University of Zurich[UZH]

MAKERERE UNIVERSITY

Data Analysis with R:
Day 8

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2017

Acknowledgement

We would like to thank **Beate Sick** for the inspiration given for these slides on *missing values* and *multiple imputation*.

Missing Data*

- **Nonresponse** occurs when information on all variables is missing for a subset of the sample. Such non-responders may cause **selection bias**, as they effectively modify the study population to include only respondents.
- **Item nonresponse** occurs when only some variables have missing responses for a given individual.
    - If missing probabilities of a predictor do not depend on the outcome, then X is missing **nondifferentially**. Association between outcome and missingness of a predictor can be tested!
    - **Differential missingness probabilities** depend on disease outcome. This mainly arise in retrospective (e. g. case-control) studies. If predictors (exposure) are measured prospectively before outcome such dependency cannot occur in a direct way.

Example: A study of HIV-positive people in NYC

The `CHAIN` project was a longitudinal cohort study of people living with HIV in New York City, which was recruited in 1994. Here we study a data subset collected from 532 subjects in the sixth round of interviews.

- `log_virus`: Log of self reported viral load level (=0 if below detection limit).
- `age`: The respondent's age at time of interview.
- `income`: The respondent's family annual income.
- `healthy`: A continuous scale of physical health (0-100).
- `mental`: Measure of poor mental health: 0/1 = No/Yes.
- `damage`: Ordered interval (0 to 5) for CD4 count.
- `treatment`: A three-level-ordered variable (0 = not currently taking HAART therapy, 1 = taking HAART nonadherent, 2 = taking HAART adherent)

Reading in the CHAIN data set in R

```
install.packages("mi")
library("mi")
data("CHAIN")
chain <- CHAIN # rename the data frame
names(chain)
str(chain)
summary(chain)
```

Descriptive Statistics of the CHAIN data set in R

```r
hist(chain$log_virus, ylim = c(0, 200), col = "darkgray")
nrow(chain) # [1] 532
apply(chain, 2, function(x) {sum(is.na(x))})
# log_virus       age    income   healthy    mental    damage treatment
#       179        24        38        24        24        63        24
summary(chain)
```

Reasons for Item Nonresponse

- Individuals may not respond to specific questions on sensitive issues (e. g. questions on household income, drug abuse, ...).
- Medical or occupational records may be incomplete. Material might be spilled or reports might be lost in the mail.
- Certain variables may be very expensive to measure for all study participants.

Missingness Mechanisms

- **MCAR**: The missing completely at random (MCAR) mechanism assumes, that the probability that a specific variable X is missing is does not depend on any other factors.
- **MAR**: The missing at random (MAR) mechanism assumes, that the probability that a specific variable X is missing does not depend on the missing value of X, but only on fully observed variables.
- **MNAR**: Otherwise data are not missing at random (MNAR).

Table 7.2 Three types of missing data mechanisms

Label	Missing mechanism	Description
MCAR	Missing completely at random	Administrative errors, accidents
MAR	Missing at random	Missingness related to known patient characteristics, time or place ("MAR on *x*"), or to the outcome ("MAR on *y*")
MNAR	Missing not at random	Missingness related to the value of the predictor, or to characteristics not available in the analysis

Table taken from Steyerberg

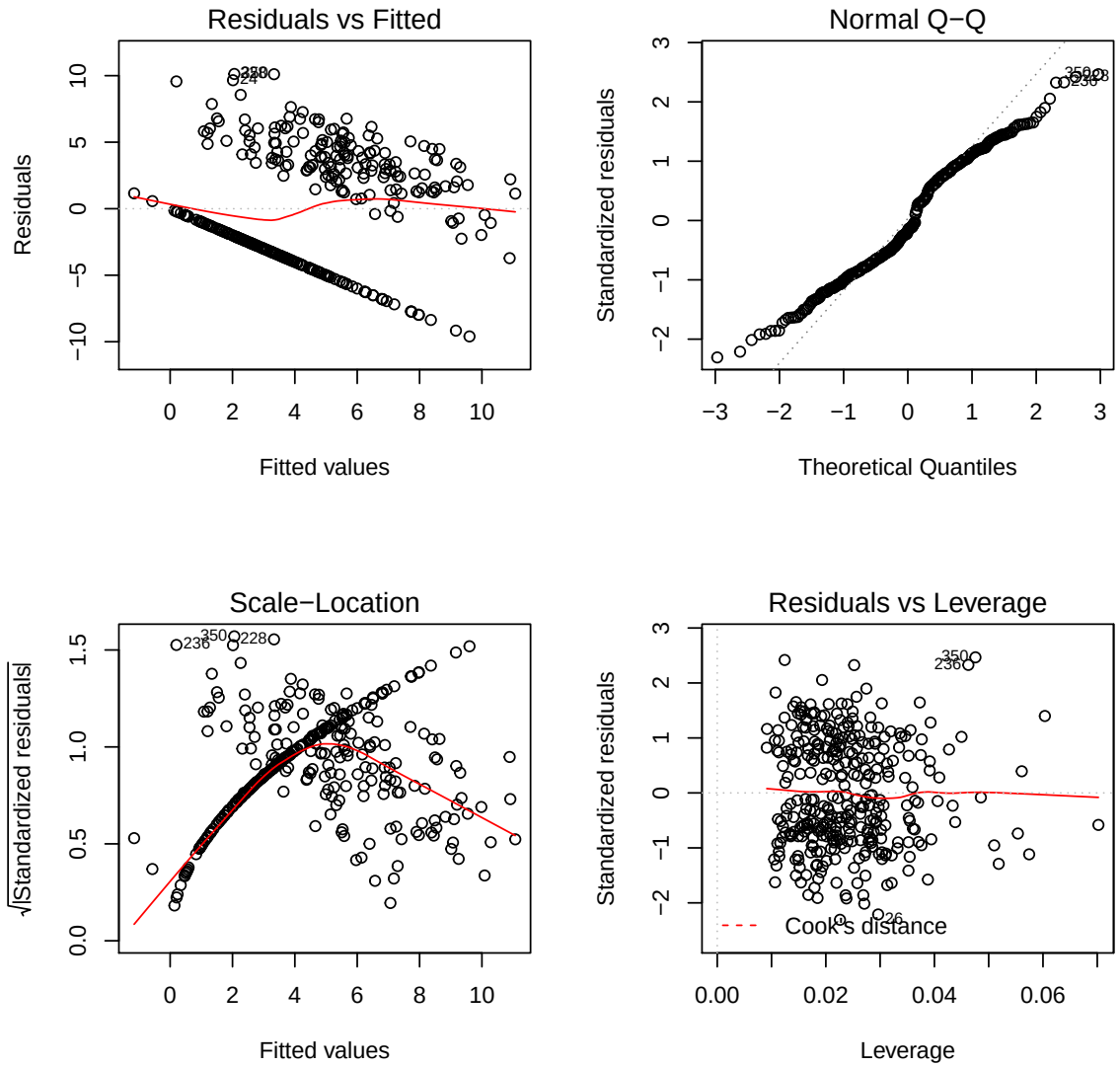
Fitting a linear regression model to the `chain` data set

```
# R performs a complete case analysis using only rows with no NAs
lm.mod.NA <- lm(log_virus ~ age + income + healthy + mental +
                  damage + treatment, data = chain)
summary(lm.mod.NA)
```

```
##
## Call:
## lm(formula = log_virus ~ age + income + healthy + mental + damage +
##     treatment, data = chain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6015 -3.2863 -0.7366  3.5620 10.1283
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.37464    1.84104   9.437  < 2e-16 ***
## age         -0.10715    0.02914  -3.677 0.000276 ***
## income      -0.36671    0.11319  -3.240 0.001320 **
## healthy     -0.03627    0.01980  -1.832 0.067858 .
## mentalyes    0.97222    0.53518   1.817 0.070191 .
## damage      -1.18613    0.17947  -6.609 1.57e-10 ***
## treatment1  -2.05802    0.59768  -3.443 0.000649 ***
## treatment2  -2.21456    0.53367  -4.150 4.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.208 on 327 degrees of freedom
##   (197 observations deleted due to missingness)
## Multiple R-squared:  0.2506,Adjusted R-squared:  0.2345
## F-statistic: 15.62 on 7 and 327 DF,  p-value: < 2.2e-16
```

Checking the model assumptions for the `chain` data set

```
# R performs a complete case analysis using only rows with no NAs
par(mfrow=c(2,2))
plot(lm.mod.NA)
```



Complete Data Analysis in linear regression

- Complete case analysis uses only those records with complete observations
- If the predictor values are missing nondifferentially, i. e. the probability that a risk factor is missing does not depend on outcome y (disease status), then complete case analysis produces valid regression coefficient estimates
- If the predictor values are missing differentially the complete case analysis can result substantial biased estimates.
- There may also be a huge loss in precision (large CIs and *p*-values) if a substantial part of the observations are omitted due to missing values.

How are missing data treated?

Simple solutions that often cause bias or other problems:

- Complete data analysis = Listwise deletion
- Pairwise deletion
- Single imputation (not discussed)
- Last observation carried forward and baseline observation carried forward (not discussed)

Better (but also more complex) approaches are

- **Multiple imputation**: mainly used to handle item missings - only parts of the variables are missing
- Inverse probability weighting (not discussed)

Iterative methods to get an imputed data set

If we have missings in more than one variable:

- **Multivariate Imputation by Chained Equations (MICE, Van Buuren and Oudshoorn (1999))**
  MICE imputes each variable stochastically using a regression model conditional on all the others, iteratively cycling through all the variables that contain missing data until convergence (”Gibbs sampling”). In R we can use the mice package. The mice function assumes in default settings linearity of associations among X variables and between X and Y in the default setting, but specific forms of imputation models can be specified by the user.
- **An iterative nonparametric missing value imputation for mixed-type data** can also be done by a method called missForest which is based on a random forest approach. It can can cope with high-dimensional data and may outperform MICE if complex interactions and nonlinear relations are suspected. In R we can use the missForest package.

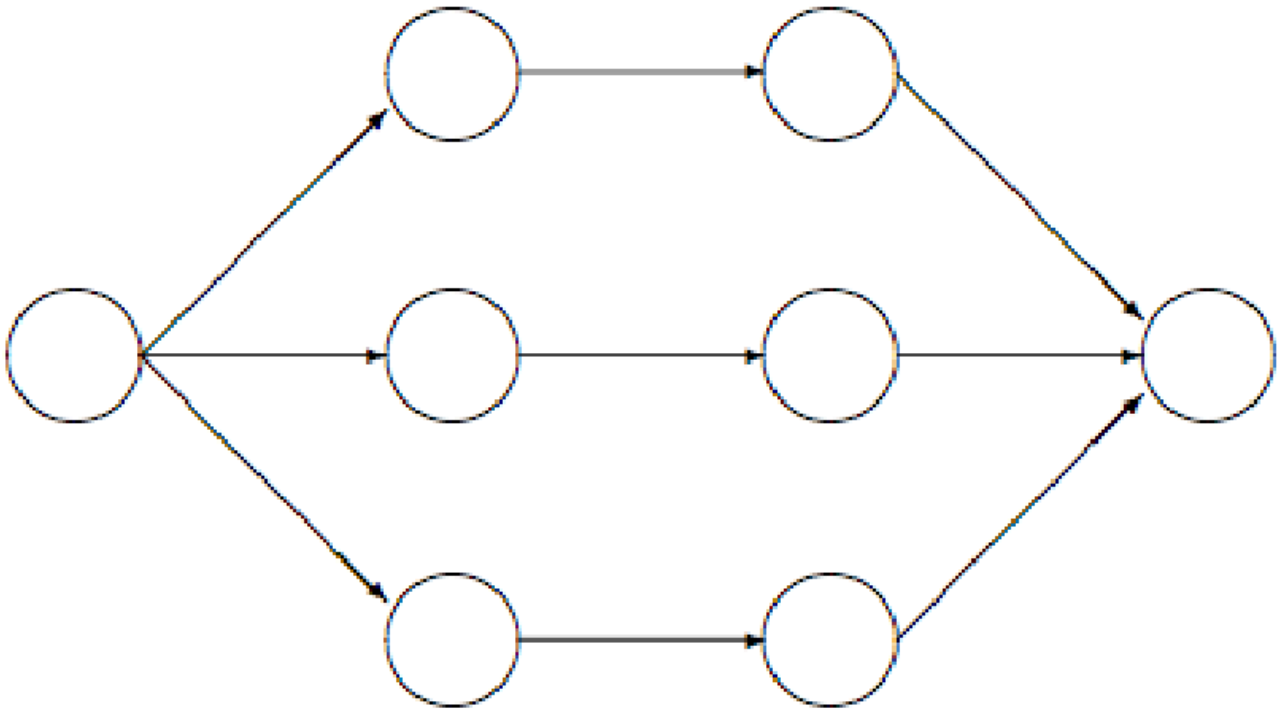
Iterative imputation based on `RandomForest`

In the R package missForest (2012) is an iterative imputation method. All imputation models are random forest models.

```
# install.packages("missForest")
library("missForest")
# Nonparametric Missing Value Imputation using Random Forest
chain.imp <- missForest(chain, verbose = TRUE, variablewise = TRUE)
chain.imp$OOBerror # estimated imputation error for each variable
chain.final.rf <- chain.imp$ximp
dim(chain.final.rf) # [1] 532   7
apply(chain.final.rf, 2, function(x){sum(is.na(x))})
# log_virus       age    income   healthy    mental    damage treatment
#         0         0         0         0         0         0         0
lm.mod.rf <- lm(log_virus ~ age + income + healthy +
                  mental + damage + treatment,
                data = chain.final.rf)
```

```
##   missForest iteration 1 in progress...done!
##     estimated error(s): 19.27203 69.01536 3.950885 145.4131 0.2755906 1.7060
##     difference(s): 0.000589767 0.009398496
##     time: 1.022 seconds
##
##   missForest iteration 2 in progress...done!
##     estimated error(s): 19.95495 66.14035 3.733105 145.3148 0.269685 1.70197
##     difference(s): 0.0003988586 0.01315789
##     time: 0.761 seconds
```

Main steps used in multiple imputation with `mice` in R



Incomplete data Imputed data Analysis results Pooled results

Multiple and iterative imputation with `mice` in R

```r
# install.packages("mice")
library("mice")
# Multivariate Imputation by Chained Equations (MICE)
# create 4 imputed data sets via mice
# IMPUTE DATA
chain.mice <- mice(chain, m = 4, print = FALSE, seed = 2017)
# ANALYSE RESULTS WITH IMPUTED DATA
# fit for each of the 4 data sets a linear regression
fit.4.mice <- with(chain.mice, lm(log_virus ~ age + income + healthy +
                                    mental + damage + treatment))
# check out the regression coefficients of the 4 fits
summary(fit.4.mice)
# POOL ANALYSIS RESULTS
# pool the results of the 4 models with Rubin's rule
lm.mod.mice <- pool(fit.4.mice)
# combined regression coefficients of the 4 fits
summary(lm.mod.mice)
```

Comparison of results: Complete case vs. missForest

```
round(summary(lm.mod.NA)$coef, 2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.37       1.84    9.44     0.00
## age            -0.11       0.03   -3.68     0.00
## income         -0.37       0.11   -3.24     0.00
## healthy        -0.04       0.02   -1.83     0.07
## mentalyes       0.97       0.54    1.82     0.07
## damage         -1.19       0.18   -6.61     0.00
## treatment1     -2.06       0.60   -3.44     0.00
## treatment2     -2.21       0.53   -4.15     0.00
```

```
round(summary(lm.mod.rf)$coef, 2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    15.19       1.22   12.46     0.00
## age            -0.09       0.02   -4.31     0.00
## income         -0.34       0.08   -4.15     0.00
## healthy        -0.03       0.01   -2.15     0.03
## mentalyes       1.03       0.36    2.89     0.00
## damage         -1.08       0.12   -8.62     0.00
## treatment1     -1.63       0.40   -4.11     0.00
## treatment2     -1.70       0.36   -4.68     0.00
```

Comparison of results: Complete case vs. `mice`

```
round(summary(lm.mod.NA)$coef, 2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.37       1.84    9.44     0.00
## age            -0.11       0.03   -3.68     0.00
## income         -0.37       0.11   -3.24     0.00
## healthy        -0.04       0.02   -1.83     0.07
## mentalyes       0.97       0.54    1.82     0.07
## damage         -1.19       0.18   -6.61     0.00
## treatment1     -2.06       0.60   -3.44     0.00
## treatment2     -2.21       0.53   -4.15     0.00
```

```
round(summary(lm.mod.mice)[,c(1,2,3,5)], 2)
```

```
##                est   se     t Pr(>|t|)
## (Intercept) 15.60 1.57  9.95     0.00
## age         -0.09 0.03 -3.25     0.00
## income      -0.35 0.11 -3.09     0.00
## healthy     -0.03 0.02 -1.55     0.13
## mental2      1.16 0.56  2.06     0.05
## damage      -1.01 0.17 -5.76     0.00
## treatment2  -2.27 0.56 -4.06     0.00
## treatment3  -2.13 0.52 -4.10     0.00
```

Multiple Imputation - the principle idea

- m imputed data sets are created instead of a single imputed data set (Rubin has shown that often $m = 5$ to 10 is sufficient)
- To create the m imputed data sets take random draws from the predictive distribution from an imputation model, e.g. 5 times or do iterative imputation several times with different starting values and draw random draws.
- The variation among the m imputations reflects the uncertainty with which the missing values can be predicted from the observed data.
- With each completed data set the planned analysis is performed.
- The results from the different analyses are then combined using Rubin's rule properly taking into account the uncertainty in the imputed values.

Summary on imputation

Missing data lead to

- Inefficient analyses of research questions
- Difficulties in interpretation when analyses differ in numbers of subjects
- Possible bias in regression coefficients

- **Imputation methods make the assumption of MAR**
- The MAR assumption is not testable, but becomes more reasonable with imputation models that include a wide range of variables.

- **Multiple imputation is recommended and superior to complete data analysis.**
- Single imputation methods are also reasonable for prediction.

Summary for Anna Mary

- ONE continuous vs. ONE categorical variable (2 levels)
  - `t.test(...)`
- ONE continuous vs. ONE categorical variable ($\geq$ 2 levels)
  - `aov(...)`
- ONE categorical variable (2 levels) vs. ONE categorical variable (2 levels) $\rightarrow$ 2 x 2 table
  - `chisq.test(...)`
  - `fisher.test(...)`

$\rightarrow$ Regression: `lm(...)` & `glm(...)`