

Chapter 7

Logistic Regression Models

{ch:logistic}

This chapter introduces the modeling framework for categorical data in the simple situation where we have a categorical response variable, often binary, and one or more explanatory variables. A fitted model provides both statistical inference and prediction, accompanied by measures of uncertainty. Data visualization methods for discrete response data must often rely on smoothing techniques, including both direct, non-parametric smoothing and the implicit smoothing that results from a fitted parametric model. Diagnostic plots help us to detect influential observations which may distort our results.

7.1 Introduction

{sec:logist-intro}

All models are wrong, but some are useful

George E. P. Box, (Box and Draper, 1987, p. 424)

Chapters 4–6 have been concerned primarily with simple, exploratory methods for studying the relations among categorical variables and with testing hypotheses about their associations through non-parametric tests and with overall goodness-of-fit statistics.

This chapter begins our study of model-based methods for the analysis of discrete data. These models differ from those we have examined earlier primarily in that they consider *explicitly* an assumed probability distribution for the observations, and make clear distinctions between the systematic component, which is explained by the model, and the random component, which is not. More importantly, the model-based approach allows a compact summary of categorical data in terms of a (hopefully) small number of parameters accompanied by measures of uncertainty (standard errors), and the ability to estimate predicted values over the range of explanatory variables.

This model-fitting approach has several advantages: (a) Inferences for the model parameters include both hypothesis tests and confidence intervals. (b) The former help us to assess which explanatory variables affect the outcome; the size of the estimated parameters and the widths of their confidence intervals help us to assess the strength and importance of these effects. (c) There are a variety of methods for model selection, designed to help determine a favorable trade-off between goodness-of-fit and parsimony. (d) Finally, the predicted values obtained from the model effec-

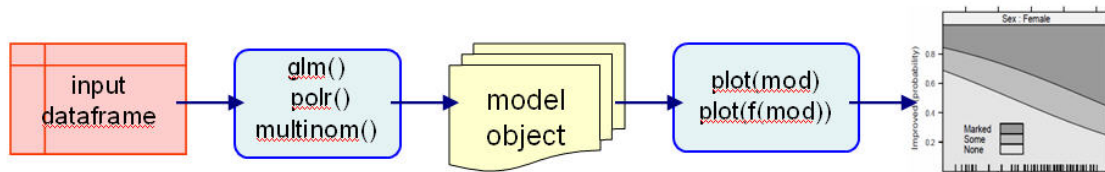


Figure 7.1: Overview of fitting and graphing for model-based methods in R.

{fig:governv}

tively smooth the discrete responses, allow predictions for unobserved values of the explanatory variables, and provide important means to interpret the fitted relationship graphically.

Figure 7.1 provides a visual overview of the steps for fitting and graphing with model-based methods in R. (a) A modeling function such as `glm()` is applied to an input data frame. The result is a *model object* containing all the information from the fitting process. (b) As is standard in R, `print()` and `summary()` methods give, respectively, basic and detailed printed output. (c) Many modeling functions have `plot()` methods that produce different types of summary and diagnostic plots. (d) For visualizing the fitted model, most model methods provide a `predict()` method that can be used to plot the fitted values from the model over the ranges of the predictors. Such plots can be customized by the addition of points (showing the observations), lines, confidence bands, and so forth.

In this chapter we consider models for a *binary response*, such as “success” or “failure”, or the number of “successes” in a fixed number of “trials”, where we might reasonably assume a binomial distribution for the random component. These methods extend readily to a *polytomous response* with more than two outcome categories, such as improvement in therapy, with categories “none,” “some” and “marked.”

These models can be seen as simple extensions of familiar ANOVA and regression models for quantitative data. They are also important special cases of a more general approach, the *generalized linear model* that subsumes a wide variety of families of techniques within a single, unified framework. However, rather than starting at the top with the fully general version, this chapter details the important special cases of models for discrete outcomes, beginning with binary responses.

This chapter proceeds as follows: in Section 7.2 we introduce the simple logistic regression model for a binary response and a single quantitative predictor. This model extends directly to models for grouped, binomial data (Section 7.2.4) and to models with any number of regressors (Section 7.3), which can be quantitative, discrete factors and more general forms. For interpreting and understanding the results of a fitted model, we emphasize plotting predicted probabilities and predicted log odds in various ways, for which effect plots (Section 7.3.3) are particularly useful for complex models. Individual observations sometimes exert great influence on a fitted model. Some measures of influence and diagnostic plots are illustrated in Section 7.5. In Section 7.6, we develop several approaches to modelling a multi-category (polytomous) response.

7.2 The logistic regression model

{sec:logist-model}

The logistic regression model describes the relationship between a discrete outcome variable, the “response”, and a set of explanatory variables. The response variable is often *dichotomous*, although extensions to the model permit multi-category, *polytomous* outcomes, discussed in Sec-

tion 7.6. The explanatory variables may be continuous or (with factor variables) discrete.

For a binary response, Y , and a continuous explanatory variable, X , we may be interested in modeling the probability of a successful outcome, which we denote $\pi(x) \equiv \Pr(Y = 1 | X = x)$. That is, at a given value $X = x$, you can imagine that there is a binomial distribution of the responses, $\text{Bin}(\pi(x), n_x)$.

The simplest naive model, called the *linear probability model*, supposes that this probability, $\pi(x)$ varies linearly with the value of x ,

$$E(Y | x) = \pi(x) = \alpha + \beta x, \quad (7.1) \quad \{\text{eq:logit0}\}$$

where the notation $E(Y | x)$ indicates that the probability $\pi(x)$ represents the population conditional average of the 1s and 0s for all observations with a fixed value of x . For binary observations, this is simply the proportion of 1s.

Figure 7.2 illustrates the basic setup for modeling a binary outcome using the `Arthritis` data, and described more fully in Example 7.1–Example 7.3. The 0/1 observations are shown as (jittered) points. The predicted values under the linear probability model Eqn. (7.1) are shown as the black line. As you can see, this model cannot be right, because it predicts a probability less than 0 for small values of Age, and would also predict probabilities greater than 1 for larger values of Age.

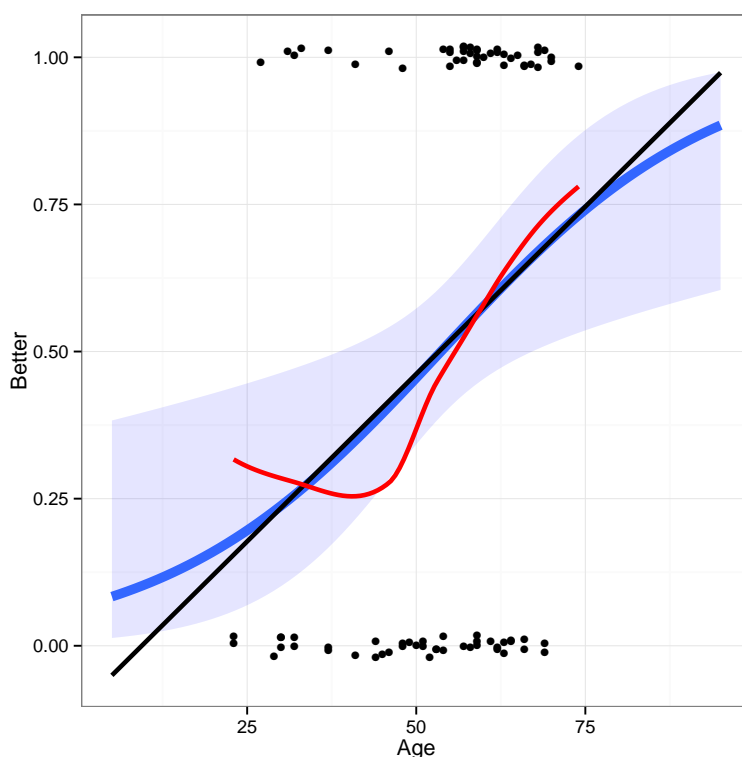


Figure 7.2: Arthritis treatment data, for the relationship of the binary response “Better” to Age. The blue curve and shaded confidence band show a fitted logistic regression to the observations shown as jittered points. The black line shows a simple linear regression and the red curve shows a non-parametric (loess) smoothed curve.

{fig:arthritis-age}

The linear probability model is also wrong because it assumes that the distribution of residu-

als, $Y_i - \hat{\pi}(x_i)$ is normal, with mean 0 and constant variance. However, because Y is dichotomous, the residuals are also dichotomous, and have variance $\pi(x_i)(1 - \pi(x_i))$, which is maximal for $\pi = 0.5$ and decreases as π goes toward 0 or 1.

One way around the difficulty of needing to constrain the predicted values to the interval $[0, 1]$ is to re-specify the model so that a *transformation* of π has a linear relation to x , and that transformation keeps $\hat{\pi}$ between 0 and 1 for all x . This idea, of modeling a transformation of the response that has desired statistical properties is one of the fundamental ones that led to the development of **generalized linear models**, which we treat more fully later in Chapter 9.

A particularly convenient choice of the transformation gives the **linear logistic regression model** (or **linear logit model**¹) which posits a linear relation between the **log odds** (or **logit**) of this probability and x ,

$$\text{logit}[\pi(x)] \equiv \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \alpha + \beta x . \quad (7.2)$$

When $\beta > 0$, $\pi(x)$ and the log odds increase as X increases; when $\beta < 0$ they decrease with X .

This model can also be expressed as a model for the probabilities $\pi(x)$ in terms of the *inverse* of the logit transformation used in Eqn. (7.2),

$$\pi(x) = \text{logit}^{-1}[\pi(x)] = \frac{1}{1 + \exp[-(\alpha + \beta x)]} \quad (7.3)$$

This transformation uses the cumulative distribution function of the logistic distribution, $\Lambda(p) = \frac{1}{1 + \exp(-p)}$, giving rise to the term *logistic regression*.²

From Eqn. (7.2) we see that the odds of a success response can be expressed as

$$\text{odds}(Y = 1) \equiv \frac{\pi(x)}{1 - \pi(x)} = \exp(\alpha + \beta x) = e^\alpha (e^\beta)^x , \quad (7.4)$$

which is a multiplicative model for the odds. So, under the logistic model,

- β is the change in the log odds associated with a unit increase in x . The odds are multiplied by e^β for each unit increase in x .
- α is log odds at $x = 0$; e^α is the odds of a favorable response at this x -value (which may not have a reasonable interpretation if $X = 0$ is far from the range of the data).

It is easy to explore the relationships among probabilities, odds and log odds using R as we show below, using the function `fractions()` in MASS to print the odds corresponding to probability `p` as a fraction.

```
library(MASS)
p <- c(.05, .10, .25, .50, .75, .90, .95)
data.frame(p,
           odds=as.character(fractions(p/(1-p))),
           logit=log(p/(1-p)))
```

¹Some writers use the term *logit model* to refer to those using only categorical predictors; we use the terms logistic regression and logit regression interchangeably.

²Any other cumulative probability transformation serves the purpose of constraining the probabilities to the interval $[0, 1]$. The cumulative normal transformation $\pi(x) = \Phi(\alpha + \beta x)$ gives the **linear probit regression** model. We don't treat probit models here because: (a) The logistic and probit models give results so similar that it is hard to distinguish them in practice; (b) The logistic model is simpler to interpret as a linear model for the log odds or multiplicative model for the odds.

```
##      p odds  logit
## 1 0.05 1/19 -2.9444
## 2 0.10 1/9  -2.1972
## 3 0.25 1/3  -1.0986
## 4 0.50 1    0.0000
## 5 0.75 3    1.0986
## 6 0.90 9    2.1972
## 7 0.95 19   2.9444
```

Thus, a probability of $\pi = 0.25$ represents an odds of 1 to 3, or $1/3$, while a probability of $\pi = 0.75$ represents an odds of 3 to 1, or 3. The logits are symmetric around 0, so $\text{logit}(.25) = -\text{logit}(.75)$.

Another simple way to interpret the parameter β in the logistic regression model is to consider the relationship between the probability $\pi(x)$ and x . From Eqn. (7.3) it can be shown that the fitted curve (the blue line in Figure 7.2) has slope equal to $\beta\pi(1 - \pi)$. This has a maximum value of $\beta/4$ when $\pi = \frac{1}{2}$, so taking $\beta/4$ gives a quick estimate of the maximum effect of x on the probability scale.

In Figure 7.2 and other plots later in this chapter we try to show the binary responses (as jittered points or a rug plot) to help you appreciate how the fitted logistic curve arises from their distribution across the range a predictor. For didactic purposes this can be seen more readily by plotting the conditional distributions of $x|y = \{0, 1\}$ as a histogram, boxplot or density plot. The function `logi.hist.plot()` in the `probio` package is a nice implementation of this idea (de la Cruz Rot, 2005). The call below produces Figure 7.3, and it is easy to see how increasing age produces a greater probability of a Better response.

```
with(Arthritis,
     logi.hist.plot(Age, Better, type="hist", counts=TRUE,
                   ylabel="Probability (Better)", xlab="Age",
                   col.cur="blue", col.hist="lightblue", col.box="lightblue")
)
```

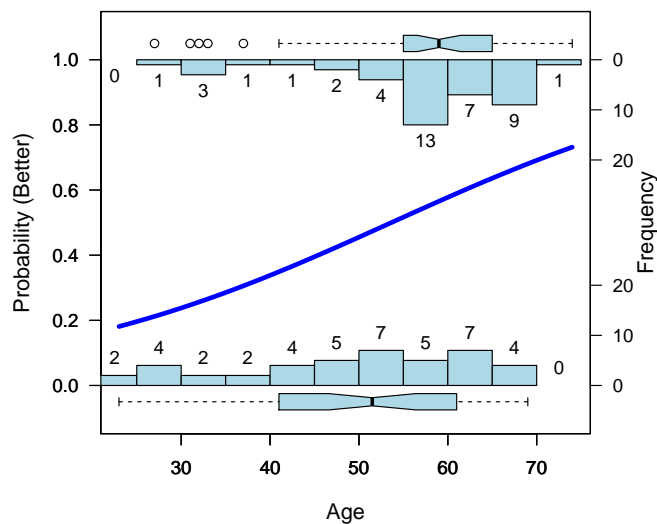


Figure 7.3: Plot of the Arthritis treatment data, showing the conditional distributions of the 0/1 observations of the Better response by histograms and boxplots. fig:arth-logi-hist

7.2.1 Fitting a logistic regression model

{sec:logist

Logistic regression models are the special case of generalized linear models fit in R using `glm()` for a binary response using `family=binomial`. We first illustrate how simple models can be fit and interpreted.

{ex:arthrit6}

EXAMPLE 7.1: Arthritis treatment

In Chapter 4 we examined the data on treatment for rheumatoid arthritis in relation to two categorical predictors, sex of patient and treatment. In addition, the `Arthritis` data gives the age of each patient in this study, and we focus here on the relationship between *Age* and the outcome, *Improved*. This response variable has three categories (none, some, or marked improvement), but for now we consider whether the patient showed any improvement at all, defining the event `Better` to be some or marked improvement.

```
data("Arthritis", package="vcd")
Arthritis$Better <- as.numeric(Arthritis$Improved > "None")
```

The logistic regression model is fit using `glm()` as shown below, specifying `family=binomial` for a binary response.

```
arth.logistic <- glm(Better ~ Age, data=Arthritis, family=binomial)
```

As usual for R modeling functions, the `print()` method for "glm" objects gives brief printed output, while the `summary()` method is more verbose, and includes standard errors and hypothesis tests for the model coefficients. To save some space, it is convenient to use the generic function `coeftest()` from the `lmtest` package. Then, we can use this instead of the more detailed `summary()`:

```
library(lmtest)
coeftest(arth.logistic)

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.6421     1.0732  -2.46   0.014 *
## Age          0.0492     0.0194   2.54   0.011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the output above, the parameter estimates are $\alpha = -2.642$, and $\beta = 0.0492$. So, the estimated odds of a better response are multiplied by $e^\beta = \exp(0.0492) = 1.05$ for each one year increase in age. Equivalently, you can think of this as a 5% increase per year (using $100(e^\beta - 1)$ to convert). Over 10 years, the odds are multiplied by $\exp(10 \times 0.0492) = 1.64$, a 64% increase, a substantial effect in the range for these data. You can do these calculations in R using the `coef()` method for the "glm" object.

```
exp(coef(arth.logistic))

## (Intercept)      Age
## 0.071214     1.050482

exp(10*coef(arth.logistic)[2])

## Age
## 1.6364
```

For comparison with the logistic model, we could fit the linear probability model Eqn. (7.1) using either `lm()` or `glm()` with the default `family=gaussian` argument.

```
arth.lm <- glm(Better ~ Age, data=Arthritis)
coef(arth.lm)

## (Intercept)          Age
## -0.107170      0.011379
```

The coefficient for age can be interpreted to indicate that the probability of a better response increases by 0.011 for each one year increase in age. You can compare this with the $\beta/4$ rule of thumb, that gives $0.0492/4 = 0.0123$. Even though the linear probability model is inappropriate theoretically, you can see in Figure 7.2 (the black line) that it gives similar predicted probabilities to those of the logistic model between age 25–75, where most of the data points are located.

△

7.2.2 Model tests for simple logistic regression

{sec:logist-tests}

There are two main types of hypothesis tests one might want to perform for a logistic regression model. We postpone general discussion of this topic until Section 7.3, but introduce the main ideas here using the analysis of the `Arthritis` data.

- The most basic test answers the question “How much better is the fitted model, $\text{logit}(\pi) = \alpha + \beta x$ than the null model $\text{logit}(\pi) = \alpha$ that includes only the regression intercept?” One answer to this question is given by the (Wald) test of the coefficient for age testing the hypothesis $H_0 : \beta = 0$ that appeared in the output from `summary(arth.logistic)` shown above. The more direct test compares the deviance of the fitted model to the deviance of the null model, and can be obtained using the `anova()` function:

```
anova(arth.logistic, test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Better
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL          83      116
## Age           1       7.29      82      109  0.007 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- A second question is “How bad is this model, compared to a model (the *saturated model*) that fits the data perfectly?” This is a test of the size of the residual deviance, that is given by the function `Summarise()` in `vcdExtra`.

```
Summarise(arth.logistic)
```

```
## Likelihood summary table:
##           AIC BIC LR Chisq Df Pr(>Chisq)
## arth.logistic 113 118      109 82      0.024 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The summary of these tests is that linear logistic model Eqn. (7.2) fits significantly better than the null model, but that model also shows significant lack of fit.

7.2.3 Plotting a binary response

{sec:logist-plotting}

It is often difficult to understand how a binary response can give rise to a smooth, continuous relation between the predicted response, usually the probability of an event, and a continuous explanatory variable. Beyond this, plots of the data together with fitted models help you to interpret what these models imply.

We illustrate two approaches below using the `Arthritis` data shown in Figure 7.2, first using R base graphics, and then with the `ggplot2` package that makes such graphs somewhat easier to do.

That plot, which was designed for didactic purposes, has the following features:

- It shows the *data*, that is, the 0/1 observations of the `Better` response in relation to `age`. To do this effectively and avoid over-plotting, the binary responses are jittered.
- It plots the predicted (fitted) logistic regression relationship on the scale of probability, together with a 95% confidence band.
- It also plots the predicted probabilities from the linear probability model.
- A smoothed, non-parametric regression curve for the binary observations is also added to the plot to give some indication of possible non-linearity in the relationship of `Better` to `age`.

{ex:arthrit7}

EXAMPLE 7.2: Arthritis treatment: Plotting logistic regression with base graphics

Here we explain how plots similar to Figure 7.2 can be constructed using R base graphics. We describe the steps needed to calculate predicted values and confidence bands and how to add these to a basic plot. These ideas are the basis for the higher-level and more convenient plotting methods illustrated later in this chapter. The steps detailed below give the plot shown in Figure 7.4.

First, we set up the basic plot of the jittered values of `Better` vs. `Age`, setting `xlim` to a larger range than that in the data, only to emphasize where the logistic and linear probability models diverge.

```
plot(jitter(Better, .1) ~ Age, data=Arthritis,
     xlim = c(15, 85), pch=16,
     ylab="Probability (Better)")
```

The fitted logistic curve can be obtained using the `predict()` method for the "glm" object `arth.logistic`. For this example, we wanted to get fitted values for the range of `Age` from 15–85, which is specified in the `newdata` argument.³ The argument `type="response"`

³Omitting the `newdata` argument would give predicted values using the linear predictors in the data used for the fitted model. Some care needs to be taken if the predictor(s) contain missing values.

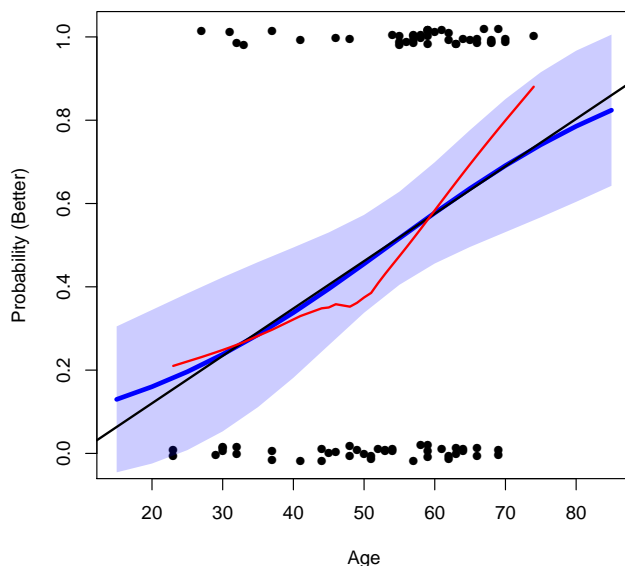


Figure 7.4: A version of plot of the Arthritis treatment data (Figure 7.2) produced with R base graphics, showing logistic, linear regression and lowess fits. fig:arthritis-age2

gives fitted values of the probabilities. (The default, `type="link"` would give predicted logits.) Standard errors of the fitted values are not calculated by default, so we set `se.fit=TRUE`.

```
xvalues <- seq(15, 85, 5)
pred.logistic <- predict(arth.logistic,
                        newdata=data.frame(Age=xvalues),
                        type="response", se.fit=TRUE)
```

When `se.fit=TRUE`, the `predict()` function returns its result in a list, with components `fit` for the fitted values and `se.fit` for the standard errors. From these, we can calculate 95% pointwise prediction intervals using the standard normal approximation.

```
upper <- pred.logistic$fit + 1.96 * pred.logistic$se.fit
lower <- pred.logistic$fit - 1.96 * pred.logistic$se.fit
```

We can then plot the confidence band using `polygon()` and the fitted logistic curve using `lines`. A graphics trick is used here to use a transparent color for the confidence band using `rgb(r, g, b, alpha)`, where `alpha` is the transparency value.

```
polygon(c(xvalues, rev(xvalues)),
       c(upper, rev(lower)),
       col=rgb(0, 0, 1, .2), border=NA)
lines(xvalues, pred.logistic$fit, lwd=4, col="blue")
```

This method, using `predict()` for calculations and `polygon()` and `lines()` for plotting can be used to display the predicted relationships and confidence bands under other models. Here, we simply used `abline()` to plot the fitted line for the linear probability model `arth.lm` and `lowess()` to calculate a smoothed, non-parametric curve.

```
abline(arth.lm, lwd=2)
lines(lowess(Arthritis$Age, Arthritis$Better, f=.9), col="red", lwd=2)
```

△

{ex:arthrit8}

EXAMPLE 7.3: Arthritis treatment: Plotting logistic regression with ggplot2

Model-based plots such as Figure 7.2 are relatively more straight-forward to produce using ggplot2. The basic steps here are to:

- set up the plot frame with ggplot() using Age and Better as (x, y) coordinates;
- use geom_point() to plot the observations, whose positions are jittered with position_jitter();
- use stat_smooth() with method = "glm" and family = binomial to plot the predicted probability curve and confidence band. By default, stat_smooth() calculates and plots 95% confidence bands on the response (probability) scale.

```
library(ggplot2)
# basic logistic regression plot
gg <- ggplot(Arthritis, aes(x=Age, y=Better)) +
  xlim(5, 95) + theme_bw() +
  geom_point(position = position_jitter(height = 0.02, width = 0)) +
  stat_smooth(method = "glm", family = binomial, alpha = 0.1, fill="blue",
             size=2.5, fullrange=TRUE)
```

Finally, we can add other smoothers to the plot, literally by using + to add these to the "ggplot" object.

```
# add linear model and loess smoothers
gg <- gg + stat_smooth(method = "lm", se=FALSE,
                     size=1.2, color="black", fullrange=TRUE)
gg <- gg + stat_smooth(method = "loess", se=FALSE,
                     span=0.95, colour="red", size=1.2)
gg # show the plot
```

△

7.2.4 Grouped binomial data

A related case occurs with grouped data, where rather than binary observations, $y_i \in \{0, 1\}$ in case form, the data is given in what is called *events/trials form* that records the number of successes, y_i that occurred in n_i trials associated with each setting of the explanatory variable(s) x_i .⁴ Case form, with binary observations is the special case where $n_i = 1$.

Data in events/trials form often arises from contingency table data with a binary response. For example in the UCBA admissions data, the response variable *Admit* with levels "Admitted", "Rejected" could be treated in this way using the number of applicants as the number of trials.

As before, we can consider y_i/n_i to estimate the probability of success, π_i and the distribution of Y to be binomial, $\text{Bin}(\pi_i, n_i)$ at each x_i .

In practical applications, there are two main differences between the cases of ungrouped, case form data and grouped, event/trials form.

⁴Alternatively, the data may record the number of successes, y_i , and number of failures, $n_i - y_i$.

{sec:logist-grouped}

- In fitting models using `glm()`, the model formula, `response ~ terms`, can be given using a response consisting of a two-column matrix, whose columns contain the numbers of successes y_i and failures $n_i - y_i$. Alternatively, the response can be given as the proportion of successes, y_i/n_i , but then it is necessary to specify the number of trials as a weight.
- In plotting the fitted model on the scale of probability, you usually have to explicitly plot the fraction of successes, y_i/n_i .

{ex:nasa-temp}

EXAMPLE 7.4: Space shuttle disaster

In Example 1.2 and Example 1.10 we described the background behind the post-mortem examination of the evidence relating to the disastrous launch of the space shuttle *Challenger* on January 28, 1986. Here we consider a simple, but proper analysis of the data available at the time of launch. We also use this example to illustrate some details of the fitting and plotting of grouped binomial data. As well, we describe some of the possibilities for dealing with missing data.

The data set `SpaceShuttle` in `vcd` contains data on the failures of the O-rings in 24 NASA launches preceding the launch of *Challenger*, as given by Dalal *et al.* (1989) and Tufte (1997) also analysed by Lavine (1991).

Each launch used two booster rockets with a total of six O-rings, and the data set records as `nFailures` the number of these that were considered damaged after the rockets were recovered at sea. In one launch (flight # 4), the rocket was lost at sea, so the relevant response variables are missing.

In this example, we focus on the variable `nFailures` as a binomial with $n_i = 6$ trials. The missing data for flight 4 can be handled in several ways in the call to `glm()`

```
data("SpaceShuttle", package="vcd")
shuttle.mod <- glm(cbind(nFailures, 6 - nFailures) ~ Temperature,
                  data = SpaceShuttle, na.action = na.exclude,
                  family = binomial)
```

Alternatively, we can add an explicit `trials` variable, represent the response as the proportion `nFailures/trials`, and use `weight = trials` to indicate the total number of observations.

```
SpaceShuttle$trials <- 6
shuttle.modw <- glm(nFailures/trials ~ Temperature, weight = trials,
                  data = SpaceShuttle, na.action = na.exclude,
                  family = binomial)
```

These two approaches give identical results for all practical purposes:

```
all.equal(coef(shuttle.mod), coef(shuttle.modw))

## [1] TRUE
```

As before, we can test whether temperature significantly improves prediction of failure probability using `anova()`:

```
# testing, vs. null model
anova(shuttle.mod, test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(nFailures, 6 - nFailures)
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                22      24.2
## Temperature  1         6.14      21      18.1  0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The code below gives a `ggplot2` version in Figure 7.5 of the plot we showed earlier in Example 1.2 (Figure 1.2). The relevant details here are:

- We specify $y = \text{nFailures} / \text{trials}$ to calculate the failure probabilities.
- Points are jittered in the call to `geom_point()` to prevent overplotting.
- In the call to `geom_smooth()`, we need to use `weight = trials`, just as in the call to `glm()` above.
- `fullrange = TRUE` makes the fitted regression curve and confidence band extend across the entire plot

```
library(ggplot2)
ggplot(SpaceShuttle, aes(x = Temperature, y = nFailures / trials)) +
  xlim(30, 81) + theme_bw() +
  xlab("Temperature (F)") +
  ylab("O-Ring Failure Probability") +
  geom_point(position=position_jitter(width=0, height=0.01),
            aes(size = 2)) +
  theme(legend.position="none") +
  geom_smooth(method = "glm", family = binomial, fill="blue",
            aes(weight = trials), fullrange = TRUE, alpha=0.2, size=2)
```

△

7.3 Multiple logistic regression models

{sec:logist-mult}

As is the case in classical regression, generalizing the simple logistic regression to an arbitrary number of explanatory variables is quite straightforward. We let $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ denote the vector of p explanatory variables for case or cluster i . Then the general logistic regression model can be expressed as

$$\begin{aligned} \text{logit}(\pi_i) &\equiv \log \frac{\pi_i}{1 - \pi_i} = \alpha + \mathbf{x}_i^\top \boldsymbol{\beta} \\ &= \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} . \end{aligned} \quad (7.5)$$

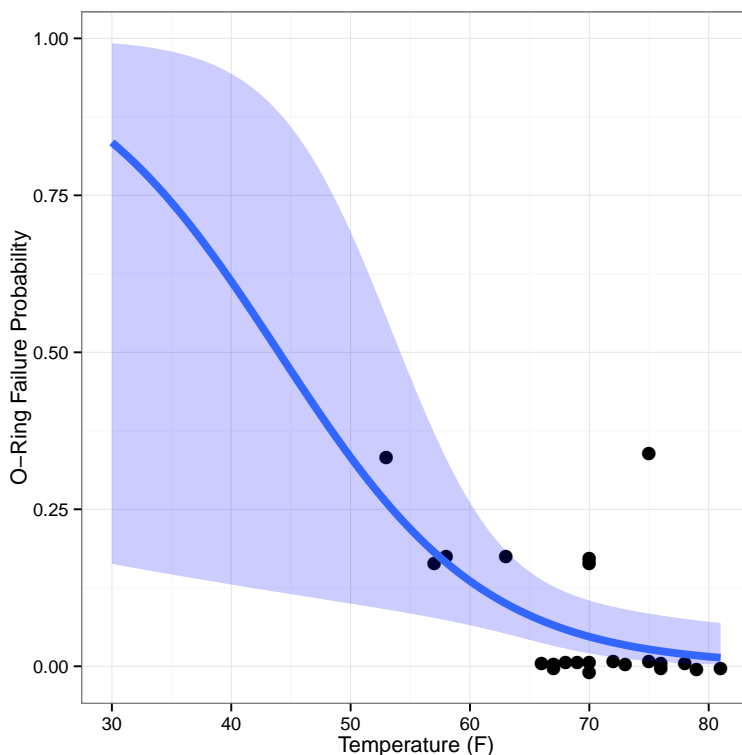


Figure 7.5: Space shuttle data, with fitted logistic regression model

Equivalently, we can represent this model in terms of probabilities as the logistic transformation of the *linear predictor*, $\eta_i = \alpha + \mathbf{x}_i^T \boldsymbol{\beta}$,

$$\begin{aligned} \pi_i = \Lambda(\eta_i) &= \Lambda(\alpha + \mathbf{x}_i^T \boldsymbol{\beta}) \\ &= \frac{1}{1 + \exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}. \end{aligned} \quad (7.6)$$

The x s can include any of the following sorts of regressors, as in the general linear model:

- **quantitative** variables (e.g., age, income)
- **polynomial** powers of quantitative variables (e.g., age, age², age³)
- **transformations** of quantitative variables (e.g., log salary)
- factors, represented as **dummy** variables for qualitative predictors (e.g., P_1, P_2, P_3 for four political party affiliations)
- **interaction** terms (e.g., sex \times age, or age \times income)

{ex:arthrit-mult}

EXAMPLE 7.5: Arthritis treatment

We continue with the analysis of the *Arthritis* data, fitting a model containing the main effects of *Age*, *Sex* and *Treatment*, with *Better* as the response. This model has the form

$$\text{logit}(\pi_i) = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3}$$

where x_1 is *Age* and x_2 and x_3 are the factors representing *Sex* and *Treatment*, respectively. Using the default (0/1) dummy coding that R uses (“treatment” contrasts against the lowest factor

level),⁵ they are defined as:

$$x_2 = \begin{cases} 0 & \text{if Female} \\ 1 & \text{if Male} \end{cases} \quad x_3 = \begin{cases} 0 & \text{if Placebo} \\ 1 & \text{if Treatment} \end{cases}$$

In this model,

- α doesn't have a sensible interpretation here, but formally it would be the log odds of improvement for a person at age $x_1 = 0$ in the baseline or reference group with $x_2 = 0$ and $x_3 = 0$ —females receiving the placebo. To make the intercept interpretable, we will fit the model centering age near the mean, by using $x_1 - 50$ as the first regressor.
- β_1 is the increment in log odds of improvement for each one-year increase in age.
- β_2 is the increment in log odds for male as compared to female. Therefore, e^{β_2} gives the odds of improvement for males relative to females.
- β_3 is the increment in log odds for being in the treated group. e^{β_3} gives the odds of improvement for the active treatment group relative to placebo.

We fit the model as follows. In `glm()` model formulas, “-” has a special meaning, so we use the identity function, `I(Age-50)` to center age.

```
arth.logistic2 <- glm(Better ~ I(Age-50) + Sex + Treatment,
                     data=Arthritis,
                     family=binomial)
```

The parameters defined here are *incremental effects*. The intercept corresponds to a baseline group (50 year-old females given the placebo); the other parameters are incremental effects for the other groups compared to the baseline group. Thus, when α , β_1 , β_2 and β_3 have been estimated, the fitted logits and predicted odds at `Age==50` are:

Sex	Treatment	Logit	Odds Improved
Female	Placebo	α	e^α
Female	Treated	$\alpha + \beta_2$	$e^{\alpha+\beta_2}$
Male	Placebo	$\alpha + \beta_1$	$e^{\alpha+\beta_1}$
Male	Treated	$\alpha + \beta_1 + \beta_2$	$e^{\alpha+\beta_1+\beta_2}$

We first focus on the interpretation of the coefficients estimated for this model shown below.

```
coefTest(arth.logistic2)

##
## z test of coefficients:
##
##              Estimate Std. Error z value Pr(>|z|)
```

⁵For factor variables with the default treatment contrasts, you can change the reference level using `relevel()`. In this example, you could make male the baseline category using `Arthritis$Sex <- relevel(Arthritis$Sex, ref = "Male")`.

```
## (Intercept)      -0.5781      0.3674     -1.57      0.116
## I(Age - 50)      0.0487      0.0207      2.36      0.018 *
## SexMale         -1.4878      0.5948     -2.50      0.012 *
## TreatmentTreated 1.7598      0.5365      3.28      0.001 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To interpret these in terms of odds ratios and also find confidence intervals, just use `exp()` and `confint()`.

```
exp(cbind(OddsRatio=coef(arth.logistic2),
          confint(arth.logistic2)))

##           OddsRatio  2.5 %  97.5 %
## (Intercept)      0.5609 0.26475 1.1323
## I(Age - 50)      1.0500 1.01000 1.0963
## SexMale          0.2259 0.06524 0.6891
## TreatmentTreated 5.8113 2.11870 17.7266
```

Here,

- $\alpha = -0.578$: At age 50, females given the placebo have an odds of improvement of $\exp(-0.578) = 0.56$.
- $\beta_1 = 0.0487$: Each year of age multiplies the odds of improvement by $\exp(0.0487) = 1.05$, or a 5% increase.
- $\beta_2 = -1.49$: Males are only $\exp(-1.49) = 0.26$ times as likely to show improvement relative to females. Equivalently, you could say that females are $\exp(1.49) = 4.437$ times more likely than males to improve.
- $\beta_3 = 1.76$: People given the active treatment are $\exp(1.76) = 5.8$ times more likely to show improvement.

As you can see, the interpretation of coefficients in multiple logistic models is straightforward, though a bit cumbersome. This becomes more difficult in larger models, particularly when there are interactions. In these cases, you can understand (and explain) a fitted model more easily through plots of predicted values, either on the scale of response probability or on the logit scale of the linear predictor. We describe these methods in Section 7.3.1–Section 7.3.3 below.

△

7.3.1 Conditional plots

{sec:logist-condplots}

The simplest kind of plots display the data together with a representation of the fitted relationship (predicted values, confidence bands) separately for subsets of the data defined by one or more of the predictors. Such plots can show the predicted values for the response variable on the ordinate against one chosen predictor on the abscissa, and can use multiple curves and multiple panels to represent other predictors.

However, these plots are *conditional plots*, meaning that the data shown in each panel and used in each fitted curve are limited to the subset of the observations defined by the curve and panel variables. As well, predictors that are not shown in a given plot are effectively ignored (or marginalized), as was the case in Figure 7.2 that showed only the effect of age in the Arthritis data.

{ex:arth-cond}

EXAMPLE 7.6: Arthritis treatment: conditional plots

For the `Arthritis` data, a basic conditional plot of `Better` vs. `Age`, showing the observations as jittered points (with `geom_point()`) and the fitted logistic curves (with `stat_smooth()` using `method="glm"`) can be produced with `ggplot2` as shown below, giving Figure 7.6.

```
library(ggplot2)
gg <- ggplot(Arthritis, aes(Age, Better, color=Treatment)) +
  xlim(5, 95) + theme_bw() +
  geom_point(position = position_jitter(height = 0.02, width = 0)) +
  stat_smooth(method = "glm", family = binomial, alpha = 0.2,
             aes(fill=Treatment), size=2.5, fullrange=TRUE)
gg # show the plot
```

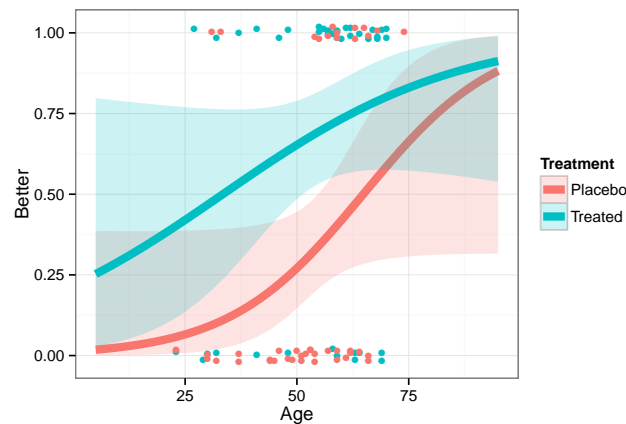


Figure 7.6: Conditional plot of Arthritis data showing separate points and fitted curves stratified by Treatment. A separate fitted curve is shown for the two treatment conditions, ignoring Sex. Fig:arth-cond1

In this call to `ggplot()`, specifying `color=Treatment` gives different point and line colors, but also automatically stratifies the fitted curves using the levels of this variable.

With such a plot, it is easy to add further stratifying variables in the data using `facets` to produce separate panels (functions `facet_wrap()` or `facet_grid()`, with different options to control the details). The following line further stratifies by `Sex`, producing Figure 7.7.

```
gg + facet_wrap(~ Sex)
```

However, you can see from this plot how this method breaks down when the sample size is small in some of the groups defined by the stratifying factors. The panel for males shows a paradoxical negative relation with age for the treated group and a step function for the placebo group. The explanation for this is shown in the two-way frequency table of the sex and treatment combinations:

```
addmargins(xtabs(~Sex + Treatment, data=Arthritis), 2)
```

```
##           Treatment
## Sex      Placebo Treated Sum
## Female      32      27  59
## Male       11      14  25
```

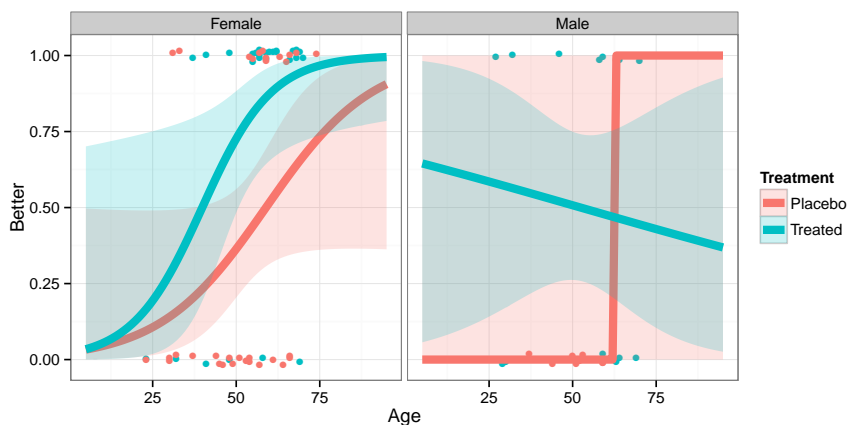



Figure 7.7: Conditional plot of Arthritis data, stratified by Treatment and Sex. The unusual patterns in the panel for Males signals a problem with this data. fig:arth-condz

Less than 1/3 of the sample were males, and of these only 11 were in the placebo group. `glm()` cannot estimate the fitted relationship against *Age* here—the slope coefficient is infinite, and the fitted probabilities are all either 0 or 1.⁶

△

7.3.2 Full-model plots

{sec:logist-fullplots}

For a model with two or more explanatory variables, *full-model plots* display the fitted response surface for all predictors together, rather than stratified by conditioning variables. Such plots show the predicted values for the response variable on the ordinate against one chosen predictor on the abscissa, and can use multiple curves and multiple panels to represent other predictors.

A simple R trick⁷ makes this method far easier and more general than the naive plotting method used in Example 7.2. The trick is simply to combine the columns in the original data frame with the result of the `predict()` method for the fitted model and plot the calculated `fit` value, together with confidence bands (if you use `se.fit=TRUE`).

{ex:arth-full}

EXAMPLE 7.7: Arthritis treatment: full-model plots

```
arth.fit2 <- cbind(Arthritis,
                  predict(arth.logistic2, se.fit = TRUE))
```

The fitted values here are on the logit scale, which means that it takes one more trick to show the data points on the plot. We simply define a new variable, `obs` with convenient logit values corresponding the `Better` values of 0 and 1.

```
arth.fit2$obs <- c(-4, 4)[1+arth.fit2$Better]
```

We can then plot the fitted logit against *Age* using `x=Age`, `y=fit` from the data frame containing the fitted values. The call to `ggplot()` below produces Figure 7.8. Here, we

⁶This is called *complete separation*, and occurs whenever the responses have no overlap on the predictor variable(s) used in fitting the logistic regression model.

⁷Thanks to Dennis Murphy for suggesting this method.

used `color=Treatment` to produce separate points, lines and confidence bands colored by *Treatment*. Confidence bands in the plot are constructed using `geom_ribbon()`.

```
ggplot(arth.fit2, aes(x=Age, y=fit, color=Treatment)) +
  geom_line(size = 2) + theme_bw() +
  geom_ribbon(aes(ymin = fit - 1.96 * se.fit,
                ymax = fit + 1.96 * se.fit,
                fill = Treatment), alpha = 0.2,
            color = "transparent") +
  labs(x = "Age", y = "Log odds (Better)") +
  geom_point(aes(y=obs),
            position=position_jitter(height=0.25, width=0)) +
  facet_wrap(~ Sex)
```

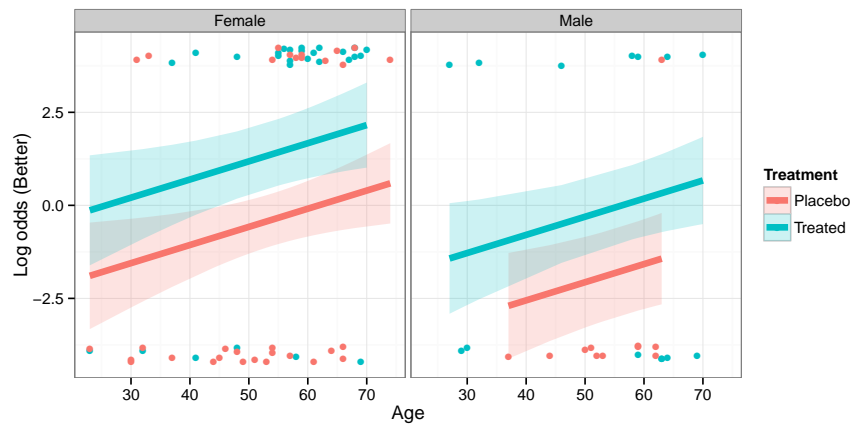


Figure 7.8: Full-model plot of Arthritis data, showing fitted logits by Treatment and Sex.^{fig:arth-full1}

This plot method has several nice features:

- Plotting on the logit scale shows the additive, linear effects of all predictors.
- It provides a visual representation of the information contained in the table of coefficients. Note, however, that the choice to display *Treatment* within each panel makes it easier to judge the size of this effect, compared to the effect of *Sex* which must be judged across the panels.
- It shows the data as points, and the fitted lines and confidence bands are restricted to the range of the data in each. You can easily see the reason for the unusual pattern in the conditional plot for Males shown in Figure 7.7.
- It generalizes directly to any fitted model, because the same plotting code can be used once the model predicted values have been calculated.
- Additional predictors, either factors or quantitative variables can easily be accommodated by including them in the `facet_wrap()` call. For example, if the patients were also categorized by education and this had been included in the model, `facet_wrap(~ Sex + Education)` would produce separate panels for the combinations of these two variables.

While plots on the logit scale have a simpler form, many people find it easier to think about such relationships in terms of probabilities, as we have done in earlier plots in this chapter. You can do the same for full-model plots with a simple extension of this method. All you need to do is to transform the `fit` and end points of the confidence band back to the scale of probabilities. The function `plotlogis()` does this for the logistic distribution.

```
arth.fit2 <- within(arth.fit2, {
  prob <- plogis(fit)
  lower <- plogis(fit - 1.96 * se.fit)
  upper <- plogis(fit + 1.96 * se.fit)
})
```

The plot step is then similar to what we used above (but with `prob`, `lower` and `upper`), producing Figure 7.9.

```
ggplot(arth.fit2, aes(x=Age, y=prob, color=Treatment)) +
  geom_line(size = 2) + theme_bw() +
  geom_ribbon(aes(ymin = lower,
                ymax = upper,
                fill = Treatment), alpha = 0.2,
            color = "transparent") +
  labs(x = "Age", y = "Probability (Better)") +
  geom_point(aes(y=Better),
            position=position_jitter(height=0.02, width=0)) +
  facet_wrap(~ Sex)
```

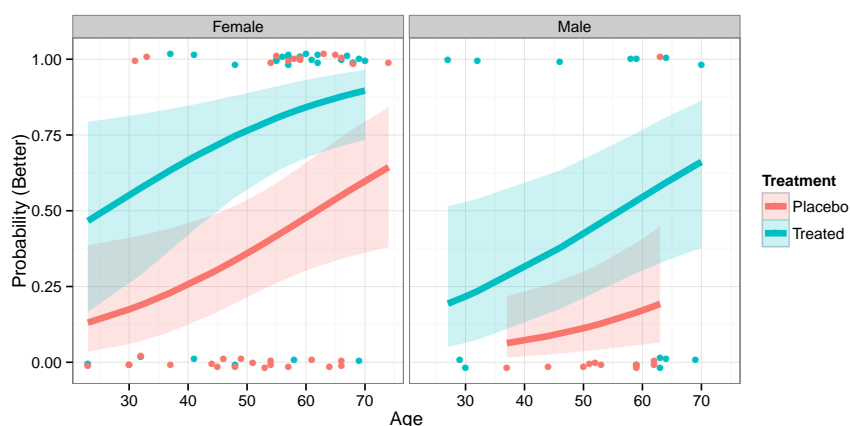


Figure 7.9: Full-model plot of Arthritis data, showing fitted probabilities by Treatment and Sex.^{fig:arth-full12}

△

7.3.3 Effect plots

{sec:logist-effplots}

For more than two variables, full-model plots of the fitted response surface can be cumbersome, particularly when the model contains interactions or when the main substantive interest is focused on a given main effect or interaction, controlling for all other explanatory variables. The method of *effect displays* (tables and graphs), developed by John Fox (1987, 2003) and implemented in the `effects` package, is a useful solution to these problems.

The idea of effect plots is quite simple but very general:⁸ consider a particular subset of predictors (*focal predictors*) we wish to visualize in a given linear model or generalized linear model. The essence is to calculate fitted values (and standard errors) for the model terms involving these

⁸Less general expressions of these ideas include the use of *adjusted means* in analysis of covariance, and *least squares means* or *population marginal means* (Searle *et al.*, 1980) in analysis of variance; for example, see the `lsmeans` package for classical linear models.

variables and all low-order relatives (e.g., main effects that are marginal to an interaction), as these variables are allowed to vary over their range. All other variables are “controlled” by being fixed at typical values. For example a quantitative covariate could be fixed at its mean or median; a factor could be fixed at equal proportions of its levels or its proportions in the data. The result, when plotted, shows all effects of the focal predictors and their low-order relatives, but with all other variables not included controlled or adjusted for.

More formally, assume we have fit a model with a linear predictor $\eta_i = \alpha + \mathbf{x}_i^T \boldsymbol{\beta}$ (on the logit scale, for logistic regression). Letting $\beta_0 = \alpha$ and $\mathbf{x}_0 = \mathbf{1}$, we can rewrite this in matrix form as $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ where \mathbf{X} is the model matrix constructed by the modeling function, such as `glm()`. Fitting the model gives the estimated coefficients \mathbf{b} and its estimated covariance matrix $\widehat{\mathcal{V}}(\mathbf{b})$.

The `Effect()` function constructs an analogous *score model matrix*, \mathbf{X}^* , where the focal variables have been varied over their range, and all other variables represented as constant, typical values. Using this as input (the `newdata` argument) to the `predict()` function then gives the fitted values $\boldsymbol{\eta}^* = \mathbf{X}^*\mathbf{b}$. Standard errors used for confidence intervals are calculated by `predict()` (when `se.fit=TRUE`) as the square roots of $\text{diag}(\mathbf{X}^*\widehat{\mathcal{V}}(\mathbf{b})\mathbf{X}^{*\top})$. Note that these ideas work not only for `glm()` models, but potentially for any modeling function that has a `predict()` and `vcov()` method.⁹

These results are calculated on the scale of the linear predictor $\boldsymbol{\eta}$ (logits, for logistic regression) when the `type` argument to `predict()` is `type="link"` or on the response scale (probabilities, here) when `type="response"`. The latter makes use of the inverse transformation, Eqn. (7.6).

There are two main calculation functions in the `effects` package:

- `Effect()` takes a character vector of the names of a subset of focal predictors and constructs the score matrix \mathbf{X}^* by varying these over their ranges, while holding all other predictors constant at “typical” values. There are many options that control these calculations. For example, `xlevels` can be used to specify the values of the focal predictors; `typical` or `given.values` respectively can be used to specify either a function (mean, median) or a list of specific typical values used for the variables that are controlled. The result is an object of class “eff”, for which there are `print()`, `summary()` and (most importantly) `plot()` methods. See `help(Effect)` for a complete description.
- `allEffects()` takes a model object, and calculates the effects for each high-order term in the model (including their low-order) relatives. Similar optional arguments control the details of the computation. The result is an object of class “efflist”.

In addition, the plotting methods for “eff” and “efflist” objects offer numerous options to control the plot details, only a few of which are used in the examples below. For logistic regression models, they also solve the problem of the trade-off between plots on the logit scale, that have a simple representation in terms of additive effects, and plots on the probability scale that are usually simpler to understand. By default, the fitted model effects are plotted on the logit scale, but the response y axis is labeled with the corresponding probability values.

{ex:arthritis-eff}

⁹For example, the `effects` package presently provides methods for models fit by `lm()` (including multivariate linear response models), `glm()`, `gls()`, multinomial (`multinom()` in the `nnet` package) and proportional odds models (`polr()` in `MASS`), polytomous latent class models (`poLCA` package), as well as a variety of multi-level and mixed-effects linear models fit with `lmer()` from the `lme4` package, or with `lme()` from the `nlme` package.

EXAMPLE 7.8: Arthritis treatment

Here we illustrate the use of the `effects` package with the simple main effects model which was fit in Example 7.5. `allEffects()` is used to calculate the predicted probabilities of the *Better* response for *Age* and the two factors, *Sex* and *Treatment*.

```
library(effects)
arth.eff2 <- allEffects(arth.logistic2)
names(arth.eff2)

## [1] "I(Age-50)" "Sex" "Treatment"
```

The result, `arth.eff2` is a list containing the fitted values (response probabilities, by default) for each of the model terms. No `xlevels` argument was specified, so by default the function calculated the effects for *Age* at a reasonable selection of equally-spaced values:

```
arth.eff2[[1]]

##
## Age effect
## Age
##      30      40      50      60      70
## 0.24289 0.34311 0.45959 0.58066 0.69274
```

The default plot method for the "efflist" object produces one plot for each high-order term, which are just the main effect in this model. The call below produces Figure 7.10.

```
plot(arth.eff2, rows=1, cols=3)
```

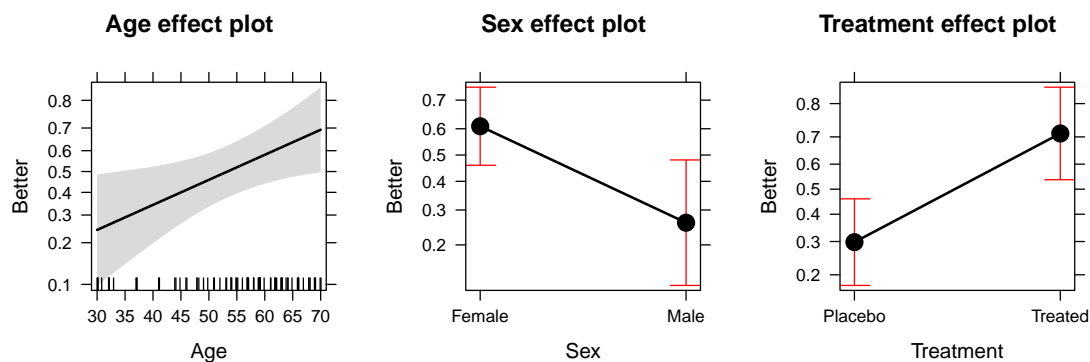


Figure 7.10: Plot of all effects in the main effects model for the Arthritis data fig:arth-effplot1

You can also produce full-model plots quite easily by using all predictors in the model in a call to `Effect()`.

```
arth.full <- Effect(c("Age", "Treatment", "Sex"), arth.logistic2)
```

Then plotting the result, with some options, gives the plot shown in Figure 7.11.

```
plot(arth.full, multiline=TRUE, ci.style="bands",
     colors = c("red", "blue"), lwd=3,
     key.args=list(x=.52, y=.92), grid=TRUE)
```

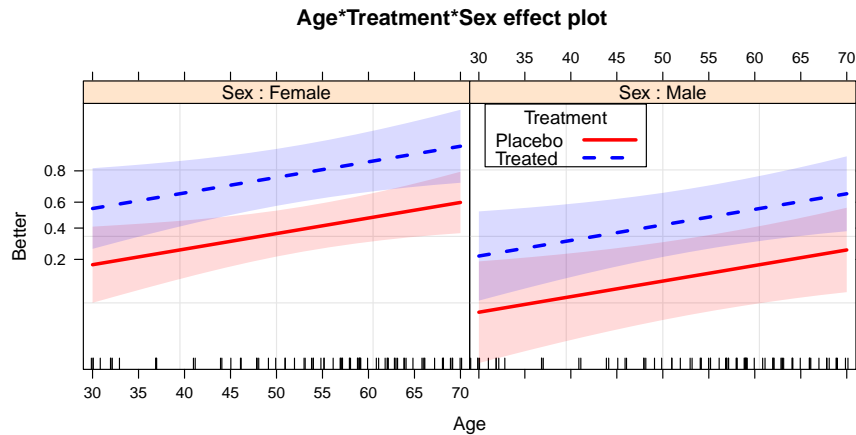


Figure 7.11: Full-model plot of the effects of all predictors in the main effects model for the Arthritis data, plotted on the logit scale. fig:arth-erplot2

Alternatively, we can plot these results directly on the scale of probabilities, as shown in Figure 7.12.

```
plot(arth.full, multiline=TRUE, ci.style="bands", rescale.axis=FALSE,
     colors = c("red", "blue"), lwd=3,
     key.args=list(x=.52, y=.92), grid=TRUE)
```

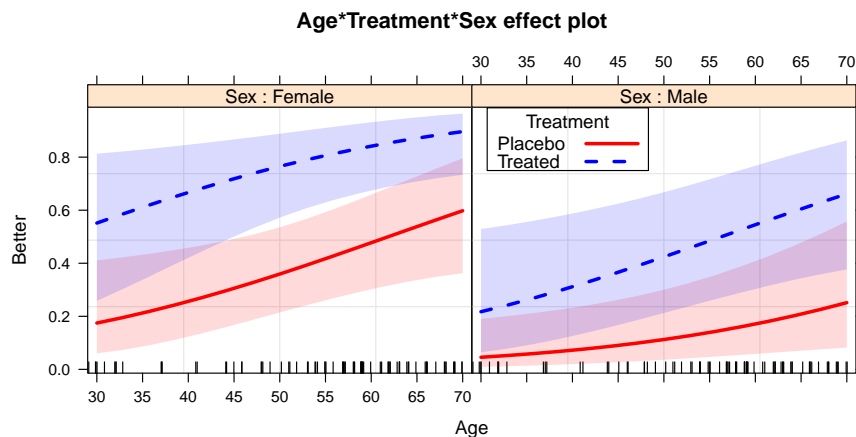


Figure 7.12: Full-model plot of the effects of all predictors in the main effects model for the Arthritis data, plotted on the probability scale. fig:arth-erplots

△

7.4 Case studies

The examples below take up some issues of data analysis, model building and visualization in the context of multiple logistic regression models. We focus on the combination of exploratory plots to see the data, modeling steps and graphs to interpret a given model.

{ex:donner1}

EXAMPLE 7.9: Donner Party

In Chapter 1, Example 1.3, we described the background behind the sad story of the Donner Party, perhaps the most famous tragedy in the history of the westward settlement in the United States. In brief, the party was stranded on the eastern side of the Sierra Nevada mountains by heavy snow in late October, 1846, and by the time the last survivor was rescued in April, 1847, nearly half of the members had died from famine and exposure to extreme cold. Figure 1.3 showed that survival decreased strongly with age.

Here we consider a more detailed analysis of these data, which are contained in the data set `Donner` in `vcdExtra`. This data set lists 90 people in the Donner Party by name, together with age, sex, survived (0/1) and the date of death for those who died.¹⁰

```
data("Donner", package="vcdExtra") # load the data
library(car) # for some() and Anova()
some(Donner, 8)

##          family age  sex survived  death
## Breen, Peter   Breen  3  Male      1    <NA>
## Donner, Jacob  Donner 65  Male      0 1846-12-21
## Foster, Jeremiah MurFosPik 1  Male      0 1847-03-13
## Graves, Nancy  Graves  9 Female      1    <NA>
## McCutchen, Harriet McCutchen 1 Female      0 1847-02-02
## Reed, James    Reed  46  Male      1    <NA>
## Reinhardt, Joseph Other 30  Male      0 1846-12-21
## Wolfinger, Doris FosdWolf 20 Female      1    <NA>
```

The main purpose of this example is to try to understand, through graphs and models, how survival was related to age and sex. However, first, we do some data preparation and exploration. The response variable, `survived` is a 0/1 integer, and it is more convenient for some purposes to make it a factor.

```
Donner$survived <- factor(Donner$survived, labels=c("no", "yes"))
```

Some historical accounts (Grayson, 1990) link survival in the Donner Party to kinship or family groups, so we take a quick look at this factor here. The variable `family` reflects a recoding of the last names of individuals to reduce the number of factor levels. The main families in the Donner party were: Donner, Graves, Breen and Reed. The families of Murphy, Foster and Pike are grouped as "MurFosPik", those of Fostick and Wolfinger are coded as "FosdWolf", and all others as "Other".

```
xtabs(~family, data=Donner)

## family
##      Breen  Donner  Eddy  FosdWolf  Graves  Keseberg
##      9      14      4      4      10      4
## McCutchen MurFosPik  Other  Reed
##      3      12      23      7
```

¹⁰Most historical sources count the number in the Donner Party at 87 or 89. An exact accounting of the members of the Donner Party is difficult, because: (a) several people joined the party in mid-route, at Fort Bridger and in the Wasatch Mountains; (b) several rode ahead to search for supplies and one (Charles Stanton) brought two more with him (Luis and Salvador); (c) five people died before reaching the Sierra Nevada mountains. It incorporates updated information from Kristin Johnson's listing, <http://user.xmission.com/~octa/DonnerParty/Roster.htm>.

For the present purposes, we reduce these 10 family groups further, collapsing some of the small families into "Other", and reordering the levels. Assigning new values to the `levels()` of a factor is a convenient trick for recoding factor variables.

```
# collapse small families into "Other"
fam <- Donner$family
levels(fam)[c(3,4,6,7,9)] <- "Other"

# reorder, putting Other last
fam = factor(fam, levels(fam)[c(1, 2, 4:6, 3)])
Donner$family <- fam
xtabs(~family, data=Donner)

## family
##      Breen      Donner      Graves MurFosPik      Reed      Other
##      9         14         10         12         7         38
```

`xtabs()` then shows the counts of survival by these family groups:

```
xtabs(~survived+family, data=Donner)

##      family
## survived Breen Donner Graves MurFosPik Reed Other
##      no      0      7      3          6      1     25
##      yes     9      7      7          6      6     13
```

Plotting this distribution of survival by family with a formula gives a *spineplot*, a special case of the mosaic plot, or a generalization of a stacked bar plot, shown in Figure 7.13. The widths of the bars are proportional to family size, and the shading highlights in light blue the proportion who survived in each family.

```
plot(survived ~ family, data=Donner, col=c("pink", "lightblue"))
```

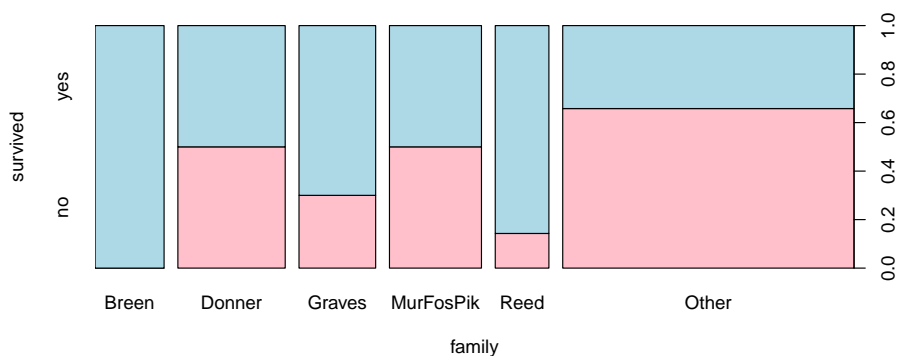


Figure 7.13: Spineplot of survival in the Donner Party by family. fig:donner1-spineplot

A generalized pairs plot (Section 5.5.1), shown in Figure 7.14 gives a visual overview of the data. The diagonal panels here show the marginal distributions of the variables as bar plots, and highlight the skewed distribution of age and the greater number of males than females in the party. The boxplots and barcode plots for `survived` and `age` show that those who survived were generally younger than those who perished.


```

library(gpairs)
library(vcd)
gpairs(Donner[,c(4,2,3,1)],
  diag.pars=list(fontsize=20, hist.color="gray"),
  mosaic.pars=list(gp=shading_Friendly), outer.rot=c(45,45)
)

```

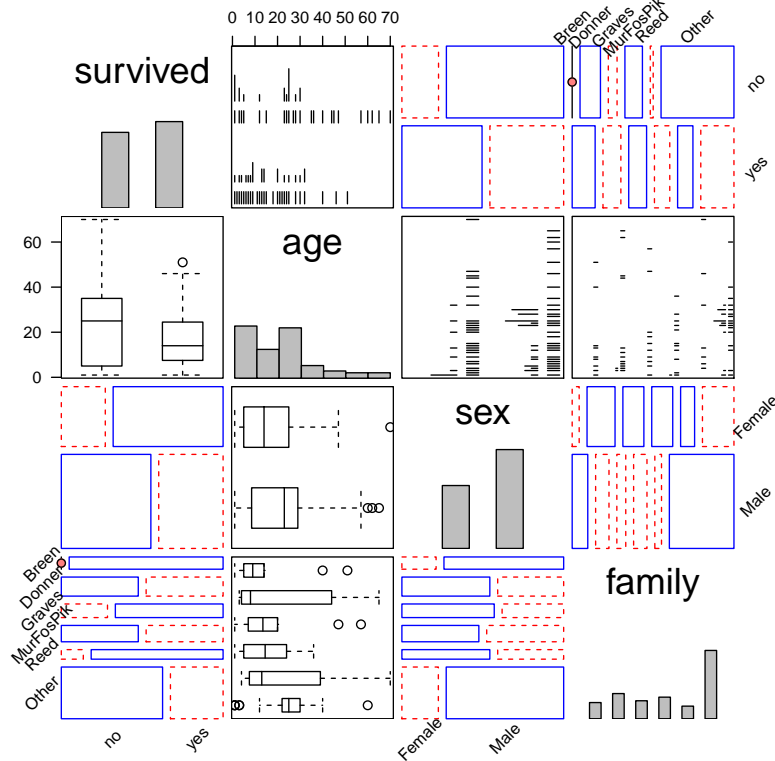


Figure 7.14: Generalized pairs plot for the Donner data fig:donner1-gpairs

From an exploratory perspective, we now proceed to examine the relationship of survival to age and sex, beginning with the kind of conditional plots we illustrated earlier (in Example 7.6). Figure 7.15 shows a plot of *survived*, converted back to a 0/1 variable as required by `ggplot()`, together with the binary responses as points and the fitted logistic regressions separately for males and females.

```

ggplot(Donner, aes(age, as.numeric(survived=="yes"), color = sex)) +
  theme_bw() + ylab("Survived") +
  geom_point(position = position_jitter(height = 0.02, width = 0)) +
  stat_smooth(method = "glm", family = binomial, formula = y ~ x,
    alpha = 0.2, size=2, aes(fill = sex))

```

It is easy to see that survival among women was greater than for men, perhaps narrowing the gap among the older people, but the data gets thin towards the upper range of age.

The curves plotted in Figure 7.15 assume a linear relationship between the log odds of survival and age (expressed as `formula = y ~ x` in the call to `stat_smooth()`). One simple way to check whether the relationship between survival and age is non-linear is to re-do this plot, but now allow a quadratic relationship with age, using `formula = y ~ poly(x, 2)`. The result is shown in the left panel of Figure 7.16.

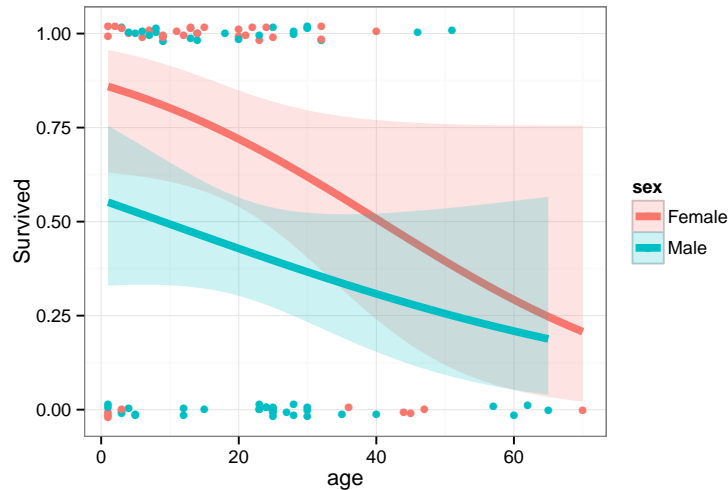


Figure 7.15: Conditional plot of the Donner data, showing the relationship of survival to age and sex. The smoothed curves and confidence bands show the result of fitting separate linear logistic regressions on age for males and females. fig:donner1-cond1

```
gg <- ggplot(Donner, aes(age, as.numeric(survived=="yes"), color = sex)) +
  theme_bw() + ylab("Survived") +
  geom_point(position = position_jitter(height = 0.02, width = 0))

gg + stat_smooth(method = "glm", family = binomial, formula = y ~ poly(x, 2),
  alpha = 0.2, size=2, aes(fill = sex))

gg + stat_smooth(method = "loess", span=0.9, alpha = 0.2, size=2,
  aes(fill = sex)) + coord_cartesian(ylim=c(-.05, 1.05))
```

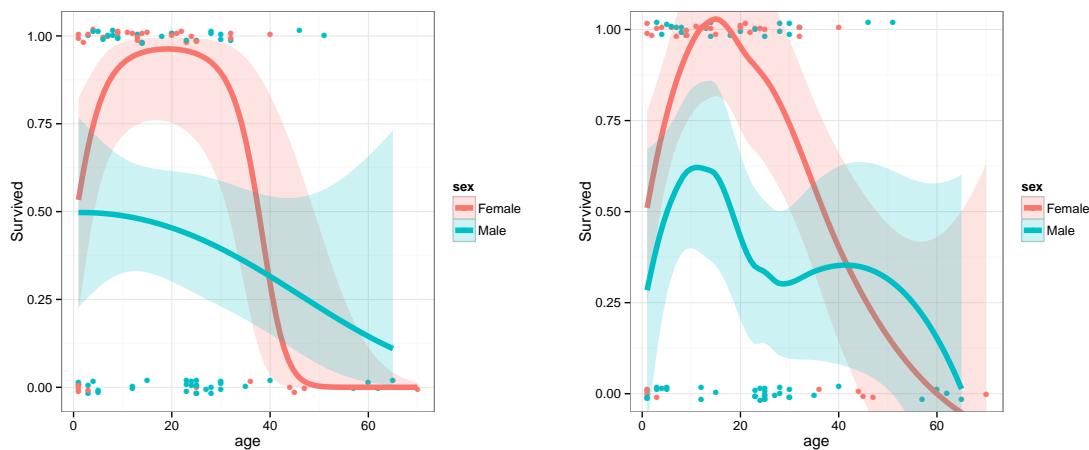


Figure 7.16: Conditionals plot of the Donner data, showing the relationship of survival to age and sex. Left: The smoothed curves and confidence bands show the result of fitting separate quadratic logistic regressions on age for males and females. Right: Separate loess smooths are fit to the data for males and females. fig:donner1-cond3

This plot is quite surprising. It suggests quite different regimes relating to survival for men and women. Among men, survival probability decreases steadily with age, at least after age 20.

For women, those in the age range 10–35 were very likely to have lived, while those over 40 were almost all predicted to perish.

Another simple technique is to fit a non-parametric loess smooth, as shown in the right panel of Figure 7.16. The curve for females is similar to that of the quadratic fit in the left panel, but the curve for males suggests that survival also has a peak around the teenage years. One lesson to be drawn from these graphs is that a linear logistic regression, such as shown in Figure 7.16 may tell only part of the story, and, for a binary response it is not easy to discern whether the true relationship is linear. If it really is, all these graphs would look much more similar. As well, we usually obtain a more realistic smoothing of the data using full-model plots or effect plots.

The suggestions from these exploratory graphs can be used to define and test some models for survival in the Donner Party. The substantive questions of interest are:

- Is relationship the same for men and women? This is, is it necessary to allow for an interaction of age with sex, or separate fitted curves for men and women?
- Is the relationship between survival and age well-represented in a linear logistic regression model?

The first question is the easiest to deal with: we can simply fit a model allowing an interaction of age (or some function of age) and sex,

```
survived ~ age * sex
survived ~ f(age) * sex
```

and compare the goodness of fit with the analogous additive, main-effects models.

From a modeling perspective, there is a wide variety of approaches for testing for non-linear relationships. We only scratch the surface here, and only for a single quantitative predictor, x , such as age in this example. One simple approach, illustrated in Figure 7.16 is to allow a quadratic (or higher-power, e.g., cubic) function to describe the relationship between the log odds and x ,

$$\begin{aligned} \text{logit}(\pi_i) &= \alpha + \beta_1 x_i + \beta_2 x_i^2 \\ \text{logit}(\pi_i) &= \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 \\ &\dots \end{aligned}$$

In R, these model terms can be fit using `poly(x, 2)`, `poly(x, 3)` ..., which generate orthogonal polynomials for the powers of x . A simple way to test for non-linearity is a likelihood ratio test comparing the more complex model to the linear one. This method is often sufficient for a hypothesis test, and, if the relationship truly is linear, the fitted logits and probabilities will not differ greatly from what they would be under a linear model. A difficulty with this approach is that polynomial models are often unrealistic, particularly for data that approach an asymptote.

Another simple approach is to use a *regression spline*, that fits the relationship with x in terms of a set of piecewise polynomials, usually cubic, joined at a collection of points, called *knots* so that the overall fitted relationship is smooth and continuous. See Fox (2008, §17.2) for a cogent, brief description of these methods.

One particularly convenient method is a *natural spline*, implemented in the `splines` package in the `ns()` function. This method constrains the fitted cubic spline to be linear at lower and upper limits of x , and, for k knots, fits $df = k + 1$ parameters not counting the intercept. The k

knots can be conveniently chosen as k cutpoints in the percentiles of the distribution of x . For example, with $k = 1$, the knot would be placed at the median, or 50th percentile; with $k = 3$, the knots would be placed at the quartiles of the distribution of x ; $k = 0$ corresponds to no knots, i.e., a simple linear regression.

In the `ns()` function, you can specify the locations of knots or the number of knots with the `knots` argument, but it is conceptually simpler to specify the number of degrees of freedom used in the spline fit. Thus, `ns(x, 2)` and `poly(x, 2)` both specify a term in x of the same complexity, the former a natural spline with $k = 1$ knot and the latter a quadratic function in x .

We illustrate these ideas in the remainder of this example, fitting a 2×2 collection of models to the `Donner` data corresponding to: (a) whether or not age and sex effects are additive; (b) whether the effect is linear on the logit scale or non-linear (quadratic, here). A brief summary of each model is given using the `Anova()` in the `car` package, providing Type II tests of each effect. As usual, `summary()` would give more detailed output, including tests for individual coefficients. First, we fit the linear models, without and with an interaction term:

```
donner.mod1 <- glm(survived ~ age + sex,
                  data=Donner, family=binomial)
Anova(donner.mod1)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
##      LR Chisq Df Pr(>Chisq)
## age      5.52  1  0.0188 *
## sex      6.73  1  0.0095 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

donner.mod2 <- glm(survived ~ age * sex,
                  data=Donner, family=binomial)
Anova(donner.mod2)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
##      LR Chisq Df Pr(>Chisq)
## age      5.52  1  0.0188 *
## sex      6.73  1  0.0095 **
## age:sex   0.40  1  0.5269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The main effects of `age` and `sex` are both significant here, but the interaction term, `age:sex` is not in model `donner.mod2`.

Next, we fit non-linear models, representing the linear and non-linear trends in age by `poly(age, 2)`. Alternatively, we could use the term `ns(age, 2)` or higher-degree polynomials or natural splines with more knots, but we don't do this here.

```
donner.mod3 <- glm(survived ~ poly(age,2) + sex,
                  data=Donner, family=binomial)
Anova(donner.mod3)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
```

```
##           LR Chisq Df Pr(>Chisq)
## poly(age, 2)      9.91  2    0.0070 **
## sex              8.09  1    0.0044 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

donner.mod4 <- glm(survived ~ poly(age,2) * sex,
                  data=Donner, family=binomial)
Anova(donner.mod4)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
##           LR Chisq Df Pr(>Chisq)
## poly(age, 2)      9.91  2    0.0070 **
## sex              8.09  1    0.0044 **
## poly(age, 2):sex  8.93  2    0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, in model `donner.mod4`, the interaction term `poly(age, 2):sex` is significant, indicating that the fitted quadratics for males and females differ in “shape,” meaning either their linear (slope) or quadratic (curvature) components.

These four models address the questions posed earlier. A compact summary of these models, giving the likelihood ratio tests of goodness of fit, together with AIC and BIC statistics are shown below, using the `Summarise()` method in `vcdExtra` for a list of “glm” models. **TODO: Summarise gives the wrong pvalues**

```
library(vcdExtra)
Summarise(donner.mod1, donner.mod2, donner.mod3, donner.mod4)

## Likelihood summary table:
##           AIC BIC LR Chisq Df Pr(>Chisq)
## donner.mod1 117 125   111.1 87    0.042 *
## donner.mod2 119 129   110.7 86    0.038 *
## donner.mod3 115 125   106.7 86    0.064 .
## donner.mod4 110 125    97.8 84    0.144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By AIC and BIC, `donner.mod4` is best, and it is also the only model with a non-significant LR χ^2 (residual deviance). Because these models comprise a 2×2 set of hypotheses, it is easier to compare models by extracting the LR statistics and arranging these in a table, together with the their row and column differences. The entries in the table below are calculated as follows.

```
mods <- list(donner.mod1, donner.mod2, donner.mod3, donner.mod4)
LR <- sapply(mods, function(x) x$deviance)
LR <- matrix(LR, 2, 2)
rownames(LR) <- c("additive", "non-add")
colnames(LR) <- c("linear", "non-lin")
LR <- cbind(LR, diff=LR[,1]-LR[,2])
LR <- rbind(LR, diff= c(LR[1,1:2]-LR[2,1:2], NA))
```

	linear	non-linear	$\Delta\chi^2$	<i>p</i> -value
additive	111.128	106.731	4.396	0.036
non-additive	110.727	97.799	12.928	0.000
$\Delta\chi^2$	0.400	8.932		
<i>p</i> -value	0.527	0.003		

Thus, the answer to our questions seems to be that: (a) there is evidence that the relationship of survival to age differs for men and women in the Donner Party; (b) these relationships are not well-described by a linear logistic regression.

For simplicity, we used a quadratic effect, `poly(age, 2)`, to test for non-linearity here. An alternative test of the same complexity could use a regression spline, `ns(age, 2)`, also with 2 degrees of freedom for the main effect and interaction, or allow more knots. To illustrate, we fit two natural spline modes models with 2 and 4 df, and compare these with the quadratic model (`donner.mod4`), all of which include the interaction of age and sex.

```
library(splines)
donner.mod5 <- glm(survived ~ ns(age, 2) * sex, data=Donner,
                  family=binomial)
Anova(donner.mod5)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
##           LR Chisq Df Pr(>Chisq)
## ns(age, 2)      9.28  2   0.0097 **
## sex             7.98  1   0.0047 **
## ns(age, 2):sex  8.71  2   0.0129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

donner.mod6 <- glm(survived ~ ns(age, 4) * sex, data=Donner,
                  family=binomial)
Anova(donner.mod6)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
##           LR Chisq Df Pr(>Chisq)
## ns(age, 4)    22.05  4   0.0002 ***
## sex          10.49  1   0.0012 **
## ns(age, 4):sex  8.54  4   0.0737 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Summarise(donner.mod4, donner.mod5, donner.mod6)

## Likelihood summary table:
##           AIC BIC LR Chisq Df Pr(>Chisq)
## donner.mod4 110 125   97.8  84   0.14
## donner.mod5 111 126   98.7  84   0.13
## donner.mod6 106 131   86.1  80   0.30
```

With four more parameters, `donner.mod6` fits better and has a smaller AIC.

We conclude this example with an effect plot for the spline model `donner.mod6` shown in Figure 7.17. The complexity of the fitted relationships for men and women is intermediate

between the two conditional plots shown in Figure 7.16. (However, note that the fitted effects are plotted on the logit scale in Figure 7.17 and labeled with the corresponding probabilities, whereas the conditional plots are plotted directly on the probability scale.)

```
library(effects)
donner.eff6 <- allEffects(donner.mod6, xlevels=list(age=seq(0,50,5)))
plot(donner.eff6, ticks=list(at=c(0.001, 0.01, 0.05, 0.1, 0.25,
                                0.5, 0.75, 0.9, 0.95, 0.99, 0.999)))
```

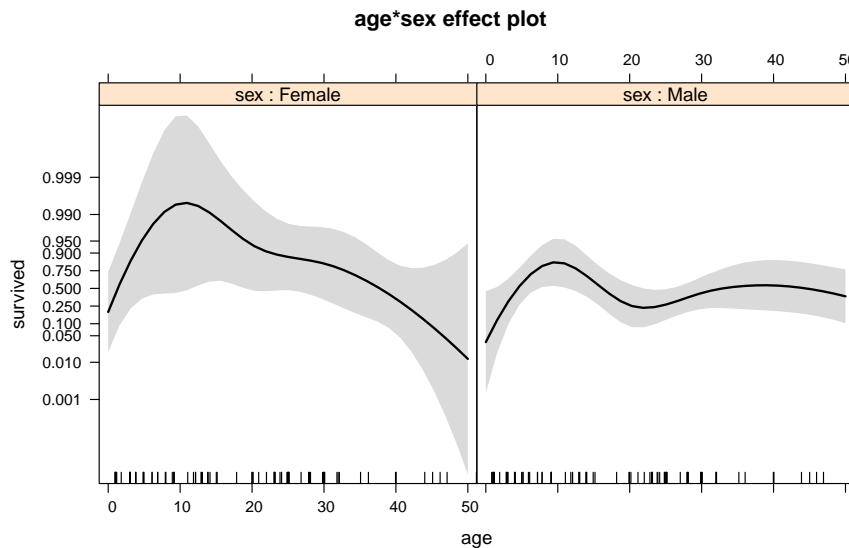


Figure 7.17: Effect plot for the Donner data fig:donner-effect

This plot confirms that for women in the Donner Party, survival was greatest for those aged 10-30. Survival among men was overall much less and there is a hint of greater survival for men aged 10-15.

Of course, this statistical analysis does not provide explanations for these effects, and it ignores the personal details of the Donner Party members and the individual causes and circumstances of death, which are generally well-documented in the historical record (Johnson, 1996). See <http://user.xmission.com/~octa/DonnerParty/> for a comprehensive collection of historical sources.

Grayson (1990) attributes the greater survival of women of intermediate age to demographic arguments that women are overall better able to withstand conditions of famine and extreme cold, and high age-specific mortality rates among the youngest and oldest members of human societies. He also concludes (without much analysis) that members with larger social and kinship networks would be more likely to survive. △

{ex:arrests}

EXAMPLE 7.10: Racial profiling: Arrests for marijuana possession

In the summer of 2002, the *Toronto Star* newspaper launched an investigation on the topic of possible racial profiling by the Toronto police service. Through freedom of information requests, they obtained a data base of over 600,000 arrest records on all potential charges in the period from 1996–2002, the largest data bases on crime arrests and disposition ever assembled in Canada. An initial presentation of this study was given in Example 1.4.

In order to examine the issue of racial profiling (different treatment as a function of race) they excluded all charges such as assault, robbery, speeding and driving under the influence, where the police have no discretion regarding the laying of a charge. They focused instead on a subset of arrests, where the police had various options.

Among these, for people arrested for a single charge of simple possession of a small amount of marijuana, police have the option of releasing the arrestee, with a summons (“Form 9”) to appear in court (similar to a parking ticket), or else the person could be given harsher treatment—brought to a police station or held in jail for a bail hearing (“Show cause”). The main question for the *Toronto Star* was whether the subject’s skin color had any influence on the likelihood that the person would be released with a summons.¹¹

Their results, published in a week-long series of articles in December 2002, concluded that there was strong evidence that black and white subjects were treated differently. For example, the analysis showed that blacks were 1.5 times more likely than whites to be given harsher treatment than release with a summons; if the subject was taken to the police station, a black was 1.6 times more likely to be held in jail for a bail hearing. An important part of the analysis and the public debate that ensued was to show that other variables that might account for these differences had been controlled or adjusted for.¹²

The data set `Arrests` in the `effects` package gives a simplified version of the *Star* database, containing records for 5226 cases of arrest on the charge of simple possession of marijuana analyzed by the newspaper. The response variable here is `released` (Yes/No) and the main predictor of interest is skin color of the person arrested, `colour` (Black/White).¹³ A random subset of the data set is shown below.

```
library(effects)
data("Arrests", package="effects")
Arrests[sample(nrow(Arrests), 6), ]

##      released colour year age  sex  employed citizen checks
## 3768      Yes  Black 2000  23 Male      No      Yes      4
## 4576      Yes  Black 2001  17 Male      Yes     Yes      0
## 3976      No   White 2002  20 Male      No      Yes      3
## 4629      Yes  White 2000  18 Male      Yes     Yes      1
## 2384      No   Black 2000  19 Male      Yes     Yes      3
##  869      Yes  White 2001  15 Male      Yes     Yes      1
```

Other available predictors, to be used as control variables included the `year` of the arrest, `age` and `sex` of the person, and binary indicators of whether the person was `employed` and a `citizen` of Canada. In addition, when someone is stopped by police, his/her name is checked in six police data bases that record previous arrests, convictions, whether on parole, etc. The variable `checks` records the number, 0–6, in which the person’s name appeared.

¹¹Another discretionary charge they investigated was police stops for non-moving violations under the Ontario *Highway Traffic Act*, such as being pulled over for a faulty muffler or having an expired license plate renewal sticker. A disproportionate rate of charges against blacks is sometimes referred to as “driving while black” (DWB). This investigation found that the number of blacks so charged, but particularly young black males, far out-weighted their representation in the population.

¹²The Toronto Police Service launched a class-action libel law suit against the *Toronto Star* and the first author of this book, who served as their statistical consultant, claiming damages of \$5,000 for every serving police officer in the city, a total of over 20 million dollars. The suit was thrown out of court, and the Toronto police took efforts to enhance training programs to combat the perception of racial profiling.

¹³The original data set also contained the categories Brown and Other, but these appeared with small frequencies.

A variety of logistic models were fit to these data including all possible main effects and some two-way interactions. To allow for possible non-linear effects of *year*, this variable was treated as a factor rather than as a (linear) numeric variable, but the effects of *age* and *checks* were reasonably linear on the logit scale. A reasonable model included the interactions of *colour* with both *year* and *age*, as fit below:

```
Arrests$year <- as.factor(Arrests$year)
arrests.mod <- glm(released ~ employed + citizen + checks
                  + colour*year + colour*age,
                  family=binomial, data=Arrests)
```

For such models, significance tests for the model terms are best carried out using the `Anova()` function in the `car` package that uses Type II tests ...

```
library(car)
Anova(arrests.mod)

## Analysis of Deviance Table (Type II tests)
##
## Response: released
##          LR Chisq Df Pr(>Chisq)
## employed      72.7  1 < 2e-16 ***
## citizen       25.8  1 3.8e-07 ***
## checks       205.2  1 < 2e-16 ***
## colour        19.6  1 9.7e-06 ***
## year           6.1  5 0.29785
## age            0.5  1 0.49827
## colour:year   21.7  5 0.00059 ***
## colour:age   13.9  1 0.00019 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The difficulty in interpreting these results from tables of coefficients can be seen in the output below:

```
coeftest(arrests.mod)

##
## z test of coefficients:
##
##          Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.34443   0.31007    1.11 0.26665
## employedYes    0.73506   0.08477    8.67 < 2e-16 ***
## citizenYes     0.58598   0.11377    5.15 2.6e-07 ***
## checks        -0.36664   0.02603   -14.08 < 2e-16 ***
## colourWhite    1.21252   0.34978    3.47 0.00053 ***
## year1998      -0.43118   0.26036   -1.66 0.09770 .
## year1999      -0.09443   0.26154   -0.36 0.71805
## year2000      -0.01090   0.25921   -0.04 0.96647
## year2001       0.24306   0.26302    0.92 0.35541
## year2002       0.21295   0.35328    0.60 0.54664
## age            0.02873   0.00862    3.33 0.00086 ***
## colourWhite:year1998 0.65196   0.31349    2.08 0.03756 *
## colourWhite:year1999 0.15595   0.30704    0.51 0.61152
## colourWhite:year2000 0.29575   0.30620    0.97 0.33411
## colourWhite:year2001 -0.38054   0.30405   -1.25 0.21073
## colourWhite:year2002 -0.61732   0.41926   -1.47 0.14091
## colourWhite:age -0.03737   0.01020   -3.66 0.00025 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By direct calculation (e.g., using `exp(coef(arrests.mod))`) you can find that the odds of a quick release was $\exp(0.735) = 2.08$ times greater for someone employed, $\exp(0.586) = 1.80$ times more likely for a Canadian citizen and $\exp(1.21) = 3.36$ times more likely for a white than a black person. It is much more difficult to interpret the interaction terms.

The primary question for the newspaper concerned the overall difference between the the treatment of blacks and whites— the main effect of `colour`. We plot this as shown below, giving the plot shown in Figure 7.18. This supports the claim by the *Star* because the 95% confidence limits for blacks and whites do not overlap, and all other relevant predictors that could account for this effect have been controlled or adjusted for.

```
plot(Effect("colour", arrests.mod),
     lwd=3, ci.style="bands", main="",
     xlab = list("Skin color of arrestee", cex=1.25),
     ylab = list("Probability(released)", cex=1.25)
)
```

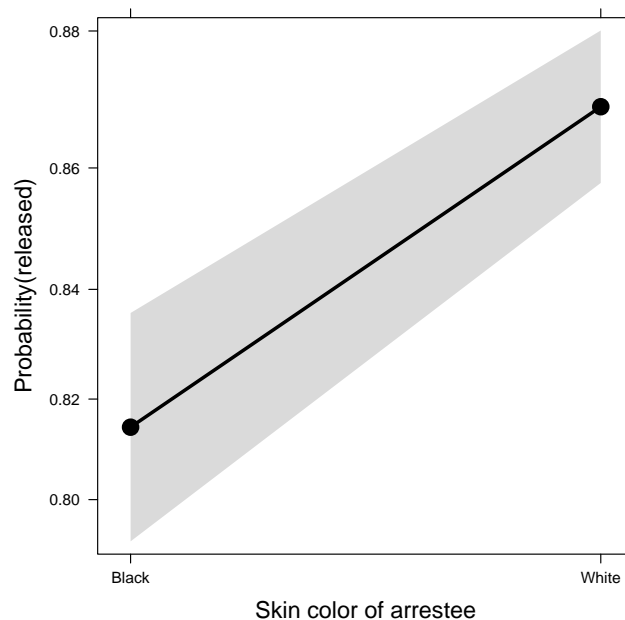


Figure 7.18: Effect plot for the main effect of skin color in the Arrests data.^{fig:arrests-eff1}

Of course, one should be very wary of interpreting main effects when there are important interactions, and the story turned out to be far more nuanced than was reported in the newspaper. In particular, the interactions of color with with age and year provided a more complete account. Effect plots for these interactions are shown in Figure 7.19.

```
# colour x age interaction
plot(Effect(c("colour", "age"), arrests.mod),
     lwd=3, multiline=TRUE,
     xlab=list("Age", cex=1.25),
     ylab=list("Probability(released)", cex=1.25),
     key.args=list(x=.05, y=.99, cex=1.2)
)
# colour x year interaction
```

```
plot(Effect(c("colour", "year"), arrests.mod),
     lwd=3, multiline=TRUE,
     xlab=list("Year", cex=1.25),
     ylab=list("Probability(released)", cex=1.25),
     key.args=list(x=.7, y=.99, cex=1.2)
)
```

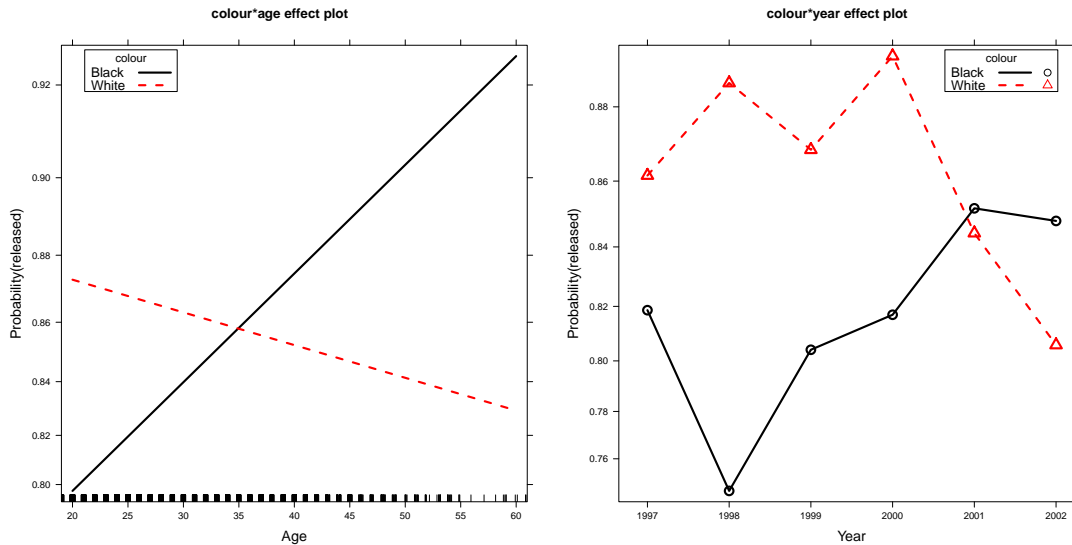


Figure 7.19: Effect plots for the interactions of color with age (left) and year (right) in the Arrests data.

From the left panel in Figure 7.19, it is immediately apparent that the effect of age was in opposite directions for blacks and whites: Young blacks were indeed treated more severely than young whites; however for older people, blacks were treated less harshly than whites, controlling for all other predictors.

The right panel of Figure 7.19 shows the changes over time in the treatment of blacks and whites. It can be seen that up to the year 2000 there was strong evidence for differential treatment on these charges, again controlling for other predictors. There was also evidence to support the claim by the police that in the year 2001 they began training of officers to reduce racial effects in treatment.

Finally, the `effects` package provides a convenience function, `allEffects()`, that calculates the effects for all high-order terms in a given model. The `plot()` method for the "efflist" object can be used to plot individual terms selectively from a graphic menu, or plot all terms together in one comprehensive display using `ask=FALSE`.

```
arrests.effects <- allEffects(arrests.mod,
                             xlevels=list(age=seq(15, 45, 5)))
plot(arrests.effects,
     ylab="Probability(released)", ci.style="bands", ask=FALSE)
```

The result, shown in Figure 7.20 is a relatively compact and understandable summary of the `arrests.mod` model: (a) people were more likely to be released if they were employed and citizens. (b) each additional police check decreased the likelihood of release with a summons. (c) the effect of skin color varied with age and year of arrest, in ways that tell a far more nuanced story than reported in the newspaper.

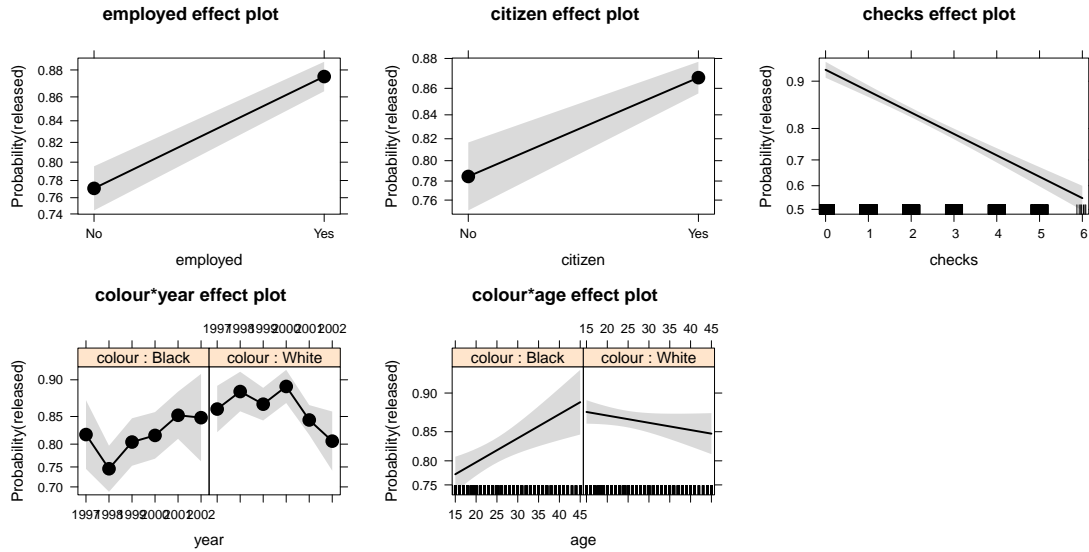


Figure 7.20: Effect plot for all high-order terms in the model for the Arrests data fig:arrests-all

Finally, another feature of this plot bears mention: by default, the scales for each effect plot are determined separately for each effect, to maximize use of the plot region. However, you have to read the Y scale values to judge the relative sizes of these effects. An alternative plot, using the *same* scale in each subplot¹⁴ would show the relative sizes of these effects.

△

7.4.1 More complex models: Model selection and visualization

{sec:complex}

Models with more predictors or more complex terms (interactions, non-linear terms) present additional challenges for model fitting, summarization, and visualization and interpretation. A very complicated model, with many terms and interactions may fit the data at hand quite well. However, because goodness-of-fit is optimized in the sample, terms that appear significant are less likely to be important in a future sample, and we need to worry about inflation of Type I error rates that accompany multiple significance tests. As well, it becomes increasingly difficult to visualize and understand a fitted model as the model becomes increasingly complex.

On the other hand, a very simple model may omit important predictors, interactions, or non-linear relationships with the response and give an illusion of a comfortable interpretation.

{ex:icul} **TODO:** Complete this brief introduction to model selection and define AIC/BIC,

EXAMPLE 7.11: Death in the ICU

In this example we examine briefly some aspects of logistic regression related to model selection and graphical display with a large collection of potential predictors, including both quantitative and discrete variables. We use data from a classic study by Lemeshow *et al.* (1988) of patients admitted to an intensive care unit at Baystate Medical Center in Springfield, Massachusetts. The major goal of this study was to develop a model to predict the probability of survival (until hospital discharge) of these patients and to study the risk factors associated with ICU mortality. The

¹⁴With the `effects` package, you can set the `ylim` argument to equate the vertical range for all plots. For this plot, `ylim = c(0.5, 1)` would work.

data, contained in the data set `ICU` in `vcdExtra`, gives the results for a sample of 200 patients that was presented in Hosmer *et al.* (2013) (and earlier editions).

The `ICU` contains 22 variables of which the first, `died` is a factor. Among the predictors, two variables (`race`, `coma`) were represented initially as 3-level factors, but then recoded to binary variables (`white`, `uncons`).

```
data("ICU", package="vcdExtra")
names(ICU)

## [1] "died"      "age"      "sex"      "race"      "service"
## [6] "cancer"   "renal"    "infect"   "cpr"       "systolic"
## [11] "hrtrate"  "previcu"  "admit"    "fracture"  "po2"
## [16] "ph"       "pco"      "bic"      "creatin"   "coma"
## [21] "white"    "uncons"

ICU <- ICU[,-c(4, 20)] # remove redundant race, coma
```

Removing the 3-level versions leaves 19 predictors, of which three (age, heart rate, systolic blood pressure) are quantitative and the remainder are either binary (service, cancer) or had previously been dichotomized ($ph < 7.25$).

As an initial step, and a basis for comparison, we fit the full model containing all 19 predictors.

```
icu.full <- glm(died ~ ., data=ICU, family=binomial)
summary(icu.full)

##
## Call:
## glm(formula = died ~ ., family = binomial, data = ICU)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8040  -0.5606  -0.2044  -0.0863   2.9773
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -6.72670    2.38551  -2.82   0.0048 **
## age            0.05639    0.01862   3.03   0.0025 **
## sexMale        0.63973    0.53139   1.20   0.2286
## serviceSurgical -0.67352    0.60190  -1.12   0.2631
## cancerYes      3.10705    1.04585   2.97   0.0030 **
## renalYes      -0.03571    0.80165  -0.04   0.9645
## infectYes     -0.20493    0.55319  -0.37   0.7110
## cprYes        1.05348    1.00661   1.05   0.2953
## systolic     -0.01547    0.00850  -1.82   0.0686 .
## hrtrate      -0.00277    0.00961  -0.29   0.7732
## previcuYes    1.13194    0.67145   1.69   0.0918 .
## admitEmergency 3.07958    1.08158   2.85   0.0044 **
## fractureYes   1.41140    1.02971   1.37   0.1705
## po2<=60       0.07382    0.85704   0.09   0.9314
## ph<7.25       2.35408    1.20880   1.95   0.0515 .
## pco>45       -3.01844    1.25345  -2.41   0.0160 *
## bic<18       -0.70928    0.90978  -0.78   0.4356
## creatin>2     0.29514    1.11693   0.26   0.7916
## whiteNon-white 0.56573    0.92683   0.61   0.5416
## unconsYes     5.23229    1.22630   4.27   2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 200.16 on 199 degrees of freedom
## Residual deviance: 120.78 on 180 degrees of freedom
## AIC: 160.8
##
## Number of Fisher Scoring iterations: 6
```

You can see that a few predictors are individually significant, but many are not.

However, it is useful to carry out a simultaneous global test of $H_0: \beta = 0$ that *all* regression coefficients are zero. If this test is not significant, it makes little sense to use selection methods to choose individually significant predictors. For convenience, we define a simple function, `LRtest()`, to calculate the likelihood ratio test from the model components.

```
LRtest <- function(model)
  c(LRchisq=(model$null.deviance - model$deviance),
    df=(model$df.null - model$df.residual))

(LR <- LRtest(icu.full))

## LRchisq      df
## 79.383  19.000

(pvalue=1-pchisq(LR[1],LR[2]))

## LRchisq
## 2.3754e-09
```

At this point, it is tempting to examine the output from `summary(icu.full)` shown above and eliminate those predictors which fail significance at some specified level such as the conventional $\alpha = 0.05$. This is generally a bad idea for many reasons.¹⁵

A marginally better approach is to remove non-significant variables whose coefficients have signs that don't make sense from the substance of the problem. For example, in the full model, both `renal` (history of chronic renal failure) and `infect` (infection probable at ICU admission) have negative signs, meaning that their presence *decreases* the odds of death. We remove those variables using `update()`; as expected they make little difference.

```
icu.full1 <- update(icu.full, . ~ . - renal - fracture)
anova(icu.full1, icu.full, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: died ~ age + sex + service + cancer + infect + cpr + systolic +
## hrtrate + previcu + admit + po2 + ph + pco + bic + creatin +
## white + uncons
## Model 2: died ~ age + sex + service + cancer + renal + infect + cpr +
## systolic + hrtrate + previcu + admit + fracture + po2 + ph +
## pco + bic + creatin + white + uncons
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      182      122      122
## 2      180      121      2      1.7      0.43
```

¹⁵It ignores the facts of (a) an arbitrary cutoff value for significance, (b) the strong likelihood that chance features of the data or outliers influence the result, (c) problems of collinearity, etc. See Harrell (2001, §4.3) for a useful discussion of these issues.

Before proceeding to consider model selection, it is useful to get a better visual overview of the current model than is available from a table of coefficients and significance tests. Some very useful `print()`, `summary()` and `plot()` methods are available in the `rsm` package. Unfortunately, these require that the logistic model is fitted with `lrm()` in that package rather than with `glm()`. We pause here to refit the same model as `icu.full1` in order to show a plot of odds ratios for the terms in this model.

```
library(rms)
dd <- datadist(ICU[,-1])
options(datadist="dd")
icu.lrm1 <- lrm(died ~ ., data=ICU)
icu.lrm1 <- update(icu.lrm1, . ~ . - renal - fracture)
```

The `summary()` method for "rms" objects produces a much more detailed descriptive summary of a fitted model, and the `plot()` method for that summary object gives a sensible plot of the odds ratios for the model terms together with confidence intervals, at levels (0.9, 0.95, 0.99) by default. The following lines produce Figure 7.21.

```
sum.lrm1 <- summary(icu.lrm1)
plot(sum.lrm1, log=TRUE, main="Odds ratio for 'died'", cex=1.25,
     col = rgb(0.1, 0.1, 0.8, alpha = c(0.3, 0.5, 0.8)))
```

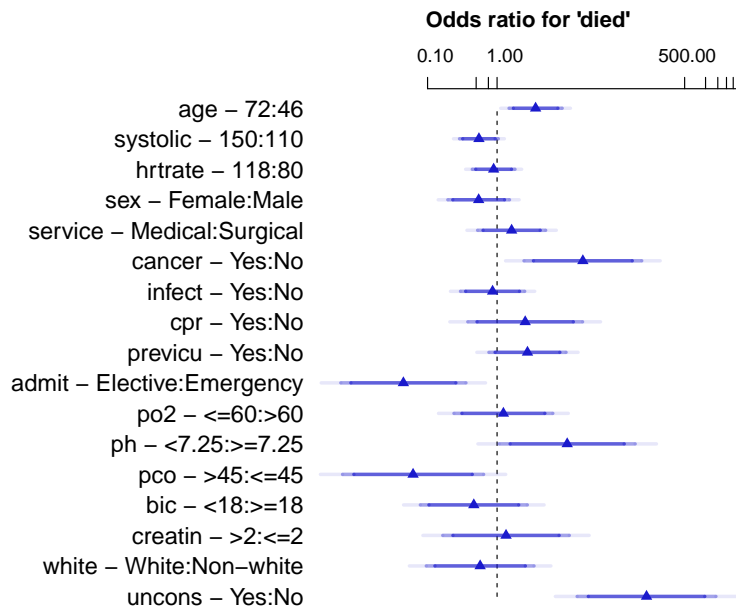


Figure 7.21: Odds ratios for the terms in the model for the ICU data. Each line shows the odds ratio for a term, together with lines for 90, 95 and 99% confidence intervals in progressively darker shades.

{fig:icul-odds-ratios}

In this plot, continuous variables are shown at the top, followed by the discrete predictors. In each line, the range or levels of the predictors are given in the form $a : b$, such that the value a corresponds to the numerator of the odds ratio plotted. Confidence intervals that don't overlap the vertical line for odds ratio = 1 are significant, but this graph shows those at several confidence levels, allowing you to decide what is "significant" visually. As well, the widths of those intervals convey the precision of these estimates.

Among several stepwise selection methods in R for "glm" models, `stepAIC()` in the MASS package implements a reasonable collection of methods for forward, backward and stepwise selection using penalized AIC-like criteria that balance goodness of fit against parsimony. The method takes an argument, `scope`, which is a list of two model formulae; `upper` defines the largest (most complex) model to consider and `lower` defines the smallest (simplest) model, e.g., `lower = ~ 1` is the intercept-only model.

By default, the function produces verbose printed output showing the details of each step, but we suppress that here to save space. It returns the final model as its result, along with an `anova` component that summarises the deviance and AIC from each step.

```
library(MASS)
icu.step1 <- stepAIC(icu.full1, trace = FALSE)
icu.step1$anova

## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## died ~ age + sex + service + cancer + infect + cpr + systolic +
##       hrtrate + previcu + admit + po2 + ph + pco + bic + creatin +
##       white + uncons
##
## Final Model:
## died ~ age + cancer + systolic + admit + ph + pco + uncons
##
##
##           Step Df Deviance Resid. Df Resid. Dev   AIC
## 1                182    122.48 158.48
## 2      - po2      1 0.062446    183    122.54 156.54
## 3 - creatin      1 0.059080    184    122.60 154.60
## 4 - hrtrate      1 0.072371    185    122.67 152.67
## 5   - infect      1 0.122772    186    122.79 150.79
## 6    - white      1 0.334999    187    123.13 149.13
## 7 - service      1 0.671313    188    123.80 147.80
## 8      - bic      1 0.377521    189    124.18 146.18
## 9      - cpr      1 1.148260    190    125.33 145.33
## 10     - sex      1 1.543523    191    126.87 144.87
## 11 - previcu     1 1.569976    192    128.44 144.44
```

Alternatively, we can use the BIC criterion, by specifying `k=log(n)`, which generally will select a smaller model when the sample size is reasonably large.

```
icu.step2 <- stepAIC(icu.full, trace = FALSE, k=log(200))
icu.step2$anova

## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## died ~ age + sex + service + cancer + renal + infect + cpr +
##       systolic + hrtrate + previcu + admit + fracture + po2 + ph +
##       pco + bic + creatin + white + uncons
##
## Final Model:
## died ~ age + cancer + admit + uncons
##
##
##           Step Df Deviance Resid. Df Resid. Dev   AIC
## 1                180    120.78 226.74
```



```
## 2 - renal 1 0.0019881 181 120.78 221.45
## 3 - po2 1 0.0067968 182 120.79 216.16
## 4 - creatin 1 0.0621463 183 120.85 210.92
## 5 - hrtrate 1 0.0658870 184 120.92 205.69
## 6 - infect 1 0.2033221 185 121.12 200.59
## 7 - white 1 0.3673180 186 121.49 195.66
## 8 - bic 1 0.6002993 187 122.09 190.96
## 9 - service 1 0.7676303 188 122.85 186.43
## 10 - fracture 1 1.3245086 189 124.18 182.46
## 11 - cpr 1 1.1482598 190 125.33 178.31
## 12 - sex 1 1.5435228 191 126.87 174.55
## 13 - previcu 1 1.5699762 192 128.44 170.83
## 14 - ph 1 4.4412370 193 132.88 169.97
## 15 - pco 1 2.7302934 194 135.61 167.40
## 16 - systolic 1 3.5231028 195 139.13 165.63
```

This model differs from model `icu.step1` selected using AIC in the last three steps, that also removed `ph`, `pco` and `systolic`.

```
coeftest(icu.step2)

##
## z test of coefficients:
##
##          Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.8698    1.3188  -5.21 1.9e-07 ***
## age          0.0372    0.0128   2.91 0.00360 **
## cancerYes    2.0971    0.8385   2.50 0.01238 *
## admitEmergency 3.1022    0.9186   3.38 0.00073 ***
## unconsYes    3.7055    0.8765   4.23 2.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These two models are nested, so we can compare them directly using a likelihood ratio test from `anova()`.

```
anova(icu.step2, icu.step1, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: died ~ age + cancer + admit + uncons
## Model 2: died ~ age + cancer + systolic + admit + ph + pco + uncons
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      195         139
## 2      192         128  3     10.7    0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The larger model is significantly better by this test, but the smaller model is simpler to interpret. We retain these both as “candidate models” to be explored furth, but for ease in this example, we do so using the smaller model, `icu.step2`.

Another important step is to check for non-linearity of quantitative predictors such as `age` and interactions among the predictors. This is easy to do using `update()` and `anova()` as shown below. First, allow a non-linear term in `age`, and all two-way interactions of the binary predictors.

```
icu.glm3 <- update(icu.step2, . ~ . -age + ns(age, 3) + (cancer+admit+uncons)^2)
anova(icu.step2, icu.glm3, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: died ~ age + cancer + admit + uncons
## Model 2: died ~ cancer + admit + uncons + ns(age, 3) + cancer:admit +
##       cancer:uncons + admit:uncons
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         195         139
## 2         191         135  4     3.73    0.44
```

Next, we can check for interactions with *age*:

```
icu.glm4 <- update(icu.step2, . ~ . + age*(cancer+admit+uncons))
anova(icu.step2, icu.glm4, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: died ~ age + cancer + admit + uncons
## Model 2: died ~ age + cancer + admit + uncons + age:cancer + age:admit +
##       age:uncons
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         195         139
## 2         192         134  3     5.37    0.15
```

None of these additional terms have much effect.

So, we will tentatively adopt the simple main effects model, `icu.step2`, and consider how to visualize in interpret this result. One interesting display is a **nomogram** that shows how values on the various predictors translate into a predicted value of the log odds, and the relative strengths of their effects on this prediction. This kind of plot is shown in Figure 7.22, produced using `nomogram()` in the `rms` package as follows. It only works with models fit using `lrm()`, so we have to refit this model.

```
icu.lrm2 <- lrm(died ~ age + cancer + admit + uncons, data=ICU)
plot(nomogram(icu.lrm2), cex.var=1.2, lplabel="Log odds death")
```

In this nomogram, each predictor is scaled according to the size of its effect on a common scale of 0–100 “points.” A representative observation is shown by the marked points, corresponding to a person of age 60, without cancer, who was admitted to emergency and was unconscious at that time. Adding the points associated with each variable value gives the result shown on the scale of total points. For this observation, the result is $50 + 0 + 84 + 100 = 234$, for which the scale of log odds at the bottom gives a predicted logit of 2.2, or a predicted probability of death of $1/(1 + \exp(-2.2)) = 0.90$.

This leaves us with the problem of how to visualize the fitted model compactly and comprehensively. Full-model plots and effect plots, as we have used them, are somewhat unwieldy with four or more predictors if we want to view all effects simultaneously because it is more difficult to make comparisons across multiple panels (particularly if the vertical scales differ).

One way to reduce the visual complexity of such graphs is to combine some predictors that would otherwise be shown in separate panels into a recoding that can be shown as multiple curves for their combinations in fewer panels. In general, this can be done by combining some predictors interactively; for example with sex and education as factors, their combinations, M:Hi, M:Lo,

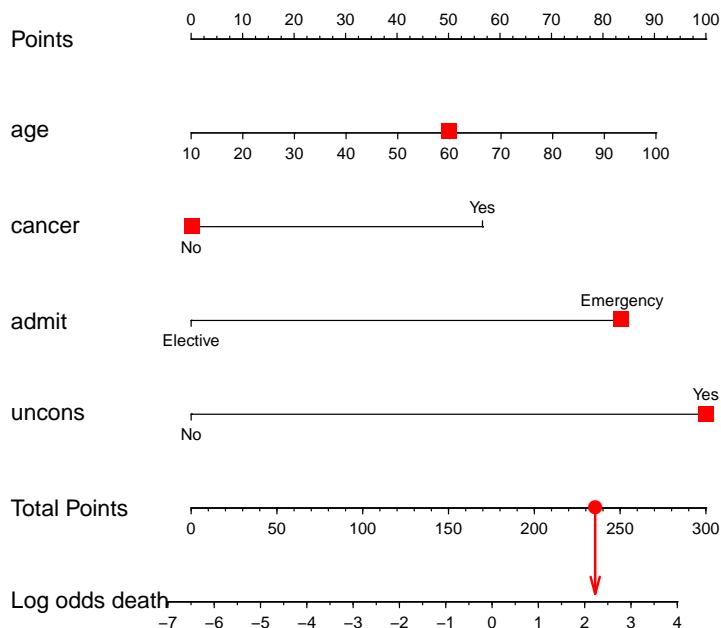


Figure 7.22: Nomogram for predicted values in the simple main effects model for the ICU data. Each predictor is scaled in relation to its effect on the outcome in terms of “points”, 0–100. Adding the points for a given case gives total points that have a direct translation to log odds. The marked points show the prediction for someone of age 60, admitted to the emergency ward and unconscious.

etc. could be used to define a new variable, `group` used as the curves in one plot, rather than separate panels.

In this case, because age is continuous, it makes sense to plot fitted values against age. With cancer, admit and uncons as binary factors associated with risk of death, it is also sensible to combine them all into a single variable, `risks`, indicating which one or more risk factors are present for each case. We first convert each variable to an abbreviation for the risk, if present, or “”, and paste these together.

```
# combine categorical risk factors to a single string
risks <- ICU[, c("cancer", "admit", "uncons")]
risks[,1] <- ifelse(risks[,1]=="Yes", "Cancer", "")
risks[,2] <- ifelse(risks[,2]=="Emergency", "Emerg", "")
risks[,3] <- ifelse(risks[,3]=="Yes", "Uncons", "")
risks <- apply(risks, 1, paste, collapse="")
risks[risks==""] <- "(none)"
```

```
table(risks)
```

```
## risks
##      (none)      Cancer CancerEmerg      Emerg EmergUncons
##          37          15           5         128          14
##      Uncons
##          1
```

The frequency counts of the risk combinations show that admission to the emergency ward alone was most frequent, and only one patient had unconsciousness as the only risk.

As done before, we can then get the fitted logit values for the chosen model, and combine these with the data and the risks variable.

```
icu.glm2 <- glm(died ~ age + cancer + admit + uncons,
               data=ICU, family=binomial)
icu.fit <- cbind(ICU, predict(icu.glm2, se=TRUE), risks)
```

In the plot step below, we use `geom_ribbon()` to plot a one standard error confidence band around the fitted logits. `color=risks` gives separate curves for each level of the risks factor.

```
gg <- ggplot( icu.fit, aes(x=age, y=fit, color=risks)) +
  geom_line(size = 1.2) + theme_bw() +
  geom_ribbon(aes(ymin = fit - se.fit,
                ymax = fit + se.fit,
                fill = risks),
            alpha = 0.2,
            color = "transparent") +
  theme_bw() +
  labs(x = "Age", y = "Log odds (died)") +
  geom_point(size=2)
```

By default, `ggplot()` uses a legend to display the labels for the curve variable, but the graph is more readable using `directlabels`, giving the plot shown in Figure 7.23.

```
library(directlabels)
direct.label(gg+xlim(10,100), last.points)
```

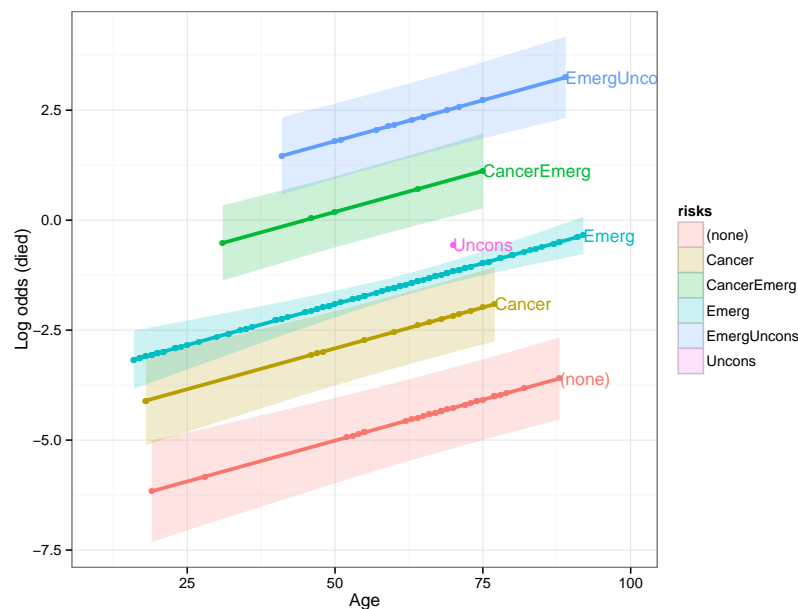


Figure 7.23: Fitted log odds of death in the ICU data. Each line shows the relationship with age, for patients having various combinations of risk factors.

From this graph, it is apparent that the log odds of mortality increases with age in all cases. Relative to the line labeled `(none)`, mortality is higher when any of these risk factors are present,

particularly when the patient is admitted to Emergency; it is highest when the patient is also unconscious at admission. The vertical gaps between lines that share a common risk (e.g., Cancer, CancerEmerg) indicate the additional increment from one more risk. Finally, the plotted points show the number and age distribution of these various combinations.

Before concluding that this model provides an adequate description of the data, we should examine whether any individual cases are unduly influencing the predicted results, and more importantly, the choice of variables in the model. We examine this question in Section 7.5 where we return to these data (Example 7.13).

△

7.5 Influence and diagnostic plots

{sec:logist-infl}

In ordinary least squares (OLS) regression, measures of *influence* (leverage, Cook's D, DFBE-TAs, etc.) and associated plots help you to determine whether individual cases (or cells in grouped data) have undue impact on the fitted regression model and the coefficients of individual predictors. Analogs of most of these measures have been suggested for logistic regression and generalized linear models. Pregibon (1981) provided the theoretical basis for these methods, exploiting the relationship between logistic models and weighted least squares. Some additional problems occur in practical applications to logistic regression because the response is discrete, and because the leave-one-out diagnostics are more difficult to compute, but the ideas are essentially the same.

7.5.1 Residuals and leverage

{sec:logist-resids}

As in ordinary least squares regression, the influence (actual impact) of an observation in logistic models depends multiplicatively on its residual (disagreement between y_i and \hat{y}_i) and its leverage (how unusual \mathbf{x}_i is in the space of the explanatory variables). A conceptual formula is

$$\text{Influence} = \text{Leverage} \times \text{Residual}$$

This multiplicative definition implies that a case is influential to the extent that it is both poorly fit *and* has unusual values of the predictors.

Residuals

In logistic regression, the simple raw residual is just $e_i \equiv y_i - \hat{p}_i$, where $\hat{p}_i = 1/[1 + \exp(-\mathbf{x}_i^T \mathbf{b})]$.

The Pearson and deviance residuals are more useful for identifying poorly fitted observations, and are components of overall goodness-of-fit statistics. The (raw) *Pearson residual* is defined as

$$r_i \equiv \frac{e_i}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}} \quad (7.7) \quad \{\text{eq:reschi}\}$$

and the Pearson chi-square is therefore $\chi^2 = \sum r_i^2$. The *deviance residual* is

$$g_i \equiv \pm 2[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)]^{1/2} \quad (7.8) \quad \{\text{eq:resdev}\}$$

where the sign of g_i is the same as that of e_i . Likewise, the sum of squares of the deviance residuals gives the overall deviance, $G^2 = -2 \log \mathcal{L}(\mathbf{b}) = \sum g_i^2$.

When y_i is a binomial count based on n_i trials (grouped data), the Pearson residuals Eqn. (7.7) then become

$$r_i \equiv \frac{y_i - n_i \hat{p}_i}{\sqrt{n_i \hat{p}_i (1 - \hat{p}_i)}}$$

with similar modifications made to Eqn. (7.8).

In R, `residuals()` is the generic function for obtaining (raw) residuals from a model fitted with `glm()` (or `lm()`). However *standardized residuals*, given by `rstandard()`, and *studentized residuals*, provided by `rstudent()` are often more useful because they rescale the residuals to have unit variance. They use, respectively, an overall estimate, $\hat{\sigma}^2$ of error variance, and the leave-one-out estimate, $\hat{\sigma}_{(-i)}^2$, omitting the i th observation; the studentized version is usually to be preferred in model diagnostics because it also accounts for the impact of the observation on residual variance.

Leverage

Leverage measures the *potential* impact of an individual case on the results, which is directly proportional to how far an individual case is from the centroid in the space of the predictors. Leverage is defined as the diagonal elements, h_{ii} , of the ‘‘Hat’’ matrix, \mathbf{H} ,

$$\mathbf{H} = \mathbf{X}^*(\mathbf{X}^{*\top} \mathbf{X}^*)^{-1} \mathbf{X}^{*\top}$$

where $\mathbf{X}^* = \mathbf{V}^{1/2} \mathbf{X}$, and $\mathbf{V} = \text{diag}[\hat{p}(1 - \hat{p})]$. As in OLS, leverage values are between 0 and 1, and a leverage value, $h_{ii} > \{2 \text{ or } 3\}k/n$ is considered ‘‘large’’; here, $k = p + 1$ is the number of coefficients including the intercept and n is the number of cases. In OLS, however, the hat values depend only on the X s, whereas in logistic regression, they also depend on the dependent variable values and the fitted probabilities (through \mathbf{V}). As a result, an observation may be extremely unusual on the predictors, yet not have a large hat value, if the fitted probability is near 0 or 1. The function `hatvalues()` calculates these values for a fitted ‘‘glm’’ model object.

7.5.2 Influence diagnostics

Influence measures assess the effect that deleting an observation has on the regression parameters, fitted values, or the goodness-of-fit statistics. In OLS, these measures can be computed exactly from a single regression. In logistic regression, the exact effect of deletion requires refitting the model with each observation deleted in turn, a time-intensive computation. Consequently, Pregibon (1981) showed how analogous deletion diagnostics may be approximated by performing one additional step of the iterative procedure. Most modern implementations of these methods for generalized linear models follow Williams (1987).

The simplest measure of influence of observation i is the standardized change in the coefficient for each variable due to omitting that observation, termed *DFBETAs*. From the relation (Pregibon, 1981, p. 716)

$$\mathbf{b} - \mathbf{b}_{(-i)} = (\mathbf{X}^\top \mathbf{V} \mathbf{X})^{-1} \mathbf{x}_i (y_i - \hat{p}_i) / (1 - h_{ii}),$$

the estimated standardized change in the coefficient for variable j is

$$\text{DFBETA}_{ij} \equiv \frac{b_{(-i)j} - b_j}{\hat{\sigma}(b_j)}, \quad (7.9)$$

{eq:dfbeta}

where $\hat{\sigma}(b_j)$ is the estimated standard error of b_j . With k regressors, there are $k + 1$ sets of DFBETAs, which makes their examination burdensome. Graphical displays ease this burden, as do various summary measures considered below.

The most widely used summary of the overall influence of observation i on the estimated regression coefficients is **Cook's distance**, which measures the average squared distance between \mathbf{b} for all the data and $\mathbf{b}_{(-i)}$ estimated without observation i . It is defined as

$$C_i \equiv (\mathbf{b} - \mathbf{b}_{(-i)})^\top \mathbf{X}^\top \mathbf{V} \mathbf{X} (\mathbf{b} - \mathbf{b}_{(-i)}) / k \hat{\sigma}^2 .$$

However, Pregibon (1981) showed that C_i could be calculated simply as

$$C_i = \frac{r_i^2 h_{ii}}{k(1 - h_{ii})^2} , \quad (7.10) \quad \{\text{eq:cookd2}\}$$

where $r_i = (y_i - \hat{p}_i) / \sqrt{v_{ii}(1 - h_{ii})}$ is the i th standardized Pearson residual and v_{ii} is the i th diagonal element of \mathbf{V} . Rules of thumb for noticeably "large" values of Cook's D are only rough indicators, and designed so that only "noteworthy" observations are nominated as unusually influential. One common cutoff for an observation to be treated as influential is $C_i > 1$. Others refer the values of C_i to a χ_k^2 or $F_{k,n-k}$ distribution.

Another commonly used summary statistic of overall influence is the **DFFITs** statistic, a standardized measure of the difference between the predicted value \hat{y}_i using all the data and the predicted value $\hat{y}_{(-i)}$ calculated omitting the i th observation.

$$\text{DFFITs}_i = \frac{\hat{y}_i - \hat{y}_{(-i)}}{\hat{\sigma}_{(-i)} \sqrt{h_{ii}}} ,$$

where $\hat{\sigma}_{(-i)}$ is the estimated standard error with the i th observation deleted. For computation, DFFITS can be expressed in terms of the standardized Pearson residual and leverage as

$$\text{DFFITs}_i = r_i \sqrt{\frac{h_{ii}}{(1 - h_{ii})} \frac{v_{ii}}{v_{(-ii)}}} . \quad (7.11) \quad \{\text{eq:dffits}\}$$

From Eqn. (7.10) and Eqn. (7.11) it can be shown that Cook's distance is nearly the square of DFFITS divided by k ,

$$C_i = \frac{v_{(-ii)}^2}{v_{ii}^2} \frac{\text{DFFITs}_i^2}{k} . \quad (7.12) \quad \{\text{eq:cook-dffits}\}$$

Noteworthy values of DFFITS are often nominated by the rule-of-thumb $\text{DFFITs}_i > 2$ or $3\sqrt{k/n - k}$.

In R, these influence measures are calculated for a fitted "glm" model using `cooks.distance()` and `dffits()`. The `dfbeta()` function calculates and returns the matrix of all standardized changes in the model coefficients (Eqn. (7.9)) due to omitting each observation in turn.¹⁶ A convenience function, `influence.measures()` gives a tabular display showing the DFBETA_{ij} for each model variable, DFFITS, Cook's distances and the diagonal elements of the hat matrix. Cases which are influential with respect to any of these measures are marked with an asterisk.¹⁷

Beyond printed output of these numerical summaries, plots of these measures can shed light on potential problems due to influential or other noteworthy cases. By highlighting them, such

¹⁶**TODO:** Not quite true: `dfbeta()` doesn't have a "glm" method. Either omit this or write a `dfbeta.glm()` method that gives the same results as `influence.measures()`.

¹⁷See `help(influence.measures)` for the description of all of these functions for residuals, leverage and influence diagnostics in generalized linear models.

plots provide the opportunity to determine if and how any of these affect your conclusions, or to take some corrective action.

A basic collection of diagnostic plots is provided by the `plot()` method for a "glm" model object. The `car` package contains a variety of other functions for model diagnostic plots. We illustrate some of these in the examples below.

{ex:donner2}

EXAMPLE 7.12: Donner Party

This example re-visits the data on the Donner Party examined in Example 7.9. For illustrative purposes, we consider the influence measures and diagnostic plots for one specific model, the model `donner.mod3`, that included a quadratic effect of age and a main effect of sex, but no interaction.

The simplest overview of the adequacy of a fitted model is provided by the `plot()` method for a "glm" (or "lm") object. This function can produce up to six different plots that can be plotted individually or selected (using the argument `which`) and composed into a single overview figure using `par(mfrow=c(rows, cols))` as shown below.

It is useful to see the entire collection because, by default, only four are plotted (`which=c(1:3, 5)`) and this selection (sometimes called the *regression quartet* of diagnostic plots) is tuned more to classical linear models for quantitative data. Important feature of these plots are that (a) plot annotations are added to each showing trends or expected behaviour under the assumptions of a fitted model; (b) noteworthy observations are labeled individually.

```
caption = list("(1) Residuals vs Fitted",
              "(2) Normal Q-Q",
              "(3) Scale-Location",
              "(4) Cook's distance",
              "(5) Residuals vs Leverage",
              expression("(6) Cook's dist vs Leverage "
                        * h[ii] / (1 - h[ii])))
op <- par(mfrow=c(3, 2), mar=c(4, 4, 2, 1)+.1, cex.lab=1.2, cex=1)
plot(donner.mod3, which=1:6, caption=caption)
par(op)
```

The six plots, corresponding to the values of `which`, shown in Figure 7.24 for the `donner.mod3` model are:

1. a plot of residuals against fitted values. In a classical linear model, this plot should appear unstructured (random around the zero line), but for logistic regression there will always be two sequences of points, corresponding to the 0/1 observations.
2. a normal Q-Q plot of ordered residuals vs. the corresponding quantiles of the gaussian distribution. In a classical linear model, all points should follow the dotted reference line, but this will rarely hold for logistic regression models.
3. a Scale-Location plot of $\sqrt{|\text{residuals}|}$ against fitted values, with a loess smoothed curve showing the trend for variance of the residual to change with the predicted value. This is useful to detect non-constant residual variance in classical models, but in logistic regression, you will almost always see a U-shaped pattern corresponding to the fact that the variance around the fitted value is a function of $\sqrt{\hat{p}_i(1 - \hat{p}_i)}$.
4. an index plot of Cook's distances versus observation numbers, with the largest `id.n` values labeled.
5. a plot of residuals against leverages, showing contours of Cook's distances. Among all of these plots, this is probably the most useful for assessment of influence in both classical

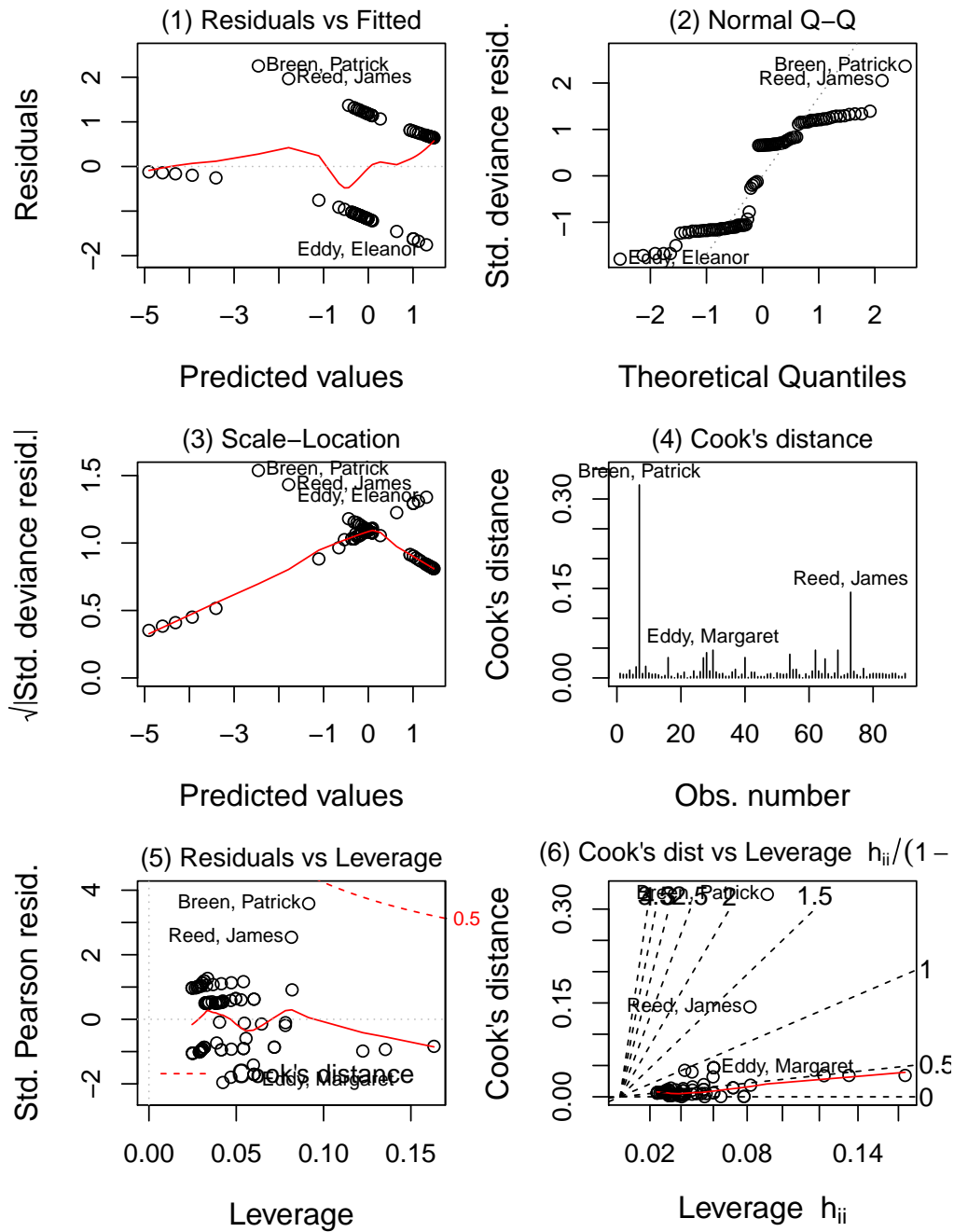


Figure 7.24: Diagnostic plots for a glm object, using the fitted model `donner.mod3` for the Donner Party data. Each plot shows some additional annotations or smoothed curves and labels observations considered noteworthy in terms of influence. Fig: donner2-plot

- and generalized linear models. The function `influencePlot()` in `car` provides a better version of this plot, using the size of a bubble symbol to also show Cook's distance directly.
- a plot of Cook's distances against leverage/(1-leverage). In this plot contours of standardized residuals that are equal in magnitude are lines through the origin, and labeled with their absolute values. Consequently, more influential observations appear toward the top.

In all these plots, three observations are labeled as noteworthy, by one criterion or another with a default number given by `id.n=3`. Plotting just the residual-leverage graph (`which=5`) with some additional annotations to show the conventional cutoff values gives Figure 7.25.

```
op <- par(mar=c(5, 4, 4, 2)+.1)
plot(donner.mod3, which=5, cex.id=1, cook.levels=c(0.25, 0.5), id.n=3)
abline(h=c(-2, 2), col="gray")
k <- length(coef(donner.mod3))
n <- nrow(Donner)
abline(v=c(2, 3)*k/n, col="gray")
text(x=c(2, 3)*k/n, y=-2.3, c("2k/n", "3k/n"))
par(op)
```

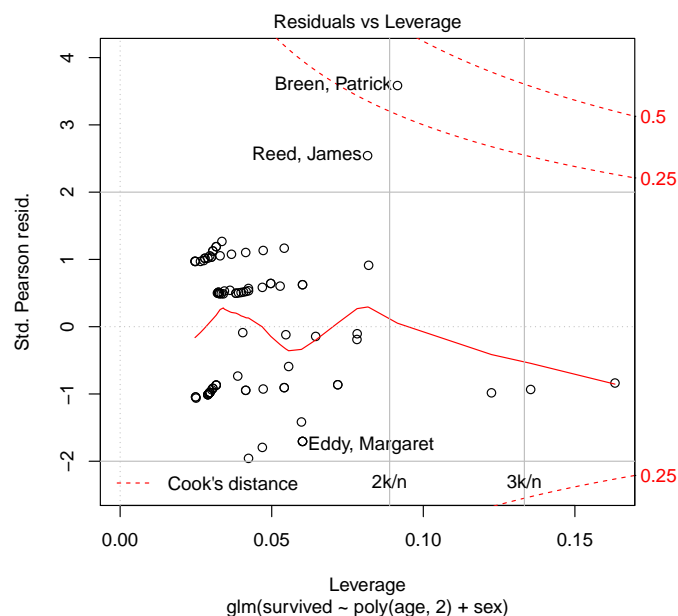


Figure 7.25: Residual vs. Leverage plot for the Donner data model. Horizontal and vertical reference lines show typical cutoff values for noteworthy residuals and leverage. Fig: donner2-plot5

Details of all the diagnostic measures for a given model including the DFBETAs for individual coefficients can be obtained using `influence.measures`. This can be useful for custom plots not provided elsewhere (see Example 7.13).

```
infl <- influence.measures(donner.mod3)
names(infl)

## [1] "infmtat" "is.inf" "call"
```

The `summary()` method for the "infl" object prints those observations considered noteworthy on one or more of these statistics, as indicated by a "*" next to the value.

```
summary(infl)
```

```
## Potentially influential observations of
##   glm(formula = survived ~ poly(age, 2) + sex, family = binomial,      data = Donner) :
##
##
##           dfb.1_ dfb.p(,2)1 dfb.p(,2)2 dfb.sxM1 dffit   cov.r   cook.d hat
## Breen, Patrick      0.08   0.65      0.56      0.23   0.69_*  0.93   0.32  0.09
## Donner, Elizabeth  -0.26  -0.34     -0.22     0.12  -0.40  1.15_*  0.03  0.14_*
## Graves, Elizabeth C. -0.24  -0.37     -0.26     0.10  -0.42  1.20_*  0.03  0.16_*
```

The function `influencePlot()` in the `car` package gives a similar plot, but uses the size (area) of the plotting symbol to also show the value of Cook's D as shown in Figure 7.26. Like other diagnostic plots in `car`, it is considerably more general than illustrated here, because it allows for different `id.methods` to label noteworthy points, including `id.method="identify"` for interactive point identification by clicking with the mouse. The `id.n` argument works differently than with `plot()`, because it selects the most extreme `id.n` observations on *each* of the studentized residual, hat value and Cook's D, and labels all of these.

```
library(car)
res <- influencePlot(donner.mod3, id.col="blue", scale=8, id.n=2)
text(x=c(2, 3)*k/n, y=-1.8, c("2k/n", "3k/n"))
```

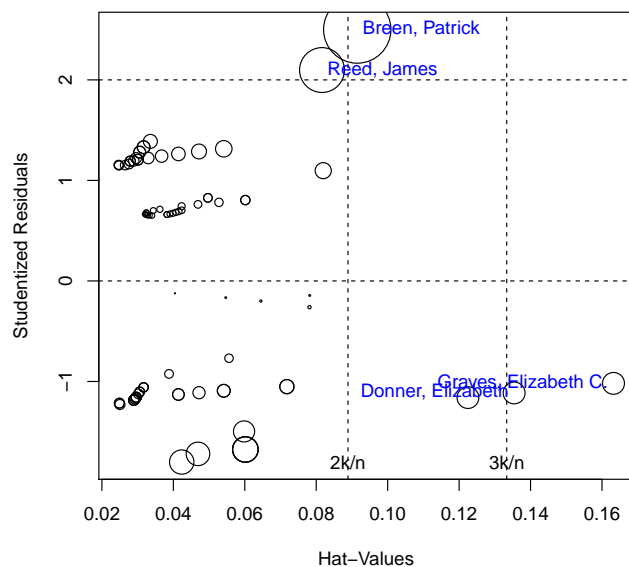


Figure 7.26: Influence plot (residual vs. leverage) for the Donner data model, showing Cook's D as the size of the bubble symbol. Horizontal and vertical reference lines show typical cutoff values for noteworthy residuals and leverage. fig:donner2-inflplot

Conveniently, `influencePlot()` returns a data frame containing the influence statistics for the points identified in the plot (`res` in the call above). We can combine this with the data values to help learn why these points are considered influential.

```
# show data together with diagnostics for influential cases
idx <- which(rownames(Donner) %in% rownames(res))
cbind(Donner[idx,2:4], res)

##          age  sex survived StudRes      Hat  CookD
## Breen, Patrick    51  Male     yes    2.501 0.09148 0.5688
## Donner, Elizabeth  45 Female     no   -1.114 0.13541 0.1846
## Graves, Elizabeth C. 47 Female     no   -1.019 0.16322 0.1849
## Reed, James       46  Male     yes    2.098 0.08162 0.3790
```

We can see that Patrick Breen and James Reed¹⁸ are unusual because they were both older men who survived, and have large positive residuals; Breen is the most influential by Cook's D, but this value is not excessively large. The two women were among the older women who died. They are selected here because they have the largest hat values, meaning they are unusual in terms of the distribution of age and sex, but they are not particularly influential in terms of Cook's D.

A related graphical display is the collection of index plots provided by `influenceIndexPlot()` in `car`, which plots various influence diagnostics against the observation numbers in the data. The `id.n` argument here works to label that number of the most extreme observations *individually* for each measure plotted. The following call produces Figure 7.27.

```
influenceIndexPlot(donner.mod3, vars=c("Cook", "Studentized", "hat"),
                  id.n=4)
```

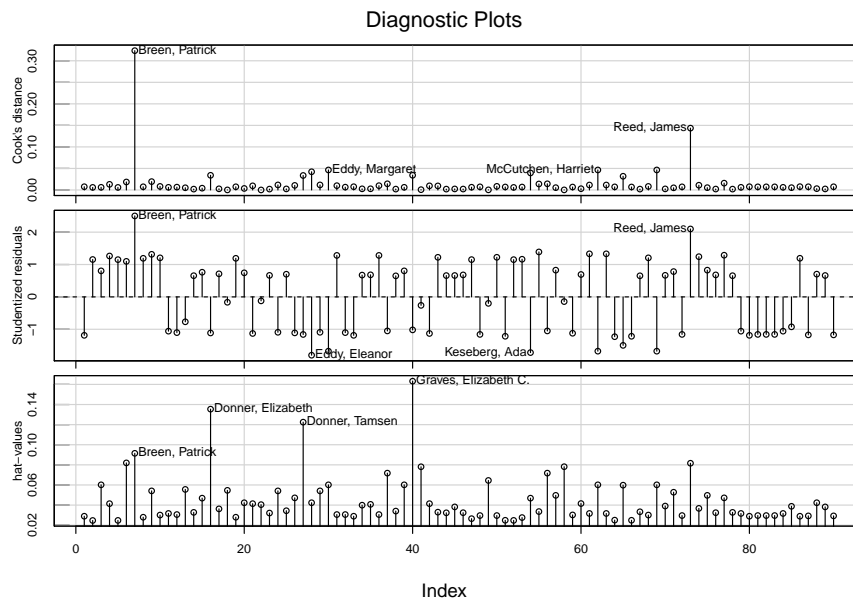


Figure 7.27: Index plots of influence measures for the Donner data model. The four most extreme observations on each measure are labeled. Fig: donner2-indexinfl

¹⁸Breen and Reed, both born in Ireland, were the leaders of their family groups. Among others, both kept detailed diaries of their experiences, from which most of the historical record derives. Reed was also the leader of two relief parties sent out to find rescue or supplies over the high Sierra mountains, so it is all the more remarkable that he survived.

In our opinion, *separate* index plots are often less useful than combined plots such as the leverage-influence plot that shows residuals, leverage and Cook's D together. However, the `car` version in Figure 7.27 does that too, and allows us to consider how unusual the labeled observations are both individually and in combination.

△

{ex:icu2}

EXAMPLE 7.13: Death in the ICU

In Example 7.11 we examined several models to account for death in the ICU data set. We continue this analysis here, with a focus on the simple main effects model, `icu.glm2`, for which the fitted logits were shown in Figure 7.23. For ease of reference, we restate that model here:

```
icu.glm2 <- glm(died ~ age + cancer + admit + uncons,
               data=ICU, family=binomial)
```

The plot of residual vs. leverage for this model is shown in Figure 7.28.

```
library(car)
res <- influencePlot(icu.glm2, id.col="red", scale=8, id.cex=1.5, id.n=3)
```

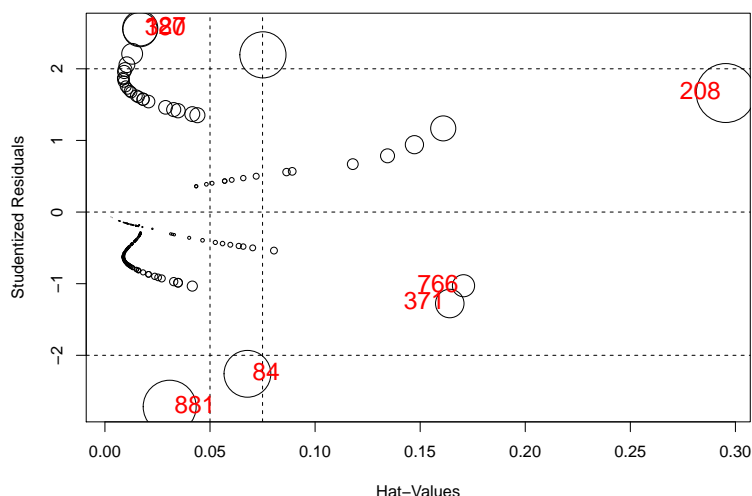


Figure 7.28: Influence plot for the main effects model for the ICU data fig:icu2-inflplot

Details for the cases identified in the figure are shown below, again using `rownames(res)` to select the relevant observations from the ICU data.

```
idx <- which(rownames(ICU) %in% rownames(res))
cbind(ICU[idx,c("died", "age", "cancer", "admit", "uncons")], res)
```

##	died	age	cancer	admit	uncons	StudRes	Hat	CookD
## 84	No	59	No	Emergency	Yes	-2.258	0.06781	0.3626
## 371	No	46	Yes	Emergency	No	-1.277	0.16408	0.2210
## 766	No	31	Yes	Emergency	No	-1.028	0.17062	0.1719
## 881	No	89	No	Emergency	Yes	-2.718	0.03081	0.4106
## 127	Yes	19	No	Emergency	No	2.565	0.01679	0.2724
## 208	Yes	70	No	Elective	Yes	1.662	0.29537	0.4568
## 380	Yes	20	No	Emergency	No	2.548	0.01672	0.2668

None of the cases are particularly influential on the model coefficients overall: the largest Cook's D is only 0.45 for case 208. This observation also has the largest hat value. It is unusual on the predictors in this sample: a 70 year old man without cancer, admitted on an elective basis, who nonetheless died. However, this case is also highly unusual in having been unconscious on admission for an elective procedure, and signals that there might have been a coding error or other anomaly for this observation.

Another noteworthy observation identified here is case 881, an 89 year old male, admitted unconscious as an emergency; this case is poorly predicted because he survived. Similarly, two other cases (127, 380) with large studentized residuals are poorly predicted because they died, although they were young, did not have cancer, and conscious at admission. However, these cases have relatively small Cook's D values. From this evidence we might conclude that, case 208 bears further scrutiny, but none of these cases greatly affects the model, its coefficients, or interpretation.

For comparison with Figure 7.28, the related index plot of these measures is shown in Figure 7.29.

```
influenceIndexPlot(icu.glm2, vars=c("Cook", "Studentized", "hat"), id.n=4)
```

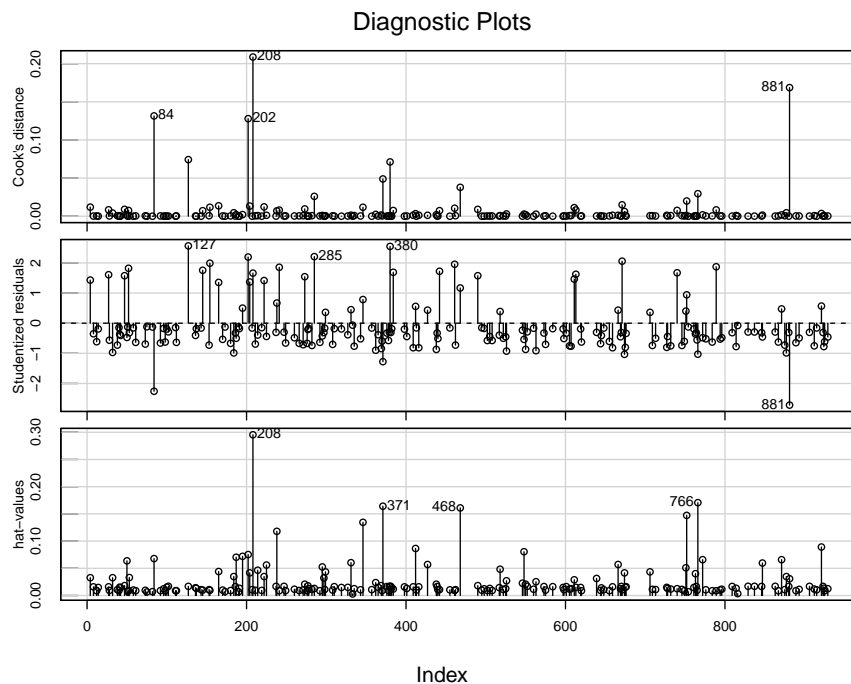


Figure 7.29: Index plots of influence measures for the ICU data model. The four most extreme observations on each measure are labeled. fig:icu2-infl-index

Cook's D and DFFITS are *overall* measures of the total influence that cases have on the regression coefficients and fitted values respectively. It might be that some cases have a large impact on some individual regression coefficients, but don't appear particularly unusual in these aggregate measures.

One way to study this is to make plots of the $DFBETA_{ij}$ statistics. Such plots are not available (as far as we know) in R packages, but it is not hard to construct them from the result returned by `influence.measures()`. To do this, we select the appropriate columns from the `infmtat` component returned by that function.

```
infl <- influence.measures(icu.glm2)
dfbetas <- data.frame(infl$infmtat[,2:5])
colnames(dfbetas) <- c("dfb.age", "dfb.cancer", "dfb.admit", "dfb.uncons")
head(dfbetas)

##      dfb.age dfb.cancer dfb.admit dfb.uncons
## 8  0.047340  0.013418  0.004067  0.009254
## 12 0.018988  0.018412 -0.004174  0.018106
## 14 -0.001051  0.014882  0.026278  0.005555
## 28  0.031562  0.018424 -0.001511  0.016640
## 32 -0.164084  0.003788 -0.036505  0.023488
## 38 -0.021525  0.016539 -0.011937  0.020803
```

To illustrate this idea, plotting an individual column of `dfbetas` using `type = "h"` gives an index plot against the observation number. This is shown in Figure 7.30 for the impact on the coefficient for age. The lines and points are colored blue or red according to whether the patient lived or died. Observations for which the $|DFBETA_{age}| > 0.2$ (an arbitrary value) are labeled.

```
cols=ifelse(ICU$died=="Yes", "red", "blue")
op <- par(mar=c(5,5,1,1)+.1)
plot(dfbetas[,1], type = "h", col=cols,
     xlab="Observation index",
     ylab=expression(Delta * beta[Age]),
     cex.lab=1.3)
points(dfbetas[,1], col=cols)
# label some points
big <- abs(dfbetas[,1]) > .25
idx <- 1:nrow(dfbetas)
text(idx[big], dfbetas[big,1], label=rownames(dfbetas)[big],
     cex=0.9, pos=ifelse(dfbetas[big,1]>0, 3, 1),
     xpd=TRUE)
abline(h=c(-.25, 0, .25), col="gray")
par(op)
```

None of the labeled points here are a cause for concern, since the standardized $DFBETAs$ are all relatively small. However, the plot shows that patients who died have generally larger impacts on this coefficient.

An alternative to individual index plots is a scatterplot matrix, that shows the pairwise changes in the regression coefficients for the various predictors. Here we use `scatterplotMatrix()` from `car` that offers features for additional plot annotations, including identifying the most unusual points in each pairwise plot. In each off-diagonal panel, a 95% data ellipse and linear regression line helps to show the marginal relationship between the two measures and highlight why the labeled points are atypical in each plot.¹⁹

```
scatterplotMatrix(dfbetas, smooth=FALSE, id.n=2,
                 ellipse=TRUE, levels=0.95, robust=FALSE,
                 diagonal="histogram",
                 groups=ICU$died, col=c("blue", "red"))
```

△

¹⁹This plot uses the `id.method="mahal"` method to label the most extreme observations according to the Mahalanobis distance of each point from the centroid in the plot.

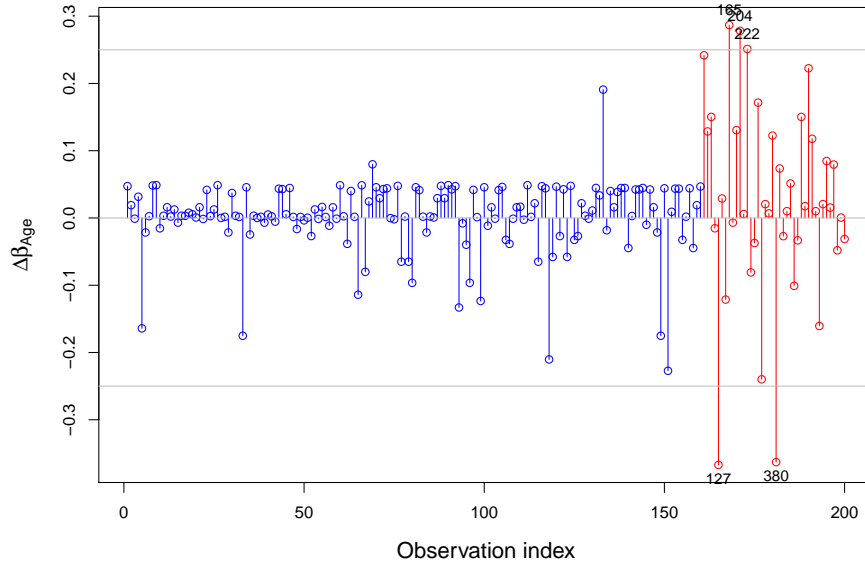


Figure 7.30: Index plot for DFBETA (Age) in the ICU data model. The observations are colored blue or red according to whether the patient lived or died. fig:icu2-dbage

7.5.3 Other diagnostic plots

{sec:logist

The graphical methods described in this section are relatively straight-forward indicators of the adequacy of a particular model, with a specified set of predictors, each expressed in a given way. More sophisticated methods have also been proposed, which focus on the need to include a particular predictor and whether its relationship is linear. These include the **component-plus-residual plot**, the **added-variable plot**, and the **constructed variable plot**, which are all analogous to techniques developed in OLS.

Component-plus-residual plots

onent-plus-residual}

The **component-plus-residual plot** (also called a **partial residual plot**) proposed originally by Larsen and McCleary (1972) is designed to show whether a given quantitative predictor, x_j , included linearly in the model, actually shows a nonlinear relation, requiring transformation. The essential idea is to move the linear term for x_j back into the residual, by calculating the **partial residuals**,

$$r_j^* = r + \beta_j x_j$$

Then, a plot of r_j^* against x_j will have the same slope, β_j , as the full model including it among other predictors. However, any non-linear trend will be shown in the pattern of the points, usually aided by a smoothed non-parametric curve.

As adapted to logistic regression by Landwehr *et al.* (1984), the partial residual for variable x_j is defined as

$$r_j^* = \mathbf{V}^{-1} \mathbf{r} + \beta_j x_j$$

The partial residual plot is then a plot of r_j^* against x_j , possibly with the addition of a smoothed lowess curve (Fowlkes, 1987) and a linear regression line to aid interpretation. The linear regression of the partial residuals on x_j has the same slope, β_j , as in the full model.

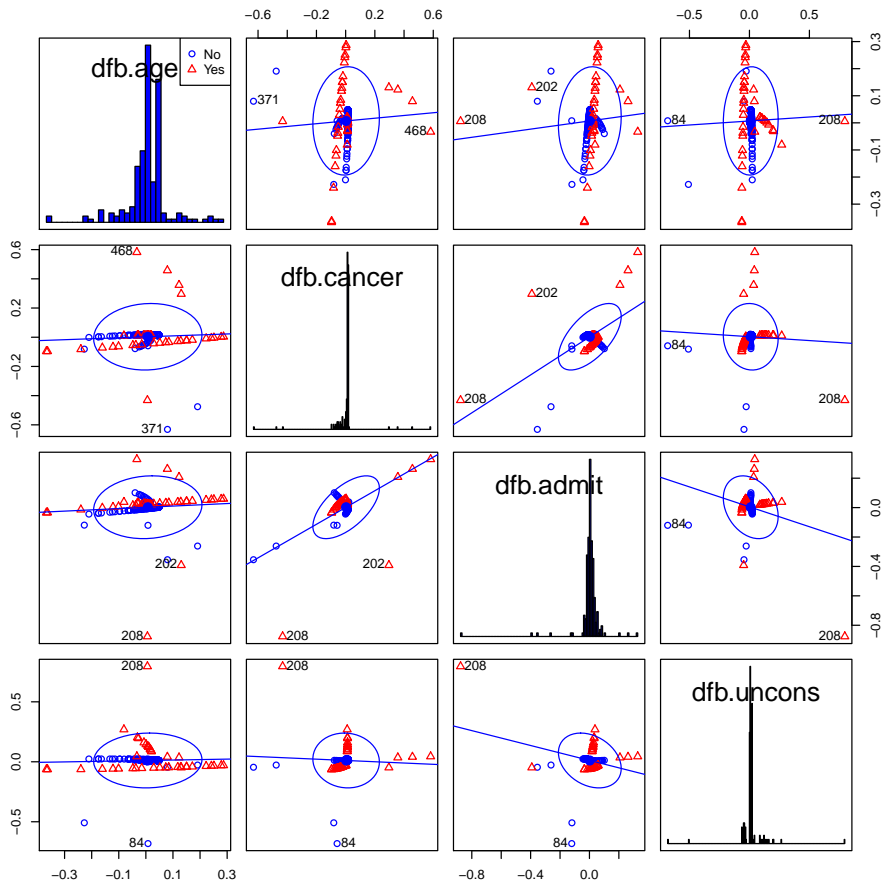


Figure 7.31: Scatterplot matrix for DFBETAs from the model for the ICU data. Those who lived or died are shown with blue circles and red triangles, respectively. The diagonal panels show histograms of each variable.

If x_j affects the binary response linearly, the plot should be approximately linear with a slope approximately equal to β_j . A nonlinear plot suggests that x_j needs to be transformed, and the shape of the relation gives a rough guide to the required transformation. For example, a parabolic shape would suggest a term in x_j^2 . These plots complement the conditional data plots described earlier (Section 7.3.1), and are most useful when there several quantitative predictors, so that it is more convenient and sensible to examine their relationships individually.

The `car` package implements these plots in the `crPlots()` and `crPlot()` functions. They also work for models with factor predictors (using parallel boxplots for the factor levels) but not for those with interaction terms.

```
{ex:donner3}
```

EXAMPLE 7.14: Donner Party

In Example 7.12, we fit several models for the Donner Party data, and we recall two here to illustrate component-plus-residual plots. Both assert additive effects of age and sex, but the model `donner.mod3` allows a quadratic effect of age.

```
donner.mod1 <- glm(survived ~ age + sex, data=Donner, family=binomial)
donner.mod3 <- glm(survived ~ poly(age,2) + sex, data=Donner, family=binomial)
```

Had we not made exploratory plots earlier (Example 7.12), and naively fit only the linear model in age, `donner.mod1`, we could use `crPlots()` to check for a non-linear relationship of survival with age as follows, giving Figure 7.32.

```
crPlots(donner.mod1, ~age, id.n=2)
```

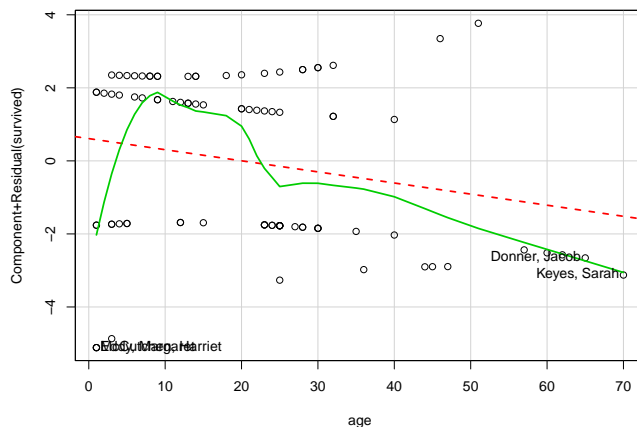


Figure 7.32: Component-plus-residual plot for the simple additive linear model, `donner.mod1`. The dashed red line shows the slope of age in the full model; the smoothed green curve shows a loess fit with `span = 0.5`.
fig:donner-crP

The smoothed loess curve in this plot closely resembles the trend we saw in the conditional plot for age by sex (Figure 7.16), suggesting the need to include a non-linear term for age. The points identified in this plot, by default, are those with either the most extreme x values (giving them high leverage) or the largest absolute Pearson residuals in the full model. The four structured bands of points in the plot correspond to the combinations of sex and survival.

For comparison, you can see the result of allowing for a non-linear relationship in age in a partial residual plot for the model `donner.mod.3` that includes the effect `poly(age, 2)` for age. Note that the syntax of the `crPlots()` function requires that you specify a *term* in the model, rather than just a predictor variable.

```
crPlots(donner.mod3, ~poly(age, 2), id.n=2)
```

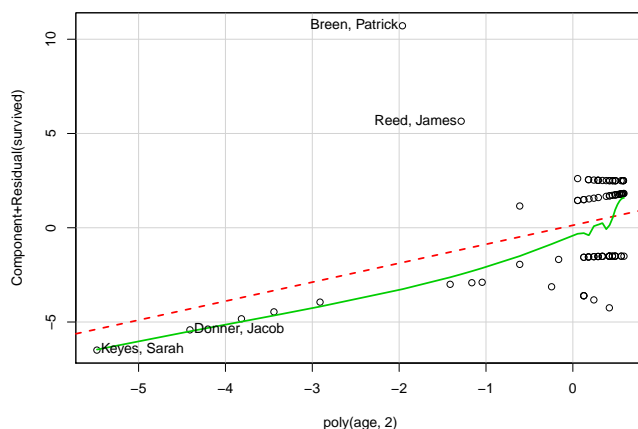


Figure 7.33: Component-plus-residual plot for the non-linear additive model, `donner.mod3` fig:donner-cr2

Except possibly at the extreme right, this plot (Figure 7.33) shows no indication of a non-linear relationship.

△

Added-variable plots

Added-variable plots (Cook and Weisberg, 1999, Wang, 1985) (also called *partial-regression plots*) are another important tool for diagnosing problems in logistic regression and other linear or generalized linear models. These are essentially plots, for each x_i , of an adjusted response, $y_i^* = y \mid \text{others}_i$, against an adjusted predictor, $x_i^* = x_i \mid \text{others}_i$, where $\text{others}_i = \mathbf{X} \setminus x_i \equiv \mathbf{X}^{(-i)}$ indicates all other predictors excluding x_i . As such, they show the *conditional* relationship between the response and the predictor x_i , controlling for, or adjusting for, all other predictors. Here, y_i^* and x_i^* represent respectively the residuals from the regressions of y and x_i on all the other x s excluding x_i .

It might seem from this description that each added-variable plot requires two additional auxiliary logistic regressions to calculate the residuals y_i^* and x_i^* . However, Wang (1985) showed that the added-variable plot may be constructed by following the logistic regression for the model $y \sim \mathbf{X}^{(-i)}$ with one weighted least squares regression of x_i on $\mathbf{X}^{(-i)}$ to find the residual part, x_i^* , of x not predicted by the other regressors.

Let r be the vector of Pearson residuals from the initial logistic fit of y on the variables in $\mathbf{X}^{(-i)}$, and let H and $V = \text{diag}[\hat{p}(1 - \hat{p})]$ be the hat matrix and V matrix from this analysis. Then, the added-variable plot is a scatterplot of the residuals r against the x_i -residuals,

$$x_i^* = (\mathbf{I} - H)V^{1/2}x .$$

There are several important uses of added-variable plots:

First, *marginal* plots of the response variable y against the predictor variables x_i can conceal or misrepresent the relationships in a model including several predictors together due to correlations or associations among the predictors. This problem is compounded by the fact that graphical methods for discrete responses (boxplots, mosaic plots) cannot easily show influential observations or non-linear relationships. Added-variable plots solve this problem by plotting the residuals, $y_i^* = y \mid \text{others}_i$, which are less discrete than the marginal responses in y .

Second, the numerical measures and graphical methods for detecting influential observations described earlier in this section are based on the idea of *single-case deletion*, comparing coefficients or fitted values for the full data, with those that result from deleting each case in turn. Yet, it is well-known (Lawrance, 1995), that sets of two (or more) observations can have *joint influence*, that greatly exceeds their individual influential. Similarly, the influence of one discrepant point can be offset by another influential point in an opposite direction, a phenomenon called *masking*. The main cases of joint influence are illustrated in Figure 7.34. Added-variable plots, showing the partial regression for one predictor controlling all others can make such cases visually apparent.

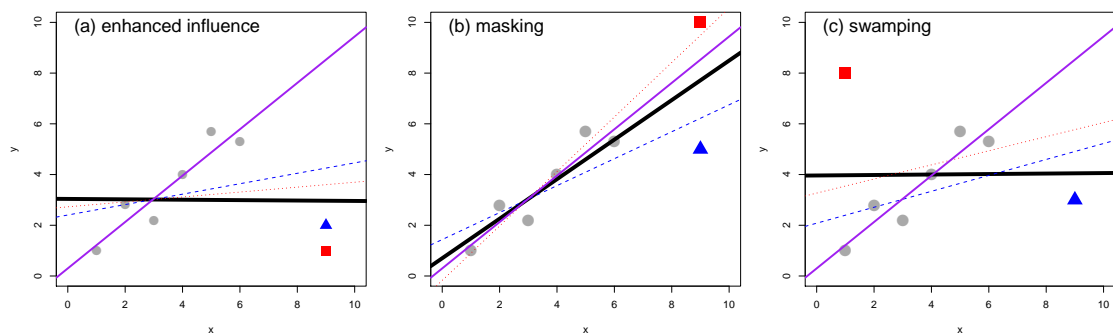


Figure 7.34: Jointly influential points in regression models. In each panel, the thick black line shows the regression of y on x using all the data points. The solid purple line shows the regression deleting *both* the red and blue points and the broken and dotted lines show the regression retaining only the point in its color in addition to the constant gray points. (a) Two points whose joint influence enhance each other; (b) two points where the influence of one is masked by that of the other; (c) two points whose combined influence greatly exceeds the effect of either one individually.

{fig:joint}

Finally, given a tentative model using predictors x , the added-variable plot for another regressor, z can provide a useful visual assessment of its additional contribution. An overall test could be based on the difference in G^2 for the enlarged model $\text{logit}(\mathbf{p}) = \mathbf{X}\boldsymbol{\beta} + \gamma\mathbf{z}$, compared to the reduced model $\text{logit}(\mathbf{p}) = \mathbf{X}\boldsymbol{\beta}$. But the added-variable plot shows whether the evidence for including z is spread throughout the sample or confined to a small subset of observations. The regressor z may be a new explanatory variable, or a higher-order term for variable(s) already in the model.

The `car` package implements these plots with the function `avPlot()` for a single term and `avPlots()` for all terms in a linear or generalized linear model, as shown in the example(s) below. See <http://www.datavis.ca/gallery/animation/duncanAV/> for an animated graphic showing the transition between a marginal plot of the relationship of y to x and the added-variable plot of y^* to x^* for the case of multiple linear regression with a quantitative

```
ex:donner4} response.
```

EXAMPLE 7.15: Donner Party

The simple additive model `donner.mod1` for the Donner Party data can be used to illustrate some features of added-variable plots. In the call to `avPlots()` below, we use color the plotting symbol to distinguish those who survived vs. died, shape to distinguish men from women.

```
col <- ifelse(Donner$survived=="yes", "blue", "red")
pch <- ifelse(Donner$sex=="Male", 16, 17)
avPlots(donner.mod1, id.n=2, col=col, pch=pch, col.lines="darkgreen")
```

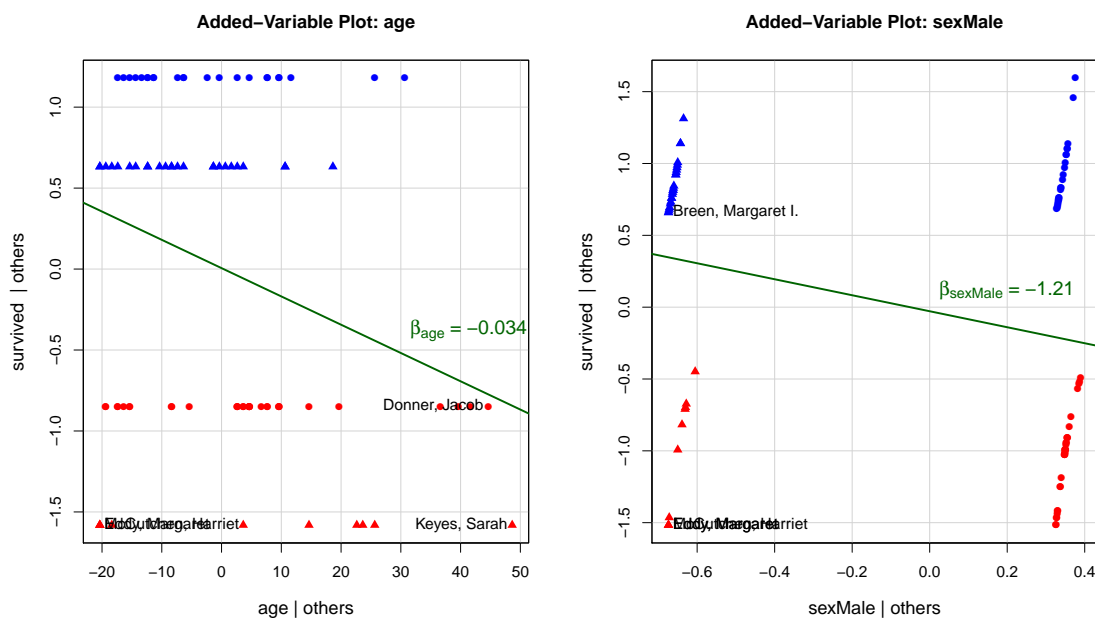


Figure 7.35: Added-variable plots for age (left) and sex (right) in the Donner Party main effects model. Those who survived are shown in blue; those who died in red. Men are plotted with filled circles; women with filled triangles. fig:donner4-avp

These plots have the following properties:

1. The slope in the simple regression of \mathbf{y}_i^* on \mathbf{x}_i^* is the same as the partial coefficient β_i in the full multiple regression model including both predictors here (or all predictors in general).
2. The residuals from this regression line are the same as the residuals in the full model.
3. Because the response, *survived*, is binary, the vertical axis \mathbf{y}_{age}^* in the left panel for *age* is the part of the logit for survival that cannot be predicted from *sex*. Similarly, the vertical axis in the right panel is the part of survival that cannot be predicted from *age*. This property allows the clusters of points corresponding to discrete variables to be seen more readily, particularly if they are distinguished by visual attributes such as color and shape, as in Figure 7.35.

△

{ex:icu3}

EXAMPLE 7.16: Death in the ICU

We illustrate some of the uses of added-variable plots using the main effects model, `icu.glm2`, predicting death in the ICU from the variables `age`, `cancer`, `admit` and `uncons`.

To see why marginal plots of the discrete response against each predictor are often unrevealing for the purpose of model assessment, consider the collection of plots in Figure 7.36 showing the default plots (spineplots) for the factor response, `died` against each predictor. These show the marginal distribution of each predictor by the widths of the bars, and highlight the proportion who died by color. Such plots are useful for some purposes, but not for assessing the adequacy of the fitted model.

```
op <- par(mfrow=c(2,2), mar=c(4,4,1,2.5)+.1, cex.lab=1.4)
plot(died ~ age, data=ICU, col=c("lightblue", "red"))
plot(died ~ cancer, data=ICU, col=c("lightblue", "red"))
plot(died ~ admit, data=ICU, col=c("lightblue", "red"))
plot(died ~ uncons, data=ICU, col=c("lightblue", "red"))
par(op)
```

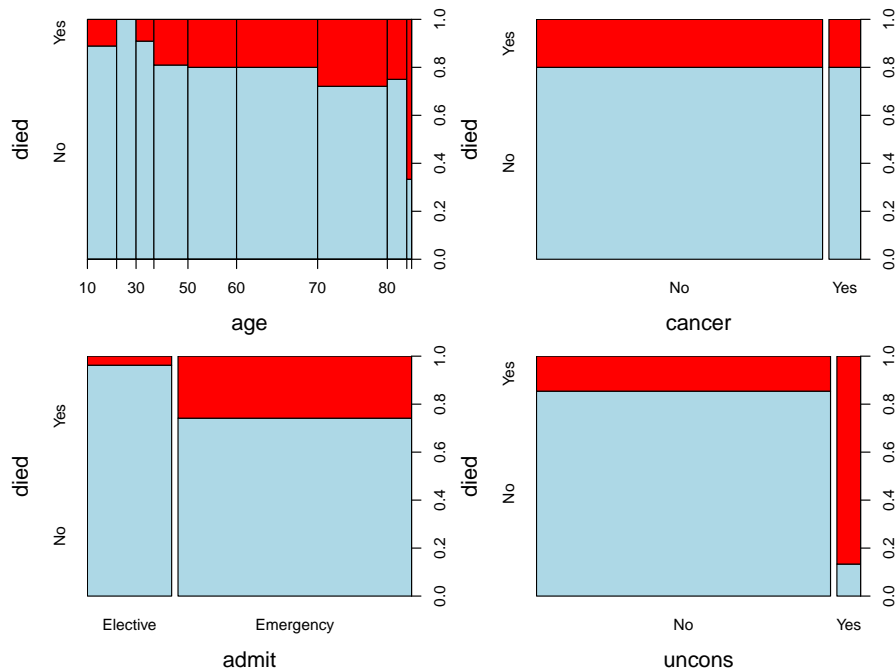


Figure 7.36: Marginal plots of the response `died` against each of the predictors in the model `icu.glm2` for the ICU data. Fig:icu3-marginal

The added-variable plot for this model is shown in Figure 7.37. In each plot, the solid red line shows the partial slope, β_j for the focal predictor, controlling for all others.

```
pch <- ifelse(ICU$died=="No", 1, 2)
avPlots(icu.glm2, id.n=2, pch=pch, cex.lab=1.3)
```

The labeled points in each panel use the default `id.method` for `avPlots()`, selecting those with either large absolute model residuals or extreme x_i^* residuals, given all other predictors. Cases 127 and 881, identified earlier as influential stand out in all these plots.

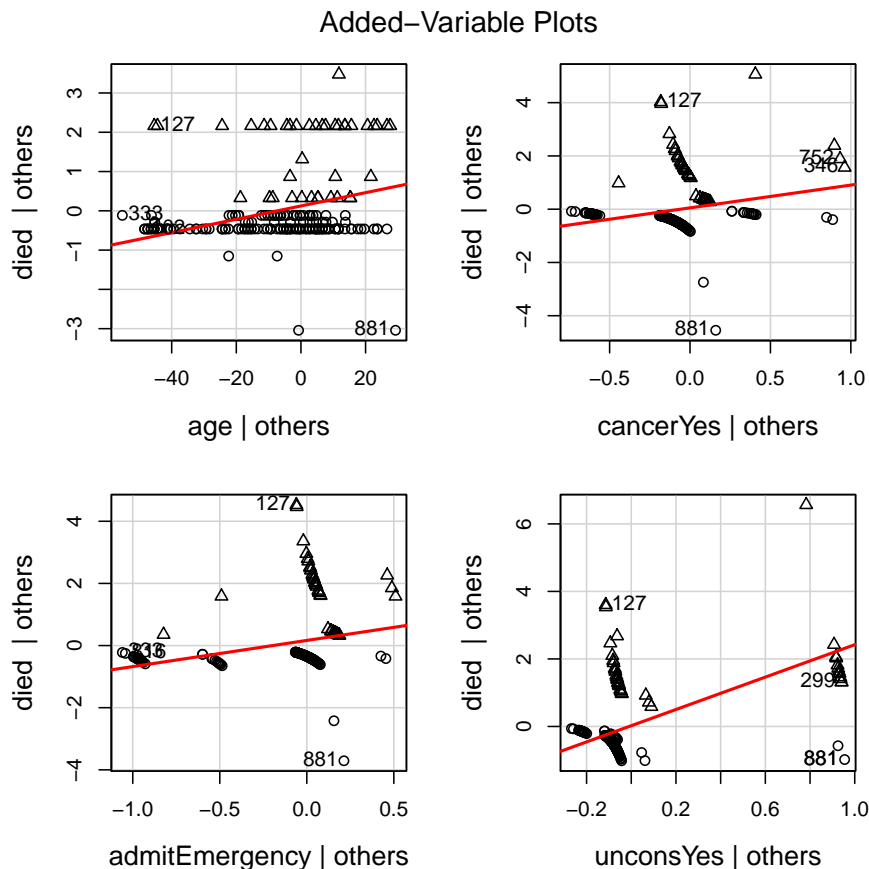


Figure 7.37: Added-variable plots for the predictors in the model for the ICU data. Those who died and survived are shown by triangles (Δ) and circles (\circ) respectively. fig:icub-avpl

Next, we illustrate the use of added-variable plots for checking the effect of influential observations on the decision to include an additional predictor in some given model. In the analysis of the ICU data using model selection methods, the variable *systolic* (systolic blood pressure at admission) was nominated by several different procedures. Here we take a closer look at the evidence for inclusion of this variable in a predictive model. We fit a new model adding *systolic* to the others and test the improvement with a likelihood ratio test:

```
icu.glm2a <- glm(died ~ age + cancer + admit + uncons + systolic,
                data=ICU, family=binomial)
anova(icu.glm2, icu.glm2a, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: died ~ age + cancer + admit + uncons
## Model 2: died ~ age + cancer + admit + uncons + systolic
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      195      139
## 2      194      136 1      3.52    0.061 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, the addition of systolic blood pressure is nearly significant at the conventional $\alpha = 0.05$ level. The added-variable plot for this variable in Figure 7.38 shows the strength of evidence for

its contribution, above and beyond the other variables in the model, as well as the partial leverage and influence of individual points.

```
avPlot(icu.glm2a, "systolic", id.n=3, pch=pch)
```

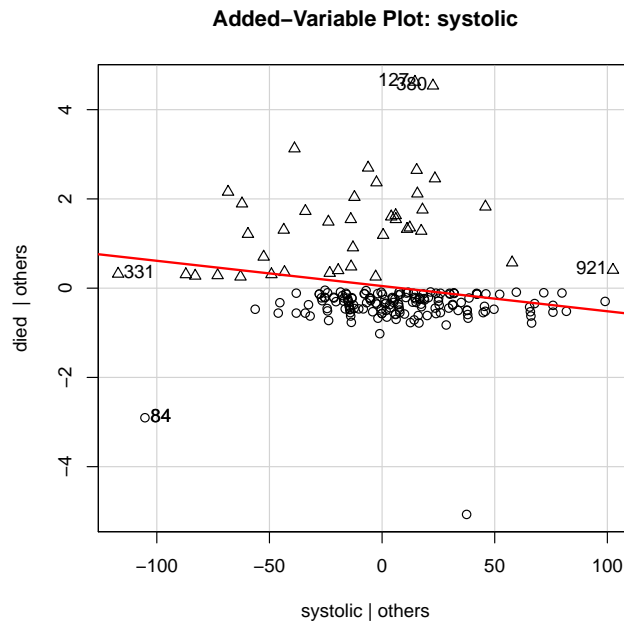


Figure 7.38: added-variable plot for the effect of adding systolic blood pressure to the main effects model for the ICU data. Fig: icu3-avp2

In this plot, cases 331 and 921 have high partial leverage, but they are not influential. Case 84, however, has high leverage and a large residual, so is possibly influential on the evidence for inclusion of *systolic* in the model. Note also that the partial regression line in this plot nicely separates nearly all the patients who died from those who survived.

△

TODO: This chapter is quite long. Could start a new chapter here, but that would require re-numbering file names, or introduce Ch07a.

7.6 Polytomous response models

{sec:logist-poly}

Polytomous response data arise when the outcome variable, Y , takes on $m > 2$ discrete values. For example, (a) patients may record that their improvement after treatment is “none,” “some” or “marked;” (b) high school students may choose a general, vocational or academic program; (c) women’s labor force participation may be recorded in a survey as not working outside the home, working part-time, or working full-time; (d) Canadian voters may express a preference for the Conservative, Liberal, NDP, Green party. These response categories may be considered ordered or simply nominal.

In this situation, there are several different ways to model the response probabilities. Let $\pi_{ij} \equiv \pi_j(\mathbf{x}_i)$ be the probability of response j for case or group i , given the predictors \mathbf{x}_i .

Because $\sum_j \pi_{ij} = 1$, only $m - 1$ of these probabilities are independent. The essential idea here is to construct a model for the polytomous (or multinomial) response composed of $m - 1$ logit comparisons among the response categories in a similar way to how factors are treated in the predictor variables.

The simplest approach uses the *proportional odds model*, described in Section 7.6.1. This model applies only when the response is ordinal (as in improvement after therapy) *and* an additional assumption (the proportional odds assumption) holds. This model can be fit using `polr()` in the MASS package, `lrm()` in the rms package, and `vglm()` in VGAM.

However, if the response is purely nominal (e.g., vote Conservative, Liberal, NDP, Green), or if the proportional odds assumption is untenable, another particularly simple strategy is to fit separate models to a set of $m - 1$ *nested dichotomies* derived from the polytomous response (described in Section 7.6.3). This method allows you to resolve the differences among the m response categories into independent statistical questions (similar to orthogonal contrasts in ANOVA). For example, for women's labor force participation, it might be substantively interesting to contrast not working vs. (part-time and full-time) and then part-time vs. full-time for women who are working. You fit such nested dichotomies by running the $m - 1$ binary logit models and combining the statistical results.

The most general approach is the *generalized logit model*, also called the *multinomial logit model*. This model fits *simultaneously* the $m - 1$ simple logit models against a baseline or reference category, for example, the last category, m . With a 3-category response, there are two generalized logits, $L_{i1} = \log(\pi_{i1}/\pi_{i3})$ and $L_{i2} = \log(\pi_{i2}/\pi_{i3})$, contrasting response categories 1 and 2 against category 3. In this approach, it doesn't matter which response category is chosen as the baseline, because all pairwise comparisons can be recovered from whatever is estimated. This model is conveniently fitted using `multinom()` in nnet.

7.6.1 Ordinal Response: Proportional Odds Model

{sec:ordinal}

For an ordered response Y , with categories $j = 1, 2, \dots, m$, the ordinal nature of the response can be taken into account by forming logits based on the $m - 1$ adjacent category cutpoints between successive categories. That is, if the cumulative probabilities are

$$\Pr(Y \leq j | \mathbf{x}) = \pi_1(\mathbf{x}) + \pi_2(\mathbf{x}) + \dots + \pi_j(\mathbf{x}) ,$$

then the *cumulative logit* for category j is defined as

$$L_j \equiv \text{logit}[\Pr(Y \leq j | \mathbf{x})] = \log \frac{\Pr(Y \leq j | \mathbf{x})}{\Pr(Y > j | \mathbf{x})} = \log \frac{\Pr(Y \leq j | \mathbf{x})}{1 - \Pr(Y \leq j | \mathbf{x})} \quad (7.13) \quad \{\text{eq:cumlogit}\}$$

for $j = 1, 2, \dots, m - 1$.

In our running example of responses to arthritis treatment, the actual response variable is *Improved*, with ordered levels "None" < "Some" < "Marked". In this case, the cumulative logits would be defined as

$$L_1 = \log \frac{\pi_1(\mathbf{x})}{\pi_2(\mathbf{x}) + \pi_3(\mathbf{x})} = \text{logit}(\text{None vs. [Some or Marked]})$$

$$L_2 = \log \frac{\pi_1(\mathbf{x}) + \pi_2(\mathbf{x})}{\pi_3(\mathbf{x})} = \text{logit}(\text{[None or Some] vs. Marked}) ,$$

where \mathbf{x} represents the predictors (sex, treatment and age).

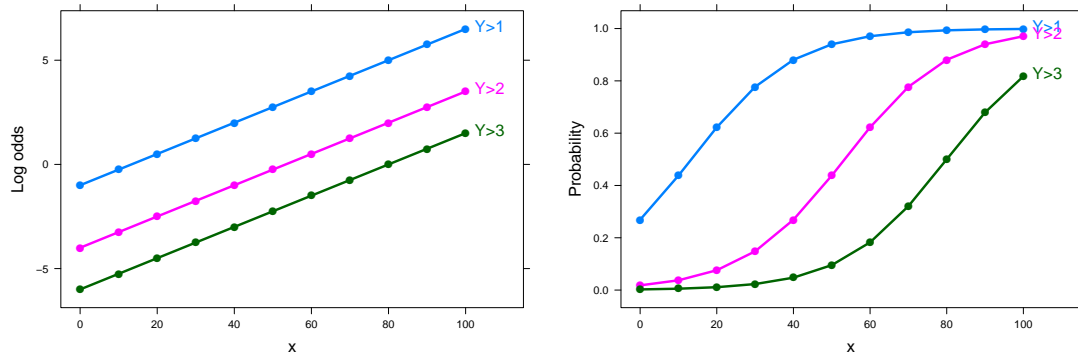


Figure 7.39: Proportional odds model for an ordinal response. The model assumes equal slopes for the cumulative response logits. Left: logit scale; right: probability scale.

{fig:podds}

The *proportional odds model* (PO) (McCullagh, 1980) proposes a simple and parsimonious account of these effects, where the predictors in (x) are constrained to have the same slopes for all cumulative logits,

$$L_j = \alpha_j + \mathbf{x}^T \boldsymbol{\beta} \quad j = 1, \dots, m - 1 . \quad (7.14)$$

{eq:propodds}

That is, the effect of the predictor x_i is the same, β_i , for all cumulative logits. The cumulative logits differ only in their intercepts. In this formulation, the $\{\alpha_j\}$ increase with j , because $\Pr(Y \leq j | \mathbf{x})$ increases with j for fixed \mathbf{x} .²⁰ Figure 7.39 portrays the PO model for a single quantitative predictor x with $m = 4$ response categories.

The name “proportional odds” stems from the fact that under Eqn. (7.14), for fixed \mathbf{x} , the cumulative log odds (logits) for categories j and j' are constant, $(\alpha_j - \alpha_{j'})$, so the odds, $\exp(\alpha_j - \alpha_{j'})$ have a constant ratio, or are proportional. Similarly, the ratio of the cumulative odds of making a response $Y \leq j$ at values of the predictors $\mathbf{x} = \mathbf{x}_1$ are $\exp((\mathbf{x}_1 - \mathbf{x}_2)^T \boldsymbol{\beta})$ times the odds of this response at $\mathbf{x} = \mathbf{x}_2$, so the log cumulative odds ratio is proportional to the difference between \mathbf{x}_1 and \mathbf{x}_2 .

Latent variable interpretation

For a binary response, an alternative motivation for logistic regression regards the relation of the observed Y as arising from a continuous, unobserved, (latent) response variable, ξ representing the propensity for a “success” (1) rather than “failure” (0). The latent response is assumed to be linearly related to the predictors \mathbf{x} according to

$$\xi_i = \alpha + \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i = \alpha + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \epsilon_i \quad (7.15)$$

{eq:latent}

However, we can only observe $Y_i = 1$ when ξ_i passes some threshold, that with some convenient scaling can be taken as $\xi_i > 0 \implies Y_i = 1$.²¹

²⁰Some authors and some software describe the PO model in terms of $\text{logit}[\Pr(Y > j | \mathbf{x})]$, so the signs and order of the intercepts, α_j are reversed.

²¹The latent variable derivation of logistic regression (and the related probit model) was fundamental in the history of statistical methods for discrete response outcomes. An early example was Thurstone’s (1927) *Law of comparative judgment* designed to account for psychological preference by assuming an underlying latent continuum of “hedonic

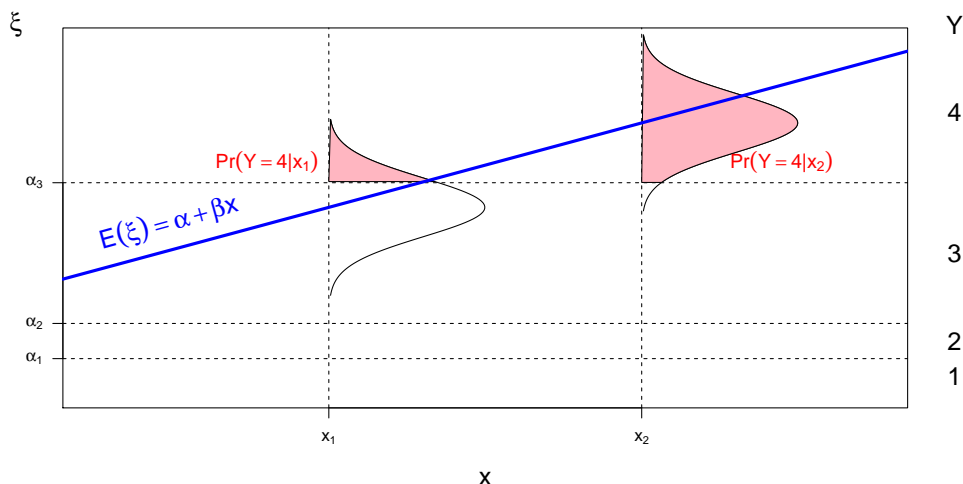


Figure 7.40: Latent variable representation of the proportional odds model for $m = 4$ response categories and a single quantitative predictor, x . *Source:* Adapted from Fox (2008, Fig 14.10), using code provided by John Fox.

{fig:latent}

The latent variable motivation extends directly to an ordinal response under the PO model. We now assume that there is a set of $m - 1$ thresholds, $\alpha_1 < \alpha_2 < \dots < \alpha_{m-1}$ for the latent variable ξ_i in Eqn. (7.15) and we observe

$$Y_i = j \quad \text{if} \quad \alpha_{j-1} < \xi_i \leq \alpha_j ,$$

with appropriate modifications to the inequalities at the end points.

This is illustrated in Figure 7.40 for a response with $m = 4$ ordered categories and a single quantitative predictor, x . The observable response Y categories are shown on the right vertical axis, and the corresponding latent continuous variable ξ on the left axis together with the thresholds $\alpha_1, \alpha_2, \alpha_3$. The (conditional) logistic distribution of ξ is shown at two values of x , and the shaded areas under the curve give the conditional probabilities $\Pr(Y = 4 | x_i)$ for the two values x_1 and x_2 .

Fitting the proportional odds model

As mentioned earlier, there are a number of different R packages that provide facilities for fitting the PO model. These have somewhat different capabilities for reporting results, testing hypotheses and plotting, so we generally use `polr()` in the MASS package, except where other packages offer greater convenience.

Unless the response variable has numeric values, it is important to ensure that it has been defined as an *ordered* factor (using `ordered()`). In the Arthritis data, the response, *Improved* was setup this way, as we can check by printing some of the values.²²

values.” Similarly, the probit model arose from dose-response studies in toxicology (Bliss, 1934, Finney, 1947) where the number killed by some chemical agent was related to its’ type, dose or concentration. The idea of a latent variable was also at the heart of the development of factor analysis **TODO: citation?** and latent class analysis (Lazarsfeld, 1950, 1954) was developed to treat the problem of classifying individuals into discrete latent classes from fallible measurements. See Bollen (2002) for a useful overview of latent variable models in the social sciences.

²²As an unordered factor, the levels would be treated as ordered alphabetically, i.e., Marked, None, Some.

```
head(Arthritis$Improved, 8)

## [1] Some   None   None   Marked Marked Marked None   Marked
## Levels: None < Some < Marked
```

We fit the main effects model for the ordinal response using `polr()` as shown below. We also specify `Hess=TRUE` to have the function return the observed information matrix (called the Hessian), that is used in other operations to calculate standard errors.

```
arth.polr <- polr(Improved ~ Sex + Treatment + Age,
                 data=Arthritis, Hess=TRUE)
summary(arth.polr)

## Call:
## polr(formula = Improved ~ Sex + Treatment + Age, data = Arthritis,
##       Hess = TRUE)
##
## Coefficients:
##              Value Std. Error t value
## SexMale        -1.2517    0.5464  -2.29
## TreatmentTreated  1.7453    0.4759   3.67
## Age              0.0382    0.0184   2.07
##
## Intercepts:
##              Value Std. Error t value
## None|Some      2.532    1.057    2.395
## Some|Marked    3.431    1.091    3.144
##
## Residual Deviance: 145.46
## AIC: 155.46
```

The output from the `summary()` method, shown above, gives the estimated coefficients (β) and intercepts (α_j) labeled by the cutpoint on the ordinal response. It provides standard errors and t -values ($\beta_i/SE(\beta_i)$), but no significance tests or p -values.

```
library(car)
Anova(arth.polr)

## Analysis of Deviance Table (Type II tests)
##
## Response: Improved
##              LR Chisq Df Pr(>Chisq)
## Sex              5.69  1  0.01708 *
## Treatment       14.71  1  0.00013 ***
## Age              4.57  1  0.03251 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing the proportional odds assumption

The simplicity of the PO model is achieved only when the proportional odds model holds for a given data set. In essence, a test of this assumption involves a contrast between the PO model and a generalized logit NPO model that allows different effects (slopes) of the predictors across the response categories:

$$\text{(eq:po)} \quad \text{PO: } L_j = \alpha_j + \mathbf{x}^T \boldsymbol{\beta} \quad j = 1, \dots, m-1 \quad (7.16)$$

$$\text{(eq:npo)} \quad \text{NPO: } L_j = \alpha_j + \mathbf{x}^T \boldsymbol{\beta}_j \quad j = 1, \dots, m-1 \quad (7.17)$$

The most general test involves fitting both models and testing the difference in the residual deviance by a likelihood ratio test or using some other measure (such as AIC) for model comparison. The PO model (Eqn. (7.16)) has $(m-1)+p$ parameters, while the NPO model (Eqn. (7.17)) has $(m-1)(1+p) = m(1+p)$ parameters, which may be difficult to fit if this is large relative to the number of observations. An intermediate model, the *partial proportional odds model* (Peterson and Harrell, 1990) allows one subset of predictors, \mathbf{x}_{po} , to satisfy the proportional odds assumption (equal slopes), while the remaining predictors \mathbf{x}_{npo} have slopes varying with the response level:

$$\text{PPO: } L_j = \alpha_j + \mathbf{x}_{po}^T \boldsymbol{\beta} + \mathbf{x}_{npo}^T \boldsymbol{\beta}_j \quad j = 1, \dots, m-1. \quad (7.18) \quad \{\text{eq:ppo}\}$$

In R, the PO and NPO models can be readily contrasted by fitting them both using `vglm()` in the VGAM package. This defines the cumulative family of models and allows a parallel option. With `parallel=TRUE`, this is equivalent to the `polr()` model, except that the signs of the coefficients are reversed.

```
library(VGAM)
arth.po <- vglm(Improved ~ Sex + Treatment + Age, data=Arthritis,
              family = cumulative(parallel=TRUE))
arth.po

## Call:
## vglm(formula = Improved ~ Sex + Treatment + Age, family = cumulative(parallel = TRUE),
##      data = Arthritis)
##
## Coefficients:
##      (Intercept):1      (Intercept):2      SexMale
##      2.531990      3.430988      1.251671
## TreatmentTreated      Age
##      -1.745304      -0.038163
##
## Degrees of Freedom: 168 Total; 163 Residual
## Residual deviance: 145.46
## Log-likelihood: -72.729
```

The more general NPO model can be fit using `parallel=FALSE`.

```
arth.npo <- vglm(Improved ~ Sex + Treatment + Age, data=Arthritis,
               family = cumulative(parallel=FALSE))
arth.npo

## Call:
## vglm(formula = Improved ~ Sex + Treatment + Age, family = cumulative(parallel = FALSE),
##      data = Arthritis)
##
## Coefficients:
##      (Intercept):1      (Intercept):2      SexMale:1
##      2.618539      3.431175      1.509827
##      SexMale:2 TreatmentTreated:1 TreatmentTreated:2
##      0.866434      -1.836929      -1.704011
##      Age:1      Age:2
##      -0.040866      -0.037294
##
## Degrees of Freedom: 168 Total; 160 Residual
## Residual deviance: 143.57
## Log-likelihood: -71.787
```

The VGAM package defines a `coef()` method that can print the coefficients in a more readable matrix form giving the category cutpoints:

```
coef(arth.po, matrix=TRUE)

##           logit (P[Y<=1])  logit (P[Y<=2])
## (Intercept)          2.531990          3.430988
## SexMale              1.251671          1.251671
## TreatmentTreated    -1.745304         -1.745304
## Age                  -0.038163         -0.038163

coef(arth.npo, matrix=TRUE)

##           logit (P[Y<=1])  logit (P[Y<=2])
## (Intercept)          2.618539          3.431175
## SexMale              1.509827          0.866434
## TreatmentTreated    -1.836929         -1.704011
## Age                  -0.040866         -0.037294
```

In most cases, nested models can be tested using an `anova()` method, but the VGAM package has not implemented this for "vglm" objects. Instead, it provides an analogous function, `lrtest()`:

```
VGAM::lrtest(arth.npo, arth.po)

## Likelihood ratio test
##
## Model 1: Improved ~ Sex + Treatment + Age
## Model 2: Improved ~ Sex + Treatment + Age
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1 160  -71.8
## 2 163  -72.7  3  1.88      0.6
```

The LR test can be also calculated as “manually” shown below using the difference in residual deviance for the two models.

```
tab <- cbind(
  Deviance = c(deviance(arth.npo), deviance(arth.po)),
  df = c(df.residual(arth.npo), df.residual(arth.po))
)
tab <- rbind(tab, diff(tab))
rownames(tab) <- c("GenLogit", "PropOdds", "LR test")
tab <- cbind(tab, pvalue=1-pchisq(tab[,1], tab[,2]))
tab

##           Deviance  df  pvalue
## GenLogit 143.5741 160 0.81966
## PropOdds 145.4579 163 0.83435
## LR test   1.8838   3 0.59686
```

The `vglm()` can also fit partial proportional odds models, by specifying a formula giving the terms for which the PO assumption should be taken as TRUE or FALSE. Here we illustrate this using `parallel=FALSE ~ Sex`, to fit separate slopes for males and females, but parallel lines for the other predictors. The same model would be fit using `parallel=TRUE ~ Treatment + Age`.

```
arth.ppo <- vglm(Improved ~ Sex + Treatment + Age, data=Arthritis,
  family = cumulative(parallel=FALSE ~ Sex))
coef(arth.ppo, matrix=TRUE)

##                logit (P[Y<=1])  logit (P[Y<=2])
## (Intercept)          2.542452          3.615561
## SexMale              1.483336          0.867362
## TreatmentTreated    -1.775742         -1.775742
## Age                 -0.039622         -0.039622
```

Graphical assessment of proportional odds

There are several graphical methods for visual assessment of the proportional odds assumption. These are all *marginal* methods, in that they treat the predictors one at a time. However, that provides one means to determine if a partial proportional odds model might be more appropriate. Harrell's 2001, Ch. 13-14 *Regression Modeling Strategies* and the corresponding rms package provide an authoritative treatment and methods in R.

One simple idea is to plot the conditional mean or expected value $E(X | Y)$ of a given predictor, X , at each level of the ordered response Y . If the response behaves ordinally in relation to X , these means should be strictly increasing or decreasing with Y . For comparison, one can also plot the estimated conditional means $\hat{E}(X | Y = j)$ under the fitted PO model X as the only predictor. If the PO assumption holds for this X , the model-mean curve should be close to the data mean curve.

```
library(rms)
arth.po2 <- lrm(Improved ~ Sex + Treatment + Age, data=Arthritis)
arth.po2

##
## Logistic Regression Model
##
## lrm(formula = Improved ~ Sex + Treatment + Age, data = Arthritis)
##
##                Model Likelihood      Discrimination      Rank Discrim.
##                Ratio Test           Indexes           Indexes
## Obs           84  LR chi2          24.46  R2           0.291  C           0.750
## None          42  d.f.              3      g           1.335  Dxy          0.500
## Some          14  Pr(> chi2) <0.0001  gr          3.801  gamma        0.503
## Marked        28  gp           0.280  tau-a         0.309
## max |deriv| 1e-07  Brier          0.187
##
##                Coef      S.E.   Wald Z  Pr(>|Z|)
## y>=Some        -2.5320  1.0570  -2.40  0.0166
## y>=Marked      -3.4310  1.0911  -3.14  0.0017
## Sex=Male       -1.2517  0.5464  -2.29  0.0220
## Treatment=Treated  1.7453  0.4759  3.67  0.0002
## Age            0.0382  0.0184  2.07  0.0382
```

The plot of conditional X means is produced using the `plot.xmean.ordinaly()` as shown below. It produces one marginal panel for each predictor in the model. For categorical predictors, it plots only the overall most frequent category. The resulting plot is shown in Figure 7.41.

```
op <- par(mfrow=c(1, 3))
plot.xmean.ordinaly(Improved ~ Sex + Treatment + Age, data=Arthritis,
                    lwd=2, pch=16, subn=FALSE)
par(op)
```

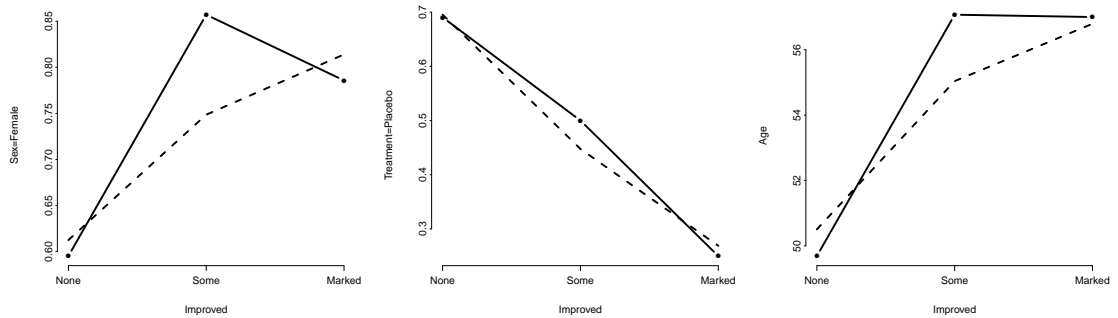


Figure 7.41: Visual assessment ordinality and the proportional odds assumption for predictors in the Arthritis data. Solid lines connect the stratified means of X given Y . Dashed lines show the estimated expected value of X given $Y=j$ if the proportional odds model holds for X . fig:arth-rmsplot

In Figure 7.41, there is some evidence that the effect of *Sex* is non-monotonic and the means differ from their model-implied values under the PO assumption. The effect of *Treatment* looks good by this method, and the effect of *Age* hints that the upper two categories may not be well-distinguished as an ordinal response.

Of course, this example has only a modest total sample size, and this method only examines the marginal effects of the predictors. Nevertheless, it is a useful supplement to the statistical methods described earlier.

7.6.2 Visualizing results for the proportional odds model

{sec:vis-propodds}

Results from the PO model (and other models for polytomous responses) can be graphed using the same ideas and methods shown earlier for a binary or binomial response. In particular, full-model plots (described earlier in Section 7.3.2) and effect plots (Section 7.3.3) are still very helpful.

But now there is the additional complication that the response variable has $m > 2$ levels and so needs to be represented by $m - 1$ curves or panels in addition to those related to the predictor variables.

Full-model plots

{sec:po-fullplots}

For full-model plots, we continue the idea of appending the fitted response probabilities (or logits) to the data frame and plotting these in relation to the predictors. The `predict()` method returns the highest probability category label by default (with `type="class"`), so to get the fitted probabilities you have to ask for `type="probs"`, as shown below.

```
arth.fitp <- cbind(Arthritis,
                  predict(arth.polr, type="probs"))
head(arth.fitp)
```



```
##   ID Treatment Sex Age Improved Better None Some Marked
## 1  57   Treated Male 27   Some      1 0.73262 0.13806 0.12932
## 2  46   Treated Male 29   None      0 0.71740 0.14443 0.13816
## 3  77   Treated Male 30   None      0 0.70960 0.14763 0.14277
## 4  17   Treated Male 32   Marked     1 0.69363 0.15400 0.15237
## 5  36   Treated Male 46   Marked     1 0.57025 0.19504 0.23471
## 6  23   Treated Male 58   Marked     1 0.45634 0.21713 0.32653
```

For plotting, it is most convenient to reshape these from wide to long format using `melt()` in the `reshape2` package. The response category is named `Level`.

```
library(reshape2)
plotdat <- melt(arth.fitp,
               id.vars = c("Sex", "Treatment", "Age", "Improved"),
               measure.vars=c("None", "Some", "Marked"),
               variable.name = "Level",
               value.name = "Probability")
## view first few rows
head(plotdat)

##   Sex Treatment Age Improved Level Probability
## 1 Male   Treated  27     Some  None    0.73262
## 2 Male   Treated  29     None  None    0.71740
## 3 Male   Treated  30     None  None    0.70960
## 4 Male   Treated  32   Marked  None    0.69363
## 5 Male   Treated  46   Marked  None    0.57025
## 6 Male   Treated  58   Marked  None    0.45634
```

We can now plot *Probability* against *Age*, using *Level* to assign different colors to the lines for the response categories. `facet_grid()` is used to split the plot into separate panels by *Sex* and *Treatment*. In this example, the `directlabels` package is also used replace the default legend created by `ggplot()` with category labels on the curves themselves, which is easier to read.

```
library(ggplot2)
library(directlabels)
gg <- ggplot(plotdat, aes(x = Age, y = Probability, colour = Level)) +
  geom_line(size=2.5) + theme_bw() + xlim(10,80) +
  geom_point(color="black", size=1.5) +
  facet_grid(Sex ~ Treatment,
            labeller = function(x, y) sprintf("%s = %s", x, y)
            )
direct.label(gg)
```

Although we now have three response curves in each panel, this plot is relatively easy to understand: (a) In each panel, the probability of no improvement decreases with age, while that for marked improvement increases. (b) It is easy to compare the placebo and treated groups in each row, showing that no improvement decreases, while marked improvement increases with the active treatment. (On the other hand, this layout makes it harder to compare panels vertically for males and females in each condition.) (c) The points show where the observations are located in each panel; so, we can see that the data is quite thin for males given the placebo.²³

²³One way to improve (pun intended) this graph would be to show the points on the lines only for the actual level of *Improve* for each observation.

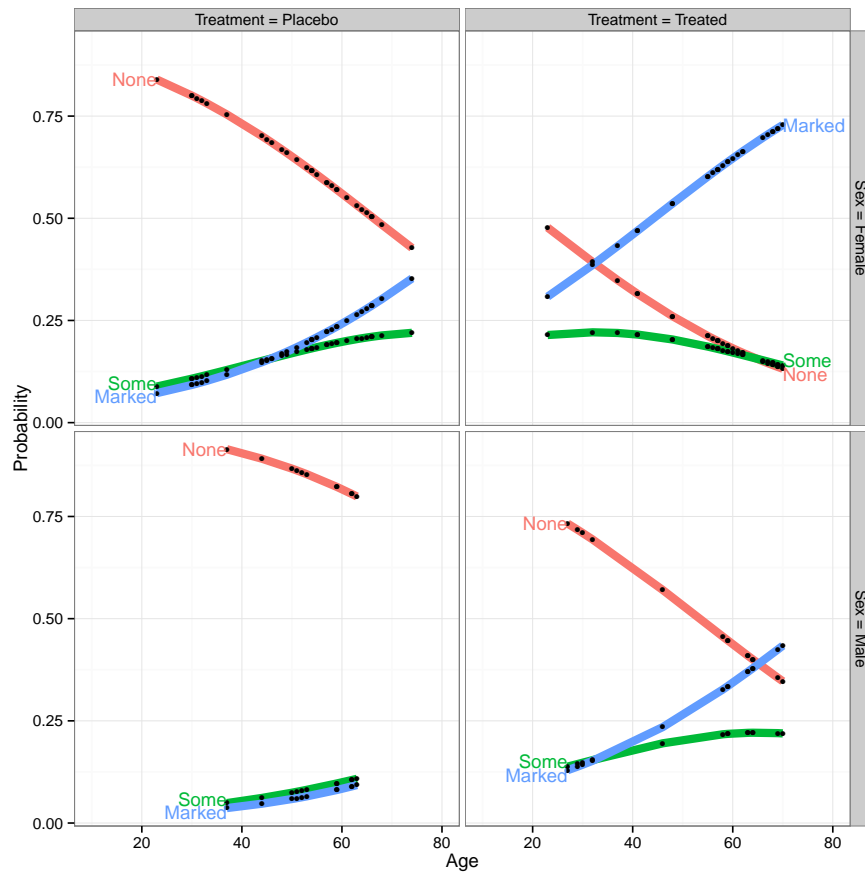


Figure 7.42: Predicted probabilities for the proportional odds model fit to the Arthritis data^{fig:arth-polr1}

Effect plots

{sec:po-efplots}

For PO models fit using `polr()`, the `effects` package provides two different styles for plotting a given effect. By default, curves are plotted in separate panels for the different response levels of a given effect, together with confidence bands for predicted probabilities. This form provides confidence bands and rug plots for the observations, but the default vertical arrangement of the panels makes it harder to compare the trends for the different response levels. The alternative *stacked* format shows the changes in response level more directly, but doesn't provide confidence bands.

Figure 7.43 shows these two styles for the main effect of *Age* in the proportional odds model, `arth.polr` fit earlier.

```
plot(Effect("Age", arth.polr))
plot(Effect("Age", arth.polr), style='stacked',
     key.args=list(x=.55, y=.9))
```

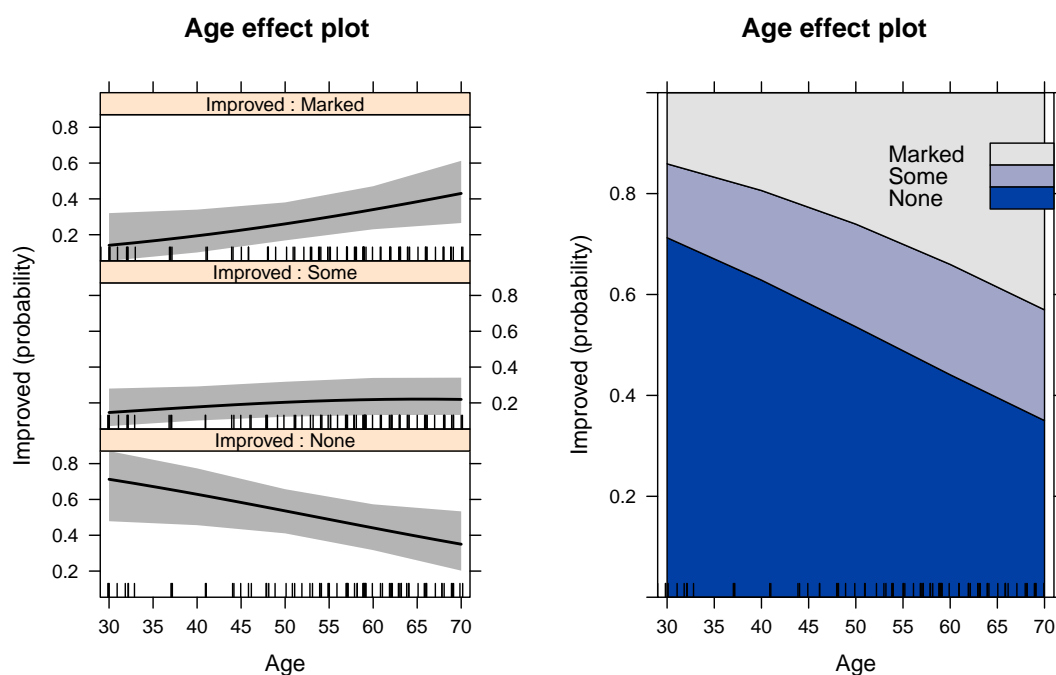


Figure 7.43: Effect plots for the effect of *Age* in the proportional odds model for the Arthritis data. Left: responses shown in separate panels. Right: responses shown in stacked format fig:arth-po-eff1

Even though this model includes only main effects, you can still plot the higher-order effects for more focal predictors in a coherent display. Figure 7.44 shows the predicted probabilities for all three predictors together. Again, visual comparison is easier horizontally for placebo versus treated groups, but you can also see that the prevalence of marked improvement is greater for females than for males.

```
plot(Effect(c("Treatment", "Sex", "Age"), arth.polr),
     style="stacked", key.arg=list(x=.8, y=.9))
```

Finally, the latent variable interpretation of the PO model provides for simpler plots on the logit scale. Figure 7.45 shows this plot for the effects of *Treatment* and *Age* (collapsed over

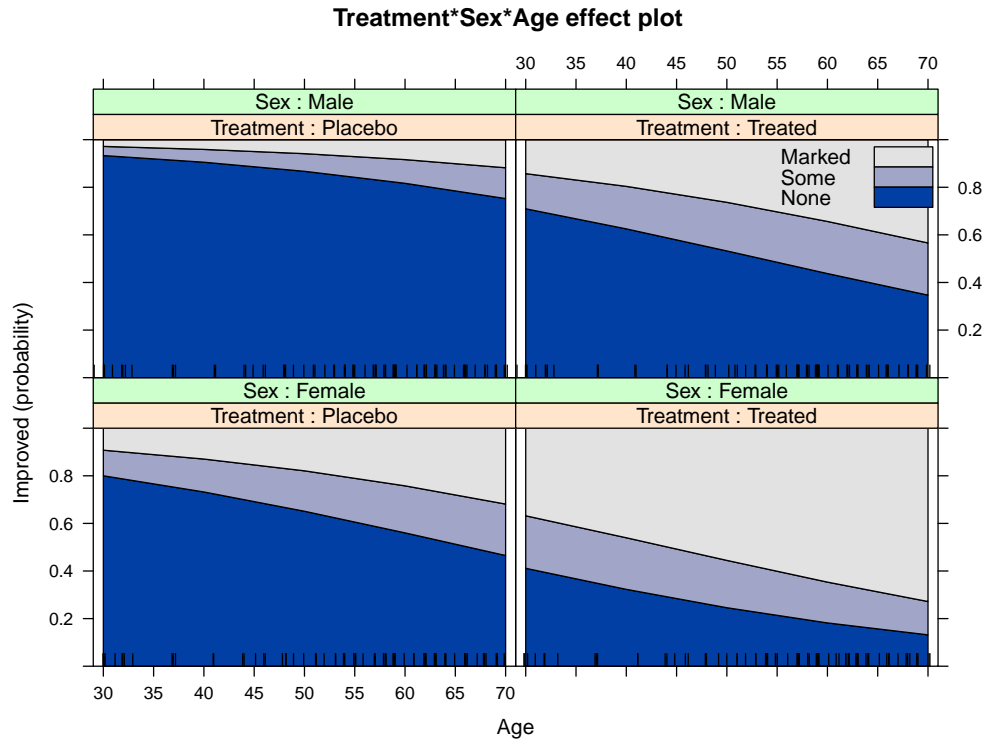


Figure 7.44: Effect plot for the effects of Treatment, Sex and Age in the Arthritis data. fig:arth-po-eff2

Sex) produced with the argument `latent=TRUE` to `Effect()`. In this plot, there is a single line in each panel for the effect (slope) of Age on the log odds. The dashed horizontal lines give the thresholds between the adjacent response categories corresponding to the intercepts.

```
plot(Effect(c("Treatment", "Age"), arth.polr, latent=TRUE), lwd=3)
```

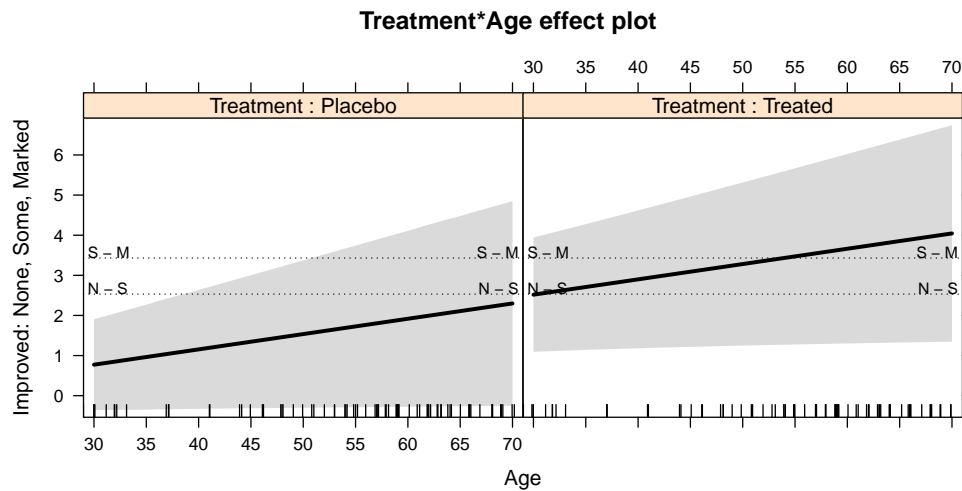


Figure 7.45: Latent variable effect plot for the effects of Treatment and Age in the Arthritis data. fig:arth-po-e

7.6.3 Nested dichotomies

sec:nested}

The method of *nested dichotomies* provides another simple way to analyse a polytomous response in the framework of logistic regression (or other generalized linear models). This method does not require an ordinal response or special software. Instead, it uses the familiar binary logistic model and fits $m - 1$ separate models for each of a hierarchically nested set of comparisons among the response categories.

Taken together, this set of models for the dichotomies comprises a complete model for the polytomous response. As well, these models are statistically independent, so test statistics such as G^2 or Wald tests can be added to give overall tests for the full polytomy.

For example, the response categories $Y = \{1,2,3,4\}$ could be divided first as $\{1,2\}$ vs. $\{3,4\}$, as shown in the left side of Figure 7.46. Then these two dichotomies could be divided as $\{1\}$ vs. $\{2\}$, and $\{3\}$ vs. $\{4\}$. Alternatively, these response categories could be divided as shown in the right side of Figure 7.46: first, $\{1\}$ vs. $\{2,3,4\}$, then $\{2\}$ vs $\{3,4\}$, and finally $\{3\}$ vs. $\{4\}$.

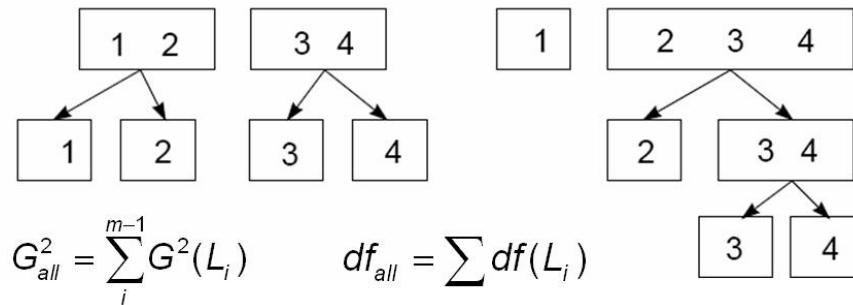


Figure 7.46: Nested dichotomies. The boxes show two different ways a four-category response can be represented as three nested dichotomies. Adapted from Fox (2008).

{fig:nested2}

Such models make the most sense when there are substantive reasons for considering the response categories in terms of such dichotomies. Two examples are shown in Figure 7.47.

- For the *Arthritis* data, it is sensible to consider one dichotomy (“better”), with logit L_1 , between the categories of "None" compared to "Some" or "Marked". A second dichotomy, with logit L_2 , would then distinguish between the some and marked response categories.
- For a second case where patients are classified into $m = 4$ psychiatric diagnostic categories, the first dichotomy, with logit L_1 distinguishes those considered normal from all others given a clinical diagnosis. Two other dichotomies are defined to further divide the non-normal categories.

Then, consider the separate logit models for these $m - 1$ dichotomies, with different intercepts α_j and slopes β_j for each dichotomy,

$$\begin{aligned}
 L_1 &= \alpha_1 + \mathbf{x}^\top \beta_1 \\
 L_2 &= \alpha_2 + \mathbf{x}^\top \beta_2 \\
 &\vdots \\
 L_{m-1} &= \alpha_{m-1} + \mathbf{x}^\top \beta_{m-1}
 \end{aligned}$$

{ex:wlfpart1}

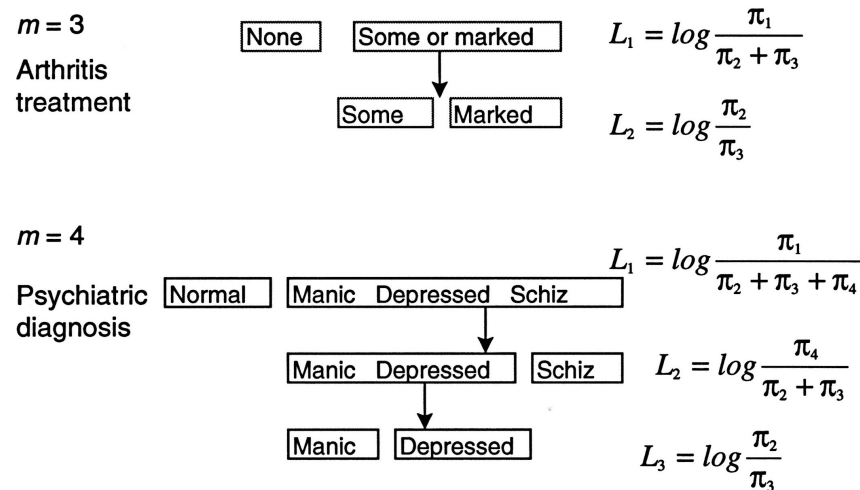


Figure 7.47: Examples of nested dichotomies and the corresponding logits

{fig:nested

EXAMPLE 7.17: Women's labor force participation

The data set `Women1f` in the `car` package gives the result of a 1977 Canadian survey. It contains data for 263 married women of age 21–30 who indicated their working status (outside the home) as not working, working part time or working full time, together with their husband's income and a binary indicator of whether they had one or more young children in their household. (Another variable, region of Canada, had no effects in these analyses, and is not examined here.) This example follows Fox and Weisberg (2011, §5.8).

```
library(car) # for data and Anova()
data("Women1f", package="car")
some(Women1f)

##      partic hincome children  region
## 6 not.work      7 present  Ontario
## 25 not.work     23 present  Ontario
## 36 not.work     19 absent  Ontario
## 83 fulltime    17 present  Ontario
## 138 not.work   13 present  Ontario
## 166 not.work    9 present  Atlantic
## 168 fulltime   13 absent  Ontario
## 173 not.work    7 present  Ontario
## 229 parttime   23 present  Quebec
## 233 fulltime   15 absent  Quebec
```

In this example, it makes sense to consider a first dichotomy (`working`) between women who are not working, vs. those who are (full time or part time). A second dichotomy (`fulltime`) contrasts full time work vs. part time work, among those women who are working at least part time. These two binary variables are created in the data frame using the `recode()` function from the `car` package.

```
# create dichotomies
Women1f <- within(Women1f, {
  working <- recode(partic, " 'not.work' = 'no'; else = 'yes' ")
  fulltime <- recode(partic,
    " 'fulltime' = 'yes'; 'parttime' = 'no'; 'not.work' = NA")})
some(Women1f)
```

```
##      partic hincome children region fulltime working
## 1  not.work      15 present Ontario <NA>      no
## 10 not.work      23 present Ontario <NA>      no
## 11 not.work      23 present Ontario <NA>      no
## 24 fulltime      11 absent Ontario  yes      yes
## 67 not.work      15 present Ontario <NA>      no
## 98 fulltime      15 absent Ontario  yes      yes
## 122 not.work     23 present Atlantic <NA>      no
## 167 not.work     15 present Ontario <NA>      no
## 208 fulltime     11 absent Quebec   yes      yes
## 241 not.work     13 present Quebec   <NA>      no
```

The tables below show how the response `partic` relates to the recoded binary variables, `working` and `fulltime`. Note that the `fulltime` variable is recoded to `NA` for women who are not working.

```
with(Womenlf, table(partic, working))
```

```
##      working
## partic  no yes
## fulltime  0 66
## not.work 155 0
## parttime  0 42
```

```
with(Womenlf, table(partic, fulltime, useNA="ifany"))
```

```
##      fulltime
## partic  no yes <NA>
## fulltime  0 66 0
## not.work  0 0 155
## parttime 42 0 0
```

We proceed to fit two separate binary logistic regression models for the derived dichotomous variables. For the `working` dichotomy, we get the following results:

```
mod.working <- glm(working ~ hincome + children, family=binomial,
                  data=Womenlf)
```

```
summary(mod.working)
```

```
##
## Call:
## glm(formula = working ~ hincome + children, family = binomial,
##      data = Womenlf)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.677  -0.865  -0.777   0.929   1.997
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.3358    0.3838   3.48  0.0005 ***
## hincome        -0.0423    0.0198  -2.14  0.0324 *
## childrenpresent -1.5756    0.2923  -5.39  7e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 356.15  on 262  degrees of freedom
```

```
## Residual deviance: 319.73 on 260 degrees of freedom
## AIC: 325.7
##
## Number of Fisher Scoring iterations: 4
```

And, similarly for the fulltime dichotomy:

```
mod.fulltime <- glm(fulltime ~ hincome + children, family=binomial,
                    data=Womenlwf)
summary(mod.fulltime)

##
## Call:
## glm(formula = fulltime ~ hincome + children, family = binomial,
##      data = Womenlwf)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.405  -0.868   0.395   0.621   1.764
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.4778     0.7671    4.53 5.8e-06 ***
## hincome       -0.1073     0.0392   -2.74 0.0061 **
## childrenpresent -2.6515     0.5411   -4.90 9.6e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 144.34 on 107 degrees of freedom
## Residual deviance: 104.49 on 105 degrees of freedom
## (155 observations deleted due to missingness)
## AIC: 110.5
##
## Number of Fisher Scoring iterations: 5
```

Although these were fit separately, we can view this as a combined model for the three-level response, with the following coefficients:

```
cbind(working=coef(mod.working), fulltime=coef(mod.fulltime))

##              working fulltime
## (Intercept)  1.335830  3.47777
## hincome     -0.042308 -0.10727
## childrenpresent -1.575648 -2.65146
```

Writing these out as equations for the logits, we have:

$$\{eq:wlf-logits\} \quad L_1 = \log \frac{\Pr(\text{working})}{\Pr(\text{notworking})} = 1.336 - 0.042 \text{ hincome} - 1.576 \text{ children} \quad (7.19)$$

$$L_2 = \log \frac{\Pr(\text{fulltime})}{\Pr(\text{parttime})} = 3.478 - 0.1072 \text{ hincome} - 2.652 \text{ children} \quad (7.20)$$

For both dichotomies, increasing income of the husband and the presence of young children decrease the log odds of a greater level of work. However, for those women who are working the effects of husband's income and children are greater on the choice between full time and part time work than they are for all women on the choice between working and not working.

As we mentioned above, the use of nested dichotomies implies that the models fit to the separate dichotomies are statistically independent. Thus, we can additively combine χ^2 statistics and degrees of freedom to give overall tests for the polytomous response.

For example, here we define a function, `LRtest()` to calculate the likelihood ratio test of the hypothesis $H_0 : \beta = \mathbf{0}$ for all predictors simultaneously. We then use this to display these tests for each sub-model, as well as the combined model based on the sums of the test statistic and degrees of freedom.

```
LRtest <- function(model)
  c(LRchisq=(model$null.deviance - model$deviance),
    df=(model$df.null - model$df.residual))
tab <- rbind(working=LRtest(mod.working),
             fulltime=LRtest(mod.fulltime))
tab <- rbind(tab, All = colSums(tab))
tab <- cbind(tab, pvalue = 1- pchisq(tab[,1], tab[,2]))
tab

##           LRchisq df      pvalue
## working    36.418  2 1.2355e-08
## fulltime   39.847  2 2.2252e-09
## All        76.265  4 1.1102e-15
```

Similarly, you can carry out tests of individual predictors, $H_0 : \beta_i = \mathbf{0}$ for the polytomy by adding the separate χ^2 s from `Anova()`.

```
Anova(mod.working)

## Analysis of Deviance Table (Type II tests)
##
## Response: working
##           LR Chisq Df Pr(>Chisq)
## hincome    4.82637  1  0.028028 *
## children   31.32288  1 2.1849e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(mod.fulltime)

## Analysis of Deviance Table (Type II tests)
##
## Response: fulltime
##           LR Chisq Df Pr(>Chisq)
## hincome    8.9813  1 0.0027275 **
## children   32.1363  1 1.4373e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For example, the test for husband's income gives $\chi^2 = 4.826 + 8.981 = 13.807$ with 2 df.

As before, you can plot the fitted values from such models, either on the logit scale (for the separate logit equations) or in terms of probabilities for the various responses. The general idea is the same: obtain the fitted values from `predict()` using data frame containing the values of the predictors. However, now we have to combine these for each of the sub-models.

We calculate these values below, on both the logit scale and the response scale of probabilities. The `newdata` argument to `predict()` is constructed as the combinations of values for

hincome and children.²⁴

```

predictors <- expand.grid(hincome=1:50,
                        children=c('absent', 'present'))
fit <- data.frame(predictors,
  p.working = predict(mod.working, predictors, type='response'),
  p.fulltime = predict(mod.fulltime, predictors, type='response'),
  l.working = predict(mod.working, predictors, type='link'),
  l.fulltime = predict(mod.fulltime, predictors, type='link')
)
print(some(fit, 5), digits=3)

##      hincome children p.working p.fulltime l.working l.fulltime
## 12         12  absent    0.696    0.899    0.828    2.191
## 17         17  absent    0.649    0.839    0.617    1.654
## 40         40  absent    0.412    0.307   -0.357   -0.813
## 65         15  present    0.294    0.314   -0.874   -0.783
## 94         44  present    0.109    0.020   -2.101   -3.893

```

One wrinkle here is that the probabilities for working full time and part time are conditional on working. We calculate the unconditional probabilities as shown below and choose to display the probability of *not* working as the complement of working.

```

fit <- within(fit, {
  full <- p.working * p.fulltime
  part <- p.working * (1 - p.fulltime)
  not <- 1 - p.working
})

```

Plotting these fitted values using `ggplot2` would require reshaping the `fit` data frame from wide to long format. Instead, we use R base graphics to produce plots of the probabilities and log odds. This method doesn't automatically give plots in separate panels, so a `for`-loop is used to generate panels for the levels of `children`. We set up an empty plot frame (`type="n"`) for each panel and then use `lines()` to plot the fitted probabilities. Using `par(mfrow=c(1,2))` places these plots in two side-by-side panels in a single display. The lines below give the plot shown in Figure 7.48.

```

op <- par(mfrow=c(1,2), mar=c(5,4,4,1)+.1)
Hinc <- 1:max(fit$hincome)
for (kids in c("absent", "present")) {
  dat <- subset(fit, children==kids)
  plot(range(Hinc), c(0,1), type="n", cex.lab=1.25,
    xlab="Husband's Income", ylab='Fitted Probability',
    main = paste("Children", kids))
  lines(Hinc, dat$not, lwd=3, col="black", lty=1)
  lines(Hinc, dat$part, lwd=3, col="blue", lty=2)
  lines(Hinc, dat$full, lwd=3, col="red", lty=3)
  if (kids=="absent") {
    legend("topright", lty=1:3, lwd=3, col=c("black", "blue", "red"),
      legend=c('not working', 'part-time', 'full-time'))
  }
}
par(op)

```

We can see how that the decision not to work outside the home increases strongly with husband's income, and is higher when there are children present. As well, among working women,

²⁴Alternatively, using the predictor values in the `Women1f` data would give the fitted values for the cases in the data, and allow a more data-centric plot as shown in Figure 7.42.

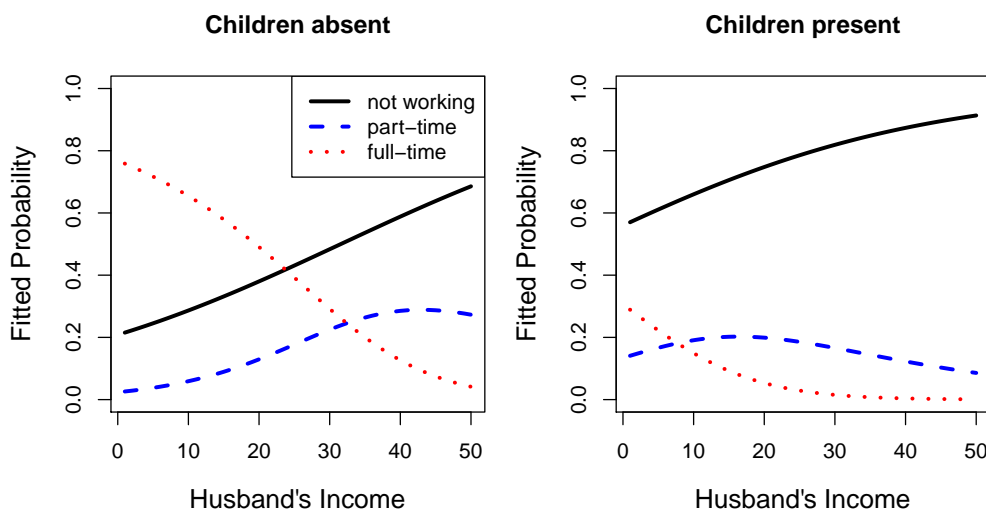


Figure 7.48: Fitted probabilities from the models for nested dichotomies fit to the data on women's labor force participation. fig:wlf-fitted-prob

the decision to work full time as opposed to part time decreases strongly with husband's income, and is less likely with young children.

Similarly, we plot the fitted logits for the two dichotomies in `l.working` and `l.fulltime` as shown below, giving Figure 7.49.

```
op <- par(mfrow=c(1,2), mar=c(5,4,1,1)+.1)
for ( kids in c("absent", "present") ) {
  dat <- subset(fit, children==kids)
  plot( range(Hinc), c(-4,5), type="n", cex.lab=1.25,
        xlab="Husband's Income", ylab='Fitted log odds')
  lines(Hinc, dat$l.working, lwd=3, col="black", lty=1)
  lines(Hinc, dat$l.fulltime, lwd=3, col="blue", lty=2)
  text(25, 4.5, paste("Children", kids), cex=1.4)
  if (kids=="absent") {
    legend("bottomleft", lty=1:3, lwd=3, col=c("black", "blue"),
          legend=c('working', 'full-time'))
  }
}
par(op)
```

This is essentially a graph of the fitted equations for L_1 and L_2 shown in Eqn. (7.19). It shows how the choice of full time work as opposed to part time depends more strongly on husband's income among women who are working than does the choice of working at all among all women. It also illustrates why the proportional odds assumption would not be reasonable for this data: that would require equal slopes for the two lines within each panel.

△

7.6.4 Generalized logit model

{sec:genlogit}

The generalized logit (or multinomial logit) approach models the probabilities of the m response categories directly as a set of $m - 1$ logits. These compare each of the first $m - 1$ categories to

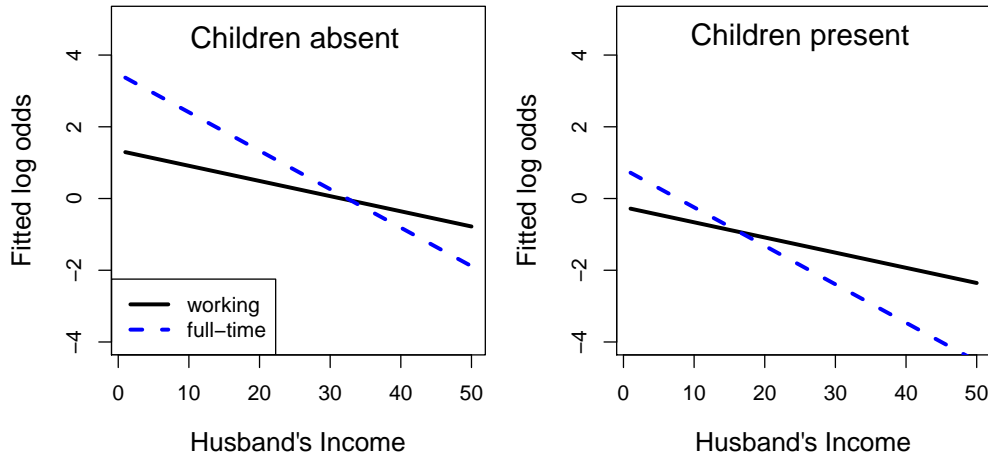


Figure 7.49: Fitted log odds from the models for nested dichotomies fit to the data on women’s labor force participation.

the last category, which serves as the baseline.²⁵ The logits for any other pair of categories can be retrieved from the $m - 1$ fitted ones.

When there are p predictors, x_1, x_2, \dots, x_p , which may be quantitative or categorical, the generalized logit model expresses the logits as

$$\begin{aligned}
 L_{jm} &\equiv \log \frac{\pi_{ij}}{\pi_{im}} = \beta_{0j} + \beta_{1j} x_{i1} + \beta_{2j} x_{i2} + \dots + \beta_{kj} x_{ip} \quad j = 1, \dots, m - 1 \\
 &= \mathbf{x}_i^T \boldsymbol{\beta}_j
 \end{aligned}
 \tag{7.21}$$

Thus, there is one set of fitted coefficients, $\boldsymbol{\beta}_j$ for each response category except the last. Each coefficient, β_{hj} , gives the effect, for a unit change in the predictor x_h , on the log odds that an observation had a response in category $Y = j$, as opposed to category $Y = m$.

The probabilities themselves can be expressed as

$$\begin{aligned}
 \pi_{ij} &= \frac{\exp(\mathbf{x}_i^T \boldsymbol{\beta}_j)}{1 + \sum_{\ell=1}^{m-1} \exp(\mathbf{x}_i^T \boldsymbol{\beta}_\ell)} \quad j = 1, 2, \dots, m - 1 \\
 \pi_{im} &= 1 - \sum_{i=1}^{m-1} \pi_{ij} \quad \text{for } Y = m
 \end{aligned}$$

Parameters in the $m - 1$ equations Eqn. (7.21) can be used to determine the probabilities or the predicted log odds for any pair of response categories by subtraction. For instance, for an arbitrary pair of categories, a and b , and two predictors, x_1 and x_2 ,

$$\begin{aligned}
 L_{ab} &= \log \frac{\pi_{ia}/\pi_{im}}{\pi_{ib}/\pi_{im}} \\
 &= \log \frac{\pi_{ia}}{\pi_{im}} - \log \frac{\pi_{ib}}{\pi_{im}} \\
 &= (\beta_{0a} - \beta_{0b}) + (\beta_{1a} - \beta_{1b})x_{i1} + (\beta_{2a} - \beta_{2b})x_{i2}
 \end{aligned}$$

²⁵When the response is a factor, any category can be selected as the baseline level using `relevel()`.

For example, the coefficient for x_{i1} in L_{ab} is just $(\beta_{1a} - \beta_{1b})$. Similarly, the predicted logit for any pair of categories can be calculated as

$$\hat{L}_{ab} = \hat{L}_{am} - \hat{L}_{bm} .$$

The generalized logit model can be fit most conveniently in R using the function `multinom()` in the `nnet` package and the `effects` package has a set of methods for "multinom" models. These models can also be fit using VGAM and the `mlogit` package.

{ex:wlfpart2}

EXAMPLE 7.18: Women's labor force participation

To illustrate this method, we fit the generalized logit model to the women's labor force participation data as explained below. The response, `partic` is a character factor, and, by default `multinom()` treats these in alphabetical order and uses the *first* level as the baseline category.

```
levels(Womenlwf$partic)
## [1] "fulltime" "not.work" "parttime"
```

Although the multinomial model does not depend on the baseline category, it makes interpretation easier to choose "not.work" as the reference level, which we do with `relevel()`.²⁶

```
# choose not working as baseline category
Womenlwf$partic <- relevel(Womenlwf$partic, ref="not.work")
```

We fit the main effects model for husband's income and children as follows. As we did with `polr()` (Section 7.6.1), specifying `Hess=TRUE` saves the Hessian and facilitates calculation of standard errors and hypothesis tests.

```
library(nnet)
wlf.multinom <- multinom(partic ~ hincome + children,
                          data=Womenlwf, Hess=TRUE)
## # weights: 12 (6 variable)
## initial value 288.935032
## iter 10 value 211.454772
## final value 211.440963
## converged
```

The `summary()` method for "multinom" objects doesn't calculate test statistics for the estimated coefficients by default. The option `Wald=TRUE` produces Wald z -test statistics, calculated as $z = \beta/SE(\beta)$.

```
summary(wlf.multinom, Wald=TRUE)
## Call:
## multinom(formula = partic ~ hincome + children, data = Womenlwf,
## Hess = TRUE)
##
## Coefficients:
## (Intercept) hincome childrenpresent
## fulltime 1.9828 -0.0972321 -2.558605
```

²⁶Alternatively, we could declare `partic` an *ordered* factor, using `ordered()`.

```
## parttime      -1.4323  0.0068938      0.021456
##
## Std. Errors:
##      (Intercept)  hincome  childrenpresent
## fulltime      0.48418 0.028096      0.36220
## parttime      0.59246 0.023455      0.46904
##
## Value/SE (Wald statistics):
##      (Intercept)  hincome  childrenpresent
## fulltime      4.0953 -3.46071      -7.064070
## parttime      -2.4176  0.29392      0.045744
##
## Residual Deviance: 422.88
## AIC: 434.88
```

Notice that the coefficients, their standard errors and the Wald test z values are printed in separate tables. The first line in each table pertains to the logit comparing full time work with the not working reference level; the second line compares part time work against not working.

For those who like p -values for significance tests, you can calculate these from the results returned by the `summary()` method in the `Wald.ratios` component, using the standard normal asymptotic approximation:

```
stats <- summary(wlf.multinom, Wald=TRUE)
z <- stats$Wald.ratios
p <- 2 * (1 - pnorm(abs(z)))
zapsmall(p)

##      (Intercept) hincome  childrenpresent
## fulltime      0.00004 0.00054      0.00000
## parttime      0.01562 0.76882      0.96351
```

The interpretation of these tests is that both husband's income and presence of children have highly significant effects on the comparison of working full time as opposed to not working, while neither of these predictors are significant for the comparison of working part time vs. not working.

So far, we have assumed that the effects of husband's income and presence of young children are additive on the log odds scale. We can test this assumption by allowing an interaction of those effects and testing it for significance.

```
wlf.multinom2 <- multinom(partic ~ hincome * children,
                          data=Womenlf, Hess=TRUE)

## # weights: 15 (8 variable)
## initial value 288.935032
## iter 10 value 210.797079
## final value 210.714841
## converged

Anova(wlf.multinom2)

## Analysis of Deviance Table (Type II tests)
##
## Response: partic
##      LR Chisq Df Pr(>Chisq)
## hincome      15.2  2  0.00051 ***
## children     63.6  2  1.6e-14 ***
```

```
## hincome:children      1.5  2    0.48378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test for the interaction term, `hincome:children` is not significant, so we can abandon this model.

Full model plots of the fitted values can be plotted as shown earlier in Example 7.17: obtain the fitted values over a grid of the predictors and plot these.

```
predictors <- expand.grid(hincome=1:50,
                        children=c('absent', 'present'))
fit <- data.frame(predictors,
                 predict(wlf.multinom, predictors, type='probs'))
```

Plotting these fitted values gives the plot shown in Figure 7.50.

```
op <- par(mfrow=c(1,2), mar=c(5,4,4,1)+.1)
Hinc <- 1:max(fit$hincome)
for (kids in c("absent", "present")) {
  dat <- subset(fit, children==kids)
  plot(range(Hinc), c(0,1), type="n", cex.lab=1.25,
        xlab="Husband's Income", ylab='Fitted Probability',
        main = paste("Children", kids))
  lines(Hinc, dat$not.work, lwd=3, col="black", lty=1)
  lines(Hinc, dat$parttime, lwd=3, col="blue", lty=2)
  lines(Hinc, dat$fulltime, lwd=3, col="red", lty=3)
  if (kids=="absent") {
    legend("topright", lty=1:3, lwd=3, col=c("black", "blue", "red"),
          legend=c('not working', 'part-time', 'full-time'))
  }
}
par(op)
```

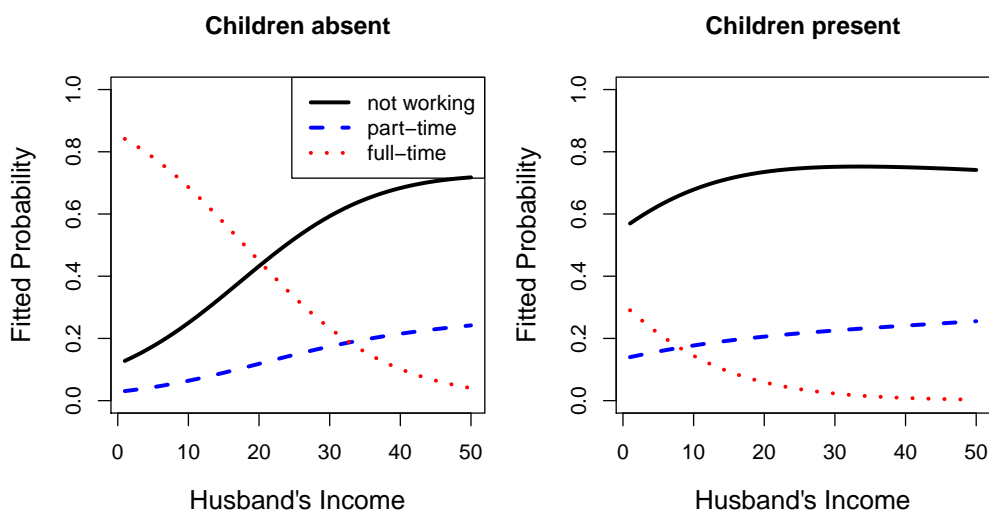


Figure 7.50: Fitted probabilities from the generalized logit model fit to the data on women's labor force participation.

The results shown in this plot are roughly similar to those obtained from the nested dichotomy models, graphed in Figure 7.48. However, the predicted probabilities of not working under the generalized logit model rise more steeply with husband's income for women with no children and level off sooner for women with young children.

The `effects` package has special methods for "multinom" models. It treats the response levels in the order given by `levels()`, so before plotting we use `ordered()` to arrange levels in their natural order. The `update()` method provides a simple way to get a new fitted model; in the call, the model formula `. ~ .` means to fit the same model as before, i.e., `partic ~ hincome + children`.

```
levels(Womenlf$partic)

## [1] "not.work" "fulltime" "parttime"

Womenlf$partic <- ordered(Womenlf$partic,
                          levels=c('not.work', 'parttime', 'fulltime'))
wlf.multinom <- update(wlf.multinom, . ~ .)

## # weights: 12 (6 variable)
## initial value 288.935032
## iter 10 value 211.454772
## final value 211.440963
## converged
```

As illustrated earlier, you can use `plot(allEffects(model), ...)` to plot all the high-order terms in the model, either with separate curves for each response level (`style="lines"`) or as cumulative filled polygons (`style="stacked"`). Here, we simply plot the effects for the combinations of husband's income and children in stacked style, giving a plot (Figure 7.51) that is analogous to the full-model plot shown in Figure 7.50.

```
plot(Effect(c("hincome", "children"), wlf.multinom),
      style="stacked", key.args=list(x=.05, y=.9))
```

△

7.7 Chapter summary

{sec:ch07-summary}

- Model-based methods for categorical data provide confidence intervals for parameters and predicted values for observed and unobserved values of the explanatory variables. Graphical displays of predicted values help us to interpret the fitted relations by smoothing a discrete response.
- The logistic regression model (Section 7.2) describes the relationship between a categorical response variable, usually dichotomous, and a set of one or more quantitative or discrete explanatory variables (Section 7.3) It is conceptually convenient to specify this model as a linear model predicting the log odds (or logit) of the probability of a success from the explanatory variables.
- The relationship between a discrete response and a quantitative predictor may be explored graphically by plotting the binary observations against the predictor with some smoothed curve(s), either parametric or non-parametric, possibly stratified by other predictors.

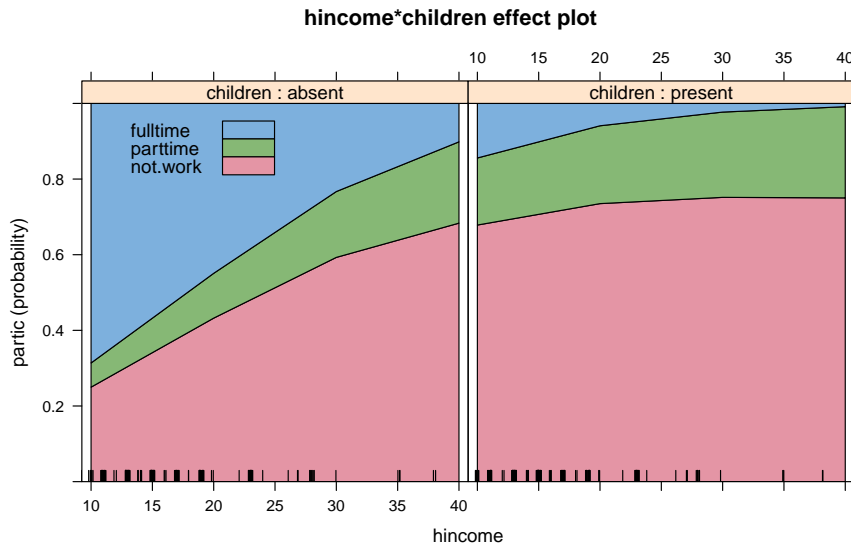


Figure 7.51: Effect plot for the probabilities of not working and working part time and full time from the generalized logit model fit to the women's labor force data. Fig.wlf-multi-effect

- For both quantitative and discrete predictors, the results of a logistic regression are most easily interpreted from full-model plots of the fitted values against the predictors, either on the scale of predicted probabilities or log odds (Section 7.3.2). In these plots, confidence intervals provide a visual indication of the precision of the predicted results.
- When there are multiple predictors and/or higher-order interaction terms, effect plots (Section 7.3.3) provide an important method for constructing simplified displays, focusing on the higher-order terms in a given model.
- Influence diagnostics (Section 7.5) assess the impact of individual cases or groups on the fitted model, predicted values, and the coefficients of individual predictors. Among other displays, plots of residuals against leverage showing Cook's D are often most useful.
- Other diagnostic plots (Section 7.5.3) include component-plus-residual plots, that are useful for detecting non-linear relationships for a quantitative predictor, and added-variable plots, that show the partial relations of the response to a given predictor, controlling or adjusting for all other predictors.
- Polytomous responses may be handled in several ways as extensions of binary logistic regression (Section 7.6): (a) The *proportional odds model* (Section 7.6.1) is simple and convenient, but its validity depends on an assumption of equal slopes for adjacent-category logits. (b) *Nested dichotomies* (Section 7.6.3) among the response categories give a set of models which may be regarded as a single, combined model for the polytomous response. (c) *Generalized logit models* (Section 7.6.4) may be used to construct models comparing any pair of categories.

{lab:7.5}

Exercise 7.5 For the women's labor force participation data (`Womenlf`) the response variable, `partic`, can be treated as ordinal by using

```
Womenlf$partic <- ordered(Womenlf$partic,
                          levels=c('not.work', 'parttime', 'fulltime'))
```

Use the methods in Section 7.6.1 to test whether the proportional odds model holds for these data.

{lab:7.6}

Exercise 7.6 The data set `housing` in the `MASS` package gives a $3 \times 3 \times 4 \times 2$ table in frequency form relating (a) satisfaction (`Sat`) of residents with their housing (`High`, `Medium`, `Low`), (b) perceived degree of influence (`Infl`) they have on the management of the property (`High`, `Medium`, `Low`), (c) Type of rental (`Tower`, `Atrium`, `Apartment`, `Terrace`), and (d) contact (`Cont`) residents have with other residents (`Low`, `High`). Consider satisfaction as the ordinal response variable.

- Fit the proportional odds model with additive (main) effects of housing type, influence in management and contact with neighbors to this data. (Hint: Using `polr()`, with the data in frequency form, you need to use the `weights` argument to supply the `Freq` variable.)
- Investigate whether any of the two-factor interactions among `Infl`, `Type` and `Cont` add substantially to goodness of fit of this model. (Hint: use `stepAIC()`, with the scope formula `~ .^2` and `direction="forward"`.)
- For your chosen model from the previous step, use the methods of Section 7.6.2 to plot the probabilities of the categories of satisfaction.
- Write a brief summary these analyses, interpreting *how* satisfaction with housing depends on the predictor variables.

{lab:7.7}

Exercise 7.7 The data `TV` on television viewing was analyzed using correspondence analysis in Example 6.4, ignoring the variable `Time` and extended in Exercise 6.7. Treating `Network` as a three-level response variable, fit a generalized logit model (Section 7.6.4) to explain the variation in viewing in relation to `Day` and `Time`. The `TV` data is a three-way table, so you will need to convert it to a frequency data frame first.

```
data("TV", package="vcdExtra")
TV.df <- as.data.frame.table(TV)
```

- Fit the main-effects model, `Network ~ Day + Time` with `multinom()`. Note that you will have to supply the `weights` argument because each row of `TV.df` represents the number of viewers in the `Freq` variable.
- Prepare an effects plot for the fitted probabilities in this model.
- Interpret these results in comparison to the correspondence analysis analysis in Example 6.4.

{lab:7.8}

Exercise 7.8 Refer to Exercise 5.8 for a description of the `Accident` data. The interest here is to model the probability that an accident resulted in death rather than injury from the predictors `age`, `mode` and `gender`. With `glm()`, and the data in the form of a frequency table, you can use the argument `weight=Freq` to take cell frequency into account.

- Fit the main effects model, `result=="Died" ~ age + mode + gender`. Use `car::Anova()` to assess the model terms.

- (b) Fit the model that allows all two-way interactions. Use `anova()` to test whether this model is significantly better than the main effects model.
- (c) Fit the model that also allows the three-way interaction of all factors. Does this offer any improvement over the two-way model?
- (d) Interpret the results of the analysis using effect plots for the two-way model, separately for each of the model terms. Describe verbally the nature of the `age*gender` effect. Which mode of transportation leads to greatest risk of death?

```
.locals$ch07 <- setdiff(ls(), .globals)
# .locals$ch07
remove(list=.locals$ch07[sapply(.locals$ch07, function(n){!is.function(get(n))})])
detach(package:gpairs)
detach(package:rms)
detach(package:VGAM)
.pkgs$ch07 <- setdiff(.packages(), .pkgs$ch07)
.pkgs$ch07

## [1] "Hmisc"          "nnet"           "reshape2"      "stats4"
## [5] "proto"          "directlabels"  "quadprog"      "SparseM"
## [9] "Formula"        "survival"      "splines"       "vcdExtra"
## [13] "gnm"           "vcd"           "car"           "effects"
## [17] "colorspace"    "grid"          "lmtest"        "zoo"
## [21] "ca"            "Lahman"        "lattice"       "ggplot2"
## [25] "MASS"          "gmodels"
```

References

- Bliss, C. I. (1934). The method of probits. *Science*, 79(2037), 38–39.
- Bollen, K. A. (2002). Latent variables in psychology and the social sciences. *Annual Review of Psychology*, 53(1), 605–634.
- Box, G. E. P. and Draper, N. R. (1987). *Empirical Model Building and Response Surfaces*. New York, NY: John Wiley & Sons.
- Cook, R. D. and Weisberg, S. (1999). *Applied Regression Including Computing and Graphics*. New York: Wiley.
- Dalal, S., Fowlkes, E. B., and Hoadley, B. (1989). Risk analysis of the space shuttle: Pre-Challenger prediction of failure. *Journal of the American Statistical Association*, 84(408), 945–957.
- de la Cruz Rot, M. (2005). Improving the presentation of results of logistic regression with r. *Bulletin of the Ecological Society of America*, 86, 41–48.
- Finney, D. J. (1947). *Probit analysis*. Cambridge England: University Press.
- Fowlkes, E. B. (1987). Some diagnostics for binary logistic regression via smoothing. *Biometrika*, 74(3), 503–5152.
- Fox, J. (1987). Effect displays for generalized linear models. In C. C. Clogg, ed., *Sociological Methodology, 1987*, (pp. 347–361). San Francisco: Jossey-Bass.
- Fox, J. (2003). Effect displays in R for generalized linear models. *Journal of Statistical Software*, 8(15), 1–27.
- Fox, J. (2008). *Applied Regression Analysis and Generalized Linear Models*. Thousand Oaks, CA: Sage, 2nd edn.
- Fox, J. and Weisberg, S. (2011). *An R Companion to Applied Regression*. Thousand Oaks CA: Sage, 2nd edn.
- Grayson, D. K. (1990). Donner party deaths: A demographic assessment. *Journal of Anthropological Research*, 46(3), 223–242.
- Harrell, Jr, F. E. (2001). *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Graduate Texts in Mathematics. New York: Springer.
- Hosmer, Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied Logistic Regression*. New York: John Wiley and Sons, 3rd edn.

- Johnson, K. (1996). *Unfortunate Emigrants: Narratives of the Donner Party*. Logan, UT: Utah State University Press.
- Landwehr, J. M., Pregibon, D., and Shoemaker, A. C. (1984). Graphical methods for assessing logistic regression models. *Journal of the American Statistical Association*, 79, 61–71.
- Larsen, W. A. and McCleary, S. J. (1972). The use of partial residual plots in regression analysis. *Technometrics*, 14, 781–790.
- Lavine, M. (1991). Problems in extrapolation illustrated with space shuttle O-ring data. *Journal of the American Statistical Association*, 86, 912–922.
- Lawrance, A. J. (1995). Deletion influence and masking in regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1), 181–189.
- Lazarsfeld, P. F. (1950). The logical and mathematical foundation of latent structure analysis. In S. A. Stouffer, L. Guttman, E. A. Suchman, P. F. Lazarsfeld, S. A. Star, and J. A. Clausen, eds., *Studies in Social Psychology in World War II, vol. IV, Measurement and Prediction*, (pp. 362–412). Princeton, NJ: Princeton University Press.
- Lazarsfeld, P. F. (1954). A conceptual introduction to latent structure analysis. In P. F. Lazarsfeld, ed., *Mathematical Thinking in the Social Sciences*, (pp. 349–387). Glencoe, IL: Free Press.
- Lemeshow, S., Avrunin, D., and Pastides, J. S. (1988). Predicting the outcome of intensive care unit patients. *Journal of the American Statistical Association*, 83, 348–356.
- McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society*, 42, 109–142.
- Peterson, B. and Harrell, F. (1990). Partial proportional odds models for ordinal response variables. *Applied Statistics*, 39, 205–217.
- Pregibon, D. (1981). Logistic regression diagnostics. *Annals of Statistics*, 9, 705–724.
- Searle, S. R., Speed, F. M., and Milliken, G. A. (1980). Population marginal means in the linear model: An alternative to least squares means. *The American Statistician*, 34(4), 216–221.
- Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review*, 34, 278–286.
- Tufte, E. R. (1997). *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire, CT: Graphics Press.
- Wang, P. C. (1985). Adding a variable in generalized linear models. *Technometrics*, 27, 273–276.
- Williams, D. A. (1987). Generalized linear model diagnostics using the deviance and single case deletions. *Applied Statistics*, 36, 181–191.