

Lab 6: Distributions of statistics, part 1
STT 421: Summer, 2004
Vince Melfi

As you know from class and from Chapter 3 of the text, the method of selecting a sample usually entails some randomness, and hence the statistics (sample mean, sample variance, sample median, etc.) can be thought of as realizations of random variables. In this context it is very important to have some idea about the distribution of these random variables. In this lab we will learn how to use SAS to simulate the sampling process. In the process we'll learn about the **proc surveyselect** procedure, which gives a variety of ways to select a sample from a population. Today we'll concentrate on some small datasets. In the next lab we'll use what we've learned to make a study of distributions of statistics in some more realistic and complicated settings.

Beginnings

First we'll use **proc surveyselect** on a small population in order to understand the basic syntax and output. We'll create a dataset **pop** which contains the variable **nums** with values 1, 3, 3, 5, 7, 9, 11, 13, 15, 15, 27. We'll print the data using **proc print** and compute basic (population) statistics using **proc means**. We'll then use **proc surveyselect** to take a simple random sample of size $n = 5$ from this population, print the values in the sample, and compute basic (sample) statistics.

```
data pop;
  input nums;
  cards;
  1
  3
  3
  5
  7
  9
  11
  13
  15
  15
  27
  ;

proc print data=pop;
  title 'the small population';

proc means data=pop;
  title 'population mean, etc.';
```

```

proc surveystest data = pop
  method=srs n=5 out=samp1;

proc print data=samp1;
  title 'a simple random sample of size 5';

proc means data=samp1;
  title 'mean of the sample of size 5';

run;

```

Explanation

1. The **data** statement, the first **proc print**, and the first **proc means** create, print, and compute basic statistics for the **pop** dataset. Note that the (population) mean is about 9.9 and the population standard deviation is about 7.5.
2. Next comes **proc surveystest**. We tell it the population to take the sample from (in our case **pop**), then the sampling method via **method = srs** (which stands for a simple random sample), then the sample size **n=5**, then the name **samp1** of the dataset that will contain the sample via **out = samp1**.
3. By looking at the output of the **print** and **means** procedures, you should be able to see the actual 5 observations selected and their means.
 - (a) Write down the 5 observations selected.
 - (b) Write down their (sample) mean.
 - (c) Do you think that everyone in the class got the same sample data as you?

Repeated samples

In order to gain some information about the distribution of the sample mean (or another statistic) we will take many independent simple random samples from the population. This is easily accomplished via the **rep** option to the **proc surveystest** procedure. The following bit of SAS code illustrates this in the context of the same dataset as above. We will take 8 simple random samples from the dataset and compute the mean of each.

```

data pop;
  input nums;
  cards;
  1
  3
  3
  5
  7
  9

```

```

11
13
15
15
27
;

proc surveysselect data = pop n=5 rep=8
  out = samp2;

proc print data=samp2;
  title '8 srs from pop data';

proc means data=samp2;
  by replicate;
  title 'means of 8 srs from pop data';

run;

```

Explanation

1. There's not a lot to say. Note that **proc surveysselect** automatically creates a variable **replicate** that goes into the created dataset (in our case **samp2**) that indicates which SRS the observations came from. We use this with **proc means** to get separate means for the 8 samples.
2. Write down the 8 sample means. Are any of them close to the population mean of 9.9?

Storing the sample statistics

What we really want to do is repeat the sampling procedure *many* times. But we don't want to look at all the sample means. Instead, we'd like to know something about their distribution, possibly by drawing a histogram of the sample means. To facilitate this, we will learn how to suppress the printout from **proc means** and to store its output in a new dataset. We'll still do all this with the same dataset as above. This time we'll take 800 samples of size $n = 5$.

```

data pop;
  input nums;
  cards;
1
3
3
5
7
9

```

```

11
13
15
15
27
;

proc surveysselect data = pop n=5 rep=800
  out = samp3;

proc means data=samp3 noprint;
  output out=means5 mean = Mean;
  var nums;
  by replicate;

proc univariate data=means5;
  histogram mean;

run;

```

Explanation

1. We used **noprint** to tell SAS not to print the 800 sample means!
2. We used **output out=means5 sampmean=Mean** to tell SAS to create a new dataset named **means5** and store in it a variable named **sampmean** that contains the 800 means computed by **proc means**.
3. The output of **proc univariate** and the associated histogram give us an idea about the distribution of the mean in this (toy) problem.
 - (a) What is the mean of the 800 sample means? Is it close to the population mean?
 - (b) What is the standard deviation of the 800 sample means? Is it smaller, larger, or about the same as the population standard deviation?
 - (c) Is the distribution of sample means approximately symmetric?

Other statistics

So far we've focused on the distribution of the sample mean. But it's just as easy to learn about the distribution of the sample median or other statistics. For example, the following SAS code does about the same thing as the above code, except it investigates the sampling distribution of the sample median. Note that we need to replace **proc means** by **proc univariate** to get the median.

```

data pop;
  input nums;

```

```

cards;
1
3
3
5
7
9
11
13
15
15
27
;

proc surveysselect data = pop n=5 rep=800
  out = samp4;

proc univariate data=samp4 noprint;
  output out=medians5 median = Median;
  var nums;
  by replicate;

proc univariate data=medians5;
  histogram median;

run;

```

Run this program and make sure you understand it well, since next lab you'll be running a lot of such programs.

Exercise

Use SAS to do Exercise 3.68 (b) in the text. Since SAS is fast, replace 10 by 1000 in (b) so that you're taking 1000 simple random samples of size $n = 4$. You should be able to mimic the code above to accomplish this.