Foundations for Statistical Inference

– Sampling Distributions

In this lab, we investigate the ways in which the statistics from a random sample of data can serve as point estimates for population parameters. We're interested in formulating a *sampling distribution* of our estimate in order to learn about the properties of the estimate, such as its distribution.[1]

The Data

We consider real estate data from the city of Ames, Iowa. The details of every real estate transaction in Ames are recorded by the City Assessor's office. Our particular focus for this lab will be residential home sales in Ames between 2006 and 2010. This collection represents our population of interest. In this lab, we would like to learn about these home sales by taking smaller samples from the full population. Let's load the data.

- Note: If you are using SAS University Edition, you need to ensure that interactive mode is **turned off**. To do this, click the button to the right of Sign Out in the upper right corner of the window and then click Preferences. In the Preferences window, on the General tab, the bottom check box (located next to the text **Start new programs** in interactive mode) should not be selected. If the box is selected, you need to clear it and save your change.

```
filename amesh url 'http://www.openintro.org/stat/data/ames_sas.csv';

proc import datafile=amesh out=work.ames dbms=csv replace;
   getnames=yes;
   guessingrows=max;
run;
```

We see that there are quite a few variables in the data set, enough to do a very in-depth analysis. For this lab, we'll restrict our attention to just two of the variables: the above-ground living area of the house in square feet (**Gr_Liv_Area**) and the sale price (**SalePrice**).

Let's look at the distribution of area in our population of home sales by calculating a few summary statistics and making a histogram within the UNIVARIATE procedure. PROC UNIVARIATE returns summary statistics for the variable **Gr_Liv_Area**, and the HISTOGRAM statement requests the generation of a histogram of the variable **Gr_Liv_Area**. The INSET statement inserts a box into the

[1] This is a product of OpenIntro that is released under a Creative Commons Attribution-ShareAlike 3.0 Unported (http://creativecommons.org/licenses/by-sa/3.0/). This lab was written for OpenIntro by Andrew Bray and Mine C¸ etinkaya-Rundel and modified by SAS Institute Inc. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries (® indicates USA registration) and are not included under the CC-BY-SA license.

northeast corner of the histogram (POS=NE) containing the requested statistics: the mean, median, standard deviation, minimum, and maximum.

```
proc univariate data=work.ames;
   var Gr_Liv_Area;
   histogram Gr_Liv_Area;
   inset mean median std min max / pos=ne;
run;
```

To change the number of bins in the histogram, you can include, in the HISTOGRAM statement, the NMIDPOINTS= option to specify the number of midpoints (bins). Or you can include MIDPOINTS= to specify the list of exact values, in increasing order, of the locations of the midpoints of each bar that you want to generate. Here is an example of how the HISTOGRAM line would look with the NMIDPOINTS= option included.

```
histogram Gr_Liv_Area / nmidpoints=5;
```

Exercise 1: Describe this population distribution.

The Unknown Sampling Distribution

In this lab, we have access to the entire population, but this is rarely the case in real life. Gathering information on an entire population is often extremely costly or impossible. Because of this, we often take a sample of the population and use that to understand the properties of the population.
If we were interested in estimating the mean living area in Ames based on a sample, we could use the following code to survey the population:

```
proc surveyselect data=work.ames out=work.amessample sampsize=50
                  method=srs ranuni;
run;
```

PROC SURVEYSELECT collects a simple random sample of size 50 from the data set **work.ames**, which is assigned to **work.amessample**. This is like going into the City Assessor's database and pulling up the files on 50 random home sales. Working with these 50 files would be considerably simpler than working with all home sales. To view this sample, the following PROC PRINT code will suffice:

```
proc print data=work.amessample;
run;
```

Exercise 2: Describe the distribution of this sample. How does it compare to the distribution of the population?

If we're interested in estimating the average living area in homes in Ames using the sample, our best single guess is the sample mean.

```
proc means data=work.amessample mean;
   var Gr_Liv_Area;
run;
```

Depending on which 50 homes you selected, your estimate could be a bit above or a bit below the true population mean. In general, though, the sample mean turns out to be a pretty good estimate of the average living area, and we were able to get it by sampling less than 3% of the population.

Exercise 3: Take a second sample, also of size 50, and call it **work.amessample2**. How does the mean of **work.amessample2** compare with the mean of **work.amessample**? Suppose we took two more samples, one of size 100 and one of size 1000.Which would you think would provide a more accurate estimate of the population mean?

Not surprisingly, every time we take another random sample, we get a different **samplemean**. It's useful to get a sense of just how much variability we should expect when estimating the population mean this way. The distribution of sample means, called the sampling distribution, can help us understand this variability. In this lab, because we have access to the population, we can build up the sampling distribution for the sample mean by repeating the above steps many times. Here we will generate 5000 samples and compute the sample mean of each saving each of them into the data set **work.reprun**.

```
proc surveyselect data=work.ames out=work.amessampler sampsize=50
                  method=srs reps=5000 ranuni;
run;

proc means data=work.amessampler mean noprint;
   by replicate;
   var Gr_Liv_Area;
   output out=work.reprun mean=sampmean;
run;
```

With the mean of all 5000 samples calculated and saved in **work.reprun**, we can generate a histogram of these values within PROC UNIVARIATE.

```
proc univariate data=work.reprun;
   var sampmean;
   histogram sampmean;
run;
```

Here we use SAS to take 5000 samples of size 50 from the population, calculate the mean of each sample, and store the results in the data set named **work.reprun**. By invoking the REPS= option in PROC SURVEYSELECT, we can take multiple samples of the same size from the population. These samples are stored within a data set named **work.amessampler**. To keep track of and separate each sample taken, SAS generates a variable named **replicate** that we then use within the BY statement of PROC MEANS to assist in the organization of the mean calculations.

Exercise 4: How many observations are there in **work.reprun**? Describe the sampling distribution, and be sure to specifically note its center. Would you expect the distribution to change if we instead collected 50,000 sample means?

Exercise 5: To make sure that you understand what you've done, try running a smaller version. This time, take only 100 samples of size 50 from the population. Change the data set by saving the sample means to **work.reprunsmall**. Print the storage data set to your screen.

```
proc print data=work.reprunsmall;
run;
```

How many observations are there in the **work.reprunsmall** data set? What does each observation represent?

Sample Size and the Sampling Distribution

Let's compute a sampling distribution, specifically, this one:

```
proc univariate data=work.reprun;
   var sampmean;
   histogram sampmean;
run;
```

The sampling distribution that we computed tells us much about estimating the average living area in homes in Ames. Because the sample mean is an unbiased estimator, the sampling distribution is centered at the true average living area of the population, and the spread of the distribution indicates how much variability is induced by sampling only 50 home sales.

To get a sense of the effect that sample size has on our distribution, let's build up two more sampling distributions: one based on a sample size of 10 and another based on a sample size of 100. With the mechanics of the setup understood, this becomes quite easy.

```
proc surveyselect data=work.ames out=work.amessampler10 sampsize=10
                  method=srs reps=5000 ranuni;
run;

proc means data=work.amessampler10 mean noprint;
   by replicate;
   var Gr_Liv_Area;
   output out=work.reprun10 mean=sampmean;
run;

proc surveyselect data=work.ames out=work.amessampler100 sampsize=100
                  method=srs reps=5000 ranuni;
run;

proc means data=work.amessampler100 mean noprint;
   by replicate;
   var Gr_Liv_Area;
   output out=work.reprun100 mean=sampmean;
run;
```

To see the effect that different sample sizes have on the sampling distribution, plot the three histograms.

```
proc univariate data=work.reprun;
   title 'Sample Size = 50';
   var sampmean;
   histogram sampmean;
run;

proc univariate data=work.reprun10;
   title 'Sample Size = 10';
   var sampmean;
   histogram sampmean;
run;

proc univariate data=work.reprun100;
   title 'Sample Size = 100';
   var sampmean;
   histogram sampmean;
run;

title;
```

An alternative is to use PROC SGPANEL to put the three histograms together. First, we combine the three **reprun** data sets into a single data set, **work.allsamples**. The IN= option creates the variables **Sample1**, **Sample2**, and **Sample3** that can be used within Boolean questioning to determine from which data set the observation originated. The value of **Sample1** will be 1 if the observation is from **work.reprun** and 0 otherwise. This is similar for **Sample2** and **Sample3**. The variable **group**, used later as

our PANELBY variable, is generated using Boolean logic to separate the samples within our overall data set. Within PROC SGPANEL, we request that a histogram be made using the **sampmean** variable (**histogram sampmean**). However, we want three histograms, separated by our PANELBY variable **group**.

```
data work.allsamples;
   set work.reprun (IN=Sample1)
       work.reprun10 (IN=Sample2)
       work.reprun100 (IN=Sample3);
   group = 1*(Sample1) + 2*(Sample2) + 3*(Sample3);
run;
proc sgpanel data=work.allsamples;
   panelby group;
   histogram sampmean;
run;
```

Exercise 6: When the sample size is larger, what happens to the center? What about the spread?

Notes:

This lab was adapted for *OpenIntro* by Andrew Bray and mine Cetinkaya-Rundel from a lab written by Mark Hansen of UCLA Statistics.