Data Cleaning

Fundamental Techniques in Data Science

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Missing Data

Outliers

Data Cleaning

When we receive new data, they are generally messy and contaminated by various anomalies and errors.

- One of the first steps in processing a new set of data is *cleaning*.
- By cleaning the data, we ensure a few properties:
  - The data are in an analyzable format.
  - All data take legal values.
  - Any outliers are located and treated.
  - Any missing data are located and treated.

MISSING DATA

What are Missing Data?

Missing data are empty cells in a dataset where there should be observed values.

- The missing cells correspond to true population values, but we haven’t observed those values.

What are Missing Data?

Missing data are empty cells in a dataset where there should be observed values.

- The missing cells correspond to true population values, but we haven’t observed those values.

Not every empty cell is a missing datum.

- Quality-of-life ratings for dead patients in a mortality study
- Firm profitability after the company goes out of business
- Self-reported severity of menstrual cramping for men
- Empty blocks of data following “gateway” items

Missing Data Pattern

Missing data (or response) patterns represent unique combinations of observed and missing items.

- P items $\Rightarrow 2^P$ possible patterns.

	X	Y
1	x	y
2	x	.
3	.	y
4	.	.

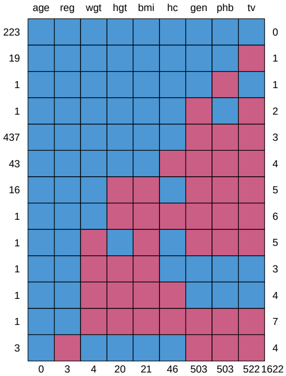
Patterns for $P = 2$

	X	Y	Z
1	x	y	z
2	x	y	.
3	x	.	z
4	.	y	z
5	x	.	.
6	.	.	z
7	.	y	.
8	.	.	.

Patterns for $P = 3$

Example

```r
## Load the mice library:
library(mice)

## Compute missing data patterns:
pats <- md.pattern(boys)
```



	age	reg	wgt	hgt	bmi	hc	gen	phb	tv	
223	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	0
19	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟥	1
1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟥	🟦	1
1	🟦	🟦	🟦	🟦	🟦	🟦	🟥	🟦	🟥	2
437	🟦	🟦	🟦	🟦	🟦	🟦	🟥	🟥	🟥	3
43	🟦	🟦	🟦	🟦	🟦	🟥	🟥	🟥	🟥	4
16	🟦	🟦	🟦	🟥	🟥	🟦	🟥	🟥	🟥	5
1	🟦	🟦	🟦	🟥	🟥	🟥	🟥	🟥	🟥	6
1	🟦	🟦	🟥	🟦	🟥	🟦	🟥	🟥	🟥	5
1	🟦	🟦	🟥	🟥	🟥	🟦	🟦	🟦	🟦	3
1	🟦	🟦	🟥	🟥	🟥	🟥	🟦	🟦	🟦	4
1	🟦	🟦	🟥	🟥	🟥	🟥	🟥	🟥	🟥	7
3	🟦	🟥	🟦	🟦	🟦	🟦	🟥	🟥	🟥	4
	0	3	4	20	21	46	503	503	522	1622

Example

```
pats

    age reg wgt hgt bmi hc gen phb  tv
223   1   1   1   1   1  1   1   1   1    0
19    1   1   1   1   1  1   1   1   0    1
1     1   1   1   1   1  1   1   0   1    1
1     1   1   1   1   1  1   0   1   0    2
437   1   1   1   1   1  1   0   0   0    3
43    1   1   1   1   1  0   0   0   0    4
16    1   1   1   0   0  1   0   0   0    5
1     1   1   1   0   0  0   0   0   0    6
1     1   1   0   1   0  1   0   0   0    5
1     1   1   0   0   0  1   1   1   1    3
1     1   1   0   0   0  0   1   1   1    4
1     1   1   0   0   0  0   0   0   0    7
3     1   0   1   1   1  1   0   0   0    4
      0   3   4  20  21 46 503 503 522 1622
```

Example

```
## How many unique response patterns?
nrow(pats) - 1

[1] 13

## What is the most common response patterns?
maxPat <- rownames(pats) %>% as.numeric() %>% which.max()
pats[maxPat, ]

age reg wgt hgt bmi  hc gen phb  tv
  1   1   1   1   1   1   0   0   0   3
```

Nonresponse Rates

Percent/Proportion Missing

- The proportion of cells containing missing data
- Should be computed for each variable, not for the entire dataset

Attrition Rate

- The proportion of participants that drop-out of a study at each measurement occasion

Covariance Coverage

- The proportion of cases available to estimate a given pairwise relationship (e.g., a covariance between two variables)
- Very important to have adequate coverage for the parameters you want to estimate

Example

We can calculate basic response rates with simple base R commands.

```
## Load some example data:
data(boys, package = "mice")

## Compute variable-wise proportions missing:
mMat <- is.na(boys)
mMat %>% colMeans() %>% round(3)

  age   hgt   wgt   bmi    hc   gen   phb    tv   reg
0.000 0.027 0.005 0.028 0.061 0.672 0.672 0.698 0.004
```

Example

```
## Compute observation-wise proportions missing:
pmRow <- rowMeans(mMat)

## Summarize the above:
range(pmRow)

[1] 0.0000000 0.7777778

pmRow[pmRow > 0] %>% range()

[1] 0.1111111 0.7777778

median(pmRow)

[1] 0.3333333

## Compute the proportion of complete cases:
mean(pmRow == 0)

[1] 0.2981283
```

Example

We can use routines from the **mice** package to calculate covariance coverage and response patterns.

```
## Compute the covariance coverage:
cc <- md.pairs(boys)$rr / nrow(boys)

## Check the result:
round(cc, 2)

     age  hgt  wgt  bmi   hc  gen  phb  tv  reg
age 1.00 0.97 0.99 0.97 0.94 0.33 0.33 0.3 1.00
hgt 0.97 0.97 0.97 0.97 0.92 0.32 0.32 0.3 0.97
wgt 0.99 0.97 0.99 0.97 0.94 0.32 0.32 0.3 0.99
bmi 0.97 0.97 0.97 0.97 0.91 0.32 0.32 0.3 0.97
hc  0.94 0.92 0.94 0.91 0.94 0.33 0.33 0.3 0.93
gen 0.33 0.32 0.32 0.32 0.33 0.33 0.33 0.3 0.33
phb 0.33 0.32 0.32 0.32 0.33 0.33 0.33 0.3 0.33
tv  0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.3 0.30
reg 1.00 0.97 0.99 0.97 0.93 0.33 0.33 0.3 1.00
```

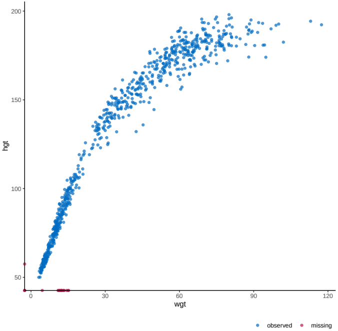
Example

```
## Range of coverages:
range(cc)

[1] 0.2994652 1.0000000

cc[cc < 1] %>% range()

[1] 0.2994652 0.9959893

## How many coverages fall below some threshold?
{cc[lower.tri(cc)] < 0.7} %>% sum()

[1] 21
```

Visualizing Incomplete Data

The **ggmice** package provides some nice ways to visualize incomplete data and objects created during missing data treatment.

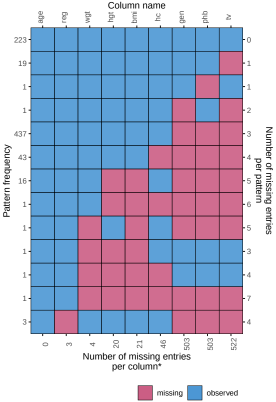
```
library(ggmice); library(ggplot2)

ggmice(boys, aes(wgt, hgt)) + geom_point()
```



Visualizing Incomplete Data

We can also create a nicer version of the response pattern plot.

```r
plot_pattern(boys, rotate = TRUE)
```

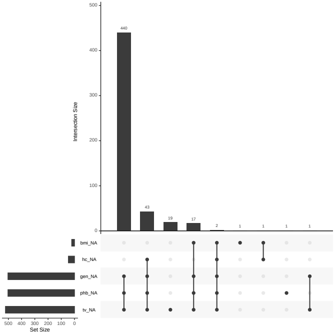


*total number of missing entries: 1622

Visualizing Incomplete Data

The **naniar** package also provides some nice visualization and numerical summary routines for incomplete data.

```
naniar::gg_miss_upset(boys)
```



OUTLIERS

What is an outlier?

We’re only considering *univariate outliers*.

- Extreme values with respect to the distribution of a variable’s other observations
  - A human height measurement of 3 meters
  - A high temperature in Utrecht of 50°
  - Annual income of €250,000 for a student
- Not accounting for any particular model (we’ll get to that later)

What is an outlier?

A univariate outlier may, or may not, be an illegal value.

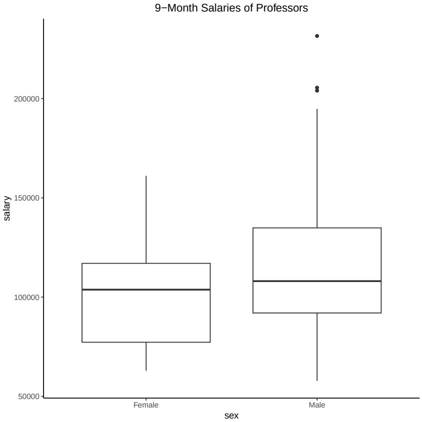
- Data entry errors are probably the most common cause.
- Outliers can also be legal, but extreme, values.

Key Point: We choose to view an outlier as arising from a different population than the one to which we want to generalize our findings.

Finding Outliers: Boxplot Method

Tukey (1977) described a procedure for flagging potential outliers based on a box-and-whiskers plot.

- Does not require normally distributed X
- Not sensitive to outliers



Boxplot Method

A *fence* is an interval defined as the following function of the *first quartile*, the *third quartile*, and the *inner quartile range* ($IQR = Q_3 - Q_1$):

$$F = \{Q_1 - C \times IQR, Q_3 + C \times IQR\}$$

- Taking $C = 1.5$ produces the *inner fence*.
- Taking $C = 3.0$ produces the *outer fence*.

We can use these fences to identify potential outliers:

- Any value that falls outside of the inner fence is a *possible outlier*.
- Any value that falls outside of the outer fence is a *probable outlier*.

Example

We can implement the boxplot method via `boxplot.stats()` .

```
## Find potentially outlying cases:
(out <- boys %$% boxplot.stats(bmi, coef = 3)$out)

[1] 30.62 31.34 31.74

## Which observations are potential outliers?
boys %$% which(bmi %in% out)

[1] 574 668 733

## View the potentially outlying cases:
boys %>% filter(bmi %in% out)

     age   hgt   wgt   bmi   hc  gen  phb tv   reg
1 15.493 182.5 102.0 30.62 57.7 <NA> <NA> NA  west
2 17.749 174.0  94.9 31.34 56.3   G5   P5 25  west
3 19.926 192.3 117.4 31.74 57.6   G5   P6 18 north
```

Breakdown Point

To compare robust statistics, we consider their *breakdown points*.

- The breakdown point is the minimum proportion of cases that must be replaced by ∞ to cause the value of the statistic to go to ∞.

The mean has a breakdown point of $1/N$.

- Replacing a single value with ∞ will produce an infinite mean.
- This is why we shouldn't use basic z-scores to find outliers.

The median has breakdown point of 50%.

- We can replace $n < N/2$ of the observations with ∞ without producing an infinite median.

Outliers for Categorical Data

Nominal, ordinal, and binary items *can* have outliers.

- Outliers on categorical variables are often more indicative of bad variables than outlying cases.

Outliers for Categorical Data

Nominal, ordinal, and binary items *can* have outliers.

- Outliers on categorical variables are often more indicative of bad variables than outlying cases.

Ordinal

- Most participant endorse one of the lowest categories on an ordinal item, but a few participants endorse the highest category.
- The participants who endorse the highest category may be outliers.

Outliers for Categorical Data

Nominal, ordinal, and binary items *can* have outliers.

- Outliers on categorical variables are often more indicative of bad variables than outlying cases.

Ordinal

- Most participant endorse one of the lowest categories on an ordinal item, but a few participants endorse the highest category.
- The participants who endorse the highest category may be outliers.

Nominal

- Groups with very low membership may be outliers on nominal grouping variables.

Outliers for Categorical Data

Nominal, ordinal, and binary items *can* have outliers.

- Outliers on categorical variables are often more indicative of bad variables than outlying cases.

Ordinal

- Most participant endorse one of the lowest categories on an ordinal item, but a few participants endorse the highest category.
- The participants who endorse the highest category may be outliers.

Nominal

- Groups with very low membership may be outliers on nominal grouping variables.

Binary

- If most endorse the item, the few who do not may be outliers.

Treating Outliers

If we locate any outliers, they must be treated.

- Outliers cause by errors, mistakes, or malfunctions (i.e., *error outliers*) should be directly corrected.
- Labeling non-error outliers is a subjective task.
  - A (non-error) outlier must originate from a population separate from the one we care about.
  - Don’t blindly automate the decision process.

Treating Outliers

If we locate any outliers, they must be treated.

- Outliers cause by errors, mistakes, or malfunctions (i.e., *error outliers*) should be directly corrected.
- Labeling non-error outliers is a subjective task.
    - A (non-error) outlier must originate from a population separate from the one we care about.
    - Don’t blindly automate the decision process.

The most direct solution is to delete any outlying observation.

- If you delete non-error outliers, the analysis should be reported twice: with outliers and without.

Treating Outliers

For univariate outliers, we can use less extreme types of deletion.

- Delete outlying values (but not the entire observation).
- These empty cells then become missing data.
- Treat the missing values along with any naturally-occurring nonresponse.

Winsorization:

- Replace the outliers with the nearest non-outlying value.

Treating Outliers

We can also use robust regression procedures to estimate the model directly in the presence of outliers.

- Weight the objective function to reduce the impact of outliers
  - M-estimation
- Trim outlying observations during estimation
  - Least trimmed squares, MCD, MVE
- Take the median, instead of the mean, of the squared residuals
  - Least median of squares
- Model some quantile of the DV’s distribution instead of the mean
  - Quantile regression
- Model the outcome with a heavy-tailed distribution
  - Laplacian, Student’s T

References

Tukey, J. W. (1977). *Exploratory data analysis* (Vol. 2). Reading, MA: Addison–Wesley.