Lab #5: Central Limit Theorem/Normalcy

A. Objectives:

1. Subsets in SAS
2. Explore the Central Limit Theorem
3. How Normal is our Histogram?

Reading from external files/subseting:

As you recall, the **PROC IMPORT** statement is the best way to enter external data sets. The CSV file we will be using is called “Samples.csv”. Download and save it in your favorite folder and mark the complete path to it. Then use the following code to import it, making sure you put the correct path on the **DATAFILE** argument.

```
PROC IMPORT OUT= Samples
            DATAFILE= "put the complete path here…\Samples.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

This set contains 5,000 samples drawn at random from the uniform distribution X = [0,10] presented in rows. The last column of this table is the mean of each sample.

Let’s try to print our dataset using the familiar **PROC PRINT** command as follows:

```
PROC PRINT data=Samples;
TITLE 'Uniform Samples';
RUN;
```

As you might notice, SAS is a bit troubled! Let’s try to “cut down” our dataset into a smaller one of the first 50 observations. What we will do is create a new data, called Samples50 from the big one, using the following code:

```
ODS HTML CLOSE;
ODS HTML;
DATA Samples50;
SET Samples(OBS=50);
RUN;
PROC PRINT data=Samples50;
TITLE ‘The first 50 observations’;
RUN;
```

We used the commands **ODS HTML CLOSE**; and **ODS HTML**; to clean the output and don’t have to see again the big dataset.

Once again the command **DATA Samples50**, names the new small dataset: *Samples50*
The command **SET Samples(OBS=50);** instructs SAS to cut the first 50 observations from the Set *Samples*

Suppose then that we wanted to keep only the observations from the 50th to the 90th. The following code will do the trick:

```
DATA Samples50_90;
SET Samples(firstobs=50 OBS=90);
RUN;
PROC PRINT data = Samples50_90;
TITLE ‘Observations 50 to 90’;
RUN;
```

The command **SET Samples(firstobs=50 OBS=90);** makes SAS start from observation 50 and stop at observation 90.

Suppose now that we only care about the Column Mean and we don’t really need the actual sample values. Let’s create a new dataset called SampleMean_20 that only has the column Mean and the Sample number, with the first 20 observations using the following code

```
DATA SamplesMean_20;
SET Samples(OBS=20);
KEEP Sample Means;
RUN;
PROC PRINT data = SamplesMean_20;
TITLE ‘Sample means for the first 20 samples’;
RUN;
```

As you can see the command: **KEEP Sample Means;** keeps only the columns with names *Sample* (the column of the names of the samples) and *Means* (the column of the means). All other columns are dropped. For more information on subsetting datasets in SAS the IDRE site at UCLA has a great tutorial found here:

https://stats.idre.ucla.edu/sas/modules/ubsetting-data-in-sas/

Exploring the Central Limit Theorem:

As we learned in the lectures, the Central Limit Theorem says roughly that if one considers the means of independent random variables then their sum tends towards the Normal Distribution independent of what the original random variables are.

Let’s try to convince ourselves about this fact by utilizing the dataset Samples we have already loaded. AS we mentioned, the Samples are drawn from a Uniform Distribution with mean 5 and standard deviation 5. To do that, let’s try to see how the means are distributed if we keep let’s say 20 samples.

Conveniently we have created the dataset SamplesMean_20 which contains that information. The only thing we need to do is create a histogram of the means and explore its shape. We will first use our familiar **PROC GCHART**

```
TITLE ‘Histogram of Means for 20 samples’;
PROC GCHART DATA= SamplesMean_20;
            VBAR Means/midpoints= 0 to 10 by 1;
RUN;
```

The histogram does not look normal. But wait! We are only using 20 samples, let’s increase that to 40 and see what happens. First let’s create a new subset with the first 40 observations and then create its GCHART as follows:

```
DATA SamplesMean_40;
SET Samples(OBS=40);
KEEP Sample Means;
RUN;
TITLE ‘Histogram of Means for 40 samples’;
PROC GCHART DATA= SamplesMean_40;
            VBAR Means/midpoints= 0 to 10 by 1;
RUN;
```

A bit better but still not exactly normal. OK how about we use 400 samples?

```
DATA SamplesMean_400;
SET Samples(OBS=400);
KEEP Sample Means;
RUN;
TITLE ‘Histogram of Means for 400 samples’;
PROC GCHART DATA= SamplesMean_400;
            VBAR Means/midpoints= 0 to 10 by 1;
RUN;
```

And what about 4000 samples?

```
DATA SamplesMean_4000;
SET Samples(OBS=4000);
KEEP Sample Means;
RUN;
TITLE ‘Histogram of Means for 4000 samples’;
PROC GCHART DATA= SamplesMean_4000;
            VBAR Means/midpoints= 0 to 10 by 1;
RUN;
```

It looks closer and closer to a Normal Distribution, with mean 5!

How normal is our Histogram?

As we have mentioned more than once, looks can be deceiving in Statistics. For example, although a histogram may look a certain way, it depends on various things like: outliers, bin width, sample size and others.

During the lectures we learned that many Random Variables, or Data sets in general can be approximated by a normal distribution. The following tests measure "how good of a fit that is". One idea is to use the QQ plot which basically measures how close the theoretical quantiles are with the ones in the dataset.

Let's try to find out how the qq plot looks like for the Dataset of the 4000 Samples. This is done once again with our favorite command **PROC UNIVARIATE** as follows:

```
TITLE "QQplot for the histrogramm of means with 4000 samples";
PROC UNIVARIATE DATA = SamplesMean_4000 normal;
QQPLOT Means/ Normal(mu=est sigma=est color=red l=1);
RUN;
```

Looks pretty normal to me! The command, that computes the QQ plot is **QQPLOT Means/ Normal (mu=est sigma=est color=red l=1);** Basically we are asking SAS to estimate the normal distribution that fits the best (mu=est, sigma=est) and then compare it to the variable *Means* in our dataset.

Still there is a quantitative way of measuring normalcy, and although we don't have all the pre-requisites yet to know everything, we should explore the idea of "fitting a normal curve" on a histogram. The code that does that is:

```
TITLE "How Normal is our Histogram?";
Ods select Histogram ParameterEstimates GoodnessOfFit FitQuantiles Bins;
PROC UNIVARIATE DATA = SamplesMean_4000;
HISTOGRAM Means/ normal(percents=20 40 60 80 midpercents);
INSET n normal(ksdpval) / pos = ne format =6.3;
RUN;
```

We will worry about the ins and outs of this code in a later lab, but for now this is the way to fit a normal distribution on your dataset and also have some measures of fit, basically the output:

```
          Goodness-of-Fit Tests for Normal Distribution
Test                        Statistic             p Value
Kolmogorov-Smirnov   D      0.00908149   Pr > D      >0.150
Cramer-von Mises     W-Sq   0.05707440   Pr > W-Sq   >0.250
Anderson-Darling     A-Sq   0.37622453   Pr > A-Sq   >0.250
```

We want the p values to be large enough! (The cutoff point is up for debate)