Simple random and systematic sampling

SAMPLING IN R



Richie Cotton
Data Evangelist at DataCamp

datacamp

Simple random sampling





Simple random sampling of coffees





Simple random sampling in R

```
set.seed(19000113)
coffee_ratings %>%
  slice_sample(n = 5)
```

```
  total_cup_points variety country_of_origin aroma flavor aftertaste body balance
1            81.00     SL14            Uganda  7.33   6.92       7.17 7.42    7.42
2            85.00 Caturra          Colombia  8.00   7.92       7.75 7.75    7.83
3            85.25 Bourbon         Guatemala  8.00   7.92       7.75 7.92    7.83
4            81.42  Catuai         Guatemala  7.42   7.33       7.08 7.33    7.25
5            82.75 Caturra          Honduras  7.58   7.50       7.42 7.50    7.50
```

Systematic sampling





Adding a row ID column

```
library(tibble)
coffee_ratings <- coffee_ratings %>%
  rowid_to_column()
```

```
# A tibble: 1,338 x 9
   rowid total_cup_points variety country_of_origin aroma flavor aftertaste  body balance
   <int>            <dbl> <chr>   <chr>              <dbl>  <dbl>      <dbl> <dbl>   <dbl>
 1     1             90.6 NA      Ethiopia            8.67   8.83       8.67  8.5     8.42
 2     2             89.9 Other   Ethiopia            8.75   8.67       8.5   8.42    8.42
 3     3             89.8 Bourbon Guatemala           8.42   8.5        8.42  8.33    8.42
 4     4             89   NA      Ethiopia            8.17   8.58       8.42  8.5     8.25
 5     5             88.8 Other   Ethiopia            8.25   8.5        8.25  8.42    8.33
...
```

Systematic sampling in R

```
sample_size <- 5
pop_size <- nrow(coffee_ratings)
```

```
1338
```

```
interval <- pop_size %/% sample_size
```

```
267
```

Systematic sampling in R 2

```
row_indexes <- seq_len(sample_size) * interval
```

```
267  534  801 1068 1335
```

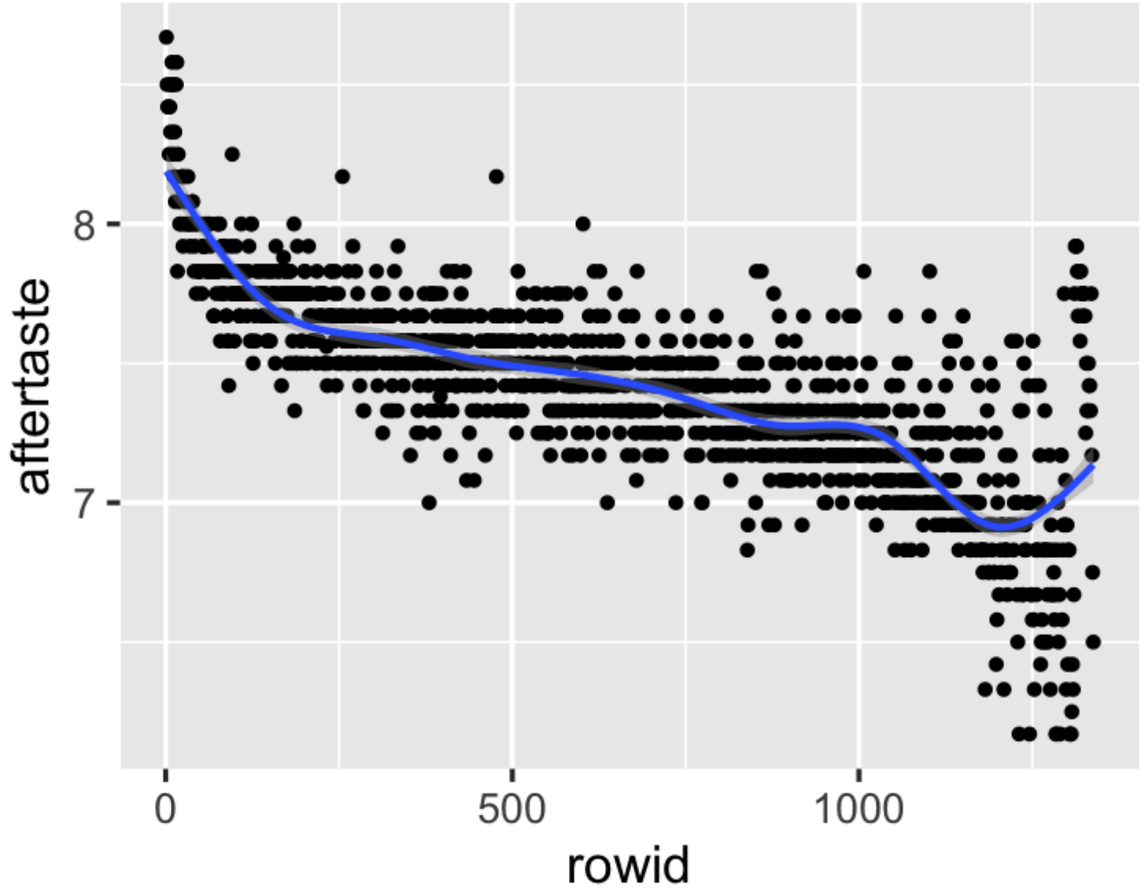
```
coffee_ratings %>%
  slice(row_indexes)
```

```
 # A tibble: 5 x 9
  rowid total_cup_points variety country_of_origin aroma flavor aftertaste  body balance
  <int>            <dbl> <chr>   <chr>              <dbl>  <dbl>      <dbl> <dbl>   <dbl>
1   267             83.9 NA      Colombia            7.92   7.67        7.5   7.58    7.67
2   534             82.9 Bourbon Brazil              7.67   7.58        7.5   7.58    7.5
3   801             82   Gesha   Malawi              7.5    7.42        7.33  7.33    7.5
4  1068             80.6 NA      Colombia            7.08   7.25        7     7.08    7.33
5  1335             78.1 NA      Ecuador             7.5    7.67        7.75  5.17    5.25
```

The trouble with systematic sampling

```
coffee_ratings %>%
  ggplot(aes(x = rowid, y = aftertaste)) +
  geom_point() +
  geom_smooth()
```

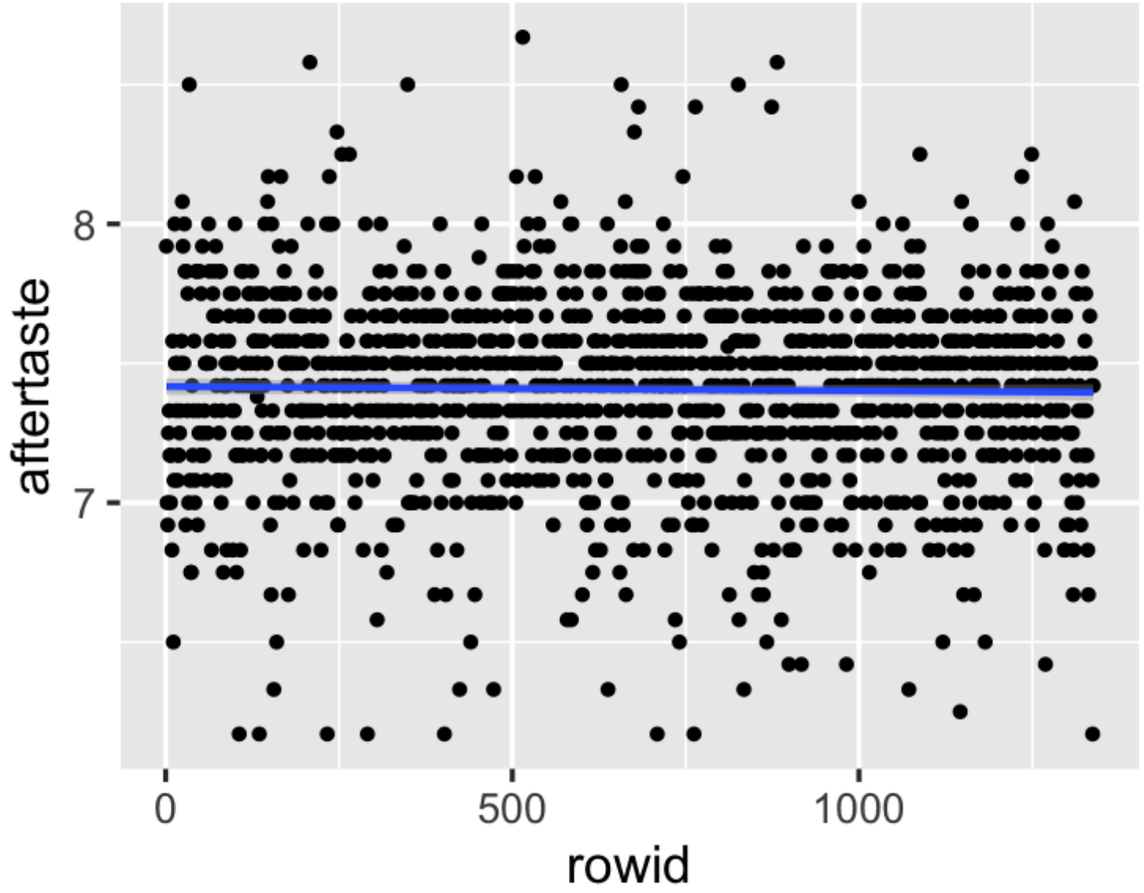
Systematic sampling is only safe if you don't see a pattern in this scatter plot.



Making systematic sampling safe

```
shuffled <- coffee_ratings %>%
  slice_sample(prop = 1) %>%
  select(- rowid) %>%
  rowid_to_column()
```

```
shuffled %>%
  ggplot(aes(x = rowid, y = aftertaste)) +
  geom_point() +
  geom_smooth()
```



Shuffling rows + systematic sampling is the same as simple random sampling.

Let's practice!

SAMPLING IN R

datacamp

Stratified and weighted random sampling

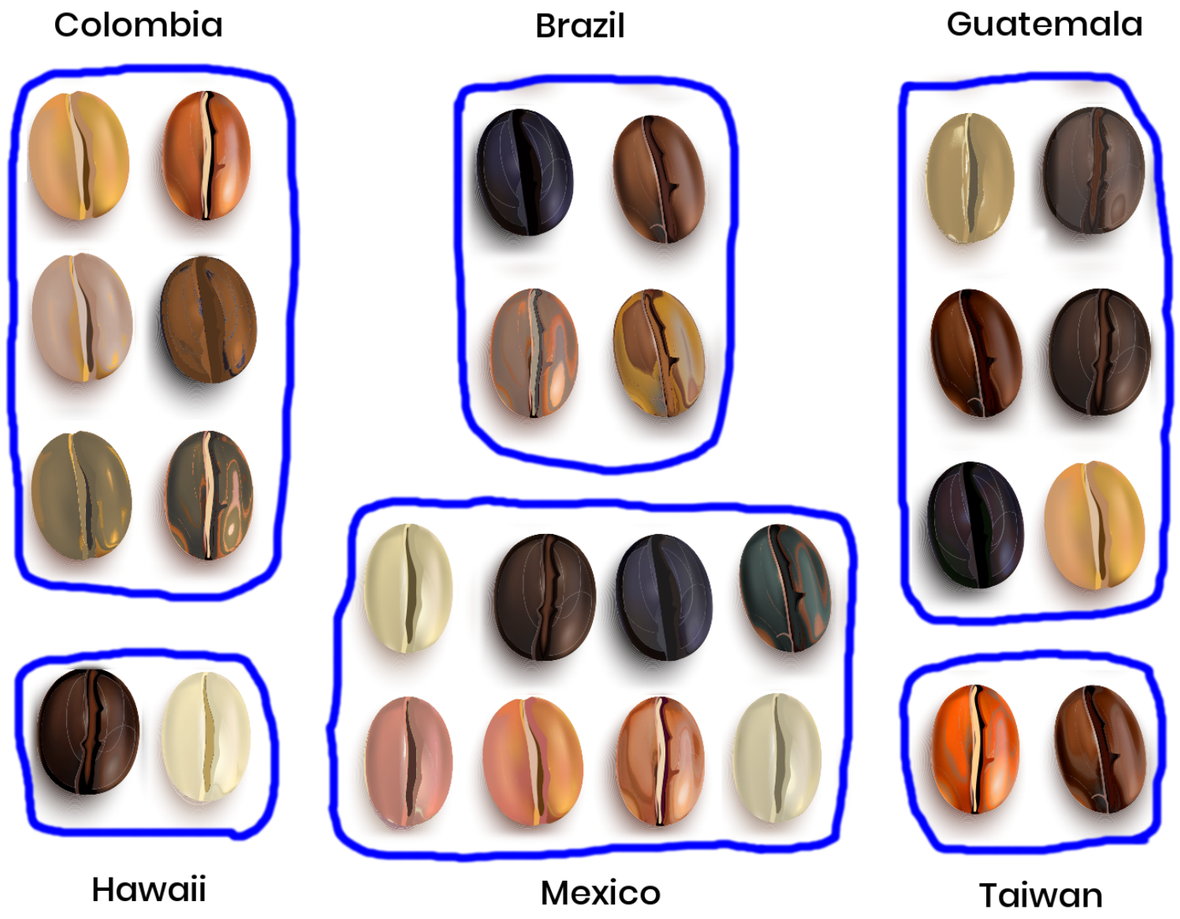
SAMPLING IN R



Richie Cotton
Data Evangelist at DataCamp

datacamp

Coffees by country




```
top_counts <- coffee_ratings %>%
  count(country_of_origin, sort = TRUE) %>%
  head()
```

```
# A tibble: 6 x 2
  country_of_origin           n
  <chr>                   <int>
1 Mexico                    236
2 Colombia                  183
3 Guatemala                 181
4 Brazil                    132
5 Taiwan                     75
6 United States (Hawaii)     73
```

[1] The dataset lists Hawaii and Taiwan as countries for convenience, as they are notable coffee growing regions.

Filtering for 6 countries

```
top_counted_countries <- c(
  "Mexico", "Colombia", "Guatemala",
  "Brazil", "Taiwan", "United States (Hawaii)"
)
coffee_ratings_top <- coffee_ratings %>%
    filter(country_of_origin %in% top_counted_countries)
```

Or, equivalently

```
coffee_ratings_top <- coffee_ratings %>%
  semi_join(top_counts)
```

[1] Learn about semi joins in "Joining Data with dplyr", Chapter 3.

Counts of a simple random sample

```
coffee_ratings_samp <- coffee_ratings_top %>%
  slice_sample(prop = 0.1)
```

```
coffee_ratings_samp %>%
  count(country_of_origin, sort = TRUE) %>%
  mutate(percent = 100 * n / sum(n))
```

```
# A tibble: 6 x 3
  country_of_origin           n percent
  <chr>                   <int>   <dbl>
1 Guatemala                  24    27.3
2 Mexico                     23    26.1
3 Brazil                     12    13.6
4 Colombia                   11    12.5
5 Taiwan                      9    10.2
6 United States (Hawaii)      9    10.2
```

Comparing counts

Population

```
# A tibble: 6 x 3
  country_of_origin           n percent
  <chr>                   <int>   <dbl>
1 Mexico                    236    26.8
2 Colombia                  183    20.8
3 Guatemala                 181    20.6
4 Brazil                    132    15
5 Taiwan                     75     8.52
6 United States (Hawaii)     73     8.30
```

10% sample

```
# A tibble: 6 x 3
  country_of_origin           n percent
  <chr>                   <int>   <dbl>
1 Guatemala                  24    27.3
2 Mexico                     23    26.1
3 Brazil                     12    13.6
4 Colombia                   11    12.5
5 Taiwan                      9    10.2
6 United States (Hawaii)      9    10.2
```

Proportional stratified sampling

```
coffee_ratings_strat <- coffee_ratings_top %>%
  group_by(country_of_origin) %>%
  slice_sample(prop = 0.1) %>%
  ungroup()
```

```
coffee_ratings_strat %>%
  count(country_of_origin, sort = TRUE) %>%
  mutate(percent = 100 * n / sum(n))
```

```
# A tibble: 6 x 3
  country_of_origin           n percent
  <chr>                   <int>   <dbl>
1 Mexico                     23   26.7
2 Colombia                   18   20.9
3 Guatemala                  18   20.9
4 Brazil                     13   15.1
5 Taiwan                      7    8.14
6 United States (Hawaii)      7    8.14
```

Equal counts stratified sampling

```
coffee_ratings_eq <- coffee_ratings_top %>%
  group_by(country_of_origin) %>%
  slice_sample(n = 15) %>%
  ungroup()
```

```
coffee_ratings_eq %>%
  count(country_of_origin, sort = TRUE) %>%
  mutate(percent = 100 * n / sum(n))
```

```
# A tibble: 6 × 3
  country_of_origin           n percent
  <chr>                   <int>   <dbl>
1 Brazil                     15    16.7
2 Colombia                   15    16.7
3 Guatemala                  15    16.7
4 Mexico                     15    16.7
5 Taiwan                     15    16.7
6 United States (Hawaii)     15    16.7
```

Weighted random sampling

```
coffee_ratings_weight <- coffee_ratings_top %>%
  mutate(
    weight = ifelse(country_of_origin == "Taiwan", 2, 1)
  ) %>%
  slice_sample(prop = 0.1, weight_by = weight)
```

```
coffee_ratings_weight %>%
  count(country_of_origin, sort = TRUE) %>%
  mutate(percent = 100 * n / sum(n))
```

10% weighted sample

```
# A tibble: 6 x 3
  country_of_origin           n percent
  <chr>                   <int>   <dbl>
1 Mexico                     23   26.1
2 Guatemala                  20   22.7
3 Taiwan                     15   17.0
4 Brazil                     12   13.6
5 Colombia                   10   11.4
6 United States (Hawaii)      8    9.09
```

Let's practice!

SAMPLING IN R

datacamp

Cluster sampling

SAMPLING IN R



Richie Cotton
Data Evangelist at DataCamp

datacamp

Stratified sampling vs. cluster sampling

Stratified sampling

- Split the population into subgroups
- Use simple random sampling on every subgroup

Cluster sampling

- Use simple random sampling to pick some subgroups
- Use simple random sampling on only those subgroups

Varieties of coffee



```
varieties_pop <- unique(
  coffee_ratings$variety
)
```

```
 [1] "Bourbon"
 [2] "Catimor"
 [3] "Ethiopian Yirgacheffe"
 [4] "Caturra"
 [5] "SL14"
...
[27] "Marigojipe"
[28] "Pache Comun"
```

Stage 1: sampling for subgroups



```
varieties_samp <- sample(
  varieties_pop,
  size = 3
)
```

```
"Sumatra"       "Blue Mountain" "SL28"
```

Stage 2: sampling each group

```
coffee_ratings %>%
  filter(variety %in% varieties_samp) %>%
  group_by(variety) %>%
  slice_sample(n = 5) %>%
  ungroup()
```

Stage 2 output

```
# A tibble: 10 x 8
   total_cup_points variety       country_of_origin aroma flavor aftertaste  body balance
              <dbl> <chr>         <chr>             <dbl>  <dbl>      <dbl> <dbl>   <dbl>
 1             81.5 Blue Mountain Haiti              7.42   7.33       7.25  7.42    7.33
 2             82.7 Blue Mountain Mexico             7.75   7.58       7.25  7.67    7.58
 3             84.5 SL28          Kenya              7.92   7.83       7.67  7.67    7.75
 4             81.9 SL28          Zambia             7.67   7.08       7.42  7.75    7.42
 5             84.7 SL28          Kenya              7.75   7.92       7.83  7.58    7.75
 6             85.5 SL28          Kenya              7.92   7.92       7.83  7.83    7.92
 7             83.8 SL28          Kenya              7.75   7.58       7.5   7.75    7.75
 8             86.6 Sumatra       Taiwan             8      8          8     8       8.17
 9             81.7 Sumatra       Indonesia          7.17   7.42       7.33  7.33    7.42
10             83.5 Sumatra       Indonesia          7.25   7.67       7.58  7.83    7.58
```

Multistage sampling

- Cluster sampling is a type of multistage sampling.
- You can have > 2 stages.
- Countrywide surveys may sample states, counties, cities, and neighborhoods.

Let's practice!

SAMPLING IN R

datacamp

Comparing sampling methods

SAMPLING IN R



Richie Cotton
Data Evangelist at DataCamp

datacamp

Review of sampling techniques

Setup

```
top_counted_countries <- c(
  "Mexico", "Colombia", "Guatemala",
  "Brazil", "Taiwan", "United States (Hawaii)"
)
coffee_ratings_top <- coffee_ratings %>%
  filter(country_of_origin %in% top_counted_countries)
```

Simple random sampling

```
coffee_ratings_srs <- coffee_ratings_top %>%
  slice_sample(prop = 1 / 3)
```

Stratified sampling

```
coffee_ratings_strat <- coffee_ratings_top %>%
  group_by(country_of_origin) %>%
  slice_sample(prop = 1 / 3) %>%
  ungroup()
```

Cluster sampling

```
top_countries_samp <- sample(top_counted_countries, size = 2)
coffee_ratings_clust <- coffee_ratings_top %>%
  filter(country_of_origin %in% top_countries_samp) %>%
  group_by(country_of_origin) %>%
  slice_sample(n = nrow(coffee_ratings_top) / 6) %>%
  ungroup()
```

Calculating mean cup points

Population

```
coffee_ratings_top %>%
  summarize(mean_points = mean(total_cup_points))
```

```
81.9
```

Simple random sample

```
coffee_ratings_srs %>%
  summarize(mean_points = mean(total_cup_points))
```

```
82.0
```

Stratified sample

```
coffee_ratings_strat %>%
  summarize(mean_points = mean(total_cup_points))
```

```
81.8
```

Cluster sample

```
coffee_ratings_clust %>%
  summarize(mean_points = mean(total_cup_points))
```

```
81.2
```

Mean cup points by country: simple random

Population

```
coffee_ratings_top %>%
  group_by(country_of_origin) %>%
  summarize(mean_points = mean(total_cup_points))
```

```
# A tibble: 6 x 2
  country_of_origin      mean_points
  <chr>                        <dbl>
1 Brazil                        82.4
2 Colombia                      83.1
3 Guatemala                     81.8
4 Mexico                        80.9
5 Taiwan                        82.0
6 United States (Hawaii)        81.8
```

Simple random sample

```
coffee_ratings_srs %>%
  group_by(country_of_origin) %>%
  summarize(mean_points = mean(total_cup_points))
```

```
# A tibble: 6 x 2
  country_of_origin      mean_points
  <chr>                        <dbl>
1 Brazil                        82.3
2 Colombia                      83.1
3 Guatemala                     81.5
4 Mexico                        81.1
5 Taiwan                        82.8
6 United States (Hawaii)        82.7
```

Mean cup points by country: stratified

Population

```
coffee_ratings_top %>%
  group_by(country_of_origin) %>%
  summarize(mean_points = mean(total_cup_points))
```

```
# A tibble: 6 x 2
  country_of_origin      mean_points
  <chr>                        <dbl>
1 Brazil                        82.4
2 Colombia                      83.1
3 Guatemala                     81.8
4 Mexico                        80.9
5 Taiwan                        82.0
6 United States (Hawaii)        81.8
```

Stratified sample

```
coffee_ratings_strat %>%
  group_by(country_of_origin) %>%
  summarize(mean_points = mean(total_cup_points))
```

```
# A tibble: 6 x 2
  country_of_origin      mean_points
  <chr>                        <dbl>
1 Brazil                        82.4
2 Colombia                      82.9
3 Guatemala                     81.7
4 Mexico                        80.7
5 Taiwan                        82.3
6 United States (Hawaii)        81.2
```

Mean cup points by country: cluster

Population

```
coffee_ratings_top %>%
  group_by(country_of_origin) %>%
  summarize(mean_points = mean(total_cup_points))
```

```
# A tibble: 6 x 2
  country_of_origin      mean_points
  <chr>                        <dbl>
1 Brazil                        82.4
2 Colombia                      83.1
3 Guatemala                     81.8
4 Mexico                        80.9
5 Taiwan                        82.0
6 United States (Hawaii)        81.8
```

Cluster sample

```
coffee_ratings_clust %>%
  group_by(country_of_origin) %>%
  summarize(mean_points = mean(total_cup_points))
```

```
# A tibble: 2 x 2
  country_of_origin mean_points
  <chr>                   <dbl>
1 Mexico                   80.8
2 Taiwan                   82.0
```

Let's practice!

SAMPLING IN R