Comparison of DemoDiff Releases
3.0 and 3.1

Rainer Walke (walke@demogr.mpg.de)

For additional material see www.demogr.mpg.de/tr/
Comparison of DemoDiff Releases 3.0 and 3.1

Rainer Walke, MPIDF Rostock*

2014-Mar-03

Abstract

In this Technical Report, further results of the program compareFinRaw are presented. compareFinRaw is a tool that is particularly useful for comparing large data sets. It is typically used to compare different releases of the same data. In this example, we compare two different releases of the project DemoDiff. To be more precise, we compare the nine revised files of DemoDiff release 3.1 with the older release 3.0 files. It should provide users of the DemoDiff data a comprehensive overview on all manipulations of the data that have occurred from release 3.0 to release 3.1.

Keywords  data analysis, data comparability, data evaluation, data processing, software

Contents

1. Introduction 2
2. Classification 3
3. Interpretation of summary tables 4
   3.1. compare a dataset with itself 4
   3.2. compare a dataset with itself but change almost all variable names 5
   3.3. compare a data set with an almost empty data set 5
   3.4. add one or two test variables to the almost empty data set 6
4. Results and Summary 6
   A. anchor1 8
   A.1. no variation 8
   A.2. variation, but no bijective mapping 10
   A.3. variation and bijective mapping 12
   A.4. comparison summary for anchor1_DD.dta release 3.0 and release 3.1 14
   A.5. selected in-depth comparison 14
   B. anchor2 18
   B.1. no variation 18
   B.2. variation, but no bijective mapping 21
   B.3. variation and bijective mapping 22
   B.4. comparison summary for anchor2_DD.dta release 3.0 and release 3.1 25
   B.5. selected in-depth comparison 25

*MPIDF, Konrad-Zuse-Straße 1, D-18057 Rostock, Germany. E-Mail: walke@demogr.mpg.de
1. Introduction

In a Technical Report by Walke and Müller [TR-2012-003] we have described a procedure to compare two datasets with little conditions. Only the ID had to be the same in each row of both datasets. The program that compares the datasets had been named compareFinRaw. It compares each column (variable) from one set with each column from the second data set. It furthermore checks whether there are bijective mappings
between variables of the two data sets. If there is no direct mapping it computes how much the levels of the variables have to be changed to get a bijective mapping.

In an earlier report [TR-2013-001] we used this technique to compare two earlier releases of the project DemoDiff. In that report we have compared release 2.0 [DemoDiff 2.0] and release 3.0 [DemoDiff 3.0].

In this report, we use compareFinRaw to compare again two different releases of the project DemoDiff. We are comparing release 3.0 [DemoDiff 3.0] and release 3.1 [DemoDiff 3.1]. Both are available by GESIS (www.gesis.org). We compare the following data sets: anchor1_DD.dta, anchor2_DD.dta, anchor4_DD.dta, partner1_DD.dta, partner2_DD.dta, partner4_DD.dta, biopart.dta, biochild.dta and weights.dta.

To make this report readable without having to read the earlier report [TR-2013-001] we have copied some parts of the older report into this document.

We are using the statistical package R [R 3.1] for all the computations and RStudio, knitR and MiKTeX for the documentation. These programs produce an annotated output. The output, for example, reports the number and the names of variables that completely match in both releases, i.e. have the same name and same content. The output also shows all modified variables such as renamed variables, bijective recoded variables and variables with marked differences. Furthermore the output reports the names of all variables without variation at all. We hope that this material helps users of DemoDiff to get a clear picture of what has been changed between the two releases. It should be noted that, depending on the number of variables, compareFinRaw takes minutes or hours to compare the files.\(^1\)

2. Classification

The basic principle is to classify all variables in data set \(A\) in respect to their relations to variables from data set \(B\). This classification is the same as in the predecessor report [TR-2013-001]. It contains the following classifications (black) and categories (dark green):

1. The selected variable from \(A\) does not have variation.
   1.1 \(x\) No variable with the same name is available in data set \(B\).
   1.2 One variable with the same name is available in data set \(B\).
   1.21 \(n\) Both variables are not identical.
   1.22 \(ni\) Both variables are identical.

2. The selected variable from \(A\) does have variation.
   2.1 There exists no bivariate mapping to one of the variables in data set \(B\).
   2.11 \(v\) No variable with the same name is available in data set \(B\).
   2.12 \(vn\) One variable with the same name is available in data set \(B\).
   2.2 There exists at least one bivariate mapping to one of the variables in data set \(B\).
   2.21 There are more than one bivariate mappings.
   2.211 All mapped variables from \(B\) do have a different name.
   2.2111 \(vh\) No mapped variable from \(B\) is identical.
   2.2112 \(vhi\) At least one mapped variable from \(B\) is identical.
   2.212 One mapped variable from \(B\) has the same name.
   2.2121 \(vbn\) The mapped variable from \(B\) is not identical.
   2.2122 \(vbni\) The mapped variable from \(B\) is identical.
   2.22 There exists exactly one bivariate mapping.

\(^1\)In our case, it took six hours to run the program for the largest data set (anchor4_DD.dta)
2.221 The mapped variable from B has a different name.

2.2211 [vb1] Both variables are not identical.

2.2212 [vb1i] Both variables are identical.

2.222 The mapped variable from B has the same name.

2.2221 [vb1n] Both variables are not identical.

2.2222 [vb1ni] Both variables are identical.

Each variable out of A will be assigned to exactly one of these 13 dark green marked categories. Figure 1 illustrates this classification scheme. The endpoints are exactly these 13 categories again. The words at the branchings give short hints for the classification criteria.

![Classification tree](image)

Figure 1: Classification tree.

3. Interpretation of summary tables

In this section we discuss selected output to get an idea on the strengths and limitations of the classification scheme.

3.1. compare a dataset with itself

If we compare a dataset with itself only the classes ni, vbni and vb1ni are filled. As an example we have compared the data set anchor2_dd.dta from release 3.0 with itself. It gives the results shown below.

\[\text{Note that the top classification criterion depends only on the classified variable itself. All other criteria depend on the existence, the name and the content of variables out of B as well.}\]
3.2. compare a dataset with itself but change almost all variable names

In this step, we compare the data set once again with itself. But we have renamed all variables (except id). As can be seen from table 2, variation helps to find renamed variables with identical content in a reasonable way.

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>compared to</td>
<td>compared to</td>
</tr>
<tr>
<td></td>
<td>release 3</td>
<td>release 3</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>649</td>
<td>649</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn1</td>
<td>1038</td>
<td>1038</td>
</tr>
<tr>
<td>vbn1i</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn1n</td>
<td>1556</td>
<td>1556</td>
</tr>
<tr>
<td>vbn1ni</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sum</td>
<td>3244</td>
<td>3244</td>
</tr>
</tbody>
</table>

Table 2: Classification of the example file anchor2_DD.dta, release 3.0 in relation to a modified copy of itself (renamed variables) and vice versa.

3.3. compare a data set with an almost empty data set

If we compare the data set anchor2_dd.dta with an empty one (contains just the ID) than we get the following table. It checks only the variation within the variables. For further checks comparison data are required.

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3^{ren}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>compared to</td>
<td>compared to</td>
</tr>
<tr>
<td></td>
<td>release 3^{ren}</td>
<td>release 3</td>
</tr>
<tr>
<td>x</td>
<td>649</td>
<td>649</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>1038</td>
<td>1038</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn1i</td>
<td>1556</td>
<td>1556</td>
</tr>
<tr>
<td>vbn1n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn1ni</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sum</td>
<td>3244</td>
<td>3244</td>
</tr>
</tbody>
</table>

Table 2: Classification of the example file anchor2_DD.dta, release 3.0 in relation to itself and vice versa.
Table 3: Classification of the example file anchor2_DD.dta, release 3.0 in relation to an all but empty data set and vice versa.

### 3.4. add one or two test variables to the almost empty data set

To get more experiences, it is recommended to extend this almost empty data set stepwise with further variables. As a thought experiment we present here first steps only.

- [add one constant variable] One possibility is that nothing happens. Depending on name and content of the test variable jumps one variable from class x to either n or ni. Alternatively could one variable jump from v to vn.

- [add one variable with variation] Depending on name and content are different changes possible.
  - [arbitrary name] Depending on the existence of a bivariate mapping one or more variables could jump from v to vb, vbi, vb1 or vb1i.
  - [same name as a constant variable] One further variable jumps from x to n.
  - [same name as a variable with variation] One further variable jumps from v to vn, vbn, vbni, vb1n or vb1ni.

### 4. Results and Summary

The appendices of this report contain the documented code. For each of the nine data sets the code follows the same logic.

1. Analyze all variables without variation
2. Analyze all variables with variation, but without a bijective mapping
3. Analyze all variables with variation and with a bijective mapping
4. Provide a comparison summary

Here in this section we display and discuss as an example the summary output that is listed in table 4. It has been computed for the data set anchor2_DD.dta. You find all further details for this data set in the appendix B on page 18.

Table 4 contains both, the classification of the release 3.0 data set in respect to release 3.1 and, vice versa, the classification of the release 3.1 data set with respect to release 3.0.
In the following part of this section we discuss the classification of the file \texttt{anchor2_DD.dta}, release 3.0 in relation to release 3.1 (left column). Most variables are unchanged (ni, vbni, vb1ni). 591 variables (ni) do not have variation and are identical. Further 954 + 1502 variables (vbni, vb1ni) are unchanged as well, but have variation.\textsuperscript{3}

The use of these unchanged variables does not affect research results if one replace the release 3.0 dataset with the release 3.1 data set. It furthermore shows that no constant variables (x) of release 3.0 have been dropped. But 2 variables (v) with variation have been dropped. Furthermore, 58 variables (n) of release 3.0 have gained variation (in release 3.1) or have changed to another constant value (in release 3.1). That means, these variables need further inspection. Further 60 variables (vn) have been changed. They have the same name, but there exist no bijective mapping.\textsuperscript{4} These variables should be checked carefully, because their content has been changed substantially.

Variables classified as vb, vbi, vbn, vb1, vb1i or vb1n give indications for renamed or recoded variables.\textsuperscript{5} It is possible that these variables have only been recoded. However, these data need to be inspected still.

\begin{table}[h]
\centering
\begin{tabular}{llll}
\hline
\textbf{class} & \textbf{release 3} & \textbf{release 3.1} & \\
& \textbf{compared to} & \textbf{compared to} & \\
& \textbf{release 3.1} & \textbf{release 3} & \\
\hline
x & 0 & 211 & \\
n & 58 & 124 & \\
ni & 591 & 591 & \\
\hline
v & 2 & 2 & \\
vn & 60 & 15 & \\
vb & 12 & 0 & \\
vbi & 3 & 0 & \\
vbn & 19 & 38 & \\
vbni & 954 & 941 & \\
vb1 & 2 & 0 & \\
vb1i & 4 & 0 & \\
vb1n & 37 & 37 & \\
vb1ni & 1502 & 1496 & \\
\hline
\textbf{sum} & 3244 & 3455 & \\
\end{tabular}
\caption{Classification of the example file \texttt{anchor2_DD.dta}, release 3.0 in relation to release 3.1 and vice versa.}
\end{table}

\textsuperscript{3}The classification as vbni suggests that 954 variables may be redundant. Redundancy means that there is not only one identical variable in the release 3.1 data. There exists at least a second variable in release 3.1 data that has a bivariate mapping.\textsuperscript{4} The provided Levenshtein distance (Appendix) gives a rough idea whether only some cases have been changed or many.\textsuperscript{5} 12 variables (vb) have been potentially both renamed and recoded. 3 variables (vbi) have been possibly just renamed. 19 variables (vbn) have been possibly recoded. 2 variables (vb1) have been possibly renamed and recoded. 4 variables (vb1i) have been potentially renamed. 37 variables (vb1n) have been possible recoded.
Acknowledgements

The report was supported by the Max Planck Institute for Demographic Research, Rostock, Germany. The author would like to thank Michaela Kreyenfeld and Tom Hensel for the possibility to discuss different aspects of the topic and for valuable comments on the paper. I thank Andreas Müller for helping with the compareFinRaw.r code preparation.

Appendix

A. anchor1

Here we compare the data set anchor1_DD.dta from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\anchor1\Results\compareFinRaw.RData")
```

A.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 0
(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 26

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 0
(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
## [1] 0
```
Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R$", "", row.names(Rnv2))
Fnv2$nm <- gsub("F$", "", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)
(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 422
F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum(!(RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 1
RFnv2[!(RFnv2$Raw.no.var == RFnv2$Fin.no.var | is.na(RFnv2$Raw.no.var == RFnv2$Fin.no.var)), 1]
## [1] "flag26"
F.nB <- R.nB
(R.n <- R.nA + R.nB)
## [1] 1
(F.n <- F.nA + F.nB)
## [1] 1

# x
(xR <- RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)])

## character(0)

# nA (n = nA + nB)
yR <- RFnv2[is.na(RFnv2$Fin.no.var), 1]
yR[!(yR %in% xR)]

## character(0)

# x
(xF <- FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)])

## [1] "flag20"  "flag21"  "flag22"  "flag23"  "flag24"
## [6] "flag25"  "flag_ehc"  "flag_igb"  "k10sex_gen"  "k10doby_gen"
## [11] "k10dobm_gen"  "k10age"  "isced2"  "mschool"  "fschool"
## [16] "mvocat"  "fvocat"  "mcasmin"  "fcasmin"  "misced"
## [21] "fisced"  "myeduc"  "fyeduc"  "intcont"  "intsex"
## [26] "intage"
R <\nobreakdashes\> is.na(RFnv2$Raw.no.var), 1
yF[!(yF %in% xF)]

## character(0)

**Release 3.0**  For 0 variable without variation exists no variable with the same name in the comparison data set. 1 variables without variation has been changed. 422 variables without variation are identical in both data sets.

**Release 3.1**  For 26 variables without variation exist no variables with the same name in the comparison data set. 1 variables without variation has been changed. 422 variables without variation are identical in both data sets.

**A.2. variation, but no bijective mapping**

```
rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R$", "", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F$", "", as.character(rb$Fin.c.nm))
sum(!RL.bi)
## [1] 28
sum(!FL.bi)
## [1] 26
(R.v <\nobreakdashes\> sum(!(RawNames[!Rnv][!RL.bi] %in% FinNames)))
## [1] 2
(R.vn <\nobreakdashes\> sum(RawNames[!Rnv][!RL.bi][!(RawNames[!Rnv][!RL.bi] %in% FinNames)])
## [1] "sexratio" "popdens"
(R.v <\nobreakdashes\> sum(RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames])
## [1] 26
RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames]
## [1] "sd19k1y" "sd19k2y" "sd19k3y" "sdp10i1" "sdp10i14" "hc9hlp2" "sex_gen" "doby_gen" "doby_gen" "k1doby_gen"
## [11] "k2doby_gen" "k3doby_gen" "mage" "hhsizemrd" "mmd" "fmr" "hhcom" "pschool" "pisced"
## [21] "pcasmin" "pyeduc" "hhincgee" "pcasprim" "pcassec" "plfs"
(F.v <\nobreakdashes\> sum(!(FinNames[!Fnv][!FL.bi] %in% RawNames)))
```
Release 3.0  For 2 variables with variation but without any bijective mapping exist no variables with the same name in the release 3.1 data. 26 variables share the name with one of the variables in release 3.1 at least.

Release 3.1  For 1 variables with variation but without any bijective mapping exists no variables with the same name in the release 3.0 data. 25 variables share the name with one of the variables in release 3.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping. Those variables imply different information. DemoDiff data user should check whether they are using those variables for possible effects on their research. (The number of pairs may differ from 26 and 25).

```
results$same.name <- gsub("R$", "", as.character(results$Raw.c.nm)) == gsub("F$", "", as.character(results$Fin.c.nm))
sum(results$same.name & results$map.di>0)
```

```
# [1] 28
```

```
print(results[(results$same.name & results$map.di>0),c(2,3,5,6,7,8)],row.names=FALSE)
```

<table>
<thead>
<tr>
<th>Fin.c.nm</th>
<th>Fin.c.ls</th>
<th>Raw.c.nm</th>
<th>Raw.c.ls</th>
<th>map.di</th>
<th>ed.di.sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>sd4gF</td>
<td>3</td>
<td>sd4gR</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>sd14k1gF</td>
<td>5</td>
<td>sd14k1gR</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>sd19k1yF</td>
<td>31</td>
<td>sd19k1yR</td>
<td>31</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>sd19k2yF</td>
<td>27</td>
<td>sd19k2yR</td>
<td>27</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>sd19k3yF</td>
<td>20</td>
<td>sd19k3yR</td>
<td>20</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>sdp10i1F</td>
<td>4</td>
<td>sdp10i1R</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>sdp10i14F</td>
<td>4</td>
<td>sdp10i14R</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>hc9h1p2F</td>
<td>18</td>
<td>hc9h1p2R</td>
<td>18</td>
<td>8</td>
<td>25</td>
</tr>
<tr>
<td>sex_genF</td>
<td>2</td>
<td>sex_genR</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>doby_genF</td>
<td>6</td>
<td>doby_genR</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>dobm_genF</td>
<td>12</td>
<td>dobm_genR</td>
<td>12</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>
## \textit{k1doby\_genF} 31 \textit{k1doby\_genR} 31 14 7
## \textit{k2doby\_genF} 27 \textit{k2doby\_genR} 27 6 6
## \textit{k3doby\_genF} 20 \textit{k3doby\_genR} 20 2 1
## \textit{mageF} 43 \textit{mageR} 43 2 1
## \textit{homosexF} 3 \textit{homosexR} 3 2 1
## \textit{hhsizemrdF} 9 \textit{hhsizemrdR} 9 6 3
## \textit{mmrdF} 2 \textit{mmrdR} 2 2 17
## \textit{fmrdF} 2 \textit{fmrdR} 3 3 20
## \textit{hhcompF} 14 \textit{hhcompR} 15 9 38
## \textit{pschoolF} 11 \textit{pschoolR} 11 24 233
## \textit{piscedF} 11 \textit{piscedR} 11 8 14
## \textit{pcasminF} 12 \textit{pcasminR} 12 14 20
## \textit{pyeducF} 20 \textit{pyeducR} 21 51 243
## \textit{hhincgceeF} 330 \textit{hhincgceeR} 330 2 8
## \textit{pcasprimF} 22 \textit{pcasprimR} 22 2 2
## \textit{pcassecF} 17 \textit{pcassecR} 17 2 1
## \textit{plfsF} 13 \textit{plfsR} 13 2 2

### A.3. variation and bijective mapping

```r
# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R\$", ",", as.character(rb$Raw.c.nm)) == gsub("F\$", ",", as.character(rb$Fin.c.nm))

rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)
rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)

(R.vb <- length(unique(rb[, !rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"])))
## [1] 0

(R.vbi <- length(unique(rb[, !rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 0

(R.vbn <- length(unique(rb[, !rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"])))
## [1] 0

(R.vbni <- length(unique(rb[, !rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 224
```
(R.vb1 <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"))))
## [1] 0
R.vb1i <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
## [1] "sd4gR"  "sd14k1gR"
(R.vb1n <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
## [1] "flag7R"
(R.vb1ni <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"))))
## [1] 724
(F.vb <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$Fin.ident, "Fin.c.nm"))))
## [1] 0
(F.vbi <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$Fin.ident, "Fin.c.nm"))))
## [1] 0
(F.vbn <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$Fin.ident, "Fin.c.nm"))))
## [1] 0
(F.vbni <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$Fin.ident, "Fin.c.nm"))))
## [1] 223
(F.vb1 <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"])))
## [1] 0
F.vb1i <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))
as.character(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))
## [1] "sex_genF"  "doby_genF"  "dobm_genF"
(F.vb1n <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))
as.character(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))
## [1] "flag7F"
(F.vb1ni <- length(unique(rb[!rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"))))
## [1] 725
A.4. comparison summary for anchor1_DD.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>26</td>
</tr>
<tr>
<td>n</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ni</td>
<td>422</td>
<td>422</td>
</tr>
<tr>
<td>v</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>vn</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbni</td>
<td>224</td>
<td>223</td>
</tr>
<tr>
<td>vb1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1i</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>vb1n</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>vb1ni</td>
<td>724</td>
<td>725</td>
</tr>
<tr>
<td>sum</td>
<td>1402</td>
<td>1427</td>
</tr>
</tbody>
</table>

Most variables are unchanged (ni, vbni, vb1ni). A number (x, v) had been dropped or added. A small number of variables (vb1, vb1i, vb1n) had been (probably) renamed or recoded. A group of variables (vn) had been changed. This analysis provide no further hints for this group.

A.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```r
table(Raw$flag26R)

##
## -10
## 1489

table(Fin$flag26F)

##
## 0
## 1489
```

The variable 'flag26' has been recoded from '-10' to '0'.

```r
table(Raw$sd4gR)

##
## -3  1  2
## 342 635 512

table(Fin$sd4gF)

##
## -3  1  2
## 342 636 511
```

D2 <- cbind(Raw$sd4gR, Fin$sd4gF)
D2[D2[, 1] != D2[, 2], ]
Both variables share the same three levels. \(3 + 3 + 2 = 8\), \(8/2 = 4\) mappings are in use. The level ‘2’ has been recoded to ‘1’ in one case. Changing ‘2’ to ‘1’ for one individual takes one Levenshtein steps (see details in [TR-2012-003]). There is no bijective mapping available.

233 individuals have a new level for ‘pschool’. It takes 233 Levenshtein steps to get them equal. There is no bijective mapping available.
One individual has a different level. There is no bijective mapping.

```r
d5 <- cbind(Raw$pcasminR, Fin$pcasminF)
dim(d5[, 1] != d5[, 2], ])

## [1] 20 2
unique(d5[, 1] != d5[, 2], ])

## [1] 4 7 5 6 4 3 5 2 3 7

results[results$Fin.c.nm == "pcasminF" & results$Raw.c.nm == "pcasminR", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 930020 pcasminF 12 pcasminR 12 14 20

20 individuals have a different level. There is no bijective mapping.

D6 <- cbind(Raw$hhincgceeR, Fin$hhincgceeF)
D6[D6[, 1] != D6[, 2], ]

## [1] "1732.05078" "2121.32031"

results[results$Fin.c.nm == "hhincgceeF" & results$Raw.c.nm == "hhincgceeR", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 942760 hhincgceeF 330 hhincgceeR 330 2 8

One individual has a different level (Levenshtein distance 8). There is no bijective mapping.

```r
table(Raw$flag7R)

##
## 0 99
## 1485 4

table(Fin$flag7F)

##
## 0 1
## 1485 4
D7 <- cbind(Raw$flag7R, Fin$flag7F)
unique(D7[D7[, 1] != D7[, 2], ])

## [,1] [,2]
## [1,] "0" "0"
## [2,] "99" "1"

results[results$same.name & results$Fin.c.nm == "flag7F", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 809483 flag7F 2 flag7R 2 0 1493

Both variables share the same number of levels. It exists a bijective mapping. We have to recode 1485 times from ' 0' to '0' and four times from '99' to '1'. This takes $1485 + 8 = 1493$ Levenshtein steps to make the levels equal. Unfortunately, this version of the comparison script distinguishes between ' 0' and '0'.

results[results$map.di == 0 & results$Raw.c.nm == "sd4gR", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 822385 psex_genF 3 sd4gR 3 0 0

results[results$map.di == 0 & results$Fin.c.nm == "sd14k1gR", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 823609 k1sex_genF 5 sd14k1gR 5 0 0

results[results$map.di == 0 & results$Raw.c.nm == "sex_genF", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 821384 sex_genF 2 sexR 2 0 0

results[results$map.di == 0 & results$Fin.c.nm == "doby_genF", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 832155 doby_genF 6 dobyR 6 0 0

results[results$map.di == 0 & results$Fin.c.nm == "dobm_genF", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 833133 dobm_genF 12 dobmR 12 0 0

Check some identical variables with different name. For example 'sd4g' from release 3.0 is identical to 'psex_gen' from release 3.1. But there exists a difference for 'sd4g' in both releases (one changed case, see above).
B. anchor2

Here we compare the data set anchor2_dd.dta from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\anchor2\Results\compareFinRaw.RData")
```

B.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 0
(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 211

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 0
(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
## [1] 66

Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv$nm <- gsub("R$", "", row.names(Rnv2))
Fnv$nm <- gsub("F$", "", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)

(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 591

F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum(!(RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 58
```
RFnv2(!RFnv2$Raw.no.var == RFnv2$Fin.no.var | is.na(RFnv2$Raw.no.var == RFnv2$Fin.no.var)), 1]

## [1] "afnat1" "cid1" "cid2" "cid3" "cid4"
## [6] "cid6" "cid7" "cid8" "cid9"
## [11] "ehc10k10h1" "ehc10k10h2" "ehc11k10m" "ehc11k10y"
## [16] "ehc12k10" "ehc5p2" "ehc5p3" "ehc7k10g" "ehc7k10m"
## [21] "ehc8k10d" "ehc8k10m" "ehc8k10y" "fid" "igr73i11"
## [26] "igr73i10" "igr73i11" "igr73i12" "igr73i12o" "igr73i12o"
## [31] "igr73i13" "igr73i14" "igr73i15" "igr73i16" "igr73i17"
## [36] "igr73i18" "igr73i19" "igr74" "igr74o" "igr77i11"
## [41] "igr77i10" "igr77i11" "igr77i12" "igr77i12o" "igr77i12o"
## [46] "igr77i14" "igr77i15" "igr77i16" "igr77i17" "igr77i18"
## [51] "igr77i19" "igr78" "igr78o" "k10dobm_gen" "k10doby_gen"
## [56] "mid" "sfid" "smid"

F.nB <- R.nB
(R.n <- R.nA + R.nB)

## [1] 58

(F.n <- F.nA + F.nB)

## [1] 124

# x
(xR <- RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)])

## character(0)

# nA (n = nA + nB)
yR <- RFnv2[is.na(RFnv2$Fin.no.var), 1]
yR[(yR %in% xR)]

## character(0)

# x
(xF <- FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)])

## [1] "cid10" "d22" "d23" "d37" "d38"
## [6] "d52" "d53" "d530" "d351" "d67"
## [11] "d68" "d82" "d83" "d97" "d98"
## [16] "d112" "d113" "d127" "d128" "d335"
## [21] "d336" "d181" "d182" "d183" "d184"
## [26] "d185" "d186" "d187" "d188" "d201"
## [31] "d202" "d203" "d204" "d205" "d206"
## [36] "d207" "d208" "d523" "d524" "d525"
## [41] "d526" "d527" "d528" "d529" "d530"
## [46] "d543" "d544" "d545" "d546" "d547"
## [51] "d548" "d549" "d550" "d402" "d403"
## [56] "d404" "d405" "d406" "d411" "d412"
## [61] "d420" "d421" "d422" "d423" "d424"
MPIDR | DemoDiff | Formal Data Comparison

## [66] "d425" "d426" "d427" "d428" "d429"
## [71] "d430" "ehc13k6" "ehc28p4" "ehc29p4" "ehc27p5i1"
## [76] "ehc27p5i2" "ehc27p5i2o" "ehc28p5" "ehc29p5" "ehc28p4m1"
## [81] "ehc28p4m2" "ehc28p4m3" "ehc28p4m4" "ehc28p4m5" "ehc28p4m6"
## [86] "ehc28p4m7" "ehc28p4m8" "ehc28p4m9" "ehc28p4m10" "ehc28p4m11"
## [91] "ehc28p4m12" "ehc28p5m1" "ehc28p5m2" "ehc28p5m3" "ehc28p5m4"
## [96] "ehc28p5m5" "ehc28p5m6" "ehc28p5m7" "ehc28p5m8" "ehc28p5m9"
## [101] "ehc28p5m10" "ehc28p5m11" "ehc28p5m12" "ehc28p5m13" "ehc28p5m14"
## [106] "ehc28p5m15" "ehc28p5m16" "ehc28p5m17" "ehc28p5m17" "ehc22p10n"
## [111] "ehc22p11n" "ehc22p12n" "ehc22p13n" "ehc23p9" "ehc23p10"
## [116] "ehc23p11" "ehc23p12" "ehc23p13" "ehc24p9m" "ehc24p10m"
## [121] "ehc24p11m" "ehc24p12m" "ehc24p13m" "ehc24p9y" "ehc24p10y"
## [126] "ehc24p11y" "ehc24p12y" "ehc24p13y" "ehc25p9h1" "ehc25p10h1"
## [131] "ehc25p11h1" "ehc25p12h1" "ehc25p13h1" "ehc25p9h2" "ehc25p10h2"
## [136] "ehc25p11h2" "ehc25p12h2" "ehc25p13h2" "ehc25p9h3" "ehc25p10h3"
## [141] "ehc25p11h3" "ehc25p12h3" "ehc25p13h3" "sep4k5" "sep7k5"
## [146] "sep8k5" "sep10k5" "sep10k5" "crn23k5" "crn23k7"
## [151] "crn13k10i1" "crn13k10i1" "crn13k10i3" "crn13k10i4" "crn13k10i5"
## [156] "crn13k10i6" "crn13k10i8" "crn13k10i9" "crn13k10i10" "crn13k10i11"
## [161] "crn13k10i12" "crn13k10i13" "crn14k10i1" "crn14k10i1" "crn14k10i12"
## [166] "crn14k10i13" "crn14k10i14" "crn14k10i14" "crn14k10i15" "crn14k10i16"
## [171] "crn14k10i17" "crn14k10i18" "crn14k10i19" "crn14k10i10" "crn14k10i11"
## [176] "crn14k10i12" "crn14k10i14" "crn14k10i14" "crn15k10" "rtr23h12"
## [181] "rtr24h12" "exp_di" "tag_identp" "flag_enc" "flag_frt6"
## [186] "mschool" "fschool" "mvocat" "fvocat" "mcasmin"
## [191] "tag_dob" "tag_dobk1" "tag_dobk2" "tag_dobk3" "tag_dobk4"
## [196] "tag_dobk5" "tag_dobk6" "tag_dobk7" "tag_dobk8" "tag_dobk9"
## [206] "tag_sexk1" "tag_sexk2" "tag_sexk3" "tag_sexk4" "tag_sexk5"
## [211] "tag_sexk6" "tag_sexk7" "tag_sexk8" "tag_sexk9"
## [216] "tag_gender"
Release 3.1  For 211 variables without variation exist no variables with the same name in the comparison data set. 124 variables without variation have been changed. 591 variables without variation are identical in both data sets.

B.2. variation, but no bijective mapping

```r
rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R$", "", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F$", "", as.character(rb$Fin.c.nm))
sum(!RL.bi)
## [1] 62
sum(!FL.bi)
## [1] 17
(R.v <- sum(!({RawNames[!Rnv][!RL.bi] %in% FinNames})))
## [1] 2
RawNames[!Rnv][!RL.bi][!(RawNames[!Rnv][!RL.bi] %in% FinNames)]
## [1] "sexratio" "popdens"
(R.vn <- sum(RawNames[!Rnv][!RL.bi] %in% FinNames))
## [1] 60
RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames]
## [1] "d9" "d9" "d90" "d91" "ehc1p1g"
## [6] "ehc8k1y" "crn1k2" "crn1k3" "crn1k4" "crn1k2o"
## [11] "crn2k2i1" "crn2k2i2" "crn2k3i1" "crn2k3i2" "crn2k4i1"
## [16] "crn2k4i2" "crn3k2" "crn3k3" "crn3k4" "crn4k2"
## [21] "crn4k3" "crn4k4" "crn5k2" "crn5k3" "crn5k4"
## [26] "crn6k2" "crn6k3" "crn6k4" "crn7k2" "crn7k3"
## [31] "crn7k4" "crn10k2i1" "crn10k2i2" "crn10k2i3" "crn10k2i4"
## [36] "crn10k3i1" "crn10k3i2" "crn10k3i3" "crn10k3i4" "crn10k4i1"
## [41] "crn10k4i2" "crn10k4i3" "crn10k4i4" "igr82i12o" "igr85i12o"
## [46] "rtr23h3" "hv1" "tag_sex" "tag_dob" "tag_sek2"
## [51] "tag_dook1" "tag_dook2" "tag_dook3" "tag_sexp" "kldoby_gen"
## [56] "pschool" "pisced" "pisced2" "pcasmin" "pyeduc"
(F.v <- sum(!({FinNames[!Fnv][!FL.bi] %in% RawNames})))
## [1] 2
FinNames[!Fnv][!FL.bi][!(FinNames[!Fnv][!FL.bi] %in% RawNames)]
## [1] "ykagecapi" "ykidcapi"
```
Release 3.0  For 2 variables with variation but without any bijective mapping exist no variables with the same name in the release 3.1 data. 60 variables share the name with one of the variables in release 3.1 at least.

Release 3.1  For 2 variables with variation but without any bijective mapping exist no variables with the same name in the release 3.0 data. 15 variables share the name with one of the variables in release 3.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping. Those variables imply different information. DemoDiff data user should check whether they are using those variables for possible effects on their research. (The number of pairs may differ from 60 and 15).

```r
(results$same.name <-
  gsub("R$","",as.character(results$Raw.c.nm)) == gsub("F$","",as.character(results$Fin.c.nm))
sum(results$same.name & results$map.di>0))
```

## [1] 15

### Print results

```r
print(results[(results$same.name & results$map.di>0),c(2,3,5,6,7,8)],row.names=FALSE)
```

### B.3. variation and bijective mapping

```r
(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))
```

## [1] 15

FinNames[!Fnv][!FL.bi][FinNames[!Fnv][!FL.bi] %in% RawNames]

## [1] "d9" "d89" "d90" "d91" "ehc1p1g"
## [6] "ehc8k1y" "rtr23h3" "hv1" "tag_sexx2" "kidoby_gen"
## [11] "pschool" "pisced" "pisced2" "pcasmin" "pyeduc"

```r
(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))
```

## [1] 15

FinNames[!Fnv][!FL.bi][FinNames[!Fnv][!FL.bi] %in% RawNames]

## [1] "d9" "d89" "d90" "d91" "ehc1p1g"
## [6] "ehc8k1y" "rtr23h3" "hv1" "tag_sexx2" "kidoby_gen"
## [11] "pschool" "pisced" "pisced2" "pcasmin" "pyeduc"

```r
(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))
```
```r
# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R\$", "", as.character(rb$Raw.c.nm)) == gsub("F\$", "", as.character(rb$Fin.c.nm))

rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)

R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))

## [1] "crn1k5R" "crn2k5i1R" "crn2k5i2R" "crn3k5R" "crn4k5R"
## [6] "crn5k5R" "crn6k5R" "crn7k5R" "crn10k5i1R" "crn10k5i2R"
## [11] "crn10k5i3R" "crn10k5i4R"

R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"]))

## [1] "tag_sexk9R" "tag_sexk8R" "tag_sexk6R"

R.vbn <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))

## [1] "d35R" "d348R" "d109R" "d124R" "d95R" "d96R" "d108R" "d123R"
## [9] "d158R" "d518R" "d538R" "d519R" "d520R" "d540R" "d521R"
## [17] "d541R" "d522R" "d542R"

(R.vbni <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))

## [1] 954

R.vb1 <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))

## [1] "tag_sexk1R" "tag_dobk5R"

R.vbi1 <- length(unique(rb[rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"]))

## [1] "tag_sexk3R" "tag_sexk4R" "tag_sexk5R" "tag_sexk7R"

R.vb1n <- length(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"]))
```
# [1] "d30R"  "d31R"  "d32R"  "d34R"  "d45R"  "d46R"  "d47R"  "d342R"
# [9]  "d343R"  "d344R"  "d345R"  "d346R"  "d347R"  "d92R"  "d93R"  "d94R"
# [17]  "d105R"  "d106R"  "d107R"  "d120R"  "d121R"  "d122R"  "d155R"  "d156R"
# [25]  "d157R"  "d196R"  "d197R"  "d198R"  "d199R"  "d200R"  "d217R"  "d283R"
# [33]  "d317R"  "d397R"  "hm1R"  "hsv1R"  "hsm1R"
(R.vb1ni <- length(unique(rb[rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 1502

(F.vb <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))
## [1] 0
(F.vb1i <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"])))
## [1] 0
(F.vb1n <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))
as.character(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"]))
## [1] "d30F"  "d31F"  "d32F"  "d34F"  "d35F"  "d36F"  "d48F"  "d49F"  "d50F"  "d51F"  "d348F"
# [9]  "d349F"  "d96F"  "d108F"  "d109F"  "d110F"  "d111F"  "d123F"  "d124F"  "d125F"  "d126F"  "d328F"  "d330F"  "d331F"  "d332F"  "d333F"  "d334F"  "d335F"  "d336F"  "d337F"  "d338F"  "d339F"  "d340F"  "d341F"  "d342F"
# [17]  "d343F"  "d344F"  "d345F"  "d346F"  "d347F"  "d92F"  "d93F"  "d94F"  "d105F"  "d106F"  "d107F"  "d120F"  "d121F"  "d122F"  "d155F"  "d156F"  "d157F"  "d196F"  "d197F"  "d198F"  "d199F"  "d200F"  "d217F"  "d283F"  "d317F"  "d397F"  "hm1F"  "hsv1F"  "hsm1F"
(F.vb1ni <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"])))
## [1] 941
(F.vb1 <- length(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"])))
## [1] 0
(F.vb1i <- length(unique(rb[rb$dup.Fin & !rb$same.name & rb$ident, "Fin.c.nm"])))
## [1] 0
(F.vb1n <- length(unique(rb[rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"])))
as.character(unique(rb[rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"]))
## [1] "d30F"  "d31F"  "d32F"  "d34F"  "d35F"  "d36F"  "d48F"  "d49F"  "d50F"  "d51F"  "d348F"
# [9]  "d349F"  "d96F"  "d108F"  "d109F"  "d110F"  "d111F"  "d123F"  "d124F"  "d125F"  "d126F"  "d328F"  "d330F"  "d331F"  "d332F"  "d333F"  "d334F"  "d335F"  "d336F"  "d337F"  "d338F"  "d339F"  "d340F"  "d341F"  "d342F"  "d343F"  "d344F"  "d345F"  "d346F"  "d347F"  "d92F"  "d93F"  "d94F"  "d105F"  "d106F"  "d107F"  "d120F"  "d121F"  "d122F"  "d155F"  "d156F"  "d157F"  "d196F"  "d197F"  "d198F"  "d199F"  "d200F"  "d217F"  "d283F"  "d317F"  "d397F"  "hm1F"  "hsv1F"  "hsm1F"
B.4. comparison summary for anchor2_DD.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>211</td>
</tr>
<tr>
<td>n</td>
<td>58</td>
<td>124</td>
</tr>
<tr>
<td>ni</td>
<td>591</td>
<td>591</td>
</tr>
<tr>
<td>v</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>vn</td>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>vb</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>19</td>
<td>38</td>
</tr>
<tr>
<td>vbi</td>
<td>954</td>
<td>941</td>
</tr>
<tr>
<td>vb1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>vb1i</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>vb1n</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>vb1ni</td>
<td>1502</td>
<td>1496</td>
</tr>
<tr>
<td>sum</td>
<td>3244</td>
<td>3455</td>
</tr>
</tbody>
</table>

Most variables are unchanged (ni, vbni, vb1ni). A number (x, v) had been dropped or added. A small number of variables (vb1, vb1i, vb1n, vbi, vbn) had been (probably) renamed or recoded. A group of variables (vn) had been changed. This analysis provide no further hints for this group.

B.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```r
table(Raw$afnat1R)
##
## NA
## 1173

table(Fin$afnat1F)
##
## -3
## 1173

table(Raw$midR)
##
## -10
## 1173

table(Fin$midF)
##
## NA
## 1173

table(Raw$tag_dobk1R)
##
## 25
```
The variables 'afnat1' and 'mid' have been recoded. The third variable 'tag_dobk1' lost all variance.

```r
d2 <- cbind(Raw$pschoolR, Fin$pschoolF)
results[results$Fin.c.nm == "pschoolF" & results$Raw.c.nm == "pschoolR", c(2, 3, 5, 6, 7, 8)]
```

Both variables share the same number of levels. It does not exists a bijective mapping. We have to recode 55 individual cases. This takes 55 Levenshtein steps to make the levels equal.
D3 <- cbind(Raw$pyeducR, Fin$pyeducF)
dim(D3[D3[, 1] != D3[, 2], ])

## [1] 48  2

unique(D3[D3[, 1] != D3[, 2], ])

## [,1] [,2]
## [1, ] "14.0" "16.0"
## [2, ] "16.0" "17.0"
## [3, ] "13.5" "14.5"
## [4, ] "14.0" "15.0"
## [5, ] "17.0" "16.0"
## [6, ] "16.0" "15.0"
## [7, ] "10.5" "11.5"
## [8, ] "15.0" "16.0"
## [9, ] "10.0" "13.0"
## [10, ] "17.0" "18.0"
## [11, ] "13.0" "10.0"
## [12, ] " 9.0" "10.0"
## [13, ] "10.5" "13.5"
## [14, ] "13.0" "15.0"
## [15, ] "11.5" "13.5"
## [16, ] "19.0" "20.0"
## [17, ] "16.0" "13.0"
## [18, ] "13.0" "16.0"
## [19, ] "11.0" "12.0"
## [20, ] "12.0" "13.0"

results[results$Fin.c.nm == "pyeducF" & results$Raw.c.nm == "pyeducR", c(2, 3, 5, 6, 7, 8)]

##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 6503045 pyeducF      19 pyeducR     20    39     50

48 individuals have a new level for 'pyeduc'. It takes 50 Levenshtein steps to get them equal. There is no bijective mapping available.

D4 <- cbind(Raw$hv1R, Fin$hv1F)
dim(D4[D4[, 1] != D4[, 2], ])

## [1] 46  2

unique(D4[D4[, 1] != D4[, 2], ])

## [,1] [,2]
## [1, ] "-3" "NA"

results[results$Fin.c.nm == "hv1F" & results$Raw.c.nm == "hv1R", c(2, 3, 5, 6, 7, 8)]

##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 5989024  hv1F      3  hv1R      4    1     92
46 individuals have a different level. They were recoded from '-3' to 'NA'. There is no bijective mapping.

```r
D5 <- cbind(Raw$piscedR, Fin$piscedF)
D5[D5[, 1] != D5[, 2], ]
## [,1] [,2]
## [1,]  "3"  "5"
## [2,]  "5"  "1"
## [3,]  "3"  "1"
## [4,]  "4"  "6"
## [5,]  "4"  "6"
## [6,]  "4"  "6"
## [7,]  "4"  "6"
results[results$Fin.c.nm == "piscedF" & results$Raw.c.nm == "piscedR", c(2, 3, 5, 6, 7, 8)]
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 6487469 piscedF 11 piscedR 11 8 7
```

There is no bijective mapping. 7 individuals have been recoded.

```r
table(Raw$tag_sexk1R)
##
## 0 1
## 1172 1
table(Fin$tag_sexk1F)
##
## 0
## 1173
results[results$map.di == 0 & results$Raw.c.nm == "tag_sexk1R", c(2, 3, 5, 6, 7, 8)]
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 2402778 ehc19i17m1F 2 tag_sexk1R 2 0 2346
```

There is a (stochastic) mapping between to strange variables. But a variable with the same name exists as well.

```r
table(Raw$tag_sexk3R)
##
## 0 1
## 1161 12
table(Fin$tag_sexk3F)
```

28
There exists an identical variable with a different name (by chance). A variable with the same name exists, but has a slightly modified content.

```
D8 <- cbind(Raw$hm1R, Fin$hm1F)
dim(D8[D8[, 1] != D8[, 2], ])
## [1] 6 2

unique(D8)
## [,1] [,2]
## [1,] "1" "1"
## [2,] "2" "2"
## [3,] "-3" "NA"

results[results$Fin.c.nm == "hm1F" & results$Raw.c.nm == "hm1R", c(2, 3, 5, 6, 7, 8)]
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 5986428 hm1F 3 hm1R 3 0 12

D9 <- cbind(Raw$hsv1R, Fin$hsv1F)
dim(D9[D9[, 1] != D9[, 2], ])
## [1] 750 2

unique(D9)
## [,1] [,2]
## [1,] "-3" "NA"
## [2,] "2" "2"
## [3,] "1" "1"

results[results$Fin.c.nm == "hsv1F" & results$Raw.c.nm == "hsv1R", c(2, 3, 5, 6, 7, 8)]
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 5996812 hsv1F 3 hsv1R 3 0 1500
```

D10 <- cbind(Raw$hsm1R, Fin$hsm1F)
dim(D10[D10[, 1] != D10[, 2], ])
```
These variables are examples for simply recoded variables.

### C. anchor4

Here we compare the data set anchor4_dd.dta from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\anchor4\Results\compareFinRaw.RData")
```

#### C.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 16
(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 342

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 4
(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
```
```r
## [1] 3
Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R\$", ",", row.names(Rnv2))
Fnv2$nm <- gsub("F\$", ",", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)
(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 886
F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum(! (RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 6
RFnv2[! (RFnv2$Raw.no.var == RFnv2$Fin.no.var | is.na(RFnv2$Raw.no.var == RFnv2$Fin.no.var)), 1]
## [1] "cid1" "cid10" "cid6" "cid7" "cid8" "cid9"
F.nB <- R.nB
(R.n <- R.nA + R.nB)
## [1] 10
(F.n <- F.nA + F.nB)
## [1] 9

# x
(xR <- RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)])
## [1] "d24" "d25" "d26" "d27" "d28"
## [6] "ehc12k9o" "ehc12k10o" "netp31n" "netp32n" "netp33n"
## [11] "netp34n" "netp35n" "netp36n" "netp37n" "netp38n"
## [16] "netp39n"

# nA (n = nA + nB)
yR <- RFnv2[is.na(RFnv2$Fin.no.var), 1]
yR[!(yR %in% xR)]
## [1] "cid2" "cid3" "cid4" "cid5"

# x
(xF <- FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)]
```

31
## [1] "f_cid" "d37" "d38" "d52" "d53"
## [6] "d350" "d351" "d67" "d68" "d82"
## [11] "d83" "d97" "d98" "d112" "d113"
## [16] "d127" "d128" "d335" "d336" "d138"
## [21] "d139" "d140" "d141" "d142" "d143"
## [26] "d157" "d158" "d159" "d160" "d161"
## [31] "d162" "d163" "d206" "d207" "d208"
## [36] "d209" "d210" "d211" "d212" "d213"
## [41] "d214" "d215" "d528" "d529" "d530"
## [46] "d531" "d532" "d533" "d534" "d535"
## [51] "d536" "d537" "d548" "d549" "d550"
## [56] "d551" "d552" "d553" "d554" "d555"
## [61] "d556" "d557" "d281" "d420" "d421"
## [66] "d422" "d423" "d424" "d425" "d426"
## [71] "d427" "d428" "d429" "d430" "d505"
## [76] "d506" "ehc8k9d" "ehc8k10d" "ehc8k9m" "ehc8k10m"
## [81] "ehc13k6" "ehc27p5i12" "ehc28p5" "ehc23p11" "ehc23p12"
## [86] "ehc24p12y" "ehc24p13y" "ehc25p11" "ehc25p12" "ehc25p13"
## [91] "ehc10k9h1" "ehc10k10h1" "sd3514" "sd3515" "sep4k3"
## [96] "ehc13k9i1" "crn2k5i1" "crn2k5i2" "crn2k5i3" "crn2k6i1"
## [101] "ehc13k9i2" "crn3k9i4" "crn3k9i5" "crn3k9i6" "crn3k9i7"
## [106] "ehc13k9i8" "crn3k9i9" "crn3k9i10" "crn3k9i11" "crn3k9i12"
## [111] "ehc13k9i13" "crn4k9i5" "crn4k9i6" "crn4k9i7" "crn4k9i8"
## [116] "ehc13k9i14" "crn4k9i10" "crn4k9i11" "crn4k9i12" "crn4k9i13"
## [121] "ehc13k9i15" "crn4k9i16" "crn4k9i17" "crn4k9i18" "crn4k9i19"
## [126] "ehc13k9i16" "crn5k9" "crn5k10" "crn5k11" "crn5k12"
## [131] "ehc13k9i17" "crn5k13" "crn5k14" "crn5k15" "crn5k16"
## [136] "ehc13k9i18" "crn5k17" "crn5k18" "crn5k19" "crn5k20"
## [141] "ehc13k9i19" "crn5k21" "crn5k22" "crn5k23" "crn5k24"
## [146] "ehc13k9i20" "crn5k25" "crn5k26" "crn5k27" "crn5k28"
## [151] "ehc13k9i21" "crn5k29" "crn5k30" "crn5k31" "crn5k32"
## [156] "ehc13k9i22" "crn5k33" "crn5k34" "crn5k35" "crn5k36"
## [161] "ehc13k9i23" "crn5k37" "crn5k38" "crn5k39" "crn5k40"
## [166] "ehc13k9i24" "crn5k41" "crn5k42" "crn5k43" "crn5k44"
## [171] "ehc13k9i25" "crn5k45" "crn5k46" "crn5k47" "crn5k48"
## [176] "ehc13k9i26" "crn5k49" "crn5k50" "crn5k51" "crn5k52"
## [181] "ehc13k9i27" "crn5k53" "crn5k54" "crn5k55" "crn5k56"
## [186] "ehc13k9i28" "crn5k57" "crn5k58" "crn5k59" "crn5k60"
## [191] "ehc13k9i29" "crn5k61" "crn5k62" "crn5k63" "crn5k64"
## [196] "ehc13k9i30" "crn5k65" "crn5k66" "crn5k67" "crn5k68"
## [201] "ehc13k9i31" "crn5k69" "crn5k70" "crn5k71" "crn5k72"
## [206] "ehc13k9i32" "crn5k73" "crn5k74" "crn5k75" "crn5k76"
## [211] "ehc13k9i33" "crn5k77" "crn5k78" "crn5k79" "crn5k80"
## [216] "ehc13k9i34" "crn5k81" "crn5k82" "crn5k83" "crn5k84"
## [221] "ehc13k9i35" "crn5k85" "crn5k86" "crn5k87" "crn5k88"
## [226] "ehc13k9i36" "crn5k89" "crn5k90" "crn5k91" "crn5k92"
## [231] "ehc13k9i37" "crn5k93" "crn5k94" "crn5k95" "crn5k96"
## [236] "ehc13k9i38" "crn5k97" "crn5k98" "crn5k99" "crn6k00"
## [241] "ehc13k9i39" "crn5k101" "crn5k102" "crn5k103" "crn5k104"
## [246] "ehc13k9i40" "crn5k105" "crn5k106" "crn5k107" "crn5k108"
## [251] "ehc13k9i41" "crn5k109" "crn5k110" "crn5k111" "crn5k112"
## [256] "ehc13k9i42" "crn5k113" "crn5k114" "crn5k115" "crn5k116"
## [261] "ehc13k9i43" "crn5k117" "crn5k118" "crn5k119" "crn5k120"
## [266] "ehc13k9i44" "crn5k121" "crn5k122" "crn5k123" "crn5k124"
## [271] "ehc13k9i45" "crn5k125" "crn5k126" "crn5k127" "crn5k128"
## [276] "ehc13k9i46" "crn5k129" "crn5k130" "crn5k131" "crn5k132"
For 16 variables without variation exist no variables with the same name in the comparison data set. 10 variables without variation have been changed. 886 variables without variation are identical in both data sets.

For 342 variables without variation exist no variables with the same name in the comparison data set. 9 variables without variation have been changed. 886 variables without variation are identical in both data sets.

C.2. variation, but no bijective mapping

```r
# nA (n = nA + nB)
yF <- RFnv2[is.na(RFnv2$Raw.no.var), 1]
yF[!(yF %in% xF)]
```

```
# [1] "tag_dobk1" "tag_dobk2" "tag_sexp"
```
Release 3.0  For 13 variables with variation but without any bijective mapping exist no variables with the same name in the release 3.1 data. 14 variables share the name with one of the variables in release 3.1 at least.

Release 3.1  For 3 variables with variation but without any bijective mapping exist no variables with the same name in the release 3.0 data. 14 variables share the name with one of the variables in release 3.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping. Those variables imply different information. DemoDiff data user should check whether they are using those variables for possible effects on their research. (The number of pairs may differ from 14 and 14).

```r
(results$same.name <- gsub("R\$","",as.character(results$Raw.c.nm)) == gsub("F\$","",as.character(results$Fin.c.nm)))
sum(results$same.name & results$map.di>0)

## [1] 11
```

```r
print(results[(results$same.name & results$map.di>0),c(2,3,5,6,7,8)],row.names=FALSE)
```
## Fin.c.nm  Fin.c.ls  Raw.c.nm  Raw.c.ls  map.di  ed.di.sum
---
d9F     3     d9R     3     4     4
## d89F     28     d89R     28     4     2
## hpgF     3     hpgR     3     4     2
## ykageF     200     ykageR     201    101   884
## ykidF     9     ykidR     9    20   849
## pschoolF     10     pschoolR     10    18    52
## piscedF     11     piscedR     11    6    6
## pisced2F     10     pisced2R     10    6    6
## pcasminF     12     pcasminR     12    6   10
## pyeducF     19     pyeducR     20    37    51

### C.3. variation and bijective mapping

```r
# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R$", ",", as.character(rb$Raw.c.nm)) == gsub("F$", ",", as.character(rb$Fin.c.nm))

rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)

rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)

R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))

## [1] "crn16k7R"  "crn16k8R"  "crn17k8R"

R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))

## [1] "crn17k7R"

(R.vbn <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))

## [1] 0

(R.vbni <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$ident, "Raw.c.nm"])))

## [1] 1047

(R.vb1 <- length(unique(rb[!rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"])))
```

35
```r
## [1] 0
R.vb1i <- length(unique(rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"))
as.character(unique(rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"))

## [1] "intnumR"
R.vb1n <- length(unique(rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"))
as.character(unique(rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"))

## [1] "he2R"
(R.vb1ni <- length(unique(rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm")))

## [1] 1793
(F.vb <- length(unique(!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))

## [1] 0
(F.vbi <- length(unique(!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm")))

## [1] 0
(F.vbn <- length(unique(!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))

## [1] 0
(F.vbni <- length(unique(!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm")))

## [1] 1047
F.vb1 <- length(unique(rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"))
as.character(unique(rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"))

## [1] "cid5F"
(F.vb1i <- length(unique(rb$dup.Fin & !rb$same.name & rb$ident, "Fin.c.nm")))

## [1] 0
F.vb1n <- length(unique(rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"))
as.character(unique(rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"))

## [1] "he2F"
(F.vb1ni <- length(unique(rb$dup.Fin & rb$same.name & rb$ident, "Fin.c.nm")))

## [1] 1793
```
C.4. comparison summary for anchor4_DD.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>16</td>
<td>342</td>
</tr>
<tr>
<td>n</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>ni</td>
<td>886</td>
<td>886</td>
</tr>
<tr>
<td>v</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>vn</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>vb</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbni</td>
<td>1047</td>
<td>1047</td>
</tr>
<tr>
<td>vb1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>vb1i</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>vb1n</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>vb1ni</td>
<td>1793</td>
<td>1793</td>
</tr>
<tr>
<td>sum</td>
<td>3785</td>
<td>4096</td>
</tr>
</tbody>
</table>

Most variables are unchanged (ni, vbni, vb1ni). A number (x, v) had been dropped or added. A small number of variables (vb1, vb1i, vb1n, vbi) had been (probably) renamed or recoded. A group of variables (vn) had been changed. This analysis provide no further hints for this group.

C.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```R
table(Raw$cid1R)
```

```
##
##  -3
##  1074
```

```R
table(Fin$cid1F)
```

```
##
##  NA
##  1074
```

```R
table(Raw$cid2R)
```

```
##
##  -3
##  1074
```

```R
table(Fin$cid2F)
```

```
##
##  -3  NA
##   4 1070
```

The variable 'cid1' has been recoded. The variable 'cid2' has gained some variation.
Both variables share the same number of levels. It exists no bijective mapping. One individual has been recoded from '65' to '66'.

2 individuals have a new level for 'hpg'. It takes 2 Levenshtein steps to get them equal. There is no bijective mapping available.
D4 <- cbind(Raw$ykageR, Fin$ykageF)

dim(D4[D4[, 1] != D4[, 2], ])

## [1] 429 2

unique(D4[D4[, 1] != D4[, 2], ])

## [,1] ,[2]
## [1,] " 13" " -7"
## [2,] " NA" " -7"
## [3,] "106" " -7"
## [4,] " 87" " -7"
## [5,] "102" " -7"
## [6,] "164" "191"
## [7,] " 44" " -7"
## [8,] "262" " -7"
## [9,] " 89" "138"
## [10,] " 90" " -7"
## [11,] "112" " -7"
## [12,] "175" " -7"
## [13,] " 52" " -7"
## [14,] " 32" " -7"
## [15,] " 31" "170"
## [16,] "209" " -7"
## [17,] " 29" " -7"
## [18,] " 76" " -7"
## [19,] " 37" " -7"
## [20,] " 14" " -7"
## [21,] "  9" " -7"
## [22,] "217" " -7"
## [23,] "223" " -7"
## [24,] "183" " -7"
## [25,] " 36" " -7"
## [26,] " 24" " -7"
## [27,] "  5" " -7"
## [28,] "286" " -7"
## [29,] " 83" "124"
## [30,] "156" " -7"
## [31,] "114" " -7"
## [32,] "108" " -7"
## [33,] "226" " -7"
## [34,] "110" " -7"
## [35,] "120" " -7"
## [36,] " 50" "140"
## [37,] " 63" " -7"
## [38,] "219" " -7"
## [39,] " 48" " -7"
## [40,] " 49" " -7"
## [41,] "161" " -7"
## [42,] " 92" " -7"
## [43,] "132" " -7"
## [44,] "129" " -7"
## [45,] " 98" "101"
## [46,] "134" " -7"
## [47,] "236" "247"
## 

.results$Fin.c.nm == "ykageF" & results$Raw.c.nm == "ykageR", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 7952374 ykageF 200 ykageR 201 101 884

429 individuals have a different level. There is no bijective mapping.

D5 <- cbind(Raw$ykidR, Fin$ykidF)
dim(D5[D5[, 1] != D5[, 2], ])

## [1] 429 2

unique(D5[D5[, 1] != D5[, 2], ])

results[results$Fin.c.nm == "ykidF" & results$Raw.c.nm == "ykidR", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 7955248 ykidF 9 ykidR 9 20 849

Also for this variable 429 individuals have a different level. There is no bijective mapping.

range(Raw$intnumR)

## [1] "2000" "2108"

results[results$map.di == 0 & results$Raw.c.nm == "intnumR", c(2, 3, 5, 6, 7, 8)]
The old variable 'intnum' has been renamed to 'intid'. The content is unchanged.

table(Raw$he2R)

##
## -3  1  2
## 281 587 206

table(Fin$he2F)

##
##  1  2 NA
## 587 206 281

D7 <- cbind(Raw$he2R, Fin$he2F)
dim(D7[D7[, 1] != D7[, 2], ])

## [1] 281 2

unique(D7[D7[, 1] != D7[, 2], ])

## [,1] [,2]
## [1,] "-3" "NA"

results[results$Fin.c.nm == "he2F" & results$Raw.c.nm == "he2R", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 7696585 he2F 3 he2R 3 0 562

The variable 'he2' has been simply recoded. But this was a simple bivariate mapping.

### D. partner1

Here we compare the data set partner1_dd.dta from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\partner1\Results\compareFinRaw.RData")
```
D.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 0
(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 0

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 0
(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
## [1] 0

Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R$", ",", row.names(Rnv2))
Fnv2$nm <- gsub("F$", ",", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)
(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 89

F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 0

F.nB <- R.nB

(R.n <- R.nA + R.nB)
## [1] 0

(F.n <- F.nA + F.nB)
## [1] 0
```

Release 3.0  89 variables without variation are identical in both data sets.
Release 3.1 89 variables without variation are identical in both data sets.

D.2. variation, but no bijective mapping

```r
rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R$", ",", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F$", ",", as.character(rb$Fin.c.nm))
sum(!RL.bi)
## [1] 3
sum(!FL.bi)
## [1] 3
(R.v <- sum(!(RawNames[!Rnv][!RL.bi] %in% FinNames)))
## [1] 0
(R.vn <- sum(RawNames[!Rnv][!RL.bi] %in% FinNames))
## [1] 3
RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames]
## [1] "pfrt6" "pfrt7" "pfrt9"
(F.v <- sum(!(FinNames[!Fnv][!FL.bi] %in% RawNames)))
## [1] 0
(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))
## [1] 3
FinNames[!Fnv][!FL.bi][FinNames[!Fnv][!FL.bi] %in% RawNames]
## [1] "pfrt6" "pfrt7" "pfrt9"
```

Release 3.0 3 variables share the name with one of the variables in release 3.1 at least.

Release 3.1 3 variables share the name with one of the variables in release 3.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping. Those variables imply different information. DemoDiff data user should check whether they are using those variables for possible effects on their research.
results\$same.name <-
gsub("R$", "", as.character(results\$Raw.c.nm)) == gsub("F$", "", as.character(results\$Fin.c.nm))
sum(results\$same.name & results\$map.di>0)

## [1] 3

print(results[(results\$same.name & results\$map.di>0),c(2,3,5,6,7,8)],row.names=FALSE)

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## pfrt6F 8 pfrt6R 8 2 393
## pfrt7F 9 pfrt7R 9 22 84
## pfrt9F 35 pfrt9R 35 32 149

D.3. variation and bijective mapping

# mark all identical cases
rb\$ident <- (rb\$ed.di.sum == 0)
# mark all not duplicated raw variables
rb\$dup.Raw <- !duplicStrict(rb\$Raw.c.nm)
# mark all not duplicated fin variables
rb\$dup.Fin <- !duplicStrict(rb\$Fin.c.nm)
# mark all equal name pairs
rb\$same.name <- gsub("R$", "", as.character(rb\$Raw.c.nm)) == gsub("F$", "", as.character(rb\$Fin.c.nm))
rb\$Raw.same.name <- (ave(rb\$same.name, rb\$Raw.c.nm, FUN = sum) > 0)
rb\$Fin.same.name <- (ave(rb\$same.name, rb\$Fin.c.nm, FUN = sum) > 0)

(R.vb <- length(unique(rb[!rb\$dup.Raw & !rb\$Raw.same.name & !rb\$Raw.ident, "Raw.c.nm"])))

## [1] 0

(R.vbi <- length(unique(rb[!rb\$dup.Raw & !rb\$Raw.same.name & rb\$Raw.ident, "Raw.c.nm"])))

## [1] 0

(R.vbn <- length(unique(rb[!rb\$dup.Raw & rb\$Raw.same.name & !rb\$Raw.ident, "Raw.c.nm"])))

## [1] 0

R.vbni <- length(unique(rb[!rb\$dup.Raw & rb\$Raw.same.name & rb\$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb\$dup.Raw & rb\$Raw.same.name & rb\$Raw.ident, "Raw.c.nm"]))

## [1] "pidR" "idR"

(R.vb1 <- length(unique(rb[rb\$dup.Raw & !rb\$same.name & !rb\$ident, "Raw.c.nm"])))

44
## [1] 0

(R.vb1 <- length(unique(rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm")))

## [1] 0

R.vbin <- length(unique(rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"))
as.character(unique(rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"))

## [1] "psd27R" "pjob2R"

(R.vb1n <- length(unique(rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm")))

## [1] 135

(F.vb <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))

## [1] 0

(F.vbi <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"])))

## [1] 0

(F.vbn <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"))))

## [1] 0

F.vbni <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"]))
as.character(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"]))

## [1] "pidF" "idF"

(F.vb1 <- length(unique(rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm")))

## [1] 0

(F.vb1i <- length(unique(rb$dup.Fin & !rb$same.name & rb$ident, "Fin.c.nm")))

## [1] 0

F.vb1n <- length(unique(rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"))
as.character(unique(rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"))

## [1] "psd27F" "pjob2F"

(F.vb1ni <- length(unique(rb$dup.Fin & rb$same.name & rb$ident, "Fin.c.nm")))

## [1] 135
D.4. comparison summary for partner1_DD.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbni</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>vb1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1i</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1n</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>vb1ni</td>
<td>135</td>
<td>135</td>
</tr>
<tr>
<td>sum</td>
<td>231</td>
<td>231</td>
</tr>
</tbody>
</table>

Most variables are unchanged (ni, vbni, vb1ni). 2 variables (vb1n) have been probably recoded. 3 variables (vn) have been changed. This analysis provide no further hints for this group.

D.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```r
D1 <- cbind(Raw$pfrt6R, Fin$pfrt6F)
dim(D1[D1[, 1] != D1[, 2], ])
## [1] 377 2

unique(D1[D1[, 1] != D1[, 2], ])
## [,1] [,2]
## [1,] " 0" " 7"
## [2,] " 7" " 5"
## [3,] ",-1" " 6"
## [4,] ",-1" ",-2"

results[results$Fin.c.nm == "pfrt6F" & results$Raw.c.nm == "pfrt6R", c(2, 3, 5, 6, 7, 8)]
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 11298 pfrt6F 8 pfrt6R 8 2 393
```

For 377 individuals the variable 'pfrt6' has been changed. There is no bijective mapping.

```r
D2 <- cbind(Raw$pfrt7R, Fin$pfrt7F)
dim(D2[D2[, 1] != D2[, 2], ])
## [1] 53 2

unique(D2[D2[, 1] != D2[, 2], ])
```
## 

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-4</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-2</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-3</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-4</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

results[results$Fin.c.nm == "pfrt7F" & results$Raw.c.nm == "pfrt7R", c(2, 3, 5, 6, 7, 8)]

For 53 individuals the variable 'pfrt7' has been changed.

D3 <- cbind(Raw$pfrt9R, Fin$pfrt9F)

dim(D3[D3[, 1] != D3[, 2], ])

## [1] 86 2

unique(D3[D3[, 1] != D3[, 2], ])

## 

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-4</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-4</td>
<td>97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-2</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-4</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-4</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-4</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-3</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-4</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-1</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-4</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-4</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-4</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-4</td>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>26</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>-4</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

results[results$Fin.c.nm == "pfrt9F" & results$Raw.c.nm == "pfrt9R", c(2, 3, 5, 6, 7, 8)]

## 

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13729</td>
<td>pfrt9F</td>
<td>35</td>
<td>pfrt9R</td>
<td>35</td>
<td>32</td>
<td>149</td>
</tr>
</tbody>
</table>
For 86 individuals the variable 'pfrt9' has been changed.

D4 <- cbind(Raw$psd27R, Fin$psd27F)
dim(D4[D4[, 1] != D4[, 2], ])

## [1] 390 2

unique(D4[D4[, 1] != D4[, 2], ])

## [,1] [,2]
## [1,] " 4" " 5"
## [2,] " 7" " 8"
## [3,] " 6" " 7"
## [4,] " 5" " 6"

results[results$Fin.c.nm == "psd27F" & results$Raw.c.nm == "psd27R", c(2, 3, 5, 6, 7, 8)]

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 15445 psd27F 9 psd27R 9 0 390

For 390 individuals the variable 'psd27' has been changed. But there is a bijective mapping. It seems that the level notation has been changed.

D5 <- cbind(Raw$pjob2R, Fin$pjob2F)
dim(D5[D5[, 1] != D5[, 2], ])

## [1] 542 2

unique(D5[D5[, 1] != D5[, 2], ])

## [,1] [,2]
## [1,] " 8" "62"
## [2,] "22" "51"
## [3,] "28" "42"
## [4,] "23" "52"
## [5,] " 9" "63"
## [6,] "19" "22"
## [7,] "17" "20"
## [8,] " 7" "61"
## [9,] "24" "53"
## [10,] "18" "21"
## [11,] " 4" "73"
## [12,] "20" "30"
## [13,] " 1" "70"
## [14,] "14" "13"
## [15,] " 6" "60"
## [16,] " 5" "74"
## [17,] " 2" "71"
## [18,] "29" "43"
## [19,] "11" "10"
## [20,] "10" "64"
## [21,] "3" "72"
## 

For 542 individuals the variable 'pjob2' has been changed. There is a bijective mapping.

### E. partner2

Here we compare the data set `partner2_dd.dta` from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with `R` and `Raw`, release 3.1 files with `F` and `Fin`, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\partner2\Results\compareFinRaw.RData")
```

#### E.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Fnv] %in% FinNames)))
```

```r
## [1] 0
```

```r
(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
```

```r
## [1] 0
```

### added or lost variance

```r
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
```

```r
## [1] 0
```

```r
(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
```

```r
## [1] 0
```
Release 3.0 96 variables without variation are identical in both data sets.

Release 3.1 96 variables without variation are identical in both data sets.

E.2. variation, but no bijective mapping

```r
rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R$", "", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F$", "", as.character(rb$Fin.c.nm))
sum(!RL.bi)
## [1] 0

sum(!FL.bi)
## [1] 0

(R.v <- sum(!RawNames[!Rnv][!RL.bi] %in% FinNames)))
## [1] 0

(R.vn <- sum(RawNames[!Rnv][!RL.bi] %in% FinNames))
There is nothing to compare in this subsection.

### E.3. variation and bijective mapping

```r
data (rub)

# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R$", ",", as.character(rb$Raw.c.nm)) == gsub("F$", ",", as.character(rb$Fin.c.nm))

rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)
rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)

(R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0
(R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0
(R.vbn <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0
R.vbni <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"]))
## [1] "idR" "pidR"

(R.vb1 <- length(unique(rb[!rb$dup.Raw & !rb$ident, "Raw.c.nm"])))
## [1] 0
```
(R.vb1i <- length(unique(rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm")))
## [1] 0

(R.vb1n <- length(unique(rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm")))
## [1] 0

(R.vb1ni <- length(unique(rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm")))
## [1] 182

(F.vb <- length(unique(!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vbi <- length(unique(!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vbn <- length(unique(!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vbni <- length(unique(!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm")))
as.character(unique(!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"))
## [1] "idF"  "pidF"

(F.vb1 <- length(unique(rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm")))
## [1] 0

(F.vb1i <- length(unique(rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm")))
## [1] 0

(F.vb1n <- length(unique(rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm")))
## [1] 0

(F.vb1ni <- length(unique(rb$dup.Fin & rb$same.name & rb$ident, "Fin.c.nm")))
## [1] 182
E.4. comparison summary for partner2_DD.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>vbl</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbli</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbln</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbini</td>
<td>182</td>
<td>182</td>
</tr>
<tr>
<td>sum</td>
<td>280</td>
<td>280</td>
</tr>
</tbody>
</table>

All variables are unchanged (ni, vbi, vbl).  

E.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```
results[results$map.di == 0 & results$Raw.c.nm == "idR", c(2, 3, 5, 6, 7, 8)]
```

```
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 1 idF   578   idR   578    0    0
## 185 pidF 578   idR   578    0   1156
```

```
results[results$map.di == 0 & results$Raw.c.nm == "pidR", c(2, 3, 5, 6, 7, 8)]
```

```
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 2 idF   578   pidR  578    0   0
## 186 pidF 578   pidR  578    0    0
```

As expected exists a bivariate mapping between 'id' and 'pid'.

F. partner4

Here we compare the data set partner4_dd.dta from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\partner4\Results\compareFinRaw.RData")
```

53
F.1. no variation

We compare the variables without variation first.

\[
(R.x \leftarrow \text{sum}(\neg (\text{RawNames}[Rnv] \text{in} \text{FinNames})))
\]

## [1] 0

\[
(F.x \leftarrow \text{sum}(\neg (\text{FinNames}[Fnv] \text{in} \text{RawNames})))
\]

## [1] 0

# added or lost variance

\[
(R.nA \leftarrow \text{sum}(\neg (\text{RawNames}[Rnv] \text{in} \text{FinNames}[Fnv])) - R.x)
\]

## [1] 0

\[
(F.nA \leftarrow \text{sum}(\neg (\text{FinNames}[Fnv] \text{in} \text{RawNames}[Rnv])) - F.x)
\]

## [1] 0

Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R", "", row.names(Rnv2))
Fnv2$nm <- gsub("F", "", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)

\[
(R.ni \leftarrow \text{sum}((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
\]

## [1] 2

F.ni <- R.ni

# changed values \(n_B\) (\(n = n_A + n_B\))

\[
(R.nB \leftarrow \text{sum}((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
\]

## [1] 0

F.nB <- R.nB

(R.n <- R.nA + R.nB)

## [1] 0

(F.n <- F.nA + F.nB)

## [1] 0

Release 3.0  2 variables without variation are identical in both data sets.
Release 3.1 2 variables without variation are identical in both data sets.

F.2. variation, but no bijective mapping

```r
r <- results.bijec
RL.bi <- RawNames[Rnv] %in% gsub("R", ",", as.character(r$Raw.c.nm))
FL.bi <- FinNames[Fnv] %in% gsub("F", ",", as.character(r$Fin.c.nm))
sum(!RL.bi)
## [1] 4
sum(!FL.bi)
## [1] 4
(R.v <- sum(!(RawNames[Rnv][!RL.bi] %in% FinNames)))
## [1] 0
(R.vn <- sum(RawNames[Rnv][!RL.bi] %in% FinNames))
## [1] 4
RawNames[Rnv][!RL.bi][RawNames[Rnv][!RL.bi] %in% FinNames]
## [1] "psex5" "psex7" "psat5" "pfrt3"
(F.v <- sum(!(FinNames[Fnv][!FL.bi] %in% RawNames)))
## [1] 0
(F.vn <- sum(FinNames[Fnv][!FL.bi] %in% RawNames))
## [1] 4
FinNames[Fnv][!FL.bi][FinNames[Fnv][!FL.bi] %in% RawNames]
## [1] "psex5" "psex7" "psat5" "pfrt3"
```

Release 3.0 4 variables share the name with one of the variables in release 3.1 at least.

Release 3.1 4 variables share the name with one of the variables in release 3.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping. Those variables imply different information. DemoDiff data user should check whether they are using those variables for possible effects on their research. (The number of pairs may differ from 4 and 4).
```r
results$same.name <-
gsub("R$", ", as.character(results$Raw.c.nm)) == gsub("F$", ", as.character(results$Fin.c.nm))
sum(results$same.name & results$map.di>0)

## [1] 6

print(results[(results$same.name & results$map.di>0),c(2,3,5,6,7,8)],row.names=FALSE)

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## psex5F 6 psex5R 7 3 48
## psex7F 8 psex7R 9 3 81
## psat5F 12 psat5R 13 1 166
## pfrt3F 5 pfrt3R 6 1 20
## pigr51p1i6F 2 pigr51p1i6R 2 2 40
## pigr51p3i6F 2 pigr51p3i6R 2 2 30

F.3. variation and bijective mapping

# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R$", ", as.character(rb$Raw.c.nm)) == gsub("F$", ", as.character(rb$Fin.c.nm))
rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)
rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)

(R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))

## [1] 0

(R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))

## [1] 0

(R.vbn <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))

## [1] 0

(R.vbni <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))
as.character(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"]))

## [1] "idR" "pidR" "pigr51p1i4R" "pigr51p3i4R" "psd19k4mR"
## [6] "psd19k4yR" "psd14k4gR" "psd15k4R"
```
R.vb1 <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))

## [1] "pigr51p1i5R" "pigr51p3i5R"

R.vb1i <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"]))

## [1] "pigr51p1i6R" "pigr51p3i6R"

R.vb1n <- length(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"]))

## [1] "pigr51p1i6R" "pigr51p3i6R"

## [6] "pbc2i1R" "pbc2i2R" "pbc2i3R" "pbc2i4R" "pbc2i5R"

## [11] "pfrt25i1R" "pfrt25i2R" "pfrt25i3R" "pfrt25i4R" "pigr34R"

## [16] "pcrn19i1R" "pcrn19i2R" "pcrn19i3R" "pcrn19i4R" "pcrn19i5R"

## [21] "pcrn19i6R" "pcrn19i7R" "pcrn19i8R" "pcrn19i9R" "pcrn19i10R"

## [26] "pcrn19i11R" "pcrn19i12R"

(R.vb1ni <- length(unique(rb[rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm"])))

## [1] 222

(F.vb <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"))))

## [1] 0

(F.vbi <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"))))

## [1] 0

(F.vbn <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"))))

## [1] 0

F.vbni <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"]))
as.character(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"]))

## [1] "idF" "pidF" "pigr51p1i4F" "pigr51p3i4F" "psd19k4mF"

## [6] "psd19k4yF" "psd19k4gF" "psd15k4F"

F.vb1 <- length(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))
as.character(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))

## [1] "pigr51p1i6F" "pigr51p3i6F"

F.vb1i <- length(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))
as.character(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"]))

## [1] "pigr51p1i7F" "pigr51p3i7F"
F.vb1n <- length(unique(rb[rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"]))
as.character(unique(rb[rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"]))

## [1] "pbce2i1F" "pbce2i2F" "pbce2i3F" "pbce2i4F" "pbce2i5F"
## [6] "pbce2i6F" "pbce2i7F" "pbce2i8F" "pbce2i9F" "pbce2i10F"
## [11] "pfrt25i1F" "pfrt25i2F" "pfrt25i3F" "pfrt25i4F" "pigr34F"
## [16] "pcrn19i1F" "pcrn19i2F" "pcrn19i3F" "pcrn19i4F" "pcrn19i5F"
## [21] "pcrn19i6F" "pcrn19i7F" "pcrn19i8F" "pcrn19i9F" "pcrn19i10F"
## [26] "pcrn19i11F" "pcrn19i12F"

(F.vb1ni <- length(unique(rb[rb$dup.Fin & rb$same.name & rb$ident, "Fin.c.nm"])))

## [1] 222

### F.4. comparison summary for partner4_DD.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th></th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>vbi</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>vbn</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>vb1ni</td>
<td>222</td>
<td>222</td>
</tr>
<tr>
<td>sum</td>
<td>267</td>
<td>267</td>
</tr>
</tbody>
</table>

Most variables are unchanged (ni, vbn, vb1ni). A number of variables (vb1, vbi, vb1n) had been (probably) renamed or recoded. A group of variables (vn) had been changed. This analysis provide no further hints for this group.

### F.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

D1 <- cbind(Raw$psex5R, Fin$psex5F)
dim(D1[D1[, 1] != D1[, 2], ])

## [1] 24 2

unique(D1[D1[, 1] != D1[, 2], ])

## [1,] `[1] [2]
## [1,] "99" "-2"
## [2,] "99" "1"

results[results$Fin.c.nm == "psex5F" & results$Raw.c.nm == "psex5R", c(2, 3, 5, 6, 7, 8)]
24 individuals have a changed value for 'psex5'. It seems, that the level '99' has been splitted into the new levels '-2' and '1'.

```r
D2 <- cbind(Raw$psat5R, Fin$psat5F)
dim(D2[D2[, 1] != D2[, 2], ])
```

```r
## [1] 83 2
```

```r
unique(D2[D2[, 1] != D2[, 2], ])
```

```r
## [,1] [,2]
## [1,] "99" "-2"
```

```r
results[results$Fin.c.nm == "psat5F" & results$Raw.c.nm == "psat5R", c(2, 3, 5, 6, 7, 8)]
```

```r
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 13035 psat5F 12 psat5R 13 1 166
```

An additional level '-2' has been introduced for the variable 'psat5'.

```r
results[results$map.di == 0 & results$Raw.c.nm == "pig5p1i6R", c(2, 3, 5, 6, 7, 8)]
```

```r
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 36177 pig5p1i7F 2 pig5p1i6R 2 0 0
```

```r
table(Raw$pig5p1i6R)
```

```r
## table(Fin$pig5p1i7F)
```

Potentially, it was either a simple renaming or it is a random effect.

```r
D4 <- cbind(Raw$pbce2i1R, Fin$pbce2i1F)
dim(D4[D4[, 1] != D4[, 2], ])
```

```r
## [1] 49 2
```

```r
unique(D4[D4[, 1] != D4[, 2], ])
```
## 

## 

### 

results[results$Fin.c.nm == "pbce2i1F" & results$Raw.c.nm == "pbce2i1R", c(2, 3, 5, 6, 7, 8)]

---

The level '9' has been probably recoded to '7'.

D5 <- cbind(Raw$pcrn19i1R, Fin$pcrn19i1F)
dim(D5[D5[, 1] != D5[, 2], ])

---

unique(D5[D5[, 1] != D5[, 2], ])

---

results[results$Fin.c.nm == "pcrn19i1F" & results$Raw.c.nm == "pcrn19i1R", c(2, 3, 5, 6, 7, 8)]

---

The results points to a systematic level recoding.

D6 <- cbind(Raw$pigr34R, Fin$pigr34F)
dim(D6[D6[, 1] != D6[, 2], ])

---

unique(D6[D6[, 1] != D6[, 2], ])

---

results[results$Fin.c.nm == "pigr34F" & results$Raw.c.nm == "pigr34R", c(2, 3, 5, 6, 7, 8)]

---

For 5 individuals the level has been recoded.
G. biopart

Here we compare the data set `biopart.dta` from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\biopart\Results\compareFinRaw.RData")
```

G.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 0
(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 0

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 0
(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
## [1] 0

Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R$", ",", row.names(Rnv2))
Fnv2$nm <- gsub("F$", ",", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)

(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 21
F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum(!(RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 0

61
F.nB <- R.nB

(R.n <- R.nA + R.nB)

## [1] 0

(F.n <- F.nA + F.nB)

## [1] 0

**Release 3.0**  21 variables without variation are identical in both data sets.

**Release 3.1**  21 variables without variation are identical in both data sets.

**G.2. variation, but no bijective mapping**

rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R$", ",", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F$", ",", as.character(rb$Fin.c.nm))

sum(!RL.bi)

## [1] 0

sum(!FL.bi)

## [1] 0

(R.v <- sum(!RawNames[!Rnv][!RL.bi] %in% FinNames))

## [1] 0

(R.vn <- sum(RawNames[!Rnv][!RL.bi] %in% FinNames))

## [1] 0

(F.v <- sum(!FinNames[!Fnv][!FL.bi] %in% RawNames))

## [1] 0

(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))

## [1] 0

In this subsection is nothing to compare.
G.3. variation and bijective mapping

```r
# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R$", "", as.character(rb$Raw.c.nm)) ==
gsub("F$", "", as.character(rb$Fin.c.nm))
rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)
rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)
(R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0
(R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))
## [1] 0
(R.vbn <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0
R.vbni <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"]))
## [1] "b2begR"  "b2endR"  "b3begR"  "b3endR"  "b2cohbegR"  "b2cohendR"
(R.vb1 <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"])))
## [1] 0
(R.vb1i <- length(unique(rb[rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 0
(R.vb1n <- length(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"])))
## [1] 0
(R.vb1ni <- length(unique(rb[rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 33
```
(F.vb <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))
## [1] 0

(F.vbi <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"])))
## [1] 0

(F.vbn <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))
## [1] 0

F.vbni <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"]))
as.character(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"]))
## [1] "b2begF"  "b3begF"  "b2endF"  "b3endF"  "b2cohbegF"  "b2cohendF"

(F.vb1 <- length(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"])))
## [1] 0

(F.vb1i <- length(unique(rb[rb$dup.Fin & !rb$same.name & rb$ident, "Fin.c.nm"])))
## [1] 0

(F.vb1n <- length(unique(rb[rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"])))
## [1] 0

(F.vb1ni <- length(unique(rb[rb$dup.Fin & rb$same.name & rb$ident, "Fin.c.nm"])))
## [1] 33

G.4. comparison summary for biopart.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>vb1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1i</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1ni</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>sum</td>
<td>60</td>
<td>60</td>
</tr>
</tbody>
</table>

64
All variables are unchanged (ni, vbni, vb1ni).

**G.5. selected in-depth comparison**

For illustrative purposes we present some detailed results.

```r
results[results$map.di == 0 & results$Raw.c.nm == "b2begR", c(2, 3, 5, 6, 7, 8)]
```

## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 921  b2begF  7  b2begR  7  0  0
## 1038 b2endF  7  b2begR  7  0  11

```r
table(Raw$b2begR)
```

```r
##
## -3 1118 1278 1284 1308 1321 1327
## 2961 1 1 1 1 1 1
```

```r
table(Fin$b2begF)
```

```r
##
## -3 1118 1278 1284 1308 1321 1327
## 2961 1 1 1 1 1 1
```

```r
table(Fin$b2endF)
```

```r
##
## -3 1156 1282 1303 1309 1331 1333
## 2961 1 1 1 1 1 1
```

The variable 'b2beg' has two bivariate mappings. One of them is the identical one.

**H. biochild**

Here we compare the data set `biochild.dta` from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with `R` and `Raw`, release 3.1 files with `F` and `Fin`, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\biochild\Results\compareFinRaw.RData")
```
H.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 0

(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 0

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 0

(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
## [1] 0

Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R", "", row.names(Rnv2))
Fnv2$nm <- gsub("F", "", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)

(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 1

F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 0

F.nB <- R.nB

(R.n <- R.nA + R.nB)
## [1] 0

(F.n <- F.nA + F.nB)
## [1] 0
```

Release 3.0 1 variable without variation is identical in both data sets.
**Release 3.1** 1 variable without variation is identical in both data sets.

**H.2. variation, but no bijective mapping**

```r
rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R$", ",", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F$", ",", as.character(rb$Fin.c.nm))
sum(!RL.bi)
## [1] 1
sum(!FL.bi)
## [1] 1
(R.v <- sum(!(RawNames[!Rnv][!RL.bi] %in% FinNames)))
## [1] 0
RawNames[!Rnv][!RL.bi][!(RawNames[!Rnv][!RL.bi] %in% FinNames)]
## character(0)
(R.vn <- sum(RawNames[!Rnv][!RL.bi] %in% FinNames))
## [1] 1
RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames]
## [1] "livkbeg"
(F.v <- sum(!(FinNames[!Fnv][!FL.bi] %in% RawNames)))
## [1] 0
FinNames[!Fnv][!FL.bi][!(FinNames[!Fnv][!FL.bi] %in% RawNames)]
## character(0)
(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))
## [1] 1
FinNames[!Fnv][!FL.bi][FinNames[!Fnv][!FL.bi] %in% RawNames]
## [1] "livkbeg"
```

**Release 3.0** 1 variables share the name with one of the variables in release 3.1 at least.
Release 3.1

1 variables share the name with one of the variables in release 3.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping. Those variables imply different information. DemoDiff data user should check whether they are using those variables for possible effects on their research.

```r
results$same.name <-
gsub("R$","",as.character(results$Raw.c.nm)) == gsub("F$","",as.character(results$Fin.c.nm))
sum(results$same.name & results$map.di>0)
```

## [1] 1

```r
print(results[(results$same.name & results$map.di>0),c(2,3,5,6,7,8)],row.names=FALSE)
```

##

<table>
<thead>
<tr>
<th>Fin.c.nm</th>
<th>Fin.c.ls</th>
<th>Raw.c.nm</th>
<th>Raw.c.ls</th>
<th>map.di</th>
<th>ed.di.sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>livkbegF</td>
<td>244</td>
<td>livkbegR</td>
<td>244</td>
<td>18</td>
<td>19</td>
</tr>
</tbody>
</table>

H.3. variation and bijective mapping

```r
# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("R$","",as.character(rb$Raw.c.nm)) == gsub("F$","",as.character(rb$Fin.c.nm))
```

```r
rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)
rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)
```

```r
(R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
```

## [1] 0

```r
(R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))
```

## [1] 0

```r
(R.vbn <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
```

## [1] 0

```r
(R.vbni <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))
```

## [1] 0
(R.vb1 <- length(unique(rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm")))
## [1] 0

(R.vb1i <- length(unique(rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm")))
## [1] 0

(R.vb1n <- length(unique(rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm")))
## [1] 0

(R.vb1ni <- length(unique(rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm")))
## [1] 23

(F.vb <- length(unique(!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vb1 <- length(unique(!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vb1i <- length(unique(!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vb1n <- length(unique(!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm")))
## [1] 0

(F.vb1ni <- length(unique(!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm")))
## [1] 23
H.4. comparison summary for biochild.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbln</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi1</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>sum</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

Most variables are unchanged (ni, vb1ni). 1 variables (vn) has been changed.

H.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```r
D1 <- cbind(Raw$livkbegR, Fin$livkbegF)
dim(D1[D1[, 1] != D1[, 2], ,])
## [1] 9 2
unique(D1[D1[, 1] != D1[, 2], ,])
## [,1] [,2]
## [1,] "1144" "1168"
## [2,] "1167" "1155"
## [3,] "1276" "1288"
## [4,] "1234" "1210"
## [5,] "1258" "1270"
## [6,] "1214" "1226"
## [7,] "1274" "1238"
## [8,] "1237" "1213"
## [9,] "1307" "1295"
results[results$Fin.c.nm == "livkbegF" & results$Raw.c.nm == "livkbegR", c(2, 3, 5, 6, 7, 8)]
## Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 451 livkbegF 244 livkbegR 244 18 19
```

This variables has been recoded for 9 individuals. The number of levels is high. This simplifies a bivariate mapping.
I. weights

Here we compare the data set weights.dta from DemoDiff release 3.0 with the same data set from DemoDiff release 3.1.

We have to load the produced data collection. release 3.0 files are denoted with R and Raw, release 3.1 files with F and Fin, respectively.

```r
rm(list = ls())
duplicStrict <- function(A) {
  return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\compareR3R31\weights\Results\compareFinRaw.RData")
```

I.1. no variation

We compare the variables without variation first.

```r
(R.x <- sum(!(RawNames[Rnv] %in% FinNames)))
## [1] 0

(F.x <- sum(!(FinNames[Fnv] %in% RawNames)))
## [1] 0

# added or lost variance
(R.nA <- sum(!(RawNames[Rnv] %in% FinNames[Fnv])) - R.x)
## [1] 0

(F.nA <- sum(!(FinNames[Fnv] %in% RawNames[Rnv])) - F.x)
## [1] 0

Rnv2 <- data.frame(Rnv.tab, stringsAsFactors = FALSE)
Fnv2 <- data.frame(Fnv.tab, stringsAsFactors = FALSE)
Rnv2$nm <- gsub("R\$", "", row.names(Rnv2))
Fnv2$nm <- gsub("F\$", "", row.names(Fnv2))
RFnv2 <- merge(Rnv2, Fnv2, by = c("nm"), all = TRUE)

(R.ni <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 0

F.ni <- R.ni

# changed values nB (n = nA + nB)
(R.nB <- sum((RFnv2$Raw.no.var == RFnv2$Fin.no.var), na.rm = TRUE))
## [1] 0
```
There is nothing to compare in this subsection.

### I.2. variation, but no bijective mapping

```r
rb <- results.bijec
RL.bi <- RawNames[!Rnv] %in% gsub("R", "", as.character(rb$Raw.c.nm))
FL.bi <- FinNames[!Fnv] %in% gsub("F", "", as.character(rb$Fin.c.nm))
sum(!RL.bi)  # [1] 0
sum(!FL.bi)  # [1] 0
(R.v <- sum(!(RawNames[!Rnv][!RL.bi] %in% FinNames)))  # [1] 0
(R.vn <- sum(RawNames[!Rnv][!RL.bi] %in% FinNames))  # [1] 0
(F.v <- sum(!(FinNames[!Fnv][!FL.bi] %in% RawNames)))  # [1] 0
(F.vn <- sum(FinNames[!Fnv][!FL.bi] %in% RawNames))  # [1] 0
```

There is nothing to compare in this subsection.

### I.3. variation and bijective mapping

```r
# mark all identical cases
rb$ident <- (rb$ed.di.sum == 0)
# mark all not duplicated raw variables
```

72
```r
rb$dup.Raw <- !duplicStrict(rb$Raw.c.nm)
# mark all not duplicated fin variables
rb$dup.Fin <- !duplicStrict(rb$Fin.c.nm)
# mark all equal name pairs
rb$same.name <- gsub("$", "", as.character(rb$Raw.c.nm)) == gsub("$", "", as.character(rb$Fin.c.nm))

rb$Raw.same.name <- (ave(rb$same.name, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.same.name <- (ave(rb$same.name, rb$Fin.c.nm, FUN = sum) > 0)

rb$Raw.ident <- (ave(rb$ident, rb$Raw.c.nm, FUN = sum) > 0)
rb$Fin.ident <- (ave(rb$ident, rb$Fin.c.nm, FUN = sum) > 0)

(R.vb <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0

(R.vbi <- length(unique(rb[!rb$dup.Raw & !rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))
## [1] 0

(R.vbn <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & !rb$Raw.ident, "Raw.c.nm"])))
## [1] 0

(R.vbni <- length(unique(rb[!rb$dup.Raw & rb$Raw.same.name & rb$Raw.ident, "Raw.c.nm"])))
## [1] 0

(R.vb1 <- length(unique(rb[rb$dup.Raw & !rb$same.name & !rb$ident, "Raw.c.nm"])))
## [1] 0

(R.vb1i <- length(unique(rb[rb$dup.Raw & !rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 0

(R.vb1n <- length(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"])))
## [1] 0

(R.vb1ni <- length(unique(rb[rb$dup.Raw & rb$same.name & rb$ident, "Raw.c.nm"])))
## [1] 14

(F.vb <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))
## [1] 0

(F.vbi <- length(unique(rb[!rb$dup.Fin & !rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"])))
```
## 

(F.vbn <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"])))

## 

(F.vbni <- length(unique(rb[!rb$dup.Fin & rb$Fin.same.name & rb$Fin.ident, "Fin.c.nm"])))

## 

(F.vb1 <- length(unique(rb[rb$dup.Fin & !rb$same.name & !rb$ident, "Fin.c.nm"])))

## 

(F.vb1i <- length(unique(rb[rb$dup.Fin & !rb$same.name & rb$ident, "Fin.c.nm"])))

## 

(F.vb1n <- length(unique(rb[rb$dup.Fin & rb$same.name & !rb$ident, "Fin.c.nm"])))

## 

(F.vb1ni <- length(unique(rb[rb$dup.Fin & rb$same.name & rb$ident, "Fin.c.nm"])))

## 

I.4. comparison summary for weights.dta release 3.0 and release 3.1

<table>
<thead>
<tr>
<th>class</th>
<th>release 3</th>
<th>release 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ni</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbn</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vbni</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1i</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1n</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vb1ni</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>sum</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>

All variables are unchanged (vb1ni).
References

[DemoDiff 2.0] Kreyenfeld, Michaela; Goldstein, Joshua; Walke, Rainer; Trappe, Heike; Huinink, Johannes (2013): Demographic Differences in Life Course Dynamics in Eastern and Western Germany (DemoDiff). GESIS Datenarchiv, Köln. ZA5684 Datenfile Version 2.0.0, http://dx.doi.org/doi:10.4232/demodiff.5684.2.0.0

[DemoDiff 3.0] Kreyenfeld, Michaela; Goldstein, Joshua; Walke, Rainer; Trappe, Heike; Huinink, Johannes (2013): Demographic Differences in Life Course Dynamics in Eastern and Western Germany (DemoDiff). GESIS Data Archive, Cologne. ZA5684 Data file Version 3.0.0, http://dx.doi.org/doi:10.4232/demodiff.5684.3.0.0

[DemoDiff 3.1] Kreyenfeld, Michaela; Goldstein, Joshua; Walke, Rainer; Trappe, Heike; Huinink, Johannes (2013): Demographic Differences in Life Course Dynamics in Eastern and Western Germany (DemoDiff). GESIS Data Archive, Cologne. ZA5684 Data file Version 3.1.0, http://dx.doi.org/doi:10.4232/demodiff.5684.3.1.0

