Missing Data: Tutorial 2

Albert Varela & José Pina-Sánchez

Table of contents

Introduction 1
The problem 2
Imputation 8
Single imputation 8
Multiple imputation 14
Non-ignorability and sensitivity analysis 18
Conclusions 22

Introduction

In this tutorial we will go through an example of missing data imputation using *mice* , probably the most popular R package for multiple imputation.

The substantive motivation for this analysis is to assess the relationship between earnings and wanting to increase the number of working hours using a subset of variables from the quarterly LFS collected between January and March 2018.

To help us do this, we will be using the following R packages:

```
library(tidyverse)
library(haven)
library(psych)
library(mice)
library(ggmice)
library(janitor)
set.seed(123)
```

And then we load the data and select relevant variables

```
lfs <-
  read_dta("data/lfsp_jm18_eul.dta") %>%
  clean_names() %>%
  select(grsswk, undhrs, sex, tothrs,
         age, hiqul15d , marsta,
         nsecmj10, empmon, in0792em, publicr)
```

The problem

We consider some form of imputation because the income variable *grsswk* has a considerable amount of missing values. Notice that the minimum value for that variable is -9, which is an implausible value for incomes.

```
describe(lfs$grsswk)
```

```
   vars     n mean     sd median trimmed mad min  max range skew kurtosis   se
X1    1 89470   52 217.67     -9      52   0  -9 9923  9932  6.1    86.19 0.73
```

Indeed, a quick glance at the codebook or using the get_labels() function will reveal that there are two negative values in this and other variables in the dataset which are indicative of missingness:

- -9: which indicates that the question is *inapplicable* - a form of intentional missing data resulting from questionnaire design
- -8: which indicates *non-response*.

Since we only want to work with cases for which income is a plausible value, we will filter out all cases that have a value of -9 for *inapplicable* over the variables grsswk, undhrs and tothrs. If we did not do this, we'd be imputing values on empty cells for which income values are not appropriate.

```
lfs <-
  lfs %>%
  filter(grsswk > -9 & undhrs > -9 & tothrs > -9)
```

And then we code as missing values all of the non-response currently indicated by -8 across all variables:

```
lfs$grsswk[lfs$grsswk == -8] <- NA
lfs$undhrs[lfs$undhrs == -8] <- NA
```

```r
lfs$tothrs[lfs$tothrs == -8] <- NA
lfs$hiqul15d[lfs$hiqul15d == -8 |
               lfs$hiqul15d == 7] <- NA
lfs$empmon[lfs$empmon == -8] <- NA
lfs$in0792em[lfs$in0792em == -8] <- NA
lfs$nsecmj10[lfs$nsecmj10 == 4] <- NA
lfs$publicr[lfs$publicr == -8] <- NA
```

The imputation functions in *mice* auto-detect the types of variables that are fed into the imputation model in order to automatically specify the correct imputation model for each variable. In particular, it does not interact nicely with labelled data (as haven imports categorical variables) nor does it like categorical variables that contain empty levels. We will do some changes here:

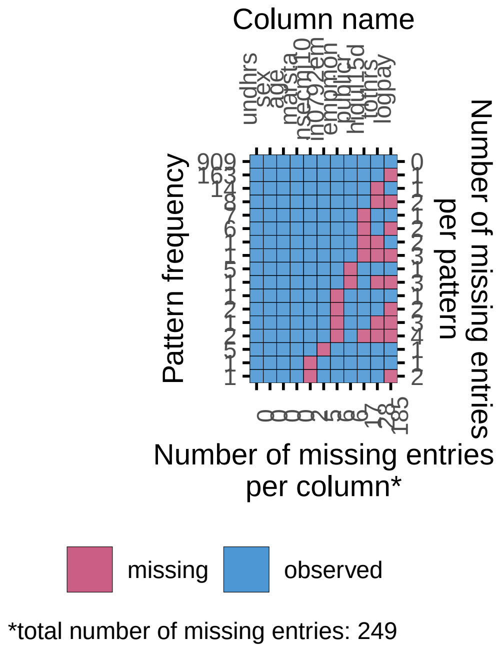
```r
lfs$grsswk <- as.numeric(lfs$grsswk)
lfs$undhrs <- as.numeric(lfs$undhrs)
lfs$tothrs <- as.numeric(lfs$tothrs)
lfs$empmon <- as.numeric(lfs$empmon)
lfs$hiqul15d <- droplevels(as.factor(lfs$hiqul15d))
lfs$nsecmj10 <- droplevels(as.factor(lfs$nsecmj10))
lfs$in0792em <- droplevels(as.factor(lfs$in0792em))
lfs$publicr <- droplevels(as.factor(lfs$publicr))
```

We will also take the logarithm of pay per week to normalise it, though this is strictly not necessary:

```r
lfs$logpay <- log(lfs$grsswk)
```

So, after all the data clean-up, now let's examine the missingness pattern in the data:

```r
lfs %>%
  select(-grsswk) %>%
  plot_pattern(rotate = TRUE)
```



There are are a total of 249 missing values in the dataset, driven largely by the 185 observations missing due to non-response in the income variable - this is an item-specific non-response rate of roughly 16%

```
round(prop.table(table(is.na(lfs$grsswk))),3)
```

```
FALSE  TRUE
0.836 0.164
```

The rest of variables have much more smaller non-response rates - some including the DV have no missing values. Nonetheless, consider the consequences of including income as a predictor in a regression model:

```
# simple regression
fit_cca0 <- lm(undhrs ~ logpay, data = lfs)
summary(fit_cca0)
```

```
Call:
lm(formula = undhrs ~ logpay, data = lfs)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-18.583  -6.197  -3.002   1.710  89.879

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  27.1971     3.1069   8.754  < 2e-16 ***
logpay       -2.7001     0.5743  -4.701 2.97e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.29 on 941 degrees of freedom
  (185 observations deleted due to missingness)
Multiple R-squared:  0.02295,	Adjusted R-squared:  0.02191 
F-statistic:  22.1 on 1 and 941 DF,  p-value: 2.968e-06
```

```r
# multiple regression
fit_cca1 <- lm(undhrs ~ logpay + age + sex + empmon +
                hiqul15d + tothrs + nsecmj10 , data = lfs)
summary(fit_cca1)
```

```
Call:
lm(formula = undhrs ~ logpay + age + sex + empmon + hiqul15d + 
    tothrs + nsecmj10, data = lfs)

Residuals:
    Min      1Q  Median      3Q     Max 
-17.654  -5.993  -2.502   1.965  89.267 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 34.392526   5.562878   6.183 9.55e-10 ***
logpay      -2.085131   0.816834  -2.553  0.01085 *  
age         -0.082095   0.040392  -2.032  0.04240 *  
sex         -1.645012   1.003147  -1.640  0.10138    
empmon      -0.005643   0.006512  -0.866  0.38646    
hiqul15d2    1.218564   1.706422   0.714  0.47535    
hiqul15d3   -1.937807   1.367809  -1.417  0.15691    
hiqul15d4   -0.793578   1.378701  -0.576  0.56503    
hiqul15d5   -0.502275   1.765016  -0.285  0.77604    
hiqul15d6    1.672801   2.232015   0.749  0.45378    
tothrs      -0.118817   0.038746  -3.067  0.00223 ** 
nsecmj102   -0.387951   2.177220  -0.178  0.85862    
```

```
nsecmj103   -2.826218   2.287765  -1.235  0.21702
nsecmj105    0.414183   2.538212   0.163  0.87041
nsecmj106   -0.463689   2.266064  -0.205  0.83791
nsecmj107   -1.088831   2.406062  -0.453  0.65099
nsecmj108   -5.335106   3.008980  -1.773  0.07656 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.45 on 902 degrees of freedom
  (209 observations deleted due to missingness)
Multiple R-squared:  0.05847,   Adjusted R-squared:  0.04177
F-statistic: 3.501 on 16 and 902 DF,  p-value: 3.893e-06
```

As a result of the *lm()* function defaulting on complete case analysis or listwise deletion we are dropping 209 out of 1129 cases from our analysis (nearly 19% of our sample) due to the combination of missing data patterns across the variables.

```
sjPlot::tab_model(fit_cca0, fit_cca1)
```

	undhrs			undhrs		
Predictors	Estimates	CI	p	Estimates	CI	p
(Intercept)	27.20	21.10 – 33.29	**<0.001**	34.39	23.47 – 45.31	**<0.001**
logpay	-2.70	-3.83 – -1.57	**<0.001**	-2.09	-3.69 – -0.48	**0.011**
Age of respondent				-0.08	-0.16 – -0.00	**0.042**
Sex of respondent				-1.65	-3.61 – 0.32	0.101
empmon				-0.01	-0.02 – 0.01	0.386
hiqul 15 d: hiqul 15 d 2				1.22	-2.13 – 4.57	0.475
hiqul 15 d: hiqul 15 d 3				-1.94	-4.62 – 0.75	0.157
hiqul 15 d: hiqul 15 d 4				-0.79	-3.50 – 1.91	0.565
hiqul 15 d: hiqul 15 d 5				-0.50	-3.97 – 2.96	0.776
hiqul 15 d: hiqul 15 d 6				1.67	-2.71 – 6.05	0.454
tothrs				-0.12	-0.19 – -0.04	**0.002**
nsecmj 10: nsecmj 102				-0.39	-4.66 – 3.89	0.859
nsecmj 10: nsecmj 103				-2.83	-7.32 – 1.66	0.217
nsecmj 10: nsecmj 105				0.41	-4.57 – 5.40	0.870
nsecmj 10: nsecmj 106				-0.46	-4.91 – 3.98	0.838
nsecmj 10: nsecmj 107				-1.09	-5.81 – 3.63	0.651
nsecmj 10: nsecmj 108				-5.34	-11.24 – 0.57	0.077
Observations	943			919		
R^2 / R^2 adjusted	0.023 / 0.022			0.058 / 0.042		

In the regression table the coefficients for pay in both the model with controls and the model without controls are negative and statistically significant - i.e. workers with higher pay prefer smaller increments in working hours than those in lower pay, all else being equal. Notice, however, how close to 0 the upper bound of the confidence interval is in the model with controls.

We first create define two different dataframes, one for the simple imputation that contains either complete variables, or incomplete variables that are continuous (this is because we will use the mice function for ad hoc imputations); and a second one for the multiple imputation procedure, that also contains categorical/factor variables:

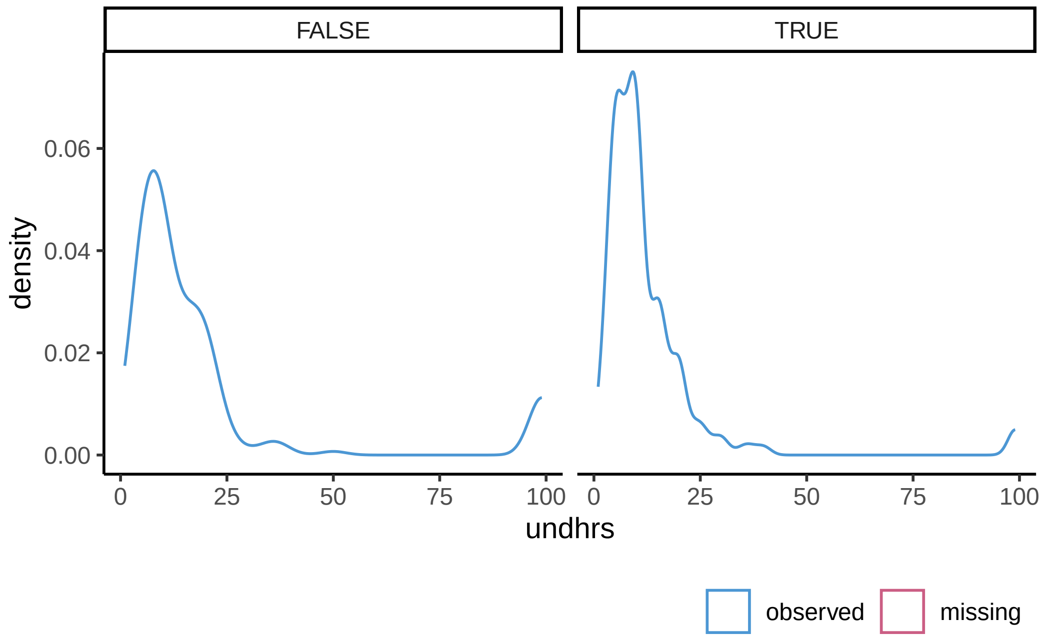
```
# Define a dataset with predictors

lfs_imp_c <-
  lfs %>%
  select(-c(grsswk, hiqul15d, nsecmj10, in0792em,
                publicr))
lfs_imp <-
  lfs %>%
  select(-grsswk)
```

Bear in mind that there is a difference in the number of preferred additional hours between those that reported and those that did not report their income:

```
lfs %>%
  group_by(is.na(grsswk)) %>%
  summarise(mean_undhrs = mean(undhrs, na.rm = TRUE))
```

```
# A tibble: 2 x 2
  `is.na(grsswk)` mean_undhrs
  <lgl>                 <dbl>
1 FALSE                  12.8
2 TRUE                   19.6
```

```
ggmice(lfs_imp, aes(undhrs)) +
  geom_density() +
  facet_wrap(~is.na(logpay) == 0)
```



It is worth examining whether our results so far are robust to adjustment for missing data.

Imputation

Given that we have a a clear target variable for imputation we can focus our attention on filling in the missing values for income. We will proceed with different forms of imputation in increasing order of sophistication to show the pros and cons of each.

Single imputation

Mean imputation

We will begin by replacing the empty cells in variable income with its mean.

```
mean(lfs_imp$logpay, na.rm = TRUE) # just to check
```

```
[1] 5.348913
```

```
mean_imp <- mice(lfs_imp_c, method = "mean", m = 1, maxit = 1)
```

```
 iter imp variable
  1   1  tothrs  empmon  logpay
```

```r
mean_imp$imp$logpay[1:15,]
```

```
 [1] 5.348913 5.348913 5.348913 5.348913 5.348913 5.348913 5.348913 5.348913
 [9] 5.348913 5.348913 5.348913 5.348913 5.348913 5.348913 5.348913
```

```r
describe(complete(mean_imp))
```

```
       vars    n  mean    sd median trimmed   mad   min    max  range  skew
undhrs    1 1128 13.88 16.96  10.00   10.51  7.41  1.00  99.00  98.00  4.05
sex       2 1128  1.60  0.49   2.00    1.62  0.00  1.00   2.00   1.00 -0.40
tothrs    3 1128 23.79 14.32  23.79   23.62 16.00  0.00  70.00  70.00  0.24
age       4 1128 38.21 12.93  38.00   37.96 14.83 16.00  73.00  57.00  0.14
marsta    5 1128  1.77  0.96   2.00    1.57  1.48  1.00   6.00   5.00  1.56
empmon    6 1128 62.40 76.82  30.00   46.90 35.58  0.00 420.00 420.00  1.90
logpay    7 1128  5.35  0.74   5.35    5.38  0.61  2.08   7.56   5.48 -0.44
       kurtosis   se
undhrs    17.36 0.51
sex       -1.84 0.01
tothrs    -0.21 0.43
age       -1.00 0.38
marsta     2.43 0.03
empmon     3.55 2.29
logpay     1.53 0.02
```

```r
fit_mean <- with(mean_imp, lm(undhrs ~ logpay))
summary(fit_mean)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    28.3      3.66       7.74 2.14e-14  1128
2 logpay         -2.70     0.677     -3.99 7.13e- 5  1128
```

Regression imputation

Then replace the missing values with regression predictions

```r
reg_imp <- mice(lfs_imp_c, method = "norm", m = 1, maxit = 1)
```

```
 iter imp variable
  1   1  tothrs  empmon  logpay
```

```r
reg_imp$imp$logpay[1:15,]
```

```
 [1] 5.126245 5.960913 3.719612 6.152339 5.278468 6.225238 4.114142 5.636660
 [9] 5.372343 4.702004 5.658449 4.713403 4.761733 5.810334 5.029644
```

```r
describe(complete(reg_imp))
```

```
       vars    n  mean    sd median trimmed   mad    min    max  range  skew
undhrs    1 1128 13.88 16.96  10.00   10.51  7.41   1.00  99.00  98.00  4.05
sex       2 1128  1.60  0.49   2.00    1.62  0.00   1.00   2.00   1.00 -0.40
tothrs    3 1128 23.71 14.57  24.00   23.53 16.31 -13.93  70.00  83.93  0.22
age       4 1128 38.21 12.93  38.00   37.96 14.83  16.00  73.00  57.00  0.14
marsta    5 1128  1.77  0.96   2.00    1.57  1.48   1.00   6.00   5.00  1.56
empmon    6 1128 62.30 77.05  30.00   46.94 35.58 -82.79 420.00 502.79  1.88
logpay    7 1128  5.34  0.80   5.39    5.37  0.77   2.08   7.56   5.48 -0.36
       kurtosis   se
undhrs    17.36 0.51
sex       -1.84 0.01
tothrs    -0.28 0.43
age       -1.00 0.38
marsta     2.43 0.03
empmon     3.49 2.29
logpay     0.62 0.02
```

```r
fit_reg <- with(reg_imp, lm(undhrs ~ logpay))
summary(fit_reg)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    29.4      3.36       8.74 8.39e-18  1128
2 logpay         -2.90     0.622     -4.66 3.49e- 6  1128
```

Regression imputation + noise

Then replace the missing values with regression predictions with added random noise to simulate uncertainty

```
# regression + noise imputation
reg_noise_imp <- mice(lfs_imp_c, method = "norm.nob", m = 1, maxit = 1)
```

```
 iter imp variable
  1   1  tothrs  empmon  logpay
```

```
reg_noise_imp$imp$logpay[1:15,]
```

```
 [1] 4.350906 4.806584 4.414024 5.885842 5.385762 4.467012 4.041711 5.948169
 [9] 5.245758 4.167390 4.002967 6.890025 5.068253 5.165777 5.053588
```

```
describe(complete(reg_noise_imp))
```

```
       vars    n  mean    sd median trimmed   mad    min    max  range  skew
undhrs    1 1128 13.88 16.96  10.00   10.51  7.41   1.00  99.00  98.00  4.05
sex       2 1128  1.60  0.49   2.00    1.62  0.00   1.00   2.00   1.00 -0.40
tothrs    3 1128 23.73 14.55  24.00   23.55 16.31 -13.83  70.00  83.83  0.21
age       4 1128 38.21 12.93  38.00   37.96 14.83  16.00  73.00  57.00  0.14
marsta    5 1128  1.77  0.96   2.00    1.57  1.48   1.00   6.00   5.00  1.56
empmon    6 1128 62.13 76.98  30.00   46.66 35.58 -55.19 420.00 475.19  1.89
logpay    7 1128  5.34  0.82   5.37    5.37  0.77   2.08   7.61   5.53 -0.32
       kurtosis   se
undhrs    17.36 0.51
sex       -1.84 0.01
tothrs    -0.29 0.43
age       -1.00 0.38
marsta     2.43 0.03
empmon     3.53 2.29
logpay     0.55 0.02
```

```
fit_reg_noise <- with(reg_noise_imp, lm(undhrs ~ logpay))
summary(fit_reg_noise)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    30.8      3.30       9.33 5.47e-20  1128
2 logpay         -3.17     0.611     -5.18 2.57e- 7  1128
```

Regression imputation + boostrap

Then replace the missing values with regression predictions with bootstrap to simulate uncertainty

```r
reg_boot_imp <- mice(lfs_imp_c, method = "norm.boot", m = 1, maxit = 1)
```

```
 iter imp variable
  1   1  tothrs  empmon  logpay
```

```r
reg_boot_imp$imp$logpay[1:15,]
```

```
 [1] 5.091170 3.910547 5.178577 5.953296 4.906156 5.782508 4.447163 5.079592
 [9] 3.024257 3.892219 5.510044 6.014886 5.914248 4.862515 4.971872
```

```r
describe(complete(reg_boot_imp))
```

```
       vars    n  mean    sd median trimmed   mad    min    max  range  skew
undhrs    1 1128 13.88 16.96  10.00   10.51  7.41   1.00  99.00  98.00  4.05
sex       2 1128  1.60  0.49   2.00    1.62  0.00   1.00   2.00   1.00 -0.40
tothrs    3 1128 23.89 14.54  24.00   23.68 16.31  -4.95  73.38  78.33  0.25
age       4 1128 38.21 12.93  38.00   37.96 14.83  16.00  73.00  57.00  0.14
marsta    5 1128  1.77  0.96   2.00    1.57  1.48   1.00   6.00   5.00  1.56
empmon    6 1128 62.49 77.10  30.00   47.12 35.58 -64.90 420.00 484.90  1.87
logpay    7 1128  5.35  0.81   5.39    5.37  0.76   2.08   7.58   5.50 -0.35
       kurtosis   se
undhrs    17.36 0.51
sex       -1.84 0.01
tothrs    -0.22 0.43
age       -1.00 0.38
marsta     2.43 0.03
empmon     3.46 2.30
logpay     0.72 0.02
```

```r
fit_reg_boot <- with(reg_boot_imp, lm(undhrs ~ logpay))
summary(fit_reg_boot)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    26.1      3.36       7.75 2.01e-14  1128
2 logpay         -2.28     0.622     -3.67 2.57e- 4  1128
```

Finally let us look at all the regression coefficients for logpay over different imputations:

```r
# compare coefficients
summary(fit_mean)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    28.3      3.66       7.74 2.14e-14  1128
2 logpay         -2.70     0.677     -3.99 7.13e- 5  1128
```

```r
summary(fit_reg)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    29.4      3.36       8.74 8.39e-18  1128
2 logpay         -2.90     0.622     -4.66 3.49e- 6  1128
```

```r
summary(fit_reg_noise)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    30.8      3.30       9.33 5.47e-20  1128
2 logpay         -3.17     0.611     -5.18 2.57e- 7  1128
```

```r
summary(fit_reg_boot)
```

```
# A tibble: 2 x 6
  term        estimate std.error statistic  p.value  nobs
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
1 (Intercept)    26.1      3.36       7.75 2.01e-14  1128
2 logpay         -2.28     0.622     -3.67 2.57e- 4  1128
```

Each of these different forms of single imputation reflect better than the previous one some aspect of the missingness mechanism either because:

- it makes better use of the information contained in the observed values in the other variables
- it reflects better some of the uncertainty involved in estimating the imputation values

Single imputation falls short, however, because it cannot reflect the fact that from any of those models, there are more than one plausible predicted values that could have been used for imputation. Using single imputation basically does not take into consideration the uncertainty involved in imputing the missing values.

Multiple imputation

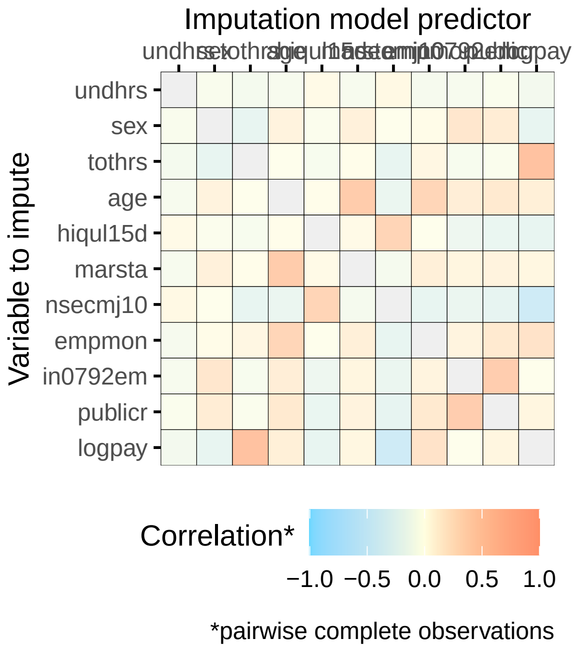
So, we now move to multiple imputation again using the mice package.

It is worth pointing out that in this tutorial we are using a dataset with already pre-selected the variables that will be used for imputation and for the analysis models, but in practice you will need to choose your own set of variables for your projects.

Choosing predictors

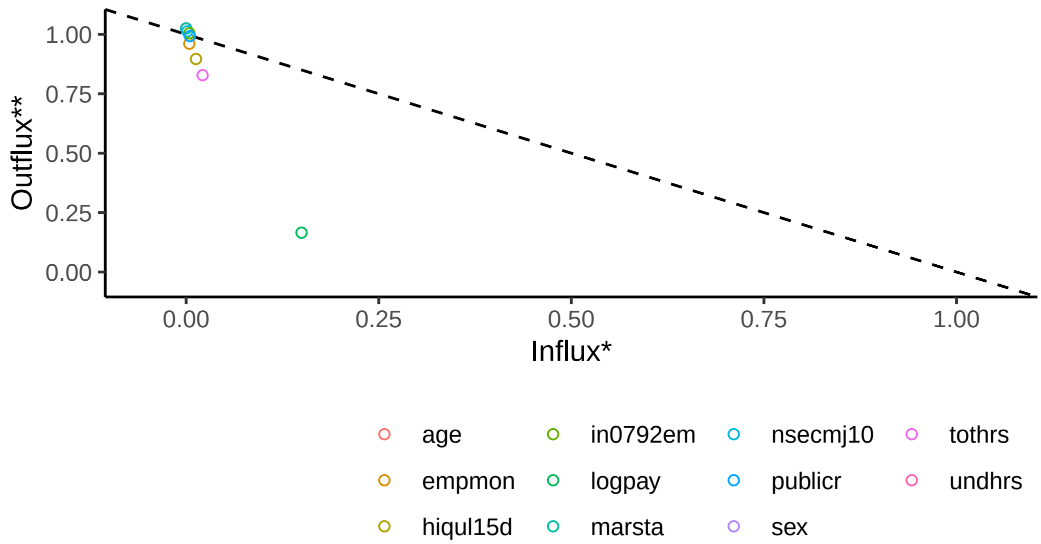
Correlation plots are a good starting point:

```r
plot_corr(lfs_imp)
```



In this sense it can be helpful to examine the outflux-influx plot, which teases out the roles that the different variables will have in providing information to impute other variables' missing values (*outflux*) and in receiving information to have their missing values imputed (*influx*)

```
plot_flux(lfs_imp,
          label = FALSE)
```



*connection of a variable's missingness indicator with observed data on other variables

**connection of a variable's observed data with missing data on other variables

Multiple imputation with mice

We can now proceed with the imputation:

```
lfs_mi <- mice(lfs_imp, m=10, print = FALSE)
```

It is worth examining the actual imputed values to check there are nothing too implausible:

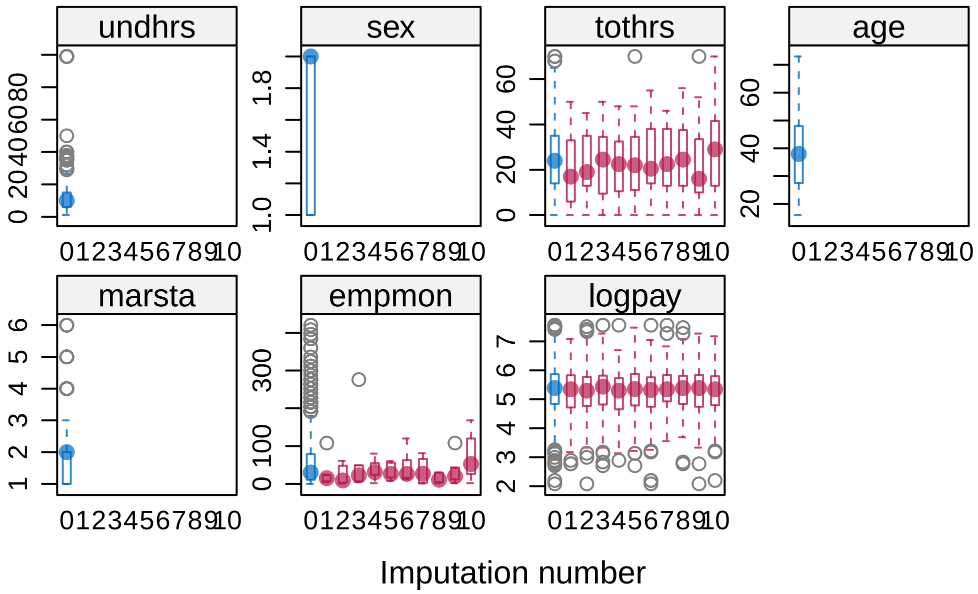
```
lfs_mi$imp$logpay[1:15, 1:5]
```

```
          1        2        3        4        5
1  5.010635 4.564348 4.406719 4.007333 5.214936
3  5.442418 5.442418 5.537334 5.337538 5.298317
7  3.912023 5.389072 3.912023 4.317488 3.135494
22 7.083388 5.010635 5.966147 5.891644 6.028279
25 5.796058 6.028279 4.663439 4.394449 5.087596
31 5.356586 6.175867 5.783825 5.488938 5.777652
38 4.094345 4.709530 4.962845 4.499810 5.521461
47 4.990433 6.802395 5.594711 3.433987 4.682131
57 3.218876 3.555348 2.708050 4.290459 3.218876
61 4.094345 4.454347 4.499810 4.653960 4.605170
71 5.398163 6.184149 6.025866 5.135798 5.442418
```

```
72 5.624018 5.575949 5.347108 5.398163 5.442418
76 5.662960 5.393628 5.192957 4.997212 3.433987
80 5.308268 6.042633 5.814131 4.143135 5.883322
92 4.356709 4.700480 4.356709 3.912023 4.290459
```

There are a number of plots available in mice and ggmice to examine the imputations, let's have a look at a few. To see the distribution of imputed values across each imputation and compare them to the distribution for the complete cases you can use the *bwplot()* function:

```
bwplot(lfs_mi)
```



Ideally there should not be a very large disparity between the imputations and the observed values.

Seeing as there is nothing, we can then run again the regression model, this time with the 10 imputed datasets which will then be pooled into a single set of estimates:

```
# regression with multiply imputed dataset
fit_mi <- with(lfs_mi, lm(undhrs ~ logpay + age + sex + empmon + hiqul15d + tothrs + nsecm
summary(pool(fit_mi))
```

```
          term    estimate   std.error  statistic       df      p.value
1  (Intercept) 41.54513994 6.478572819  6.4126994 449.0711 3.627022e-10
2       logpay -2.29304884 0.948941029 -2.4164292 353.2906 1.617986e-02
```

```
3          age -0.10253795 0.043939243 -2.3336304 1102.8298 1.979404e-02
4          sex -3.58424224 1.115802940 -3.2122538 1012.0775 1.358527e-03
5       empmon -0.01194933 0.007398483 -1.6151053  978.5373 1.066103e-01
6     hiqul15d2 -0.61616025 1.908684452 -0.3228193 1080.5862 7.468945e-01
7     hiqul15d3 -3.21561457 1.548544130 -2.0765405  932.7145 3.811733e-02
8     hiqul15d4 -1.76759483 1.580125596 -1.1186420  960.0804 2.635727e-01
9     hiqul15d5  0.64596231 1.981482324  0.3259995  985.0871 7.444939e-01
10    hiqul15d6  0.52178889 2.511230872  0.2077821  435.7418 8.354962e-01
11       tothrs -0.15711776 0.045333301 -3.4658355  416.2108 5.835413e-04
12    nsecmj102  0.80929949 2.483767564  0.3258354 1104.0465 7.446106e-01
13    nsecmj103 -2.01035309 2.608765422 -0.7706147 1104.3567 4.411001e-01
14    nsecmj105  5.01431909 2.901764073  1.7280244  990.6285 8.429549e-02
15    nsecmj106  0.45431627 2.577914911  0.1762340 1099.1547 8.601426e-01
16    nsecmj107 -0.31368786 2.744358384 -0.1143028 1099.4064 9.090186e-01
17    nsecmj108 -6.76273136 3.374410639 -2.0041222  947.6850 4.534115e-02
```

We can see that the coefficient for logpay remains negative and significant when using the multiply imputed data. At least we can have some reassurance that, if the missing values in our data are indeed MAR and that our imputation and analytical model are correctly specified, our substantive findings are defensible.

Non-ignorability and sensitivity analysis

There is some evidence in the literature that income non-response is particularly prevalent among those survey participants who have higher incomes. This would constitute a case of MNAR because the data predicting the missingness mechanism would itself be missing, so even if we did a fully conditional imputation, as we just did, we cannot assume with any certainty that the respondents and the non-respondents are equal.

Given this, there are few options, normally involving auxiliary data or a selection model. These are however complex techniques and may not be a convenient tool for a researcher that does not have a substantive interest in the selection or MNAR model itself, but rather wants to check whether the possibility of MNAR could be a potential threat to their findings.

A relatively easy way of achieving this is by conducting sensitivity analysis on the imputed values.

```
ini <- mice(lfs_imp, maxit = 0)
ini$nmis
```

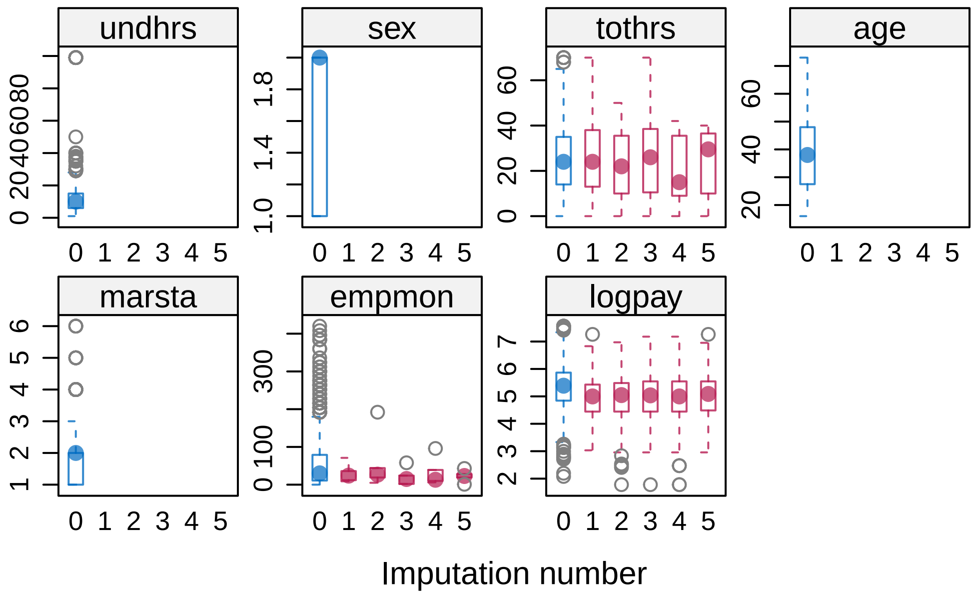
```
  undhrs       sex    tothrs       age  hiqul15d    marsta  nsecmj10    empmon 
       0         0        28         0        17         0         2         6 
```

```
in0792em  publicr    logpay
       5        6       185
```

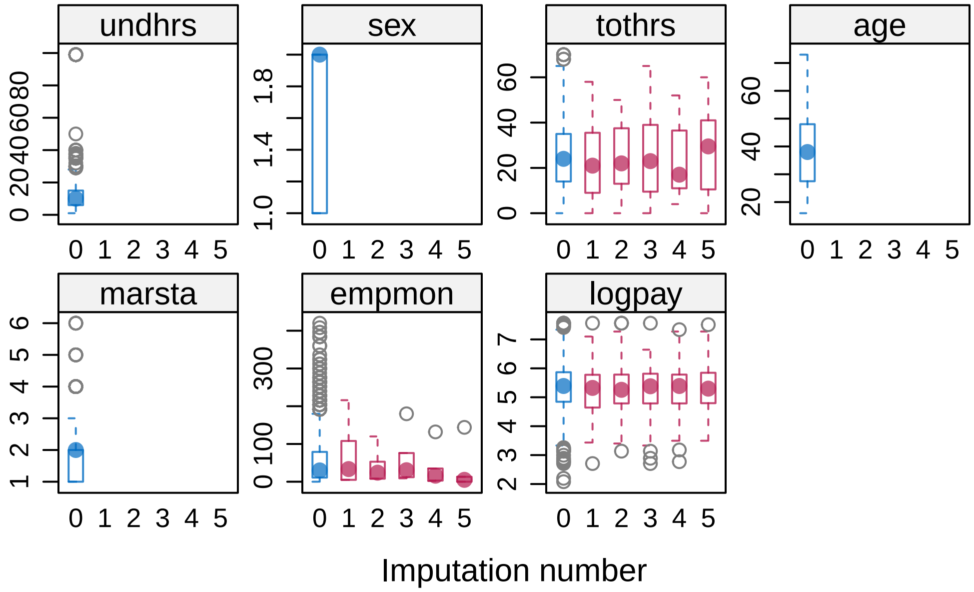
```
delta <- c(-.3, -.1, 0, .1, .3)

imp.all <- vector("list", length(delta))
post <- ini$post

for (i in 1:length(delta)){
  d <- delta[i]
  cmd <- paste("imp[[j]][,i] <- imp[[j]][,i] +", d)
  post["logpay"] <- cmd
  imp <- mice(lfs_imp, post = post, maxit = 10, seed = i, print = FALSE)
  imp.all[[i]] <- imp
}
```

Notice how the imputations are slightly higher or lower than than the observed values as a function of our delta values specified earlier:

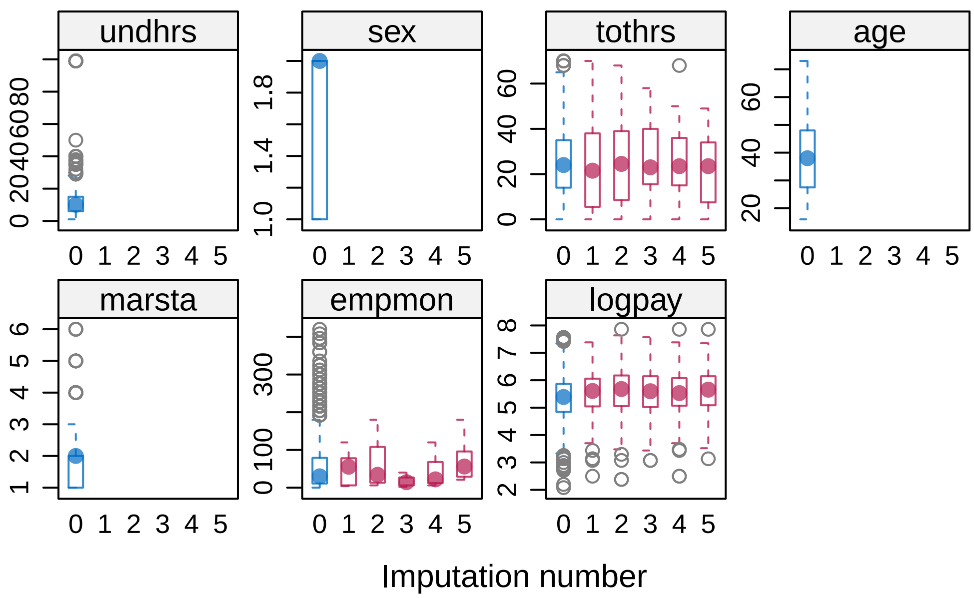
```
bwplot(imp.all[[1]])
```




```
bwplot(imp.all[[3]])
```




```
bwplot(imp.all[[5]])
```



Now let's have a look as to how the new imputations shape the results for our model regressing the number of working hours people would like to take on as a function of pay. We begin by fitting the pooled imputed regression for delta value -.1 (roughly 30% lower) such that imputed values are artificially reduced:

```r
fit_ss0 <- with(imp.all[[1]], lm(undhrs ~ logpay + age + sex + empmon +
                              hiqul15d + tothrs + nsecmj10))
summary(pool(fit_ss0))
```

```
          term    estimate   std.error   statistic          df      p.value
1  (Intercept) 45.90974025 8.327323006  5.51314513   16.182600 4.534071e-05
2       logpay -3.14577023 1.421323711 -2.21326796    9.841508 5.169199e-02
3          age -0.09738534 0.044385583 -2.19407596  902.522762 2.848423e-02
4          sex -3.67567640 1.109757193 -3.31214469  920.019548 9.619055e-04
5       empmon -0.01107172 0.007564682 -1.46360625  406.223937 1.440749e-01
6    hiqul15d2 -0.60990783 1.897210952 -0.32147602 1104.768513 7.479105e-01
7    hiqul15d3 -3.49774074 1.547421436 -2.26036725  802.835014 2.406555e-02
8    hiqul15d4 -1.93625243 1.587368378 -1.21978771  677.812932 2.229696e-01
9    hiqul15d5  0.62756492 2.040097977  0.30761509  370.850552 7.585479e-01
10   hiqul15d6  0.10568184 2.441593152  0.04328397  525.291362 9.654916e-01
11      tothrs -0.13206760 0.058482288 -2.25824961   17.272547 3.714385e-02
12   nsecmj102  0.46116956 2.500766461  0.18441129  957.484555 8.537299e-01
13   nsecmj103 -2.52484935 2.661081761 -0.94880563  660.188889 3.430667e-01
14   nsecmj105  4.42558059 3.021066630  1.46490665  250.332152 1.442012e-01
15   nsecmj106 -0.16544551 2.639027279 -0.06269185  551.223791 9.500346e-01
16   nsecmj107 -0.81431805 2.810412051 -0.28975041  526.233810 7.721213e-01
17   nsecmj108 -7.99499815 3.518467019 -2.27229589  231.776672 2.398561e-02
```

Now we do the same for delta value 0 (no change)

```r
fit_ss1 <- with(imp.all[[3]], lm(undhrs ~ logpay + age + sex + empmon +
                                     hiqul15d + tothrs + nsecmj10))
summary(pool(fit_ss1))
```

```
          term    estimate   std.error  statistic        df      p.value
1  (Intercept) 41.67732492 6.327018620  6.5871981  734.3396 8.544575e-11
2       logpay -2.33629826 0.922338415 -2.5330163  664.3603 1.153753e-02
3          age -0.10214827 0.044108880 -2.3158211 1062.4859 2.075770e-02
4          sex -3.61371063 1.109289856 -3.2576793 1070.2506 1.158569e-03
5       empmon -0.01160769 0.007484423 -1.5509135  646.5752 1.214117e-01
6    hiqul15d2 -0.53972814 1.904249981 -0.2834334 1103.6840 7.768977e-01
7    hiqul15d3 -3.19229733 1.529800928 -2.0867404 1067.5399 3.714842e-02
8    hiqul15d4 -1.64442747 1.565873968 -1.0501659 1055.3779 2.938822e-01
9    hiqul15d5  0.89691875 1.991542185  0.4503639  812.1901 6.525682e-01
10   hiqul15d6  0.24084354 2.435703103  0.0988805  586.4539 9.212669e-01
```

```
11      tothrs -0.15308896 0.044191385 -3.4642264  544.2130 5.736558e-04
12   nsecmj102  0.75410700 2.483248526  0.3036776 1105.6110 7.614306e-01
13   nsecmj103 -2.03287571 2.614319573 -0.7775927 1096.5211 4.369769e-01
14   nsecmj105  4.88538047 2.891234339  1.6897214  995.3662 9.139441e-02
15   nsecmj106  0.45878479 2.589179670  0.1771931 1079.9974 8.593899e-01
16   nsecmj107 -0.31077171 2.754719243 -0.1128143 1080.7140 9.101987e-01
17   nsecmj108 -6.90947243 3.350950697 -2.0619439 1041.0330 3.946083e-02
```

Now we do the same for delta value +3 (imputed values are increased)

```r
fit_ss4 <- with(imp.all[[5]], lm(undhrs ~ logpay + age + sex + empmon +
                                   hiqul15d + tothrs + nsecmj10))
summary(pool(fit_ss4))
```

```
          term    estimate  std.error    statistic         df      p.value
1  (Intercept) 39.19348087 8.46573646  4.629659932   17.18528 0.0002331541
2       logpay -1.84155387 1.29982801 -1.416767336   13.12429 0.1798486593
3          age -0.10624515 0.04414164 -2.406914257 1049.88979 0.0162597724
4          sex -3.51950774 1.13034910 -3.113646688  670.87798 0.0019265306
5       empmon -0.01247560 0.00744244 -1.676278441  725.93251 0.0941142345
6    hiqul15d2 -0.60435717 1.91005087 -0.316408940 1031.73146 0.7517561352
7    hiqul15d3 -3.18585374 1.55826169 -2.044492113  692.85639 0.0412823968
8    hiqul15d4 -1.67485389 1.62123082 -1.033075528  421.00708 0.3021616192
9    hiqul15d5  0.87585875 2.03497126  0.430403500  422.99310 0.6671213571
10   hiqul15d6  0.74562895 2.57137740  0.289972585  147.22125 0.7722452242
11      tothrs -0.16873607 0.04772089 -3.535895088   85.68144 0.0006583564
12   nsecmj102  1.13859925 2.50433538  0.454651263  972.07882 0.6494616580
13   nsecmj103 -1.75576522 2.67195867 -0.657107924  632.28610 0.5113506330
14   nsecmj105  5.16094142 2.92490467  1.764481921  676.99088 0.0781017718
15   nsecmj106  0.84518103 2.69080999  0.314099111  317.92319 0.7536517700
16   nsecmj107 -0.00399468 2.84580763 -0.001403707  424.06183 0.9988806644
17   nsecmj108 -6.34540807 3.60001990 -1.762603609  147.97346 0.0800319187
```

Note that each delta value alters in some way the coefficient for log pay, potentially leading to different conclusions.

Conclusions

In this tutorial we have looked at how imputation can help assess the effects of missing data on our regression estimates for a relatively run-of-the-mill survey analysis of income and preferences for working hours.

Ignorability (certainly MAR) is a reasonable **starting** point for any analyst (even when MNAR is suspect) so provided that you have a sensible imputation model that maintains the structure and the relations amongst variables and that your analysis model is correctly specified then you may want to consider incorporating MI in any research that involves incomplete data.

How you build the imputation model should be based on your substantive knowledge about the missingness as well as the nature of the relationship between variables (how well they predict each other).

If you suspect that missingness could still be not random after conditioning on the observed data (i.e. that respondents and non-respondents cannot be assumed to be equal) then a sensitivity analysis is a worthwhile check, though the choice of delta values has to again be informed by substantive knowledge of the research area. Use with caution!