Paper 343-2012

Using SAS® for the Design, Analysis, and Visualization of Complex Surveys

Sharon L. Lohr, Arizona State University, Tempe, AZ, USA

ABSTRACT

SAS® PROC SURVEYMEANS, SURVEYREG, SURVEYSELECT, and other members of the SURVEY family are powerful tools for designing surveys and analyzing data from complex surveys. We present an overview of the procedures and their standard uses. We then present examples from complex surveys to illustrate how recent developments to these procedures allow them to be used for advanced applications such as graphing complex survey data and making inferences employing the bootstrap.

INTRODUCTION

Data from complex surveys require special software for correct analyses because of two features. First, observations are often selected with different probabilities. The sampling weights, which are the reciprocals of these selection probabilities, then vary for different observations. In addition, weights are also often adjusted for nonresponse, to calibrate to known population quantities, and other purposes. Almost all large surveys have a column of weights, and the weights are generally unequal for the observations because of disproportionate sampling fractions and/or nonresponse adjustments.

The weights suffice to compute any point estimate of a finite population quantity from the data (but not the standard error). Suppose the finite population has *N* units. Let y_i be a measurement on unit i in the sample, and let w_i be the weight associated with unit i. The population total for the variable *y,* $\sum_{i=1}^{N} y_i$, is then estimated by $\sum_{i\epsilon S} w_i y_i$., where the sum is over all observations in the sample, S. The sum of the weights, $\sum_{i\epsilon S} w_i$ estimates the number of units in the finite population, *N*, so the population mean may be estimated by $(\sum_{i\epsilon S} w_i y_i)/(\sum_{i\epsilon S} w_i\ )$. The median, regression coefficients, and almost any other statistic may be estimated similarly by using the weights. Lohr (2010) describes how survey weights are constructed from different sampling designs.

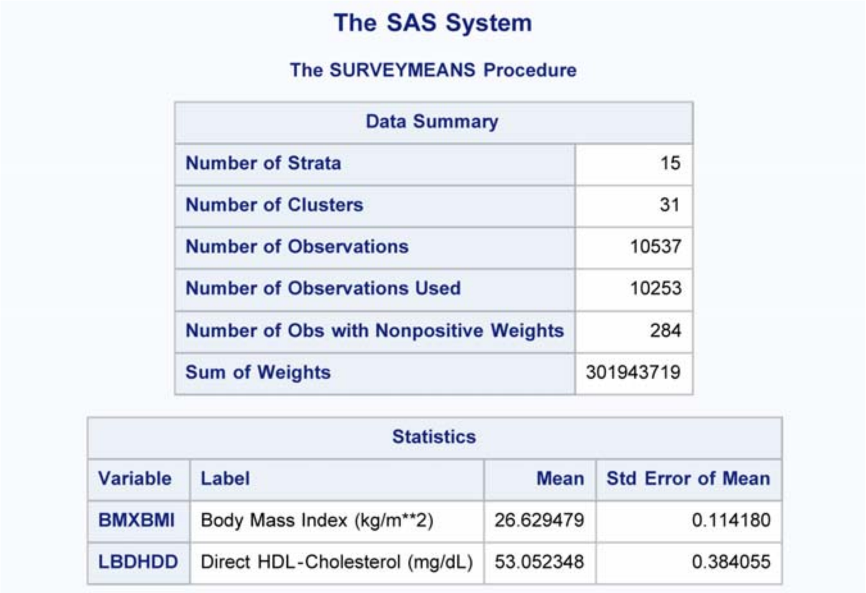
The second feature of complex surveys that makes special software necessary is the use of stratification and clustering when selecting the sample. It is not necessary to know the stratification and clustering information if only point estimates are desired: the weights are sufficient for that. But it is necessary to know this information to be able to estimate variances of statistics calculated from the data.

Prior to the introduction of the SURVEY procedures, SAS® users could obtain point estimates from complex survey data by using certain procedures that allowed weights (for example, the MEANS and REG procedures could be used to calculate univariate summary statistics and regression coefficients for survey data by including a WEIGHT statement), but standard errors from nonSURVEY procedures are generally incorrect unless the survey design is a simple random sample.

The introduction of the SURVEYMEANS and SURVEYREG procedures in 1998 allowed data analysts to obtain correct inferences from survey data collected using a stratified cluster design. The SURVEYFREQ and SURVEYLOGISTIC procedures were introduced in Version 9.1 to extend the capabilities to allow analysis of contingency tables and logistic regression. Version 9.2 included computer-intensive (replication-based) variance estimation methods for jackknife and balanced repeated replication, and Version 9.22 saw the introduction of the SURVEYPHREG procedure for proportional hazards regression. Version 9.3 in 2011 extended the replication-based variance estimation to encompass domain statistics and quantiles.

All of the SURVEY analysis procedures have the same command structure. We use the public-use data from the 2009-2010 National Health and Nutrition Examination Survey (NHANES; see www.cdc.gov/nchs) to illustrate the SURVEY procedures. NHANES is a stratified cluster survey in which areas with high minority populations are oversampled. As a consequence, the weights for persons in those areas are reduced. Output 1 shows the estimated mean of body mass index (*bmxbmi)* and HDL (“good”) cholesterol (*lbdhdd*) from the NHANES data, using PROC SURVEYMEANS. We only need to specify the variables containing the weights, stratification information, and clustering information to do a proper analysis. The public-use data give pseudo-strata (*sdmvstra)* and pseudo-clusters (*sdmvpsu)* to maintain confidentiality of the data, and we use these here.

```
/* Output 1 */
proc surveymeans data=nhanes mean;
  weight wtmec2yr;
  stratum sdmvstra;
  cluster sdmvpsu;
  var bmxbmi lbdhdd;
```



The SAS System

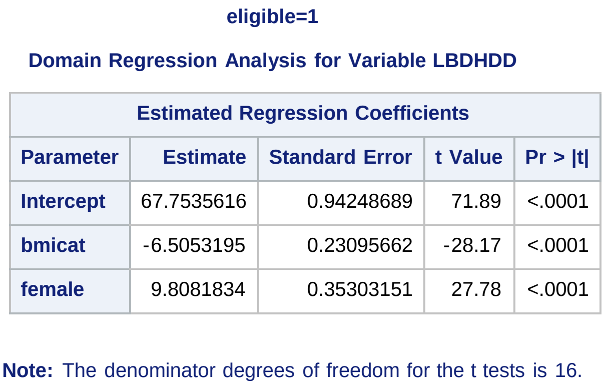
The SURVEYMEANS Procedure

Data Summary	
Number of Strata	15
Number of Clusters	31
Number of Observations	10537
Number of Observations Used	10253
Number of Obs with Nonpositive Weights	284
Sum of Weights	301943719

Statistics			
Variable	Label	Mean	Std Error of Mean
BMXBMI	Body Mass Index (kg/m**2)	26.629479	0.114180
LBDHDD	Direct HDL-Cholesterol (mg/dL)	53.052348	0.384055

Output 1. Output from PROC SURVEYMEANS, giving means and standard errors for body mass index and HDL cholesterol.

All of the other SURVEY analysis procedures use the same commands of WEIGHT, STRATUM, and CLUSTER to specify the survey design. The following example performs a linear regression of HDL cholesterol (*lbdhdd*) on a grouped variable constructed from the body mass index called *bmicat* and an indicator variable *female* that takes the value 1 if the person is female and 0 if the person is male. We are interested in the relationship between the variables for adults. If the data were collected using a simple random sample, we could simply carry out the regression analysis for a subset of data containing only the adults. In complex survey designs, however, such an approach may give incorrect standard errors; instead, in the PROC SURVEYs, we can do separate data analyses for subpopulations of the data by using the DOMAIN statement. The domains are specified by the variable *eligible* that takes the value 1 for persons who are age 20 or older and have values for the response variable and covariates, and 2 for other persons in the data set. Output 2 gives the estimated regression coefficients for the adults in the sample (the domain with *eligible* = 1).



eligible=1

Domain Regression Analysis for Variable LBDHDD

Estimated Regression Coefficients				
Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	67.7535616	0.94248689	71.89	<.0001
bmicat	-6.5053195	0.23095662	-28.17	<.0001
female	9.8081834	0.35303151	27.78	<.0001

Note: The denominator degrees of freedom for the t tests is 16.

Output 2. Partial output from PROC SURVEYREG, showing the regression of HDL cholesterol on BMI category and female for the adults in the sample.

```
/* Output 2 */
data nhanes;
  set nhanes;
  if 0 le bmxbmi lt 18.5 then bmicat = 1;
  else if 18.5 le bmxbmi lt 25 then bmicat = 2;
  else if 25 le bmxbmi lt 30 then bmicat = 3;
  else if bmxbmi ge 30 then bmicat = 4;
  if (lbdhdd ne . and female ne . and bmxbmi ne . and age ge 20) then eligible = 1;
  else eligible = 2;

proc surveyreg data=nhanes;
  weight wtmec2yr;
  stratum sdmvstra;
  cluster sdmvpsu;
  domain eligible;
  model lbdhdd = bmicat female;
  output out=regout predicted=hdlpred residual = hdlresid;
```

Output 2 gives the regression coefficients and test statistics, but does not assess whether the model is appropriate for the data. To do that, we need to look at some graphs of the data and of the residuals that are produced by PROC SURVEYREG. We'll come back to this regression analysis after exploring some methods for graphing data from complex surveys.

GRAPHING COMPLEX SURVEY DATA

One of the first principles students are taught in introductory statistics classes is that you should always plot your data. Very few analyses of survey data, however, include such plots. The large size of many survey data sets, coupled with the presence of unequal weights from disproportionate sampling or nonresponse adjustments, makes it challenging to plot the data. When surveys contain unequal weights, standard graphical displays show the distribution from the observed data but may be misleading for reflecting relationships in the population of interest.

All of the plots shown in this section estimate the plots we would obtain if we knew the entire finite population. They make use of the principle that we estimate the total of any population quantity $\sum_{i=1}^{N} y_i$ by $\sum_{i\epsilon S} w_i y_i$. These plots use the weights only and do not display clustering or stratification information. Some of these plots and other plots are described in Korn and Graubard (1999) and Lohr (2010).

We illustrate the plots using data from the National Science Foundation's Scientists and Engineers Statistical System (SESTAT). The SESTAT data are collected to study the science and engineering (S&E) workforce in the United States (see National Science Foundation, 2011 for a precise description of the target population and a description of the design of the surveys). We use the 2006 public use data set, available for download from www.nsf.gov/statistics/sestat. The survey weights in the data range from approximately 1.2 to 2100. The weights vary greatly in this data set because minorities, persons employed in an S&E field, and other groups of interest have high inclusion probabilities and hence have lower survey weights. Although the public use file provides the weights, it does not provide stratification and clustering information that could be used to estimate variances of statistics, so we can construct plots and find point estimates but we cannot provide standard errors or confidence intervals. We use the following variables in the plots: *weight* = survey weight; *agep* = age, in years*, salaryt* = annual salary, in thousands of dollars (calculated as *salarp*/1000 from the public use data file); and *job* = occupation category (calculated from variables *nocprmg* and *nocprpb* in the public use data file). In this paper, the GOPTIONS statements and other graph formatting statements are omitted to save space; the full code is available from the author.

UNIVARIATE PLOTS

Let's start by considering simple histograms and boxplots. We include the sample weights in the histogram, so the histogram plotted from the sample estimates the histogram we would obtain if we knew the entire finite population. Thus, we would draw the histogram of the entire population by dividing the population into *J* bins of width *b*. The height of bin *j* is the relative frequency for that bin. To express this in terms of population totals, let $u_i(j) = 1$ if observation *i* falls in bin *j* and 0 otherwise. Then the height of bin *j* for the finite population is $\sum_{i=1}^{N} u_i(j)/\left[\sum_{i=1}^{N} 1\right]$. We estimate the population totals in numerator and denominator by using the sample weights, to obtain

$$\text{Height of bin } j = \sum_{i\epsilon S} w_i u_i(j) \,/ [\sum_{i\epsilon S} w_i].$$

To plot a histogram in SAS, then, we need to adapt the existing histogram procedures so that they can incorporate the weights when calculating heights of bars in the histogram. First, note that the UNIVARIATE procedure is often used for histograms with simple random samples, but is not appropriate for complex survey data since currently the HISTOGRAM statement cannot be used with the WEIGHT statement. Instead, we use the GCHART procedure to draw histograms for complex survey data. Figure 1 shows the difference from using the weights in a relative frequency histogram. Both histograms are constructed using the SUMVAR option; so that both graphs will be on the same scale, the histogram without weights on the left uses the sum of the variable *relfq = 1/(number of observations);* the histogram with weights on the right uses the sum of the variable *relwt = weight/(sum of all weights).* The histogram with weights, in Figure 1(b), shows relatively more people in the $40,000-$60,000 range for annual salary. This occurs because persons who were not employed in an S&E field at the time of the 2000 Census have higher survey weight and are more likely to be in that salary range. Both graphs also show the spike at 150, which occurs because all salaries in the public use files are truncated at $150,000. This suggests that a data analyst may want to summarize the data using percentiles rather than means.

```
  proc gchart data=epses06; /* Constructs Figure 1(a) */
    vbar salaryt / midpoints= 5 to 155 by 10 sumvar = relfq space=0;
  proc gchart data=epses06; /* Constructs Figure 1(b) */
    vbar salaryt / midpoints= 5 to 155 by 10 sumvar = relwt space=0;
```

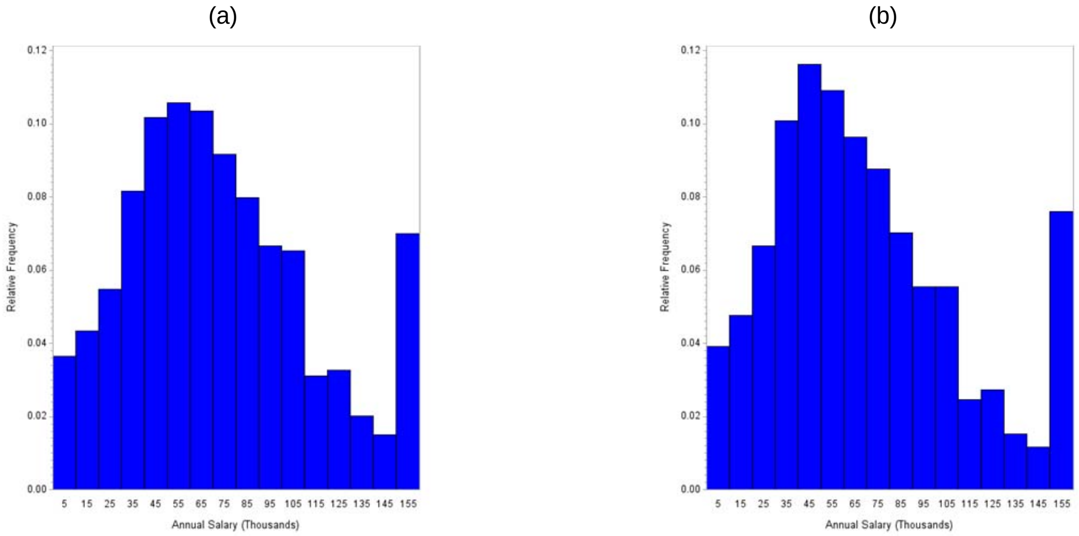


Figure 1. Relative frequency histograms for annual salary from SESTAT data. (a) The histogram constructed without using the survey weights displays the sample values, but does not reflect the population distribution. (b) The histogram constructed using the survey weights estimates the distribution of salary in the target population of the survey.

Another option for drawing a histogram with weights is in SAS/INSIGHT®: choose Analyze > Distribution from the menu, then input the response variable and the weight variable. SAS/INSIGHT® will also construct a kernel density estimate using the weights, as will the KDE procedure. See Lohr (2010, p. 298) for more information on density plots with survey data.

Boxplots are constructed similarly to histograms: use the weights to calculate the percentiles of the data that will be displayed. These are then converted to the same format that the BOXPLOT procedure would create as an OUTHISTORY= data set, and used as input to either the BOXPLOT procedure or the ANOM procedure. The macro %*surveybox* may be used to construct the needed quantiles. PROC SURVEYMEANS does not yet produce quantiles for domains, so we use the BY statement to calculate the quantiles for each group using the weights. This is acceptable for our purpose of drawing boxplots since we are interested only in the point estimates of the quantiles; in general, though, using a BY statement instead of a DOMAIN statement may produce incorrect standard errors.

```
%macro surveybox (fulldata, y, group, wt, outquant);

  /* Input parameters
     fulldata     data set name
     y            name of variable for boxplot
     group        name of variable to use for grouping
     wt           name of weight variable
     outquant     name of data set to contain output quantiles */

  /* Output variables in data set outquant
     group        grouping variable
     q            Variable q has the quantiles
                  Note that qN = sum of the weights, not number of observations.
                  Also, qS = standard error, not standard deviation, but qS
                  is not used to construct the skeletal boxplots. */

proc sort data=&fulldata;
  by &group;
```

```
proc surveymeans data=&fulldata mean percentile=(0 25 50 75 100) sumwgt;
  by &group;
  weight &wt;
  var &y;
  ods output Statistics= &outquant;
run;

data &outquant (keep= &group q:);
  set &outquant;
  qL = Pctl_0;
  q1 = Pctl_25;
  qX = Mean;
  qM = Pctl_50;
  q3 = Pctl_75;
  qH = Pctl_100;
  qS = StdErr;
  qN = SumWgt;
run;

%mend surveybox;
```

Figure 2 displays side-by-side skeletal boxplots of *salaryt* for different occupational categories. To obtain a more informative display than afforded by an alphabetical ordering of occupational categories, we sorted the categories from smallest median to largest median. Because the salary values are truncated at 150, the means displayed in the boxplot likely underestimate the mean salary for each group; the median is a better statistic to use to describe the center of the data.

```
/* Figure 2 */
%surveybox(epses06, salaryt, job, weight, jobquant)

proc sort data=jobquant;
  by qM;

/* Either of the following two procedures may be used to produce Figure 2 */

proc boxplot history=jobquant;
   plot q*job/ haxis = axis3 vaxis=axis4;
run;

proc anom summary=jobquant;
   boxchart q*job / nolimits boxwidth= 4 haxis=axis3 vaxis=axis4 nolegend serifs;
run;
```

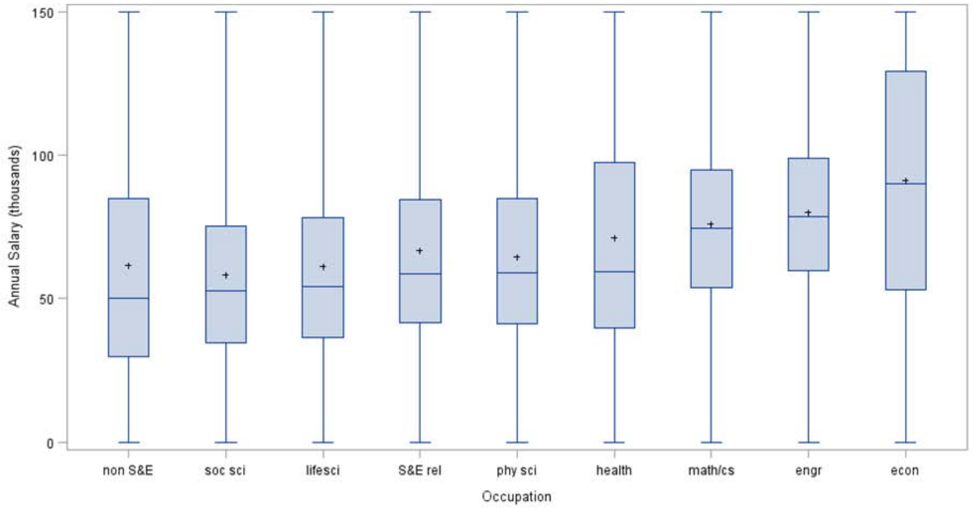


Figure 2. Side-by-side boxplots for salary data.

SCATTERPLOTS

Many persons analyzing survey data wish to explore multivariate relationships. As with the univariate plots, we need to incorporate the weights. But there is an additional complication: large surveys often have thousands of observations, so even if all weights are the same, a standard scatterplot of the data will simply show a big black mass, making it difficult to see anything. There are more than 90,000 observations in the SESTAT data with valid salary information, and a scatterplot of the raw data in Figure 3(a) provides no insight into the data structure. We need to summarize the information for the viewer in addition to including the weights. We did this in the univariate plots by exploiting the data reduction properties of histograms and boxplots: histograms bin the data, and boxplots summarize the data through percentiles. The same principles are used in scatterplots. We present several different methods for displaying bivariate data; some methods work better than others on particular data sets, so it is often good to try a variety of different plots.

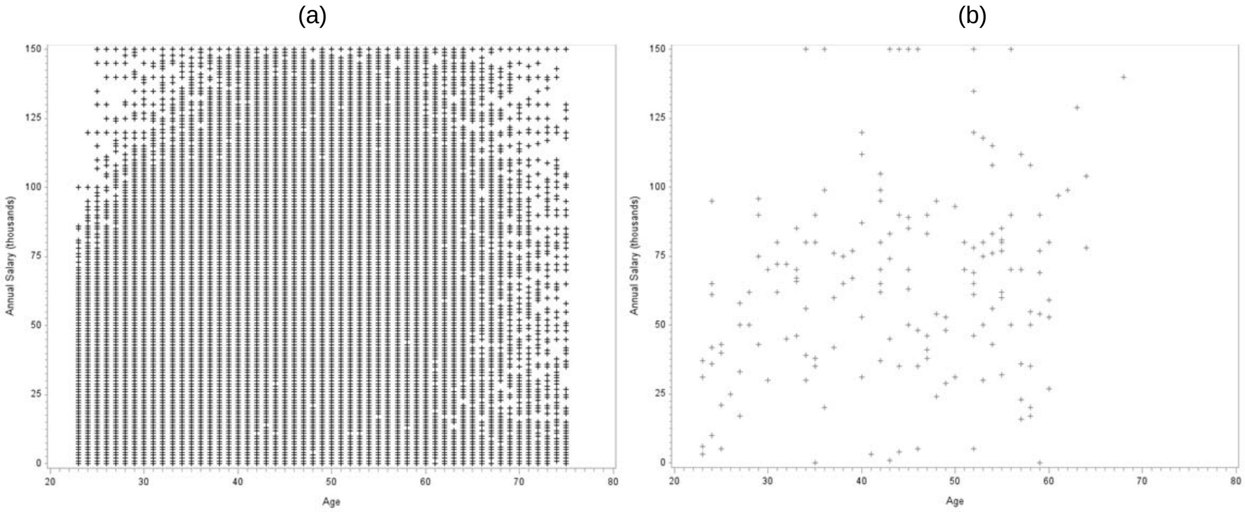


Figure 3. (a) Scatterplot of raw data from SESTAT. The graph does not display the multiple observations that occur at each plot symbol, and obscures patterns in the data. (b) Scatterplot of subsample of 200 points, selected with probability proportional to survey weight.

To display individual points in a scatterplot, we can construct a series of plots that each display a subsample of the sample. Each subsample is selected with probability proportional to the sample weights. One example is given in Figure 3(b). This sort of plot works well with some data sets (for example, it works well for displaying patterns in the NHANES data), but is less successful for the SESTAT data because it provides information about such a small fraction of the data and still obscures the multiple values at some of the points. For these data, then, we also need alternative methods for displaying the bivariate relationship.

```
/* Figure 3(b) */
proc surveyselect data=epses06 method=pps_wr sampsize=200 out=ppssamp seed=1142938;
   size weight;
proc gplot data=ppssamp;
   plot salaryt*agep;
```

A natural extension of the boxplot idea for bivariate data is to display side-by-side boxplots for different strips of the *x* values. Thus, we bin the *x* values into 10-20 categories, then display a boxplot for each strip that is calculated using the survey weights. The macro %surveybox may be used to calculate the quantiles. Figure 4 shows a boxplot in which the option *boxwidthscale* sets the width of each box to be proportional to the sum of weights of observations in that category. The following code is used:

```
/* Figure 4 */
data groupage;
   set epses06;
   agegroup = round(agep,5);
%surveybox (groupage, salaryt, agegroup, weight, outquant);

proc anom summary=outquant;
   boxchart q*agegroup / nolimits boxwidth= 6 boxwidthscale=1
            npanelpos = 100 nolegend haxis = axis3 vaxis=axis4;
```

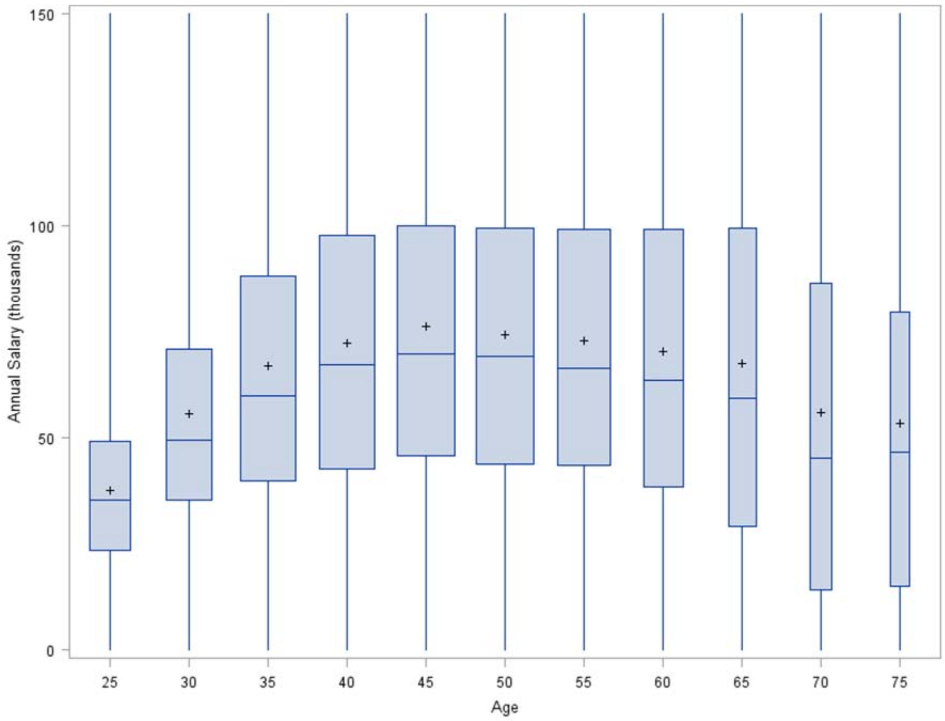


Figure 4. Side-by-side boxplots of groups of ages. The width of each box is proportional to the sum of the weights in that age category.

Bubble plots or shading extend the binning method used in histograms to two dimensions. Divide the plotting region into rectangular subregions. Then, sum the weights of the observations falling in each rectangle. A bubble plot draws a bubble in the center of the rectangle with area proportional to the sum of the weights; a shaded plot shades the rectangle with a darkness intensity proportional to the sum of the weights. Since we can fill in the entire rectangle, we can often have a finer grid for the shaded plot than for weighted bubble plots. Figures 5 and 6 display two different bubble plots for the data, with different sizes of binning rectangles. Figure 7 displays a shaded plot, using the GCONTOUR procedure. Each of these plots shows the large number of observations truncated to 150, a feature that is not visible in Figure 4. Figures 6 and 7 also display a pattern of horizontal stripes, indicating more observations at round numbers of salary such as 50 and 100.

```
/* Figure 5 */
data groupage;
   set epses06;
   salgroup = round(salaryt,10);
   agegroup = round(agep,5);

proc sort data=groupage;
   by salgroup agegroup;

proc means data=groupage;
   by salgroup agegroup;
   var weight;
   output out=circleage sum=sumwts;

proc gplot data=circleage;
   bubble salgroup*agegroup=sumwts/bsize=25 haxis = axis3 vaxis = axis4;
```

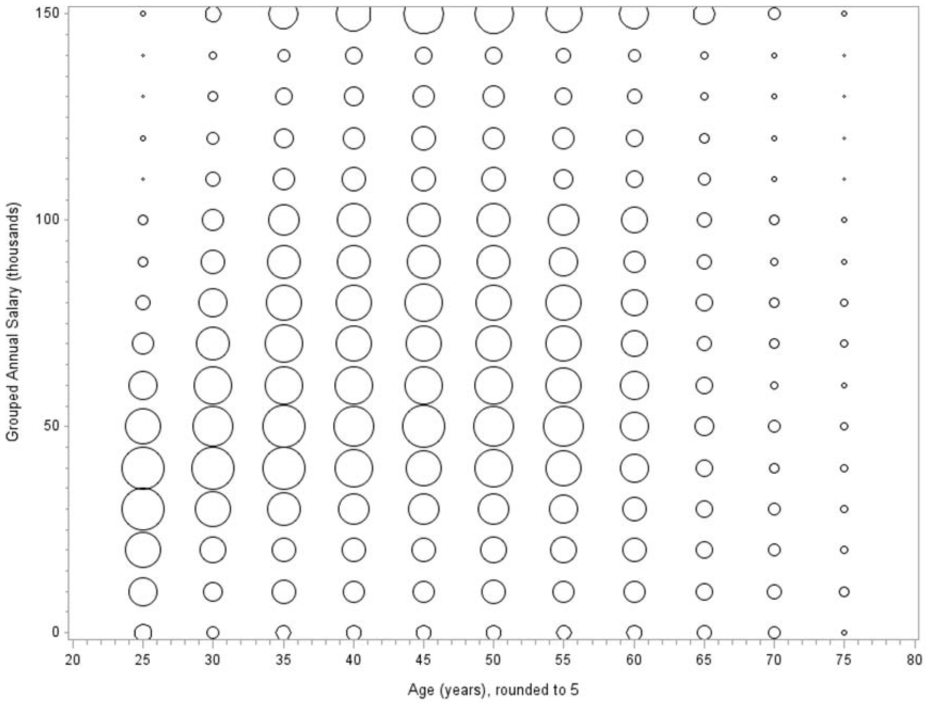


Figure 5. Bubble plot of salary by age, for grouped data. The area of each bubble is proportional to the sum of weights for the area.

```
/* Figure 6 */
proc sort data=epses06;
   by salaryt agep;

proc means data=epses06 noprint;
   by salaryt agep;
   var weight;
   output out=bubbleage  sum=sumwts;

proc gplot data=bubbleage;
   bubble salaryt*agep = sumwts / bsize=20 bfill = solid haxis = axis1 vaxis = axis2;
run;

/* Figure 7 */
/* You need to specify the colors for each level in pattern statements
   before calling PROC GCONTOUR */

proc gcontour data=bubbleage incomplete;
   plot salaryt*agep=sumwts /levels = 0 2000 4000 6000 8000 10000 12000 70000 pattern
        haxis=axis3 vaxis=axis4 nolegend;
```

For many plots, it is helpful to see a trend line along with the plot. Nonparametric trend lines can be calculated in the LOESS or SURVEYREG procedures (if least squares would be appropriate for a nonparametric regression) or the QUANTREG procedure (if a more robust trend line is desired) by using an EFFECT statement. The SESTAT salary data are topcoded at $150,000, so for this data set PROC QUANTREG, which gives quantile regression lines, is more appropriate. Here, we use a cubic B-spline basis with 9 equally spaced interior knots. The predicted values for the three quantile regressions (for 25th, 50th, and 75th percentiles) calculated in PROC QUANTREG are output to a data set, and then superimposed on a weighted bubble plot of the data in Figure 8. Because the QUANTREG procedure is computationally intensive, for some data sets it may be helpful to obtain initial values by running the procedure on a subsample of the data such as the subsample used in constructing Figure 3(b).

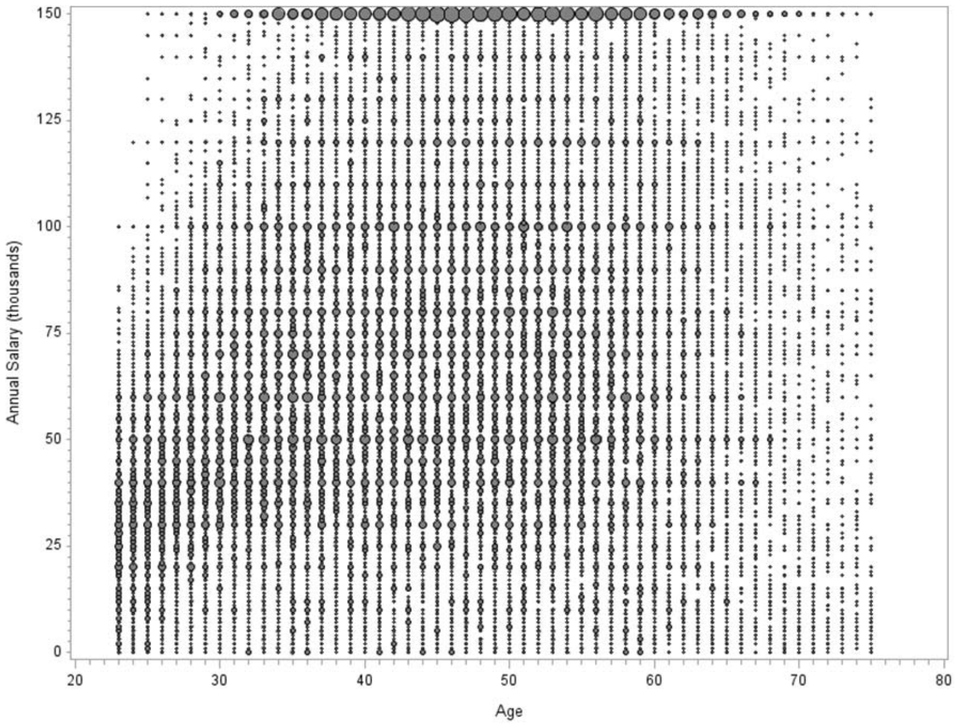


Figure 6. Bubble plot of salary by age. The area of each bubble is proportional to the sum of weights at that value of (*x*,*y*).

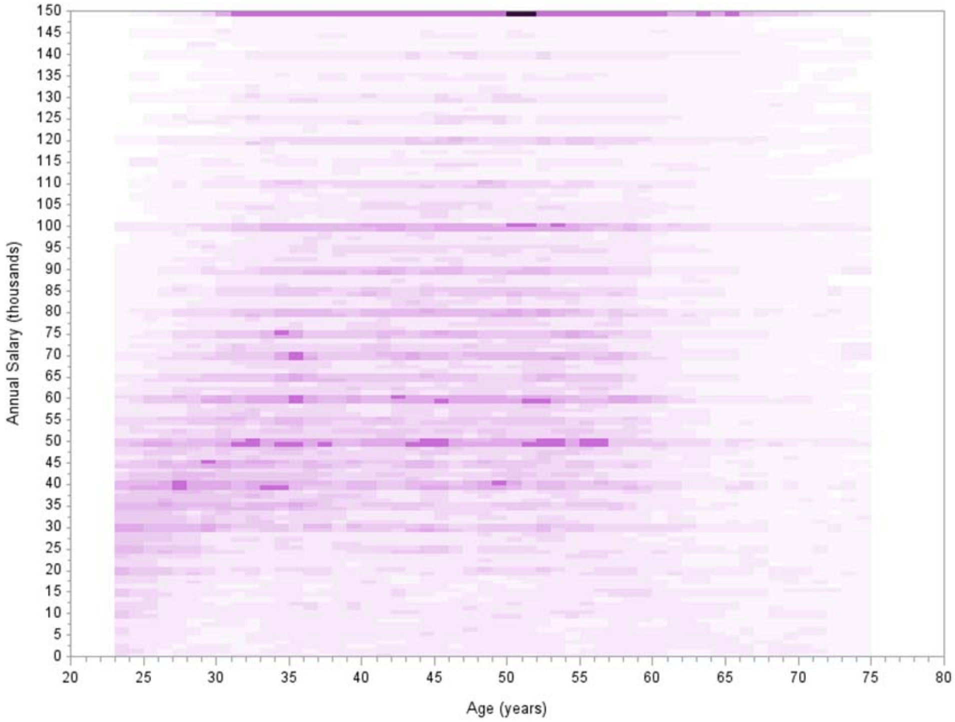


Figure 7. Shaded plot of salary by age. The darkness of each rectangle is proportional to the sum of weights at that value of (*x*,*y*).

```
/* Figure 8 */
ods graphics on;
proc quantreg data=epses06 algorithm=interior ci=none
              plot=fitplot(nodata) plots(maxpoints=none);
   effect spl = spline( agep / knotmethod = equal(9) );
   model salaryt = spl / quantile = .25 0.5 .75  seed=38274;
   weight weight;
   output out = outpred predicted = pred residual = resid;
run;
ods graphics off;

proc sort data=outpred;
   by agep;

data bubbleage; /* data bubbleage is from Figure 6 */
   set bubbleage;
   salarytn = salaryt; /* rename the response variable for plotting */

data plotqr;
   set bubbleage outpred;

goptions reset=all;
axis1 label=('Age') order=(20 to 80 by 10);
axis2 label=(angle=90 'Annual Salary (thousands)') order=(0 to 150 by 25);
axis5  order=(0 to 150 by 25) major=none minor=none value=none;
symbol1 interpol=join width=1.5 color = red;
symbol2 interpol=join width=1.5 color = indigo;
symbol3 interpol=join width=1.5 color = red;

proc gplot data=plotqr;
   bubble salarytn*agep=sumwts / bsize=20 bcolor=turquoise haxis=axis1 vaxis=axis2;
   plot2 (pred1 pred2 pred3)*agep/ overlay haxis = axis1 vaxis = axis5;
```

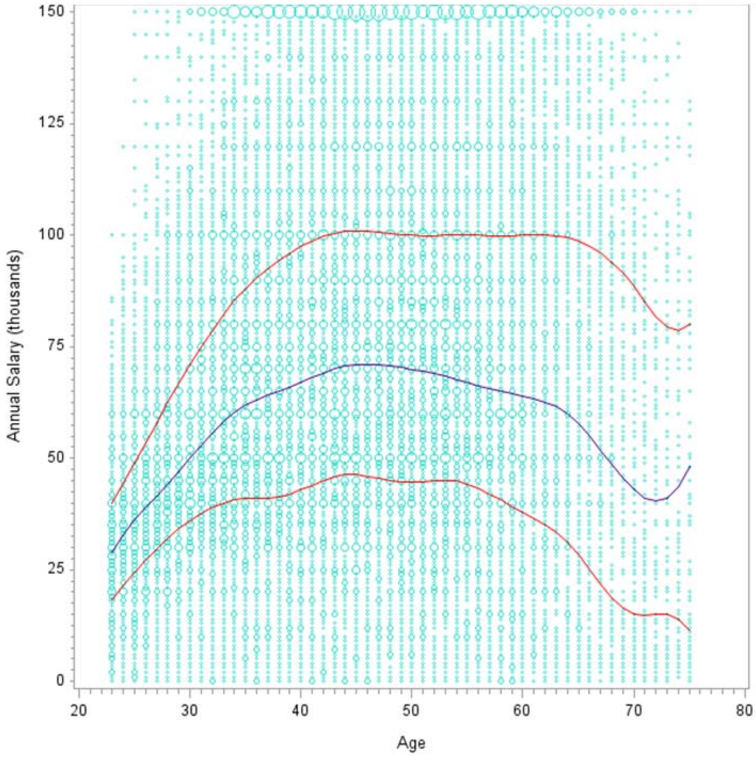


Figure 8. Weighted bubble plot of salary by age, with nonparametric quantile regression lines for 25th, 50th (blue), and 75th quantiles.

All of the graphs presented in this section may be used to assess the fit of regression models with multiple predictors as well as displaying the data. PROC SURVEYREG will output the residuals and predicted values from a model fit. The graphical methods in this section may then be used to construct residual plots by including the survey weights. Figure 9 shows boxplots of the data, and of the residuals vs. predicted values for the regression analysis given in Output 2. Although there are a few outliers in the data, there are no serious indications of model inadequacy.

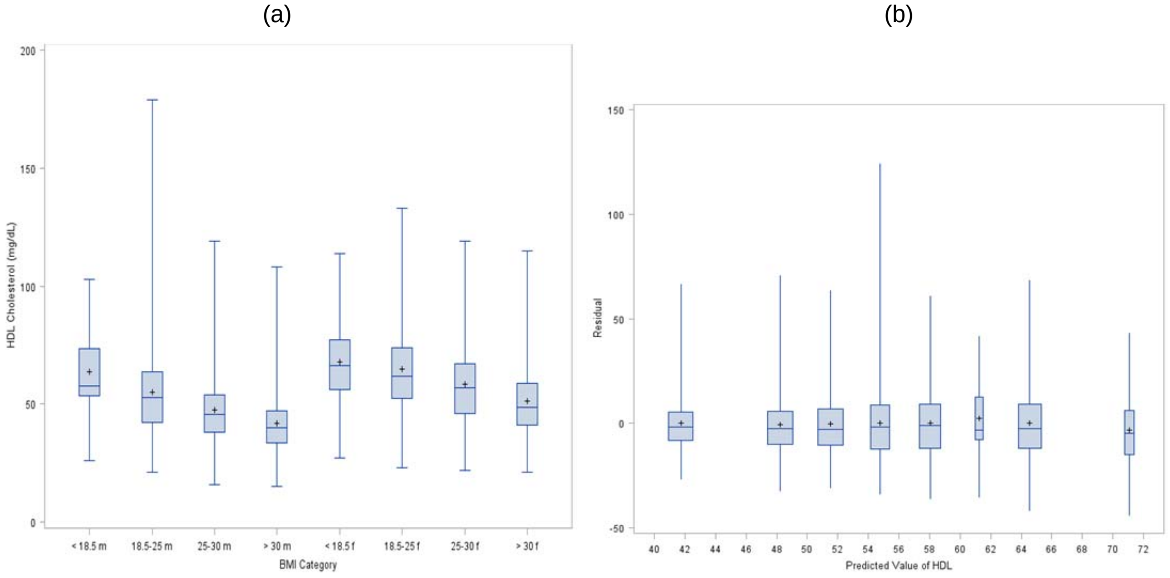


Figure 9. (a) Boxplot of data, showing BMI category separately for males (with suffix m) and females (with suffix f) and (b) Residuals vs. predicted values for regression of HDL cholesterol on *bmicat* and *female*.

BOOTSTRAP

SAS Version 9.2 allowed the user to perform analyses with replicate weights using the balanced repeated replication and jackknife methods (see Mukhopadhyay et al., 2008). In some cases, however, users may want to estimate standard errors with the bootstrap (Lohr, 2010, Section 9.3). Since the SESTAT data truncate the values for salary, one would want to use percentiles rather than means to summarize the distribution, and quantile regression rather than least squares regression to explore multivariate relationships. Unfortunately, although the jackknife method gives consistent estimates of the variance of smooth functions of means, it generally works poorly for estimating the variance of quantiles (Martin, 1990). The documentation for PROC SURVEYMEANS gives the following warning: “When you use the replication method, PROC SURVEYMEANS uses the usual variance estimates for a quantile as described in the section Replication Methods for Variance Estimation. However, you should proceed cautiously because this variance estimator can have poor properties.”

The bootstrap, on the other hand, can be used to estimate the variance of quantiles (Shao and Chen, 1998) when the number of first-stage clusters in the sample is sufficiently large and other conditions are met for the survey design. We use PROC SURVEYSELECT to construct bootstrap weights. Then the replication methods in PROC SURVEYMEANS, SURVEYREG, or other procedures can be used to calculate estimates.

In general, replication methods work as follows: a parameter θ is estimated by $\hat{\theta}$, which is calculated using a formula involving the data values and the weights w_i. Then, using the desired variance estimation method, *R* columns of replicate weights are formed, where observation *i* has weight $w_i^{(r)}$ for replicate *r*. The replicate estimate $\hat{\theta}^{(r)}$ is calculated in exactly the same way as $\hat{\theta}$, except that the replicate weights $w_i^{(r)}$ are substituted for the original weights w_i in the formula used to calculate the estimate. The jackknife estimate of the variance of $\hat{\theta}$ in PROC SURVEYMEANS, using option VARMETHOD=JACKKNIFE, is then calculated as

$$\hat{V}(\hat{\theta}) = \sum_{r=1}^{R} \alpha_r \left(\hat{\theta}^{(r)} - \hat{\theta}\right)^2,$$

where α_r is provided by the user in the JKCOEFS= option. Although the jackknife variance estimator is intended for use with jackknife replicate weights, the method is perfectly general and may be used with other replicate weight methods for which the estimated variance has the same form. Thus, the jackknife method in PROC SURVEYMEANS may be used with replicate weights created by the bootstrap provided the correct JKCOEFS value is used.

As with all resampling methods for complex surveys, the bootstrap must keep all observations of the same cluster together when forming the replicates. Thus, the person creating the bootstrap weights must know the stratification and clustering information. The McCarthy and Snowden (1985) bootstrap, which is a special case of the rescaling bootstrap of Rao and Wu (1988), can be used with a stratified multistage sampling design. The macro %bootwt creates *R* columns of replicate weights using PROC SURVEYSELECT; these can then be used with the jackknife capabilities of the other SURVEY routines to calculate estimates and standard errors. Suppose the design has *H* strata, and stratum *h* has n_h clusters. For the *r*th set of replicate weights, take a simple random sample *with replacement* of $(n_h - 1)$ clusters from stratum *h*, for *h* = 1,… *H*. This can be done in SURVEYSELECT by taking a with-replacement stratified sample of the cluster identifiers. Since the bootstrap sample is selected with replacement any individual cluster may appear 0, … , $n_h - 1$ times. Let $m_{hj}(r)$ denote the number of times cluster *j* of stratum *h* is selected for the sample used in replicate *r*. To form the replicate weight for observation *i* that occurs in cluster *j* of stratum *h*, $w_i^{(r)}$, set

$$w_i^{(r)} = w_i \frac{n_h}{n_h - 1} m_{hj}(r).$$

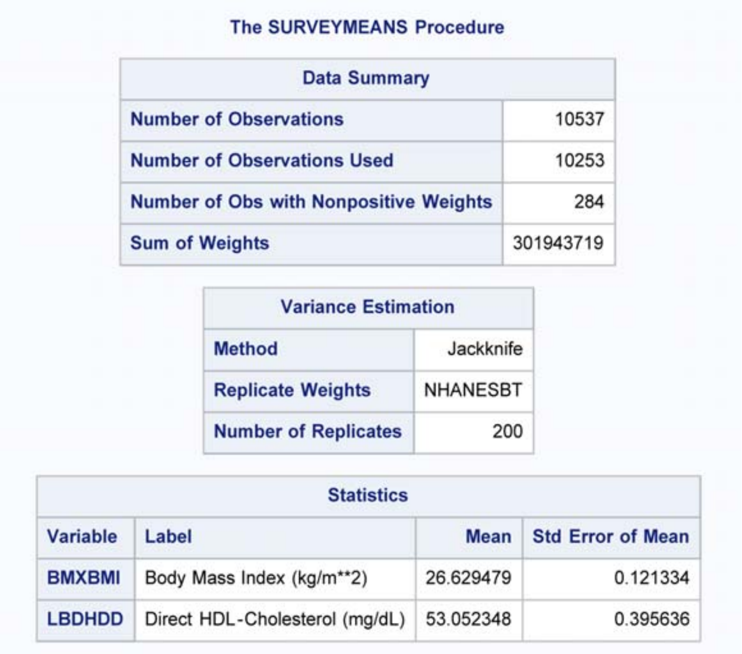
The replicate bootstrap weights may be used in other SAS SURVEY procedures with VARMETHOD=JACKKNIFE to obtain estimates of variances of characteristics of interest. The jackknife coefficient is JKCOEFS= 1/(*R*-1). To obtain the “usual” degrees of freedom for confidence intervals, input the degrees of freedom directly through the DF= option in the REPWEIGHTS statement; generally the degrees of freedom will be df = (total number of clusters) – (total number of strata).

```
%macro bootwt (fulldata, wt, stratvar, psuvar, numboot, fullrep, repwt);

  /* Input
     fulldata     full data set name
     wt           name of weight variable
     stratvar     name of stratification variable
     psuvar       name of psu (cluster) variable
     numboot      number of bootstrap replicates desired
     fullrep      name of data set to contain replicate weights
     repwt        name of array in fullrep to contain bootstrap weights */

  /* In this implementation, output data set fullrep also contains all of the
     information in the original data set fulldata. */
  /* MUST HAVE at least 2 clusters in each stratum */

  /* Construct data set with list of strata and clusters */

  proc sort data= &fulldata  out= fulldata;
    by &stratvar &psuvar;
  run;

  proc sql stimer;
    create table psulist as
      select distinct &stratvar, &psuvar
      from fulldata
      order by &stratvar, &psuvar
    ;
    /*  Set stratum sample size to n_h - 1 */
    create table numpsu as
      select distinct &stratvar, count(*)-1 as _nsize_
      from psulist
      group by &stratvar
      order by &stratvar
    ;
    quit;

  data fulldata (drop= _nsize_);
    merge fulldata (in= inf)
```

```
          numpsu (in= inn);
    by &stratvar;
    if inf & inn;
    wtmult = (_nsize_ + 1) / _nsize_;
    run;

  /* Select samples for replicate bootstrap weights */

  proc surveyselect data=psulist method=urs sampsize=numpsu out=repout outall
                    reps=&numboot;
    strata &stratvar;
    id &psuvar;
    run;

  proc sort data= repout (keep= &stratvar &psuvar replicate numberhits)
            out= repout_sorted
            ;
    by &stratvar &psuvar replicate;
    run;

  proc transpose data= repout_sorted (keep= &stratvar &psuvar replicate numberhits)
                 out= repout_tr (keep= &stratvar &psuvar repmult: )
                 prefix=repmult;
    by &stratvar &psuvar;
    id replicate;
    var numberhits;
    run;

  /* Ok, now we have a dataset repout with the number of hits for
     each bootstrap replicate.  Now merge this data set
     with the original full data and multiply each original weight
     times the number of hits in repmult times (n_h/(n_h-1)). */

  data &fullrep (drop= i repmult1-repmult&numboot wtmult);
    array &repwt (&numboot);
    array repmult (&numboot);
    merge fulldata (in= inf)
          repout_tr (in= inr)
          ;
    by &stratvar &psuvar;
    do i = 1 to &numboot;
      &repwt(i) = &wt * repmult(i) * wtmult;
    end;
    run;

%mend bootwt;
```

We now use the %bootwt macro to create replicate bootstrap weights for the NHANES data analyzed in Outputs 1 and 2. The replicate weights are then used with PROC SURVEYMEANS and PROC SURVEYREG to calculate the standard errors of the statistics. The replicate weights capture the stratification and clustering information in the survey design, so the STRATA and CLUSTER statements should not be included when replication methods are used. Here, we used 200 bootstrap replicates, so we set JKCOEF equal to 1/199 = 0.005026. We set DF equal to 16, which is the degrees of freedom indicated by the survey design. Note that since the bootstrap takes repeated with-replacement stratified samples of the clusters, each time you run %bootwt you will obtain a different set of replicate weights and thus have a different estimate of the variance of a statistic. With sufficiently large values for the number of bootstrap iterations *R*, though, these estimates will be close to each other.

```
%bootwt (nhanes, wtmec2yr, sdmvstra, sdmvpsu, 200, nhanesbt, repwt);

/* Output 3 */
proc surveymeans data=nhanesbt varmethod=jk mean;
  weight wtmec2yr;
  var bmxbmi lbdhdd;
  repweights repwt1-repwt200 / jkcoef = .005026 df=16;
run;

/* Output 4 */
proc surveyreg data=nhanesbt varmethod=jk;
  weight wtmec2yr;
  domain eligible;
  model lbdhdd = bmicat female;
  repweights repwt1-repwt200 / jkcoef = .005026 df=16;
run;
```



The SURVEYMEANS Procedure

Data Summary	
Number of Observations	10537
Number of Observations Used	10253
Number of Obs with Nonpositive Weights	284
Sum of Weights	301943719

Variance Estimation	
Method	Jackknife
Replicate Weights	NHANESBT
Number of Replicates	200

Statistics

Variable	Label	Mean	Std Error of Mean
BMXBMI	Body Mass Index (kg/m**2)	26.629479	0.121334
LBDHDD	Direct HDL-Cholesterol (mg/dL)	53.052348	0.395636

Output 3. Output from PROC SURVEYMEANS using bootstrap replicate weights. The standard errors are not exactly the same as in Output 1; in data sets with larger numbers of clusters, the standard errors from the two methods will often be more similar.

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	67.7535616	0.97611876	69.41	<.0001
bmicat	-6.5053195	0.23977562	-27.13	<.0001
female	9.8081834	0.35622424	27.53	<.0001

Note: The denominator degrees of freedom for the t tests is 16.

Output 4. Output from PROC SURVEYREG using bootstrap replicate weights. The standard errors are not exactly the same as in Output 2, but are close.

CONCLUSION

The introduction of PROC SURVEYMEANS, PROC SURVEYREG, and PROC SURVEYSELECT in version 7 of SAS® software gave analysts powerful new tools for analyzing survey data. Subsequent developments have made the procedures more flexible and extended the analyses available for survey data. In this paper, we have shown how the SURVEY procedures may be used in conjunction with other procedures to create graphical displays of complex survey data and to extend the capacity of the replication variance methods to include the bootstrap. Many other graphical displays can be created using the same principles; Lohr (2010, p. 323), for example, shows how to

construct quantile-quantile plots for survey data. The adaptability of the SURVEY procedures make them useful for many analyses that were not anticipated in the original implementations, and we look forward to future developments.
.

REFERENCES

- Korn, E.L. and Graubard, B.I. (1999). *Analysis of Health Surveys.* New York: Wiley.
- Lohr, S. (2010). *Sampling: Design and Analysis, 2nd ed.* Boston: Brooks/Cole.
- Martin, M. (1990). On using the jackknife to estimate quantile variance. *Canadian Journal of Statistics,* 18, 149-153.
- McCarthy, P.J. and Snowden, C.B. (1985). The bootstrap and finite population sampling. *Vital and Health Statistics 2-95.* Washington, DC: U.S. Government Printing Office.
- Mukhopadhyay, P.K., An, A.B., Tobias, R.D., and Watts, D.L. (2008). Try, try again: Replication-based variance estimation methods for survey data analysis in SAS® 9.2. *SAS Global Forum 2008,* Available at www2.sas.com/proceedings/forum2008/367-2008.pdf.
- National Science Foundation, National Center for Science and Engineering Statistics (2011). *Characteristics of Scientists and Engineers in the United States: 2006.* Detailed Statistical Tables NSF 11-318, Available at www.nsf.gov/statistics/nsf11318/.
- Rao, J.N.K. and Wu, C.F. (1988). Resampling inference with complex survey data. *Journal of the American Statistical Association,* 83, 231-241.
- Shao, J. and Chen, Y. (1998). Bootstrapping sample quantiles based on complex survey data under hot deck imputation. *Statistica Sinica,* 8, 1071-1085.

ACKNOWLEDGMENTS

I wish to thank Tyler Smith and Rachael Biel for giving me the opportunity to present this paper. Sharon Hirabayashi of Westat tested and greatly improved the code for creating the plots and calculating the bootstrap weights, and her expertise has been invaluable. I am most grateful to Anthony An of SAS Institute, Inc. for many discussions and much cheerfully given help concerning the PROC SURVEYs. This work was partially supported by the National Science Foundation under grant DRL-0909630. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not reflect the views of the National Science Foundation.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Sharon L. Lohr
Enterprise: Arizona State University
Address: School of Mathematical and Statistical Sciences
City, State ZIP: Tempe, AZ 85287-1804
Work Phone: 480 965-4440
Fax: 480 965-8119
E-mail: sharon.lohr@asu.edu
Web: stat.asu.edu/~lohr