STAT 426

Logistic Regression: Building, Checking, Applying

Spring 2018

Variable Selection

Sometimes an explanatory variable X is required to be in the model, e.g. when it is the variable of interest.

In other situations, we are free to choose whichever X variables seem best.

Goal: Choose the simplest model (fewest explanatory variables) that still explains the data well.

When there is **collinearity** — strong linear relationships among the X variables — the effect of one X variable may *mask* the effect of another.

Suggests that we should consider adding or removing variables *one-at-a-time*.

Stepwise Procedures

forward selection: starting with none, add the "best" variable at each step, until the model stops improving

backward elimination: starting with all, remove the "worst" variable at each step, stopping before the model becomes inadequate

stepwise selection: perform forward selection, but also allow removal of variables (if appropriate) after each step

Notes:

- If possible, consider interaction terms, not just main effects.

  (An interaction can be in the model only if the corresponding lower-order interactions/main effects are.)

- When evaluating a categorical variable (or its interactions), all of its indicator variables must be added/removed together.

- P-values (e.g., from LRTs) are useful for defining the "best" or "worst" variable.

Akaike Information Criterion (AIC)

$$\text{AIC} \;=\; -2(\text{max log-likelihood} \;-\; \text{effective \# parameters})$$

Equivalent to an adjusted deviance:

$$D(\boldsymbol{y};\hat{\boldsymbol{\mu}}) \;-\; 2 \cdot \text{df}$$

where df is the (residual) degrees of freedom.

Note: First term penalizes lack of fit, second term penalizes complexity (too many variables).

Usage: Choose model with smallest AIC.

R Example: Horseshoe Crabs (Variable Selection)

```
> horseshoe <- read.table("horseshoe.txt", header=TRUE)

> head(horseshoe)
  color spine width satell weight y
1     3     3  28.3      8   3050 1
2     4     3  22.5      0   1550 0
3     2     1  26.0      9   2300 1
4     4     3  24.8      0   2100 0
5     4     3  26.0      4   2600 1
6     3     3  23.8      0   2100 0
```

We will fit binary logistic regressions with response `y`.

Variables `color` and `spine` will be treated as nominal.

Try an initial fit with main effects only:

```
> hsfit <- glm(y ~ weight + width + factor(color) + factor(spine),
+              family=binomial, data=horseshoe)

> summary(hsfit)

...

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)    -8.0650134  3.9285518  -2.053   0.0401 *
weight          0.0008258  0.0007038   1.173   0.2407
width           0.2631279  0.1952986   1.347   0.1779
factor(color)3 -0.1029024  0.7825912  -0.131   0.8954
factor(color)4 -0.4888642  0.8531183  -0.573   0.5666
factor(color)5 -1.6086658  0.9355326  -1.720   0.0855 .
factor(spine)2 -0.0959809  0.7033698  -0.136   0.8915
factor(spine)3  0.4002868  0.5027043   0.796   0.4259
---
...

    Null deviance: 225.76  on 172  degrees of freedom
Residual deviance: 185.20  on 165  degrees of freedom
AIC: 201.2
...
```

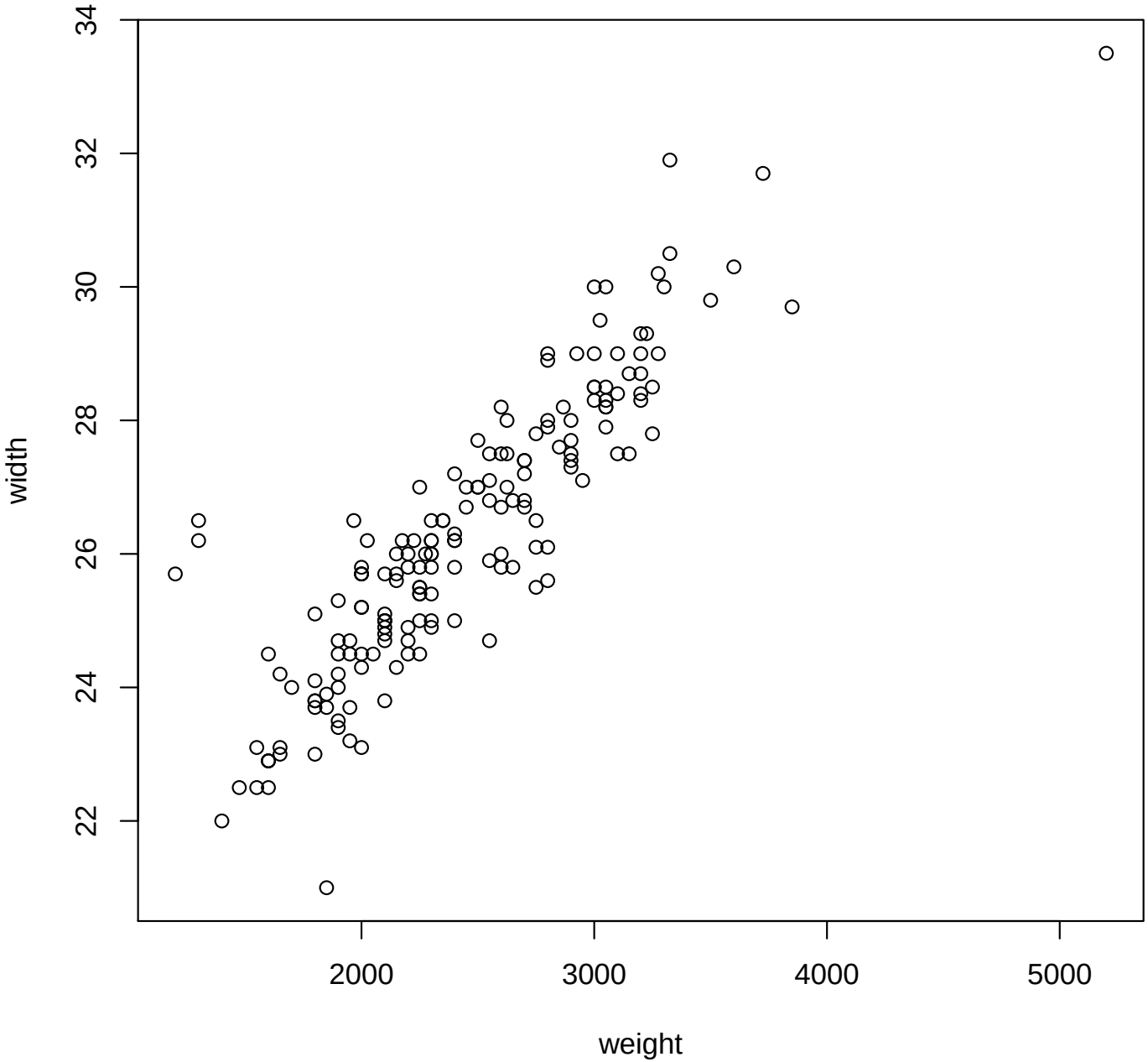
```
> drop1(hsfit, test="Chisq")
Single term deletions

Model:
y ~ weight + width + factor(color) + factor(spine)
              Df Deviance    AIC    LRT Pr(>Chi)
<none>             185.20 201.20
weight         1   186.61 200.61 1.4099  0.23507
width          1   187.00 201.00 1.7968  0.18010
factor(color)  3   192.80 202.80 7.5958  0.05515 .
factor(spine)  2   186.21 198.21 1.0091  0.60377
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Note: No significant terms (even though overall fit would be significant).

We might suspect collinearity between `weight` and `width`.

To check:

```
> with(horseshoe, cor(weight, width))
[1] 0.8868715

> with(horseshoe, plot(weight, width))
```


width
22
24
26
28
30
32
34
2000
3000
4000
5000
weight

Only one of `weight` and `width` may be needed.

For illustration, we will choose `width` ...

First try forward selection (starting from intercept only):

```
> nullmod <- glm(y ~ 1, family=binomial, data=horseshoe)

> formod <- step(nullmod, ~ width * factor(color) * factor(spine),
+                 direction="forward")
Start:  AIC=227.76
y ~ 1

                Df Deviance    AIC
+ width          1   194.45 198.45
+ factor(color)  3   212.06 220.06
<none>               225.76 227.76
+ factor(spine)  2   223.23 229.23

Step:  AIC=198.45
y ~ width

                Df Deviance    AIC
+ factor(color)  3   187.46 197.46
<none>               194.45 198.45
+ factor(spine)  2   194.43 202.43
```

```
Step:  AIC=197.46
y ~ width + factor(color)

                       Df Deviance    AIC
<none>                      187.46 197.46
+ width:factor(color)  3   183.08 199.08
+ factor(spine)        2   186.61 200.61
```

```
> summary(formod)

Call:
glm(formula = y ~ width + factor(color), family = binomial, data = horseshoe)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1124  -0.9848   0.5243   0.8513   2.1413

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -11.38519    2.87346  -3.962 7.43e-05 ***
width            0.46796    0.10554   4.434 9.26e-06 ***
factor(color)3   0.07242    0.73989   0.098    0.922
factor(color)4  -0.22380    0.77708  -0.288    0.773
factor(color)5  -1.32992    0.85252  -1.560    0.119
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 225.76  on 172  degrees of freedom
Residual deviance: 187.46  on 168  degrees of freedom
AIC: 197.46

Number of Fisher Scoring iterations: 4
```

```
> drop1(formod, test="Chisq")
Single term deletions

Model:
y ~ width + factor(color)
             Df Deviance    AIC     LRT  Pr(>Chi)
<none>            187.46 197.46
width         1   212.06 220.06 24.6038 7.041e-07 ***
factor(color) 3   194.45 198.45  6.9956   0.07204 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Note: `color` was selected by AIC, even though it does not reach the usual significance level.

It is easily possible that AIC will be smallest for a model that still has some insignificant terms.

Now try backward elimination:

We will start from a fully interacting model (all interactions considered, up to 3-way):

```
> fullmod <- glm(y ~ width * factor(color) * factor(spine), family=binomial,
+                data=horseshoe)
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Note the warning messages. This model for the data exhibits a type of problem we will examine later.

```
> backmod <- step(fullmod)  # backward elimination is the default
Start:  AIC=3433.49
y ~ width * factor(color) * factor(spine)

                                     Df Deviance    AIC
- width:factor(color):factor(spine)  4    173.7  209.7
<none>                                    3389.5 3433.5

Step:  AIC=209.67
y ~ width + factor(color) + factor(spine) + width:factor(color) +
    width:factor(spine) + factor(color):factor(spine)

                               Df Deviance    AIC
- factor(color):factor(spine)  6   181.56 205.56
- width:factor(spine)          2   173.68 205.68
- width:factor(color)          3   177.34 207.34
<none>                             173.67 209.67

Step:  AIC=205.56
y ~ width + factor(color) + factor(spine) + width:factor(color) +
    width:factor(spine)

                       Df Deviance    AIC
- width:factor(spine)  2   181.64 201.64
- width:factor(color)  3   186.41 204.41
<none>                     181.56 205.56
```

```
Step:  AIC=201.64
y ~ width + factor(color) + factor(spine) + width:factor(color)

                      Df Deviance    AIC
- factor(spine)        2   183.08 199.08
- width:factor(color)  3   186.61 200.61
<none>                     181.64 201.64

Step:  AIC=199.08
y ~ width + factor(color) + width:factor(color)

                      Df Deviance    AIC
- width:factor(color)  3   187.46 197.46
<none>                     183.08 199.08

Step:  AIC=197.46
y ~ width + factor(color)

                Df Deviance    AIC
<none>               187.46 197.46
- factor(color)  3   194.45 198.45
- width          1   212.06 220.06
```

So, in this case, backward elimination by AIC produced the same final model as forward selection:

```
> summary(backmod)

Call:
glm(formula = y ~ width + factor(color), family = binomial, data = horseshoe)

...
```

Finally, try stepwise selection, starting from the intercept-only model and allowing up to 3-way interactions:

```
> stepmod <- step(nullmod, ~ width * factor(color) * factor(spine),
+                 direction="both")
Start:  AIC=227.76
y ~ 1

                Df Deviance    AIC
+ width          1   194.45 198.45
+ factor(color)  3   212.06 220.06
<none>               225.76 227.76
+ factor(spine)  2   223.23 229.23

Step:  AIC=198.45
y ~ width

                Df Deviance    AIC
+ factor(color)  3   187.46 197.46
<none>               194.45 198.45
+ factor(spine)  2   194.43 202.43
- width          1   225.76 227.76
```

```
Step:  AIC=197.46
y ~ width + factor(color)

                      Df Deviance    AIC
<none>                     187.46 197.46
- factor(color)        3   194.45 198.45
+ width:factor(color)  3   183.08 199.08
+ factor(spine)        2   186.61 200.61
- width                1   212.06 220.06
```

So, in this case, stepwise selection by AIC produced the same final model as forward selection and backward elimination:

```
> summary(stepmod)

Call:
glm(formula = y ~ width + factor(color), family = binomial, data = horseshoe)

...
```

Diagnostics

We have already seen general goodness-of-fit tests, but there are fit-related questions that concern individual observations:

Does the model fit *all* of the observations well?

Is the fit especially sensitive to certain observations?

Residuals

- Pearson:

$$e_i = \frac{y_i - n_i\hat{\pi}_i}{\sqrt{n_i\hat{\pi}_i(1-\hat{\pi}_i)}}$$

  The sum of their squares is a kind of generalized Pearson X^2 statistic.

Residuals

- Pearson:

$$e_i = \frac{y_i - n_i\hat{\pi}_i}{\sqrt{n_i\hat{\pi}_i(1-\hat{\pi}_i)}}$$

  The sum of their squares is a kind of generalized Pearson X^2 statistic.

- Deviance:

$$\text{sign}(y_i - n_i\hat{\pi}_i)\sqrt{d_i}$$

  where

$$d_i = 2\left(y_i \ln\frac{y_i}{n_i\hat{\pi}_i} + (n_i - y_i)\ln\frac{n_i - y_i}{n_i - n_i\hat{\pi}_i}\right)$$

  The sum of their squares is the deviance (G^2).

- Standardized:

$$r_i = \frac{e_i}{\sqrt{1-\hat{h}_i}}$$

where $\hat{h}_i$ is the ith leverage (from the hat matrix)

Idea: Divide the raw residual by (approximately) its standard deviation.

Advantage: Closer to $N(0,1)$, when model fits.

- Standardized:

$$r_i = \frac{e_i}{\sqrt{1-\hat{h}_i}}$$

where $\hat{h}_i$ is the ith leverage (from the hat matrix)

Idea: Divide the raw residual by (approximately) its standard deviation.

Advantage: Closer to $N(0,1)$, when model fits.

Usage: Check for unusual observations (outliers) or trends (when plotted versus unused variables).

Warning: For ungrouped data ($Y = 0$ or 1), residuals tend not to be very useful.

Influence

- Leverages $\hat{h}_i$ measure **potential influence**.

  Outliers having high leverage are particularly influential.

  (Remark: Unlike in linear regression, observations outlying in the X variables do not necessarily have high leverage.)

Influence

- Leverages $\hat{h}_i$ measure **potential influence**.

  Outliers having high leverage are particularly influential.

  (Remark: Unlike in linear regression, observations outlying in the X variables do not necessarily have high leverage.)

- **Cook's Distance**: for each observation, an overall measure of change in $\hat{\boldsymbol{\beta}}$ when that observation is removed

  Generally indicates a problem if > 1.

  (Agresti instead uses a quantity c that is larger by a constant factor.)

- **Dfbeta**: for each parameter and each observation, the (standardized) change in the parameter estimate when the observation is removed

  Indicates a problem if its *magnitude* is large, e.g. > 2 or > 3, since standardized. (But even just > 1 could indicate undue influence.)

R Example: Heart Disease/BP (Diagnostics)

```
> hdbp <- data.frame(cases = c(3, 17, 12, 16, 12, 8, 16, 8),
+                    total = c(156, 252, 284, 271, 139, 85, 99, 43),
+                    bloodpressure = factor(c("<117","117-126","127-136",
+                                             "137-146","147-156","157-166",
+                                             "167-186",">186")))

> hdbp
  cases total bloodpressure
1     3   156          <117
2    17   252       117-126
3    12   284       127-136
4    16   271       137-146
5    12   139       147-156
6     8    85       157-166
7    16    99       167-186
8     8    43          >186
```

Is there a relationship between probability of heart disease and (systolic) blood pressure?

First fit an intercept-only model:

```
> nullmod <- glm(cbind(cases,total-cases) ~ 1, family=binomial, data=hdbp)

> 1 - pchisq(deviance(nullmod), df.residual(nullmod))
[1] 9.405884e-05
```

Lack of fit is apparent.

```
> rstandard(nullmod, type="pearson")
         1          2          3          4          5          6          7
-2.6184346 -0.1225923 -2.0193620 -0.7402622  0.8396338  0.9345002  3.7644737
         8
 3.0679293
```

There are some large-magnitude standardized residuals. They also seem to show an increasing trend.

Now fit a linear logit model, using scores:

```
> bpscore <- c(111.5, 121.5, 131.5, 141.5, 151.5, 161.5, 176.5, 191.5)

> llmod <- glm(cbind(cases,total-cases) ~ bpscore, family=binomial, data=hdbp)

> 1 - pchisq(deviance(llmod), df.residual(llmod))
[1] 0.4334429
```

Model fit is adequate.

```
> rstandard(llmod, type="pearson")
         1          2          3          4          5          6          7
-1.1057850  2.3746058 -0.9452701 -0.5727440  0.1260886 -0.3260730  0.6519547
         8
-0.1773473
```

One residual has moderately large magnitude, but there is no residual trend.

Now examine Cook's distances:

```
> cooks.distance(llmod)
          1           2           3           4           5           6
0.167920407 1.132466593 0.156701250 0.045520798 0.001191592 0.007901458
          7           8
0.130092239 0.009703537
```

Second observation has possibly a high influence.

Now examine Dfbetas:

```
> dfbetas(llmod)
    (Intercept)      bpscore
1 -0.6107162077  0.564352232
2  2.4998699972 -2.236644148
3 -0.4088707941  0.341660170
4 -0.1201954685  0.078811128
5 -0.0004280323  0.007170298
6  0.0453945193 -0.061375625
7 -0.3270351788  0.375943804
8  0.1045444927 -0.114799503
```

Second observation appears to have high influence on both parameters.

(Remark: Influence measures in Agresti differ slightly from these.)

Predictive Power

Want to evaluate

- strength of association (like R^2 in linear regression)
- performance of model as a classifier

Correlation Measure

$$R(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) \quad = \quad \text{sample correlation between } y_i\text{s and fitted values } \hat{\mu}_i$$

Agresti assumes data are in *ungrouped* format:

$$y_i \quad = \quad 0 \text{ or } 1 \qquad\qquad \hat{\mu}_i \quad = \quad \hat{\pi}_i$$

Note: Similar to $\sqrt{R^2}$ in linear regression, where R^2 is the coefficient of determination.

Likelihood Measures

Consider a model M (that has an intercept).

Model	Maximum log-likelihood
intercept-only	L_0
M	L_M
saturated	L_S

(so $L_0 \leq L_M \leq L_S$)

Then use

$$\frac{L_M - L_0}{L_S - L_0} \quad \in \quad [0, 1]$$

with larger values indicating M better relative to intercept-only model.

Note: This likelihood measure may depend on the definition of the saturated model.

Recall: The saturated models are generally different for ungrouped (binary) and grouped (binomial) formats of the data.

Call the likelihood measure

D for the ungrouped (binary) format

D^* for the grouped (binomial) format

Agresti recommends D over D^*.

Classification Tables

A logistic regression fit

$$\hat{\pi}(\boldsymbol{x}) \;\; = \;\; \text{estimate of } \; \mathrm{P}(\text{success} \mid \boldsymbol{x})$$

can be used as a **classifier**:

$$\hat{y} \;\; = \;\; \begin{cases} 1 \text{ (success)} & \text{if } \; \hat{\pi}(\boldsymbol{x}) > \pi_0 \\ 0 \text{ (failure)} & \text{if } \; \hat{\pi}(\boldsymbol{x}) \leq \pi_0 \end{cases}$$

for some cutoff π_0.

Could take $\pi_0 = 0.5$, or perhaps $\pi_0 =$ the observed fraction of successes.

A **classification table** is a 2×2 contingency table of actual (binary) response y versus classified $\hat{y}$:

		Prediction	
		$\hat{Y} = 1$	$\hat{Y} = 0$
Actual	$Y = 1$		
	$Y = 0$		

Can be used to estimate

$$\begin{aligned} \textbf{sensitivity} &= \mathrm{P}(\hat{Y} = 1 \mid Y = 1) \\ \textbf{specificity} &= \mathrm{P}(\hat{Y} = 0 \mid Y = 0) \end{aligned}$$

Q: What happens to sensitivity and specificity as π_0 increases?

The proportion correct is

$$\mathrm{P}(Y = 1, \hat{Y} = 1) \; + \; \mathrm{P}(Y = 0, \hat{Y} = 0)$$

and the error rate is its complement:

$$\mathrm{P}(Y = 1, \hat{Y} = 0) \; + \; \mathrm{P}(Y = 0, \hat{Y} = 1)$$

Note: These depend on the *marginal* distribution of Y (unlike sensitivity and specificity).

Therefore, they cannot be estimated from *retrospectively* sampled data alone.

Problem: If same data is used to fit the model and to construct the table, performance estimates can be too optimistic.

Remedy: Use **leave-one-out cross validation**, in which each observation is classified according to a model fit without it.

Then build the table using the cross-validated predictions.

Receiver Operating Characteristic (ROC) Curves

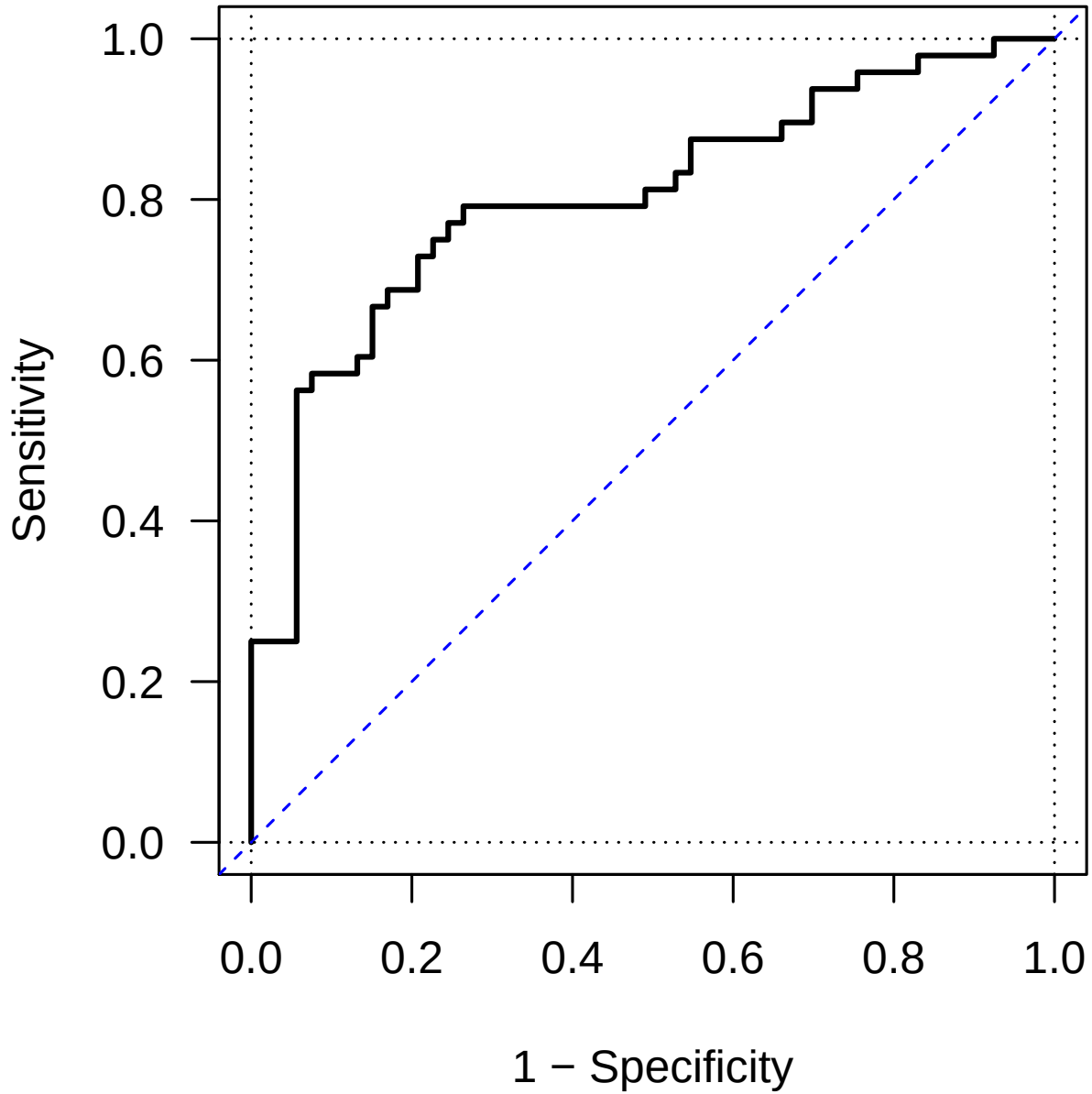
Plot

$$\text{sensitivity} \qquad \text{versus} \qquad 1 - \text{specificity}$$

for all possible cutoffs π_0.

(Usually estimated from data.)

Idea: Want to evaluate classification performance of a model without having to choose a cutoff π_0.


Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1 − Specificity

Note: ROC curve

- is non-decreasing
- goes from $(0, 0)$ to $(1, 1)$
- is often above the 45° line
- indicates a better classifier if it is higher

The 45° line represents random guessing.

The area under the ROC curve, a.k.a. the "concordance index," measures predictive power.

See Agresti, Sec. 6.3.4, for more information.

(Remark: Agresti's definition of concordance index is slightly incorrect: Pairs that have the same estimated probability $\hat{\pi}$ should be half-counted.)

R Example: Horseshoe Crabs (Predictive Power)

```
> horseshoe <- read.table("horseshoe.txt", header=TRUE)
```

Recall that y is binary — data are already ungrouped.

Compute correlation measures for three models:

```
> mod1 <- glm(y ~ width, family=binomial, data=horseshoe)

> cor(horseshoe$y, fitted(mod1))
[1] 0.4019782

> mod2 <- glm(y ~ factor(color), family=binomial, data=horseshoe)

> cor(horseshoe$y, fitted(mod2))
[1] 0.2852595

> mod3 <- glm(y ~ width + factor(color), family=binomial, data=horseshoe)

> cor(horseshoe$y, fitted(mod3))
[1] 0.4522131
```

Compute an apparent classification table for the third model:

```
> pi0 <- 0.5

> table(y=horseshoe$y, yhat=as.numeric(fitted(mod3) > pi0))
   yhat
y    0  1
  0 31 31
  1 15 96
```

Note the ordering of the rows and columns!

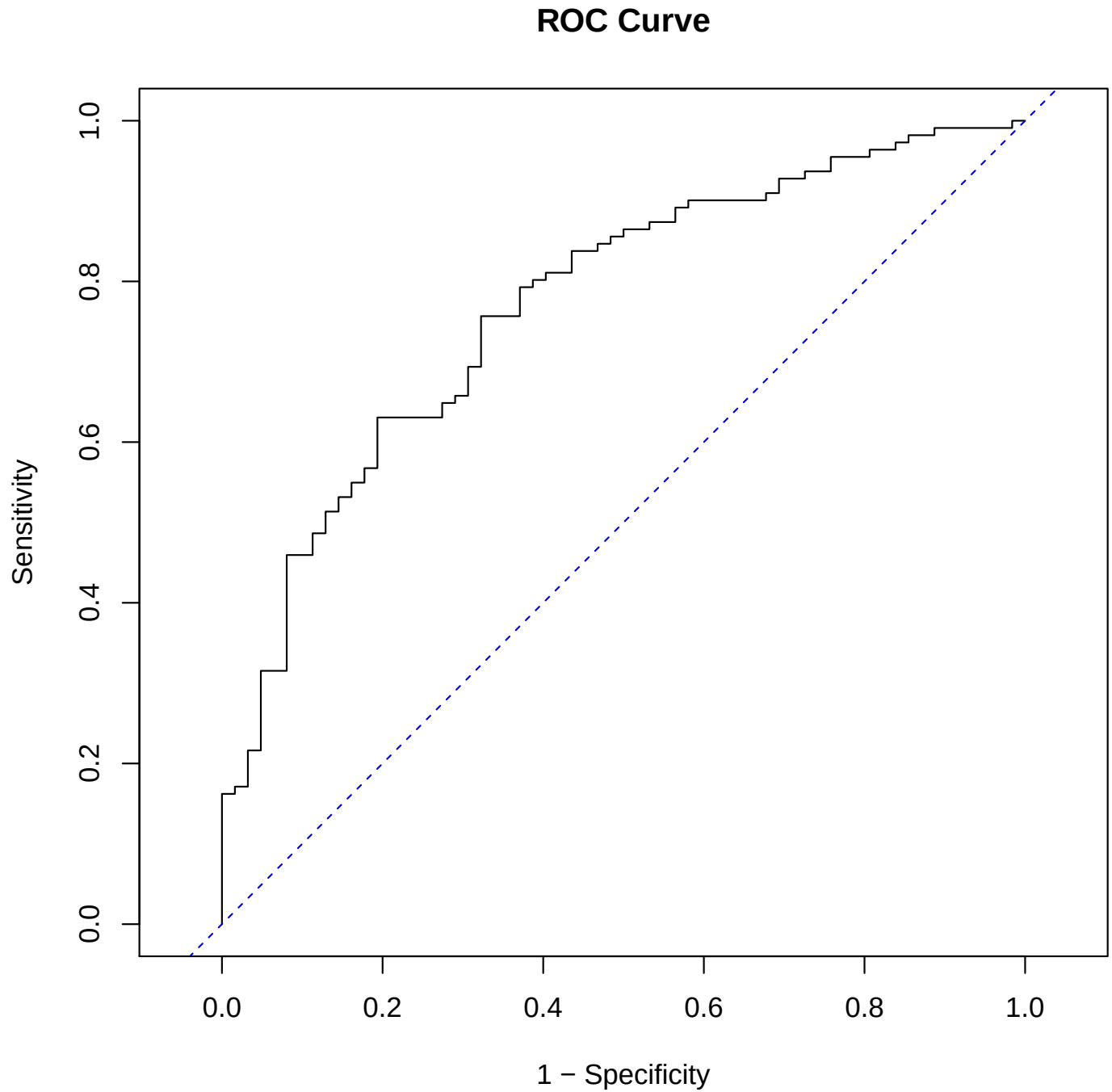
```
> 96 / (15 + 96)  # apparent sensitivity
[1] 0.8648649

> 31 / (31 + 31)  # apparent specificity
[1] 0.5

> (31 + 96) / (31 + 31 + 15 + 96)  # apparent prop. correct
[1] 0.734104
```

Now compute a cross-validated table and statistics:

```
> pihatcv <- numeric(nrow(horseshoe))

> for(i in 1:nrow(horseshoe))
+   pihatcv[i] <- predict(update(mod3, subset=-i), newdata=horseshoe[i,],
+                         type="response")

> table(y=horseshoe$y, yhat=as.numeric(pihatcv > pi0))
   yhat
y    0  1
  0 28 34
  1 17 94

> 94 / (94 + 17)  # cross-validated sensitivity
[1] 0.8468468

> 28 / (28 + 34)  # cross-validated specificity
[1] 0.4516129

> (28 + 94) / (28 + 34 + 17 + 94)  # cross-validated prop. correct
[1] 0.7052023
```

Construct and plot an ROC curve (for the third model):

```
> n <- nrow(horseshoe)

> pihat <- fitted(mod3)

> true.pos <- c(0,cumsum(horseshoe$y[order(pihat, decreasing=TRUE)]))

> false.pos <- 0:n - true.pos

> plot(false.pos/false.pos[n+1], true.pos/true.pos[n+1], type="l",
+      main="ROC Curve", xlab="1 - Specificity", ylab="Sensitivity", asp=1)
> abline(a=0, b=1, lty=2, col="blue")
```


ROC Curve
Sensitivity
1 − Specificity
0.0
0.2
0.4
0.6
0.8
1.0

Area under the curve (concordance index):

```
> mean(outer(pihat[horseshoe$y==1], pihat[horseshoe$y==0], ">") +
+      0.5 * outer(pihat[horseshoe$y==1], pihat[horseshoe$y==0], "=="))
[1] 0.7713601
```

Mantel-Haenszel Tests

Consider an X-Y relationship as in a 2×2 table:

e.g. $X =$ exposure (yes, no), $Y =$ disease presence

e.g. $X =$ treatment (A or B), $Y =$ whether successful

Let there be a categorical **stratification variable** Z that may mediate the X-Y relationship:

e.g. $Z =$ age range

e.g. $Z =$ treatment clinic

Z is possibly a confounding variable, for which we want to adjust when assessing the X-Y relationship.

If Z has K categories (**strata**), the data form a $2 \times 2 \times K$ contingency table.

(Each stratum has its own 2×2 partial table.)

Main question: Are X and Y conditionally independent given Z?

If not, what is their conditional association (odds ratio)?

(Also: Consider testing for homogeneous association.)

Model-Based: Logistic Regression

$$\text{logit}(\pi_{ik}) \;=\; \alpha \,+\, \beta x_i \,+\, \beta_k^Z \qquad i = 1, 2 \quad k = 1, \ldots, K$$

$$\pi_{ik} \;=\; \text{P}(Y = 1 \mid X = i, Z = k) \qquad x_i \;=\; \begin{cases} 1 & i = 1 \\ 0 & i = 2 \end{cases}$$

(For identifiability, set, e.g., $\beta_1^Z \equiv 0$).

Can verify that this model assumes homogeneous XY association over Z.

Testing conditional independence is the same as testing

$$H_0 : \beta = 0$$

which can be done with Wald, likelihood-ratio, or score.

Testing conditional independence is the same as testing

$$H_0 : \beta = 0$$

which can be done with Wald, likelihood-ratio, or score.

The common conditional odds ratio is

$$\theta_{XY(k)} = e^{\beta} \qquad \text{(all } k\text{)}$$

for which the MLE is

$$e^{\hat{\beta}}$$

Can also form a CI (with Wald or profile likelihood).

With *grouped* data, the assumption of homogeneous XY association may be tested by comparison with the saturated model (e.g., using it as the full model in a LRT).

(If homogeneous association is rejected, it may be worthwhile to investigate how XY association varies between strata.)

Non-Model-Based: CMH

Consider the kth partial table:

$Z = k$:	$Y = 1$	$Y = 0$	
$X = 1$	n_{11k}	n_{12k}	n_{1+k}
$X = 0$	n_{21k}	n_{22k}	n_{2+k}
	n_{+1k}	n_{+2k}	n_{++k}

so that

$$n_{ijk} \quad = \quad \text{count in row } i \text{ \& col } j \text{ of table } k$$

Suppose there is no XY association when $Z = k$.

Then conditioning on the kth partial table's marginal totals (as in Fisher's exact test) yields

$$\mu_{11k} = E(N_{11k}) = \frac{n_{1+k}\, n_{+1k}}{n_{++k}}$$

$$\mathrm{var}(N_{11k}) = \frac{n_{1+k}\, n_{2+k}\, n_{+1k}\, n_{+2k}}{n_{++k}^2\, (n_{++k} - 1)}$$

based on the hypergeometric distribution.

Define the (**Cochran-**)**Mantel-Haenszel statistic**

$$\text{CMH} \quad = \quad \frac{\left(\sum_k (n_{11k} - \mu_{11k})\right)^2}{\sum_k \text{var}(N_{11k})}$$

Under

$$H_0 : \text{ conditional } X\text{-}Y \text{ independence}$$

for large samples,

$$\text{CMH} \quad \dot{\sim} \quad \chi^2_1$$

Note: If $\theta_{XY(k)} > 1$ then $n_{11k} - \mu_{11k}$ tends to be positive.

If this is true for all k, CMH will tend to be larger.

(Likewise if $\theta_{XY(k)} < 1$ for all k.)

Thus reject

$$H_0 : \text{ conditional } X\text{-}Y \text{ independence}$$

if

$$\mathrm{CMH} \quad > \quad \chi^2_1(\alpha)$$

Asymptotics: See Agresti, Sec. 6.4.4.

For estimating the (common) conditional odds ratio, Mantel & Haenszel proposed

$$\hat{\theta}_{\mathrm{MH}} = \frac{\sum_k n_{11k}\, n_{22k}/n_{++k}}{\sum_k n_{12k}\, n_{21k}/n_{++k}}$$

(Q: What is this if $K = 1$?)

See Agresti (Sec. 6.4.5) for an asymptotic variance estimate for $\ln(\hat{\theta}_{\mathrm{MH}})$.

Remark: Even if homogeneous XY association is not exactly correct, the CMH test and $\hat{\theta}_{\mathrm{MH}}$ still serve a summaries of association, especially if all true associations have the same direction.

This allows use in meta-analysis — see Agresti, Sec. 6.4.6.

Remark: In the model

$$\text{logit}(\pi_{ik}) \;=\; \alpha \,+\, \beta x_i \,+\, \beta_k^Z$$

the condition

$$\beta_k^Z \;=\; 0 \qquad \text{all } k$$

is equivalent to conditional independence of Y and Z given X.

In that case, the conditional and marginal X-Y odds ratios are the same.

R Example: Multi-Center Trial

From a clinical trial of a drug for curing infection, conducted at 8 different centers:

```
> infectioncure <- read.table("infectioncure.txt", header=TRUE)

> head(infectioncure)
  center treat response count
1      a     1        1    11
2      a     1        2    25
3      a     2        1    10
4      a     2        2    27
5      b     1        1    16
6      b     1        2     4
```

`treat` is 1 for Drug, 2 for Control

`response` is 1 for Success, 2 for Failure

```
> ic <- reshape(infectioncure, varying=list(c("success","failure")),
+               v.names="count", timevar="response", idvar=c("center","treat"),
+               direction="wide")

> ic
   center treat success failure
1       a     1      11      25
3       a     2      10      27
5       b     1      16       4
7       b     2      22      10
9       c     1      14       5
11      c     2       7      12
13      d     1       2      14
15      d     2       1      16
17      e     1       6      11
19      e     2       0      12
21      f     1       1      10
23      f     2       0      10
25      g     1       1       4
27      g     2       1       8
29      h     1       4       2
31      h     2       6       1
```

Estimate odds ratio for each center separately:

```
> for(k in unique(ic$center))
+   with(ic[ic$center==k,], print(success[treat==1] * failure[treat==2] /
+                                 (success[treat==2] * failure[treat==1])))
[1] 1.188
[1] 1.818182
[1] 4.8
[1] 2.285714
[1] Inf
[1] Inf
[1] 2
[1] 0.3333333
```

Most are in the same direction (> 1).

```
> Treatment <- factor(c("Drug","Control")[ic$treat])

> mod <- glm(cbind(success,failure) ~ Treatment + center, family=binomial,
+            data=ic)

> summary(mod)

...

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.3220     0.3165  -4.177 2.95e-05 ***
TreatmentDrug  0.7769     0.3067   2.533  0.01130 *
centerb        2.0554     0.4201   4.893 9.94e-07 ***
centerc        1.1529     0.4246   2.715  0.00662 **
centerd       -1.4185     0.6636  -2.138  0.03255 *
centere       -0.5199     0.5338  -0.974  0.33007
centerf       -2.1469     1.0614  -2.023  0.04310 *
centerg       -0.7977     0.8149  -0.979  0.32764
centerh        2.2079     0.7195   3.069  0.00215 **

...
```

```
> drop1(mod, test="Chisq")
Single term deletions

Model:
cbind(success, failure) ~ Treatment + center
          Df Deviance     AIC    LRT  Pr(>Chi)
<none>           9.746  66.136
Treatment  1   16.415  70.805  6.669  0.009811 **
center     7   90.960 133.350 81.214 7.788e-15 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> exp(0.7769)  # MLE of common conditional odds ratio
[1] 2.17472

> exp(0.7769 + c(-1,1) * 1.96 * 0.3067)  # transformed Wald interval
[1] 1.192161 3.967087

> exp(confint(mod)["TreatmentDrug",])  # profile likelihood interval
Waiting for profiling to be done...
   2.5 %   97.5 %
1.203376 4.021155
```

A G^2 test of homogeneous association (goodness of fit):

```
> deviance(mod)
[1] 9.746317
> df.residual(mod)
[1] 7
> 1 - pchisq(deviance(mod), df.residual(mod))
[1] 0.203411
```

A X^2 test of homogeneous association (goodness of fit):

```
> ( X2 <- sum(residuals(mod, type="pearson")^2) )
[1] 8.025646
> 1 - pchisq(X2, df.residual(mod))
[1] 0.3303379
```

(However, we might doubt these tests because of several small expected counts.)

Now try the Cochran-Mantel-Haenszel approach ...

First make an array of partial tables:

```
> ic.array <- xtabs(count ~ treat + response + center, data=infectioncure)

> ic.array
, , center = a

     response
treat  1  2
    1 11 25
    2 10 27

, , center = b

     response
treat  1  2
    1 16  4
    2 22 10
```

```
, , center = c

     response
treat  1  2
    1 14  5
    2  7 12

, , center = d

     response
treat  1  2
    1  2 14
    2  1 16

, , center = e

     response
treat  1  2
    1  6 11
    2  0 12
```

```
, , center = f

     response
treat  1  2
    1  1 10
    2  0 10

, , center = g

     response
treat  1  2
    1  1  4
    2  1  8

, , center = h

     response
treat  1  2
    1  4  2
    2  6  1
```

```
> mantelhaen.test(ic.array, correct=FALSE)

Mantel-Haenszel chi-squared test without continuity correction

data:  ic.array
Mantel-Haenszel X-squared = 6.3841, df = 1, p-value = 0.01151
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
 1.177590 3.869174
sample estimates:
common odds ratio
         2.134549
```

Remark: If there are very few observations, the χ^2 approximation in the CMH test may be unreliable.

An alternative is an *exact conditional test* — see Agresti, Sec. 7.3.5.

(Such tests generalize Fisher's exact test.)

When the MLE Does Not Exist

The logistic regression MLE is unique if it exists (since the log-likelihood is strictly concave when $\boldsymbol{X}$ has full rank).

But it doesn't always exist ...

For simplicity, assume binary Y (ungrouped data), so that

$$y_i \;=\; 0 \text{ or } 1$$

Let

$$\boldsymbol{x}_i^T \;=\; i\text{th row of } \boldsymbol{X}$$

Technically, the logistic regression MLE exists and is unique if and only if there does *not* exist $\boldsymbol{b} \neq \mathbf{0}$ such that, for all i,

$$\boldsymbol{b}^T \boldsymbol{x}_i > 0 \quad \Rightarrow \quad y_i = 1$$

$$\boldsymbol{b}^T \boldsymbol{x}_i < 0 \quad \Rightarrow \quad y_i = 0$$

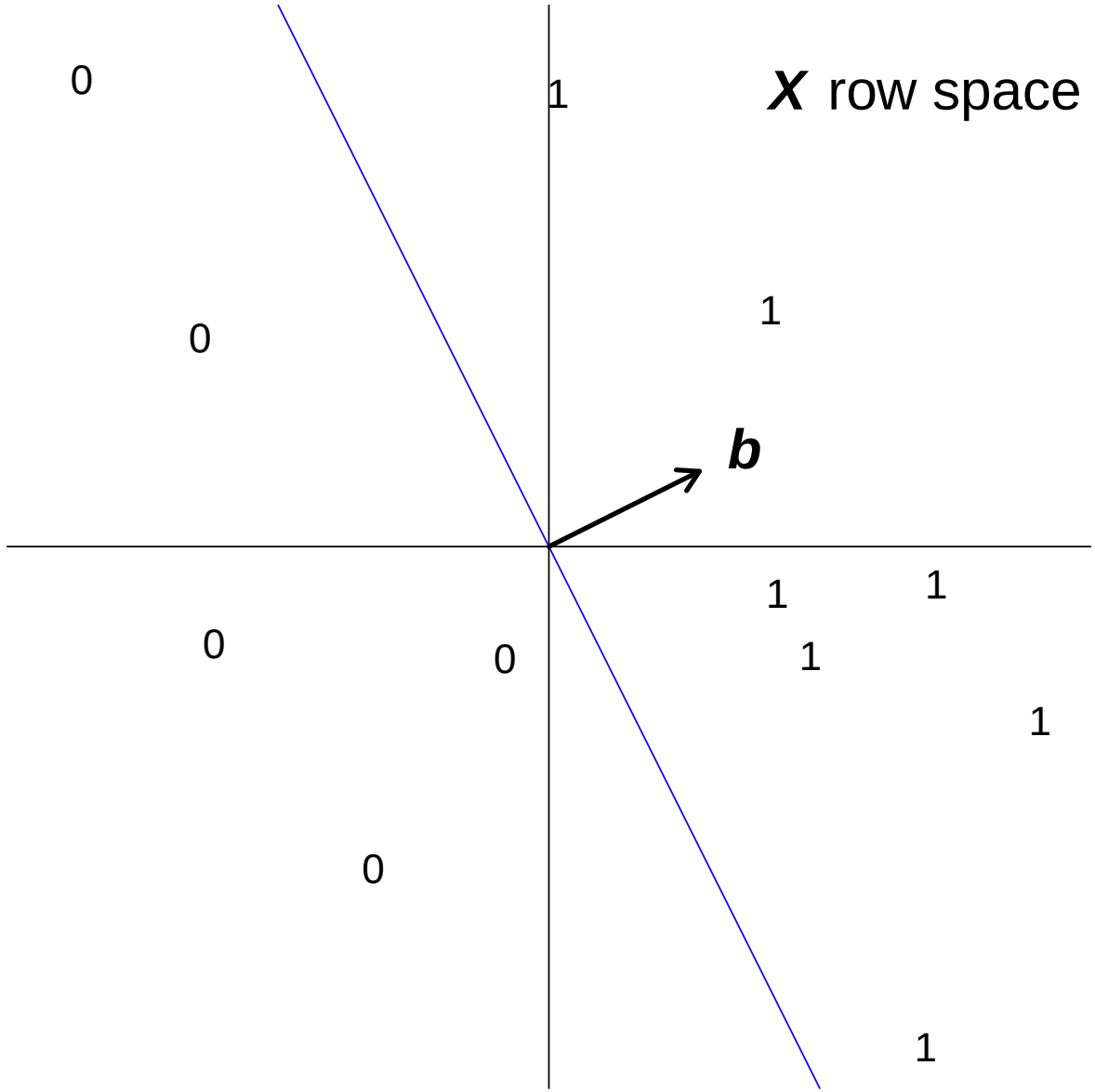
If $\boldsymbol{X}$ has full rank, then existence of the $\boldsymbol{b}$ above implies that an MLE *does not exist*. There is no maximum.

One special case when the MLE does not exist is **complete separation**: There exists $\boldsymbol{b}$ such that

$$y_i = 1 \quad \Rightarrow \quad \boldsymbol{b}^T \boldsymbol{x}_i > 0$$

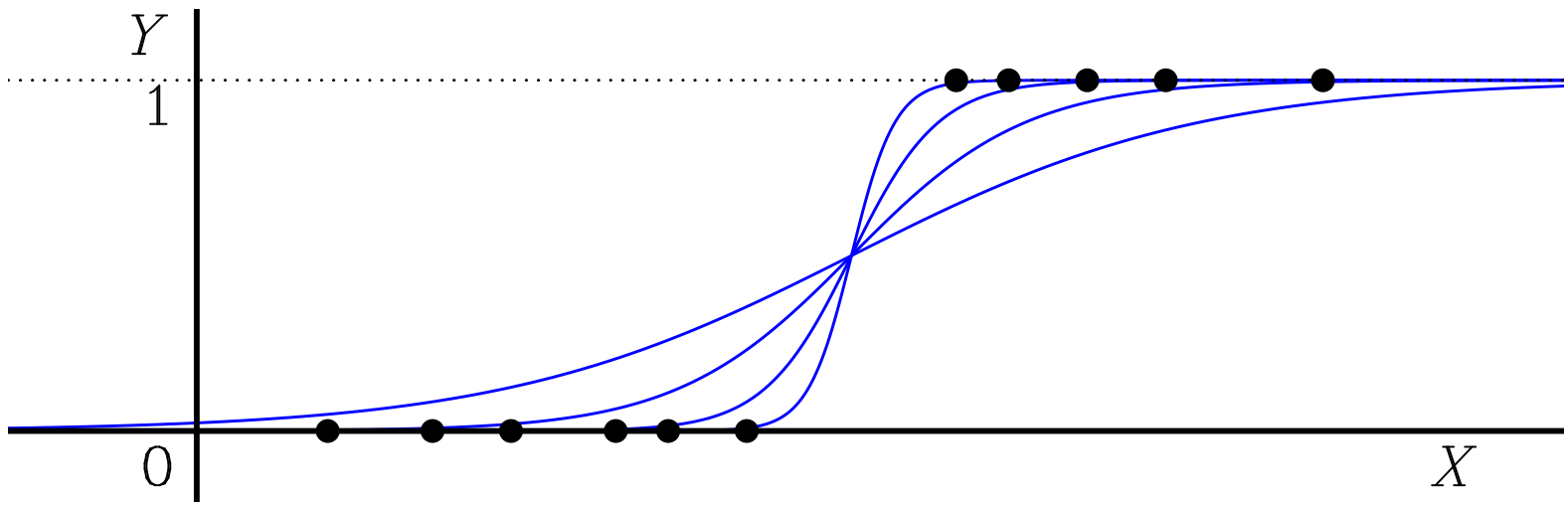
$$y_i = 0 \quad \Rightarrow \quad \boldsymbol{b}^T \boldsymbol{x}_i < 0$$

i.e., a subspace separates the rows of $\boldsymbol{X}$ with $Y = 1$ from those with $Y = 0$ ...


0
1
X row space
0
1
b
1
1
0
0
1
1
0
1
0

The MLE fails to exist because the likelihood keeps increasing as one or more parameters become infinite.

For example ($p = 1$):

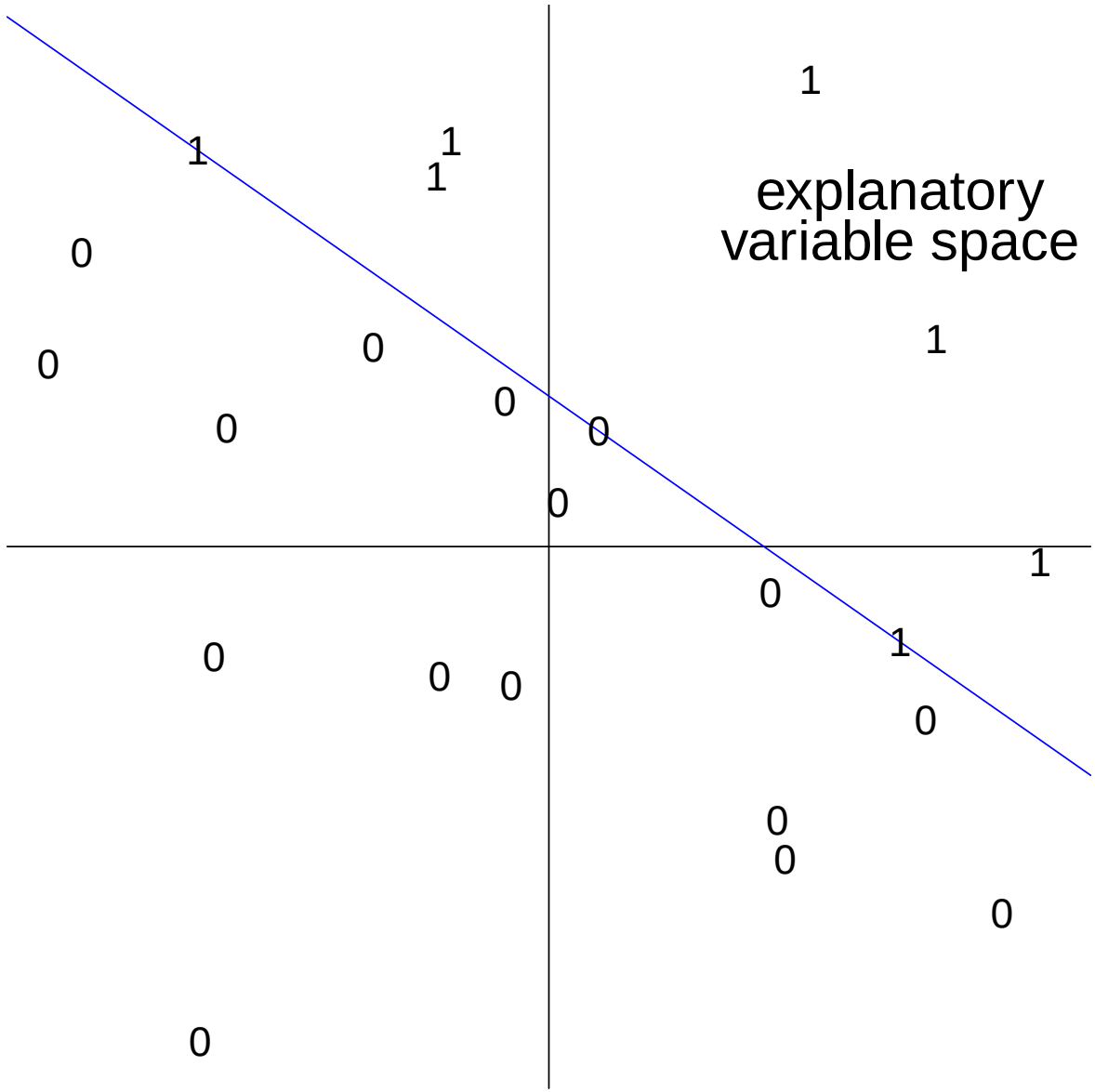


The fit continues to improve as the slope goes to infinity.

Even when complete separation does not hold, the MLE may still fail to exist.

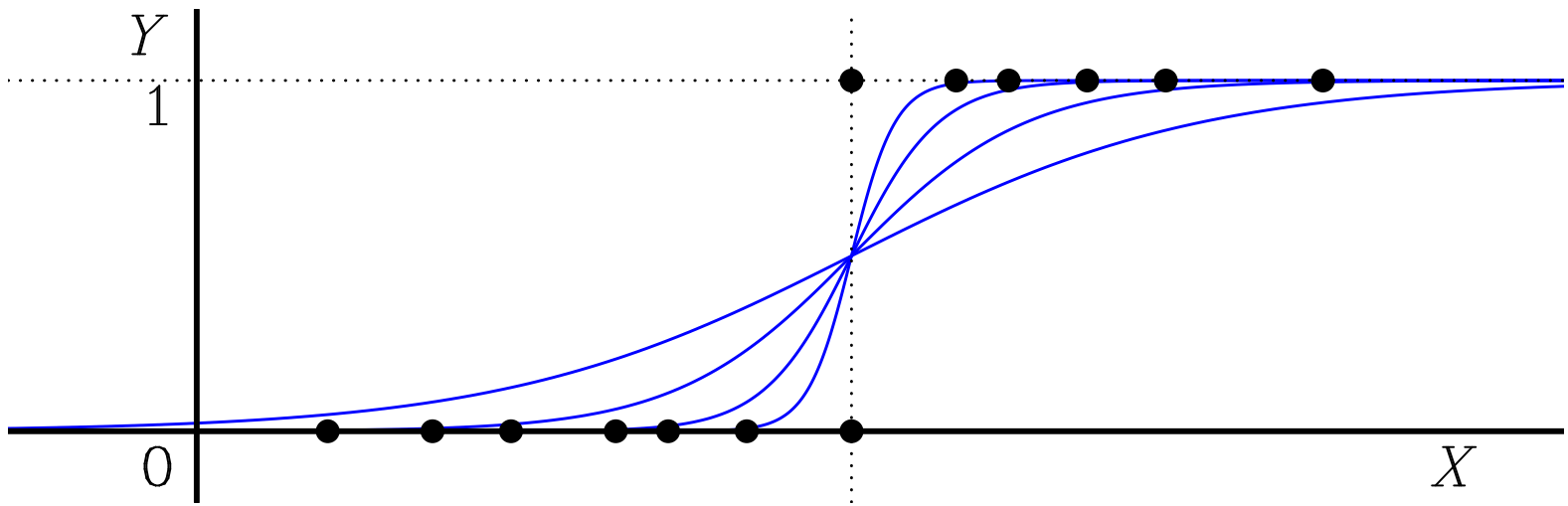
Assuming the logistic regression has an intercept, **quasi-complete separation** is when a *hyperplane* in *explanatory variable space* separates $Y = 0$ and $Y = 1$ cases, except that both cases exist on the hyperplane itself.

An MLE fails to exist under quasi-complete separation.


explanatory
variable space

As with complete separation, quasi-complete separation allows the likelihood to keep increasing as one or more parameters become infinite.

For example ($p = 1$):



One common situation can cause quasi-complete separation:

One category of a nominal explanatory variable has only $Y = 0$ cases or only $Y = 1$ cases.

For example, suppose X is type of treatment and $Y = 1$ means cured. Then quasi-complete separation usually occurs if one treatment type cures (or fails to cure) all of its cases.

Signs the MLE may not exist:

- numerical estimates have large magnitude
- standard errors are huge (z-values near 0, P-values near 1)
- some fitted values ($\hat{\pi}(\boldsymbol{x})$) are almost exactly 0 or 1
- unusually large iteration count (or non-convergence warning)

Possible remedies:

- Use likelihood-ratio or score inference (not Wald)
- Use alternative estimation techniques (e.g., penalized likelihood, Bayes)

R Example: Incontinence Data

From Agresti, Exercise 6.20: Incontinence ($Y = 1$ for yes) to be explained by three lower urinary tract variables —

```
> incontinence <- read.table("incontinence.txt", header=TRUE)

> head(incontinence)
  y   x1   x2  x3
1 0 -1.9 -5.3 -43
2 0 -1.5  3.9 -15
3 0 -0.1 -5.2 -32
4 0  0.5 27.5   8
5 0  0.8 -3.0 -12
6 0  0.8 -1.6  -2
```

Try fitting a model with a linear term in each variable:

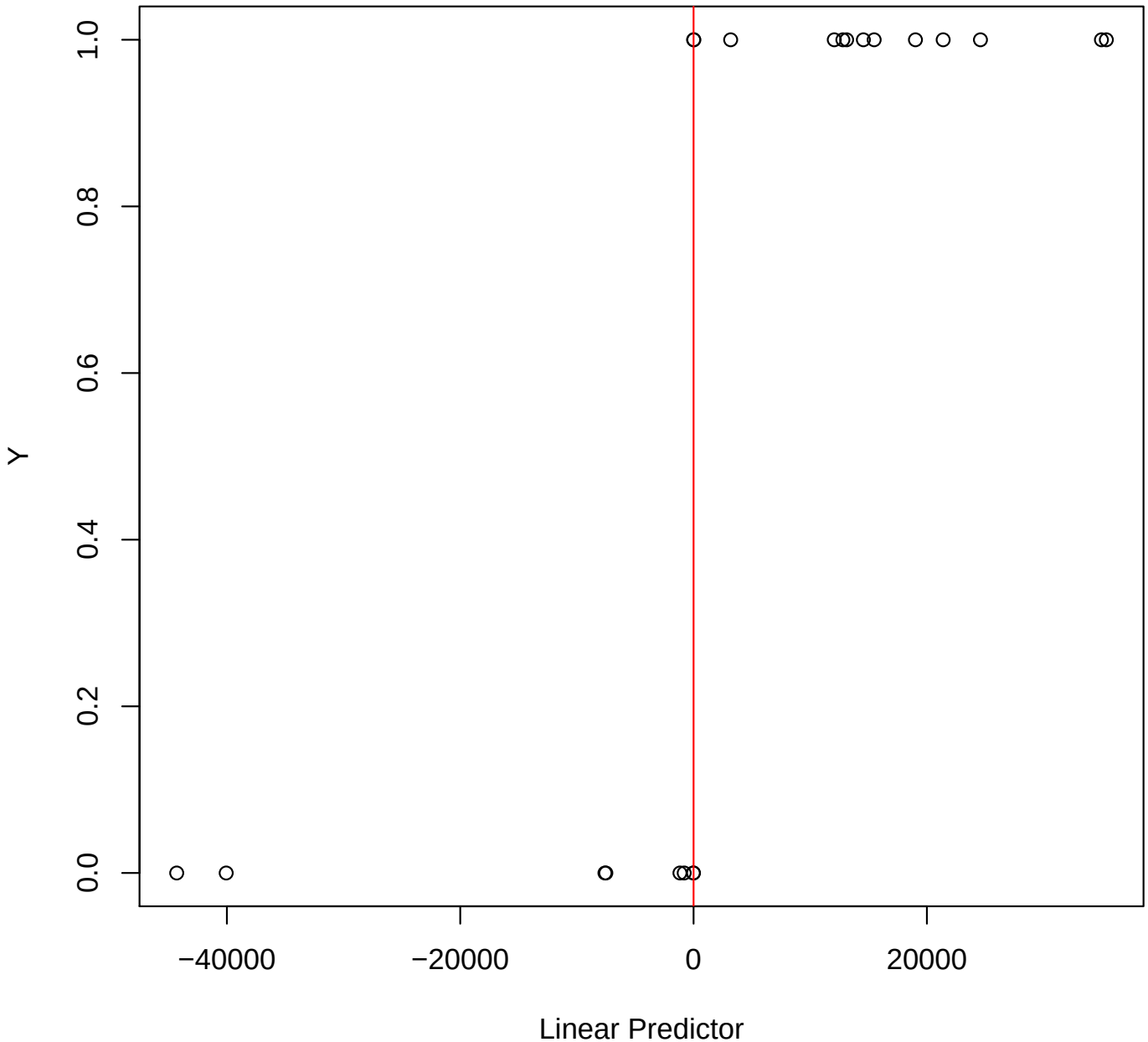
```
> mod <- glm(y ~ x1 + x2 + x3, family=binomial, data=incontinence)
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
> summary(mod)  # note estimates, SEs, iteration count

Call:
glm(formula = y ~ x1 + x2 + x3, family = binomial, data = incontinence)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.056e-03  -2.000e-08   2.000e-08   2.000e-08   1.038e-03

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -83.84    2879.50  -0.029    0.977
x1          -2445.04   79162.56  -0.031    0.975
x2          -1653.76   53562.97  -0.031    0.975
x3            310.27   10043.85   0.031    0.975

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2.7910e+01  on 20  degrees of freedom
Residual deviance: 3.3402e-06  on 17  degrees of freedom
AIC: 8

Number of Fisher Scoring iterations: 25
```

Is this complete separation? Quasi-complete separation? Something else?

Let's plot y versus the fitted linear predictor value:

```
> plot(predict(mod), incontinence$y, xlab="Linear Predictor", ylab="Y")

> abline(v=0, col="red")
```


Y
0.0
0.2
0.4
0.6
0.8
1.0
−40000
−20000
0
20000
Linear Predictor

Plot inconclusive, because unclear what happens near boundary. Let's check the numerical values:

```
> cbind(incontinence$y, predict(mod))  # y determines sign
   [,1]          [,2]
1     0    -15.09434
2     0  -7520.04049
3     0  -1168.53511
4     0 -44302.55894
5     0   -801.87245
6     0    -14.40058
7     0  -7596.88235
8     0 -40061.57684
9     1  34962.37116
10    1  35380.29534
11    1     15.03288
12    1  19018.67364
13    1  12060.76101
14    1  14555.95211
15    1   3180.29680
16    1  21394.91686
17    1  15477.64872
18    1  13119.87959
19    1  24590.35169
20    1  12797.12809
21    1     14.43414
```

Looks like complete separation.

Try likelihood-ratio inference:

```
> drop1(mod, test="Chisq")
Single term deletions

Model:
y ~ x1 + x2 + x3
       Df Deviance    AIC    LRT  Pr(>Chi)
<none>       0.000  8.000
x1      1   11.126 17.126 11.126 0.0008515 ***
x2      1   22.235 28.235 22.235 2.412e-06 ***
x3      1   14.332 20.332 14.332 0.0001532 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Try score inference:

```
> drop1(mod, test="Rao")
Single term deletions

Model:
y ~ x1 + x2 + x3
       Df Deviance    AIC Rao score Pr(>Chi)
<none>       0.000  8.000
x1      1   11.126 17.126    3.6941 0.054605 .
x2      1   22.235 28.235    9.6528 0.001891 **
x3      1   14.332 20.332    6.4233 0.011263 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```