1.3.4: Missing Data and Sampling Weights (brief intro)

(In Person)

COMING SUMMER 2025

Module “1.3.4: Missing Data and Sampling Weights” will be posted prior to the In-Person Workshops in Summer 2025.

Session Objectives

1. Identify and summarize missing data.
2. Learn methods to handle missing data according to variable type.
3. Use a survey sampling weight to generate more representative descriptive and inferential statistical values (brief intro)
4. Discuss potential bias when removing missing observations without careful examination.

[to be removed…………]

Key points:

1. R packages that support missing data examination
2. Mean/median imputation for continuous variables
3. What to do with missing observations for categorical variables
4. Ways to examine potential differences between complete and missing observations in association between certain independent and dependent variables
   - What to do if such association significantly differs between complete and missing observations
5. R packages for complex survey data (e.g., survey package)
   - R codes to generate weighted descriptive statistics and contingency tables, as well as to develop weighted linear models

---

0. Prework - Before You Begin

Install Packages

Before you begin, please go ahead and install the following packages - these are all on CRAN, so you can install them using the RStudio Menu "Tools/Install" Packages interface:

- `VIM` on CRAN and `VIM` package website
- `skimr` on CRAN and `skimr` website - OPTIONAL
- `modelsummary` on CRAN and `modelsummary` website - OPTIONAL
- `tinytable` on CRAN and `tinytable` website - OPTIONAL
- `summarytools` on CRAN and `summarytools` on Github - OPTIONAL
- mice
- mi
-
-
- `palmerpenguins` on CRAN

See Module 1.3.1 on Installing Packages

See additional resources below…

add to prework?

Begin with a NEW RStudio Project

Let's begin with a new RStudio Project.

---

1. Identify and summarize missing data.

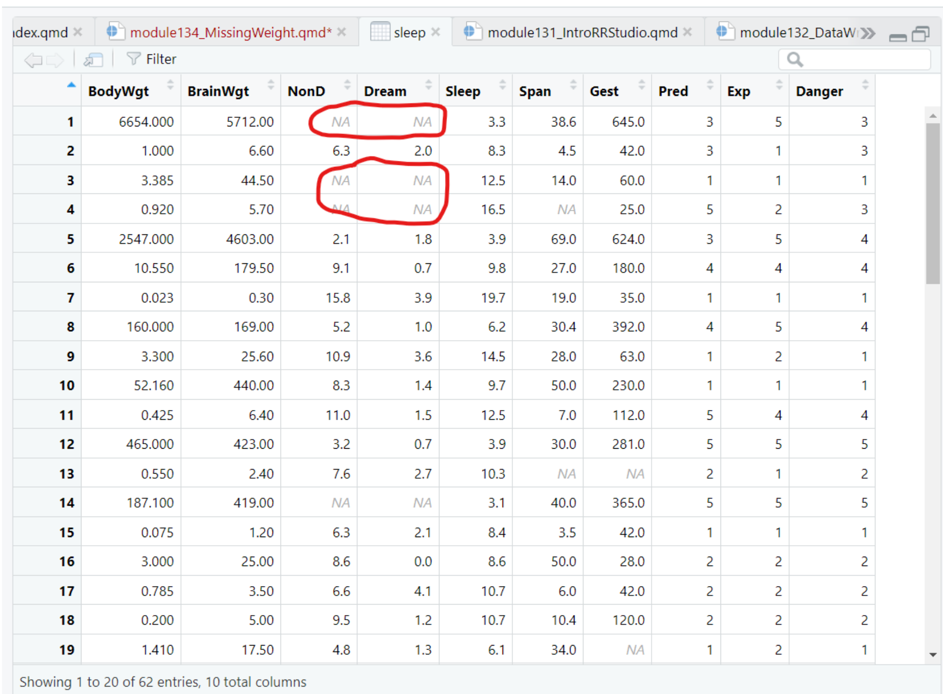
Find Missing Data in Your Dataset.

One simple way to find missing data is to open it in the Data Viewer window and sort the data.

For example, load the `VIM` package and take a look at the `sleep` dataset provided within this package.

```
library(VIM)
data("sleep")
```

Click on the sleep dataset to open it in the data viewer:



	BodyWgt	BrainWgt	NonD	Dream	Sleep	Span	Gest	Pred	Exp	Danger
1	6654.000	5712.00	NA	NA	3.3	38.6	645.0	3	5	3
2	1.000	6.60	6.3	2.0	8.3	4.5	42.0	3	1	3
3	3.385	44.50	NA	NA	12.5	14.0	60.0	1	1	1
4	0.920	5.70	NA	NA	16.5	NA	25.0	5	2	3
5	2547.000	4603.00	2.1	1.8	3.9	69.0	624.0	3	5	4
6	10.550	179.50	9.1	0.7	9.8	27.0	180.0	4	4	4
7	0.023	0.30	15.8	3.9	19.7	19.0	35.0	1	1	1
8	160.000	169.00	5.2	1.0	6.2	30.4	392.0	4	5	4
9	3.300	25.60	10.9	3.6	14.5	28.0	63.0	1	2	1
10	52.160	440.00	8.3	1.4	9.7	50.0	230.0	1	1	1
11	0.425	6.40	11.0	1.5	12.5	7.0	112.0	5	4	4
12	465.000	423.00	3.2	0.7	3.9	30.0	281.0	5	5	5
13	0.550	2.40	7.6	2.7	10.3	NA	NA	2	1	2
14	187.100	419.00	NA	NA	3.1	40.0	365.0	5	5	5
15	0.075	1.20	6.3	2.1	8.4	3.5	42.0	1	1	1
16	3.000	25.00	8.6	0.0	8.6	50.0	28.0	2	2	2
17	0.785	3.50	6.6	4.1	10.7	6.0	42.0	2	2	2
18	0.200	5.00	9.5	1.2	10.7	10.4	120.0	2	2	2
19	1.410	17.50	4.8	1.3	6.1	34.0	NA	1	2	1

Notice the light grey `NA`s shown for the missing data spots in this dataset.

If we click on the column for the `Dream` variable and sort these values, notice that the `NA`s all now show up at the bottom of the viewer window. It does not matter if you sort ascending or descending, the `NA`s are always at the bottom of the viewer.

	BodyWgt	BrainWgt	NonD	Dream	Sleep	Span	Gest	Pred	Exp	Danger
27	0.101	4.00	10.4	3.4	13.8	9.0	28.0	5	1	3
9	3.300	25.60	10.9	3.6	14.5	28.0	63.0	1	2	1
7	0.023	0.30	15.8	3.9	19.7	19.0	35.0	1	1	1
17	0.785	3.50	6.6	4.1	10.7	6.0	42.0	2	2	2
39	1.700	6.30	13.8	5.6	19.4	5.0	12.0	2	1	1
20	60.000	81.00	12.0	6.1	18.1	7.0	NA	1	1	1
61	3.500	3.90	12.8	6.6	19.4	3.0	14.0	2	1	1
1	6654.000	5712.00	NA	NA	3.3	38.6	645.0	3	5	3
3	3.385	44.50	NA	NA	12.5	14.0	60.0	1	1	1
4	0.920	5.70	NA	NA	16.5	NA	25.0	5	2	3
14	187.100	419.00	NA	NA	3.1	40.0	365.0	5	5	5
24	207.000	406.00	NA	NA	12.0	39.3	252.0	1	4	1
26	36.330	119.50	NA	NA	13.0	16.2	63.0	1	1	1
30	100.000	157.00	NA	NA	10.8	22.4	100.0	1	1	1
31	35.000	56.00	NA	NA	NA	16.3	33.0	3	5	4
47	4.288	39.20	NA	NA	12.5	13.7	63.0	2	2	2
53	14.830	98.20	NA	NA	2.6	17.0	150.0	5	5	5
55	1.400	12.50	NA	NA	11.0	12.7	90.0	2	2	2
62	4.050	17.00	NA	NA	NA	13.0	38.0	3	1	1

Showing 44 to 62 of 62 entries, 10 total columns

This method is ok for a small dataset with not too many variables or rows of data. But let's look at other ways to summarize the amounts of missing data in your dataset.

Describe Missing Data.

Built-in `summary()` function

As we saw back in Module 1.3.2, Section 5, we can use the `summary()` function to get some basic statistics for each variable in the dataset, including the number of `NA`s.

```r
summary(sleep)
```

```
    BodyWgt            BrainWgt            NonD             Dream      
 Min.   :   0.005   Min.   :   0.14   Min.   : 2.100   Min.   :0.000  
 1st Qu.:   0.600   1st Qu.:   4.25   1st Qu.: 6.250   1st Qu.:0.900  
 Median :   3.342   Median :  17.25   Median : 8.350   Median :1.800  
 Mean   : 198.790   Mean   : 283.13   Mean   : 8.673   Mean   :1.972  
 3rd Qu.:  48.203   3rd Qu.: 166.00   3rd Qu.:11.000   3rd Qu.:2.550  
 Max.   :6654.000   Max.   :5712.00   Max.   :17.900   Max.   :6.600  
                                      NA's   :14       NA's   :12     
     Sleep            Span              Gest             Pred      
 Min.   : 2.60   Min.   :  2.000   Min.   : 12.00   Min.   :1.000  
```

```
 1st Qu.: 8.05   1st Qu.:  6.625   1st Qu.: 35.75   1st Qu.:2.000
 Median :10.45   Median : 15.100   Median : 79.00   Median :3.000
 Mean   :10.53   Mean   : 19.878   Mean   :142.35   Mean   :2.871
 3rd Qu.:13.20   3rd Qu.: 27.750   3rd Qu.:207.50   3rd Qu.:4.000
 Max.   :19.90   Max.   :100.000   Max.   :645.00   Max.   :5.000
 NA's   :4       NA's   :4         NA's   :4
      Exp            Danger
 Min.   :1.000   Min.   :1.000
 1st Qu.:1.000   1st Qu.:1.000
 Median :2.000   Median :2.000
 Mean   :2.419   Mean   :2.613
 3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :5.000   Max.   :5.000
```

skimr package

Another helpful package is the `skimr` package which has the `skim()` function which provides a count of the amount of missing data and the proportion of complete data for that variable.

> **Rmarkdown for `skimr` package**
>
> When "knitting" to HTML the code below creates the summary table with the miniture histograms. However, when "knitting" to PDF (using the default portrait layout)m the histograms get cutoff on the page. Additional LaTex customization is needed to change the layout to landscape to be able to see the histograms.

```r
library(skimr)
skim(sleep)
```

Table 1: Data summary

Name	sleep
Number of rows	62
Number of columns	10
Column type frequency:	
numeric	10
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
BodyWgt	0	1.00	198.79	899.16	0.00	0.60	3.34	48.20	6654.0	
BrainWgt	0	1.00	283.13	930.28	0.14	4.25	17.25	166.00	5712.0	
NonD	14	0.77	8.67	3.67	2.10	6.25	8.35	11.00	17.9	
Dream	12	0.81	1.97	1.44	0.00	0.90	1.80	2.55	6.6	
Sleep	4	0.94	10.53	4.61	2.60	8.05	10.45	13.20	19.9	
Span	4	0.94	19.88	18.21	2.00	6.62	15.10	27.75	100.0	
Gest	4	0.94	142.35	146.81	12.00	35.75	79.00	207.50	645.0	
Pred	0	1.00	2.87	1.48	1.00	2.00	3.00	4.00	5.0	
Exp	0	1.00	2.42	1.60	1.00	1.00	2.00	4.00	5.0	
Danger	0	1.00	2.61	1.44	1.00	1.00	2.00	4.00	5.0	

`modelsummary` package

Another helpful package is the `modelsummary` package which has the `datasummary_skim()` function which is a slightly better version built off the `skimr::skim()` package and function.

> **i** Rmarkdown for `modelsummary` package
>
> The `tinytable` package is also used below with the `modelsummary` output to better control the placement of the resulting table when "knitting" to PDF.

```
library(modelsummary)
library(tinytable)
datasummary_skim(sleep) %>%
  theme_tt("placement", latex_float = "H")
```

	Unique	Missing Pct.	Mean	SD	Min	Median	Max	Histogram
BodyWgt	60	0	198.8	899.2	0.0	3.3	6654.0	
BrainWgt	59	0	283.1	930.3	0.1	17.2	5712.0	
NonD	40	23	8.7	3.7	2.1	8.4	17.9	
Dream	31	19	2.0	1.4	0.0	1.8	6.6	
Sleep	45	6	10.5	4.6	2.6	10.4	19.9	
Span	48	6	19.9	18.2	2.0	15.1	100.0	
Gest	50	6	142.4	146.8	12.0	79.0	645.0	
Pred	5	0	2.9	1.5	1.0	3.0	5.0	
Exp	5	0	2.4	1.6	1.0	2.0	5.0	
Danger	5	0	2.6	1.4	1.0	2.0	5.0	

`summarytools` package

Another package that also provides a nice summary of the variables in the dataset, is the `dfSummary()` from the `summarytools` dataset.

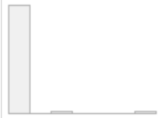
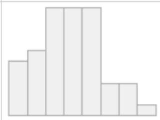
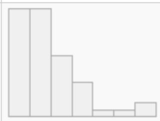
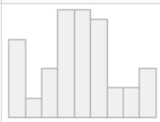
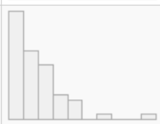
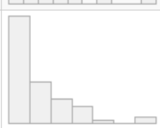
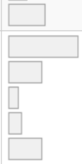
NOTE: Learn more about how to use `summarytools::dfSummary()` in an Rmarkdown document at https://cran.r-project.org/web/packages/summarytools/vignettes/rmarkdown.html.

```
library(summarytools)
view(dfSummary(sleep))
```

Data Frame Summary

sleep

Dimensions: 62 x 10
Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	BodyWgt [numeric]	Mean (sd) : 198.8 (899.2) min ≤ med ≤ max: 0 ≤ 3.3 ≤ 6654 IQR (CV) : 47.6 (4.5)	60 distinct values		62 (100.0%)	0 (0.0%)
2	BrainWgt [numeric]	Mean (sd) : 283.1 (930.3) min ≤ med ≤ max: 0.1 ≤ 17.2 ≤ 5712 IQR (CV) : 161.8 (3.3)	59 distinct values		62 (100.0%)	0 (0.0%)
3	NonD [numeric]	Mean (sd) : 8.7 (3.7) min ≤ med ≤ max: 2.1 ≤ 8.4 ≤ 17.9 IQR (CV) : 4.8 (0.4)	39 distinct values		48 (77.4%)	14 (22.6%)
4	Dream [numeric]	Mean (sd) : 2 (1.4) min ≤ med ≤ max: 0 ≤ 1.8 ≤ 6.6 IQR (CV) : 1.7 (0.7)	30 distinct values		50 (80.6%)	12 (19.4%)
5	Sleep [numeric]	Mean (sd) : 10.5 (4.6) min ≤ med ≤ max: 2.6 ≤ 10.4 ≤ 19.9 IQR (CV) : 5.1 (0.4)	44 distinct values		58 (93.5%)	4 (6.5%)
6	Span [numeric]	Mean (sd) : 19.9 (18.2) min ≤ med ≤ max: 2 ≤ 15.1 ≤ 100 IQR (CV) : 21.1 (0.9)	47 distinct values		58 (93.5%)	4 (6.5%)
7	Gest [numeric]	Mean (sd) : 142.4 (146.8) min ≤ med ≤ max: 12 ≤ 79 ≤ 645 IQR (CV) : 171.8 (1)	49 distinct values		58 (93.5%)	4 (6.5%)
8	Pred [integer]	Mean (sd) : 2.9 (1.5) min ≤ med ≤ max: 1 ≤ 3 ≤ 5 IQR (CV) : 2 (0.5)	1 : 14 (22.6%) 2 : 15 (24.2%) 3 : 12 (19.4%) 4 : 7 (11.3%) 5 : 14 (22.6%)		62 (100.0%)	0 (0.0%)
9	Exp [integer]	Mean (sd) : 2.4 (1.6) min ≤ med ≤ max: 1 ≤ 2 ≤ 5 IQR (CV) : 3 (0.7)	1 : 27 (43.5%) 2 : 13 (21.0%) 3 : 4 (6.5%) 4 : 5 (8.1%) 5 : 13 (21.0%)		62 (100.0%)	0 (0.0%)
10	Danger [integer]	Mean (sd) : 2.6 (1.4) min ≤ med ≤ max: 1 ≤ 2 ≤ 5 IQR (CV) : 3 (0.6)	1 : 19 (30.6%) 2 : 14 (22.6%) 3 : 10 (16.1%) 4 : 10 (16.1%) 5 : 9 (14.5%)		62 (100.0%)	0 (0.0%)

Generated by summarytools 1.1.4 (R version 4.5.1)
2025-06-20

Try It On Your Own

Try running `summary()` or `skim()` on the `penguins` dataset from the `palmerpenguins` package. Notice the summaries for the numeric and the factor type variables.

```
library(palmerpenguins)
summary(penguins)
```

```
      species          island    bill_length_mm  bill_depth_mm  
 Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10  
 Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60  
 Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30  
                                 Mean   :43.92   Mean   :17.15  
                                 3rd Qu.:48.50   3rd Qu.:18.70  
                                 Max.   :59.60   Max.   :21.50  
                                 NA's   :2       NA's   :2      
 flipper_length_mm  body_mass_g       sex           year     
 Min.   :172.0     Min.   :2700   female:165   Min.   :2007  
 1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007  
 Median :197.0     Median :4050   NA's  : 11   Median :2008  
 Mean   :200.9     Mean   :4202                Mean   :2008  
 3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009  
 Max.   :231.0     Max.   :6300                Max.   :2009  
 NA's   :2         NA's   :2                                 
```

```
skim(penguins)
```

Table 3: Data summary

Name	penguins
Number of rows	344
Number of columns	8
Column type frequency:	
factor	3
numeric	5
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
species	0	1.00	FALSE	3	Ade: 152, Gen: 124, Chi: 68
island	0	1.00	FALSE	3	Bis: 168, Dre: 124, Tor: 52
sex	11	0.97	FALSE	2	mal: 168, fem: 165

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
bill_length_mm	2	0.99	43.92	5.46	32.1	39.23	44.45	48.5	59.6	
bill_depth_mm	2	0.99	17.15	1.97	13.1	15.60	17.30	18.7	21.5	
flipper_length_mm	2	0.99	200.92	14.06	172.0	190.00	197.00	213.0	231.0	
body_mass_g	2	0.99	4201.75	801.95	2700.0	3550.00	4050.00	4750.0	6300.0	
year	0	1.00	2008.03	0.82	2007.0	2007.00	2008.00	2009.0	2009.0	

Visualize Missing Data.

Making plots with `VIM` package

The `VIM` package has an "aggregate" function `aggr()` which counts up the amounts of missing data for each variable and combinations of variables. The `sleep` dataset only has 10 variables.

> ⚠ WARNING - Too Many Variables
>
> Before using the `aggr()` function, limit the number of variables. FIRST create a dataset with only the variables you are interested in BEFORE running the function - otherwise you may lock up your computer if you feed it too many variables at once.

```
# get a quick count of the amount of missing
# data in the sleep dataset for each variable
a <- aggr(sleep, plot = FALSE)
a
```

```
 Missings in variables:
 Variable Count
     NonD    14
    Dream    12
```

```
 Sleep     4
  Span     4
  Gest     4
```

The default output from above only lists the variables that have one or more rows with missing data. However, you can get a complete list with:

```
a$missings
```

```
         Variable Count
BodyWgt   BodyWgt     0
BrainWgt BrainWgt     0
NonD         NonD    14
Dream       Dream    12
Sleep       Sleep     4
Span         Span     4
Gest         Gest     4
Pred         Pred     0
Exp           Exp     0
Danger     Danger     0
```

Next let's get some plots.

The plot on the LEFT below is a simple bar plot showing the missing counts for each variable in the dataset.

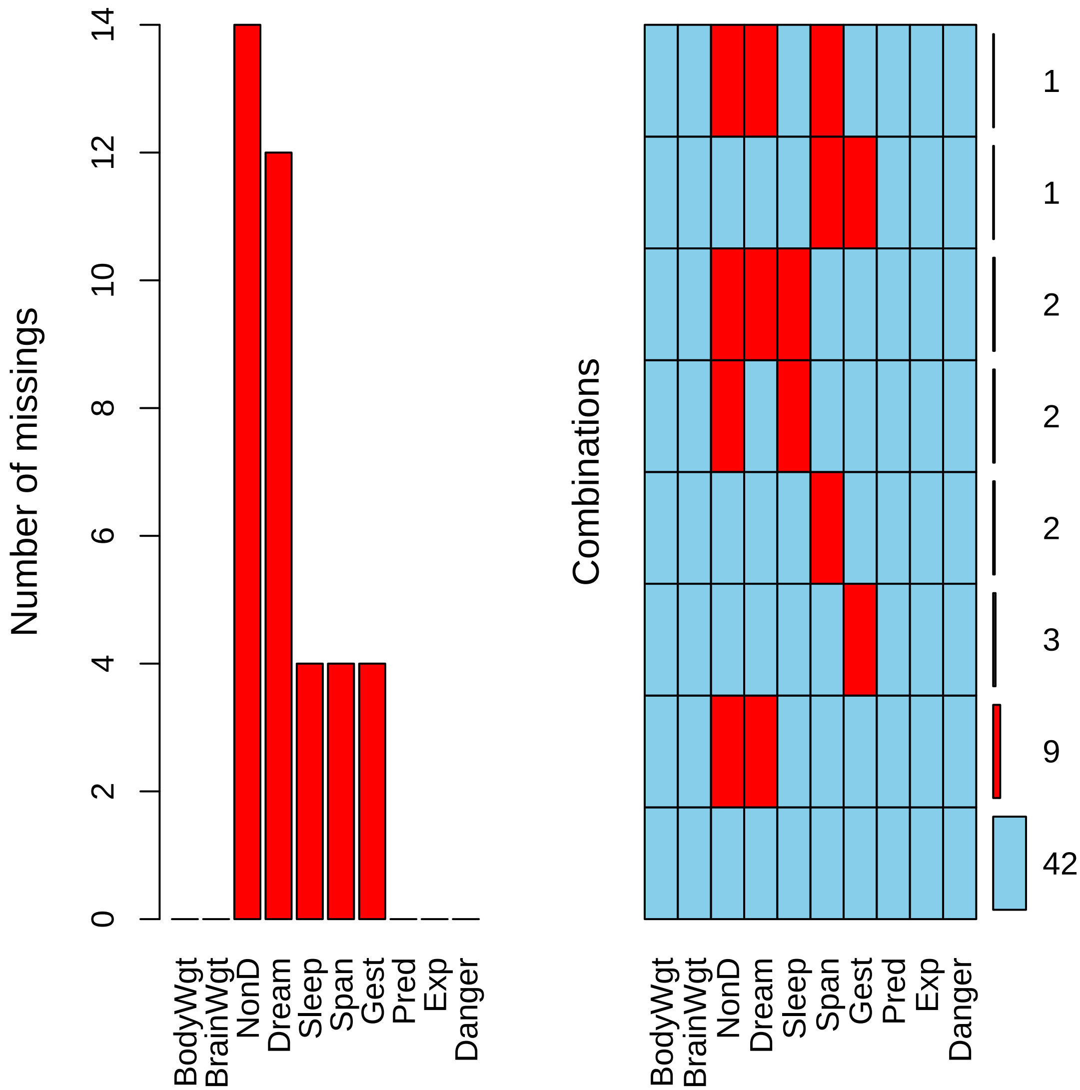
- Also notice that there are only 5 variables with one or more missing values:
  - `NonD`
  - `Dream`
  - `Sleep`
  - `Span`
  - `Gest`

The plot on the RIGHT however, shows the amounts of missing data for the various patterns of missing data for the 10 variables in the `sleep` dataset.

For example, notice that of the 62 rows of data in the `sleep` dataset:

- there are only 42 rows with complete data with no missing data on all 10 variables;
- the next largest "pattern" of missing data is 9 rows that are missing both `NonD` and `Dream` variables.; and
- there are 3 rows of data with the `gest` variable with missing data.

```r
# make plots of the amounts and patterns of missing data
plot(a, numbers = TRUE, prop = FALSE)
```

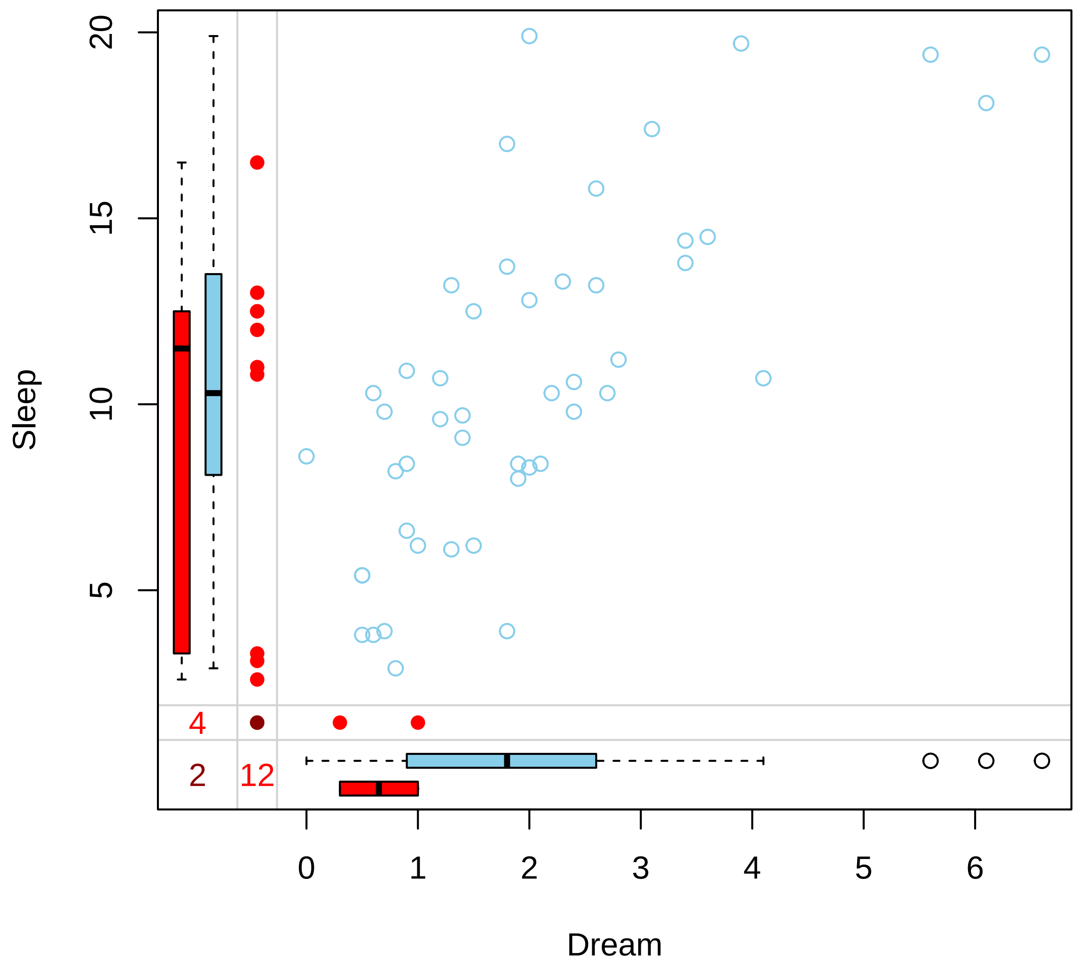


Marginplots - see how missingness varys with other measures

In addition to a usual scatterplot, the `marginplot()` function in the `VIM` package, also shows information about missing values in the plot margins.

The red boxplot on the left shows the distrubution of all values of `Sleep` where `Dream` contains a missing value. The blue boxplot on the left shows the distribution of the values of `Sleep` where `Dream` is observed.

```r
x <- sleep[, c("Dream", "Sleep")]
marginplot(x)
```



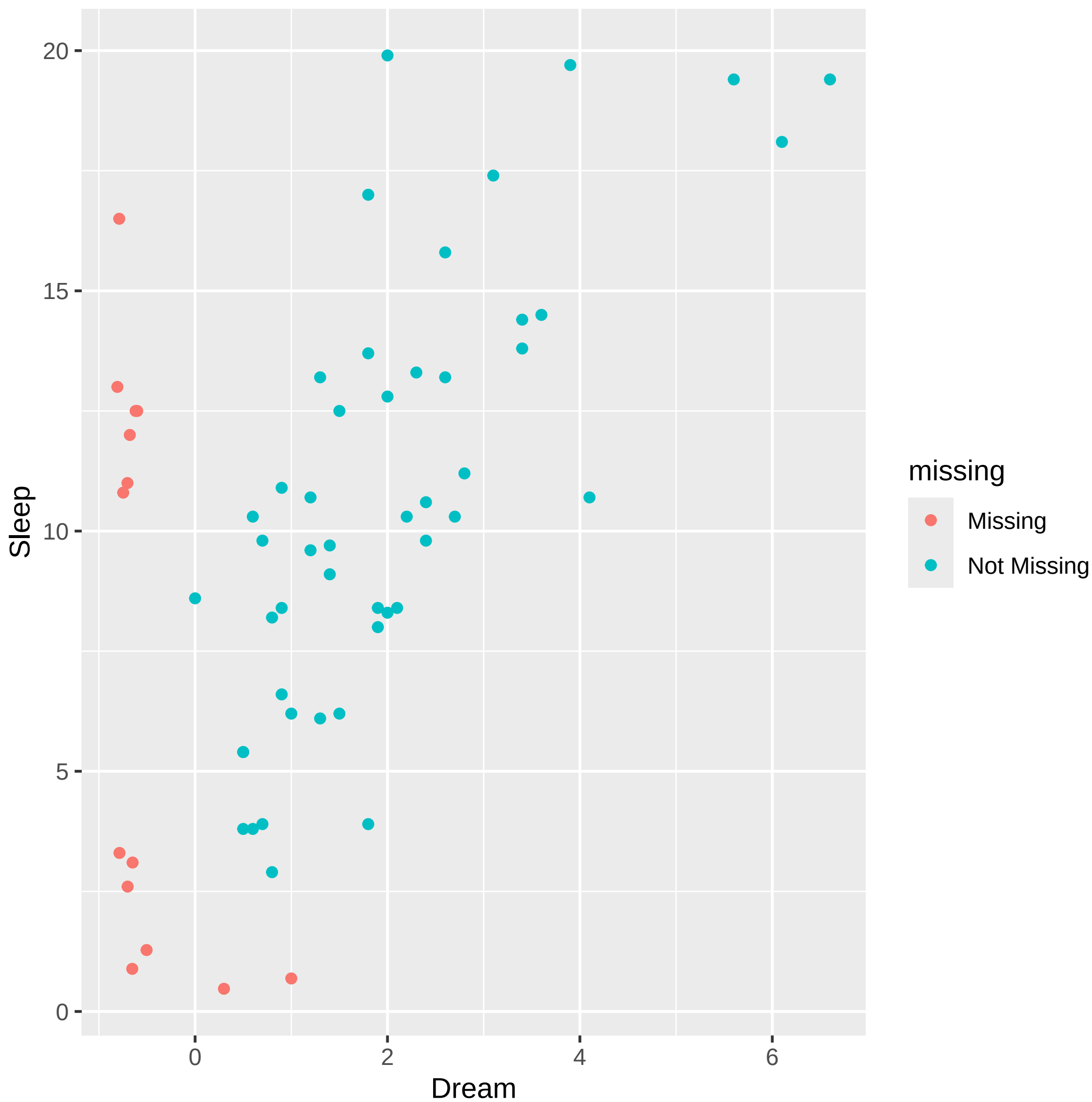
Visualize Missing Data with the `naniar` package

The naniar package "provides principled, tidy ways to summarise, visualise, and manipulate missing data with minimal deviations from the workflows in ggplot2 and tidy data." *See `naniar` website.*

For example, let's make a similar to plot to what we did above to visualize the scatterplot between `Dream` and `Sleep` but also consider the amounts of missing data of one variable relative to the other variable in the plot. We can do this using the `geom_miss_point()` function provided in the `naniar` package which works with `ggplot2`.

```
library(naniar)
library(ggplot2)

ggplot(sleep,
       aes(x = Dream,
           y = Sleep)) +
  geom_miss_point()
```



2. Learn methods to handle missing data according to variable type.

discuss pairwise versus listwise and discuss impacts on modeling especially for stepwise variable selection - always check the final N for each model

show correlations pairwise and listwise

add details on modeling adjustments - covariate predicted missingness

options on imputation - brief intro

```r
sleep$NonD_missing <-
  as.numeric(is.na(sleep$NonD))

correlation::correlation(sleep)
```

```
# Correlation Matrix (pearson-method)

Parameter1 |   Parameter2 |     r |         95% CI |     t | df |         p
----------------------------------------------------------------------------
BodyWgt    |     BrainWgt |  0.93 | [ 0.89,  0.96] | 20.28 | 60 | < .001***
BodyWgt    |         NonD | -0.38 | [-0.60, -0.10] | -2.75 | 46 | 0.211
BodyWgt    |        Dream | -0.11 | [-0.38,  0.17] | -0.76 | 48 | > .999
BodyWgt    |        Sleep | -0.31 | [-0.52, -0.05] | -2.42 | 56 | 0.418
BodyWgt    |         Span |  0.30 | [ 0.05,  0.52] |  2.37 | 56 | 0.441
BodyWgt    |         Gest |  0.65 | [ 0.47,  0.78] |  6.42 | 56 | < .001***
BodyWgt    |         Pred |  0.06 | [-0.19,  0.30] |  0.46 | 60 | > .999
BodyWgt    |          Exp |  0.34 | [ 0.10,  0.54] |  2.78 | 60 | 0.186
BodyWgt    |       Danger |  0.13 | [-0.12,  0.37] |  1.04 | 60 | > .999
BodyWgt    | NonD_missing |  0.23 | [-0.02,  0.45] |  1.80 | 60 | > .999
BrainWgt   |         NonD | -0.37 | [-0.59, -0.10] | -2.69 | 46 | 0.226
BrainWgt   |        Dream | -0.11 | [-0.37,  0.18] | -0.73 | 48 | > .999
BrainWgt   |        Sleep | -0.36 | [-0.56, -0.11] | -2.87 | 56 | 0.156
BrainWgt   |         Span |  0.51 | [ 0.29,  0.68] |  4.43 | 56 | 0.002**
BrainWgt   |         Gest |  0.75 | [ 0.61,  0.84] |  8.41 | 56 | < .001***
BrainWgt   |         Pred |  0.03 | [-0.22,  0.28] |  0.26 | 60 | > .999
BrainWgt   |          Exp |  0.37 | [ 0.13,  0.57] |  3.06 | 60 | 0.098
BrainWgt   |       Danger |  0.15 | [-0.11,  0.38] |  1.14 | 60 | > .999
BrainWgt   | NonD_missing |  0.18 | [-0.07,  0.41] |  1.41 | 60 | > .999
NonD       |        Dream |  0.51 | [ 0.27,  0.70] |  4.07 | 46 | 0.007**
NonD       |        Sleep |  0.96 | [ 0.93,  0.98] | 24.14 | 46 | < .001***
NonD       |         Span | -0.38 | [-0.61, -0.10] | -2.73 | 43 | 0.219
NonD       |         Gest | -0.59 | [-0.76, -0.36] | -4.79 | 42 | < .001***
```

```
NonD        |        Pred | -0.32 | [-0.55, -0.04] | -2.28 | 46 | 0.551
NonD        |         Exp | -0.54 | [-0.72, -0.31] | -4.39 | 46 | 0.002**
NonD        |      Danger | -0.48 | [-0.68, -0.23] | -3.75 | 46 | 0.017*
NonD        | NonD_missing |      |                |       | 46 |
Dream       |       Sleep |  0.73 | [ 0.56,  0.84] |  7.18 | 46 | < .001***
Dream       |        Span | -0.30 | [-0.54, -0.01] | -2.08 | 45 | 0.828
Dream       |        Gest | -0.45 | [-0.66, -0.18] | -3.35 | 44 | 0.055
Dream       |        Pred | -0.45 | [-0.65, -0.19] | -3.47 | 48 | 0.038*
Dream       |         Exp | -0.54 | [-0.71, -0.30] | -4.41 | 48 | 0.002**
Dream       |      Danger | -0.58 | [-0.74, -0.36] | -4.92 | 48 | < .001***
Dream       | NonD_missing | -0.19 | [-0.44,  0.09] | -1.33 | 48 | > .999
Sleep       |        Span | -0.41 | [-0.61, -0.16] | -3.24 | 52 | 0.066
Sleep       |        Gest | -0.63 | [-0.77, -0.44] | -5.87 | 52 | < .001***
Sleep       |        Pred | -0.40 | [-0.59, -0.15] | -3.23 | 56 | 0.066
Sleep       |         Exp | -0.64 | [-0.77, -0.46] | -6.27 | 56 | < .001***
Sleep       |      Danger | -0.59 | [-0.73, -0.39] | -5.44 | 56 | < .001***
Sleep       | NonD_missing | -0.08 | [-0.33,  0.18] | -0.60 | 56 | > .999
Span        |        Gest |  0.61 | [ 0.42,  0.76] |  5.68 | 53 | < .001***
Span        |        Pred | -0.10 | [-0.35,  0.16] | -0.77 | 56 | > .999
Span        |         Exp |  0.36 | [ 0.11,  0.57] |  2.89 | 56 | 0.153
Span        |      Danger |  0.06 | [-0.20,  0.32] |  0.46 | 56 | > .999
Span        | NonD_missing |  0.08 | [-0.18,  0.33] |  0.63 | 56 | > .999
Gest        |        Pred |  0.20 | [-0.06,  0.44] |  1.53 | 56 | > .999
Gest        |         Exp |  0.64 | [ 0.45,  0.77] |  6.20 | 56 | < .001***
Gest        |      Danger |  0.38 | [ 0.13,  0.58] |  3.06 | 56 | 0.098
Gest        | NonD_missing |  0.20 | [-0.06,  0.44] |  1.55 | 56 | > .999
Pred        |         Exp |  0.62 | [ 0.44,  0.75] |  6.09 | 60 | < .001***
Pred        |      Danger |  0.92 | [ 0.86,  0.95] | 17.69 | 60 | < .001***
Pred        | NonD_missing |  0.05 | [-0.20,  0.29] |  0.37 | 60 | > .999
Exp         |      Danger |  0.79 | [ 0.67,  0.87] |  9.89 | 60 | < .001***
Exp         | NonD_missing |  0.25 | [ 0.00,  0.47] |  1.96 | 60 | 0.981
Danger      | NonD_missing |  0.07 | [-0.19,  0.31] |  0.51 | 60 | > .999

p-value adjustment method: Holm (1979)
Observations: 44-62
```

```r
t.test(BodyWgt ~ NonD_missing,
       data = sleep)
```

```
    Welch Two Sample t-test
```

```
data:  BodyWgt by NonD_missing
t = -1.0239, df = 13.351, p-value = 0.3241
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -1502.0470   534.3593
sample estimates:
mean in group 0 mean in group 1
       89.53492       573.37879
```

```
t.test(BrainWgt ~ NonD_missing,
       data = sleep)
```

```

	Welch Two Sample t-test

data:  BrainWgt by NonD_missing
t = -0.96511, df = 14.651, p-value = 0.3502
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -1272.563   480.454
sample estimates:
mean in group 0 mean in group 1
       193.7025        589.7571
```

```
t.test(Sleep ~ NonD_missing,
       data = sleep)
```

```

	Welch Two Sample t-test

data:  Sleep by NonD_missing
t = 0.57598, df = 12.498, p-value = 0.5749
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -2.683161  4.623161
sample estimates:
mean in group 0 mean in group 1
          10.70            9.73
```

```r
t.test(Span ~ NonD_missing,
       data = sleep)
```

```
    Welch Two Sample t-test

data:  Span by NonD_missing
t = -0.86773, df = 38.531, p-value = 0.3909
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -12.021726   4.805658
sample estimates:
mean in group 0 mean in group 1
       19.06889        22.67692
```

```r
t.test(Gest ~ NonD_missing,
       data = sleep)
```

```
    Welch Two Sample t-test

data:  Gest by NonD_missing
t = -1.2435, df = 16.636, p-value = 0.2309
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -185.81590   48.15032
sample estimates:
mean in group 0 mean in group 1
       125.7386        194.5714
```

```r
t.test(Pred ~ NonD_missing,
       data = sleep)
```

```
    Welch Two Sample t-test

data:  Pred by NonD_missing
t = -0.33283, df = 18.543, p-value = 0.743
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
```

```
 -1.2165041  0.8831708
sample estimates:
mean in group 0 mean in group 1
       2.833333        3.000000
```

```r
t.test(Exp ~ NonD_missing,
       data = sleep)
```

```
	Welch Two Sample t-test

data:  Exp by NonD_missing
t = -1.7467, df = 18.278, p-value = 0.09747
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -2.0573419  0.1882943
sample estimates:
mean in group 0 mean in group 1
       2.208333        3.142857
```

```r
t.test(Danger ~ NonD_missing,
       data = sleep)
```

```
	Welch Two Sample t-test

data:  Danger by NonD_missing
t = -0.44672, df = 18.075, p-value = 0.6604
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -1.2726861  0.8262576
sample estimates:
mean in group 0 mean in group 1
       2.562500        2.785714
```

Marginplot of Imputed Values - Example

Let's take the little dataset `x` which is a subset of the `sleep` dataset which has all 62 rows but only the `Dream` and `Sleep` variables.

There are a few built-in imputation functions in `VIM`. Let's see what the `kNN()` (k-nearest neighbor) function does.

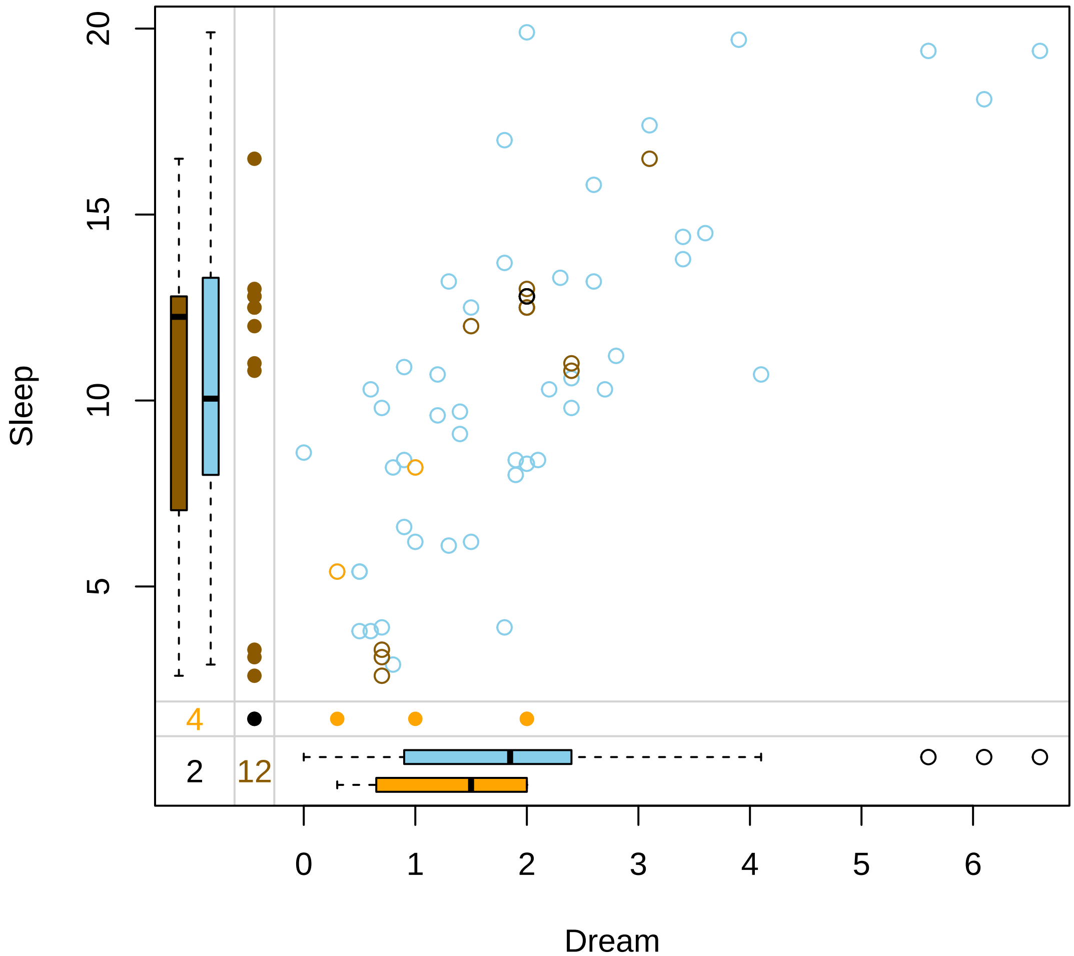
```
x_imputed <- kNN(x)
```

Now look at the plot again for these new `Dream` and `Sleep` variables for the k-nearest neighbor imputed variables. Notice the coloring of the points - the blue are the original values and the other colors represent the structure of missings.

- brown points represent values where Dream was missing initially
- beige points represent values where Sleep was missing initially
- black points represent values where both Dream and Sleep were missing initially

The `kNN()` method appears to preserve the correlation between `Dream` and `Sleep`.

```
marginplot(x_imputed, delimiter = "_imp")
```


Sleep
20
15
10
5
4
2
12
0
1
2
3
4
5
6
Dream

3. Use a survey sampling weight to generate more representative descriptive and inferential statistical values (brief intro)

introduction to survey weights

show how this impacts the amounts of missing data

4. Discuss potential bias when removing missing observations without careful examination.

talk about assumptions for missing data - MCAR, MAR and NMAR (or MNAR)

add more examples here

also for publication - running models for comparison - sensitivity tests - model for all complete data - models based on pairwise selections - n changes - models before and after covariate adjustments - models before and after imputation

---

R Code For This Module

- `module_134.R`

References

Arel-Bundock, Vincent. 2022. "modelsummary: Data and Model Summaries in R." *Journal of Statistical Software* 103 (1): 1–23. https://doi.org/10.18637/jss.v103.i01.

———. 2025a. *Modelsummary: Summary Tables and Plots for Statistical Models and Data: Beautiful, Customizable, and Publication-Ready*. https://modelsummary.com.

———. 2025b. *Tinytable: Simple and Configurable Tables in HTML, LaTeX, Markdown, Word, PNG, PDF, and Typst Formats*. https://vincentarelbundock.github.io/tinytable/.

Horst, Allison, Alison Hill, and Kristen Gorman. 2022. *Palmerpenguins: Palmer Archipelago (Antarctica) Penguin Data*. https://allisonhorst.github.io/palmerpenguins/.

Ihaka, Ross, Paul Murrell, Kurt Hornik, Jason C. Fisher, Reto Stauffer, Claus O. Wilke, Claire D. McWhite, and Achim Zeileis. 2023. *Colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes*. https://colorspace.R-Forge.R-project.org/.

Kowarik, Alexander, and Matthias Templ. 2016. "Imputation with the R Package VIM." *Journal of Statistical Software* 74 (7): 1–16. https://doi.org/10.18637/jss.v074.i07.

R Core Team. 2025. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Stauffer, Reto, Georg J. Mayr, Markus Dabernig, and Achim Zeileis. 2009. "Somewhere over the Rainbow: How to Make Effective Use of Colors in Meteorological Visualizations." *Bulletin of the American Meteorological Society* 96 (2): 203–16. https://doi.org/10.1175/BAMS-D-13-00155.1.

Templ, Matthias, Alexander Kowarik, Andreas Alfons, Gregor de Cillia, and Wolfgang Rannetbauer. 2022. *VIM: Visualization and Imputation of Missing Values*. https://github.com/statistikat/VIM.

Tierney, Nicholas, and Dianne Cook. 2023. "Expanding Tidy Data Principles to Facilitate Missing Data Exploration, Visualization and Assessment of Imputations." *Journal of Statistical Software* 105 (7): 1–31. https://doi.org/10.18637/jss.v105.i07.

Tierney, Nicholas, Di Cook, Miles McBain, and Colin Fay. 2024. *Naniar: Data Structures, Summaries, and Visualisations for Missing Data*. https://github.com/njtierney/naniar.

Waring, Elin, Michael Quinn, Amelia McNamara, Eduardo Arino de la Rubia, Hao Zhu, and Shannon Ellis. 2022. *Skimr: Compact and Flexible Summaries of Data*. https://docs.ropensci.org/skimr/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, and Teun van den Brand. 2024. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://ggplot2.tidyverse.org.

Zeileis, Achim, Jason C. Fisher, Kurt Hornik, Ross Ihaka, Claire D. McWhite, Paul Murrell, Reto Stauffer, and Claus O. Wilke. 2020. "colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes." *Journal of Statistical Software* 96 (1): 1–49. https://doi.org/10.18637/jss.v096.i01.

Zeileis, Achim, Kurt Hornik, and Paul Murrell. 2009. "Escaping RGBland: Selecting Colors for Statistical Graphics." *Computational Statistics & Data Analysis* 53 (9): 3259–70. https://doi.org/10.1016/j.csda.2008.11.033.

Other Helpful Resources

Other Helpful Resources

Missing Data Resources

- CRAN Task View for Missing Data
- R-miss-tastic Website
- Flexible Imputation of Missing Data (online book for 2nd edition) by Stef van Buuren
- more …
- https://www.datawim.com/post/missing-data-visualization-in-r/
- https://libguides.princeton.edu/R-Missingdata
- https://cran.r-project.org/web/packages/mice/index.html
- https://cran.r-project.org/web/views/MissingData.html
- https://rmisstastic.netlify.app/
- https://rmisstastic.netlify.app/tutorials/josse_tierney_bookdown_user2018tutorial_2018
- https://modelsummary.com/vignettes/datasummary.html
- https://dabblingwithdata.amedcalf.com/2018/01/02/my-favourite-r-package-for-summarising-data/
- https://cran.r-project.org/web/packages/summarytools/vignettes/introduction.html
- https://cran.r-project.org/web/packages/skimr/vignettes/skimr.html