Serie 9

Logistische Regression: Zuerst "einfach", dann multipel:

1. Hier geht es um einen Anästhesie-Datensatz. Ziel ist eine gute Dosierung des Betäubungsmittels zu finden. Dazu wurden in einer Studie bei 30 Personen 6 verschiedene Konzentrationen angewendet und (nach einer best. Zeit) festgestellt, ob die Person auf eine Reizung „zuckt" ($\tilde{Y} = 0$) oder still ist ($\tilde{Y} = 1$). Die abhängige Variable $\tilde{Y}$ ist also binär, bzw. binomial ($Y_k := \sum_{i;\text{gleiche Konz. } x_k} \tilde{Y}_i$) nach Zusammenfassen.
Unser Modell ist $Y_k \sim Bin(\pi_k, m_k)$ $(k = 1, \dots 6)$, wobei die Wahrscheinlichkeiten π_k von der Konzentration abhängen, als $logit(\pi_k(X_k)) = \beta_0 + \beta_1 X_k$.

a) Zuerst wollen wir selber logit–transformieren: Sie können die (zusammengefassten) Daten selber eingeben:

```
Y _ c(1,1,4,4,4,2)                  # Anzahl ''still''
m _ c(7,5,6,6,4,2)                  # Total-Anzahl
X _ c(0.8,1.0,1.2,1.4,1.6,2.5)      # Konzentration des Anaestheticums
p.dach _ Y/m                        # Anteil ''Stiller'' = geschätzte pi[k]
p.tr _ (m*p.dach +1/2)/(m +1)       # Korrektur (weshalb wohl) ?
logit.p _ log(p.tr/(1-p.tr))        # logit-Transformation: logit(p) := log<p/(1 - p)>
            # #-- Graphik-Fenster oeffnen, falls noetig: -> openlook()
plot(X,logit.p)                     # zeichnet das Streudiagramm
```

b) Jetzt wollen wir die Logistische Regression durchführen. Weil die Daten schon gruppiert (statt `0/1`) sind, muss der Aufruf hier lauten
`Res.. <- glm( cbind(Y, m - Y) ~ X, family = binomial)`.
Zeichnen Sie die Gerade im transformierten Modell ein mit `abline(glm.anae)`.
Vergleichen Sie mit `lm(logit.p ~ X)` und dessen Gerade. Interpretation?

***c)** Zeichnen Sie das untransformierte Modell. Statt einer Gerade erhalten Sie eine logistische Kurve (S-plus: `lines(.)` verwenden !)

2. Hier wollen wir das Überleben von Frühgeborenen mit sehr kleinem Geburtsgewicht betrachten. Es handelt sich um eine retrospektive Studie (Hibbard, 1986) von 247 Babies. Die Daten sind schon in S-plus verfügbar als data.frame `d.baby`.

Die Matrix enthält folgende Spalten:

1:	*Survival*	(1=überlebt,0=gestorben)
2:	*Weight*	(Gewicht in Gramm)
3:	*Age*	(Alter in Wochen)
4:	*1min Apgar*	(Test 1 Min. nach Geburt, score 0–9)
5:	*5min Apgar*	(Test 5 Min. nach Geburt, score 0–9))
6:	*pH*	(pH-Wert des venösen Blutes)

a) Eine Übersicht über die Daten erhält man mit `summary(d.baby)`.
Passen Sie nun mit `glm` dieses Modell an.

b) Betrachten Sie die Residuenplots. Kommentar? Wie sehen die Parameterschätzungen aus?

c) Bestimmen Sie die Faktoren, welche das Überleben beeinflussen, indem Sie das Modell soweit als möglich vereinfachen. Das Vorgehen ist hier genau gleich wie in der multiplen linearen Regression. Wie lautet das definitive Modell?

S–Plus Befehl `glm(..)`

Für log-lineare und weitere Modelle verwenden die Formulierung als *verallgemeinertes lineares Modell* = *"Generalized Linear Model"* mit S-plus Funktion

`glm(` *formula*`, data=`*dataframe*`, family=`*linkclass* `),`

wobei mit *linkclass* z.B. `binomial`, `poisson` oder `normal` gemeint ist (ohne '"..."'). (Die üblichen linearen Modelle (Regression, VA.) sind ein Spezialfall für `family = normal`).

formula hat die gleiche Syntax wie in `lm` und ermöglicht eine elegante Formulierung von Modellen mit Wechselwirkungen, hier für den Fall zweier „Faktoren":

$A + B$	Additive Terme
$A : B$	Wechselwirkung („Interaction") zwischen A und B
$A * B$	$A + B + A : B$
1	Intercept (falls keine andern)

Um eine brauchbare und vollständige Darstellung des durch `glm()` berechneten Modells zu erhalten, kann man die Funktionen `print()`, `summary()`, `anova()` und auch `plot` verwenden, wobei man das mit `glm`(..) berechnete Resultat jeweils als Argument übergeben muss.

Abgabe: Mo, 20. Jan. 97.
Präsenz, Hilfe: Im SOL F5 (Sonneggstr.33), gemäss Vereinbarung: Tel. (63) 2-3408 oder via e-mail To: `maechler@stat.math.ethz.ch`.