MA678 Homework 5

JingJianGao

10/25/2022

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 over-dispersion?

```r
library(ggplot2)
library(rstanarm)
```

```
## Loading required package: Rcpp

## This is rstanarm version 2.21.3

## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

##   options(mc.cores = parallel::detectCores())
```

```r
library(performance)
risky <- read.csv("/Users/billg/Desktop/MA-678-Homework/MA678-HW5/risky.csv")
fupac <- round(risky$fupacts)
women_alone <- as.factor(risky$women_alone)
Reg15.1 <- stan_glm(fupac ~ women_alone, family= poisson(link="log"),
                    data=risky, refresh=0)
summary(Reg15.1)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       poisson [log]
##  formula:      fupac ~ 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)  2.9    0.0  2.9   2.9   2.9 
## women_alone -0.4    0.0 -0.4  -0.4  -0.4 
##
## 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 outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2780 
## women_alone   0.0  1.0  3034 
## mean_PPD      0.0  1.0  3025 
## log-posterior 0.0  1.0  1816 
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
check_overdispersion(Reg15.1)
```

```
## # Overdispersion test
##
##        dispersion ratio =    43.199
##   Pearson's Chi-Squared = 18662.105
##                 p-value =   < 0.001

## Overdispersion detected.
```

```
# The model does not seem to fit well. And there is over-dispersion
```

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?

```
Reg15.1b <- stan_glm(fupac~women_alone+couples+bs_hiv+sex+bupacts,
                     family=poisson("log"),data=risky,refresh=0)
summary(Reg15.1b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       poisson [log]
##  formula:      fupac ~ women_alone + couples + bs_hiv + sex + bupacts
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   6
##
## Estimates:
##                  mean   sd   10%   50%   90%
## (Intercept)     2.8    0.0  2.8   2.8   2.8 
## women_alone    -0.7    0.0 -0.7  -0.7  -0.6 
## couples        -0.4    0.0 -0.4  -0.4  -0.4 
## bs_hivpositive -0.4    0.0 -0.5  -0.4  -0.4
## sexwoman        0.1    0.0  0.1   0.1   0.1
## bupacts         0.0    0.0  0.0   0.0   0.0
##
## 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 outcome variable (for de
##
## MCMC diagnostics
##                mcse Rhat n_eff
## (Intercept)    0.0  1.0  3612
## women_alone    0.0  1.0  2600
## couples        0.0  1.0  3001
## bs_hivpositive 0.0  1.0  3247
## sexwoman       0.0  1.0  3872
## bupacts        0.0  1.0  4439
## mean_PPD       0.0  1.0  3158
## log-posterior  0.0  1.0  1664
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
check_overdispersion(Reg15.1b)
```

```
## # Overdispersion test
##
##        dispersion ratio =    30.030
##   Pearson's Chi-Squared = 12852.626
##                 p-value =   < 0.001
```

```
## Overdispersion detected.
```

```
# The model fits better but there is still over-dispersion.
```

c)

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

```
Reg15.1c <- stan_glm(fupac~women_alone+couples+bs_hiv+sex+bupacts,
                     family=neg_binomial_2(link="log"),data=risky,refresh=0)
summary(Reg15.1c)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       neg_binomial_2 [log]
##  formula:      fupac ~ women_alone + couples + bs_hiv + sex + bupacts
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   6
##
## Estimates:
##                         mean   sd   10%   50%   90%
## (Intercept)           2.5    0.2  2.2   2.5   2.7
## women_alone          -0.7    0.2 -1.0  -0.7  -0.5
## couples              -0.4    0.2 -0.6  -0.4  -0.1
## bs_hivpositive       -0.5    0.2 -0.8  -0.5  -0.3
## sexwoman              0.0    0.2 -0.2   0.0   0.2
## bupacts               0.0    0.0  0.0   0.0   0.0
## reciprocal_dispersion  0.4    0.0  0.4   0.4   0.5
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 49.3   63.1 18.0  31.6  94.9
##
## 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  4237
## women_alone           0.0  1.0  3734
## couples               0.0  1.0  3598
## bs_hivpositive        0.0  1.0  4788
## sexwoman              0.0  1.0  4885
## bupacts               0.0  1.0  5273
## reciprocal_dispersion 0.0  1.0  3858
## mean_PPD              1.0  1.0  3639
## log-posterior         0.0  1.0  1733
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
check_overdispersion(Reg15.1c)
```

```
## # Overdispersion test
##
##        dispersion ratio =    49.836
##   Pearson's Chi-Squared = 21280.037
##                 p-value =   < 0.001
```

```
## Overdispersion detected.
```

```
# I gave up making graphs because R kept saying Polygon Edge not Found.
# I would say the intervention had a positive impact on lowering the unprotexted sex act.
```

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 does give me concern with regard to our modeling assumptions.
# Because there may be unexpected interactions in the model which will affect the simulation.
```

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.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
set.seed(110)
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)
data <- data.frame(n=n, y=y, height=height)
fit_1a <- stan_glm(cbind(y, n-y) ~ height, family=binomial(link="logit"),
     data=data,refresh=0)
summary(fit_1a)
```

```
##
## 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) -13.4    1.4 -15.2 -13.4 -11.7
## height        0.2    0.0   0.2   0.2   0.2
##
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 7.6    0.3   7.2   7.6   8.0
##
## 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  2915
## height        0.0  1.0  2923
## mean_PPD      0.0  1.0  3407
## log-posterior 0.0  1.0  1610
##
## 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)
p <- 0.4 + 0.1*(height - 72)/3
n <- rep(20,N)
y <- rbinom(N, n, p)
datab <- data.frame(n=n, y=y, height=height)
fit_1b <- stan_glm(cbind(y, n-y) ~ height, family=binomial(link="logit"),
     data=datab,refresh=0)
summary(fit_1b)
```

```
##
## 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) -10.1    1.1 -11.5 -10.1  -8.7
## height        0.1    0.0   0.1   0.1   0.2
##
## Fit Diagnostics:
##             mean   sd   10%   50%   90%
## mean_PPD 7.9    0.3  7.5   7.9   8.3
##
## 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  2708
## height        0.0  1.0  2705
## mean_PPD      0.0  1.0  3201
## log-posterior 0.0  1.0  1827
##
## 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.

```
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
##
## Attaching package: 'VGAM'

## The following objects are masked from 'package:rstanarm':
##
##     cauchy, dirichlet, exponential, laplace, logit
```

```
lalonde <- foreign::read.dta("NSW_dw_obs.dta")
summary(lalonde)
```

```
##       age             educ           black           married      
##  Min.   :16.00   Min.   : 0.00   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:24.00   1st Qu.:11.00   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :31.00   Median :12.00   Median :0.0000   Median :1.0000  
##  Mean   :33.37   Mean   :12.02   Mean   :0.1048   Mean   :0.7272  
##  3rd Qu.:42.00   3rd Qu.:14.00   3rd Qu.:0.0000   3rd Qu.:1.0000  
##  Max.   :55.00   Max.   :18.00   Max.   :1.0000   Max.   :1.0000  
##     nodegree           re74             re75             re78       
##  Min.   :0.0000   Min.   :     0   Min.   :     0   Min.   :     0  
##  1st Qu.:0.0000   1st Qu.:  4898   1st Qu.:  4726   1st Qu.:  6158  
##  Median :0.0000   Median : 15525   Median : 14899   Median : 16957  
##  Mean   :0.3012   Mean   : 14621   Mean   : 14253   Mean   : 15657  
##  3rd Qu.:1.0000   3rd Qu.: 23882   3rd Qu.: 23274   3rd Qu.: 25565  
##  Max.   :1.0000   Max.   :137149   Max.   :156653   Max.   :121174  
##       hisp              sample          treat            educ_cat4    
##  Min.   :0.00000   Min.   :1.000   Min.   :0.00000   Min.   :1.000  
##  1st Qu.:0.00000   1st Qu.:2.000   1st Qu.:0.00000   1st Qu.:1.000  
##  Median :0.00000   Median :2.000   Median :0.00000   Median :2.000  
##  Mean   :0.06664   Mean   :2.123   Mean   :0.00991   Mean   :2.165  
##  3rd Qu.:0.00000   3rd Qu.:2.000   3rd Qu.:0.00000   3rd Qu.:3.000  
##  Max.   :1.00000   Max.   :3.000   Max.   :1.00000   Max.   :4.000
```

```
re78 <- round(lalonde$re78)
Reg15.7 <- vglm(re78 ~ treat,family=tobit,data=lalonde,refresh=0)
summary(Reg15.7)
```

```
##
## Call:
## vglm(formula = re78 ~ treat, family = tobit, data = lalonde,
##     refresh = 0)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept):1  1.496e+04  9.127e+01  163.92   <2e-16 ***
## (Intercept):2  9.414e+00  5.662e-03 1662.61   <2e-16 ***
## treat         -1.065e+04  9.584e+02  -11.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: mu, loglink(sd)
##
## Log-likelihood: -177990.8 on 37331 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
```

```
## '(Intercept):2'
```

```
Reg15.72 <- vglm(re78~ re75+treat+educ+age,family=tobit,data=lalonde,refresh=0)
summary(Reg15.72)
```

```
##
## Call:
## vglm(formula = re78 ~ re75 + treat + educ + age, family = tobit,
##     data = lalonde, refresh = 0)
##
## Coefficients:
##                 Estimate Std. Error  z value Pr(>|z|)
## (Intercept):1  5.431e+03  3.665e+02   14.818  < 2e-16 ***
## (Intercept):2  9.067e+00  5.589e-03 1622.197  < 2e-16 ***
## re75           8.521e-01  6.990e-03  121.899  < 2e-16 ***
## treat          1.498e+02  6.731e+02    0.222    0.824
## educ           1.261e+02  2.309e+01    5.462 4.71e-08 ***
## age           -1.234e+02  6.389e+00  -19.313  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: mu, loglink(sd)
##
## Log-likelihood: -172040.3 on 37328 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
```

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("congress.csv")
```

(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.

```
Reg15.8 <- stan_glm(v88_adj~v86_adj+inc88,data=congress,refresh=0)
summary(Reg15.8)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      v88_adj ~ v86_adj + inc88
##  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
## v86_adj     0.5    0.0  0.5   0.5   0.6
## inc88       0.1    0.0  0.1   0.1   0.1
## 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 outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  1940
## v86_adj       0.0  1.0  1890
## inc88         0.0  1.0  1826
## sigma         0.0  1.0  2231
## mean_PPD      0.0  1.0  3720
## log-posterior 0.0  1.0  1678
##
## 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.

```
library(brms)
```

```
## Loading 'brms' package (version 2.18.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
```

```
##
## Attaching package: 'brms'
```

```
## The following objects are masked from 'package:VGAM':
##
##     acat, cratio, cumulative, dfrechet, dirichlet, exponential,
##     frechet, geometric, lognormal, multinomial, negbinomial, pfrechet,
##     qfrechet, rfrechet, s, sratio
```

```
## The following objects are masked from 'package:rstanarm':
##
##     dirichlet, exponential, get_y, lasso, ngrps
```

```
## The following object is masked from 'package:stats':
##
##     ar
```

```
Reg15.8b <- brm(v88_adj~ v86_adj+inc88,family=student,data=congress,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 -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Framew
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/
## In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen/src/Cor
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen/src/Cor
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen/Core:96
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

```
## Start sampling
```

```
summary(Reg15.8b)
```

```
##  Family: student
##   Links: mu = identity; sigma = identity; nu = identity
## Formula: v88_adj ~ v86_adj + inc88
##    Data: congress (Number of observations: 435)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 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     1867     1800
## v86_adj       0.55      0.03     0.48     0.62 1.00     1803     2035
## inc88         0.09      0.01     0.08     0.11 1.00     1837     1952
##
## 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     1873     2091
## nu        6.16      2.42     3.31    12.56 1.00     1929     2139
##
## 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?

#I would prefer to use the t distribution since it works better to get prediction.

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.

```
library(rstanarm)
Reg15.9 <- stan_glm(v88_adj>0.5 ~ v86_adj+inc88,
                    family=binomial(link="logit"),data=congress,refresh=0)
summary(Reg15.9)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      v88_adj > 0.5 ~ v86_adj + inc88
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 435
##  predictors:   3
##
## Estimates:
##                 mean   sd   10%   50%   90%
## (Intercept) -5.7    1.3 -7.5  -5.7  -4.1
## v86_adj     11.6    2.5  8.4  11.5  15.0
## inc88        2.7    0.5  2.1   2.7   3.3
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.6    0.0  0.6   0.6   0.6
##
## 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  2609
## v86_adj       0.1  1.0  2516
## inc88         0.0  1.0  2371
## mean_PPD      0.0  1.0  3653
## log-posterior 0.0  1.0  1640
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(b)

Fit a probit regression and assess model fit.

```
Reg15.9b <- stan_glm(v88_adj>0.5 ~ v86_adj+inc88,
                     family=binomial(link="probit"),data=congress,refresh=0)
summary(Reg15.9)
```

```
##
```

```
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      v88_adj > 0.5 ~ v86_adj + inc88
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 435
##  predictors:   3
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -5.7    1.3 -7.5  -5.7  -4.1
## v86_adj     11.6    2.5  8.4  11.5  15.0
## inc88        2.7    0.5  2.1   2.7   3.3
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.6    0.0  0.6   0.6   0.6
##
## 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  2609
## v86_adj       0.1  1.0  2516
## inc88         0.0  1.0  2371
## mean_PPD      0.0  1.0  3653
## log-posterior 0.0  1.0  1640
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(c)

Which model do you prefer?

#Even though the results are pretty much the same, I would prefer to use "Probit"

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.

```r
Reg15.14 <- stan_glm(fupac~bs_hiv, family=poisson(link="log"),data=risky,refresh=0)
ppredict <- posterior_predict(Reg15.14,draws=1000, newdata=risky)
for (i in 1:1000){
  per0 <- sum(ppredict[i,]==0)
  per10 <- sum(ppredict[i,]>10)
}
```

```
per0 <- per0/434
per10 <- per10/434
print(per0)
```

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

```
print(per10)
```

```
## [1] 0.8525346
```

```
mean0 <- mean(risky$fupacts == 0)
mean10 <- mean(risky$fupacts >10)
print(mean0)
```

```
## [1] 0.2926267
```

```
print(mean10)
```

```
## [1] 0.3640553
```

(b)

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

```
Reg15.14b <- stan_glm(fupac~bs_hiv, family=neg_binomial_2(link="log"),data=risky,refresh=0)
ppredict2 <- posterior_predict(Reg15.14b,draws=1000,data=risky)
for (i in 1:1000){
  p0 <- sum(ppredict2[i,]==0)
  p10 <- sum(ppredict2[i,]>10)
}
percent0 <- p0/434
percent10 <- p10/434
print(percent0)
```

```
## [1] 0.2534562
```

```
print(percent10)
```

```
## [1] 0.3364055
```

(c)

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

```
Reg15.14c <- stan_glm(fupac ~ bs_hiv+bupacts,
                      family=neg_binomial_2(link="log"),data=risky,refresh=0)
ppredict3 <- posterior_predict(Reg15.14c,draws=1000,data=risky)
for (i in 1:1000){
  p0 <- sum(ppredict3[i,]==0)
  p10 <- sum(ppredict3[i,]>10)
}
percent0 <- p0/434
percent10 <- p10/434
print(percent0)
```

```
## [1] 0.2442396
```

```
print(percent10)
```

```
## [1] 0.3248848
```

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.

```
summary(lalonde)
```

```
##       age             educ           black           married
##  Min.   :16.00   Min.   : 0.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:24.00   1st Qu.:11.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :31.00   Median :12.00   Median :0.0000   Median :1.0000
##  Mean   :33.37   Mean   :12.02   Mean   :0.1048   Mean   :0.7272
##  3rd Qu.:42.00   3rd Qu.:14.00   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :55.00   Max.   :18.00   Max.   :1.0000   Max.   :1.0000
##     nodegree          re74             re75             re78
##  Min.   :0.0000   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:0.0000   1st Qu.:  4898   1st Qu.:  4726   1st Qu.:  6158
##  Median :0.0000   Median : 15525   Median : 14899   Median : 16957
##  Mean   :0.3012   Mean   : 14621   Mean   : 14253   Mean   : 15657
##  3rd Qu.:1.0000   3rd Qu.: 23882   3rd Qu.: 23274   3rd Qu.: 25565
##  Max.   :1.0000   Max.   :137149   Max.   :156653   Max.   :121174
##       hisp             sample          treat            educ_cat4
##  Min.   :0.00000   Min.   :1.000   Min.   :0.00000   Min.   :1.000
##  1st Qu.:0.00000   1st Qu.:2.000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :0.00000   Median :2.000   Median :0.00000   Median :2.000
##  Mean   :0.06664   Mean   :2.123   Mean   :0.00991   Mean   :2.165
##  3rd Qu.:0.00000   3rd Qu.:2.000   3rd Qu.:0.00000   3rd Qu.:3.000
##  Max.   :1.00000   Max.   :3.000   Max.   :1.00000   Max.   :4.000
```

```
zero_earning <- lalonde$re78 ==0
positive_earning <- lalonde$re78 >0
Reg15.15 <-  stan_glm(zero_earning ~ educ+age+re74+re75,
                      family = binomial(link="logit"), data=lalonde,refresh=0)
Reg15.15b <- lm(zero_earning ~ educ+age+re74+re75, data=lalonde)
Reg15.152 <- stan_glm(positive_earning ~ educ+age+re74+re75,
                      family = binomial(link="logit"), data=lalonde,refresh=0)
Reg15.152b <- lm(positive_earning ~ educ+age+re74+re75, data=lalonde)

summary(Reg15.15)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      zero_earning ~ educ + age + re74 + re75
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 18667
##  predictors:   5
##
## Estimates:
```

```
##                mean   sd   10%   50%   90%
## (Intercept) -3.3    0.1 -3.5  -3.3  -3.2
## educ         0.1    0.0  0.1   0.1   0.1
## age          0.1    0.0  0.1   0.1   0.1
## re74         0.0    0.0  0.0   0.0   0.0
## re75         0.0    0.0  0.0   0.0   0.0
##
## Fit Diagnostics:
##             mean   sd   10%   50%   90%
## mean_PPD 0.1    0.0  0.1   0.1   0.1
##
## 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  3055
## educ          0.0  1.0  3376
## age           0.0  1.0  3370
## re74          0.0  1.0  2391
## re75          0.0  1.0  2254
## mean_PPD      0.0  1.0  3598
## log-posterior 0.0  1.0  1601
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
summary(Reg15.15b)
```

```
##
## Call:
## lm(formula = zero_earning ~ educ + age + re74 + re75, data = lalonde)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5639 -0.1746 -0.0877  0.0079  1.7155
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.373e-02  1.298e-02  -4.911 9.14e-07 ***
## educ         1.217e-02  8.209e-04  14.830  < 2e-16 ***
## age          7.708e-03  2.305e-04  33.436  < 2e-16 ***
## re74        -4.962e-06  4.570e-07 -10.859  < 2e-16 ***
## re75        -9.346e-06  4.583e-07 -20.395  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3137 on 18662 degrees of freedom
## Multiple R-squared:  0.1527, Adjusted R-squared:  0.1525
## F-statistic: 840.9 on 4 and 18662 DF,  p-value: < 2.2e-16
```

```
summary(Reg15.152)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      positive_earning ~ educ + age + re74 + re75
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 18667
##  predictors:   5
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  3.3    0.1  3.2   3.3   3.5
## educ        -0.1    0.0 -0.1  -0.1  -0.1
## age         -0.1    0.0 -0.1  -0.1  -0.1
## re74         0.0    0.0  0.0   0.0   0.0
## re75         0.0    0.0  0.0   0.0   0.0
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.9    0.0  0.9   0.9   0.9
##
## 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  3263
## educ          0.0  1.0  3598
## age           0.0  1.0  3254
## re74          0.0  1.0  2827
## re75          0.0  1.0  2931
## mean_PPD      0.0  1.0  4244
## log-posterior 0.0  1.0  1843
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
summary(Reg15.152b)
```

```
##
## Call:
## lm(formula = positive_earning ~ educ + age + re74 + re75, data = lalonde)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7155 -0.0079  0.0877  0.1746  0.5639
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.064e+00  1.298e-02   81.97   <2e-16 ***
## educ        -1.217e-02  8.209e-04  -14.83   <2e-16 ***
## age         -7.708e-03  2.305e-04  -33.44   <2e-16 ***
## re74         4.962e-06  4.570e-07   10.86   <2e-16 ***
## re75         9.346e-06  4.583e-07   20.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3137 on 18662 degrees of freedom
## Multiple R-squared:  0.1527, Adjusted R-squared:  0.1525
## F-statistic: 840.9 on 4 and 18662 DF,  p-value: < 2.2e-16
```