ASK THE EXPERT

How Do I Use the Bootstrap Method in SAS®?

Rick Wicklin

Rick Wicklin

Distinguished Researcher in Computational Statistics, SAS

Rick is the principal developer of the SAS/IML matrix language. His areas of expertise include simulation, computational statistics, and statistical graphics. Rick is an author of two books, including Simulating Data with SAS. He writes a popular blog about statistical programming in SAS (http://blogs.sas.com/content/iml). Rick frequently presents talks, tutorials, and workshops at conferences.

THE ESSENTIAL GUIDE TO BOOTSTRAPPING IN SAS®

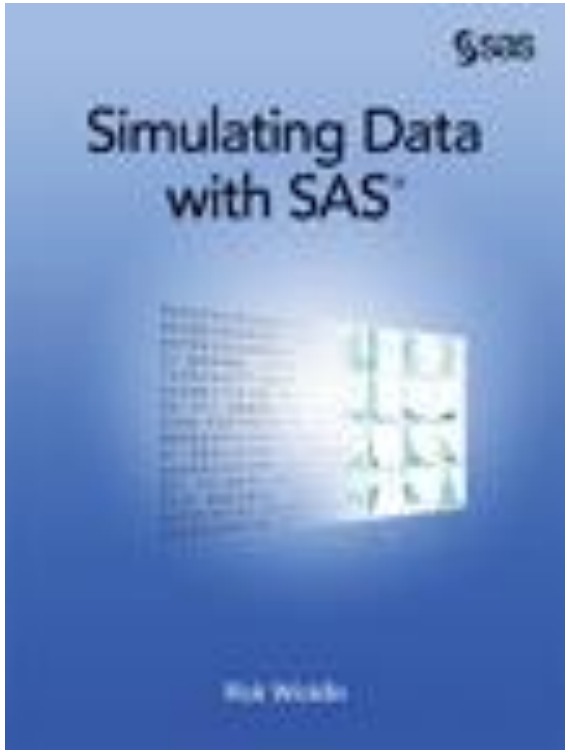
RICK WICKLIN, SAS
@RICKWICKLIN
blogs.sas.com/content/iml

ABSTRACT

The bootstrap method (Efron, 1979) is one of the most important innovations in statistics in the 20th century. This talk introduces the bootstrap method and discusses when it should be used. This example-driven presentation includes best practices for implementing bootstrap programs efficiently in SAS. An inefficient bootstrap program can take hours to run, whereas a well-written program can give you an answer in an instant. If you want to learn more about the bootstrap method and prefer "instants" to "hours," this talk is for you!

INTRODUCTION

- The bootstrap method (Efron, 1979) is one of the most important innovations in statistics in the 20th century
- This talk includes tips for implementing bootstrap methods efficiently in SAS
- Some tips are from *Simulating Data with SAS* (Wicklin 2013, Chap 15)
- Others from The DO Loop blog:
  Search the internet for
  Essential Guide Bootstrapping SAS



OUTLINE OF THIS PRESENTATION

- What is the bootstrap method?
- How to bootstrap a univariate statistic in SAS?
- Efficiency and details of the implementation
- How to bootstrap regression estimates in SAS?
- SAS procedures that support built-in bootstrapping

ESSENTIAL BOOTSTRAPPING

What is the bootstrap method?

THE SAMPLING DISTRIBUTION OF A STATISTIC

- Main assumption: Data are a random sample of size N from a population distribution
- A statistic is a sample estimate of a population parameter
- A different random sample gives a different estimate
- The distribution of a statistic over **all** possible samples of size *N* is the *sampling distribution*
- The bootstrap method is an *inferential* technique that estimates the sampling distribution by using **some** samples of size *N*
  - Standard error is the standard deviation of the sampling distribution
  - Confidence intervals (CIs) correspond to quantiles such as 5^{th} to 95^{th}

WHEN SHOULD YOU USE THE BOOTSTRAP METHOD?

Use the bootstrap method when

- You want to estimate the precision of a point estimate
  - Standard error or confidence interval
- The sampling distribution is not known or is known only asymptotically (large N)
  - The sample size is small
  - The sample distribution is far from normal

THE EMPIRICAL DISTRIBUTION REPLACES THE POPULATION

Why is it called the “bootstrap”?

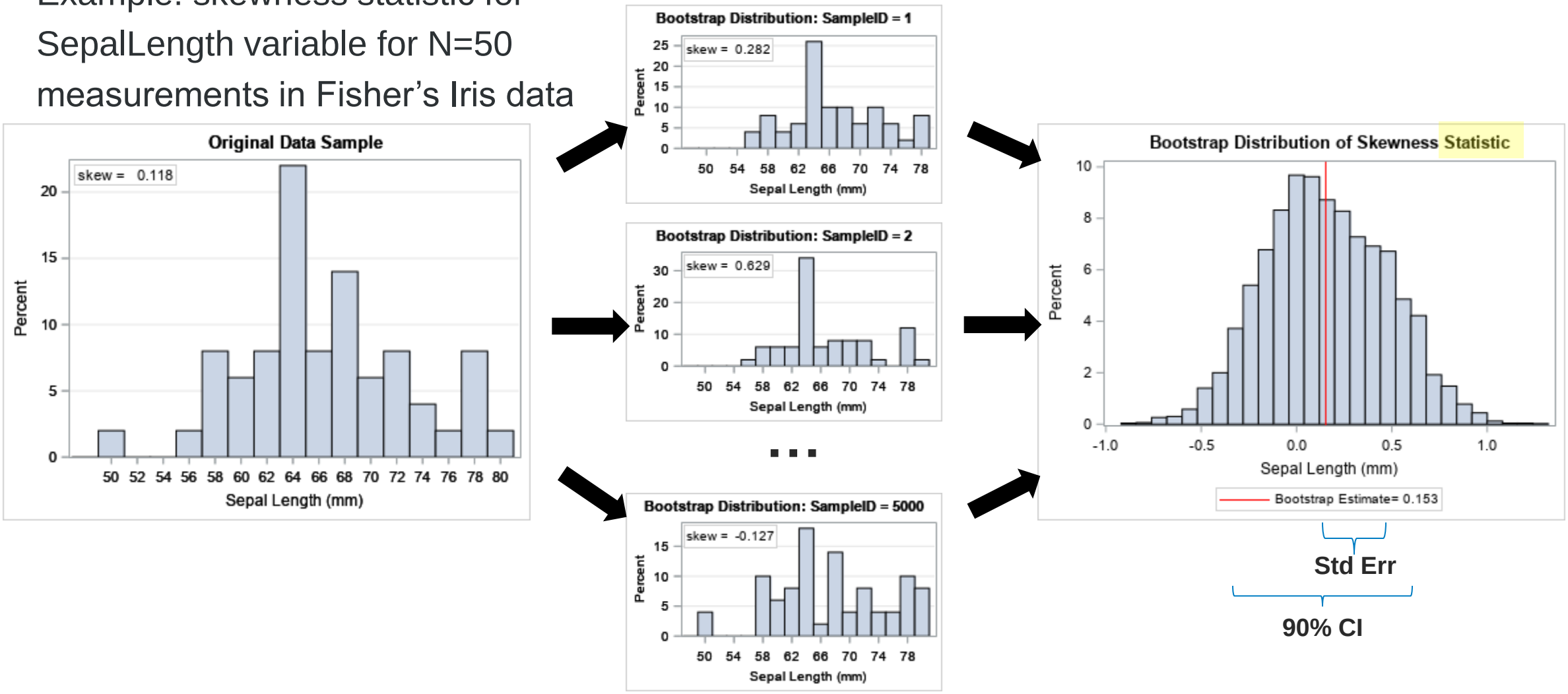
- If you fit a *model* the data, you can use simulation to create random samples from the parametric model. Sometimes called *parametric bootstrap*, but that is a misnomer.
- The bootstrap replaces the unknown cumulative distribution function (CDF) with the known empirical CDF
- The bootstrap method is a *nonparametric* method, which “simulates” from the empirical CDF
  - You *sample with replacement* to obtain one of the N^N re-samples
  - The method approximates the population distribution by using only ONE sample!

THE BOOTSTRAP METHOD

- Repeat *B* times (*B* large):
  - Draw *N* observations with replacement from the data. This is a *bootstrap sample*, also called the *resample*.
  - Compute the statistic on each bootstrap sample.
- Union of statistics is the *bootstrap distribution* (BD), which approximates the sampling distribution
  - *Mean* of BD is the bootstrap estimate
  - *Standard deviation* of BD is the bootstrap standard error
  - *Percentiles* of BD estimate a CI

THE BOOTSTRAP METHOD IN PICTURES

Example: skewness statistic for SepalLength variable for N=50 measurements in Fisher's Iris data



DRAWBACKS OF THE BOOTSTRAP METHOD

- Assumes the original sample is representative of the population
- Never uses values that are not in original sample (175 cm male)
- Rare events either ignored (if not in sample) or have undue influence (if in sample)
- Not useful for some statistics
  - Extreme quantiles, maximum, and minimum
- Bootstrap distribution might be non-smooth
  - Median of small sample
  - Can use variation called *smooth bootstrap*

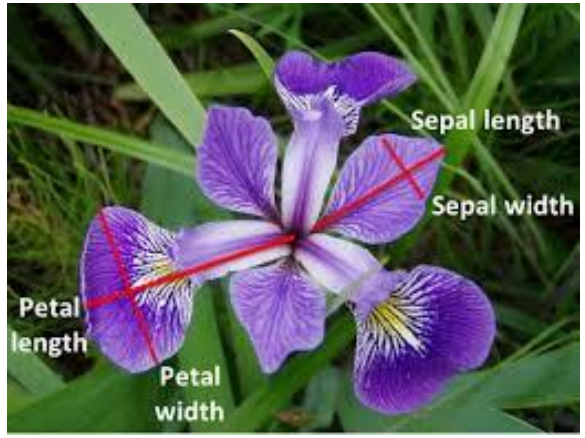
ESSENTIAL BOOTSTRAPPING

How to bootstrap a univariate statistic in SAS?

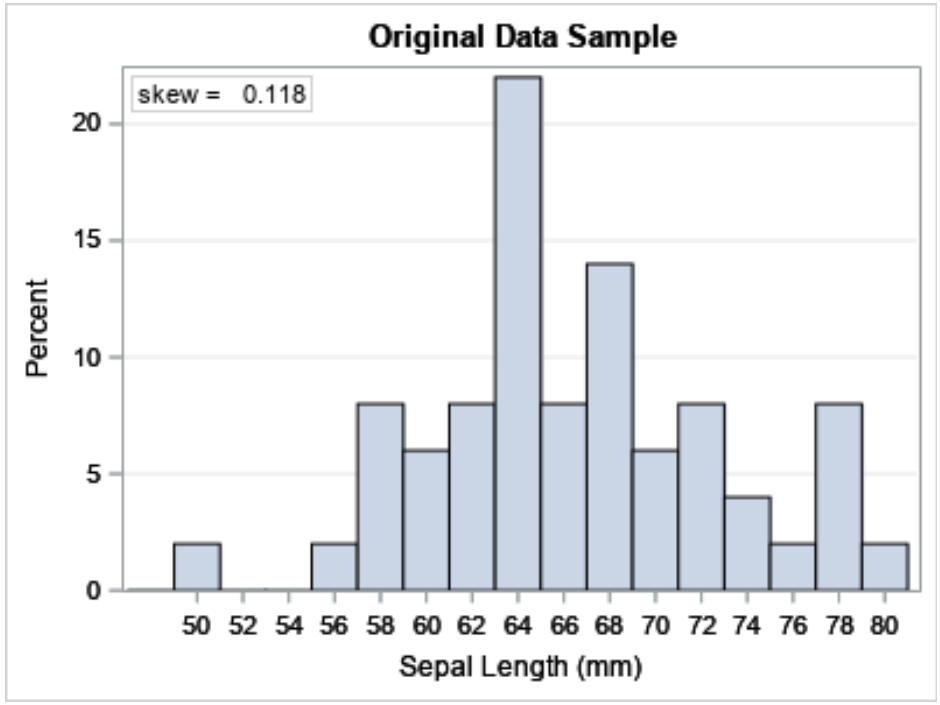
HOW TO BOOTSTRAP IN SAS

1. Compute the statistic for the original data (the Sample data set)
2. Resample (with replacement) B times to form B bootstrap samples
   - Put all samples into ONE data set: the BootSamp data set
   - How you resample might depend on the analysis
3. Compute the statistic on each bootstrap sample
   - BY-group processing
   - Put results in the BootStats data set
   - Avoid macro loops
4. Use bootstrap distribution to estimate standard error, confidence intervals, and evidence to reject null hypothesis

COMPUTE THE STATISTIC FOR THE ORIGINAL DATA



- **Data:** SepalLength variable in Fisher's Iris data, N=50 observations
- **Statistic:** Skewness
- **Goals:** Standard error, 90% CI for skewness, H_0: Skewness = 0




```
/* Sample size N = 50 */
data Sample(keep=x);
   set Sashelp.Iris(rename=(SepalLength=x));
   where Species="Virginica";
run;

/* compute statistic on data */
proc means data=Sample nolabels Skew;
   var x;
run;
```

Skewness
0.1180151

WHAT IS THE DISTRIBUTION OF THE SKEWNESS STATISTIC?

The sample skewness is 0.118. Can we infer that the skewness of the population is different from 0?

The sampling distribution of the skewness statistic is not known:

➔ Use the bootstrap!

GENERATE RESAMPLES: SAMPLE WITH REPLACEMENT

Three main techniques for sampling with replacement

1. DATA step: use the POINT= option
2. PROC SURVEYSELECT: use the METHOD=URS option to sample with replacement
3. PROC IML: use the SAMPLE function

PROC SURVEYSELECT: SAMPLE WITH REPLACEMENT

Advantages of PROC SURVEYSELECT

- Requires no programming.
- About as fast as the DATA step
- Supports many sampling schemes
- Optional: create a frequency variable, which decreases the number of rows in the bootstrap samples

GENERATE MANY BOOTSTRAP SAMPLES

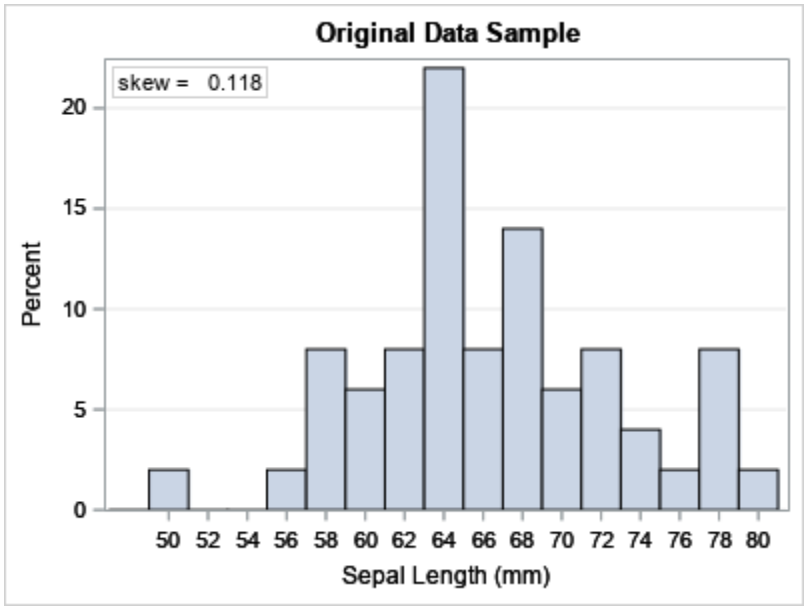
```
%let NumSamples = 5000;       /* number of bootstrap resamples */
/* 2. Generate many bootstrap samples */
proc surveyselect data=Sample NOPRINT seed=1 /* SEED for RNG */
     method=urs               /* URS = resample with replacement */
     samprate=1               /* each bootstrap sample has N obs */
     OUTHITS                  /* do not use a frequency var */
     reps=&NumSamples         /* generate NumSamples samples */
     out=BootSamp(rename=(Replicate=SampleID));
run;
```

When you use the OUTHITS option, the BootSamp data set contains N x NumSamples rows:

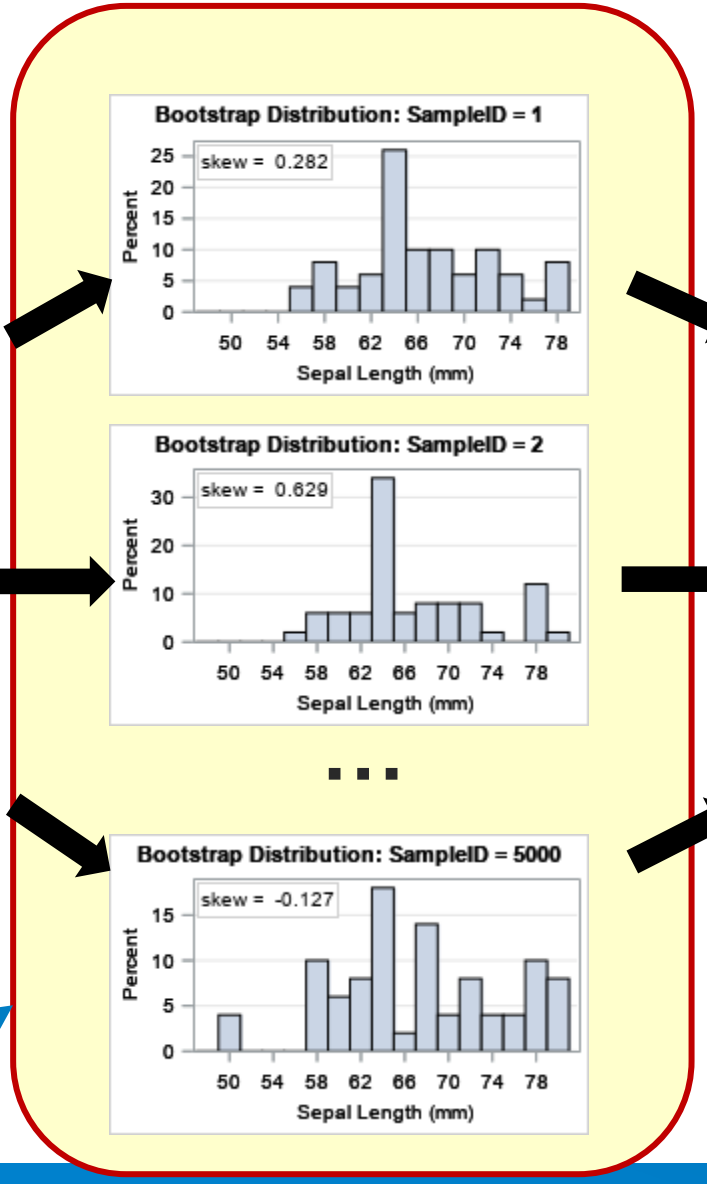
- N observations for SampleID = 1
- N observations for SampleID = 2
- And so on

ESSENTIAL BOOTSTRAPPING

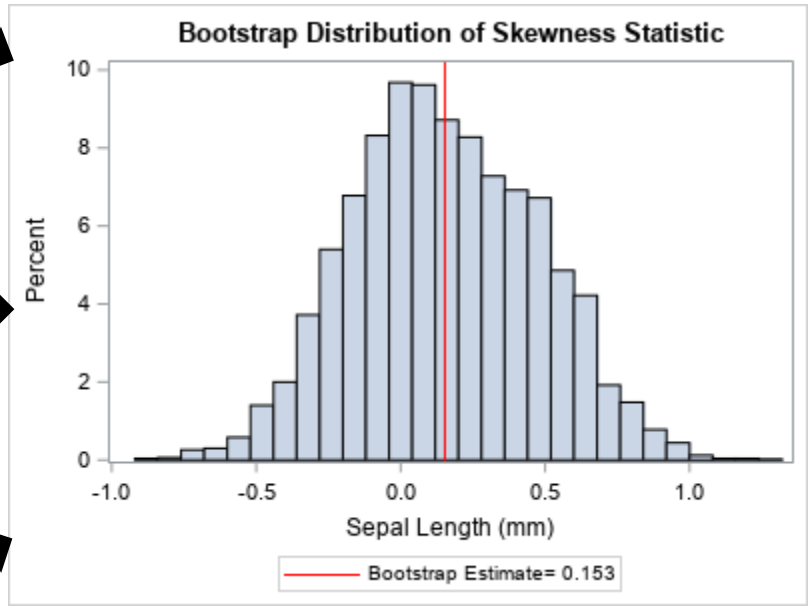
THE BOOTSTRAP METHOD IN PICTURES



Sample data set



BootSamp data set



Std Err

90% CI

BootStats data set

COMPUTE THE STATISTIC FOR EACH BOOTSTRAP SAMPLE

```
/* 3. Compute the statistic for each
      bootstrap sample */
proc means data=BootSamp NOPRINT;
   by SampleID;
   var x;
   output out=BootStats skew=Skewness;  /* bootstrap distribution */
run;
```

Obs	SampleID	Skewness
1	1	0.28206
2	2	0.62902
3	3	-0.08908
4	4	0.59312
5	5	0.55024
6	6	0.50150

4994	4994	-0.20187
4995	4995	-0.21925
4996	4996	-0.39973
4997	4997	0.36382
4998	4998	-0.20712
4999	4999	0.23525
5000	5000	-0.12704

ESTIMATE STANDARD ERROR AND CONFIDENCE INTERVAL

```
proc means data=BootStats nolabels N Mean StdDev P5 P95;
   var Skewness;
run;
```

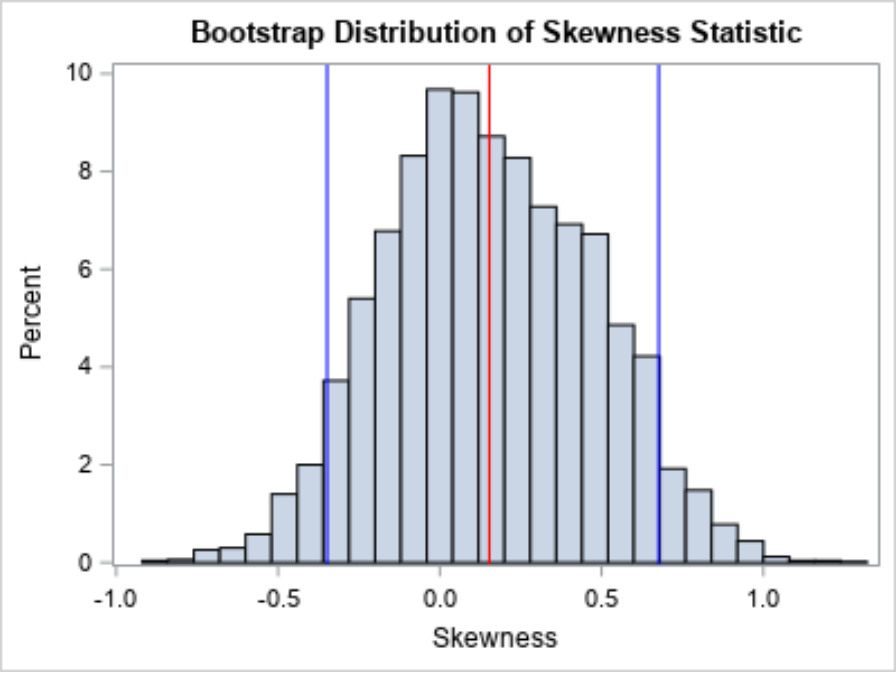
Analysis Variable : Skewness				
N	Mean	Std Dev	5th Pctl	95th Pctl
5000	0.1534985	0.3206183	-0.3482734	0.6766532

Bootstrap estimates:

- Skewness = 0.153 (compare with 0.118 for sample)
- Standard error = 0.321
- 90% CI = [-0.348, 0.677] ➔ Do not reject H_0. Assuming skewness = 0 is consistent with the data.

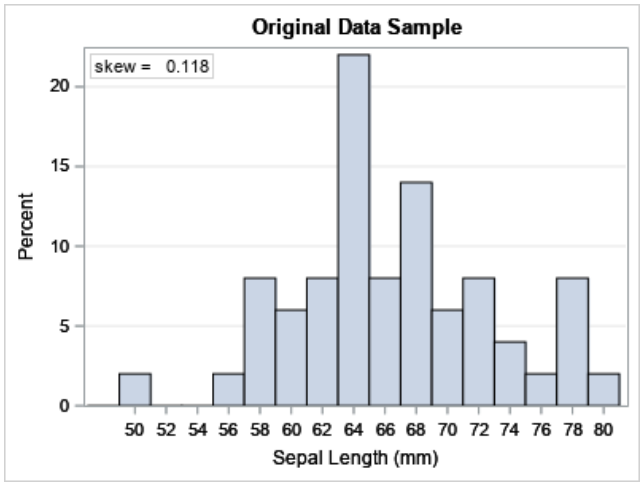
VISUALIZE THE SAMPLING DISTRIBUTION

```
title "Bootstrap Distribution of Skewness Statistic";
proc sgplot data=BootStats;
   histogram Skewness;
   refline 0.153 / axis=x lineattrs=(color=red);
   refline -0.348 0.677 / axis=x  lineattrs=(color=blue);
run;
```



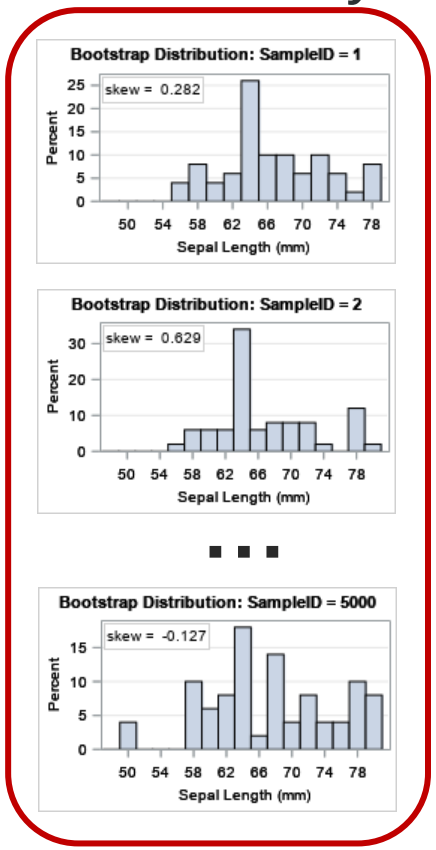
REVIEW: BOOTSTRAPPING IN SAS

Data ➔ Resample ➔ By-Group analysis ➔ Analyze bootstrap distribution



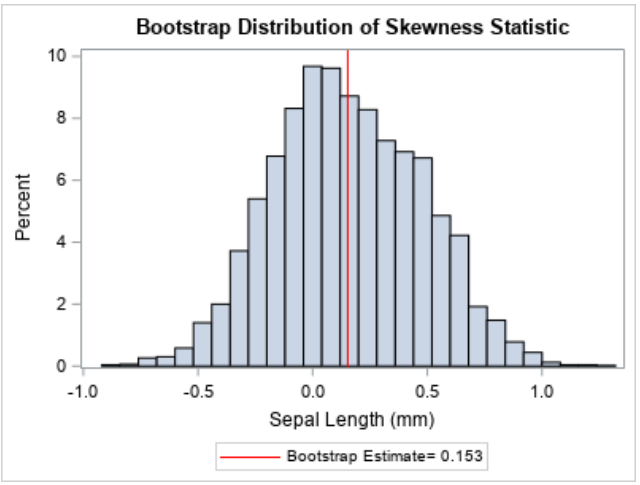
Sample data set

PROC SURVEY-SELECT



BootSamp data set

Compute statistics on each resample



BootStats data set

ESSENTIAL BOOTSTRAPPING

Efficiency and details of the implementation

TIPS FOR EFFICIENT SAS PROGRAMMING

Every program in this presentation runs in less than 1 second!

Observe these SAS **efficiency** programming tips:

✓ 1. Put all resamples in one data set
2. Use BY-group analysis to analyze all resamples
3. Suppress output during BY-group analysis
4. Avoid writing macro loops
5. To get smaller data set, omit the OUTHITS option in PROC SURVEYSELECT

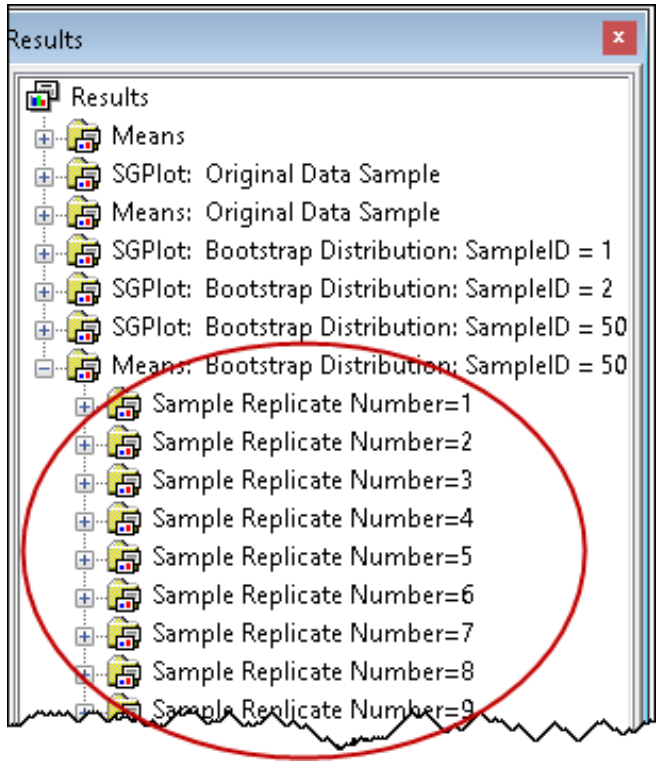
USE THE BY STATEMENT TO ANALYZE EACH SAMPLE

```
/* Use BY-group analysis to analyze resamples.
   Suppress output during BY-group analysis */
proc means data=BootSamp NOPRINT;
   BY SampleID;
   var x;
   output out=BootStats skew=Skewness;
run;
```

Obs	SampleID	Skewness
1	1	0.28206
2	2	0.62902
3	3	-0.08908
4	4	0.59312
5	5	0.55024
6		150

SUPPRESS OUTPUT DURING BY-GROUP ANALYSIS

- SAS procedures can create a LOT of output!
- For simple computations, it takes longer to print 5,000 tables than it does to compute them
- Suppress display of intermediate graphs and tables
  - About 50 procedures support the NOPRINT option
  - Use ODS to suppress output in other procedures
- Any unnecessary output will slow the bootstrap analysis
  - NOTES can fill up the SAS Log
  - ODS Results window makes a node in a tree for every BY group



SAS MACROS FOR CONTROLLING OUTPUT

```
%macro ODSOff();     /* call prior to BY-group processing */
   ods graphics off;
   ods exclude all; /* all open destinations */
   ods noresults;   /* no updates to tree view */
   options nonotes; /* sometimes useful */
%mend;


%macro ODSOn();      /* call after BY-group processing */
   ods graphics on;
   ods exclude none;
   ods results;
   options notes;
%mend;
```

TEMPLATE FOR BOOTSTRAP

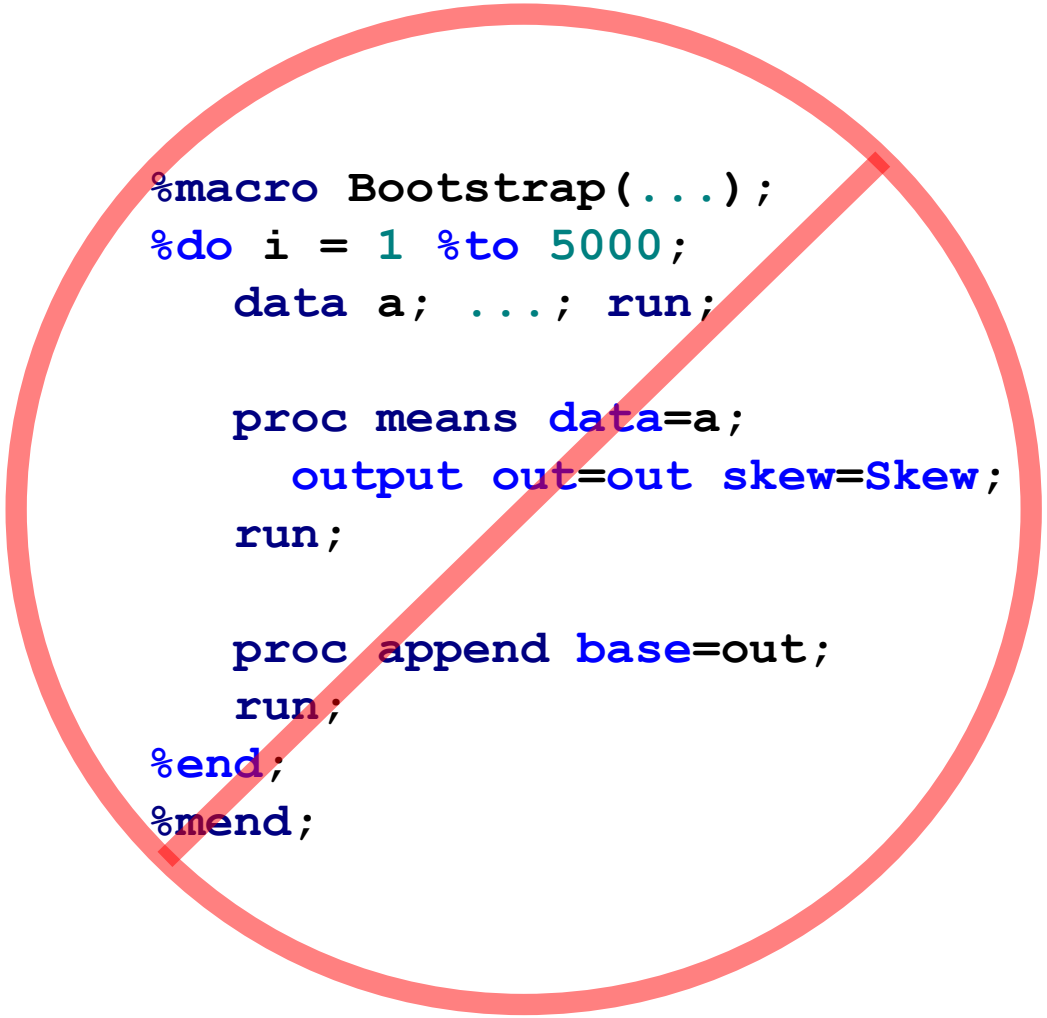
```
proc surveyselect data=Sample NOPRINT seed=1
   method=urs samprate=1 reps=&NumSamples outhits /* Optional */
   out=BootSamp(rename=(Replicate=SampleID));
run;

%ODSOff  /* or use NOPRINT option in procedure */
proc WhatEver data=BootSamp;
   by SampleID; /* Optional: FREQ NumberHits; */
   /* generate statistic for each resample */
   output out=BootStats ...;
run;
%ODSOn

/* use bootstrap distribution of statistic to answer
   questions about CIs, hypothesis tests, and more */
proc means data=BootStats;   /* or PROC UNIVARIATE */
   var Stats;
run;
```

AVOID WRITING MACRO LOOPS

Do NOT write a SAS macro that creates and analyzes one resample at a time!



```
%macro Bootstrap(...);
%do i = 1 %to 5000;
   data a; ...; run;

   proc means data=a;
     output out=out skew=Skew;
   run;

   proc append base=out;
   run;
%end;
%mend;
```

Novikov (2003) compares times for his application:
Macro > 7 minutes
BY group < 5 secs

➔ 100x faster!

WHY AVOID MACRO LOOPS?

- Each PROC or DATA step call requires overhead
  - Start procedure, parse statements, load data,…
  - [*Do computation*]
  - Close data, exit procedure
- The macro code repeats these steps 5,000 times; the overhead costs **large** relative to the computation
- In contrast, BY group processing uses *one* call to generate the resamples and *one* call to analyze the statistics; overhead costs **small** relative to the computation

THE OUTHITS OPTION IN PROC SURVEYSELECT

- The bootstrap method samples N observations *with replacement* from the data
- The average bootstrap sample contains 63.2% of the observations
  - Example: Data are {1,2,3,4,5,6,7,8,9,10}. A possible bootstrap sample is {1,2,2,2,5,5,6,7,8,8}, which contains 60% of values.
- If the analysis procedure supports a FREQ statement:
  - Omit the **OUTHITS** option to get a BootSamp data set that has 37% fewer rows
  - Add

```
FREQ NumberHits;
```

    to the analysis procedure

OMIT THE OUTHITS OPTION (CONTINUED)

```
/* OUTHITS ==> suppress frequency var */
proc surveyselect
   seed=&seed ... OUTHITS;
run;

/* create a frequency var */
proc surveyselect
   seed=&seed ...;
run;

/* Use BY-group analysis */
proc means data=BootSamp noprint;
   freq NumberHits;
   by SampleID;
   var x;
   output out=BootStats skew=Skewness;
run;
```

Obs	SampleID	x	NumberHits
1	1	69	2
2	1	65	1
3	1	58	1
4	1	68	2
5	1	76	1
6	1	59	1
7	1	64	3

Obs	SampleID	x
1	1	69
2	1	69
3	1	65
4	1	58
5	1	68
6	1	68
7	1	76
8	1	59
9	1	64
10	1	64
11	1	64

DETAILS OF SAS IMPLEMENTATION

1. Results depend on the random number stream
2. How to get a bootstrap p-value for a hypothesis test
3. Bootstrap in the SAS/IML language
4. Alternatives to percentile confidence intervals

DEPENDENCE ON THE RANDOM NUMBER STREAM

- You perform **a** bootstrap analysis, not **THE** bootstrap analysis
- The resamples depend on the random number stream, which is set by the SEED= option in PROC SURVEYSELECT
  - DATA step uses CALL STREAMINIT to set the seed
  - PROC IML uses CALL RANDSEED to set the seed
- All numbers will change if you change the seed
- Decisions (reject/fail to reject) should rarely change

Bootstrap Analysis of Skewness
SEED = 1

The MEANS Procedure

Analysis Variable : Skewness				
N	Mean	Std Dev	5th Pctl	95th Pctl
5000	0.1534985	0.3206183	-0.3482734	0.6766532

Bootstrap Analysis of Skewness
SEED = 321

The MEANS Procedure

Analysis Variable : Skewness				
N	Mean	Std Dev	5th Pctl	95th Pctl
5000	0.1580319	0.3233697	-0.3581482	0.6849739

(TWO-SIDED) P-VALUES FOR THE NULL HYPOTHESIS

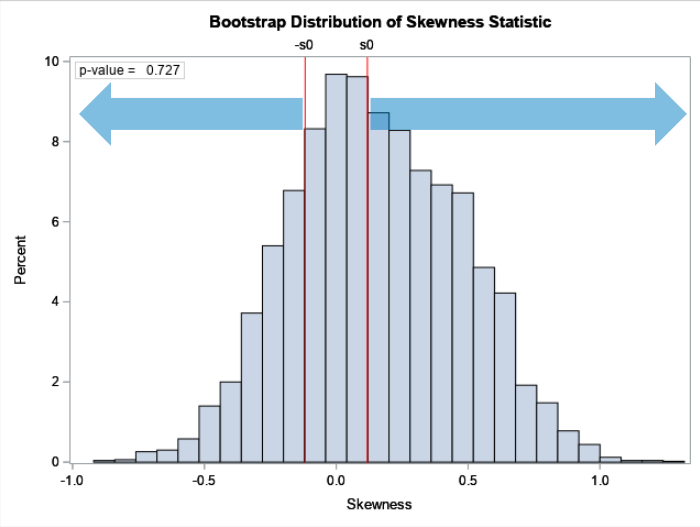
H_0: Skewness = 0.

A bootstrap p-value is the proportion of bootstrap statistics that are more extreme than the observed statistic:

$$p = \Sigma_{i=1}^{B} I(|s_i| > |s_0|)/B$$

or Davidson-Hinkley correction

$$p = (1 + \Sigma_{i=1}^{B} I(|s_i| > |s_0|))/(B+1)$$




```
%let Skew0 = 0.1180151;
proc iml;
use BootStats; read all var "Skewness";
/* Two-sided p-value for H0: Skewness = 0 */
B = nrow(Skewness);
s0 = abs(&Skew0);
p = sum(abs(Skewness) >= s0) / B;
pDH = (1 + sum(abs(Skewness) >= s0)) / (B+1);
print p pDH;
```

p	pDH
0.727	0.7270546

For large B, the two estimates are very close

BOOTSTRAP IN SAS/IML LANGUAGE

```
proc iml;
start EvalStat(M);          /* Function to eval stat for each col of a matrix */
   return skewness(M);      /* <== put computation here */
finish;
call randseed(1); alpha=0.1; B=5000; /* B = num bootstrap samples */
use Sample; read all var "x"; close; /* read data into x */

s0 = EvalStat(x);                    /* 1. observed stat */
s = sample(x, B // nrow(x));         /* 2. sample WR: s is N x B matrix   */
bStat = T( EvalStat(s) );            /* 3. compute stat for each column   */
bootEst = mean(bStat);               /* 4. summarize bootstrap distrib:   */
SE = std(bStat);                     /*    mean, standard deviation,      */
call qntl(CI, bStat, alpha/2 || 1-alpha/2);    /*   general percentile CI */
```

Bootstrap Analysis in SAS/IML

Obs	BootEst	StdErr	LowerCL	UpperCL
0.1180	0.1511	0.3192	-0.3611	0.6843

Bootstrap estimates are similar. Results differ only because random samples are different.

BETTER CONFIDENCE INTERVALS

- The percentile CI is first-order accurate
  - Does not use estimate from original data
  - Does not adjust for skewness of the bootstrap distribution
- The bias-corrected and accelerated interval (*BCa interval*) is second-order accurate
  - Estimate bias-correction parameter, z_0, from difference between sample and bootstrap estimates
  - Estimate acceleration parameter, a, based on the skewness
  - Obtain adjusted percentiles to use for CI
    - Ex: Instead of using 5th and 95th quantiles for 90% CI, might use 13th and 99th for the CI
- Details

  Wicklin, R (2017) “The bias-corrected and accelerated (BCa) bootstrap interval”
  https://blogs.sas.com/content/iml/2017/07/12/bootstrap-bca-interval.html

ESSENTIAL BOOTSTRAPPING

How to bootstrap regression estimates in SAS?

BOOTSTRAP REGRESSION ESTIMATES

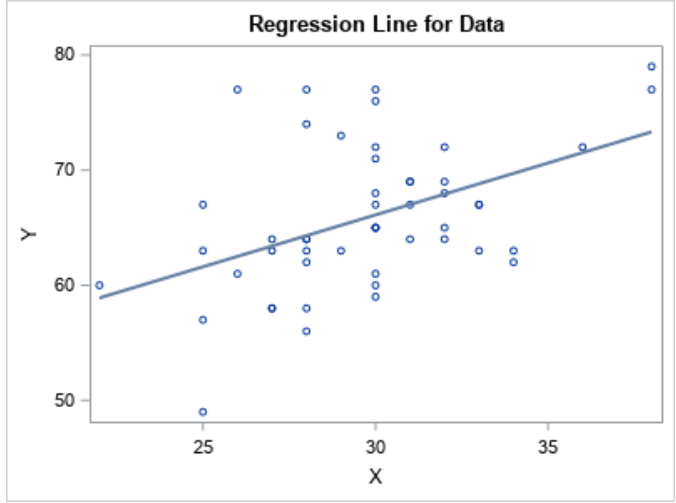
- The multivariate bootstrap is similar to the univariate bootstrap
- Regression models:
  - Case resampling: Randomly select observations with replacement
  - Residual resampling (semi-parametric): Predict $\widehat{Y}_i = f(x_i)$. Form residuals $r_i = Y_i - \widehat{Y}_i$. Bootstrap *residuals* and form bootstrap samples $\widehat{Y}_i + r_k$ where r_k is random residual.
  - Block bootstrap: For time-series and repeated-measures models with correlated errors

BOOTSTRAP REGRESSION EXAMPLE

You want to bootstrap the regression coefficients. Asymptotically, $\hat{\beta} \sim$ MVN

```
data sample(keep=x y);          /* rename vars */
  set Sashelp.Iris(rename=(SepalLength=Y SepalWidth=X));
  where Species="Virginica";
run;

/* 1. compute statistics on the original data (N=50) */
proc reg data=Sample plots(only)=FitPlot(noclm nocli);
   model Y = X / CLB covb;   /* original estimates */
quit;
```



Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	95% Confidence Limits	
Intercept	Intercept	1	39.06836	7.57061	23.84664	54.29009
X		1	0.90153	0.25311	0.39263	1.41044

Covariance of Estimates

Variable	Intercept	X
Intercept	57.314064817	-1.905215711
X	-1.905215711	0.0640623978

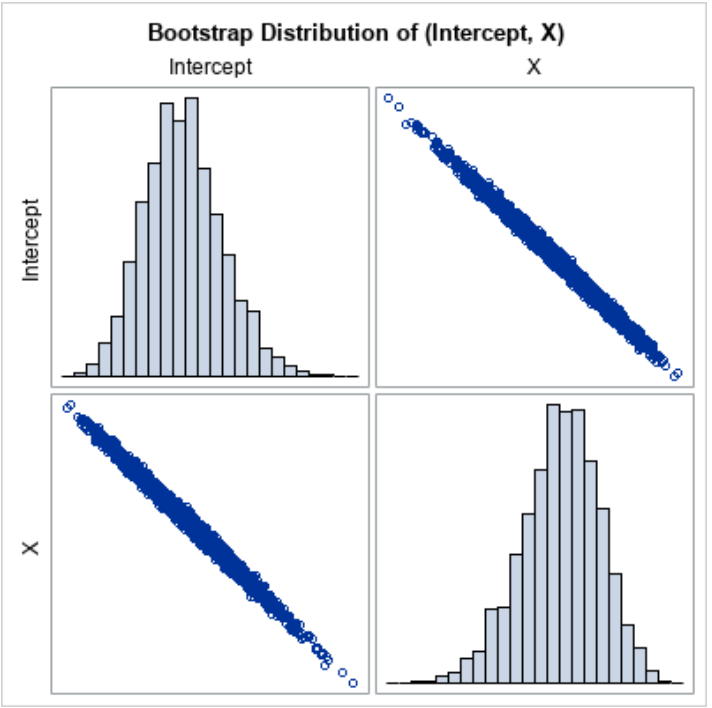
BOOTSTRAP REGRESSION EXAMPLE

```
/* 2. Generate many bootstrap samples by using PROC SURVEYSELECT */
%let NumSamples = 5000;          /* number of bootstrap resamples */
proc surveyselect data=sample NOPRINT seed=1
   method=urs samprate=1 reps=&NumSamples /* no OUTHITS */
   out=BootCases(rename=(Replicate=SampleID));
run;

/* 3. Compute the statistic for each bootstrap sample */
proc reg data=BootCases outest=PEBoot noprint;
   freq NumberHits;
   by SampleID;
   model Y = X;
quit;
```

BOOTSTRAP REGRESSION RESULTS

```
/* 4. Visualize bootstrap distribution */
proc means data=PEBoot N Mean Std P5 P95;
   var Intercept X;
run;
proc corr data=PEBoot cov vardef=N
   plots(maxpoints=none)=matrix(histogram);
   var Intercept X;
run;
```



Variable	N	Mean	Std Dev	5th Pctl	95th Pctl
Intercept	5000	39.5427779	7.9242596	27.2820239	53.4261215
X	5000	0.8855444	0.2568972	0.4321832	1.2805875

Covariance Matrix, DF = 5000			
		Intercept	X
Intercept	Intercept	62.78133221	-2.02605471
X		-2.02605471	0.06598298

ESSENTIAL BOOTSTRAPPING

SAS procedures that support built-in bootstrapping

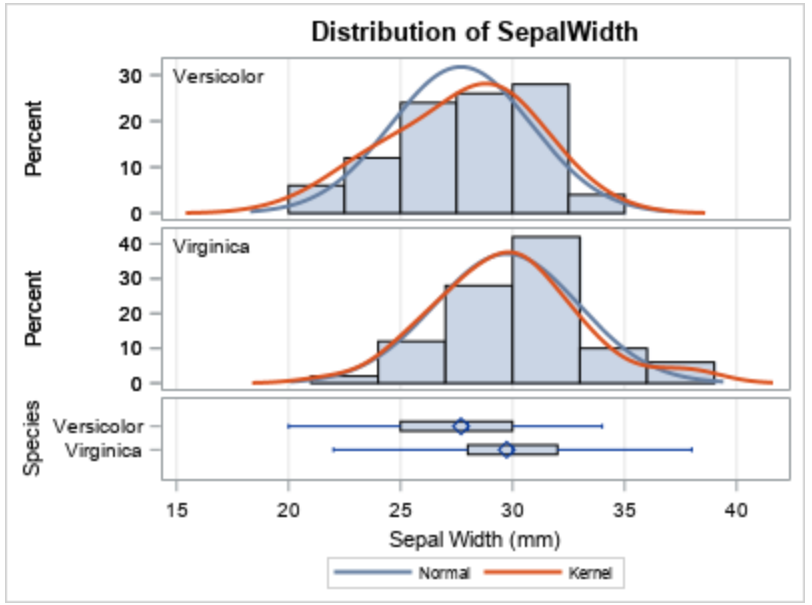
SAS/STAT PROCEDURES THAT SUPPORT BOOTSTRAP ESTIMATES

- Many confidence intervals are based on distributional assumptions
- Bootstrap estimates are distribution-free!
- The following procedures support bootstrap estimates:
  - PROC CAUSALMED and CAUSALTRT: standard errors and confidence intervals
  - PROC MULTTEST: p-values
  - PROC NLIN: confidence intervals for parameters and the covariance of the parameter estimates
  - PROC QUANTREG: regression quantiles
  - The SURVEY* procedures: variance
  - PROC TTEST: standard errors, bias, and confidence intervals

PROC TTEST EXAMPLE

- PROC TTEST supports the BOOTSTRAP statement
- Can run a bootstrap analysis of the difference of means:
  - Is the mean of the Species='Versicolor' group different from the mean of the Species='Virginica' group?

Data might not be normal!




```
/* BOOTSTRAP stmt in PROC TTEST */
proc ttest data=sashelp.iris;
class Species;
where  Species in ('Versicolor' 'Virginica');
var SepalWidth;
bootstrap / seed=123 nsamples=5000 bootci=percentile;
          /* or BOOTCI=BC */
run;
```

OUTPUT FROM PROC TTEST BOOTSTRAP STATEMENT

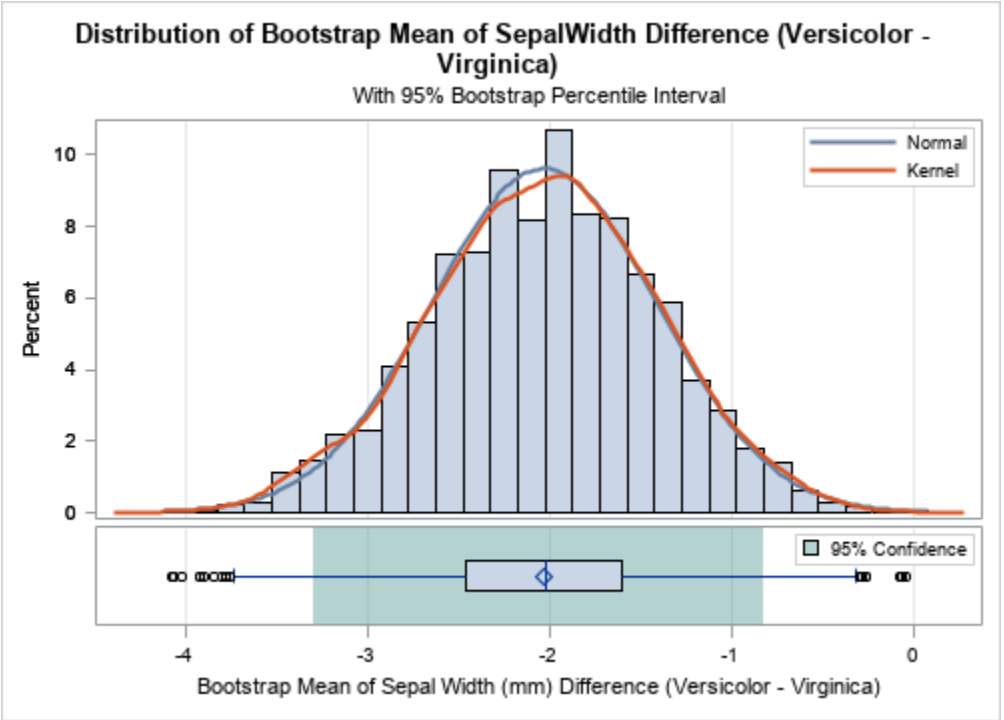
Species	Method	Mean	95% CL Mean		Std Dev
Versicolor		27.7000	26.8082	28.5918	3.1380
Virginica		29.7400	28.8235	30.6565	3.2250
Diff (1-2)	Pooled	-2.0400	-3.3028	-0.7772	3.1818

Classical parametric *t* test:
Conclusion: group means are different.
Assumes *normality* of data and *equal* variances.

Bootstrap Statistics and Confidence Limits						
Species	Method	Parameter	Std Error	Bias	95% CL	
Diff (1-2)		Mean	0.6221	0.00752	-3.3000	-0.8200

Bootstrap *t* test:
Conclusion: group means are different.
No parametric assumptions.
Reports Bias = Observed – Bootstrap estimate

Graph produced automatically! ➔



SUMMARY

The bootstrap method **estimates a standard error** for a statistic **(or CI)**

- Use when the sampling distribution of a statistic is unknown
- Efficiency matters! Use the tips:
  - Use PROC SURVEYSELECT and METHOD=URS to generate bootstrap samples
  - Use BY statement to analyze bootstrap samples and create a bootstrap distribution
- The results depend (slightly) on the random number stream
- The SAS/IML language enables compact implementation of univariate and multivariate bootstrapping
- Several SAS/STAT procedures support built-in bootstrap analyses

FURTHER READING

- Theory and practice:
  - Chernick, M. (1999) *Bootstrap methods: A practitioner's guide*
  - Davison, A.C. and Hinkley, D.V. (1997) *Bootstrap methods and their application*
- Implementation in SAS: *The DO Loop* blog
  - http://blogs.sas.com/content/iml
  - Click on **Bootstrap and Resampling** in the word cloud to see articles
  - "The Essential Guide to Bootstrapping in SAS" links to more than 30 blog posts that demonstrate bootstrapping in SAS

TWITTER: @RICKWICKLIN
LINKEDIN: WWW.LINKEDIN.COM/IN/RICKWICKLIN/
BLOG: blogs.sas.com/content/iml

Explore Helpful Resources

Ask the Expert
View other user webinars that provide insights into using SAS products to make your job easier.

FREE Training
Learn from home – free for 30 days. Get software labs to practice and online support if needed.

SAS Support Communities
Ask questions, get answers and share insights with SAS users.

SAS Analytics Explorers
An exclusive platform to collaborate, learn and share your expertise. Gain access to a diverse network to advance your career. Special rewards and recognition exclusively for SAS users.

SAS Users YouTube Channel
A plethora of videos on hundreds of topics, just for SAS users.

Newsletters
Get the latest SAS news plus tips, tricks and more.

Users Groups
Meet local SAS users, network and exchange ideas – virtually.

SAS Profile
If you haven't already done so, create your SAS Profile to access free training, SAS Support Communities, technical support, software downloads, newsletters and more.

Thank you

for joining us for
this SAS webinar