HW5Blank

Anna Cook

10/2/2020

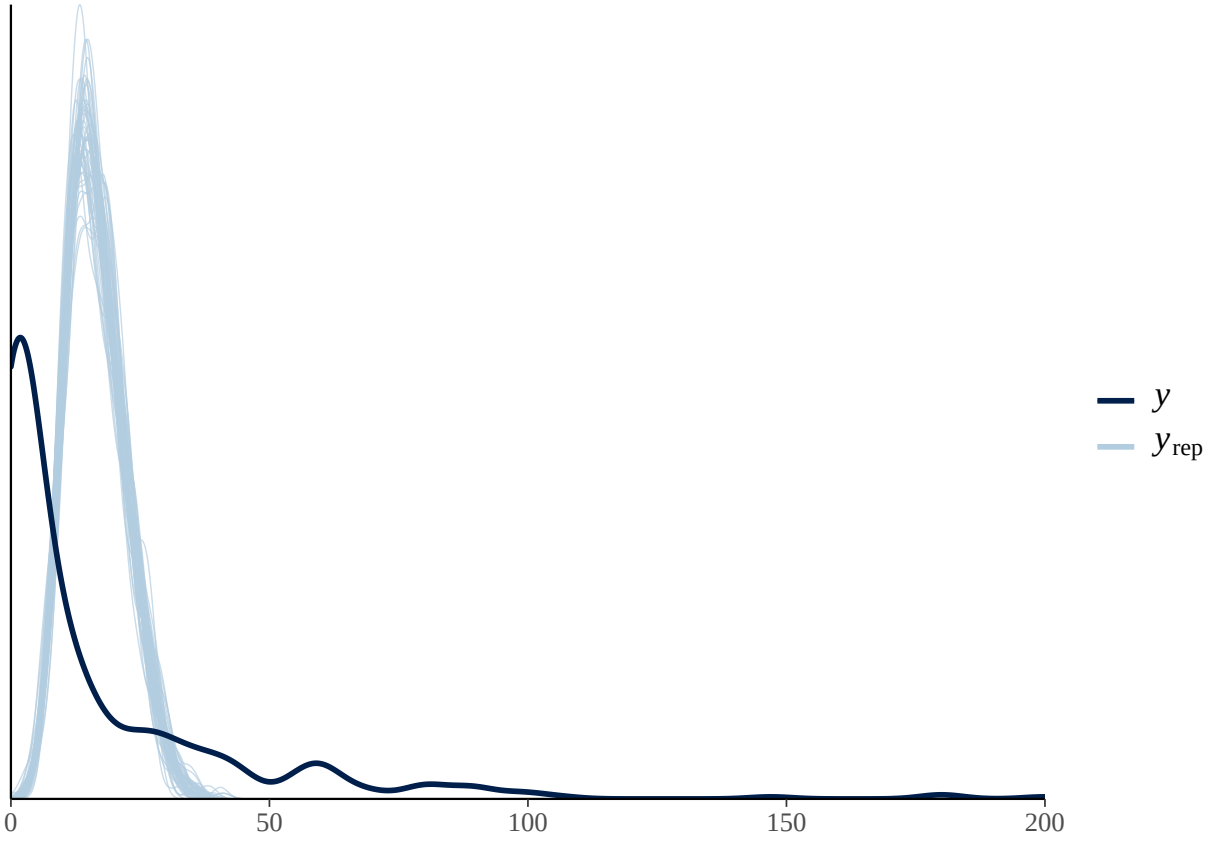
15.1 Poisson and negative binomial regression:

The folder RiskyBehavior contains data from a randomized trial targeting couples at high risk of HIV infection. The intervention provided counseling sessions regarding practices that could reduce their likelihood of contracting HIV. Couples were randomized either to a control group, a group in which just the woman participated, or a group in which both members of the couple participated. One of the outcomes examined after three months was "number of unprotected sex acts."

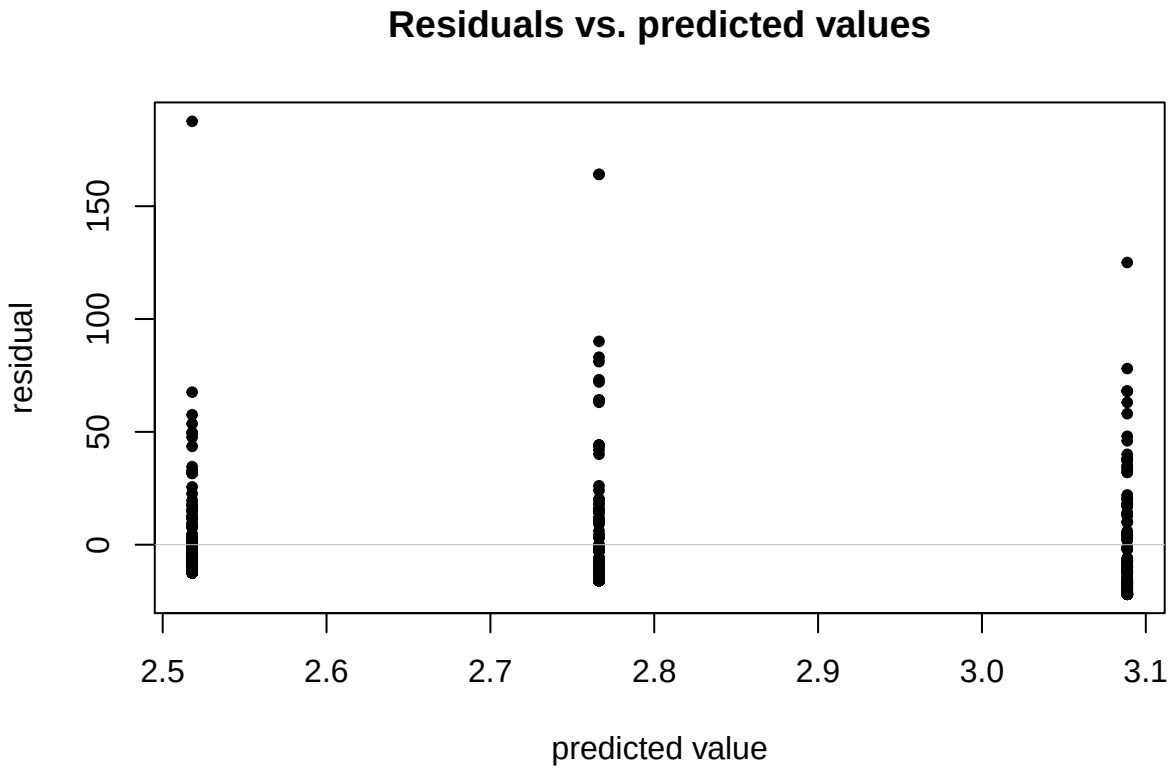
a)

Model this outcome as a function of treatment assignment using a Poisson regression. Does the model fit well? Is there evidence of overdispersion?

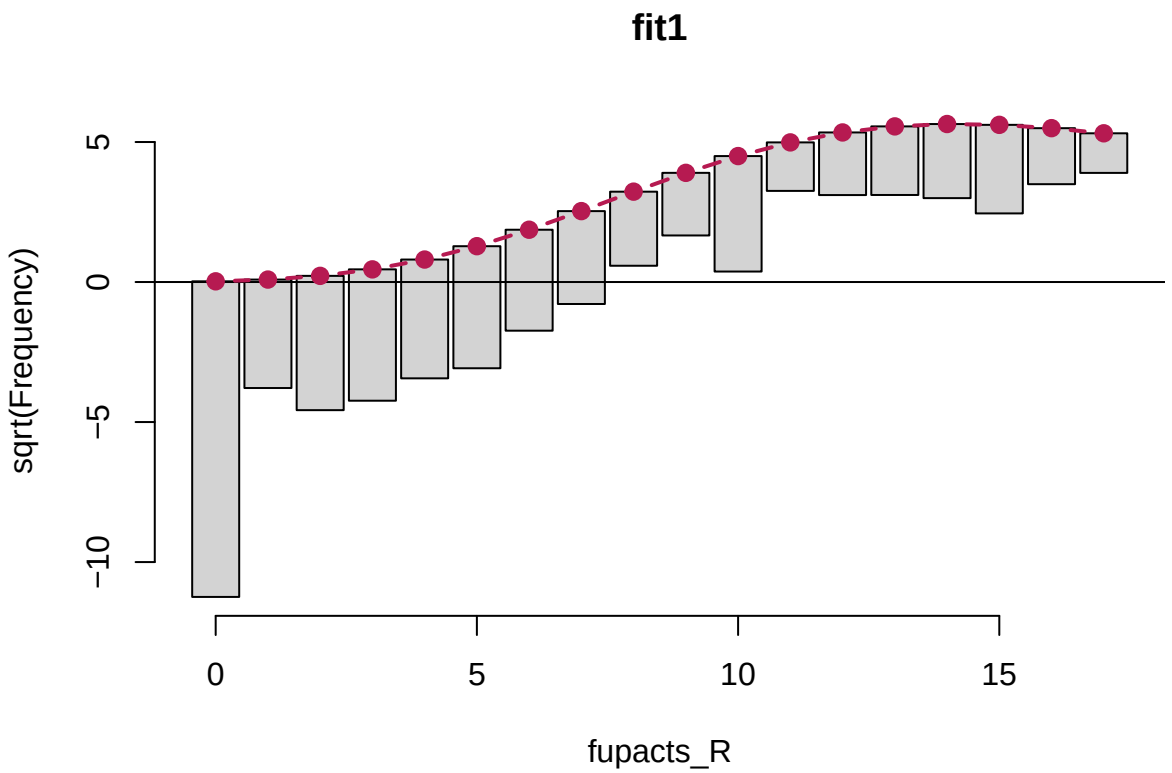
```
risk <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/RiskyBehavior/data/risk
risk$fupacts_R = round(risk$fupacts)

fit1 <- stan_glm(fupacts_R ~ couples + women_alone, family = poisson(link = "log"), data = risk, refresh

pp_check(fit1)
```




```
predicted <- predict(fit1)
residuals <- resid(fit1)
plot(predicted, residuals, xlab="predicted value", ylab="residual",
     main="Residuals vs. predicted values", pch=20)
abline(0, 0, col="gray", lwd=.5)
```




```
rootogram(fit1)
```



Yes, there is evidence of overdispersion. The residuals are huge, and the posterior predictive check shows that the model is not fitting the data well. The rootogram shows that at some points the model is over-predicting and at some points it is under-predicting.

To summarize:

- `sex` is the sex of the person, recorded as "man" or "woman" here
- `couples` is an indicator for if the couple was counseled together
- `women_alone` is an indicator for if the woman went to counseling by herself
- `bs_hiv` indicates if the individual is HIV positive
- `bupacts` is the number of unprotected sex acts reported as a baseline (before treamtnet)
- `fupacts` is the number of unprotected sex acts reported at the end of the study

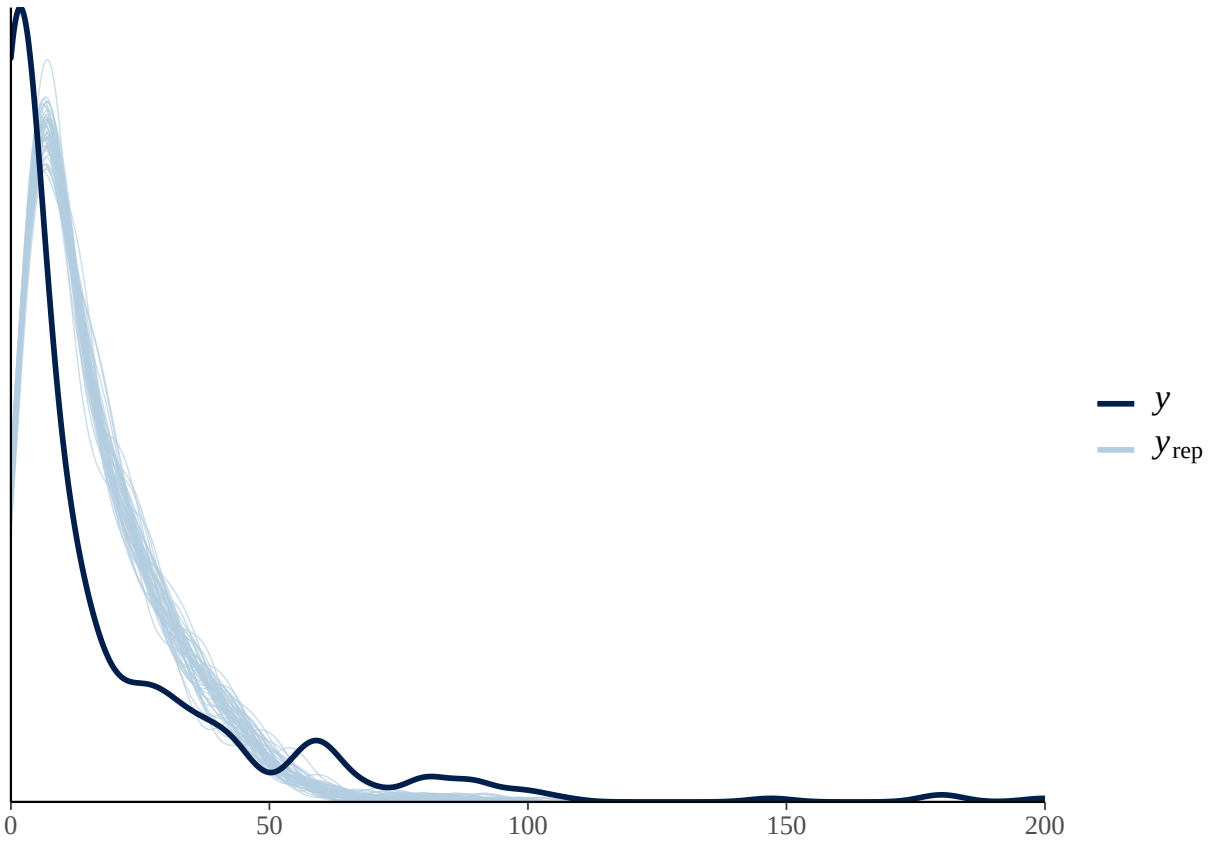
b)

Next extend the model to include pre-treatment measures of the outcome and the additional pre-treatment variables included in the dataset. Does the model fit well? Is there evidence of overdispersion?

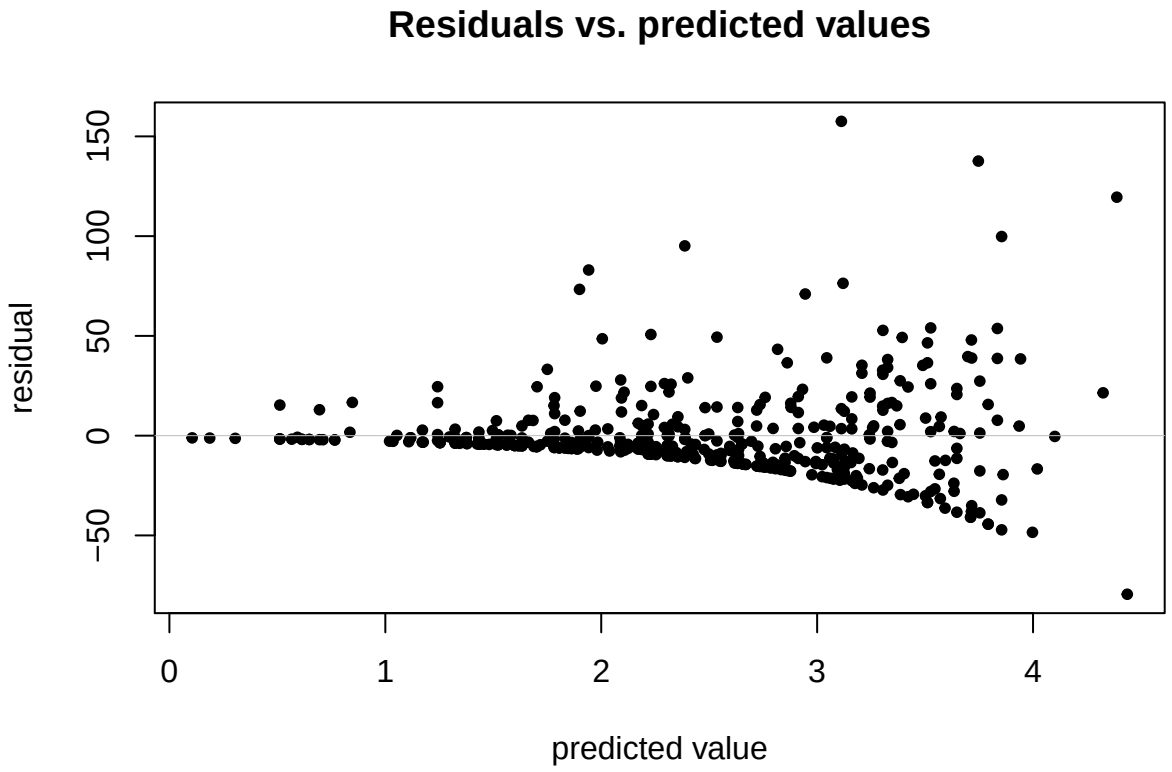
```
fit2 <- stan_glm(fupacts_R ~ couples + women_alone + bs_hiv + log(bupacts + 1) + sex, family = poisson(
fit2
```

```
## stan_glm
##  family:       poisson [log]
##  formula:      fupacts_R ~ couples + women_alone + bs_hiv + log(bupacts + 1) +
##     sex
##  observations: 434
##  predictors:   6
## ------
##                   Median MAD_SD
## (Intercept)        1.0    0.0
## couples           -0.3    0.0
## women_alone       -0.5    0.0
## bs_hivpositive    -0.4    0.0
## log(bupacts + 1)   0.7    0.0
## sexwoman           0.1    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

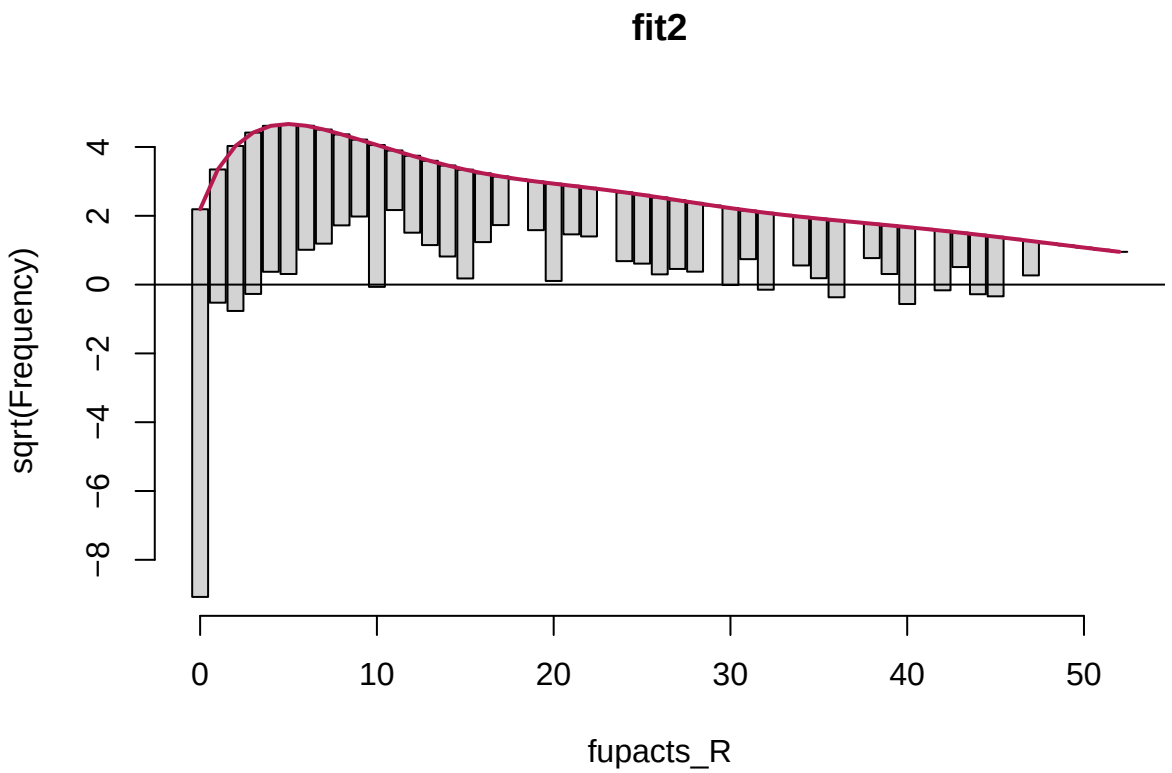
```
pp_check(fit2)
```




```
predicted <- predict(fit2)
residuals <- resid(fit2)
plot(predicted, residuals, xlab="predicted value", ylab="residual",
     main="Residuals vs. predicted values", pch=20)
abline(0, 0, col="gray", lwd=.5)
```




```
rootogram(fit2)
```



The model does not fit well (based on pp_check plot) and there is evidence of overdispersion (based on residual plot being very spread out). The rootogram also shows that the model is mostly over-predicting.

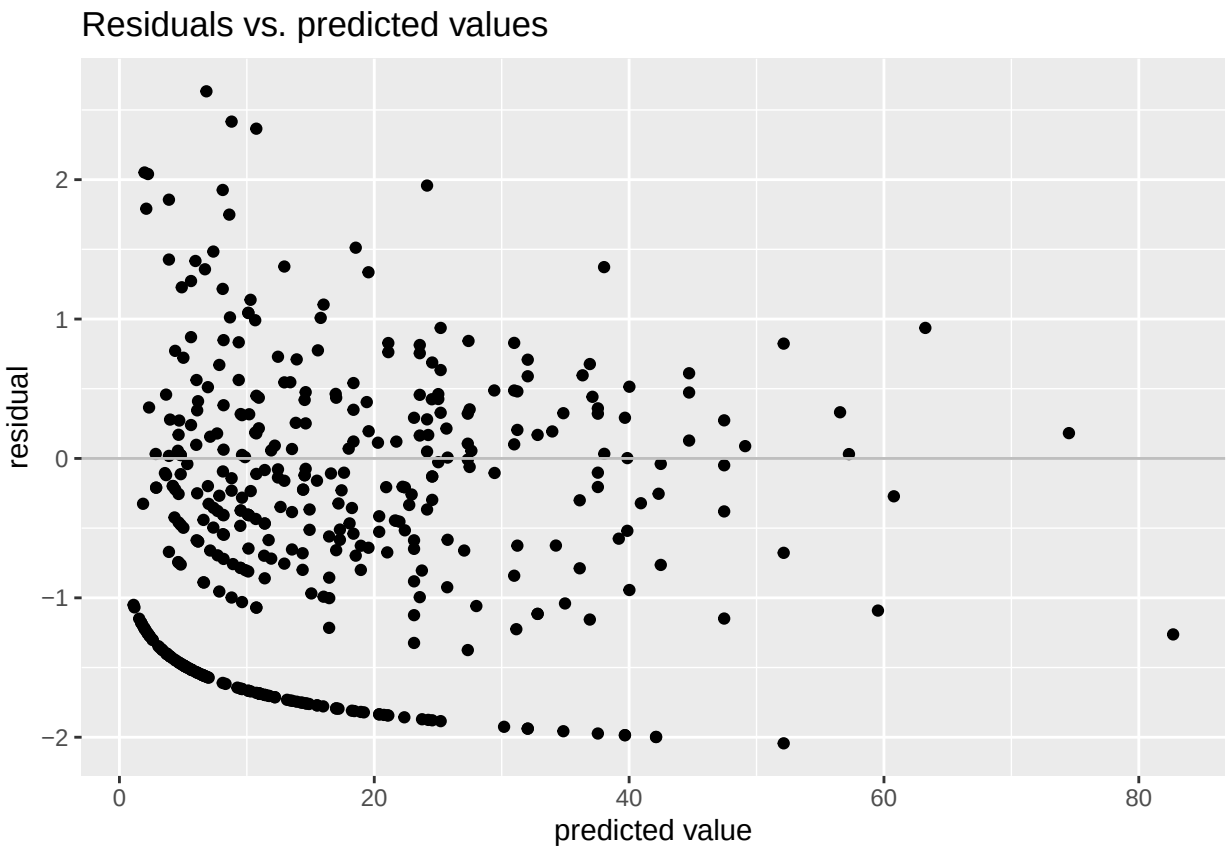
c)

Fit a negative binomial (overdispersed Poisson) model. What do you conclude regarding effectiveness of the intervention?

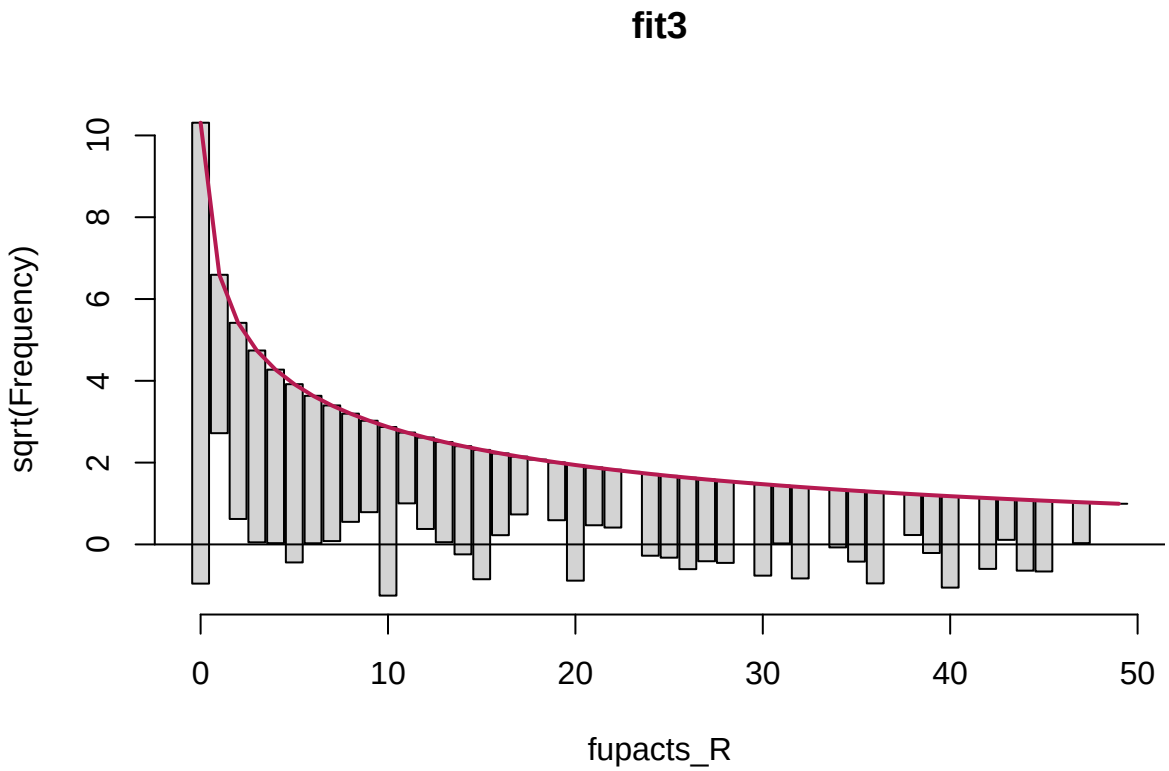
```
fit3 <- glm.nb(fupacts_R ~ couples + women_alone + bs_hiv + log(bupacts + 1) + sex, data=risk, link="log")
fit3
```

```
##
## Call:  glm.nb(formula = fupacts_R ~ couples + women_alone + bs_hiv +
##     log(bupacts + 1) + sex, data = risk, link = "log", init.theta = 0.4357586657)
##
## Coefficients:
##      (Intercept)           couples       women_alone    bs_hivpositive
##          1.31804          -0.36679          -0.64007          -0.51314
## log(bupacts + 1)          sexwoman
##          0.61832          -0.05974
##
## Degrees of Freedom: 433 Total (i.e. Null);  428 Residual
## Null Deviance:       603.1
## Residual Deviance: 488   AIC: 2953
```

```
#pp_check(fit3)

ggplot()+
  geom_point(aes(x=predict(fit3, type="response"), y=resid(fit3)))+
  labs(x="predicted value", y="residual", title = "Residuals vs. predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```



```
rootogram(fit3)
```



You can conclude that the intervention is effective, but it appears to be more effective for women alone than for couples. This is shown by the women_alone coefficient having a larger absolute value than the couples. However, both of the coefficients are negative, suggesting that either intervention reduces the number of unprotected sex acts.

d)

These data include responses from both men and women from the participating couples. Does this give you any concern with regard to our modeling assumptions?

This could be an issue because if both people in the couple are responding, this means that we are getting double the amount of information for these couples, thus doubling the response variable for these observations. This also means that not all the data points are independent of one another, but there is some collinearity going on.

15.3 Binomial regression:

Redo the basketball shooting example on page 270, making some changes:

(a)

Instead of having each player shoot 20 times, let the number of shots per player vary, drawn from the uniform distribution between 10 and 30.

```
set.seed(100)
N <- 100
height <- rnorm(N, 72, 3)
p <- 0.4 + 0.1*(height - 72)/3
n <- round(runif(N, 10, 30), digits = 0)
```

```
for (i in 1:n) {
  y <- rbinom(N, i, p)
}
```

```
## Warning in 1:n: numerical expression has 100 elements: only the first used
```

```
data <- data.frame(n=n, y=y, height=height)
fit1 <- stan_glm(cbind(y, i-y) ~ height, family = binomial(link = "logit"), data = data, refresh = 0)
fit1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      cbind(y, i - y) ~ height
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -11.5    1.3
## height        0.2    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

(b)

Instead of having the true probability of success be linear, have the true probability be a logistic function, set so that Pr(success) = 0.3 for a player who is 5'9" and 0.4 for a 6' tall player.

```
N <- 100
height <- rnorm(N, 72, 3)
p <- invlogit(-.405 + .441*((height - 72)/3))
n <- round(runif(N, 10, 30), digits = 0)
for (i in 1:n) {
  y <- rbinom(N, i, p)
}
```

```
## Warning in 1:n: numerical expression has 100 elements: only the first used
```

```
data <- data.frame(n=n, y=y, height=height)
round(data$y, digits = 0)
```

```
##   [1]  5  7 14 11  8 13  8 11 11  9 15 18 11 11  7 13  7 17 14 11 13 14  7 10  2
##  [26] 14 15 13 11 13  9  9  9 11 11  9 10  8 15 13 12 13 15 11 16 14  8 17 14 11
##  [51] 13  9  8 13 11 11  8  9 13 15 11 11  9 13 13 14  9  2 15 12 10  8  9 15  8
##  [76] 12  9 13  9  6 11 14 13 16  4 12  8  8  9  8 20 19  7 11  5 11  4 16  8 11
```

```
fit2 <- stan_glm(cbind(y, i-y) ~ height, family = binomial(link = "logit"), data = data, refresh = 0)
fit2
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      cbind(y, i - y) ~ height
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -10.3    1.1
```

```
## height        0.1    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

15.7 Tobit model for mixed discrete/continuous data:

Experimental data from the National Supported Work example are in the folder Lalonde. Use the treatment indicator and pre-treatment variables to predict post-treatment (1978) earnings using a Tobit model. Interpret the model coefficients.

```
lalonde = foreign::read.dta("https://github.com/avehtari/ROS-Examples/blob/master/Lalonde/NSW_dw_obs.dta

fit <- vglm(re78 ~ treat + age + educ + black + hisp + married + nodegree + sample + educ_cat4, tobit()
summary(fit)
```

```
##
## Call:
## vglm(formula = re78 ~ treat + age + educ + black + hisp + married +
##     nodegree + sample + educ_cat4, family = tobit(), data = lalonde)
##
## Pearson residuals:
##                  Min      1Q  Median      3Q    Max
## mu          -2.6786 -0.5623  0.2785 0.76573  7.689
## loglink(sd) -0.9911 -0.6686 -0.3877 0.06843 61.550
##
## Coefficients:
##                 Estimate Std. Error  z value Pr(>|z|)
## (Intercept):1 -9.172e+03  8.814e+02  -10.406  < 2e-16 ***
## (Intercept):2  9.343e+00  5.635e-03 1658.059  < 2e-16 ***
## treat          3.693e+03  9.781e+02    3.776 0.000159 ***
## age            5.513e+01  8.662e+00    6.365 1.96e-10 ***
## educ           3.574e+02  6.952e+01    5.142 2.73e-07 ***
## black         -3.211e+03  2.979e+02  -10.779  < 2e-16 ***
## hisp          -6.430e+02  3.504e+02   -1.835 0.066470 .
## married        4.759e+03  2.137e+02   22.264  < 2e-16 ***
## nodegree      -1.005e+03  2.864e+02   -3.510 0.000448 ***
## sample         6.588e+03  2.551e+02   25.819  < 2e-16 ***
## educ_cat4      5.345e+02  1.994e+02    2.681 0.007349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: mu, loglink(sd)
##
## Log-likelihood: -176905.5 on 37323 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):2'
```

By looking at the signs of the coefficients, we can see that being black, hispanic, and having no degree all degrease expected post-treatment earnings. All other variables increase average expected earnings as the level of the predictor increases. For example, as a person's education

level increases, their average expected earings increases as well. In addition, the only variable that is not statistically significant is hisp, although the p-value is 0.06, so it is barely above the threshold for being considered "significant", so it is hard to say whether we are seeing a true effect or not.

15.8 Robust linear regression using the t model:

The folder Congress has the votes for the Democratic and Republican candidates in each U.S. congressional district in 1988, along with the parties' vote proportions in 1986 and an indicator for whether the incumbent was running for reelection in 1988. For your analysis, just use the elections that were contested by both parties in both years.

```
congress = read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Congress/data/congr
congress88 <- data.frame(vote=congress$v88_adj,pastvote=congress$v86_adj,inc=congress$inc88)
```

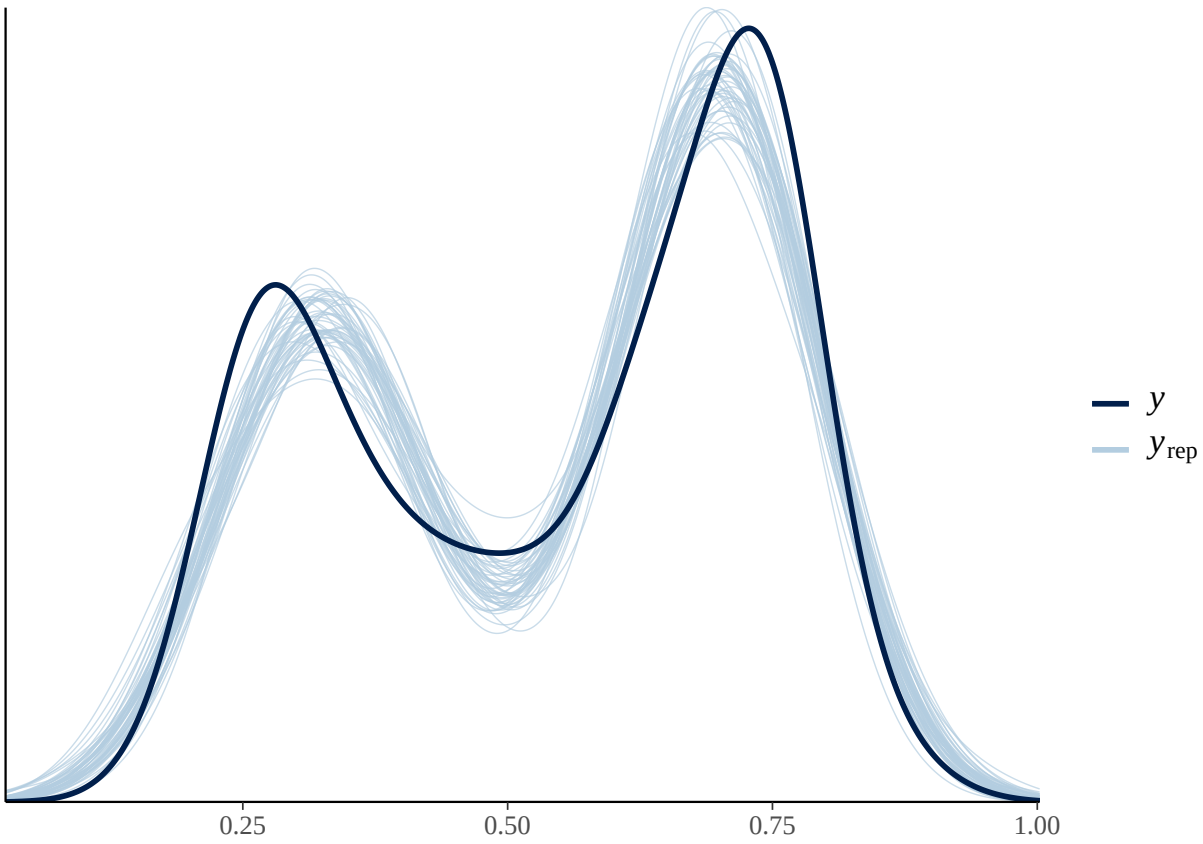
(a)

Fit a linear regression using stan_glm with the usual normal-distribution model for the errors predicting 1988 Democratic vote share from the other variables and assess model fit.

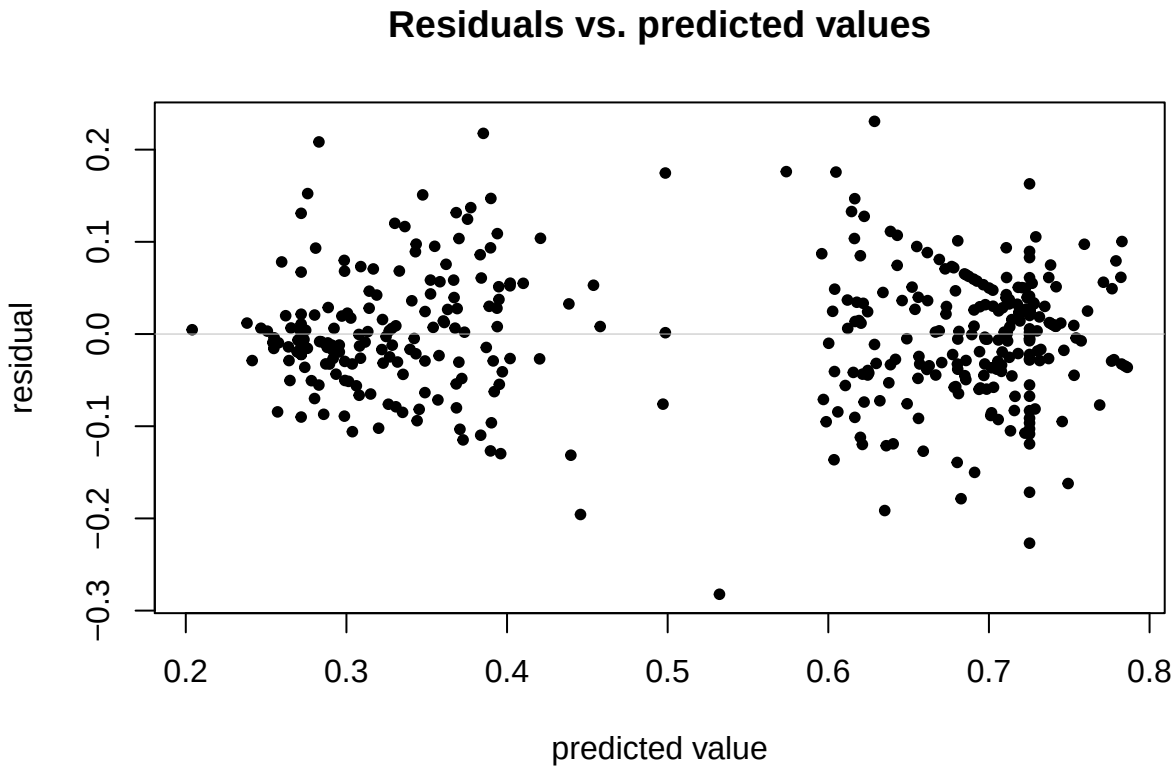
```
fit1 <- stan_glm(vote ~ pastvote + inc, data = congress88, refresh = 0)
fit1
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      vote ~ pastvote + inc
##  observations: 435
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 0.2    0.0
## pastvote    0.5    0.0
## inc         0.1    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.1    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
pp_check(fit1)
```




```
predicted <- predict(fit1)
residuals <- resid(fit1)
plot(predicted, residuals, xlab="predicted value", ylab="residual",
     main="Residuals vs. predicted values", pch=20)
abline(0, 0, col="gray", lwd=.5)
```



based on the pp_check plot and the residuals, this fit looks to be not great, but also not terrible. The pp_check plot shows that the model has the same bimodal shape as the simulations, but it seems a bit off. The residuals also show this bimodal pattern, but the residuals don't appear to be overdispersed.

(b)

Fit the same sort of model using the brms package with a t distribution, using the brm function with the student family. Again assess model fit.

```
fit2 <- brm(vote ~ pastvote + inc, data = congress88, family = student, refresh = 0)
```

```
## Compiling Stan program...
## Trying to compile a simple C file
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
##                 ^
##                 ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
```

```
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fa
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

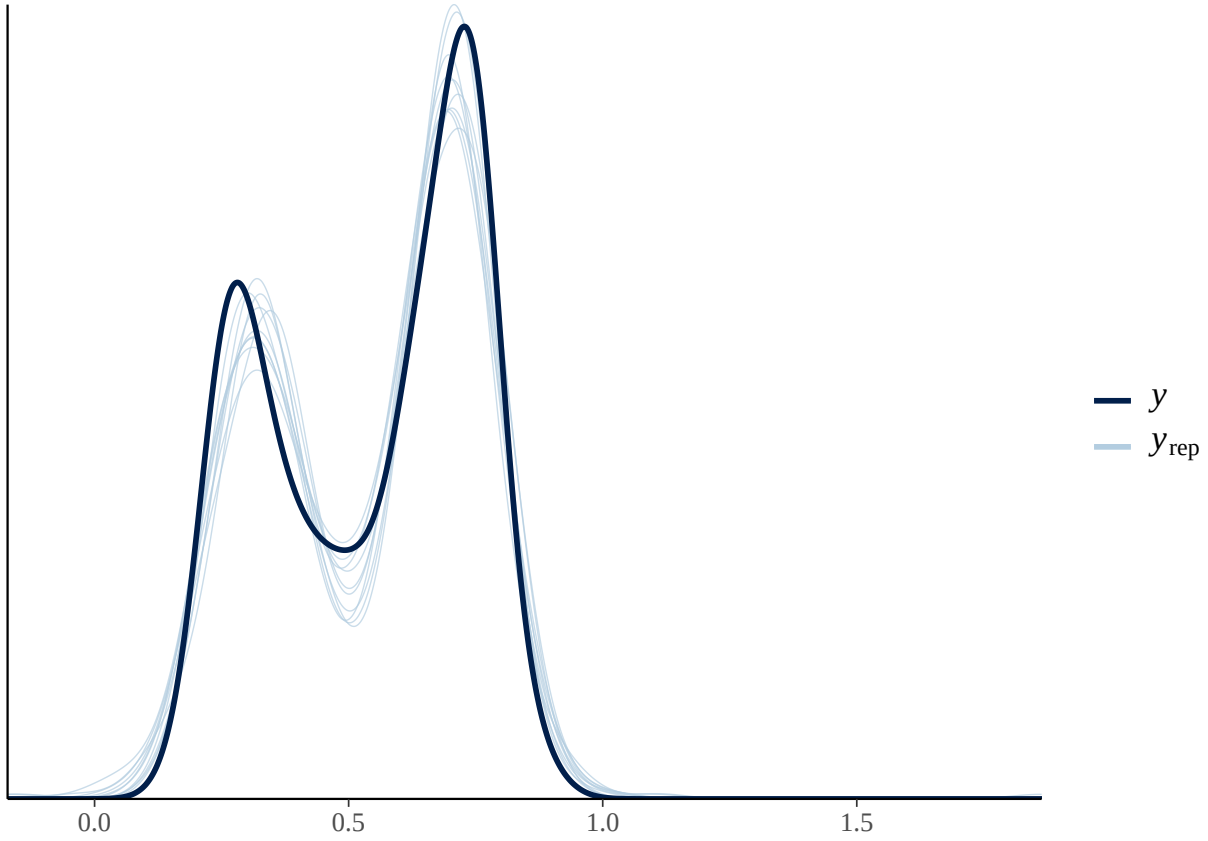
```
## Start sampling
```

```
summary(fit2)
```

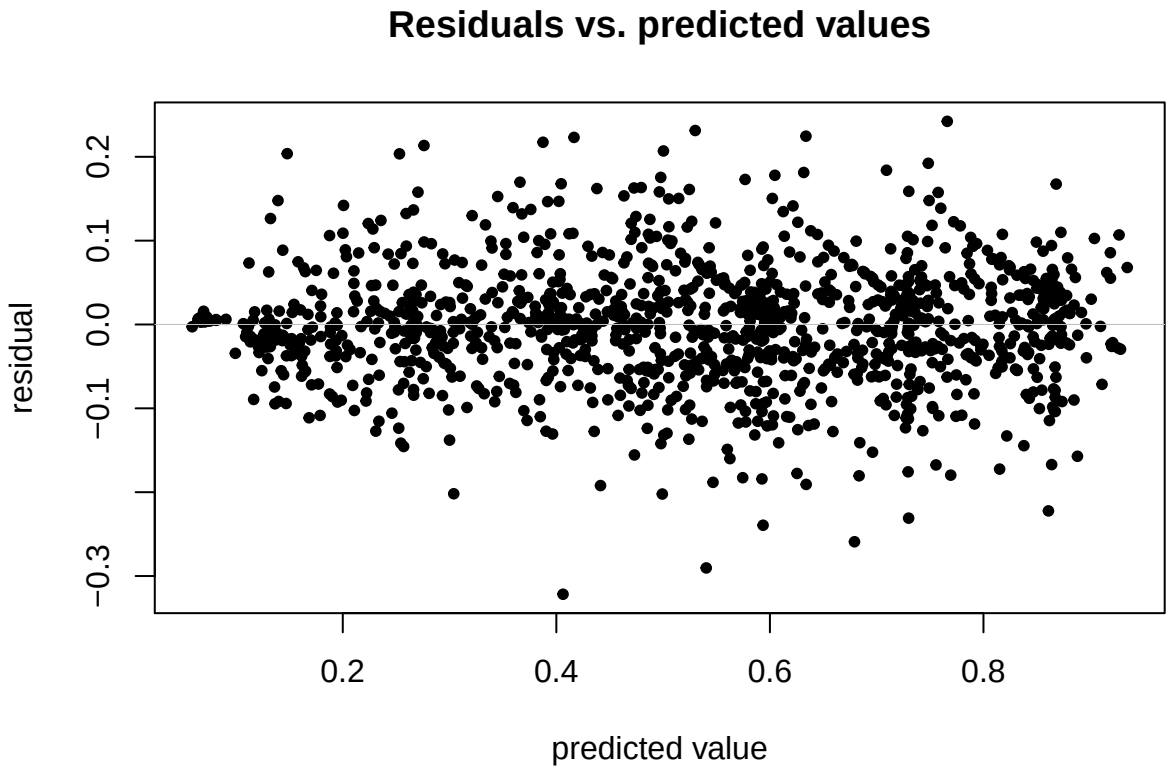
```
##  Family: student
##   Links: mu = identity; sigma = identity; nu = identity
## Formula: vote ~ pastvote + inc
##    Data: congress88 (Number of observations: 435)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.22      0.02     0.19     0.26 1.00     1979     2214
## pastvote      0.55      0.04     0.48     0.62 1.00     1915     2031
## inc           0.09      0.01     0.08     0.11 1.00     1983     2015
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.05      0.00     0.05     0.06 1.00     1764     1908
## nu        6.22      2.49     3.30    12.81 1.00     1822     1982
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
pp_check(fit2)
```

```
## Using 10 posterior samples for ppc type 'dens_overlay' by default.
```




```
predicted <- predict(fit2)
residuals <- resid(fit2)
plot(predicted, residuals, xlab="predicted value", ylab="residual",
     main="Residuals vs. predicted values", pch=20)
abline(0, 0, col="gray", lwd=.5)
```



Although the pp_check plot still isn't a perfect fit, the residuals no longer show a clear bimodal pattern, so this fit seems to be a bit better than the previous model.

(c)

Which model do you prefer?

I prefer the second model because the residuals look more evenly spread out instead of in a bimodal pattern, and the pp_check plot seems to be fitting just slightly better.

15.9 Robust regression for binary data using the robit model:

Use the same data as the previous example with the goal instead of predicting for each district whether it was won by the Democratic or Republican candidate.

(a)

Fit a standard logistic or probit regression and assess model fit.

```
congress88$winner <- ifelse(congress88$vote>0.5,1,0)
fit1 <- glm(winner ~ pastvote + inc, family = binomial(link = "logit"), data = congress88)
fit1
```

```
##
## Call:  glm(formula = winner ~ pastvote + inc, family = binomial(link = "logit"),
##     data = congress88)
##
## Coefficients:
## (Intercept)     pastvote          inc
##      -5.563       11.283        2.694
##
```

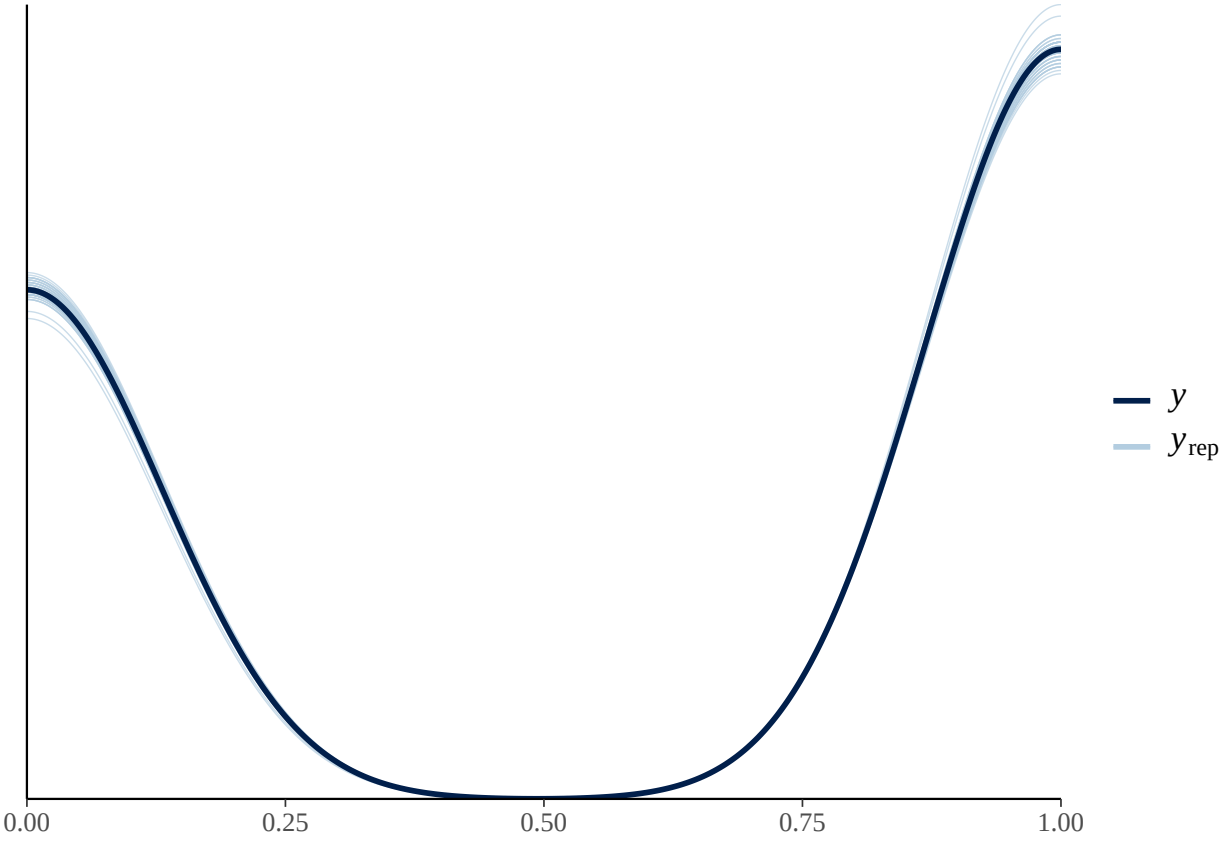
```
## Degrees of Freedom: 434 Total (i.e. Null);  432 Residual
## Null Deviance:       587.1
## Residual Deviance: 75.39     AIC: 81.39
```

```
fit1 <- stan_glm(winner ~ pastvote + inc, family = binomial(link = "logit"), data = congress88)
```

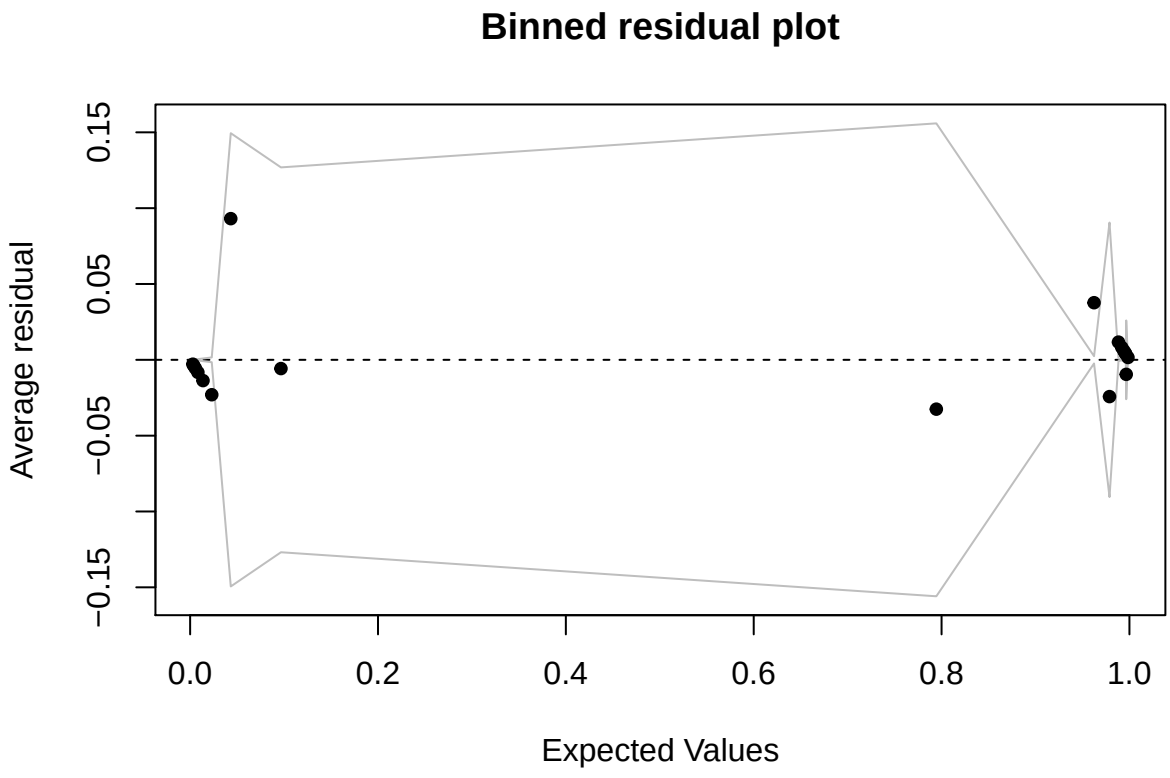
```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 6.6e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.66 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.208805 seconds (Warm-up)
## Chain 1:                0.203365 seconds (Sampling)
## Chain 1:                0.41217 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000114 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.14 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.182063 seconds (Warm-up)
## Chain 2:                0.220203 seconds (Sampling)
## Chain 2:                0.402266 seconds (Total)
```

```
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2.8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.28 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.190996 seconds (Warm-up)
## Chain 3:                0.196744 seconds (Sampling)
## Chain 3:                0.38774 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 3e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.3 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.17676 seconds (Warm-up)
## Chain 4:                0.175476 seconds (Sampling)
## Chain 4:                0.352236 seconds (Total)
## Chain 4:
```

```
pp_check(fit1)
```




```
binnedplot(fitted(fit1),resid(fit1))
```



```
# the residuals are a bit wacky but the pp_check plot appears to be fitting very well.
```

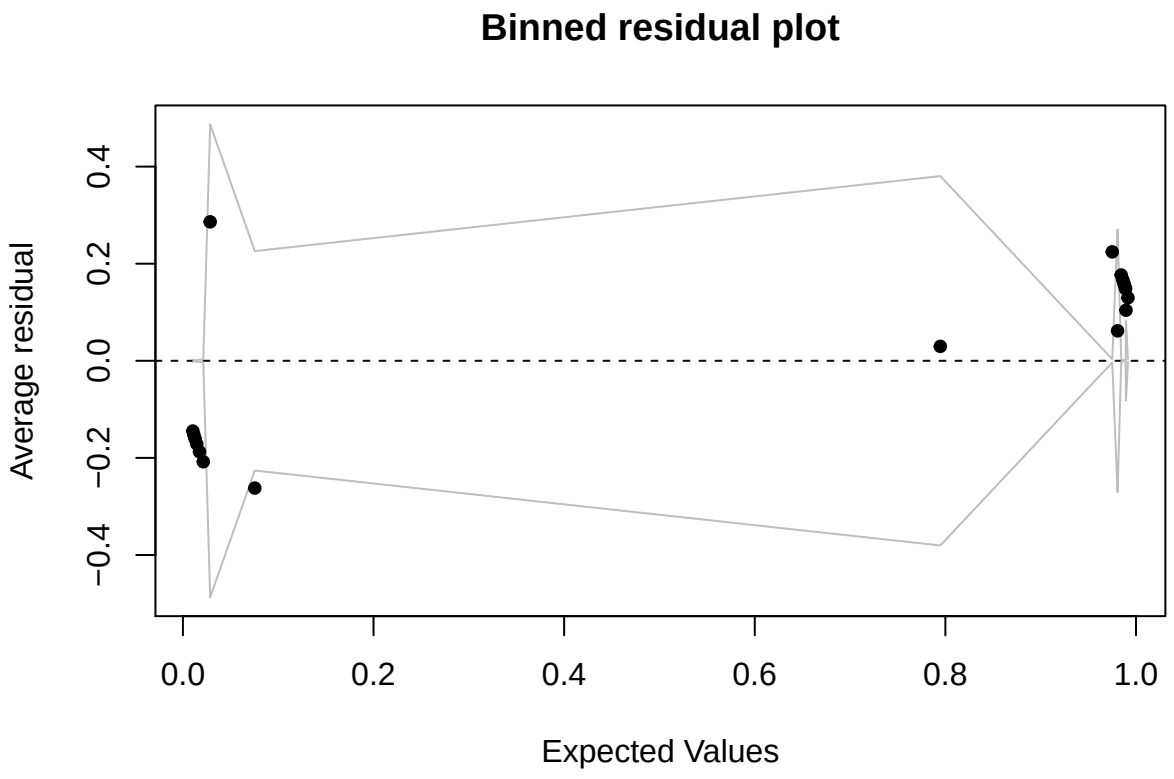
(b)

Fit a robit regression and assess model fit.

```
fit2 <- glm(winner ~ pastvote + inc, family = binomial(link = gosset(2)), data = congress88)
fit2
```

```
##
## Call:  glm(formula = winner ~ pastvote + inc, family = binomial(link = gosset(2)),
##     data = congress88)
##
## Coefficients:
## (Intercept)     pastvote          inc
##      -5.668       11.594        3.715
##
## Degrees of Freedom: 434 Total (i.e. Null);  432 Residual
## Null Deviance:       587.1
## Residual Deviance: 78.96     AIC: 84.96
```

```
binnedplot(fitted(fit2),resid(fit2))
```



(c)

Which model do you prefer?

I think I prefer the standard logistic model because it has a slightly lower residual deviance and AIC values, and the pp_check plot looked like a really good fit.

15.14 Model checking for count data:

The folder RiskyBehavior contains data from a study of behavior of couples at risk for HIV; see Exercise 15.1.

(a)

Fit a Poisson regression predicting number of unprotected sex acts from baseline HIV status. Perform predictive simulation to generate 1000 datasets and record the percentage of observations that are equal to 0 and the percentage that are greater than 10 (the third quartile in the observed data) for each. Compare these to the observed value in the original data.

```
set.seed(100)
risky <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/RiskyBehavior/data/ris
risky$fupacts_R <- round(risky$fupacts, digits = 0)
fit1 <- stan_glm(fupacts_R ~ bs_hiv, family = poisson(link = "log"), data = risky, refresh = 0)
y_rep1 <- posterior_predict(fit1)
subset1 <- sample(y_rep1, 1000)
p_0 <- (sum(subset1==0))/1000
p_10 <- (sum(subset1>10))/1000
p_0
```

```
## [1] 0
```

```
p_10
```

```
## [1] 0.843
```

(b)

Repeat (a) using a negative binomial (overdispersed Poisson) regression.

```
set.seed(100)
fit2 <- stan_glm.nb(fupacts_R ~ bs_hiv, data=risky, link="log", refresh = 0)
y_rep2 <- posterior_predict(fit2)
subset2 <- sample(y_rep2, 1000)
p_0 <- (sum(subset2==0))/1000
p_10 <- (sum(subset2>10))/1000
p_0
```

```
## [1] 0.274
```

```
p_10
```

```
## [1] 0.344
```

###(c) Repeat (b), also including ethnicity and baseline number of unprotected sex acts as inputs. #### There is no ethnicity variable in this dataset?

```
set.seed(100)
fit3 <- stan_glm.nb(fupacts_R ~ bs_hiv + bupacts + sex, data=risky, link="log", refresh = 0)
y_rep3 <- posterior_predict(fit3)
subset3 <- sample(y_rep3, 1000)
p_0 <- (sum(subset3==0))/1000
p_10 <- (sum(subset3>10))/1000
p_0
```

```
## [1] 0.249
```

```
p_10
```

```
## [1] 0.316
```

15.15 Summarizing inferences and predictions using simulation:

Exercise 15.7 used a Tobit model to fit a regression with an outcome that had mixed discrete and continuous data. In this exercise you will revisit these data and build a two-step model: (1) logistic regression for zero earnings versus positive earnings, and (2) linear regression for level of earnings given earnings are positive.

Compare predictions that result from each of these models with each other.

```
# (1)
lalonde$earn <- ifelse(lalonde$re78==0,0,1)
fit1 <- stan_glm((re78 > 0) ~ treat + age + educ + black + hisp + married + nodegree + sample + educ_ca
fit1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      (re78 > 0) ~ treat + age + educ + black + hisp + married + nodegree +
##     sample + educ_cat4
##  observations: 18667
##  predictors:   10
## ------
##             Median MAD_SD
## (Intercept)  2.4    0.2
## treat       -0.4    0.2
## age          0.0    0.0
## educ         0.0    0.0
## black       -0.2    0.1
## hisp        -0.1    0.1
## married      0.4    0.1
## nodegree     0.1    0.1
## sample       0.2    0.1
## educ_cat4    0.1    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# (2)
fit2 <- stan_glm(re78 ~ treat + age + educ + black + hisp + married + nodegree + sample + educ_cat4, da
fit2
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      re78 ~ treat + age + educ + black + hisp + married + nodegree +
##     sample + educ_cat4
##  observations: 16164
##  predictors:   10
## ------
##             Median   MAD_SD
## (Intercept) -11061.5    696.7
## treat         5067.0    782.5
## age            168.9      7.0
## educ           538.2     53.8
## black        -3067.4    224.8
## hisp          -448.3    273.8
```

```
## married       4146.3    164.1
## nodegree     -1544.6    225.5
## sample        6684.8    195.4
## educ_cat4      279.7    153.6
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 8393.7   47.3
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

The first model predicts the likelihood that a person will have positive earnings, based on the level of the various predictors. The second model predicts what those earnings will be, given that the earnings are positive. For both models, the positive coefficients indicate a predicted increase in likelihood/earnings as the level of the predictor increases, and the negative coefficients indicate a predicted decrease in likelihood/earnings as the level of the predictor increases.