Control charts methodology

Michy Alice

Saturday, November 28, 2015

Definitions

- n_i Number of observations of the ith sample batch ($i = 1...m$)
- m Number of batches of samples
- p Fraction of unfit products/units/other
- D_i Number of unfit products/units/other in the ith batch ($i = 1...m$)
- $\hat{x}_i$ Sample mean of the ith batch of observations ($i = 1...m$)
- $\hat{x}^* = \sum_{i=1}^{m} \frac{\hat{x}_i}{m}$ Average of sample means
- $\hat{s}_i$ Sample standard deviation of the ith batch of observations (5.29 Montgomery)
- $\hat{r}_i = \max x_i - \min x_i$ Range of the i -th sample batch
- $\hat{R} = \sum_{i=1}^{m} \frac{\hat{r}_i}{m}$ Average range
- $W = \frac{\hat{R}}{\sigma}$ Relative range

P chart

Standards given: p is known $p = P$

Central line (given $p = P$)

$$CL = P$$

Upper control limit

$$UCL = P + 3\sqrt{P(1-P)/n_i}$$

Lower control limit

$$LCL = P - 3\sqrt{P(1-P)/n_i}$$

Measures of quantity of interest

$$\hat{P}_i = \frac{D_i}{n_i}$$

No standards given: p is unknown

Average fraction of unfit for each batch

$$\hat{p}_i = \frac{D_i}{n_i}$$

P estimator

$$P^* = \frac{\sum_{i=1}^{m} D_i}{\sum_{i=1}^{m} n_i}$$

Central line

$$CL = P^*$$

Upper control limit

$$UCL_i = P^* + 3\sqrt{P^*(1-P^*)/n_i}$$

Lower control limit

$$LCL_i = P^* - 3\sqrt{P^*(1-P^*)/n_i}$$

s chart

Standards given: σ and μ are known

Central line

$$CL = c_4\sigma$$

Upper control limit

$$UCL = c_4\sigma + 3\sigma\sqrt{1 - c_4^2}$$

Lower control limit

$$LCL = c_4\sigma - 3\sigma\sqrt{1 - c_4^2}$$

No standards given: σ is unknown

Central line

$$CL = \hat{s}^* = \sqrt{\frac{\sum_{i=1}^{m}(n_i - 1)\hat{s}_i^2}{\sum_{i=1}^{m} n_i - m}}$$

Upper control limit

$$UCL = \hat{s}^* + 3\frac{\hat{s}^*\sqrt{1 - c_4^2}}{c_4}$$

Lower control limit

$$LCL = \hat{s}^* - 3\frac{\hat{s}^*\sqrt{1 - c_4^2}}{c_4}$$

R chart

Sample size is assumed constant for the R chart

Central line

$$CL = \hat{R} = \sum_{i=1}^{m}\frac{\hat{r}_i}{m}$$

Upper control limit

$$UCL = \hat{R} + 3d_3\frac{\hat{R}}{d_2}$$

Lower control limit

$$LCL = \hat{R} - 3d_3\frac{\hat{R}}{d_2}$$

xbar chart

r method

Sample size is assumed constant for the r method xbar chart. Should the sample size be different then it is strongly suggested to use the s method xbar chart

Central line

$$CL = x^* = \sum_{i=1}^{m}\frac{\hat{x}_i}{m}$$

Upper control limit

$$UCL = x^* + \frac{3\hat{R}}{d_2\sqrt{n}}$$

Lower control limit

$$LCL = x^* - \frac{3\hat{R}}{d_2\sqrt{n}}$$

s method

Standards given: σ and μ are known

Central line

$$CL = \mu$$

Upper control limit

$$UCL = \mu + 3\frac{\sigma}{\sqrt{n}}$$

Lower control limit

$$LCL = \mu - 3\frac{\sigma}{\sqrt{n}}$$

No standards given: σ and μ are unknown

Central line

$$CL = x^* = \frac{\sum_{i=1}^{m} n_i\hat{x}_i}{\sum_{i=1}^{m} n_i}$$

Upper control limit

$$UCL = \hat{x}^* + 3\frac{\hat{\sigma}}{\sqrt{n}} = \hat{x}^* + 3\frac{\hat{s}^*}{c_4\sqrt{n}}$$

Lower control limit

$$LCL = \hat{x}^* - 3\frac{\hat{\sigma}}{\sqrt{n}} = \hat{x}^* - 3\frac{\hat{s}^*}{c_4\sqrt{n}}$$

I and MR chart (Chart for individual observations)

the moving range MR

$$MR_i = |x_i - x_{i-1}|$$

and the average moving range

$$\hat{MR} = \sum_{i=1}^{n-1} \frac{MR_i}{n-1}$$

then:

I chart

Central line

$$CL = \hat{x}$$

Upper control limit

$$UCL = \hat{x} + \frac{3\hat{M}R}{d_2}$$

Lower control limit

$$LCL = \hat{x} - \frac{3\hat{M}R}{d_2}$$

Note: d_2 depends on the moving range size (e.g. if a moving range of $n = 2$ observations is used, then $d_2 = d_2(2)$)

MR chart

Central line

$$CL = \hat{MR}$$

Upper control limit

$$UCL = D_4\hat{MR}$$

Lower control limit

$$LCL = D_3\hat{MR}$$

Note: if a moving range of $n = 2$ observations is used, then $d_2 = d_2(2)$

c chart (Control Chart for Nonconformities)

Constant sample size is assumed for a c chart.

Standards given: c is known (c is the parameter of the Poisson distribution)

Central line

$$CL = c$$

Upper control limit

$$UCL = c + 3\sqrt{c}$$

Lower control limit

$$LCL = max(c - 3\sqrt{c}, 0)$$

Standards not given: c is unknown

Central line

$$CL = \hat{c} = \frac{\sum_{i=1}^{m} D_i}{m}$$

Upper control limit

$$UCL = \hat{c} + 3\sqrt{\hat{c}}$$

Lower control limit

$$LCL = \hat{c} - 3\sqrt{\hat{c}}$$

u chart (Control Chart for Average Number of Nonconformities per Unit)

- $u_i = \frac{D_i}{n_i}$ is the total number of nonconformities in a sample of n_i inspection units.

Then,

Center line

$$CL = \hat{u} = \frac{\sum_{i=1}^{m} D_i}{\sum_{i=1}^{m} n_i}$$

Upper control limit

$$UCL = \hat{u} + 3\sqrt{\frac{\hat{u}}{n_i}}$$

Lower control limit

$$LCL = \hat{u} - 3\sqrt{\frac{\hat{u}}{n_i}}$$

Standardized u chart

Preferred choice with variable sample size

Plotted statistic

$$Z_i = \frac{u_i - \hat{u}}{\sqrt{\frac{\hat{u}}{n_i}}}$$

Central line

$$CL = 0$$

Upper control limit

$$UCL = 3$$

Lower control limit

$$LCL = -3$$

C_4 Bias correction

The c4 constant depends only on the number of samples n and is defined as follows:

$$c_4 = \sqrt{\frac{2}{n-1}\frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})}}$$

where Γ is Euler's Gamma function.

```
#' c4
#' @description Returns bias correction for estimating sigma.
#' @param n number of samples in the current batch
#' @return numeric bias correction
#' @examples
#' biasc2 <-c4(2)
```

```
#' @export

c4 <- function(n)
{
    return(sqrt(2/(n-1))*gamma(n/2)/gamma((n-1)/2))
}

# Plot of correction for n from 2 to 20
#plot(2:20,sapply(2:20,c4),type='l')
```

d_2 constant for estimating σ given the estimated average range $\hat{R}$

$$d_2(n) = \int_{-\infty}^{\infty} [1 - [1 - F_z(z)]^n - F_z^n(z)] \, \mathrm{d}z$$

where $F(z) = \Phi(z)$

Note: samples are assumed to come from a normal distribution. The formula used can be found at the following link

```
#' d2
#' @description Calculates expected value of W for estimating sigma given the average range
#' @param n number of samples in the current batch
#' @return Expected value of W
#' @examples
#' biasd2 <-d2(2)
#' @export

d2 <- function(n)
{
    f <- function(z)
    {
        return(1-(1-pnorm(z))^n-pnorm(z)^n)
    }

    d2_value <- integrate(f,lower=-Inf,upper=Inf)$value

    return(d2_value)
}

# Plot of correction for n from 2 to 20
#plot(2:20,sapply(2:20,d2),type='l')
```

d_3 constant

$$E(W^2) = 2 \int_u^{\infty} \int_{-\infty}^{v} [1 - [1 - F(u)]^n - F^n(v) + [F(v) - F(u)]^n] \, \mathrm{d}u\mathrm{d}v$$

$$d_3 = \sqrt{E(W^2 - d_2^2)}$$

where $F(z) = \Phi(z)$, $u = \frac{x}{\sigma}$ and $v = \frac{y}{\sigma}$

```r
#' d3
#' @description Calculates correction constant for estimating sigma from the range
#' @param n number of samples in the current batch
#' @return Correction constant d3
#' @examples
#' d3_correction <-d3(2)
#' @export

d3 <- function (n)
{
    d2 <- d2(n)
    e <- vector()
    for (i in 1:length(n)) {
        int <- integrate(function(w) {
            w * (1 - ptukey(w, n[i], Inf))
        }, 0, Inf)
        e <- append(e, sqrt(2 * int[[1]] - (d2[1])^2))
    }
    return(e)
}
```

Group definition function

Assumption about the data structure. The data in input is assumed to be a data.frame object. Additionally, the data.frame should have the following structure

```r
#    x1,y1
#    x2,y2
#    ...
#    xn,yn
```

Note that subsequent measurement of the x and y variable have been stacked on top of each other.

```r
#' define_subgroups
#' @description Separates the samples in the data dataframe into a list of subgroups of numerosity n
#' @param data: a dataframe of k samples
#'        n: numerosity of subgroups
#' @return Correction constant d3
#' @examples
#' subgroups <- define_subgroups(data,3)
#' @export

define_subgroups <- function(data,n)
{
    library(dplyr)

    # Number of samples in the data dataframe
    samples <- nrow(data)
    # Then the maximum number of subgroups of numerosity n is:
    m <- floor(samples/n)
    # Remainder (remaining not allocated samples). What if r == 1?
```

```
    r <- samples %% n
    # Total subgroups. r > 0. So if r == 0 m is the number of subgroups
    sg <- m + sign(r)

    ##############################################################################
    # Debug information. Uncomment to print.
    #v <- c(samples,n,sg,r)
    #names(v) <- c("Total samples","Group numerosity","Number of subgroups","Remaining samples")
    #print(v)
    ##############################################################################

    # We can assign a unique id to each sample and then group the sample by id.
    data <- data %>% mutate(sample_group_id = c(rep(1:m,each=n),rep(sg,r)))

    # List to return
    sample_groups <- list()

    # Populate the list
    for(i in 1:sg)
    {
        sample_groups[[i]] <- filter(data,sample_group_id==i)
    }

    return(sample_groups)
}

##############################################################################
# Example. Divide 20 samples into subgroups of numerosity 3
data <- data.frame(x = 1:20, y = 21:40)
define_subgroups(data,3)
```

```
## [[1]]
##   x  y sample_group_id
## 1 1 21               1
## 2 2 22               1
## 3 3 23               1
##
## [[2]]
##   x  y sample_group_id
## 1 4 24               2
## 2 5 25               2
## 3 6 26               2
##
## [[3]]
##   x  y sample_group_id
## 1 7 27               3
## 2 8 28               3
## 3 9 29               3
##
## [[4]]
##    x  y sample_group_id
## 1 10 30               4
## 2 11 31               4
```

```
## 3 12 32               4
##
## [[5]]
##    x  y sample_group_id
## 1 13 33               5
## 2 14 34               5
## 3 15 35               5
##
## [[6]]
##    x  y sample_group_id
## 1 16 36               6
## 2 17 37               6
## 3 18 38               6
##
## [[7]]
##    x  y sample_group_id
## 1 19 39               7
## 2 20 40               7
```

Should the samples be arranged in the following format

```
#         x1 x2 x3 x4 x5
#Sample1  1  2  3 NA NA
#Sample2  4  5  6  7  8
#Sample3 NA  9 10 11 12
```

Then the following function could be used:

```
define_subgroups <- function(data)
{
    sample_groups <- list()

    for(i in 1:nrow(data))
    {
        sample_groups[[i]] <- as.numeric(data[i,])
    }

    return(sample_groups)
}

################################################################################
## Example

d <- data.frame()
d <- rbind(d,c(1,2,3,NA,NA))
d <- rbind(d,c(4:8))
d <- rbind(d,c(NA,9,10,11,12))

colnames(d) <- c("x1","x2","x3","x4","x5")

define_subgroups(d)
```

```
## [[1]]
```

```
## [1]  1  2  3 NA NA
##
## [[2]]
## [1] 4 5 6 7 8
##
## [[3]]
## [1] NA  9 10 11 12
```