Experiment-3 Ransom Sampling

siju.swamy@saintgits.org

08/07/2021

Random samples How to generate random numbers. Study how to select a random sample with replacement from normal and uniform distribution. Students can use the built in functions to explore random sample selection.

Sampling from a Dataset

Researchers are often interested in answering questions about populations like:

1.is the average height of a certain species of plant?

2. What is the average weight of a certain species of bird?
3. What percentage of citizens in a certain city support a certain law?

One way to answer these questions is to go around and collect data on every single individual in the population of interest.

However, this is typically too costly and time-consuming which is why researchers instead take a sample of the population and use the data from the sample to draw conclusions about the population as a whole.

How to Select Random Samples in R

To select a random sample in R we can use the sample() function, which uses the following syntax:

`sample(x, size, replace = FALSE, prob = NULL)`

where:

x: A vector of elements from which to choose.

size: Sample size.

replace: Whether to sample with replacement or not. Default is FALSE.

prob: Vector of probability weights for obtaining elements from vector. Default is NULL.

This tutorial explains how to use this function to select a random sample in R from both a vector and a data frame.

Example 1: Random Sample from a Vector

The following code shows how to select a random sample from a vector without replacement:

```r
data <- c(1, 3, 5, 6, 7, 8, 10, 11, 12, 14)

#select random sample of 5 elements without replacement
sample(x=data, size=5)
```

```
## [1]  5 11 10  7  1
```

Example 2: Random Sample from a Data Frame

```
#create vector of data
data <- c(1, 3, 5, 6, 7, 8, 10, 11, 12, 14)

#select random sample of 5 elements with replacement
sample(x=data, size=5, replace=TRUE)
```

```
## [1]  6  3 12  5 11
```

The following code shows how to select a random sample from a data frame:

```
#create data frame
df <- data.frame(x=c(3, 5, 6, 6, 8, 12, 14),
                 y=c(12, 6, 4, 23, 25, 8, 9),
                 z=c(2, 7, 8, 8, 15, 17, 29))

#view data frame
df
```

```
##    x  y  z
## 1  3 12  2
## 2  5  6  7
## 3  6  4  8
## 4  6 23  8
## 5  8 25 15
## 6 12  8 17
## 7 14  9 29
```

```
#select random sample of three rows from data frame
rand_df <- df[sample(nrow(df), size=3), ]

#display randomly selected rows
rand_df
```

```
##    x  y  z
## 7 14  9 29
## 1  3 12  2
## 5  8 25 15
```

Here's what's happening in this bit of code:

1. To select a subset of a data frame in R, we use the following syntax: df[rows, columns]
2. In the code above, we randomly select a sample of 3 rows from the data frame and all columns.
3. The end result is a subset of the data frame with 3 randomly selected rows.

It's important to note that each time we use the `sample()` function, R will select a different sample since the function chooses values randomly.

In order to replicate the results of some analysis, be sure to use set.seed(some number) so that the `sample()` function chooses the same random sample each time. For example:

```
#make this example reproducible
set.seed(23)

#create data frame
df <- data.frame(x=c(3, 5, 6, 6, 8, 12, 14),
                 y=c(12, 6, 4, 23, 25, 8, 9),
                 z=c(2, 7, 8, 8, 15, 17, 29))

#select random sample of three rows from data frame
rand_df <- df[sample(nrow(df), size=3), ]

#display randomly selected rows
rand_df
```

```
##    x  y  z
## 5  8 25 15
## 2  5  6  7
## 6 12  8 17
```

Sampling from a normal distribution

```
pop=rnorm(100,0,1)
sample1=sample(pop,10,replace=TRUE)
sample1
```

```
##  [1]  2.03858973  2.01658182 -0.46035791  0.74365197  0.51821280
##  [6] -1.38649383  1.46926146  0.02151535  1.34136740 -1.02235325
```

Sampling from uniform distribution

```
pop2=runif(100,-1,1)
sample2=sample(pop2,10,replace=TRUE)
sample2
```

```
##  [1]  0.4479859 -0.6324760  0.5122819 -0.8579948  0.2906329  0.1552436
##  [7] -0.4428663 -0.6600549  0.5187372 -0.4428663
```

Stratified Sampling in R

Researchers often take samples from a population and use the data from the sample to draw conclusions about the population as a whole.

One commonly used sampling method is stratified random sampling, in which a population is split into groups and a certain number of members from each group are randomly selected to be included in the sample.

This tutorial explains how to perform stratified random sampling in R.

Example: Stratified Sampling in R A high school is composed of 400 students who are either Freshman, Sophomores, Juniors, or Seniors. Suppose we'd like to take a stratified sample of 40 students such that 10 students from each grade are included in the sample.

The following code shows how to generate a sample data frame of 400 students:

```
#make this example reproducible
set.seed(1)

#create data frame
df <- data.frame(grade = rep(c('Freshman', 'Sophomore', 'Junior', 'Senior'), each=100),
                 gpa = rnorm(400, mean=85, sd=3))

#view first six rows of data frame
head(df)
```

```
##      grade      gpa
## 1 Freshman 83.12064
## 2 Freshman 85.55093
## 3 Freshman 82.49311
## 4 Freshman 89.78584
## 5 Freshman 85.98852
## 6 Freshman 82.53859
```

Stratified Sampling Using Number of Rows

The following code shows how to use the `group_by()` and `sample_n()` functions from the `dplyr` package to obtain a stratified random sample of 40 total students with 10 students from each grade:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#obtain stratified sample
strat_sample <- df %>%
                  group_by(grade) %>%
                  sample_n(size=10)

#find frequency of students from each grade
table(strat_sample$grade)
```

```
##
##  Freshman    Junior    Senior Sophomore
##        10        10        10        10
```

Stratified Sampling Using Fraction of Rows

The following code shows how to use the `group_by()` and `sample_frac()` functions from the dplyr package to obtain a stratified random sample in which we randomly select 15% of students from each grade:

```
library(dplyr)

#obtain stratified sample
strat_sample <- df %>%
                  group_by(grade) %>%
                  sample_frac(size=.15)

#find frequency of students from each grade
table(strat_sample$grade)
```

```
##
##  Freshman    Junior    Senior Sophomore
##        15        15        15        15
```