36-617: Applied Linear Models

Logistic Regression

Brian Junker
132E Baker Hall
brian@stat.cmu.edu

Statistics & Data Science Social

hosted by **Women in Statistics**

social hour + board games

coffee shop gift card raffle

pizza + refreshments

open to **all students**
(undergraduate and graduate!!)
interested in statistics and data science!

chat about hobbies, life, and more!

stop in whenever and stay however long you like!

thursday, september 29
4-6 pm
hamburg hall 1007

In the coming days…

- HW 04 due tonight at 11:59
- No Quiz next Monday
- Takehome Midterm will be out around 6pm
  - Covers through Ch 7; Due Wed Oct 5 at 11:59pm
  - Open-book, open-notes, etc., ***but not open-people***
    - Do the work on your own, no collaborators
      - In particular, no current or former MSP students or solutions
    - Feel free to use web resources (incl stackexchange etc) but you ARE NOT ALLOWED to post questions or interact on the web
  - Office hours (me or TA) or private Piazza questions are fine (ok to learn from my answers to other students)

Outline

- Logistic Regression
- Interpreting the Coefficients
- Example: Extract from the Coleman Report
- Improving the Model
- Overfitting and Identifiability
- Effect of Dichotomization
- Assessing Residuals
- Example: Wells in Bangladesh
  - A simple model
  - Variable selection

Logistic Regression

- Basic Setup
  - y = 0 or 1, indicating some outcome of interest (passed test, responded to treatment, is a water well of type A rather than type B, switched brands of soap, etc.)
  - x_1, x_2, …, x_k are continuous or discrete predictor variables (income, SES, test score, mother's IQ, amount of sulphur, parents divorced, etc.)
- We want to build a ***linear model*** to predict y from the x's, just like linear regression

Logistic Regression

- The *linear regression* model was

$$\begin{aligned} y_i &\overset{indep}{\sim} N(\theta_i, \sigma^2),\ i = 1, \ldots, n \\ \theta_i &= X_i\beta = \beta_1 X_{i1} + \cdots \beta_k X_{ik} \end{aligned}$$

  - Each y_i has some mean $\theta_i = E[y_i]$
  - Each θ_i has some linear structure
  - There is a statistical distribution N(*, σ^2) that describes unmodeled variation around θ_i
- Obviously y = 0 or 1 cannot have a normal distribution, but we want the same structure!

Logistic Regression

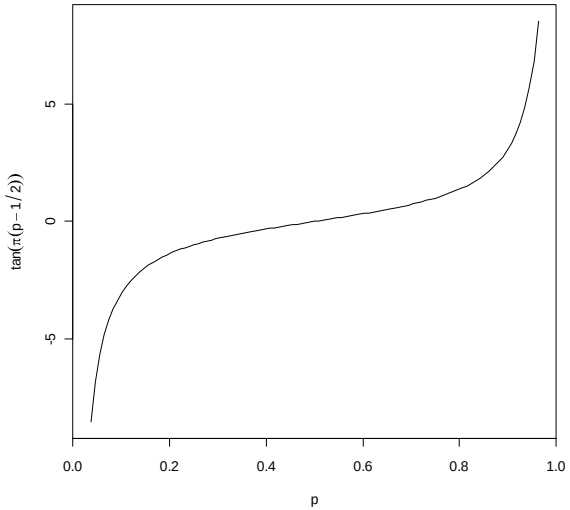
- By analogy with linear regression, we model as

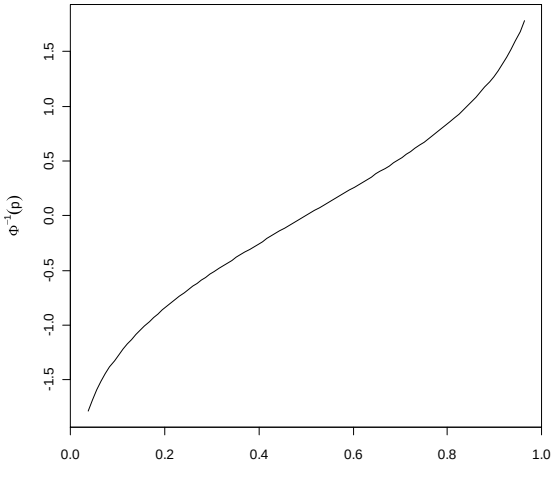
$$y_i \quad \sim \quad \text{some distribution depending on}$$

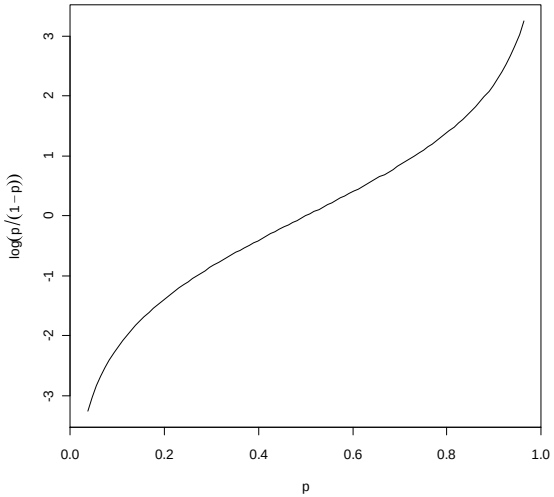
$$E[y_i] = P(Y_i = 1) = p_i$$

- Since $p_i \in [0,1]$, we often use an S-shaped function to stretch p_i out to the whole real line (so unrestricted linear modeling is possible)
- Some choices:
  - Tangent function: $\theta_i = \tan(\pi \cdot (p_i - \frac{1}{2}))$
  - Probit function: $\theta_i = \Phi^{-1}(p_i)$
  - Logit function: $\theta_i = \log \frac{p_i}{1-p_i}$

Aside… S –shaped Functions







$$\theta_i = \tan(\pi \cdot (p_i - \tfrac{1}{2})) \qquad \theta_i = \Phi^{-1}(p_i) \qquad \theta_i = \log \frac{p_i}{1-p_i}$$

```
curve(tan(pi*(x-1/2)),xlab="p",ylab=expression(tan(pi*(p-1/2))))
curve(qnorm(x),xlab="p",ylab=expression({Phi^{-1}}(p)))
curve(log(x/(1-x)),xlab="p",ylab=expression(log(p/(1-p))))
```

Not much difference between $\Phi^{-1}(p)$ and $\log(p/(1-p))$. We usually use $\log(p/(1-p))$.

Logistic Regression

- The ***logistic regression*** model is:

$$\begin{aligned} y_i &\overset{indep}{\sim} Bernoulli(p_i),\ i = 1, \ldots, n \\ \theta_i &= \log \frac{p_i}{1 - p_i} = X_i\beta \\ &= \beta_1 X_{i1} + \cdots \beta_k X_{ik} \end{aligned}$$

- Two useful functions:

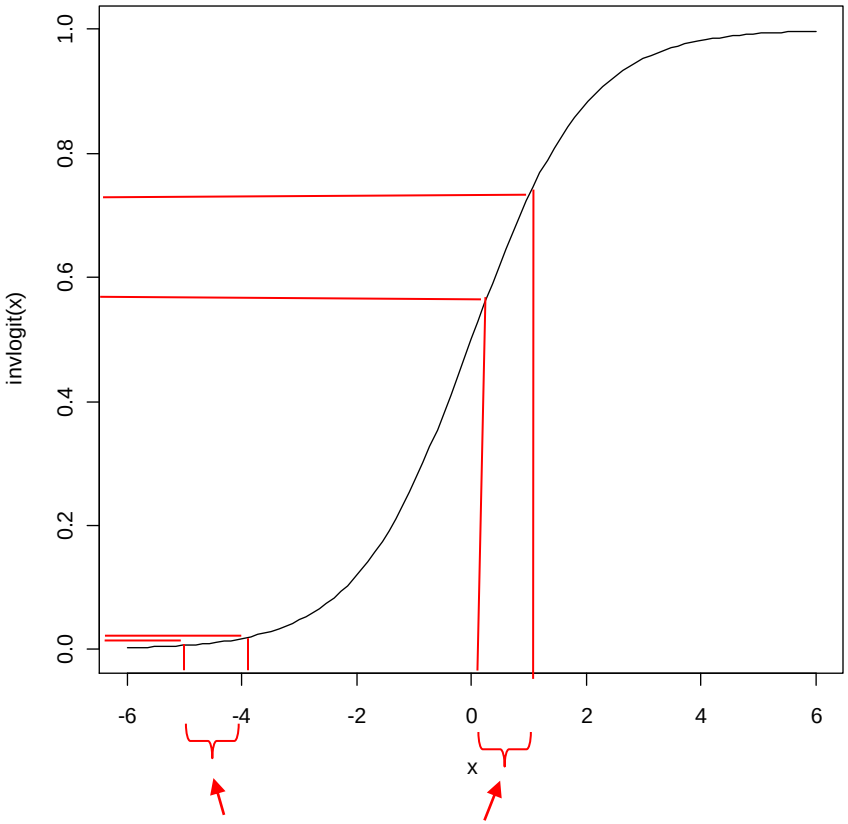
```
logit <- function (p) { log(p/(1-p)) }
invlogit <- function(x) {exp(x)/(1 + exp(x))}
```

(sometimes invlogit known as “expit”…)

Interpreting the Coefficients

- $p_i = \frac{\exp[\beta_0 + \beta_1 x_i]}{1 + \exp[\beta_0 + \beta_1 x_i]}$
- Difficult to predict effect of change from x_i to $x_i + 1$ on p_i because it depends on where p_i (or x_i) is!
- Maximum effect when $\beta_o + \beta_1 x_i = 0$; can show the effect is to change p_i by $\beta_1/4$

  ("divide by 4" rule)



Interpreting the Coefficients

- $\log \frac{p_i}{1-p_i} = \beta_0 + \beta_1 x_i$
- $O_i = p_i/(1-p_i)$ is the Odds
  - If there is a 50-50 chance, $p_i=1/2$, and so $O_i = 1$ (even odds)
  - If $p_i = 1/3$ then $O_i = 1/2$, two-to-one odds against
  - log O_i = log-odds (logit)
- Going from x_i to x_i+1 produces
  - An ***additive*** change of β_1 in the log-odds
  - A ***multiplicative*** change of e^{β_1} in the odds
  - No matter where x_i or p_i are!

Interpreting the Coefficients

- When there is more than one predictor

$$p_i = \frac{\exp\{\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}\}}{1 + \exp\{\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}\}}$$

  is useful for prediction, but difficult to interpret

- The log-odds (logit) form

$$\log \frac{p_i}{1 - p_i} = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_j x_j + \cdots + \beta_k x_{ik}$$

  has the same interpretation as before: a change from x_j to $x_j + 1$ produces a change of β_j in the log odds (holding the other x's fixed)

  - Assumes x_j can be manipulated w/o changing other x's

Digression: Odds Ratios

- If p_1 and p_2 are probabilities with odds $O_1 = p_1/(1-p_1)$ and $O_2 = p_2/(1-p_2)$ then $OR_{12} = O_1/O_2$ is the ***odds ratio***
  - If $p_1 = 2/3$ and $p_2 = 1/3$ then $OR_{12} = 2/(1/2) = 4$, so the odds of event 1 are 4 times the odds of event 2.
  - $\log(OR_{12})$ is the ***log odds ratio***
- Suppose

$$\log \frac{p_1}{1-p_1} = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_j x_j + \cdots + \beta_k x_k$$

$$\log \frac{p_2}{1-p_2} = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_j (x_j + 1) + \cdots + \beta_k x_k$$

then

$$\beta_j = \log \frac{p_2}{1-p_2} - \log \frac{p_1}{1-p_1} = \log O_2/O_1 = \log(OR_{21})$$

- β_j is the log-odds ratio for going from x_j to $x_j + 1$

Example

- Mosteller & Tukey (1977) data on average verbal test scores for 6th graders at 20 mid-Atlantic schools taken from The Coleman Report:

```
      X1    X2      X3    X4   X5 Y     Z
1   3.83 28.87    7.20 26.60 6.19 1 37.01
2   2.89 20.10 -11.71 24.40 5.17 0 26.51
.
.
.
20 2.37 76.73  12.77 24.51 6.96 1 41.01
```

- X1 = staff salaries per pupil; X2 = percent of fathers in white collar jobs; X3 = socioeconomic status; X4 = average verbal test scores for *teachers* at each school; X5 = (mothers' years of schooling)/2; Z = mean verbal test scores for *students* at each school; and Y = 1 if Z > 37 and Y = 0 if not

Example, Cont'd

- We begin by fitting an additive (main effects only) logistic regression to the above data

```
> schools <- read.table("mosteller-tukey.txt")
> summary(fit0 <- glm(y ~ x1 + x2 + x3 +x4 +x5,data=schools,family=binomial))

Call:
glm(formula = y ~ x1 + x2 + x3 + x4 + x5, family = binomial,data = schools)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.5635    33.1771  -0.138    0.891
x1 (staff sal) 2.1346     3.3235   0.642    0.521
x2 (% wht col) 0.1135     0.1592   0.713    0.476
x3 (SES)       0.9789     0.8487   1.153    0.249
x4 (tchr test) 2.0242     1.3251   1.528    0.127
x5 (mom yrs  -10.0928     9.7992  -1.030    0.303
  of school)
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27.526  on 19  degrees of freedom
Residual deviance:  8.343  on 14  degrees of freedom
AIC: 20.343
```

No R^2 but think of this as a χ^2 test of fit…

Interpreting the Coefficients, Cont'd

- Reading off the coefficients table in the example,
  - If we increase staff salaries per pupil by 1 unit, the model predicts an increase in log-odds of a successful school of 2.13;
  - If we increase the percent of fathers in white collar jobs by one unit, the model predicts an increase in log-odds of a successful school increase by 0.11; etc.
  - This assumes we can manipulate x_j, and can do so without affecting the other x_j's!

Interpreting the Coefficients, Cont'd

- When β_j is (insignificantly different from) zero, we can infer that *y* and x_j are independent, conditional on the other *x*'s in the model
- In our example, *none* of the coefficients are significantly different from zero! Same sorts of suspects as with ordinary linear regression:
  - Small sample size—only 20 observations
  - Collinearity in the *x*'s—indeed:

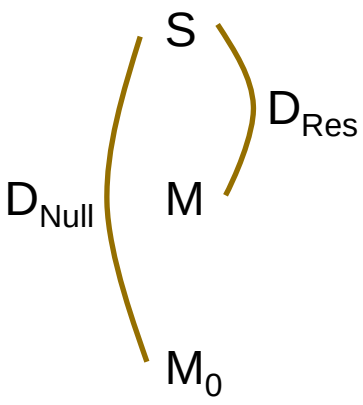
```
> vif(fit0)
       x1        x2        x3        x4        x5
 1.749053 22.382765 22.795807  2.111019 55.066313
> X <- model.matrix(fit0)
> cor(X[,-1])
           x1         x2        x3         x4        x5
x1 1.0000000 0.18113980 0.2296278 0.50266385 0.1967731
x2 0.1811398 1.00000000 0.8271829 0.05105812 0.9271008
x3 0.2296278 0.82718291 1.0000000 0.18332924 0.8190633
x4 0.5026638 0.05105812 0.1833292 1.00000000 0.1238087
x5 0.1967731 0.92710081 0.8190633 0.12380866 1.0000000
```

Null & Residual Deviance

- At the end of glm summary you will see

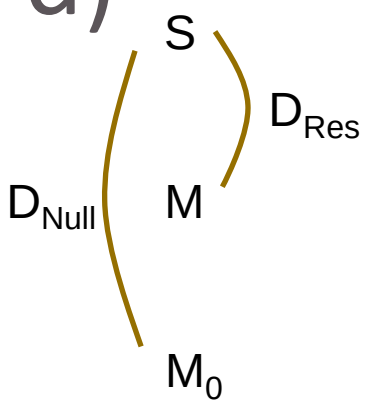
```
    Null deviance: 27.526  on 19  degrees of freedom
Residual deviance:  8.343  on 14  degrees of freedom
AIC: 20.343
```

- Let $D_0(M) = -2\log(\text{likelihood}_M)$, and let
  - S = "saturated" model that perfectly fits data
  - M = model fitted by glm
  - M_0 = the intercept only model



- Null Deviance: $D_{Null} = D_0(M_0) - D_0(S)$;
  Null df: $df_{Null} = df(S) - df(M_0)$
- Residual Deviance: $D_{Res} = D_0(M) - D_0(S)$;
  Residual df: $df_{Res} = df(S) - df(M)$

Null & Residual Deviance (cont'd)



- This gives rise to two *likelihood ratio* Chi-squared tests:
- H_0: M (fitted model) vs. H_A: S (saturated model)
  - Under H_0, D_{Res} ~ Chi-squared on df_{Res}
  - 8.343 on 14 df: Don't reject H_0, fitted model OK
- H_0: M_0 (intcpt-only) vs. H_A: M (fitted model)
  - Under H_0, $D_{Null} - D_{Res}$ ~ Chi-squared on $df_{Null} - df_{Res}$
  - 27.526 – 8.343 = 19.183 on 19 – 14 = 5 df: Reject H_0; fitted model better than intercept-only

Improving the Model

- Improving logistic regression models is like improving linear regression models
  - Add variables and interactions that make sense
  - Add variables and interactions if they greatly increase R^2, or if they improve residuals, etc.
  - Transform X variables to reduce leverate and/or improve interpretation (functional form, etc.)
- Unlike lm(), glm() does not report R^2. Instead it reports AIC:
  - AIC = -2*log(likelihood) + 2*(df) ***[small is good]***
  - Like a *likelihood ratio test*, but penalized for the complexity of the model (df = number of regression coefficients)

Improving the Model

- stepAIC() in library(MASS) will search through a set of models, minimizing AIC.

```
> library(MASS)
> basemodel <- glm(y ~x1 + x2 + x3 +x4 + x5 ,
+  data=schools,family=binomial)
> fit1 <- eval(stepAIC(basemodel, scope=list(lower=.~1,
+ upper=.~x1 + x2 + x3 +x4 + x5,k=2))$call)
> anova(fit1,fit0,test="Chisq")
Analysis of Deviance Table
Model 1: y ~x3 + x4
Model 2: y ~x1 + x2 + x3 + x4 + x5
     Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1           17    10.1414
2           14     8.3429  3   1.7984    0.6153
> summary(fit1)$coef
               Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -41.8188263 24.5239239 -1.705226 0.08815233
x3            0.3646223  0.1798581  2.027277 0.04263408
x4            1.5614704  0.9427877  1.656227 0.09767586
```

Chi-squared test finds no evidence against smaller model

ses
tchr sco

Improving the Model

- If we try to expand the model to consider interactions of all orders, something interesting happens:

```
> fit2 <- eval(stepAIC(basemodel,
+ scope=list(lower=.~ 1,
+ upper=.~(x1 + x2 + x3 +x4 + x5)^5,k=2))$call)
y ~ x3 + x4 + x5 + x4:x5

        Df Deviance    AIC
<none>       0.0000 10.000
+ x3:x5  1   0.0000 12.000
+ x3:x4  1   0.0000 12.000
+ x1     1   0.0000 12.000
+ x2     1   0.0000 12.000
- x3     1   9.2741 17.274
- x4:x5  1   9.2821 17.282
There were 50 or more warnings
(use warnings() to see the first 50)
```

```
> warnings()
Warning messages:
1: glm.fit: fitted probabilities
   numerically 0 or 1 occurred
2: glm.fit: algorithm did not converge
3: glm.fit: fitted probabilities
   numerically 0 or 1 occurred
4: glm.fit: algorithm did not converge
5: glm.fit: fitted probabilities
   numerically 0 or 1 occurred
```

Overfitting and Identifiability

- Comparing fitted(fit2) to the actual y's you will see that they agree closely: $|y_i - p_i| \approx 0.$

```
> y - fitted(fit2)
             1              2              3              4              5
 2.220446e-16 -2.220446e-16 -2.712309e-09  1.053467e-09  2.171825e-10
             6              7              8              9             10
-2.220446e-16  2.220446e-16 -2.220446e-16  6.313647e-10  2.220446e-16
            11             12             13             14             15
-2.220446e-16 -2.220446e-16 -2.220446e-16 -2.220446e-16 -2.220446e-16
            16             17             18             19             20
 2.220446e-16 -2.220446e-16 -2.220446e-16  1.574083e-09  2.220446e-16
```

- log $p_i/(1-p_i)$ can't be evaluated accurately when p $\approx$ 0 or 1. Estimates of the regression coefficients can go haywire too.

The Effect of Dichotomization

- Finally we recall that y is a dichotomized version of z: y = 1 if z > 37; otherwise y = 0

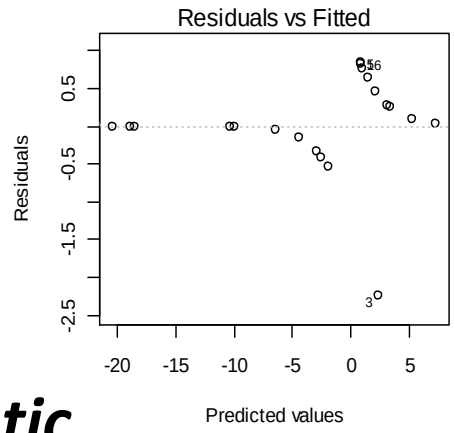
```
> basemodel <- lm(z ~x1 + x2 + x3 +x4 + x5 ,data=schools)
> norm1 <- eval(stepAIC(basemodel,
                  scope=list(lower=.~1,
                   upper=.~(x1 + x2 + x3 +x4 + x5)^5),
                   k=2)$call) # k=2 for AIC
> norm1$call
lm(formula = z ~x1 + x3 + x4, data = schools)
```

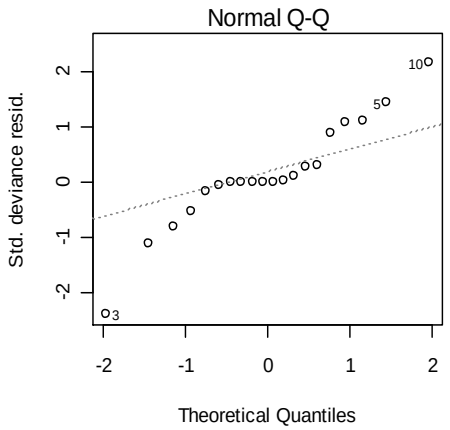
- Even though the stepwise procedure had access to interactions of all orders, x1 was not in the final (logistic) model for y.
- This suggests that x1 was more useful for predicting the more complex response z than for predicting the simpler response y (dichotomized z).
  - We should dichotomize with care, and then only if the substantive question requires it.
  - Dichotomization always changes the information in the data.
  - If you must dichotomize, I'd suggest doing a sensitivity analysis (try different dichotomizations and see how that affects the results).

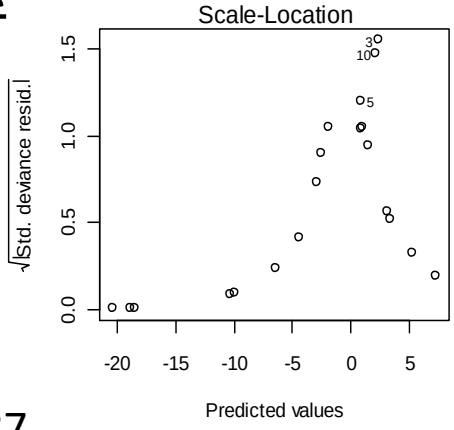
Assessing Residuals

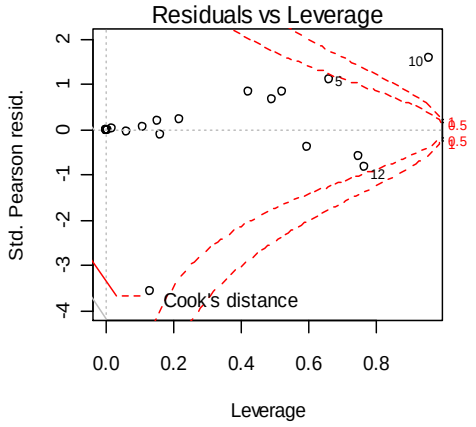
```
par(mfrow=c(2,2))
plot(fit0,
   add.smooth=F)
```

- ***Residual plots for logistic regression usually look terrible!***
- Fit is pretty good:
  - Resid deviance = 8.3
  - Good fit: $P\left[\chi^2_{14} > 8.34\right]$
    = pchisq(8.3,14,lower=F) = 0.87









Final Example: Wells in Bangladesh

- Researchers classified wells as “safe” or “contaminated with arsenic” and collected data on families using the wells. They encouraged those with unsafe wells to switch to safe wells (a neighbor’s well, a community well, or a new well).
- Several years later they came back to see who switched.

```
> wells <- read.table("wells.dat“)
> str(wells)
#'data.frame':   3020 obs. of  5 variables:
# $ switch : int  1 1 0 1 1 1 1 1 1 1 ...          did the family switch wells?
# $ arsenic: num  2.36 0.71 2.07 1.15 1.1 ... how much arsenic in old well?
# $ dist   : num  16.8 47.3 21 21.5 40.9 ...   distance (m) to nearest safe well
# $ assoc  : int  0 0 0 0 1 1 1 0 1 1 ...          anyone in fam active in cmty?
# $ educ   : int  0 0 10 12 14 9 4 10 0 0 ...    education level of head of h'hold
```

Bangladesh Wells – Fitting a Simple Model

I rescaled dist to make $\hat{\beta}_{dist}$ easier to read; in later analyses I will just use dist…

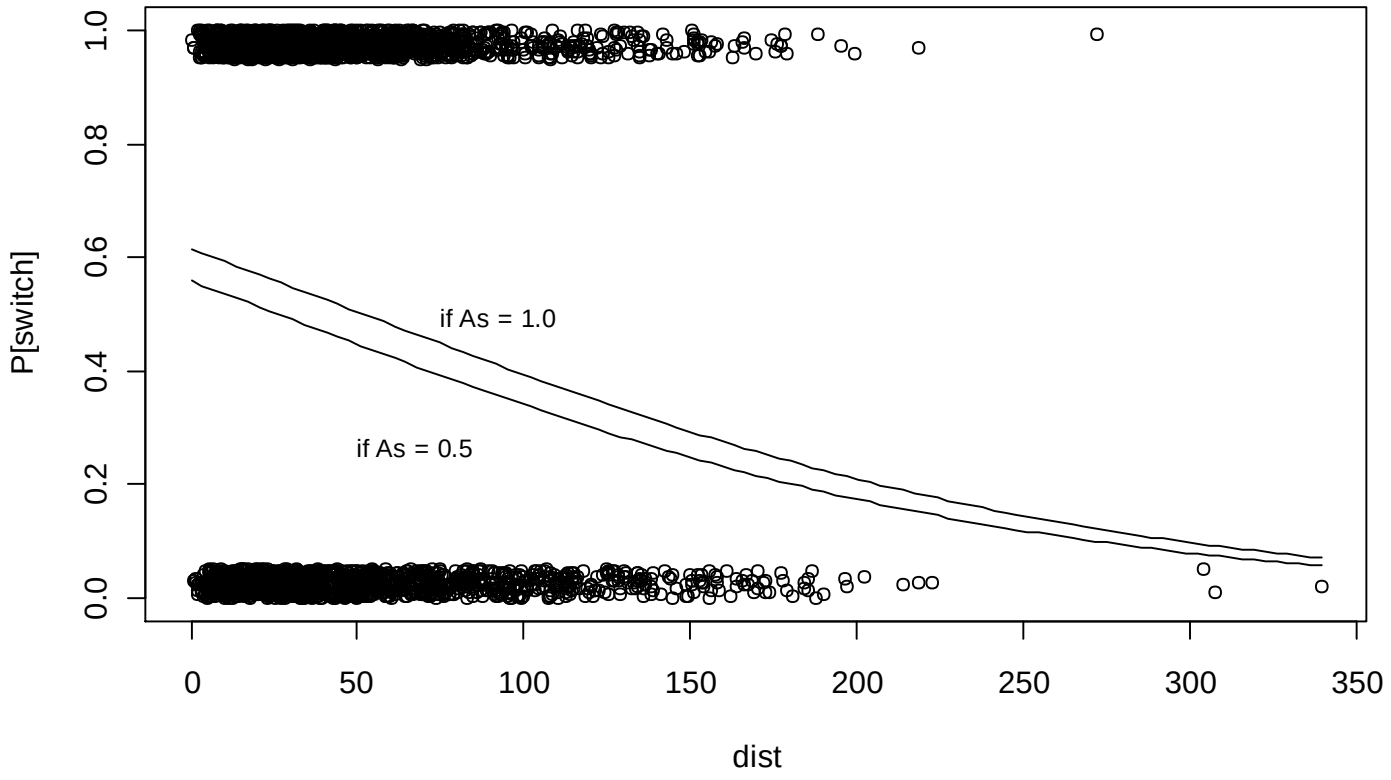
```
> attach(wells)
> dist100 <- dist/100
> fit.3 <- glm (switch ~ dist100 + arsenic,
+     family=binomial(link="logit"))
> summary(fit.3)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.002749   0.079448   0.035    0.972
dist100     -0.896644   0.104347  -8.593   <2e-16 ***
arsenic      0.460775   0.041385  11.134   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
Null deviance: 4118.1  on 3019  degrees of freedom
Residual deviance: 3930.7  on 3017  degrees of freedom
AIC: 3936.7
```

Bangladesh Wells – Plotting P[switch] vs distance to safe well

```
jitter.binary <- function(a, jitt=.05){
  ifelse (a==0, runif (length(a), 0, jitt), runif (length(a), 1-
jitt, 1))
}

switch.jitter <- jitter.binary(switch)

plot(dist,switch.jitter,xlim=c(0,max(dist)),ylab="P[switch]")
curve (invlogit(cbind (1, x/100, .5) %*% coef(fit.3)), add=TRUE)
curve (invlogit(cbind (1, x/100, 1.0) %*% coef(fit.3)), add=TRUE)
text (50, .27, "if As = 0.5", adj=0, cex=.8)
text (75, .50, "if As = 1.0", adj=0, cex=.8)
```

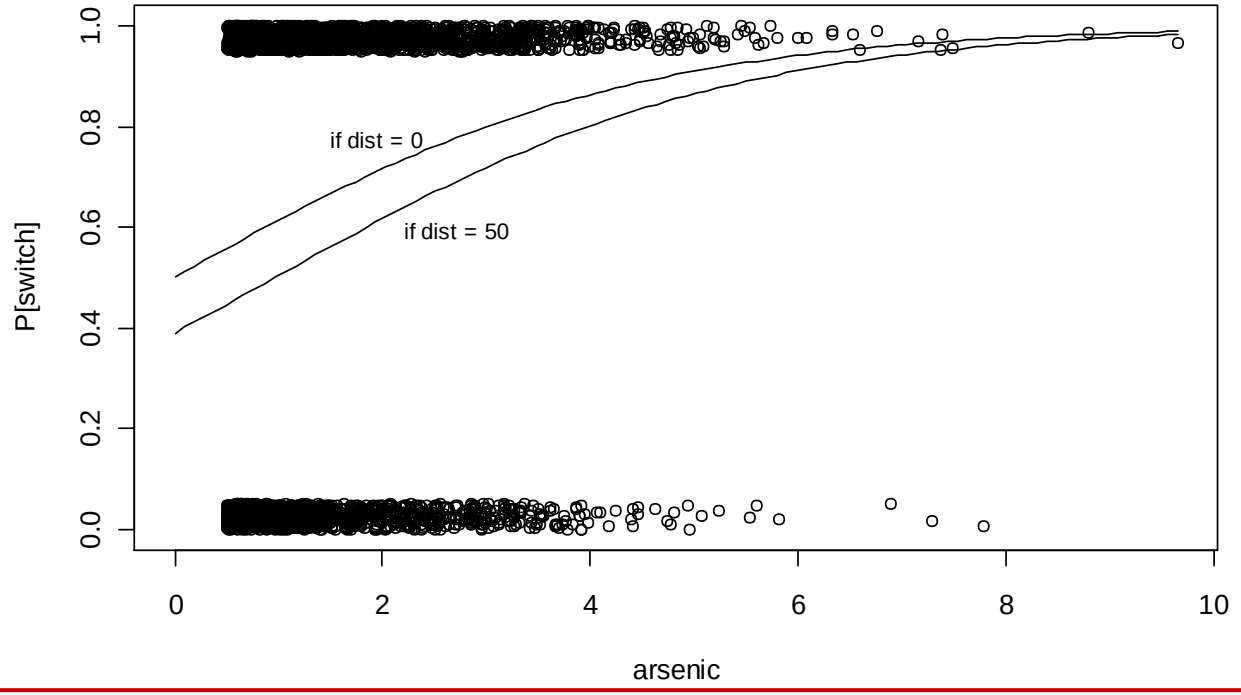
(plot on next page)

Bangladesh Wells – Plotting P[switch] vs distance to safe well



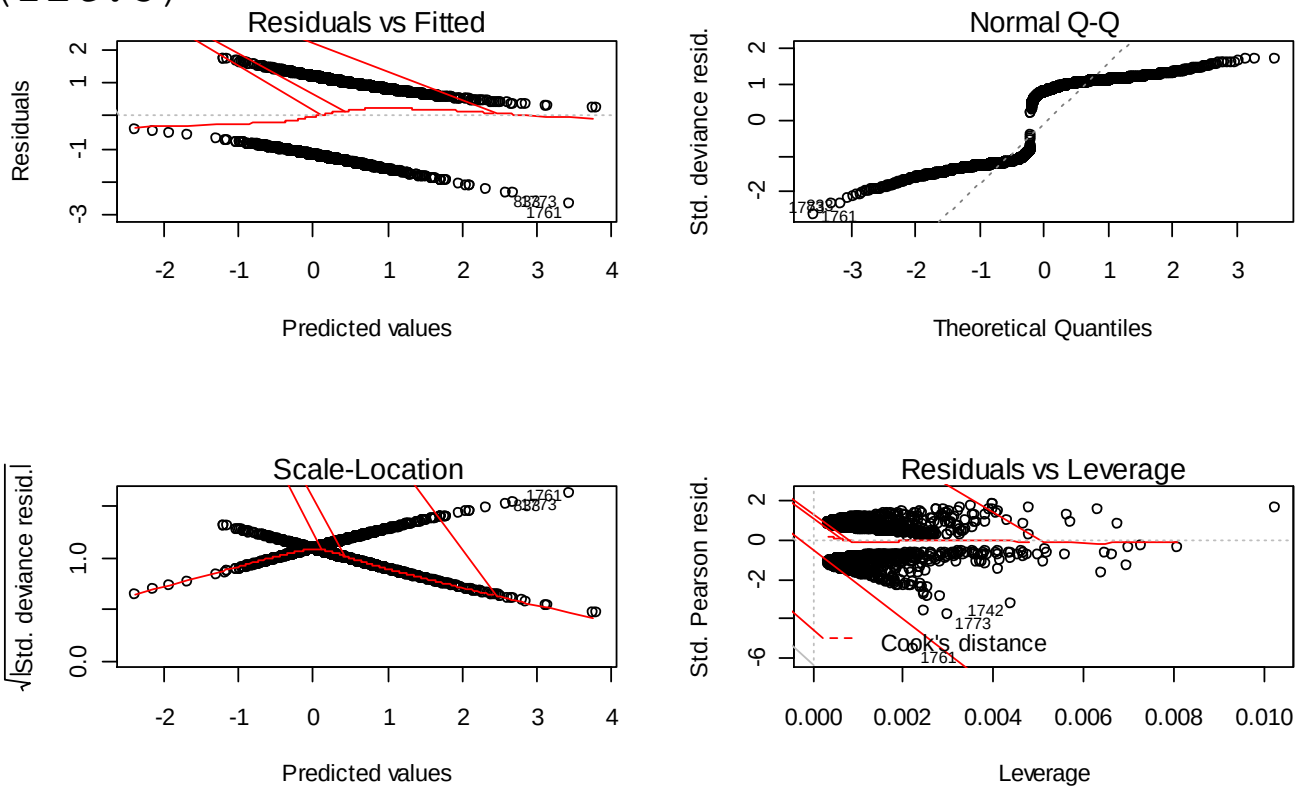
Bangladesh Wells – Plotting P[switch] vs arsenic level of old well

```
plot(arsenic,switch.jitter,xlim=c(0,max(arsenic)),ylab="P[switch]")
curve (invlogit(cbind (1, 0/100,  x) %*% coef(fit.3)), add=TRUE)
curve (invlogit(cbind (1, 50/100, x) %*% coef(fit.3)), add=TRUE)
text (1.5, .78, "if dist = 0", adj=0, cex=.8)
text (2.2, .6, "if dist = 50", adj=0, cex=.8)
```



Bangladesh Wells – Standard R Residual Plots

```
par(mfrow=c(2,2))
plot(fit.3)
```



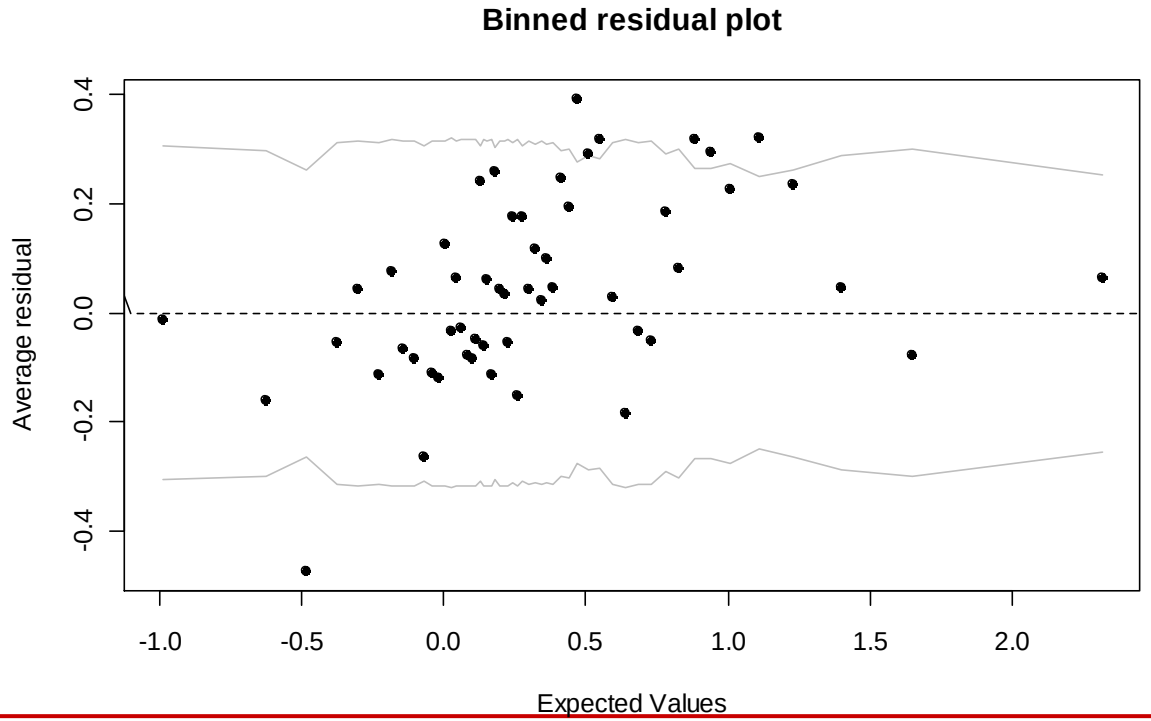
Assessing Residuals

- We can make the behavior of the residual vs fitted plot more like residuals in linear regression by binning the data
  - Make 10 or more bins of fitted values, and then average the residuals in each bin
- Library(arm) has the binnedplot() function to help create the plot (default # of bins: $\sqrt{n}$)
  - Really need at least $n \geq 100$ and still not always useful
- Next lecture: DHARMa residuals based on the inverse probability transform (much more useful!)

Bangladesh – Binned Residuals

```
Library(arm)
par(mfrow=c(1,1))
x <- predict(fit.3)
y <- resid(fit.3)
binnedplot(x,y)
```

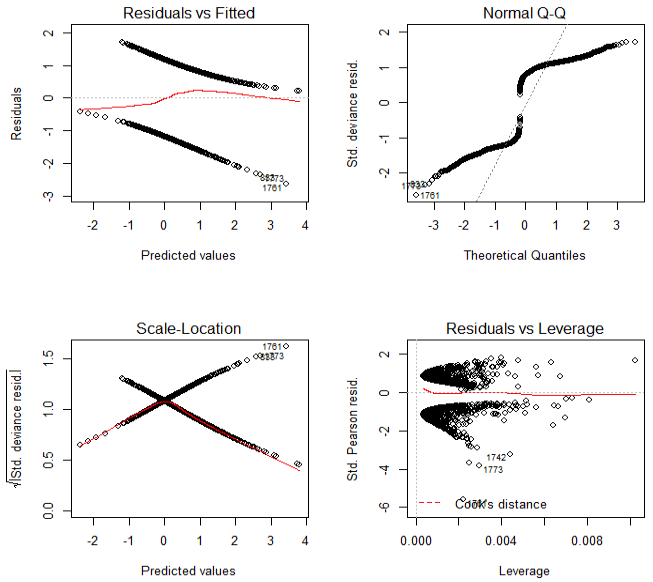
- Grey lines are 95% envelope
- If we believe inverted U-shape, could transform one or more x's
  - log(dist), log(arsenic)
  - dist + dist^2 …

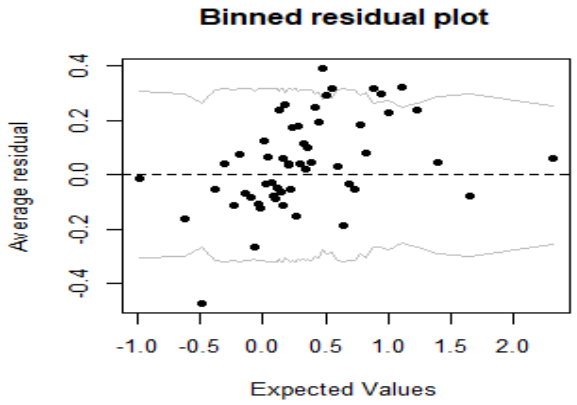


Not so useful for small samples (want $n \geq 100$!)

Comparing unbinned vs binned resids

```
> data <-
read.table("wells.dat",
+   header=T)
> glm.1 <- glm(switch ~
+ dist + arsenic,
+ data=data,
+ family=binomial)
> par(mfrow=c(2,2))
> plot(glm.1)
> library(arm)
# for binnedplot();
# also brings in MASS
# which has AIC(), BIC()
> binnedplot(predict(glm.1),
+ resid(glm.1))
```

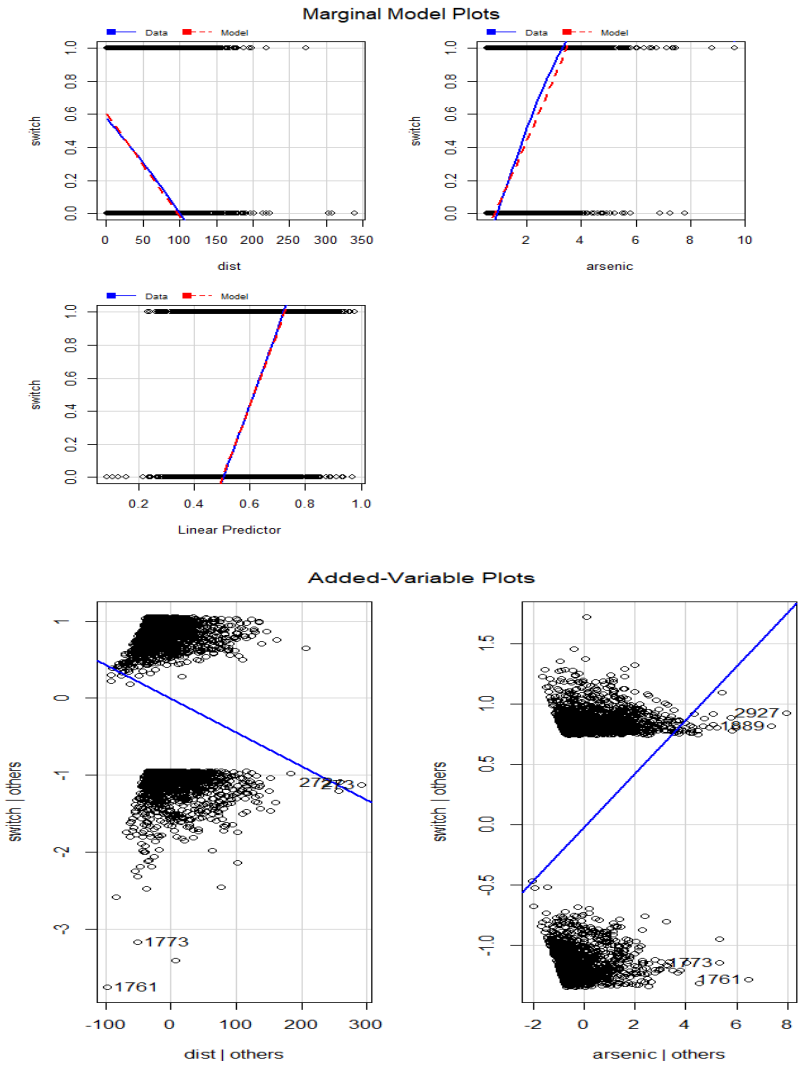




Variable Selection for Arsenic Data…

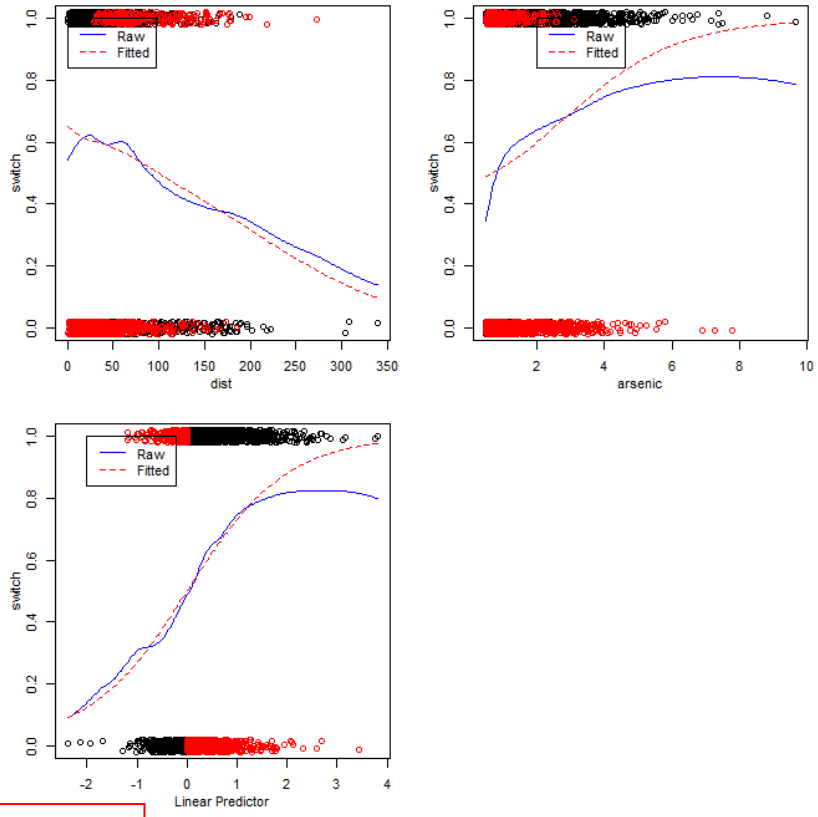
```
> library(car)
# for avPlots(),
# mmps(), plot()
# method for lasso...
> mmps(glm.1)
> avPlots(glm.1)
# not very useful for
# logistic regression!
```

Marginal model plots (mmps function) actually are useful for thinking about variable selection & transformation, but the “car” mmps function doesn’t work for glms – see next slide!



Aside: problems with mmps(), fixed with mmplot()

```
# (1) install locfit
# from CRAN
# (2) install
# marginalmodelplots
# from
#  marginalmodelplots_0.4.2.tar.gz
> library(marginalmodelplots)
> mmplot(glm.1)
```




```
> # download marginalmodelplots_0.4.2.tar.gz
> # from canvas
> install.packages("locfit")
> install.packages("marginalmodelplots_0.4.2.tar.gz", repos=NULL)
```

Variable Selection for Arsenic Data…

```
> glm.2 <- glm(switch ~ dist + I(dist^2) + arsenic,
+ data=data, family=binomial)
> glm.3 <- glm(switch ~ dist + arsenic +
+ I(arsenic^2), data=data, family=binomial)
> glm.4 <- glm(switch ~ dist + I(dist^2) +
+ arsenic + I(arsenic^2), data=data, family=binomial)
> glm.5 <- glm(switch ~ dist*arsenic, data=data,
+ family=binomial)
> glm.5 <- glm(switch ~ dist+arsenic+dist:arsenic,
+ data=data, family=binomial)
> glm.5 <- glm(switch ~ dist+arsenic+I(dist*arsenic),
+ data=data, family=binomial)
#
# All three ways of writing glm.5 specify exactly
# the same model!
```

Variable Selection for Arsenic Data…

```
> anova(glm.1,glm.2,glm.3,glm.4,glm.5)
Analysis of Deviance Table
Model 1: switch ~ dist + arsenic
Model 2: switch ~ dist + I(dist^2) + arsenic
Model 3: switch ~ dist + arsenic
                  + I(arsenic^2)
Model 4: switch ~ dist + I(dist^2) + arsenic
                  + I(arsenic^2)
Model 5: switch ~ dist + arsenic
                  + I(dist * arsenic)
  Resid. Df Resid. Dev Df Deviance
1      3017     3930.7
2      3016     3930.1  1   0.5683
3      3016     3911.4  0  18.7481
4      3015     3911.0  1   0.3893
5      3016     3927.6 -1 -16.6659
> ## can't do LR tests, models not nested!
> anova(glm.1,glm.2,glm.3,glm.4,glm.5,
+ test="AIC")
Error in match.arg(test) :
  'arg' should be one of "Rao", "LRT",
  "Chisq", "F", "Cp"
> ## whoops, can't use R's anova() function
> ## for these comparisons...
```

```
> AIC(glm.1,glm.2,glm.3,glm.4,glm.5)
      df      AIC
glm.1  3 3936.668
glm.2  4 3938.100
glm.3  4 3919.352
glm.4  5 3920.963
glm.5  4 3935.628
> BIC(glm.1,glm.2,glm.3,glm.4,glm.5)
      df      BIC
glm.1  3 3954.707
glm.2  4 3962.152
glm.3  4 3943.404
glm.4  5 3951.028
glm.5  4 3959.680
> ## and if we want to put them all together:
> check <- function(x)
+ { round(c(AIC=AIC(x),BIC=BIC(x)),2) }
> cbind(glm.1=check(glm.1),glm.2=check(glm.2),
+ glm.3=check(glm.3),glm.4=check(glm.4),
+ glm.5=check(glm.5))
      glm.1   glm.2   glm.3   glm.4   glm.5
AIC 3936.67 3938.10 3919.35 3920.96 3935.63
BIC 3954.71 3962.15 3943.40 3951.03 3959.68
```

Variable Selection for Arsenic Data…

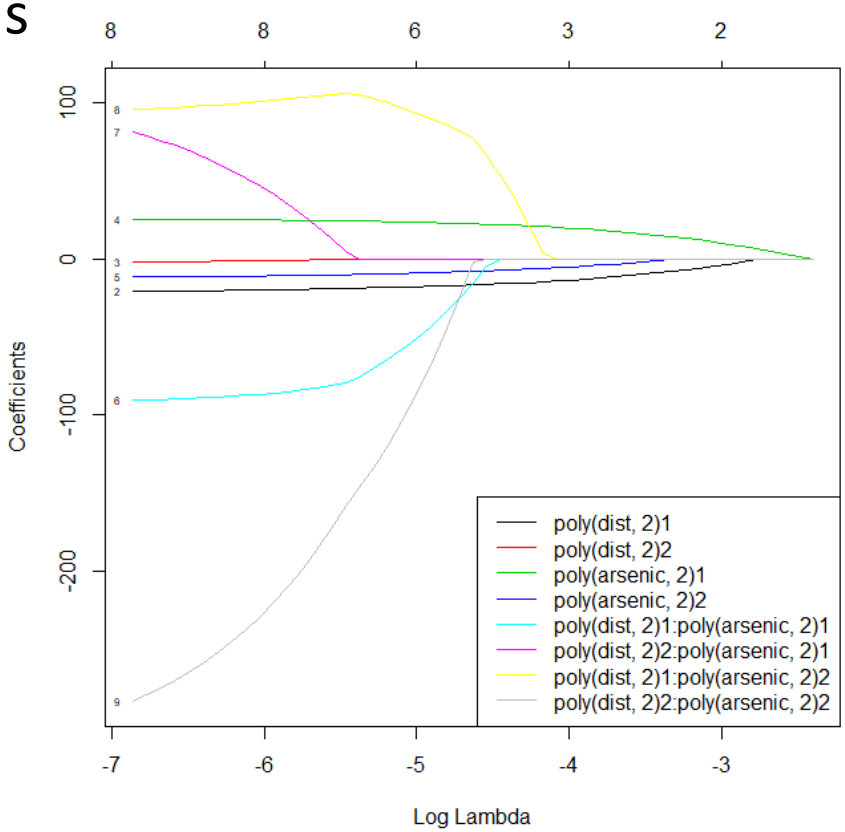
```
> summary(glm.00<-glm(switch ~ poly(dist,2) * poly(arsenic,2),
data=data,family=binomial))
> ## shorthand for switch ~ dist + arsenic + I(dist^2) + I(arsenic^2) +
> ## I(dist*arsenic) + I(dist^2*arsenic) + I(dist*arsenic^2) + I(dist^2*arsenic^2)
                                        Est         SE z value Pr(>|z|)
(Int)                                0.34633    0.04083   8.482  < 2e-16 ***
poly(dist, 2)1                     -21.33994    2.68168  -7.958 1.75e-15 ***
poly(dist, 2)2                      -2.43042    3.00617  -0.808   0.4188
poly(arsenic, 2)1                   26.09360    2.37766  10.974  < 2e-16 ***
poly(arsenic, 2)2                  -11.68443    2.31552  -5.046 4.51e-07 ***
poly(dist, 2)1:poly(arsenic, 2)1   -91.79873  134.61008  -0.682   0.4953
poly(dist, 2)2:poly(arsenic, 2)1   109.52318  163.80153   0.669   0.5037
poly(dist, 2)1:poly(arsenic, 2)2    90.24115  148.07164   0.609   0.5422
poly(dist, 2)2:poly(arsenic, 2)2 -327.07352  197.60501  -1.655   0.0979 .

> lasso.data <- glmnet(model.matrix(glm.00),data$switch,family="binomial")
> ## glmnet defaults to alpha=1 (lasso penalty)
> plot(lasso.data,"lambda",label=T,col=1:8)
> legend("bottomright",legend=dimnames(model.matrix(glm.00))[[2]][2:9],lty=1,col=1:8)
> ## plot on next page...
```

These are the same predictors as glm.3

Variable Selection for Arsenic Data…

- A fairly stable part of the plot is around log(lambda)=-4
- The variables remaining here are
  - poly(dist,2) 1: dist
  - ploy(arsenic,2) 1: arsenic
  - poly(arsenic,2) 2: arsenic^2

  i.e. glm.3 variables again…

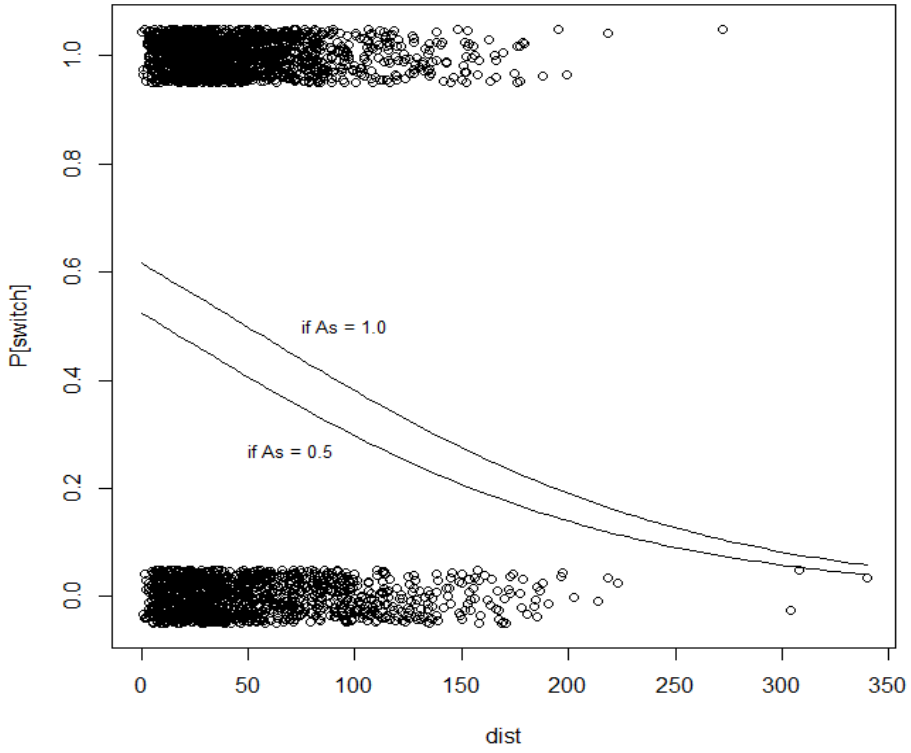



```
> summary(glm.3)
                 Est      SE    z      Pr(>|z|)
(Intercept)   -0.319   0.107  -2.993   0.00277 **
dist          -0.010   0.001  -9.079   < 2e-16 ***
arsenic        0.879   0.101   8.717   < 2e-16 ***
I(arsenic^2) -0.087   0.018  -4.765 1.89e-06 ***
```

- We haven't tried "assoc" or "educ" yet, but let's see what the model is telling us…

Illustrating glm.3…

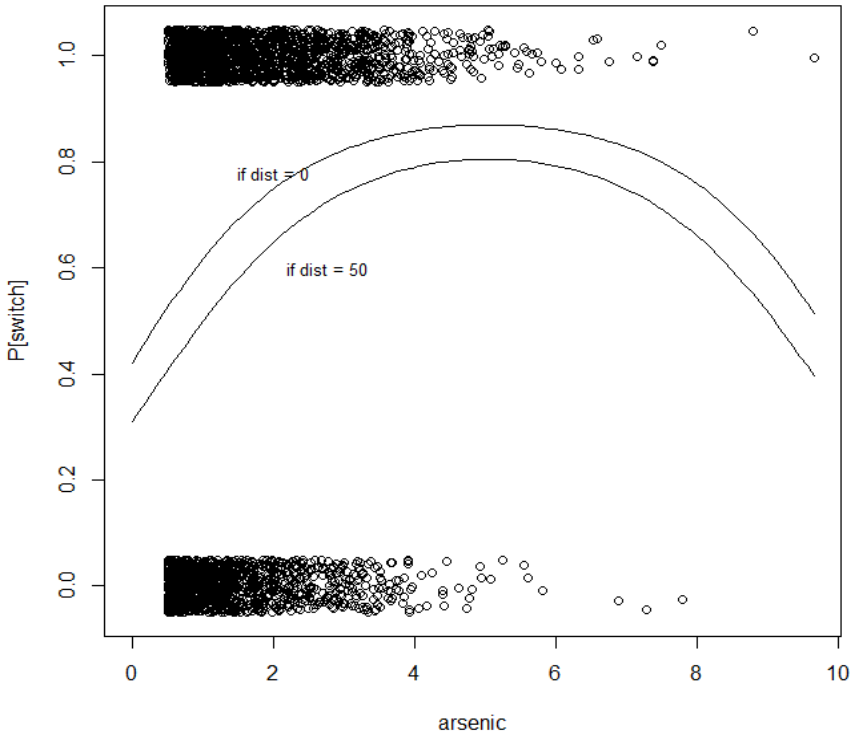
```
attach(data)
plot(dist,jitter(switch,factor=.25),
     xlim=c(0,max(dist)),ylab="P[switch]")
curve (invlogit(cbind (1, x, 0.5, 0.5^2) %*% coef(glm.3)), add=TRUE)
curve (invlogit(cbind (1, x, 1.0, 1.0^2) %*% coef(glm.3)), add=TRUE)
text (50, .27, "if As = 0.5", adj=0, cex=.8)
text (75, .50, "if As = 1.0", adj=0, cex=.8)
```



This looks pretty good. At each fixed level of arsenic, the probability of switching wells decreases as the distance to the nearest "safe well" increases.

Illustrating glm.3…

```
> plot(arsenic,jitter(switch,factor=.25),
+      xlim=c(0,max(arsenic)),ylab="P[switch]")
> curve (invlogit(cbind (1, 0,  x, x^2) %*% coef(glm.3)), add=TRUE)
> curve (invlogit(cbind (1, 50, x, x^2) %*% coef(glm.3)), add=TRUE)
> text (1.5, .78, "if dist = 0", adj=0, cex=.8)
> text (2.2, .6, "if dist = 50", adj=0, cex=.8)
> detach(data)
```



This seems worrisome… it doesn't make sense that the probability of switching wells should go down as arsenic increases from 6 to 10.

It suggests that perhaps we should

- Drop the arsenic^2 term after all, or
- Look for arsenic^3, etc. to try to get the functional form right, or
- Talk to researchers to see if there is a substantive reason this makes sense afer all…

Summary

- Logistic Regression
- Interpreting the Coefficients
- Example: Extract from the Coleman Report
- Improving the Model
- Overfitting and Identifiability
- Effect of Dichotomization
- Assessing Residuals
- Example: Wells in Bangladesh
  - A simple model
  - Variable selection