Means**: Filling in missing values using the mean of the entire dataset.
- **Unconditional Distributions**: Filling in missing values by drawing randomly from the observed scores.
- **Conditional Means**: Filling in missing values using predictions from a model, such as a regression model.
- **Conditional Distributions**: Filling in missing values based on model predictions plus a random error component.

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

Disadvantages: - Can be challenging to implement, especially in multivariate cases. - Standard errors, p-values, and other measures of uncertainty can be misleading as they do not account for the additional uncertainty introduced by missing values.

2.2 Multiple Imputation

Multiple imputation addresses missing data by repeating the imputation process multiple times to reflect the uncertainty introduced by the imputation. This process typically involves the following assumptions:

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

Steps:

1. Impute missing data multiple times, generating several complete datasets.
2. Analyze each dataset to obtain estimates of the parameters.
3. Use the variation across completed data sets to capture the additional uncertainty due to imputation.
4. Combine estimates, standard errors, test statistics, etc., to form a single inference.

Advantages: - When correctly applied, produces consistent, asymptotically efficient, and asymptotically normal estimates. - Applicable to almost any type of data or model, and analysis can be performed using conventional software.

Disadvantages: - Implementation can be cumbersome. - Each use of multiple imputation generates different estimates.

3 Implementation Example

```
library(VIM)
library(DataExplorer)
library(mice)
library(finalfit)
library(dplyr)
```

3.0.1 Sleep Data

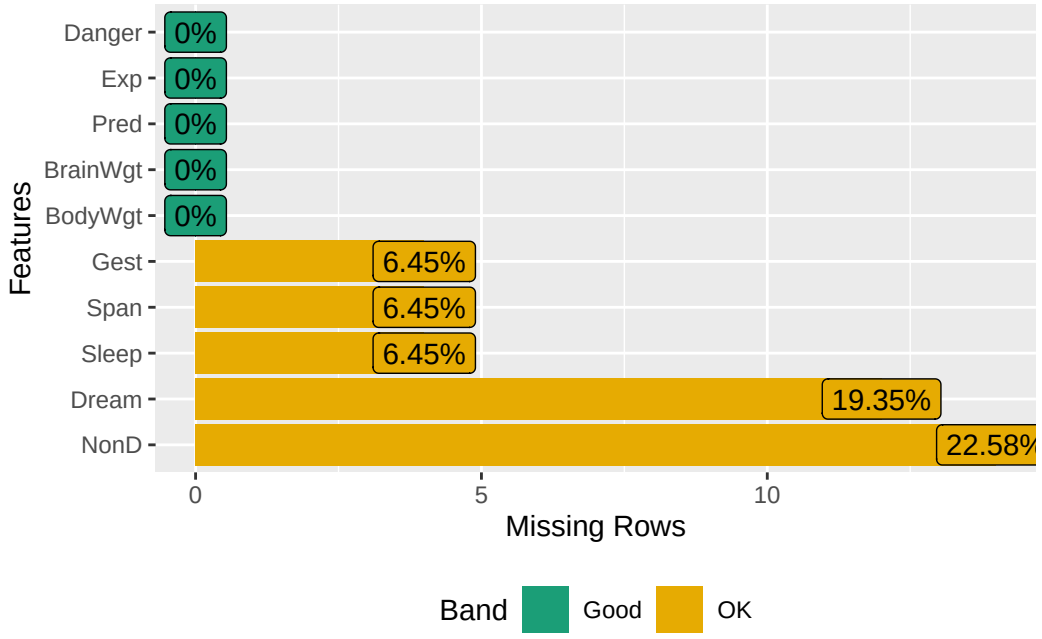
```r
# Loading and examining sleep data
data(sleep, package="VIM")
head(sleep)
```

```
   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
```

```r
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 ...
```

```r
# Plotting missing data
DataExplorer::plot_missing(sleep)
```




```
# Implementing multiple imputation
imp <- mice(sleep, seed=1234, m=5)
```

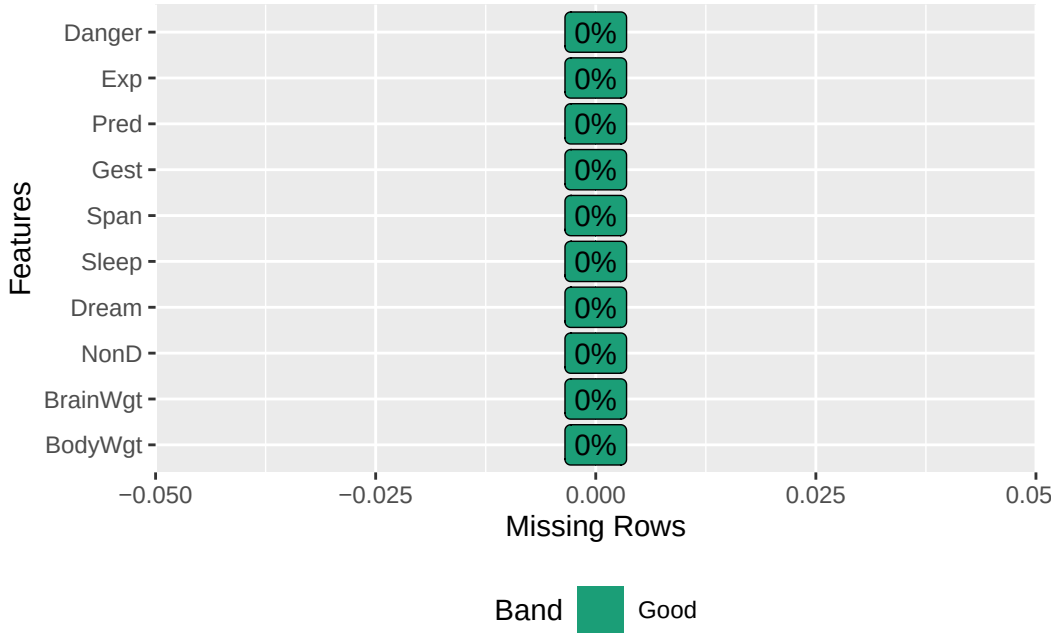
```
 iter imp variable
  1   1  NonD  Dream  Sleep  Span  Gest
  1   2  NonD  Dream  Sleep  Span  Gest
  1   3  NonD  Dream  Sleep  Span  Gest
  1   4  NonD  Dream  Sleep  Span  Gest
  1   5  NonD  Dream  Sleep  Span  Gest
  2   1  NonD  Dream  Sleep  Span  Gest
  2   2  NonD  Dream  Sleep  Span  Gest
  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
fit <- with(imp, lm(Dream ~ Span + Gest))
pooled <- pool(fit)

# Displaying the results
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
```

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



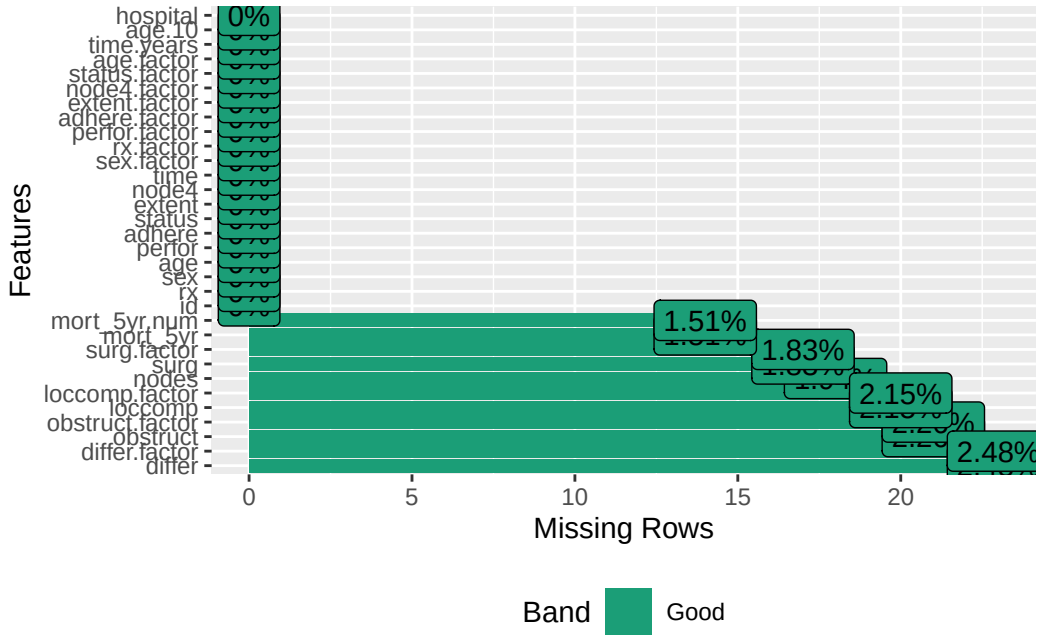
4 Case Study

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

```
# Load and examine the colon_s dataset
dim(colon_s) # Display dimensions of the dataset
```

```
[1] 929  32
```

```
DataExplorer::plot_missing(colon_s) # Visualize missing data in the dataset
```




```
str(colon_s) # Display the structure of the dataset
```

```
'data.frame':   929 obs. of  32 variables:
 $ id             : num  1 2 3 4 5 6 7 8 9 10 ...
 $ rx             : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 1 3 1 3 2 1 2 3 ...
 $ sex            : num  1 1 0 0 1 0 1 1 1 0 ...
 $ age            : num  43 63 71 66 69 57 77 54 46 68 ...
  ..- attr(*, "label")= chr "Age (years)"
 $ obstruct       : num  NA 0 0 1 0 0 0 0 0 0 ...
 $ perfor         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ adhere         : num  0 0 1 0 0 0 0 0 1 0 ...
 $ nodes          : num  5 1 7 6 22 9 5 1 2 1 ...
 $ status         : num  1 0 1 1 1 1 1 0 0 0 ...
 $ differ         : num  2 2 2 2 2 2 2 2 2 2 ...
 $ extent         : num  3 3 2 3 3 3 3 3 3 3 ...
 $ surg           : num  0 0 0 1 1 0 1 0 0 1 ...
 $ node4          : num  1 0 1 1 1 1 1 0 0 0 ...
 $ time           : num  1521 3087 963 293 659 ...
 $ sex.factor     : Factor w/ 2 levels "Female","Male": 2 2 1 1 2 1 2 2 2 1 ...
  ..- attr(*, "label")= chr "Sex"
 $ rx.factor      : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 1 3 1 3 2 1 2 3 ...
```

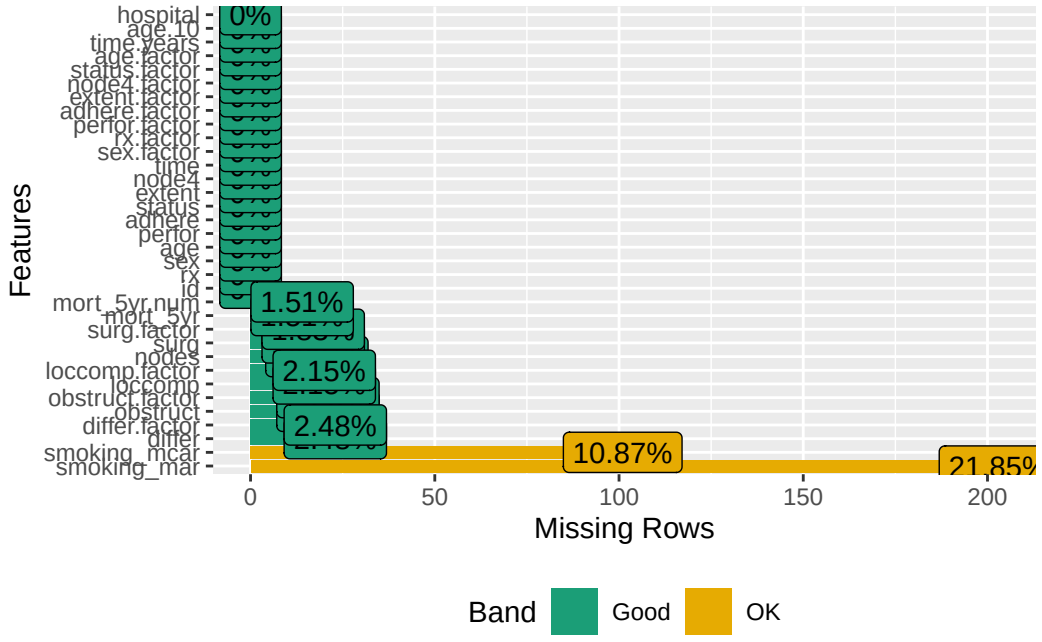
```
  ..- attr(*, "label")= chr "Treatment"
 $ obstruct.factor: Factor w/ 2 levels "No","Yes": NA 1 1 2 1 1 1 1 1 1 ...
  ..- attr(*, "label")= chr "Obstruction"
 $ perfor.factor  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "label")= chr "Perforation"
 $ adhere.factor  : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 2 1 ...
  ..- attr(*, "label")= chr "Adherence"
 $ differ.factor  : Factor w/ 3 levels "Well","Moderate",..: 2 2 2 2 2 2 2 2 2 2 ...
  ..- attr(*, "label")= chr "Differentiation"
 $ extent.factor  : Factor w/ 4 levels "Submucosa","Muscle",..: 3 3 2 3 3 3 3 3 3 3 ..
  ..- attr(*, "label")= chr "Extent of spread"
 $ surg.factor    : Factor w/ 2 levels "Short","Long": 1 1 1 2 2 1 2 1 1 2 ...
  ..- attr(*, "label")= chr "Time from surgery"
 $ node4.factor   : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 1 1 1 ...
  ..- attr(*, "label")= chr ">4 positive nodes"
 $ status.factor  : Factor w/ 2 levels "Alive","Died": 2 1 2 2 2 2 2 1 1 1 ...
  ..- attr(*, "label")= chr "Status"
 $ age.factor     : Factor w/ 3 levels "<40 years","40-59 years",..: 2 3 3 3 3 2 3 2 2
  ..- attr(*, "label")= chr "Age"
 $ loccomp        : num  NA 0 1 1 0 0 0 0 1 0 ...
 $ loccomp.factor : Factor w/ 2 levels "No","Yes": NA 1 2 2 1 1 1 1 2 1 ...
  ..- attr(*, "label")= chr "Local complications"
 $ time.years     : num  4.167 8.458 2.638 0.803 1.805 ...
  ..- attr(*, "label")= chr "Time (years)"
 $ mort_5yr       : Factor w/ 2 levels "Alive","Died": 2 1 2 2 2 2 2 1 1 1 ...
  ..- attr(*, "label")= chr "Mortality 5 year"
 $ age.10         : num  4.3 6.3 7.1 6.6 6.9 5.7 7.7 5.4 4.6 6.8 ...
 $ mort_5yr.num   : num  2 1 2 2 2 2 2 1 1 1 ...
 $ hospital       : Factor w/ 5 levels "hospital_1","hospital_2",..: 5 3 5 4 5 4 2 2 2
```

```r
# Set seed for reproducibility
set.seed(1)

# Manipulate the dataset to introduce additional missing data
colon_s <- colon_s %>%
  mutate(
    # Introducing missing data for 'smoking' 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)"),
```

```
    # Introducing missing data for 'smoking' 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)")
  )

# Display the updated dimensions of the dataset to check for changes
dim(colon_s) # Chemotherapy for Stage B/C colon cancer
```

```
[1] 929  34
```

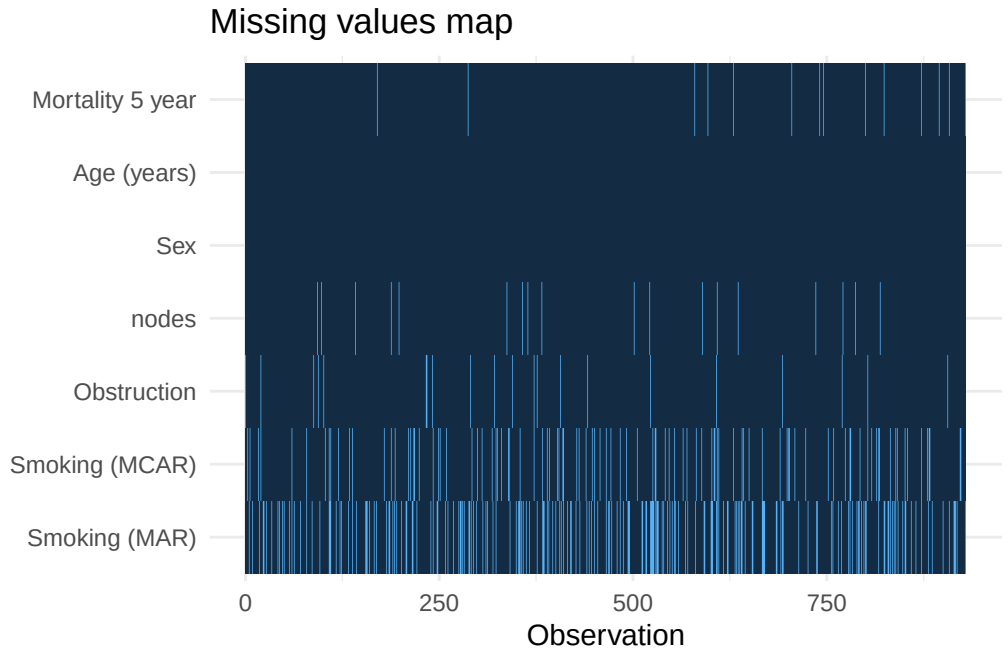
```
# Re-visualize missing data in the modified dataset
DataExplorer::plot_missing(colon_s)
```



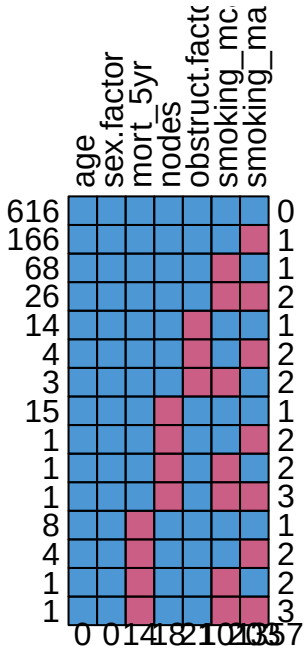
```r
# Use the ff_glimpse() function to examine the variables of interest.
explanatory <- c("age", "sex.factor", "nodes", "obstruct.factor",
"smoking_mcar", "smoking_mar")
dependent <- "mort_5yr"

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
```

```r
# Missing values map
colon_s %>%
finalfit::missing_plot(dependent, explanatory)
```




```
# Look for ptterns of missingness
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

```
# Including missing data in demographics tables
table1 <- colon_s %>%
  summary_factorlist(
    dependent = dependent,
    explanatory = 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)
)
```

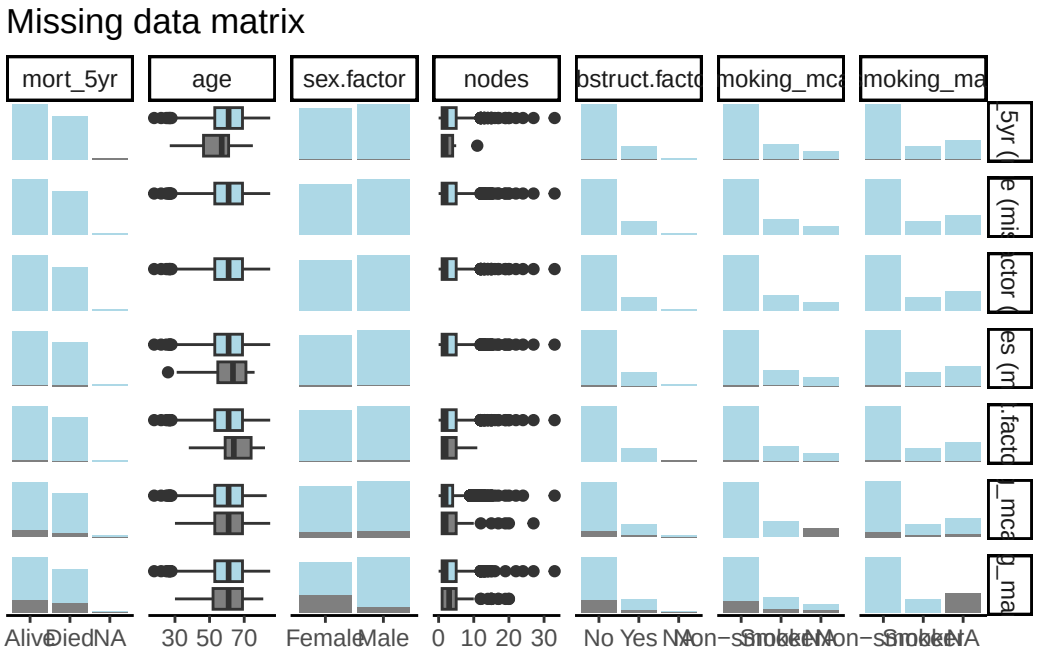
Table 1: Simulated missing completely at random (MCAR) and missing at random (MAR) dataset.

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)	
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)	

```
# check for associations bewtwwen missing and observed data
dependent <- "mort_5yr"
explanatory <- c("age", "sex.factor", "nodes", "obstruct.factor", "smoking_mcar", "smo
colon_s %>%
  missing_pairs(
    dependent = dependent,
    explanatory = explanatory,
    title = "Missing data matrix",
    use_labels = FALSE,
    showYAxisPlotLabels = FALSE
  )
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```
Warning: Removed 18 rows containing non-finite values (`stat_boxplot()`).
Removed 18 rows containing non-finite values (`stat_boxplot()`).
Removed 18 rows containing non-finite values (`stat_boxplot()`).
Removed 18 rows containing non-finite values (`stat_boxplot()`).
Removed 18 rows containing non-finite values (`stat_boxplot()`).
Removed 18 rows containing non-finite values (`stat_boxplot()`).
Removed 18 rows containing non-finite values (`stat_boxplot()`).
```




```r
# MCAR
dependent <- "smoking_mcar"
explanatory <- c("age", "sex.factor", "nodes", "obstruct.factor")

missing_mcar <- colon_s %>%
  missing_compare(
    dependent = dependent,
    explanatory = 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)	

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

missing_mar <- colon_s %>%
  missing_compare(
    dependent = dependent,
    explanatory = explanatory
  )

knitr::kable(missing_mar)
```

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
Obstruction	No	568 (77.6)	164 (22.4)	0.533
	Yes	141 (80.1)	35 (19.9)	

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

```
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)

```r
dependent <- "mort_5yr"
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...
```

```r
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)
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)

```r
# Multivariate Imputation
dependent <- "mort_5yr"
explanatory <- c("age", "sex.factor", "nodes", "obstruct.factor", "smoking_mar")
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>.
```

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

```
 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
```

```
# AICs
fits %>%
  getfit() %>%
  purrr::map(AIC) %>%
  unlist() %>%
  mean()
```

```
[1] 1193.216
```

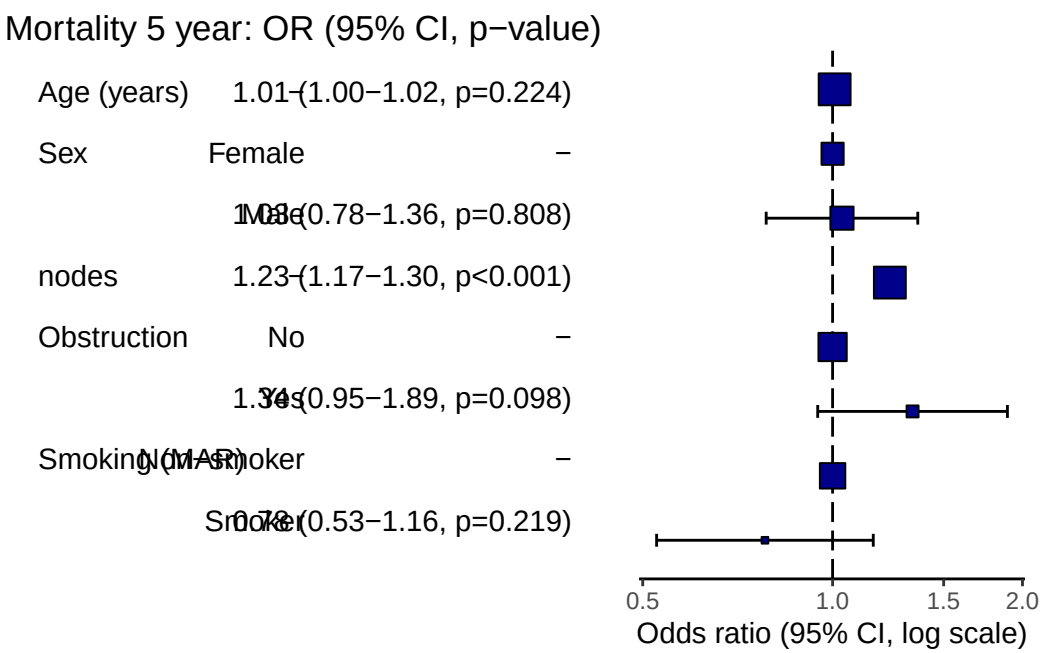
```
# Pool results
fits_pool <- fits %>%
  pool()
knitr::kable(fits_pool$pooled)
```

term	m	estimate	ubar	b	t	dfcom	df	riv	lambda	fmi
(Intercept)	10	-1.4501982	0.1566905	0.0095674	0.1672147	723	625.4584	0.0671653	0.0629380	0.0659201
age	10	0.0073772	0.0000355	0.0000012	0.0000368	723	789.0676	0.0369913	0.0356717	0.0381067
sex.factorMale	10	0.0342079	0.0194230	0.0004106	0.0198746	723	855.8758	0.0232530	0.0227246	0.0250003
nodes	10	0.2089765	0.0006534	0.0000270	0.0006833	723	743.4049	0.0454546	0.0434783	0.0460413
obstruct.factorYes	10	0.2919575	0.0304573	0.0005987	0.0311159	723	862.7930	0.0216240	0.0211663	0.0234274
smoking_marSmoker	10	-0.2469609	0.0320638	0.0073386	0.0401362	723	170.8304	0.2517621	0.2011261	0.2103176

```
colon_s %>%
  or_plot(dependent, explanatory, glmfit = fits_pool, table_text_size=4)
```

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

```
Warning: Removed 3 rows containing missing values (`geom_errorbarh()`).
```




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

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...
```

```r
# 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)
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)

5 Summary

The document "Introduction to Missing Data" provides an extensive overview of techniques to address missing data in statistical analysis. It explains the nature of missing

data and its impact on analysis, explores conventional methods like case deletion and weighting, and discusses single and multiple imputation techniques. Single imputation methods, which replace missing values with a single estimate, are contrasted with multiple imputation approaches that create several datasets for robust statistical inference. The document also includes practical applications through a case study, demonstrating how these methods can be implemented in real-world data analysis to manage the challenges posed by missing data effectively.

6 References

1. ChatGPT4