MSG500/MVE190
Linear Models - Lecture 11

Rebecka Jörnsten
Mathematical Statistics
University of Gothenburg/Chalmers University of Technology

November 29, 2012

1 Binary response

So far, we have assumed that the response variable y is continuous and that the expected value of y_i is

$$E[y_i] = \sum_{j=0}^{p-1} \beta_j x_{ij},$$

where x_{i0} is a vector of ones, such that β_0 is the intercept. If y_i only takes on values 0 or 1, this modeling assumption is inadequate. In Figure 1 I depict a linear regression model fit to 0/1 response data. The

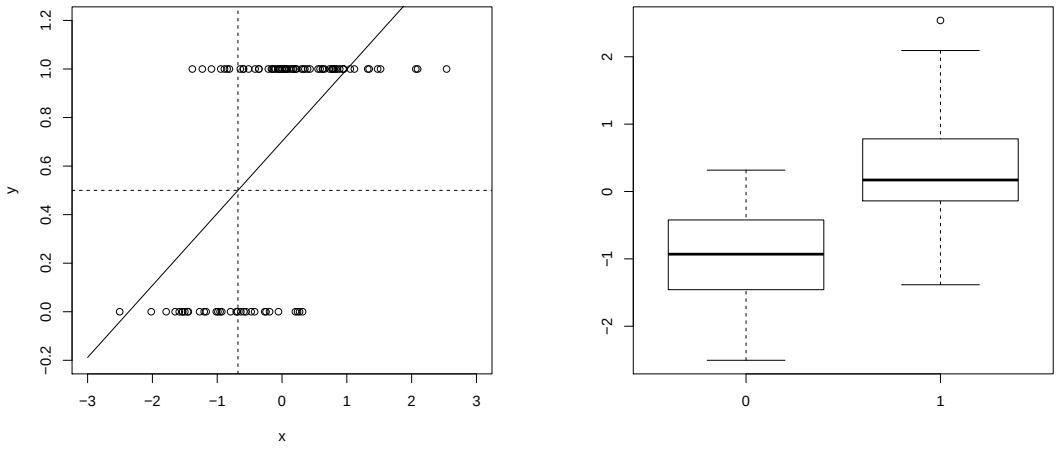

Figure 1: 0/1 regression

dotted horizontal line corresponds to fitted value 0.5, and the vertical dotted line the x-value that gives fitted value 0.5. That is, for x-values to the right of the vertical dotted line, the fitted values are closer to outcome $y = 1$. This vertical line is also referred to as the *decision boundary* (see Multivariate class). The right panel is a boxplot that shows how the distribution of the x-variable differs when $y = 0$ or $y = 1$. As you can see, there is a mean shift toward higher x-values for $y = 1$.

Now, there is a problem with 0/1 regression model; we obtain predictions outside the range $[0, 1]$ since we assume $E[y]$ is linear in x. When y is a random variable that takes on values 0 and 1 only, with some underlying probability π, this means that $E[y] = P(y = 1) = \pi$, and the model values should lie in interval $[0, 1]$. Our "fix" for this problem is to find a transformation such that the model fit is constrained to lie in this range.

2 Logistic regression

Logistic regression is one such transform. We assume that $\pi_i = P(y_i = 1|x_i)$, i.e. that each observations y_i is associated with a probability of attaining the value 1 that depends on the value of the explanatory, x_i. We further assume that this dependency can be modeled as

$$logit(P(y_i = 1|x_i)) = \frac{\log(P(y_i = 1|x_i))}{\log(1 - P(y_i = 1|x_i))} = \text{"log-odds"} = \sum_{j=0}^{p-1} \beta_j x_{ij}. \quad (1)$$

That is,

$$\frac{P(y_i = 1|x_i)}{P(y_i = 0|x_i)} = \text{OR,odds-ratio} = e^{\sum_{j=0}^{p-1} \beta_j x_{ij}} \quad (2)$$

or

$$P(y_i = 1|x_i) = \frac{e^{\sum_{j=0}^{p-1} \beta_j x_{ij}}}{1 + e^{\sum_{j=0}^{p-1} \beta_j x_{ij}}}. \quad (3)$$

This last expression guarantees that the probability $P(y_i = 1|x_i)$ is between 0 and 1. Equation (1) is a modeling statement for the log-odds, equation (2) for the odds-ratio and equation (3) for the probability level model.

The impact of and x-variable on the response y is interpreted differently in the logistic setting compared with a standard linear model. If we increase x_j by one unit, holding all other x's fixed, the odds-ratio (relative probability between outcome 1 and 0) increases by a multiplicative factor e^{β_j}. A positive β_j means that increasing x_j increases the probability of an outcome $y = 1$ and vice versa. The intercept, β_0 shifts the mean probability of $y = 1$ upward or downward, depending on the sign.

What does this model look like? In Figure 2 I depict the model using equation (3), response level (or

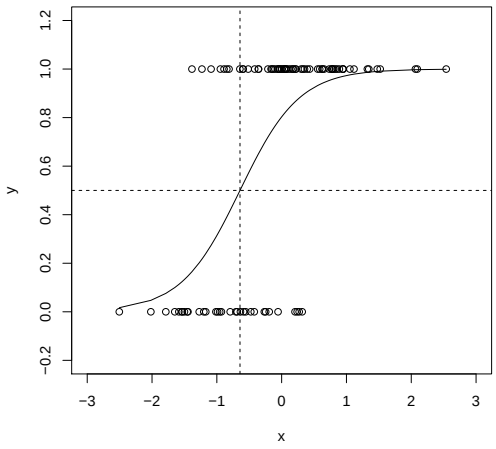

Figure 2: Dose-response curve. Decision boundary (black dotted).

probability). This is also commonly referred to as dose-response curve, where the "dose" is the value of the x, and the response a probability of an outcome "1" (stems from clinical research).

In Figure 3 you can see how the sign and magnitude of β changes the dose-response curve. The magnitude of β determines how sharp the curve is, i.e. how much knowing the value of x helps us determine the outcome (or how separable the $y = 1$ and $y = 0$ data sets are along the x. Note, if we have more than one x-variable in the model, the does response curve has $x\beta$ as its x-axis. This $\eta = x\beta$ is called the *linear predictor*.

Where does this logit transform come from? A natural distribution assumption for 0/1 outcomes is a binomial distribution $y_i \sim Bin(1, \pi_i)$. Under this distribution, $E[y_i] = \pi_i$ and $V[y_i] = \pi_i(1 - \pi_i)$.

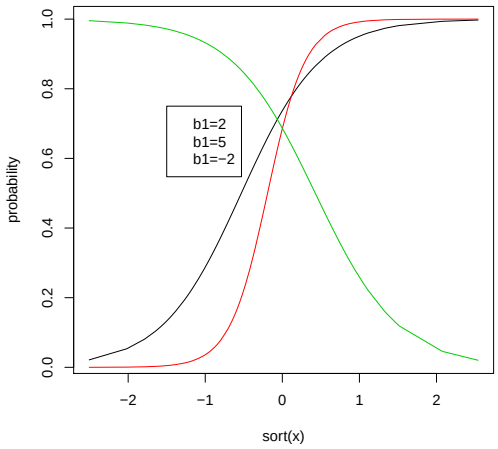

Figure 3: The impact of β on the shape of the dose-response curve.

(Note, that means we no longer have a constant variance for observations y_i. This will have an impact on how we fit the model to the data (we will look into this next lecture).) We do assume that the y_i are independent, just as we did in the linear model case, and also that there are no outliers.

We can write the likelihood for the data as follows:

$$L = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i}.$$

We write the log-likelihood

$$logL = \sum_{i=1}^{n} y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i) = \sum_{i=1}^{n} y_i \log(\frac{\pi_i}{1 - \pi_i}) + \log(1 - \pi_i).$$

You can see the logit transform appearing the in the log-likelihood expression. Assuming a binomial distribution for the data thus leads to a logit transformation. However, we can consider other transformations h of $x\beta$ such that we get a response fit $p(y|x) = h(x\beta)$ in the range $[0, 1]$, or a transformation $g(p(y|x)) = x\beta$. The transformation $g(p) = h^{-1}(p) = x\beta$ is called the *link* function. An additional assumption in the modeling of 0/1 response is thus that the appropriate *link* that connects response probability and x-variables is the logit.

Assumptions:

- y_i independent
- no outliers
- $E(y_i) = \pi_i$, $V(y_i) = \pi_i(1 - \pi_i)$
- $logit(\pi_i) = x_i\beta$

As mentioned above, when we have multiple x-variables in the model, the dose-response curve is most frequently presented with a constructed x-axis given by $\eta = x\hat{\beta}$ (the linear predictor). The basic model assumes an additive structure. However, you can also include interactions in a logistic model. We simulate data from an additive model where $logit(P(y_i = 1|x_i)) = 1 + 1 * x_1 + 2 * x_2$, where $x_1 \sim N(0, 1)$ and $x_2 \sim Bin(1, .5)$. In Figure 4 I depict the log-odds $(x\hat{\beta})$ as a function of x_1, with two different curves corresponding to $x2 = 0$ and $x2 = 1$. For an additive model, the log-odds curves are parallel. In the right panel I depict the corresponding dose-response curves.

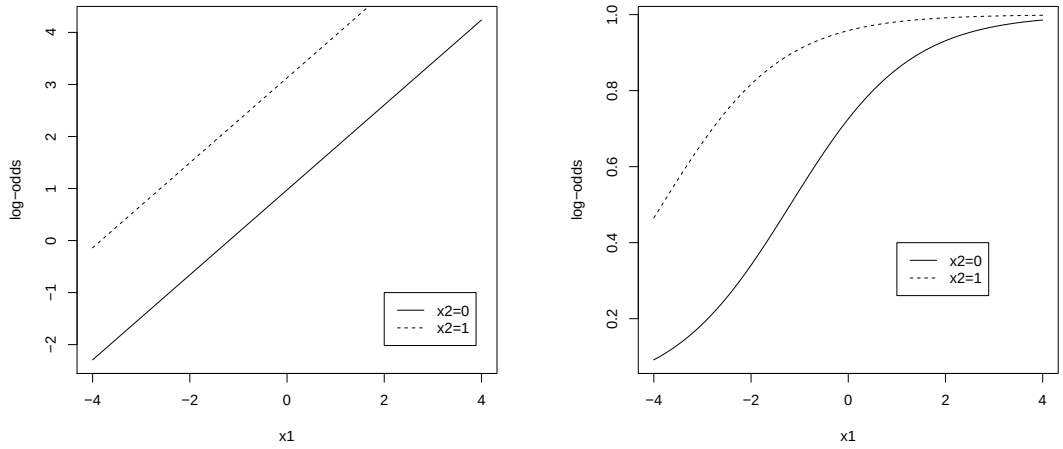

Figure 4: Additive model: log-odds and dose-response curves

We then simulate from a model with an interaction: $logit(P(y_i = 1|x_i)) = 1 + x1 + 2 * x2 + 2 * x1 * x2$. In Figure 5 you see that the log-odds is the easier scale to detect interactions. Different slopes suggest an interaction is needed to capture the structure of the data.

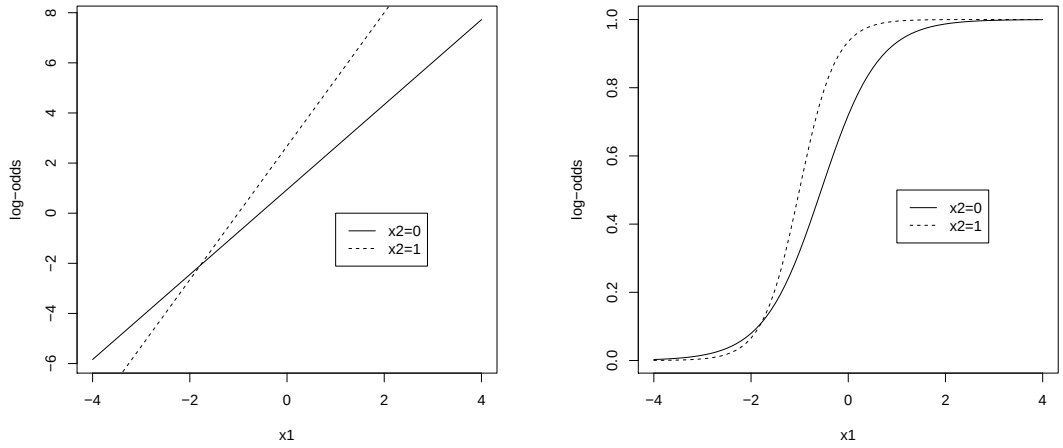

Figure 5: Interaction model: log-odds and dose-response curves

The additive model means that log-odds profiles are shifted vertically by changing a value of one x-variable, holding the other x's fixed. Another way of stating this is looking at the odds-ratios. The impact on the odds-ratio of changing both x_1 and x_2 by one unit can be separated into the impact of each in a multiplicative fashion: that is $OR(x_1 + 1, x_2 + 1) = OR(x_1 + 1) * OR(x_2 + 1)$. We can compare the observed odds-ratio to the product of the odds-ratios increasing only one of the x's. If these expressions don't agree, we have an interaction. (This is something you have already seen before in terms of Chi-square test of 2*2 tables.)

2.1 Diagnostics

Diagnostics of logistic regression model is more complicated than linear models. The residuals are more difficult to interpret. Still, a diagnostic analysis of the residuals should reveal that there are no trends -

but it's not so easy to spot this directly as you will see below.

Residuals come in several flavors in the logistic regression setting. We first have the response residuals $y_i - \hat{\pi}_i$, where $logit(\hat{\pi}_i) = x_i\hat{\beta}$. There are difficult to interpret since all y_i have different variance ($V(y_i) = \pi_i(1-\pi_i)$). It is therefore more common to use the so-called *Pearson residuals*:

$$r_i = \frac{y_i - \hat{\pi}_i}{\sqrt{\pi_i(1-pi_i)}}.$$

The Pearson residuals have the same variance so are directly comparable. It is common to consider $|r_i| > 2$ as large residuals. We check the fit of the model by plotting the Pearson residuals versus the linear predictor η

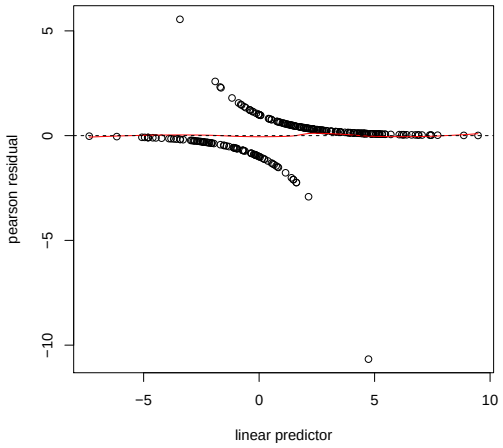

Figure 6: Pearson residuals. The red line is a local average model fit.

In Figure 6 I depict a Pearon residual plot. The x-axis is the fitted values (linear predictor) $x\hat{\beta}$. I have added the x-axis (black horizontal line) and a smooth local average fit (red line). You can spot a problem with the fit by comparing the smooth fit to the horizontal line. If the smooth local fit exhibits a trend, the model does not fit the data and we may need to explore transformations of the x-variables, or perhaps a different link or distribution for the response y.

Another set of residuals commonly used for diagnostics are the *deviance residuals*. These measure the influence or contribution of each observation to the log-likelihood. -2 times the log-likelihood can be written as

$$-2\sum_i y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i) = \sum_i d_i^2,$$

where we define the deviance residual as

$$d_i = \pm\sqrt{-2(y_i \log(\hat{\pi}_i) + (1-y_i)\log(1-\hat{\pi}_i))},$$

with the sign of d_i determined by the sign of $y_i - \hat{\pi}_i$. We consider values of $|d_i| > 2$ influential. Figure 7 depics the deviance residuals.

Other diagnostic tools of interest include the leverage, diagonal elements of the hat-matrix. This do describe how much an observation's x-value influences the fit, but is a bit more difficult to interpret for logistic regression models since their impact on the fit are down-weighted some for π_i close to 0 or 1. We can also combine residual values and leverage into the Cook's distance:

$$D_i = \frac{r_i^2 h_{ii}}{(1-h_{ii})^2}$$

In Figure 8 you see that for the simulated data, no observations stand out in terms of leverage or Cook's

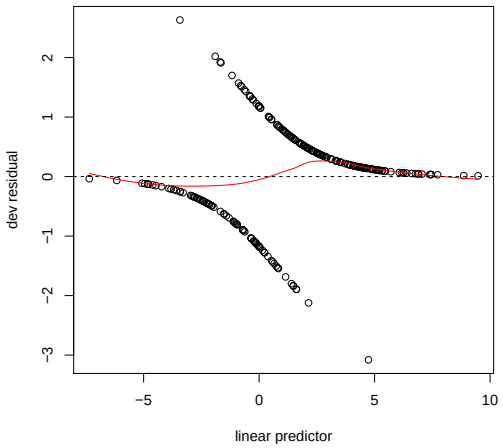

Figure 7: Deviance residuals. The red line is a local average model fit.

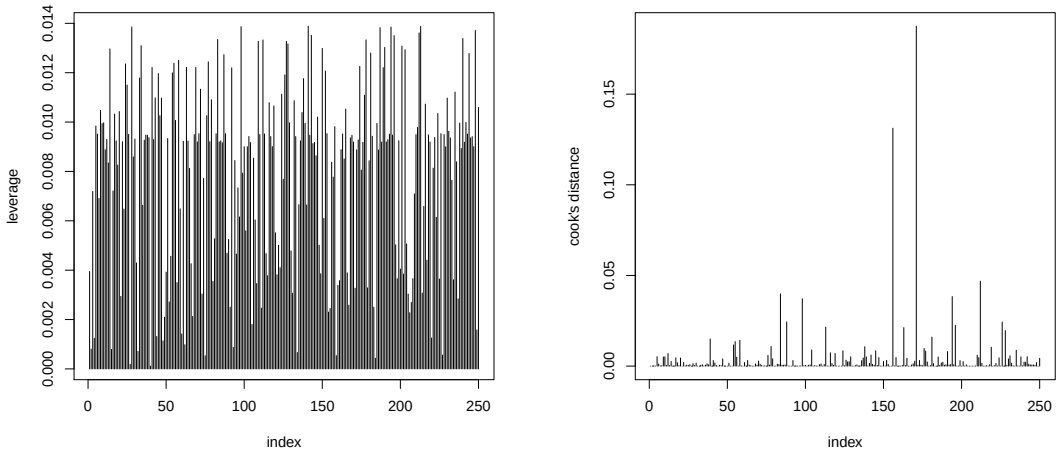

Figure 8: Leverage and Cook's distance

distance.

The generalization of the change in σ^2 (RSS) to logistic regression is called *Deviance change*. The deviance is given by -2*log-likelihood. One can show that the change in deviance (Figure 9) we see when we drop observation i can be approximated by

$$dev.change_i = d_i^2 + r_i^2 \frac{h_{ii}}{1 - h_{ii}}.$$

Finally, we have to check that the logit link is appropriate for our data. To do this we want to compare the actual observed impact of x on the $p(y = 1)$ and compare that to the assumed logit scale. We bin the x values into groups, and compute the proportion of $y = 1$ in each bin. We then plots the logit of these proportions against the grouped x-values. The plot should exhibit a linear trend if the logit link is appropriate. Curvature in the plot means that we have to consider transformations of x or possibly another type of link function.

In Figure 10 (left panel) I show an example of a link diagnostic plot that reveals an inadequate fit. After applying a log-transformation to the x-variable the link seems works much better.

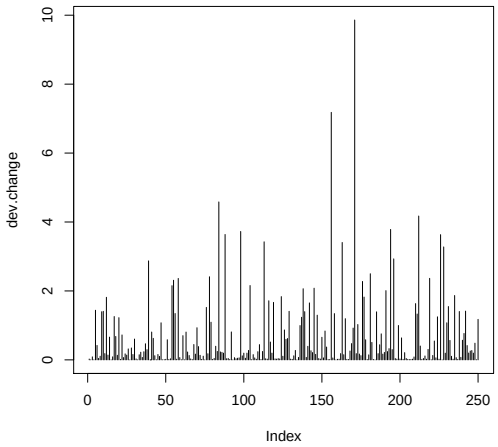

Figure 9: Change in deviance

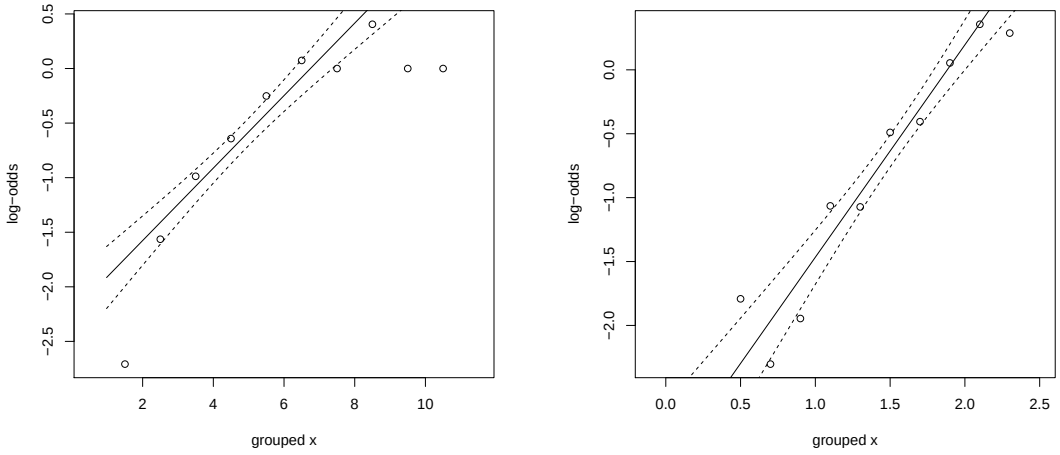

Figure 10: Link test. Left - inadequate. Right - adequate

2.2 Validation and Inference

The goodness-of-fit F-test used in linear regression is replaced with a Chi-square test on the Deviance = -2 loglikelihood. Now, in the normal error distribution model, with known error variance σ^2, the Deviance is just the RSS/σ^2 and is thus distributed χ^2 with $n-p$ degrees of freedom of the model specification is correct. When y is not normally distributed, the Deviance is only approximately distributed as χ^2_{n-p}, with the approximation being more accurate for large n. For small sample sizes the approximation can be quite off. If the Chi-square test leads to a rejection of the goodness-of-fit we have to consider the possibility that we may need to transform the x-variables, include omitted x-variables or perhaps use a different model (link or distribution for y).

Inference for model coefficient are more difficult for in logistic regression. The distribution of $\hat{\beta}_j$ can be approximated by a normal distribution for large n, and z-tests can thus be used to test each hypothesis $\beta_j = 0$. The same problems with these marginal tests, in terms of multiple testing or collinearity, exist in logistic regression as we have encountered in linear regression, so use these tests with caution. In addition, logistic regression can "crash" when the separation of $y = 1$ and $y = 0$ along an x-variable is near complete. You will obtain accurate estimates for π_i *but* highly unstable estimates for β. The reason

is because almost any dose-response curve fits the data equally well. This will be reflected in very large values for the standard errors of $\hat{beta}$.

3 Demo 11

We revisit the South African heart disease data, now treating chronic heart disease (chd) as the outcome variable. Let us first look at a

```
> SA <- data.frame(read.table("SA.dat", sep = "\t", header = T))
> xt <- SA$age
> y <- SA$chd
> hh <- hist(xt, plot = F)
> x2 <- xt
> for (kk in (1:length(hh$breaks) - 1)) {
+     x2[xt <= hh$breaks[kk + 1] & xt > hh$breaks[kk]] <- hh$mid[kk]
+ }
> tt <- table(y, x2)
> ttt <- logit(tt[2, ]/apply(tt, 2, sum))
> plot(sort(unique(x2)), ttt, xlab = "age", ylab = "log-odds")
> gg <- glm(y ~ xt, "binomial")
> pp <- predict(gg, response = "link", se = T)
> lines(sort(xt), pp$fit[sort.list(xt)])
> lines(sort(xt), (pp$fit - pp$se)[sort.list(xt)], lty = 2)
> lines(sort(xt), (pp$fit + pp$se)[sort.list(xt)], lty = 2)

> xt <- log(SA$age)
> hh <- hist(xt, plot = F)
> x2 <- xt
> for (kk in (1:length(hh$breaks) - 1)) {
+     x2[xt <= hh$breaks[kk + 1] & xt > hh$breaks[kk]] <- hh$mid[kk]
+ }
> tt <- table(y, x2)
> ttt <- logit(tt[2, ]/apply(tt, 2, sum))
> plot(sort(unique(x2)), ttt, xlab = "log(age)", ylab = "log-odds")
> gg <- glm(y ~ xt, "binomial")
> pp <- predict(gg, response = "link", se = T)
> lines(sort(xt), pp$fit[sort.list(xt)])
> lines(sort(xt), (pp$fit - pp$se)[sort.list(xt)], lty = 2)
> lines(sort(xt), (pp$fit + pp$se)[sort.list(xt)], lty = 2)
```

In Figure 11 I try first age and then log(age) as a predictor of the probability of having a diagnosis of heart disease. The right panel (log(age)) agrees better with a linear log-odds assumption of log(age) than using age as a predictor. Cycling through the other variables, I decide to also use log(ldl).

We fit a model using all the predictor variables:

```
> gg <- glm(chd ~ log(ldl) + log(age) + alcohol + tobind + sbp +
+     typea + adiposity + obesity + as.factor(famhist), data = SA,
+     "binomial")
> print(gs <- summary(gg))

Call:
glm(formula = chd ~ log(ldl) + log(age) + alcohol + tobind +
    sbp + typea + adiposity + obesity + as.factor(famhist), family = "binomial",
    data = SA)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7972  -0.8210  -0.4236   0.9044   2.9618
```

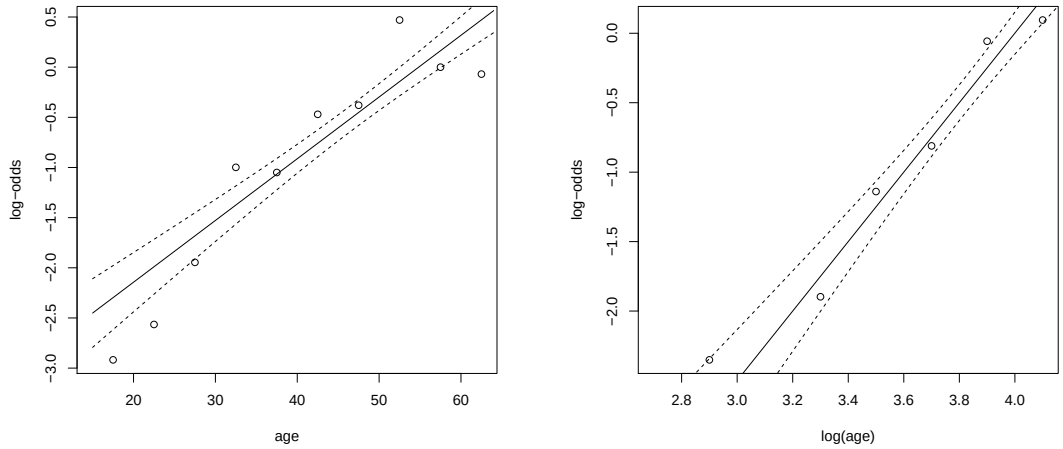

Figure 11: Link test. Left - age. Right - log(age)

```
Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -13.369488   2.645028  -5.055 4.31e-07 ***
log(ldl)             1.370256   0.414935   3.302 0.000959 ***
log(age)             2.204754   0.600493   3.672 0.000241 ***
alcohol              0.006429   0.005965   1.078 0.281151
tobind               0.764701   0.410127   1.865 0.062245 .
sbp                  0.007167   0.006494   1.104 0.269770
typea                0.038939   0.014722   2.645 0.008168 **
adiposity           -0.003801   0.036612  -0.104 0.917323
obesity             -0.062509   0.053859  -1.161 0.245808
as.factor(famhist)2  0.687097   0.279471   2.459 0.013950 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 401.21  on 311  degrees of freedom
Residual deviance: 314.10  on 302  degrees of freedom
AIC: 334.1

Number of Fisher Scoring iterations: 5
```

The Chi-square goodness-of-fit test compares the deviance 314.103 to the Chi-square distribution with 302. The p-value $P(\chi^2_{302} > 314.103) = 0.304$. We do not reject the fit of the logistic model.

We examine residual diagnostic plots next.

```
> res <- residuals(gg, "pearson")
> plot(logit(gg$fit), res, xlab = "linear predictor", ylab = "dev residual")
> ll <- loess(res[sort.list(logit(gg$fit))] ~ sort(logit(gg$fit)))
> lines(ll$x, ll$fit, col = 2)
> abline(h = 0, lty = 2)

> res <- residuals(gg, "deviance")
> plot(logit(gg$fit), res, xlab = "linear predictor", ylab = "dev residual")
> ll <- loess(res[sort.list(logit(gg$fit))] ~ sort(logit(gg$fit)))
```

```
> lines(ll$x, ll$fit, col = 2)
> abline(h = 0, lty = 2)

> hh <- hatvalues(gg)
> plot(hh, type = "h", ylab = "leverage", xlab = "index")

> library(stats)
> dd <- cooks.distance(gg)
> plot(dd, type = "h", ylab = "cook's distance", xlab = "index")
```

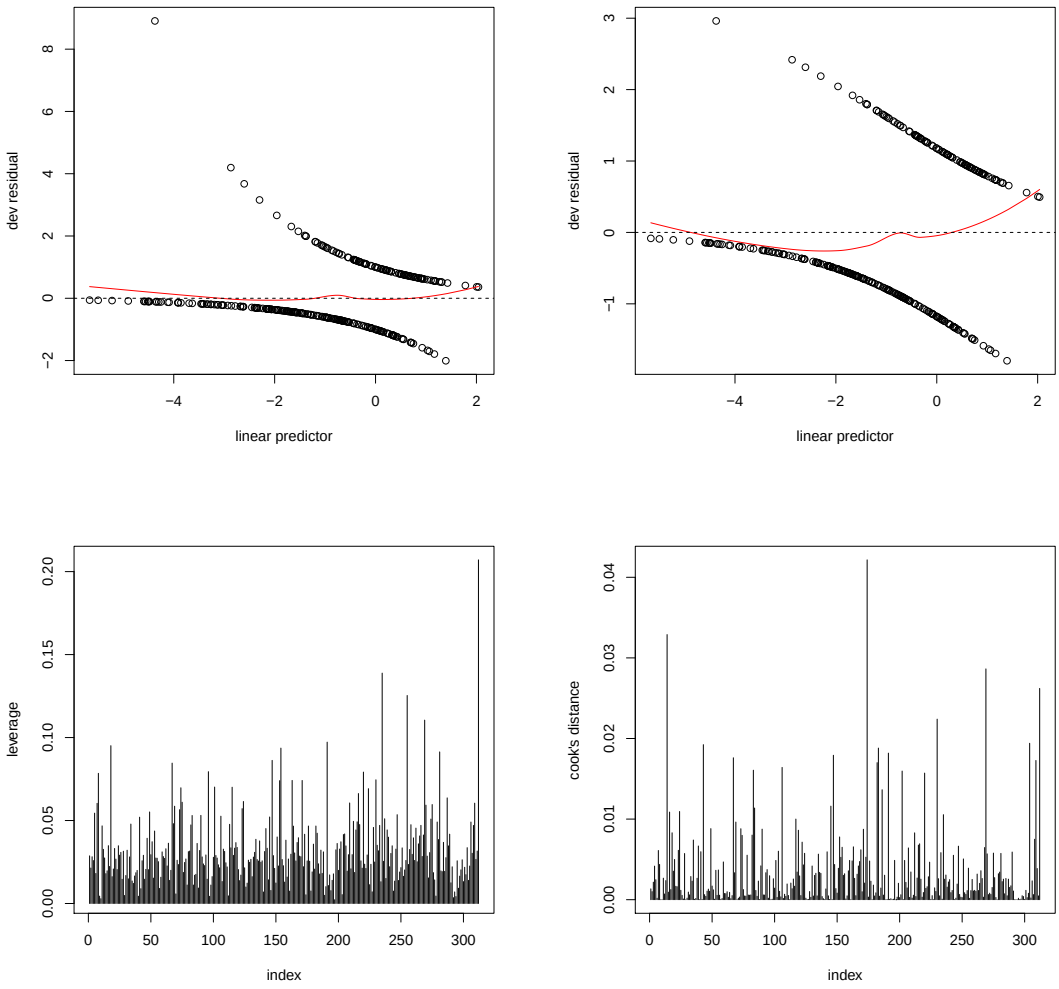

Figure 12: Diagnostic plots. Top-Left: Pearson residuals. Top-Right: Deviance residuals. Bottom-Left: Leverage. Bottom-right: Cook's distance.

The `glm()` function used to fit logistic models in R also has its own built in diagnostic plots.

```
> par(mfrow = c(2, 2))
> plot(gg)
```

In Figure 13 we see four panels of residuals. The top left panel is the pearson residual plot. The top right panel is a QQplot comparing the pearson residuals to a normal distribution (it may be relevant for e.g. Poisson models that sometimes can be well approximated by a normal distribution, but in general we don't expect to see a good agreement here). The bottom left panel is, just like in the linear model case, a way to check the variance of y as a function of the fitted value. For binomial models we expect the variance to be non-constant and maximized at $\pi = .5$ corresponding to the linear predictor equal to

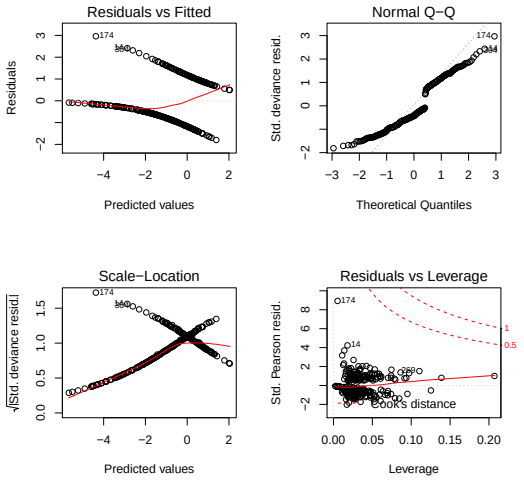

Figure 13: Diagnostic plots.

0. The bottom right panel combines the leverage and the pearson residuals. Extremes in either or both directions constitutes a large Cook's distance, indicated by red, dashed lines in the plot.

```
> par(mfrow = c(1, 1))
> dev.change <- residuals(gg, "deviance")^2 + residuals(gg, "pearson")^2 *
+     (hh/(1 - hh))
> plot(seq(1, dim(SA)[1]), dev.change, type = "h")
> id <- identify(dev.change, pos = T)
```

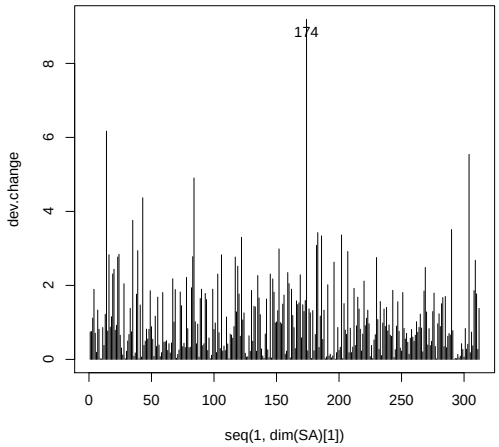

Figure 14: Diagnostic plot. Change in deviance

The change of deviance plot looks for observations that impact the overall measure of goodness-of-fit (similarly to the change of σ^2 in linear regression. Here, observaion 174 is identified as possible outliers. I will not drop it here, but leave that for you to try out at home using the R code for this demo.

Using the z-test, tabulated in the model summary above, we reject the null that $\beta_j = 0$ for the intercept, log(ldl), log(age) and typea behaviour. Next lecture we will discuss model selection, but for

now we try a simple backward test based on the AIC. The AIC is defined as

$$AIC = Deviance + 2p.$$

```
> print(step(gg))

Start:  AIC=334.1
chd ~ log(ldl) + log(age) + alcohol + tobind + sbp + typea +
    adiposity + obesity + as.factor(famhist)

                     Df Deviance    AIC
- adiposity           1   314.11 332.11
- alcohol             1   315.26 333.26
- sbp                 1   315.33 333.33
- obesity             1   315.47 333.47
<none>                    314.10 334.10
- tobind              1   317.79 335.79
- as.factor(famhist)  1   320.19 338.19
- typea               1   321.54 339.54
- log(ldl)            1   325.79 343.79
- log(age)            1   329.30 347.30

Step:  AIC=332.11
chd ~ log(ldl) + log(age) + alcohol + tobind + sbp + typea +
    obesity + as.factor(famhist)

                     Df Deviance    AIC
- alcohol             1   315.27 331.27
- sbp                 1   315.34 331.34
<none>                    314.11 332.11
- obesity             1   317.43 333.43
- tobind              1   317.94 333.94
- as.factor(famhist)  1   320.26 336.26
- typea               1   321.64 337.64
- log(ldl)            1   326.83 342.83
- log(age)            1   335.23 351.23

Step:  AIC=331.27
chd ~ log(ldl) + log(age) + tobind + sbp + typea + obesity +
    as.factor(famhist)

                     Df Deviance    AIC
- sbp                 1   316.78 330.78
<none>                    315.27 331.27
- obesity             1   318.33 332.33
- tobind              1   319.81 333.81
- as.factor(famhist)  1   321.71 335.71
- typea               1   323.08 337.08
- log(ldl)            1   326.88 340.88
- log(age)            1   336.86 350.86

Step:  AIC=330.78
chd ~ log(ldl) + log(age) + tobind + typea + obesity + as.factor(famhist)

                     Df Deviance    AIC
<none>                    316.78 330.78
- obesity             1   319.42 331.42
- tobind              1   321.42 333.42
- as.factor(famhist)  1   323.05 335.05
```

```
- typea               1   324.64 336.64
- log(ldl)            1   328.66 340.66
- log(age)            1   344.30 356.30

Call:  glm(formula = chd ~ log(ldl) + log(age) + tobind + typea + obesity +
    as.factor(famhist), family = "binomial", data = SA)

Coefficients:
        (Intercept)             log(ldl)             log(age)
          -12.88888              1.26657              2.33510
             tobind                typea              obesity
            0.83374              0.03976             -0.05872
as.factor(famhist)2
            0.69103

Degrees of Freedom: 311 Total (i.e. Null);  305 Residual
Null Deviance:            401.2
Residual Deviance: 316.8         AIC: 330.8
```

```
> selg <- step(gg, trace = F)
> print(summary(selg))
```

```
Call:
glm(formula = chd ~ log(ldl) + log(age) + tobind + typea + obesity +
    as.factor(famhist), family = "binomial", data = SA)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8143  -0.8413  -0.4095   0.9338   2.9389

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         -12.88888    2.22957  -5.781 7.43e-09 ***
log(ldl)              1.26657    0.38026   3.331 0.000866 ***
log(age)              2.33510    0.49365   4.730 2.24e-06 ***
tobind                0.83374    0.40123   2.078 0.037710 *
typea                 0.03976    0.01465   2.715 0.006629 **
obesity              -0.05872    0.03656  -1.606 0.108241
as.factor(famhist)2   0.69103    0.27675   2.497 0.012525 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 401.21  on 311  degrees of freedom
Residual deviance: 316.78  on 305  degrees of freedom
AIC: 330.78

Number of Fisher Scoring iterations: 5
```

We can use the selected model to construct a dose-response curve.

```
> plot(logit(selg$fit), SA$chd, xlab = "linear predictor", ylab = "chd")
> lines(sort(logit(selg$fit)), selg$fit[sort.list(selg$fit)])
```

In Figure 15 we see the model for $p(chd = 1|x)$ as a function $x\hat{\beta}$. If we use the decision boundary corresponding to probability 0.5 we can check if the logistic model can classify patients in terms of heart disease.

```
> pp <- predict(selg, type = "response")
> pp[pp < 0.5] <- 0
```

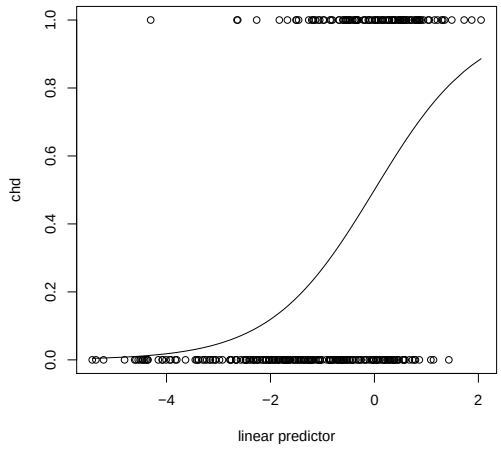

Figure 15: Dose-response, selected model.

```
> pp[pp >= 0.5] <- 1
> table(pp, SA$chd)

pp    0   1
  0 170  47
  1  35  60

> print(c("Misclassification error=", round(sum(pp != SA$chd)/length(SA$chd),
+     3)))

[1] "Misclassification error=" "0.263"
```

The error rate is not so impressive with more than 1/4 of the data being mislabeled. In addition, this is an optimistic estimate since the same data was used for fitting as well as classification. Next lecture we will use training and test data to get a better estimate of the predictive performance of the logistic model.

Should we include interactions in the model?

```
> par(mfrow = c(2, 2))
> gg <- glm(chd ~ log(age) * as.factor(famhist), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(log(SA$age)), max(log(SA$age)), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "log(age)",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
> gg <- glm(chd ~ log(ldl) * as.factor(famhist), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(log(SA$ldl)), max(log(SA$ldl)), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "log(ldl)",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
> gg <- glm(chd ~ typea * as.factor(famhist), "binomial", data = SA)
> pp <- predict(gg, type = "response")
```

```
> xvec <- seq(min(SA$typea), max(SA$typea), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "typea",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
> gg <- glm(chd ~ obesity * as.factor(famhist), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(SA$obesity), max(SA$obesity), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "obesity",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
```

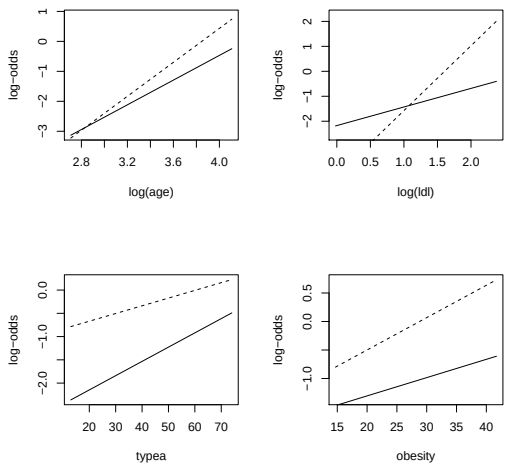

Figure 16: Interactions with famhist.

In Figure 16, we identify a possible interaction between `famhist` and `log(ldl)` (You can check the significance of the the interaction in each of the above fits).

```
> par(mfrow = c(2, 2))
> gg <- glm(chd ~ log(age) * as.factor(tobind), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(log(SA$age)), max(log(SA$age)), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "log(age)",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
> gg <- glm(chd ~ log(ldl) * as.factor(tobind), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(log(SA$ldl)), max(log(SA$ldl)), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "log(ldl)",
```

```
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
> gg <- glm(chd ~ typea * as.factor(tobind), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(SA$typea), max(SA$typea), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "typea",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
> gg <- glm(chd ~ obesity * as.factor(tobind), "binomial", data = SA)
> pp <- predict(gg, type = "response")
> xvec <- seq(min(SA$obesity), max(SA$obesity), by = 0.1)
> p0 <- gg$coef[1] + gg$coef[2] * xvec
> p1 <- gg$coef[1] + gg$coef[2] * xvec + gg$coef[3] + gg$coef[4] *
+     xvec
> plot(xvec, p0, type = "l", ylab = "log-odds", xlab = "obesity",
+     ylim = c(min(logit(gg$fit)), max(logit(gg$fit))))
> lines(xvec, p1, lty = 2)
```

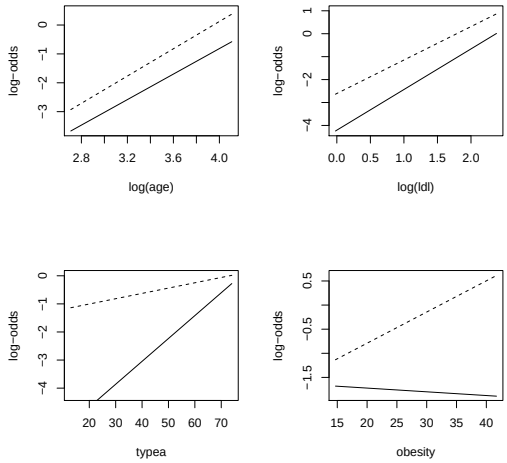

Figure 17: Interactions with tobind.

In Figure 17, we identify a possible interaction between `tobind` and `obesity` but a closer inspection reveals that the interaction is not significant.

We try a model including interactions with `famhist`:

```
> gg <- glm(chd ~ as.factor(famhist) * (log(ldl) + log(age) + sbp +
+     adiposity + obesity + alcohol + typea + as.factor(tobind)),
+     "binomial", data = SA)
> print(summary(gg))

Call:
glm(formula = chd ~ as.factor(famhist) * (log(ldl) + log(age) +
    sbp + adiposity + obesity + alcohol + typea + as.factor(tobind)),
    family = "binomial", data = SA)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.0261  -0.7714  -0.3869   0.8199   2.6188

Coefficients:
                                      Estimate Std. Error z value Pr(>|z|)
(Intercept)                         -10.714938   3.358056  -3.191  0.00142
as.factor(famhist)2                  -6.243649   5.740262  -1.088  0.27673
log(ldl)                              0.421541   0.554816   0.760  0.44738
log(age)                              1.967974   0.747764   2.632  0.00849
sbp                                   0.005175   0.008904   0.581  0.56108
adiposity                             0.014435   0.056869   0.254  0.79962
obesity                              -0.091440   0.088341  -1.035  0.30063
alcohol                               0.001644   0.009212   0.178  0.85834
typea                                 0.046484   0.020910   2.223  0.02621
as.factor(tobind)1                    0.628661   0.575134   1.093  0.27436
as.factor(famhist)2:log(ldl)          2.265445   0.887209   2.553  0.01067
as.factor(famhist)2:log(age)          0.657307   1.294304   0.508  0.61156
as.factor(famhist)2:sbp               0.006074   0.013887   0.437  0.66182
as.factor(famhist)2:adiposity        -0.024917   0.076850  -0.324  0.74576
as.factor(famhist)2:obesity           0.038949   0.114037   0.342  0.73269
as.factor(famhist)2:alcohol           0.010272   0.012702   0.809  0.41869
as.factor(famhist)2:typea            -0.017587   0.030699  -0.573  0.56672
as.factor(famhist)2:as.factor(tobind)1   0.538570   0.837941   0.643  0.52040

(Intercept)                         **
as.factor(famhist)2
log(ldl)
log(age)                            **
sbp
adiposity
obesity
alcohol
typea                               *
as.factor(tobind)1
as.factor(famhist)2:log(ldl)        *
as.factor(famhist)2:log(age)
as.factor(famhist)2:sbp
as.factor(famhist)2:adiposity
as.factor(famhist)2:obesity
as.factor(famhist)2:alcohol
as.factor(famhist)2:typea
as.factor(famhist)2:as.factor(tobind)1
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 401.21  on 311  degrees of freedom
Residual deviance: 302.81  on 294  degrees of freedom
AIC: 338.81

Number of Fisher Scoring iterations: 5

> selg <- step(gg, trace = F)
> print(summary(selg))

Call:
glm(formula = chd ~ as.factor(famhist) + log(ldl) + log(age) +
    obesity + typea + as.factor(tobind) + as.factor(famhist):log(ldl),
    family = "binomial", data = SA)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9788  -0.7976  -0.4027   0.8951   2.7723

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -11.50579    2.26251  -5.085 3.67e-07 ***
as.factor(famhist)2         -2.57988    1.16725  -2.210  0.02709 *
log(ldl)                     0.33347    0.47788   0.698  0.48530
log(age)                     2.33524    0.49617   4.707 2.52e-06 ***
obesity                     -0.05851    0.03677  -1.591  0.11154
typea                        0.03792    0.01497   2.534  0.01129 *
as.factor(tobind)1           0.94049    0.41275   2.279  0.02269 *
as.factor(famhist)2:log(ldl)   2.12788    0.73656   2.889  0.00387 **
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 401.21  on 311  degrees of freedom
Residual deviance: 307.97  on 304  degrees of freedom
AIC: 323.97

Number of Fisher Scoring iterations: 5
```

```
> par(mfrow = c(2, 2))
> plot(selg)
```

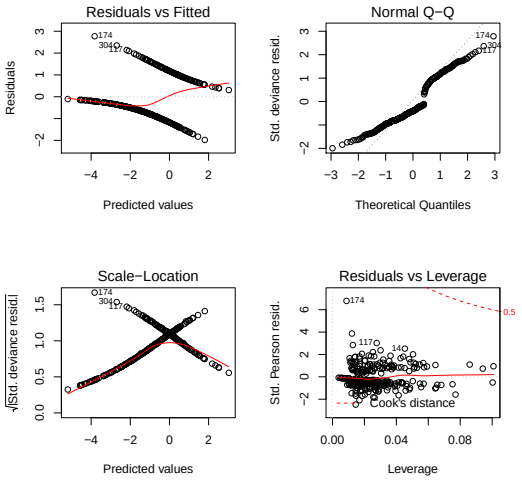

Figure 18: Interactions with famhist.

A backward selection (using AIC) keeps only the interaction between `famhist` and `log(ldl)`. However, if you take a closer look at the model summary, the inclusion of the interaction has markedly dropped the significance of the main effect of `log(ldl)`. This should raise your suspicion that perhaps the introduction of an interaction has created a collinearity problem, and the interaction provides little information about `chd` not already provided by `log(ldl)`. In fact, you can compute the correlation matrix for the coefficient estimates:

```
> ss <- summary(selg)
> c2 <- solve(diag(diag(ss$cov.sc)))^{
```

```
+     1/2
+ } %*% ss$cov.sc %*% solve(diag(diag(ss$cov.sc)))^{
+     1/2
+ }
> print(c2)

            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
[1,]  1.00000000 -0.13307670 -0.13680831 -0.81439919 -0.18275930 -0.52499344
[2,] -0.13307670  1.00000000  0.54400760 -0.05713988  0.01218211  0.02146770
[3,] -0.13680831  0.54400760  1.00000000 -0.04853715 -0.29628916  0.03518698
[4,] -0.81439919 -0.05713988 -0.04853715  1.00000000 -0.15684332  0.22836237
[5,] -0.18275930  0.01218211 -0.29628916 -0.15684332  1.00000000 -0.07489667
[6,] -0.52499344  0.02146770  0.03518698  0.22836237 -0.07489667  1.00000000
[7,] -0.07514361 -0.08462992 -0.11348971 -0.09531484  0.05528966  0.01607170
[8,]  0.15030009 -0.96954641 -0.58603212  0.03801928 -0.02562207 -0.02007436
            [,7]        [,8]
[1,] -0.07514361  0.15030009
[2,] -0.08462992 -0.96954641
[3,] -0.11348971 -0.58603212
[4,] -0.09531484  0.03801928
[5,]  0.05528966 -0.02562207
[6,]  0.01607170 -0.02007436
[7,]  1.00000000  0.10254993
[8,]  0.10254993  1.00000000
```

Notice that the correlation between the coefficient estimate $\hat{\beta}_{ldl}$ and the $\hat{\beta}_{ldl*famhist}$ is over 0.95!. With such extreme collinearity, I drop the interaction term from the model and revert back to the additive model above.

(Try looking for interactions between the numerical variables - you need to bin the values of one of them first in order to plot the log-odds comparisons as above.)