WORD versions of the R code from R. K. Hanson (2022), *Prediction statistics for psychological assessment*. American Psychological Association.

The same code is provided in the book, with additional explanation; however, certain readers may find it easier to copy directly from this WORD file rather than copying from the book.

R syntax is presented in Courier New 10 pitch font. The output from R is presented in **Courier New 10 pitch bold.**

Note that the identification of the datasets is specific for each computer. The syntax in the examples below tells R where my files are stored on my desktop. You will need to change the locations to tell R where your files are stored when using commands such as **read.csv*,* write.table**, and **read.table**.

**Chapter 4 Proportions**

The following compute code calculates the proportion, its standard error, and 95% confidence intervals using both the Wald method and the Agresti and Caffo adjustment described in Chapter 4. The compute code is based on Prof. Breuer imaginary study of recidivism: events/total = p = 9/24 = 0.375

data <- c(1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

p <- mean (data)

p

**[1] 0.375**

n <- length (data)

n

**[1] 24**

The following calculates the 95% confidence intervals using the Wald method:

upperCI <- p + (1.96)\*(sqrt((p)\*(1-p)/n))

upperCI

**[1] 0.5686895**

lowerCI <- p - (1.96)\*(sqrt((p)\*(1-p)/n))

lowerCI

**[1] 0.1813105**

The following calculates the Agresti and Caffo confidence intervals:

data <- c(1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

events <- sum(data)

n <- length (data)

N = n + 4

P = (events + 2)/N

P

**[1] 0.3928571**

ACupperCI <- P + (1.96)\*(sqrt((P)\*(1-P)/N))

ACupperCI

**[1] 0.5737577**

AClowerCI <- P - (1.96)\*(sqrt((P)\*(1-P)/N))

AClowerCI

**[1] 0.2119566**

**Chapter 5 Discrete Time Survival Analysis**

The example below uses the dataset from the 2019 Vail University victimization study, which followed students for one year in order to estimate 4-year rates (Table 5.2). Although the actual sample size was 4981, the cases are aggregated to only 2045 rows, corresponding to the number of students in the first cohort. The relevant package in R statistics is called **discSurv** (Welchowski & Schmid, 2019). The **lifeTable** routine computes life-table that give ½ weight to cases withdrawn during the interval. The dataset for this example is called *Chap5\_DTime\_example.csv*.

install.packages("discSurv")#if needed, first download the package

library ('discSurv')

labels <-c("Interval", "Event")

data <-read.csv(file="d:\\APA data\\Chap5\_DTime\_example.csv", col.names= labels)

names (data)

**[1] "Interval" "Event"**

model <- lifeTable(dataSet=data, timeColumn="Interval", censColumn="Event")

model

**Chapter 6 Kaplan-Meier Survival Analysis**

This example is a scaled-up version of the hypothetical rehospitalization study discussed in Chapter 6, with 150 cases. The dataset is called *Chap6\_KM\_example.csv*. The first few steps transform the raw variables *atrisk.date,* *end.followup*, and *event.date*, into a single survival time variable.

install.packages (‘survival’)#if needed, download the package

require ('survival')

KMdata <-read.csv("d:\\APA data\\Chap6\_KM\_example.csv")

attach (KMdata)

Atrisk.date<-strptime(atrisk.date, format = "%m/%d/%Y")

End.followup<-strptime(end.followup, format = "%m/%d/%Y")

Event.date<-strptime(event.date, format = "%m/%d/%Y")

For the purpose of date arithmetic, the dates are then transformed from character variables (e.g., “01/01/2010”) into seconds (from 1-Jan-1970, e.g., 1262304000).

Atrisk.datep <- as.POSIXlt(Atrisk.date)

End.followupp <- as.POSIXlt(End.followup)

Event.datep<-as.POSIXlt(Event.date)

followup.days <-difftime (End.followupp, Atrisk.datep)

mean (followup.days)

**Time difference of 253.6397 days**

Now calculate the variable *time.to.event*, which indicates the time between release and readmission. The survival time variable is *survival.days*.

time.to.event <- Event.datep - Atrisk.datep

time.to.event999 <-ifelse(is.na(time.to.event),999, time.to.event)

survival.days <- pmin (followup.days, time.to.event999)

KMdata <- data.frame (KMdata, followup.days, survival.days)

write.table(KMdata, "c:\\run.files\\Book\\KMoutdata.table")

KMmodel <- survfit(Surv(survival.days, event) ~ 1, type = "kaplan-meier")

summary (KMmodel)

To obtain confidence intervals based on ±1.96(*SE*(*St*)), specify “plain” as the option:

KMmodel2 <- survfit(Surv(survival.days, event) ~ 1, conf.type = "plain")

To plot the survival curve.

plot (KMmodel, xlab ="Time in Days", ylab = "Proportion Surviving",

cex.lab = 1.3)

**Chapter 7 Dichotomous Predictors**

**Fourfold Table.** This example uses the mock example of Dr. Piaget’s examination of the relationship between living with both parents to age 16 and high school graduation.

A <- 60 #the number with the predictor and the outcome (hits)

B <- 5 #the number with the predictor but not the outcome (false alarms)

C <- 20 #the number with the outcome but not the predictor (misses, false negatives)

D <- 15 #the number with neither the predictor nor the outcome (true negatives)

N <- sum(A, B, C, D)

baserate.outcome <- (A + C)/N

baserate.predictor <- (A + B)/N

PosPredAccuracy <- A/(A + B)

NegPredAccuracy <- D/(C + D)

sensitivity <- A/(A + C)

specificity <- D/(B + D)

baserate.outcome

**[1] 0.8**

baserate.predictor

**[1] 0.65**

PosPredAccuracy

**[1] 0.9230769**

NegPredAccuracy

**[1] 0.4285714**

sensitivity

**[1] 0.75**

specificity

**[1] 0.75**

**Rate ratios.** This example uses the same values as the previous worked mock example of Dr. Piaget’s examination of the relationship between living with both parents to age 16 and high school graduation in Geneva. However, the variables are reversed so that the predictor is not living with both parents and the outcome is failing to complete high school.

A <- 15 #the number with the predictor and the outcome (hits)

B <- 20 #the number with the predictor but not the outcome (false alarms)

C <- 5 #the number with the outcome but not the predictor (misses, false negatives)

D <- 60 #the number with neither the predictor nor the outcome (true negatives)

rate.separate <- A/(A + B)

rate.together <- C/(C + D)

rate.ratio <- rate.separate/rate.together

ln.rate.ratio <- log(rate.ratio) # the *log* command calculates the natural logarithm (base *e*)

SE.rate.ratio <- sqrt ((1/A)-(1/(A + B))+(1/C)-(1/(C + D)))

lowerCI <- exp(ln.rate.ratio - ((1.96)\*SE.rate.ratio))

upperCI <- exp(ln.rate.ratio + ((1.96)\*SE.rate.ratio))

rate.ratio

**[1] 5.571429**

lowerCI

**[1] 2.209318**

upperCI

**[1] 14.04995**

**Odds ratios.** This example uses the same values as the previous worked mock example of Dr. Piaget’s examination of the relationship between living with both parents to age 16 and high school graduation in Geneva. The data is organized such that the predictor is *not living with both parents* and the outcome is *failing to complete high school*.

A <- 15 #the number with the predictor and the outcome (hits)

B <- 20 #the number with the predictor but not the outcome (false alarms)

C <- 5 #the number with the outcome but not the predictor (misses, false negatives)

D <- 60 #the number with neither the predictor nor the outcome (true negatives)

odds.ratio <- ((A +.5)\*(D + .5))/((B + .5)\*(C + .5))

ln.odds.ratio <- log(odds.ratio) #The *log* command calculates the natural logarithm (base *e*)

SE.odds.ratio <- sqrt ( 1/(A+.5) + 1/(B+.5) + 1/(C+.5) + 1/(D+.5) )

lowerCI <- exp (ln.odds.ratio - ((1.96)\*SE.odds.ratio))

upperCI <- exp (ln.odds.ratio + ((1.96)\*SE.odds.ratio))

odds.ratio

**[1] 8.317073**

> lowerCI

**[1] 2.784697**

> upperCI

**[1] 24.84065**

**Chapter 8 AUC**

The dataset used in this example is called *Chap8\_AUC\_example.csv*; it is same data presented in Table 8.2 (SANE, Study 1), reorganized with one row per case.

install.packages ("pROC") #if needed, download the package

library(pROC) #tell R that you want to use that package

AUCdata<-read.csv("d:\\APA data\\Chap8\_AUC\_example.csv")

attach (AUCdata)

roc(Relapse, Level) #the basic analysis

exampleROC <-roc(Relapse, Level) #creating an ROC object

ci.auc(exampleROC)

**95% CI: 0.634-0.8244 (DeLong)**

plot (exampleROC)

**Chapter 9 Cohen’s d**

This example uses the Self-Control Questionnaire as a predictor of relapse (Table 9.2).

install.packages ('effsize') #if needed, download the package

library ('effsize') #tell R you want to use it

The vector called *relapse* has values for the 58 cases, whereas the vector called *no.relapse* has values for 40 cases. The raw data was entered in EXCEL.csv files called *Chap4c\_relapse.csv* and *Chap4c\_no\_relapse.csv*.

V1 <- read.csv ("d:\\APA data\\Chap9\_relapse.csv")

relapse <- (V1$x)

V2 <- read.csv ("d:\\APA data\\Chap9\_no\_relapse.csv")

no.relapse <- (V2$x)

mean (relapse)

sd (relapse)

length (relapse)

mean (no.relapse)

sd (no.relapse)

length (no.relapse)

cohen.d (no.relapse, relapse)

Another option is to use a dataset with two variables “self.control” and “outcome”.

Chap9.data <- read.table (file = "d:\\APA data\\Chap9\_example.table")

attach(Chap9.data)

outcome <- factor (outcome)

cohen.d (self.control ~ outcome)

For calculating Hedges’ g:

cohen.d (self.control ~ outcome, hedges.correction = TRUE, noncentral = TRUE)

Chapter 10 Cox Regression

The following example concerns the Static-99R risk tool as a predictor of sexual recidivism. The dataset (*Chap10\_Cox\_example.csv*) contains 767 cases of which 83 were known to have reoffended sexually during the follow-up period.

install.packages (‘survival’) #if you do not already have it

require ('survival')

Cox.data <-read.csv(file = "d:\\APA data\\Chap10\_Cox\_example.csv")

attach (Cox.data)

names (Cox.data)#checking the variable names

Cox.data [1, ] #another way of checking the variable names

model1<-coxph(Surv(Time.Sex, Recid.Sex) ~ Stat99Rtot)

summary (model1)

**Chapter 11 logistic regression**

The computer code examples use data from the worked example involving 555 cases of Static-2002 as predictor of fixed 10-year sexual recidivism rates (*Chap4e\_LR\_example.csv*).

LR.data <-read.csv("d:\\APA data\\Chap11\_LR\_example.csv")

attach (LR.data)

names (LR.data)

LR.model<-glm(sexRCD10yr ~ Static2002, family=binomial)

summary (LR.model)

Calculating observed and fitted values for plotting.

p.recidivism <- fitted.values (LR.model)

expected <- tapply (p.recidivism, Static2002, mean)

observed <- tapply (sexRCD10yr, Static2002, mean)

scores <-c(0:13)

plot (scores, expected, ylab="10-year sexual recidivism", xlab="Static-2002 total scores", cex.lab = 1.3)

lines (scores, expected, lwd = 2)

lines (scores, observed, lty = 3, lwd = 2)

legend(.12, .55, legend=c("Logistic", "Fixed Follow-up"),lty=c(1,3), cex=1.1)

The Hosmer-Lemeshow test.

install.packages ("generalhoslem") #if you have not used it before

library ("generalhoslem")

HL.test <- logitgof(sexRCD10yr,p.recidivism, g = 10)

HL.test

If you want to see the table of values, type the following:

cbind (HL.test$observed, HL.test$expected)

Syntax for generating 95% confidence intervals for the predicted values, and reports them as proportions.

inverse\_logit = function(x){exp(x)/(1+exp(x))}

LR.values <- predict(LR.model, type='link', se.fit=TRUE)

se\_high <- inverse\_logit(LR.values$fit + (LR.values$se.fit\*1.96))

se\_low <- inverse\_logit(LR.values$fit - (LR.values$se.fit\*1.96))

upper <- tapply (se\_high, Static2002, mean)

lower <- tapply (se\_low, Static2002, mean)

table.ci <- cbind (0:13, lower, expected, upper)

table.ci

**Chapter 12 Chi-square Goodness-of-Fit**

The example uses the data from the evaluation study of Dr. Ebing-Krafft’s risk scheme presented in Table 12.2.

O <- c(7,43,12,17,12,28,11,79) #the observed frequencies

E <- c(10,40,17.4,11.6,8,32,9,81) #the expected frequencies

EP <- E/sum(E) #the expected frequencies as proportions

chisq.test (O, p = EP)

**Chi-squared test for given probabilities**

**data: O**

**X-squared = 8.3085, df = 7, p-value = 0.3062**

#chi-square value is correct, disregard df and p-value

#Use 1 – **pchisq** (chi-square value, correct df) for the upper tail p-value

1 - pchisq (8.3085, 3)

**[1] 0.04004818**

**Chapter 13 E/O Index**

The following R syntax computes the E/O index with 95% confidence interval for studies with fixed follow-up. In this example, the expected number has already been calculated (23) and the observed is the raw number of observed events in the study (20).

E <- 23

O <- 20

EO <- (E/O)

EO

**[1] 1.15**

upper95 <- EO\*exp((1.96)\*(sqrt(1/O)))

upper95

**[1] 1.782526**

lower95 <- EO\*exp((-1.96)\*(sqrt(1/O)))

lower95

**[1] 0.7419248**

The following syntax calculate the observed values from Kaplan-Meier survival analysis. The dataset for this example is called Chap13\_EO\_KM\_example.csv.

install.packages (‘survival’)#if needed, download the package

require ('survival')

data<-read.csv(file="d:\\APA data\\Chap13\_EO\_KM\_example.csv")

attach (data)

KMmodel <- survfit(Surv(time, outcome)~ 1, type="kaplan-meier")

summary (KMmodel)

With ragged follow-up, you should use Formula 13.3 and Formula 13.4 to estimate the 95% confidence intervals. The following example assumes the expected number has been calculated (53.4) and the observed number is total sample (767) times the proportion estimated to have who reoffended (0.083) = 63.661.

E <- 53.4

O <- (767)\*(.083)

O

**[1] 63.661**

EO <- E/O

EO

**[1] 0.8388181**

SE <- 767\*(.01047)

SE

**[1] 8.03049**

upper95 <- EO\*exp((1.96)\*(SE/O))

upper95

**[1] 1.074099**

lower95 <- EO\*exp((-1.96)\*(SE/O))

lower95

**[1] 0.6550755**

The following syntax creates the survival table for Question 5.b.2. The dataset is called *Chap13\_EO\_KM\_Question2.csv*.

require ('survival')

Q2.data<-read.csv(file="d:\\APA data\\Chap13\_EO\_KM\_Question2.csv")

attach (Q2.data)

Q2.model <- survfit(Surv (Months, Remission) ~ 1)

summary (Q2.model)

**Chapter 14 Meta-analysis**

This example re-analyzes the data from the first worked example in Chapter 5.c. The data can be in a pre-existing datafile, or the datafile can be created using the following commands:

install.packages ('metafor')#if you have not used it before

library ('metafor')

setting <- c("Penetongue", "Fridgewater", "Fort Worth-It", "Mouse Lake",

"Candy Ridge", "Caligula")

ni <- c(80, 35,72, 110, 28, 15)

xi <- c(32, 10, 30, 50, 11, 5)

The following line transforms the raw proportion (xi/ni) into , and the next line creates the variances (vi) from the sample sizes (ni).

Ai <- 2\*asin(sqrt(xi/ni))

vi <-(1/ni)

The following lines create a datafile that can be read by **metafor,** and then the analysis is conducted using the **rma** command.

E1.data <- data.frame (setting, ni, Ai, vi)

E1.data <- escalc (measure = "GEN", yi=Ai, vi=vi, data=E1.data)

Example1 <- rma(yi,vi, data=E1.data, method = "FE")

Example1

The following is syntax converts variance stabilized/arcsine transformed p values ( ) back into proportions.

p <- (sin(Ai/2))^2

Example:

sin(1.3801/2))^2

**[1] 0.4052287**

The lower and upper limits of the 95% confidence interval (1.2728 to 1.4864) can be similarly converted to proportions (35.3% to 45.8%):

(sin(1.2738/2))^2

**[1] 0.3536753**

(sin(1.4864/2))^2

**[1] 0.4578519**

The simplest syntax for creating a forest plot in R is *forest(meta object)*, e.g., *forest (Example1).* A somewhat prettier display can be created by tweaking the options, as in the following syntax:

forest (Example1, slab=paste(E1.data$setting), xlim = c(-1, 4))

text(-.5, 8, "Setting", pos=1)

text(4, 7.7, "Arcsine Transformed p [95% CI]", pos =2)

The following R code conducts a fixed-effect meta-analysis of the logistic regression coefficients from Example 2 in Chapter 14. Specifically, it examines the between-group variability of the intercept centered on a score of 6 (B0[6]).

couple\_type <-c("Opposite Sex", "Same Sex Men", "Same Sex Women")

B06 <- c(1.45862, 1.69860, 0.86795)

SEB06 <- c(0.10533, 0.16419, 0.12978)

VarB06 <- SEB06\*SEB06

E2.data <- data.frame (couple\_type, B06, VarB06)

E2.data <-escalc (measure = "GEN", yi = B06, vi = VarB06, data=E2.data)

Example2 <-rma(yi,vi, data=E2.data, method = "FE")

Example2

**Chapter 15 Calibration plots**

The following R code creates the four examples in Chapter 15.

**Example 1.** This does not require any additional datasets. It can be run directly from the R prompt.

a <-c(0, 0, 1, 1)

A <-matrix (a, nrow = 2, ncol=2, byrow=TRUE)

plot (A, xlab = "Predicted Probability", ylab = "Observed Probability", cex.lab = 1.3, xaxs = "i", yaxs = "i")

lines (A)

b <-c(0, .35, 1, .35)

B <-matrix (b, nrow = 2, ncol=2, byrow=TRUE)

lines (B, cex = 0.5, col = "gray85")

grid (10, 10)

text (.20, .14, "A")

text (.60, .414, "B")

text (.20, .30, "C")

text (.10, .122, "D")

**Example 2.** For Figure 15.2, the datafile is called *Chap5d\_Fig2.table.*

install.packages ("plotrix") #if you have used it before

library ("plotrix")

Fig2.data <- read.table (file = "d:\\APA data\\Chap15\_Fig2.table")

plotCI (Fig2.data$Expected, Fig2.data$Observed, Fig2.data$CI, xlim=c(0, .40), ylim = c(0, .40), xlab="Expected", ylab = "Observed with 95% CI", cex=1.8, cex.lab = 1.3)

b<-c(0, 0, .40, .40)

B <-matrix (b, nrow = 2, ncol=2, byrow=TRUE)

lines (B)

LWS <-loess (Fig2.data$Observed ~ Fig2.data$Expected)

lines (LWS, lty = 3)

**Example 3.** The datafile for the analysis is called Chap15\_Fig3.table.

library ("plotrix")

Fig3.data <- read.table (file = "d:\\APA data\\Chap15\_Fig3.table")

plotCI (Fig3.data$Static99R, Fig3.data$Observed, Fig3.data$SE.p, ylim=(0:1), xlab="Static-99R Scores", ylab = "Sexual Recidivism Rate", cex=1.3)

lines (Fig3.data$Static99R, Fig3.data$Expected)

label.names <- c("Observed with SE", "Expected")

label.symbols <- c(1, 95)

legend (-1, .50, label.names, pch = label.symbols, bty = "n")

**Example 4.** The datafile for the analysis is called *Chap15\_Fig4.table*. Recognize the cover?

Fig4.data <- read.table (file = "d:\\APA data\\Chap15\_Fig4.table")

ylim.values <- c(0, .50)

plot (-3:10, Fig4.data$p.fitted, xlab = "Static-99R", ylab = "Recidivism Rate", ylim = ylim.values, cex=1.3, cex.lab =1.2)

lines (-3:10, Fig4.data$p.fitted, lty=2)

lines (-3:10, Fig4.data$upper, lty = 3)

lines (-3:10, Fig4.data$lower, lty = 3)

lines (-3:10, Fig4.data$p.norms, lty = 1)

legend.names <- c("Observed ", "95% Confidence Intervals ", "Expected ")

legend.lines <- c(2, 3, 1)

legend (-2, .30, legend.names, lty = legend.lines, bty = "n")

**Chapter 16 Percentiles**

The R code below computes the four percentile columns of Table 16.1 (below, same, above, midpoint average) from the observed number with each risk score.

n <- c(45, 50, 97, 129, 169, 181, 138, 86, 77, 72, 36, 16, 3, 1) #the data

N <- sum(n)

same <- (n/N)\*100

same.pct <- round(same, 1) #rounding to one decimal point

same.pct

**[1] 4.1 4.5 8.8 11.7 15.4 16.5 12.5 7.8 7.0 6.5 3.3 1.5 0.3 0.1**

n.1 <-c(0,45,50,97,129,169,181, 138, 86, 77, 72, 36, 16, 3, 1)#zero added

n.below <- cumsum (n.1) #cumulative sum of n.1

p.below <- round(((n.below/N)\*100), 1) #calculating and rounding in one step

p.below

**[1] 0.0 4.1 8.6 17.5 29.2 44.5 61.0 73.5 81.4 88.4 94.9 98.2 99.6 99.9 100.0**

n.below2 <- cumsum (n)

n.above <- (N - n.below2)

p.above <- round (((n.above/N)\*100), 1)

p.above

**[1] 95.9 91.4 82.5 70.8 55.5 39.0 26.5 18.6 11.6 5.1 1.8 0.4 0.1 0.0**

n.2 <-c(45,50,97, 129, 169, 181, 138, 86, 77,72,36, 16, 3, 1, 0)#zero added

p.midpoint1 <- ((n.below + .5\*n.2)/N)\*100

p.midpoint <- round (p.midpoint1, 1)

p.midpoint

**[1] 2.0 6.4 13.0 23.3 36.9 52.8 67.3 77.5 84.9 91.6 96.5 98.9 99.8 100.0 100.0**