Appendix A
Shewhart Constants for Control Charts

The main Shewhart constants d_2, d_3, and c_4 can be obtained for any n using R as shown in the following examples:

```
library(SixSigma)
ss.cc.getd2(n = 5)

  ##       d2
  ## 2.325929

ss.cc.getd3(n = 5)

  ##        d3
  ## 0.8640819

ss.cc.getc4(n = 5)

  ##        c4
  ## 0.9399856
```

The rest of Shewhart constants that can be found at any textbook are computed using those three basic constants. A full table of constants can also be generated using R. Table A.1 shows the constants used in this book. There are other constants not covered by this book which could also be computed just using the appropriate formula. A data frame with the constants in Table A.1 can be obtained with the following code:

```
nmax <- 25
n <- 2:nmax
d2 <- sapply(2:nmax, ss.cc.getd2)
d3 <- sapply(2:nmax, ss.cc.getd3)
c4 <- sapply(2:nmax, ss.cc.getc4)
A2 <- 3/(d2*sqrt(n))
```


E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6

Table A.1 Shewhart constants

n	d2	d3	c4	A2	D3	D4	B3	B4
2	1.1284	0.8525	0.7979	1.8800	0.0000	3.2665	0.0000	3.2665
3	1.6926	0.8884	0.8862	1.0233	0.0000	2.5746	0.0000	2.5682
4	2.0588	0.8798	0.9213	0.7286	0.0000	2.2821	0.0000	2.2660
5	2.3259	0.8641	0.9400	0.5768	0.0000	2.1145	0.0000	2.0890
6	2.5344	0.8480	0.9515	0.4832	0.0000	2.0038	0.0304	1.9696
7	2.7044	0.8332	0.9594	0.4193	0.0757	1.9243	0.1177	1.8823
8	2.8472	0.8198	0.9650	0.3725	0.1362	1.8638	0.1851	1.8149
9	2.9700	0.8078	0.9693	0.3367	0.1840	1.8160	0.2391	1.7609
10	3.0775	0.7971	0.9727	0.3083	0.2230	1.7770	0.2837	1.7163
11	3.1729	0.7873	0.9754	0.2851	0.2556	1.7444	0.3213	1.6787
12	3.2585	0.7785	0.9776	0.2658	0.2833	1.7167	0.3535	1.6465
13	3.3360	0.7704	0.9794	0.2494	0.3072	1.6928	0.3816	1.6184
14	3.4068	0.7630	0.9810	0.2354	0.3281	1.6719	0.4062	1.5938
15	3.4718	0.7562	0.9823	0.2231	0.3466	1.6534	0.4282	1.5718
16	3.5320	0.7499	0.9835	0.2123	0.3630	1.6370	0.4479	1.5521
17	3.5879	0.7441	0.9845	0.2028	0.3779	1.6221	0.4657	1.5343
18	3.6401	0.7386	0.9854	0.1943	0.3913	1.6087	0.4818	1.5182
19	3.6890	0.7335	0.9862	0.1866	0.4035	1.5965	0.4966	1.5034
20	3.7349	0.7287	0.9869	0.1796	0.4147	1.5853	0.5102	1.4898
21	3.7783	0.7242	0.9876	0.1733	0.4250	1.5750	0.5228	1.4772
22	3.8194	0.7199	0.9882	0.1675	0.4345	1.5655	0.5344	1.4656
23	3.8583	0.7159	0.9887	0.1621	0.4434	1.5566	0.5452	1.4548
24	3.8953	0.7121	0.9892	0.1572	0.4516	1.5484	0.5553	1.4447
25	3.9306	0.7084	0.9896	0.1526	0.4593	1.5407	0.5648	1.4352

```
D3 <- sapply(1:(nmax-1), function(x){
      max(c(0, 1 - 3*(d3[x]/d2[x])))})
D4 <- (1 + 3*(d3/d2))
B3 <- sapply(1:(nmax-1), function(x){
      max(0, 1 - 3*(sqrt(1-c4[x]^2)/c4[x]))})
B4 <- 1 + 3*(sqrt(1-c4^2)/c4)
constdf <- data.frame(n, d2, d3, c4, A2,
    D3, D4, B3, B4)
```

The table of constants is also available as a one-page pdf document through one of the `SixSigma` package *vignettes*:

```
vignette(topic = "Shewhart Constants",
         package = "SixSigma")
```

Appendix B
ISO Standards Published by the ISO/TC69: Application of Statistical Methods

This appendix contains all the international standards and technical reports published by the ISO TC69—Application of Statistical Methods, grouped by subcommittees. Please note that ISO standards are continously evolving. All references to standards in this appendix and throughout the book are specific for a given point in time. In particular, this point in time is end of June 2015. Therefore, some new standards may have appeared when you are reading this book, or even other changes may have happen in ISO. For example, at the time of publishing a subcommittee has changed its denomination! Keep updated in the committee website: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=49742.

TC69/SCS: Secretariat

ISO 11453:1996 Statistical interpretation of data—Tests and confidence intervals relating to proportions.
ISO 11453:1996/Cor 1:1999 .
ISO 16269-4:2010 Statistical interpretation of data—Part 4: Detection and treatment of outliers.
ISO 16269-6:2014 Statistical interpretation of data—Part 6: Determination of statistical tolerance intervals.
ISO 16269-7:2001 Statistical interpretation of data—Part 7: Median—Estimation and confidence intervals.
ISO 16269-8:2004 Statistical interpretation of data—Part 8: Determination of prediction intervals.
ISO 2602:1980 Statistical interpretation of test results—Estimation of the mean—Confidence interval.
ISO 2854:1976 Statistical interpretation of data—Techniques of estimation and tests relating to means and variances.

E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6

ISO 28640:2010 Random variate generation methods.
ISO 3301:1975 Statistical interpretation of data—Comparison of two means in the case of paired observations.
ISO 3494:1976 Statistical interpretation of data—Power of tests relating to means and variances.
ISO 5479:1997 Statistical interpretation of data—Tests for departure from the normal distribution.
ISO/TR 13519:2012 Guidance on the development and use of ISO statistical publications supported by software.
ISO/TR 18532:2009 Guidance on the application of statistical methods to quality and to industrial standardization.

TC69/SC1: Terminology and Symbols

Statistics
ISO 3534-1:2006 —Vocabulary and symbols—Part 1: General statistical terms and terms used in probability.
ISO 3534-2:2006 Statistics—Vocabulary and symbols—Part 2: Applied statistics.
ISO 3534-3:2013 Statistics—Vocabulary and symbols—Part 3: Design of experiments.
ISO 3534-4:2014 Statistics—Vocabulary and symbols—Part 4: Survey sampling.

TC69/SC4: Applications of Statistical Methods in Process Management

ISO 11462-1:2001 Guidelines for implementation of statistical process control (SPC)—Part 1: Elements of SPC.
ISO 11462-2:2010 Guidelines for implementation of statistical process control (SPC)—Part 2: Catalogue of tools and techniques.
ISO 22514-1:2014 Statistical methods in process management—Capability and performance—Part 1: General principles and concepts.
ISO 22514-2:2013 Statistical methods in process management—Capability and performance—Part 2: Process capability and performance of time-dependent process models.
ISO 22514-3:2008 Statistical methods in process management—Capability and performance—Part 3: Machine performance studies for measured data on discrete parts.

ISO 22514-6:2013 Statistical methods in process management—Capability and performance—Part 6: Process capability statistics for characteristics following a multivariate normal distribution.
ISO 22514-7:2012 Statistical methods in process management—Capability and performance—Part 7: Capability of measurement processes.
ISO 22514-8:2014 Statistical methods in process management—Capability and performance—Part 8: Machine performance of a multi-state production process.
ISO 7870-1:2014 Control charts—Part 1: General guidelines.
ISO 7870-2:2013 Control charts—Part 2: Shewhart control charts.
ISO 7870-3:2012 Control charts—Part 3: Acceptance control charts.
ISO 7870-4:2011 Control charts—Part 4: Cumulative sum charts.
ISO 7870-5:2014 Control charts—Part 5: Specialized control charts.
ISO/TR 22514-4:2007 Statistical methods in process management—Capability and performance—Part 4: Process capability estimates and performance measures.

TC69/SC5: Acceptance Sampling

ISO 13448-1:2005 Acceptance sampling procedures based on the allocation of priorities principle (APP)—Part 1: Guidelines for the APP approach.
ISO 13448-2:2004 Acceptance sampling procedures based on the allocation of priorities principle (APP)—Part 2: Coordinated single sampling plans for acceptance sampling by attributes.
ISO 14560:2004 Acceptance sampling procedures by attributes—Specified quality levels in nonconforming items per million.
ISO 18414:2006 Acceptance sampling procedures by attributes—Accept-zero sampling system based on credit principle for controlling outgoing quality.
ISO 21247:2005 Combined accept-zero sampling systems and process control procedures for product acceptance.
ISO 24153:2009 Random sampling and randomization procedures.
ISO 2859-10:2006 Sampling procedures for inspection by attributes—Part 10: Introduction to the ISO 2859 series of standards for sampling for inspection by attributes.
ISO 2859-1:1999 Sampling procedures for inspection by attributes—Part 1: Sampling schemes indexed by acceptance quality limit (AQL) for lot-by-lot inspection.
ISO 2859-1:1999/Amd 1:2011 .
ISO 2859-3:2005 Sampling procedures for inspection by attributes—Part 3: Skip-lot sampling procedures.
ISO 2859-4:2002 Sampling procedures for inspection by attributes—Part 4: Procedures for assessment of declared quality levels.

ISO 2859-5:2005 Sampling procedures for inspection by attributes—Part 5: System of sequential sampling plans indexed by acceptance quality limit (AQL) for lot-by-lot inspection.

ISO 28801:2011 Double sampling plans by attributes with minimal sample sizes, indexed by producer's risk quality (PRQ) and consumer's risk quality (CRQ).

ISO 3951-1:2013 Sampling procedures for inspection by variables—Part 1: Specification for single sampling plans indexed by acceptance quality limit (AQL) for lot-by-lot inspection for a single quality characteristic and a single AQL.

ISO 3951-2:2013 Sampling procedures for inspection by variables—Part 2: General specification for single sampling plans indexed by acceptance quality limit (AQL) for lot-by-lot inspection of independent quality characteristics.

ISO 3951-3:2007 Sampling procedures for inspection by variables—Part 3: Double sampling schemes indexed by acceptance quality limit (AQL) for lot-by-lot inspection.

ISO 3951-4:2011 Sampling procedures for inspection by variables—Part 4: Procedures for assessment of declared quality levels.

ISO 3951-5:2006 Sampling procedures for inspection by variables—Part 5: Sequential sampling plans indexed by acceptance quality limit (AQL) for inspection by variables (known standard deviation).

ISO 8422:2006 Sequential sampling plans for inspection by attributes.

ISO 8423:2008 Sequential sampling plans for inspection by variables for percent nonconforming (known standard deviation).

TC69/SC6: Measurement Methods and Results

ISO 10576-1:2003 Statistical methods—Guidelines for the evaluation of conformity with specified requirements—Part 1: General principles.

ISO 10725:2000 Acceptance sampling plans and procedures for the inspection of bulk materials.

ISO 11095:1996 Linear calibration using reference materials.

ISO 11648-1:2003 Statistical aspects of sampling from bulk materials—Part 1: General principles.

ISO 11648-2:2001 Statistical aspects of sampling from bulk materials—Part 2: Sampling of particulate materials.

ISO 11843-1:1997 Capability of detection—Part 1: Terms and definitions.

ISO 11843-2:2000 Capability of detection—Part 2: Methodology in the linear calibration case.

ISO 11843-3:2003 Capability of detection—Part 3: Methodology for determination of the critical value for the response variable when no calibration data are used.

ISO 11843-4:2003 Capability of detection—Part 4: Methodology for comparing the minimum detectable value with a given value.

ISO 11843-5:2008 Capability of detection—Part 5: Methodology in the linear and non-linear calibration cases.

ISO 11843-6:2013 Capability of detection—Part 6: Methodology for the determination of the critical value and the minimum detectable value in Poisson distributed measurements by normal approximations.

ISO 11843-7:2012 Capability of detection—Part 7: Methodology based on stochastic properties of instrumental noise.

ISO 21748:2010 Guidance for the use of repeatability, reproducibility and trueness estimates in measurement uncertainty estimation.

ISO 5725-1:1994 Accuracy (trueness and precision) of measurement methods and results—Part 1: General principles and definitions.

ISO 5725-2:1994 Accuracy (trueness and precision) of measurement methods and results—Part 2: Basic method for the determination of repeatability and reproducibility of a standard measurement method.

ISO 5725-3:1994 Accuracy (trueness and precision) of measurement methods and results—Part 3: Intermediate measures of the precision of a standard measurement method.

ISO 5725-4:1994 Accuracy (trueness and precision) of measurement methods and results—Part 4: Basic methods for the determination of the trueness of a standard measurement method.

ISO 5725-5:1998 Accuracy (trueness and precision) of measurement methods and results—Part 5: Alternative methods for the determination of the precision of a standard measurement method.

ISO 5725-6:1994 Accuracy (trueness and precision) of measurement methods and results—Part 6: Use in practice of accuracy values.

ISO/TR 13587:2012 Three statistical approaches for the assessment and interpretation of measurement uncertainty.

ISO/TS 21749:2005 Measurement uncertainty for metrological applications—Repeated measurements and nested experiments.

ISO/TS 28037:2010 Determination and use of straight-line calibration functions.

TC69/SC7: Applications of Statistical and Related Techniques for the Implementation of Six Sigma

ISO 13053-1:2011 Quantitative methods in process improvement—Six Sigma—Part 1: DMAIC methodology.

ISO 13053-2:2011 Quantitative methods in process improvement—Six Sigma—Part 2: Tools and techniques.

ISO 17258:2015 Statistical methods—Six Sigma—Basic criteria underlying benchmarking for Six Sigma in organisations.

ISO/TR 12845:2010 Selected illustrations of fractional factorial screening experiments.

ISO/TR 12888:2011 Selected illustrations of gauge repeatability and reproducibility studies.
ISO/TR 14468:2010 Selected illustrations of attribute agreement analysis.
ISO/TR 29901:2007 Selected illustrations of full factorial experiments with four factors.
ISO/TR 29901:2007/Cor 1:2009 .

TC69/SC8: Application of Statistical and Related Methodology for New Technology and Product Development

ISO 16336:2014 Applications of statistical and related methods to new technology and product development process—Robust parameter design (RPD).

Appendix C
R Cheat Sheet for Quality Control

R Console

↑ ↓ Navigate expressions history
`CTRL+L` Clear console
`ESC` Cancel current expression

RStudio

`CTRL + number` Go to panel:

- 1: Editor
- 2: Console
- 3: Help
- 4: History
- 5: Files
- 6: Plots
- 7: Packages
- 8: Environment

`CTRL + MAYÚS + K` *knit* current R Markdown report
`CTRL + MAYÚS + I` Compile R Sweave (LaTeX) current report
`CTRL + S` Save file
`F1` Contextual help (upon the cursor position)
`CTRL + F` Activates search (within different panels)[1]
`<console>`

[1] See 'Edit' menu for further options.

E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6

↑ ↓ Expressions history
CTRL+L Clear console
ESC Cancel current expression
<editor and console>
TAB Prompt menu:

- Select objects in the workspace
- Select function arguments (when in parenthesis)
- Select list elements (after the $ character)
- Select chunk options (when in *chunk* header)
- Select files (when in quotes)

<editor>
CTRL + ENTER Run current line or selection
CTRL + MAYÚS + S Source full script
CTRL + ALT + I Insert code *chunk*
CTRL + ALT + C Run current code *chunk* (within a chunk)
CTRL + MAYÚS + P Repeat las code run
CTRL + MAYÚS + C Comment current line or selection (add # at the beginning of the line)
CTRL + D Delete current line
ALT + ↑ ↓ Move current line or selection up or down
ALT + MAYÚS + ↑ ↓ Copy current line or selection up or down

Help

?, help Help on a function

```
help("mean")
?mean
```

??, help.search Search help over a topic

```
help.search("topic")
```

apropos Show function containing a given string

```
apropos("prop.test")

  ## [1] "pairwise.prop.test" "power.prop.test"
  ## [3] "prop.test"
```

General

`;` Separate expressions in the same line
`<-` Assignment operator
`{ <code> }` Code blocks within curly brackets
Comment (ignores the remaining of the line)
`` `<string>` `` (backtick) Allow using identifiers with special characters and/or blank spaces

```
?(`[`)
`my var` <- 1:5
`my var`
```

Math Operators

`+, - /, *, ^` Arithmetic

```
5 + 2
  ## [1] 7
pi - 3
  ## [1] 0.1415927
1:5 * 2
  ## [1]  2  4  6  8 10
3 / 1:3
  ## [1] 3.0 1.5 1.0
3^4
  ## [1] 81
```

`<, >, <=, >=, ==, !=, %in%` Comparisons

```
5 >= 3
```

```
  ## [1] TRUE
5 %in% 1:4
  ## [1] FALSE
"a" %in% letters
  ## [1] TRUE
3.14 != pi
  ## [1] TRUE
```

`&, &&, |, ||, !` Logic operations[2]

```
5 >= 3 | 8 > 10
  ## [1] TRUE
5 >= 3 & 8 > 10
  ## [1] FALSE
1:2 < 3 & 3:4 > 2
  ## [1] TRUE TRUE
1:2 < 3 && 3:4 > 2
  ## [1] TRUE
```

Integer Operations

`%/%` Integer division

```
15 %/% 2
  ## [1] 7
```

`%%` Module (remainder of a division)

```
15 %% 2
  ## [1] 1
```

[2]Double operators && and || are used to compare vectors globally. Single operators, element-wise.

Comparison Functions

all Are all elements TRUE?

```
all(1 > 2, 1 <2)
  ## [1] FALSE
```

any Is any element TRUE?

```
any(1 > 2, 1 < 2)
  ## [1] TRUE
```

Math Functions

sqrt Square root

```
sqrt(16)
  ## [1] 4
```

exp, log Exponential and logarithmic

```
exp(-5)
  ## [1] 0.006737947
log(5)
  ## [1] 1.609438
```

sin, cos, tan Trigonometry

```
sin(pi)
  ## [1] 1.224647e-16
```

asin, acos, atan Inverse trigonometry

```
asin(1)
  ## [1] 1.570796
```

abs Absolute value

```
abs(log(0.5))

  ## [1] 0.6931472
```

round, floor, ceiling Rounding

```
round(5.5)

  ## [1] 6

floor(5.5)

  ## [1] 5

ceiling(5.4)

  ## [1] 6
```

max, min Maximum and minimum

```
x <- 1:10
max(x)

  ## [1] 10

min(x)

  ## [1] 1
```

sum, prod Sums and products

```
sum(x)

  ## [1] 55

prod(x)

  ## [1] 3628800
```

cumsum, cumprod, cummax, cummin Cumulative operations

```
cumsum(x)

  ##  [1]  1  3  6 10 15 21 28 36 45 55

cumprod(1:5)

  ## [1]   1   2   6  24 120
```

factorial Factorial

```
factorial(5)
  ## [1] 120
```

choose Binomial coefficient

```
choose(5,3)
  ## [1] 10
```

Vectors

c Create a vector (combine values)

```
svec <- c(1, 2, 5, 7, 4); svec
  ## [1] 1 2 5 7 4
```

seq : Creates a sequence

```
seq(4, 11, 2)
  ## [1]  4  6  8 10
4:11
  ## [1]  4  5  6  7  8  9 10 11
```

rep Repeat values

```
rep(1:2, each = 2)
  ## [1] 1 1 2 2
rep(1:2, times = 2)
  ## [1] 1 2 1 2
```

length Vector length

```
length(svec)
  ## [1] 5
```

[ ] Item selection

```
x[3]
  ## [1] 3
x[-3]
  ## [1]  1  2  4  5  6  7  8  9 10
```

sort Sorting

```
svec
  ## [1] 1 2 5 7 4
sort(svec)
  ## [1] 1 2 4 5 7
```

order Get indices ordered by magnitude

```
order(svec)
  ## [1] 1 2 5 3 4
```

rev Reverse order

```
rev(sort(svec))
  ## [1] 7 5 4 2 1
```

unique Get unique values

```
x2 <- c(1, 2, 2, 3, 4, 5, 5); x2
  ## [1] 1 2 2 3 4 5 5
unique(x2)
  ## [1] 1 2 3 4 5
```

which Devuelve índices que cumplen condición

```
which(x2 == 5)
  ## [1] 6 7
```

union, intersect, setdiff, setequal, %in% Sets operations

```
union(1:3, 3:5)

  ## [1] 1 2 3 4 5

intersect(1:3, 3:5)

  ## [1] 3

setdiff(1:3, 3:5)

  ## [1] 1 2

setequal(1:3, 3:5)

  ## [1] FALSE

3 %in% 1:3

  ## [1] TRUE
```

Matrices

matrix Create a matrix

```
A <- matrix(1:4, nrow=2); A

  ##      [,1] [,2]
  ## [1,]    1    3
  ## [2,]    2    4

B <- matrix(1:2, ncol=1); B

  ##      [,1]
  ## [1,]    1
  ## [2,]    2
```

`%*%` Matrix multiplication

```
A %*% B

  ##      [,1]
  ## [1,]    7
  ## [2,]   10
```

t Transpose a matrix

```
t(A)

  ##      [,1] [,2]
  ## [1,]    1    2
  ## [2,]    3    4
```

solve Inverse a matrix

```
solve(A)

  ##      [,1] [,2]
  ## [1,]   -2  1.5
  ## [2,]    1 -0.5
```

colSums, rowSums Sum by rows or columns

```
colSums(A)

  ## [1] 3 7
```

colMeans, rowMeans Average by rows or columns

```
rowMeans(B)

  ## [1] 1 2
```

colnames, rownames Column or rows names

```
colnames(A) <- c("col1", "col2"); A

  ##      col1 col2
  ## [1,]    1    3
  ## [2,]    2    4
```

dim, nrow, ncol Dimensions

```
dim(A)

  ## [1] 2 2

nrow(A)

  ## [1] 2

ncol(A)

  ## [1] 2
```

rbind, cbind Add columns or rows to a matrix matrix

```
rbind(A, 10:11)

  ##      col1 col2
  ## [1,]    1    3
  ## [2,]    2    4
  ## [3,]   10   11

cbind(B, 10:11)

  ##      [,1] [,2]
  ## [1,]    1   10
  ## [2,]    2   11
```

`[ , ]` Items selection

```
A[1, ] # row

  ## col1 col2
  ##    1    3

A[, 1] # column

  ## [1] 1 2

A[1, 2]#cell

  ## col2
  ##    3

A[1, , drop = FALSE]

  ##      col1 col2
  ## [1,]    1    3
```

Factors

factor Create a factor

```
xf <- factor(rep(1:2, 2)); xf

  ## [1] 1 2 1 2
  ## Levels: 1 2
```

gl Generate levels of a factor

```
xgl <- gl(3, 2, labels = LETTERS[1:3])
```

expand.grid Generate factors combinations

```
my.factors <- expand.grid(xf, xgl)
```

Dates

as.Date Convert to date

```
my.date <- as.Date("10/06/2014",
    format("%d/%m/%Y")); my.date

  ## [1] "2014-06-10"
```

format Returns a date in a given format

```
format(my.date, "%m-%y")

  ## [1] "06-14"
```

ISOweek Returns the week of a date in ISO format (`ISOweek` package)

```
library(ISOweek)
ISOweek(my.date)

  ## [1] "2014-W24"
```

Character String

nchar Get number of characters

```
my.string <- "R is free software"
nchar(my.string)

  ## [1] 18
```

paste, paste0 *Paste* character strings

```
your.string <-   "as in Beer"
paste(my.string, your.string)

  ## [1] "R is free software as in Beer"
```

cat Print a character string in the console

```
cat("Hello World!")

  ## Hello World!
```

Lists

list Create a list

```
my.list <- list(a_string = my.string,
    a_matrix = A,
    a_vector = svec)
my.list

  ## $a_string
  ## [1] "R is free software"
  ##
  ## $a_matrix
  ##      col1 col2
  ## [1,]    1    3
  ## [2,]    2    4
  ##
  ## $a_vector
  ## [1] 1 2 5 7 4

my.list$a_vector

  ## [1] 1 2 5 7 4

my.list[1]

  ## $a_string
  ## [1] "R is free software"

my.list[[1]]

  ## [1] "R is free software"
```

Data Frames

data.frame Create a data frame

```
my.data <- data.frame(variable1 = 1:10,
    variable2 = letters[1:10],
    group = rep(1:2, each = 5))
my.data$variable1
```

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

```
my.data[2, ]
```

```
  ##   variable1 variable2 group
  ## 2         2         b     1
```

str Get data frame structure: column names, types, and sample data

```
str(my.data)
```

```
  ## 'data.frame': 10 obs. of  3 variables:
  ##  $ variable1: int  1 2 3 4 5 6 7 8 9 10
  ##  $ variable2: Factor w/ 10 levels "a","b","c","..
  ##  $ group    : int  1 1 1 1 1 2 2 2 2 2
```

head, tail Get first or last rows of a data frame

```
head(my.data)
```

```
  ##   variable1 variable2 group
  ## 1         1         a     1
  ## 2         2         b     1
  ## 3         3         c     1
  ## 4         4         d     1
  ## 5         5         e     1
  ## 6         6         f     2
```

```
tail(my.data)
```

```
  ##    variable1 variable2 group
  ## 5          5         e     1
  ## 6          6         f     2
  ## 7          7         g     2
  ## 8          8         h     2
  ## 9          9         i     2
  ## 10        10         j     2
```

subset Get a (filtered) subset of data

```
subset(my.data, group == 1)

  ##   variable1 variable2 group
  ## 1         1         a     1
  ## 2         2         b     1
  ## 3         3         c     1
  ## 4         4         d     1
  ## 5         5         e     1
```

aggregate Get aggregate data applying a function over groups

```
aggregate(variable1 ~ group, my.data, mean)

  ##   group variable1
  ## 1     1         3
  ## 2     2         8
```

Files

download.file Download files

```
download.file(
    url = "http://emilio.lcano.com/qcrbook/lab.csv",
    destfile = "lab.csv")
```

read.table Import data

```
importedData <- read.table("lab.csv",
    header = TRUE,
    sep = ",",
    dec = ".")
```

read.csv2 Import data from csv file

```
importedData <- read.csv("lab.csv")
```

write.csv2 Save csv data file

```
write.csv2(importedData,
    file = "labnew.csv",
    row.names = FALSE)
```

scan Read data from the console or text

```
scannedVector <- scan()
typedData <- scan(text = "1 2 3 4 5 6")
```

save Save an R data file

```
save(importedData, file = "lab.RData")
```

load Load an R data file into the workspace

```
load("lab.RData")
```

Data Simulation

set.seed Fix the seed[3]

```
set.seed(1234)
```

sample Draw a random sample from a set

```
sample(letters, 5)

  ## [1] "c" "p" "o" "x" "s"

sample(1:6, 10, replace = TRUE)

  ##  [1] 4 1 2 4 4 5 4 2 6 2
```

rnorm, rbinom, rpois, ... Draw random variates from probability distributions (normal, binomial, Poisson, ...)

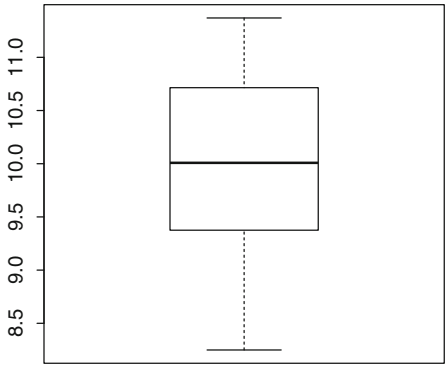
```
snorm <- rnorm(20, mean = 10, sd = 1)
spois <- rpois(20, lambda = 3)
```

[3]This makes results reproducible.

Graphics

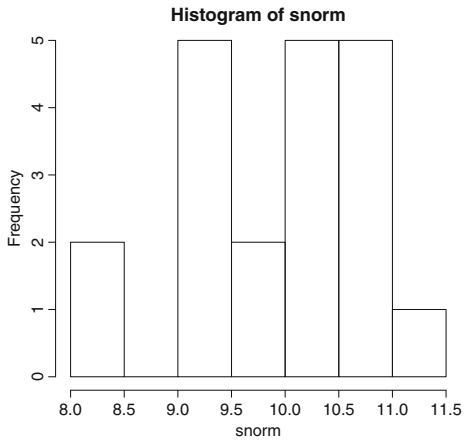
boxplot Box plot

```
boxplot(snorm)
```



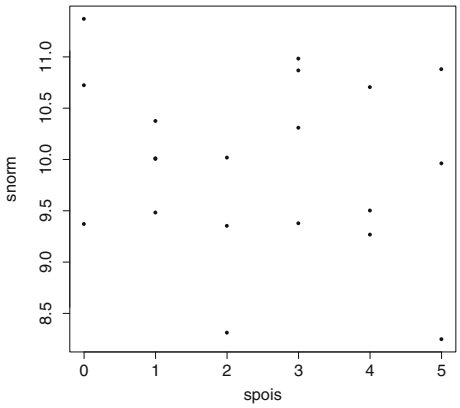
hist Histogram

```
hist(snorm)
```



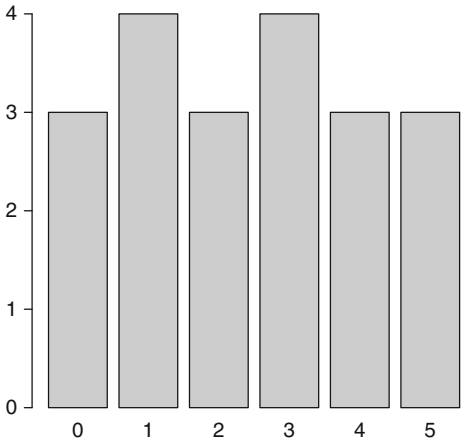
plot Scatter plot (for two numeric vectors)

```
plot(spois, snorm, pch = 20, )
```



barplot Bar plot (for counts)

```
barplot(table(spois))
```

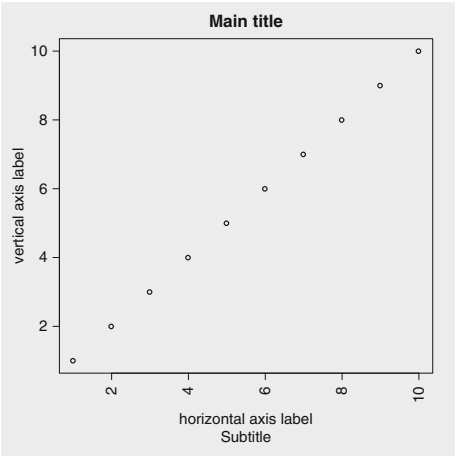


par Graphical parameters (see `?par`)

par Get or set graphical parameters
main Add a title to a plot (top)
sub Add a subtitle to a plot (bottom)
xlab, ylab Set horizontal and vertical axes labels
legend Add a legend
col Set color (see link at the end)
las Axes labels orientation
lty Line type
lwd Line width
pch Symbol (for points)

```
par(bg = "gray90")
```

```
plot(1:10, main = "Main title", sub = "Subtitle",
     xlab = "horizontal axis label",
     ylab = "vertical axis label",
     las = 2)
```

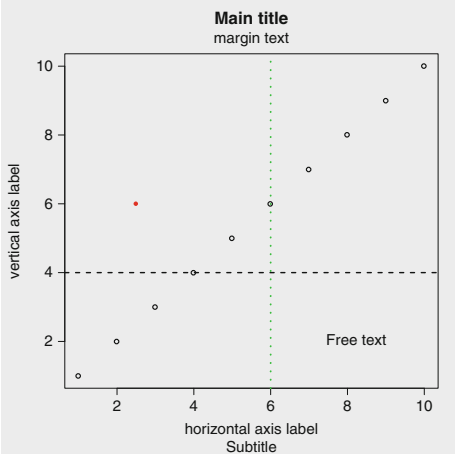



```
par(bg = "white")
```

graphics Graphical functions

points Add points to a plot
abline Draw a straight line (horizontal, vertical, or with a slope)
text Put text in the plot
mtext Add text in the margins

```
par(bg = "gray90")
plot(1:10, main = "Main title", sub = "Subtitle",
     xlab = "horizontal axis label",
     ylab = "vertical axis label",
     las = 2)
par(bg = "white")
points(x = 2.5, y = 6, col = "red", pch = 16)
abline(h = 4, lty = 2, lwd = 2)
abline(v = 6, lty = 3, lwd = 3, col = 3)
text(x = 8, y = 2, labels = "Free text")
mtext(text = "margin text", side = 3)
```



Descriptive Statistics

table Count the elements within each category

```
table(spois)

  ## spois
  ## 0 1 2 3 4 5
  ## 3 4 3 4 3 3
```

summary Five-num summary (plus the mean)

```
summary(snorm)

  ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##   8.249   9.376  10.010   9.957  10.710  11.370
```

mean Average

```
mean(snorm)

  ## [1] 9.956699
```

median Median

```
median(snorm)

  ## [1] 10.0082
```

quantile Percentiles, quantiles

```
quantile(snorm, 0.1)

  ##       10%
  ## 9.172885
```

var Variance

```
var(snorm)

  ## [1] 0.7240189
```

sd Standard deviation

```
sd(snorm)

  ## [1] 0.850893
```

cor Correlation

```
cor(snorm, spois)

  ## [1] -0.1691365
```

max, min, range Maximum, minimum, range

```
max(x)

  ## [1] 10

min(x)

  ## [1] 1

range(x)

  ## [1]  1 10

diff(range(x))

  ## [1] 9
```

Acceptance Sampling

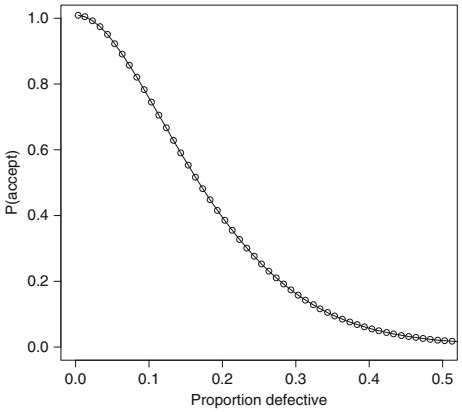
- Simple sampling plan

```
x <- OC2c(10,1); x

  ## Acceptance Sampling Plan (binomial)
  ##
  ##                 Sample 1
```

```
  ## Sample size(s)        10
  ## Acc. Number(s)         1
  ## Rej. Number(s)         2

plot(x, xlim=c(0,0.5))
```

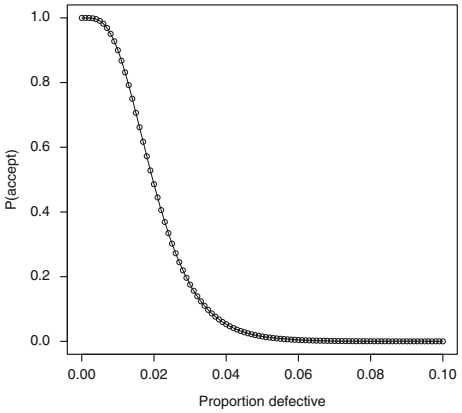


- Double sampling plan

```
x <- OC2c(c(125,125), c(1,4), c(4,5),
    pd = seq(0,0.1,0.001)); x

  ## Acceptance Sampling Plan (binomial)
  ##
  ##                 Sample 1 Sample 2
  ## Sample size(s)       125      125
  ## Acc. Number(s)         1        4
  ## Rej. Number(s)         4        5

plot(x)
```



- Assess plan

```
assess(x, PRP=c(0.01, 0.95), CRP=c(0.05, 0.04))
```

```
## Acceptance Sampling Plan (binomial)
##
##                 Sample 1 Sample 2
## Sample size(s)       125      125
## Acc. Number(s)         1        4
## Rej. Number(s)         4        5
##
## Plan CANNOT meet desired risk point(s):
##
##             Quality   RP P(accept) Plan P(accept)
## PRP            0.01           0.95     0.89995598
## CRP            0.05           0.04     0.01507571
```

Control Charts

qcc Library

```
library(qcc)
data(pistonrings)
str(pistonrings)
```

```
## 'data.frame': 200 obs. of  3 variables:
##  $ diameter: num  74 74 74 74 74 ...
##  $ sample  : int  1 1 1 1 1 2 2 2 2 2 ...
##  $ trial   : logi  TRUE TRUE TRUE TRUE TRUE TRUE..
```

```
head(pistonrings)
```

```
##   diameter sample trial
## 1   74.030      1  TRUE
## 2   74.002      1  TRUE
## 3   74.019      1  TRUE
## 4   73.992      1  TRUE
## 5   74.008      1  TRUE
## 6   73.995      2  TRUE
```

```
table(pistonrings$trial)
```

```
##
## FALSE  TRUE
##    75   125
```

```
str(qcc)

  ## function (data, type = c("xbar", "R", "S",
  ##     "xbar.one", "p", "np", "c", "u", "g"),
  ##     sizes, center, std.dev, limits, data.name,
  ##     labels, newdata, newsizes, newlabels,
  ##     nsigmas = 3, confidence.level, rules = shewh..
  ##     plot = TRUE, ...)
```

qcc.groups Create object with grouped data

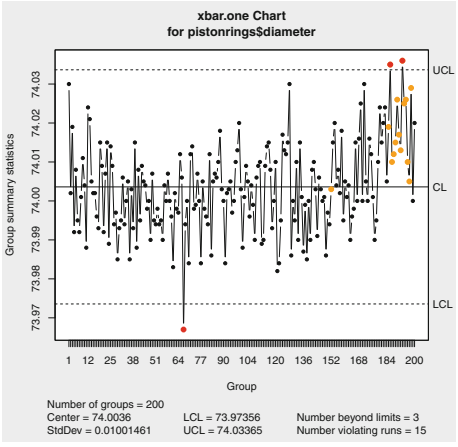
```
my.groups <- qcc.groups(data = pistonrings$diameter,
    sample = pistonrings$sample)
```

qcc Create control chart object. Some options:

data Vector, matrix or data frame with the data
type One of: “xbar”, “R”, “S”, “xbar.one”, “p”, “np”, “c”, “u”, “g”
sizes Vector with sample sizes for charts: “p”, “np”, o “u”
center Known center value
std.dev Known standard deviation
limits Phase I limits (vector with LCL, UCL)
plot If FALSE the chart is not shown
newdata Phase II data
newsizes Phase II sample sizes
nsigmas Number of standard deviations to compute control limits
confidence.level Confidence level to compute control limits (instead of nsigmas)

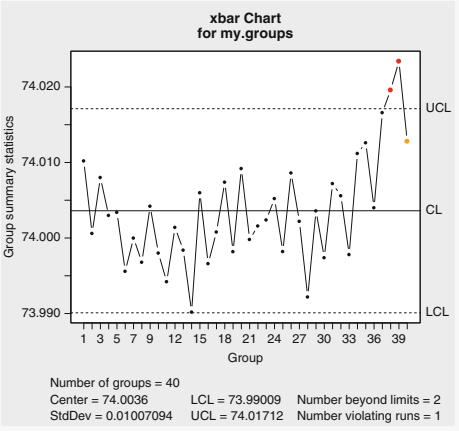
Control charts for variables:

```
# Individual values chart
qcc(pistonrings$diameter, type = "xbar.one")
```



```
  ## List of 11
  ##  $ call      : language qcc(data = pistonrings"..
  ##  $ type      : chr "xbar.one"
  ##  $ data.name : chr "pistonrings$diameter"
  ##  $ data      : num [1:200, 1] 74 74 74 74 74 ...
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ statistics: Named num [1:200] 74 74 74 74 74..
  ##   ..- attr(*, "names")= chr [1:200] "1" "2" "3"..
  ##  $ sizes     : int [1:200] 1 1 1 1 1 1 1 1 1 1 ..
  ##  $ center    : num 74
  ##  $ std.dev   : num 0.01
  ##  $ nsigmas   : num 3
  ##  $ limits    : num [1, 1:2] 74 74
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ violations:List of 2
  ##  - attr(*, "class")= chr "qcc"
```

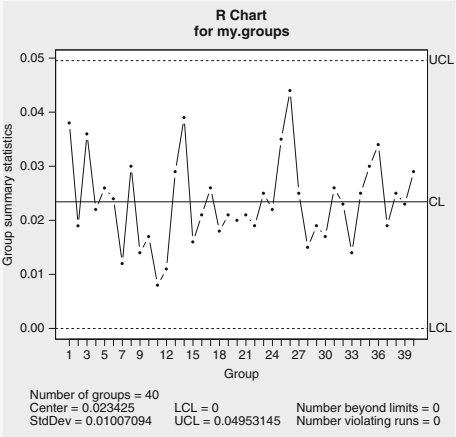
```
# X-bar chart
qcc(my.groups, type = "xbar")
```




```
  ## List of 11
  ##  $ call      : language qcc(data = my.groups, "..
  ##  $ type      : chr "xbar"
  ##  $ data.name : chr "my.groups"
  ##  $ data      : num [1:40, 1:5] 74 74 74 74 74 ...
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ statistics: Named num [1:40] 74 74 74 74 74 ..
  ##   ..- attr(*, "names")= chr [1:40] "1" "2" "3""..
  ##  $ sizes     : Named int [1:40] 5 5 5 5 5 5 5 5..
  ##   ..- attr(*, "names")= chr [1:40] "1" "2" "3""..
```

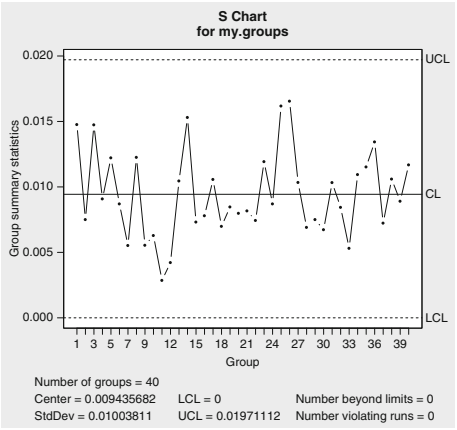
```
##  $ center    : num 74
##  $ std.dev   : num 0.0101
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 74 74
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
# Range chart
qcc(my.groups, type = "R")
```




```
## List of 11
##  $ call      : language qcc(data = my.groups, "..
##  $ type      : chr "R"
##  $ data.name : chr "my.groups"
##  $ data      : num [1:40, 1:5] 74 74 74 74 74 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:40] 0.038 0.019 0.0..
##   ..- attr(*, "names")= chr [1:40] "1" "2" "3""..
##  $ sizes     : Named int [1:40] 5 5 5 5 5 5 5 5..
##   ..- attr(*, "names")= chr [1:40] "1" "2" "3""..
##  $ center    : num 0.0234
##  $ std.dev   : num 0.0101
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0 0.0495
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
# S chart
qcc(my.groups, type = "S")
```



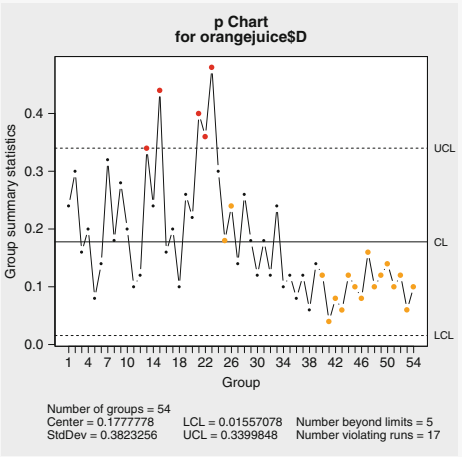

```
## List of 11
##  $ call      : language qcc(data = my.groups, "..
##  $ type      : chr "S"
##  $ data.name : chr "my.groups"
##  $ data      : num [1:40, 1:5] 74 74 74 74 74 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:40] 0.01477 0.0075 ..
##   ..- attr(*, "names")= chr [1:40] "1" "2" "3""..
##  $ sizes     : Named int [1:40] 5 5 5 5 5 5 5 5..
##   ..- attr(*, "names")= chr [1:40] "1" "2" "3""..
##  $ center    : num 0.00944
##  $ std.dev   : num 0.01
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0 0.0197
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

Control charts for attributes:

```
# p chart
data(orangejuice)
str(orangejuice)
```

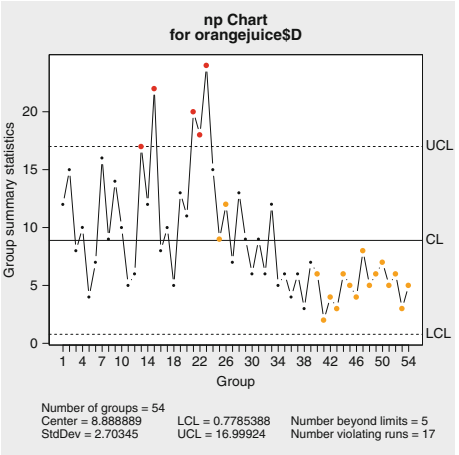
```
## 'data.frame': 54 obs. of  4 variables:
##  $ sample: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ D     : int  12 15 8 10 4 7 16 9 14 10 ...
##  $ size  : int  50 50 50 50 50 50 50 50 50 50 ...
##  $ trial : logi  TRUE TRUE TRUE TRUE TRUE TRUE ..
```

```
qcc(orangejuice$D, sizes = orangejuice$size,
    type = "p")
```




```
  ## List of 11
  ##  $ call      : language qcc(data = orangejuice"..
  ##  $ type      : chr "p"
  ##  $ data.name : chr "orangejuice$D"
  ##  $ data      : int [1:54, 1] 12 15 8 10 4 7 16 ..
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ statistics: Named num [1:54] 0.24 0.3 0.16 0..
  ##   ..- attr(*, "names")= chr [1:54] "1" "2" "3""..
  ##  $ sizes     : int [1:54] 50 50 50 50 50 50 50 ..
  ##  $ center    : num 0.178
  ##  $ std.dev   : num 0.382
  ##  $ nsigmas   : num 3
  ##  $ limits    : num [1, 1:2] 0.0156 0.34
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ violations:List of 2
  ##  - attr(*, "class")= chr "qcc"

# np chart
qcc(orangejuice$D, sizes = orangejuice$size,
    type = "np")
```

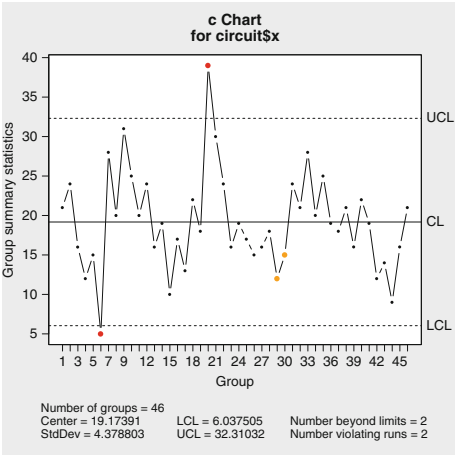



```
  ## List of 11
  ##  $ call      : language qcc(data = orangejuice"..
  ##  $ type      : chr "np"
  ##  $ data.name : chr "orangejuice$D"
  ##  $ data      : int [1:54, 1] 12 15 8 10 4 7 16 ..
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ statistics: Named int [1:54] 12 15 8 10 4 7 ..
  ##   ..- attr(*, "names")= chr [1:54] "1" "2" "3""..
  ##  $ sizes     : int [1:54] 50 50 50 50 50 50 50 ..
  ##  $ center    : num 8.89
  ##  $ std.dev   : num 2.7
  ##  $ nsigmas   : num 3
  ##  $ limits    : num [1, 1:2] 0.779 16.999
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ violations:List of 2
  ##  - attr(*, "class")= chr "qcc"

# chart for counts
data(circuit)
str(circuit)

  ## 'data.frame': 46 obs. of  3 variables:
  ##  $ x    : int  21 24 16 12 15 5 28 20 31 25 ...
  ##  $ size : int  100 100 100 100 100 100 100 100 ..
  ##  $ trial: logi  TRUE TRUE TRUE TRUE TRUE TRUE ...

qcc(circuit$x, sizes = circuit$size, type = "c")
```

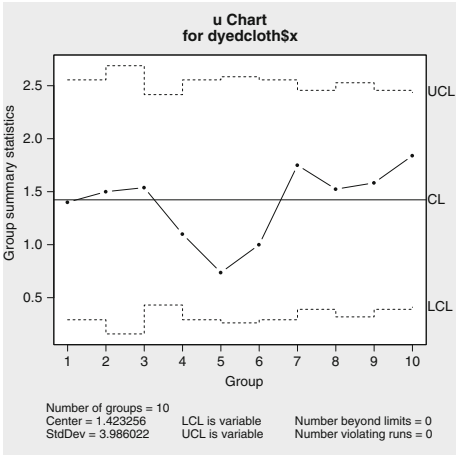



```
  ## List of 11
  ##  $ call      : language qcc(data = circuit$x, "..
  ##  $ type      : chr "c"
  ##  $ data.name : chr "circuit$x"
  ##  $ data      : int [1:46, 1] 21 24 16 12 15 5 2..
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ statistics: Named int [1:46] 21 24 16 12 15 ..
  ##   ..- attr(*, "names")= chr [1:46] "1" "2" "3""..
  ##  $ sizes     : int [1:46] 100 100 100 100 100 1..
  ##  $ center    : num 19.2
  ##  $ std.dev   : num 4.38
  ##  $ nsigmas   : num 3
  ##  $ limits    : num [1, 1:2] 6.04 32.31
  ##   ..- attr(*, "dimnames")=List of 2
  ##  $ violations:List of 2
  ##  - attr(*, "class")= chr "qcc"

# Chart for counts per unit
data(dyedcloth)
str(dyedcloth)

  ## 'data.frame': 10 obs. of  2 variables:
  ##  $ x   : int  14 12 20 11 7 10 21 16 19 23
  ##  $ size: num  10 8 13 10 9.5 10 12 10.5 12 12.5

qcc(dyedcloth$x, sizes = dyedcloth$size, type = "u")
```



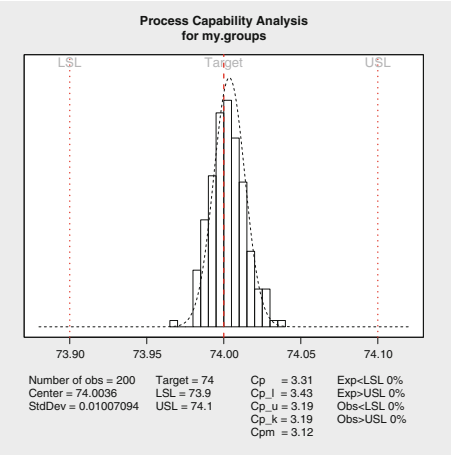

```
## List of 11
##  $ call      : language qcc(data = dyedcloth$x"..
##  $ type      : chr "u"
##  $ data.name : chr "dyedcloth$x"
##  $ data      : int [1:10, 1] 14 12 20 11 7 10 2..
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:10] 1.4 1.5 1.538 1..
##   ..- attr(*, "names")= chr [1:10] "1" "2" "3""..
##  $ sizes     : num [1:10] 10 8 13 10 9.5 10 12 ..
##  $ center    : num 1.42
##  $ std.dev   : num 3.99
##  $ nsigmas   : num 3
##  $ limits    : num [1:10, 1:2] 0.291 0.158 0.43..
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

Process Capability

qcc Package

process.capability Needs a qcc object

```
my.qcc <- qcc(my.groups, type = "xbar", plot = FALSE)
process.capability(my.qcc,
    spec.limits = c(73.9, 74.1),
    target = 74)
```

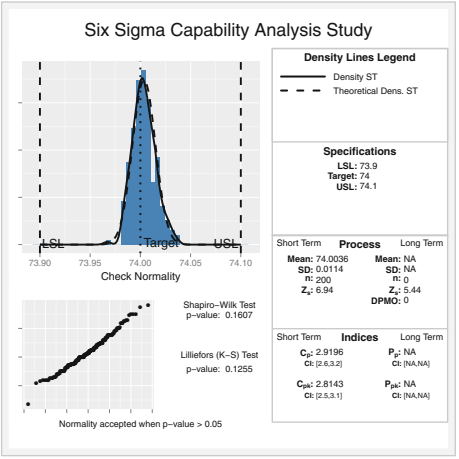



```
##
## Process Capability Analysis
##
## Call:
## process.capability(object = my.qcc, spec.limits
    = c(73.9, 74.1),     target = 74)
##
## Number of obs = 200          Target = 74
##        Center = 74              LSL = 73.9
##        StdDev = 0.01007         USL = 74.1
##
## Capability indices:
##
##       Value   2.5%  97.5%
## Cp    3.310  2.985  3.635
## Cp_l  3.429  3.144  3.715
## Cp_u  3.191  2.925  3.456
## Cp_k  3.191  2.874  3.507
## Cpm   3.116  2.794  3.438
##
## Exp<LSL 0%  Obs<LSL 0%
## Exp>USL 0%  Obs>USL 0%
```

`SixSigma` Package

ss.study.ca Returns graphical and numerical capability analysis

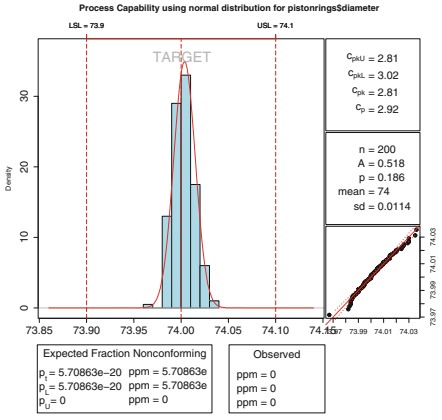
```
ss.study.ca(pistonrings$diameter,
    LSL = 73.9, USL = 74.1, Target = 74)
```



`qualityTools` Package

cp Process capability indices

```
library(qualityTools)
cp(x = pistonrings$diameter,
    lsl = 73.9, usl = 74.1,
    target = 74)
```



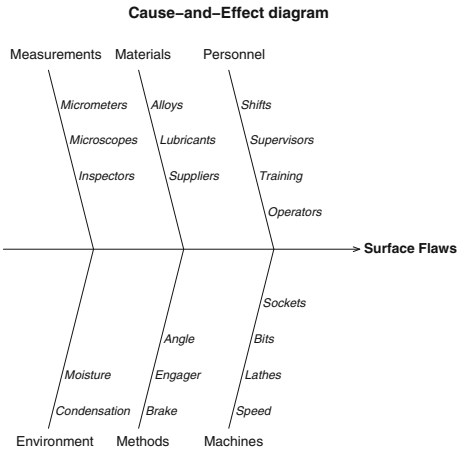
```
##
##  Anderson Darling Test for normal
##  distribution
##
## data:  pistonrings$diameter
## A = 0.5181, mean = 74.004, sd = 0.011,
## p-value = 0.1862
## alternative hypothesis: true distribution is not
     equal to normal
```

Pareto Analysis

qcc Package

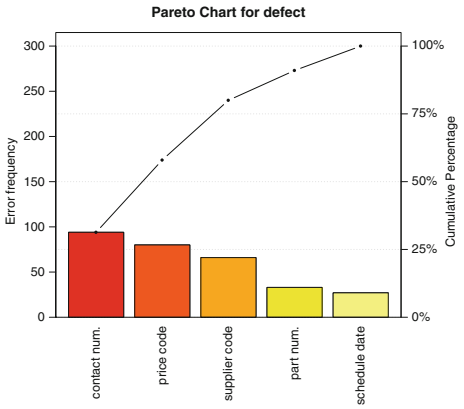
cause.and.effect Cause-and-effect analysis

```
cause.and.effect(cause = list(
        Measurements=c("Micrometers", "Microscopes",
            "Inspectors"),
        Materials=c("Alloys", "Lubricants",
            "Suppliers"),
        Personnel=c("Shifts", "Supervisors",
            "Training", "Operators"),
        Environment=c("Condensation", "Moisture"),
        Methods=c("Brake", "Engager", "Angle"),
        Machines=c("Speed", "Lathes", "Bits",
            "Sockets")),
    effect = "Surface Flaws")
```



pareto.chart Pareto Chart

```
defect <- c(80, 27, 66, 94, 33)
names(defect) <- c("price code", "schedule date",
    "supplier code", "contact num.", "part num.")
pareto.chart(defect, ylab = "Error frequency")
```

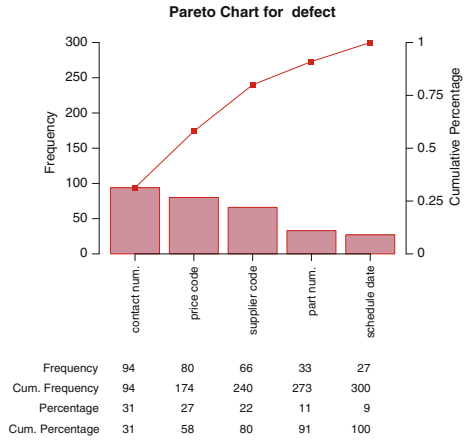



```
##
## Pareto chart analysis for defect
##                  Frequency Cum.Freq. Percentage
##   contact num.          94        94   31.33333
##   price code            80       174   26.66667
##   supplier code         66       240   22.00000
##   part num.             33       273   11.00000
##   schedule date         27       300    9.00000
##
## Pareto chart analysis for defect
##                  Cum.Percent.
##   contact num.       31.33333
##   price code         58.00000
##   supplier code      80.00000
##   part num.          91.00000
##   schedule date     100.00000
```

`qualityTools` Package

paretoChart Pareto chart

```
paretoChart(defect, las = 2)
```



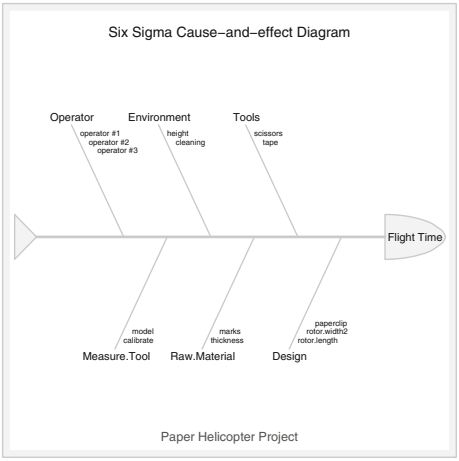

```
##
## Frequency           94     80     66     33      27
## Cum. Frequency      94    174    240    273     300
## Percentage       31.3% 26.7% 22.0% 11.0%    9.0%
## Cum. Percentage 31.3% 58.0% 80.0% 91.0% 100.0%
##
## Frequency       94.00000  80.00000  66  33  27
## Cum. Frequency  94.00000 174.00000 240 273 300
## Percentage      31.33333  26.66667  22  11   9
## Cum. Percentage 31.33333  58.00000  80  91 100
```

`SixSigma` Package

ss.ceDiag Cause-and-effect diagram

```
effect <- "Flight Time"
causes.gr <- c("Operator", "Environment", "Tools",
    "Design", "Raw.Material", "Measure.Tool")
causes <- vector(mode = "list",
    length = length(causes.gr))
causes[1] <- list(c("operator #1", "operator #2",
        "operator #3"))
causes[2] <- list(c("height", "cleaning"))
causes[3] <- list(c("scissors", "tape"))
causes[4] <- list(c("rotor.length", "rotor.width2",
        "paperclip"))
causes[5] <- list(c("thickness", "marks"))
causes[6] <- list(c("calibrate", "model"))
```

```
ss.ceDiag(effect, causes.gr, causes,
    sub = "Paper Helicopter Project")
```



Probability

p* Distribution function for a given value

```
pnorm(q = 8, mean = 10, sd = 1)

  ## [1] 0.02275013

## help("distributions") for further distributions
```

q* Quantile for a given cumulative probability probabilidad «»= qnorm(p = 0.95, mean = 10, sd = 1) @

d* Density for a given value (probability in discrete distributions)

```
dpois(2, lambda = 3)

  ## [1] 0.2240418
```

Objects

str Get the structure of an object

```
str(log)

  ## function (x, base = exp(1))

str(xgl)

  ##  Factor w/ 3 levels "A","B","C": 1 1 2 2 3 3
```

class Get the class of an object

```
class(xgl)

  ## [1] "factor"
```

is.* Return a logic value `TRUE` if the object is of the specified class, for example, numeric

as.* Coerce to the specified class

```
as.character(1:3)

  ## [1] "1" "2" "3"

as.data.frame(A)

  ##   col1 col2
  ## 1    1    3
  ## 2    2    4
```

Vectorized Functions

tapply Apply a function to values for each level of a factor

```
tapply(pistonrings$diameter, pistonrings$trial,
summary)

  ## $`FALSE`
  ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##   73.98   74.00   74.00   74.01   74.02   74.04
  ##
  ## $`TRUE`
  ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##   73.97   73.99   74.00   74.00   74.01   74.03
```

lapply Apply a function to each element of a list returning a list

```
lapply(1:3, factorial)
```

```
  ## [[1]]
  ## [1] 1
  ##
  ## [[2]]
  ## [1] 2
  ##
  ## [[3]]
  ## [1] 6
```

sapply Apply a function to each element of a list returning a list,vector, or matrix

```
sapply(1:3, factorial)

  ## [1] 1 2 6
```

apply Apply a function to the rows or columns of a matrix

```
apply(A, 1, median)

  ## [1] 2 3
```

split Divide an object over factor levels returning a list

```
groups <- split(pistonrings$diameter, pistonrings$
  trial)
str(groups)

  ## List of 2
  ##  $ FALSE: num [1:75] 74 74 74 74 74 ...
  ##  $ TRUE : num [1:125] 74 74 74 74 74 ...
```

mappy Multivariate version of `sapply`

```
mapply(rep, 1:4, 4:1)

  ## [[1]]
  ## [1] 1 1 1 1
  ##
  ## [[2]]
  ## [1] 2 2 2
  ##
  ## [[3]]
  ## [1] 3 3
  ##
  ## [[4]]
  ## [1] 4
```

rapply Recursive version of `lapply`

```
X <- list(list(a = pi, b = list(c = 1:1)), d =
 "a test")
rapply(X, sqrt, classes = "numeric", how = "replace")
```

```
## [[1]]
## [[1]]$a
## [1] 1.772454
```

```
##
## [[1]]$b
## [[1]]$b$c
## [1] 1
##
##
##
## $d
## [1] "a test"
```

Programming

for Loop over the values of a vector or list

```
x <- numeric()
for (i in 1:3){
  x[i] <- factorial(i)
}
x
```

```
## [1] 1 2 6
```

if . . . else Control flow

```
if (is.numeric(x)){
  cat("Is numeric")
} else if (is.character(x)){
  cat("Is character")
} else{
  cat("Is another thing")
}
```

```
## Is numeric
```

function Create functions

```
# Function that computes the difference between
two vectors' means
mifuncion <- function(x, y){
  mean(x) - mean(y)
}
mifuncion(1:10, 11:20)

  ## [1] -10
```

Useful functions within a function:

warning
```
warning("This is a warning")
## Warning: This is a warning
```

message
```
message("This is a message")
## This is a message
```

stop Stops the execution of code

```
stop("An error occurs")

## Error in eval(expr, envir, enclos): An error occurs
```

Reports

`xtable` Package

xtable Create tables in different formats, e.g., LaTeX, HTML

caption Table caption
label Table label
align Alignment
digits Number of significant digits
display Format (see `?xtable`)

More options can be passed to the print generic function `?print.xtable`

```
library(xtable)
xtable(A)
```

	col1	col2
1	1	3
2	2	4

Package `knitr`

knit Converts Rmd, Rhtml and Rnw files into HTML, MS Word o PDF reports.
See documentación at http://yihui.name/knitr/.
Main options in a code chunk header:

echo Show code in the report
error Show error messages in the report
warning Show warning messages in the report
message Show messages in the report
eval Evaluate the chunk
fig.align Figure alignment
fig.width Figure width (in inches, 7 by default)
fig.height Figure height (in inches, 7 by default)
out.width Figure width within the report
out.height Figure height within the report
fig.keep Keep plots in the report
include Show text output in the report
results How to show the reports

Useful Links

- R-Project: http://www.r-project.org
- RStudio: http://www.rstudio.com
- Easy R practice: http://tryr.codeschool.com/
- List of colours with names: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
- http://www.cyclismo.org/tutorial/R/
- http://www.statmethods.net/index.html
- *Recipes*: http://www.cookbook-r.com/
- Search documentation: http://www.rdocumentation.org/
- http://www.computerworld.com/s/article/9239625/Beginner_s_guide_to_R_Introduction
- http://www.inside-r.org/
- http://www.r-bloggers.com/
- Google R styleguide: http://google-styleguide.googlecode.com/svn/trunk/Rguide.xml
- Book *Six Sigma with R*: www.sixsigmawithr.com

R Packages and Functions Used in the Book

E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6

ISO Standards Referenced in the Book

E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6

Subject Index

E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6