ST2304 Exercises Week 13: Binomial Models

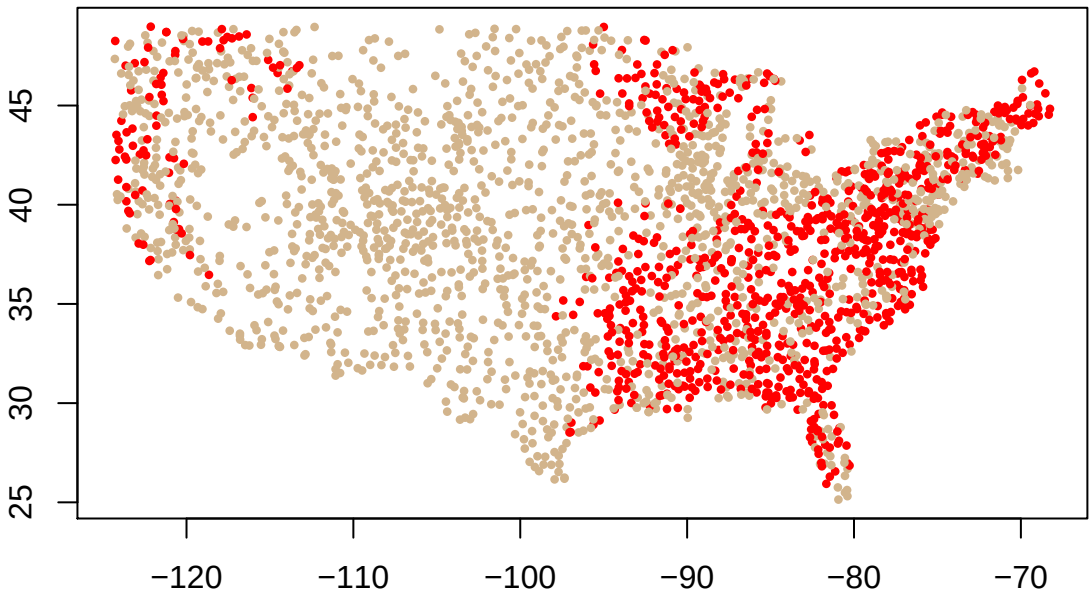
Bob O'Hara

14 March 2018

Fitting Binomial GLMs

Fitting binomials is almost the same as the Poisson GLM. First we will read in the data for poleated woodpecker, from the lecture, and then plot it on a map. The data is for all of the North American BBS routes from 2010. For each route, observers stop 50 times, mnad at each stop record the birds they see. So we have 50 trials, at each one we have preesence or absence of the bird: in this case the pileated woodpecker.

```
# as always, your file path may differ
DryoPil <- read.csv(file="../Data/Dpileatus.csv")
# Scale the covariates
DryoPil$prec.mean.sc <- as.vector(scale(DryoPil$prec.mean))
DryoPil$temp.mean.sc <- as.vector(scale(DryoPil$temp.mean))

# Plot the data
plot(DryoPil$long, DryoPil$lat, col=c("tan", "red")[1 + (DryoPil$NPres>0)],
     pch=16, cex=0.6, ann=FALSE)
```



We can start by looking at presence/absence at the route level, i.e. whether the bird was seen on at least one stop. We define `present` as a logical node, and convert it to a factor, `PresentF`. Strictly we don't need to make it a factor: in this case R will convert the logical `DryoPil$Present` to a factor. But doing that can be dangerous because the next time you forget to convert something to a factor, R might not do it the way you want it to (there is a whole section of the 8th circle of R Hell devoted to these problems: https://www.burns-stat.com/pages/Tutor/R_inferno.pdf).

Once we have presence as a factor, we can fit the model. This is really no different to previous models, other than we specify a different link function. If we want to be complete, we can specify the logit link (it is the default, so if we don't, that's what R will use):

```
DryoPil$Present <- DryoPil$NPres>0
DryoPil$PresentF <- factor(DryoPil$Present)

mod.method1 <- glm(PresentF ~ temp.mean.sc, data=DryoPil,
                family="binomial")
mod.methodcomplete <- glm(PresentF ~ temp.mean.sc, data=DryoPil,
                family=binomial("logit"))
```

With only single trials, the factor method is probably the easiest to use. But when we have several trials (and particularly when that number varies), the success/failure way of writing is easier. For example, just using the presence at the route level:

```
DryoPil$Absent <- 1-DryoPil$Present
mod.method2 <- glm(cbind(Present, Absent) ~ temp.mean.sc,
                data=DryoPil, family="binomial")
```

Both methods give the same results. Here are the parameters:

```
summary(mod.method1)
```

```
Call:
glm(formula = PresentF ~ temp.mean.sc, family = "binomial", data = DryoPil)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2928  -0.9326  -0.8215   1.2970   1.7255

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.58674    0.04185 -14.020   <2e-16 ***
temp.mean.sc  0.35688    0.04196   8.506   <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: 3360.9  on 2568  degrees of freedom
Residual deviance: 3286.7  on 2567  degrees of freedom
AIC: 3290.7

Number of Fisher Scoring iterations: 4
```

So we can see the positive effect of mean temperature in this model. We know from the lecture that this isn't the best model, so we can fit a bigger model and use analysis of deviance (with anova()) to compare the models. I have done it twice here, changing the order of the covariates:

```
mod.big1 <- glm(Present ~ prec.mean.sc + temp.mean.sc + I(prec.mean.sc^2) + I(temp.mean.sc^2),
                data=DryoPil, family="binomial")
mod.big2 <- glm(Present ~ temp.mean.sc + I(temp.mean.sc^2) + prec.mean.sc + I(prec.mean.sc^2),
                data=DryoPil, family="binomial")

anova(mod.big1, test="Chisq")
```

```
Analysis of Deviance Table
```

```
Model: binomial, link: logit

Response: Present

Terms added sequentially (first to last)

                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                2568     3360.9
prec.mean.sc      1   698.13      2567     2662.7 < 2.2e-16 ***
temp.mean.sc      1    22.96      2566     2639.8 1.651e-06 ***
I(prec.mean.sc^2) 1   111.15      2565     2528.6 < 2.2e-16 ***
I(temp.mean.sc^2) 1     2.87      2564     2525.8   0.09035 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod.big2, test="Chisq")
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: Present

Terms added sequentially (first to last)

                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                                2568     3360.9
temp.mean.sc      1    74.19      2567     3286.7  < 2e-16 ***
I(temp.mean.sc^2) 1     4.78      2566     3281.9  0.02875 *
prec.mean.sc      1   642.35      2565     2639.6  < 2e-16 ***
I(prec.mean.sc^2) 1   113.80      2564     2525.8  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lovers of the strict guideline that we should include variablers when the p-value is less than 0.05 will have a problem here: the deviance of the quadratic term for temperature is either side of that line, depending on the order of the variables. I decided to drop it in the lecture, because there is a lot of data, and the deviance is always much smaller than for the other terms. The reason for this variation is the (weak) correlation between temperature and precipitation of 0.42.

So, what do the parameters look like in the model with the quadratic term?

```
summary(mod.big1)
```

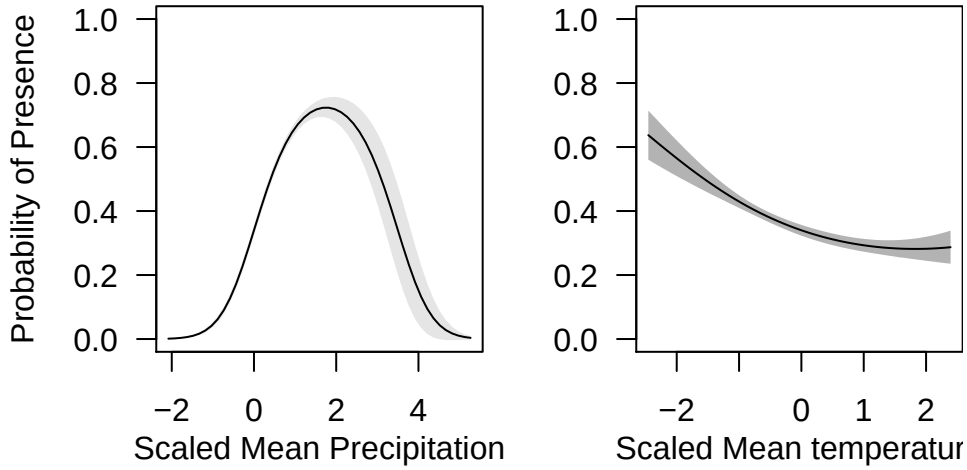
```
Call:
glm(formula = Present ~ prec.mean.sc + temp.mean.sc + I(prec.mean.sc^2) +
    I(temp.mean.sc^2), family = "binomial", data = DryoPil)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8360  -0.9121  -0.2282   0.9772   2.9818

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.66427    0.07508  -8.847  < 2e-16 ***
```

```
prec.mean.sc       1.85690    0.09956  18.651  < 2e-16 ***
temp.mean.sc      -0.29876    0.06342  -4.711 2.47e-06 ***
I(prec.mean.sc^2) -0.53045    0.06181  -8.583  < 2e-16 ***
I(temp.mean.sc^2)  0.08186    0.04825   1.697   0.0898 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3360.9  on 2568  degrees of freedom
Residual deviance: 2525.8  on 2564  degrees of freedom
AIC: 2535.8

Number of Fisher Scoring iterations: 6
```

The model has curved effects for both: for precipitation the curve is negative, so there is a maximum somewhere. Temperature goes the other way: there is a minimum. When we plot the curves and look at them, we see that over the range of most of the data the probability of presence increases, but for really wet areas the probability is lowered. In oter wrds the woodpecker prefers wetter areas, as long as they are not too wet. Most of the really wet areas are near Seattle (use the commented out code to see this).

```
# Function to calculate the predictions
GetPreds <- function(data, model) {
  pred <- predict(model, newdata = data, type="response", se.fit=TRUE)

  pred$lower <- pred$fit - pred$se.fit
  pred$upper <- pred$fit + pred$se.fit
  cbind(data, pred)
}
# Make data frames to predict onto: these get passed into the function
PredData.prec <- data.frame(prec.mean.sc = seq(min(DryoPil$prec.mean.sc),
                                               max(DryoPil$prec.mean.sc), length=50),
                            temp.mean.sc = 0)
PredData.temp <- data.frame(temp.mean.sc = seq(min(DryoPil$temp.mean.sc),
                                              max(DryoPil$temp.mean.sc), length=50),
                            prec.mean.sc = 0)

pred.prec.big <- GetPreds(data=PredData.prec, model=mod.big1)
pred.temp.big <- GetPreds(data=PredData.temp, model=mod.big1)

# Plot the predictions
par(mfrow=c(1,2), mar=c(3,4,1,0), oma=c(2,0,0,0))
# hist(DryoPil$prec.mean.sc, xlab="", main="Mean Precipitation")
# hist(DryoPil$temp.mean.sc, xlab="", main="Mean Temperature")
plot(pred.prec.big$prec.mean.sc, pred.prec.big$fit, type="n", las=1, ylim=c(0,1),
     xlab="", ylab="Probability of Presence")
mtext("Scaled Mean Precipitation", 1, line=2)
polygon(c(pred.prec.big$prec.mean.sc, rev(pred.prec.big$prec.mean.sc)),
        c(pred.prec.big$lower, rev(pred.prec.big$upper)),
        col="grey90", border=NA)
lines(pred.prec.big$prec.mean.sc, pred.prec.big$fit)

plot(pred.temp.big$temp.mean.sc, pred.temp.big$fit, type="n", las=1, ylim=c(0,1),
     xlab="", ylab="")
polygon(c(pred.temp.big$temp.mean.sc, rev(pred.temp.big$temp.mean.sc)),
```

```
        c(pred.temp.big$lower, rev(pred.temp.big$upper)),
        col="grey70", border=NA)
lines(pred.temp.big$temp.mean.sc, pred.temp.big$fit)
mtext("Scaled Mean temperature", 1, line=2)
```




```
# Helpful...
# plot(DryoPil$long, DryoPil$lat, pch=16, cex=1, ann=FALSE,
#     col=c("grey90", "red")[1 + (DryoPil$prec.mean.sc>2)])
```

Question 1

For the same data, look at a model where you model the probability that a pileated woodpecker is observed at a specific stop (i.e. one of the 50 stops per route). You can modify the following code, as well as modifying the code above.

```
# number of absences
DryoPil$NAbs <- DryoPil$Ntrials-DryoPil$NPres
mod.A <- glm(cbind(NPres, NAbs) ~ prec.mean.sc,
               data=DryoPil, family="binomial")
# To test and calculate over-dispersion
# Because the model is too simple, the effect of unmeasured covariates goes into the over-dispersion
pchisq(deviance(mod.A), df.residual(mod.A), lower.tail = FALSE)
Disp <- deviance(mod.A)/df.residual(mod.A)
# The next line is commented out to make the document shorter.
# summary(mod.A, dispersion=Disp)
```

Fit the model with linear and quadratic terms, as above. Use `anova()` to compare the models (or use AICif you want to try that!)

- would you include the precipitation and temperature in the model?
- would you include quadratic terms for either or both of these models?
- Look at the parameter estimates, using `summary()`. Qualitatively, are there any differences between the models? Obviously the parameters will be different, but are there any large differences, and can you explain why? (if you can't work this out from just the summary, try plotting the predictions).
- Is there any evidence of over-dispersion in the model? If there is, does it change the model you would chose, and if so, how?

```
Call:
```

```
glm(formula = cbind(NPres, NAbs) ~ prec.mean.sc + temp.mean.sc +
    I(prec.mean.sc^2) + I(temp.mean.sc^2), family = "binomial",
    data = DryoPil)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1691  -1.1728  -0.4530  -0.0191   9.5033

Coefficients:
                  Estimate Std. Error  z value Pr(>|z|)
(Intercept)       -4.18503    0.03627 -115.379   <2e-16 ***
prec.mean.sc       1.37197    0.05220   26.282   <2e-16 ***
temp.mean.sc      -0.05328    0.02754   -1.935    0.053 .
I(prec.mean.sc^2) -0.45227    0.03214  -14.070   <2e-16 ***
I(temp.mean.sc^2)  0.03216    0.02134    1.507    0.132
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6321.0  on 2568  degrees of freedom
Residual deviance: 4574.5  on 2564  degrees of freedom
AIC: 6979.7

Number of Fisher Scoring iterations: 6
```

```
Analysis of Deviance Table

Model: binomial, link: cloglog

Response: cbind(NPres, NAbs)

Terms added sequentially (first to last)

                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                2568     6321.0
prec.mean.sc       1  1221.28      2567     5099.7 < 2.2e-16 ***
temp.mean.sc       1    25.47      2566     5074.2 4.482e-07 ***
I(prec.mean.sc^2)  1   496.97      2565     4577.3 < 2.2e-16 ***
I(temp.mean.sc^2)  1     2.12      2564     4575.2     0.145
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Call:
glm(formula = cbind(NPres, NAbs) ~ prec.mean.sc + temp.mean.sc +
    I(prec.mean.sc^2) + I(temp.mean.sc^2), family = binomial("cloglog"),
    data = DryoPil)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1665  -1.1717  -0.4538  -0.0185   9.5038

Coefficients:
                  Estimate Std. Error  z value Pr(>|z|)
```

```
(Intercept)       -4.19300    0.03589 -116.841   <2e-16 ***
prec.mean.sc       1.36042    0.05182   26.254   <2e-16 ***
temp.mean.sc      -0.05119    0.02717   -1.884   0.0595 .
I(prec.mean.sc^2) -0.45055    0.03188  -14.132   <2e-16 ***
I(temp.mean.sc^2)  0.03084    0.02103    1.466   0.1426
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6321.0  on 2568  degrees of freedom
Residual deviance: 4575.2  on 2564  degrees of freedom
AIC: 6980.4

Number of Fisher Scoring iterations: 6

[1] 8.078738e-117

Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NPres, NAbs)

Terms added sequentially (first to last)

                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                              2568     6321.0
prec.mean.sc      1  1248.30      2567     5072.7 < 2.2e-16 ***
temp.mean.sc      1    17.64      2566     5055.1  0.001667 **
I(prec.mean.sc^2) 1   478.35      2565     4576.7 < 2.2e-16 ***
I(temp.mean.sc^2) 1     2.24      2564     4574.5  0.262491
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NPres, NAbs)

Terms added sequentially (first to last)

                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                              2568     6321.0
prec.mean.sc      1  1248.30      2567     5072.7 < 2.2e-16 ***
temp.mean.sc      1    17.64      2566     5055.1 2.675e-05 ***
I(prec.mean.sc^2) 1   478.35      2565     4576.7 < 2.2e-16 ***
I(temp.mean.sc^2) 1     2.24      2564     4574.5    0.1345
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem 2: Bumpus' Sparrows

In Exercise 6 you looked at survival of sparrows from the data Bumpus collected, using a horrible model. To sumamrise: Bumpus (or one of his technicians) collected sparrows that had been blown onto the road after a storm. They measured different aspects of body size, and looked at whether the birds survived or died. We want to find out

The Status variable shows whether the bird survived or not: in the csv data file it is a string, which R automatically converts to a factor. This is good news, as R can use a factor in the model:

```
Bumpus <- read.csv("../Data/31396_Bumpus_English_Sparrow_Data.csv", na.strings = "Unknown")
names(Bumpus) <- gsub("\\.+[[:alpha:]]*$", "", names(Bumpus))
Bumpus$Total.Length.sc <- scale(Bumpus$Total.Length)
# using relevel() this means that we estimate the probability of survival
Bumpus$Status <- relevel(Bumpus$Status, ref="Dead")
Bumpus$Survived <- as.numeric(Bumpus$Status=="Alive")
```

Fit a model with Total Length ("Total.Length") and Sex as covariates (just as main effects), and survival as a response.

- What effect does the total length have: do larger or smaller birds survive more??
- What is the difference in the odds of survival of birds that are 5mm different in length?
- Does Sex have an effect on survival?

Add an interaction between Sex and Total Length.

- What is the effect of the interaction? Is the estimated effect of body length stronger or weaker in males (i.e. is the slope of the effect steeper)?
- Use analysis of deviance (i.e. `anova()`) to compare the models with and without an interaction. Would you include an interaction?

(once you have the R output, these questions should act as exam-style questions)