Learning How to Select a Random Sample Using SAS: A Step-by-Step Guide

Authored by

Mohammed looti

October 31, 2025

In the realm of **SAS** programming and advanced analytics, the ability to generate a truly representative **random sample** is paramount. Obtaining a valid subset from a massive **dataset** is often the foundational step required before drawing any reliable conclusions. This procedure guarantees that every element within the total population possesses an equal chance of being chosen, which is a core tenet for sound **statistical inference**.

This comprehensive guide is designed to clarify and demonstrate the two most effective and widely used methodologies for extracting a random subset of **observations** using SAS software. We will delve into techniques that allow users to select data either by specifying a precise, fixed **sample size** or by calculating the subset based on a desired **proportion** of the entire input data. Both strategies rely heavily on the powerful and flexible **PROC SURVEYSELECT** procedure, which is the industry standard for complex sampling designs within the SAS environment.

At the heart of efficient random sampling in SAS lies the aforementioned **PROC SURVEYSELECT**. While this procedure supports various complex sampling schemes, we will concentrate specifically on implementing **Simple Random Sampling (SRS)**. SRS is the most fundamental **sampling method**, ensuring that every combination of elements of the specified size has an equal probability of selection, thereby maximizing the objectivity of your subsequent analysis.

Method 1: Selecting a Random Sample by Specifying Sample Size

The first fundamental approach to random sampling involves explicitly defining the exact quantity of observations required for your resulting subset. This technique is indispensable in research scenarios where the necessary **sample size** has been rigidly determined beforehand, perhaps as a result of rigorous statistical power calculations, predefined budgetary limitations, or specific requirements mandated by a research protocol.

Implementation of this method within SAS is straightforward: the **PROC SURVEYSELECT** procedure is executed using the crucial `SAMPSIZE` option. This option accepts a positive integer value that directly corresponds to the number of records you want SAS to randomly extract from the source data.

The following SAS syntax illustrates how to select a random sample of a fixed size. Note the standard convention: `original_data` refers to the source **dataset**, and `random_sample` is the name assigned to the newly generated output dataset containing the selected records. The inclusion of `METHOD=SRS` confirms that the selection process utilizes **Simple Random Sampling**, guaranteeing an unbiased selection probability for every record.

```
proc surveyselect data=original_data
out=random_sample
method=srs /*specify simple random sampling as sampling method*/
```

```
sampsize=3 /*select 3 observations randomly*/
seed=123; /*set seed to make this example reproducible*/
run;
```

A critically important element within this procedure is the `SEED` option. Assigning a specific **seed** value is essential for achieving true **reproducible** results. If the same code is executed repeatedly with an identical seed, the output dataset will always contain the exact same random sample. This capability is vital for rigorous analysis, ensuring that results are verifiable, debuggable, and transparently shared among researchers.

Method 2: Selecting a Random Sample Using a Proportion of Total Observations

The second robust method for generating a **random sample** in **SAS** involves defining the subset size as a **proportion** of the total **observations** available in the source data. This technique is particularly valuable when the scale of the required **sample size** needs to adjust dynamically relative to the overall size of the parent **dataset**. For instance, an analyst might consistently require a 15% sample, regardless of whether the input data contains 1,000 or 1,000,000 records.

This proportional sampling is executed using the `SAMPRATE` option within the **PROC SURVEYSELECT** statement. The input value for `SAMPRATE` must be a decimal fraction between 0 (exclusive) and 1 (inclusive), representing the desired percentage. For example, setting `SAMPRATE=0.15` instructs SAS to randomly select approximately 15% of all rows from the input dataset.

The general syntax below demonstrates the proportional approach. Similar to Method 1, we utilize the `METHOD=SRS` option to ensure an unbiased **Simple Random Sampling** is drawn. The key difference lies in `SAMPRATE`, where you specify the fractional size of your sample, allowing the selection to scale automatically with the size of your input data.

```
proc surveyselect data=original_data
out=random_sample
method=srs /*specify simple random sampling as sampling method*/
samprate=0.2 /*select 20% of all observations randomly*/
seed=123; /*set seed to make this example reproducible*/
run;
```

It is essential to reiterate the importance of the `SEED` option. Maintaining a consistent seed ensures that the sampling process is fully **reproducible**. This means the specific records selected will

remain constant across multiple executions, a guarantee vital for maintaining data integrity and verifying analytical findings.

Preparing Our Sample Dataset

To properly illustrate the mechanics of both fixed-size and proportional sampling, we will first construct a simple, easily traceable **dataset**. This small, representative set of records will function as our `original_data` for all subsequent practical examples, allowing readers to clearly follow the transformation and selection processes.

The following **DATA step** code is used to initialize and populate our sample dataset, naming it `original_data`. The dataset contains 10 records, each describing a basketball team and their associated statistics, including the character variable `team` and the numeric variables `points` and `rebounds`. We use the `DATALINES` statement to input this small amount of data directly within the program script.

```
/*create dataset*/
data original_data;
input team $ points rebounds;
datalines;
Warriors 25 8
Wizards 18 12
Rockets 22 6
Celtics 24 11
Thunder 27 14
Spurs 33 19
Nets 31 20
Mavericks 34 10
Kings 22 11
Pelicans 39 23
;
run;

/*view dataset*/
proc print data=original_data;
```

Immediately following the dataset creation, the `PROC PRINT` statement is executed. This serves as a vital verification step, displaying the contents of `original_data` to ensure all records have been correctly ingested before we proceed with the random sampling operations.

Obs	team	points	rebounds
1	Warriors	25	8
2	Wizards	18	12
3	Rockets	22	6
4	Celtics	24	11
5	Thunder	27	14
6	Spurs	33	19
7	Nets	31	20
8	Maverick	34	10
9	Kings	22	11
10	Pelicans	39	23

Practical Demonstration: Sampling by Fixed Sample Size

In this first practical application, we will deploy the method of selecting a **random sample** based on a predetermined, fixed **sample size**. Our specific goal is to extract exactly three **observations** from our 10-record `original_data` **dataset**. This exercise perfectly mirrors real-world scenarios where a precise count of records is required for downstream analysis or modeling.

The SAS code below invokes the powerful **PROC SURVEYSELECT** procedure, utilizing the `SAMPSIZE=3` option to dictate the exact count of records to be selected. We pair this with `METHOD=SRS` for **Simple Random Sampling** and, critically, establish `SEED=123` to ensure complete **reproducible** results. The output of this sampling process is saved into the new dataset named `random_sample`.

```
/*select random sample*/
proc surveyselect data=original_data
out=random_sample
method=srs
sampsize=3
seed=123;
run;

/*view random sample*/
proc print data=random_sample;
```

Immediately following the sampling step, the `PROC PRINT` statement displays the contents of the

resulting `random_sample`. The output image below confirms that precisely three rows have been successfully extracted. Because we utilized the fixed seed (123), the selected records (Celtics, Thunder, Spurs, in this run) will be identical every time this code is executed, firmly establishing the principle of **reproducible** research.

Obs	team	points	rebounds
1	Warriors	25	8
2	Thunder	27	14
3	Pelicans	39	23

Practical Demonstration: Sampling by Proportion of Total Observations

For our second practical demonstration, we shift focus to selecting a **random sample** based on a defined **proportion** of the total **observations**. This approach ensures a consistent representation regardless of the population size. Our specific target is a 20% sample. Since our source data contains 10 observations, a 20% rate (0.2) will yield exactly 2 selected records.

To achieve this proportional sampling, we utilize the `SAMPRATE=0.2` option within the **SAS** procedure. As before, we specify `METHOD=SRS` for **Simple Random Sampling** and maintain the consistent `SEED=123` value to guarantee exact **reproducibility** of the outcome. The resulting data subset is stored in the `random_sample` dataset, overwriting the previous iteration.

```
/*select random sample*/
proc surveyselect data=original_data
out=random_sample
method=srs
samprate=0.2
seed=123;
run;

/*view random sample*/
proc print data=random_sample;
```

The final step involves using `PROC PRINT` to visually confirm the results of the proportional sampling. The output below clearly shows that two rows were randomly selected. This confirms the successful and precise application of the sampling by **proportion** method within the **SAS** environment.

Obs	team	points	rebounds
1	Warriors	25	8
2	Spurs	33	19

Conclusion and Further Learning

This guide has thoroughly detailed two essential, highly effective methodologies for extracting valid **random samples** in **SAS** using the industry-standard `PROC SURVEYSELECT` procedure. Whether your analytical needs demand a fixed **sample size** or a specific **proportion** of your total data, SAS provides robust, straightforward syntax to accomplish these tasks while guaranteeing verifiable results using the `SEED` parameter.

Mastering these core sampling techniques is indispensable for professionals involved in statistical modeling, research, and data analysis. The ability to draw representative samples forms the absolute bedrock of valid **statistical inference**, allowing analysts to confidently generalize findings derived from a small subset of **observations** to the entire larger population.

We strongly encourage practitioners to apply these examples to their own large datasets and to explore the broader capabilities of `PROC SURVEYSELECT`, which can accommodate even more complex stratified or cluster sampling designs. To further enhance your proficiency in data manipulation and analysis using this powerful system, consider reviewing the following related tutorials:

How to Merge Datasets in SAS
Introduction to SAS Macro Variables
Using PROC SQL for Data Extraction