MA678 Homework 5

Reese Mullen

10/22/2024

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)

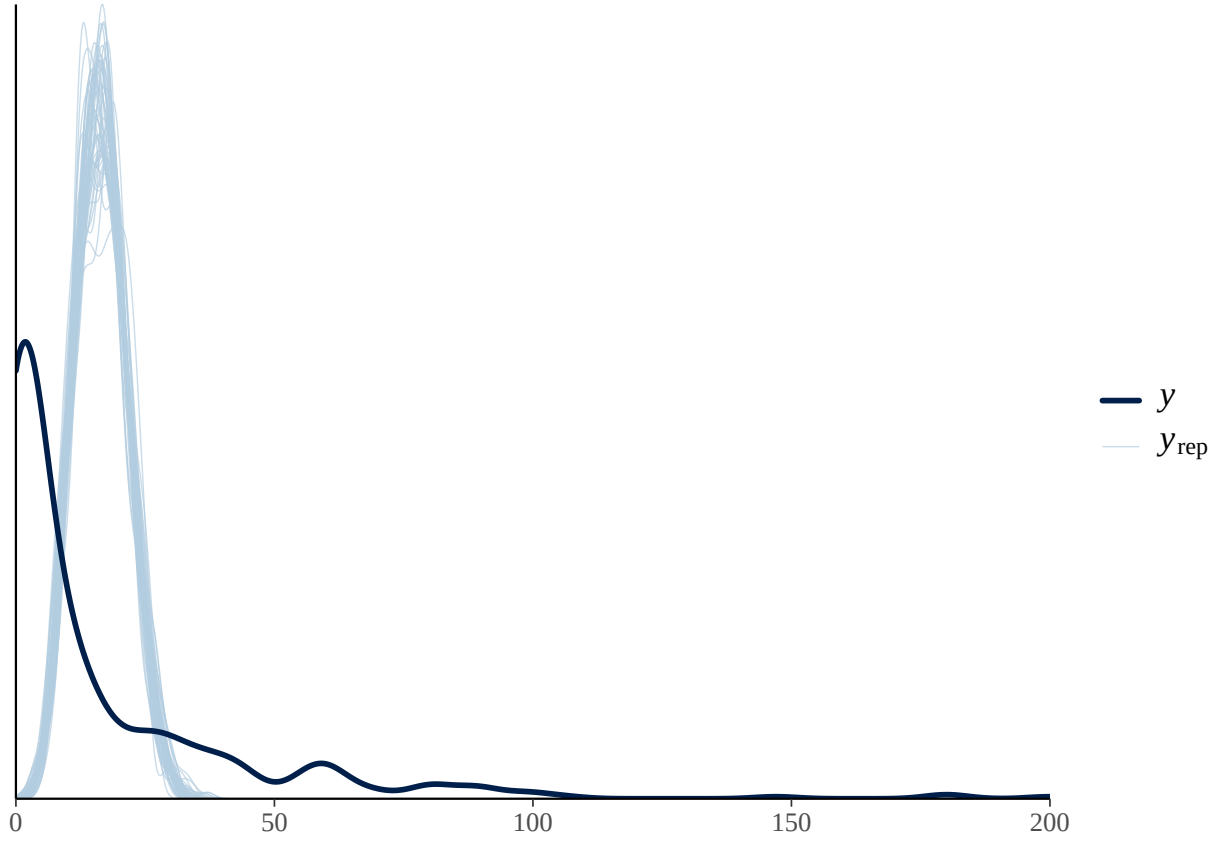
Model this outcome as a function of treatment assignment using a Poisson regression. Does the model fit well? Is there evidence of overdispersion?
The variables in the model do have statistical significance, but the overdispersion is way larger than 1, so it is not a good model.

```
risky<-read.csv("~/Desktop/Statistical Practice/MA 678/678-HW/risky.csv")

risky$fupacts_round<-round(risky$fupacts)
model15.1a<-stan_glm(fupacts_round~women_alone, data = risky, family = poisson(link = "log"), refresh =

summary(model15.1a, digits = 3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       poisson [log]
##  formula:      fupacts_round ~ women_alone
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   2
##
## Estimates:
##                mean   sd     10%    50%    90%
## (Intercept)  2.921  0.014  2.904  2.922  2.939
## women_alone -0.404  0.028 -0.439 -0.404 -0.369
##
## Fit Diagnostics:
##            mean   sd     10%    50%    90%
```

```
## mean_PPD 16.492  0.274 16.134 16.495 16.843
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse  Rhat  n_eff
## (Intercept)   0.000 1.001 2282
## women_alone   0.001 1.000 2423
## mean_PPD      0.005 1.001 2495
## log-posterior 0.025 1.002 1769
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
pp_check(model15.1a)
```




```
dispersiontest(model15.1a)
```

```
##
##  Overdispersion test
##
## data:  model15.1a
## z = 4.9321, p-value = 4.067e-07
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##   42.96139
```

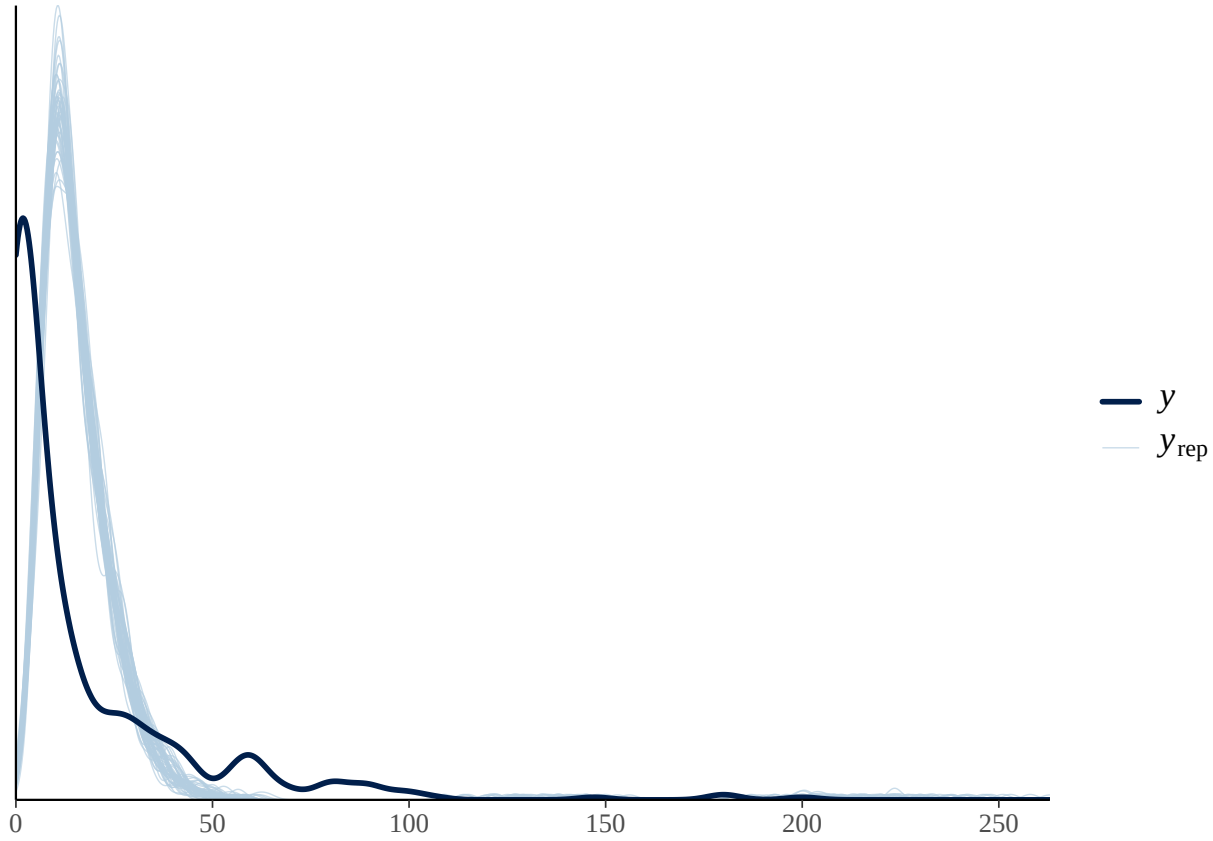
b)

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?
THe model fits better than the last one with 4 significant predictors, however the overdispersion is again quite large.

```
model15.1b<-stan_glm(fupacts_round~ women_alone+ couples+bs_hiv+bupacts, data= risky, family = poisson
summary(model15.1b, digits = 3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       poisson [log]
##  formula:      fupacts_round ~ women_alone + couples + bs_hiv + bupacts
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   5
##
## Estimates:
##                   mean   sd     10%    50%    90%
## (Intercept)      2.842  0.020  2.816  2.842  2.867
## women_alone     -0.657  0.030 -0.695 -0.657 -0.617
## couples         -0.414  0.029 -0.450 -0.414 -0.377
## bs_hivpositive -0.433  0.035 -0.478 -0.434 -0.388
## bupacts          0.011  0.000  0.011  0.011  0.011
##
## Fit Diagnostics:
##            mean   sd     10%    50%    90%
## mean_PPD 16.489  0.266 16.152 16.488 16.827
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                mcse  Rhat  n_eff
## (Intercept)    0.000 1.000 3871
## women_alone    0.001 1.001 2531
## couples        0.001 1.001 2883
## bs_hivpositive 0.001 0.999 3162
## bupacts        0.000 0.999 4397
## mean_PPD       0.004 1.000 3681
## log-posterior  0.036 1.002 1886
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
pp_check(model15.1b)
```




```
dispersiontest(model15.1b)
```

```
##
##  Overdispersion test
##
## data:  model15.1b
## z = 5.6824, p-value = 6.642e-09
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##   29.73382
```

c)

Fit a negative binomial (overdispersed Poisson) model. What do you conclude regarding effectiveness of the intervention? Since the three coefficients besides the before treatment are all negative it would suggest that the treatment worked.

```
model15.1c<-glm.nb(fupacts_round~ women_alone+ couples+bs_hiv+bupacts, data= risky, link = "log")
summary(model15.1c)
```

```
##
## Call:
## glm.nb(formula = fupacts_round ~ women_alone + couples + bs_hiv +
##     bupacts, data = risky, link = "log", init.theta = 0.4168882054)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.450274   0.154745  15.834  < 2e-16 ***
## women_alone    -0.715797   0.192398  -3.720 0.000199 ***
## couples        -0.349628   0.188987  -1.850 0.064311 .
## bs_hivpositive -0.551637   0.185920  -2.967 0.003007 **
## bupacts         0.022374   0.002362   9.471  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4169) family taken to be 1)
##
##     Null deviance: 581.84  on 433  degrees of freedom
## Residual deviance: 487.67  on 429  degrees of freedom
## AIC: 2967.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4169
##           Std. Err.:  0.0313
##
##  2 x log-likelihood:  -2955.7500
```

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, because we did not account for differences across gender, and with the women alone category it suggests a higher percentage of women in the study.

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.

```
N<-100
height<-rnorm(N,72,3)
p<- 0.4 +0.1*(height-72)/3
n<-runif(N,10, 30) %>% round()
y<-rbinom(N,n,p)
data15.3a<- data.frame( n= n, y =y, height = height)
model15.3a<-stan_glm(cbind(y,n-y)~height, family = binomial(link = 'logit'), data = data15.3a, refresh =
```

```
summary(model15.3a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      cbind(y, n - y) ~ height
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 100
##  predictors:   2
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept)  -9.6    1.1 -11.0  -9.6  -8.2
## height        0.1    0.0   0.1   0.1   0.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 8.1    0.3  7.7   8.1   8.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2592
## height        0.0  1.0  2607
## mean_PPD      0.0  1.0  3302
## log-posterior 0.0  1.0  1823
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(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)
n<-rep(20,N)
y<-rbinom(N,n,p)
data15.3b<-data.frame(n=n, y=y, height= height)
model15.3b<-stan_glm(cbind(y,n-y)~height, family = binomial(link = "logit"), data =data15.3b, refresh =

summary(model15.3b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
```

```
##  formula:      cbind(y, n - y) ~ height
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 100
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -0.3    1.1 -1.7  -0.2   1.2
## height       0.0    0.0  0.0   0.0   0.0
##
## Fit Diagnostics:
##           mean   sd   10%   50%   90%
## mean_PPD 7.8    0.3  7.4   7.8   8.2
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2474
## height        0.0  1.0  2509
## mean_PPD      0.0  1.0  3349
## log-posterior 0.0  1.0  1898
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

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.

The intercept shows on average 422.8 after treatment for individuals with all other predictors as 0. The treatment shows a decrease of 5,977 on average for those in that group. Each additional year older is associated with a 59.55 increase on average. Each year of education is associated with an increase of 710.6 on average. The married group is on average 5,517 higher than the non married group. The log scale is related to the standard deviation.

```
lalonde<-read_dta("~/Desktop/Statistical Practice/MA 678/678-HW/NSW_dw_obs.dta")
model15.7<-tobit(re78~treat +age+educ+married, data = lalonde, left = 0)
summary(model15.7)
```

```
##
## Call:
## tobit(formula = re78 ~ treat + age + educ + married, left = 0,
##     data = lalonde)
##
## Observations:
##          Total  Left-censored     Uncensored Right-censored
##          18667           2503          16164              0
##
## Coefficients:
```

```
##               Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  4.228e+02  4.891e+02    0.864    0.387
## treat       -5.977e+03  9.041e+02   -6.611 3.81e-11 ***
## age          5.955e+01  8.819e+00    6.752 1.45e-11 ***
## educ         7.106e+02  3.030e+01   23.450  < 2e-16 ***
## married      5.517e+03  2.161e+02   25.530  < 2e-16 ***
## Log(scale)   9.366e+00  5.776e-03 1621.486  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Scale: 11680
##
## Gaussian distribution
## Number of Newton-Raphson Iterations: 3
## Log-likelihood: -1.773e+05 on 6 Df
## Wald-statistic:  1673 on 4 Df, p-value: < 2.22e-16
```

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("~/Desktop/Statistical Practice/MA 678/678-HW/congress.csv")
congress15.8<-data.frame(v86 = congress$v86_adj, v88 = congress$v88_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.

```
model15.8a<-rstanarm::stan_glm(v88~v86+inc, data=congress15.8, refresh =0)
summary(model15.8a, digits = 3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      v88 ~ v86 + inc
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 435
##  predictors:   3
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) 0.238  0.017 0.216 0.238 0.260
## v86         0.520  0.033 0.479 0.520 0.563
## inc         0.097  0.007 0.088 0.097 0.105
## sigma       0.067  0.002 0.065 0.067 0.070
```

```
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 0.543  0.005 0.537 0.543 0.549
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse  Rhat  n_eff
## (Intercept)   0.000 1.000 1518
## v86           0.001 1.000 1454
## inc           0.000 1.000 1527
## sigma         0.000 1.002 2103
## mean_PPD      0.000 0.999 3725
## log-posterior 0.036 1.003 1661
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(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.

```
model15.8b<-brm(v88~v86+inc, data=congress15.8, refresh =0)
```

```
## Compiling Stan program...
```

```
## Trying to compile a simple C file
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: 'Apple clang version 16.0.0 (clang-1600.0.26.3)'
## using SDK: 'MacOSX15.0.sdk'
## clang -arch x86_64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Frame
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/library/RcppEiger
## In file included from /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/library/RcppEiger
## /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/library/RcppEigen/include/Eigen/src/Co
##   679 | #include <cmath>
##       |          ^~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```
## Start sampling
```

```
summary(model15.8b, digits =3)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: v88 ~ v86 + inc
##    Data: congress15.8 (Number of observations: 435)
```

```
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.24      0.02     0.20     0.27 1.00     1839     2071
## v86           0.52      0.03     0.45     0.58 1.00     1793     2071
## inc           0.10      0.01     0.08     0.11 1.00     1817     2224
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.07      0.00     0.06     0.07 1.00     2392     2230
##
## Draws were sampled 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).
```

(c)

Which model do you prefer? The two models are very similar, but the smaple size is large enough where I think it would be better to us the nomral distribution.

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.

```
model15.9a<-rstanarm::stan_glm(as.numeric(v88)~v86+inc, data=congress15.8, refresh =0, binomial(link =
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model15.9a, digits = 3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [probit]
##  formula:      as.numeric(v88) ~ v86 + inc
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 435
```

```
##  predictors:   3
##
## Estimates:
##               mean    sd      10%     50%     90%
## (Intercept)  -7.130   5.509 -14.548  -6.481  -0.675
## v86           0.120   9.430 -11.891   0.129  12.387
## inc           0.509   2.383  -2.450   0.395   3.620
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 0.000  0.001 0.000 0.000 0.002
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse  Rhat  n_eff
## (Intercept)   0.194 1.004  806
## v86           0.292 1.005 1040
## inc           0.074 1.003 1032
## mean_PPD      0.000 1.000 3105
## log-posterior 0.047 1.006  748
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(b)

Fit a robit regression and assess model fit.

```
model15.9b<-glm(as.numeric(v88)~v86+inc, data=congress15.8, binomial(link = gosset(2)))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(model15.9b, digits = 3)
```

```
##
## Call:
## glm(formula = as.numeric(v88) ~ v86 + inc, family = binomial(link = gosset(2)),
##     data = congress15.8)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8637     0.4025  -2.146   0.0319 *
## v86           1.7220     0.7662   2.248   0.0246 *
## inc           0.2794     0.1542   1.812   0.0700 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 72.2482  on 434  degrees of freedom
## Residual deviance:  9.0419  on 432  degrees of freedom
## AIC: 345.21
```

```
##
## Number of Fisher Scoring iterations: 4
```

(c)

Which model do you prefer? The robit model is a little bit better because the intercept and vote 86 are both significant while in the probit they are not.

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.

```
model15.14a<-rstanarm::stan_glm(fupacts_round~bs_hiv, data= risky, poisson(link ="log"), refresh =0)

pred15.14a<-posterior_predict(model15.14a, 1000, newdata = risky)
for (i in 1:1000){
  eq0a<- sum(pred15.14a[i,]==0)
  over10a <- sum(pred15.14a[i,]>10)
}
print(eq0a, digits = 4)
```

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

```
print(over10a/434, digits = 4)
```

```
## [1] 0.8664
```

(b)

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

```
model15.14b<-rstanarm::stan_glm(fupacts_round~bs_hiv, data= risky, neg_binomial_2(link ="log"), refresh

pred15.14b<-posterior_predict(model15.14b, 1000, newdata = risky)
for (i in 1:1000){
  eq0b<- sum(pred15.14b[i,]==0)
  over10b <- sum(pred15.14b[i,]>10)
}
print(eq0b/434, digits = 4)
```

```
## [1] 0.2143
```

```
print(over10b/434, digits = 4)
```

```
## [1] 0.4194
```

(c)

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

```
model15.14c<-rstanarm::stan_glm(fupacts_round~bs_hiv+bupacts, data= risky, neg_binomial_2(link ="log"),

pred15.14c<-posterior_predict(model15.14c, 1000, newdata = risky)
for (i in 1:1000){
  eq0c<- sum(pred15.14c[i,]==0)
  over10c <- sum(pred15.14c[i,]>10)
}
print(eq0c/434, digits = 4)
```

```
## [1] 0.2235
```

```
print(over10c/434, digits =4)
```

```
## [1] 0.4055
```

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.

```
model15.15a<- glm(re78>0~treat +age+educ+married, data = lalonde, binomial)
summary(model15.15a)
```

```
##
## Call:
## glm(formula = re78 > 0 ~ treat + age + educ + married, family = binomial,
##     data = lalonde)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.911523   0.123602  23.556  < 2e-16 ***
## treat       -0.823391   0.176476  -4.666 3.08e-06 ***
## age         -0.030555   0.002124 -14.389  < 2e-16 ***
## educ        -0.021468   0.007416  -2.895  0.00379 **
## married      0.378918   0.052981   7.152 8.56e-13 ***
## ---
## 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: 14490  on 18662  degrees of freedom
## AIC: 14500
##
## Number of Fisher Scoring iterations: 4
```

```r
model15.15b<-lm(log(re78)~treat+age+educ+married, data = lalonde[(lalonde$re78>0)==1,])
summary(model15.15b)
```

```
##
## Call:
## lm(formula = log(re78) ~ treat + age + educ + married, data = lalonde[(lalonde$re78 >
##     0) == 1, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6818 -0.1521  0.2443  0.4743  2.4421
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.1016386  0.0395154 205.025  < 2e-16 ***
## treat       -0.5395909  0.0759441  -7.105 1.25e-12 ***
## age          0.0128681  0.0007223  17.815  < 2e-16 ***
## educ         0.0584326  0.0024542  23.810  < 2e-16 ***
## married      0.4394528  0.0175060  25.103  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8879 on 16159 degrees of freedom
## Multiple R-squared:  0.1217, Adjusted R-squared:  0.1215
## F-statistic: 559.8 on 4 and 16159 DF,  p-value: < 2.2e-16
```