Heidi's data

Robert Bagchi

February 9, 2016

```
## first read in data
fish <- read.csv("../data/StatsLunch_HGolden_02_09_2016.csv")
summary(fish)
```

```
##       X.1               X              Year         antenna
##  Min.   :   1.0   Min.   :  113   Min.   :2010   GCL    :599
##  1st Qu.: 460.8   1st Qu.: 6402   1st Qu.:2011   Kup2   :307
##  Median : 920.5   Median : 8994   Median :2011   Kup4   :304
##  Mean   : 920.5   Mean   : 9954   Mean   :2011   Kup3   :212
##  3rd Qu.:1380.2   3rd Qu.:12393   3rd Qu.:2012   Kup5   :196
##  Max.   :1840.0   Max.   :23646   Max.   :2013   Kup4_5 :115
##                                                  (Other):107
##       tag                 date        early_trap     M.lake0.river1
##  Min.   :157361335   9/8/11 : 201   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:174440033   9/12/11: 115   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :174441148   9/13/11:  85   Median :0.0000   Median :1.0000
##  Mean   :172513528   9/9/11 :  71   Mean   :0.2174   Mean   :0.5821
##  3rd Qu.:174476635   7/28/11:  65   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :177350994   9/14/11:  56   Max.   :1.0000   Max.   :1.0000
##                      (Other):1247
##   R.spr0.fall1
##  Min.   :0.0000
##  1st Qu.:1.0000
##  Median :1.0000
##  Mean   :0.7652
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

```
head(fish)
```

```
##   X.1   X Year antenna       tag    date early_trap M.lake0.river1
## 1   1 113 2010     GCL 174475948 5/24/10          0              0
## 2   2 161 2010     GCL 174475996 5/24/10          0              0
## 3   3 174 2010     GCL 174476009 5/24/10          0              0
## 4   4 232 2010     GCL 174476068 5/24/10          0              0
## 5   5 305 2010     GCL 174476142 5/24/10          0              0
## 6   6 445 2010     GCL 174475851 5/25/10          0              0
##   R.spr0.fall1
## 1            0
## 2            0
## 3            0
## 4            0
## 5            0
## 6            0
```

```
## couple of weird columns not in Heidi's dataset - removing

fish <- fish[, -c(1:2)]
head(fish)
```

```
##   Year antenna       tag    date early_trap M.lake0.river1 R.spr0.fall1
## 1 2010     GCL 174475948 5/24/10          0              0            0
## 2 2010     GCL 174475996 5/24/10          0              0            0
## 3 2010     GCL 174476009 5/24/10          0              0            0
## 4 2010     GCL 174476068 5/24/10          0              0            0
## 5 2010     GCL 174476142 5/24/10          0              0            0
## 6 2010     GCL 174475851 5/25/10          0              0            0
```

```
summary(fish)
```

```
##       Year         antenna          tag                 date
##  Min.   :2010   GCL    :599   Min.   :157361335   9/8/11 : 201
##  1st Qu.:2011   Kup2   :307   1st Qu.:174440033   9/12/11: 115
##  Median :2011   Kup4   :304   Median :174441148   9/13/11:  85
##  Mean   :2011   Kup3   :212   Mean   :172513528   9/9/11 :  71
##  3rd Qu.:2012   Kup5   :196   3rd Qu.:174476635   7/28/11:  65
##  Max.   :2013   Kup4_5 :115   Max.   :177350994   9/14/11:  56
##                 (Other):107                       (Other):1247
##    early_trap     M.lake0.river1    R.spr0.fall1
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000
##  Median :0.0000   Median :1.0000   Median :1.0000
##  Mean   :0.2174   Mean   :0.5821   Mean   :0.7652
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##
```

```
table(fish$antenna)
```

```
##
##    GCL   Kup2   Kup3   Kup4 Kup4_5   Kup5   Kup6   Kup7    KUS
##    599    307    212    304    115    196     16      5     86
```

Reading through Heidi's emails and from talking to her, I think the aim here is to model whether or not an individual migrated is affected by whether it was trapped. We can get at this quite easily with a binomial glm.

```
## Fit a model
mod.mig <- glm(M.lake0.river1 ~ early_trap, data=fish, family=binomial)
```

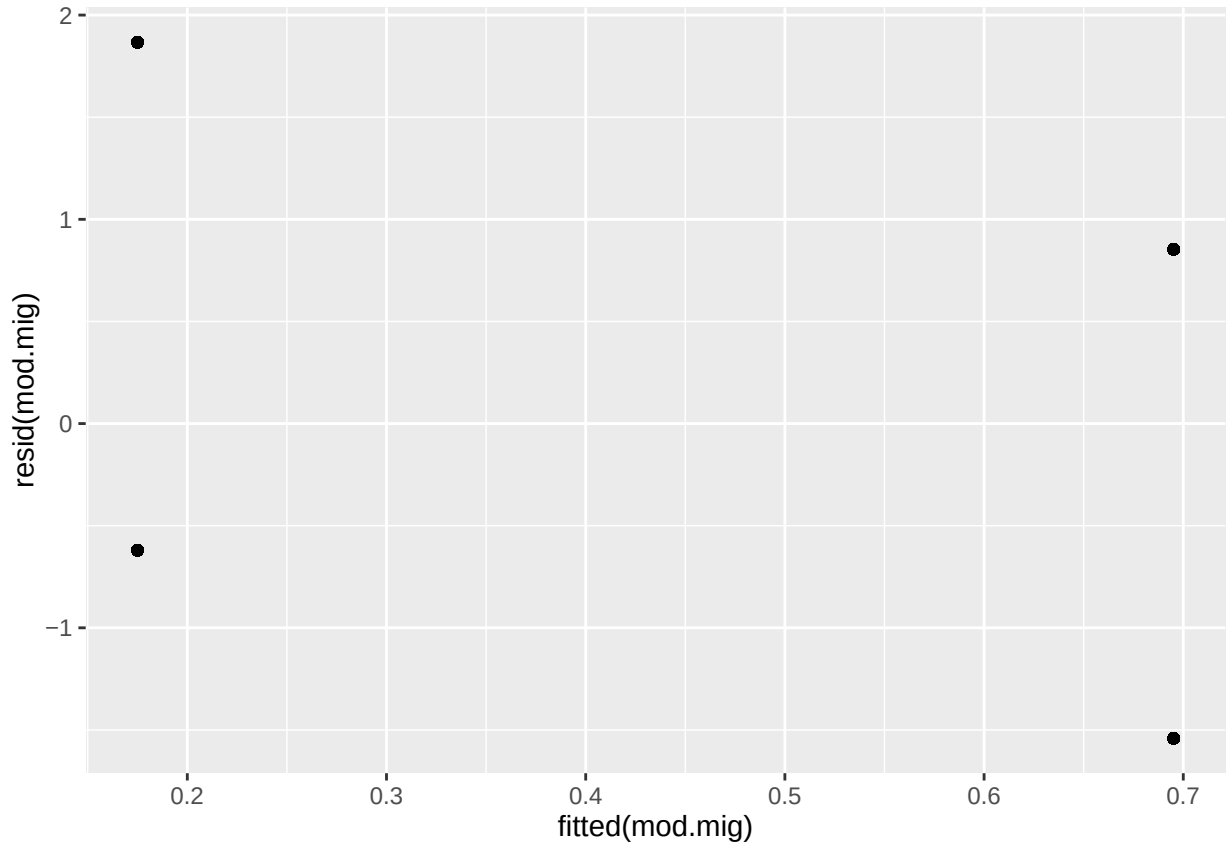
So before interpreting, let us check some model diagnostics

```
## overdispersion
phi <- sum(resid(mod.mig, type='pearson')^2)/df.residual(mod.mig)
phi ## approx 1, which is good, but binary data are not that good for testing this
```

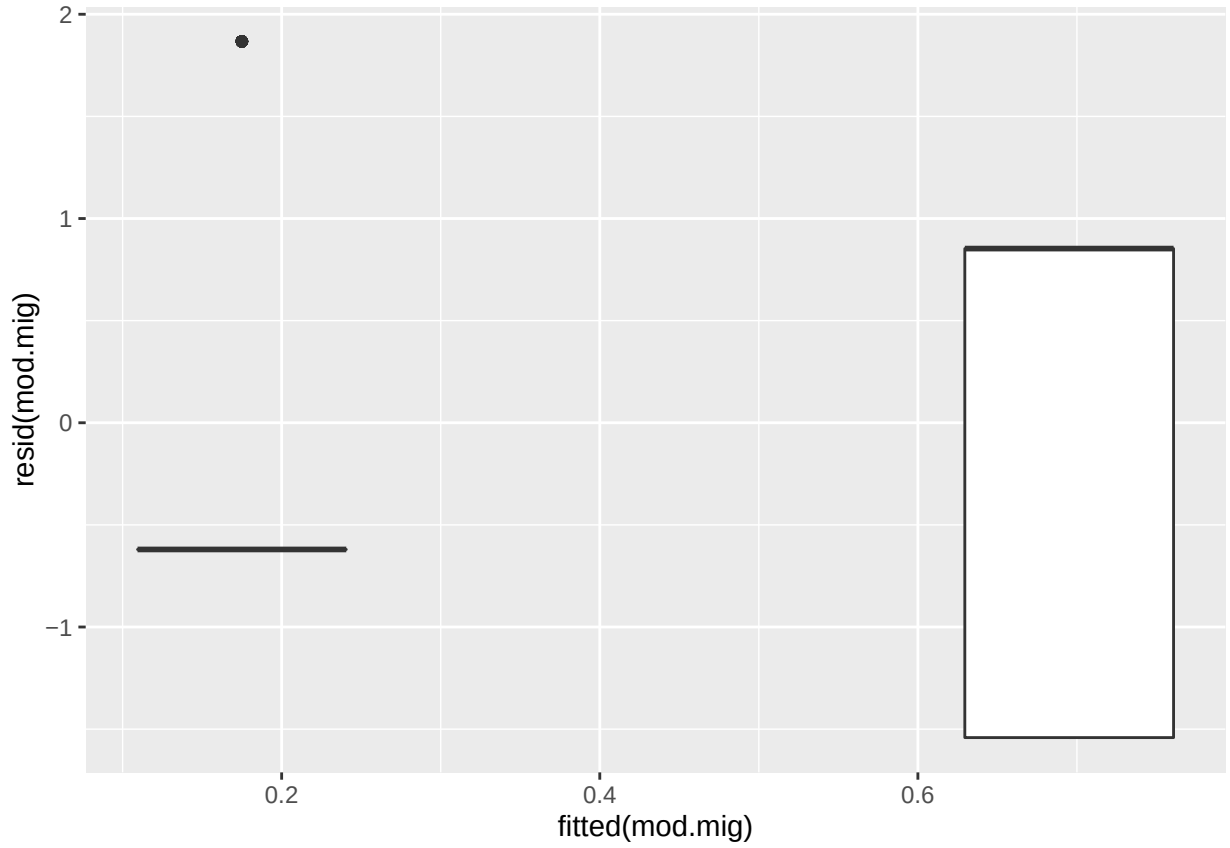
```
## [1] 1.001088
```

```
## Another test is whether the residuals show a trend against fitted values
## Use a smoother to deal with discrete nature of binary values
library(ggplot2)
qplot(x=fitted(mod.mig), y=resid(mod.mig), geom=c('point', 'smooth'))
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```




```
## actually this highlights that because we have a binary predictor, we only have
## 2 values on the x axis - so need to be a little smarter
qplot(x=fitted(mod.mig), y=resid(mod.mig), group=fitted(mod.mig),
      geom=c('boxplot')) ## Certainly more variation in the ones that migrated
```




```
## than the ones that didn't, but difficult to be sure.
## AUC is a useful statistic with binary data
library(verification)
```

```
## Loading required package: fields
```

```
## Loading required package: spam
```

```
## Loading required package: grid
```

```
## Spam version 1.3-0 (2015-10-24) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps

##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990. #
##  # Type '?world' or 'news(package="maps")'. See README_v3. #

## Loading required package: boot

## Loading required package: CircStats

## Loading required package: MASS

## Loading required package: dtw

## Loading required package: proxy

##
## Attaching package: 'proxy'

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in publication.
```

```
roc.area(obs=mod.mig$y, pred=fitted(mod.mig))
```

```
## $A
## [1] 0.6818846
##
## $n.total
## [1] 1840
##
## $n.events
## [1] 1071
##
## $n.noevents
## [1] 769
##
## $p.value
## [1] 5.905462e-78
```

```
## AUC is 0.68 which isn't great, but is actually not that far from usual with these sorts of
##data
```

We could go on, but probably not efficient to do so. Let's try and examine the outputs.

```
summary(mod.mig)
```

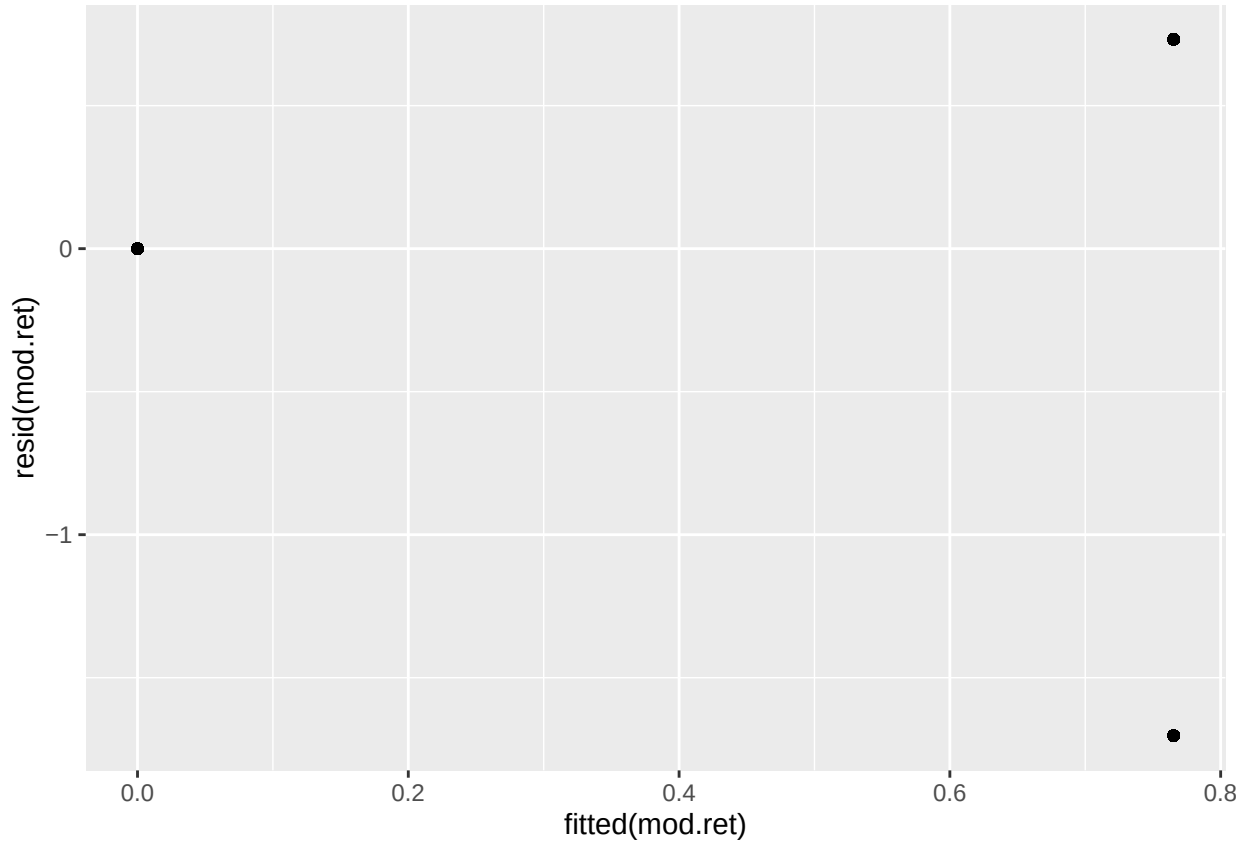
```
##
## Call:
## glm(formula = M.lake0.river1 ~ early_trap, family = binomial,
##     data = fish)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5414  -0.6203   0.8528   0.8528   1.8671
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.82426    0.05724   14.40   <2e-16 ***
## early_trap  -2.37485    0.14350  -16.55   <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: 2501  on 1839  degrees of freedom
## Residual deviance: 2142  on 1838  degrees of freedom
## AIC: 2146
##
## Number of Fisher Scoring iterations: 4
```

Seems pretty clear that trapped individuals are less likely to migrate.

So now onto the next question - given they have migrated, what is the effect on their probability of returning? In this case there is no need to look at the ones that didn't migrate, so subset the data for this analysis.

```
mod.ret <- glm(R.spr0.fall1 ~ early_trap,data=fish, subset=M.lake0.river1==1,
               family=binomial)

qplot(x=fitted(mod.ret), y=resid(mod.ret), ##group=fitted(mod.ret),
      geom=c('point')) ## hmm, few 0s in the fitted.
```




```
summary(subset(fish, M.lake0.river1==1))
```

```
##       Year         antenna         tag                date
##  Min.   :2010   Kup2   :307   Min.   :157361335   7/28/11: 64
##  1st Qu.:2011   Kup4   :304   1st Qu.:174439628   9/7/11 : 45
##  Median :2011   Kup3   :212   Median :174441221   8/4/11 : 44
##  Mean   :2011   Kup5   :130   Mean   :171960815   9/8/11 : 44
##  3rd Qu.:2011   Kup4_5 : 84   3rd Qu.:174476388   7/30/11: 40
##  Max.   :2013   Kup6   : 16   Max.   :177350987   9/9/11 : 40
##                 (Other): 18                       (Other):794
##    early_trap      M.lake0.river1  R.spr0.fall1
##  Min.   :0.00000   Min.   :1      Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:1      1st Qu.:0.0000
##  Median :0.00000   Median :1      Median :1.0000
##  Mean   :0.06536   Mean   :1      Mean   :0.7152
##  3rd Qu.:0.00000   3rd Qu.:1      3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1      Max.   :1.0000
##
```

```
## so very few (6%) of returning fish were early trap
fish.mig <- subset(fish, M.lake0.river1 ==1)
with(fish.mig, table(early_trap, R.spr0.fall1)) ## none returned if trapped
```

```
##           R.spr0.fall1
```

```
## early_trap   0   1
##          0 235 766
##          1  70   0
```

```
summary(mod.ret) ## and there is an associated drop in return probability
```

```
##
## Call:
## glm(formula = R.spr0.fall1 ~ early_trap, family = binomial, data = fish,
##     subset = M.lake0.river1 == 1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.70245  -0.00022   0.73154   0.73154   0.73154
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.18160    0.07457   15.85   <2e-16 ***
## early_trap  -18.74767  472.85400   -0.04    0.968
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1279.7  on 1070  degrees of freedom
## Residual deviance: 1091.0  on 1069  degrees of freedom
## AIC: 1095
##
## Number of Fisher Scoring iterations: 16
```

```
## but huge standard errors - the Hauck-Donner effect.
```

The Hauck Donner effect is really irritating - basically, if you have a category where the predicted response is exactly 0 or 1 (i.e. no or total success) then glms give you huge standard errors.

There are some solutions out there - try googling Hauck-Donner - perhaps for another day.

Basically we have a situation where the overall response, will an individual migrate and return given it was trapped, is a two part problem. We have broken it down into the components migrate or not and given it migrated did it return or not. But can we predict what the effect of being trapped on the overall probability?

The answer is yes - very simply we have $Pr(migrate\&return|trapped) = Pr(migrate, trapped) \cdot Pr(return|migrate, trapped)$. But can we get confidence intervals on that? Once again yes, but we will have to leave that for another time.