TestPoint™

Statistical Process Control

Toolkit

User's Manual

TestPoint SPC Toolkit
First edition

Capital Equipment Corporation
900 Middlesex Turnpike, Building 2
Billerica, Massachusetts 01821
(978) 663-2002

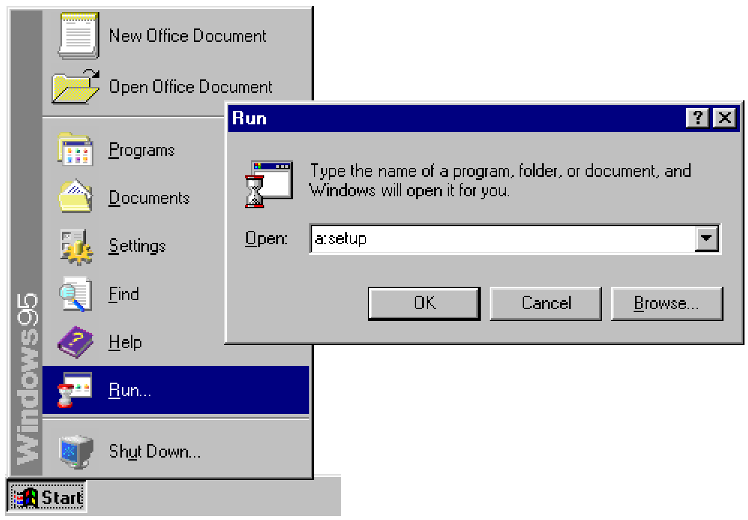
Chapter 1. Installation

System Requirements

- TestPoint version 2.0 or later
- Windows 3.x, 95, 98, or NT
- At least 8M RAM, and 1M free disk space

Setup

To install the TestPoint SPC Toolkit, run SETUP from the distribution diskette. In Windows 3.x, you use the File/Run menu command in Program Manager. In Windows 95, 98, or NT, you use the Start button and Run command as shown below:

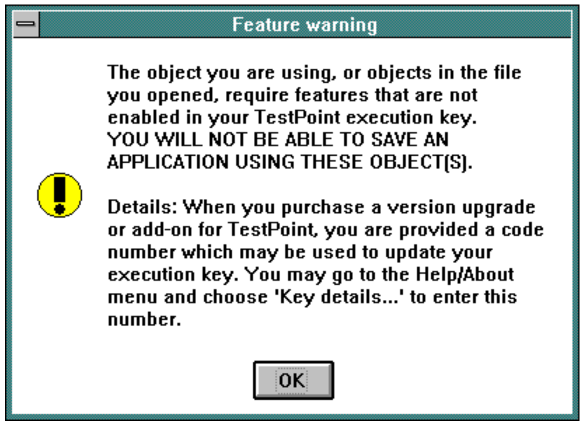


Install the SPC Toolkit into the **same** directory as your existing TestPoint development system.

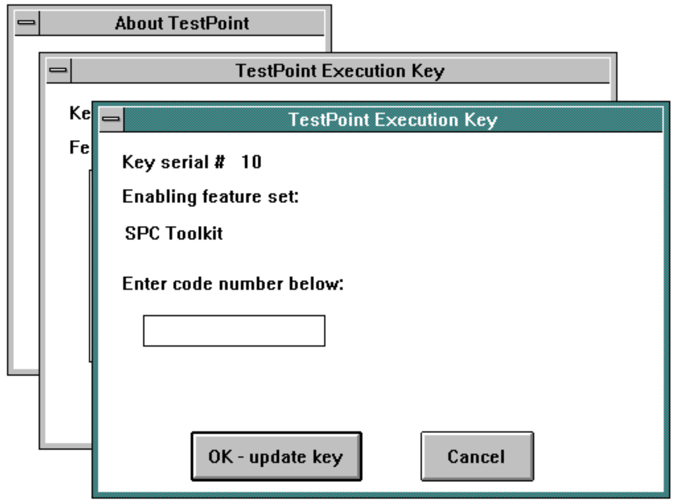
Enabling the Toolkit

The next part of installation is to enable the new features, by updating your TestPoint execution control key. To do this, follow these steps:

1. Run the TestPoint editor.
2. Use File/Open and select the SPC.TST example from the SPC directory. You will get a warning about features that are not yet enabled:



3. Select Help/About/Key details.../Add key features:



4. Enter the update code number provided with your copy of the TestPoint SPC Toolkit. This code number is specific to your key's serial number.

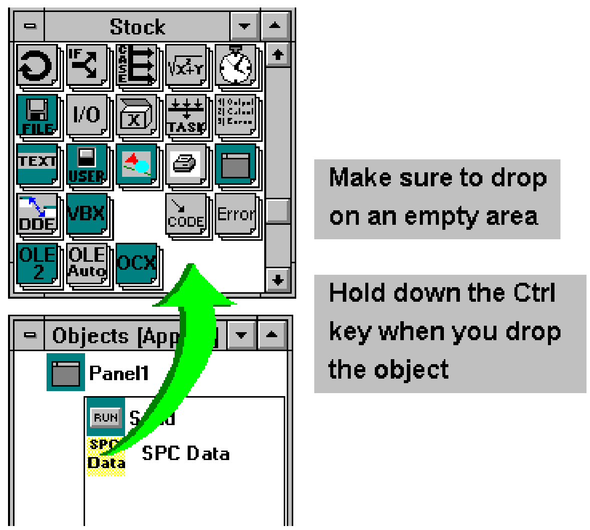
If you do not have an update code sheet included in your package (perhaps because it was in stock on the shelf of a reseller), simply contact us and we will immediately issue the code to you:

phone 978-663-2002
FAX 978-663-2626
email support@cec488.com

Please include your key serial number, your name and company and phone number and e-mail address.

Optional - Add the SPC objects to your Stock window

You may wish to add the SPC objects to your Stock window, so they are easy to use in creating new applications. (If not, you can always copy them from SPC.TST and paste them into new applications). To add each object to the stock, drag it from the Objects window to an **empty** area in the Stock window, hold down the **Ctrl** key, and drop it into the stock:



Chapter 2. Introduction

Quality control is an essential part of any manufacturing or production operation. Initiatives to monitor and improve processes can be found everywhere, as ISO 9000 programs, TQM, or other names.

Statistical Process Control (SPC) is a method of gathering, charting, and analyzing data to solve practical quality problems. Statistical techniques for production sampling were developed back in the 1920s at places like Western Electric and Bell Telephone. These methods, with further refinements, were adopted by the Japanese in the 1950s with the help of Dr. W. Edwards Deming. Deming's quality techniques were later reintroduced into the United States, and have become very popular in the 1980s and 90s. A well-run quality program is now considered essential to remaining competitive in any manufacturing industry.

Quality begins with specification and design. A specification describes the features and attributes of the product, and the allowable variations, that make it suitable for the intended use. The product's design allows it to meet these goals.

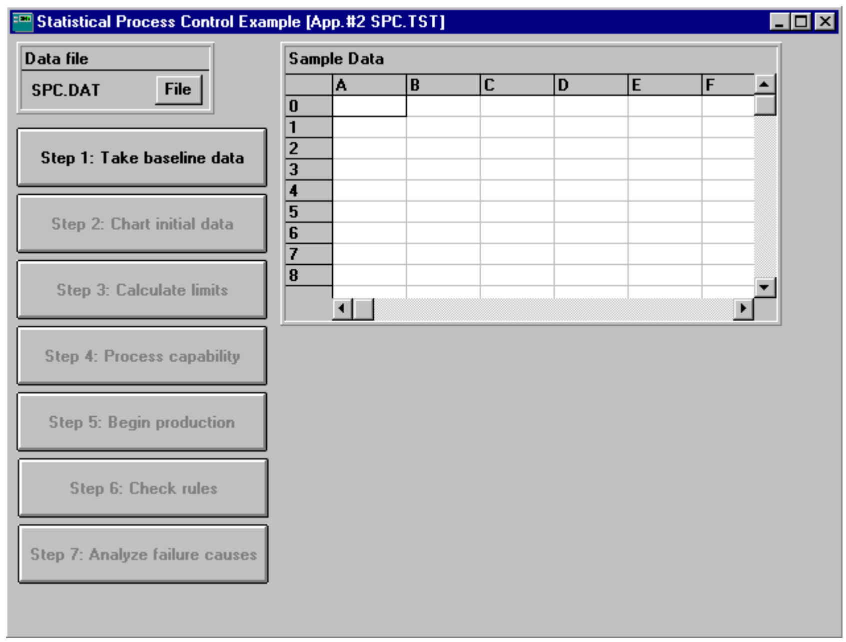
SPC techniques are applied in production, measuring process variables to see how well they conform to specifications. Every process will have some natural variation due to mechanical tolerances, operator interaction, and environmental effects. With statistical methods, these effects can be understood and watched to ensure that the highest possible portion of products meet the desired specifications.

The TestPoint SPC Toolkit adds objects to the TestPoint development system that can calculate SPC values, chart them, and compare against standard patterns and rules.

Chapter 3. An Example

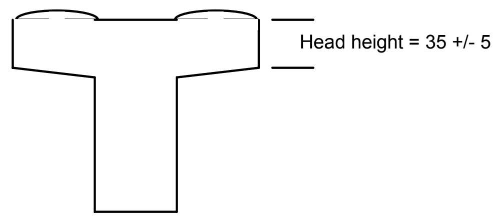
Before diving into the technical terms and details of statistical process control, it is best to get a feeling for the overall usage by looking at a realistic example. This section steps you all the way through all the parts of SPC in a production application: manufacturing steel rivets.

Run TestPoint and open up the SPC.TST example file.



Specifications and Measurement

In this example, we'll be applying SPC to a manufacturing line that makes steel rivets. The design specification is given below:



For simplicity, only one parameter is measured in this example. The unit of measure is not important, as long as it remains consistent. A real production environment might involve measuring dozens of values for each unit.

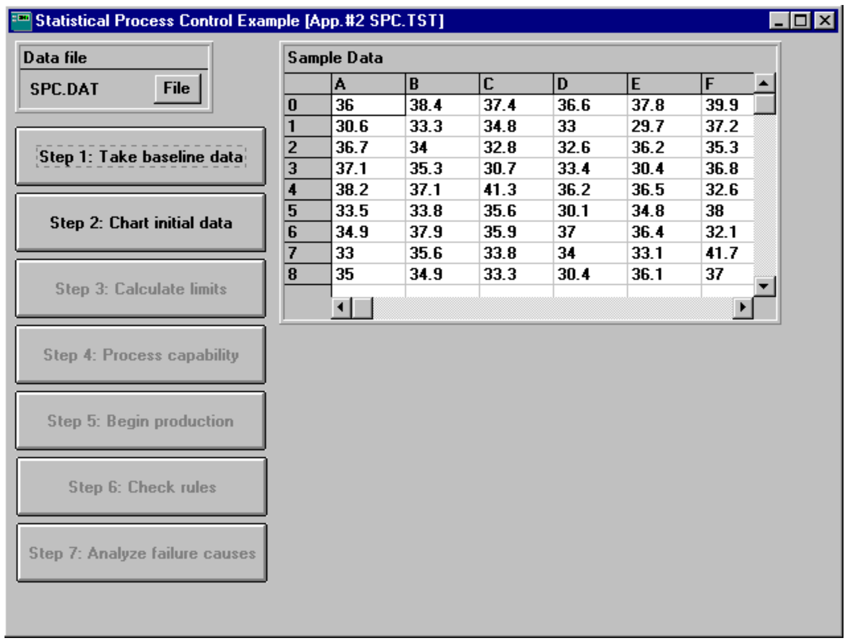
In this example, the data will come from a disk file (SPC.DAT). In an actual production test, the measurement data could be in a file, measured directly by data acquisition hardware plugged into the computer or connected to the computer, or entered manually by an operator.

Baseline Data

We know the specification for the rivet head height, but not the actual variation of the production-line parts. To see this, we need to take an initial set of data samples.
We've chosen to use a sample size of 10 units. This means that every time we want to sample the production line, we take 10 rivets and measure them. Since the production machines are adjusted about every hour, we'll take samples every 30 minutes.

For an initial set of data, 20 samples of 10 units each are taken. This data is in the file SPC.DAT.

Push the button "Take baseline data".



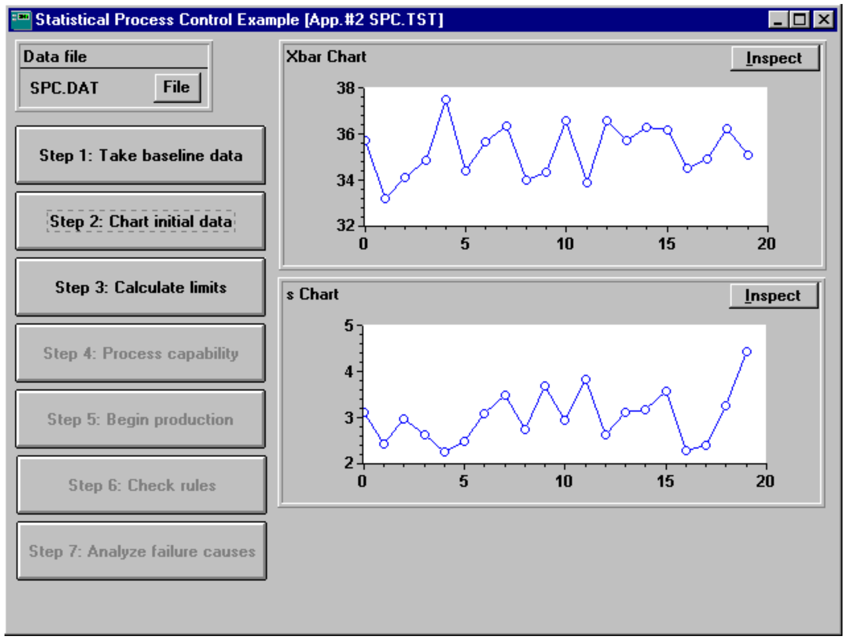
If you look at the TestPoint action lists, you'll see that this button simply reads the data file and displays the information in a Grid object.

Defining Limits

From the baseline data samples, we're going to determine the average head height and expected variation using SPC techniques.

First we do an SPC chart of the data, showing the mean and standard deviation (Xbar and s) of each sample group of 10 units.

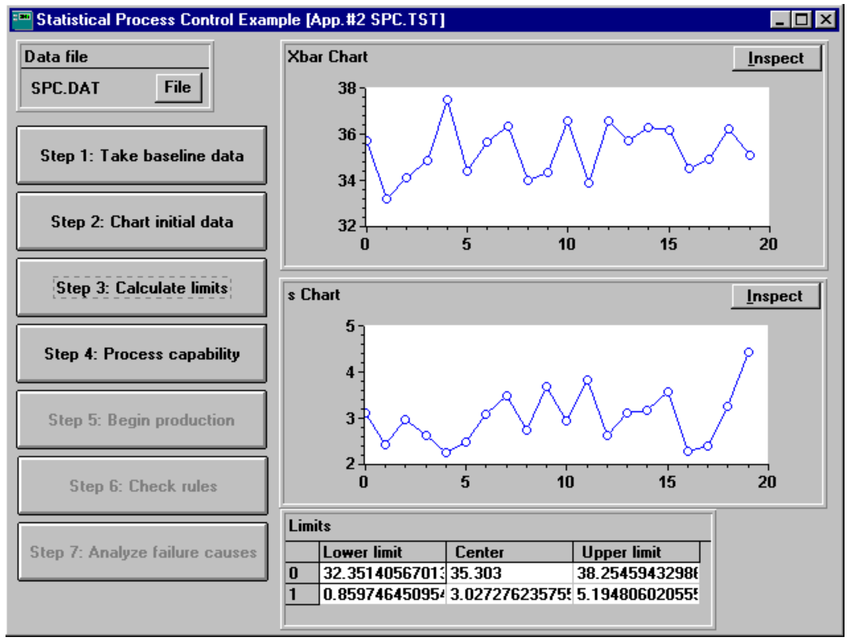
Push the "Chart initial data" button.



This chart is calculated and displayed using two of the SPC toolkit objects.

Next, we use the SPC toolkit to calculate the natural process control limits.

Push the "Calculate limits" button.



The limits show the expected values for the process, out to three sigma (standard deviations).

In this case, the initial data was well controlled and the calculated limits are appropriate for ongoing production usage. With real-world measurements you may find one or more bad data points caused by special situations or measurement errors that you wish to exclude from your baseline calculations. This will be covered in a later section of the manual.

Process Capability

Another useful analysis that can be done on the initial data set (or any set of data samples) is determining the "process capability index".

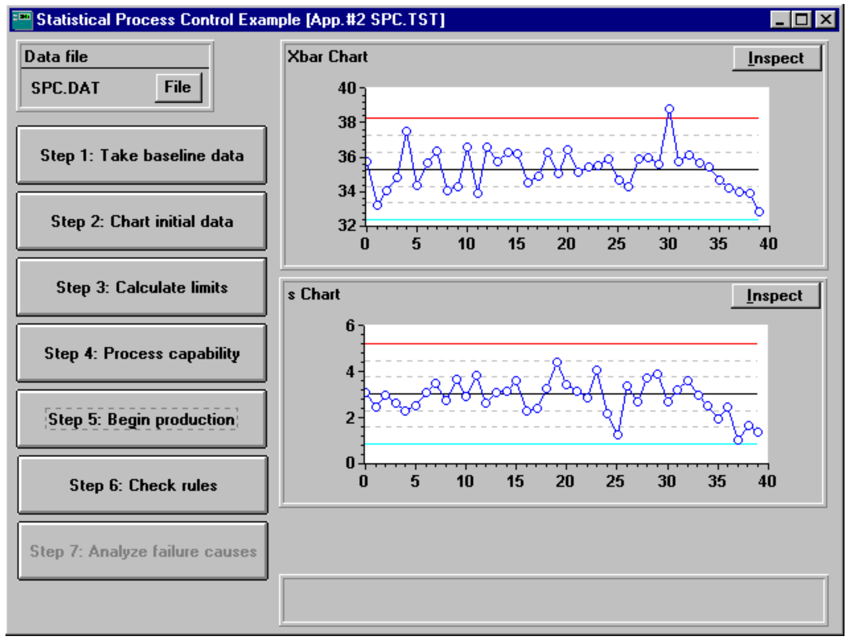
☞ **Push the "Process capability" button.**

This index number indicates how the process is performing as compared to the specification limits set by the designers. Details are given in a later section, but an index of greater than one is generally needed, and many manufacturers look for values in the 4 to 6 range.

Sampling and Charting

Now we're ready to begin using SPC in actual production. We'll take samples of 10 units on a regular basis, charting the results against the limits we've chosen, and watch for process problems.

Push the "Begin production" button.



As each sample is taken, the SPC toolkit objects are used to calculate the mean and standard deviation and add a new point to the control charts.

The sampling will continue to run for a short time, simulating actual production (speeded up for this example). After the example data file is exhausted, the chart updating will stop.

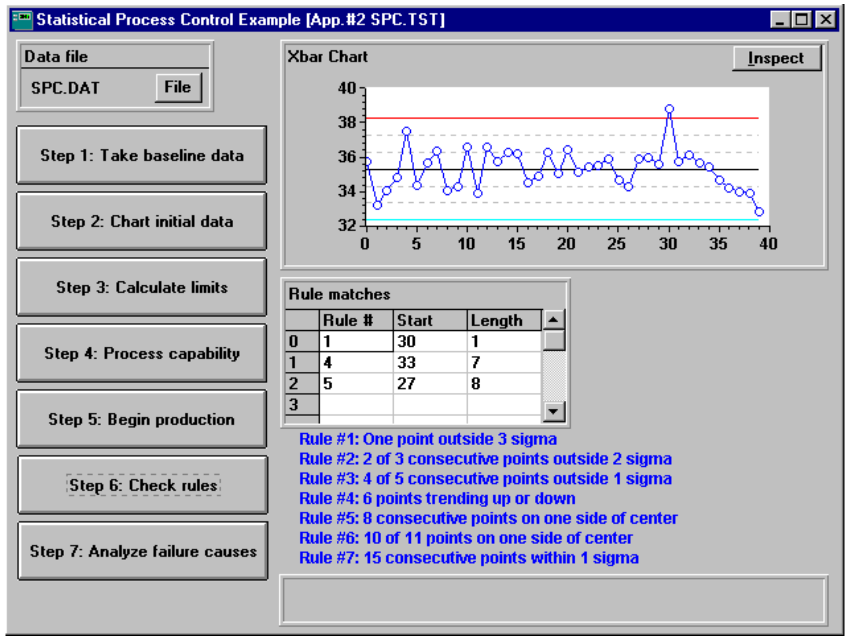
Analysis

The SPC charts are useful in themselves, providing a very understandable indication of the production process. If points appear outside the control limits, there is a problem. If an obvious trend appears, moving upward toward a limit, it may be possible to catch a problem early.

SPC rule checking automates some of this analysis. A number of useful rules for recognizing process problems on SPC charts have been developed at Western Electric and other companies. The SPC toolkit includes an object that checks your actual data against these rules.

☞ **Push the "Check Rules" button.**

If the data sampling has completed and all the information in the SPC.DAT file has been displayed, you should see a number of lines appear in the Rules display.



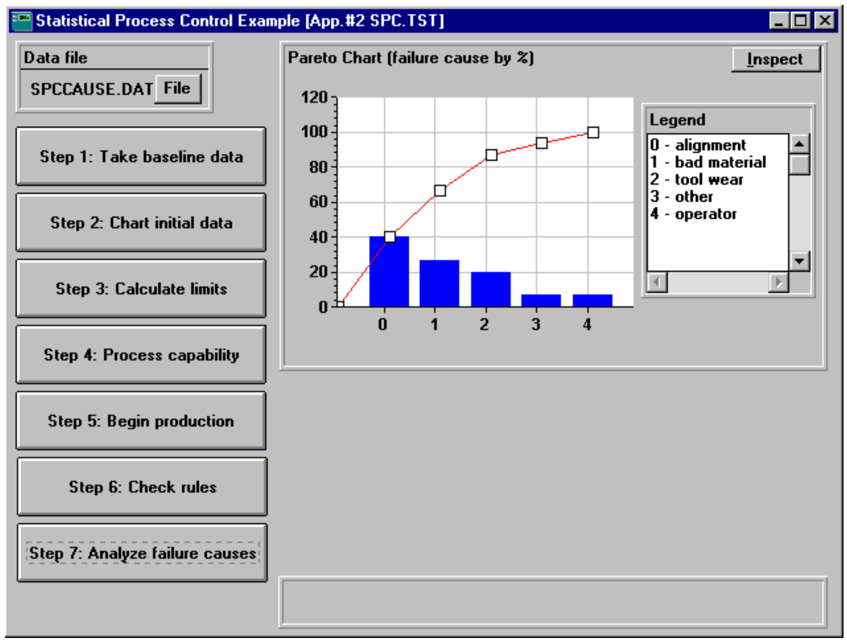
Each line shows the rule number that matched a pattern in the SPC data, the starting index number, and the length of the data run. For example, a downward trend was found (rule number 4), starting at sample number 33 with a length of 7 samples. This might mean that something has happened in the production process leading to a lowering of the average rivet head height over time. Investigating the cause of this change might reveal excessive wear in some part of the manufacturing machinery.

Failures and Causes

When problems do occur, it is important to find out why. Proper analysis of the cause of failures is the first step in correcting the trouble.

A commonly used tool for visualizing the relative importance of failure causes is the Pareto chart.

Push the "Analyze failure causes" button.



The Pareto chart shows the failure causes ranked in order by their frequency of occurrence, giving both the individual values and the cumulative value (the red curve on the plot).

Chapter 4. Charting

Statistical Process Control: Sampling and Terminology

Statistical process control begins with data gathering, known as **sampling**. First, you must define the information or measurements that are important in your particular application. There are two major types of data that can be taken:

Variables are measurements - continuously varying values such as length, weight, chemical concentration, resistance, voltage, frequency, and so on. A variable can be any physical quantity you can measure.

Attributes are counts - the number of items of a particular type such as defective pieces, scratches on a component, broken pins on an IC.

The decisions of what data to gather, how many data items to collect, and how often to collect them are key to effective use of SPC techniques. This requires a combination of knowledge about your process, which can come from production line operators, supervisors, and engineers, and training in SPC methods and sources of statistical or experimental error.

The TestPoint SPC Toolkit allows easy calculation and charting, but the validity of the data and usefulness of your results will depend on judgment and experience in the choice of data and sampling techniques. This manual provides an introduction to SPC, and a thorough documentation of the SPC Toolkit, but should be supplemented by training or the advice of an expert.

In SPC usage, one data value or measurement is known as an **individual** or an **observation**. The entire collection of possible measurements on every production unit is the **population**.

A **sample** is a small portion of the population chosen to represent the whole. On a production line, a sample might consist of a group of 5 parts produced at a particular time, such as 9:00 AM. In many cases, the sample will be more than one observation, so that both the value of the measurement and some idea of its variability can be determined at that point in time.

SPC is most often concerned with a time series of samples. We want to compare the measured results as a process continues to operate, watch for changes or problems, and take corrective action.

Depending on the sample size used, samples are categorized as:

Group data consists of samples with a sample size greater than one. Typical sample sizes range from 2 to 25 observations, depending on what is being measured and the practical issues of taking the data. With group data, both the average value and the variability can be calculated from each sample group.

Group data are used with the following types of SPC calculations and charts: Xbar/R and Xbar/s.

Individual data consists of samples with a sample size of one - that is, single measurement values. Attribute data is always individual data. Individual measurements may also be used for low volume production, or in cases where 100% automated measurement can be made of every item. With individual data, the process variability must be determined by grouping consecutive values to get a "moving" range.

Individual data are used with the following types of SPC calculations and charts: x/mR, mXbar/mR, p, and c.

Group data

Group data in TestPoint is represented as one row per sample, in an **array** or a **list of vectors**:

32.4	31.2	33.1	32.6	30.8
32.5	33.2	31.7	34.4	31.4
31.7	32.8	33.1	30.5	32.9
		⋮		
31.4	33.2	34.0	30.7	32.4

One sample

This data set could be either an array with 5 columns, or a TestPoint list data type containing 5 items, each of which is a vector.

A single sample can also be represented as a **vector** or **list of numbers**. This can be convenient when you wish to add one sample at a time to an SPC chart. You can use a vector or list instead of the more complex data types (array or list of vectors).

Data from files

If your data is in a disk file, just read it in with a File object:

1)	Input from	File	up to 32768 bytes, stop at ___
2)	Calculate Xbar_R	SPC Data	data=File

No special formatting is needed in most cases, since TestPoint will automatically choose a list of vector data type for files containing rows of numbers separated by commas, spaces, or tabs.
There is an example of this in **SPC.TST**, in the action list for button #1.

Measuring data samples

If you are measuring data in TestPoint using data acquisition hardware or instruments, so that your data values are available one individual measurement at a time, you can easily combine the values using a Math object to build a list of numbers. For example:

```
1)  Clear              DataContainer
2)  Linear series      DataLoop        from 1 to 20, step by 1
3)    Set              Sample          to ____
4)    Linear series    SampleLoop      from 1 to 5, step by 1
5)      Input from     Meter           up to 256 bytes, stop on CR
6)      Calculate      Sample          with newvalue=Meter
7)    End              SampleLoop
8)    Append to        DataContainer from Sample
9)  End                DataLoop
10) Calculate Xbar_R   SPC Data        data=DataContainer
```

where the formula for the Math object **Sample** is:

```
list( previous(), newvalue )
```

Lines 3 to 7 build a single sample, consisting of 5 measurements (a sample size of 5). First, the sample is initialized in line 3 to blank (NoData). Then five values are taken and added to the sample. Each "calculate" action adds a value to the **list** data in the math object.

The outer loop in the example above gathers 20 samples and puts them in a Container object, using the Append action. Your application might have a fixed loop like this, or a Timer taking samples at a particular interval, or even a pushbutton to manually trigger each sample.

One sample at a time

In some applications, samples may be processed one at a time. For example, if an SPC chart is being updated in real time while a process is occurring, each sample may be added to a strip chart as the sample is measured.

1)	Clear	Container	
2)	Linear series	SampleLoop	from 1 to 5, step by 1
3)	Input from	Meter	up to 256 bytes, stop on CR
4)	Append to	Container	from Meter
5)	End	SampleLoop	
6)	Calculate Xbar_R	SPC Data	data=Container
7)	Add points to	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0

Lines 1 to 5 create a single sample of 5 values. A Container object and the Append action are used, resulting in a **vector** data type, which is assumed by the SPC objects to represent a single sample when group data calculations such as "Xbar_R" are used.

Individual data

Individual data in TestPoint is represented as a **vector of numbers**, one value per measurement. A single number can also be used when processing samples one at a time (as in updating a strip chart).

Summary

Group data (for Xbar/R and Xbar/s):

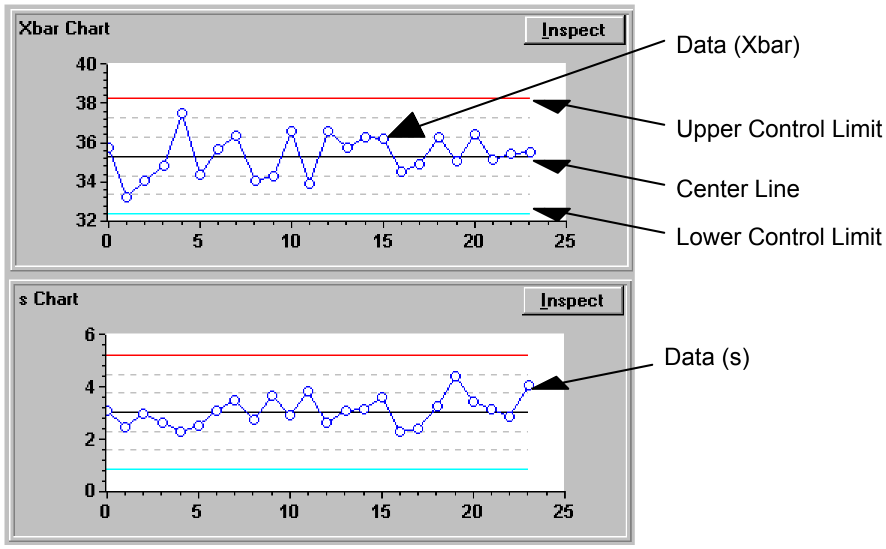
list of vectors	each row is a sample, # items = sample size
array	each row is a sample, # cols = sample size
vector	one sample, dim(vector) = sample size
list of numbers	one sample, dim(list) = sample size
all other types	not allowed

Individual data (x/mR, mXbar/mR, p, and c):

vector	multiple samples, dim(vector) = # samples
number	one sample
string	(automatically converted to a number)
all other types	not allowed

Charts and Limits

There are many chart types that can be used in statistical process control, depending on the type of data being measured. These types will be covered in the next few sections of the manual. In general, though, most of these calculations produce two charts, one to show the data values (or averages), another to indicate the amount of variation. For example:



A change in either the actual values being measured or in the amount of variability can signal a problem that must be studied and corrected, so both charts are important. In fact, if the variability chart is not "in control", meaning that it is not centered and normally random, then the value chart is not reliable for any analysis.

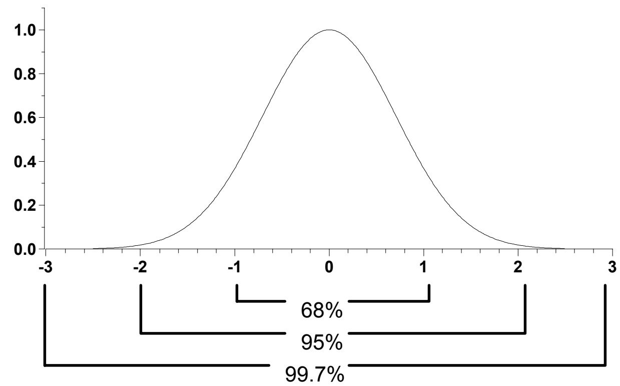
Limit lines are used on the charts to indicate the expected range of values. Anything outside the limits, especially multiple values, indicates a possible problem.

For some SPC calculations, such as monitoring attribute data, a single chart may be used.

Limit Calculation and Data Selection

Limits are generally set by measuring the natural variation of the process, using an initial run of data samples. Statistical calculations on this data set are used to estimate the **sigma**, or standard deviation, of the entire population of values.

Limits are then set at the center value plus and minus 3 sigma. For a process with a standard normal probability distribution, almost all measurements (99.7%) will be within these limits. In fact, about 2/3 of the measured values will be within plus or minus one sigma.



With the TestPoint SPC Toolkit, you calculate the limits by using the **SPC Limits** object:

1) Calculate Xbar_R SPC Limits data=File, start_index=0, end_index=9999, exclude=___

Your initial data set can be stored in any TestPoint object: a File, a Container, etc., as long as it follow the data type rules mentioned previously. The calculation can be any of the SPC chart types. In this example, Xbar/R (mean and range) is used.

After calculating these limits, the initial data set should be processed and charted for review, using the **SPC Data** and **SPC Chart** objects:

2)	Calculate Xbar_R	SPC Data	data=File
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0
4)	Draw	SPC Chart	graph=Graph2, data=SPC Data, limits=SPC Limits, value index=1

Both charts have been drawn in this case - the Xbar chart in line 3 and the R chart in line 4 (the **value index** parameter selects which statistic is to be drawn).

The charts must be inspected to make sure they are **in control**. If many points are outside the control limits or there is some other obvious non-randomness, the cause should be determined and new data obtained.

In some cases, however, only one or two samples are seen to be unusual when this initial analysis is done. If enough other samples are remaining to provide sufficient basis for further charting, then the "bad" samples can just be excluded and the limits recalculated. This is done using the other parameters on the SPC Limit object's action line. For example:

1)	Calculate Xbar_R	SPC Limits	data=File, start_index=0, end_index=9999, exclude=Grid1

In this case, the start and end indices remain at 0 and 9999 (larger than the number of samples actually provided in the data), but the **exclude** parameter is provided by a TestPoint Grid object, into which the operator can enter one or more sample numbers. These samples will not be included in the limit calculation.

A good example of limit calculations can be found in **SPCLIMIT.TST**.

Group Measurement Charts

For group data (sample size greater than one), the TestPoint SPC Toolkit provides the following chart types in the SPC Data and SPC Limits objects:

Xbar and R

Xbar is the mean, or average, of the values in each sample. R is the range, or difference between the largest and smallest value in the sample set. R provides an indication of the variation of the measurement. Xbar/R charts are most often used with fairly small sample sizes, and were particularly popular when SPC calculations were carried out manually, due to the simplicity of the arithmetic involved.

1)	Calculate Xbar_R	SPC Data	data=File1
2)	Calculate Xbar_R	SPC Limits	data=File1, start_index=0, end_index=9999, exclude=___
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0
4)	Draw	SPC Chart	graph=Graph2, data=SPC Data, limits=SPC Limits, value index=1

Xbar and s

Xbar is the mean, or average, of the values in each sample. s is the standard deviation of the values in the sample set. Xbar/s charts are most often used with sample sizes of 10 or more values.

1)	Calculate Xbar_s	SPC Data	data=File1
2)	Calculate Xbar_s	SPC Limits	data=File1, start_index=0, end_index=9999, exclude=___
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0
4)	Draw	SPC Chart	graph=Graph2, data=SPC Data, limits=SPC Limits, value index=1

Individual Measurement Charts

For individual data (sample size of one), the TestPoint SPC Toolkit provides the following chart types in the SPC Data and SPC Limits objects:

X and mR

X is simply the input values, unmodified. mR is a range, determined from the most recent N values, where N may be chosen through a setting value of the SPC object.

For example, if N=3 and the most recent 3 input values are:

4.5, 5.2, 4.8

then mR will be (5.2-4.5), or 0.7.

Since the moving range calculation requires at least N values, no results are calculated from the first N-1 input data samples.

1)	Calculate X_mR	SPC Data	data=File1
2)	Calculate X_mR	SPC Limits	data=File1, start_index=0, end_index=9999, exclude=___
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0
4)	Draw	SPC Chart	graph=Graph2, data=SPC Data, limits=SPC Limits, value index=1

mXbar and mR

This chart is very similar to the X and mR chart. In this chart, mXbar, a moving average, is substituted for X. mXbar is calculated from the most recent N samples, just like mR.

1)	Calculate mXbar_mR	SPC Data	data=File1
2)	Calculate mXbar_mR	SPC Limits	data=File1, start_index=0, end_index=9999, exclude=___
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0
4)	Draw	SPC Chart	graph=Graph2, data=SPC Data, limits=SPC Limits, value index=1

Attribute Data Charts

When monitoring attribute data (integer counts of occurrences), the TestPoint SPC Toolkit provides the following chart types. Attribute calculations produce only one chart (no variability chart corresponding to the range or s charts in the other types).

p (fraction nonconforming)

The p chart is used when the data consists of a count of bad items out of a known lot size. For example, out of 100 units from a production line, a count of 8 faulty units would result in a fraction nonconforming of 8/100, or 0.08.

1)	Calculate p	SPC Data	data=File1, sample size=100
2)	Calculate p	SPC Limits	data=File1, sample size=100, start_index=0, end_index=9999, exclude=___
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0

Note that the sample, or lot, size must be provided in these calculations (lines 1 and 2). Note also that only one chart is drawn (with value index zero).

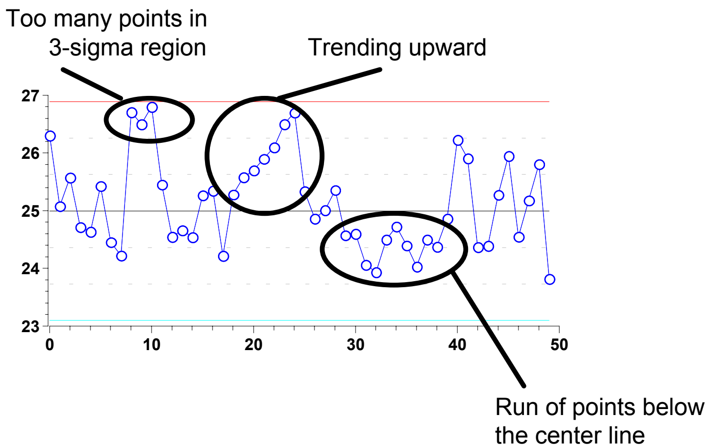
c (count of defects)

The defect count chart is simply a graph of the input data, with control limits and other SPC calculations applied. The input data in this case consists of counts of some item, normally defects in some item, like scratches on a piece of furniture.

1)	Calculate c	SPC Data	data=File1
2)	Calculate c	SPC Limits	data=File1, start_index=0, end_index=9999, exclude=___
3)	Draw	SPC Chart	graph=Graph1, data=SPC Data, limits=SPC Limits, value index=0

Chapter 5. Chart Analysis: Rules & Patterns

Once an SPC chart has been constructed, it can be used to look for patterns and problems in the process data.



The chart above shows a number of patterns that could indicate trouble and would warrant investigation. For example, the upward trend in the middle of the chart might indicate that the average value of the process is increasing, perhaps due to tool wear.

Over the years, a number of rules have been developed at companies like AT&T and Boeing for use with SPC charts. These rules are derived from statistical principles. For example, given a normal distribution, it is expected that 95% of the values will fall within two sigma of the average. From the normal curve, the probability of getting a run with two or more points out of three outside this range can be determined. Since this chance is low enough to be statistically meaningful, a rule is created: "If 2 out of 3 points fall outside 2 sigma, signal a possible problem".

All the SPC rule checking is based on probability. A rule violation does not tell you with 100% certainty that there is a problem. Nor does it give any indication as to what is wrong. What it **will** do for you is alert you to look at the process before any problem gets out of control.

Listed below are the rules implemented in the TestPoint SPC Toolkit.

One point outside 3 sigma

2 out of 3 points outside 2 sigma

4 out of 5 points outside 1 sigma

These rules all fall in the category of **limit violations**. In each case, SPC chart points are too far from the expected center line, and a problem may exist.

The number of points required for a rule violation to be indicated varies with the distance from the center line (1 sigma, 2 sigma, or 3 sigma). Since only 68% of values are expected to fall within 1 standard deviation, it is not unusual to see one or two points in a row outside 1 sigma. However, when 4 out of 5 points are this far away, it is unusual enough to warrant investigation.

6 points trending up or down

This rule is designed to detect **process shift**. If a consistent trend is found in either direction, the average may be changing from the originally defined center line.

8 consecutive points on one side of the center line

This rule also detects process shift. Even though the values may not be going continuously downward or upward, if they are all on one side of center, it is still statistically unusual.

M out of N points on one side of the center line

This is still another process shift detector. You can define the parameters M and N yourself to change the threshold of detection. In this rule, the points do not need to be consecutive points on one side of center, but merely some defined proportion of points.

Typical values for M and N that are commonly used are:

10 out of 11

12 out of 14

14 out of 17

16 out of 20

15 consecutive points within 1 sigma

This rule detects **decrease in variation**. It is more an indicator of improvement than of a problem. This many points in a row with none outside 1 standard deviation is statistically unusual, and may indicate a change in the process. While a change in the direction of less variation is usually considered good, it is still unexpected and may be worth investigating.

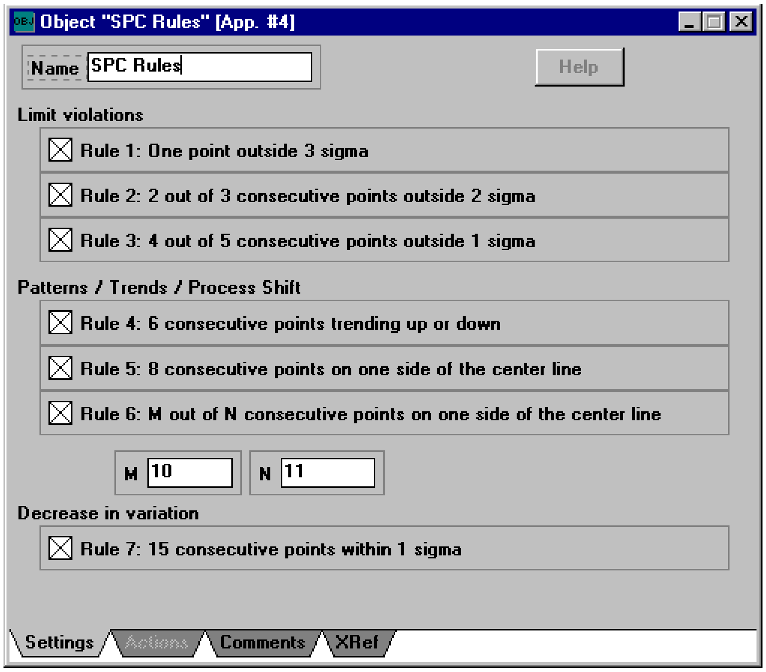
Using Rule Checking in TestPoint

To do rule checking, just use the Check action of the SPC Rules object:

1)	Check	SPC Rules	data=SPC data, limits=SPC Limits, value index=0

The check action uses the given data and limits and tests one or more of the rules listed earlier. The value index parameter indicates which SPC value to test for those charts with two values (such as Xbar and R, for example).

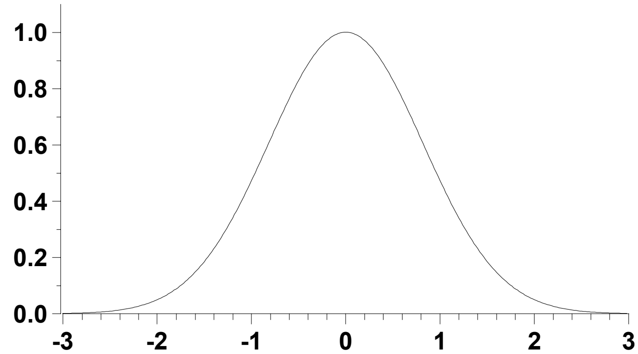
The rules to be tested are selected through settings of the SPC Rules object:



Chapter 6. Histograms and Statistics

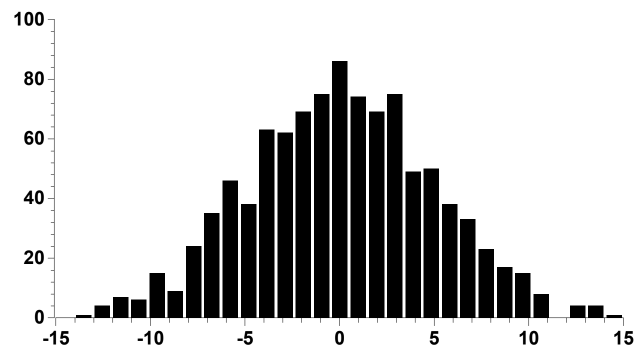
Distribution of Data: Histograms

Throughout much of this manual, it has been assumed that the measured process data will be randomly distributed on a standard normal curve:



While this is true for most natural processes, it is an assumption that may be worth testing for your particular data.

A **histogram** is a graph that shows the distribution of values in a data set. It is constructed by dividing the range of values into a number of equal-sized subranges, or "bins", then counting the number of values that fall into each bin. These counts are charted on a bar chart, looking something like this:



To draw a histogram in TestPoint, you make use of the Math object and Graph object. The Math object's formula should be:

```
histogram( data, Nbins )
```

and your action list is:

1)	Calculate	Histogram	with data=File1, Nbins=30
2)	Draw graph	Graph1	with Histogram

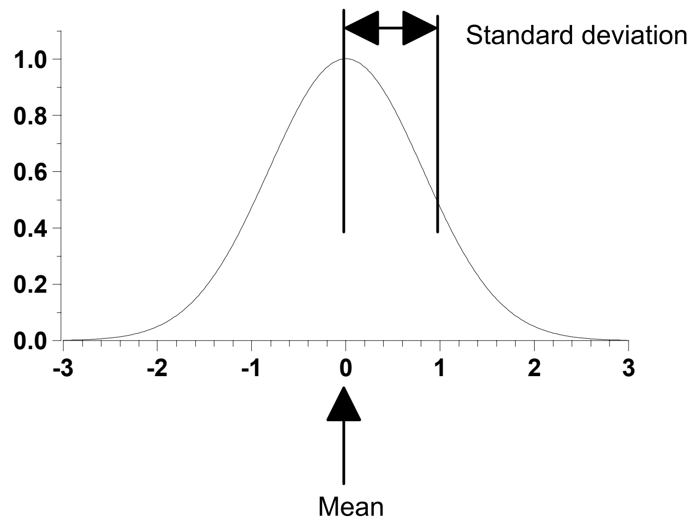
The Graph object should be set to X vs Y mode, and Bar chart mode. The math formula histogram() returns both X and Y values. For X, the location of each bin on the X axis is used, and for Y, the count of data values in that bin is returned.

Statistical Measures

There are a number of statistical functions that can give you a measure of your data set's distribution. These can all be calculated using the SPC Stats object.

Mean and Standard Deviation

Mean, or arithmetic average, is a measure of the central value of a set of values. Standard deviation measures the variability of the values.



Calculate these using the SPC Stats object:

1) Calculate Mean SPC Stats data=File1
2) Calculate StdDev SPC Stats data=File1

Median

The median is another measure of the center of a data set. The median value is found by sorting the data in order from lowest to highest and taking the number in the middle of the list (for even-length list, the middle two numbers are averaged).

Median differs from mean when the distribution of values does not perfectly match an even-sided normal curve.

Calculate the median using the SPC Stats object:

1) Calculate Median SPC Stats data=File1

Skewness

When a distribution is not evenly balanced on both sides of the mean value, it is said to be skewed. Skewness is a calculated measure of the amount of unevenness. The skewness of a normal distribution is zero.

Calculate the skewness using the SPC Stats object:

1) Calculate Skewness SPC Stats data=File1

Kurtosis

Kurtosis measures the "flatness" of the distribution, or how quickly it falls off. The kurtosis of a normal distribution is 3.

Calculate kurtosis using the SPC Stats object:

1) Calculate Kurtosis SPC Stats data=File1

Chapter 7. Process Capability Analysis

Specifications and Capability

SPC charts are drawn to show actual process variables, along with control limits. Control limits are normally set to the natural limits derived from the actual measured data. These are the +/- 3 sigma points on the normal distribution, containing 99.5% of the expected measurement results.

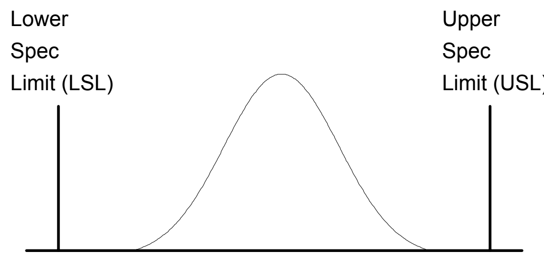
However, any process, component, or engineering design will also have designed limits, known as **specification limits**. These are set by the requirements of the product, and not the same thing as the measured distribution, or natural limits.

Comparing these two types of limits is very important. If the natural limits vary much more widely than the specification allows, many of the items produced will not meet specifications, and will have to be rejected.

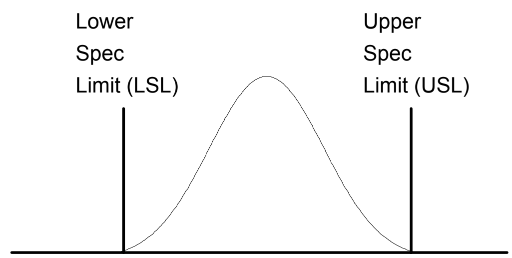
Various measures of **process capability** have been defined which indicate how well the natural process variation from the measured data conforms to the desired specification limits.

Cp

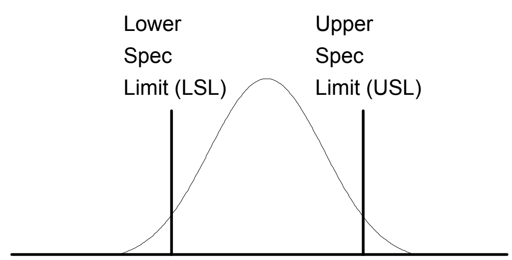
Cp is the **process capability index**. It is defined as the ratio of the specification limit range to the 6-sigma natural variation range. When this ratio is one, the natural limits exactly match the 6-sigma (99.5%) variation. A ratio of more than one indicates better performance, with even less of the distribution outside the specified limits.



Cp > 1.0



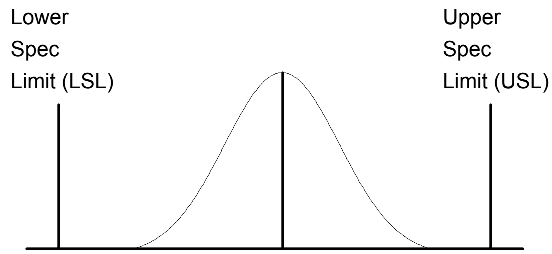
Cp = 1.0



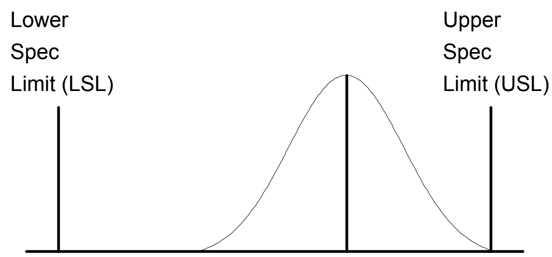
Cp < 1.0

Cpk

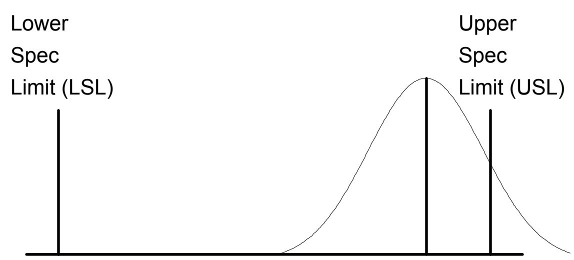
Cpk is the **centered process capability index**. It takes into account the possibility that your measured process distribution may not be centered within your specified limits. If the average process value is not exactly in the center, Cp alone does not accurately reflect the ability of your process to produce good results. Cpk calculates the index separately for the low and high limits, and uses the lesser of the two values (the worst case result).



Cp =1.5, Cpk = 1.5



Cp =1.5, Cpk = 1.0

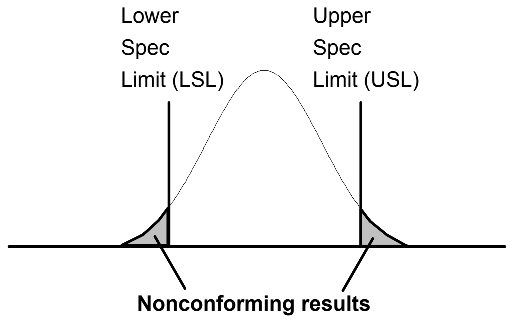


Cp =1.5, Cpk =0.5

Predicted nonconformance rate

Related to the Cp and Cpk measures is the prediction of the fraction or percentage of results that will fail to meet the specifications. Assuming the process follows a normal distribution curve, the portion of the curve that falls outside the limits can be calculated.

For example, when Cpk=1.0 and the specification limits just match the 6 sigma variation of the process, we have already seen that about 99.5% of the values fall within this range. So 0.5% can be expected to be nonconforming.



Calculating Process Capability

Calculating the process capability measures using the TestPoint SPC Toolkit is simply a matter of using the Process Capability object:

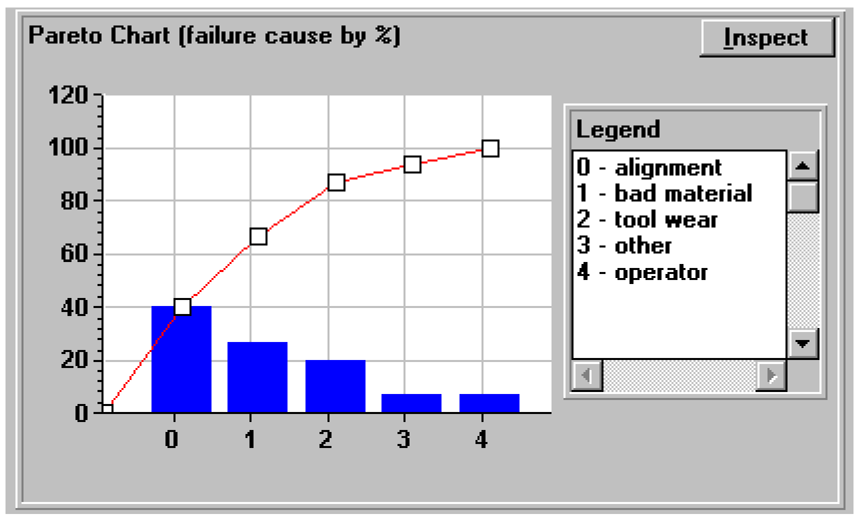
1) Calculate Cp — Process Capability data=File1, upper spec=45, lower spec=35
2) Calculate Cpk — Process Capability data=File1, upper spec=45, lower spec=35
3) Calculate PPM nonconforming — Process Capability data=File1, upper spec=45, lower spec=35

Any data set can be given to this object for the calculation - it is always converted to a simple vector of numeric values for processing. The data can come from a File object, Grid object, math, Container, or any other suitable data source in TestPoint.

The specification limits can be constants on the action line, or can come from entry fields on a panel or any other desired source.

Note that the nonconformance calculation returns its results in parts per million (PPM).

Chapter 8. Cause Analysis: Pareto Charts



Assignable Cause

SPC charts and statistics are suitable for monitoring and flagging the presence of problems in a process. Solving these problems, however, requires determining the **assignable cause**. Many things can cause variation in measured results. These can be actual failures and mistakes, or they can be unavoidable randomness or accuracy limits in measurement. Assignable causes of variation are those causes that can be tracked to a specific condition that can be affected by some action.

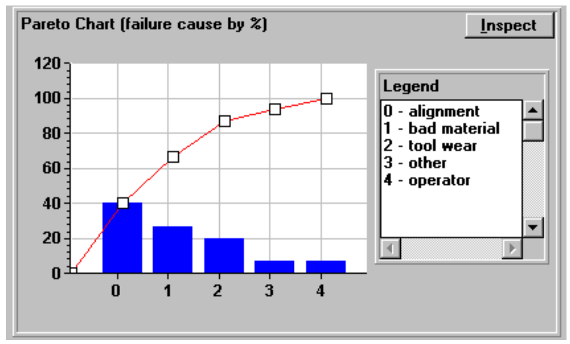
A list of causes that are appropriate for a particular process can come only from expert knowledge of that process. You may find this knowledge is available from engineers who designed the process, from production line workers who interact with it every day, or a consultant brought in specially to study it.

Once a list of **cause codes** has been decided upon, categorizing the possible assignable causes into a small number of identifiable classes, failures in the process can be studied. Every time a failure or bad result occurs, it should be classified and a cause code assigned and noted. When a sufficient number of failures have been accumulated, the data can be analyzed.

Drawing Pareto Charts

Failure causes are often analyzed with the use of a tool called Pareto charts, named after the economist Vilfredo Pareto, who was an early advocate of what is now commonly known as the 80/20 rule. This is based on the observation that for many phenomena that can be grouped in categories, 80% of the population is in 20% of the categories. Pareto made his initial observations about the distribution of economic wealth, but it applies very well to causes of process failure.

To determine the most important causes, the cause data that is gathered must be grouped and sorted to show the most important contributing causes, and the relative importance of these. This is done in the Pareto chart:



The cause codes, in this case "alignment", "bad material", etc. are grouped and graphed by their relative frequency of occurrence. The original measured data in this case is a long file of these key words, in random-looking order, but with "alignment" occurring most frequently, at 40% of the total.

A chart like this points out clearly which areas are most effective to address with corrective measures.

Pareto Charts in TestPoint

Drawing a Pareto chart using the TestPoint SPC Toolkit is done with the Pareto Chart object:

1)	Draw	Pareto Chart	graph=Graph1, legend-text=Text1, cause strings=File1, mode="percentage"

This action takes a graph object, data-entry (text) object, and cause strings in the form of a string vector, and draws the Pareto chart. It sorts and summarizes the cause code strings, derives the percentages, and graphs the results.

The graph and data-entry objects must be set up with appropriate settings, line types, and so forth. This is detailed further in the Reference chapter of the manual. One good way to get graph and legend text objects suitable for your application is to copy them from SPC.TST and paste them into your program.

Chapter 9. Reference

SPC Data Object

The SPC Data Object computes any of the basic process control statistics from sample data.

Data can be given to the SPC Data object in blocks of multiple samples, such as from a data file, or it can be calculated one sample at a time, as in real-time measured data from a production line.

Actions

Calculate Xbar_R - Given sample data (in groups with sample size greater than one), compute Xbar and R for each sample. Xbar is the mean, or average, of each sample set. R is the range of values in each sample set. Input data must conform to the data type rules given at the end of the reference section. The result, given as the new data value of the SPC Data object, is a LIST containing Xbar value(s) in the first item, R value(s) in the second item.

Calculate Xbar_s - Given sample data (in groups with sample size greater than one), compute Xbar and s for each sample. Xbar is the mean, or average, of each sample set. s is the standard deviation of values in each sample set. Input data must conform to the data type rules given at the end of the reference section. The result, given as the new data value of the SPC Data object, is a LIST containing Xbar value(s) in the first item, s value(s) in the second item.

The standard deviation is calculated according to the formula:

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{samplesize - 1}}$$

Calculate X_mR - Given sample data (individual measurements with sample size of one), compute X and a moving range for each sample. X is simply a copy of the input data samples. mR is the range of values of the most recent N samples, where N is a setting of the SPC Data object. Input data must conform to the data type rules given at the end of the reference section. The result, given as the new data value of the SPC Data object, is a LIST containing X value(s) in the first item, mR value(s) in the second item.

Calculate mXbar_mR - Given sample data (individual measurements with sample size of one), compute mXbar and a moving range for each sample. mXbar is the mean, or average, of the most recent N samples, where N is a setting of the SPC Data object.. mR is the range of values of the most recent N samples. Input data must conform to the data type rules given at the end of the reference section. The result, given as the new data value of the SPC Data object, is a LIST containing mXbar value(s) in the first item, mR value(s) in the second item.

Calculate p - Given sample data, each of which is a measure of the number of defective items in a sample, compute p, the fraction of defective units. p is calculated by dividing the number of defective items by the given sample size. p ranges from 0 to 1. Input data must conform to the data type rules given at the end of the reference section. The result, given as the new data value of the SPC Data object, is a VECTOR containing values of p.

Calculate c - Given sample data, each of which is a measure of the number of defective items in a sample, compute c, the defect count. c is simply a copy of the input, put into VECTOR form. The result, given as the new data value of the SPC Data object, is a VECTOR containing values of c.

Reset - Resets the internal sample history maintained by the SPC Data object. This history is used in moving average and moving range calculations (X and mR, or mXbar and mR).

Data

The data value of the object is the result of the most recent Calculate action performed. The data type varies with the particular calculation chosen.

Settings

# of samples for moving range/avg. - sets the number of samples used in calculating mXbar and mR values. After a Reset action (or when the application first begins), no mXbar or mR values are returned as result data until at least this given number of samples of input data have been provided. The input samples can be provided all at once in one Calculate action or in a sequence of separate actions. The SPC Data object keeps an internal history of the most recent sample data to allow for moving range and average calculations.

SPC Limits Object

The SPC Limits object calculates natural process limits from a baseline set of sample data. The natural variation is measured statistically and an estimate of the overall process mean and variation is produced. The limits are placed at the estimated +/-3 sigma points.

The limit calculations allow specification of a subset of the input data through both starting and ending indices and also a vector of indices to be excluded from the calculation. This feature is provided to allow a person studying the results to eliminate data that is considered to be invalid measurements or unusual values.

For example, if the input data contains 50 samples, and the start index is given as 10, the end index as 44, and the exclude values is a vector containing:

12, 23, 24, 25

then the sample indices actually used in calculating the limits will be:

10,11, 13,14,15,...,21,22, 26,27,28,...,42,43,44

Actions

Calculate Xbar_R - Given sample data (in groups with sample size greater than one), compute Xbar and R limits. Starting and ending indices to select a subset of the input data, and an optional vector of sample index numbers to exclude may be specified.

Calculate Xbar_s - Given sample data (in groups with sample size greater than one), compute Xbar and s limits. Starting and ending indices to select a subset of the input data, and an optional vector of sample index numbers to exclude may be specified.
Calculate X_mR - Given sample data, compute X and mR limits. Starting and ending indices to select a subset of the input data, and

an optional vector of sample index numbers to exclude may be specified.

Calculate mXbar_mR - Given sample data, compute mXbar and mR limits. Starting and ending indices to select a subset of the input data, and an optional vector of sample index numbers to exclude may be specified.

Calculate p - Given sample data, each of which is a measure of the number of defective items in a sample, compute limits for p, the fraction of nonconforming parts. Starting and ending indices to select a subset of the input data, and an optional vector of sample index numbers to exclude may be specified.

Calculate c - Given sample data, each of which is a measure of the number of defective items in a sample, compute limits for c, the defect count. Starting and ending indices to select a subset of the input data, and an optional vector of sample index numbers to exclude may be specified.

Data

The data value of the Limits object is a LIST data type. The first item gives the lower control limit (LCL). The second item is the center line (CL). The final item is the upper control limit (UCL).

For calculation types that consist of two parameters (Xbar/R, Xbar/s, X/mR, mXbar/mR), each item in the list is actually a vector containing the limits for the two results. For the attribute data calculations that have a single result (p and c), the list items are simple numbers.

Settings

# of samples for moving range/avg. - sets the number of samples used in calculating mXbar and mR values. The SPC Limits object, unlike the SPC Data object, does **not** keep a history of input data. Each limit calculation is separate and is intended to operate on a complete baseline data set.

SPC Chart Object

The SPC Chart object draws SPC data onto a Graph object.

Actions

Draw - Draws the result of an SPC Data calculation onto a graph object. Action line parameters specify the graph object, the data, the limits (optional), and which data value is to be plotted. Most of the SPC calculations produce two results (like Xbar and R). The **value index** parameter specifies which is to be shown (0 for the first value, 1 for the next). For single value calculations (p and c), value index must be zero.

The data to be plotted comes from an SPC Data object. The limits information usually will come from an SPC Limits object, but could also be in a File object (if the limits are saved in a data file) or some other object such as a Grid for user entry of limits. The limit information must be the correct data type (see the SPC Limits object).

Add Points - Similar to Draw, but appends additional points to an existing graph. This action is used to update a graph as new samples are measured in an ongoing process.

Data - has no data value

Settings

Show zones (A,B,C) - Indicates whether the graph should include lines for the control zones, typically called zones A, B, and C. These zones divide the area between the center line (CL) and the control limits into three parts to aid in recognizing patterns in the data.

SPC Rules Object

The SPC Rules object tests the results of an SPC calculation for commonly recognized patterns and trends. This can be useful in identifying process problems or improvements.

Actions

Check - Test the given data against the rules selected in the object settings. Any matches to the rules are placed in the result data (the new data value of the Rules object).

Data

The data value is a LIST containing three items (each of which can be a number or a vector containing multiple numbers, if multiple rule pattern matches are found). The first item is the rule number. The second item is the starting index of the matching data. The final item is the length of the matching data run.

It is often most useful to display the result data in a Grid object, like this:

Rule Matches

	Rule #	Start Point	Length
0	2	1	3
1	2	5	3
2	3	0	9
3			
4			
5			
6			

Settings

Rule 1: one point outside 3 sigma
Rule 2: 2 out of 3 consecutive points outside 2 sigma
Rule 3: 4 out of 5 consecutive points outside 1 sigma
Rule 4: 6 consecutive points trending up or down
Rule 5: 8 consecutive points on one side of the center line
Rule 6: M out of N points on one side of the center line
Rule 7: 15 consecutive points within 1 sigma

Each checkbox setting indicates whether the corresponding rule should be applied to the input data when the Check action is executed.

M and N - the parameter values used in applying rule number 6.

SPC Statistics Object SPC Stats

This object can derive a variety of standard statistical measures from a data set. The data can be any TestPoint data type - it will always be converted to a single vector for purposes of calculation.

Actions

Calculate Median - take the median of the input data. The median is defined as the "middle" value, found by arranging the values in increasing order and choosing the center value in this sorted list. (If the number of values provided is even, the middle two values are averaged).

Calculate Mean - take the mean, or arithmetic average, of the input data. This is calculated as the sum of the values divided by the number of values.

Calculate StdDev - take the standard deviation (a measure of variance) of the input data. The formula is:

$$\sqrt{\frac{\sum (x-\overline{x})^2}{n-1}}$$

Calculate Skewness - calculates a measure of the skew of the input data, or its deviation from an even-sided normal distribution curve. The skew of a perfect normal curve is zero. Skew is defined as:

$$\frac{\sum (x-\overline{x})^3}{n \bullet stddev^3}$$

Calculate Kurtosis - calculates kurtosis, a measure of the "flatness" of the input distribution curve. The kurtosis of a perfect normal curve is 3. Kurtosis is defined as:

$$\frac{\sum (x - \bar{x})^4}{n \bullet stddev^4}$$

Data

The data value of the object is the result of the most recent calculation action.

Settings

none

Process Capability Object

This object calculates the process capability index from a given set of measured data and specification limits for that data.

Process capability measures how well the process can produce results within the specifications. A capability index of one means that the normal 6 sigma variation of the process exactly matches the specified limits. An index greater than one is better. Many companies use an index of 4 to 6 as the goal for their processes.

Actions

Calculate Cp - calculate the basic process capability index. The sigma, or population standard deviation, is approximated by calculating the standard deviation of the input data values. Cp is calculated as:

$$\frac{upperspec - lowerspec}{6s}$$

Calculate Cpk - calculate the alternative process capability index Cpk, which is used when the process may not be producing results exactly centered between the specification limits. This index calculates the upper and lower halves of the index independently and uses the lesser of the two values. It is calculated as:

$$\min\left(\frac{upperspec - \overline{x}}{3s}, \frac{\overline{x} - lowerspec}{3s}\right)$$

Calculate PPM nonconforming - calculate the portion of results expected to fall outside the specification limits, in parts per million. This is found by calculating the area under the normal distribution curve that falls past the specification limits.

Data

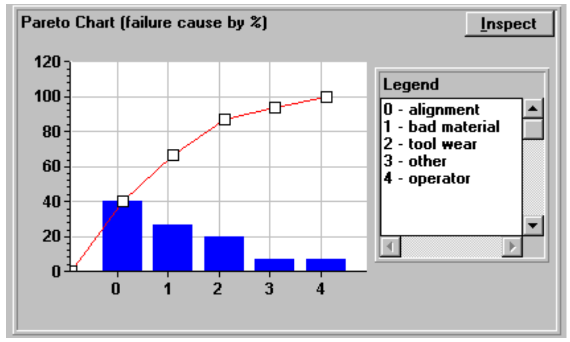
The data value of the object is the result of the most recent calculation action.

Settings

none

Pareto Chart Object

This object is used to summarize and graph the causes of failures in order of frequency of occurrence.



Actions

Draw - draw the Pareto chart. The input data should be a **vector of strings**, each of which is the name of a failure cause. Normally only a small set (2 to 10) different causes are used, but the vector can consist of any desired number of instances of each string.

The action parameters include:

graph - a Graph object into which the chart will be drawn. This graph object should be set to "bar chart" mode. It should also have its X axis set to non-autoscale mode, with "from" set to -0.9, and "to" set to the number of different causes - 0.1. Copying the example in SPC.TST and modifying the X axis settings as appropriate is the best way to create this graph object.

legend-text - a Data-Entry object, set to multi-line mode, into which the legend for the chart will be placed.

cause strings - the string vector containing the causes to be processed.

mode - "percentage" or "frequency". Percentage charts give each cause by percentage of the total, with a cumulative total of 100%. Frequency charts give the actual raw number of occurrences of each cause.

Data

The data value of the object is a **list** containing 5 **vectors**: the cause strings (names), their frequency of occurrence, the cumulative frequencies, the percentage values, and the cumulative percentages. These vectors are sorted in order from most frequent cause to the least frequent.

The easiest way to view this result data is to place it into a Grid object.

Settings

none

TestPoint Data Types and Usage for SPC

Group data (for Xbar/R and Xbar/s):

list of vectors	each row is a sample, # items = sample size
array	each row is a sample, # cols = sample size
vector	one sample, dim(vector) = sample size
list of numbers	one sample, dim(list) = sample size
all other types	not allowed

Individual data (x/mR, mXbar/mR, p, and c):

vector	multiple samples, dim(vector) = # samples
number	one sample
string	(automatically converted to a number)
all other types	not allowed

Chapter 10. Distributing Runtimes

Licensing Policy

No runtime fees are required to distribute applications written using TestPoint or the TestPoint SPC Toolkit. No execution key is required on the printer port for runtimes.

You may freely copy and distribute the runtime support files, including the file TPSPC.DLL with your applications.

You may not distribute the TestPoint editor or other development-only portions of the software to anyone else.

Creating a runtime - Files to distribute

To create a runtime distribution, use the standard Utilities menu command Make Runtime Disk, just as you would for any other TestPoint application.

Use the **Add Files** button to add the following file, which is **required** for SPC applications:

TPSPC.DLL

Appendices

Appendix A: Error Messages

17000 - Invalid input data type

This message appears when the SPC Data or SPC Limits object is used, and the data provided is not of a type appropriate for the selected calculation. See the Reference section under TestPoint Data Types and Usage for SPC for details. Use the View/Data menu command to check the actual data type of the object you are passing into the calculation.

17001 - Invalid sample size, must be greater than zero

In a p (fraction nonconforming) calculation, the sample size parameter must be greater than zero, since it is divided into the input data values.

17002 - Invalid sample size

In the SPC Limits object, the sample size must be valid for the type of calculation chosen. All group statistics (Xbar/R, Xbar/s) require sample sizes greater than one. For range (R), the sample size must be between 2 and 15. For standard deviation (s), the sample size must be between 2 and 25. For moving range (mR), the # samples used in the moving range calculation must be between 2 and 8. These limitations are imposed by the SPC constants tables built into the SPC objects (see Constants appendix).

17003 - Invalid data set for limits calculation

The data used for SPC Limits calculations must contain enough samples, **even after the start index, end index, and exclude parameters are applied**, to allow for limit calculations. If all the input samples are excluded, no limits can be calculated.

17005 - Invalid data for Pareto analysis

The input data set for the Pareto chart object must be a **vector of strings** data type in TestPoint.

Appendix B: List of Examples and Files

SPC.TST	The main example program, used in this manual, which illustrates all the parts of the SPC toolkit. This example contains all the SPC Toolkit objects, so it is a good source for copying and pasting objects.
SPCLIMIT.TST	A smaller example of determining limits from an initial data set. This example allows for modifying the start and end indices and entering a list of points to exclude from the calculation.
SPC.DAT	Sample data for use with SPC.TST.
SPCCAUSE.DAT	Sample failure cause data for Pareto analysis in SPC.TST

Appendix C: Formulae and Constants

Xbar and R

Xbar is the arithmetic average of the values in a sample:

$$\bar{x} = \frac{\sum x}{samplesize}$$

R is the range of values in a sample

$$R = \max(x) - \min(x)$$

The control limits are calculated as follows:

$$\bar{\bar{x}} = \frac{\sum \bar{x}}{\#samples} \quad , \quad \bar{R} = \frac{\sum R}{\#samples}$$

$$LCL(x) = \bar{\bar{x}} - A_2 \cdot \bar{R} \qquad LCL(R) = D_3 \cdot \bar{R}$$

$$CL(x) = \bar{\bar{x}} \qquad CL(R) = \bar{R}$$

$$UCL(x) = \bar{\bar{x}} + A_2 \cdot \bar{R} \qquad UCL(x) = D_4 \cdot \bar{R}$$

where the constants depend on the sample size:

samplesize	A2	D3	D4
2	1.88	0	3.267
3	1.023	0	2.574
4	0.729	0	2.282
5	0.577	0	2.114
6	0.483	0	2.004
7	0.419	0.076	1.924
8	0.373	0.136	1.864
9	0.337	0.184	1.816
10	0.308	0.223	1.777
11	0.285	0.256	1.744
12	0.266	0.283	1.717
13	0.249	0.307	1.693
14	0.235	0.328	1.672
15	0.223	0.347	1.653

Xbar and s

Xbar is the arithmetic average of the values in a sample:

$$\bar{x} = \frac{\sum x}{samplesize}$$

s is the standard deviation of the values in the sample:

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{samplesize - 1}}$$

The control limits are calculated as follows:

$$\bar{\bar{x}} = \frac{\sum \bar{x}}{\#samples} \quad , \quad \bar{s} = \frac{\sum s}{\#samples}$$

$$LCL(x) = \bar{\bar{x}} - A_3 \cdot \bar{s} \qquad LCL(s) = B_3 \cdot \bar{s}$$

$$CL(x) = \bar{\bar{x}} \qquad CL(s) = \bar{s}$$

$$UCL(x) = \bar{\bar{x}} + A_3 \cdot \bar{s} \qquad UCL(x) = B_4 \cdot \bar{R}$$

where the constants depend on the sample size:

samplesize	A3	B3	B4
2	2.659	0	3.267
3	1.954	0	2.568
4	1.628	0	2.266
5	1.427	0	2.089
6	1.287	0.03	1.97
7	1.182	0.118	1.882
8	1.099	0.185	1.815
9	1.032	0.239	1.761
10	0.975	0.284	1.716
11	0.927	0.321	1.679
12	0.886	0.354	1.646
13	0.85	0.382	1.618
14	0.817	0.406	1.594
15	0.789	0.428	1.572
16	0.763	0.448	1.552
17	0.739	0.466	1.534
18	0.718	0.482	1.518
19	0.698	0.487	1.503
20	0.68	0.51	1.49
21	0.663	0.523	1.477
22	0.647	0.534	1.466
23	0.633	0.545	1.455
24	0.619	0.555	1.445
25	0.606	0.565	1.435

X and mR

X is simply the input values, unmodified.

mR is the range of the most recent N samples, where N is a setting of the SPC Data or Limits object.

The control limits are calculated as follows:

$$\overline{\overline{x}} = \frac{\sum x}{\#samples} \quad , \quad m\overline{R} = \frac{\sum mR}{\#samples}$$

$$LCL(x)= \overline{\overline{x}} - d_2 \cdot m\overline{R} \qquad LCL(mR)= D_3 \cdot m\overline{R}$$

$$CL(x)= \overline{\overline{x}} \qquad CL(mR)= m\overline{R}$$

$$UCL(x)= \overline{\overline{x}} + d_2 \cdot m\overline{R} \qquad UCL(x)= D_4 \cdot \overline{R}$$

where the constants above depend on the sample size. D3 and D4 are the same as in the earlier table given for Xbar and R charts. d2 is as follows:

samplesize	d2
2	2.66
3	1.772
4	1.457
5	1.29
6	1.183
7	1.109
8	1.054

mXbar and mR

mXbar is the average of the most recent N samples

mR is the range of the most recent N samples

Limits are calculated exactly as for X and mR charts, above.

p

p is the fraction of nonconforming parts, given by:

$$p = \frac{\text{\#nonconforming}}{\text{samplesize}}$$

Limits are calculated as follows:

$$\overline{p} = \frac{\sum p}{\#samples} \quad , \quad s = \sqrt{\frac{\overline{p}(1-\overline{p})}{samplesize}}$$

$$UCL(p) = \overline{p} + 3s$$

$$CL(p) = \overline{p}$$

$$LCL(p) = \overline{p} - 3s$$

c

c is the count of defects - simply the input values unmodified.

Limits are calculated as follows:

$$\overline{c} = \frac{\sum c}{\#samples} \quad , \quad s = \sqrt{\overline{c}}$$

$$UCL(c) = \overline{c} + 3s$$

$$CL(c) = \overline{c}$$

$$LCL(c) = \overline{c} - 3s$$

Appendix D: Bibliography

Evans, James R., *Statistical Process Control for Quality Improvement*, Prentice Hall, 1991.

Wise, Stephen A. and Fair, Douglas C., *Innovative Control Charting*, ASQ Quality Press, 1998.

American Society for Quality Control, *American National Standard: Definitions, Symbols, Formulas, and Tables for Control Charts*, 1987 (ANSI/ASQC A1-1987).