Paper 196-29

How I Stopped Worrying and Learned to Love Pearsons ρ: Increasing Robustness in Correlation

Eric Rosenberg, Consumers Union, Yonkers, NY

ABSTRACT

Consumer Reports magazine tests the widest range of products of any consumer product testing organization on the planet. As the number of products increases, keeping data collection and analysis costs low without losing data quality is essential to fulfilling our mission. One way to do this is to eliminate redundant test or survey procedures. Though simple methods of determining redundancy such as correlation are excellent for convincing the non-mathematically inclined, they suffer from a lack of robustness. By repurposing some regression diagnostics tools from PROC REG, such as Leverage, Cooks Distance statistic, and Dffits, we can determine outliers and influential points to increase the robustness of our correlation results. Furthermore, visualizing the data using a SAS graphs visibly demonstrate the impact of removing questionable data points. Data re-sampling techniques, such as the jackknife, also can be used to determine the true value of a correlation coefficient, and are an alternative to removing data.

Data collection costs money. Any chance to reduce testing without degrading data quality is always welcome, whether it be generating a survey with fewer questions or performing fewer mechanical tests on a production item. Of course, fewer tests also mean less time spent analyzing the data and interpreting results. The paper below details a few methods for determining the existence of redundant tests using tools from regression analysis, Pearson's correlation statistic **(ρ),** and data re-sampling techniques.

Consumers Union, Publisher of Consumer Reports Magazine, is arguably the largest tester of consumer products on the planet. The range of products we test is prodigious -- from cars to pregnancy tests, wine and power tools. Unlike most manufacturer product testing however, which usually concerns either product improvement or Quality and Assurance (quality control) testing, Consumers union seeks to rank the products on the more nebulous **overall quality**. The **overall quality** score often is the result of several different tests and judgments, each requiring its own experimental design, statistical analysis, panelist/technician training and of course, testing.

Recently Consumers Union tested 25 circular saws. Though standard industry tests include power ratings (watts) and dynamometer tests (to determine torque), the product testing staff has long felt that performance on these two tests might not adequately represent the true ability of the saws to cut wood. For example, the quality of saw blade could also have an impact. Several additional tests were therefore devised:

1. Pine boards (standardized for density) were *cross cut* repeatedly by an experienced saw technician.
2. Pine boards (standardized for density) were *rip cut* repeatedly by an experienced saw technician.
3. Saws were attached to a weighted pulley system that pulled the saws at a constant force through medium density hardboard.

This extensive testing methodology, though thorough, was extremely time and labor intensive. Each test took almost an entire day just to set up. This was unacceptable from both a time cost perspective, as well as making it prohibitively difficult to test additional models as they appeared throughout the year, a necessary step to ensure that CU's ratings remained current.

All data collected at Consumers Union is double-checked before it even reaches the statistician to eliminate PBKAC errors (Problem between Keyboard and Chair), and we are confident that the collected data is an accurate and representative sample of the 'true' value. Furthermore, we believe that our product selection methodology represents an accurate and sufficient representation of the universe of products available to the consumer.

The statistics department felt that there were probably substantial time savings to be had if redundant tests could be eliminated. Though reducing replications was also considered, while it may have been statistically viable, would not result in much time savings because it does not take much more time to cut several boards rather than one. However, eliminating several of the tests would result in substantial savings. But could we reduce the testing methodology and still maintain the accuracy of our ratings? Not just the statistician had to be convinced -- the testing team included mechanical engineers who were much less sophisticated in their understanding of statistics.

In the majority of organizations, projects requiring statistical analysis (and thereby SAS code) rarely involve only

statisticians and programmers – most often there are non-technical staff involved as well. Since the non-technical staff often have a significant voice in the final deliverables, they must be made to understand the reasons made behind decisions made in the data collection and analysis stage of the projects. Simpler statistical measures can be a good means of including the entire project team in statistical decisions. Fortunately, even the most statistics-phobic engineer or marketing executive is usually comfortable with the concept of correlation. If two tests can be shown to be overwhelmingly correlated, then it should be easy to convince team members to eliminate redundant tests.

Unfortunately, correlation is not a robust procedure, as may be seen in the example below. Notice, how single observation A changes the correlation coefficient from 0.0 to 0.94!

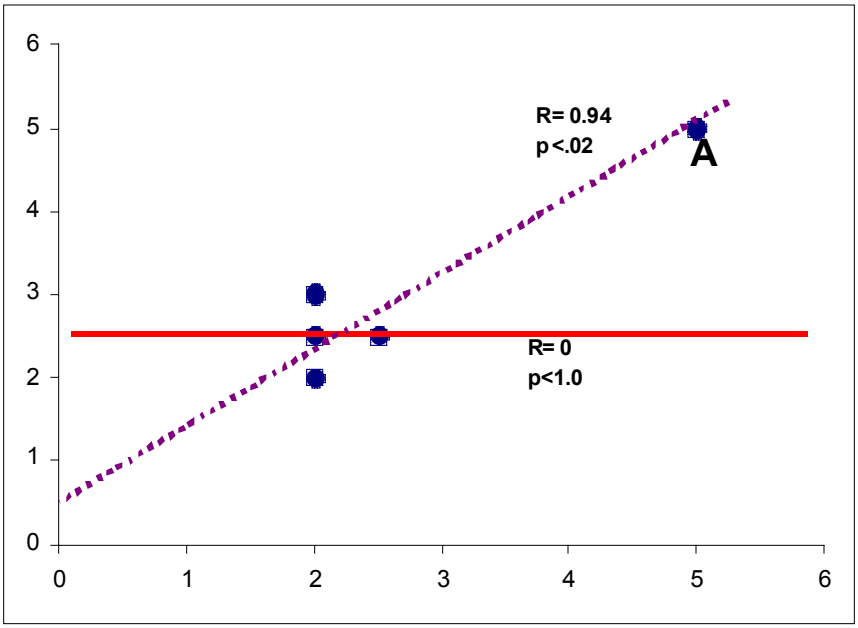


Figure 1

Though there has not been much statistical research towards improving the robustness of Pearson's correlation coefficient, there have been many methods developed for outlier detection for linear regression modeling. Luckily, in the single independent variable case, Pearson's correlation coefficient is calculated in the same method as the square root of r^2 in linear regression. This suggested that the tools used for outlier detection in Linear Regression, and in PROC REG, could be easily modified to determine the robustness of our correlation results. The three most applicable outlier detection methods available in PROC REG are *leverage, Cooks distance statistic,* and *dffits.*

In a regression model, the fraction of the total variation in ***Y*** that is accounted for by the association between ***Y*** and ***X*** is the r^2 statistic. In other words, r^2 measures how well a regression line approximates real data points; an r-squared of 1.0 (100%) indicates a perfect fit. In regression, we assume that we know the X-values exactly – only the Y is dependent, so we only seek to minimize differences in the vertical axis, whereas in correlation, both X and Y are considered to be random variables, so that Pearson's correlation statistic is the average minimization in both directions, vertical and horizontal.

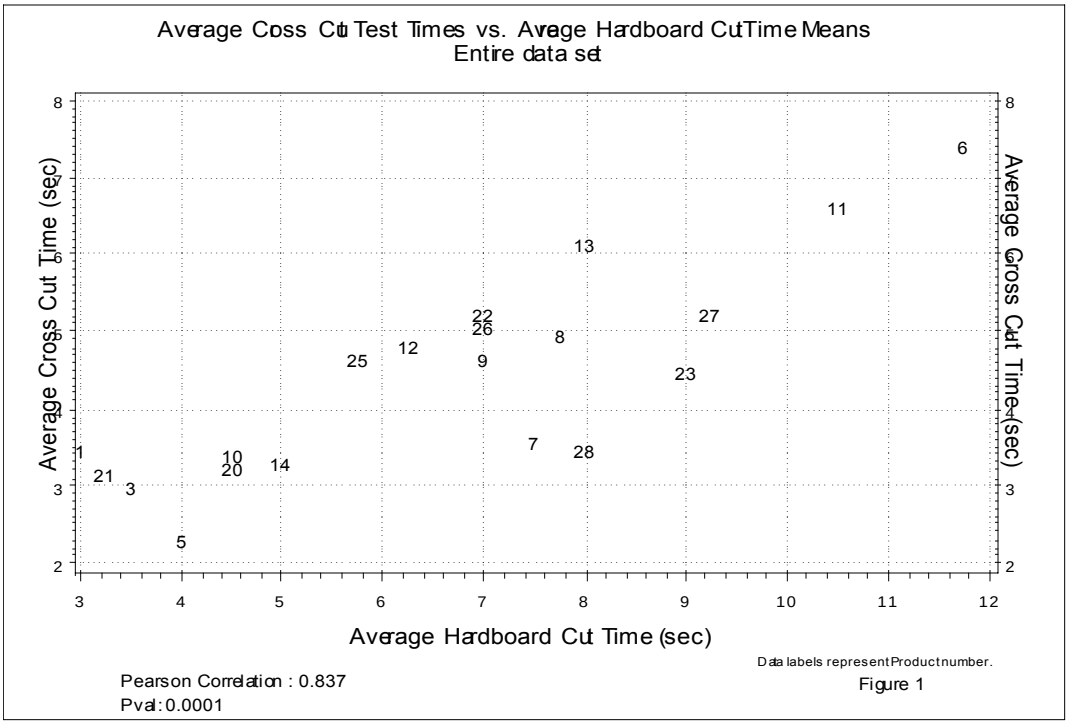
A ***high-leverage point*** is an observation that heavily sways the path of the fitted equation by causing the regression line to pivot towards the point around the centroid of the data, as though the centroid were a fulcrum, and the regression line a lever. This may cause the regression equation to fit the data poorly, or in our case, the correlation coefficient to over(under)state the correlation. Therefore, if a point has high leverage, then removing it can have a substantial effect on the estimates of the correlation coefficient. Generally, the criteria for removing an observation is a leverage score > 2p/n, where p is the number of coefficients (in our case always 2: the intercept + the single ß), and n the number of observations.

DFFITS measures the change in the predicted value of Y for the *i*th observation when the *i*th point is removed from the data set. Values of DFFITS greater than 1 for small data sets or greater than $2(p/n)^{1/2}$ (see above), suggest that the corresponding data point is influential.

Cook's distance measures the change in the sum of squared differences for every *i*th observation when the *i*th point is removed. Usually in regression diagnostics, only values close to 1 are of concern.

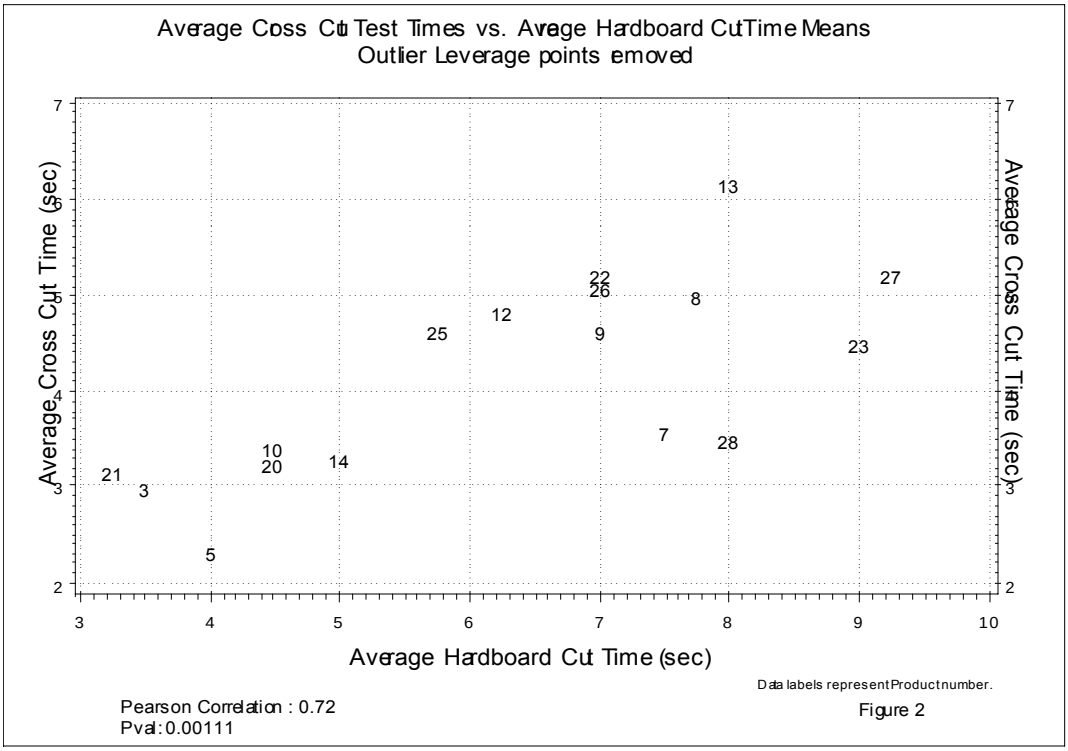
A macro was constructed to generate each of these statistics using PROC REG. The standard criteria for

eliminating outlier data was used as noted above, except in the case of Cooks Distance, where the top 10% of the most influential points were dropped to calculate new correlation coefficients. The 10% figure was chosen to assure that some data was always eliminated. The SAS code in Appendix 2 creates separate data sets for each measure of influence or leverage, deletes potential outliers and then generates scatter plots using the censored data. Below are the SAS program output graphs. Notice how eliminating the leverage points greatly changes the results.



Pearson Correlation : 0.837
Pval: 0.0001

Figure 1



Pearson Correlation : 0.72
Pval: 0.00111

Figure 2

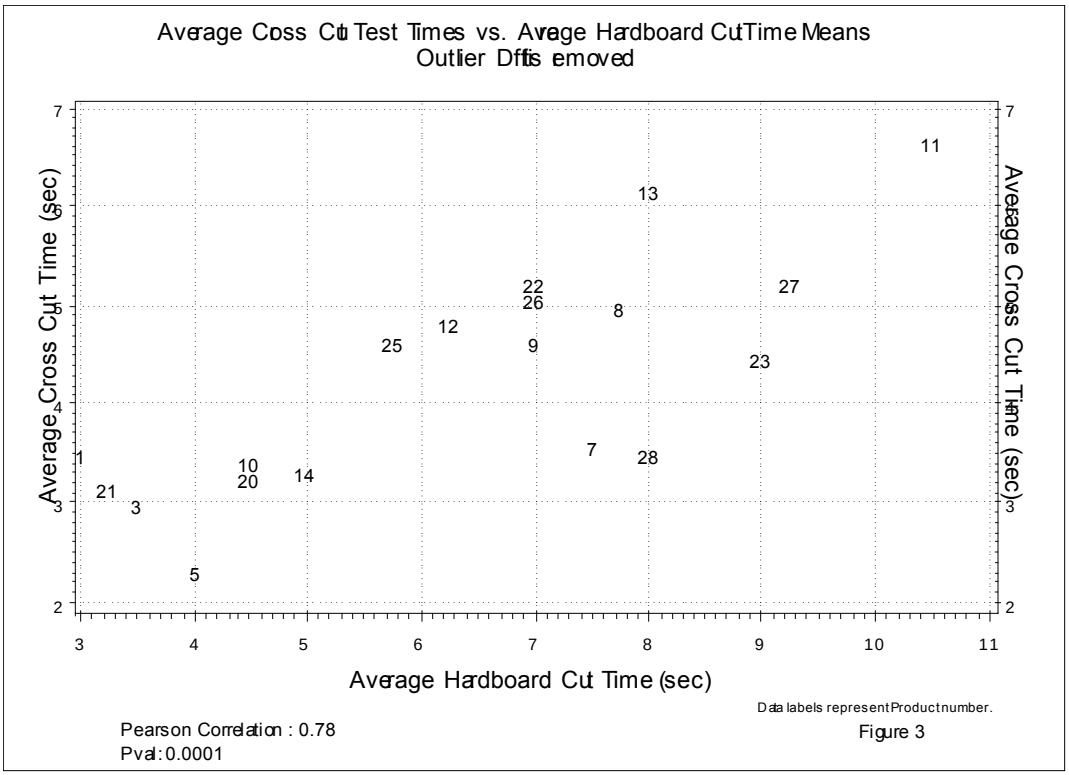


Figure 3

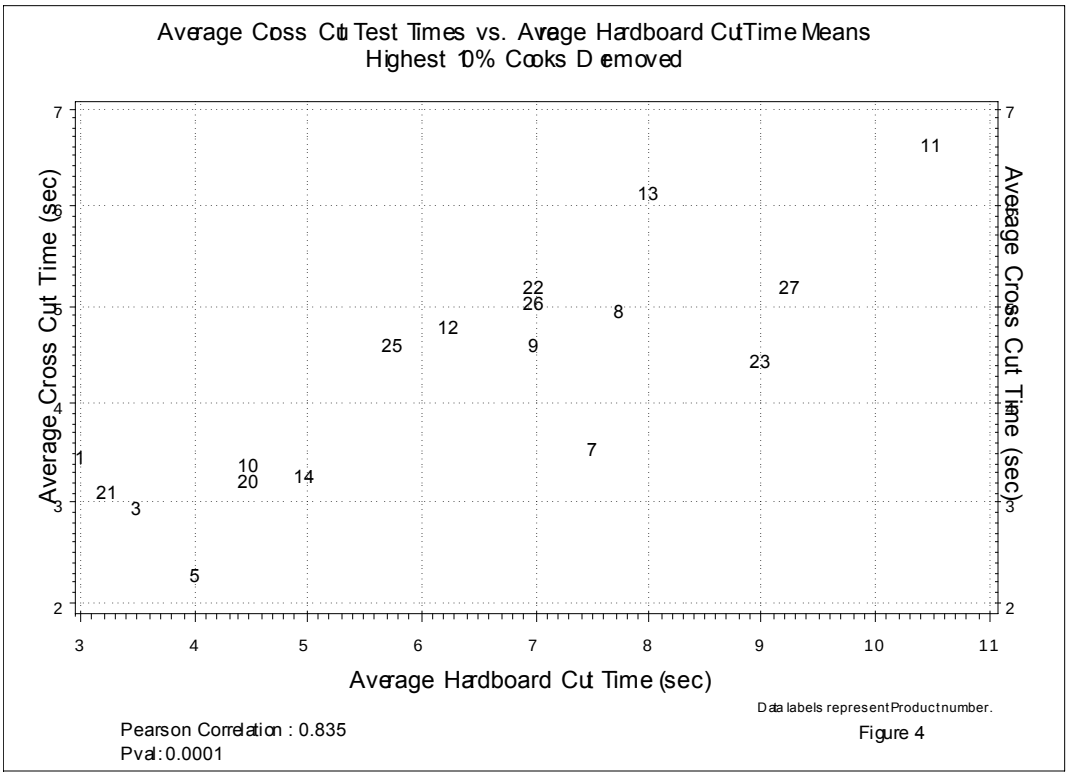


Figure 4

Though the above plots addressed the effects of potential outliers, testers at Consumers Union are not keen on removing observations. We would rather be conservative, and make as few assumptions as possible about the underlying distributions of test data. It is very difficult to determine if a data point is an outlier because of test data, or whether it is a truly exceptional (good or bad) product. Furthermore, dropped observations could mean that we were eliminating the data from a tested model -- and then we could not report on it!

A re-sampling technique such as jackknife or bootstrap would help to determine the accuracy of the correlation

without eliminating possible true product values from the data set. Though Jackknife and Bootstrap are often used interchangeably in statistical parlance, they are two different methodologies. In Jackknife, the same test is repeated by leaving one subject out each time. Thus, this technique is also called **leave one out**. In bootstrap, the sample is duplicated many times and treated as a virtual population. Then samples are drawn from this population to construct an empirical sampling distribution.

A modified jackknife procedure was used to generate a distribution of possible correlation coefficients. Instead of only removing one observation, 10% of the data was eliminated each time (or 90% re-sampling) to ensure the removal of more than one model. Jackknife was chosen for being less computationally intensive.

The macro in Appendix 3 lets the user define the number of jackknife simulations she would like to run, and uses a random number generator to remove 10% of the observations. The results are then compiled into a histogram using PROC CHART. The histogram below was generated using 50 iterations.

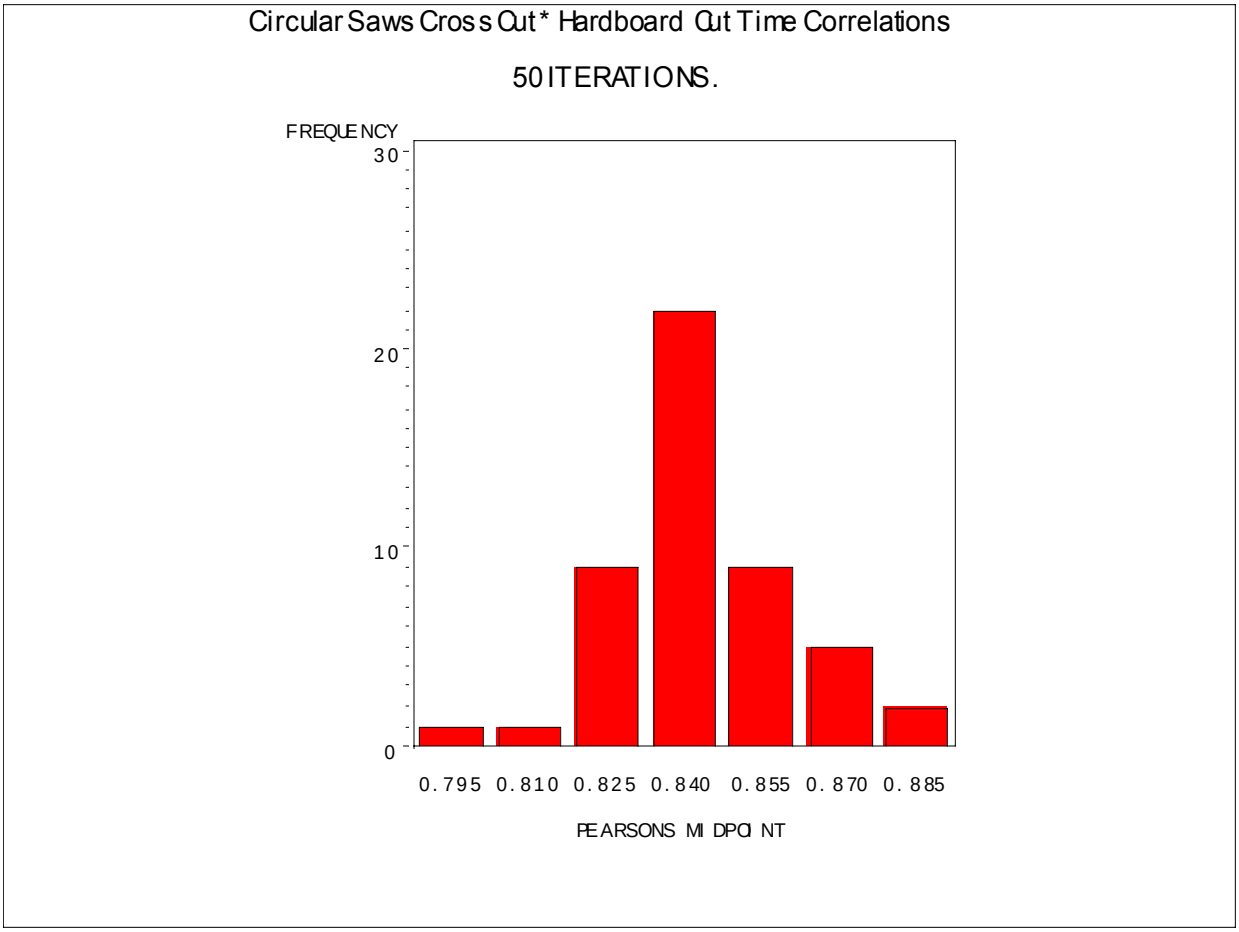


Figure 5

Notice how the distributions of generated correlation statistics have a roughly normal distribution around 0.84, presenting a fairly convincing case for accepting this as the 'true' value. However there is the possibility, albeit small, that the value is 0.8. Though it is possible (though unlikely) that consumers union would consider a test redundant with a correlation of 0.84, the possibility that the true value is 0.8 would strongly sway us against eliminating either the Hardboard or Crosscut test.

The above analysis methodology and macros present the statistician with several types of sensitivity analysis. This is essential when trying to make decisions about population data where the underlying distribution is unknown and therefore identification of 'true' outliers is problematic. Furthermore, by using a simple statistic, such as Pearson's correlation, even non -statisticians can feel comfortable with the eventual conclusions. Though the example cited was within the framework of product testing, the methodologies explored may easily be extended to any circumstance where the statistician/programmer suspects that there may be room to streamline the data collection/ analysis process.

APPENDIX 1: SAS CODE TO GENERATE FIGURE 1

Graph itself was made in Microsoft Excel.

```
data data1;
input x y;
cards;
2 2
2 3
2.5 2.5
2 2.5
5 5
;
run;


data data2;
input x y;
cards;
2 2
2 3
2.5 2.5
2 2.5

;
run;

%macro correg(data);
   proc corr data=&data;
      var x y;
   run;

   proc reg data=&data;
     model y =  x/CLB;
     plot y*x;
   run;
%mend cor_reg;

%correg(data1);
%correg(data2);
```

APPENDIX 2: SAS CODE TO GENERATE FIGURES 2-4

```
/* ********************************************************************** */
/* MACRO CORRPLOTS: GENERATES PRETTY SCATTERPLOTS FOR CORRELATION ANALYSIS  */
/* Variable Descriptions                                                  */
/* mtitle - primary title                                                 */
/* mtitle2 - secondary title                                              */
/* fignum - figure number                                                 */
/* dataset - dataset name                                                 */
/* vary -variable on y-axis on graph                                     */
/* varx - variable for x-axis on graph                                    */
/* yleg - legend for y-axis on graph                                      */
/* xleg - legend for independent variable (4 word max)                    */
/* labelvar - variable which labels the observations in the scatter graph */
/* label - legend for the labeling variable                               */
/* ********************************************************************** */

%annomac;

%macro corrplots(mtitle, mtitle2, fignum,dataset,vary,varx, yleg, xleg, labelvar, label );
goptions border;

AXIS1 /* ORDER= */
      LABEL=(ANGLE=90 font='Arial' h=1.5 "&yleg" );
AXIS3 /* ORDER= */
      LABEL=(ANGLE=0 font='Arial' h=1.5 "&xleg" );
AXIS5 /* ORDER= */
      LABEL=(ANGLE=270 font='Arial' h=1.5 "&yleg" );

data labels;
   length  x y 8.;
   set &dataset (keep= &labelvar &varx &vary);
               if &varx ne . and &vary ne .;
               %system(2,2,3);
               %label(&varx,&vary,left(&labelvar),black,0,0,3,'Arial',5);
run;

proc corr data=&dataset;
        var &vary   &varx;
        ods exclude VarInformation ;
        ods exclude SimpleStats;
        ods exclude PearsonCorr;ods output PearsonCorr=sim;
run;

data sim;
  set sim;
        if variable="&varx";
        call symput("correl",left(round(&vary,.001)));
        if p&vary < .0001 then  call symput("pvals",left(.0001));
        else call symput("pvals",left(round(p&vary,.00001)));
run;

PROC GPLOT DATA=&dataset;
        symbol1 v=none;
        PLOT &vary *  &varx /
            caxis=Black overlay Frame autovref lvref=33 autohref lhref=33
                haxis=axis3
                vaxis=axis1
        annotate=labels ;
        PLOT2 &vary *  &varx  /
        overlay
                haxis=axis3
                vaxis=axis5
                frame;
        title1 h=.5 ' ';
        title2 h=1.5 font='Arial'  "&mtitle";
        title3 h=1.5 font='Arial'  "&mtitle2";
        footnote   h=1.2 j=L font='Arial'   "                    Pearson Correlation : &correl";
```

```
        footnote2  h=1.2 j=L font='Arial'   "                    Pval: &pvals";
        footnote3 h=.3 '  ';
    footnote4 h=1.2 font='Arial' move =(80,1.6)"Figure &fignum";
        footnote5 h=1.0 font='Arial' move =(71,3)"Data labels represent &label.";
RUN;
%mend corrplots;

/*------------------------------------------------------------------------- */
/* MACRO ROBUST: GENERATES SCATTERPLOTS WITH CORRELATION STATISTICS AFTER   */
/* REMOVING POSSIBLE OUTLIERS                                                */
/* Variable Descriptions:                                                   */
/* mtitle - primary title                                                   */
/* data - dataset name                                                      */
/* yvar  - variable on y-axis on graph                                      */
/* xvar  - variable for x-axis on graph                                     */
/* yleg  - legend for y-axis on graph                                       */
/* xleg  - legend for independent variable (4 word max)                     */
/* labelvar - variable which labels the observations in the scatter graph   */
/* label - legend for the labeling variable                                 */
/* -------------------------------------------------------------------------*/

%macro robust(data,yvar,xvar,mtitle, yleg, xleg,labelvar, label );

/* Counts the number of obervations in data set. */
data temp;
  set &data;
    i=_n_;
    call symput('num',_n_);
    call symput('num10',int(_n_/10));
run;

proc reg data=&data;
  model &yvar =  &xvar/influence;
  output out=&data._out
   COOKD=cookd
   DFFITS=dffits
   H=Leverage ;
run ;
%corrplots(&mtitle, Entire data set, 1,&data,&yvar,&xvar, &yleg, &xleg,&labelvar, &label  );

/* LEVERAGE POINTS */
DATA LEVERAGE;
 SET &data._out;
 if 4/&num >= leverage;
RUN;
%corrplots(&mtitle, Outlier Leverage points removed, 2,leverage,&yvar,&xvar, &yleg, &xleg,&labelvar,
&label  );

/* DFFITS */
DATA DFFITS;
 SET &data._out;
 if 2*SQRT(2/&num) >= DFFITS ;
RUN;
%corrplots(&mtitle, Outlier Dffits removed, 3,dffits,&yvar,&xvar, &yleg, &xleg,&labelvar, &label );

/* COOKS D */
proc sort
  data= &data._out;
     BY descending cookd;
run;
data cooks;
set &data._out;
DELETME=_N_;
if DELETME > &num10;
run;
%corrplots(&mtitle, Highest 10% Cooks D removed,4,cooks,&yvar,&xvar,&yleg,&xleg,&labelvar, &label);
%mend robust;
```

APPENDIX 3: SAS CODE TO GENERATE FIGURE 5

```
/*------------------------------------------------------------------------- */
/* MACRO REMOVE10: GENERATES SCATTERPLOTS WITH CORRELATION STATISTICS AFTER */
/* REMOVING POSSIBLE OUTLIERS                                               */
/* Variable Descriptions:                                                   */
/* MTITLE - primary title                                                   */
/* data - dataset name                                                      */
/* BOOT - NUMBER OF JACKNIFES                                               */
/* yvar - variable on y-axis on graph                                      */
/* xvar - variable for x-axis on graph                                      */
/* ------------------------------------------------------------------------ */


%macro remove10(MTITLE ,boot,data,yvar,xvar);
 PROC DATASETS;
  DELETE CORRSTATS;
 RUN;


TITLE1 h=1.5 font='Arial' "&MTITLE" ;
Title3 h=1.5 font='Arial' "&boot ITERATIONS.";

data temp;
set &data;
    i= n ;
    call symput('num10',int(_n_/10));
run;

%DO A = 1 %TO &boot;

  DATA &data._&A;
  SET &data;
    MIXX = RANUNI(0);
  run;

  PROC SORT data=&data._&A;;
   BY MIXX;
  run;

  data &data. &A;
  set &data. &A;
     DELETME= N ;
    if DELETME > &num10;
  run;

  proc corr data=&data. &A;
       var &Yvar   &Xvar;
       ods exclude VarInformation ;
       ods exclude SimpleStats;
       ods exclude PearsonCorr;ods output PearsonCorr=sim;
run;

data sim;
  set sim;
       if variable="&Xvar";
       PEARSONS =(round(&Yvar,.001));
       if p&Yvar < .0001 then  PVAL=(.0001);
       else PVAL=(round(p&Yvar,.00001));
run;

data corrstats;
set CORRSTATS SIM;
RUN;

%END;


PROC GCHART DATA=CORRSTATS;
VBAR PEARSONS ;
RUN;
%MEND REMOVE10;
```

REFERENCES

http://seamonkey.ed.asu.edu/~alex/teaching/WBI/resampling.html

http://www.basic.nwu.edu/statguidefiles/mulreg_ass_viol.html

ACKNOWLEDGMENTS

Thanks to M. Romm and H. Eckholdt for pilfered code.

CONTACT INFORMATION

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

Eric Rosenberg
Consumers Union
101 Truman Avenue
Yonkers New York 10703-1057
914 378 2305
914 378 2908
erosenberg@consumer.org