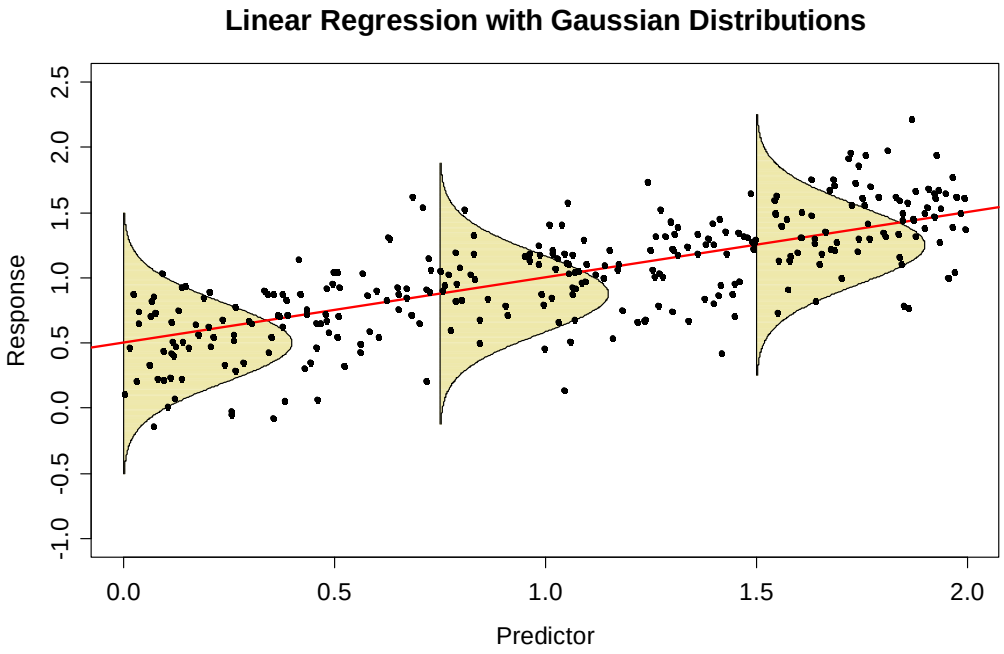
4 Extending the Linear Model

Linear models are central to the practice of statistics and can be seen as part of the core knowledge of any applied statistician. While they are very versatile, there are situations that cannot be handled within the standard framework. Here, we will take care of some of these.

4.1 What is the Problem?

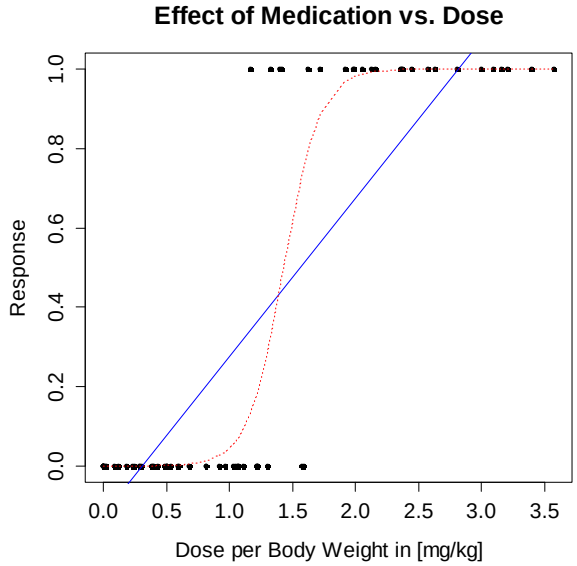
In all our previous theory, the response y_i was assumed to be a continuous random variable whose range was (at least theoretically) reaching from minus to plus infinity. The principal goal was to estimate and predict the conditional expectation, i.e. $\hat{y}_i = E[y_i \mid X_i]$ from the data. All theory, algorithms, tests and confidence intervals operated under the assumption that the conditional distribution was Gaussian, i.e. $y_i \mid X_i \sim N(\hat{y}_i, \sigma_E^2)$. The figure below shows the density of these in a simple linear regression at $x = 0$, $x = 0.75$ and $x = 1.50$.



On the other hand, there are response variables that are not continuous, but binary, i.e. with values in $\{0,1\}$, or which are a proportion in $[0,1]$. Still, it can be very worthwhile to study the dependence of this response on a number of predictors with a regression approach. However, applying the standard multiple regression framework will ultimately result in responses that are beyond the set of values which are foreseen in that problem. Thus, we need some additional techniques which can deal with these types of situations. Depending on how exactly the response variable is, there are several different approaches, which all fit within the framework of a more widely formulated concept entitled *generalized linear modeling* (GLM). Here follows a brief overview of the covered topics:

4.1.1 Binary Response

In toxicological studies, one tries to infer whether a lab mouse survives when it is given a particular dose of some poisonous medication. In human medicine, we are often interested in the contrary case: how much "dose" has an effect, i.e. reduces pain or other symptoms. Here, the response variable is a binary variable in $\{0,1\}$, standing for either *survival vs. death*, or *status quo vs. reduction*. The conditional distribution of the response is $y_i \mid X_i \sim Bernoulli(p_i)$ and hence much different from a Gaussian. Our interest lies in modeling the expectation of this conditional distribution which is $E[y_i \mid X_i] = p_i$, the probability of death resp. pain reduction. We illustrate this with an example where we acquired the response for a cohort of 72 patients and also the dose given per bodyweight (in $[mg/kg]$) was known. The data present themselves as follows:



The naïve approach is to use simple linear regression with the 0/1 outcome as response and the dose as predictor. This yields the blue regression line. Obviously, this results in fitted values outside of the interval $[0,1]$, whose interpretation is unclear. A good statistical model for the above example must yield a $[0,1]$-restricted probability for positive response. This is a coherent approach and takes into account that for a given (intermediate) concentration, we will only have an effect on some of the subjects, but not on all of them. A potential solution is offered by the red dotted line in the above plot. The question is how the curved red line is obtained/estimated. While the full details are covered in section 4.2, we briefly mention that fitting such a logistic regression model is based on estimating the positive response probability $p_i = P(y_i = 1 \mid x_i)$ for each observation i such that:

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 \cdot x_i \text{, or equivalently } p_i = \frac{\exp(\beta_0 + \beta_1 x_i)}{1+\exp(\beta_0 + \beta_1 x_i)}$$

The regression parameters β_0, β_1 are determined by optimizing the goodness-of-fit of the p_i with a maximum likelihood approach, for details see section 4.2.

4.1.2 Count Response

What are predictors for the abundance of starfish (*in German: Seestern*) at several locations in the sea? For answering this question, we could analyze a dataset consisting of counts in different areas, plus the values of several predictors. Obviously, the response y_i is a count – the simplest and natural model for the conditional distribution $y_i \mid X_i$ is a Poisson distribution. We then assume that the parameter λ_i at location i depends on the predictors. Since $\lambda_i > 0$, we must use a log-transformation to link it to the linear combination:

$$y_i \mid X \sim Pois(\lambda_i) \text{ where } \log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}.$$

4.1.3 Categorical Response

Another extension of the linear model is necessary for the case where we try to predict a nominal response variable. For example, we may be interested in giving probabilities for the favorite political party of a person, depending on predictors such as education, age, etc. Such data can be summarized and displayed in contingency tables. The goal is modeling conditional probabilities $P(y_i = k \mid X_i)$ for the categories $k = 1, ..., K$.

4.1.4 Generalized Linear Models

The above mentioned models for binary, count and categorical response all fit within the framework of generalized linear models, which also encompasses the multiple linear regression approach from chapter 3. GLMs are based on the notion that the suitably transformed conditional expectation of the response y_i has a linear relation to the predictors, i.e.:

$$g(E[y_i \mid X_i]) = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}.$$

As we had argued above, in case of multiple linear regression the link $g(\cdot)$ will be the identity function, and the conditional distribution $y_i \mid X_i$ is a Gaussian. The specifics of the model extensions for counts and categorical response will be discussed on the following pages. Please note that formally, GLMs also require that the responses' variance y_i is of the form $\phi \cdot v(E[y])$, where ϕ is an additional parameter, and $v(\cdot)$ a specific function. Moreover, the choice of conditional distributions $y_i \mid X_i$ that are tractable within the limits of GLMs is restricted.

While the GLM formulation and the restrictions may sound complicated, they allow for formulating a unified mathematical theory that encompasses common basic principles for estimation, inference and model diagnostics. We will not deeply embark into the formal aspects, but limit ourselves to the practically relevant do's and dont's of applied generalized linear modeling. For readers who are interested in pursuing the theory on GLMs, we refer to the seminal work "Generalized Linear Models" by McCullagh and Nelder (Chapman and Hall, 1989).

4.2 Logistic Regression

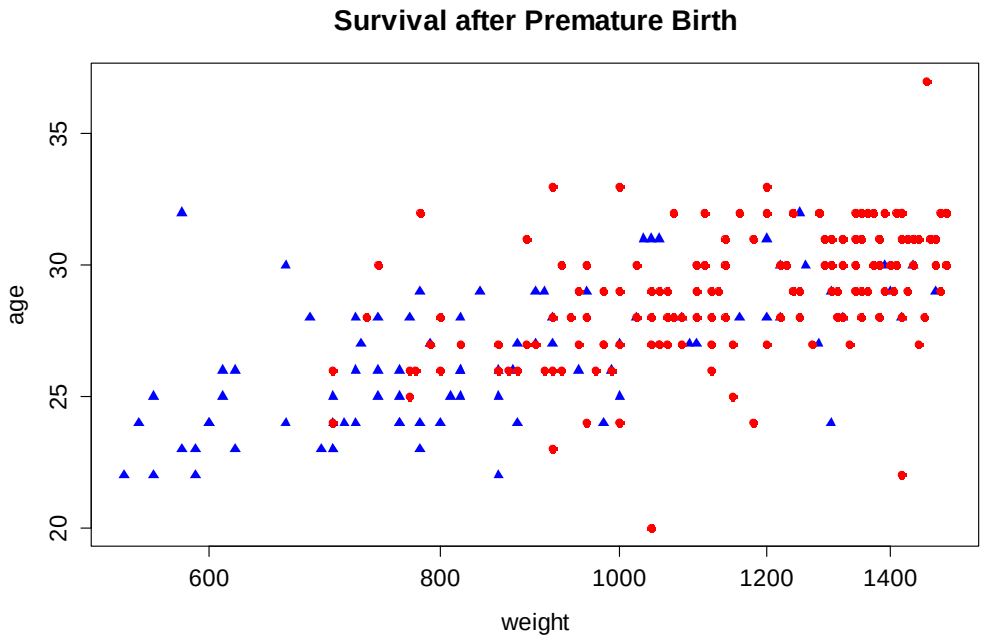
As explained above, datasets with a binary response variable are not multiple linear regression problems. Here, we will discuss the necessary extension. While in many aspects, techniques and ideas are similar to what is already known, some novel issues appear as well. We will take care of model formulation, estimation, inference, diagnostics, prediction and model choice.

4.2.1 Example: Survival after Premature Birth

We discuss an example dealing with survival after premature birth. A study of Hubbard (1986) contains data of 247 early born babies. Predictors for survival are *birth weight* (in grams), *birth age* (in weeks of pregnancy), the *apgar scores* (judging the vital functions one and five minutes after birth) and the *pH-value* of the babies' blood (providing information on oxygen saturation). For reasons of simplicity, we limit ourselves to the two most informative predictors, *age* and *weight*. Due to positive skewness, we perform a log-transformation for the latter.

If we color-code the response, i.e. *survival with red dots* and *death with blue triangles*, the data can be displayed in a 2d-scatterplot, see below. It is apparent that the proportion of surviving babies depends on age and weight: the older and heavier a baby is born prematurely, the better the odds for surviving are. The goal with our logistic regression analysis will be the quantitatively model the probability for survival conditional on the two predictors weight and age.

```
> plot(age ~ weight, data=baby, log="x", type="n")
> points(age ~ weight, subset=(survival==0), data=baby, ...)
> points(age ~ weight, subset=(survival==1), data=baby, ...)
> title("Survival after Premature Birth")
```



4.2.2 Model and Estimation

In the premature birth example, the response is binary: $y_i \in \{0,1\}$. Hence y_i follows a *Bernoulli distribution*, whose parameter (called the "success probability") is denoted by p_i. Typically, the parameter and the distribution are conditional on the predictor(s), for which we use the general notation X_i:

$$p_i = P(y_i = 1 | X_i) = E[y_i | X_i] = \mu_i,$$

It is important to note that p_i is not only the parameter of the responses' Bernoulli distribution, but also the conditional expectation μ_i of the response variable y_i. In that situation, the logistic regression model is defined as

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + ... + \beta x_{ip}$$

As we can see, the linear predictor $\eta_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}$ is linked to the conditional expectation $p_i = \mu_i$ via the logit function $p \mapsto \log(p/(1-p))$, which maps from $[0,1]$ to $(-\infty, +\infty)$. Thus, we are "safe" to use a linear combination of the predictors, i.e. it is certain that the output will always be a $[0,1]$-restricted probability. The logit function has a descriptive interpretation: $p/(1-p)$ is called the *odds* (*"Wettverhältnis"* in German) for an event. A probability of $1/2$ yields to a $1:1$ odds, i.e. both outcomes are equally likely. If we have $p_i = 1/4$, then the odds turns out to be $1:3$, i.e. the second outcome is three times as likely. Odds are always positive, so that we require a log-transformation to obtain a full real-valued scale. Thus, logistic regression equals describing the *log-odds* with a linear model.

A peculiarity of logistic regression is that there is no explicit, additive error term as in multiple linear regression. It is not needed because we model p_i: the variation in the babies' survival for a given combination of birth age and weight is already dealt with by the Bernoulli distribution of the response variable with parameter p_i.

Estimation

For practical application, it is important to estimate the regression coefficients $\beta_0, ..., \beta_p$ on a given dataset. While it would conceptually be possible to minimize the sum of squared raw residuals $r_i = p_i - y_i$, this approach is not theoretically sound and does not prove to be fruitful in practice. Instead, we perform *maximum likelihood estimation* (MLE) where the regression coefficients β_j are determined such that the likelihood of the observed data is maximized. By assuming independence of the cases, this boils down to determine the parameters such that the Bernoulli log-likelihood

$$l(\beta) = \sum_{i=1}^{n} \left(y_i \log(p_i) + (1-y_i) \log(1-p_i) \right)$$

is maximized. Note that this is a sensible goodness-of-fit measure. For all observations with $y_i = 1$ we aspire for high p_i to keep the contribution to $l(\beta)$ low,

and vice versa for the observations with $y_i = 0$. The dependence of $l(\beta)$ on the data and the regression coefficients becomes more obvious if p_i is replaced with:

$$p_i = \frac{\exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})}$$

The log-likelihood maximization can be approached by taking partial derivatives of $l(\beta)$ with respect to $\beta_0, ..., \beta_p$. This still yields an equation system, but in contrast to multiple linear regression, it is no longer a linear one that is easy to solve. However, under some mild conditions, the maximum exists, though it cannot be written in closed form. We thus require numerical optimization. It turns out that a good method is to employ linear approximations that are solved using a sequence of *iteratively reweighted least squares regressions* (the *IRLS algorithm*). We do without giving further details, but instead focus on the practical application.

R-Code for Estimating Logistic Regression Models

In R, routines for estimating logistic regression models are readily available. We illustrate their syntax on the premature birth example:

```
> glm(survival~log10(weight)+age, family=binomial, data=baby)

Coefficients:
  (Intercept)  log10(weight)           age
     -33.9711        10.1685        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 $\hat{\beta}_2$. Please note that this is a numerically optimized solution, so it may happen that the following warning message appears:

```
Warning message:
glm.fit: algorithm did not converge
```

It obviously means that the IRLS algorithm did not converge, and hence the coefficients are not trustworthy. Unfortunately, it is not possible to make general statements on how to achieve convergence. On the other hand, convergence problems are rare in well-posed regression problems. Another issue (and warning) that can arise is the one of fitted 0 or 1 probabilites:

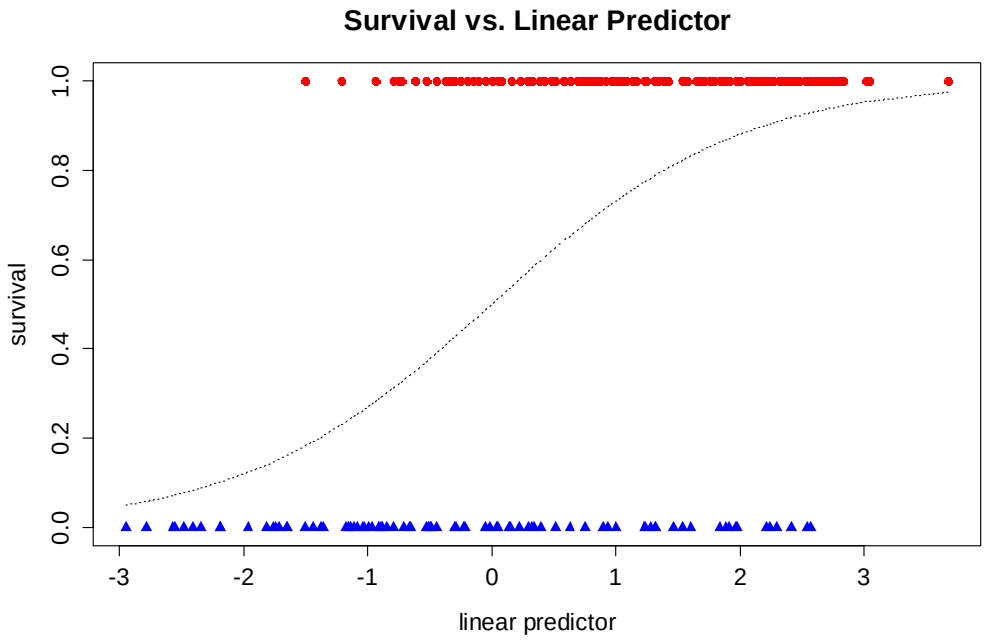
```
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
```

This is also known as the *Hauck-Donner phenomenon* and indicates that there is a subspace in predictor space with perfect separation of observations with $y = 0$ and $y = 1$. If that is the case, the optimal regression coefficient estimates could be arbitrarily large what makes it difficult to explicitly determine them. As a way out, R artificially limits the estimated coefficients to a maximum of +/- 10 and issues the above warning. Working with such models is usually fine despite the warning.

Displaying the Fit

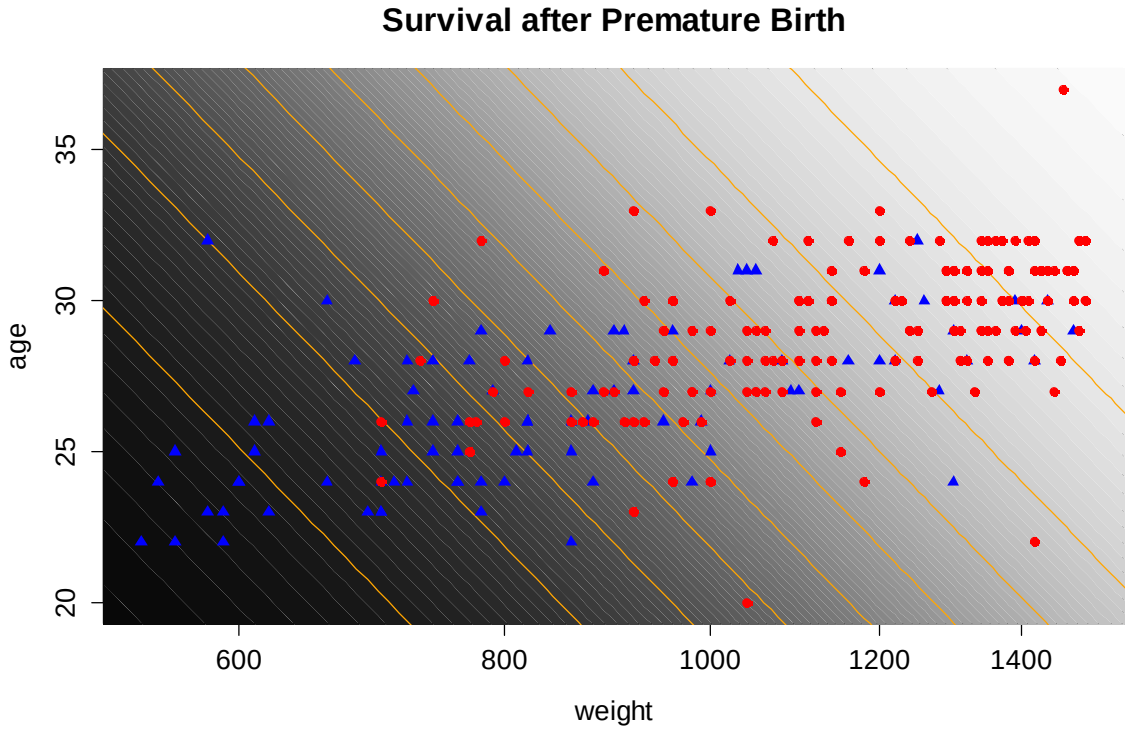
A simple option for displaying the GLM fit is to plot the response vs. the linear predictor. This always works, no matter how many predictor variables one uses. The following code is required:

```
> fit <- glm(survival ~ log10(weight) + age, family=binomial)
> fvl <- predict(fit, type="link")
> fpr <- predict(fit, type="response")
> plot(fvl, survival, type="n", xlab="linear predictor")
> points(fvl[survival==0], survival[survival==0])
> points(fvl[survival==1], survival[survival==1], col="red")
> lines(sort(fvl), sort(fpr), lty=3)
> title("Survival vs. Linear Predictor")
```



Small values in the linear predictor correspond to low survival probability, and vice versa. The fitted values show their typical S-shaped curve that is created by the inverse of the logit function. The value 0 always marks the midpoint: it corresponds to an odds of 1 and hence equal chances for survival vs. death.

Another option for displaying the results (that is restricted to examples with two predictors as in premature birth) is to color code the probability of survival. By keeping the probability of survival fixed, we can express the *age* value as function of *log10(weight)*. Some quick calculations show that the resulting function is linear, hence the contours of equal probability are given by parallel straight lines. The frame on the next page illustrates this for the age vs. weight plot: black background color would correspond to a survival probability of 0, and white to 1. The orange contours stand for probability 0.1, 0.2, …, 0.9. As we can observe, there are babies who survive with estimated probabilities below 0.2, whereas others die despite of estimated survival probabilities above 0.9.



Interpretation of the Regression Coefficients

We now turn our attention to the interpretation of the regression coefficients β_j. As we had stated above, the log-odds for $y_i = 1$ are a linear function of the predictors. Thus, if predictor x_j is increased by 1 unit, then the log-odds in favor of $y = 1$ increase by β_j if all other predictors remain unchanged. We illustrate this with the premature birth example, where we consider an individual with $\log_{10}(weight) = 3$ and birth age of 30 weeks. We have:

$$\eta = -33.9711 + 10.1685 \cdot 3.0 + 0.1474 \cdot 30 = 0.957 ,$$

which are the log-odds for survival. If we take $\exp(0.957) = 2.604$, we obtain the odds for survival. It is thus 2.604 times more likely to survive than die when born at this particular combination of age and weight. On the other hand, the probability for survival is:

$$g^{-1}(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)} = \frac{\exp(0.957)}{1+\exp(0.957)} = 0.723$$

Now, if we compare to an individual with birth age 31 weeks (and the rest remaining as above), we obtain the odds as $\exp(1.104) = 3.017$. If we divide the two odds, we obtain the odds-ratio:

$$\frac{3.017}{2.604} = 1.159 = \exp(\hat{\beta}_2)$$

The odds for surviving increase by $\exp(\hat{\beta}_2)$ (i.e. about 15%) when a baby of the same weight is born one week later – this is a more illustrative way to see the parameter $\hat{\beta}_2$. In other words, we can say that the coefficients from logistic regression models are *log-odds ratios*.

Alternative Link Functions

The role of the link function is to map between the conditional expectation $E[y \mid X]$ and the linear predictor η. In logistic regression, we must ensure that this mapping is between the intervals $[0,1]$ and $(-\infty,+\infty)$. Above, we had argued that the logit function can play this part and is attractive due to its clear interpretation. However, we could use any function that maps between these intervals and in fact, the inverse of any *cumulative distribution function (cdf)* will do so.

Hence, an intuitive alternative choice for the link function is $\eta = \Phi^{-1}(p)$, the inverse of the Gaussian cdf. The resulting procedure is known as *Probit Regression*. In most applied problems, the difference between probit and logistic regression are negligible. Unless you know exactly that you are in a setting where one needs to use the probit link function, it is probably better to stick to the logit link. Even more exotic is the complementary log-log link $\eta = \log(-\log(1-p))$. There are some special cases where it is useful, but giving the details is far beyond the scope of this course.

4.2.3 Inference for Logistic Regression

We base our discussion about inferring a logistic regression model on the summary output in R. Most concepts are already known from our previous discussion about multiple linear regression, but reappear in slightly different form.

```
> summary(fit)

Call: glm(survival ~ log10(weight) + age, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2983  -0.7451   0.4303   0.7557   1.8459

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -33.97108    4.98983  -6.808 9.89e-12 ***
log10(weight)  10.16846    1.88160   5.404 6.51e-08 ***
age             0.14742    0.07427   1.985   0.0472 *
---
Dispersion parameter for binomial family taken to be 1

    Null deviance: 319.28  on 246  degrees of freedom
Residual deviance: 235.94  on 244  degrees of freedom
AIC: 241.94
Number of Fisher Scoring iterations: 4
```

Perhaps the most important difference is that the multiple R-squared and the global F-test are missing here, and only some information about the deviance is given. For deeper insight, we need to consider the goodness-of-fit measure that is used here, which is the so-called deviance:

$$D(y,\hat{p}) = -2 \cdot \sum_{i=1}^{n} \left(y_i \log(\hat{p}_i) + (1-y_i)\log(1-\hat{p}_i) \right)$$

We can see that $D = -2l(\hat{\beta})$, i.e. minus twice the log-likelihood of our model. The summary output lists the value under `Residual deviance`, together with the degrees of freedom of that model, which are $df = n-(p+1)$. The `Null deviance` conceptually is the same, but for the simplest possible model that only has no predictors but only the intercept, and which fits the a-priori probability (i.e. the relative frequency of observations with $y_i = 1$) to all observations.

Coefficient of Determination

In multiple linear regression, the multiple R-squared was an intuitive concept for the explanatory content in the predictors. We might consider applying the same idea for logistic regression models by measuring the proportion of deviance explained. In particular, this would yield $1-235.94/319.28 = 0.26$, or in R:

```
> 1-fit$dev/fit$null
[1] 0.2610193
```

This simple measure is often reasonable for practical application, though the proportion of deviance explained is an often debated topic. A better statistic to measure the explanatory content is:

$$R^2 = \frac{1-\exp((D_{res}-D_{null})/n)}{1-\exp(-D_{null}/n)},$$

where D_{res} and D_{null} are the residual resp. null deviance and n is the number of observations. When this alternative measure is implemented in R, we obtain:

```
> (1-exp((fit$dev-fit$null)/247))/(1-exp(-fit$null/247))
[1] 0.3947567
```

Individual Hypothesis Tests and Confidence Intervals

In multiple linear regression, when assuming Gaussian errors, it is quite easy to show that the estimated regression coefficients $\hat{\beta}_j$ are normally distributed. That property can be used for constructing the individual hypothesis test and the confidence interval. Both require the use of the standard error $\hat{\sigma}_{\hat{\beta}_j}$ for standardization, hence the t-distribution comes into play. All this is no longer true for logistic regression. Maximum likelihood theory tells us that under some mild conditions, the $\hat{\beta}_j$ are asymptotically Gaussian. In practice, we are of course lacking infinitely many observations, but the asymptotic result is used for concluding that the regression coefficients are approximately Gaussian.

This property can be used to assess the individual hypothesis tests and to determine the confidence intervals for the coefficients. We simply assume that under the null hypothesis $H_0 : \beta_j = b$

$$Z = \frac{\hat{\beta}_j - b}{\hat{\sigma}_{\hat{\beta}_j}} \sim N(0,1)$$

The p-values of the individual hypothesis tests for $H_0 : \beta_j = 0$ are given in the summary output. Due to the normal assumption, the respective columns are entitled “z-value” rather than the “t-value” we had in multiple linear regression. A 95%-confidence interval for β_1 can be hand-constructed via:

```
> 10.16846+qnorm(c(0.025,0.975))*1.88160
[1] 6.480592 13.856328
```

A convenient way for obtaining the confidence interval is with the R command `confint()`. However, it uses a more sophisticated method for deriving the result, hence it is not numerically identical to the hand-constructed one.

```
> confint(fit, "log10(weight)")
Waiting for profiling to be done...
    2.5 %    97.5 %
 6.618496 14.032741
```

Comparing Hierachical Models

This section discusses the analogue to the partial F-test of multiple linear regression. The idea behind is still the same, namely comparing two hierarchical models by their goodness-of-fit measure and their difference in degrees of freedom. However, some adjustments are necessary here, because we are now using the deviance rather than the residual sum of squares which leads to a different distribution in the test statistic. Let us assume that we have Big and $Small$ models where $(p+1)$ resp $(q+1)$ parameters are estimated. Our interest lies in the null hypothesis $H_0 : \beta_{q+1} = \beta_{q+2} = ... = \beta_p = 0$. This means that the additional predictors in the big model have zero coefficients and thus are useless. The MLE theory suggest to use the likelihood ratio (or log-likelihood difference) as the test statistic:

$$2 \cdot (ll^{Big} - ll^{Small}) = D(y, \hat{p}_{Small}) - D(y, \hat{p}_{Big}) \sim \chi^2_{p-q}$$

The log-likelihood difference can be computed from the difference in deviance between the two models. Under the null hypothesis, it asymptotically follows a Chisquare distribution with $p-q$ degrees of freedom. We illustrate the procedure by performing the **global test** against the null model. This is the easiest model that is possible and contains the intercept term only. In our example about baby survival, the null model fits the overall survival proportion $\hat{p}_{Null} = \sum y_i / n$ to all observations, no matter what the birth weight or age were. Our goal is now to compare against the full model with age- and weight-specific survival probabilities.

The two deviances are reported in the summary output. We observe a value of 319.28 for the null model (the `Null deviance`) and 235.94 for the model with two predictors (the `Residual deviance`). The full model has two parameters more, and hence the difference of the deviances follows a χ^2_2 distribution. We can thus compute the p-value for the null hypothesis $H_0 : \beta_1 = \beta_2 = 0$:

```
> 1-pchisq(fit$null-fit$dev, df=(fit$df.null-fit$df.res))
[1] 0
```

We obtain a p-value that is (numerically) zero; hence there is a strongly significant contribution of the two predictors on the odds for survival. While typing the above command into R is not a big effort, there is a quick check that can be done by looking at the summary output. The Chisquare distribution with $p-q$ degrees of freedom has an expectation of $p-q$ and standard deviation $\sqrt{2(p-q)}$. The consequence is that if the difference between null and residual deviance is large with respect to the difference in degrees of freedom, then the predictors do yield a significant contribution.

For factor variables with multiple levels, where a hierarchical model comparison is required to test their contribution to the model, the R command `drop1()` is very useful. The difference of deviance test is implemented there. We type:

```
> drop1(fit, test="Chisq")
Single term deletions

Model:
survival ~ log10(weight) + age
             Df Deviance    AIC    LRT  Pr(>Chi)
<none>            235.94 241.94
log10(weight)  1   270.19 274.19 34.247 4.855e-09 ***
age            1   239.89 243.89  3.948   0.04694 *
```

The function tests the exclusion of all model terms, using the difference of deviance as a test statistic. Please note that for all variables with one degree of freedom only, this is a special case of hierarchical model comparison, namely an individual hypothesis test. However, in case of GLMs the results will be slightly different to the ones that are reported in the summary. The latter are based on the approximation of the estimated coefficients' distribution to the Gaussian, whereas here the comparison is against the Chisquare. Hence, the p-values are not equal, though the difference is relatively small. Asymptotically, it will even vanish – the two tests will be one and the same when infinitely many datapoints are available.

4.2.4 Model Diagnostics

Residual analysis is important for logistic regression too, but somewhat more difficult than for multiple linear regression. Although there is no error term in logistic regression, the diagnostics will be based on differences between observed and fitted values. The intuitive concept is to work with the so-called response residuals:

$$r_i = y_i - \hat{p}_i .$$

One of the major problems with these response residuals is that they are heteroskedastic. Their variance is bigger when $\hat{p}_i \approx 1/2$ and smaller when $\hat{p}_i$ is close to either zero or one. For overcoming the issue, one could try to standardize by $\sqrt{\hat{p}_i(1-\hat{p}_i)}$, the estimated standard deviation of r_i. The result is:

$R_i = \frac{y_i - \hat{p}_i}{\sqrt{\hat{p}_i(1-\hat{p}_i)}}$, called the *Pearson Residuals*.

In R, the Pearson Residuals can easily be obtained by using the command `resid(fit, type="pearson")`, when object `fit` contains the results of a logistic regression. Another fruitful approach to the definition of a residual lies in using the contribution d_i of each instance to the goodness-of-fit measure $D(y, \hat{p})$:

$$d_i = -2 \cdot \left(y_i \log\left(\hat{p}_i\right) + (1-y_i) \log\left(1-\hat{p}_i\right) \right)$$

For obtaining a residual that can be well interpreted, we take the square root and enhance it by the sign of the difference between true and fitted value. Thus:

$D_i = sign(y_i - \hat{p}_i) \cdot \sqrt{d_i}$ are called the *Deviance Residuals*.

Again, in R, computation is simple: `resid(fit, type="deviance")`, when as before, object `fit` contains the results of a logistic regression. It is crucial to understand the properties of these residuals well. For a correctly specified model, both the Pearson and Deviance versions have expectation zero and roughly constant variance. The latter can be improved by standardizing or studentizing with $1/\sqrt{1-h_{ii}}$ to remove the estimation-induced heteroskedasticity, though this hardly proves practically necessary if no extreme leverage points are present. Please note that the residual for a particular instance i has a binary distribution, i.e. can only take two values. Obviously, this distribution is not a Gaussian and furthermore, it is generally not identical for different cases i. Please note that the generic `plot(fit)` command in R produces the well-known four standard plots for model diagnostics also if object `fit` is from a GLM. However, these are inadequate here for several reasons, as will be explained below.

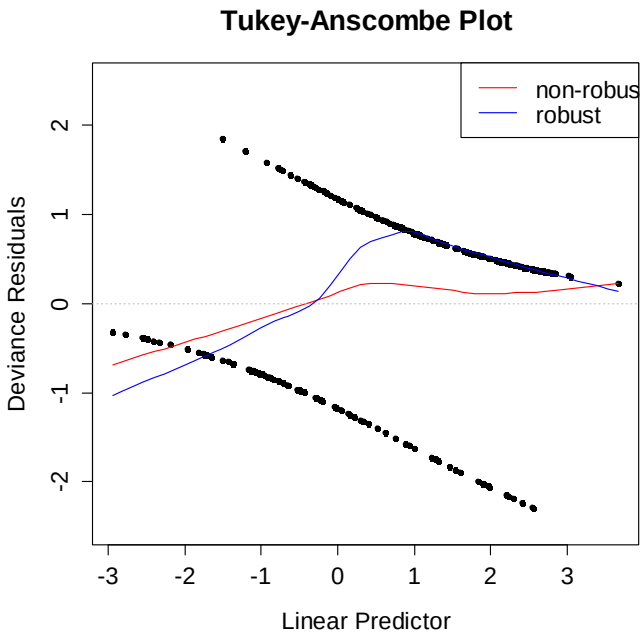
Tukey-Anscombe Plot

As in multiple linear regression, the Tukey-Anscombe plot is important for identifying potential model missspecification and displays the residuals vs. the fitted values. In logistic regression, there is some freedom and also debate whether it shall be based on Pearson or Deviance residuals. Mostly, the difference between the two versions can be neglected from a practical viewpoint. Hence in this scriptum, we restrict to showing all residual plots with the Deviance residuals only. Also for the x-axis there is some choice: we can either use the linear predictor or the fitted probabilities. Since there is a non-linear transformation between the two, the plots will look (often only slightly) different. Our choice for this scriptum is to work with the linear predictor as the x-variable.

The interpretation of these plots is more difficult than in multiple linear regression. As can be seen from the Tukey-Anscombe plot below, the residual for a given value of the linear predictor can only take two different values and thus has a binary distribution. That distribution (i.e. the values it can take and also the likelihood they are taken with) will be non-identical for different linear predictor values. For example, it is evident that positive residuals will be rare for small values of the linear predictor, and vice versa.

Due to this very specific distribution of the residuals, the Tukey-Anscombe plot is somewhat more difficult to read than in multiple linear regression. We can only detect a potential model inadequacy if a smoother is displayed, which basically amounts to comparing against a more flexible (non-parametric) logistic regression model. It is absolutely crucial to *use a non-robust smoother* for this task. Else, at high or low values for η_i, the rare positive resp. negative residuals will be considered as outliers and hence down-weighted in the smoother fit. This is highly unwanted and hampers the correct interpretation of the plot. Unfortunately, the R version of the Tukey-Anscombe plot generated by `plot(fit)` relies on a robust smoother – we need to code a better version by ourselves. The fragment of code below produces a Tukey-Anscombe plot with a non-robust smoother:

```
> xx <- predict(fit, type="link")
> yy <- residuals(fit, type="deviance")
> plot(xx, yy, pch=20, main="Tukey-Anscombe Plot")
> lines(loess.smooth(xx, yy, family="Gaussian"), col="red")
> abline(h=0, lty=3, col="grey")
```



For expositional purposes, we also added a robust smoother. As we can observe, the survival probability tends to be slightly overestimated for low weight and age, and vice versa. The model missspecification is not severe here, though an interaction term or further predictor transformations might be beneficial.