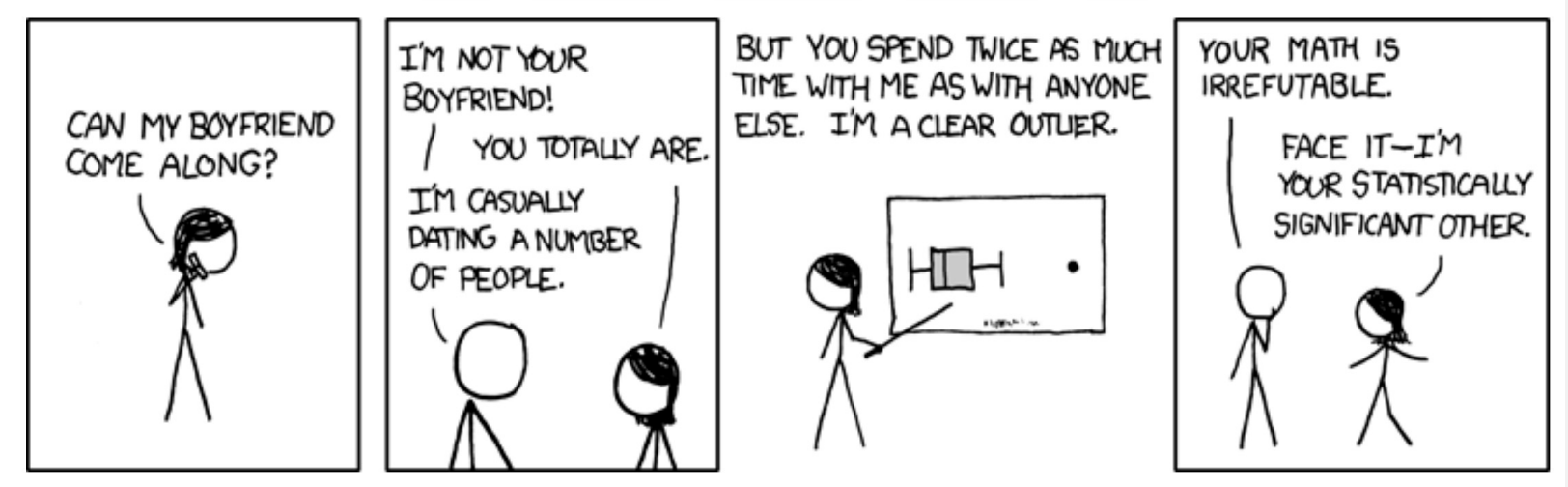
Binary Response Variables, Random vs Fixed Effects, and Outliers

Biology 683

Lecture GLM2

Heath Blackmon



Likelihood and AIC model selection

Think of likelihood as how probable the observed data are under the current model. When we fit a glm we are getting the likelihood of the data assuming the best values for the parameters (beta coefficients). For any empirical dataset these will be very small numbers. It is easy for computers to store the log of very small numbers which means that log likelihoods will be negative numbers.

likelihood = .01	ln(L) = -4.6	better fit
likelihood = .0001	ln(L) = -9.2	worse fit

Adding additional predictor variables will always improve the fit of the data under the model.

R example

So we must penalize for extra parameters

R example

```
fit1 <- glm(y ~ a)
fit2 <- glm(y ~ a + b)
logLik(fit1)
anova(fit1, fit2, test="LRT")
```

Likelihood and AIC model selection

$$AIC = 2k - 2\ln(L)$$

k number of parameters
ln(L) log likelihood of the data (a big negative number)

Models with smaller numbers are better

Differences in AIC can be used to evaluate evidence for one model vs. another.

A model with a ΔAIC value within 1-2 of the best model has substantial support, and should be considered along with the best model.

A ΔAIC value within only 4-7 units of the best model has considerably less support.

A ΔAIC value > 10 indicates that the worse model has virtually no support and should not be consideration.

Burnham and Anderson 1998

Likelihood and AIC model selection

LRT: Does adding this predictor variable reduce the deviation from predictions more than I would expect if the predictor had no impact on the response variable (alpha = .05)

AIC: Is about comparing models and finding the model(s) that seem to match with the data we are observing.

If you are going to use AIC to choose “best” model pay attention to close seconds and whether they “tell a different story”

Logistic Regression

Questions where the response variable is binary are common.

- Dispersal vs Non-dispersal
- Survival
- Presence/Absence of a trait
- Infected/uninfected

Logistic Regression

```
# simulating data for amount of time practiced
coding.pract <- runif(n=100, min=10, max=70)

# creating a vector of probabilites that a student
# passes based on their amount of practice time
probs <- (coding.pract/45)
probs[probs>1] <- 1

# simulating student outcomes based on probabilites
pf <- rbinom(n=100, size=1, prob=probs)
```

Logistic Regression

```
# fitting the model to the data
fit <- glm(pf ~ coding.pract, family=binomial)

# inspecting the model fit
summary(fit)
```

Logistic Regression

```
Call:
glm(formula = pf ~ coding.pract, family = binomial)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.14200   0.07636   0.19673   0.45411   1.63292

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.39414    0.75650  -3.165  0.00155 **
coding.pract  0.12059    0.02778   4.341 1.42e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 102.791  on 99  degrees of freedom
Residual deviance:  65.506  on 98  degrees of freedom
AIC: 69.506

Number of Fisher Scoring iterations: 6
```

log(odds)

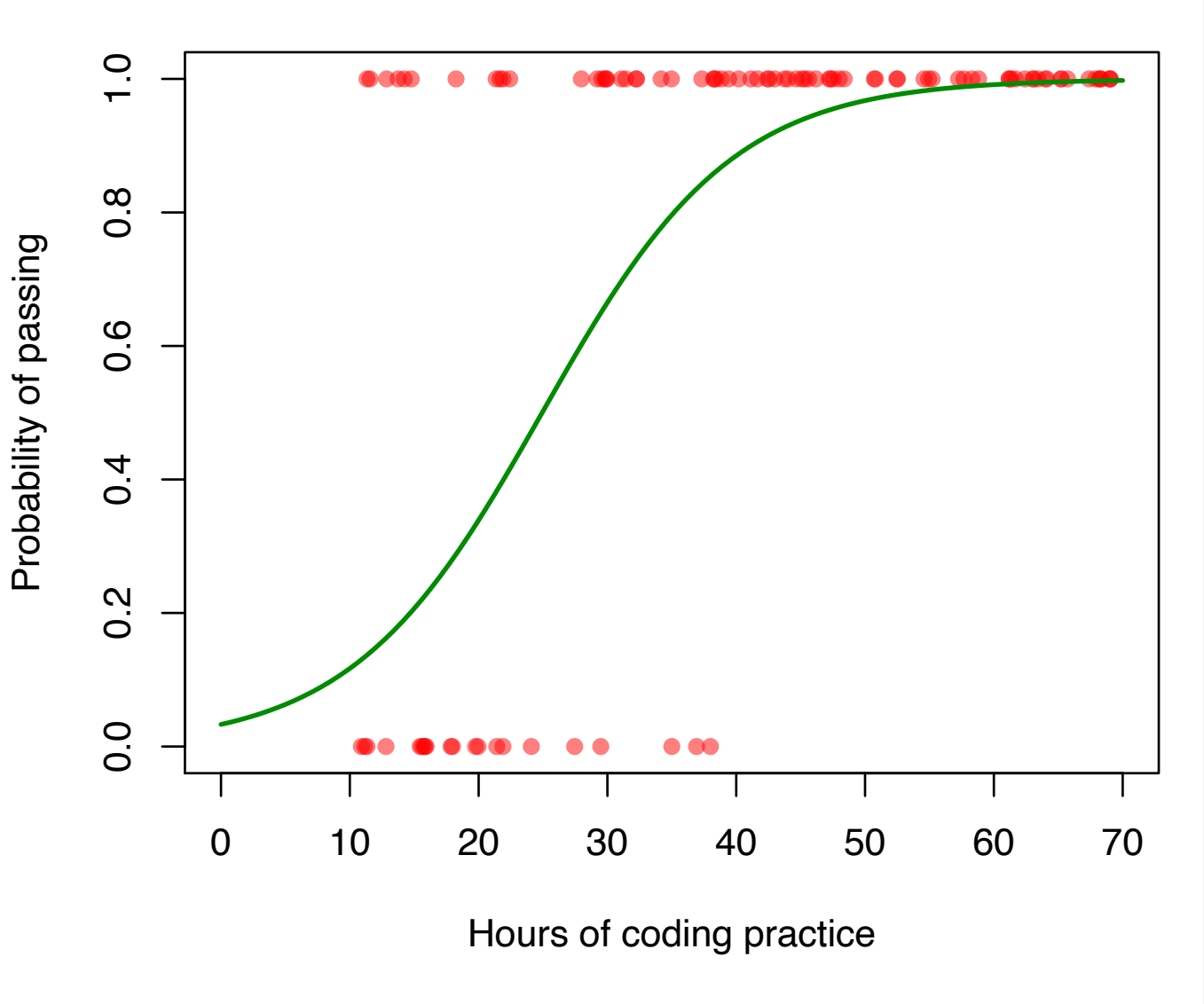
$$\text{ODDS} = \frac{\text{Probability of winning}}{\text{Probability of losing}} = \frac{p}{1-p}$$

Logistic Regression

```
# getting the result from fitting the model
newdat <- data.frame(coding.pract=seq(0, 70,len=100))
newdat$pf = predict(fit, newdata=newdat, type="response")

plot(pf ~ coding.pract, pch=16, col=rgb(1,0,0,.5),xlim=c(0,70),
     ylab="Probability of passing",
     xlab="Hours of coding practice")
lines(newdat$pf ~ newdat$coding.pract, col="green4", lwd=2)
```

Logistic Regression



Logistic Regression

```
# find out when a student has a 50% chance of passing
newdat$coding.pract[min(which(newdat$pf>.5))]
```

> 25.45

Logistic Regression

```
# find out when a student has a 50% chance of passing
newdat$coding.pract[min(which(newdat$pf>.5))]
```

```
> 25.45
```

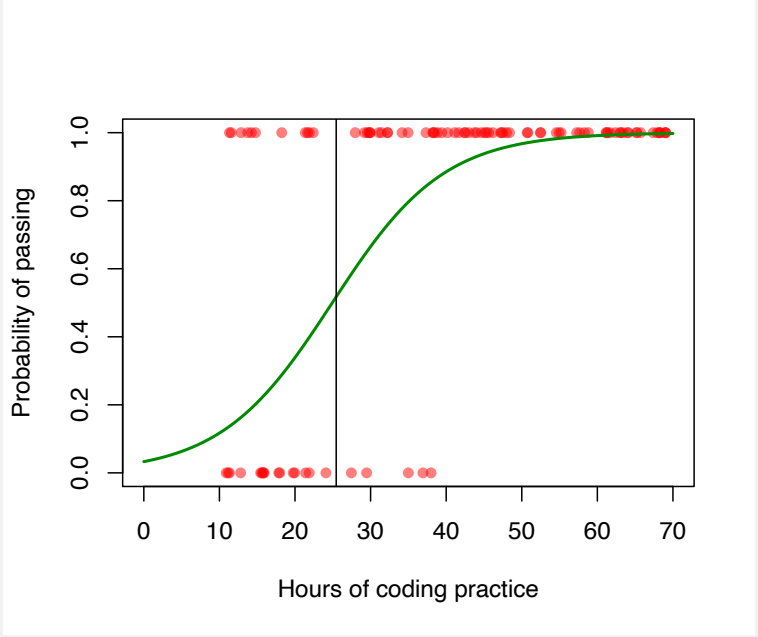
```
plot(pf ~ coding.pract, pch=16, col=rgb(1,0,0,.5),xlim=c(0,70),
     ylab="Probability of passing",
     xlab="Hours of coding practice")
lines(newdat$pf ~ newdat$coding.pract, col="green4", lwd=2)
abline(v=25.45)
```

Logistic Regression

```
# find out when a student has a 50% chance of passing
newdat$coding.pract[min(which(newdat$pf>.5))]
```

```
> 25.45
```

```
plot(pf ~ coding.pract, pch=16, col=rgb(1,0,0,.5),xlim=c(0,70),
     ylab="Probability of passing",
     xlab="Hours of coding practice")
lines(newdat$pf ~ newdat$coding.pract, col="green4", lwd=2)
abline(v=25.45)
```



Logistic Regression

```
# find out when a student has a 50% chance of passing
newdat$coding.pract[min(which(newdat$pf>.5))]
```

> 25.45

How does this compare to what we expected from the simulation?

```
# creating a vector of probabilites that a student
# passes based on their amount of practice time
probs <- (coding.pract/45)
probs[probs>1] <- 1
```

> 25/45
[1] 0.5555556