Verallg. lineare Modelle (Rg-2a) — Serie 1 (logistische Regression)

1. Das Data Frame `heart.dat` enthält die nach Alter sortierten Daten von 99 Personen. Für jede Altersgruppe age_k ist die totale Anzahl Personen (m_k) und die Anzahl Personen $\tilde{y}_k$ mit Symptomen einer Herzkrankheit gegeben.

a) Schauen Sie sich die Daten $\tilde{y}/m \sim age$ an.

b) Schätzen Sie die Parameter einer einfachen logistischen Regression, welche die Wahrscheinlichkeit, Symptome zu zeigen, mit dem Alter in Beziehung setzt. Testen Sie die Hypothese, dass das Alter (`age`) die Wahrscheinlichkeit, Symptome zu zeigen, beeinflusst.

R-Hinweise:
Falls ($\tilde{Y}_k \sim \mathcal{B}(m_k, \pi_k)$) ist mit $m_k > 1$, wird die Zielvariable als Matrix (mit 2 Spalten) eingegeben, wobei in der ersten Spalte die Anzahl "Erfolge" ($\tilde{Y}_k$) und in der zweiten Spalte die Anzahl "Misserfolge" ($m_k - \tilde{Y}_k$) stehen. Sie können die logistische Regression entweder mit der Funktion `glm` rechnen, oder mit der Funktion `regr`, die Sie aus dem Block *lineare Regression* kennen, und die Sie mit `source("ftp://.../R/regr.R")` laden können.

```
> r.glm <- glm(cbind(y, m-y)~age, family=binomial, data=d.heart)
> r.glm.regr <- regr(cbind(y,m-y)~age, data=d.heart, method="glm",
                     family="binomial")
```

c) Überprüfen Sie die Residuen. Für den Tukey-Anscombe Plot können Sie die Funktion `TA.plot` benutzen.

R-Hinweise:

```
> TA.plot(r.glm.regr, res=..., labels="*", show.call=F)
> termplot(r.glm.regr, partial=TRUE, rug=TRUE)
...
```

Bemerkung: Wenn Sie mit `regr` arbeiten, erhalten Sie mit `plot(r.glm.regr)` nicht nur die beiden oben beschriebenen Plots frei Haus, sondern zusätzlich noch den Leverage-Plot.

d) Zeichnen Sie die logistische Regressions-Kurve und schätzen Sie das Alter, bei welchem Sie erwarten würden, dass 10%, 20%, ..., 90% der Personen Symptome zeigen. Diskutieren Sie diese Resultate.

R-Hinweise:
Verwenden Sie die Funktion `predict` zur Bestimmung der erwarteten Wahrscheinlichkeiten:

```
> r.pred <- predict(r.glm.regr, newdata=data.frame(age=0:100),
  type="response")
```

Die tatsächlich beobachteten Werte kann man direkt aus dem Dataframe holen und zeichnen:

```
> plot(d.heart$age, d.heart$y/d.heart$m, xlim=c(0,100), ylim=c(0,1))
```

In diesen Plot können die erwarteten Werte mit `lines(...,r.pred)` dann eingezeichnet werden.

Quelle: D.W. Hosmer and S. Lemeshow (1989), *Applied Logistic Regression*, Wiley, New York, p. 3.

2. Nach der kantonalen Abstimmung über die Initiative zur Trennung von Kirche und Staat vom 24. Sept. 1995 wurden 401 Stimmberechtigte des Kantons Zürich, die an der Abstimmung teilgenommen hatten, über ihr Abstimmungsverhalten und zu ihrer Person befragt. 393 davon gaben vollständige Antworten, welche im Dataframe `initiat.dat` mit u. a. folgenden Variablen enthalten sind:

`y`	Abstimmungsverhalten: 1 = zugestimmt, 0 = abgelehnt
`sex`	Geschlecht: m = Mann, w = Frau
`polit`	politischer Standort: rechts (SVP, FDP, FPS), Mitte (CVP, EVP, LdU), links (SP, Grüne), keine.

Man möchte nun der Frage nachgehen, welchen Einfluss die Variablen `sex` und `polit` auf das Abstimmungsverhalten hatten. Lesen Sie dazu zunächst den Datensatz ein:

```
> d.initiative <- read.table("http://stat.ethz.ch/Teaching/Datasets/NDK/
+                                          initiat.dat", sep="\t", header=T)
```

a) Verschaffen Sie sich zuerst einen graphischen Überblick über die Daten. Welche Personengruppen scheinen dem Anliegen der Initiative eher positiv gegenüberzustehen?
R-Hinweise:

```
> plot.design(d.initiative[,c(1,2,4)])
```

b) Formulieren Sie das Modell der logistischen Regression. Geben Sie die Zielvariable und deren Verteilung an. Nennen Sie auch die erklärenden Variablen und die Link-Funktion.

c) Schätzen Sie Ihr Modell mit R und geben Sie einen kurzen Kommentar zur Güte des Modells (Likelihood-Ratio-Test).

d) Wie gross ist die Wahrscheinlichkeit, dass eine Zürcherin mit politischem Standort “links” die Initiative ablehnt?