3 Generalisierte lineare Modelle (III)

Aufgabe 7

```
# Einlesen der Daten
leafblotch <- read.table("leafblotch.dat", header = TRUE)
# Struktur der Daten
str(leafblotch)

## 'data.frame': 90 obs. of  3 variables:
##  $ blotch : num  0.0005 0 0 0.001 0.0025 0.0005 0.005 0.013 0.015 0.015 ...
##  $ site   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ variety: int  1 2 3 4 5 6 7 8 9 10 ...

# Die Kovariablen site und variety werden nun als Faktor-Variablen definiert
# (d.h. bei der Modellierung identifiziert sie R als Dummy-Variablen mit der
# ersten (!) Kategorie als Referenz)
leafblotch$sitef <- as.factor(leafblotch$site)
leafblotch$varietyf <- as.factor(leafblotch$variety)
```

a) Das verlangte Modell kann man in R auf zweierlei Arten rechnen, mit family=quasibinomial (hier wird automatisch von der Varianzfunktion $v(\mu) = \mu(1-\mu)$, Modellierung der Dispersion) ausgegangen:

```
quasiglm1a <- glm(blotch ~ sitef + varietyf, data = leafblotch,
        family = quasibinomial(link = "logit"))
summary(quasiglm1a)


##
## Call:
## glm(formula = blotch ~ sitef + varietyf, family = quasibinomial(link = "logit"),
##     data = leafblotch)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.64431  -0.13546  -0.02061   0.09628   0.81005
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.0546     1.4220  -5.664 2.84e-07 ***
## sitef2        1.6391     1.4433   1.136 0.259880
## sitef3        3.3265     1.3492   2.465 0.016068 *
## sitef4        3.5822     1.3445   2.664 0.009512 **
## sitef5        3.5838     1.3444   2.666 0.009479 **
## sitef6        3.8932     1.3402   2.905 0.004876 **
## sitef7        4.7299     1.3348   3.544 0.000698 ***
## sitef8        5.5226     1.3346   4.138 9.39e-05 ***
## sitef9        6.7945     1.3407   5.068 3.00e-06 ***
## varietyf2     0.1501     0.7237   0.207 0.836293
## varietyf3     0.6895     0.6724   1.025 0.308599
## varietyf4     1.0481     0.6494   1.614 0.110919
## varietyf5     1.6147     0.6257   2.581 0.011897 *
```

```
## varietyf6     2.3711     0.6090   3.893 0.000219 ***
## varietyf7     2.5712     0.6065   4.240 6.55e-05 ***
## varietyf8     3.3419     0.6015   5.556 4.39e-07 ***
## varietyf9     3.4999     0.6014   5.820 1.51e-07 ***
## varietyf10    4.2529     0.6042   7.038 9.39e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 0.08878094)
##
##     Null deviance: 40.8029  on 89  degrees of freedom
## Residual deviance:  6.1264  on 72  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 8
```

oder mit family = quasi. Diese Funktion ist allgemeiner (funktioniert für verschiedene Link-Funktionen, z.B. auch "log", vgl. Quasi-Poisson-Modelle), man muss die Varianzfunktion explizit spezifizieren (es sind verschiedene weitere Möglichkeiten implementiert, z.B. mu^2 oder mu^3):

```
quasiglm1b <- glm(blotch ~ sitef + varietyf, data = leafblotch,
        family = quasi(link = "logit", var = "mu(1-mu)"))
summary(quasiglm1b)
```

```
##
## Call:
## glm(formula = blotch ~ sitef + varietyf, family = quasi(link = "logit",
##     var = "mu(1-mu)"), data = leafblotch)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.64431  -0.13546  -0.02061   0.09628   0.81005
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.0546     1.4220  -5.664 2.84e-07 ***
## sitef2        1.6391     1.4433   1.136 0.259881
## sitef3        3.3265     1.3492   2.465 0.016069 *
## sitef4        3.5822     1.3445   2.664 0.009512 **
## sitef5        3.5838     1.3444   2.666 0.009479 **
## sitef6        3.8932     1.3402   2.905 0.004877 **
## sitef7        4.7299     1.3348   3.544 0.000698 ***
## sitef8        5.5226     1.3346   4.138 9.39e-05 ***
## sitef9        6.7945     1.3407   5.068 3.00e-06 ***
## varietyf2     0.1501     0.7237   0.207 0.836293
## varietyf3     0.6895     0.6724   1.025 0.308599
## varietyf4     1.0481     0.6494   1.614 0.110919
## varietyf5     1.6147     0.6257   2.581 0.011897 *
## varietyf6     2.3711     0.6090   3.893 0.000219 ***
## varietyf7     2.5712     0.6065   4.240 6.55e-05 ***
## varietyf8     3.3419     0.6015   5.556 4.39e-07 ***
## varietyf9     3.4999     0.6014   5.820 1.51e-07 ***
## varietyf10    4.2529     0.6042   7.038 9.39e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasi family taken to be 0.08878102)
##
##     Null deviance: 40.8029  on 89  degrees of freedom
## Residual deviance:  6.1264  on 72  degrees of freedom
```

```
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

Man erkennt: Beide Verfahren liefern denselben Output. Der geschätzte Dispersionsparameter ϕ ist mit 0.0887 weit unter dem Wert 1 (wie er bei Annahme einer Binomialverteilung festgelegt wäre). Bei Betrachtung der geschätzten Parameter erweisen sich die Sorten 1 und 2 als am stärksten resistent gegen die Blattkrankheit, während die Sorten 8-10 am stärksten befallen sind (in dieser Reihenfolge). Ähnliches gilt für die Felder.

```
# Beachte, dass man beim Schätzen unter Annahme einer Binomialverteilung die
# selben Schätzer erhält (aber andere Werte für Tests etc.); plus Warnung,
# weil Werte zwischen 0 und 1 bei Binomialverteilung ja nicht vorkommen dürften.
binomialModell  <- glm(blotch ~ sitef + varietyf, data = leafblotch,
        family = binomial)
summary(binomialModell)
```

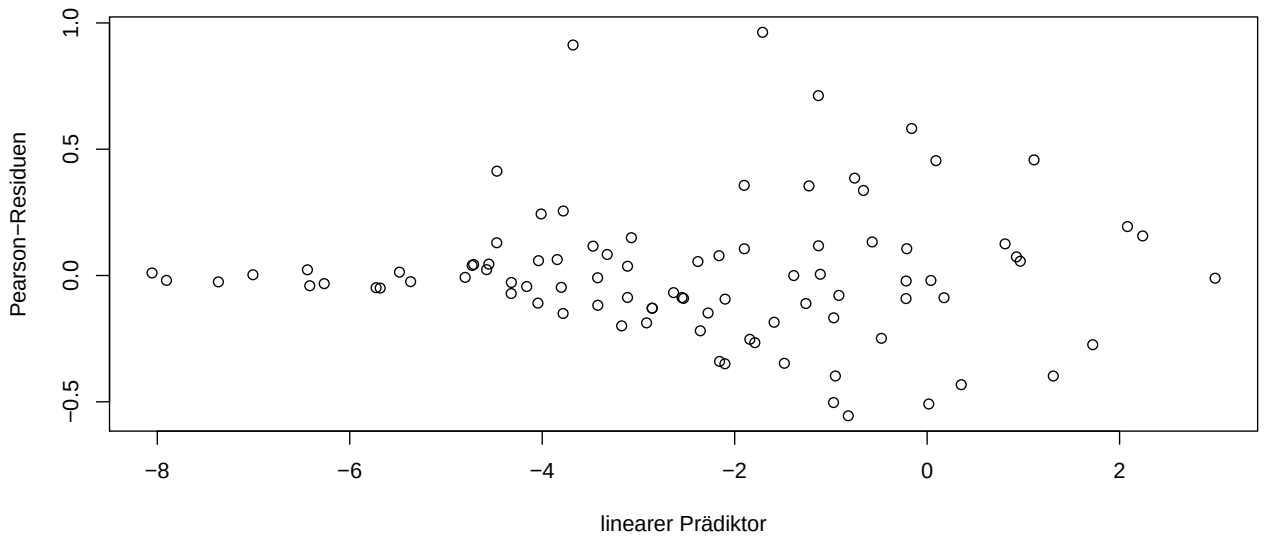
```
##
## Call:
## glm(formula = blotch ~ sitef + varietyf, family = binomial, data = leafblotch)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.64431  -0.13546  -0.02061   0.09628   0.81005
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.0546     4.7723  -1.688   0.0915 .
## sitef2        1.6391     4.8440   0.338   0.7351
## sitef3        3.3265     4.5282   0.735   0.4626
## sitef4        3.5822     4.5122   0.794   0.4273
## sitef5        3.5838     4.5121   0.794   0.4270
## sitef6        3.8932     4.4980   0.866   0.3867
## sitef7        4.7299     4.4798   1.056   0.2910
## sitef8        5.5226     4.4792   1.233   0.2176
## sitef9        6.7945     4.4996   1.510   0.1310
## varietyf2     0.1501     2.4288   0.062   0.9507
## varietyf3     0.6895     2.2566   0.306   0.7600
## varietyf4     1.0481     2.1796   0.481   0.6306
## varietyf5     1.6147     2.0998   0.769   0.4419
## varietyf6     2.3711     2.0440   1.160   0.2460
## varietyf7     2.5712     2.0354   1.263   0.2065
## varietyf8     3.3419     2.0189   1.655   0.0979 .
## varietyf9     3.4999     2.0182   1.734   0.0829 .
## varietyf10    4.2529     2.0279   2.097   0.0360 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 40.8029  on 89  degrees of freedom
## Residual deviance:  6.1264  on 72  degrees of freedom
## AIC: 70.44
##
## Number of Fisher Scoring iterations: 8
```

```
all.equal(coef(binomialModell), coef(quasiglm1a))
```

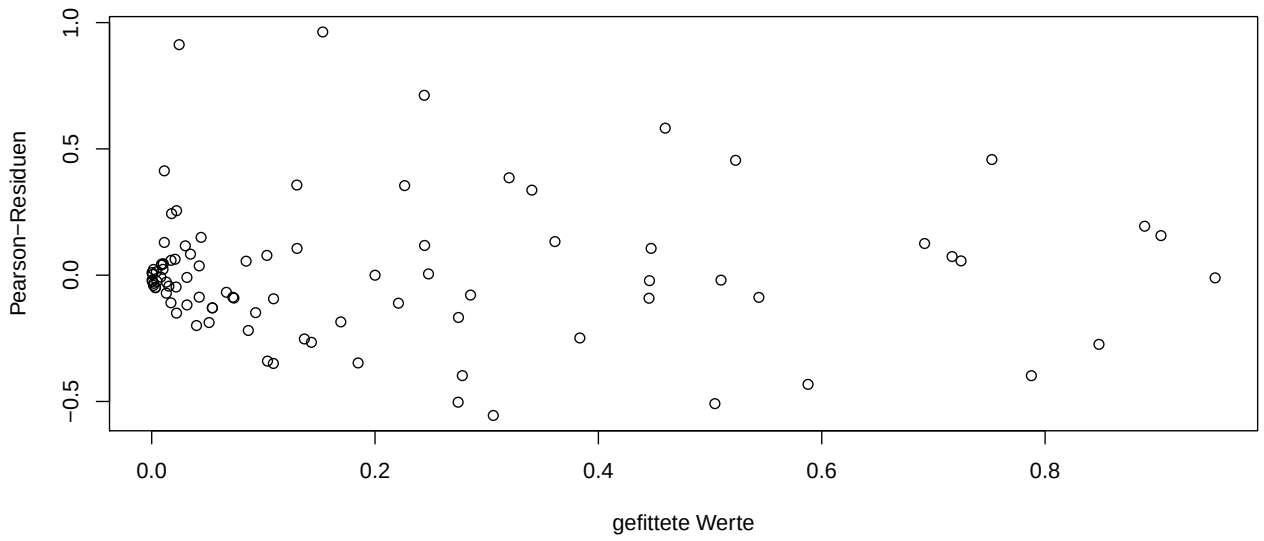
```
## [1] TRUE
```

b) Ein geeigneter Diagnostik-Plot (vgl. auch später) wäre, die Pearson-Residuen $r_i^P = r_i/v(\hat{\mu}_i)$ gegen den geschätzten linearen Prädiktor (im GLM oft besser als gefittete Werte; auch wg. Skalierung, z.B. Poisson- oder Binomial-Modell, hier aber weniger relevant)

```
# gegen linearen Prädiktor
plot(predict(quasiglm1b), residuals(quasiglm1b, type = "pearson"),
xlab = "linearer Prädiktor", ylab = "Pearson-Residuen")
```




```
# gegen gefittete Werte
plot(predict(quasiglm1b, type = "response"), residuals(quasiglm1b, type = "pearson"),
xlab = "gefittete Werte", ylab = "Pearson-Residuen")
```



Im Idealfall sollte man (wie im linearen Modell) keine Struktur erkennen können. Ferner sollte der Plot eine konstante Varianz aufweisen. Das ist hier nicht der Fall (die Varianz ist für kleine Werte des linearen Prädiktors geringer; bis ca. -5). Das ist ein Hinweis, dass die Varianzfunktion nicht gut gewählt ist.

c)
```
# D(\beta) als Funktion
Dbeta <- function(eta) {
  1/(exp(-eta) + 2 + exp(eta))
}

# Design-Matrix X zusammenbauen
# leafblotch
```

```r
X <- cbind(matrix(0, 10, 8), rbind(0, diag(1, 9)))
for (si in 1:8) {
    A <- matrix(0, 10, 8)
    A[, si] <- 1
    X <- rbind(X, cbind(A, rbind(0, diag(1, 9))))
}
X <- cbind(1, X)

# V-Marix (siehe Skript)
V <- t(X) %*% diag(Dbeta(predict(quasiglm1b))^2 *
        residuals(quasiglm1b,type = "pearson")^2 /
        (summary(quasiglm1b)$dispersion^2 * quasiglm1b$fitted *
                (1-quasiglm1b$fitted))) %*% X

# geschätzte Kovarianz-Matrix F^{-1}VF^{-1}
Covbeta <- summary(quasiglm1b)$cov.scaled %*% V %*%
        summary(quasiglm1b)$cov.scaled

# Varianzen (Beachte aber: wegen der vergleichsweise geringen Zahl an
# Beobachtungen ist diese Schätzungen vermutlich nicht besonders gut)
diag(Covbeta)
```

```
## (Intercept)      sitef2      sitef3      sitef4      sitef5      sitef6
##  0.11595372  0.09643726  0.13526879  0.21318768  0.05197312  0.17105540
##      sitef7      sitef8      sitef9   varietyf2   varietyf3   varietyf4
##  0.14122397  0.08774074  0.11405413  0.07121236  0.10481827  0.16010020
##   varietyf5   varietyf6   varietyf7   varietyf8   varietyf9  varietyf10
##  0.31519837  0.23589968  0.14188218  0.18410656  0.12011451  0.11577316
```

```r
# bzw. Standardfehler
sqrt(diag(Covbeta))
```

```
## (Intercept)      sitef2      sitef3      sitef4      sitef5      sitef6
##   0.3405198   0.3105435   0.3677891   0.4617225   0.2279761   0.4135884
##      sitef7      sitef8      sitef9   varietyf2   varietyf3   varietyf4
##   0.3757978   0.2962106   0.3377190   0.2668564   0.3237565   0.4001252
##   varietyf5   varietyf6   varietyf7   varietyf8   varietyf9  varietyf10
##   0.5614253   0.4856951   0.3766725   0.4290764   0.3465754   0.3402546
```

```r
# Vergleich mit Standardfehlern falls Var(y)  =  \phi\mu(1-\mu) richtig spezifiziert
sqrt(diag(summary(quasiglm1b)$cov.scaled))
```

```
## (Intercept)      sitef2      sitef3      sitef4      sitef5      sitef6
##   1.4219754   1.4433157   1.3492405   1.3444569   1.3444302   1.3402298
##      sitef7      sitef8      sitef9   varietyf2   varietyf3   varietyf4
##   1.3348001   1.3346434   1.3407134   0.7236855   0.6723672   0.6494401
##   varietyf5   varietyf6   varietyf7   varietyf8   varietyf9  varietyf10
##   0.6256686   0.6090223   0.6064769   0.6015408   0.6013526   0.6042328
```

```r
# berechenbar durch
W<-diag(Dbeta(predict(quasiglm1b))^2 /
          (quasiglm1b$fitted *(1-quasiglm1b$fitted))/summary(quasiglm1b)$dispersion)
F2<-t(X) %*% W %*% X
sqrt(diag(solve(F2)))
```

```
##  [1] 1.4219757 1.4433160 1.3492407 1.3444571 1.3444304 1.3402300 1.3348003
##  [8] 1.3346436 1.3407136 0.7236857 0.6723674 0.6494404 0.6256688 0.6090225
## [15] 0.6064771 0.6015410 0.6013528 0.6042330
```

d)
```r
#install.packages("gnm")
library(gnm)

# Modell fitten
glmwedderburn <- glm(blotch ~ sitef + varietyf, data = leafblotch,
        family = wedderburn)

# Output
summary(glmwedderburn)
```

```
##
## Call:
## glm(formula = blotch ~ sitef + varietyf, family = wedderburn,
##     data = leafblotch)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1807  -0.6293   0.0000   0.3888   1.9711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.92251    0.44463 -17.818  < 2e-16 ***
## sitef2       1.38306    0.44463   3.111  0.00268 **
## sitef3       3.86009    0.44463   8.682 8.19e-13 ***
## sitef4       3.55697    0.44463   8.000 1.53e-11 ***
## sitef5       4.10836    0.44463   9.240 7.48e-14 ***
## sitef6       4.30535    0.44463   9.683 1.13e-14 ***
## sitef7       4.91810    0.44463  11.061  < 2e-16 ***
## sitef8       5.69486    0.44463  12.808  < 2e-16 ***
## sitef9       7.06759    0.44463  15.896  < 2e-16 ***
## varietyf2   -0.46722    0.46868  -0.997  0.32216
## varietyf3    0.07883    0.46868   0.168  0.86691
## varietyf4    0.95420    0.46868   2.036  0.04544 *
## varietyf5    1.35275    0.46868   2.886  0.00514 **
## varietyf6    1.32868    0.46868   2.835  0.00594 **
## varietyf7    2.34065    0.46868   4.994 3.99e-06 ***
## varietyf8    3.26268    0.46868   6.961 1.30e-09 ***
## varietyf9    3.13558    0.46868   6.690 4.10e-09 ***
## varietyf10   3.88737    0.46868   8.294 4.33e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for wedderburn family taken to be 0.9884572)
##
##     Null deviance: 370.523  on 89  degrees of freedom
## Residual deviance:  66.267  on 72  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 13
```

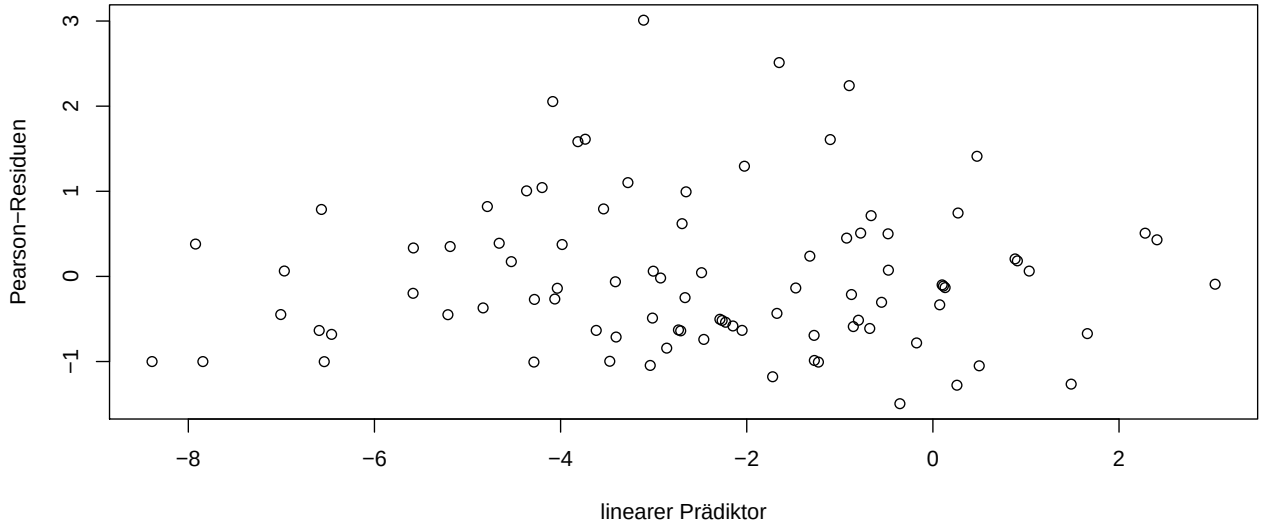
```r
# Vergleich
summary(quasiglm1b)
```

```
##
## Call:
## glm(formula = blotch ~ sitef + varietyf, family = quasi(link = "logit",
##     var = "mu(1-mu)"), data = leafblotch)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.64431  -0.13546  -0.02061   0.09628   0.81005
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.0546     1.4220  -5.664 2.84e-07 ***
## sitef2        1.6391     1.4433   1.136 0.259881
## sitef3        3.3265     1.3492   2.465 0.016069 *
## sitef4        3.5822     1.3445   2.664 0.009512 **
## sitef5        3.5838     1.3444   2.666 0.009479 **
## sitef6        3.8932     1.3402   2.905 0.004877 **
## sitef7        4.7299     1.3348   3.544 0.000698 ***
## sitef8        5.5226     1.3346   4.138 9.39e-05 ***
## sitef9        6.7945     1.3407   5.068 3.00e-06 ***
## varietyf2     0.1501     0.7237   0.207 0.836293
## varietyf3     0.6895     0.6724   1.025 0.308599
## varietyf4     1.0481     0.6494   1.614 0.110919
## varietyf5     1.6147     0.6257   2.581 0.011897 *
## varietyf6     2.3711     0.6090   3.893 0.000219 ***
## varietyf7     2.5712     0.6065   4.240 6.55e-05 ***
## varietyf8     3.3419     0.6015   5.556 4.39e-07 ***
## varietyf9     3.4999     0.6014   5.820 1.51e-07 ***
## varietyf10    4.2529     0.6042   7.038 9.39e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasi family taken to be 0.08878102)
##
##     Null deviance: 40.8029  on 89  degrees of freedom
## Residual deviance:  6.1264  on 72  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

Man erkennt: die Parameterschätzer verändern sich. Man beachte, dass sich die Reihenfolge bei den Parametern für die Sorten nun etwas ändert, auch wenn die grobe Tendenz die selbe ist wie zuvor. Der Skalenparameter nimmt einen Wert von 0.988 an.

```
# Der Plot der Pearson-Residuen gegen den geschätzten linear Prädiktor,
 plot(predict(glmwedderburn), residuals(glmwedderburn, type = "pearson"),
xlab = "linearer Prädiktor", ylab = "Pearson-Residuen")
```




```
# weist deutlich stärker in Richtung von gleichen Varianzen als zuvor, was
# darauf hindeutet, dass es sich hierbei um das bessere Modell handelt.
```