Lecture 13: Binomial Models

Bob O'Hara

bob.ohara@ntnu.no

Before we start. . .

Number of Exercises: `ceiling(N/2)`

This week's exercises don't have to be handed in until after Easter (I will out them up tomorrow)

The Binomial Distribution

We have seen the Binomial distribution a few times, now we'll take it seriously...

We have N tests, of which r are a 'success'. Problem is to find $Pr(r|N)$

- may depend on covariates

Likelihood

The probability of r is

$$Pr(r|N,p) = \frac{N!}{r!(N-r)!}p^r(1-p)^{N-r}$$

So the likelihood is

$$\begin{aligned} l(p|N,r) &= r\log p + (N-r)\log(1-p) + \log N! - \log r! - \log(N-r)! \\ &= r\left(\log p - \log(1-p)\right) + N\log(1-p) + f(N,r) \\ &= r\log\left(\frac{p}{1-p}\right) + N\log(1-p) + f(N,r) \end{aligned}$$

This is a GLM with a logit link function

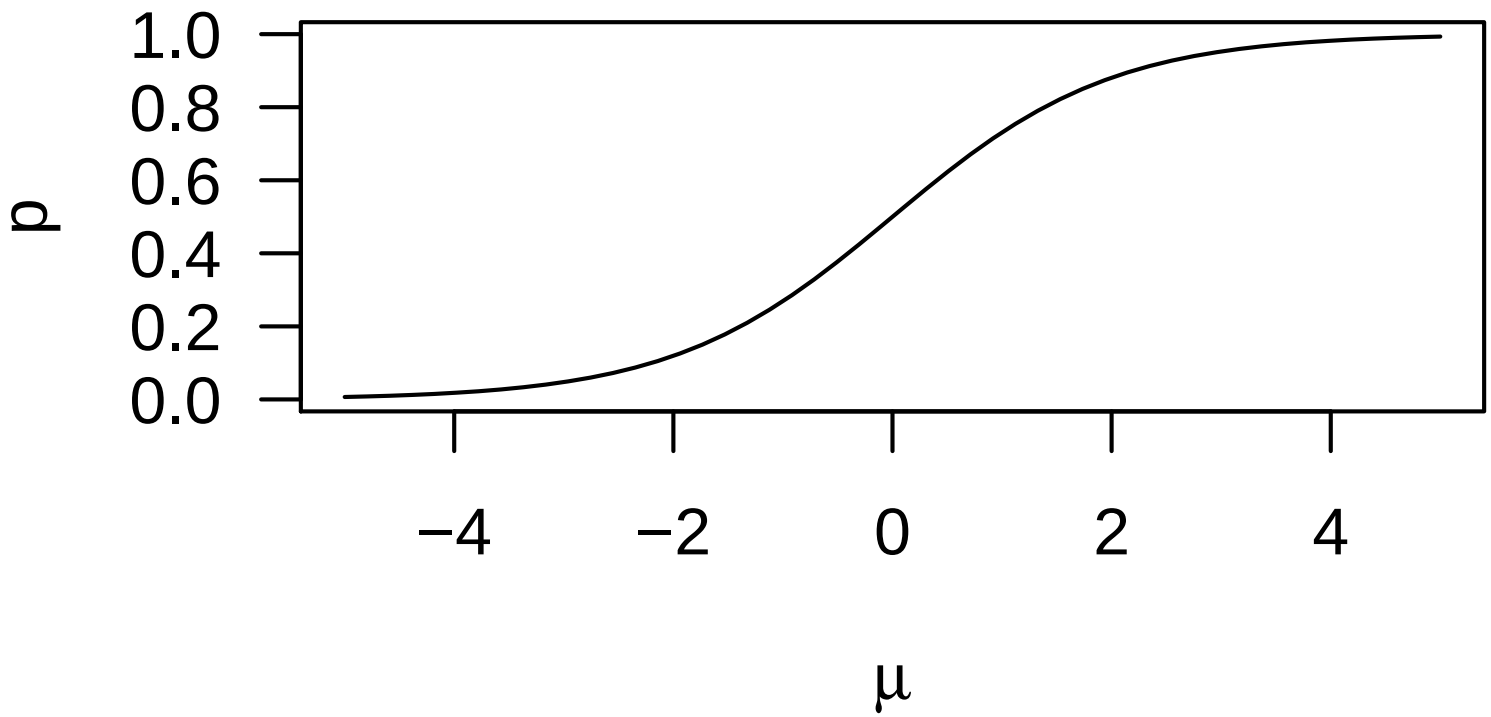
- we will see some other link functions soon

The logit Link

$$\mu = \log \frac{p}{1-p}$$

The inverse is

$$p = \frac{e^{\mu}}{1+e^{\mu}} = \frac{1}{1+e^{-\mu}}$$



Interpreting the logit

The logit is the log-odds

Odds: p/(1-p)

Tiger Roll won the Grand National on Saturday with an odds of 10:1

- if I bet £1, I win £10
- if the odds are fair, this means that for every Grand National won by Tiger Roll, he would lose 10.

The probability is Success/(Success + Failure), i.e.

$$1/(1+10) = 1/11 = 0.09$$

More interpretation of the Logit

if we have a basline effect α (so $p = e^{\alpha}/(1 + e^{\alpha})$) and we increase it by a small β (i.e. $\beta = 0 + \varepsilon$, so $e^{\beta} \approx \varepsilon$) then

$$p = \frac{e^{\alpha+\beta}}{1 + e^{\alpha+\beta}} = \frac{e^{\alpha}e^{\beta}}{1 + e^{\alpha}e^{\beta}} \approx \frac{e^{\alpha}}{1 + e^{\alpha}}e^{\beta} \approx \frac{e^{\alpha}}{1 + e^{\alpha}}(1 + \beta)$$

So a small effect (approximately) adds the effect to the probability

- especially if p is small

More interpretation of the Logit

e.g. if $\mu = -3$,

$$p = \frac{e^{-3}}{(1 + e^{-3})} = 0.047$$

Now let $\beta = 0.2$,

$$p = \frac{e^{-3+0.2}}{(1 + e^{-3+0.2})} = 0.057$$

and $0.047 \times e^{0.2} = 0.047 \times 1.221 = 0.057$

An example

```
## Loading required package: sp
```

The data comne from the North American Breeding Bird survey. We have presence of the Pileated woodpecker (*Dryocopus pileatus*)

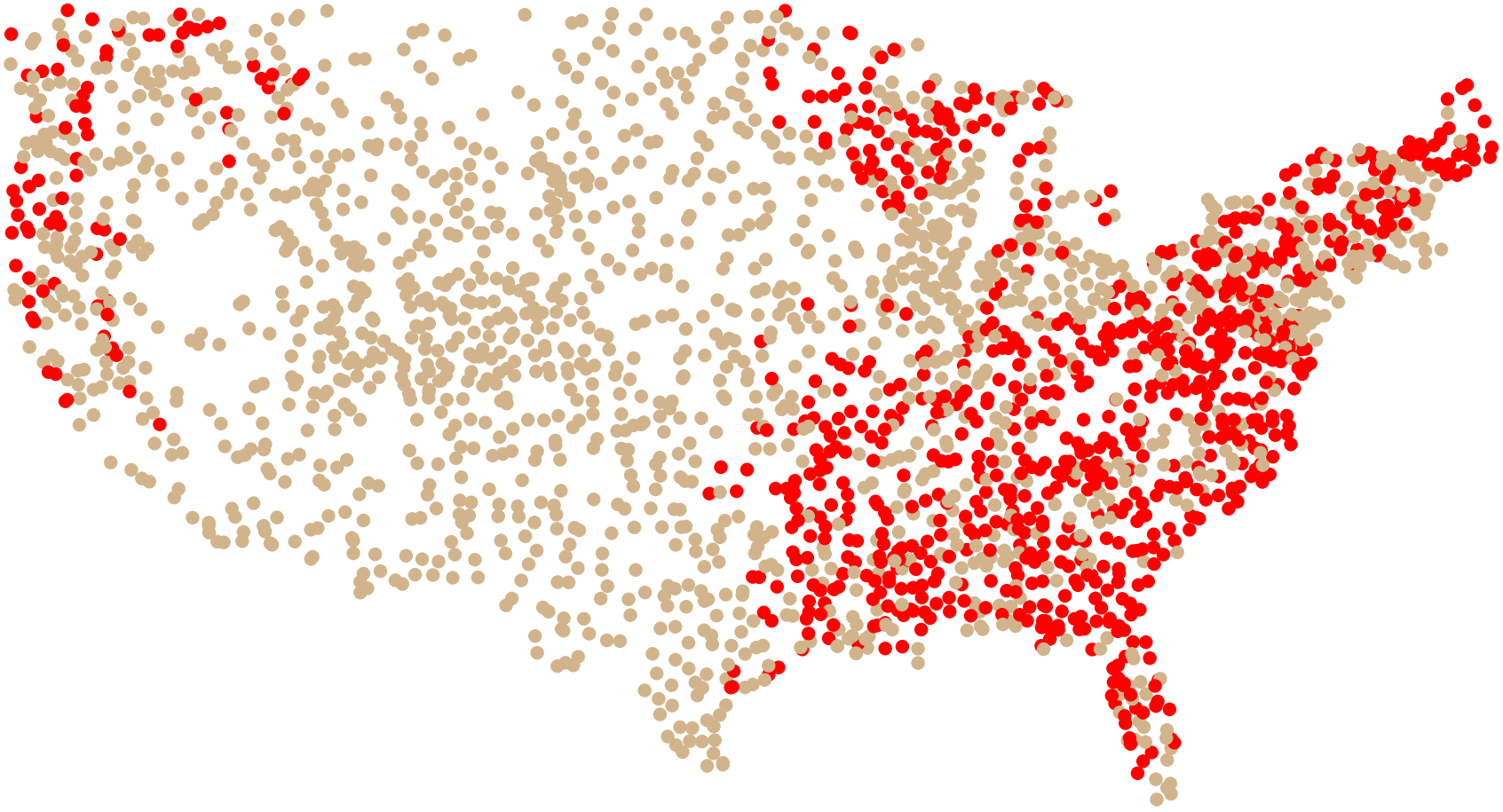
- a toal of 2569 routes, at each one there are 50 stops (so 50 trials)

Can we explain its distribution with rain & temperature?

Figure 1: By Joshlaymon (CC BY-SA 3.0), from Wikimedia Commons

The data

Red: recorded in 2010, grey: not recorded in 2010



Fitting the model

First, we will ignore the 50 trials, and just look at whether the pileated woodpecker was seen at least one on each route

There are several ways to specify the response, depending on the data

- as a factor (first level is failure, rest is success)
- as 2 columns, with successes and failures

Fitting the model: Method 1

As a logical vector

```
DryoPil$Present <- DryoPil$NPres>0
DryoPil$PresentF <- factor(DryoPil$Present)
DryoPil$PresentF[7:11]
```

```
[1] TRUE  FALSE FALSE FALSE TRUE
Levels: FALSE TRUE
```

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

Fitting the model: Method 2

As success and failure columns

```
DryoPil$Absent <- 1-DryoPil$Present
cbind(DryoPil$Present, DryoPil$Absent)[7:11,]
```

```
     [,1] [,2]
[1,]    1    0
[2,]    0    1
[3,]    0    1
[4,]    0    1
[5,]    1    0
```

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

Fitting the model: Method 2

The advantage of this method is that it allows for a varying number of trials

```
DryoPil$NAbs <- DryoPil$Ntrials-DryoPil$NPres
cbind(DryoPil[7:11,c("NPres", "NAbs")])
```

```
   NPres NAbs
7      1   49
8      0   50
9      0   50
10     0   50
11     1   49
```

```
mod.mod.method2all <- glm(cbind(NPres, NAbs) ~ temp.mean.sc
                  data=DryoPil, family="binomial")
```

Model results

Methods 1 & 2 provide identical results

- just different ways of writing the same thing

```
round(summary(mod.method1)$coefficients, 3)
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.587      0.042 -14.020        0
temp.mean.sc    0.357      0.042   8.506        0
```

```
round(summary(mod.method2)$coefficients, 3)
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.587      0.042 -14.020        0
temp.mean.sc    0.357      0.042   8.506        0
```

Model Interpretation: intercept

```
round(summary(mod.method1)$coefficients, 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.587	0.042	-14.020	0
temp.mean.sc	0.357	0.042	8.506	0

Mean temperature is scaled, so at the mean temperature ($11.3^{\circ}C$), the log odds of the species being observed is -0.59. This is the same as a probability of

$$\frac{e^{-0.59}}{1 + e^{-0.59}} = 0.36$$

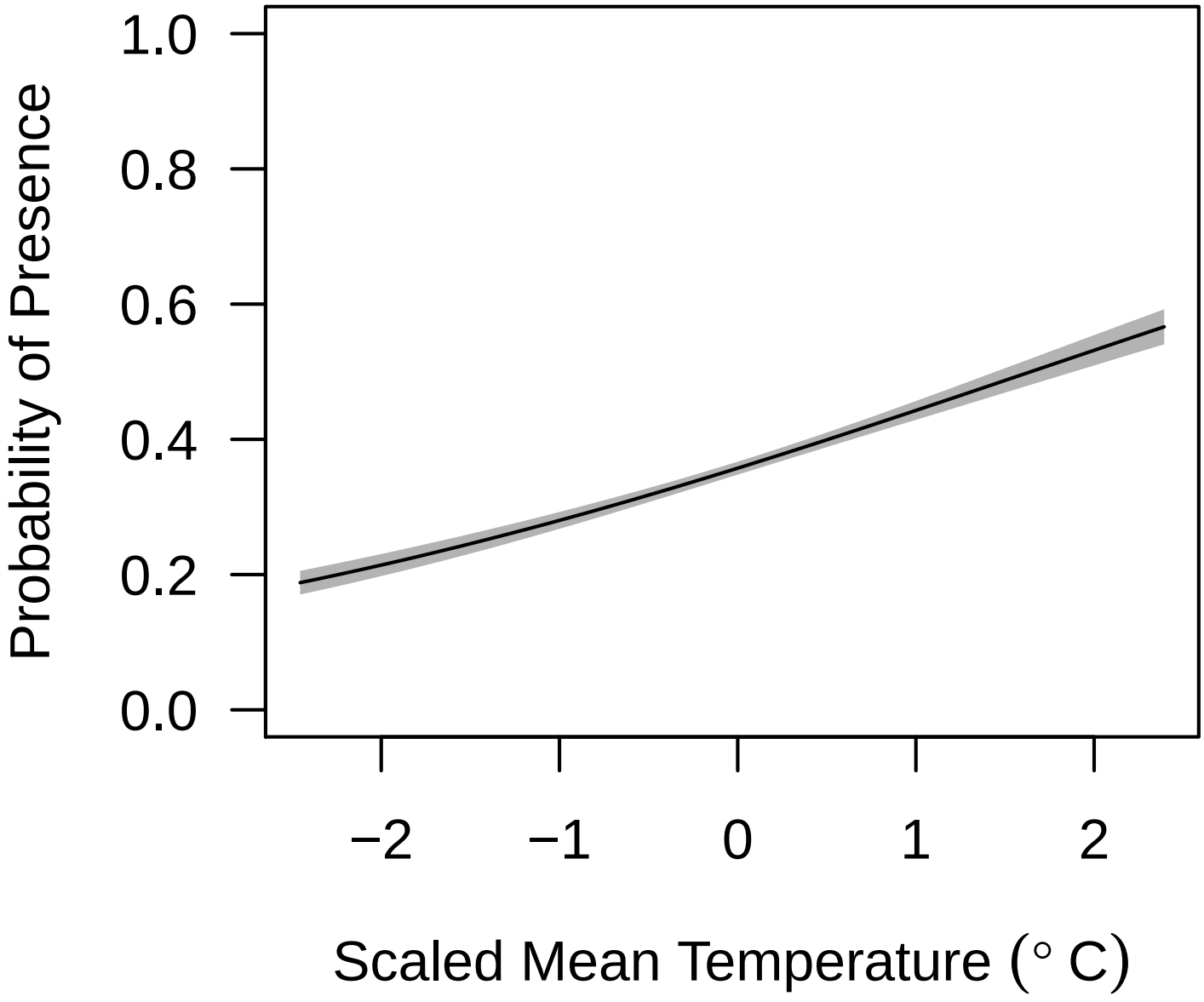
Model Interpretation: slope

if we change the mean temperature by 1 standard deviation ($5.4^\circ C$), the probability of observed presence is

$$\frac{e^{-0.59+0.36\times 1}}{1+e^{-0.59+0.36\times 1}} = 0.44$$

Model Interpretation: Response

We can plot the whole curve over the range of the data:



Analysis of Deviance

Let's fit a bigger model

```
mod.big <- glm(Present ~ prec.mean.sc + temp.mean.sc + I(pr
               data=DryoPil, family="binomial")
an.big <- paste(capture.output(print(anova(mod.big, test="C
cat(an.big[10:17])
```

```
                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi
NULL                                 2568     3360.9
prec.mean.sc         1   698.13      2567     2662.7 < 2.2e-
temp.mean.sc         1    22.96      2566     2639.8 1.651e-
I(prec.mean.sc^2)    1   111.15      2565     2528.6 < 2.2e-
I(temp.mean.sc^2)    1     2.87      2564     2525.8   0.090
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

Suggests a quadratic effect of precipitation (prec.mean.sc) and linear effect of temperature.

A better model

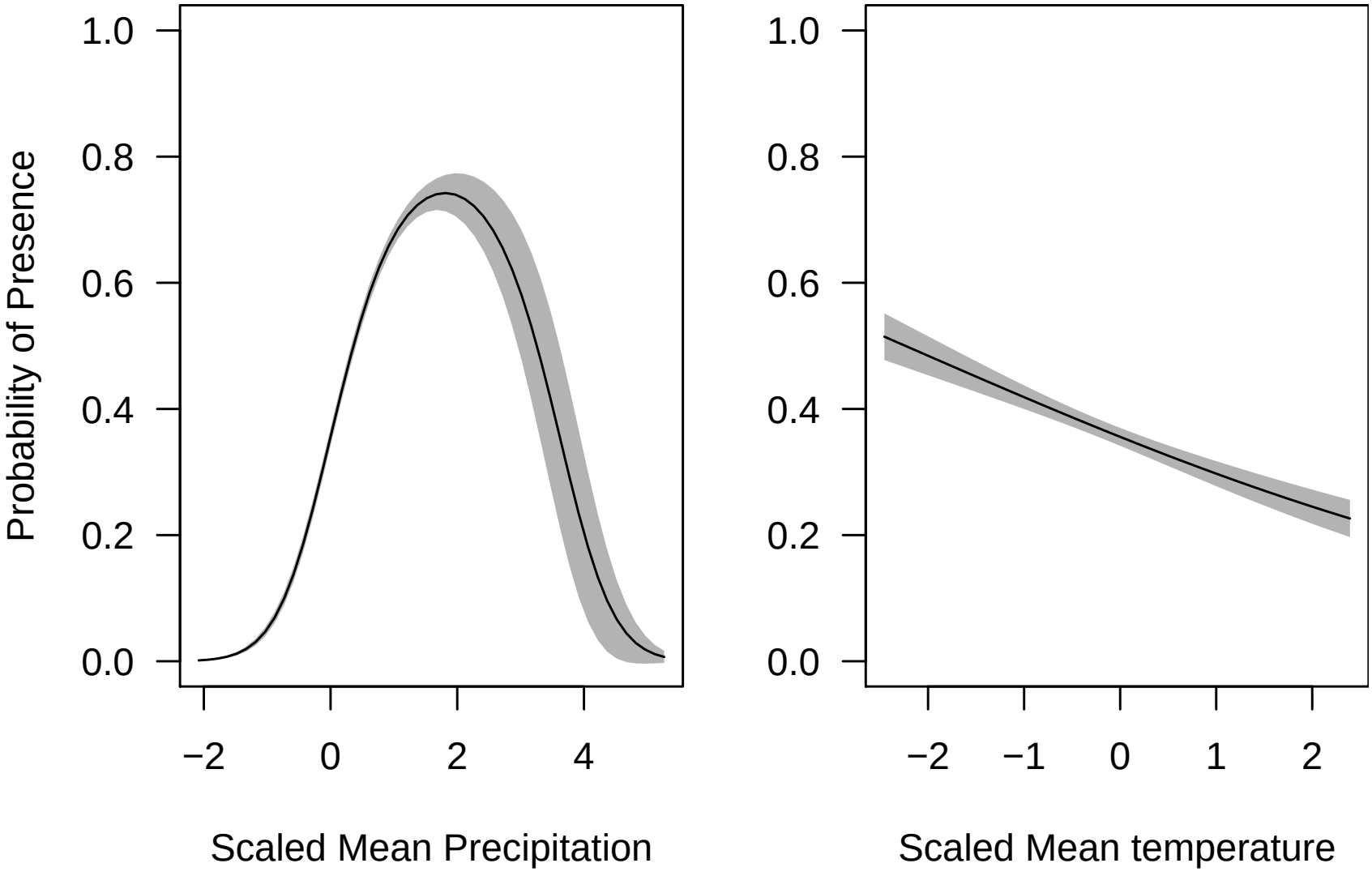
```
mod.better <- glm(Present ~ prec.mean.sc +
                      I(prec.mean.sc^2) + temp.mean.sc,
                  data=DryoPil, family="binomial")
round(summary(mod.better)$coefficients, 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.594	0.062	-9.587	0
prec.mean.sc	1.826	0.096	18.982	0
I(prec.mean.sc^2)	-0.505	0.058	-8.712	0
temp.mean.sc	-0.266	0.060	-4.409	0

The quadratic term for precipitation is negative, so there is a maximum.

But the temperature effect has reversed sign

A better Prediction



The maximum for precipitation is in the range of the data (this is not always the case!)

Link Functions

Unlike the Poisson, the binomial distribution has 3 link functions that are used:

- logit
- probit
- cloglog

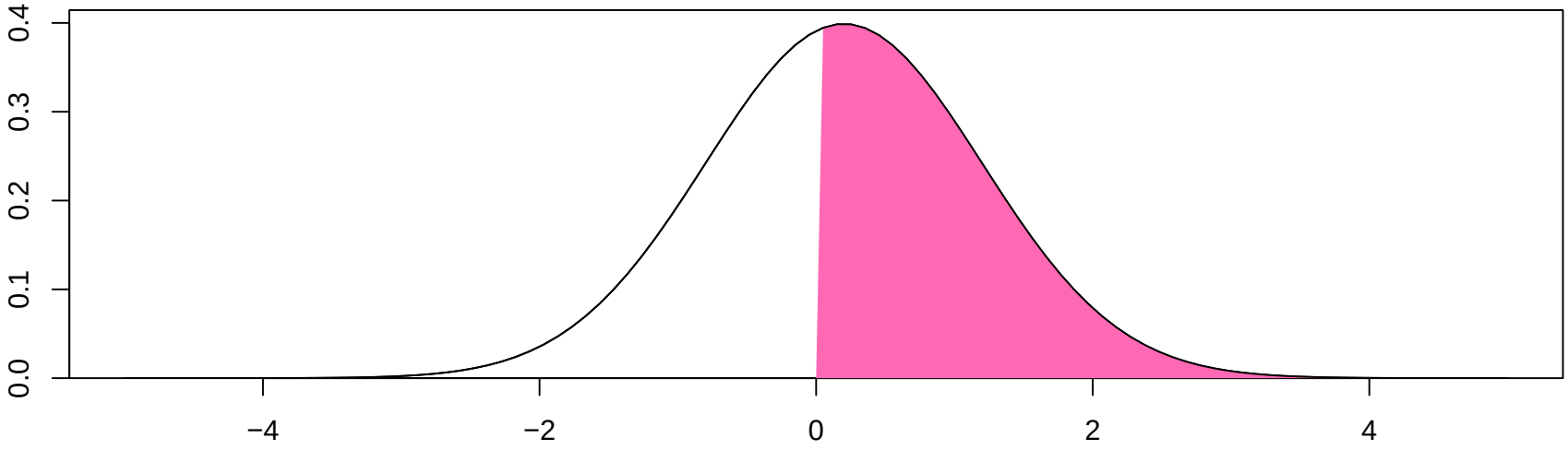
logit

We've seen above that this is the natural link function

- "canonical link"

Probit: a threshold model

Imagine we have a normal distribution



if the distribution is >0 we have a success, if it is < 0 we have a failure

- the large the mean the greater the probability of success
- but still a random chance

This is the same as a probit link

- the inverse normal distribution

cloglog

Last week we looked at the Poisson distribution

Sometimes we have presence/absence for something that is really a count

Dilution assays

We take a sample that might be contaminated by a microorganism

We serially dilute the sample

- concentrations $x, x/2, x/4, x/8, \ldots x/2^n$

Streak out onto agar plates. See if anything grows

If a sample contains the microorganism, it will grow. Assume a Poisson distribution for the organism, then if we had a count of the organism we would model it with a Poisson distribution and log link. But we only have presence/absence.

c the log log

$$p = Pr(growth) = Pr(n > 0) = 1 - Pr(n = 0) = 1 - \frac{\lambda_i^0 e^{-\lambda}}{0!} = 1 - e^{-\lambda}$$

So,

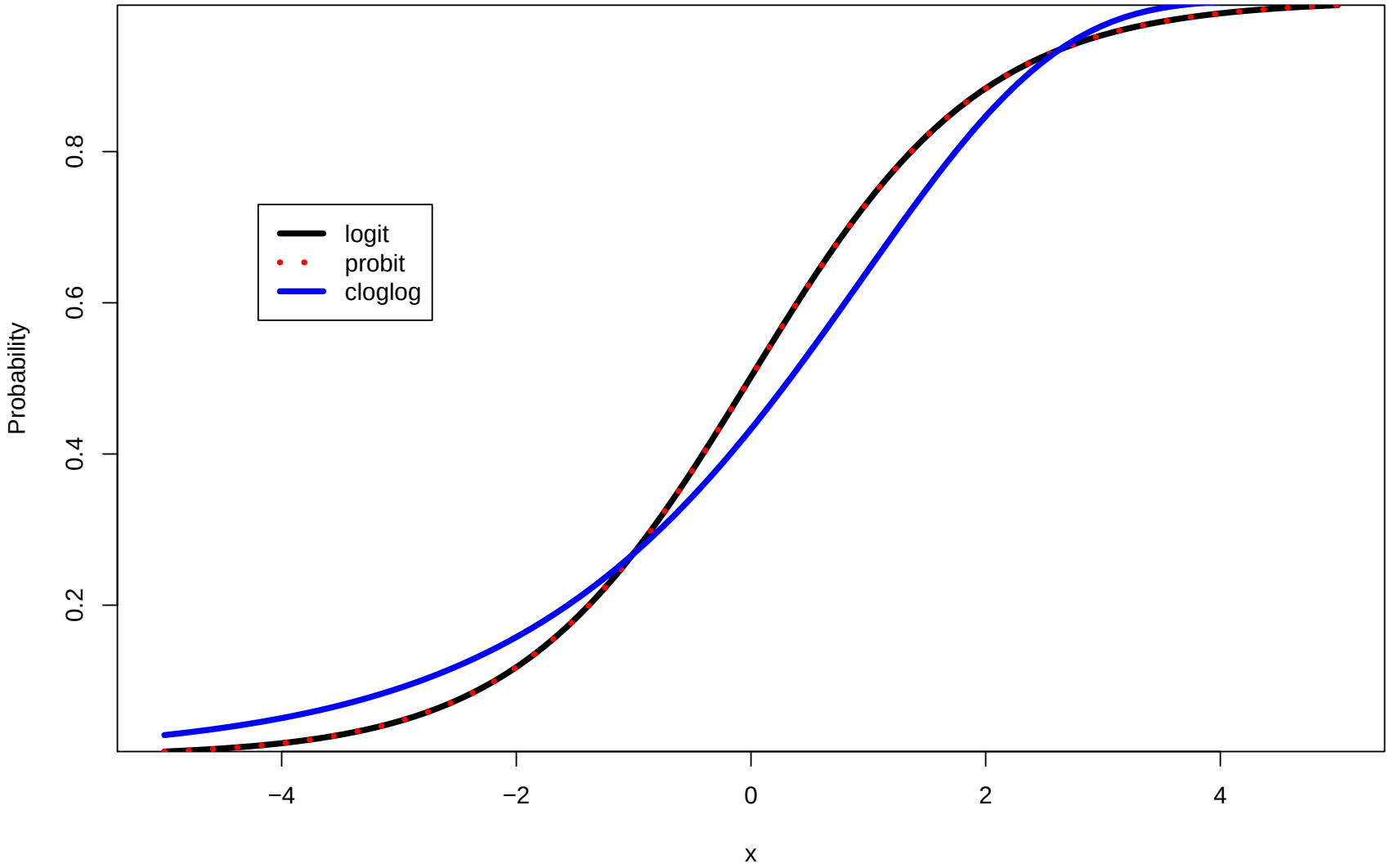
$$\begin{aligned} p &= 1 - e^{-\lambda} \\ e^{-\lambda} &= 1 - p \\ \lambda &= -\log(1 - p) \\ \log(\lambda) &= \log(-\log(1 - p)) \end{aligned}$$

and $\log(-\log(1 - p))$ is the cloglog link function.

The link functions

logit & probit almost the same

cloglog asymmetrical

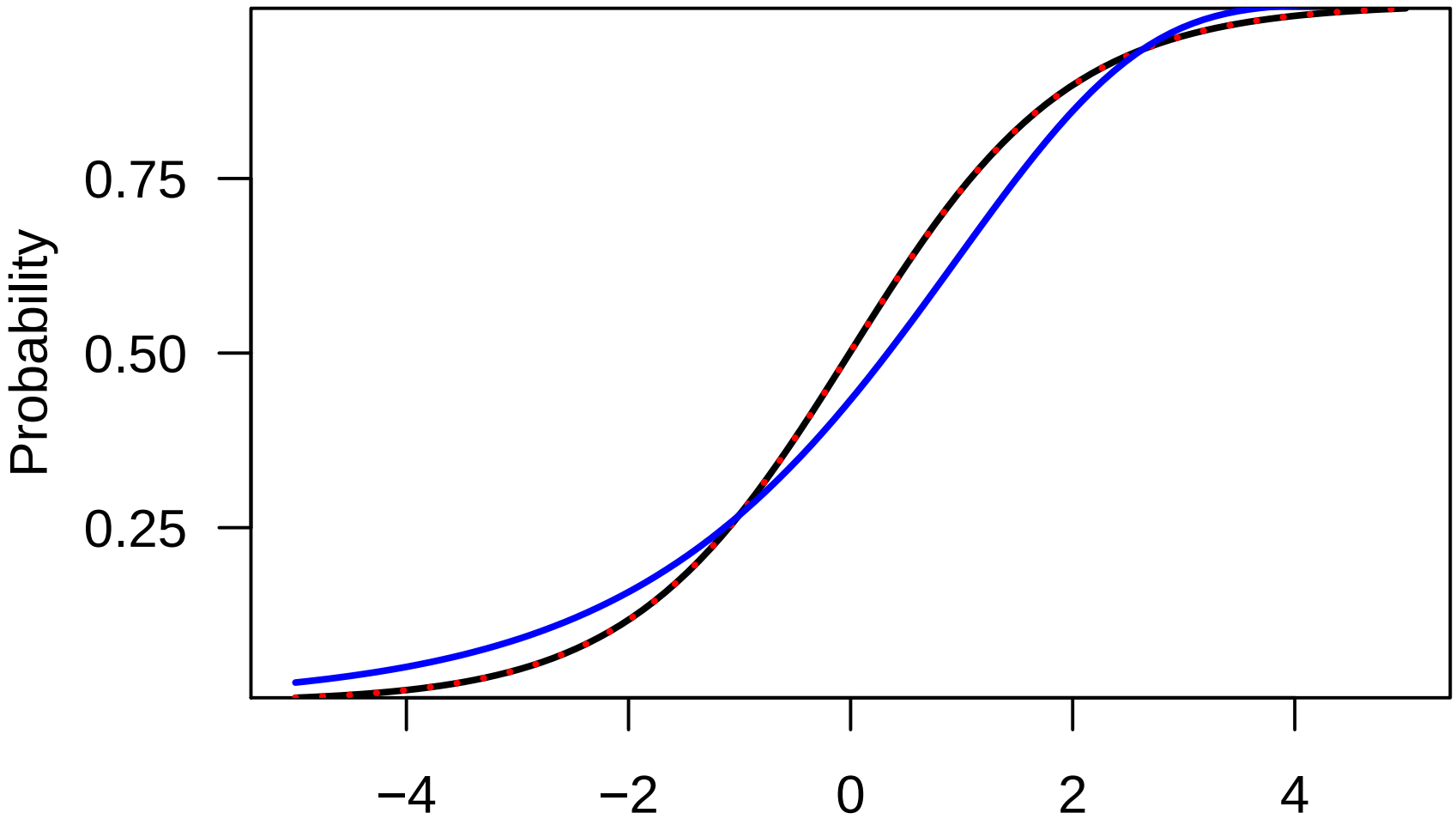


Symmetry

For logit & probit

$$Pr(success|x = 0) = 0.5$$

$$Pr(success|x - \mu) = Pr(failure| - (x - \mu))$$

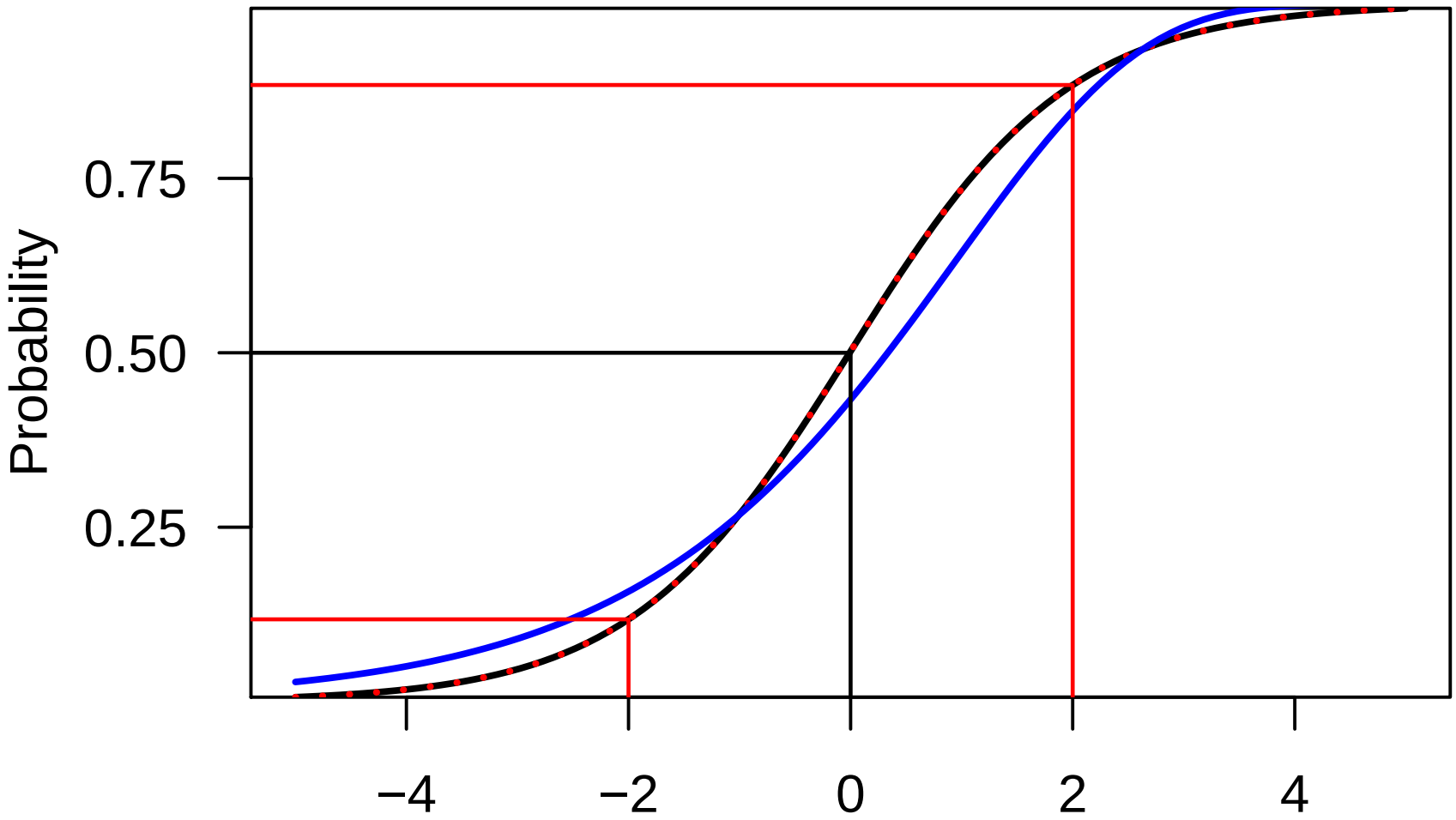


Symmetry

For logit & probit

$$Pr(success|x = 0) = 0.5$$

$$Pr(success|x - \mu) = Pr(failure| - (x - \mu))$$



When to use the different link functions

logit

- default. Usually makes sense

probit

- sometimes with mixed models it's easier to understand. Otherwise, use a logit

cloglog

- when you have Poisson-like counts
- not all count data!
  - sometimes it is not the counting that dominates

A better model, different links

```
fm <- Present ~ prec.mean.sc+I(prec.mean.sc^2)+temp.mean.sc
mod.logit <- glm(fm, data=DryoPil, family=binomial("logit"))
round(mod.logit$coefficients, 3)
```

```
      (Intercept)      prec.mean.sc I(prec.mean.sc^2)
           -0.594             1.826            -0.505
```

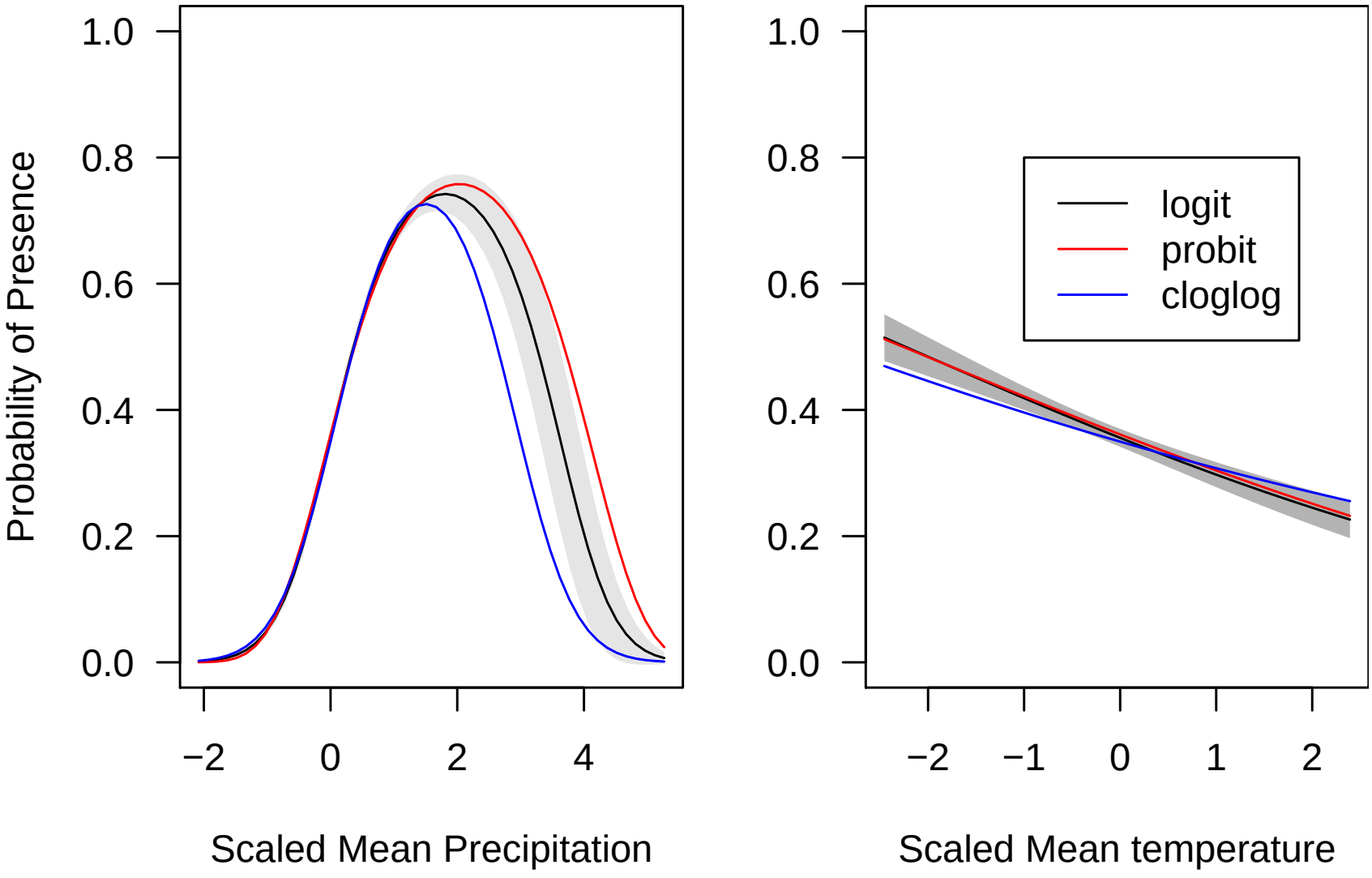
```
mod.probit <- glm(fm, data=DryoPil, family=binomial("probit"))
round(mod.probit$coefficients, 3)
```

```
      (Intercept)      prec.mean.sc I(prec.mean.sc^2)
           -0.355             1.039            -0.256
```

```
mod.cloglog <- glm(fm, data=DryoPil, family=binomial("cloglog"))
round(mod.cloglog$coefficients, 3)
```

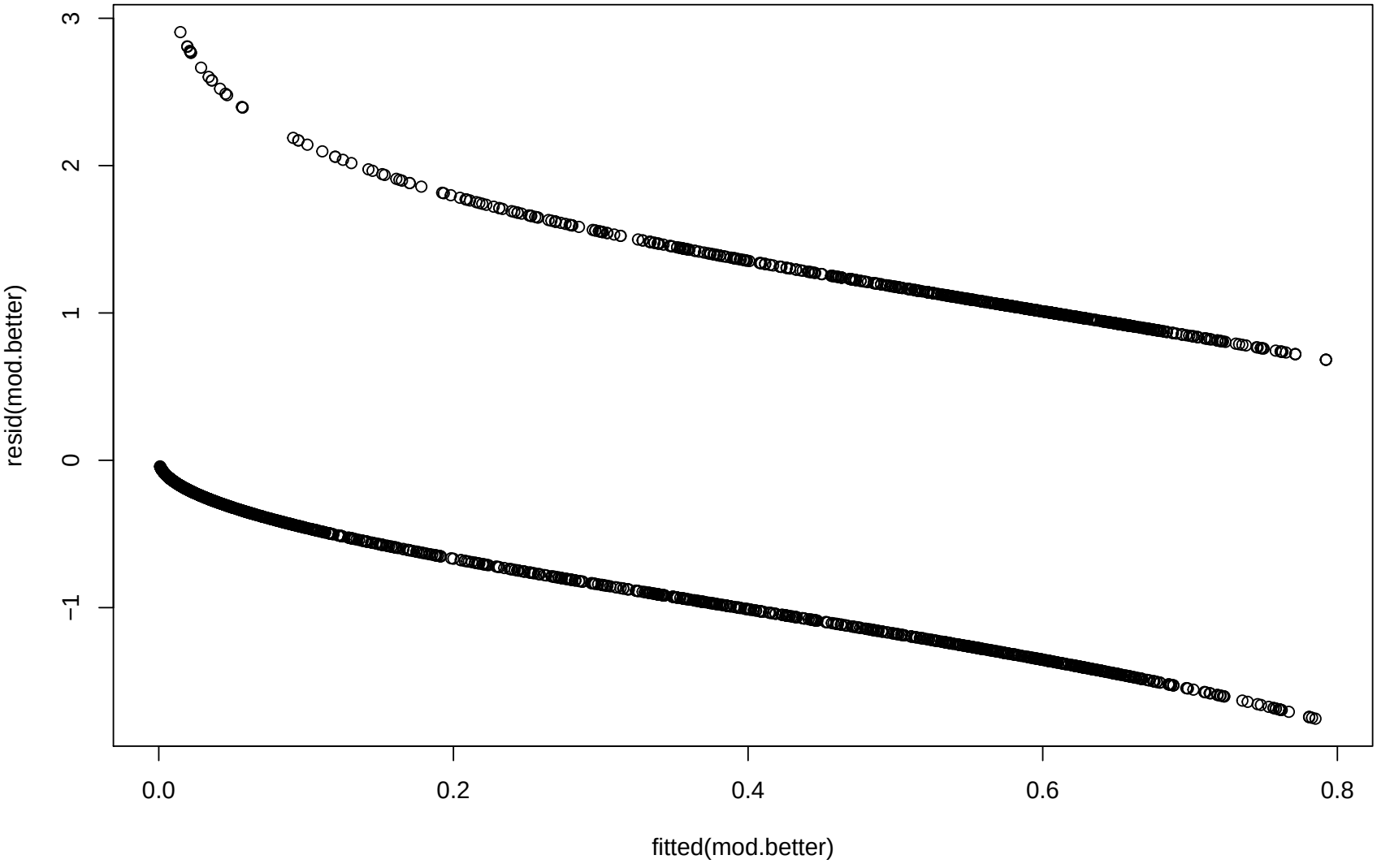
```
      (Intercept)      prec.mean.sc I(prec.mean.sc^2)
           -0.843             1.468            -0.489
```

A better model, different links



Model Checking

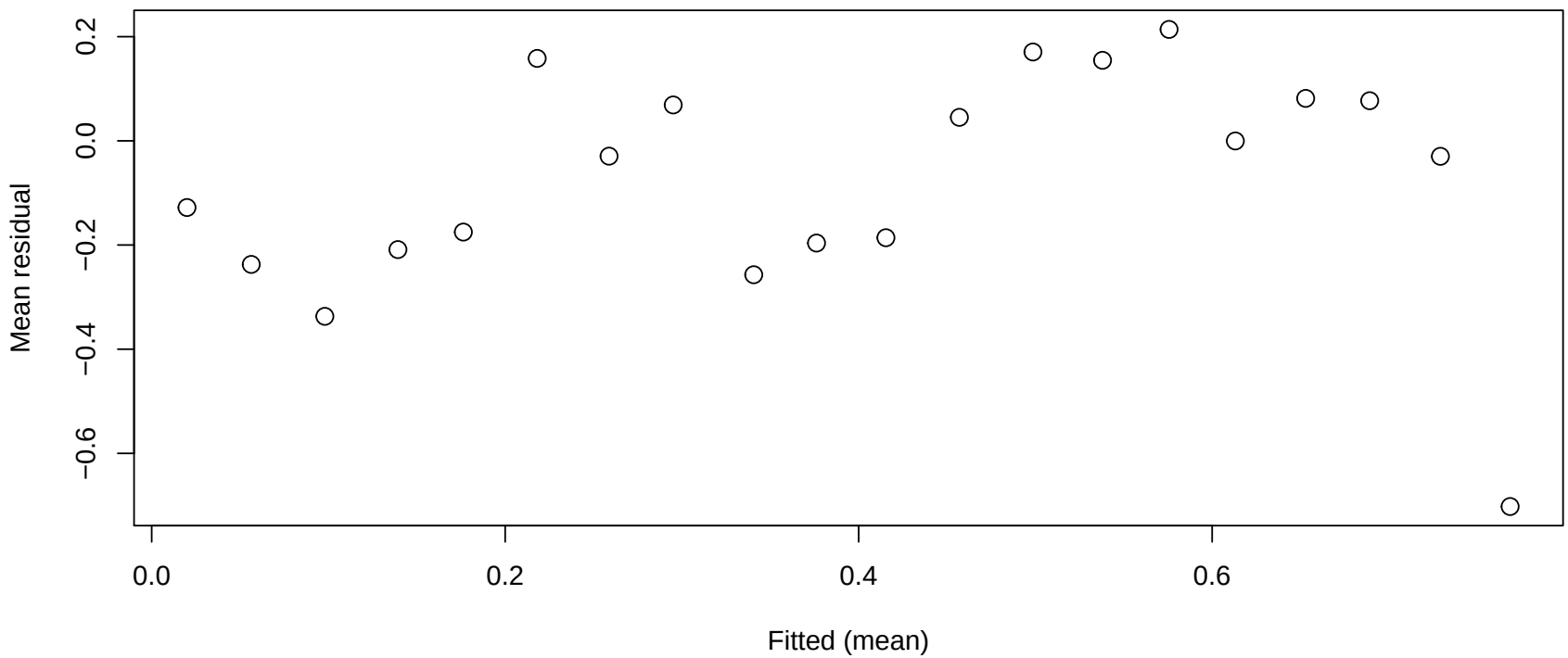
With binary data residuals are useless



Model Checking: grouping residuals

But we can group residuals and take the mean

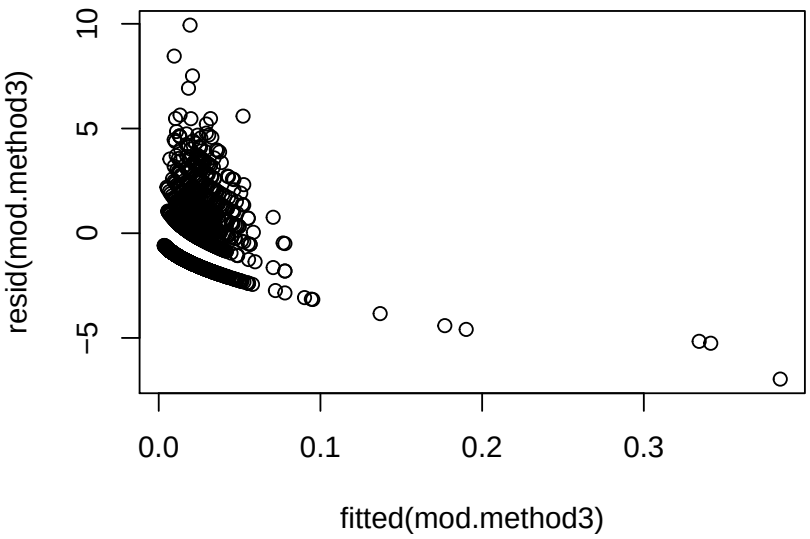
```
# Split data into groups, and calc. mean of residuals
FitCut <- cut(fitted(mod.better), breaks=20, labels=FALSE)
ResGrp <- tapply(resid(mod.better), list(FitCut), mean)
FitGrp <- tapply(fitted(mod.better), list(FitCut), mean)
plot(FitGrp, ResGrp, xlab="Fitted (mean)", ylab="Mean resi
```

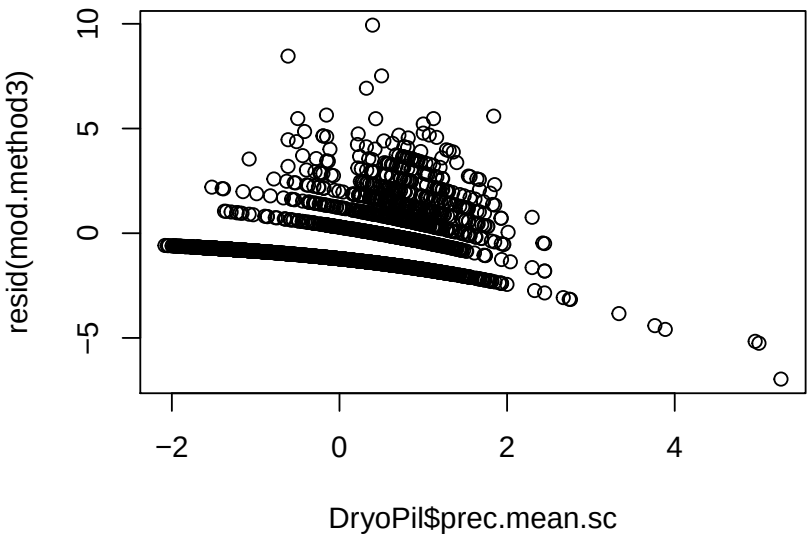


Model Checking

Better when we have a larger number of trials

```
mod.method3 <- glm(cbind(NPres, NAbs) ~ prec.mean.sc,
                   data=DryoPil, family="binomial")
par(mfrow=c(1,2))
plot(fitted(mod.method3), resid(mod.method3))
plot(DryoPil$prec.mean.sc, resid(mod.method3))
```



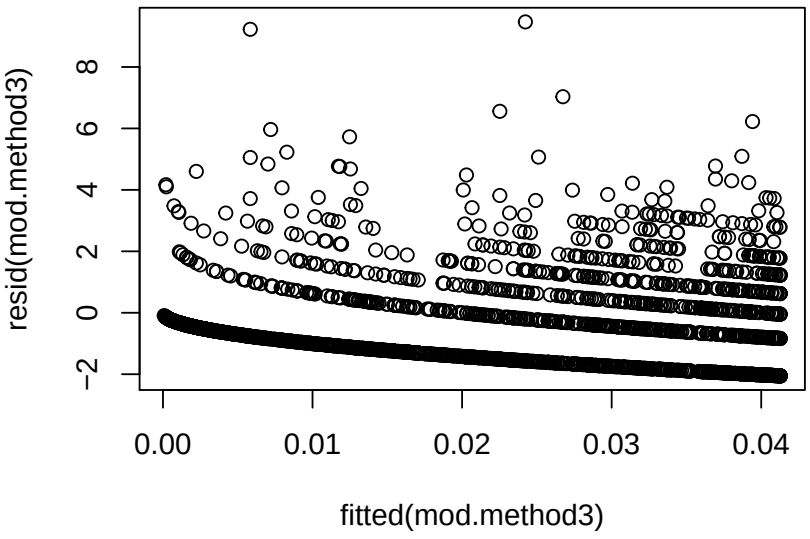


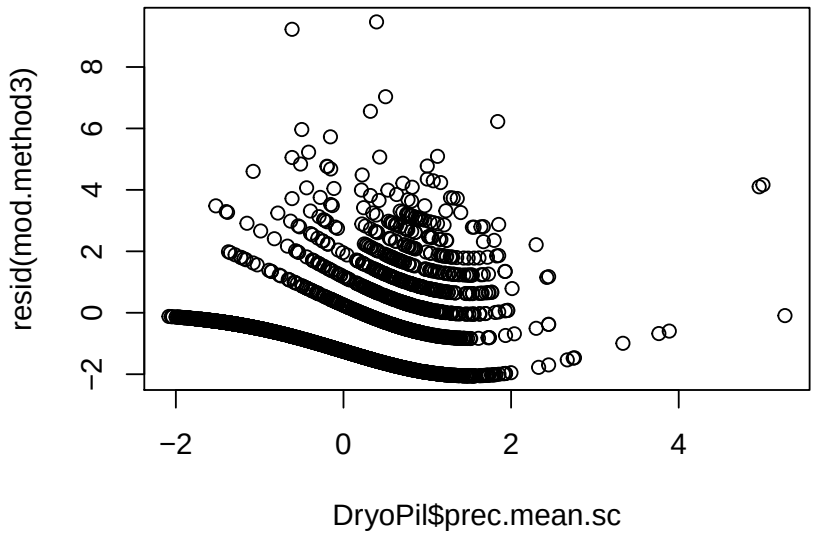
Linear model; residuals negative at both extremes for precipitation

Model Checking

Add a quadratic term...

```
form <- cbind(NPres, NAbs)~prec.mean.sc+I(prec.mean.sc^2)
mod.method3 <- glm(form, data=DryoPil, family="binomial")
par(mfrow=c(1,2))
plot(fitted(mod.method3), resid(mod.method3))
plot(DryoPil$prec.mean.sc, resid(mod.method3))
```





Overdispersion

In the binomial, like the Poisson, the variance is controlled by the mean.

So if we have more variation than expected, we can have overdispesion

We can test for it in the same way.

Overdispersion

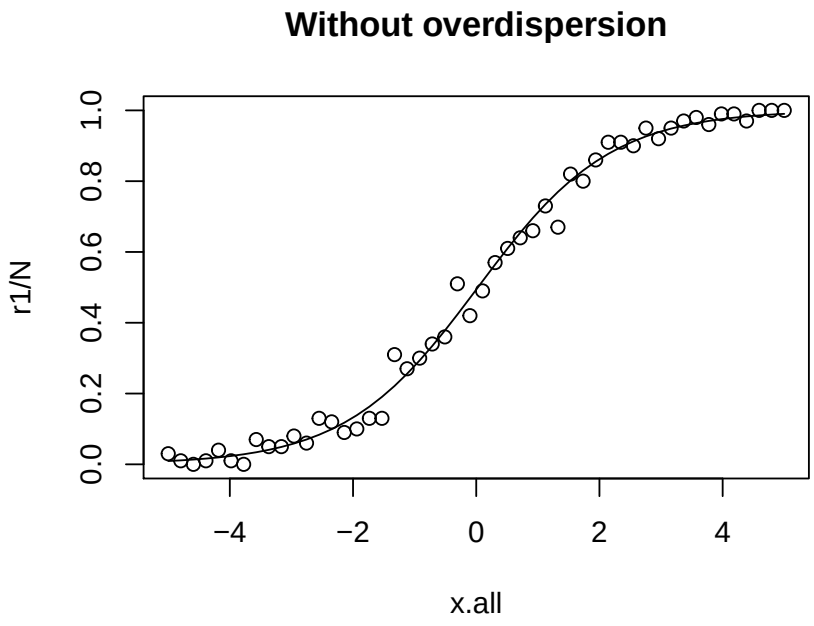
Without, then with overdispersion

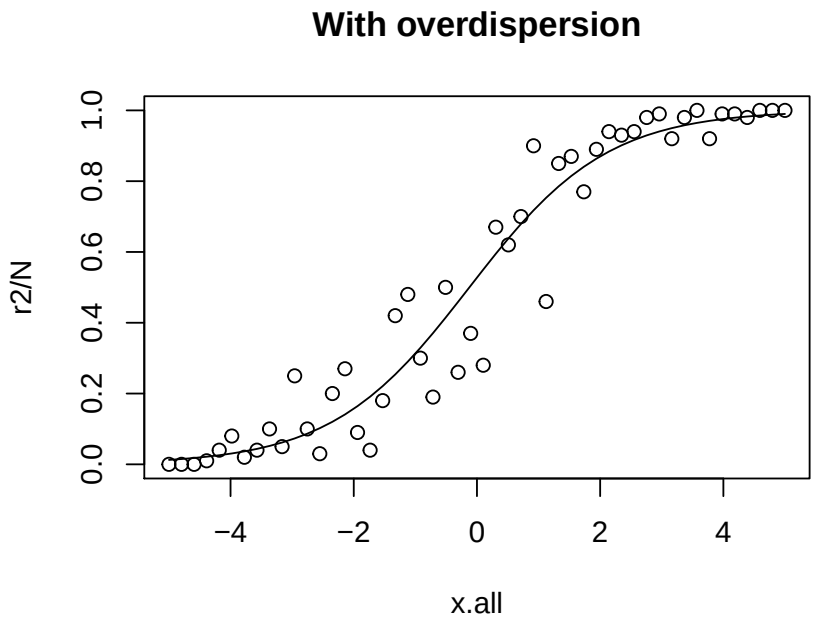
```
N <- 100
x.all <- seq(-5,5,length=50)
p1 <- exp(x.all)/(1+exp(x.all))
r1 <- rbinom(length(p1), size=N, p1)
mod1 <- glm(cbind(r1, N-r1) ~ x.all, family="binomial")

x2 <- rnorm(length(x.all), x.all, 1)
p2 <- exp(x2)/(1+exp(x2))
r2 <- rbinom(length(p2), size=N, p2)
mod2 <- glm(cbind(r2, N-r2) ~ x.all, family="binomial")
```

Overdispersion

```
par(mfrow=c(1,2))
plot(x.all,r1/N, main="Without overdispersion")
lines(x.all, predict(mod1, type="response"))
plot(x.all,r2/N, main="With overdispersion")
lines(x.all, predict(mod2, type="response"))
```





Overdispersion

Without Overdispersion

```
 Deviance =  57.01322  with  48 DF.
p is 0.1748889

Deviance ratio =  1.19
```

With Overdispersion

```
Deviance =  317.3353  with  48 DF.
p is 2.278975e-41

Deviance ratio =  6.61
```

Sparseness

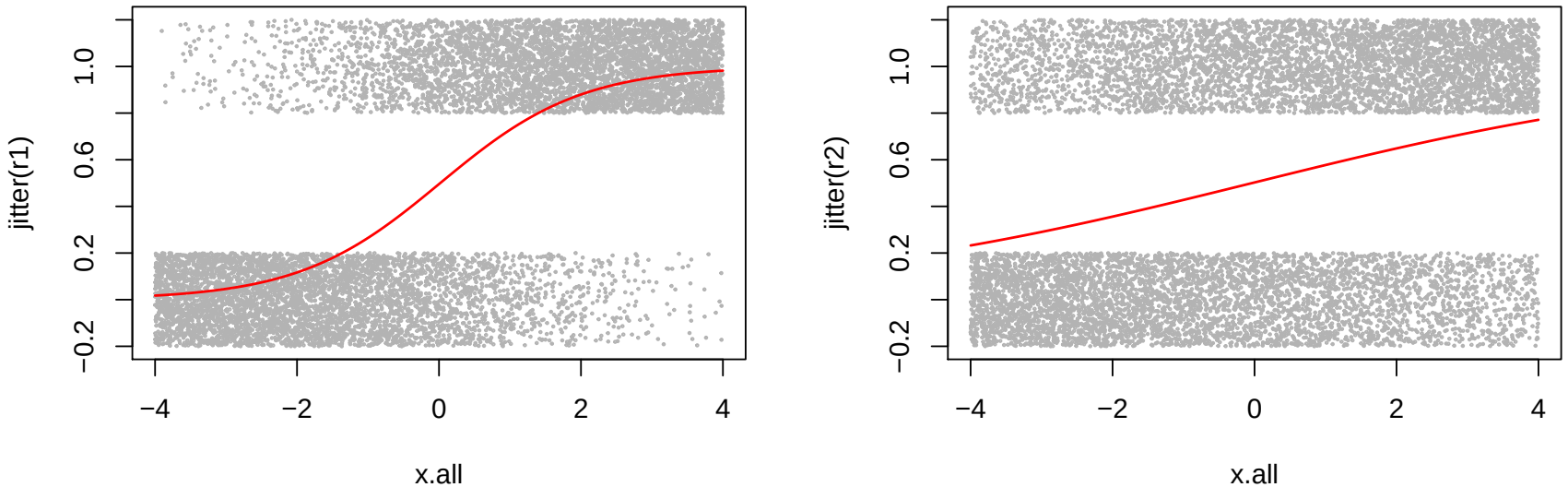
Overdisperson can't always be detected. If we only have 0 s and 1s, then the overdispersion affects the model

```
x.all <- seq(-4,4,length=1e4)
p1 <- exp(x.all)/(1+exp(x.all))
r1 <- rbinom(length(p1), size=1, p1)
mod1 <- glm(r1 ~ x.all, family="binomial")

x2 <- rnorm(length(x.all), x.all, 5)
p2 <- exp(x2)/(1+exp(x2))
r2 <- rbinom(length(p2), size=1, p2)
mod2 <- glm(r2 ~ x.all, family="binomial")
```

Plot Sparseness

```
par(mfrow=c(1,2))
plot(x.all,jitter(r1), cex=0.2, col="grey70")
lines(x.all, predict(mod1, type="response"), col=2, lwd=1.5
plot(x.all,jitter(r2), cex=0.2, col="grey70")
lines(x.all, predict(mod2, type="response"), col=2, lwd=1.5
```



Can't tell if we have a poor model or overdispersion!

Next week

Summary of the course

- this will give me time to write a curriculum etc.