of *iteratively reweighted least squares regressions* (the *IRLS algorithm*). We do without giving further details, but instead focus on the practical application.

```
> glm(survival~log(weight)+age, family="binomial", data=baby)

Coefficients:
(Intercept)  log(weight)       age
   -33.9711       4.4161    0.1474
```

This is only a part of the output, but for the moment the most interesting one, namely the estimated coefficients $\hat{\beta}_0, \hat{\beta}_1$ and $\hat{\beta}_2$.

4.3 Poisson Regression

The method of Poisson Regression is the formally correct way of dealing with response variables that are counts. This is a strong statement, given the fact that throughout the discussion on simple linear regression, this scriptum uses the count variable *number of passengers* as a response. While being contradictory to some extent, it is tolerable: the counts are all large (*in the millions*) and their range is relatively small. In such a situation, we can profit from the fact that the approximation of the Poisson to the Gaussian distribution works well, i.e. there is little difference between the two. However, there are examples where the use of Poisson Regression is a must:

- *if the size of the population is unknown and the counts are small.*
- *if the size of the population is large and hard to come by, and the probability of an event, and thus the expected counts are small.*

A typical example for the latter case is modeling the incidence of rare forms of cancer in a given geographical area. For illustrating the former case, we will consider the following example.

4.3.1 Example: Tortoise Species on Galapagos

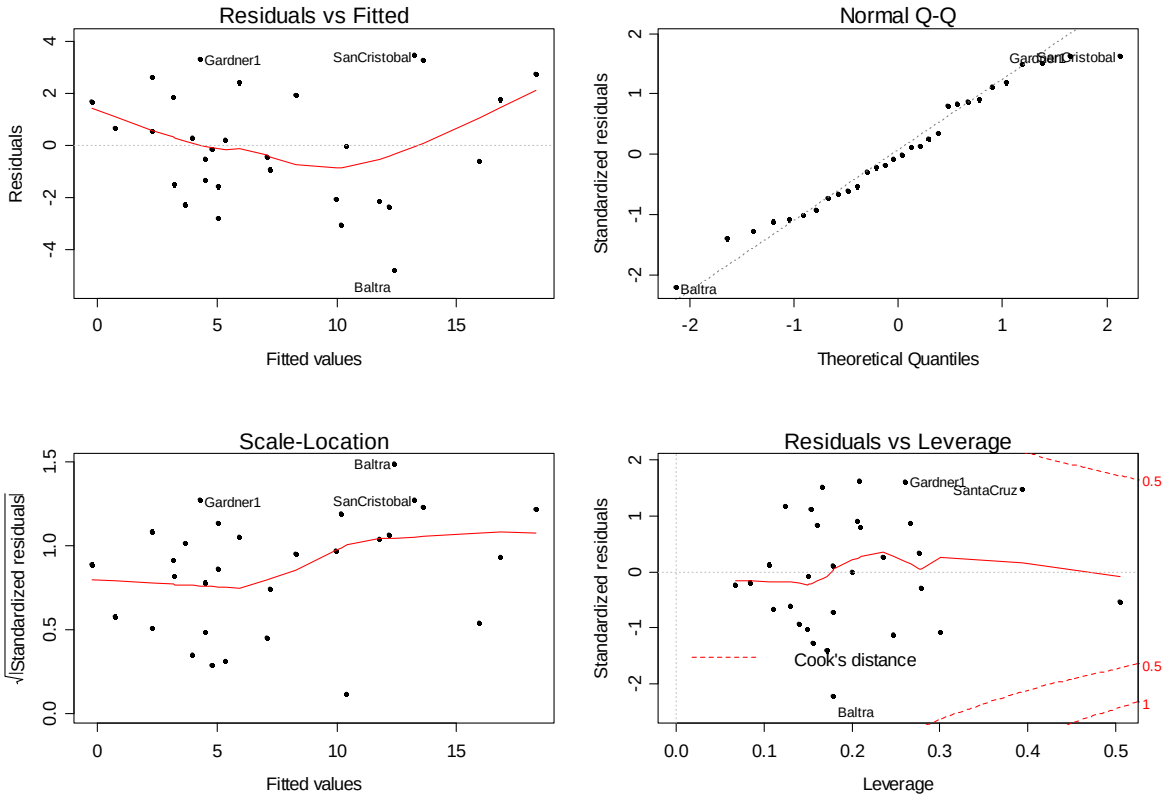
For 30 of Galapagos Islands' we have the response variable *Species*, i.e. the number of species of tortoise that are present, plus five geographic predictors. These are the area of the island, the highest elevation, the distance to the nearest island, the distance to Santa Cruz and the area of the adjacent island.

```
> library(faraway); data(gala); head(gala[,-2])
            Species  Area Elevation Nearest Scruz Adjacent
Baltra           58 25.09       346     0.6   0.6     1.84
Bartolome        31  1.24       109     0.6  26.3   572.33
Caldwell          3  0.21       114     2.8  58.7     0.78
Champion         25  0.10        46     1.9  47.4     0.18
Coamano           2  0.05        77     1.9   1.9   903.82
Daphne.Major     18  0.34       119     8.0   8.0     1.84
```

Fitting a multiple linear regression without doing any transformations yields very poor results: there is a bias, the error distribution is long-tailed and has non-constant variance. Additionally, there are some strong leverage points. We leave generating these diagnostic plots *as an exercise*. The first-aid transformations from section 2.6.7 suggest that all predictors, which can only take positive values and are strongly skewed to the right, need to be log-transformed. Since *Scruz* has a zero entry, we add its smallest positive value. The response, which is a count, is due for taking the square root. The output is:

```
> fit02 <- lm(sqrt(Species) ~ log(Area) + log(Elevation) +
              log(Nearest) + I(log(Scruz+0.4)) +
              log(Adjacent), data=gala[,-2])
> summary(fit02)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           9.8839     5.0523   1.956   0.0622 .
log(Area)             1.5609     0.3025   5.160 2.77e-05 ***
log(Elevation)       -0.4440     0.9777  -0.454   0.6538
log(Nearest)         -0.5273     0.3495  -1.509   0.1444
I(log(Scruz + 0.4))  -0.3989     0.3267  -1.221   0.2339
log(Adjacent)        -0.2737     0.1387  -1.973   0.0602 .
---
Residual standard error: 2.382 on 24 degrees of freedom
Multiple R-squared: 0.8398, Adjusted R-squared: 0.8064
F-statistic: 25.15 on 5 and 24 DF, p-value: 8.176e-09
```



While error distribution/variance are acceptable (not perfect though!) and there are no leverage points anymore, there is a bias. The square root transformation on the response is somewhat too strong. A better fitting relation can be obtained by taking a log-transformation on *Species* instead of the square root. Better yet is to deal correctly with the situation: due to low counts in the response, the Gaussian approximation is questionable. Thus, it is better to use Poisson Regression.

4.3.2 Model and Estimation

We have count responses Y_i for which we, given the predictors, assume a Poisson distribution with parameter λ_i, i.e. $Y_i \mid X \sim Pois(\lambda_i)$. Our goal is to relate the parameter to the predictors, and because λ_i can take positive values only, we will employ the log as a link function:

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}$$

Because the conditional expectation is $E[Y_i \mid X] = \lambda_i$, we again have the previously recognized situation that a link functions opens the door to modeling the expected value of the conditional distribution of Y_i by the linear predictor. For estimating the coefficients $\beta_0,...,\beta_p$, we again employ the MLE principle. By assuming independence of the cases, the likelihood function can be written as the product of the marginal distributions:

$$P(Y_1 = y_1,...,Y_n = y_n \mid X) = \prod_{i=1}^{n} P(Y_i = y_i \mid X) = \prod_{i=1}^{n} \frac{\lambda_i^{y_i} \cdot e^{-\lambda_i}}{y_i!}$$

The parameters and predictors enter the above equation by replacing λ_i with $\exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_p)$. The goal is to maximize the likelihood. In the multiplicative form, this is inconvenient, thus we again employ the trick of changing to the log-likelihood:

$$l(\beta) = \sum_{i=1}^{n} \left(y_i \cdot \log(\lambda_i) - \lambda_i - \log(y_i!) \right)$$

For finding the optimum, we take partial derivatives with respect to $\beta_0,...,\beta_p$. As usual for GLMs, this results in a non-linear equation system. There is no closed form solution and we have to resort to the *iteratively reweighted least squares* (IRLS) approach for an approximation. This is already implemented in R, and we can conveniently fit the Poisson Regression model to the Galapagos data:

```
> fit <- glm(Species ~ log(Area) + log(Elevation) +
             log(Nearest) + I(log(Scruz+0.4)) +
             log(Adjacent), data=gala, family=poisson)
```

Again, be aware of the fact that estimating the coefficients is based on numerical optimization. Should a warning be given that the algorithm did not converge, it is for a reason and to be taken seriously. The summary output is as follows:

```
> summary(fit)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          3.323245   0.286430  11.602  < 2e-16 ***
log(Area)            0.350370   0.018005  19.459  < 2e-16 ***
log(Elevation)       0.033108   0.057034   0.580  0.56158
log(Nearest)        -0.040153   0.014071  -2.854  0.00432 **
I(log(Scruz + 0.4)) -0.035848   0.013207  -2.714  0.00664 **
log(Adjacent)       -0.089452   0.006944 -12.882  < 2e-16 ***
---
    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  359.94  on 24  degrees of freedom
AIC: 532.77
```

4.3.3 Diagnostics and Inference

As for Binomial Regression, there is a quick check for model adequacy: if the residual deviance is far in excess of the degrees of freedom in the model, there is too much variation in the response. More precisely, we have that:

$$D = 2\sum_{i=1}^{n}\left[y_i \log\left(\frac{y_i}{\hat{\lambda}_i}\right) - (y_i - \hat{\lambda}_i)\right] \sim \chi^2_{n-(p+1)}$$

In our example, the residual deviance is 359.94 on just 24 degrees of freedom. This is far in excess of the degrees of freedom. We can compute the p-value for the null hypothesis that the fitted model is correct:

```
> pchisq(deviance(fit), df.residual(fit), lower=FALSE)
[1] 1.185031e-61
```

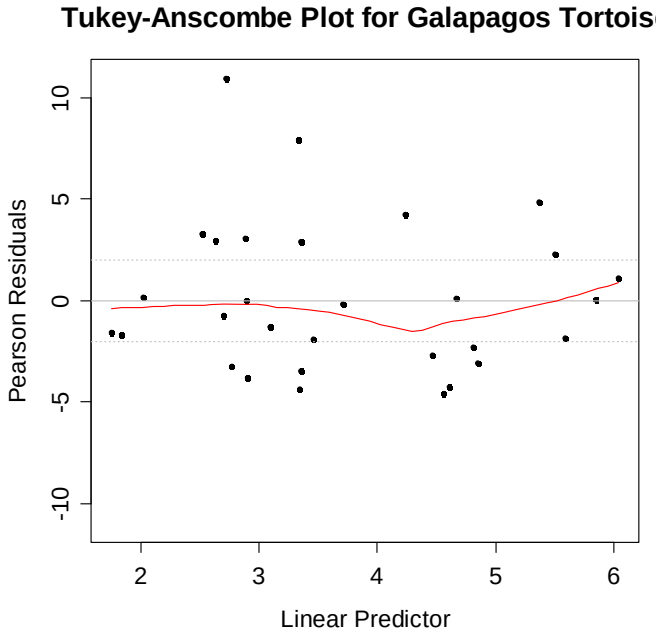
The p-value is very small, indicating that we have an ill-fitting model if a Poisson distribution for the response is correct. Our next job is to recognize why and where the fit is poor. This is done using some diagnostic visualization. We start with the Tukey-Anscombe plot, where either the deviance residuals or the Pearson residuals are plotted vs. the linear predictor. We go in line with R and use the latter, which are defined as:

$$P_i = \frac{(y_i - \hat{\lambda}_i)}{\sqrt{\hat{\lambda}_i}}$$

The Pearson residuals P_i approximately follow a standard normal distribution. This suggests that cases with $|P_i| > 2$ are extraordinary, i.e. show bigger residuals than the Poisson model suggests.

```
> xx <- predict(fit, type="link")
> yy <- resid(fit, type="pearson")
> plot(xx, yy, main=”Tukey-Anscombe Plot...”)
```

```
> smoo <- loess.smooth(xx, yy)
> abline(h=0, col="grey")
> lines(smoo, col="red")
```



We observe that there is hardly a bias, thus the functional form might be correct. However, there most of the residuals are larger than the Poisson distribution suggests. This is most likely due to the fact that our predictors are too simple: we just take distance and area of the nearest island into account, which does not reflect clusters of islands in close vicinity well.

In the present situation, where the predictor-response relation is correct, but the variance assumption of the Poisson distribution is broken, the point estimates for $\hat{\beta}_0,...,\hat{\beta}_p$ and thus $\hat{\lambda}_i$ are unbiased, but the standard errors will be wrong. This makes the model still suitable for prediction, though the inference results are in question, i.e. we cannot say which variables are statistically significant, because the p-values from the summary output are flawed.

Because the Poisson distribution has only one single parameter, it is not very flexible for empirical fitting purposes. This can be cured by estimating the dispersion parameter, which is in turn used for generating better standard errors. The estimate is defined as:

$$\hat{\phi} = \frac{\sum_i (y_i - \hat{\lambda}_i)^2 / \hat{\lambda}_i}{n - (p+1)}$$

This is the sum of squared Pearson residuals, divided by the degrees of freedom in this model. In R, the command is:

```
> sum(resid(fit, type="pearson")^2)/df.residual(fit)
[1] 16.64651
```

An alternative estimate for the dispersion parameter would be to take the quotient of the residual deviance and the degrees of freedom. Indeed, the result is similar with $\hat{\phi} \approx 15$.

```
> summary(fit, dispersion=16.64651)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          3.32325    1.16864   2.844  0.00446 **
log(Area)            0.35037    0.07346   4.769 1.85e-06 ***
log(Elevation)       0.03311    0.23270   0.142  0.88686
log(Nearest)        -0.04015    0.05741  -0.699  0.48430
I(log(Scruz + 0.4)) -0.03585    0.05389  -0.665  0.50589
log(Adjacent)       -0.08945    0.02833  -3.157  0.00159 **
---
Dispersion parameter for poisson family: 16.647
Null deviance: 3510.73 on 29 degrees of freedom
Residual deviance: 359.94 on 24 degrees of freedom
AIC: 532.77
```

Note that the estimation of dispersion and regression parameters are independent, thus modifying the dispersion parameter does not change the coefficients. In our example, some of the predictors now turn out to be non-significant. Also, the p-values are now not too different to the ones from the multiple linear regression model. For the mathematically interested, please note that since the Gaussian distribution has two parameters, the dispersion parameter is naturally estimated in multiple linear regression with the residual standard error.