The Dataset Project: Handling survey data in R

Emmanuel Rousseaux and Gilbert Ritschard

NCCR LIVES – IP 14

Institute for Demographic and Life Course Studies

University of Geneva
1211 Geneva 4, Switzerland

emmanuel.rousseaux@unige.ch

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

LIVES
Swiss National Centre of Competence in Research

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Motivation

- Population studies strongly rely on survey data
- Survey data management has specific needs
- Currently R does not offer a robust framework to handle survey data
- Much time is needed to manage and prepare data

  especially: create partner/sibling/parent files, deal with doublons
- Especially for panel survey data and network survey data

⇒ Need for a specific software framework in R

Overview

- Storing, documenting and sharing complex survey data in R (cross-sectional data, panel data, network data)
- Merging data and metadata describing the survey
- Helping at efficiently and securely prepare data for a study
- Helping at quickly focus on results when running into analysis
- Facilitating reproducible research

Overview

- Storing, documenting and sharing complex survey data in R (cross-sectional data, panel data, network data)
- Merging data and metadata describing the survey
- Helping at efficiently and securely prepare data for a study
- Helping at quickly focus on results when running into analysis
- Facilitating reproducible research

Overview

- Storing, documenting and sharing complex survey data in R (cross-sectional data, panel data, network data)
- Merging data and metadata describing the survey
- Helping at efficiently and securely prepare data for a study
- Helping at quickly focus on results when running into analysis
- Facilitating reproducible research

Overview

- Storing, documenting and sharing complex survey data in R (cross-sectional data, panel data, network data)
- Merging data and metadata describing the survey
- Helping at efficiently and securely prepare data for a study
- Helping at quickly focus on results when running into analysis
- Facilitating reproducible research

Overview

- Storing, documenting and sharing complex survey data in R (cross-sectional data, panel data, network data)
- Merging data and metadata describing the survey
- Helping at efficiently and securely prepare data for a study
- Helping at quickly focus on results when running into analysis
- Facilitating reproducible research

Overview

- Storing, documenting and sharing complex survey data in R (cross-sectional data, panel data, network data)
- Merging data and metadata describing the survey
- Helping at efficiently and securely prepare data for a study
- Helping at quickly focus on results when running into analysis
- Facilitating reproducible research

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

LIVES
Swiss National Centre of Competence in Research

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

Key functionalities: storage

- Allows to store metadata about the survey conducted
- Accepts user-defined missing values
- Natively accounts for weights
- Generates codebooks directly in PDF format
- Automatic data consistency checks
- Automatic "loss of representativeness" checks

Key functionalities: preparing data

- Search for specific variables across the whole database
- Specify the measure (scale, nominal, ordinal, . . .)
- Turn a missing value to valid case and vice-versa
- Easy to use/remember recoding methods
- Detailed frequency tables

Key functionalities: preparing data

- Search for specific variables across the whole database
- Specify the measure (scale, nominal, ordinal, . . .)
- Turn a missing value to valid case and vice-versa
- Easy to use/remember recoding methods
- Detailed frequency tables

Key functionalities: preparing data

- Search for specific variables across the whole database
- Specify the measure (scale, nominal, ordinal, . . .)
- Turn a missing value to valid case and vice-versa
- Easy to use/remember recoding methods
- Detailed frequency tables

Key functionalities: preparing data

- Search for specific variables across the whole database
- Specify the measure (scale, nominal, ordinal, . . .)
- Turn a missing value to valid case and vice-versa
- Easy to use/remember recoding methods
- Detailed frequency tables

Key functionalities: preparing data

- Search for specific variables across the whole database
- Specify the measure (scale, nominal, ordinal, . . .)
- Turn a missing value to valid case and vice-versa
- Easy to use/remember recoding methods
- Detailed frequency tables

Key functionalities: preparing data

- Search for specific variables across the whole database
- Specify the measure (scale, nominal, ordinal, . . .)
- Turn a missing value to valid case and vice-versa
- Easy to use/remember recoding methods
- Detailed frequency tables

Key functionalities: statistical analysis

- Programming syntax oriented for scientists in social sciences
- Automatically verify validity of models computed
- Format outputs to quickly focus on interpretation, in a PDF file
- Print in this file all settings used
- Export outputs for reuse in other softwares

Key functionalities: statistical analysis

- Programming syntax oriented for scientists in social sciences
- Automatically verify validity of models computed
- Format outputs to quickly focus on interpretation, in a PDF file
- Print in this file all settings used
- Export outputs for reuse in other softwares

Key functionalities: statistical analysis

- Programming syntax oriented for scientists in social sciences
- Automatically verify validity of models computed
- Format outputs to quickly focus on interpretation, in a PDF file
- Print in this file all settings used
- Export outputs for reuse in other softwares

Key functionalities: statistical analysis

- Programming syntax oriented for scientists in social sciences
- Automatically verify validity of models computed
- Format outputs to quickly focus on interpretation, in a PDF file
- Print in this file all settings used
- Export outputs for reuse in other softwares

Key functionalities: statistical analysis

- Programming syntax oriented for scientists in social sciences
- Automatically verify validity of models computed
- Format outputs to quickly focus on interpretation, in a PDF file
- Print in this file all settings used
- Export outputs for reuse in other softwares

Key functionalities: statistical analysis

- Programming syntax oriented for scientists in social sciences
- Automatically verify validity of models computed
- Format outputs to quickly focus on interpretation, in a PDF file
- Print in this file all settings used
- Export outputs for reuse in other softwares

Tools for panel data

- Automatically check for missings values/valids cases across years
- Extract a whole trajectory in one step
- Switch missing/valid values across years in one step
- Perform recoding operation across years in one step
- Export to sequence objects ready to be analysed with the `TraMineR` toolbox (Gabadinho et al., 2011)

Tools for panel data

- Automatically check for missings values/valids cases across years
- Extract a whole trajectory in one step
- Switch missing/valid values across years in one step
- Perform recoding operation across years in one step
- Export to sequence objects ready to be analysed with the `TraMineR` toolbox (Gabadinho et al., 2011)

Tools for panel data

- Automatically check for missings values/valids cases across years
- Extract a whole trajectory in one step
- Switch missing/valid values across years in one step
- Perform recoding operation across years in one step
- Export to sequence objects ready to be analysed with the `TraMineR` toolbox (Gabadinho et al., 2011)

Tools for panel data

- Automatically check for missings values/valids cases across years
- Extract a whole trajectory in one step
- Switch missing/valid values across years in one step
- Perform recoding operation across years in one step
- Export to sequence objects ready to be analysed with the `TraMineR` toolbox (Gabadinho et al., 2011)

Tools for panel data

- Automatically check for missings values/valids cases across years
- Extract a whole trajectory in one step
- Switch missing/valid values across years in one step
- Perform recoding operation across years in one step
- Export to sequence objects ready to be analysed with the `TraMineR` toolbox (Gabadinho et al., 2011)

Tools for panel data

- Automatically check for missings values/valids cases across years
- Extract a whole trajectory in one step
- Switch missing/valid values across years in one step
- Perform recoding operation across years in one step
- Export to sequence objects ready to be analysed with the `TraMineR` toolbox (Gabadinho et al., 2011)

A short demonstration

Importing an SPSS file

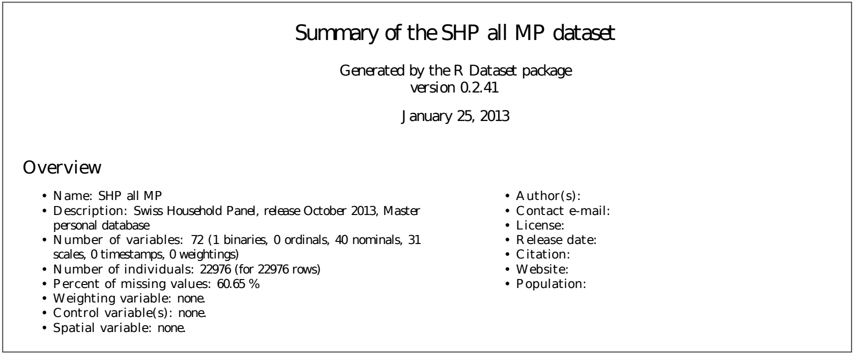
Here we use data from the Swiss Household Panel (Voorpostel et al., 2012)

```
shp.all <- get.spss.file(
  file = "SHP_MP.sav",
  datadir = datadir.all,
  name = "SHP all MP",
  description = "Swiss Household Panel, release October 2012, Master
    personal database"
)
```

Getting a codebook of the database

```
exportPDF(shp.all)
```

Getting a codebook of the database



Summary of the SHP all MP dataset

Generated by the R Dataset package
version 0.2.41

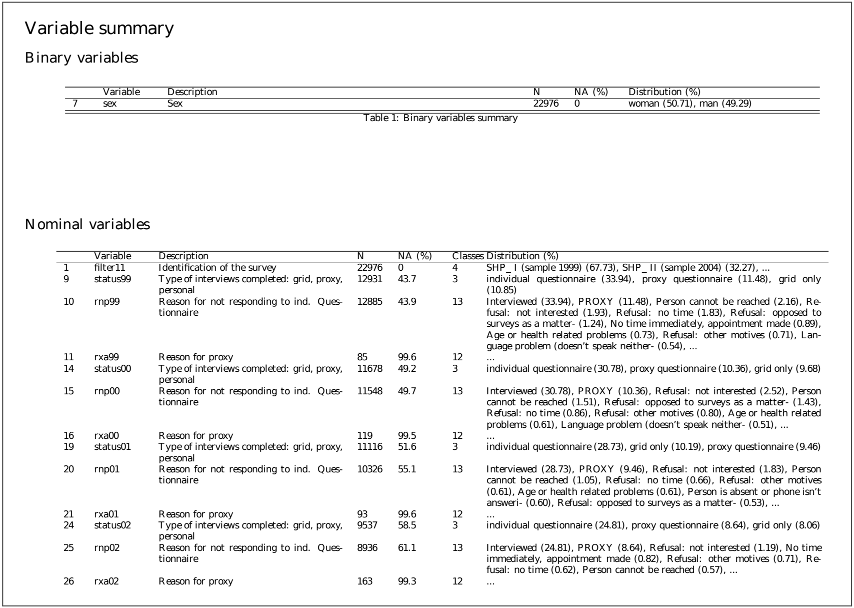
January 25, 2013

Overview

- Name: SHP all MP
- Description: Swiss Household Panel, release October 2013, Master personal database
- Number of variables: 72 (1 binaries, 0 ordinals, 40 nominals, 31 scales, 0 timestamps, 0 weightings)
- Number of individuals: 22976 (for 22976 rows)
- Percent of missing values: 60.65 %
- Weighting variable: none.
- Control variable(s): none.
- Spatial variable: none.

- Author(s):
- Contact e-mail:
- License:
- Release date:
- Citation:
- Website:
- Population:

Figure: *Example of codebook generation, page 1.*



Variable summary

Binary variables

	Variable	Description	N	NA (%)	Distribution (%)
7	sex	Sex	22976	0	woman (50.71), man (49.29)

Table 1: Binary variables summary

Nominal variables

	Variable	Description	N	NA (%)	Classes	Distribution (%)
1	filter11	Identification of the survey	22976	0	4	SHP_I (sample 1999) (67.73), SHP_II (sample 2004) (32.27), ...
9	status99	Type of interviews completed: grid, proxy, personal	12931	43.7	3	individual questionnaire (33.94), proxy questionnaire (11.48), grid only (10.85)
10	rnp99	Reason for not responding to ind. Questionnaire	12885	43.9	13	Interviewed (33.94), PROXY (11.48), Person cannot be reached (2.16), Refusal: not interested (1.93), Refusal: no time (1.83), Refusal: opposed to surveys as a matter- (1.24), No time immediately, appointment made (0.89), Age or health related problems (0.73), Refusal: other motives (0.71), Language problem (doesn't speak neither- (0.54), ...
11	rxa99	Reason for proxy	85	99.6	12	...
14	status00	Type of interviews completed: grid, proxy, personal	11678	49.2	3	individual questionnaire (30.78), proxy questionnaire (10.36), grid only (9.68)
15	rnp00	Reason for not responding to ind. Questionnaire	11548	49.7	13	Interviewed (30.78), PROXY (10.36), Refusal: not interested (2.52), Person cannot be reached (1.51), Refusal: opposed to surveys as a matter- (1.43), Refusal: no time (0.86), Refusal: other motives (0.80), Age or health related problems (0.61), Language problem (doesn't speak neither- (0.51), ...
16	rxa00	Reason for proxy	119	99.5	12	...
19	status01	Type of interviews completed: grid, proxy, personal	11116	51.6	3	individual questionnaire (28.73), grid only (10.19), proxy questionnaire (9.46)
20	rnp01	Reason for not responding to ind. Questionnaire	10326	55.1	13	Interviewed (28.73), PROXY (9.46), Refusal: not interested (1.83), Person cannot be reached (1.05), Refusal: no time (0.66), Refusal: other motives (0.61), Age or health related problems (0.61), Person is absent or phone isn't answeri- (0.60), Refusal: opposed to surveys as a matter- (0.53), ...
21	rxa01	Reason for proxy	93	99.6	12	...
24	status02	Type of interviews completed: grid, proxy, personal	9537	58.5	3	individual questionnaire (24.81), proxy questionnaire (8.64), grid only (8.06)
25	rnp02	Reason for not responding to ind. Questionnaire	8936	61.1	13	Interviewed (24.81), PROXY (8.64), Refusal: not interested (1.19), No time immediately, appointment made (0.82), Refusal: other motives (0.71), Refusal: no time (0.62), Person cannot be reached (0.57), ...
26	rxa02	Reason for proxy	163	99.3	12	...

Figure: *Example of codebook generation, page 4.*

Preparing data for analysis

We load Personnal database of the 2011 wave

```
shp.w2011p <- get.spss.file(
  file = "SHP11_P_USER.sav",
  datadir = datadir.w2011,
  name = "SHP wave 2011 personal",
  description = "Swiss Household Panel, release October 2012,
    wave 2011, personal database"
)
```

Then we merge both databases

```
shp <- merge(shp.all, shp.w2011p, by = "idpers")
```

First we correctly weight data

How many variables in our database?

```
nvariable(shp)
## NULL
## [1] 531
```

But we can easily retrieve them

```
weights.var <- contains("weight", shp)
##                                                                                    Description
## p11c46                                                                            Weight in kg
## wp11t1p           PSMI-PSMII transversal individual weight inflating to size of CH-population
## wp11t1s                          PSMI-PSMII transversal individual weight keeping sample size
## wp11lp1p       PSMI longitudinal individual weight inflating to size of CH-population in 1999
## wp11lp1s                              PSMI longitudinal individual weight keeping sample size
## wp11l1p  PSMI-PSMII longitudinal individual weight inflating to size of CH-population in 2004
## wp11l1s                         PSMI-PSMII longitudinal individual weight keeping sample size
```

We use the variable `wp11t1s`

We check the variable is valid for weighting data

```
shp$wp11t1s <- wvar(shp$wp11t1s)
```

Then we set weights in the database

```
weighting(shp) <- "wp11t1s"
```

And compare the number of individuals to the number of rows

```
nrow(shp)
## [1] 11178
```

```
nindividual(shp)
## [1] 7459
```

Retrieving variables of interest: health

```
health.var <- contains("health", shp)
```

```
##                                                         Description
## p11c01                                               Health status
## p11c02                             Satisfaction with health status
## p11c03                       Improvement in health: Last 12 months
## p11c04a               Health problems: Back problems: Last 4 weeks
## p11c05a         Health problems: Weakness, weariness: Last 4 weeks
## p11c06a           Health problems: Sleeping problems: Last 4 weeks
## p11c07a                   Health problems: Headaches: Last 4 weeks
## p11c08         Health impediment in everyday activities: Extension
## p11c19a                Chronic illness or long-term health problem
## p11c11  Number of days affected by health problems: Last 12 months
## p11p54                                      Public expenses: Health
## x11c05                                  Assessment of health status
## x11c06                               Suffering from health problems
## x11c07                                     Cause of health problems
## x11c09                 Days of suffering from health problems: Days
```

Retrieving variables of interest: association membership

```
association.var <- contains("association", shp)
##                                                              Description
## p11n40                     Associational membership: Sports or leisure
## p11n41                               Associational membership: Culture
## p11n42                             Associational membership: Syndicate
## p11n43                       Associational membership: Political Party
## p11n44         Associational membership: Protection of the environment
## p11n45               Associational membership: Charitable organisation
## p11n50       Associational membership: Religious organisation or group
## p11n51 Associational membership: Local, parents' or women's association
## p11n52                 Associational membership: Other interest groups
```

Retrieving variables of interest: working status

```
work.var <- contains(c("work", "status"), shp, and = TRUE)
##            Description
## wstat11 Working status
```

Then we extract our study sample

```
study.variables <- c(
  "wp11t1s",
  "p11c01",
  "age11",
  "sex11",
  "canton11",
  "p11n40",
  "wstat11"
)
```

```
study <- shp[, study.variables]
```

Quick overview of the variables in our database

```
alldescriptions(study)
##                                                           Description
## wp11t1s PSMI-PSMII transversal individual weight keeping sample size
## sex11                                                             Sex
## age11                                       Age in year of interview
## p11c01                                                 Health status
## wstat11                                               Working status
## p11n40                   Associational membership: Sports or leisure
```

We can rename variables to be more clear

```
study <- rename(study,
"wp11t1s" = "weights",
"p11c01" = "health",
"age11" = "age",
"sex11" = "sex",
"canton11" = "canton",
"p11n40" = "association",
"wstat11" = "work.stat"
)
```

```
alldescriptions(study)
##                                                               Description
## weights     PSMI-PSMII transversal individual weight keeping sample size
## sex                                                                   Sex
## age                                              Age in year of interview
## health                                                      Health status
## work.stat                                                  Working status
## association                   Associational membership: Sports or leisure
```

With the same function we also can rename values

```
study$health <- rename(study$health,
  "so, so (average)" = "so, so",
  "not well at all" = "poor"
)
```

```
valids(study$health)
##     very well          well        so, so not very well          poor
##             1             2             3             4             5
```

Subsampling population: lost of representativness check

We define variables on which we want to perform checks

```
checkvars(study) <- c("sex", "work.stat")
```

And we subsample our study database

```
shp.association <- subset(study, association == "Active member")
## => control on sex:  warning, p-value < 0.05
## man are overrepresented
## woman are underrepresented
## => control on work.stat:  warning, p-value < 0.05
## active occupied are overrepresented
## unemployed, not in labor force are underrepresented
```

Computing frequencies: for categorical variables

```
frequencies("health", study)
##    Coding Missing         Label    N N total Percent Percent (all) Percent total
## 1       1             very well 1428           19.16         19.15
## 2       2                  well 4811           64.52         64.50
## 3       3                so, so 1037           13.92         13.91
## 4       4         not very well  157            2.11          2.11
## 5       5                  poor   21    7456    0.29          0.29         99.97
## 7      -2       x     no answer    2          100.00          0.03
## 6      -1       x does not know    0            0.00          0.00
## 8      -3       x  inapplicable    0            0.00          0.00
## 9      -7       x  filter error    0            0.00          0.00
## 10     -8       x   other error    0       2    0.00          0.00          0.03
## 11                                      7459                                 100
```

Computing frequencies: for scale variables

```
frequencies("age", study)
##    Coding Missing         Label    N N total Percent Percent (all) Percent total
## 1       1               [0,9.7]    0            0.00          0.00
## 2       2            (9.7,19.4]  592            7.95          7.95
## 3       3           (19.4,29.1] 1066           14.30         14.30
## 4       4           (29.1,38.8]  990           13.27         13.27
## 5       5           (38.8,48.5] 1477           19.81         19.81
## 6       6           (48.5,58.2] 1245           16.70         16.70
## 7       7           (58.2,67.9]  926           12.42         12.42
## 8       8           (67.9,77.6]  716            9.60          9.60
## 9       9           (77.6,87.3]  394            5.29          5.29
## 10     10             (87.3,97]   48    7459    0.65          0.65        100.00
## 11     -1       x does not know    0            0.00          0.00
## 12     -2       x     no answer    0            0.00          0.00
## 13     -3       x  inapplicable    0            0.00          0.00
## 14     -7       x  filter error    0            0.00          0.00
## 15     -8       x   other error    0       0    0.00          0.00          0.00
## 16                                      7459                                 100
```

Exporting frequency tables

Export in a PDF file or in a LaTeX document/presentation

```
exportTEX(frequencies("health", study))
```

Coding	Missing	Label	N	N total	Percent	Percent (all)	Percent total
1		very well	1428		19.16	19.15	
2		well	4811		64.52	64.50	
3		so, so	1037		13.92	13.91	
4		not very well	157		2.11	2.11	
5		poor	21	7456	0.29	0.29	99.97
-2	x	no answer	2		100.00	0.03	
-1	x	does not know	0		0.00	0.00	
-3	x	inapplicable	0		0.00	0.00	
-7	x	filter error	0		0.00	0.00	
-8	x	other error	0	2	0.00	0.00	0.03
				7459			100

Recoding: categorical variables

Merging values

```
study$health.2 <- recode(
  study$health,
  "well" = 1:2,
  "poor" = 3:5
)
## number of missings:  3587 ( 32.09 %)
## Operation completed successfully.
## Here is the allocation of the rows in the different classes.

##
##                 well poor
##   very well     1500    0
##   well          4926    0
##   so, so           0 1015
##   not very well    0  136
##   poor             0   14
```

Recoding: scale variables

Discretization

```
study$age.3 <- cut(
  study$age,
  breaks = c(30,65)
)
```

Coding	Missing	Label	N	N total	Percent	Percent (all)	Percent total
1		[0,30]	1775		23.80	23.80	
2		(30,65]	4325		57.99	57.99	
3		(65,97]	1358	7459	18.21	18.21	100.00
-1	x	does not know	0		0.00	0.00	
-2	x	no answer	0		0.00	0.00	
-3	x	inapplicable	0		0.00	0.00	
-7	x	filter error	0		0.00	0.00	
-8	x	other error	0	0	0.00	0.00	0.00
				7459			100

Running into analysis: univariate

The package extends classical univariate descriptive statistic methods for taking weights into account.

Methods provided are: `min`, `max`, `mode`, `mean`, `standard deviation` and `variance`.

Running into analysis: bivariate

```
bivan(
health.2 ~ sex + age.3 + association + work.stat,
study
)
```

	chi2	cramer.v	gk.tau.sqrt	somer.d
sex	23.83 ***	0.06 ***	0.06 ***	0.04 ***
age.3	273.95 ***	0.19 ***	0.19 ***	0.13 ***
association	85.84 ***	0.11 ***	0.11 ***	0.08 ***
work.stat	232.88 ***	0.18 ***	0.18 ***	0.14 ***

Table: *Bivariate analysis with the self-reported health as dependend variable. Legend: *** $<$ 0.001, ** $<$ 0.01, * $<$ 0.05, + $<$ 0.1*

Running into analysis: logistic regression

```
reglog(
  formula = health.2 ~ sex + age.3,
  imbric = list(
    . ~ association,
    . ~ work.stat
  ),
  target = 'poor',
  reference = list(
    'association' = 'Not a member',
    'age.3' = '[0,30]'
  ),
  data = study
)
```

Running into analysis: logistic regression

	Model 1	Model 2	Model 3
sexwoman	1.321 ***	1.254 ***	1.145 *
age.3(30,65]	3.113 ***	2.994 ***	3.468 ***
age.3(65,97]	5.946 ***	5.635 ***	3.598 ***
associationActive member		0.579 ***	0.595 ***
associationPassive member		0.618 ***	0.619 ***
work.statunemployed			2.460 ***
work.statnot in labor force			2.393 ***
(Intercept)	0.056 ***	0.071 ***	0.054 ***

Table: *Estimated coefficients (odds ratios) , $*** < 0.001$, $** < 0.01$, $* < 0.05$, $+ < 0.1$, $'' = NA$*

Running into analysis: logistic regression

	Model 1	Model 2	Model 3
Deviance	6317.61	6259.56	6147.26
Deviance H0	6626.53	6626.53	6626.53
Model Chi2	308.92 ***	366.97 ***	479.27 ***
Model DF	3.00	5.00	7.00
Block Chi2	308.92 ***	58.05 ***	112.30 ***
Block DF	3.00	2.00	2.00
R2 Cox-Snell	0.04	0.05	0.06
R2 Nagelkerke	0.07	0.08	0.11
N parameters	4.00	6.00	8.00
AIC	6520.87	6468.36	6354.56
BIC	7223.78	7190.42	7071.45
N	7454.00	7454.00	7454.00

Table: Quality measures , *** < 0.001, ** < 0.01, * < 0.05, + < 0.1, '' = NA

Conclusion

- The toolbox provides an efficient and secure framework for handling complex survey data
- Encouraging feedback from users
- Longitudinal and network versions forthcoming

Conclusion

- The toolbox provides an efficient and secure framework for handling complex survey data
- Encouraging feedback from users
- Longitudinal and network versions forthcoming

Conclusion

- The toolbox provides an efficient and secure framework for handling complex survey data
- Encouraging feedback from users
- Longitudinal and network versions forthcoming

Future work

- Facilitate export of data and analysis outputs in csv/tsv
- Add front-ends for other popular methods, especially:
  - Survival analysis
  - Structural equation modeling

To request features: `dataset-requests@lists.r-forge.r-project.org`

Future work

- Facilitate export of data and analysis outputs in csv/tsv
- Add front-ends for other popular methods, especially:
  - Survival analysis
  - Structural equation modeling

To request features: `dataset-requests@lists.r-forge.r-project.org`

Swiss National Centre of Competence in Research

Future work

- Facilitate export of data and analysis outputs in csv/tsv
- Add front-ends for other popular methods, especially:
  - Survival analysis
  - Structural equation modeling

To request features: `dataset-requests@lists.r-forge.r-project.org`

LIVES
Swiss National Centre of Competence in Research

Future work

- Facilitate export of data and analysis outputs in csv/tsv
- Add front-ends for other popular methods, especially:
  - Survival analysis
  - Structural equation modeling

To request features: `dataset-requests@lists.r-forge.r-project.org`

Future work

- Facilitate export of data and analysis outputs in csv/tsv
- Add front-ends for other popular methods, especially:
  - Survival analysis
  - Structural equation modeling

To request features: `dataset-requests@lists.r-forge.r-project.org`

Future work

- Facilitate export of data and analysis outputs in csv/tsv
- Add front-ends for other popular methods, especially:
  - Survival analysis
  - Structural equation modeling

To request features: `dataset-requests@lists.r-forge.r-project.org`

Selected bibliography I

[De Vries] De Vries, A. (2012) surveydata: Tools to manipulate survey data. *R package version 0.1-11*.

[Elff] Elff, M. (2013) memisc: Tools for Management of Survey Data, Graphics, Programming, Statistics and Simulation. *R package version 0.95-39*.

[Gabadinho et al.] Gabadinho, A., Ritschard, G., Müller, N.S. & Studer, M. (2011) Analyzing and visualizing state sequences in R with TraMineR *Journal of Statistical Software*. Vol. 40(4), pp. 1-37.

[Rousseaux et al.] Rousseaux E. Ritschard G. (2013) he Dataset project: Handling survey data in R *In 7th International Conference of Panel Data Users in Switzerland*. February 15-16th, 2013, pp. 37-38.

[Voorpostel et al.] Voorpostel, M., Tillmann, R., Lebert, F., Kuhn, U., Lipps, O., Ryser, V.-A., Schmid, F., Rothenbuehler, M., Wernli, B. *Swiss Household Panel Userguide (1999-2011), Wave 13*. Lausanne: FORS.(October 2012).

Thank you for your attention

Any question?