Missing Data Example with MICE in R

Consequences of missing data

Researchers usually address missing data by including in the analysis only complete cases —those individuals who have no missing data in any of the variables required for that analysis. However, results of such analyses can be biased. Furthermore, the cumulative effect of missing data in several variables often leads to exclusion of a substantial proportion of the original sample, which in turn causes a substantial loss of precision and power.

The risk of bias due to missing data depends on the reasons why data are missing. Reasons for missing data are commonly classified as: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). This nomenclature is widely used, even though the phrases convey little about their technical meaning and practical implications, which can be subtle. When it is plausible that data are missing at random, but not completely at random, analyses based on complete cases may be biased. Such biases can be overcome using methods such as multiple imputation that allow individuals with incomplete data to be included in analyses. Unfortunately, it is not possible to distinguish between missing at random and missing not at random using observed data. Therefore, biases caused by data that are missing not at random can be addressed only by sensitivity analyses examining the effect of different assumptions about the missing data mechanism.

Statistical methods to handle missing data

A variety of ad hoc approaches are commonly used to deal with missing data. These include replacing missing values with values imputed from the observed data (for example, the mean of the observed values), using a missing category indicator, and replacing missing values with the baseline or the last measured value (Baseline observation carried forward & Last observation carried forward). None of these approaches is statistically valid in general, and they can lead to serious bias. Single imputation of missing values usually causes standard errors to be too small, since it fails to account for the fact that we are uncertain about the missing values.

What is multiple imputation?

Multiple imputation is a general approach to the problem of missing data that is available in several commonly used statistical packages. It aims to allow for the uncertainty about the missing data by creating several different plausible imputed data sets and appropriately combining results obtained from each of them.

The first stage is to create multiple copies of the dataset, with the missing values replaced by imputed values. These are sampled from their predictive distribution based on the observed data—thus multiple imputation is based on a bayesian approach. The imputation procedure must fully account for all uncertainty in predicting the missing values by injecting appropriate variability into the multiple imputed values; we can never know the true values of the missing data.

The second stage is to use standard statistical methods to fit the model of interest to each of the imputed datasets. Estimated associations in each of the imputed datasets will differ because of the variation introduced in the imputation of the missing values, and they are only useful when averaged together to give overall estimated associations. Standard errors are calculated using Rubin's rules, which take account of the variability in results between the imputed datasets, reflecting the uncertainty associated with the missing values. Valid inferences are obtained because we are averaging over the distribution of the missing data given the observed data.

Consider, for example, a study investigating the association of systolic blood pressure with the risk of subsequent coronary heart disease, in which data on systolic blood pressure are missing for some people. The probability that systolic blood pressure is missing is likely to decrease with age (doctors are more likely to measure it in older people), increase with body mass index, and history of smoking (doctors are more likely to measure it in people with heart disease risk factors or comorbidities). If we assume that data are missing at random and that we have systolic blood pressure data on a representative sample of individuals within strata of age, smoking, body mass index, and coronary heart disease, then we can use multiple imputation to estimate the overall association between systolic blood pressure and coronary heart disease.

Multiple imputation has potential to improve the validity of medical research. However, the multiple imputation procedure requires the user to model the distribution of each variable with missing values, in terms of the observed data. The validity of results from multiple imputation depends on such modelling being done carefully and appropriately. Multiple imputation should not be regarded as a routine technique to be applied at the push of a button—whenever possible specialist statistical help should be obtained.

Although there are several packages (`mi` developed by Gelman, Hill and others; `hot.deck` by Gill and Cramner, `Amelia` by Honaker, King, Blackwell) in R that can be used for multiple imputation, we will be using the `mice` package, developed by Stef van Buuren.

The basic steps in R are as follows:

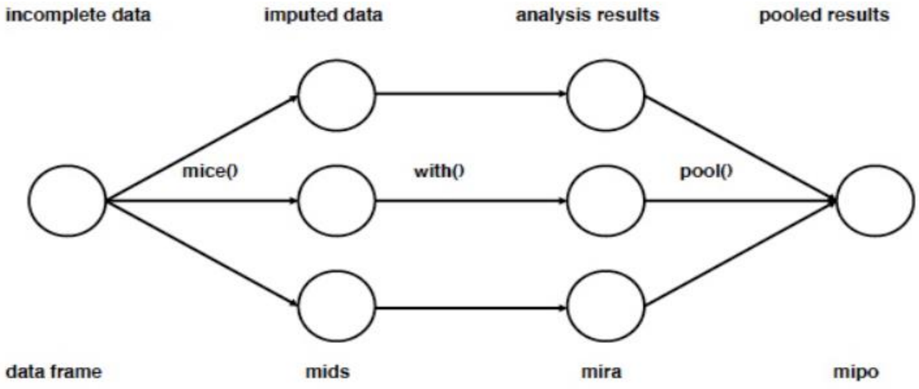
mice() Impute the data. That is, make m copies of the original dataset and fill in the missing values.

with() Analyze each of the completed datasets.

pool() Combine the parameter estimates using Rubin's rules.

The figure below depicts the three main steps to multiple imputation:

Figure 1: Main steps used in multiple imputation



1. Complete Cases analysis

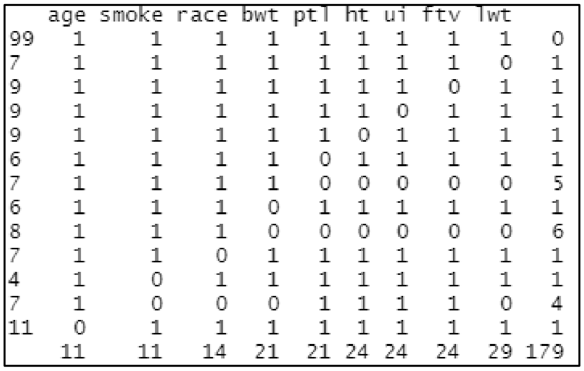
The birthwt dataset has 189 rows and 9 columns. The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

The variables are:

- bwt birth weight in grams.
- age mother's age in years.
- lwt mother's weight in pounds at last menstrual period.
- race mother's race (1 = white, 2 = black, 3 = other).
- smoke smoking status during pregnancy.
- ptl number of previous premature labours.
- ht history of hypertension.
- ui presence of uterine irritability.
- ftv number of physician visits during the first trimester.

Using **md.pattern()** we can examine the pattern of missingness in the data. Each combination of missing variables is given a row in the output, with 1 in the row indicating observed and 0 indicating missing. Here we see that while most of the 189 subjects have complete observations [99, top row], many observations have a single missing variable (only one 0 in the row), and some observations have multiple variables missing (several 0s in the row). Other functions to visualize patterns in missing data are also available from the VIM package.

```
md.pattern(birthwt,plot = FALSE)
```

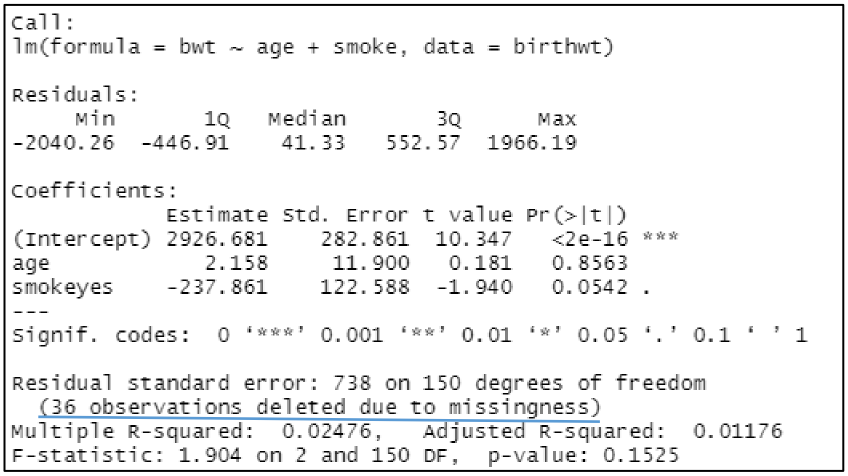


```
   age smoke race bwt ptl ht ui ftv lwt
99   1     1    1   1   1  1  1   1   1   0
7    1     1    1   1   1  1  1   1   0   1
9    1     1    1   1   1  1  1   0   1   1
9    1     1    1   1   1  1  0   1   1   1
9    1     1    1   1   1  0  1   1   1   1
6    1     1    1   1   0  1  1   1   1   1
7    1     1    1   1   0  0  0   0   0   5
6    1     1    1   0   1  1  1   1   1   1
8    1     1    1   0   0  0  0   0   0   6
7    1     1    0   1   1  1  1   1   1   1
4    1     0    1   1   1  1  1   1   1   1
7    1     0    0   0   1  1  1   1   0   4
11   0     1    1   1   1  1  1   1   1   1
    11    11   14  21  21 24 24  24  29 179
```

First, we consider a regression model using only the complete cases: predictors for birthweight using linear regression.

m1.cc <- lm(bwt ~ age + smoke, data = birthwt)

summary(m1.cc)



```
Call:
lm(formula = bwt ~ age + smoke, data = birthwt)

Residuals:
     Min       1Q   Median       3Q      Max 
-2040.26  -446.91    41.33   552.57  1966.19 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2926.681    282.861  10.347   <2e-16 ***
age            2.158     11.900   0.181   0.8563    
smokeyes    -237.861    122.588  -1.940   0.0542 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 738 on 150 degrees of freedom
  (36 observations deleted due to missingness)
Multiple R-squared:  0.02476,	Adjusted R-squared:  0.01176 
F-statistic: 1.904 on 2 and 150 DF,  p-value: 0.1525
```

36 observations deleted. As a result, the analysis was conducted using 189-36=153 observations.

The **coefficients (b)** can be interpreted as:

The related p-values are 0.856 and 0.054 for mother’s age and mother’s smoking status respectively, meaning that the results are non-significant.

2. Multiple Imputation

Now, we use MI to guess what the missing variables "could have been", and then use our imputed datasets to estimate the regression coefficients again.

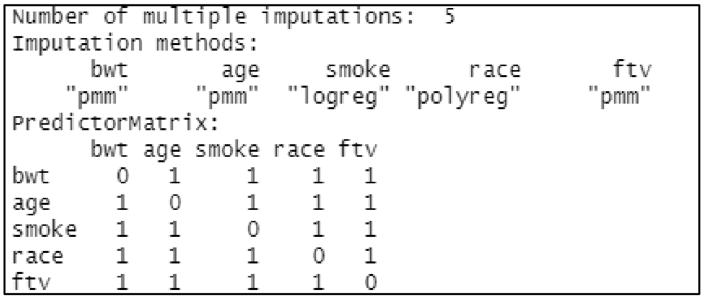
2.a. Impute the missing observations using mice().

Before running mice(), check that your dataset contains only variables you will use in your analysis, or variables you think are related to missingness. The model above considered 3 variables (bwt, age, smoke), and we think that race and ftv may be related to missingness, so we keep only those 5 variables. Then, we use the mice() command to impute the missing variables multiple times, using all other variables (including the outcome) as predictors in the imputation procedure.

```
small <- birthwt[, c("bwt", "age", "smoke", "race", "ftv")]

imp <- mice(small, m = 5, print = FALSE, seed = 12345)

imp
```



```
Number of multiple imputations:  5
Imputation methods:
      bwt       age     smoke      race       ftv
    "pmm"     "pmm"  "logreg" "polyreg"     "pmm"
PredictorMatrix:
      bwt age smoke race ftv
bwt     0   1     1    1   1
age     1   0     1    1   1
smoke   1   1     0    1   1
race    1   1     1    0   1
ftv     1   1     1    1   0
```

Here is an explanation of the parameters used:

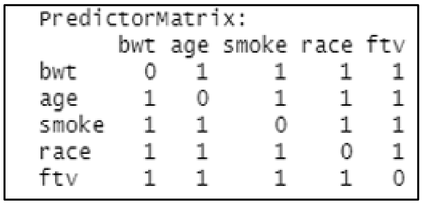
1. m – Refers to 5 imputed data sets
2. maxit – Refers to no. of iterations taken to impute missing values
3. method – Refers to method used in imputation. We used predictive mean matching.

It is recommended
20 imputations for 10% to
30% missing information, and
40 imputations for
50% missing information.

In this example, we see that mice() used pmm (predictive mean matching) to impute all continuous variables except smoke which used logistic regression and race which used polyreg, a form of multinomial logistic regression. (Variables with no missing values will have no method (" ") because no imputation method is needed).

By default, mice() uses the following default methods: predictive mean matching (pmm) for numeric data, logistic regression for factors with 2 levels, and multinomial logistic regression for factors with 3 levels. Note that binary variables that are still coded numerically (0/1, 1/2, etc) will have the default method "pmm", unless you recode it as a factor.

Reading across the rows of the predictor matrix, we can also see that mice() used each of the other variables to impute both variables.



```
 PredictorMatrix:
       bwt age smoke race ftv
 bwt     0   1     1    1   1
 age     1   0     1    1   1
 smoke   1   1     0    1   1
 race    1   1     1    0   1
 ftv     1   1     1    1   0
```

Check imputed values

imp$imp$bwt

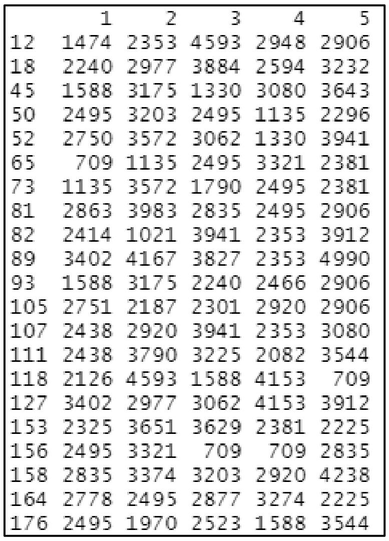
imp$imp$age

imp$imp$smoke

imp$imp$race

imp$imp$ftv

For instance, the imputed values for missing data of bwt variable are:

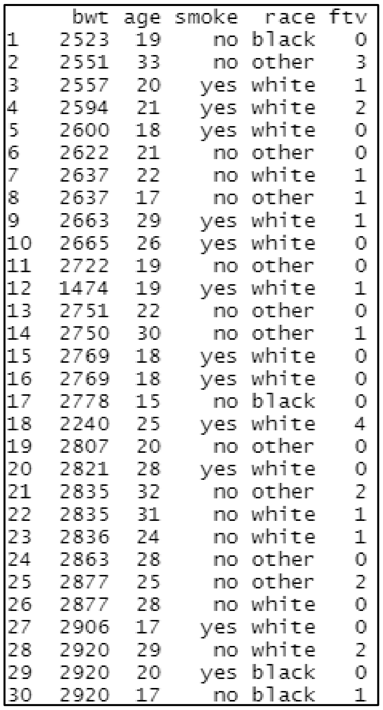


```
       1    2    3    4    5
12  1474 2353 4593 2948 2906
18  2240 2977 3884 2594 3232
45  1588 3175 1330 3080 3643
50  2495 3203 2495 1135 2296
52  2750 3572 3062 1330 3941
65   709 1135 2495 3321 2381
73  1135 3572 1790 2495 2381
81  2863 3983 2835 2495 2906
82  2414 1021 3941 2353 3912
89  3402 4167 3827 2353 4990
93  1588 3175 2240 2466 2906
105 2751 2187 2301 2920 2906
107 2438 2920 3941 2353 3080
111 2438 3790 3225 2082 3544
118 2126 4593 1588 4153  709
127 3402 2977 3062 4153 3912
153 2325 3651 3629 2381 2225
156 2495 3321  709  709 2835
158 2835 3374 3203 2920 4238
164 2778 2495 2877 3274 2225
176 2495 1970 2523 1588 3544
```

Since there are 5 imputed data sets, you can select any using complete() function.

completeData1 <- complete(imp,1)

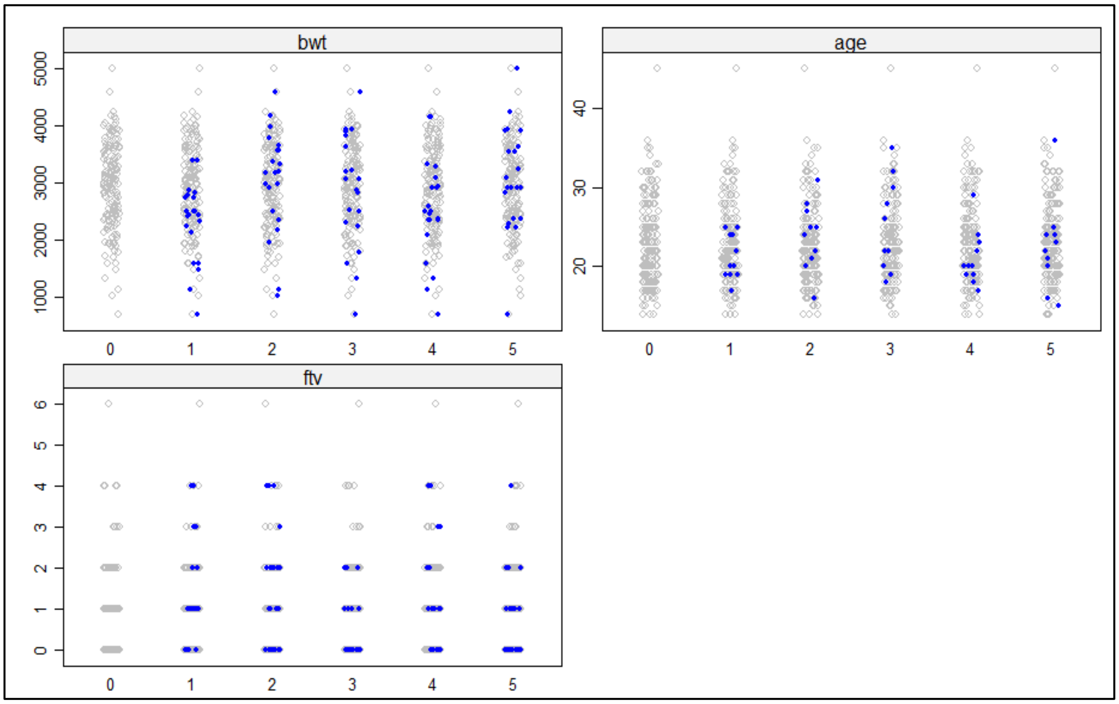
View(completeData1)



```
     bwt age smoke  race ftv
1   2523  19    no black   0
2   2551  33    no other   3
3   2557  20   yes white   1
4   2594  21   yes white   2
5   2600  18   yes white   0
6   2622  21    no other   0
7   2637  22    no white   1
8   2637  17    no other   1
9   2663  29   yes white   1
10  2665  26   yes white   0
11  2722  19    no other   0
12  1474  19   yes white   1
13  2751  22    no other   0
14  2750  30    no other   1
15  2769  18   yes white   0
16  2769  18   yes white   0
17  2778  15    no black   0
18  2240  25   yes white   4
19  2807  20    no other   0
20  2821  28   yes white   0
21  2835  32    no other   2
22  2835  31    no white   1
23  2836  24    no white   1
24  2863  28    no other   0
25  2877  25    no other   2
26  2877  28    no white   0
27  2906  17   yes white   0
28  2920  29    no white   2
29  2920  20   yes black   0
30  2920  17    no black   1
```

After imputation, we can use the stripplot.mids() function to compare the imputed values with observed values for continuous variables. We can see that the imputed values are similar to the observed values, indicating that imputation is probably appropriate for this analysis.

```
stripplot(imp, col=c("grey", "blue"), pch = c(1, 20))
```

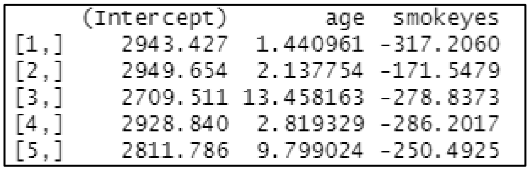


2.b. Run regression model on imputed data using with().

To run a regression model with imputed data, we have to use with(). Note that we use the same model formulation as above, but we leave out the data option. Note that we use the same imputed data to run several models, so there is no need to impute new data for every model of interest. In general we will not look at the results of with() directly, but instead pool() them first. We can however take a look at the analyses and get the results of each of the m fitted regression models.

```
m1.mi <- with(imp, lm(bwt ~ age + smoke))
```

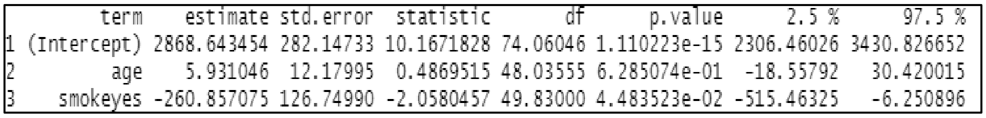
```
t(sapply(m1.mi$analyses, coef))
```



```
     (Intercept)       age  smokeyes
[1,]    2943.427  1.440961 -317.2060
[2,]    2949.654  2.137754 -171.5479
[3,]    2709.511 13.458163 -278.8373
[4,]    2928.840  2.819329 -286.2017
[5,]    2811.786  9.799024 -250.4925
```

2.c. Combine the results using pool().

```
summary(pool(m1.mi), conf.int = TRUE)
```



```
         term    estimate std.error  statistic       df      p.value      2.5 %      97.5 %
1 (Intercept) 2868.643454 282.14733 10.1671828 74.06046 1.110223e-15 2306.46026 3430.826652
2         age    5.931046  12.17995  0.4869515 48.03555 6.285074e-01  -18.55792   30.420015
3    smokeyes -260.857075 126.74990 -2.0580457 49.83000 4.483523e-02 -515.46325   -6.250896
```

The **coefficients (b)** can be interpreted as:

- The p-value of mother’s age is 0.629, meaning that the result is non-significant.
- The mean birth weight of an infant whose mother smokes is significantly lower (on average) about 260 gr relative to an infant whose mother does not smoke (p=0.045) adjusted (or controlling) for mother’s age.

Extended example-Logistic regression model

Now let’s suppose that we want to categorize the numeric variable bwt to infants with born weighting less than 2500 grams (<2500 gr) and infants with born weighting more than 2500 grams (≥2500 gr). Outcome variable (bwt_cat) will take value 2 if an infant was not born with a low birth weight and the value 1 if an infant was born with a low birth weight.

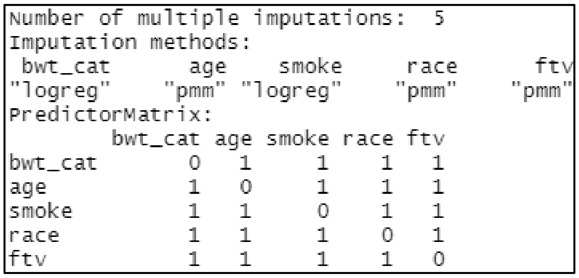
```
birthwt$bwt_cat <- cut(birthwt$bwt,
                    breaks=c(-Inf, 2500, Inf),
                    levels=c(1,2),
                    labels=c("2500 gr or less","more than 2500 gr"))
```

Impute the missing observations using mice().

```
small_2 <- birthwt[, c("bwt_cat", "age", "smoke", "race", "ftv")]

imp_2 <- mice(small_2, m = 5, print = FALSE, seed = 12345)

imp_2
```

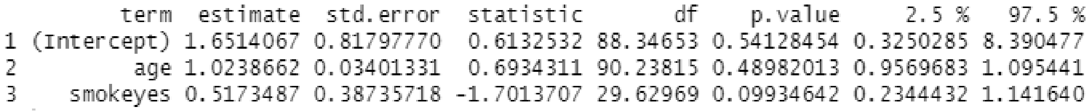


```
Number of multiple imputations:  5
Imputation methods:
 bwt_cat      age    smoke     race      ftv
"logreg"    "pmm" "logreg"    "pmm"    "pmm"
PredictorMatrix:
        bwt_cat age smoke race ftv
bwt_cat       0   1     1    1   1
age           1   0     1    1   1
smoke         1   1     0    1   1
race          1   1     1    0   1
ftv           1   1     1    1   0
```

Combine the results using pool().

```
m2.mi <- with(imp_2, glm(bwt_cat ~ age + smoke,family = binomial()))

summary(pool(m2.mi), exponentiate = TRUE, conf.int = TRUE)
```



```
         term  estimate  std.error  statistic       df    p.value     2.5 %   97.5 %
1 (Intercept) 1.6514067 0.81797770  0.6132532 88.34653 0.54128454 0.3250285 8.390477
2         age 1.0238662 0.03401331  0.6934311 90.23815 0.48982013 0.9569683 1.095441
3    smokeyes 0.5173487 0.38735718 -1.7013707 29.62969 0.09934642 0.2344432 1.141640
```

The **coefficients (b)** can be interpreted as:

- The p-value of mother's age is 0.490, meaning that the result is non-significant.
- Smoking during pregnancy can decrease the odds of normal birth weight by (1-0.52) =48% adjusted (or controlling) for mother's age.

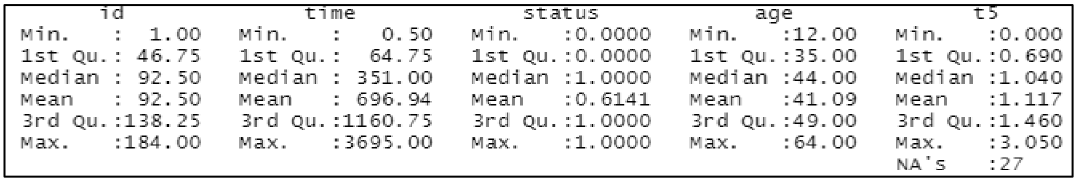
3. Cox proportional hazards regression

The Stanford 2 dataset from the R package survival includes five variables. The data were collected from patients on the waiting list for the Stanford heart transplant program.

The variables are:

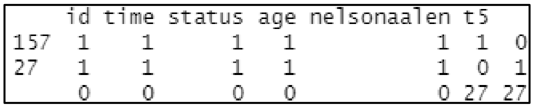
- ID ID number.
- time survival or censoring time.
- status censoring status.
- age patients' age in years.
- t5 T5 mismatch score.

summary(stanford2)

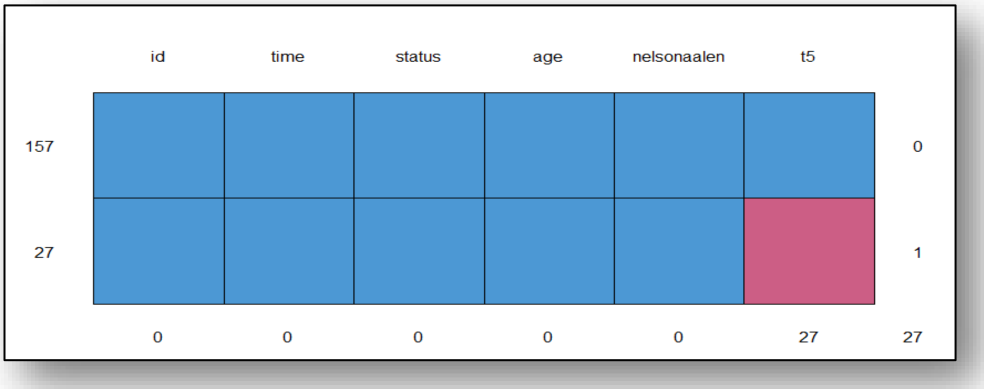


```
       id              time              status            age              t5       
 Min.   :  1.00   Min.   :   0.50   Min.   :0.0000   Min.   :12.00   Min.   :0.000  
 1st Qu.: 46.75   1st Qu.:  64.75   1st Qu.:0.0000   1st Qu.:35.00   1st Qu.:0.690  
 Median : 92.50   Median : 351.00   Median :1.0000   Median :44.00   Median :1.040  
 Mean   : 92.50   Mean   : 696.94   Mean   :0.6141   Mean   :41.09   Mean   :1.117  
 3rd Qu.:138.25   3rd Qu.:1160.75   3rd Qu.:1.0000   3rd Qu.:49.00   3rd Qu.:1.460  
 Max.   :184.00   Max.   :3695.00   Max.   :1.0000   Max.   :64.00   Max.   :3.050  
                                                                     NA's   :27     
```

md.pattern(stanford2, plot = TRUE)



```
    id time status age nelsonaalen t5   
157  1    1      1   1           1  1  0
27   1    1      1   1           1  0  1
     0    0      0   0           0 27 27
```



There are 157 observations with no missing values and 27 observations with missing values in the variable of t5.

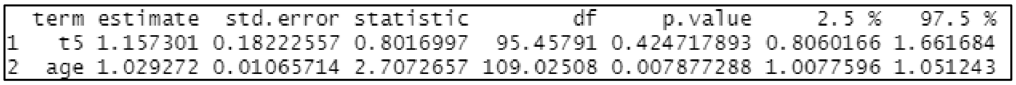
It is recommended to include two variables related to the survival endpoint in the imputation models, the Nelson-Aalen estimate of the cumulative hazard (nelsonaalen()) and the event indicator, in the imputation process.

```
stanford2$nelsonaalen <- nelsonaalen(stanford2, time, status)

imp.surv <- mice(stanford2[,c("time","status","age","t5","nelsonaalen")], m = 20, print =
FALSE)m2.mi <- with(imp.surv, coxph(Surv(time, status) ~ t5 + age))

summary(pool(m2.mi), conf.int = TRUE, exponentiate = TRUE)
```

Since we used the exponentiate = TRUE option, the colum labelled estimate shows the hazard ratios



```
  term estimate  std.error statistic        df     p.value     2.5 %   97.5 %
1   t5 1.157301 0.18222557 0.8016997  95.45791 0.424717893 0.8060166 1.661684
2  age 1.029272 0.01065714 2.7072657 109.02508 0.007877288 1.0077596 1.051243
```

The **coefficients (b)** can be interpreted as:

- The p-value of t5 is 0.425, meaning that the result is non-significant.
- The risk of death is increased by about (1.03-1=0.03) 3% as the patient's age increases by one year adjusted (or controlling) for t5