Chapter 10 Logit Models

Paul E. Johnson

April 3, 2016

1 Glossary

Notation continuing from previous

ξ_{ij}: "xi" total (or combined) individual row-level error term for group i row j,

Longitudinal terminology: "group" is often a single person (j is group of rows) and the rows for each person are differentiated by time (which is i in this book). Economists tend to call these i, and t, whereas Laird-Ware/Bates would call them i j

ζ_j: "zeta" a group level random error, j indexes a grouping variable, $Var(\zeta_j) = \psi$

ϵ_{ij}: "epsilon" $Var(\epsilon_{ij}) = \theta$. Individual row-level error uncorrelated with ζ_j

$Cov(y_{ij}, y_{i'j}|\zeta_j) = 0$. Apart from ζ_j, the observed scores are "conditionally uncorrelated"

2 Quick Overview

1. In a Binomial probablity model,

   a) The expected value of y_i is the same as the probability that $y_i = 1$. Hence

   $$E[y_i|x_i] = Pr(y_i = 1|x_i) \tag{1}$$

   b) The predictor variables are all weighted (slope coefficients) and added together

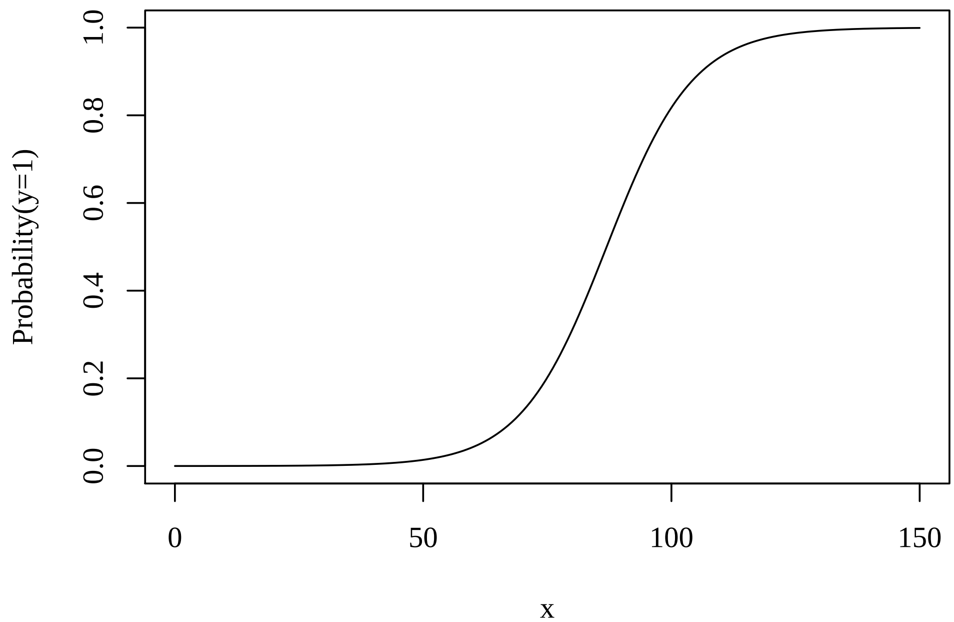
   $$X\beta = 1 + x1_i\beta_1 + x2_i\beta_2 + \ldots \tag{2}$$

   c) Insert any interactions or squares of x you want in there: but there is reason to be cautious. The Logistic functional form automatically creates interactions.

   d) Call $X_i\beta$ "eta" η_i for short. It is "the **linear predictor**".

   e) And then transformed by the "**inverse link function**"

   f) A graph of the logit transformation



g) That particular curve was

$$Prob(y = 1|x) = \frac{1}{1 + e^{-(-10+0.115x)}} \tag{3}$$

more generally:

$$Prob(y = 1|x) = \frac{1}{1 + e^{-(X_i\beta)}} \tag{4}$$

That is mathematically equivalent to

$$Prob(y = 1|x) = \frac{e^{X_i\beta}}{1 + e^{X_i\beta}} \tag{5}$$

Writing it as in expression (5) makes it easier to generalize this to a multi-outcome dependent variable.

Note that the probability of getting $y = 0$ is

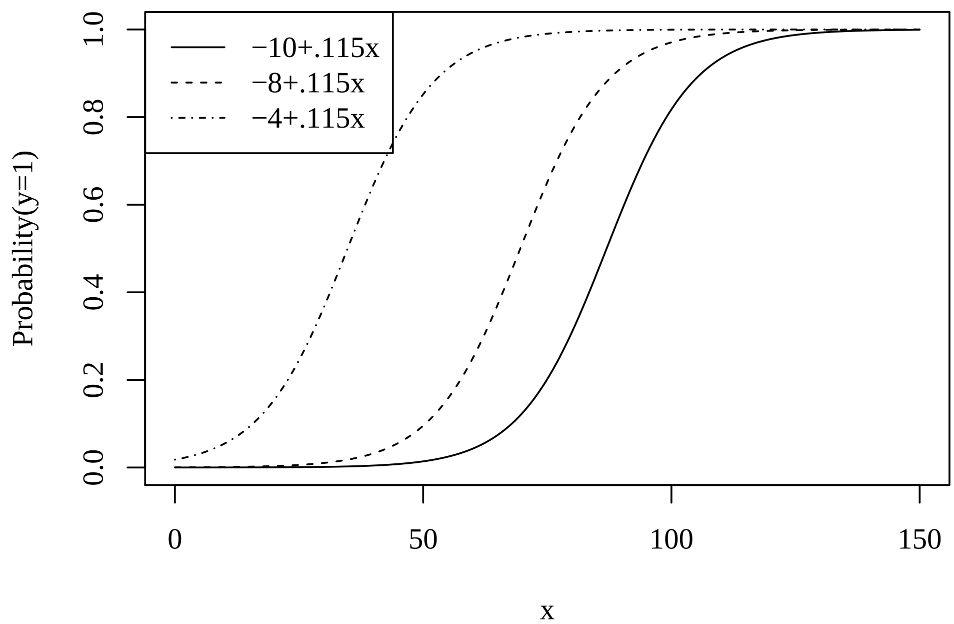
$$1 - \frac{e^{X_i\beta}}{1 + e^{X_i\beta}} = \frac{1}{1 + e^{X_i\beta}} \tag{6}$$

In some formulae for maximum likelihood estimation, then, we can represent the *probability of observing what was actually found* as

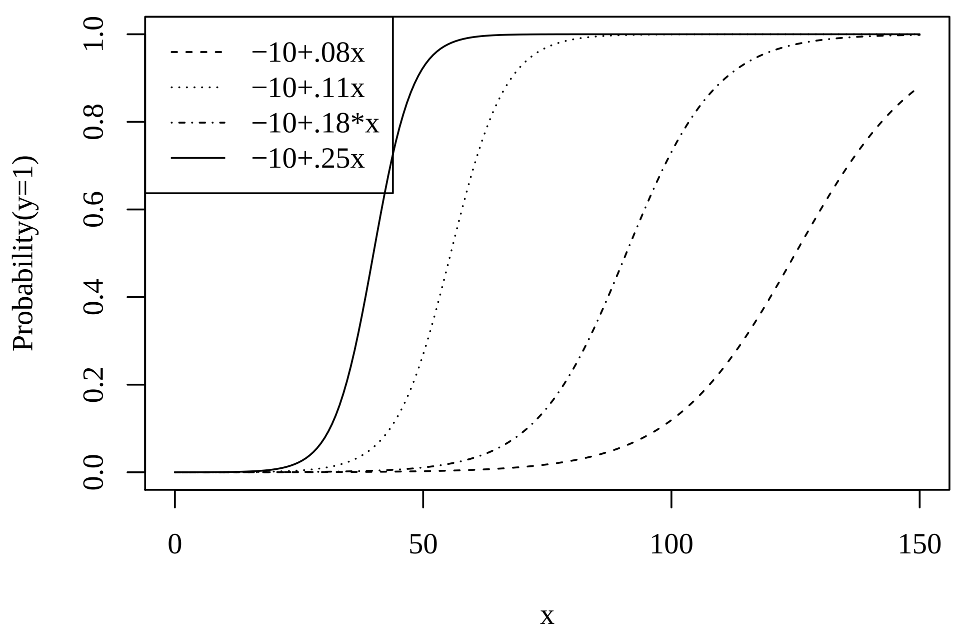
$$\frac{\{e^{X_i\beta}\}^{y_i}}{1 + e^{X_i\beta}} \tag{7}$$

Note if $y = 1$, this gives back (5) but if $y = 0$ it gives back (6).

h) The effect of adding a "dummy variable" (same as shifting the intercept, yes?). The S shaped curve appears to shift horizontally.

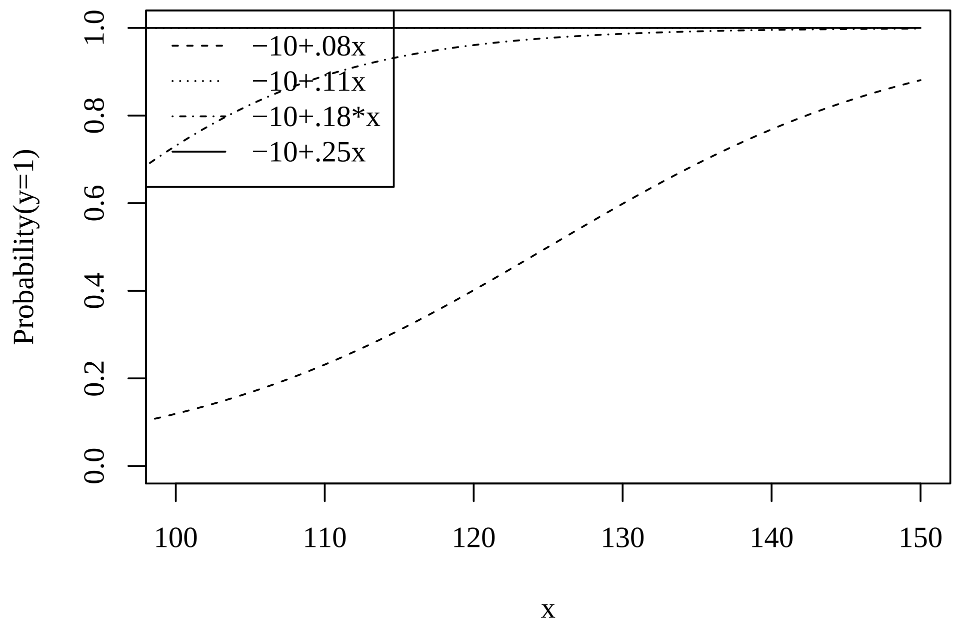


i) The effect of adjusting the slope

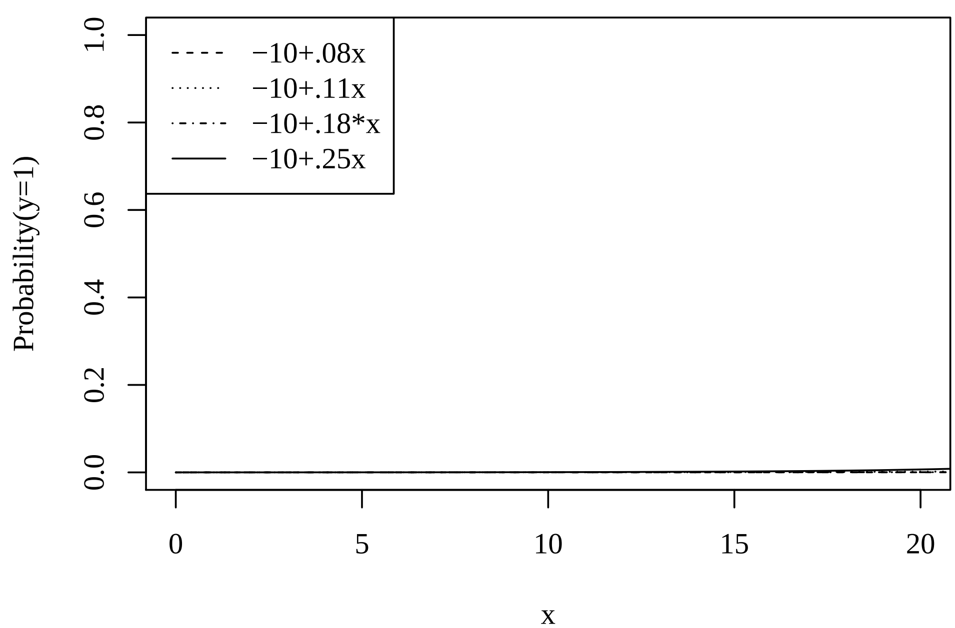


j) The range of observed x tends to be very important in empirical applications.

   i. Suppose we only observe when $x > 100$.



ii. Or $0 < x < 20$.



k) A change in the "intercept" or the "slope" has a change that is much more complicated and hard to understand than an ordinary linear regression model.

i. A change in the "intercept" does not really change the intercept at all, it shifts the S-shaped curve from side to side

ii. A change in the "slope" has a counter-intuitive effect. If the slope is IN-

CREASED, than a change in x has a larger effect for *just a small piece of the domain of* x and for the rest of the domain, the effect is flatter and less important than it was.

l) Why do they call this an inverse link function

$$\frac{e^{\eta_i}}{1+e^{\eta_i}} = \frac{1}{1+e^{-\eta_i}}? \tag{8}$$

Answer: McCullagh and Nelder thought of mapping back from the expected value of y_i to get the value of the linear predictor. They conceptualized the transformation as acting on the left hand side, so $g(E[y]) = X_i\beta = \eta_i$

$$ln\left[\frac{Pr(y_i=1|x_i)}{1-Pr(y_i=1|x_i)}\right] = \eta_i \tag{9}$$

Simplify some notation, that thing on the left is the "log of the odds ratio"

$$ln\left[\frac{P_i}{1-P_i}\right] = \eta_i = X_i\beta \tag{10}$$

where I make typing easier by $P_i = Pr(y_i = 1|x_i)$.

The formula on the left is called the "**logit transform**"

$$logit(P_i) = ln\left(\frac{P_i}{1-P_i}\right) \tag{11}$$

m) In the Generalized Linear Model, the thing on the left is the **link function**, frequently symbolized as $g()$.

$$g(P_i) = X_i\beta = \eta_i \tag{12}$$

$$\frac{e^{\eta_i}}{1+e^{\eta_i}} = g^{-1}(\eta_i) \tag{13}$$

The inverse link function goes the other way, from the inputs into the expected value.

$$Pr(y_i = 1) = logitinv(X_i\beta) \tag{14}$$

n) Why do they call this a Binomial probability model?

   i. Superficial answer: Binomial is for the number of "successes" out of N events. You can think of 1 row that is scored 0 or 1 as a failure or success out of 1 draw.

ii. Historically informative: The earliest models of this type were created for **grouped data**, which was presented in the form

N of rats	poison potency	number dead within 1 day
20	0	1
20	0.2	10
20	0.4	12
20	0.6	14
20	0.8	15
20	1.0	18

iii. Any time we have all categorical predictors, it is possible to group the data rows into this kind of format. This is suggested as a way to speed up estimation in RHS, pp. 543.

2. The womenlf example pp. 505. I ran this in R

```
wdir <- "womenlf/"
dat <- readRDS(paste0(wdir, "womenlf.rds"))
library(rockchalk)
dat$workstat.old <- dat$workstat
dat$workstat <- combineLevels(dat$workstat, levs = c("parttime", "fulltime"),
    newLabel = "employed")
```

```
The original levels not work parttime fulltime
have been replaced by not work employed
```

```
m1 <- glm(workstat ~ husbinc + chilpres, data = dat, family = binomial(link = "logit"
    ))
summary(m1)
```

```
Call:
glm(formula = workstat ~ husbinc + chilpres, family = binomial(link = "logit"),
    data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6767  -0.8652  -0.7768   0.9292   1.9970

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.33583    0.38376   3.481   0.0005 ***
husbinc         -0.04231    0.01978  -2.139   0.0324 *
chilprespresent -1.57565    0.29226  -5.391    7e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 356.15  on 262  degrees of freedom
Residual deviance: 319.73  on 260  degrees of freedom
AIC: 325.73

Number of Fisher Scoring iterations: 4
```

```
confint(m1)
```

```
                      2.5 %       97.5 %
(Intercept)      0.60590835  2.115336975
husbinc         -0.08225743 -0.004471769
chilprespresent -2.16143847 -1.012720928
```

My parameter estimates are similar to the RHS book, but the confidence intervals are quite different.

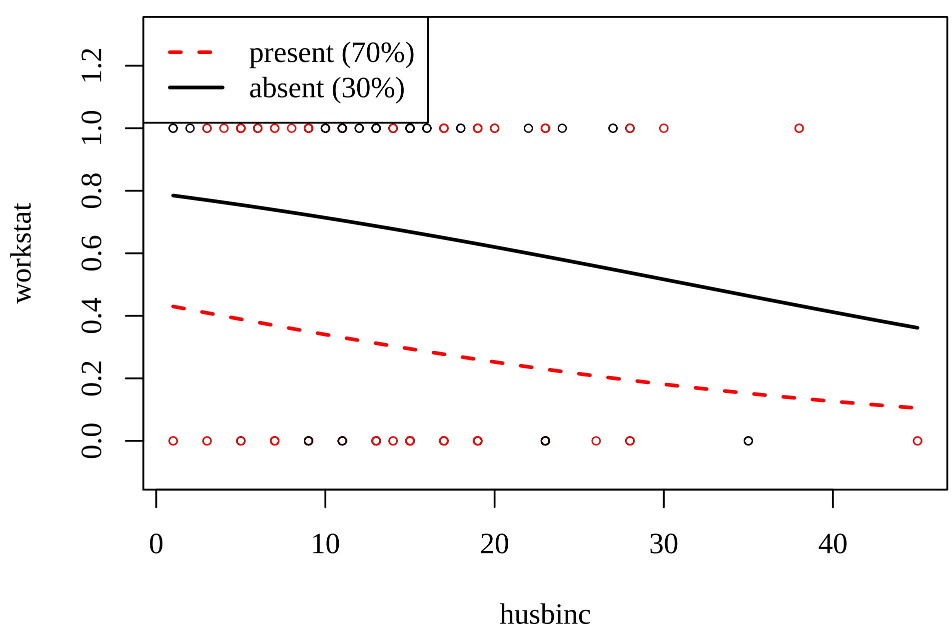
I think this table is nice-enough for now:

```
outreg(list("Logistic Model 1" = m1), tight = TRUE)
```

	Logistic Model 1 Estimate (S.E.)
(Intercept)	1.336***
	(0.384)
husbinc	-0.042*
	(0.020)
chilprespresent	-1.576***
	(0.292)
N	263
Deviance	319.733
$-2LLR(Model\chi^2)$	36.418***

$*p \leq 0.05{*}{*}\ p \leq 0.01{*}{*}{*}p \leq 0.001$

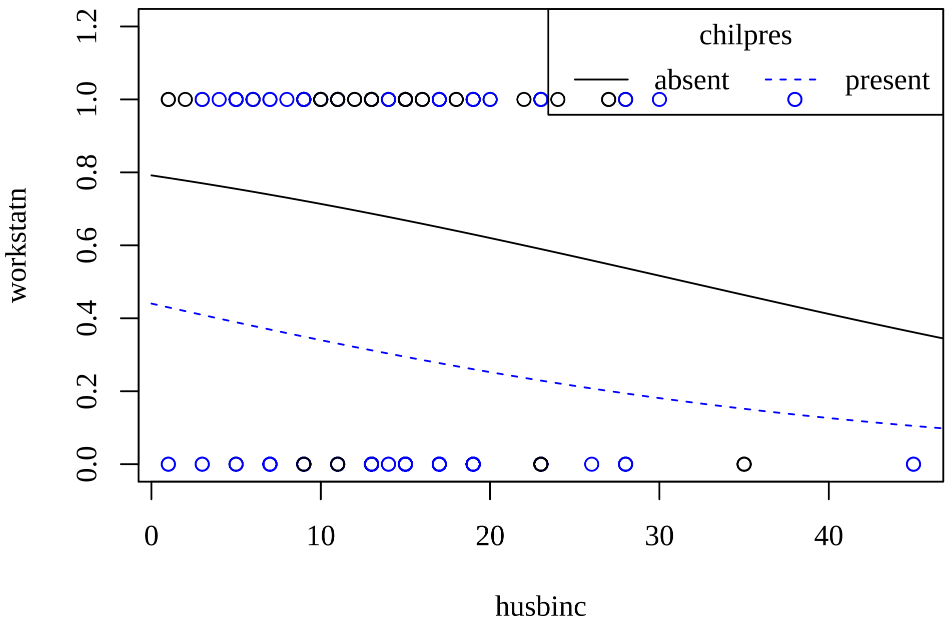
In rockchalk:plotCurves I've got a little bug I need to fix in the plotting of the points.



That's a little humiliating, have to do the old fashioned way

```
dat$workstatn <- ifelse(levels(dat$workstat)[dat$workstat] %in% "employed", 1, 0)
plot(workstatn ~ husbinc, data = dat,
     col = ifelse(levels(chilpres)[chilpres]=="absent", 1, 4), ylim = c(0, 1.2))
pom <- predictOMatic(m1, predVals = list(husbinc = seq(0,50), chilpres = levels(dat$
     chilpres)))
lines(fit ~ husbinc, data = subset(pom, subset = chilpres=="absent"), col = 1)
```

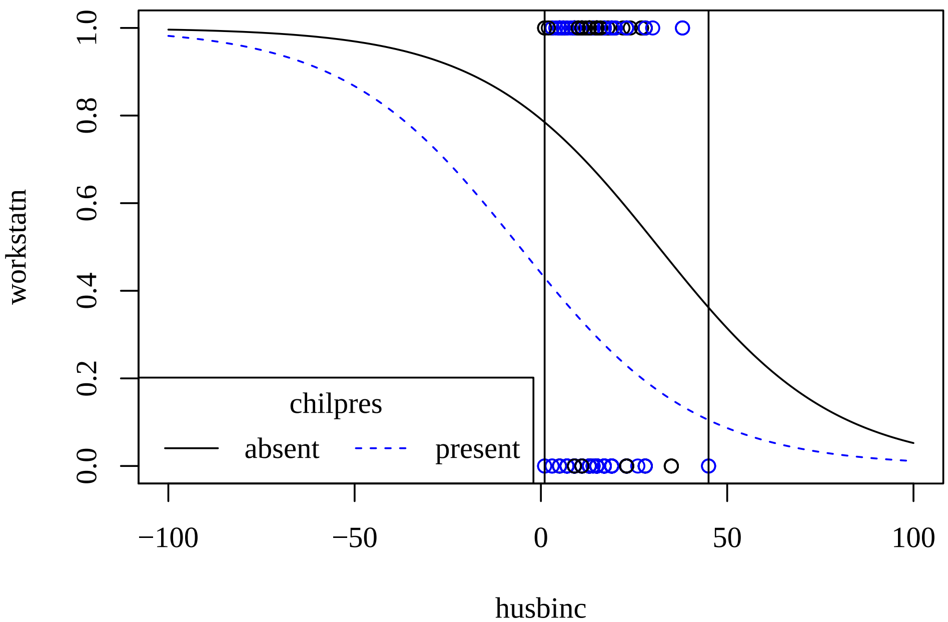
```
lines(fit ~ husbinc, data = subset(pom, subset = chilpres=="present"), lty = 2, col
    = 4)
legend("topright", c("absent", "present"), lty = c(1, 2), col = c(1, 4), title = "
    chilpres", horiz = TRUE)
```



It is worth mentioning "the old fashioned way" because Stata software has also made quite a bit of effort to make predicted value plots "automatically". However, many model types do not cooperate with that margins tool and hence Stata users are also forced to fallback to the old fashioned way. See:

`http://www.ats.ucla.edu/stat/stata/faq/predictive_margins_xtmelogit.htm`

In RHS p. 509, they show same for a wider range of husbinc, I might as well do same. compare to RHS p. 509



3. The Odds Ratio

   Some authors, like RHS, are very enthusiastic about interpreting the odds ratio as an indicator of the impact of a predictor. IMHO, this only makes sense when the predictor is a categorical, two-valued variable, one that can be "dummied up" and recoded as 0 and 1.

   a) But, if you do have a dummy variable coded 0 and 1, it is very easy to see that the effect of changing that predictor from 0 to 1 is $exp(\beta)$.

   b) Definition:

$$Odds = \frac{Pr(y=1)}{1-Pr(y=1)} \tag{15}$$

   c) Suppose I call P_1 the prediction when x=1 and P_0 is probability when x = 0. The change in the log odds ratio

$$\begin{aligned} ln(\frac{P_1}{1-P_1}) - ln(\frac{P_0}{1-P_0}) &= \beta_0 + \beta_1\{1\} - \beta_0 - \beta_1\{0\} \\ ln\left\{\frac{\frac{P_1}{1-P_1}}{\frac{P_0}{1-P_0}}\right\} &= \beta_1 \end{aligned} \tag{16}$$

   so

$$\begin{aligned} exp\left\{ln\left\{\frac{\frac{P_1}{1-P_1}}{\frac{P_0}{1-P_0}}\right\}\right\} &= exp(\beta_1) \\ \frac{\frac{P_1}{1-P_1}}{\frac{P_0}{1-P_0}} &= exp(\beta_1) \\ Odds\,Ratio &= exp(\beta_1) \end{aligned} \tag{17}$$

4. What's the difference in the Probit model? To see this, best to formulate the "**latent response model**".

   a) RHS write the latent model this way

$$y^* = X_i\beta + \epsilon_i$$

$$y = \begin{cases} 1 & \text{if } y* > 0 \\ 0 & y* \le 0 \end{cases} \tag{18}$$

      i. Lets sketch that on the board, the "success" cases (where y = 1) will be on the right side.

   b) I usually write it down with the sign of the error term reversed. (Recall, I write X_i to refer to a ROW vector of data (i'th row))

$$y_i^* = X_i\beta - \epsilon_i \tag{19}$$

   and I think if "success" in the left end of the graph. The binary outcome model can be viewed as an "if then" calculation based on a random variable ϵ_i

$$if\, \epsilon_i < X\beta \qquad then\, y_i = 1 \tag{20}$$

$$if\, X\beta \le \epsilon_i \quad then \quad y_i = 0 \tag{21}$$

   c) Logit or Probit

      i. The logistic distribution for ϵ leads to the "logit model"

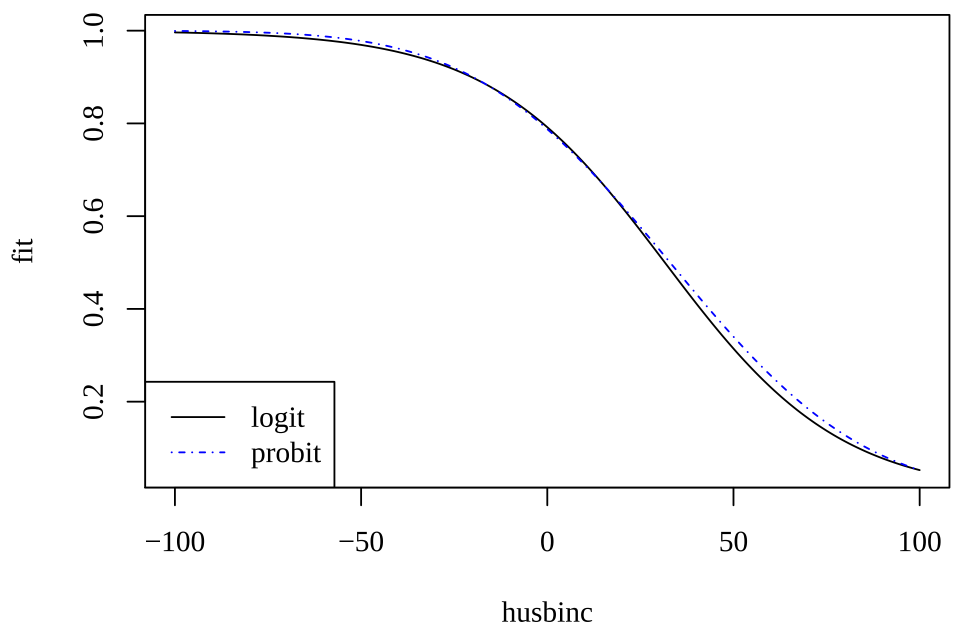
      ii. The Normal distribution for ϵ leads to the "probit model". A probit is an alternative link function that is only very slightly different.

      iii. Here's why I like to write it down this way: It makes it easier to see that we are using the **Cumulative Distribution Function.**

   Insert some sketches here.

   d) Nice graph RHS p. 511

   e) Lets get predicted values from probit and logit and compare



5. The VARIANCE IS **UNIDENTIFIABLE**! There's an unidentifiable coefficient in all of these models. The variance of the error term *cannot be estimated*. It is fixed at a specific value and the coefficients of the variables "re-scale themselves" accordingly. Suppose

$$X_i\beta - \epsilon_i > 0 \tag{22}$$

If we divide through all elements by the standard deviation of ϵ_i, then the inequality still holds

$$X_i\frac{\beta}{\sigma_\epsilon}\beta - \frac{\epsilon_i}{\sigma_\epsilon} > 0 \tag{23}$$

a) That's what it means when a parameter is "unidentifiable", it has no empirical referent.

b) Probit/Logit difference

i. Normal distribution of ϵ : assume variance of ϵ is 1.0 (so the most pleasant $N(0,1)$ is assumed)

ii. Logistic distribution of ϵ: assume variance is $\pi^2/3$. This number is a result of "standardizing" the Logistic so that it has EV = 0 and Var = 1

iii. The estimates of the β are scaled to these values.

iv. Caution about interpreting Logit models because of this un-identified variance parameter:

Mood, C. (2010). Logistic Regression: Why We Cannot Do What We Think We Can Do, and What We Can Do About It. *European Sociological Review*, 26(1), 67–82. http://doi.org/10.1093/esr/jcp006

Allison, Paul D. 1999. Comparing Logit and Probit Coefficients Across Groups. *Sociological Methods and Research* 28:186-208.

c) When calculating the ICC for random effects models, then, we have a weird looking formula,

$$\begin{array}{cc} Logit & Probit \\ \frac{Var(\zeta_j)}{Var(\zeta_j)+\pi^2/3} & \frac{Var(\zeta_j)}{Var(\zeta_j)+1} \end{array} \tag{24}$$

6. Estimation: Maximum Likelihood

a) Choose values of β_0, β_1, β_2, and such that make the sample as likely as possible.

b) For a stipulated set of guesses, one must recalculate the Likelihood function. Suppose we sort the data so the first k observed cases have $y = 0$ and then the rest have $y = 1$.

$$Likelihood = Pr(y_1 = 0) \cdot Pr(y_2 = 0) \cdot \ldots Pr(y_k = 0) \cdot Pr(y_{k+1} = 1) \cdot \ldots Pr(y_N = 1) \tag{25}$$

To avoid working with that giant product, convert that to the log likelihood

$$lnL = lnPr(y_1 = 0) + lnPr(y_2 = 0) + lnPr(y_k = 0) + lnPr(y_{k+1} = 1) + \ldots + lnPr(y_N = 1) \tag{26}$$

$$\sum_{i=1}^{k} ln(Pr(y_i = 0)) + \sum_{k+1}^{N} ln(Pr(y_i = 1)) \tag{27}$$

Recall the trick in equation (7). The probability that a case equals its observed value is $\frac{\{e^{X_i\beta}\}^{y_i}}{1+e^{X_i\beta}}$, so we can write the above more concisely

$$Likelihood = \Pi_{i=1}^{N} \frac{\{e^{X_i\beta}\}^{y_i}}{1+e^{X_i\beta}} \tag{28}$$

Here, I'm using my style of referring to a row from X as X_i and treating it as a row vector.

$$\begin{gathered} logL = \sum_{i=1}^{N} ln\left(\frac{\{e^{X_i\beta}\}^{y_i}}{1+e^{X_i\beta}}\right) \\ \sum ln(exp(X_i\beta)^{y_i} - \sum ln(1+exp(X_i\beta)) \\ \sum y_i X_i\beta - \sum log(1+exp(X_i\beta) \end{gathered} \tag{29}$$

The ML estimation process has to adjust each coefficient in β. For each one, the first order condition is found as

$$\frac{\partial lnL}{\partial \beta_j} = \sum_{i=1}^{N} X_i(y - Pr(y = 1|\beta)) \tag{30}$$

The math to work out the optimization is written out in Hastie, Tibhsirani, Friedman, The Elements of Statistical Learning, 2ed, pp. 120-2

c) Properties of these estimates. ML

   i. Not Unbiased

   ii. Consistent

   iii. Asymptotically Normally distributed

   iv. The variance/covariance matrix of the $\hat{\beta}$ can be derived in various ways. The most common is the "Information matrix". Derived from the second order conditions ("Hessian" matrix) evaluated at the MLE.

d) Hypo testing:

   i. A ratio of $\hat{\beta}_j/s.e.(\hat{\beta}_j)$ is an approximate N(0,1) value, in the sense that if sample $\rightarrow \infty$, then the ratio $\rightarrow N(0, 1)$

   ii. Likelihood Ratio test applies to logit models, in R as anova(), in Stata as lrtest

   iii. Confidence intervals for 95% often calculated as Wald's approximate method $\hat{\beta}_j \pm 1.96s.e.(\hat{\beta}_j)$

7. Calculating predicted values

a) To calculate a predicted value, YOU MUST fill in values for all the x's and all of the β's in here

$$\hat{\eta}_i = \hat{\beta}_0 + \hat{\beta}_1 x1_i + \hat{\beta}_2 x2_i + \ldots \tag{31}$$

   i. and IF you want predicted probabilities after that, apply the inverse link function, say

$$logitinv(\hat{\eta}_i) = logitinv(\widehat{X\beta}) \tag{32}$$

b) Software generally gives you a choice, whether you to investigate the predicted value on the linear predictor scale, $X\hat{\beta}$ or do you want $Prob(y_i = 1|X, \hat{\beta})$.

   i. Stata predict function allows users to choose with options

   xb

   or

   pr

c) To understand the effect of any one variable, IT IS NECESSARY to set the values of all the other predictors in the model.

i. Very common to set all of them at the "mean", but I prefer the mode of categorical predictors (as in rockchalk).

d) Recall the caution about "the old fashioned way" and Stata's margins functions (shortcomings)

e) We did not yet insert random effects, but if we did, it would be necessary to confront one more problem here: what value do you assume for the random effect when you calculate your predictions? If you designate, for example, that there is a unit in which no random effect is experienced (supposing $\zeta_j = 0$), then the predicted value concerns the "population representative case?" in some sense?

f) Could average together estimates of effects from the cases, weighting them by the probability of the random effect. More to come on this.

3 Various fitting functions for models with random effects

1. Stata:

   a) xtlogit: random intercept only. Faster because it has analytical derivatives

   b) xtmlelogit: similar in style to xtmixed, allows request for many random effects.

   c) gllamm: Rabe-Hesketh, Skrondal, & Pickles offer a package for Stata which, as RHS mention, will do some calculations that no other package will do.

   d) Question: why was xtlogit available so long before the other packages were available?

      Answer: When the model is simple enough, the use of numerical approximations can be minimized and more exact analytical formulae can be put into use.

      When we want more random effect estimates, this becomes impossible.

2. R

   a) Before lme4 was released, there were several packages that, like xtlogit, could fit a single random intercept quickly

   b) lme4 was eventually released, it uses a more general algorithm to fit these things.

3. MCMC alternatives

   a) The random effects model allow the inclusion of Normally distributed random effects

   b) Inside a model that assumes errors are Logistic, wouldn't it seem strange to insert a Normal random effect?

4 Random Intercept

1. The random intercept is denoted ζ_j, that's added to the model

2. Two stage formulation

   a) The RHS way, p. 522. I still can't track through their symbols.

   b) My way

   $$logit(P(y=1|X)) = \beta^*_{0j} + \beta_1 treatment_i + \beta_2 visit_i + \beta_3 treatment_i \times visit_i \quad (33)$$

   $$\beta^*_{0j} = \beta_0 + \zeta_{0j} \quad (34)$$

   the linear predictor is easy

   $$\eta_{ji} = \beta_0 + \zeta_{0j} + \beta_1 treatment_i + \beta_2 visit_i + \beta_3 treatment_i \times visit_i \quad (35)$$

   In a case like this, the two level formulation does not simplify or help anything.

   c) Introduce some level 2 predictors of the intercept

   $$logit(P(y=1|X)) = \beta^*_{0j} + \beta_1 treatment_i + \beta_2 visit_i + \beta_3 treatment_i \times visit_i \quad (36)$$

   $$\beta^*_{0j} = \beta_0 + \beta_4 groupPredictor_j + \zeta_{0j} \quad (37)$$

   $$\eta_{ji} = \beta_0 + \beta_4 groupPredictor_j + \zeta_{0j} + \beta_1 treatment_i + \beta_2 visit_i + \beta_3 treatment_i \times visit_i \quad (38)$$

3. Use xtlogit, or xtmelogit, or gllamm.

5 Marginal Model: ignores individual differences, just fits predictive part.

1. See Stata code below, items 6 and 7

2. fits predictive model with outcome ˜ treatment + visit + treatment * visit

   $$logit(P(y=1|X)) = \beta_0 + \beta_1 treatment_i + \beta_2 visit_i + \beta_3 treatment_i \times visit_i \quad (39)$$

3. Generalized Estimating Equations offer a more elaborate way of trying to estimate these things.

4. Nice Comparison of Marginal versus Conditional models

   "The difference between population-averaged and subject-specific effects is due to the average of a nonlinear function not being the same as the nonlinear function of the average." p. 530.

6 BLUPS, or other statements about random effects. (What about the conditional modes?)

1. lme4 in R offers ranef(), which gives guesses about the values of ζ_j.

2. RHS claim there is bad news here: "Such predictions are useful for making inferences for the clusters in the data, important examples being assessment of institutional performance (see section 4.8.5) or of abilities in item response theory (see exercise 10.4). The estimated or predicted values of ζ_j should generally not be used for model diagnostics in random-intercept logistic regression because their distribution if the model is true is not known. In general, the values should also not be used to obtain cluster-specific predicted probabilities (see section 10.13.2)." RHS p. 543-4

3. Another surprise. We know the max likelihood estimates are not good for ζ_j. Here we find that the "Empirical Bayes" and the "Modal Empirical Bayes" calculations are different.

   "The predictions are nearly identical. This is not surprising because marginal effects derived from generalized linear mixed models are close to true marginal effects even if the random-intercept distribution is misspecified (Heagerty and Kurland 2001)." p. 549

4. Although they warn about use of ζ estimates above, they do show how to calculate "Predicted subject-specific probabilities". See Stata code below.

7 Stata Code notes

1. Logit fit, after combining full and part time workers. Recode style RHS p. 205 seems dangerous to me

   ```
   recode workstat 2=1
   logit workstat husbinc chilpres
   ```

   If you want to see the odds ratios instead

   ```
   logit workstat husbinc chilpres , or
   ```

   I hate that one, I'd never do it.

2. Plots

   a) Plot for the observed range of husbinc

   ```
   predict prob, pr
   twoway (line prob husbinc if chilpres==0, sort)
   (line prob husbinc if chilpres==1, sort lpatt(dash)),
   legend(order(1 "No child" 2 "Child"))
   xtitle("Husband's income/$1000") ytitle("Probability that
      wife works")
   ```

b) Plot for a wider range of values of husband income to see the full width of the scale

```
twoway (function y=invlogit(_b[husbinc]*x+_b[_cons]),
    range(-100 100))
(function y=invlogit(_b[husbinc]*x+_b[chilpres]+_b[_cons])
    ,
range(-100 100) lpatt(dash)),
xtitle("Husband's income/$1000") ytitle("Probability that
    wife works")
legend(order(1 "No child" 2 "Child")) xline(1) xline(45)
```

3. Logit models existed before the "generalized linear model" created a broad framework that united many types of models. The Stata "logit" function is an example of the way we were doing this in the olden days, but now it is suggested to thing of it as a GLM and run this instead

```
glm workstat husbinc chilpres, link(logit) family(binomial)
```

4. Probit

```
probit workstat husbinc chilpres
```

Make a nice twoway plot showing the similarity of predicted values from probit and logit. This code requires one to type in values estimated from the model fits. HORRIBLE.

```
twoway (function y=invlogit(1.3358-0.0423*x), range(-100 100))
(function y=normal(0.7982-0.0242*x), range(-100 100) lpatt(
    dash)),
xtitle("Husband's income/$1000") ytitle("Probability that wife
     works")
legend(order(1 "Logit link" 2 "Probit link")) xline(1) xline
    (45)
```

5. Toenail data import

```
xtset patient visit
xtdescribe if outcome < .
```

6. Marginal Data: Two side-by-side barplots of outcomes

```
label define tr 0 "Itraconazole" 1 "Terbinafine"
label values treatment tr
graph bar (mean) proportion = outcome, over(visit) by(
    treatment) ytitle(Proportion with onycholysis)
```

A line plot of the same information using time as the x axis

```
egen prop = mean(outcome), by(treatment visit)
egen mn_month = mean(month), by(treatment visit)
twoway line prop mn_month, by(treatment) sort xtitle(Time in
   months) ytitle(Proportion with onycholysis)
```

7. Marginal Data: Logistic fit, report odds ratios

```
generate trt_month = treatment*month
logit outcome treatment month trt_month, or vce(cluster
   patient)

predict prob, pr
twoway (line prop mn_month, sort) (line prob month, sort lpatt
   (dash)), ///
by(treatment) legend(order(1 "Observed proportions" 2 "Fitted
   probabilities")) ///
xtitle(Time in months) ytitle(Probability of onycholysis)
```

8. Random effects Logistic:

   a) xtlogit

   ```
   quietly xtset patient
   xtlogit outcome treatment month trt_month, intpoints(30)
   xtlogit, or
   ```

   b) xtmelogit

```
xtmelogit outcome treatment month trt_month || patient:,
   intpoints(30)
estimates store xtmelogit
```

9. Using gllamm

```
ssc install gllamm, replace

gllamm outcome treatment month trt_month, i(patient) link(
   logit) family(binomial) nip(30) adapt
estimates store gllamm
gllamm, eform
```

See: p. 542: Advice for speeding up estimation in gllamm

10. Posterior prediction/estimation of random effects

    a) Clearly, we need to make statements about ζ_j. Lets don't quibble about whether to call them estimates or predictions.

b) At the time of writing this book, the only Stata command that provides empirical Bayes predictions for generalized linear mixed models is the postestimation command gllapred for gllamm with the u option: .

```
estimates restore gllamm
gllapred eb, u
```

c) MODE prediction, see RHS 10.12.3

```
estimates restore xtmelogit
predict ebmodal, reffects
predict se2, reses
egen num0 = total(outcome==0), by(patient)
egen num1 = total(outcome==1), by(patient)
list patient num0 num1 ebm1 ebmodal ebs1 se2 if visit==1&
    patient<=12, noobs
```

d) Humph. Empirical Bayes is different from Modal predictions? see. p. 547

```
twoway (rspike mlest ebmodal ebm1 if visit==1)
(scatter mlest ebm1 if visit==1, msize(small) msym(th)
    mcol(black))
(scatter ebmodal ebm1 if visit==1, msize(small) msym(oh)
    mcol(black))
(function y=x, range(ebm1) lpatt(solid)),
xtitle(Empirical Bayes prediction)
legend(order(2 "Maximum likelihood" 3 "Empirical Bayes
    modal"))
```

11. Predicted probabilities:

a) Population averaged

```
estimates restore gllamm
gllapred margprob, mu marginal

twoway (line prob month, sort) (line margprob month, sort
    lpatt(dash)), ///
by(treatment) legend(order(1 "Ordinary logit" 2 "Random-
    intercept logit")) ///
xtitle(Time in months) ytitle(Fitted marginal
    probabilities of onycholysis)
```

b) gllamm predictions of individual probabilities

```
generate zeta1 = 0
gllapred condprob0, mu us(zeta)
generate lower1 = -4
gllapred condprobm4, mu us(lower)
```

```
generate upper1 = 4
gllapred condprob4, mu us(upper)
replace lower1 = −2
gllapred condprobm2, mu us(lower)
replace upper1 = 2
gllapred condprob2, mu us(upper)

twoway (line prop mn_month, sort) ///
(line margprob month, sort lpatt(dash)) ///
(line condprob0 month, sort lpatt(shortdash_dot)) ///
(line condprob4 month, sort lpatt(shortdash)) ///
(line condprobm4 month, sort lpatt(shortdash)) ///
(line condprob2 month, sort lpatt(shortdash)) ///
(line condprobm2 month, sort lpatt(shortdash)), ///
by(treatment) ///
legend(order(1 "Observed proportion" 2 "Marginal
   probability" 3 ///
"Median probability" 4 "Conditional probabilities")) ///
xtitle(Time in months) ytitle(Probabilities of onycholysis
   )
```