# The R 'sampling' package

Alina Matei and Yves Tillé
University of Neuchâtel

Cardiff, Q2006
April 2006

# The R language

- Shareware available on http://cran.r-project.org/

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network
- Installation: 10 minutes

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network
- Installation: 10 minutes
- All the manuels are available in pdf

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network
- Installation: 10 minutes
- All the manuels are available in pdf
- Everyone can write an additional package (600 packages are available)

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network
- Installation: 10 minutes
- All the manuels are available in pdf
- Everyone can write an additional package (600 packages are available)
- Packages are loaded directly from R.

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network
- Installation: 10 minutes
- All the manuels are available in pdf
- Everyone can write an additional package (600 packages are available)
- Packages are loaded directly from R.
- The manual of the package is available online and in pdf.

# The R language

- Shareware available on http://cran.r-project.org/
- The Comprehensive R Archive Network
- Installation: 10 minutes
- All the manuels are available in pdf
- Everyone can write an additional package (600 packages are available)
- Packages are loaded directly from R.
- The manual of the package is available online and in pdf.
- Package 'sampling' written by Matei and Tillé.

# Continuous distributions

- EFTA course for public statisticians (April 2005).

# Continuous distributions

- EFTA course for public statisticians (April 2005).
- Objective : to apply directly the theory with the R language.

# Continuous distributions

- EFTA course for public statisticians (April 2005).
- Objective : to apply directly the theory with the R language.
- Theory + Exercices with a laptop and R.

# Continuous distributions

- EFTA course for public statisticians (April 2005).
- Objective : to apply directly the theory with the R language.
- Theory + Exercices with a laptop and R.
- Writing of a large set of procedures.

# Continuous distributions

- EFTA course for public statisticians (April 2005).
- Objective : to apply directly the theory with the R language.
- Theory + Exercices with a laptop and R.
- Writing of a large set of procedures.
- Finally, decision of submitting the package to the CRAN.

# Content of the package

- Stratification, two-stage, unequal probabilities, balanced sampling

# Content of the package

- Stratification, two-stage, unequal probabilities, balanced sampling
- Estimation: calibration and regression estimator

# Content of the package

- Stratification, two-stage, unequal probabilities, balanced sampling
- Estimation: calibration and regression estimator
- Tools : computation of inclusion probabilities, crossing strata

# Content of the package

- Stratification, two-stage, unequal probabilities, balanced sampling
- Estimation: calibration and regression estimator
- Tools : computation of inclusion probabilities, crossing strata
- Data bases, Swiss municipalities, Belgian municipalities.

# Tools

- `writesample`: return the list of all the samples of fixed sample size

# Tools

- `writesample`: return the list of all the samples of fixed sample size
- `cleanstrata`: renumbering of the strata

# Tools

- `writesample`: return the list of all the samples of fixed sample size
- `cleanstrata`: renumbering of the strata
- `disjonctive` return a matrix wit 0 and 1 that is the disjonctive representation of the stratum.

# Tools

- `writesample`: return the list of all the samples of fixed sample size
- `cleanstrata`: renumbering of the strata
- `disjonctive` return a matrix wit 0 and 1 that is the disjonctive representation of the stratum.
- `inclusionprobabilities`: compute unequal inclusion probabilities from an auxiliary variable variable.

# Data bases

- `MU284` A data frame with 284 municipalities on the following 11 variables : populations, political results.

# Data bases

- `MU284` A data frame with 284 municipalities on the following 11 variables : populations, political results.
- `swissmunicipalities`: 2896 Swiss municipalities. Surfaces and population.

# Data bases

- `MU284` A data frame with 284 municipalities on the following 11 variables : populations, political results.
- `swissmunicipalities`: 2896 Swiss municipalities. Surfaces and population.
- `belgianmunicipalities`: 589 Belgian municipalities 11 variables, population and taxes.

# Simple random sampling

- `srswor`: Simple random sampling with replacement.

# Simple random sampling

- `srswor`: Simple random sampling with replacement.
- `srswor1`: Simple random sampling without replacement (sequential method).

# Simple random sampling

- `srswor`: Simple random sampling with replacement.
- `srswor1`: Simple random sampling without replacement (sequential method).
- `srswr`: Simple random sampling with replacement.

# Unequal probability sampling

- `UPbrewer`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,
- `UPmultinomial`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,
- `UPmultinomial`,
- `UPpivotal`, `UPrandompivotal`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,
- `UPmultinomial`,
- `UPpivotal`, `UPrandompivotal`,
- `UPpoisson`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,
- `UPmultinomial`,
- `UPpivotal`, `UPrandompivotal`,
- `UPpoisson`,
- `UPsampford`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,
- `UPmultinomial`,
- `UPpivotal`, `UPrandompivotal`,
- `UPpoisson`,
- `UPsampford`,
- `UPsystematic`, `UPrandomsystematic`, `UPsystematicpi2`,

# Unequal probability sampling

- `UPbrewer`,
- `UPmaxentropy`, (set of function)
- `UPmidzuno`, `UPmidzunopi2`,
- `UPmultinomial`,
- `UPpivotal`, `UPrandompivotal`,
- `UPpoisson`,
- `UPsampford`,
- `UPsystematic`, `UPrandomsystematic`, `UPsystematicpi2`,
- `UPtille`, `UPtillepi2`,

# Balanced sampling

- Design that satisfies the balancing equations

$$\sum_{k \in S} \frac{\mathbf{x}_k}{\pi_k} = \sum_{k \in U} \mathbf{x}_k,$$

where $\mathbf{x}_k$ is a vector of auxiliary variables.

# Balanced sampling

- Design that satisfies the balancing equations

$$\sum_{k \in S} \frac{\mathbf{x}_k}{\pi_k} = \sum_{k \in U} \mathbf{x}_k,$$

  where $\mathbf{x}_k$ is a vector of auxiliary variables.
- Cube algorithm: flight phase and landing phase.

# Balanced sampling

- Design that satisfies the balancing equations

$$\sum_{k \in S} \frac{\mathbf{x}_k}{\pi_k} = \sum_{k \in U} \mathbf{x}_k,$$

  where $\mathbf{x}_k$ is a vector of auxiliary variables.
- Cube algorithm: flight phase and landing phase.
- `samplecube`, `fastflightcube`, `landingcube`

# Balanced sampling

- Design that satisfies the balancing equations

$$\sum_{k \in S} \frac{\mathbf{x}_k}{\pi_k} = \sum_{k \in U} \mathbf{x}_k,$$

  where $\mathbf{x}_k$ is a vector of auxiliary variables.
- Cube algorithm: flight phase and landing phase.
- `samplecube`, `fastflightcube`, `landingcube`
- Complex survey `balancedstratification` `balancedcluster` `balancedtwostage`

# Exercises

## Exercise

*Compute inclusion probabilities 200 Belgian municipalities with inclusion probabilities proportional to the population in 2004.*

# Exercises

## Exercise

*Use the Belgian database. Select a sample of 200 municipalities with unequal probabilities proportional to the number of inhabitants in 2004.*

- *with Poisson sampling*
- *with a method of unequal probabilities and fixed sample size*
- *with simple random sampling.*

*Compute the Horvitz-Thompson estimators of the taxable income for 50 samples and draw a boxplot of the estimators for each method.*

# Exercises

## Exercise

*Use the database of Swiss municipalities, and select a stratified balanced sample. A balanced sample is first selected in each strata. Next the results of the flight phase are merged and a flight phase is applied again on the whole population. Finally, a landing phase is applied on all the population. Use the following balancing variables: HApoly, Surfacesbois, P00BMTOT, P00BWTOT, POPTOT, Pop020, Pop2040, Pop4065, Pop65P, H00PTOT. The sample size is* 400 *and the municipalities must be selected with inclusion probabilities proportional to POPTOT. The stratification variable is REG (swiss regions). Next, print the names of the selected municipalities.*

# Exercises

## Exercise

*Use the Belgian database. Select a sample of 200 municipalities with unequal probabilities proportional to the number of inhabitants in 2004 with Poisson sampling design. Next calibrate the sample by means of the raking ratio estimator on the variables: mean(Men03), mean(Women03), Diffmen, Diffwom, TaxableIncome, Totaltaxation, averageincome, medianincome. The division by the means is necessary to avoid too large numbers. Compute the Horvitz-Thompson estimators and the calibrated estimators for the calibration variables. Limit the variation of the g-weights between 0.5 and 1.5.*