SSH Bnk@Glue.umd.edu, Tap Sas913 (or Sas82), Sas

https://www.statlab.umd.edu/sasdoc/sashtml/onldoc.htm

0. Do Loops
   a. Generation of RV’s
   b. Demonstration of LLN using simulated data
   c. Demonstration of CLT using simulated data
   d. Proc SUMMARY

0. DO Loops

DO Loops are used when a statement or set of statements needs to be repeated many times. DO loops begin with a DO statement and end with an END statement. The statements in between are repeated the number of times indicated by the index variable in the DO statement.

Example

OPTION PS=35 LS=70;

DATA DoLoop;
DO i = 1 TO 100;
   x = 3 + 5*RANNOR(59); /* N(3,25) */;
OUTPUT;
END;
RUN;

Can apply PROCs to x:

PROC MEANS DATA=DoLoop; <--- The generated data are here!!!
OUTPUT OUT=TRY MEAN=Mean;
VAR x;
RUN;
This gives the sample mean and SD of the N(3,25) sample of size 100!!!

The MEANS Procedure

Analysis Variable : x

<table>
<thead>
<tr>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2.6718930</td>
<td>5.0943355</td>
<td>-8.1962986</td>
<td>16.2624590</td>
</tr>
</tbody>
</table>

OR Which is the same:

```
proc means data=DoLoop;
output;
var x;
run;
```

OR Which is the same:

```
proc means data=DoLoop;
var x;
run;
```

OR Which is the same:

```
/* SAS automatically outputs the data into WORK.DOLOOP */;
proc means data=WORK.DOLOOP;
var x;
run;
```

From the Log Window:

"NOTE: There were 100 observations read from the data set WORK.DOLOOP."

```
proc print  data=WORK.DOLOOP;
run;
```

The SAS System
Obs  i    x

1  1  3.1979
2  2  1.2737
3  3  5.6871

......................

......................

98 98  2.89315
99 99  -4.35282
100 100  -0.27217

NOTE: the new data set "try" contains only ONE obs "Mean".
If we want to see what is in the data set "try" use:

proc print data=try;
run;

Obs  _TYPE_  _FREQ_  Mean

     1     0    100     2.67189

or:

proc print data=try;
var Mean;
run;

Obs  Mean

   1     2.67189

Example: Verify Cody and Smith p. 356

=====================================================================

DATA DoLoop;
DO i = 1 TO 100;
x = 1 + 99* RANUNI(0);
OUTPUT;
END;
RUN;

Can apply PROCs to x:

proc print data=DoLoop; <--- The generated data are here!!!
var x;
run;

Get data from Unif(1,100)!!!

<table>
<thead>
<tr>
<th>Obs</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78.0351</td>
</tr>
<tr>
<td>2</td>
<td>85.2269</td>
</tr>
<tr>
<td>3</td>
<td>21.7456</td>
</tr>
<tr>
<td>4</td>
<td>15.7471</td>
</tr>
<tr>
<td>5</td>
<td>51.2031</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>95</td>
<td>19.3039</td>
</tr>
<tr>
<td>96</td>
<td>35.7787</td>
</tr>
<tr>
<td>97</td>
<td>10.1345</td>
</tr>
<tr>
<td>98</td>
<td>28.2734</td>
</tr>
<tr>
<td>99</td>
<td>2.7773</td>
</tr>
<tr>
<td>100</td>
<td>44.4092</td>
</tr>
</tbody>
</table>

a. Generation of RV’s
========================

SAS can generate RV’s from a given dist. Some examples:

-------------------------------------------------------------
ranbin(seed,n,p) Binomial distribution b(n,p)
runexp(seed) Exponential distribution Exp(1)

rannor(seed) Standard normal N(0,1)

ranpoi(seed,m) Poisson distribution Poisson(m)

ranuni(seed) Uniform distribution Unif(0,1)

Example: Generate N(0,1) and LN(0,1) data

DATA random;
DO i = 1 TO 100;
  r = RANNOR(0);
  y=exp(r); /* Lognormal data */;
  output;
END;
RUN;

/* OK!!! */
/* SAS automatically outputs the data into WORK.RANDOM */;
proc print data=WORK.RANDOM;
id i;
run;

r ~ N(0,1), y = exp(r) ~ LN(0,1)

The SAS System

<table>
<thead>
<tr>
<th>i</th>
<th>r</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.98409</td>
<td>7.27239</td>
</tr>
<tr>
<td>2</td>
<td>-0.62146</td>
<td>0.53716</td>
</tr>
<tr>
<td>3</td>
<td>0.75260</td>
<td>2.12251</td>
</tr>
<tr>
<td>4</td>
<td>-1.41778</td>
<td>0.24225</td>
</tr>
<tr>
<td>5</td>
<td>0.23668</td>
<td>1.26704</td>
</tr>
<tr>
<td>6</td>
<td>1.95064</td>
<td>7.03318</td>
</tr>
</tbody>
</table>

5
 DATA random;
 DO i = 1 TO 1000;
  r = RANNOR(0);
  y=exp(r);
  output;
 END;
 RUN;

 /* Nice histogram on GUI using GCHART */;
 proc gchart data=WORK.RANDOM;
  var r;
  run;

 /* Simpler using CHART */;
 proc chart data=WORK.RANDOM;
  var r;
  run;
/* Histogram of the lognormal data */;
proc chart data=WORK.RANDOM;
var y;
run;

......
3 2 2 1 1 0 0 0 1 1 2 2 3
. . . . . . . . . . . . . . . .
2 7 2 7 2 7 2 2 7 2 7 2 7 2
5 5 5 5 5 5 5 5 5 5 5 5 5 5

r Midpoint

Frequency

| **
600 + **
| **
| **
| **
400 + **
proc plot data=WORK.RANDOM;
plot y*r;
run;

Plot of y*r. Legend: A = 1 obs, B = 2 obs, etc.
b. Demonstration of LLN using simulated data

==============================================
1. Demonstration of LLN with iid exp(1) r.vs.

```r
options nodate nonumber;
data EXPO;
do sample=1 to 2; /*Number of runs*/;
do n=1 to 100; /*Number of obs in each run*/;
y = ranexp(7);
output;
end;
end;
run;

proc means data=EXPO;
output out=EXPLLN mean=Mean;
var y;
by sample;
run;
```

NOTE: 315 obs hidden.
Results from samples of size N from \( \text{Exp}(1) \)

Note: SD of \( \text{Exp}(1) \) is 1.

The MEANS Procedure

Analysis Variable : \( y \)

<table>
<thead>
<tr>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.9697695</td>
<td>0.9767435</td>
<td>0.0058798</td>
<td>4.5488584</td>
</tr>
<tr>
<td>100</td>
<td>0.9389630</td>
<td>0.8614748</td>
<td>0.0013313</td>
<td>4.1462119</td>
</tr>
<tr>
<td>10000</td>
<td>0.9986598</td>
<td>0.9980727</td>
<td>0.000056625</td>
<td>10.1128538</td>
</tr>
<tr>
<td>10000</td>
<td>1.0040726</td>
<td>1.0006146</td>
<td>0.000030292</td>
<td>9.6453046</td>
</tr>
<tr>
<td>1000000</td>
<td>1.0013987</td>
<td>1.0018283</td>
<td>4.2375178E-8</td>
<td>14.4117537</td>
</tr>
<tr>
<td>1000000</td>
<td>0.9991823</td>
<td>0.9975673</td>
<td>1.4072294E-6</td>
<td>15.3075459</td>
</tr>
</tbody>
</table>

2. Demonstration of LLN with iid \( \text{Unif}(0,1) \) r.vs.

```plaintext
options nodate nonumber;
data UNIF;
do sample=1 to 2; /*Number of runs*/;
do n=1 to 10000; /*Number of obs in each run*/;
y = RANUNI(0);
output;
end;
end;
run;

proc means data=UNIF;
output out=UNIFLLN mean=Mean;
```
var y;
by sample;
run;

Results from samples of size N from Unif(0,1)
==================================================

Note: SD of Unif(0,1) is sqrt(1/12)=0.2886751

The MEANS Procedure

Analysis Variable : y

<table>
<thead>
<tr>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.5385640</td>
<td>0.3179112</td>
<td>0.0013313</td>
<td>0.9980128</td>
</tr>
<tr>
<td>100</td>
<td>0.5622033</td>
<td>0.2852505</td>
<td>0.0246569</td>
<td>0.9936232</td>
</tr>
<tr>
<td>10000</td>
<td>0.5008272</td>
<td>0.2879409</td>
<td>0.000022463</td>
<td>0.9999637</td>
</tr>
<tr>
<td>10000</td>
<td>0.4980235</td>
<td>0.2852955</td>
<td>0.000035856</td>
<td>0.9998519</td>
</tr>
<tr>
<td>1000000</td>
<td>0.5000877</td>
<td>0.2886437</td>
<td>1.6298145E-7</td>
<td>0.9999996</td>
</tr>
<tr>
<td>1000000</td>
<td>0.5003632</td>
<td>0.2886714</td>
<td>4.2328611E-7</td>
<td>0.9999991</td>
</tr>
</tbody>
</table>

3. Demonstration of LLN with iid N(0,1) r.vs.

options nodate nonumber;
data NORM;
do sample=1 to 2; /*Number of runs*/;
do n=1 to 10000; /*Number of obs in each run*/;
y = RANNOR(0);
output;
end;
end;
run;
proc means data=NORM;
output out=NORMLLN mean=Mean;
var y;
by sample;
run;

Results from samples of size N from N(0,1)
=================================================================================
The MEANS Procedure

Analysis Variable : y

<table>
<thead>
<tr>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>-0.0277138</td>
<td>0.9218208</td>
<td>-2.2105632</td>
<td>2.3709443</td>
</tr>
<tr>
<td>100</td>
<td>-0.0957943</td>
<td>1.0360914</td>
<td>-2.9719010</td>
<td>2.3719339</td>
</tr>
<tr>
<td>10000</td>
<td>-0.0209077</td>
<td>1.0133153</td>
<td>-3.7308816</td>
<td>3.9232886</td>
</tr>
<tr>
<td>10000</td>
<td>-0.0081349</td>
<td>0.9902589</td>
<td>-3.3161692</td>
<td>4.1345405</td>
</tr>
<tr>
<td>1000000</td>
<td>-0.0011122</td>
<td>1.0002769</td>
<td>-4.6627020</td>
<td>5.0426949</td>
</tr>
<tr>
<td>1000000</td>
<td>-0.000215533</td>
<td>0.9997978</td>
<td>-4.8857179</td>
<td>4.8376764</td>
</tr>
</tbody>
</table>

C. Demonstration of CLT using simulated data
=================================================================================

TO Generate 10000 sampleS of size 100 from an exponential distribution and then generate a histogram of the sample means,’Y.

OPTION PS=35 LS=70;
options nodate nonumber;
data one;
do sample=1 to 10000;
do n=1 to 100;
y = ranexp(4637);
output;
end;
end;
run;

proc means data=one noprint;
output out=new mean=Mean;
var y;
by sample;
run;

proc gchart data=new;
vbar mean / space=0;
title 'Sampling distribution of ybar';
run;
quit;

Simpler to use PROC CHART not GCHART:

proc chart data=new;
vbar mean / space=0;
title 'Sampling distribution of ybar';
run;
quit;

Sampling distribution of ybar

Frequency

1000 + ********
| ********
| ********
| ********

13
Do Again with different seed.

OPTION PS=35 LS=70;

options nodate nonumber;
data TWO;
do sample=1 to 1000;
do n=1 to 100;
y = ranexp(2);
output;
end;
end;un;
proc means data=TWOPRINT noprint;
 output out=YDATA mean=Mean;
 var y;
 by sample;
 run;

Simpler to use PROC CHART not GCHART:

proc chart data=YDATA;
 vbar mean / space=0;
 title'Sampling distribution of ybar';
 run;
 quit;

Sampling distribution of ybar

<table>
<thead>
<tr>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>150 +</td>
</tr>
<tr>
<td>100 +</td>
</tr>
</tbody>
</table>

****
****

************

************

************

************

************

************

************

************

************
Better to use PROC SUMMARY to demonstrate CLT
===============================================================================

EX 1: Generate 500 samples of size 10 from Poisson(2), then consider the 500 sample sums and means, min, max, etc.

OPTION PS=35 LS=70;
options nodate nonumber;
data POISSON;
do sample=1 to 500; /*Number of runs*/;
do n=1 to 10; /*Number of obs in each run*/;
y = RANPOI(0,2); /* Seed=0, Poisson parameter lambda=m=2.*/
output;
end;
end;
run;

First look at the sample sums.

proc summary data=POISSON nway;
class sample;
var y;
output out=newdata sum=sumy;
run;

proc print data=newdata;
run;

OK! Get sums around 20 since 
\[ E(\text{sumy}) = E(X_1 + \ldots + X_{10}) = 20 \] for Poisson(2)!!

<table>
<thead>
<tr>
<th>Obs</th>
<th>sample</th>
<th><em>TYPE</em></th>
<th><em>FREQ</em></th>
<th>sumy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>498</td>
<td>498</td>
<td>1</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>499</td>
<td>499</td>
<td>1</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>500</td>
<td>500</td>
<td>1</td>
<td>10</td>
<td>18</td>
</tr>
</tbody>
</table>

proc chart data=newdata;
  vbar sumy / space=0;
title ‘Sampling distribution of sumy’;
run;

OK! Histogram is centered at 20!!! approx.

Sampling distribution of sumy

Frequency
Test normality with normal probability plot using PROC UNIVARIATE!!

/*normal, box, stem leaf plots using UNIVARIATE*/;
proc UNIVARIATE data=newdata normal plot;
  var sumy;
run;

The UNIVARIATE Procedure
Variable: sumy

Normal Probability Plot

32.5+ *  *
|    *  ****
|      ****
Also look at Box plot and stem and leaf. Not bad!!

<table>
<thead>
<tr>
<th>Histogram</th>
<th>#</th>
<th>Boxplot</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.5+++</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>.*</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>.*</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>.*</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>.***</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>.******</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>.*******</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>**********</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>***********</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>***********</td>
<td>28</td>
<td>-------</td>
</tr>
</tbody>
</table>
Also look at basic statistics:

Basic Statistical Measures

<table>
<thead>
<tr>
<th>Location</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>19.95000</td>
</tr>
<tr>
<td>Median</td>
<td>20.00000</td>
</tr>
<tr>
<td>Mode</td>
<td>18.00000</td>
</tr>
<tr>
<td>Std Deviation</td>
<td>4.28301</td>
</tr>
<tr>
<td>Variance</td>
<td>18.34419</td>
</tr>
<tr>
<td>Range</td>
<td>24.00000</td>
</tr>
<tr>
<td>Interquartile Range</td>
<td>6.00000</td>
</tr>
</tbody>
</table>

Now do the same with the 500 sample means. Recall each sample is of size 10 from Poisson(2).

```
proc summary data=POISSON nway;
class sample;
var y;
output out=newdata mean=meany; <- Use the same name "newdata"!!!
run;  
```

but we dont have to!!!
proc print data=newdata; <-- Get also _FREQ_
run;

proc print data=newdata; <-- This is good enough!!! Only sample means!!!
var meany;
run;

<table>
<thead>
<tr>
<th>Obs</th>
<th>meany</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.8</td>
</tr>
<tr>
<td>2</td>
<td>1.7</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
</tr>
<tr>
<td>4</td>
<td>2.4</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>2.8</td>
</tr>
<tr>
<td>7</td>
<td>2.0 &lt;-- OK!!! Since E(Xbar)=2!!!</td>
</tr>
<tr>
<td>8</td>
<td>1.8</td>
</tr>
<tr>
<td>9</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>2.0</td>
</tr>
<tr>
<td>11</td>
<td>2.2</td>
</tr>
<tr>
<td>12</td>
<td>1.5</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>497</td>
<td>3.2</td>
</tr>
<tr>
<td>498</td>
<td>1.9</td>
</tr>
<tr>
<td>499</td>
<td>2.0</td>
</tr>
<tr>
<td>500</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Now apply PROC UNIVARIATE to "var meany" in "data=newdata"!!!

/*normal, box, stem leaf plots*/;
proc UNIVARIATE data=newdata normal plot;
var meany;
run;

OK! Mean about 2, var about 2/10!!!
Basic Statistical Measures

<table>
<thead>
<tr>
<th>Location</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>1.995000</td>
</tr>
<tr>
<td>Median</td>
<td>2.000000</td>
</tr>
<tr>
<td>Mode</td>
<td>1.800000</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Box plot, normal prob plot, stem and leaf the same as above except for the constant. For example:

![Normal Probability Plot]

2.05+
|     |
| 3.25+
|     | * |
|     | * |
|     | **** |
|     | **** |
|     | ***** |
|     | ***** |
|     | ***** |
|     | **** |
|     | *** |
| 2.05+ | *** |
|     | **** |
proc chart data=newdata;
$vbar meany / space=0;
title'Sampling distribution of meany';
run;

Again get the same thing but centered at 2!!

Sampling distribution of meany

Frequency

| | ****
| | ****+
| | *****
| | ******
| | *******
| +++++
| ++

0.85++

+----------------------------------------------------------+
| -2 | -1 | 0 | +1 | +2 |
+----------------------------------------------------------+

23
Now do the same with the 500 sample maxima. Recall each sample is of size 10 from Poisson(2).

```plaintext
proc summary data=POISSON nway;
  class sample;
  var y;
  output out=newdata max=maxy; <- Use the same name "newdata"!!!
run;

proc chart data=newdata;
  vbar maxy / space=0;
  title 'Sampling distribution of MAX(Y)';
run;

Histogram of Maxy=Max(X_1,...,X_10) is far from normal!!!

Sampling distribution of MAX(Y)

Frequency

180 + *****
| *****
| *****
```
/normal, box, stem leaf plots for sample maxima/;
proc UNIVARIATE data=newdata normal plot;
var maxy;
run;

Basic Statistical Measures

<table>
<thead>
<tr>
<th>Location</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean 4.496000</td>
<td>Std Deviation 1.11196</td>
</tr>
<tr>
<td>Median 4.000000</td>
<td>Variance 1.23646</td>
</tr>
<tr>
<td>Mode 4.000000</td>
<td>Range 6.000000</td>
</tr>
<tr>
<td></td>
<td>Interquartile Range 1.00000</td>
</tr>
</tbody>
</table>
Histogram

8.25***
.*****
.**********
.**********
.**********
5.25*****************
.**********
.**********
.**********
.**********
.**********
2.25+++****
-------------------------
* may represent up to 4 counts

Normal Probability Plot

8.25+***
| **
| **************
| +++++
| **************
| +++++
|**************
| +++++
5.25+********
| +++++
|**************
| +++++
|**************
| +++++
2.25+++++++++
---------------------------------------
Get results about sums, means, maxima all at the same time

Use: "mean(y)=meany sum(y)=sumy max(y)=maxy" in
PROC SUMMARY. But also "mean=meany sum=sumy max=maxy" works!!!

OPTION PS=35 LS=70;
options nodate nonumber;
data POISSON;
do sample=1 to 500; /*Number of runs*/;
do n=1 to 10; /*Number of obs in each run*/;
y = RANPOI(0,2); <--- Seed=0, Poisson parameter lambda=m=2.
output;
end;
end;
run;

Get sums, means, maxima all at the same time

proc summary data=POISSON nway;
class sample;
var y;
output out=newdata mean(y)=meany sum(y)=sumy max(y)=maxy;
run;

proc UNIVARIATE data=newdata normal plot;
var meany sumy maxy;
run;

Will give all the results for meany sumy maxy.