STATS 330

Handout 6
Using residuals to assess goodness-of-fit

Department of Statistics, University of Auckland

Recap from STATS 20X

The errors

Recall the simple linear regresion model:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

$$\epsilon_i \overset{iid}{\sim} \text{Normal}(0, \sigma^2),$$

where ϵ_i is the ith observation's *error*: the difference between the observed value, Y_i, and the expected value, μ_i:

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 x_i + \epsilon_i \\ \epsilon_i &= Y_i - (\beta_0 + \beta_1 x_i) \\ \epsilon_i &= Y_i - \mu_i. \end{aligned}$$

If the model is correct, the errors all come from the same distribution: a normal distribution with mean 0 and variance σ^2.

Recap from STATS 20X

The errors

In practice, we do not know the true parameter values β_0 and β_1, and so we do not know the true expectation μ_i for any given observation.

Therefore, we cannot calculate the ith error,

$$\epsilon_i = Y_i - (\beta_0 + \beta_1 x_i), \text{or}$$
$$\epsilon_i = Y_i - \mu_i.$$

However, we can calculate *estimates* of β_0, β_1, and therefore μ_i. These are denoted by $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\mu}_i$, respectively.

Recap from STATS 20X

The residuals

We can therefore calculate an *estimate* of the ith error:

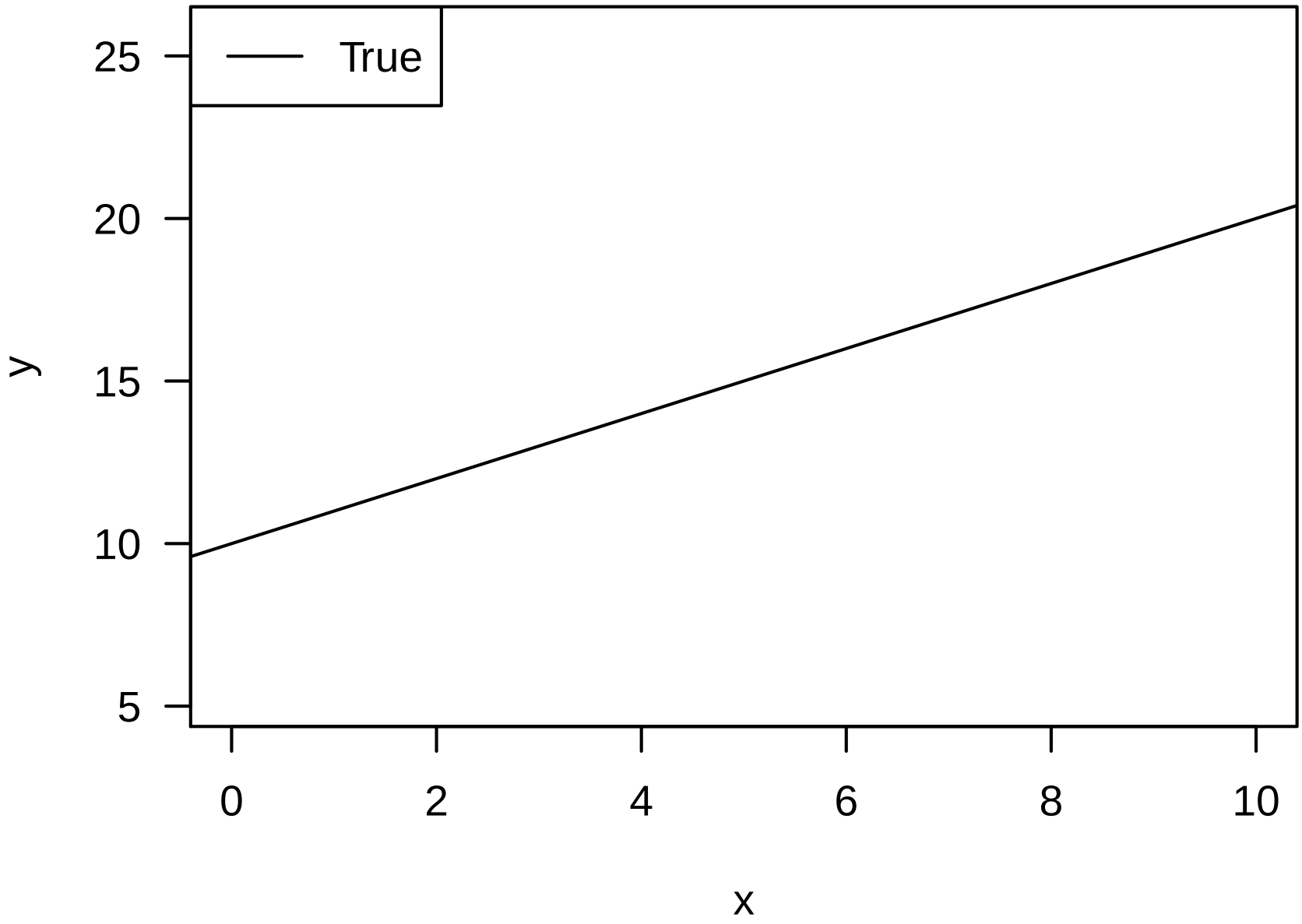
$$\hat{\epsilon}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i), \text{or}$$
$$\hat{\epsilon}_i = Y_i - \hat{\mu}_i$$

These estimates of the errors are known as the *residuals*, and are sometimes denoted r_i, rather than $\hat{\epsilon}_i$.

Each residual is the difference between the observed value of the response and the expected value of the response, as estimated by our model.

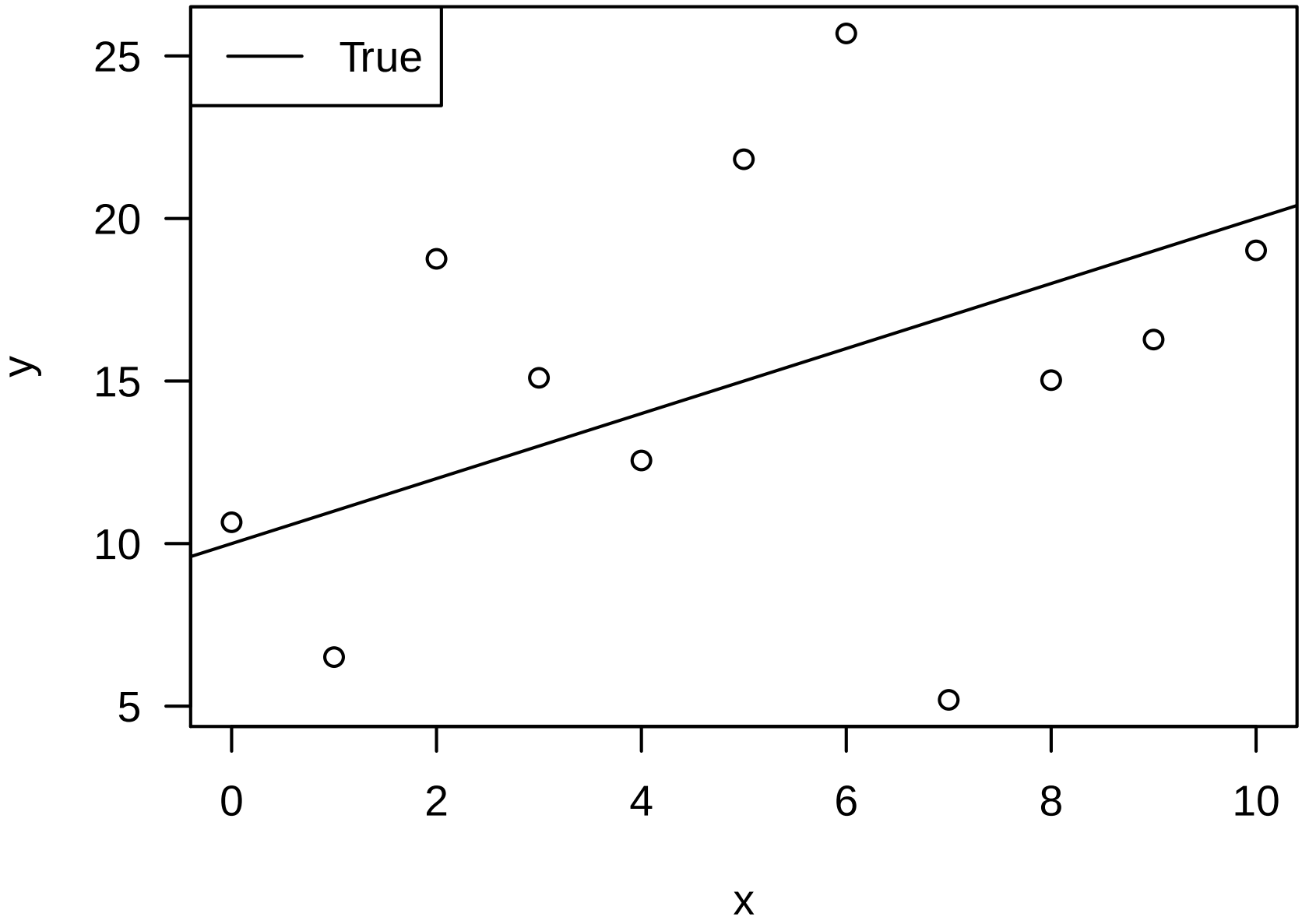
Recap from STATS 20X

The true regression line



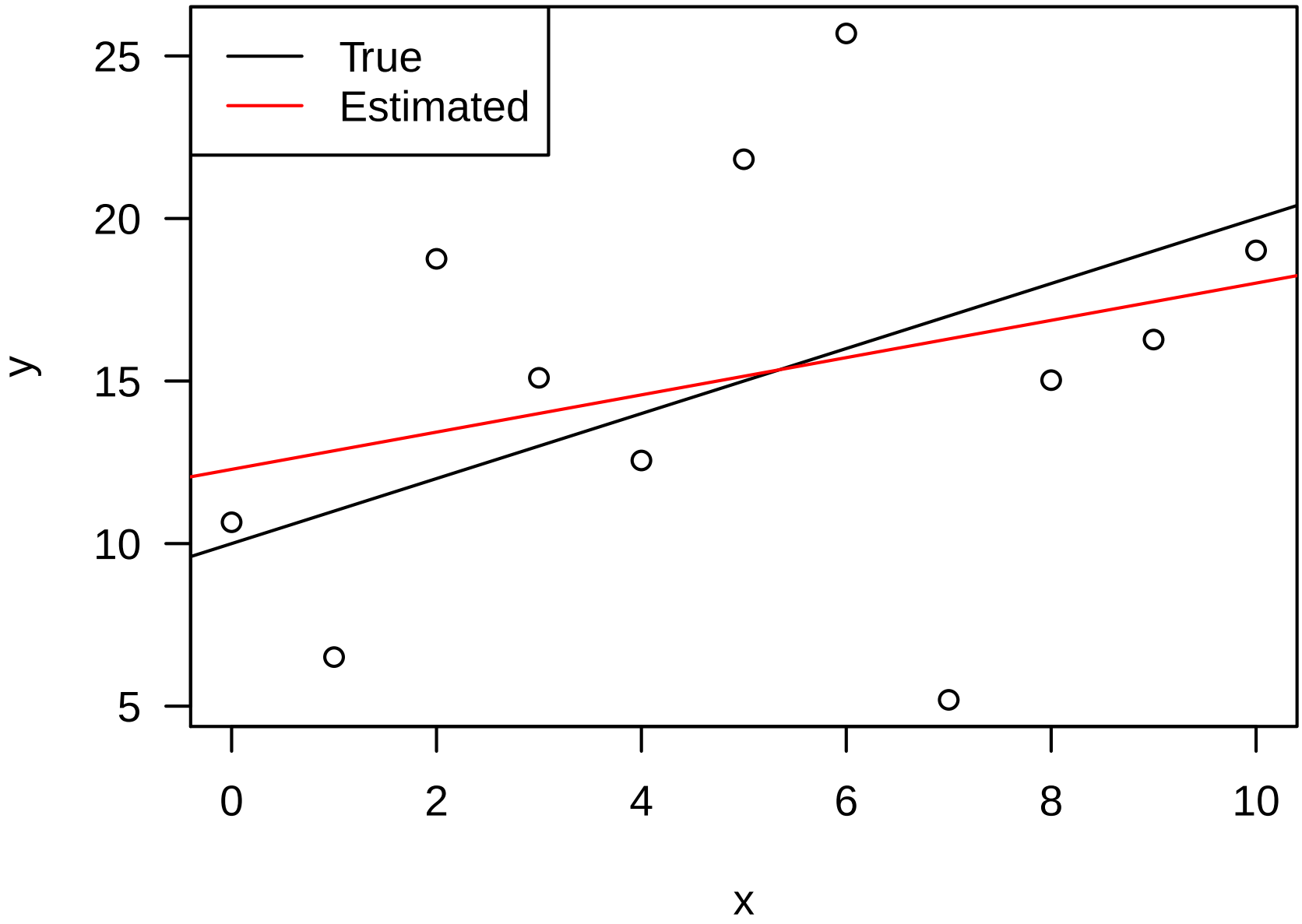
Recap from STATS 20X

Some observed data



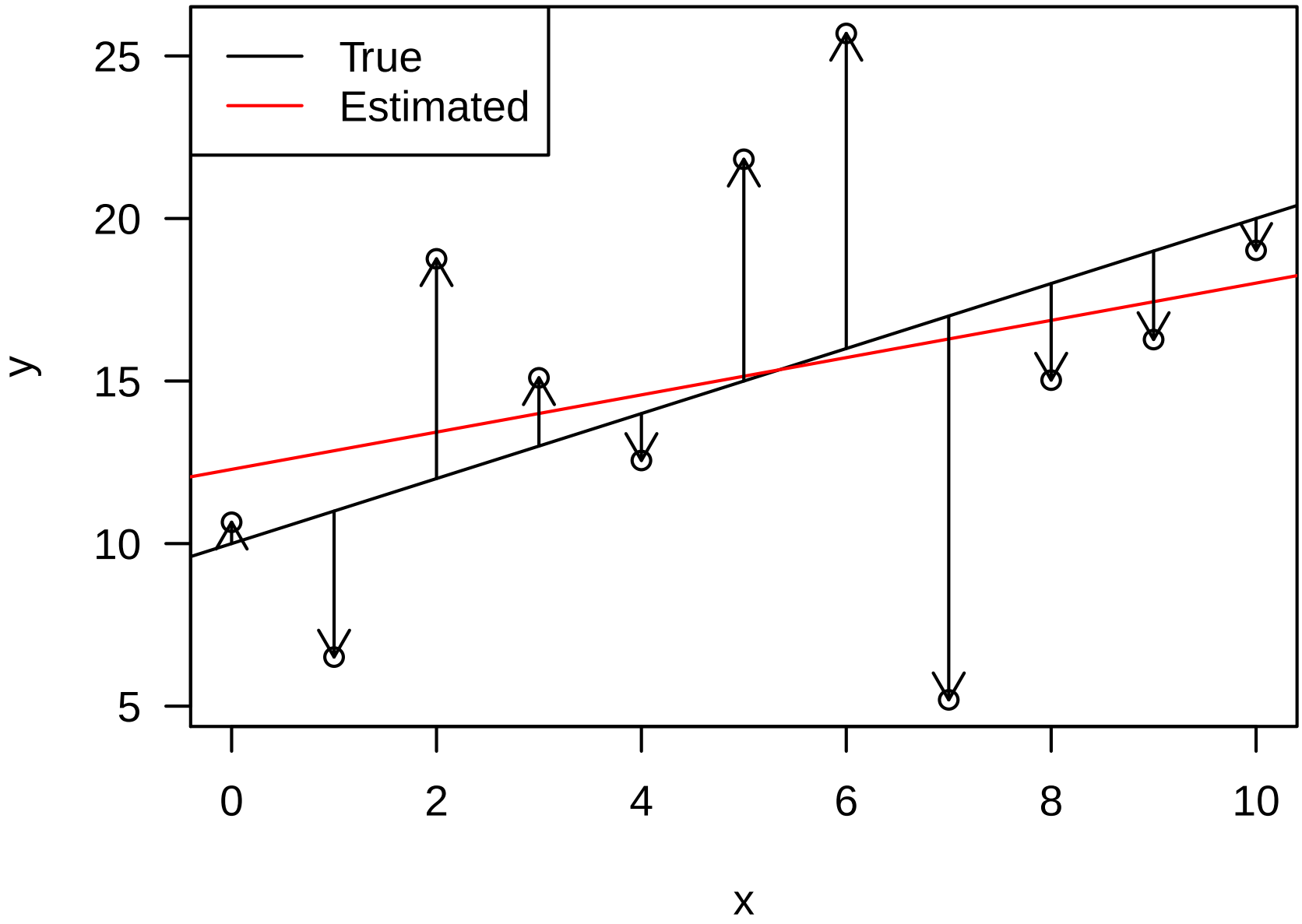
Recap from STATS 20X

The estimated regression line



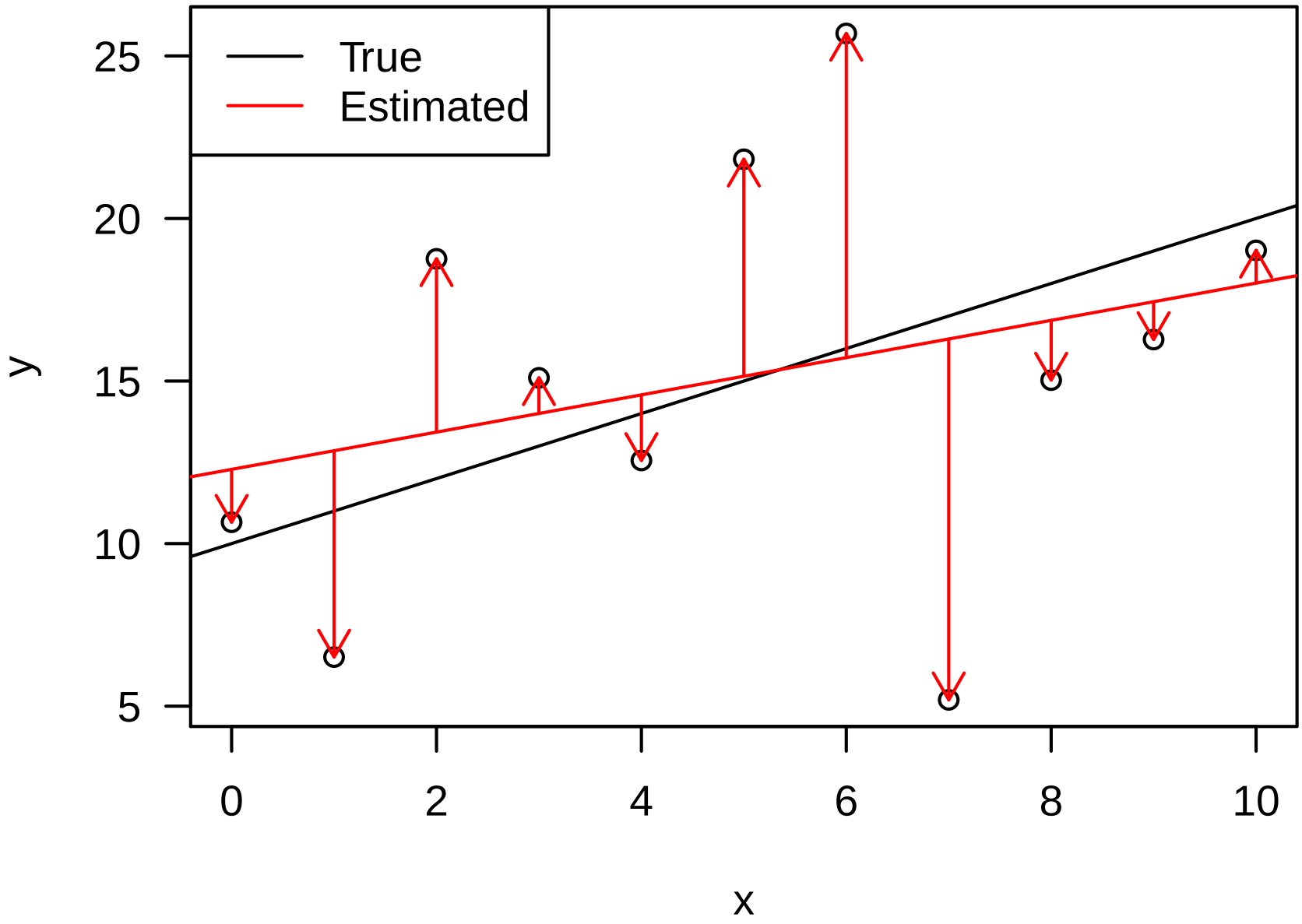
Recap from STATS 20X

The errors



Recap from STATS 20X

The residuals



Recap from STATS 20X

In R

If we fit a model with `lm()`...

```
fit <- lm(y ~ x)
```

... We can extract fitted values using `predict()`:

```
head(predict(fit))
```

```
##        1        2        3        4        5        6
## 12.28335 12.85622 13.42909 14.00196 14.57482 15.14769
```

... And residuals using `residuals()`:

```
head(residuals(fit))
```

```
##         1         2         3         4         5         6
## -1.625651 -6.345771  5.330634  1.098417 -2.017122  6.675116
```

Recap from STATS 20X

Assessing goodness-of-fit via the residuals

We assume

$$\epsilon_i \overset{iid}{\sim} \text{Normal}(0, \sigma^2).$$

In other words, our model assumes that all errors come from a normal distribution with mean zero, and all have the same variance, σ^2. However, we don't observe the errors, so we cannot test these assumptions directly.

However, we can calculate the residuals. In principle, if we do a good job of estimating the regression line (i.e., if $\beta_0 \approx \hat{\beta}_0$ and $\beta_1 \approx \hat{\beta}_1$) then the errors and the residuals will be similar (i.e., $r_i \approx \epsilon_i$).

If the red and black lines in the previous plot are similar, then the residuals will be a close approximation of the errors.

Recap from STATS 20X

Assessing goodness-of-fit via the residuals

Therefore, if the model is correct, then we might expect that all residuals come from an approximately normal distribution with mean zero, and have approximately equal variance, $\hat{\sigma}^2$.

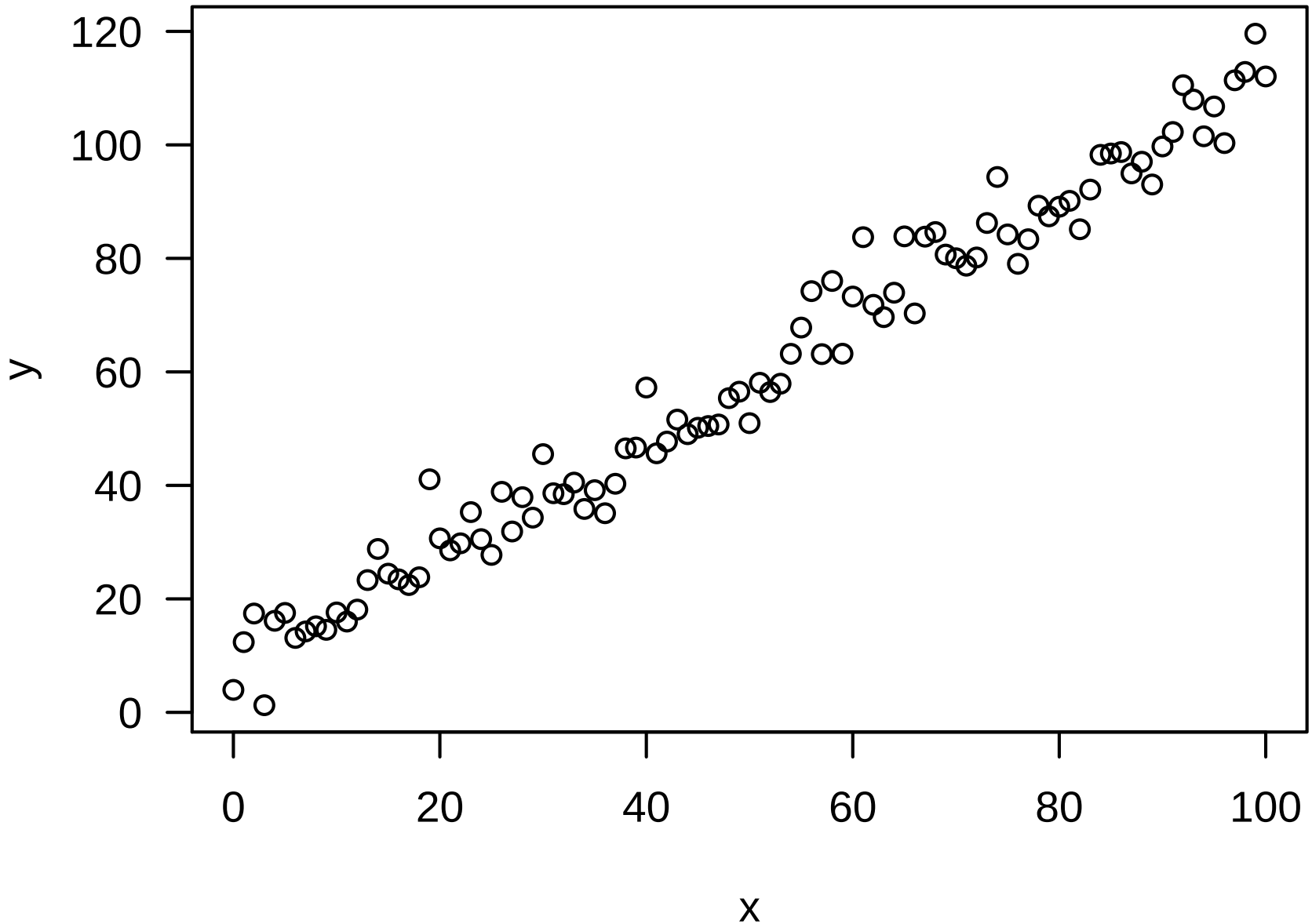
It is common to plot the residuals (r_i) against the fitted values (μ_i) and determine subjectively whether it is plausible that

1. the distribution of the residuals is approximately normal,
2. the residuals have mean zero across all fitted values, and
3. the residuals have constant variance across all fitted values.

If our sample size is large, our inference is robust to a violation of (1) due to the central limit theorem. We therefore focus chiefly on (2) and (3).

Recap from STATS 20X

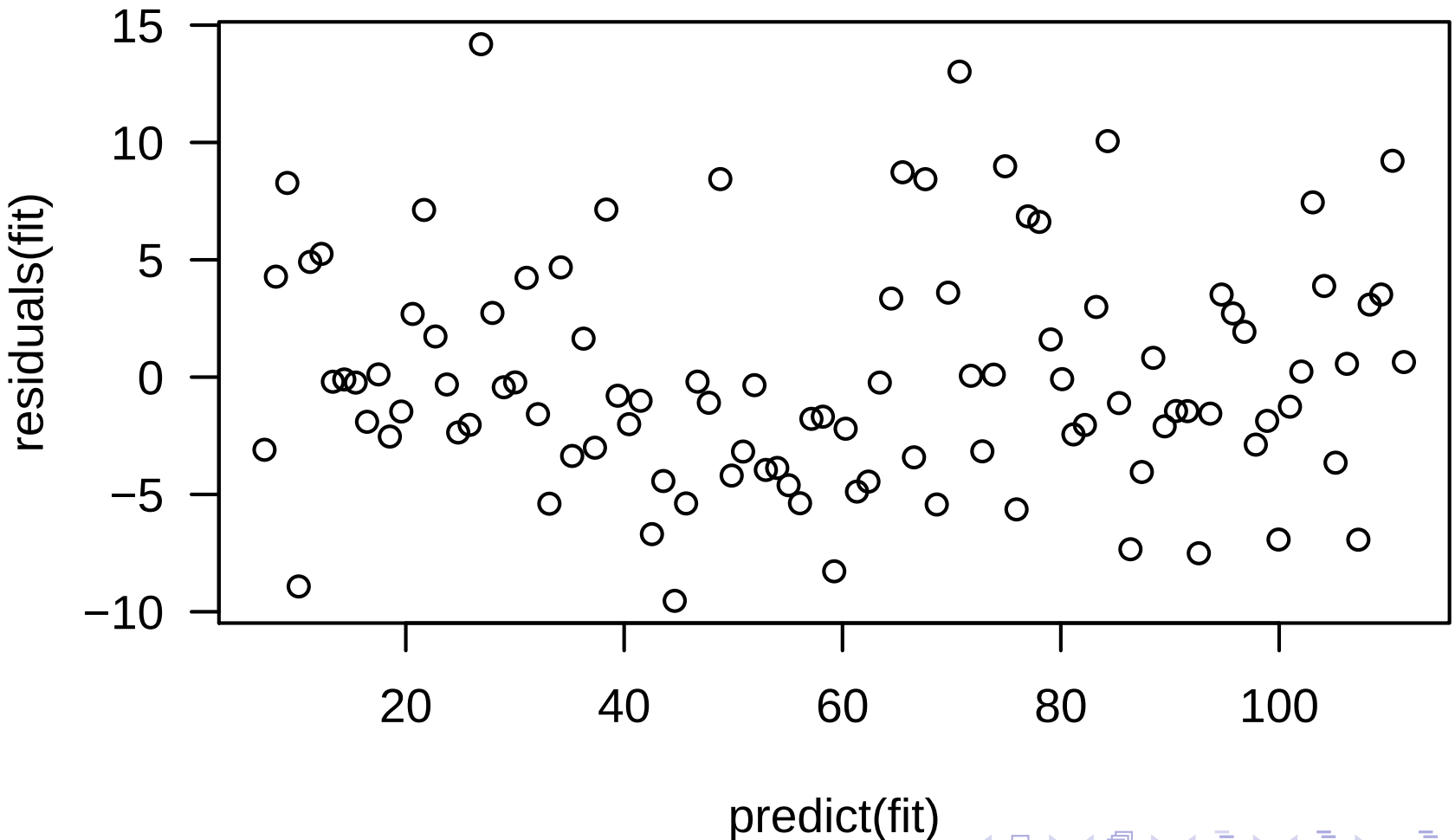
Example 1: Assumptions appear satisfied



Recap from STATS 20X

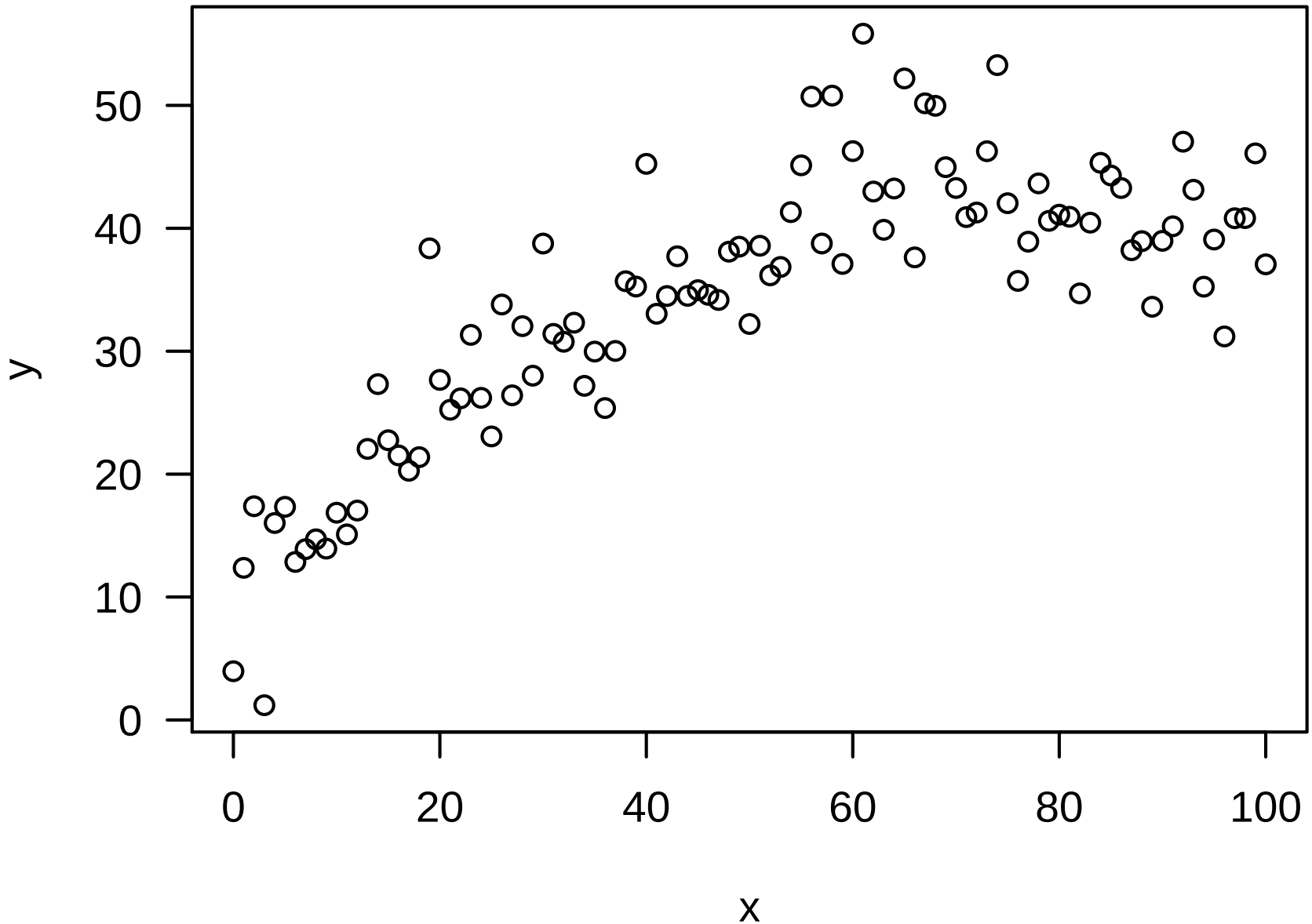
Example 1: Assumptions appear satisfied

```
fit <- lm(y ~ x)
plot(predict(fit), residuals(fit))
```



Recap from STATS 20X

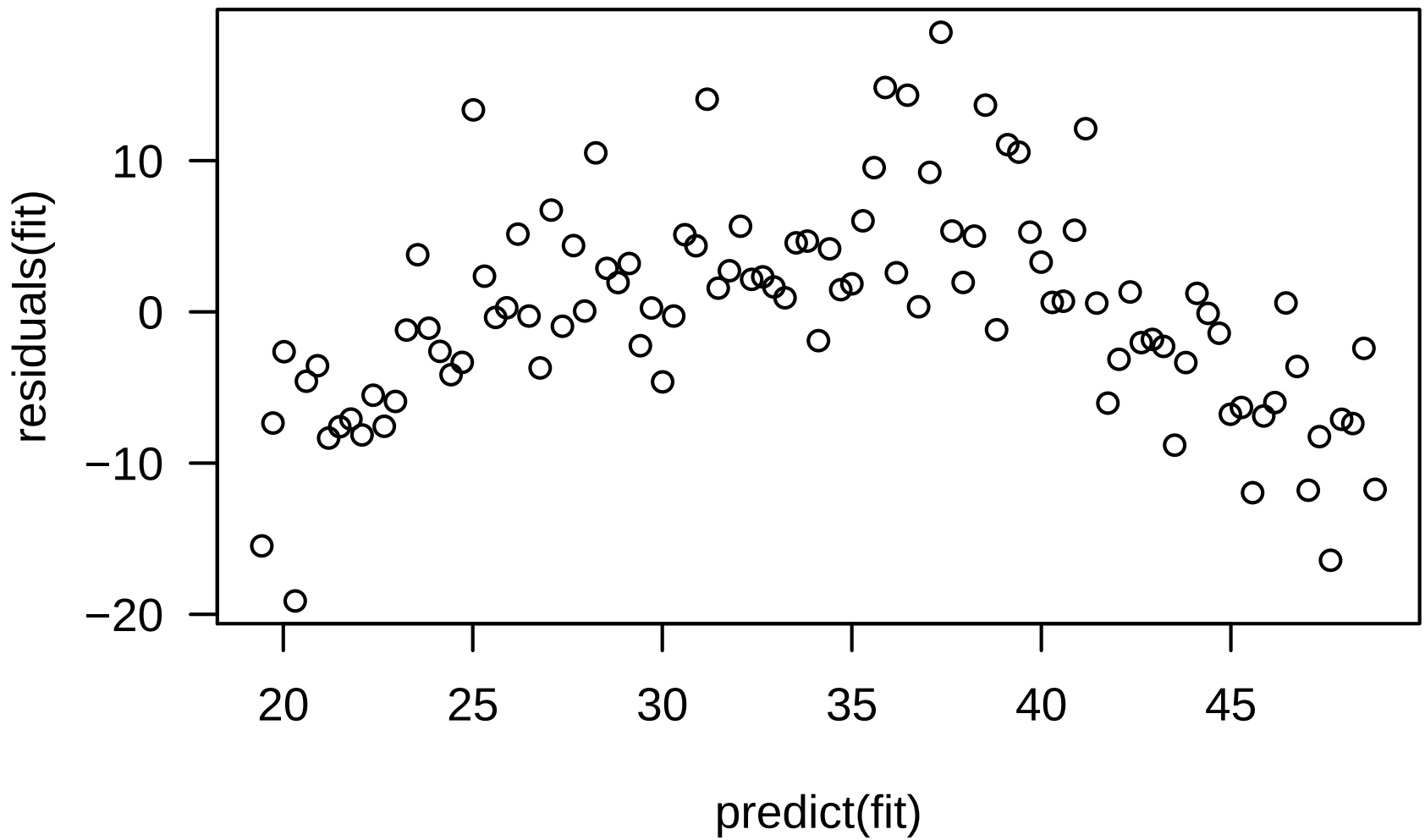
Example 2: Residuals do not all have mean zero



Recap from STATS 20X

Example 2: Residuals do not all have mean zero

```
fit <- lm(y ~ x)
plot(predict(fit), residuals(fit))
```



Recap from STATS 20X

Example 2: Residuals do not all have mean zero

If there is a pattern in the mean of our residuals, we should change the way the explanatory variable(s) are related to the mean of the response variable.

In this case, we could account for curvature by adding a quadratic effect of x by fitting a model that assumes

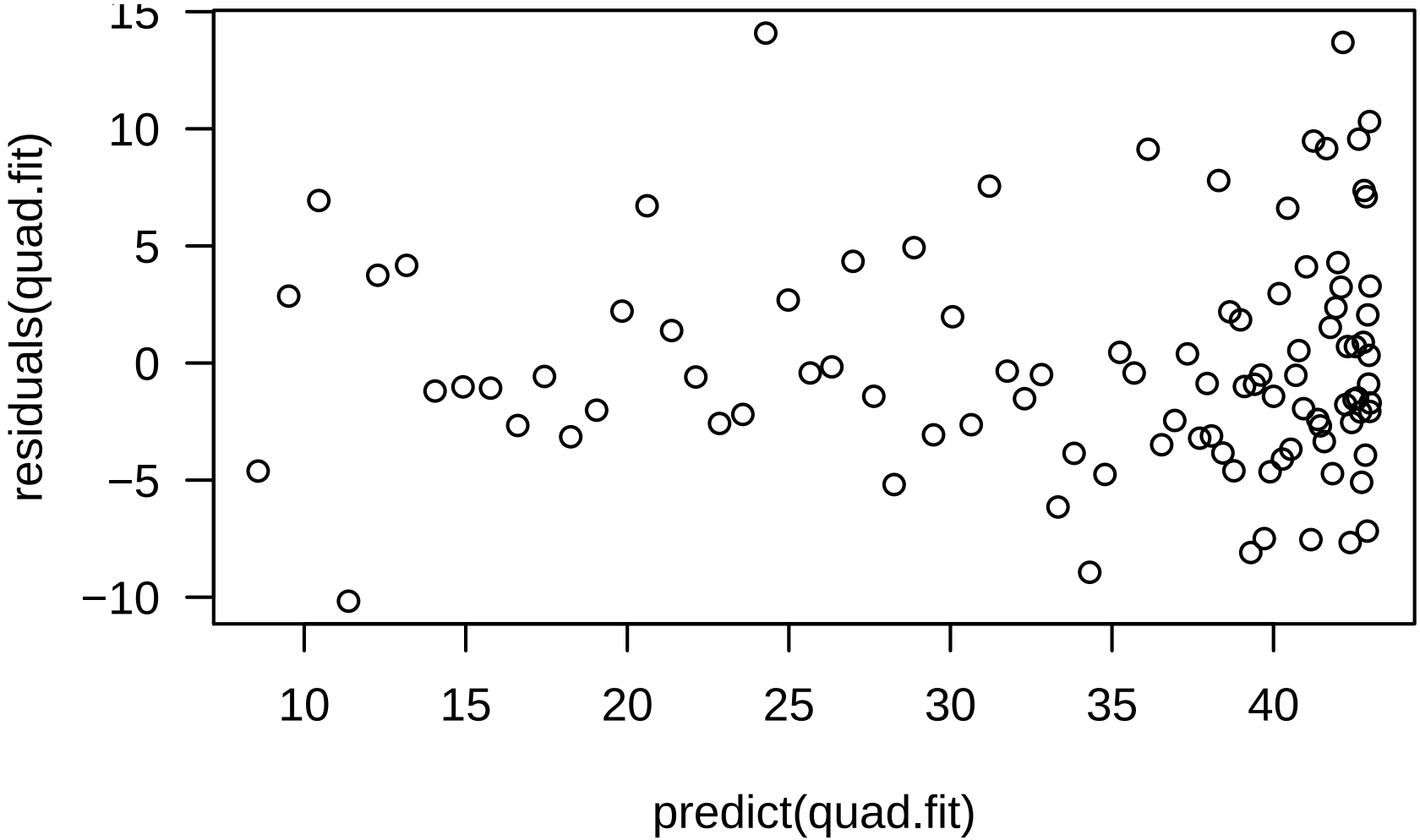
$$\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

The residual plot of the resulting model looks good.

Recap from STATS 20X

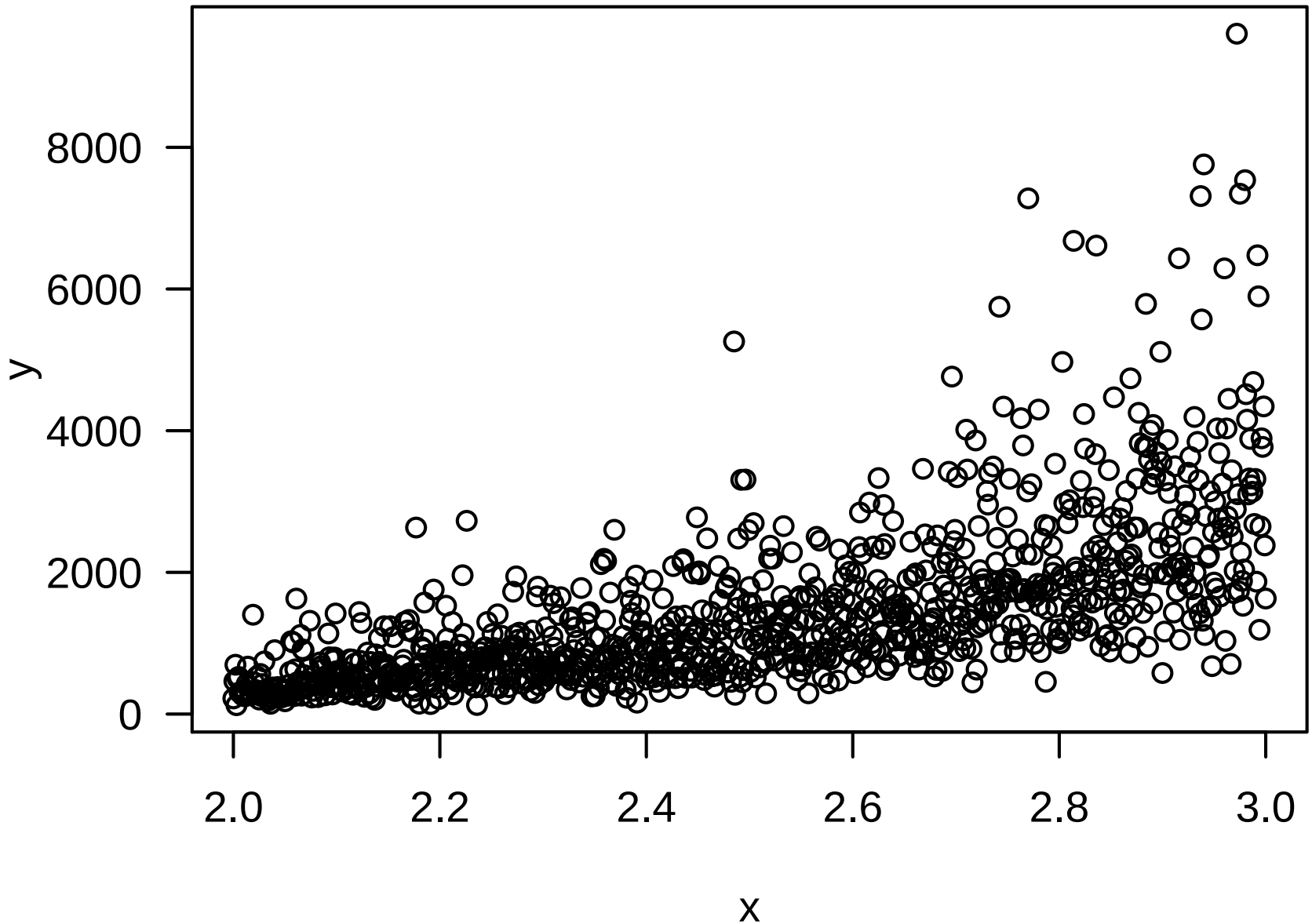
Example 2: Residuals do not all have mean zero

```
quad.fit <- lm(y ~ x + I(x^2))
plot(predict(quad.fit), residuals(quad.fit))
```



Recap from STATS 20X

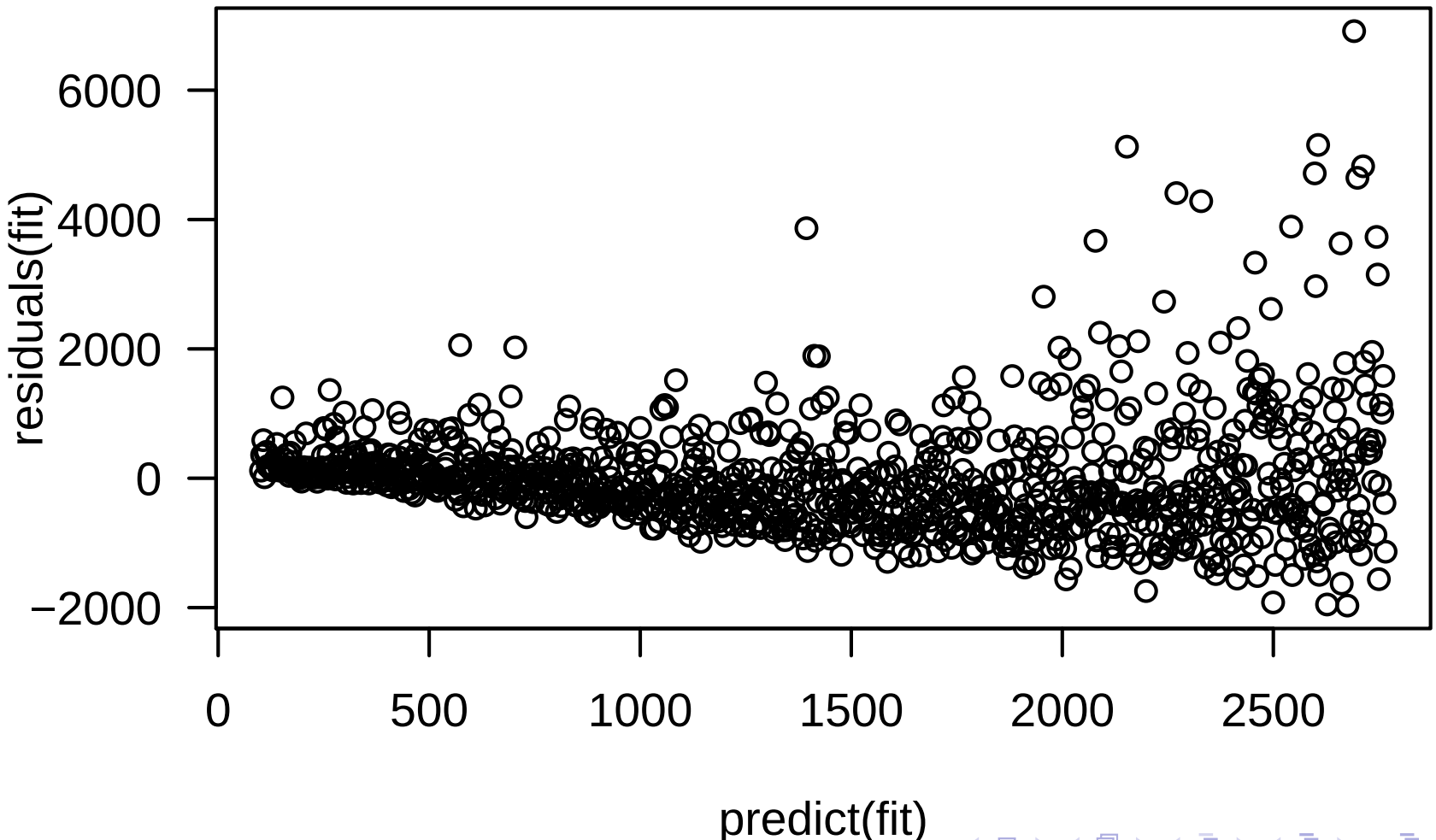
Example 3: Residuals do not have constant variance



Recap from STATS 20X

Example 3: Residuals do not have constant variance

```
fit <- lm(y ~ x)
plot(predict(fit), residuals(fit))
```



Recap from STATS 20X

Example 3: Residuals do not have constant variance

If there is a pattern in the variance of our residuals we have two options:

1. Change the *data*: Transform the response variable so that the resulting model appears to have constant variance
   - See STATS 20X: for continuous y a log-transformation works well if we believe that the relationship between x and y is multiplicative (geometric), and-or
   - the variance increases with the mean.

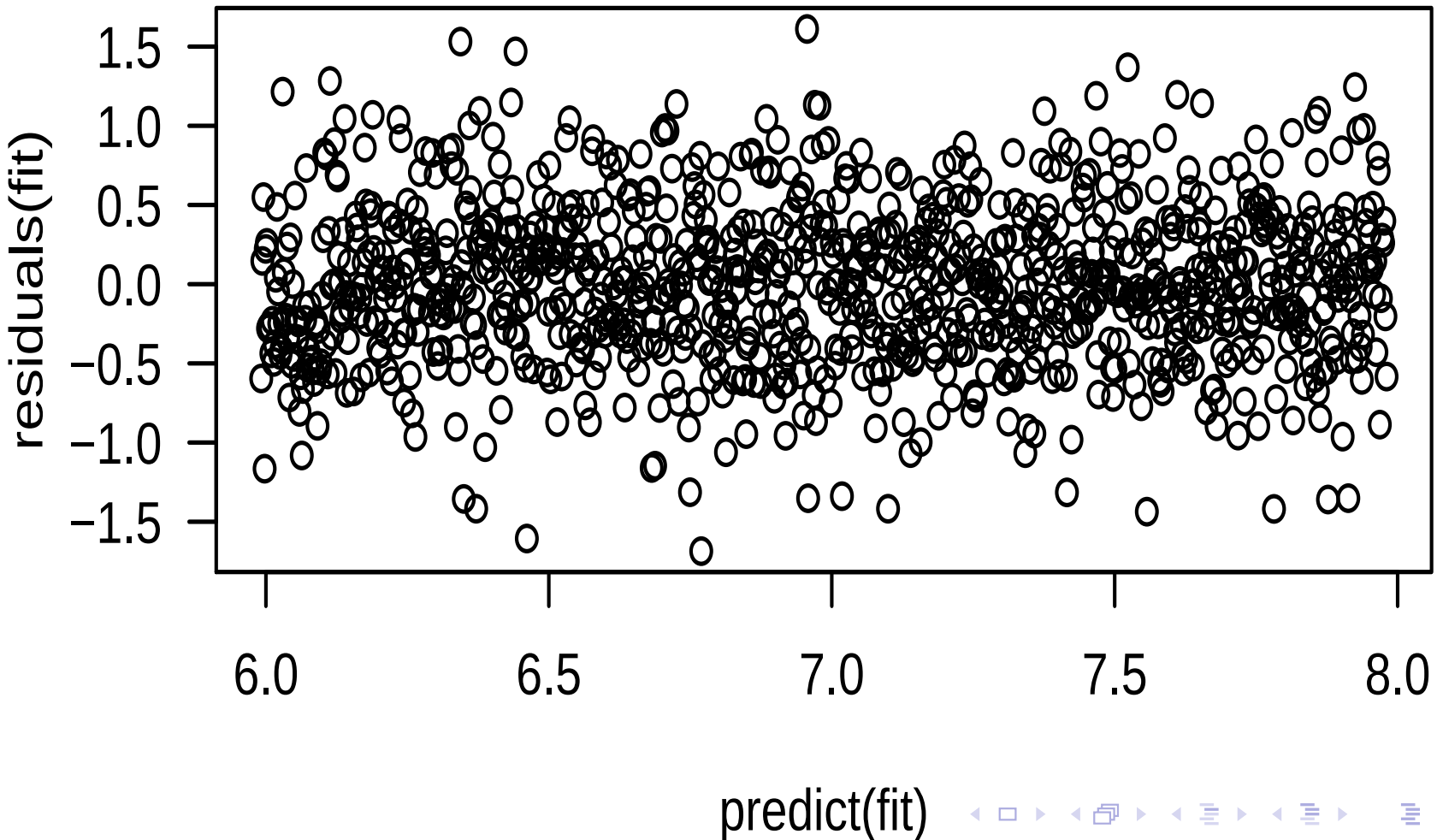
2. Change the *model*: Fit a model that no longer assumes constant variance.
   - A model with a Poisson response for count data.
   - A model with a gamma or inverse-Gaussian distribution for positive, continuous data.

Recap from STATS 20X

Example 3: transform the the data

Solution for STATS20x: fit $\log(y_i) = \beta_0 + \beta_1 x_i + \epsilon_i$.

```
fit <- lm(log(y) ~ x)
plot(predict(fit), residuals(fit))
```



Raw residuals

Poisson regression

The type of residuals we have considered so far are sometimes called *raw* residuals. We will encounter other types of residuals shortly. We can calculate raw residuals for GLMs in the same way. In general:

$$r_i = y_i - \hat{\mathrm{E}}(Y_i),$$

where $\hat{\mathrm{E}}(Y_i)$ is the expected value of the response under our model's estimated parameters.

For a simple Poisson regression model we have

$$\mu_i = \exp(\beta_0 + \beta_1 x_i)$$

We can calculate the ith raw residual as

$$\begin{aligned} r_i &= y_i - \hat{\mathrm{E}}(Y_i) \\ r_i &= y_i - \hat{\mu}_i \\ r_i &= y_i - \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i) \end{aligned}$$

Raw residuals

Logistic regression

For a simple logistic regression model we have

$$p_i = \exp(\beta_0 + \beta_1 x_i)/[1 + \exp(\beta_0 + \beta_1 x_i)]$$

We can calculate the ith raw residual as

$$\begin{aligned} r_i &= y_i - \hat{\mathrm{E}}(Y_i) \\ r_i &= y_i - n_i \hat{p}_i \\ r_i &= y_i - n_i \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i)/[1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i)] \end{aligned}$$

This is because the expected value of a Binomial(n, p) distribution is given by $\mathrm{E}(Y) = np$.

- For example, flipping a fair coin has probability $p = 0.5$ of resulting in a head. If we flip it $n = 10$ times, on average we obtain $np = 10 \times 0.5 = 5$ heads.

Raw residuals

Raw residuals are the difference between the observed response and its expected value.

For any model, we expect raw residuals to come from a distribution with mean zero. However, the property of raw residuals having approximately constant variance does *not* hold in general.

In other words, for Poisson and logistic regression models, raw residuals will *not* have approximately constant variance, even if the model is correct. This is because the response itself does not have constant variance.

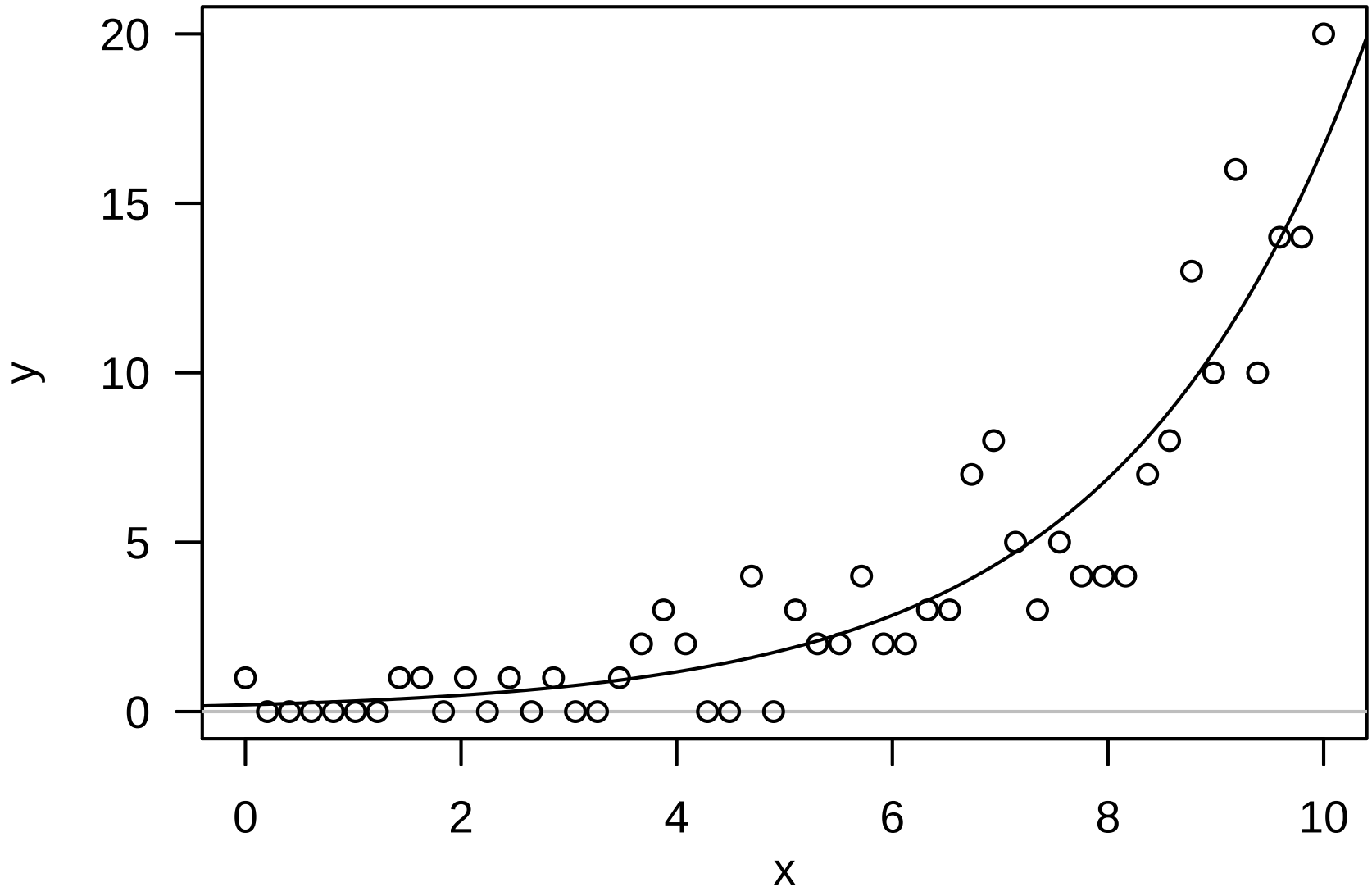
Note that, for GLMs, we typically plot fitted values on their *link* scale. We plot

- residuals against $\log(\hat{\mu}_i)$ for Poisson regression, and
- residuals against $\text{logit}(\hat{p}_i)$ for logistic regression.

Raw residuals

Poisson regression

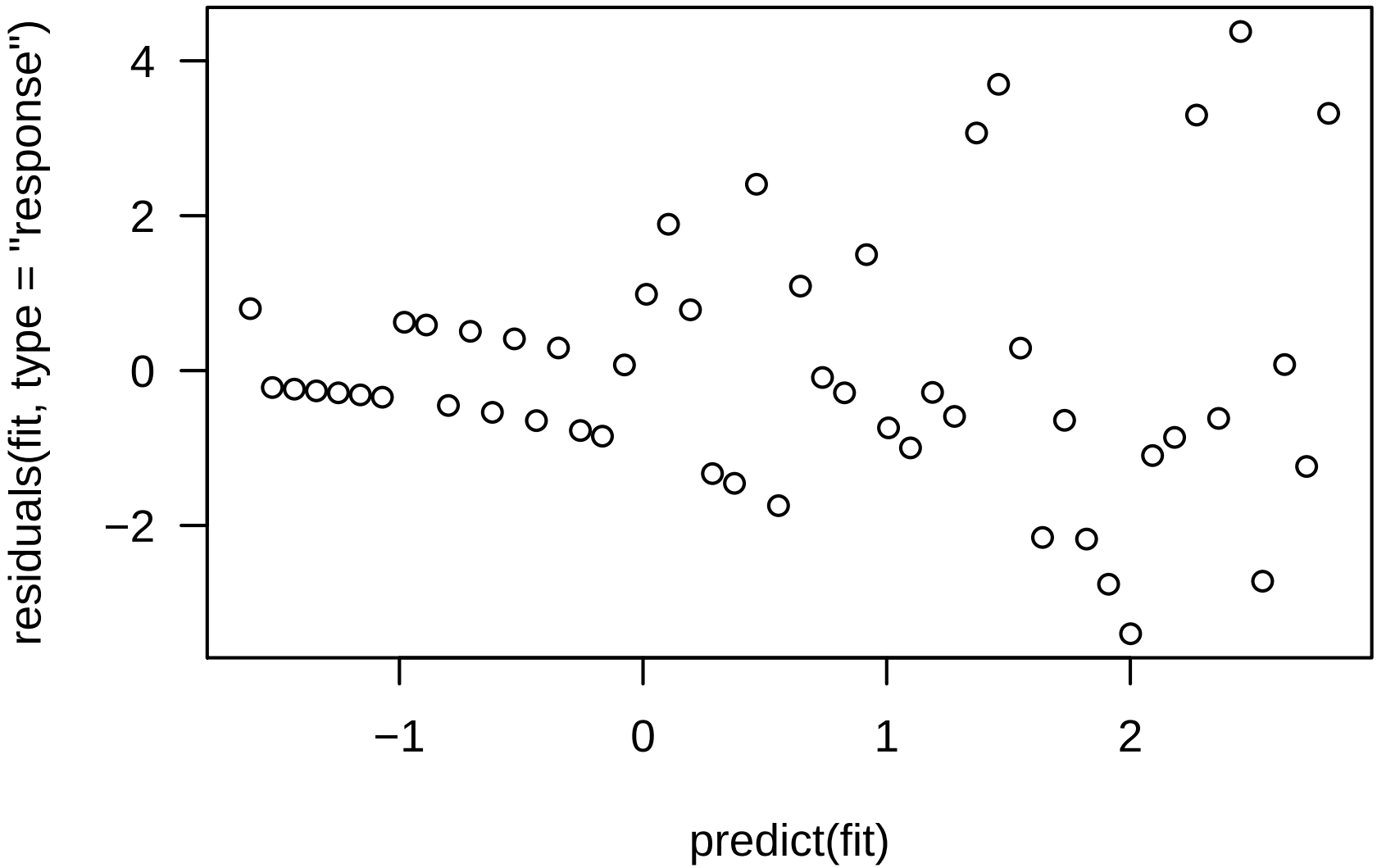
Response variance increases with expected value...



Raw residuals

Poisson regression

... So raw residual variance also increases with expected value.



Pearson residuals

We need an improved type of residual that allows us to tell how well we have modelled the variance of the response.

Pearson residuals are a type of residual that are standardised to have constant variance:

$$r_i = \frac{y_i - \hat{\mathrm{E}}(Y_i)}{\sqrt{\widehat{\mathrm{Var}}(Y_i)}} = \frac{y_i - \hat{\mathrm{E}}(Y_i)}{\widehat{\mathrm{SD}}(Y_i)},$$

where $\widehat{\mathrm{Var}}(Y_i)$ is the variance and $\widehat{\mathrm{SD}}(Y_i)$ is the standard deviation of the response, under the fitted model.

So, unlike raw residuals, if the model is correct then we expect the Pearson residuals to have approximately constant variance.

Pearson residuals

1. For a linear regression model:

$$r_i = \frac{y_i - \hat{\mu}_i}{\hat{\sigma}}$$

2. For a Poisson regression model:

$$r_i = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}}$$

   - Because the variance of the Poisson distribution is μ_i.

3. For a logistic regression model:

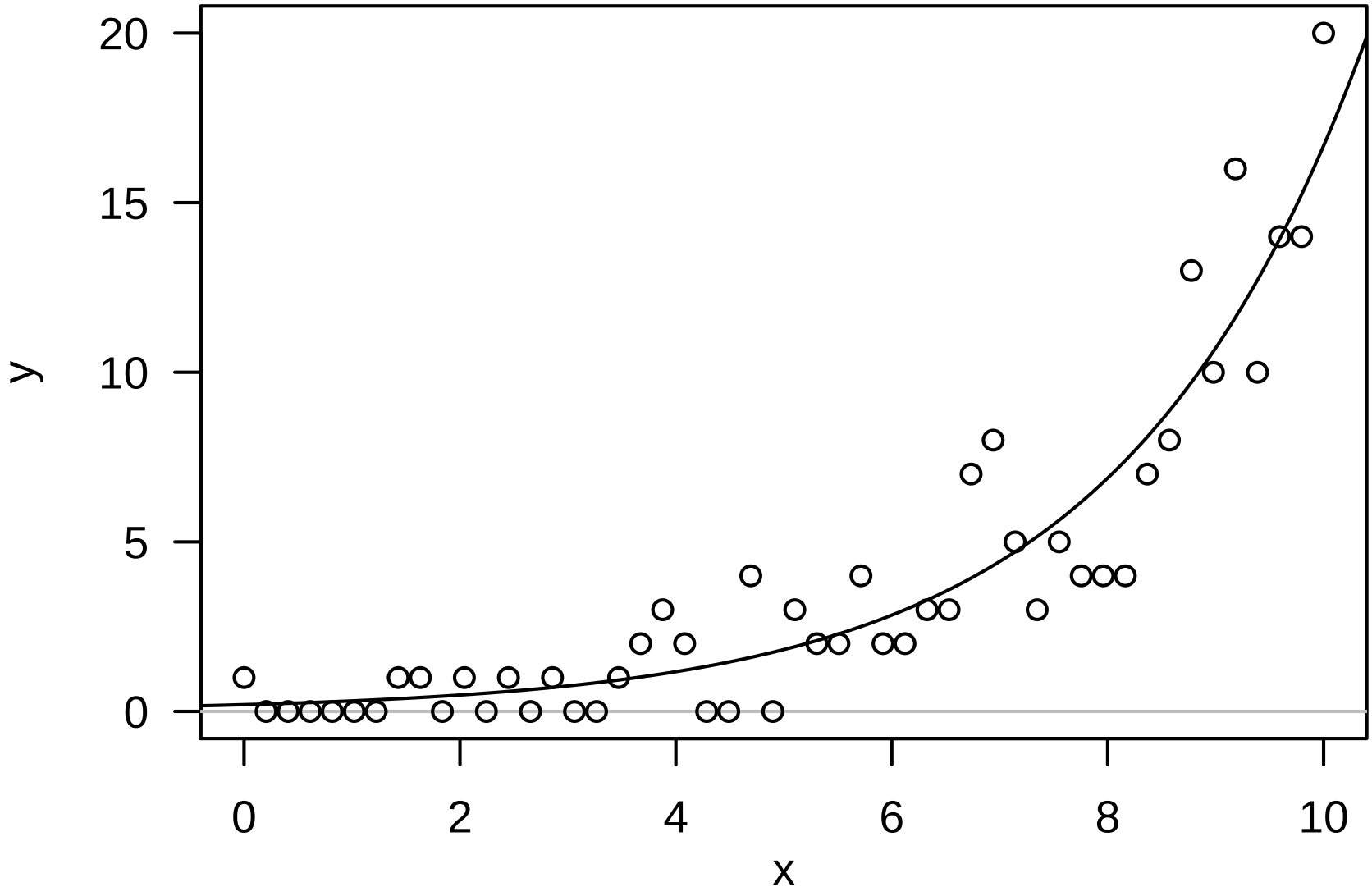
$$r_i = \frac{y_i - n_i \hat{p}_i}{\sqrt{n_i \hat{p}_i (1 - \hat{p}_i)}}$$

   - Because the variance of the Binomial distribution is $n_i p_i (1 - p_i)$.

Pearson residuals

Poisson regression

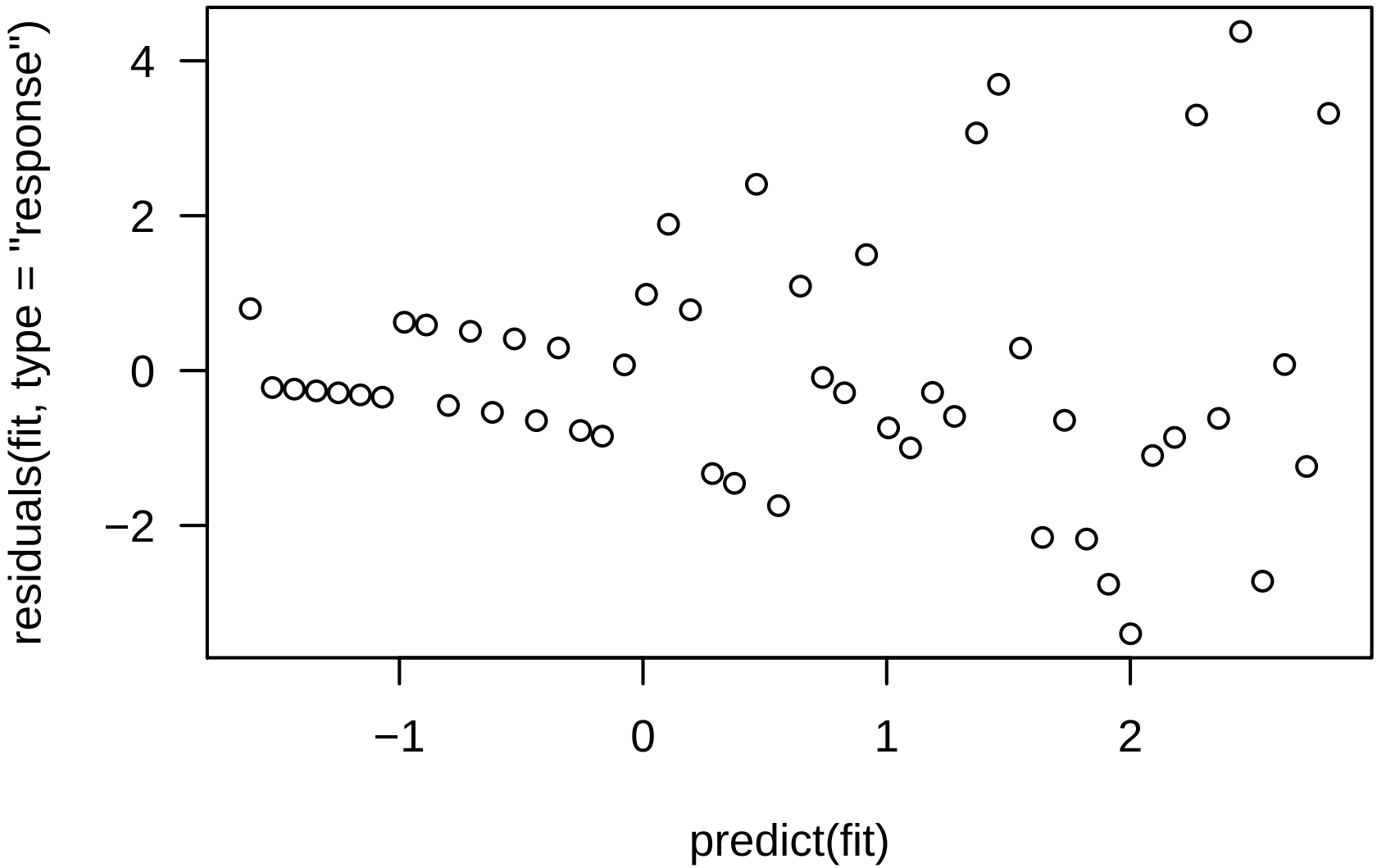
Response variance increases with expected value...



Pearson residuals

Poisson regression

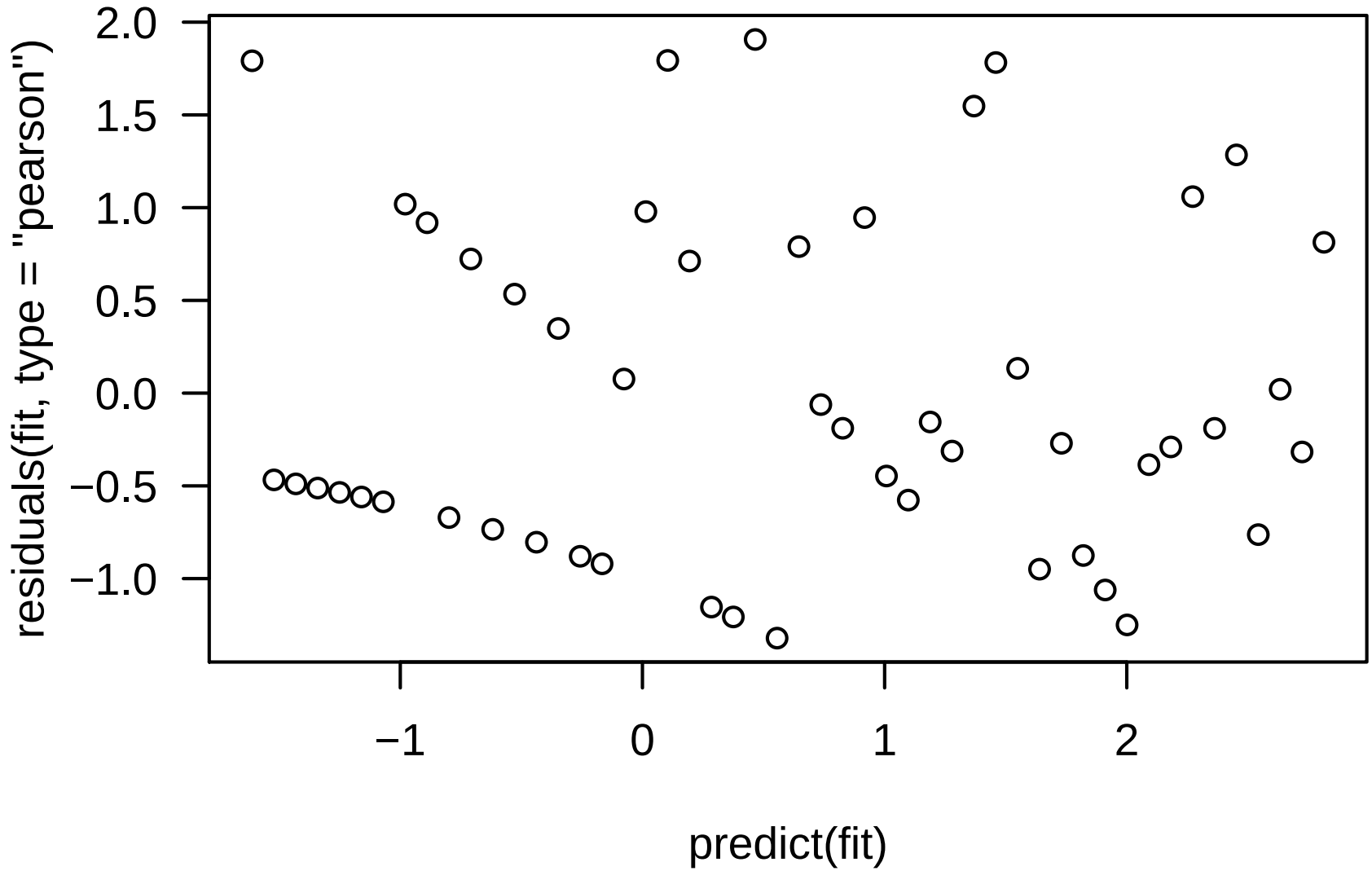
... So raw residual variance also increases with expected value...



Pearson residuals

Poisson regression

... But Pearson residual variance does not.



Pearson residuals

Properties

If the model is appropriate, Pearson residuals will

- have mean zero across the range of fitted values,
- have approximately constant variance, and
- have an approximate standard normal distribution, $\mathrm{N}(0, 1)$, for observations that are not 'sparse': those with
  - A large expectation ($\hat{\mu}_i \geq 5$) for Poisson regression, or
  - Large n for logistic regression:
    - When p_i is close to 0.5, $n_i \geq 5$ is probably sufficient.
    - But if p_i is close to 0 or 1, n_i must be much larger.

So, for observations that are not sparse, we are looking for a residual plot that looks like a patternless band around zero—just as we normally would for a linear regression model. However, we may observe apparent patterns for sparse data.

Deviance residuals

Deviance residuals are another type of residuals for GLMs.

Recall parameter estimation from Handout 3:

- For a linear model, we estimate model parameters by minimising the residual sum of squares, and
- For a GLM, we estimate model parameters by maximising the log-likelihood.

Also recall from Handout 4 that minimising the deviance of a GLM is equivalent to maximising the log-likelihood.

Residuals in a linear model

For a linear model, we estimate model parameters by minimising the residual sum of squares:

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

So, the ith observation contributes $(y_i - \hat{\mu}_i)^2$ to this sum.

We can think of the raw residual for the ith observation as the signed square root of this contribution:

$$r_i = \begin{cases} +\sqrt{(y_i - \hat{\mu}_i)^2} & \text{If } y_i \geq \hat{\mu}_i \\ -\sqrt{(y_i - \hat{\mu}_i)^2} & \text{If } y_i < \hat{\mu}_i \end{cases}$$

$$r_i = y_i - \hat{\mu}_i$$

Deviance residuals

For a GLM, we estimate model parameters by minimising the deviance:

$$\mathrm{D} = \sum_{i=1}^{n} 2\left\{\log[f(y_i;\boldsymbol{\beta}_S)] - \log[f(y_i;\hat{\boldsymbol{\beta}})]\right\},$$

where $\boldsymbol{\beta}_S$ are parameters of the saturated model. So, the ith observation contributes

$$2\left\{\log[f(y_i;\boldsymbol{\beta}_S)] - \log[f(y_i;\hat{\boldsymbol{\beta}})]\right\}$$

to this sum.

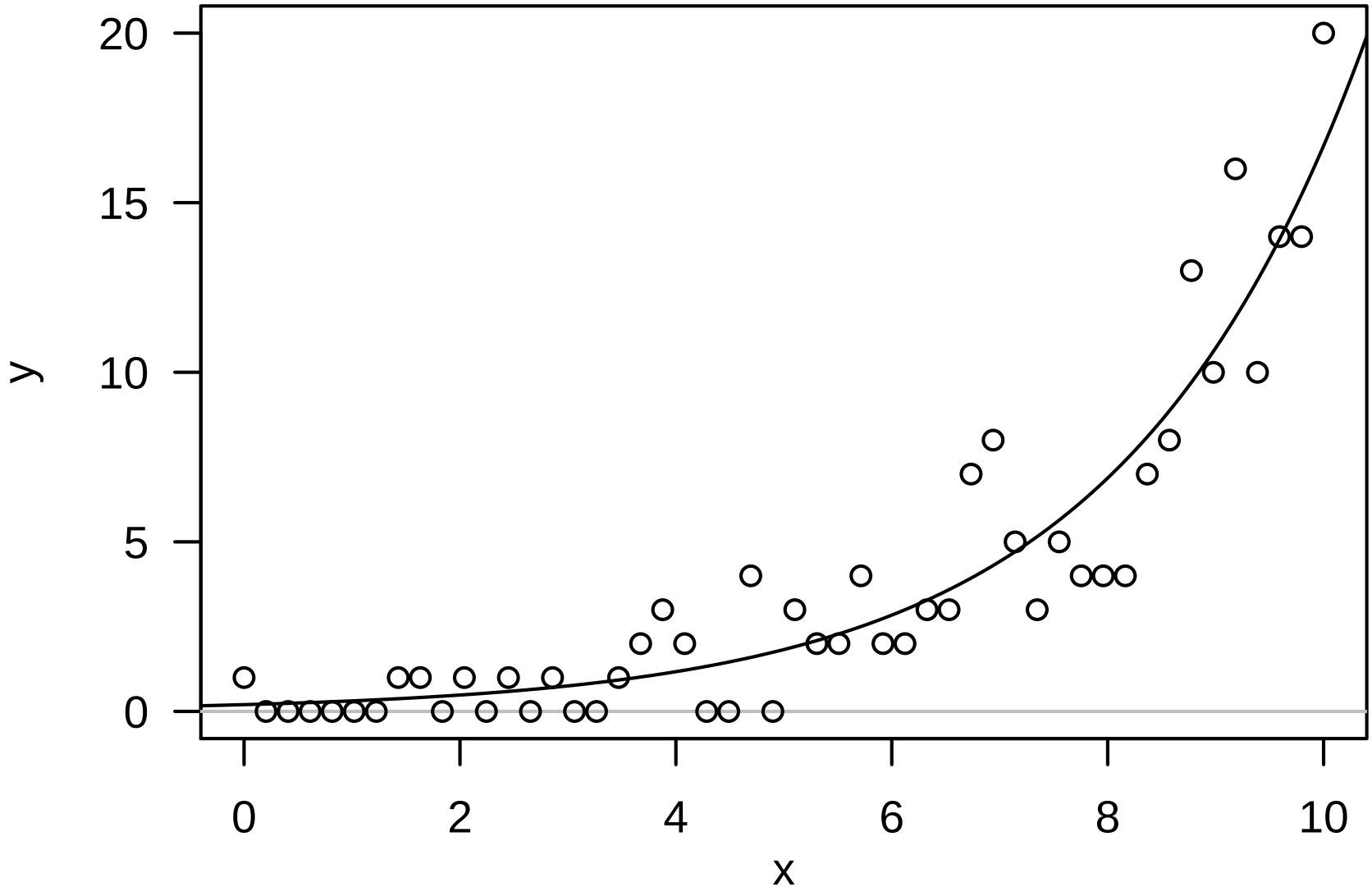
We can think of the deviance residual for the ith observation as the signed square root of this contribution:

$$r_i = \begin{cases} +\sqrt{2\left\{\log[f(y_i;\boldsymbol{\beta}_S)] - \log[f(y_i;\hat{\boldsymbol{\beta}})]\right\}} & \text{If } y_i \geq \hat{\mathrm{E}}(Y_i) \\ -\sqrt{2\left\{\log[f(y_i;\boldsymbol{\beta}_S)] - \log[f(y_i;\hat{\boldsymbol{\beta}})]\right\}} & \text{If } y_i < \hat{\mathrm{E}}(Y_i) \end{cases}$$

Deviance residuals

Poisson regression

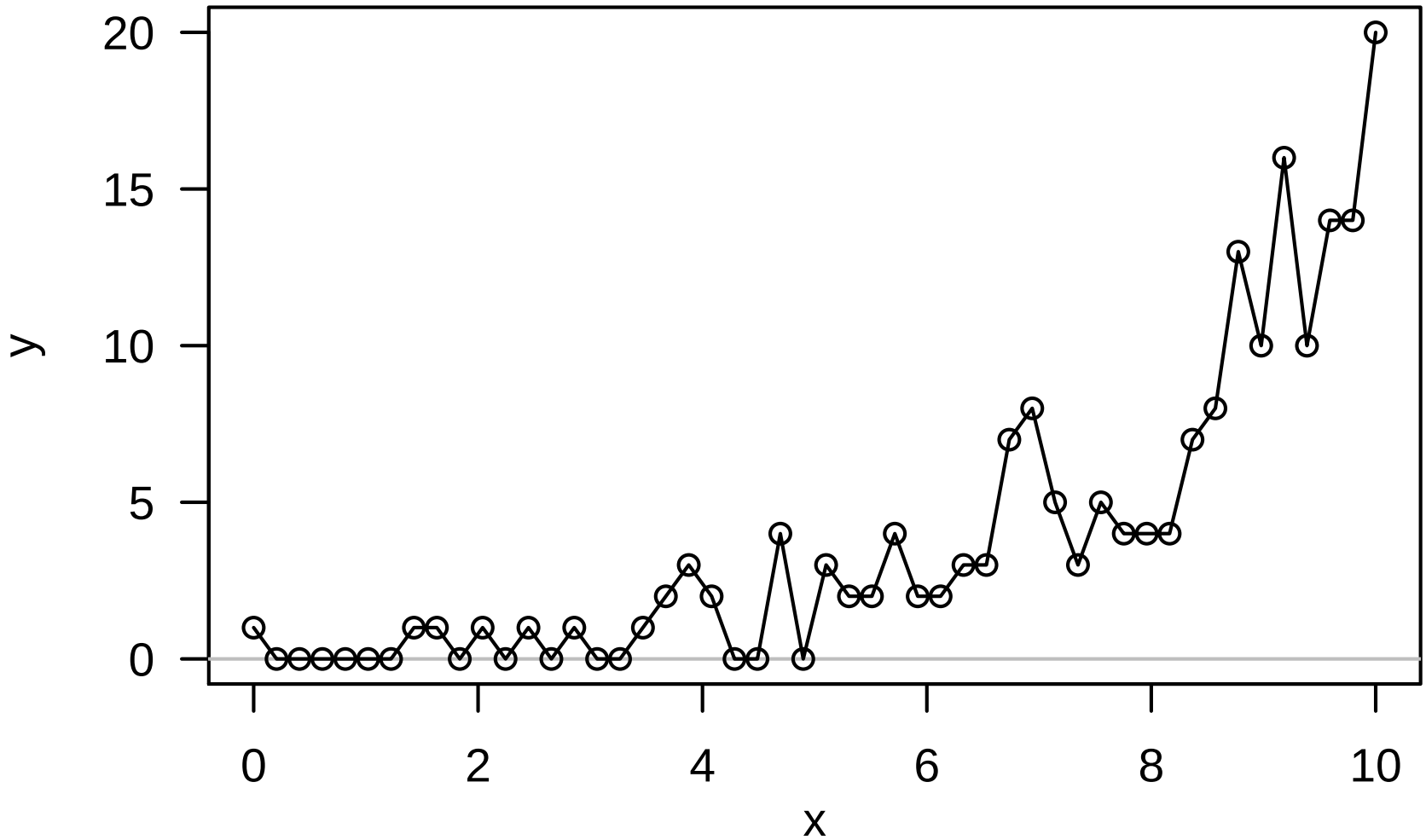
Sounds complicated! Let's look at an example to clarify.



Deviance residuals

Poisson regression: Saturated model

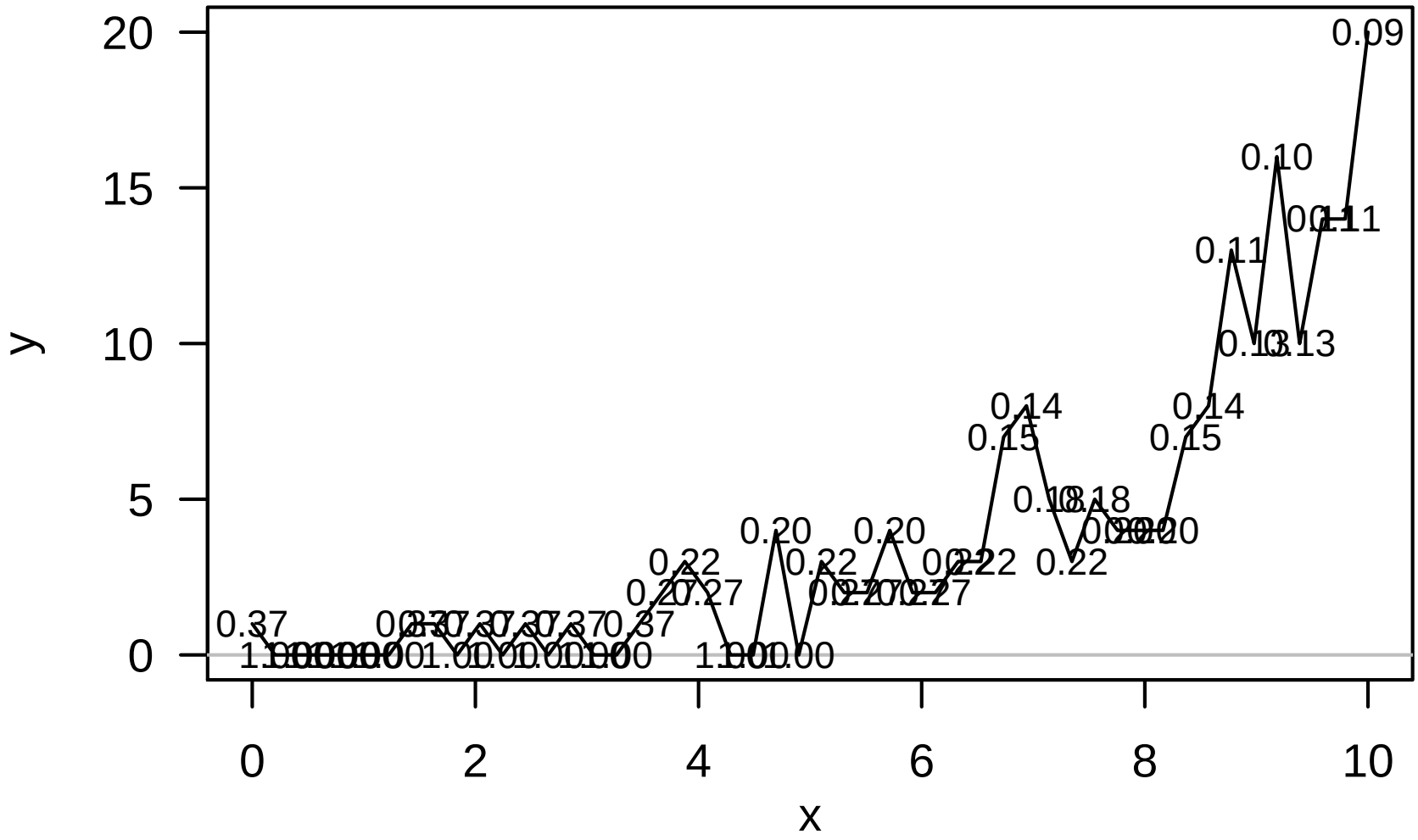
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Probabilities, saturated model

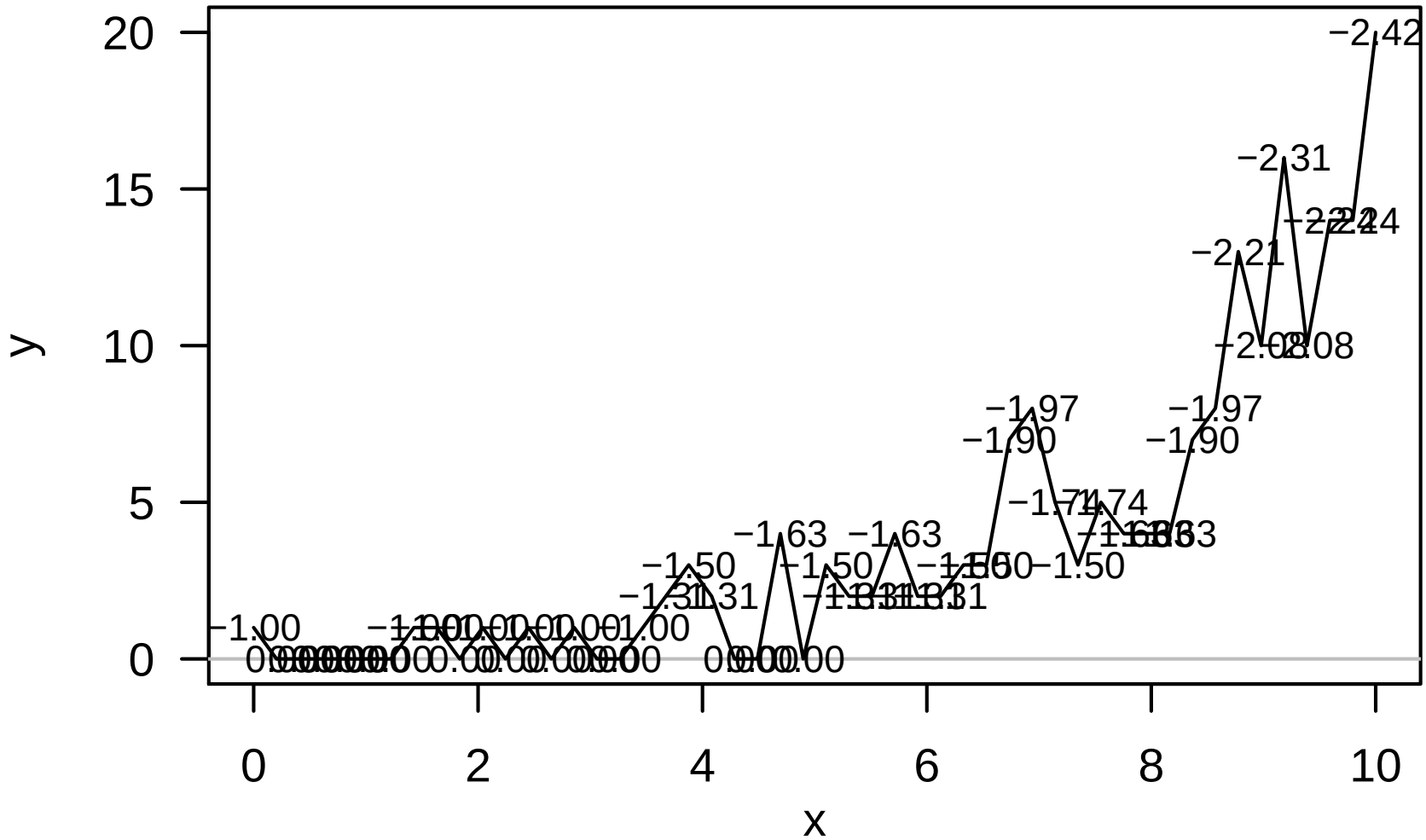
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Log-probabilities, saturated model

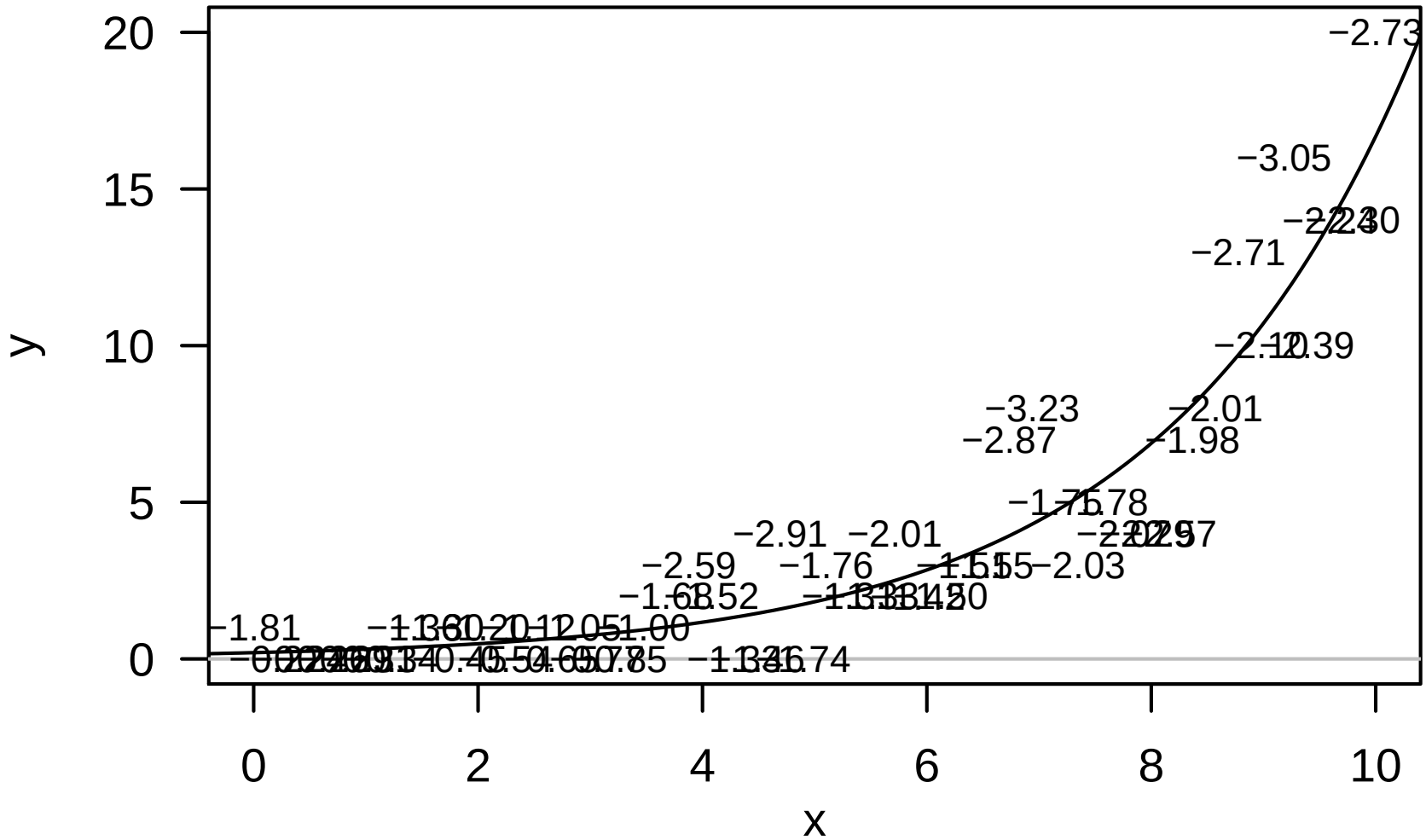
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Log-probabilities, fitted model

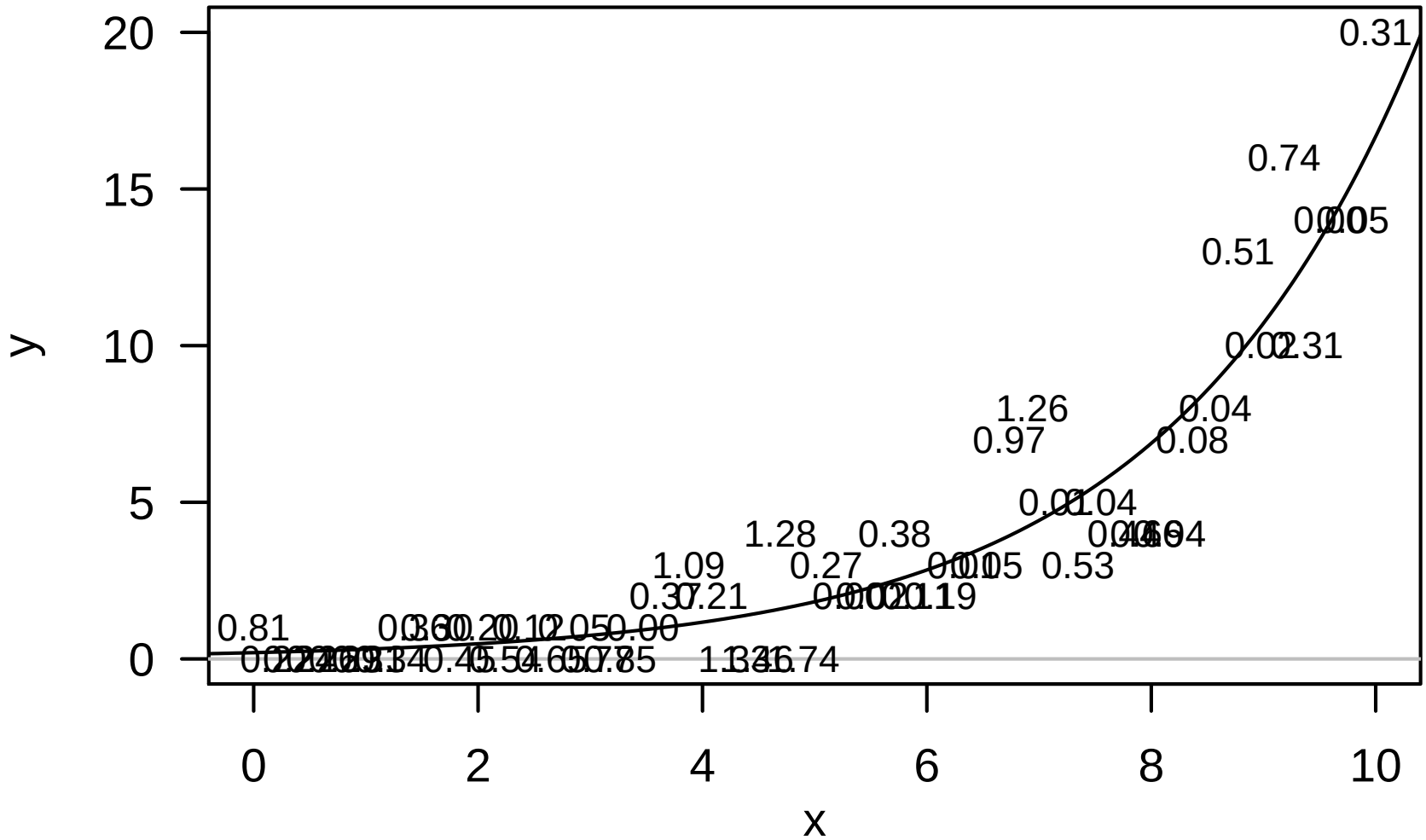
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Take the difference

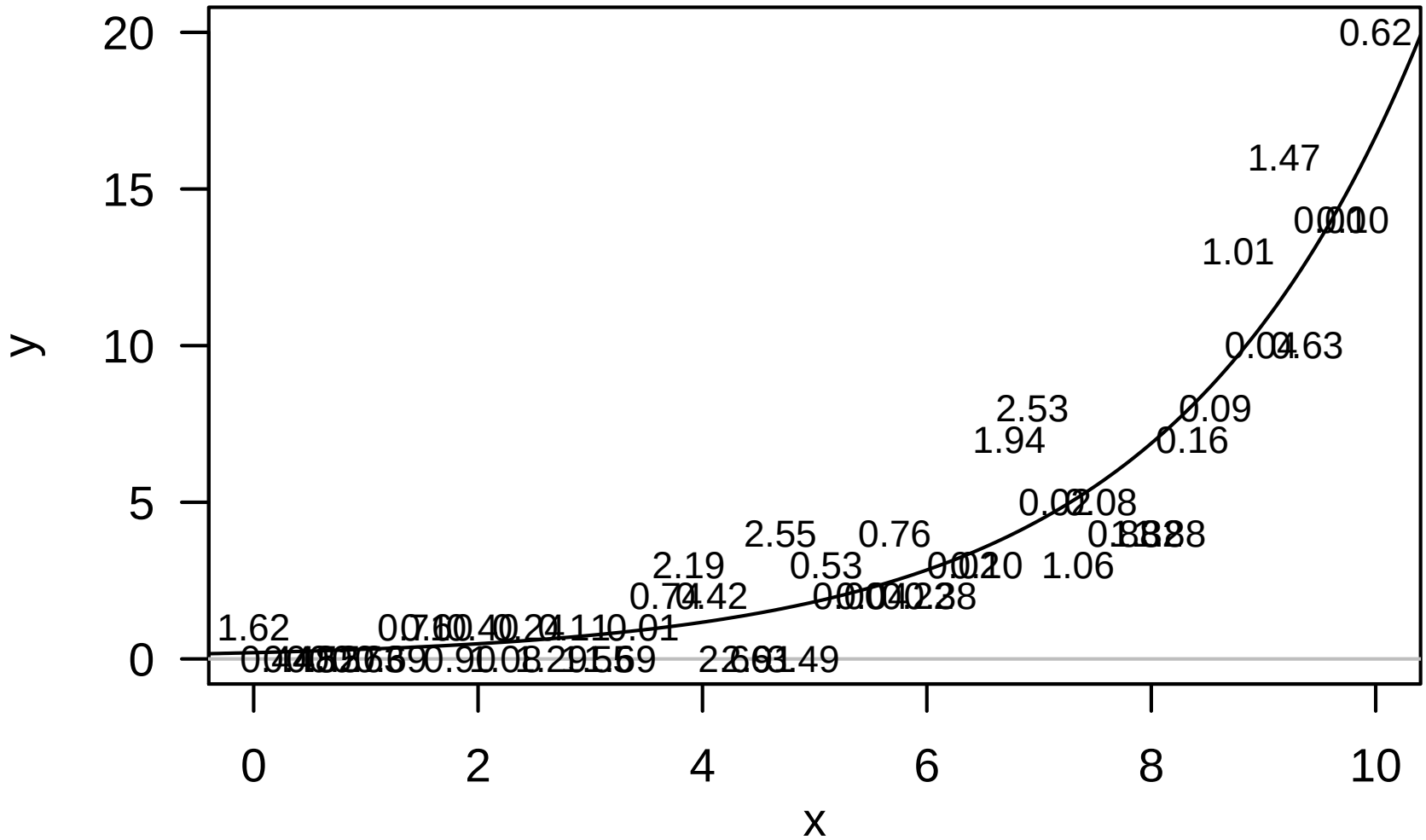
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Double the difference

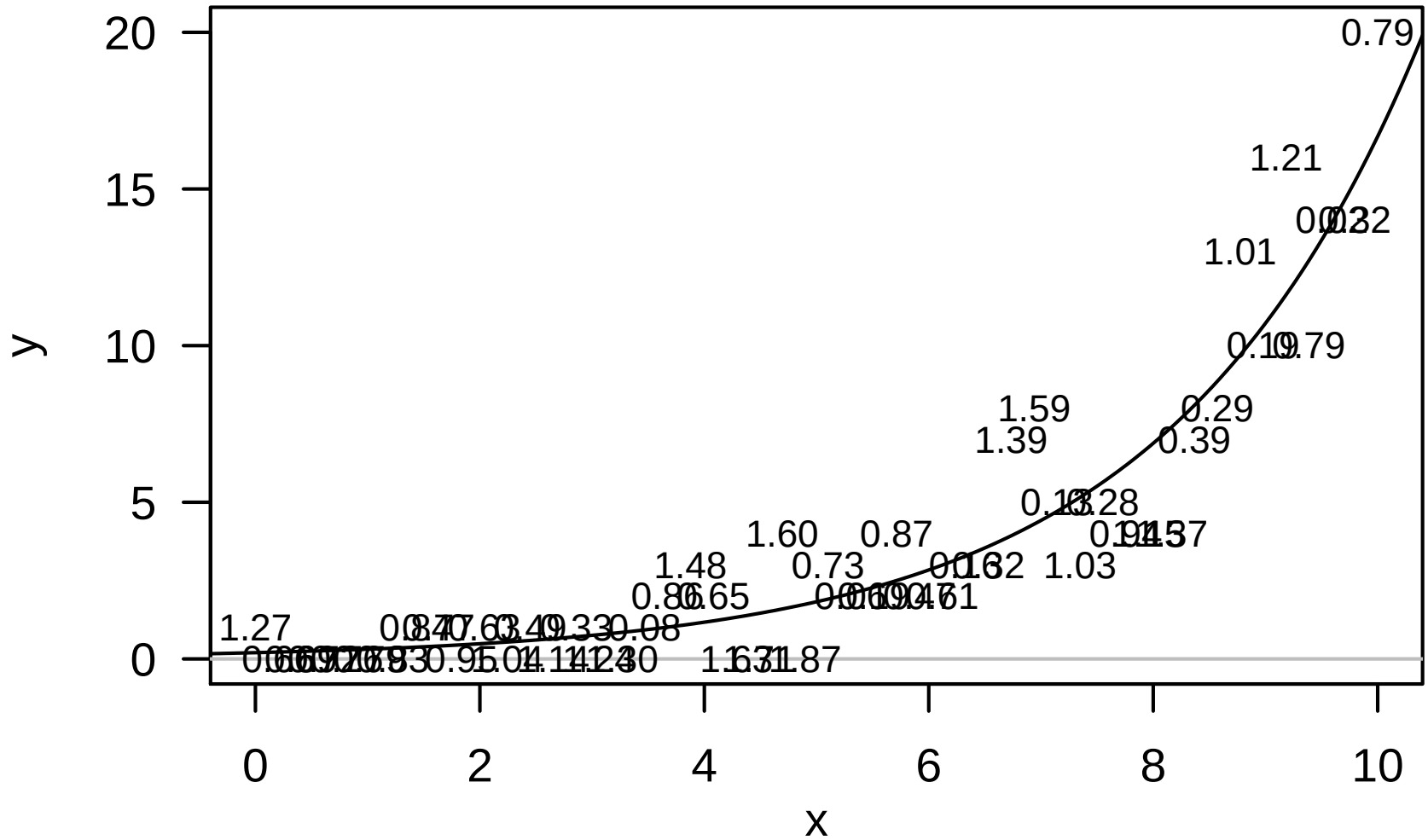
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Take the square-root

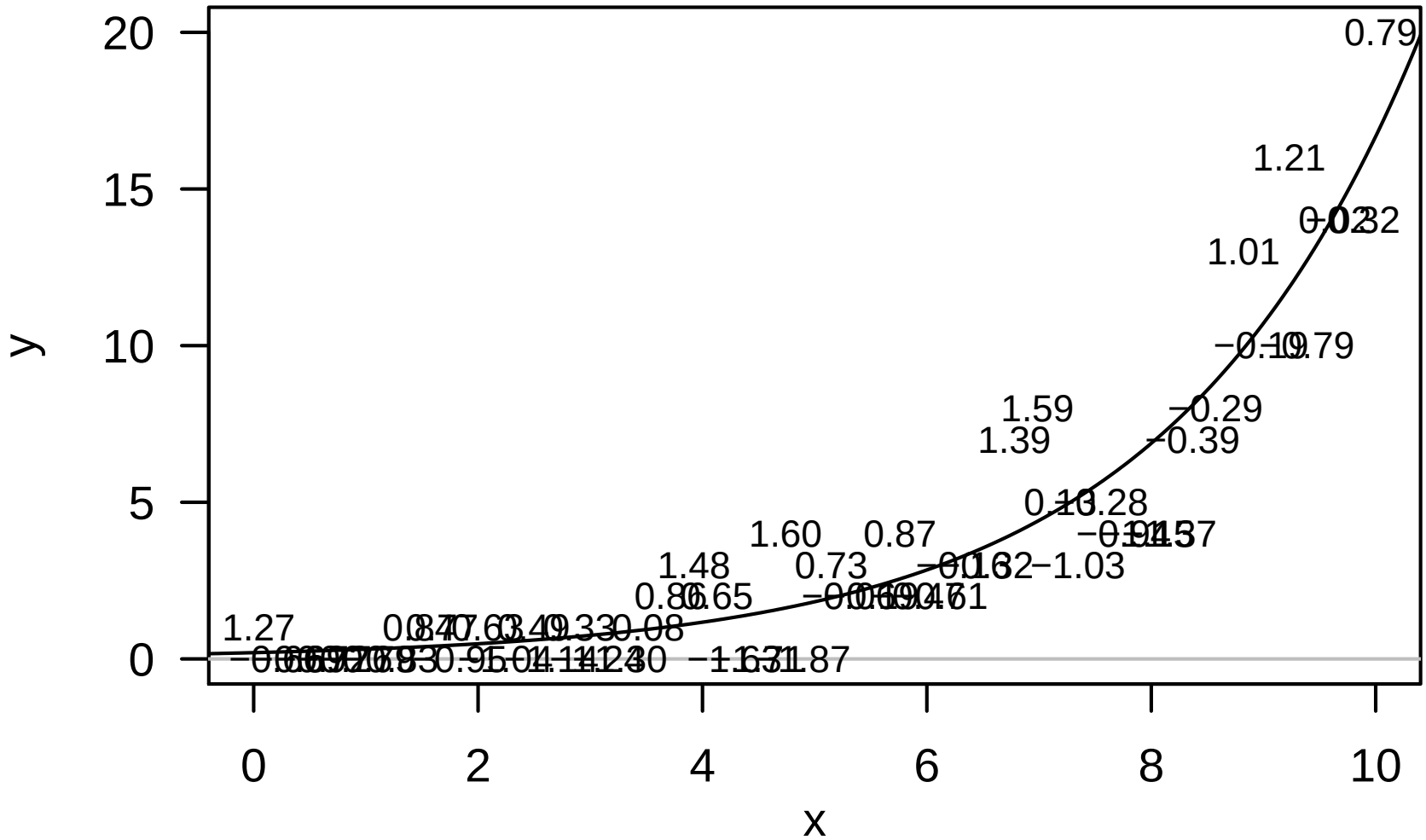
$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\hat{\beta}_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

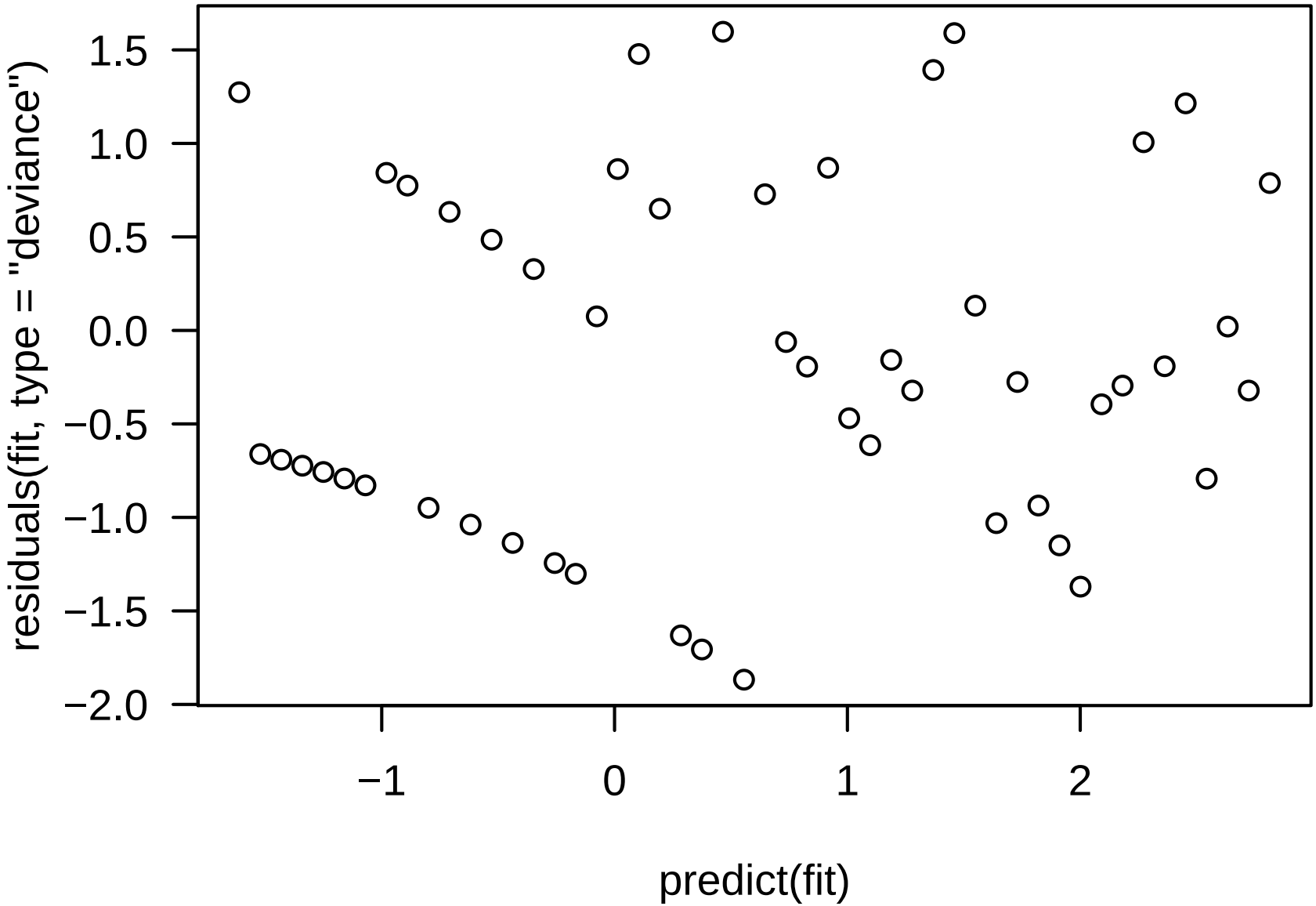
Poisson regression: Apply the appropriate sign

$$r_i = \pm\sqrt{2\left\{\log[f(y_i;\beta_S)] - \log[f(y_i;\hat{\beta})]\right\}}$$



Deviance residuals

Poisson regression: Residual plot



Deviance residuals

Comparison with RSS

This is another example of the link between residual sum of squares in linear regression models and deviance in GLMs:

1. Linear regression models:
   - We estimate the parameters by minimising the RSS
   - The RSS is the sum of the squared residuals

2. GLMs:
   - We estimate the parameters by minimising the deviance
   - The deviance is the sum of the squared deviance residuals

Deviance residuals

Properties

From the plots, deviance residuals have the properties we're after:

- Points near the regression line have a residual close to zero.
- Points far above the line have large positive residuals.
- Points far below the line have large negative residuals.
- Residuals appear to have approximately constant variance.

In fact, deviance residuals have roughly the same properties as Pearson residuals.

Deviance residuals

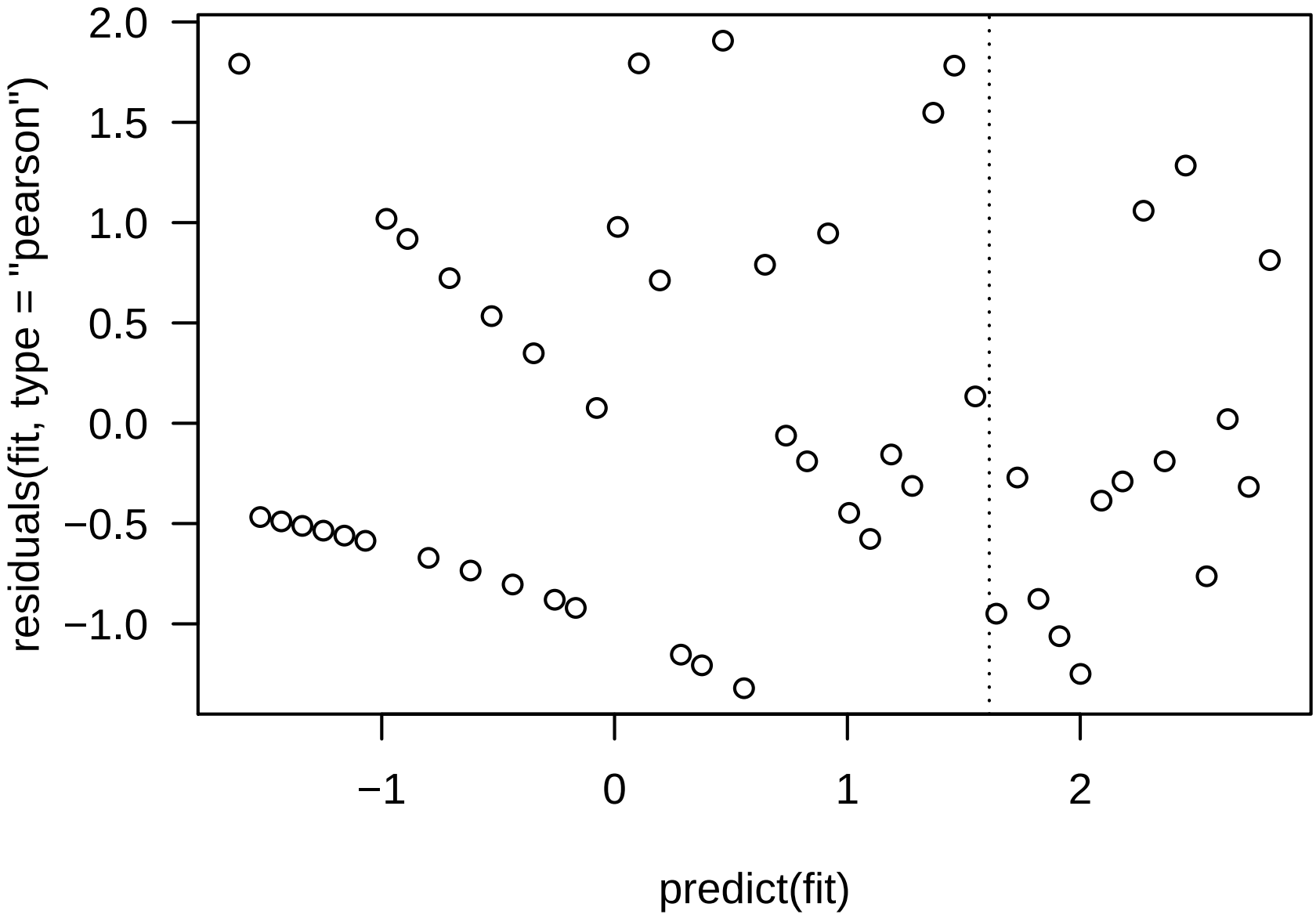
Properties

If the model is appropriate, *both Pearson and deviance residuals* will

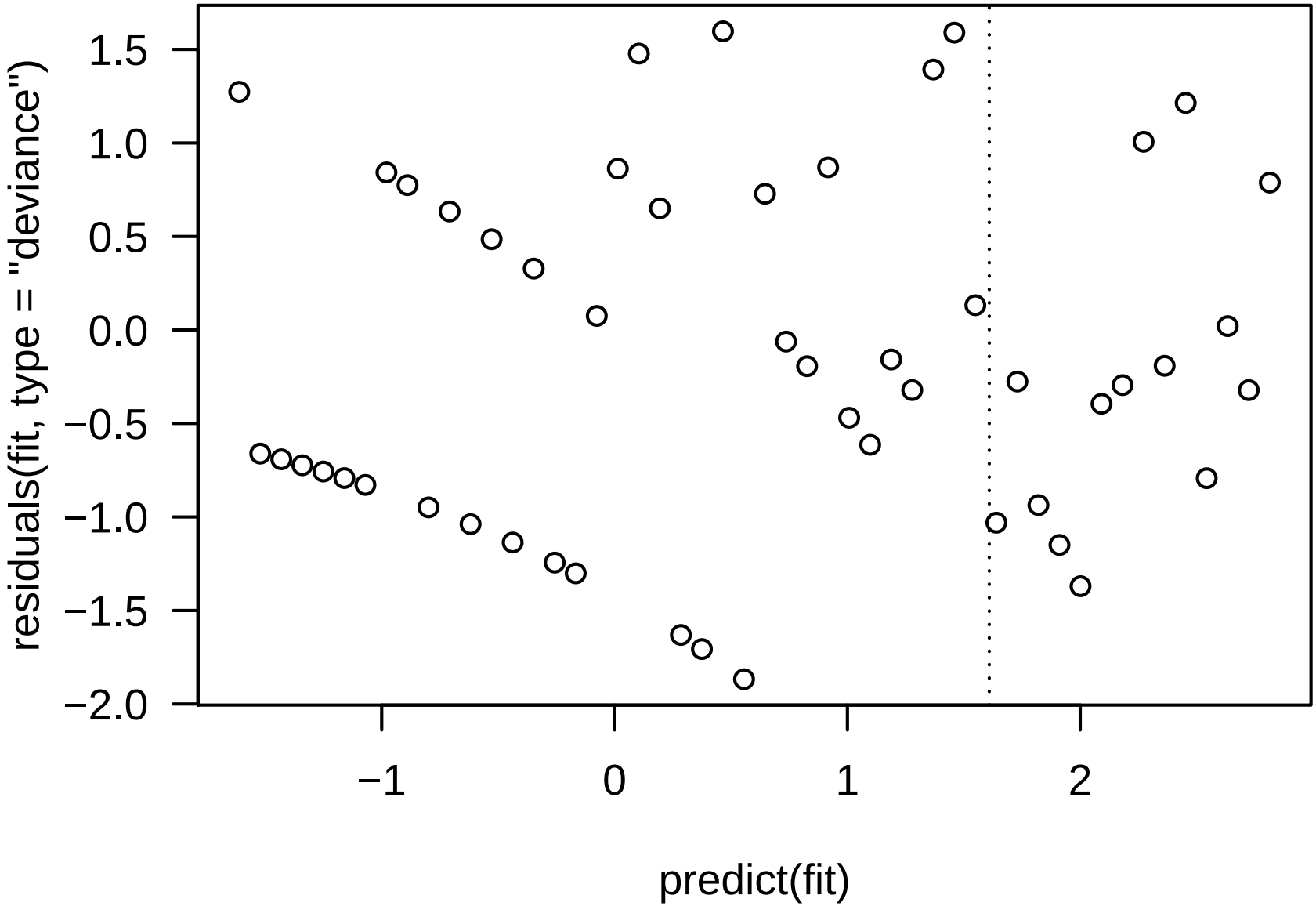
- have mean zero across the range of fitted values,
- have approximately constant variance, and
- have an approximate standard normal distribution, $N(0, 1)$, for observations with
  - A large expectation ($\mu_i \geq 5$) for Poisson regression, or
  - Large n_i for logistic regression:
    - When p_i is close to 0.5, $n_i \geq 5$ is probably sufficient.
    - But if p_i is close to 0 or 1, n_i must be much larger.

So, for observations falling into the categories above, we are looking for a residual plot that looks like a patternless band around zero—just as we normally would for a linear regression model.

Pearson residuals



Deviance residuals



Pearson and deviance residuals

A comparison

Compare the two preceding plots: both the Pearson and deviance residuals suggest that the model is appropriate. Pearson and deviance residuals typically tell the same story.

The dotted line represents $\log(\hat{\mu}_i) = \log(5)$, or $\hat{\mu}_i = 5$. Compare the points to the left of the vertical dotted line to those on the right:

- Points on the right are not sparse, and we see the nice patternless band around zero.
- Points on the left are associated with sparse observations. We observe an apparent pattern, with 'bands' of points.

Pearson and deviance residuals

Sparse data

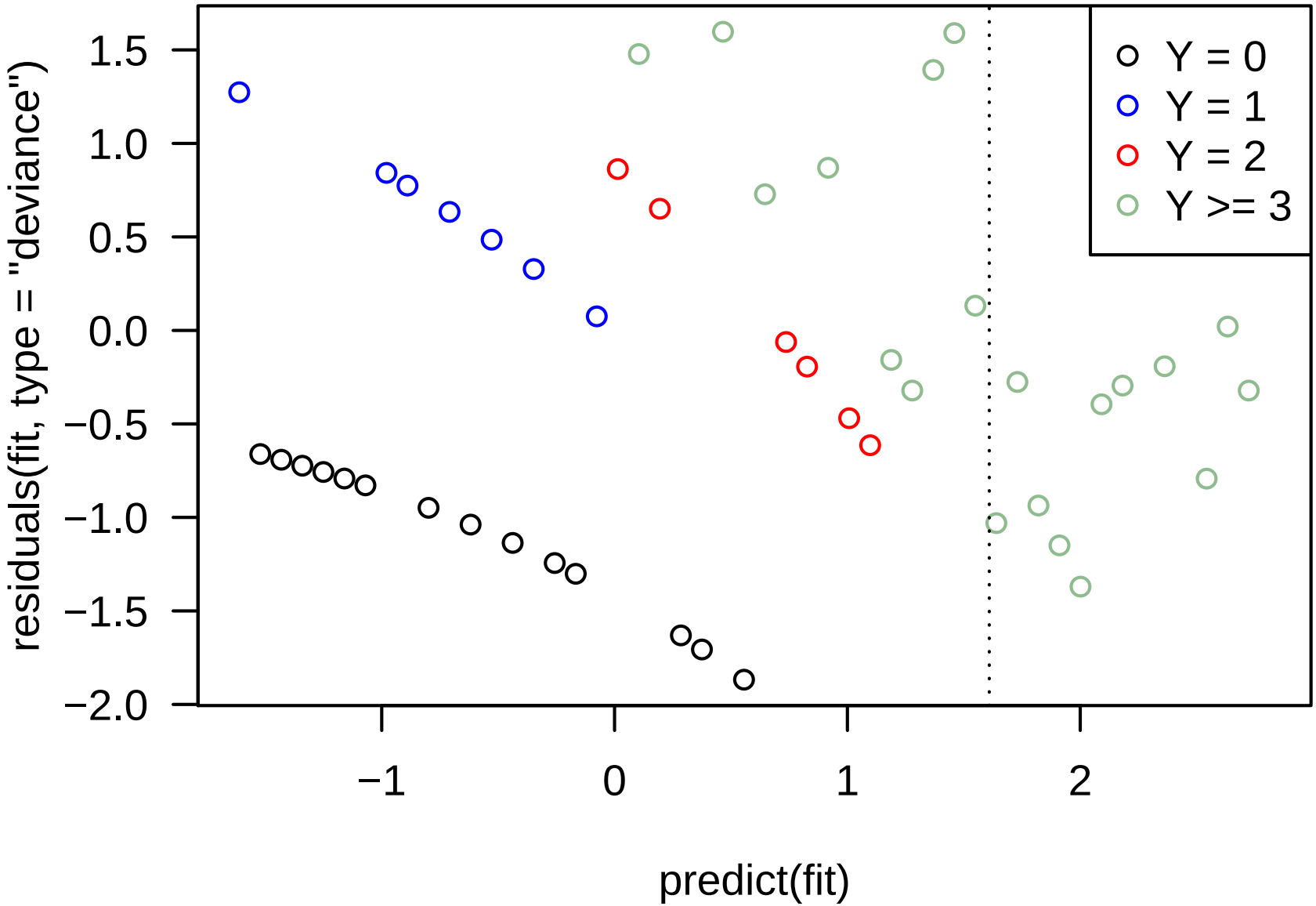
The 'banding' is because observations in this region of the plot are very discrete; they can only take one of a few possible values. For example,

- a Poisson random variable with expectation 0.1 will almost always be either 0 or 1, but
- a Poisson random variable with expectation 7.5 could plausibly be anywhere between 2 and 15.

When the data are not 'sparse', then the response variable and the residuals are well approximated by the continuous normal distribution, making it easier to assess goodness-of-fit with the residual plots.

Pearson and deviance residuals

Sparse data



Pearson and deviance residuals

Sparse data

Ungrouped data in a logistic regression model are an extreme case of sparsity: every single observation can only be one of two possible outcomes.

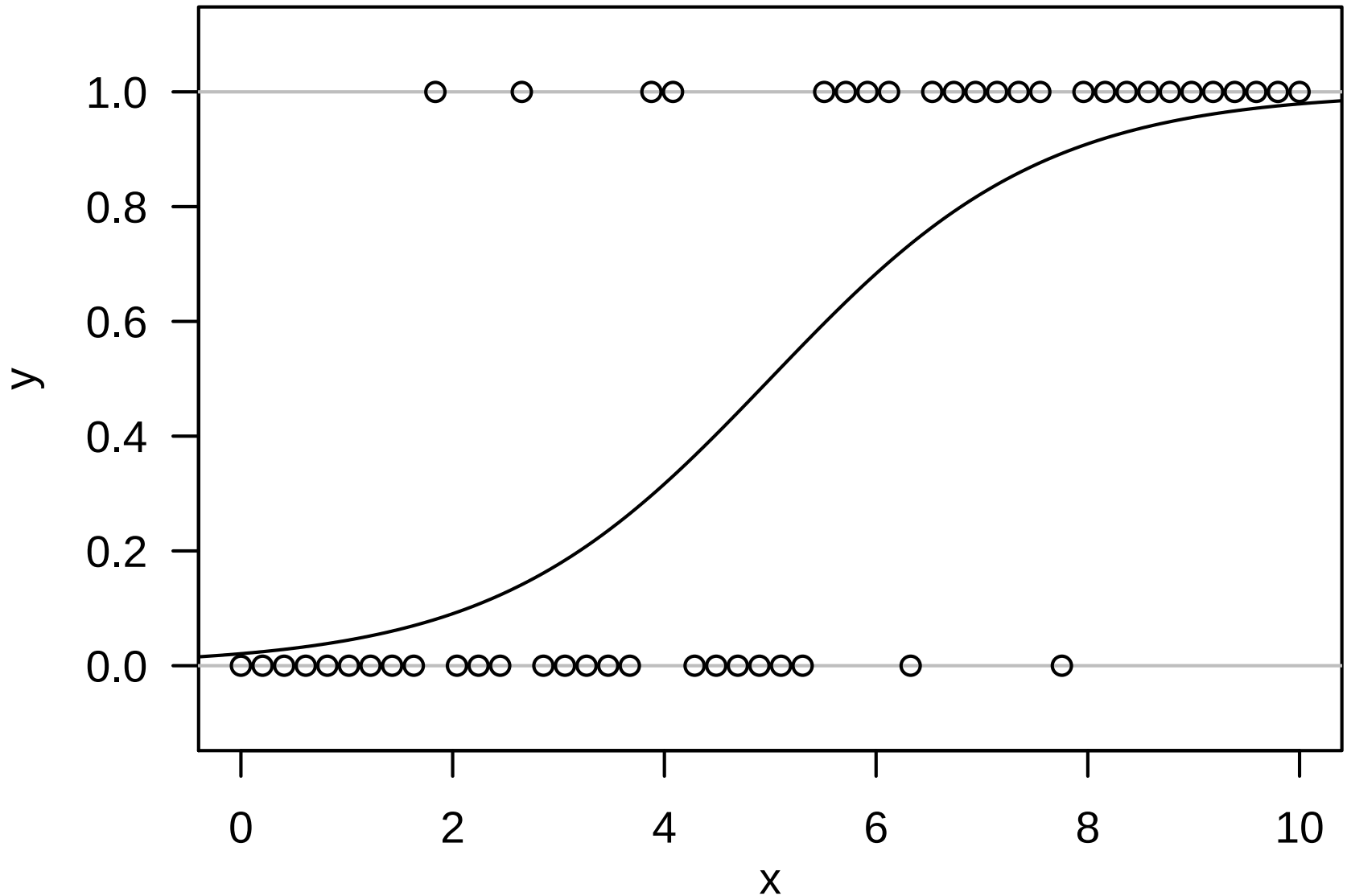
In this case, both Pearson and deviance residual plots will always show two lines of points:

- a line of points above zero, corresponding to 'successes' ($y_i = 1$), and
- a line of points below zero, corresponding to 'failures' ($y_i = 0$).

This makes it particularly difficult to interpret the plots.

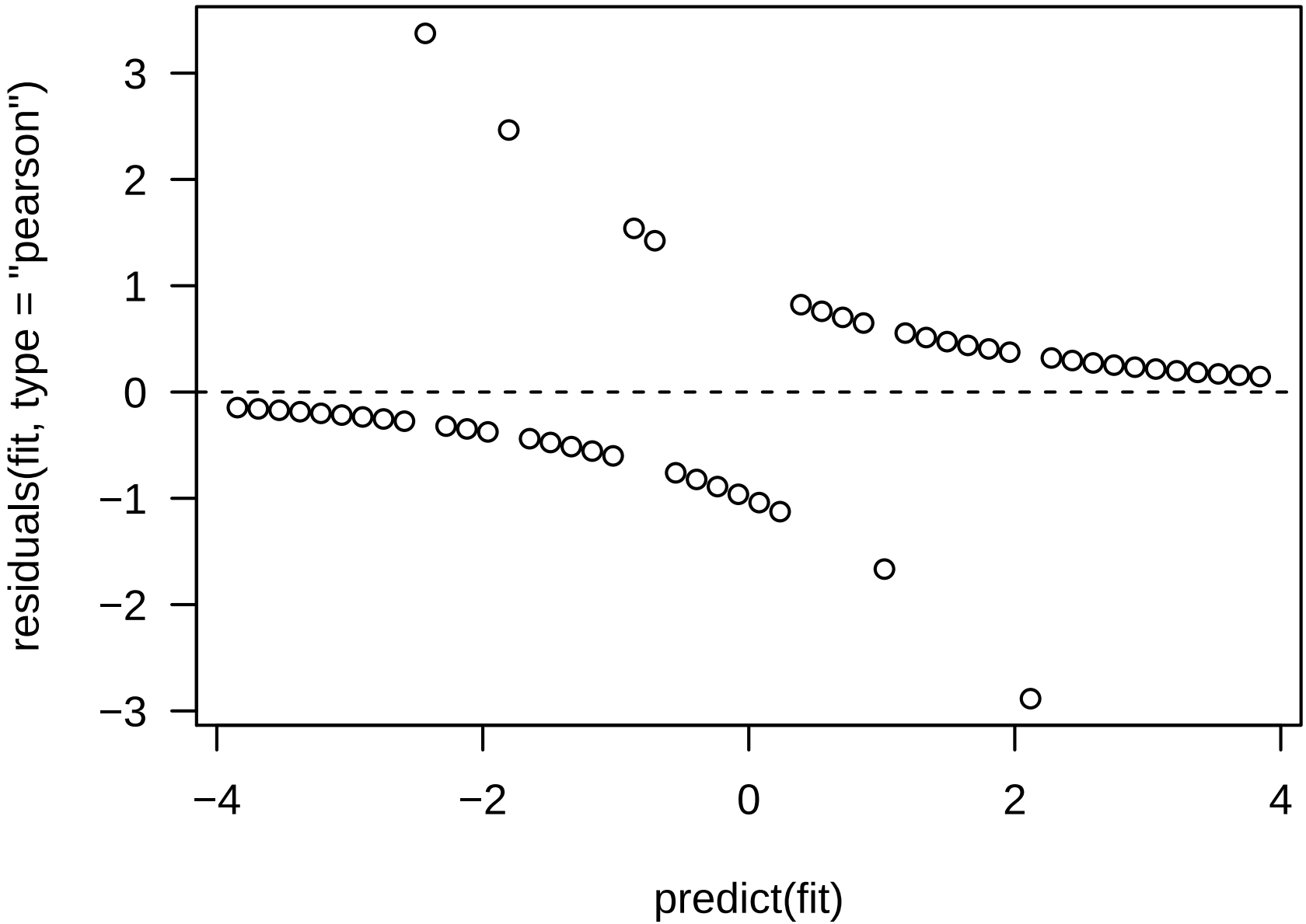
Pearson and deviance residuals

Sparse data: Ungrouped logistic regression



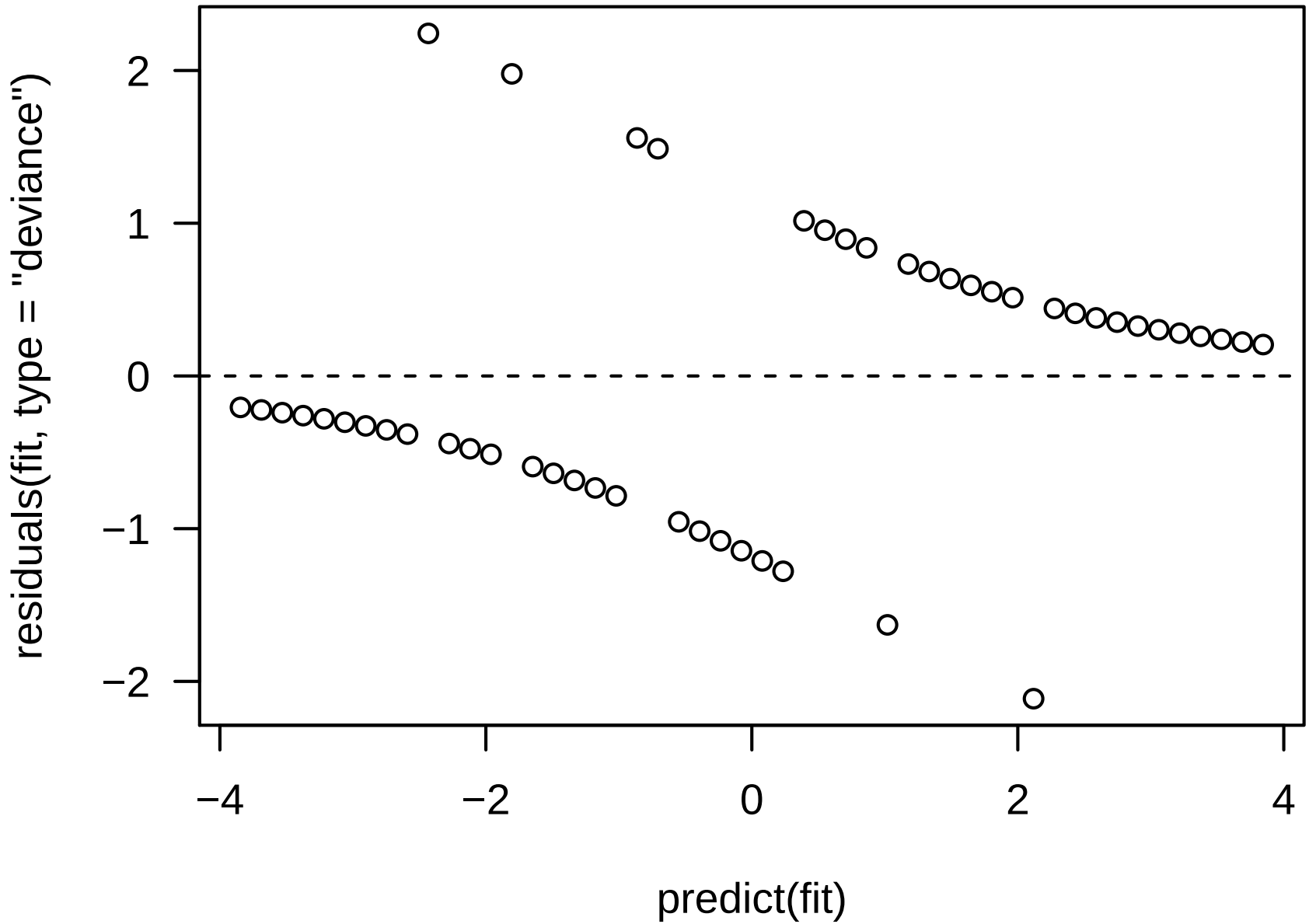
Pearson and deviance residuals

Sparse data: Pearson residuals



Pearson and deviance residuals

Sparse data: Deviance residuals



Randomised quantile residuals

Interpreting Pearson and deviance residual plots is problematic when data are sparse. The plots show an apparent pattern and the residuals are non-normal, even if the model is correct.

Randomised quantile residuals randomly 'jitter' the residuals to break up the banding in the residual plots. The jittering is done in a clever way so that the residuals are approximately normal if the model is correct, regardless of whether or not the data are sparse.

Advantage:

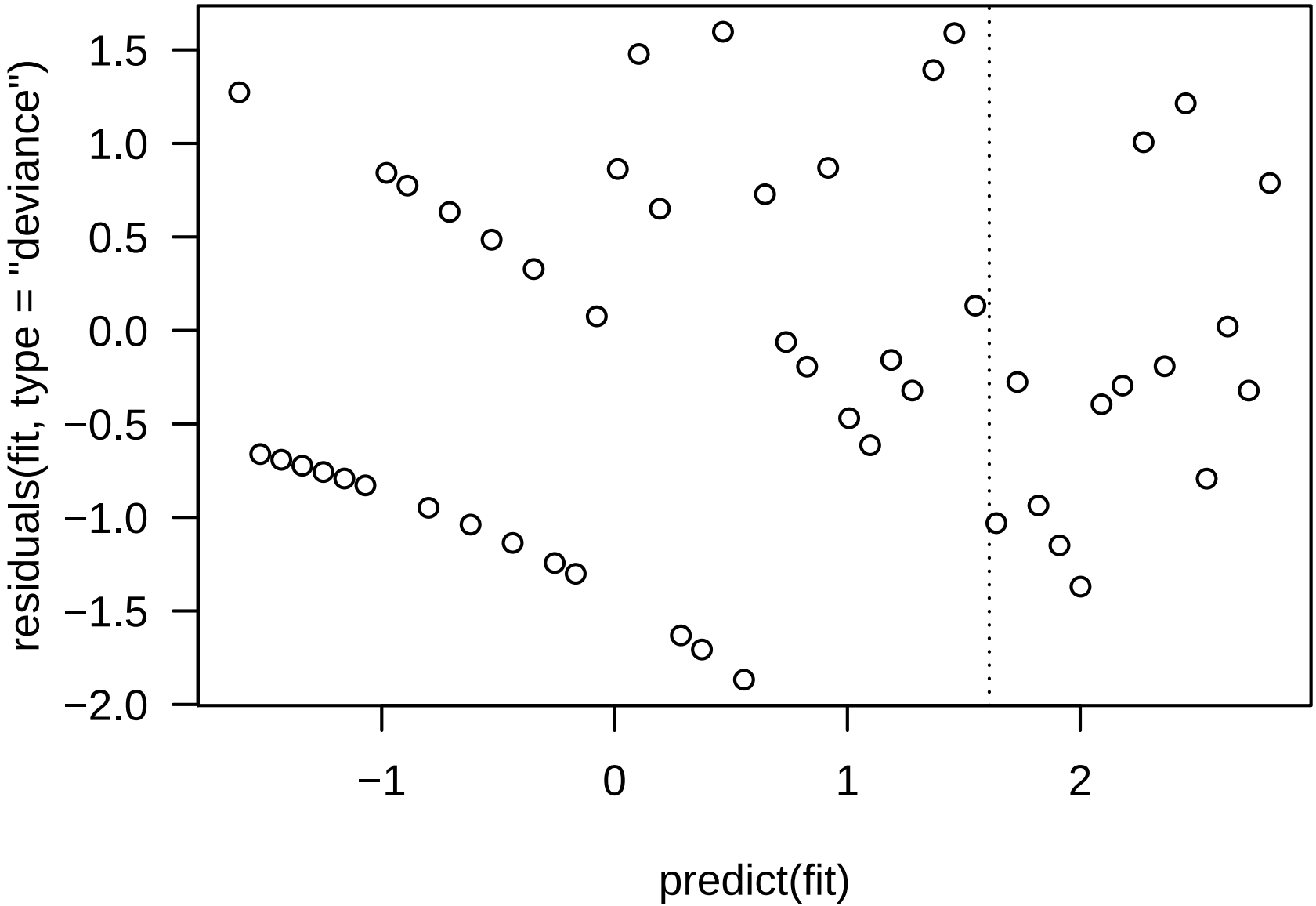
- We can look for a patternless band in our residual plot, even if the data are sparse.

Disadvantage:

- Because the jittering is random, generating residuals from the same model more than once will result in a slightly different plot each time.

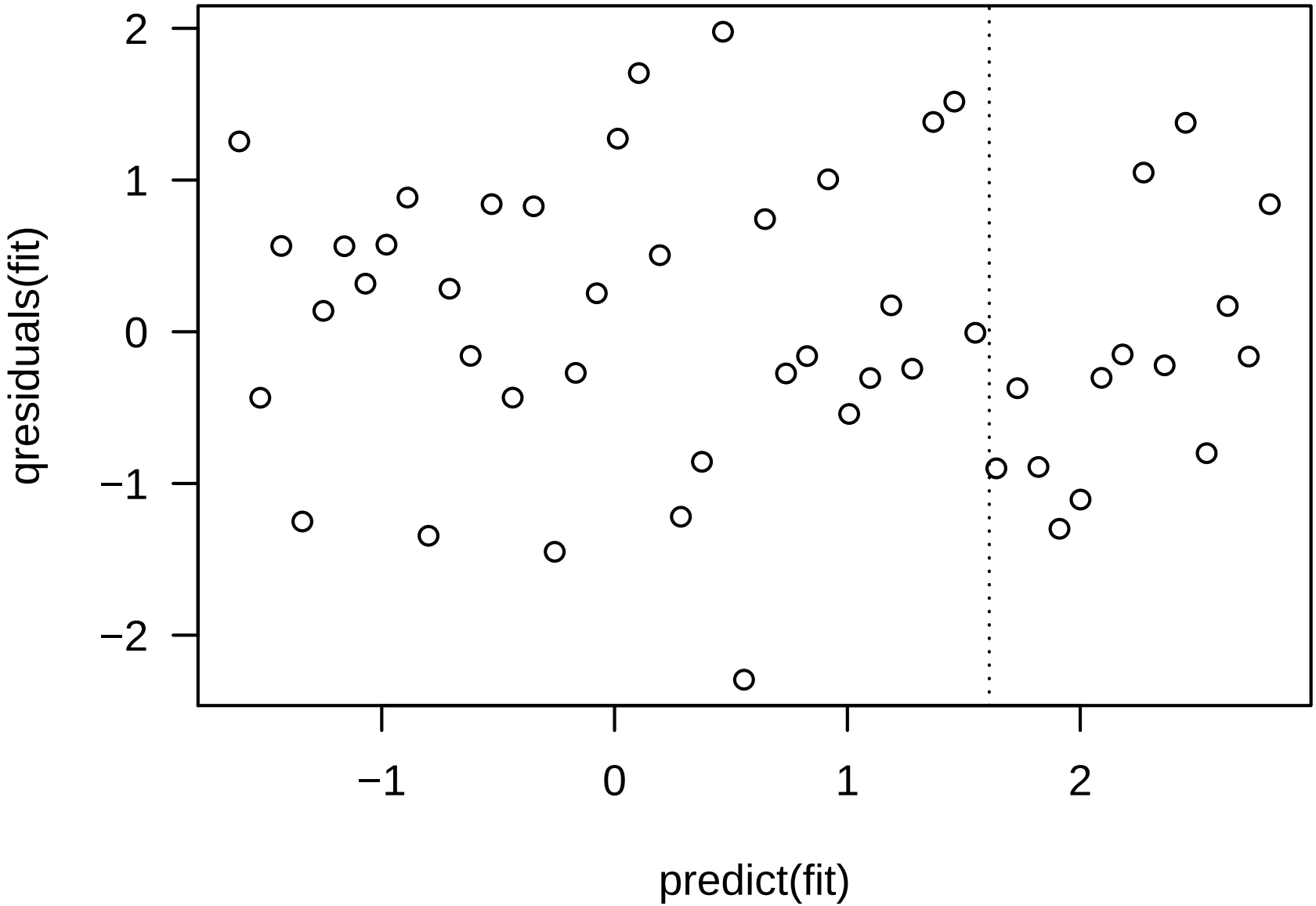
Deviance residuals

Poisson regression



Randomised quantile residuals

Poisson regression



Randomised quantile residuals

Poisson regression

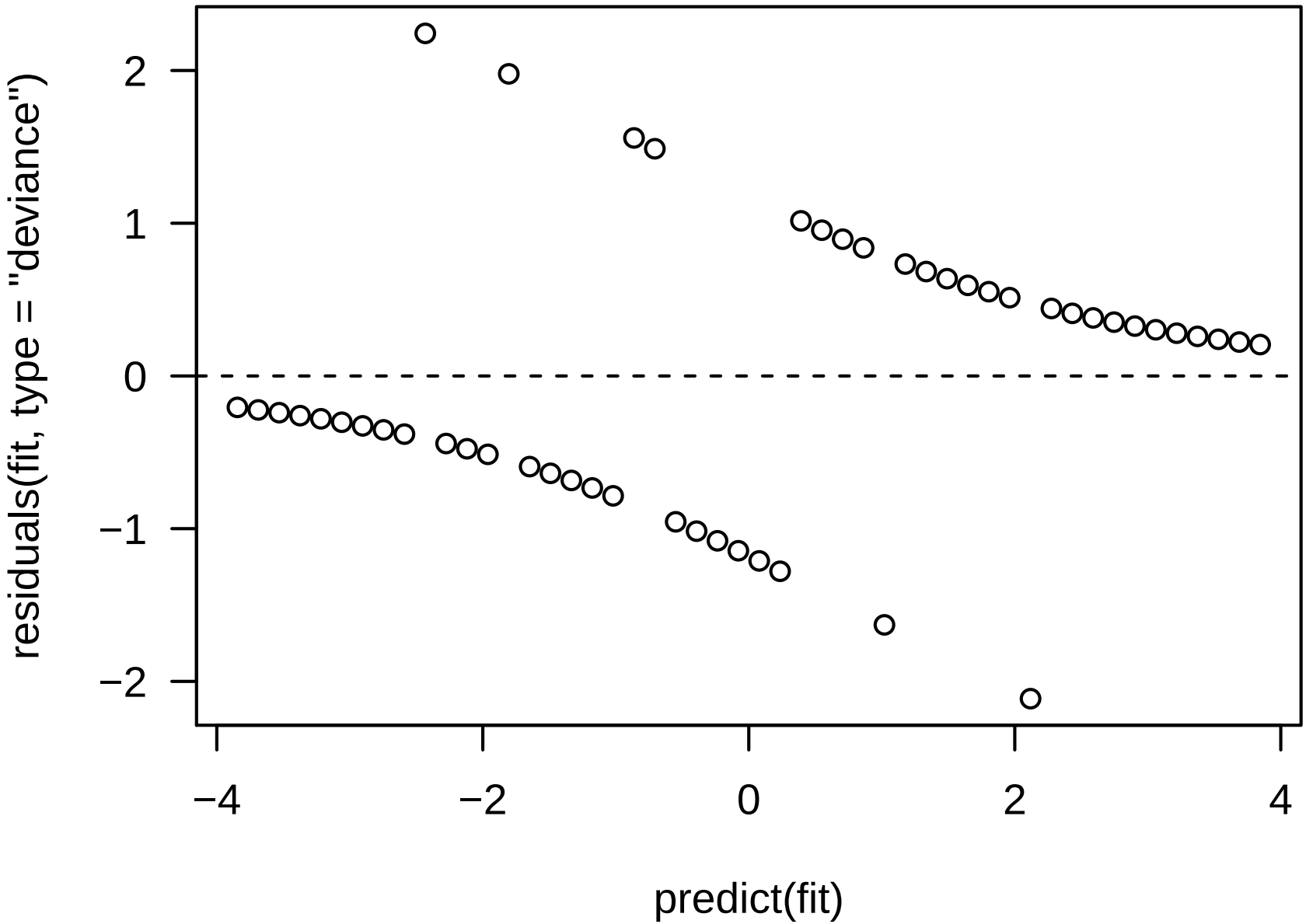
Compare the two preceding plots:

- Residuals from non-sparse observations on the right-hand side—those that were already approximately normal—have only changed slightly.
- The bands of residuals on the left-hand side have been broken up substantially, and now appear to show a patternless band around zero.

We see something similar with our ungrouped logistic regression model—compare the following two plots.

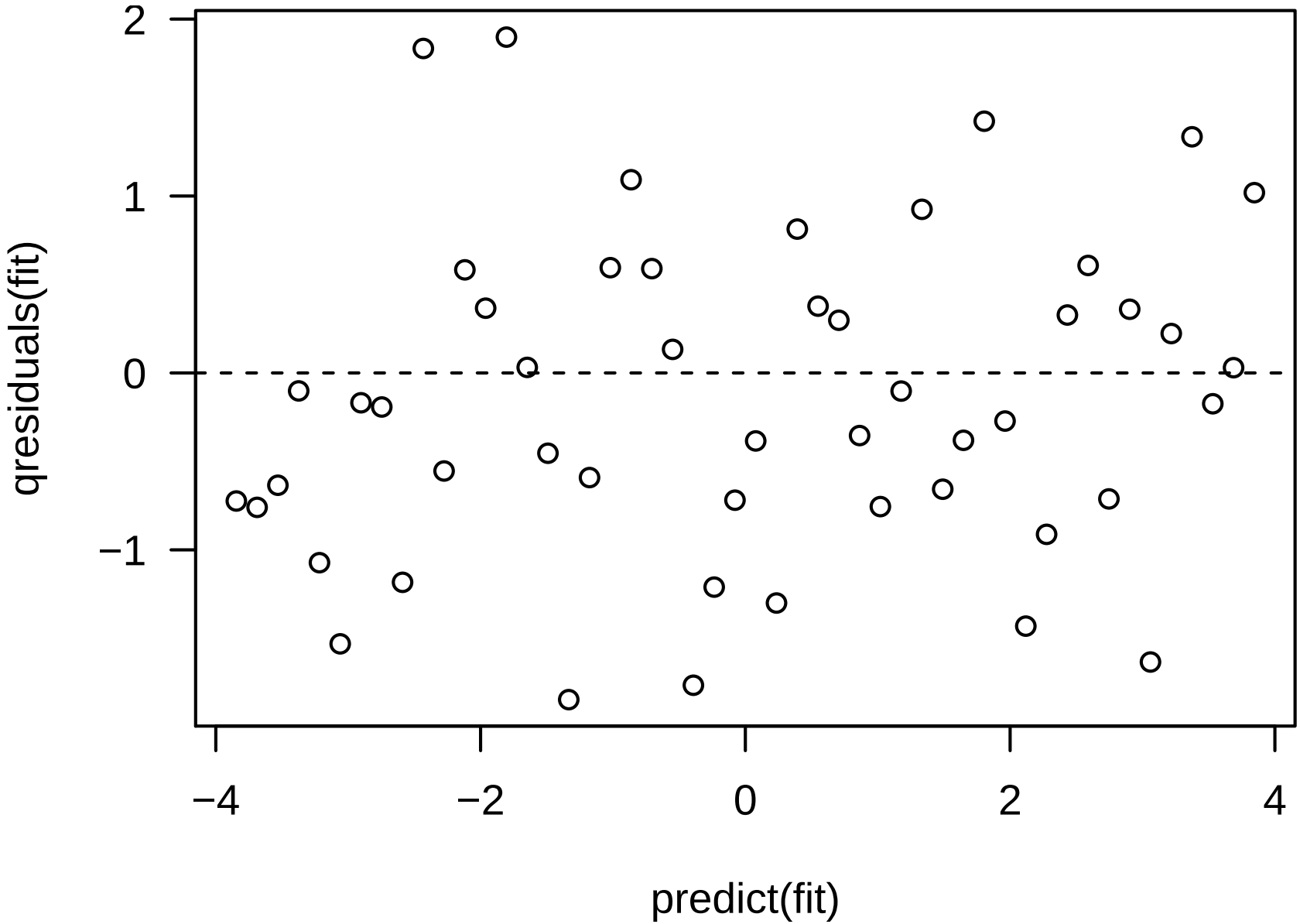
Deviance residuals

Sparse data: Ungrouped logistic regression



Randomised quantile residuals

Sparse data: Ungrouped logistic regression



Doing it in R

Calculating residuals for GLMs

For raw residuals:

```
residuals(fit, type = "response")
```

For Pearson residuals:

```
residuals(fit, type = "pearson")
```

For deviance residuals, the default:

```
residuals(fit, type = "deviance")
```

For randomised quantile residuals:

```
library(statmod)
qresiduals(fit)
```

Doing it in R

Creating a residual plot

Simply plot the residuals against the fitted values:

```
plot(predict(fit), residuals(fit, type = "deviance"))
```

You can replace the `residuals()` function above to use whatever type of residuals you like.

If you like, the following adds a dashed horizontal line at zero:

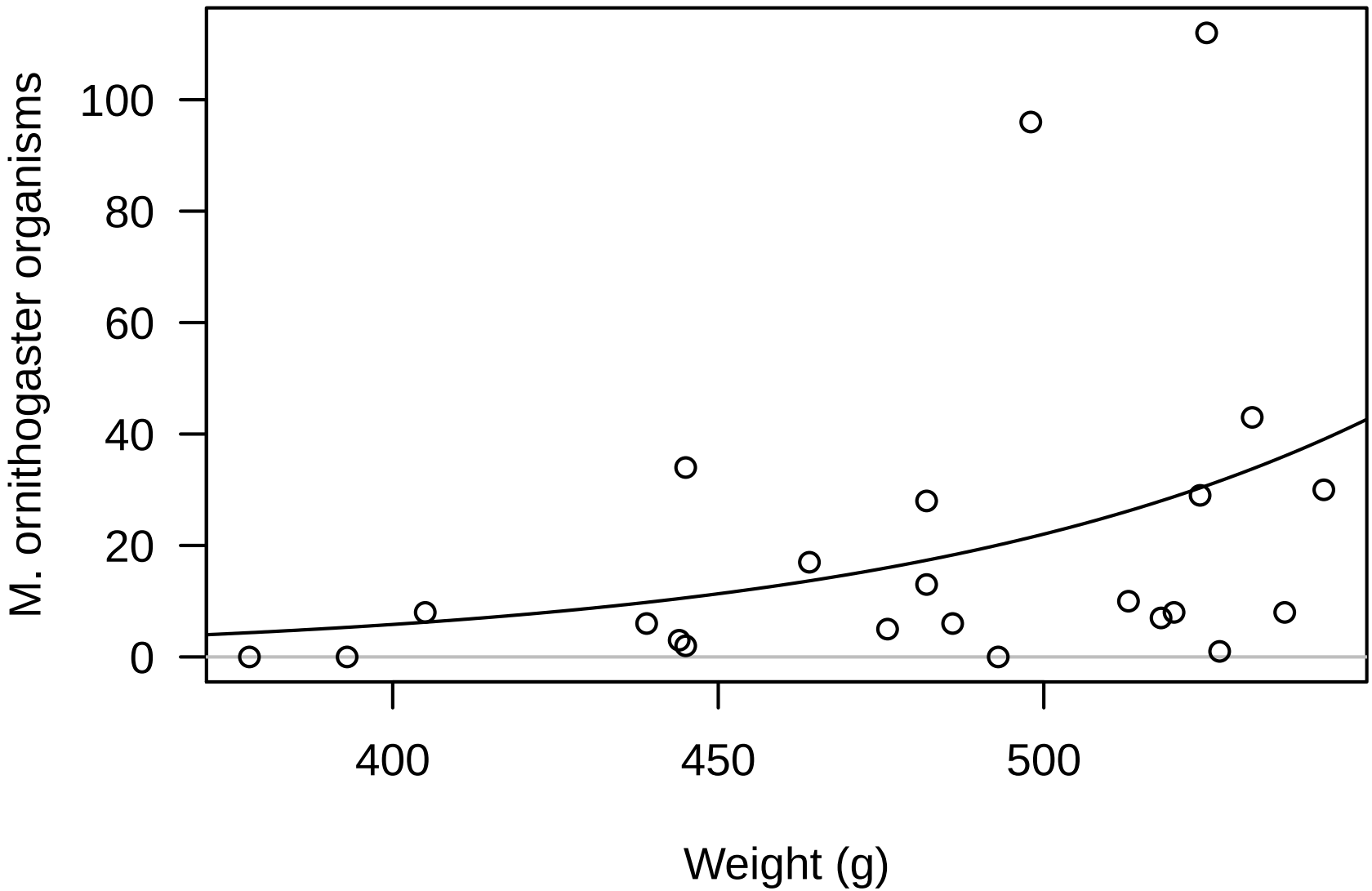
```
abline(h = 0, lty = "dashed")
```

Alternatively, the following will create a plot using deviance residuals:

```
plot(fit, which = 1)
```

Macrorhabdus ornithogaster chicken analysis

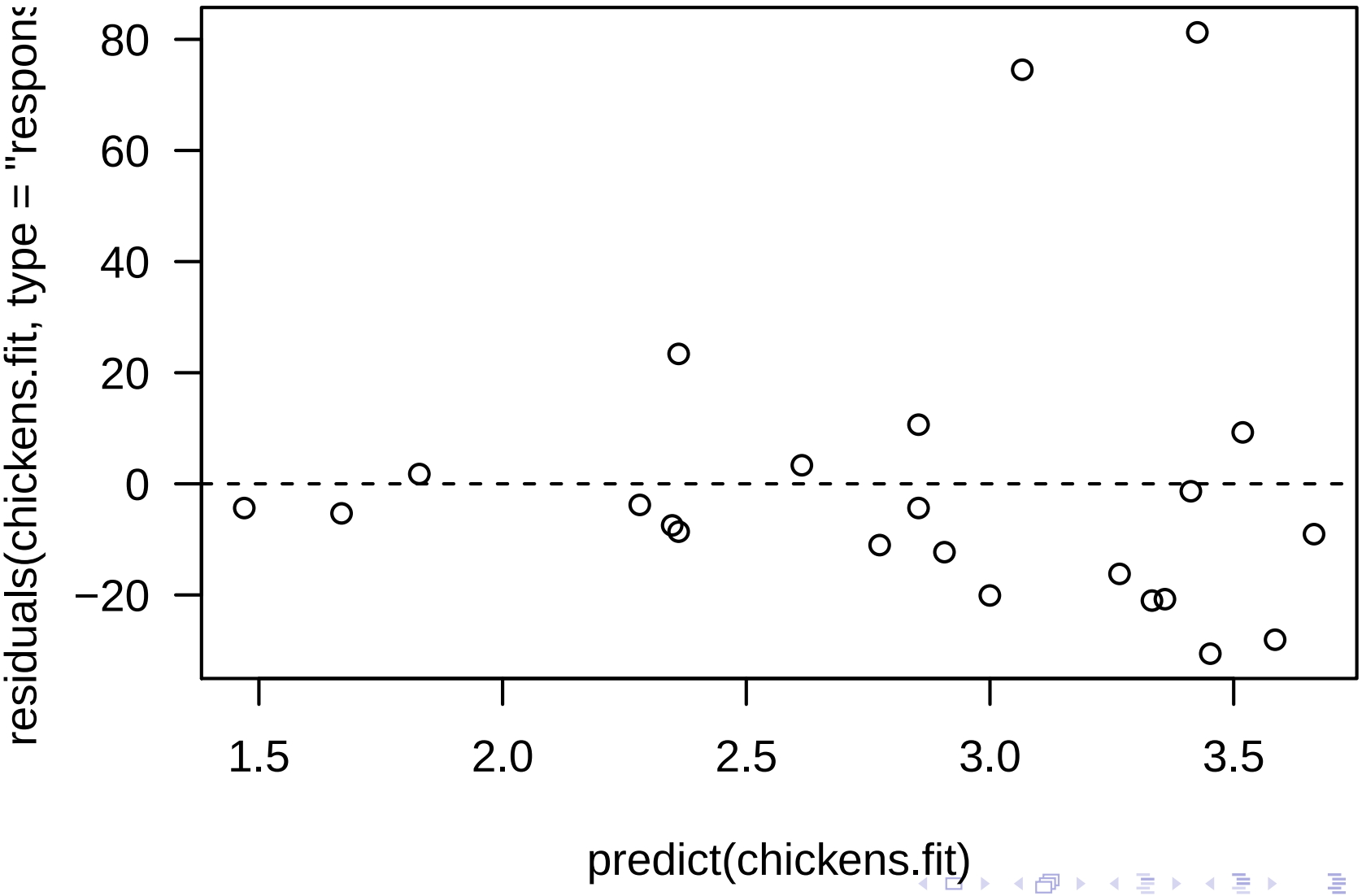
```
chickens.fit <- glm(mo ~ weight, family = "poisson")
```



Macrorhabdus ornithogaster chicken analysis

Raw residuals

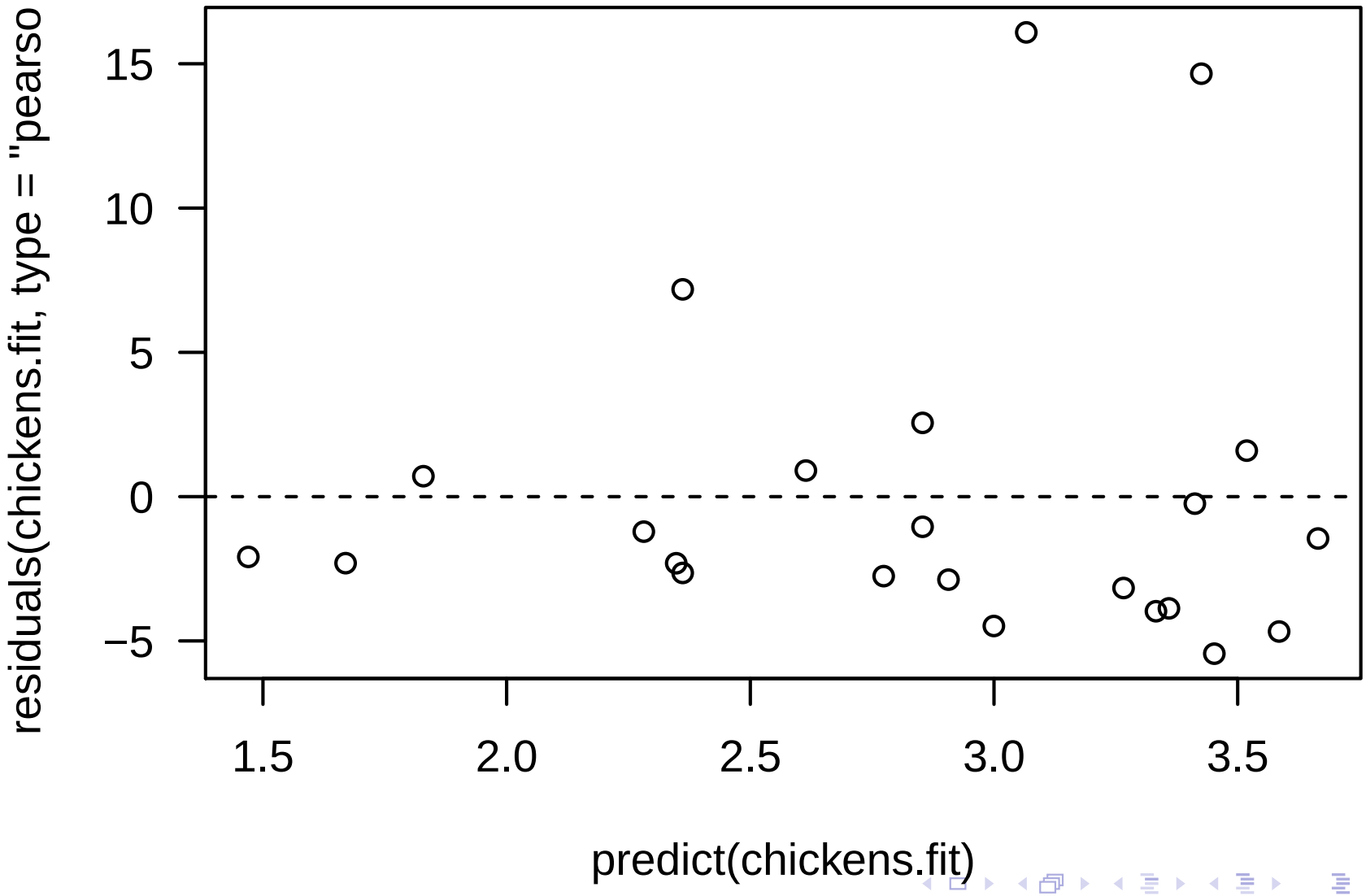
```
plot(predict(chickens.fit), residuals(chickens.fit, type = "response"))
```



Macrorhabdus ornithogaster chicken analysis

Pearson residuals

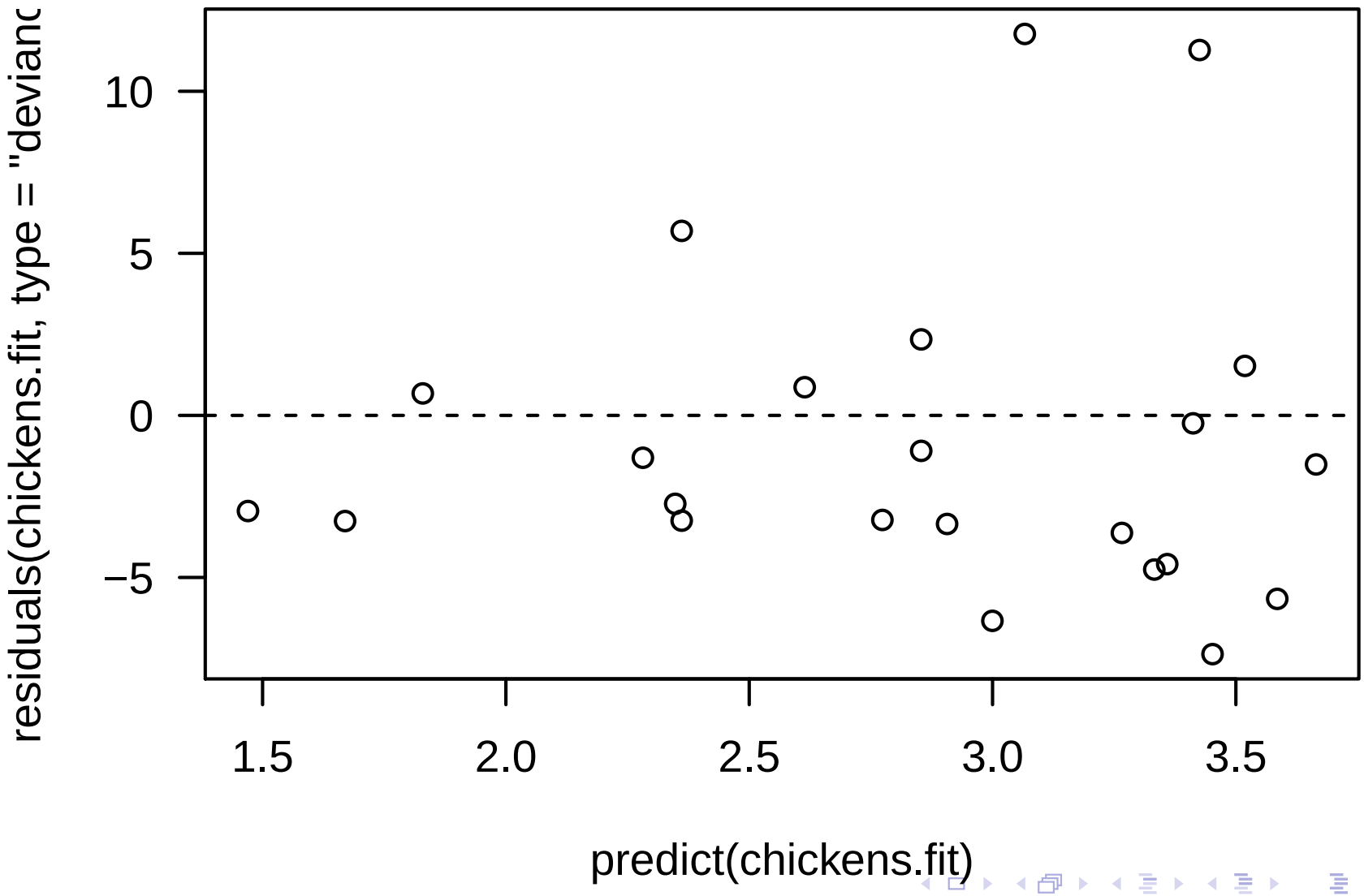
```
plot(predict(chickens.fit), residuals(chickens.fit, type = "pearson"))
```



Macrorhabdus ornithogaster chicken analysis

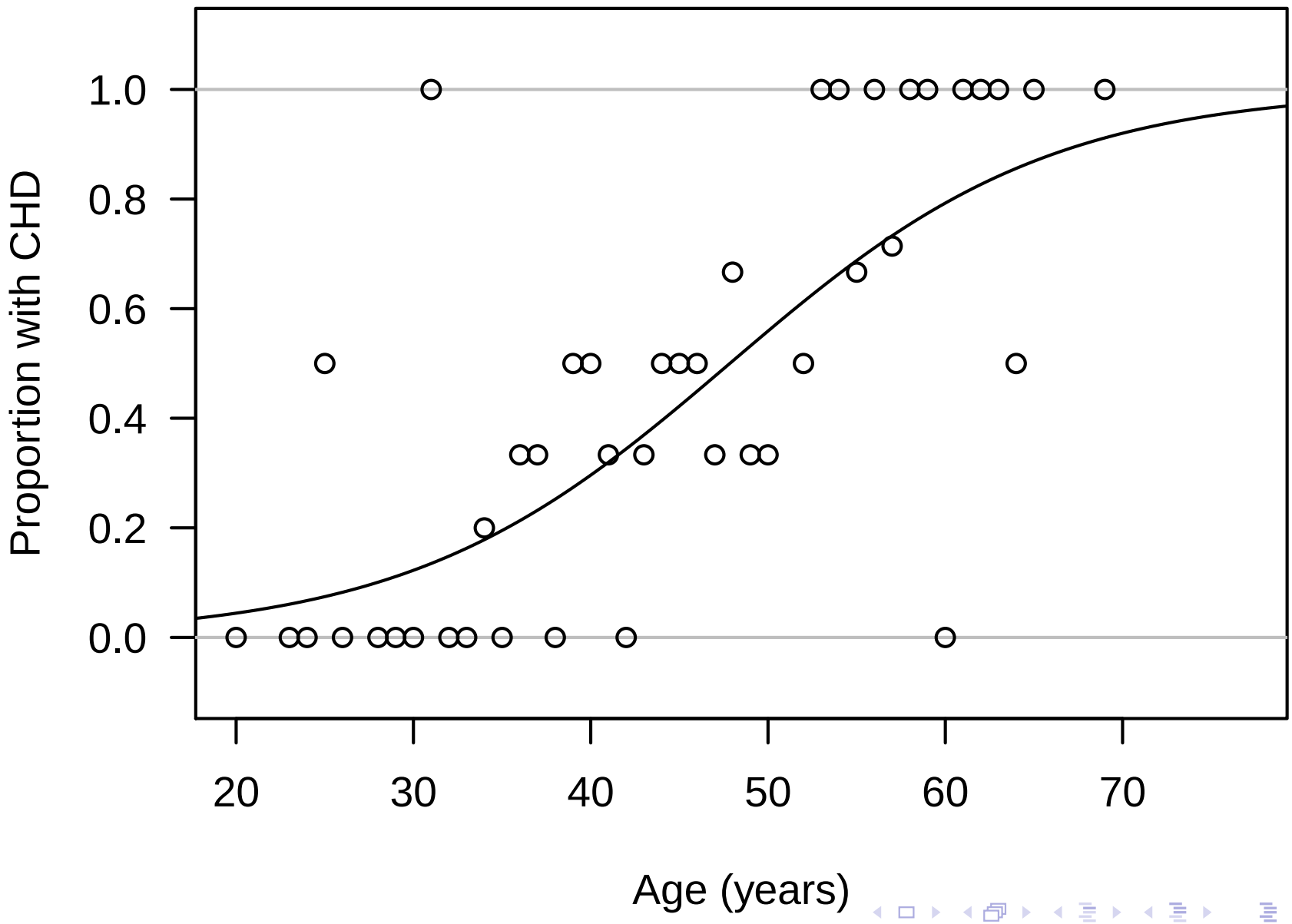
Deviance residuals

```
plot(predict(chickens.fit), residuals(chickens.fit, type = "deviance"))
```



Coronary heart disease analysis

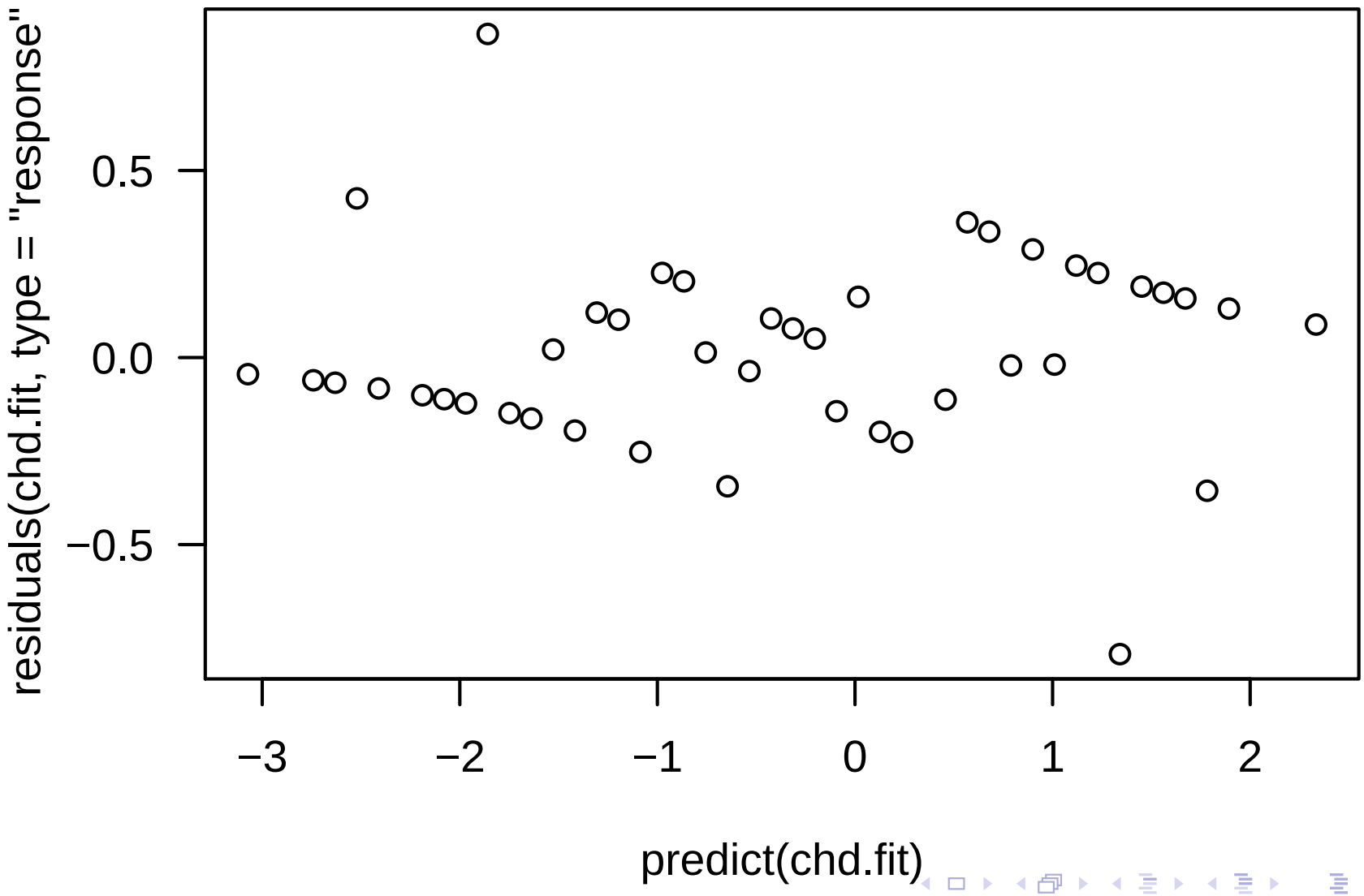
```
chd.fit <- glm(cbind(y, n - y)   age, family = "binomial")
```



Coronary heart disease analysis

Raw residuals

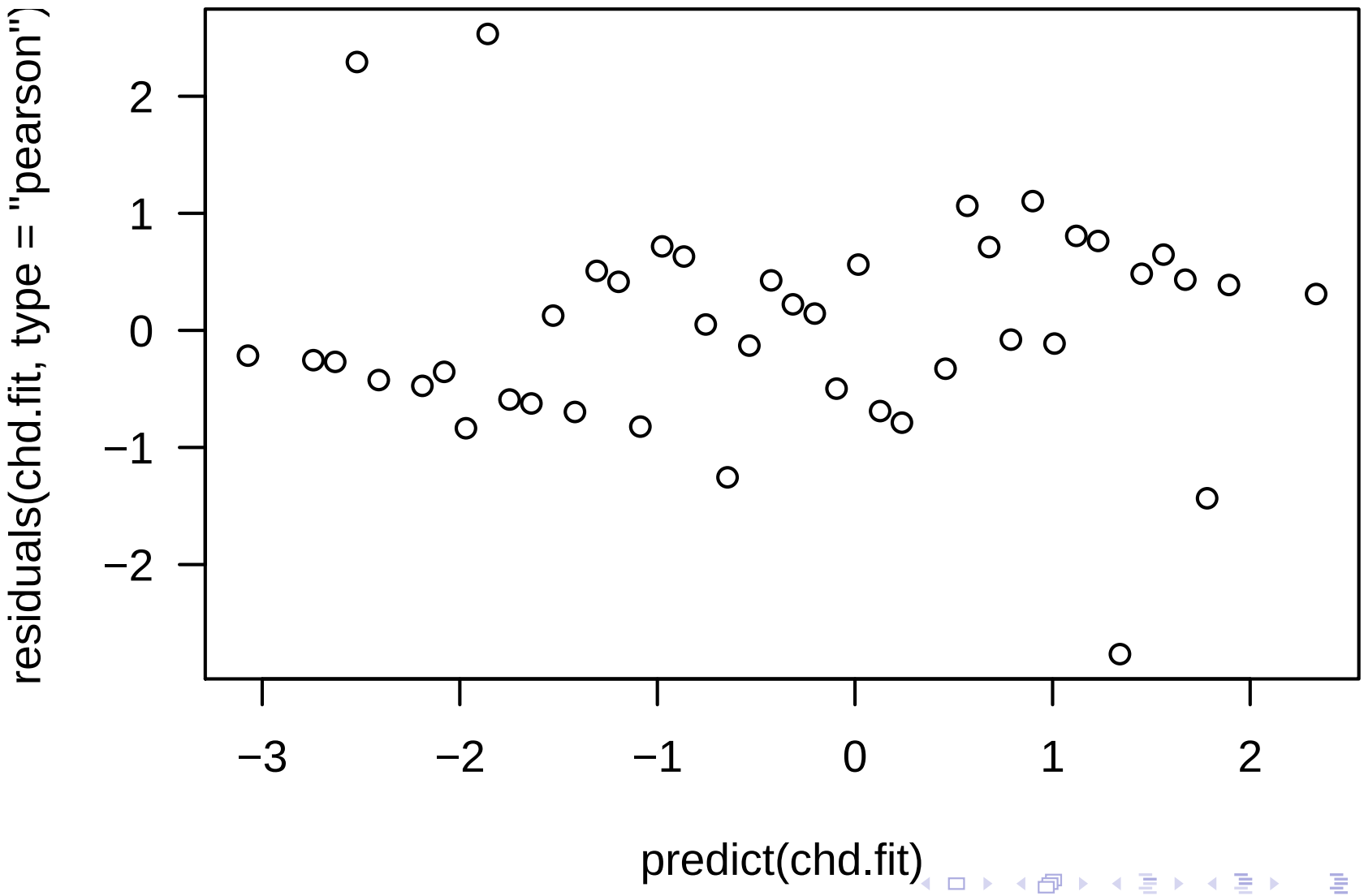
```
plot(predict(chd.fit), residuals(chd.fit, type = "response"))
```



Coronary heart disease analysis

Pearson residuals

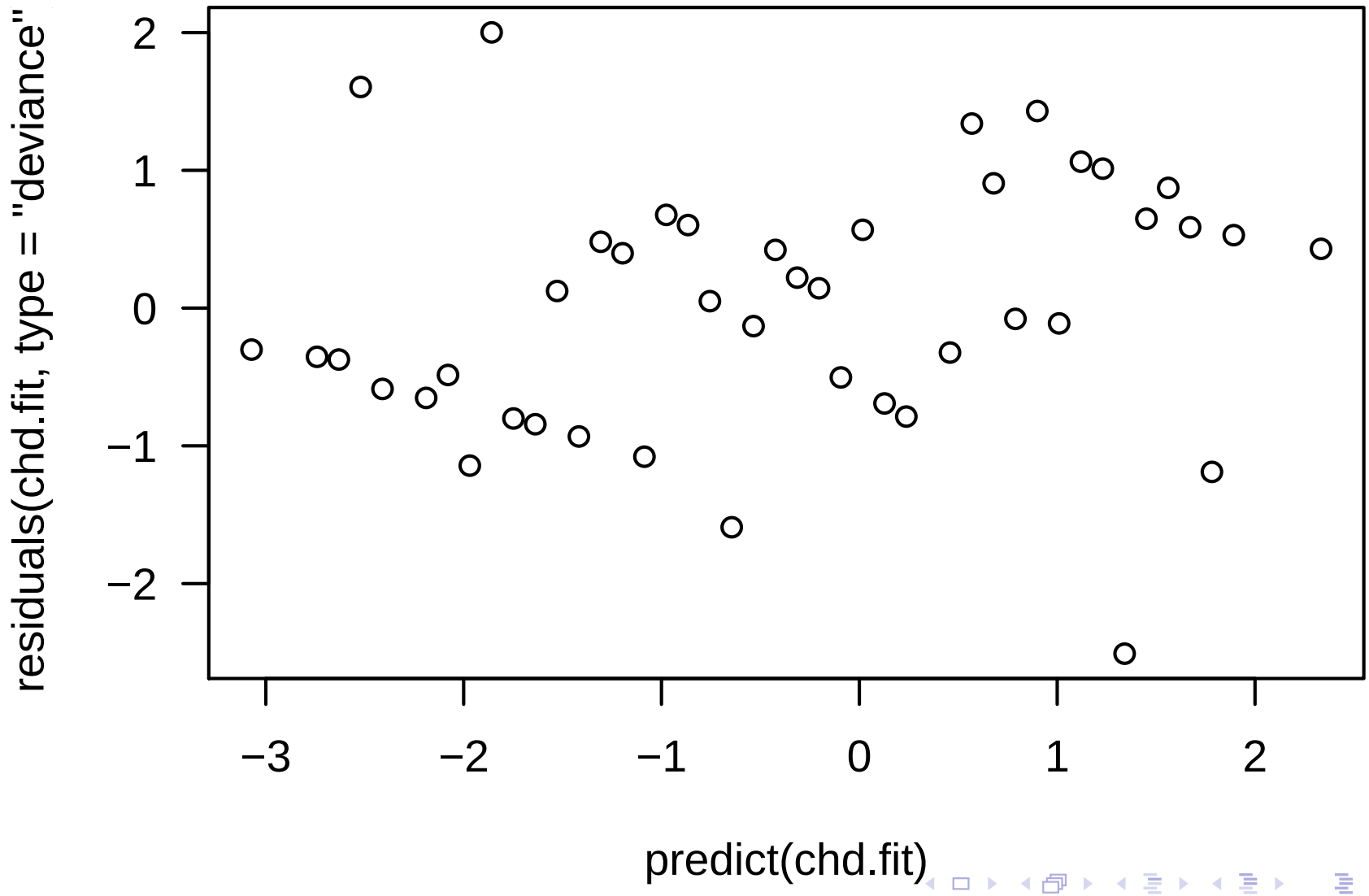
```
plot(predict(chd.fit), residuals(chd.fit, type = "pearson"))
```



Coronary heart disease analysis

Deviance residuals

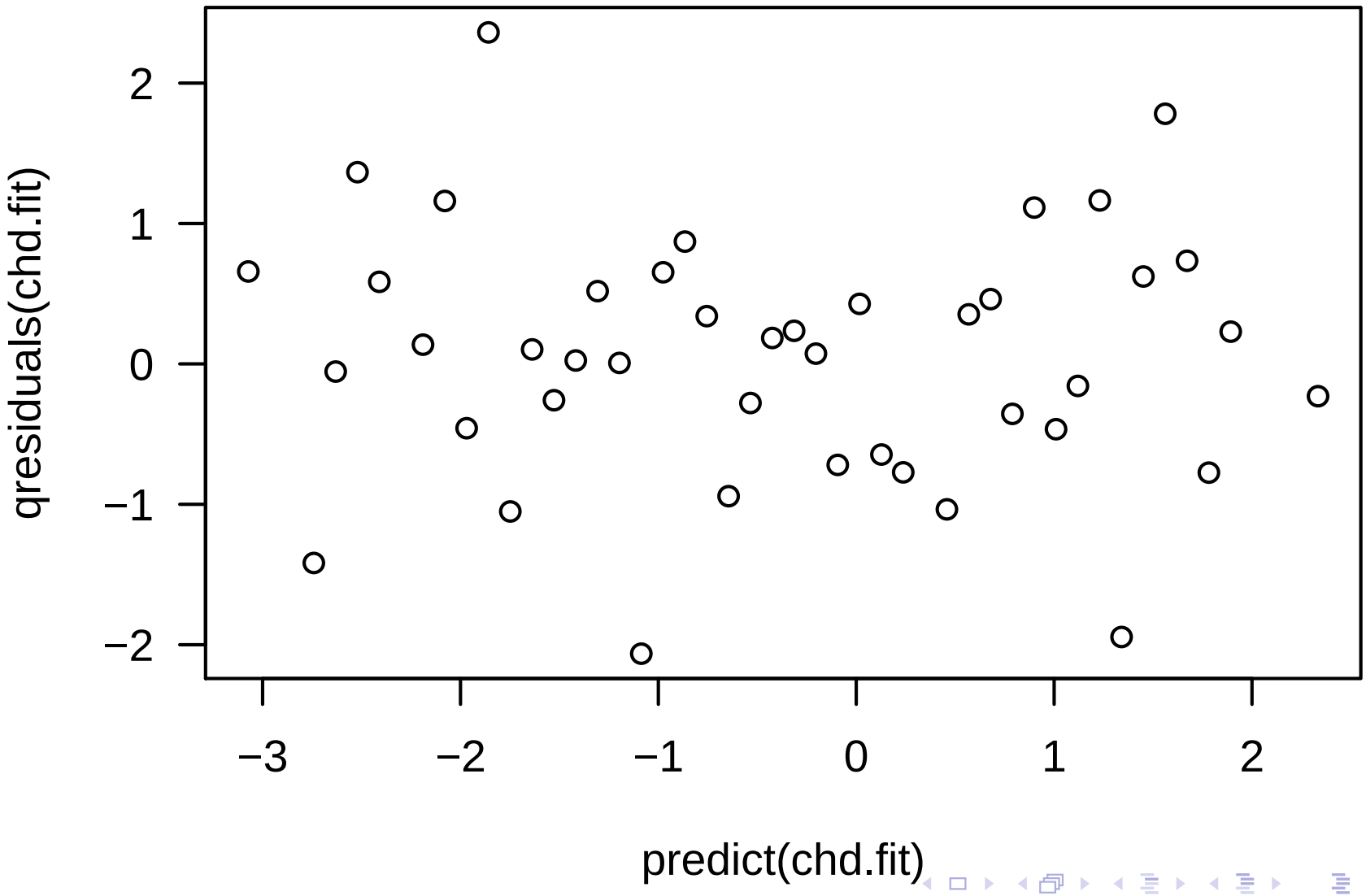
```
plot(predict(chd.fit), residuals(chd.fit, type = "deviance"))
```



Coronary heart disease analysis

Randomised quantile residuals

```
plot(predict(chd.fit), qresiduals(chd.fit))
```



Interpretations

For the *Macrorhabdus ornithogaster* chicken analysis:

- The Pearson and deviance residuals do not appear to have approximately constant variance.
- This suggests there may be a problem with our model.
- See the following handout to see what we can do.

For the coronary heart disease analysis:

- It is a little hard to interpret the Pearson residual plot due to sparse data, but on the whole the deviance residual plot looks fine.
- The randomised quantile residual plot removes the problem of interpreting residuals for sparse data, and it does not suggest any problems with the model.