Logistic regression models for binary data

FW8051 Statistics for Ecologists

Department of Fisheries, Wildlife and Conservation Biology

Outline

- Logistic regression
  - Model formulation
  - Interpretation of regression parameters
- Comparing Models and Assessing Fit
  - Likelihood Ratio tests, AIC
  - Goodness-of-fit testing
  - Residual plots
- Plotting/visualizing model fit
  - Logit scale versus probability scale
  - Confidence intervals on probability scale
- Bayesian formulation (practice!)

Logistic regression

Model for binary (0/1) data or binomial data (number of 1's out of n trials).

$$Y_i|X_i \sim \text{Binomial}(n_i, p_i)$$

$$logit(p_i) = log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{1,i} + \dots \beta_p X_{p,i}$$

- Random component = Bernoulli or binomial distribution
- Systematic component: logit(p_i) or log(odds) = linear combination of predictors

Remember, for binary data, $E[Y_i|X_i] = p_i$
$\Rightarrow logit(p_i) = \beta_0 + \beta_1 X_{1,i} + \dots \beta_p X_{p,i}$

$\Rightarrow p_i = \frac{\exp(\beta_0+\beta_1 X_{1,i}+\dots\beta_p X_{p,i})}{1+\exp(\beta_0+\beta_1 X_{1,i}+\dots\beta_p X_{p,i})}$ (can use `plogis` function in R)

Logistic regression

$$Y_i|X_i \sim \text{Binomial}(n_i, p_i)$$

$$logit(p_i) = log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{1,i} + \dots \beta_p X_{p,i}$$

$\frac{p}{1-p}$ is referred to as the odds.

The link function, $\log\left(\frac{p}{1-p}\right)$, is referred to as logit.

Thus, we can describe our model in the following ways:

- We are modeling $\log\left(\frac{p}{1-p}\right)$ as a linear function of $X_1, \dots, X_p$.
- We are modeling the logit of p as a linear function of $X_1, \dots, X_p$.
- We are modeling the log odds of p as a linear function of $X_1, \dots, X_p$.

Odds = $\frac{p}{1-p}$

If the probability of winning a bet is = 2/3, what are the odds of winning?

odds = $\frac{p}{1-p}$ = (2/3 $\div$ 1/3) = 2 (or "2 to 1").

Table 6.1. Various probabilities, odds and log odds. The table shows how log odds are calculated from probabilities.

P_i	0.001	0.1	0.3	0.4	0.5	0.6	0.7	0.9	0.999
$1-P_i$	0.999	0.9	0.7	0.6	0.5	0.4	0.3	0.1	0.001
O_i	0.001	0.11	0.43	0.67	1	1.5	2.33	9	999
$\mathrm{Ln}(O_i)$	−6.91	−2.20	−0.85	−0.41	0	0.41	0.85	2.20	6.91

{From Zuur et al. 2007. Analyzing Ecological data}

Odds can vary between 0 and ∞, so log(odds) can live on $-\infty$ to ∞.

Odds Ratios: exp(β)

Consider a regression coefficient for a categorical variable:

$$logit(p_i) = log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 I(group == B)_i$$

$I(group == B)_i$ = 1 if observation i is from Group B and 0 if Group A

- Odds for group B = $\frac{p_B}{1-p_B}$ = $\exp(\beta_0 + \beta_1)$
- Odds for group A = $\frac{p_A}{1-p_A}$ = $\exp(\beta_0)$

Consider the ratio of these odds:

$$\frac{\frac{p_B}{1-p_B}}{\frac{p_A}{1-p_A}} = \frac{\exp(\beta_0+\beta_1)}{\exp(\beta_0)} = \frac{e^{\beta_0}e^{\beta_1}}{e^{\beta_0}} = \exp(\beta_1)$$

So, $\exp(\beta_1)$ gives an odds ratio (or ratio of odds) for Group B relative to group A.

Odds Ratios: exp(β)

Consider a continuous predictor, X:

$$logit(p_i) = log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 X_i$$

β_1 gives the change in log odds per unit change in X.

- Odds when $X_i = a$ is given by $\frac{p_i}{1-p_i}$ = $\exp(\beta_0 + \beta_1 a)$
- Odds when $X_j = a + 1$ is given by $\frac{p_j}{1-p_j}$ = $\exp(\beta_0 + \beta_1(a + 1))$

Consider the ratio of these odds:

$$\frac{\frac{p_j}{1-p_j}}{\frac{p_i}{1-p_i}} = \frac{\exp(\beta_0+\beta_1(a+1))}{\exp(\beta_0+\beta_1 a)} = \frac{e^{\beta_0}e^{\beta_1 a}e^{\beta_1}}{e^{\beta_0}e^{\beta_1 a}} = \exp(\beta_1)$$

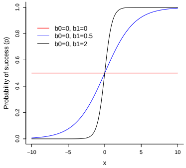
So, $\exp(\beta_1)$, gives the odds ratio for two observation that differ by 1 unit of X.

Multiple predictors

For multiple predictor models,

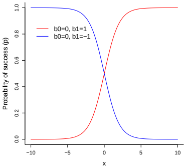
$\exp(\beta_i)$ gives the odds ratio for observations where X_i differs by 1 unit, while holding everything else constant!

$$logit(p_i) = log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 X_i$$



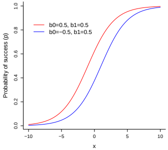
The slope coefficient β_1 controls how quickly we transition from 0 to 1.

$$logit(p_i) = log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_i$$



The sign of β_1 determines if p increases or decreases as we increase X.

$$logit(p_i) = log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 X_i$$



β_0:

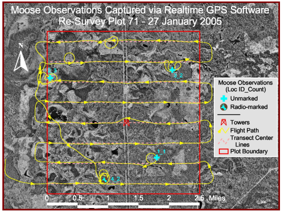
- Controls the height of the curve when $X = 0$.
- Gives the log odds of detection when all predictor variables = 0
- $E[Y_i|X_i = 0] = \frac{exp(\beta_0)}{1+exp(\beta_0)}$ (equals 1/2 if $\beta_0 = 0$).

Sightability Surveys: Minnesota Moose

124 'trials', 2005-2007

n_0 = 65 missed groups

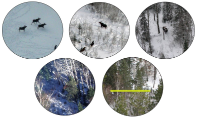
n_1 = 59 observed groups



- Binary observations, $Y_i = 0$ (missed) or 1 (seen).
- Covariates thought to influence detection.

Covariates

- Visual obstruction
- Survey year (may be due to different observers)



$$Y_i|X_i \sim Bernouli(p_i)$$

$$logit(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 voc_i$$

Assumptions:

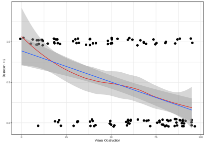
- observations are independent
- log odds is a linear function of *voc*
- mean and variance depend on *voc*

$E[Y_i|X_i] = p_i; var[Y_i|X_i] = p_i(1-p_i)$ with:

$$p_i = \frac{\exp(\beta_0+\beta_1 voc_i)}{1+\exp(\beta_0+\beta_1 voc_i)}$$

Visual Obstruction

```
ggplot(exp.m, aes(voc,observed))+theme_bw()+
   geom_point(position = position_jitter(w = 2, h = 0.05), size=3) +
   geom_smooth(colour="red")  +geom_smooth(method="lm")+
 xlab("Visual Obstruction") +
  ylab("Detection = 1")
```

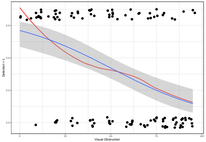


- `lm` would eventually predict $p_i \geq 1$ and $p_i \leq 0$
- would assume constant variance when $var(p_i) = p_i(1-p_i)$

```
mod1<-glm(observed~voc, data=exp.m, family=binomial())
summary(mod1)
```

```
Call:
glm(formula = observed ~ voc, family = binomial(), data = exp.m)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8056  -0.9071  -0.6218   0.9745   1.8647

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.759933   0.460137   3.825 0.000131 ***
voc         -0.034792   0.007753  -4.487 7.21e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 171.61  on 123  degrees of freedom
Residual deviance: 147.38  on 122  degrees of freedom
AIC: 151.38

Number of Fisher Scoring iterations: 4
```

```r
ggplot(exp.m, aes(voc,observed))+ theme_bw() +
    geom_point(position = position_jitter(w = 2, h = 0.05), size=3) +
    xlab("Visual Obstruction") +  geom_smooth(se=F, colour="red") +
    stat_smooth(method="glm", method.args = list(family = "binomial"))
    ylab("Detection = 1")
```




```r
exp.m$year<-as.factor(exp.m$year)
mod2<-glm(observed~voc+year, data=exp.m, family=binomial())
summary(mod2)
```

```
Call:
glm(formula = observed ~ voc + year, family = binomial(), data = exp.m)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9351  -0.8411  -0.4561   0.9493   1.8680

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.453203   0.622248   3.942 8.06e-05 ***
voc         -0.037391   0.008199  -4.560 5.11e-06 ***
year2006    -0.453862   0.516567  -0.879   0.3796
year2007    -1.111884   0.508269  -2.188   0.0287 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 171.61  on 123  degrees of freedom
Residual deviance: 142.23  on 120  degrees of freedom
AIC: 150.23

Number of Fisher Scoring iterations: 4
```

```r
mod1$coef
```

```
(Intercept)         voc
 1.75993309 -0.03479153
```

Regression coefficient for voc (visual obstruction) = -0.039.

- The log odds of being detected decreases by 0.039 per unit increase in visual obstruction
- The odds of being detected decreases by exp(0.039) = 0.96 per unit increase in visual obstruction

Intercept = 2.12 = log(odds) when VOC = 0.

```r
# p(Y=1|voc=0)  = exp(coef(mod1)[1])/(1+exp(coef(mod1)[1]))
plogis(coef(mod1)[1])
```

```
(Intercept)
  0.8532013
```

We see roughly 90% of moose if there is no visual obstruction.

```r
coef(mod2)
```

```
(Intercept)         voc    year2006    year2007
 2.45320264 -0.03739118 -0.45386154 -1.11188432
```

Year 2004: $log(p_i/(1-p_i))$ = 2.75 − 0.0422VOC

Year 2005: $log(p_i/(1-p_i))$ = 2.75 − 0.0422VOC + 0.006879

So, 0.006879 gives the difference in log odds between years 2005 and 2004 (if we hold VOC constant).

exp(0.006879) = 1.0069 = odds ratio (year 2005 to year 2004)

odds ratio = $\frac{p_{2005}/(1-p_{2005})}{p_{2004}/(1-p_{2004})}$ = 1.0069

If confidence limits for β include 0 or confidence limits for $exp(\beta)$ include 1, then we do not have enough evidence to say that years differ in their detection probabilities.

```
mod2$coef
```

```
(Intercept)         voc    year2006    year2007
 2.45320264 -0.03739118 -0.45386154 -1.11188432
```

```
sqrt(diag(vcov(mod2)))
```

```
(Intercept)         voc    year2006    year2007
0.622247867 0.008199483 0.516567443 0.508269279
```

```
exp(rep(0.006879, 2)+c(-1.96, 1.96)*0.53664) # exp(beta +/-1.96SE)
```

```
[1] 0.3517145 2.8826021
```

95% Confidence interval for odds ratio = (0.35, 2.88) includes 1 (not statistically significant)

Confint

```
(ci.prof<-confint(mod2))
```

```
Waiting for profiling to be done...
```

```
                  2.5 %     97.5 %
(Intercept)  1.30341777  3.7586692
voc         -0.05448153 -0.0221268
year2006    -1.48479529  0.5516852
year2007    -2.14380706 -0.1382692
```

These are profile-likelihood based confidence intervals based on "inverting" the likelihood ratio test (see Maximum Likelihood notes).

```
exp(ci.prof[3,])
```

```
    2.5 %    97.5 %
0.2265487 1.7361764
```

Profile-likelihood based intervals should have better statistical properties with small data sets (better coverage rates).

Likelihood ratio tests

We can again use difference in deviences (equivalent to likelihood ratio tests) to compare full and reduced models.

```
drop1(mod2, test="Chisq")
```

```
Single term deletions

Model:
observed ~ voc + year
       Df Deviance    AIC     LRT  Pr(>Chi)
<none>      142.23 150.23
voc     1   168.20 174.20 25.9720 3.464e-07 ***
year    2   147.38 151.38  5.1558   0.07593 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

voc is an important predictor, the importance of `year` is less clear.

ANOVA function (car package)

Or use Anova in car package

```
library(car)
```

```
Anova(mod2)
```

```
Analysis of Deviance Table (Type II tests)

Response: observed
     LR Chisq Df Pr(>Chisq)
voc   25.9720  1  3.464e-07 ***
year   5.1558  2    0.07593 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

AIC

We can compare nested or non-nested models using the AIC function

```
AIC(mod1, mod2)
```

```
     df      AIC
mod1  2 151.3824
mod2  4 150.2266
```

Goodness-of-fit

Can adapt our general approach for testing goodness-of-fit using Pearson residuals (r_i)

$$r_i = \frac{Y_i - E[Y_i|X_i]}{\sqrt{Var[Y_i|X_i]}}$$

- $E[Y_i|X_i] = p_i = \frac{exp(\beta_0+\beta_1 x_1+\ldots\beta_k x_k)}{1+exp(\beta_0+\beta_1 x_1+\ldots\beta_k x_k)}$
- $Var[Y_i|X_i] = p_i(1-p_i)$

See textbook for an implementation of this test...

Hosmer-Lemeshow test (similar test)

Group Observations by deciles of their predicted values to form groups, then calculate the expected and observed number of successes and failures for each group:

Model Results

	$G_1 = [0, \hat{\pi}_1]$	$G_2 = (\hat{\pi}_1, \hat{\pi}_2]$	...	$G_{10} = (\hat{\pi}_9, 1]$
Successes	$\sum_{x_i \in G_1} \hat{\pi}_i$	$\sum_{x_i \in G_2} \hat{\pi}_i$	...	$\sum_{x_i \in G_{10}} \hat{\pi}_i$
Failures	$n_1 - \sum_{x_i \in G_1} \hat{\pi}_i$	$n_2 - \sum_{x_i \in G_2} \hat{\pi}_i$	...	$n_{10} - \sum_{x_i \in G_{10}} \hat{\pi}_i$

Observed Results

	$G_1 = [0, \hat{\pi}_1]$	$G_2 = (\hat{\pi}_1, \hat{\pi}_2]$	...	$G_{10} = (\hat{\pi}_9, 1]$
Successes	$\sum_{x_i \in G_1} y_i$	$\sum_{x_i \in G_2} y_i$	...	$\sum_{x_i \in G_{10}} y_i$
Failures	$n_1 - \sum_{x_i \in G_1} y_i$	$n_2 - \sum_{x_i \in G_2} y_i$	...	$n_{10} - \sum_{x_i \in G_{10}} y_i$

See: **Goodness of fit with binary data** here: http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture21.htm

Hosmer-Lemeshow Test

$$\chi^2 = \sum_{i=1}^{n_g} \frac{(O_i - E_i)^2}{E_i} \sim \chi^2_{g-2}$$

where g = number of groups.

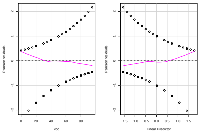
Hosmer-Lemeshow test

```
library(ResourceSelection)
hoslem.test(exp.m$observed, fitted(mod1), g=8)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  exp.m$observed, fitted(mod1)
X-squared = 3.2505, df = 6, p-value = 0.7768
```

Residual plots

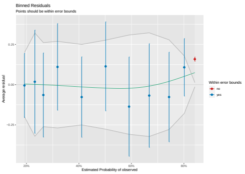
```
residualPlots(mod1)
```




```
    Test stat Pr(>|Test stat|)
voc    0.6873           0.4071
```

Binned residual plot

```
binplot<-performance::binned_residuals(mod1)
plot(binplot)
```



Visualizing models

Probability Scale

We can also summarize models by getting predicted values: $P(\text{detect animal}|\text{voc})$:

- $logit(p_i) = \beta_0 + \beta_1 x_1 + \ldots \beta_k x_k$
- $P(Y_i = 1|X = x) = p_i = \frac{e^{\beta_0+\beta_1 x_1+\ldots\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\ldots\beta_k x_k}}$ (inverse logit)
- Can use predict function with `"type= response"`
- Or `plogis(predict(type="link"))`; plogis(u) $= \frac{e^u}{1+e^u}$

Supporting Theory

The estimates of β are maximum likelihood estimates, found by maximizing:

$$L(\beta; y, x) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}, \text{ with}$$

$$p_i = \frac{e^{\beta_0+\beta_1 x_1+\ldots\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\ldots\beta_k x_k}}$$

Remember, for large samples, $\hat{\beta} \sim N(\beta, \Sigma)$.

We can use this theory to conduct tests (t-statistics output by the `summary` function) and to get confidence intervals.

- logit(p) = $\beta_0 + \beta_1 x_1 + \ldots \beta_k x_k$ is more "Normal" than p
- Generate confidence intervals for logit(p), then back-transform to get confidence intervals for p
- Ensures the confidence intervals will live on the (0,1) scale
- Intervals will not be symmetric

Confidence Intervals

Statistical Theory:

- $Var(ax + by) = a^2 var(x) + b^2 var(y) + 2abCov(x, y)$
- If x and y and normally distributed, $ax + by$ will be normally distributed

Let X = matrix of data

- Here, X has 2 columns (column of 1's, column with x)
- n rows (one for each observation).
- We can use `model.matrix` to see it

Let β = p x 1 vector of parameters

- Here, p = 2, $\beta = (\beta_0, \beta_1)$

We can generate $logit(\hat{p}|X)$ and its SE using matrix multiplication:

- $logit(\hat{p}|X) = X\hat{\beta}$ (use %*% in R to perform matrix multiplication)
- Variance/covariance of $logit(\hat{p}|X) = X\Sigma_{\beta}X^T$

Confidence Intervals

In practice, we can use `predict(model, newdata=, type="link", se=TRUE)` to get predictions on logit scale.

Then use `plogis(p.hat$fit +/- 1.95*p.hat$se.fit)` to transform the limits back to the probability scale.

A note on model visualization

Model 2: observed $\sim$ voc + year is additive on the logit scale

- Differences in logit(p) among years will not depend on voc
- Differences in p, will however, depend on voc!

See: LogisticModelsFrequentist.html

- Can always create your own "effect" plots by calculating predicted values for different combinations of your predicted values (as in the in-class exercise)
- Can use the `effects` package to do something similar

Effect plots on probability scale

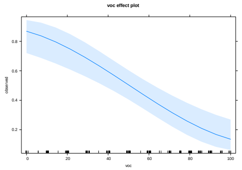
Use `effects` package:

- Fixes all continuous covariates (other than the one of interest) at their mean values
- Categorical predictors: averages predictions on link scale, weighted by proportion of data in each category

Effect plots

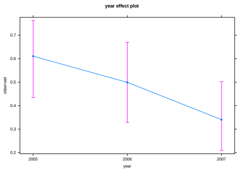
Use `type="response"` to plot on response scale

```
library(effects)
plot(effect("voc", mod2), type="response")
```



Effect plots

```
plot(effect("year", mod2), type="response")
```



Year Effects

```
effect("year", mod2)
```

```
 year effect
year
     2005      2006      2007
0.6107432 0.4991440 0.3404147
```

Estimates of $P(\text{Observed} = 1 | year = year_i, voc = \bar{voc})$

```
newdata<-data.frame(voc=rep(mean(exp.m$voc),3),
                    year=c("2005","2006","2007"))
predict(mod2, newdata=newdata, type="resp")
```

```
        1         2         3
0.6107432 0.4991440 0.3404147
```

VOC Effect

```
effect("voc", mod2)
```

```
 voc effect
voc
        0        20        50        80       100 
0.8684553 0.7575962 0.5044524 0.2490051 0.1356677 
```

$P(\text{Observed} = 1|VOC, \bar{year})$?

Weighted mean (across categories, here = “year”), with weights given by the proportion of observations in each category.

VOC Effect

```
effect("voc", mod2)
```

```
 voc effect
voc
        0        20        50        80       100 
0.8684553 0.7575962 0.5044524 0.2490051 0.1356677 
```

```
p.years<-table(exp.m$year)/nrow(exp.m)
newdata<-data.frame(expand.grid(year=c("2005","2006","2007"),
                                voc=seq(0,100,20)))
newdata$pred<-predict(mod2, newdata=newdata, type="link")
plogis(sum(newdata$pred[1:4]*p.years))
```

```
[1] 0.8654255
```

Confidence intervals

```
summary(effect("year", mod2))
```

```
 year effect
year
     2005      2006      2007 
0.6107432 0.4991440 0.3404147 

 Lower 95 Percent Confidence Limits
year
     2005      2006      2007 
0.4340997 0.3285547 0.2088034 

 Upper 95 Percent Confidence Limits
year
     2005      2006      2007 
0.7624246 0.6699327 0.5023149 
```

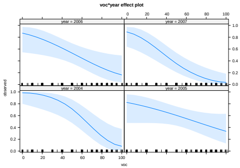
Models with interactions

```
mod3<-glm(observed~voc+year+voc:year, data=exp.m, family=binomial())
summary(mod3)
```

```
Call:
glm(formula = observed ~ voc + year + voc:year, family = binomial(), 
    data = exp.m)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.86221  -0.98652  -0.06984   0.86697   2.03839  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)   
(Intercept)   4.23935    1.45979   2.904  0.00368 **
voc          -0.06661    0.02259  -2.949  0.00319 **
year2005     -2.70469    1.68093  -1.609  0.10761   
year2006     -2.34628    1.70158  -1.379  0.16793   
year2007     -2.15134    1.65108  -1.303  0.19258   
voc:year2005  0.04437    0.02600   1.707  0.08789 . 
voc:year2006  0.03127    0.02730   1.145  0.25212   
voc:year2007  0.01286    0.02717   0.473  0.63604   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 221.81  on 159  degrees of freedom
Residual deviance: 171.08  on 152  degrees of freedom
AIC: 187.08

Number of Fisher Scoring iterations: 5
```

Effect Plot

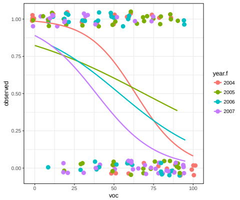
```
plot(effect("voc:year", mod3), type="response")
```



ggplot2

```
ggplot(exp.m, aes(x=voc, y=observed, colour=year)) + theme_bw()+
   geom_point(position = position_jitter(w = 2, h = 0.05), size=3) +
   stat_smooth(method="glm", method.args = list(family = "binomial"),
               se=FALSE)
```

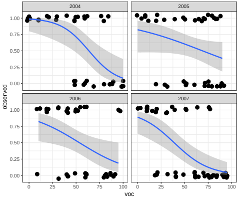
```
`geom_smooth()` using formula 'y ~ x'
```



ggplot2

```
ggplot(exp.m, aes(x=voc, y=observed)) + theme_bw()+
   geom_point(position = position_jitter(w = 2, h = 0.05), size=3) +
   stat_smooth(method="glm", method.args = list(family = "binomial"))+
    facet_wrap(~year)
```

```
`geom_smooth()` using formula 'y ~ x'
```



JAGS

Will use a similar structure as we used for count models:

- A linear predictor, $\eta = \beta_0 + \beta_1 x_1$ (x_1 = voc)
- $p_i = g^{-1}(\eta) = \frac{e^{\eta_i}}{1+e^{\eta_i}}$
- $Y[i] \sim$ dbin(p[i],1)
- Require priors for β_0 and β_1, e.g., N(0, 0.01)

Gelman's recommendations (see arxiv.org/pdf/0901.4011.pdf):

- scale continuous predictors so they have mean 0 and sd = 0.5
- using a non-informative Cauchy prior dt(0, pow(2.5,-2), 1)

In class exercise: adapt the JAGS code for fitting mod1 (voc only) to allow fitting of mod2 (voc + year).

ROC curves

ROC = Receiver operating characteristic curve

- Often used to evaluate and compare fit of binary regression models.

We could use a threshold, T, for $\hat{p}_i$ and set $\hat{Y}_i$ equal to 1 when $\hat{p}_i \geq T$ and 0 otherwise. Then, compare $\hat{Y}_i$ to Y_i.

- Results would depend on our chosen threshold, T.
- ROC curve: considers all possible thresholds and plots True Positive Rate, $P(\hat{Y}_i = 1|Y_i = 1)$ versus False Positive Rate, $P(\hat{Y}_i = 1|Y_i = 0)$
- AUC gives the area under the ROC curve (higher values suggest better predictive value)