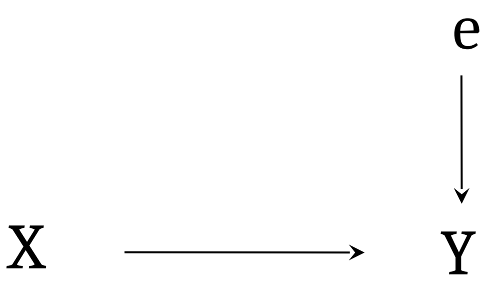
Practical exercise Missing data – with answers

Part A: study missingness mechanisms through simulations

We generate datasets with different types of missing mechanisms (using R). We will explore if we are able to estimate the true parameter values back from the simulated data with different analysis approaches.

Try to construct a DAG for each data generating mechanism. The basic DAG (without any missing observations) is presented below. X is the exposure, Y is the outcome, e is a (residual) error term acting on Y. You can add the missingness mechanism into the DAG by adding a 'selection' node S that is conditioned on when restricting the analysis to only cases with fully observed data.



1. **Data generation.**

   Generate a dataframe (named 'dat') with 1000 cases (rows), each with a value for variable X and variable Y belonging to the following 'true' model:

   $Y = 2 + 5X + e$,

   With

   X a sequence of 1000 equally spaced numbers between 0 and 1: 0.001, 0.002, ..., 1
   e a normally distributed random variable with mean 0 and standard deviation σ=1

   Save the dataframe in such a way that you can easily re-use it in questions 2-7 below.

   See R code

2. **No missings**

   Estimate a linear model relating X to Y. Are the parameters estimated accurately?

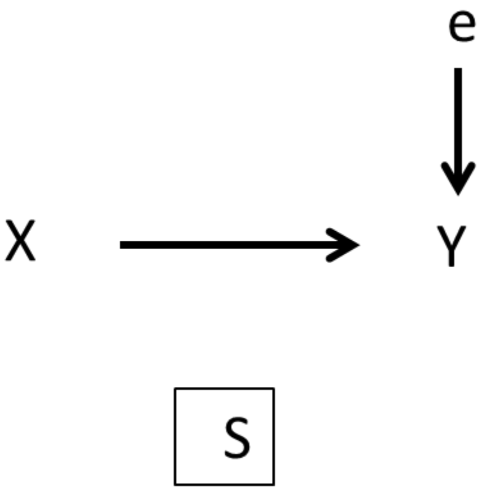
   (hint: lm(Y~X, data=df)).

Parameters are estimated accurately:

Intercept close to 2, standard error around 0.06/0.07

X-coefficient close to 5, standard error around 0.11

3. Y missing completely at random



a. Make half of the Y values missing using a completely at random mechanism.

   (hint: add a random 'missing indicator' to the dataframe and put Y-values at NA for those cases with missing indicator value 1).

   See R code

b. Estimate the same linear model as in (2) but now using only the cases with observed Y value. Are the parameters accurate (look both at point estimates and standard error and compare to 2)?

   Standard errors are somewhat larger (a factor sqrt(2) larger), but taking this random noise into account, point estimates are still accurate

c. Use multiple imputation to 5 times impute the missing Y values and estimate the linear model on the imputed datasets. Are the parameters estimated correctly now (again look both at point estimates and standard errors)?

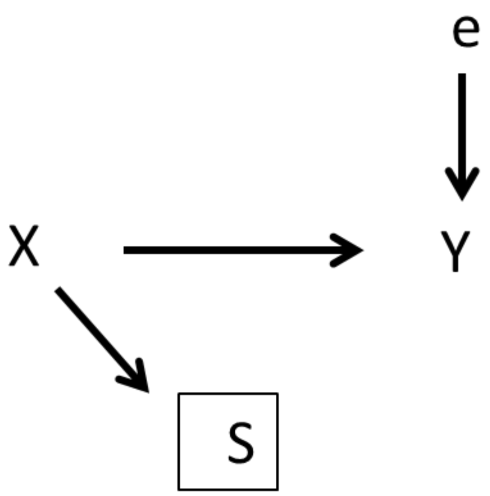
   (hint:

```
imp1 <- mice(dat,method="norm", m=5)

# Inspect the multiple imputed information

imp1

# Explore imputed values
```

```
imp1$imp

# Perform linear regression on all imputed datasets

fitmi <- with(imp1, lm(Y ~ X))

summary(fitmi)

# pooled results

summary(pool(fitmi))

# make a combined dataframe

dat_imp <- complete(imp1, "long", inc = TRUE)

view(dat_imp)

)
```

Imputation yields comparable estimates and se’s (not much to gain in this setting with only one covariate).

4. **Y Missing dependent on X**
   a. Start anew from the complete data frame generated in part 1. Set all Y values to NA for cases where X>0.5.
   See R script
   b. How would you classify this missing mechanism?
   Y is missing at random dependent on X
   c. Draw the DAG matching this mechanism



   d. Estimate the same linear model using the cases with complete observations for X and Y. Are the parameter estimates accurate? Look both at point estimates and standard errors
   Estimates are accurate (except for random noise). Standard error is larger.

e. Now use multiple imputation like in 3. Again look whether estimated coefficients are accurate.

Imputation yields comparable estimates and standard errors.

5. Y Missing dependent on Y

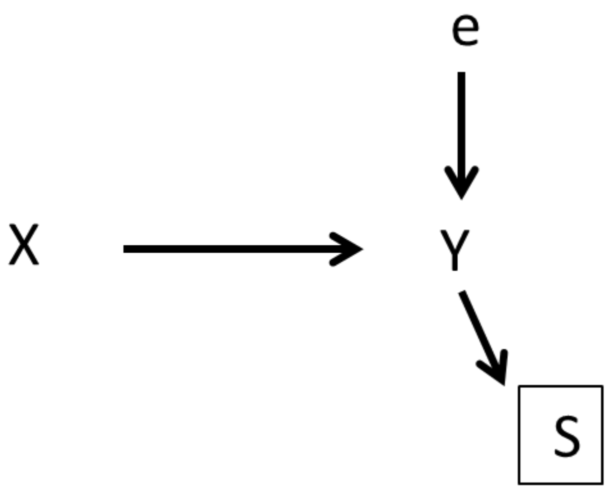
a. Introduce missing values similar to 4a. But now put the Y values to NA if Y>0.5.

See R script

b. How would you classify this missing mechanism?

Y is missing NOT at random

c. Draw the DAG matching this mechanism



Note in this DAG there is selection on a collider!

d. Estimate the same linear model using the cases with complete observations for X and Y. Are the parameter estimates accurate? Look both at point estimates and standard errors

The point estimates are biased

e. Now use multiple imputation like in 3. Again look whether estimated coefficients are accurate.

Imputation does not help

6. Y missing dependent on X and Z

a. Generate a new version of Y using the following model:

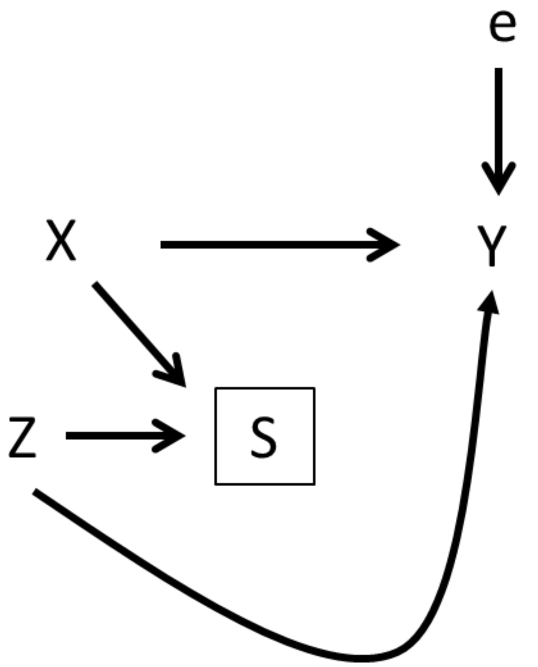
$Y = 2 + 5X + Z + e$,

with

X a sequence of numbers 0.001, 0.002, ..., 1

e a normally distributed random variable with mean 0 and standard deviation σ=1

Z a second standard normal variable



b. Set Y values to NA for cases with X>0.5 and Z>0.

   See R script

c. Pretend that Z was not measured/available. Fit a linear model relating X to Y and check the estimates of the parameters.

   Estimates are biased.

d. Use multiple imputation (again using only X), fit a linear model relating X to Y and check the estimates of the parameters

   Results still biased

e. Now include Z during the multiple imputation and again fit a linear model relating X to Y and check the parameter estimates.

   Now estimates are accurate

f. How would you classify the missing mechanism if Z was measured?

   Y is missing at random dependent on X and Z

   And if it was not measured?

   In this setting Y is missing NOT at random

7. Should the outcome be included in the imputation model for missing covariate values?

   a. Start from the dataset generated under 1. Introduce missing X values according to a missing completely at random mechanism. Fit a linear model on the complete observations.

      DAG as in exercise 3. Estimates are accurate

   b. Use multiple imputation to generate 5 imputed datasets. Instruct R not to use Y when imputing X.

      see R script

   c. Fit again a linear model relating X to Y on the imputed dataset. Are the estimates accurate?

See R script. Estimates are no longer accurate

d. Now use multiple imputation using default settings in which X is imputed based on Y. Fit the linear model relating X to Y. Are the estimates accurate?
   Estimates are accurate. We need the outcome in the imputation model, otherwise the effect of X on Y is diluted

Part B: Specification of imputation method in real data application

In this exercise we study specification of the imputation model using a (modified version of) real data set. The Netherlands Cooperative Study on the Adequacy of Dialysis (NECOSAD) study was a prospective multicenter cohort study in which patients with end stage renal disease were included at dialysis initiation if they were 18 years or older and had no previous kidney transplantation and no previous dialysis. Patients were followed for several outcomes. Here we focus on the outcome death (yes/no). The research question of interest is whether the type of dialysis - peritoneal dialysis (PD) versus hemodialysis (HD) - is related to the risk of dying.

(Some background: With HD, blood is pumped out of the patient's body and filtered by an artificial kidney machine. With PD, cleansing fluid is pumped into the patient's abdominal cavity and the lining of the abdomen acts as a natural filter to wash out waste and toxins.)

1. **Data import and exploration**
   a. Import the data file 'necosad_death_miss.csv' into your R session and store it in a dataframe named 'dat'.
      See R script
   b. Explore the data set and missing data patterns using e.g. the functions 'str', 'summary' and 'md.pattern'.
      - The data contains 10 variables. Besides the patient id, the treatment variable 'therapy' and the outcome variable 'death' there are 7 other patient characteristics recorded: age, sex, bmi, albumin, kidneydisease, smoking and gfr. Out of the 1179 patients, less than half (541) have complete observations. Therapy and gfr have the most missing values.

2. **Complete case analysis**
   a. Relate death to therapy type (peritoneal versus hemodialysis) using logistic regression. Perform first a univariable 'unadjusted' regression including only 'therapy' and then perform a model additionally including all 7 other covariates as they are potential confounders for the relation between therapy and death.
      See R script
   b. Look at the difference in estimated odds ratio for therapy between the two models: does adding the potential confounders change the estimate of the therapy effect?
      OR for therapy (PD vs HD) univariable model: 0.66; p-value 0.005
      OR for therapy (PD vs HD) multivariable model: 1.06; p-value 0.172
      A very large difference in estimate of treatment effect after adding the potential confounders

c. How many patients are used in each of the analyses? How does that impact your answer to question b?

In univariable analysis 200 patients are excluded, in multivariable analysis 638 are excluded. The analysis sets are thus not comparable and we cannot know whether the difference in odds ratio described under b is due to confounding adjustment or due to looking at a different set of patients.

3. Multiple imputation based on linear regression for gfr

a. Use multiple imputation (with m=5). The default setting of mice is to use predictive mean matching for numeric data. We here change that setting for the variable ‘gfr’. In particular, specify that you would like to use ‘Bayesian linear model’ for imputing ‘gfr’. You can do this by setting the ‘method’ argument to “norm” for ‘gfr’.

(Hint:

```
#extract default methods settings
ini <- mice(dat, maxit = 0)
meth <- ini$meth
meth
#change the method for “gfr”
meth["gfr"] <- "norm"
meth
#use the new ‘meth’ settings during imputation
imp1 <- mice(dat,method=meth, m=5))
```

See R script

b. Run the multivariable logistic regression now using the imputed data and compare the odds ratios for gfr and for therapy with those from the complete case analysis in 2b.

After imputation odds ratio for therapy is 1.01 (p-value 0.74). Imputation did not change the estimate of the therapy effect much compared to the multivariable model using the complete cases.

c. Explore the imputed values especially for ‘gfr’ and compare these to the distribution of the original gfr values. Did the imputation give plausible values for ‘gfr’?

No! The imputed values follow a symmetric distribution and contain negative values. The original values were skewed and only positive values

4. Multiple imputation based on linear regression for log(gfr)

a. Transform the gfr variable using a log(x+1) transformation. Explore distribution of this transformed variable.

(hint:

```
dat$loggfr <- log(dat$gfr+1))
```

Distribution of the transformed values is more symmetrical.

b. Redo the imputation after log-transforming the ‘gfr’ variable. You have to additionally tell mice not to use the original ‘gfr’ variable during imputation

(Hint:

```
ini <- mice(dat, maxit = 0)
# define methods for imputation
meth["loggfr"] <- "norm"
meth["gfr"] <- "~I(exp(loggfr)-1)"
# and do not use gfr in the imputation models
predMat <- ini$predictorMatrix
predMat[,"gfr"] <- 0
imp2 <- mice(dat,method=meth,predictorMatrix = predMat, m=5))
```

See r script

c. Check the distribution of the imputed ‘gfr’ variable. Are the values plausible now?

Imputed values are more similar to the original values now, so more plausible.

d. Run the multivariable logistic regression now using the data imputed after log transformation and compare the odds ratios for gfr and for therapy with those from the analysis in b. Did the improvement of the imputation method change the odds ratios for therapy and gfr?

Odds ratios did not change much with improving imputation of gfr. Note the imputed values of all other variables have not been checked yet.