PROBLEMA VALUTARE L’ACCURATEZZA DELLA STIMA BOOTSTRAP DELL’INDICE DI SKEWNESS DELLA VARIABILE VEC

data sample(keep=x);  
 set scmm.score( rename=(vec=x));  
run;  
   
/\* 1. compute value of the statistic on original data: Skewness = -0.324 \*/  
proc means data=sample nolabels Skew; var x; run;  
%let NumSamples = 5000; /\* number of bootstrap resamples \*/  
/\* 2. Generate many bootstrap samples \*/  
proc surveyselect data=sample NOPRINT seed=1  
 out=BootSSFreq(rename=(Replicate=SampleID))  
 method=urs /\* resample with replacement \*/  
 samprate=1 /\* each bootstrap sample has N observations \*/  
 /\* OUTHITS option to suppress the frequency var \*/  
 reps=&NumSamples; /\* generate NumSamples bootstrap resamples \*/  
run;  
/\* 3. Compute the statistic for each bootstrap sample \*/  
proc means data=BootSSFreq noprint;  
 by SampleID;  
 freq NumberHits;  
 var x;  
 output out=OutStats skew=Skewness; /\* approx sampling distribution \*/  
run;  
proc means data=OutStats nolabels N StdDev;  
 var Skewness;  
run;  
/\* 4. Use approx sampling distribution to make statistical inferences \*/  
proc univariate data=OutStats noprint;  
 var Skewness;  
 output out=Pctl pctlpre =CI95\_  
 pctlpts =2.5 97.5 /\* compute 95% bootstrap confidence interval \*/  
 pctlname=Lower Upper  
 pctlpre =Mean\_ mean=BootMean std=BootStdErr;  
  
run;  
   
proc print data=Pctl noobs; run;  
title "Bootstrap Distribution";  
%let Est = -0.324;  
proc sgplot data=OutStats;  
 label Skewness= ;  
 histogram Skewness;  
 /\* Optional: draw reference line at observed value and draw 95% CI \*/  
 refline &Est / axis=x lineattrs=(color=red)   
 name="Est" legendlabel="Observed Statistic = &Est";  
 refline -0.79536 0.23424 / axis=x lineattrs=(color=blue)   
 name="CI" legendlabel="95% CI";  
 keylegend "Est" "CI";  
run;  
/\* Bootstrap distribution of the sample mean. Example taken from   
 R. Wicklin (2010), Statistical Programming with SAS/IML Software,  
 SAS Press: Cary, NC, pp. 350-356.  
\*/  
%let DSName = scmm.score;  
%let VarName = vec;  
%let alpha = 0.05; /\* significance; (1-alpha)100% conf limits \*/  
  
proc iml;  
use &DSName;  
read all var {&VarName} into x;  
close &DSName;  
  
/\* Resample B times from the data (with replacement)   
 to form B bootstrap samples. \*/  
B = 5000; /\* number of bootstrap samples \*/  
call randseed(12345);  
xBoot = Sample(x, B||nrow(x)); /\* each column is a resample \*/  
  
/\* Compute the statistic on each bootstrap resample \*/   
s = T( mean(xBoot) ); /\* mean of each resample \*/  
title "Bootstrap distribution of the mean";  
if num(symget("SYSVER"))>=9.4 then do;  
 call Histogram(s) density="Kernel"; /\* graph bootstrap distrib \*/  
end;   
  
Mean = mean(x); /\* sample mean of original data \*/  
/\* Analyze the bootstrap distribution \*/  
MeanBoot = s[:]; /\* a. mean of bootstrap dist \*/  
StdErrBoot = std(s); /\* b. estimate of std error \*/  
prob = &alpha/2 || 1-&alpha/2; /\* lower/upper percentiles \*/  
call qntl(CIBoot, s, prob); /\* c. quantiles of bootstrap dist\*/  
pct = putn(1-&alpha, "PERCENT5.");  
  
print Mean MeanBoot StdErrBoot   
 (CIBoot`)[c=("Lower "+pct+" CL" || "Upper "+pct+" CL")];  
quit;  
title;  
  
/\* By the Central Limit Theorem, the sampling distribution of  
 the mean is approximately normally distributed.   
 If desired, compare the bootstrap estimates with   
 estimates of the SEM and CLM that assume normality. \*/  
/\*  
proc means data=&DSName mean stderr clm alpha=&alpha;  
var &VarName;  
run;  
\*/