eatRep: a package to analyze multiple imputed data in complex survey designs

Sebastian Weirich
Humboldt University Berlin, Germany

November 30, 2018

Abstract

Estimation of simple descriptive statistics becomes cumbersome, if the sample cannot be considered to be a (completely) random draw from the population for which descriptives should be interpreted. This occurs in weighted samples or clustered samples. The same is true if the variables of interest stem from a multiple imputation process and occur, for example, as plausible values. In the estimation of standard error, we then have to account for two possible sources of uncertainty: first the uncertainty due to a clustered sample, and second the uncertainty due to multiple imputed data. This tutorial describes some basic analyses to compute descriptives in complex survey designs using the R package `eatRep`, which was designed mainly to supply replications methods in R. Such methods are appropriate to analyze both clustered and multiple imputed data as well. To date, the Jackknife-1 (JK1), Jackknife-2 (JK2) and the balanced repeated replicates (BRR) methods are supported. Some functions overlap with methods provided in the computer software WesVar (Westat, 2000)—in this case the package only allows for executing these analyses in R, which may be easier to implement due to a syntax related interface. Some methods in WesVar are not completely implemented in `eatRep` yet, for example bootstrapping methods. For bootstrapping, alternative `R` packages (e.g. `boot`) may be used. However, some methods are only implemented in `eatRep`, for example analyses for nested imputed data, linear logistic regression models, or trend analyses. Examples considering the latter one are not yet included in this vignette. The

examples 6 to 6d from the help page of the `defineModel` function in the `eatModel` package contain some exhaustive demonstrations of trend analyses.

`eatRep` heavily relies on the `survey` package (Lumley, 2012) which functions has been extended by methods for multiple imputed data. While the functional principle of `survey` is based on replication of conventional analyses, `eatRep` is based on replication of `survey` analyses to take multiple imputed data into account.

1 Introduction

In a completely random sample, the mean

$$\bar{x} = n^{-1} \sum_{i=1}^{n} (x_i) \tag{1}$$

is an unbiased estimate for the corresponding mean

$$\mu = N^{-1} \sum_{i=1}^{N} (x_i) \tag{2}$$

of the underlying population the sample was drawn from. This does not hold for dispersion measures (variance and standard deviation), as the variance in a sample is always less than the variance in the population the sample was drawn from. The transformation, however, is very easy made: The variance in a sample is multiplied by $n/(n-1)$ to obtain population variance, where n is the sample size. Based on

$$\sigma^2 = N^{-1} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{3}$$

for the population with N elements, we apply

$$s^2 = (n-1)^{-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{4}$$

to estimate population variance from a sample of size n. In a weighted sample, i.e. if the population weights differ between examinees in the sample, mean

and variance may be estimated by incorporating these population weights. (In a completely random sample, these weights equal 1 for each examinee.)

$$\bar{x}_w = \sum_{i=1}^{n} \left(\frac{w_i}{W} x_i \right), \tag{5}$$

$$s_w^2 = \sum_{i=1}^{n} \frac{w_i}{W-1} (x_i - \bar{x})^2, \tag{6}$$

where w_i is the case weight of the ith person, and W is the sum over all case weights, i.e. $W = \sum w_i$. To summary, the crucial point in the estimation of population variance estimates is the factor $n/(n-1)$. Unfortunately, this factor only applies when we sample (conditionally) independently from the population, as in completely random samples or weighted random samples. In a clustered sample, however, where schools or classes are sampling units instead of single persons, the relationship between sample and population variance is not so clear at all. The reason is that persons *within* a cluster (for example pupils in a class) often share a common variance. The sample variance underestimates the population variance, but more severely than indicated by the factor $n/(n-1)$. To estimate the relationship between sample and population variance, it is necessary to estimate the variance explained by the cluster.

Without taking the cluster structure into account, we would not only obtain biased variance estimates but biased standard errors, too (Luke, 2009). This problem occurs in the same way for estimation of frequency tables, quantiles or estimates of (linear) regression models. To gain unbiased estimates, several replication methods were introduced, which based on the same principle: To estimate the proportion by which the variance in the sample is underestimated due to a clustered structure (Lumley, 2004). In the Jackknife-2 (JK2) procedure this is implemented by reproducing the original sample to several replicates. In each replicate one clustering unit (e.g. one class) of only one primary sampling unit (PSU) is replaced by another class, which therefore occurs two times in the sample. Each replicate is analyzed if it would have been a completely random sample. Recognize what is to be expected then: If the variance is explained partially by the clusters, removing one sampling unit should decrease the variance of the sample slighty. Conversely, the point estimates of each replication sample should vary slightly.

The variance in the point estimates between the replicates is used to estimate the corresponding standard errors. Otherwise, if there is no variance between clusters, removing one cluster would have no effect on the variance estimate, and the point estimates between replicates would have no or only very little variance. In this case replication methods will result in exactly the same variance estimates and standard errors as they would follow from conventional analysis. The balanced repeated replicates (BRR) method is quite similar. The original sample is reproduced to several replicates. In each replicate one clustering unit (e.g. one class) *of each PSU* is replaced by another class, which therefore occurs two times in the sample. Each replicate then is analyzed if it would have been a completely random sample. For further details, see Rust and Rao (1996).

For the purpose of illustration, assume a simple population mean which has to be estimated from a completely random sample of $N = 1000$. To estimate the standard error of this mean, we may apply a rather laborious method: to draw 100 samples (with replacement) from our original sample, each of $N = 1000$, and compute the mean in each sample. The standard deviation of the 100 mean estimates is the standard error of the mean. Of course, this bootstrap method is far to cumbersome, as in a random sample the standard error can be estimated in a much more easier way. However, in a clustered sample, an extension of this bootstrap method is appropriate indeed. Several software (Westat, 2000) and free R packages such as `survey` (Lumley, 2012) do allow for several replication methods.

The situation is becoming still more complicated when the variables in the data to be analysed occur as (multiple) imputed data, for example as plausible values. Where missing values may cause biased parameters, analyses are conducted with imputed data. Often, the original data which includes missing values is reproduced several times, whereas the missing entries are filled with a set of plausible values, which results in several imputed data sets. To gain unbiased parameter estimates, the analyses are conducted for each data set separately and pooled afterwards according to Rubin (1987).

If we have both, a clustered sample with multiple imputed data, both methods have to be combined. This leads to a replication of replications. Analyses have to be repeated to account for the clustered structure, and the results of these replications have to be repeated to account for multiple im-

puted data. In the following, we refer to "cluster replicates" and "imputation replicates" to differentiate between both.

2 Estimate some population descriptives

In this example, we use some artificial data from the context of educational research. We may think of a stratified clustered sample of German fourth-grade primary school students whose reading and listening competencies are measured. Proficiency estimates obtained from a Item response Theory (IRT) marginal model are included as plausible values. Each plausible value may be recognized as an imputation of the latent competence construct. The data are represented in the long format.

```
> library(eatRep)
> data(lsa)
> str(lsa, give.attr = FALSE)
'data.frame':        90216 obs. of  22 variables:
 $ year     : num  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
 $ idstud   : Factor w/ 7518 levels "P0001","P0002",..: 2362 175 1283 783 728 2122 732 2429 2899 2191 ...
 $ wgt      : num  2.51 5.19 6.1 4.71 4.42 ...
 $ jkzone   : num  18 86 79 5 3 8 3 20 13 12 ...
 $ jkrep    : num  1 0 1 0 1 1 1 1 0 0 ...
 $ imp      : num  3 3 2 2 2 2 1 2 3 2 ...
 $ nest     : num  1 2 1 1 2 1 2 1 2 2 ...
 $ country  : Factor w/ 3 levels "LandA","LandB",..: 2 3 2 3 3 2 3 2 1 2 ...
 $ sex      : Factor w/ 2 levels "female","male": 2 2 1 2 2 2 1 2 1 1 ...
 $ ses      : num  24.8 28.5 23.5 64.4 70.3 ...
 $ mig      : num  0 1 0 0 0 0 0 0 0 1 ...
 $ domain   : Factor w/ 2 levels "listening","reading": 1 1 1 1 1 1 1 1 1 1 ...
 $ score    : num  342 317 286 327 360 ...
 $ comp     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ failMin  : num  1 1 1 1 1 1 1 1 1 1 ...
 $ passReg  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ passOpt  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ leScore  : num  1.21 1.21 1.21 1.21 1.21 ...
 $ leComp   : num  0.00582 0.00582 0.00582 0.00582 0.00582 ...
 $ leFailMin: num  0.00494 0.00494 0.00494 0.00494 0.00494 ...
 $ lePassReg: num  0.00601 0.00601 0.00601 0.00601 0.00601 ...
 $ lePassOpt: num  0.00141 0.00141 0.00141 0.00141 0.00141 ...
```

Requesting the data structure provides us with information about the number and type of variables and the number of examinees. `"idstud"` denotes the year of the assessment, `"idstud"` is a person identifier for 7,518 distinct examinees, and `"wgt"` a person weight. `"JKZone"` and `"JKrep"` denote jackknifing variables which contains information about which unit has to be replaced by which other unit in which replicate of the original data.

The next two variables, `"imp"` and `"nest"` describe the multiple imputed structure of the data. The data stem from a nested multiple imputation model with 2 nests and 3 imputations in each nest. The principles of nested imputations will be elucidated later; for the moment, we may content ourself with data from the first nest only, i.e. we split the data and only consider cases for which $nest = 1$. `"country"` denotes the country the person stems from, `"sex"` denotes each person's sex, `"ses"` is each person's socio-economical status, `"mig"` is an indicator for migration background. `"domain"` denotes whether the corresponding score value is related to reading or listening. We may think of `"score"` as the the plausible value estimate for the reading or listening competence. Hence, if $nest = 1$ and $imputation = 3$ and `"domain"` equals "reading", the value in the `"score"` column refers to the third plausible value in the first nest for the reading competence. `"comp"` is a distinct competence level for each person, where 1 corresponds to the lowest competence level, and 5 corresponds to the highest competence level. `"failMin"` is an indicator which equals 1 if the examinee fails to fulfill the minimal standard, 0 otherwise. `"passReg"` is an indicator which equals 1 if the examinee fulfills or outperforms the regular standard, and `"passOpt"` is an indicator which equals 1 if the examinee fulfills the optimal standard. The following variables beginning with "le" denote the linking errors of each criteria, i.e. `"lePassreg"` is the linking error for the indicator of fulfilling the regular standard.

Please note that the `"score"` variable contains the three imputations of the reading competence. Hence, each individual is represented in several rows. This is quite usual if multiple imputed data is presented in a long format dataset. `eatRep` strictly requires the long format. To transform wide-format data frame into long-format data frames (and vice versa), use the `reshape2` package. Please note further that the dataset does not contain any replicates, only the information required for generating them are captured in the `"JKZone"` and `"JKrep"` variables.

2.1 Populations means, standard deviations, variances and mean differences

We now want to compute the means by each country, considering the clustered structure as well as the multiple imputed data structure. The replicates do not need to be created separately, as they will be generated in each analy-

sis automatically. Even in large data sets this takes only a few seconds. First we create a subset which only contains reading data from the first nest from the year 2010. The analysis then is conducted with this subset.

```
> read  <- subset(lsa, domain == "reading")
> readN1<- subset(read, nest == 1 )
> read10<- subset(readN1, year == 2010 )
> means <- jk2.mean(datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+           PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = "country",
+           dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
```

When applying the jackknife method, the primary sampling unit (PSU) often is the jackknife zone (jkzone), and the replication indicator often is the jackknife replicate indicator (jkrep). While the function is operating, some additional information is displayed on console. First we see that only one analysis is run according to `'group.splits = 1'`. We will subsequently exemplify this enigmatic message. Further, we see that `jk2.mean` assumes an "unnested", i.e. a structure with three imputations. This refers to what we've called "imputation replicates". The output then speaks about 62 replicate weights which are created due to 62 distinct jackknifing zones in the JKZone variable. This information refers to the "cluster replicates" and implies that the subsequent analysis has to be repeated 62 times *for each imputation*. Hence, $3 \times 62 = 186$ analyses are run overall.

In each of the 62 replication samples, one unit (e.g. school) of a certain jackknifing zone is missing and the weights of the other school within the same zone are doubled. The data in all other zones remain unchanged. The analyses are repeated 62 times, *using the same plausible value* as the dependent variable. Only the weights vary between the "cluster replicates": each of the 62 replicates once a time is used as the weighting variable. Each of the 62 analysis revealed slighty different results. This variation is used to estimate the sampling variance *due to the clustered structure* which then is used to compute the standard error, which is pooled across 62 "cluster replicates". When finished, the analysis approaches the second "imputation replicate" and switches to the second plausible value which now is used as the dependent variable in 62 analyses due to the 62 "cluster replicates". After all, three pooled estimates and three pooled standard errors resulted which

differ slightly in each "imputation replicate". The three estimates and standard errors are pooled once again, this time according to Rubin (1987). To sum up, the pooled results are pooled again to account for both: multiple imputed data in a clustered sample.

The little dots continuously appearing on the console therefore refer to "imputation replicates" and are intended to work as a rough progress bar. Each dot represents one "imputation replication". When the procedure finished, the results are pooled in the case of more than one imputation.

The output is a list of data frames which is not intended to be inspected by the user. Instead, a reporting function transform the raw output in a more user-friendly format which can be saved as an Excel or csv file for further treatment. The reporting function has an additional argument `add` which can be used to "enrich" the output by further columns which contain, for example, the domain. The raw output does not include any information about the domain. The reporting function may be used as follows:

```
> res01 <- report(jk2.out = means, add = list(domain = "reading"))
> print(res01, digits = 4)
  group depVar    modus comparison parameter country  domain    est          p
1 LandA  score JK2.mean       <NA>      mean   LandA reading 509.76  0.000e+00
2 LandA  score JK2.mean       <NA>        sd   LandA reading  92.19 8.378e-277
3 LandB  score JK2.mean       <NA>      mean   LandB reading 472.60  0.000e+00
4 LandB  score JK2.mean       <NA>        sd   LandB reading 100.34 3.421e-205
5 LandC  score JK2.mean       <NA>      mean   LandC reading 502.46  0.000e+00
6 LandC  score JK2.mean       <NA>        sd   LandC reading  98.26 7.058e-300
     se
1 4.833
2 2.593
3 6.493
4 3.283
5 4.613
6 2.655
```

For each subpopulation denoted by the `groups` statement (here: `LandA`, `LandB` and `LandC`), each dependent variable (here: only the reading competence "score"), each parameter and each coefficient (i.e., the estimate and the corresponding standard error) the corresponding value is given. Can we see see how the results would change if we do not consider the clustered structure? Yes, we can. We simply leave out the jackknifing arguments `JKZone` and `JKrep`. The type argument then is automatically ignored likewise, whether specified or not. The results will be pooled only due to multiple imputed data:

```
> means <- jk2.mean(datL = read10, ID = "idstud", wgt = "wgt",
+          imp = "imp", groups = "country", dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
> res02 <- report(jk2.out = means, add = list(domain = "reading"))
> print(res02, digits = 4)
  group depVar     modus comparison parameter country  domain    est   p    se
1 LandA  score CONV.mean       <NA>      mean   LandA reading 509.76   0 2.686
2 LandA  score CONV.mean       <NA>        sd   LandA reading  92.15 NaN   NaN
3 LandB  score CONV.mean       <NA>      mean   LandB reading 472.60   0 2.906
4 LandB  score CONV.mean       <NA>        sd   LandB reading 100.31 NaN   NaN
5 LandC  score CONV.mean       <NA>      mean   LandC reading 502.46   0 3.031
6 LandC  score CONV.mean       <NA>        sd   LandC reading  98.22 NaN   NaN
```

We see that the means are completely unaffected, but the standard errors for the mean estimates are considerably lower. (Standard errors for standard deviations are not implemented yet.) If we decide to leave out the weights as well, we would additionally expect to receive different means now:

```
> means <- jk2.mean(datL = read10, ID = "idstud", imp = "imp", groups = "country",
+          dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
> res03 <- report(jk2.out = means, add = list(domain = "reading"))
> print(res03, digits = 4)
  group depVar     modus comparison parameter country  domain    est   p    se
1 LandA  score CONV.mean       <NA>      mean   LandA reading 508.60   0 2.700
2 LandA  score CONV.mean       <NA>        sd   LandA reading  92.28 NaN   NaN
3 LandB  score CONV.mean       <NA>      mean   LandB reading 474.87   0 2.939
4 LandB  score CONV.mean       <NA>        sd   LandB reading 100.78 NaN   NaN
5 LandC  score CONV.mean       <NA>      mean   LandC reading 503.65   0 2.857
6 LandC  score CONV.mean       <NA>        sd   LandC reading  97.41 NaN   NaN
```

If we additionally decide to ignore the imputations and treat, for example, the first plausible value as it would have been a fully observed measure of the latent competency, the function call would be the following:

```
> means <- jk2.mean(datL = subset(read10,imp==1),  ID = "idstud",
+          imp = "imp", groups = "country", dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 1 imputations.
> res04 <- report(jk2.out = means, add = list(domain = "reading"))
> print(res04, digits = 4)
  group depVar comparison parameter country  domain    est  p    se
1 LandA  score         NA      mean   LandA reading 508.71  0 2.626
2 LandA  score         NA        sd   LandA reading  91.07 NA    NA
3 LandB  score         NA      mean   LandB reading 475.42  0 2.806
4 LandB  score         NA        sd   LandB reading  99.73 NA    NA
5 LandC  score         NA      mean   LandC reading 503.54  0 2.770
6 LandC  score         NA        sd   LandC reading  97.67 NA    NA
```

The estimation of standard errors now no longer accounts for the uncertainty due to imputation. Furthermore, also the mean estimates have changed as the estimation now is based only on the first plausible value.

Two possible interesting features should be emphasized in the following. First assume that we do not have one, but two grouping variables, namely `country` and `sex`. As we have three countries and two sex values, the whole population is splitted into $3 \times 2 = 6$ subpopulations for which descriptives can be requested. If we additionally are interested in the descriptives of the whole population or the descriptives *within each country, but together for both sex groups*, we can use the `group.splits` argument to particularly specify the groups we are interested in. Let us consider for example the two grouping variables `country` and `sex`. If `group.splits` equals 2 (the default, i.e., the number of grouping variables), descriptives for the $3 \times 2 = 6$ subpopulations are computed. If `group.splits` is `1:2`, descriptives for each country (e.g., across sex) and each sex group (e.g. across countries) additionally are computed. If `group.splits` is `0:2`, descriptives also for the whole population (e.g. across sex *and* countries) are computed.

The second feature is about mean differences. Suppose you are interested in sex differences *within* each country. The grouping variable for which mean differences should be computed has to be specified in the `group.differences.by` argument. For a grouping variable with K levels, all $K!/(2! \times (K-2)!)$ comparisons are computed. It is important that the group defined in `group.differences.by` also has to occur in the `groups` statement, otherwise `group.differences.by` will be ignored. To estimate sex differences *within each country*, sex and country have to be part of the `groups` statement, whereas only sex has to be used in the `group.differences.by` argument. Both features are illustrated in the following example:

```
> means <- jk2.mean(datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = c("sex","country"),
+          group.splits = c(0,2), group.differences.by = "sex", dependent = "score")
2 analyse(s) overall according to: 'group.splits = 0 2'.

  analysis.number hierarchy.level groups.divided.by group.differences.by
1               1               0                                  <NA>
2               2               2     sex + country                   sex

Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
```

```
> res05 <- report(jk2.out = means, add = list(domain = "reading"))
> print(res05, digits = 4)
                            group depVar    modus comparison parameter
1  country=LandA____female.vs.male  score JK2.mean  groupDiff      mean
2  country=LandB____female.vs.male  score JK2.mean  groupDiff      mean
3  country=LandC____female.vs.male  score JK2.mean  groupDiff      mean
4                     female_LandA  score JK2.mean       <NA>      mean
5                     female_LandA  score JK2.mean       <NA>        sd
6                     female_LandB  score JK2.mean       <NA>      mean
7                     female_LandB  score JK2.mean       <NA>        sd
8                     female_LandC  score JK2.mean       <NA>      mean
9                     female_LandC  score JK2.mean       <NA>        sd
10                      male_LandA  score JK2.mean       <NA>      mean
11                      male_LandA  score JK2.mean       <NA>        sd
12                      male_LandB  score JK2.mean       <NA>      mean
13                      male_LandB  score JK2.mean       <NA>        sd
14                      male_LandC  score JK2.mean       <NA>      mean
15                      male_LandC  score JK2.mean       <NA>        sd
16                      wholeGroup  score JK2.mean       <NA>      mean
17                      wholeGroup  score JK2.mean       <NA>        sd
             sex country  domain       es     est          p    se
1  female.vs.male   LandA reading -0.16894 -15.446  3.903e-02 7.484
2  female.vs.male   LandB reading -0.09655  -9.694  1.682e-01 7.035
3  female.vs.male   LandC reading -0.20294 -19.847  4.592e-03 7.002
4          female   LandA reading       NA 518.091  0.000e+00 6.118
5          female   LandA reading       NA  88.410 4.241e-117 3.843
6          female   LandB reading       NA 477.369  0.000e+00 8.081
7          female   LandB reading       NA 100.967 7.475e-117 4.394
8          female   LandC reading       NA 512.126  0.000e+00 5.786
9          female   LandC reading       NA  98.329 3.861e-152 3.743
10           male   LandA reading       NA 502.644  0.000e+00 5.964
11           male   LandA reading       NA  94.750 8.862e-172 3.391
12           male   LandB reading       NA 467.674  0.000e+00 6.631
13           male   LandB reading       NA  99.518 6.050e-151 3.803
14           male   LandC reading       NA 492.278  0.000e+00 5.899
15           male   LandC reading       NA  97.213 4.867e-180 3.398
16           <NA>    <NA> reading       NA 507.681  0.000e+00 4.314
17           <NA>    <NA> reading       NA  93.266  0.000e+00 2.268
```

First note the `group.splits` is set to `c(0,2)`, which means that we request descriptives for the whole population and the 6 subpopulations. Consequently, two analyses are conducted. The `group.differences.by` only applies for the second analysis, as the gender group is not considered relating to the whole population analysis. To estimate sex differences *across all countries*, only sex has to be part of the `group` statement, and only sex has to be used in the `group.differences.by` argument. The output of the analysis is nearly the same as we would have omitted the `group.differences.by` argument, but now, some additional lines have joined. Additionally to several group columns, one column for group membership is provided. The last line labelled `wholeGroup` provides results concerning the whole population. The line labelled `male_LandC` contains values for the males in LandC. Moreover,

three mean differences were computed. In each federal state, the difference between males and females is given.

At last for this chapter, let's consider a further comparison: We see group differences according to `sex` in each country. It is plausible to assume that there are group differences also in the whole population. But: Is there any country for which the group differences differ substantially from the group differences in the whole population? To investigate this, we make an exception from the rule that `group.differences.by` must contain values which are included in `groups`: We add the term `wholePop` into the argument `group.differences.by`:

```
> means <- jk2.mean(datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = c("sex","country"),
+          group.splits = 0:2, group.differences.by = "sex", cross.differences = TRUE,
+          dependent = "score")
4 analyse(s) overall according to: 'group.splits = 0 1 2'.

  analysis.number hierarchy.level groups.divided.by group.differences.by
1               1               0                                  <NA>
2               2               1                 sex                sex
3               3               1             country               <NA>
4               4               2      sex + country                sex

Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
> res06 <- report(jk2.out = means, add = list(domain = "reading"))
```

We see in the first line of the `results` object that the group differences in the whole population are -27.39 points. Line number 2 includes the group differences in `LandA`, which amounts -34.53. The difference between both, i.e. $-34.53 - (-27.39) = -7.14$ is contained in line 3. Considering the corresponding standard error of 8.7 yields that the difference is not significant: The amount of the deviation (7.13) is less than twice the standard error of the amount.

2.2 Frequency tables

Computation of frequency tables works in the same manner as in the examples mentioned before. Representative for several possible analyses only one example is given below. Consider the `score` column we used as the dependent variable in all previous analyses. Suppose we define a cut score criterion, i.e.

all persons with at least 465 points passed, the other ones failed. The indicator variable `passReg` defines whether the "regular standard" was fulfilled or not. We now may be interested whether the percentage of pass/fails differs between countries. We henceforward consider the column `passReg` as dependent variable which is a simple indicator, i.e. a categorical variable with two categories (computation of frequency tables for variables with more than two categories are possible, too). Categorical variables are often represented as factors in R, which is quite straightforward. However, the `passReg` variable is of class numeric. This is an inconsistency which may cause annoying misinterpretations when such variables are called in functions related to the generalized linear model like `aov()`, `glm()` etc. For the computations of frequency tables it is not necessary to convert the variable class to factor.

We now are interested in the relative frequencies of this groups in the different countries and within each country for different groups of gender. As before, we want to take the cluster structure and multiple imputations into account. Moreover, we are interested whether the distribution of pass/fail is different for males vs. females within specific countries. This is done via a chi square test. The test statistic is pooled according to the clustered structure and the imputations. To call for the chi square test, we can use the `group.differences.by` argument and additionally specify `chiSquare = TRUE`:

```
> freqs <- jk2.table( datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = c("country", "sex"),
+                group.differences.by = "sex", chiSquare = TRUE, dependent = "comp")
1 analyse(s) overall according to: 'group.splits = 2'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
> res07 <- report(jk2.out = freqs, add = list(domain = "reading"))
> print(res07, digits = 4)
           group depVar     modus comparison     parameter country    sex
1   LandA_female   comp JK2.table       <NA>             1   LandA female
2   LandA_female   comp JK2.table       <NA>             2   LandA female
3   LandA_female   comp JK2.table       <NA>             3   LandA female
4   LandA_female   comp JK2.table       <NA>             4   LandA female
5   LandA_female   comp JK2.table       <NA>             5   LandA female
6     LandA_male   comp JK2.table       <NA>             1   LandA   male
7     LandA_male   comp JK2.table       <NA>             2   LandA   male
8     LandA_male   comp JK2.table       <NA>             3   LandA   male
9     LandA_male   comp JK2.table       <NA>             4   LandA   male
10    LandA_male   comp JK2.table       <NA>             5   LandA   male
11  LandB_female   comp JK2.table       <NA>             1   LandB female
12  LandB_female   comp JK2.table       <NA>             2   LandB female
13  LandB_female   comp JK2.table       <NA>             3   LandB female
14  LandB_female   comp JK2.table       <NA>             4   LandB female
```

```
15  LandB_female   comp JK2.table       <NA>             5   LandB female
16    LandB_male   comp JK2.table       <NA>             1   LandB   male
17    LandB_male   comp JK2.table       <NA>             2   LandB   male
18    LandB_male   comp JK2.table       <NA>             3   LandB   male
19    LandB_male   comp JK2.table       <NA>             4   LandB   male
20    LandB_male   comp JK2.table       <NA>             5   LandB   male
21  LandC_female   comp JK2.table       <NA>             1   LandC female
22  LandC_female   comp JK2.table       <NA>             2   LandC female
23  LandC_female   comp JK2.table       <NA>             3   LandC female
24  LandC_female   comp JK2.table       <NA>             4   LandC female
25  LandC_female   comp JK2.table       <NA>             5   LandC female
26    LandC_male   comp JK2.table       <NA>             1   LandC   male
27    LandC_male   comp JK2.table       <NA>             2   LandC   male
28    LandC_male   comp JK2.table       <NA>             3   LandC   male
29    LandC_male   comp JK2.table       <NA>             4   LandC   male
30    LandC_male   comp JK2.table       <NA>             5   LandC   male
31 country=LandA   comp JK2.table  groupDiff chiSquareTest    <NA>   <NA>
32 country=LandB   comp JK2.table  groupDiff chiSquareTest    <NA>   <NA>
33 country=LandC   comp JK2.table  groupDiff chiSquareTest    <NA>   <NA>
    domain D2statistic chi2Approx df1   df2     est         p pApprox      se
1  reading          NA         NA  NA    NA 0.08371 1.836e-05      NA 0.01954
2  reading          NA         NA  NA    NA 0.18556 1.351e-17      NA 0.02173
3  reading          NA         NA  NA    NA 0.31800 1.135e-38      NA 0.02445
4  reading          NA         NA  NA    NA 0.27691 5.565e-23      NA 0.02805
5  reading          NA         NA  NA    NA 0.13582 1.125e-11      NA 0.02000
6  reading          NA         NA  NA    NA 0.11287 9.861e-08      NA 0.02118
7  reading          NA         NA  NA    NA 0.21061 8.114e-16      NA 0.02616
8  reading          NA         NA  NA    NA 0.31579 2.516e-55      NA 0.02016
9  reading          NA         NA  NA    NA 0.25397 1.304e-23      NA 0.02536
10 reading          NA         NA  NA    NA 0.10675 1.377e-13      NA 0.01443
11 reading          NA         NA  NA    NA 0.19614 4.319e-11      NA 0.02975
12 reading          NA         NA  NA    NA 0.25409 9.950e-27      NA 0.02374
13 reading          NA         NA  NA    NA 0.26502 3.134e-15      NA 0.03361
14 reading          NA         NA  NA    NA 0.19948 2.806e-11      NA 0.02997
15 reading          NA         NA  NA    NA 0.08527 2.068e-07      NA 0.01642
16 reading          NA         NA  NA    NA 0.22956 8.146e-14      NA 0.03074
17 reading          NA         NA  NA    NA 0.25145 1.072e-17      NA 0.02935
18 reading          NA         NA  NA    NA 0.26594 1.902e-29      NA 0.02360
19 reading          NA         NA  NA    NA 0.18879 7.090e-17      NA 0.02262
20 reading          NA         NA  NA    NA 0.06427 8.691e-06      NA 0.01445
21 reading          NA         NA  NA    NA 0.12271 2.062e-09      NA 0.02048
22 reading          NA         NA  NA    NA 0.18009 8.243e-17      NA 0.02163
23 reading          NA         NA  NA    NA 0.28398 2.064e-29      NA 0.02522
24 reading          NA         NA  NA    NA 0.26840 1.805e-30      NA 0.02339
25 reading          NA         NA  NA    NA 0.14483 4.129e-19      NA 0.01621
26 reading          NA         NA  NA    NA 0.15140 1.121e-12      NA 0.02128
27 reading          NA         NA  NA    NA 0.20550 9.044e-12      NA 0.03013
28 reading          NA         NA  NA    NA 0.31705 3.239e-18      NA 0.03643
29 reading          NA         NA  NA    NA 0.23110 3.518e-17      NA 0.02742
30 reading          NA         NA  NA    NA 0.09495 2.517e-11      NA 0.01423
31 reading       1.217         NA   4 12.85      NA 3.512e-01      NA      NA
32 reading       1.097         NA   4 41.57      NA 3.706e-01      NA      NA
33 reading       2.267         NA   4 12.60      NA 1.195e-01      NA      NA
```

The reporting function `report` summarizes the results.
The output is a single data frame in the long format. To make the output

more pleasing to the eye, a short summary function `dT` is just waiting to do her job, to summarize the results. The first column refers to the groups specified in the analysis (in our example: `country` and `sex`). The next column gives the name of the dependent variable (i.e. "passReg"). The "modus" simply tells which analysis was conducted. If a specific kind of comparison was computed, the "comparison" columns gives information whether group differences, cross level differences or cross level differences of group differences were computed. The "labels" of the dependent variable now are captured in the `parameter` column. We see that in country `"LandA"` 73.1 percent of the females and 67.7 percent of the males fulfills the regular standard. The first three lines of the output indicate that in all countries the percentage of females fulfilling the standards exceeds the percentage of males. The differences, however, is non significant in each country.

Unfortunately, the results of the chi square test are not yet captured by the `report` function, so we have to extract them from the `freqs` object by ourself:

```
> options(scipen=4)
> cols <- c("group", "coefficient", "value")
> frqs <- freqs[["resT"]][["noTrend"]]
> res  <- frqs[which(frqs[,"parameter"] == "chiSquareTest"), cols ]
> wide <- reshape2::dcast(res, group~coefficient, value.var = "value")
> wide <- wide[,-grep("Approx", colnames(wide))]
> print(wide, digits = 1)
```

```
          group D2statistic df1 df2   p
1 country=LandA           1   4  13 0.4
2 country=LandB           1   4  42 0.4
3 country=LandC           2   4  13 0.1
```

In each of the three countries a chi square test was conducted separately. For `LandA`, the p is $<.001$, hence the distribution of passed/failed significantly differs between males and females in `LandA`. However, `LandB` and `LandC` reveals another picture—there are no sex differences in the rate of pass/fail. As we have imputed data, we additionally have an chi square approximation. See the help page of `micombine.chisquare` of the `miceadds` package for further details.

There is a second alternative to analyze whether the distribution of pass/fail is different for males vs. females within specific countries. With `chiSquare = FALSE` we get the differences *separately for each category* of the dependent

variable within each country. The conclusion we draw from this analysis, however, is quite equivalent to the preceding analysis:

```
> freqs <- jk2.table( datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = c("country", "sex"),
+                group.differences.by = "sex", chiSquare = FALSE, dependent = "passReg")
1 analyse(s) overall according to: 'group.splits = 2'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
```

Let's devote little attention to the problem of missing values. In the example mentioned above, it does not seem plausible to assume missing values on the `"passed"` variable. Without available data for an examinee, the case will be excluded from the data previously. But consider a questionnaire where pupils are asked about there parents' profession, for example to compute the family's highest socio-economical income (HISEI). Some examinees might have choosen the option "I don't know my parents' profession". Conceptually, it makes considerably more sense to define a separate category during the data preparation, for example "lowest HISEI", "medium HISEI", "highest HISEI", "unknown HISEI". Families without valid HISEI information then will be considered as a separate group in the analyses. Applying `jk2.table` then will give frequencies for four groups. However, if the "Dont't know" cases appear as "NA" values, `eatRep` has to know whether it should handle these values as missing or as a new distinct category named "Dont't know" (or whatever), for which relative frequencies also can be computed. Only for illustration, let us generate some missing values in some of the imputations and repeat the analysis subsequently. You will see that a new category has joined to the output, which is labelled `"<NA>"`.

```
> read10[,"passedNA"] <- read10[,"passReg"]
> read10[ sample(nrow(read10), 100, FALSE) ,"passedNA"]   <- NA
> freqs2<- jk2.table( datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = c("country", "sex"),
+                dependent = "passedNA", separate.missing.indicator = TRUE)
1 analyse(s) overall according to: 'group.splits = 2'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
```

2.3 Quantiles

Estimation of quantiles for numerical variables is possible using the function `jk2.quantile`. All related analyses mentioned up to this point apply in the

same way. Note that these analyses apply for numerical dependent variables. See the examples in the help file of `jk2.quantile()`.

3 Generalized linear models

Considering multiple imputations and clustered structure in the estimation of generalized linear models is based on the same principles as aforementioned. However, some additional comments due to specific characteristics of regression models have to be made. First we now have another type of variable—independent variables, which may occur as multiple imputed variables, too. Second, we have to specify the regression expression, as in `glm()`, for example. Third, we will have to specify the kind of regression we propose to estimate, for example linear or logistic regression. We start with a simple example using the same data as before.

```
> mod1  <- jk2.glm(datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp", groups = "country",
+          formula = score~sex*ses, family=gaussian(link="identity"), poolMethod = "scalar")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
```

As we might have expected, the outcome is a single data frame in the long format. And long really means long! For our purpose, it may be sufficient to content ourself with the summary provided by `dG`. But beforehand let us consider how many regression analyses are conducted and how many results we expect to find. The message on the console speaks of about "1 analysis overall" according to `group.splits = 1`. But strictly speaking, we have estimated three regression analyses, as the model is fitted in each group (i.e., in each country) separately. As we have specified one grouping variable dividing the data into three distinct groups (according to `country`) for which we have instructed `jk2.glm()` to fit the regression model separately, we find results of the three models in the results. More specifically, for each country, an intercept and three regression coefficients according to `gender`, `INCOME` and their interaction are estimated. The `dG()` function allows us to have a look only at a specific result out of the 3 analyses. `analyses = 1:2` advises the function to display the results of the first and second analysis. First we should consider that each single analyses is characterized by two variables,

the group for which the model is fitted, and the dependent variable. In the heading we find information about both. The actual regression results are displayed underneath.

```
> res   <- report(mod1, printGlm = TRUE)
       Trend group: 'noTrend'.
            groups: country = LandA
dependent Variable: score

    parameter     est     se t.value p.value
1 (Intercept) 438.021 11.907  36.786   0.000
2         ses   1.444  0.187   7.709   0.000
3     sexmale -29.416 16.544  -1.778   0.076
4 sexmale:ses   0.269  0.248   1.085   0.278

            R-squared: 0.134; SE(R-squared): 0.001
Nagelkerkes R-squared: 0.61; SE(Nagelkerkes R-squared): 0.004
1203 observations and 1199 degrees of freedom.
------------------------------------------------------------------
            groups: country = LandB
dependent Variable: score

    parameter     est     se t.value p.value
1 (Intercept) 367.323 13.060  28.126   0.000
2         ses   2.255  0.241   9.358   0.000
3     sexmale  -4.027 16.196  -0.249   0.804
4 sexmale:ses  -0.167  0.338  -0.495   0.621

            R-squared: 0.222; SE(R-squared): 0
Nagelkerkes R-squared: 0.832; SE(Nagelkerkes R-squared): 0.001
1263 observations and 1259 degrees of freedom.
------------------------------------------------------------------
            groups: country = LandC
dependent Variable: score

    parameter     est     se t.value p.value
1 (Intercept) 413.788 13.914  29.738   0.000
2         ses   2.081  0.250   8.323   0.000
3     sexmale -12.438 16.359  -0.760   0.447
4 sexmale:ses  -0.238  0.301  -0.791   0.429

            R-squared: 0.169; SE(R-squared): 0
Nagelkerkes R-squared: 0.727; SE(Nagelkerkes R-squared): 0.004
1243 observations and 1239 degrees of freedom.
```

Remember what was said about factors in the chapter about frequency tables: The gender variable now has to be defined explicitly to be of class factor! Otherwise, albeit gender variable may be coded as 0/1, it would be treated to be a continuous numeric variable. With only two levels—male and female—this may have no effect on the results, but consider a factor variable with three levels, which may be coded 0, 1 and 2. We are interested in two

coefficients which correspond to the effect of level 1 vs. level 0 and the effect of level 2 vs. level 0. If we refrain from defining the variable to be of class factor, only one coefficient is computed, and the variable is assumed to be continuous. What we see additionally is that R implicitly defined the female group to be the reference—the regression parameter was labelled `sexmale`.

Now we try something different. First we define `"passed"` to be our dependent variable. This leads to a binomial regression model which models whether the probability of pass/fail depends on certain independent variables. Secondly, we also use country as a predictor (instead of a grouping variable). This is to test whether the effect of sex varies across countries. To simplify displaying the results, we use the same workaround as in the example before.

```
> mod1  <- jk2.glm(datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp",
+          formula = passReg~country*sex, family=binomial(link="logit") )
1 analyse(s) overall according to: 'group.splits = 0'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
> res   <- report(mod1, printGlm = TRUE)
       Trend group: 'noTrend'.
            groups:
dependent Variable: passReg

             parameter    est    se t.value p.value
1          (Intercept)  1.000 0.170   5.891   0.000
2         countryLandB -0.800 0.256  -3.121   0.002
3 countryLandB:sexmale  0.138 0.267   0.517   0.605
4         countryLandC -0.165 0.199  -0.828   0.408
5 countryLandC:sexmale  0.016 0.223   0.073   0.941
6               sexmale -0.262 0.203  -1.287   0.198

           R-squared: 0.026; SE(R-squared): NA
Nagelkerkes R-squared: 0.007; SE(Nagelkerkes R-squared): NA
3709 observations and 3703 degrees of freedom.
```

Inspecting the output, we found that the probability of success significantly depends on the country an examinee stems from and on an examinee's sex. The probability of passing the test is significantly lower for males and for examinees who stem from `"LandB"`. Examinees who stem from `"LandC"` do not significanty differ in their probability of passing the test from examinees who stem from the reference country, `"LandA"`. Moreover, the disadvantage of boys is not consistent across countries: In `"LandB"`, this difference is significantly less substantial.

Please note that—although we have only defined one independent variable—we obtain two regression coefficients for the two categories of the country variable. Again, R choosed its favorite reference group by itself. The effects are expressed in relation to `LandA`. To interpretate the effects, the coefficients may be transformed to odds ratios:

```
> #exp(mod1[c(1,3,5,7,9),"value"])
```

In `LandB` the odds ratio to pass is 0.58 times the corresponding odds ratio in `LandA`. The following subsections address two little questions one might ask oneself.

3.1 How to change reference group at costumer's option

As we saw in the preceding section, R choosed the reference group of factor variables by itself. Persuading R to meet our needs is easier said than done. The essentially easiest way is to redefine the factor variable and choose its levels manually. We will demonstrate this procedure about the gender variable in our fictitious data set. Remember the first example in section 3—R choosed the females to be the reference. Why? Simply because "female" comes before "male" in the alphabet. Let's redefine the gender variable:

```
> read10[,"sexRecoded"] <- factor(read10[,"sex"], levels = c("male", "female") )
```

The simple intervention provokes R to use the first label mentioned in the `levels`-argument as reference group when repeating the last example:

```
> mod1  <- jk2.glm(datL = read10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", imp = "imp",
+          formula = passReg~country*sexRecoded, family=binomial(link="logit") )
1 analyse(s) overall according to: 'group.splits = 0'.
Assume unnested structure with 3 imputations.
Create 62 replicate weights according to JK2 procedure.
> res   <- report(mod1, printGlm = TRUE)
       Trend group: 'noTrend'.
            groups:
dependent Variable: passReg

                     parameter    est    se t.value p.value
1                  (Intercept)  0.738 0.126   5.865   0.000
2                countryLandB -0.662 0.168  -3.935   0.000
```

```
3 countryLandB:sexRecodedfemale -0.138 0.267  -0.517   0.605
4                  countryLandC -0.149 0.177  -0.838   0.402
5 countryLandC:sexRecodedfemale -0.016 0.223  -0.073   0.941
6              sexRecodedfemale  0.262 0.203   1.287   0.198

           R-squared: 0.026; SE(R-squared): NA
Nagelkerkes R-squared: 0.007; SE(Nagelkerkes R-squared): NA
3709 observations and 3703 degrees of freedom.
```

3.2 Which of both determination coefficients should I pay attention?

The output of each `jk2.glm()` analysis also contains the pooled determination coefficient, R^2 and Nagelkerke's R^2, which may be considered as a pseudo-R^2 for log-linear regression models. In linear regression models, i.e. if the identity link is used and normally distributed errors are assumed, the conventional R^2 should be used to interpret explained variance. In log-linear regression models, i.e. if the binomial link function is used, Nagelkerke's R^2 may be used instead.

4 Nested imputations

The next to last chapter of this little tutorial is reserved to the problem of nested imputation. The general concept is described in Rubin (2003). At this point, only some specific aspects which are relevant in large scale assessments, are mentioned briefly. Suppose you want to estimate IRT proficiencies (often denoted θ) in a specific domain. Applying an extensive marginal model which comprehends of item responses and background information as well, the posterior distribution of each examinees' θ is specified. Without any certain proficiency value of a specific examinee—remember that θ is considered to be latent, i.e. an inherently missing variable—, plausible values are drawn from the posterior of each examinee. Conceptually, plausible values are multiple imputations of the missing variable θ and may analyzed in standard statistic procedures. To obtain valid estimates and standard errors, the results have to be pooled according to Rubin (1987).

Suppose you have missing data in the background variables as well, which have to be imputed in the first step, which may result in $M = 5$ data sets. For each data set a marginal IRT model is specified and $N = 20$ plausible values

are drawn. Overall $5 \times 20 = 100$ plausible values in a dependency structure will result from the analysis. Formally, we now have nested imputed data. To pool the results, the formulas in Rubin (1987) cannot be applied, as the plausible values do not stem from a common 'nest'. The interdependence has to be taken into account. Whereas the conventional pooling formulas split the overall variance in the variance within imputation and the variance between imputation (where the latter one is used to estimate the uncertainty due to imputation), the formulas for nested imputation extend the old ones by splitting the variance between imputation in the within-nest variance and the variance between nests. See Rubin (2003) for further details. These varied formulas are also implemented in `eatRep`.

If the data analysed with `eatRep` stem from a nested multiple imputation structure, this structure has to be specified. More specifically, the structure has to be represented in the long-format data frame. `eatRep` has to know the number of nests and the number of imputations in each nest. The above procedure sounds more complicated than it hopefully is.

4.1 Example: Compute descriptives from a nested imputation structure

At the beginning of this little tutorial, we have created a subset of our data set which was used for all analyses so far. Now it's time to consider the whole data set. The variable `"nest"` denotes the nest or first-stage imputation variable. As we only have two nests, only two imputations were created in the first step. Within each of this two imputations, three plausible values were drawn from the marginal (or conditioning) model. Hence, we would expect that the plausible values (captured in column `"score"`) vary between nests *and* between imputations, whereas the conditioning variables (e.g. income) only vary between nests, but not between imputations within each nest! To date, the `eatRep` does not provide any consistency checks whether this requirements are fulfilled.

All analyses specified so far treated 3 imputations. Considering the nested structure now comprises $3 \times 2 = 6$ imputations. For the purpose of illutration, we repeat our very first example, using nested imputations now:

```
> read      <- subset(lsa, domain == "reading")
> readN1.10 <- subset(read, year == 2010 )
> means <- jk2.mean(datL = readN1.10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", nest="nest", imp = "imp",
+          groups = "country", dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume nested structure with 2 nests and 3 imputations in each nest. This will result in 2 x 3 = 6 imputa
Create 62 replicate weights according to JK2 procedure.
> res   <- report(means)
```

The only thing we have to change is that we use now the whole data and additionally specify the variable which denotes the "nests".

4.2 Example: Fit a linear regression model in a nested imputation structure

The principles of considering the nested structure are quite the same as in the preceding example. We now want to predict "reading ability" by `sex` and `income`. Using `country` as group variable likewise allows for investigating whether the potential effects vary across countries.

```
> mod1  <- jk2.glm(datL = readN1.10, ID = "idstud", wgt = "wgt", type = "JK2",
+          PSU = "jkzone", repInd = "jkrep", nest="nest", imp = "imp",
+          groups = "country", formula = score~sex+ses, family=gaussian(link="identity") )
Method 'mice' is not available for nested imputation. Switch to method 'scalar'.
1 analyse(s) overall according to: 'group.splits = 1'.
Assume nested structure with 2 nests and 3 imputations in each nest. This will result in 2 x 3 = 6 imputa
Create 62 replicate weights according to JK2 procedure.
> res   <- report(mod1)
```

References

Douglas A. Luke. *Multilevel modeling*. Sage, Thousand Oaks, CA, 2009.

Thomas Lumley. Analysis of complex survey samples. *Journal of Statistical Software*, 9(1):1–19, 2004.

Thomas Lumley. *Survey: analysis of complex survey samples*. R package version 3.28-2, 2012.

Donald B. Rubin. *Multiple imputation for no nonresponse in surveys*. Wiley, New York, 1987.

Donald B. Rubin. Nested multiple imputation of nmes via partially incompatible mcmc. *Statistica Neerlandica*, 51(1):3–18, 2003.

Westat. *WesVar*. Westat, Rockville, MD, 2000.