Paper 193-31

Wait Wait, Don't Tell Me… You're Using the Wrong Proc!

David L. Cassell, Design Pathways, Corvallis, OR

ABSTRACT

Statisticians and data analysts frequently have data sets which are difficult to analyze, with characteristics that break the underlying assumptions of the simpler analytical tools. But SAS® has tools to handle more complex problems. And, now that SAS has survey analysis procedures, there's no longer a need to pretend that survey samples should be analyzed using non-survey design tools. Now that large databases are commonly sampled for marketing, data mining, and scientific research, the need to use survey analysis procedures is increasing. In this paper, we'll look at several common situations and illustrate how the data are usually analyzed. Then we will show how the data ought to be analyzed using the SAS survey analysis procedures, and what the consequences of choosing the wrong procedures may be.

INTRODUCTION

In my consulting work, I often see people working as hard as they can to fight their way through a morass of messy data, with all manner of perils awaiting them. But all too often, these people are (inadvertently) taking the hardest path through that swamp. People now have access to large, complex databases of information that they need to slog through to get their work done. But all too often, those data represent sampling designs, instead of experimental designs.

Typically, data can fall into four general categories, based on how the data were gathered. Sample survey data, experimental design data, observational study data, and other. This is a pretty rough classification, but it brings out what we need to address in this paper.

Experimental design data have all the properties that we learned about in statistics classes. We have some underlying model that represents the structure and variability of the data. Classically, the data are going to be independent, identically-distributed observations with some known error distribution. This type of model can be extended in many ways, including non-linear structures and covariance matrices for non-independent data. But after dealing with all of these features, we are still left with errors which are supposed to be independent and identically distributed. Just as important, there is an underlying assumption that the data come to use as a finite number of observations from a conceptually infinite population - that's assumed in the way that we get the variance estimates.

Sample survey data, on the other hand, come from a finite target population, with known rules for getting the sample observations so that we can see how to derive features like sampling weights. The sample survey data do not have independent errors. The sample survey data do not come from a conceptually infinite population. The sample survey data may cover many small sub-populations, so we do not expect that the errors are identically distributed. This means that sample survey analysis tools don't work in quite the same way as classical linear models and general linear models methods.

It also means that using classical methods may give us the wrong answers. So we need to be sure we are using the right SAS procedures before we start analyzing our data.

WAIT WAIT DON'T TELL ME… YOU'RE TRYING TO GET A STANDARD ERROR

One of the most common problems is the simplest. In theory, anyway. When we have a database that is our population, and we have a sample from that database, we don't want to use the classical formulas for the mean and standard error. The most popular formula for the standard error of the mean assumes:

- Infinite population
- Simple random sampling without replacement for the sample data
- Independent observations
- Identically distributed errors

But we don't meet those assumptions when we start out with a finite population. The finite population leads us to a basic problem. Even with simple random sampling without replacement (all data point can be selected with equal probability and no data point can occur more than once in the sample), we do not have independence.

This means that the 'usual' formula for the standard error of the mean is not the one we should be using! We need to use a SAS procedure other than the ones we have used for decades: PROC MEANS, PROC SUMMARY, and PROC UNIVARIATE. We need to use PROC SURVEYMEANS.

We now see this problem cropping up everywhere. Marketers have databases of mailing lists, or databases of personal preferences made by people filling out questionnaires. Data miners have data warehouses of information on people, or choices made by people, and these are often sample survey data. Clinical researchers have databases from national health studies - more sample survey data. All of these are finite databases based on finite target populations, with sample survey issues lurking under the surface.

Let's see how a simple thing like our estimate of a mean and standard error can depend on sample survey procedures.

A SIMPLE EXAMPLE FOR MEANS AND STANDARD ERRORS

Let's build a nice little data set to show how the differences between sample survey estimates and classical estimates can differ.

```
data AuditFrame (drop=seed);
  seed=18354982;

  do i=1 to 600;
    if        i<101 then region='H';
      else if i<201 then region='S';
      else if i<401 then region='R';
      else               region='G';
    Amount = round ( 9990*ranuni(seed)+10, 0.01);
    output;
    end;

  do i=601 to 617;
    if        i<603 then region='H';
      else if i<606 then region='S';
      else if i<612 then region='R';
      else               region='G';
    Amount = round ( 10000*ranuni(seed)+10000, 0.01);
    output;
    end;

  do i = 618 to 647;
    if        i<628 then region='H';
      else if i<638 then region='S';
      else if i<642 then region='R';
      else               region='G';
    Amount = round ( 9*ranuni(seed)+1, 0.01);
    output;
    end;

  run;
```

This gives us a simple audit data set. We have a target population of 647 receipt amounts, classified by the company region. In the real world we would have thousands or millions of records. But this should suffice to illustrate our points.

If we need to perform a full audit, and it is too expensive to perform the full audit on every one of the receipts in the company database, then we need to take a sample. We want a simple random sample without replacement, but we want to sample the larger receipt amounts more frequently. After all, we have to spend the same amount of effort to validate the receipt whether the receipt is for a $2.99 box of paper clips or for a $3879.99 personal computer. So we will do PPS sampling: we will sample 'proportional to size'. That means that we choose a multiplier variable, and we make our choices based on the size of that multiplier. If the multiplier for receipt A is five times the multiplier for receipt B, then receipt A will be five times more likely to be selected than receipt B. For this example, we will use the

receipt amount as the multiplier. So a receipt for $800 will be eighty times more likely to be selected than a receipt for $10.

We can perform this sample selection using PROC SURVEYSELECT:

```
proc surveyselect data=AuditFrame  out=AuditSample3
    method=PPS
    seed=39563462
    sampsize=100;
  size Amount;
  run;
```

This gives us a weighted random sample of size 100. Now we'll just (artificially) create the data set of the audit results. We will have a validated receipt amount for each receipt. Ideally, the validated amount would be exactly equal to the listed amount in every case. (This doesn't happen in the real world.)

```
data AuditCheck3;
  set AuditSample3;
  ValidatedAmt = Amount;
  if region='S' and mod(i,3)=0
      then ValidatedAmt = round(Amount*(.8+.2*ranuni(1234)),0.01);
  if region='H' then do;
    if floor(Amount/100)=13 then ValidatedAmt=1037.50;
    if floor(Amount/100)=60 then ValidatedAmt=6035.30;
    if floor(Amount/100)=85 then ValidatedAmt=8565.97;
    if floor(Amount/100)=87 then ValidatedAmt=8872.92;
    if floor(Amount/100)=95 then ValidatedAmt=9750.05;
    end;
  diff = ValidatedAmt - Amount;
  run;
```

Now we have a data set that we can use to explore the differences between the 'usual' means and standard errors, and those from sample survey analysis.

WAIT WAIT… THAT'S THE WRONG STANDARD ERROR

The naïve approach would now be to use PROC MEANS, and just compute the mean and standard error. But we sampled proportional to size. That means that we have sampling weights now. Should we merely use the WEIGHT statement in PROC MEANS to adjust for this?

Of course not. If that was all we had to do, then why would we be going to all this trouble? The WEIGHT statement in PROC MEANS and PROC SUMMARY allows a user to give some data points more emphasis, much like using a FREQ statement. But that isn't the right way to address the weights we have here. We built a sample using a specific sample design, and we have sampling weights which have a real, physical meaning. A sampling weight for a given data point is the number of receipts in the target population which that sample point represents.

Sampling weights should not be scaled, as we frequently do with weights in experimental design settings. Sampling weights must be at least 1. Think about it. A sampling weight of one means that the point represents only itself, and no other data form the target population. A sampling weight cannot be less than 1, since it cannot represent something less than itself. And sampling weights are the inverse of the inclusion probabilities for the sampling design. Sampling weights cannot be less than 1, any more than inclusion probabilities can be greater than 1.

So let's try analyzing our sample data in three ways. PROC MEANS with and without weights, and PROC SURVEYMEANS. Here is the standard PROC MEANS code, with a request in the PROC statement for the sample mean, the standard error of the mean, and the associated confidence interval (a 95% confidence interval by default).

```
proc means data=AuditCheck3 mean stderr clm;
  var ValidatedAmt  diff;
  title 'PROC MEANS with no weights';
  run;
```

The way to add weights to PROC MEANS is through the use of the WEIGHT statement.

```
proc means data=AuditCheck3 mean stderr clm;
  var ValidatedAmt  diff;
  weight SamplingWeight;
  title 'PROC MEANS with weights';
  run;
```

However, the weights in PROC MEANS are not survey sample weights with the associated meaning. In order to get survey sample weights into the process, with the proper standard error, we use PROC SURVEYMEANS.

```
proc surveymeans data=AuditCheck3 mean stderr clm total=647;
  var ValidatedAmt  diff;
  weight SamplingWeight;
  title 'PROC SURVEYMEANS - the correct way to do it';
  run;
```

We can see that the code for PROC SURVEYMEANS looks almost exactly like that of PROC MEANS when we use the sampling weights. The primary difference is the inclusion of the TOTAL= option. PROC SURVEYMEANS allows us to compute a Finite Population Correction Factor and adjust the error estimates accordingly. This factor adjusts for the fact that we already know the answers for some percentage of the finite population, and so we really only need to make error estimates for the remainder of that finite population.

Now, when we compare results, we can see that we don't get the same results, even on something as simple as computing a standard error when we have a (PPS) random sample. Here are the results for DIFF, the difference between the validated and listed amounts. In a real audit, we would want to check that there is not a significant difference from zero.

Statistics for the variable DIFF

Procedure	Mean	Std Error of Mean	Lower 95% CL for Mean	Upper 95% CL for Mean
SURVEYMEANS	-34.0081624	15.3189950	-64.4043730	-3.6119520
MEANS	-42.2456000	21.9552988	-85.8096760	1.3184760
MEANS w/ weights	-34.0081624	17.3182627	-68.3713528	0.3550279

We can see several key points right away. One of them is that PROC MEANS with the WEIGHT statement does get the correct estimate of the mean. But it does not get the correct standard error. Which means that the confidence interval for the variable is off. While the correct confidence interval shows a distinct bias from zero, both the confidence intervals computed from PROC MEANS miss this. Also notice that PROC MEANS without the WEIGHT statement does not even get the correct estimate of the sample mean.

Clearly, there can be serious consequences to using the wrong procedure when we need to compute descriptive statistics from sample survey data. And, when the syntax for PROC SURVEYMEANS is so much like that of PROC MEANS, there really is no reason not to switch to the right proc whenever we have survey data.

WAIT WAIT DON'T TELL ME… YOU'RE DOING LINEAR MODELS

As we move to more complicated statistical analyses, the need for more appropriate analyses continues to grow. Linear regression, analysis of variance, and other basic linear models are typically tackled using PROC REG and PROC GLM. But if the data come from sample surveys, then the results from these procs may be misleading. Or worse.

A number of authors have studied the problem. DuMouchel and Duncan (1983), as well as Winship and Radbill (1994) have concluded that using survey sampling weights in a regular linear regression will give you the wrong standard errors. Winship and Radbill concluded that the bias on the standard errors was hard to predict: the stated standard errors might be higher or lower than they should be. This makes error adjustment essentially impossible. We can see this with a simple example out of the SAS Online Documentation for PROC SURVEYREG, which has data with which we can experiment.

So let's start. You come to me and say: "I have data from a sample survey. It's stratified by regions within Iowa and Nebraska. I need to regress on farm area, with separate intercept and slope for each state."

I, of course, say "Wait, wait, don't tell me. You're using PROC GLM for this, right?" And you nod.

```
data Farms;
  input State $ Region FarmArea CornYield Weight;
  datalines;
   Iowa     1 100  54 33.333
   Iowa     1  83  25 33.333
   Iowa     1  25  10 33.333
   Iowa     2 120  83 10.000
   Iowa     2  50  35 10.000
   Iowa     2 110  65 10.000
   Iowa     2  60  35 10.000
   Iowa     2  45  20 10.000
   Iowa     3  23   5  5.000
   Iowa     3  10   8  5.000
   Iowa     3 350 125  5.000
   Nebraska 1 130  20  5.000
   Nebraska 1 245  25  5.000
   Nebraska 1 150  33  5.000
   Nebraska 1 263  50  5.000
   Nebraska 1 320  47  5.000
   Nebraska 1 204  25  5.000
   Nebraska 2  80  11 20.000
   Nebraska 2  48   8 20.000
   ;
  run;

data FarmsByState; set Farms;
  if State='Iowa' then do;
    FarmAreaIA=FarmArea ; FarmAreaNE=0 ;
    end;
  else do;
    FarmAreaIA=0 ; FarmAreaNE=FarmArea;
    end;
  run;

proc glm data=FarmsByState;
  class State;
  model CornYield = State FarmAreaIA FarmAreaNE / noint solution ;
  title 'Without weights';
  run;

proc glm data=FarmsByState;
  class State;
  model CornYield = State FarmAreaIA FarmAreaNE / noint solution ;
  weight Weight;
  title 'With weights';
  run;
```

But we don't really want to use PROC GLM. We already know that this is a stratified sample of Iowa and Nebraska farms. We know that simple random samples of farms were taken with five strata. We know that the strata were designed as subsets of the two states, so that three of the regions are contained within Iowa, and the other two are contained within Nebraska. And we know that the weight variable is not an analytic weight, but a real sampling weight. Since the farms are sampled with equal probability within each stratum, we also know that the weights are the number of farms from the given stratum of the target population which are represented by each sample point in that stratum.

So what should we do? We know there isn't any "proc surveyglm" in SAS. But there is PROC SURVEYREG. And PROC SURVEYREG does general linear models for sample survey data. That includes some of the standard tasks we usually want from PROC GLM. This means that we can use PROC SURVEYREG for more than just linear regressions. We can use it for analysis of variance, and for analysis of covariance. PROC SURVEYREG comes with CONTRAST and ESTIMATE statements, as well.

So how do we analyze these data? First, we need to specify any stratum and cluster information, and provide a weight variable. We have that information ready. The stratum variables are STATE and REGION, while the sub-population totals for the five strata are in a separate data set named StratumTotals. The sampling weight is WEIGHT. There is no clustering, so we do not need a CLUSTER statement. So we have the following code.

```
data StratumTotals;
  input State $ Region _TOTAL_;
  datalines;
  Iowa 1 100
  Iowa 2 50
  Iowa 3 15
  Nebraska 1 30
  Nebraska 2 40
  ;
  run;

proc surveyreg data=FarmsByState total=StratumTotals;
  class state;
  model CornYield = State FarmAreaIA FarmAreaNE / noint covb solution;
  strata State Region;
  weight Weight;
  run;
```

Notice that the stratum totals are in a variable named _TOTAL_ with an underscore on each side. That's a special variable name used by the survey sample analysis procs to hold stratum totals. You must use that variable name when feeding stratum or sub-population totals to any of these procs. Just as we used the TOTAL= option to give a fixed population total to PROC SURVEYMEANS before, we can also use TOTAL= to feed a table of stratum sizes to any of the survey analysis procs.

Now let's look at some results from the output of these runs.

	PROC SURVEYREG	PROC GLM using sampling weights
R squared	0.818	.930
Root Mean Square Error	11.98	42.14
Degrees of freedom for error	15	14

Note that the two procs do not even use the same degrees of freedom for error. So it is not surprising that the two do not come up with the same error estimates! PROC GLM (and PROC REG, for that matter) will use the classical degrees of freedom for error of n-p , where n is the number of observations and p is the number of parameters being fit in the model (including the intercept). PROC SURVEYREG will use design-based estimates. That includes degrees of freedom for error which are n-k, where k is the number of strata in the design. PROC GLM with the WEIGHT statement will find the same parameter estimates as PROC SURVEYREG will. But the standard errors may be quite different. And that can affect everything else: the test statistics, the p-values, the confidence intervals, and so on.

Let's look at the parameter estimates for our model. We'll list one parameter estimate, since both procs give the same results:

	Parameter estimate	PROC SURVEYREG standard error	PROC GLM standard error
Intercept IA	5.28	5.27	5.41
Intercept NE	0.65	1.70	8.96
Slope IA	0.407	0.065	0.056
Slope NE	0.146	0.020	0.057

Note that there is no clear pattern in the relative sizes of the standard errors. The PROC GLM estimate of the standard error may be about right. It may be drastically high. And it can be lower than the correct value. That makes adjustment after the fact essentially impossible.

That leaves us with only one option. We don't use PROC REG or PROC GLM when we really need to be using PROC SURVEYREG.

WAIT WAIT DON'T TELL ME… YOU'RE DOING A LOGISTIC REGRESSION

The problem we discussed above with standard errors for linear regression does not go away when we move to more complex linear forms. Winship and Radbill (1994) stated that the problem continues in probit regression, logit regression, and other generalized linear models. So let's look at a typical problem. (Note: the data for this example come from the SAS OnlineDoc examples for PROC SURVEYLOGISTIC.)

You come to me and tell me about your project. "We did a marketing survey. It's just a random sample. We took 300 students from each of the classes: freshman, sophomore, junior, and senior classes. So we can do basic logistic regression, right?"

At this point, I get to say, "Wait wait, don't tell me… You're using the wrong proc, aren't you?"

First, it is important to realize that the survey was not done as a simple random sample. The 'simple random sampling without replacement' was done independently within each class of students. That means that the marketing survey was actually done as a stratified random sample. (Note that Example 69.1 in the SAS OnlineDoc for SAS 9.1.3, from whence these data come, has a typo in the title. This is a stratified sample, not a stratified cluster sample.)

Here are the data and the PROC SURVEYLOGISTIC code, as well as the corresponding (but incorrect) PROC LOGISTIC code:

```
proc format;
  value Design 1='A' 2='B' 3='C';
  value Rating 1='dislike very much'
               2='dislike'
               3='neutral'
               4='like'
               5='like very much';
  value Class  1='Freshman' 2='Sophomore'
               3='Junior'   4='Senior';
  run;
data Enrollment;
  format Class Class.;
  input Class _TOTAL_;
  datalines;
  1 3734
  2 3565
  3 3903
  4 4196
  ;
  run;
```

```
data WebSurvey;
  format Class Class. Design Design. Rating Rating. ;
  do Class=1 to 4;
    do Design=1 to 3;
      do Rating=1 to 5;
        input Count @@;
        output;
        end;
      end;
    end;
  datalines;
  10 34 35 16 15   8 21 23 26 22   5 10 24 30 21
   1 14 25 23 37  11 14 20 34 21  16 19 30 23 12
  19 12 26 18 25  11 14 24 33 18  10 18 32 23 17
   8 15 35 30 12  15 22 34  9 20   2 34 30 18 16
  ;
  run;

data WebSurvey;
  set WebSurvey;
  if Class=1 then Weight=3734/300;
  if Class=2 then Weight=3565/300;
  if Class=3 then Weight=3903/300;
  if Class=4 then Weight=4196/300;
  run;

proc logistic data=WebSurvey;
  freq Count;
  class Design;
  model Rating (ref='neutral') = Design /link=glogit;
  weight Weight;
  run;

proc surveylogistic data=WebSurvey total=Enrollment;
  freq Count;
  class Design;
  model Rating (ref='neutral') = Design /link=glogit;
  stratum Class;
  weight Weight;
  run;
```

The two procedures will generate the same point estimates for the parameters. But the standard errors, and hence the statistics developed from the data, may be markedly different. This can have serious consequences for the conscientious analyst. Since the parameter estimates are the same in both procedures, the odds ratio estimates are the same. But the standard errors differ, so the Wald confidence limits for the odds ratios are not the same. Let's look at a small subset of the output: the ratings for design B vs. design C. Remember that the models use the rating 'neutral' as the reference category.

Ratings for Design B vs. Design C	Odds Ratio point estimate	95% Wald confidence interval from PROC SURVEYLOGISTIC	95% Wald confidence interval from PROC LOGISTIC
Dislike very much	1.615	(0.975, 2.675)	(1.395, 1.870)
Dislike	0.984	(0.659, 1.471)	(0.877, 1.104)
Like	1.218	(0.838, 1.768)	(1.093, 1.357)
Like very much	1.389	(0.925, 2.086)	(1.235, 1.562)

Notice the differences. The correct standard errors from PROC SURVEYLOGISTIC tell us that all the confidence intervals contain 1. We really can't make a distinction between Design B and Design C, based on the stated ratings. However, the standard errors from PROC LOGISTIC would lead us to a completely different conclusion.

WAIT WAIT DON'T TELL ME… YOU'RE DOING A CHI-SQUARED TEST

When we move on to categorical data analysis techniques like the classical chi-squared test, we run into different difficulties. The standard chi-squared test needs to be modified in order to adjust for the design effects of our sample survey design. This means that we have to use PROC SURVEYFREQ if we want to get the right p-values and make the right statistical decisions. We can show this using the SIS_Survey data set from the SAS OnlinDoc for PROC SURVEYFREQ .

Let's try this once again. You come to me and tell me about your project. "We have a huge data set. We picked schools from three states. Then we went to each school and interviewed three teachers and two staff members. We did a chi-squared test to see if the two departments differed in their responses. The faculty are insisting on a p-value less than 0.01, and we got it. A chi-squared of 17.23 with 4 degrees of freedom. That's a p-value of 0.0017 . Right? Here's our data."

```
data collapsed_SIS;
  length department $ 14;
  input department $ response frequency;
  datalines;
Faculty        1 209
Faculty        2 203
Faculty        3 346
Faculty        4 254
Faculty        5 098
Admin/Guidance 1 095
Admin/Guidance 2 123
Admin/Guidance 3 235
Admin/Guidance 4 201
Admin/Guidance 5 086
  ;
  run;

proc freq data=collapsed_SIS;
  tables department * response / chisq;
  weight frequency;
  run;
```

Then we begin looking. Are the data from a survey? Yes.

How did the schools get chosen? They were split into categories by state and customer status before any sampling was done. Then lists of schools were selected proportional to the enrollment.

And after that? For each selected school five people were selected, as we said before.

That's when I get to say, "Wait wait, don't tell me… You're using the wrong proc."

The problem is simple. These data are from a probability sample. A survey sample. The sample was split into six groups (three states x two customer categories), and sampling of schools (proportional to enrollment) was done separately in each group. Then each selected school had five people (three in one department, two in the other) randomly chosen for the questionnaire. That tells us that this is really a two-stage sample. The first stage is a stratified cluster sample. So this needs to be analyzed using PROC SURVEYFREQ instead.

When we analyze these data using PROC SURVEYFREQ (or any of the survey analysis procs), we only feed in the stratification variables and cluster variables for the first stage of sampling. Stratification and cluster variables used in later stages of the sample do not need to be given to the procedure, due to the way that variance estimation is performed. This can simplify the information which we have to track.

So let's do this properly, using PROC SURVEYFREQ and the full SIS_Survey data set:

```
proc surveyfreq data=SIS_Survey nosummary;
  tables Department * Response / wchisq;
  strata State NewUser;
  cluster School;
  weight SamplingWeight;
  run;
```

When we analyze the data now, we need to use a Wald chi-squared test instead of the classical chi-squared test of categorical data analysis. And we get different results.

We have a Wald Chi-squared statistic of 11.44 , which gives us an adjusted F value of 2.84 , which leads to a p-value of 0.0243 . If the faculty are insistent on a p-value of 0.01 , then you would not find the strongly significant result of the original (incorrect) chi-squared test.

WAIT WAIT DON'T TELL ME… YOU'RE DOING MIXED MODELS

So far, things have been simple. When we were computing means and standard errors for survey samples (or features we haven't discussed, like confidence intervals for means or hypothesis tests for means), we used PROC SURVEYMEANS. When we were doing regression or analysis of variance or other typical linear models, we used PROC SURVEYREG. When we were performing linear logistic regressions, we used PROC SURVEYLOGISTIC. (PROC SURVEYLOGISTIC also fits probit models, cumulative logit models, generalized logit models, multinomial models, and a couple more.) When we wanted a chi-squared test for a table of categorical variables, we use PROC SURVEYFREQ.

But other models are often needed. For example, there is no SURVEYLIFEREG. We often see people turn to some of the standard SAS procs in order to cope with the data they have. And we often see mistakes being made. Let's look at a couple more examples. The names have been changed to protect the (statistically) innocent.

Mister X says to us, "I have to use PROC NLMIXED because I have random effects and I want to do a logistic regression. I gathered data from 400 hospitals across the country. I could only afford to send trained field workers to those 400 hospitals. I picked them randomly from a list of all the licensed hospitals in the country, but I want to make inferences about all the hospitals. That's a random effect, right? Then my workers went to each hospital on the list and gathered all the available data on the color preferences of each of the hospital administrators. I mean, there must be some reason why they all like that nasty greenish color that looks like Morticia Addams picked it out. So we'll perform regressions with the data. That means we'll need PROC NLMIXED. Right?"

That's when we get to say, "Wait wait, don't tell me… You're using the wrong proc."

This is not a case of random effects vs. fixed effects. This is a cluster sample. This is a data set suitable for PROC SURVEYLOGISTIC. The hospital does not represent a random effect. It represents a cluster variable. And once we use survey sample analysis with a CLUSTER statement, then we can indeed make inferences about the entire population of hospitals nation-wide.

No sooner than we straighten out Mister X, then Madame Y comes to us.

She says, "I need your help. I have to work with PROC MIXED. I wish to do data mining, but I find myself buried in hierarchical linear models. We are trying to predict insurance costs. But our data are so complex. We have data from a subset of all the different insurance agencies. So we have agency-wide characteristics. Then the agencies split their customers up into market segments. And we have subsets of the segment-level data. And finally, we have samples of the customers themselves, with insurance costs and customer-level data. How can we handle this without hierarchical linear models? And how do we determine intra-class coefficients so we can decide how to build the models?"

That's when we get to say, once again, "Wait wait, don't tell me… You're using the wrong proc."

This does not have to be a hierarchical linear model. The data are drawn from well-defined target populations. We can compute the sampling weights after the fact. We can see the three different levels of sampling. So we have a multi-stage sample. We don't have to use hierarchical linear models, because the data are really a probability survey.

We can use PROC SURVEYREG with a CLUSTER statement and a cluster variable that only has to represent the first stage of clustering.

CONCLUSION

The statistical methodologies above are important. Don't throw out tools like linear regression, or logistic regression, or mixed models, or hierarchical linear models, or any of the others discussed above. But don't use them mindlessly, either. Use them when it is appropriate. But scrap them when they should **not** be used.

These examples are just illustrations of the types of problems people wade into every day. When we forget that we have survey sample data, we inevitably end up struggling in a quicksand of statistical problems. Using the right SAS procs can make our path a much simpler one. Using the right SAS procs is even easy. The SAS survey analysis procs have been designed to look so much like their classical counterparts that programming with them and interpreting their output is surprisingly easy.

REFERENCES

Cochran, W.G., 1977. Sampling Techniques. Third edition, New York: John Wiley & Sons, Inc.

DuMouchel, W. H., and G. Duncan. 1983. "Using Sample Survey Weights in Multiple Regression Analyses of Stratified Samples." *Journal of the American Statistical Association* 78(383):535-542.

SAS OnlineDoc® 9.1.3, Copyright © 2002-2005, SAS Institute Inc., Cary, NC, USA; All rights reserved. Produced in the United States of America.

Winship, C., and L. Radbill. 1994. "Sampling Weights and Regression Analysis." *Sociological Methods and Research* 23(2):230-257.

ACKNOWLEDGMENTS

The author would like to thank the readers of SAS-L (the international SAS mailing list) and comp.lang.soft-sys.sas (the UseNet group covering SAS topics) for putting up with assorted kvetching on this subject for several years now.

The title of this paper, as well as the author of this paper, has no connection with National Public Radio (NPR) or any of its programming.

Three of the four data collations in this paper come directly from the SAS OnlineDoc for SAS 9.1.3 .

CONTACT INFORMATION

The author welcomes questions and comments. He can be reached at his private consulting company, Design Pathways:

David L. Cassell
Design Pathways
3115 NW Norwood Pl.
Corvallis, OR 97330
Cassell.David@epamail.epa.gov
541-754-1304