Missing values

Andrew Roth

2024-06-19

Motivation

A common issue in data analysis is that some values will be missing. For example, when doing patient cohort studies entries in a survey may be left blank by some members of the study. As another example, when dealing with clinical records entries may not have been filled in on some days because the staff forgot.

The first consideration when dealing with missigness is whether the missing values are missing due to systematic reason or it is random. An example of a systematic reason, would be failure to collect information for patients that go to sick during a drug trial. Random missingness on the other hand happen in the example above when staff occassionally leave an entry blank.

The approaches we will discuss for dealing with missing values typically assume missing at random. Thus before applying these approaches it is important to think carefully about whether that assumption holds. You should also report that you assumed data was missing at random if so. It is also good practice to check whether the approach used for dealing with missingness impacts your results.

Simple strategies

Setup

First we will create some a toy dataset with missing values. I will use the builtin airquality R dataset and add missing values.

```r
# Reassign the dataset so we can see it in RStudio
df <- airquality
# Check how many entries we have
nrow(df)
```

```
## [1] 153
```

```r
# Get some summary info about the data columns
summary(df)
```

```
##      Ozone           Solar.R           Wind             Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
##  NA's   :37       NA's   :7                                       
##      Month            Day      
##  Min.   :5.000   Min.   : 1.0  
##  1st Qu.:6.000   1st Qu.: 8.0  
##  Median :7.000   Median :16.0  
##  Mean   :6.993   Mean   :15.8  
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

From the summary we can see that two columns, Ozone (NA=37) and Solar.R (NA=7), have missing values.

Removing missing values

Given the size of the dataset n=153 we could potentially remove observations with missing values in these columns. For Solar.R that is probably okay, as it is small fraction of the data. But for Ozone it would be challenging, as it is more thant 20% of the data.

Let's take a look at how we could remove rows with missing values for Solar.R

```
df <- df[!is.na(df$Solar.R),]
nrow(df)
```

```
## [1] 146
```

```
summary(df)
```

```
##      Ozone          Solar.R           Wind            Temp           Month
##  Min.   :  1.0   Min.   :  7.0   Min.   : 1.7   Min.   :57.00   Min.   :5.000
##  1st Qu.: 18.0   1st Qu.:115.8   1st Qu.: 7.4   1st Qu.:73.00   1st Qu.:6.000
##  Median : 31.0   Median :205.0   Median : 9.7   Median :79.00   Median :7.000
##  Mean   : 42.1   Mean   :185.9   Mean   :10.0   Mean   :78.12   Mean   :7.027
##  3rd Qu.: 62.0   3rd Qu.:258.8   3rd Qu.:11.5   3rd Qu.:84.00   3rd Qu.:8.000
##  Max.   :168.0   Max.   :334.0   Max.   :20.7   Max.   :97.00   Max.   :9.000
##  NA's   :35
##       Day
##  Min.   : 1.00
##  1st Qu.: 9.00
##  Median :16.00
##  Mean   :16.12
##  3rd Qu.:23.75
##  Max.   :31.00
##
```

We use the `is.na` function to find out if an entry of the Solar.R column is missing. We then filter for the rows where values are not missing using the not `!` operator. The code `df[!is.na(df$Solar.R),]` cases give us all rows of `df` where Solar.R is not missing.

As we can see from the summary there are no more missing values in Solar.R and our dataset has shrunk in size.

Simple imputation

A common approach to dealing with missing values is to impute them. There are many strategies, in fact an entire field of statistics that thinks about this. The simplest strategies are to replace the values with some summary statistic. For continuous values the mean or median is often used. For discrete values the mean doesn't really make sense, as it usually won't be an integer. So for discrete values, usually the median or sometimes the mode (most often occurring value is used). The mode is also a reasonable choice for categorical values.

I will use the median for Ozone.

In the following code I assign all missing entries of Ozone to the median value of the observed entries.

```
df$Ozone[is.na(df$Ozone)] <- median(df$Ozone, na.rm=TRUE)
summary(df)
```

```
##      Ozone           Solar.R           Wind            Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.7   Min.   :57.00  
##  1st Qu.: 21.00   1st Qu.:115.8   1st Qu.: 7.4   1st Qu.:73.00  
##  Median : 31.00   Median :205.0   Median : 9.7   Median :79.00  
##  Mean   : 39.44   Mean   :185.9   Mean   :10.0   Mean   :78.12  
##  3rd Qu.: 45.75   3rd Qu.:258.8   3rd Qu.:11.5   3rd Qu.:84.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.7   Max.   :97.00  
##      Month            Day       
##  Min.   :5.000   Min.   : 1.00  
##  1st Qu.:6.000   1st Qu.: 9.00  
##  Median :7.000   Median :16.00  
##  Mean   :7.027   Mean   :16.12  
##  3rd Qu.:8.000   3rd Qu.:23.75  
##  Max.   :9.000   Max.   :31.00  
```

Advanced imputation

The key weakness of the simple imputation strategy we used is that it ignores the values of other observed variables in a row. More sophisticated imputation strategies will leverage these observations to make a more sophisticated imputation. I will illustrate one more advanced approach for imputation using the `mice` package from R. However, I am not endorsing this as the only or even best way. Depending on the type of data you have in your columns i.e. categorical, ordinal, continuous you will want to pick wisely. At this stage it would be worth consulting a statistician before proceeding.

Disclaimer aside, let's use `mice` to impute. I am going to reload the data so we get the missing values back.

```r
library(mice)
```

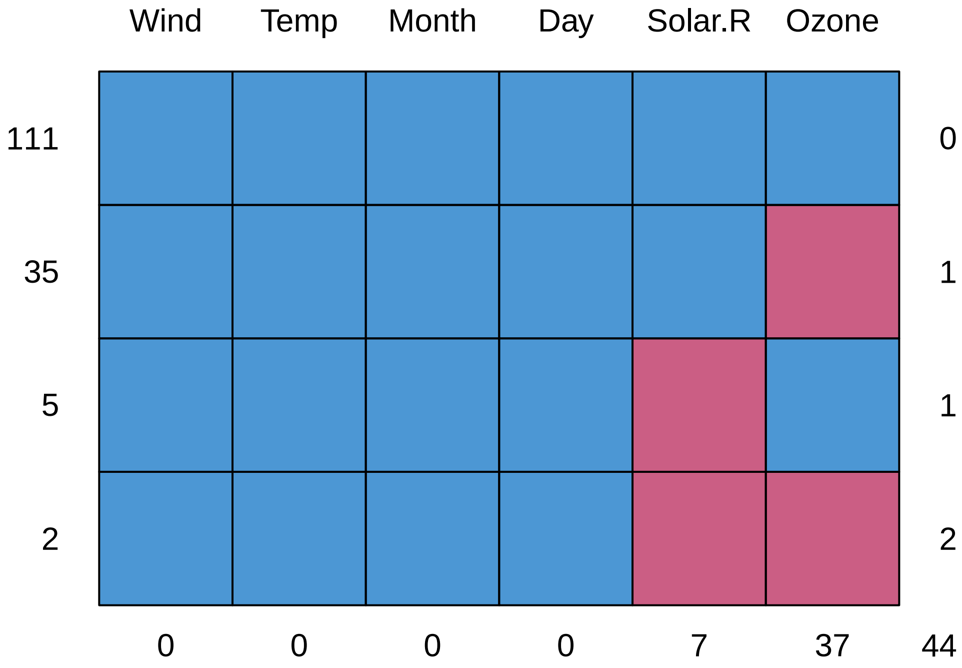
```
## 
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
## 
##     filter

## The following objects are masked from 'package:base':
## 
##     cbind, rbind
```

```r
df <- airquality
summary(df)
```

```
##      Ozone           Solar.R           Wind             Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
##  NA's   :37       NA's   :7                                       
##      Month            Day      
##  Min.   :5.000   Min.   : 1.0  
##  1st Qu.:6.000   1st Qu.: 8.0  
##  Median :7.000   Median :16.0  
##  Mean   :6.993   Mean   :15.8  
##  3rd Qu.:8.000   3rd Qu.:23.0  
##  Max.   :9.000   Max.   :31.0
##
```

One nice feature of `mice` is that you visualize the "pattern" of missing values.

```
md.pattern(df)
```



```
##     Wind Temp Month Day Solar.R Ozone
## 111    1    1     1   1       1     1  0
## 35     1    1     1   1       1     0  1
## 5      1    1     1   1       0     1  1
## 2      1    1     1   1       0     0  2
##        0    0     0   0       7    37 44
```

From this we can see there are 35 rows missing only Ozone, 5 rows missing only Solar.R and 2 rows missing both.

Now, let's do the imputations using the `mice` function.

```
imp <- mice(df, m=5, method="pmm")
```

```
##
##  iter imp variable
##   1   1  Ozone  Solar.R
##   1   2  Ozone  Solar.R
##   1   3  Ozone  Solar.R
##   1   4  Ozone  Solar.R
##   1   5  Ozone  Solar.R
##   2   1  Ozone  Solar.R
##   2   2  Ozone  Solar.R
##   2   3  Ozone  Solar.R
##   2   4  Ozone  Solar.R
##   2   5  Ozone  Solar.R
##   3   1  Ozone  Solar.R
##   3   2  Ozone  Solar.R
##   3   3  Ozone  Solar.R
##   3   4  Ozone  Solar.R
##   3   5  Ozone  Solar.R
##   4   1  Ozone  Solar.R
##   4   2  Ozone  Solar.R
##   4   3  Ozone  Solar.R
##   4   4  Ozone  Solar.R
##   4   5  Ozone  Solar.R
##   5   1  Ozone  Solar.R
##   5   2  Ozone  Solar.R
##   5   3  Ozone  Solar.R
##   5   4  Ozone  Solar.R
##   5   5  Ozone  Solar.R
```

```r
summary(imp)
```

```
## Class: mids
## Number of multiple imputations:  5 
## Imputation methods:
##   Ozone Solar.R    Wind    Temp   Month     Day 
##   "pmm"   "pmm"      ""      ""      ""      "" 
## PredictorMatrix:
##         Ozone Solar.R Wind Temp Month Day
## Ozone       0       1    1    1     1   1
## Solar.R     1       0    1    1     1   1
## Wind        1       1    0    1     1   1
## Temp        1       1    1    0     1   1
## Month       1       1    1    1     0   1
## Day         1       1    1    1     1   0
```

The `mice` function actually returns an imputation object. To get the imputed data frame we need to use the `complete` command, passing the imputed object.

```r
df_imp <- complete(imp)
summary(df_imp)
```

```
##      Ozone           Solar.R           Wind             Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 18.00   1st Qu.:115.0   1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 34.00   Median :212.0   Median : 9.700   Median :79.00  
##  Mean   : 41.81   Mean   :186.6   Mean   : 9.958   Mean   :77.88  
##  3rd Qu.: 61.00   3rd Qu.:259.0   3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
##      Month            Day      
##  Min.   :5.000   Min.   : 1.0  
##  1st Qu.:6.000   1st Qu.: 8.0  
##  Median :7.000   Median :16.0  
##  Mean   :6.993   Mean   :15.8  
##  3rd Qu.:8.000   3rd Qu.:23.0  
##  Max.   :9.000   Max.   :31.0
```

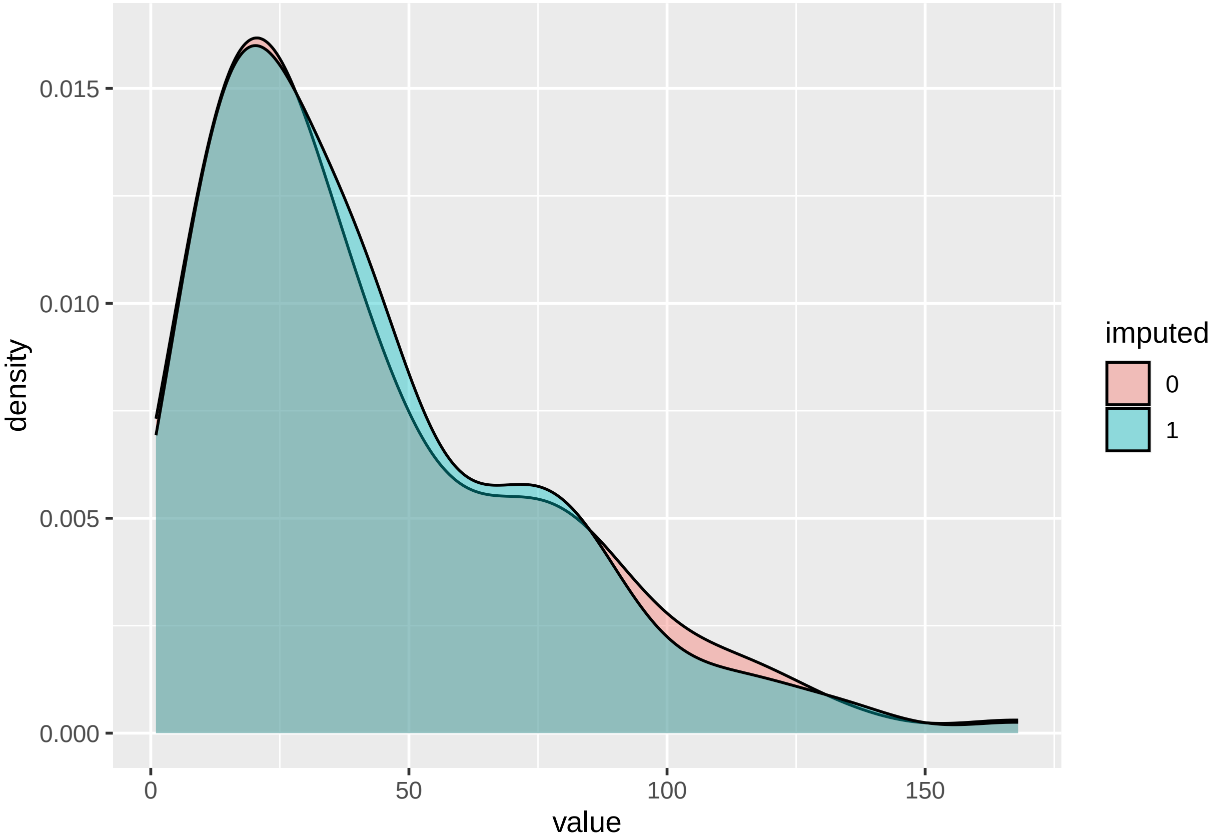
No we see that there are no missing values in either Ozone or Solar.R.

Finally, we will check that the imputation has not drastically changed the characteristics of our data. To do so we will look at a plot of the density of Ozone values before and after imputation.

In the code below I create a new data frame for the plotting and use ggplot to plot densities.

```r
library(ggplot2)

df_plot <- data.frame(
  value=c(df$Ozone, df_imp$Ozone),
  imputed=factor(c(rep(0, nrow(df)), rep(1, nrow(df_imp))))
)
ggplot(df_plot, aes(x=value, fill=imputed)) + geom_density(alpha=0.4)
```

```
## Warning: Removed 37 rows containing non-finite outside the scale range
## (`stat_density()`).
```



We see that unimputed and imputed data have very similar distributions for the Ozone variable. Importantly the peaks are in the same place, and the tails have not shifted drastically.

You can also use `mice` to impute categorical variables using the same procedure as above. Just remember to use `factor` to let R now that your variables are categorical.