Statistical Quality Control in R

Yabebal Ayalew

2022-12-30

Data Source

The data for this tutorial are coming from a `qcc` package. The data sets are

1. `pistonrings`
2. `orangejuice`
3. `circuit`
4. `antifreeze`

More description about these data sets can be obtained if we type `?<dataName>`. *i.e.*, `?pistonrings`, `?circuit`

Downloading Latest `qcc` Package

You can install the latest released version of `qcc` from CRAN:

```
install.packages("qcc")
```

or the development version from GitHub:

```
# install.packages("devtools")
devtools::install_github("luca-scr/qcc", build = TRUE,
                         build_opts = c("--no-resave-data", "--no-manual"))
```

Load the data sets and `qcc` library

```
# Clear everything stored in the memory prior to this tutorial
rm(list = ls())

# Load the library
library(qcc)

# Load the data
data("pistonrings")
data("orangejuice")
data("circuit")
data("antifreeze")
```

Process Monitoring Using $\bar{x}$, R and S Charts

Changing the format of `pistonrings` data

This data set has *three variables*, namely, diameter, sample, trial. `diameter` of a piston ring is the key quality characteristics that we would like to monitor in the production process. `sample` indicates the rational subgroup that a particular observation belongs. `trial` indicates the production phase. As we have discussed in class, Phase I production (`trial = TRUE`) data mainly used to compute the control limits. Moreover, it is known that the main purpose of control charts in production phase I is not to *monitor* the production process instead to help the production to be stable and identify and remove assignable causes that cab potentially cause major shifts in the production.

`qcc.groups(data, sample)` is the function to reshape the data for older version of `qcc` package. For `qcc` version 3.0, we have to use `qccGroups(data, x, sample)` as `qcc.groups` is deprecated.

```
#diameter <- with(pistonrings, qcc.groups(diameter, sample = sample))
diameter <- qccGroups(data = pistonrings, diameter, sample)
```

Let's preview the `diamond` data frame.

```
head(diameter)
```

```
##     [,1]   [,2]   [,3]   [,4]   [,5]
## 1 74.030 74.002 74.019 73.992 74.008
## 2 73.995 73.992 74.001 74.011 74.004
## 3 73.988 74.024 74.021 74.005 74.002
## 4 74.002 73.996 73.993 74.015 74.009
## 5 73.992 74.007 74.015 73.989 74.014
## 6 74.009 73.994 73.997 73.985 73.993
```

As can be seen above, the data is converted into a wide shape. The main reason for changing the data frame from long to wide is because of the fact that `qcc` function, which we will be using to draw control charts, demands the data to be arranged in the wider format.

Few Points on `qcc` Function

The classical control charts developed by Shewhart are implemented by `qcc` function of the `qcc` library. To get help about this function type `?qcc` on the console.

```
Description

Create an object of class 'qcc' to perform statistical quality control.
This object may then be used to plot Shewhart charts, drawing OC curves,
computes capability indices, and more.

Usage

qcc(data,
    type = c("xbar", "R", "S", "xbar.one", "p", "np", "c", "u", "g"),
    sizes, center, std.dev, limits,
    newdata, newsizes,
    nsigmas = 3, confidence.level,
    rules = c(1,4), ...)
```

The arguments are

1. `data` The data frame where columns are sample measurements of key quality characteristics and rows are rational subgroups usually the time that indicates when the samples are taken
2. `type` This argument specifies the type of chart we would like to draw. For instance, for $\bar{x}$ chart, we can type `type = 'xbar'`
3. `sizes` The sample size or number of items samples in each rational subgroup. Usually used for `p`, `np`, `c` and `u` charts
4. `center` This is the process average. If the standard value of the process average is known, we have to explicitly specify it here. By doing so, we can prevent the function to compute the process average based on a given data.
5. `std.dev` As like as `center`, we can specify the process variability (standard deviation) here.
6. `limits` A two-values vector specifying lower and upper control limits. If this value is not given, the function computes the upper and lower control limits from the data provided
7. `newdata` A data frame from phase II production process. This data set is different from `data` in the way that it can't be used to compute control limits.
8. `newsizes` A vector of sample sizes from phase II production process
9. `nsigmas` By default 3 sigma is being used. If special interest arises, we can use different value of sigma
10. `confidence.limit` This alternative way of overriding `nsigma`. *i.e.*, if `confidence.limit` is specified `nsigma` will be ignored by the function
11. `rules` It is the rule that will be used to judge whether the production process is in-control or not. By default the first and fourth rules are used (`rules = c(1,4)`). However, we can use other rules. The rules are developed by Western electric (Type `?qccRules` for more information)

- `rules = 1` One point plots outside 3-sigma control limits.
- `rules = 2` Two of three consecutive points plot beyond a 2-sigma limit.
- `rules = 3` Four of five consecutive points plot beyond a 1-sigma limit.
- `rules = 4` Eight consecutive points plot on one side of the center line.

We will see how each arguments can be used in the next section.

$\bar{x}$ Chart

Note: There are two methods of estimating process variability. When the sample size for each rational subgroup is small, we can use the *range method*. However, this method loses its efficiency when the sample size is greater than 10. Therefore, when we are using $\bar{x}$ chart to monitor process average, we have to think about the appropriate control chart to monitor process variability with respect to the key quality characteristics.

Based on the `diamond` data set, we know that the first 25 rational subgroups are taken from Phase I production process and can be used to estimate the control limits and central line ($\bar{\bar{x}}$).

```
# x-bar chart
output1 <- qcc(data = diameter[1:25,], type = "xbar")
```

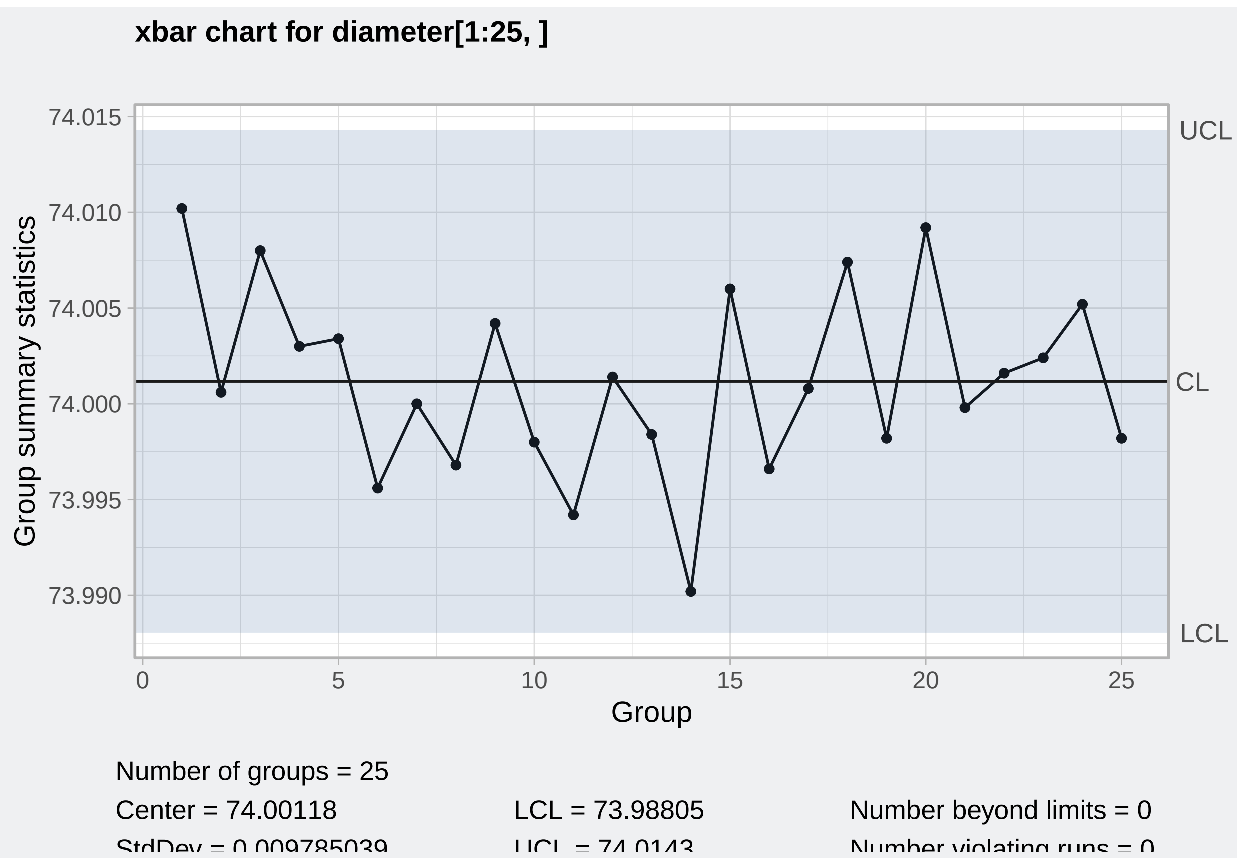
`output1` is the `qcc` object. We can summarize the result stored in `output1` by using `summary` function as follows

```
# Summarize the control chart
summary(output1)
```

```
## -- Quality Control Chart -------------------------
##
## Chart type                 = xbar
## Data (phase I)             = diameter[1:25, ]
## Number of groups           = 25
## Group sample size          = 5
## Center of group statistics = 74.00118
## Standard deviation         = 0.009785039
##
## Control limits at nsigmas  = 3
##        LCL      UCL
##  73.98805 74.0143
```

Using `plot` function we can draw the control chart.

```
# Drawing the x-bar chart
plot(output1)
```



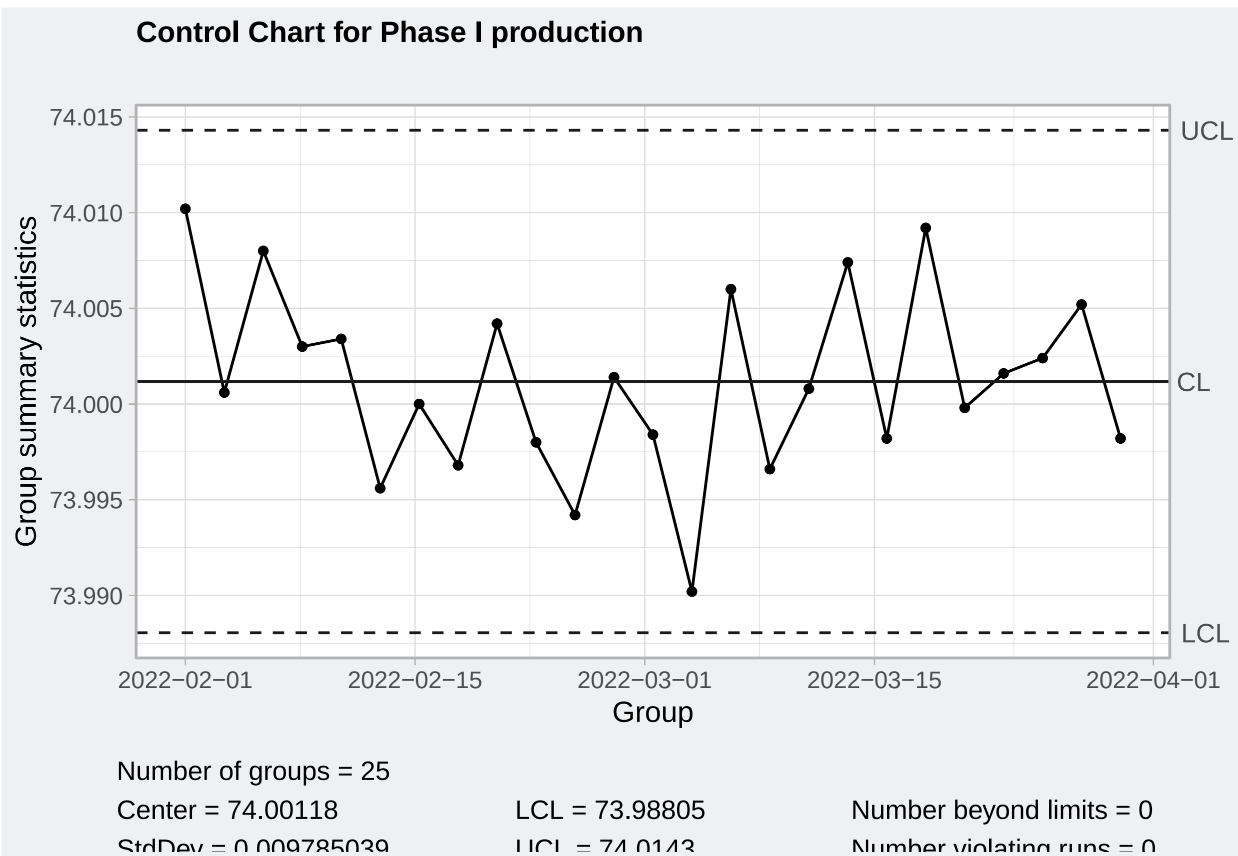
As you know `plot` is a standard drawing library in `R`. However, for `qcc` objects, it has few additional arguments

```
plot(x,
     xtime = NULL,
     add.stats = qcc.options("add.stats"),
     chart.all = qcc.options("chart.all"),
     fill = qcc.options("fill"),
```

```
label.center = "CL",
label.limits = c("LCL ", "UCL"),
title, xlab, ylab, xlim, ylim,
digits = getOption("digits"), ...)
```

- `xtime` is the time stamp that should be printed on the x-axis instead of the row numbers of the rational subgroups
- `add.stats` is a logical value indicating whether statistics and other information should be printed at the bottom of the chart. The default is `add.stats = TRUE`
- `chart.all` is a logical value indicating whether both statistics for data and for `newdata` (if given) should be plotted.The default is `chart.all = TRUE`
- `fill` is a logical value specifying if the in-control area should be filled with the color
- `label.center` is a character specifying the label for center line.
- `label.limits` is a character vector specifying the labels for control limits. The default is `abel.limits = c("LCL ", "UCL")` Now let's create 25 time stamps and put it on x-axis of the above plot.

```r
# Creating sequence of time
timex <- seq(as.Date('2022-02-01'), as.Date('2022-03-30'), length.out = 25)

# Modify the above x-bar chart
plot(output1, xtime = timex, fill = FALSE, title = "Control Chart for Phase I production")
```

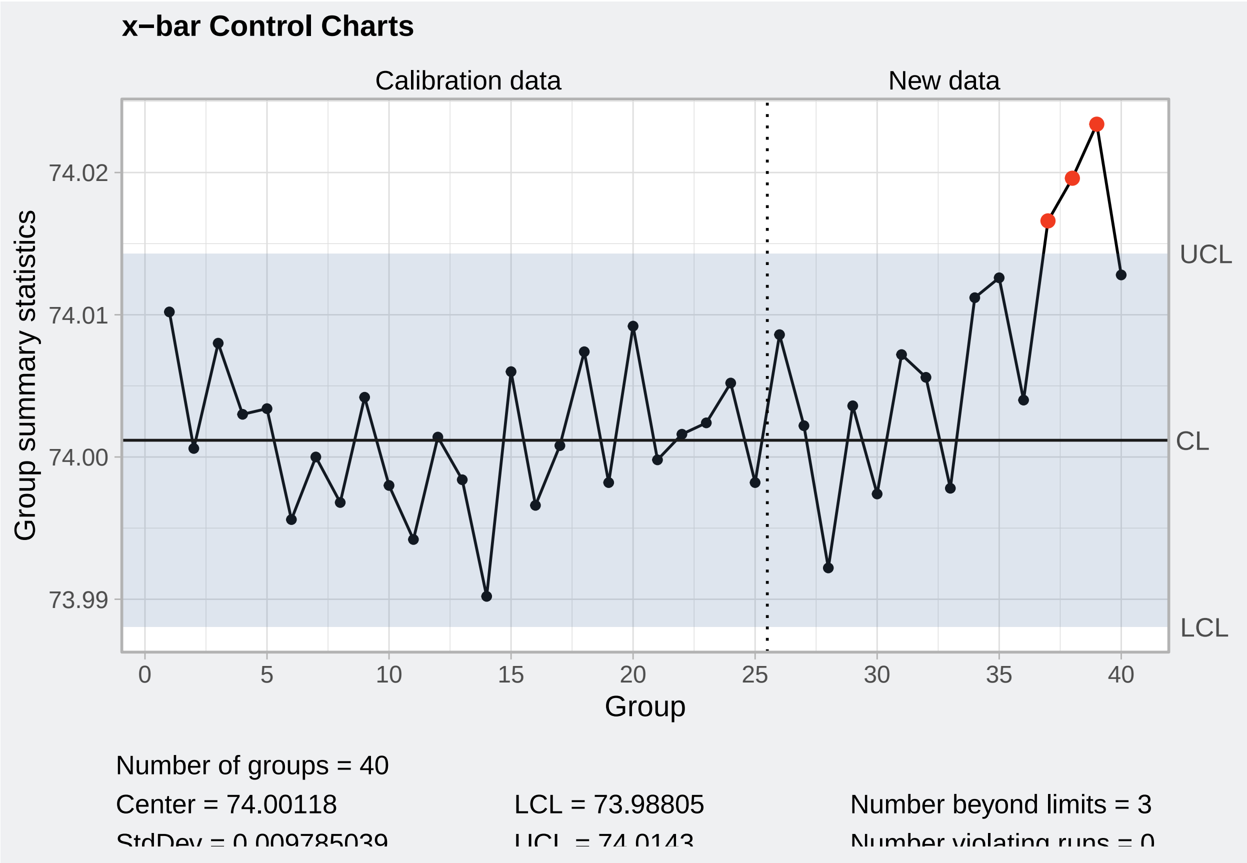


From `26-40` rows, the `diameter` data set has phase II production data set. Let us put it into the existing control limits. To do so, we have to specify the phase II data set on `newdata` argument of `qcc` as follows. For clarity compare the below plot with the above one.

```r
# Adding the phase II production data
output2 <- qcc(data = diameter[1:25,], type = "xbar", newdata = diameter[26:40,])

# summaries the chart
summary(output2)
```

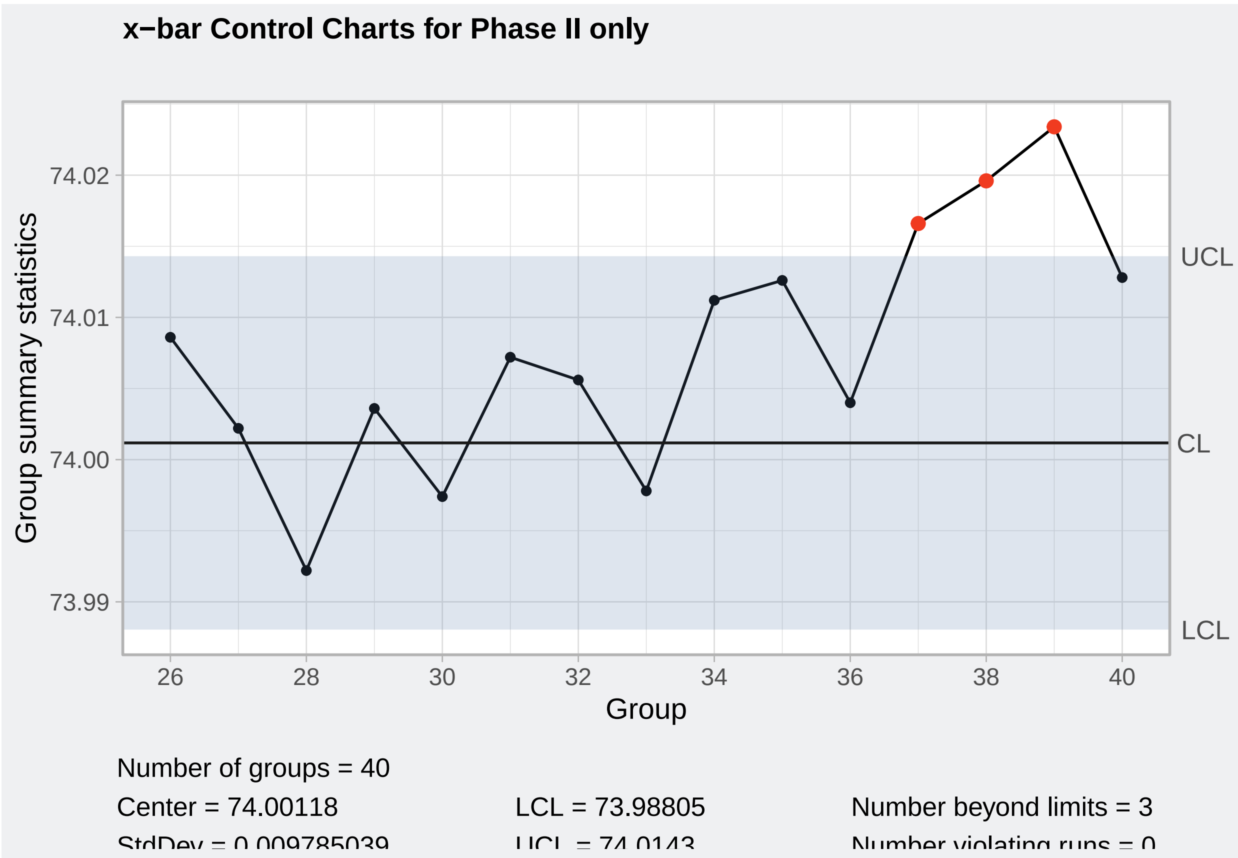
```
## -- Quality Control Chart -------------------------
##
## Chart type                 = xbar
## Data (phase I)             = diameter[1:25, ]
## Number of groups           = 25
## Group sample size          = 5
## Center of group statistics = 74.00118
## Standard deviation         = 0.009785039
##
## New data (phase II)        = diameter[26:40, ]
## Number of groups           = 15
## Group sample size          = 5
##
## Control limits at nsigmas  = 3
##       LCL     UCL
##  73.98805 74.0143
```

```r
# Drawing the chart
plot(output2, title = "x-bar Control Charts")
```



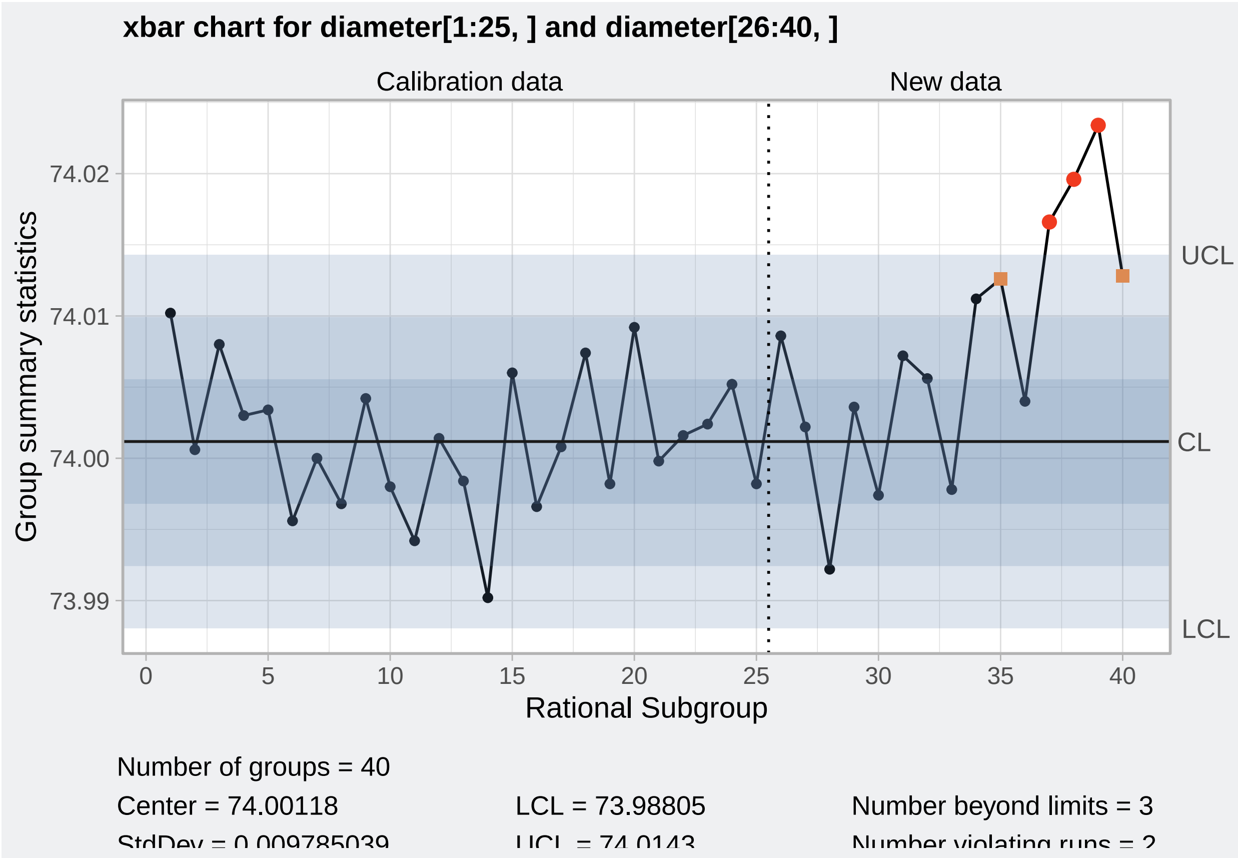
If we want to draw the $\bar{x}$ chart for phase II production alone (*i.e.*, hiding phase I control chart), we have to set `chart.all = F` of the `plot` function as below.

```
# Excluding Phase I part of the chart
plot(output2, title = "x-bar Control Charts for Phase II only", chart.all = F)
```



As can be seen in the above graph, only rule 1 and 4 are being used to conclude that the process is out of control. At rational subgroup `40`, the graph showed that the process comes back to in-control situation and the first out of control situation was detected at rational subgroup `37`. However, We can add additional Western electric rules and able to detect the out of control situation as early as possible. To do that, we have to set `rules = 1:4`.

```
# Western Electric Rules
output3 <- qcc(data = diameter[1:25,], type = "xbar", newdata = diameter[26:40,],
               rules = 1:4)

# Drawing the chart with modified name on the x-axis
plot(output3, xlab = 'Rational Subgroup')
```



The graph indicated that the out of control situation was first noticed on rational subgroup `35`. From this particular example, we can understand that by applying the Western Electric sensitizing rules, we can detect shifts in the production process as early as possible as it is better to detect out-of control production process at `35` than detecting it at `37`. The factory/company will be benefited if we manage to detect shift in the production process as early as possible.

R Chart

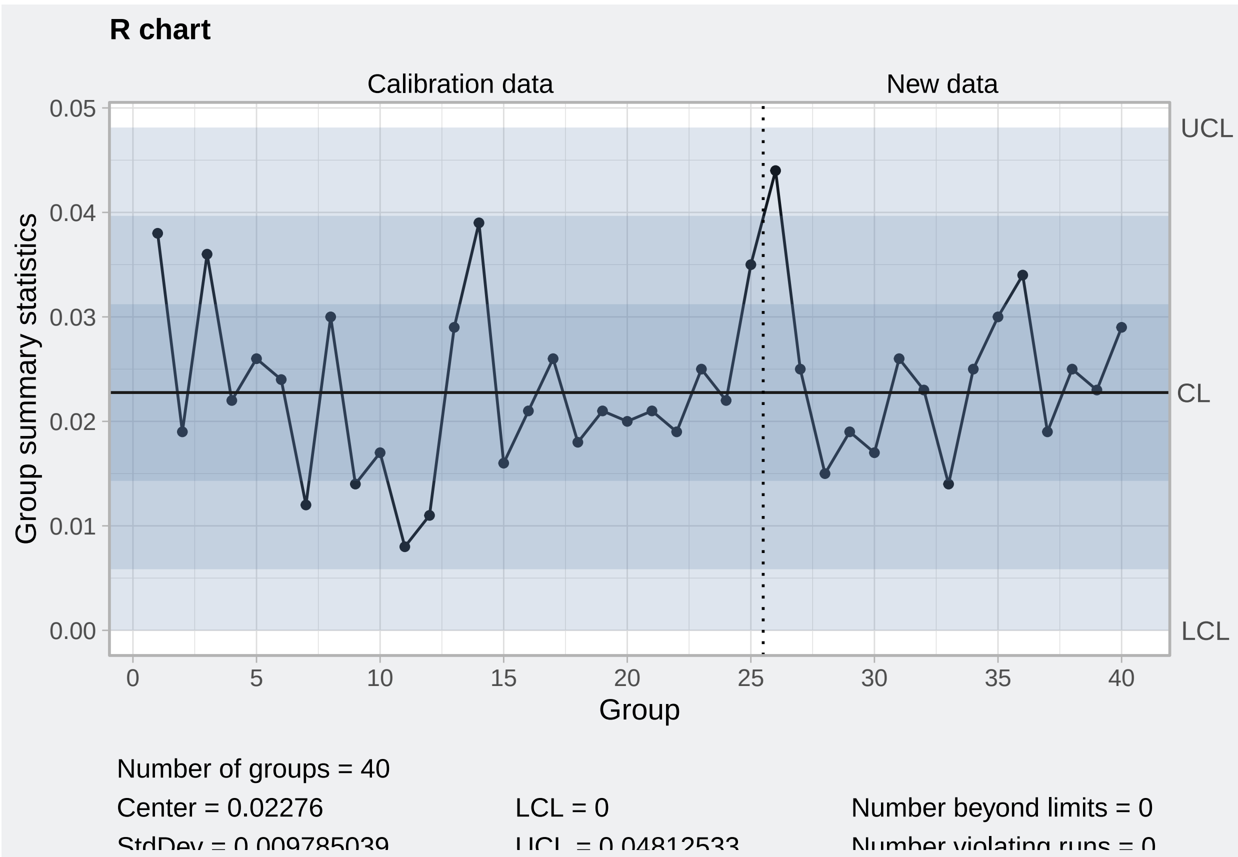
One of the control charts that can be used to monitor process variability is R chart. The `qcc` function argument `type = 'R'` can be used to draw the R chart.

```
# R control chart
Rp <- qcc(data = diameter[1:25,], type = "R", rules = 1:4, newdata = diameter[26:40,])

# Summary of the output
summary(Rp)
```

```
## -- Quality Control Chart -------------------------
##
## Chart type                 = R
## Data (phase I)             = diameter[1:25, ]
## Number of groups           = 25
## Group sample size          = 5
## Center of group statistics = 0.02276
## Standard deviation         = 0.009785039
##
```

```
## New data (phase II)        = diameter[26:40, ]
## Number of groups           = 15
## Group sample size          = 5
##
## Control limits at nsigmas  = 3
##  LCL         UCL
##    0 0.04812533
```

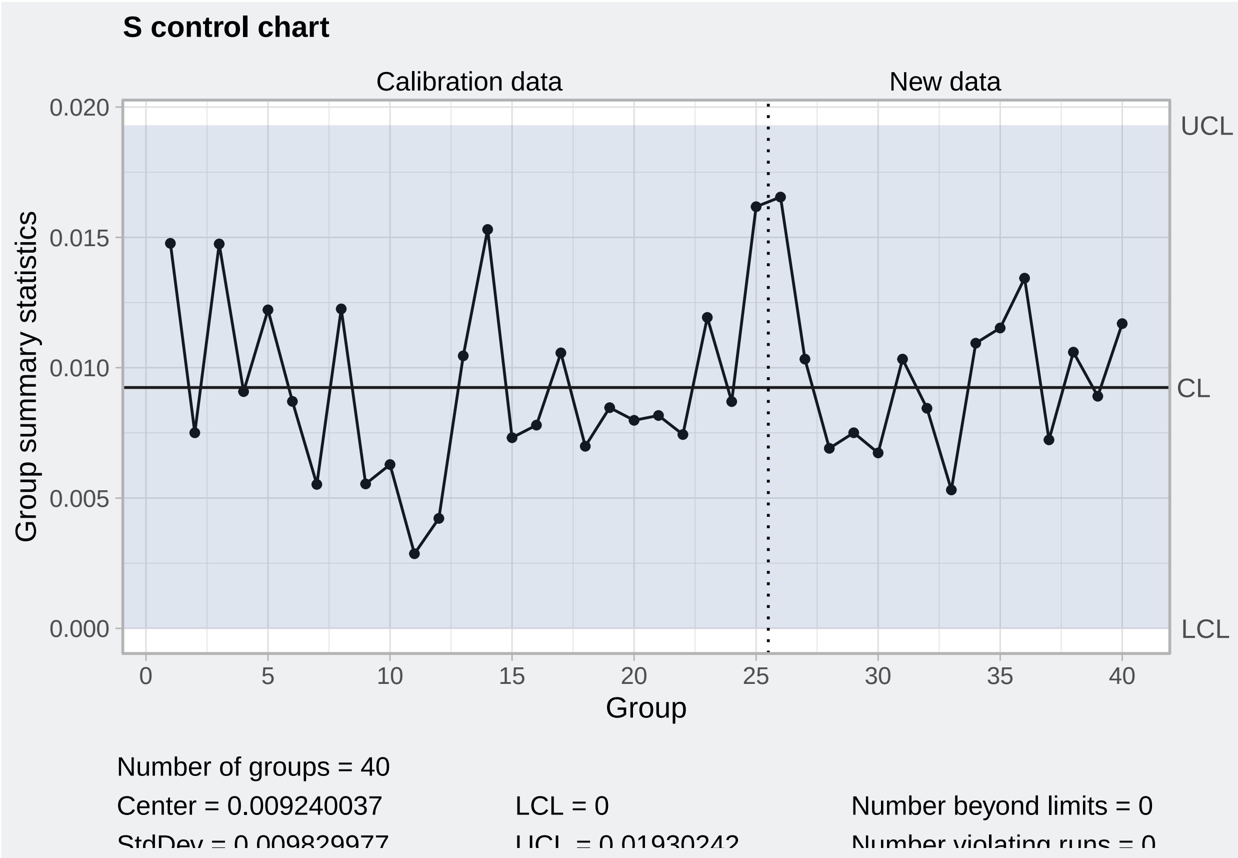
```
# Plot the chart
plot(Rp, title = "R chart")
```



S Chart

S chart is the alternative control chart to monitor process variability. It can be used when the sample size is large ($n \geq 10$) and the sample size is *variable*.

```
# S chart
Sc <- qcc(data = diameter[1:25,], newdata = diameter[26:40,], type = "S")

# Plot the chart
plot(Sc, title = "S control chart")
```



S Control Chart for Variable Sample Size

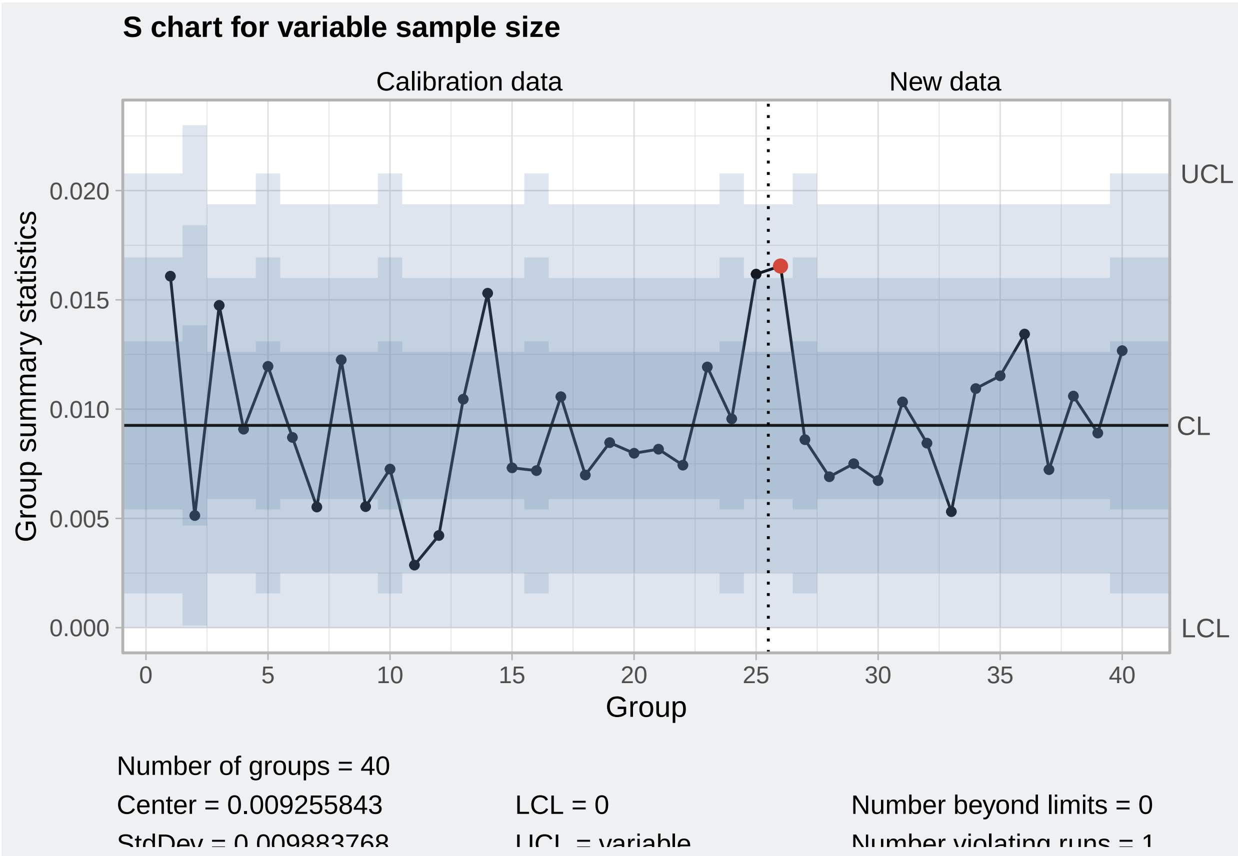
Let's create variable sample size by letting some of the sample elements to be deleted form `pistonrings` data

```
# Removing some observations
idx <- c(3, 6, 7, 23, 46, 78, 120, 134, 200)

# Changing the shape of the data
diameter_new <- qccGroups(pistonrings[-idx,], diameter, sample)

# Preview the data
head(diameter_new)
```

```
##     [,1]   [,2]   [,3]   [,4]   [,5]
## 1 74.030 74.002 73.992 74.008     NA
## 2 74.001 74.011 74.004     NA     NA
## 3 73.988 74.024 74.021 74.005 74.002
## 4 74.002 73.996 73.993 74.015 74.009
## 5 73.992 74.007 73.989 74.014     NA
## 6 74.009 73.994 73.997 73.985 73.993
```

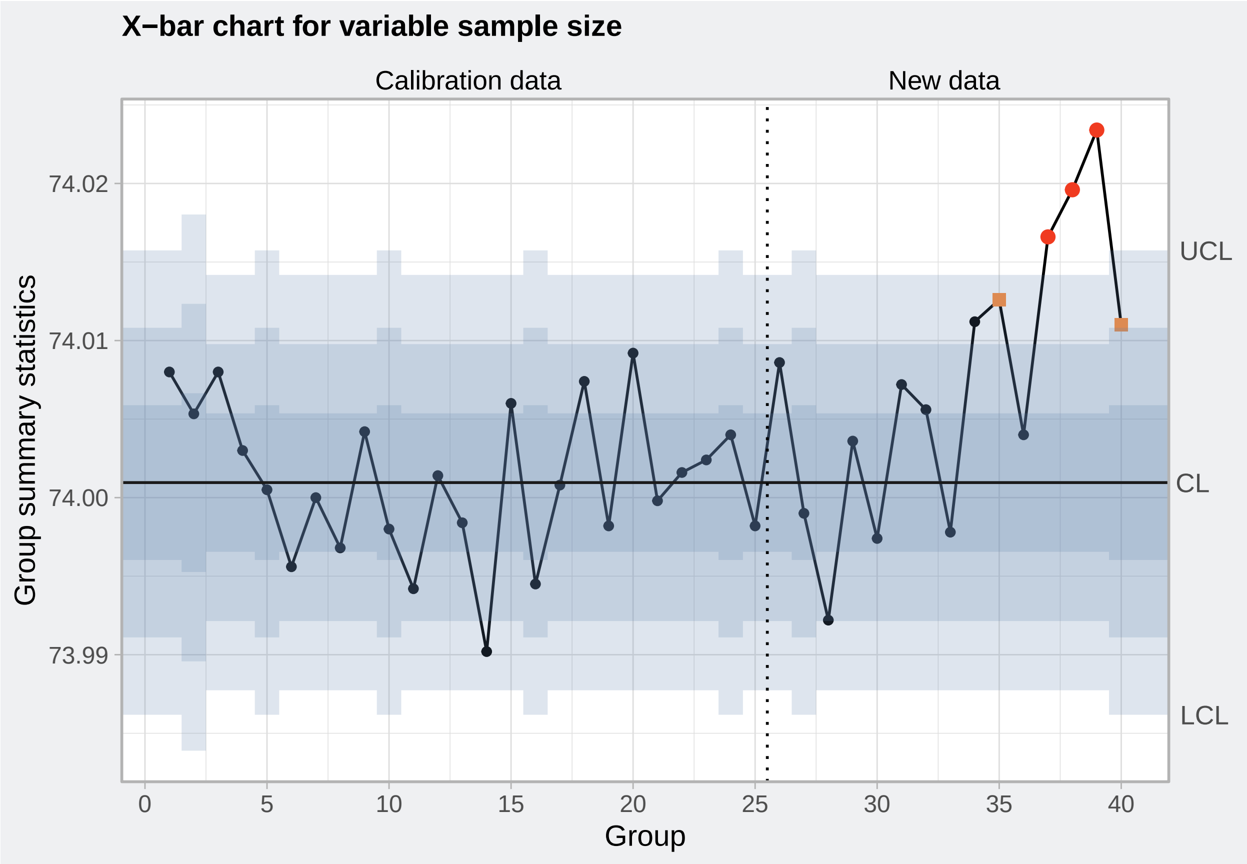
```
# S chart for variable sample size
svar <- qcc(data = diameter_new[1:25,], newdata = diameter_new[26:40,], type = "S", rules = 1:4)
```

```r
# Plot the chart
plot(svar, title = "S chart for variable sample size")
```



$\bar{x}$ Chart for Variable Sample Size

```r
# S chart for variable sample size
xbar <- qcc(data = diameter_new[1:25,], newdata = diameter_new[26:40,], type = "xbar", rules = 1:4)

# Plot the chart: The stats at the bottom are eliminated
plot(xbar, title = "X-bar chart for variable sample size", add.stats = F)
```



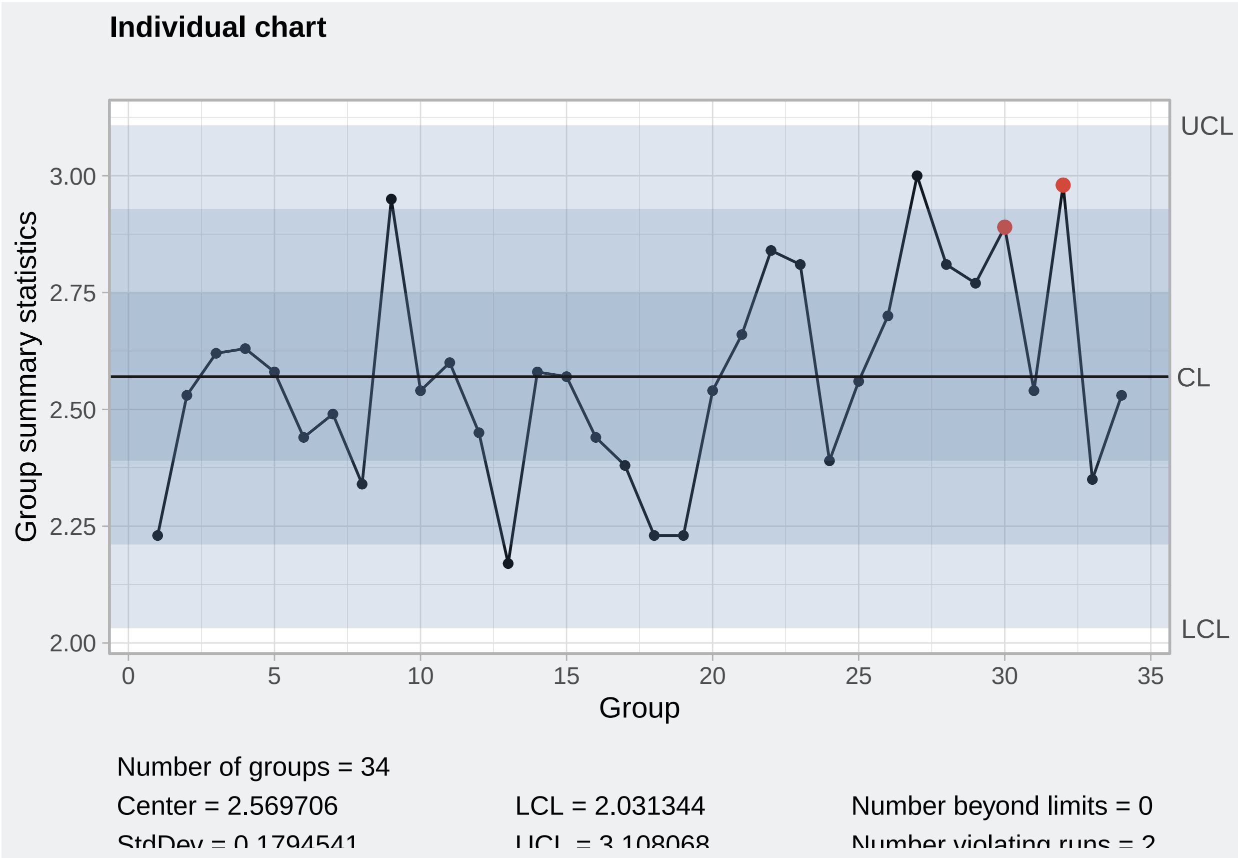
Moving Range and Individual Chart

Both the moving range and individual charts are used when forming rational subgroup is irrelevant. When the production process is too slow to produce new items or every items are being inspected automatically, we prefer to use MR chart over the standard Shewhart's control charts.

For this exercise, we are going to use `antifreeze` data set. Type `?antifreeze` to learn more about the data set. It has 34 reads on water content of batches of *antifreez*. According to wikipedia

> An antifreeze is an additive which lowers the freezing point of a water-based liquid. An antifreeze mixture is used to achieve freezing-point depression for cold environments. Common antifreezes also increase the boiling point of the liquid, allowing higher coolant temperature. However, all common antifreeze additives also have lower heat capacities than water, and do reduce water's ability to act as a coolant when added to it.

```
# Individual chart to monitor process average
xone <- qcc(data = antifreeze, type = "xbar.one", rule = 1:4)

# Plot the chart
plot(xone, title = 'Individual chart')
```



The moving range chart can be drawn by setting `type = "R"`. However, the methods needs at least two columns to be included in the data frame. Since we construct the MR chart by finding the absolute difference between two consecutive samples, we can create a data frame of two columns.

1. The first column will contain values of `antifreeze` from 1 to `nrow(antifreeze)-1`
2. The second column will contain values of `antifreeze` from 2 to `nrow(antifreeze)`

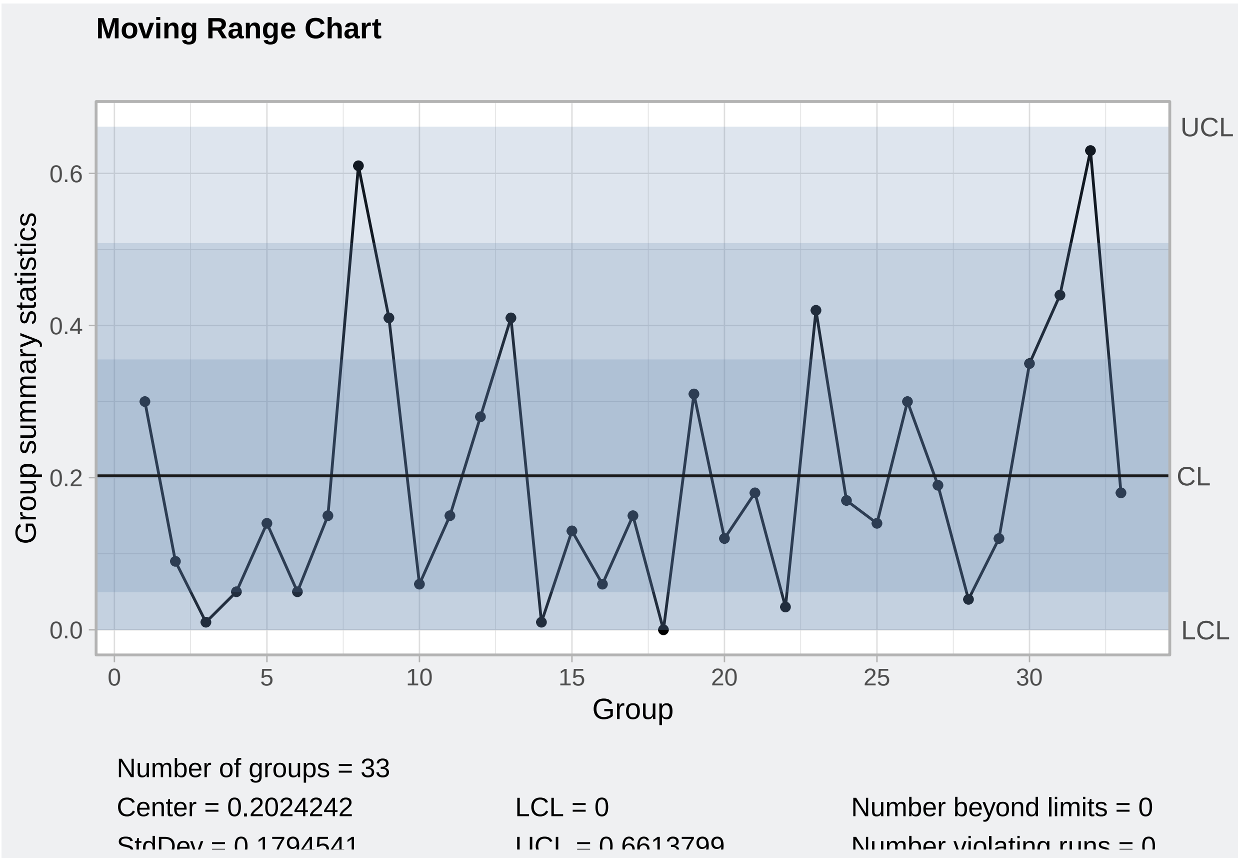
Therefore, the absolute difference between the two columns will be $|x_i - x_{i-1}|$ (range of two values). Now, let's prepare the data frame

```
# Prepare the data frame
n <- nrow(antifreeze)
antifreez_new <- cbind(antifreeze[1:(n-1),], antifreeze[2:n,])

# Preview the data
head(antifreez_new)
```

```
##      [,1] [,2]
## [1,] 2.23 2.53
## [2,] 2.53 2.62
## [3,] 2.62 2.63
## [4,] 2.63 2.58
## [5,] 2.58 2.44
## [6,] 2.44 2.49
```

The MR chart is

```r
# Moving Range
mr <- qcc(data = antifreez_new, type = "R", rule = 1:4)

# Plot moving range chart
plot(mr, title = "Moving Range Chart")
```



p and np Charts

The description of `orangejuice` data set indicated that a can is considered as **nonconforming** if a leak has happened when filled with a liquid.

> Frozen orange juice concentrate is packed in 6-oz cardboard cans. These cans are formed on a machine by spinning them from cardboard stock and attaching a metal bottom panel. A can is then inspected to determine whether, when filled, the liquid could possible leak either on the side seam or around the bottom joint. If this occurs, a can is considered nonconforming. The data were collected as 30 samples of 50 cans each at half-hour intervals over a three-shift period in which the machine was in continuous operation. From sample 15 used a new batch of cardboard stock was punt into production. Sample 23 was obtained when an inexperienced operator was temporarily assigned to the machine. After the first 30 samples, a machine adjustment was made. Then further 24 samples were taken from the process.

The Data has 54 observations on the following 4 variables:

1. `sample` sample id

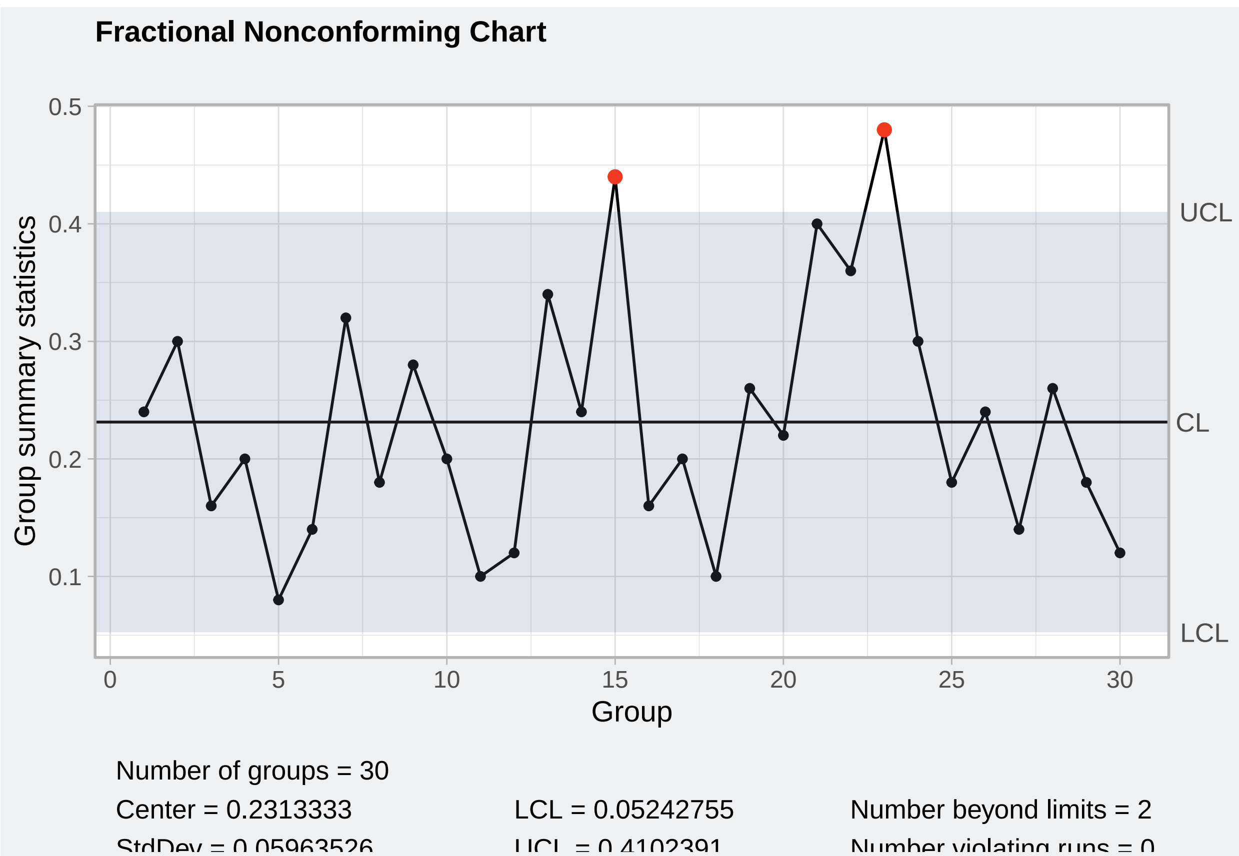
2. `D` number of defectives
3. `size` sample sizes
4. `trial`trial samples (TRUE/FALSE)

```
# Preview the data
head(orangejuice)
```

```
##   sample  D size trial
## 1      1 12   50  TRUE
## 2      2 15   50  TRUE
## 3      3  8   50  TRUE
## 4      4 10   50  TRUE
## 5      5  4   50  TRUE
## 6      6  7   50  TRUE
```

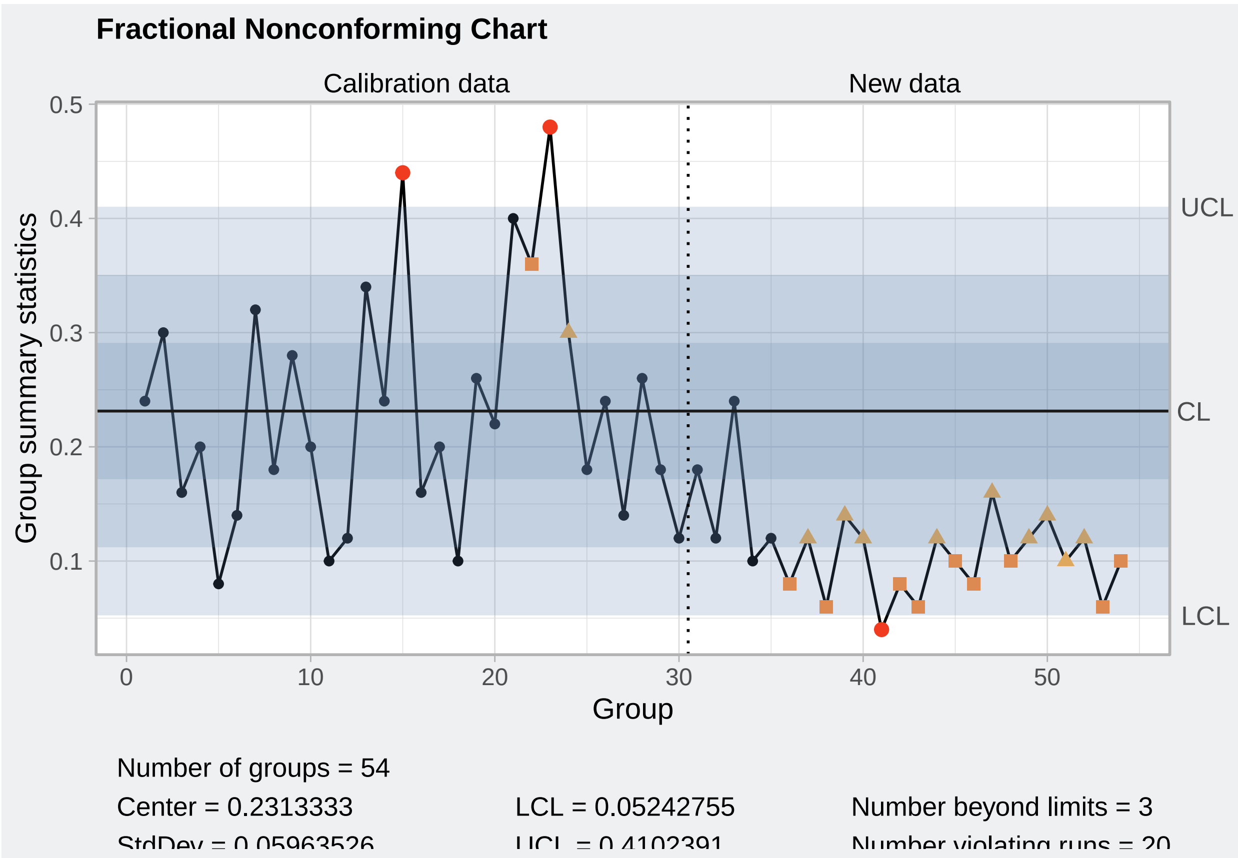
The p chart is

```
# P chart
pc <- qcc(data = orangejuice$D[1:30], sizes = orangejuice$size[1:30], type = "p")

# Plot the chart
plot(pc, title = "Fractional Nonconforming Chart")
```



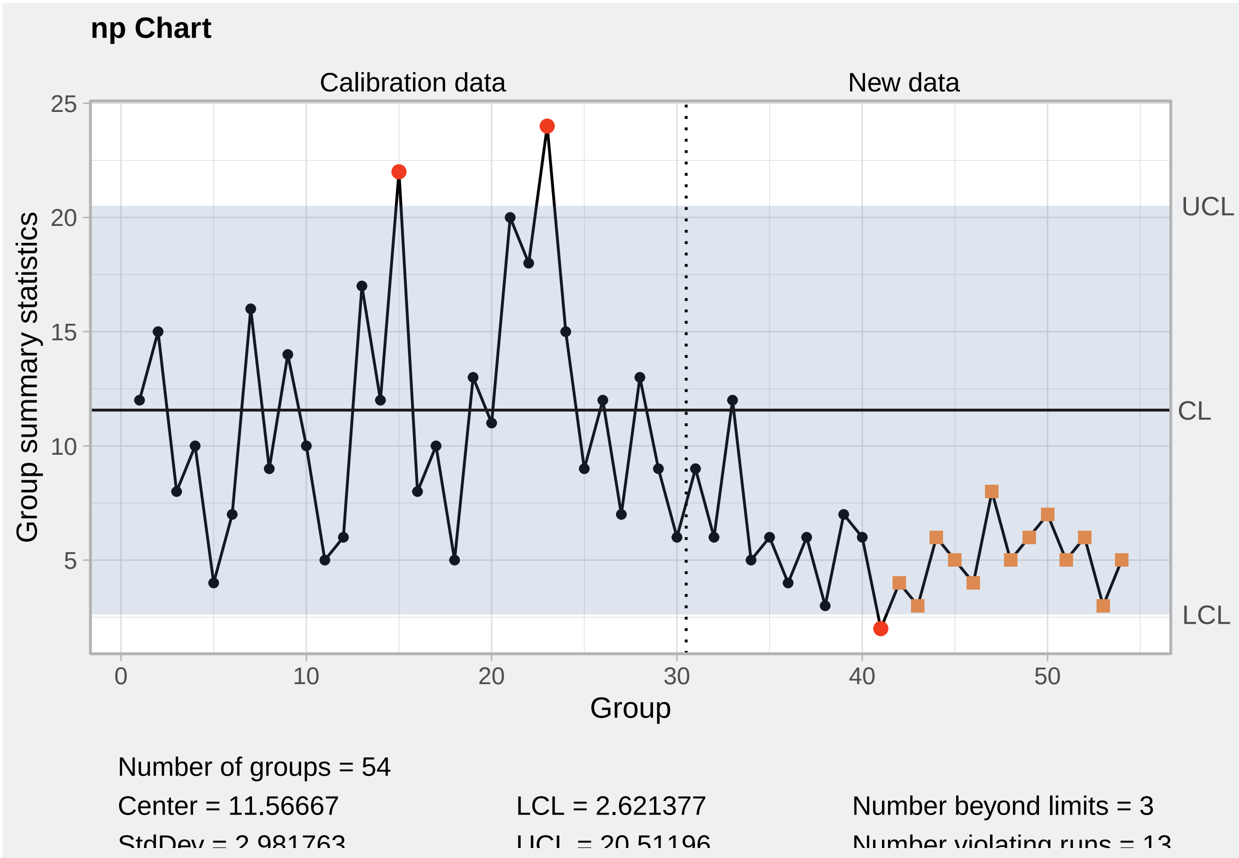
The p chart for Phase II is

```r
# P chart
pc <- qcc(data = orangejuice$D[1:30], sizes = 50, type = "p",
          newdata = orangejuice$D[31:54], newsizes = 50, rules = 1:4)

# Plot the chart
plot(pc, title = "Fractional Nonconforming Chart")
```



The *np* control chart can be drawn if we set `type = "np"` of the `qcc` function.

```r
# nP chart
np <- qcc(data = orangejuice$D[1:30], sizes = 50, type = "np",
          newdata = orangejuice$D[31:54], newsizes = 50)

# Plot the chart
plot(np, title = "np Chart")
```



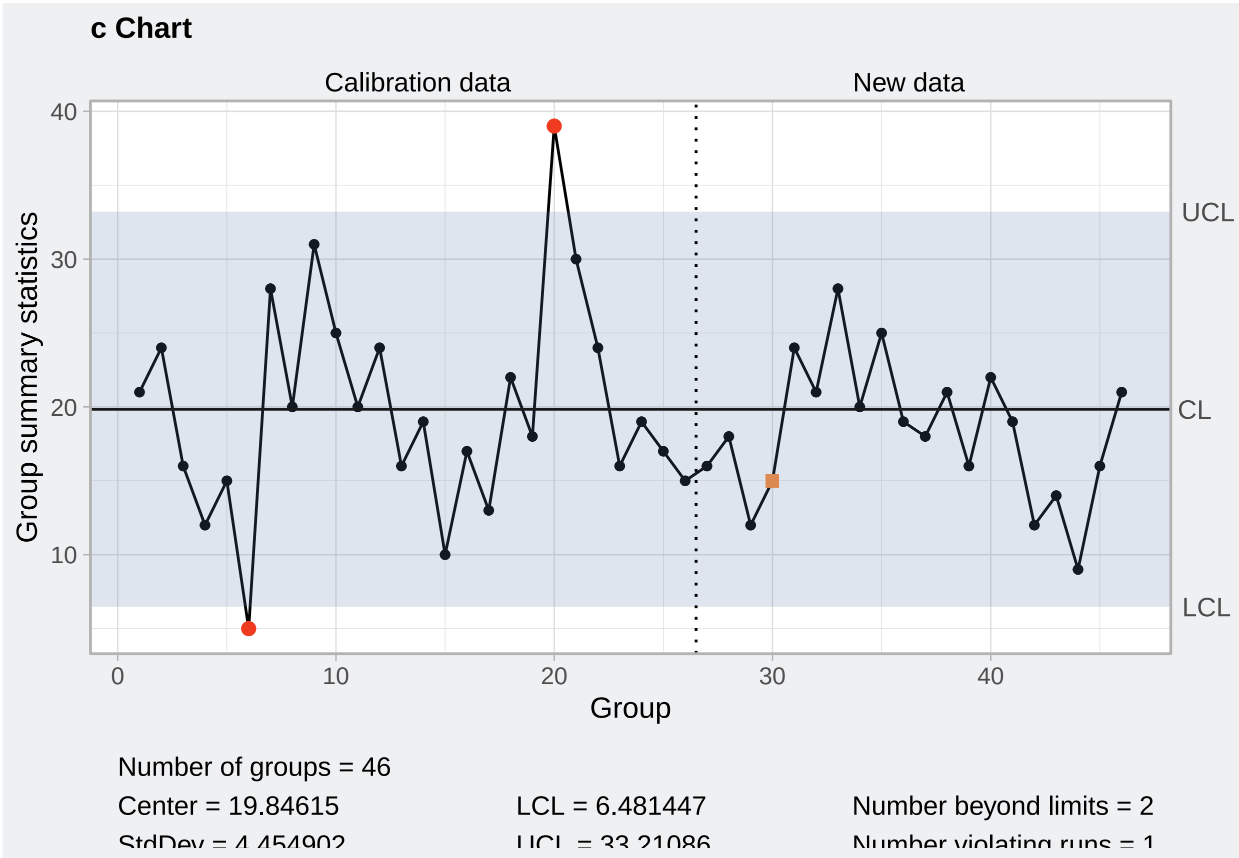
Note As can be seen in the above two charts, the basic concision that we will be making whether the process is in or out-of control is exactly the same.

c and u Control Charts

The description of `circuit` data set indicated that number of nonconformities were recorded for 100 printed circuits for 26 successive samples.

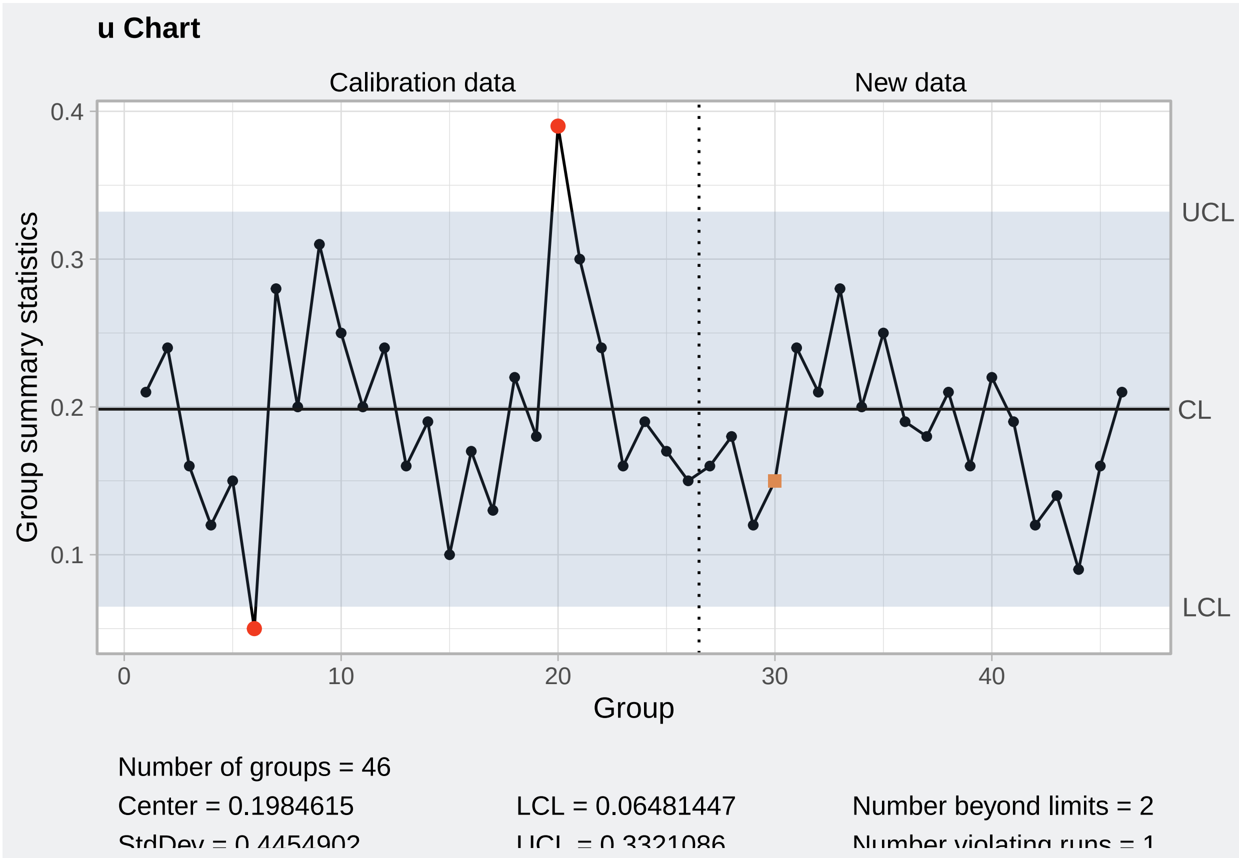
> Number of nonconformities observed in 26 successive samples of 100 printed circuit boards. Sample 6 and 20 are outside the control limits. Sample 6 was examined by a new inspector and he did not recognize several type of nonconformities that could have been present. Furthermore, the unusually large number of nonconformities in sample 20 resulted from a temperature control problem in the wave soldering machine, which was subsequently repaired. The last 20 samples are further samples collected on inspection units (each formed by 100 boards).

```
# c chart
cc <- qcc(data = circuit$x[1:26], sizes = 100, type = "c",
          newdata = circuit$x[27:46], newsizes = 100)

# Plot the chart
plot(cc, title = "c Chart")
```



The u chart is

```
# u chart
uc <- qcc(data = circuit$x[1:26], sizes = 100, type = "u",
          newdata = circuit$x[27:46], newsizes = 100)

# Plot the chart
plot(uc, title = "u Chart")
```



Have you seen the similarity of the conclusion we can possibly made about the production process? The conclusion is exactly the same. So, the preference to use either p over np or c over u is just merely depending on quality engineers interest.

c and u for Variable Sample Size

```r
# Randomly select 10 row numbers
set.seed(23)
idx <- sample(1:46, size = 10)

# Change the number of circuits for randomly selected rows
circuit$size[idx] <- sample(50:200, size = 10)

# Preview the data
head(circuit)
```

```
##   sample  x size trial
## 1      1 21  100  TRUE
## 2      2 24  100  TRUE
## 3      3 16  100  TRUE
## 4      4 12  100  TRUE
## 5      5 15  100  TRUE
## 6      6  5  100  TRUE
```

```r
# U and C charts
uchart <- qcc(data = circuit$x[1:26], sizes = circuit$size[1:26], type = "u",
              newdata = circuit$x[27:46], newsizes = circuit$size[27:46], rules = 1:4)

# Plot the chart
plot(uchart, title = "u Chart")
```