7 Anhang

7.1 Abkürzungen und Spezialbegriffe

CART Klassifikations- und Regressions-Baum (**C**lassification **a**nd **R**egression **T**ree) [19]

Cox-PH-Regression Proportionale Hazrad-Regression nach Cox [38]

Design S-Plus-Bibliothek mit speziellen Funktionen zur Entwicklung und Validierung von multivariaten Modellen (Frank Harrell)

Hmisc S-Plus-Bibliothek mit vielen nützlichen Funktionen (Frank Harrell)

KNN Abkürzung für **K**ünstliche **N**euronale **N**etze – [175]

ROC **R**eceiver **O**peration **C**haracteristic

Rpart Rekursive Partitionierung. Der Name für die S-Plus-Bibiliothek von T. Therneau wurde gewählt, weil der Name CART schon benutzt wurde. Bei Rpart handelt es sich aber um erweiterte Funktionen zur Generierung von CART

SurvNN Diese S-Plus-Bibliothek wurde im Rahmen der Dissertation von Ruth M. Ripley entwickelt und an mehreren Beispielen getestet [134, 135]. Sie basiert in wesentlichen Punkten auf der Bibliothek NNet von Brian Ripley.

Impute Der in der englischsprachigen Literatur häufig zu findende Begriff *Imputation* wurde in dieser Arbeit mit Zuschreibung übersetzt. Gemeint ist die Ersetzung fehlender Angaben durch Werte, die mit statistischen Verfahren geschätzt wurden. Genauere Angaben dazu finden sich im Methodenteil.

7.2 Abbildungsverzeichnis

7.3 Tabellenverzeichnis

7.4 Programm zur Datensimulation

```
* Generierung von Zufallszahlen.
compute status=0.
compute id=$casenum.
compute v1=rv.normal(1,0.5).
compute v2=rv.uniform(1,5).
compute v3=rv.uniform(1,10).
compute Zeit = rv.weibull(1,1)*12.
compute zz1 = rv.uniform(0,1).
compute zz2 = rv.uniform(0,1).
compute zz3 = rv.uniform(0,1).

* Erzeugung der Variablen Status und Nachbeobachtungszeit.
if (v1+v2 > 5) and (zz1 <= 0.3) Status=1.
if (v1+v2 > 5) and (zz1 <= 0.5) Zeit=Zeit*(2/3).
if (v1+v3 > 10) and (zz2 <= 0.4) Status=1.
if (v1+v3 > 10) and (zz2 <= 0.4) Zeit=Zeit*(2/3).
if (v2*v3 > 40) and (zz3 <= 0.5) Status=1.
if (v2*v3 > 40) and (zz3 <= 0.5) Zeit=Zeit*(2/3).

* Erzeugung von Variablen/Hinzufügen von Rauschen.
compute f1=v1+v4/2.
compute f2=trunc(v2).
compute f3=v3+v5.
compute f4=v6.
compute f5=trunc(v1).
execute.
```

7.5 Beispiele für S-Plus-Auswertungsprogramme

```
# Berechnung eines CART
library(RPart)
fit <- rpart(Surv(NACHZT,STATUS)
      ~TD+LEVEL+TANS+HISTONEU+GESCHL+ALTER+HULZ,
      data=training.om,control=rpart.control(cp=0.0001))
plotcp(fit)
fit1->prune(fit,cp=0.004)
plot(fit1)
text(fit1)
post(fit1,file="")

training.om$pred.rpart<-predict(fit1)
test.om$pred.rpart<-predict(fit1,newdata=test.om)
```

```
# COX-PH-Analyse
library(hmisc,T)
library(Design,T)
attach(training.om)
GESCHLN<-factor(GESCHL)
HISTON<-factor(HISTONEU)
HULZN<-factor(HULZ)
TANSN<-factor(TANS)
LEVELN<-ordered(LEVEL)
TDN<-log(TD)
dd<-datadist(GESCHLN,TDN,ALTER,HULZN,HISTON,TANSN,LEVELN)
options(datadist='dd')
S<-Surv(NACHZT,STATUS)
f<-cph(S~GESCHLN+rcs(TDN,4)+HULZN+HISTON+LEVELN+TANSN+ALTER,
       eps=0.000001,x=T,y=T,surv=T,time.inc=60)
anova(f)
fastbw(f)
z<-predict(f,type='terms')
fs<-cph(S~z,x=T,y=T)
phtest<-cox.zph(fs,transform='identity')
plot(phtest,var='TDN')

par(mfrow=c(2,2))
plot(f,GESCHLN=NA)
plot(f,TDN=NA)
plot(f,HULZN=NA)
plot(f,HISTON=NA)

par(mfrow=c(2,2))
plot(f,LEVELN=NA)
plot(f,TANSN=NA)
plot(f,ALTER=NA)

validate(f,B=200,dxy=T)
cali1<-calibrate(f,B=500,u=60,m=1000)

training.om$pred.cox<-predict(f,)
detach()
```

```
# KNN
library(hmisc,T)
library(Design,T)
library(survnnet,T)
attach(training.om)
GESCHLN<-factor(GESCHL)
HISTON<-factor(HISTONEU)
HULZN<-factor(HULZ)
TANSN<-factor(TANS)
LEVELN<-ordered(LEVEL)
TDN<-log(TD)
dd<-datadist(GESCHLN,TDN,ALTER,HULZN,HISTON,TANSN,LEVELN)
options(datadist='dd')
S<-Surv(NACHZT,STATUS)
nn<-phnnet(S~GESCHLN+TDN+HULZN+HISTON+LEVELN+TANSN+ALTER,
           decay = 0.01,size=3,skip=T,trace=T,maxit=1000,bias.decay=25)
training.om$pred.nn2 <-predict(nn)

detach()
attach(test.om)
GESCHLN<-factor(GESCHL)
HISTON<-factor(HISTONEU)
HULZN<-factor(HULZ)
TANSN<-factor(TANS)
LEVELN<-ordered(LEVEL)
TDN<-log(TD)
dd<-datadist(GESCHLN,TDN,ALTER,HULZN,HISTON,TANSN,LEVELN)
options(datadist='dd')
S<-Surv(NACHZT,STATUS)
test.om$pred.nn2 <-predict(nn,newdata=test.om)
```

```
# Rangkorrelationen für zensierte Daten
attach(training.mm)
S<-Surv(NACHZT,STATUS)
rcorr.cens(-pred.td,S)
rcorr.cens(-pred.rpart,S)
rcorr.cens(-pred.cox,S)
rcorr.cens(-pred.nn,S)
rcorr.cens(-pred.nn2,S)
detach()

attach(test.mm)
S<-Surv(NACHZT,STATUS)
rcorr.cens(-pred.td,S)
rcorr.cens(-pred.rpart,S)
rcorr.cens(-pred.cox,S)
rcorr.cens(-pred.nn,S)
rcorr.cens(-pred.nn2,S)
detach()
```