Sampling Populations

practical examples of sampling populations in veterinary medicine using R

NCSU: Nicolas Cardenas & Gustavo Machado

2022-11-01

Set some packages and data in R.

data is avaliable in

https://github.com/machado-lab/sample_size_practice

```
#load and install the packages (if required)
if(!require(tidyverse)){install.packages("tidyverse")}; library(tidyverse)
if(!require(epiR)){install.packages("epiR")}; library(epiR)
if(!require(readxl)){install.packages("readxl")}; library(readxl)
if(!require(sampler)){install.packages("sampler")}; library(sampler)
if(!require(readr)){install.packages("readr")}; library(readr)
if(!require(plot3D)){install.packages("plot3D")}; library(plot3D)
if(!require(sampling)){install.packages("sampling")}; library(sampling)

# load the example database
herds <- read_excel("~/repos/scc_40.xlsx")
# set the data for use here!
herds <- herds %>% distinct(cowid, .keep_all = T)
# round  herd size
herds$h_size <- ceiling(herds$h_size)
```

Why of Sample Size Calculations

- How many animals/subjects do I need for my experiment?
  - Too small of a sample size can under detect the effect of interest in your experiment.
  - Too large of a sample size may lead to unnecessary wasting of resources and animals Like Goldilocks, we want our sample size to be 'just right'.
- The answer: Goal: Sample Size Calculation.

Sample size and the expected proportion i.e.

Here we will calculate the number of animals needed to estimate disease prevalence in a finite population. For this example, the expected prevalence is **15%**. We want to know how the sample size in which a **95%** confidence interval is needed. We know that our total target population is `N` of 1000 animals.

```
size <- rsampcalc(N=1000,    # total of the population
                  e=5,       # tolerable margin of error this case 5%
                  ci=95,     # confidence interval of 95%
                  p=0.15)    # expected prevalence
print(size)
```

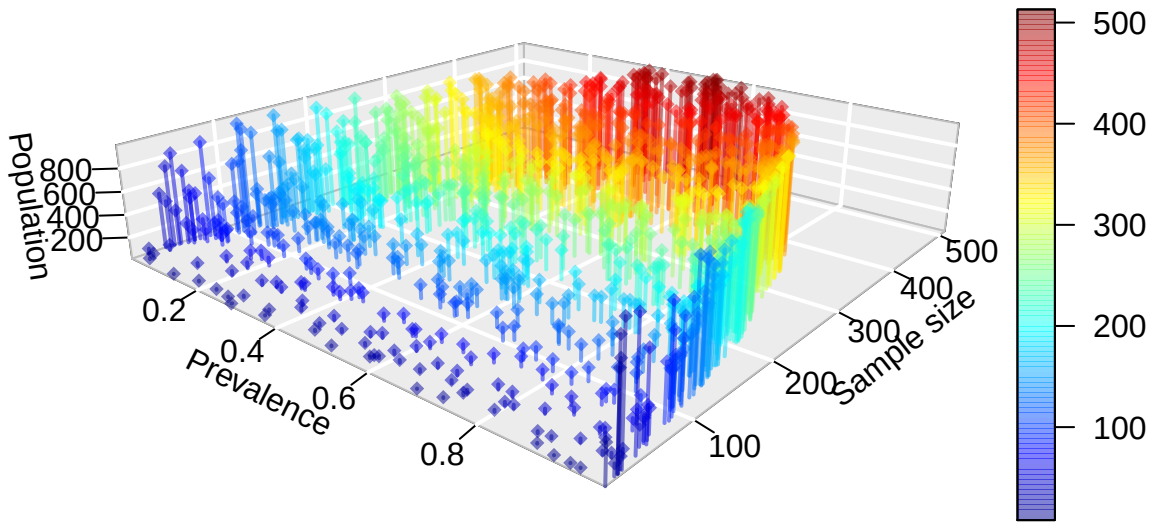
```
## [1] 164
```

Relationship between population size, proportion, and sample size

In the next analysis, we are going to simulate 1000 sampling designs. Here, we are going to consider an initial prevalence of 1% while increasing the expected prevalence all the way to 100%, in this same vein, let's consider that population size from 10 animals to 1000 animals. In the next plot, is elucidated the effect of all these factorial combinations. the color reflects the sample size whereas warm colors represent larger sample sizes.

```
# Set up the number of samples.
N <- 1000
myprevalence <- runif(n=N, min=0.01, max=1) # prevalence between 1 and 100%.
mypopulation<- runif(n=N, min=10, max=1000) # population within 10 and 1000.

mysamplesize <- c()
for (i in 1:length(mypopulation)){
  aux <- rsampcalc(mypopulation[i],
                   e=3, ci=95,
                   p=myprevalence[i])
  mysamplesize <- rbind(mysamplesize, aux)
}
# Plot the results.
mydata <- tibble(myprevalence, mypopulation, mysamplesize= as.numeric(mysamplesize))
scatter3D(mydata$myprevalence,
          mydata$mysamplesize,
          mydata$mypopulation,
          bty = "g", pch = 18,
          lwd = 2, alpha = 0.5,
          expand =0.2,
          phi = 20,
          colvar = mysamplesize,
          ticktype = "detailed",
          type = "h",
          xlab = "Prevalence", ylab = "Sample size", zlab = "Population")
```



Simple random sampling.

We will take a sample of **N** from a list of target invidious:

- I wanted to sample from a list of friend and calculate who will provide me with my daily cookie?

My options are the following: Felipe, Arthur, Denilson, Jason, Abby, Gustavo and Kelsey.

I can run one time random sample selection once.

```
# Options to get my cookie
labmates <- c("Felipe",
       "Jason",
       "Arthur",
       "Denilson",
        "Abby",
       "Gustavo",
       "Kelsey" )

# Calculate the sample
sample(labmates,              # total of the population
       1,              # one person
       replace = F)    # sampling without replacement
```

```
## [1] "Felipe"
```

To better represent the sample drawing we will simulate this sampling for 1000 times

```
# Simple random sample repeated 1000 times.
result <- sample(labmates,       # total of the population
          1000,           # one person
          replace = T)    # sample without replacement

# Present into a table with the results.
sort(table(result))
```

```
## result
##  Gustavo    Jason Denilson   Kelsey   Felipe   Arthur      Abby
##      129      136      138      140      146      155      156
```

Herds data as an example of a dataset

In the next examples, we will use **herds** dataframe which is a subset of a large mastitis dataset collected by Jens Agger and the Danish Cattle Organization. This dataset contains records from 14,357 test-day observations in 2,178 cows from 40 herds. Milk weights (production records) were collected approximately monthly, and only records from a single lactation for each cow were included in this dataset. Factors that may have affected the somatic cell count (SCC) were also recorded. The major objective of this study was to determine if the relationship between the somatic cell count and milk production varies for cows with different characteristics (age, breed, grazing or not etc).

variables description

variable	Description	Codes/units
herdid	herd id	
cowid	cow id	
test	the approximate month of lactation	0 to 10
h_size	average herd size	
c_heifer	parity of the cow	1 = heifer 0 = multiparous
t_season	season of test day	l = jan-mar 2 = apr-jun 3 = jul-sep 4 = oct-dec
t_dim	days in milk on test-day	days
t_lnscc	log somatic cell count on test day	

1. A complete list of the population to be sampled is not required.
2. The sampling interval is computed as population size divided by the required sample size.

Let's calculate the sample size first

```r
#get the total of animals in the population
N_total_cows <- nrow(herds)                # Total of animal in the population

# Calculate sample size.
my_sample_size <- rsampcalc(N_total_cows,  # Total number of records
                  e=5,                     # Tolerable margin of error this case 3%
                  ci=95,                   # Confidence interval of 95%
                  p=0.5)                   # Anticipated response distribution

print(my_sample_size)
```

```
## [1] 327
```

Thus, we have to sample 327 cows from the herd's population.

Systematic random sample

```r
my_selected_cows <- ceiling(seq(from= 1,           # Initial number of cows
                    to = nrow(herds),              # Total number of cows
                    length.out = my_sample_size))  # my sample size

# select the cows sampled
my_sampled_cows <- herds %>%
  filter(cowid %in% my_selected_cows)
head(my_sampled_cows)
```

```
## # A tibble: 6 x 10
##   herdid cowid  test   obs h_size c_heifer t_season t_dim t_lnscc t_ecm
##    <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl>    <dbl> <dbl>   <dbl> <dbl>
## 1      1     1     0     1     45        0        2    21    6.57  23.1
## 2      1     8     3    42     45        0        3   110    6.35  17.1
## 3      1    15     4    94     45        0        2   134    5.39  18.3
## 4      1    22     3   137     45        0        3   102    2.30  16.7
## 5      1    28     1   171     45        0        2    42    3.40  16.8
## 6      1    35     0   212     45        1        2    29    2.30  17.8
```

Stratified random sample

suppose that im interesting in taking a sample that represents well all the different samples size because that may impact the analysis of my study.

let's see how is my strata

```r
# table the cows by strata
herds %>%
  group_by(h_size)%>%
  count() %>%
  arrange(-h_size)
```

```
## # A tibble: 29 x 2
## # Groups:   h_size [29]
##    h_size     n
##     <dbl> <int>
##  1    102    21
##  2     84   190
##  3     77    77
##  4     76    97
##  5     63    71
##  6     61    78
##  7     57    72
##  8     56    74
##  9     51    77
## 10     50   214
## # ... with 19 more rows
```

We found 29 strata in my data, no here we will create a subset of `herds` database by using a stratified random sample approach

```
# Stratify the data by the variable 'h_size' which represents the herd size classification
stratifiedsample <- ssamp(df = herds,           # database to sample
                          n = my_sample_size,   # sample size
                          strata = h_size)      #strata to be used
# check the sampled data
stratifiedsample %>%           # my final database after the stratified random sampling
  group_by(h_size) %>%         # group by the strata variable
  count() %>%                  # count the item by strata
  arrange(-h_size)             # sort in decreasing order
```

```
## # A tibble: 29 x 2
## # Groups:   h_size [29]
##    h_size     n
##     <dbl> <int>
##  1    102     3
##  2     84    29
##  3     77    12
##  4     76    15
##  5     63    11
##  6     61    12
##  7     57    11
##  8     56    11
##  9     51    12
## 10     50    32
## # ... with 19 more rows
```

Cluster sampling

1. A cluster is a natural or convenient collection of study subjects with one or more characteristics in common
    - a dairy herd is a cluster of cattle.
2. Cluster sampling is done because it might be easier to get a list of clusters (farms) than it would be to get a list of individuals (calves).

lets select the farms id as clusters

```
# create my clusters list
clusters <- unique(herds$herdid)
```

select a sample size of the clusters

```
# Calculate the sample size of the clusters
my_sample_size_cluster <- rsampcalc(length(clusters),  # Total number of clusters
                  e=5,                          # Tolerable margin of error this case 3%
                  ci=95,                        # Confidence interval of 95%
                  p=0.15)                        # Anticipated response distribution
my_sample_size_cluster
```

```
## [1] 34
```

here, given the small N of our sample, we take a sample of 34

Filtering the herd's data by the clusters selected

```
# create a sample of the clusters (herds ids)
herds_to_be_sampled <-  sample(herds$herdid,
                               my_sample_size_cluster)

herds_sampled <- herds %>%
  filter(herdid %in% herds_to_be_sampled)

head(herds_sampled)
```

```
## # A tibble: 6 x 10
##   herdid cowid  test   obs h_size c_heifer t_season t_dim t_lnscc t_ecm
##    <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl>    <dbl> <dbl>   <dbl> <dbl>
## 1      2    69     1   416     57        1        2    53    5.35  30.3
## 2      2    70     0   423     57        1        4    13    3.69  22.2
## 3      2    71     1   429     57        0        3    41    6.76  29.3
## 4      2    72     1   436     57        0        2    45    4.79  29.1
## 5      2    73     0   444     57        0        2    11    5.94  32.4
## 6      2    74     0   449     57        0        2    20    6.70  21.9
```

Multi-stage sampling

Here we will generate artificial data (a 235X3 matrix with 3 columns: state, region, and income).

- The variable "`state`" has 2 categories ('***Triangle***','***No__triangle***').
- The variable "`region`" has 5 categories ('Cary', 'Raleigh', 'Durham', 'Ashville', 'Carolina').

- The variable "`income`" is generated using the U(0,1) distribution.

```
# create a random data
data<- rbind(matrix(rep('Triangle',165),165,1,byrow=TRUE),
             matrix(rep('No_triangle',70),70,1,byrow=TRUE))

data <- cbind.data.frame(data,c(rep('Cary',115),
                                rep('Raleigh',10),
                                rep('Durham',40),
                                rep('Ashville',30),
                                rep('Carolina beach',40)),
                         100*runif(235))


names(data)=c("state","region","income")
```

The method is simple random sampling without replacement where # 25 units are drawn in the first-stage and in the second-stage, 10 units are drawn from the already 25 selected units

```
m=mstage(data,
         size=list(25,  # first strata a sample of 25
                   10), # strata
         method=list("srswor","srswor")) # stands for simple random sampling without replacement
```

The first stage is `m[[1]]`, the second stage is `m[[2]]`

```
# extracts the observed data
xx=getdata(data,m)[[2]]

# check the result
table(xx$state,xx$region)
```

```
##
##               Ashville Carolina beach Cary Durham
##   No_triangle        2              1    0      0
##   Triangle           0              0    5      2
```

Assignment

You will need to take a sample to calculate the prevalence of mastitis in cows by using a cell count. Column `t_lnscc` in the `herds` dataset indicates the cell count results. Please consider an estimated prevalence of 15% and a tolerable margin of error of 5%.