# [3 Easy Ways to Find Outliers in SAS](https://sasexamplecode.com/3-easy-ways-to-find-outliers-in-sas/)

*/\* 0. Create Sample \*/*

data work.my\_data;

call streaminit(123);

do i = 1 to 1000;

my\_var = rand("Normal", 0, 1);

output;

end;

drop i;

run;

proc print data=work.my\_data (obs=10) noobs;

run;

## How to Find Outliers in SAS?

The first method to finding outliers in SAS is based on the assumption that your data follow a normal distribution.

If the normality assumption holds, then all observations that are more than 3 standard deviations away from the mean are considered to be outliers. These observations can be above or below the mean.

Note that, PROC UNIVARIATE carries out the Shapiro-Wilk test for datasets up to 2000 observations. If your dataset has more observations, then you need to use one of the other tests for normality.

If your data is not normally distributed, then you can use the [boxplot method](https://sasexamplecode.com/3-easy-ways-to-find-outliers-in-sas/#outliers-boxplot) to find outliers.

To make your code reusable and to find the outliers more efficiently, we save the p-value of the Shapiro-Wilk test, the mean, and [the standard deviation](https://sasexamplecode.com/3-easy-ways-to-calculate-the-standard-deviation-in-sas-examples/) as three macro variables with a [SELECT INTO statement](https://sasexamplecode.com/create-macro-variables-with-select-into/).

**proc sql**;

select pValue label= 'p-value' into :pvalue from work.normal\_test where test = 'Shapiro-Wilk';

select LocValue label = 'Mean' into :mean from work.measures where LocMeasure ='Mean';

select VarValue label = 'Std Dev' into :stddev from work.measures where VarMeasure ='Std Deviation';

**quit**;

**data** work.outliers\_normaldistr;

set work.my\_data;

if my\_var lt (**&amp**;mean. - **3***\*&amp;*stddev.)

or my\_var gt (**&amp**;mean. + **3***\*&amp;*stddev.) then output;

**run**;

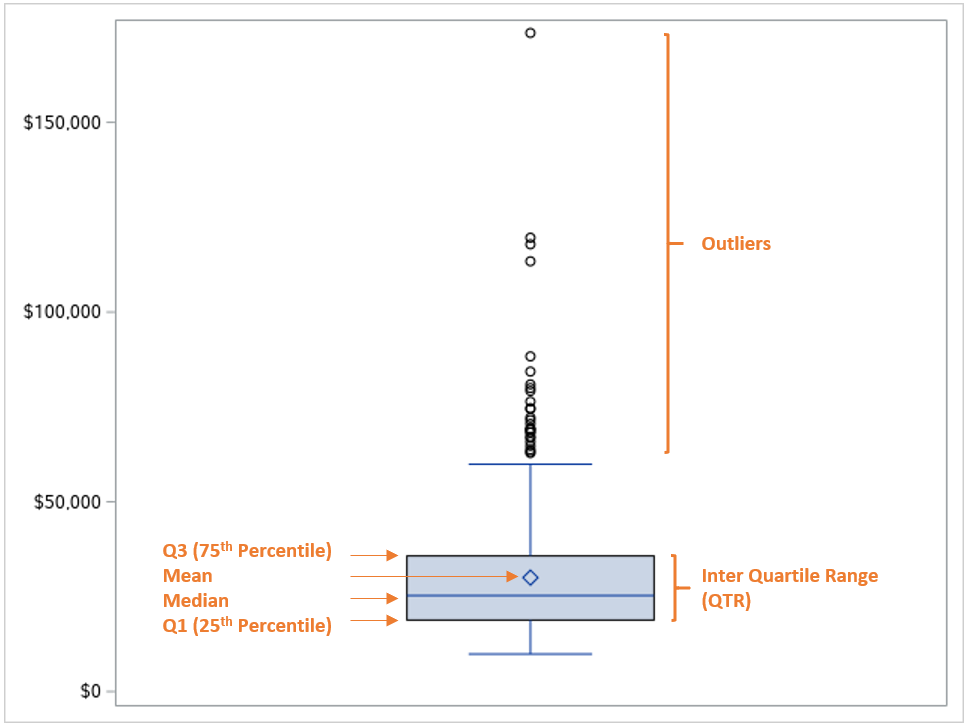
**proc print** **data**=work.outliers\_normaldistr noobs;

**run**;

## How to Find Outliers in SAS with a Boxplot?

A boxplot is a standardized way to summarize the distribution of a dataset. It does so by showing the minimum, the first quartile (Q1), the [median](https://sasexamplecode.com/3-easy-ways-to-calculate-the-median-in-sas/), the third quartile (Q3), and the maximum.

The distance between the first quartile (Q1) and third quartile (Q3) is called the Inter Quartile Range (IQR). **All observations that lie 1.5 \* IQR below the first quartile, or 1.5 \* IQR above the third quartile, are considered outliers.**



In SAS, you can use PROC UNIVARIATE to check if your data follow a normal distribution. You do this by adding the NORMAL option to the UNIVARIATE statement.

ods output TestsForNormality = work.normal\_test;

ods output BasicMeasures = work.measures;

**proc univariate** **data**=work.my\_data normal;

var my\_var;

histogram my\_var / normal;

**run**;

**proc print** **data**=work.normal\_test noobs;

**run**;

**proc print** **data**=work.measures noobs;

**run**;

ods output sgplot=work.sgplotdata;

**proc sgplot** **data**=my\_data;

vbox my\_var;

**run**;

**data** work.outliers\_boxplot (keep = Value Statistic);

set work.sgplotdata

(rename=(BOX\_MY\_VAR\_\_\_\_Y = Value

BOX\_MY\_VAR\_\_\_ST = Statistic));

where find(Statistic, "OUTLIER") **&gt**; **0**;

**run**;

**proc print** **data**=work.outliers\_boxplot noobs;

**run**;

## How to Find Outliers in SAS with Winsorization?

The third option to find outliers in SAS is with [Winsorization](https://en.wikipedia.org/wiki/Winsorizing" \l ":~:text=Winsorizing%20or%20winsorization%20is%20the,effect%20of%20possibly%20spurious%20outliers.&text=The%20effect%20is%20the%20same,be%20heavily%20influenced%20by%20outliers.).

Winsorization is a strategy to reduce the effect of outliers by removing all observations below the 5th percentile and above the 95th percentile. So, how do you apply winsorization to find outliers in SAS?

**Finding outliers with winsorization in SAS is a 3 step process. Firstly, you**[**calculate the 5th and 95th percentiles**](https://sasexamplecode.com/3-easy-ways-to-calculate-percentiles-in-sas-examples/)**with the PROC MEANS procedure. Secondly, you save these percentiles as macro variables using the**[**SELECT INTO statement**](https://sasexamplecode.com/create-macro-variables-with-select-into/)**. Finally, you**[**filter**](https://sasexamplecode.com/how-to-filter-data-in-sas-easily/)**the outliers with an IF statement.**

**proc means** **data**=work.my\_data p1 p99;

var my\_var;

output out=work.percentiles\_p1\_p99

p1 = P\_1

p99 = P\_99;

**run**;

**proc sql**;

select p\_1 into :p1 from work.percentiles\_p1\_p99;

select p\_99 into :p99 from work.percentiles\_p1\_p99;

**quit**;

**data** work.outliers\_winsorization;

set work.my\_data;

if my\_var lt **&amp**;p1.

or my\_var gt **&amp**;p99. then output;

**run**;

Complete code

data work.my\_data;

call streaminit(123);

do i = 1 to 1000;

my\_var = rand("Normal", 0, 1);

output;

end;

drop i;

run;

proc print data=work.my\_data (obs=10) noobs;

run;

ods output TestsForNormality = work.normal\_test;

ods output BasicMeasures = work.measures;

proc univariate data=work.my\_data normal;

var my\_var;

histogram my\_var / normal;

run;

proc print data=work.normal\_test noobs;

run;

proc print data=work.measures noobs;

run;

proc sql;

select pValue label= 'p-value' into :pvalue from work.normal\_test where test = 'Shapiro-Wilk';

select LocValue label = 'Mean' into :mean from work.measures where LocMeasure ='Mean';

select VarValue label = 'Std Dev' into :stddev from work.measures where VarMeasure ='Std Deviation';

quit;

data work.outliers\_normaldistr;

set work.my\_data;

if my\_var lt (&mean. - 3\*&stddev.)

or my\_var gt (&mean. + 3\*&stddev.) then output;

run;

proc print data=work.outliers\_normaldistr noobs;

run;

ods output sgplot=work.sgplotdata;

proc sgplot data=my\_data;

vbox my\_var;

run;

data work.outliers\_boxplot (keep = Value Statistic);

set work.sgplotdata

(rename=(BOX\_MY\_VAR\_\_\_\_Y = Value

BOX\_MY\_VAR\_\_\_ST = Statistic));

where find(Statistic, "OUTLIER") &gt; 0;

run;

proc print data=work.outliers\_boxplot noobs;

run;

proc means data=work.my\_data p1 p99;

var my\_var;

output out=work.percentiles\_p1\_p99

p1 = P\_1

p99 = P\_99;

run;

proc sql;

select p\_1 into :p1 from work.percentiles\_p1\_p99;

select p\_99 into :p99 from work.percentiles\_p1\_p99;

quit;

data work.outliers\_winsorization;

set work.my\_data;

if my\_var lt &p1.

or my\_var gt &p99. then output;

run;

proc print data=work.outliers\_winsorization noobs;

run;