Appendix A
R Basic Reference Guide

This appendix is intended to provide a brief but broad collection of functions commonly used in R. See references and Sect. 2.8 in Chap. 2 to study R in depth. Type `help(foo)` in the R Console to see the documentation and the complete list of arguments for the function `foo`.

Remember that R is case sensitive, and `a` is not equal to `A`.

Table A.1 Help functions

Function	Action
`help()` or ?	Opens help about a topic
`help.search()` or ??	Returns list of topics containing some text
`apropos()`	Returns list of functions containing some text
`find()`	Tells what package something is in
`find()`	Shows which libraries and dataframes are attached
`example()`	Runs example for a topic
`demo()`	Runs demo about a topic
`args()`	Returns syntax of a function
`attach()`	Makes variables in a dataframe accessible by name
`detach()`	For a dataframe, stops accessibility by name; for a package, unloads it

Table A.2 Functions for packages and workspace

Function	Action
`library()`	Loads package or shows available package
`save.image()`	Saves all objects of R session
`save()`	Saves list of objects to file
`load()`	Loads objects from file system to R session
`rm()`	Removes objects
`source()`	Runs script in file
`sink()`	Sends output to text file instead of R Console
`savehistory()`	Saves command history of session
`loadhistory()`	Loads previously saved history
`q()`	Exits R

E.L. Cano et al., *Six Sigma with R: Statistical Engineering for Process Improvement*, Use R! 36, DOI 10.1007/978-1-4614-3652-2,

Table A.3 Constants and special values

Expression	Value
`letters`	26 lowercase letters
`LETTERS`	26 uppercase letters
`month.abb`	Three-letter abbreviation of months
`month.name`	English names of months
`pi`	π constant
`NA`	Not available
`NaN`	Not a number (e.g. $\frac{0}{0}$)
`Inf`	Infinite
`NULL`	No value
`1i`	Imaginary constant $\sqrt{-1}$

Table A.4 Operators for objects

Operator	Operation
`<-`	Assigns value
`[ ]`	Subscript vector
`[[ ]]`	Subscript list
`:`	Generates sequences
$\sim$	Model formula
`$`	List indexing

Table A.5 Functions for data types

Create	Verify	Convert	Description
`character()`	`is.character()`	`as.character()`	String type
`numeric()`	`is.numeric()`	`as.numeric()`	Numeric type
`logical()`	`is.logical()`	`as.logical()`	Logical data
`vector()`	`is.vector()`	`as.vector()`	One-dimensional data
`matrix()`	`is.matrix()`	`as.matrix()`	Two-dimensional data
`array()`	`is.array()`	`as.array()`	Any dimensional data
`factor()`	`is.factor()`	`as.factor()`	Categorical variable in a vector
`list()`	`is.list()`	`as.list()`	Collection of objects of different types
`data.frame()`	`is.data.frame()`	`as.data.frame()`	Data set as a list of vectors
`integer()`	`is.integer()`	`as.integer()`	Integers
`ts()`	`is.ts()`	`as.ts()`	Time-series objects
`complex()`	`is.complex()`	`as.complex()`	Complex vector (real and imaginary parts)
	`class()`		Returns the class of any object

Table A.6 Functions for importing and exporting data

Function	Action
`read.table()`	Imports text files
`read.csv(), read.csv2()`	Imports csv files
`read.delim()`	Imports delimited text files
`read.spss()`	Imports SPSS files
`read.mtp()`	Imports Minitab files
`read.arff()`	Imports Weka files (Data mining)
`read.dta()`	Imports Stata binary files
`write.table()`	Writes a text file
`write.csv()`	Writes a csv file
`writeClipboard()`	Saves data in Clipboard to paste in other applications

Table A.7 Functions for managing data

Function	Action
`c()`	Concatenates elements
`seq()`	Generates regular sequences
`rep()`	Replicates values
`length()`	Gets or sets length of an object
`sort()`	Sorts vectors
`rev()`	Reverses vectors
`order()`	Returns permutation of vector indices
`rank()`	Returns position of each vector element
`unique()`	Returns unique values of a vector
`duplicated()`	Returns indices that contain duplicated values
`which()`	Gives TRUE indices of logical object
`levels()`	Returns levels of a factor
`unlist()`	Converts a list into a vector
`cut()`	Converts a vector into a factor
`transform()`	Transforms a dataframe
`aggregate()`	Gets subtotals of variables in data set
`subset()`	Gets subset of data set

Table A.8 String functions

Function	Action
`nchar()`	Counts number of characters in a string
`substr()`	Returns a substring
`paste()`	Builds a string with substrings

Table A.9 Functions for exploring data

Function	Action
`data()`	Loads a data set
`head()`	Returns first rows of a data set
`tail()`	Returns last rows of a data set
`str()`	Returns structure of data set
`attributes()`	Accesses object attributes

Table A.10 Operators

Operator	Meaning
`+, -, *, /`	Arithmetic operators
`%/%`	Integer division
`%%`	Remainder of a division (modulo)
`%*%`	Matrix multiplication
`^`	Powers
`>, >=, <, <=, ==, !=`	Relational operators
`!, &, |`	Logical operators

Table A.11 Set operations

Function	Action
`union()`	Set union
`intersection()`	Set intersection
`setdiff()`	Set difference
`setequal()`	Set equality
`is.element()`	Set membership

Table A.12 Mathematical functions

Function	Meaning
`sqrt()`	Square root
`log(), exp()`	Logarithmic and exponential functions
`sin(), cos(), tan(), asin(), acos(), atan()`	Trigonometry functions
`cosh(), sinh(), tanh(), acosh(), asinh(), atanh()`	Hyperbolic functions
`round(), trunc(), ceiling(), floor()`	Different ways to round a number
`abs()`	Returns absolute value of a number

Table A.13 Vector functions

Function	Meaning
`max(), min()`	Maximum and minimum values in vector
`sum(), prod()`	Sum and product
`mean(), var(), sd()`	Sample mean, variance, and standard deviation
`median()`	Median value in vector
`range()`	Vector with a maximum and minimum
`cor()`	Correlation between vectors
`cumsum(), cumprod(), cummax(), cummin()`	Cumulative functions
`colMeans(), colSums(), rowMeans(), rowSums()`	Totals by rows or columns for dataframes and matrices
`ifelse()`	Returns a value for each component depending on a logical expression

Table A.14 Summarize functions

Function	Action
`apply`	Applies function to margins in matrix or vector
`lapply`	Applies function to all elements of list or vector
`tapply`	Applies function to ragged array
`sapply`	Friendly version of lapply
`summary`	Returns summary of object; for data objects, some statistics
`table`	Returns data frequency table

Table A.15 Probability distributions

Name	Probability	Random	Quantile	Density	Parameters	Default
Binomial	`pbinom()`	`rbinom()`	`qbinom`	`dbinom`	size, prob	
Normal	`pnorm()`	`rnorm()`	`qnorm()`	`dnorm()`	mean, sd	0,1
Uniform	`punif()`	`runif()`	`qunif()`	`dunif()`	min, max	0,1
Poisson	`ppois()`	`rpois()`	`qpois()`	`dpois()`	lambda	
Exponential	`pexp()`	`rexp()`	`qexp()`	`dexp()`	rate	1
t-Student	`pt()`	`rt()`	`qt()`	`dt()`	df	
Chi-square	`pchisq()`	`rchisq()`	`qchisq()`	`dchisq()`	df	
F	`pf()`	`rf()`	`qf()`	`df()`	df1, df2	
Beta	`pbeta()`	`rbeta()`	`qbeta()`	`dbeta()`	shape1, shape2	
Gamma	`pgamma()`	`rgamma()`	`qgamma()`	`dgamma()`	shape, scale	`NULL`, 1
Cauchy	`pcauchy()`	`rcauchy()`	`qcauchy`	`dcauchy`	location, scale	0,1
Weibull	`pweibull()`	`rweibull()`	`qweibull`	`dweibull`	shape, scale	`NULL`, 1
Tukey	`ptukey()`	—	`qtukey`	—	nmeans, nranges	`NULL`, 1

Table A.16 Sampling and combinatorial

Function	Action
`set.seed()`	Sets seed for replicable randomization
`sample()`	Extracts sample from a set
`choose()`	Computes binomial coefficient

Table A.17 Graphic functions

Function	Action
`plot()`	Produces different types of plots
`boxplot()`	Produces boxplot
`hist()`	Produces histogram
`coplot()`	Conditional plots
`dotchart()`	Produces dot chart
`stripchart()`	Produces 1-D scatterplot
`mosaicplot()`	Produces mosaic plot
`matplot()`	Plots matrix columns
`pairs()`	Matrix of scatterplots
`persp()`	Produces perspective plot (three dimensions)
`spineplot()`	Produces spine plot and spinogram
`text()`	Superposes text on a plot
`abline()`	Prints line in plot
`lines()`	Superposes line on a plot
`points()`	Superposes points on a plot
`polygon()`	Draws polygon
`xspline()`	Draws an X-spline
`identify()`	Identifies point in a plot with mouse

Table A.18 Plot arguments

Argument	Meaning
`type`	Type: p(points), l(lines), b(both), h(histogram), s(steps)
`main`	Main title
`sub`	Subtitle
`xlab`	Label for *x* axis
`ylab`	Label for *y* axis
`asp`	*yx* aspect ratio

Table A.19 Graphic parameters

Parameter	Description	Values
`col`	Color	Names, numbers, hex codes
`lwd`	Line width	Number
`lty`	Line type	Number
`pch`	Point symbol	Number
`las`	Style of axis labels	Number
`cex`	Magnifying ratio	Number (1 default)

Table A.20 Hypothesis tests and confidence intervals

Function	Task
`binom.test`	Exact test for binomial distributions
`bartlett.test`	Tests equal variances
`chisq.test`	Test for congingency tables and goodness-of-fit tests
`lillie.test`	Normality test
`pairwise.t.test`	Pairwise comparison with correction for multiple testing
`poisson.test`	Test for Poisson distribution parameter
`prop.test`	Test for proportions
`shapiro.test`	Normality test
`t.test`	Test for means
`var.test`	Test for comparing variances

Table A.21 Model fitting

Package	Function	Models
`lm`	`stats`	Linear model
`anova`	`stats`	Gets ANOVA table for a model
`manova`	`stats`	Performs multivariate ANOVA
`glm`	`stats`	Generalized linear model
`gam`	`gam`	Generalized additive model
`gam`	`mgcv`	Generalized additive model
`lm.ridge`	`MASS`	Linear models by ridge regression
`coef`	`stats`	Extracts coefficients from a model
`predict`	`stats`	Computes predictions for data input based on a model
`confint`	`stats`	Gets a confidence interval for model parameters
`residuals`	`stats`	Gets residuals of a model
`step`	`stats`	Selects variables for a model
`update`	`stats`	Updates and refits formula of a model

Table A.22 `SixSigma` package data sets

Function	Task
`ss.data.bolts`	Bolts example, Chap. 4
`ss.data.ca`	Winery example, Chap. 7
`ss.data.doe1`, `ss.data.doe2`	Pizza dough example, Chap. 11
`ss.data.pastries`	Pastries exercise, Chap. 5
`ss.data.pc`, `ss.data.pc.big`, `ss.data.pc.r`	Print cartriges example, Chap. 8
`ss.data.strings`	Guitar strings example, Chaps. 9 and 10
`ss.data.rr`	Data for a paper helicopter experiment

Table A.23 `SixSigma` package functions

Function	Task
`ss.ceDiag`	Cause-and-effect diagram
`ss.pMap`	Process map
`ss.lfa`	Loss function analysis
`ss.lfa`	Computes loss function value
`ss.rr`	Gage R&R study
`ss.study.ca`	Capability analysis study
`ss.ca.cp`	Capability index
`ss.ca.cpk`	Corrected capability index
`ss.ca.yield`	Computes yield of a process
`ss.ci`	Confidence interval for mean and normality test

Table A.24 Cited packages

Package	Use	Reference
base, stats, utils, grid, graphics, tools	Base packages	[84]
AcceptanceSampling	Acceptance sampling	[52]
AMORE	Neural network modeling	[62]
e1071	SVM, data mining, classification	[21]
kernlab	Data mining, classification, SVM	[48]
foreign	Data format conversion	[83]
gam	General additive model (inference)	[37]
ggplot2	Elegant graphics using grammar of graphics	[105]
Hmisc	Harrell miscellaneous (useful functions for many tasks by Frank E. Harrell)	[35]
knitr	Report generation	[111]
lattice	Lattice graphics	[91]
lpridge	Ridge regression	[95]
MASS	Support functions and data sets for Venables and Ripley's MASS	[102]
mgcv	Generalized additive and other models	[109]
monmlp	Neural network modeling	[14]
neuralnet	Neural network modeling	[27]
nnet	Neural network and multinomial log-linear models	[102]
nortest	Normality test	[33]
odfWeave	Reporting	[57]
plan	Gantt chart	[50]
plotrix	Gantt chart and other plots	[61]
pls	Partial least squares (PLS) regression	[67]
plsdof	Partial least squares (PLS) regression	[56]
plspm	Partial least squares (PLS) regression	[90]
plsRbeta	Partial least squares (PLS) regression	[26]
plsRcox	Partial least squares (PLS) regression	[6]
plsRglm	Partial least squares (PLS) regression	[7]
qAnalyst	Control chart and capability analysis	[98]
qcc	Quality control chart	[93]
qualityTools	Methods associated with DMAIC cycle	[88]
RODBC	Data source connection	[87]
RWeka	SVM and other data mining tools	[39, 108]
SixSixgma	Six Sigma tools and data sets for examples in this book	[15]
sp	Spatial data analysis	[78]
spc	Statistical process control computations	[54]
spcadjust	Calibration of control charts	
survival	Survival analysis (including ridge regression); reliability studies	[100]
XLConnect	Import/export, manipulate Microsoft Excel files	[29]
xtable	Elegant tables for reports and publications	[19]

References

1. Agresti, A., & Coull, B. A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *American Statistician*, *52*, 119–126.
2. Allen, T. T. (2010). *Introduction to engineering statistics and lean Six Sigma—Statistical quality control and design of experiments and systems*. New York: Springer.
3. Arthur, J. (2006). *Lean six sigma demystified. Demystified series*. New York: McGraw-Hill.
4. Bentley, W., & Davis, P. (2009). *Lean Six Sigma secrets for the CIO: ITIL, COBIT, and beyond*. USA: CRC Press.
5. Berger, P., & Maurer, R. (2002). *Experimental design: With applications in management, engineering, and the sciences. Duxbury titles of related interest*. CA: Duxbury/Thomson Learning.
6. Bertrand, F., Maumy-Bertrand, M., & Meyer, N. (2011) *Partial least squares regression for Cox models and related techniques*. http://CRAN.R-project.org/package=plsRcox, r package version 0.2.0.
7. Bertrand, F., Meyer, N., & Maumy-Bertrand, M. (2011). *Partial least squares regression for generalized linear models*. http://CRAN.R-project.org/package=plsRglm, r package version 0.7.6.
8. Bivand, R., Pebesma, E., & Gómez-Rubio, V. (2008). *Applied spatial data analysis with R. Use R!*. New York: Springer.
9. Box G (1992). Teaching engineers experimental design with a paper helicopter. *Quality Engineering*, *4*(3), 453–459.
10. Box, G., & Jones, S. (1992). Designing products that are robust to the environment. *Total Quality Management*, *3*(3), 265–285.
11. Box, G., Hunter, J., & Hunter, W. (2005). *Statistics for experimenters: Design, innovation, and discovery. Wiley series in probability and statistics*. New York: Wiley.
12. Burdick, R., Borror, C., & Montgomery, D. (2005). *Design and analysis of gauge R&R studies: Making decisions with confidence intervals in random and mixed ANOVA models. ASA-SIAM series on statistics and applied probability*. Philadelphia: Society for Industrial Applied Mathematics.
13. Burton, T. T., & Sams, J. L. (2005). *Six Sigma for small and mid-sized organizations: Success through scaleable deployment*. Boca Rotan: J. Ross Publishing.
14. Cannon, A. J. (2011). *monmlp: Monotone multi-layer perceptron neural network*. http://CRAN.R-project.org/package=monmlp, r package version 1.1.
15. Cano, E. L., Moguerza, J. M., & Redchuk, A. (2011). *SixSigma: Six Sigma tools for quality improvement*. R package version 0.6.0.
16. Chambers, J. M. (2008). *Software for data analysis. Programming with R. Statistics and computing*. Berlin: Springer.

E.L. Cano et al., *Six Sigma with R: Statistical Engineering for Process Improvement*, Use R! 36, DOI 10.1007/978-1-4614-3652-2,

17. Cook, D., & Swayne, D. (2007). *Interactive and dynamic graphics for data analysis: With R and GGobi. Use R!*. New York: Springer.
18. Crawley, M. J. (2007). *The R Book*. UK: Wiley.
19. Dahl, D. B. (2009). *xtable: Export tables to LaTeX or HTML*. http://CRAN.R-project.org/package=xtable, r package version 1.5-6.
20. Dalgaard, P. (2008). *Introductory statistics with R. Statistics and computing*. New York: Springer.
21. Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., & Weingessel, A. (2011). *e1071: Misc functions of the department of statistics (e1071), TU Wien*. http://CRAN.R-project.org/package=e1071, r package version 1.6.
22. Eckes, G. (2003). *Six sigma for everyone*. New Jersey: Wiley.
23. El-Haik, B., & Roy, D. (2005). *Service design for six sigma: A roadmap for excellence*. New York: Wiley.
24. Faraway, J. (2006). *Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models. Texts in statistical science*. Boca Raton: Chapman & Hall/CRC. http://books.google.es/books?id=ODcRsWpGji4C.
25. Faraway, J. J. (2002). *Practical regression and anova using r*. http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf. Accessed 01.08.2011.
26. Bertrand, F., & Myriam Maumy-Bertrand N. M. (2011). *plsRbeta: Partial least squares regression for beta regression models*. http://CRAN.R-project.org/package=plsRbeta, r package version 0.1.1.
27. Fritsch, S., Guenther, F., & following earlier work by Marc Suling (2010). *neuralnet: Training of neural networks*. http://CRAN.R-project.org/package=neuralnet, r package version 1.31.
28. Gandy, A., & Kvaloy, J. T. (2011). *spcadjust: Functions for calibrating control charts*. http://CRAN.R-project.org/package=spcadjust, r package version 0.1-1.
29. GmbH MS (2011). *XLConnect: Excel Connector for R*. http://CRAN.R-project.org/package=XLConnect, r package version 0.1-7.
30. Graham, B. (2004). *Detail process charting: Speaking the language of process*. New Jersey: Wiley.
31. Grömping, U. (2011). *Cran task view: Design of experiments (doe) & analysis of experimental data*. http://cran.r-project.org/web/views/ExperimentalDesign.html. Retrieved 24.01.2012.
32. Grömping, U. (2012). *Project: (industrial) doe in r*. http://prof.beuth-hochschule.de/groemping/software/design-of-experiments/project-industrial-doe-in-r/. Retrieved 24.01.2012.
33. Gross, J. (2006). *nortest: Tests for normality*. R package version 1.0.
34. Gygi, C., DeCarlo, N., & Williams, B. (2005). *Six sigma for dummies*. Hoboken: Wiley.
35. Harrell, F. E. (2011). *Hmisc: Harrell miscellaneous*. http://CRAN.R-project.org/package=Hmisc, r package version 3.9-0.
36. Harry, M., & Schroeder, R. (1999). *Six sigma: The breakthrough management strategy revolutionizing the world's top corporations*. New York: Doubleday Business.
37. Hastie, T. (2011). *gam: Generalized additive models*. http://CRAN.R-project.org/package=gam, r package version 1.06.2.
38. Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning: Data mining, inference, and prediction. Springer series in statistics*. New York: Springer. http://www-stat.stanford.edu/~tibs/ElemStatLearn/.
39. Hornik, K., Buchta, C., & Zeileis, A. (2009) Open-source machine learning: R meets weka. *Computational Statistics*, *24*(2), 225–232.
40. Hsu, H. (2010). *Schaum's outline of probability, random variables, and random processes. Schaum's Outline Series* (2nd ed.). New York: McGraw-Hill.
41. Ishikawa, K. (1990). *Introduction to quality control*. Japan: 3A Corporation
42. ISO (1994). *ISO 5725-1: Accuracy (trueness and precision) of measurement methods and results—Part 1: General principles and definitions*. Geneva: International Organization for Standardization.

43. ISO (2009). *ISO/TS 16949: Quality management systems—Particular requirements for the application of ISO 9001:2008 for automotive production and relevant service part organizations*. Geneva: International Organization for Standardization.
44. ISO (2011). *Iso 13053-1:2011–Quantitative methods in process improvement—six sigma —part 1: DMAIC methodology*. Geneva: International Organization for Standardization.
45. Jakir, B. (2011). *Introduction to statistical thinking (with r, without calculus)*. http://pluto.huji.ac.il/~msby/StatThink/index.html. Accessed 01.08.2011.
46. John, A., Meran, R., Roenpage, O., Lunau, S., Staudter, C., & Schmitz, A. (2008). *Six Sigma+Lean Toolset*. Berlin: Springer.
47. Juran, J., & Defeo, J. (2010). *Juran's quality handbook: The complete guide to performance excellence*. New York: McGraw-Hill.
48. Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab—An S4 package for kernel methods in R. *Journal of Statistical Software*, *11*(9), 1–20. http://www.jstatsoft.org/v11/i09/.
49. Keller, P., & Keller, P. (2011). *Six Sigma demystified. Demystified series*. New York: McGraw-Hill.
50. Kelley, D. (2009). *plan: Tools for project planning*. http://CRAN.R-project.org/package=plan, r package version 0.3-1.
51. Kerns, G. J. (2010). *Introduction to probability and statistics using r*. http://ipsur.r-forge.r-project.org/book/index.html. Accessed 01.08.2011.
52. Kiermeier, A. (2008). Visualizing and assessing acceptance sampling plans: The R package AcceptanceSampling. *Journal of Statistical Software*, *26*(6), 1–20. http://www.jstatsoft.org/v26/i06/.
53. Knight, E., Russell, M., Sawalka, D., & Yendell, S. (2007). *Taguchi quality loss function and specification tolerance design*. https://controls.engin.umich.edu/wiki/index.php/Taguchi_quality_loss_function_and_specification_tolerance_design in Michigan chemical process dynamics and controls open textbook. Accessed 20.09.2011.
54. Knoth, S. (2011). *spc: Statistical process control*. http://CRAN.R-project.org/package=spc, r package version 0.4.1.
55. Koch, R. (2005). *Living the 80/20 way: Work less, worry less, succeed more, enjoy more*. London: Nicholas Brealey Publishing.
56. Kraemer, N., & Braun, M. L. (2011). *plsdof: Degrees of freedom and statistical inference for partial least squares regression*. http://CRAN.R-project.org/package=plsdof, r package version 0.2-2.
57. Kuhn, M. (2011). *odfWeave: Sweave processing of open document format (ODF) files*. http://CRAN.R-project.org/package=odfWeave, r package version 0.7.17.
58. Lalanne, C. (2006). *R companion to montgomery's design and analysis of experiments*. http://www.aliquote.org/articles/tech/dae/. Retrieved 19.01.2012.
59. Larsson, A. (2008). *Dia software*. http://live.gnome.org/Dia.
60. Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle & B. Rönz (Eds.), *Compstat 2002—Proceedings In Computational Statistics* (pp. 575–580). Heidelberg: Physica Verlag. http://www.stat.uni-muenchen.de/~leisch/Sweave, ISBN 3-7908-1517-9.
61. Lemon, J. (2006). Plotrix: A package in the red light district of r. *R-News*, *6*(4), 8–12.
62. Limas, M. C., Meré, J. B. O., Marcos, A. G., de Pisón Ascacibar, F. J. M., Espinoza, A. V. P., & Elías, F. A. (2010). *AMORE: A MORE flexible neural network package*. http://CRAN.R-project.org/package=AMORE, r package version 0.2-12.
63. Lopez-Fidalgo, J. (2009). A critical overview on optimal experimental designs. *Boletin de Estadística e Investigación Operativa*, *25*(1), 14–21. http://www.seio.es/BEIO/files/BEIOv25n1_ES_J.Lopez-Fidalgo.pdf. Retrieved 19.01.2012.
64. Maindonald, J. H. (2008). *Using r for data analysis and graphics. Introduction, code and commentary*. http://cran.r-project.org/doc/contrib/usingR.pdf. Accessed 01.08.2011

65. Martin, J. (2007). *Operational excellence: Using lean six sigma to translate customer value through global supply chains. Series on resource management*. PA: Auerbach Publications.
66. Mee, R. (2009). *A comprehensive guide to factorial two-level experimentation*. New York: Springer.
67. Mevik, B. H., Wehrens, R., & Liland, K. H. (2011). *pls: Partial least squares and principal component regression*. http://CRAN.R-project.org/package=pls, r package version 2.3-0.
68. Moguerza, J. M., Muñoz, A. (2006). Support vector machines with applications. *Statistical Science*, *21*(3), 322–336.
69. Montgomery, D. (2005). *Introduction to statistical quality control* (6th ed.). New York: Wiley.
70. Montgomery, D. (2008). *Design and analysis of experiments. Student solutions manual*. New York: Wiley.
71. Muir, A. (2005). *Lean Six sigma statistics: Calculating process efficiencies in transactional projects. Six sigma operational methods series*. New York: McGraw-Hill.
72. Murrell, P. (2009). Drawing Diagrams with R. *The R Journal*, *1*(1), 15–21. http://journal.r-project.org/archive/2009-1/RJournal_2009-1_Murrell.pdf.
73. Murrell, P. (2009). *Introduction to data technologies. Series in computer science and data analysis*. Boca Raton: CRC Press. http://www.stat.auckland.ac.nz/~paul/ItDT/.
74. Murrell, P. (2011). *R graphics* (2nd ed.). Boca Raton: Chapman & Hall/CRC.
75. Myers, R., Montgomery, D., & Anderson-Cook, C. (2009). *Response surface methodology: Process and product optimization using designed experiments. Wiley series in probability and statistics*. New York: Wiley.
76. Pande, P., Neuman, R., & Cavanagh, R. (2000). *The Six Sigma way: How GE, Motorola, and other top companies are honing their performance*. New York: McGraw-Hill.
77. Pearn, W., & Kotz, S. (2006). *Encyclopedia and handbook of process capability indices: A comprehensive exposition of quality control measures. Series on quality, reliability & engineering statistics*. New York: World Scientific.
78. Pebesma, E., & Bivand, R. (2005). Classes and methods for spatial data in r. *R News*, *5*(2), 9–13.
79. Poincaré, H. (1914). Science and method. London: Nelson. Originally published in French as Science et Méthode in 1908.
80. Popper, K. (1959). *The logic of scientific discovery*. London: Hutchinson. Originally published in German as Logik der Forschung in 1934.
81. Prieto, M. (2005). 6 sigma Qué es y cómo aplicarlo a la empresa española. Asociacion Española para la Calidad.
82. Pyzdek, T., & Keller, P. (2009). *The Six Sigma handbook: A complete guide for green belts, black belts, and managers at all levels*. New York: McGraw-Hill.
83. R-core (2011). *foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...* http://CRAN.R-project.org/package=foreign, r package version 0.8-48.
84. R Development Core Team (2011). *R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing*, Vienna, Austria. http://www.R-project.org/, ISBN 3-900051-07-0.
85. Rasch, D., Pilz, J., & Simecek, P. (2010). *Optimal experimental design with R*. London: Taylor & Francis.
86. Recchia, D. R., Barbosa, E. P., & de Jesus Goncalves, E. (2010). *IQCC: Improved quality control charts*. http://CRAN.R-project.org/package=IQCC, r package version 0.5.
87. Ripley, B. (2012). *RODBC: ODBC Database Access*. R package version 1.3-4.
88. Roth, T. (2011). *qualityTools: Statistics in quality science*. R package version 1.50.
89. Roy, R. (2010). *A primer on the Taguchi Method* (2nd ed.). Michigan: Society of Manufacturing Engineers.
90. Sanchez, G., & Trinchera, L. (2012). *plspm: Partial least squares data analysis methods*. http://CRAN.R-project.org/package=plspm, r package version 0.2-2.
91. Sarkar, D. (2008). *Lattice: Multivariate data visualization with R*. New York: Springer. http://lmdvr.r-forge.r-project.org, ISBN 978-0-387-75968-5.

92. Schonberger, R. (2008). *Best practices in lean six sigma process improvement: A deeper look*. Safari Books Online. Chichester: Wiley.
93. Scrucca, L. (2004). qcc: An r package for quality control charting and statistical process control. *R News*, *4*(1), 11–17, http://CRAN.R-project.org/doc/Rnews/.
94. Searle, S., Casella, G., & McCulloch, C. (2006). Variance components. Wiley series in probability and statistics. New York: Wiley.
95. Seifert, B. (2011). lpridge: Local polynomial (ridge) regression. http://CRAN.R-project.org/package=lpridge, r package version 1.0-6; Packaged for R by Martin Maechler.
96. Shaffer, L., Young, T., Guess, F., Bensmail, H., & León, R. (2008). Using r software for reliability data analysis. *International Journal of Reliability and Application*, *91*(1), 53–70.
97. Snee, R. D. (2004). Six-sigma: The evolution of 100 years of business improvement methodology. *International Journal of Six Sigma and Competitive Advantage*, *1*(1), 4–20.
98. Spano', A. (2011). *qAnalyst: Control charts, capability and distribution identification*. http://CRAN.R-project.org/package=qAnalyst, r package version 0.6.4.
99. Taguchi, G., Chowdhury, S., & Wu, Y. (2005). *Taguchi's quality engineering handbook*. USA: Wiley.
100. Therneau, T. (2011). *survival: Survival analysis, including penalised likelihood*. http://CRAN.R-project.org/package=survival, r package version 2.36-10. Original Splus->R port by Thomas Lumley.
101. Vardeman, S. B., & VanValkenburg, E. S. (1999). Two-way random-effects analyses and gauge r&r studies. *Technometrics*, *41*(3), 202–211.
102. Venables, W., & Ripley, B. (2002). *Modern applied statistics with S. Statistics and computing*. New York: Springer.
103. Vikneswaran (2005). *An r companion to "experimental design"*. http://cran.r-project.org/doc/contrib/Vikneswaran-ED_companion.pdf. Retrieved 19.01.2012.
104. Wahlbrink, S. (2012). *Statet: Eclipse based ide for r*. http://www.walware.de/goto/statet. Retrieved 16.01.2012.
105. Wickham, H. (2009). *ggplot2: Elegant graphics for data analysis. Use R!*. New York: Springer.
106. Wilkinson, L., & Wills, G. (2005). *The grammar of graphics. Statistics and computing*. New York: Springer.
107. Wilson, G. (2005). Six sigma and the product development cycle. Butterworth-Heinemann: Elsevier.
108. Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.
109. Wood, S. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society*, *73*(1), 3–36.
110. Workgroup, M. (2010). *MSA manual* (4th ed.). Automotive Industry Action Group (AIAG).
111. Xie, Y. (2012). *knitr: A general-purpose package for dynamic report generation in R*. http://CRAN.R-project.org/package=knitr, r package version 0.1.
112. Zuur, A., Ieno, E., & Meesters, E. (2009). *A beginner's guide to R. Use R!*. New York: Springer.

Solutions

Practice for Chap. 2

2.1

```
> install.packages("SixSigma", dependencies = TRUE)
```

Select a mirror close to your location when prompted.

2.2

```
> FailureTime <- c(0.29, 0.32, 1.21, 0.95, 0.14, 2, 0.81,
    0.88)
> Temp <- c(63.89, 63.38, 65.05, 62.31, 68.04, 59.12, 62.80,
    61.89)
> Factory <- c("A", "B", "C", "C", "B", "B", "A", "B")
> sol1a <- data.frame(FailureTime,
        Temp, Factory = as.factor(Factory))
> summary(sol1a)
> write.csv(sol1a, file = "sol1a.csv")
```

2.3

```
> plot(sol1a$Temp, sol1a$FailureTime)
> boxplot(sol1a$FailureTime ~ sol1a$Factory)
> hist(sol1a$Temp)
```

2.4

```
> table(sol1a$Factory)
> sol3 <- sol1a$Temp[sol1a$FailureTime < 1]
```

E.L. Cano et al., *Six Sigma with R: Statistical Engineering for Process Improvement*, Use R! 36, DOI 10.1007/978-1-4614-3652-2,

Practice for Chap. 3

3.1

```
> grid.roundrect(width = .25,
    height = unit(1.8, "inches"),
    x = 0.25)
> grid.text("INPUTS\n\nData\nComputer\nOperator",
    x = 0.25,
    y = 0.66,
    just = "top")
> grid.roundrect(width = .25,
    height = unit(1.8, "inches"),
    x = 0.75)
> grid.text("OUTPUTS (Y)\n\nInvoice\n(pages,\nsize,\ncolor)",
    x = 0.75,
    y = 0.66,
    just = "top")
> grid.lines(x = c(0.375, 0.625),
    y = c(0.5, 0.5),
    arrow = arrow())
```

3.2

```
>   inputs <-c ("Clothes", "Machine", "Powder")
>   outputs <- c("dryness", "cleanness", "time", "creases")
>   steps <- c("PREPARE", "WASH", "HANG OUT")
>   io <- list()
>   io[[1]] <- list("Clothes", "Machine", "Powder")
>   io[[2]] <- list("Clothes", "MachineState")
>   io[[3]] <- list("dryness", "cleanness", "time")
>   param <- list()
>   param[[1]] <- list(c("type", "P"), c("amount", "N"))
>   param[[2]] <- list(c("time", "Cr"), c("Powder.Brand", "Cr
    "),
                    c("weight", "P"), c("MachineAge", "Cr"))
>   param[[3]] <- list(c("WindSpeed", "N"), c("temperature",
    "N"),
            c("location", "C"))
>   feat <- list()
>   feat[[1]] <- list("cleanness")
>   feat[[2]] <- list("cleanness", "time", "dryness")
>   feat[[3]] <- list("cleanness", "time", "creases")
>   ss.pMap(steps, inputs, outputs,
        io, param, feat,
        sub = "Laundry Process")
```

Practice for Chap. 4

4.1
$L(Y) = 175(Y - 15)^2$

4.2
Formula:

$$L(Y) = 0.125 \cdot (Y - 750)^2$$

Graphic of function:

```
> curve(0.125*(x-750)^2,
    from = 735, to = 765)
```

4.3 Loss function analysis:

```
> ss.lfa(ss.data.ca, "Volume", 10, 750, 1.25, 850)
```

The average cost is $0.53 and the total loss is $452.9.

Practice for Chap. 5

5.1

The variable `lab` (laboratory) is the appraiser factor (three levels), and the variable `batch` is the part factor, also with three levels. Two measures of each batch were taken in each laboratory.

5.2
The command to run the MSA is as follows:

```
> ss.rr(var = comp, part = batch, appr = lab,
        data = ss.data.pastries)
```

This is a good measurement system. The number of distinct categories is greater than 4 (6) and the %Study Var for R&R is lower than 30%. However, this value (20.27%) is not lower than 10%, and the measurement system may be improved. The plots confirm these results.

Practice for Chap. 6

6.1
First we create the data from the example:

```
> b.effect <- "Delay"
> b.groups <- c("Personnel", "Weather", "Suppliers", "
    Planning")
```

```
> b.causes <- vector(mode="list", length=length(b.groups))
> b.causes[1] <- list(c("Training", "Inadequate"))
> b.causes[2] <- list(c("Rain", "Temperature", "Wind"))
> b.causes[3] <- list(c("Materials", "Delays", "Rework"))
> b.causes[4] <- list(c("Customer", "Permissions", "Errors"))
>
```

We can plot a cause-and-effect diagram:

```
> ss.ceDiag(b.effect, b.groups, b.causes, sub="Construction
    Example")
> b.data <- data.frame(cause = factor(unlist(b.causes)),
    count = c(5, 1, 3, 1, 2, 18, 20, 4, 15, 2, 4),
    cost = c(50, 150, 50, 10, 20, 180, 200, 10, 5, 20, 150))
```

Next we plot the Pareto chart:

```
> require(qcc)
> b.vector <- b.data$cost
> names(b.vector) <- b.data$cause
> pareto.chart(b.vector, cumperc = c(80))
```

6.2 We need the two charts in order to choose the adequate causes to focus on.

If we only use the Pareto chart for the count of errors, then we will select "Customer" as one of the "vital few." However, in the cost chart, it is apparent that delays due to customer needs do not generate a high cost. It is due to the fact that a new specification usually involves a revision of the contract and, therefore, an extended deadline without extra costs.

Using the cost chart, the factors to focus on are the delay in the receipt of materials, quality of materials, inadequateness of the personnel, and planning errors.

Practice for Chap. 7

7.1

With $\sigma = 2$:

$$Z = \min\left\{\frac{(14-10)}{2}, \frac{(10-4)}{2}\right\} = \min 2,3 = 2,$$

$$Z_{LT} = Z_{ST} - 1.5 = 2 \times 1.5 = 0.5.$$

With $\sigma = 1$:

$$Z = \min\left\{\frac{(14-10)}{1}, \frac{(10-4)}{1}\right\} = \min 4,6 = 4,$$

$$Z_{LT} = Z_{ST} - 1.5 = 4 \times 1.5 = 2.5,$$

7.2 Now we have less than 308,000 DPMO (previously we had more than 690,000 DPMO).

Practice for Chap. 8

8.1

```
> newData <- aggregate(pc.volume ~ pc.filler,
    sum, data = ss.data.pc)
> barplot(newData$pc.volume,
    names.arg = newData$pc.filler,
    main = "Total volume by filler")
```

8.2

```
> newData <- ss.data.pc.big[ss.data.pc.big$batch == 1,5]
> plot(newData, type = "b")
```

There seems to be a descendant trend over time.

Practice for Chap. 9

9.1 To save the subset:

```
> data.stats.prob1 <- ss.data.strings[ss.data.strings$type ==
    "E1",]
```

```
> summary(data.stats.prob1$res)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    5.75    6.50    6.20    8.00    9.00
```

```
> sd(data.stats.prob1$res)
```

```
[1] 2.067289
```

```
> IQR(data.stats.prob1$res)
```

```
[1] 2.25
```

The mean and median are not equal, so the distribution might not be symmetric. We can generate a boxplot to see the data from the summary.

```
> boxplot(data.stats.prob1$res)
```

9.2

```
> pbinom(4, 100, 0.01)
```

```
[1] 0.9965677
```

Practice for Chap. 10

10.1

Using the `shapiro.test` function:

```
> shapiro.test(ss.data.strings$res)
```

Using the `ss.ci` function from the `SixSigma` package:

```
> ss.ci(res, data = ss.data.strings)
```

These data are nonnormal. We need other methods for inference.

10.2

```
> prob2.model <- lm(res ~ len, data = ss.data.strings)
> summary(prob2.model)
```

The linear model is not good for these data. The p-value for the goodness-of-fit hypothesis test is very high (0.7149). Furthermore, the R-squared statistic is very low.

10.3 Solution using `lm` function:

```
> prob3.model1 <- lm(len ~ type, data = ss.data.strings)
> summary(prob3.model1)
```

Solution using `aov` function:

```
> prob3.model2 <- aov(len ~ type, data = ss.data.strings)
> summary(prob3.model2)
```

The length of the strings is independent of the type of string (p-value > 0.05). Plotting the effects:

```
> qplot(type, len, data = ss.data.strings) +
    stat_summary(fun.y = mean, geom = "line",
        aes(group = 1), col = "orangered") +
    stat_summary(fun.y = mean, geom = "point",
        shape = 17, size = 3, col = "red") +
    opts(title = "Effects of factor Type of string on Length"
        )
```

Practice for Chap. 11

11.1

```
> myDesign <- expand.grid(factor1 = gl(2, 1, labels = c("-",
    "+")),
        factor2 = gl(2, 1, labels = c("-", "+")),
        factor3 = gl(2, 1, labels = c("-", "+")),
        factor4 = gl(2, 1, labels = c("-", "+")),
        response = NA)
```

```
> myDesign$ord <- sample(1:16, 16)
> myDesign[order(myDesign$ord),]
```

11.2

The formula in the following code allows one to check up to three-way interactions.

```
> model.prob1 <- lm(score ~ (.-repl)^3  , data = ss.data.doe2
    )
> summary(model.prob1)
```

There are some important interactions; therefore, we keep them in the model jointly with the significant main effects.

```
> selectionvar <- step(model.prob1, method="backwards")
> summary(selectionvar)
> coef(selectionvar)
```

Practice for Chap. 12

12.1

```
> pb.groups.s <- with(ss.data.pb2,
    qcc.groups(pb.humidity, pb.group))
> pb.s <- qcc(pb.groups.s, type = "S")
> summary(pb.s)
> plot(pb.s)
```

12.2

As the number of errors is registered by surface unit, the sample size for each group is 1, and we will have straight control limit lines. The *u* chart is plotted with the following command:

```
> with(ss.data.pb4,
       plot(qcc(defects, sizes = rep(1, 80), type = "u")))
```

R Packages and Functions Used in the Book

E.L. Cano et al., *Six Sigma with R: Statistical Engineering for Process Improvement*, Use R! 36, DOI 10.1007/978-1-4614-3652-2,

Subject Index

E.L. Cano et al., *Six Sigma with R: Statistical Engineering for Process Improvement*, Use R! 36, DOI 10.1007/978-1-4614-3652-2,