# Aufgabe 1

**a)**

```
load('awards.RData')
load('DebTrivedi.RData')

plm = glm(
  num_awards ~ prog + math,
  data = awards,
  family= poisson
)

lambda_voc <- predict.glm(plm, data.frame(prog = "Vocational", math = 60), type="response")
lambda_acd <- predict.glm(plm, data.frame(prog = "Academic", math = 60), type="response")
lambda_gen <- predict.glm(plm, data.frame(prog = "General", math = 60), type="response")

1- sum(dpois(0:2, lambda_gen))
#> [1] 0.005690656
1- sum(dpois(0:2, lambda_voc))
#> [1] 0.0153646
1- sum(dpois(0:2, lambda_acd))
#> [1] 0.08913637
```

**b)**

```
library("lmtest")
#> Warning: package 'lmtest' was built under R version 3.5.3

plm_without = glm(
  num_awards ~ math,
  data = awards,
  family= poisson
)

lmtest::lrtest(plm, plm_without)
#> Likelihood ratio test
#>
#> Model 1: num_awards ~ prog + math
#> Model 2: num_awards ~ math
#>   #Df  LogLik Df  Chisq Pr(>Chisq)
#> 1   4 -182.75
#> 2   2 -190.04 -2 14.572  0.0006852 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
AIC(plm_without, plm)
#>             df      AIC
#> plm_without  2 384.0762
#> plm          4 373.5045
```

Um zu überprüfen, ob der Ausbildungstyp einen signifikanten Einfluss auf die Anzahl an Awards hat, wurden zwei Poissonregressionen gefittet (1. num_awards ~ prog + math; 2. num_awards ~ math) und mittels AIC

und Likelihood-Ratio-Test miteinander verglichen.

Beide Vergleichsmethoden kommen zum Ergebnis, dass der Ausbildungsyp ("prog") einen Einfluss auf die Zielgröße ("num_awards") hat.

**c)**

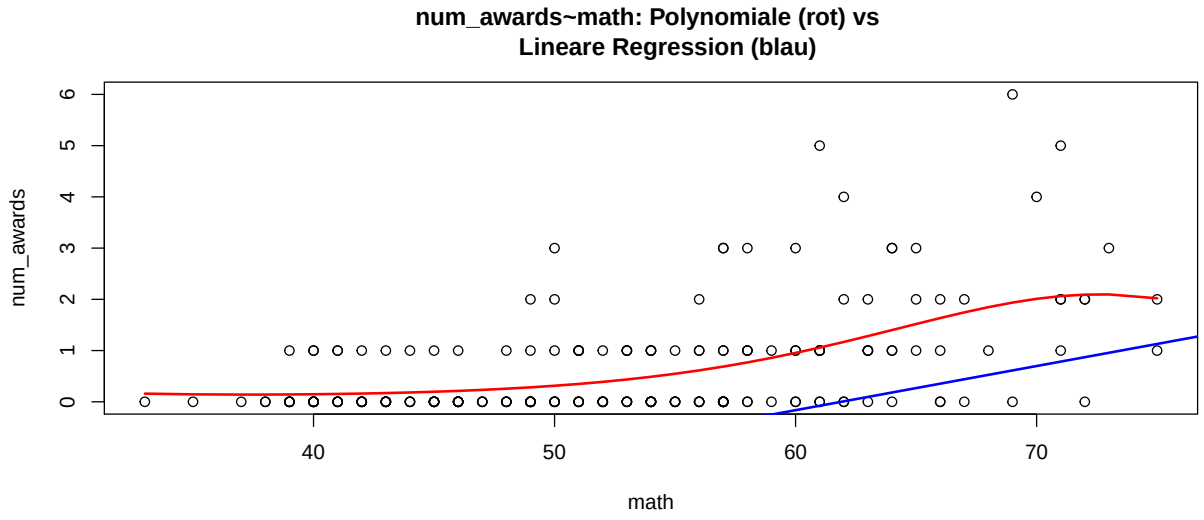
```
anova <- aov(num_awards ~ prog+math, data=awards)
summary(anova)
#>              Df Sum Sq Mean Sq F value   Pr(>F)
#> prog          2  30.30  15.150   18.62 3.94e-08 ***
#> math          1  30.88  30.877   37.96 4.03e-09 ***
#> Residuals   196 159.44   0.813
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Um den Einfluss der Variablen auf die Zielgröße zu überprüfen, wurde eine Varianzanalyse (Anova) durchgeführt. Laut dieser haben sowohl math als auch prog einen signifikaten Einfluss auf num_awards.

**d)**

```
poly <- glm(num_awards ~ math + I(math^2) + I(math^3), data=awards, family='poisson')

plot(num_awards~math, data=awards, main="num_awards~math: Polynomiale (rot) vs
     Lineare Regression (blau)")
lines(sort(awards$math), fitted(poly)[order(awards$math)], col='red', lw=2)
abline(plm_without, col='blue', lw=2)
```




```
AIC(poly, plm_without)
#>             df      AIC
#> poly         4 385.5319
#> plm_without  2 384.0762
anova(poly, plm_without, test='Chisq')
#> Analysis of Deviance Table
```

```
#>
#> Model 1: num_awards ~ math + I(math^2) + I(math^3)
#> Model 2: num_awards ~ math
#>   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
#> 1       196     201.48
#> 2       198     204.02 -2  -2.5442   0.2802
```

Laut AIC ist das polynomiale Modell 3. Grades nicht signifikant besser als das lineare Modell (beide mit math als einziger Kovariate).

# Aufgabe 2

**a)**

```
library(tidyr)
poisreg <- glm(ofp~health+numchron+hosp+married+medicaid, data=DebTrivedi, family = 'poisson')

dev_model <- deviance(poisreg)

coeff <- poisreg$coefficients

data_a  <- DebTrivedi
data_a$ofp[data_a$ofp == 0] <- 0.000001

data_a <- spread(data_a, health, health)
data_a$poor <- as.numeric(data_a$poor)
data_a$poor[is.na(data_a$poor)] <- 0
data_a$average <- as.numeric(data_a$average)
data_a$average[is.na(data_a$average)] <- 0
data_a$average[data_a$average == 2] <- 1
data_a$excellent <- as.numeric(data_a$excellent)
data_a$excellent[is.na(data_a$excellent)] <- 0
data_a$excellent[data_a$excellent == 3] <- 1
data_a <- spread(data_a, medicaid, medicaid, sep='')
data_a$medicaidyes <- as.numeric(data_a$medicaidyes)
data_a$medicaidno <- as.numeric(data_a$medicaidno)
data_a$medicaidyes[is.na(data_a$medicaidyes)] <- 0
data_a$medicaidyes[data_a$medicaidyes == 2] <- 1
data_a$medicaidno[is.na(data_a$medicaidno)] <- 0
data_a <- spread(data_a, married, married, sep='')
data_a$marriedyes <- as.numeric(data_a$marriedyes)
data_a$marriedno <- as.numeric(data_a$marriedno)
data_a$marriedyes[is.na(data_a$marriedyes)] <- 0
data_a$marriedyes[data_a$marriedyes == 2] <- 1
data_a$marriedno[is.na(data_a$marriedno)] <- 0


ll_m <-c()

for(i in 1:nrow(data_a)){
ll_m[i]<--exp( coeff[1] +
                 coeff[2]*data_a$poor[i] +
```

```
                  coeff[3]*data_a$excellent[i] +
                  coeff[4]*data_a$numchron[i] +
                  coeff[5]*data_a$hosp[i] +
                  coeff[6]*data_a$marriedyes[i] +
                  coeff[7]*data_a$medicaidyes[i]) +
  data_a$ofp[i]*(coeff[1] + coeff[2]*data_a$poor[i] +
                   coeff[3]*data_a$excellent[i] +
                   coeff[4]*data_a$numchron[i] +
                   coeff[5]*data_a$hosp[i] +
                   coeff[6]*data_a$marriedyes[i] +
                   coeff[7]*data_a$medicaidyes[i]) -
  log(factorial(data_a$ofp[i]))
}

ll_reg <- sum(ll_m)

ll_opt <- sum(-data_a$ofp + data_a$ofp * log(data_a$ofp) - log(factorial(data_a$ofp)))

dev_manual <- -2*(ll_reg-ll_opt)

dev_model
#> [1] 24178.54
dev_manual
#> [1] 24178.51
```

**b)**

```
summary(poisreg)
#>
#> Call:
#> glm(formula = ofp ~ health + numchron + hosp + married + medicaid,
#>     family = "poisson", data = DebTrivedi)
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -5.2623  -2.0484  -0.6898   0.7949  16.1776
#>
#> Coefficients:
#>                  Estimate Std. Error z value Pr(>|z|)
#> (Intercept)      1.490539   0.013363 111.538   <2e-16 ***
#> healthpoor       0.338277   0.016536  20.457   <2e-16 ***
#> healthexcellent -0.372756   0.030241 -12.326   <2e-16 ***
#> numchron         0.095171   0.006124  15.541   <2e-16 ***
#> hosp             0.500663   0.013982  35.809   <2e-16 ***
#> marriedyes      -0.019759   0.012860  -1.537    0.124
#> medicaidyes      0.032753   0.021101   1.552    0.121
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for poisson family taken to be 1)
#>
#>     Null deviance: 26943  on 4405  degrees of freedom
```

```
#> Residual deviance: 24179  on 4399  degrees of freedom
#> AIC: 36968
#>
#> Number of Fisher Scoring iterations: 5
pchisq(poisreg$deviance, df=poisreg$df.residual)
#> [1] 1
nrow(DebTrivedi)
#> [1] 4406
```

Um die Abweichung als Gütemetrik zu nutzen, müssen wir unter der Annahme, dass unser Modell korrekt ist, herausfinden, wie viel Variation wir bei den beobachteten Ergebnissen um ihre vorhergesagten Mittel herum erwarten würden.

Da die Abweichung als Likelihood-Ratio-Test zum Vergleich des aktuellen Modells mit dem gesättigten Modell abgeleitet werden kann, wird vermutet, dass (vorausgesetzt, das Modell ist korrekt spezifiziert) die Abweichung einer Chi-Quadrat-Verteilung folgt, deren Freiheitsgrade der Differenz in der Anzahl der Parameter entsprechen. Das gesättigte Modell kann als ein Modell betrachtet werden, das für jede Beobachtung einen eigenen Parameter verwendet und somit n Parameter hat. Wenn unser Modell p-Parameter hat, bedeutet dies, dass die Abweichung mit einer Chi-Quadrat-Verteilung auf n-p-Parameter verglichen wird.

Die Abweichung wird hier von der glm-Funktion als "residual deviance" bezeichnet, hier 24180. Es gibt 4406 Beobachtungen, und unser Modell hat sechs Parameter, so dass die Freiheitsgrade 4399 sind, angegeben durch df.residual. Um den p-Wert für die Varianzgüte des Fit-Tests zu berechnen, berechnen wir einfach die Wahrscheinlichkeit rechts neben dem Varianzwert für die Chi-Quadrat-Verteilung auf 4399 Freiheitsgrade.

Die Nullhypothese ist, dass unser Modell korrekt spezifiziert ist. Ein p-Wert von 1 spricht für ein gut gefittetes Modell.

**c)**

```
poisreg2 <- glm(ofp~health+numchron+hosp+married+medicaid, data=DebTrivedi, family = 'quasipoisson')
summary(poisreg2)
#>
#> Call:
#> glm(formula = ofp ~ health + numchron + hosp + married + medicaid,
#>     family = "quasipoisson", data = DebTrivedi)
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -5.2623  -2.0484  -0.6898   0.7949  16.1776
#>
#> Coefficients:
#>                 Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      1.49054    0.03558  41.898  < 2e-16 ***
#> healthpoor       0.33828    0.04402   7.684 1.88e-14 ***
#> healthexcellent -0.37276    0.08050  -4.630 3.76e-06 ***
#> numchron         0.09517    0.01630   5.838 5.67e-09 ***
#> hosp             0.50066    0.03722  13.451  < 2e-16 ***
#> marriedyes      -0.01976    0.03423  -0.577    0.564
#> medicaidyes      0.03275    0.05617   0.583    0.560
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for quasipoisson family taken to be 7.086866)
```

```
#>
#>     Null deviance: 26943  on 4405  degrees of freedom
#> Residual deviance: 24179  on 4399  degrees of freedom
#> AIC: NA
#>
#> Number of Fisher Scoring iterations: 5
```

Bei der Poissonregression treffen wir eine starke Modellannahme:

Da bei der Poissonverteilung $\lambda$ gleich der Erwartungswert, ALS AUCH der Varianz ist, nehmen wir dies auch für die Verteilung in unserem Modell an. Dies ist häufig nicht der Fall.

Überdispersion ist ein Problem, wenn die bedingte Varianz größer ist als der bedingte Mittelwert. Um den Überdispersionparameter zu schätzen, fitten wir ein Quasi-Poisson-Modell auf unsere Daten.

Laut dem neuen Modell ist der geschätzte Überdispersionsparameter bei ~ 7. Das heißt, die bedingte Varianz ist 7-mal größer als der bedingte Mittelwert.

**d)**

```
summary(poisreg)
#>
#> Call:
#> glm(formula = ofp ~ health + numchron + hosp + married + medicaid,
#>     family = "poisson", data = DebTrivedi)
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -5.2623  -2.0484  -0.6898   0.7949  16.1776
#>
#> Coefficients:
#>                 Estimate Std. Error z value Pr(>|z|)
#> (Intercept)     1.490539   0.013363 111.538   <2e-16 ***
#> healthpoor      0.338277   0.016536  20.457   <2e-16 ***
#> healthexcellent -0.372756   0.030241 -12.326   <2e-16 ***
#> numchron        0.095171   0.006124  15.541   <2e-16 ***
#> hosp            0.500663   0.013982  35.809   <2e-16 ***
#> marriedyes     -0.019759   0.012860  -1.537    0.124
#> medicaidyes     0.032753   0.021101   1.552    0.121
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for poisson family taken to be 1)
#>
#>     Null deviance: 26943  on 4405  degrees of freedom
#> Residual deviance: 24179  on 4399  degrees of freedom
#> AIC: 36968
#>
#> Number of Fisher Scoring iterations: 5
summary(poisreg2)
#>
#> Call:
#> glm(formula = ofp ~ health + numchron + hosp + married + medicaid,
#>     family = "quasipoisson", data = DebTrivedi)
```

```
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -5.2623  -2.0484  -0.6898   0.7949  16.1776
#>
#> Coefficients:
#>                 Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      1.49054    0.03558  41.898  < 2e-16 ***
#> healthpoor       0.33828    0.04402   7.684 1.88e-14 ***
#> healthexcellent -0.37276    0.08050  -4.630 3.76e-06 ***
#> numchron         0.09517    0.01630   5.838 5.67e-09 ***
#> hosp             0.50066    0.03722  13.451  < 2e-16 ***
#> marriedyes      -0.01976    0.03423  -0.577    0.564
#> medicaidyes      0.03275    0.05617   0.583    0.560
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for quasipoisson family taken to be 7.086866)
#>
#>     Null deviance: 26943  on 4405  degrees of freedom
#> Residual deviance: 24179  on 4399  degrees of freedom
#> AIC: NA
#>
#> Number of Fisher Scoring iterations: 5
```

Es ändern sich nur die p-Werte der Koeffizienten. Dies rührt von der Veränderung der Verteilungsannahme (von Poisson zu Quasi-Poisson).

**e)**