Chapter 33
CCHART Statement

Chapter Table of Contents

Chapter 33
CCHART Statement

Overview

The CCHART statement creates *c* charts for the numbers of nonconformities (defects) in subgroup samples.

You can use options in the CCHART statement to

- specify the number of inspection units per subgroup. Typically (but not necessarily), each subgroup consists of a single unit.
- compute control limits from the data based on a multiple of the standard error of the counts or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control the layout and appearance of the chart

Getting Started

This section introduces the CCHART statement with simple examples that illustrate commonly used options. Complete syntax for the CCHART statement is presented in the "Syntax" section on page 1115, and advanced examples are given in the "Examples" section on page 1134.

Creating c Charts from Defect Count Data

See SHWCCHR1 in the SAS/QC Sample Library

A *c* chart is used to monitor the number of paint defects on new trucks. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named TRUCKS, which contains the defect counts:

```
data trucks;
   input truckid $ defects @@;
   label truckid='Truck Identification Number'
         defects='Number of Paint Defects';
   datalines;
C1   5    C2   4    C3   4    C4   8
C5   7    C6  12    C7   3    C8  11
E4   8    E9   4    E7   9    E6  13
A3   5    A4   4    A7   9    Q1  15
Q2   8    Q3   9    Q9  10    Q4   8
;
```

A partial listing of TRUCKS is shown in Figure 33.1.

Paint Defects on New Trucks

truckid	defects
C1	5
C2	4
C3	4
C4	8
C5	7
.	.
.	.
.	.

Figure 33.1. The Data Set TRUCKS

There is a single observation per truck. The variable TRUCKID identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable DEFECTS contains the number of nonconformities in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *c* chart shown in Figure 33.2:

```
title 'c Chart for Paint Defects on New Trucks';
proc shewhart data=trucks lineprinter;
   cchart defects*truckid='*';
run;
```

This example illustrates the basic form of the CCHART statement. After the keyword CCHART, you specify the *process* to analyze (in this case, DEFECTS) followed by an asterisk and the *subgroup-variable* (TRUCKID).

Since the LINEPRINTER option is specified in the PROC SHEWHART statement, line printer output is produced. The asterisk (*) specified in single quotes after the *subgroup-variable* specifies the *character* used to plot points. Note that this character must follow an equal sign.

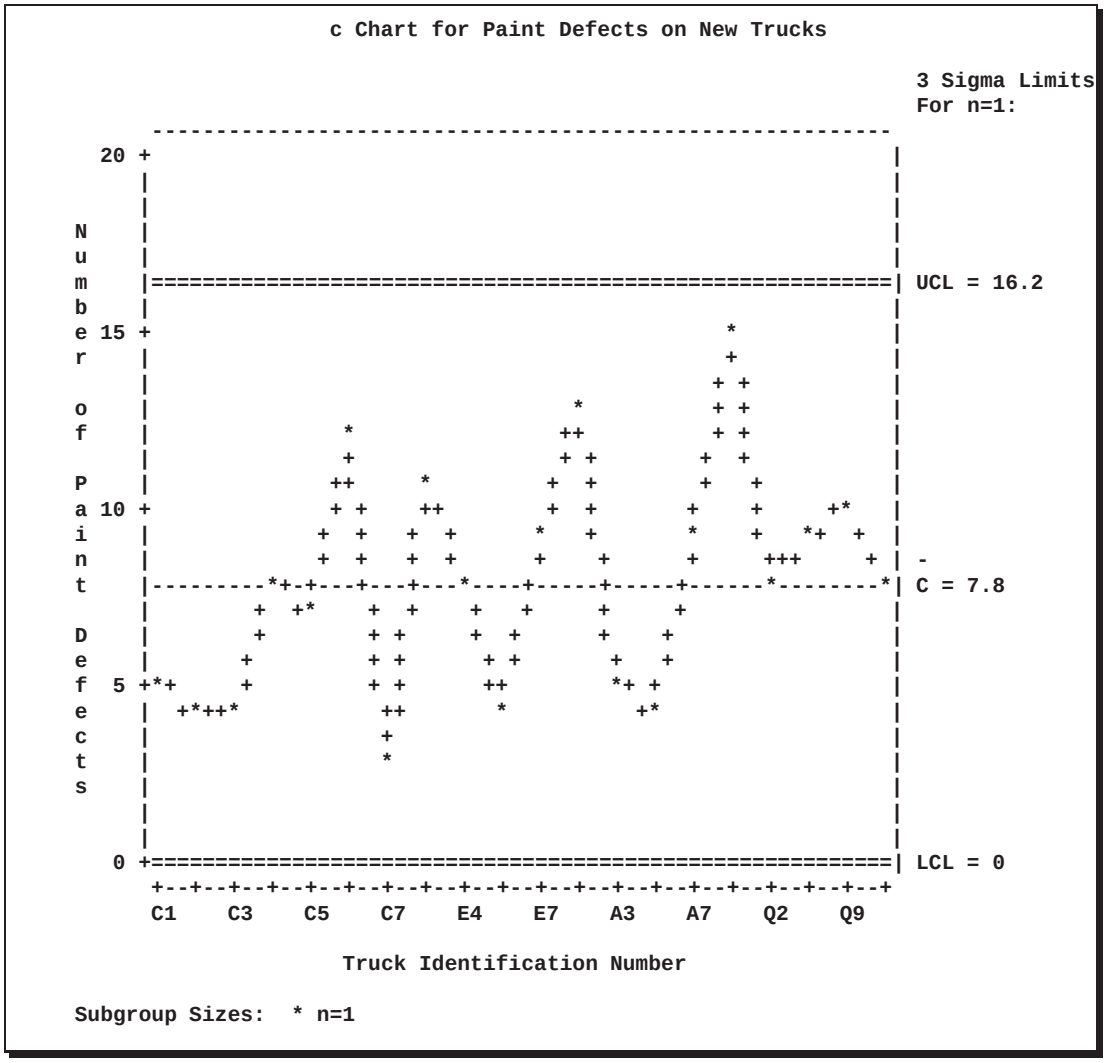


Figure 33.2. A c Chart of Paint Defects

Each point on the c chart represents the number of nonconformities for a particular subgroup. For instance, the value plotted for the first subgroup is 5 (since there are five paint defects on the first truck). By default, the control limits shown are 3σ limits estimated from the data; the formulas are given on page 1126. Since none of

the points exceed the 3σ limits, the c chart indicates that the painting process is in statistical control.

See "Constructing Charts for Numbers of Nonconformities (c Charts)" on page 1125 for details concerning c charts. For more details on reading raw data, see "DATA= Data Set" on page 1130.

Saving Control Limits

See SHWCCHR1 in the SAS/QC Sample Library

You can save the control limits for a c chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1110) or subsequently modify the limits with a DATA step program.

The following statements read the data set TRUCKS introduced on page 1106 and saves the control limit information displayed in Figure 33.2 in a data set named DEFLIM:

```
proc shewhart data=trucks;
   cchart defects*truckid / outlimits=deflim
                            nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. Options such as OUTLIMITS= and NOCHART are specified after the slash (/) in the CCHART statement. A complete list of options is presented in the "Syntax" section on page 1115. The data set DEFLIM is listed in Figure 33.3.

Control Limits Data Set DEFLIM

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_SIGMAS_	_C_	_U_	_LCLC_	_UCLC_
defects	truckid	ESTIMATE	1	.002902622	3	7.8	0	7.8	16.1785

Figure 33.3. The Data Set DEFLIM Containing Control Limit Information

The data set DEFLIM contains one observation with the limits for the *process* DEFECTS. The variables _LCLC_, and _UCLC_ contain the lower and upper control limits. The variable _C_ contains the central line, and the variable _U_ contains the average number of nonconformities per inspection unit. Since all the subgroups contain a single inspection unit, the values of _C_ and _U_ are the same. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see "OUTLIMITS= Data Set" on page 1127.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
title 'Number of Nonconformities and Control Limit Information';
proc shewhart data=trucks;
   cchart defects*truckid / outtable=trucktab
                            nochart;
run;
```

The OUTTABLE= data set TRUCKTAB is listed in Figure 33.4.

Number of Nonconformities and Control Limit Information

VAR	truckid	_SIGMAS_	_LIMITN_	_SUBN_	_LCLC_	_SUBC_	_C_	_UCLC_	_EXLIM_
defects	C1	3	1	1	0	5	7.8	16.1785	
defects	C2	3	1	1	0	4	7.8	16.1785	
defects	C3	3	1	1	0	4	7.8	16.1785	
defects	C4	3	1	1	0	8	7.8	16.1785	
defects	C5	3	1	1	0	7	7.8	16.1785	
defects	C6	3	1	1	0	12	7.8	16.1785	
defects	C7	3	1	1	0	3	7.8	16.1785	
defects	C8	3	1	1	0	11	7.8	16.1785	
defects	E4	3	1	1	0	8	7.8	16.1785	
defects	E9	3	1	1	0	4	7.8	16.1785	
defects	E7	3	1	1	0	9	7.8	16.1785	
defects	E6	3	1	1	0	13	7.8	16.1785	
defects	A3	3	1	1	0	5	7.8	16.1785	
defects	A4	3	1	1	0	4	7.8	16.1785	
defects	A7	3	1	1	0	9	7.8	16.1785	
defects	Q1	3	1	1	0	15	7.8	16.1785	
defects	Q2	3	1	1	0	8	7.8	16.1785	
defects	Q3	3	1	1	0	9	7.8	16.1785	
defects	Q9	3	1	1	0	10	7.8	16.1785	
defects	Q4	3	1	1	0	8	7.8	16.1785	

Figure 33.4. The Data Set TRUCKTAB

This data set contains one observation for each subgroup sample. The variables _SUBC_ and _SUBN_ contain the number of nonconformites per subgroup and the number of inspection units per subgroup. The variables _LCLC_ and _UCLC_ contain the lower and upper control limits, and the variable _C_ contains the central line. The variables _VAR_ and TRUCKID contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1129.

An OUTTABLE= data set can be read later as a TABLE= data set in the SHEWHART procedure. For example, the following statements read TRUCKTAB and display a *c* chart (not shown here) identical to the chart in Figure 33.2:

```
title 'c Chart for Paint Defects in New Trucks';
proc shewhart table=trucktab;
   cchart defects*truckid='*';
label _SUBC_ = 'Number of Paint Defects';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see Chapter 49, "Specialized Control Charts"). For more information, see "TABLE= Data Set" on page 1132.

Reading Preestablished Control Limits

See SHWCCHR1 in the SAS/QC Sample Library

In the previous example, control limits were saved in a SAS data set named DEFLIM. This example shows how these limits can be applied to defect data for a second group of trucks, which are provided in the following data set:

```
data trucks2;
   input truckid $ defects @@;
   label truckid='Truck Identification Number'
         defects='Number of Paint Defects';
   datalines;
R1   7   R2   3   R3   3   R4  14   R5   7
R6   9   R7   3   R8  11   S4   6   S9   8
S7   3   S6  10   T3   9   T4   4   T7   7
X1   7   X2  18   X3   3   X9  11   X4   5
;
```

The following statements plot the number of paint defects for the second group of trucks on a *c* chart using the control limits in DEFLIM. The chart is shown in Figure 33.5.

```
title 'c Chart for Paint Defects in New Trucks';
symbol v=dot;
proc shewhart data=trucks2 limits=deflim;
   cchart defects*truckid;
run;
```

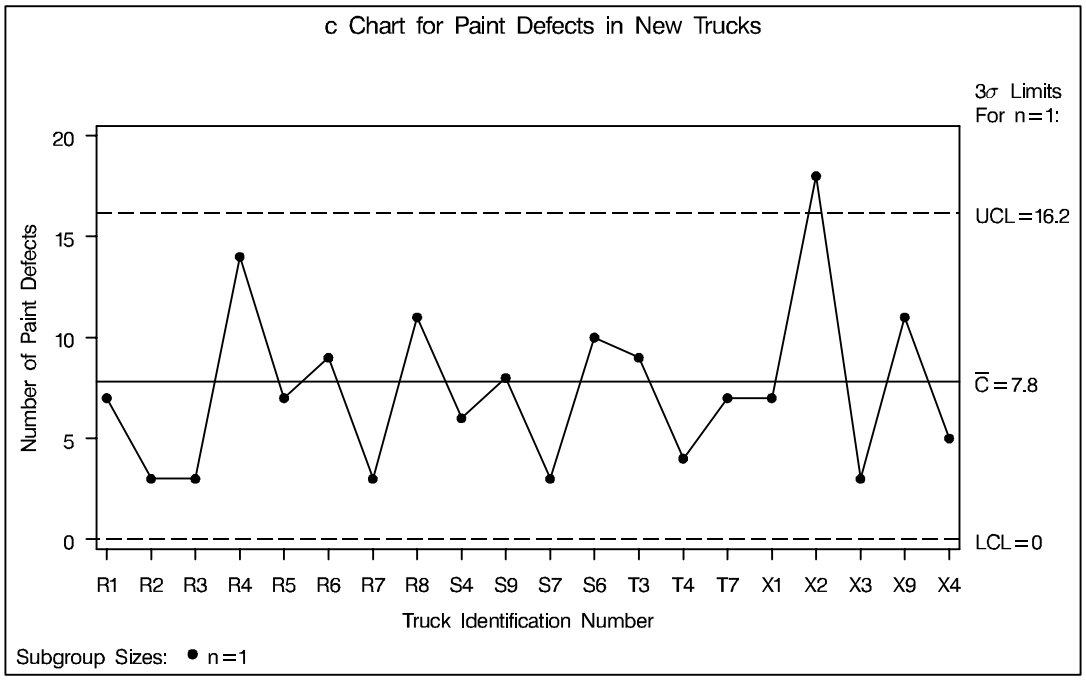


Figure 33.5. A *c* Chart for the Second Set of Trucks

Note that the number of defects on the truck with identification number X2 exceeds the upper control limit, indicating that the process is out-of-control. The LIMITS=

option in the PROC SHEWHART statement specifies the data set containing the control limits. By default,* this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name DEFECTS
- the value of _SUBGRP_ matches the *subgroup-variable* name TRUCKID

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1130 for details concerning the variables that you must provide.

Creating c Charts from Nonconformities per Unit

See SHWCCHR1 in the SAS/QC Sample Library

In the previous example, the input data set provided the number of nonconformities per subgroup sample. However, in some applications, as illustrated here, the data may be provided as the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Since the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in each box is divided by ten and then recorded. The following statements create a SAS data set named SHIRTS, which contains the average number of flaws per shirt for 25 boxes:

```
data shirts;
   input box avgdefu @@;
   avgdefn=10;
   datalines;
 1  0.4    2  0.7    3  0.5    4  1.0   5  0.3
 6  0.2    7  0.0    8  0.4    9  0.4  10  0.6
11  0.2   12  0.7   13  0.3   14  0.1  15  0.3
16  0.6   17  0.6   18  0.3   19  0.7  20  0.3
21  0.0   22  0.1   23  0.5   24  0.6  25  0.4
;
```

A partial listing of SHIRTS is shown in Figure 33.6.

Average Number of Shirt Flaws

box	avgdefu	avgdefn
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
.	.	.
.	.	.
.	.	.

Figure 33.6. The Data Set SHIRTS

*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 data set SHIRTS contains three variables: the box number (BOX), the average number of flaws per shirt (AVGDEFU), and the number of shirts per box (AVGDEFN). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup consists of ten inspection units.

To create a c chart plotting the total number of flaws per box (instead of per shirt), you can specify SHIRTS as a HISTORY= data set.

```
title 'Total Flaws per Box of Shirts';
symbol v=dot;
proc shewhart history=shirts;
   cchart avgdef*box;
run;
```

Note that AVGDEF is *not* the name of a SAS variable in the data set but is instead the common prefix for the SAS variable names AVGDEFU and AVGDEFN. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name referred to as the *process*. The name BOX specified after the asterisk is the name of the *subgroup-variable*. The c chart is shown in Figure 33.7.

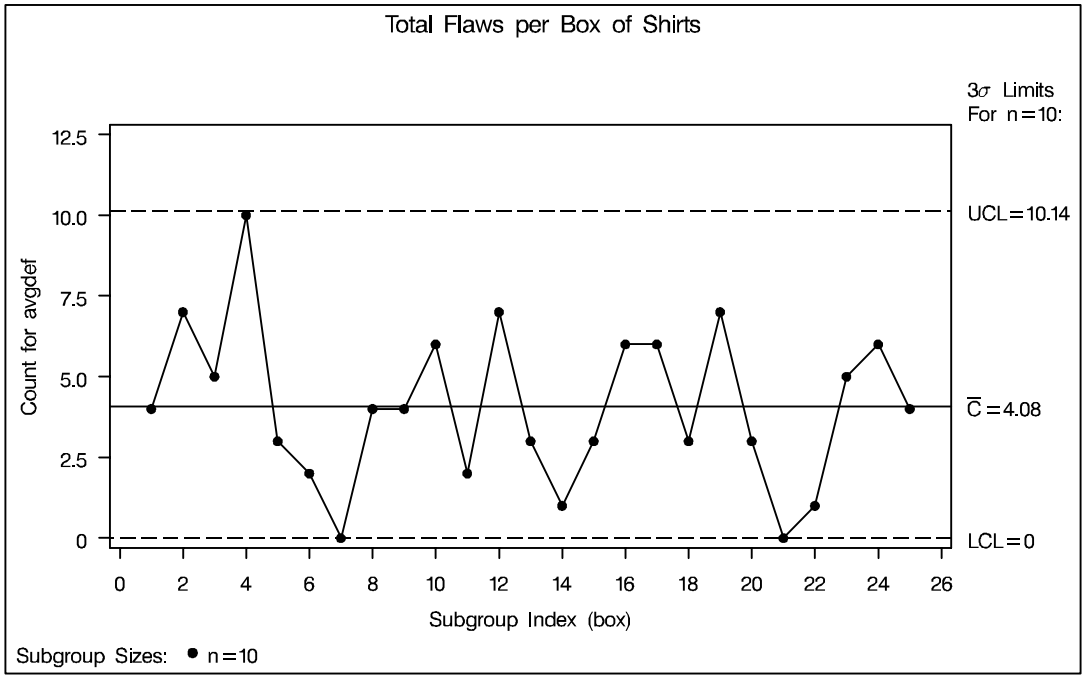


Figure 33.7. A c Chart for Boxes of Shirts

In general, a HISTORY= input data set used with the CCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the CCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables AVGDEFU and AVGDEFN, the data set SHIRTS contained the variables SHIRTDEF and SIZES. The following statements would temporarily rename SHIRTDEF and SIZES to AVGDEFU and AVGDEFN:

```
proc shewhart
   history=shirts (rename=(shirtdef = avgdefu
                           sizes    = avgdefn ));
   cchart avgdef*box;
run;
```

For more information, see “HISTORY= Data Set” on page 1131.

Saving Nonconformities per Unit

See SHWCCHR1 in the SAS/QC Sample Library

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named SHIRTS2, which contains the total defects per box for 20 boxes:

```
data shirts2;
   input box flaws nshirts @@;
   datalines;
 1  3  10    2  8  10    3 15  25    4 20  25
 5  9  25    6  1  10    7  1  10    8 21  50
 9  3  10   10  7  10   11  1  10   12 21  25
13  9  25   14  3  25   15 12  50   16 18  50
17  7  10   18  4  10   19  8  10   20  4  10
;
```

A partial listing of SHIRTS2 is shown in Figure 33.8.

Number of Shirt Flaws per Box

box	flaws	nshirts
1	3	10
2	8	10
3	15	25
4	20	25
5	9	25
.	.	.
.	.	.
.	.	.

Figure 33.8. The Data Set SHIRTS2

The variable BOX contains the box number, the variable FLAWS contains the number of flaws in each box, and the variable NSHIRTS contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=shirts2;
   cchart flaws*box / subgroupn =nshirts
                      outhistory=shrthist
                      nochart;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of SHRTHIST is shown in Figure 33.9.

Average Defects Per Shirt

box	flawsU	flawsN
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25
.	.	.
.	.	.
.	.	.

Figure 33.9. The Data Set SHRTHIST

There are three variables in the data set SHRTHIST.

- BOX contains the subgroup index.
- FLAWSU contains the numbers of nonconformities per inspection unit.
- FLAWSN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* DEFECTS specified in the CCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see “OUTHISTORY= Data Set” on page 1128.

Syntax

The basic syntax for the CCHART statement is as follows:

CCHART *process*******subgroup-variable* **;**

The general form of this syntax is as follows:

CCHART (*processes***)****subgroup-variable* <**(***block-variables* **)** >
< =*symbol-variable* | =*'character'* > < **/** *options* >**;**

You can use any number of CCHART statements in the SHEWHART procedure. The components of the CCHART statement are described as follows.

process
processes
identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities.

  For an example, see "Creating c Charts from Defect Count Data" on page 1106.

- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set.

  For an example, see "Creating c Charts from Nonconformities per Unit" on page 1111.

- If numbers of nonconformities per subgroup, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set.

  For an example, see "Saving Control Limits" on page 1108.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *c* charts for DEFECTS and FLAWS:

```
proc shewhart data=info;
   cchart (defects flaws)*sample;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding CCHART statement, SAMPLE is the subgroup variable. For details, see "Subgroup Variables" on page 1534.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 1684 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities.

- If you produce a chart on a line printer, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce a chart on a graphics device, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOLn statements. See "Displaying Stratification in Levels of a Classification Variable" on page 1683 for an example.

character
specifies a plotting character for charts produced on line printers. For example, the following statements create a *c* chart using an asterisk (*) to plot the points:

```
proc shewhart data=info;
   cchart defects*sample='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" section, which follows, lists all options by function. Chapter 46, "Dictionary of Options," describes each option in detail.

Summary of Options

The following tables list the CCHART statement options by function. For complete descriptions, see Chapter 46, "Dictionary of Options."

Table 33.1. Tabulation Options

TABLE	creates a basic table of subgroup sample sizes, subgroup numbers of nonconformities, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Note that specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.

Table 33.2. Reference Line Options

CHREF=*color*	specifies color for lines requested by the HREF= option
CVREF=*color*	specifies color for lines requested by the VREF= option
HREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFCHAR='*character*'	specifies line character for HREF= lines
HREFDATA=*SAS-data-set*	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS='*label1*'...'*labeln*'	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies position of HREFLABELS= labels
LHREF=*linetype*	specifies line type for HREF= lines
LVREF=*linetype*	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=*values*|*SAS-data-set*	specifies position of reference lines perpendicular to vertical axis
VREFCHAR='*character*'	specifies line character for VREF= lines
VREFLABELS='*label1*'...'*labeln*'	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies position of VREFLABELS= labels

Table 33.3. Options for Specifying Tests for Special Causes

NO3SIGMACHECK	allows tests to be applied with control limits other than 3σ limits
TESTS=*value-list*| *customized-pattern-list*	specifies tests for special causes
TEST2RUN=*n*	specifies length of pattern for Test 2
TEST3RUN=*n*	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL='*label*'| (*variable*)|*keyword*	provides labels for points where test is positive
TESTLABEL*n*='*label*'	specifies label for n^{th} test for special causes
TESTNMETHOD= STANDARDIZE	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=*n*	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values

Table 33.4. Graphical Options for Displaying Tests for Special Causes

CTESTS=*color*| *test-color-list*	specifies color for labels indicating points where test is positive
CZONES=*color*	specifies color for lines and labels delineating zones A, B, and C
LABELFONT=*font*	specifies software font for labels at points where test is positive (alias for the TESTFONT= option)
LABELHEIGHT=*value*	specifies height of labels at points where test is positive (alias for the TESTHEIGHT= option)
LTESTS=*linetype*	specifies type of line connecting points where test is positive
LZONES=*linetype*	specifies line type for lines delineating zones A, B, and C
TESTFONT=*font*	specifies software font for labels at points where test is positive
TESTHEIGHT=*value*	specifies height of labels at points where test is positive

Table 33.5. Line Printer Options for Displaying Tests for Special Causes

TESTCHAR='*character*'	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
ZONECHAR='*character*'	specifies character for lines that delineate zones for tests for special causes

Table 33.6. Options for Plotting and Labeling Points

ALLLABEL=VALUE|(*variable*)	labels every point
CCONNECT=*color*	specifies color for line segments that connect points on chart
CFRAMELAB=*color*	specifies fill color for frame around labeled points
CNEEDLES=*color*	specifies color for needles that connect points to central line
CONNECTCHAR='*character*'	specifies character used to form line segments that connect points on chart
COUT=*color*	specifies color for portions of line segments that connect points outside control limits
COUTFILL=*color*	specifies color for shading areas between the connected points and control limits outside the limits
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=VALUE|(*variable*)	labels points outside control limits
SYMBOLCHARS='*characters*'	specifies characters indicating *symbol-variable*
SYMBOLLEGEND=NONE|*name*	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=*keyword*	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically

Table 33.7. Clipping Options

CCLIP=*color*	specifies color for plot symbol for clipped points
CLIPCHAR='*character*'	specifies plot character for clipped points
CLIPFACTOR=*value*	determines extent to which extreme points are clipped
CLIPLEGEND='*string*'	specifies text for clipping legend
CLIPLEGPOS=*keyword*	specifies position of clipping legend
CLIPSUBCHAR='*character*'	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=*symbol*	specifies plot symbol for clipped points
CLIPSYMBOLHT=*value*	specifies symbol marker height for clipped points

Table 33.8. Standard Value Options

TYPE=*keyword*	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U0=*value*	specifies known average number of nonconformities per unit

Table 33.9. Axis and Axis Label Options

CAXIS=*color*	specifies color for axis lines and tick marks
CFRAME=*color*|(*color-list*)	specifies fill colors for frame for plot area
CTEXT=*color*	specifies color for tick mark values and axis labels
HAXIS=*values*|AXISn	specifies major tick mark values for horizontal axis
HEIGHT=*value*	specifies height of axis label and axis legend text
HMINOR=*n*	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=*value*	specifies length of offset at both ends of horizontal axis
INTSTART=*value*	specifies first major tick mark value for numeric horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=*n*	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=*values*|AXISn	specifies major tick mark values for vertical axis
VMINOR=*n*	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=*value*	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=*n*	specifies width of axis lines

Table 33.10. Input Data Set Options

MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN=*n*|*variable*	specifies subgroup sample sizes as constant number n or as values of *variable* in a DATA= data set

Table 33.11. Output Data Set Options

OUTHISTORY=*SAS-data-set*	creates output data set containing subgroup numbers of nonconformities per unit and subgroup sample sizes
OUTINDEX='*string*'	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=*SAS-data-set*	creates output data set containing control limits
OUTTABLE=*SAS-data-set*	creates output data set containing subgroup numbers of nonconformities, subgroup sample sizes, and control limits

Table 33.12. Options for Specifying Control Limits

ALPHA=*value*	requests probability limits for control charts
LIMITN=*n*|VARYING	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (Release 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEXES=ALL| '*label1*'...'*labeln*'	reads multiple sets of control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (Release 6.09 and earlier releases)
SIGMAS=*k*	specifies width of control limits in terms of multiple k of standard error of plotted statistic

Table 33.13. Options for Displaying Control Limits

CINFILL=*color*	specifies color for area inside control limits
CLIMITS=*color*	specifies color of control limits, central line, and related labels
CSYMBOL='*string*'| *keyword*	specifies label for central line
LCLLABEL='*label*'	specifies label for lower control limit
LIMLABSUBCHAR= '*character*'	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=*linetype*	specifies line type for control limits
NDECIMAL=*n*	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOLIMIT0	suppresses display of lower control limit if it is 0
NOUCL	suppresses display of upper control limit
UCLLABEL='*string*'	specifies label for upper control limit
WLIMITS=*n*	specifies width for control limits and central line

Table 33.14. Phase Options

CPHASEBOX=*color*	specifies color for box enclosing all plotted points for a phase
CPHASEBOX-CONNECT=*color*	specifies color for line segments connecting adjacent enclosing boxes
CPHASEBOXFILL=*color*	specifies fill color for box enclosing all plotted points for a phase
CPHASELEG=*color*	specifies text color for *phase* legend
CPHASEMEAN-CONNECT=*color*	specifies color for line segments connecting average value points within a phase
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE='*string*'	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=*value*|*keyword*	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEMEANSYMBOL=*symbol*	specifies symbol marker for average of values within a phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES= ALL|'*label1*'...'*labeln*'	specifies *phases* to be read from an input data set

Table 33.15. Options for Interactive Control Charts

HTML=(*variable*)	specifies a variable whose values are URLs to be associated with subgroups
HTML_LEGEND=(*variable*)	specifies a variable whose values are URLs to be associated with symbols in the symbol legend
TESTURLS=*SAS-data-set*	associates URLs with tests for special causes
WEBOUT=*SAS-data-set*	creates an OUTTABLE= data set with additional graphics co-ordinate data

Table 33.16. Block Variable Legend Options

BLOCKLABELPOS=*keyword*	specifies position of label for *block-variable* legend
BLOCKLABTYPE=*value*|*keyword*	specifies text size of *block-variable* legend
BLOCKPOS=*n*	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=*color*	specifies color for filling background in *block-variable* legend
CBLOCKVAR=*variable*|(*variables*)	specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 33.17. Plot Layout Options

ALLN	plots numbers of nonconformities for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=*keyword*	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=*n*	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=*n*	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position on next panel
TOTPANELS=*n*	specifies number of pages or screens to be used to display chart
ZEROSTD	displays c chart regardless of whether $\hat{\sigma} = 0$

Table 33.18. Graphical Enhancement Options

ANNOTATE= *SAS-data-set*	specifies annotate data set that adds features to chart
DESCRIPTION='*string*'	specifies string that appears in the description field of the PROC GREPLAY master menu
FONT=*font*	specifies software font for labels and legends on charts
NAME='*string*'	specifies name that appears in the name field of the PROC GREPLAY master menu
PAGENUM='*string*'	specifies the form of the label used in pagination
PAGENUMPOS= *keyword*	specifies the position of the page number requested with the PAGENUM= option

Table 33.19. Grid Options

ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=*linetype*	specifies line type for grid requested with the ENDGRID option
LGRID=*linetype*	specifies line type for grid requested with the GRID option
WGRID=*n*	specifies width of grid lines

Table 33.20. Star Options

CSTARCIRCLES=*color*	specifies color for circles specified by the STARCIRCLES= option
CSTARFILL=*color*|*(variable)*	specifies color for filling stars
CSTAROUT=*color*	specifies outline color for stars exceeding inner or outer circles
CSTARS=*color*| *(variable)*	specifies color for outlines of stars
LSTARCIRCLES=*linetypes*	specifies line types for the circles requested with the STARCIRCLES= option
LSTARS=*linetype*|*(variable)*	specifies line types for outlines of stars requested with the STARVERTICES= option
STARBDRADIUS=*value*	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=*value-list*	specifies reference circles for stars
STARINRADIUS=*value*	specifies inner radius of stars
STARLABEL=*keyword*	specifies vertices to be labeled
STARLEGEND=*keyword*	specifies style of legend for star vertices
STARLEGENDLAB=*'label'*	specifies label for STARLEGEND= legend
STAROUTRADIUS=*value*	specifies outer radius of stars
STARSPEC=*value*|*SAS-data-set*	specifies method used to standardize vertex variables
STARSTART=*value*	specifies angle for first vertex
STARTYPE=*keyword*	specifies graphical style of star
STARVERTICES=*variable*|*(variables)*	superimposes star at each point on chart
WSTARCIRCLES=*n*	specifies width of circles requested by the STARCIRCLES= option
WSTARS=*n*	specifies width of stars requested by the STARVERTICES= option

Details

Constructing Charts for Numbers of Nonconformities (c Charts)

The following notation is used in this section:

u	expected number of nonconformities per unit produced by the process
u_i	number of nonconformities per unit in the i^{th} subgroup
c_i	total number of nonconformities in the i^{th} subgroup
n_i	number of inspection units in the i^{th} subgroup. Typically, $n_i = 1$ and $u_i = c_i$ for c charts. In general, $u_i = c_i/n_i$.
$\bar{u}$	average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_N u_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
χ^2_ν	has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a c chart represents the total number of nonconformities (c_i) in a subgroup. For example, Figure 33.10 displays three sections of pipeline that are inspected for defective welds (indicated by an *X*). Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long sections. The number of units in the i^{th} subgroup is denoted by n_i, which is the subgroup sample size. The value of n_i can be fractional; Figure 33.10 shows $n_3 = 2.5$ units in the third subgroup.

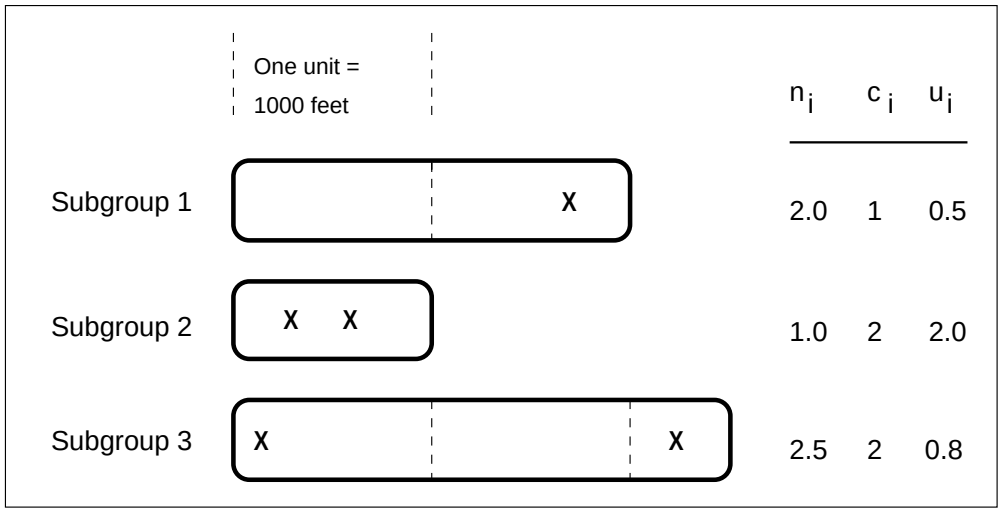


Figure 33.10. Terminology for c Charts and u Charts

The *number of nonconformities* in the i^{th} subgroup is denoted by c_i. The *number of nonconformities per unit* in the i^{th} subgroup is denoted by $u_i = c_i/n_i$. In Figure 33.10, the number of welds per inspection unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A u chart created with the UCHART statement plots the quantity u_i for the i^{th} subgroup (see Chapter 41). An advantage of a u chart is that the value of the central line at the i^{th} subgroup does not depend on n_i. This is not the case for a c chart, and consequently, a u chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a c chart, the central line indicates an estimate for $n_i u$, which is computed as $n_i \bar{u}$. If you specify a known value (u_0) for u, the central line indicates the value of $n_i u_0$.

Note that the central line varies with subgroup sample size n_i. When $n_i = 1$ for all subgroups, the central line has the constant value $\bar{c} = (c_1 + \cdots + c_N)/N$.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of c_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that c_i exceeds the limits

The lower and upper control limits, LCLC and UCLC respectively, are given by

$$\begin{aligned} \text{LCLC} &= \max\left(n_i\bar{u} - k\sqrt{n_i\bar{u}}\,, 0\right) \\ \text{UCLC} &= n_i\bar{u} + k\sqrt{n_i\bar{u}} \end{aligned}$$

The upper and lower control limits vary with the number of inspection units per subgroup n_i. If $n_i = 1$ for all subgroups, the control limits have constant values.

$$\begin{aligned} \text{LCLC} &= \max\left(\bar{c} - k\sqrt{\bar{c}}\,, 0\right) \\ \text{UCLC} &= \bar{c} + k\sqrt{\bar{c}} \end{aligned}$$

An upper probability limit UCLC for c_i can be determined using the fact that

$$\begin{aligned} P\{c_i > \text{UCLC}\} &= 1 - P\{c_i \le \text{UCLC}\} \\ &= 1 - P\{\chi^2_{2(\text{UCLC}+1)} \ge 2n_i\bar{u}\} \end{aligned}$$

The upper probability limit UCLC is then calculated by setting

$$1 - P\{\chi^2_{2(\text{UCLC}+1)} \ge 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLC.

A similar approach is used to calculate the lower probability limit LCLC, using the fact that

$$P\{c_i < \text{LCLC}\} = P\{\chi^2_{2(\text{LCLC}+1)} > 2n_i\bar{u}\}$$

The lower probability limit LCLC is then calculated by setting

$$P\{\chi^2_{2(\text{LCLC}+1)} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLC. This assumes that the process is in statistical control and that c_i has a Poisson distribution. For more information, refer to Johnson, Kotz, and Kemp (1992). Note that the probability limits vary with the number of inspection units per subgroup (n_i) and are asymmetric about the central line.

If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits. You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 33.21. OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_C_	value of central line on c chart ($n_i\bar{u}$ or n_iu_0)
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLC	lower control limit for number of nonconformities
LIMITN	sample size associated with the control limits
SIGMAS	multiple (k) of standard error of c_i
SUBGRP	*subgroup-variable* specified in the CCHART statement
TYPE	type (estimate or standard value) of _U_
_U_	average number of nonconformities per unit ($\bar{u}$ or u_0)
UCLC	upper control limit for number of nonconformities
VAR	*process* specified in the CCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _C_, _LCLC_, _UCLC_, and _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of c_i, the value of _ALPHA_ is computed as $P\{c_i < \text{\_LCLC\_}\} + P\{c_i > \text{\_UCLC\_}\}$. If control limits vary with subgroup sample size and are determined in terms of k, _ALPHA_ is assigned the special missing value *V*.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLC\_} - \text{\_C\_})/\sqrt{\text{\_C\_}}$. If probability limits vary with subgroup sample size, _SIGMAS_ is assigned the special missing value *V*.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the CCHART statement. For an example, see "Saving Control Limits" on page 1108.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup number of nonconformities per unit variable named by *process* suffixed with *U*

Given a *process* name that contains eight characters, the procedure first shortens the name to its first four characters and its last three characters, and then it adds the suffix. For example, the procedure shortens the *process* NDEFECTS to NDEFCTS before adding the suffix.

Subgroup summary variables are created for each *process* specified in the CCHART statement. For example, consider the following statements:

```
proc shewhart data=fabric;
   cchart (flaws ndefects)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, FLAWSU, FLAWSN, NDEFCTSU, and NDEFCTSN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example that creates an OUTHISTORY= data set, see "Saving Nonconformities per Unit" on page 1113. Note that an OUTHISTORY= data set created with the

CCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on c chart
LCLC	lower control limit for number of nonconformities
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBC	subgroup number of nonconformities
SUBN	subgroup sample size
TESTS	tests for special causes signaled on c chart
UCLC	upper control limit for number of nonconformities
VAR	*process* specified in the CCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).

2. The variable _TESTS_ is saved if you specify the TESTS= option. The k^{th} character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for c charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.

3. The variables _VAR_, _EXLIM_, and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 16. All other variables are numeric.

For an example, see “Saving Control Limits” on page 1108.

ODS Tables

The following table summarizes the ODS tables that you can request with the CCHART statement.

Table 33.22. ODS Tables Produced with the CCHART Statement

Table Name	Description	Options
CCHART	*c* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
Tests	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

Input Data Sets

DATA= Data Set

You can read the number of nonconformities in subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the CCHART statement must be a SAS variable in the data set. This variable provides the number of nonconformities in subgroup samples indexed by the *subgroup-variable*. Typically (but not necessarily), the subgroup consists of a single inspection unit. The *subgroup-variable*, specified in the CCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set must contain one observation per subgroup. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see “Displaying Stratification in Phases” on page 1689).

For an example of a DATA= data set, see “Creating c Charts from Defect Count Data” on page 1106.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from

the data set CONLIMS:*

```
proc shewhart data=info limits=conlims;
   cchart defects*lot;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLC_, _C_, and _UCLC_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see page 1126)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables of length 8.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable of length 16.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are `ESTIMATE` and `STANDARD`.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1110.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set. A HISTORY= data set used with the CCHART statement must contain the following variables:

- *subgroup-variable*
- subgroup number of nonconformities per unit variable for each *process*
- subgroup sample size variable (number of units per subgroup) for each *process*

The names of the subgroup number of nonconformities per unit and subgroup sample size variables must be the *process* name concatenated with the special suffix characters U and N, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   cchart (flaws ndefects)*lot;
run;
```

*In Release 6.09 and in earlier releases, it is necessary to specify the READLIMITS option.

The data set SUMMARY must include the variables LOT, FLAWSU, FLAWSN, NDEFCTSU, and NDEFCTSN.

Note that if you specify a *process* name that contains eight characters, the names of the summary variables must be formed from the first four characters and the last three characters of the *process* name, suffixed with the appropriate character. Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 1689 for an example).

For an example of a HISTORY= data set, see "Creating c Charts from Nonconformities per Unit" on page 1111.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in Chapter 49, "Specialized Control Charts."

The following table lists the variables required in a TABLE= data set used with the CCHART statement:

Table 33.23. Variables Required in a TABLE= Data Set

Variable	Description
_C_	average number of nonconformities
LCLC	lower control limit for nonconformities
LIMITN	nominal sample size associated with the control limits
subgroup-variable	values of the *subgroup-variable*
SUBC	subgroup number of nonconformities
SUBN	subgroup sample size
UCLC	upper control limit for nonconformities

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable of length 16.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable of length 8.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1108.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup defect counts variable
Vertical	TABLE=	_SUBC_

For an example, see "Labeling Axes" on page 1719.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples

This section provides advanced examples of the CCHART statement.

Example 33.1. Applying Tests for Special Causes

See SHWCEX1 in the SAS/QC Sample Library

This example illustrates how you can apply tests for special causes to make *c* charts more sensitive to special causes of variation. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named TRUCKS3:

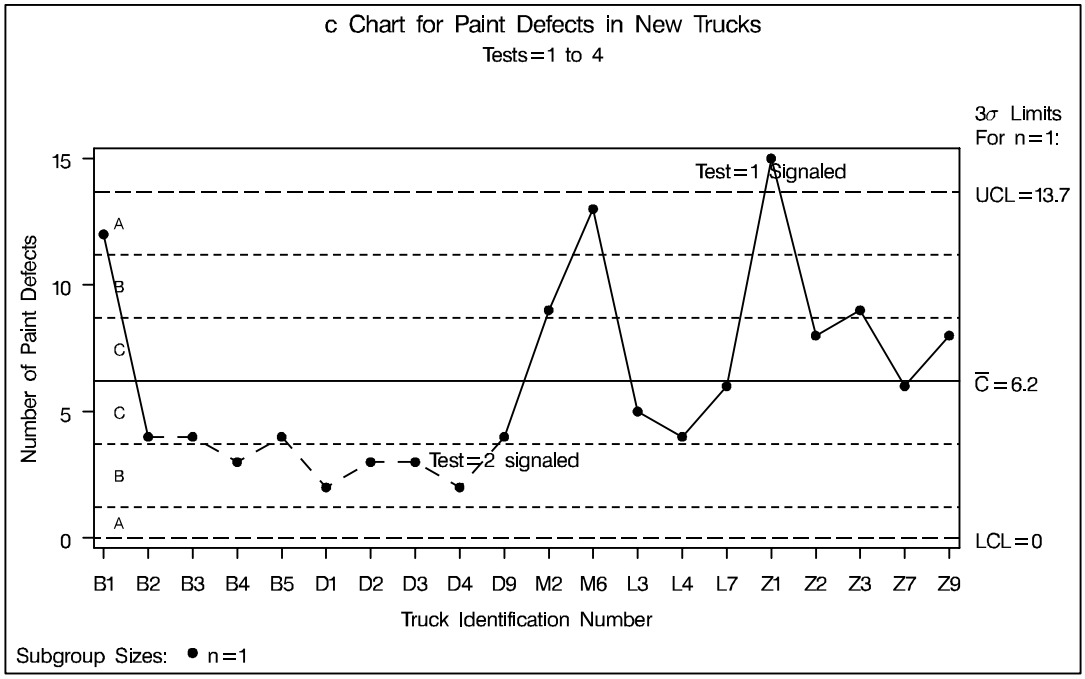
```
data trucks3;
   input truckid $ defects @@;
   label truckid='Truck Identification Number'
         defects='Number of Paint Defects';
   datalines;
B1  12   B2   4   B3   4   B4   3   B5   4
D1   2   D2   3   D3   3   D4   2   D9   4
M2   9   M6  13   L3   5   L4   4   L7   6
Z1  15   Z2   8   Z3   9   Z7   6   Z9   8
;
```

The following statements create a *c* chart and tabulate the information on the chart. The chart and table are shown in Output 33.1.1 and Output 33.1.2.

```
symbol v=dot;
title1 'c Chart for Paint Defects in New Trucks';
title2 'Tests=1 to 4';
proc shewhart data=trucks3;
   cchart defects*truckid / tests     =1 to 4
                            testlabel1='Test=1 Signaled'
                            testlabel2='Test=2 Signaled'
                            testfont  =swiss
                            ltests    =20
                            zonelabels
                            tabletests
                            tablelegend;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in Chapter 48, "Tests for Special Causes." Only Tests 1, 2, 3, and 4 are recommended for *c* charts. The TESTLABEL1= and TESTLABEL2= options specify the labels for points where Tests 1 and 2 are positive. The TESTFONT= option specifies the font for the labels indicating points at which the tests are positive.

Output 33.1.1. Tests for Special Causes Displayed on c Chart



Output 33.1.2. Tabular Form of c Chart

c Chart for Paint Defects in New Trucks
Tests=1 to 4

truckid	Subgroup Sample Size	-3 Sigma Limits with n=1 for Count- Lower Limit	Subgroup Count	Upper Limit	Special Tests Signaled
B1	1.00000	0	12.000000	13.669940	
B2	1.00000	0	4.000000	13.669940	
B3	1.00000	0	4.000000	13.669940	
B4	1.00000	0	3.000000	13.669940	
B5	1.00000	0	4.000000	13.669940	
D1	1.00000	0	2.000000	13.669940	
D2	1.00000	0	3.000000	13.669940	
D3	1.00000	0	3.000000	13.669940	
D4	1.00000	0	2.000000	13.669940	
D9	1.00000	0	4.000000	13.669940	2
M2	1.00000	0	9.000000	13.669940	
M6	1.00000	0	13.000000	13.669940	
L3	1.00000	0	5.000000	13.669940	
L4	1.00000	0	4.000000	13.669940	
L7	1.00000	0	6.000000	13.669940	
Z1	1.00000	0	15.000000	13.669940	1
Z2	1.00000	0	8.000000	13.669940	
Z3	1.00000	0	9.000000	13.669940	
Z7	1.00000	0	6.000000	13.669940	
Z9	1.00000	0	8.000000	13.669940	

Test Descriptions

Test 1 One point beyond Zone A (outside control limits)
Test 2 Nine points in a row on one side of center line

The ZONELABELS option requests zone lines and displays zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled. The TABLETESTS option requests a table of counts of nonconformities, subgroup sample sizes, and control limits, together with a column indicating the subgroups at which the tests are positive. The TABLELEGEND option adds a legend describing the tests that are positive.

Output 33.1.1 and Output 33.1.2 indicate that Test 1 is positive at Truck Z1 and Test 2 is positive at Truck D9.

Example 33.2. Specifying a Known Expected Number of Nonconformities

See SHWCEX2 in the SAS/QC Sample Library

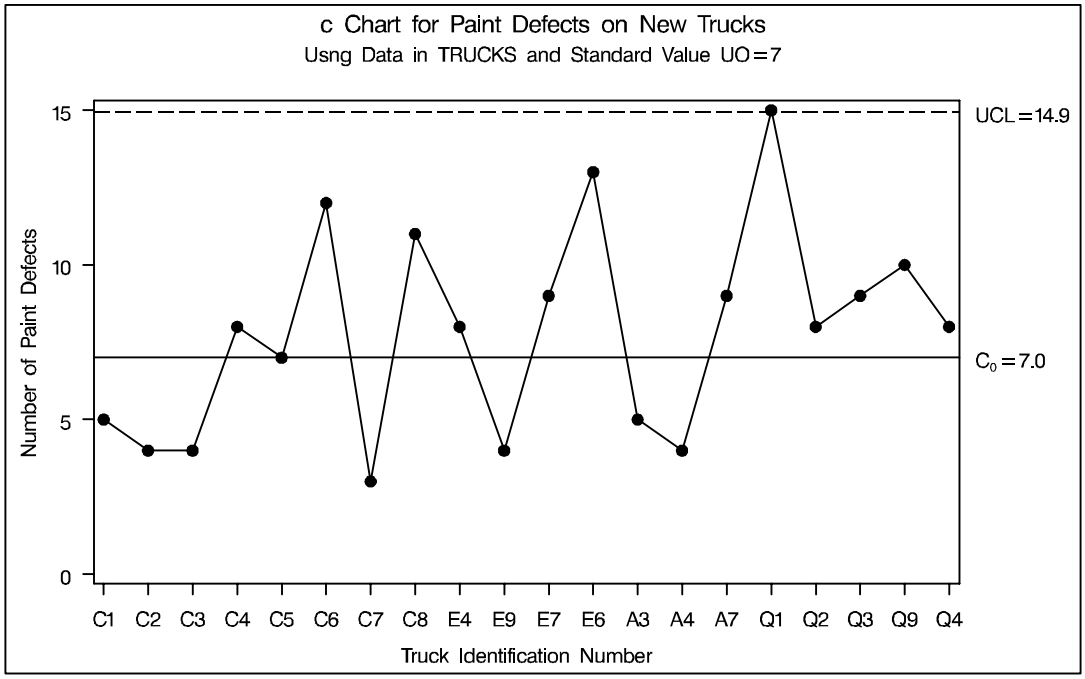
This example illustrates how you can create a c chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

A c chart is used to monitor the number of paint defects per truck. The defect counts are provided as values of the variable DEFECTS in the data set TRUCKS given on page 1106. Based on previous testing, it is known that $u_0 = 7$. The following statements create a c chart with control limits derived from this value:

```
title 'c Chart for Paint Defects on New Trucks';
title2 'Using Data in TRUCKS and Standard Value U0=7';
symbol v=dot;
proc shewhart data=trucks;
   cchart defects*truckid / u0     = 7
                            csymbol= c0
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart is shown in Output 33.2.1. The U0= option specifies u_0, and the CSYMBOL= option requests a label for the central line indicating that the line represents a standard value. The NOLEGEND option suppresses the legend for the subgroup sample size, and the NOLIMITSLEGEND option suppresses the legend for the control limits that appears by default in the upper right corner of the chart. The NOLIMIT0 option suppresses the display of the lower limit when it is equal to zero.

Output 33.2.1. A c Chart with Standard Value u_0



The number of paint defects on Truck Q1 exceeds the upper control limit, indicating that the process is out of control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data tlimits;
   length _subgrp_ _var_ _type_ $8;
   _U_      = 7;
   _subgrp_ = 'truckid';
   _var_    = 'defects';
   _limitn_ = 1;
   _type_   = 'STANDARD';

proc shewhart data=trucks limits=tlimits;
   cchart defects*truckid / csymbol=c0
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart produced by these statements is identical to the one in Output 33.2.1.

For further details, see "LIMITS= Data Set" on page 1130.

Example 33.3. Creating c Charts for Varying Numbers of Units

See SHWCEX3 in the SAS/QC Sample Library

In applications where the number of inspection units per subgroup is not equal to one, a u chart is typically used to analyze the number of nonconformities *per unit* (see Chapter 41, "UCHART Statement"). However, as shown in this example, you can use the CCHART statement to create a c chart for this type of data.

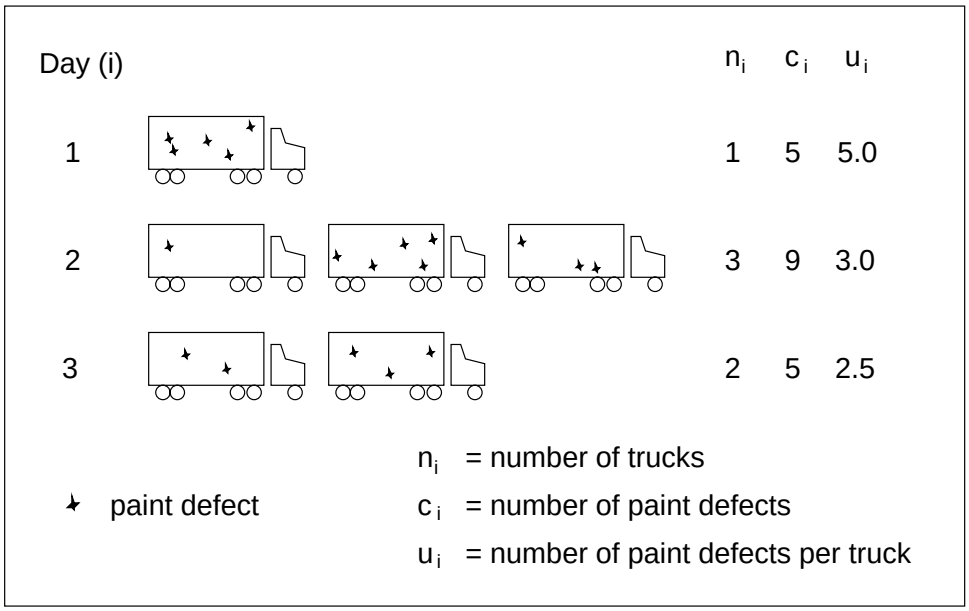


Figure 33.11. lDifference between c Charts and u Charts

Figure 33.11 illustrates a situation in which varying numbers of trucks are painted each day. Trucks painted on the same day are regarded as *subgroups*, and each truck is regarded as an *inspection unit*. The following statements create a SAS data set named TRUCKS4, which contains paint defects for trucks painted on 26 days:

```
data trucks4;
   input day defects ntrucks @@;
   label day='Day'
         defects='Number of Paint Defects';
   datalines;
 1   5  1    2   9  3
 3   5  2    4   9  2
 5  24  4    6  10  2
 7  15  3    8  17  3
 9  16  3   10  13  2
11  28  4   12  18  5
13   8  2   14   7  2
15   5  1   16  17  3
17   2  1   18  17  3
19  15  4   20  19  5
21   6  3   22  23  5
23  27  4   24   6  2
25  12  2   26  12  3
;
```

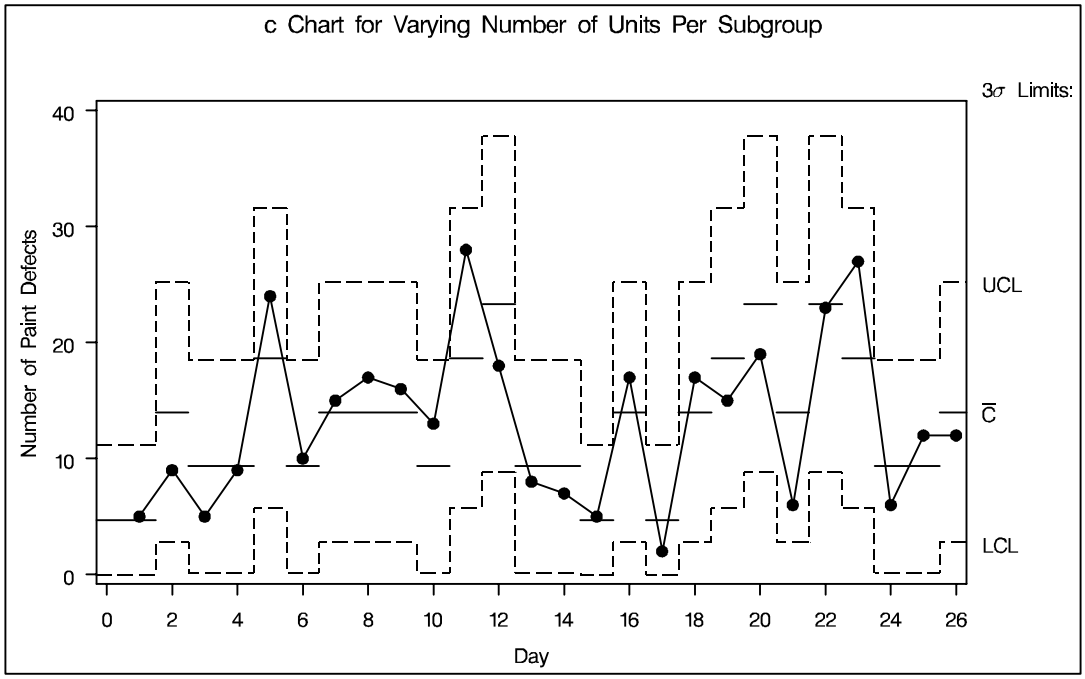
The variable DEFECTS provides the defect count (c_i) for the i^{th} day, and the variable NTRUCKS provides the number of inspection units (n_i). The following statements create a c chart for this data:

```
title 'c Chart for Paint Defects on New Trucks';
symbol v=dot;
proc shewhart data=trucks4;
   cchart defects*day /
      subgroupn=ntrucks
      nolegend;
run;
```

The SUBGROUPN= option specifies the subgroup sample size variable NTRUCKS (in general, the values of this variable need not be integers). Alternatively, you can specify a fixed value with the SUBGROUPN= option. When this option is not specified, it is assumed that $n_i = 1$.

The chart is shown in Output 33.3.1. Note that the central line and the control limits vary with the number of inspection units.

Output 33.3.1. c Chart for Varying Number of Units



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

SAS/QC® User's Guide, Version 8

ISBN 1–58025–493–4

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

1st printing, October 1999