# Generalized Linear Models

# Introduction

- Linear models are well suited for regression analyses when the response variable is continuous and at least approximately normal.
- In many applications the response is not a continuous variable, but rather binary, categorical, or a count variable.
- Difficulties with linear regression models are also encountered for continuous variables which are considerably skewed.
- Generalized linear models (GLMs) unify many regression approaches with response variables that do not necessarily follow a normal distribution.

# Binary Regression

# Binary Regression

- Assume that (ungrouped) data on $n$ objects or individuals are given in the form $(y_i, x_{i1}, \ldots, x_{ik})$, $i = 1, \ldots, n$, with the binary response $y$ coded by 0 and 1 and covariates denoted by $x_1, \ldots, x_k$.
- The covariates $x_1, \ldots, x_k$ may have been derived from an appropriate transformation or coding of the original covariates.
- The main goal of a binary regression analysis is then to model and estimate the effects of the covariates on the (conditional) probability

  $$\pi_i = \mathbb{P}(y_i = 1) = \mathbb{E}(y_i),$$

  for the outcome $y_i = 1$ and given values of the covariates $x_{i1}, \ldots, x_{ik}$.
- The response variables are assumed to be (conditionally) independent.

# Linear Probability Model

The **linear probability model** is given by

$$\pi_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik},$$

with linear predictor

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} = \mathbf{x}_i' \boldsymbol{\beta},$$

with $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)'$ and $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ik})'$.

- The linear predictor is equal to the success probability.
- The linear predictor must lie in the interval $[0, 1]$ for all vectors $\mathbf{x}$.
- This requires restrictions on the parameters $\boldsymbol{\beta}$ that are difficult to handle in the estimation process.

# Binary Regression Models

- Combine the probability $\pi_i$ with the linear predictor $\eta_i$ through a relation of the form

$$\pi_i = h(\eta_i) = h(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}),$$

  where $h$ is a strictly monotonically increasing cumulative distribution function on the real line.
- This ensures $h(\eta) \in [0, 1]$.
- In addition one can also write

$$\eta_i = g(\pi_i),$$

  with the inverse function $g = h^{-1}$.
- Within the framework of GLMs, $h$ is called the **response function** and $g = h^{-1}$ is known as the **link function**.
- Logit and probit models are the most widely used binary regression models.

# Binary Regression Models / 2

**Logit model:**

$$\pi = \frac{\exp(\eta)}{1 + exp(\eta)} = \text{plogis}(\eta) \qquad \Leftrightarrow \qquad \eta = \log \frac{\pi}{1 - \pi} = \text{qlogis}(\pi).$$

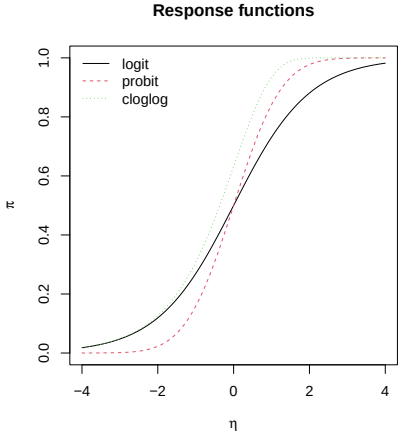
**Probit model:**

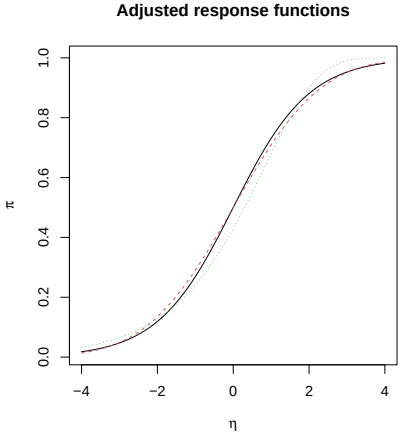
$$\pi = \Phi(\eta) = \text{pnorm}(\eta) \qquad \Leftrightarrow \qquad \eta = \Phi^{-1}(\pi) = \text{qnorm}(\eta).$$

**Complementary log-log model:**

$$\pi = 1 - \exp(-\exp(\eta)) \qquad \Leftrightarrow \qquad \eta = \log(-\log(1 - \pi)).$$

# Binary Regression Models / 3



# Binary Regression Models / 4

- Instead of the probit one could have used the more general cumulative function $h$ of a $N(0, \sigma^2)$ distribution with any choice of variance $\sigma^2 \neq 1$.
- Standardizing $h$ yields the relation

$$\pi(\eta) = h(\mathbf{x}'\boldsymbol{\beta}) = \Phi(\mathbf{x}'\boldsymbol{\beta}/\sigma) = \Phi(\mathbf{x}'\tilde{\boldsymbol{\beta}}),$$

  where $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}/\sigma$.
- The resulting model for the probability $\pi(\eta)$ based on $h(\eta)$ with $\eta = \mathbf{x}'\boldsymbol{\beta}$ is equivalent to a probit model with the rescaled parameters $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}/\sigma$.

# Binary Regression Models / 5

- For an adequate comparison of the models the mean and variance of the CDFs need to be matched.
- The logistic distribution function has variance $\pi^2/3$. Thus it needs to be compared to a rescaled normal distribution whose variance is adjusted to $\sigma^2 = \pi^2/3$.
- The logit and adjusted probit response functions are very similar.
- The estimated coefficients of a logit model differ from the corresponding values of a probit model (with $\sigma^2 = 1$) approximately by the factor $\sigma = \pi/\sqrt{3}$.
- The estimated probabilities $\pi(\eta)$ are very similar.
- This indicates that instead of the absolute values of the (estimated) coefficients rather the ratios should be interpreted.

# Binary Regression Models / 6

- Similar considerations apply to the comparison with the complementary log-log model.
- The extreme-value distribution has variance $\sigma^2 = \pi^2/6$ and expectation $-0.5772$.
- The response function has to be adjusted to the variance $\sigma^2 = \pi^2/3$ and expectation 0 for a comparison with the logistic distribution function.
- This adjustment does have additional impact on the (estimated) intercept $\beta_0$.
- The corresponding adjusted response function follows a similar form as those of the logit and probit function for small $\eta$, but also shows clear differences for larger $\eta$.

# Binary Models and Latent Linear Models

- Binary regression models can be derived by considering a **latent (unobserved) continuous response variable**.
- The latent variable is connected with the observed binary response via a threshold mechanism.
- Suppose we are investigating the decision of some individuals $i = 1, \ldots, n$ when choosing between two alternatives $y = 0$ and $y = 1$.
- Assume that individuals assign utilities $u_{i0}$ and $u_{i1}$ to each of the two alternatives.
- The alternative that maximizes the utility is chosen, i.e.,

$$y_i = \begin{cases} 1 & u_{i1} > u_{i0}, \\ 0 & u_{i1} \leq u_{i0}. \end{cases}$$

# Binary Models and Latent Linear Models / 2

- Assuming that the unobserved utilities can be additively decomposed and follow a linear model, we obtain

$$u_{i1} = \mathbf{x}_i' \tilde{\boldsymbol{\beta}}_1 + \tilde{\epsilon}_{i1},$$
$$u_{i0} = \mathbf{x}_i' \tilde{\boldsymbol{\beta}}_0 + \tilde{\epsilon}_{i0},$$

  with $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ik})'$.
- The unknown coefficient vectors $\tilde{\boldsymbol{\beta}}_1$ and $\tilde{\boldsymbol{\beta}}_0$ determine the effect of the explanatory variables on the utilities.
- The "errors" $\tilde{\epsilon}_{i1}$ and $\tilde{\epsilon}_{i0}$ include the effects of unobserved explanatory variables.
- Equivalently, we may choose to investigate utility differences

$$\tilde{y}_i = u_{i1} - u_{i0} = \mathbf{x}_i'(\tilde{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_0) + \tilde{\epsilon}_{i1} - \tilde{\epsilon}_{i0} = \mathbf{x}_i' \boldsymbol{\beta} + \epsilon_i,$$

  with $\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_0$ and $\epsilon_i = \tilde{\epsilon}_{i1} - \tilde{\epsilon}_{i0}$.

# Binary Models and Latent Linear Models / 3

- Based on this framework, the binary responses $y_i$ follow a Bernoulli distribution with

$$\pi_i = \mathbb{P}(y_i = 1) = \mathbb{P}(\tilde{y}_i > 0) = \mathbb{P}(\mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i > 0) = \int I(\mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i > 0) f(\epsilon_i) d\epsilon_i,$$

  where $I(\cdot)$ is the indicator function and $f$ is the probability density of $\epsilon_i$.
- We obtain different models depending on the choice of $f$:
  - When $\epsilon_i$ follows a logistic distribution, we obtain the logit model.
  - When $\epsilon_i$ follows a standard normal distribution, we obtain the probit model.

# Interpretation of the Logit Model

Based on the linear predictor

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} = \mathbf{x}_i' \boldsymbol{\beta},$$

the odds

$$\frac{\pi_i}{1 - \pi_i} = \frac{\mathbb{P}(y_i = 1 | \mathbf{x}_i)}{\mathbb{P}(y_i = 0 | \mathbf{x}_i)}$$

follow the multiplicative model

$$\frac{\mathbb{P}(y_i = 1 | \mathbf{x}_i)}{\mathbb{P}(y_i = 0 | \mathbf{x}_i)} = \exp(\beta_0) \cdot \exp(\beta_1 x_{i1}) \cdot \ldots \cdot \exp(\beta_k x_{ik}).$$

# Interpretation of the Logit Model / 2

If $x_{i1}$ increases by 1 unit to $x_{i1} + 1$, the following changes apply to the relationship of the odds:

$$\frac{\mathbb{P}(y_i = 1|x_{i1} + 1, \ldots)}{\mathbb{P}(y_i = 0|x_{i1} + 1, \ldots)} \Big/ \frac{\mathbb{P}(y_i = 1|x_{i1}, \ldots)}{\mathbb{P}(y_i = 0|x_{i1}, \ldots)} = \exp(\beta_1).$$

- $\beta_1 > 0$: $\mathbb{P}(y_i = 1)/\mathbb{P}(y_i = 0)$ increases,
- $\beta_1 < 0$: $\mathbb{P}(y_i = 1)/\mathbb{P}(y_i = 0)$ decreases,
- $\beta_1 = 0$: $\mathbb{P}(y_i = 1)/\mathbb{P}(y_i = 0)$ remains unchanged.

# Deviance and Deviance Residuals

- The deviance is defined by

$$D = -2 \sum_{i=1}^{n} \{\ell_i(\hat{\mu}_i) - \ell_i(y_i)\}$$

  with $\hat{\mu}_i$ are the estimated expectations. $\ell_i(y_i)$ is the log-likelihood of the saturated model where the number of observations is equal to the number of parameters.

- The deviance residuals are defined as

$$d_i = \text{sign}(y_i - \hat{y}_i)\sqrt{-2(\ell_i(\hat{\mu}_i) - \ell_i(y_i))}.$$

# Estimation in R

The function for fitting a binary regression model in R is:

```
R> glm(formula, family = binomial(link = "logit"), data, weights, subset,
+   na.action, ...)
```

- The formula is specified as for the linear model to model effects on the linear predictor.
- The link for the `binomial()` family can be specified to be logit, probit or complementary log-log.
- Maximum likelihood estimation is performed using an iterative procedure.

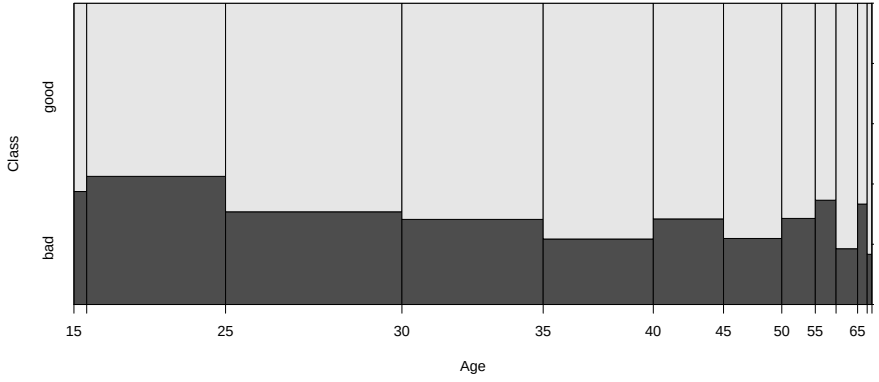
This returns a fitted model object (of class 'glm') with suitable methods for the basic statistical modeling generics.

In particular,

- `fitted()` gives the fitted mean values
- `predict()` can be used for predictions on the given (default) or new data, on the link (default) or response scale.

# Example: German Data Set

```
R> load("german.rda")
R> spineplot(Class ~ Age, data = german)
```



# Example: German Data Set / 2

```
R> model1 <- glm(Class ~ Age, data = german, family = binomial())
R> model1

Call:  glm(formula = Class ~ Age, family = binomial(), data = german)

Coefficients:
(Intercept)          Age
   -0.20092     -0.01844

Degrees of Freedom: 999 Total (i.e. Null);  998 Residual
Null Deviance:            1222
Residual Deviance: 1213         AIC: 1217
```

# Example: German Data Set / 3

```
R> summary(model1)

Call:
glm(formula = Class ~ Age, family = binomial(), data = german)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9540  -0.8781  -0.8063   1.4600   1.8736

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.200919   0.232853  -0.863  0.38822
Age         -0.018440   0.006434  -2.866  0.00416 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

# Example: German Data Set / 4

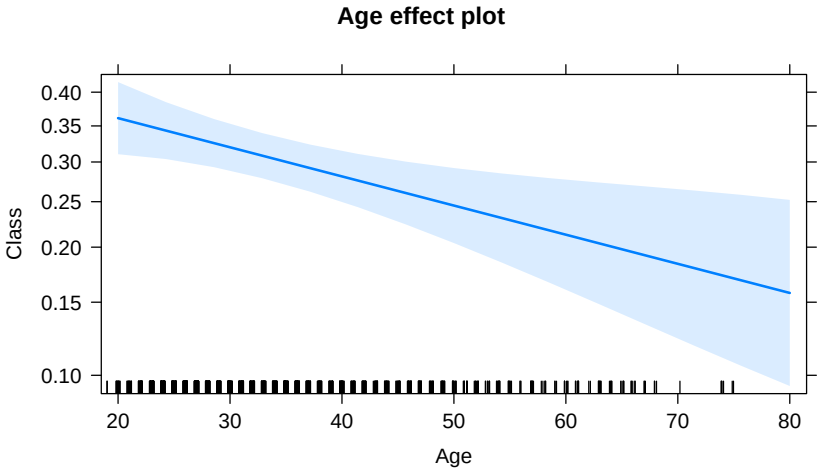
```
    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1213.1  on 998  degrees of freedom
AIC: 1217.1

Number of Fisher Scoring iterations: 4
```

# Example: German Data Set / 5

```
R> library("effects")
R> Effect("Age", model1)

 Age effect
Age
       20        30        50        60        80
0.3613020 0.3199249 0.2454709 0.2129365 0.1576108

R> eff1 <- allEffects(model1)
```

# Example: German Data Set / 6

```
R> plot(eff1)
```

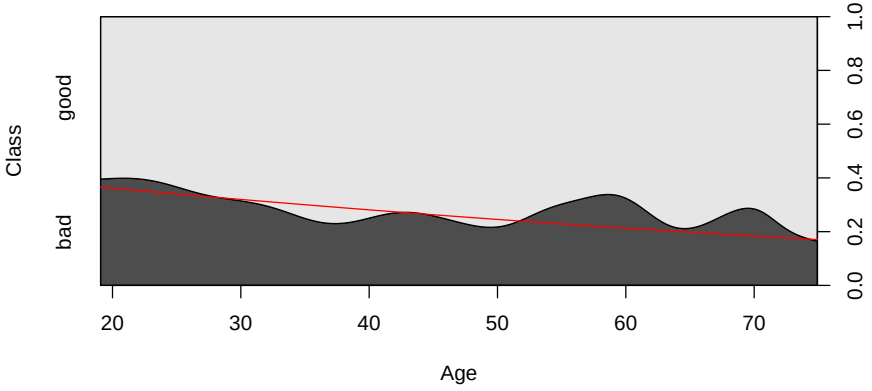


# Example: German Data Set / 7

We could also visualize the model predictions "by hand", by using a conditional density plot instead of a spine plot, and adding the predicted responses for suitable age values:

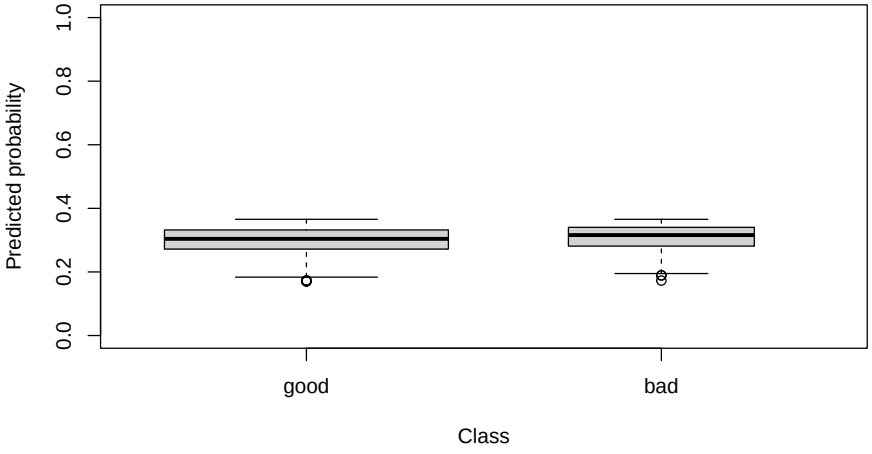
```
R> cdplot(Class ~ Age, data = german)
R> ages <- with(german, seq(from = min(Age), to = max(Age)))
R> p1 <- predict(model1, data.frame(Age = ages), type = "response")
R> lines(ages, p1, type = "l", col = "red")
```

# Example: German Data Set / 8



# Example: German Data Set / 9

```
R> boxplot(predict(model1, type = "response") ~ Class, data = german,
+    ylab = "Predicted probability", varwidth = TRUE, ylim = 0:1)
```



# Example: German Data Set / 10

```
R> model2 <- glm(Class ~ Status_of_checking_account + Age, data = german,
+   family = binomial())
R> model2

Call:  glm(formula = Class ~ Status_of_checking_account + Age, family = binomial(),
    data = german)

Coefficients:
                  (Intercept)  Status_of_checking_accountp_lo
                      0.50860                         -0.43861
Status_of_checking_accountp_hi  Status_of_checking_accountnone
                     -1.20449                         -1.98720
                          Age
                     -0.01524

Degrees of Freedom: 999 Total (i.e. Null);  995 Residual
Null Deviance:            1222
Residual Deviance: 1085         AIC: 1095
```

# Example: German Data Set / 11

```
R> summary(model2)

Call:
glm(formula = Class ~ Status_of_checking_account + Age, family = binomial(),
    data = german)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2651  -0.9761  -0.5082   1.1428   2.2771

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                      0.508603   0.264135   1.926 0.054161 .
Status_of_checking_accountp_lo -0.438609   0.174808  -2.509 0.012104 *
Status_of_checking_accountp_hi -1.204490   0.327253  -3.681 0.000233 ***
Status_of_checking_accountnone -1.987202   0.198488 -10.012  < 2e-16 ***
Age                             -0.015239   0.006663  -2.287 0.022191 *
---
```

# Example: German Data Set / 12

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

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1085.0  on 995  degrees of freedom
AIC: 1095

Number of Fisher Scoring iterations: 4
```

# Example: German Data Set / 13

```
R> anova(model1, model2)

Analysis of Deviance Table

Model 1: Class ~ Age
Model 2: Class ~ Status_of_checking_account + Age
  Resid. Df Resid. Dev Df Deviance
1       998     1213.1
2       995     1085.0  3   128.12
```

# Example: German Data Set / 14

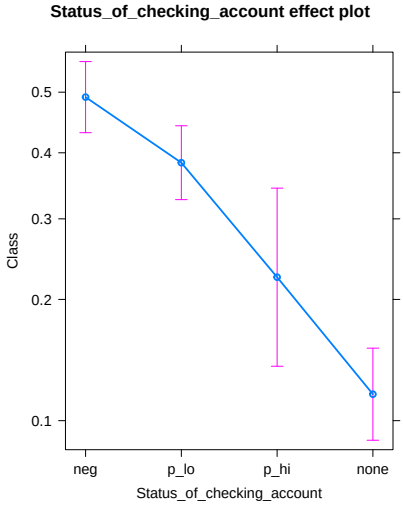
```
R> anova(model1, model2, test = "Chisq")

Analysis of Deviance Table

Model 1: Class ~ Age
Model 2: Class ~ Status_of_checking_account + Age
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       998     1213.1
2       995     1085.0  3   128.12 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

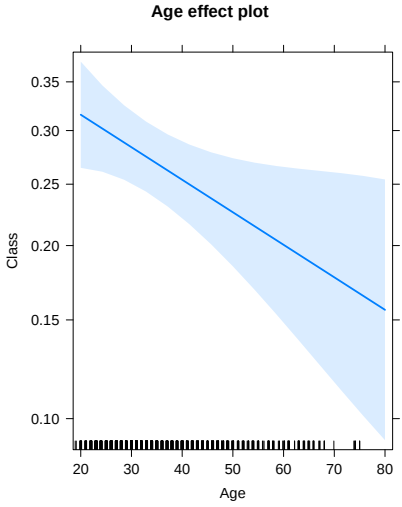
# Example: German Data Set / 15

```
R> eff2 <- allEffects(model2)
R> eff2

 model: Class ~ Status_of_checking_account + Age

 Status_of_checking_account effect
Status_of_checking_account
      neg       p_lo       p_hi       none
0.4917287 0.3842146 0.2248578 0.1170890

 Age effect
Age
       20        30        50        60        80
0.3158291 0.2838593 0.2261497 0.2005960 0.1561238
```

# Example: German Data Set / 16

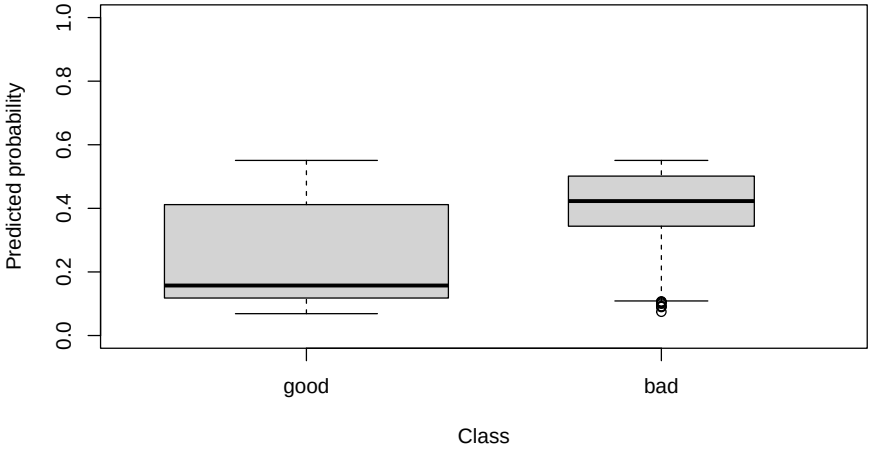
```
R> plot(eff2)
```



# Example: German Data Set / 17

```
R> boxplot(fitted(model2, type = "response") ~ Class, data = german,
+    ylab = "Predicted probability", varwidth = TRUE, ylim = 0:1)
```



# Overdispersion

- Observations with the same covariate values $\mathbf{x}_i$ can be grouped into $G$ groups with each group containing $n_i$ observations.
- The number of successes for group $i$ are then binomially distributed with success probability $\pi_i$.
- For grouped data, we can estimate the variance within a group via $(\bar{y}_i(1 - \bar{y}_i))/n_i$.
- In applications the **empirical** variance is often much larger than the variance

$$\frac{\hat{\pi}_i(1 - \hat{\pi}_i)}{n_i}$$

predicted by a binomial regression model with

$$\hat{\pi}_i = h(\mathbf{x}_i'\hat{\boldsymbol{\beta}}).$$

- If the data show a higher variability than is presumed by the model, one refers to **overdispersion**.

# Overdispersion / 2

- The two main reasons for overdispersion are:
  - unobserved heterogeneity
  - positive correlations
- The easiest way to address the increased variability is through the introduction of a multiplicative overdispersion parameter $\phi > 1$ into the variance formula

$$\mathrm{var}(y_i) = \phi \frac{\pi(1-\pi)}{n_i}.$$

- The overdispersion parameter $\phi$ can be estimated as the average Pearson statistic:

$$\hat{\phi}_P = \frac{1}{n-p}\chi^2.$$

  This is analogous to the estimation of the error variance in the linear model, with $\chi^2$ replacing the residual sum of squares.
- The `quasibinomial()` family in R estimates $\phi$ and uses it as ad-hoc adjustment in standard error calculations.

# Example: German Data Set

```
R> model1a <- glm(Class ~ Age, data = german, family = quasibinomial())

R> summary(model1a)

Call:
glm(formula = Class ~ Age, family = quasibinomial(), data = german)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9540  -0.8781  -0.8063   1.4600   1.8736

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.200919   0.233292  -0.861  0.38932
Age         -0.018440   0.006446  -2.861  0.00432 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasibinomial family taken to be 1.00378)
```

# Example: German Data Set / 2

```
    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1213.1  on 998  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

# (Quasi-)Complete Separation

- Complete separation happens when the outcome variable separates a predictor variable or a combination of predictor variables completely.
- Quasi-complete separation happens when the outcome variable separates a predictor variable or a combination of predictor variables to a certain degree.
- (Quasi-)complete separation leads to large coefficient estimates and standard errors.
- In R no check for (quasi-)complete separation is performed by default. In general issues are indicated by a warning about fitted probabilities being numerically 0 or 1.
- Strategies to deal with (quasi-)complete separation:
  - Assess if the outcome variable is not a dichotomous version of a variable in the model.
  - One might decide not to include the problematic variable in the model. But this may lead to biased estimates.

# Example: German Data Set

```
R> german0 <- subset(german,
+   (Age <= 30 & Class == "bad") | (Age >= 30 & Class == "good"))
R> model0 <- glm(Class ~ Status_of_checking_account + Age,
+   data = german0, family = binomial())

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

# Example: German Data Set / 2

```
R> summary(model0)

Call:
glm(formula = Class ~ Status_of_checking_account + Age, family = binomial(),
    data = german0)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9331   0.0000   0.0000   0.0000   1.9348

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     599.8350 59255.6414   0.010    0.992
Status_of_checking_accountp_lo    0.2412     0.8566   0.282    0.778
Status_of_checking_accountp_hi   -2.2677 45403.9903   0.000    1.000
Status_of_checking_accountnone   -0.8575     1.0330  -0.830    0.407
Age                             -20.0227  1975.1881  -0.010    0.992
```

# Example: German Data Set / 3

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 678.198  on 613  degrees of freedom
Residual deviance:  45.454  on 609  degrees of freedom
AIC: 55.454

Number of Fisher Scoring iterations: 25
```

# Generalized Linear Regression Models

# General Model Definition

The linear model and the regression models for non-normal response variables have common properties that can be summarized in a unified framework:

1. The mean $\mu = \mathbb{E}(y)$ of the response $y$ is connected with the linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$ by a response function $h$ or by a link function $g = h^{-1}$:

   $$\mu = h(\eta) \qquad \text{or} \qquad \eta = g(\mu).$$

2. The distribution of the response variables (normal, binomial, Poisson, and gamma distribution) can be written in the form of a **univariate exponential family**. (For purists: exponential dispersion model.)

# Univariate Exponential Families

- The density of a univariate exponential family for the response variable $y$ is defined by

$$f(y|\theta) = \exp\left(\frac{y\theta - b(\theta)}{\psi} w + c(y, \psi, w)\right).$$

- The parameter $\theta$ is called the natural or canonical parameter.
- For the function $b(\theta)$ it is required that $f(y|\theta)$ can be normalized and the first and second derivative $b'(\theta)$ and $b''(\theta)$ exist.
- The second parameter $\psi$ is a dispersion parameter, while $w$ is a known value (usually a weight).
- It can be shown that

$$\mathbb{E}(y) = \mu = b'(\theta), \qquad \text{var}(y) = \psi b''(\theta)/w.$$

# Univariate Exponential Families / 2

Examples:

| Distribution | | $\theta(\mu)$ | $b(\theta)$ | $\psi$ |
|---|---|---|---|---|
| Normal | $N(\mu, \sigma^2)$ | $\mu$ | $\theta^2/2$ | $\sigma^2$ |
| Bernoulli | $B(1, \pi)$ | $\log(\pi/(1-\pi))$ | $\log(1+\exp(\theta))$ | 1 |
| Poisson | $Po(\lambda)$ | $\log(\lambda)$ | $\exp(\theta)$ | 1 |

# Generalized Linear Model

## Distributional Assumptions

For given covariates $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ik})'$, the response variables are (conditionally) independent and the (conditional) density of $y_i$ belongs to the exponential family with

$$f(y_i|\theta_i) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi} w_i + c(y_i, \psi, w_i)\right).$$

The parameter $\theta_i$ is the natural parameter and $\psi$ is a common dispersion parameter, independent of $i$. For $\mathbb{E}(y_i) = \mu_i$ and $\mathrm{var}(y_i)$, we have

$$\mathbb{E}(y_i) = \mu_i = b'(\theta_i), \qquad \mathrm{var}(y_i) = \psi b''(\theta_i)/w_i.$$

The weight parameter $w_i$ is 1 for ungrouped data ($i = 1, \ldots, n$). $w_i = n_i$ if the data is grouped and $n_i$ is the sample size of group $i$ and the response is the group mean.

# Generalized Linear Model / 2

## Structural Assumptions

The (conditional) mean $\mu_i$ is connected to the linear predictor $\eta_i = \mathbf{x}_i'\boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$ through

$$\mu_i = h(\eta_i) = h(\mathbf{x}_i'\boldsymbol{\beta}) \qquad \text{or} \qquad \eta_i = g(\mu_i),$$

where

$g$ is a (one-to-one and twice differentiable) response function and

$h$ is the link function, i.e., the inverse $g = h^{-1}$.

# Maximum Likelihood Estimation

**Definition**

The ML estimator $\hat{\boldsymbol{\beta}}$ maximizes the (log-)likelihood and is defined as the solution

$$\mathbf{s}(\hat{\boldsymbol{\beta}}) = 0$$

of the score function given by

$$\mathbf{s}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i \frac{h'(\eta_i)}{\sigma_i^2}(y_i - \mu_i) = \mathbf{X}'\mathbf{D}\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}),$$

where

$$\begin{aligned} \mathbf{D} &= \text{diag}(h'(\eta_1), \ldots, h'(\eta_n)), \\ \boldsymbol{\Sigma} &= \text{diag}(\sigma_1^2, \ldots, \sigma_n^2) \\ \boldsymbol{\mu} &= (\mu_1, \ldots, \mu_n)'. \end{aligned}$$

# Maximum Likelihood Estimation / 2

The Fisher matrix is

$$\mathbf{F}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i' \tilde{w}_i = \mathbf{X}'\mathbf{W}\mathbf{X},$$

where

$$\mathbf{W} = \text{diag}(\tilde{w}_1, \ldots, \tilde{w}_n)$$

is the diagonal matrix of working weights

$$\tilde{w}_i = \frac{(h'(\eta_i))^2}{\sigma_i^2}.$$

# Maximum Likelihood Estimation / 3

## Numerical Computation

The ML estimator $\hat{\beta}$ is obtained iteratively using Fisher scoring in form of iteratively weighted least squares estimates

$$\hat{\beta}^{(t+1)} = (\mathbf{X}'\mathbf{W}^{(t)}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{(t)}\tilde{\mathbf{y}}^{(t)}, \quad t = 0, 1, 2, \ldots,$$

with working observations given by

$$\tilde{y}_i^{(t)} = \hat{\eta}_i^{(t)} + \frac{(y_i - h(\hat{\eta}_i^{(t)}))}{h'(\hat{\eta}_i^{(t)})}.$$

# Likelihood Inference

## Asymptotic Properties of the ML Estimator

Let $\hat{\boldsymbol{\beta}}_n$ denote the ML estimator based on a sample of size $n$. Under regularity conditions, $\hat{\boldsymbol{\beta}}_n$ is consistent and asymptotically normal:

$$\hat{\boldsymbol{\beta}}_n \overset{a}{\sim} \mathrm{N}(\boldsymbol{\beta}, \mathbf{F}^{-1}(\boldsymbol{\beta})).$$

This result holds even if the estimator $\mathbf{F}(\hat{\boldsymbol{\beta}})$ replaces $\mathbf{F}(\boldsymbol{\beta})$.

## Estimation of the Scale or Overdispersion Parameter

Using the variance function the dispersion parameter can then be estimated consistently by

$$\hat{\psi} = \frac{1}{G-p} \sum_{i=1}^{G} \frac{(y_i - \hat{\mu}_i)^2}{v(\hat{\mu}_i)/n_i},$$

where $p$ denotes the number of regression parameters, $\hat{\mu}_i = h(\mathbf{x}_i'\hat{\boldsymbol{\beta}})$ is the estimated expectation, $v(\hat{\mu}_i)$ is the estimated variance function, and the data should be grouped as much as possible.

# Testing Linear Hypotheses

## Hypotheses

$$H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{d} \qquad \text{against} \qquad H_1 : \mathbf{C}\boldsymbol{\beta} \neq \mathbf{d}.$$

## Test Statistics

1. Likelihood ratio statistic: $lr = -2(\ell(\tilde{\boldsymbol{\beta}}) - \ell(\hat{\boldsymbol{\beta}}))$.
2. Wald statistic: $w = (\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})'[\mathbf{C}\mathbf{F}^{-1}(\hat{\boldsymbol{\beta}})\mathbf{C}']^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})$
3. Score statistic: $u = \mathbf{s}'(\tilde{\boldsymbol{\beta}})\mathbf{F}^{-1}(\tilde{\boldsymbol{\beta}})\mathbf{s}(\tilde{\boldsymbol{\beta}})$,

where $\tilde{\boldsymbol{\beta}}$ is the ML estimator under $H_0$.

# Testing Linear Hypotheses / 2

## Test Decision

For large $n$ and under $H_0$, we have the asymptotic results

$$lr, w, u \overset{a}{\sim} \chi^2_r,$$

where $r$ is the (full) row rank of $\mathbf{C}$. We reject $H_0$ when

$$lr, w, u > \chi^2_r(1 - \alpha).$$

# Testing Linear Hypotheses / 3

- Wald tests are mathematically convenient when an estimated model is to be tested against a simplified submodel, since it does not require additional estimation of the submodel.
- The score test is convenient when an estimated model is to be tested against a more complex model alternative.
- For moderate sample sizes, the approximation through the $\chi^2$-distribution is generally sufficient.
- For a smaller sample size, e.g., $n \leq 50$, the values of the test statistics can, however, differ considerably.

# Example: German Data Set

```
R> model3 <- glm(Class ~ Status_of_checking_account + Age + Employment_since,
+   data = german, family = binomial())
R> summary(model3)

Call:
glm(formula = Class ~ Status_of_checking_account + Age + Employment_since,
    family = binomial(), data = german)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3904  -0.9214  -0.5038   1.1211   2.2275

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                      0.454769   0.290772   1.564 0.117817
Status_of_checking_accountp_lo -0.459128   0.176341  -2.604 0.009224 **
Status_of_checking_accountp_hi -1.252733   0.330261  -3.793 0.000149 ***
Status_of_checking_accountnone -1.969903   0.199575  -9.870  < 2e-16 ***
```

# Example: German Data Set / 2

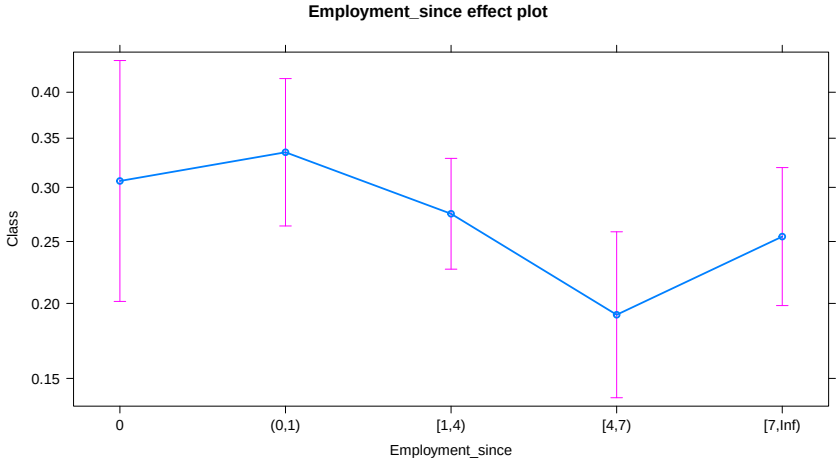
```
Age                          -0.013322   0.007293  -1.827 0.067756 .
Employment_since.L           -0.402007   0.218513  -1.840 0.065806 .
Employment_since.Q            0.073814   0.208236   0.354 0.722985
Employment_since.C            0.395661   0.190381   2.078 0.037686 *
Employment_since^4            0.093241   0.157964   0.590 0.555012
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1075.3  on 991  degrees of freedom
AIC: 1093.3

Number of Fisher Scoring iterations: 4
```

# Example: German Data Set / 3

```
R> eff3 <- Effect("Employment_since", model3)
R> eff3

 Employment_since effect
Employment_since
        0     (0,1)     [1,4)     [4,7)   [7,Inf)
0.3062719 0.3353066 0.2748948 0.1916990 0.2542879
```

# Example: German Data Set / 4

```
R> plot(eff3)
```



# Example: German Data Set / 5

Likelihood-ratio test:

```
R> anova(model2, model3, test = "Chisq")

Analysis of Deviance Table

Model 1: Class ~ Status_of_checking_account + Age
Model 2: Class ~ Status_of_checking_account + Age + Employment_since
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       995     1085.0
2       991     1075.3  4   9.6866  0.04605 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Example: German Data Set / 6

Score test:

```
R> anova(model2, model3, test = "Rao")

Analysis of Deviance Table

Model 1: Class ~ Status_of_checking_account + Age
Model 2: Class ~ Status_of_checking_account + Age + Employment_since
  Resid. Df Resid. Dev Df Deviance   Rao Pr(>Chi)
1       995     1085.0
2       991     1075.3  4   9.6866 9.552  0.04869 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Example: German Data Set / 7

Wald test:

```
R> library("lmtest")
R> waldtest(model2, model3, test = "Chisq")
```

```
Wald test

Model 1: Class ~ Status_of_checking_account + Age
Model 2: Class ~ Status_of_checking_account + Age + Employment_since
  Res.Df Df  Chisq Pr(>Chisq)
1    995
2    991  4 9.4344    0.05111 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Criteria for Model Fit and Model Selection

- Assessing the fit of an estimated model relies on the following idea: When the data have been maximally grouped, we can estimate the group-specific parameter $\mu_i$ using the mean value $\bar{y}_i$.
- The use of these mean values as estimators corresponds to the **saturated model**.

# Criteria for Model Fit and Model Selection / 2

- Global statistics to verify the fit of a model relative to the saturated model:
  - **Pearson statistic**

$$\chi^2 = \sum_{i=1}^{G} \frac{(y_i - \hat{\mu}_i)^2}{v(\hat{\mu}_i)/n_i}$$

  - **Deviance**

$$D = -2 \sum_{i=1}^{G} \{\ell_i(\hat{\mu}_i) - \ell_i(\bar{y}_i)\}$$

  $\hat{\mu}_i$ and $v(\hat{\mu}_i)$ are the estimated expectations and variance functions.
- For finite samples, the Pearson and the deviance statistic will differ, but it can be shown that they are asymptotically equivalent for grouped data.
- For both model fit statistics, the data should be grouped as much as possible.
- When $n_i$ is sufficiently large in **all** groups $i = 1, \ldots, G$, both statistics are approximately or asymptotically (for $n \to \infty$) $\psi\chi^2(G - p)$-distributed.
- $p$ denotes the number of estimated parameters.