missing data

Contents

Missing data are quite common in real datasets. This may be driven by a number of reasons including, data was not available, was not gathered, or coded incorrectly.

A number of predictive models require complete data. And, even among the predictive models that can function with less than complete data, missing data can only exist in predictors not the response variable. Before we proceed to find a solution to missing data let us examine the reasons for it. The reasons may be broadly categorized as 1. Random: The cause of missing data may be completely random. Survey respondents may neglect to answer certain questions by chance. Person coding data may commit a data entry error. 2. Data deficiency: This may be a missing component of a predictor. Consider a survey containing the alternatives male and female for a question on gender. Those not conforming to a particular gender may not respond.
3. Specific Causes: There are situations where missing data can be clearly attributed to a cause. Survey respondents may refuse to respond to questions on politically sensitive issues such as abortion and right to bear arms. In clinical studies where patients are measured periodically over time, a patient may drop off due to an adverse side effect.

Examine Missing Data

The pattern of missing data is likely to shed light on the reasons behind it. For small and medium datasets, it is possible to visualize missing data in a chart such as a heatmap. When the dataset has a large number of variables or observations, the data must be suitably condensed before visualizing.

To illustrate, let us create some missing data in the `mtcars` dataset

```
mtcars_missing = mtcars
for(i in 1:35){
  set.seed(i)
  x = sample(1:nrow(mtcars), 1)
  y = sample(1:ncol(mtcars), 1)
  mtcars_missing[x, y] = NA
}
mtcars_missing
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4   NA
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1   NA   NA
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
```

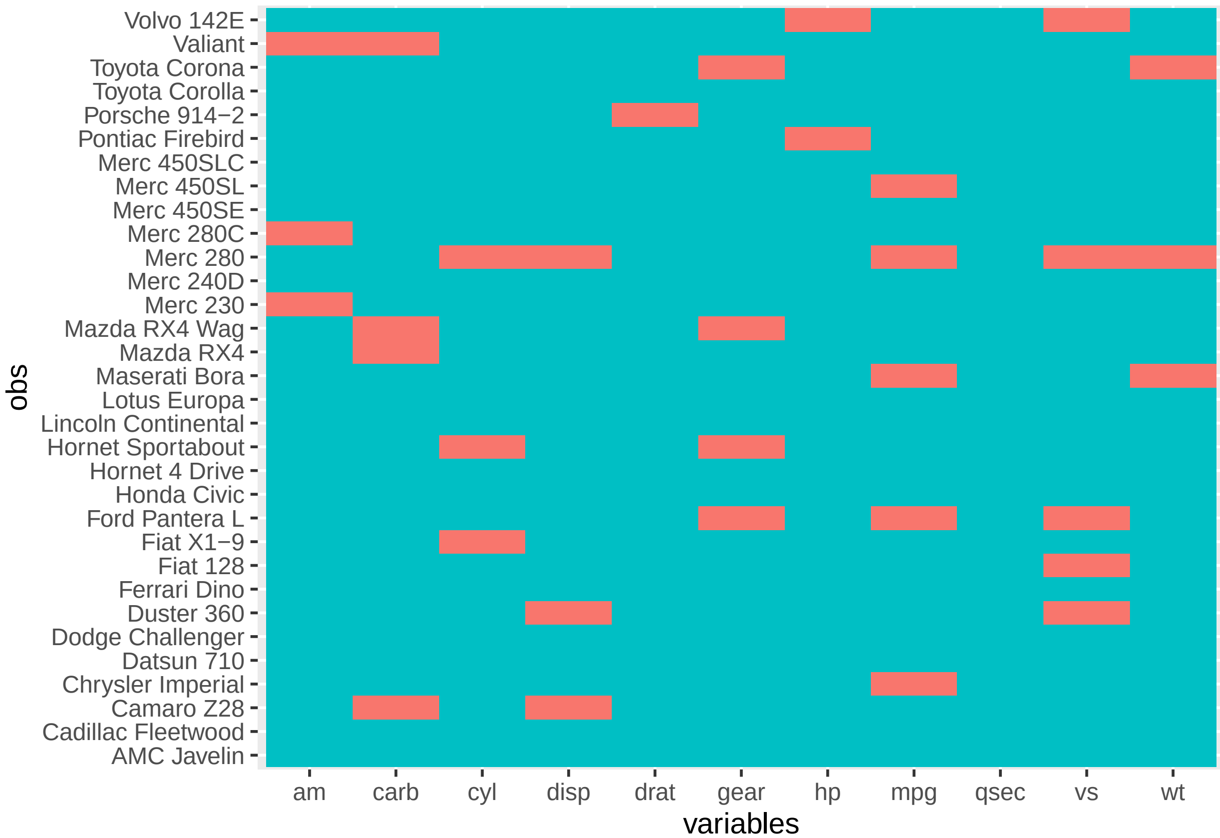
```
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7  NA 360.0 175 3.15 3.440 17.02  0  0   NA    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1 NA    3   NA
## Duster 360          14.3   8    NA 245 3.21 3.570 15.84 NA  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1 NA    4    2
## Merc 280              NA  NA    NA 123 3.92    NA 18.30 NA  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1 NA    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL            NA   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial     NA   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47 NA  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70    NA 20.01  1  0   NA    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8    NA 245 3.73 3.840 15.41  0  0    3   NA
## Pontiac Firebird    19.2   8 400.0  NA 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3  NA  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91   NA 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L        NA   8 351.0 264 4.22 3.170 14.50 NA  1   NA    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora         NA   8 301.0 335 3.54    NA 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0  NA 4.11 2.780 18.60 NA  1    4    2
```

When the dataset has a reasonable number of observations and variables, a simple method to visualize missing information is a heatmap. With library(ggplot2), `geom_tile` can be used to generate a simple heatmap.

We first classify all observations based on whether they are missing (0) or not (1)

```
mtcars_missing_bi = mtcars_missing
mtcars_missing_bi[!is.na(mtcars_missing)] = 1
mtcars_missing_bi[is.na(mtcars_missing)] = 0
```

```
library(tidyr); library(dplyr); library(ggplot2)
mtcars_missing_bi %>%
  as_tibble()%>%
  mutate(obs = rownames(mtcars_missing_bi))%>%
  select(obs, everything())%>%
  pivot_longer(cols = 2:12, names_to = 'variables', values_to = 'values' )%>%
  ggplot(aes(variables, obs, fill= factor(values))) +
  geom_tile()+
  guides(fill=F)
```



Identifying patterns in a moderate sized heatmap can be challenging. To aid intepretation, `library(ComplexHeatMap)` groups variables and observations and represents the groups using a dendrogram. The author has a nice book on the bells and whistles of `ComplexHeatMap`.

`ComplexHeatMap` can be installed from Bioconductor

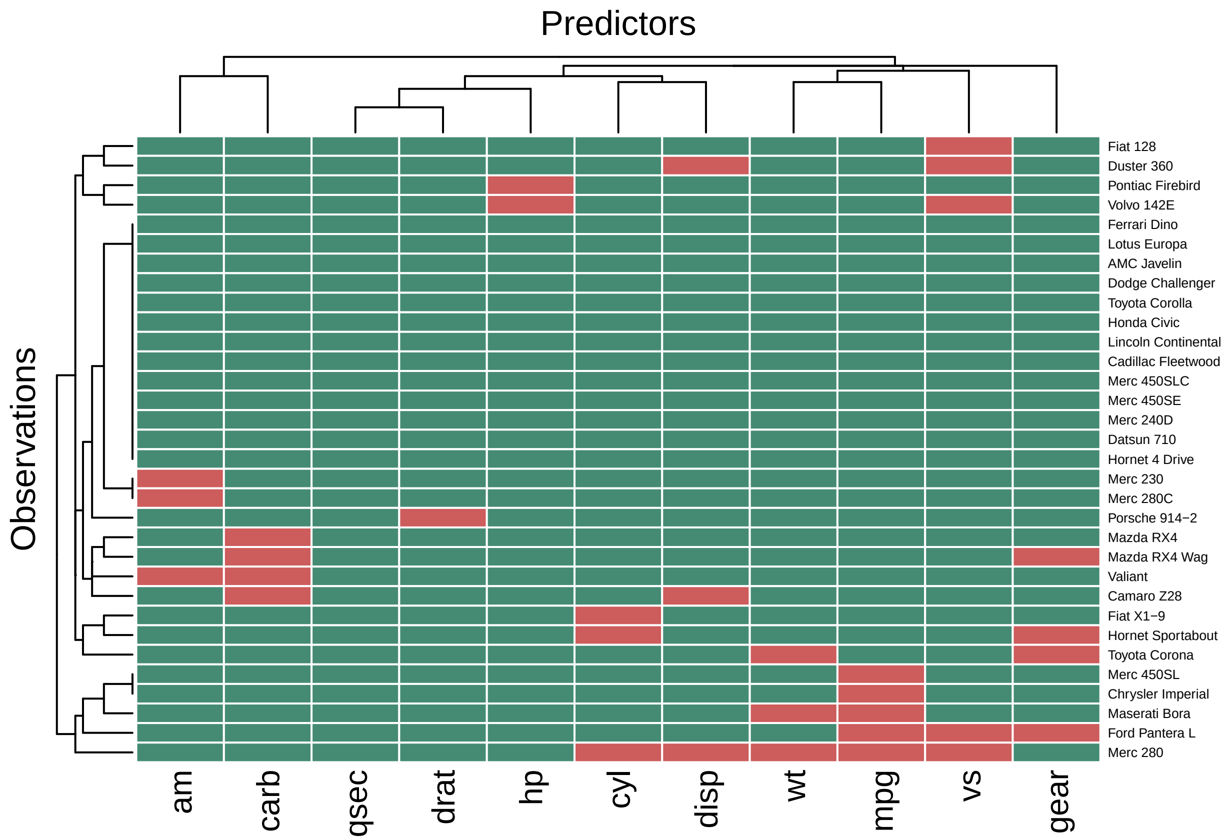
```
# install.packages('BiocManager')
# BiocManager::install('ComplexHeatmap')
```

Alternatively, the most recent version can be obtained from github

```
#library(devtools)
#install_github("jokergoo/ComplexHeatmap")
```

The following heatmap groups the tiles to make it easier to spot patterns in missing data.

```
library(ComplexHeatmap)
library(circlize)
Heatmap(mtcars_missing_bi,
        rect_gp = gpar(col = "white", lwd = 1),
        name = 'Seeing What is Missing',
        column_title = 'Predictors',
        row_title = 'Observations',
        col = circlize::colorRamp2(c(0,1),c('indianred','aquamarine4')),
        show_heatmap_legend = F,
        row_names_gp = gpar(fontsize = 5))
```



For larger datasets, the data needs to be reduced (using a technique such as principal components analysis) before constructing a heatmap. Another valuable tool for examining the nature and degree of missing data are numerical summaries. For the `mtcars_missing`, for each variable, here are the (a) number of missing values

```
apply(mtcars_missing,
      MARGIN = 2,
      FUN = function(x) sum(is.na(x)))
```

```
##  mpg  cyl disp   hp drat   wt qsec   vs   am gear carb
##    5    3    3    2    1    3    0    5    3    4    4
```

(b) percent of missing values

```
apply(mtcars_missing,
      MARGIN = 2,
      FUN = function(x) 100*sum(is.na(x))/(sum(is.na(x))+ sum(!is.na(x))))
```

```
##    mpg    cyl   disp     hp   drat     wt   qsec     vs     am   gear   carb
## 15.625  9.375  9.375  6.250  3.125  9.375  0.000 15.625  9.375 12.500 12.500
```

Solutions

Before we examine solutions, it is worth noting that while many popular predictive models such as linear regression, penalized regression models, support vector machines and neural networks cannot tolerate any amount of missing values, there are a few predictive models such as classification and regression trees and other tree-based models that can handle incomplete data.

For the vast majority of predictive models and also a number unsupervised learning methods, missing values need to be addressed. There are three broad categories of solutions, (a) delete (or ignore) missing values, (b) encode missing data, (c) impute.

Delete

By far, the simplest solution to missing data is remove them either by deleting or ignoring. The most common approach is to conduct listwise deletion wherein the entire row is deleted if even a single element is missing. In `mtcars_missing`, the following rows are missing at least one value.

```
apply(mtcars_missing, MARGIN = 1, function(x) any(is.na(x)))
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive
##                TRUE                TRUE               FALSE               FALSE
##   Hornet Sportabout             Valiant          Duster 360           Merc 240D
##                TRUE                TRUE                TRUE               FALSE
##            Merc 230            Merc 280           Merc 280C          Merc 450SE
##                TRUE                TRUE                TRUE               FALSE
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental
##                TRUE               FALSE               FALSE               FALSE
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla
##                TRUE                TRUE               FALSE               FALSE
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28
##                TRUE               FALSE               FALSE                TRUE
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
##                TRUE                TRUE                TRUE               FALSE
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
##                TRUE               FALSE                TRUE                TRUE
```

Removing them leaves only 13 of the 32 rows in `mtcars_missing`!

```
mtcars_missing[!apply(mtcars_missing, MARGIN = 1, function(x) any(is.na(x))),]
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

Many predictive modeling functions (e.g., `lm`) will automatically conduct listwise deletion. Note in the following output, "19 observations deleted due to missingness"

```
summary(lm(mpg~.,mtcars_missing))
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars_missing)
##
## Residuals:
```

```
##          Datsun 710      Hornet 4 Drive           Merc 240D           Merc 450SE
##          -1.108e+00          -7.352e-01          -6.661e-16           9.495e-01
##         Merc 450SLC  Cadillac Fleetwood Lincoln Continental          Honda Civic
##           2.421e-01           2.706e-01           2.980e-02           4.134e-01
##      Toyota Corolla    Dodge Challenger         AMC Javelin        Lotus Europa
##           6.943e-01           1.810e+00          -2.566e+00           7.352e-01
##        Ferrari Dino
##          -7.352e-01
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -280.98109  166.37953  -1.689    0.233
## cyl           19.61883   13.19206   1.487    0.275
## disp          -0.04405    0.04758  -0.926    0.452
## hp            -0.17579    0.16670  -1.055    0.402
## drat           7.01319    5.20404   1.348    0.310
## wt             8.77079   11.02143   0.796    0.510
## qsec           3.41823    2.04149   1.674    0.236
## vs            26.01966   19.08962   1.363    0.306
## am            10.87792   10.71197   1.015    0.417
## gear          25.82345   15.41268   1.675    0.236
## carb          -3.63447    3.04106  -1.195    0.355
##
## Residual standard error: 2.683 on 2 degrees of freedom
##   (19 observations deleted due to missingness)
## Multiple R-squared:  0.9794, Adjusted R-squared:  0.8765
## F-statistic: 9.518 on 10 and 2 DF,  p-value: 0.09876
```

An important consideration in using deletion is loss of sample. With listwise deletion, even a small number of missing elements can result in a large number of rows getting deleted. Consider the above example where the original data was missing only 11.34375% values but listwise deletion eliminated 59.375% of the rows

```
sum(is.na(mtcars_missing))/nrow(mtcars)*ncol(mtcars) # Percent missing values
```

```
## [1] 11.34375
```

```
100* nrow(mtcars_missing[apply(mtcars_missing, MARGIN = 1, function(x) any(is.na(x))),])/nrow(mtcars_mis
```

```
## [1] 59.375
```

A second consideration is the cost of acquiring more observations.

A third and perhaps most important consideration is the likelihood of introducing bias into the model. Consider a clinical study where half the respondents receive the current standard care of treatment while the other half receive the new treatment. The new treatment may induce adverse effects causing patients to drop out of the study. Such missing data is clearly not random and deleting them is bound to bias results.

Encode Missing Data

Missing data may represent information. In a survey that only contains two alternatives for gender, those not conforming to either gender may skip the question. In this case, non-response may contain information valuable gender orientation. As another example, survey respondents may refuse to respond to questions on politically sensitive issues such as abortion and right to bear arms. Here again, non-response may contain useful information.

In a categorical variable, missing data may be encoded as another level. Consider the following example.

```
set.seed(61710)
gender = sample(c('Male','Female',NA),10,T)
```

```r
gender
```

```
##  [1] NA       "Female" "Male"   NA       "Male"   "Male"   "Female" "Male"
##  [9] "Female" "Female"
```

```r
gender_encoded = gender
gender_encoded[is.na(gender)] = 'Did Not Respond'
gender_encoded
```

```
##  [1] "Did Not Respond" "Female"          "Male"            "Did Not Respond"
##  [5] "Male"            "Male"            "Female"          "Male"
##  [9] "Female"          "Female"
```

Missing data in a continuous variable may be represented using a dummy variable.

```r
mtcars_missing_encoded = mtcars_missing
mtcars_missing_encoded %>%
  mutate(cyl_missing = ifelse(is.na(mtcars_missing_encoded$cyl),0,1))%>%
  select(mpg, cyl, cyl_missing, everything())
```

```
##                      mpg cyl cyl_missing  disp  hp drat    wt  qsec vs am gear
## Mazda RX4           21.0   6           1 160.0 110 3.90 2.620 16.46  0  1    4
## Mazda RX4 Wag       21.0   6           1 160.0 110 3.90 2.875 17.02  0  1   NA
## Datsun 710          22.8   4           1 108.0  93 3.85 2.320 18.61  1  1    4
## Hornet 4 Drive      21.4   6           1 258.0 110 3.08 3.215 19.44  1  0    3
## Hornet Sportabout   18.7  NA           0 360.0 175 3.15 3.440 17.02  0  0   NA
## Valiant             18.1   6           1 225.0 105 2.76 3.460 20.22  1 NA    3
## Duster 360          14.3   8           1    NA 245 3.21 3.570 15.84 NA  0    3
## Merc 240D           24.4   4           1 146.7  62 3.69 3.190 20.00  1  0    4
## Merc 230            22.8   4           1 140.8  95 3.92 3.150 22.90  1 NA    4
## Merc 280              NA  NA           0    NA 123 3.92    NA 18.30 NA  0    4
## Merc 280C           17.8   6           1 167.6 123 3.92 3.440 18.90  1 NA    4
## Merc 450SE          16.4   8           1 275.8 180 3.07 4.070 17.40  0  0    3
## Merc 450SL            NA   8           1 275.8 180 3.07 3.730 17.60  0  0    3
## Merc 450SLC         15.2   8           1 275.8 180 3.07 3.780 18.00  0  0    3
## Cadillac Fleetwood  10.4   8           1 472.0 205 2.93 5.250 17.98  0  0    3
## Lincoln Continental 10.4   8           1 460.0 215 3.00 5.424 17.82  0  0    3
## Chrysler Imperial     NA   8           1 440.0 230 3.23 5.345 17.42  0  0    3
## Fiat 128            32.4   4           1  78.7  66 4.08 2.200 19.47 NA  1    4
## Honda Civic         30.4   4           1  75.7  52 4.93 1.615 18.52  1  1    4
## Toyota Corolla      33.9   4           1  71.1  65 4.22 1.835 19.90  1  1    4
## Toyota Corona       21.5   4           1 120.1  97 3.70    NA 20.01  1  0   NA
## Dodge Challenger    15.5   8           1 318.0 150 2.76 3.520 16.87  0  0    3
## AMC Javelin         15.2   8           1 304.0 150 3.15 3.435 17.30  0  0    3
## Camaro Z28          13.3   8           1    NA 245 3.73 3.840 15.41  0  0    3
## Pontiac Firebird    19.2   8           1 400.0  NA 3.08 3.845 17.05  0  0    3
## Fiat X1-9           27.3  NA           0  79.0  66 4.08 1.935 18.90  1  1    4
## Porsche 914-2       26.0   4           1 120.3  91   NA 2.140 16.70  0  1    5
## Lotus Europa        30.4   4           1  95.1 113 3.77 1.513 16.90  1  1    5
## Ford Pantera L        NA   8           1 351.0 264 4.22 3.170 14.50 NA  1   NA
## Ferrari Dino        19.7   6           1 145.0 175 3.62 2.770 15.50  0  1    5
## Maserati Bora         NA   8           1 301.0 335 3.54    NA 14.60  0  1    5
## Volvo 142E          21.4   4           1 121.0  NA 4.11 2.780 18.60 NA  1    4
##                     carb
## Mazda RX4             NA
## Mazda RX4 Wag         NA
## Datsun 710             1
```

```
## Hornet 4 Drive         1
## Hornet Sportabout      2
## Valiant               NA
## Duster 360             4
## Merc 240D              2
## Merc 230               2
## Merc 280               4
## Merc 280C              4
## Merc 450SE             3
## Merc 450SL             3
## Merc 450SLC            3
## Cadillac Fleetwood     4
## Lincoln Continental    4
## Chrysler Imperial      4
## Fiat 128               1
## Honda Civic            2
## Toyota Corolla         1
## Toyota Corona          1
## Dodge Challenger       2
## AMC Javelin            2
## Camaro Z28            NA
## Pontiac Firebird       2
## Fiat X1-9              1
## Porsche 914-2          2
## Lotus Europa           2
## Ford Pantera L         4
## Ferrari Dino           6
## Maserati Bora          8
## Volvo 142E             2
```

Impute

In imputation, information and relationships among non-missing predictors is used to estimate missing values. Approaches to imputation differ based on whether they are designed for inferences or prediction. Our focus is going to be on the latter, so the quality of an imputation procedure will be judged by its ability to accurately predict missing values.

It is important to remember that imputed values are merely estimates of the true value for missing data. Thus, one must limit how much missing data is imputed. While there isn't a universal rule on how much data can be imputed, no more than 20% for a variable may be a good thumb rule to use.

Imputation must occur prior to other steps in data preparation.

There are a number of methods available for imputing missing data (e.g., predictive mean matching, k-nearest neighbors, trees and tree-based methods) and a number of R package that implement these (e.g., `mice`, `caret`)

mice

mice() implements a number of imputation methods. This illustration uses the default, which for numeric data is predictive mean matching. For more information, see author's website. In version 3.12.0, mice library implemented a new matchindex C function that makes predictive mean matching 50 to 600 times faster, however this affects reproducibility of the algorithm. Read more about the update (here)[https://cran.r-project.org/web/packages/mice/news/news.html]. (If you wish to reproduce the behavior of the previous version of `mice`, include use.matcher=T in the mice function). Finally, `tidyr` has an identically named `complete` function. To prevent conflicts, it is best to include the package reference, mice::complete(…)

```r
library(mice)
mtcars_mice = mice::complete(mice(mtcars_missing,seed = 617))
```

```
##
##  iter imp variable
##   1   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
```

```r
head(mtcars_mice)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    3
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```r
library(mice)
mtcars_mice_rf = mice::complete(mice(mtcars_missing,method = 'rf',seed = 617))
```

```
##
##  iter imp variable
##   1   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   1   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   2   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
```

```
##   3   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   3   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   4   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   1  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   2  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   3  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   4  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
##   5   5  mpg  cyl  disp  hp  drat  wt  vs  am  gear  carb
```

```
head(mtcars_mice_rf)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    2
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    5    1
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

caret

caret is a full blown machine learning framework that contains a number of handy functions. preprocess() can be used for imputing, among other things. Here, we are using a bagImpute, which works by fitting a bagged tree model for each predictor (as a function of all the others). This method is quite accurate, but has much higher computational cost than say medianImpute.

```
library(caret)
set.seed(617)
mtcars_caret = predict(preProcess(mtcars_missing,
                                  method = 'bagImpute'),
                       newdata = mtcars_missing)
head(mtcars_caret)
```

```
##                    mpg      cyl disp  hp drat    wt  qsec vs        am     gear
## Mazda RX4         21.0 6.000000  160 110 3.90 2.620 16.46  0 1.0000000 4.000000
## Mazda RX4 Wag     21.0 6.000000  160 110 3.90 2.875 17.02  0 1.0000000 3.615385
## Datsun 710        22.8 4.000000  108  93 3.85 2.320 18.61  1 1.0000000 4.000000
## Hornet 4 Drive    21.4 6.000000  258 110 3.08 3.215 19.44  1 0.0000000 3.000000
## Hornet Sportabout 18.7 6.246154  360 175 3.15 3.440 17.02  0 0.0000000 3.615385
## Valiant           18.1 6.000000  225 105 2.76 3.460 20.22  1 0.3538462 3.000000
##                       carb
## Mazda RX4         2.438462
## Mazda RX4 Wag     2.438462
## Datsun 710        1.000000
## Hornet 4 Drive    1.000000
## Hornet Sportabout 2.000000
## Valiant           2.438462
```

There are a number of imputation methods available but no single method is the best in every situation. Here is a comparison of the different imputation methods on the missing data.

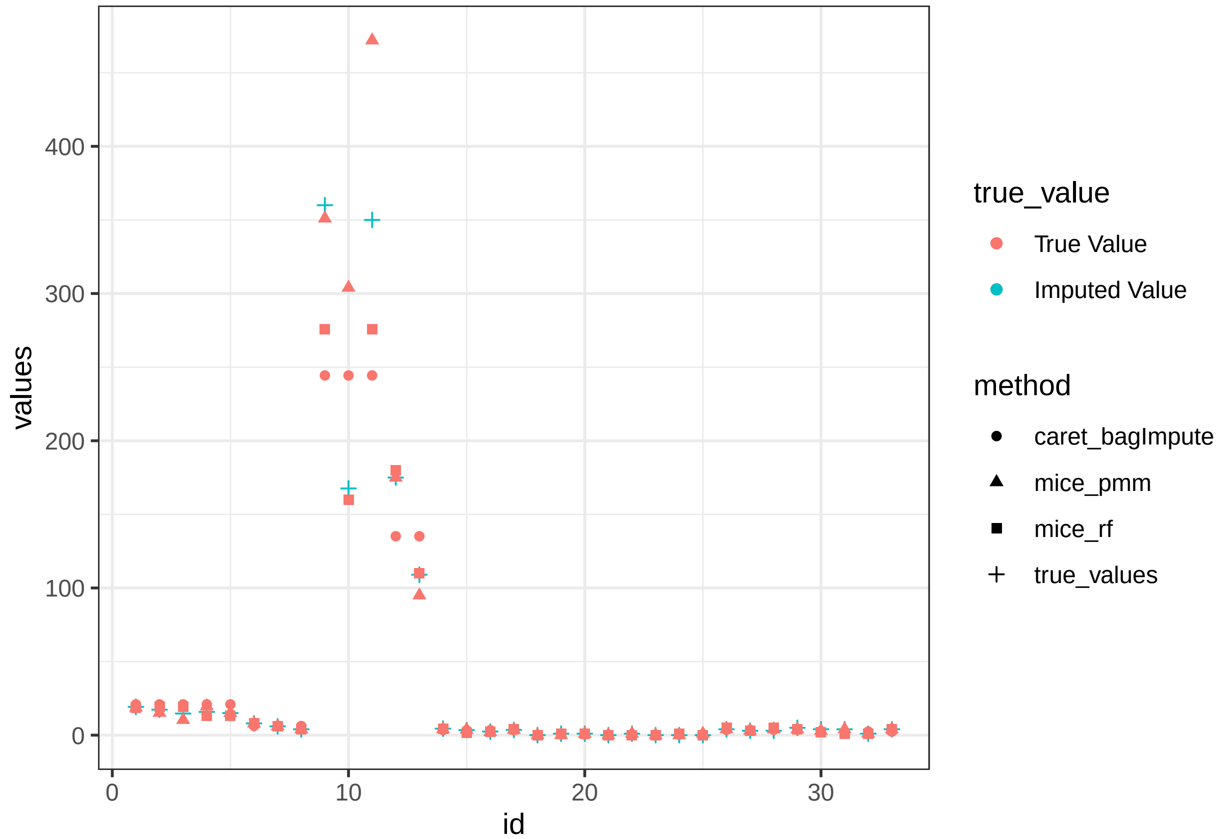
```
df = data.frame(true_values = mtcars[is.na(mtcars_missing)],
           mice_pmm = mtcars_mice[is.na(mtcars_missing)],
           mice_rf = mtcars_mice_rf[is.na(mtcars_missing)],
           caret_bagImpute = mtcars_caret[is.na(mtcars_missing)])
df
```

```
##    true_values mice_pmm mice_rf caret_bagImpute
## 1       19.200    18.70  18.700      20.9530769
## 2       17.300    15.20  19.200      20.9530769
## 3       14.700    10.40  19.200      20.9530769
## 4       15.800    19.70  13.300      20.9530769
## 5       15.000    15.20  13.300      20.9530769
## 6        8.000     8.00   8.000       6.2461538
## 7        6.000     6.00   6.000       6.2461538
## 8        4.000     4.00   4.000       6.2461538
## 9      360.000   351.00 275.800     244.3669231
## 10     167.600   304.00 160.000     244.3669231
## 11     350.000   472.00 275.800     244.3669231
## 12     175.000   175.00 180.000     135.1461538
## 13     109.000    95.00 110.000     135.1461538
## 14       4.430     4.08   4.220       3.4791538
## 15       3.440     3.73   1.513       3.3680077
## 16       2.465     2.14   2.320       3.3680077
## 17       3.570     3.73   3.845       3.3680077
## 18       0.000     0.00   0.000       0.4307692
## 19       1.000     0.00   1.000       0.4307692
## 20       1.000     1.00   1.000       0.4307692
## 21       0.000     0.00   0.000       0.4307692
## 22       1.000     1.00   0.000       0.4307692
## 23       0.000     0.00   0.000       0.3538462
## 24       0.000     0.00   1.000       0.3538462
## 25       0.000     1.00   0.000       0.3538462
## 26       4.000     4.00   5.000       3.6153846
## 27       3.000     3.00   3.000       3.6153846
## 28       3.000     4.00   5.000       3.6153846
## 29       5.000     4.00   4.000       3.6153846
## 30       4.000     3.00   2.000       2.4384615
## 31       4.000     4.00   1.000       2.4384615
## 32       1.000     1.00   1.000       2.4384615
## 33       4.000     4.00   4.000       2.4384615
```

```
library(ggplot2); library(tidyr)
rownames(df)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
## [31] "31" "32" "33"
```

```
df %>%
  mutate(id = as.integer(rownames(df)))%>%
  select(id, everything())%>%
  pivot_longer(2:5,names_to = 'method',values_to ='values')%>%
  mutate(true_value = factor(ifelse(method == 'true_values',1,0),labels = c('True Value','Imputed Value
  ggplot(aes(x = id, y=values, color=true_value, shape = method))+
  geom_point()+theme_bw()
```


values
400
300
200
100
0
0
10
20
30
id
true_value
True Value
Imputed Value
method
caret_bagImpute
mice_pmm
mice_rf
true_values