MIRROR data manipulation

Thijs Noordzij

5 mei 2016

# Open MIRROR-file from SPSS .sav file extension

## Require haven package

Haven package is faster than foreign package, and without error message. Variable labels are stored in the "label" attribute of each variable.

if (require("haven")){  
 print("haven is loaded correctly")  
} else {  
 print("trying to install haven")  
 install.packages("haven")  
 if (require("haven")){  
 print("haven installed and loaded")  
 } else {  
 stop("could not install haven")  
 }  
}

## Load data

It takes almost 1 minute to load the data.

data <- read\_spss("C:/Users/Thijs/Documents/MIRROR/data/mirror\_formatie\_ tm2014.sav")

The output of system.time() of previous code section:

|user|system|elapsed| |-|-|-| |40.72|0.33|41.08|

# Recoding missing values correctly

## Define NA in FUNGRP(\_PVE1), BRIN, GEBDAT, GESLACHT ### FUNGRP NA defined as in SPSS file for MIRROR provided by DUO. FUNGRP also has 2431 system.missings in SPSS.

missing\_FUNGRP <- c(-1)  
data$FUNGRP <- lapply(data$FUNGRP, function(x) replace(x, x %in% missing\_FUNGRP, NA))  
data$FUNGRP <- as.numeric(data$FUNGRP)

The output of system.time() of previous code section:

|  |  |  |
| --- | --- | --- |
| user | system | elapsed |
| 29.65 | 0.13 | 29.92 |

### FUNGRP\_PVE

NA defined as in SPSS file for MIRROR provided by DUO. FUNGRP also has 2869 system.missings in SPSS.

missing\_FUNGRP\_PVE1 <- c(-1)  
data$FUNGRP\_PVE1 <- lapply(data$FUNGRP\_PVE1, function(x) replace(x, x %in% missing\_FUNGRP\_PVE1, NA))  
data$FUNGRP\_PVE1 <- as.numeric(data$FUNGRP\_PVE1)

### BRIN

NA defined as in SPSS file for MIRROR provided by DUO.

missing\_BRIN <- c("O")   
data$BRIN <- lapply(data$BRIN, function(x) replace(x, x %in% missing\_BRIN, NA))

The output of system.time() of previous code section:

|  |  |  |
| --- | --- | --- |
| user | system | elapsed |
| 30.81 | 0.02 | 30.83 |

There are many more unlikely, such as "O", "VMB", "0000", "P&O", "PR", "SB O", "STAF", "TEAM". In a [next section](#ValidateBRIN) this will be handled.

### GEBDAT

NA defined as in SPSS file for MIRROR provided by DUO.

missing\_GEBDAT <- c(-1)   
data$GEBDAT <- lapply(data$GEBDAT, function(x) replace(x, x %in% missing\_GEBDAT, NA))

The output of system.time() of previous code section:

|  |  |  |
| --- | --- | --- |
| user | system | elapsed |
| 54.51 | 0.11 | 54.81 |

### GESLACHT

NA defined as in SPSS file for MIRROR provided by DUO, "" (empty string) added.

missing\_GESLACHT <- c("", "O")   
data$GESLACHT <- lapply(data$GESLACHT, function(x) replace(x, x %in% missing\_GESLACHT, NA))

The output of system.time() of previous code section:

|  |  |  |
| --- | --- | --- |
| user | system | elapsed |
| 30.50 | 0.06 | 30.60 |

## Total missing values per variable

sum(is.na(data$BRIN)) # 1393  
sum(is.na(data$GEBDAT)) # 4156  
sum(is.na(data$GESLACHT)) # 4059 (if " " is added as missing value, else 0)  
sum(is.na(data$FUNGRP)) # 10559, that includes 2431 system.missing in SPSS  
sum(is.na(data$FUNGRP\_PVE1)) # 9067, that includes 2869 system.missing in SPSS

|  |  |  |
| --- | --- | --- |
| variable | missing values | comments |
| BRIN | 1393 | - |
| GEBDAT | 4156 | - |
| GESLACHT | 4059 | if " " is added as missing value, else 0 |
| FUNGRP | 10559 | that includes 2431 system.missing in SPSS |
| FUNGRP\_PVE1 | 9067 | that includes 2869 system.missing in SPSS |

## Validate BRIN

BRIN is a unique identifier for a school. BRIN has a standard format: 4 characters, number, number, character, character (e.g. "11AO"). Some BRIN in the MIRROR-file do not represent the school where the staff (teacher or otherwise) works. These 'BRIN' represent staff working under direct responsibility of the board. These BRIN have a free format, which deviates from the previously mentioned standard format. Possibly, these BRIN are also not identical for the same school in different years.

valid.brin is a function that identifies (in)correct values for BRIN. This function is used to make a new variable in the data of the MIRROR-file (data$validbrin). valid.brin is sourced from a separate file.

source('E:/Google Drive/Promotie/Analyse/Teacher-attrition-with-MIRROR-data/validBRIN.R')  
data$validbrin <- lapply(data$BRIN, valid.brin) # the funtion works, but it takes ca. 35 minutes to complete the operation.   
data$validbrin <- as.logical(data$validbrin)

The validBRIN funtion works, but it takes ca. 35 minutes to complete the operation.

length(data$validbrin[data$validbrin==FALSE])

43636 invalid BRIN, which includes the 1393 missing values.

-# To do: check if/how incorrect BRIN are related to other variables data\_invalidbrin <- subset(data, validbrin==FALSE) table(data$validbrin, exclude = NULL) # FALSE TRUE # 43636 6407015 0

BRIN\_FUNGRP\_table <- table(dataFUNGRP\_PVE1, exclude = NULL) BRIN\_FUNGRP\_prop.table <- prop.table(table(dataFUNGRP\_PVE1, exclude = NULL), 1) write.csv2(BRIN\_FUNGRP\_table, file = "C:/Users/Thijs/Documents/MIRROR/Output/BRIN\_FUNGRP\_table.csv") write.csv2(BRIN\_FUNGRP\_prop.table, file = "C:/Users/Thijs/Documents/MIRROR/Output/BRIN\_FUNGRP\_prop.table.csv") barplot(BRIN\_FUNGRP\_prop.table, main = "Proportion of valid BRIN per job", xlab = "Job", legend = c("invalid BRIN", "valid BRIN"), col=c("red","darkgreen"), beside = TRUE)

head(data$FUNGRP\_PVE1)

mytable <- table(data\_invalidbrinFUNGRP\_PVE1, exclude=NULL) mytable

mytable <- table(dataFUNGRP\_PVE1, exclude=NULL) mytable prop.table(mytable, 1)

class(data\_invalidbrin$FUNGRP)

sum(is.na(data\_invalidbrinFUNGRP)

complete.cases(data\_invalidbrin$FUNGRP) # NA's a problem?

getwd() tmp <- tempfile(fileext = ".sav") write.csv(data\_invalidbrin, "C:/Users/Thijs/Documents/MIRROR/data/invalid\_brin.csv")

head(data\_invalidbrin)

# install.packages("ggplot2")

library(ggplot2) invalidbrin\_FUNGRP <- ggplot(data\_invalidbrin, aes(FUNGRP)) invalidbrin\_FUNGRP + geom\_histogram(stat = "count", binwidth = 1, bins = NULL)

# datavalidbrin==FALSE]

# count(data[,'JAAR'][!y])

# count(data[!y,], vars = c('BESTUUR1'))

# sum(datavalidbrin==FALSE][data$FUNGRP==1])

# dataFUNGRP==1][data$validbrin==FALSE]

compare\_FUNGRP\_FUNGRP\_PVE1 <- table(dataFUNGRP\_PVE1, exclude = NULL) write.csv2(compare\_FUNGRP\_FUNGRP\_PVE1, file = "C:/Users/Thijs/Documents/MIRROR/Output/compare\_FUNGRP\_FUNGRP\_PVE1.csv") # Value labels are still missing.

# Duplicate records

library(plyr)

data1 <- data[, c("id\_2015", "JAAR", "BRIN", "BESTUUR1", "GEBDAT", "GESLACHT", "FUNGRP")] data1$koppelprobleem <- apply(data1, 1, anyNA) # user system elapsed # 20.10 0.26 20.38 sum(data1$koppelprobleem) # 18450 records have one or more NA's in variables used for linking to other data.

# check for persons (id\_2015) with one or more recors with one or more NA's in variables used for linking to other data.

id\_data <- ddply(data1, "id\_2015", summarise, koppelprobleem\_record = max(koppelprobleem)) sum(id\_data$koppelprobleem\_record) # 16868 out of 629793 persons (2.68%)

# idem for most recent year (2014)

id\_data\_2014 <- ddply(data1[data1$JAAR==2014,], "id\_2015", summarise, koppelprobleem\_record = max(koppelprobleem)) sum(id\_data\_2014$koppelprobleem\_record) # 505 out of 332418 persons (0,15%)

# idem for most recent 6 years (2008 - 2014)

id\_data\_20092014 <- ddply(subset(data1, JAAR >= 2008), "id\_2015", summarise, koppelprobleem\_record = max(koppelprobleem)) sum(id\_data\_20092014$koppelprobleem\_record) # 16759 out of 478643 persons (3,50%)

head(data1[which(data1$JAAR==(2009:2014)), ]) head(data1[data1$JAAR==2014, ]) head(subset(data1, JAAR >= 2008))

min(c(T,T,F))

duplicated(testdata)

# Create variable for entry and exit into the labor force.

testdata2 <- ddply(testdata, "id\_2015", summarise, first\_year = min(JAAR)) # summarise function aggregates data frame by "id\_2015" testdata3 <- ddply(testdata, "id\_2015", summarise, last\_year = max(JAAR)) # user system elapsed # 2.52 0.03 2.55

testdata4 <- merge(testdata2, testdata3) testdata5 <- merge(testdata, testdata4)

head(testdata2) head(testdata3) head(testdata4) head(testdata5)

testdata6 <- ddply(testdata, c("id\_2015", "FUNGRP"), summarise, last\_year = max(JAAR)) # FUNGRP is used for MIRROR, a model for labour market estimates. Most likely the best variable to use to determine if someone is a teacher. head(testdata6)

# To do:

# 1. How to get the value labels out of the SPSS-file?

# The not-sophisticated way: <http://www.statmethods.net/input/valuelabels.html>

dataFUNGRP\_PVE1, levels = c(-1,1,2,3,4,5), labels = c("onbekend", "directie, management", "onderwijzend personeel, leraren", "onderwijsondersteunend personeel", "beheer- en administratief personeel", "leraren in opleiding")) # Dit geeft problemen bij het benoemen van NA's in eerdere syntax.

# Or, alternatively:

# dataFUNGRP\_PVE1,

# levels = c(1,2,3,4,5),

# labels = c("directie/management", "OP", "OOP", "OBP", "LIO"))

# 2. Convert some variables to nominal variables (factor?)

# 3. Create date variable out of GEBDAT

# 4. Fill missing GEBDAT is known within same id\_2015? Maybe not: id\_2015 might not be very reliable if date of birth is missing.

{r setup, include=FALSE} knitr::opts\_chunk$set(echo = TRUE)

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

{r cars} summary(cars)

## Including Plots

You can also embed plots, for example:

{r pressure, echo=FALSE} plot(pressure)

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.