Motivation

- EU-SILC → poverty rates
- High quality indicators on national- but estimates on sub-national level have poor accuracy
  - SAE-Methods → modelling assumptions
  - Use administrative data (see (Qinghua and Lanjouw 2009)) → not always available
- Estimate error of differences between waves → many covariates (tedious)
- Methodology, which is easy to apply and yields better estimates on sub-national levels?
- → R-Package `surveysd`

Motivation

- EU-SILC → poverty rates
- High quality indicators on national- but estimates on sub-national level have poor accuracy
  - SAE-Methods → modelling assumptions
  - Use administrative data (see (Qinghua and Lanjouw 2009)) → not always available
- Estimate error of differences between waves → many covariates (tedious)
- Methodology, which is easy to apply and yields better estimates on sub-national levels?
- → R-Package `surveysd`

surveysd

- R-package for variance estimation on surveys with rotating panel design

- Variance estimation via bootstrap techniques
  - Rescaled bootstrap for stratified multistage sampling (Preston, 2009)

- Improve accuracy by using multiple (consecutive) waves of the survey
  - Average bootstrap replicates over waves (Betti et al., 2012)

- 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
- Describe sampling design with `strata` and `cluster`
- Automatic detection and dealing with single PSUs
- Replicates are taken forward to mimic rotational panel design
    - Split households are considered

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
- Describe sampling design with `strata` and `cluster`
- Automatic detection and dealing with single PSUs
- Replicates are taken forward to mimic rotational panel design
    - Split households are considered

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"),...)
```

- Calibration with `ipu2()` from Package simPop
- Define households and/or personal variables to be calibrated onto

Estimate standard errors

```r
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` 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

```
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` 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

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

Estimate standard errors

```
calc.stError(UDB_AT,weights="weights",
             year="year",b.weights=paste0("w",1:10),
             var="poverty",cross_var=list("region",c("gender","region")))

## Calculated point estimates for variable(s)
##
##  poverty
##
## using function weightedRatio
##
## Results hold 448 point estimates for 9 years in 28 subgroups
##
## Estimted standard error exceeds 10 % of the the point estimate in 246 c
```

Estimate standard errors

```
# Apply function which is not in package 'surveysd'
# take the gini - index
library(laeken,quietly=TRUE)
# simulate income
set.seed(1234)
UDB_AT[,income:=
               exp(rnorm(.N,mean=sample(7:10,1),sd=1)),
             by=list(urban)]

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

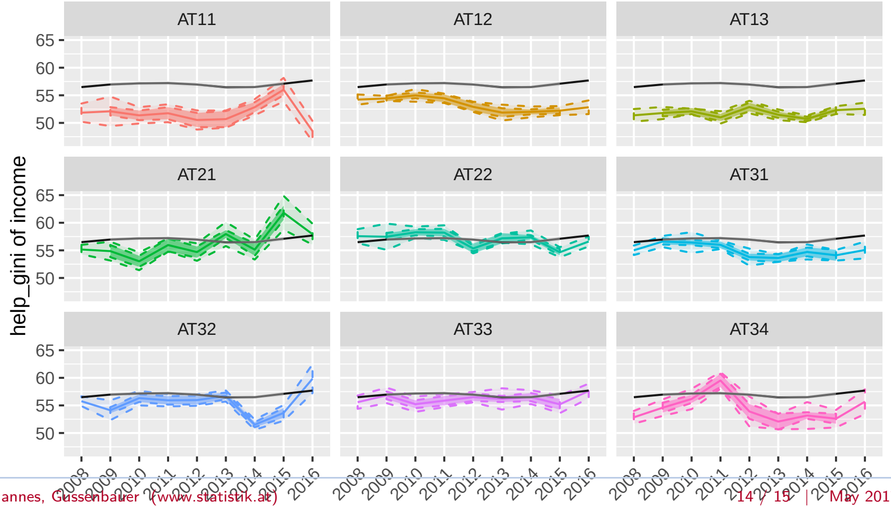
Estimate standard errors

```r
calc.stError(UDB_AT,fun="help_gini",
             weights="weights",year="year",b.weights=paste0("w",1:10),
             var="income",cross_var=list("region",c("gender","region")),
             year.diff=c("2014-2008"),p=c(.025,.975))

## 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 in 22 ca
```

Plot Method

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



Final Remarks

- Simple to use R-Package
- Supports a harmonious approach for estimating standard errors on surveys with rotating panel design
  - Achieve more accuracy by averaging over multiple years
  - No need for administrative data or modelling assumptions
- Check it out on github: https://github.com/statistikat/surveysd