

# Chapter 47

## Graphical Enhancements

### Chapter Table of Contents

---

<b>OVERVIEW</b> . . . . .	1681
<b>DISPLAYING STRATIFIED PROCESS DATA</b> . . . . .	1682
Displaying Stratification in Levels of a Classification Variable . . . . .	1683
Displaying Stratification in Blocks of Observations . . . . .	1684
Displaying Stratification in Phases . . . . .	1689
<b>DISPLAYING MULTIPLE SETS OF CONTROL LIMITS</b> . . . . .	1692
<b>DISPLAYING AUXILIARY DATA WITH STARS</b> . . . . .	1701
Creating a Basic Star Chart . . . . .	1702
Adding Reference Circles to Stars . . . . .	1703
Specifying the Style of Stars . . . . .	1705
Specifying the Method of Standardization . . . . .	1708
<b>DISPLAYING TRENDS IN PROCESS DATA</b> . . . . .	1711
Step 1: Preliminary Mean and Standard Deviation Charts . . . . .	1712
Step 2: Modeling the Trend . . . . .	1713
Step 3: Displaying the Trend Chart . . . . .	1714
<b>CLIPPING EXTREME POINTS</b> . . . . .	1715
<b>LABELING AXES</b> . . . . .	1719
Default Labels . . . . .	1719
Labeling the Horizontal Axis . . . . .	1720
Labeling the Vertical Axis . . . . .	1720
<b>SELECTING SUBGROUPS FOR COMPUTATION AND DISPLAY</b> . . . . .	1723
Using WHERE Statements . . . . .	1723
Using Switch Variables . . . . .	1726



# Chapter 47

## Graphical Enhancements

---

### Overview

This chapter provides details on the following topics:

- displaying process data stratified into levels using a *symbol-variable*
- displaying process data stratified into blocks using *block-variables*
- displaying process data stratified into time phases using the READPHASES= option
- displaying multiple sets of control limits using the READPHASES= and READINDEXES= options
- displaying multivariate process data using star charts
- displaying trends in process data
- clipping extreme points to create more readable charts
- labeling axes
- selecting subgroups for computation and display

The options described in this chapter can be specified in all the chart statements available in the SHEWHART procedure.

## Displaying Stratified Process Data

If the data for a Shewhart chart can be classified by factors relevant to the process (for instance, machines or operators), displaying the classification on the chart can facilitate the identification of special or common causes of variation that are related to the factors. Kume (1985) refers to this type of classification as “stratification” and describes various ways to create stratified control charts.

There are important differences between stratification and subgrouping. The data must always be classified into subgroups before a control chart can be produced. Subgrouping affects how control limits are computed from the data as well as the outcome of tests for special causes (see Chapter 48, “Tests for Special Causes”). The values of the *subgroup-variable* specified in the chart statement classify the data into subgroups. In contrast, stratification is optional and involves classification variables other than the *subgroup-variable*. Displaying stratification influences how the chart is interpreted, but it does not affect control limits or tests for special causes.

This section describes three types of variables that you can specify to create stratified control charts.

- A *symbol-variable* stratifies data into levels of a classification variable.
- The *block-variables* stratify data into blocks of consecutive observations.
- A *\_PHASE\_* variable stratifies data into *time phases*.

You can specify any combination of these three variables. You should be careful, however, since it is possible to generate confusing charts by overusing these methods.

The data for the examples in this section consist of diameter measurements for a part produced on one of three different machines. Three subgroups, each consisting of six parts, are sampled each day, corresponding to three shifts worked each day. The data are provided in the data set PARTS, which is created by the following statements:

```
data parts;
  length machine $ 4;
  input sample machine $ day shift diamx diams;
  diamn=6;
  datalines;
1  A386  01  1  4.32  0.39
2  A386  01  2  4.49  0.35
3  A386  01  3  4.44  0.44
4  A386  02  1  4.45  0.17
5  A386  02  2  4.21  0.53
6  A386  02  3  4.56  0.26
7  A386  03  1  4.63  0.39
8  A386  03  2  4.38  0.47
9  A386  03  3  4.47  0.40
10 A455  04  1  4.42  0.37
11 A455  04  2  4.45  0.32
12 A455  04  3  4.62  0.36
13 A455  05  1  4.33  0.31
14 A455  05  2  4.29  0.33
15 A455  05  3  4.17  0.25
```

```

16 C334 08 1 4.15 0.28
17 C334 08 2 4.21 0.33
18 C334 08 3 4.16 0.19
19 C334 09 1 4.14 0.13
20 C334 09 2 4.11 0.19
21 C334 09 3 4.10 0.27
22 C334 10 1 3.99 0.14
23 C334 10 2 4.24 0.16
24 C334 10 3 4.23 0.14
25 A386 11 1 4.27 0.28
26 A386 11 2 4.70 0.45
27 A386 11 3 4.51 0.45
28 A386 12 1 4.34 0.16
29 A386 12 2 4.38 0.29
30 A386 12 3 4.28 0.24
31 A386 15 1 4.47 0.26
32 A386 15 2 4.31 0.46
33 A386 15 3 4.52 0.33
;

```

## Displaying Stratification in Levels of a Classification Variable

To display process data stratified into levels of a classification variable, specify the name of this variable after an equal sign (=) immediately following the *subgroup-variable* in the chart statement. The classification variable, referred to as the *symbol-variable*, must be a variable in the input data set (a DATA=, HISTORY=, or TABLE= data set). The subgroup summary statistics are classified into groups according to the levels of the *symbol-variable* and are identified on the chart with unique plotting symbols.

See SHWSYM1  
in the SAS/QC  
Sample Library

If you use a graphics device, you can specify the symbols with SYMBOL statements. It is recommended that you place the SYMBOL statements before the PROC SHEWHART statement. If you omit the SYMBOL statements, the procedure uses the default symbol (+) for all levels of the *symbol-variable* but plots the points for each level in a distinct color. The following example illustrates the use of a *symbol-variable* to stratify the points on an  $\bar{X}$  chart according to the machine that produced the parts in each subgroup:

```

symbol1 c=black value=star;
symbol2 c=black value=circle;
symbol3 c=black value=triangle;
title 'Control Chart for Diameter Stratified by Machine';

proc shewhart history=parts;
  xchart diam*sample=machine / stddeviations
                                symbollegend=legend1;
  label sample = 'Sample Number'
        diamx  = 'Average Diameter' ;
  legend1 label=('Machine') frame;
run;

```

The symbols are specified with the SYMBOL1, SYMBOL2, and SYMBOL3 statements. The SYMBOLLEGEND= option requests a customized legend for the sym-

bols. For more information on the LEGEND and SYMBOL statements, refer to *SAS/GRAPH Software: Reference*. The  $\bar{X}$  chart, shown in Figure 47.1, reveals an effect due to MACHINE. In particular, Machine C334 is associated with a run of parts whose diameters are systematically below average, suggesting that this machine may require adjustment.

For charts produced on a line printer, you can use the SYMBOLCHARS= option to specify the characters that identify the stratification of the points. For details, see the entry for the SYMBOLCHARS= option in Chapter 46, "Dictionary of Options."

In this example, Machine A386 is associated with two different blocks of observations that are identified with a common symbol. However, a *symbol-variable* is particularly useful for situations where the stratification is not necessarily chronological or associated with blocks of consecutive groups of observations.

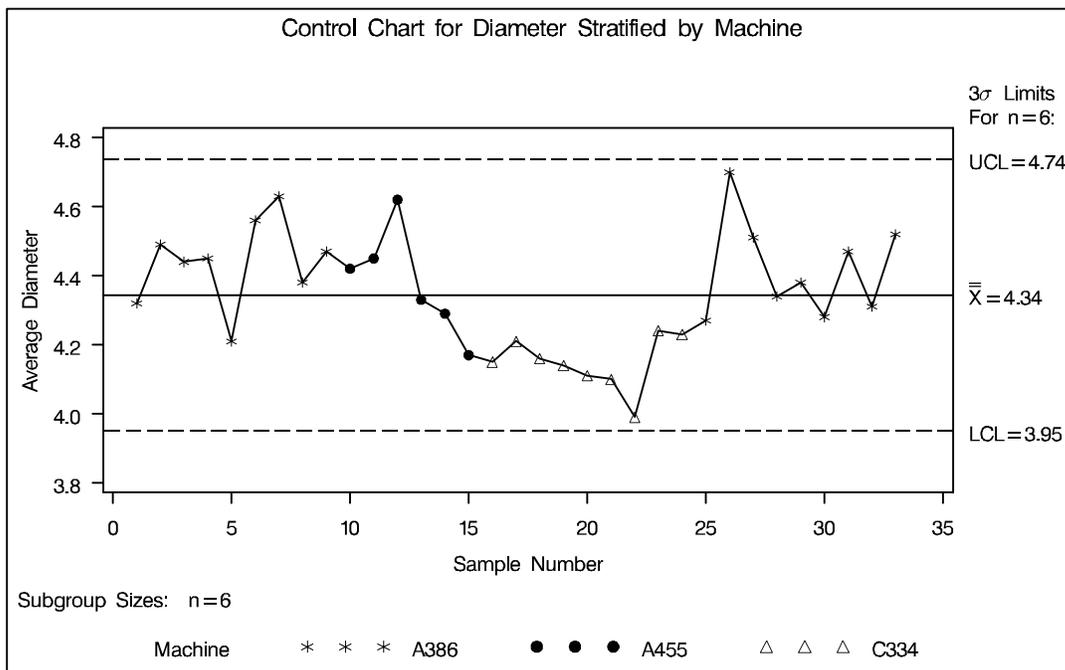


Figure 47.1. Control Chart Stratified into Levels Using Symbols

## Displaying Stratification in Blocks of Observations

See SHWBLK  
in the SAS/QC  
Sample Library

To display process data stratified into blocks of consecutive observations, specify one or more *block-variables* in parentheses after the *subgroup-variable* in the chart statement. The procedure displays a legend identifying blocks of consecutive observations with identical values of the *block-variables*. The legend displays one track of values for each *block-variable*. The values are the formatted values of the *block-variable*. For example, Figure 47.2 displays a legend with a single track for MACHINE, while Figure 47.3 displays a legend with two tracks corresponding to MACHINE and DAY. You can label the tracks themselves by using the LABEL statement to associate labels with the corresponding *block-variables*; see Figure 47.4 on page 1687 for an illustration.

By default, the legend is placed above the chart as in Figure 47.2. You can control the position of the legend with the `BLOCKPOS=` option and the position of the legend labels with the `BLOCKLABELPOS=` option. See the entries in Chapter 46, “Dictionary of Options,” as well as the following examples.

The *block-variables* must be variables in the input data set (a `DATA=`, `HISTORY=`, or `TABLE=` data set). If the input data set is a `DATA=` data set that contains multiple observations with the same value of the *subgroup-variable*, the values of a *block-variable* must be the same for all observations with the same value of the *subgroup-variable*. In other words, subgroups must be nested within groups determined by *block-variables*. The following statements create an  $\bar{X}$  chart for the data in `PARTS` stratified by the *block-variable* `MACHINE`. The chart is shown in Figure 47.2.

```
symbol value=dot;
title 'Control Chart for Diameter Stratified By Machine';

proc shewhart history=parts;
  xchart diam*sample (machine) / stddeviations ;
  label sample = 'Sample Number'
        diamx  = 'Average Diameter' ;
run;
```

The unique consecutive values of `MACHINE` (A386, A455, C334, and A386) are displayed in a track above the chart, and they indicate the same relationship between part diameter and machine as the previous example. Note that the track is not labeled (as in Figure 47.4), since no label is associated with `MACHINE`. A `LABEL` statement is used to provide labels for the axes.

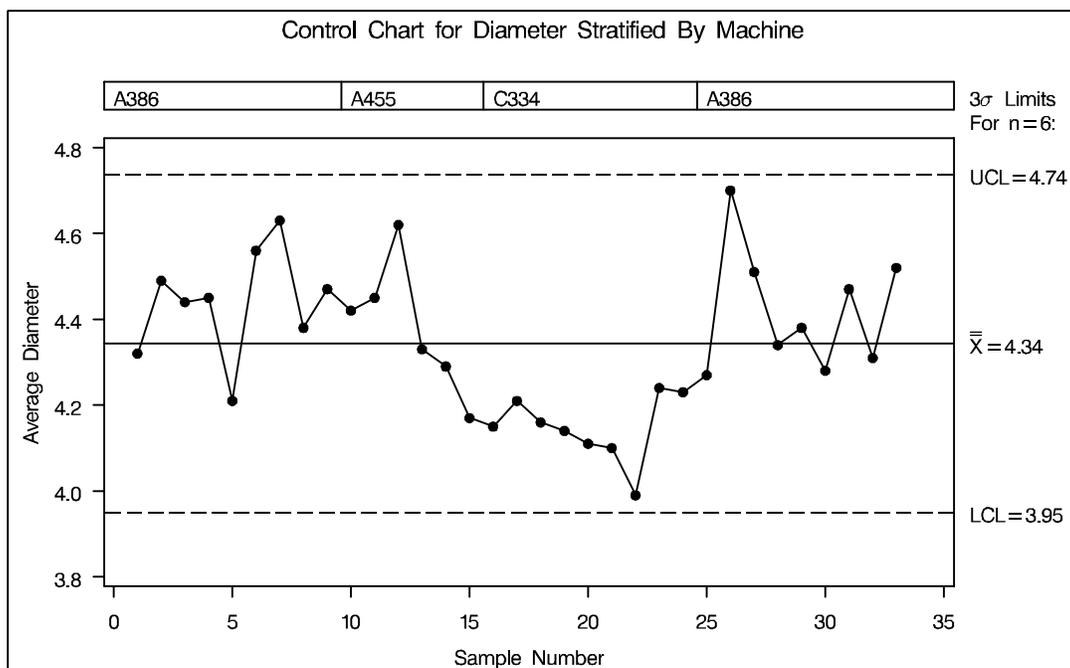


Figure 47.2. Stratified Control Chart Using a Single Block Variable

**Multiple block variables.** You can use multiple *block-variables* to study more than one classification factor with the same chart. The following statements create an  $\bar{X}$  chart for the data in PARTS, with MACHINE and DAY as *block-variables*:

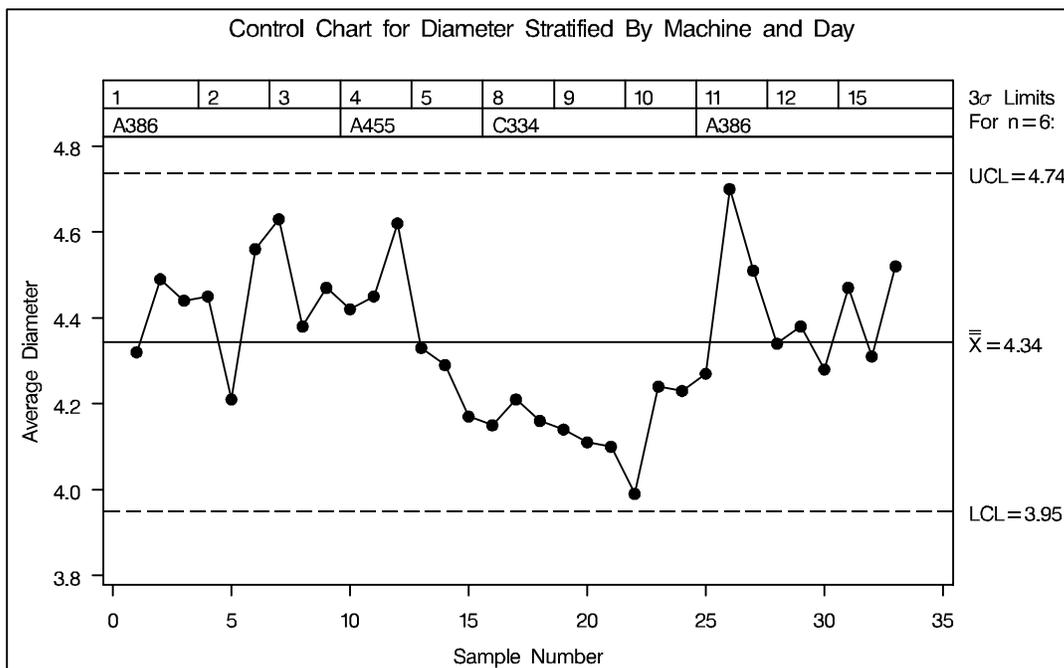
```

title 'Control Chart for Diameter Stratified By Machine and Day';

proc shewhart history=parts;
  xchart diam*sample (machine day) /
    stdeviations
    nolegend
    blockpos = 2 ;
  label sample = 'Sample Number'
        diamx  = 'Average Diameter' ;
run;

```

The chart is displayed in Figure 47.3. Specifying BLOCKPOS=2 displays the *block-variable* legend immediately above the chart, without the gap shown in Figure 47.2. The NOLEGEND option suppresses the sample size legend that appears in the lower left of Figure 47.2.



**Figure 47.3.** Stratified Control Chart Using Multiple Block Variables

**Color fills for legend.** You can use the CBLOCKVAR= option to fill the legend track sections with colors corresponding to the values of the *block-variables*. Provide the colors as values of variables specified with the CBLOCKVAR= option. These variables must be defined as character variables of length 8. The procedure matches the color variables with the *block-variables* in the order specified. Each section is filled with the color for the first observation in the block. For example, the following statements produce an  $\bar{X}$  chart using a color variable named CMACHINE to fill the legend for the *block-variable* MACHINE:

```

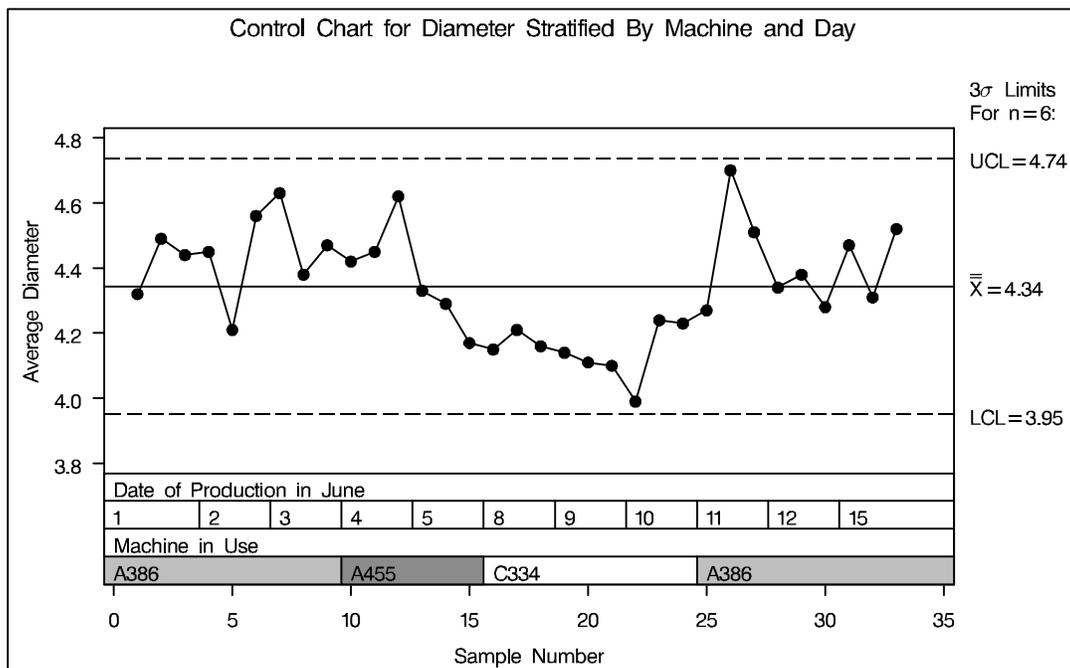
data parts2;
  length cmachine $8;
  set parts;
  if      machine='A386' then cmachine='ligr' ;
  else if machine='A455' then cmachine='megr' ;
  else if machine='C334' then cmachine='white';
  else                               cmachine='dagr' ;

proc shewhart history=parts2;
  xchart diam*sample (machine day) / stddeviations
                                nolegend
                                blockpos   = 3
                                cblockvar  = cmachine;

  label sample   = 'Sample Number'
        diamx    = 'Average Diameter'
        day      = 'Date of Production in June'
        machine  = 'Machine in Use';

run;

```



**Figure 47.4.** Color Fill for *Block-Variable* Legend

The sections for Machine A386 are filled with light gray, the section for Machine A455 is filled with medium gray, and the section for Machine C334 is left white. The legend track for DAY is not filled, since a second color variable was not specified with the CBLOCKVAR= option. Specifying BLOCKPOS=3 positions the legend at the bottom of the chart and facilitates comparison with the subgroup axis. The LABEL statement is used to label the tracks with the labels associated with the *block-variables*.

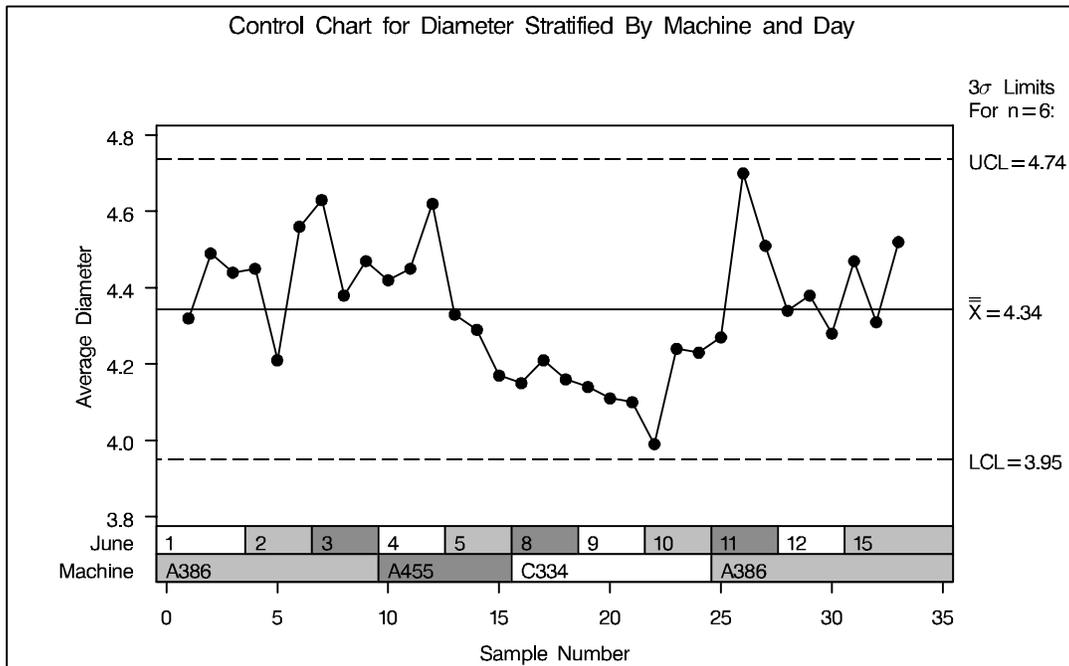
## Part 9. The CAPABILITY Procedure

The following statements produce an  $\bar{X}$  chart in which both legend tracks are filled:

```
data parts3;
  length cday $8;
  set parts2;
  if      day='01' then cday='white';
  else if day='02' then cday='ligr' ;
  else if day='03' then cday='megr' ;
  else if day='04' then cday='white';
  else if day='05' then cday='ligr' ;
  else if day='08' then cday='megr' ;
  else if day='09' then cday='white';
  else if day='10' then cday='ligr' ;
  else if day='11' then cday='megr' ;
  else if day='12' then cday='white';
  else if day='15' then cday='ligr' ;
  else
      cday='black';

proc shewhart history=parts3;
  xchart diam*sample (machine day) /
    stddeviations
    nolegend
    ltmargin      = 5
    blockpos      = 3
    blocklabelpos = left
    cblockvar     = (cmachine cday);
  label sample   = 'Sample Number'
        diamx    = 'Average Diameter'
        day      = 'June'
        machine  = 'Machine';
run;
```

The chart is displayed in Figure 47.5. The color values of CMACHINE are used to fill the track for MACHINE, and the color values of CDAY are used to fill the track for DAY. Specifying BLOCKLABELPOS=LEFT displays the block variable labels to the left of the block legend. The LTMARGIN= option provides extra space in the left margin to accommodate the label *Machine*.



**Figure 47.5.** Stratified Control Chart Using Multiple Block Variables

## Displaying Stratification in Phases

The preceding section describes the use of *block-variables* to display blocks of consecutive observations that correspond to changes in factors such as machines, shifts, and raw materials. This section describes the use of a *\_PHASE\_* variable to display phases of consecutive observations (as in Figure 47.6). Although the terms *block* and *phase* have similar meanings, there are differences in the two methods:

See SHWPHSE  
in the SAS/QC  
Sample Library

- You can provide only one *\_PHASE\_* variable, whereas you can specify multiple *block-variables*.
- You can display distinct control limits for each phase (see page 1692) but not for each block.
- Different sets of graphical options are available for identifying blocks and phases.

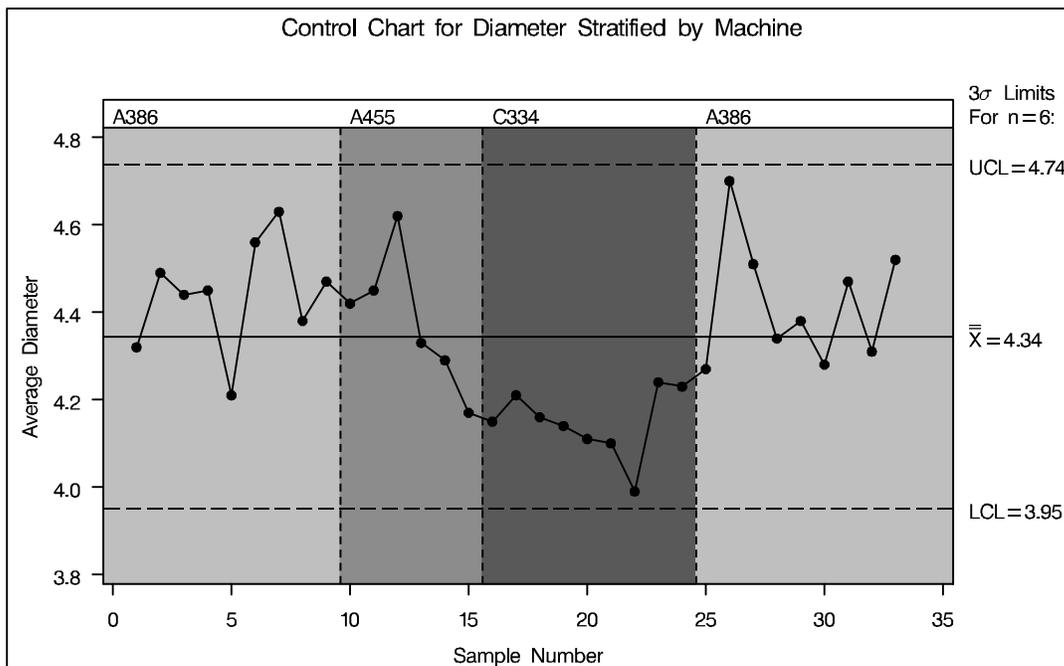
To display phases, your input data set must include a character variable named *\_PHASE\_* of length 16 or less, and you must specify the *READPHASES=* option in the chart statement. (If your data set does not include a variable named *\_PHASE\_*, you can temporarily rename another character variable to *\_PHASE\_*, as illustrated by the following statements.) The procedure classifies the data into phases (groups of consecutive observations with the same value of *\_PHASE\_*) and reads only those observations whose *\_PHASE\_* value matches one of the values specified with the *READPHASES=* option.

## Part 9. The CAPABILITY Procedure

You can identify and highlight the phases with various options, as illustrated by the following statements, which produce the chart shown in Figure 47.6. The PHASELEGEND option displays a legend with the \_PHASE\_ values, and the CPHASELEG= option specifies the color of the legend text. The PHASEREF option delineates the phases with vertical reference lines. The CFRAME= option fills the framed areas for the phases with different colors.

```
symbol v=dot;
title 'Control Chart for Diameter Stratified by Machine';

proc shewhart history=parts(rename=(machine=_phase_));
  xchart diam*sample /
    stddeviations
    readphases = ('A386' 'A455' 'C334' 'A386')
    cframe      = ( ligr  megr  dagr  ligr )
    phaselegend
    cphaseleg  = black
    phaseref
    nolegend;
  label  sample = 'Sample Number'
        diamx  = 'Average Diameter';
run;
```



**Figure 47.6.** Control Chart Stratified by Phases

Note that the data set PARTS does not contain a variable named \_PHASE\_, so the variable MACHINE is renamed as \_PHASE\_ for the duration of the procedure step.

The observations read from PARTS are those whose value of MACHINE matches one of the values listed with the READPHASES= option in that order. Here, the value A386 is listed twice; consequently, both groups of observations for which MACHINE equals A386 are read.

In this example, the input data set contains a single observation for each subgroup. If your input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the value of `_PHASE_` must be the same for all observations with the same value of the *subgroup-variable*. Thus, in general, subgroups must be nested within phases.

Recall that the horizontal axis scale is determined by the *subgroup-variable* (see “Subgroup Variables” on page 1534). If your *subgroup-variable* is numeric, this scale is continuous; consequently, you should select phases that are reasonably contiguous in order to avoid large empty gaps in your chart. For instance, if you were to specify

```
readphases = ('A386' 'A455' 'A386')
```

in the preceding XCHART statement, there would be a gap between the 15<sup>th</sup> and 25<sup>th</sup> points (these points would be connected unless you specified the PHASEBREAK option). You can avoid gaps by specifying a character *subgroup-variable*\* for which a discrete horizontal axis scale will be displayed.

Note that the values listed in the READPHASES= option must be listed in the same order as they occur in the input data set. Thus, in order to display all the observations in the data set PARTS, A386 must be listed as both the first and last value. An alternative method for selecting all the phases from your input data is to specify READPHASES=ALL, as described in the next section.

The control limits shown in Figure 47.6 are computed from the data and are, therefore, the same across all phases. More generally, you can display a distinct set of control limits for each phase. To do so, you must provide the control limits in a LIMITS= data set and specify the READINDEXES= option in addition to the READPHASES= option, as described in the next section.

\*You can use the PUT function in a DATA step to create a character *subgroup-variable* from a numeric *subgroup-variable*.

---

## Displaying Multiple Sets of Control Limits

See SHWCLMS  
in the SAS/QC  
Sample Library

This section describes the use of the READPHASES= and READINDEXES= options for creating Shewhart charts that display distinct sets of control limits for multiple phases of observations. The term *phase* refers to a group of consecutive observations in the input data set. For example, the phases might correspond to the time periods during which a new process was brought into production and then put through successive changes.

To display phases, your input data must include a character variable named `_PHASE_`, whose length cannot exceed 16. (If your data set does not include a variable named `_PHASE_`, you can temporarily rename another character variable to `_PHASE_`, as illustrated in the statements in Displaying Stratification in Phases, on page 1690.) Each phase consists of a group of consecutive observations with the same value of `_PHASE_`.

To display distinct sets of predetermined control limits for the phases, you must provide the limits in a LIMITS= data set. This data set must include a character variable named `_INDEX_`, whose length cannot exceed 16. This variable identifies the sets of control limits (observations) in the LIMITS= data set that are to be associated with the phases. This data set must also include a number of other variables with reserved names that begin and end with an underscore. The particular structure of a LIMITS= data set depends on the chart statement that you are using; for details, see the sections titled “LIMITS= Data Set” in the chapters for the various chart statements. In addition to specifying a LIMITS= data set, you must also specify the READINDEXES= and READPHASES= options in the chart statement.

**Note:** To display a *single* set of predetermined control limits with multiple phases, simply specify a LIMITS= data set in the procedure statement. If you are using Release 6.09 or an earlier release, you must also specify the READLIMITS option. The control limits are read from the first observation in the LIMITS= data for which the variable `_VAR_` is equal to the name of the *process* and the variable `_SUBGRP_` is equal to the name of the *subgroup-variable*. For an example, see “Reading Preestablished Control Limits” on page 1512.

This section describes the combinations of the READINDEXES= and READPHASES= options that you can specify. The examples that follow use the HISTORY= data set FLANGE listed in Figure 47.7 and the LIMITS= data set FLANGLIM listed in Figure 47.8. The data in FLANGE consist of means and ranges of flange width measurements for subgroups of size five. The observations are grouped into three phases determined by the `_PHASE_` values `Production`, `Change 1`, and `Change 2`. Three sets of control limits are provided in FLANGLIM, corresponding to the `_INDEX_` values `Start`, `Production`, and `Change 1`.

Obs	_phase_	day	sample	flwidthx	flwidthr	flwidthn
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
4	Production	11FEB90	9	0.95509	0.08378	5
5	Production	12FEB90	10	1.00348	0.09993	5
6	Production	15FEB90	11	1.02566	0.06766	5
7	Production	16FEB90	12	0.97053	0.07608	5
8	Production	17FEB90	13	0.94713	0.10170	5
9	Production	18FEB90	14	1.00377	0.04875	5
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
13	Change 1	24FEB90	18	1.02235	0.10541	5
14	Change 1	25FEB90	19	0.99950	0.11476	5
15	Change 1	26FEB90	20	0.99271	0.05395	5
16	Change 1	01MAR90	21	0.98695	0.03833	5
17	Change 1	02MAR90	22	1.00969	0.06183	5
18	Change 1	03MAR90	23	0.98791	0.05836	5
19	Change 1	04MAR90	24	1.00170	0.05243	5
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
23	Change 2	10MAR90	28	1.01463	0.05558	5
24	Change 2	11MAR90	29	0.99812	0.03648	5
25	Change 2	12MAR90	30	1.00047	0.04309	5
26	Change 2	15MAR90	31	0.99714	0.03689	5
27	Change 2	16MAR90	32	0.98642	0.04809	5
28	Change 2	17MAR90	33	0.98891	0.07777	5
29	Change 2	18MAR90	34	1.00087	0.06409	5
30	Change 2	19MAR90	35	1.00863	0.02649	5

Figure 47.7. Listing of the HISTORY= Data Set FLANGE

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_
1	Change 1	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
2	Production	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
3	Start	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3
Obs	_lclx_	_mean_	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	0.96167	0.99924	1.03680	0	0.06513	0.13771	0.028000
2	0.93792	0.98827	1.03862	0	0.08729	0.18458	0.037530
3	0.87088	0.96803	1.06517	0	0.16842	0.35612	0.072409

Figure 47.8. Listing of the LIMITS= Data Set FLANLIM

For each of the READINDEXES= and READPHASES= options, you can specify a single value, a list of values, or the keyword ALL. You can also leave these options unspecified. Thus, there are 16 possible combinations of specifications for the two options, as explained by the following table and notes. The two most commonly encountered combinations are

- reading a single set of limits for one or more phases (see Note 1)
- reading a set of limits matched with a set of phases (see Note 4)

**Table 47.1.** Combinations of READPHASES= and READINDEXES= Specifications

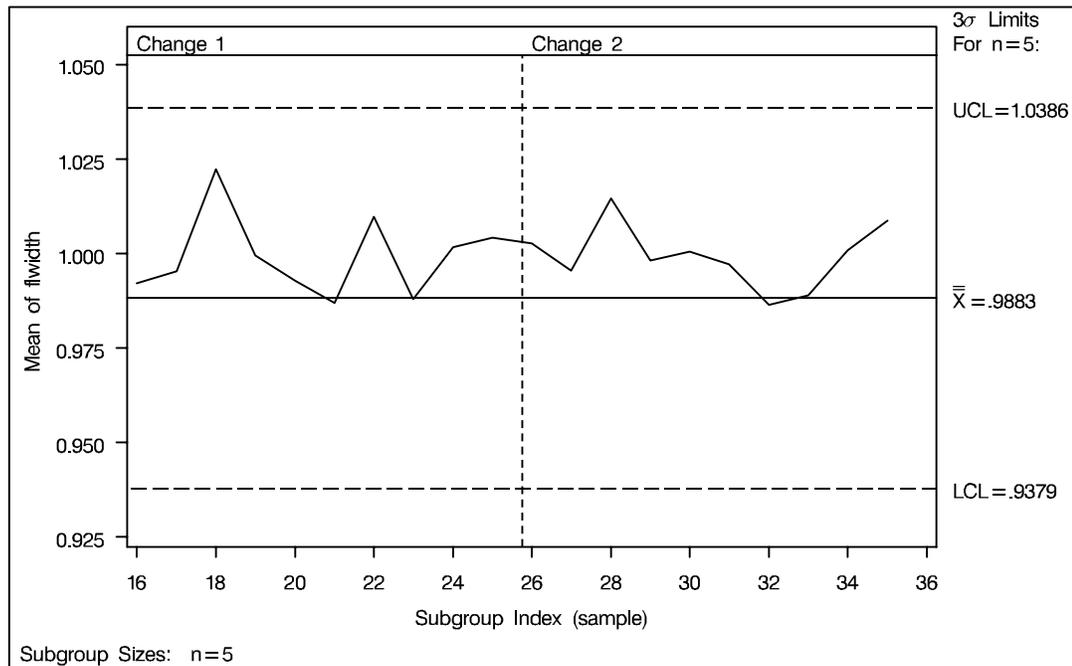
READINDEXES=	READPHASES=			
	Single Value	Multiple Values	Keyword ALL	Not Specified
Single Value	See Note 1	See Note 1	See Note 2	See Note 3
Multiple Values	See Note 9	See Note 4	See Note 2	See Note 2
Keyword ALL	See Note 5	See Note 5	See Note 6	See Note 6
Not Specified	See Note 7	See Note 7	See Note 8	See Note 8

**Note 1.** READPHASES=*value|value-list* and READINDEXES=*value*

The only phases (groups of observations) read are those for which `_PHASE_` equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits given by the first observation in the LIMITS= data set for which `_INDEX_` is equal to the READINDEXES= *value*.

For example, the following statements create a chart for the phases Change 1 and Change 2, with control limits read from the second observation in FLANGLIM. The chart is displayed in Figure 47.9.

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphase = ('Change 1' 'Change 2')
    readindex = ('Production')
    phaseref
    phaselegend ;
run;
```



**Figure 47.9.** A Single Set of Control Limits for Multiple Phases

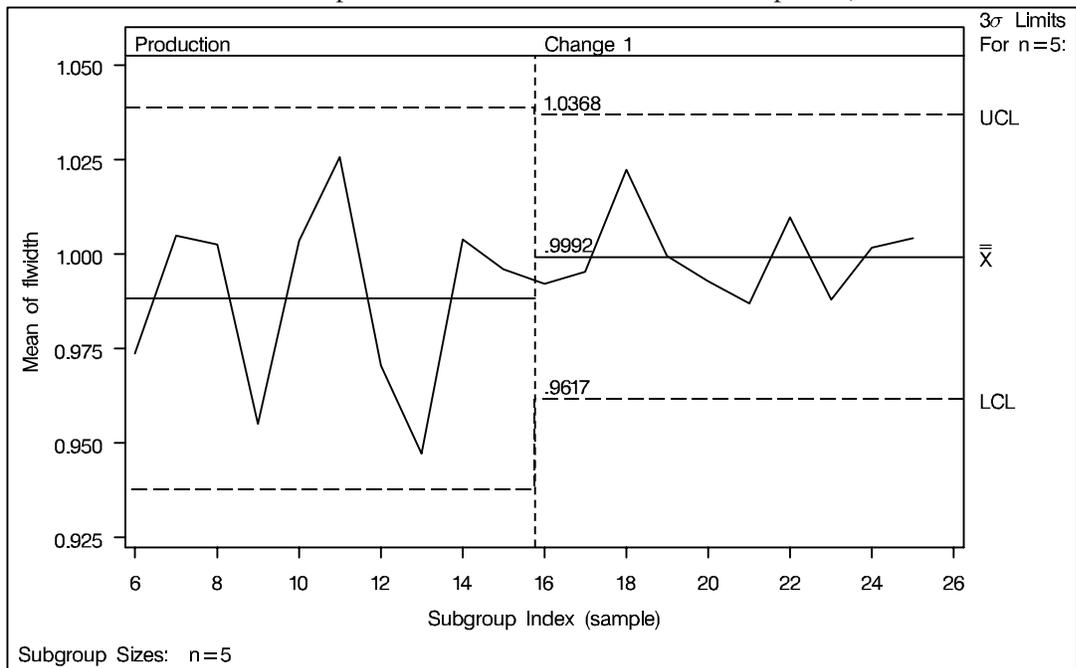
**Note 2.** `READPHASES=ALL` and `READINDEXES=value|value-list` or `READPHASES=` is omitted and `READINDEXES=value-list`

The only phases read are those for which `_PHASE_` equals one of the *values* specified with the `READINDEXES=` option. The chart displays a different set of control limits for each phase, read from the first observation in the `LIMITS=` data set for which `_INDEX_` is equal to the corresponding *value*.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the second and first observations in `FLANGLIM`, respectively. The chart is displayed in Figure 47.10.

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphase = all
    readindex = ('Production' 'Change 1')
    phaseref
    phaselegend ;
run;
```

If you wish to specify a single set of control limits to use with all the phases, use the `READINDEXES=` option *without* the `READPHASES=` option (see Note



**Figure 47.10.** `READPHASES=ALL` with a List of Values for `READINDEXES=`

**Note 3.** `READPHASES=` is omitted and `READINDEXES=value`

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the `LIMITS=` data for which `_INDEX_` equals the *value*.

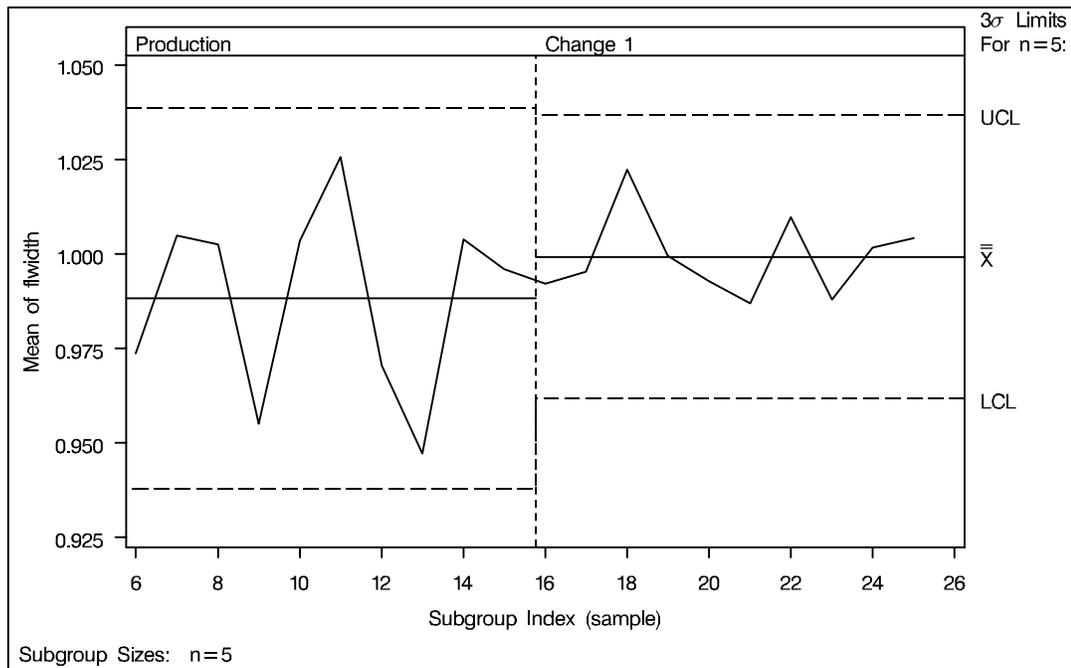
**Note 4. READPHASES=*value-list* and READINDEXES=*value-list***

The only phases read are those for which `_PHASE_` equals one of the values specified with the `READPHASES=` option. The chart displays a different set of control limits for each phase, given by the first observation in the `LIMITS=` data set for which `_INDEX_` equals the `READINDEXES=value`. Control limits are matched with phases in the order listed.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the first and second observations in `FLANGLIM`, respectively. The chart produced by these statements is identical to the chart in Figure 47.10.

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphases = ('Production' 'Change 1')
    readindexes = ('Production' 'Change 1')
    phaseref
    phaselegend ;
run;
```

The order of the `READINDEX=value-list` is critical. For instance, the previous statements with `READINDEXES=('Change 1' 'Production')` create the chart in Figure 47.11, in which the control limits are mismatched with the phases.



**Figure 47.11.** Multiple Phases with Mismatched Control Limits

**Note 5. READPHASES=*value*|*value-list* and READINDEXES=ALL**

The only phases read are those for which `_PHASE_` equals one of the *values* specified with the `READPHASES=` option. The chart displays a different set of control limits for each phase, read from the first observation in the `LIMITS=` data set for which `_INDEX_` equals the *value* corresponding to the phase.

For example, the following statements create a chart for the phases `Production` and `Change 1` with the control limits read from the second and first observations in `FLANGLIM`, respectively:

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphases = ('Production' 'Change 1')
    readindexes = all
    phaseref
    phaselegend ;
run;
```

The chart is identical to the chart in Figure 47.10. In general, to read a set of phases with identically labeled control limits, you can specify the phases with either the `READPHASES=` or `READINDEXES=` option, and you can specify the keyword `ALL` with the other option.

**Note 6. READPHASES=ALL and READINDEXES=ALL or  
READPHASES= is omitted and READINDEXES=ALL**

All phases are read for which `_PHASE_` is a value of `_INDEX_` in the `LIMITS=` data set. The chart displays a different set of control limits for each phase, read from the first observation in the `LIMITS=` data set for which `_INDEX_` equals the value of `_PHASE_`.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the second and first observations in `FLANGLIM`, respectively. These two phases are read because they are the only phases in `FLANGE` with matching `_INDEX_` values in `FLANGLIM`. The chart is identical to that in Figure 47.10.

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphase = all
    readindex = all
    phaseref
    phaselegend ;
run;
```

Note that an identical chart would be produced if you were to omit the `READPHASES=` option.

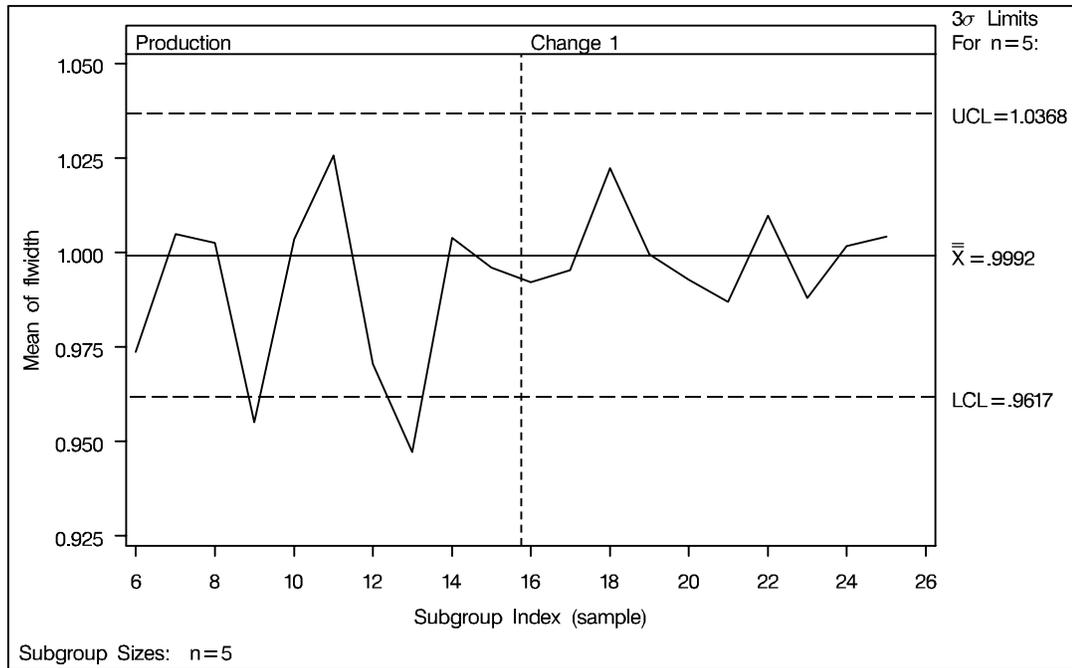
**Note 7. READPHASES=*value|value-list* and READINDEXES= is omitted**

The only phases read are those for which `_PHASE_` equals one of the *values* specified with the `READPHASES=` option. The chart displays a single set of control limits read from the first observation in the `LIMITS=` data set for which `_VAR_` equals the *process* and `_SUBGRP_` equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for the phases `Production` and `Change 1` with control limits read from the first observation in `FLANGLIM`, because this is the first observation for which `_VAR_` equals `FLWIDTH` and `_SUBGRP_` equals `SAMPLE`.

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphase = ('Production' 'Change 1')
    phaseref
    phaselegend ;
run;
```

The chart is displayed in Figure 47.12.



**Figure 47.12.** *Value-list* for `READPHASES=` with `READINDEXES=` Omitted

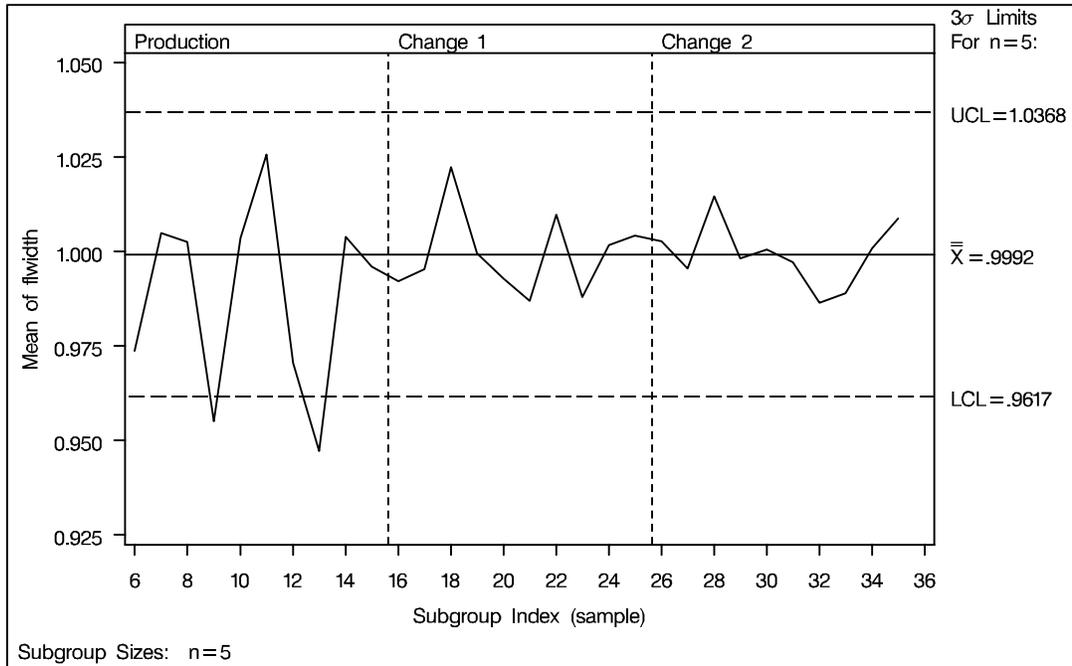
**Note 8. READPHASES=ALL and READINDEXES= is omitted or READPHASES= is omitted and READINDEXES= is omitted**

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which \_VAR\_ equals the *process* and \_SUBGRP\_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for all the phases in FLANGE with control limits read from the first observation in FLANGLIM, because this is the first observation for which \_VAR\_ equals FLWIDTH and \_SUBGRP\_ equals SAMPLE:

```
proc shewhart history=flange limits=flanlim;
  xchart flwidth*sample /
    readphase = all
    phaseref
    phaselegend ;
run;
```

The chart is shown in Figure 47.13. Note that an identical chart would be produced if you were to omit the READPHASES= option (except that the phase reference lines and phase legends would be omitted).



**Figure 47.13.** READPHASES=ALL with READINDEXES= Omitted

**Note 9. READPHASES=value and READINDEXES=value-list**

The procedure generates an error message.

Part 9. The CAPABILITY Procedure

The following tables summarize the various combinations of the READPHASES= and READINDEXES= options that you can specify.

**Table 47.2.** READINDEXES=*index-value*

READPHASES=	Phases Displayed	Control Limits Displayed
<i>phase-value</i>	<i>_PHASE_ = phase-value</i>	<i>_INDEX_ = index-value</i>
<i>phase-value list</i>	<i>_PHASE_ = phase-value list</i>	<i>_INDEX_ = index-value</i>
Keyword ALL	<i>_PHASE_ = index-value</i>	<i>_INDEX_ = index-value</i>
Not Specified	All phases	<i>_INDEX_ = index-value</i>

**Table 47.3.** READINDEXES=*index-value list*

READPHASES=	Phases Displayed	Control Limits Displayed
<i>phase-value</i>	No chart displayed	No chart displayed
<i>phase-value list</i>	<i>_PHASE_ = phase-value list</i>	<i>_INDEX_ = index-value list</i> with control limits matched to phases in the order listed
Keyword ALL	<i>_PHASE_ = index-value list</i>	<i>_INDEX_ = index-value list</i>
Not Specified	<i>_PHASE_ = index-value list</i>	<i>_INDEX_ = index-value list</i>

**Table 47.4.** READINDEXES=ALL

READPHASES=	Phases Displayed	Control Limits Displayed
<i>phase-value</i>	<i>_PHASE_ = phase-value</i>	<i>_INDEX_ = phase-value</i>
<i>phase-value list</i>	<i>_PHASE_ = phase-value list</i>	<i>_INDEX_ = phase-value list</i>
Keyword ALL	<i>_PHASE_ = _INDEX_</i>	<i>_INDEX_ = _PHASE_</i>
Not Specified	<i>_PHASE_ = _INDEX_</i>	<i>_INDEX_ = _PHASE_</i>

**Table 47.5.** READINDEXES= Not Specified

READPHASES=	Phases Displayed	Control Limits Displayed
<i>phase-value</i>	<i>_PHASE_ = phase-value</i>	First LIMITS= observation for which <i>_VAR_ = process</i> name and <i>_SUBGRP_ = subgroup-variable</i> name
<i>phase-value list</i>	<i>_PHASE_ = phase-value list</i>	same as previous entry
Keyword ALL	All phases	same as previous entry
Not Specified	All phases	same as previous entry

## Displaying Auxiliary Data with Stars

In many control chart applications, it is useful to relate the variation of the process to other variables that are being observed simultaneously with the variable that is charted. You can use the features described here to represent auxiliary multivariate data with stars (polygons) that are superimposed on the control chart. See Figure 47.16 on page 1703 for an illustration.

See SHWSTR1  
in the SAS/QC  
Sample Library

This display, referred to here as a *star chart*, enables you to analyze a process with a control chart while visualizing other quantities such as environmental variables, experimental control variables, or other process variables. The control chart itself can be a standard Shewhart chart, a moving average chart (such as an EWMA chart), or a cumulative sum control chart.

The examples in this section use the HISTORY= input data set PAINT (listed in Figure 47.14) and the LIMITS= data set PAINTLIM (listed in Figure 47.15). The data in PAINT consist of the subgroup means, ranges, and sample size (PINDEXX, PINDEXR, and PINDEXN) for an index of paint quality that was monitored on an hourly basis, with six auxiliary variables that were measured simultaneously: thickness, gloss, defects, dust, humidity, and temperature.

hour	pindexx	pindexr	pindexn	thick	gloss	defects	dust	humid	temp
1	5.8	3.0	5	0.2550	0.6800	0.2550	0.2125	0.1700	0.5950
2	6.2	2.0	5	0.2975	0.5950	0.0850	0.1700	0.2125	0.5525
3	3.7	2.5	5	0.3400	0.3400	0.4250	0.2975	0.2550	0.2125
4	3.2	6.5	5	0.3400	0.4675	0.3825	0.3485	0.2125	0.2125
5	4.7	0.5	5	0.5100	0.4250	0.5950	0.4080	0.5100	0.4675
6	5.2	3.0	5	0.5100	0.3400	0.6800	0.5525	0.5525	0.5525
7	2.6	2.0	5	0.4250	0.0425	0.8500	0.5355	0.5525	0.2550
8	2.1	1.0	5	0.3400	0.0170	0.8075	0.5950	0.5950	0.1700

Figure 47.14. Listing of the HISTORY= Data Set PAINT

	s	l	s									
	u	i	i									
	b	t	m	g	l	m	u	l		u	t	
	v	y	i	m	c	e	c	c		c	d	
O	a	r	p	t	a	l	a	l		l	e	
b	r	p	e	n	s	x	n	x	r	r	v	
s												
1	pindex	hour	estimate	5	3	2.395	3.875	5.355	0	2.5625	5.4184	1.10171

Figure 47.15. Listing of the LIMITS= Data Set PAINTLIM

The basic variable analyzed with the control chart (in this case, paint index) is referred to as the *process*. The auxiliary variables (in this case, thickness, gloss, defects, dust, humidity, and temperature) are referred to as *vertex variables*, because their values are represented by the vertices of the stars. A star chart can reveal relationships between the process and the vertex variables, and it can reveal relationships among the vertex variables.

You can create star charts for any number of vertex variables. However, the resolution of your graphics device and the number of subgroups per page will limit your ability to distinguish the vertices of the stars. A practical upper limit is twelve vertex variables.

You can specify star options in all chart statements of the SHEWHART procedure except the BOXCHART statement. You can use these options to

- specify the style of the star
- add reference circles to indicate limits of variation for the stars
- add a legend identifying the relationship between vertices and vertex variables
- label the vertices
- specify colors and line types for individual stars
- specify the size of the stars
- specify different methods of standardization for the vertex variables

The star options apply only with control charts created with high-resolution graphics devices.

**NOTE:** A star chart is *not* the same as a multivariate control chart or a  $T^2$  chart. A star chart is simply a univariate control chart enhanced with stars that represent auxiliary multivariate data. A multivariate control chart displays summary statistics (such as  $T^2$ ) and control limits determined for a number of processes simultaneously. For an example of a multivariate control chart, see Figure 49.31 on page 1786. Figure 49.32 on page 1787 displays a multivariate control chart in which the principal components of the  $T^2$  statistic are displayed with stars.

---

## Creating a Basic Star Chart

See SHWSTR1  
in the SAS/QC  
Sample Library

The following statements create the star chart shown in Figure 47.16:

```
title 'Variables Related to Paint Index';
proc shewhart history=paint limits=paintlim;
  xchart pindex*hour /
  nolegend
  starvertices = (thick gloss defects dust humid temp);
run;
```

This chart is essentially an  $\bar{X}$  chart for paint index. However, the chart also provides information about thickness, gloss, defects, dust, humidity, and temperature. These six variables are represented by the vertices of the stars, as indicated by the legend at the bottom of the chart. By default, the legend uses a clock representation for the vertices; for instance, dust corresponds to the vertex at the six o'clock position.

The stars are centered at the points for average paint index, and the distance from the center to a vertex represents the standardized value of the variable corresponding to the vertex. The star chart reveals that relatively high values of gloss (two o'clock) and temperature (ten o'clock) are associated with high out-of-control averages for paint index. Likewise, relatively high values of defects (four o'clock) and humidity (eight o'clock) are associated with low out-of-control averages for paint index. The star shapes reveal similarities in the data for runs 1 and 2, runs 3 and 4, runs 5 and 6, and runs 7 and 8.

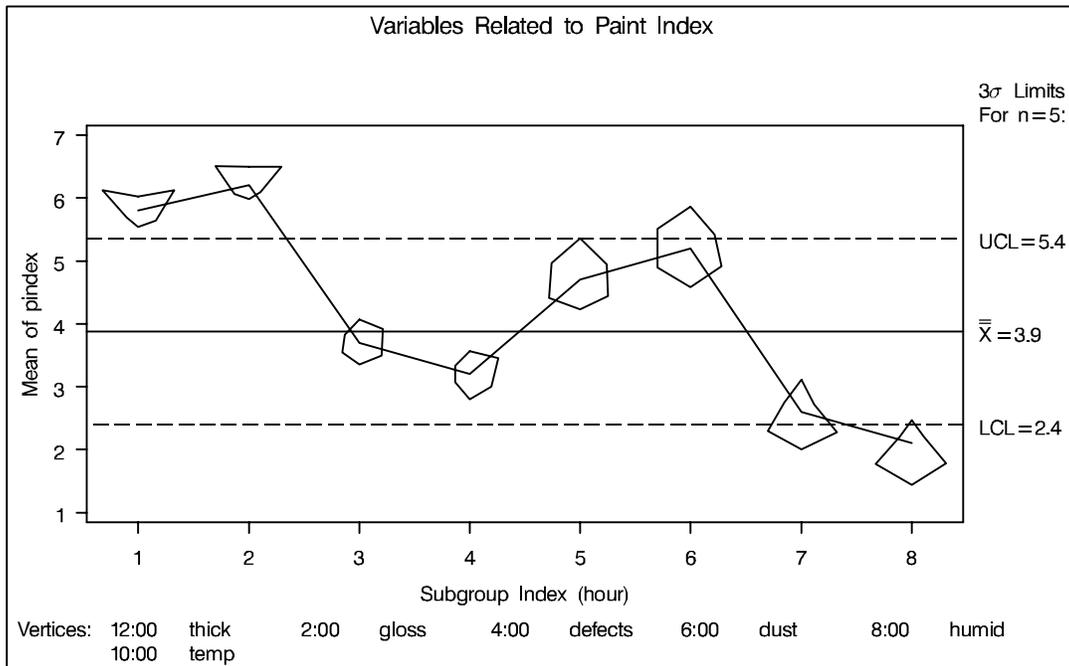


Figure 47.16. A Basic Star Chart

## Adding Reference Circles to Stars

You can add reference circles to a star chart to represent limits of variation for the vertex variables. The following statements add two special reference circles, called the *inner circle* and the *outer circle*, to the star chart in Figure 47.16:

See SHWSTR1  
in the SAS/QC  
Sample Library

```

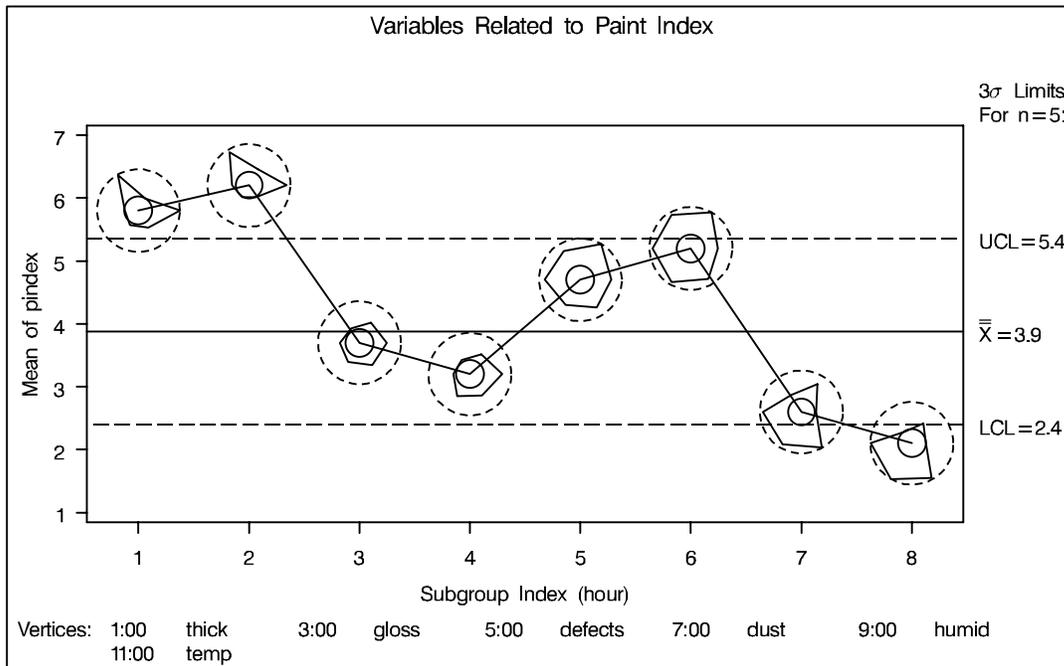
title 'Variables Related to Paint Index';

proc shewhart history=paint limits=paintlim;
  xchart pindex*hour /
    nolegend
    starvertices = (thick gloss defects dust humid temp)
    starcircles = 0.0 1.0
    lstarcircles = 1 2
    starstart = '1:00'T ;
run;

```

The star chart shown in Figure 47.17 displays the two reference circles centered about each point. The `STARCIRCLES=` value 0.0 requests the *inner circle*, and the value 1.0 requests the *outer circle*. Whether or not they are displayed, these circles are always associated with each star.

The interpretation of the inner and outer circles depends on the method used to standardize the vertex variables. By default (as in this example), the data for each vertex variable are standardized by the range of the variable values taken across subgroups. That is, the inner circle represents the minimum value, and the outer circle represents the maximum value. You can specify other methods of standardization (see “Specifying the Method of Standardization” on page 1708).



**Figure 47.17.** Star Chart with Inner and Outer Circles Added

Note that the STARCIRCLES= option does not specify the physical radius of a reference circle. Instead, this option specifies the radius relative to the radii of the inner and outer circles. Thus, specifying STARCIRCLES=0.0 always displays the inner circle, and specifying STARCIRCLES=1.0 always displays the outer circle. Specifying STARCIRCLES=0.5 displays a reference circle halfway between the inner and outer circles. You can specify the physical radii (in percent screen units) of the inner and outer circles using the STARINRADIUS= and STAROUTRADIUS= options. In the preceding statements, the LSTARCIRCLES= option specifies line types (1=solid and 2=dashed) for the inner and outer circles. You can also use the WSTARCIRCLES= option to control the thickness of the circles.

The STARSTART= option gives the starting position for the first vertex variable listed. In the preceding example, this option specifies that the vertex corresponding to (THICK) is to be positioned at one o'clock. The remaining vertices are uniformly spaced clockwise and correspond to the vertex variables in the order listed with the STARVERTICES= option.

For more information about the star options, see the appropriate entries in Chapter 46, "Dictionary of Options."

## Specifying the Style of Stars

The following statements create star charts for paint index using different styles for the stars specified with the `STARTYPE=` option:

See SHWSTR2  
in the SAS/QC  
Sample Library

```

title 'Variables Related to Paint Index';

proc shewhart history=paint limits=paintlim;

    xchart pindex * hour /
        nolegend
        starvertices = ( thick gloss defects dust humid temp )
        starstart    = '1:00'T
        startype     = wedge ;

    xchart pindex * hour /
        nolegend
        starvertices = ( thick gloss defects dust humid temp )
        starstart    = '1:00'T
        startype     = radial ;

    xchart pindex * hour /
        nolegend
        starvertices = ( thick gloss defects dust humid temp )
        starstart    = '1:00'T
        startype     = spoke ;

    xchart pindex * hour /
        nolegend
        starvertices = ( thick gloss defects dust humid temp )
        starstart    = '1:00'T
        startype     = corona ;

run;

```

The charts are shown in Figure 47.18, Figure 47.19, Figure 47.20, and Figure 47.21. The default style for the stars is `STARTYPE=POLYGON`, which is illustrated in Figure 47.16 and Figure 47.17. For more information, see the entry for the `STARTYPE=` option in Chapter 46, “Dictionary of Options.”

Part 9. The CAPABILITY Procedure

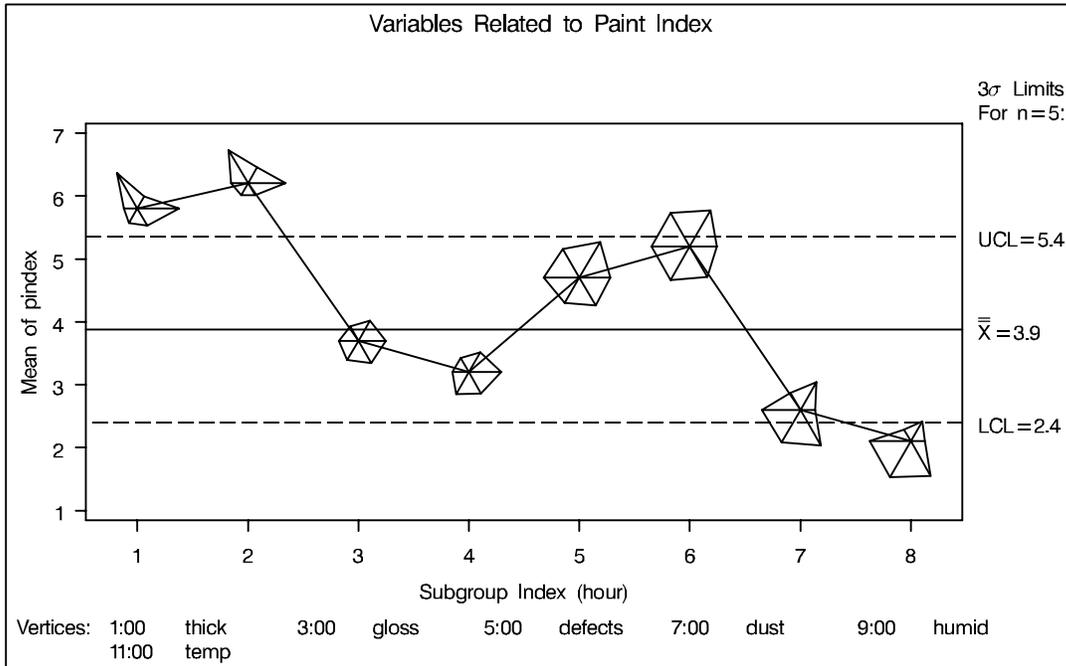


Figure 47.18. Star Chart Using STARTYPE=WEDGE

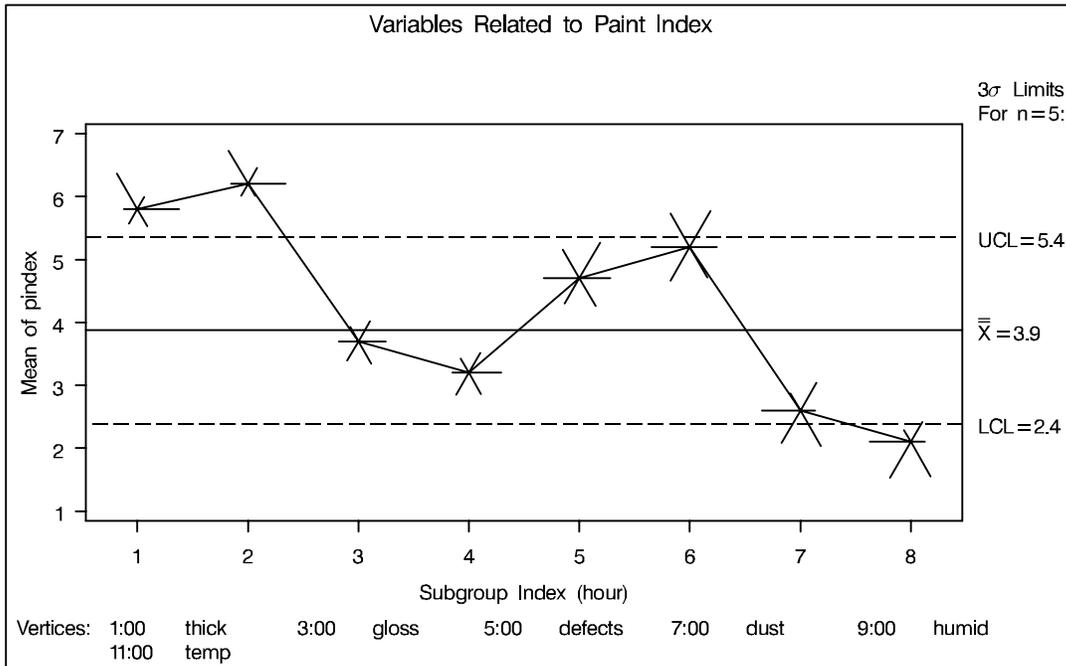


Figure 47.19. Star Chart Using STARTYPE=RADIAL

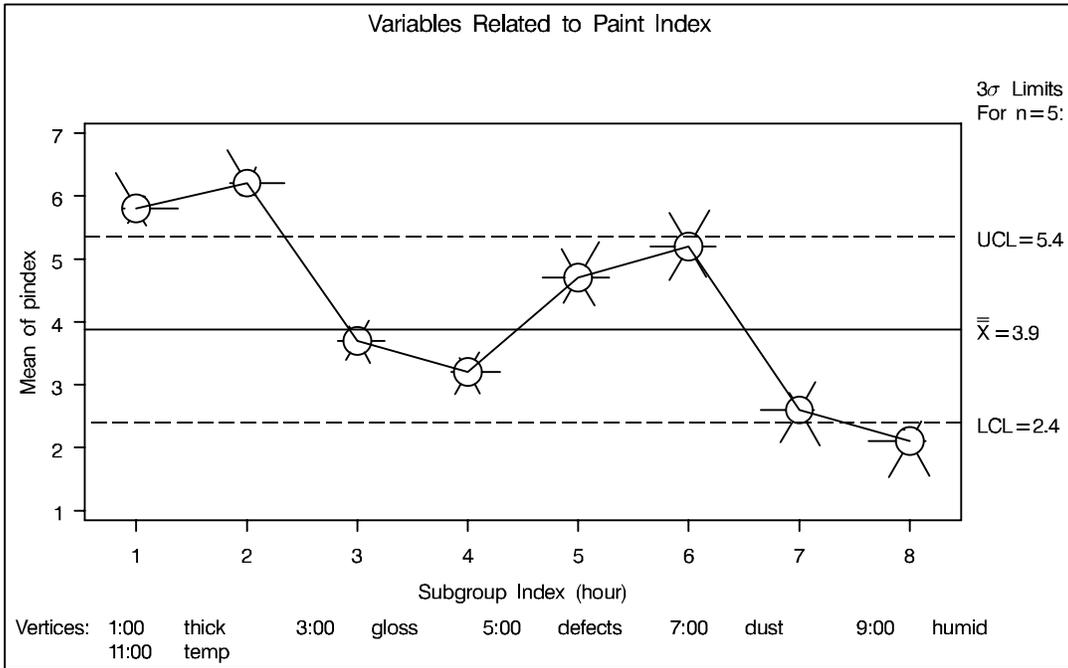


Figure 47.20. Star Chart Using STARTYPE=SPOKE

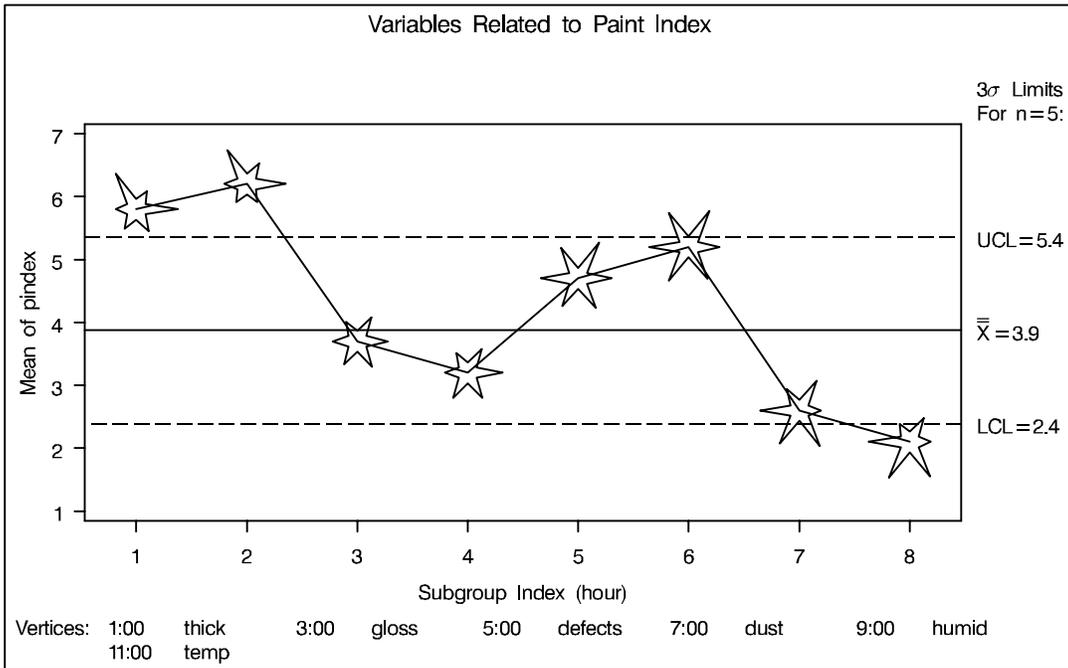


Figure 47.21. Star Chart Using STARTYPE=CORONA

## Specifying the Method of Standardization

See SHWSTR3  
in the SAS/QC  
Sample Library

In the previous examples in this section, the default method of standardization (based on ranges) is used for all six vertex variables. You can specify alternative methods with the STARSPECS= option. For example, specifying STARSPECS=3 standardizes each vertex variable so that the inner circle corresponds to three standard deviations below the mean and the outer circle corresponds to three standard deviations above the mean (that is, the circles represent  $3\sigma$  limits). Specifying STARSPECS= $k$  requests circles corresponding to  $k\sigma$  limits, and specifying STARSPECS=0 requests the default method.

In some applications, it may be necessary to use distinct methods of standardization for the vertex variables. You can do this by creating an input SAS data set that provides the method for each vertex variable and specifying this data set with the STARSPECS= option.

The following statements create a data set named MYSPECS that specifies standardization methods for the vertex variables used in the previous examples:

```
data myspecs;
  length _var_      $8
         _label_   $16 ;
  input  _var_ _label_ _lspoke_ _sigmas_ _lsl_ _usl_ ;
  datalines;
    thick  Thickness  1      .      0.25  0.50
    gloss  Gloss      1      .      0.10  0.60
    defects Defects   1      .      0.10  0.60
    dust   Dust       2      3.0    .      .
    humid  Humidity   2      0.0    .      .
    temp   Temperature 2      0.0    .      .
  ;
```

This data set contains a number of special variables whose names begin and end with an underscore.

Variable Name	Description
_LABEL_	label for identifying the vertex (used in conjunction with the STARLABEL= option). This must be a character variable of length 16 or less.
_LSL_	lower specification limit
_LSPOKE_	line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
_SIGMAS_	multiple of standard deviations above and below the average. A value of zero specifies standardization based on the range.
_USL_	upper specification limit
_VAR_	name of vertex variable. This must be a character variable of length 8.

Standardization is specified with the variables `_SIGMAS_`, `_LSL_`, and `_USL_`, as follows:

- Since nonmissing specification limits (`_LSL_` and `_USL_`) are provided for the variables `THICK`, `GLOSS`, and `DEFECTS`, the values of these variables are scaled so that the inner circle represents the lower specification limit and the outer circle represents the upper specification limit.
- Since `_SIGMAS_` is equal to 3 for `DUST` (and since both `_LSL_` and `_USL_` are missing), values of `DUST` are scaled so that the inner circle represents three standard deviations below the mean, and the outer circle represents three standard deviations above the mean. The mean and standard deviation are calculated across all subgroups.
- Since `_SIGMAS_` is equal to 0 for `HUMID` and `TEMP` (and since both `_LSL_` and `_USL_` are missing), values of `HUMID` and `TEMP` are scaled so that the inner circle represents the minimum and the outer circle represents the maximum. The minimum and maximum are calculated across all subgroups.

The following statements use the data set `MYSPECS` to create a star chart for paint index:

```

title 'Variables Related to Paint Index';

proc shewhart history=paint limits=paintlim;
  xchart pindex * hour /
    nolegend
    starvertices = ( thick gloss defects dust humid temp )
    startype      = wedge
    starcircles   = 0.0 1.0
    lstarcircles  = 2 2
    starstart     = -30
    labelfont     = simplex
    starlegend    = degrees
    starspecs     = myspecs
    starlabel     = high ;
run;

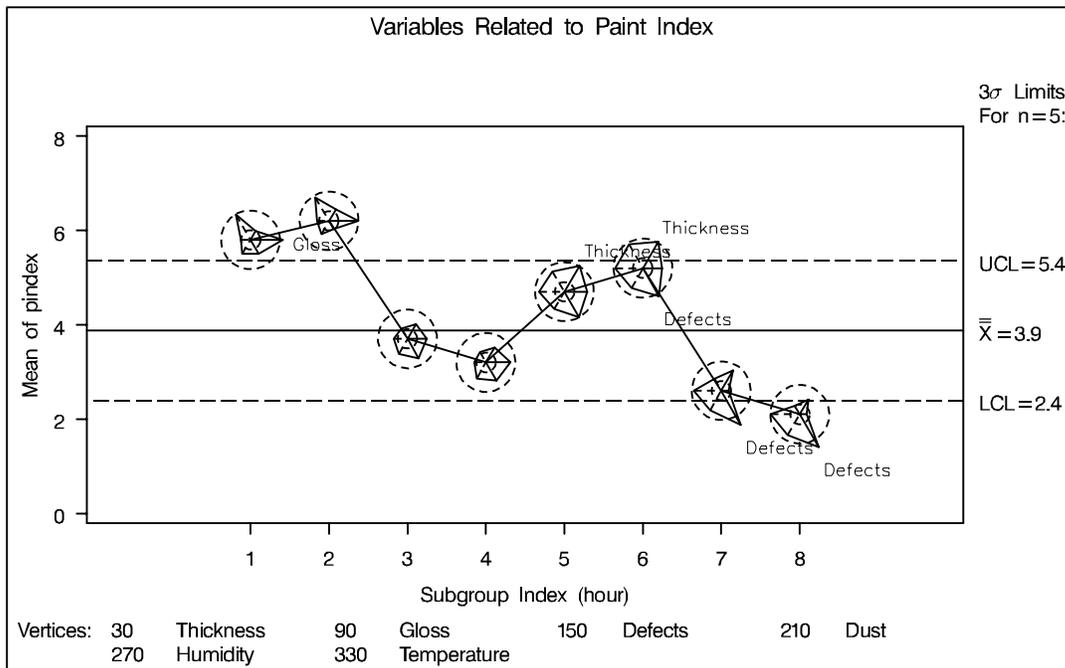
```

The chart is shown in Figure 47.22. Specifying `STARLEGEND=DEGREES` requests a legend that identifies the vertex variables by their angles (in degrees) rather than their clock positions. Here, zero degrees corresponds to twelve o'clock, and the degrees are measured clockwise. The first vertex variable is positioned at 30 degrees, as specified with the `STARSTART=` option. Note that you specify the `STARSTART=` value as a negative number to indicate that it is in degrees.

In Figure 47.22 the vertices that exceed the outer circle are labeled with the value of the variable `_LABEL_` in the `STARSPECS=` data set. This type of labeling is requested by specifying `STARLABEL=HIGH`. A font (`SIMPLEX`) for the labels is specified with the `LABELFONT=` option.

The vertices for THICK at HOUR=5, 6, and 7 are truncated, as indicated in the SAS log. The truncation value is the physical radius of an imaginary circle referred to as the *bounding circle* that lies outside the outer circle. In general, any vertex that exceeds the bounding circle is truncated to the *bounding radius*. This is done so that unusually large vertex variable values will not result in grossly distorted stars. You can specify a different bounding radius with the STARBDRADIUS= option.

The spokes corresponding to the environmental variables DUST, HUMID, and TEMP are drawn with a dashed line style to distinguish them from the quality variables THICK, GLOSS, and DEFECTS, whose spokes are drawn with a solid line. The styles are specified by the variable `_LSPOKE_`. Refer to *SAS/GRAPH Software: Reference* for a complete list of line styles. If you are producing charts in color, you can also use the variable `_CSPOKE_` in the STARSPECS= data set to assign colors to the spokes.



**Figure 47.22.** Star Chart Using STARSPECS= Specifications

For more information about the options used in this example, see the appropriate entries in Chapter 46, "Dictionary of Options."

## Displaying Trends in Process Data

Time trends due to tool wear, environmental changes, and other gradual process changes are sometimes observed in  $\bar{X}$  charts. The presence of a systematic trend makes it difficult to interpret the chart because the control limits are designed to indicate expected variation strictly due to common causes.

See SHWTREN  
in the SAS/QC  
Sample Library

You can use the REG procedure (or other modeling procedure) in conjunction with the SHEWHART procedure to determine whether a process with a time trend is in control. With the REG procedure, you can model the trend and save the fitted subgroup means ( $\hat{X}_t$ ) and the residual subgroup means ( $\bar{X}_t - \hat{X}_t$ ) in an output data set. Then, using this data as input to the SHEWHART procedure, you can create a *trend chart*, which displays a trend plot of the fitted subgroup means together with an  $\bar{X}$  chart for the residual subgroup means, thus removing the time-dependent component of the data from its random component. Having accounted for the time trend, you can decide whether the process is in control by examining the  $\bar{X}$  chart.

The following example illustrates the steps used to create a trend chart for a SAS data set named TOOLWEAR that contains diameter measurements for 20 subgroup samples each consisting of eight parts:

```

data toolwear;
  input hour @;
  do i=1 to 8;
    input diameter @;
    output;
  end;
  label hour      = 'Hour'
        diameter = 'Mean Diameter in mm';
  datalines;
1  10.04  9.94  9.95  9.81  10.08  10.03  10.12  10.02
2  10.20  9.97  10.04  10.12  10.10  10.16  10.14  10.13
3  10.06  10.07  10.25  10.18  10.13  10.12  10.14  10.21
4  10.16  10.14  10.24  10.26  10.18  10.16  10.10  10.00
5   9.96  10.43  10.11  10.27  10.03  10.14  10.27  10.06
6  10.22  10.23  10.24  10.23  10.20  10.28  10.33  10.17
7  10.27  10.38  10.25  10.40  10.33  10.13  10.20  10.18
8  10.25  10.20  10.29  10.43  10.32  10.31  10.20  10.32
9  10.25  10.19  10.37  10.18  10.32  10.24  10.25  10.34
10 10.37  10.52  10.29  10.31  10.51  10.32  10.31  10.16
11 10.35  10.32  10.46  10.51  10.37  10.12  10.39  10.21
12 10.33  10.33  10.19  10.28  10.24  10.36  10.29  10.27
13 10.17  10.24  10.18  10.23  10.31  10.34  10.33  10.38
14 10.36  10.46  10.54  10.48  10.54  10.46  10.37  10.18
15 10.30  10.42  10.36  10.47  10.35  10.45  10.46  10.40
16 10.60  10.19  10.34  10.32  10.34  10.35  10.38  10.31
17 10.40  10.36  10.32  10.41  10.48  10.48  10.22  10.39
18 10.45  10.45  10.34  10.45  10.42  10.35  10.39  10.16
19 10.34  10.33  10.45  10.47  10.40  10.27  10.30  10.63
20 10.26  10.38  10.42  10.31  10.57  10.42  10.41  10.38
;

```

## Step 1: Preliminary Mean and Standard Deviation Charts

The following statements create  $\bar{X}$  and  $s$  charts for the diameter data:

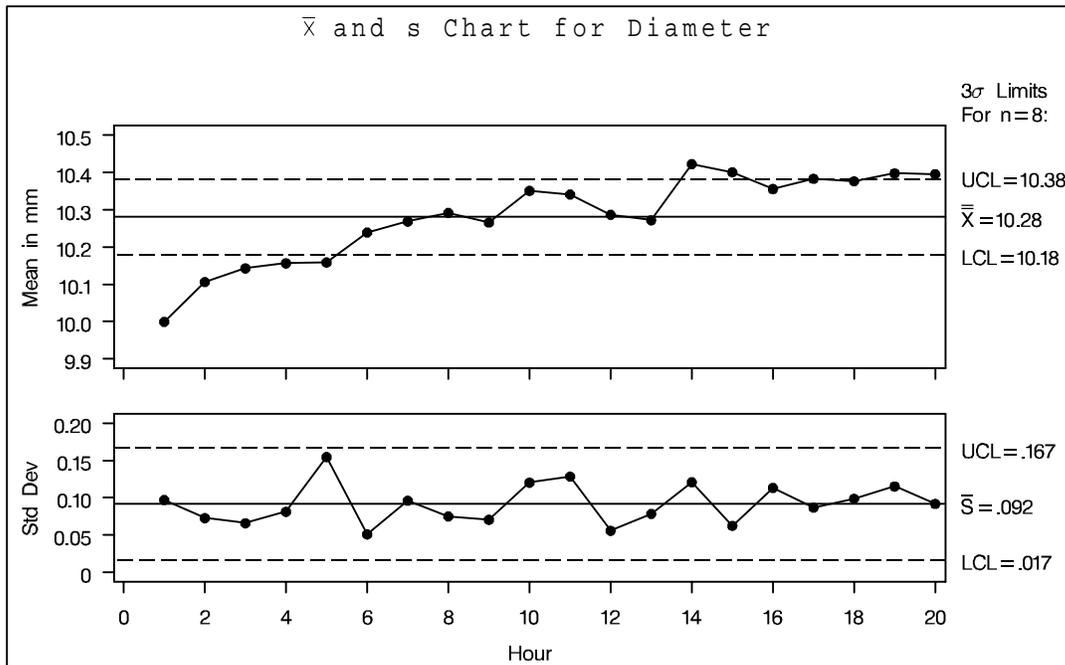
```

symbol value=dot;
title font=qcfont4 'X'
      font=swiss  ' and s Charts for Diameter';

proc shewhart data=toolwear;
  xschart diameter*hour /
    outhistory = submeans
    nolegend ;
run;

```

The charts are shown in Figure 47.23. The subgroup standard deviations are all within their control limits, indicating the process variability is stable. However, the  $\bar{X}$  chart displays a nonlinear trend that makes it difficult to decide if the process is in control. Subsequent investigation reveals that the trend is due to tool wear.



**Figure 47.23.**  $\bar{X}$  and  $s$  Charts for TOOLWEAR Data

Note that the symbol  $\bar{X}$  is displayed in the title with the special font QCFONT4, which matches the SWISS font used for the remainder of the title. See Appendix D, “Special Fonts in SAS/QC Software” for a description of the fonts available for displaying  $\bar{X}$  and related symbols.

## Step 2: Modeling the Trend

The next step is to model the trend as a function of hour. The  $\bar{X}$  chart in Figure 47.23 suggests that the mean level of the process (saved as DIAMTERX in the OUTLIM-ITS= data set SUBMEANS) grows as the log of HOUR. The following statements fit a simple linear regression model in which DIAMTERX is the response variable and LOGHOUR (the log transformation of HOUR) is the predictor variable. Part of the printed output produced by PROC REG is shown in Figure 47.24.

```
data submeans;
  set submeans;
  loghour=log(hour);

proc reg data=submeans ;
  model diamterx=loghour;
  output out=regdata predicted=fitted ;

proc print data=regdata noobs;
run;
```

Model: MODEL1						
Dependent Variable: diameterX Mean of diameter						
Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	Intercept	1	9.99056	0.02185	457.29	<.0001
loghour		1	0.13690	0.00967	14.16	<.0001

**Figure 47.24.** Trend Analysis for DIAMETER from PROC REG

Figure 47.24 shows that the fitted equation can be expressed as

$$\hat{X}_t = 9.99 + 0.14 \times \log(t)$$

where  $\hat{X}_t$  is the fitted subgroup average.\* A partial listing of the OUT= data set REGDATA created by the REG procedure is shown in Figure 47.25.

hour	diameterX	diameterS	diameterN	loghour	fitted
1	9.9992	0.09726	8	0.00000	9.9906
2	10.1060	0.07290	8	0.69315	10.0855
3	10.1428	0.06601	8	1.09861	10.1410
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
20	10.3950	0.09185	8	2.99573	10.4007

**Figure 47.25.** Listing of the Output Data Set REGDATA from the REG Procedure

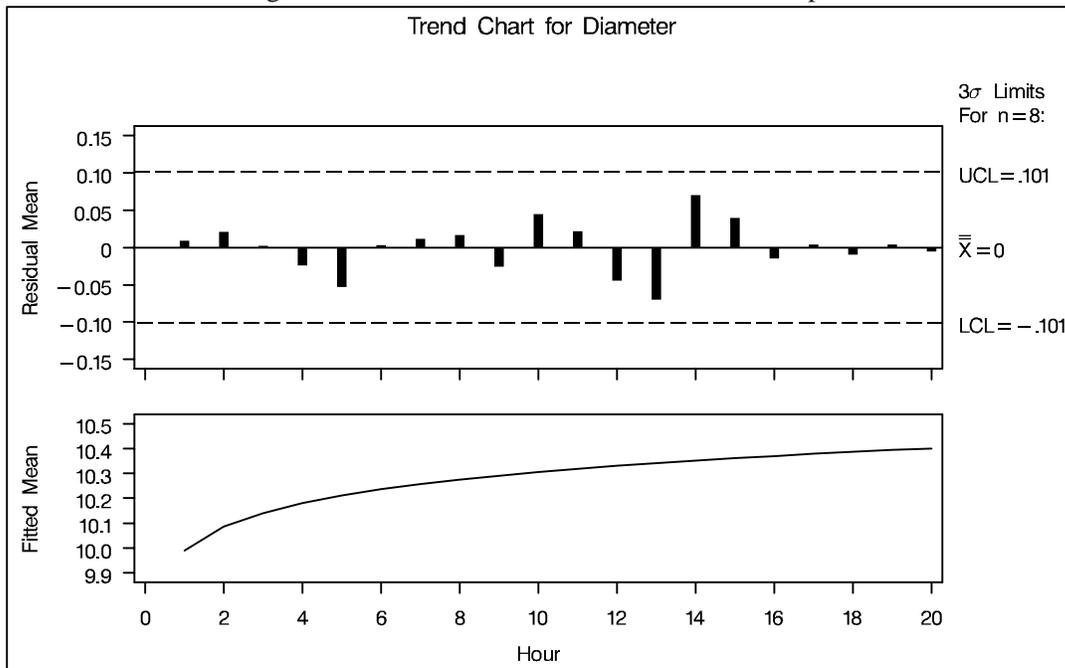
\*Although this example does not check for the existence of a trend, you should do so by using the hypothesis tests provided by the REG procedure.

### Step 3: Displaying the Trend Chart

The third step is to create a trend chart with the SHEWHART procedure, as follows:

```
symbol value=none width=5;
title 'Trend Chart for Diameter';
proc shewhart history=regdata;
  xchart diameter*hour /
    stddeviations
    trendvar = fitted
    cneedles = black
    wtrend   = 1
    split='/'
    nolegend ;
  label diamterx='Residual Mean/Fitted Mean';
run;
```

The chart is shown in Figure 47.26. The values of FITTED are plotted in the lower half of the trend chart. The upper half of the trend chart is an  $\bar{X}$  chart for the residual means (DIAMTERX – FITTED). The CNEEDLES= option specifies that the residuals are to be represented by vertical bars as deviations from the central line. The  $\bar{X}$  chart in Figure 47.26 shows that, after accounting for the trend, the mean level of the process is in control.



**Figure 47.26.** Trend Chart for Diameter Data

If the data are correlated in time, you can use the ARIMA or AUTOREG procedures in place of the REG procedure to remove autocorrelation structure and display a control chart for the residuals; for an example, see “Autocorrelation in Process Data” on page 1756. Another application of the TRENDVAR= option is the display of nominal values in control charts for short runs; see “Short Run Process Control” on page 1770.

## Clipping Extreme Points

In some control chart applications, the out-of-control points can be so extreme that the remaining points are compressed to a scale that is difficult to read. In such cases, you can clip the extreme points so that a more readable chart is displayed, as illustrated in the following example.

See SHWCLIP  
in the SAS/QC  
Sample Library

A company producing copper tubing uses  $\bar{X}$  and  $R$  charts to monitor the diameter of the tubes. Based on previous production, known values of 70mm and 0.75mm are available for the mean and standard deviation of the diameter. The diameter measurements (in millimeters) for 15 batches of five tubes each are provided in the data set NEWTUBES.

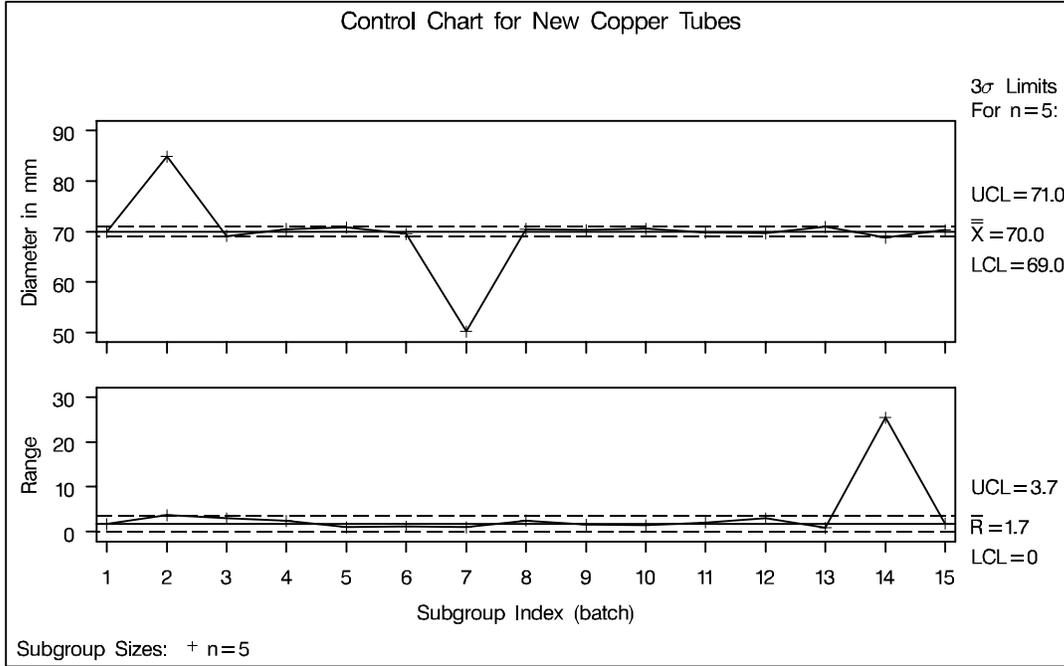
```
data newtubes;
  label diameter='Diameter in mm';
  do batch = 1 to 15;
    do i = 1 to 5;
      input diameter @@;
      output;
    end;
  end;
datalines;
69.13 69.83 70.76 69.13 70.81
85.06 82.82 84.79 84.89 86.53
67.67 70.37 68.80 70.65 68.20
71.71 70.46 71.43 69.53 69.28
71.04 71.04 70.29 70.51 71.29
69.01 68.87 69.87 70.05 69.85
50.72 50.49 49.78 50.49 49.69
69.28 71.80 69.80 70.99 70.50
70.76 69.19 70.51 70.59 70.40
70.16 70.07 71.52 70.72 70.31
68.67 70.54 69.50 69.79 70.76
68.78 68.55 69.72 69.62 71.53
70.61 70.75 70.90 71.01 71.53
74.62 56.95 72.29 82.41 57.64
70.54 69.82 70.71 71.05 69.24
;
```

The following statements create the  $\bar{X}$  and  $R$  charts shown in Figure 47.27 for the tube diameter:

```
symbol value=plus;
title 'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
  xrchart diameter*batch /
    mu0 = 70
    sigma0 = 0.75 ;
run;
```

Part 9. The CAPABILITY Procedure

Batches 2 and 7 result in extreme out-of-control points on the mean chart, and batch 14 results in an extreme out-of-control point on the range chart. The vertical axes are scaled to accommodate these extreme out-of-control points, and this in turn forces the control limits to be compressed.



**Figure 47.27.**  $\bar{X}$  and  $R$  Charts Without Clipping

You can request clipping by specifying the option `CLIPFACTOR=factor`, where *factor* is a value greater than one (useful values are typically in the range 1.5 to 2). Clipping is applied in two steps, as follows:

1. If a plotted statistic is greater than  $y_{\max}$ , it is temporarily set to  $y_{\max}$ , where

$$y_{\max} = LCL + (UCL - LCL) \times factor$$

- If a plotted statistic is less than  $y_{\min}$ , it is temporarily set to  $y_{\min}$ , where

$$y_{\min} = UCL - (UCL - LCL) \times factor$$

2. Axis scaling is applied to the clipped statistics. Then the  $y_{\max}$  values are reset to the maximum value on the axis and the  $y_{\min}$  values are reset to the minimum value on the axis.

Notes:

- Clipping is applied only to the plotted statistics and not to the statistics tabulated or saved in an output data set.
- Because the *factor* must be greater than one, clipping does not affect whether a plotted statistic is inside or outside the control limits.
- Tests for special causes are applied to the plotted statistics before they are clipped, and clipping does not affect how the tests are flagged on the chart. In some situations, however, clipping can make the patterns associated with the tests less evident on the chart.
- When primary and secondary charts are displayed, the same clipping *factor* is applied to both charts.
- A special symbol is used for clipped points (the default symbol is a square), and a legend is added to the chart indicating the number of points that were clipped.

The following statements create  $\bar{X}$  and  $R$  charts, shown in Figure 47.28, that use a clipping factor of 1.5:

```
proc shewhart data=newtubes;
  xrchart diameter*batch /
    mu0      = 70
    sigma0   = 0.75
    clipfactor = 1.5 ;
run;
```

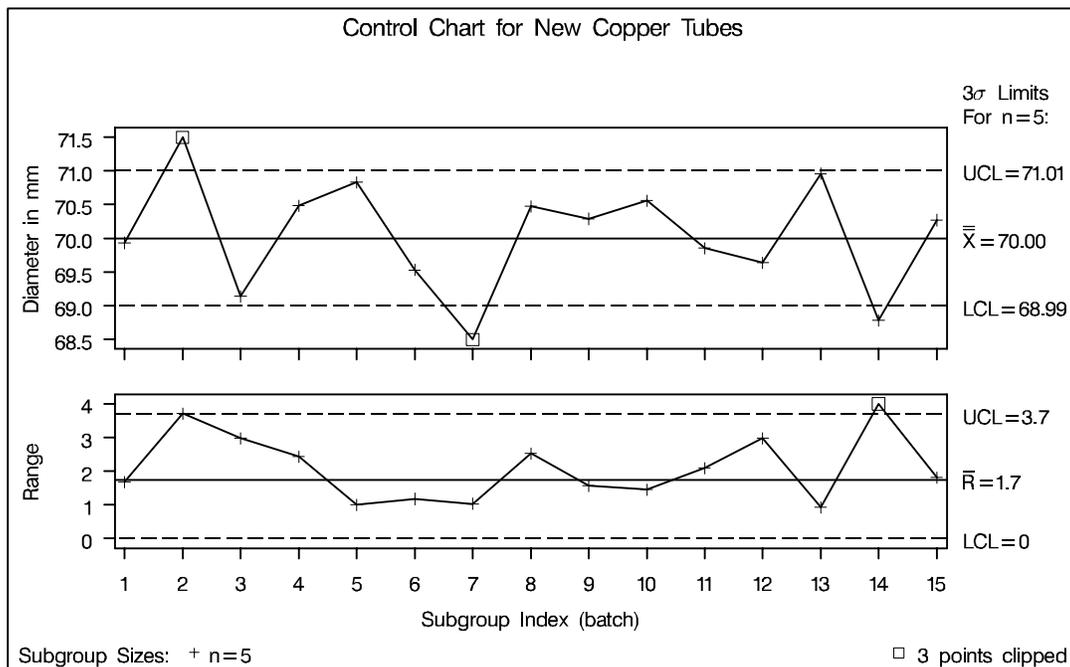


Figure 47.28.  $\bar{X}$  and  $R$  Charts with Clip Factor of 1.5

Part 9. The CAPABILITY Procedure

In Figure 47.28, the extreme out-of-control points are clipped making the points plotted within the control limits more readable. The clipped points are marked with a square, and a clipping legend is added at the lower right of the display.

Other clipping options are available, as illustrated by the following statements:

```
proc shewhart data=newtubes;
  xrchart diameter*batch /
    mu0          = 70
    sigma0       = 0.75
    clipfactor   = 1.5
    clipsymbol   = dot
    cliplegpos   = top
    cliplegend   = '# Clipped Points'
    clipsubchar  = '#';
run;
```

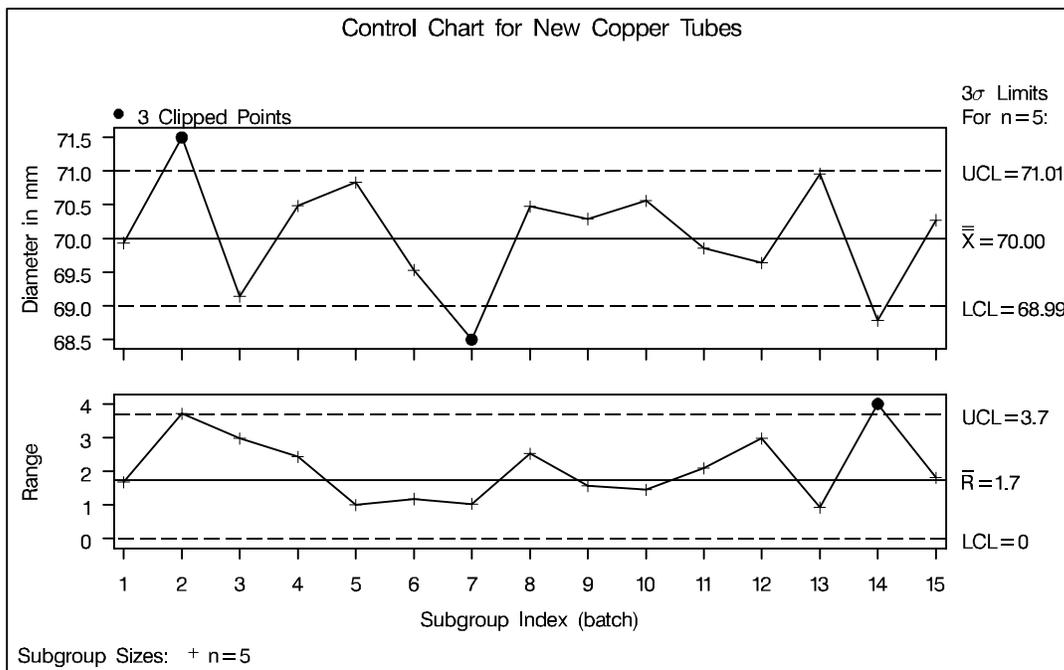


Figure 47.29.  $\bar{X}$  and  $R$  Charts Using Clipping Options

Specifying CLIPSYMBOL=DOT marks the clipped points with a dot instead of the default square. Specifying CLIPLEGPOS=TOP positions the clipping legend at the top of the chart. The options CLIPLEGEND='# Clipped Points' and CLIPSUBCHAR='#' request the clipping legend 3 *Clipped Points*. For more information about the clipping options, see the appropriate entries in Chapter 46, "Dictionary of Options."

## Labeling Axes

The SHEWHART procedure provides default labels for the horizontal and vertical axes of control charts. You can specify axis labels by assigning labels to variables, as discussed in the following sections.

See SHWLAB  
in the SAS/QC  
Sample Library

### Default Labels

If a label is not associated with the *subgroup-variable*, the default horizontal axis label is “Subgroup Index (*subgroup-variable*).” The default vertical axis label for a primary chart identifies the chart type and the process variable. The default vertical axis label for a secondary chart identifies the chart type only.

For example, the following statements create  $\bar{X}$  and *s* charts with default labels using the data set PARTS given in “Displaying Stratified Process Data” on page 1682. The resulting charts are displayed in Figure 47.30.

```

title 'Control Chart for Diameter';
proc shewhart history=parts;
  xschart diam*sample;
run;

```

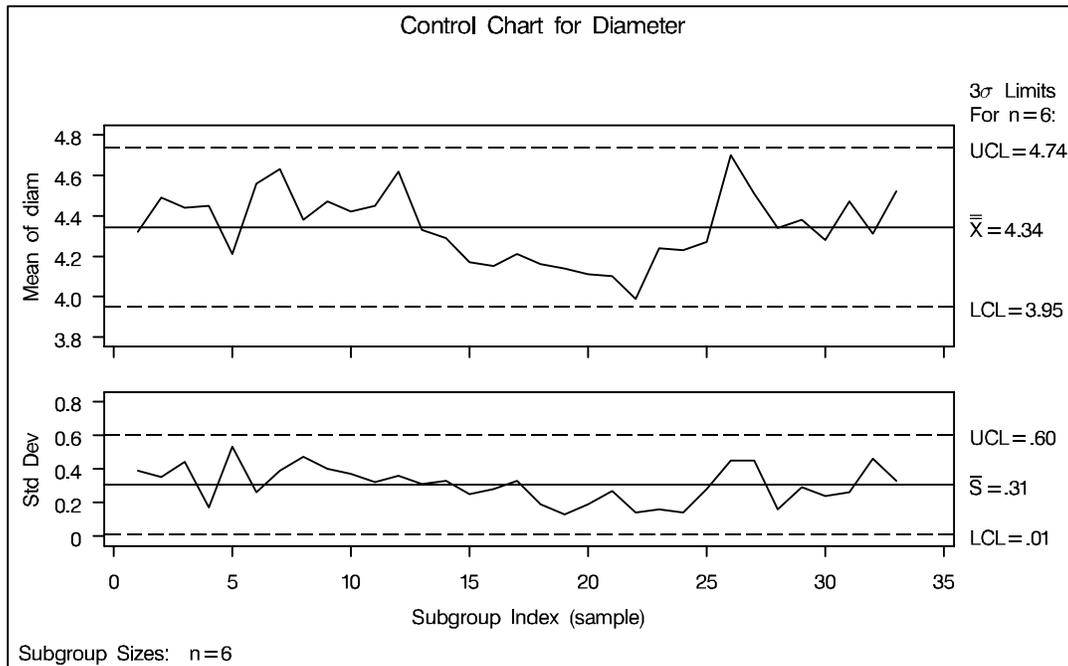


Figure 47.30. Control Charts with Default Labels

---

## Labeling the Horizontal Axis

You can specify a label of up to 40 characters for the horizontal axis by assigning the label to the *subgroup variable* with a LABEL statement (refer to *SAS Language Reference: Dictionary* for a description of LABEL statements). If you use a LABEL statement after the PROC SHEWHART statement and before the RUN statement, the label is associated with the variable only for the duration of the PROC step.

For an example, see page 1722, where Figure 47.31 redisplayes the  $\bar{X}$  and *s* charts in Figure 47.30 with specified horizontal and vertical axis labels.

---

## Labeling the Vertical Axis

You can specify a label for the vertical axis of a primary chart by using a LABEL statement to assign the label to a particular variable in the input data set. The type of input data set, the chart statement, and the *process* specified in the chart statement determine which variable to use in the LABEL statement.

- If the input data set is a DATA= data set, assign the label to the process variable (*process*) specified in the chart statement.
- If the input data set is a HISTORY= data set, assign the label to the variable specified in the chart statement whose name begins with the prefix *process* and ends with the appropriate suffix given by the following list:

Chart Statement	Suffix
BOXCHART with CONTROLSTAT=MEAN	X
BOXCHART with CONTROLSTAT=MEDIAN	M
CCHART	U
IRCHART	none
MCHART	M
MRCHART	M
NPCHART	P
PCHART	P
RCHART	R
SCHART	S
UCHART	U
XCHART	X
XRCHART	X
XSCHART	X

If the prefix *process* consists of 8 characters, shorten the prefix to its first 4 characters and last 3 characters before adding the suffix.

- If the input data set is a TABLE= data set, assign the label to the predefined variable given by the following table:

Chart Statement	Variable
BOXCHART with CONTROLSTAT=MEAN	_SUBX_
BOXCHART with CONTROLSTAT=MEDIAN	_SUBMED_
CCHART	_SUBC_
IRCHART	_SUBI_
MCHART	_SUBMED_
MRCHART	_SUBMED_
NPCHART	_SUBNP_
PCHART	_SUBP_
RCHART	_SUBR_
SCHART	_SUBS_
UCHART	_SUBU_
XCHART	_SUBX_
XRCHART	_SUBX_
XSCHART	_SUBX_

If the chart statement produces primary and secondary charts, as in the case of the XSCHART statement, you can break the label into two parts by including a split character in the label. The part before the split character labels the vertical axis of the primary chart, and the part after the split character labels the vertical axis of the secondary chart. To specify the split character, use the SPLIT= option in the chart statement.

For example, the following statements redisplay the  $\bar{X}$  and  $s$  charts in Figure 47.30 with specified labels for the horizontal and vertical axes:

```

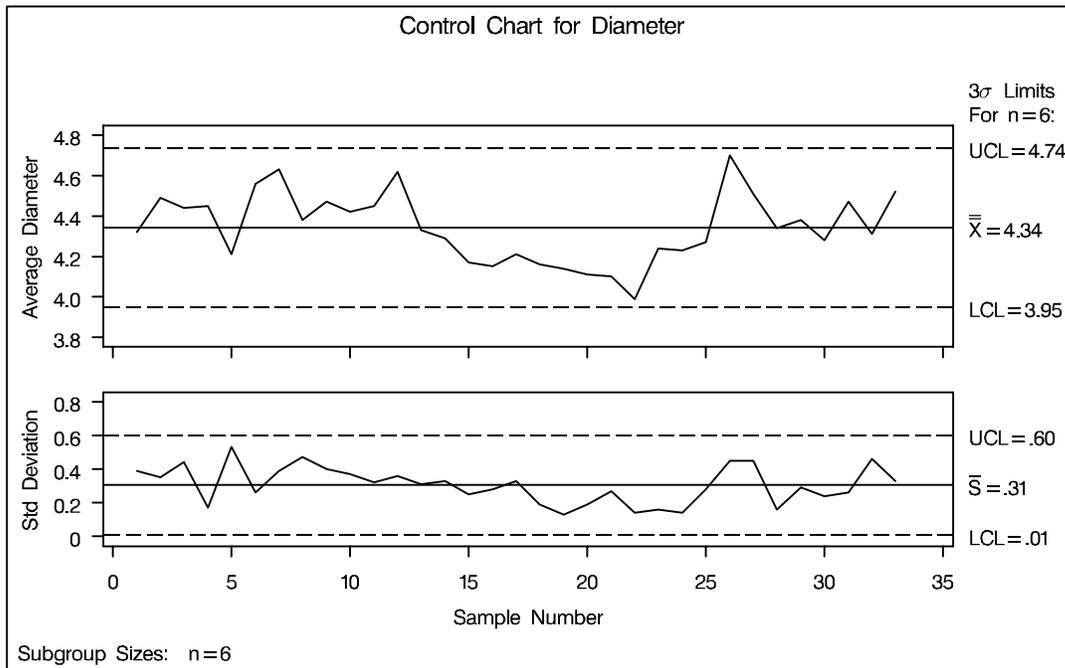
title 'Control Chart for Diameter';

proc shewhart history=parts;
  xschart diam*sample / split = '/' ;
  label sample = 'Sample Number'
    diamx = 'Average Diameter/Std Deviation';
run;

```

The charts are displayed in Figure 47.31. Because the input data set PARTS is a HISTORY= data set, the vertical axes are labeled by assigning a label to the subgroup mean variable DIAMX (that is, the *process* DIAM with the suffix X).<sup>\*</sup> Assigning a label to DIAM would result in an error message since DIAM is interpreted as a prefix rather than a SAS variable.

<sup>\*</sup>If the *process* were DIAMETER rather than DIAM, the label would be assigned to the variable DIAMTERX.



**Figure 47.31.** Control Charts with Axis Labels Specified

If the input data set were a DATA= data set rather than a HISTORY= data set, you would associate the label with the variable DIAM. If the input data set were a TABLE= data set, you would associate the label with the variable \_SUBX\_.

For another illustration, see Example 36.2 on page 1257.

---

## Selecting Subgroups for Computation and Display

This section describes methods for specifying which subgroups of observations in an input data set (DATA=, HISTORY=, or TABLE=) are to be used to compute control limits and which subgroups are to be displayed as points on the chart.

---

### Using WHERE Statements

The following statements create a data set named BOTTLES that records the number of cracked bottles encountered each day during two months (January and February) of a soft drink bottling operation:

See SHWWHR  
in the SAS/QC  
Sample Library

```

data bottles;
informat day date7.;
format day date7. ;
  nbottles = 3000;
  input day ncracks @@;
datalines;
04JAN94 61 05JAN94 56 06JAN94 71 07JAN94 56
10JAN94 51 11JAN94 64 12JAN94 71 13JAN94 91
14JAN94 98 17JAN94 68 18JAN94 63 19JAN94 60
20JAN94 58 21JAN94 55 24JAN94 78 25JAN94 47
26JAN94 54 27JAN94 69 28JAN94 73 31JAN94 66
01FEB94 57 02FEB94 55 03FEB94 63 04FEB94 50
07FEB94 69 08FEB94 54 09FEB94 64 10FEB94 66
11FEB94 70 14FEB94 49 15FEB94 57 16FEB94 56
17FEB94 59 18FEB94 66 21FEB94 60 22FEB94 58
23FEB94 67 24FEB94 60 25FEB94 62 28FEB94 48
;

```

The variable NBOTTLES contains the number of bottles sampled each day, and the variable NCRACKS contains the number of cracked bottles in each sample.

The following statements create a *p* chart for the number of cracked bottles based on the January production:

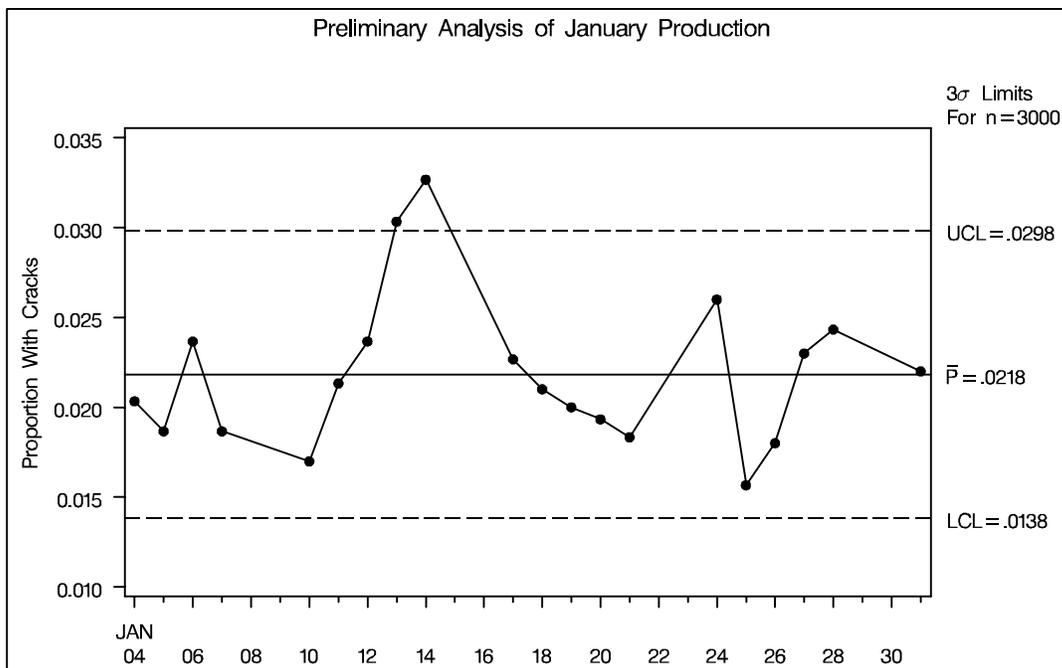
```

title 'Preliminary Analysis of January Production';
symbol v=dot;
proc shewhart data=bottles;
  where day <= '31JAN94'D;
  pchart ncracks * day / subgroupn = nbottles
          nohlabel
          nolegend
          outlimits = mylim;
  label ncracks = 'Proportion With Cracks';
run;

```

## Part 9. The CAPABILITY Procedure

The chart is shown in Figure 47.32. The WHERE statement restricts the observations read from BOTTLES so that the control limits are estimated from the January data, and only the January data are displayed on the chart. For details concerning the WHERE statement, refer to *SAS Language Reference: Dictionary*.



**Figure 47.32.** Preliminary  $p$  Chart for January Data

In Figure 47.32, a special cause of variation is signaled by the proportions for January 13 and January 14, which exceed the upper control limit. Since the cause, an improper machine setting, was corrected, it is appropriate to recompute the control limits by excluding the data for these two days. Again, this can be done with a WHERE statement, as follows:

```
title 'Final Analysis of January Production';
proc shewhart data=bottles;
  where ( day <= '31JAN94'D ) &
        ( day ne '13JAN94'D ) &
        ( day ne '14JAN94'D ) ;
  pchart ncracks * day / subgroupn = nbottles
        nohlabel
        nolegend
        outlimits = janlim;
  label ncracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 47.33.

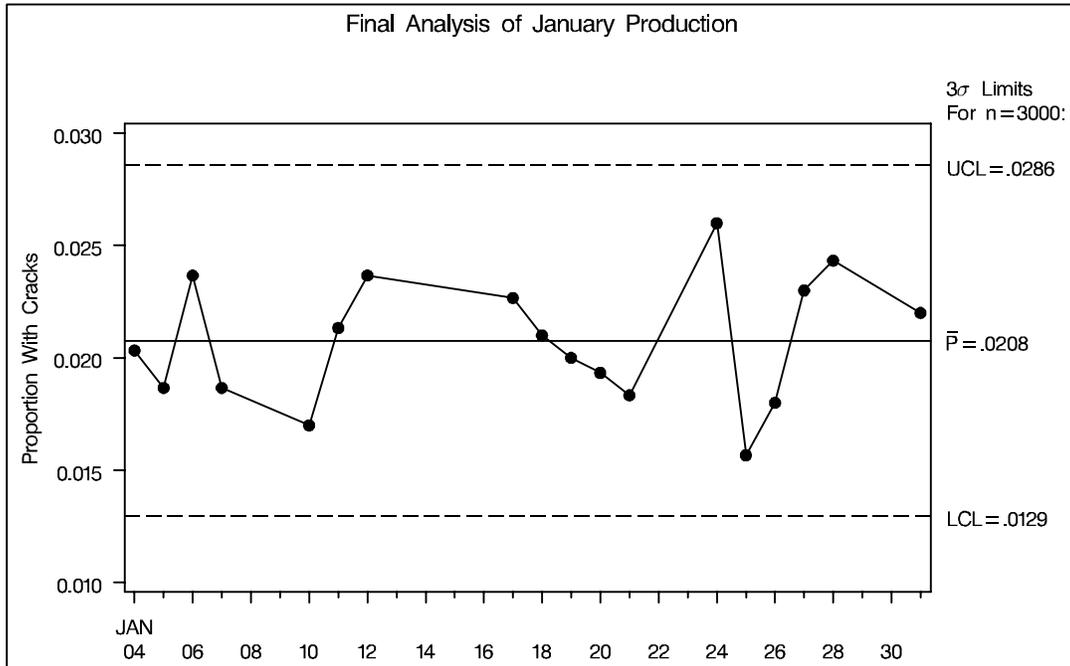


Figure 47.33. Final *p* Chart for January Data

The data set JANLIM, which saves the control limits, is listed in Figure 47.34.

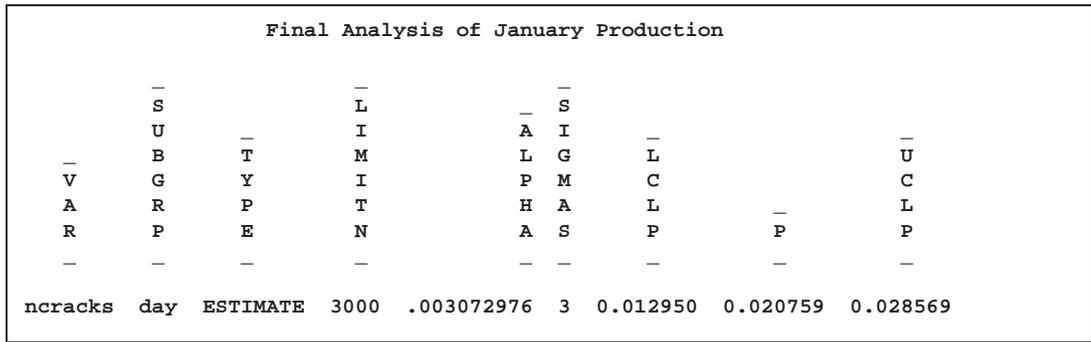


Figure 47.34. Listing of the LIMITS= Data Set JANLIM

Now, the control limits based on the January data are to be applied to the February data. Again, this can be done with a WHERE statement, as follows:\*

```

title 'Analysis of February Production';
proc shewhart data=bottles limits=janlim;
  where day > '31JAN94'D;
  pchart ncracks * day / subgroupn = nbottles
         nolegend
         nohlabel;
  label ncracks = 'Proportion With Cracks';
run;
    
```

\*In Release 6.09 and in earlier releases, it is also necessary to specify the READLIMITS option to read control limits from a LIMITS= data set.

The chart is shown in Figure 47.35.

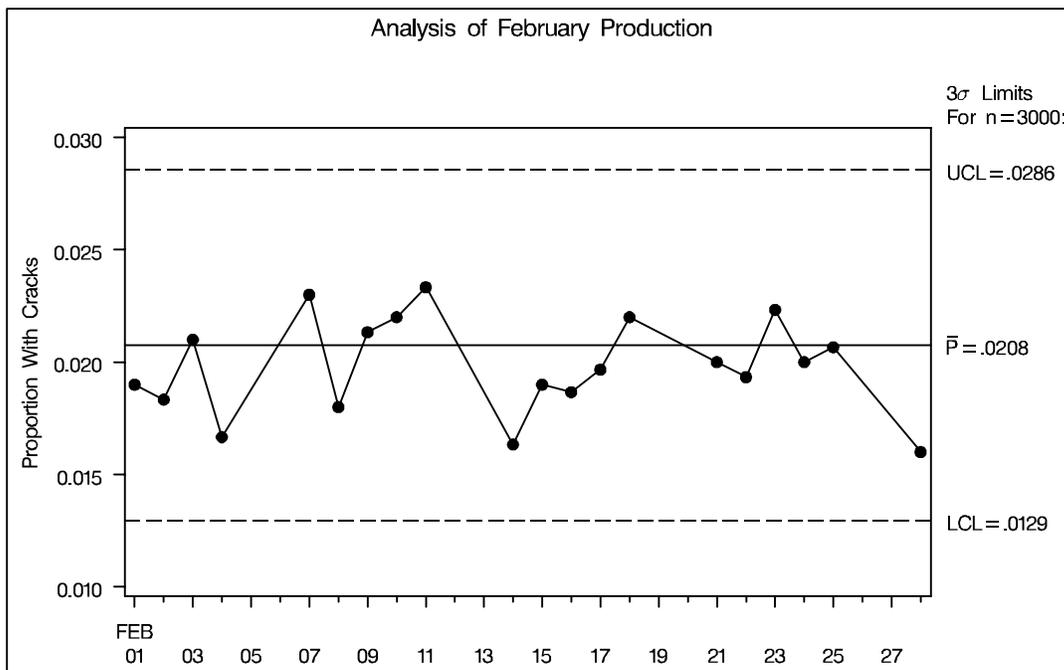


Figure 47.35. *p* Chart for February Data

## Using Switch Variables

See SHWSVAR  
in the SAS/QC  
Sample Library

As an alternative to reading a LIMITS= data set and using a WHERE statement, you can provide two special switch variables named `_COMP_` and `_DISP_` in the input data set. The rules for using these variables are as follows:

- Switch variables must be character variables of length one. Valid values for these variables are Y (or y) and N (or n). A blank value is treated as Y.
- Subgroups for which `_COMP_` is equal to Y are included in computations of parameter estimates and control limits, and observations for which `_COMP_` is equal to N are excluded.
- Subgroups for which `_DISP_` is equal to Y are displayed on the chart, and subgroups for which `_DISP_` is equal to N are not displayed.
- If the chart statement creates a chart for variables, you can provide two additional switch variables named `_COMP2_` and `_DISP2_`, which are defined similarly to `_COMP_` and `_DISP_`. In this case, the variable `_COMP_` specifies which subgroups are used to estimate the process mean  $\mu$ , and the variable `_COMP2_` specifies which subgroups are used to estimate the process standard deviation  $\sigma$ . The variable `_DISP_` specifies which subgroups are displayed on the primary chart ( $\bar{X}$  chart, median chart, or individual measurements chart), and the variable `_DISP2_` specifies which subgroups are displayed on the secondary chart (*R* chart or *s* chart).

- The variables `_COMP_` and `_COMP2_` are not applicable when control limits or control limit parameters are read from a `LIMITS=` data set.
- The variables `_DISP_` and `_DISP2_` take precedence over the display controlled by the `LIMITN=` and `ALLN` options.
- If the input data set is a `DATA=` data set with multiple observations per subgroup, switch variable values must be constant within a subgroup.
- Switch variables are saved in `OUTHISTORY=` and `OUTTABLE=` data sets. Subgroups for which `_DISP_` is equal to `N` are not saved in an `OUTTABLE=` data set, and such subgroups are not displayed in tables created with the `TABLE` and related options.

The following statements illustrate how the switch variables `_COMP_` and `_DISP_` can be used with the bottle production data:

```

data bottles;
  length _comp_ _disp_ $ 1;
  set bottles;
  if      day = '13JAN94'D then _comp_ = 'n';
  else if day = '14JAN94'D then _comp_ = 'n';
  else if day <= '31JAN94'D then _comp_ = 'y';
  else           _comp_ = 'n';
  if      day <= '31JAN94'D then _disp_ = 'n';
  else           _disp_ = 'y';
run;

title 'Analysis of February Production';
proc shewhart data=bottles;
  pchart ncracks * day / subgroupn = nbottles
           nolegend
           nohlabel;
  label ncracks = 'Proportion With Cracks';
run;

```

The chart is identical to the chart in Figure 47.35.

In general, switch variables are more versatile than `WHERE` statements in applications where subgroups are simultaneously selected for computation and display. Switch variables also provide a permanent record of which subgroups were selected. The `WHERE` statement does not alter the input data set; it simply restricts the observations that are read; consequently, the `WHERE` statement can be more efficient than switch variables for processing large data sets.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/QC<sup>®</sup> User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1994 pp.

**SAS/QC<sup>®</sup> User's Guide, Version 8**

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-493-4

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS<sup>®</sup> and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute in the USA and other countries. <sup>®</sup> indicates USA registration.

IBM<sup>®</sup>, ACF/VTAM<sup>®</sup>, AIX<sup>®</sup>, APPN<sup>®</sup>, MVS/ESA<sup>®</sup>, OS/2<sup>®</sup>, OS/390<sup>®</sup>, VM/ESA<sup>®</sup>, and VTAM<sup>®</sup> are registered trademarks or trademarks of International Business Machines Corporation. <sup>®</sup> indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.