432 Class 15 Slides

github.com/THOMASELOVE/2019-432

2019-03-26

Setup

```
library(skimr)
library(arm)
library(rms)
library(boot)
library(MASS)
library(HSAUR)
library(broom)
library(tidyverse)
```

Today's Materials

Regression Models for Count Outcomes (Notes Chapter 18)

- The `polyps` example
- Poisson Regression
- Overdispersion and Quasi-Poisson Regression
- Negative Binomial Regression

The polyps example

The polyps data frame within the HSAUR package describes the results of a placebo-controlled trial of a non-steroidal anti-inflammatory drug in the treatment of a condition called familial adenomatous polyposis (FAP).

```
head(HSAUR::polyps, 4)
```

```
  number   treat age
1     63 placebo  20
2      2    drug  16
3     28 placebo  18
4     17    drug  22
```

Let's clean this up a little, and make it a tibble.

Cleaning the Polyps Data

```
pol432 <- HSAUR::polyps %>%
    mutate(subject = 1:20) %>%
    rename(polyps12m = number) %>%
    select(subject, age, treat, polyps12m) %>%
    tbl_df
```

Details

The tibble includes 20 observations (skim on next slide) on:

- `age` = the age of the patient at the start of the trial, in years
- `treat` = the patient's treatment arm
- `polyps12m` = (our outcome), the number of colonic polyps at 12 months for this patient

We want to understand how the number of colonic polyps at 12 months is related to both `treat` and `age`.

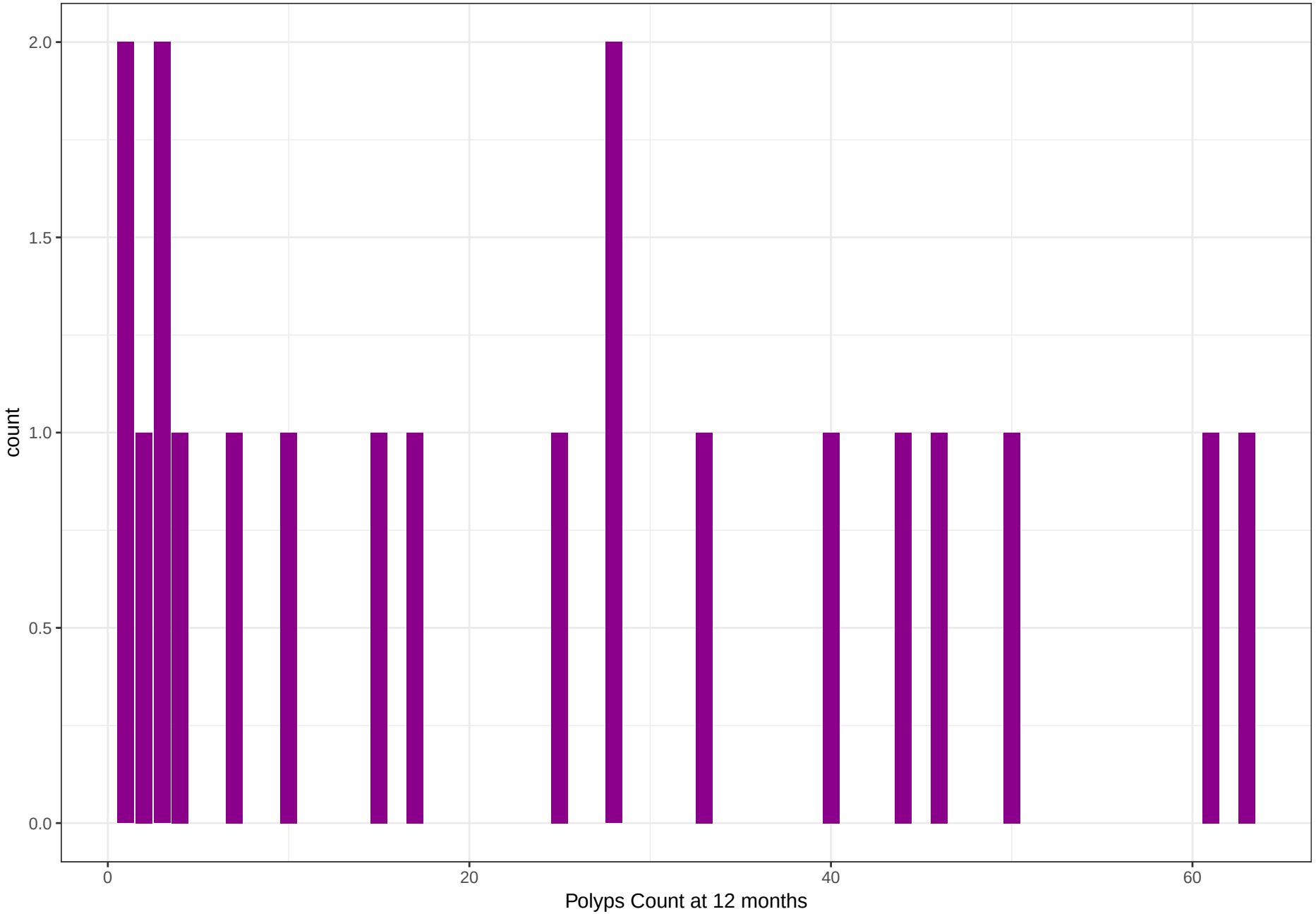
- Note that the actual trial was halted after a planned interim analysis suggested compelling evidence in favor of the `drug` over `placebo`. Data sources and the original NEJM reference (1993) may be found in `?HSAUR::polyps`

Skim of the pol432 data

```
Skim summary statistics
 n obs: 20
 n variables: 3

Variable type: factor
 variable missing complete  n n_unique             top_counts ordered
    treat       0       20 20        2 pla: 11, dru: 9, NA: 0   FALSE

Variable type: numeric
  variable missing complete  n  mean    sd p0   p25 median   p75 p100     hist
       age       0       20 20 25     9.05 13 19.75   22.5 27.75   50 ▃▇▃▂▁▁▁▁
 polyps12m       0       20 20 24.05 20.85  1  3.75   21   41      63 ▇▂▁▂▁▂▁▂
```

Regression on a Count Outcome

Plot the Outcome



Why not model count data with a linear regression?

The data on `polyps12m` is count data. Why wouldn't we model this with linear regression?

- A count can only be positive. Linear regression would estimate some subjects as having negative counts.
- A count is unlikely to follow a Normal distribution. In fact, it's far more likely that the log of the counts will follow a Poisson distribution.

So, we'll try that. We'll run a generalized linear model with a log link function, ensuring that all of the predicted values will be positive, and using a Poisson error distribution. This is called **Poisson regression.**

Poisson regression may be appropriate when the dependent variable is a count of events. The events must be independent - the occurrence of one event must not make any other more or less likely.

Fit a Poisson Regression Model with `glm`

The default link function with the `poisson` family is the log.

```
mod_1 <- glm(polyps12m ~ treat + age,
             data=pol432,
             family=poisson())
mod_1
```

```
Call:  glm(formula = polyps12m ~ treat + age, family = poisson

Coefficients:
(Intercept)    treatdrug          age
    4.52902     -1.35908     -0.03883

Degrees of Freedom: 19 Total (i.e. Null);  17 Residual
Null Deviance:      378.7
Residual Deviance: 179.5    AIC: 273.9
```

Model Equation and confidence intervals

The model equation is:

```
log(polyps at 12 months) = 4.53 - 1.36 treat - 0.039 age
```

```
> confint(mod_1)
Waiting for profiling to be done...
                   2.5 %     97.5 %
(Intercept)   4.24184712  4.817700
treatdrug    -1.59549678 -1.133761
age          -0.05074339 -0.027393
```

Both `treatdrug` and age have CIs that exclude 0.

Model Summary (edited)

```
Call: glm(formula = polyps12m ~ treat + age,
                      family = poisson(), data = pol432)

             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.529024   0.146872   30.84  < 2e-16 ***
treatdrug   -1.359083   0.117643  -11.55  < 2e-16 ***
age         -0.038830   0.005955   -6.52 7.02e-11 ***

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 378.66  on 19  degrees of freedom
Residual deviance: 179.54  on 17  degrees of freedom
AIC: 273.88

Number of Fisher Scoring iterations: 5
```

There's a real problem here, but we'll get back to it.

Making Predictions

A subject of age 30 on the drug is predicted to have:

```
log(polyps at 12 months) = 4.53 - 1.36 treat - 0.039 age
log(polyps at 12 months) = 4.53 - 1.36 (1) - 0.039 (30)
log(polyps at 12 months) = 4.53 - 1.36 - 1.17 = 2, so...
polyps at 12 months = exp(2) = 7.4
```

so this subject is estimated by Model 1 to have 7.4 polyps.

Making the Prediction Automatically. . .

```
predict(mod_1, data.frame(treat = "drug", age = 30),
        type = "response", se.fit = TRUE)
```

```
$fit
       1
7.426338

$se.fit
        1
0.8642734

$residual.scale
[1] 1
```

The `residual.scale` specifies the square root of the *dispersion* value used in computing the standard errors.

The Poisson Regression Model

The Poisson distribution is used to model a *count* outcome - that is, an outcome with possible values (0, 1, 2, . . .). The model takes a somewhat familiar form to the models we've used for linear and logistic regression. If our outcome is y and our linear predictors X, then the model is:

$$y_i \sim \text{Poisson}(\theta_i)$$

The parameter θ must be positive, so it makes sense to fit a linear regression on the logarithm of this. . .

$$\theta_i = exp(\beta_0 + \beta_1 X_1 + ...\beta_k X_k)$$

Poisson Regression

The coefficients β can be exponentiated and treated as multiplicative effects. For example, if our model is for y_i = counts of polyps, with the regression equation:

$$y_i \sim \text{Poisson}(exp(4.5 - 1.4(\text{treat} = \text{drug}) - 0.04 \text{ age}))$$

where `treat = drug` is 1 if the treatment is drug, and 0 if the treatment is placebo, and age is in years, we can interpret the coefficients as follows. . .

Interpreting the Poisson Model

- The constant term, 4.5, gives us the intercept of the regression - the prediction if `treat = placebo` and `age = 0`. Since we have no one with age of zero, we try not to interpret this term.
- The coefficient of `treat = drug`, -1.4, tells us that the predictive difference between the `drug` and `placebo` treatments can be found by multiplying the polyps count by exp(-1.4) = 0.25, yielding a reduction of 75%.
- The coefficient of `age`, -0.04, is the expected difference in count of polyps (on the log scale) for each additional year of age. Thus, the expected multiplicative *increase* is $e^{-0.04} = 0.96$, corresponding to a 4% negative difference in the count.

As with linear or logistic regression, each coefficient is interpreted as a comparison where one predictor changes by one unit, while the others remain constant.

Looking for Overdispersion

The notion of *overdispersion* arises here. When fitting generalized linear models with Poisson error distributions, the residual deviance and its degrees of freedom should be approximately equal if the model fits well.

If the residual deviance is far greater than the degrees of freedom, then overdispersion may well be a problem. In this case, the residual deviance is more than 10 times the size of the residual degrees of freedom, so that's a clear indication of overdispersion.

```
Residual deviance: 179.54  on 17  degrees of freedom
```

Poisson regression model requires that the outcome (here the polyps counts) be independent. A possible reason for the overdispersion we see here is that polyps likely do not occur independently of one another but may "cluster" together.

Dealing with Over-Dispersion

Poisson regressions do not supply an independent variance parameter σ, and as a result can be overdispersed, and usually are. Under the Poisson distribution, the variance equals the mean - so the standard deviation equals the square root of the mean. Gelman and Hill provide an overdispersion test in R for a Poisson model as follows. . .

```
yhat <- predict(mod_1, type = "response")
display(mod_1)
```

```
glm(formula = polyps12m ~ treat + age, family = poisson(), dat
            coef.est coef.se
(Intercept)  4.53     0.15
treatdrug   -1.36     0.12
age         -0.04     0.01
---
  n = 20, k = 3
  residual deviance = 179.5, null deviance = 378.7 (difference
```

Overdispersion Test, Part 2

```
n <- 20; k <- 3
z <- (pol432$polyps12m - yhat) / sqrt(yhat)
cat("overdispersion ratio is ", sum(z^2)/ (n - k), "\n")
```

```
overdispersion ratio is  10.72783
```

```
cat("p value of overdispersion test is ",
    pchisq(sum(z^2), df = n-k, lower.tail = FALSE), "\n")
```

```
p value of overdispersion test is  9.711657e-30
```

The sum of squared standardized residuals $\sum z^2 = 182.37$. The estimated overdispersion factor is $182.37/17 = 10.73$, and the p value here is 0 for all intents and purposes.

Conclusions from Overdispersion Test

This indicates that the probability is essentially zero that a random variable from a χ^2 distribution with (n - k) = 17 degrees of freedom would be as large as 182.37.

In summary, the polyps counts are overdispersed by a factor of more than 10, which is enormous (even a factor of 2 would be considered large) and also highly statistically significant. The basic correction for overdisperson is to multiply all regression standard errors by $\sqrt{10.73} = 3.28$.

Our main inferences are not too seriously affected by this adjustment, as it turns out, and we will see this in what's called an overdispersed Poisson model that we'll fit next.

Deal with Overdispersion via a Quasi-Likelihood Estimation Procedure

To deal with overdispersion, we could apply a quasi-likelihood estimation procedure.

```
mod_2 <- glm(polyps12m ~ treat + age,
             data=pol432, family=quasipoisson())
mod_2
```

```
Call:  glm(formula = polyps12m ~ treat + age, family = quasipo
    data = pol432)

Coefficients:
(Intercept)    treatdrug          age
    4.52902     -1.35908     -0.03883

Degrees of Freedom: 19 Total (i.e. Null);  17 Residual
Null Deviance:      378.7
```

Summary of mod_2

```
Call: glm(formula = polyps12m ~ treat + age,
        family = quasipoisson(), data = pol432)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.52902    0.48106   9.415 3.72e-08 ***
treatdrug   -1.35908    0.38533  -3.527  0.00259 **
age         -0.03883    0.01951  -1.991  0.06284 .

(Dispersion parameter for quasipoisson family
                                  taken to be 10.72805)

    Null deviance: 378.66  on 19  degrees of freedom
Residual deviance: 179.54  on 17  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

Comparing the Two Models

Estimate	Poisson Est (SE)	Quasi-Poisson Est (SE)
Intercept	4.53 (0.147)	4.53 (0.481)
`treat`	-1.36 (0.118)	-1.36 (0.385)
`age`	-0.039 (0.006)	-0.039 (0.020)

Poisson model and Quasi-Poisson model:

```
log(polyps at 12 months) = 4.53 - 1.36 treat - 0.039 age
```

Confidence Intervals for `mod_2` coefficients

```
Waiting for profiling to be done...
```

	2.5 %	97.5 %
(Intercept)	3.59170079	5.481738085
treatdrug	-2.18466623	-0.653930343
age	-0.07969808	-0.003027068

The estimates in Model 2 are still statistically significant, but the standard errors for each coefficient are considerably larger when we account for overdispersion.

The Quasi-Poisson (Overdispersed Poisson) Regression

The `quasipoisson` and negative binomial (coming soon) models are very similar. We write the overdispersed "quasiPoisson" model as:

$$y_i \sim \text{overdispersed Poisson}(\mu_i = exp(X_i\beta_i), \omega)$$

where ω is the overdispersion parameter, 10.73, in our case.

The Poisson model is just the overdispersed Poisson model with $\omega = 1$.

ANOVA for Poisson or Quasi-Poisson Regression

The ANOVA approach here (as with `glm`, generally) produces sequential tests, so in this case, we see whether `treat` by itself has a significant effect, and then whether age, given `treat` already in the model, has an impact. If we want to test the coefficients in another order, we need only to specify that order when we fit the model.

```
anova(mod_2, test = "Chisq")
```

```
Analysis of Deviance Table

Model: quasipoisson, link: log

Response: polyps12m

Terms added sequentially (first to last)

      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     19     378.66
treat  1  150.101        18     228.56 0.0001836 ***
age    1   49.018        17     179.54 0.0325526 *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Making Predictions with Model 2

```
predict(mod_2, data.frame(treat = "drug", age = 30),
        type = "response", se.fit = TRUE)
```

```
$fit
       1
7.426338

$se.fit
       1
2.830815

$residual.scale
[1] 3.27537
```

Fitting Poisson and Quasi-Poisson models using `Glm` from `rms`

The `Glm` function in the `rms` package can be used to fit these models.

Original Poisson Regression Model

Here's our original Poisson regression:

```
d <- datadist(pol432)
options(datadist = "d")
model1 <- Glm(polyps12m ~ treat + age, data=pol432,
              family=poisson(), x = T, y = T)
```

model1 from Glm

General Linear Model

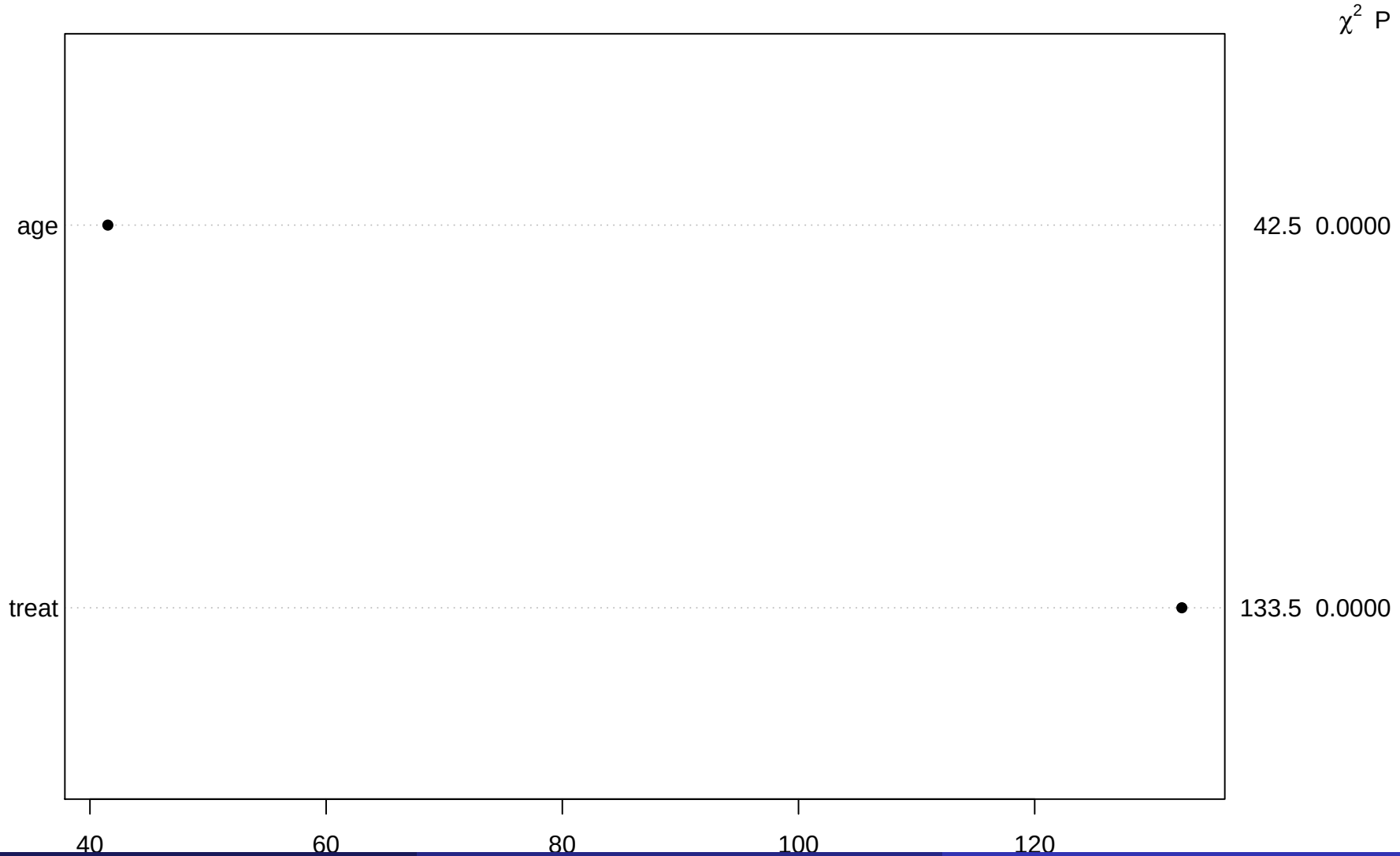
```
Glm(formula = polyps12m ~ treat + age, family = poisson(), da
    x = T, y = T)
```

		Model Likelihood Ratio Test	
Obs	20	LR chi2	199.12
Residual d.f.	17	d.f.	2
g	0.8324108	Pr(> chi2)	<0.0001

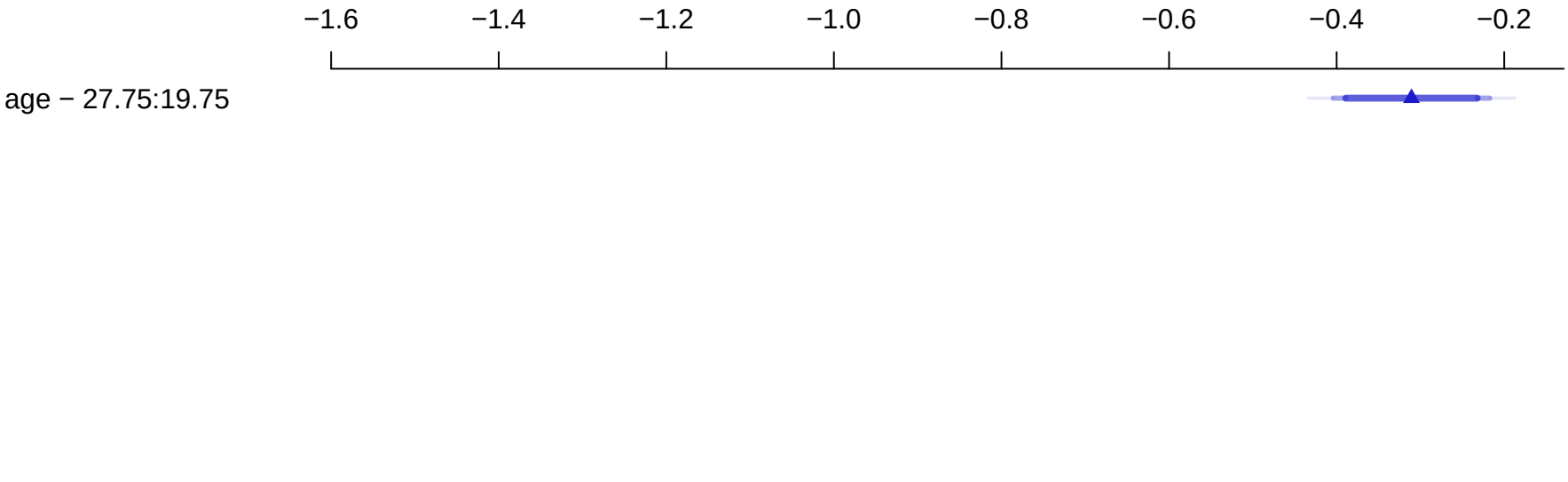
	Coef	S.E.	Wald Z	Pr(>|Z|)
Intercept	4.5290	0.1469	30.84	<0.0001
treat=drug	-1.3591	0.1176	-11.55	<0.0001
age	-0.0388	0.0060	-6.52	<0.0001

plot(anova(model1))

```
plot(anova(model1))
```



plot(summary(model1))

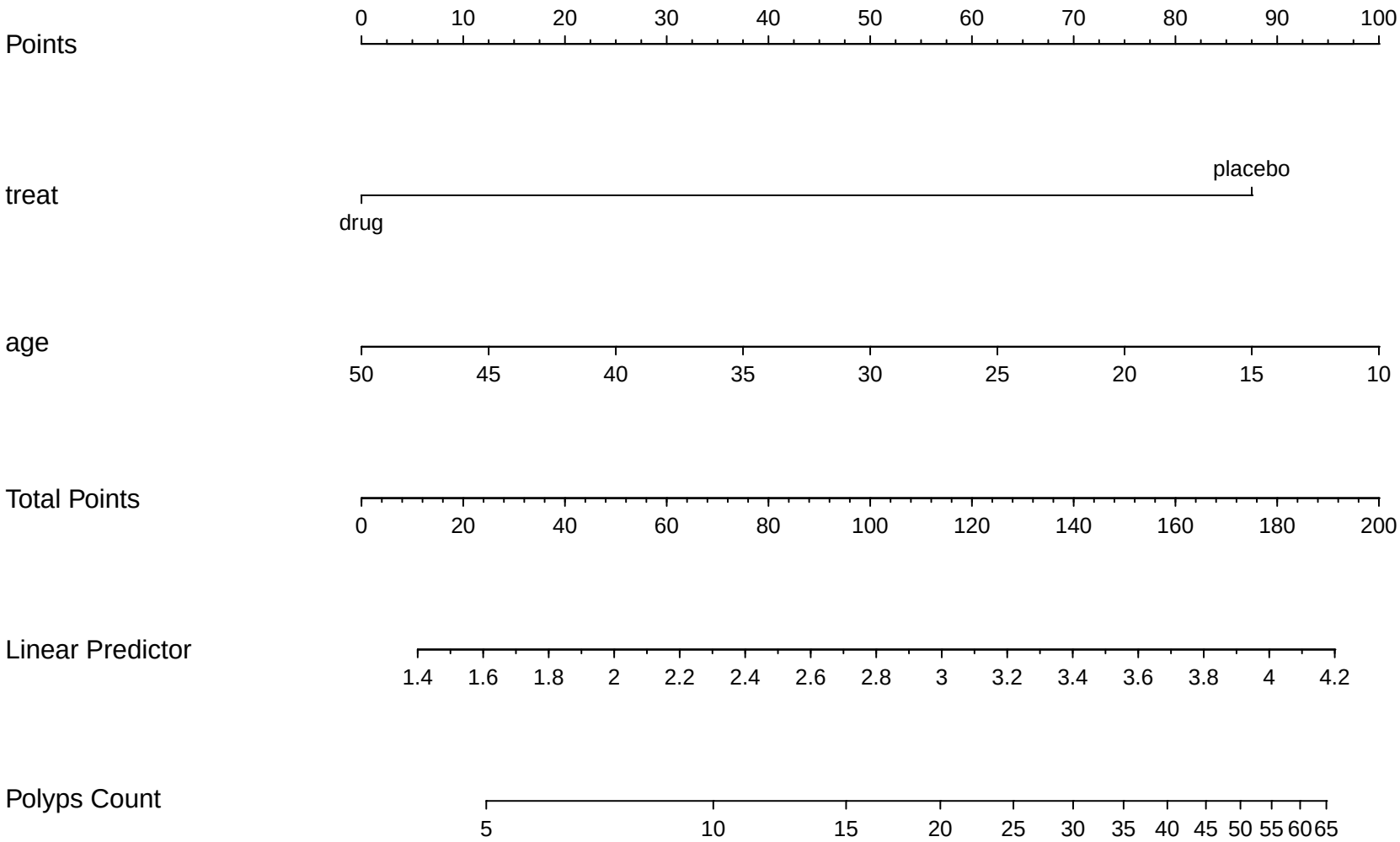


summary(model1)

```
             Effects                  Response : polyps12m

Factor                   Low   High  Diff. Effect   S.E.
age                      19.75 27.75  8    -0.31064 0.047642
treat - drug:placebo      1.00  2.00 NA    -1.35910 0.117640
Lower 0.95 Upper 0.95
-0.41116   -0.21013
-1.60730   -1.11090
```

Nomogram

```
plot(nomogram(model1, fun = exp, funlabel = "Polyps Count"))
```



Accounting for Overdispersion and Adding an Interaction Term

We can run an overdispersed model in `rms`, too. Just to mix things up a little, let's add an interaction term between `treat` and age and see if that improves our model at all.

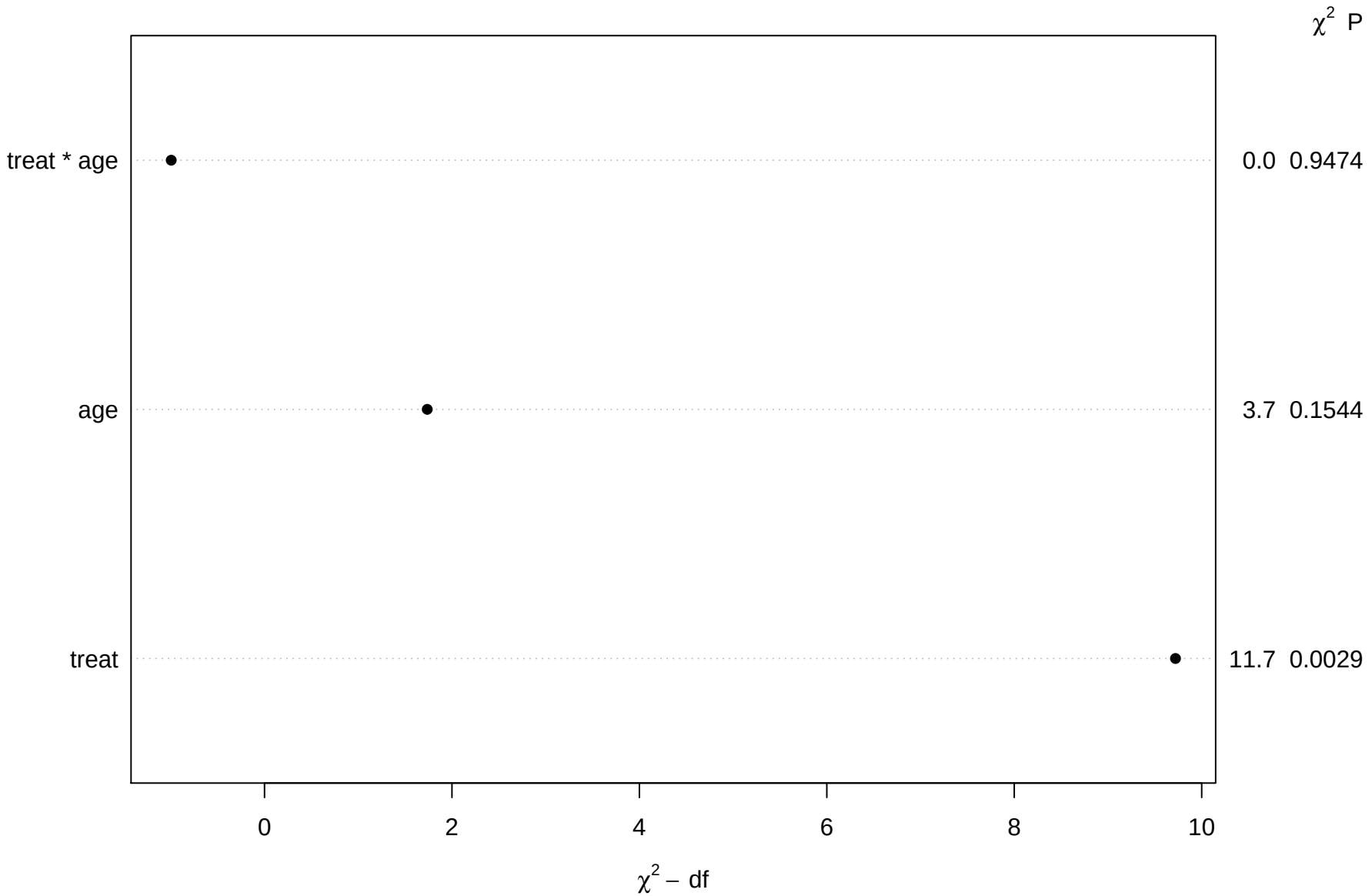
```
d <- datadist(pol432)
options(datadist = "d")
model3 <- Glm(polyps12m ~ treat * age, data=pol432,
              family=quasipoisson(), x = T, y = T)
```

model3

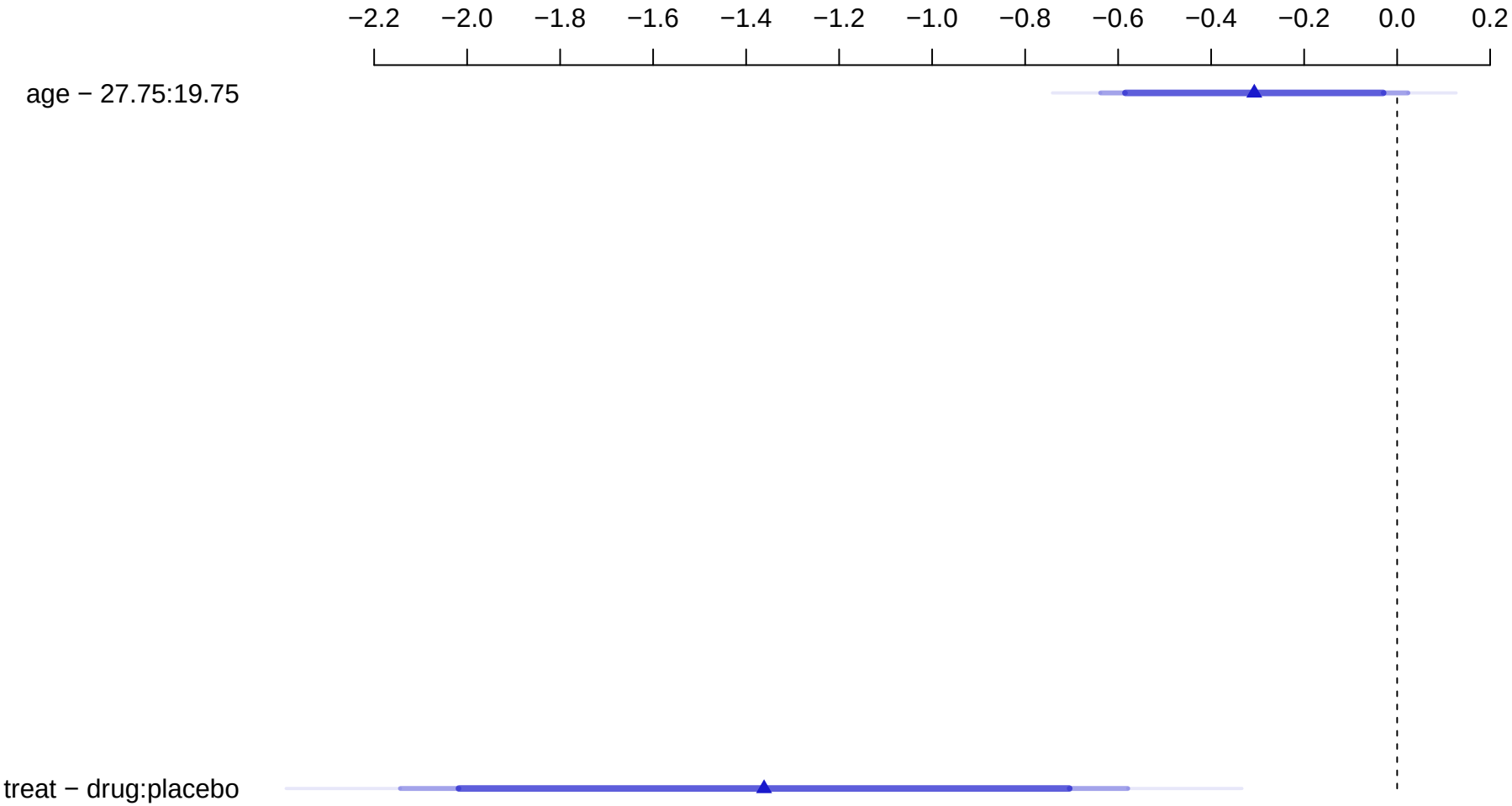
General Linear Model

```
Glm(formula = polyps12m ~ treat * age, family = quasipoisson(
    data = pol432, x = T, y = T)

                     Model Likelihood
                        Ratio Test
Obs          20        LR chi2     199.17
Residual d.f.16        d.f.             3
g 0.8406447            Pr(> chi2) <0.0001

                   Coef    S.E.   Wald Z Pr(>|Z|)
Intercept           4.5191 0.5173  8.74  <0.0001
treat=drug         -1.2573 1.5907 -0.79  0.4293
age                -0.0384 0.0211 -1.82  0.0683
treat=drug * age   -0.0046 0.0702 -0.07  0.9474
```

ANOVA: model3



Effects Summary Plot: model3



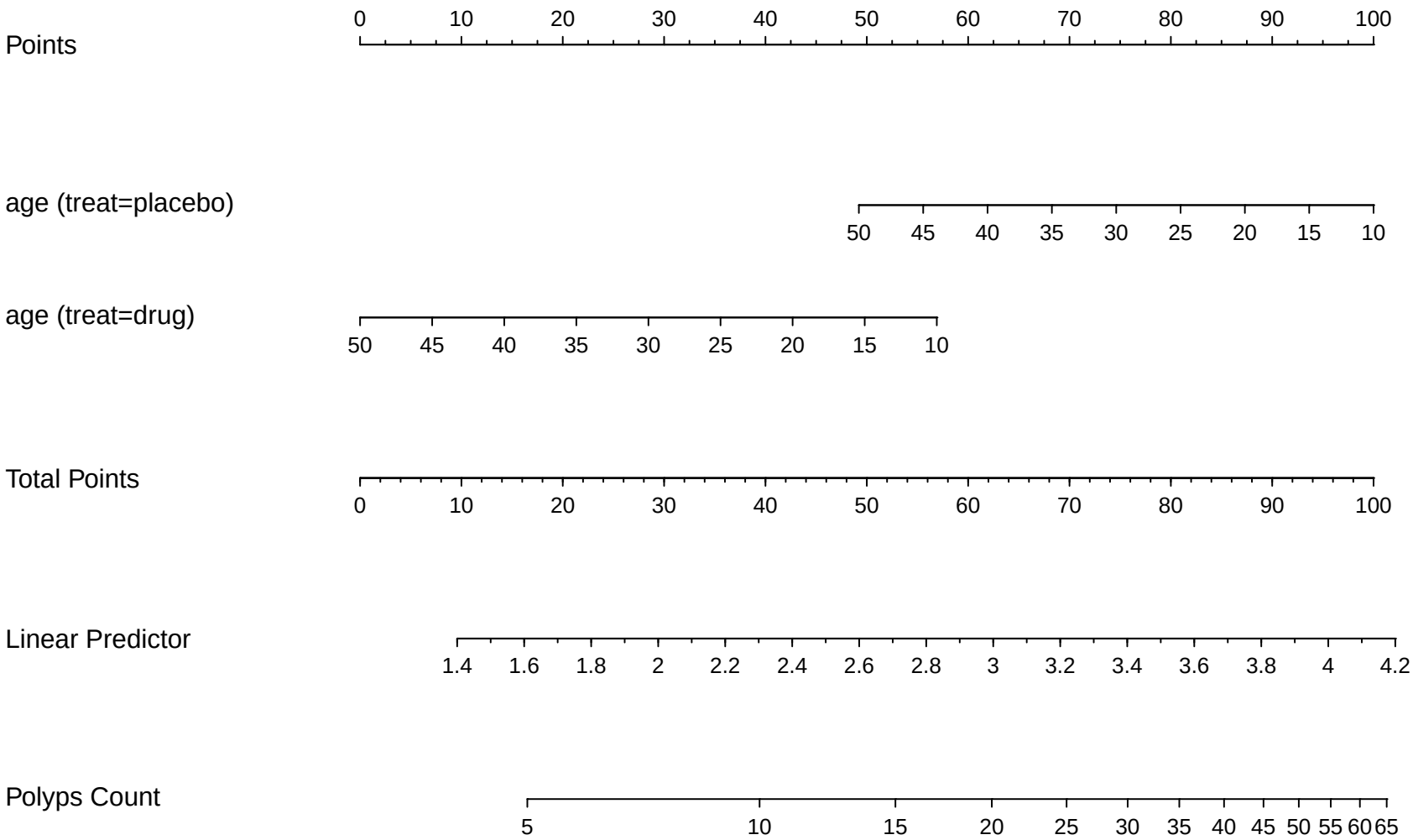
Adjusted to:treat=placebo age=22.5

Effects Summary Table: model3

```
              Effects                    Response : polyps12m

 Factor                    Low   High  Diff. Effect   S.E.
 age                       19.75 27.75  8    -0.30722 0.16851
 treat - drug:placebo       1.00  2.00 NA    -1.36140 0.39898
 Lower 0.95 Upper 0.95
 -0.66446    0.050012
 -2.20720   -0.515650

Adjusted to: treat=placebo age=22.5
```

Nomogram: model3

```
plot(nomogram(model3, fun = exp, funlabel = "Polyps Count"))
```



Negative Binomial Regression

To fit a negative binomial regression model to predict the log(polyp counts), I'd use the `glm.nb` function from the `MASS` package, as follows. . .

```
mod_nb <- glm.nb(polyps12m ~ treat + age,
                 data=pol432, link = log)

mod_nb
```

```
Call:  glm.nb(formula = polyps12m ~ treat + age, data = pol432
    init.theta = 1.719491)

Coefficients:
(Intercept)    treatdrug          age
    4.52603     -1.36812     -0.03856

Degrees of Freedom: 19 Total (i.e. Null);  17 Residual
Null Deviance:      36.73
```

summary(mod_nb)

```
Call: glm.nb(formula = polyps12m ~ treat + age,
    data = pol432, link = log, init.theta = 1.719491)

             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.52603    0.59466   7.611 2.72e-14 ***
treatdrug   -1.36812    0.36903  -3.707 0.000209 ***
age         -0.03856    0.02095  -1.840 0.065751 .

(Dispersion parameter for Negative Binomial(1.7195)
                                    family taken to be 1)

    Null deviance: 36.734  on 19  degrees of freedom
Residual deviance: 22.002  on 17  degrees of freedom
AIC: 164.88

Number of Fisher Scoring iterations: 1
```

Estimates and Standard Errors

Model	Poisson	Quasi-Poisson	Negative Binomial
Intercept	4.53 (0.147)	4.53 (0.481)	4.53 (0.594)
`treat`	-1.36 (0.118)	-1.36 (0.385)	-1.37 (0.369)
`age`	-0.039 (0.006)	-0.039 (0.020)	-0.039 (0.021)

Confidence Intervals for Coefficients

```
confint(mod_nb)
```

```
Waiting for profiling to be done...
```

	2.5 %	97.5 %
(Intercept)	3.3812139	5.716737794
treatdrug	-2.0960752	-0.633666100
age	-0.0776086	0.004319903

ANOVA for Negative Binomial Regression

The best way to run an ANOVA in this setting is to fit the model with and without the parameter you want to test.

```
mod_nb_notreat <- glm.nb(polyps12m ~ age,
                         data=pol432, link = log)

anova(mod_nb_notreat, mod_nb)
```

```
Likelihood ratio tests of Negative Binomial Models

Response: polyps12m
        Model    theta Resid. df    2 x log-lik.   Test
1         age 1.042819         18         -166.818
2 treat + age 1.719491         17         -156.880 1 vs 2
     df LR stat.     Pr(Chi)
1
2     1 9.937965 0.001619045
```

ANOVA for age in Negative Binomial model

```
mod_nb_noage <- glm.nb(polyps12m ~ treat,
                       data=pol432, link = log)

anova(mod_nb_noage, mod_nb)
```

```
Likelihood ratio tests of Negative Binomial Models

Response: polyps12m
        Model    theta Resid. df    2 x log-lik.   Test
1       treat 1.467894         18       -159.8004
2 treat + age 1.719491         17       -156.8800 1 vs 2
     df LR stat.    Pr(Chi)
1
2     1 2.920354 0.08746867
```

Making Predictions, with the Negative Binomial Model

```
predict(mod_nb, data.frame(treat = "drug", age = 30),
        type = "response", se.fit = TRUE)
```

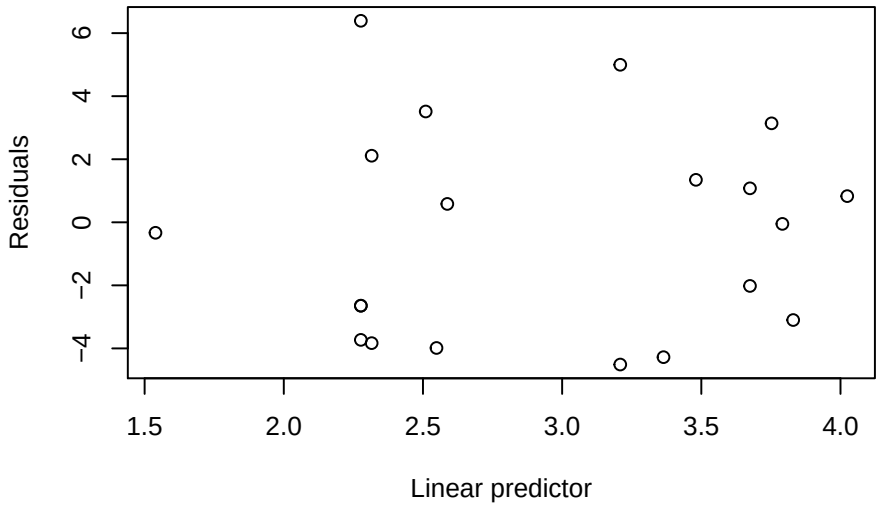
```
$fit
       1
7.397731

$se.fit
       1
2.310031

$residual.scale
[1] 1
```

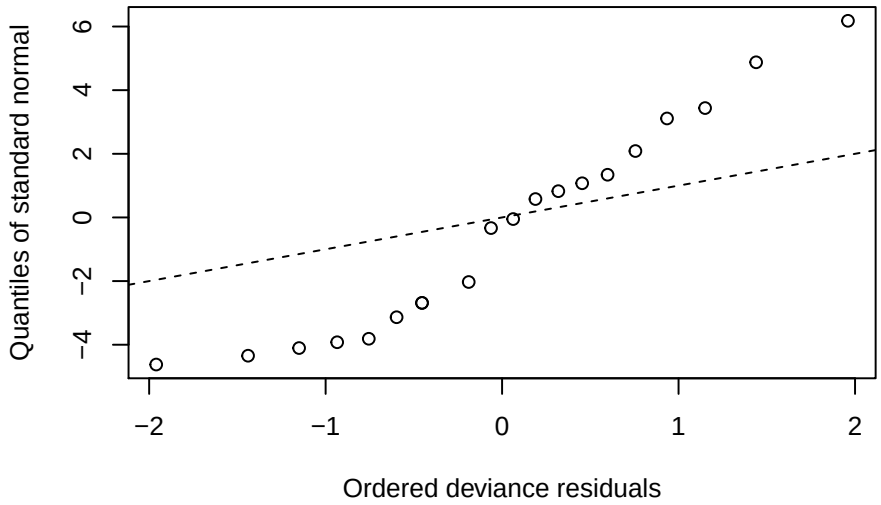
Diagnostic Plots for a Generalized Linear Model

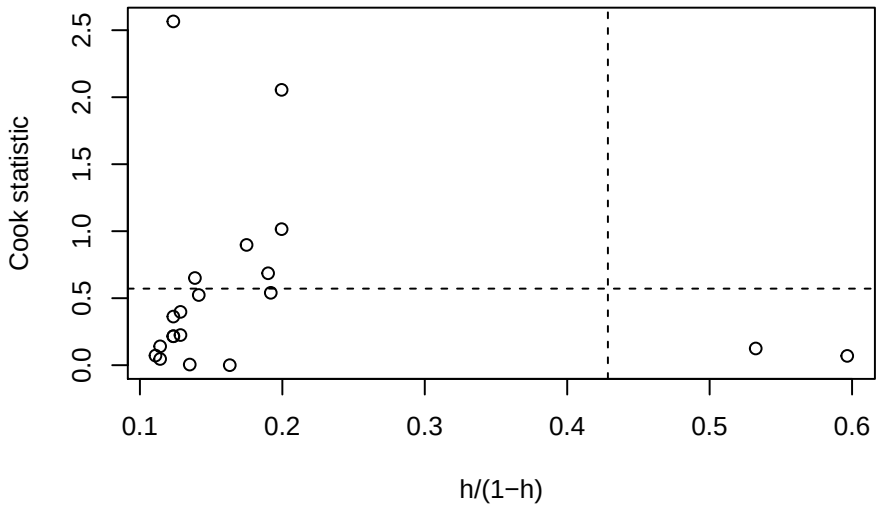
The `plots` function applied to one of these glms doesn't do anything that's too helpful. The `glm.diag.plots` function from the `boot` package makes a set of four plots:

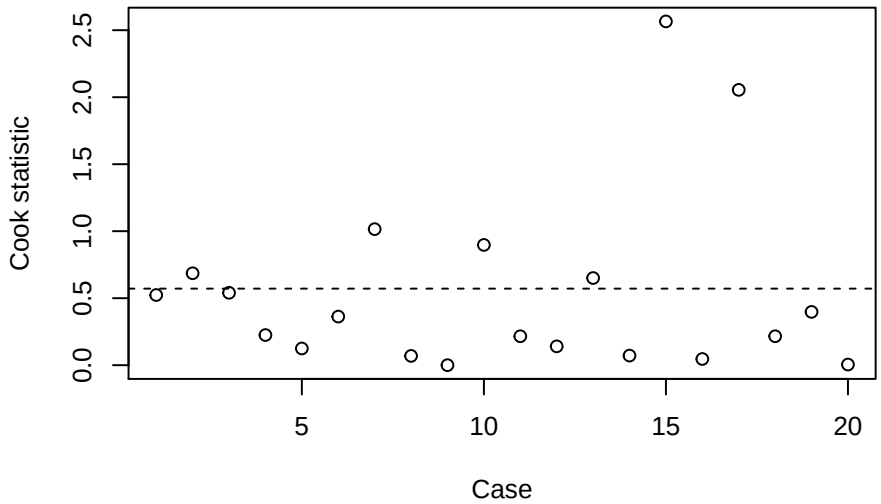
- (Top, Left) Jackknife deviance residuals against fitted values. A *jackknife deviance* residual, also called a likelihood residual, is the change in deviance when this observation is omitted.
- (Top, Right) Normal Q-Q plot of standardized deviance residuals. (Dotted line shows expectation if those standardized residuals followed a Normal distribution, and these residuals generally should.)
- (Bottom, Left) Cook statistic vs. standardized leverage
  - n = # of observations, p = # of parameters estimated
  - Horizontal dotted line at $\frac{8}{n-2p}$. Points above line have high influence.
  - Vertical line is at $\frac{2p}{n-2p}$. Points to the right of the line have high leverage.
- (Bottom, Right) Index plot of Cook's statistic to identify observations with high influence.

glm.diag.plots(mod_1)

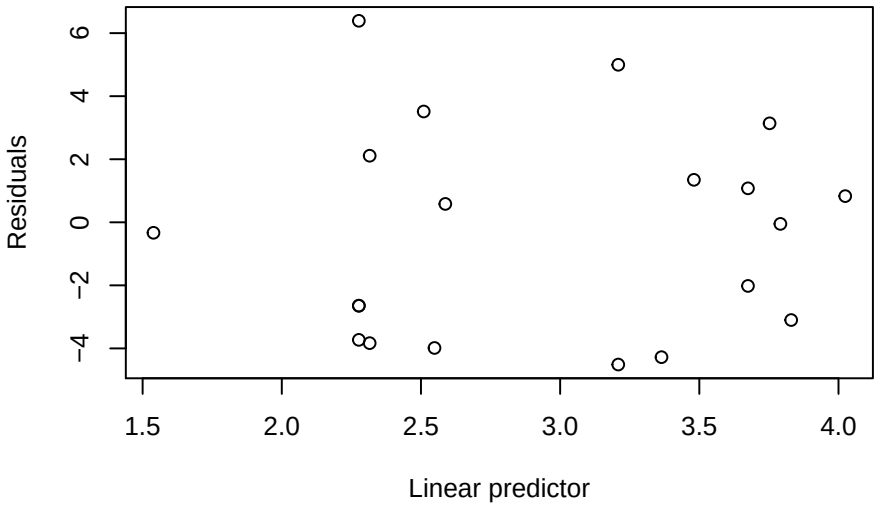


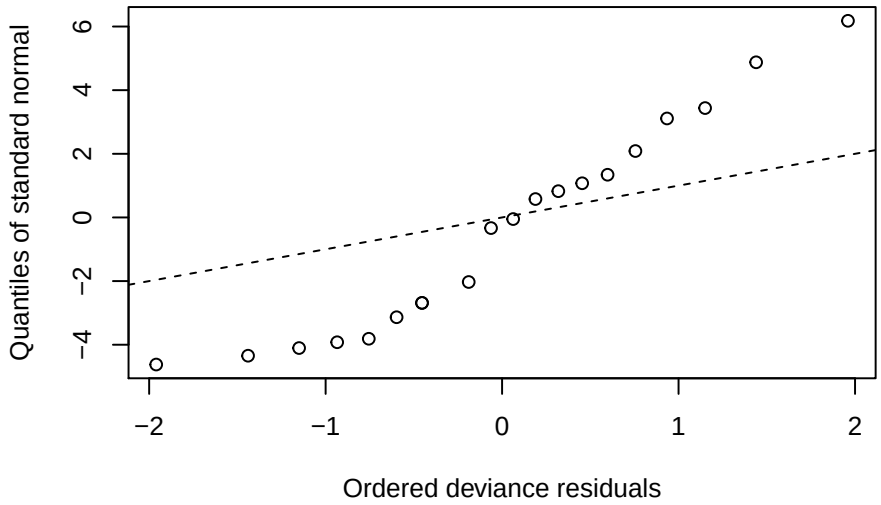


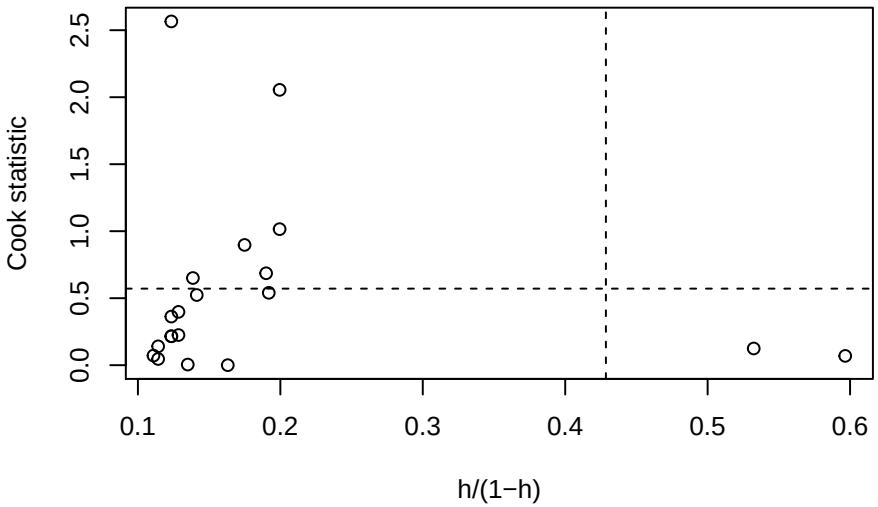


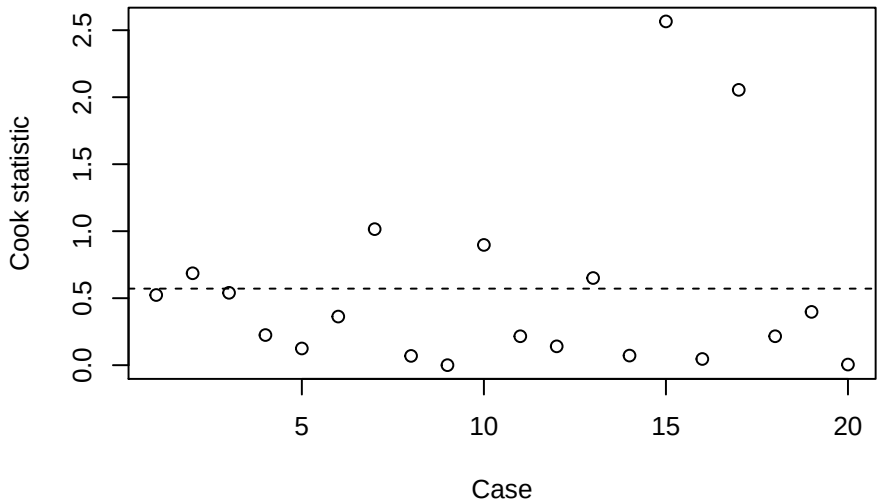


Diagnostic Plots for `mod_2` are similar

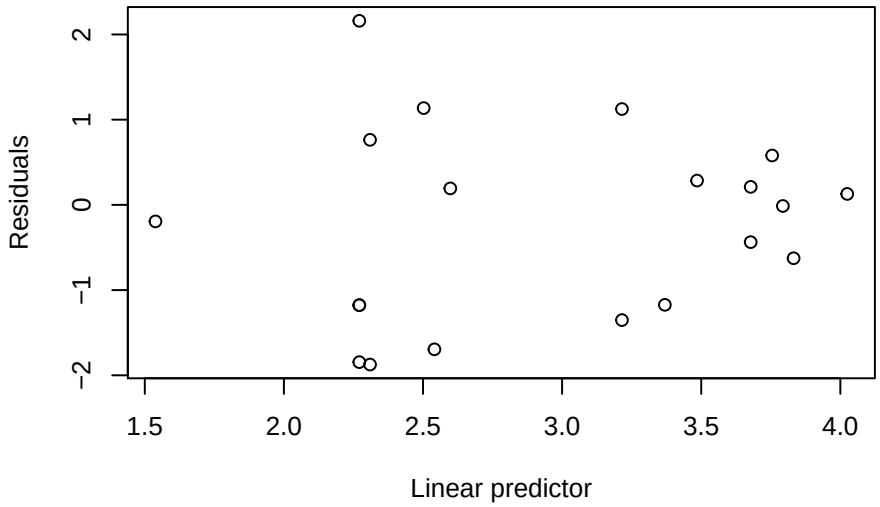


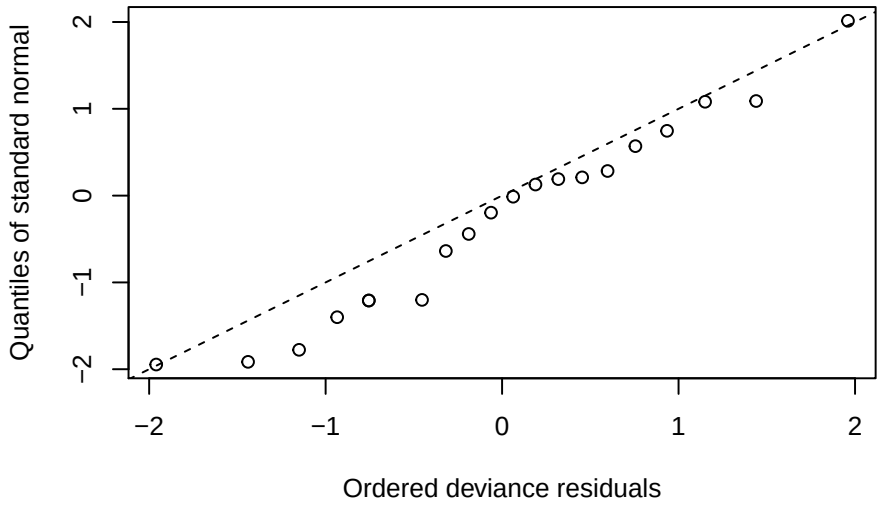


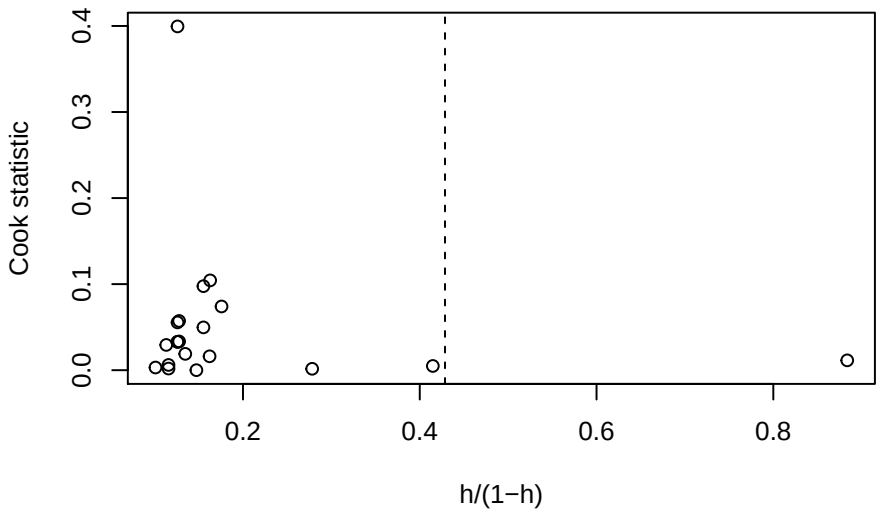


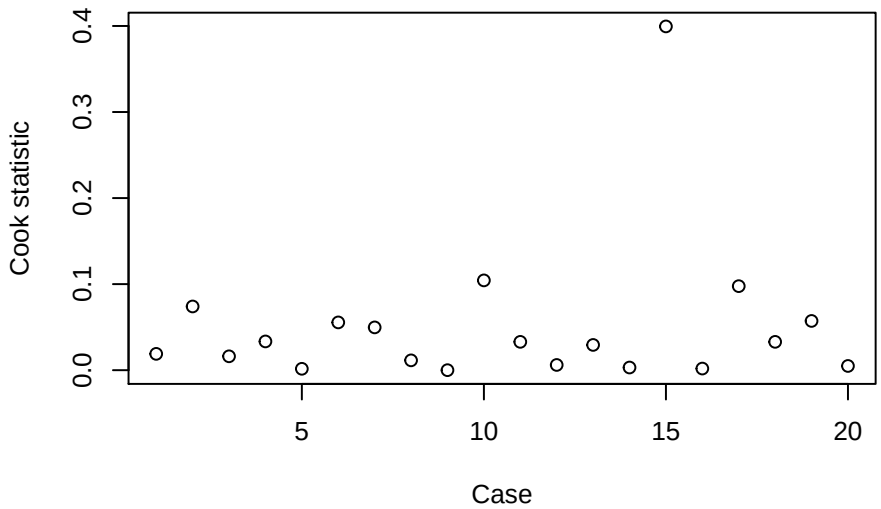


Diagnostic Plots for `mod_nb`









Model Comparisons

Summary	Poisson	QuasiPoisson	Negative Binomial
log(counts) =	4.53 - 1.36	4.53 - 1.36	4.53 - 1.37
...	treat - 0.039	treat - 0.039	treat - 0.039
	age	age	age
treat effect[1]	1.36 (1.13, 1.60)	1.36 (0.65, 2.18)	1.37 (0.63, 2.10)
age effect	-0.04 (-0.05, -0.03)	-0.04 (-0.08, 0)	-0.04 (-0.08, 0)

[1] Here we display the effect of being in the placebo group as compared to the drug group.

Model Comparisons

Summary	Poisson	Quasi-Poisson	Negative Binomial
Residual Deviance	179.5	179.5	22.0
Residual df	17	17	17
AIC	273.88	NA	164.9
Age 30, drug	7.43 (se = 0.86)	7.43 (2.83)	7.40 (2.31)
Deviance residuals	Not Normal	Not Normal	More Normal
Influential cases	15, 17	15, 17	maybe 15

Conclusion in this setting: It looks like the Negative Binomial model works best, of these options.

Next Time

More from our Notes: Chapter 18

- Zero-Inflated Models
- Hurdle Models
- Tobit Models