J. Dairy Sci. 97:5645–5656
http://dx.doi.org/10.3168/jds.2013-7724

SAS INPUT FILE

SASv 9.3 code procedure for application of removal of outliers within feed populations.

```
/* Remove nutrient observations that are zero and for minor nutrients   */
/* set nutrient concentrations to missing.                              */

DATA a01;
  SET hay;
  IF dm = 0 THEN DELETE;
  IF cp = 0 THEN DELETE;
  IF ndf = 0 THEN DELETE;
  IF ash = 0 THEN DELETE;
  IF lig = 0 THEN DELETE;
  IF adfpro = 0 THEN adfpro = '.';
  IF ndfpro = 0 THEN ndfpro = '.';
  IF solpro = 0 THEN solpro = '.';
  IF starch = 0 THEN starch = '.';
  IF fat = 0 THEN fat = '.';
  IF sugar = 0 THEN sugar = '.';
RUN;
PROC UNIVARIATE DATA=a01;
  VAR dm cp ndf ash lig adfpro ndfpro solpro starch fat sugar;
RUN;

/* STEP 1: Remove nutrient concentrations 3.5 STD units from the mean.  */
/* ---------------------------------------------------------------------*/
/* Values in this step are typed in the code from the output of         */
/* PROC UNIVARIATE.                                                     */
/* This step could be programmed from the output to increase automation. */
/* This wasn't done here to improve the clarity of the code.            */

DATA a01_a;
  SET a01;
  IF dm < 74.89 or dm > 100 THEN DELETE;
  IF cp < 9.69 or cp > 31.08 THEN DELETE;
  IF ndf < 16.72 or ndf > 67.40 THEN DELETE;
  IF ash < 2.88 or ash > 16.86 THEN DELETE;
  IF lig < 2.78 or lig > 11.94 THEN DELETE;
RUN;
DATA a01_b;
  SET a01_a;
  zDM = dm;
  zCP = cp;
  zNDF = ndf;
  zAsh = ash;
  zLig = lig;
RUN;
/* Standardize all nutrient concentrations                              */
PROC STANDARD DATA=a01_b MEAN=0 STD=1 OUT=stand_a01;
  VAR zDM zCP zNDF zASH zLig;
RUN;
/* Prin. component analysis using the cov option                        */
PROC PRINCOMP DATA=stand_a01 COV OUT=a01_princomp;
  VAR zDM zCP zNDF zASH zLig;
```

J. Dairy Sci. 97:5645–5656
http://dx.doi.org/10.3168/jds.2013-7724


```
RUN;
PROC UNIVARIATE DATA=a01_princomp;
  VAR Prin1 Prin2 Prin3 Prin4 Prin5;
RUN;

/* STEP 2: Remove PCA scores that are 3.5 STD from the mean.              */
/* ----------------------------------------------------------------------*/
/* Values in this step are typed in the code from the output of           */
/* PROC UNIVARIATE.                                                       */
/* This step could be programmed from the output to increase automation. */
/* This wasn't done here to improve the clarity of the code.              */

DATA a01_princomp1;
  SET a01_princomp;
  IF Prin1 < -5.79 or Prin1 > 5.79 THEN DELETE;
  IF Prin2 < -3.56 or Prin2 > 3.56 THEN DELETE;
  IF Prin3 < -2.64 or Prin3 > 2.64 THEN DELETE;
  IF Prin4 < -2.41 or Prin4 > 2.41 THEN DELETE;
  IF Prin5 < -1.52 or Prin5 > 1.52 THEN DELETE;
RUN;
/* Two stage cluster analysis, k = n^0.3 */
PROC CLUSTER DATA=stand_a01  OUTTREE=tree1 METHOD=twostage
             k=15 PRINT=40 PSEUDO;
  VAR  zCP zNDF zdm zash zlig;
RUN;
PROC GPLOT DATA=tree1;
/* Repeat gplot of tree1 two times, once for PSF and once for PST2.       */
  PLOT _PSF_ *_NCL_;
RUN;
/* Determine number of clusters from PSF and PST2 maximum values.         */
PROC TREE DATA=tree1 NOPRINT OUT=out n=6;
  COPY  zCP zNDF zdm zash zlig _dens_ _freq_;
RUN;
PROC SORT DATA=out;
  BY CLUSTER;
RUN;

/* STEP 3: Delete cluster if cluster is less than 10% of the dataset.    */
/* ----------------------------------------------------------------------*/

PROC CORR DATA=out;
  BY cluster;
  VAR dm cp ndf lig ash;
RUN;
DATA out1;
  SET out;
  IF CLUSTER = 6 THEN DELETE;
RUN;
/* Repeat cluster analysis only when a cluster is removed and 2 or more  */
/* clusters remain by following the same clustering steps.                */
/* If further clustering analysis is not needed, then summarize           */
/* statistics of interest.                                                */
```