R-Package `surveysd` - Error estimation for surveys with rotating panel design

Johannes, Gussenbauer, Kowarik Alexander, Matthias Till
Statistik Austria

May, 2017

Motivation

- EU-SILC and at risk of social exclusion (‘arose‘)
- Qualitatively high well-being indicators at national or NUTS1
- Lower NUTS-Levels usually yield poor estimates
- Methodology, which is easy to apply and yields better estimates on sub-national levels?

Methods

- Many techniques already exist for estimating indicators on sub-national levels.
- Already existing techniques, for example
  - Small area estimation
  - Use administrative data to impute variable of interest (see povmap)
- Modells need assumptions and administrative data is not always available
- Need a more harmonious approach R-Package → `surveysd`

surveysd

- R-package for variance estimation on regional levels
- Variance estimation via bootstrap techniques
- Uses multiple (consecutive) waves of a survey
  - Similar approach as proposed by VIJAY, but with bootstrap instead of jack-knife
- Easy to use, even for R-Beginners

Main functionality

- Draw bootstrap replicates ‘draw.bootstrap()‘
- Calibrate bootstrap replicates ‘recalib()‘
- Estimate standard errors ‘calc.stError()‘

Draw bootstrap replicates

```
draw.bootstrap(dat,REP=1000,hid="DB030",weights="RB050",
               year="RB010",strata="DB040",cluster=NULL,
               totals=NULL,single.PSU=c("merge","mean"),
               boot.names=NULL,country=NULL,split=FALSE,pid=NULL)
```

- Rectangular data set with household identifier
- Column with sampling weight, year
- Define arbitrary sampling design with `strata` and `cluster`
- Automatic detection and dealing with single PSUs
- Bootstrap replicates are drawn for each year.
  - Applies rescaled bootstrap for stratified multistage sampling
- Replicates are taken forward to mimic rotational panel design.
  - Split households can be considered for this step, `split=TRUE`

Draw bootstrap replicates

```
UDB_AT_boot <- draw.bootstrap(UDB_AT,REP=10,hid="DB030",
                              weights="RB050",year="RB010",
                              strata=c("DB040","RB090"),
                              split=TRUE,pid="RB030")

unique(UDB_AT_boot[,.(DB030,w1,w2,w3)])

##          DB030          w1          w2          w3
##     1:       4 0.009675454 1.999939348 1.999939348
##     2:       5 1.998747116 0.001252884 1.998747116
##     3:       6 1.999860899 0.002486192 0.002486192
##     4:       7 0.003477683 1.996522317 1.996522317
##     5:       9 1.998747116 1.998747116 1.998747116
##    ---
## 18152: 4954200 0.001175986 0.001175986 0.001175986
## 18153: 4954300 0.002080739 0.002080739 1.999856969
## 18154: 4954600 0.002080739 1.999856969 1.999856969
## 18155: 4954800 0.001175986 1.998824014 1.998824014
## 18156: 4954900 1.998824014 0.001175986 1.998824014
```

Calibrate Bootsrap Replicates

```
recalib(dat,hid="DB030",weights="RB050",
        b.rep=paste0("w",1:1000),year="RB010",
        country=NULL,conP.var=c("RB090"),
        conH.var=c("DB040","DB100"),...)
```

- Use output of `draw.bootstrap()` or
- Rectangular data set with household identifier and bootstrap replicates.
- Define households and/or personal variables to be calibrated onto
- Calibration with `ipu2()` from Package simPop

Estimate standard errors

```
calc.stError(dat,weights="RB050",b.weights=paste0("w",1:1000),
             year="RB010",var="HX080",fun="weightedRatio",
             cross_var=NULL,year.diff=NULL,year.mean=3,bias=FALSE,
             add.arg=NULL,size.limit=20,cv.limit=10,p=NULL)
```

- Use output of `recalib()` or rectangular data with bootstrap weights.
- Function fun is applied on Variable var for, using each bootstrap weight.
- Predefined functions available, also able to handle custom functions or functions from other packages
  - Must return double or integer and second argument is weight

Estimate standard errors

- Define subgroups of sample using `cross_var` (optional)
- Estimate standard errors for changes between years with `year.diff` (optional)
- Results of point estimates are averaged over `year.mean` years (optional)
  - Apply filter with equal filter weights over time series
- Estimate quantiles using parameter `p`.

Estimate standard errors

```
res <- calc.stError(UDB_AT_calib,weights="RB050",
                    year="RB010",b.weights=paste0("w",1:10),
                    var="HX080",cross_var=list("DB040",c("RB090","D
res

## Calculated point estimates for variable(s)
##
##  HX080
##
## using function weightedRatio
##
## Results hold 448 point estimates for 9 years in 28 subgroups
##
## Estimted standard error exceeds 10 % of the the point estimate i
```

Estimate standard errors

```r
# Apply function which is not in package 'surveysd'
# take the gini - index
library(laeken,quietly=TRUE)
# simulate income
UDB_AT_calib[,income:=
               exp(rnorm(.N,mean=sample(7:10,1),sd=0.5)),
             by=list(DB100)]

# gini() returns list
# calc.stError needs function that returns double or integer
help_gini <- function(x,w){
  return(gini(x,w)$value)
}
```

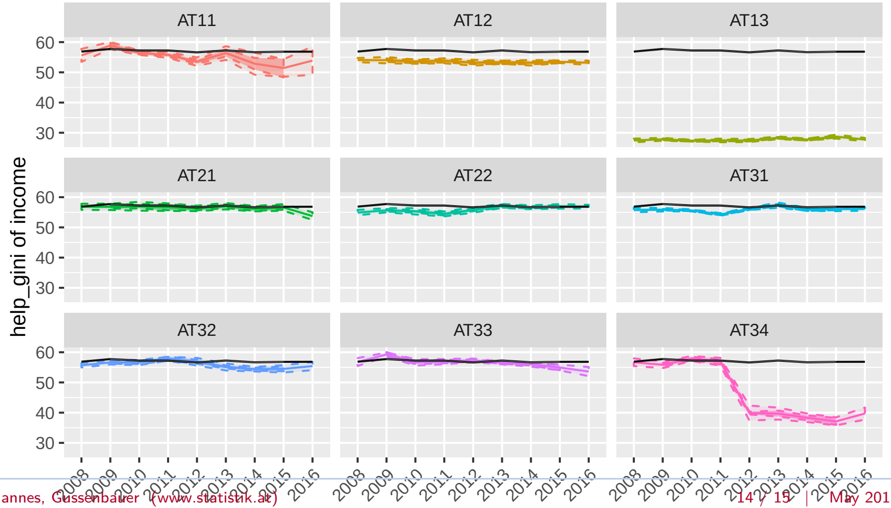
Estimate standard errors

```
res_inc <- calc.stError(UDB_AT_calib,fun="help_gini",
                    weights="RB050",year="RB010",b.weights=paste0("
                    var="income",cross_var=list("DB040",c("RB090","
                    year.diff=c("2014-2008"),p=c(.025,.975))
res_inc

## Calculated point estimates for variable(s)
##
##  income
##
## using function help_gini from .GlobalEnv
##
## Results hold 504 point estimates for 9 years in 28 subgroups
##
## Estimted standard error exceeds 10 % of the the point estimate i
```

Plot Method

```
plot(res_inc,type="grouping",
     groups="DB040",sd.type="ribbon")
```



Final Remarks

- R-Package surveysd for error estimation on surveys with rotating panel design
  - Can be applied on surveys without rotating panel design or single year
  - But less functionality available
- Simple to use R-Package which support a harmonious approach for estimating standard errors
  - Small area estimation needs modelling assumptions
  - Administrative data not always available
- Other R-Package `vardpoor` for error estimation using *ultimate cluster approach*
  - Sampling design not fully represented
  - Error estimation through linearization of given point estimates
- Check it out on github: https://github.com/statistikat/surveysd