Metody losowania

Maciej Beręsewicz

23 Oct 2014

Wstęp

Poniższy dokument zawiera informację o tym jak losować jednostki według schematów nieprostych z pakietem *sampling*. Struktura dokumentu jest następująca. W pierwszej części poznamy schematy losowania jednostek w przypadku gdy prawdopodobieństwa wylosowania poszczególnych jednostek są takie same. W drugiej części poznamy schematy losowania jednostek gdy prawdopodbieństwo inkluzji jednostek jest różne. Ostatnia część poświęcona będzie pakietowi *survey* oraz temu w jaki sposób deklarujemy poszczególne schematy losowania, które poznamy we wcześniejszych podpunktach.

Przed poznaniem funkcji musimy zainstalować pakiety oraz wczytać dane. Poniższy kod umożliwia instalowanie oraz wczytywanie pakietów.

```
#### instalowanie pakietów ----------------------------------
install.packages(c('survey','sampling','ggplot2','dplyr'),
                 dependencies=T)
#### wczytanie pakietów ----------------------------------
library(survey)
library(sampling)
library(ggplot2)
library(dplyr)
```

Możemy sprawdzić funkcją *sessionInfo()* czy pakiety, które wczytaliśmy są rzeczywiście uruchomione. Szukamy ich nazw pod *other attached packages*.

```
sessionInfo()
```

```
## R version 3.1.1 (2014-07-10)
## Platform: x86_64-apple-darwin13.1.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] knitr_1.7     dplyr_0.3.0.2 ggplot2_1.0.0 sampling_2.6  survey_3.30-3
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.1   colorspace_1.2-4 DBI_0.3.1        digest_0.6.4
##  [5] evaluate_0.5.5   formatR_1.0      gtable_0.1.2     htmltools_0.2.6
##  [9] lpSolve_5.6.10   magrittr_1.0.1   MASS_7.3-35      munsell_0.4.2
## [13] parallel_3.1.1   plyr_1.8.1       proto_0.3-10     Rcpp_0.11.3
## [17] reshape2_1.4     rmarkdown_0.3.3  scales_0.2.4     stringr_0.6.2
## [21] tools_3.1.1      yaml_2.1.13
```

Po sprawdzeniu czy pakiety są zainstalowane możemy przejść do wczytania danych. Skorzystamy ze zbioru dla badania PISA 2009 i na potrzeby tego badania przyjmiemy, że jest to pseudopopulacja.

```
setwd("/Users/MaciejBeresewicz/Documents/Projects/GitHub/UNIVERSITY/Course Materials/Survey Design and A
load('datasets/pisa2009pol.rda')
```

Poniżej znajdują się podstawowe informacje o zbiorze danych o szkołach:

```
summary(school2009)
```

```
##     SCHOOLID      SC02Q01          SC04Q01      SC06Q01
##  00001  :  1   Public :166   Village   :58   Min.   : 12.0
##  00002  :  1   Private: 19   Small Town:36   1st Qu.: 89.5
##  00003  :  1                 Town      :44   Median :142.5
##  00004  :  1                 City      :38   Mean   :158.8
##  00005  :  1                 Large City: 9   3rd Qu.:217.5
##  00006  :  1                                 Max.   :407.0
##  (Other):179                                 NA's   :7
##     SC06Q02          SCHSIZE
##  Min.   :  0.00   Min.   : 28.0
##  1st Qu.: 88.25   1st Qu.:174.2
##  Median :141.00   Median :283.0
##  Mean   :161.75   Mean   :320.6
##  3rd Qu.:222.50   3rd Qu.:442.8
##  Max.   :483.00   Max.   :890.0
##  NA's   :7        NA's   :7
```

oraz o studentach

```
summary(student2009)
```

```
##     SCHOOLID       STIDSTD       ST04Q01              ST21Q03
##  00028  :  35   00001  :   1   Female:2474   None         : 245
##  00030  :  35   00002  :   1   Male  :2443   One          :2714
##  00058  :  35   00003  :   1                 Two          :1320
##  00067  :  35   00004  :   1                 Three or more: 595
##  00073  :  35   00005  :   1                 NA's         :  43
##  00126  :  35   00006  :   1
##  (Other):4707   (Other):4911
##                 ST22Q01        PV1MATH         PV2MATH
##  0-10 books         : 460   Min.   :229.6   Min.   :186.9
##  11-25 books        : 936   1st Qu.:438.7   1st Qu.:439.9
##  26-100 books       :1638   Median :499.1   Median :498.5
##  101-200 books      : 863   Mean   :499.2   Mean   :500.0
##  201-500 books      : 610   3rd Qu.:562.2   3rd Qu.:563.1
##  More than 500 books: 371   Max.   :818.6   Max.   :810.7
##  NA's               :  39
##     PV3MATH         PV4MATH         PV5MATH
##  Min.   :201.6   Min.   :208.2   Min.   :207.1
##  1st Qu.:437.6   1st Qu.:436.8   1st Qu.:438.3
##  Median :499.9   Median :499.1   Median :499.2
##  Mean   :499.2   Mean   :500.0   Mean   :500.2
```

```
##  3rd Qu.:561.5   3rd Qu.:561.4   3rd Qu.:561.5
##  Max.   :785.2   Max.   :792.0   Max.   :813.0
##
```

Losowanie z wykorzystaniem funkcji sample oraz pakietu sampling

W tej części zajmiemy się różnymi schematami losowania, które są dostępne w pakiecie sampling, jak również z domyślną funkcją *sample*.

Losowanie proste bez zwracania

Pierwszym i najprostrzym schematem losowania jest losowanie proste bez zwracania. W takim przypadku losujemy określony podzbiór jednostek bez uwzględniania zmiennych pomocniczych (np. płci, wieku). W R możemy wykorzystać w tym celu funkcję sample lub sampling::srswor

Poniżej przykład wykorzystania funkcji dla zbioru studenci. Załóżmy, że naszym celem jest oszacowanie liczby książek, którą posiadają studenci w domu, jak również średniego poziomu dla zmiennej PV1MATH określającego poziom umiejętności matematycznych.

```
### ID Studentów
id_student <- student2009$STIDSTD

### Frakcja, którą losujemy
prop_wylos <- 0.1

### Wielkość próby
n_wylos <- round(length(id_student)*prop_wylos)

### identyfikatory wylosowanych studetów
set.seed(123) ## ustawienie ziarna losowania tak aby każda osoba dostała taki sam wynik
id_s_wylos <- sample(x = id_student,
                     size = n_wylos)

### wylosowani studenci
student2009_samp <- subset(student2009, STIDSTD %in% id_s_wylos)
```

Sprawdzmy teraz jakie wyniki otrzymujemy w przypadku losowania prostego i porównamy do populacji.

```
### porównanie średniej
c(SREDNIA_PROBA = mean(student2009_samp$PV1MATH),
  SREDNIA_POPULACJA = mean(student2009$PV1MATH))
```

```
##     SREDNIA_PROBA SREDNIA_POPULACJA
##          499.0563          499.2163
```

```
### porównanie rozkładu
summary(student2009_samp$PV1MATH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   248.4   438.2   497.7   499.1   556.0   818.6
```

```r
summary(student2009$PV1MATH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   229.6   438.7   499.2   499.2   562.2   818.6
```

```r
### porównanie rozkładu dla ksiażek
round(prop.table(table(student2009_samp$ST22Q01))*100,2)
```

```
##
##          0-10 books         11-25 books        26-100 books
##                8.85               17.28               34.57
##       101-200 books       201-500 books More than 500 books
##               17.90               14.61                6.79
```

```r
round(prop.table(table(student2009$ST22Q01))*100,2)
```

```
##
##          0-10 books         11-25 books        26-100 books
##                9.43               19.19               33.58
##       101-200 books       201-500 books More than 500 books
##               17.69               12.51                7.61
```

Teraz wykorzystamy funkcję sampling::srswor

```r
### ID Studentów
id_student <- student2009$STIDSTD

### Frakcja, którą losujemy
prop_wylos <- 0.1

### Wielkość próby
n_wylos <- round(length(id_student)*prop_wylos)

### losujemy
set.seed(123)
row_wylos <- srswor(n_wylos,nrow(student2009))

### sprawdzmy jak wygląda wynik
head(row_wylos)
```

```
## [1] 0 0 1 1 0 0
```

```r
### wybieranie jednostek
stu_samp <- student2009[row_wylos, ]
```

Losowanie proste ze zwracaniem

Losowanie systematyczne

Losowanie systematyczne polega na wylosowaniu określonej liczby jednostek wykorzystując stały interwał między wybieranymi jednostkami (np. co 5). Interwał określany jest przez stosunek wielkości populacji

do wielkości próby. Losowanie systematyczne może również uwzględniać fakt, że jednostki maja nierówne prawdopodobieństwa dostania się do próby. W tym przykładzie zastosujemy dwa podejścia - losowanie z równymi prawdopodobieństwami oraz losowanie z nierównymi prawdopodobieństwami.

W pierwszym przypadku wykorzystamy funkcję *seq*, która umożliwia tworzenie wektorów z pewnym krokiem. Istotny jest również element określający start losowania systematycznego, co wpływa na dobór jednostek.

```
### wielkość populacji
N <- nrow(student2009)

## wielkość próby
n <- round(N*prop_wylos)

## krok
k <- N/n

## losowanie początku
set.seed(123)
start <- sample(1:k,1)

## losowanie systematyczne
id_wylos <- seq(from = start,
                to = N,
                by = round(k))

### wylosowanie jednostki
stu_samp <- student2009[id_wylos,]
```

Poniżej przedstawiamy porównanie w przypadku losowania systematycznego

```
### porównanie średniej
c(SREDNIA_PROBA = mean(stu_samp$PV1MATH),
  SREDNIA_POPULACJA = mean(student2009$PV1MATH))
```

```
##     SREDNIA_PROBA SREDNIA_POPULACJA
##          498.6766          499.2163
```

```
### porównanie rozkładu
summary(stu_samp$PV1MATH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   281.4   443.8   496.8   498.7   556.2   732.4
```

```
summary(student2009$PV1MATH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   229.6   438.7   499.2   499.2   562.2   818.6
```

```
### porównanie rozkładu dla ksiażek
round(prop.table(table(stu_samp$ST22Q01))*100,2)
```

```
##
##          0-10 books         11-25 books        26-100 books
##               10.02               17.79               34.97
##       101-200 books       201-500 books More than 500 books
##               16.97               13.70                6.54
```

```
round(prop.table(table(student2009$ST22Q01))*100,2)
```

```
##
##          0-10 books         11-25 books        26-100 books
##                9.43               19.19               33.58
##       101-200 books       201-500 books More than 500 books
##               17.69               12.51                7.61
```

Losowanie proporcjonalne do zmiennej pomocniczej

W tym celu wykorzystamy funkcję sampling::UPpoisson - jeden ze schematów losowania wykorzystywany w przypadku przyjęcia za zmienną pomocniczą zmienną ciągłą. Najczęściej takie losowanie wykorzysujemy gdy chcemy losować jednostki proporcjonalnie do ich wielkości (np. liczby zatrudnionych, przychodów). W naszym przypadku możemy losować szkoły proporcjonalnie do liczby uczniów (zmienna SCHSIZE).

Przyjrzyjmy się zmiennej SCHSIZE.

```
summary(school2009$SCHSIZE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    28.0   174.2   283.0   320.6   442.8   890.0       7
```

Niestety w zmiennej mamy braki danych dlatego musimy ogarniczyć zbiór danych do rekordów pozbawionych wartości brakujacych.

```
### wybranie pełnych obserwacji
school2009clean <- school2009[complete.cases(school2009),]

### sprawdzenie czy są braki danych
summary(school2009clean$SCHSIZE)
```

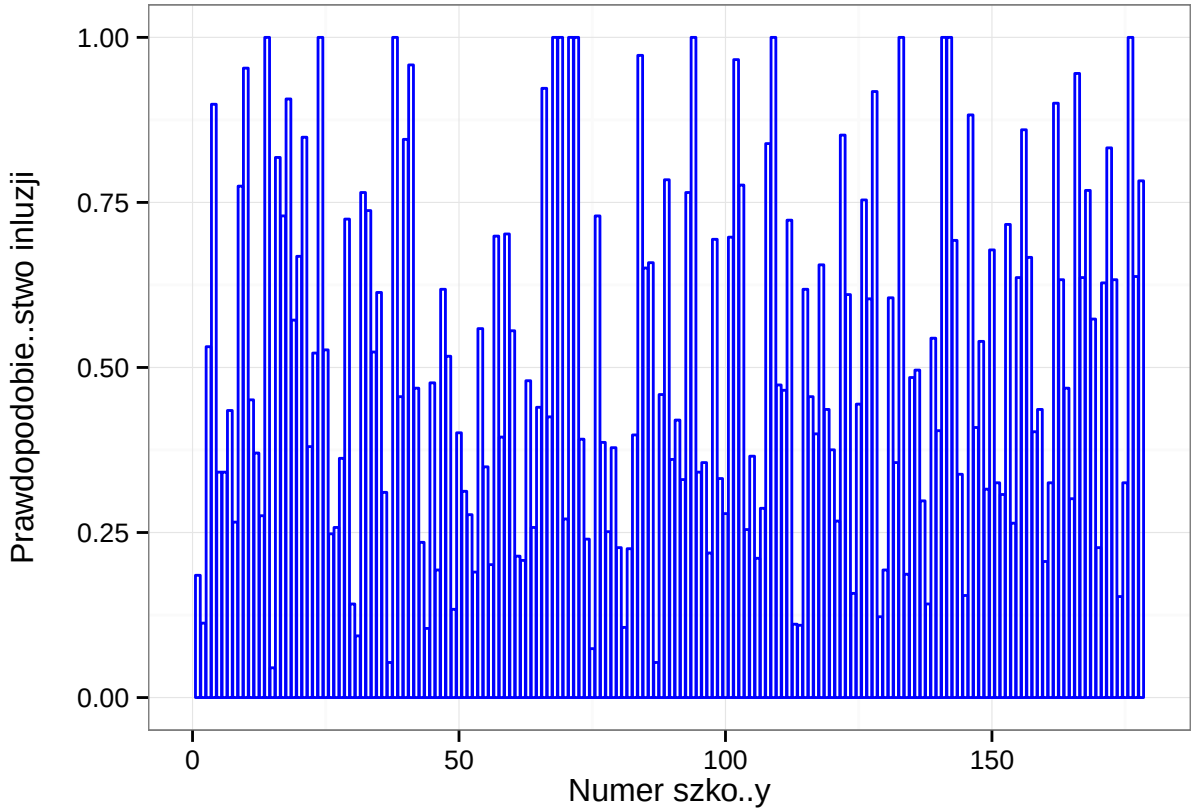
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    28.0   174.2   283.0   320.6   442.8   890.0
```

Wykorzystamy teraz funkcję sampling::UPpoisson oraz sampling::inclusionprobabilities, która służy do określenia prawdopodobieństw inkluzji poszczególnych jednostek.

```
### wielkość próby
n <- 90
### obliczamy prawdopodobieństwa
pik <- inclusionprobabilities(school2009clean$SCHSIZE,n)

### wykres jak zmienia jak wygląda prawdopodobieństwo
n_sch <- nrow(school2009clean)
```

```
qplot(x=1:n_sch,y=pik,geom='bar') +
  geom_bar(col='blue',fill=NA,stat='identity') +
  theme_bw() +
  xlab('Numer szkoły') + ylab('Prawdopodobieństwo inluzji\n')
```



Po ustaleniu prawdopodobieństw wylosowania poszczególnych jednostek stosujemy funkcję sampling::UPpoisson. W wyniku otrzymujemy wektor składający się z 0-1, który określa czy dana jednostka jest wylosowana (1) lub niejest wylosowania (0).

```
### losowanie z wykorzystaniem funkcji UPpoisson (losowa wielkość jednak zbliżona do 38)
set.seed(123)
pp_wylos <- UPpoisson(pik)
sum(pp_wylos)
```

```
## [1] 86
```

```
### identyfikatory szkoły

### wybranie jednostek
stu_pp <- school2009clean[pp_wylos==1,]
stu_pp$pik <- pik[pp_wylos==1]
stu_pp$fpc <- nrow(school2009clean)
```

Wartość globalną możemy oszacować jako suma iloczynów kolumny *SCHSIZE* oraz odwrotności *pik*, która będzie określać wagi dla poszczególnych szkół.

```r
sum(stu_pp$SCHSIZE * 1/stu_pp$pik)
```

```
## [1] 54576.36
```

Porównamy teraz szacunki z wykorzystaniem funkcji survey::svydesign

```r
### deklaracja schematu losowania
des_up <- svydesign(ids=~1,
                    probs = ~pik,
                    fpc = ~fpc,
                    data = stu_pp)

### szacowanie wielkości szkół
svytotal(~SCHSIZE,des_up)
```

```
##         total     SE
## SCHSIZE 54576 298.33
```

Losowanie warstwowe

W przypadku losowania warstwowego wykorzystamy funkcję sampling::strata. Funkcja ta umożliwia następujące losowania jednostek:

- srswor - losowanie proste bez zwracania
- srswr - losowania proste ze zwracaniem
- poisson - losowanie proporcjonalne do zmiennej pomocniczej
- systematic - losowanie systematyczne

Jeżeli chcemy wylosować jednostki proporcjonalnie do poszczególnych warstw musimy wcześniej odpowiednio przygotować zbiór danych. Naszym celem jest wylosowanie szkół proporcjonalnie do jej typu oraz lokalizacji.

Spójrzmy jaki jest rozkład tej cechy.

```r
prop.table(xtabs(~SC02Q01+SC04Q01,school2009clean))
```

```
##          SC04Q01
## SC02Q01      Village Small Town       Town       City Large City
##   Public  0.30898876 0.19662921 0.20224719 0.17977528 0.02808989
##   Private 0.01123596 0.00000000 0.02808989 0.02808989 0.01685393
```

Załóżmy, że chcemy wylosować 100 szkół. W związku z tym powinniśmy wylosować następującą liczbę szkół.

```r
round(prop.table(xtabs(~SC02Q01+SC04Q01,school2009clean))*100)
```

```
##          SC04Q01
## SC02Q01   Village Small Town Town City Large City
##   Public       31         20   20   18          3
##   Private       1          0    3    3          2
```

Poniżej funkcja, którą napisałem na potrzeby tego skryptu

```
strat_samp <- function(data,
                       samp_size,
                       ...) {

  ### sample function
  samp_one <- function(x) {
    sample(1:length(x),length(x))
  }

  ### function
  ds <- data %>%
            group_by(...) %>%
            mutate(N_strat = n(),
                   los = samp_one(N_strat)) %>%
            group_by() %>%
            mutate(N = n(),
                   n_samp = round(N_strat/N*samp_size),
                   filter = los <= n_samp) %>%
            filter(filter) %>%
            select(-(N_strat:filter))
  return(ds)
}
```

Wykorzystamy teraz powyższą funkcję do losowania (losowanie proste bez zwracania)

```
strat_samp(data=school2009, ### zbiór wejściowy
           samp_size=100, ### wielkość próby
           SC02Q01,SC04Q01)  ### zmienne warstwujące
```

```
## Source: local data frame [100 x 6]
##
##     SCHOOLID SC02Q01    SC04Q01 SC06Q01 SC06Q02 SCHSIZE
## 1      00003  Public       Town     188     142     330
## 2      00004  Public       City     212     346     558
## 3      00005  Public Small Town     105     107     212
## 4      00007  Public       City     125     145     270
## 5      00010  Public       City     311     281     592
## 6      00011  Public       City     163     117     280
## 7      00012  Public       Town     110     120     230
## 8      00013  Public    Village      79      92     171
## 9      00014  Public       Town     351     368     719
## 10     00016  Public Small Town     251     257     508
## 11     00018  Public       Town     272     291     563
## 12     00020  Public Small Town     171     184     355
## 13     00022  Public       City     255     272     527
## 14     00023  Public    Village     128     108     236
## 15     00024  Public       City     167     157     324
## 16     00028  Public    Village      75      85     160
## 17     00029 Private       City     144      81     225
## 18     00030  Public    Village     230     220     450
## 19     00031  Public    Village      38      50      88
## 20     00034  Public Small Town     230     228     458
## 21     00035  Public    Village     158     167     325
```

```
## 22     00036  Public       Town     196     185     381
## 23     00038 Private    Village      18      15      33
## 24     00042  Public       Town     294     301     595
## 25     00043  Public       City     128     163     291
## 26     00045 Private       Town      NA      NA      NA
## 27     00049  Public       Town     185     199     384
## 28     00050  Public       City     180     141     321
## 29     00053  Public       Town     101      93     194
## 30     00054  Public    Village      95      77     172
## 31     00055 Private       Town      63      55     118
## 32     00058  Public    Village      49      76     125
## 33     00059  Public Small Town     221     213     434
## 34     00060  Public Small Town     127     118     245
## 35     00061  Public       Town     235     201     436
## 36     00062  Public Large City     161     184     345
## 37     00064  Public    Village      79      50     129
## 38     00066  Public    Village      73      87     160
## 39     00067  Public    Village     148     125     273
## 40     00068  Public       Town     304     269     573
## 41     00069  Public    Village     139     125     264
## 42     00071  Public    Village      NA      NA      NA
## 43     00072  Public       Town     309     386     695
## 44     00073  Public Small Town      84      84     168
## 45     00074  Public Small Town     407     483     890
## 46     00075  Public       City     391     395     786
## 47     00078 Private       Town      30      16      46
## 48     00080  Public    Village     116     124     240
## 49     00084  Public    Village      34      32      66
## 50     00085  Public    Village      78      62     140
## 51     00086  Public    Village     131     116     247
## 52     00087  Public       City     202     402     604
## 53     00089  Public Large City     223     186     409
## 54     00090 Private Small Town      NA      NA      NA
## 55     00091  Public    Village      20      13      33
## 56     00092  Public       Town     144     141     285
## 57     00095  Public Small Town     129     132     261
## 58     00096  Public    Village      96     109     205
## 59     00097  Public       Town     218     257     475
## 60     00102  Public Large City     232     199     431
## 61     00103  Public    Village      98     108     206
## 62     00105  Public    Village     221     212     433
## 63     00110  Public    Village      61      70     131
## 64     00112  Public       Town     249     272     521
## 65     00113  Public       City     388     437     825
## 66     00115  Public       City     138     151     289
## 67     00116  Public    Village     216     233     449
## 68     00118 Private Large City      33      35      68
## 69     00119  Public       Town     196     188     384
## 70     00121  Public       City     120     128     248
## 71     00125  Public    Village     116     117     233
## 72     00128  Public Small Town     182     197     379
## 73     00129 Private       City      57      41      98
## 74     00136  Public Small Town     208     168     376
## 75     00138  Public Small Town     353     340     693
```

```
## 76     00144  Public       City     192     146     338
## 77     00145 Private Large City      NA      NA      NA
## 78     00148  Public       Town     348     347     695
## 79     00149  Public Small Town     194     236     430
## 80     00150  Public    Village     107     103     210
## 81     00152  Public       Town      NA      NA      NA
## 82     00153  Public Small Town     278     270     548
## 83     00155  Public Small Town     175     160     335
## 84     00156  Public    Village      95     101     196
## 85     00159  Public    Village      96      95     191
## 86     00161  Public    Village      84      80     164
## 87     00163  Public       Town     171     363     534
## 88     00164  Public       City     216     198     414
## 89     00165  Public Small Town     138     112     250
## 90     00167 Private       City      63      65     128
## 91     00168  Public       Town      81     121     202
## 92     00170  Public Small Town     197     196     393
## 93     00172  Public    Village      89      98     187
## 94     00176  Public       Town     193     163     356
## 95     00177  Public       City      74      67     141
## 96     00178  Public Small Town     201     189     390
## 97     00179  Public       City     275     242     517
## 98     00180  Public Small Town     218     175     393
## 99     00182  Public    Village      99     103     202
## 100    00183  Public       Town     323     315     638
```

Losowanie zespołowe

W tym miejscu połączymy zbiór szkół oraz uczniów.

```
pisa <- merge(x = student2009,
              y = school2009clean,
              all.x=T)
pisa <- pisa[complete.cases(pisa$SCHSIZE),]
dim(pisa)
```

```
## [1] 4803   15
```

```
head(pisa)
```

```
##   SCHOOLID STIDSTD ST04Q01 ST21Q03       ST22Q01 PV1MATH PV2MATH PV3MATH
## 1    00001   00023  Female     One  26-100 books  480.46  512.39  497.59
## 2    00001   00004  Female    None  26-100 books  460.36  464.25  416.74
## 3    00001   00022  Female     One  26-100 books  503.44  512.78  501.10
## 4    00001   00018  Female     One 101-200 books  641.07  662.10  584.99
## 5    00001   00012  Female     One   11-25 books  410.35  379.19  433.72
## 6    00001   00024  Female     One  26-100 books  530.31  492.92  595.74
##   PV4MATH PV5MATH SC02Q01 SC04Q01 SC06Q01 SC06Q02 SCHSIZE
## 1  537.32  444.63  Public Village      66      49     115
## 2  467.37  427.64  Public Village      66      49     115
## 3  543.16  526.80  Public Village      66      49     115
## 4  613.81  620.04  Public Village      66      49     115
```

```
## 5  439.17  411.13  Public Village      66      49     115
## 6  551.34  586.39  Public Village      66      49     115
```

W pakiecie *sampling* jest funkcja *cluster*, która umożliwia losowanie zespołowe z równymi bądź nierównymi prawdopodobieństwami inkluzji. Funkcja ma następujące argumenty:

- data - zbiór danych wejściowych
- clustername - zmienna, która określa zespoł (np. id szkoły)
- size - określa wielkość próby
- method - określa metodę losowania - prostą bez zwracania (srswor), prostą ze zwracaniem (srswr), proporcjonalną do zmiennej pomocniczej (poisson) lub systematycznie (systematic)
- pik - zmienna wg której dokonujemy losowania poissona

Poniżej kilka wywołań funkcji. W poniższym przykładzie dokonujemy losowania 60 szkół w których badamy wszystkich uczniów. Stosujemy losowanie proste bez zwracania. W wyniku otrzymujemy obiekt data.frame z informacją o wylosowanej szkole, jednostkach w ramach szkół oraz prawdopodobieństwie wylosowania, które jest przypisane do każdego ucznia.

```
wyn <- cluster(data = pisa,
               clustername = 'SCHOOLID',
               size = 60,
               method = 'srswor')
table(droplevels(wyn$SCHOOLID))
```

```
##
## 00004 00012 00015 00016 00017 00021 00026 00029 00030 00033 00035 00036
##     2    27     7    33    27    29    27    26    35    30    27    34
## 00037 00041 00043 00046 00047 00048 00053 00057 00061 00062 00064 00066
##    33    33    24    29    32    33    16    26    32    31    24    29
## 00067 00069 00074 00076 00080 00081 00083 00087 00088 00094 00099 00100
##    35    20     3    27    29    22    30     5    25    29    26    25
## 00113 00121 00123 00127 00131 00133 00143 00146 00147 00149 00151 00154
##    33    30    32     4    26    26    30    26    28    31    29    28
## 00155 00156 00160 00165 00167 00171 00173 00174 00177 00180 00181 00185
##    31    31    26    35    24    35    33    33    34    33    23    35
```

Zobaczmy jakbyśmy chcieli uwzględnić zmienną określającą wielkość szkoły tak aby wykorzystać losowanie proporcjonalne do zmiennej pomocniczej. Aby to zrobić korzystamy ze zbioru dla szkół ponieważ przy wyborze zmiennej pomocniczej musielibyśmy użyć innej zmiennej, która jest przyposana do każdego ucznia.

```
wyn <- cluster(data = school2009clean,
               clustername = 'SCHOOLID',
               size = 60,
               description = TRUE,
               pik = school2009clean$SCHSIZE,
               method = 'poisson')
```

```
## Number of selected clusters: 57
##
## Population total and number of selected units: 178 57
```

```r
head(wyn)
```

```
##   SCHOOLID ID_unit      Prob
## 1    00004       4 0.5867508
## 2    00006       6 0.2229232
## 3    00010      10 0.6225026
## 4    00012      12 0.2418507
## 5    00014      14 0.7560463
## 6    00016      16 0.5341746
```

Losowanie dwustopniowe

Przejdziemy teraz do najczęściej stosowanego schematu losowania tj. losowania dwustopniowego. W pierwszym kroku losuje się zwykle terenowe rejony spisowe (jednostki statystyczne), a w ramach tych jednostek następnie losowane są kolejne jednostki (np. mieszkania, uczniowe). W przypadku badania *PISA* w pierwszym etapie losowane były szkoły proporcjonalnie do liczby uczniów, a następnie w ramach szkół uczniowe. Spróbujmy odtworzyć to losowanie.

Możemy w tym celu wykorzystać funkcję *sampling::mstage*, która ma następujace argumenty:

- data - zbiór danych wejściowy
- stage - określamy w jaki sposób mają być losowane jednostki na poszczególnych etapach - do wyboru mamy "stratified", "cluster" lub "" co oznacza brak stosowania losowania zespołowego oraz stratyfikacyjnego.
- varames - wskazujemy zmienne, które określają zespoły lub warstwy
- pik - zmienna wykorzystywana do losowania proporcjonalnego (poissona)
- method - do wyboru są następujące metody: "srswor", "srswr", "poisson" oraz "systematic"
- description - zwraca informacje podsumowanie losowania.

Na początek przeprowadzimy losowanie jednostopniowe ale uwzględniające stratfikację względem dwóch zmiennych - SC02Q01 (status) oraz SC04Q01 (lokalizacja), a następnie losujemy szkoły.

```r
pisa <- pisa %>% arrange(SC02Q01,SCHOOLID)
wyn <- mstage(data = pisa,
              stage = list('stratified', 'cluster'),
              varnames = list('SC02Q01','SCHOOLID'),
              size = list(c(45,5),c(10,1)),
              method = 'srswor')
str(wyn,2)
```

```
## List of 2
##  $ 1:'data.frame':   50 obs. of  4 variables:
##   ..$ SC02Q01       : Factor w/ 2 levels "Public","Private": 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ ID_unit       : int [1:50] 99 189 199 404 435 467 660 785 985 1019 ...
##   ..$ Prob_ 1 _stage: num [1:50] 0.00998 0.00998 0.00998 0.00998 0.00998 ...
##   ..$ Stratum       : int [1:50] 1 1 1 1 1 1 1 1 1 1 ...
##  $ 2:'data.frame':   12 obs. of  4 variables:
##   ..$ SCHOOLID      : Factor w/ 1535 levels "00001","00002",..: 17 61 79 110 119 147 148 157 159 171
##   ..$ ID_unit       : int [1:12] 404 1437 1869 2627 2836 3482 3516 3707 3786 4059 ...
##   ..$ Prob_ 2 _stage: num [1:12] 0.238 0.238 0.238 0.238 0.238 ...
##   ..$ Prob          : num [1:12] 0.00238 0.00238 0.00238 0.00238 0.00238 ...
```

Przeprowadzimy teraz losowanie dwustopniowe - w pierwszym stopniu losujemy szkoły, a następnie losujemy uczniów.

```
### losujemy szkoły
szkoly <- strat_samp(data= school2009clean,
                     samp_size=50,
                     SC02Q01,SC04Q01)
```