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: ordinal and nominal

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 call 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
- 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!)
- **Why care?** To get the most correct **standard errors** for fixed effects

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**

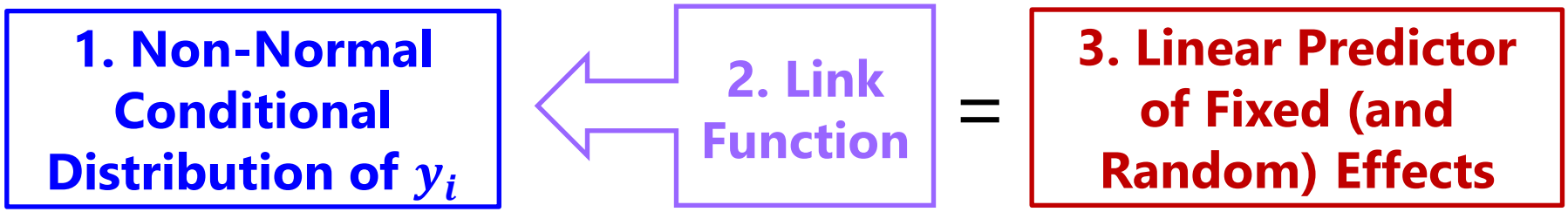
2. **Link Function = $g(\cdot)$**: How the conditional mean to be predicted is transformed so that the model predicts an **unbounded** outcome instead
   - **Inverse link** $g^{-1}(\cdot)$= how to go back to conditional mean in data scale
   - Predicted outcomes (found via inverse link) will then stay within bounds
   - e.g., binary outcome: **conditional mean to be predicted is probability of $y_i = 1$,** 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

I won $10!

So what's my bonus multiplier???

3 Parts of Generalized Linear Models



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

- This is the same as usual, except the linear predictor **directly predicts the link-transformed (model-scale) 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 (on 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

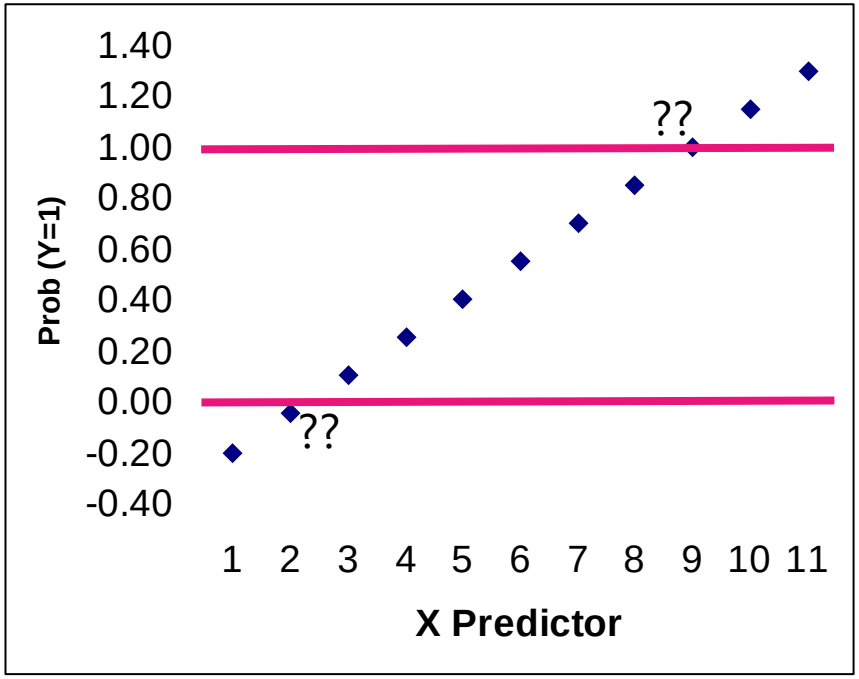
Normal GLM for Binary Outcomes?

- 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???**

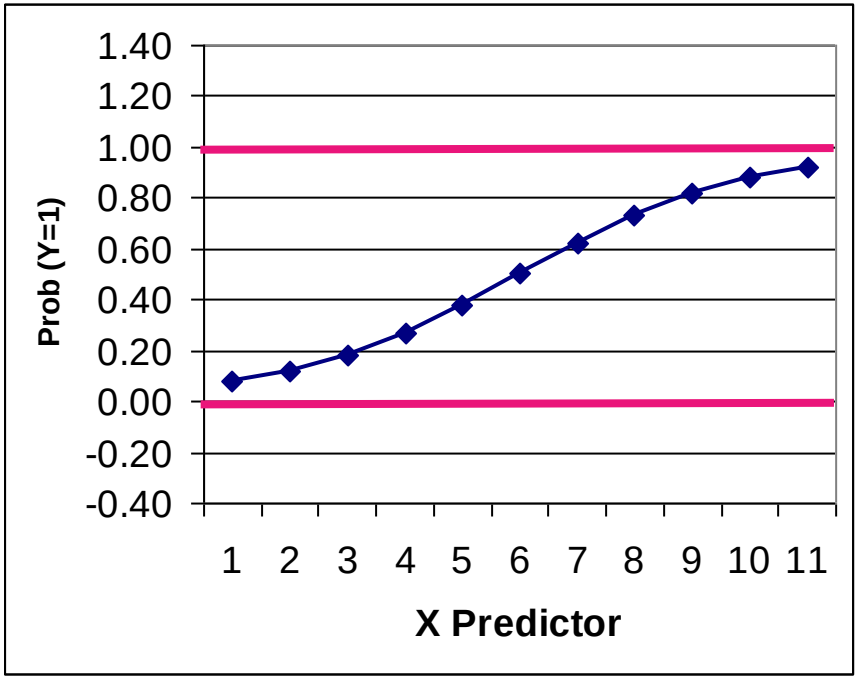
Normal GLM for Binary Outcomes?

- 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…

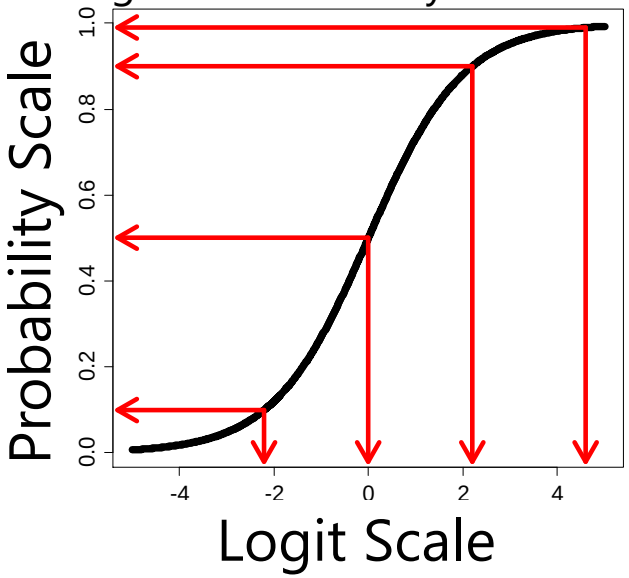


But we need this…



Generalized Models for Binary Outcomes

- Solution to #1: Rather than predicting $\boldsymbol{p(y_i = 1)}$ directly, the model transforms 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:** $\textbf{Log}\left[\frac{p_i}{1-p_i}\right]$
    - 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?

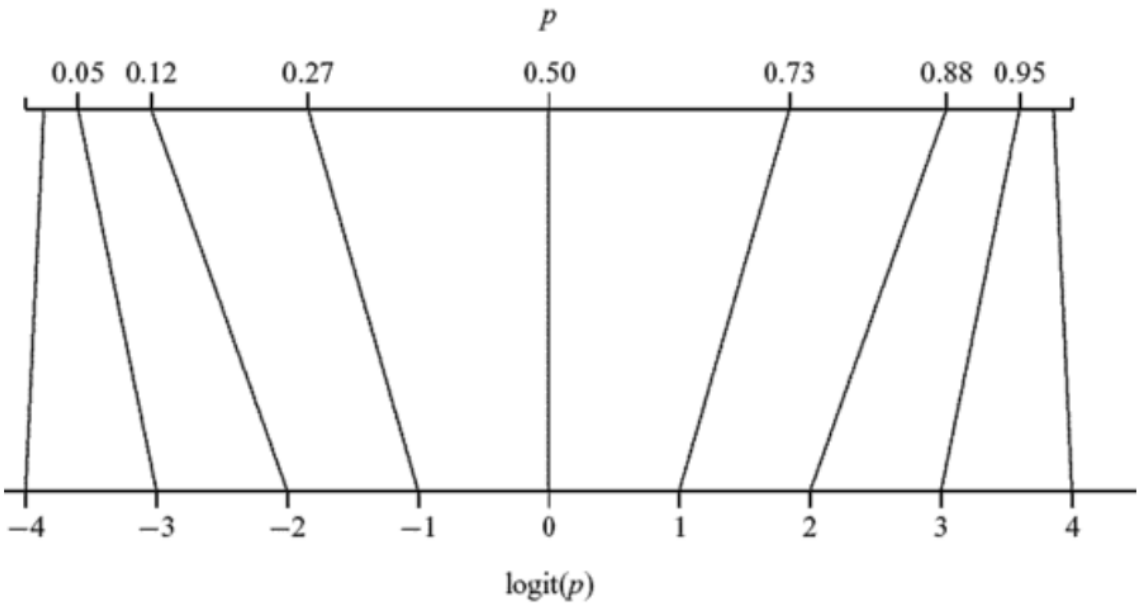
Solution #1: From Probability to Logits

- **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* (2nd 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

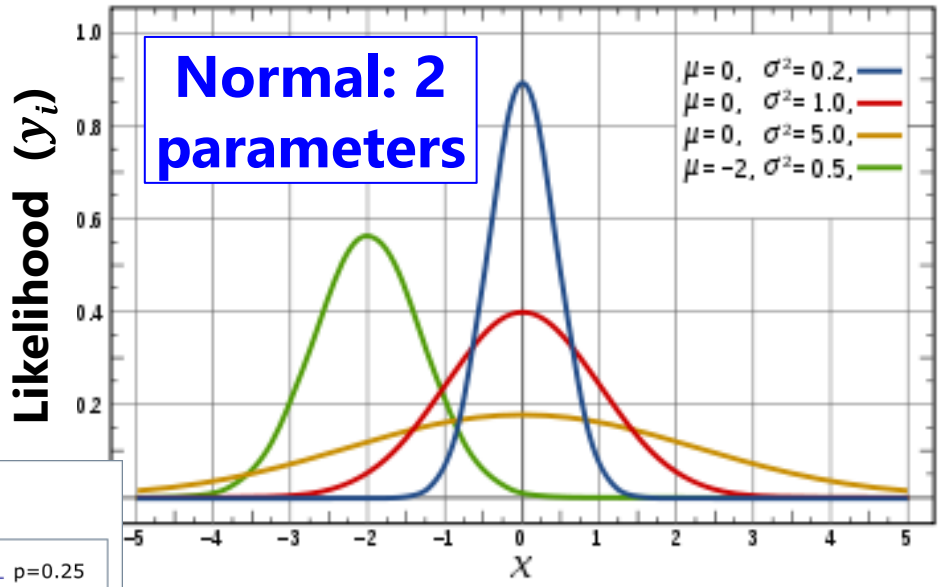
Mean (p)	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
Variance	.0	.09	.16	.21	.24	.25	.24	.21	.16	.09	.0

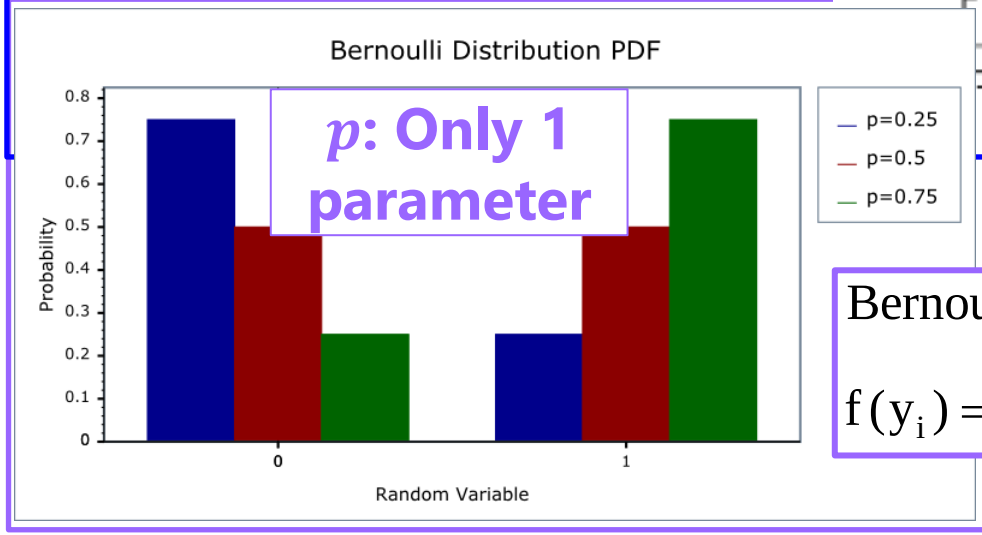
Solution to #2: Bernoulli Distribution

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

Univariate Normal PDF:

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





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

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

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

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

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

Converting Across the 3 Outcome 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 (given 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 changes in probability)
- Order of operations: build predicted logit outcome, then logit → probability

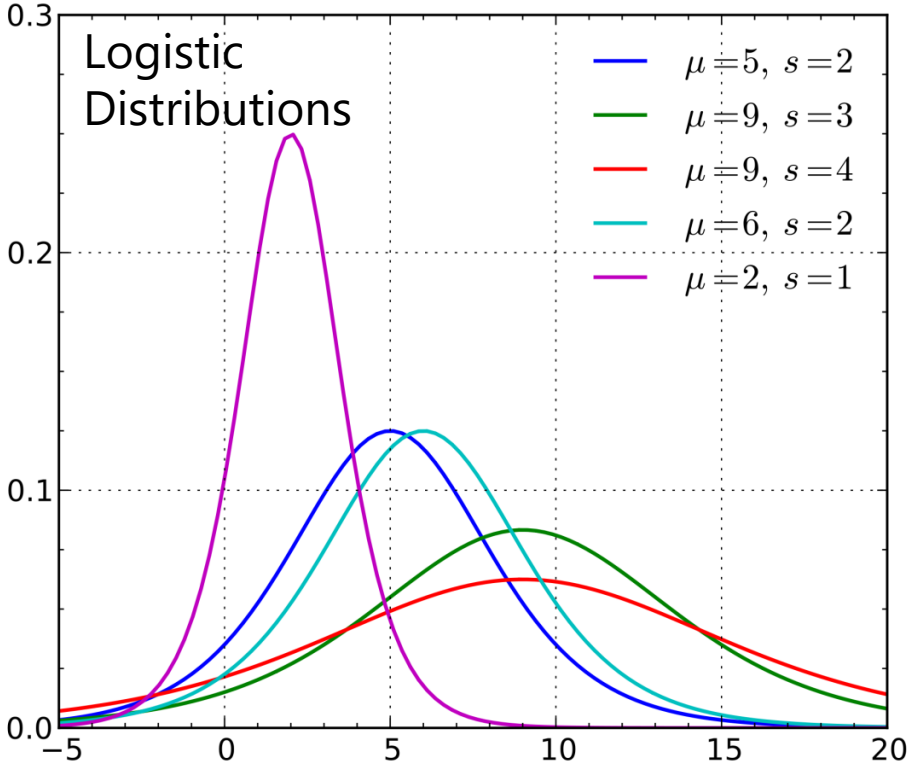
Intercepts (β_0) vs. Thresholds ($-\beta_0$)

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

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

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



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

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

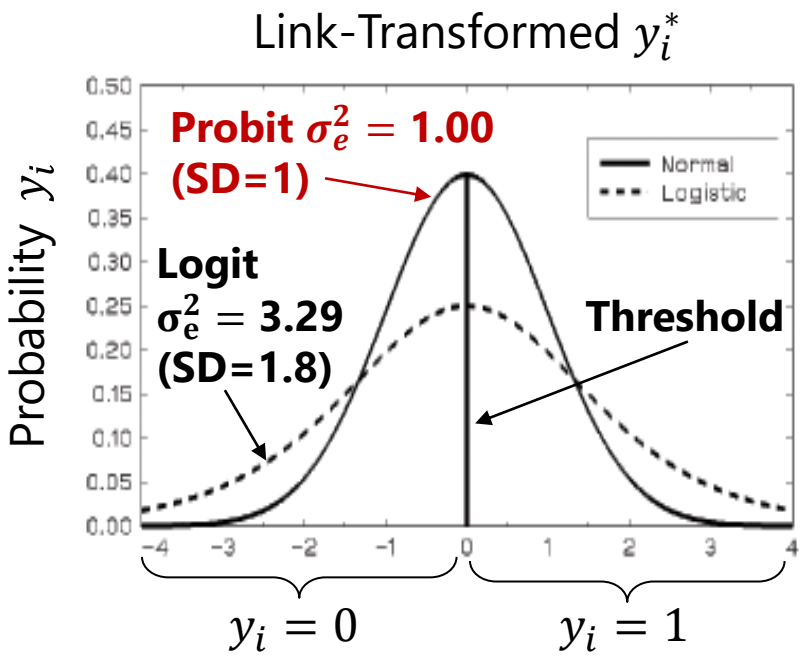
- The idea that a "latent" continuous variable underlies an observed binary response also appears in a "**Probit Regression**" model:

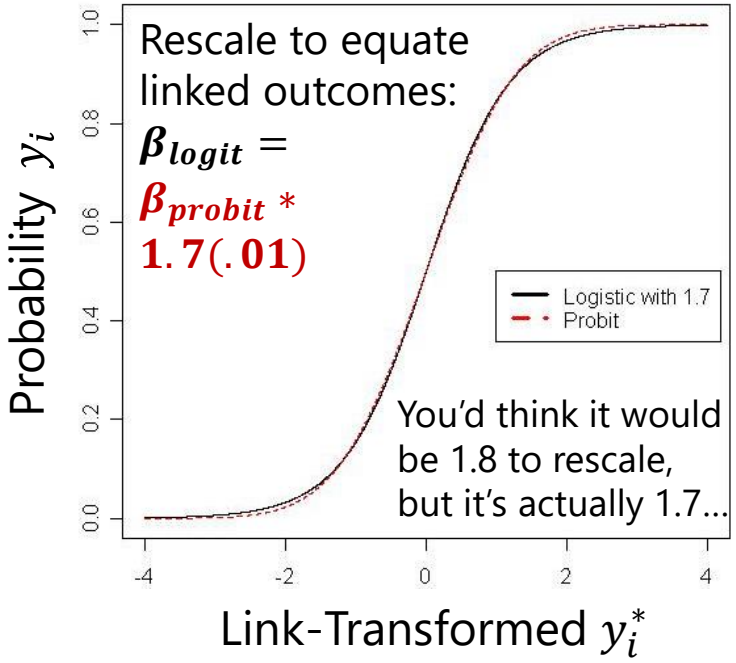
  - A ***probit*** link, such that the linear model predicts a different transformed $\boldsymbol{y_i}$:
    $\text{Probit}(y_i = 1) = \Phi^{-1}[p(y_i = 1)] = linear\ predictor$ ← **g(·) link**
    - Φ = standard normal cumulative distribution function, so the link-transformed $\boldsymbol{y_i}$ **is the 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: $\boldsymbol{y_i^* = -threshold + e_i}$
    - But Probit says $\boldsymbol{e_i \sim Normal(0, \sigma_e^2 = 1.00)}$, whereas logit $\boldsymbol{\sigma_e^2 = \frac{\pi^2}{3} = 3.29}$

  - 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.



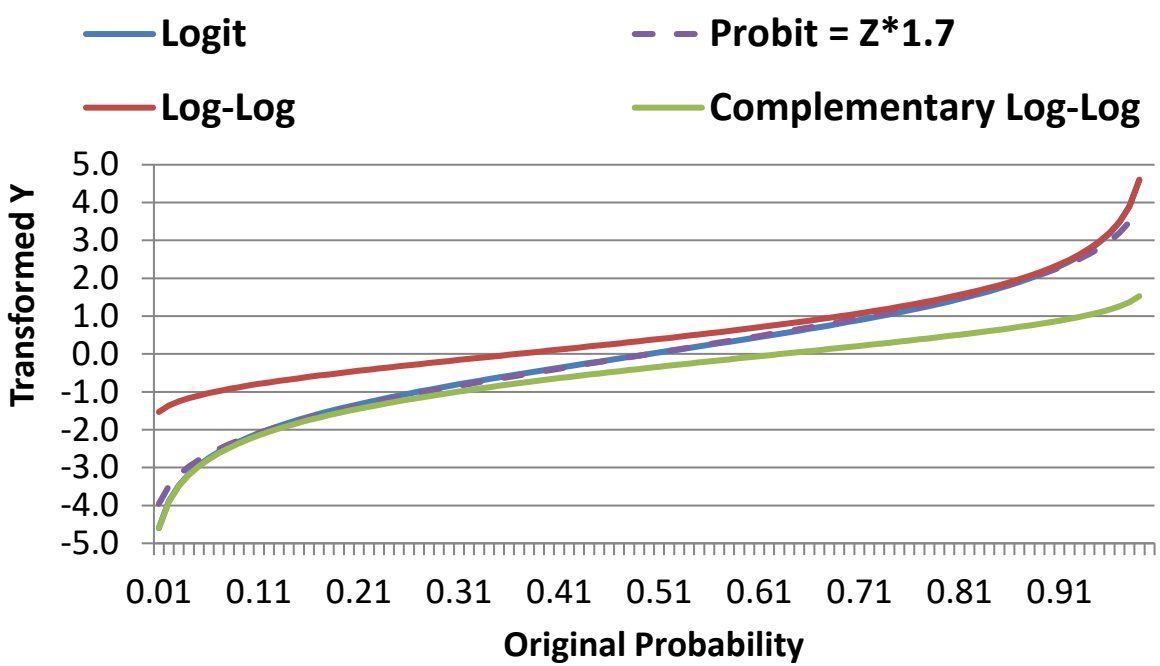


- Other fun facts about probit:
  - **Probit** = "**ogive**" in the Item Response Theory (IRT) world
  - Probit has no odds ratios (because it's not made from odds)

- 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
both of which 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):	$p_i = \frac{exp(\hat{y}_i)}{1+exp(\hat{y}_i)}$	$p_i = \Phi^{-1}(\hat{y}_i)$	$p_i = \exp[-\exp(-\hat{y}_i)]$ $e_i \sim$ logWeibull "extreme value" $\left(0.577, \sigma_e^2 = \frac{\pi^2}{6}\right)$ Also known as "Gumbel"	$p_i = 1-\exp[-\exp(\hat{y}_i)]$

Significance Testing for Binary Outcomes

- **Wald tests *can* be used to test fixed slopes**, but most programs do NOT use denominator DF

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	$F (= t^2)$

  - If so, p-values may be too optimistic in small samples

$$F * \#\ slopes = \chi^2$$

  - Btw, R results for Wald test χ^2 can differ from SAS/STATA because of how fixed effect standard errors are found (expected vs. observed info)

- For models estimated using ML, the **model log-likelihood (LL)** can also be used to assess relative fit (i.e., through model comparisons)
  - 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 equivalent to ordinary least squares)
  - Two directions: LL **(bigger is better)** or $-2LL$ **(smaller is better)**

Likelihood Ratio Tests (LRTs)

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

1. Calculate $-2\Delta LL$: $(-2LL_{fewer}) - (-2LL_{more})$ OR $-2*(LL_{fewer} - LL_{more})$
2. Calculate Δdf: $(\#\ Parms_{more}) - (\#\ Parms_{fewer})$
3. Compare $-2\Delta LL$ 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 conditional distribution (e.g., logit vs. log-log)

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

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

  - **OR** for unique effect of $x1_i = \exp(\beta_1) = \frac{p(y_i = 1|x1_i = 1)/p(y_i = 0|x1_i = 1)}{p(y_i = 1|x1_i = 0)/p(y_i = 0|x1_i = 0)}$
  - **OR** for unique effect of $x2_i = \exp(\beta_2)$: same principle, but denominator is some reference value (e.g., mean) and numerator is "one unit" higher
  - 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!*

slope	pred logit	pred odds	odds ratio
1	1	2.72	
1	2	7.39	2.72
1	3	20.09	2.72
1	4	54.60	2.72

slope	pred logit	pred odds	odds ratio
-1	-1	0.37	
-1	-2	0.14	0.37
-1	-3	0.05	0.37
-1	-4	0.02	0.37

Converting Across the 3 Outcome 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 (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 (given 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 changes in probability)
- Order of operations: build predicted logit outcome, then logit → 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 SINGLE residual variance (around $\hat{y}_i$) is estimated separately from the fixed effects**
  - Allows direct calculation of R^2 for variance explained and change in R^2 between nested models (and F-tests thereof)

- In contrast, **General*ized* Linear Models** for binary outcomes use a conditional Bernoulli distribution for y_i in which there is **no single separately estimated residual variance (that is constant around $\hat{y}_i$)**
  - 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 some here: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/
  - Btw, STATA LOGIT provides McFadden's $R^2 = 1 - \frac{LL_{model}}{LL_{empty}}$ but the user-created function fitstat provides several others

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 = $\frac{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})}{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})+\boldsymbol{Var}(\boldsymbol{e_i})} = \frac{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})}{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})+\mathbf{3.29}}$
  - M&Z pseudo-R^2 on **probit** scale = $\frac{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})}{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})+\boldsymbol{Var}(\boldsymbol{e_i})} = \frac{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})}{\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})+\mathbf{1.00}}$

  - $\boldsymbol{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{*})$ = variance of the predicted logit outcomes
    - Save model-scale predicted outcomes, then calculate their variance
  - $\boldsymbol{Var}(\boldsymbol{e_i})$ = 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 $c=0,1,2,3$):

0 vs. **1,2,3** (Submodel$_1$) | **0,1** vs. **2,3** (Submodel$_2$) | **0,1,2** vs. **3** (Submodel$_3$)

I've named these submodels based on what they predict, but each program output will name them in 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: $Logit[p(y_i > 0)] = \beta_{01} \rightarrow p(y_i > 0) = exp(\beta_{01})/[1 + exp(\beta_{01})]$
- Submodel 2: $Logit[p(y_i > 1)] = \beta_{02} \rightarrow p(y_i > 1) = exp(\beta_{02})/[1 + exp(\beta_{02})]$
- Submodel 3: $Logit[p(y_i > 2)] = \beta_{03} \rightarrow p(y_i > 2) = exp(\beta_{03})/[1 + exp(\beta_{03})]$

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 $c = 0,1,2,3$):

0 vs. **1,2,3** → Submodel$_1$ → Prob$_1$

0,1 vs. **2,3** → Submodel$_2$ → Prob$_2$

0,1,2 vs. **3** → Submodel$_3$ → Prob$_3$

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

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

- 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 or probit) 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 *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)

Bivariate Association of Binary Variables

- The possible **Pearson's r for binary variables will be limited** when they are not evenly split into 0/1 because their variance depends on their mean
  - Remember: Mean = p_i , Variance = $p_i(1-p_i) = p_i q_i$

- If two binary variables (x_i and y_i) differ in p_i, such that $p_y > p_x$
  - Maximum covariance: $Cov(x,y) = p_x(1-p_y)$
  - This problem is known as **"range restriction"**
  - **Here this means the maximum Pearson's r will be smaller than ± 1 it should be:**

$$r_{x,y} = \sqrt{\frac{p_x(1-p_y)}{p_y(1-p_x)}}$$

  - Some examples using this formula to predict maximum Pearson r values →
  - **So Pearson correlations may not adequately describe relations of categorical variables…**

px	py		max r
0.1	0.2		0.67
0.1	0.5		0.33
0.1	0.8		0.17
0.5	0.6		0.82
0.5	0.7		0.65
0.5	0.9		0.33
0.6	0.7		0.80
0.6	0.8		0.61
0.6	0.9		0.41
0.7	0.8		0.76
0.7	0.9		0.51
0.8	0.9		0.67

Correlations for Binary or Ordinal Variables

- **Pearson correlation**: between two quantitative variables, working with the observed distributions as they actually are
- **Phi correlation**: between two binary variables, still working with the observed distributions (= Pearson with computational shortcut)
- **Point-biserial correlation**: between one binary and one quantitative variable, still working with the observed distributions (and still = Pearson)

Line of Suspended Disbelief to Reduce Impact of Range Restriction

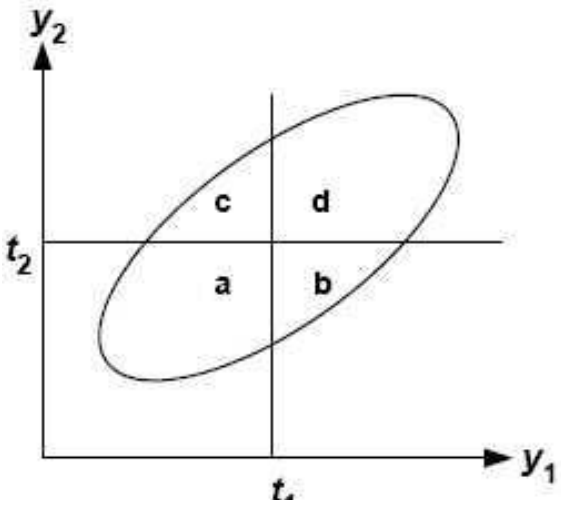
- **Tetrachoric correlation**: between "underlying continuous" distributions of two actually binary variables (not = Pearson) → based on probit!
- **Polychoric correlation**: between "underlying continuous" distributions of two ordinal variables (not = Pearson) → based on probit!
- **(Bi/Poly)serial correlation**: between "underlying continuous" (but really binary/ordinal) and observed quantitative variables (not = Pearson)

- Bivariate statistics related to categorical variables should be provided using **tetrachoric, biserial, or polychoric correlations** instead of Pearson

Tetrachoric and Polychoric Correlation

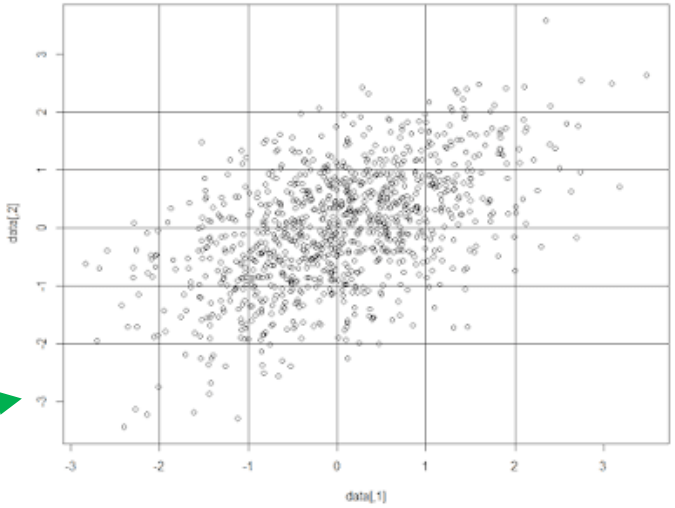
Data	$y_2 = 0$	$y_2 = 1$
$y_1 = 0$	a	c
$y_1 = 1$	b	d

Tetrachoric reasoning: Given a bivariate normal distribution of the underlying continuous variables (y-star version), what correlation would have created the observed proportion in each quadrant (→ cell)?



- Polychoric and tetrachoric correlations are similar:
  - Both based on a bivariate normal distribution,
  - Both try to represent the correlation that would have created the proportion of responses in each cell (unique combo of row by column)

- See this website for a more thorough description with this helpful example of an extension to polychoric!



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	**General*ized* 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 commonly, convert t into d or r)
Can fixed effect estimates be compared directly between models?	Yes	No, because they change scale due to different total variance… see Winship & Mare (1983, 1984)