Introduction to Data Science: Missing Data

Héctor Corrada Bravo

University of Maryland, College Park, USA

2019-08-14

Handling Missing Data

We can now move on to a very important aspect of data preparation and transformation: how to deal with missing data?

Values that are unrecorded, unknown or unspecified in a dataset.

Handling Missing Data

```
## # A tibble: 22 x 35
##    id     year month element    d1    d2    d3    d4    d5    d6    d7
##    <chr> <dbl> <dbl> <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 MX17…  2010     1 tmax       NA  NA    NA      NA  NA      NA    NA
##  2 MX17…  2010     1 tmin       NA  NA    NA      NA  NA      NA    NA
##  3 MX17…  2010     2 tmax       NA  27.3  24.1    NA  NA      NA    NA
##  4 MX17…  2010     2 tmin       NA  14.4  14.4    NA  NA      NA    NA
##  5 MX17…  2010     3 tmax       NA  NA    NA      NA  32.1    NA    NA
##  6 MX17…  2010     3 tmin       NA  NA    NA      NA  14.2    NA    NA
##  7 MX17…  2010     4 tmax       NA  NA    NA      NA  NA      NA    NA
##  8 MX17…  2010     4 tmin       NA  NA    NA      NA  NA      NA    NA
##  9 MX17…  2010     5 tmax       NA  NA    NA      NA  NA      NA    NA
```

Handling Missing Data

Temperature observations coded as `NA` are considered *missing*.

- measurement failed in a specific day for a specific weather station, or
- certain stations only measure temperatures on certain days of the month.

Knowing which of these applies can change how we approach this missing data.

Handling Missing Data

Treatment of missing data depends highly on how the data was obtained,

The more you know about a dataset, the better decision you can make.

Handling Missing Data

Central question with missing data is:

Should we *remove* observations with missing values, or should we *impute* missing values?

This also relates to the difference between values that are missing *at random* vs. values that are missing *systematically*.

In the weather example above, the first case (of failed measurements) could be thought of as missing *at random*, and the second case as missing *systematically*.

Handling Missing Data

Data that is missing systematically can significantly bias an analysis.

For example: Suppose we want to predict how sick someone is from test result.

If doctors do not carry out the test because a patient is too sick, then the fact test is missing is a great predictor of how sick the patient is.

Handling Missing Data

The **first step** when dealing with missing data is to understand *why* and *how* data may be missing.

I.e., talk to collaborator, or person who created the dataset.

Once you know that data is not missing systematically and a relatively small fraction of observations contain have missing values, then it may be safe to remove observations.

Dealing with data missing at random

Encoding as missing

For categorical attributes: encode the fact that a value is missing as a new category and in subsequent modeling.

```
## # A tibble: 4 x 6
##   iso2     year sex   age       n iso2_missing
##   <chr>   <dbl> <chr> <chr> <dbl> <lgl>
## 1 missing  1985 m     04       NA TRUE
## 2 missing  1986 m     04       NA TRUE
## 3 AD       1989 m     04       NA FALSE
## 4 AD       1990 m     04       NA FALSE
```

Dealing with data missing at random

Imputation

In the case of numeric values, we can use a simple method for imputation where we replace missing values for a variable with, for instance, the mean of non-missing values

```
library(nycflights13)

flights %>%

  tidyr::replace_na(list(dep_delay=mean(.$dep_delay, na.rm=TRUE)))
```

Dealing with data missing at random

A more complex method is to replace missing values for a variable predicting from other variables when variables are related (we will see linear regression using the `lm` and `predict` functions later on)

```
dep_delay_fit <- flights %>% lm(dep_delay~origin, data=.)

# use average delay conditioned on origin airport
flights %>%
  modelr::add_predictions(dep_delay_fit, var="pred_delay") %>%
  mutate(dep_delay_fixed =
           ifelse(!is.na(dep_delay), dep_delay,
                  pred_delay)) %>%
  select(origin, dest, dep_delay, dep_delay_fixed) %>%
```

Dealing with data missing at random

These two approaches also work for categorical variables:

- Impute with most common category for categorical variables
- Predict category from other attributes using predictive model

Dealing with data missing at random

After imputation it is useful to add an additional indicator attribute stating if a missing value was imputed

```
flights %>%
  mutate(dep_delay_missing = is.na(dep_delay))
```

Dealing with data missing at random

Note that imputing missing values as discussed has two effects.

Central tendency of data is retained

If we impute missing data using the mean of a numeric variable, the mean after imputation will not change.

This is a good reason to impute based on estimates of central tendency.

Dealing with data missing at random

The spread of the data will change

After imputation, the spread of the data will be smaller relative to spread if we ignore missing values.

This could be problematic as underestimating the spread of data can yield over-confident inferences in downstream analysis.

We may not address these issues later, but you should be aware of this.