First the medcare data are loaded:

```r
library(catdata)
data(medcare)
attach(medcare)

## Das folgende Objekt ist maskiert children:
##
## ## age
```

The dependent variable "ofp" (numbers of physician visits) is a count variable, so a poisson-family glm seems to be a good choice.

```r
med1=glm(ofp ~ hosp + healthpoor + healthexcellent + numchron + age + married + school, family=poisson, data=medcare[male==1 & ofp<=30,])
summary(med1)
```

```
## Call:
glm(formula = ofp ~ hosp + healthpoor + healthexcellent + numchron +
    age + married + school, family = poisson, data = medcare[male ==
    1 & ofp <= 30, ])
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.3338 -1.9118 -0.6178  0.8085  7.5113
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.289181   0.140378  2.060  0.0394 *
## hosp        0.161705   0.010324 15.663 < 2e-16 ***
## healthpoor  0.131090   0.031910  4.108  3.99e-05 ***
## healthexcellent -0.269974  0.047458 -5.689  1.28e-08 ***
## numchron    0.153347   0.007691 19.939 < 2e-16 ***
## age         0.076527   0.017635  4.340  1.43e-05 ***
## married     0.145469   0.027905  5.213  1.86e-07 ***
## school      0.029470   0.002858 10.311  < 2e-16 ***
```
In many real-world datasets the variance of count-data is higher than predicted by the Poisson distribution, so we fit a quasi-Poisson model with dispersion parameter.

```r
med2 = glm(ofp ~ hosp + healthpoor + healthexcellent + numchron + age + married + school,
  family = quasipoisson, data = medcare[male == 1 & ofp <= 30, ])
summary(med2)
```

```
Call:
glm(formula = ofp ~ hosp + healthpoor + healthexcellent + numchron +
    age + married + school, family = quasipoisson, data = medcare[male ==
    1 & ofp <= 30, ])

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-5.3338  -1.9118  -0.6178   0.8085   7.5113

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.289181   0.304171   0.951   0.34188
hosp        0.161705   0.022371  7.228 7.26e-13 ***
healthpoor  0.131090   0.069142  1.896   0.05813 .
healthexcellent -0.269974  0.102833 -2.625  0.00873 **
numchron    0.153347   0.016664  9.202  < 2e-16 ***
age         0.076527   0.038211  2.003  0.04536 *
married     0.145469   0.060465  2.406  0.01624 *
school     0.029470   0.006193  4.759 2.11e-06 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 4.695025)

Null deviance: 8830.3 on 1760 degrees of freedom
Residual deviance: 7655.9 on 1753 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```
With an estimated dispersion parameter of 4.69 the standard errors are much bigger now. An alternative to a quasi-poisson model is to use the negative binomial distribution.

```
library(MASS)
med3=glm.nb(ofp ~ hosp+healthpoor+healthexcellent+numchron+age+married+school, 
data=medcare[male==1 & ofp<=30,])
summary(med3)
```

```
## Call:
glm.nb(formula = ofp ~ hosp + healthpoor + healthexcellent + 
        numchron + age + married + school, data = medcare[male == 
        1 & ofp <= 30, ], init.theta = 1.235593605, link = log)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4084  -0.9827  -0.2823   0.3482   3.0269
## Coefficients:
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  0.201812   0.317908  0.635   0.52555
## hosp        0.226922   0.032299  7.026  2.13e-12 ***
## healthpoor  0.198313   0.079353  2.499  0.01245 *
## healthexcellent -0.290092  0.093235 -3.111  0.00186 **
## numchron     0.171727   0.018834  9.118  < 2e-16 ***
## age          0.075012   0.040340  1.859   0.06296 .
## married      0.166799   0.060681  2.749   0.00598 **
## school       0.030996   0.006335  4.893   9.92e-07 ***
## ---
## Signif. codes:  < 0.001 ***  0.001 '**'  0.01 '*'  0.05 '.'  1 '
## (Dispersion parameter for Negative Binomial(1.2356) family taken to be 1)
## Null deviance: 2293.3 on 1760 degrees of freedom
## Residual deviance: 2040.5 on 1753 degrees of freedom
## AIC: 9291.5
## Number of Fisher Scoring iterations: 1
##
## Theta:  1.2356
## Std. Err.:  0.0581
## 2 x log-likelihood:  -9273.4800
```

In this model the standard errors are slightly lower with the result that "healthexcellent" and "married" are now significant. (level=0.05)

In count data there are often much more zeros than expected. Therefore one can fit a "zero-inflated" model using the pscl package. In the first "zero-inflated"
model one assumes that the occurrence of zeros does depend on covariates:

```r
library(pscl)
```

```r
med4 <- zeroinfl(ofp ~ hosp + healthpoor + healthexcellent + numchron + age + married + school | 1, data = medcare[!male == 1 & ofp <= 30,])
summary(med4)
```

```
## Call:
## zeroinfl(formula = ofp ~ hosp + healthpoor + healthexcellent + numchron +
##     age + married + school | 1, data = medcare[!male == 1 & ofp <= 30,]
##
## Pearson residuals:
##    Min  1Q Median  3Q Max
## -1.73 1.25 -0.37 0.63 7.44
##
## Count model coefficients (poisson with log link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.1855    0.1452   8.17 3.18e-16 ***
## hosp            0.1357    0.0107  12.72 < 2e-16 ***
## healthpoor      0.1524    0.0319   4.77 1.87e-06 ***
## healthexcellent -0.2206    0.0500  -4.41 1.04e-05 ***
## numchron        0.1024    0.0080  12.80 < 2e-16 ***
## age             0.0249    0.0181   1.38 0.177
## married         0.0239    0.0286   0.84 0.403
## school          0.0158    0.0029   5.34 9.15e-08 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.52       0.0636  -23.85 <2e-16 ***
##
## Number of iterations in BFGS optimization: 12
## Log-likelihood: -5577 on 9 Df
```

In the second "zero-inflated" model the occurrence of zeros can depend on covariates:
```r
med5 <- zeroInfl(ofp ~ hosp + healthpoor + healthexcellent + numchron + age + married + school, 
                  data = medcare[medcare$male == 1 & ofp <= 30, ])
summary(med5)
```

```
## Call:
## zeroInfl(formula = ofp ~ hosp + healthpoor + healthexcellent + numchron + 
##           age + married + school, data = medcare[medcare$male == 1 & ofp <= 30, ])
## Pearson residuals:
##          Min 1Q Median 3Q Max
## -3.5146 -1.0496 -0.4430 0.6023 7.9454
## Count model coefficients (poisson with log link):
##            Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 1.22709    0.14415   8.513   < 2e-16 ***
## hosp        0.13549    0.01069  12.676   < 2e-16 ***
## healthpoor  0.15193    0.03195   4.755    1.98e-06 ***
## healthexcellent -0.20314   0.04859  -4.181    2.90e-05 ***
## numchron    0.10045    0.00797  12.604   < 2e-16 ***
## age         0.02212    0.01800   1.229      0.219
## married     0.01771    0.02825   0.627      0.531
## school      0.01485    0.00292   5.087    3.64e-07 ***
## Zero-inflation model coefficients (binomial with logit link):
##               Estimate Std. Error z value Pr(>|z|)  
## (Intercept)   3.13376    0.88944  3.523   0.000426 ***
## hosp         -0.60179    0.15686  -3.836   0.000125 ***
## healthpoor   -0.21235    0.24601  -0.863     0.388048
## healthexcellent 0.26134    0.21546   1.213     0.225149
## numchron    -0.47280    0.06538  -7.231    4.78e-13 ***
## age         -0.34563    0.11432  -3.023    0.002500 **
## married     -0.69907    0.14796  -4.725    2.31e-06 ***
## school     -0.09232    0.01674  -5.515    3.50e-08 ***
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Number of iterations in BFGS optimization: 19
## Log-likelihood: -5491 on 16 Df
```

An alternative to "zero-inflation" is the "zero-hurdle" model. In the following similar models as above are fitted.

```r
med6 <- hurdle(ofp ~ hosp + healthpoor + healthexcellent + numchron + age + married + school | 1, 
                data = medcare[medcare$male == 1 & ofp <= 30, ])
summary(med6)
```

```
## Call:
## hurdle(formula = ofp ~ hosp + healthpoor + healthexcellent + numchron + 
##         age + married + school | 1, data = medcare[medcare$male == 1 & ofp <= 30, ])
## Log-likelihood: -5491 on 16 Df
```
```r
### age + married + school | 1, data = medcare[male == 1 & ofp <= 30,]
###
### Count model coefficients (truncated poisson with log link):
### Estimate Std. Error z value Pr(>|z|)
### (Intercept) 1.228410 0.144000 8.531 < 2e-16 ***
### hosp 0.135443 0.010691 12.669 < 2e-16 ***
### healthpoor 0.152058 0.031945 4.760 1.94e-06 ***
### healthexcellent -0.204398 0.048755 -4.192 2.76e-05 ***
### numchron 0.100331 0.007964 12.599 < 2e-16 ***
### age 0.022058 0.017985 1.226 0.220
### married 0.017420 0.028232 0.617 0.537
### school 0.014812 0.002919 5.075 3.88e-07 ***
###
### Zero hurdle model coefficients (binomial with logit link):
### Estimate Std. Error z value Pr(>|z|)
### (Intercept) 1.47077 0.06114 24.06 <2e-16 ***
### ---
### Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
###
### Number of iterations in BFGS optimization: 14
### Log-likelihood: -5582 on 9 Df
```

```r
med7=hurdle(ofp ~ hosp+healthpoor+healthexcellent+numchron+age+married+school,
data=medcare[male==1 & ofp<=30,])
summary(med7)
```

```r
### Call:
### hurdle(formula = ofp ~ hosp + healthpoor + healthexcellent + numchron +
### age + married + school, data = medcare[male == 1 & ofp <= 30,])
###
### Count model coefficients (truncated poisson with log link):
### Estimate Std. Error z value Pr(>|z|)
### (Intercept) 1.228410 0.144000 8.531 < 2e-16 ***
### hosp 0.135443 0.010691 12.669 < 2e-16 ***
### healthpoor 0.152058 0.031945 4.760 1.94e-06 ***
### healthexcellent -0.204398 0.048755 -4.192 2.76e-05 ***
### numchron 0.100331 0.007964 12.599 < 2e-16 ***
### age 0.022058 0.017985 1.226 0.220
### married 0.017420 0.028232 0.617 0.537
### school 0.014812 0.002919 5.075 3.88e-07 ***
```
## Zero hurdle model coefficients (binomial with logit link):

| Estimate  | Std. Error | z value | Pr(>|z|) | Signif. codes |
|-----------|------------|---------|----------|---------------|
| (Intercept) | -3.14201  | 0.87104 | -3.607   | 0.00031 ***   |
| hosp      | 0.60986    | 0.15535 | 3.926    | 8.65e-05 ***  |
| healthpoor | -0.20092  | 0.24410 | -0.823   | 0.41043       |
| healthexcellent | -0.28448 | 0.20846 | -1.365   | 0.17236       |
| numchron  | 0.47781    | 0.06438 | 7.422    | 1.15e-13 ***  |
| age       | 0.34266    | 0.11187 | 3.063    | 0.00219 **    |
| married   | 0.69079    | 0.14560 | 4.745    | 2.09e-06 ***  |
| school    | 0.09278    | 0.01642 | 5.651    | 1.60e-08 ***  |

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 14

Log-likelihood: -5491 on 16 Df