Appendix: The Analysis of Questionnaire Data using R: Memory Card

A.1 Data Manipulations

A.1.1 Importation/Exportation of Datasets

```
read.csv("c:/.../toto.csv")          # import file in text format
read.csv2("c:/.../toto.csv")         # csv: coma separated value
read.delim("c:/.../toto.tab")        # delim: tab separated value
read.delim2("c:/.../toto.tab")       # csv2 or delim2 for specific
                                       countries
odbc.toto <- odbcConnectExcel("c:/.../toto.xls")        # library(RODBC)
sqlTables(odbc.toto)$TABLE_NAME                         # gives nameoftable
totoxls <- sqlFetch(odbc.toto, sqtable = "nameoftable") # import Excel file
read.spss("c:/.../toto.sav")         # read spss file
read.dta("c:/.../toto.dta")          # read stata file
write.csv(toto, "c:/.../toto.csv")   # write file in csv format
write.csv2(toto, "c:/.../toto.csv")  # same for countries where decimals
                                       are ","
save(toto1, toto2, "c:/.../toto")    # save a series of data frame
load("c:/.../toto")                  # read an object saved by save()
str(toto)                            # informations concerning an R oject
```

A.1.2 Manipulation of Datasets

```
toto.young <- toto[toto$age < 18, ]   # selection of observations
toto.somevar <- toto[, c("namevar1", "namevar2", "namevar3")]
                                      # selection of variables
toto2 <- subset(toto, age < 18, select = c("namevar1", "namevar2")]
                                      # other function for selection
toto <- merge(totovar1, totovar2, by = c("id1", "id2"))
                                      # merge datasets (add variables)
toto <- rbind(totoobs1, totoobs2)     # merge datasets (add observations)
toto.sort <- toto[order(toto$id), ]   # sort dataset according to "id"
sum(duplicated(toto))                 # number of duplicated obervations
toto[duplicated(toto)]                # which obervations are duplicated
identical(toto, toto2)                # are two dataframes the same
which(toto != toto2, arr.ind = TRUE)  # differences between two data frames
```

A.1.3 Manipulation of Variables

```
toto$var1.fact <- factor(toto$var1)   # transforms as a categorical variable
toto$var1 <- as.numeric(levels(toto$var1.fact))[toto$var1.fact]
                                      # backtransformation as a number
```

```
levels(toto$var1.fact) <- c("lev1", "lev2", ...)
                                    # change levels
toto$var1bin <- ifelse(toto$var1 < 20, 1, 0)
                                    # recode into a binary variable
toto$var1recode <- recode(toto$var1, "cond1; cond2; ...")
                                    # recode a variable; library(car)
is.na(toto$var1) <- toto$var1 == 9   # transform 9 a mining data
toto$var1cut <- cut(toto$var1, breaks = c(-Inf, lim1, ..., limq, Inf),
   labels = FALSE)                  # cut a numerical variable into pieces
   [contrasts() and relevel() are in the section statistical modelling]
```

A.2 Descriptive Statistics

A.2.1 Univariate

```
summary(toto)                       # mean, median, minimum, etc.
describe(toto) #library(prettyR)    # mean, median, sd, etc.
tab <- table(toto$var1, toto$var2, deparse.level = 2, useNA = "ifany")
                                    # crosstabulation of 2 variables
prop.table(tab, 1)                  # crosstabulation with %
by(toto$varcont1, toto$varcat1, mean, na.rm = TRUE)
                                    # subgroup analysis (mean)
hist(toto$var1)                     # histogram
plot(density(toto$varcont1, na.rm = TRUE))      # density curve
int <- hist(toto$varcont1, freq = FALSE, plot = FALSE)
hist(toto$varcont1, xlim = range(c(dest$x, int$breaks)),
   ylim = range(c(dest$y, int$density)), freq = FALSE)
   lines(dest, lty = 2, lwd = 2)
box()                               # histogram and density curve
qqnorm(toto$varcont1); qqline(toto$varcont1)
                                    # normal probability plot
barplot(table(toto$varcat1))        # barplot
boxplot(toto$varcont1 ~ toto$var1cat)# boxplots in subgroups
plot(toto$varcont1 ~ jitter(toto$var1cat))
                                    # distribution in subgroups
plotmeans(toto$varcont1 ~ toto$time)  # temperature diagram; library(gplots)
```

A.2.2 Bivariate

```
twoby2(toto$exposure,tot$outcome)    # odds-ratios and RR; library(Epi)
cor(toto[, c("var1", ..., "varp")], use = "complete.obs")
                                    # correlation matrix
rcor.test(toto[, c("var1", ..., "varp")], use = "pairwise.complete.obs")
                                    # correlation matrix; library(ltm)
plot(toto$var1, toto$var2)          # cartesian diagram
plot(jitter(toto$var1), jitter(toto$var2))
abline(lm(toto$var2 ~ toto$var1, data = mhp), lwd = 2, lty = 2)
nona <- !(is.na(toto$var1) | is.na(toto$var2))
lines(lowess(toto$var1[nona], toto$var2[nona]), lwd = 2)
                                    # cartesian diagram with regression
```

A.2.3 Multidimensional

```
cha <- hclust(dist(t(scale(toto[, c("var1", ..., "varp")])))), method = "ward")
plot(cha)                                # hierarchical clustering
obj <- cor(toto[, c("var1", ..., "varp")], use = "pairwise.complete.obs")
heatmap(obj)                             # shaded representation of correlations
mdspca(toto[, c("var1", ..., "varp")]) # PCA representation of a correlation
                                         # matrix; library(psy)
sphpca(toto[, c("var1", ..., "varp")]) # spherical representation of a
                                         # correlation matrix; library(psy)
fpca(outcome ~ exposure1 + ... + exposure, data = toto)
                                         # focused PCA; library(psy)
```

A.3 Statistical Inference

```
y <- na.omit(toto$varbin1)
binom.confint(x = sum(y), n = length(y), method = "Wilson")
                                         # CI of a proportion; library(binom)
chisq.test(toto$varcat1, toto$varcat2, correct = FALSE)
                                         # comparison of two proportions
fisher.test(toto$varcat1, toto$varcat2)
                                         # comparison of two proportions
t.test(toto$varcont1 ~ toto$varbin1, var.equal = TRUE)
                                         # comparison of two means
wilcox.test(toto$varcont1 ~ toto$varbin1)
                                         # comparison of two means
cor.test(toto$var1, toto$var2)         # test that a correlation is zero
events <- table(toto$varbin1, toto$varord1)[, 2]
trials <- events + table(toto$varbin1, toto$varord1)[, 1]
prop.trend.test(events,trials)         # chisquare test for trend
n.for.survey(p = 0.01, delta = 0.02)  # sample size for a survey
n.for.2p(p1 = 0.125, p2 = 0.05, power = 0.9, ratio = 3)
                                         # sample size for a survey
```

A.4 Statistical Modelling

```
mod <- lm(outcomecont ~ var1 + ... + varp, data = toto)
summary(mod)                             # linear regression, ANOVA
drop1(mod, .~., test = "F")            # "type III" hypothesis testing
plot(mod,2)                              # regression diagnosis
plot(mod,1)                              # regression diagnosis
plot(mod,4)                              # regression diagnosis
mod <- glm(outcome01 ~ var1 + ... + varp, data = toto, family = "binomial")
summary(mod)                             # logistic regression
exp(coefficients(mod))                 # adjusted odd-ratios
exp(confint(mod))                      # CI of adjusted OR
drop1(mod, .~., test = "Chisq")        # "type III" hypothesis testing
plot(mod,4)                              # regression diagnosis
```

```
logistic.display(mod, decimal = 3)    # display log reg; library(epicalc)
summary(multinom(outcomecat ~ var1 + ... + varp, data = toto))
                                      # multinomial log reg; library(nnet)
summary(polr(outcomeord ~ var1 + ... + varp, data = toto))
                                      # polytomic log reg; library(MASS)
summary(glm(outcomecount ~ var1 + ... + varp, data = toto,
   family = "quasipoisson")
                                      # Poisson regression (overdispersion)
summary(glm.nb(outcomecount ~ var1 + ... + varp, data = toto)
                                      # negative binomial reg; library(MASS)
plot(gam(outcomecont ~ s(varcont1), data = toto))
                                      # regression spline; library(mgcv)
plot(gam(outcome01 ~ s(varcont1), data = toto, family = binomial))
                                      # regression spline; library(mgcv)
summary(glm(outcome01 ~ poly(varcont1, degree = 3),data = toto,
   family = binomial))
                                      # orthogonal polynomials
contrasts(toto$varcat1)               # coding of a factor
contrasts(toto$varcat1) <- contr.sum
                                      # 1 -1 coding
contrasts(toto$varcat1) <- contr.treatment
                                      # 0 1 coding
toto$varcat1 <- relevel(toto$varcat1, ref = "newreflevel")
                                      # change level of reference of a factor
estimable(mod, c(0, ..., 0, 1, ..., -1, ..., 0))
                                      # estimation and test of a contrast
summary(step(mod, scope = list(upper = ~var1 + ... + varp,
   lower = ~varb1 + ... + varbq), trace = FALSE))
                                      # stewise regression
scale(toto)                           # normalize data set
calc.relimp(mod, type = c("lmg", "last"), rela = TRUE)
                                      # relative importance of predictors
plot(naclus(toto))                    # pattern of miss data; library(Hmisc)
micetoto <- mice(toto, seed = 1)      # impute 5 date sets; library(mice)
imputetoto <- complete(micetoto)      # completed dataset; library(mice)
summary(glm.mids(outcome01 ~ var1, ..., varp, data = micetoto,
   family = "binomial")               # multiple imputations, library(mice)
lm.boot <- function(data, index) {
mhp.boot <- data[index, ]
mod <- lm(outcomecont ~ var1 + ... + varp, data = mhp.boot)
coefficients(mod)
}
resboot <- boot(mhp.mod, lm.boot, 10000)
boot.ci(resboot, index = 2, type = "bca")
                                      # bootstrapped coefs; library(boot)
summary(glmer(outcome01 ~ var1 + ... + varp + (1 | varrandomeffect),
   data = toto, family = "binomial")  # random effect; library(lme4)
```

A.5 Validation of a Composite Score

```
toto$sumvar1p <- apply(toto[, c("var1", ..., "varp"], 1, sum, na.rm = TRUE)
                                      # summation of p items
```

```
mtmm(toto, list(c("var11", ..., "var1p"), ..., c("varq1", ..., "varqp"),
   itemTot = TRUE, namesDim = c("namescale1", ..., "namescaleq"))
                                    # multitraits multimethods;
                                    # library(psy)
scree.plot(toto[, c("var1", ..., "varp")], simu = 20)
                                    # scree plot; library(psy)
resfa <- factanal(na.omit(toto[, c("var1", ..., "varp")], factors = k)
print(resfa, cutoff = 0)            # factor analysis
print.psych(promax(loadings(resfa)), cut = 0)
                                    # oblique rotation; library(psych)
cronbach(toto[, c("var1", ..., "varp")])
                                    # Cronbach alpha; library(psy)
ckappa(toto[, c("rater1", "rater2")])  # Cohen kappa; library(psy)
lkappa(toto[, c("rater1", ..., "raterq")])
                                    # kappa for q raters; library(psy)
wkappa(toto[, c("rater1", "rater2")])  # weighted kappa; library(psy)
icc(toto[, c("rater1", "rater2")])   # intraclass correlation; library(psy)
```