

## 2 Binäre Regression

### Lösung zu Aufgabe 2

- a) siehe Übungsmitschrift
- b) siehe Übungsmitschrift

```
c) # keine individuellen Beobachtungen gegeben -> gruppierte Daten, gruppiertes Logit-Modell
# n willkürlich festlegen
n <- 10 # estimates unabhängig von konkretem n, nicht aber se -> z -> p
probs <- c(0.9, 0.7, 0.6, 0.4)
df.unemployment <- data.frame(yes = probs * n, no = n * (1-probs),
                              xG = c(1,1,0,0), xA = c(1,0,1,0))
glm.unemployment <- glm(cbind(yes, no) ~ xG*xA, data = df.unemployment,
                        family = binomial())
summary(glm.unemployment) # deviance residuen = 0, da perfekte anpassung moeglich

##
## Call:
## glm(formula = cbind(yes, no) ~ xG * xA, family = binomial(),
##      data = df.unemployment)
##
## Deviance Residuals:
## [1] 0 0 0 0
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4055     0.6455  -0.628  0.530
## xG           1.2528     0.9449   1.326  0.185
## xA           0.8109     0.9129   0.888  0.374
## xG:xA        0.5390     1.5558   0.346  0.729
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 6.1563e+00 on 3 degrees of freedom
## Residual deviance: -1.5543e-15 on 0 degrees of freedom
## AIC: 18.071
##
## Number of Fisher Scoring iterations: 4

# vgl. fitted(glm.unemployment)
```

- d) siehe Übungsmitschrift

```
e) # Loglikelihood des unrestringierten Modells:
logLik(glm.unemployment)

## 'log Lik.' -5.035414 (df=4)
```

```

# Modell unter H0:
glm.unemployment.h0 <- glm(cbind(yes, no) ~ xG + xA, data = df.unemployment,
  family=binomial)
logLik(glm.unemployment.h0)

## 'log Lik.' -5.096811 (df=3)

# LQ-Statistik:
(lambda <- -2*(logLik(glm.unemployment.h0) - logLik(glm.unemployment)))

## 'log Lik.' 0.122793 (df=3)

# Testentscheidung zu alpha = 0.05:
(quant <- qchisq(0.95, 1))

## [1] 3.841459

lambda > quant

## [1] FALSE

# p-Wert:
1 - pchisq(lambda, 1) # 0.72

## 'log Lik.' 0.7260248 (df=3)

# Problem: der wahre Stichprobenumfang und die Häufigkeit der einzelnen xG-xA-
# Kombinationen sind hier nicht bekannt. Was passiert z.B. bei doppelt sovielen
# aber identischen Beobachtungen?
n.2 <- 20
df.unemployment.2 <- data.frame(yes = probs * n.2, no = n.2 * (1-probs),
  xG = c(1,1,0,0), xA = c(1,0,1,0))

# Modelle
glm.unemployment.2 <- glm(cbind(yes, no)~ xG*xA, data = df.unemployment.2,
  family = binomial())
glm.unemployment.2.h0 <- glm(cbind(yes, no) ~ xG + xA, data = df.unemployment.2,
  family=binomial)

# LQ-Statistik:
(lambda2 <- -2*(logLik(glm.unemployment.2.h0) - logLik(glm.unemployment.2)))

## 'log Lik.' 0.245586 (df=3)

# p-Wert:
1 - pchisq(lambda2, 1)

## 'log Lik.' 0.6202004 (df=3)

# Vergleich:
c(lambda, lambda2)

## [1] 0.122793 0.245586

## => wie bereits bekannt gilt auch hier: Signifikanz hängt vom Stichproben-
## umfang ab!

# elegantere Berechnung:
anova(glm.unemployment, glm.unemployment.h0, test = "LRT")

```

```
## Analysis of Deviance Table
##
## Model 1: cbind(yes, no) ~ xG * xA
## Model 2: cbind(yes, no) ~ xG + xA
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         0     0.00000
## 2         1     0.12279 -1 -0.12279    0.726

anova(glm.unemployment.2, glm.unemployment.2.h0, test = "LRT")

## Analysis of Deviance Table
##
## Model 1: cbind(yes, no) ~ xG * xA
## Model 2: cbind(yes, no) ~ xG + xA
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         0     0.00000
## 2         1     0.24559 -1 -0.24559    0.6202
```