Combinations of clustering and stratification

Peter Lugtig

15 december, 2021

Introduction

This week, we will discuss combinations of stratification and clustering. Lets first revisit stratification and clustering as covered last week by revisiting the final questions of those exercises.

Clustering (Q4 last week but with slightly different numbers).

Load the "boys.R" dataset. 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.

Question 1:

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

```
# I am sampling clusters PPS
table(boys$town)
samplesizestown <- c(table(boys$town))

boys$id <- 1:nrow(boys)
set.seed(367)

#sample1 <- sample(5,1, prob = samplesizestoum)
sample2 <- sample(5,2, prob = samplesizestown)
sample3 <- sample(5,3, prob = samplesizestown)
sample4 <- sample(5,4, prob = samplesizestown)

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

```
# draw the five cluster samples. Sample sizes within the clusters:
# 1 cluster: 200
# 2 cluster: 150 (each 75)
# 3 cluster: 100 (each 33)
# 4 cluster: 50 (each 13)

#clustersample1 <-
#    boys2 %>%
#    filter(town %in% sample1) %>%
#    group_by(town) %>%
#    filter(srswor(200, n()) == 1)%>%
#    mutate(ncluster=5)
# I drew cluster 3 here, but for every cluster,
    # the number of SSus will be too large!


clustersample2 <-
    boys2 %>%
    filter(town %in% sample2) %>%
    group_by(town) %>%
    filter(srswor(75, n()) == 1)%>%
    mutate(ncluster=5)
clustersample3 <-
    boys2 %>%
    filter(town %in% sample3) %>%
    group_by(town) %>%
    filter(srswor(66, n()) == 1)%>%
    mutate(ncluster=5)
clustersample4 <-
    boys2 %>%
    filter(town %in% sample4) %>%
    group_by(town) %>%
    filter(srswor(25, n()) == 1)%>%
    mutate(ncluster=5)

clustersample2 <-
     boys2 %>%
     group_by(town) %>%
     summarize(clustersize = n()) %>%
     right_join(clustersample2)
```

```
## Joining, by = "town"
```

```
clustersample3 <-
     boys2 %>%
     group_by(town) %>%
     summarize(clustersize = n()) %>%
     right_join(clustersample3)
```

```
## Joining, by = "town"
```

```
clustersample4 <-
     boys2 %>%
     group_by(town) %>%
     summarize(clustersize = n()) %>%
     right_join(clustersample4)
```

```
## Joining, by = "town"
```

```
# and specify the cluster design three times.

clusterdesign2 <- svydesign(id = ~town + id,
                            fpc = ~ncluster + clustersize,
                            data = clustersample2)
clusterdesign3 <- svydesign(id = ~town + id,
                            fpc = ~ncluster + clustersize,
                            data = clustersample3)
clusterdesign4 <- svydesign(id = ~town + id,
                            fpc = ~ncluster + clustersize,
                            data = clustersample4)
```

And now compute the mean in 'wgt' for the different cluster designs

Question 2

In practice, we may not want to draw clusters (PSUs) using an SRS, or Proportional to Size (PPS - as we have done so far), but we may want to stratify *before* drawing the clusters. In order to understand this: Imagine we are interested in doing a survey among ethnic minorities, which - at least in many countries - are to be found in the big cities. We therefore want to increase the probability of drawing cities in our sample.

Thinking about the 'wgt' variable - that we are lucky to know for our entire population - do you think it makes sense to stratify at the PSU level? And how is this for the variable 'hgt'?

Question 3

Assume (see question 1) that because we want to balance costs and precision, we want to sample 2 clusters. We have so far used sampling proportional to size for the cluster sizes. Can we stratify on 'wgt' in the first step when we draw PSUs. How would we do this? (you may take a sneak peak at .RMD file to see how I solved this using Neyman allocation at the PSU level)

Below you can find code where we adjust cluster probabilities based on neyman allocation criteria. Run the code, and work out line-by-line what happens.

Question 4:

what happens to the s.e. and precision compared to the earlier design where you drew 2 clusters?

```
set.seed(123)
sampleneyman <- sample(5,2, prob = neymanclusters$alloc)
clusterneyman <-
    boys2 %>%
    filter(town %in% sampleneyman) %>%
```

```
    group_by(town) %>%
    filter(srswor(75, n()) == 1)%>%
    mutate(ncluster=5)
# and specify the sample sizes within the clusters.
clusterneyman <-
     boys2 %>%
     group_by(town) %>%
     summarize(clustersize = n()) %>%
     right_join(clusterneyman)
```

```
## Joining, by = "town"
```

```
# we specify the svydesign object
clusterneyman <- svydesign(id = ~town + id,
                           fpc = ~ncluster + clustersize,
                           data = clusterneyman)
svymean(~wgt,na.rm=T, design=clusterneyman)
```

```
##       mean     SE
## wgt 41.428 3.6558
```

Finally, we can take a cluster sample like we have before, but then within every PSU, stratify the sample (possibly on a different variable) before taking a sample.

Question 5:

What about truly combining clustering (to save costs) with stratification (to optimize precision?): We can combine them. For example:

Take a clustersample proportional to size with 2 PSUs, and then stratify the sample on the variable 'agecat' using neyman allocation. We can do the stratification at two moments in our sampling procedure: 1. We can stratify using population data information (like we did last week), then draw the clusters (PPS), and draw the stratified sample within each of the clusters in the same way. 2. We can first draw the cluster sample, and then within each cluster figure out what would be a good way to stratify. This could lead to a design where we use different allocation proportions in each cluster, or even different variables! Such a design sounds strange, but it is commonly used in multi-country surveys, where in one country you for example have information on your sampling frame about age, and in the other country you don't.

For this exercise, use approach 1. Take the stratification design you specified last week for the variable 'agecat', implying you can sample 200 cases within the clusters in total. First draw 2 clusters, and then draw a stratified sample within every cluster. Note that in some cases you may run into the issue that there is only one PSU for a stratum (e.g. when you stratify on towns). In that case the option

```
options(survey.lonely.psu="adjust")
```

will tell R how to deal with this. The method "adjust" ensures that the stratum contribution to the variance is taken to be the average of all the strata with more than one PSU. This implies this PSU does not contribute to the variance. For more info, see http://r-survey.r-forge.r-project.org/survey/exmample-lonely.html

Question 6 (optional - no answer available):

If you feel like practicing more: You can try to see whether you can improve precision by stratifying within the clusters in a different way. Use the same two clusters. Figure out how you could stratify for each cluster separately. Compare your answer to question 4. What do you find?

Question 7 (optional)

Can you compute the inclusion probabilities for the design you specified in question 5, and use HT-estimation (use the probs= or weights= command rather than the fpc)?

– END of Document –