Paper SA11

Cleaning Data the Chauvenet Way

Lily Lin, MedFocus, San Mateo, CA
Paul D Sherman, Independent Consultant, San Jose, CA

ABSTRACT

Throwing away data is a touchy subject. Keep the maverick and you contaminate a general trend. Toss away good data points and you don't know if something important has happened -- until its too late. How do you know what, if any, data to exclude? Chauvenet is the answer. Here is a really gentle test you can apply to any distribution of numbers. It works equally well for normal, skewed, and even multi-modal populations. This article gives you a macro tool for cleaning up your data and separating the good from the bad.
Skill Level: Basic statistics, Data step, SAS/MACRO, and Proc SQL

INTRODUCTION

You often want to compare data sets. You can't really do this point-by-point, so you summarize each set individually and compare their descriptive statistics on the aggregate. By looking only at the summary, you are making an assumption that all observations are related in some way.

How do you verify this assumption? By testing for outliers. Values that are *spurious* or unrelated to the others must be excluded from summarization.

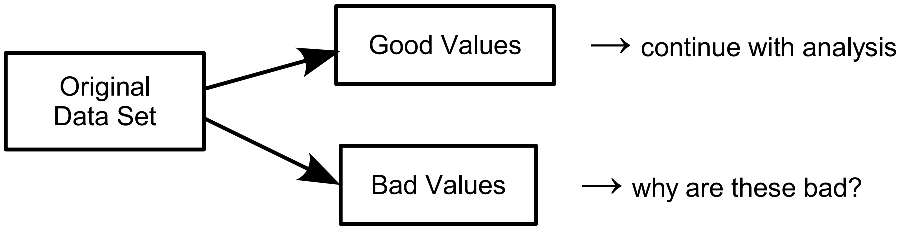
In this paper we present a simple, efficient, and gentle macro for filtering a data set. There are many techniques, and the subject of robust statistice is modern and rich though not at all simple. Chauvenet's criteria is easy to understand, can be quickly computed for a billion rows of data, and even gentle enough to be used on a tiny set of only ten points.

We have seen that Chauvenet's criterion is used in astronomy, nuclear technology, geology, epidemiology, molecular biology, radiology and many fields of physical science. It is widely used by government laboratories, industries, and universities.

Although Chauvenet's criterion is not currently used in Clinical trials, we would like to explore it for possibility of being applied to the trials environment as well.

THE FILTERING PROCESS

You want some way to identify what observations in your data set need closer study. It's not appropriate to simply throw away or delete an observation; you must keep it around to look at later. The picture is as follows.



Our macro does this easily. To scrutinize variable `x` and split `origdata` into good and bad pieces, use this macro call:

```
   %chauv(origdata, x, good=theGood, bad=theBad);
```

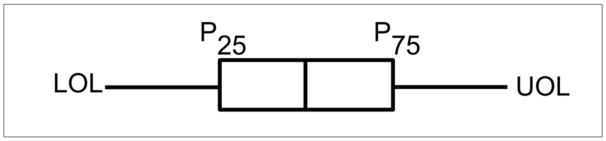
Then, for example,

```
   proc means data=theGood; run;   * summarize the good ... *;
   proc print data=theBad; run;   * ... and show us the bad *;
```

INTERQUARTILE RANGE (IQR) TEST

The IQR test is commonly used in clinical studies to reject data points. The one-way analysis of variance using "box plots" also uses the IQR test. Whiskers of the box are called Outlier Limits and set 50% further away than the IQR. The range for a data point to be considered "good" is defined below.

$$LOL = P_{25} - 1.5 \cdot IQR$$
$$UOL = P_{75} + 1.5 \cdot IQR$$
where $IQR = P_{75} - P_{25}$



geometry of the box plot

An acceptable data value must lie within these limits: $2.5 \cdot P_{25} - 1.5 \cdot P_{75} < x < 2.5 \cdot P_{75} - 1.5 \cdot P_{25}$

IQR is based on percentile statistics. Just like the median, percentiles presume specific ordering of observations in a dataset. The sort step required can be costly, especially when the dataset is huge.
For example, a data set with a billion observations takes half an hour to calculate the percentiles, even with the piecewise-parabolic algorithm, while only six minutes to generate distribution moments μ and σ.

Number of	**Computation Time**	
Observations	**Median, P_{25}, P_{75}**	**Mean, Std**
1,000,000,000	32:31.43	6:07.45
100,000,000	3:13.79	30.15
10,000,000	18.62	1.35
1,000,000	1.89	0.14
100,000	0.20	0.01
10,000	0.03	0.00
1,000	0.01	0.00
100	0.00	0.01

Sorting and observation numbering cannot be done in SQL because rows of a table are independent among themselves. You cannot compute the median in a database query. Neither can you use the ORDER BY clause in a sub-query. Therefore, we must seek an alternative outlier test which can be performed within a database query environment.

WHO IS CHAUVENET

A character in a play? The auntie of someone who's best friend is a pooka. Only Mrs. Chauvenet knows the truth about Elwood P. Dowd. But this observation is itself an outlier.
French mathematician William Chauvenet, 1820-1870, is best known for his clear and simple writing style and pioneering contributions to the U.S. Naval Academy. He mathematically verified the first bridge spanning the Mississippi River, and was the second chancellor of Washington University in St. Louis. To his honor, each year since 1925 a well-written mathematical article receives the Chauvenet award.

CHAUVENET'S CRITERIA

If the expected number of measurements at least as bad as the suspect measurement is less than 1/2, then the suspect measurement should be rejected.

PROCEDURE

Let's assume you have a data set with numeric variable `x`. Suppose there are `n` observations in your dataset. You want to throw away all observations which are "not good enough". How do you do this? Remember that in clinical practice, no point is not good enough, so the subject of outliers does not apply.

1. Calculate μ and σ
2. If $n \cdot \mathrm{erfc}(| x_i - \mu | / \sigma) < ½$ then Reject x_i
3. Repeat steps 1 and 2
   until step 2 passes or too many points removed
4. Report final μ, σ, and n

When the dust settles, you have two data sets: The set of all good data points, and the set of "bad" points. Although most often we don't care about bad data points, sometimes the bad points tell us much more information than do the good points. We must not be too hasty to completely forget about the bad data points, but keep them aside for later and careful evaluation.

Some questions to ask about the "bad" data points, which we will talk about later: 1. Why were these particular points excluded? 2. What do all the bad points have in common?

EXAMPLE

Suppose we have 14 measurements of some parameter, shown below in Table 1. It takes two iterations of the Chauvenet procedure to eliminate all the "bad" data values. The first pass marks 2 values as bad: 29.87 and 25.71. Then, the second pass marks another value bad: 20.46. Further and subsequent applications of Chauvenet don't mark any more points.

As "bad" data points are excluded, notice how the standard deviation significantly improves from 8.77 to 5.17 to 3.38.

	Original	***Pass #1***	***Pass #2***
	8.02	.	.
	8.16	.	.
	3.97	.	.
	8.64	.	.
	0.84	.	.
	4.46	.	.
	0.81	.	.
	7.74	.	.
	8.78	.	.
	9.26	.	.
	20.46	.	**outlier**
	29.87	**outlier**	.
	10.38	.	.
	25.71	**outlier**	.
avg:	10.51	7.63	6.46
stdev:	8.77	5.17	3.38
n:	14	12	11

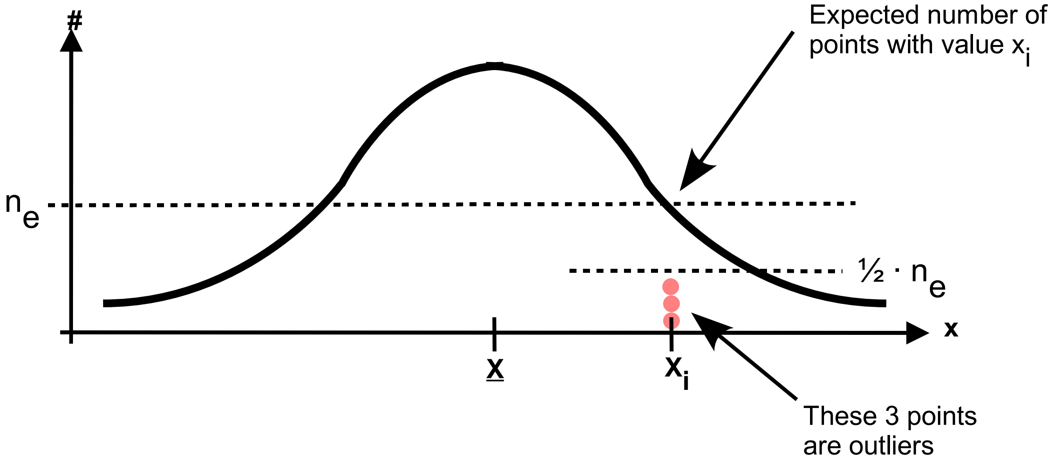
Shielded outlier

The third outlier value, caught and excluded in Pass #2, is called a *shielded outlier*. At first, its value is small enough -- or close enough to the mean -- to be considered good. Only when the most extreme value is removed does this next most largest value become noticeable.

As we will see later, each removal of a data point "lightens the mass" of a distribution. Smaller sample sizes require their values to be closer together. The shielding effect produced by very large values is precisely why we must perform an outlier test *iteratively*.

HOW IT WORKS

If there are fewer points than you expect, then throw away those few points.



Expected number of points with value x_i

These 3 points are outliers

WHAT IS ERFC?

The complementary error function, *erfc*, is the residual area under the tails of a distribution. Its value gets smaller for values further away from the center of the distribution. Thus, the error function value of infinity is always zero. That's like saying there is nothing else to see when you look at the whole picture.

There is nothing specific to any particular distribution. *erfc* simply is the integral of a probability density function. In most database systems and statistics programs, *erfc* assumes the normal or Gaussian distribution. Using the appropriate calculation or look-up table makes Chauvenet's criterion a universal test.

WHY THE ½?

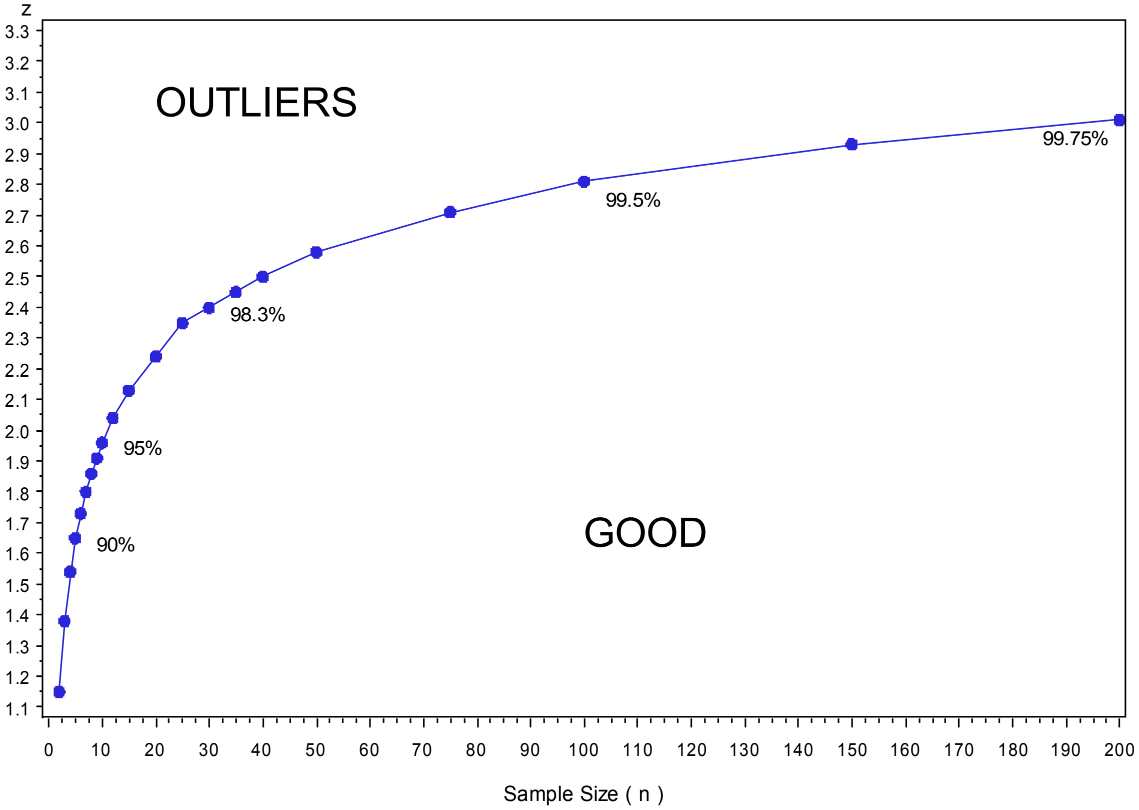
This is the magical Chauvenet number. We give each value a 50% chance of survival. Said another way, there must be as many points *closer* to the mean as there are further away. A value is an outlier if it is so far away that there's hardly any other values greater than it.

Sample size is very important. A distribution with more data points is less likely to be affected by any single point, regardless of its value. Think about sample size as the "mass" of a physical system. It is difficult for a cat to get a bowling ball moving. But a cat can easily play with a ping pong ball all day. Greater mass means more inertia.

This analogy is exactly the same for distributions. Greater numbers of data points means that there is little chance for any single data point to affect the distribution shape. A value must be *very far away* from the mean in order to "move" the distribution of other points and be considered an outlier. With 200 data points, an outlying value is more than 3σ distant -- very far away from the mean!

On the other hand, suppose we have a nearly "mass-less", lightweight distribution with only 10 values. A bad value or outlier need only be 1.96·σ away from the mean. Therefore, smaller sample sizes place more rigid requirements on the individual values.

The critical threshold which separates good values from bad is shown in the figure below. Z is the usual z-score, |x-μ|/σ, indicating how far away a value is from the mean. Percentages show how confident we are that a particular value belongs to the distribution. This plot assumes a normal, gaussian distribution, although the basic concept here is universal and other distributions may be similarly considered by tabulating the appropriate integral.



This figure is simply the z-score corresponding to the confidence level of (1 - 1/(2N)). The "2" in this formula is the magical Chauvenet number. Two example calculations make this picture clear.

$\boxed{N=5}$ 1 / (2N) = 1 / (2·5) = 1/10 = 10/100 = 0.10 1-p = 0.10
p = 0.90 ⟶ table look-up ⟶ z = 1.65

$\boxed{N=10}$ 1 / (2N) = 1 / (2·10) = 1/20 = 5/100 = 0.05 1-p = 0.05
p = 0.95 ⟶ table look-up ⟶ z = 1.96

WHAT HAPPENS IF WE USE 1/3 OR 5/6 OR ...?

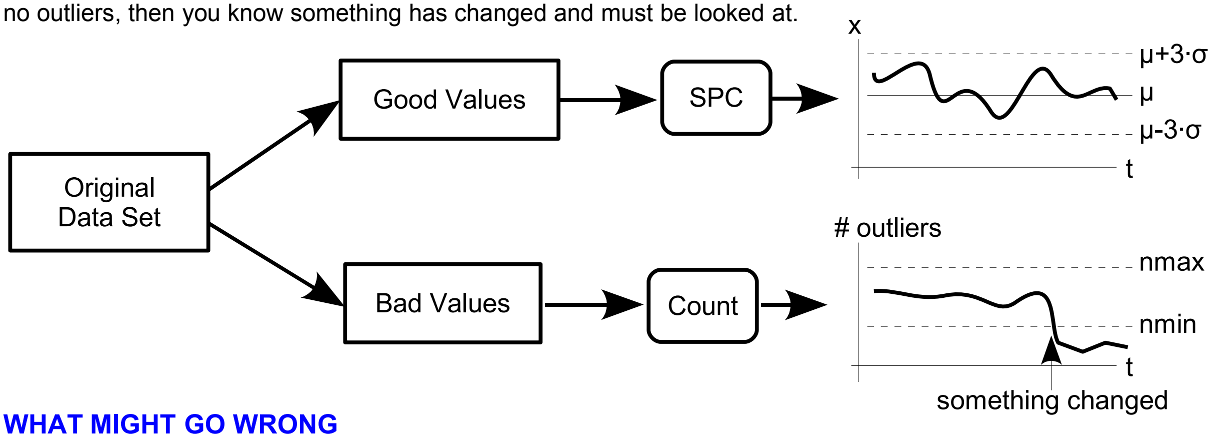
This changes the sensitivity of the outlier test, and corresponds to the nature of the distribution with which you are testing. Using 1/3 means that there must be *twice as many smaller values* as there are larger values. Similarly, 5/6 means there should be only *one smaller value* for every *five larger* values.
Shown in the able below are a few values for the Chauvenet factor and a qualitative comparison.

Chauvenet factor:	1/4	1/3	1/2	2/3	5/6
Distribution shape:	peaky	skinny	normal	disperse	long-tailed
Rejection sensitivity:	very lenient	loose	moderate	tight	very rigid
outlier if ...	x>2.25·σ	x>2.14·σ	x>1.96·σ	x>1.84·σ	x>1.74·σ (for N=10)
Acceptance criteria:	allows more points closer to mean	...	equal quantity of points close to and far from mean	...	requires more points further from mean

SUGGESTION FOR GOOD PROCESS CONTROL

Monitoring summary statistics from complete detail or raw data sets can lead to many anomalous out-of-control alerts. Too many warnings dilutes the effectiveness of a quality control team. Too few, obviously, is not good either; you don't want to be so blind as to pass all the junk.

Using a gentle outlier filtering method such as Chauvenet's criteria is a good idea. You split your raw data set into two pieces: good and bad. Existing SPC (Statistical Process Control) methods are then performed on the good data set. There is no distribution among the bad observations; they are all useless garbage. However, the *quantity* of them *is* useful. Simply count the number of observations in the bad data set and watch a trend chart of that. A consistent, in-control process should have a similar number of outliers over time. If, for example, one day you come in to work and see your process within SPC control yet with almost no outliers, then you know something has changed and must be looked at.



WHAT MIGHT GO WRONG

It might appear that we assume normality of the data distribution. Our first step is to comp1ute mean and sigma. However, we only consider the z-score, which is the ratio of mean and sigma so their distribution inference is nullified. When the mean or sigma does not exist, such as for Lorentzian distributions found in Nuclear Magnetic Resonance experiments, other criteria must be applied.

Chauvenet's criteria has trouble when the data distribution is strongly bi-modal. When there are widely separated, resolvable modes, all data points will be rejected. That's why we put a "stop limit" in step 3 of the procedure so that the entire data set isn't whacked away.

In practice, a bi-modal or multi-modal distribution of a parametric usually means that we have mixed disparate data sources which should not be mixed at all.

CONCLUSION

We have presented a simple, efficient, and gentle macro to make a cleaner data set. It is easier to interpret a summary of test results when the raw results are clean and all related (i.e., not spurious). The number of points excluded from summarization is an important parameter. Keep a log of which points were excluded. If many events exclude the same measurement, look for a systematic trend such as wrong test position, incorrect precondition, etc. With Chauvenet's criteria you can be sure your analyses are free of invisible rabbits.

AUTHOR CONTACT

Lily Lin
2715 South Norfolk Street, Apt. 103
San Mateo, CA 94403
(973) 978-8292
lily4lin@yahoo.com

Paul D Sherman
335 Elan Village Lane, Apt. 424
San Jose, CA 95134
(408) 383-0471
sherman@idiom.com

`www.idiom.com/~sherman/paul/pubs/chauv`

On-line Demonstration
`http://www.idiom.com/~sherman/paul/pubs/demo/chauvdemo.html`

REFERENCES

Chase, Mary Coyle. 1953. Harvey. UK: Oxford University Press.

Dixon, W. J. (1953). "Processing data for outliers." *Biometrics*, vol.9, pp. 74-89.

Ferguson, T. S. (1961) "On the rejection of outliers." *Proceedings of the 4th Berkeley Symp. On Mathematical Statistics and Probability*, 1. pp. 253-187.

Grubbs, F. (1969). "Procedures for Detecting Outlying Observations in Samples." *Technometrics*, 11, pp.1-21.

Herzog, Erik D. (2003, Jan. 24). "Picturing Our Past." In *Record*, vol. 27, no.17, St. Louis, MO: Washington University. Retrieved July 27, 2007, from http://record.wustl.edu/2003/1-24-03/picturing_our_past.html

Mathematical Association of America. "The Mathematical Association of America's Chauvenet Prize," Retrieved July 27, 2007, from http://www.maa.org/awards/chauvent.html

Ross, Stephen M. (2003). "Peirce's Criterion for the Elimination of Suspect Experimental Data." *J. Engr.* Technology.

Peirce, B. (1852). "Criterion for the rejection of doubtful observations." *Astronomical Journal*, 11 (21), pp. 161-163.

Taylor, John R. 1997. An Introduction to Error Analysis : The Study of Uncertainties in Physical Measurements, second ed. Herndon, VA: University Science Books.

Tietjen, Gary L. and Roger H. Moore. (1972, August). "Some Grubbs-Type Statistics for the Detection of Several Outliers." *Technometrics*, 14 (3), pp.583-597.

ACKNOWLEDGMENTS

The authors would like to thank Grant Luo for creating and discussing the IQR comparison study. Annie Perlin deserves a warm round of applause for her role as the other Chauvenet. We greatly appreciate the generosity of MedFocus for allowing us to work on this article.

TRADEMARK INFORMATION

THE CHAUVENET OUTLIER FILTERING MACRO

```
options nosource nonotes;
/* ==================================================== */
/* CHAUV - Chauvenet's criteria data cleaner            */
/*                                                      */
/* INDAT - input dataset                                */
/* VAR - variable name to process                       */
/* GOOD - output dataset for the GOOD observations      */
/* BAD - ditto, for the OUTLIERS. Dot (.) throws away   */
/* CHAUFAC - sensitivity factor. positive, less than 1. */
/*                                                      */
/* 1.0  2007-04-24  pds/lpl   initial cut               */
/* ==================================================== */

* perform a single step of filtering *;
%macro chauv0(iodat, var, chaufac, macvar, loopnum);

  * (re)compute summary on only the good points *;
  proc means data=&iodat. noprint mean std;
    where isgood=1;
    output out=summ (drop=_type_ _freq_) mean=x std=s n=n;
  run;

  * apply the test *;
  proc sql;
    create table &iodat. as (
      SELECT raw.&var.,
             summ.n*erfc((raw.&var.-summ.x)/summ.s)>&chaufac. AS isgood
      FROM summ
           inner join &iodat. AS raw ON raw.isgood=1
    );
  quit;

  %let &macvar.=TRUE;
  data loopdat&loopnum.;
    set &iodat.;
    if isgood eq 0 then do;
      call symput("&macvar.", 'FALSE');
      output loopdat&loopnum.;
    end;
  run;

%mend chauv0;
```

```
* the main macro *;
%macro chauv(indat, var, good=outdatg, bad=outdatb, chaufac=0.5);
  %local isAllgood loopnum;

  * initialize all data points GOOD *;
  * assumes there is not already a variable called ISGOOD *;
  data chaudat;
    set &indat.;
    isgood=1;
  run;

  * loop forever until all values pass the test *;
  %let isAllgood=FALSE;
  %let loopnum=1;
  %do %until(&isAllgood. eq TRUE);
    %chauv0(chaudat, &var., &chaufac., isAllgood, &loopnum.);
    %let loopnum=%eval(&loopnum.+1);
  %end;

  data &good. (drop=isgood);
    set chaudat;
  run;

  %if &bad. ne . %then %do;
    data &bad. (drop=isgood);
      set %do i=1 %to &loopnum.-1; loopdat&i. %end; ;
    run;
  %end;

  proc datasets lib=work nodetails nolist nowarn;
    delete chaudat summ;
    delete %do i=1 %to &loopnum.-1; loopdat&i. %end; ;
  quit;

%mend chauv;


***************************************
*** CLEANING DATA THE CHAUVENET WAY ***
***         a demonstration         ***
***  by Lily Lin and Paul D Sherman ***
***************************************;
*** fake some data ***;
proc sql noprint;
create table raw (value integer);
insert into raw (value)
  values (8.02)   values (8.16)   values (3.97)   values (8.64)
  values (0.84)   values (4.46)   values (0.81)   values (7.74)
  values (8.78)   values (9.26)   values (20.46)  values (29.87)
  values (10.38)  values (25.71)
;
quit;

%chauv(raw, value, good=theGood, bad=theBad);

proc means data=theGood; run;  * summarize the good ... *;
proc print data=theBad; run;  * ... and show us the bad *;

%chauv(raw, value, good=raw, bad=.);  * overwrite the original data *;
```

EXAMPLE - PERCENTILES AND QUARTILES TAKE A LONG TIME TO COMPUTE

Percentile-based calculations take significantly longer to perform than do distribution moments. This is due to the former involving internal sorting steps. These results appear to be quite general, and invariant of how many CPU's or threads are allocated to a system.

```
options nosource notes;
%macro means(n);
data dum;
do i=0 to &n.; x=ranexp(6789); output; end;
run;

proc means data=dum noprint nonobs p50 p25 p75 qmethod=p2 qntldef=5;
output out=dums (drop=_type_ _freq_) median=p50 p25=p25 p75=p75;
run;

proc means data=dum noprint nonobs mean std;
var x;
output out=dumss (drop=_type_ _freq_) mean=avg std=std;
run;
%mend;
%means(1000000000);
%means(100000000);
%means(10000000);
%means(1000000);
%means(100000);
%means(10000);
%means(1000);
%means(100);
%put *** DONE ***;
```

EXAMPLE - COMPARING IQR TEST AND CHAUVENET'S CRITERIA

The IQR test is a single-pass algorithm. Therefore, its number of outlier values is constant. For small sample sizes of a normal distribution with a few artificially "bad" points thrown in, Chauvenet's criteria somewhat overestimates the number of bad points. On the other hand, you might think that IQR somewhat *underestimates* the number of bad points when the data set is small. Remember that smaller sample sizes place more strict rules on what is an outlying value.

When the data set is large, more than 10,000 observations, the situation is reversed. IQR rejects many more points than does Chauvenet. The cross-over point, where both tests report about the same quantity of outlying values, seems to be about 4,000 observations.

Therefore, with very large data sets Chauvenet's criteria is superior. It rejects only those few really bad values, and without percentile calculations is much more time and memory efficient in its calculations.

Loop	N=200		N=4,000		N=10,000		N=1,000,000	
Number	IQR	Chauv	IQR	Chauv	IQR	Chauv	IQR	Chauv
1	10	10	39	14	84	29	7032	408
2	10	15	39	39	84	49	7032	430
3	10	20	39	42	84	55	7032	431
4	10	22	39	44	84	57	.	.
5	10	23	39	45	.	.	.	.

The code which generates this data is shown below.

```
%let n_obs=10000;
data a;
   group=1;
   do i= 1 to 5;
      x=25+0.7*rannor(6789);
      output;
     end;
   do i= 6 to 10;
      x=3+0.2*rannor(6789);
      output;
   end;
   do i =11 to &n_obs;
      x=10+1*rannor(6789);
      output;
   end;
run;

%macro mycompare;
  data final;
    merge a a_good(keep=i in=b) temp(keep=i in=c);
    by i;
    format flag $45.;
    if (not b) and (not c) then flag="Removed by Both IQR and Chauv  Method";
    if (not b) and c then flag="Removed by IQR ONLY";
    if b and (not c) then flag="Removed by Chauv Method ONLY";
  run;

  proc freq data=final; table flag; run;
%mend mycompare;

%macro iqr(inds=a);
  proc univariate data=&inds noprint;
     var x;
     output out=sum_&inds mean=mean median=p50 q1=p25 q3=p75 std=std;
  run;

  data sum_&inds;
     set sum_&inds;
     range=p75-p25;
     UL=p75+1.5*range;
     LL=p25-1.5*range;
     call symput ('UL', UL);
     call symput('LL', LL);
  run;

  title "Inter Quartile Range Method";
  title2 "All Data Point";
  footnote "Program: SESUG 2007 Paper SA-11       Output: outlier";

  data &inds._good &inds._bad;
     set &inds;
     if &LL<=x<=&UL then output &inds._good;
     else output &inds._bad;
  run;
  proc sql ;
    title "Number of Bad Recods";
    select count(*) into: n_bad from &inds._bad;


  title "Inter Quartile Range Method";
  title2 "Good Data Point";
  %mend iqr;

%iqr(inds=a);



%macro chauv(inds=a, loop=5);
  %do i=1 %to &loop;
    proc univariate data=&inds noprint;
       var x;
       output out=sum_&inds n=n mean=mean median=p50 q1=p25 q3=p75 std=std;
    run;

    proc sql;
      create table &inds. as
        select r.i, r.x, s.n*erfc(abs(r.x-s.mean)/s.std)>0.5 as is_good
        from sum_&inds as s
             inner join &inds as r on 1=1
        ;

      title "Number of Bad Recods in Loop &i";
      select count(*) into: n_bad from &inds where is_good=0;
    quit;

    data &inds;
       set &inds;
       group=1;
       where is_good=1;
    run;

    %if &n_bad=0 %then %do;
       %let i=1000;
       title "Chauvenet Method";
       title2 "Good Data Point after Loop End";
    %end; %else %do;
       title "Chauvenet Method";
       title2 "Good Data Point after Loop &i";
    %end;

    %mycompare;
  %end;
%mend chauv;

data temp; set a; run;
%chauv(inds=temp, loop=50);
```