

Rona Tracy
2702 Mar Vista St
Stillwater, OK 74074
(405) 743-5413 – work
rtracy@okvotech.org

Implementing Robust Statistical Calculations for Minimizing the Effect of Outliers

System performance measures based on traditional mean and standard deviation calculations can be misleading if they are distorted by outliers in the data. Robust statistical estimates of the mean and standard deviation provide a way to lessen the effects of outliers without going through the process of excluding them.

Traditional mean and standard deviation calculations are not considered robust because they have very low resistance to outliers. Since a single outlier can significantly alter the mean and standard deviation, these calculations are said to have low breakdown points. The breakdown point of a statistic is "...the fraction of observations that must be contaminated in order to force the estimate beyond any bound" (Hettmansperger and Sheather, 146).

The **Median Absolute Deviation (MAD)** provides a robust alternative to the traditional standard deviation calculation. A robust iterative estimate of the mean called Huber's m-estimator can replace the traditional mean calculation.

Calculations

The MAD is calculated as follows (Miller, 458):

$$MAD = median[|x_i - median(x_j)|],$$

where $|x_i - median(x_j)|$ = the absolute value of $[x_i - median(x_j)]$

A robust estimate of standard deviation based on the MAD is calculated as follows (Miller, 458):

$$\text{standard deviation} = \hat{\sigma} = MAD / 0.6745$$

For example, the MAD for the number set (3,3,6,8,15) is calculated as follows. The median of the set is 6. The set of absolute deviations from the median is (3,3,0,2,9). The MAD is the median of the deviations, or 3. The MAD has a breakdown point of 50 percent.

Miller's reference article provides details for iteratively calculating Huber's m-estimator for the mean. The iterative process uses the median as an initial estimate of the mean. Each data point is looked at to see if it falls in an acceptable range from the estimate of the mean. The acceptable range is calculated as $1.5 * (MAD/0.6745)$. Any data points falling outside this range around the mean are given less extreme values but are not discarded. After the initial pass through the data set, a new estimate of the mean is calculated using the traditional mean calculation instead of the median. The second iteration begins, and the iterative process continues until all data points fall within an acceptable distance of the mean.

Performance measures based on traditional mean and standard deviation calculations can be distorted by one or more outliers in the data. When removal of outliers is not feasible or practical, robust statistical estimates such as the MAD and Huber's m-estimator can be used to provide a realistic picture of system performance.

SAS Code

The SAS code below calculates the robust standard deviation and mean. While not terribly difficult, the code for calculation of the robust mean is a bit tricky due to the use of iteration.

```
* ROBSTAT.SAS *;

* ----- Calculate Robust Standard Deviation -----;
*** calculate the median ***;
proc univariate data=one noprint;
  var recov;
  by site agent level station;
  output out=outmed n=nfirst nmiss=nummiss max=maxrecov min=minrecov
    median=median std=stdev mean=mean;
run;

*** determine a robust estimator of scale to replace the ***;
*** traditional standard deviation. ***;

*** first, determine the absolute deviations from the sample median
***;
data devmed;
  merge one outmed(keep=site agent level station median nfirst stdev
mean);
  by site agent level station;
  deviate = abs(recov-median);
run;
proc univariate data=devmed noprint;
  by site agent level station;
  var deviate;
  output out=devmed2 median=scale;

proc sort data=devmed2;
  by site agent level station;

** create data set with median and location parameter by station ***;
data standard(keep=site agent level station rob_sd);
  merge outmed(keep=site agent level station median nfirst stdev mean)
    devmed2;
  by site agent level station;
  *** add correction factor to make MAD analogous to standard deviation
  ***;
  rob_sd = scale/0.6745; *** 1.483 x scale;
run;

data find1;
  merge one(keep=site agent level station recov)
    standard;
  by site agent level station;
run;
*** create single records w/all recovery for a station ***;
proc transpose data=find1 out=tranray;
  by site agent level station;
```

```

data tranray2; set tranray; *** create file containing only % recov
***;
  if _name_='RECOV';
run;

*proc sort data = outmed;
* by site agent level station;
*run;
data allfind;
  merge tranray2
    standard
    outmed(keep=site agent level station median nfirst);
  by site agent level station;
run;

* ----- Calculate the iterative robust mean -----;

data findmean; set allfind;
  BY SITE AGENT LEVEL STATION;
  ARRAY XNEW{500} ; * defines variables xnew1-xnew100 *;
                      *** new estimates of values for iterative
process;
  ARRAY DIFF{500} ; *** abs value of recovery minus estimate of mean
***;
  ARRAY XVAL{500} COL1-COL500; *** original % recov values ***;
  *** iterative process to estimate mean recovery value based on median
*** ;
  * NOTE:  rob_sd = MAD/0.6745;
  sigrange = 1.5 * rob_sd;
  SIGRANGE = ROUND(SIGRANGE,.01);
  COUNTIT=0;
  trackdif=0;
  *** initial guess for mean is the median;
  GUESS = median;
  nrecs = nfirst;

  do until(TRACKDIF>=nrecs);
    trackdif=0;

    do L=1 to nrecs;
      DIFF{L} = ABS(XVAL{L}-GUESS);
      DIFF{L} = ROUND(DIFF{L},.01) ;
      IF DIFF{L} > SIGRANGE THEN do;
        TRACKDIF=0 ;
        IF XVAL{L} > GUESS THEN XNEW{L}=GUESS+SIGRANGE;
        IF XVAL{L} < GUESS THEN XNEW{L}=GUESS-SIGRANGE;
        *
        put 'diff is > sigrange, NEW X is ' XNEW{L};
      end;
      *** if diff not more than 1.5 sigma, then keep old numbers
***;
      IF DIFF{L} <= SIGRANGE THEN do;
        XNEW{L}=XVAL{L};
        trackdif=trackdif+1 ;
      end;

    end;
    *** do L = 1 to nrecs;

```

```

*** FIND NEW ESTIMATE OF MEAN *** ;
TOTAL=0 ;
do k = 1 to nrecs;
    TOTAL = TOTAL+XNEW{k};
end;
NEWGUESS = TOTAL/nrecs;
GUESS=NEWGUESS;
*** note the last value of guess will be the mean estimate *** ;

*** USE NEW X-VALUES FOR NEXT ITERATION *** ;
do R = 1 to nrecs;
    XVAL{R} = XNEW{R};
end;

countit= COUNTIT+1 ; *** count the number of iterations ***;
end; *** while trackdif <= nrecs, continue ***;

robmean=guess; * do not round prior to calculations *;
* put "agent and station are " agent station;
* put ' number of iterations is ' countit;
* put "final iterative mean is " robmean;
RUN; *** end of data step to find iterative mean;

```

References

1. Hettmansperger, T.P., and S.J. Sheather. 1992. *Perspectives in Contemporary Statistics*. Washington, D.C.: Mathematical Association of America.
2. Hoaglin, D. C., and B. Iglewicz. 1993. *How to Detect and Handle Outliers*. [The ASQC Basic References in Quality Control: Statistical Techniques, Vol. 16]. Milwaukee, WI: American Society for Quality Control.
3. Miller, J. N. 1993. "Tutorial Review. Outliers in Experimental Data and Their Treatment", *Analyst* 118: 445-461.
4. Potvin, C., and D.A. Roff. 1993. "Distribution-Free and Robust Statistical Methods: Viable Alternatives to Parametric Statistics?" *Ecology* 74(6): 1617-1628.