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 2013-001
AUGUST 2013

Comparison of DemoDiff Releases 2.0 and 3.0

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 2.0 and 3.0

Rainer Walke, MPIDF Rostock*

2013-Aug-26

Abstract

In this Technical Report, 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 five revised files of DemoDiff Release 3.0 with the older Release 2.0 files. It should provide users of the **DemoDiff** data a comprehensive overview on all manipulations of the data that have occurred from Release 2.0 to Release 3.0.

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

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 this report, we use **compareFinRaw** to compare two different releases of the project **DemoDiff**. We are comparing Release 2.0 [DemoDiff 2.0] and Release 3.0 [DemoDiff 3.0]. Both are available by GESIS (www.gesis.org). We compare the following data sets: `anchor1_DD.dta`, `anchor2_DD.dta`, `partner1_DD.dta`, `partner2_DD.dta` and `weights.dta`[1].

We are using the statistical package R [R 2.15] for all the computations and RStudio, knitR and MiKTeX for the documentation. The output of these programs is a readable list that reports the number and the names of variables that completely match between the two releases, i.e. have the same name and same content. The program also identifies all modified variables such as renamed variables, bijective recoded variables and variables with marked differences. Furthermore we get the names of all variables without variation at all. We hope, that this material helps users of **DemoDiff** to get a clear picture on 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. [2].

---

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

[1] There are further files in the releases. However, they have been omitted because they are not suitable for comparison.

[2] In our case, it took 7 hours to run the program for the largest data set (`anchor2_DD.dta`)

2. Classification

The idea is to classify all variables in data set **A** in respect to their relations to variables from data set **B**.

A typical solution would be first to check whether a variable with the same name is available in data set **B**. If this is true we could distinguish between identical and non identical content. This way we would classify all variable from set **A** into one of three possible categories.

This would give only a rough picture of the differences between the data sets. To detect renamed and recoded variables as well we use a more sophisticated classification scheme.

classification tree

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.

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.

Every variable out of **A** will be assign to exactly one of these 13 classes.

3. Results and Summary

The appendix contains the documented code. For each of the five 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

As an example we display here the summary output that our program has produced for the data set `anchor2_DD.dta`.

class	Release 2	Release 3
x	8	0
n	15	37
ni	612	612
v	1	7
vn	369	359
vb	2	0
vbi	11	1
vbn	2	1
vbni	954	956
vb1	4	5
vb1i	3	2
vb1n	3	4
vb1ni	1262	1260
sum	3246	3244

The results may be interpreted as follows:
Most variables are unchanged (ni, vbni, vb1ni). 612 variables do not have variation and are identical. 954 + 1262 variables (vbni, vb1ni, Release 2) are unchanged, but have variation. There is a weak probability that 954 variables (vbni) show some redundancy.

8 constant variables (x, Release 2) have been dropped. One variable (v, Release 2) with variation has been dropped as well.

15 variables (n, Release 2) have been gained variation or have changed to another constant value.

2 variables (vb, Release 2) have been (less probably) renamed and recoded. 11 variables (vbi, Release 2) have been (less probably) renamed. 2 variables (vbn, Release 2) have been (probably) recoded.

4 variables (vb1, Release 2) have been (probably) renamed and recoded. 3 variables (vb1i, Release 2) have been (probably) renamed. 3 variables (vb1n, Release 2) have been (probably) recoded.

369 variables (vn, Release 2) had been changed between releases. They have the same name, but there exist no bijective mapping. The provided Levenshtein distance (appendix) gives a rough idea whether only some cases have been changed or not.

4. Acknowledgements

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

A. anchor1

Here we compare the data set `anchor1_dd.dta` from DemoDiff Release 2.0 with the same data set from DemoDiff Release 3.0.

We have to load the produced data collection. Release 2.0 files are denoted with **R** and **Raw**, release 3.0 files with **F** and **Fin**, respectively.

```
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\\compareR2R3\\anchor1\\Results\\compareFinRaw.RData")
```

A.1. no variation

We compare the variables without variation first.

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

## [1] 1

(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)
Fnv2 <- data.frame(Fnv.tab)
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] 423

F.ni <- R.ni

# changed values
(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


RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)]

## [1] "k10age"

RFnv2[is.na(RFnv2$Fin.no.var), 1]

## [1] "k10age"


FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)]

## character(0)

RFnv2[is.na(RFnv2$Raw.no.var), 1]

## character(0)
```

Release 2.0 For 1 variables without variation (k10age) exist no variables with the same name in the comparison data set. 423 variables without variation are identical in both data sets.

Release 3.0 423 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] 43

sum(!FL.bi)

## [1] 48
```

```
(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] 43

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

##  [1] "sex"       "dobm"      "sd4g"      "sdp1m"     "sdp1y"
##  [6] "frt11v1i1" "frt11v1i2" "frt11v1i3" "frt11v1i4" "frt11v1i5"
## [11] "frt11v1i6" "frt11v1i7" "frt11v1i8" "frt11v2i1" "frt11v2i2"
## [16] "frt11v2i3" "frt11v2i4" "frt11v2i5" "frt11v2i6" "frt11v2i7"
## [21] "frt11v2i8" "flag1"     "flag18"    "sex_gen"   "psex_gen"
## [26] "dobm_gen"  "pdoby_gen" "pdobm_gen" "age"       "page"
## [31] "mage"      "fage"      "k1age"     "k2age"     "homosex"
## [36] "infertile" "pregnant"  "hhsizemrd" "othmrd"    "hhcomp"
## [41] "isei"      "siops"     "hhincgcee"


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

## [1] 5

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

## [1] "ykage"                "ykid"                 "flag_isco08_kldb2010"
## [4] "kldb2010"             "isco08"

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

## [1] 43

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

##  [1] "sex"       "dobm"      "sd4g"      "sdp1m"     "sdp1y"
##  [6] "frt11v1i1" "frt11v1i2" "frt11v1i3" "frt11v1i4" "frt11v1i5"
## [11] "frt11v1i6" "frt11v1i7" "frt11v1i8" "frt11v2i1" "frt11v2i2"
## [16] "frt11v2i3" "frt11v2i4" "frt11v2i5" "frt11v2i6" "frt11v2i7"
## [21] "frt11v2i8" "flag1"     "flag18"    "sex_gen"   "psex_gen"
## [26] "dobm_gen"  "pdoby_gen" "pdobm_gen" "age"       "page"
## [31] "mage"      "fage"      "k1age"     "k2age"     "homosex"
## [36] "infertile" "pregnant"  "hhsizemrd" "othmrd"    "hhcomp"
## [41] "isei"      "siops"     "hhincgcee"
```

Release 2.0 43 variables share the name with one of the variables in Release 3.0 at least.

Release 3.0 For 5 variables with variation but without any bijective mapping exist no variables with the same name in the Release 2.0 data. 43 variables share the name with one of the variables in Release 2.0 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 43 and 43).

```
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] 43

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
##        sexF        2       sexR        2      4         2
##       dobmF       12      dobmR       12     10         7
##       sd4gF        3      sd4gR        3      4         4
##      sdp1mF       17     sdp1mR       17     20        17
##      sdp1yF       45     sdp1yR       45     30        25
##  frt11v1i1F       10 frt11v1i1R        9      1         1
##  frt11v1i2F       10 frt11v1i2R        9      1         1
##  frt11v1i3F       10 frt11v1i3R        9      1         1
##  frt11v1i4F       10 frt11v1i4R        9      1         1
##  frt11v1i5F       10 frt11v1i5R        9      1         1
##  frt11v1i6F       10 frt11v1i6R        9      1         1
##  frt11v1i7F        9 frt11v1i7R        8      1         1
##  frt11v1i8F       10 frt11v1i8R        9      1         1
##  frt11v2i1F        6 frt11v2i1R        5      1         1
##  frt11v2i2F        6 frt11v2i2R        5      1         1
##  frt11v2i3F        6 frt11v2i3R        5      1         1
##  frt11v2i4F        6 frt11v2i4R        5      1         1
##  frt11v2i5F        6 frt11v2i5R        5      1         1
##  frt11v2i6F        6 frt11v2i6R        5      1         1
##  frt11v2i7F        6 frt11v2i7R        5      1         1
##  frt11v2i8F        6 frt11v2i8R        5      1         1
##      flag1F        2     flag1R        2      4         5
##     flag18F        3    flag18R        3      2         3
##    sex_genF        2   sex_genR        2      4         2
##   psex_genF        3  psex_genR        3      4         4
##   dobm_genF       12  dobm_genR       12     10         7
##  pdoby_genF       45 pdoby_genR       45     30        25
##  pdobm_genF       17 pdobm_genR       17     20        17
##        ageF        9       ageR        9      2         1
##       pageF       45      pageR       45     36        25
##       mageF       43      mageR       43     16        13
##       fageF       46      fageR       46     20        12
##      k1ageF       29     k1ageR       29      4         2
##      k2ageF       26     k2ageR       26      4         3
##    homosexF        3   homosexR        3      4         6
```

```
##   infertileF        3 infertileR        3      2          2
##    pregnantF        4  pregnantR        4      4          6
##   hhsizemrdF        9 hhsizemrdR        9      2          1
##      othmrdF        6    othmrdR        6      2          1
##      hhcompF       15    hhcompR       15      2          1
##        iseiF      174      iseiR       60    462       2486
##       siopsF      177     siopsR       57    478       2529
##   hhincgceeF      330 hhincgceeR      331      1          7
```

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] 225


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] "kldbR" "iscoR"

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] "flag_isco_kldbR"

(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] 703


(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] 225


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] "kldb1992F" "isco88F"

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] "flag_isco88_kldb1992F"

(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] 703
```

A.4. comparison summary for `anchor1_DD.dta` Release 2.0 and Release 3.0

class	Release 2	Release 3
x	1	0
n	0	0
ni	423	423
v	0	5
vn	43	43
vb	0	0
vbi	0	0
vbn	0	0
vbni	225	225
vb1	2	2
vb1i	1	1
vb1n	0	0
vb1ni	703	703
sum	1398	1402

Most variables are unchanged (ni, vbni, vb1ni). A small number (x, v) had been dropped or added. A number of variables (vb1, vb1i) 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.

```
D1 <- cbind(Raw$kldbR, Fin$kldb1992F)
D1[D1[, 1] != D1[, 2], ]

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

Both variables share the same number of levels. The level '-1' has been recoded to '-7'. Changing '-1' to '-7' for two individuals takes 2 Levenshtein steps (see details in [TR-2012-003]).

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

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

Both variables share the same number of levels. $12 + 12 + 10 = 34$, $34/2 = 17$ mappings between the levels are in use. For a bijective mapping we need 5 mappings less. Coincidentally 5 individual show a changed level. It takes 7 Levenshtein steps to make the levels equal.

```
D3 <- cbind(Raw$pageR, Fin$pageF)
D3[D3[, 1] != D3[, 2], ]
```

```
##       [,1] [,2]
##  [1,] "-7" "36"
##  [2,] "21" "23"
##  [3,] "25" "24"
##  [4,] "21" "22"
##  [5,] "-7" "46"
##  [6,] "20" "30"
##  [7,] "26" "27"
##  [8,] "25" "24"
##  [9,] "25" "26"
## [10,] "38" "37"
## [11,] "35" "37"
## [12,] "34" "33"
## [13,] "26" "27"
## [14,] "21" "22"
## [15,] "30" "33"
## [16,] "29" "28"
## [17,] "42" "41"
## [18,] "20" "19"
## [19,] "32" "33"
## [20,] "44" "45"
## [21,] "26" "27"
## [22,] "26" "24"
```

22 idividals have a new level for 'page'. It takes 25 Levenshtein steps to get them equal. There is no bijective mapping available.

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

## [1] 918   2

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

##      [,1]   [,2]
## [1,] "  23" "5153"
## [2,] "  53" "  37"
## [3,] "  38" "  51"
## [4,] "  48" "3435"
## [5,] "  51" "  53"
## [6,] "  51" "3344"
```

918 indiduals do have a different level. There is no bijective mapping.

B. anchor2

Using the same tools we compare the data set `anchor2_dd.dta` from DemoDiff Release 2.0 with the same data set from DemoDiff Release 3.0.

At first we have to load the produced data collection. Release 2.0 files are denoted with **R** and **Raw**, release 3.0 files with **F** and **Fin**, respectively.

```
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\\compareR2R3\\anchor2\\Results\\compareFinRaw.RData")
```

B.1. no variation

We compare the variables without variation first.

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

## [1] 8

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

## [1] 0


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

## [1] 15

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

## [1] 37


Rnv2 <- data.frame(Rnv.tab)
Fnv2 <- data.frame(Fnv.tab)
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] 612

F.ni <- R.ni

# changed values
(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] 15
```

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

## [1] 37


RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)]

## [1] "d402" "d403" "d404" "d405" "d406" "d411" "d412" "d420"

RFnv2[is.na(RFnv2$Fin.no.var), 1]

##  [1] "d402"      "d403"      "d404"      "d405"      "d406"
##  [6] "d411"      "d412"      "d420"      "ehc27p4i1" "ehc27p4i2"
## [11] "ehc28p2m1" "ehc28p2m2" "ehc28p3m1" "ehc28p3m2" "ehc28p3m3"
## [16] "ehc4p2m10" "ehc4p2m11" "ehc4p2m12" "ehc4p2m13" "ehc4p2m14"
## [21] "ehc4p2m15" "ehc4p2m16" "ehc4p2m17"


FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)]

## character(0)

RFnv2[is.na(RFnv2$Raw.no.var), 1]

##  [1] "ehc10k5h2" "ehc10k6h2" "ehc10k7h2" "ehc10k8h2" "ehc10k9h2"
##  [6] "ehc16m1"   "ehc16m2"   "ehc16m3"   "ehc17"     "ehc17m10"
## [11] "ehc17m11"  "ehc17m12"  "ehc17m13"  "ehc17m14"  "ehc17m15"
## [16] "ehc17m16"  "ehc17m17"  "ehc17m4"   "ehc17m5"   "ehc17m6"
## [21] "ehc17m7"   "ehc17m8"   "ehc17m9"   "ehc25p1h3" "ehc25p2h3"
## [26] "ehc25p3h3" "ehc25p4h3" "ehc25p5h2" "ehc25p5h3" "ehc25p6h2"
## [31] "ehc25p6h3" "ehc25p7h2" "ehc25p7h3" "ehc25p8h2" "ehc25p8h3"
## [36] "ehc26h3"   "flag25"
```

Release 2.0 For 8 variables without variation (d402, d403, d404, d405, d406, d411, d412, d420) exist no variables with the same name in the comparison data set. 612 variables without variation are identical in both data sets. 15 variables share the same name, but are not identical (got variance, changed value).

Release 3.0 612 variables without variation are identical in both data sets. 37 variables share the same name, but are not identical (lost variance, changed value).

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] 370
```

```r
sum(!FL.bi)

## [1] 366


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

## [1] 1

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

## [1] "isco"

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

## [1] 369

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

##   [1] "d4"           "sex"          "ehc1p1n"      "ehc1p1g"
##   [5] "ehc2p1"       "ehc3p1"       "ehc4p1"       "ehc1p2n"
##   [9] "ehc1p2g"      "ehc2p2"       "ehc3p2"       "ehc4p2"
##  [13] "ehc6p2"       "ehc4p1m8"     "ehc4p1m9"     "ehc4p1m10"
##  [17] "ehc4p1m11"    "ehc4p1m12"    "ehc4p1m13"    "ehc4p1m14"
##  [21] "ehc4p1m15"    "ehc4p1m16"    "ehc4p1m17"    "ehc2p2m8"
##  [25] "ehc2p2m9"     "ehc2p2m10"    "ehc2p2m11"    "ehc2p2m12"
##  [29] "ehc2p2m13"    "ehc2p2m14"    "ehc2p2m15"    "ehc2p2m16"
##  [33] "ehc2p2m17"    "ehc3p2m8"     "ehc3p2m9"     "ehc3p2m10"
##  [37] "ehc3p2m11"    "ehc3p2m12"    "ehc3p2m13"    "ehc3p2m14"
##  [41] "ehc3p2m15"    "ehc3p2m16"    "ehc3p2m17"    "ehc27p1i2"
##  [45] "ehc28p1"      "ehc29p1"      "ehc27p2i1"    "ehc27p2i2"
##  [49] "ehc28p2"      "ehc29p2"      "ehc27p3i1"    "ehc27p3i2"
##  [53] "ehc28p3"      "ehc29p3"      "ehc28p1m8"    "ehc28p1m9"
##  [57] "ehc28p1m10"   "ehc28p1m11"   "ehc28p1m12"   "ehc28p1m13"
##  [61] "ehc28p1m14"   "ehc28p1m15"   "ehc28p1m16"   "ehc28p1m17"
##  [65] "ehc28p2m3"    "ehc28p2m4"    "ehc28p2m5"    "ehc28p2m6"
##  [69] "ehc28p2m7"    "ehc28p2m8"    "ehc28p2m9"    "ehc28p2m10"
##  [73] "ehc28p2m11"   "ehc28p2m12"   "ehc28p2m13"   "ehc28p2m14"
##  [77] "ehc28p2m15"   "ehc28p2m16"   "ehc28p2m17"   "ehc28p3m4"
##  [81] "ehc28p3m5"    "ehc28p3m6"    "ehc28p3m7"    "ehc28p3m8"
##  [85] "ehc28p3m9"    "ehc28p3m10"   "ehc28p3m11"   "ehc28p3m12"
##  [89] "ehc28p3m13"   "ehc28p3m14"   "ehc28p3m15"   "ehc28p3m16"
##  [93] "ehc28p3m17"   "ehc16m4"      "ehc16m5"      "ehc16m6"
##  [97] "ehc16m7"      "ehc16m8"      "ehc16m9"      "ehc16m10"
## [101] "ehc16m11"     "ehc16m12"     "ehc16m13"     "ehc16m14"
## [105] "ehc16m15"     "ehc16m16"     "ehc16m17"     "ehc17m5"
## [109] "ehc17m6"      "ehc17m7"      "ehc17m8"      "ehc17m9"
## [113] "ehc17m10"     "ehc17m11"     "ehc17m12"     "ehc17m13"
## [117] "ehc17m14"     "ehc17m15"     "ehc17m16"     "ehc17m17"
## [121] "ehc16"        "ehc17"        "ehc22p1n"     "ehc22p2n"
## [125] "ehc22p3n"     "ehc22p4n"     "ehc22p5n"     "ehc23p1"
## [129] "ehc23p2"      "ehc23p3"      "ehc23p4"      "ehc23p5"
## [133] "ehc24p1m"     "ehc24p2m"     "ehc24p3m"     "ehc24p4m"
## [137] "ehc24p5m"     "ehc24p6m"     "ehc24p1y"     "ehc24p2y"
```

```
## [141] "ehc24p3y"      "ehc24p4y"      "ehc24p5y"      "ehc24p6y"
## [145] "ehc25p1h1"     "ehc25p2h1"     "ehc25p3h1"     "ehc25p4h1"
## [149] "ehc25p5h1"     "ehc25p1h2"     "ehc25p2h2"     "ehc25p3h2"
## [153] "ehc25p4h2"     "ehc25p5h2"     "ehc25p7h2"     "ehc25p1h3"
## [157] "ehc25p2h3"     "ehc25p3h3"     "ehc25p4h3"     "ehc25p5h3"
## [161] "ehc25p7h3"     "ehc26h1"       "ehc26h2"       "ehc26h3"
## [165] "ehc10k1h1"     "ehc10k2h1"     "ehc10k3h1"     "ehc10k4h1"
## [169] "ehc10k5h1"     "ehc10k6h1"     "ehc10k7h1"     "ehc10k1h2"
## [173] "ehc10k2h2"     "ehc10k3h2"     "ehc10k4h2"     "hhnr"
## [177] "hh1nr"         "hh2nr"         "hh1a"          "hh2a"
## [181] "hh1l"          "hh2l"          "hh1t0"         "hh2t0"
## [185] "hp"            "hpnr"          "hpn"           "hpg"
## [189] "hp0"           "hpsep"         "hpmt"          "hpm"
## [193] "hpc"           "hps2"          "hp0nr"         "hp0n"
## [197] "hp0g"          "hp0ct0"        "hc6h1"         "hc6h2"
## [201] "job17"         "job18h"        "sat2"          "sin3i1"
## [205] "sin3i2"        "sin1"          "sin2"          "sin6i1"
## [209] "sin6i2"        "sin6i3"        "sin6i4"        "sin6i5"
## [213] "pa2m"          "pa30"          "pa3"           "sdp1d"
## [217] "sdp1m"         "sdp1y"         "sdp2i1"        "sdp2i2"
## [221] "sdp2i3"        "sdp2i4"        "sdp2i5"        "sdp2i6"
## [225] "sdp2i7"        "sdp2i8"        "sdp2i9"        "sdp2i10"
## [229] "sdp2i11"       "sdp2i12"       "sdp3"          "sdp6"
## [233] "sdp21"         "sdp22"         "sdp10i16"      "sdp10i17"
## [237] "sdp10i7"       "sdp10i18"      "sdp10i19"      "sdp10i9"
## [241] "sdp10i20"      "sdp10i21"      "sdp10i22"      "sdp10i1"
## [245] "sdp10i2"       "sdp10i3"       "sdp10i4"       "sdp10i5"
## [249] "sdp10i6"       "sdp10i13"      "sdp10i8"       "sdp10i10"
## [253] "sdp10i11"      "sdp10i12"      "sdp10i15"      "sdp10i14"
## [257] "sdp16"         "sdp17"         "sdp18"         "hcp1i1"
## [261] "hcp1i2"        "hcp2"          "hcp3h"         "hcp3m"
## [265] "hcp4h"         "hcp5"          "pa9"           "pa11"
## [269] "pa13"          "pa14i1"        "pa14i2"        "pa14i3"
## [273] "pa14i4"        "pa14i5"        "pa15"          "cps1"
## [277] "sat3"          "pa26"          "pa27"          "pa28"
## [281] "sep1i1"        "sep1i2"        "sep2"          "sep18"
## [285] "sep12"         "sep13"         "sep14"         "sep17i1"
## [289] "sep17i2"       "sep17i3"       "sep17i4"       "sep16"
## [293] "frt2"          "frt11v1i6"     "frt11v1i7"     "frt11v1i8"
## [297] "frt11v2i6"     "frt11v2i7"     "frt11v2i8"     "crn22p1"
## [301] "crn33p1i1"     "crn33p1i2"     "crn33p1i3"     "crn34p1"
## [305] "igr22"         "igr25"         "igr28"         "igr39p3"
## [309] "igr40p3"       "igr41p1"       "rtr43"         "inc24"
## [313] "inc26i2"       "inc26i3"       "inc27i2"       "inc27i3"
## [317] "int4i6"        "lmhh1"         "lmhh2"         "lvhh1"
## [321] "lvhh2"         "avhh1"         "hv1"           "tag_sex"
## [325] "tag_dobp"      "flag25"        "sex_gen"       "psex_gen"
## [329] "pdoby_gen"     "pdobm_gen"     "page"          "mage"
## [333] "k1age"         "k2age"         "pcob"          "pnat1"
## [337] "marstat"       "np"            "ncoh"          "nmar"
## [341] "meetdur"       "reldur"        "cohabdur"      "mardur"
## [345] "homosex"       "infertile"     "pregnant"      "pnkidsbioalv"
## [349] "res2nd"        "hhsizemrd"     "mmrd"          "fmrd"
## [353] "childmrd"      "othmrd"        "hhcomp"        "penrol"
## [357] "pschool"       "pvocat"        "pisced"        "pisced2"
## [361] "pcasmin"       "pyeduc"        "egp"           "isei"
```

```
## [365] "siops"        "hhincgcee"    "pcasprim"     "pcassec"
## [369] "plfs"
```

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

```
## [1] 7
```

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

```
## [1] "flag_isco88_kldb1992" "flag_isco08_kldb2010" "ykage"
## [4] "ykid"                 "kldb2010"             "isco88"
## [7] "isco08"
```

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

```
## [1] 359
```

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

```
##   [1] "d4"           "sex"          "ehc1p1n"      "ehc1p1g"
##   [5] "ehc2p1"       "ehc3p1"       "ehc4p1"       "ehc1p2n"
##   [9] "ehc1p2g"      "ehc2p2"       "ehc3p2"       "ehc4p2"
##  [13] "ehc6p2"       "ehc4p1m8"     "ehc4p1m9"     "ehc4p1m10"
##  [17] "ehc4p1m11"    "ehc4p1m12"    "ehc4p1m13"    "ehc4p1m14"
##  [21] "ehc4p1m15"    "ehc4p1m16"    "ehc4p1m17"    "ehc2p2m8"
##  [25] "ehc2p2m9"     "ehc2p2m10"    "ehc2p2m11"    "ehc2p2m12"
##  [29] "ehc2p2m13"    "ehc2p2m14"    "ehc2p2m15"    "ehc2p2m16"
##  [33] "ehc2p2m17"    "ehc3p2m8"     "ehc3p2m9"     "ehc3p2m10"
##  [37] "ehc3p2m11"    "ehc3p2m12"    "ehc3p2m13"    "ehc3p2m14"
##  [41] "ehc3p2m15"    "ehc3p2m16"    "ehc3p2m17"    "ehc4p2m8"
##  [45] "ehc4p2m9"     "ehc4p2m10"    "ehc4p2m11"    "ehc4p2m12"
##  [49] "ehc4p2m13"    "ehc4p2m14"    "ehc4p2m15"    "ehc4p2m16"
##  [53] "ehc4p2m17"    "ehc27p1i2"    "ehc28p1"      "ehc29p1"
##  [57] "ehc27p2i1"    "ehc27p2i2"    "ehc28p2"      "ehc29p2"
##  [61] "ehc27p3i1"    "ehc27p3i2"    "ehc28p3"      "ehc29p3"
##  [65] "ehc27p4i1"    "ehc27p4i2"    "ehc28p1m8"    "ehc28p1m9"
##  [69] "ehc28p1m10"   "ehc28p1m11"   "ehc28p1m12"   "ehc28p1m13"
##  [73] "ehc28p1m14"   "ehc28p1m15"   "ehc28p1m16"   "ehc28p1m17"
##  [77] "ehc28p2m1"    "ehc28p2m2"    "ehc28p2m3"    "ehc28p2m4"
##  [81] "ehc28p2m5"    "ehc28p2m6"    "ehc28p2m7"    "ehc28p2m8"
##  [85] "ehc28p2m9"    "ehc28p2m10"   "ehc28p2m11"   "ehc28p2m12"
##  [89] "ehc28p2m13"   "ehc28p2m14"   "ehc28p2m15"   "ehc28p2m16"
##  [93] "ehc28p2m17"   "ehc28p3m1"    "ehc28p3m2"    "ehc28p3m3"
##  [97] "ehc28p3m4"    "ehc28p3m5"    "ehc28p3m6"    "ehc28p3m7"
## [101] "ehc28p3m8"    "ehc28p3m9"    "ehc28p3m10"   "ehc28p3m11"
## [105] "ehc28p3m12"   "ehc28p3m13"   "ehc28p3m14"   "ehc28p3m15"
## [109] "ehc28p3m16"   "ehc28p3m17"   "ehc16m5"      "ehc16m6"
## [113] "ehc16m7"      "ehc16m8"      "ehc16m9"      "ehc16m10"
## [117] "ehc16m11"     "ehc16m12"     "ehc16m13"     "ehc16m14"
## [121] "ehc16m15"     "ehc16m16"     "ehc16m17"     "ehc16"
## [125] "ehc22p1n"     "ehc22p2n"     "ehc22p3n"     "ehc22p4n"
## [129] "ehc22p5n"     "ehc23p1"      "ehc23p2"      "ehc23p3"
## [133] "ehc23p4"      "ehc23p5"      "ehc24p1m"     "ehc24p2m"
```

```
## [137] "ehc24p3m"     "ehc24p4m"     "ehc24p5m"     "ehc24p7m"
## [141] "ehc24p1y"     "ehc24p2y"     "ehc24p3y"     "ehc24p4y"
## [145] "ehc24p5y"     "ehc24p7y"     "ehc25p1h1"    "ehc25p2h1"
## [149] "ehc25p3h1"    "ehc25p4h1"    "ehc25p5h1"    "ehc25p7h1"
## [153] "ehc25p1h2"    "ehc25p2h2"    "ehc25p3h2"    "ehc26h1"
## [157] "ehc26h2"      "ehc10k1h1"    "ehc10k2h1"    "ehc10k3h1"
## [161] "ehc10k4h1"    "ehc10k5h1"    "ehc10k6h1"    "ehc10k7h1"
## [165] "ehc10k1h2"    "ehc10k2h2"    "ehc10k3h2"    "ehc10k4h2"
## [169] "hhnr"         "hh1nr"        "hh2nr"        "hh1a"
## [173] "hh2a"         "hh1l"         "hh2l"         "hh1t0"
## [177] "hh2t0"        "hp"           "hpnr"         "hpn"
## [181] "hpg"          "hp0"          "hpsep"        "hpmt"
## [185] "hpm"          "hpc"          "hps2"         "hp0nr"
## [189] "hp0n"         "hp0g"         "hp0ct0"       "hc6h1"
## [193] "hc6h2"        "job17"        "job18h"       "sat2"
## [197] "sin3i1"       "sin3i2"       "sin1"         "sin2"
## [201] "sin6i1"       "sin6i2"       "sin6i3"       "sin6i4"
## [205] "sin6i5"       "pa2m"         "pa30"         "pa3"
## [209] "sdp1d"        "sdp1m"        "sdp1y"        "sdp2i1"
## [213] "sdp2i2"       "sdp2i3"       "sdp2i4"       "sdp2i5"
## [217] "sdp2i6"       "sdp2i7"       "sdp2i8"       "sdp2i9"
## [221] "sdp2i10"      "sdp2i11"      "sdp2i12"      "sdp3"
## [225] "sdp6"         "sdp21"        "sdp22"        "sdp10i16"
## [229] "sdp10i17"     "sdp10i7"      "sdp10i18"     "sdp10i19"
## [233] "sdp10i9"      "sdp10i20"     "sdp10i21"     "sdp10i22"
## [237] "sdp10i1"      "sdp10i2"      "sdp10i3"      "sdp10i4"
## [241] "sdp10i5"      "sdp10i6"      "sdp10i13"     "sdp10i8"
## [245] "sdp10i10"     "sdp10i11"     "sdp10i12"     "sdp10i15"
## [249] "sdp10i14"     "sdp16"        "sdp17"        "sdp18"
## [253] "hcp1i1"       "hcp1i2"       "hcp2"         "hcp3h"
## [257] "hcp3m"        "hcp5"         "pa9"          "pa11"
## [261] "pa13"         "pa14i1"       "pa14i2"       "pa14i3"
## [265] "pa14i4"       "pa14i5"       "pa15"         "cps1"
## [269] "sat3"         "pa26"         "pa27"         "pa28"
## [273] "sep1i1"       "sep1i2"       "sep2"         "sep18"
## [277] "sep12"        "sep13"        "sep14"        "sep17i1"
## [281] "sep17i2"      "sep17i3"      "sep17i4"      "sep16"
## [285] "frt2"         "frt11v1i6"    "frt11v1i7"    "frt11v1i8"
## [289] "frt11v2i6"    "frt11v2i7"    "frt11v2i8"    "crn22p1"
## [293] "crn33p1i1"    "crn33p1i2"    "crn33p1i3"    "crn34p1"
## [297] "igr22"        "igr25"        "igr28"        "igr39p3"
## [301] "igr40p3"      "igr41p1"      "rtr43"        "inc24"
## [305] "inc26i2"      "inc26i3"      "inc27i2"      "inc27i3"
## [309] "int4i6"       "lmhh1"        "lmhh2"        "lvhh1"
## [313] "lvhh2"        "hv1"          "tag_sex"      "tag_dobp"
## [317] "sex_gen"      "psex_gen"     "pdoby_gen"    "pdobm_gen"
## [321] "page"         "mage"         "k1age"        "k2age"
## [325] "pcob"         "pnat1"        "marstat"      "np"
## [329] "ncoh"         "nmar"         "meetdur"      "reldur"
## [333] "cohabdur"     "mardur"       "homosex"      "infertile"
## [337] "pregnant"     "pnkidsbioalv" "res2nd"       "hhsizemrd"
## [341] "mmrd"         "fmrd"         "childmrd"     "othmrd"
## [345] "hhcomp"       "penrol"       "pschool"      "pvocat"
## [349] "pisced"       "pisced2"      "pcasmin"      "pyeduc"
## [353] "egp"          "isei"         "siops"        "hhincgcee"
## [357] "pcasprim"     "pcassec"      "plfs"
```

Release 2.0 For 1 variables with variation but without any bijective mapping exist no variables with the same name in the Release 3.0 data. 369 variables share the name with one of the variables in Release 3.0 at least.

Release 3.0 For 7 variables with variation but without any bijective mapping exist no variables with the same name in the Release 2.0 data. 359 variables share the name with one of the variables in Release 2.0 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 369 and 359.

```
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] 351

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
##            d4F        2           d4R        2      2         1
##           sexF        2          sexR        2      2         1
##       ehc1p1nF        3      ehc1p1nR        3      2        16
##       ehc1p1gF        3      ehc1p1gR        3      4        14
##        ehc2p1F        3       ehc2p1R        3      2        24
##        ehc3p1F        3       ehc3p1R        3      2        22
##        ehc4p1F        3       ehc4p1R        3      4        21
##       ehc1p2nF        2      ehc1p2nR        2      2         2
##       ehc1p2gF        3      ehc1p2gR        3      2         2
##        ehc2p2F        3       ehc2p2R        3      2         2
##        ehc3p2F        3       ehc3p2R        3      2         2
##        ehc4p2F        2       ehc4p2R        2      2         2
##        ehc6p2F        2       ehc6p2R        2      2         2
##      ehc4p1m8F        3     ehc4p1m8R        3      2         1
##      ehc4p1m9F        3     ehc4p1m9R        3      2         1
##     ehc4p1m10F        2    ehc4p1m10R        2      2         1
##     ehc4p1m11F        2    ehc4p1m11R        2      2         1
##     ehc4p1m12F        2    ehc4p1m12R        2      2         1
##     ehc4p1m13F        2    ehc4p1m13R        2      2         1
##     ehc4p1m14F        2    ehc4p1m14R        2      2         1
##     ehc4p1m15F        2    ehc4p1m15R        2      2         1
##     ehc4p1m16F        2    ehc4p1m16R        2      2         1
##     ehc4p1m17F        2    ehc4p1m17R        2      2         1
##      ehc2p2m8F        3     ehc2p2m8R        3      2         2
##      ehc2p2m9F        3     ehc2p2m9R        3      2         2
##     ehc2p2m10F        3    ehc2p2m10R        2      1      1174
##     ehc2p2m11F        3    ehc2p2m11R        2      1      1174
##     ehc2p2m12F        3    ehc2p2m12R        2      1      1174
##     ehc2p2m13F        3    ehc2p2m13R        2      1      1174
##     ehc2p2m14F        3    ehc2p2m14R        2      1      1174
##     ehc2p2m15F        3    ehc2p2m15R        2      1      1174
##     ehc2p2m16F        3    ehc2p2m16R        2      1      1174
##     ehc2p2m17F        3    ehc2p2m17R        2      1      1174
```

```
##      ehc3p2m8F        2      ehc3p2m8R        3      1         2
##      ehc3p2m9F        3      ehc3p2m9R        3      2         2
##     ehc3p2m10F        3     ehc3p2m10R        2      1      1174
##     ehc3p2m11F        3     ehc3p2m11R        2      1      1174
##     ehc3p2m12F        3     ehc3p2m12R        2      1      1174
##     ehc3p2m13F        3     ehc3p2m13R        2      1      1174
##     ehc3p2m14F        3     ehc3p2m14R        2      1      1174
##     ehc3p2m15F        3     ehc3p2m15R        2      1      1174
##     ehc3p2m16F        3     ehc3p2m16R        2      1      1174
##     ehc3p2m17F        3     ehc3p2m17R        2      1      1174
##      ehc4p2m8F        2      ehc4p2m8R        2      2         2
##      ehc4p2m9F        2      ehc4p2m9R        2      2         2
##     ehc27p1i2F        9     ehc27p1i2R        8      1         2
##       ehc28p1F        2       ehc28p1R        2      2        12
##       ehc29p1F        3       ehc29p1R        3      4        19
##     ehc27p2i1F        3     ehc27p2i1R        2      3        33
##     ehc27p2i2F       16     ehc27p2i2R       15     11        34
##       ehc28p2F        4       ehc28p2R        3      5        33
##       ehc29p2F        4       ehc29p2R        3      1         8
##     ehc27p3i1F        3     ehc27p3i1R        2      1         4
##     ehc27p3i2F        6     ehc27p3i2R        5      1         4
##       ehc28p3F        4       ehc28p3R        3      1         4
##       ehc29p3F        3       ehc29p3R        2      1         3
##     ehc28p1m8F        3     ehc28p1m8R        3      2         1
##     ehc28p1m9F        3     ehc28p1m9R        3      2         2
##    ehc28p1m10F        2    ehc28p1m10R        2      2         4
##    ehc28p1m11F        2    ehc28p1m11R        2      2         4
##    ehc28p1m12F        2    ehc28p1m12R        2      2         5
##    ehc28p1m13F        2    ehc28p1m13R        2      2         9
##    ehc28p1m14F        2    ehc28p1m14R        2      2        10
##    ehc28p1m15F        2    ehc28p1m15R        2      2        10
##    ehc28p1m16F        2    ehc28p1m16R        2      2        11
##    ehc28p1m17F        2    ehc28p1m17R        2      2        12
##     ehc28p2m3F        3     ehc28p2m3R        2      1         3
##     ehc28p2m4F        4     ehc28p2m4R        3      3         5
##     ehc28p2m5F        4     ehc28p2m5R        3      3        16
##     ehc28p2m6F        4     ehc28p2m6R        3      3        16
##     ehc28p2m7F        4     ehc28p2m7R        3      3        22
##     ehc28p2m8F        4     ehc28p2m8R        3      5        28
##     ehc28p2m9F        4     ehc28p2m9R        3      5        33
##    ehc28p2m10F        4    ehc28p2m10R        3      5        33
##    ehc28p2m11F        4    ehc28p2m11R        3      5        33
##    ehc28p2m12F        4    ehc28p2m12R        3      5        33
##    ehc28p2m13F        4    ehc28p2m13R        3      5        33
##    ehc28p2m14F        4    ehc28p2m14R        3      5        33
##    ehc28p2m15F        4    ehc28p2m15R        3      5        33
##    ehc28p2m16F        4    ehc28p2m16R        3      5        33
##    ehc28p2m17F        4    ehc28p2m17R        3      5        33
##     ehc28p3m4F        3     ehc28p3m4R        2      1         2
##     ehc28p3m5F        3     ehc28p3m5R        2      1         3
##     ehc28p3m6F        3     ehc28p3m6R        2      1         4
##     ehc28p3m7F        4     ehc28p3m7R        3      1         4
##     ehc28p3m8F        4     ehc28p3m8R        3      1         4
##     ehc28p3m9F        4     ehc28p3m9R        3      1         4
##    ehc28p3m10F        4    ehc28p3m10R        3      1         4
##    ehc28p3m11F        4    ehc28p3m11R        3      1         4
```

```
##    ehc28p3m12F        4   ehc28p3m12R        3      1         4
##    ehc28p3m13F        4   ehc28p3m13R        3      1         4
##    ehc28p3m14F        4   ehc28p3m14R        3      1         4
##    ehc28p3m15F        4   ehc28p3m15R        3      1         4
##    ehc28p3m16F        4   ehc28p3m16R        3      1         4
##    ehc28p3m17F        4   ehc28p3m17R        3      1         4
##       ehc16m4F        3      ehc16m4R        3      2       308
##       ehc16m5F        3      ehc16m5R        3      2       660
##       ehc16m6F        3      ehc16m6R        3      4      1000
##       ehc16m7F        4      ehc16m7R        4      4      1412
##       ehc16m8F        4      ehc16m8R        4      4      1822
##       ehc16m9F        4      ehc16m9R        4      4      2276
##      ehc16m10F        4     ehc16m10R        4      4      2280
##      ehc16m11F        3     ehc16m11R        3      4      2280
##      ehc16m12F        3     ehc16m12R        3      4      2282
##      ehc16m13F        3     ehc16m13R        4      5      2281
##      ehc16m14F        4     ehc16m14R        4      6      2282
##      ehc16m15F        3     ehc16m15R        4      5      2280
##      ehc16m16F        3     ehc16m16R        4      5      2285
##      ehc16m17F        3     ehc16m17R        4      5      2284
##         ehc16F        3        ehc16R        4      5      2284
##      ehc22p1nF        5     ehc22p1nR        4      5        63
##      ehc22p2nF        4     ehc22p2nR        5      7        69
##      ehc22p3nF        4     ehc22p3nR        4      6        35
##      ehc22p4nF        4     ehc22p4nR        3      5        13
##      ehc22p5nF        3     ehc22p5nR        3      4         5
##      ehc22p7nF        2     ehc22p7nR        3      1         1
##       ehc23p1F       12      ehc23p1R       12     30        71
##       ehc23p2F       12      ehc23p2R       14     26        74
##       ehc23p3F       12      ehc23p3R       13     23        40
##       ehc23p4F        8      ehc23p4R        8     10        16
##       ehc23p5F        4      ehc23p5R        4      2         4
##      ehc24p1mF       15     ehc24p1mR       14     13        48
##      ehc24p2mF       15     ehc24p2mR       14     15        49
##      ehc24p3mF       14     ehc24p3mR       13      9        28
##      ehc24p4mF        8     ehc24p4mR        8      4        10
##      ehc24p5mF        5     ehc24p5mR        5      2         2
##      ehc24p6mF        4     ehc24p6mR        3      1         1
##      ehc24p7mF        3     ehc24p7mR        2      1         1
##      ehc24p1yF       45     ehc24p1yR       44     13        62
##      ehc24p2yF       42     ehc24p2yR       43     11        64
##      ehc24p3yF       24     ehc24p3yR       24      8        32
##      ehc24p4yF       11     ehc24p4yR       11      4        12
##      ehc24p5yF        6     ehc24p5yR        6      2         2
##      ehc24p6yF        4     ehc24p6yR        3      1         1
##      ehc24p7yF        3     ehc24p7yR        2      1         1
##     ehc25p1h1F        5    ehc25p1h1R        4      5        68
##     ehc25p2h1F        5    ehc25p2h1R        4      7        81
##     ehc25p3h1F        5    ehc25p3h1R        4      3        44
##     ehc25p4h1F        5    ehc25p4h1R        4      5        14
##     ehc25p5h1F        4    ehc25p5h1R        4      2         6
##     ehc25p7h1F        3    ehc25p7h1R        4      1         2
##     ehc25p1h2F        3    ehc25p1h2R        4      3       316
##     ehc25p2h2F        3    ehc25p2h2R        4      3       286
##     ehc25p3h2F        2    ehc25p3h2R        4      2       114
##     ehc25p4h2F        2    ehc25p4h2R        4      2        30
```

```
##       ehc26h1F        3      ehc26h1R        3      6       261
##       ehc26h2F        3      ehc26h2R        3      4      1796
##     ehc10k1h1F        3    ehc10k1h1R        3      6       188
##     ehc10k2h1F        3    ehc10k2h1R        3      4        75
##     ehc10k3h1F        3    ehc10k3h1R        3      4        35
##     ehc10k4h1F        3    ehc10k4h1R        3      4        12
##     ehc10k5h1F        2    ehc10k5h1R        3      3         6
##     ehc10k6h1F        2    ehc10k6h1R        3      1         6
##     ehc10k7h1F        2    ehc10k7h1R        3      1         2
##     ehc10k1h2F        3    ehc10k1h2R        3      4      1448
##     ehc10k2h2F        3    ehc10k2h2R        3      4       836
##     ehc10k3h2F        3    ehc10k3h2R        3      2       244
##     ehc10k4h2F        3    ehc10k4h2R        3      2        84
##          hhnrF        3         hhnrR        3      4      1177
##         hh1nrF        4        hh1nrR        5      9      1197
##         hh2nrF        4        hh2nrR        3      3         3
##          hh1aF        3         hh1aR        2      3         8
##          hh2aF        3         hh2aR        2      3         5
##          hh1lF       13         hh1lR       13      6         9
##          hh2lF       12         hh2lR       11      5         4
##         hh1t0F        2        hh1t0R        2      4         4
##         hh2t0F        3        hh2t0R        3      4         4
##            hpF        2           hpR        2      4         2
##          hpnrF        3         hpnrR        3      4         4
##           hpnF        2          hpnR        2      4         4
##           hpgF        3          hpgR        3      6         5
##           hp0F        2          hp0R        2      2         3
##         hpsepF        2        hpsepR        2      4         2
##          hpmtF        2         hpmtR        2      2         1
##           hpmF        3          hpmR        3      4         4
##           hpcF        3          hpcR        3      2         4
##          hps2F        2         hps2R        2      4         2
##         hp0nrF        2        hp0nrR        2      2         2
##          hp0nF        2         hp0nR        2      4         4
##          hp0gF        3         hp0gR        3      2         2
##        hp0ct0F        3       hp0ct0R        3      4         4
##         hc6h1F        7        hc6h1R        6      1         2
##         hc6h2F        8        hc6h2R        7      1         1
##         job17F        9        job17R        8      1         2
##        job18hF        6       job18hR        5      1         2
##          sat2F       14         sat2R       13      3         2
##        sin3i1F        8       sin3i1R        7      3         2
##        sin3i2F        8       sin3i2R        7      3         2
##          sin1F        6         sin1R        5      3         3
##          sin2F        6         sin2R        5      3         3
##        sin6i1F        9       sin6i1R        8      1         1
##        sin6i2F        9       sin6i2R        8      1         1
##        sin6i3F        9       sin6i3R        8      1         2
##        sin6i4F        9       sin6i4R        8      1         2
##        sin6i5F        9       sin6i5R        8      1         2
##          pa2mF       19         pa2mR       18      1         2
##          pa30F        3         pa30R        3      2         2
##           pa3F       11          pa3R       10      1         1
##         sdp1dF        5        sdp1dR        4      1         2
##         sdp1mF       16        sdp1mR       15      3        15
##         sdp1yF       29        sdp1yR       28      3        23
```

```
##        sdp2i1F        4       sdp2i1R        3      1         2
##        sdp2i2F        3       sdp2i2R        2      1         2
##        sdp2i3F        3       sdp2i3R        2      1         2
##        sdp2i4F        3       sdp2i4R        2      1         2
##        sdp2i5F        3       sdp2i5R        2      1         2
##        sdp2i6F        3       sdp2i6R        2      1         2
##        sdp2i7F        3       sdp2i7R        2      1         2
##        sdp2i8F        3       sdp2i8R        2      1         2
##        sdp2i9F        3       sdp2i9R        2      1         2
##       sdp2i10F        3      sdp2i10R        2      1         2
##       sdp2i11F        3      sdp2i11R        2      1         2
##       sdp2i12F        4      sdp2i12R        3      1         2
##          sdp3F        3         sdp3R        2      1         2
##          sdp6F        5         sdp6R        4      1         2
##         sdp21F        7        sdp21R        6      1         2
##         sdp22F       18        sdp22R       18      4         2
##      sdp10i16F        6     sdp10i16R        6      4         2
##      sdp10i17F        6     sdp10i17R        6      4         2
##       sdp10i7F        6      sdp10i7R        6      4         2
##      sdp10i18F        6     sdp10i18R        6      4         2
##      sdp10i19F        6     sdp10i19R        6      4         2
##       sdp10i9F        6      sdp10i9R        6      4         2
##      sdp10i20F        6     sdp10i20R        6      4         2
##      sdp10i21F        6     sdp10i21R        6      4         2
##      sdp10i22F        5     sdp10i22R        5      4         2
##       sdp10i1F        6      sdp10i1R        6      4         2
##       sdp10i2F        6      sdp10i2R        6      4         2
##       sdp10i3F        6      sdp10i3R        6      4         2
##       sdp10i4F        6      sdp10i4R        6      4         2
##       sdp10i5F        6      sdp10i5R        6      4         2
##       sdp10i6F        6      sdp10i6R        6      4         2
##      sdp10i13F        6     sdp10i13R        6      4         2
##       sdp10i8F        6      sdp10i8R        6      4         2
##      sdp10i10F        5     sdp10i10R        5      4         2
##      sdp10i11F        6     sdp10i11R        6      4         2
##      sdp10i12F        6     sdp10i12R        6      4         2
##      sdp10i15F        6     sdp10i15R        6      4         2
##      sdp10i14F        6     sdp10i14R        6      4         2
##         sdp16F        5        sdp16R        4      3         3
##         sdp17F        5        sdp17R        4      1         2
##         sdp18F        6        sdp18R        6      4         2
##        hcp1i1F        5       hcp1i1R        6      1         2
##        hcp1i2F       17       hcp1i2R       18      1         2
##          hcp2F        8         hcp2R        9      1         2
##         hcp3hF       15        hcp3hR       16      1         2
##         hcp3mF       18        hcp3mR       19      1         2
##         hcp4hF        5        hcp4hR        4      1         2
##          hcp5F        6         hcp5R        6      4         2
##           pa9F        7          pa9R        8      1         2
##          pa11F        9         pa11R        8      1         1
##          pa13F        5         pa13R        5      2         1
##        pa14i1F        9       pa14i1R        9      2         2
##        pa14i2F        8       pa14i2R        8      2         2
##        pa14i3F       10       pa14i3R       10      2         2
##        pa14i4F        9       pa14i4R        9      2         2
##        pa14i5F        8       pa14i5R        7      1         1
```

```
##          pa15F        9         pa15R        9      2         2
##          cps1F        4         cps1R        4      4         2
##          sat3F       14         sat3R       14      4         2
##          pa26F        6         pa26R        6      4         2
##          pa27F        6         pa27R        6      4         2
##          pa28F        6         pa28R        6      4         2
##        sep1i1F        8       sep1i1R        8      2         1
##        sep1i2F        8       sep1i2R        8      2         1
##          sep2F        7         sep2R        7      2         1
##         sep18F        5        sep18R        5      2         1
##         sep12F        6        sep12R        6      2         1
##         sep13F        4        sep13R        4      2         1
##         sep14F        6        sep14R        6      2         1
##       sep17i1F        7      sep17i1R        6      3         3
##       sep17i2F        7      sep17i2R        6      3         3
##       sep17i3F        7      sep17i3R        6      3         3
##       sep17i4F        7      sep17i4R        6      3         3
##         sep16F        6        sep16R        6      2         3
##          frt2F        8         frt2R        8      4         2
##     frt11v1i6F       10    frt11v1i6R       10      4         2
##     frt11v1i7F        9    frt11v1i7R        9      4         2
##     frt11v1i8F        9    frt11v1i8R        9      4         2
##     frt11v2i6F        6    frt11v2i6R        6      4         2
##     frt11v2i7F        6    frt11v2i7R        6      4         2
##     frt11v2i8F        6    frt11v2i8R        6      4         2
##       crn22p1F       11      crn22p1R       11      2         2
##     crn33p1i1F        7    crn33p1i1R        7      2         2
##     crn33p1i2F        6    crn33p1i2R        5      1         2
##     crn33p1i3F        8    crn33p1i3R        8      2         2
##       crn34p1F        7      crn34p1R        7      2         2
##         igr22F        5        igr22R        4      1        14
##         igr25F        5        igr25R        4      3        13
##         igr28F        5        igr28R        4      1         2
##       igr39p3F       10      igr39p3R        9      1         2
##       igr40p3F        9      igr40p3R        8      1         1
##       igr41p1F       10      igr41p1R        9      5        14
##         rtr43F        5        rtr43R        5      2         1
##         inc24F       38        inc24R       37      1         7
##       inc26i2F        7      inc26i2R        6     11        20
##       inc26i3F        7      inc26i3R        6      9        20
##       inc27i2F        8      inc27i2R        7      1         9
##       inc27i3F        8      inc27i3R        7      1         9
##        int4i6F        4       int4i6R        4      4         2
##         lmhh1F        2        lmhh1R        2      2         4
##         lmhh2F        2        lmhh2R        2      2         3
##         lvhh1F        2        lvhh1R        2      4         4
##         lvhh2F        2        lvhh2R        2      2         3
##         avhh1F        2        avhh1R        2      2         1
##           hv1F        4          hv1R        3      1         2
##       tag_sexF        2      tag_sexR        2      2         1
##      tag_dobpF        2     tag_dobpR        2      4        25
##       sex_genF        2      sex_genR        2      2         1
##      psex_genF        3     psex_genR        4      5         5
##     pdoby_genF       43    pdoby_genR       44     15        15
##     pdobm_genF       14    pdobm_genR       15      9         9
##          pageF       44         pageR       44     16        11
```

```
##            mageF        40            mageR        40     4          3
##           k1ageF        28           k1ageR        28     2          1
##           k2ageF        26           k2ageR        26     4          2
##            pcobF        18            pcobR        19     3          4
##           pnat1F        15           pnat1R        16     3          3
##         marstatF         5         marstatR         5     2          1
##              npF        10              npR        10    20         23
##            ncohF         7            ncohR         7    10         17
##            nmarF         3            nmarR         3     6         12
##         meetdurF       279         meetdurR       278    13         18
##          reldurF       265          reldurR       262    37         94
##        cohabdurF       242        cohabdurR       249   187        134
##          mardurF       191          mardurR       192    13         14
##         homosexF         3         homosexR         3     4          2
##       infertileF         3       infertileR         3     4          4
##        pregnantF         4        pregnantR         4     6          4
##    pnkidsbioalvF         7    pnkidsbioalvR         7    12         64
##          res2ndF         3          res2ndR         2     3       1175
##       hhsizemrdF        13       hhsizemrdR        13    16         17
##            mmrdF         2            mmrdR         2     2          4
##            fmrdF         2            fmrdR         3     3          4
##        childmrdF         9        childmrdR         8     7          8
##          othmrdF         5          othmrdR         5     2          1
##          hhcompF        14          hhcompR        15    21         29
##          penrolF        11          penrolR        11     2          1
##         pschoolF        10         pschoolR        10     6          5
##          pvocatF        11          pvocatR        11     4          2
##          piscedF        11          piscedR        11     6          3
##         pisced2F        11         pisced2R        11     6          3
##         pcasminF        13         pcasminR        13     4          2
##          pyeducF        20          pyeducR        20     6          3
##             egpF       129             egpR        11   264       3514
##            iseiF        73            iseiR        56   451       1720
##           siopsF        55           siopsR        54   431       1331
##        hhincgceeF       413       hhincgceeR       414    21        118
##         pcasprimF        22         pcasprimR        22     4          3
##          pcassecF        17          pcassecR        17     2          1
##             plfsF        14             plfsR        14     4          3
```

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)
```

```r
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] "ehc24p7yR"  "ehc25p7h1R"

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] "ehc16m1R"   "ehc16m2R"   "ehc16m3R"   "ehc4p2m8R"  "ehc4p2m9R"
##  [6] "ehc10k6h2R" "ehc10k9h2R" "ehc24p7mR"  "ehc25p8h2R" "ehc25p8h3R"
## [11] "ehc10k8h2R"

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] "dpidR"  "hcp4mR"

(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] "ehc17m4R"   "ehc22p7nR"  "ehc25p6h2R" "ehc25p6h3R"

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] "ehc10k5h2R" "ehc10k7h2R" "kldbR"

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] "ehc21h1R" "job18mR"  "pa2yR"

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

## [1] 1262


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

```
## [1] "ehc25p4h2F"

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] "dpidF"

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

## [1] 956


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] "ehc16m4F"  "ehc22p7nF" "ehc24p6mF" "ehc24p6yF" "hcp4hF"

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] "avhh1F"    "kldb1992F"

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] "ehc21h1F" "job18mF"  "pa2yF"    "hcp4mF"

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

## [1] 1260
```

B.4. comparison summary for `anchor2_DD.dta` Release 2.0 and Release 3.0

class	Release 2	Release 3
x	8	0
n	15	37
ni	612	612
v	1	7
vn	369	359
vb	2	0
vbi	11	1
vbn	2	1
vbni	954	956
vb1	4	5
vb1i	3	2
vb1n	3	4
vb1ni	1262	1260
sum	3246	3244

Most variables are unchanged (ni, vbni, vb1ni). A small number (x, v) had been dropped or added. Some variables (n) had lost or gained variation or changed to another constant value. A number of variables (vb, vbi, vbn, vb1, vb1i, vb1n) had been (probably) renamed or recoded. A huge group of variables (vn) had been changed.

B.5. selected in-depth comparison

For illustrative purposes we present some detailed results.

```
D1 <- cbind(Raw$pyeducR, Fin$pyeducF)
D1[D1[, 1] != D1[, 2], ]

##      [,1]   [,2]
## [1,] "-3.0" "-7.0"
## [2,] "14.5" "11.5"
## [3,] "-7.0" "-3.0"
```

Both variables share the same number of levels. The level ’-3.0’ has been recoded to ’-7.0’, the level ’14.5’ to ’11.5’ and the level ’-7.0’ to ’-30’. This takes 3 Levenshtein steps (see details in [TR-2012-003]).

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

##       [,1] [,2]
##  [1,] "0"  "1"
##  [2,] "0"  "1"
##  [3,] "0"  "1"
##  [4,] "0"  "1"
##  [5,] "0"  "1"
##  [6,] "1"  "0"
##  [7,] "1"  "0"
##  [8,] "0"  "1"
##  [9,] "0"  "1"
## [10,] "0"  "1"
## [11,] "0"  "1"
## [12,] "1"  "2"
```

Both variables share the same number of levels. $3 + 3 + 6 = 12$, $12/2 = 6$ mappings between the levels are in use. For a bijective mapping we need 3 mappings less. 12 individuals show a changed level. It takes 12 Levenshtein steps to make the levels equal.

```
D3 <- cbind(Raw$pageR, Fin$pageF)
D3[D3[, 1] != D3[, 2], ]

##      [,1] [,2]
## [1,] "-7" "37"
## [2,] "26" "25"
## [3,] "-3" "25"
## [4,] "36" "38"
## [5,] "30" "29"
## [6,] "29" "30"
## [7,] "44" "46"
## [8,] "-7" "-3"
```

8 idividals have a new level for ’page’. It takes 11 Levenshtein steps to get them equal. There is no bijective mapping available.

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

## [1] 1128    2

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

##      [,1] [,2]
## [1,] "23" "22"
## [2,] "32" "33"
## [3,] "53" "31"
## [4,] "38" "56"
## [5,] "48" "50"
## [6,] "51" "48"
```

1128 indiduals do have a different level. There is no bijective mapping.

C. partner1

As usual we have first to load the produced data collection.

```
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\\compareR2R3\\partner1\\Results\\compareFinRaw.RData")
```

C.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)
Fnv2 <- data.frame(Fnv.tab)
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
(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


RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)]

## character(0)

RFnv2[is.na(RFnv2$Fin.no.var), 1]

## character(0)


FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)]

## character(0)

RFnv2[is.na(RFnv2$Raw.no.var), 1]

## character(0)
```

Release 2.0 89 variables without variation are identical in both data sets.

Release 3.0 89 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] 2

sum(!FL.bi)

## [1] 2


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

## [1] 0

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

## [1] 2

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

## [1] "pdobm" "pdoby"


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

## [1] 0

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

## [1] 2

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

## [1] "pdobm" "pdoby"
```

Release 2.0 2 variables share the name with one of the variables in Release 3.0 at least.

Release 3.0 2 variables share the name with one of the variables in Release 2.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping.

```
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] 2

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
##    pdobmF       13   pdobmR       13      8        12
##    pdobyF       38   pdobyR       38     14        36
```

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"])))

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

## [1] "idR"

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"


(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] "psd10R"

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

## [1] 137


(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] "idF"

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"



(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] "psd10F"

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

## [1] 137
```

C.4. comparison summary for `partner1_DD.dta` Release 2.0 and Release 3.0

class	Release 2	Release 3
x	0	0
n	0	0
ni	89	89
v	0	0
vn	2	2
vb	0	0
vbi	0	0
vbn	1	1
vbni	1	1
vb1	0	0
vb1i	0	0
vb1n	1	1
vb1ni	137	137
sum	231	231

Most variables are unchanged (ni, vbni, vb1ni). A small number of variables (vbn, vb1n) had been (probably) renamed or recoded. A small group of variables (vn) had been changed.

D. partner2

As usual we have first to load the produced data collection.

```
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\\compareR2R3\\partner2\\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)
Fnv2 <- data.frame(Fnv.tab)
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
(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


RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)]

## character(0)

RFnv2[is.na(RFnv2$Fin.no.var), 1]

## character(0)


FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)]

## character(0)

RFnv2[is.na(RFnv2$Raw.no.var), 1]

## character(0)
```

Release 2.0 96 variables without variation are identical in both data sets.

Release 3.0 96 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] 2

sum(!FL.bi)

## [1] 2


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

## [1] 0

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

## [1] 2

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

## [1] "pdobm" "pdoby"


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

## [1] 0

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

## [1] 2

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

## [1] "pdobm" "pdoby"
```

Release 2.0 2 variables share the name with one of the variables in Release 3.0 at least.

Release 3.0 2 variables share the name with one of the variables in Release 2.0 at least.

We compare all variables pairs (with variation) which share the same name but are not connected with a bijective mapping.

```
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] 2

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
##    pdobmF       13   pdobmR       13      4         3
##    pdobyF       38   pdobyR       38      4         2
```

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] "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] 180


(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] 180
```

D.4. comparison summary for `partner2_DD.dta` Release 2.0 and Release 3.0

class	Release 2	Release 3
x	0	0
n	0	0
ni	96	96
v	0	0
vn	2	2
vb	0	0
vbi	0	0
vbn	0	0
vbni	2	2
vb1	0	0
vb1i	0	0
vb1n	0	0
vb1ni	180	180
sum	280	280

Most variables are unchanged (ni, vbni, vb1ni). A small group of variables (vn) had been changed.

E. weights

As usual we have first to load the produced data collection.

```
rm(list = ls())
duplicStrict <- function(A) {
    return(duplicated(A) | duplicated(A, fromLast = TRUE))
}
load("..\\compareR2R3\\weights\\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
```

```r
Rnv2 <- data.frame(Rnv.tab)
Fnv2 <- data.frame(Fnv.tab)
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
(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


RawNames[Rnv][!(RawNames[Rnv] %in% FinNames)]

## character(0)

RFnv2[is.na(RFnv2$Fin.no.var), 1]

## character(0)


FinNames[Fnv][!(FinNames[Fnv] %in% RawNames)]

## character(0)

RFnv2[is.na(RFnv2$Raw.no.var), 1]

## character(0)
```

Release 2.0 No change for variables without variation.

Release 3.0 No change for variables without variation.

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] 2


(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] 2

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

## [1] "lweightDD4" "lweightPF4"

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

## [1] 0
```

Release 2.0 No changed variables with variation but without any bijective mapping.

Release 3.0 For 2 variables with variation but without any bijective mapping exist no variables with the same name in the Release 2.0 data.

See the comparison table as an illustration.

```
results$same.name <-
  gsub("R$","",as.character(results$Raw.c.nm)) == gsub("F$","",as.character(results$Fin.c.nm))
print(results[(results$same.name),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
##           idF    13891          idR    13891      0         0
##      pairfamF        2     pairfamR        2      0         0
##     caweightF      535    caweightR      535      0         0
```

```
##          westF        2          westR        2       0          0
##     ca1weightF     1470     ca1weightR     1470       0          0
##      ddweightF        3      ddweightR        3       0          0
##    ddcaweightF      581    ddcaweightR      581       0          0
##      d1weightF        6      d1weightR        6       0          0
##   d1ca1weightF     2380   d1ca1weightR     2380       0          0
##    lweightDD3F        3    lweightDD3R        3       0          0
##    lweightPF2F        7    lweightPF2R        7       0          0
##    lweightPF3F        7    lweightPF3R        7       0          0
```

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"])))

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

##  [1] "idR"          "pairfamR"     "caweightR"    "westR"
##  [5] "ca1weightR"   "ddweightR"    "ddcaweightR"  "d1weightR"
##  [9] "d1ca1weightR" "lweightDD3R"  "lweightPF2R"  "lweightPF3R"


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

##  [1] "idF"          "pairfamF"     "caweightF"    "westF"
##  [5] "ca1weightF"   "ddweightF"    "ddcaweightF"  "d1weightF"
##  [9] "d1ca1weightF" "lweightDD3F"  "lweightPF2F"  "lweightPF3F"
```

E.4. comparison summary for `weights_DD.dta` Release 2.0 and Release 3.0

class	Release 2	Release 3
x	0	0
n	0	0
ni	0	0
v	0	2
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	12	12
sum	12	14

All old variables are unchanged (vb1ni). A small number (v) had been added.

Literatur

[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

[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.

[R 2.15] R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.