Logistic Regression

Response is now a Bernoulli RV

$y_i \sim Bernoulli(p_i)$

p_i changes with predictors. Is variance constant? We model the p_i's.

What is a residual now?

$logit(p_i) = \log(\frac{p_i}{1-p_i}) = \mathbf{X}\beta$ where $p_i = Pr(y_i = 1)$ and

Inverse logit transformation:

$p_i = \frac{exp(\mathbf{X}\beta)}{1+exp(\mathbf{X}\beta)}$ must fall in (0,1)

This is a form of a generalized linear model (Chapter 6)

Set up NES data

Note: load arm after xtable so display works properly.

```
brdata <- read.csv("http://www.math.montana.edu/~jimrc/classes/stat505/data/nes
elections <- subset(brdata, complete.cases(black, female, educ1, age,
    income, state) & year %in% 1952:2000)
elections$year.new <- unclass(factor((elections$year)))
elections$c.income <- unclass(elections$income) - 3
elections$income <- elections$c.income + 3
elections$c.age <- (elections$age - mean(elections$age))/10
names(elections)[49] <- "y"
nes.year <- elections[, "year"]
elections$cut.age <- unclass(cut(elections$age, c(0, 29.5, 44.5, 64.5,
    200)))
elections$race.adj <- ifelse(unclass(elections$race) >= 3, 1.5, unclass(electio
elections$presvote <- unclass(elections$presvote)
elections$rvote <- with(elections, ifelse(presvote == 1, 0, ifelse(presvote ==
    2, 1, NA)))
region.codes <- c(3, 4, 4, 3, 4, 4, 1, 1, 5, 3, 3, 4, 4, 2, 2, 2, 2,
    3, 3, 1, 1, 1, 2, 2, 3, 2, 4, 2, 4, 1, 1, 4, 1, 3, 2, 2, 3, 4,
    1, 1, 3, 2, 3, 3, 4, 1, 3, 4, 1, 2, 4)
plotyear <- unique(sort(elections$year))
n.year <- max(elections$year.new)
```


Simple Logistic Regression

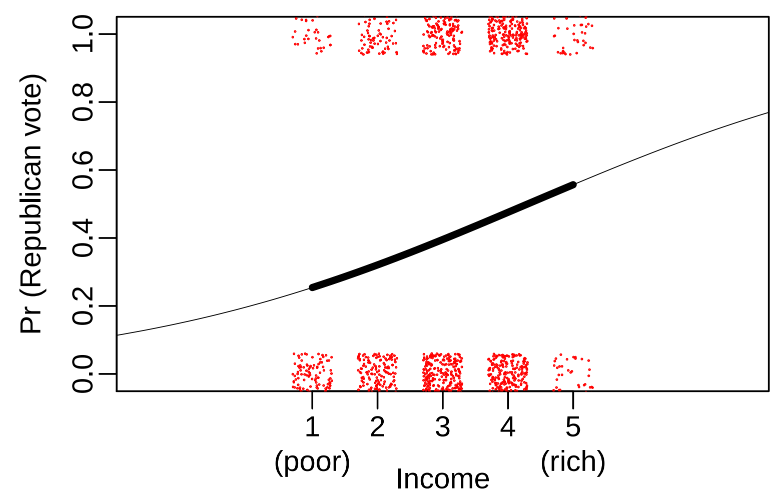
```
vote.fit1 <- glm(rvote ~ income, data = elections, family = binomial(link = "lo
    subset = year == 1992 & presvote < 3)
```

	Estimate	Std. Error	z value
(Intercept)	-1.40	0.19	-7.40
income	0.33	0.06	5.73

Table: n = 1179 rank = 2 Resid Deviance = 1556.916

```
curve(invlogit(cbind(1, x) %*% vote.fit1$coef), 1, 5, xlim = c(-2,
    8), ylim = c(-0.01, 1.01), xaxt = "n", xaxs = "i", mgp = c(2, 0.5,
    0), ylab = "Pr (Republican vote)", xlab = "Income", lwd = 4)
curve(invlogit(cbind(1, x) %*% vote.fit1$coef), -2, 8, lwd = 0.5, add = T)
axis(1, 1:5, mgp = c(2, 0.5, 0))
mtext(c("(poor)", "(rich)"), side = 1, 1.5, at = c(1, 5), adj = 0.5)
points(jitter(rvote, 0.3) ~ jitter(income, 1.5), data = elections,
    subset = year == 1992 & presvote < 3, pch = 20, col = 2, cex = 0.1)
```

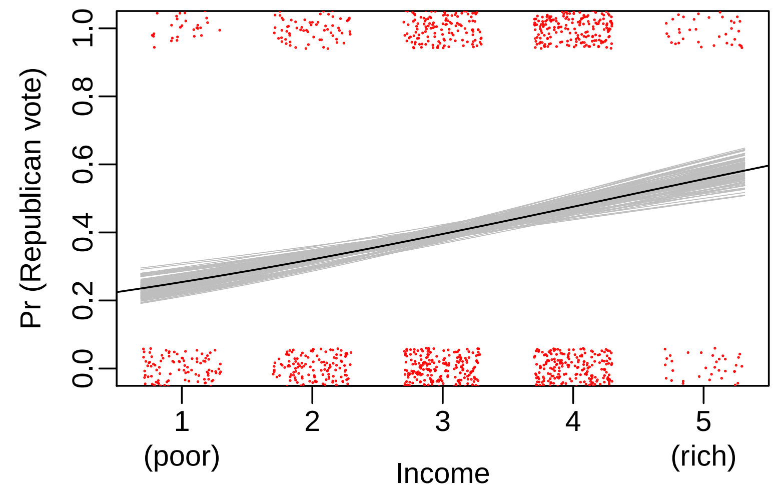

Plot 5.1a



Simulate Logistic Regression

```r
set.seed(24354)
sim.1 <- sim(vote.fit1)  ## plot sims as gray background
curve(invlogit(cbind(1, x) %*% vote.fit1$coef), 0.5, 5.5, ylim = c(-0.01,
    1.01), xlim = c(0.5, 5.5), xaxt = "n", xaxs = "i", mgp = c(2, 0.5,
    0), ylab = "Pr (Republican vote)", xlab = "Income", lwd = 1)
for (j in 1:100) {
    curve(invlogit(cbind(1, x) %*% sim.1@coef[j, ]), col = "gray",
        lwd = 0.5, add = T)
}  ## plot the  estimated fit in black on top
curve(invlogit(cbind(1, x) %*% vote.fit1$coef), add = T)
axis(1, 1:5, mgp = c(2, 0.5, 0))
mtext("(poor)", 1, 1.5, at = 1, adj = 0.5)
mtext("(rich)", 1, 1.5, at = 5, adj = 0.5)  ## add points
points(jitter(rvote, 0.3) ~ jitter(income, 1.5), data = elections,
    subset = year == 1992 & presvote < 3, pch = 20, col = 2, cex = 0.1)
```


Plot 5.1b



Interpret Coefficients

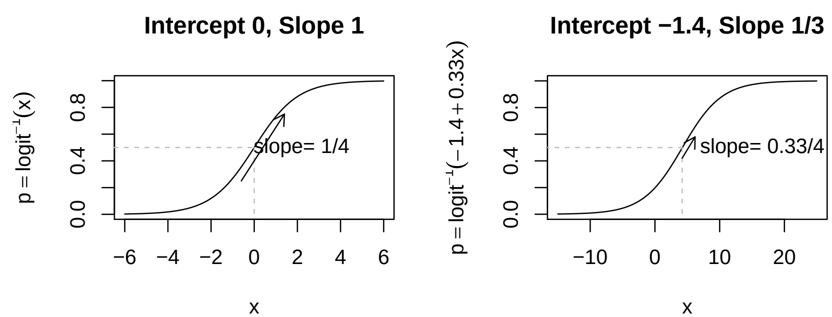
For what x does $\widehat{p} = \text{logit}^{-1}(b_0 + b_1 x) = .5$?
$\Longleftrightarrow \text{logit}(.5) = \log(1) = b_0 + b_1 x$

Slope of inverse logit? $\frac{\partial e^{b_0+b_1x}(1+e^{b_0+b_1x})^{-1}}{\partial x} =$

$$\frac{b_1 e^{b_0+b_1x}(1 + e^{b_0+b_1x}) - b_1 e^{2(b_0+b_1x)}}{(1 + e^{b_0+b_1x})^2} = b_1 \frac{e^{b_0+b_1x}}{(1 + e^{b_0+b_1x})^2}$$

Which is $b_1/4$ when $\widehat{p} = .5$ (i.e. $\text{logit}(\hat{p}) = 0$).

Figure 5.2



On the left: Change x from 0 to 0.4 to get increase from .5 to .6 in inverse logit. Or from 2.2 to 2.6 to change in probability of .90 to .93. By symmetry, -2.6 to -2.2 changes us from _____ to _____.

NES coefficients

For income categories, $x = 1, \ldots, 5$. What is predicted probability of voting for Bush?

```
invlogit(1:5 * 0.33 - 1.4)

## [1] 0.255 0.323 0.399 0.480 0.562
```

Or use $\overline{x} = 3.1$, $\text{logit}^{-1}(-1.40 + .33(3.1)) = 0.40$
Using derivative, change for 1 unit x near midpoint is

```
0.33 * exp(-1.4 + 0.33 * 3.1)/(1 + exp(-1.4 + 0.33 * 3.1))^2

## [1] 0.0796
```

Close to "divide by 4" (conservative)

Odds ratios

$\frac{p}{1-p}$ is the odds. Logit is "log odds".

Odds ratio is the ratio of two odds, for instance odds of voting for Bush in high income group versus odds of the same event in low income group.

Increasing x by 1 unit changes log odds by b_1 or odds by a factor of e^{b_1}.
Here that's $e^{.33} = 1.4$, odds increase multiplicatively as $e^{.33\Delta_x}$.
Log(odds ratio) in P(vote for Bush) from low to high income group is estimated as $4 \times .33 = 1.33$, so the odds ratio is $e^{1.33} = 3.79$

Estimation

Maximize the binomial likelihood to estimate $\boldsymbol{\beta}$.

As $n \to \infty$ these estimators have a normal distribution. (Figure 5.3)
Asymptotic standard errors are available.

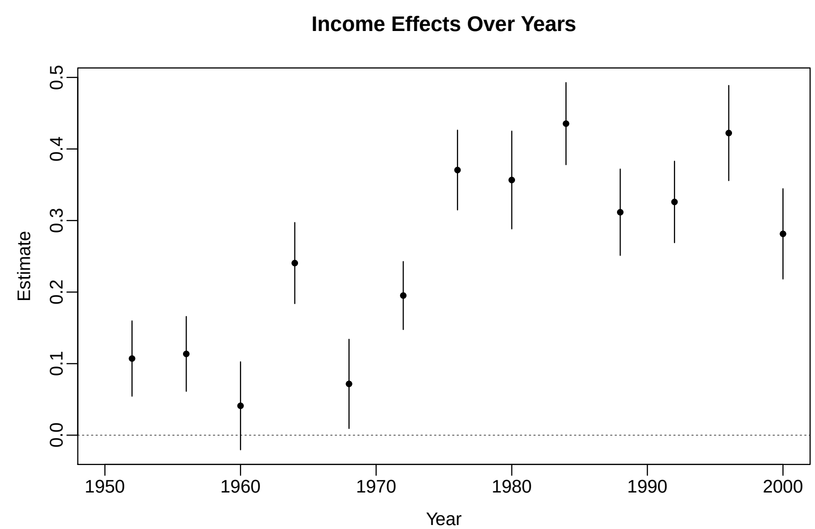
Wald test for $H_0: \ \beta_i = 0$ uses a normal distribution to evaluate how far $\frac{\widehat{\beta}_i}{SE(\widehat{\beta}_i)}$ is from 0. (Not ideal for small samples)

As with MLR, each ratio and p-value is conditional on all other variables being in the model.

Note use of sim() to generate other possible curves from the same population of data. (Figure 5.1b)

Compare fits across years

Plot shows coefficient estimate and ± 1 SE for logistic regressions done on each year.

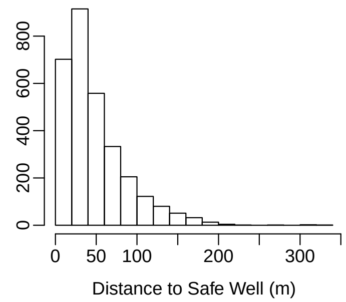


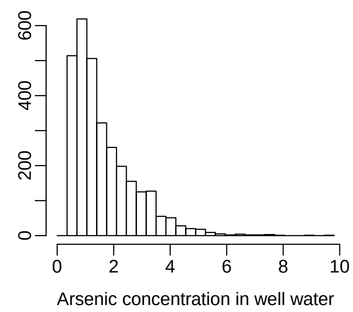
Arsenic Wells in Bangladesh

- Arsenic contamination is prevalent
- Cleaner wells are available – at some distance
- Wells were tested, owners notified if over 50μg/l and asked to switch
- Several years later: Did they switch?

Available predictors: Arsenic (As) concentration, distance to cleaner well, activity in community, education level of head of household.

Distance and Arsenic Histograms





Fit first model

Probability of switching modeled by distance to safe well.

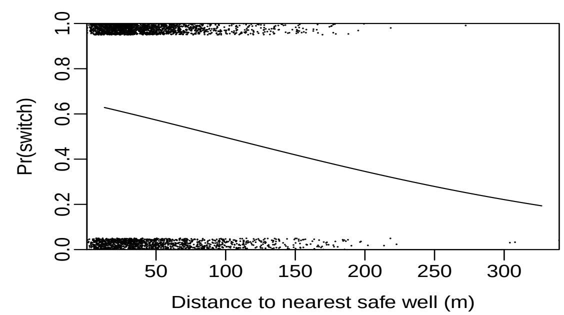
```
binary.jitter <- function(a, jitt = 0.05) {
    jitter <- runif(length(a), 0, jitt)
    a + (a == 0) * jitter - (a == 1) * jitter
}
wells.fit1 <- glm(switch ~ I(dist/100), data = wells, family = binomial)
```

	Estimate	Std. Error	z value
(Intercept)	0.61	0.06	10.05
I(dist/100)	-0.62	0.10	-6.38

Table: n = 3020 rank = 2 Resid Deviance = 4076.238

Wells with Jitter

```
plot(switch ~ dist, data = wells, type = "n", xaxs = "i", yaxs = "i",
    mgp = c(2, 0.5, 0), ylab = "Pr(switch)", xlab = "Distance to nearest safe w
curve(invlogit(cbind(1, x/100) %*% coef(wells.fit1)), add = T)
points(wells$dist, binary.jitter(wells$switch), pch = 20, cex = 0.1)
```



Interpretation 1

$\widehat{P(\text{switch})} = \text{logit}^{-1}(0.61 - 0.62 \times \text{distance}/100)$

- If distance $= 0$, the probability is $\text{logit}^{-1}(0.61) = .65$
- Average distance is 48m. At that point, $\widehat{p} = \text{logit}^{-1}(0.61 - 0.62 \cdot 0.48) = 0.31$ with slope $(-0.62)(0.31)(0.69) = -0.15$
- Divide by 4 rule agrees, $-0.62/4 = -0.155$

Fitted Model 2

```r
wells.fit3 <- update(wells.fit1, . ~ . + arsenic)
display.xtable(wells.fit3)
```

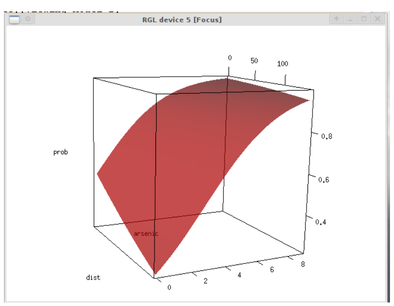
	Estimate	Std. Error	z value
(Intercept)	0.00	0.08	0.03
I(dist/100)	-0.90	0.10	-8.59
arsenic	0.46	0.04	11.13

Table: n = 3020 rank = 3 Resid Deviance = 3930.668

Slopes: -0.22 and 0.11 are not close to zero.
To compare strengths, multiply by $s_{dist/100} = 0.38$ or $s_{As} = 1.10$ to get slope changes of -0.34 and 0.51 for a 1 SD change in x.

Viewing Model 2

```r
dist <- seq(0, 160, 5)
arsenic <- seq(0.5, 9, 0.2)
prob <- outer(dist/100, arsenic, function(x, y) invlogit(-0.9 * x +
    0.46 * y))  ## outer applies function to each combination of (x,y)
require(rgl)
open3d()
persp3d(dist, arsenic, prob, col = "red", alpha = 0.7)
```



Interaction Model

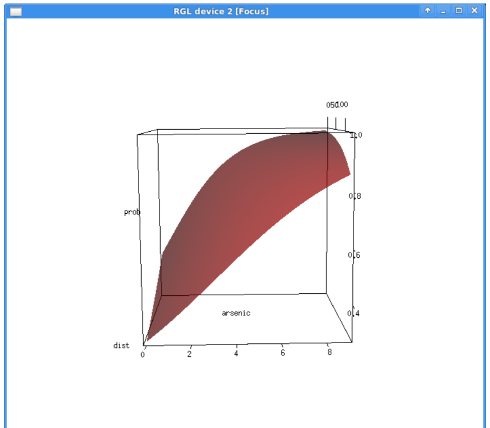
```r
wells.fit4 <- update(wells.fit3, . ~ .^2)
display.xtable(wells.fit4)
```

	Estimate	Std. Error	z value
(Intercept)	-0.15	0.12	-1.26
I(dist/100)	-0.58	0.21	-2.76
arsenic	0.56	0.07	8.02
I(dist/100):arsenic	-0.18	0.10	-1.75

Table: n = 3020 rank = 4 Resid Deviance = 3927.628

Viewing Model 3

```
dist <- seq(0, 160, 5)
arsenic <- seq(0.5, 9, 0.2)
prob <- outer(dist/100, arsenic, function(x, y) invlogit(-0.15 - 0.58 *
    x + 0.56 * y - 0.18 * x * y))
open3d()
persp3d(dist, arsenic, prob, col = "red", alpha = 0.7)
rm(arsenic)
```



Interpretation 2

- Use mean distance/100 of .48, mean arsenic of 1.66
- Constant: $\text{logit}^{-1}(-.15) = 0.47$, but dist = 0 is uninteresting, and arsenic is at least .5. At the center we estimate P(switch) as $\text{logit}^{-1}(-.15 - .58(.48) + .56(1.66) - .18(.48)(1.66)) = 0.59$
- Distance changes by 100m when arsenic =0 is not of interest. At arsenic $= 1.66$, coefficient is $-.58 - .18(1.66) = -0.88$ divide by 4 to get -.22. Increase of 100m when arsenic is at mean level decreases P(switch) by 22%
- Arsenic when distance $= 58$m: slope is $[0.56 - 0.18(0.58)]/4 = 0.12$
- Interaction: Change in arsenic coef. when distance goes up 100m, or change in dist coef when arsenic goes up by 1.

Centering

```
c.dist100 <- with(wells, dist - mean(dist))/100
c.arsenic <- with(wells, arsenic - mean(arsenic))
wells.fit5 <- update(wells.fit4, . ~ c.dist100 * c.arsenic
display.xtable(wells.fit5)
```

	Estimate	Std. Error	z value
(Intercept)	0.35	0.04	8.81
c.dist100	-0.87	0.10	-8.34
c.arsenic	0.47	0.04	11.16
c.dist100:c.arsenic	-0.18	0.10	-1.75

Table: n = 3020 rank = 4 Resid Deviance = 3927.628

Interpretation?
Keep the interaction?

Other Predictors

```
educ4 <- wells$educ/4
wells.fit6 <- update(wells.fit5, . ~ . + assoc + educ4)
display.xtable(wells.fit6)
```

	Estimate	Std. Error	z value
(Intercept)	0.20	0.07	2.92
c.dist100	-0.88	0.11	-8.33
c.arsenic	0.48	0.04	11.24
assoc	-0.12	0.08	-1.60
educ4	0.17	0.04	4.37
c.dist100:c.arsenic	-0.16	0.10	-1.58

Table: n = 3020 rank = 6 Resid Deviance = 3905.351

Other interactions?

Drop assoc

```
c.educ4 <- educ4 - mean(educ4)
wells.fit7 <- update(wells.fit5, . ~ (c.dist100 + c.arseni
display.xtable(wells.fit7)
```

	Estimate	Std. Error	z value
(Intercept)	0.36	0.04	8.84
c.dist100	-0.90	0.11	-8.41
c.arsenic	0.49	0.04	11.50
c.educ4	0.18	0.04	4.72
c.dist100:c.arsenic	-0.12	0.10	-1.14
c.dist100:c.educ4	0.32	0.11	3.03
c.arsenic:c.educ4	0.07	0.04	1.65

Table: n = 3020 rank = 7 Resid Deviance = 3891.744

Lessons

Keep educ interactions?
Centering worked well because 0 values were not of interest.
Standardizing (÷ by 2SD) also would help with interpretation.

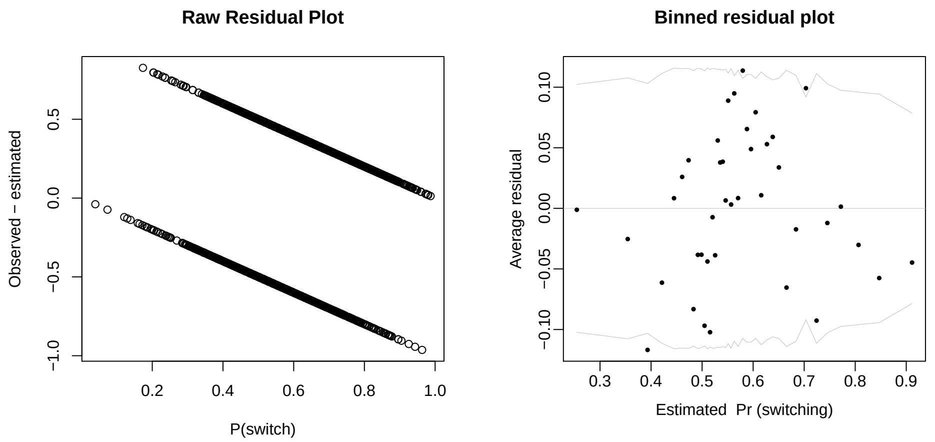
Checking, Comparing Logistic Regressions

R code to bin the residuals

```
## Residual Plot (Figure 5.13 (a))
par(mfrow = c(1, 2))
plot(pred.8, wells$switch - pred.8, ylab = "Observed - estimated",
    xlab = "P(switch)", main = "Raw Residual Plot")

br.8 <- binned.resids(pred.8, wells$switch - pred.8, nclass = 40)
plot(range(br.8[, 1]), range(br.8[, 2], br.8[, 6], -br.8[, 6]), xlab = "Estimat
    ylab = "Average residual", type = "n", main = "Binned residual plot",
    mgp = c(2, 0.5, 0))
abline(0, 0, col = "gray", lwd = 0.5)
lines(br.8[, 1], br.8[, 6], col = "gray", lwd = 0.5)
lines(br.8[, 1], -br.8[, 6], col = "gray", lwd = 0.5)
points(br.8[, 1], br.8[, 2], pch = 19, cex = 0.5)
```

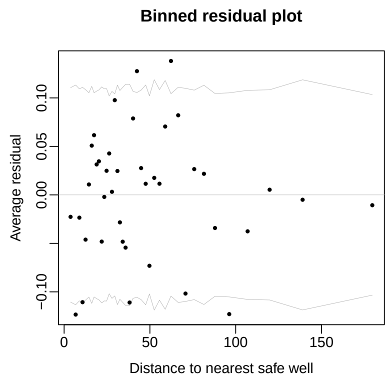

Binned Residuals Fig 5.13

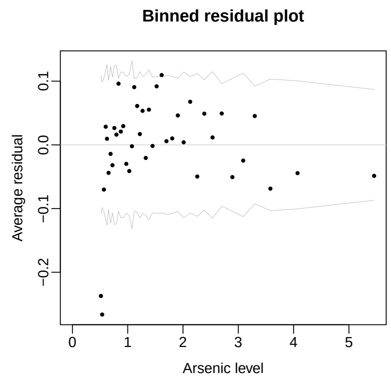


Fitted values are either 0 or 1, so residuals are weird.
Binned residuals split the residuals up into groups (here 40 bins of about 75 each) and average residuals within each group.

Binned Residuals vs Distance and Arsenic

Look at Distance effects on binned residuals.





Distance fits well, but As fits poorly at low levels. Needs some sort of curvature or transformation. Try log(As).

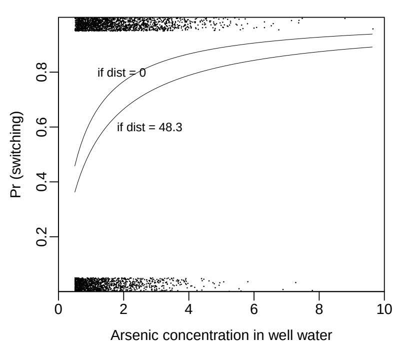
Final fit

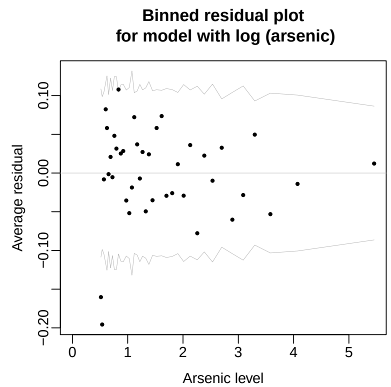
```
c.log.arsenic <- log(wells$arsenic) - mean(log(wells$arsen
wells.fit9 <- update(wells.fit8, . ~ (c.dist100 + c.log.ars
display.xtable(wells.fit9)
```

	Estimate	Std. Error	z value
(Intercept)	0.35	0.04	8.53
c.dist100	-0.98	0.11	-8.81
c.log.arsenic	0.90	0.07	13.00
c.educ4	0.18	0.04	4.58
c.dist100:c.log.arsenic	-0.16	0.19	-0.85
c.dist100:c.educ4	0.34	0.11	3.14
c.log.arsenic:c.educ4	0.06	0.07	0.85

Table: n = 3020 rank = 7 Resid Deviance = 3863.108

Final Plots





Error Rates

Did we predict correctly?
How well did we do versus guessing with the more popular outcome?

	Prediction	
Observed	0 because $\widehat{P(y=1)} < c$	1 because $\widehat{P(y=1)} >= c$
0	Correct	overshot
1	undershot	Correct

```
errorRate <- function(y, pred, c) {
    sum(y != (pred >= c))/length(y)
}
c(errorRate(wells$switch, pred.9, 0.5), errorRate(wells$switch, 1,
    0.5))

## [1] 0.365 0.425
```

Error rate of 42% for always predicting "switch" versus 36.5% for the model (Set $c = .495$ gives 36.2%).
Not impressive.

Deviance

For regular linear models, SSE is also -2log(likelihood).
GLMs define deviance as -2log(likelihood).

- Lower is better
- Adding a "random noise" predictor lowers expected deviance by 1.
- Adding a useful predictor (k of them) should lower deviance by more than 1 (k).
- Asymptotical comparison via χ^2 dist with k df.

```
xtable(anova(wells.fit5, update(wells.fit6, . ~ . - assoc)
```

	Resid. Df	Resid. Dev	Df	Deviance
1	3016	3927.63		
2	3015	3907.91	1	19.72
3	3014	3905.35	1	2.56

Average Prediction for 100m Distance effect

Examine changes on the probability scale while shifting one variable by some amount and sampling others from the population.

```
wells.fit10 <- update(wells.fit3, . ~ . + educ4)
```

	Estimate	Std. Error	z value
(Intercept)	-0.21	0.09	-2.30
I(dist/100)	-0.90	0.10	-8.56
arsenic	0.47	0.04	11.26
educ4	0.17	0.04	4.47

Table: n = 3020 rank = 4 Resid Deviance = 3910.433

```
delta <- invlogit(cbind(1, 1, Arsenic, educ4) %*% coef(wells.fit10)) -
    invlogit(cbind(1, 0, Arsenic, educ4) %*% coef(wells.fit10))
c(mean(delta), sd(delta))

## [1] -0.2045  0.0271
```


Average Prediction arsenic and education

Now shift arsenic from 1 to .5

```
delta <- invlogit(cbind(1, Dist100, 1, educ4) %*% coef(wells.fit10)) -
    invlogit(cbind(1, Dist100, 0.5, educ4) %*% coef(wells.fit10))
c(mean(delta), sd(delta))

## [1] 0.0564 0.0037
```

and education from 3 to 0.

```
delta <- invlogit(cbind(1, Dist100, Arsenic, 3) %*% coef(wells.fit10)) -
    invlogit(cbind(1, Dist100, Arsenic, 0) %*% coef(wells.fit10))
c(mean(delta), sd(delta))

## [1] 0.1167 0.0171
```


Average Prediction w interaction

Same shifts in an interaction model. Distance goes from 100m to 0

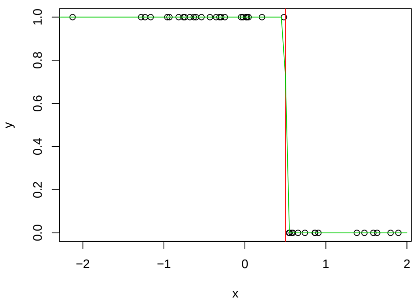
```
wells.fit11 <- update(wells.fit10, . ~ . + I(dist/100):arsenic)
delta <- invlogit(cbind(1, 1, Arsenic, educ4, 1 * Arsenic) %*% coef(wells.fit11
    invlogit(cbind(1, 0, Arsenic, educ4, 0 * Arsenic) %*% coef(wells.fit11))
c(mean(delta), sd(delta))

## [1] -0.1944  0.0201
```

Pay attention to the columns in the **X** matrix to know what is changing.

Problem of Too Good a Predictor

If, as x increases, y suddenly goes from all ones to all zeroes, (or vice versa), logistic regression will fail. Slope tries to go to ∞, and intercept is not pinned down.



This curve uses center at .502 and slope = 100, but slope = 1000 looks just the same.
Good place to use a classification tree.