University of Glasgow

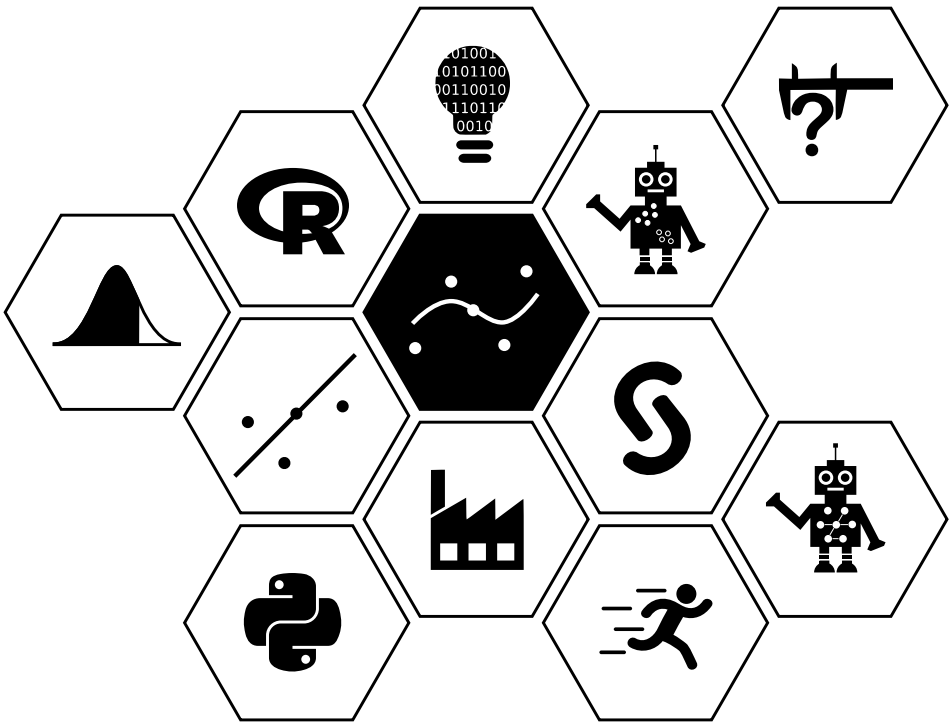
Advanced Predictive Models

Tereza Neocleous

Academic Year 2020-21

Week 5:

Models for counts



DATA ANALYTICS GLASGOW

Models for count responses

In this week's material we will cover GLMs that can be applied to count data. Examples of count data can be either rates (counts per unit time/area/distance), such as the number of

- hospital admissions due to respiratory disease in each of the 1235 intermediate geographies that make up Scotland
- nests per $9m^2$ of birds in a certain habitat
- train accidents in a given year

or cell frequencies in a **contingency table**, for instance numbers cross-classified by sex, age group and importance of power steering and air-conditioning in the car preference data we analysed in Week 4.

A common distribution for such data is the Poisson distribution. For Y the number of occurrences, we assume that Y follows the Poisson distribution $\text{Po}(\mu)$ with probability mass function given by

$$f(y) = \frac{\mu^y e^{-\mu}}{y!}, \qquad y = 0, 1, 2, \ldots$$

The mean and variance of $Y \sim \text{Po}(\mu)$ are both equal to μ. The parameter μ should be defined carefully: *e.g.* the average number of customers who buy a particular product out of every 100 customers who enter the store. The rate should also include a time scale, *e.g.* number of motor vehicle crashes per 1000 population per year. In general, the rate is specified in terms of units of **exposure**. Consider occupational injuries as an example: each worker is exposed for the period spent at work, so the rate can be defined in terms of person-years at risk. We wish to model the effect of explanatory variables on the response Y through the parameter μ.

Poisson regression

Let $Y_1, \ldots, Y_n$ be independent random variables with Y_i denoting the number of events occurred from exposure n_i for the ith covariate pattern. Then

$$E(Y_i) = \mu_i = n_i \theta_i.$$

For instance, Y_i could be the number of insurance claims for a particular make and model of car. This will depend on the number, n_i, of cars of this type that are insured and other variables that affect θ_i such as the age of the cars and where they are used. The subscript i denotes ith covariate pattern, that is the different combinations of explanatory variables such as the make and model of the car, its age, location etc.

The dependence on explanatory variables is usually modelled by $\theta_i = e^{\mathbf{x}_i^\top \boldsymbol{\beta}}$. The corresponding GLM is

$$E(Y_i) = \mu_i = n_i e^{\mathbf{x}_i^\top \boldsymbol{\beta}}; \qquad Y_i \sim \text{Po}(\mu_i)$$

This corresponds to the log link:

$$\log \mu_i = \log n_i + \mathbf{x}_i^\top \boldsymbol{\beta}$$

The term $\log n_i$ is called the **offset**.

Example 1 (Epidemiology data).

Suppose that we wish to model Y_i, the number of cancer cases in the ith intermediate geography (IG) in Glasgow for $i = 1, \ldots, 271$. The intermediate geographies are small areas that contain between 2,500 and 6,000 people. As the IGs can differ in population and demographics, a direct comparison of the number of cancer cases from each IG may not be appropriate. Instead, we use offsets, E_i, which are expected numbers of cancer cases in each IG, to allow for differences in population sizes and demographic structures.

Assume that the Y_i are independent $\text{Po}(\mu_i)$ with

$$\log(\mu_i) = \log(E_i) + \mathbf{x}_i^\top \boldsymbol{\beta}.$$

We can write this as $\log(\mu_i/E_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$ to emphasise that we are modelling the rate of occurrence of cancer. From the model equation we can see that the offset $\log(E_i)$ is a term with a fixed coefficient of 1.

The dataset `cancer.txt` contains the following variables:

- `Y_all`: number of cases of all types of cancer in the IG for the year 2013

- `E_all`: expected number of cases of all types of cancer for the IG based on the population size and demographics of the IG in 2013
- `pm10`: air pollution
- `smoke`: percentage of people in an area that smoke
- `ethnic`: percentage of people who are non-white
- `logprice`: natural log of average house price
- `easting` and `northing`: co-ordinates of the central point of the IG divided by 10,000

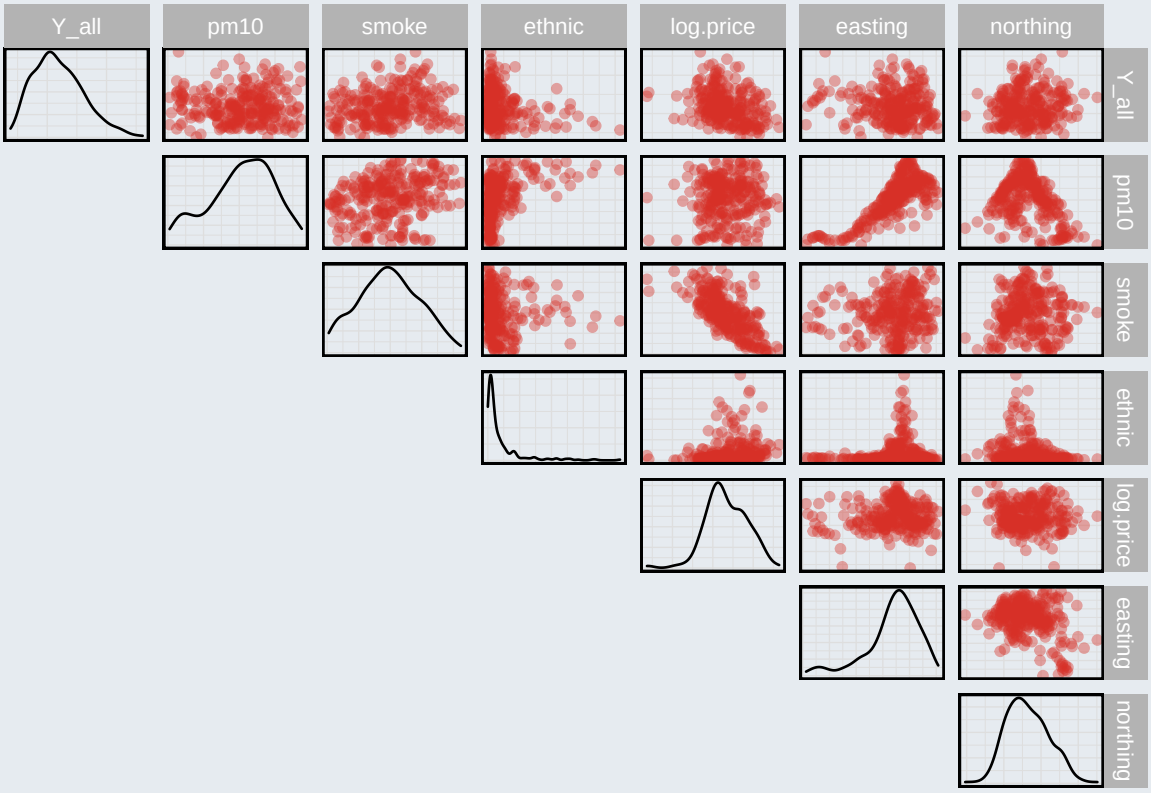
The first ten rows of the data are shown below.

```
cancer <- read.table(url("http://www.stats.gla.ac.uk/~tereza/rp/cancer.txt"),
                     header=TRUE)
head(cancer)
```

```
         IG Y_all     E_all pm10 smoke ethnic log.price  easting northing
1 S02000260   133 106.17907 17.8  21.9   5.58  11.59910 26.16245 66.96574
2 S02000261    38  62.43131 18.6  21.8   7.91  11.84940 26.29271 67.00278
3 S02000262    97 120.00694 18.6  20.8   9.58  11.74106 26.21429 67.04280
4 S02000263    80 109.10245 17.0  14.0  10.39  12.30138 25.45705 67.05938
5 S02000264   181 149.77821 18.6  15.2   5.67  11.88449 26.12484 67.09280
6 S02000265    77  82.31156 17.0  14.6   5.61  11.82004 25.37644 67.09826
```

Pair plots of the data are shown below.

```
ggpairs(cancer[,c(-1,-3)],
        upper=list(continuous=wrap("points", alpha=0.4, color="#d73027")),
        lower="blank", axisLabels="none")
```



We fit a Poisson regression model for these data, including a term for the offset, as follows:

```
epid1 <- glm(Y_all ~ pm10 + smoke + ethnic + log.price + easting +
                     northing+offset(log(E_all)), family = poisson, data = cancer)
summary(epid1)
```

```
Call:
glm(formula = Y_all ~ pm10 + smoke + ethnic + log.price + easting +
```

```
    northing + offset(log(E_all)), family = poisson, data = cancer)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2011  -0.9338  -0.1763   0.8959   3.8416

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.8592657  0.8029040  -1.070 0.284531
pm10         0.0500269  0.0066724   7.498 6.50e-14 ***
smoke        0.0033516  0.0009463   3.542 0.000397 ***
ethnic      -0.0049388  0.0006354  -7.773 7.66e-15 ***
log.price   -0.1034461  0.0169943  -6.087 1.15e-09 ***
easting     -0.0331305  0.0103698  -3.195 0.001399 **
northing     0.0300213  0.0111013   2.704 0.006845 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 972.94  on 270  degrees of freedom
Residual deviance: 565.18  on 264  degrees of freedom
AIC: 2356.2

Number of Fisher Scoring iterations: 4
```

Interpretation of regression coefficients

For a binary explanatory variable denoted by an indicator variable ($x_j = 0$ if the factor is absent and $x_j = 1$ if it is present), the *rate ratio*, RR, for presence vs. absence is

$$RR = \frac{E(Y_i|\text{present})}{E(Y_i|\text{absent})} = e^{\beta_j}.$$

Similarly, for a continuous explanatory variable x_k, a one-unit increase is associated with a multiplicative effect of e^{β_k} on the rate μ.

Hypothesis tests

Hypotheses about the parameters β_j can be tested using the **Wald** test (z-statistic and p-value obtained from a standard normal distribution). This is because, as we saw in Week 2, for the MLE $\hat{\beta}_j$ of parameter β_j, we have the asymptotic result

$$\frac{\hat{\beta}_j - \beta_j}{\text{se}(\hat{\beta}_j)} \sim N(0,1) \text{ approximately.}$$

Confidence intervals can be obtained in a similar way. By the same asymptotic result, we can take

$$\hat{\beta}_j \pm 1.96\text{se}(\hat{\beta}_j)$$

to obtain an approximate 95% confidence interval for $\hat{\beta}_j$. For intervals on the rate ratio scale, we simply take

$$\exp(\hat{\beta}_j \pm 1.96\text{se}(\hat{\beta}_j)).$$

Alternatively, hypothesis tests for nested models can be performed by comparing the difference in **deviance** with a chi-squared distribution with the appropriate degree of freedom.

Example 2 (Epidemiology data continued).

Suppose that we are interested in the effect of air pollution on health, and that we wish to interpret the coefficient of `pm10` to describe this effect on the cancer incidence rate. The coefficient is positive, which

suggests that the cancer incidence rate increases with increased pollution. The rate ratio allows us to quantify by how much. For every unit increase in `pm10`, the rate increases by a factor of $\exp(0.0500269) = 1.051$. For an approximate confidence interval we take $\exp(0.0500269 \pm 1.96 \times 0.0066724) = (1.038, 1.065)$. As the confidence interval does not include 1, this effect is significant.

Note: it is sometimes more useful to interpret the rate ratio associated with an increase of ω units in the exposure, where ω is often taken to be one standard deviation of that variable. For `pm10`, the standard deviation is

```
sd(cancer$pm10)
```

```
[1] 1.791206
```

and the corresponding rate ratio is

```
exp(0.0500269*sd(cancer$pm10))
```

```
[1] 1.093746
```

Task 1.

Interpret the coefficient of `smoke` in a similar way.

Fitted values and residuals

Fitted values can be obtained as

$$\hat{Y}_i = \hat{\mu}_i = n_i \exp(\mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}}), \qquad i = 1, \ldots, n.$$

These are denoted by e_i as they are estimates of the expected values $E(Y_i) = \mu_i$. In a similar way, we denote the observed values y_i by o_i.

For the Poisson distribution $\mathrm{Var}(Y_i) = E(Y_i)$ so the standard error of Y_i is estimated by $\sqrt{\hat{\mu}_i} = \sqrt{e_i}$.

In a normal linear model, residuals are defined as $y_i - \hat{\mu}_i = o_i - e_i$, and they are used to check assumptions such as linearity, normality and constant variance. In GLMs the variance is not constant but it varies with the mean and in fact in the Poisson model the variance is equal to the mean. For this reason, we cannot rely on the raw (also known as *response*) residuals $o_i - e_i$ and we use **Pearson** or **deviance** residuals instead. Note that these residuals were first introduced in the context of binomial models in Week 3.

Definition 1 (Pearson residuals).

*We define the ith **Pearson residual** as*

$$r_i = \frac{o_i - e_i}{\sqrt{e_i}}$$

where o_i is the observed value of Y_i and e_i is the corresponding fitted value.

Definition 2 (Pearson goodness-of-fit statistic).

*If we sum the Pearson residuals squared we obtain the **Pearson chi-squared statistic** X^2:*

$$X^2 = \sum r_i^2 = \sum \frac{(o_i - e_i)^2}{e_i}.$$

Note that this is the usual chi-squared goodness-of-fit statistic for contingency tables.

It can be shown that the **deviance** D for a Poisson model is

$$D = 2\left[\sum y_i \log(y_i/\hat{y}_i) - \sum (y_i - \hat{y}_i)\right] = 2\left[\sum o_i \log(o_i/e_i) - \sum (o_i - e_i)\right]$$

For most models $\sum o_i = \sum e_i$ so this simplifies even further to

$$D = 2 \sum o_i \log(o_i/e_i).$$

Definition 3 (Deviance residuals).

*The ith **deviance residual** is the square root of the contribution of the ith covariate pattern to the deviance, D:*

$$d_i = \text{sign}(o_i - e_i)\sqrt{2[o_i \log(o_i/e_i) - (o_i - e_i)]}, \qquad i = 1, \ldots, n,$$

so that $D = \sum d_i^2$. Here

$$\text{sign}(o_i - e_i) = \begin{cases} 1 & \text{if } o_i > e_i, \\ 0 & \text{if } o_i = e_i, \\ -1 & \text{if } o_i < e_i. \end{cases}$$

Pearson and deviance residuals can be used for identifying outliers and for checking the linearity assumption when plotted against explanatory variables in a GLM. However, residuals are not useful for GLMs with binary responses, binomial responses with small group sizes and Poisson responses that are relatively small because of the discreteness of the residuals in these cases.

Goodness-of-fit statistics

In addition to residual plots, we can check for lack of fit of a model using the **Pearson chi-squared statistic** and the **deviance**. These are closely related (they are both obtained by summing the respective residuals squared). Furthermore, they can be compared with a $\chi^2(n-p)$ distribution to assess the goodness of fit of a model in which p parameters are estimated.

The chi-squared distribution is likely to be a better approximation for X^2 than for D, although both rely on having sufficiently large fitted values.

We will look at diagnostics, measures of goodness of fit and overdispersion in Poisson regression through an example.

Regression models for count data – Galapagos example

https://youtu.be/GPsFwdZoDKo

Duration: 10m11s

Example 3 (GLMs for plant species in the Galapagos).

For 30 Galapagos islands the number of plant species found in each was recorded, along with several geographical variables.[a]

The dataset `gala` is available from `library(faraway)` in R and contains the following variables:

- *Species*, the number of species found on the island,
- *Endemics*, the number of endemic species,
- *Area*, the area of the island (km^2),
- *Elevation*, the highest elevation of the island (m),
- *Nearest*, the distance from the nearest island (km),
- *Scruz*, the distance from Santa Cruz island (km),
- *Adjacent*, the area of the adjacent island (km^2).

The first six rows of the data are shown below.

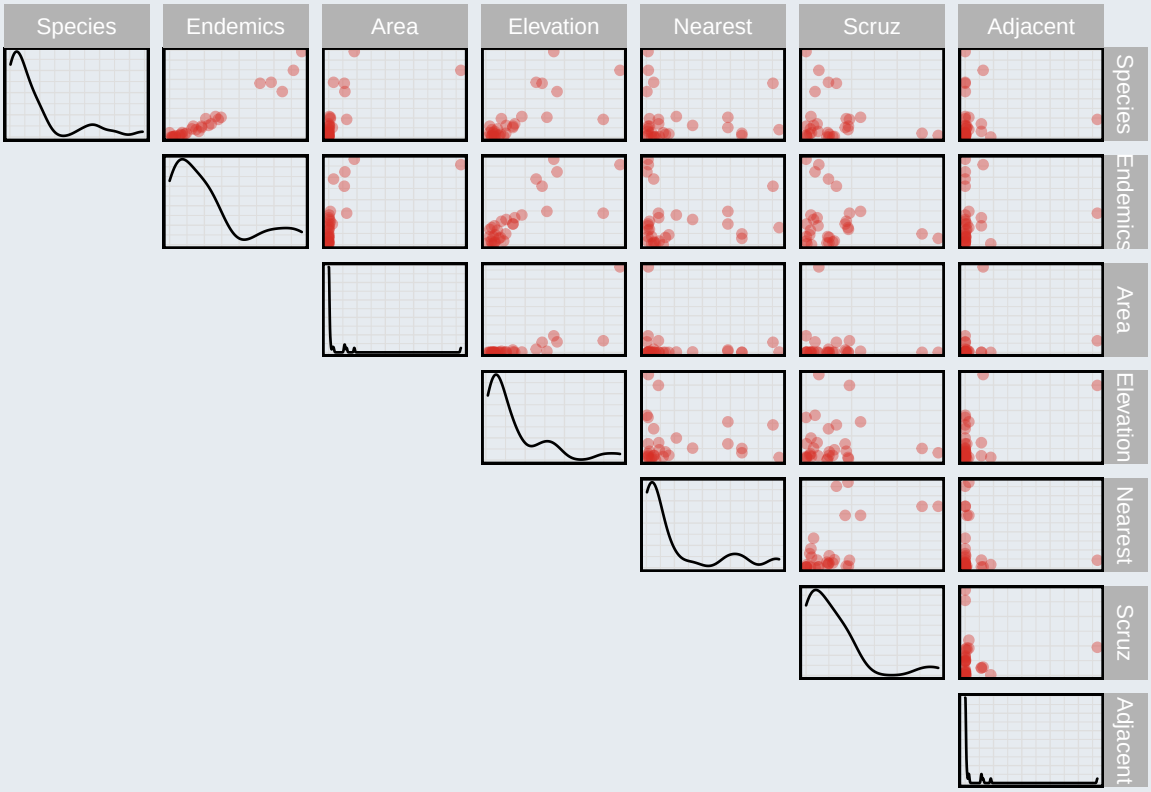
```
library(faraway)
head(gala)
```

```
             Species Endemics  Area Elevation Nearest Scruz Adjacent
Baltra            58       23 25.09       346     0.6   0.6     1.84
```

```
Bartolome          31       21  1.24       109      0.6  26.3    572.33
Caldwell            3        3  0.21       114      2.8  58.7      0.78
Champion           25        9  0.10        46      1.9  47.4      0.18
Coamano             2        1  0.05        77      1.9   1.9    903.82
Daphne.Major       18       11  0.34       119      8.0   8.0      1.84
```

We can explore these variables by constructing a pairs plot; notice there is a positive relationship between `Species` and `Elevation`, for instance, but that for many of the variables it is hard to see what the relationship with `Species` might be without log-transforming first.

```
ggpairs(gala, upper=list(continuous=wrap("points", alpha=0.4, color="#d73027")),
        lower="blank", axisLabels="none")
```



We fit a Poisson model for the number of species as a function of the geographical variables using the `glm()` function in R, making sure to specify `family = poisson` in the arguments:

```
gal1 <- glm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
            family = poisson, data = gala)
summary(gal1)
```

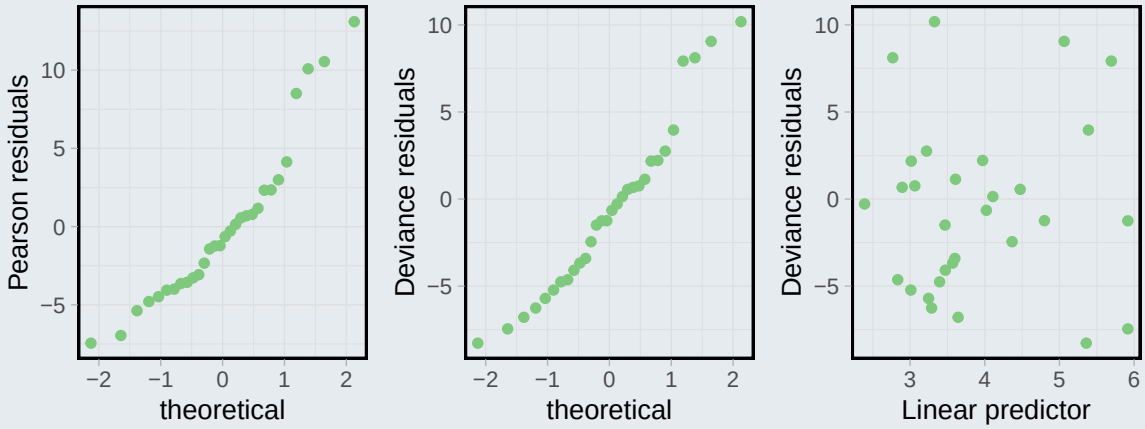
```
Call:
glm(formula = Species ~ Area + Elevation + Nearest + Scruz +
    Adjacent, family = poisson, data = gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.2752  -4.4966  -0.9443   1.9168  10.1849

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 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: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

The deviance is $D = 716.85$ which is very large compared with a $\chi^2(24)$, indicating a poor fit if the Poisson is the correct model for the response. The Pearson statistic is $X^2 = 761.97$, also indicating a poor fit.

We can check that the large deviance is not the result of an outlier by looking at residual plots. Although the first two plots below are normal probability plots, we are only using them here to spot any points that don't follow the straight line. We can also plot the deviance (or Pearson) residuals against the linear predictor to look for nonlinearity in the relationship between the fitted values and the residuals as shown in the third panel below. There is no obvious pattern here.

```
resp <- resid(gal1, type = "pearson")
resd <- resid(gal1, type = "deviance")

p1<- ggplot(gal1, aes(sample = resp)) + geom_point(stat = "qq", color = "#7fc97f") +
     ylab("Pearson residuals")
p2<- ggplot(gal1, aes(sample = resd)) + geom_point(stat = "qq", color = "#7fc97f") +
     ylab("Deviance residuals")
p3<- ggplot(gal1, aes(x = predict(gal1, type="link"), y =resd))+
        geom_point(col = "#7fc97f") +
        ylab("Deviance residuals") + xlab("Linear predictor")
grid.arrange(p1, p2, p3, nrow = 1)
```

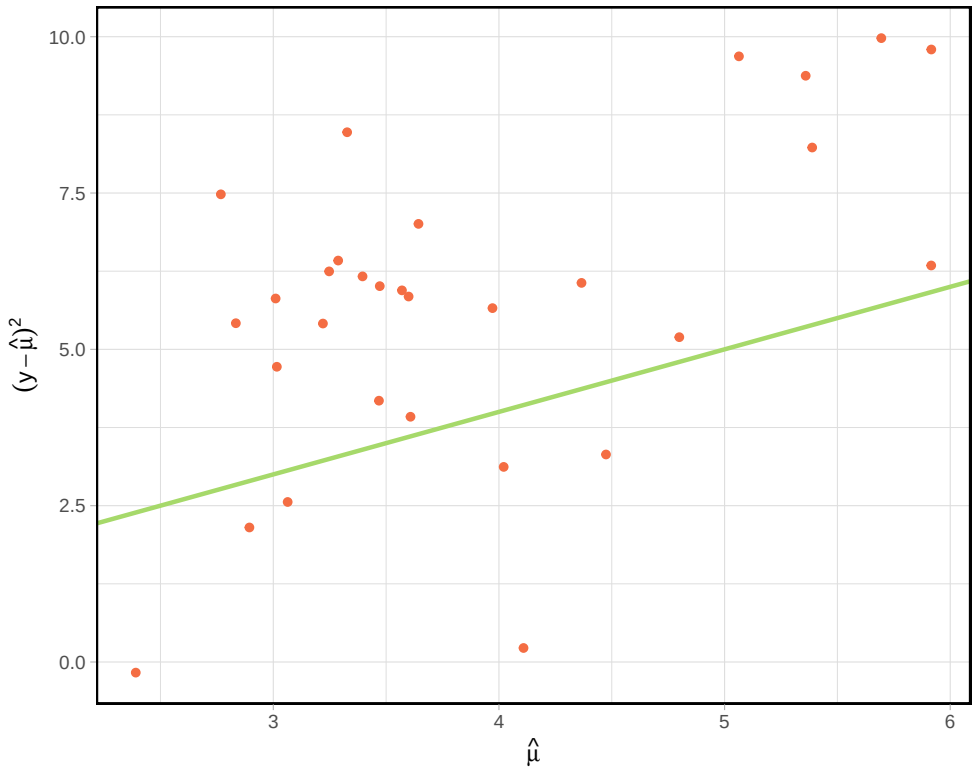


[a] M. P. Johnson and P. H. Raven (1973) "Species number and endemism: The Galapagos Archipelago revisited" Science, 179, 893-895.

Overdispersion

Suppose that in a Poisson regression model the link function and choice of explanatory variables is correct, but that the assumption that $\mathrm{Var}(Y_i) = \mu_i$ does not hold. If $\mathrm{Var}(Y_i) > \mu_i$ we say that we have **overdispersion**. This appears to be the case for the Galapagos data – notice in the figure below that most of the points lie above the line of equality for mean and variance.

```
ggplot(gal1, aes(x=log(fitted(gal1)), y=log((gala$Species-fitted(gal1))^2)))+
       geom_point(col="#f46d43") +
       geom_abline(slope=1, intercept=0, col="#a6d96a", size=1) +
       ylab(expression((y-hat(mu))^2)) + xlab(expression(hat(mu)))
```



The issue with overdispersion is that while the regression parameter estimates are stil consistent, their standard errors will be wrong. In this case we are not able to determine which explanatory variables are significant.

Quasi-Poisson model

One way to deal with overdispersion is to introduce a dispersion parameter ϕ such that $\mathrm{Var}(Y_i) = \phi\mu_i$. We can estimate this dispersion parameter by

$$\hat{\phi} = \frac{X^2}{n-p}.$$

```
X2 <- sum(resid(gal1, type = "pearson")^2)
dp <- X2 / gal1$df.res
dp
```

```
[1] 31.74914
```

The dispersion parameter is then used to adjust the standard errors in the summary. Notice that the regression coefficients do not change.

```
summary(gal1, dispersion = dp)
```

```
Call:
glm(formula = Species ~ Area + Elevation + Nearest + Scruz +
    Adjacent, family = poisson, data = gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.2752  -4.4966  -0.9443   1.9168  10.1849

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.1548079  0.2915897  10.819  < 2e-16 ***
Area        -0.0005799  0.0001480  -3.918 8.95e-05 ***
Elevation    0.0035406  0.0004925   7.189 6.53e-13 ***
Nearest      0.0088256  0.0102621   0.860    0.390
Scruz       -0.0057094  0.0035251  -1.620    0.105
Adjacent    -0.0006630  0.0001653  -4.012 6.01e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 31.74914)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

With the use of the estimated dispersion parameter the Wald tests are not very reliable, so we turn to an F test to determine the significance of the regression coefficients:

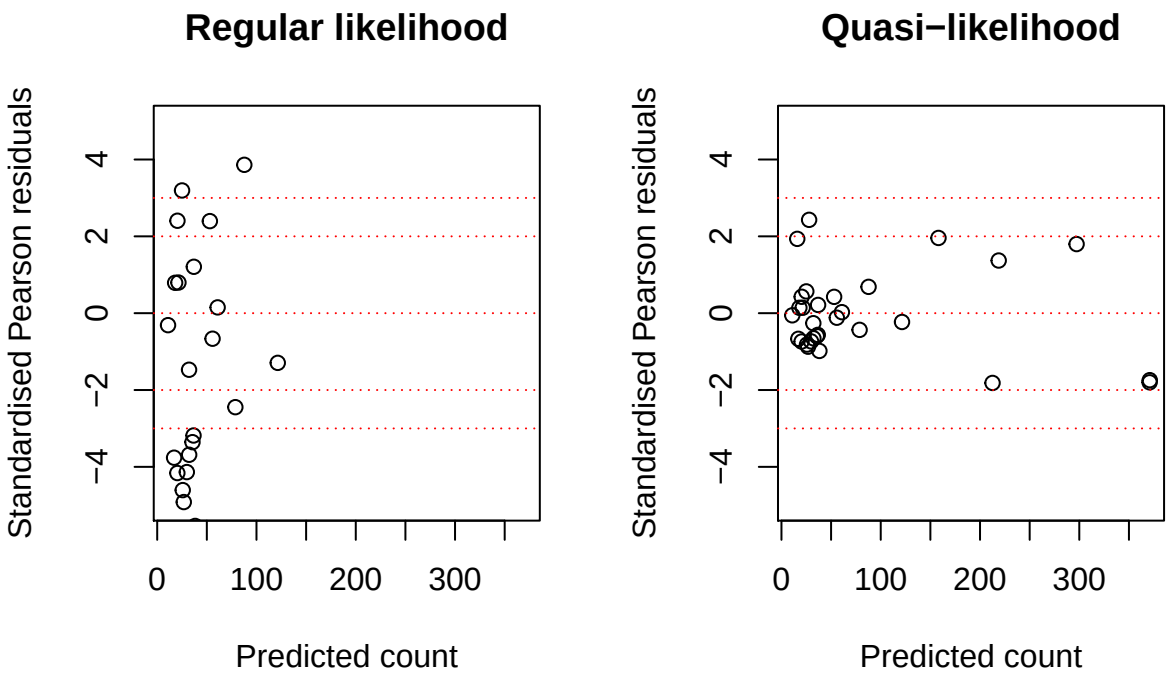
```
drop1(gal1, test = "F")
```

```
Single term deletions

Model:
Species ~ Area + Elevation + Nearest + Scruz + Adjacent
          Df Deviance     AIC F value    Pr(>F)
<none>         716.85  889.68
Area       1  1204.35 1375.18 16.3217 0.0004762 ***
Elevation  1  2389.57 2560.40 56.0028 1.007e-07 ***
Nearest    1   739.41  910.24  0.7555 0.3933572
Scruz      1   813.62  984.45  3.2400 0.0844448 .
Adjacent   1  1341.45 1512.29 20.9119 0.0001230 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We could now perform variable selection by dropping `Nearest` first and then repeating the process until only significant terms are left in the model.

The following residual plots show the effect of the quasi-Poisson model on the residuals.

```
# Residual plots vs. predicted
pred <- predict(gal1, type = "response")
stand.resid <- rstandard(model = gal1, type = "pearson") # Standardised Pearson residuals

par(mfrow=c(1,2))
plot(x = pred, y = stand.resid, xlab = "Predicted count", ylab = "Standardised Pearson residuals",
   main = "Regular likelihood", ylim = c(-5,5))
abline(h = c(-3, -2, 0, 2, 3), lty = "dotted", col = "red")


gal2 <- glm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
            family = quasipoisson(link = "log"), data = gala) # Quasi-Poisson model

pred <- predict(gal2, type = "response")
stand.resid <- rstandard(model = gal2, type = "pearson") # Standardised Pearson residuals

plot(x = pred, y = stand.resid, xlab = "Predicted count", ylab = "Standardised Pearson residuals",
   main = "Quasi-likelihood", ylim = c(-5,5))
abline(h = c(-3, -2, 0, 2, 3), lty = "dotted", col = "red")
```



We can see in the plots that all residuals are contained within ± 3 in the quasi-Poisson model, while quite a few are outside this range for the original Poisson model.

Another way to deal with overdispersion is to assume a more flexible distribution for the response that would allow for a variance that is larger than the mean. The negative binomial distribution is one such distribution.

Negative binomial models

We can deal with overdispersion by assuming a negative binomial distribution for the response, which allows for a variance larger than the mean. The form of the GLM is still the same as in the Poisson model with a linear component $\mathbf{x}_i^\top \boldsymbol{\beta}$. The link function is taken to be the log link, and the model equation is $g(\mu_i) = \log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$. Hence, the interpretation of the regression coefficients is similar to the Poisson case.

One parameterisation for a random variable Y_i following the negative binomial distribution is

$$f(y_i; \theta, \mu_i) = \frac{\Gamma(\theta + y_i)}{\Gamma(\theta) y_i!} \cdot \frac{\mu_i^{y_i} \theta^\theta}{(\mu_i + \theta)^{y_i + \theta}} \quad \text{for } y = 0, 1, 2, \ldots$$

The mean is $E(Y_i) = \mu_i$ but we can also see that $\text{Var}(Y_i) = \mu_i + \dfrac{\mu_i^2}{\theta} > E(Y_i)$. When fitting a negative binomial GLM, we estimate both the mean, μ_i, and the parameter θ.

Example 4 (Negative binomial model for the Galapagos plant species data).

To fit a negative binomial model to the Galapagos data to account for the overdispersion, we use the function `glm.nb()` from `library(MASS)`:

```
library(MASS)
gal3 <- glm.nb(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
               data = gala)
summary(gal3)
```

```
Call:
glm.nb(formula = Species ~ Area + Elevation + Nearest + Scruz +
    Adjacent, data = gala, init.theta = 1.674602286, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1344  -0.8597  -0.1476   0.4576   1.8416
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.9065247  0.2510344  11.578  < 2e-16 ***
Area        -0.0006336  0.0002865  -2.211 0.027009 *
Elevation    0.0038551  0.0006916   5.574 2.49e-08 ***
Nearest      0.0028264  0.0136618   0.207 0.836100
Scruz       -0.0018976  0.0028096  -0.675 0.499426
Adjacent    -0.0007605  0.0002278  -3.338 0.000842 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.6746) family taken to be 1)

    Null deviance: 88.431  on 29  degrees of freedom
Residual deviance: 33.196  on 24  degrees of freedom
AIC: 304.22

Number of Fisher Scoring iterations: 1

              Theta:  1.675
          Std. Err.:  0.442

 2 x log-likelihood:  -290.223
```

We can compare the Poisson and negative binomial models by looking at their deviances and AIC scores:

```
# Poisson model
c(gal1$deviance, gal1$aic)
```

```
[1] 716.8458 889.6767
```

```
# Negative binomial model
c(gal3$deviance, gal3$aic)
```

```
[1]  33.19644 304.22284
```

Both are much smaller for the negative binomial model.

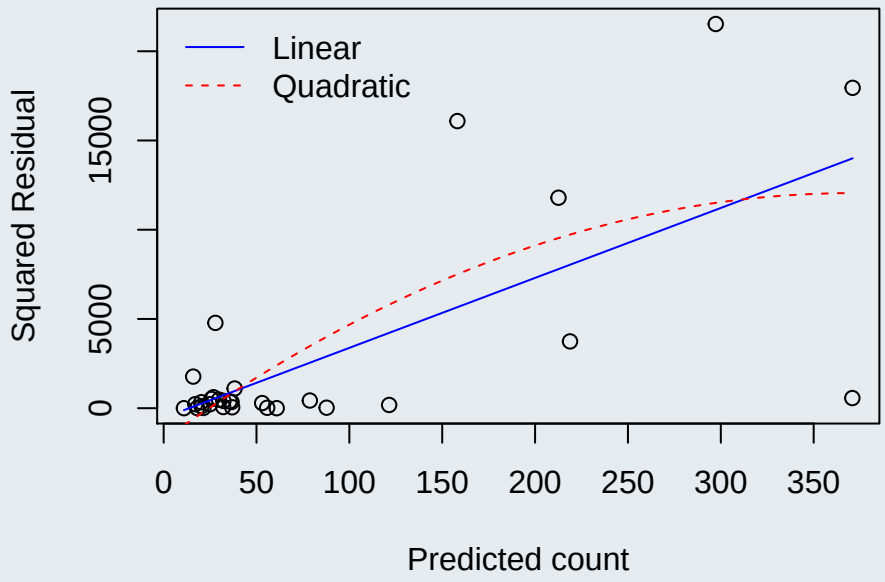
Which correction for overdispersion should we choose, the quasi-Poisson or the negative binomial model? It may not matter much in practice, but one way to decide between the two is by plooting $(y_i - \hat{\mu}_i)^2$ v $\hat{\mu}_i$. We then plot a linear and quadratic fit to see which one of them fits better. If the relationship is linear the quasi-Poisson model is better – if quadratic, the negative binomial model is better.

```
# Plot of squared residuals v predicted values
res.sq <- residuals(gal1, type = "response")^2
set1 <- data.frame(res.sq, mu.hat = gal1$fitted.values)

fit.lin <- lm(formula = res.sq ~ mu.hat, data = set1)
fit.quad <- lm(formula = res.sq ~ mu.hat + I(mu.hat^2), data = set1)
summary(fit.quad)
```

```
Call:
lm(formula = res.sq ~ mu.hat + I(mu.hat^2), data = set1)

Residuals:
     Min       1Q   Median       3Q      Max
-11484.2  -1415.1    174.6    717.2  10022.3

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.760e+03  1.427e+03  -1.233   0.2281
mu.hat       7.444e+01  3.027e+01   2.459   0.0206 *
I(mu.hat^2) -1.003e-01  8.345e-02  -1.202   0.2399
---
```

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

Residual standard error: 4222 on 27 degrees of freedom
Multiple R-squared:  0.5146,    Adjusted R-squared:  0.4787
F-statistic: 14.31 on 2 and 27 DF,  p-value: 5.782e-05
```

```
plot(set1$mu.hat, y = set1$res.sq, xlab = "Predicted count",
     ylab = "Squared Residual")
curve(expr = predict(fit.lin, newdata = data.frame(mu.hat = x), type = "response"),
      col = "blue", add = TRUE, lty = "solid")
curve(expr = predict(fit.quad, newdata = data.frame(mu.hat = x), type = "response"),
      col = "red", add = TRUE, lty = "dashed")
legend("topleft", legend = c("Linear", "Quadratic"), col = c( "blue", "red"),
    lty = c("solid", "dashed"), bty = "n")
```



Here the blue line is the straight line fit which would indicate that the quasi-Poisson model suffices, and the red line is the quadratic fit which would suggest using a negative binomial model.

The quadratic coefficient in the above model is not significant (p-value of 0.2399). We don't have any evidence that the negative binomial model is better.

Note: Correcting for overdispersion should **not** be the first step in an analysis. If more explanatory variables are available or if the model can be improved in other ways, it's best to do that rather than just account for the excess variance. Corrections for overdispersion are just a patch, and are unlikely to solve problems with poor predictive performance. It's preferable to try and build a better model before resorting to these fixes.

Task 2.

Fit a Poisson regression model to the Galapagos data after log-transforming the explanatory variables. Does this model fit the data better than the original Poisson model? Are all the explanatory variables significant?

Task 3.

In Example 1, the residual deviance of the fitted model is quite high compared to the degrees of freedom. Could this be due to overdispersion? Fit a quasi-Poisson model and a negative binomial model and explore diagnostic plots to check if one of these might be more appropriate.

Example 5 (Predicting the total number of medals in the 2012 Olympics).

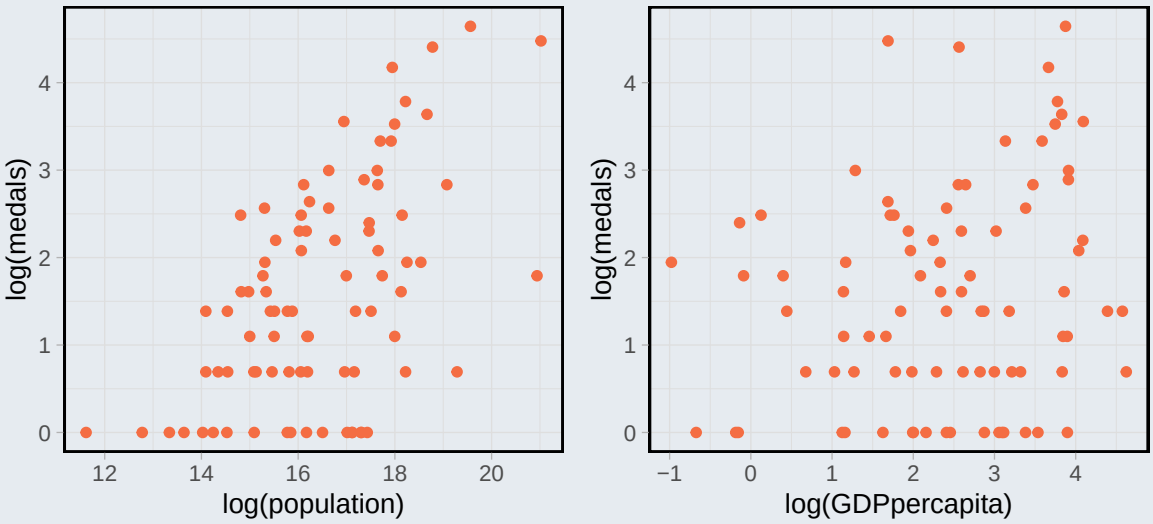
We wish to model the number of medals won by each country in the London Olympics in 2012 as a function of the country's population and GDP per capita. The dataset `OlympicMedals2012.csv` contains this information for all the countries that won at least one medal in the 2012 Games. A ranking of countries based on the total medals won in the Olympics can be found at https://en.wikipedia.org/wiki/2012_Summer_Olympics_medal_table. We start by creating a variable for GDP per capita in 1000 US dollars and looking at some plots of the data.

```
olympics0 <- read.csv(url("http://www.stats.gla.ac.uk/~tereza/rp/OlympicMedals2012.csv"))
olympics <- data.frame(country = olympics0$Country, medals = olympics0$Medals,
                       population = olympics0$Population,
                       gold = olympics0$Gold.Medal,
                       GDP = olympics0$GDP..US.Billion)
olympics$GDPpercapita <- olympics$GDP * 10^6 / olympics$population
head(olympics)
```

```
              country medals population gold    GDP GDPpercapita
1             Grenada      1     110821    1   0.82     7.399320
2             Jamaica     12    2705827    4  15.07     5.569462
3 Trinidad and Tobago      4    1317714    1  22.48    17.059848
4         New Zealand     13    4432620    6 130.68    29.481435
5             Bahamas      1     353658    1   7.79    22.026930
6            Slovenia      4    2057540    1  49.54    24.077296
```

The wide range of values for both population and GDP per capita suggests log-transforming both explanatory variables. From the plots we see a positive association between log(population) and log(medals) and also between log(GDP per capita) and log(medals).

```
p1 <- ggplot(olympics, aes(x=log(population), y=log(medals))) +
       geom_point(col="#f46d43")
p2 <- ggplot(olympics, aes(x=log(GDPpercapita), y=log(medals))) +
       geom_point(col="#f46d43")

grid.arrange(p1, p2, nrow=1)
```



We start the analysis by fitting a Poisson regression model.

```
ol1 <- glm(medals ~ log(population) + log(GDPpercapita),
           family = poisson, data = olympics)
summary(ol1)
```

```
Call:
glm(formula = medals ~ log(population) + log(GDPpercapita), family = poisson,
    data = olympics)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.0028  -2.2661  -0.3188   1.1572   8.2493

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -9.11251    0.39869  -22.86   <2e-16 ***
log(population)    0.59485    0.02025   29.38   <2e-16 ***
log(GDPpercapita)  0.49756    0.02973   16.74   <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: 1567.7  on 84  degrees of freedom
Residual deviance:  547.5  on 82  degrees of freedom
AIC: 849.33

Number of Fisher Scoring iterations: 5
```

Large population and high GDP per capita appear to contribute to Olympic success as indicated by the positive coefficients of both terms in the Poisson model.

Task 4.

The fit of the Poisson model is not particularly good (just look at the deviance of 547.5 on 82 degrees of freedom). Supposing that this is due to overdispersion, fit a quasi-Poisson model to adjust for it.

Task 5.

Fit a negative binomial model to the Olympics data.

Task 6.

Examine the residuals from the negative binomial model and comment on any unusual observations.

Task 7.

Do you think that either the Poisson or the negative binomial model with log(GDP) and log(population) as predictors would do a good job predicting the total number of medals won by countries in 2012? Explain.

Excess zeros

Sometimes overdispersion is observed due to excess zeros in the data. As an example, consider counting burst pipes in a city's water mains system: most of the time the number will be zero. A Poisson distribution does not cope well with too many zeros. A negative binomial distribution usually does better, but there are other options specifically for this type of data. Two types of models, **zero-inflated** and **hurdle** are particularly relevant.

Zero-inflated models

In zero-inflated Poisson or negative binomial models, we assume that there are two processes that could be generating zeros in the data. One is a Bernoulli process and the other a Poisson process and the resulting data distribution is a

mixture of the two. Such models may be appropriate in the following contexts:

- We are interested in the number of items bought, where the decision to buy these items can be influenced by a number of factors (these would go into the binary logistic regression part of the model). But even after the decision to buy the items, some people end up with zero items bought because they were out of stock or for some other reason.
- We are interested in the number of visits to the doctor, where the *decision* to visit the doctor is influenced by gender, age and other variables, and the *number* of visits by some other factors (or even the same, but not necessarily so). Zero visits could be because a person never goes to the doctor or because there was no occasion for which to go to the doctor.

To fit zero-inflated models, one can use the `zeroinfl()` function from `library(pscl)` in R. You can read more about this type of model and see examples of zero-inflated Poisson and negative binomial models in the package vignette available from https://cran.r-project.org/web/packages/pscl/vignettes/countreg.pdf.

Hurdle models

Another type of model for data with excess zeros is a hurdle model. It assumes that there is a sequential process that first determines whether the count will be zero, and if the count is not zero then it has to be positive. Examples:

- Consider the number of days spent in hospital for patients arriving at A&E. The patients will either be admitted, in which case they will spend a number of days in hospital, or not, in which case the number of days hospitalised should be zero.
- Suppose that we are interested in the number of cigarettes consumed per week. Subjects will either be non-smokers, in which case their consumption is zero, or smokers and will therefore have a positive consumption. A set of explanatory variables could be used to predict whether a person is a smoker or not, and a potentially different set of explanatory variables could be used to predict the number of cigarettes for those who smoke.

To fit hurdle models, one can use the `hurdle()` function from `library(pscl)` in R.

Task 8.

In the Olympic medals example, suppose we also had data on the countries that participated in the 2012 Olympics but did not win any medals. What type of model would you consider for these data and why?

Loglinear models

The last component of this unit is on models for contingency tables, in which we have a count response tabulated by the levels of categorical explanatory variables. An example of such data is the car preference dataset we analysed in Week 4.

If there is no constraint on the row or column totals of the table, we can assume that the counts are Poisson-distributed. Otherwise, depending on the constraint, we have a multinomial or product-multinomial distribution. All of these can be modelled using a GLM with a Poisson response and the log link. Usually the question of interest for such data is whether there is an association between the factors. To illustrate this, let us look at an example.

Loglinear models applied to food poisoning data

https://youtu.be/Ye6DSu_nRp0

Duration: 8m14s

Example 6 (Loglinear model for food poisoning data).

After a food poisoning outbreak, conference participants were surveyed in an effort to identify the cause of the outbreak. The potato salad and crab salad served at the conference were considered as possible sources. Participants were asked if they had either, both or none, and also if the got sick (food poisoned). The data are shown in the table below.

	Potato		No Potato	
	Crab	No Crab	Crab	No Crab
Not sick	80	24	31	23
Sick	120	22	4	0
Total	200	46	35	23

The data can be read into R as follows:

```
fp <- data.frame(potato=rep(c("yes","yes","no","no"),2),
                            crab=rep(c("yes","no"),4),
                            sick=c(rep("no",4),rep("yes",4)),
                            freq = c(80,24,31,23,120,22,4,0))
```

Because the column totals in the contingency table are fixed (they are determined by which salad(s) people had), the minimal model we can fit must include terms for `crab`, `potato` and the interaction between the two. We use the shortcut notation $[PC]$ for this model. The model fit is given below:

```
summary(glm(freq ~ crab*potato, family=poisson, data=fp))
```

```
Call:
glm(formula = freq ~ crab * potato, family = poisson, data = fp)

Deviance Residuals:
      1       2       3       4       5       6       7       8
-2.0729  0.2070  2.9070  2.9807  1.9383 -0.2101 -3.8978 -4.7958

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         2.4423     0.2085  11.714  < 2e-16 ***
crabyes             0.4199     0.2684   1.564 0.117766
potatoyes           0.6931     0.2554   2.714 0.006641 **
crabyes:potatoyes   1.0498     0.3143   3.340 0.000837 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 295.253  on 7  degrees of freedom
Residual deviance:  63.669  on 4  degrees of freedom
AIC: 108

Number of Fisher Scoring iterations: 5
```

Note that we won't focus on the regression coefficients themselves, but rather on which terms are included in the model.

To explore any relationship with food poisoning, we need to include terms involving `sick`. The independence model $[PC, S]$, which assumes that neither the potato salad, nor the crab salad had anything to do with the food poisoning, can be fit by using:

```
glm(freq ~ crab*potato+sick, family=poisson, data=fp)
```

```
Call:  glm(formula = freq ~ crab * potato + sick, family = poisson,
    data = fp)

Coefficients:
      (Intercept)            crabyes          potatoyes            sickyes  crabyes:potatoyes
          2.48106            0.41985            0.69315           -0.07899            1.04982

Degrees of Freedom: 7 Total (i.e. Null);  3 Residual
```

```
Null Deviance:      295.3
Residual Deviance: 63.2     AIC: 109.5
```

What we want to keep from this model is the residual deviance, to be used for comparisons with other nested models:

```
deviance(glm(freq ~ crab*potato+sick, family=poisson, data=fp))
```

```
[1] 63.19565
```

Model $[PC, SC]$ assumes that only the crab salad is associated with food poisoning:

```
deviance(glm(freq ~ crab*potato+sick*crab, family=poisson, data=fp))
```

```
[1] 53.68259
```

Model $[PC, SP]$ assumes that only the potato salad is associated with food poisoning:

```
l0 <-glm(freq ~ crab*potato+sick*potato, family=poisson, data=fp)

deviance(l0)
```

```
[1] 6.481655
```

Model $[PC, SP, SC]$ assumes that both the potato salad and the crab salad are associated with food poisoning:

```
l1 <- glm(freq ~ crab*potato + crab*sick+potato*sick, family=poisson, data=fp)
summary(l1)
```

```
Call:
glm(formula = freq ~ crab * potato + crab * sick + potato * sick,
    family = poisson, data = fp)

Deviance Residuals:
       1         2         3         4         5         6         7         8
 0.12164  -0.21783  -0.19230   0.22947  -0.09857   0.23481   0.59995  -1.47176

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         3.0873     0.2102  14.686  < 2e-16 ***
crabyes             0.3811     0.2697   1.413   0.1577
potatoyes           0.1349     0.2837   0.476   0.6343
sickyes            -3.0075     0.5676  -5.299 1.17e-07 ***
crabyes:potatoyes   0.7651     0.3432   2.229   0.0258 *
crabyes:sickyes     0.6097     0.3170   1.923   0.0544 .
potatoyes:sickyes   2.8259     0.5362   5.270 1.36e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 295.2526  on 7  degrees of freedom
Residual deviance:   2.7427  on 1  degrees of freedom
AIC: 53.074

Number of Fisher Scoring iterations: 5
```

We see that the interaction term between `sick` and `crab` has a p-value of just above 0.05, suggesting that the relationship between crab salad and sickness is marginally significant. The potato salad has a much smaller p-value and looks like the most likely source of the outbreak, although there is some indication that the crab salad may have something to do with it, too.

In terms of how well the two models fit the data, we can look at their fitted values in comparison to the observed data:

```
cbind(fp$freq, round(fitted(l0),2), round(fitted(l1),2))
```

```
  [,1]   [,2]   [,3]
1   80  84.55  78.92
2   24  19.45  25.08
3   31  32.59  32.08
4   23  21.41  21.92
5  120 115.45 121.08
6   22  26.55  20.92
7    4   2.41   2.92
8    0   1.59   1.08
```

In general, we can tabulate the deviances and degrees of freedom from each of the possible models.

Terms in model	Deviance	DF
$[PC]$	63.669	4
$[PC, S]$	63.196	3
$[PC, SC]$	53.683	2
$[PC, SP]$	6.482	2
$[PC, SP, SC]$	2.743	1
$[PCS]$	4.123×10^{-10}	0

We see that the models in the first three rows have relatively large deviance indicating a poor fit, but that models $[PC, SP]$ and $[PC, SP, SC]$ fit the data well. (Notice though that in both models there are some small fitted values, so we shouldn't overinterpret the deviance as a measure of goodness of fit.) $[PCS]$ is the saturated model so it fits the data perfectly – notice the zero deviance.

The comparison between models $[PC, SP]$ and $[PC, SP, SC]$ is based on a $\chi^2(1)$ distribution:

```
6.482-2.743
```

```
[1] 3.739
```

```
qchisq(df=1,p=0.95)
```

```
[1] 3.841459
```

This is the same test that is performed when using

```
anova(l0,l1)
```

```
Analysis of Deviance Table

Model 1: freq ~ crab * potato + sick * potato
Model 2: freq ~ crab * potato + crab * sick + potato * sick
  Resid. Df Resid. Dev Df Deviance
1         2     6.4817
2         1     2.7427  1   3.7389
```

Task 9.

When one of the binary variables in a contingency table can be thought of as a response, it is possible to use a logistic regression model to test for association between this response and the other factors in the contingency table. Try this with the food poisoning data, by fitting logistic regression models corresponding to the loglinear models [PC,SP] and [PC,SP,SC]. Confirm that you reach the same conclusions and get the same fitted values.

Simpson's paradox

Simpson's paradox is a phenomenon in which associations get reversed when we look at aggregates. A very simple example of this is the following:

Suppose that I get better grades than you in easy courses and I also get better grades than you in hard courses, and yet your GPA is higher than mine. How can this be?

I take lots of hard courses where I get mostly Cs and you get mostly Ds, and you take lots of easy courses where you get mostly Bs and I get mostly As. Even though I do better than you when we control for the difficulty of the course,

your overall GPA will be higher.

Here is another example with a contingency table.

Example 7 (Death penalty in the US).

This dataset comes from *Categorical Data Analysis* by Alan Agresti. The $2 \times 2 \times 2$ table shows homicide cases in Florida over the period 1976-77. The defendant's race and victim's race, each having categories white or black, and whether there was a death penalty verdict (yes/no), was recorded.

Defendant's Race	Victim's Race	Death Penalty Yes	Death Penalty No	Percentage Yes
White	White	19	132	12.6
	Black	0	9	0
Black	White	11	52	17.5
	Black	6	97	5.8

The following is the marginal table obtained by summing the cell counts over the level's of victim's race.

Defendant's Race	Death Penalty Yes	Death Penalty No	Total	Percentage Yes
White	19	141	160	11.9
Black	17	149	166	10.2
Total	36	290	326	

About 12% of white defendants and about 10% of black defendants received the death penalty. Ignoring the victim's race, the percentage of "yes" death penalty verdicts was lower for blacks than for whites.

However taking the victim's race into account, things look completely different: When the victim was white, the death penalty was imposed about 5 percentage points more often for black defendants than for white defendants. When the victim was black, the death penalty was imposed over 5 percentage points more often for black defendants than for white defendants. Controlling for the victim's race, the percentage of "yes" death penalty verdicts was higher for blacks than for whites.

The phenomenon in which a pair of variables have marginal association of different direction from their partial associations is called *Simpson's paradox*. In the death penalty example, it arises because whites tended to kill whites, and killing a white person was more likely to result in the death penalty.

Task 10.

Fit a suitable loglinear model to the death penalty data and interpret it in terms of the associations between the variables.

Additional resources on count regression and loglinear models

Extending linear models with R: generalized linear, mixed effects and nonparametric regression models by Julian Faraway:

- Chapter 3 has more details and examples of count regression (Poisson, quasi-Poisson and negative binomial models).
- Chapter 4 has information on contingency tables and loglinear models.
- Chapter 6 discusses GLM diagnostics including residuals and goodness of fit statistics.

R examples on the following topics are available from UCLA's Institute for Digital Research and Education:

- Poisson regression
- Negative binomial regression
- Zero-inflated Poisson regression

- Zero-inflated negative binomial regression

Week 5 learning outcomes

- Be able to select an appropriate model for count responses (count regression for rates, loglinear model for contingency tables)
- Fit Poisson regression models for count data, using an offset when appropriate
- Fit negative binomial models for count responses using the `glm.nb()` function in `library(MASS)`, using an offset when appropriate
- Interpret Poisson and negative binomial model coefficients in terms of rate ratios
- Obtain fitted values and residuals from a Poisson regression or a negative binomial model
- Use deviance and Pearson residuals for diagnostic plots, recognising the limitations of these residuals
- Use the deviance and Pearson X^2 goodness of fit statistics to detect lack of fit in a GLM for counts, remembering that these tests only apply when the fitted values are large
- Recognise the signs of overdispersion in a Poisson model fit and be able to adjust the model output using a quasi-Poisson approach
- Identify the presence of excess zeros in a count regression and be able to explore a zero-inflated or hurdle model as an alternative to Poisson/negative binomial regression
- Choose between nested models by comparing deviances and between non-nested models by comparing AIC (or similar)
- Use loglinear models for contingency tables and be able to test hypotheses about independence between variables
- Recognise the relationship between loglinear models and logistic regression when one of the categorical variables in the contingency table is binary and can be thought of as a response
- Be familiar with Simpson's paradox: the direction of association can change if we aggregate over one of the variables in a three-way (or multi-way) contingency table

Answers to tasks

Answer to Task 1. We can either take $\exp(0.00335159) = 1.00336$ and interpret it as the rate ratio associated with one unit increase in the percentage of people who smoke or we can choose another percentage, *e.g.* 10%, so that we interpret $\exp(0.00335159 * 10) = 1.034$ as the rate ratio associated with an increase of 10 units in the percentage of people who smoke, or we can go with the standard deviation of `smoke`

```
sd(cancer$smoke)
```

```
[1] 9.637063
```

which in this case turns out to be quite similar. A point estimate and an approximate confidence interval for the rate ratio associated with one standard deviation increase in the percentage of people who smoke can be obtained as follows:

```
exp(0.00335159*sd(cancer$smoke)) # point estimate
```

```
[1] 1.032827
```

```
exp((0.00335159-1.96*0.0009462959)*sd(cancer$smoke)) # approx 95% CI lower limit
```

```
[1] 1.01453
```

```
exp((0.00335159+1.96*0.0009462959)*sd(cancer$smoke)) # approx 95% CI upper limit
```

```
[1] 1.051454
```

Answer to Task 2. We first take log of every variable with the exception of distance to Santa Cruz for which we have to take log of `Scruz` plus a small value because Santa Cruz has distance to Santa Cruz=0.

```
gal4<- glm(Species ~ log(Area)+ log(Elevation) + log(Nearest) +
            log(Scruz+0.1) + log(Adjacent),family=poisson,data = gala)
summary(gal4)
```

```
Call:
glm(formula = Species ~ log(Area) + log(Elevation) + log(Nearest) +
    log(Scruz + 0.1) + log(Adjacent), family = poisson, data = gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.4479  -2.6717  -0.4547   2.5613   8.2970

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.287941   0.284661  11.550  < 2e-16 ***
log(Area)         0.348445   0.018029  19.327  < 2e-16 ***
log(Elevation)    0.036421   0.056983   0.639  0.52272
log(Nearest)     -0.040644   0.013781  -2.949  0.00318 **
log(Scruz + 0.1) -0.030045   0.010492  -2.864  0.00419 **
log(Adjacent)    -0.089014   0.006948 -12.812  < 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: 3510.73  on 29  degrees of freedom
Residual deviance:  359.12  on 24  degrees of freedom
AIC: 531.96

Number of Fisher Scoring iterations: 5
```

In the Poisson model there is a substantial reduction in deviance when using the log-transformed variables. Also `log(Elevation)` does not appear to be significant. We can drop the term for elevation from the model:

```
gal5 <- glm(Species ~ log(Area) + log(Nearest) +
            log(Scruz+0.1) + log(Adjacent),family=poisson, data = gala)
summary(gal5)
```

```
Call:
glm(formula = Species ~ log(Area) + log(Nearest) + log(Scruz +
    0.1) + log(Adjacent), family = poisson, data = gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.3457  -2.7891  -0.6233   2.5129   8.1217

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.466484   0.053704  64.549  < 2e-16 ***
log(Area)         0.358711   0.008254  43.460  < 2e-16 ***
log(Nearest)     -0.041117   0.013733  -2.994  0.00275 **
log(Scruz + 0.1) -0.030098   0.010478  -2.873  0.00407 **
log(Adjacent)    -0.088224   0.006842 -12.895  < 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: 3510.73  on 29  degrees of freedom
Residual deviance:  359.54  on 25  degrees of freedom
AIC: 530.37

Number of Fisher Scoring iterations: 5
```

Since there are still signs of overdispersion, we can estimate the dispersion parameter and use it in a quasi-Poisson model:

```
dp <- sum(residuals(gal5,type="pearson")^2)/gal5$df.res #dispersion parameter
summary(gal5, dispersion=dp) # update standard errors
```

```
Call:
glm(formula = Species ~ log(Area) + log(Nearest) + log(Scruz +
    0.1) + log(Adjacent), family = poisson, data = gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.3457  -2.7891  -0.6233   2.5129   8.1217

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.46648    0.21366  16.224  < 2e-16 ***
log(Area)         0.35871    0.03284  10.924  < 2e-16 ***
log(Nearest)     -0.04112    0.05463  -0.753  0.45171
log(Scruz + 0.1) -0.03010    0.04169  -0.722  0.47029
log(Adjacent)    -0.08822    0.02722  -3.241  0.00119 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 15.82846)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  359.54  on 25  degrees of freedom
AIC: 530.37

Number of Fisher Scoring iterations: 5
```

It looks like more terms can be dropped. When dropping terms, make sure you don't drop more than one at a time.

```
drop1(gal5, test="F")
```

```
Single term deletions
```

```
Model:
Species ~ log(Area) + log(Nearest) + log(Scruz + 0.1) + log(Adjacent)
                 Df Deviance    AIC  F value    Pr(>F)
<none>                359.5  530.4
log(Area)         1   3266.1 3434.9 202.1022 1.752e-13 ***
log(Nearest)      1    368.5  537.3   0.6218   0.43780
log(Scruz + 0.1)  1    367.7  536.6   0.5700   0.45732
log(Adjacent)     1    528.6  697.4  11.7563   0.00211 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It looks like distance to Santa Cruz can be dropped.

Repeating the process we see that distance to nearest neighbour can also be dropped and the final model is one with terms for the logarithm of the area of the island and the logarithm of the area of the adjacent island.

Answer to Task 3. Start by fitting a quasi-Poisson model following the analysis steps for the Galapagos data:

```
epid2 <- glm(Y_all ~ pm10 + smoke + ethnic + log.price + easting +
                     northing+offset(log(E_all)), family=quasipoisson(link = "log"),
                     data = cancer)
summary(epid2)
```

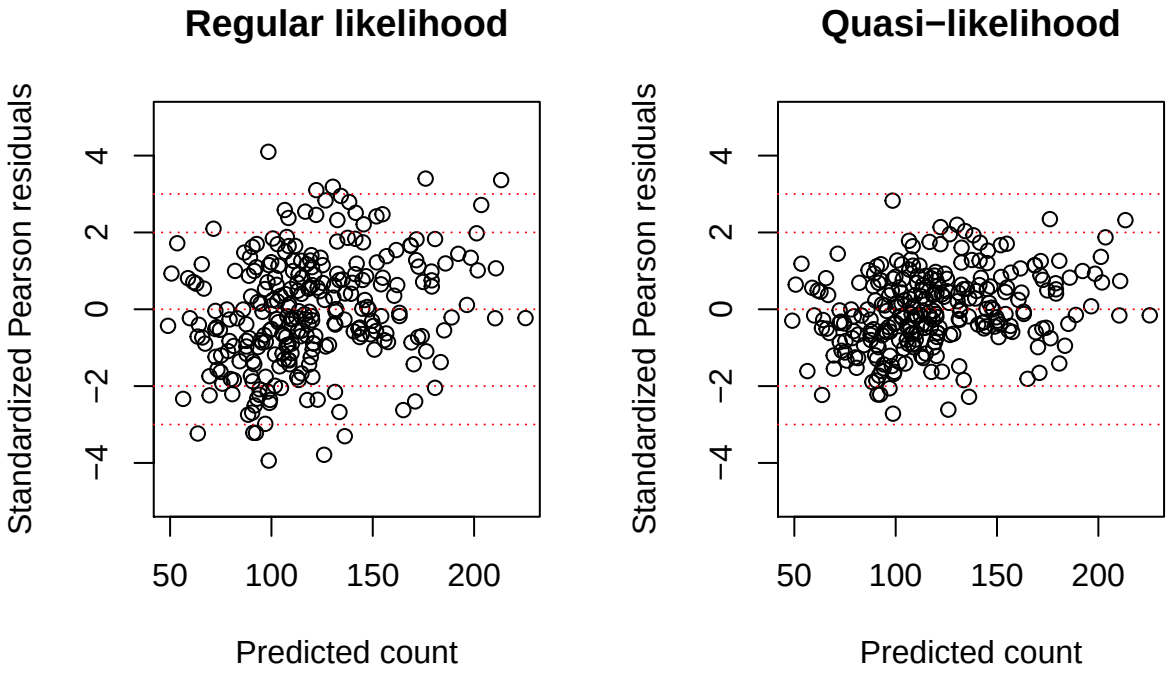
```
Call:
glm(formula = Y_all ~ pm10 + smoke + ethnic + log.price + easting +
    northing + offset(log(E_all)), family = quasipoisson(link = "log"),
    data = cancer)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2011  -0.9338  -0.1763   0.8959   3.8416

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.8592657  1.1639164  -0.738   0.4610
pm10         0.0500269  0.0096725   5.172 4.58e-07 ***
smoke        0.0033516  0.0013718   2.443   0.0152 *
ethnic      -0.0049388  0.0009211  -5.362 1.80e-07 ***
log.price   -0.1034461  0.0246356  -4.199 3.67e-05 ***
easting     -0.0331305  0.0150324  -2.204   0.0284 *
northing     0.0300213  0.0160928   1.866   0.0632 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 2.101436)

    Null deviance: 972.94  on 270  degrees of freedom
Residual deviance: 565.18  on 264  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

```
# Calculate the dispersion parameter:
pearson <- residuals(epid1, type = "pearson")
sum(pearson^2)/epid1$df.residual
```

```
[1] 2.101436
```

```
# Residual plots vs. predicted (using standardised residuals):
pred <- predict(epid1, type = "response")
stand.resid <- rstandard(model = epid1, type = "pearson")

par(mfrow=c(1,2))
plot(x = pred, y = stand.resid, xlab = "Predicted count", ylab = "Standardized Pearson residuals",
```

```
   main = "Regular likelihood", ylim = c(-5,5))
abline(h = c(-3, -2, 0, 2, 3), lty = "dotted", col = "red")

pred <- predict(epid2, type = "response")
stand.resid <- rstandard(model = epid2, type = "pearson")

plot(x = pred, y = stand.resid, xlab = "Predicted count", ylab = "Standardized Pearson residuals",
   main = "Quasi-likelihood", ylim = c(-5,5))
abline(h = c(-3, -2, 0, 2, 3), lty = "dotted", col = "red")
```



We see that for the cancer data, just like for the Galapagos data, all of the residuals are contained within ± 3 in the quasi-Poisson model, while quite a few are outside this range for the original Poisson model.

Next we fit a negative binomial model:

```
library(MASS)
epid3 <- glm.nb(Y_all ~ pm10 + smoke + ethnic + log.price + easting +
                      northing+offset(log(E_all)),  data = cancer)
summary(epid3)
```

```
Call:
glm.nb(formula = Y_all ~ pm10 + smoke + ethnic + log.price +
    easting + northing + offset(log(E_all)), data = cancer, init.theta = 108.8173108,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1490  -0.6290  -0.0581   0.6381   2.7392

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.6532266  1.1527051  -0.567   0.5709
pm10         0.0487285  0.0096957   5.026 5.01e-07 ***
smoke        0.0029701  0.0013652   2.176   0.0296 *
ethnic      -0.0046392  0.0008794  -5.275 1.33e-07 ***
log.price   -0.1143346  0.0246668  -4.635 3.57e-06 ***
easting     -0.0316406  0.0149626  -2.115   0.0345 *
northing     0.0286009  0.0159557   1.793   0.0730 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

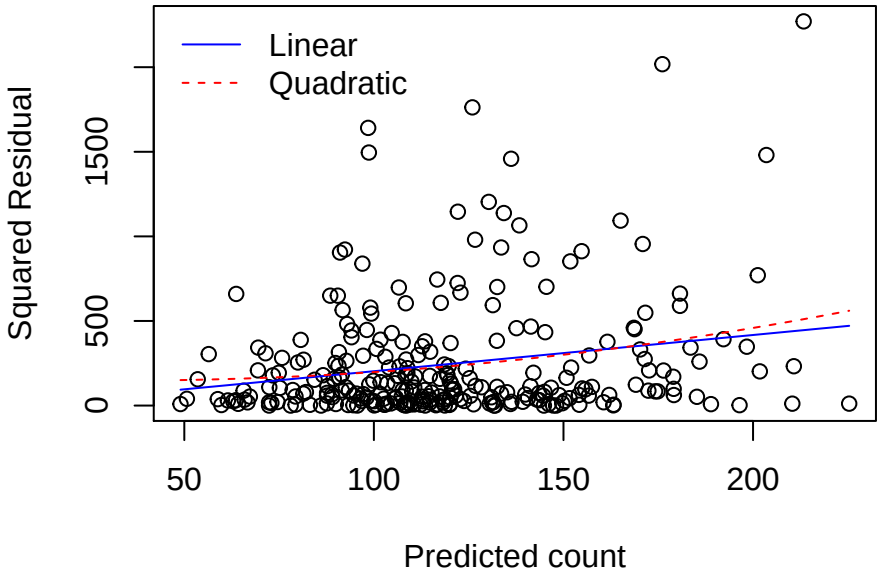
```
(Dispersion parameter for Negative Binomial(108.8173) family taken to be 1)

    Null deviance: 472.65  on 270  degrees of freedom
Residual deviance: 279.68  on 264  degrees of freedom
AIC: 2269

Number of Fisher Scoring iterations: 1

              Theta:  108.8
          Std. Err.:  18.7

 2 x log-likelihood:  -2252.955
```

To decide between the quasi-Poisson and the negative binomial model, we plot $(y_i - \hat{\mu}_i)^2$ v $\hat{\mu}_i$ and compare the linear and quadratic fit to see which one of them fits better.

```
# Plot of squared residuals v predicted
res.sq <- residuals(epid1, type = "response")^2
set1 <- data.frame(res.sq, mu.hat = epid1$fitted.values)

fit.lin <- lm(formula = res.sq ~ mu.hat, data = set1)
fit.quad <- lm(formula = res.sq ~ mu.hat + I(mu.hat^2), data = set1)
summary(fit.quad)
```

```
Call:
lm(formula = res.sq ~ mu.hat + I(mu.hat^2), data = set1)

Residuals:
    Min      1Q  Median      3Q     Max
-549.25 -196.40 -124.92   82.03 1760.17

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 159.91559  221.07240   0.723    0.470
mu.hat       -0.73615    3.54642  -0.208    0.836
I(mu.hat^2)   0.01115    0.01354   0.824    0.411

Residual standard error: 349.3 on 268 degrees of freedom
Multiple R-squared:  0.04612,   Adjusted R-squared:  0.039
F-statistic: 6.479 on 2 and 268 DF,  p-value: 0.001787
```

```
plot(set1$mu.hat, y = set1$res.sq, xlab = "Predicted count",
     ylab = "Squared Residual")
curve(expr = predict(fit.lin, newdata = data.frame(mu.hat = x), type = "response"),
      col = "blue", add = TRUE, lty = "solid")
curve(expr = predict(fit.quad, newdata = data.frame(mu.hat = x), type = "response"),
      col = "red", add = TRUE, lty = "dashed")
legend("topleft", legend = c("Linear", "Quadratic"), col = c("blue","red"),
    lty = c("solid", "dashed"), bty = "n")
```



The straight line fit (blue line) would indicate that the quasi-Poisson model should be preferred, and the quadratic fit (red line) would suggest using a negative binomial model.

The quadratic coefficient in the above model has a p-value of 0.411 – not significant. We don't have any evidence that the negative binomial model is better and we can probably go with either one.

Answer to Task 4. Estimate the dispersion parameter:

```
X2 <- sum(resid(ol1, type = "pearson")^2)
X2
```

```
[1] 602.8932
```

```
dp <- X2/ol1$df.res
dp
```

```
[1] 7.352356
```

Refit the model using the dispersion parameter:

```
summary(ol1, dispersion= dp)
```

```
Call:
glm(formula = medals ~ log(population) + log(GDPpercapita), family = poisson,
    data = olympics)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.0028  -2.2661  -0.3188   1.1572   8.2493

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -9.1125     1.0811  -8.429  < 2e-16 ***
log(population)    0.5948     0.0549  10.835  < 2e-16 ***
log(GDPpercapita)  0.4976     0.0806   6.173 6.69e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 7.352356)

    Null deviance: 1567.7  on 84  degrees of freedom
Residual deviance:  547.5  on 82  degrees of freedom
AIC: 849.33

Number of Fisher Scoring iterations: 5
```

```
drop1(ol1, type="F")
```

```
Single term deletions

Model:
medals ~ log(population) + log(GDPpercapita)
                  Df Deviance     AIC
<none>                 547.50  849.33
log(population)    1  1441.10 1740.93
log(GDPpercapita)  1   867.52 1167.34
```

The parameter estimates (and fitted values) are the same, and although the standard errors change, both predictors are still highly significant.

Answer to Task 5. A negative binomial model can be fit using

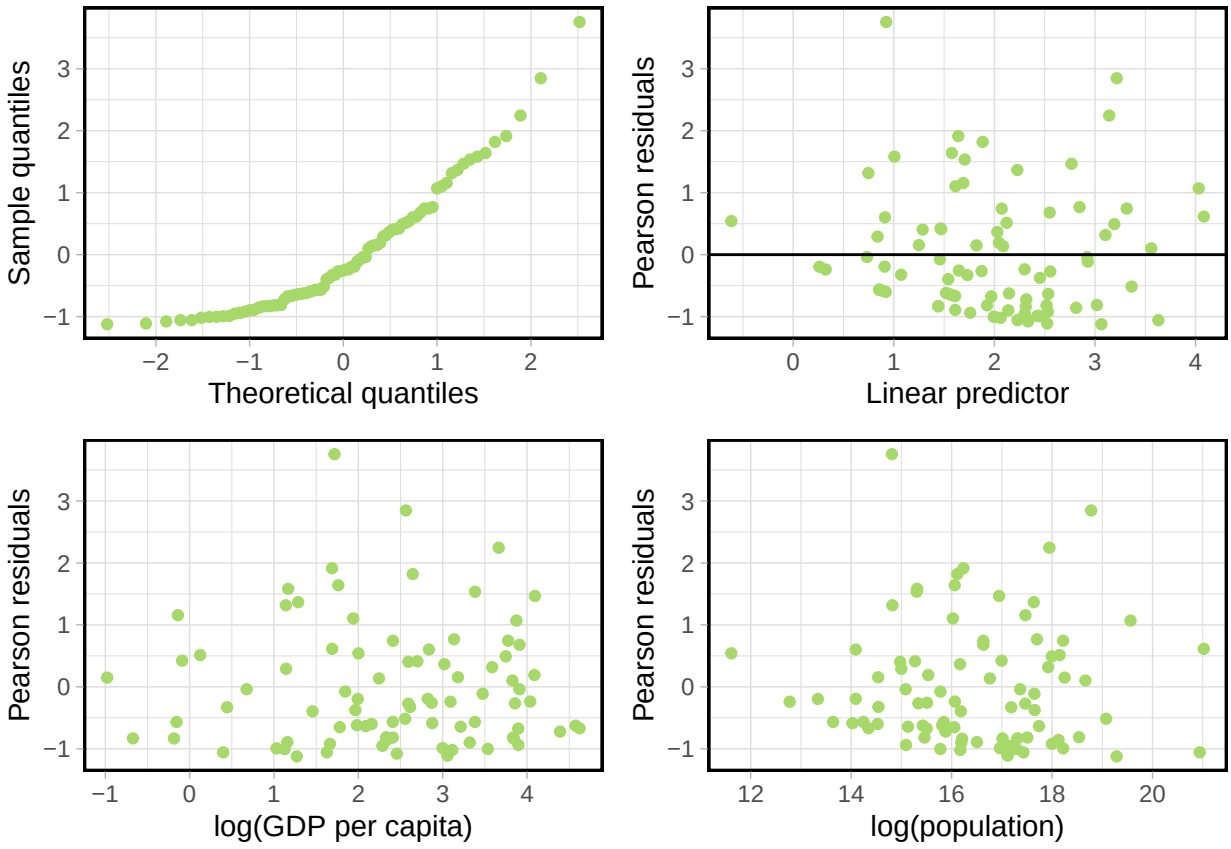
```
library(MASS)
ol2 <- glm.nb(medals ~ log(population) + log(GDPpercapita), data=olympics)
summary(ol2)
```

```
Call:
glm.nb(formula = medals ~ log(population) + log(GDPpercapita),
    data = olympics, init.theta = 1.65008023, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9940  -1.1049  -0.2763   0.4575   2.2539

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -7.17156    1.07756  -6.655 2.83e-11 ***
log(population)    0.50984    0.06078   8.389  < 2e-16 ***
log(GDPpercapita)  0.31739    0.07645   4.152 3.30e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.6501) family taken to be 1)

    Null deviance: 190.561  on 84  degrees of freedom
Residual deviance:  85.484  on 82  degrees of freedom
AIC: 522.95

Number of Fisher Scoring iterations: 1

              Theta:  1.650
          Std. Err.:  0.300

 2 x log-likelihood:  -514.946
```

Answer to Task 6. Pearson residual plots for this model:

```
res <- resid(ol2, type = "pearson")
d2 <- data.frame(res=resid(ol2, type = "pearson"))

p3 <- ggplot(d2, aes(sample=res)) + geom_point(stat="qq",col="#a6d96a") +
    xlab("Theoretical quantiles") + ylab("Sample quantiles")

p4 <- ggplot(d2, aes(x=predict(ol2, type="link"), y=res)) +
        geom_point(col="#a6d96a") + geom_hline(yintercept = 0) +
        xlab("Linear predictor") + ylab("Pearson residuals")

p5 <- ggplot(d2, aes(x=log(olympics$GDPpercapita), y=res)) +
```

```
        geom_point(col="#a6d96a") + xlab("log(GDP per capita)") +
        ylab("Pearson residuals")

p6 <- ggplot(d2, aes(x=log(olympics$population), y=res)) +
        geom_point(col="#a6d96a") + xlab("log(population)") +
        ylab("Pearson residuals")

grid.arrange(p3, p4, p5, p6, nrow = 2)
```



The normal probability plot shows some large positive residuals. No patterns are seen in the plot of residuals against the linear predictor, and there is no obvious nonlinear pattern in the plots of residuals against the explanatory variables.

We can also look at the cases where the model does not fit the data better (these correspond to large positive or negative residuals).

```
pres.n2 <- resid(ol2, type = "pearson")
expmedals2 <- round(fitted(ol2)[abs(pres.n2) > 1], 2)
cbind(olympics[abs(pres.n2) > 1, 1:2], expmedals2)
```

```
          country medals expmedals2
2         Jamaica     12       2.52
4     New Zealand     13       5.50
7        Mongolia      5       2.11
8         Hungary     17       6.58
11        Georgia      7       2.73
13      Australia     35      15.92
16        Belarus     12       4.84
17           Cuba     14       5.16
21     Azerbaijan     10       5.02
23 Great Britain     65      23.14
34         Russia     82      24.93
46        Ukraine     20       9.29
49 United States    104      56.38
51          Kenya     11       5.42
62      Hong Kong      1       7.34
```

```
67      Portugal      1       7.86
78  Saudi Arabia      1      12.49
79     Venezuela      1      10.33
80       Morocco      1       7.42
82       Algeria      1       9.31
84     Indonesia      2      21.43
85         India      6      37.73
```

Here we see that the model does a poor job of predicting strong Olympic performances by countries such as the USA, UK and China, and not so strong performances by countries such as India and Indonesia.

Answer to Task 7. Clearly there is more to winning Olympic medals than just the country's population and GDP per capita. The large residuals from the negative binomial model show that this is the case. Adjusting for overdispersion does not improve predictive performance in general, and may not be as useful as including additional predictors in the model.

Answer to Task 8. A zero-inflated Poisson or negative binomial model might be appropriate for simultaneously considering

- the probability of a country winning a medal in the 2012 Olympics (using a logit model), and
- the number of medals won (using a Poisson or negative binomial model).

In practice such a model could be fit using function `zero.infl()` from `library(pscl)`.

Answer to Task 9. First read in the data in a slightly different format:

```
fp2 <- data.frame(sick=c(120,22,4,0), not.sick=c(80,24,31,23),
                  potato=c("yes","yes","no","no"),crab= c("yes","no","yes","no"))
fp2
```

```
  sick not.sick potato crab
1  120       80    yes  yes
2   22       24    yes   no
3    4       31     no  yes
4    0       23     no   no
```

This is the logistic regression model corresponding to $[PC, SP]$:

```
s0 <- glm(cbind(sick,not.sick) ~ potato, data=fp2, family=binomial)
summary(s0)
```

```
Call:
glm(formula = cbind(sick, not.sick) ~ potato, family = binomial,
    data = fp2)

Deviance Residuals:
      1       2       3       4
 0.6534  -1.3494   0.9731  -1.8130

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.6027     0.5182  -5.023 5.10e-07 ***
potatoyes     2.9141     0.5340   5.457 4.84e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 63.1957  on 3  degrees of freedom
Residual deviance:  6.4817  on 2  degrees of freedom
AIC: 23.628

Number of Fisher Scoring iterations: 5
```

This is the logistic regression model corresponding to $[PC, SP, SC]$:

```
s1 <- glm(cbind(sick,not.sick) ~ potato+crab, data=fp2, family=binomial)
summary(s1)
```

```
Call:
glm(formula = cbind(sick, not.sick) ~ potato + crab, family = binomial,
    data = fp2)

Deviance Residuals:
      1       2       3       4
-0.1566  0.3203  0.6300 -1.4895

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.0075     0.5676  -5.299 1.17e-07 ***
potatoyes    2.8259     0.5362   5.271 1.36e-07 ***
crabyes      0.6097     0.3170   1.923   0.0544 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 63.1957  on 3  degrees of freedom
Residual deviance:  2.7427  on 1  degrees of freedom
AIC: 21.889

Number of Fisher Scoring iterations: 4
```

The deviance comparison between the two is the same as the comparison between the corresponding nonlinear models:

```
anova(s0,s1)
```

```
Analysis of Deviance Table

Model 1: cbind(sick, not.sick) ~ potato
Model 2: cbind(sick, not.sick) ~ potato + crab
  Resid. Df Resid. Dev Df Deviance
1         2     6.4817
2         1     2.7427  1   3.7389
```

The fitted values are the same as those of the corresponding loglinear models:

```
cbind(c(fp2$sick, fp2$not.sick), round(fitted(l0),2), round(fitted(l1),2))
```

```
  [,1]   [,2]   [,3]
1  120  84.55  78.92
2   22  19.45  25.08
3    4  32.59  32.08
4    0  21.41  21.92
5   80 115.45 121.08
6   24  26.55  20.92
7   31   2.41   2.92
8   23   1.59   1.08
```

Answer to Task 10. Let P stand for death penalty, V for victim's race and D for defendant's race. Create a data.frame with the data:

```
deathpenalty <- data.frame (D=rep(c("white","white","black","black"),2),
                            V=rep(c("white","black"),4),
                            P=c(rep("yes",4),rep("no",4)),
                            freq = c(19,0,11,6,132,9,52,97))

xtabs(freq ~ D + V+ P, data=deathpenalty)
```

```
, , P = no

       V
D       black white
  black    97    52
  white     9   132

, , P = yes

       V
D       black white
  black     6    11
  white     0    19
```

Start from fitting

```
glm(freq ~ D + V+ P, family=poisson, data=deathpenalty) #[D,V,P]
```

```
Call:  glm(formula = freq ~ D + V + P, family = poisson, data = deathpenalty)

Coefficients:
(Intercept)       Dwhite       Vwhite         Pyes
    3.92657     -0.03681      0.64748     -2.08636

Degrees of Freedom: 7 Total (i.e. Null);  4 Residual
Null Deviance:      395.9
Residual Deviance: 137.9    AIC: 181.6
```

and try all models up to the saturated model

```
glm(freq ~ D*V*P, family=poisson, data=deathpenalty) #[DVP]
```

```
Call:  glm(formula = freq ~ D * V * P, family = poisson, data = deathpenalty)

Coefficients:
       (Intercept)              Dwhite              Vwhite                Pyes       Dwhite:Vwhite
            4.5747             -2.3775             -0.6235             -2.7830              3.3090
       Dwhite:Pyes         Vwhite:Pyes  Dwhite:Vwhite:Pyes
          -21.7169              1.2296             21.3318

Degrees of Freedom: 7 Total (i.e. Null);  0 Residual
Null Deviance:      395.9
Residual Deviance: 4.122e-10    AIC: 51.68
```

A summary of all possible models is given in the following table:

Terms in the model	DF	Deviance
$[D, V, P]$	4	137.93
$[D, VP]$	3	131.68
$[V, DP]$	3	137.71
$[P, DV]$	3	8.13
$[DP, VP]$	2	131.46
$[DP, DV]$	2	7.91
$[VP, DV]$	2	1.88
$[DP, VP, DV]$	1	0.70
$[DPV]$	0	0

Notice that any model that does not include the term `D*V` has a large deviance. This suggests an important association between the defendant's race and the victim's race. Other than the saturated model, the two models that appear to fit the data well when we compare the deviance with a chi-squared distribution with the corresponding degrees of freedom, are $[VP, DV]$ and $[DP, VP, DV]$. The simpler model says that the death penalty verdict is independent of the defendant's race, given the victim's race. In the model with all two-way interactions, all pairs of variables are conditionally dependent.