Sample Size Calculation in Single-stage Sampling

Richard Valliant, Jill A. Dever, and Frauke Kreuter

2020-07-28

A basic issue in sample design is how many units should be selected at each stage in order to efficiently estimate population values. If strata are used, the number of units to allocate to each stratum must be determined. In this vignette, we review some basic techniques for sample size determination in single-stage samples using the package `PracTools` (Valliant, Dever, and Kreuter 2019) that contains specialized routines to facilitate the calculations, most of which are not found in other packages. We briefly summarize some of selection methods and associated formulas used in designing single-stage samples and describe the capabilities of `PracTools`. Technical background is in Valliant, Dever, and Kreuter (2018), ch. 3. First, the package must be loaded with

```
library(PracTools)
```

Alternatively, `require(PracTools)` can be used.

Complex samples can involve any or all of stratification, clustering, multistage sampling, and sampling with varying probabilities. Many texts cover these topics, including Cochran (1977), Lohr (1999), Särndal, Swensson, and Wretman (1992), and Valliant, Dever, and Kreuter (2018). Here we discuss single-stage designs with and without stratification and formulas that are needed for determining sample allocations. The `PracTools` package will not select samples, but the R `sampling` package (Tillé and Matei 2016) will select almost all the types used in practice.

Simple Random Sampling

Simple random sampling without replacement (*srswor*) is a method of probability sampling in which all samples of a given size n have the same probability of selection. The function `sample` in R base (R Core Team 2020) will select simple random samples either with or without replacement. One way of determining an *srswor* sample size is to specify that a population value θ be estimated with a certain coefficient of variation (CV) which is defined as the ratio of the standard error of the estimator, $\hat{\theta}$, to the value of the parameter: $CV(\hat{\theta}) = \sqrt{Var(\hat{\theta})}/\theta$. For example, suppose that y_k is a value associated with element k, U denotes the set of all elements in the universe, N is the number of elements in the population, and the population parameter to be estimated is the mean, $\bar{y}_U = \sum_{k \in U} y_k/N$. With a simple random sample, this can be estimated by the sample mean, $\bar{y}_s = \sum_{k \in s} y_k/n$, where s is the set of sample elements and n is the sample size. Setting the required CV of $\bar{y}_s$ to some desired value CV_0 in an *srswor* leads to a sample size of

$$n = \frac{\frac{S_U^2}{\bar{y}_U^2}}{CV_0^2 + \frac{S_U^2}{N\bar{y}_U^2}}$$

The R function, `nCont`, will compute a sample size using either a target CV_0 or a target variance, V_0, of $\bar{y}_s$ as input. The parameters used by the function are shown below and are described in the help page for the function:

```
nCont(CV0=NULL, V0=NULL, S2=NULL, ybarU=NULL, N=Inf, CVpop=NULL)
```

- **Example 1: Sample size for a target CV.** Suppose that we estimate from a previous survey that the population CV of some variable is 2.0. If the population is extremely large and `CV0` (the target CV) is set to 0.05, then the call to the R function is `nCont(CV0 = 0.05, CVpop = 2)`. The resulting sample size is 1,600. If the population size is $N = 500$, then `nCont(CV0 = 0.05,CVpop = 2, N = 500)` results in a sample size of 381 after rounding. The finite population correction (*fpc*) factor has a substantial effect in the latter case.

The function `nProp` will perform the same computation for estimated proportions.

- **Example 2: Sample sizes for a vector of target CVs.** Often it will be useful to show a client the sample sizes for a series of precision targets. This will be especially true when the budget is uncertain and a researcher would like to think about options. We can ask for the sample sizes for a vector of values of `CV0` from 0.01 to 0.21 in increments of 0.02 with:

```
ceiling(nCont(CV0 = seq(0.01, 0.21, 0.02), CVpop=2))
[1] 40000  4445  1600   817   494   331   237   178   139   111    91
```

`ncont` returns unrounded sample sizes having quite a few decimal places; `ceiling` rounds to the next highest integer.

Using a margin of error to find sample sizes

Many investigators prefer to think of setting a tolerance for how close the estimate should be to the population value. If the tolerance, sometimes called the *margin of error* (MOE), is e and the goal is to be within e of the population mean with probability $1 - \alpha$, this translates to

$$Pr\left(|\bar{y}_s - \bar{y}_U| \leq e\right) = 1 - \alpha. \tag{1}$$

This is equivalent to setting the half-width of a $100\,(1-\alpha)$ two-sided confidence interval (CI) to $e = z_{1-\alpha/2}\sqrt{V\left(\bar{y}_s\right)}$, assuming that $\bar{y}_s$ can be treated as being normally distributed. The term $z_{1-\alpha/2}$ is the $100\,(1-\alpha/2)$ percentile of the standard normal distribution, i.e., the point with $1-\alpha/2$ of the area to its left. On the other hand, if we require

$$Pr\left(\left|\frac{\bar{y}_s - \bar{y}_U}{\bar{y}_U}\right| \leq e\right) = 1 - \alpha, \tag{2}$$

this corresponds to setting $e = z_{1-\alpha/2}\,CV\left(\bar{y}_s\right)$. If we set the MOE in (1) to e_0, then the above equation can be manipulated to give the required sample size as

$$n = \frac{z^2_{1-\alpha/2}\,S^2_U}{e_0^2 + z^2_{1-\alpha/2}\,S^2_U/N}. \tag{3}$$

Similarly, if the MOE in (2) is set to e_0, we obtain

$$n = \frac{z^2_{1-\alpha/2}\,S^2_U\,/\bar{y}^2_U}{e_0^2 + z^2_{1-\alpha/2}\,S^2_U/(N\bar{y}^2_U)} \tag{4}$$

The functions `nContMoe` and `nPropMoe` will make the sample size calculations based on MOEs for continuous variables and for proportions.

- **Example 3: Sample sizes for proportions based on an MOE.** Suppose that we want to estimate a proportion for a characteristic where an advance estimate is $p_U = 0.5$. The MOE is to be e when $\alpha = 0.05$. In other words, the sample should be large enough that a normal-approximation 95% confidence interval should be $0.50 \pm e$ as implied by (1). For example, if $e = 0.03$ and the estimated proportion were actually 0.5, we want the confidence interval to be $0.50 \pm 0.03 = [0.47, 0.53]$. The sample size is highly dependent on the width of the confidence interval as seen in the following table. Sample sizes are evaluated using the formula given in (3) with $S^2_U = Np_U(1-p_U)/(N-1)$, $p_U = 0.5$ and $z_{0.975} = 1.96$. The command to generate the sample sizes listed in the table below is

```
    ceiling(nPropMoe(moe.sw=1, e=seq(0.01,0.08,0.01), alpha=0.05, pU=0.5))
#> [1] 9604 2401 1068  601  385  267  196  151
```

e	n	e	n
0.01	9,604	0.05	385
0.02	2,401	0.06	267
0.03	1,068	0.07	196
0.04	601	0.08	151

The parameter `moe.sw=1` says to compute the sample size based on (3). `moe.sw=2` would use the MOE relative to $\bar{y}_U$ in (4).

Stratified Simple Random Sampling

Simple random samples are rare in practice for several reasons. Most surveys have multiple variables and domains for which estimates are desired. Selecting a simple random sample runs the risk that one or more important domains will be poorly represented or omitted entirely. In addition, variances of survey estimates often can be reduced by using a design that is not *srswor*.

A design that remedies some of the problems noted for an *srswor* is referred to as stratified simple random sampling (without replacement) or *stsrswor*. As the name indicates, an *srswor* design is administered within each design stratum. Strata are defined with one or more variables known for *all* units and partition the entire population into mutually exclusive groups of units. We might, for example, divide a population of business establishments into retail trade, wholesale trade, services, manufacturing, and other sectors. A household population could be divided into geographic regions—north, south, east, and west. For an *stsrswor*, we define the following terms:

N_h = the known number of units in the population in stratum h $(h = 1, 2, \ldots, H)$

n_h = the size of the *srswor* selected in stratum h

y_{hi} = the value of the y variable for unit i in stratum h

$S^2_{Uh} = \sum_{i=1}^{N_h} \left(y_{hi} - \bar{y}_{U_h}\right)^2 / (N_h - 1)$, the population variance in stratum h

U_h = set of all units in the population from stratum h

s_h = set of n_h sample units from stratum h

c_h = cost per sample unit in stratum h

The population mean of y is $\bar{y}_U = \sum_{h=1}^{H} W_h \bar{y}_{U_h}$, where $W_h = N_h / N$ and $\bar{y}_{U_h}$ is the population mean in stratum h. The sample estimator of $\bar{y}_U$ based on an *stsrswor* is

$$\bar{y}_{st} = \sum_{h=1}^{H} W_h \bar{y}_{s_h},$$

where $\bar{y}_{s_h} = \sum_{i \in s_h} y_{hi} / n_h$. The population sampling variance of the stratified estimator of the mean is

$$Var\left(\bar{y}_{st}\right) = \sum_{h=1}^{H} W_h^2 \frac{1 - f_h}{n_h} S^2_{Uh},$$

where $f_h = n_h / N_h$. The total cost of the sample is

$$C = \sum_{h=1}^{H} c_h n_h.$$

There are various ways of allocating the sample to the strata, including:

1. Proportional to the N_h population counts
2. Equal allocation (all n_h the same)
3. Cost-constrained optimal in which the allocation minimizes the variance of $\bar{y}_{st}$ subject to a fixed budget
4. Variance-constrained optimal in which the allocation minimizes the total cost subject to a fixed variance target for $\bar{y}_{st}$
5. Neyman allocation, which minimizes the variance of the estimated mean disregarding the c_h costs

The R function, `strAlloc`, will compute the proportional, Neyman, cost-constrained, and variance-constrained allocations. The parameters accepted by the function are shown below.

```
n.tot = fixed total sample size
Nh = vector of pop stratum sizes or pop stratum proportions (required parameter)
Sh = stratum unit standard deviations, required unless alloc = "prop"
cost = total variable cost
ch = vector of costs per unit in strata
V0 = fixed variance target for estimated mean
CV0 = fixed CV target for estimated mean
ybarU = pop mean of y
alloc = type of allocation, must be one of "prop", "neyman", "totcost", "totvar"
```

If the stratum standard deviations are unknown (as would usually be the case), estimates can be used.

The parameters can only be used in certain combinations, which are checked at the beginning of the function. Basically, given an allocation, only the parameters required for the allocation are allowed and no more. For example, the Neyman allocation requires `Nh`, `Sh`, and `n.tot`. The function returns a list with three components—the allocation type, the vector of sample sizes, and the vector of sample proportions allocated to each stratum. Three examples of allocations are Neyman, cost constrained, and variance constrained (via a target CV):

```
# Neyman allocation
Nh <- c(215, 65, 252, 50, 149, 144)
Sh <- c(26787207, 10645109, 6909676, 11085034, 9817762, 44553355)
strAlloc(n.tot = 100, Nh = Nh, Sh = Sh, alloc = "neyman")
#>
#>
#>
#>                        allocation = neyman
#>                                Nh = 215, 65, 252, 50, 149, 144
#>                                Sh = 26787207, 10645109, 6909676, 11085034, 9817762, 44553355
#>                                nh = 34.641683, 4.161947, 10.473487, 3.333804, 8.798970, 38.590108
#>                             nh/n = 0.34641683, 0.04161947, 0.10473487, 0.03333804, 0.08798970, 0.385
#> anticipated SE of estimated mean = 1727173

# cost constrained allocation
ch <- c(1400, 200, 300, 600, 450, 1000)
strAlloc(Nh = Nh, Sh = Sh, cost = 100000, ch = ch, alloc = "totcost")
#>
#>
#>
#>                        allocation = totcost
#>                                Nh = 215, 65, 252, 50, 149, 144
#>                                Sh = 26787207, 10645109, 6909676, 11085034, 9817762, 44553355
#>                                nh = 30.605403, 9.728474, 19.989127, 4.499121, 13.711619, 40.340301
```

```
#>                              nh/n = 0.2574608, 0.0818385, 0.1681538, 0.0378478, 0.1153458, 0.3393533
#> anticipated SE of estimated mean = 1636053

# allocation with CV target of 0.05
strAlloc(Nh = Nh, Sh = Sh, CV0 = 0.05, ch = ch, ybarU = 11664181, alloc = "totvar")
#>
#>
#>
#>                        allocation = totvar
#>                                Nh = 215, 65, 252, 50, 149, 144
#>                                Sh = 26787207, 10645109, 6909676, 11085034, 9817762, 44553355
#>                                nh = 104.54922, 33.23283, 68.28362, 15.36917, 46.83941, 137.80400
#>                             nh/n = 0.2574608, 0.0818385, 0.1681538, 0.0378478, 0.1153458, 0.3393533
#> anticipated SE of estimated mean = 583209.1
```

The output of `strAlloc` is a list with components: `allocation` (the type of allocation), `Nh`, `Sh`, `nh`, `nh/n`, and `anticipated SE of estimated mean`. If the results are assigned to an object, e.g., `neyman <- strAlloc(n.tot = 100, Nh = Nh, Sh = Sh, alloc = "neyman")`, the components in the list can be accessed with syntax like `neyman$nh`.

There are many variations on how to allocate a sample to strata. In most practical applications, there are multiple variables for which estimates are needed. This complicates the allocation problem because each variable may have a different optimal allocation. This type of multicriteria problem can be solved using mathematical programming as discussed in Valliant, Dever, and Kreuter (2018), ch. 5.

Probability Proportional to Size Sampling

Probability proportional to size (*pps*), single-stage sampling is used in situations where an auxiliary variable (i.e., a covariate) is available on the frame that is related to the variable(s) to be collected in a survey. For example, the number of employees in a business establishment one year ago is probably related to the number of employees the establishment has in the current time period.

The variance formula for an estimated mean in a *pps* sample selected without replacement is too complex to be useful in determining a sample size. Thus, a standard workaround is to use the with-replacement (*ppswr*) variance formula to calculate a sample size. The result may be somewhat larger than needed to hit a precision target, but if the sample is reduced by nonresponse, beginning with a larger sample is prudent anyway. The simplest estimator of the mean that is usually studied with *ppswr* sampling is called "*p*-expanded with replacement" or *pwr* (see Särndal, Swensson, and Wretman (1992), ch.2) and is defined as

$$\hat{\bar{y}}_{pwr} = \frac{1}{Nn}\sum_{i\in s}\frac{y_i}{p_i}$$

where p_i is the probability that unit i would be selected in a sample of size 1. The variance of $\hat{\bar{y}}_{pwr}$ in *ppswr* sampling is

$$Var\left(\hat{\bar{y}}_{pwr}\right) = \frac{1}{N^2 n}\sum_{U} p_i\left(\frac{y_i}{p_i} - t_U\right)^2 \equiv \frac{V_1}{N^2 n} \tag{5}$$

where t_U is the population total of y.

If the desired coefficient of variation is CV_0, (5) can be solved to give the sample size as

$$n = \frac{V_1}{N^2}\frac{1}{\bar{y}_U^2 CV_0^2}\ . \tag{6}$$

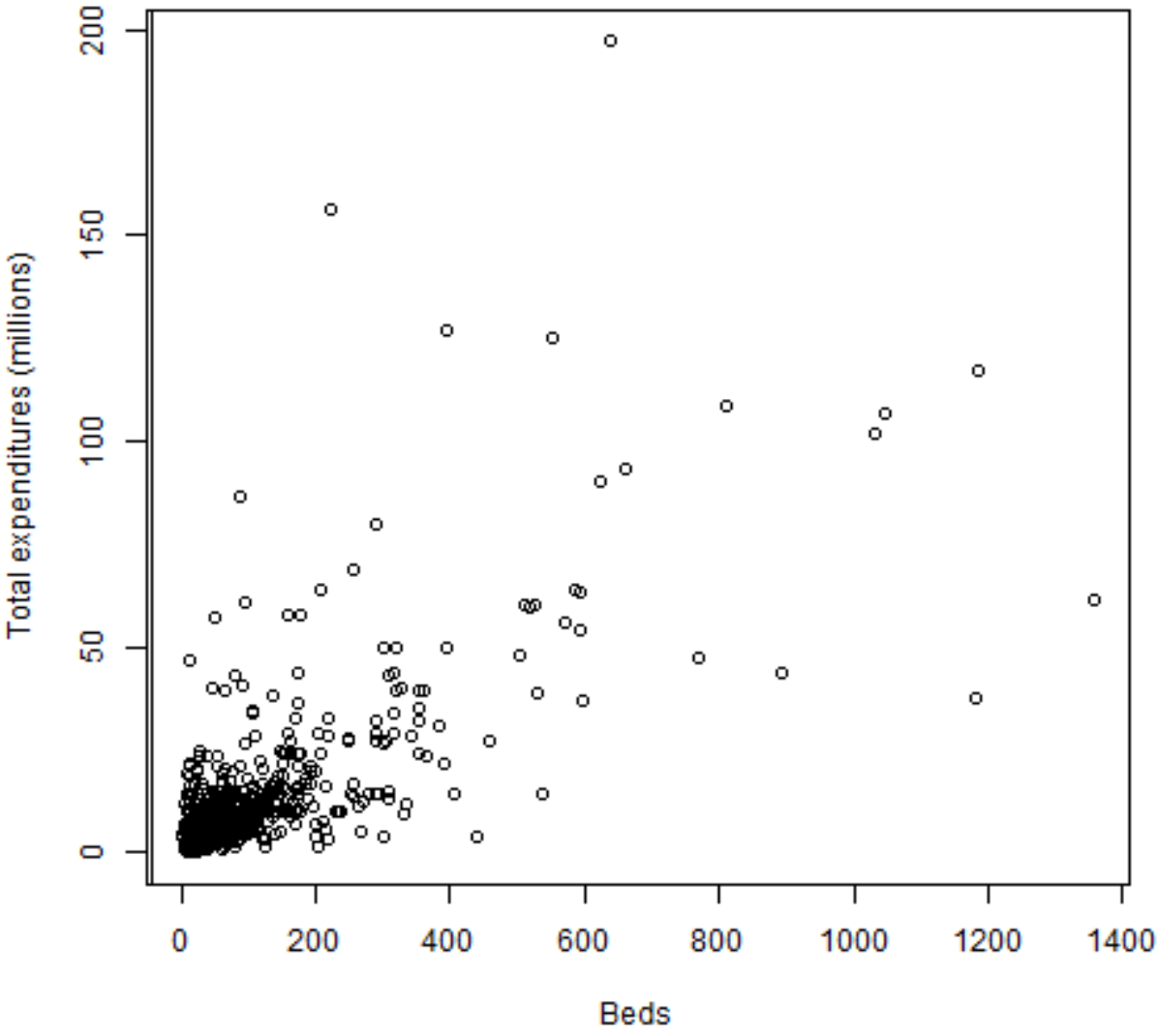

Figure 1: Figure 1. Plot of expenditures vs. beds in hospital population

- **Example 4: Sample size in a *pps* sample.** We use `smho.N874`, which is one of the example populations in the `PracTools` package, to illustrate a *pps* sample size calculation. Figure 1 plots annual expenditures per hospital versus number of beds for the 670 hospitals that have inpatient beds. Although the relationship is fairly diffuse, the correlation of beds and expenditures is 0.70 so that *pps* sampling with beds as a measure of size could be efficient. The code below evaluates (6) giving a *pps* sample of $n = 57$, which will produce an anticipated CV of 0.149. In contrast, an *srs* of $n = 82$ would be necessary to obtain the same size CV.

```
require(PracTools)
data("smho.N874")

y <- smho.N874[,"EXPTOTAL"]
x <- smho.N874[, "BEDS"]
y <- y[x>0]
x <- x[x>0]
ybarU <- mean(y)

(N <- length(x))
#> [1] 670
CV0 <- 0.15

  # calculate V1 based on pp(x) sample
pik <- x/sum(x)
T <- sum(y)
(V1 <- sum( pik*(y/pik - T)^2))
#> [1] 9.53703e+19

n <- V1 / (N*ybarU*CV0)^2
(n <- ceiling(n))
#> [1] 57

  # Anticipated SE for the pps sample
(cv.pps <- sqrt(V1/(N^2*n)) / ybarU)
#> [1] 0.1495183

  # sample size for an srs to produce the same SE
ceiling(nCont(CV0 = cv.pps, S2 = var(y), ybarU = ybarU, N = N))
#> [1] 82
```

The `PracTools` package includes a variety of other functions relevant to the design of single-stage samples that are not discussed in this vignette:

Function	Description
`gammaFit`	Iteratively computes estimate of γ in a model with $E_M(y) = \mathbf{x}^T\boldsymbol{\beta}$ and $\sigma^2\mathbf{x}^\gamma$. This is useful in determining a measure of size for *pps* sampling.
`nDep2sam`	Compute a simple random sample size for estimating the difference in means when samples overlap
`nDomain`	Compute a simple random sample size using either a target coefficient of variation or target variance for an estimated mean or total for a domain
`nLogOdds`	Calculate the simple random sample size for estimating a proportion using the log-odds transformation

Function	Description
`nProp`	Compute the simple random sample size for estimating a proportion based on different precision requirements
`nProp2sam`	Compute a simple random sample size for estimating the difference in proportions when samples overlap
`nWilson`	Calculate a simple random sample size for estimating a proportion using the Wilson method

References

Cochran, W. G. 1977. *Sampling Techniques*. New York: John Wiley & Sons, Inc.

Lohr, S. L. 1999. *Sampling: Design and Analysis*. Pacific Grove CA: Duxbury Press.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Särndal, C.-E., B. Swensson, and J. Wretman. 1992. *Model Assisted Survey Sampling*. New York: Springer-Verlag.

Tillé, Y., and A. Matei. 2016. *sampling: Survey Sampling, Version 2.8*. http://CRAN.R-project.org/package=sampling.

Valliant, R., J. A. Dever, and F. Kreuter. 2018. *Practical Tools for Designing and Weighting Survey Samples*. 2nd ed. New York: Springer-Verlag.

Valliant, Richard, Jill A. Dever, and Frauke Kreuter. 2019. *PracTools: Tools for Designing and Weighting Survey Samples, Version 1.2.1*. https://CRAN.R-project.org/package=PracTools.