Paper 93

When the Mean Isn't Enough: Methods for Assessing Individual Differences using SAS

Melissa L. McTernan, California State University, Sacramento

ABSTRACT

Many programs of research are focused on understanding the "average" individual, leading to the use of statistical methods that emphasize means (e.g. mean differences, mean trajectories, etc.). However, "typical" values are often insufficient and sometimes not representative of any one individual (or one group of people) in a sample. In this paper, I will discuss how SAS is a flexible tool for researchers interested in individual differences. I will include methods of data visualization as well as methods of statistical analysis when the individual is a unit of interest. I will primarily focus on longitudinal data analysis and how individuals change across time. This paper will include syntax for PROC MIXED, PROC NLMIXED, PROC GLIMMIX, PROC SGPLOT, and PROC SGPANEL.

INTRODUCTION

There are certainly times when understanding or describing the average individual in a population is valuable. However, "typical" may sometimes not representative of any one individual (or one group of people) in a sample. At minimum, these typical values are often insufficient and uninteresting. In such cases, it is not satisfying to use data analytic approaches and data visualization methods that only represent the mean, and other methods may be more appropriate. SAS Software is a flexible tool for researchers interested in individual differences. Using SAS procedures, researchers and other analysts can access methods of data visualization and statistical analysis that are appropriate for understanding individual differences in a given outcome.

Understanding individual differences across time or across repeated measures is a common goal of psychological and behavioral research. In this paper, I will primarily focus on longitudinal data analysis and SAS procedures that support the goal of understanding how individuals change throughout across multiple observations.

Mixed-effects modeling (sometimes called random-effects or multilevel modeling) is a tool for analyzing longitudinal data when individual differences are of interest. SAS PROC MIXED and PROC NLMIXED are two SAS procedures that would commonly be used for such a data approach. These procedures were built for mixed-effects models and nonlinear mixed-effects models, respectively. PROC MIXED is limited to linear models, but PROC NLMIXED may be used for linear or nonlinear modeling needs. In the next section I will explain mixed-effects modeling and the distinction between linear and nonlinear mixed-effects models.

Using a mixed-effects model to capture individual differences in an outcome across time is extremely valuable. This approach can increase understanding about how well the average curve fits all the individuals in the sample. In other words, provides information about variability in the way that people change across time. The variability in growth parameters are estimated using random effects. A model may be parameterized to include a random intercept, random slope(s), or both. Sometimes it may be helpful to visualize the data for a more intuitive understanding of how individuals change across time, and how they differ in the ways they change across time. In this paper I will give an overview of SAS procedures for mixed effects modeling and for visualizing longitudinal data with attention to individual differences in the growth parameters. I used data from the National Longitudinal Survey of Youth (NLSY97) public-use database to demonstrate the use of SAS procedures for assessing individual differences. I am using three waves of data collection, 2006-2008, and specifically at three variables: (1) Did the participant have health coverage this year?; (2) What is the participant's overall outlook on life on a scale of 0, being extremely negative, to 10, being extremely positive; and (3) a measure of general health, measured on a continuous scale. The NLSY97 survey is sponsored and directed by the U.S.

Bureau of Labor Statistics and conducted by the National Opinion Research Center at the University of Chicago, with assistance from the Center for Human Resource Research at The Ohio State University.

Note that the examples in this paper are to demonstrate SAS syntax for observing individual differences and are not necessarily theoretically motivated. Therefore, any finding presented in this paper should be interpreted with caution and replicated with formal theory driven hypotheses.

DATA TRANSFORMATION: WIDE TO LONG FORMAT

Many longitudinal public use datasets and other databases containing repeated measures are structured in "wide" format, in which there is a single record per person. In wide format (also called single record format) there is a new variable column for each measurement occasion. For example, if variable *X* is measured for 100 individuals across 3 measurement occasions, the data in wide format would contain 100 rows, one for each individual. It would contain 4 columns: ID, *X* at Time 1, *X* at Time 2, and *X* at Time 3. Many procedures in SAS, including those discussed in this paper topic, require that the data be in "long" format, in which there are multiple rows per individual, with the number of rows per individual matching the number of measurement occasions. In the previous example, the data in long format would include 300 rows (100*3) and 3 columns: ID, *X,* and a variable that is an indicator of the measurement occasion or time. See Figure 1.

ID	T1_X	T2_X	T3_X
1	10	11	14
2	8	8	10
3	10	9	12
.	.	.	.
.	.	.	.
N	13	11	12

vs.

ID	Time	X
1	1	10
1	2	11
1	3	14
2	1	8
2	2	8
2	3	10
3	1	10
3	2	9
3	3	12
.	.	.
.	.	.
N	1	13
N	2	11
N	3	12

Figure 1. Comparison of wide (left) and long (right) formatted data

Transforming data from wide format to long format can be a tedious chore. There are multiple methods available in SAS to assist with this task, and each has its limitation. Using a series of ARRAY statements inside of a DATA step is one straightforward and efficient approach. The following syntax demonstrates how to transform NLSY97 data from wide format to long format. The original dataset, called "NLSY1" is in wide format, and it is transformed into a long dataset titled "NLSYLong" that will include three rows per person, with each row containing the variables Age, Time, Healthcare Coverage, Life Rating, and General Health Score at years 2006, 2007, and 2008.

```
DATA NLSYLong ;
  SET NLSY1 ;

  ARRAY newAge(06:08) Age06-Age08 ;
  ARRAY newTime(06:08) Time06-Time08 ;
  ARRAY newGenH(06:08) GenH06-GenH08 ;
```

```
  ARRAY newCover(06:08) Cover06-Cover08 ;
  ARRAY newLifeRate(06:08) LifeRate06-LifeRate08 ;

  DO year = 06 to 08 ;
    Age = newAge(year) ;
    Time = newTime(year) ;
    GenH = newGenH(year) ;
    Cover = newCover(year) ;
    LifeRate  = newLifeRate(year) ;
    OUTPUT ;
  END ;

    DROP Age06-Age08 Time06-Time08 GenH06-GenH08 Cover06-Cover08
    LifeRate06-LifeRate08 ;

RUN ;
```

Note that this approach would be inefficient for extremely large datasets, for which a PROC TRANSPOSE might be more appropriate. PROC TRANSPOSE is efficient, but is limited in that it can only transpose a single variable at a time.

VISUALIZING INDIVIDUAL DIFFERENCES ACROSS TIME

If a dataset contains multiple observations for individuals across time (i.e. repeated measures or longitudinal studies), plotting the datapoints can provide a wealth of information, and sometimes may inform the data analytic approach. PROC SGPLOT is a data visualization tool that proffers a diverse range of capabilities, because it allows you to build a chart by adding layers of graphics. Simply, the procedure creates a base of four axes, and you options build chart components from 16 different kinds of plots. For example, a user could create a histogram with a density curve, or a scatterplot with a fitted line.

Given its flexibility, PROC SGPLOT is often used to visualize complex longitudinal data. Of the range of possibilities, I demonstrate a few in this paper, starting with a spaghetti plot. The spaghetti plot displays data across time of every individual in the sample. The code is relatively simple. The first piece of code build the base spaghetti plot:

```
PROC SGPLOT DATA = NLSYLong (RENAME=(GenH=GeneralHealth)) NOAUTOLEGEND ;
    YAXIS min = 0 max = 8;
    REG x = Time y = GeneralHealth
    / group = ID nomarkers LINEATTRS = (COLOR = gray PATTERN = 1
          THICKNESS = 1) ;
```

and the average trend line is added as a layer to the plot using the code below:

```
    REG x=Time y=GeneralHealth
    / NOMARKERS LINEATTRS = (COLOR= red PATTERN = 1 THICKNESS = 3) ;

RUN ;
```

The preceding syntax results in the spaghetti plot shown in Figure 2. Time points 1, 2, and 3 correspond to years 2006, 2007, and 2008, respectively.

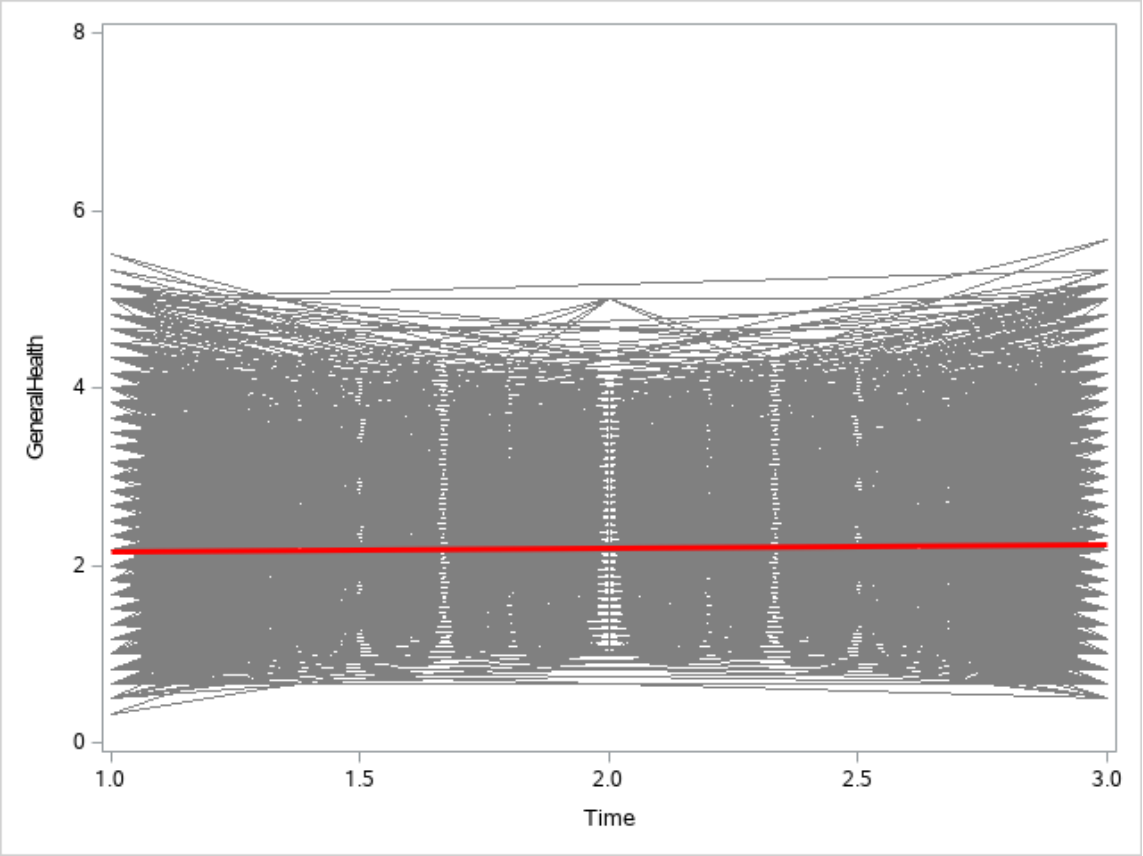

Figure 2: Spaghetti plot of General Health across study time points, with an overlay (red line) exhibiting the average trend

With a large amount of data, it can be difficult to see individual trends, but we can see that in general, it seems as though general health is not changing across time for this sample. However, what this plot also tells us is exactly how much people vary in their typical General Health scores. The average score in 2006 is about 2.2, but the scores range from about 0.2 to about 7.5. That range is visible in the spaghetti plot.

Sometimes you may want to use data visualization to observe group differences in trends across time. You can use PROC SGPANEL to create multiple spaghetti plots using a grouping variable, allowing you to observe how individuals change across time relative to people in other groups. In this example, I plot individual curves of Life Rating across time, from 2006-2008, and group the plots by health care coverage. The result will show whether individuals with health care coverage exhibit a change in life satisfaction across time differently than those individuals without healthcare coverage. Note that the syntax is similar to that we used in the PROC SGPLOT procedure, with an added option for a grouping variable. The PROC SGPANEL syntax is:

```
PROC SGPANEL DATA=NLSYLong NOAUTOLEGEND ;
 PANELBY Cover;
  REG x=Time y=LifeRate
    / group = ID NOMARKERS LINEATTRS = (COLOR = gray PATTERN = 1
          THICKNESS = 1) ;
   REG x=Time y=LifeRate
    / NOMARKERS LINEATTRS = (COLOR= red PATTERN = 1 THICKNESS = 3) ;
RUN ;
```

The plot generated by PROC SGPANEL is shown in Figure 3.

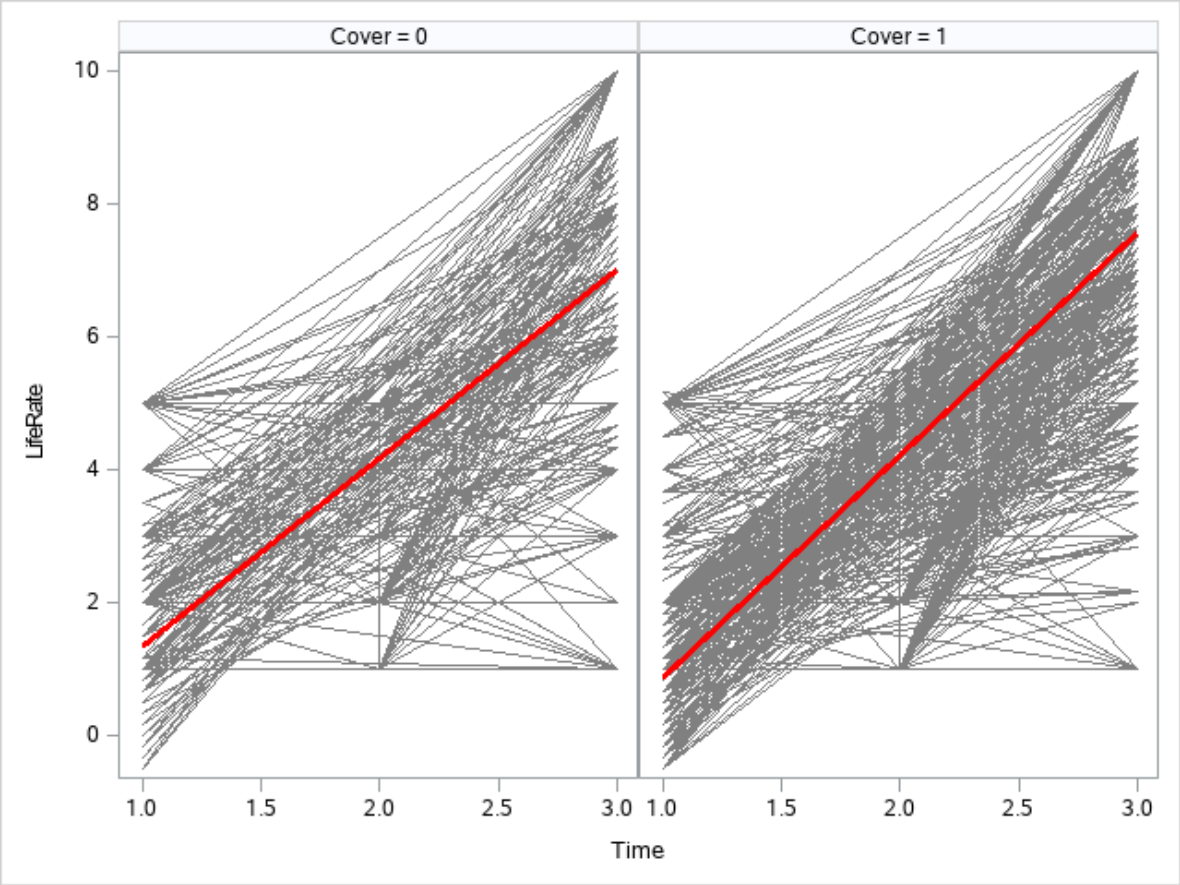

Figure 3. PROC SGPANEL result exhibiting side by side plots comparing individuals without (left) and with (right) health care coverage and how their life ratings change across time

The plot resulting from the SGPANEL procedure shows us that people with healthcare coverage seem to have a slightly steeper rate of increase in life satisfaction across time, on average. The patterns of within-group difference look similar across the two groups. Again, the spaghetti plot demonstrates the variation around the average trend lines that we would not see if we had only plotted the group averages.

You may also choose to compare the coverage groups on a single plot, which can be helpful when the differences are hard to see when side-by-side. This syntax is very simple in PROC SGPLOT:

```
PROC SGPLOT DATA = NLSYLong;
  REG x=Time y=LifeRate / group=Cover ;
RUN;
```

and results in the plot shown in Figure 4. Note that this plot doesn't offer as much information about individual differences, but does give us a clearer picture of group differences. The gray dashed line in the figure shows the average trend in Life Rating across time. It is clear that if this were the only line we plotted, we would be missing the information that individuals without healthcare coverage gain life satisfaction at a slower rate than those with health insurance.

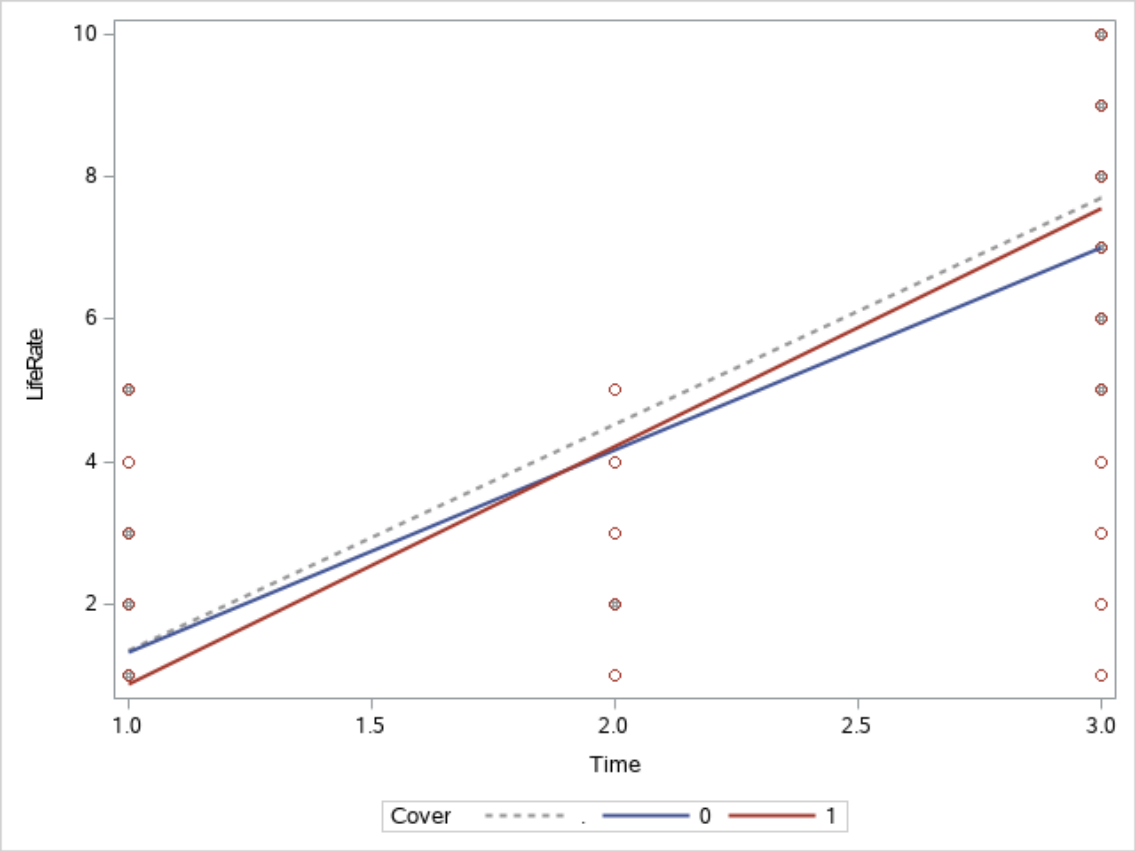

Figure 4. PROC SGPLOT results displaying average Life Rating across time for those with and without health care coverage

These examples using the SGPLOT and SGPANEL procedures demonstrate only a few of the many capabilities that SAS has to offer when it comes to going beyond the "average individual" in data visualization. In the next section, I will discuss several SAS procedures for modeling data with the same goal in mind.

ACCOUNTING FOR INDIVIDUAL DIFFERENCES ACROSS TIME

There are multiple SAS procedures for fitting mixed-effects models to data. In this paper, I cover a non-exhaustive list, including PROC MIXED, PROC NLMIXED, and PROC GLIMMIX.

PROC MIXED

SAS PROC MIXED is a procedure that accommodates linear mixed-effect models. It is flexible in that it allows for different covariance structures, and it also allows for a fixed effects analysis. This can be useful in comparing models to test whether a random effect is useful in describing the data. First, I demonstrate a linear regression model to predict Life Ratings from participant age, whether the participant has health care coverage. The model also includes an interaction term for the two predictors. This model only contains fixed effects, so the PROC MIXED syntax is without a REPEATED or RANDOM statement:

```
PROC MIXED DATA = NLSYLong noclprint covtest;
    CLASS ID;
    MODEL LifeRate = Age Cover Age*Cover / solution ddfm=bw;
RUN;
```

Note that the results of the linear model only describe the "typical" person in the population. In other words, this model allows us to describe how age and health care coverage affect life-satisfaction for the

average individual. If it is reasonable to assume that individuals may differ in life satisfaction independent of the predictors, a random intercept model might be appropriate. The PROC MIXED syntax has a REPEATED statement that easily accommodates random effects. (Note: random effects can also be included in the model by adding a RANDOM statement. This will produce equivalent R and G matrices for a linear mixed model.) See below:

```
PROC MIXED DATA = NLSYLong noclprint covtest;
   CLASS ID Cover;
   MODEL LifeRate = Age Cover Age*Cover / solution ddfm=bw;
   REPEATED /sub=ID type=UN;
RUN;
```

Note that the output from this model, a linear random intercept model, includes an additional parameter estimate. In addition to a fixed intercept and fixed slope for each linear predictor, this model includes a random intercept parameter that estimates the variance in level of Life Rating across individuals.

PROC NLMIXED

The MIXED procedure is limited in that it can only accommodate linear relationships between the predictors and the outcome. PROC NLMIXED is similar in functionality to PROC MIXED, but it allows for nonlinear models. In the example below, I use a generalized linear mixed effects model to the likelihood of having healthcare coverage across time. The outcome “Cover” is a binary variable, so I use the built in option for a “binary” distribution in the MODEL statement. The complete syntax is:

```
PROC NLMIXED DATA = NLSYLong;
   eta    = beta0 + beta1*Time + u;
   expeta = exp(eta);
   p      = expeta/(1+expeta);
   MODEL Cover ~ binary(p);
   RANDOM u ~ normal(0,s2u) subject=ID;
   PREDICT eta out=eta;
   ESTIMATE '1/beta1' 1/beta1;
RUN;
```

Lines two through four of the PROC NLMIXED syntax in this example define “eta” as being a linear function of the growth parameters, and we define “p” as a function of “eta.” This is similar to using a logit link function for the binomial generalized linear model.

PROC GLIMMIX

In addition to PROC MIXED and PROC NLMIXED, a flexible tool for fitting mixed models in SAS is PROC GLIMMIX. This procedure is designed for generalized mixed effects models (i.e. models for data that do not necessarily follow a Normal distribution). The syntax for a generalized linear model predicting health care coverage across time from life rating scores is shown below:

```
PROC GLIMMIX DATA = NLSYLong noclprint;
   CLASS ID ;
   MODEL Cover = Time LifeRate/CL DIST=BINARY
         LINK=LOGIT SOLUTION ODDSRATIO;
   RANDOM INTERCEPT / SUBJECT=ID TYPE=VC;
   COVTEST / WALD;
RUN;
```

The results of this syntax will be similar to results in PROC NLMIXED in most cases.

CONCLUSION

This paper demonstrates multiple approaches for data visualization and data analysis in SAS for situations in which the mean or average will not sufficiently describe a sample of individual scores.

Longitudinal and repeated measures data are rich in information about individuals and how they change across time. Plotting average trends can obscure important details about the data. It ignores interpersonal variation, leaving you with an incomplete understanding of the data. SAS procedures like SGPLOT and SGPANEL are flexible tools that offer more opportunities to observe individual and group differences in your data.

Similarly, statistical methods that only provide information about the "typical" person in a population can fail to capture important information about how variables are related for the individuals in the population. This is true because the average curve can theoretically fail to describe even a single individual in the sample! Linear mixed effects models and generalized linear mixed effects models go beyond the mean. In this paper, I gave examples of random intercept models that provide information about variation in the level of the outcome across individuals. In SAS, these models are easy to implement using PROC MIXED, PROC NLMIXED, and PROC GLIMMIX.

With tools available to study individual-level data, we can go beyond studying "typical" trends, and begin to understand *real* trends for the *real* individuals that we claim to study.

REFERENCES

Bureau of Labor Statistics, U.S. Department of Labor. National Longitudinal Survey of Youth 1997 cohort, 1997-2013 (rounds 1-16). Produced by the National Opinion Research Center, the University of Chicago and distributed by the Center for Human Resource Research, The Ohio State University. Columbus, OH: 2015.

CONTACT INFORMATION

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

Melissa L McTernan
California State University, Sacramento
(916) 278-5714
mcternan@csus.edu