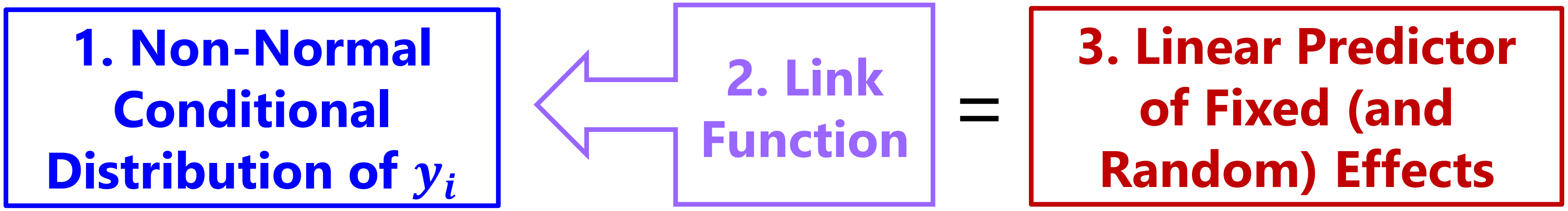
Generalized Linear Models for Binary and Categorical Outcomes

- Topics:
  - 3 parts of a generalized (single-level or multilevel) model
  - Link functions and conditional distributions for binary outcomes
  - Effect sizes for binary outcomes
  - From binary to categorical outcomes
  - Bonus: From generalized linear models to item response theory (IRT) models for categorical outcomes

3 Parts of Generalized Linear Models



1. Non-Normal Conditional Distribution of y_i

2. Link Function

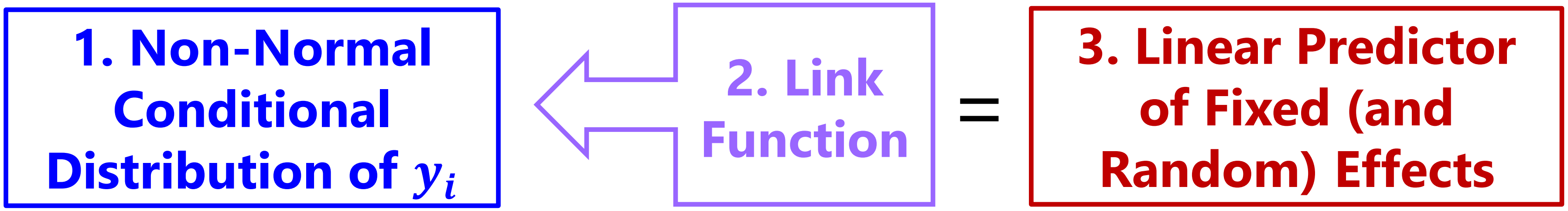
=

3. Linear Predictor of Fixed (and Random) Effects

1. Non-normal conditional distribution of y_i:

- General linear models use a *normal* conditional distribution to describe the y_i variance remaining after prediction via the fixed effects → we called this residual variance, which is estimated separately and usually assumed constant across observations (unless modeled otherwise)
- Other distributions are **more plausible** for categorical/bounded/skewed outcomes, so ML function maximizes the likelihood using those instead
- **Why?** To get the most correct **standard errors** for fixed effects
- Btw, not all conditional distributions will have a single, separately estimated residual variance (e.g., binary → Bernoulli, count → Poisson)
- Agresti calls this part the "**random component**" (≠ random effects!)

3 Parts of Generalized Linear Models



1. Non-Normal Conditional Distribution of $\boldsymbol{y_i}$

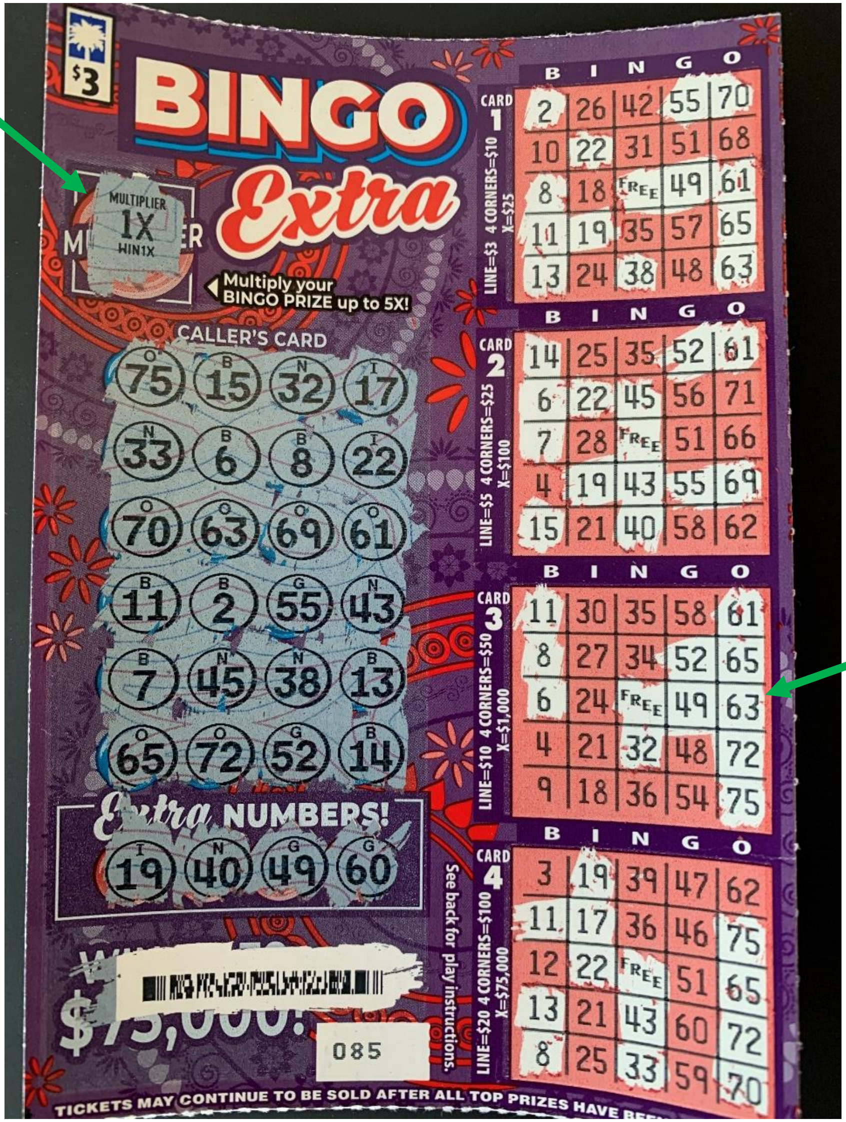
2. Link Function

=

3. Linear Predictor of Fixed (and Random) Effects

2. **Link Function = $g(\cdot)$**: How the conditional mean to be predicted is transformed so that the model predicts an **unbounded** outcome instead
   - **Inverse link** $g^{-1}(\cdot)$= how to go back to conditional mean in data scale
   - Predicted outcomes (found via inverse link) will then stay within bounds
   - e.g., binary outcome: **conditional mean to be predicted is probability of $\boldsymbol{y_i = 1}$,** so the model predicts a linked outcome (when inverse-linked, the predicted probability outcome will stay between 0 and 1)
   - e.g., count outcome: **conditional mean is expected count**, so log of the expected count is predicted so that the expected count stays > 0
   - e.g., normal outcome: an "identity" link function (y_i * 1) is used given that the conditional mean to be predicted is already unbounded…

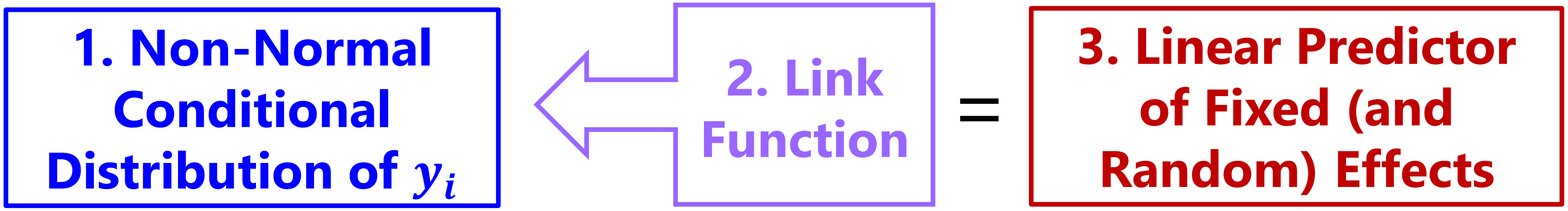
A Real-Life Bummer of an Identity Link


$3
BINGO
Extra
MULTIPLIER
1X
WIN1X
Multiply your BINGO PRIZE up to 5X!
CALLER'S CARD
Extra NUMBERS!
CARD 1 LINE=$3 4 CORNERS=$10 X=$25
CARD 2 LINE=$5 4 CORNERS=$25 X=$100
CARD 3 LINE=$10 4 CORNERS=$50 X=$1,000
CARD 4 LINE=$20 4 CORNERS=$100 X=$75,000
B I N G O
FREE
See back for play instructions.
085
TICKETS MAY CONTINUE TO BE SOLD AFTER ALL TOP PRIZES HAVE BEEN

I won $10!

So what's my bonus multiplier???

3 Parts of Generalized Linear Models



1. Non-Normal Conditional Distribution of y_i

2. Link Function

=

3. Linear Predictor of Fixed (and Random) Effects

3. **Linear Predictor**: How the fixed (and random) effects of predictors combine additively to predict a link-transformed conditional mean

- This works the same as usual, except the linear predictor model **directly predicts the link-transformed conditional mean**, which we then convert (via inverse link) back into the data-scale conditional mean
  - e.g., predict **logit** of probability directly, but inverse-link back to probability
  - e.g., predict **log** of expected count, but inverse-link back to expected count
- That way we can still use the familiar "one-unit change" language to describe effects of model predictors (of the linked conditional mean)
- Btw, fixed effects are no longer determined: they now have to be found through ML iteratively, the same as any variance-related parameters

Normal GLM for Binary Outcomes?

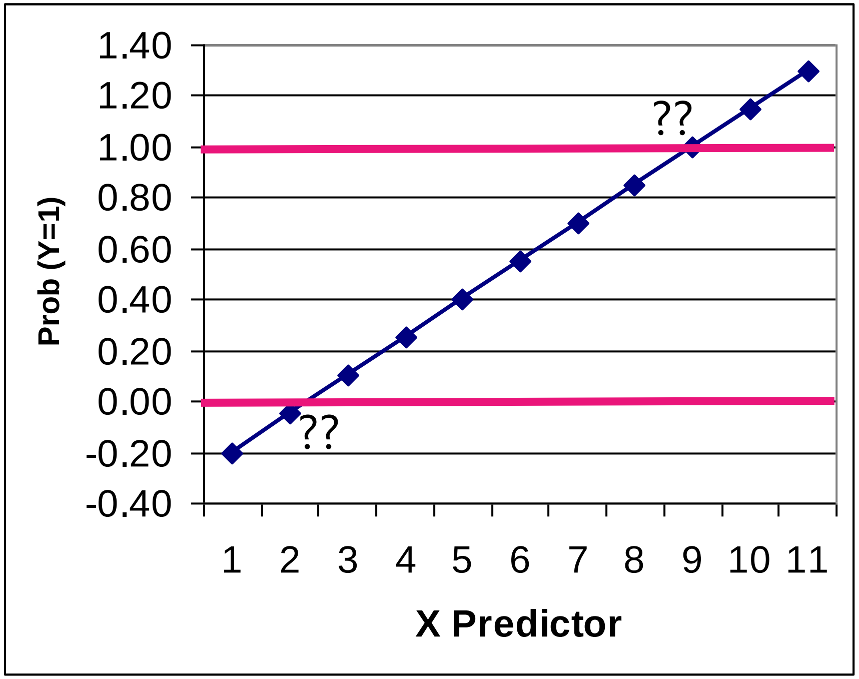
- Let's say we have a single binary (0 or 1) outcome…
- The mean of a binary outcome is the proportion of 1 values
  - So given each person's predictor values, the model tries to predict the **conditional mean**: the **probability of having a 1**: $\boldsymbol{p(y_i = 1)}$
    - The conditional mean has more possible values than the outcome!
  - **What about a GLM???** $\boldsymbol{p(y_i = 1) = \beta_0 + \beta_1(x1_i) + \beta_2(x2_i) + e_i}$
    - $\boldsymbol{\beta_0}$ = expected probability of $\boldsymbol{y_i = 1}$ when all predictors = 0
    - $\boldsymbol{\beta}$'s = expected change in $\boldsymbol{p(y_i = 1)}$ for per unit change in predictor
    - $\boldsymbol{e_i}$ = difference between observed and predicted binary values

  - Model becomes $\boldsymbol{y_i = (\text{predicted probability of } 1) + e_i}$
  - **What could possibly go wrong?**

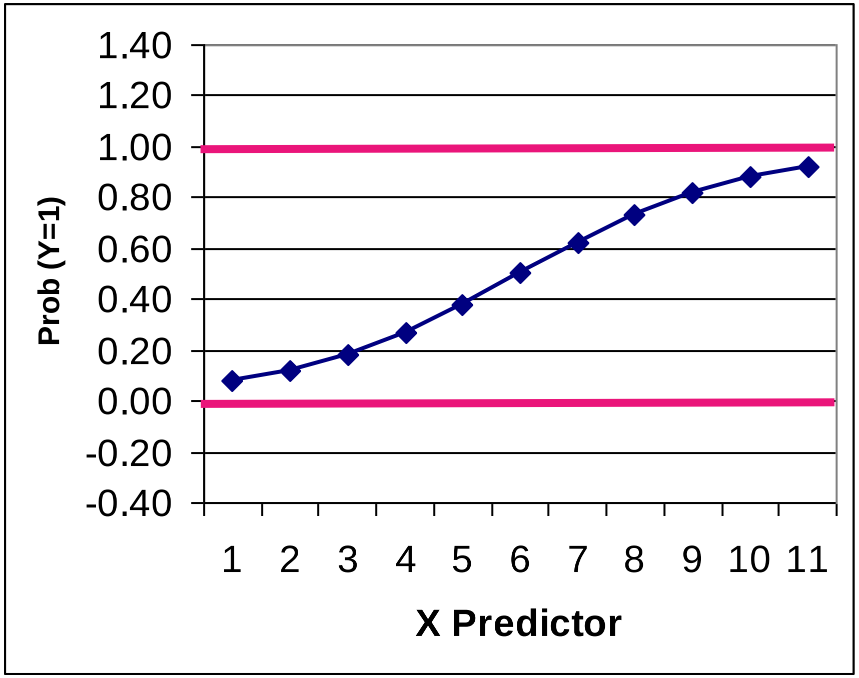
Normal GLM for Binary Outcomes?

- Problem #1: A **linear** relationship between x_i and y_i???
- Probability of a 1 is bounded between 0 and 1, but predicted probabilities from a linear model aren't going to be bounded
- Linear relationship needs to shut off → made nonlinear

We have this…

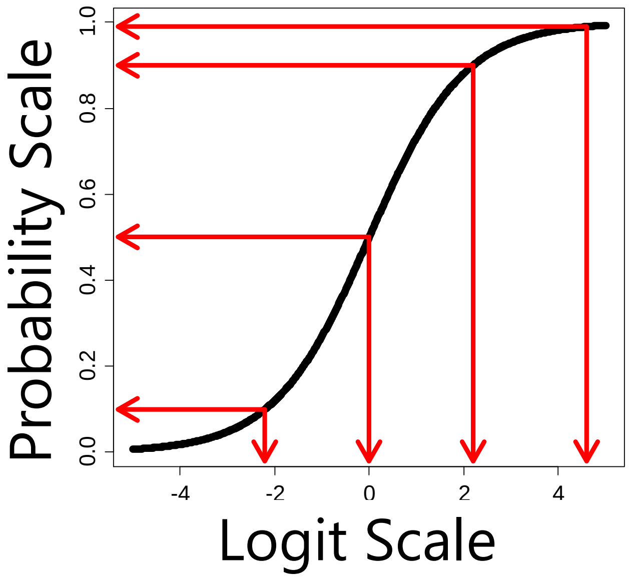


But we need this…



Generalized Models for Binary Outcomes

- Solution to #1: Rather than predicting $\boldsymbol{p(y_i = 1)}$ directly, we must transform it into an unbounded outcome using a **link function**:
  - Step 1: Transform **probability** into **odds**: $\frac{p_i}{1-p_i} = \frac{\text{prob}(y_i=1)}{\text{prob}(y_i=0)}$
    - If $p(y_i = 1) = .7$ then $\text{Odds}(1) = 2.33$; $\text{Odds}(0) = 0.429$
    - But odds scale is skewed, asymmetric, and ranges 0 to +∞ → Not a good outcome!
  - Step 2: **Take *natural log of odds* → "logit" link:** $\mathbf{Log}\left[\frac{\boldsymbol{p_i}}{\mathbf{1}-\boldsymbol{p_i}}\right]$
    - If $p(y_i = 1) = .7$, then $\text{Logit}(1) = 0.846$; $\text{Logit}(0) = -0.846$
    - Logit scale is now symmetric about 0, range is ±∞ → Now a good outcome to predict!



Probability → "data scale"	**Logit → "model scale"**
0.99	4.6
0.90	2.2
0.50	0.0
0.10	−2.2

Can you guess what $p(.01)$ would be on the logit scale?

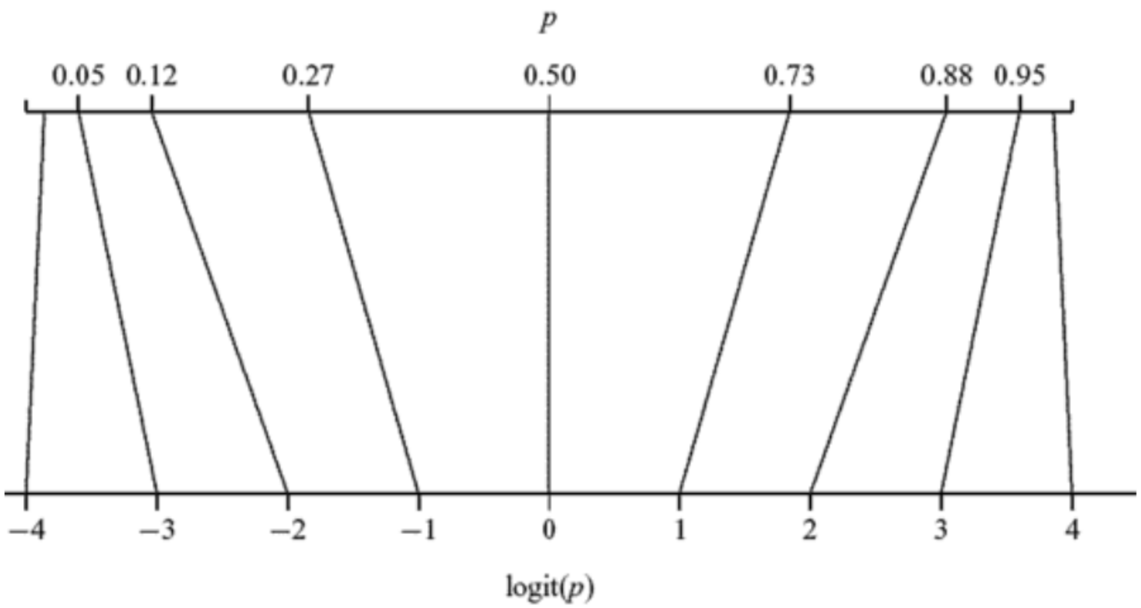
Solution #1: From Probability to Logits

- **A Logit link is a nonlinear transformation of probability:**
  - Equal intervals in logits are NOT equal intervals of probability
  - Logits range from ±∞ and are symmetric around prob = .5 (→ logit = 0)
  - Now we can use a linear model → The model will be **linear with respect to the predicted logit**, which translates into a nonlinear prediction with respect to probability → **the outcome conditional mean shuts off at 0 or 1 as needed**

Probability: $p(y_i = 1)$

Odds: $\left[\frac{p_i}{1-p_i}\right]$

Logit (log odds): $\text{Log}\left[\frac{p_i}{1-p_i}\right]$



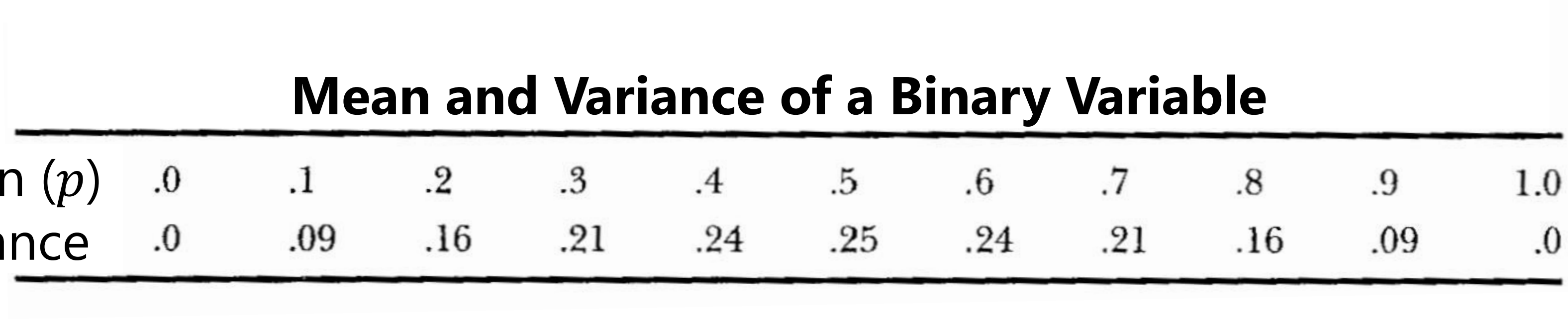
Zero-point on each scale:

Prob = .5
Odds = 1
Logit = 0

Image borrowed from Figure 17.3 of: Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2^{nd} ed.). Sage.

Normal GLM for Binary Outcomes?

- What about a GLM? $\boldsymbol{p(y_i = 1) = \beta_0 + \beta_1(x1_i) + \beta_2(x2_i) + e_i}$
- If $\boldsymbol{y_i}$ is binary, then $\boldsymbol{e_i}$ can only be 2 things: $\boldsymbol{e_i = y_i - \hat{y}_i}$
  - If $\boldsymbol{y_i} = 0$ then $\boldsymbol{e_i}$ = (0 − predicted probability)
  - If $\boldsymbol{y_i} = 1$ then $\boldsymbol{e_i}$= (1 − predicted probability)
- Problem #2a: So the residuals can't be normally distributed
- Problem #2b: The residual variance can't be constant over $\boldsymbol{\hat{y}_i}$ as in GLM because the **mean and variance are dependent**
  - Variance of binary variable: $\boldsymbol{Var(y_i) = p * (1 - p)}$



Mean and Variance of a Binary Variable

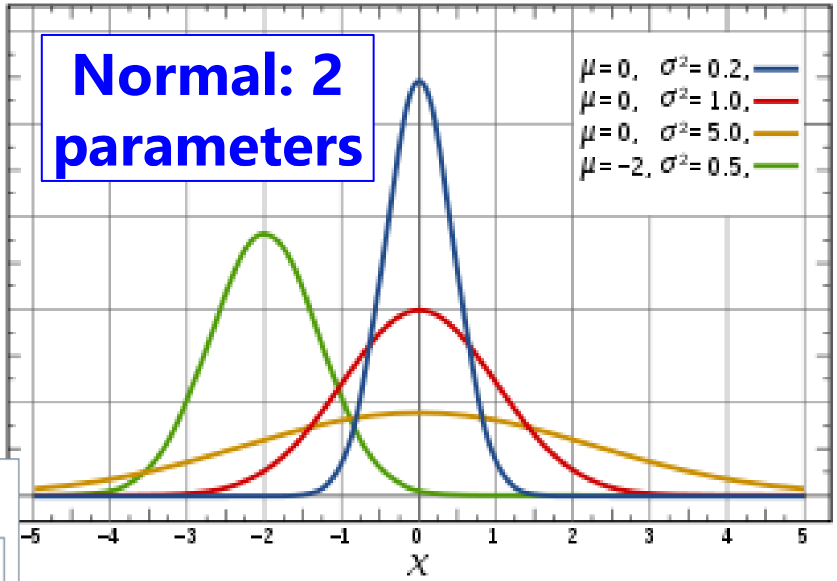
Mean (p)	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
Variance	.0	.09	.16	.21	.24	.25	.24	.21	.16	.09	.0

Solution to #2: Bernoulli Distribution

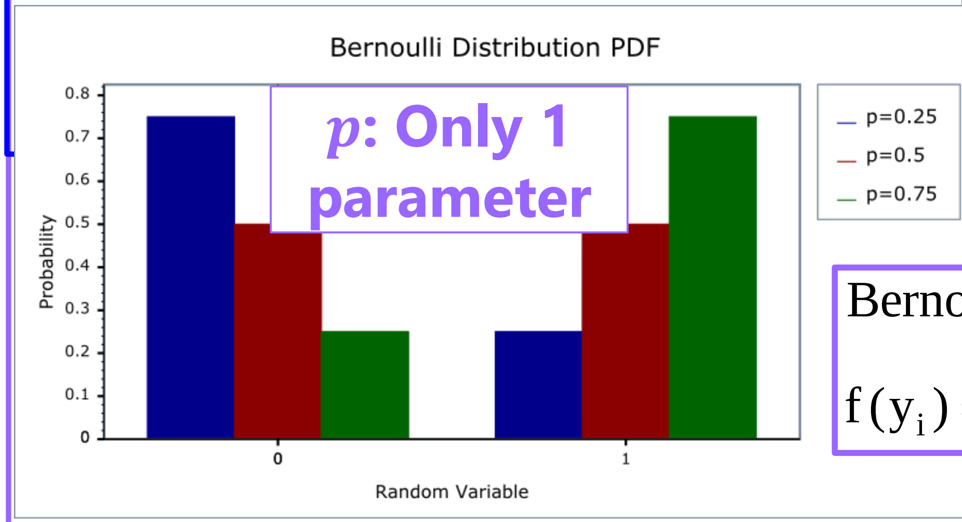
- Rather than using a **normal conditional distribution** for the outcome, we will use a **Bernoulli conditional distribution**

Univariate Normal PDF:

$$f(y_i) = \frac{1}{\sqrt{2\pi\sigma_e^2}} * \exp\left[-\frac{1}{2} * \frac{\left(y_i - y_i\right)^2}{\sigma_e^2} \right]$$



Normal: 2 parameters



***p*: Only 1 parameter**

Bernoulli PDF:

$$f(y_i) = (p_i)^{y_i}(1 - p_i)^{1 - y_i}$$

= *p*(1) if 1, *p*(0) if 0

Top image borrowed from: https://en.wikipedia.org/wiki/Normal_distribution

Bottom image borrowed from: https://www.boost.org/doc/libs/1_70_0/libs/math/doc/html/math_toolkit/dist_ref/dists/bernoulli_dist.html

3 Scales of Predicted Binary Outcomes

- **Logit:** $\mathbf{Log}\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = \beta_0 + \beta_1(x1_i) + \beta_2(x2_i)$ ← $\mathbf{g(\cdot)}$ **link**
  - Predictor slopes are linear and additive just like in GLM, but β = difference in **logit** per one-unit difference in predictor

- **Odds:** $\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = exp(\beta_0 + \beta_1 x1_i + \beta_2 x2_i)$

- **Probability:** $p(y_i = 1) = \frac{exp(\beta_0+\beta_1 x1_i+\beta_2 x2_i)}{1+exp(\beta_0+\beta_1 x1_i+\beta_2 x2_i)}$ ← $\mathbf{g^{-1}(\cdot)}$ **inverse link**

  or equivalently $p(y_i = 1) = \frac{1}{1+exp[-1(\beta_0+\beta_1 x1_i+\beta_2 x2_i)]}$

- This "**logistic regression**" can be estimated using SAS PROC GLIMMIX (LINK=LOGIT, DIST=BINARY) or PROC LOGISTIC; STATA LOGIT/GLM; R GLM family = binomial(link = logit))

Converting Across the 3 Scales

- e.g., for $\mathbf{Log}\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = \hat{y}_i = \beta_0 + \beta_1(x1_i) + \beta_2(x2_i)$

Direction	**Conditional Mean**	**Slope for $x1_i$**	**Slope for $x2_i$**
Predicted logit outcome (i.e., given by "the link"):	$\hat{y}_i$	β_1	β_2
From logits to odds (or odds ratios for effect sizes):	*Odds:* $\exp(\hat{y}_i)$	*Odds ratio:* $\exp(\beta_1)$	*Odds ratio:* $\exp(\beta_2)$
From logits to probability (i.e., by the "inverse link"):	$\frac{\exp(\hat{y}_i)}{1+\exp(\hat{y}_i)}$	***Doesn't make any sense!***	***Doesn't make any sense!***

- You can unlogit the model-predicted conditional mean all the way back into probability to express predicted outcomes, but **you can only unlogit the slopes back into odds ratios** (not all the way back to probability)

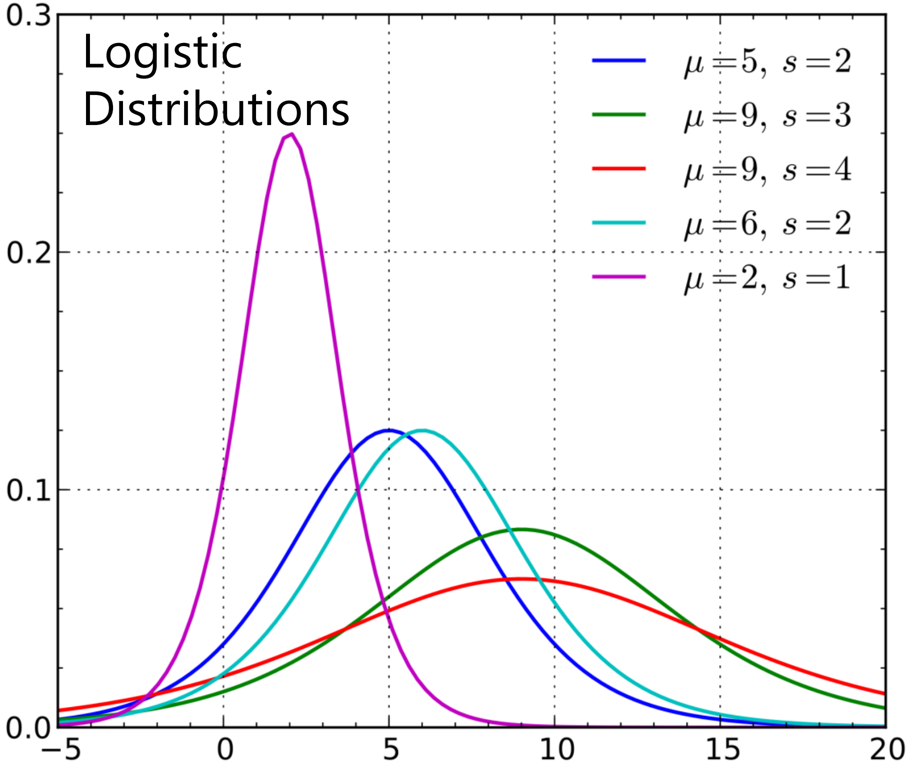
Intercepts (β_0) vs. Thresholds ($-\beta_0$)

- This model is sometimes expressed by calling the logit($\boldsymbol{y_i}$) an underlying continuous ("latent") response of $\boldsymbol{y_i^*}$ instead:

Empty Model for: $\boldsymbol{y_i^*} = \boldsymbol{-threshold} + \boldsymbol{e_i}$

$\boldsymbol{threshold} = \boldsymbol{intercept\ \beta_0 * -1}$

- In which $\boldsymbol{y_i} = \mathbf{1}$ if $(y_i^* > threshold)$, or $\boldsymbol{y_i} = \mathbf{0}$ if $(y_i^* \leq threshold)$



So **when predicting** $\boldsymbol{y_i^*}$, then $e_i \sim \text{Logistic}\left(0, \boldsymbol{\sigma_e^2} = \mathbf{3.29}\right)$

From the **Logistic** Distribution:

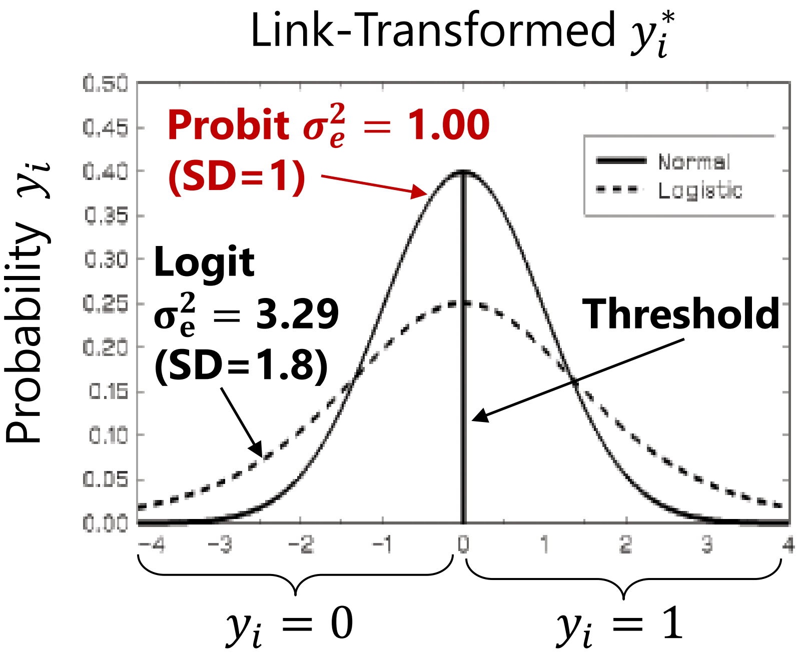
Mean = μ, Variance = $\frac{\pi^2}{3}s^2$, where *s* = scale factor that allows for "over-dispersion" (must be fixed to 1 in binary outcomes for identification)

Image borrowed from: https://en.wikipedia.org/wiki/Logistic_distribution

Other Link Functions for Binary Data

- The idea that a "latent" continuous variable underlies an observed binary response also appears in a "**Probit Regression**" model:
  - A ***probit*** link, such that the linear model predicts a different transformed $\boldsymbol{y_i}$:
    $\text{Probit}(y_i = 1) = \Phi^{-1}[p(y_i = 1)] = \textit{linear predictor} \longleftarrow \boxed{\mathbf{g(\cdot)\ link}}$
    - Φ = standard normal cumulative distribution function, so the link-transformed $\boldsymbol{y_i}$ **is the z-value** that corresponds to the location on standard normal curve **below** which the conditional mean probability is found (i.e., z-value for area to the left)
    - Requires integration to inverse link from probits to predicted probabilities
  - Same Bernoulli distribution for the conditional binary outcomes, in which residual variance cannot be separately estimated (so no e_i in the model)
    - Model scale: Probit can also predict "latent" response: $\boldsymbol{y_i^*} = \boldsymbol{-threshold} + \boldsymbol{e_i}$
    - But Probit says $\boldsymbol{e_i} \sim \boldsymbol{Normal}(\mathbf{0}, \boldsymbol{\sigma_e^2} = \mathbf{1.00})$, whereas logit $\boldsymbol{\sigma_e^2} = \frac{\boldsymbol{\pi^2}}{\mathbf{3}} = \boldsymbol{3.29}$
  - So given this difference in variance, probit coefficients are on a different scale than logit coefficients, and so their estimates won't match… however…

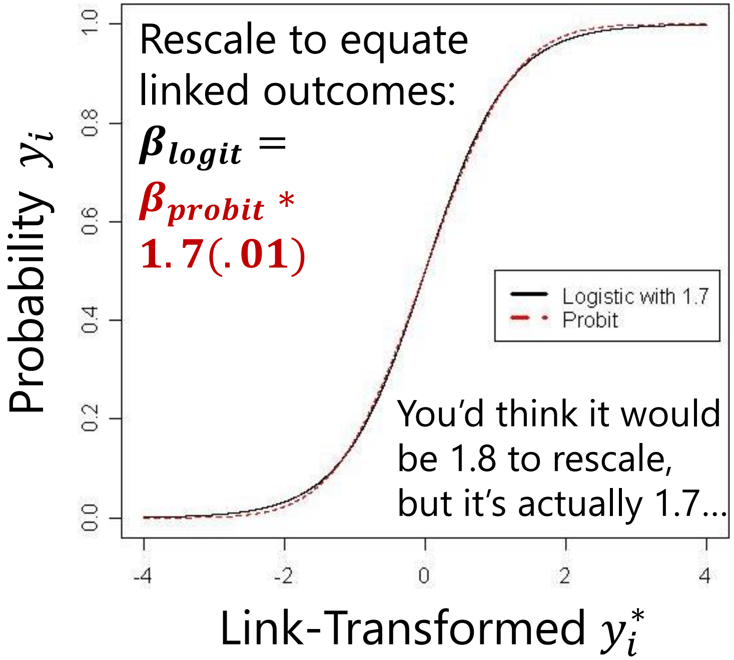
Probit vs. Logit: Should you care? Pry not.



Probit $\sigma_e^2 = 1.00$ (SD=1)

Logit $\sigma_e^2 = 3.29$ (SD=1.8)

Threshold



Rescale to equate linked outcomes: $\boldsymbol{\beta_{logit}} = \boldsymbol{\beta_{probit}} * \mathbf{1.7(.01)}$

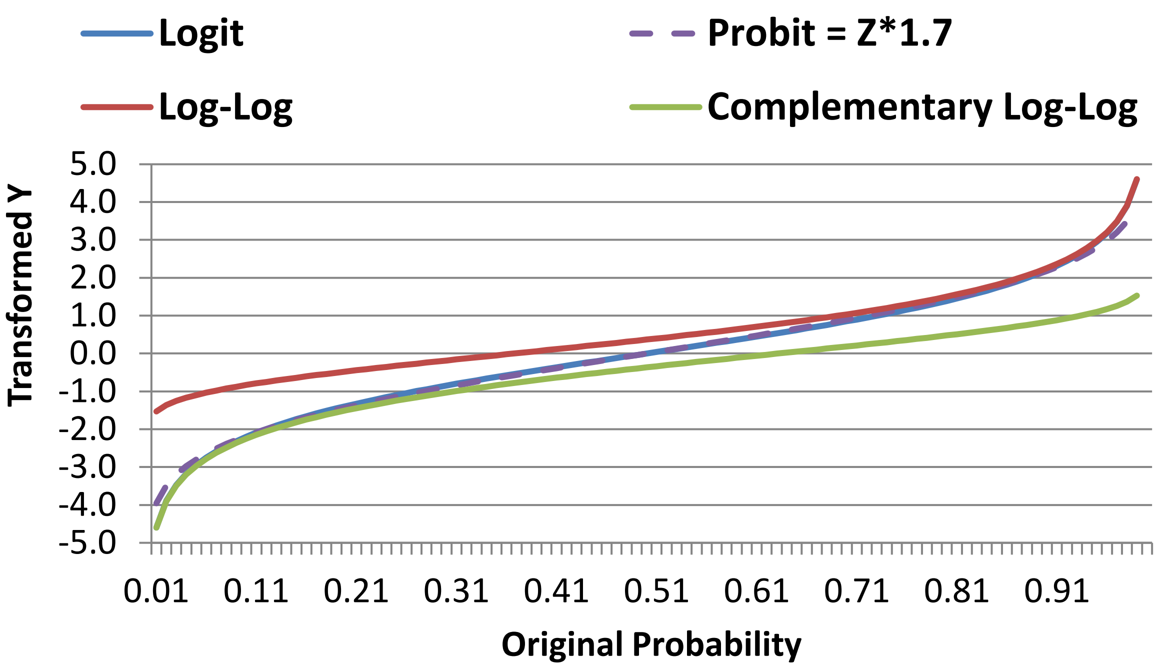
You'd think it would be 1.8 to rescale, but it's actually 1.7…

- Other fun facts about probit:
  - **Probit** = "**ogive**" in the Item Response Theory (IRT) world
  - Probit has no odds ratios (because it's not made from odds)

- Both logit and probit assume **symmetry** of the probability curve, but there are other *asymmetric* options as well…

Left image: exact source now unknown, but I think it was from Don Hedeker
Right image: borrowed from Jonathan Templin

Other Link Functions for Binary Outcomes



Logit = Probit*1.7 which both assume symmetry of prediction

Log-Log is for outcomes in which 1 is more frequent

Complementary Log-Log is for outcomes in which 0 is more frequent

Model→ $\hat{y}_i$	**Logit**	**Probit**	**Log-Log**	**Complement. Log-Log**
g(·) link	$\text{Log}\left(\frac{p_i}{1-p_i}\right)=\hat{y}_i$	$\Phi^{-1}(p_i)=\hat{y}_i$	$-\text{Log}[-\text{Log}(p_i)]=\hat{y}_i$	$\text{Log}[-\text{Log}(1-p_i)]=\hat{y}_i$
$g^{-1}(\cdot)$ inverse link (go back to probability):	$p_i = \frac{exp(\hat{y}_i)}{1+exp(\hat{y}_i)}$	$p_i = \Phi^{-1}(\hat{y}_i)$	$p_i = \exp[-\exp(-\hat{y}_i)]$ $e_i \sim \log - \text{Weibull extreme value}\left(0.577, \sigma_e^2 = \frac{\pi^2}{6}\right)$	$p_i = 1 - \exp[-\exp(\hat{y}_i)]$

Significance Testing for Binary Outcomes

- **Wald tests can be used to test fixed slopes**, but many programs do not use denominator DF
  - If so, p-values may be too optimistic in small samples

Uses Denominator Degrees of Freedom?	Test 1 Slope*	Test >1 Slope*
No: implies infinite N	z	$\chi^2(= z^2)$
Yes: adjusts based on N	t	$F(= t^2)$

$\boxed{F * \# slopes = \chi^2}$

- For models estimated using ML, the **model log-likelihood (LL)** can also be used to assess relative fit (i.e., through model comparisons)
  - $\boldsymbol{LL}$ **= sum across individual LL values** that results from the optimum values of the model parameters (that make the outcomes the tallest)
  - Two flavors: **Maximum Likelihood** (**ML**) or Restricted ML (REML)
    - REML is only possible for conditionally normal outcomes, in which it works better for smaller samples (is same as ordinary least squares)
  - Two directions: $\boldsymbol{LL}$ **(bigger is better)** or $\boldsymbol{-2LL}$ **(smaller is better)**

Likelihood Ratio Tests (LRTs)

- **Nested models can be compared using an LRT: (−2ΔLL Test)**

1. Calculate −2ΔLL: $(-2\text{LL}_{\text{fewer}}) - (-2\text{LL}_{\text{more}})$ OR $-2*(\text{LL}_{\text{fewer}} - \text{LL}_{\text{more}})$
2. Calculate Δdf: $(\#\ \text{Parms}_{\text{more}}) - (\#\ \text{Parms}_{\text{fewer}})$
3. Compare −2ΔLL to χ^2 distribution with df = Δdf
   CHIDIST in excel gives exact p-values for the difference test; so will STATA LRTEST and various functions in R

1. & 2. must be positive values!

- **Add** parameters? Model fit can be **BETTER** (signif) or **NOT BETTER**
- **Remove** parameters? Model fit can be **WORSE** (signif) or **NOT WORSE**

- Non-nested models can be compared by **Information Criteria (IC)** that also reflect model parsimony
  - No p-values or critical values, just "smaller is better"
  - **AIC** = Akaike IC = −2LL + 2 *(#parameters)
  - **BIC** = Bayesian IC = −2LL + log(N)*(#parameters)
  - AIC and BIC can also be used to compare the fit of different link functions for the same distribution

Effect Sizes for Binary Outcomes

- **Odds Ratio (OR)** → effect size for predictors of binary outcomes
- e.g., if $x1_1$ is binary and $x2_i$ is quantitative

$$\mathbf{Log}\left[\frac{\boldsymbol{p(y_i=1)}}{\boldsymbol{1-p(y_i=1)}}\right] = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}(\boldsymbol{x1_i}) + \boldsymbol{\beta_2}(\boldsymbol{x2_i})$$

  - **OR** for unique effect of $x1_i = \exp(\beta_1) = \dfrac{p(y_i = 1|x1_i = 1)/p(y_i = 0|x1_i = 1)}{p(y_i = 1|x1_i = 0)/p(y_i = 0|x1_i = 0)}$
  - **OR** for unique effect of $x2_i = \exp(\beta_2)$: same principle, but denominator is some reference value (e.g., mean by default) and numerator is "one unit" higher (and "unit" can be defined)
  - For each, you'll have to decide at what value to hold other predictors to get the exact probabilities, but the odds ratio will only change if the predictors are part of an interaction (from marginal → conditional)

- **OR is asymmetric**: ranges from 0 to +∞; where 1 = no relationship
  - e.g., if $\boldsymbol{\beta_1} = \mathbf{1}$, then $\exp(\beta_1) = 2.72$ → odds of $y_i = 1$ are 2.72 **times** higher per unit greater $x1_i$
  - e.g., if $\boldsymbol{\beta_1} = -\mathbf{1}$, then $\exp(\beta_1) = 0.37$→ odds of $y_i = 1$ are 0.37 **times** higher per unit greater $x1_i$
  - *Can be more intuitive to phrase results as positive!*

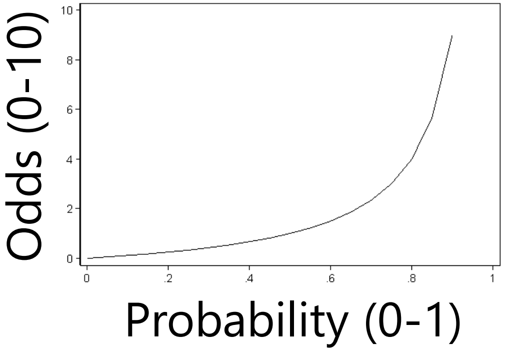


Image borrowed from: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/

Converting Across the 3 Scales

- e.g., for $\mathbf{Log}\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = \hat{y}_i = \beta_0 + \beta_1(x1_i) + \beta_2(x2_i)$

Direction	**Conditional Mean**	**Slope for $x1_i$**	**Slope for $x2_i$**
Predicted logit outcome (i.e., given by "the link"):	$\hat{y}_i$	β_1	β_2
From logits to odds (or odds ratios for effect sizes):	*Odds:* $\exp(\hat{y}_i)$	*Odds ratio:* $\exp(\beta_1)$	*Odds ratio:* $\exp(\beta_2)$
From logits to probability (i.e., by the "inverse link"):	$\frac{\exp(\hat{y}_i)}{1+\exp(\hat{y}_i)}$	***Doesn't make any sense!***	***Doesn't make any sense!***

- You can unlogit the model-predicted conditional mean all the way back into probability to express predicted outcomes, but **you can only unlogit the slopes back into odds ratios** (not all the way back to probability)

R^2 for binary outcomes? Not really.

- **General Linear Models** use a conditional normal distribution for y_i (i.e., the e_i residuals are normal) in which a residual variance is estimated separately from the fixed effects for the conditional mean
  - Allows direct calculation of R^2 for variance explained and change in R^2 between nested models (and F-tests thereof)

- In contrast, **Genera*lized* Linear Models** for binary outcomes use a conditional Bernoulli distribution for y_i in which there is no single (and constant) separately estimated residual variance
  - Instead, residual variance is determined by AND varies with the conditional mean, so an exact R^2 is not possible in the same way
  - There are lots of attempts at "**pseudo-R^2**" variants that **disagree wildly in practice**, see here: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/

Pseudo-R^2 through Expected Variances

- This approach (credited to McKelvey & Zavoina, 1975) applies to many kinds of generalized linear (and mixed effects) models:
  - M&Z pseudo-R^2 on **logit** scale = $\frac{Var(\hat{y}_i^*)}{Var(\hat{y}_i^*)+Var(e_i)} = \frac{Var(\hat{y}_i^*)}{Var(\hat{y}_i^*)+3.29}$
  - M&Z pseudo-R^2 on **probit** scale =$\frac{Var(\hat{y}_i^*)}{Var(\hat{y}_i^*)+Var(e_i)} = \frac{Var(\hat{y}_i^*)}{Var(\hat{y}_i^*)+1.00}$
  - $Var(\hat{y}_i^*)$ = variance of the predicted logit outcomes
    - Save model-scale predicted outcomes, then calculate their variance
  - $Var(e_i)$ = model-scale conditional variance (of "residuals")
    - Substitute known value based on underlying distribution of y_i^*
    - Keep in mind this uses model scale, not data scale (not probabilities), and so these R^2 values are not really comparable to OLS variants
  - Btw, this expected variance approach generalizes to calculation of intraclass correlation (ICC) when random effects are also included…

Too Logit to Quit* https://www.youtube.com/watch?v=HFCv86OIk8E

- The **logit** is the basis for many other generalized models for categorical (ordinal or nominal; IRT "polytomous") outcomes

- Next we'll see how C possible response categories can be predicted using $C - 1$ binary "submodels" whose link functions carve up the categories in different ways, in which each binary submodel (usually) uses a logit link to predict its outcome

- Types of categorical outcomes:
  - Definitely ordered categories: "**cumulative logit**" → ordinal
  - Maybe ordered categories: "**adjacent category logit**" (not used much)
  - Definitely NOT ordered categories: "**generalized logit**" → nominal (or "baseline category logit" or "multinomial regression"

* Starts about 8 minutes into 15-minute video (and MY joke for the last 10+ years!)

Logit Models for C Ordinal Categories

- Known as “**cumulative logit**” or “**proportional odds**” model in generalized models; known as “**graded response model**” in IRT
  - SAS GLIMMIX (LINK=CLOGIT DIST=MULT) or PROC LOGISTIC; STATA OLOGIT/GOLOGIT2/GLM; R VGLM family=cumulative(parallel=TRUE)
- Models the probability of **lower vs. higher** cumulative categories via $C - 1$ submodels (e.g., if $C = 4$ possible responses of $c = 0,1,2,3$):

$$\underbrace{\textbf{0}\text{ vs. }\textbf{1,2,3}}_{\text{Submodel}_1} \qquad \underbrace{\textbf{0,1}\text{ vs. }\textbf{2,3}}_{\text{Submodel}_2} \qquad \underbrace{\textbf{0,1,2}\text{ vs. }\textbf{3}}_{\text{Submodel}_3}$$

I’ve named these submodels based on what they predict, but each program output will name them their own way…

- In software what the binary submodels predict depends on whether the model is predicting **DOWN ($y_i = 0$)** or **UP ($y_i = 1$)** **cumulatively**
- **Example predicting UP in an empty model (subscripts=parm, submodel)**
- Submodel 1: $Logit[p(y_i > 0)] = \beta_{01}$ → $p(y_i > 0) = exp(\beta_{01})/[1 + exp(\beta_{01})]$
- Submodel 2: $Logit[p(y_i > 1)] = \beta_{02}$ → $p(y_i > 1) = exp(\beta_{02})/[1 + exp(\beta_{02})]$
- Submodel 3: $Logit[p(y_i > 2)] = \beta_{03}$ → $p(y_i > 2) = exp(\beta_{03})/[1 + exp(\beta_{03})]$

Logit Models for C Ordinal Categories

- Models the probability of **lower vs. higher** cumulative categories via $C - 1$ submodels (e.g., if $C = 4$ possible responses of $c = 0,1,2,3$):

$$\underbrace{\text{0 vs. 1,2,3}}_{\substack{\text{Submodel}_1 \\ \rightarrow \text{Prob}_1}} \qquad \underbrace{\text{0,1 vs. 2,3}}_{\substack{\text{Submodel}_2 \\ \rightarrow \text{Prob}_2}} \qquad \underbrace{\text{0,1,2 vs. 3}}_{\substack{\text{Submodel}_3 \\ \rightarrow \text{Prob}_3}}$$

$$\mathrm{Logit}[p(y_i > 2)] = \beta_{03}$$

$$\rightarrow p(y_i > 2) = \frac{exp(\beta_{03})}{1+exp(\beta_{03})}$$

- In software, what the binary submodels predict depends on whether the model is predicting **DOWN $(y_i = 0)$** or **UP $(y_i = 1)$** **cumulatively**
  - **Start with an empty model to verify which way your program is predicting!**
  - Either way, the model predicts the middle category responses ***indirectly***

- **Example if predicting UP with an empty model:**
  - Probability of 0 = 1 – Prob_1
    Probability of 1 = Prob_1 – Prob_2
    Probability of 2 = Prob_2 – Prob_3
    Probability of 3 = Prob_3 – 0

The cumulative submodels that create these probabilities are each estimated using **all the data** (good, especially for categories not chosen often), but **assume order in doing so** (may be bad or ok, depending on your response format)

Logit Models for C Ordinal Categories

- Btw, ordinal models usually use a logit link transformation, but they can also use cumulative log-log or cumulative complementary log-log links

- Assume **proportional odds: that SLOPES of predictors ARE THE SAME across binary submodels**—for example (subscripts = parm, submodel)
  - Submodel 1: $Logit[p(y_i > 0)] = \boldsymbol{\beta_{01}} + \beta_1 x1_i + \beta_2 x2_i + \beta_3 x1_i x2_i$
  - Submodel 2: $Logit[p(y_i > 1)] = \boldsymbol{\beta_{02}} + \beta_1 x1_i + \beta_2 x2_i + \beta_3 x1_i x2_i$
  - Submodel 3: $Logit[p(y_i > 2)] = \boldsymbol{\beta_{03}} + \beta_1 x1_i + \beta_2 x2_i + \beta_3 x1_i x2_i$

- Proportional odds essentially means no interaction between submodel and predictor slope, which greatly reduces the number of estimated parameters
  - Can be tested and changed to "partial" proportional odds in SAS LOGISTIC, STATA GOLOGIT2, and R VGLM (but harder to find in mixed-effects models)
  - If the proportional odds assumption fails, you can use a nominal model instead (dummy-coding to create separate outcomes can approximate a nominal model)

Logit-Based Models for C Categories

- Uses **multinomial distribution**: e.g., PDF for $C = 4$ categories of $c = 0,1,2,3$; an observed $y_i = c$; and indicators I if $c = y_i$

$$f(y_i = c) = p_{i0}^{I[y_i=0]} p_{i1}^{I[y_i=1]} p_{i2}^{I[y_i=2]} p_{i3}^{I[y_i=3]}$$

Only p_{ic} for response $y_i = c$ gets used

  - Maximum likelihood estimation finds the most likely parameters for the model to predict the probability of each response through the (usually logit) link function; probabilities sum to 1: $\sum_{c=1}^{C} p_{ic} = 1$

- Other models for categorical data that use a multinomial PDF:
  - Adjacent category logit (IRT "partial credit"): Models probability of **each next highest** category via $C - 1$ submodels (e.g., if $C = 4$):

    0 vs. **1** **1** vs. **2** **2** vs. **3**

  - Baseline category logit (nominal or "multinomial"): Models probability of **reference vs. each other** $\boldsymbol{c}$ via $C - 1$ submodels (e.g., if $C = 4$ and $0 = \text{ref}$):

    0 vs. **1** **0** vs. **2** **0** vs. **3**

    ALL parameters are estimated **separately** per nominal submodel

    - Nominal also assumes "independence of irrelevant alternatives"—that the same fixed effects would be found if the possible choices were not the same (empirically testable)

Effect Size for Categorical Outcomes

- Because models for categorical outcomes are built using submodels for binary outcomes, **odds ratios** (OR) can still be used as an effect sizes for individual slopes in submodels for categorical outcomes

- Pseudo-R^2 for categorical outcomes will be trickier to compute…
  - To use M&Z pseudo-R^2, you'd need to represent the sources of variance for each binary submodel, which translates readily into nominal models, but not so much into cumulative or adjacent-category models
  - When it doubt (and you must provide some type of R^2 value), find a way to **correlate actual outcomes with** a similarly-ranged **model-predicted outcome** that still maintains error; here, do this for each person:
    - Binary: draw a random 0/1 value from a Bernoulli distribution with a mean given by their predicted probability of a 1
    - Categorical: calculate predicted probability of each of C categories, then draw from a random multinomial distribution with those probabilities
    - Type of correlation will be dictated by outcome type (e.g., tetrachoric for binary or nominal submodels, polychoric or Spearman for ordinal response)

Wrapping Up: Significant Differences

	General(ized) Models for Conditionally Normal Outcomes	**Generalized Models for Categorical Outcomes**
What is predicted?	y_i directly (via "identity link function")	Link-transformed probability of "1" or "0" (via logit, probit, etc.)
What estimator and conditional distribution (i.e., for y_i after predictors) are typically used?	REML (is equal to OLS) and normal	ML and multinomial (with Bernoulli as special case with $C = 2$)
How are global and specific effect sizes assessed?	Global: True R^2 Specific: d, r, semi-partial eta^2, or standardized slopes	Global: Pseudo-R^2 Specific: usually odds ratios (or less common, convert t into d or r)
Can fixed effect estimates be compared directly between models?	Yes	No, because they change scale with total variance… see Winship & Mare (1983, 1984)

Relating Regression, Item Factor Analysis (IFA), and Item Response Theory (IRT)

- CFA → linear regression as IRT → logistic regression

i = item, s = subject

  - Predictor x_s is observed, but predictor F_s is latent (*aka*, factor, variable, trait)

- **Linear regression model** and **(linear) CFA model** (for continuous responses):

$$y_{is} = \beta_{0i} + \beta_{1i} x_s + \ e_{is} \qquad y_{is} = \mu_i + \lambda_i F_s + e_{is}$$

- **Logistic regression model** (for 0/1 responses, so there is no e_{is} residual):

$$\text{Log}\left[\frac{p(y_{is}=1)}{1-p(y_{is}=1)}\right] = \beta_{0i} + \beta_{1i} x_s$$

- **2-PL IRT model** (for 0/1 responses, so there is no e_{is} residual):

$$\text{Log}\left[\frac{p(y_{is}=1)}{1-p(y_{is}=1)}\right] = a_i(\theta_s - b_i)$$

Why does this IRT model look so different than the CFA model? Here's how these models all relate…

Relating Regression, CFA, IFA, and IRT

- **Linear regression model** and **(Linear) Confirmatory FA model**:

  $y_{is} = \beta_{0i} + \beta_{1i}x_s + e_{is}$ $\quad$ $y_{is} = \mu_i + \lambda_i F_s + e_{is}$

- **Binary regression models** and **Binary Item Factor Analysis models:**

  $\text{Logit}\,[p(y_{is} = 1)] = \beta_{0i} + \beta_{1i}x_s$ $\quad$ $\text{Logit}[p(y_{is} = 1)] = -\tau_i + \lambda_i F_s$

  $\text{Probit}[p(y_{is} = 1)] = \beta_{0i} + \beta_{1i}x_s$ $\quad$ $\text{Probit}[p(y_{is} = 1)] = -\tau_i + \lambda_i F_s$

- **Binary Item Response Theory models:**

  2PL: $\text{Logit}\,[p(y_{is} = 1)] = a_i(\theta_s - b_i)$

  Ogive: $\text{Probit}[p(y_{is} = 1)] = a_i(\theta_s - b_i)$

Logit to Probability:

$$\text{prob} = \frac{exp(\text{logit})}{1 + exp(\text{logit})}$$

- In CFA, item loading $\boldsymbol{\lambda_i}$ = "**discrimination**" and item intercept $\boldsymbol{\mu_i}$ = "**difficulty**", but difficulty was backwards (easier or less severe items had higher means)…
- In IFA for binary items within Mplus, the **intercept** $\boldsymbol{\mu_i}$ (which was really "**easiness**") becomes a "**threshold**" $\boldsymbol{\tau_i}$ that really IS "**difficulty**": $\boldsymbol{\mu_i = -\tau_i}$
  → this provides continuity of direction with the IRT b_i difficulty values
- The IRT and IFA models get re-arranged into each other as follows…

From IFA to IRT

IFA with “easiness” **intercept $\boldsymbol{\mu_i}$**: $\textbf{Logit or Probit } \boldsymbol{y_{is}} = \boldsymbol{\mu_i + \lambda_i F_s}$ $\boldsymbol{\mu_i = -\tau_i}$

IFA with “difficulty” **threshold $\boldsymbol{\tau_i}$**: $\textbf{Logit or Probit } \boldsymbol{y_{is}} = \boldsymbol{-\tau_i + \lambda_i F_s}$

IFA model with “difficulty” thresholds can be written as a **2PL IRT Model:**

IRT model: **IFA model:**

$$\textbf{Logit or Probit } \boldsymbol{y_{is}} = \boldsymbol{a_i(\theta_s - b_i)} = -\underbrace{\boldsymbol{a_i b_i}}_{\boldsymbol{\tau_i}} + \underbrace{\boldsymbol{a_i}}_{\boldsymbol{\lambda_i}}\boldsymbol{\theta_s}$$

$\boldsymbol{a_i}$ = discrimination
$\boldsymbol{b_i}$ = difficulty
$\boldsymbol{\theta_s}$ = $\boldsymbol{F_s}$ latent trait

Convert IFA to IRT:

$$a_i = \lambda_i * \sqrt{\text{Theta Variance}}$$

$$b_i = \frac{\tau_i - (\lambda_i * \text{Theta Mean})}{\lambda_i * \sqrt{\text{Theta Variance}}}$$

Convert IRT to IFA:

$$\lambda_i = \frac{a_i}{\sqrt{\text{Theta Variance}}}$$

$$\tau_i = a_i b_i + \frac{a_i * \text{Theta Mean}}{\sqrt{\text{Theta Variance}}}$$

Note: These formulas rescale a_i and b_i so that theta M=0, VAR=1

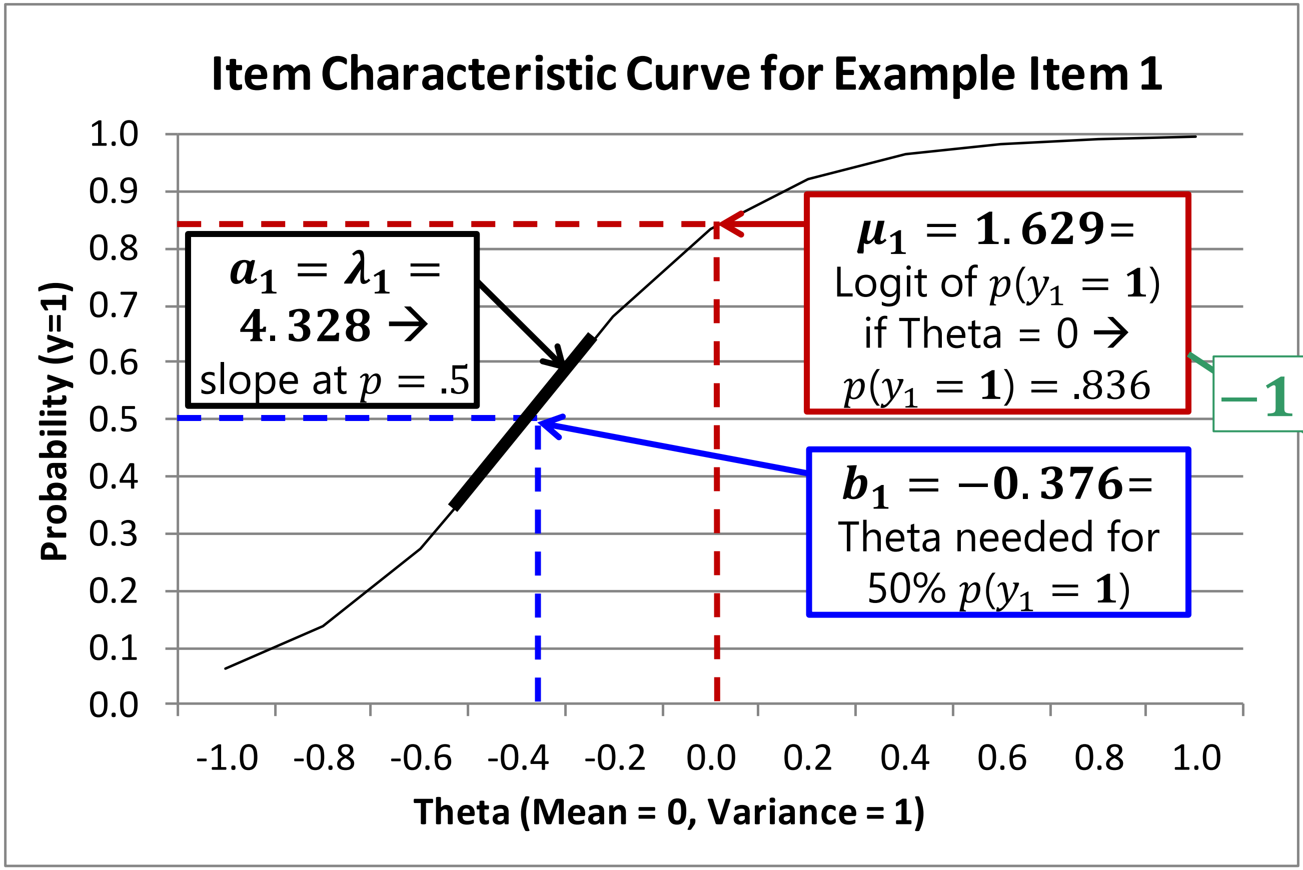
If you don’t want to rescale theta, use M=0 and VAR=1 to keep your current scale

Item Parameter Interpretations

IFA model with **loading** and "easiness" **intercept** $\boldsymbol{\mu_i}$: **Logit** $\boldsymbol{y_{is} = \mu_i + \lambda_i F_s}$

IFA model with **loading** and "difficulty" **threshold** $\boldsymbol{\tau_i}$: **Logit** $\boldsymbol{y_{is} = -\tau_i + \lambda_i F_s}$

2PL IRT model with **discrimination** and **difficulty**: **Logit** $\boldsymbol{y_{is} = a_i(\theta_s - b_i)}$



$\boldsymbol{a_1 = \lambda_1 = 4.328}$ → slope at $p = .5$

$\boldsymbol{\mu_1 = 1.629}$= Logit of $p(y_1 = \mathbf{1})$ if Theta = 0 → $p(y_1 = \mathbf{1}) = .836$

$\boldsymbol{b_1 = -0.376}$= Theta needed for 50% $p(y_1 = \mathbf{1})$

$\mathbf{-1} *$

$\boldsymbol{\tau_1 = -1.629}$= Logit of $p(y_1 = \mathbf{0})$ if Theta = 0 → $p(y_1 = \mathbf{0}) = .164$

From IRT to IFA:
$\lambda_i = a_i$
$\tau_i = a_i b_i$

Item Parameter Interpretation

IFA model with **loading** and "easiness" **intercept** μ_i: $\textbf{Logit } y_{is} = \mu_i + \lambda_i F_s$

IFA model with **loading** and "difficulty" **threshold** τ_i: $\textbf{Logit } y_{is} = -\tau_i + \lambda_i F_s$

2PL IRT model with **discrimination** and **difficulty**: $\textbf{Logit } y_{is} = a_i(\theta_s - b_i)$

- IFA and IRT item slope parameters are interpreted similarly:
  - IFA loading λ_i = Δ in logit/probit of $y_{is} = 1$ for a 1-unit Δ in Theta
  - IRT discrimination a_i = slope of ICC at prob=.50 (logit/probit = 0)

- IFA and IRT item location parameters are interpreted differently:
  - **IFA intercept** μ_i = logit/probit of $y_{is} = 1$ when **Theta = 0**
  - **IFA threshold** τ_i = logit/probit of $y_{is} = 0$ when **Theta = 0**
  - **IRT difficulty** b_i = **amount of Theta needed** for logit/probit of $y_{is} = 1$
    - So b_i difficulty values are more interpretable as measures of **location**