Stratified and cluster sampling

Peter Lugtig

03 oktober, 2021

Introduction

In this class exercise, we are going to continue working on the “Boys” dataset, which you also used last week.

Open the `boys.RDS`. The key variables in the dataset are the height and weight for *all* 748 boys under the age of 21 living in an imaginary county (gemeente) in the Netherlands in 2014. In other words, we have the whole population, and the idea is we explore different hypothetical ways to sample from this population. The dataset stems from a surveillance system, in which every child is seen every few years. The following measures are available:

```
# install.packages("tidyverse")
# install.packages("magrittr")
# install.packages("survey")
# install.packages("sampling")

library(tidyverse)
library(magrittr)
library(survey)
```

```
## Warning: package 'survey' was built under R version 4.1.1
```

```
library(sampling)
```

```
## Warning: package 'sampling' was built under R version 4.1.1
```

```
# and load the data
boys <- read_rds("boys.RDS")
```

Table 1: Variables in the data

Variable	Description
age	Decimal age (0-21 years)
hgt	Height (cm)
wgt	Weight (kg)
bmi	Body Mass Index
hc	Head circumference (cm)
gen	Genital Tanner Stage (G1-G5)
phb	Pubic hair (Tanner P1-P6)
tv	Testicular volume (ml)
town	Town 1 to Town 5

Exercise 1: stratified sampling

In the next exercise we will will again use the dataset boys.rds. See the exercise from last week if you want to review the ideas of Simple Random Sampling for this dataset (SRS).

There are several reasons why we may not want to use SRS to sample. For example, we may want to limit our costs by sampling boys from 1 or perhaps a few towns, instead of all the towns like in SRS. This design is called cluster sampling, and we will look at this later in this exercise. First, we will focus on stratifcation. Two reasons to do this are:

a. We may want to get a minimum sample size in different groups, for example because our primary interest lies in comparing groups, rather than getting a precise estimate for the population mean.
b. We may actually want to estimate the population mean with higher precision (less uncertainty) than SRS.

The code below will draw (`without replacement`) a stratified sample based on `town` with 40 cases sampled randomly from each town (stratification on town). Use the same random seed (`123`). There are a few cases in the population where the town is unknown; these are deleted. Run the code below to draw the sample using the `strata` function. The 'strata' command comes from the 'sampling' package, and allows us to draw K sampling units from L strata.

```r
# for replication purposes
set.seed(123)
# sample size 40 in every stratum, you need to sort
# the data, to get the strata right (note that you
# then have to respecify the ID variables, because
# the upcoming strata function will treat the row id
# as the identifier).
boys2 <- boys %>%
    filter(!is.na(town)) %>%
    arrange(town) %>%
    mutate(id = 1:nrow(.))

selected <- sampling::strata(boys2, c("town"), size=c(40,40,40,40,40), "srswor")
samplestrat <- filter(boys2, id %in% selected$ID_unit)
```

You may be wondering whether sampling 40 cases from every town is wise or not. We will postpone this for now, and first show how with a stratified sample design, one specifies the 'svydesign()' correctly. One thing we need for this is to specify the sample size in the *population* for every stratum.

Q1.

What are the population sizes in every town?

The easiest way to add the sample sizes to our sample we have just drawn, is to add a variable ('fpc') to the stratified sample dataset, where we include the population size for each town. Please see the code below.

```r
# we now need to add the sample sized per stratum,
# so that R can automatically include probabilities
table(boys2$town)
```

```
##
##   1   2   3   4   5
## 191  81 239 161  73
```

```r
samplestrat$fpc[samplestrat$town==1] <- 191
samplestrat$fpc[samplestrat$town==2] <- 81
samplestrat$fpc[samplestrat$town==3] <- 239
```

```
samplestrat$fpc[samplestrat$town==4] <- 161
samplestrat$fpc[samplestrat$town==5] <- 73

# Or simpler (especially when the number of strata is larger)
# samplestrat <-
#     table(boys$town) %>%
#     as.data.frame() %>%
#     rename(town = Var1,
#            fpc  = Freq) %>%
#     mutate(town = as.numeric(town)) %>%
#     right_join(samplestrat)
```

Now, we can specify the svydesign object using the code below. The svydesign object looks very similar to the object we had for SRS, but we now add the *fpc per stratum* AND tell R *which variable we used for stratification* in our sample using the 'strata=~' command.

Q2

Run the code below. How do you interpret the design effect? Is this a good way to stratify our sample?

```
stratdesign1 <- svydesign(ids = ~1,
                          probs = NULL,
                          strata = ~town,
                          variables = NULL,
                          fpc = ~fpc,
                          weights = NULL,
                          data = samplestrat)

mean(samplestrat$age,na.rm=T) # wrong, no sampling design
```

```
## [1] 10.08854
```

```
svymean(~age, stratdesign1,deff=T)
```

```
##         mean       SE   DEff
## age 10.09323  0.45295 1.2304
```

We will discuss the answer to Q2 in class. You may continue this exercise until we reconvene, and then finish the rest of this exercise as the Take Home Exercise.

Q3

Stratified sampling can be used to draw a sample size that is 'big enough' in every stratum, as we did in question 1. However, more commonly we use stratification to increase precision in our sample, or to reduce the required sample size. One popular way is to to stratify "proportional to size" so that the sample sizes in the population are exactly reflected in the sample. Draw a sample proportional to size based on the variable "agecat" created below.

```
boys <-
    boys %>%
    mutate(agecat = cut(age,
                        breaks = c(0, 1, 5, 10, 22),
                        labels = c("Younger than 1",
                                   "1 - 5",
                                   "5 - 10",
                                   "Older than 10")))
```

Age	Sample size height	Sample size weight
Younger than 1		
1 - 5		
5 - 10		
Older than 10		

Follow these steps:

a) First, compute how many cases we need from every of the four strata if we want to draw a sample of 200.
b) draw a sample size of 200 PPS from the strata
c) add the variable '$fpc' to the new sample, and specificy the svydesign object
d) compute both the mean in height and weight (our real dependent variables!) for the boys using the sampling design.; What is the design effect?

Q4

The real benefit of stratification is to oversample from strata in the population that exhibit more variance. The dependent variables in our dataset (height, weight) are strongly predicted by age, but variance in these variables is likely to differ too for the different age categories. Compute the overall variance in height and weight, and the variance in both variables for the different age categories, along with the sample sizes. Try this first, but answers can be found on Blackboard if you get stuck.

Q5

Neyman allocation helps to define how many cases should be drawn from what stratum. Let's assume again that we can draw 200 cases from the population. Using the variable `height` only as our dependent variable, how many cases would we need from every stratum for the variable 'agecat'? Complete the table below:

Q6

And what if we would optimize the sample based on weight? Compute the optimal sample sizes again across the 4 age groups in the table above.

Q7

In practice you often are in this situation, where you have multiple dependent variables, and you have to make a choice how to optimize your sample given multiple stratification variables. Also bear in mind, that you are often using imperfect estimates for the variances in the subgroups (e.g. based on a survey from a previous year) and (as Stuart discusses in section 30A) that you often don't know how large the strata actually are. Taken all of this into account, how would you here stratify?

Q8

Compute the mean weight and height in the population using the stratified design you specified under 7. Compute the design effect. How much smaller is this than in question 1? We will start our discussion in week 41 with a discussion of your findings.

```
selected <- strata(boys3,
                   c("agecat"),
                   size = c(13,26,12,147),
                   method = "srswor")$ID_unit

samplestrat3 <- subset(boys3, id %in% selected)
```

```
# we now need to add the sample sized per stratum,
# so that R can automatically include probabilities
samplestrat3$fpc[samplestrat3$agecat=="Younger than 1"] <- 136
samplestrat3$fpc[samplestrat3$agecat=="1 - 5"] <- 155
samplestrat3$fpc[samplestrat3$agecat=="5 - 10"] <- 68
samplestrat3$fpc[samplestrat3$agecat=="Older than 10"] <- 389

# Or again, in shorthand, which is especially useful
# if the number of categories is large
# samplestrat3 <-
#     table(boys3$agecat) %>%
#     as.data.frame() %>%
#     rename(agecat = Var1,
#            fpc = Freq) %>%
#     right_join(samplestrat3)

stratdesign3 <- svydesign(ids = ~1,
                          probs = NULL,
                          strata = ~agecat,
                          variables = NULL,
                          fpc = ~fpc,
                          weights = ~NULL,
                          data = samplestrat3)

mean(samplestrat3$age) # biased, just for comparison
```

```
## [1] 12.07249
```

```
# means
svymean(~hgt, na.rm = TRUE, stratdesign3, deff = TRUE)
```

```
##          mean        SE DEff
## hgt 152.81959   0.93479  Inf
```

```
svymean(~wgt, na.rm = TRUE, stratdesign3, deff = TRUE)
```

```
##         mean       SE DEff
## wgt 47.55260  0.78964  Inf
```

Exercise 2 - Cluster sampling

We will again use the Boys.RDS dataset. The main reason to do cluster sampling is to reduce costs. In the “boys.rds” dataset, we have a population from one council, and perhaps it is safe to assume that travel costs do not matter so much. In other circumstances, travel costs do matter (as well as things like training interviewers, acessing multiple sample frames). The idea of cluster sampling is to reduce the number of clusters to save costs. Typically one does this in multiple stages: first one or several clusters are sampled, and then within the clusters a further sample is drawn.

Q1

The clusters in this dataset consist of the 5 towns. How many boys live in each town in the population?

```
table(boys$town)
```

```
##
##   1   2   3   4   5
## 191  81 239 161  73
```

```
samplesizestown <- c(table(boys$town))
```

Q2

Cluster sampling can be one-stage (we interview everyone in a cluster), or two-stage. We will concentrate on two-stage samples here, because one-stage cluster samples are rare in real life. One way to draw a cluster sample is to first draw clusters with equal probabilities. That is, we draw an SRS of clusters. For the Boys example dataset, we can for example select 2 out of 5 towns. In a second stage, we can then decide to draw the same number of sampling units within each of the 2 towns (eg. 50) , or we can draw sampling units ‘proportional to size’ in the cluster. E.g. if we draw town 2 and 4, but town 4 is twice the size of town 2, we draw 66 units in town 4 and 33 units from town 2.

Below is code for drawing a two-stage cluster sample. We first draw towns, and then in each town draw 50 cases.

```
boys$id <- 1:nrow(boys)
set.seed(123)

sample3 <- sample(5,2, prob = samplesizestown)
sample3
```

```
## [1] 3 2
```

```
# removing missing values in town
boys2 <- boys %>%
    filter(!is.na(town))%>%
    arrange(town)

clustersample1 <-
    boys2 %>%
    filter(town %in% sample3) %>%
    group_by(town) %>%
    filter(srswor(50, n()) == 1)
```

Now we have drawn a cluster sample, we need to know how to specify a cluster design in R. For this, we need to specify:

svydesign(id = ~clustervar + individual,fpc = ~nclusters + clustersize,data=dataset)

Please see the code for how to get these variables added to our cluster sample dataset below..

```r
# specify the fpc, which now exists of two components:
# the number of clusters, and the size within every
# sampled cluster
clustersample1$ncluster <- 5 # 5 towns in population

clustersample1$clustersize <- NA # empty first, then add sample sizes in towns 2 and 3
clustersample1$clustersize[clustersample1$town == 2] <- nrow(boys2[boys2$town == 2, ])
clustersample1$clustersize[clustersample1$town == 3] <- nrow(boys2[boys2$town == 3, ])

# or in a tidy way
# clustersample1 <-
#     boys2 %>%
#     group_by(town) %>%
#     summarize(clustersize = n()) %>%
#     right_join(clustersample1)

# and specify the cluster design
clusterdesign1 <- svydesign(id = ~town + id,
                            fpc = ~ncluster + clustersize,
                            data = clustersample1)
```

Q3

What is the bias in the clusterdesign if we do not specify the svydesign object at all? What is the design effect (add deff=T to your svymean())? How successful was our cluster sample in terms of the loss in precision?

Q4

You can either optimize your cluster-design based on costs or precision, or balance both. Imagine it costs 10 euros to conduct one interview (variable costs).

However, it also costs 500 euros to conduct interviews in one cluster (fixed costs per cluster).

An SRS of size 100 will in this case normally cost you "100 * 10 + 5 *500 = 3500 euros.

The cluster design specified in question 3 will cost you "100 * 10 + 2 * 500 = 2000 euros.

You now have to balance costs and precision. Assume that you have a fixed budget of 2500 euros. Doing an SRS is here not an option, because you would spend all your money on fixed costs. At the other extreme, you can sample in just 1 cluster and interview 400 cases.

Can you work out, using the variable wgt (weight) as your dependent variable, what would be the right number of clusters to draw?

(you may think that you would ideally like to stratify perhaps as well, either at the cluster-level or within the cluster (varying the sample size within each cluster). Indeed! That is what we will turn to next week though, for now, lets stick to clusters alone)

— END OF DOCUMENT —