Max-Planck-Institut für demografische Forschung
Max Planck Institute for Demographic Research
Konrad-Zuse-Strasse 1 · D-18057 Rostock · GERMANY
Tel +49 (0) 3 81 20 81 - 0; Fax +49 (0) 3 81 20 81 - 202;
http://www.demogr.mpg.de

MPIDR TECHNICAL REPORT 2014-001
MARCH 2014

Comparison of DemoDiff Releases 3.0 and 3.1

Rainer Walke (walke@demogr.mpg.de)

For additional material see www.demogr.mpg.de/tr/

This technical report has been approved for release by: Dirk Vieregg (vieregg@demogr.mpg.de), Coordinator of the IT Group.

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

*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 [vb] No mapped variable from **B** is identical.

2.2112 [vbi] 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.[2] 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.

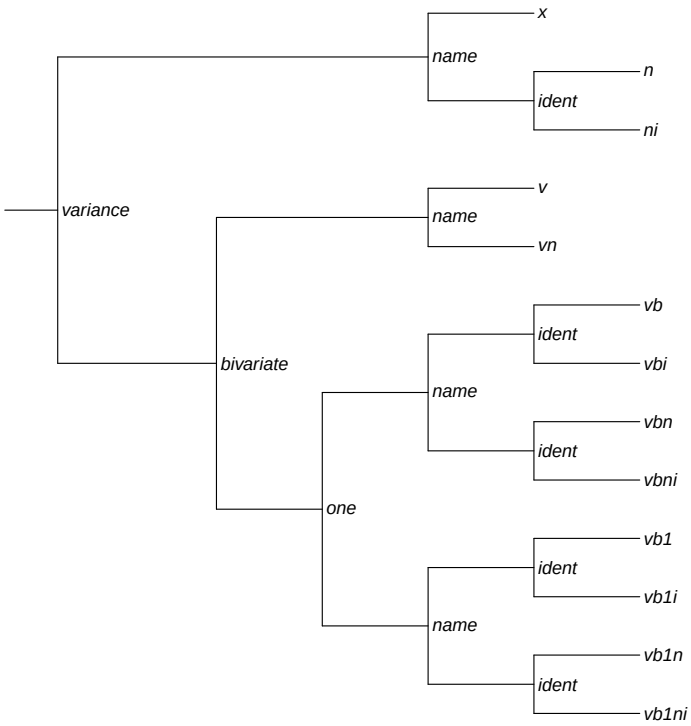

Figure 1: Classification tree.

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 data set 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.

[2] 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.

class	release 3 compared to release 3	release 3 compared to release 3
x	0	0
n	0	0
ni	649	649
v	0	0
vn	0	0
vb	0	0
vbi	0	0
vbn	0	0
vbni	1038	1038
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	1557	1557
sum	3244	3244

Table 1: Classification of the example file `anchor2_DD.dta`, release 3.0 in relation to itself and vice versa.

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.

class	release 3 compared to release 3^{ren}	release 3^{ren} compared to release 3
x	649	649
n	0	0
ni	0	0
v	0	0
vn	0	0
vb	0	0
vbi	1038	1038
vbn	0	0
vbni	0	0
vb1	0	0
vb1i	1556	1556
vb1n	0	0
vb1ni	1	1
sum	3244	3244

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.

class	release 3 compared to empty data	empty data compared to release 3
x	649	0
n	0	0
ni	0	0
v	2594	0
vn	0	0
vb	0	0
vbi	0	0
vbn	0	0
vbni	0	0
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	1	1
sum	3244	1

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 .

class	release 3 compared to release 3.1	release 3.1 compared to release 3
x	0	211
n	58	124
ni	591	591
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
sum	3244	3455

Table 4: Classification of the example file `anchor2_DD.dta`, release 3.0 in relation to release 3.1 and vice versa.

In the following part of this section we discuss the classification of the file `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.[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.[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.[5] It is possible that these variables have only been recoded. However, these data need to be inspected still.

[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.

[4] The provided Levenshtein distance (Appendix) gives a rough idea whether only some cases have been changed or many.

[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.

```
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.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
```

```r
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"
```

```
# nA (n = nA + nB)
yF <- RFnv2[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 <- 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] 26

RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames]

##  [1] "sd19k1y"    "sd19k2y"    "sd19k3y"    "sdp10i1"    "sdp10i14"
##  [6] "hc9h1p2"    "sex_gen"    "doby_gen"   "dobm_gen"   "k1doby_gen"
## [11] "k2doby_gen" "k3doby_gen" "mage"       "homosex"    "hhsizemrd"
## [16] "mmrd"       "fmrd"       "hhcomp"     "pschool"    "pisced"
## [21] "pcasmin"    "pyeduc"     "hhincgcee"  "pcasprim"   "pcassec"
## [26] "plfs"

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

```
## [1] 1

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

## [1] "childmrd"

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

## [1] 25

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

##  [1] "sd4g"       "sd14k1g"    "sd19k1y"    "sd19k2y"    "sd19k3y"
##  [6] "sdp10i1"    "sdp10i14"   "hc9h1p2"    "k1doby_gen" "k2doby_gen"
## [11] "k3doby_gen" "mage"       "homosex"    "hhsizemrd"  "mmrd"
## [16] "fmrd"       "hhcomp"     "pschool"    "pisced"     "pcasmin"
## [21] "pyeduc"     "hhincgcee"  "pcasprim"   "pcassec"    "plfs"
```

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)

##     Fin.c.nm Fin.c.ls    Raw.c.nm Raw.c.ls map.di ed.di.sum
##        sd4gF        3       sd4gR        3      2         1
##     sd14k1gF        5    sd14k1gR        5      2         1
##     sd19k1yF       31    sd19k1yR       31     14         7
##     sd19k2yF       27    sd19k2yR       27      6         6
##     sd19k3yF       20    sd19k3yR       20      2         1
##     sdp10i1F        4    sdp10i1R        4      2         1
##    sdp10i14F        4   sdp10i14R        4      2         1
##     hc9h1p2F       18    hc9h1p2R       18      8        25
##     sex_genF        2    sex_genR        2      4         2
##    doby_genF        6   doby_genR        6      8         4
##    dobm_genF       12   dobm_genR       12      6         4
```

```
##  k1doby_genF       31 k1doby_genR       31     14         7
##  k2doby_genF       27 k2doby_genR       27      6         6
##  k3doby_genF       20 k3doby_genR       20      2         1
##        mageF       43       mageR       43      2         1
##     homosexF        3    homosexR        3      2         1
##   hhsizemrdF        9  hhsizemrdR        9      6         3
##        mmrdF        2       mmrdR        2      2        17
##        fmrdF        2       fmrdR        3      3        20
##      hhcompF       14     hhcompR       15      9        38
##     pschoolF       11    pschoolR       11     24       233
##      piscedF       11     piscedR       11      8        14
##     pcasminF       12    pcasminR       12     14        20
##      pyeducF       20     pyeducR       21     51       243
##   hhincgceeF      330  hhincgceeR      330      2         8
##    pcasprimF       22   pcasprimR       22      2         2
##     pcassecF       17    pcassecR       17      2         1
##        plfsF       13       plfsR       13      2         2
```

A.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"])))

## [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$dup.Fin & !rb$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"])))

## [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

class	release 3	release 3.1
x	0	26
n	1	1
ni	422	422
v	2	1
vn	26	25
vb	0	0
vbi	0	0
vbn	0	0
vbni	224	223
vb1	0	0
vb1i	2	3
vb1n	1	1
vb1ni	724	725
sum	1402	1427

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.

```
table(Raw$flag26R)

##
##  -10
## 1489

table(Fin$flag26F)

##
##    0
## 1489
```

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

```
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], ]
```

```
## [1] " 2" " 1"

results[results$Fin.c.nm == "sd4gF" & results$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
## 23521    sd4gF        3    sd4gR        3      2         1
```

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.

```
D3 <- cbind(Raw$pschoolR, Fin$pschoolF)
dim(D3[D3[, 1] != D3[, 2], ])

## [1] 233   2

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

##       [,1] [,2]
##  [1,] " 6" " 7"
##  [2,] " 4" " 3"
##  [3,] " 5" " 6"
##  [4,] " 7" " 8"
##  [5,] " 7" " 6"
##  [6,] " 4" " 6"
##  [7,] " 4" " 7"
##  [8,] " 3" " 6"
##  [9,] " 4" " 2"
## [10,] " 4" " 8"
## [11,] " 8" " 6"
## [12,] " 7" " 3"

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

##        Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 924140 pschoolF       11 pschoolR       11     24       233
```

233 individuals have a new level for 'pschool'. It takes 233 Levenshtein steps to get them equal. There is no bijective mapping available.

```
D4 <- cbind(Raw$mageR, Fin$mageF)
D4[D4[, 1] != D4[, 2], ]

## [1] "64" "63"

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

##        Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 858483    mageF       43    mageR       43      2         1
```

One individual has a different level. There is no bijective mapping.

```
D5 <- cbind(Raw$pcasminR, Fin$pcasminF)
dim(D5[D5[, 1] != D5[, 2], ])

## [1] 20  2

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

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

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.

```
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$map.di == 0 & results$Raw.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$map.di == 0 & results$Fin.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.

```
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.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)
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] 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] "cid5"        "cid6"        "cid7"        "cid8"        "cid9"
## [11] "ehc10k10h1"  "ehc10k10h2"  "ehc11k10"    "ehc11k10m"   "ehc11k10y"
## [16] "ehc12k10"    "ehc5p2"      "ehc5p3"      "ehc7k10g"    "ehc7k10n"
## [21] "ehc8k10d"    "ehc8k10m"    "ehc8k10y"    "fid"         "igr73i1"
## [26] "igr73i10"    "igr73i11"    "igr73i12"    "igr73i12o"   "igr73i2"
## [31] "igr73i3"     "igr73i4"     "igr73i5"     "igr73i6"     "igr73i7"
## [36] "igr73i8"     "igr73i9"     "igr74"       "igr74o"      "igr77i1"
## [41] "igr77i10"    "igr77i11"    "igr77i12"    "igr77i2"     "igr77i3"
## [46] "igr77i4"     "igr77i5"     "igr77i6"     "igr77i7"     "igr77i8"
## [51] "igr77i9"     "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"         "d350"        "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"
```

```
##  [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"   "ehc28p4m13"   "ehc28p4m14"   "ehc28p4m15"   "ehc28p4m16"
##  [96] "ehc28p4m17"   "ehc28p5m1"    "ehc28p5m2"    "ehc28p5m3"    "ehc28p5m4"
## [101] "ehc28p5m5"    "ehc28p5m6"    "ehc28p5m7"    "ehc28p5m8"    "ehc28p5m9"
## [106] "ehc28p5m10"   "ehc28p5m11"   "ehc28p5m12"   "ehc28p5m13"   "ehc28p5m14"
## [111] "ehc28p5m15"   "ehc28p5m16"   "ehc28p5m17"   "ehc22p9n"     "ehc22p10n"
## [116] "ehc22p11n"    "ehc22p12n"    "ehc22p13n"    "ehc23p9"      "ehc23p10"
## [121] "ehc23p11"     "ehc23p12"     "ehc23p13"     "ehc24p9m"     "ehc24p10m"
## [126] "ehc24p11m"    "ehc24p12m"    "ehc24p13m"    "ehc24p9y"     "ehc24p10y"
## [131] "ehc24p11y"    "ehc24p12y"    "ehc24p13y"    "ehc25p9h1"    "ehc25p10h1"
## [136] "ehc25p11h1"   "ehc25p12h1"   "ehc25p13h1"   "ehc25p9h2"    "ehc25p10h2"
## [141] "ehc25p11h2"   "ehc25p12h2"   "ehc25p13h2"   "ehc25p9h3"    "ehc25p10h3"
## [146] "ehc25p11h3"   "ehc25p12h3"   "ehc25p13h3"   "sep4k5"       "sep7k5"
## [151] "sep8k5"       "sep9k5"       "sep10k5"      "crn23k5"      "crn23k7"
## [156] "crn13k10i1"   "crn13k10i2"   "crn13k10i3"   "crn13k10i4"   "crn13k10i5"
## [161] "crn13k10i6"   "crn13k10i7"   "crn13k10i8"   "crn13k10i9"   "crn13k10i10"
## [166] "crn13k10i11"  "crn13k10i12"  "crn13k10i14"  "crn13k10i13"  "crn14k10i1"
## [171] "crn14k10i2"   "crn14k10i3"   "crn14k10i4"   "crn14k10i5"   "crn14k10i6"
## [176] "crn14k10i7"   "crn14k10i8"   "crn14k10i9"   "crn14k10i10"  "crn14k10i11"
## [181] "crn14k10i12"  "crn14k10i14"  "crn14k10i13"  "crn15k10"     "rtr23h12"
## [186] "rtr24h12"     "exp_di"       "tag_identp"   "flag_ehc"     "flag_frt6"
## [191] "mschool"      "fschool"      "mvocat"       "fvocat"       "mcasmin"
## [196] "fcasmin"      "misced"       "fisced"       "myeduc"       "fyeduc"
## [201] "psweight"     "dweight"      "dxpsweight"   "ppanel"       "pcontact"
## [206] "panswer"      "intcont"      "intdur"       "intid"        "intsex"
## [211] "intage"
```

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

```
##  [1] "crn10k2i1" "crn10k2i2" "crn10k2i3" "crn10k2i4" "crn10k3i1"
##  [6] "crn10k3i2" "crn10k3i3" "crn10k3i4" "crn10k4i1" "crn10k4i2"
## [11] "crn10k4i3" "crn10k4i4" "crn10k5i1" "crn10k5i2" "crn10k5i3"
## [16] "crn10k5i4" "crn1k2"    "crn1k2o"   "crn1k3"    "crn1k4"
## [21] "crn1k5"    "crn2k2i1"  "crn2k2i2"  "crn2k3i1"  "crn2k3i2"
## [26] "crn2k4i1"  "crn2k4i2"  "crn2k5i1"  "crn2k5i2"  "crn3k2"
## [31] "crn3k3"    "crn3k4"    "crn3k5"    "crn4k2"    "crn4k3"
## [36] "crn4k4"    "crn4k5"    "crn5k2"    "crn5k3"    "crn5k4"
## [41] "crn5k5"    "crn6k2"    "crn6k3"    "crn6k4"    "crn6k5"
## [46] "crn7k2"    "crn7k3"    "crn7k4"    "crn7k5"    "igr82i12o"
## [51] "igr85i12o" "tag_dob"   "tag_dobk1" "tag_dobk2" "tag_dobk3"
## [56] "tag_dobk5" "tag_sex"   "tag_sexk1" "tag_sexk3" "tag_sexk4"
## [61] "tag_sexk5" "tag_sexk6" "tag_sexk7" "tag_sexk8" "tag_sexk9"
## [66] "tag_sexp"
```

Release 3.0 For 0 variable without variation exists no variable with the same name in the comparison data set. 58 variables without variation have been changed. 591 variables without variation are identical in both data sets.

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

```
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"         "d89"        "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_sexk2"
## [51] "tag_dobk1"  "tag_dobk2"  "tag_dobk3"  "tag_sexp"   "k1doby_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"
```

```
(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_sexk2"  "k1doby_gen"
## [11] "pschool"    "pisced"     "pisced2"    "pcasmin"    "pyeduc"
```

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).

```
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[(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         5
##         d89F       29        d89R       29     14       965
##         d90F       26        d90R       26      6      1576
##         d91F       19        d91R       19      2      2127
##     ehc1p1gF        3    ehc1p1gR        3      2         1
##     ehc8k1yF       30    ehc8k1yR       30      4         2
##     rtr23h3F        6    rtr23h3R        6      2         2
##         hv1F        3        hv1R        4      1        92
##   tag_sexk2F        2  tag_sexk2R        3      1        31
##  k1doby_genF       30 k1doby_genR       30      4         2
##     pschoolF       10    pschoolR       10     22        55
##      piscedF       11     piscedR       11      8         7
##     pisced2F       11    pisced2R       11      8         7
##     pcasminF       13    pcasminR       13     10        11
##      pyeducF       19     pyeducR       20     39        50
```

B.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$Raw.ident, "Raw.c.nm"]))
as.character(unique(rb[!rb$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" "d539R" "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.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] "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.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$Fin.ident, "Fin.c.nm"]))
as.character(unique(rb[!rb$dup.Fin & rb$Fin.same.name & !rb$Fin.ident, "Fin.c.nm"]))
```

```
##  [1] "d33F"  "d35F"  "d36F"  "d48F"  "d49F"  "d50F"  "d51F"  "d348F"
##  [9] "d349F" "d95F"  "d96F"  "d108F" "d109F" "d110F" "d111F" "d123F"
## [17] "d124F" "d125F" "d126F" "d328F" "d329F" "d330F" "d331F" "d332F"
## [25] "d333F" "d334F" "d139F" "d158F" "d518F" "d519F" "d520F" "d521F"
## [33] "d522F" "d538F" "d539F" "d540F" "d541F" "d542F"
```

```
(F.vbni <- 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"  "d45F"  "d46F"  "d47F"  "d342F"
##  [9] "d343F" "d344F" "d345F" "d346F" "d347F" "d92F"  "d93F"  "d94F"
## [17] "d105F" "d106F" "d107F" "d120F" "d121F" "d122F" "d155F" "d156F"
## [25] "d157F" "d196F" "d197F" "d198F" "d199F" "d200F" "d217F" "d283F"
## [33] "d317F" "d397F" "hm1F"  "hsv1F" "hsm1F"
```

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

```
## [1] 1496
```

B.4. comparison summary for `anchor2_DD.dta` release 3.0 and release 3.1

class	release 3	release 3.1
x	0	211
n	58	124
ni	591	591
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
sum	3244	3455

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.

```
table(Raw$afnat1R)

##
##   NA
## 1173

table(Fin$afnat1F)

##
##   -3
## 1173

table(Raw$midR)

##
##  -10
## 1173

table(Fin$midF)

##
##   NA
## 1173

table(Raw$tag_dobk1R)
```

```
##
##    0    1
## 1167    6

table(Fin$tag_dobk1F)

##
##    0
## 1173
```

The variables 'afnat1' and 'mid' have been recoded. The third variable 'tag_dobk1' lost all variance.

```
table(Raw$pschoolR)

##
##  -3  -7   1   2   3   4   5   6   7   8
## 258  12  25  63 217  26 232  64 275   1

table(Fin$pschoolF)

##
##  -3  -7   1   2   3   4   5   6   7   8
## 258  12  25  63 226  20 220  48 298   3

D2 <- cbind(Raw$pschoolR, Fin$pschoolF)
dim(D2[D2[, 1] != D2[, 2], ])

## [1] 55  2

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

##       [,1] [,2]
##  [1,] " 5" " 6"
##  [2,] " 6" " 7"
##  [3,] " 7" " 6"
##  [4,] " 4" " 3"
##  [5,] " 5" " 3"
##  [6,] " 5" " 7"
##  [7,] " 7" " 8"
##  [8,] " 4" " 8"
##  [9,] " 4" " 6"
## [10,] " 3" " 6"
## [11,] " 7" " 3"

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

##         Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 6477085 pschoolF       10 pschoolR       10     22        55
```

Both variables share the same number of levels. It does not exists a bijective mapping. We have to recode 55 individal 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 where recoded from '-3' to 'NA'. There is no bijective mapping.

```
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.

```
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.

```
table(Raw$tag_sexk3R)

##
##    0    1
## 1161   12

table(Fin$tag_sexk3F)
```

```
##
##    0
## 1173

results[results$map.di == 0 & results$Raw.c.nm == "tag_sexk3R", c(2, 3, 5, 6,
    7, 8)]

##           Fin.c.nm Fin.c.ls   Raw.c.nm Raw.c.ls map.di ed.di.sum
## 6105845 tag_biok3F        2 tag_sexk3R        2      0         0
```

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], ])
```

```
## [1] 923   2

unique(D10)

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

results[results$Fin.c.nm == "hsm1F" & results$Raw.c.nm == "hsm1R", c(2, 3, 5,
    6, 7, 8)]

##         Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 5999408    hsm1F        3    hsm1R        3      0      1846
```

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.

```
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.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)
```

```
## [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)])
```

```
##   [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] "ehc8k9y"     "ehc8k10y"    "ehc12k9"     "ehc12k10"    "ehc13k5"
##  [86] "ehc13k6"     "ehc27p5i2"   "ehc28p5"     "ehc23p11"    "ehc23p12"
##  [91] "ehc23p13"    "ehc24p11m"   "ehc24p12m"   "ehc24p13m"   "ehc24p11y"
##  [96] "ehc24p12y"   "ehc24p13y"   "ehc25p11"    "ehc25p12"    "ehc25p13"
## [101] "ehc10k9h1"   "ehc10k10h1"  "sd35i4"      "sd35i5"      "sep4k3"
## [106] "sep4k4"      "crn2k5i1"    "crn2k5i2"    "crn2k5i3"    "crn2k6i1"
## [111] "crn2k6i2"    "crn2k6i3"    "crn31k9"     "crn13k9i1"   "crn13k9i2"
## [116] "crn13k9i3"   "crn13k9i4"   "crn13k9i5"   "crn13k9i6"   "crn13k9i7"
## [121] "crn13k9i8"   "crn13k9i9"   "crn13k9i10"  "crn13k9i11"  "crn13k9i12"
## [126] "crn13k9i14"  "crn13k9i13"  "crn14k9i1"   "crn14k9i2"   "crn14k9i3"
## [131] "crn14k9i4"   "crn14k9i5"   "crn14k9i6"   "crn14k9i7"   "crn14k9i8"
## [136] "crn14k9i9"   "crn14k9i10"  "crn14k9i11"  "crn14k9i12"  "crn14k9i14"
## [141] "crn14k9i13"  "crn15k9"     "crn31k10"    "crn13k10i1"  "crn13k10i2"
## [146] "crn13k10i3"  "crn13k10i4"  "crn13k10i5"  "crn13k10i6"  "crn13k10i7"
## [151] "crn13k10i8"  "crn13k10i9"  "crn13k10i10" "crn13k10i11" "crn13k10i12"
## [156] "crn13k10i14" "crn13k10i13" "crn14k10i1"  "crn14k10i2"  "crn14k10i3"
## [161] "crn14k10i4"  "crn14k10i5"  "crn14k10i6"  "crn14k10i7"  "crn14k10i8"
## [166] "crn14k10i9"  "crn14k10i10" "crn14k10i11" "crn14k10i12" "crn14k10i14"
## [171] "crn14k10i13" "crn15k10"    "ccs1k1"      "ccs2k1"      "ccs1k2"
## [176] "ccs2k2"      "ccs1k3"      "ccs2k3"      "ccs1k4"      "ccs2k4"
## [181] "ccs1k5"      "ccs2k5"      "ccs1k6"      "ccs2k6"      "ccs1k7"
## [186] "ccs2k7"      "cpas1"       "cpas2"       "cpas3"       "cpas4"
## [191] "igr51p2i1"   "igr51p2i2"   "igr51p2i3"   "igr51p2i4"   "igr51p2i6"
## [196] "igr51p4i1"   "igr51p4i2"   "igr51p4i3"   "igr51p4i4"   "igr51p4i6"
## [201] "cprs1p1"     "cprs2p1"     "cprs3p1"     "cprs1p2"     "cprs2p2"
## [206] "cprs3p2"     "cprs1p3"     "cprs2p3"     "cprs3p3"     "cprs1p4"
## [211] "cprs2p4"     "cprs3p4"     "net1p17"     "net1p18"     "net1p19"
## [216] "net1p20"     "net1p21"     "net1p22"     "net1p23"     "net15p1"
## [221] "net16p1"     "net15p2"     "net16p2"     "net15p3"     "net16p3"
## [226] "net15p4"     "net16p4"     "net15p5"     "net16p5"     "net15p6"
## [231] "net16p6"     "net15p7"     "net16p7"     "net15p8"     "net16p8"
## [236] "net15p9"     "net16p9"     "net15p10"    "net16p10"    "net15p11"
## [241] "net16p11"    "net15p12"    "net16p12"    "net15p13"    "net16p13"
## [246] "net15p14"    "net16p14"    "net15p15"    "net16p15"    "net15p16"
## [251] "net16p16"    "net15p17"    "net16p17"    "net8p18"     "net9p18"
## [256] "net15p18"    "net16p18"    "net8p19"     "net15p19"    "net16p19"
## [261] "net15p20"    "net16p20"    "net15p21"    "net16p21"    "net7p22"
## [266] "net8p22"     "net9p22"     "net10p22"    "net11p22"    "net12p22"
## [271] "net13p22"    "net14p22"    "net15p22"    "net16p22"    "net15p23"
## [276] "net16p23"    "net8p24"     "net9p24"     "net15p24"    "net16p24"
```

```
## [281] "net7p25"     "net10p25"    "net11p25"    "net12p25"    "net13p25"
## [286] "net14p25"    "net7p26"     "net8p26"     "net9p26"     "net10p26"
## [291] "net11p26"    "net12p26"    "net13p26"    "net14p26"    "net8p27"
## [296] "net9p27"     "net15p27"    "net16p27"    "net7p28"     "net10p28"
## [301] "net11p28"    "net12p28"    "net13p28"    "net14p28"    "net15p28"
## [306] "net16p28"    "cas1"        "cas2"        "tag_sexk9"   "tag_sexk10"
## [311] "tag_dobk9"   "tag_dobk10"  "tag_biok9"   "tag_biok10"  "tag_biokp9"
## [316] "tag_biokp10" "tag_identp"  "flag23"      "flag_ehc"    "flag_frt6"
## [321] "ykagecapi"   "ykidcapi"    "mschool"     "fschool"     "mvocat"
## [326] "fvocat"      "mcasmin"     "fcasmin"     "misced"      "fisced"
## [331] "myeduc"      "fyeduc"      "psweight"    "dweight"     "dxpsweight"
## [336] "lweight"     "ppanel"      "pcontact"    "panswer"     "intcont"
## [341] "intsex"      "intage"
```

```
# 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 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.

Release 3.1 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

```
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] 27
```

```
sum(!FL.bi)
```

```
## [1] 17
```

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

```
## [1] 13
```

```
RawNames[!Rnv][!RL.bi][!(RawNames[!Rnv][!RL.bi] %in% FinNames)]
```

```
##  [1] "d175"   "d271"   "d322"   "d323"   "d501"   "d502"   "hpbm"
##  [8] "hpby"   "hp0em"  "hp0c"   "hp0mt0" "cps8i1" "cps8i2"
```

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

## [1] 14

RawNames[!Rnv][!RL.bi][RawNames[!Rnv][!RL.bi] %in% FinNames]

##  [1] "d9"        "d89"       "hpg"       "tag_dobk1" "tag_dobk2"
##  [6] "tag_sexp"  "mage"      "ykage"     "ykid"      "pschool"
## [11] "pisced"    "pisced2"   "pcasmin"   "pyeduc"

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

## [1] 3

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

## [1] "erw0" "erw1" "erw2"

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

## [1] 14

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

##  [1] "cid2"    "cid3"    "cid4"    "d9"      "d89"     "hpg"     "mage"
##  [8] "ykage"   "ykid"    "pschool" "pisced"  "pisced2" "pcasmin" "pyeduc"
```

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).

```
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

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
##      mageF       42     mageR       42      2         1
##     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

```
# 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$Raw.ident, "Raw.c.nm"])))

## [1] 1047

(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] "intnumR"
```

```
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] "he2R"
```

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

```
## [1] 1793
```

```
(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"])))
```

```
## [1] 1047
```

```
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] "cid5F"
```

```
(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] "he2F"
```

```
(F.vb1ni <- length(unique(rb[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

class	release 3	release 3.1
x	16	342
n	10	9
ni	886	886
v	13	3
vn	14	14
vb	3	0
vbi	1	0
vbn	0	0
vbni	1047	1047
vb1	0	1
vb1i	1	0
vb1n	1	1
vb1ni	1793	1793
sum	3785	4096

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.

```
table(Raw$cid1R)

##
##   -3
## 1074

table(Fin$cid1F)

##
##   NA
## 1074

table(Raw$cid2R)

##
##   -3
## 1074

table(Fin$cid2F)

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

The variable 'cid1' has been recoded. The variable 'cid2' has gained some variation.

```
table(Raw$mageR)
```

```
##
## -3 -7 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
## 82 51  2  3  1  6 21 22 36 34 47 43 44 43 34 34 55 48 53 46 59 56 29 31 18
## 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 83 85
## 19 30 25 10 12 22 13  6 12  8  5  9  1  1  1  1  1
```

```
table(Fin$mageF)
```

```
##
## -3 -7 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
## 82 51  2  3  1  6 21 22 36 34 47 43 44 43 34 34 55 48 53 46 59 56 29 31 17
## 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 83 85
## 20 30 25 10 12 22 13  6 12  8  5  9  1  1  1  1  1
```

```
D2 <- cbind(Raw$mageR, Fin$mageF)
D2[D2[, 1] != D2[, 2], ]
```

```
## [1] "65" "66"
```

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

```
##         Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 7917886    mageF       42    mageR       42      2         1
```

Both variables share the same number of levels. It exists no bijective mapping. One individual has been recoded frome '65' to '66'.

```
D3 <- cbind(Raw$hpgR, Fin$hpgF)
dim(D3[D3[, 1] != D3[, 2], ])
```

```
## [1] 2 2
```

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

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

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

```
##         Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 3486165     hpgF        3     hpgR        3      4         2
```

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"
```

```
## [48,] " 38" " -7"
## [49,] " 54" " -7"
## [50,] " 66" "106"
## [51,] " 88" " -7"
## [52,] "100" " -7"
## [53,] "242" " -7"
## [54,] " 65" " -7"
## [55,] " 81" "160"
## [56,] "133" " -7"
## [57,] "  1" " -7"
```

```
results[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], ])
```

```
##       [,1] [,2]
##  [1,] " 1" "-7"
##  [2,] "NA" "-7"
##  [3,] " 3" "-7"
##  [4,] " 3" " 2"
##  [5,] " 2" "-7"
##  [6,] " 6" "-7"
##  [7,] " 8" " 5"
##  [8,] " 4" "-7"
##  [9,] " 4" " 3"
## [10,] " 2" " 1"
## [11,] " 4" " 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)]
```

```
##         Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 8199549   intidF       83  intnumR       83      0         0
```

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.

```
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.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

```
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)

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] "pidR" "idR"
```

```
(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$ident, "Raw.c.nm"]))
as.character(unique(rb[rb$dup.Raw & rb$same.name & !rb$ident, "Raw.c.nm"]))
```

```
## [1] "psd27R" "pjob2R"
```

```
(R.vb1ni <- length(unique(rb[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[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] "psd27F" "pjob2F"
```

```
(F.vb1ni <- length(unique(rb[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

class	release 3	release 3.1
x	0	0
n	0	0
ni	89	89
v	0	0
vn	3	3
vb	0	0
vbi	0	0
vbn	0	0
vbni	2	2
vb1	0	0
vb1i	0	0
vb1n	2	2
vb1ni	135	135
sum	231	231

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.

```
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.

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

## [1] 53  2

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

```
##       [,1] [,2]
##  [1,] "-4" " 1"
##  [2,] "-4" "-1"
##  [3,] "-4" " 3"
##  [4,] "-2" "-3"
##  [5,] "-4" " 2"
##  [6,] "-4" " 4"
##  [7,] "-3" "-2"
##  [8,] "-4" " 7"
##  [9,] " 1" "-4"
## [10,] " 7" "-4"
## [11,] " 4" "-4"
```

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

```
##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 13586   pfrt7F        9   pfrt7R        9     22        84
```

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], ])
```

```
##       [,1] [,2]
##  [1,] "-1" "97"
##  [2,] "-4" "-2"
##  [3,] "-4" "97"
##  [4,] "-2" "-3"
##  [5,] " 3" "41"
##  [6,] " 4" "38"
##  [7,] "-4" "28"
##  [8,] "-4" "30"
##  [9,] "-4" "38"
## [10,] "-4" "35"
## [11,] "-3" "-2"
## [12,] "-4" "34"
## [13,] "-1" "-4"
## [14,] "-4" "29"
## [15,] "-4" "24"
## [16,] "-4" "26"
## [17,] "-4" "43"
## [18,] "26" "-4"
## [19,] "-4" "48"
```

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

```
##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 13729   pfrt9F       35   pfrt9R       35     32       149
```

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"
```

```
## [22,] "21" "50"
## [23,] "27" "41"
## [24,] "15" "14"
## [25,] "26" "40"
## [26,] "25" "54"
## [27,] "16" "15"
## [28,] "12" "11"
```

```
results[results$Fin.c.nm == "pjob2F" & results$Raw.c.nm == "pjob2R", c(2, 3,
    5, 6, 7, 8)]
```

```
##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 18734   pjob2F       30   pjob2R       30      0      1056
```

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.

```
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.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] 96

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 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

```
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.

E.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"
```

```
(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] 182
```

```
(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"
```

```
(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] 182
```

E.4. comparison summary for `partner2_DD.dta` release 3.0 and release 3.1

class	release 3	release 3.1
x	0	0
n	0	0
ni	96	96
v	0	0
vn	0	0
vb	0	0
vbi	0	0
vbn	0	0
vbni	2	2
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	182	182
sum	280	280

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

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      1156
## 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")
```

F.1. no variation

We compare the variables without variation first.

```
(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] 2

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 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

```
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] 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).

```
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] "pbce2i1R"   "pbce2i2R"   "pbce2i3R"   "pbce2i4R"   "pbce2i5R"
##  [6] "pbce2i6R"   "pbce2i7R"   "pbce2i8R"   "pbce2i9R"   "pbce2i10R"
## [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"   "psd14k4gF"   "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

class	release 3	release 3.1
x	0	0
n	0	0
ni	2	2
v	0	0
vn	4	4
vb	0	0
vbi	0	0
vbn	0	0
vbni	8	8
vb1	2	2
vb1i	2	2
vb1n	27	27
vb1ni	222	222
sum	267	267

Most variables are unchanged (ni, vbni, vb1ni). A 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.

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] [,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)]
```

```
##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 12503   psex5F        6   psex5R        7      3        48
```

24 individals have a changed value for 'psex5'. It seems, that the level '99' has been splitted into the new levels '-2' and '1'.

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

## [1] 83  2

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

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

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

##       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'.

```
results[results$map.di == 0 & results$Raw.c.nm == "pigr51p1i6R", c(2, 3, 5,
    6, 7, 8)]

##          Fin.c.nm Fin.c.ls    Raw.c.nm Raw.c.ls map.di ed.di.sum
## 36177 pigr51p1i7F        2 pigr51p1i6R        2      0         0

table(Raw$pigr51p1i6R)

##
##  -9   0
##  15 535

table(Fin$pigr51p1i7F)

##
##  -9   0
##  15 535
```

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

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

## [1] 49  2

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

```
##      [,1] [,2]
## [1,] " 9" " 7"
```

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

```
##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 13301 pbce2i1F        7 pbce2i1R        7      0        49
```

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

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

```
## [1] 39  2
```

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

```
##      [,1] [,2]
## [1,] " 0" " 1"
## [2,] " 9" "10"
## [3,] " 8" " 9"
## [4,] " 7" " 8"
## [5,] " 3" " 4"
## [6,] " 1" " 2"
## [7,] " 4" " 5"
```

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

```
##        Fin.c.nm Fin.c.ls  Raw.c.nm Raw.c.ls map.di ed.di.sum
## 56925 pcrn19i1F       10 pcrn19i1R       10      0        44
```

The results points to a systematic level recoding.

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

```
## [1] 5 2
```

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

```
##      [,1] [,2]
## [1,] "-8" "-9"
```

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

```
##       Fin.c.nm Fin.c.ls Raw.c.nm Raw.c.ls map.di ed.di.sum
## 26335  pigr34F       12  pigr34R       12      0         5
```

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.

```
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.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
```

```
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

```
# 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

class	release 3	release 3.1
x	0	0
n	0	0
ni	21	21
v	0	0
vn	0	0
vb	0	0
vbi	0	0
vbn	0	0
vbni	6	6
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	33	33
sum	60	60

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

G.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```
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

table(Raw$b2begR)

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

table(Fin$b2begF)

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

table(Fin$b2endF)

##
##   -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.

```
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.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

```
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.

```
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

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
##  livkbegF      244 livkbegR      244     18        19
```

H.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"])))

## [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] 23

(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"])))

## [1] 0

(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] 23
```

H.4. comparison summary for `biochild.dta` release 3.0 and release 3.1

class	release 3	release 3.1
x	0	0
n	0	0
ni	1	1
v	0	0
vn	1	1
vb	0	0
vbi	0	0
vbn	0	0
vbni	0	0
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	23	23
sum	25	25

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.

```
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.

```
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.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
```

```
F.nB <- R.nB

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

## [1] 0

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

## [1] 0
```

There is nothing to compare in this subsection.

I.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
```

There is nothing to compare in this subsection.

I.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"])))

## [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"])))
```

```
## [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"])))

## [1] 0

(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] 14
```

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

class	release 3	release 3.1
x	0	0
n	0	0
ni	0	0
v	0	0
vn	0	0
vb	0	0
vbi	0	0
vbn	0	0
vbni	0	0
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	14	14
sum	14	14

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

[TR-2012-003] Walke, Rainer; Müller, Andreas (2012): compareFinRaw.r - an R program to measure the difference between datasets. MPIDR Technical Report TR-2012-003. http://www.demogr.mpg.de/papers/technicalreports/tr-2012-003.pdf

[TR-2013-001] Walke, Rainer (2013): Comparison of DemoDiff Releases 2.0 and 3.0. MPIDR Technical Report TR-2013-001. http://www.demogr.mpg.de/papers/technicalreports/tr-2013-001.pdf

[R 3.1] R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.