MAST90104: Introduction to Statistical Learning

Week 9 Lab and Workshop

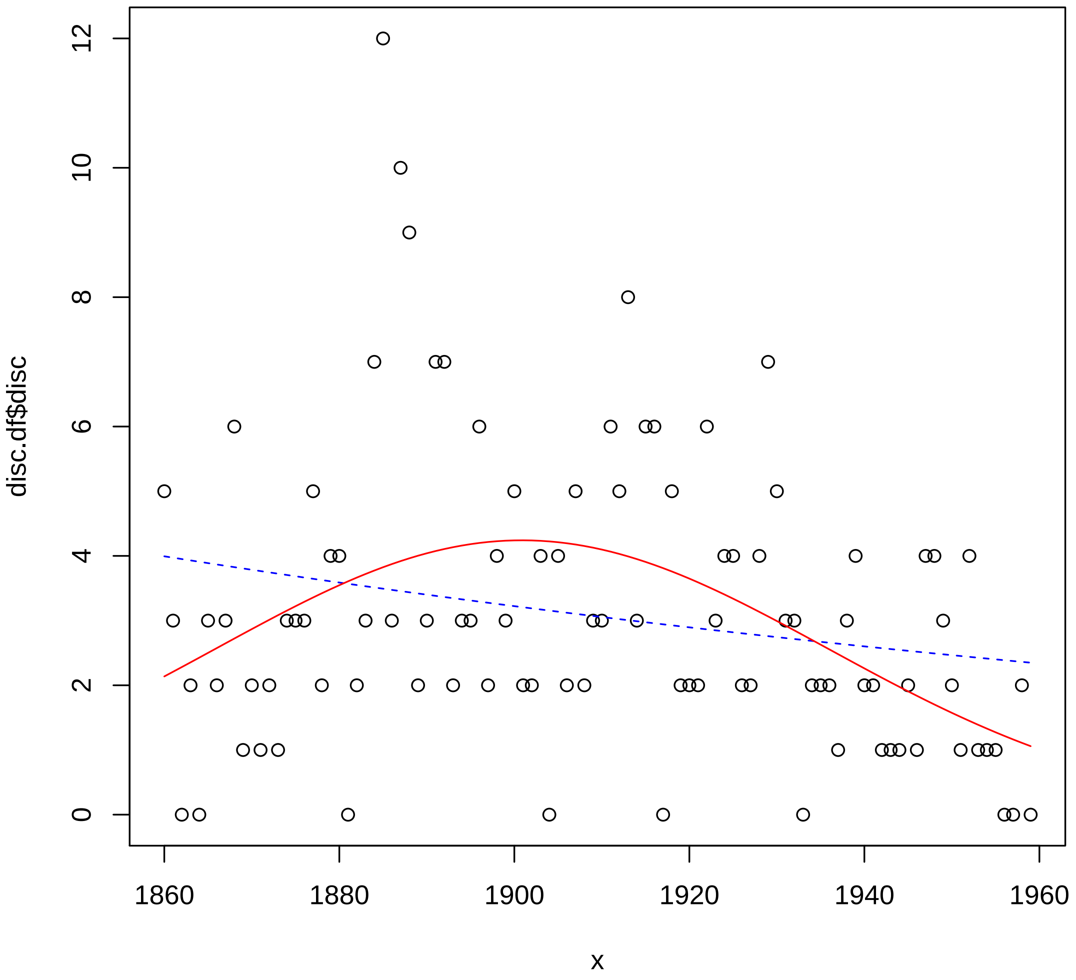
1. The dataset `discoveries` lists the number of great scientific discoveries for the years 1860 to 1959, as chosen by “The World Almanac and Book of Facts”, 1975 Edition. Has the discovery rate remained constant over time?

   To answer this question, fit a poisson regression model with a log link, and use the deviance to compare a null model with models including the year and year squared as predictors.

   Solution First we fit two models, the first including the year and the second the year and the year squared. The plot gives the fitted rates in each case.

```
data(discoveries)
disc.df <- data.frame(year=1860:1959, disc=discoveries)
model1 <- glm(disc ~ year, family=poisson, disc.df)
summary(model1)

##
## Call:
## glm(formula = disc ~ year, family = poisson, data = disc.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8112  -0.9482  -0.3533   0.6637   3.5504
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.354807   3.775677   3.007  0.00264 **
## year        -0.005360   0.001982  -2.705  0.00683 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 164.68  on 99  degrees of freedom
## Residual deviance: 157.32  on 98  degrees of freedom
## AIC: 430.32
##
## Number of Fisher Scoring iterations: 5

model2 <- glm(disc ~ year + I(year^2), family=poisson, disc.df)
summary(model2)

##
## Call:
## glm(formula = disc ~ year + I(year^2), family = poisson, data = disc.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9066  -0.8397  -0.2544   0.4776   3.3303
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.482e+03  3.163e+02  -4.685 2.79e-06 ***
## year         1.561e+00  3.318e-01   4.705 2.54e-06 ***
## I(year^2)   -4.106e-04  8.699e-05  -4.720 2.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 164.68  on 99  degrees of freedom
## Residual deviance: 132.84  on 97  degrees of freedom
## AIC: 407.85
##
## Number of Fisher Scoring iterations: 5
```

```
x <- disc.df$year
plot(x, disc.df$disc)
beta1 <- model1$coefficients
lines(x, exp(beta1[1] + beta1[2]*x), col="blue", lty=2)
beta2 <- model2$coefficients
lines(x, exp(beta2[1] + beta2[2]*x + beta2[3]*x^2), col="red")
```



From the plot both year and year squared look significant, but we need to quantify this observation.

For a poisson model the deviance only looks χ^2 if the responses are large enough to look vaguely normal, which they are not in this case. None-the-less, we can use deviance differences to perform likelihood ratio tests. From the above, the null model has deviance 164.68, the model with just year has deviance 157.32, and the model with year and year squared has deviance 132.84. We test the significance of adding year and then year squared:

```
pchisq(164.68-157.32, 1, lower.tail=FALSE)

## [1] 0.006669079

pchisq(157.32-132.84, 1, lower.tail=FALSE)

## [1] 7.508521e-07
```

There is strong evidence that year improves the model, and very strong evidence that year squared has something to add. We conclude that there is strong evidence that the discovery rate has changed over time.

2. The `ships` dataset from the `MASS` package gives the number of damage incidents and aggregate months of service for different types of ships broken down by year of construction and period of operation. Load the dataset using the commands `library(MASS)` then `data(ships)`.

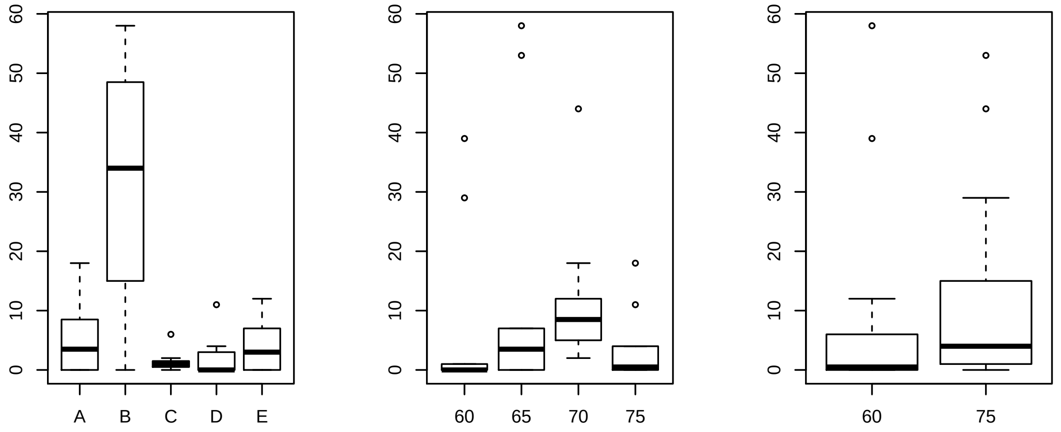
   Develop a model for the rate of incidents (i.e. a poisson regression model with log link), describing the effect of the important predictors.

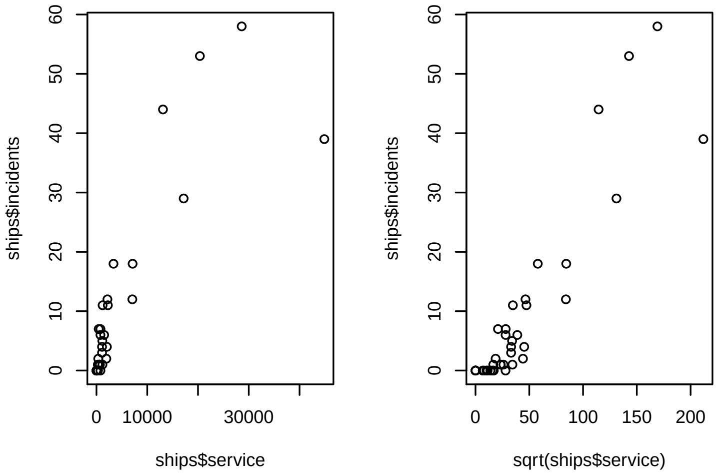
   Solution After loading and inspecting the data, it seems that `year` and `period` are really ordered factors rather than numerical predictors, so we alter these variables appropriately.

```
library(MASS)
data(ships)
ships$year <- factor(ships$year, levels=seq(60, 75, 5), ordered=TRUE)
ships$period <- factor(ships$period)
```

   Next we explore the relations between the variables. All the variables look important, and we note that applying a square root transform to `service` improves the relation between `service` and `incidents`

```
par(mfrow=c(2,3))
plot(ships$type, ships$incidents)
plot(ships$year, ships$incidents)
plot(ships$period, ships$incidents)
plot(ships$service, ships$incidents)
plot(sqrt(ships$service), ships$incidents)
par(mfrow=c(1,1))
```





We can fit now a log-poisson model. From the Wald tests each variable looks significant. We could confirm this using likelihood ratio tests based on the deviance.

```
ships$rootserv <- sqrt(ships$service)
model <- glm(incidents ~ type + year + period + rootserv, family=poisson, ships)
summary(model)

##
## Call:
## glm(formula = incidents ~ type + year + period + rootserv, family = poisson,
##     data = ships)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1509  -1.2833  -0.7905   0.2751   2.6875
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.207853   0.234122   0.888 0.374649
## typeB       -0.121206   0.250163  -0.485 0.628024
## typeC       -1.005644   0.329657  -3.051 0.002284 **
## typeD       -0.574643   0.289933  -1.982 0.047481 *
## typeE       -0.025521   0.236667  -0.108 0.914127
## year.L       0.654626   0.194109   3.372 0.000745 ***
## year.Q      -0.822592   0.122829  -6.697 2.13e-11 ***
## year.C      -0.128340   0.097295  -1.319 0.187142
## period75     0.726592   0.125831   5.774 7.73e-09 ***
## rootserv     0.021648   0.002202   9.830  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 730.253  on 39  degrees of freedom
## Residual deviance:  67.035  on 30  degrees of freedom
## AIC: 184.9
##
## Number of Fisher Scoring iterations: 5
```

Note that because `year` is ordered R has used linear, quadratic and cubic contrasts. You can see them exactly using `contrasts`

```
contrasts(ships$year)

##              .L   .Q         .C
## [1,] -0.6708204  0.5 -0.2236068
## [2,] -0.2236068 -0.5  0.6708204
## [3,]  0.2236068 -0.5 -0.6708204
## [4,]  0.6708204  0.5  0.2236068
```

Next we look for interactions.

```
model1 <- glm(incidents ~ type + year + period + rootserv + type:year, family=poisson, ships)
pchisq(deviance(model) - deviance(model1), df.residual(model) - df.residual(model1), lower.tail=FALSE)

## [1] 0.0001099668

model2 <- glm(incidents ~ type + year + period + rootserv + type:period, family=poisson, ships)
pchisq(deviance(model) - deviance(model2), df.residual(model) - df.residual(model2), lower.tail=FALSE)

## [1] 0.08820292

model3 <- glm(incidents ~ type + year + period + rootserv + type:rootserv, family=poisson, ships)
pchisq(deviance(model) - deviance(model3), df.residual(model) - df.residual(model3), lower.tail=FALSE)

## [1] 0.003187932

model4 <- glm(incidents ~ type + year + period + rootserv + year:period, family=poisson, ships)
pchisq(deviance(model) - deviance(model4), df.residual(model) - df.residual(model4), lower.tail=FALSE)

## [1] 0.0001018208

model5 <- glm(incidents ~ type + year + period + rootserv + year:rootserv, family=poisson, ships)
pchisq(deviance(model) - deviance(model5), df.residual(model) - df.residual(model5), lower.tail=FALSE)

## [1] 0.0153112

model6 <- glm(incidents ~ type + year + period + rootserv + period:rootserv, family=poisson, ships)
pchisq(deviance(model) - deviance(model6), df.residual(model) - df.residual(model6), lower.tail=FALSE)
```

```
## [1] 0.4123239
```

```
model7 <- glm(incidents ~ type + year + period + rootserv + type:year + period:year, family=poisson, ships
pchisq(deviance(model1) - deviance(model7), df.residual(model1) - df.residual(model7), lower.tail=FALSE)
```

```
## [1] 0.0005265296
```

```
model8 <- glm(incidents ~ type + year + period + rootserv + type:year + period:year + type:rootserv, famil
pchisq(deviance(model7) - deviance(model8), df.residual(model7) - df.residual(model8), lower.tail=FALSE)
```

```
## [1] 0.07904069
```

```
model9 <- glm(incidents ~ type + year + period + rootserv + type:year + period:year + year:rootserv, famil
pchisq(deviance(model7) - deviance(model9), df.residual(model7) - df.residual(model9), lower.tail=FALSE)
```

```
## [1] 0.8730395
```

```
summary(model7)
```

```
##
## Call:
## glm(formula = incidents ~ type + year + period + rootserv + type:year +
##     period:year, family = poisson, data = ships)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.80944  -0.00785  -0.00005   0.00847   2.06533
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -9.556e+00  3.042e+03  -0.003    0.997
## typeB            5.649e+00  2.668e+03   0.002    0.998
## typeC            3.815e+00  2.668e+03   0.001    0.999
## typeD           -5.655e+00  4.666e+03  -0.001    0.999
## typeE           -3.376e-01  3.779e+03   0.000    1.000
## year.L           1.206e+00  8.164e+03   0.000    1.000
## year.Q          -2.107e+01  6.085e+03  -0.003    0.997
## year.C          -1.162e-01  2.721e+03   0.000    1.000
## period75         5.714e+00  1.463e+03   0.004    0.997
## rootserv         1.426e-02  1.341e-02   1.063    0.288
## typeB:year.L    -1.467e+01  7.158e+03  -0.002    0.998
## typeC:year.L    -1.451e+01  7.158e+03  -0.002    0.998
## typeD:year.L     4.069e+00  1.043e+04   0.000    1.000
## typeE:year.L    -1.669e+00  1.014e+04   0.000    1.000
## typeB:year.Q     1.019e+01  5.335e+03   0.002    0.998
## typeC:year.Q     1.040e+01  5.335e+03   0.002    0.998
## typeD:year.Q     1.028e+01  9.333e+03   0.001    0.999
## typeE:year.Q    -1.288e+00  7.559e+03   0.000    1.000
## typeB:year.C    -4.192e+00  2.386e+03  -0.002    0.999
## typeC:year.C    -5.540e+00  2.386e+03  -0.002    0.998
## typeD:year.C    -1.430e+01  8.091e+03  -0.002    0.999
## typeE:year.C     2.112e-01  3.380e+03   0.000    1.000
## year.L:period75  1.364e+01  3.926e+03   0.003    0.997
## year.Q:period75  1.044e+01  2.926e+03   0.004    0.997
## year.C:period75  4.229e+00  1.309e+03   0.003    0.997
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 730.25  on 39  degrees of freedom
## Residual deviance:  10.53  on 15  degrees of freedom
## AIC: 158.4
##
## Number of Fisher Scoring iterations: 18
```

Curiously, although the `type:year` and `period:year` interactions are significant, none of the Wald tests are significant in the model with interactions. This suggests dependency between our predictors. We look for a more parsimoneous model using `step`.

```
model10 <- step(model7)

## Start:  AIC=158.4
## incidents ~ type + year + period + rootserv + type:year + period:year
##
##               Df Deviance    AIC
## - rootserv     1   11.694 157.56
## <none>             10.530 158.40
## - type:year   12   45.965 169.83
## - year:period  3   28.151 170.02
##
## Step:  AIC=157.56
## incidents ~ type + year + period + type:year + year:period
##
##               Df Deviance    AIC
## <none>             11.694 157.56
## - year:period  3   72.163 212.03
## - type:year   12  123.483 245.35

summary(model10)

##
## Call:
## glm(formula = incidents ~ type + year + period + type:year +
##     year:period, family = poisson, data = ships)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86294  -0.03467  -0.00005   0.03221   2.18897
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -9.0814  3030.6260  -0.003    0.998
## typeB                6.9806  2733.6363   0.003    0.998
## typeC                3.7125  2733.6364   0.001    0.999
## typeD               -5.7949  4742.5185  -0.001    0.999
## typeE               -0.5124  3865.9456   0.000    1.000
## year.L               0.7141  8132.0229   0.000    1.000
## year.Q             -21.2793  6061.2520  -0.004    0.997
## year.C              -0.1840  2710.6743   0.000    1.000
## period75             5.5950  1308.4060   0.004    0.997
## typeB:year.L       -16.0763  7335.1160  -0.002    0.998
## typeC:year.L       -15.0317  7335.1161  -0.002    0.998
## typeD:year.L         4.0042 10660.4227   0.000    1.000
## typeE:year.L        -1.8407 10373.4206   0.000    1.000
## typeB:year.Q        10.3502  5467.2727   0.002    0.998
## typeC:year.Q        10.4696  5467.2728   0.002    0.998
## typeD:year.Q        10.5783  9485.0369   0.001    0.999
## typeE:year.Q        -1.3731  7731.8912   0.000    1.000
## typeB:year.C        -3.9304  2445.0387  -0.002    0.999
## typeC:year.C        -5.6421  2445.0388  -0.002    0.998
## typeD:year.C       -14.2745  8141.6974  -0.002    0.999
## typeE:year.C         0.1601  3457.8069   0.000    1.000
## year.L:period75     14.9779  3510.8218   0.004    0.997
## year.Q:period75     10.2033  2616.8120   0.004    0.997
## year.C:period75      4.1899  1170.2739   0.004    0.997
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 730.253  on 39  degrees of freedom
## Residual deviance:  11.694  on 16  degrees of freedom
## AIC: 157.56
##
## Number of Fisher Scoring iterations: 18
```

We see that, given the `type:year` and `period:year` interactions, `rootserv` is no longer significant. Formally, the reason for this is that `rootserv` itself can be predicted using `type`, `year`, `period`, `type:year` and `period:year`, so it is no longer needed when it comes to predicting `incidents`. Having said that, there is a clear scientific reason for wanting `rootserv` in the model, so given that the AIC for `model10` is not much smaller than that for `model7`, I would be inclined to keep it.

The fact that the individual parameters in `model10` are all close to zero is not necessarily a problem, but does suggest that some of these levels could be grouped. Testing that two levels of a factor are the same is not as easy for a glm as for a linear model, but can still be done indirectly using likelihood ratio tests. What we have to do is fit a model where the levels are combined, and then see if it performs significantly worse.

3. The `infert` dataset from the `survival` package presents data from a study of infertility after spontaneous and induced abortion. Using a logistic regression model, analyse and report on the factors related to infertility based on this data. (Don't use the factor stratum, as it is confounded with the other predictors.)

   Solution The response is `case`, with 1 indicating infertility and 0 fertility. The data comes from a case-control study, the aim of which was to estimate the effect of the number of prior induced and spontaneous abortions on the probability of becoming infertile. In the original study it was believed that education, age and parity (something numeric, whatever it is) were confounding variables, so the cases were separated into 83 strata based on these variables, and two controls were recruited from each stratum. (One control from one of the strata was subsequently omitted from the dataset, for reasons unexplained.)

   Because of how the data were collected, the observations are *not* independent, so a logistic regression model is not actually appropriate. None-the-less we will carry on as if it is, and next week will analyse the data using a conditional logistic regression

```
library(survival)
data(infert)
model1 <- glm(case ~ age+parity+education+spontaneous+induced,
              data = infert, family = binomial())
summary(model1)
```

```
##
## Call:
## glm(formula = case ~ age + parity + education + spontaneous +
##     induced, family = binomial(), data = infert)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7603  -0.8162  -0.4956   0.8349   2.6536
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.14924    1.41220  -0.814   0.4158
## age               0.03958    0.03120   1.269   0.2046
## parity           -0.82828    0.19649  -4.215 2.49e-05 ***
## education6-11yrs -1.04424    0.79255  -1.318   0.1876
## education12+ yrs -1.40321    0.83416  -1.682   0.0925 .
## spontaneous       2.04591    0.31016   6.596 4.21e-11 ***
## induced           1.28876    0.30146   4.275 1.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.17  on 247  degrees of freedom
## Residual deviance: 257.80  on 241  degrees of freedom
## AIC: 271.8
##
## Number of Fisher Scoring iterations: 4
```

```
model2 <- glm(case ~ parity+education+spontaneous+induced,
              data = infert, family = binomial())
summary(model2)
```

```
##
## Call:
## glm(formula = case ~ parity + education + spontaneous + induced,
##     family = binomial(), data = infert)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8372  -0.8194  -0.4737   0.8909   2.5822
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.2646     0.8669   0.305   0.7602
## parity            -0.8043     0.1964  -4.095 4.22e-05 ***
## education6-11yrs  -1.1494     0.7868  -1.461   0.1441
## education12+ yrs  -1.6123     0.8185  -1.970   0.0489 *
## spontaneous        1.9882     0.3048   6.523 6.90e-11 ***
## induced            1.2329     0.2986   4.128 3.66e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.17  on 247  degrees of freedom
## Residual deviance: 259.43  on 242  degrees of freedom
## AIC: 271.43
##
## Number of Fisher Scoring iterations: 4
```

```
pchisq(deviance(model2) - deviance(model1), 1, lower.tail=FALSE)
```

```
## [1] 0.2019603
```

Continuing in this manner we find that all the remaining variables are significant at the 5% level (using the χ^2 test).

4. The dataset `africa` from the `faraway` package gives information about the number of military coups in sub-saharan Africa and various political and geographical information.

   Use the AIC to choose a parsimonious generalised linear model for the number of coups. Give an interpretation of the effect on the response of the variables you include in your model.

   Solution Firstly we load the data and remove observations with missing variables. The variable `pollib` is converted to a factor.

```
library(faraway)

##
## Attaching package:  'faraway'
## The following object is masked from 'package:survival':
##
##     rats

data(africa)
africa <- africa[complete.cases(africa),]
africa$pollib <- factor(africa$pollib, levels=0:2)
```

It is odd that the number of years since liberation is not included as a variable, but we carry on regardless (see the help function `?africa` for details). Fitting an additive model and applying `step` leaves the variables `oligarchy`, `pollib` and `parties`.

```
model1 <- glm(miltcoup ~ ., family=poisson, africa)
model1a <- step(model1, scope=~.)

## Start:  AIC=113.06
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     numelec + numregim
##
##             Df Deviance    AIC
## - numelec    1   28.430 111.24
## - numregim   1   29.059 111.87
## - size       1   29.238 112.05
## <none>           28.249 113.06
## - pctvote    1   30.572 113.38
## - popn       1   30.601 113.41
## - oligarchy  1   32.354 115.16
## - pollib     2   35.581 116.39
## - parties    1   35.311 118.12
##
## Step:  AIC=111.24
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     numregim
##
##             Df Deviance    AIC
## - numregim   1   29.081 109.89
## - size       1   29.452 110.26
## <none>           28.430 111.24
## - pctvote    1   30.590 111.40
## - popn       1   30.605 111.41
## + numelec    1   28.249 113.06
## - pollib     2   36.872 115.68
## - parties    1   35.773 116.58
## - oligarchy  1   36.595 117.40
##
## Step:  AIC=109.89
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size
##
##             Df Deviance    AIC
## - size       1   30.040 108.85
## - popn       1   30.614 109.42
## <none>           29.081 109.89
## - pctvote    1   31.599 110.41
## + numregim   1   28.430 111.24
## + numelec    1   29.059 111.87
## - pollib     2   37.830 114.64
## - parties    1   36.304 115.11
## - oligarchy  1   40.291 119.10
##
## Step:  AIC=108.85
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn
##
##             Df Deviance    AIC
## - popn       1   31.069 107.88
## <none>           30.040 108.85
## - pctvote    1   32.241 109.05
## + size       1   29.081 109.89
## + numregim   1   29.452 110.26
## + numelec    1   30.002 110.81
## - pollib     2   38.022 112.83
## - parties    1   37.547 114.36
## - oligarchy  1   40.468 117.28
##
## Step:  AIC=107.88
## miltcoup ~ oligarchy + pollib + parties + pctvote
##
##             Df Deviance    AIC
## - pctvote    1   32.822 107.63
## <none>           31.069 107.88
## + popn       1   30.040 108.85
## + size       1   30.614 109.42
## + numregim   1   31.044 109.85
## + numelec    1   31.069 109.88
## - parties    1   37.547 112.36
## - pollib     2   39.762 112.57
## - oligarchy  1   48.196 123.00
##
## Step:  AIC=107.63
## miltcoup ~ oligarchy + pollib + parties
##
##             Df Deviance    AIC
## <none>           32.822 107.63
## + pctvote    1   31.069 107.88
## + popn       1   32.241 109.05
## + size       1   32.533 109.34
## + numelec    1   32.594 109.40
## + numregim   1   32.643 109.45
## - pollib     2   40.025 110.83
## - parties    1   38.162 110.97
## - oligarchy  1   49.458 122.27
```

```
summary(model1a)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties, family = poisson,
##     data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3609  -1.0407  -0.3153   0.6145   1.7536
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.207981   0.445679   0.467   0.6407
## oligarchy    0.091466   0.022563   4.054 5.04e-05 ***
## pollib1     -0.495414   0.475645  -1.042   0.2976
## pollib2     -1.112086   0.459492  -2.420   0.0155 *
## parties      0.022358   0.009098   2.458   0.0140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 32.822  on 31  degrees of freedom
## AIC: 107.63
##
## Number of Fisher Scoring iterations: 5
```

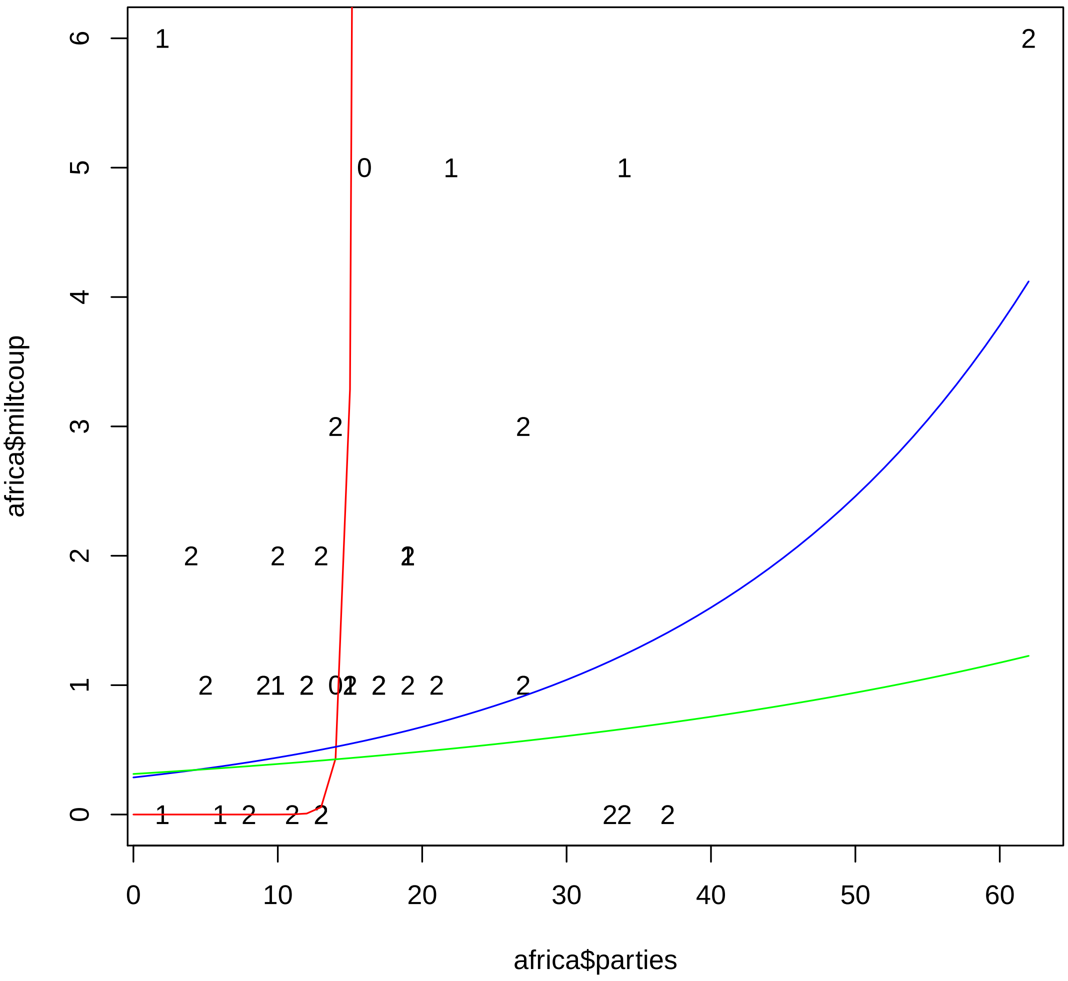
One can imagine `pollib` and `partied` interacting, so we repeat the analysis including this interaction. The interaction is significant, and when it is included a number of other variables become significant.

```
model2 <- glm(miltcoup ~ . + pollib:parties, family=poisson, africa)
model2a <- step(model2, scope=~.)

## Start:  AIC=107.3
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     numelec + numregim + pollib:parties
##
##                  Df Deviance    AIC
## - numelec         1   18.505 105.31
## - numregim        1   19.158 105.97
## <none>                18.489 107.30
## - pctvote         1   20.727 107.53
## - popn            1   22.117 108.93
## - oligarchy       1   24.907 111.72
## - size            1   26.191 113.00
## - pollib:parties  2   28.249 113.06
##
## Step:  AIC=105.31
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     numregim + pollib:parties
##
##                  Df Deviance    AIC
## - numregim        1   19.186 104.00
## <none>                18.505 105.31
## - pctvote         1   20.798 105.61
## - popn            1   22.429 107.24
## + numelec         1   18.489 107.30
## - pollib:parties  2   28.430 111.24
## - size            1   26.519 111.33
## - oligarchy       1   29.022 113.83
##
## Step:  AIC=104
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     pollib:parties
##
##                  Df Deviance    AIC
## <none>                19.186 104.00
## - pctvote         1   21.908 104.72
## - popn            1   22.435 105.24
## + numregim        1   18.505 105.31
## + numelec         1   19.158 105.97
## - size            1   27.030 109.84
## - pollib:parties  2   29.081 109.89
## - oligarchy       1   33.605 116.41

summary(model2a)
```

```
## 
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote + 
##     popn + size + pollib:parties, family = poisson, data = africa)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.1966  -0.7277  -0.1284   0.2610   1.6898  
## 
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)    
## (Intercept)     -2.923e+01  1.144e+01  -2.556 0.010595 *  
## oligarchy        1.083e-01  2.946e-02   3.676 0.000237 ***
## pollib1          2.798e+01  1.149e+01   2.434 0.014919 *  
## pollib2          2.807e+01  1.144e+01   2.453 0.014154 *  
## parties          2.028e+00  7.657e-01   2.648 0.008088 ** 
## pctvote          1.661e-02  1.010e-02   1.645 0.099979 .  
## popn             1.392e-02  7.911e-03   1.759 0.078501 .  
## size            -1.207e-03  4.679e-04  -2.579 0.009895 ** 
## pollib1:parties -1.985e+00  7.701e-01  -2.578 0.009937 ** 
## pollib2:parties -2.006e+00  7.657e-01  -2.620 0.008783 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 19.186  on 26  degrees of freedom
## AIC: 104
## 
## Number of Fisher Scoring iterations: 5
```

The final model has positive coefficients for `pollib1` and `pollib2`, which curiously seems to suggest that more liberal countries have more coups. However, to make sence the `pollib` varaible needs to be interpreted together with its interaction with `parties`. We plot the contribution to the overall rate of coups from these two variables. Note the exponential transform because we used a log link.

```r
plot(africa$parties, africa$miltcoup, type="n")
text(africa$parties, africa$miltcoup, africa$pollib)
x <- 0:62
y0 <- exp(-29.23 + 2.028*x)
y1 <- exp(-29.23+27.98 + (2.028-1.985)*x)
y2 <- exp(-29.23+28.07 + (2.028-2.006)*x)
lines(x, y0, col="red")
lines(x, y1, col="blue")
lines(x, y2, col="green")
```



This plot shows that the strange numbers are the result of the model fitting the cases where `pollib` is zero rather too closely. It is not plausable that for countries with no liberties the rate of coups should suddenly skyrocket as soon as you have 13 political parties. The root cause of the problem from the modelling point of view is that we only have two cases where `pollib` is zero. Accordingly we combined levels 0 and 1 of `pollib` and repeated the analysis.

```
x <- africa$pollib == 0
africa$pollib[x] <- 1
model3 <- glm(miltcoup ~ . + pollib:parties, family=poisson, africa)
model3a <- step(model3, scope=~.)

## Start:  AIC=114.6
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     numelec + numregim + pollib:parties
##
##                  Df Deviance    AIC
## - numelec         1   29.809 112.62
## - size            1   29.823 112.63
## - pctvote         1   30.297 113.11
## - numregim        1   30.412 113.22
## - pollib:parties  1   30.939 113.75
## <none>                29.789 114.60
## - popn            1   31.923 114.73
## - oligarchy       1   34.331 117.14
##
## Step:  AIC=112.62
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##     numregim + pollib:parties
##
##                  Df Deviance    AIC
## - size            1   29.830 110.64
## - pctvote         1   30.366 111.17
## - numregim        1   30.582 111.39
## - pollib:parties  1   31.040 111.85
## <none>                29.809 112.62
## - popn            1   32.233 113.04
## + numelec         1   29.789 114.60
## - oligarchy       1   36.730 117.54
##
## Step:  AIC=110.64
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + numregim +
##     pollib:parties
##
##                  Df Deviance    AIC
## - pctvote         1   30.385 109.19
## - numregim        1   30.594 109.40
## - pollib:parties  1   31.088 109.90
## <none>                29.830 110.64
## - popn            1   32.238 111.05
## + size            1   29.809 112.62
## + numelec         1   29.823 112.63
## - oligarchy       1   36.735 115.54
##
## Step:  AIC=109.19
## miltcoup ~ oligarchy + pollib + parties + popn + numregim + pollib:parties
##
##                  Df Deviance    AIC
## - numregim        1   31.402 108.21
## - pollib:parties  1   32.359 109.17
## <none>                30.385 109.19
## - popn            1   33.028 109.84
## + pctvote         1   29.830 110.64
## + numelec         1   30.344 111.15
## + size            1   30.366 111.17
## - oligarchy       1   37.097 113.91
##
## Step:  AIC=108.21
## miltcoup ~ oligarchy + pollib + parties + popn + pollib:parties
##
##                  Df Deviance    AIC
## - popn            1   33.109 107.92
## - pollib:parties  1   33.333 108.14
## <none>                31.402 108.21
## + numregim        1   30.385 109.19
## + pctvote         1   30.594 109.40
## + numelec         1   31.126 109.94
## + size            1   31.393 110.20
## - oligarchy       1   41.673 116.48
##
## Step:  AIC=107.92
## miltcoup ~ oligarchy + pollib + parties + pollib:parties
##
##                  Df Deviance    AIC
## - pollib:parties  1   33.818 106.63
## <none>                33.109 107.92
## + popn           1   31.402 108.21
## + pctvote        1   32.269 109.08
## + numelec        1   32.550 109.36
## + numregim       1   33.028 109.84
## + size           1   33.090 109.90
## - oligarchy      1   49.565 122.37
##
## Step:  AIC=106.63
## miltcoup ~ oligarchy + pollib + parties
##
##                  Df Deviance    AIC
## <none>                33.818 106.63
## + pctvote        1   32.542 107.35
## + numelec        1   33.092 107.90
## + pollib:parties  1   33.109 107.92
## + popn           1   33.333 108.14
## + numregim       1   33.590 108.40
## + size           1   33.818 108.63
## - parties        1   39.338 110.15
## - pollib         1   40.025 110.83
## - oligarchy      1   49.733 120.54
```

```
summary(model3a)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties, family = poisson,
##     data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4012  -1.0593  -0.3945   0.5598   1.7182
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.153867   0.307817  -0.500   0.6172
## oligarchy    0.086951   0.021593   4.027 5.65e-05 ***
## pollib2     -0.717419   0.285632  -2.512   0.0120 *
## parties      0.022562   0.009038   2.496   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 33.818  on 32  degrees of freedom
## AIC: 106.63
##
## Number of Fisher Scoring iterations: 5
```

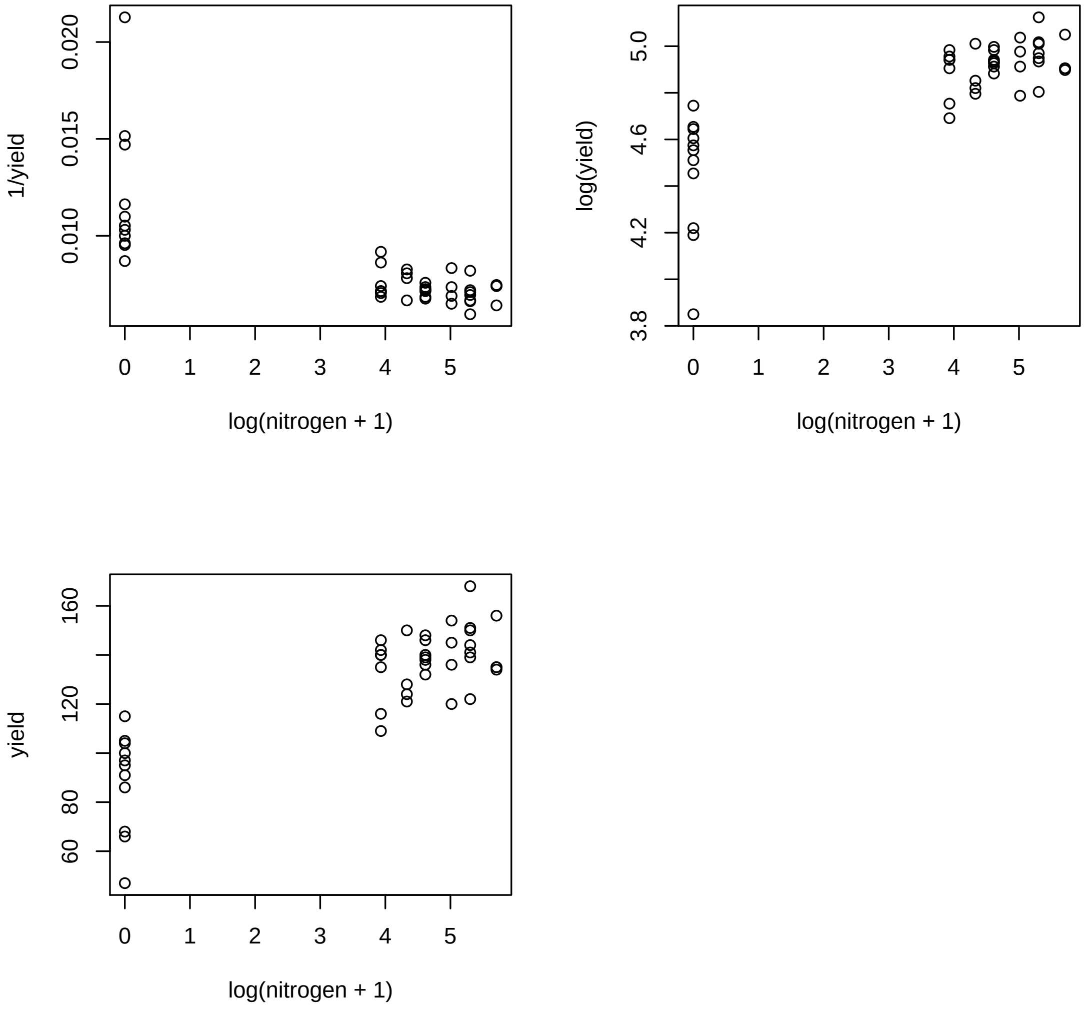
We are back where we started! The interaction between `pollib` and `parties` was just an artifact of the small number of observations of `pollib` at level 0. For the final model we see that each year of oligarchy increases the rate of coups by $e^{0.08695} = 1.0908$; full civil rights reduce the rate of coups by $e^{0.7174} = 2.0491$; and each additional political party increases the rate by $e^{0.02256} = 1.0228$.

5. The `cornnit` dataset in the `faraway` package contains data on the effect of nitrogen on the yield of corn. Fit a gamma regression to this data, using the `glm` command. You will need to pay attention to the choice of link function (inverse, identity or log), and consider transforming the predictor variable (your first step should be to plot the data).

   Solution: As suggested we plot the data first, using different link functions. It was found that taking a log transform of the nitrogen variable improves the linearity in all cases (note that we add a small constant before taking the log because nitrogren has zero values).

We suppose that the mean behaves like $g^{-1}(\eta)$, where in this case $\eta = \beta_0 + \beta_1 \log(1 + x)$ and x is the level of nitrogen. Thus a plot of $g(y)$ against $\log(1 + x)$ should look (vaguely) linear.

```r
library(faraway)
data(cornnit)
par(mfrow=c(2,2))
plot(1/yield ~ log(nitrogen+1), data=cornnit)
plot(log(yield) ~ log(nitrogen+1), data=cornnit)
plot(yield ~ log(nitrogen+1), data=cornnit)
```

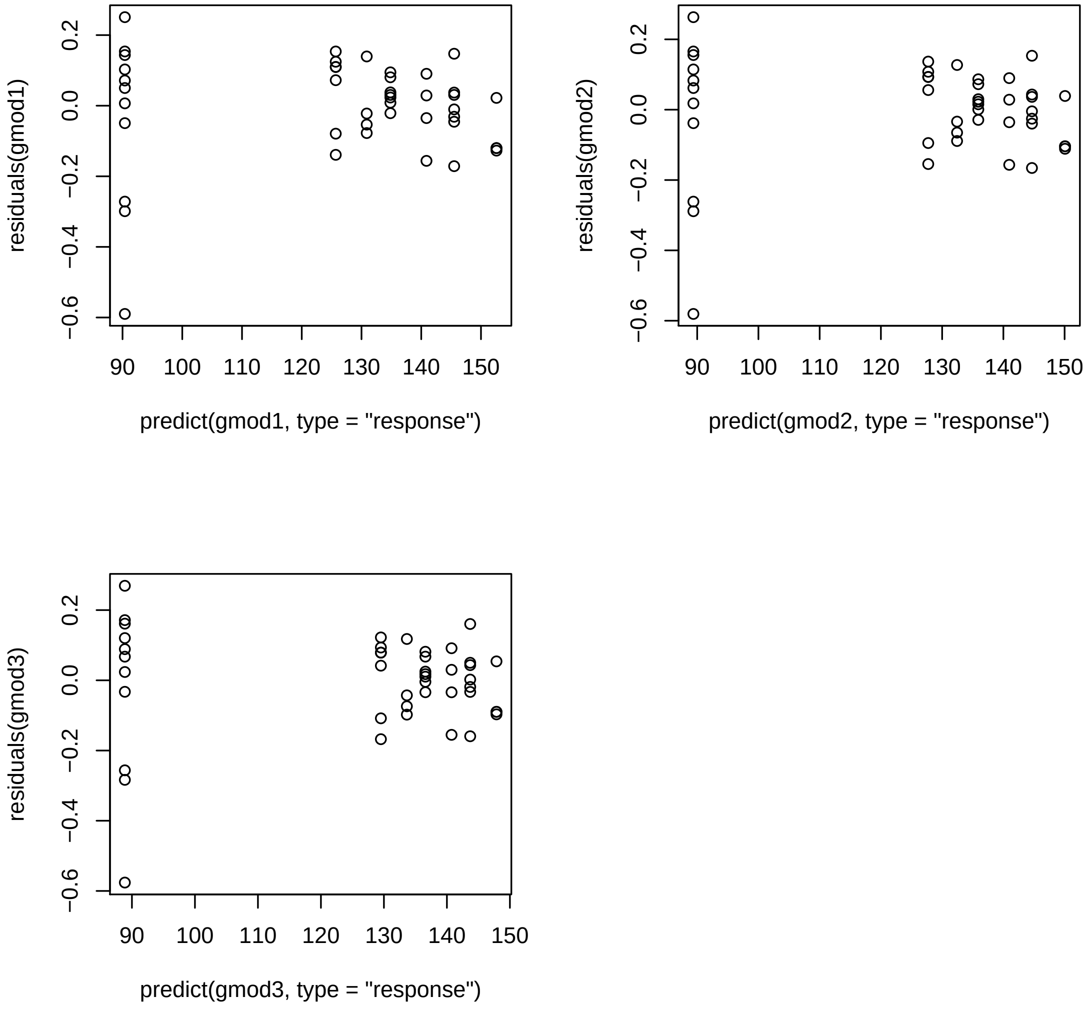


In all three plots there is an undesirable gap in the observed nitrogen values. We can reduce this a little by using the transform $\log(\text{nitrogen} + k)$ for larger k, but this impinges on the linearity.

In the first and second plots there is noticably more variance in $g(y)$ when nitrogen is zero. For a gamma model the variance should be proportional to the mean squared. Thus when the yield is larger we expect the data to be more variable, which is not what we see here. However, when we transform the responses we transform their variances as well as their means, and both the inverse and log links have larger slopes at small values, so this will be magnifying the variance when the yield is small. Unfortunately we can't really disentangle these two effects using these plots (but see the residual plots below).

Of the three I think the plot of yield against log(nitrogen + 1) looks most linear, but the other two are not unreasonable. Accordingly we will try all three link functions are compare the residuals.

```
par(mfrow=c(2,2))
gmod1 <- glm(yield ~ log(nitrogen+1), data=cornnit, family=Gamma(link="inverse"))
plot(predict(gmod1,type="response"), residuals(gmod1))
gmod2 <- glm(yield ~ log(nitrogen+1), data=cornnit, family=Gamma(link="log"))
plot(predict(gmod2,type="response"), residuals(gmod2))
gmod3 <- glm(yield ~ log(nitrogen+1), data=cornnit, family=Gamma(link="identity"))
plot(predict(gmod3,type="response"), residuals(gmod3))
```



There is not much difference between these plots. In all three cases there is slightly more variation when the fitted values are small, but for a gamma model we would expect the variance to grow as the fitted values got larger. Thus all three models are problematic, however if we look at the AIC for each model we see that it is smallest for the model with the identity link (just), so we will take this model from here on.

```
gmod1$aic

## [1] 383.7435
```

```
gmod2$aic

## [1] 382.4205

gmod3$aic

## [1] 381.7124
```

(a) Extract the Pearson residuals from the fitted model using the `residuals` function, then use them to estimate the dispersion parameter. Check that your answer agrees with the summary output from your model.
Solution: From the summary we see the dispersion parameter is estimated to be 0.01810, which we can reproduce using Pearson's chi-squared statistic. Note that the model has 42 d.f.

```
summary(gmod3)

## 
## Call:
## glm(formula = yield ~ log(nitrogen + 1), family = Gamma(link = "identity"), 
##     data = cornnit)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -0.57604  -0.07789   0.02067   0.07948   0.26927  
## 
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         88.875      3.571   24.89  < 2e-16 ***
## log(nitrogen + 1)   10.337      1.009   10.24 5.46e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Gamma family taken to be 0.01810187)
## 
##     Null deviance: 2.40614  on 43  degrees of freedom
## Residual deviance: 0.87603  on 42  degrees of freedom
## AIC: 381.71
## 
## Number of Fisher Scoring iterations: 4

(phihat <- sum(residuals(gmod3, "pearson")^2)/42)

## [1] 0.01810169
```

(b) Suppose your fitted model is `gmod`, then the command `anova(gmod, test="F")` will compare your model against the null model, using an F test. Using the deviances and dispersion estimates reported by `summary(gmod)`, check that the F statistic reported by the `anova` function is correct.
Solution:

```
anova(gmod3, test="F")

## Analysis of Deviance Table
## 
## Model: Gamma, link: identity
## 
## Response: yield
```

```
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev      F    Pr(>F)
## NULL                                 43    2.40614
## log(nitrogen + 1)  1   1.5301        42    0.87603 84.528 1.297e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

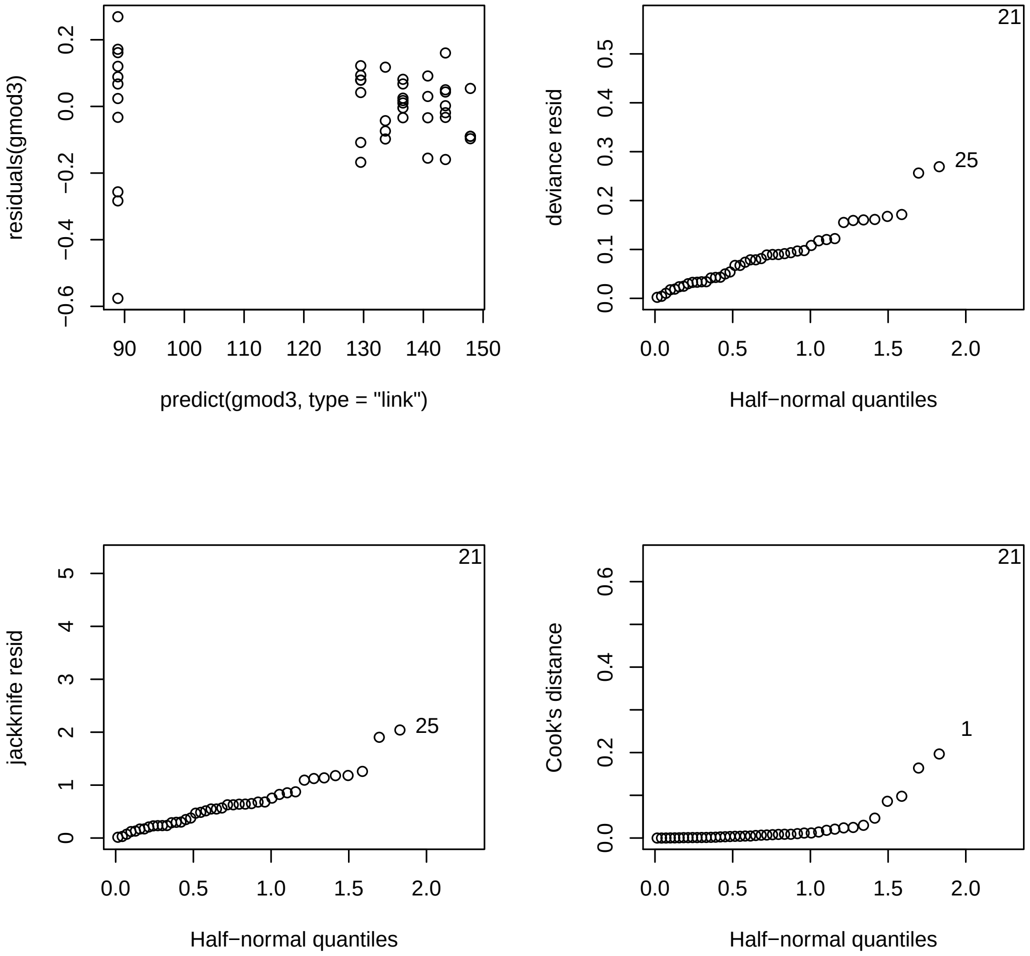
```
model_dev <- .87603
null_dev <- 2.40614
(F_statistic <- (null_dev - model_dev)/phihat)
```

```
## [1] 84.52857
```

(c) Now do some diagnostic plots. Can you identify a potential outlier?

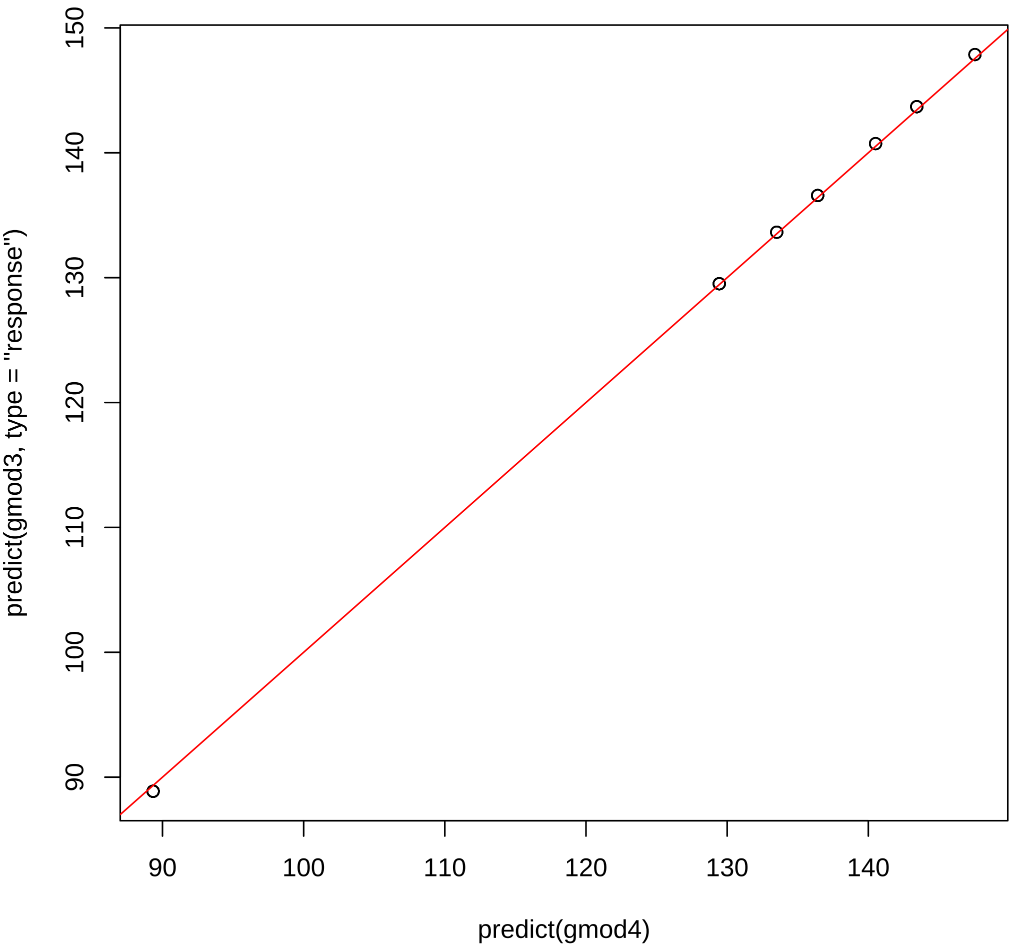
Solution: We have already observed more variation than we would like when the responses are small. It also looks like point 21 could be an outlier.

```
par(mfrow=c(2,2))
plot(predict(gmod3, type="link"), residuals(gmod3))
halfnorm(residuals(gmod3), ylab="deviance resid")
halfnorm(rstudent(gmod3), ylab="jackknife resid")
halfnorm(cooks.distance(gmod3), ylab="Cook's distance")
```



(d) Fit a linear model to the `cornnit` data.
Which do you prefer, the linear model or the gamma model, and why?
Solution: A gamma variable with a large mean looks a lot like a normal, so we expect a linear model fit to look a lot like our gamma model fit, and it does. We can see this by plotting the fitted values for the two models against each other, and seeing that the points lie very close to the diagonal.

```
gmod4 <- lm(yield ~ log(nitrogen+1), data=cornnit)
summary(gmod4)

##
## Call:
## lm(formula = yield ~ log(nitrogen + 1), data = cornnit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.335 -10.261   2.126  10.558  25.665
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         89.335      4.227   21.13  < 2e-16 ***
## log(nitrogen + 1)   10.201      1.017   10.03 1.03e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.34 on 42 degrees of freedom
## Multiple R-squared:  0.7055,Adjusted R-squared:  0.6985
## F-statistic: 100.6 on 1 and 42 DF,  p-value: 1.025e-12

par(mfrow=c(1,1))
plot(predict(gmod4), predict(gmod3, type="response"))
abline(0, 1, col="red")
```



However, for a linear model we expect the variance to stay fixed, rather than grow with the mean, and this is more in keeping with this data, so we should go with the linear model.

6. The `dvisits` data in the `faraway` package comes from the Australian Health Survey of 1977–78 and consist of 5190 observations on single adults, where young and old have been oversampled.

   (a) Build a Poisson regression model with `doctorco` as the response and `sex, age, agesq, income, levyplus, freepoor, freerepa, illness, actdays, hscore, chcond1` and `chcond2` as possible predictor variables. Considering the deviance of this model, does this model fit the data?

   Solution: Using stepwise model selection based on the AIC, we end up with the model `doctorco   sex + age + income + levyplus + freepoor + illness + actdays + hscore`. The deviance of 4385.5 is clearly not significant given that we have 5181 degrees of freedom, though note that the responses are not that large, so the deviance may not be close to a chi-squared distribution.

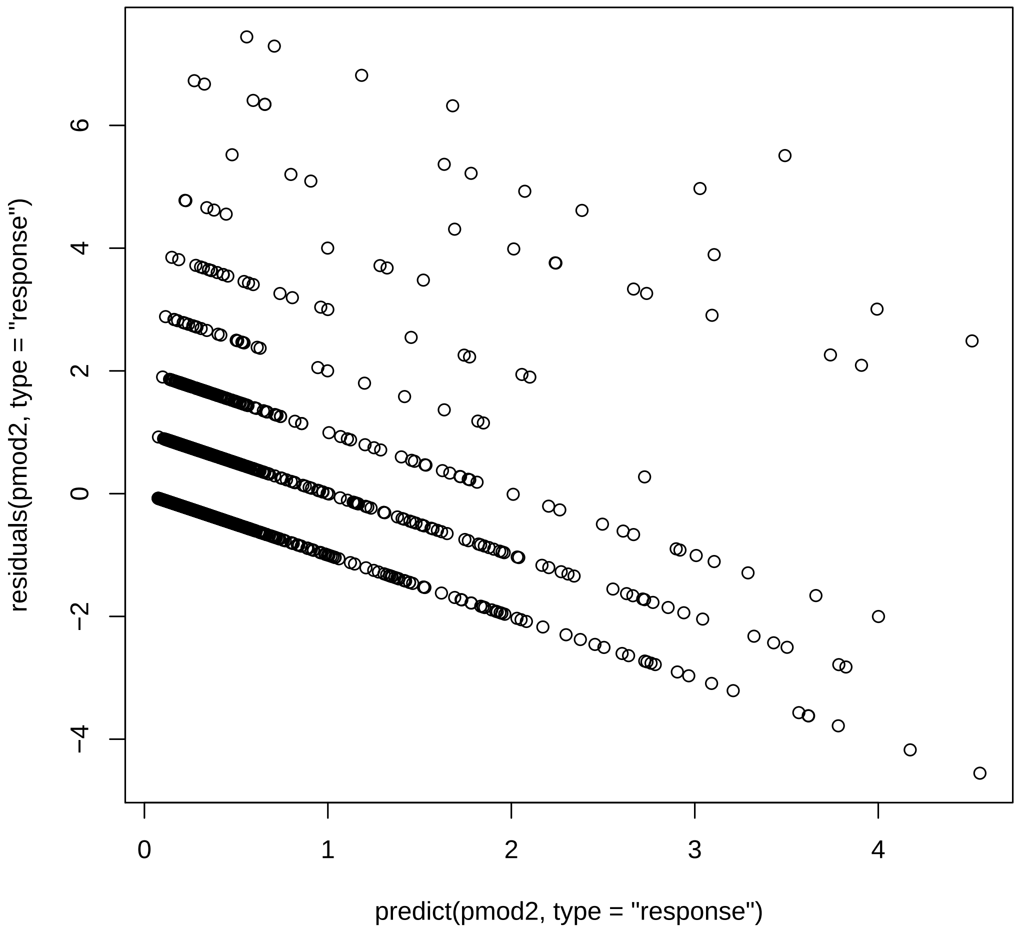
```
data(dvisits)
pmod <- glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor
+ freerepa + illness + actdays + hscore + chcond1,
family=poisson, data=dvisits)
pmod2 <- step(pmod, scope=~., trace=0)
summary(pmod2)
```

```
##
## Call:
## glm(formula = doctorco ~ sex + age + income + levyplus + freepoor +
##     illness + actdays + hscore, family = poisson, data = dvisits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0180  -0.6811  -0.5772  -0.4916   5.6590
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.072446   0.100191 -20.685  < 2e-16 ***
## sex          0.167591   0.055604   3.014 0.002578 **
## age          0.437894   0.137070   3.195 0.001400 **
## income      -0.203978   0.084206  -2.422 0.015420 *
## levyplus     0.087156   0.053501   1.629 0.103304
## freepoor    -0.465788   0.176364  -2.641 0.008265 **
## illness      0.196366   0.017603  11.155  < 2e-16 ***
## actdays      0.127994   0.004905  26.097  < 2e-16 ***
## hscore       0.032854   0.009961   3.298 0.000973 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5634.8  on 5189  degrees of freedom
## Residual deviance: 4385.5  on 5181  degrees of freedom
## AIC: 6735
##
## Number of Fisher Scoring iterations: 6
```

(b) Plot the response residuals against the fitted values. Why are there lines of observations on the plot?

Solution: The lines appear because the response reisuals are given by $y_i - g(\eta_i)$ and y_i only takes on finitely many values. Each line corresponds to a different possible value.

```
plot(predict(pmod2, type="response"), residuals(pmod2, type="response"))
```



```
table(dvisits$doctorco)

##
##    0    1    2    3    4    5    6    7    8    9
## 4141  782  174   30   24    9   12   12    5    1
```

(c) Use backward elimination with a critical p-value of 5% to reduce the model as much as possible.

Solution: Using backward elimination and chi-squared tests we end up with the model `doctorco   sex + age + income + freepoor + illness + actdays + hscore`, which is slightly smaller than the model achieved using the AIC and forward-backward elimination (just missing levyplus).

Note that the `step` function uses the AIC, so we have to use `drop1` instead. Here I just give the final step, which shows that we don't need to drop any more variables.

```
pmod3 <- glm(doctorco ~ sex + age +income + freepoor + illness + actdays
+ hscore, family=poisson, data=dvisits)
drop1(pmod3, scope=~., test="Chisq")

## Single term deletions
##
## Model:
## doctorco ~ sex + age + income + freepoor + illness + actdays +
##     hscore
##           Df Deviance    AIC    LRT  Pr(>Chi)
## <none>          4388.1 6735.7
## sex        1    4398.2 6743.8  10.14  0.001453 **
## age        1    4398.2 6743.7  10.06  0.001518 **
## income     1    4392.5 6738.1   4.43  0.035274 *
## freepoor   1    4397.4 6742.9   9.27  0.002335 **
## illness    1    4508.9 6854.5 120.82 < 2.2e-16 ***
## actdays    1    4956.5 7302.1 568.41 < 2.2e-16 ***
## hscore     1    4398.4 6744.0  10.31  0.001322 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) What sort of person would be predicted to visit the doctor the most under your selected model?

Solution Using a log link we have $\mu = e^{\eta}$, so we wish to maximise $\eta = \mathbf{x}^T\beta$. Looking at the coefficients this means female; as old as possible; no income; not entitled to free health care; very ill in the past two weeks; many days of reduced activity in the last two weeks; and a high hscore.

```
pmod3$coefficients

## (Intercept)         sex         age      income    freepoor     illness
## -2.05196250  0.17552865  0.43353243 -0.17105283 -0.49632492  0.19600786
##     actdays      hscore
##  0.12779329  0.03243268
```

(e) For the last person in the dataset, compute the predicted probability distribution for their visits to the doctor, i.e., give the probability they visit 0,1,2, etc. times.

Solution:

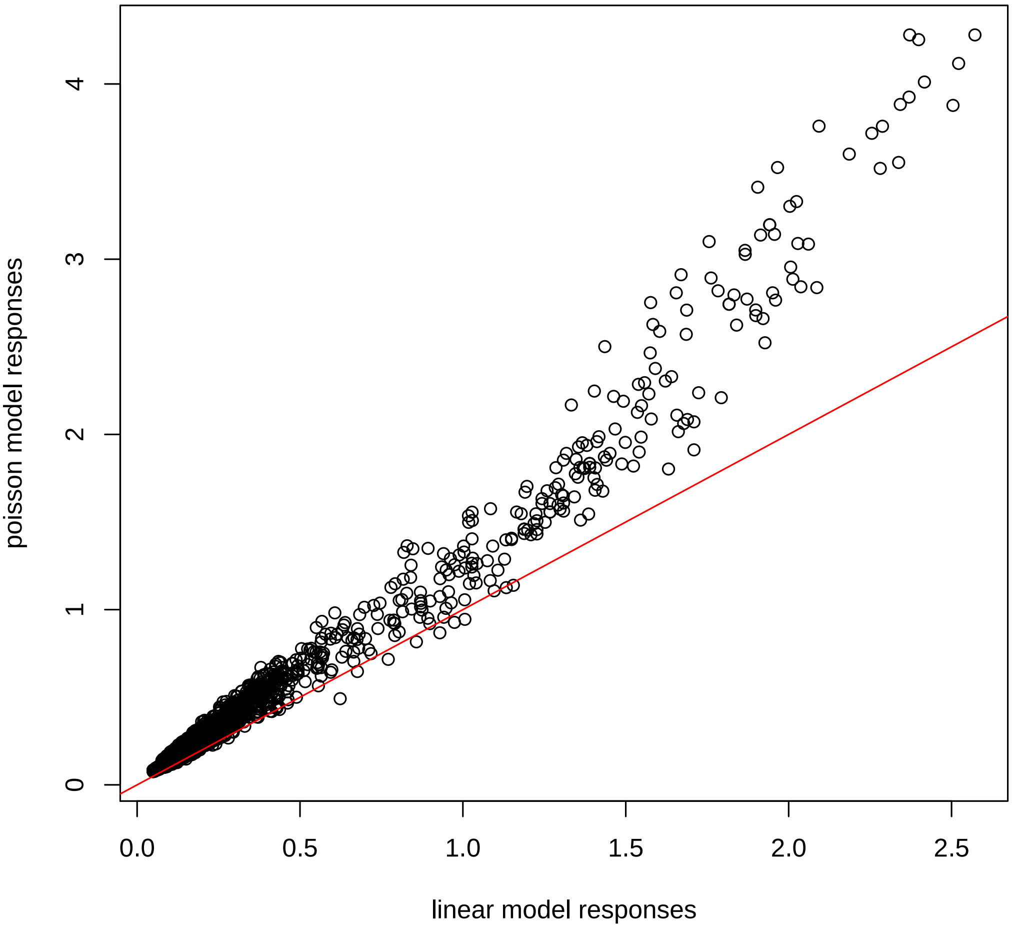
```
dim(dvisits)

## [1] 5190   19

lambda <- exp(predict(pmod3, dvisits[5190,]))
dpois(0:9, lambda)

##  [1] 8.451821e-01 1.421623e-01 1.195608e-02 6.703505e-04 2.818878e-05
##  [6] 9.482888e-07 2.658420e-08 6.387927e-10 1.343087e-11 2.510129e-13
```

(f) Fit a comparable (Gaussian) linear model and graphically compare the fits. Describe how the Gaussian and Poisson models differ.

Solution: We get a better fit by taking the log of the response (offset by 0.1, as the response can take zero values). The resulting linear model produces fitted values a lot like those of the poisson model.

Note that the mean of a log-normal random variable is given by $\exp(\mu + \sigma^2/2)$. Thus if Y is log-normal, and we estimate the mean μ and variance σ^2 of $\log(Y)$, then our estimate for $\mathbb{E}Y$ is $\exp(\hat{\mu} + \hat{\sigma}^2/2)$.

```
mod <- lm(log(doctorco + .1) ~ sex + age + agesq + income + levyplus
+ freepoor + freerepa + illness + actdays + hscore + chcond1,
data=dvisits)
mod2 <- step(mod, scope=~., trace=0)
mod2si2 <- deviance(mod2)/mod2$df.residual
plot(exp(predict(mod2) + mod2si2/2) - .1, predict(pmod3, type="response"),
xlab="linear model responses", ylab="poisson model responses")
abline(0, 1, col="red")
```



Although the linear model does surprisingly well, its fitted values are all a little smaller than the corresponding fitted values for the poisson model. The most important difference between how these two models are fitted is their variance structure. The poisson model assumes that $\operatorname{Var} Y \propto \mathbb{E} Y$ and the linear model assumes that $\operatorname{Var} \log Y$ and hence $\operatorname{Var} Y$ is constant. Thus the linear model will be giving too much weight to large responses.

7. Suppose that $Y_i \sim \text{Poisson}(\lambda_i)$, where $\lambda_i \propto t_i$. For example, if we record the number of burglaries reported in different cities, the observed number will depend on the number of households in these cities. In other cases, the size variable t may be time. For example, if we record the number of customers served by sales people, we must take account of the differing amounts of time worked.

   We can model the rate *per unit time* using a log link via

   $$\log(\lambda_i / t_i) = x_i^T \beta$$

   where x_i are known predictors and β unknown parameters. That is

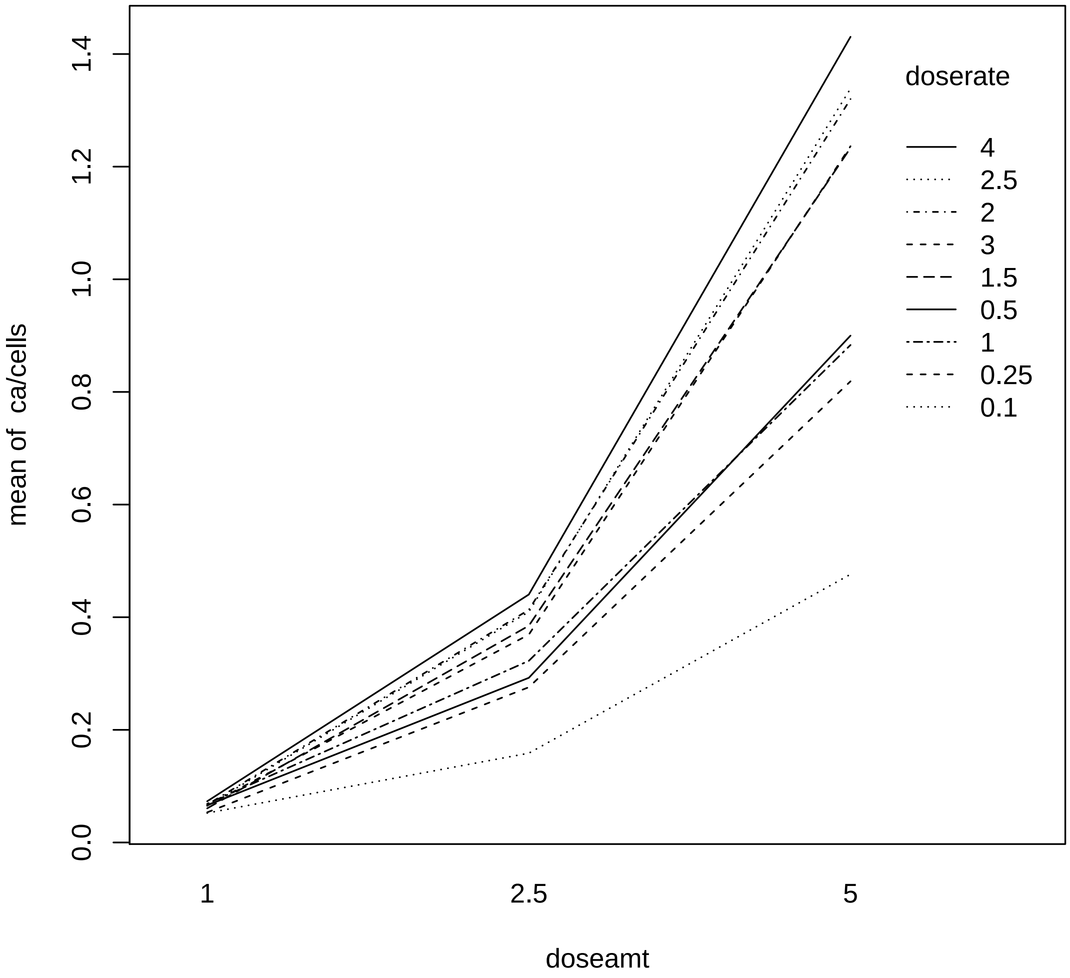
   $$\log(\lambda_i) = \log t_i + x_i^T \beta.$$

   This is of the form of a Poisson glm with log link, but where the coefficient of $\log t_i$ has been constrained to be 1. This is called a *rate model.*

   In an R model description we can fix the coefficient of a variable to 1 by enclosing it in the `offset` function, viz `y ∼ offset(log(t)) + x1 + x2 + ···`.

   In Purott and Reeder (1976), some data is presented from an experiment conducted to determine the effect of gamma radiation on the numbers of chromosomal abnormalities (ca) observed. The number (cells), in hundreds of cells exposed in each run, differs. The dose amount (doseamt) and the rate (doserate) at which the dose is applied are the predictors of interest. We can plot the data as follows

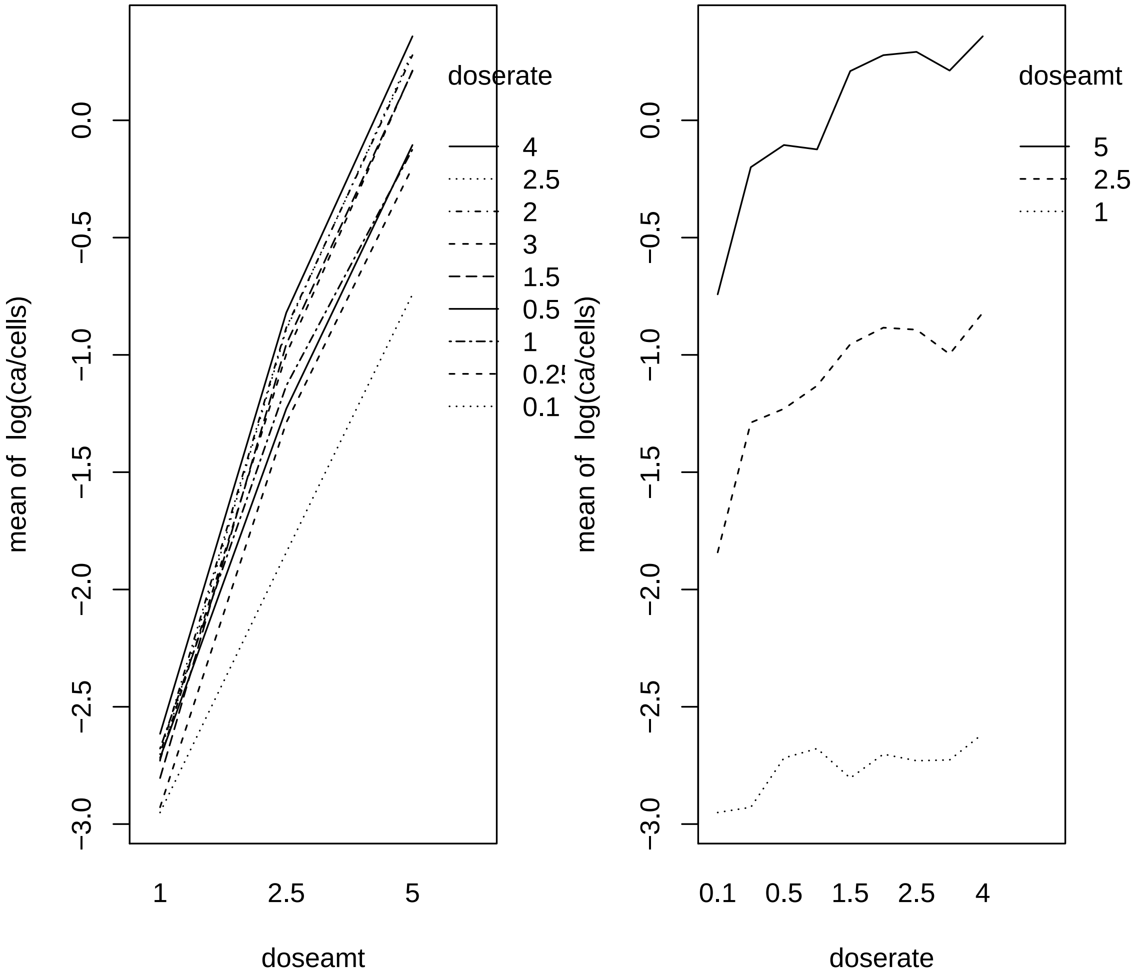
```
library(faraway)
data(dicentric)
with(dicentric, interaction.plot(doseamt, doserate, ca/cells))
```



Fit a rate model to this data. Use it to predict the rate of abnormalities when you have 200 cells, doserate 3.5 and doseamt 5.

Solution: Plotting log(ca/cells) against doseamt and doserate helps us judge linearity.

```
par(mfrow=c(1,2))
with(dicentric, interaction.plot(doseamt, doserate, log(ca/cells)))
with(dicentric, interaction.plot(doserate, doseamt, log(ca/cells)))
```



```
par(mfrow=c(1,1))
```

The plots show nice linear relationships between log(ca/cells) and both doseamt and doserate. They also show a possible interaction between doseamt and doserate, since the slope of doserate vs. log(ca/cells) seems to depend on doseamt (and vice versa). We can now fit the model:

```
model <- glm(ca ~ offset(log(cells)) + doserate*doseamt, family=poisson, data=dicentric)
summary(model)

##
## Call:
## glm(formula = ca ~ offset(log(cells)) + doserate * doseamt, family = poisson,
##     data = dicentric)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7308  -2.2842  -0.6264   3.3487   5.8272
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.29994    0.06160 -53.567  < 2e-16 ***
## doserate          0.06401    0.02922   2.191 0.028476 *
## doseamt           0.61224    0.01707  35.862  < 2e-16 ***
## doserate:doseamt  0.02715    0.00765   3.549 0.000387 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4753.00  on 26  degrees of freedom
## Residual deviance:  270.26  on 23  degrees of freedom
## AIC: 453.67
##
## Number of Fisher Scoring iterations: 4
```

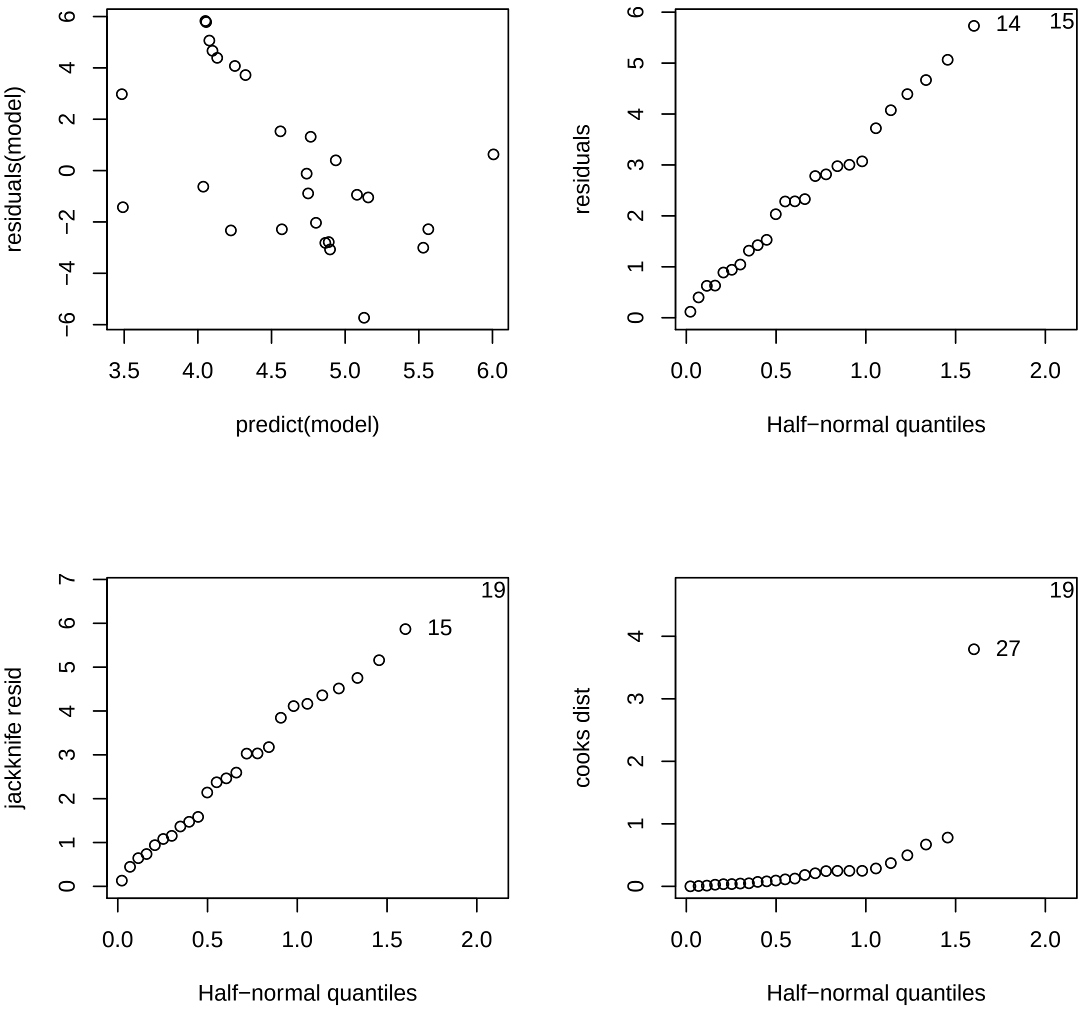
We can test the significance of the interaction using a chi-squared test. Not surprisingly, given its z-value, it appears very significant (but see below).

```
anova(model, test="Chi")

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: ca
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                26     4753.0
## doserate          1    231.3        25     4521.7 < 2.2e-16 ***
## doseamt           1   4238.7        24      282.9 < 2.2e-16 ***
## doserate:doseamt  1     12.7        23      270.3 0.0003681 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The deviance of our fitted model is very high. Our counts ca are reasonably large (the smallest is 25), so the deviance should look roughly chi-squared. Thus something is amiss with the model.

The residuals look mostly OK. Points 19 and 27 have a large Cook's distance, but aren't distinguished otherwise. You can check that if you fit a model omitting these points, then the coefficients do not change much and the deviance is still very high.

```
par(mfrow=c(2,2))
plot(predict(model), residuals(model))
halfnorm(residuals(model), ylab="residuals")
halfnorm(rstudent(model), ylab="jackknife resid")
halfnorm(cooks.distance(model), ylab="cooks dist")
```



```
par(mfrow=c(1,1))
```

The resaon for the high deviance is overdispersion.

8. Verify that for the binomial regression model with logistic link

$$\begin{aligned}\mathbb{E}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i} &= 0\\ -\mathbb{E}\frac{\partial^2 l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i\partial\theta_j} &= \mathbb{E}\left(\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_j}\right)\end{aligned}$$

Solution:

Suppose that $Y_k, k = 1, \cdots, n \sim Binomial(m_k, p_k = g^{-1}(\mathbf{x}_k^T\boldsymbol{\theta})$ where $\mathbf{x}_k, i = k, \cdots, n$ are explanatory predictors and $g(p) = \log(\frac{p}{1-p})$ is the logistic link function. Hence

$$\begin{aligned}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i} &= \sum_{k=1}^{n} Y_k\frac{1}{p_k}\frac{\partial p_k}{\partial\theta_i} - (m_k - Y_k)\frac{1}{1-p_k}\frac{\partial p_k}{\partial\theta_i}\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}Z_k\end{aligned}$$

where $Z_k = Y_k/p_k - (m_k - Y_k)/(1-p_k)$, so $\mathbb{E}(Z_k) = m_k p_k/p_k - m_k(1-p_k)/(1-p_k) = 0$ and $\mathrm{Var}\,(Z_k) = \mathrm{Var}\,(Y_k/(p_k(1-p_k)) - m_k/(1-p_k)) = m_k/(p_k(1-p_k))$. Thus

$$\begin{aligned}\mathbb{E}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i} &= \sum_{k=1}^{n}\mathbb{E}\left(\frac{\partial p_k}{\partial\theta_i}Z_k\right)\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\mathbb{E}(Z_k) = 0.\end{aligned}$$

Now

$$\begin{aligned}\frac{\partial^2 l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i\partial\theta_j} &= \sum_{k=1}^{n}\frac{\partial^2 p_k}{\partial\theta_i\partial\theta_j}Z_k + \frac{\partial p_k}{\partial\theta_i}\frac{\partial Z_k}{\partial\theta_j}\\ &= \sum_{k=1}^{n}\frac{\partial^2 p_k}{\partial\theta_i\partial\theta_j}Z_k + \frac{\partial p_k}{\partial\theta_i}\frac{\partial p_k}{\partial\theta_j}\left(\frac{-Y_k}{p_k^2} - \frac{m_k - Y_k}{(1-p_k)^2}\right)\end{aligned}$$

so

$$\begin{aligned}-\mathbb{E}\frac{\partial^2 l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i\partial\theta_j} &= -0 + \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\frac{\partial p_k}{\partial\theta_j}\left(\frac{m_k}{p_k} + \frac{m_k}{1-p_k}\right)\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\frac{\partial p_k}{\partial\theta_j}\frac{m_k}{p_k(1-p_k)}.\end{aligned}$$

Whereas

$$\left(\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_j}\right) = \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}Z_k\sum_{l=1}^{n}\frac{\partial p_l}{\partial\theta_j}Z_l$$

and, given the $Z_k, k = 1\cdots n$ are independent,

$$\begin{aligned}\mathbb{E}\left(\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_j}\right) &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\mathbb{E}\left(Z_k\sum_{l=1}^{n}\frac{\partial p_l}{\partial\theta_j}Z_l\right)\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\sum_{l=1}^{n}\frac{\partial p_l}{\partial\theta_j}\mathbb{E}(Z_kZ_l)\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\frac{\partial p_l}{\partial\theta_j}\left(\mathbb{E}(Z_k^2) + \sum_{l\neq k}\mathbb{E}(Z_kZ_l)\right)\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\frac{\partial p_l}{\partial\theta_j}\left(\mathrm{Var}\,(Z_k) + \sum_{l\neq k}\mathbb{E}(Z_k)\mathbb{E}(Z_l)\right)\\ &= \sum_{k=1}^{n}\frac{\partial p_k}{\partial\theta_i}\frac{\partial p_l}{\partial\theta_j}\frac{m_k}{p_k(1-p_k)} + 0\end{aligned}$$

as required. Notice that the proof does not rely on the form of the relationship between p_k and $\boldsymbol{\theta}$.

9. Suppose that $\mathbf{Y}$ has pdf $f(\mathbf{y};\boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$ in some fixed open set of $\mathbb{R}^k$ for some $k = 1, 2, \cdots$. Show that:

$$\begin{aligned}\mathbb{E}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i} &= 0\\ -\mathbb{E}\frac{\partial^2 l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i\partial\theta_j} &= \mathbb{E}\left(\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_i}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial\theta_j}\right)\end{aligned}$$

(Hint: You may assume that f is sufficiently regularly that you may interchange integration and differentiation in computing the expectations.)

Solution:

$$\begin{aligned}
\mathbb{E}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial \theta_i} &= \int \frac{\partial l(\boldsymbol{\theta};\mathbf{y})}{\partial \theta_i} f(\boldsymbol{\theta};\mathbf{y})\ d\mathbf{y} \\
&= \int \frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i} \frac{1}{f(\mathbf{y};\boldsymbol{\theta})} f(\mathbf{y};\boldsymbol{\theta})\ d\mathbf{y} \\
&= \int \frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i}\ d\mathbf{y} \\
&= \frac{\partial}{\partial \theta_i} \int f(\mathbf{y};\boldsymbol{\theta})\ d\mathbf{y} \\
&= \frac{\partial}{\partial \theta_i} 1 \\
&= 0
\end{aligned}$$

Now

$$\begin{aligned}
\frac{\partial^2 l(\boldsymbol{\theta};\mathbf{y})}{\partial \theta_i \partial \theta_j} &= \frac{\partial}{\partial \theta_i}\left(\frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_j}\frac{1}{f(\mathbf{y};\boldsymbol{\theta})}\right) \\
&= \frac{\partial^2 f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\frac{1}{f(\mathbf{y};\boldsymbol{\theta})} - \frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i}\frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_j}\frac{1}{f^2(\mathbf{y};\boldsymbol{\theta})}
\end{aligned}$$

Hence

$$\begin{aligned}
-\mathbb{E}\frac{\partial^2 l(\boldsymbol{\theta};\mathbf{Y})}{\partial \theta_i \partial \theta_j} &= -\int \frac{\partial^2 f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\frac{1}{f(\mathbf{y};\boldsymbol{\theta})} f(\mathbf{y};\boldsymbol{\theta})\ d\mathbf{y} \\
&+ \int \frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i}\frac{1}{f(\mathbf{y};\boldsymbol{\theta})}\frac{\partial f(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_j}\frac{1}{f(\mathbf{y};\boldsymbol{\theta})} f(\mathbf{y};\boldsymbol{\theta})\ d\mathbf{y} \\
&= -\frac{\partial}{\partial \theta_i}\int \frac{\partial l(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_j} f(\mathbf{y};\boldsymbol{\theta})\ d\mathbf{y} + \int \frac{\partial l(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_i}\frac{\partial l(\mathbf{y};\boldsymbol{\theta})}{\partial \theta_j} f(\mathbf{y};\boldsymbol{\theta})\ d\mathbf{y} \\
&= -\frac{\partial}{\partial \theta_i} 0 + \mathbb{E}\left(\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial \theta_i}\frac{\partial l(\boldsymbol{\theta};\mathbf{Y})}{\partial \theta_j}\right)
\end{aligned}$$

as required.

10. Suppose that students answer questions on a test and that a specific student has an aptitude T. A particular question might have difficulty d_i and the student will get the answer correct only if $T > d_i$. Consider d_i fixed and $T \sim N(\mu, \sigma^2)$, then the probability that a randomly selected student will get the answer wrong is $p_i = \mathbb{P}(T < d_i)$.

    Show how you might model this situation using a probit regression model.

    Solution: We have

$$\begin{aligned}
p_i &= \mathbb{P}(T < d_i) \\
&= \mathbb{P}\left(\frac{T-\mu}{\sigma} < \frac{d_i-\mu}{\sigma}\right) \\
&= \Phi\left(\frac{1}{\sigma}d_i - \frac{\mu}{\sigma}\right)
\end{aligned}$$

which is in the form of a probit regression model with predictor variable d, $\beta_0 = -\mu/\sigma$ and $\beta_1 = 1/\sigma$.

11. Show that the Gamma density, f, in the form

$$f(y;\lambda,\alpha) = \frac{1}{\Gamma(\alpha)}\lambda^\alpha y^{\alpha-1} e^{-\lambda y}$$

is an exponential family with $\theta = -\frac{\lambda}{\alpha}, \phi = \frac{1}{\alpha}$. Identify the functions a, b, c and find the mean and variance functions as functions of θ.

Solution: Notice that $\theta/\phi = \theta\alpha = -\lambda$, so

$$\log f(y; \theta, \phi) = \frac{y\theta + \log(-\theta)}{\phi} + \frac{1-\phi}{\phi}\log(y) - \frac{\log(\phi)}{\phi} - \log\left(\Gamma(1/\phi)\right)$$

So the functions are $a(\phi) = \phi, b(\theta) = -\log(-\theta), c(y, \phi) = \log(y)(1-\phi)/\phi - \log(\phi)/\phi - \log\left(\Gamma(1/\phi)\right)$. This gives $b'(\theta) = -1/\theta, b''(\tau) = \theta^{-2}$ giving $E(Y) = -1/\theta (= \alpha/\lambda)$ and $var(Y) = b''(\theta)a(\phi) = \theta^2\phi = \theta^2/\alpha = (\frac{\alpha}{\lambda^2})$.

Note that the formulas for the mean and variance are, of course, the same as those derived in MAST90105 using moment generating functions (or the change parameters trick). Also note that b' is self inverse so the canonical link function is the negative inverse.

12. Show that the inverse Gaussian density, f, in the form

$$f(y; \mu, \lambda) = \frac{\lambda}{\sqrt{2\pi y^3}} e^{-\frac{\lambda(y-\mu)^2}{2\mu^2 y}}$$

is an exponential family with $\theta = \frac{-1}{2\mu^2}, \phi = \frac{1}{\lambda}$. Identify the functions a, b, c and find the mean and variance functions as functions of μ, λ.

Solution: Writing the density function in terms of the defined parameters, θ, ϕ

$$\log f(y; \theta, \phi) = \frac{y\theta - (-\sqrt{-2\theta})}{\phi} + (-\log(\phi) - \log(2\pi)/2 - \frac{3}{2}\log(y) - \frac{1}{2\phi y}$$

.

So the functions are $a(\phi) = \phi, b(\theta) = -\sqrt{-2\theta}, c(y, \phi) = -\log(\phi) - \log(2\pi)/2 - \frac{3}{2}\log(y) - \frac{1}{2\phi y}$.

This gives $b'(\theta) = 1/\sqrt{-2\theta}, b''(\tau) = (-2\theta)^{-3/2}$ giving $E(Y) = 1/\sqrt{-2\theta} = \mu$ and $var(Y) = b''(\tau)a(\phi) = \theta^{-2}\phi = \mu^3/\lambda$.

Note that the canonical link is half the negative inverse of the square.