

MATH523 Assignment 4

Kabilan Sriranjani

March 29, 2018

```
library(aod)
library(VGAM)
```

```
## Loading required package: stats4
## Loading required package: splines
library(nnet)
```

A12

a)

```
#A12
#A12
game <- 1:24
attempts <- c(4,9,11,6,6,7,7,1,8,9,5,5,5,4,7,3,7,2,11,8,4,4,5,7)
shots <- c(0,7,4,3,5,2,3,0,1,6,0,2,0,2,5,1,3,0,8,0,0,0,2,2)

model.rayallen <- glm(cbind(shots,attempts-shots) ~ 1, family=binomial)
summary(model.rayallen)
```

```
##
## Call:
## glm(formula = cbind(shots, attempts - shots) ~ 1, family = binomial)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7946  -1.7282  -0.1721   0.4872   2.3952
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.4633     0.1706  -2.716   0.0066 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 58.307  on 23  degrees of freedom
## Residual deviance: 58.307  on 23  degrees of freedom
## AIC: 96.384
##
## Number of Fisher Scoring iterations: 4
pchisq(deviance(model.rayallen), df=23, lower.tail = FALSE)

## [1] 6.687648e-05
```

We get a significant p-value from the deviance test, which suggests that the model is not adequate. Perhaps we need to consider the dispersion.

b)

```
model.rayallen.quasi <- glm(cbind(shots, attempts-shots) ~ 1, family=quasibinomial)
summary(model.rayallen.quasi)
```

```
##
## Call:
## glm(formula = cbind(shots, attempts - shots) ~ 1, family = quasibinomial)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7946  -1.7282  -0.1721   0.4872   2.3952
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4633      0.2434  -1.904   0.0696 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 2.035939)
##
##      Null deviance: 58.307  on 23  degrees of freedom
## Residual deviance: 58.307  on 23  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

The estimate for β is the same. This is because we have only scaled the score equations, thus they still have the same solutions.

c)

```
rayallen.data <- data.frame(shots, attempts-shots)
model.rayallen.quasi2 <- quasibin(cbind(shots, attempts-shots) ~ 1, data=rayallen.data, link = c("logit")
model.rayallen.quasi2@fm$coefficients
```

```
## (Intercept)
##  -0.5981343
```

Here the estimate is indeed different from the previous two models. This is because we are changing the likelihood beyond scaling so we changed our solution to the score equations.

d)

```
model.rayallen.betabin <- vglm(cbind(shots, attempts-shots) ~ 1, betabinomial(zero=2, irho=0.2))
```

```
## Warning in checkwz(wz, M, trace = trace, wzepsilon = control$wzepsilon): 1
## diagonal elements of the working weights variable 'wz' have been replaced
## by 1.819e-12

## Warning in checkwz(wz, M = M, trace = trace, wzepsilon = control
## $wzepsilon): 1 diagonal elements of the working weights variable 'wz' have
## been replaced by 1.819e-12

## Warning in checkwz(wz, M = M, trace = trace, wzepsilon = control
## $wzepsilon): 1 diagonal elements of the working weights variable 'wz' have
## been replaced by 1.819e-12

## Warning in checkwz(wz, M = M, trace = trace, wzepsilon = control
## $wzepsilon): 1 diagonal elements of the working weights variable 'wz' have
## been replaced by 1.819e-12

## Warning in checkwz(wz, M = M, trace = trace, wzepsilon = control
## $wzepsilon): 1 diagonal elements of the working weights variable 'wz' have
## been replaced by 1.819e-12

summary(model.rayallen.betabin)

##
## Call:
## vglm(formula = cbind(shots, attempts - shots) ~ 1, family = betabinomial(zero = 2,
##   irho = 0.2))
##
##
## Pearson residuals:
##           Min       1Q   Median       3Q      Max
## logit(mu) -1.536 -1.0346 0.10048 0.5447 1.620
## logit(rho) -1.078 -0.8538 0.04236 0.8579 1.387
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -0.6224     0.2544  -2.446   0.0144 *
## (Intercept):2  -1.3460     0.5385  -2.500   0.0124 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors: 2
##
## Names of linear predictors: logit(mu), logit(rho)
##
## Log-likelihood: -41.6041 on 46 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
```

We again get different estimates as we are changing the likelihood beyond scaling.

e)

```
beta.rayallen <- summary(model.rayallen)$coefficients[,1]
stderr.rayallen <- summary(model.rayallen)$coefficients[,2]
beta.quasi <- summary(model.rayallen.quasi)$coefficients[,1]
stderr.quasi <- summary(model.rayallen.quasi)$coefficients[,2]
beta.quasi2 <- model.rayallen.quasi2$fm$coefficients
stderr.quasi2 <- sqrt(vcov(model.rayallen.quasi2))
beta.betabin <- coefficients(summary(model.rayallen.betabin))[,1][1]
stderr.betabin <- coefficients(summary(model.rayallen.betabin))[,2][1]

rayallen.ci <- c(beta.rayallen - qnorm(0.975)*stderr.rayallen, beta.rayallen + qnorm(0.975)*stderr.rayallen)
quasi.ci <- c(beta.quasi - qnorm(0.975)*stderr.quasi, beta.quasi + qnorm(0.975)*stderr.quasi)
quasi2.ci <- c(beta.quasi2 - qnorm(0.975)*stderr.quasi2, beta.quasi2 + qnorm(0.975)*stderr.quasi2)
betabin.ci <- c(beta.betabin - qnorm(0.975)*stderr.betabin, beta.betabin + qnorm(0.975)*stderr.betabin)

rayallen.ci

## [1] -0.7975898 -0.1289795
quasi.ci

## [1] -0.94029250 0.01372314
quasi2.ci

## [1] -1.0953114 -0.1009573
betabin.ci

## (Intercept):1 (Intercept):1
## -1.1209739 -0.1237375
```

Here we have all the confidence intervals for β in every model. β represents the log odds of Ray Allen making a shot, taking into account that he usually either has a very good success rate or very bad one.

f)

A variable that could affect his success rate in a game would be the opposing team. It may be harder to get a 3-point shot against some teams, and maybe certain teams cause him to be more nervous and perform worse. The betabinomial model has an extra parameter so it can account for the fact that there is some randomness to whether he has a high success rate or a low one.

A13

a)

```
#A13
I <- c(14, 483, 497, 1008)
notI <- c(1105, 411111, 4624, 157342)
S <- c("Seat belt", "Seat belt", "None", "None")
E <- c("Yes", "No", "Yes", "No")
```

```

model.fatal <- glm(cbind(I, notI) ~ S+E, family=binomial)
#likelihood test
pchisq(deviance(model.fatal), df=1, lower.tail=FALSE)

## [1] 0.09114565

#pearson chi square test
pchisq(sum(residuals(model.fatal, type="pearson")^2), df=1, lower.tail=FALSE)

## [1] 0.1098935

```

The p-values we get using both tests are not significant, which means that our model is adequate.

b)

```

summary(model.fatal)

##
## Call:
## glm(formula = cbind(I, notI) ~ S + E, family = binomial)
##
## Deviance Residuals:
##      1      2      3      4
## -1.6132  0.3142  0.3256 -0.2165
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.04362    0.03120 -161.65  <2e-16 ***
## SSeat belt  -1.71732    0.05402  -31.79  <2e-16 ***
## EYes         2.79779    0.05526   50.63  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 3567.723  on 3  degrees of freedom
## Residual deviance:  2.854  on 1  degrees of freedom
## AIC: 38.039
##
## Number of Fisher Scoring iterations: 3

beta <- coef(model.fatal)
odds.ratio <- exp(beta)
stdErr <- summary(model.fatal)$coefficients[,2]

lowerCI <- exp(beta-qnorm(0.975)*stdErr)
higherCI <- exp(beta+qnorm(0.975)*stdErr)
#Odds ratio and confidence interval for Seatbelt
odds.ratio[2]

## SSeat belt
## 0.1795466

```

```

lowerCI[2]

## SSeat belt
## 0.1615099
higherCI[2]

## SSeat belt
## 0.1995975
#Odds ratio and confidence interval for Ejected
odds.ratio[3]

## EYes
## 16.40842
lowerCI[3]

## EYes
## 14.72424
higherCI[3]

## EYes
## 18.28524

```

The odds ratio, r , for Seatbelt can be obtained by

$$\begin{aligned}
 r &= \frac{e^{\beta_0 + \beta_1(1) + \beta_2 x_{i2}}}{e^{\beta_0 + \beta_1(0) + \beta_2 x_{i2}}} \\
 &= e^{\beta_1}
 \end{aligned}$$

And we can do the same for β_2 to get the odds ratio for Ejected. The confidence intervals are obtained through the formula $\hat{\beta} \pm se(\hat{\beta})z_{0.025}$

c)

```

model.fatal2 <- glm(cbind(I, notI) ~ E, family=binomial)
beta2 <- coef(model.fatal2)
odds.ratio2 <- exp(beta2)
stdErr2 <- summary(model.fatal2)$coefficients[,2]
lowerCI2 <- exp(beta2 - qnorm(0.975)*stdErr2)
higherCI2 <- exp(beta2 + qnorm(0.975)*stdErr2)

#Odds and confidence interval
odds.ratio2[2]

## EYes
## 34.00627
lowerCI2[2]

## EYes
## 30.65391

```

```
higherCI2[2]
```

```
##      EYes  
## 37.72525
```

We compute the odds ratio and confidence interval for the new model the exact same way as before.

A14

a)

```
#A14  
race <- c(1, 1, 0, 0)  
gender <- c(1, 0, 1, 0)  
heaven.yes <- c(88, 54, 397, 235)  
heaven.mb <- c(16, 7, 141, 189)  
heaven.no <- c(2, 5, 24, 39)  
  
model.heaven <- multinom(cbind(heaven.yes, heaven.mb, heaven.no) ~ race+gender)
```

```
## # weights: 12 (6 variable)  
## initial value 1315.038910  
## iter 10 value 931.782315  
## final value 930.880924  
## converged
```

```
summary(model.heaven)
```

```
## Call:  
## multinom(formula = cbind(heaven.yes, heaven.mb, heaven.no) ~  
##      race + gender)  
##  
## Coefficients:  
##      (Intercept)      race      gender  
## heaven.mb -0.2633509 -1.1486396 -0.725226  
## heaven.no -1.7942192 -0.6721255 -1.034177  
##  
## Std. Errors:  
##      (Intercept)      race      gender  
## heaven.mb  0.09588936 0.2367592 0.1318292  
## heaven.no  0.16750483 0.4113510 0.2586682  
##  
## Residual Deviance: 1861.762  
## AIC: 1873.762
```

$\log(\pi_{i1}/\pi_{i3})$ is given by

$$\alpha_1 + \beta_1^G x_{i1} + \beta_1^R x_{i2}$$

$\log(\pi_{i2}/\pi_{i3})$ is given by

$$\alpha_2 + \beta_2^G x_{i1} + \beta_2^R x_{i2}$$

$\log(\pi_{i1}/\pi_{i2})$ is given by

$$\log(\pi_{i1}/\pi_{i3}) - \log(\pi_{i2}/\pi_{i3})$$

b)

```
model.heavenB <- glm(cbind(heaven.yes, heaven.no) ~ race+gender, family=binomial)
summary(model.heavenB)
```

```
##
## Call:
## glm(formula = cbind(heaven.yes, heaven.no) ~ race + gender, family = binomial)
##
## Deviance Residuals:
##      1       2       3       4
## 0.3442 -0.2510 -0.1100  0.0895
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.7806     0.1684  10.573 < 2e-16 ***
## race           0.7183     0.4119   1.744  0.0812 .
## gender        1.0485     0.2588   4.051 5.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 20.81612  on 3  degrees of freedom
## Residual deviance:  0.20156  on 1  degrees of freedom
## AIC: 22.502
##
## Number of Fisher Scoring iterations: 4
```

```
beta.gender <- coef(model.heavenB)[3]
stderr.gender <- summary(model.heavenB)$coefficients[,2][3]
gender.oddsratio <- exp(beta.gender)
lowerCI.gender <- exp(beta.gender-qnrm(0.975)*stderr.gender)
higherCI.gender <- exp(beta.gender+qnrm(0.975)*stderr.gender)
#Odds ratio and confidence interval
gender.oddsratio
```

```
## gender
## 2.853366
```

```
lowerCI.gender
```

```
## gender
## 1.718175
```

```
higherCI.gender
```

```
## gender
## 4.738571
```

From looking at the odds ratio it appears that males have roughly triple the odds of not believing in heaven than females.

c)

```
model.heavenRace <- glm(cbind(heaven.yes, heaven.no) ~ race, family=binomial)
summary(model.heavenRace)
```

```
##
## Call:
## glm(formula = cbind(heaven.yes, heaven.no) ~ race, family = binomial)
##
## Deviance Residuals:
##      1       2       3       4
##  1.233  -1.238   2.565  -2.773
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3058     0.1321  17.452  <2e-16 ***
## race          0.7042     0.4091   1.721   0.0852 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 20.816  on 3  degrees of freedom
## Residual deviance: 17.322  on 2  degrees of freedom
## AIC: 37.622
##
## Number of Fisher Scoring iterations: 4
```

```
anova(model.heavenB, model.heavenRace, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(heaven.yes, heaven.no) ~ race + gender
## Model 2: cbind(heaven.yes, heaven.no) ~ race
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1          1      0.2016
## 2          2     17.3220 -1    -17.12 3.508e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can check if gender has an effect on the response when race is controlled by comparing the model with both race and gender to the model with only race. If the test is significant that that means the nested model is not an adequate simplification and so gender does indeed affect the response. Since we do get a significant p-value then that means belief in heaven is not independent of gender when race is controlled for.

A15

If the binary response is the result of an underlying latent variable, this suggests that after some certain threshold, T , the response changes from 0 to 1. If both groups have similar location then they are both close to T . But because one has significantly higher dispersion than the other, it will have a wide confidence interval that overlaps with the other side of T . This makes it hard to fit a model because if a response is 0 we can't tell if it is truly 0 or actually 1 but the dispersion makes it fall on the wrong side of T .