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

Sebastian Weirich
Humboldt University Berlin, Germany

September 4, 2014

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, only the 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 implemented in `eatRep` yet, for example bootstrapping methods or even JK1. 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 or linear logistic regression models.

`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 imputed 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 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.

```
> str(reading)
'data.frame':        27714 obs. of  11 variables:
 $ idstud    : chr  "LandA01010401" "LandA01010401" "LandA01010401" "LandA01010401" ...
 $ wgtSTUD   : num  60 60 60 60 60 ...
 $ sex       : Factor w/ 2 levels "female","male": 1 1 1 1 1 1 2 2 2 2 ...
 $ country   : Factor w/ 3 levels "LandA","LandB",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ JKZone    : num  40 40 40 40 40 40 40 40 40 40 ...
 $ JKrep     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ income    : num  2136 2136 2136 2154 2154 ...
 $ imputation: Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1 ...
 $ nest      : Factor w/ 2 levels "1","2": 2 2 2 1 1 1 2 2 2 1 ...
 $ score     : num  636 707 672 631 708 ...
 $ passed    : num  1 1 1 1 1 1 1 1 1 1 ...
```

Requesting the data structure provides us with information about the number and type of variables and the number of examinees. `"idstud"` is a person identifier for 4,619 distinct examinees, `"wgtSTUD"` a person weight, `"sex"` denotes each person's sex, `"country"` denotes the country the person comes from. `"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. `"income"` is each person's financial income. The next two variables, `"nest"` and `"imputation"` 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 princiles of nested imputations will be elucidated later; for the moment, we may content ourself with data from the first nest only, i.e. wo split the data and only cosider cases for which $nest = 1$. We may think of `"score"` as the the plausible value estimate for the reading competence. Hence, if $nest = 1$ and $imputation = 3$, the value in the `"score"` columns refers to the third plausible value in the first nest for the reading competence. Please note that the three imputations of the reading competence occur as one variable in the data set. 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, 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 analysis automatically. Even in large data sets this takes only a few seconds. First we create a subset which only contains data from the first nest. The analysis then is conducted with this subset.

```
> readN1<- subset(reading, nest == 1 )
> means <- jk2.mean(datL = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation", groups = "country",
+          dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
Create 81 replicate weights.
...
```

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 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 81 replicate weights which are created due to 81 distinct jackknifing zones in the JKZone variable. This information refers to the "cluster replicates" and implies that the subsequent analysis has to be repeated 81 times *for each imputation*. Overall, $3 \times 81 = 243$ analyses are run overall.
In each of the 81 replication samples, one unit (e.g. school) of a certain jackknifing zone is missing and the weights of the other unit of the same zone are

doubled. The data in all other zones remain unchanged. The analyses are repeated 81 times, *using the same plausible value* as the dependent variable. Only the weights vary between the "cluster replicates": each of the 81 replicates once a time is used as the weighting variable. Each of the 81 analysis revealed slighty different results. This variation is used to estimate the sampling variance which then is used to compute the standard error, which is pooled across 81 "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 81 analyses due to the 81 "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 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.

```
> means[c(1:4,19:20),]
   group depVar     modus parameter coefficient        value country
1  LandA  score noch_leer    Ncases         est 1.187318e+05   LandA
2  LandA  score noch_leer    Ncases          se 1.726642e+03   LandA
3  LandB  score noch_leer    Ncases         est 5.398038e+04   LandB
4  LandB  score noch_leer    Ncases          se 1.135124e+03   LandB
19 LandA  score noch_leer        sd         est 1.041336e+02   LandA
20 LandA  score noch_leer        sd          se 2.071056e+00   LandA
```

The output is a data frame in the long format with 30 rows and at least six columns. To keep the overview, only a few selected rows are displayed here. For each subpopulation denoted by the `groups` statement (here: `LandA`, `LandB` and `LandC`), each dependent variable (here: only the reading competence), each parameter (we requested mean, variance, standard deviation and sample size or population size) and each coefficient (i.e., the estimate and the corresponding standard error) the corresponding value is given. However, the output rather is appropriate for further processing as for clear arragement, as the values are displayed in the long format as well as in exponential notation, for example. To display the results in the more common wide format, use the `reshape2` package. A possible call would be `reshape2::dcast(means, group ~ parameter+coefficient, value.var = "value")`. Alternatively, an

abbreviated display of the results is provided by the `dM` function (whereas dM stands for "display means"). You may think of `dM` as a simple summary function which is not intended for saving results or further processing as the results are displayed in an abbreviated (i.e., rounded) manner to offer clear arrangement on console. The `dM` function has an additional argument to omit displaying parameters or coefficients you are not interested at the moment.

```
> dM(means, omitTerms = c("var", "Ncases","NcasesValid", "meanGroupDiff") )
  group depVar mean_est mean_se  sd_est sd_se country
1 LandA  score  515.749   5.320 104.134 2.071   LandA
2 LandB  score  492.773   5.627 103.384 4.199   LandB
3 LandC  score  511.811   4.030 103.330 3.080   LandC
```

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 = readN1, ID = "idstud", wgt = "wgtSTUD",
+          imp = "imputation", groups = "country", dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
> dM(means, omitTerms = c("var", "Ncases","NcasesValid", "meanGroupDiff") )
  group depVar mean_est mean_se  sd_est sd_se country
1 LandA  score  515.749   2.665 101.224   NaN   LandA
2 LandB  score  492.773   3.218 101.277   NaN   LandB
3 LandC  score  511.811   2.745 100.405   NaN   LandC
```

We see that the means are completely unaffected, but the standard deviation now is lower. Consequently, also the standard errors for the mean estimates are considerably lower. (Standard errors for standard deviations and for variances 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 = readN1, ID = "idstud",
+          imp = "imputation", groups = "country", dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
> dM(means, omitTerms = c("var", "Ncases","NcasesValid", "meanGroupDiff") )
  group depVar mean_est mean_se  sd_est sd_se country
1 LandA  score  518.783   2.810 101.224   NaN   LandA
2 LandB  score  493.894   2.940 101.277   NaN   LandB
3 LandC  score  514.113   2.775 100.405   NaN   LandC
```

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

```
> means <- jk2.mean(datL = subset(readN1,imputation==1),  ID = "idstud",
+          imp = "imputation", groups = "country", dependent = "score")
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 1 imputations.
> dM(means, omitTerms = c("var", "Ncases","NcasesValid", "meanGroupDiff") )
  group depVar mean_est mean_se  sd_est country
1 LandA  score  520.085   2.506 102.088   LandA
2 LandB  score  492.841   2.528 100.262   LandB
3 LandC  score  514.523   2.684  99.963   LandC
```

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 group 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 $K!/(2!*(K-2)!)$ 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 = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation", 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'.
Assume unnested structure with 3 imputations.
Create 81 replicate weights.
......
```

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. Again, we may use the `dM` function to display the part of the results we are interested in—note that now we do *not* exclude `meanGroupDiff` from the results to summarize:

```
> dM(means, omitTerms = c("var","Ncases", "NcasesValid"))
                            group depVar mean_est mean_se meanGroupDiff_est
1  country=LandA____female.vs.male  score       NA      NA           -34.529
2  country=LandB____female.vs.male  score       NA      NA           -15.307
3  country=LandC____female.vs.male  score       NA      NA           -19.671
4                     female_LandA  score  534.061   6.030                NA
5                     female_LandB  score  500.607   6.071                NA
6                     female_LandC  score  521.785   4.849                NA
7                       male_LandA  score  499.532   6.799                NA
8                       male_LandB  score  485.300   6.820                NA
9                       male_LandC  score  502.114   5.623                NA
10                      wholeGroup  score  508.857   3.675                NA
   meanGroupDiff_se  sd_est sd_se    sex country
1             7.232      NA    NA   <NA>    <NA>
2             6.318      NA    NA   <NA>    <NA>
3             6.872      NA    NA   <NA>    <NA>
4                NA  99.255 3.744 female   LandA
5                NA  98.768 4.378 female   LandB
6                NA 100.097 4.166 female   LandC
7                NA 105.682 3.866   male   LandA
8                NA 107.116 5.420   male   LandB
9                NA 105.533 3.596   male   LandC
10               NA 104.336 1.702   <NA>    <NA>
```

The output now changed slightly: 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 `LandC_male` 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.

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 500 points passed, the other ones failed. We now may be interested whether the percentage of pass/fails differs between countries. We henceforward consider the column `passed` as dependent variable which is a simple indicator, i.e. a categorical variable. Categorical variables are often represented as factors in R, which is quite straightforward. However, the `"passed"` 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.

```
> freqs <- jk2.table( datL = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation", groups = c("country", "sex"),
+                dependent = "passed")
1 analyse(s) overall according to: 'group.splits = 2'.
Assume unnested structure with 3 imputations.
Create 81 replicate weights.
...
> dT(freqs)
         group depVar 0_est  0_se 1_est  1_se country    sex
1 LandA_female passed 0.358 0.024 0.642 0.024   LandA female
2   LandA_male passed 0.478 0.031 0.522 0.031   LandA   male
```

```
3 LandB_female passed 0.493 0.027 0.507 0.027   LandB female
4   LandB_male passed 0.534 0.026 0.466 0.026   LandB   male
5 LandC_female passed 0.400 0.028 0.600 0.028   LandC female
6   LandC_male passed 0.458 0.032 0.542 0.032   LandC   male
```

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. "passed"). The "labels" of the dependent variable now are captured in the column names of the summary table. We see that in country `"LandA"` 35.8 percent of the females and 47.8 percent of the males *do not pass*. As in the examples mentioned before, these analyses may be conducted without considering clustered structure. See whether the standard errors will change.

Let's devote little attention to the problem of missing values. In the example mentioned above, it does not seem plausible to assume missnig 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, it is possible to define them as a distince category for which relative frequencies 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 `"missing"`.

```
> readN1[,"passedNA"] <- readN1[,"passed"]
> readN1[ sample(nrow(readN1), 100, FALSE) ,"passedNA"]   <- NA
> freqs2<- jk2.table( datL = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation", 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 81 replicate weights.
...
> dT(freqs2)
         group   depVar 0_est  0_se 1_est  1_se <NA>_est <NA>_se country    sex
1 LandA_female passedNA 0.357 0.024 0.637 0.023    0.006   0.006   LandA female
2   LandA_male passedNA 0.476 0.031 0.518 0.031    0.007   0.003   LandA   male
3 LandB_female passedNA 0.487 0.030 0.505 0.027    0.009   0.007   LandB female
4   LandB_male passedNA 0.529 0.026 0.464 0.028    0.007   0.006   LandB   male
5 LandC_female passedNA 0.396 0.026 0.595 0.027    0.009   0.005   LandC female
6   LandC_male passedNA 0.455 0.032 0.538 0.032    0.007   0.003   LandC   male
```

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 principes mentioned before. 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 = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation", groups = "country",
+          formula = score~sex*income, family=gaussian(link="identity") )
1 analyse(s) overall according to: 'group.splits = 1'.
Assume unnested structure with 3 imputations.
Create 81 replicate weights.
...
```

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 two `'group.splits = 1'`. But strictly speaking, we have estimated three regression analyses, as the model is fitted in each group separately. As we specified one group variable dividing the data into three distinct groups, for which we instruct `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 two regression coefficients according to `gender` and `INCOME` 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.

```
> dG(mod1, analyses = 1:2)
            groups: country = LandA
dependent Variable: score

       parameter      est     se t.value p.value
1    (Intercept)  -22.476 67.101  -0.335   0.738
2         income    0.276  0.032   8.509   0.000
3        sexmale -171.308 85.158  -2.012   0.044
4 sexmale:income    0.069  0.042   1.667   0.096

            R-squared: 0.155; SE(R-squared): 0
Nagelkerkes R-squared: 0.614; SE(Nagelkerkes R-squared): 0
1659 observations and 1655 degrees of freedom.
------------------------------------------------------------------
            groups: country = LandB
dependent Variable: score

       parameter      est      se t.value p.value
1    (Intercept)  -40.461  86.390  -0.468   0.640
2         income    0.270   0.042   6.447   0.000
3        sexmale -112.739 111.276  -1.013   0.311
4 sexmale:income    0.048   0.055   0.879   0.380

            R-squared: 0.113; SE(R-squared): 0
Nagelkerkes R-squared: 0.521; SE(Nagelkerkes R-squared): 0
1573 observations and 1569 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 miss to define 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 `gendermale`.

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 independetn 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 = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation",
+          formula = passed~country*sex, family=binomial(link="logit") )
1 analyse(s) overall according to: 'group.splits = 0'.
Assume unnested structure with 3 imputations.
Create 81 replicate weights.
...
> dG(mod1)
            groups:
dependent Variable: passed

             parameter    est    se t.value p.value
1          (Intercept)  0.582 0.103   5.655   0.000
2         countryLandB -0.553 0.148  -3.728   0.000
3 countryLandB:sexmale  0.330 0.155   2.137   0.033
4         countryLandC -0.178 0.163  -1.091   0.275
5 countryLandC:sexmale  0.260 0.223   1.166   0.244
6               sexmale -0.495 0.127  -3.886   0.000

            R-squared: 0.014; SE(R-squared): 0
Nagelkerkes R-squared: 0.012; SE(Nagelkerkes R-squared): 0
4619 observations and 4613 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"`. 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"])
[1] 1.789915      Inf      Inf 1.014515 1.012512
```

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 a rather dummy method: recode the variable so that your favourite reference group occurs at first when ordered alphabetically. 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 use the `recode()` function from the `car` package to define a new variable `sexRecoded`:

```
> readN1[,"sexRecoded"] <- car::recode(readN1[,"sex"], "'male' = '_male'")
```

The simple trick of adding a `"_"` sign to the `"male"` label provokes R to sort `"_male"` before `"female"` alphabetically. Consequently, `"male"` is used as reference group when repeating the last example:

```
> mod1  <- jk2.glm(datL = readN1, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", imp = "imputation",
+          formula = passed~country*sexRecoded, family=binomial(link="logit") )
1 analyse(s) overall according to: 'group.splits = 0'.
Assume unnested structure with 3 imputations.
Create 81 replicate weights.
...
> dG(mod1)
```

```
           groups:
dependent Variable: passed

                      parameter    est    se t.value p.value
1                   (Intercept)  0.087 0.123   0.707   0.480
2                  countryLandB -0.223 0.172  -1.298   0.195
3 countryLandB:sexRecodedfemale -0.330 0.155  -2.137   0.033
4                  countryLandC  0.082 0.153   0.536   0.592
5 countryLandC:sexRecodedfemale -0.260 0.223  -1.166   0.244
6              sexRecodedfemale  0.495 0.127   3.886   0.000

            R-squared: 0.014; SE(R-squared): 0
Nagelkerkes R-squared: 0.012; SE(Nagelkerkes R-squared): 0
4619 observations and 4613 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, most frequently the conventional R^2 and Nagelkerke's R^2. However, in linear regression models, i.e. if the identity link is used, assuming normally distributed errors, 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 should be used. The reason for reporting both coefficients is the programming disability of the package developer.

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. Weirich et al. (in press) described the same procedure more elaboratively. Suppose you want to estimate IRT proficiencies (often denoted θ) in a specific domain. Applying an extensive marginal model which comprehends of items responses and background information as well, the posterior distribution of each examinees' θ is specified. Without any certain proficiency value of a specific examinee, plausible values are drawn from the posterior of each examinee. Conceptually, plausible values are multiple imputations of the inherently missing variable θ and may analyzed in standard statistic procedures according to the generalized linear model. To obtain valid estimates and standard errors, the results have to be pooled ac-

cording 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 imputation 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 with each nest! To date, the `eatRep` does not provide any consistency checks whether this re-

quirements 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:

```
> means <- jk2.mean(datL = reading, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", nest="nest", imp = "imputation",
+          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 81 replicate weights.
......
> dM(means, omitTerms = c("var", "Ncases", "NcasesValid", "meanGroupDiff"))
  group depVar mean_est mean_se  sd_est sd_se country
1 LandA  score  513.910   6.425 102.399 3.669   LandA
2 LandB  score  492.774   5.887 104.842 5.114   LandB
3 LandC  score  499.787  21.244 105.895 5.363   LandC
```

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 = reading, ID = "idstud", wgt = "wgtSTUD", type = "JK2",
+          PSU = "JKZone", repInd = "JKrep", nest="nest", imp = "imputation",
+          groups = "country", formula = score~sex+income, family=gaussian(link="identity") )
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 81 replicate weights.
......
> dG(mod1)
            groups: country = LandA
dependent Variable: score

    parameter     est     se t.value p.value
1 (Intercept) -82.900 57.286  -1.447   0.148
2      income   0.302  0.032   9.507   0.000
```

```
3     sexmale -16.524 26.841  -0.616   0.538

            R-squared: 0.134; SE(R-squared): 0.001
Nagelkerkes R-squared: 0.548; SE(Nagelkerkes R-squared): 0.013
1659 observations and 1656 degrees of freedom.
------------------------------------------------------------------
            groups: country = LandB
dependent Variable: score

    parameter      est     se t.value p.value
1 (Intercept) -100.122 74.154  -1.350   0.177
2      income    0.298  0.035   8.415   0.000
3     sexmale   -7.712 15.234  -0.506   0.613

            R-squared: 0.11; SE(R-squared): 0
Nagelkerkes R-squared: 0.52; SE(Nagelkerkes R-squared): 0
1573 observations and 1570 degrees of freedom.
------------------------------------------------------------------
            groups: country = LandC
dependent Variable: score

    parameter     est     se t.value p.value
1 (Intercept) -24.751 68.653  -0.361   0.719
2      income   0.267  0.040   6.734   0.000
3     sexmale  -9.901 23.106  -0.429   0.668

            R-squared: 0.081; SE(R-squared): 0.001
Nagelkerkes R-squared: 0.462; SE(Nagelkerkes R-squared): 0.005
1387 observations and 1384 degrees of freedom.
```

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.