Homework 5

Yuxi Wang

10/13/2020

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("/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/RiskyBehavior/
data/risky.csv",header=T)
risk$fupacts_R = round(risk$fupacts)
head(risk)
```

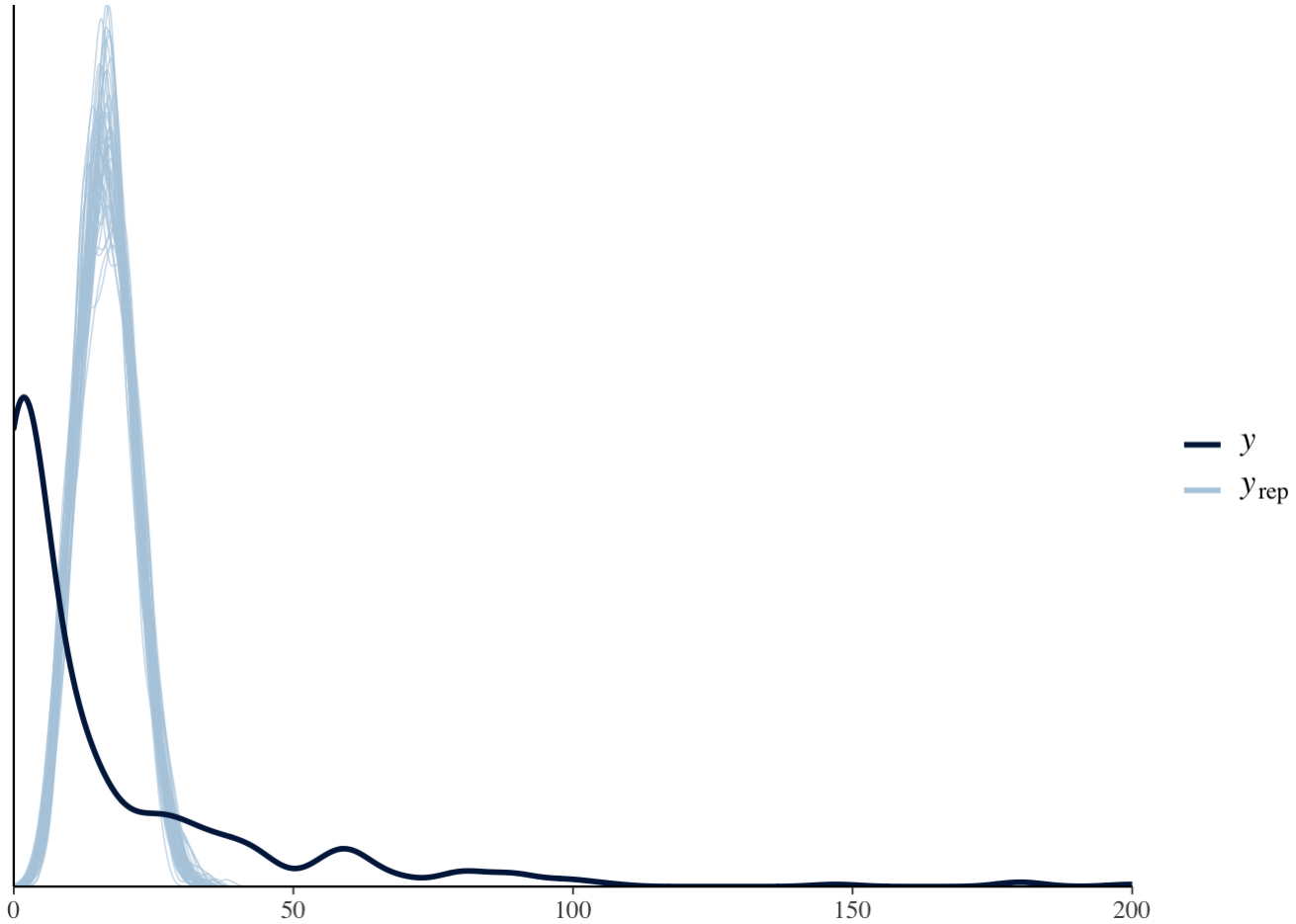
	sex <chr>	couples <int>	women_alone <int>	bs_hiv <chr>	bupacts <int>	fupacts <dbl>	fupacts_R <dbl>
1	woman	0	1	negative	7	32	32
2	woman	0	0	negative	2	5	5
3	woman	0	0	positive	0	15	15
4	woman	0	0	negative	24	9	9
5	woman	1	0	negative	2	2	2
6	woman	1	0	negative	15	4	4

6 rows

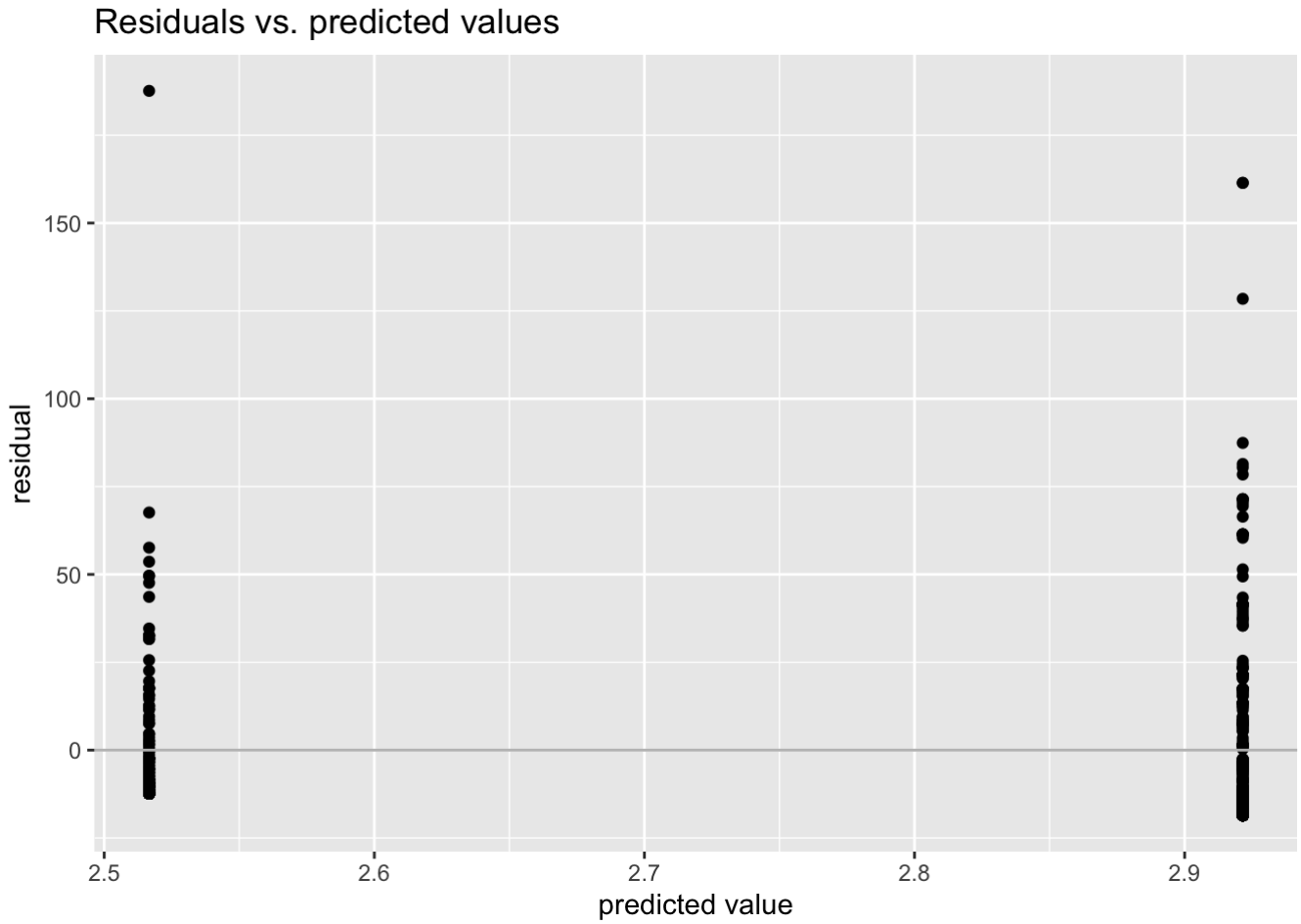
```
# Then we fit the Poisson model to the data:
fit_1 <- stan_glm(risk$fupacts_R ~ women_alone, family=poisson(link="log"), data=ris
k,refresh=0 )
print(fit_1)
```

```
## stan_glm
##  family:       poisson [log]
##  formula:      risk$fupacts_R ~ women_alone
##  observations: 434
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  2.9    0.0
## women_alone -0.4    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Then we check the fitting:
pp_check(fit_1)
```




```
y <- risk$fupacts_R
x <- risk$women_alone
ggplot()+
  geom_point(aes(x=predict(fit_1), y=resid(fit_1)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```




```
dispersiontest(fit_1, trafo=1) # Overdispersion corresponds to alpha > 0 and underdis
persion to alpha < 0.
```

```
##
##  Overdispersion test
##
## data:  fit_1
## z = 4.9301, p-value = 4.109e-07
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##    alpha
## 41.97773
```

By observing the residual graph and use the Test for Overdispersion by Cameron & Trivedi, we can find that the degree of dispersion of the model is very high, and the result of the Overdispersion test is 41.99716>>0.

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

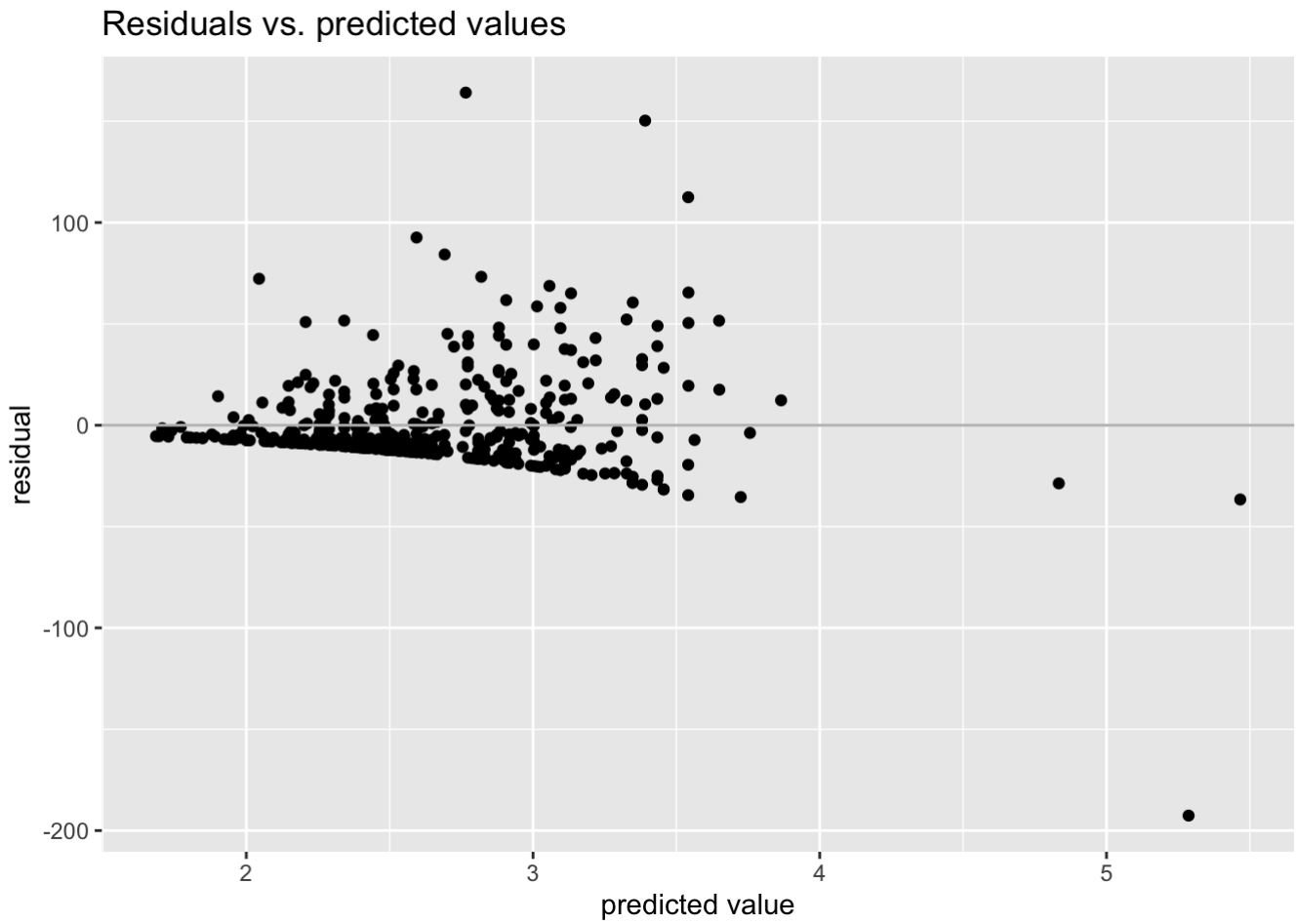
b)

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?

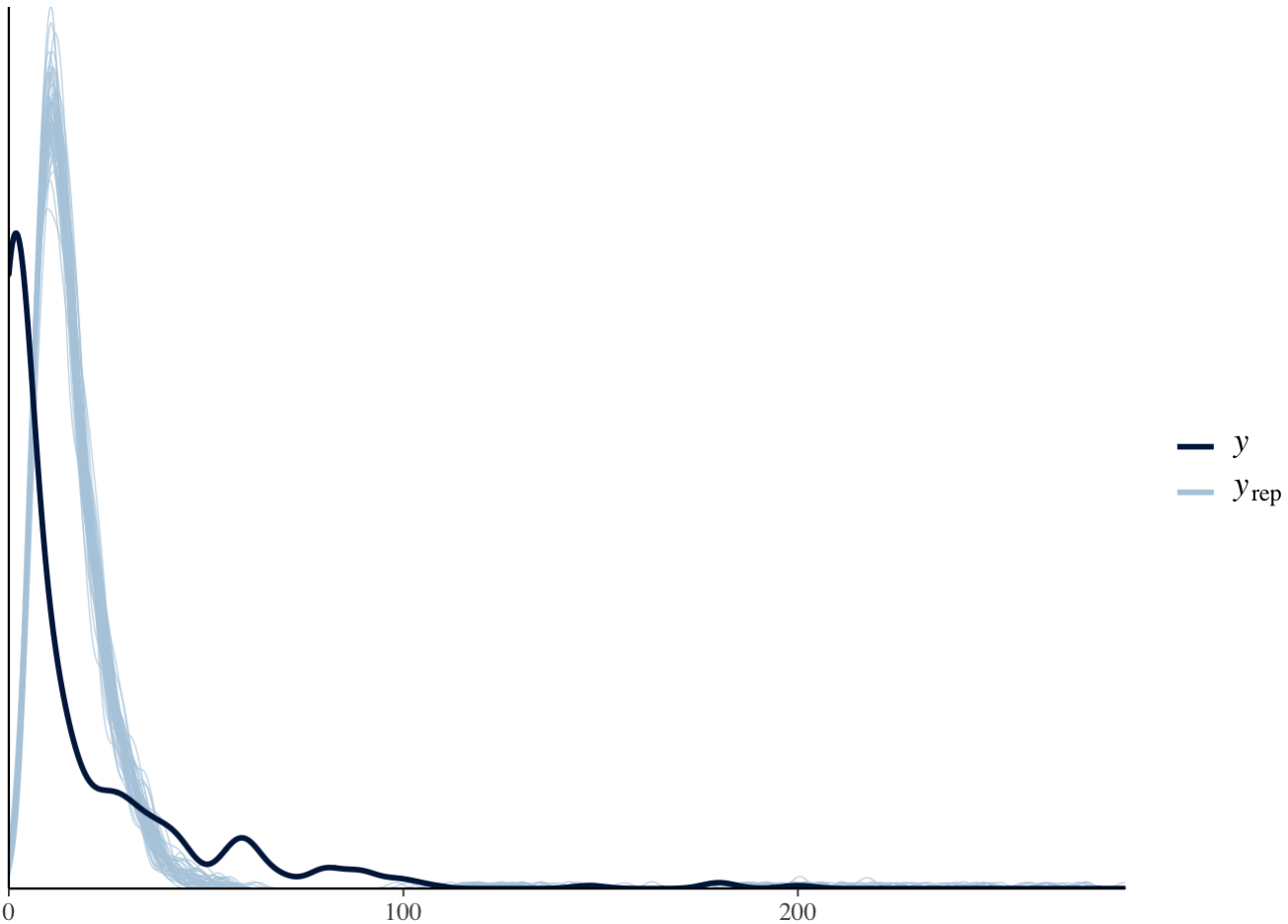
```
# Since only bupacts is a continuous variable among the variables, it will cause cert
ain problems when compared with other binary variables, so the standardization proces
s is carried out first.
risk$bupacts_new <- (risk$bupacts - mean(risk$bupacts))/(2*sd(risk$bupacts))
# Fitting with additional pre-treatment variables
fit1_b<- stan_glm(risk$fupacts_R ~risk$women_alone + risk$sex + risk$bupacts_new + ri
sk$couples + risk$bs_hiv, family=poisson(link="log"), data=risk, refresh=0)
print(fit1_b)
```

```
## stan_glm
##  family:       poisson [log]
##  formula:      risk$fupacts_R ~ risk$women_alone + risk$sex + risk$bupacts_new +
##     risk$couples + risk$bs_hiv
##  observations: 434
##  predictors:   6
## ------
##                     Median MAD_SD
## (Intercept)          3.1    0.0
## risk$women_alone    -0.7    0.0
## risk$sexwoman        0.1    0.0
## risk$bupacts_new     0.7    0.0
## risk$couples        -0.4    0.0
## risk$bs_hivpositive -0.4    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Then we make the residual plot and check the fitting
ggplot()+
  geom_point(aes(x=predict(fit1_b), y=resid(fit1_b)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```




```
pp_check(fit1_b)
```



```
dispersiontest(fit1_b, trafo=1) # Overdispersion corresponds to alpha > 0 and underdi
spersion to alpha < 0.
```

```
##
##  Overdispersion test
##
## data:  fit1_b
## z = 5.5692, p-value = 1.28e-08
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##    alpha
## 28.66243
```

By observing the residual graph and use the Test for Overdispersion by Cameron & Trivedi, we can find that the degree of fitting has improved. At this time, the result of the Overdispersion test is 28.65245>> 0, and there seems to be overdispersion.

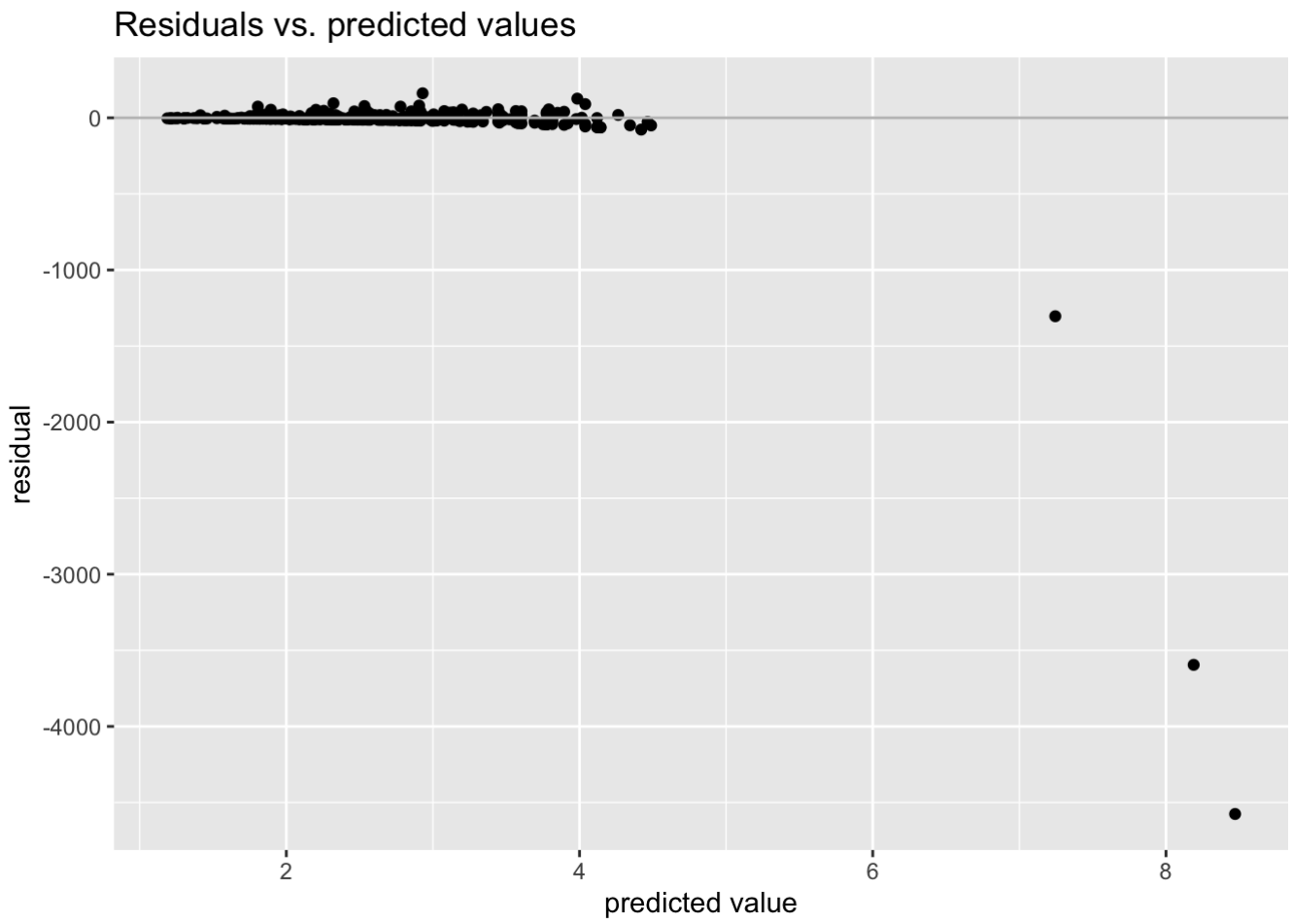
c)

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

```
# Fitting a negative binomial (overdispersed Poisson) model.
fit1_c<- stan_glm(risk$fupacts_R ~risk$women_alone + risk$sex + risk$bupacts_new + ri
sk$couples + risk$bs_hiv, family=neg_binomial_2(link="log"), data=risk, refresh=0)
print(fit1_c)
```

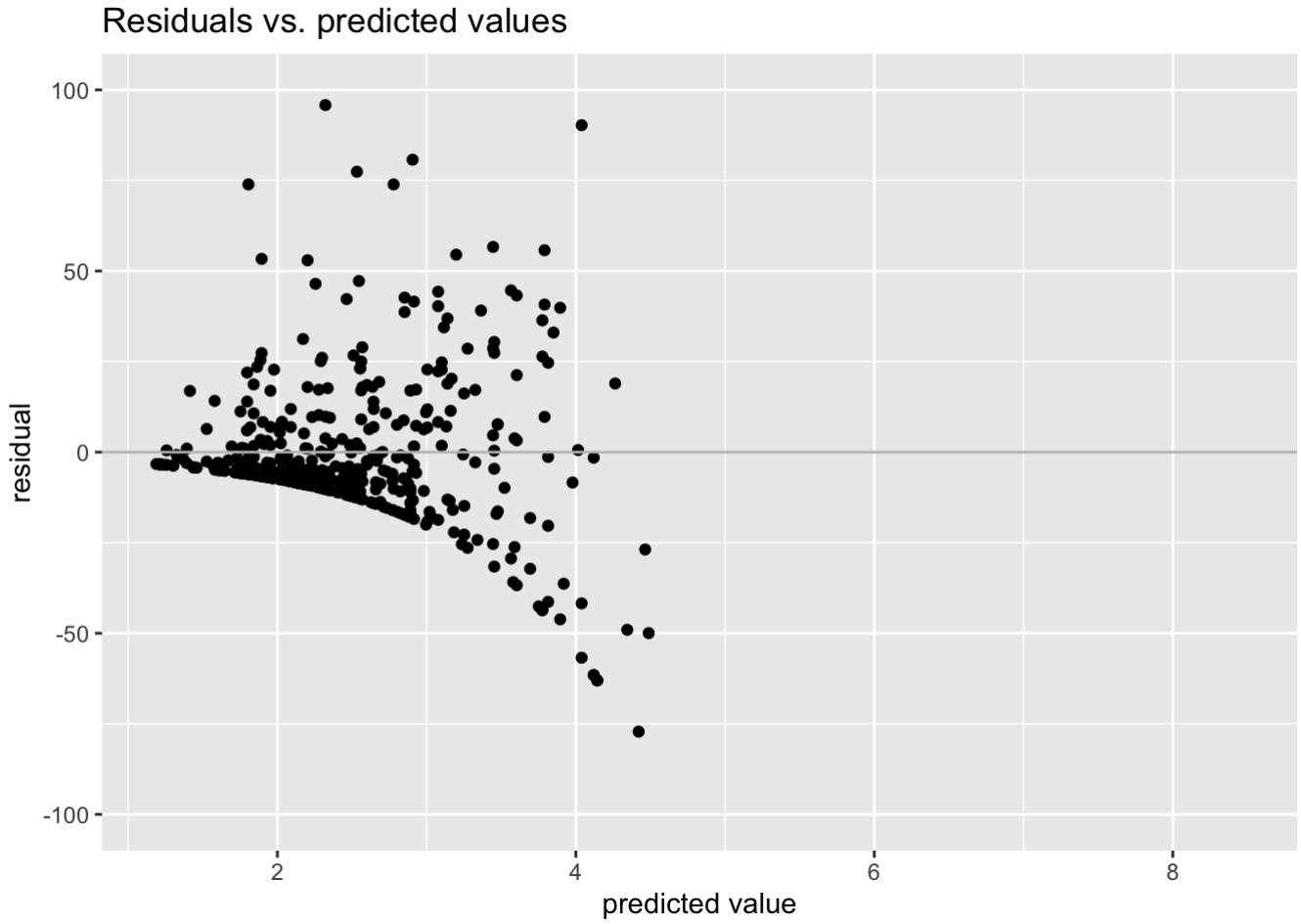
```
## stan_glm
##  family:       neg_binomial_2 [log]
##  formula:      risk$fupacts_R ~ risk$women_alone + risk$sex + risk$bupacts_new +
##     risk$couples + risk$bs_hiv
##  observations: 434
##  predictors:   6
## ------
##                      Median MAD_SD
## (Intercept)           3.0    0.2
## risk$women_alone     -0.7    0.2
## risk$sexwoman         0.0    0.2
## risk$bupacts_new      1.4    0.2
## risk$couples         -0.3    0.2
## risk$bs_hivpositive -0.5    0.2
##
## Auxiliary parameter(s):
##                        Median MAD_SD
## reciprocal_dispersion 0.4    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
ggplot()+
  geom_point(aes(x=predict(fit1_c), y=resid(fit1_c)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```



In this plot, we can see that most of the point are near 0, and only a few of point are scatter in -1000,-3500 and others. So, I make a second residuals plot with residual in (-100,100).

```
ggplot()+
  geom_point(aes(x=predict(fit1_c), y=resid(fit1_c)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")+
  ylim(-100, 100)
```



#In this residual plot, we can find point are evenly dispersed. So the model is better fitted. # Moreover, we can conclude that the intervention had a positive impact on decreasing the number of unprotected sex acts. Therefore, we can find out how couples whose only women participated in the counseling saw a reduction in unprotected sexual behavior e^(-0.66) = 0.51685. Interestingly, when both partners attended the consultation meeting, the reduction was very small, only 33.63%.

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?

Yes, I think this is one of the problems, because the couple's observations of these two elements will not be i.i.d. A wife will effet her partner, so we may think this have a very high positive correlation.

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(1213)
N <- 100
height <- rnorm(N, 72, 3)
p <- 0.4 + 0.1*(height - 72)/3

n <- rep(10, N)
for (i in 1:N) {
    a <- runif(1,min=10,max=30)
    n[i] <- round(a) # since shooting time can only be Interger.
}
y <- rbinom(N, n, p)
data <- data.frame(n=n, y=y, height=height)
fit3_a <- stan_glm(cbind(y, n-y) ~ height, family=binomial(link="logit"),data=data,re
fresh=0)
print(fit3_a)
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      cbind(y, n - y) ~ height
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -9.7    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
```

(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.

```
# Firstly, to calculate the logistic function.
a1 <- logit(0.3)
a2 <- logit(0.4)
b <- (a2-a1)/3;b
```

```
## [1] 0.1472776
```

```
a <- a2-72*b;a
```

```
## [1] -11.00945
```

```
# So, we got the logistic function: Pr(y[i]=1)=invlogit(-11.00945+0.1472776*height)
# Then, do the regression
N <- 100
height <- rnorm(N, 72, 3)
p <- invlogit(-11.00945+0.1472776*height)
n <- rep(20, N)
y <- rbinom(N, n, p)
data <- data.frame(n=n, y=y, height=height)
fit3_b <- stan_glm(cbind(y, n-y) ~ height, family=binomial(link="logit"),data=data,re
fresh=0)
print(fit3_b)
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      cbind(y, n - y) ~ height
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -10.2    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("/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/L
alonde/NSW_dw_obs.dta")
head(lalonde)
```

...	educ	black	married	nodegree	re74	re75	re78	hisp
	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<int>

	...	educ	black	married	nodegree	re74	re75	re78	hisp
	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<int>
1	42	16	0	1	0	0.000	0.000	100.4854	0
2	20	13	0	0	0	2366.794	3317.468	4793.7451	0
3	37	12	0	1	0	25862.322	22781.855	25564.6699	0
4	48	12	0	1	0	21591.121	20839.355	20550.7441	0
5	51	12	0	1	0	21395.193	21575.178	22783.5879	0
6	18	11	0	0	1	1310.750	1455.532	2157.4807	0

6 rows | 1-10 of 13 columns

```
# creating factors
lalonde$sample <- factor(lalonde$sample, labels=c("NSW", "CPS", "PSID"))
lalonde$black <- factor(lalonde$black)
lalonde$hisp <- factor(lalonde$hisp)
lalonde$nodegree <- factor(lalonde$nodegree)
lalonde$married <- factor(lalonde$married)
lalonde$treat <- factor(lalonde$treat)
lalonde$educ_cat4 <- factor(lalonde$educ_cat4, labels=c("less than high school", "hig
h school", "sm college", "college"))

# To create a function to normalise and standardise numeric variables
standardise <- function(X) {
    cols <- ncol(X)
    for (c in 1:cols) {
        if (is.numeric(X[, c])) {
            start <- ncol(X)
            c.c <- (X[, c] - mean(X[, c], na.rm=TRUE)) / (2 * sd(X[, c], na.rm=TRUE))
            X[start+1] <- c.c
            colnames(X)[start+1] <- paste0("c.", colnames(X)[c])
        }
    }
    return(X)
}
lalonde_1 <- standardise(lalonde)
summary(lalonde_1)
```

```
##       age              educ       black     married   nodegree       re74
##  Min.   :16.00   Min.   : 0.00   0:16711   0: 5093   0:13045   Min.   :     0
##  1st Qu.:24.00   1st Qu.:11.00   1: 1956   1:13574   1: 5622   1st Qu.:  4898
##  Median :31.00   Median :12.00                                 Median : 15525
##  Mean   :33.37   Mean   :12.02                                 Mean   : 14621
##  3rd Qu.:42.00   3rd Qu.:14.00                                 3rd Qu.: 23882
##  Max.   :55.00   Max.   :18.00                                 Max.   :137149
##       re75             re78        hisp       sample      treat
##  Min.   :     0   Min.   :     0   0:17423   NSW :  185   0:18482
##  1st Qu.:  4726   1st Qu.:  6158   1: 1244   CPS :15992   1:  185
##  Median : 14899   Median : 16957             PSID: 2490
##  Mean   : 14253   Mean   : 15657
##  3rd Qu.: 23274   3rd Qu.: 25565
##  Max.   :156653   Max.   :121174
##                  educ_cat4        c.age             c.educ
##  less than high school:5622   Min.   :-0.7913   Min.   :-2.074555
##  high school           :7144   1st Qu.:-0.4269   1st Qu.:-0.176481
##  sm college            :3105   Median :-0.1079   Median :-0.003929
##  college               :2796   Mean   : 0.0000   Mean   : 0.000000
##                                3rd Qu.: 0.3933   3rd Qu.: 0.341176
##                                Max.   : 0.9856   Max.   : 1.031385
##      c.re74            c.re75             c.re78
##  Min.   :-0.7047   Min.   :-0.70089   Min.   :-0.71864
##  1st Qu.:-0.4686   1st Qu.:-0.46850   1st Qu.:-0.43598
##  Median : 0.0436   Median : 0.03179   Median : 0.05966
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.4464   3rd Qu.: 0.44364   3rd Qu.: 0.45474
##  Max.   : 5.9058   Max.   : 7.00266   Max.   : 4.84307
```

```
# In probit regression, all outcome values must be 0 or 1 for Bernoulli models.
# So we have to generate a outcome from c.re78.
lalonde_1$outcome <- rep(NA, nrow(lalonde_1))
lalonde_1$outcome <- ifelse(lalonde_1$re78>=25564.669921875, 1, 0)
lalonde_1$outcome <- factor(lalonde_1$outcome, labels=c("lt", "gte"))
# When lalonde_1$re78<25564.669921875, we will use:
fit7_1 <-vglm(lalonde_1$re78 ~ lalonde_1$c.age + lalonde_1$c.educ + lalonde_1$c.re75+
lalonde_1$black + lalonde_1$married, tobit(Lower=0, Upper=25563), data=lalonde_1, sub
set=re78<25564)
summary(fit7_1)
```

```
##
## Call:
## vglm(formula = lalonde_1$re78 ~ lalonde_1$c.age + lalonde_1$c.educ +
##     lalonde_1$c.re75 + lalonde_1$black + lalonde_1$married, family = tobit(Lower =
0,
##     Upper = 25563), data = lalonde_1, subset = re78 < 25564)
##
## Pearson residuals:
##                    Min      1Q  Median     3Q    Max
## mu           -134.0283 -0.7539  0.1359 0.6839  2.034
## loglink(sd)    -0.9993 -0.7229 -0.4178 0.2422 71.099
##
## Coefficients:
##                       Estimate Std. Error  z value Pr(>|z|)
## (Intercept):1       1.230e+04  1.560e+02   78.831  < 2e-16 ***
## (Intercept):2       9.027e+00  7.283e-03 1239.462  < 2e-16 ***
## lalonde_1$c.age    -3.364e+03  1.651e+02  -20.383  < 2e-16 ***
## lalonde_1$c.educ   -6.901e+02  1.524e+02   -4.527 5.99e-06 ***
## lalonde_1$c.re75    1.355e+04  1.971e+02   68.773  < 2e-16 ***
## lalonde_1$black1   -3.763e+02  2.266e+02   -1.660   0.0969 .
## lalonde_1$married1  1.624e+02  1.795e+02    0.905   0.3656
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: mu, loglink(sd)
##
## Log-likelihood: -118525.6 on 27219 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):2'
```

```
# When lalonde_1$re78>=25564.669921875, we will use:
fit7_2 <- glm(lalonde_1$outcome ~ lalonde_1$c.age + lalonde_1$c.educ + lalonde_1$c.re
75 + lalonde_1$black + lalonde_1$married, family=binomial(link="probit"), data=lalond
e_1)
display(fit7_2)
```

```
## glm(formula = lalonde_1$outcome ~ lalonde_1$c.age + lalonde_1$c.educ +
##     lalonde_1$c.re75 + lalonde_1$black + lalonde_1$married, family = binomial(link
= "probit"),
##     data = lalonde_1)
##                    coef.est coef.se
## (Intercept)        -1.03     0.03
## lalonde_1$c.age     0.01     0.03
## lalonde_1$c.educ    0.42     0.03
## lalonde_1$c.re75    1.99     0.04
## lalonde_1$black1   -0.19     0.04
## lalonde_1$married1  0.19     0.03
## ---
##   n = 18667, k = 6
##   residual deviance = 14739.9, null deviance = 21803.0 (difference = 7063.1)
```

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("/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/Congress/data/congress.csv")
congress88 <- data.frame(vote=congress$v88_adj,pastvote=congress$v86_adj,inc=congress$inc88)
head(congress88)
```

	vote <dbl>	pastvote <dbl>	inc <int>
1	0.7724427	0.7450362	1
2	0.6361816	0.6738455	1
3	0.6649283	0.6964566	1
4	0.2738342	0.4645901	-1
5	0.2636131	0.3910945	-1
6	0.3341927	0.3582454	-1

6 rows

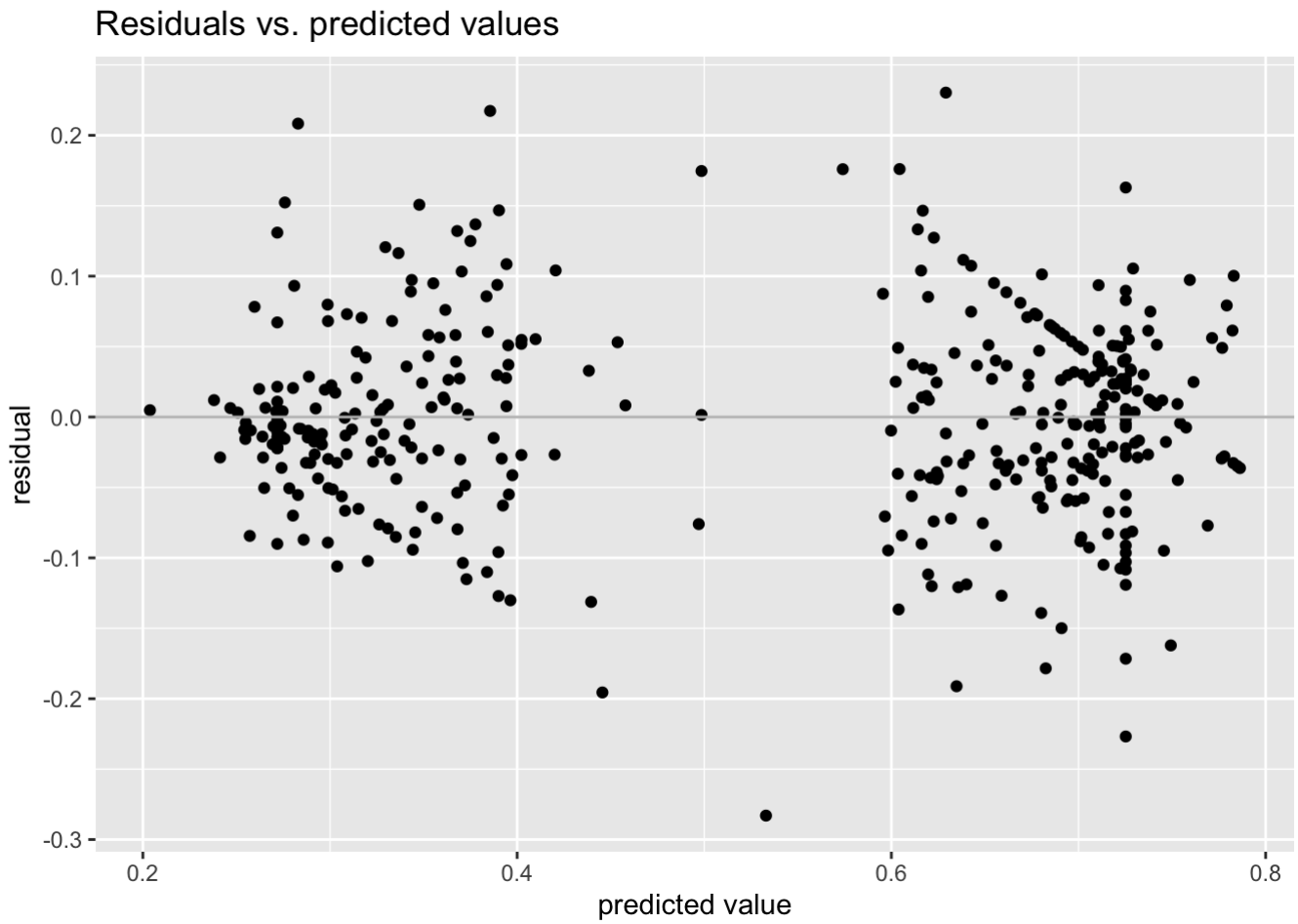
(a)

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.

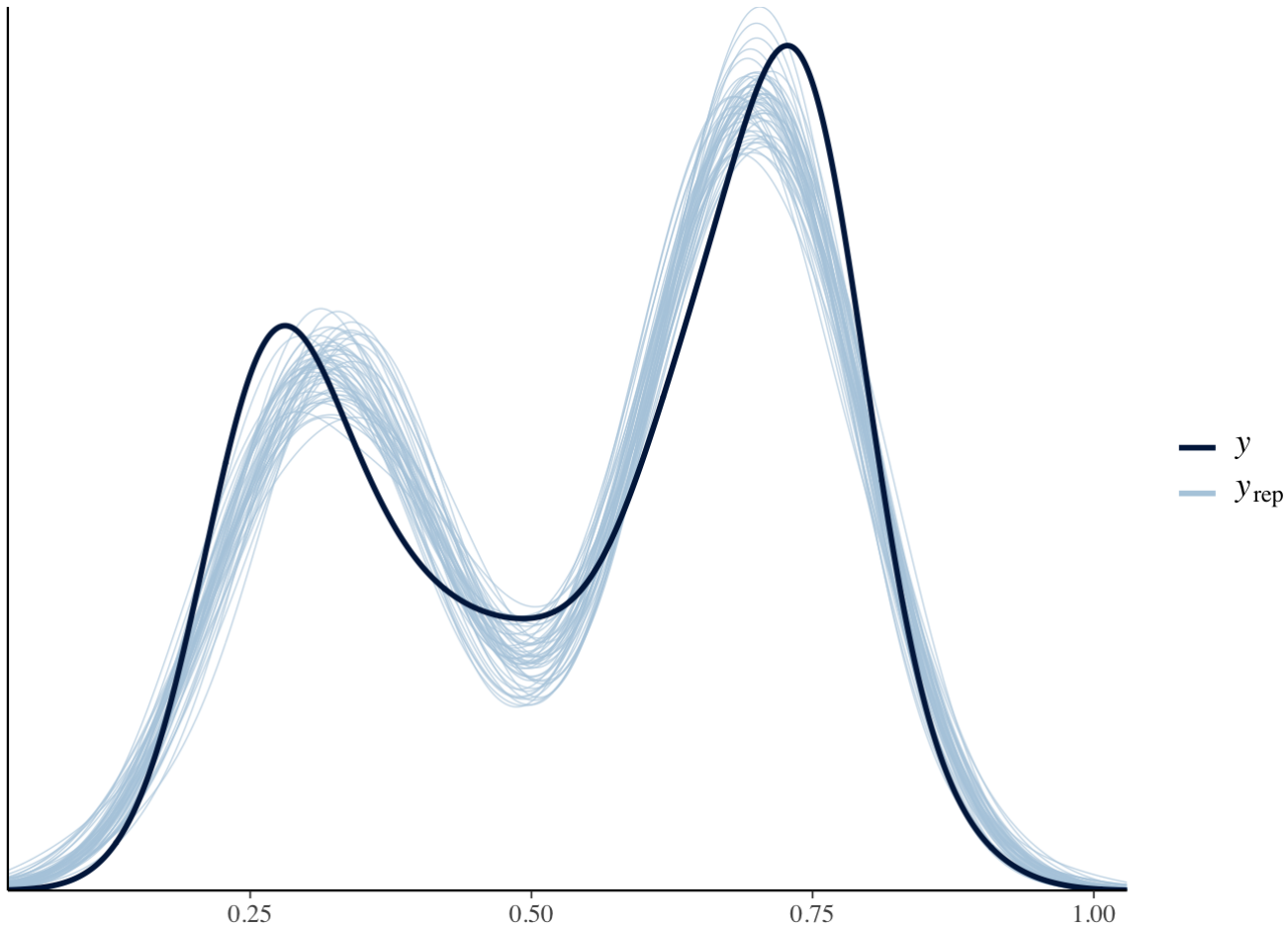
```
fit8_a <- stan_glm(congress88$vote ~ congress88$inc + congress88$pastvote, data=congress88, refresh=0)
summary(fit8_a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      congress88$vote ~ congress88$inc + congress88$pastvote
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 435
##  predictors:   3
##
## Estimates:
##                       mean   sd   10%   50%   90%
## (Intercept)          0.2    0.0  0.2   0.2   0.3
## congress88$inc       0.1    0.0  0.1   0.1   0.1
## congress88$pastvote 0.5    0.0  0.5   0.5   0.6
## sigma                0.1    0.0  0.1   0.1   0.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.5    0.0  0.5   0.5   0.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcom
e variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                      mcse Rhat n_eff
## (Intercept)          0.0  1.0  1591
## congress88$inc       0.0  1.0  1618
## congress88$pastvote 0.0  1.0  1558
## sigma                0.0  1.0  2181
## mean_PPD             0.0  1.0  3864
## log-posterior        0.0  1.0  1706
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure o
f effective sample size, and Rhat is the potential scale reduction factor on split ch
ains (at convergence Rhat=1).
```

```
# Also, to make residual plot and do pp_check
ggplot()+
  geom_point(aes(x=predict(fit8_a), y=resid(fit8_a)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```




```
pp_check(fit8_a)
```



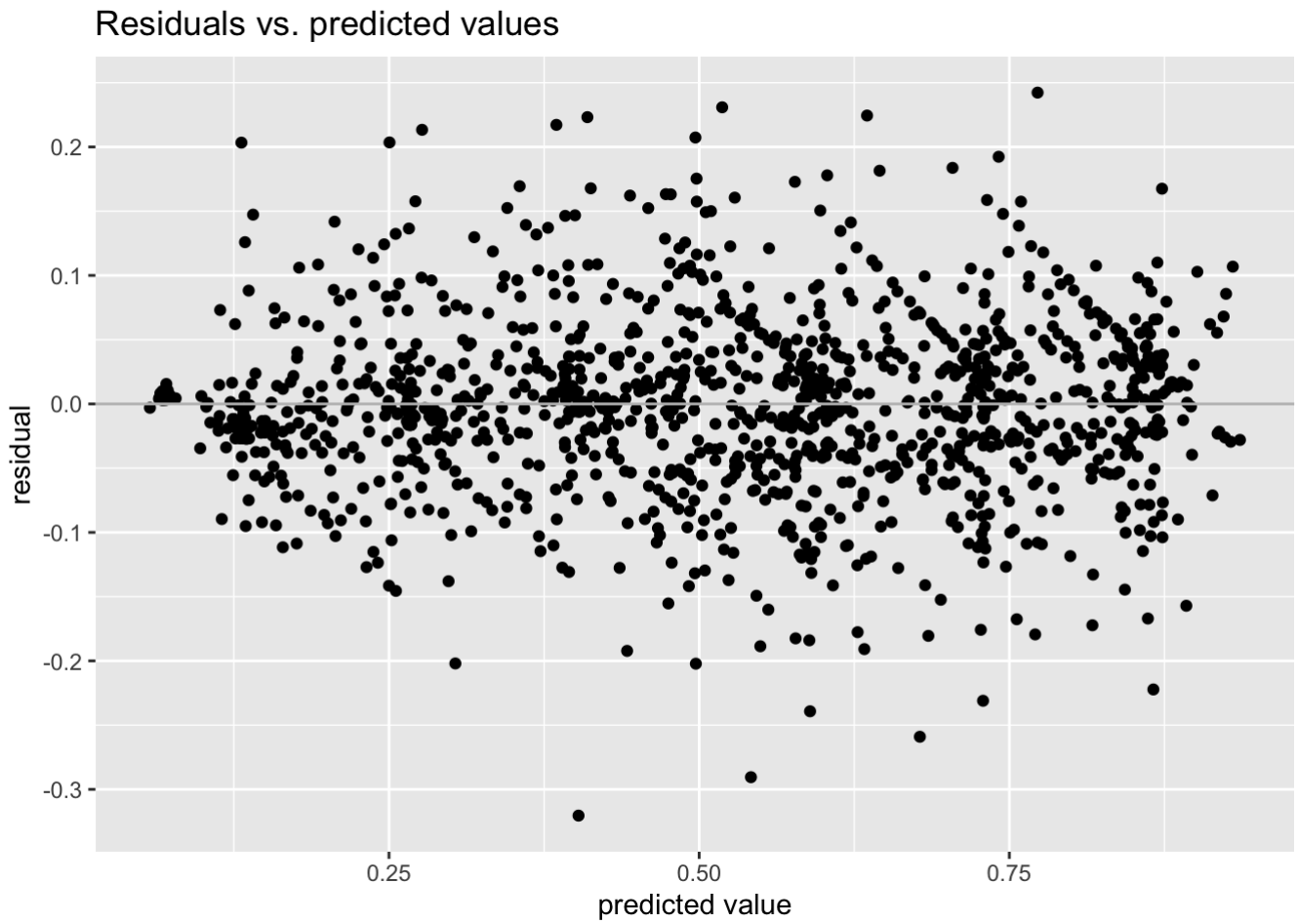
(b)

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.

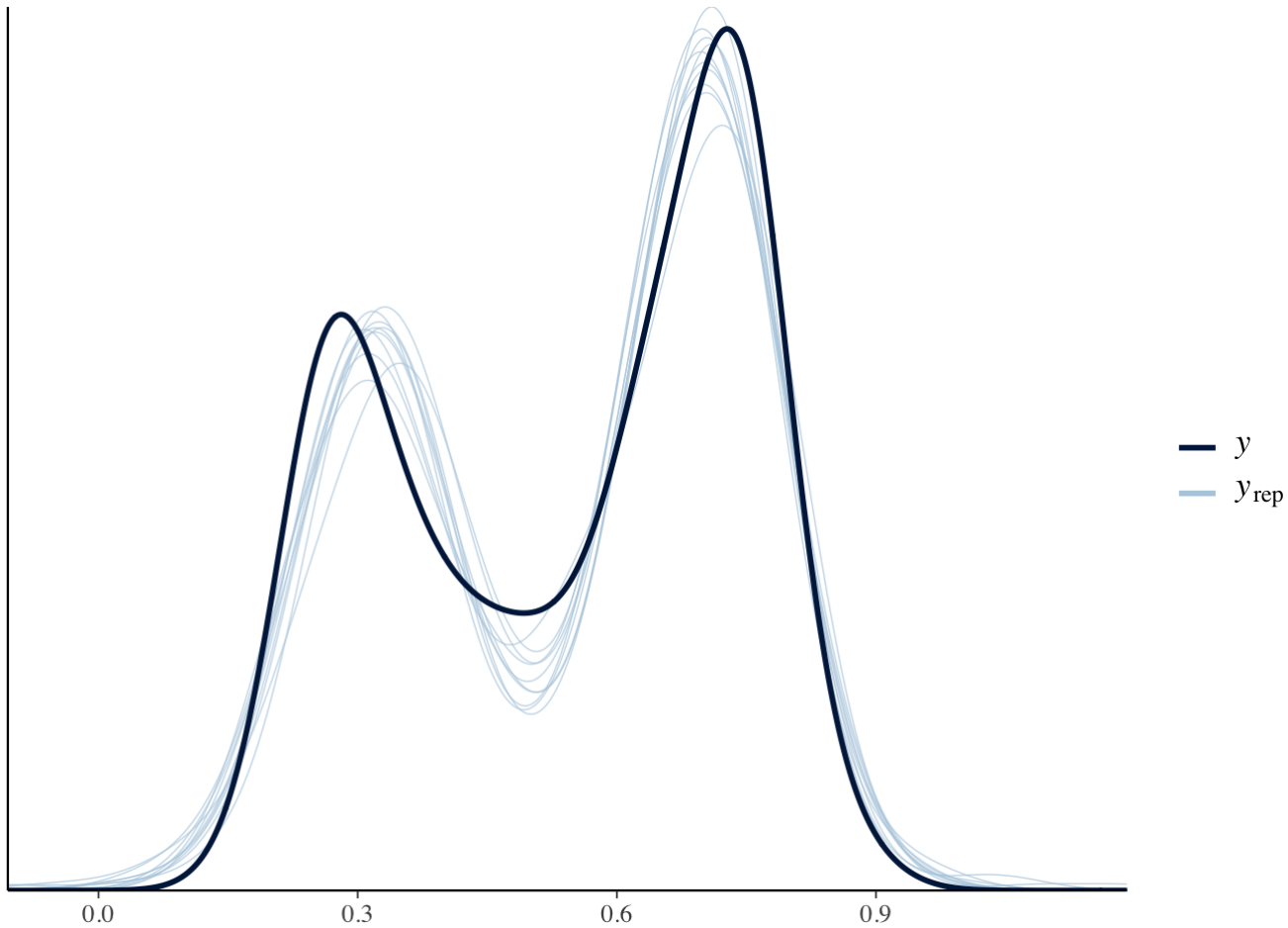
```
fit8_b <- brm(vote ~ inc + pastvote, family=student(), data = congress88, refresh=0)
summary(fit8_b)
```

```
##  Family: student
##   Links: mu = identity; sigma = identity; nu = identity
## Formula: vote ~ inc + pastvote
##    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     1833     1978
## inc           0.09      0.01     0.08     0.11 1.00     1811     1968
## pastvote      0.55      0.03     0.48     0.62 1.00     1763     2025
##
## 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     1807     2092
## nu        6.24      2.53     3.42    12.39 1.00     1792     1979
##
## 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).
```

```
# Also, to make residual plot and do pp_check
ggplot()+
  geom_point(aes(x=predict(fit8_b), y=resid(fit8_b)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```




```
pp_check(fit8_b)
```



(c)

Which model do you prefer?

I prefer the second model, which is using the brm function with the student family. Because the residual plot of each model has been drewed. We can easiy see that the second residual plot is more mess and more evenly. Also, we can easily say the second model is better by watch their pp_check() plot.

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.

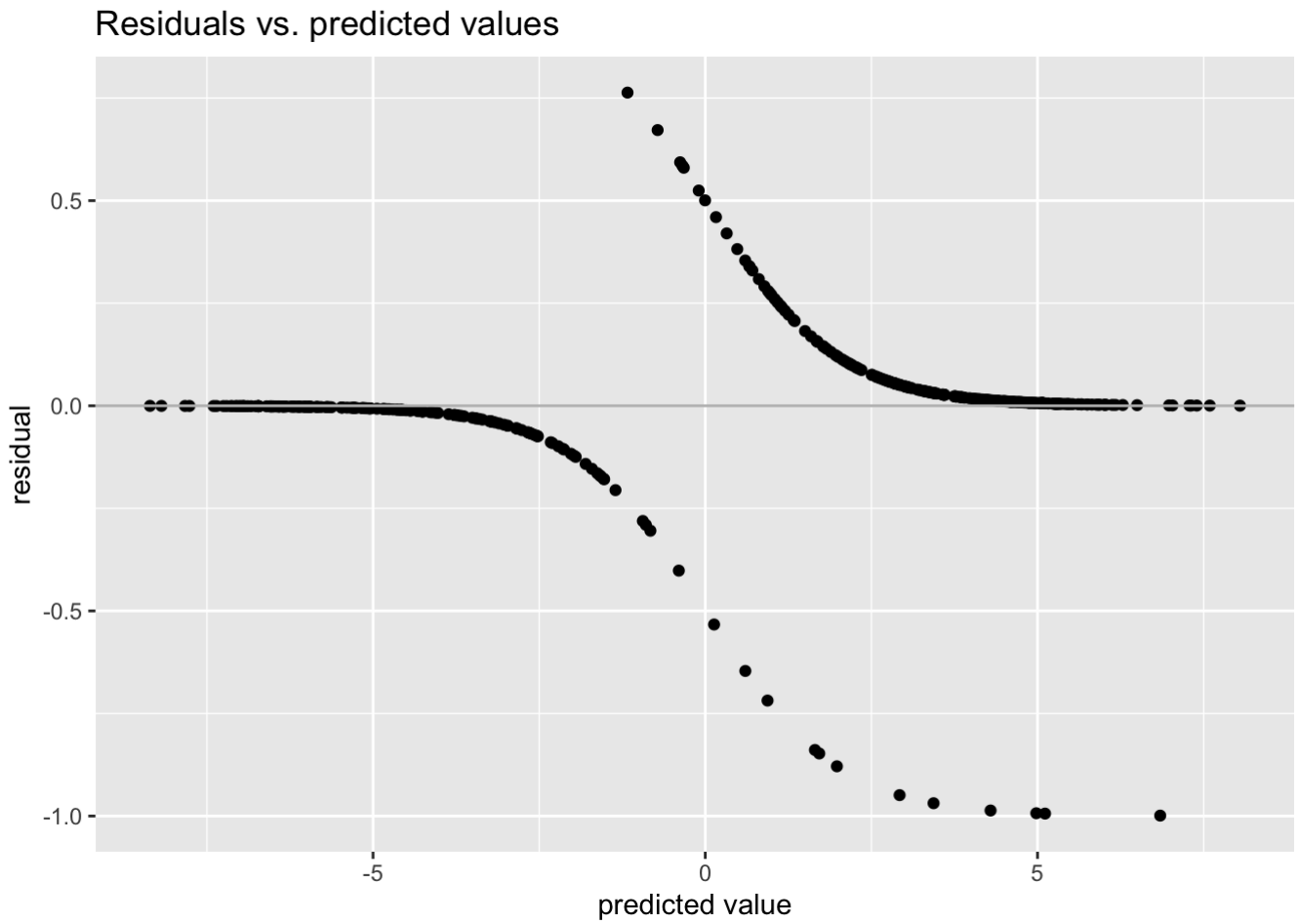
(a)

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

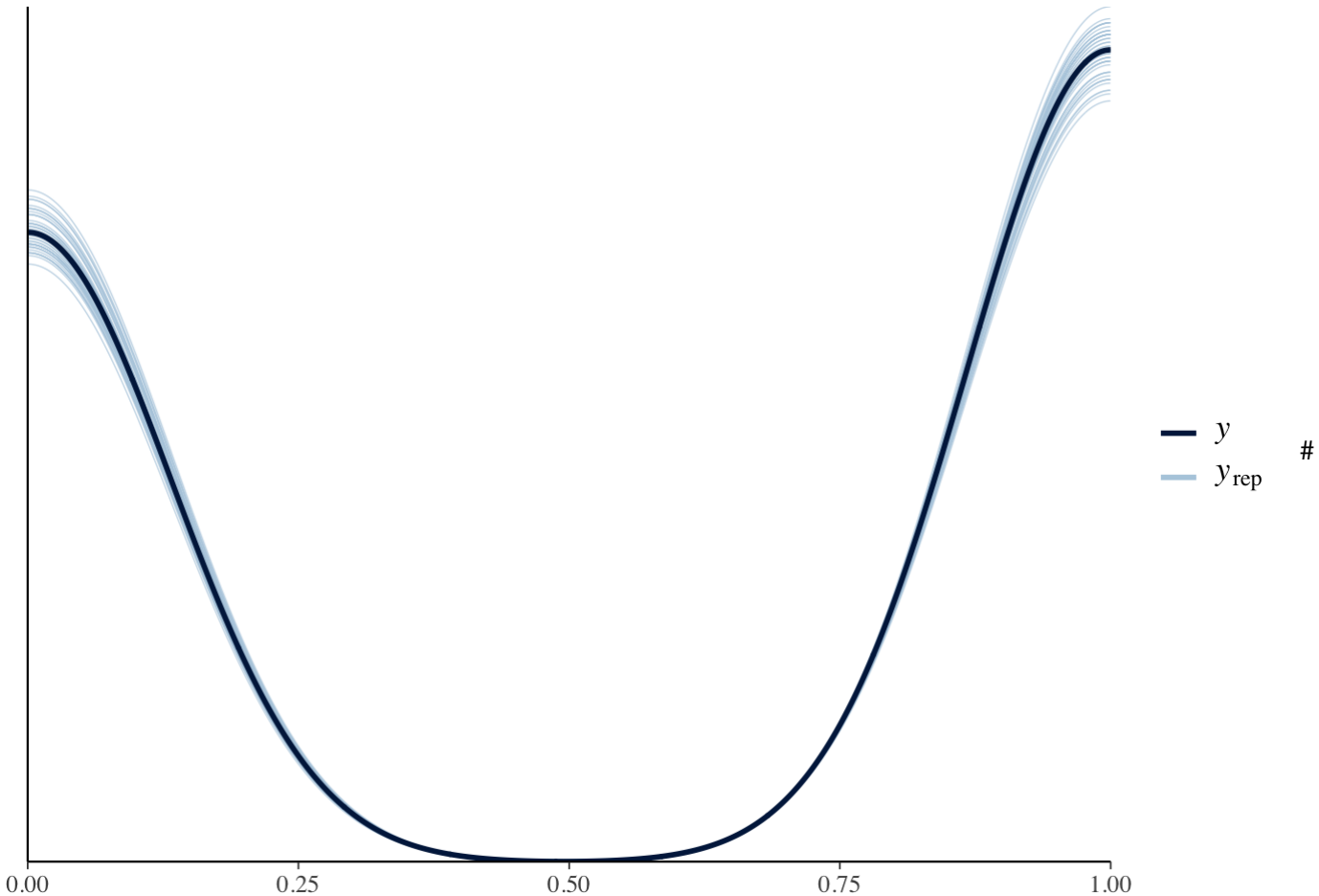
```
congress = read.csv("/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/Congress/da
ta/congress.csv")
congress88 <- data.frame(vote=congress$v88_adj,pastvote=congress$v86_adj,inc=congress
$inc88)
congress88[congress88==-1] <- 0
fit9_a <- stan_glm(congress88$inc~ congress88$vote + congress88$pastvote,family = bin
omial(link = "logit"), data=congress88, refresh=0)
summary(fit9_a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      congress88$inc ~ congress88$vote + congress88$pastvote
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 435
##  predictors:   3
##
## Estimates:
##                       mean   sd    10%   50%   90%
## (Intercept)         -12.7    1.4 -14.6 -12.6 -10.9
## congress88$vote      16.4    2.9  12.8  16.2  20.1
## congress88$pastvote   7.4    2.4   4.4   7.4  10.6
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.6    0.0  0.5   0.6   0.6
##
## The mean_ppd is the sample average posterior predictive distribution of the outcom
e variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                     mcse Rhat n_eff
## (Intercept)         0.0  1.0  3369
## congress88$vote     0.1  1.0  2373
## congress88$pastvote 0.0  1.0  2459
## mean_PPD            0.0  1.0  2789
## log-posterior       0.0  1.0  1754
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure o
f effective sample size, and Rhat is the potential scale reduction factor on split ch
ains (at convergence Rhat=1).
```

```
# Also, to make residual plot and do pp_check
ggplot()+
  geom_point(aes(x=predict(fit9_a), y=resid(fit9_a)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```




```
pp_check(fit9_a)
```



In the residual plot and the pp_check plot we can say the model is okay.

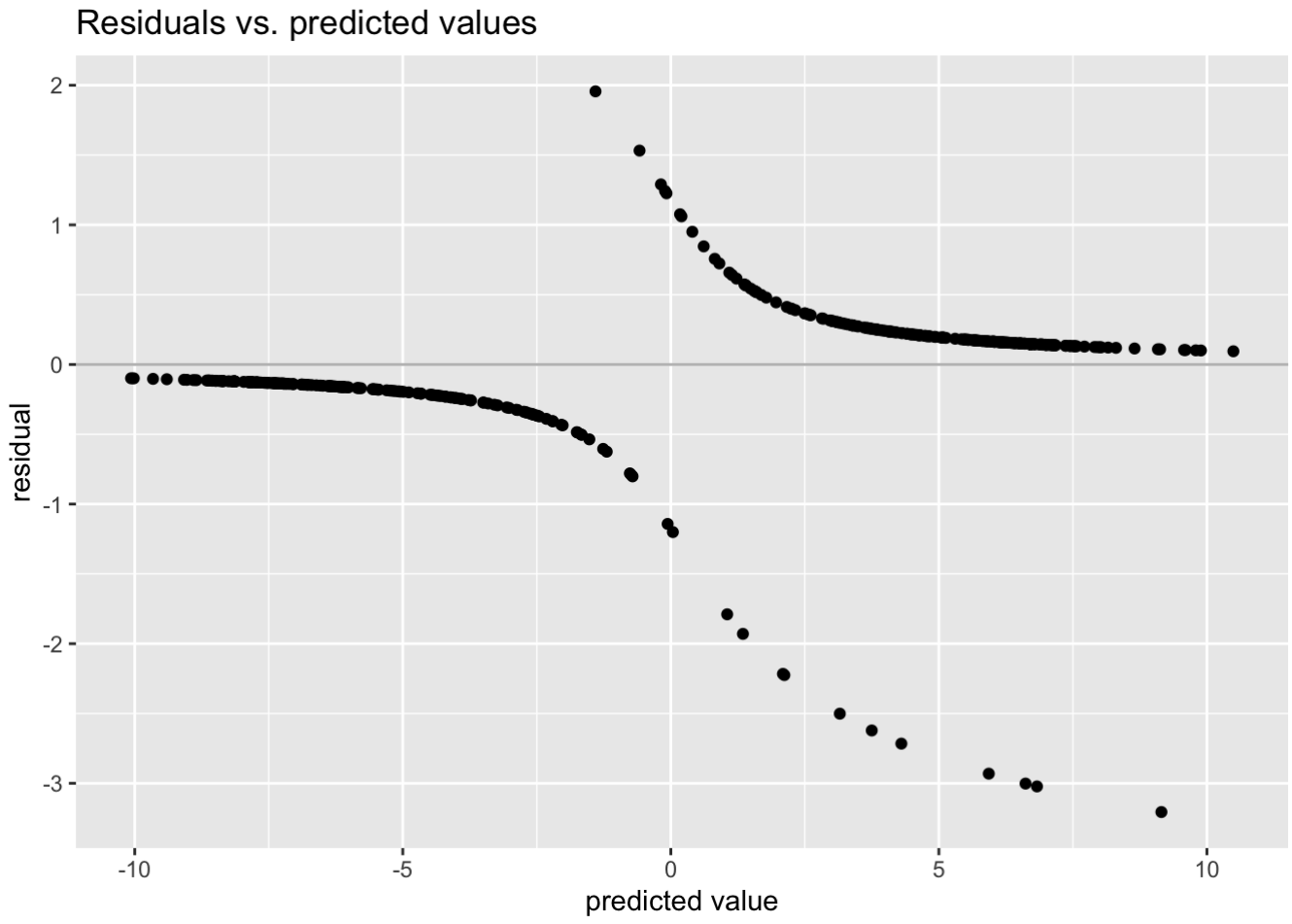
(b)

Fit a robit regression and assess model fit.

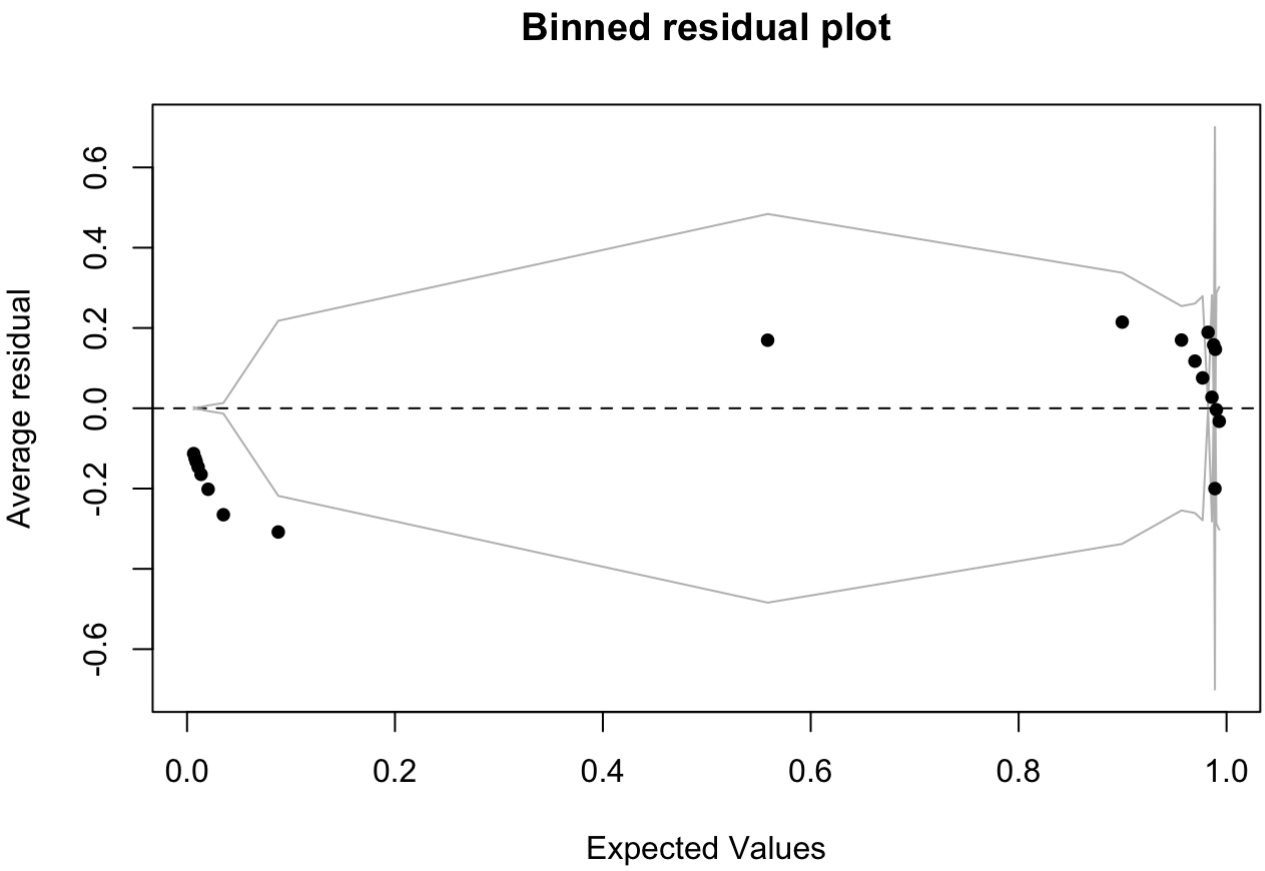
```
# Fit a robit regression
fit9_b <- glm(congress88$inc~ congress88$vote + congress88$pastvote, family = binomia
l(link = gosset(2)), data=congress88)
display(fit9_b)
```

```
## glm(formula = congress88$inc ~ congress88$vote + congress88$pastvote,
##     family = binomial(link = gosset(2)), data = congress88)
##                     coef.est coef.se
## (Intercept)         -15.56     2.76
## congress88$vote      22.32     5.24
## congress88$pastvote   7.36     3.21
## ---
##   n = 435, k = 3
##   residual deviance = 116.7, null deviance = 596.1 (difference = 479.4)
```

```
# Using the residual plot to assess model fit
ggplot()+
  geom_point(aes(x=predict(fit9_b), y=resid(fit9_b)))+
  labs(x="predicted value", y="residual", title = "Residuals vs.\ predicted values")+
  geom_abline(slope=0, intercept=0, color="gray")
```




```
binnedplot(fitted(fit9_b), resid(fit9_b))
```



#

I will talk about the residual plot later, but the Binned residual plot shows that most of the point are outside the range of the line. So, the model may not fitted well.

(c)

Which model do you prefer?

I think use the logistic regression is better in this data set. Because when we look at each residual plot of each method, for the first plot, it is better than the second. At the mean time the pp_check plot of the first plot shows that the model fit the data. Also, the Binned residual plot of the second model shows that most of the point are outside the range of the line. So, I have to say, the logistic regression is better in this data set.

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.

```
risk <- read.csv("/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/RiskyBehavior/
data/risky.csv",header=T)
risk$fupacts_R <-  round(risk$fupacts)
risk$bupacts_R <-  round(risk$bupacts)
head(risk)
```

sex	couples	women_alone	bs_hiv	bupacts	fupacts	fupacts_R	bupacts_R
<chr>	<int>	<int>	<chr>	<int>	<dbl>	<dbl>	<dbl>
1 woman	0	1	negative	7	32	32	7
2 woman	0	0	negative	2	5	5	2
3 woman	0	0	positive	0	15	15	0
4 woman	0	0	negative	24	9	9	24
5 woman	1	0	negative	2	2	2	2
6 woman	1	0	negative	15	4	4	15

6 rows

```
risk$bs_hiv <- ifelse(
  risk$bs_hiv=="negative",0,1
)
fit14_a <-stan_glm(risk$fupacts_R ~ risk$bs_hiv, family=poisson(link="log"), data=ris
k,refresh=0 )
summary(fit14_a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       poisson [log]
##  formula:      risk$fupacts_R ~ risk$bs_hiv
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  2.9    0.0  2.9   2.9   2.9
## risk$bs_hiv -0.6    0.0 -0.7  -0.6  -0.6
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 16.5    0.3 16.1  16.5  16.8
##
## The mean_ppd is the sample average posterior predictive distribution of the outcom
e variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2668
## risk$bs_hiv   0.0  1.0  2485
## mean_PPD      0.0  1.0  3221
## log-posterior 0.0  1.0  1550
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure o
f effective sample size, and Rhat is the potential scale reduction factor on split ch
ains (at convergence Rhat=1).
```

```
# Performing predictive simulation to generate 1000 datasets
set.seed(1213)
n <- 1000
pred <- posterior_predict(fit14_a,draws=n)
```

```
# To make a prediction
a <- length(pred[pred==0])*100/1000
b <- mean(pred>10)*100
print(paste("the percentage of observations that are equal to 0 is:", a,"%",sep=""))
```

```
## [1] "the percentage of observations that are equal to 0 is:0.6%"
```

```
print(paste("the percentage of observations that are greater than 10 is:", b,"%",sep=
""))
```

```
## [1] "the percentage of observations that are greater than 10 is:84.7513824884793%"
```

```
# Compare the result to the observed value in the original data.
a1 <- length(risk$fupacts_R[risk$fupacts_R==0])*100/1000
b1 <- mean(risk$fupacts_R>0)*100
print(paste("the percentage of observations that are equal to 0 in original data is:"
,a1, "%", sep = ""))
```

```
## [1] "the percentage of observations that are equal to 0 in original data is:12.7%"
```

```
print(paste("the percentage of observations that are greater than 10 in original data
is:",b1, "%", sep = ""))
```

```
## [1] "the percentage of observations that are greater than 10 in original data is:7
0.7373271889401%"
```

(b)

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

```
# Redo the process above
risk$fupacts_R <-  round(risk$fupacts)
risk$bupacts_R <-  round(risk$bupacts)
head(risk)
```

sex <chr>	couples <int>	women_alone <int>	bs_hiv <dbl>	bupacts <int>	fupacts <dbl>	fupacts_R <dbl>	bupacts_R <dbl>
1 woman	0	1	0	7	32	32	7
2 woman	0	0	0	2	5	5	2
3 woman	0	0	1	0	15	15	0
4 woman	0	0	0	24	9	9	24
5 woman	1	0	0	2	2	2	2
6 woman	1	0	0	15	4	4	15

6 rows

```
risk$bs_hiv <- ifelse(
  risk$bs_hiv=="negative",0,1
)
fit14_b <-stan_glm(risk$fupacts_R ~ risk$bs_hiv, family=neg_binomial_2, data=risk,ref
resh=0 )
summary(fit14_b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       neg_binomial_2 [log]
##  formula:      risk$fupacts_R ~ risk$bs_hiv
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   2
##
## Estimates:
##                         mean   sd   10%   50%   90%
## (Intercept)           2.8    0.1  2.7   2.8   2.9
## reciprocal_dispersion 0.3    0.0  0.3   0.3   0.4
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 16.5    1.9 14.1  16.3  19.1
##
## The mean_ppd is the sample average posterior predictive distribution of the outcom
e variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                       mcse Rhat n_eff
## (Intercept)           0.0  1.0  2780
## reciprocal_dispersion 0.0  1.0  2776
## mean_PPD              0.0  1.0  3189
## log-posterior         0.0  1.0  1358
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure o
f effective sample size, and Rhat is the potential scale reduction factor on split ch
ains (at convergence Rhat=1).
```

```
# Performing predictive simulation to generate 1000 datasets
set.seed(1213)
n <- 1000
pred <- posterior_predict(fit14_b,draws=n)
```

```
# To make a prediction
a <- mean(pred==0)*100
b <- mean(pred>10)*100
print(paste("the percentage of observations that are equal to 0 is:", a,"%",sep=""))
```

```
## [1] "the percentage of observations that are equal to 0 is:24.094470046083%"
```

```
print(paste("the percentage of observations that are greater than 10 is:", b,"%",sep=
""))
```

```
## [1] "the percentage of observations that are greater than 10 is:42.460599078341%"
```

```
# Compare the result to the observed value in the original data.
a1 <- length(risk$fupacts_R[risk$fupacts_R==0])*100/1000
b1 <- mean(risk$fupacts_R>0)*100
print(paste("the percentage of observations that are equal to 0 in original data is:"
,a1, "%", sep = ""))
```

```
## [1] "the percentage of observations that are equal to 0 in original data is:12.7%"
```

```
print(paste("the percentage of observations that are greater than 10 in original data
is:",b1, "%", sep = ""))
```

```
## [1] "the percentage of observations that are greater than 10 in original data is:7
0.7373271889401%"
```

###(c) Repeat (b), also including ethnicity and baseline number of unprotected sex acts as inputs.

```
# There is no variable called "ethnicity" in the data
# Redo the process again:
risk$fupacts_R <-  round(risk$fupacts)
risk$bupacts_R <-  round(risk$bupacts)
head(risk)
```

	sex <chr>	couples <int>	women_alone <int>	bs_hiv <dbl>	bupacts <int>	fupacts <dbl>	fupacts_R <dbl>	bupacts_R <dbl>
1	woman	0	1	1	7	32	32	7
2	woman	0	0	1	2	5	5	2
3	woman	0	0	1	0	15	15	0
4	woman	0	0	1	24	9	9	24
5	woman	1	0	1	2	2	2	2
6	woman	1	0	1	15	4	4	15

6 rows

```
risk$bs_hiv <- ifelse(
  risk$bs_hiv=="negative",0,1
)
fit14_c <-stan_glm(risk$fupacts_R ~ risk$bs_hiv+ risk$bupacts_R, family=neg_binomial_
2, data=risk,refresh=0 )
summary(fit14_c)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       neg_binomial_2 [log]
##  formula:      risk$fupacts_R ~ risk$bs_hiv + risk$bupacts_R
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   3
##
## Estimates:
##                          mean   sd   10%   50%   90%
## (Intercept)           2.0    0.1  1.9   2.0   2.2
## risk$bupacts_R        0.0    0.0  0.0   0.0   0.0
## reciprocal_dispersion 0.4    0.0  0.4   0.4   0.4
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD  58.5   94.6  17.6  33.5 110.8
##
## The mean_ppd is the sample average posterior predictive distribution of the outcom
e variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                       mcse Rhat n_eff
## (Intercept)           0.0  1.0  3968
## risk$bupacts_R        0.0  1.0  4326
## reciprocal_dispersion 0.0  1.0  3177
## mean_PPD              1.5  1.0  3891
## log-posterior         0.0  1.0  1624
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure o
f effective sample size, and Rhat is the potential scale reduction factor on split ch
ains (at convergence Rhat=1).
```

```
# Performing predictive simulation to generate 1000 datasets
set.seed(1213)
n <- 1000
pred <- posterior_predict(fit14_c,draws=n)
```

```
# To make a prediction
a <- mean(pred==0)*100
b <- mean(pred>10)*100
print(paste("the percentage of observations that are equal to 0 is:", a,"%",sep=""))
```

```
## [1] "the percentage of observations that are equal to 0 is:7.74493087557604%"
```

```
print(paste("the percentage of observations that are greater than 10 is:", b,"%",sep=
""))
```

```
## [1] "the percentage of observations that are greater than 10 is:79.0995391705069%"
```

```
# Compare the result to the observed value in the original data.
a1 <- length(risk$fupacts_R[risk$fupacts_R==0])*100/1000
b1 <- mean(risk$fupacts_R>0)*100
print(paste("the percentage of observations that are equal to 0 in original data is:"
,a1, "%", sep = ""))
```

```
## [1] "the percentage of observations that are equal to 0 in original data is:12.7%"
```

```
print(paste("the percentage of observations that are greater than 10 in original data
is:",b1, "%", sep = ""))
```

```
## [1] "the percentage of observations that are greater than 10 in original data is:7
0.7373271889401%"
```

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.

```
# Load the data
lalonde <-  foreign::read.dta("/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/L
alonde/NSW_dw_obs.dta")
head(lalonde)
```

	...	educ	black	married	nodegree	re74	re75	re78	hisp
	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<int>
1	42	16	0	1	0	0.000	0.000	100.4854	0
2	20	13	0	0	0	2366.794	3317.468	4793.7451	0
3	37	12	0	1	0	25862.322	22781.855	25564.6699	0
4	48	12	0	1	0	21591.121	20839.355	20550.7441	0
5	51	12	0	1	0	21395.193	21575.178	22783.5879	0
6	18	11	0	0	1	1310.750	1455.532	2157.4807	0

6 rows | 1-10 of 13 columns

```
# creating factors
lalonde$sample <- factor(lalonde$sample, labels=c("NSW", "CPS", "PSID"))
lalonde$black <- factor(lalonde$black)
lalonde$hisp <- factor(lalonde$hisp)
lalonde$nodegree <- factor(lalonde$nodegree)
lalonde$married <- factor(lalonde$married)
lalonde$treat <- factor(lalonde$treat)
lalonde$educ_cat4 <- factor(lalonde$educ_cat4, labels=c("less than high school", "hig
h school", "sm college", "college"))
# To create a function to normalise and standardise numeric variables
standardise <- function(X) {
    cols <- ncol(X)
    for (c in 1:cols) {
        if (is.numeric(X[, c])) {
            start <- ncol(X)
            c.c <- (X[, c] - mean(X[, c], na.rm=TRUE)) / (2 * sd(X[, c], na.rm=TRUE))
            X[start+1] <- c.c
            colnames(X)[start+1] <- paste0("c.", colnames(X)[c])
        }
    }
    return(X)
}
lalonde_1 <- standardise(lalonde)
summary(lalonde_1)
```

```
##       age             educ       black     married   nodegree       re74
##  Min.   :16.00   Min.   : 0.00   0:16711   0: 5093   0:13045   Min.   :     0
##  1st Qu.:24.00   1st Qu.:11.00   1: 1956   1:13574   1: 5622   1st Qu.:  4898
##  Median :31.00   Median :12.00                                 Median : 15525
##  Mean   :33.37   Mean   :12.02                                 Mean   : 14621
##  3rd Qu.:42.00   3rd Qu.:14.00                                 3rd Qu.: 23882
##  Max.   :55.00   Max.   :18.00                                 Max.   :137149
##       re75             re78        hisp       sample      treat
##  Min.   :     0   Min.   :     0   0:17423   NSW :  185   0:18482
##  1st Qu.:  4726   1st Qu.:  6158   1: 1244   CPS :15992   1:  185
##  Median : 14899   Median : 16957             PSID: 2490
##  Mean   : 14253   Mean   : 15657
##  3rd Qu.: 23274   3rd Qu.: 25565
##  Max.   :156653   Max.   :121174
##                  educ_cat4        c.age             c.educ
##  less than high school:5622   Min.   :-0.7913   Min.   :-2.074555
##  high school           :7144   1st Qu.:-0.4269   1st Qu.:-0.176481
##  sm college            :3105   Median :-0.1079   Median :-0.003929
##  college               :2796   Mean   : 0.0000   Mean   : 0.000000
##                                3rd Qu.: 0.3933   3rd Qu.: 0.341176
##                                Max.   : 0.9856   Max.   : 1.031385
##      c.re74            c.re75             c.re78
##  Min.   :-0.7047   Min.   :-0.70089   Min.   :-0.71864
##  1st Qu.:-0.4686   1st Qu.:-0.46850   1st Qu.:-0.43598
##  Median : 0.0436   Median : 0.03179   Median : 0.05966
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.4464   3rd Qu.: 0.44364   3rd Qu.: 0.45474
##  Max.   : 5.9058   Max.   : 7.00266   Max.   : 4.84307
```

```
# logistic regression for zero earnings versus positive earnings
lalonde_1$zero <- ifelse(lalonde_1$re78==0, 0, 1)
fit15_1 <- glm(zero~ age + educ + black +married, family=binomial(link="logit"),data=
lalonde_1)
display(fit15_1)
```

```
## glm(formula = zero ~ age + educ + black + married, family = binomial(link = "logi
t"),
##     data = lalonde_1)
##             coef.est coef.se
## (Intercept)  2.93     0.13
## age         -0.03     0.00
## educ        -0.02     0.01
## black1      -0.23     0.07
## married1     0.39     0.05
## ---
##   n = 18667, k = 5
##   residual deviance = 14497.7, null deviance = 14712.7 (difference = 214.9)
```

```
# linear regression for level of earnings given earnings are positive.
# firstly, find earnings are positive.
lalonde_2 <- filter(lalonde_1,re78!="0")
lalonde_2$level <-  ifelse(lalonde_2$re78>25563,1 ,0)
fit15_2 <- glm(level ~ age+ educ+ black+ married, data=lalonde_2)
display(fit15_2)
```

```
## glm(formula = level ~ age + educ + black + married, data = lalonde_2)
##             coef.est coef.se
## (Intercept) -0.51     0.02
## age          0.01     0.00
## educ         0.04     0.00
## black1      -0.08     0.01
## married1     0.17     0.01
## ---
##   n = 16164, k = 5
##   residual deviance = 3007.8, null deviance = 3473.8 (difference = 466.0)
##   overdispersion parameter = 0.2
##   residual sd is sqrt(overdispersion) = 0.43
```

```
# Compare predictions that result from each of these models with each other.
pred15_1 <- predict(fit15_1,lalonde)
pred15_2 <- predict(fit15_2,lalonde)
summary(fit15_1)
```

```
##
## Call:
## glm(formula = zero ~ age + educ + black + married, family = binomial(link = "logi
t"),
##     data = lalonde_1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3427   0.4531   0.4998   0.5682   0.8775
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.931917   0.125604  23.343  < 2e-16 ***
## age         -0.030334   0.002120 -14.305  < 2e-16 ***
## educ        -0.023347   0.007497  -3.114 0.001844 **
## black1      -0.227670   0.067981  -3.349 0.000811 ***
## married1     0.392033   0.052612   7.451 9.24e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 14713  on 18666  degrees of freedom
## Residual deviance: 14498  on 18662  degrees of freedom
## AIC: 14508
##
## Number of Fisher Scoring iterations: 4
```

```
summary(fit15_2)
```

```
##
## Call:
## glm(formula = level ~ age + educ + black + married, data = lalonde_2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7619  -0.3384  -0.1535   0.4904   1.1024
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.511547   0.019469 -26.275  < 2e-16 ***
## age          0.007575   0.000351  21.584  < 2e-16 ***
## educ         0.038341   0.001203  31.867  < 2e-16 ***
## black1      -0.079824   0.011346  -7.036 2.06e-12 ***
## married1     0.166690   0.008481  19.654  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1861387)
##
##     Null deviance: 3473.8  on 16163  degrees of freedom
## Residual deviance: 3007.8  on 16159  degrees of freedom
## AIC: 18703
##
## Number of Fisher Scoring iterations: 2
```