Stats 412 HW 2

Chris Meade

5/14/2018

Risky Behavior

The data `risky_behaviors.dta` is from a randomized experiment that targeted couples at high risk of HIV infection. Counseling sessions were provided to the treatment group 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. The response variable to be examined after three months was "number of unprotected sex acts."

```
library(foreign)
rb <- read.dta("http://www.stat.columbia.edu/~gelman/arm/examples/risky.behavior/risky_behaviors.dta", c
```

1

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

```
rb$fupacts <- round(rb$fupacts)

mod1 <- glm(fupacts ~ couples + women_alone, data = rb, family = poisson)
summary(mod1)
```

```
##
## Call:
## glm(formula = fupacts ~ couples + women_alone, family = poisson,
##     data = rb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.6285  -4.9794  -3.2015   0.9847  27.1502
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.08960    0.01901  162.55   <2e-16 ***
## couples     -0.32243    0.02737  -11.78   <2e-16 ***
## women_alone -0.57212    0.03023  -18.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 13299  on 433  degrees of freedom
## Residual deviance: 12925  on 431  degrees of freedom
## AIC: 14256
##
## Number of Fisher Scoring iterations: 6
```

```r
pchisq(mod1$deviance, df=mod1$df.residual, lower.tail=FALSE)
```

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

Since the p-value for the Deviance Goodness of Fit Test is 0, we conclude that the model is not a good fit for the data. To check for overdispersion, we fit a quasipoisson model to the data.

```r
mod2 <- glm(fupacts ~ couples + women_alone, data = rb, family = quasipoisson)
summary(mod2)
```

```
##
## Call:
## glm(formula = fupacts ~ couples + women_alone, family = quasipoisson,
##     data = rb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.6285  -4.9794  -3.2015   0.9847  27.1502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0896     0.1263  24.468   <2e-16 ***
## couples      -0.3224     0.1818  -1.773   0.0769 .
## women_alone  -0.5721     0.2008  -2.849   0.0046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 44.13468)
##
##     Null deviance: 13299  on 433  degrees of freedom
## Residual deviance: 12925  on 431  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

The dispersion parameters is 44.13, indicating that the conditional variance is much larger than the conditional expectation. Therefore there is strong evidence for overdispersion.

2

Estimate Extension: 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?

```r
mod3 <- glm(fupacts ~ sex + couples + women_alone + bs_hiv, data = rb, family = poisson)
summary(mod3)
```

```
##
## Call:
## glm(formula = fupacts ~ sex + couples + women_alone + bs_hiv,
##     family = poisson, data = rb)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -6.998  -4.996  -3.216   1.014  26.182
##
```

```
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.19800    0.02254 141.865  < 2e-16 ***
## sexman         -0.07737    0.02367  -3.269  0.00108 **
## couples        -0.25447    0.02757  -9.231  < 2e-16 ***
## women_alone    -0.54229    0.03026 -17.920  < 2e-16 ***
## bs_hivpositive -0.59183    0.03493 -16.941  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 13299  on 433  degrees of freedom
## Residual deviance: 12590  on 429  degrees of freedom
## AIC: 13924
##
## Number of Fisher Scoring iterations: 6
```

Residual deviance is smaller in this model, indicating that it is a better fit than the first.

```
mod4 <- glm(fupacts ~ sex + couples + women_alone + bs_hiv, data = rb, family = quasipoisson)
summary(mod4)
```

```
##
## Call:
## glm(formula = fupacts ~ sex + couples + women_alone + bs_hiv,
##     family = quasipoisson, data = rb)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -6.998  -4.996  -3.216   1.014  26.182
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.19800    0.14670  21.799  < 2e-16 ***
## sexman         -0.07737    0.15402  -0.502  0.61571
## couples        -0.25447    0.17939  -1.418  0.15677
## women_alone    -0.54229    0.19693  -2.754  0.00614 **
## bs_hivpositive -0.59183    0.22734  -2.603  0.00955 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 42.35095)
##
##     Null deviance: 13299  on 433  degrees of freedom
## Residual deviance: 12590  on 429  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

With a dispersion parameter of 42.35, we are still experiencing overdisperion.

3

Overdispersion: Fit an overdispersed (quasi-)Poisson model. Fit a negative binomial model. Compare the models to previous two you have fit. Finally, what do you conclude regarding effectiveness of the intervention?

```
library(MASS)
library(visreg)
```

```
## Warning: package 'visreg' was built under R version 3.4.3
```

```
quasipois <- glm(fupacts ~ sex + couples + women_alone + bs_hiv,
                 data = rb, family = quasipoisson)
negbin <- glm.nb(fupacts ~ sex + couples + women_alone + bs_hiv,
              data = rb)

summary(quasipois)
```

```
##
## Call:
## glm(formula = fupacts ~ sex + couples + women_alone + bs_hiv,
##     family = quasipoisson, data = rb)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -6.998  -4.996  -3.216   1.014  26.182
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.19800    0.14670  21.799  < 2e-16 ***
## sexman         -0.07737    0.15402  -0.502  0.61571
## couples        -0.25447    0.17939  -1.418  0.15677
## women_alone    -0.54229    0.19693  -2.754  0.00614 **
## bs_hivpositive -0.59183    0.22734  -2.603  0.00955 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 42.35095)
##
##     Null deviance: 13299  on 433  degrees of freedom
## Residual deviance: 12590  on 429  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```
summary(negbin)
```

```
##
## Call:
## glm.nb(formula = fupacts ~ sex + couples + women_alone + bs_hiv,
##     data = rb, init.theta = 0.3492889439, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7239  -1.5138  -0.5830   0.1688   2.7728
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.18800    0.17594  18.119  < 2e-16 ***
## sexman         -0.05703    0.16451  -0.347 0.728831
## couples        -0.17682    0.20479  -0.863 0.387912
## women_alone    -0.58904    0.20853  -2.825 0.004732 **
## bs_hivpositive -0.67376    0.20034  -3.363 0.000771 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3493) family taken to be 1)
##
##     Null deviance: 503.74  on 433  degrees of freedom
## Residual deviance: 486.58  on 429  degrees of freedom
## AIC: 3035.3
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3493
##           Std. Err.:  0.0252
##
##  2 x log-likelihood:  -3023.3370
```

```
mean(rb$fupacts-predict(quasipois,rb, response = "count"))
```

```
## [1] 13.73885
```

```
mean(rb$fupacts-predict(negbin,rb, response = "count"))
```

```
## [1] 13.74374
```

The AIC for the negative binomial model is an order of magnitude less than for the corresponding poisson model, indicating that it is a much better fit than the latter. However, with the quasi-poisson model, we can't make such inferences based on liklihood. As a result, it becomes difficult to compare the quasipoisson against other models. We look at the difference in residuals amongst the two models and find them to be almost equal. Thus we conclude that both models were a successful intervention against overdisperion and the negative binomial has the added bonus of liklihood comparability.

4

Hurdle Model?: Fit a hurdle model to this data. This is a classic data set for Poisson regression and overdispersion... i'm honestly curious if the hurdle model makes sense and improves over any of the other previous models you have built. Also compare rootograms for all.

```
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 3.4.2
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
hurdlePoisson <- hurdle(fupacts ~ sex + couples + women_alone + bs_hiv,
                 data = rb, dist = 'poisson', zero.dist = 'binomial')
```

```r
hurdleNB <- hurdle(fupacts ~ sex + couples + women_alone + bs_hiv,
                   data = rb, dist = 'negbin', zero.dist = 'binomial')

summary(hurdlePoisson)
```

```
##
## Call:
## hurdle(formula = fupacts ~ sex + couples + women_alone + bs_hiv,
##     data = rb, dist = "poisson", zero.dist = "binomial")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -2.1142 -1.4233 -0.9173  0.3941 18.4208
##
## Count model coefficients (truncated poisson with log link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.35852    0.02230 150.640  < 2e-16 ***
## sexman         -0.09030    0.02371  -3.808  0.00014 ***
## couples        -0.12221    0.02775  -4.405 1.06e-05 ***
## women_alone    -0.34171    0.03029 -11.282  < 2e-16 ***
## bs_hivpositive -0.17183    0.03521  -4.880 1.06e-06 ***
## Zero hurdle model coefficients (binomial with logit link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.6417     0.2617   6.273 3.55e-10 ***
## sexman           0.0796     0.2201   0.362  0.71757
## couples         -0.5452     0.2915  -1.870  0.06143 .
## women_alone     -0.7834     0.2929  -2.675  0.00747 **
## bs_hivpositive  -1.1835     0.2449  -4.833 1.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 12
## Log-likelihood: -4991 on 10 Df
```

```r
summary(hurdleNB)
```

```
##
## Call:
## hurdle(formula = fupacts ~ sex + couples + women_alone + bs_hiv,
##     data = rb, dist = "negbin", zero.dist = "binomial")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.7217 -0.5908 -0.4482  0.1933  8.3603
##
## Count model coefficients (truncated negbin with log link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.24183    0.15585  20.801  < 2e-16 ***
## sexman         -0.09694    0.15040  -0.645   0.5192
## couples        -0.11808    0.18679  -0.632   0.5273
## women_alone    -0.39132    0.18775  -2.084   0.0371 *
## bs_hivpositive -0.23313    0.21690  -1.075   0.2825
## Log(theta)     -0.57493    0.13491  -4.262 2.03e-05 ***
## Zero hurdle model coefficients (binomial with logit link):
```

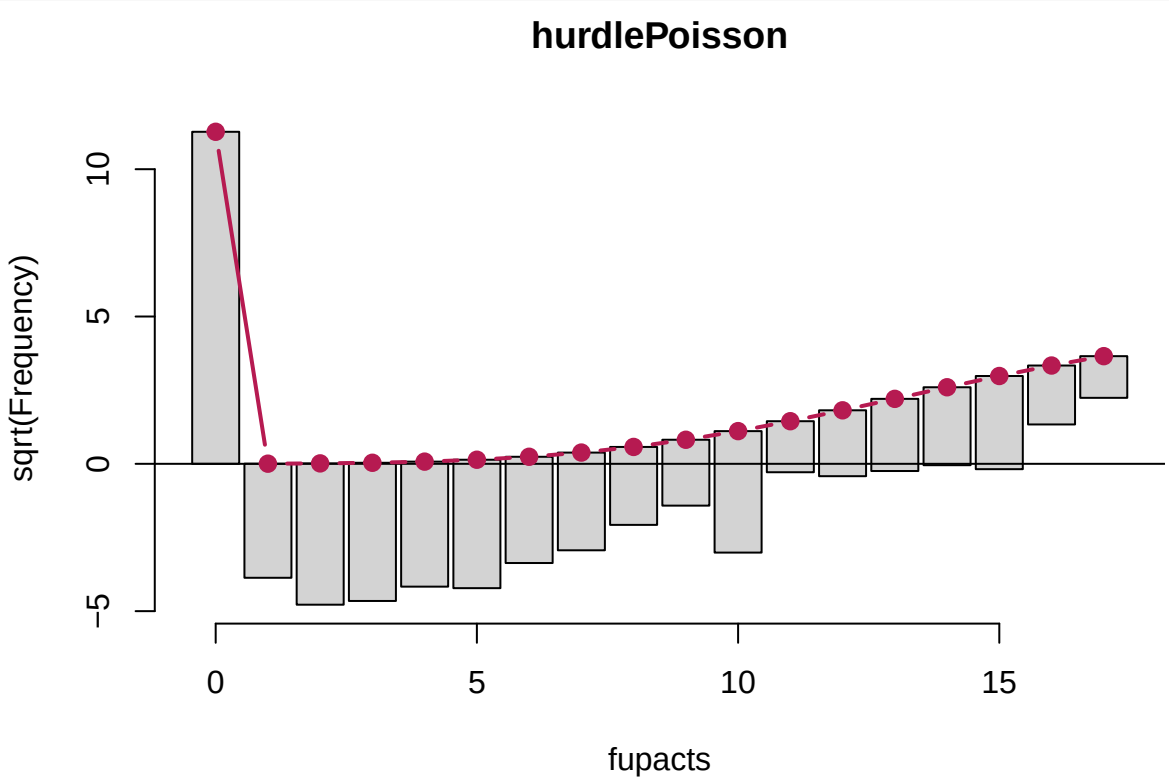
```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.6417     0.2617   6.273 3.55e-10 ***
## sexman           0.0796     0.2201   0.362  0.71757
## couples         -0.5452     0.2915  -1.870  0.06143 .
## women_alone     -0.7834     0.2929  -2.675  0.00747 **
## bs_hivpositive  -1.1835     0.2449  -4.833 1.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 0.5627
## Number of iterations in BFGS optimization: 13
## Log-likelihood: -1495 on 11 Df
```

```
library(countreg)
```

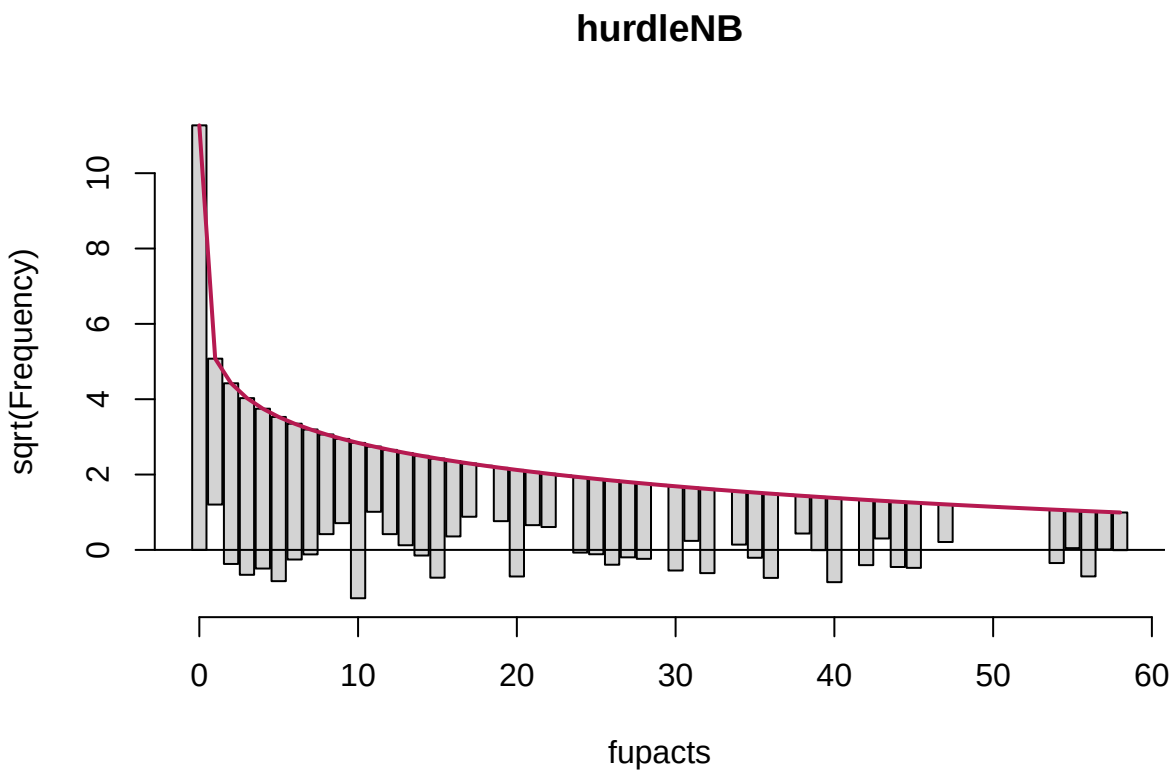
```
##
## Attaching package: 'countreg'
```

```
## The following objects are masked from 'package:pscl':
##
##     hurdle, hurdle.control, hurdletest, zeroinfl, zeroinfl.control
```

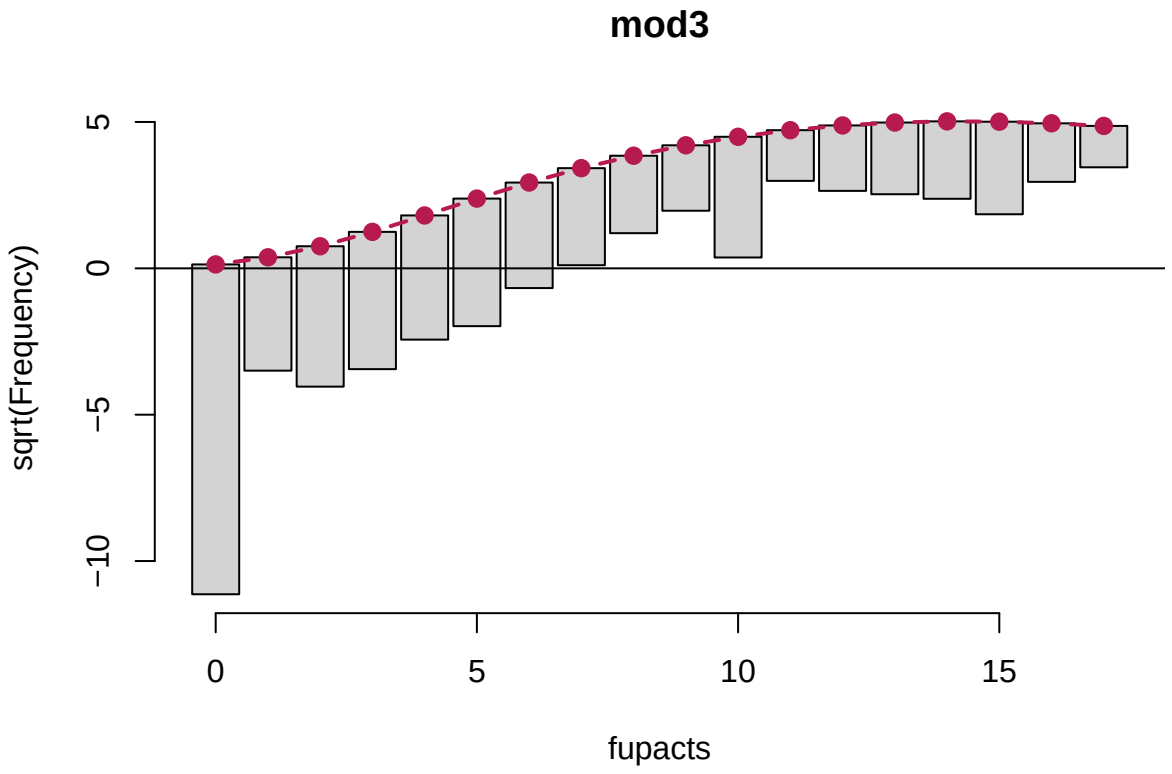
```
rootogram(hurdlePoisson)
```



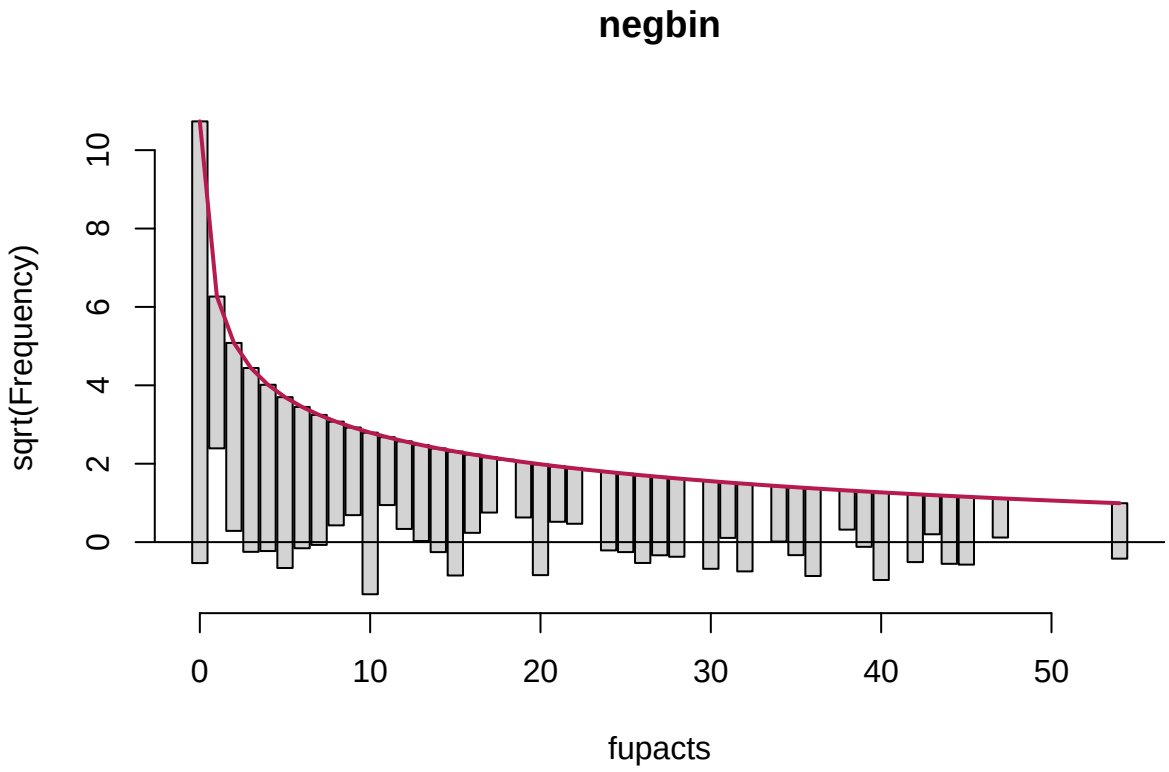

```
rootogram(hurdleNB)
```




```
rootogram(mod3)
```




```
rootogram(negbin)
```




```
AIC(hurdlePoisson)
```

```
## [1] 10002.97
```

```
AIC(hurdleNB)
```

```
## [1] 3012.827
```

```
AIC(negbin)
```

```
## [1] 3035.337
```

```
AIC(mod3)
```

```
## [1] 13924.45
```

I first fit two hurdle models using a poisson and negative binomial distribution. Both follow the binomial in the zero distribution. Based on the AIC of these two hurdle models, the poisson model, and the negative binomial model, is appears that the negative binomial hurdle model provides the best fit of the data. Visually, this conclusion is supported by the rootograms for the models.

5

Assumptions: These data include responses from both men and women from the participating couples. Does this give you any concern?

It may not be reasonable to assume independence, as two observations may come from the same couple.

Pulling Punches

The two `.Rdata` files under week 4 come as an abbreviated version of punch profiles from a boxing system to measure acceleration of boxers during live fights. The `profiles` list from the first file below has each

individual punch profile which has a list item being a 3 column data frame of time (in ms around the middle of the punch event), acceleration in x (forward-back in g's), and acceleration in y (side-to-side in g's). Also attached are some other fields which are of less importance and contribute to this being a somewhat messy data set.

```
load(file = 'punch_profiles.Rdata')
load(file = 'punch_types.Rdata')
```

There are 2135 labeled punch profiles each with a labeled punch type. Use the `punch_types` data frame as ground truth for punch type (labeled 1-6) in addition to the boxers stance (orthodox or southpaw), and punching head (right or left). The punch types are below.

```
###### PUNCH TYPES
#1 - Cross
#2 - Hook
#3 - Jab
#4 - Upper Cut
#5 - Overhand (shouldn't be any of these)
#6 - Unknown (shouldn't be any of these)
```

6

Features: Create at least 10 new features from the punch profiles. They can be combinations of x and y acceleration or individually from either. Explain how these features have been constructed.

Features:

1.) Maximum value of Y 2.) Minimum value of Y 3.) Slope of Y: Max(Y)-Min(Y)/ Time Range 4.) Maximum value of X 5.) Minimum value of X 6.) Slope: Max(X)-Min(Y)/ Time Range 7.) IQR X 8.) IQR Y 9.) Median X 10.) Median Y

First I convert the list of punch profiles into a data frame using the variables defined above.

```
datalist = list()

for(i in 1:nrow(punch_types)){
  row <- c()
  row[1] <- max(profiles[[i]]$profile[,3]) #Max Y
  row[2] <- min(profiles[[i]]$profile[,3]) #Min Y
  row[3] <- (row[1]-row[2])/(which.max( profiles[[i]]$profile[,3])-which.min( profiles[[i]]$profile[,3]
  row[4] <- max(profiles[[i]]$profile[,2]) #Max X
  row[5] <- min(profiles[[i]]$profile[,2]) #Min X
  row[6] <- (row[4]-row[5])/(which.max( profiles[[i]]$profile[,2])-which.min( profiles[[i]]$profile[,2]
  row[7] <- IQR(profiles[[i]]$profile[,3]) #IQR(Y)
  row[8] <- IQR(profiles[[i]]$profile[,2]) #IQR(X)
  row[9] <- median(profiles[[i]]$profile[,3]) #Median(Y)
  row[10] <- median(profiles[[i]]$profile[,2]) #Median(X)
  row[11] <- punch_types$hand[i] #left or right handed
  row[12] <- punch_types$st[i] #stance
  row[13] <- punch_types$pt[i] #Punchtype
  datalist[[i]] <- row
}

profileDF <- as.data.frame(do.call("rbind", datalist))
profileDF$V13 <- as.factor(profileDF$V13)
```

7

Multinomial Model Fit a multinomial model to estimate each of the punch types. Which of the punch types have the most difficulty in being separated?

```
library(nnet)
fit <- multinom(V13 ~., data = profileDF, trace = F)
summary(fit)
```

```
## Call:
## multinom(formula = V13 ~ ., data = profileDF, trace = F)
##
## Coefficients:
##   (Intercept)         V1         V2            V3          V4         V5
## 2 -0.04624763 0.02983983 0.09453967 -0.0005610226  0.09037998 0.06611879
## 3  0.08161592 0.12206914 0.13819255 -0.0558167493 -0.08967976 0.09603534
## 4  2.57211962 0.06402563 0.09925936 -0.0361885245 -0.01032654 0.14414664
##            V6        V7          V8        V9       V10        V11
## 2 -0.04199192 0.3725018 -0.10346418 0.4323742 0.9963085  0.9975063
## 3 -0.06399246 0.5213331 -0.59101671 1.0349830 0.3472058  2.2860905
## 4 -0.22064231 0.4961168 -0.07605391 0.7762033 1.3170475 -0.1179839
##          V12
## 2 -1.3615015
## 3  0.2232095
## 4 -2.3926740
##
## Std. Errors:
##   (Intercept)         V1          V2         V3         V4         V5
## 2   0.6535319 0.01024802 0.007049643 0.02146295 0.01756236 0.01737323
## 3   0.7681026 0.01176907 0.010329566 0.02751629 0.01837450 0.02277668
## 4   1.1525050 0.01711132 0.013812777 0.04316338 0.02501609 0.03674868
##           V6        V7         V8        V9       V10       V11       V12
## 2 0.04134190 0.1129394 0.08862001 0.1279072 0.1586611 0.2051947 0.2199249
## 3 0.05755331 0.1364641 0.10783277 0.1462184 0.1867305 0.2428450 0.2468339
## 4 0.08212537 0.1685577 0.12852143 0.1983083 0.2166848 0.3688897 0.5311020
##
## Residual Deviance: 3032.27
## AIC: 3110.27
```

```
pred <- predict(fit, profileDF, type = "class")
table(profileDF$V13, pred)
```

```
##    pred
##       1   2   3   4
##   1 366 104  19   1
##   2 110 425 205   4
##   3  20 121 677   0
##   4   5  65  13   0
```

```
cat("Accuracy = ", mean(pred == profileDF$V13))
```

```
## Accuracy =  0.6875878
```

This model isn't great – it has a lot of trouble predicting Upper cuts especially.

8

Logistic Regression Consider bucketing the punches into two groups (straights and hooks). Are you able to improve accuracy in any way?

```
profileDF$V14 <- as.factor(ifelse(profileDF$V13 == 1 | profileDF$V13 == 2, 0, 1))
fitbin <- glm(V14~. -V13,data = profileDF, family=binomial)
summary(fitbin)
```

```
##
## Call:
## glm(formula = V14 ~ . - V13, family = binomial, data = profileDF)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2804  -0.5928  -0.1747   0.7261   2.9526
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.867927   0.523996   1.656 0.097649 .
## V1           0.090224   0.007183  12.561  < 2e-16 ***
## V2           0.061104   0.007092   8.615  < 2e-16 ***
## V3          -0.050621   0.019006  -2.663 0.007735 **
## V4          -0.137784   0.011844 -11.633  < 2e-16 ***
## V5           0.048909   0.016484   2.967 0.003006 **
## V6          -0.070654   0.042631  -1.657 0.097453 .
## V7           0.182494   0.085800   2.127 0.033422 *
## V8          -0.363651   0.067427  -5.393 6.92e-08 ***
## V9           0.527289   0.088016   5.991 2.09e-09 ***
## V10         -0.192268   0.107083  -1.796 0.072572 .
## V11          0.856925   0.152988   5.601 2.13e-08 ***
## V12          0.650823   0.169996   3.828 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2907.6  on 2134  degrees of freedom
## Residual deviance: 1797.3  on 2122  degrees of freedom
## AIC: 1823.3
##
## Number of Fisher Scoring iterations: 5
```

```
pred <- predict(fitbin, profileDF, type = "response")
pred <- ifelse(pred>=.5,1,0)
cat("Accuracy = ", mean(pred == profileDF$V14))
```

```
## Accuracy =  0.8023419
```

We can see that accuracy increases to around 80%.