Generalized Multilevel Models for Two-Level Nested Data

- Topics:
  - Clarifying distribution terminology
  - 3 parts of a generalized (multilevel) model
  - Single-level models for categorical outcomes
  - Complications for generalized multilevel models
  - A brief tour of other kinds of generalized models

Clarifying Distribution Terminology

- The MLM variants we've seen so far all fit under the "**general**" (→ all normal distributions) linear mixed model family:
  - **G** matrix: Variances and covariances of **level-2 random effects** (denoted with $\boldsymbol{U_c}$), which are assumed multivariate normal over clusters
  - **R** matrix: Variances and covariances of **level-1 residuals** (denoted with $\boldsymbol{e_{pc}}$), which are assumed multivariate normal (over persons & clusters)

- e.g., a random slope for cluster-MC $\boldsymbol{WC_{pc}}$ for four persons in a cluster:

L1: $\boldsymbol{y_{pc} = \beta_{0c} + \beta_{1c}(WCx_{pc}) + e_{pc}}$

L2: $\boldsymbol{\beta_{0c} = \gamma_{00} + \gamma_{01}(CMx_c) + U_{0c}}$

$\boldsymbol{\beta_{1c} = \gamma_{10} + U_{1c}}$

Level-2 **G** matrix: TYPE=UN

$$\begin{bmatrix} \tau^2_{U_0} & \tau_{U_{10}} \\ \tau_{U_{01}} & \tau^2_{U_1} \end{bmatrix}$$

Level-1 **R** matrix: TYPE=Diagonal

$$\begin{bmatrix} \sigma^2_e & 0 & 0 & 0 \\ 0 & \sigma^2_e & 0 & 0 \\ 0 & 0 & \sigma^2_e & 0 \\ 0 & 0 & 0 & \sigma^2_e \end{bmatrix}$$

Same Model for $L1n = 4$ in One Cluster

Composite Model: $\gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{pc}) + U_{0c} + U_{1c}(WCx_{pc}) + e_{pc}$

$$\mathbf{Y}_c = \mathbf{X}_c \quad \boldsymbol{\gamma} + \mathbf{Z}_c \quad \mathbf{U}_c + \mathbf{E}_c$$

$$\begin{bmatrix} y_{1c} \\ y_{2c} \\ y_{3c} \\ y_{4c} \end{bmatrix} = \begin{bmatrix} 1 & CMx_c & WCx_{1c} \\ 1 & CMx_c & WCx_{2c} \\ 1 & CMx_c & WCx_{3c} \\ 1 & CMx_c & WCx_{4c} \end{bmatrix} \begin{bmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \end{bmatrix} + \begin{bmatrix} 1 & WCx_{1c} \\ 1 & WCx_{2c} \\ 1 & WCx_{3c} \\ 1 & WCx_{4c} \end{bmatrix} \begin{bmatrix} U_{0c} \\ U_{1c} \end{bmatrix} + \begin{bmatrix} e_{1c} \\ e_{2c} \\ e_{3c} \\ e_{4c} \end{bmatrix}$$

$$\begin{bmatrix} y_{1c} \\ y_{2c} \\ y_{3c} \\ y_{4c} \end{bmatrix} = \begin{bmatrix} \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{1c}) \\ \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{2c}) \\ \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{3c}) \\ \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{4c}) \end{bmatrix} + \begin{bmatrix} U_{0c} + U_{1c}(WCx_{1c}) \\ U_{0c} + U_{2c}(WCx_{2c}) \\ U_{0c} + U_{3c}(WCx_{3c}) \\ U_{0c} + U_{4c}(WCx_{4c}) \end{bmatrix} + \begin{bmatrix} e_{1c} \\ e_{2c} \\ e_{3c} \\ e_{4c} \end{bmatrix}$$

$$\begin{bmatrix} y_{1c} \\ y_{2c} \\ y_{3c} \\ y_{4c} \end{bmatrix} = \begin{bmatrix} \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{1c}) + U_{0c} + U_{1c}(WCx_{1c}) + e_{1c} \\ \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{2c}) + U_{0c} + U_{2c}(WCx_{2c}) + e_{2c} \\ \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{3c}) + U_{0c} + U_{3c}(WCx_{3c}) + e_{3c} \\ \gamma_{00} + \gamma_{01}(CMx_c) + \gamma_{10}(WCx_{4c}) + U_{0c} + U_{4c}(WCx_{4c}) + e_{4c} \end{bmatrix}$$

$\mathbf{X}_c$ = $L1n \times k$ values of **predictors with fixed effects**, so can differ by cluster ($k = 3$ here)

$\boldsymbol{\gamma}$ = $k \times 1$ estimated **fixed effects** → same for all clusters ($k = 3$ here)

$\mathbf{Z}_c$ = $L1n \times u$ values of **predictors with random effects**, so can differ by cluster ($u = 2$ here)

$\mathbf{U}_c$ = $u \times 2$ estimated cluster-specific **random effects** (here, U_{0c} and U_{1c})

$\mathbf{E}_c$ = $L1n \times L1n$ person-specific cluster residuals

Clarifying Distribution Terminology

L1: $y_{pc} = \beta_{0c} + \beta_{1c}(WCx_{pc}) + e_{pc}$

L2: $\beta_{0c} = \gamma_{00} + \gamma_{01}(CMx_c) + U_{0c}$

$\beta_{1c} = \gamma_{10} + U_{1c}$

$$\mathbf{Y_c} = \mathbf{X_c}\boldsymbol{\gamma} + \mathbf{Z_c}\mathbf{U_c} + \mathbf{E_c}$$

Model for the Variance creates $\mathbf{V_c}$ as:

$$\mathbf{V_c} = \mathbf{Z_c}\ \mathbf{G}\ \mathbf{Z_c^T} + \mathbf{R_c}$$

$$\mathbf{V_c} = \begin{bmatrix} 1 & WCx_{1c} \\ 1 & WCx_{2c} \\ 1 & WCx_{3c} \\ 1 & WCx_{4c} \end{bmatrix} \begin{bmatrix} \tau^2_{U_0} & \tau_{U_{01}} \\ \tau_{U_{01}} & \tau^2_{U_1} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ WCx_{1c} & WCx_{2c} & WCx_{3c} & WCx_{4c} \end{bmatrix} + \begin{bmatrix} \sigma^2_e & 0 & 0 & 0 \\ 0 & \sigma^2_e & 0 & 0 \\ 0 & 0 & \sigma^2_e & 0 \\ 0 & 0 & 0 & \sigma^2_e \end{bmatrix}$$

- $\boldsymbol{\mu_c} = \mathbf{X_c}\boldsymbol{\gamma}$ = conditional mean from fixed effects of $\mathbf{Y}$ for cluster c
- The "**marginal**" distribution of total $\mathbf{Y}$ column is: $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\gamma}, \mathbf{V})$
- The "**conditional**" distribution of total $\mathbf{Y}$ column is: $\mathbf{Y}|\mathbf{U} \sim N(\mathbf{X}\boldsymbol{\gamma} + \mathbf{Z}\mathbf{U}, \mathbf{R})$
  - Conditional = after controlling for both fixed and random effects
  - Marginal and conditional "general" MLMs both have same multivariate normal distribution (which makes ML estimation relatively straightforward)

Clarifying Terminology

- **Conditional** distribution: $\mathbf{Y}|\mathbf{U} \sim N(\mathbf{X}\boldsymbol{\gamma} + \mathbf{ZU}, \mathbf{R})$
- Distribution of level-1 residuals: $\mathbf{E} = \mathbf{Y} - \mathbf{X}\boldsymbol{\gamma} + \mathbf{ZU}$, $\mathbf{E} \sim N(\mathbf{0}, \mathbf{R})$

- Thus far in "general" MLms, we could have used the terms "level-1 residual distribution" and "conditional distribution" interchangeably (and I have used the former)
  - "Level-1 residual distribution" is assumed multivariate normal
  - Therefore "conditional distribution" is assumed multivariate normal

- This will not be the case for outcomes with non-normal distributions (and thus, non-normal *conditional* distributions)
  - Level-1 residual variance may not be estimated, even though we still expect the conditional model predictions to be imperfect

Dimensions for Organizing Models

- Outcome type: General (normal) vs. General*ized* (not normal)
- Dimensions of sampling: One (so one variance term per outcome) vs. **Multiple** (so multiple variance terms per outcome) → **OUR WORLD**

- **General Linear Models:** conditionally normal outcome distribution, **fixed effects** (identity link; only one dimension of sampling)
- **General*ized* Linear Models:** **any conditional outcome distribution**, **fixed effects** through **link functions**, no random effects (one dimension)

Note: Least Squares is only for GLM

- **General Linear Mixed Models:** conditionally normal outcome distribution, **fixed and random effects** (identity link, but multiple sampling dimensions)
- **General*ized* Linear Mixed Models:** **any conditional outcome distribution**, **fixed and random effects** through **link functions** (multiple dimensions)

- "Linear" means fixed effects predict the *link-transformed* conditional mean (μ) of DV in a linear combination of (effect*predictor) + (effect*predictor)…

Generalized Linear Models

- **Generalized linear models:** link-transformed conditional mean is predicted instead; ML estimator uses not-normal conditional distributions in the outcome data likelihood
  - **Level-1** conditional model uses some not-normal distribution that may not have a residual variance, but level-2 random effects are still MVN
  - What follows starts with a **single-level** generalized model for now

- Many kinds of non-normally distributed outcomes have some kind of generalized linear model to go with them via ML:
  - Categorical: binary, ordinal (ordered), or nominal (unordered)
  - Counts (discrete, positive values)
  - Censored (piled up and cut off at one end)
  - Zero-inflated (pile of 0's, then some distribution after)
  - Continuous but skewed (long tail)

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:

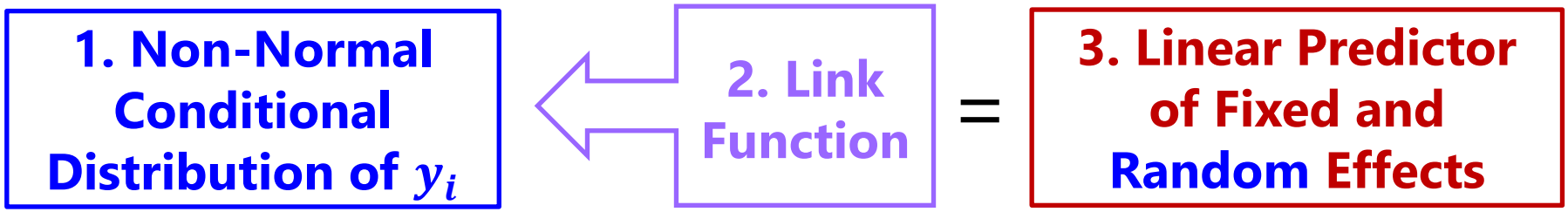
- General linear models use a ***normal*** conditional distribution to describe the y_i variance remaining after prediction via the fixed effects → we call this residual variance, which is estimated separately and **usually assumed constant** across observations (unless modeled otherwise)
- Other distributions 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**" (but ≠ 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 data-scale conditional mean
   - 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 version (when inverse-linked, the predicted probability outcome will stay between 0 and 1)
   - e.g., count outcome: **conditional mean is expected count**, so the 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

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)
- **Fixed effects are no longer determined—**they now must be found through the ML algorithm, the same as any variance-related parameters

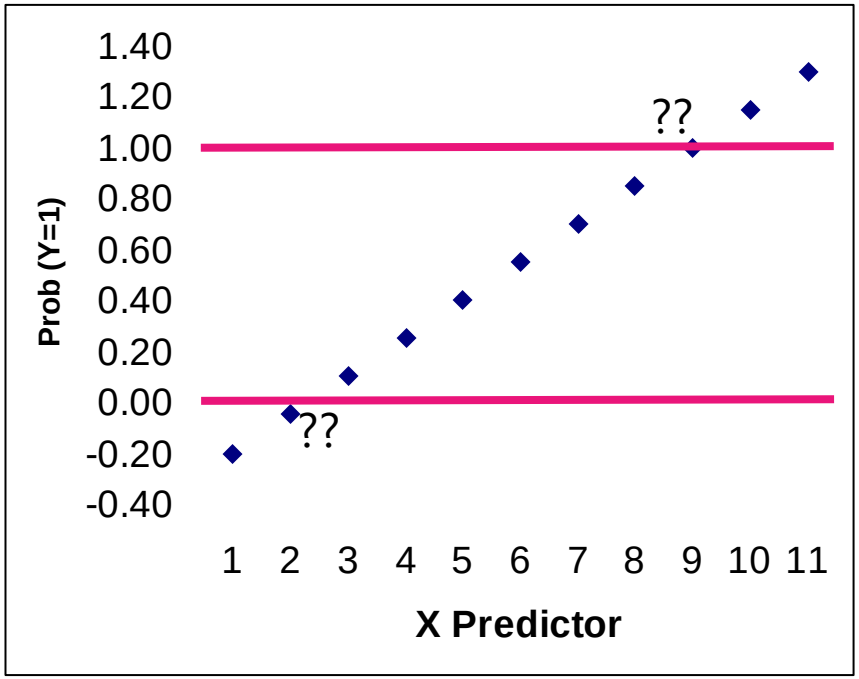
Normal GLM for Binary Outcomes?

- Let's say we have a single binary (0 or 1) outcome…
- 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 <u>binary</u> values
  - Model becomes $\boldsymbol{y_i}$ = **(predicted probability of 1)** + $\boldsymbol{e_i}$
  - **What could possibly go wrong?**

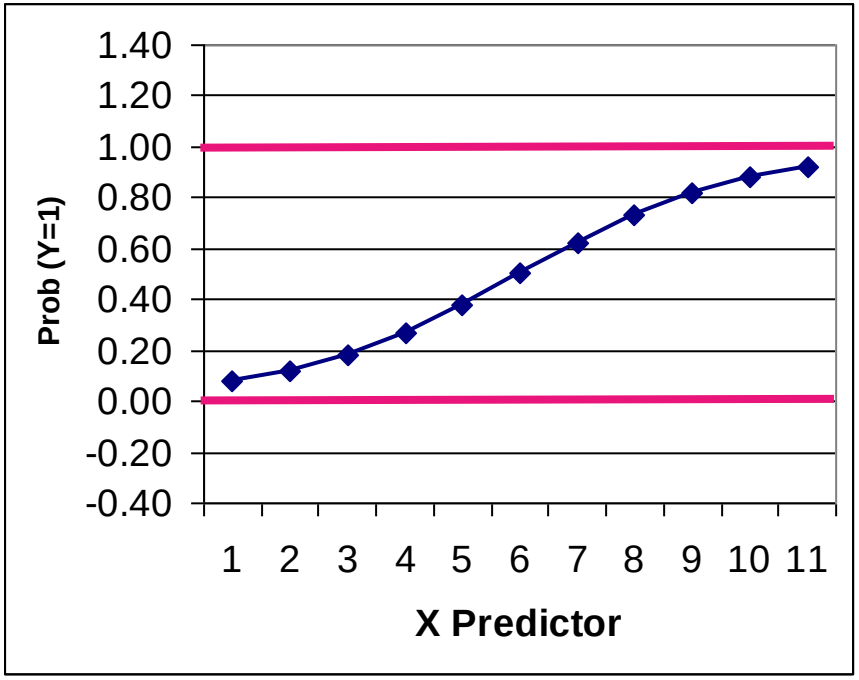
Normal GLM for Binary Outcomes?

- 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…

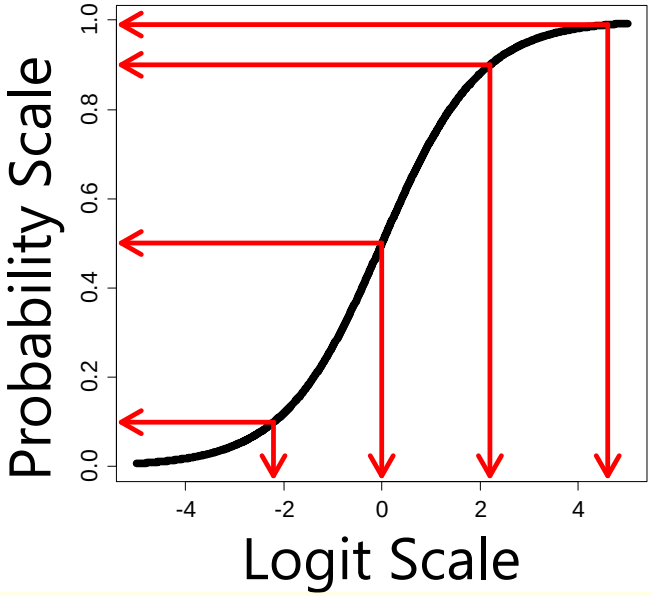


But we need this…



Generalized Models for Binary Outcomes

- Solution to #1: Rather than predict $\boldsymbol{prob(y_i = 1)}$ 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	Logit
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?

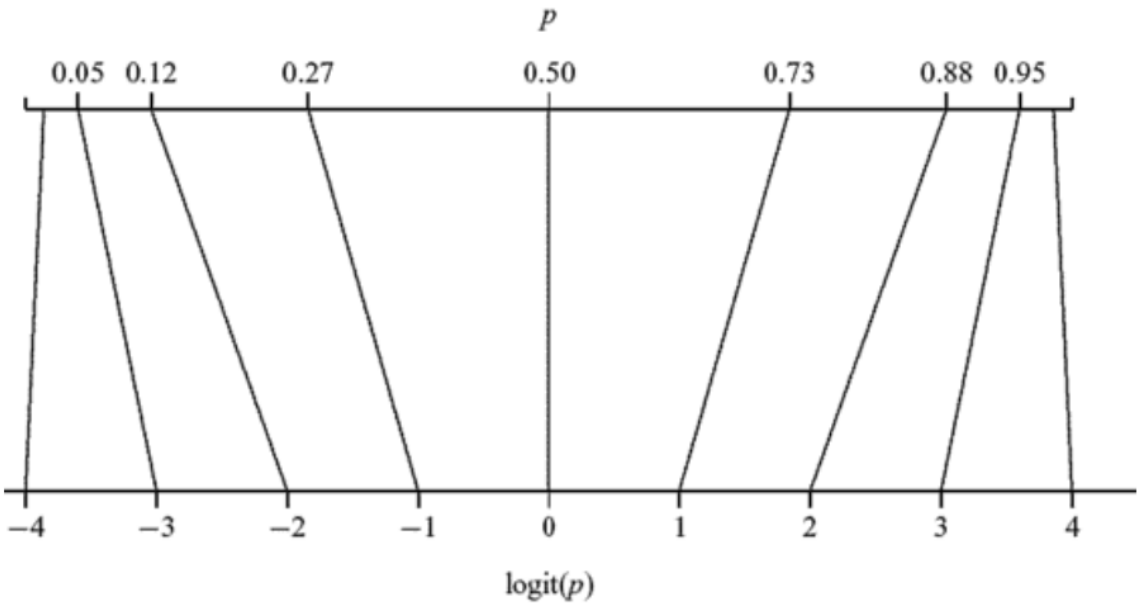
Solution #1: Probability into Logits

- **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

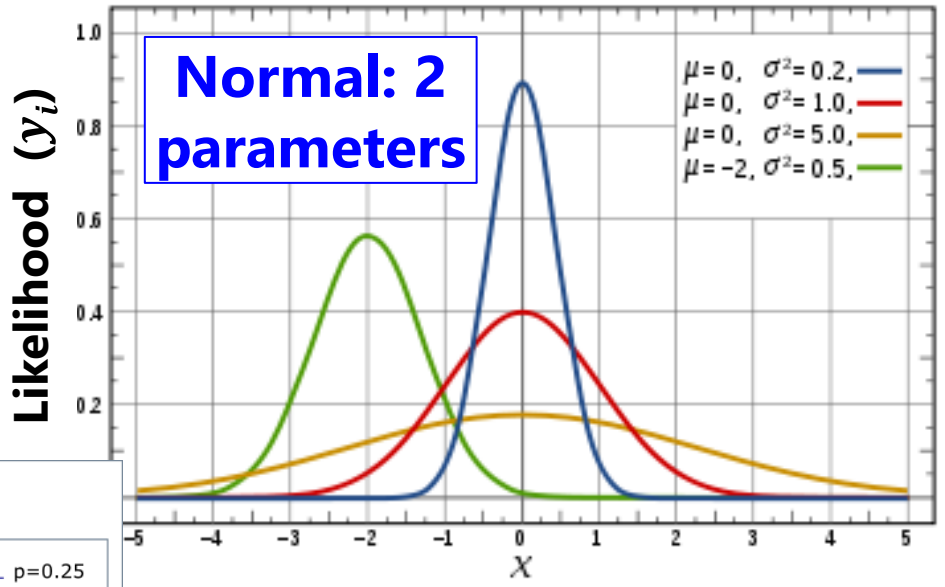
Mean (p)	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
Variance	.0	.09	.16	.21	.24	.25	.24	.21	.16	.09	.0

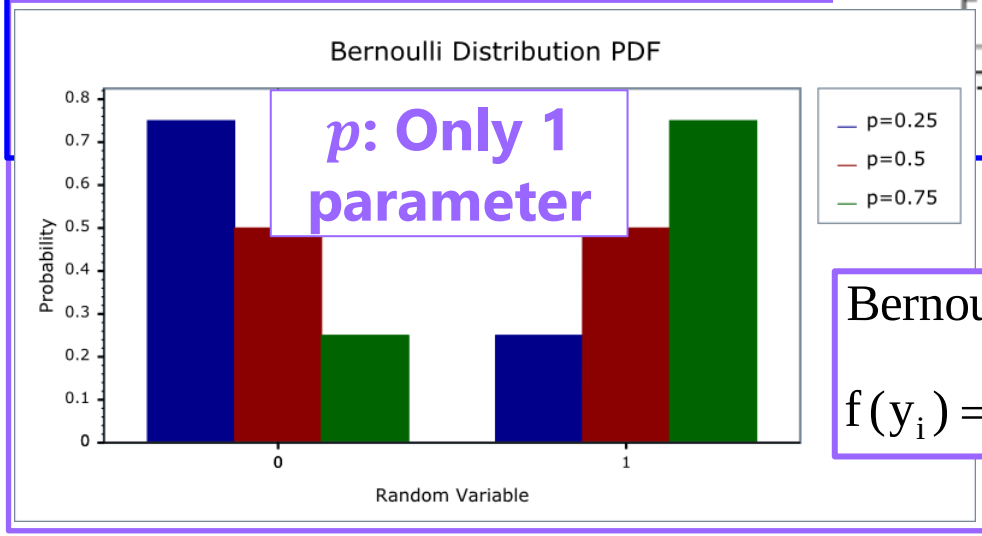
Solution to #2: Bernoulli Distribution

- 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 effects are linear and additive like in GLM, but $\boldsymbol{\beta}$ = difference in **logit** per 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 is usually called a "**logistic regression**" model

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 $x2_i$
Using logits to predict probability (i.e., 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., 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)
- Order of operations: build predicted logit outcome, then → probability

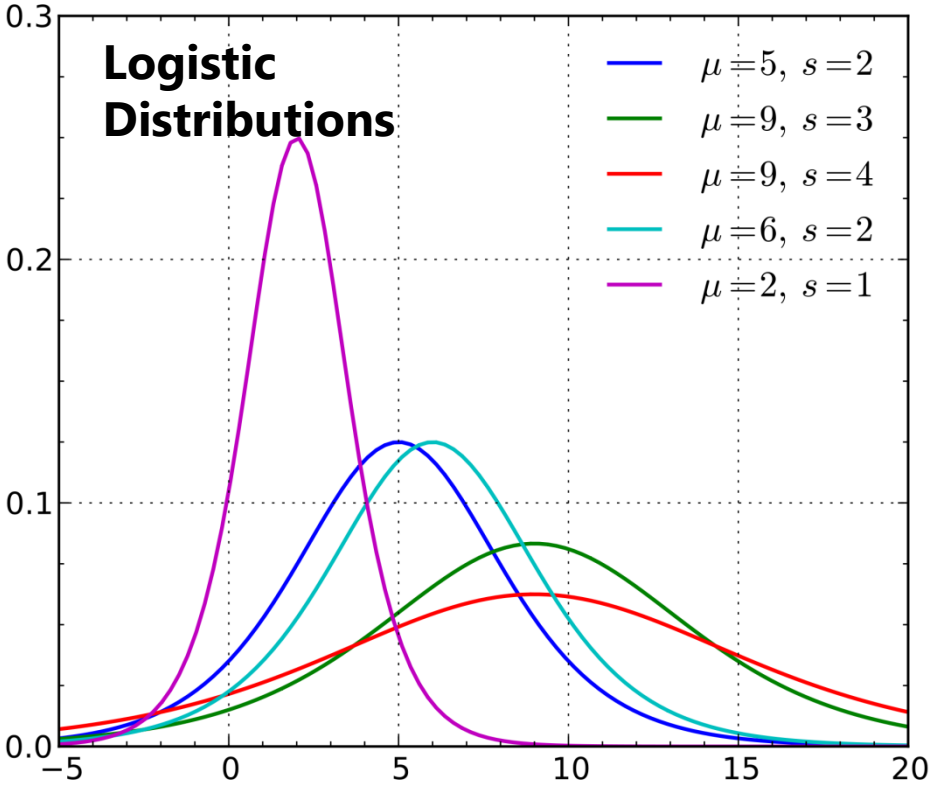
Intercepts (β_0) vs. Thresholds ($-\beta_0$)

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

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

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

  - 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 $e_i \sim \text{Logistic}\left(0, \boldsymbol{\sigma_e^2 = 3.29}\right)$

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

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

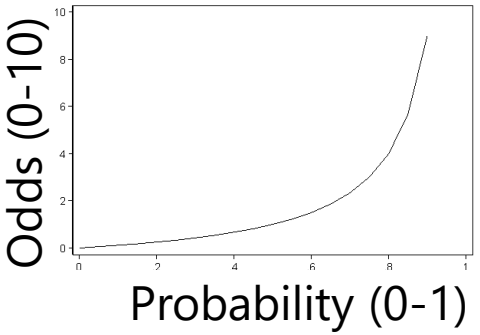


Image borrowed from: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/

Other Link Functions for Binary Data

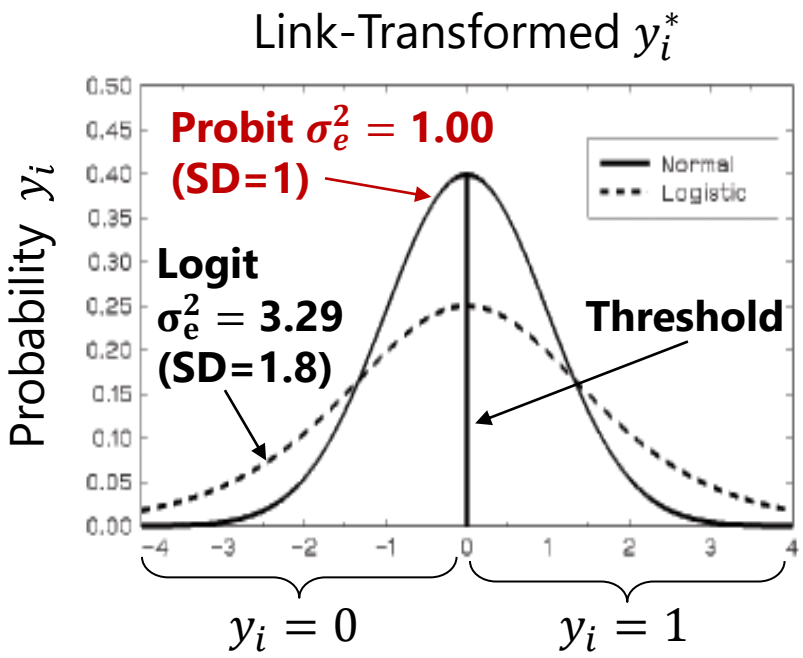
- The idea that a "latent" continuous variable underlies an observed binary response also appears in a "**Probit Regression**" model:

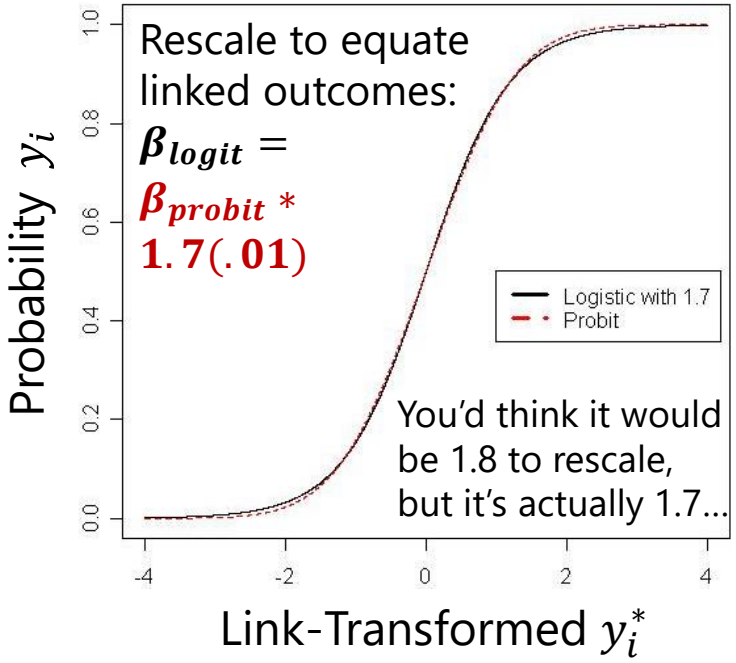
  - A ***probit*** link, such that the linear model predicts a different transformed $\boldsymbol{y_i}$:
    $\text{Probit}(y_i = 1) = \Phi^{-1}[p(y_i = 1)] = linear\ predictor$ ← **g(·) link**
    - Φ = standard normal cumulative distribution function, so the link-transformed $\boldsymbol{y_i}$ **is the 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.



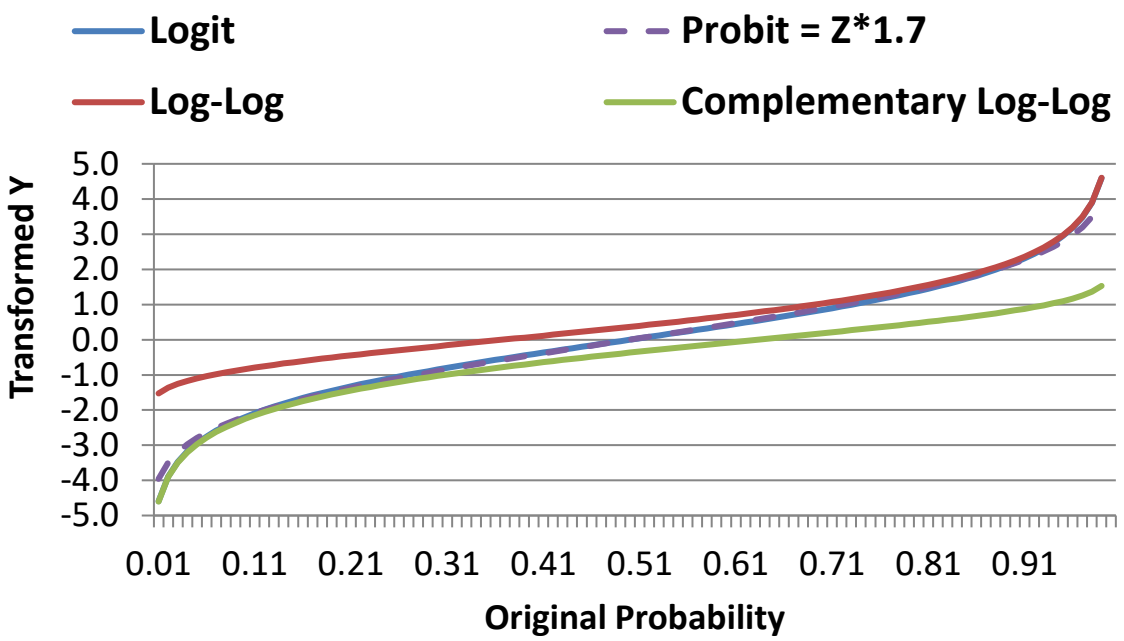


- Other fun facts about probit:
  - **Probit** = "**ogive**" in the Item Response Theory (IRT) world
  - Probit has no odds ratios (because it's not made from odds)

- 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"	$p_i = 1-\exp[-\exp(\hat{y}_i)]$ $\left(0.577, \sigma_e^2 = \frac{\pi^2}{6}\right)$

Too Logit to Quit* https://www.youtube.com/watch?v=HFCv86Olk8E

- 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
- 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**
  - **Red = threshold, blue = intercept!**
- **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)
    - In STATA gllamm (user-created routine); in SAS PROC NLMIXED; in R ???
  - If the proportional odds assumption fails, it may be more understandable to use a nominal model instead (see next slide; dummy-coding to create separate binary outcomes could also 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)

Generalized MLM: Intermediate Summary

- Statistical models use probability distributions
  - Outcomes are assumed to have some **conditional** distribution
  - The normal distribution is one choice, but there are many others: so far we've seen Bernoulli and multinomial
  - ML estimation tries to maximize the height of the data using that chosen distribution along with the model parameters

- Generalized linear models have three parts:
  1. Non-normal conditional outcome distribution
  2. Link function: how bounded conditional mean of y_i gets transformed into something unbounded we can predict linearly
     - So far we've seen identity, logit, probit, log-log, and cumulative log-log
  3. Linear predictor: how we predict that linked conditional mean using fixed (and random) effects

From Single-Level to Multilevel…

- Multilevel generalized models have the same 3 parts as single-level generalized models:
  - Alternative conditional distribution for the outcome (e.g., Bernoulli)
  - Link function to transform bounded conditional mean into unbounded
  - Linear model that directly predicts the linked conditional mean instead

- But in adding random effects (i.e., additional piles of variance) to address dependency in multilevel data:
  - Piles of variance will appear to be ADDED TO, not EXTRACTED FROM, the original residual variance when fixed (e.g., 3.29=logit, 1.0=probit), which causes all coefficients to **change scale** across models
  - ML estimation is way more difficult because normal random effects + not-normal residuals does not have a known distribution like MVN
  - No such thing as REML for generalized multilevel models with true ML
  - Pseudo-R^2 is not possible for level-1 effects (so use odds ratios instead)

Empty Multilevel Model for Binary Outcomes

- **Level 1:** $\boldsymbol{Logit}\,[\boldsymbol{p}(\boldsymbol{y_{pc}} = \boldsymbol{1})] = \boldsymbol{\beta_{0c}}$
- **Level 2:** $\boldsymbol{\beta_{0c}} = \boldsymbol{\gamma_{00}} + \boldsymbol{U_{0c}}$
- **Composite:** $\boldsymbol{Logit}\,[\boldsymbol{p}(\boldsymbol{y_{pc}} = \boldsymbol{1})] = \boldsymbol{\gamma_{00}} + \boldsymbol{U_{0c}}$

Notice what's NOT in level 1…

- $\boldsymbol{\sigma_e^2}$ residual variance is not estimated → **$\pi^2/3$ = 3.29 in logits**
- $\text{Logit ICC} = \frac{\text{BC}}{\text{BC+WC}} = \frac{\boldsymbol{\tau_{U_0}^2}}{\boldsymbol{\tau_{U_0}^2}+\boldsymbol{\sigma_e^2}} = \frac{\boldsymbol{\tau_{U_0}^2}}{\boldsymbol{\tau_{U_0}^2}+\mathbf{3.29}}$

- Can do LRT to see if logit $\boldsymbol{\tau_{U_0}^2} > 0$; the ICC is problematic to interpret on the data scale due to non-constant and not estimated residual variance
- ICC formulas for other outcomes besides binary vary widely
  - Probit link replaces residual variance with 1; others use a function of the mean when the variance is mean-dependent (e.g., Poisson) – see this article in your reading list for details

Example Random Slope Model for Binary Outcomes using Cluster-MC WCx_{pc}

- **Level 1:** $Logit\left[p\left(y_{pc}=1\right)\right]=\beta_{0c}+\beta_{1c}(WCx_{pc})$
- **Level 2:** $\beta_{0c}=\gamma_{00}+\gamma_{01}(CMx_{c})+U_{0c}$
  $\beta_{1c}=\gamma_{10}+\gamma_{11}(CMx_{c})+U_{1c}$

- γ_{01} main effect of CMx_c will reduce level-2 random intercept variance $\tau^2_{U_0}$; γ_{11} cross-level interaction of CMx_c* WCx_{pc} will reduce level-2 random slope variance $\tau^2_{U_1}$ for WCx_{pc}
- σ^2_e residual variance is still not estimated → **$\pi^2/3 = 3.29$**, which means we cannot use it to make a pseudo-R^2 for WCx_{pc} (even though that is still what its fixed slope is trying to reduce)
- Can test new fixed OR random effects with LRTs (−2ΔLL) when using true ML estimation (or use univariate or multivariate Wald test *p*-values for fixed effects, but usually without denominator DF)

Example Random Slope Model for an **Ordinal** Outcome ($y_{pc} = 0, 1, \text{or } 2$)

- **L1:** $\boldsymbol{Logit}\,[\boldsymbol{p}(\boldsymbol{y_{pc}} > \boldsymbol{0})] = \boldsymbol{\beta_{0c1}} + \boldsymbol{\beta_{1c1}}(\boldsymbol{WCx_{pc}})$

  $\boldsymbol{Logit}\,[\boldsymbol{p}(\boldsymbol{y_{pc}} > \boldsymbol{1})] = \boldsymbol{\beta_{0c2}} + \boldsymbol{\beta_{1c2}}(\boldsymbol{WCx_{pc}})$

- **L2:** $\boldsymbol{\beta_{0c1}} = \boldsymbol{\gamma_{001}} + \boldsymbol{U_{0c1}}$ $\boldsymbol{\beta_{1c1}} = \boldsymbol{\gamma_{101}} + \boldsymbol{U_{1c1}}$

  $\boldsymbol{\beta_{0c2}} = \boldsymbol{\gamma_{002}} + \boldsymbol{U_{0c2}}$ $\boldsymbol{\beta_{1c2}} = \boldsymbol{\gamma_{102}} + \boldsymbol{U_{1c2}}$

Last subscript of 1 or 2 is for which submodel

- Cumulative logit link defaults to proportional odds → $\boldsymbol{\gamma_{001}} \neq \boldsymbol{\gamma_{002}}$ but $\boldsymbol{\gamma_{101}} = \boldsymbol{\gamma_{102}}$ and $\boldsymbol{U_{0c1}} = \boldsymbol{U_{0c2}}$ and $\boldsymbol{U_{1c1}} = \boldsymbol{U_{1c2}}$
  - Testable directly using a "partial" proportional odds model in which some can be constrained or indirectly via nominal model (all unequal)
  - $\boldsymbol{\sigma_e^2}$ residual variance is still not estimated → **$\pi^2/3 = 3.29$** (if link=logit)
- Btw, for nominal models (baseline category link), all parameters are separate across submodels by default
  - For more on ordinal and nominal MLMs, see Don Hedeker's slides

New Interpretation of Fixed Effects

- In general MLMS, the fixed effects are interpreted as the "average" effect for the sample, such as in an empty model:
  - e.g., **fixed intercept γ_{00}** is "mean of cluster means"
  - e.g., **random intercept U_{0c}** is "cluster c deviation from sample mean"

- What "average" means in general*ized* MLMs is different, because of the use of nonlinear link functions:
  - e.g., mean of log-transformed(y) ≠ log-transformed mean(y)
  - Therefore, the fixed effects are not the "sample average" effect, they are the effect for ***specifically for corresponding $U_c = 0$***
    - So fixed effects are *conditional* on the random effects
    - This is called a "**unit-specific**" or "**subject-specific**" model
    - This distinction does not exist when using a normal conditional distribution

Comparing Results across Models is Tricky!

- Level-1 fixed effects cannot be compared directly across models, because they are not on the same scale! (Bauer, 2009)
- e.g., if residual variance = 3.29 in logit models:
  - When adding a random intercept variance to an empty model, the **total variation in the outcome has increased** → the fixed effects will increase in size because they are *unstandardized* slopes

$$\gamma_{\text{mixed}} \approx \sqrt{\frac{\tau^2_{U_0} + 3.29}{3.29}} \left(\beta_{\text{fixed}}\right)$$

  - **Level-1 predictors cannot decrease the level-1 residual variance** like usual, so all other model estimates must **increase** to compensate
    - If x_{pc} is uncorrelated with other predictors and is a pure level-1 variable (ICC ≈ 0), then fixed and $SD(U_{0c})$ will increase by same factor
  - **Random effects variances can decrease**, so level-2 fixed effects should be on the same scale across models given the same level-1 model

A Little Bit about Estimation

- Goal: End up with maximum likelihood estimates for all model parameters (because they are consistent and most efficient)
  - Given a conditional normal distribution (i.e., $\mathbf{V}$ matrix based on MVN $\boldsymbol{e_{pc}}$ level-1 residuals and MVN $\boldsymbol{U_c}$ level-2 random effects), ML estimation is relatively easy because we don't need to know the $\boldsymbol{U_c}$ values: the marginal log-likelihood does not include them
  - Given a non-normal conditional distribution (i.e., binary outcomes are Bernoulli after conditioning on the MVN $\boldsymbol{U_c}$ level-2 random effects), ML estimation is much harder because we do need the $\boldsymbol{U_c}$ values in creating linear predictor outcomes and a log-likelihood per cluster

- 3 main families of estimation approaches:
  - Quasi-Likelihood methods ("marginal/penalized quasi ML")
  - Numerical Integration ("adaptive Gaussian quadrature")
  - Also Bayesian methods (MCMC, increasingly available)

Quasi-Likelihood Estimation

- Older methods, also known as "pseudo-likelihood"
  - Predict link-transformed conditional mean using a general MLM
  - "Marginal QL" → linear approximation using fixed part of model
  - "Penalized QL" → linear approximation using fixed + random
  - Come in ML and REML variants (MSPL and RSPL in SAS GLIMMIX)
  - Are the DEFAULT in SAS GLIMMIX and only option in SPSS!

- Why not use them?
  - Provide too small random effects variances (2nd-order PQL is supposed to be better than 1st-order MQL in this regard)
  - THEY DO NOT PERMIT MODEL $-2\Delta LL$ TESTS
    - Modern software may also add a Laplace approximation to QL, which then does permit $-2\Delta LL$ tests (also in SAS GLIMMIX and STATA melogit)

Marginal Maximum Likelihood Estimation

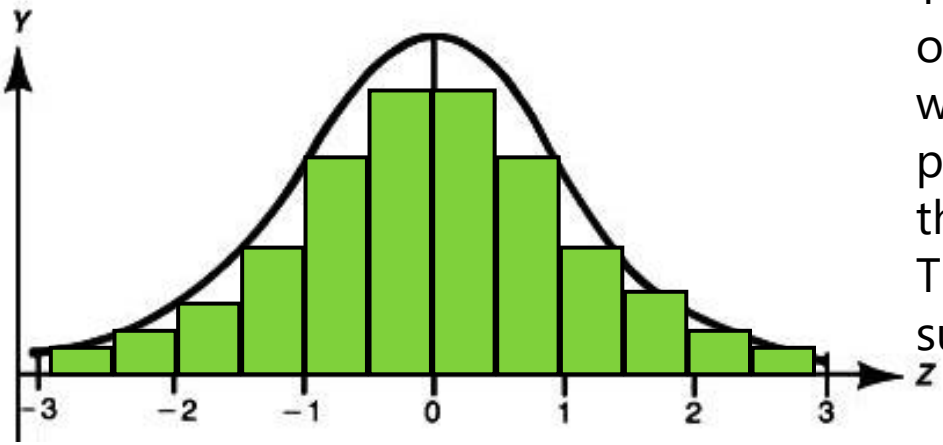
- **ML via Numeric(al) Integration** → gold standard of ML
  - Synonyms: (adaptive) Gaussian quadrature
  - Provides much better estimates and valid $-2\Delta LL$ tests (ML flavor only)
  - Can take forever or not converge at all in models with many random effects; not available for models with crossed random effects
    - "Laplace" approximation can be used, which is equivalent to 1 integration point (???)
  - Start values can help speed estimation (i.e., from QL methods)
  - Relies on assumptions of local independence, like usual → all level-1 dependency has been modeled; level-2 units are independent
  - No such thing as an **R** matrix correlation pattern (only relevant for longitudinal data), so any differences in variance or additional sources of covariance must be specified via random effects in **G**
    - Using _RESIDUAL_ option in SAS GLIMMIX RANDOM statements triggers QL
    - Also no V matrix, so it can be hard to discern the predicted variance pattern

ML via Numeric(al) Integration

- **Step 1**: Select **starting values** for all **fixed effects**
- **Step 2**: Compute the **likelihood** of each observation given by the *current* parameter values using chosen distribution of residuals
  - Model creates link-predicted outcome given parameter estimates, but the $\boldsymbol{U_c}$ **values are not parameters**—their variances and covariances are instead
  - But so long as we can assume the $\boldsymbol{U_c}$ values are MVN, we can still proceed…
  - Computing the likelihood for each set of possible parameters requires *removing* the contribution of the individual $\boldsymbol{U_c}$ values from the model equation—by ***integrating*** across possible $\boldsymbol{U_c}$ values for each level-2 unit
  - Integration is accomplished by "Gaussian Quadrature" → summing up rectangles that approximate the integral (area under the curve) for each level-2 unit
- **Step 3:** Decide if it has the "right" answers, which occurs when the log-likelihood changes very little across iterations (i.e., it converges)
- **Step 4:** If it hasn't converged, choose new parameters values
  - Newton-Rhapson or Fisher Scoring (calculus), EM algorithm (U's =missing data)

ML via Numeric(al) Integration

- More on Step 2: Divide the $\boldsymbol{U_c}$ distribution into rectangles
  - → "Gaussian Quadrature" (# rectangles = # "quadrature points")
  - First divide the whole $\boldsymbol{U_c}$ distribution into rectangles, then repeat by taking the most likely section for each level-2 unit and rectangling that
    - This is "adaptive quadrature" and is computationally more demanding, but gives more accurate results with fewer rectangles (defaults differ by program)



The likelihood of each level-2 unit's outcomes at each $\boldsymbol{U_c}$ rectangle is then weighted by that rectangle's probability of being observed (from the multivariate normal distribution). The weighted likelihoods are then summed across all rectangles…

→ ta da! "**numerical integration**"

Example of Numeric Integration:
Binary DV, Fixed x_{pc} Slope, Random Intercept

1. Start with values for fixed effects: intercept: $\gamma_{00} = 0.5$, x_{pc}: $\gamma_{10} = 1.5$,
2. Compute likelihood for real data based on fixed effects and plausible U_{0c} = (−2,0,2) using model: $Logit(y_{pg} = 1) = \gamma_{00} + \gamma_{10}(x_{pc}) + U_{0c}$
   - Here for one cluster of two persons with $y_{pc} = 1$ for both persons

			IF y=1	IF y=0		Likelihood	U0	U0	Product
	U0 = -2	**Logit**	**Prob**	**1-Prob**		**if both y=1**	**prob**	**width**	**per U0**
x=0	(0.5+0-2)	-1.5	0.18	0.82		0.091213	0.05	2	0.00912
x=1	(0.5+1.5-2)	0.0	0.50	0.50					
	U0 = 0	**Logit**	**Prob**	**1-Prob**					
x=0	(0.5+0-0)	0.5	0.62	0.38		0.54826	0.40	2	0.43861
x=1	(0.5+1.5-0)	2.0	0.88	0.12					
	U0 = +2	**Logit**	**Prob**	**1-Prob**					
x=0	(0.5+0+2)	2.5	0.92	0.08		0.90752	0.05	2	0.09075
x=1	(0.5+1.5+2)	4.0	0.98	0.02					
Overall Likelihood (Sum of Products over All U0 Values):									**0.53848**
(do this for each person, then multiply this whole thing over all clusters)									
(repeat with new values of fixed effects until find highest overall likelihood)									

Summary: Complications of Generalized Multilevel Models

- Analyze link-transformed conditional mean (e.g., via logit, probits…)
  - ***Linear*** relationship: predictors → ***link-transformed*** conditional mean outcome
  - ***Nonlinear*** relationship: predictors → ***data-scale*** conditional mean outcome
    - Conditional outcomes (after fixed+random effects) then follow a non-normal distribution

- In models for binary or categorical outcomes, level-1 residual variance is fixed and varies with the conditional mean (gets smaller → boundaries)
  - So it can't go decrease after being explained by level-1 predictors, which means that the scale of all model parameters must go UP to compensate
  - Scale of model will also be different after adding random effects for the same reason—the total variation in the model is now bigger
  - Fixed effects may not be comparable across models as a result

- Estimation is trickier, takes longer, and true ML does not come in REML flavor
  - Numerical integration is best but may blow up in complex models
  - Start values are often essential (can get those with pseudo-likelihood estimators)

A Taxonomy of Not-Normal Outcomes

- **"Discrete" outcomes**—all responses are **whole** numbers
  - **Categorical variables** in which **values are labels**, not numbers
    - Bernoulli (2 options) or multinomial (3+ options) distributions
    - Question: Are the values ordered → **Which link function?**
  - **Count of things that happened**, so values < 0 cannot exist
    - Outcome values range from 0 to +∞ (whole numbers only)
    - Usually some kind of Poisson or Negative Binomial distribution
    - **Usually log link so predicted outcomes can't go below 0**
    - Questions: Which conditional distribution? Are there *extra* 0 values?

- **"Continuous" outcomes**—responses can be **any** number
  - Question: What does the conditional distribution look like?
    - Symmetric or skewed? Are there boundaries?

There's (Pry) a Model for That!

- Many kinds of **non-normal outcomes** can be analyzed with generalized MLMs through the **magic of ML (or Bayes)**
  - Can be fewer choices in MLM than for single-level models (as in next slides from PSQF 6270 Generalized Linear Models)

- **Two parts: Link function + other conditional distribution**
  - **Binary** → **Logit** + **Bernoulli**
  - **Ordinal** or **Nominal** → **Logit** + **Multinomial**
  - **Proportion** → **Logit** + **Binomial/Beta-Binomial**
  - **Count** → **Log** + **Poisson/Negative Binomial**
  - **Censored** → **Tobit** + **Normal/Bernoulli**
  - **Skewed Continuous** → **Log** + **Log-Normal/Gamma**
  - **Bimodal Continuous** → **Logit** + **Beta**
  - **Zero-Inflated** (if and how much) → **Logit/Log** + **Bernoulli/other**

Too Logit to Quit: Predicting Proportions

- **Logit-type links** can be useful in predicting **proportions**:
  - Range between 0 and 1, so model needs to "shut off" predictions for conditional mean as they approach those ends, just as in binary data
    - **We are predicting the logit of p_i, the probability of $y_i = 1$ for any trial, when multiplied by the # trials, it becomes predicted # of 1 values = μ_i**
  - Any outcome can be transformed to range between 0 and 1 to be modeled this way: Proportion = $(y_i - \min)/(\max - \min)$
  - Data to model: → predict $\hat{y}_i$ in logits $= \text{Log}\left(\frac{p_i}{1-p_i}\right)$ ← **$g(\cdot)$ Link**
  - Model back to data → $p_i = \frac{exp(\hat{y}_i)}{1+exp(\hat{y}_i)}$ ← **$g^{-1}(\cdot)$ Inverse-Link**

- Odds ratios can be used as effect size: OR = exp(slope)
- Distributions? Binomial (discrete), Beta (continuous), or hybrid
  - **Binomial**: Less flexible (just one hump), but can include 0 and 1 values
  - **Beta**: Way more flexible (but ????), but cannot directly include 0 or 1 values
  - **Beta-binomial**: Flexible hybrid well-suited for multiplicative overdispersion (see also "observation-level random effects" for additive overdispersion)

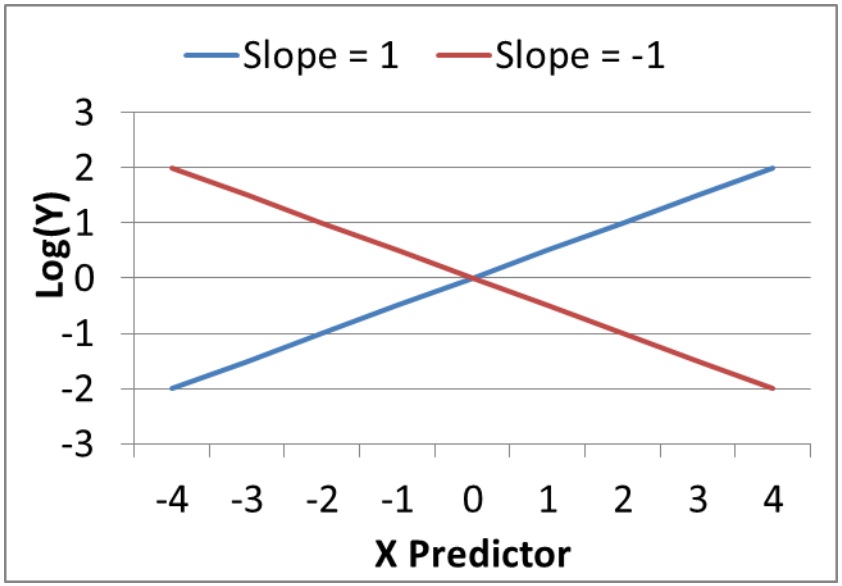
Natural Log Link for Count Outcomes

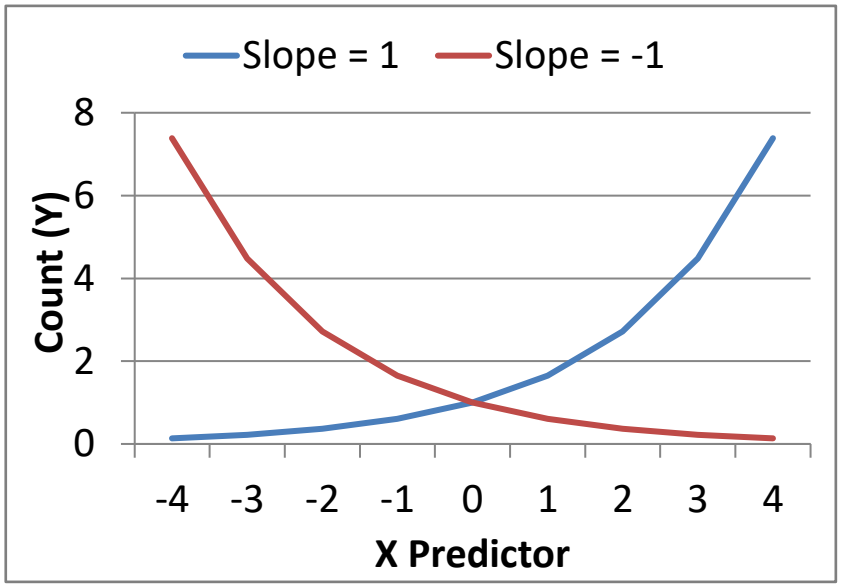
This is an ***unbounded*** **linear model** that predicts the Log of the Expected Count…

$$Log[E(y_i)] = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}(\boldsymbol{x_i})$$

…that becomes an expected count bounded at 0 via an inverse link of exp(log count):

$$[E(y_i)] = \boldsymbol{exp}[\boldsymbol{\beta_0} + \boldsymbol{\beta_1}(\boldsymbol{x_i})]$$





Models for Count Outcomes

- **Counts**: non-negative integer responses (unbounded positive)
  - **Link:** $\mathrm{g}(\bullet)\ Log[E(y_i)] = Log(\mu_i) = [\mathrm{model}]$ → predicts log of count as $\hat{y}_i$
  - **Inverse Link:** $\mathrm{g}^{-1}(\bullet)\ E(y_i) = \exp(\hat{y}_i)$→ to un-log $\hat{y}_i$ back to expected count
  - e.g., if the model-scale predicted log count: $Log(\hat{\mu}_i) = \hat{y}_i = -1$, the data-scale expected count is: $\hat{\mu}_i = exp(-1) = 0.368$
    - So even though counts are only integers, expected counts are not!
  - Btw, you can control for differences in time measured via an **offset** (or **exposure**) log-transformed predictor variable whose slope is fixed =1

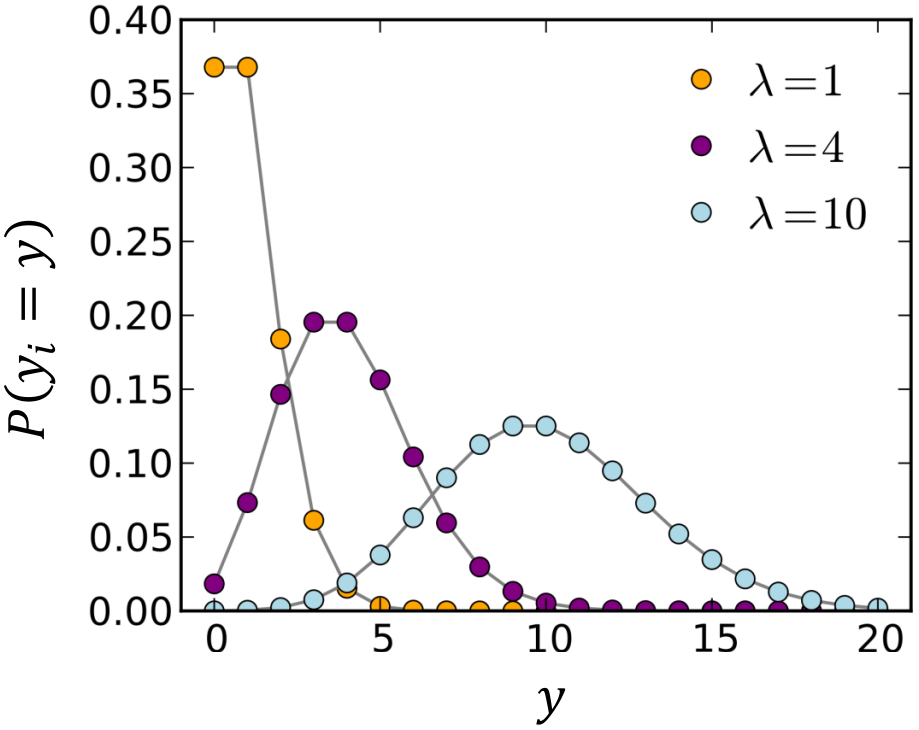
- $\mathbf{exp}(\boldsymbol{\beta_x})$ gives an effect size called an "**incidence-rate ratio**" (**IRR**) that is on same scale as an odds ratio (IRR = 1 means no effect)
  - e.g., IRR = 1.25 for $x_i = 0$ or 1? $x_i = 1$ counts are "25% higher"
  - e.g., IRR = 0.75 for $x_i = 0$ or 1? $x_i = 1$ counts are "25% lower"
  - Stata also gives McFadden's **pseudo-R^2** $= 1 - (LL_{model}/LL_{empty})$

- Choosing the "right" **conditional distribution** is the tricky part!
  - Poisson and Negative Binomial are most common variants

Poisson Conditional Distribution

- Poisson distribution has **one parameter, λ,** which is both its mean and its variance (so λ = mean μ = variance in Poisson)
- PDF: $f(y_i) = Prob(y_i = y) = \frac{exp(-\mu)*\mu^y}{y!}$

$y!$ = factorial of y = gamma function $\Gamma(y+1)$



The dots indicate that only integer values are observed.

Distributions with a small expected value (mean λ) are predicted to have a lot of 0's.

Once $\lambda > 6$ or so, the shape of the distribution is close to a that of a normal distribution.

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

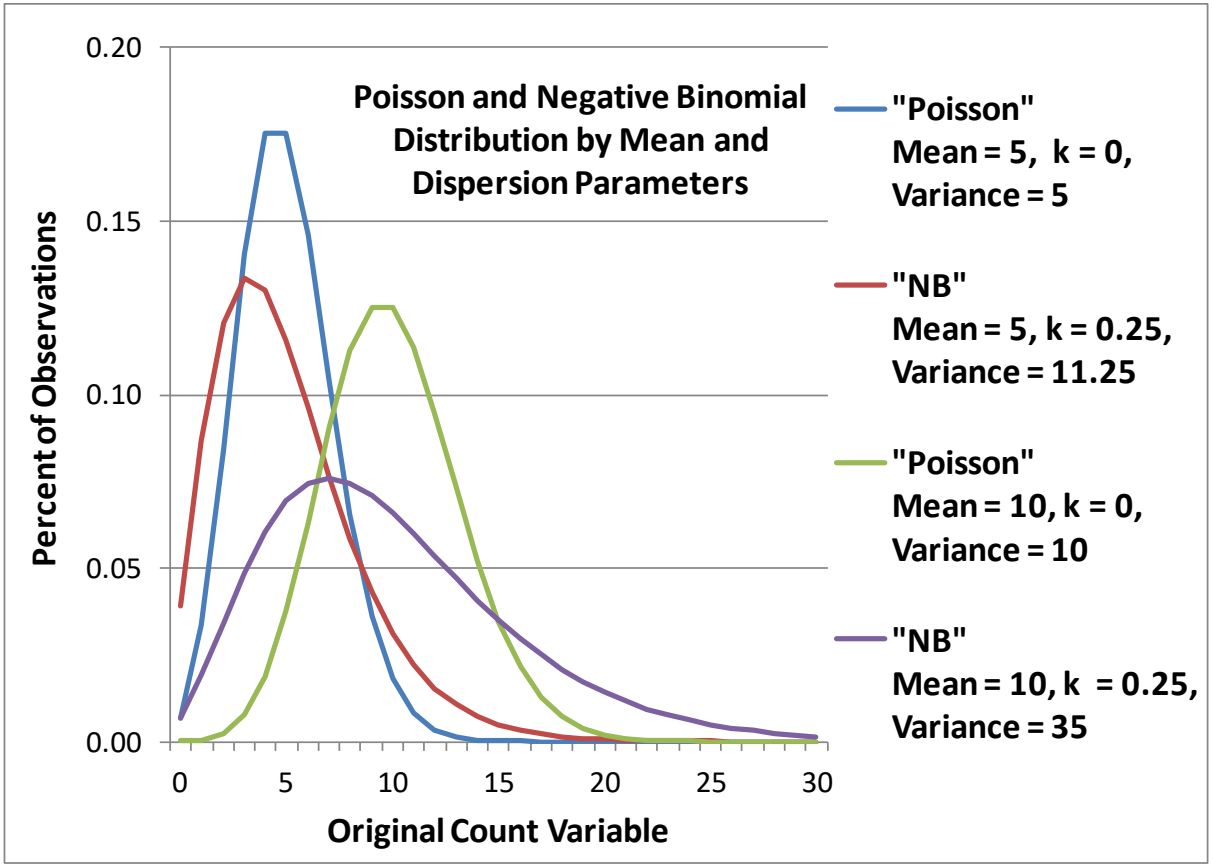
When Variance > Mean = Over-dispersion

- To fix it, we must add a parameter that allows the variance to exceed the mean… it is then a **Negative Binomial (Negbin)** distribution
  - Two types of extra variance: constant = NB-1, quadratic = NB-2 (better)

- **NB-2** has **mean μ** and **dispersion** = **"scale" k** (or $1/k = \theta$ instead):
  - PDF: $Prob(y_i = y) = \frac{\Gamma\left(y+\frac{1}{k}\right)}{\Gamma(y+1)*\Gamma\left(\frac{1}{k}\right)} * \left(\frac{1}{1+k\mu}\right)^{\frac{1}{k}} * \left(1 - \frac{1}{1+k\mu}\right)^{y}$
  - **k is a multiplier:** $Var(y_i) = \mu + k\mu^2$ (so Negbin ≈ Poisson if $k = 0$)
  - Can test if $k > 0$ via LRT (-2ΔLL), although LL for $k = 0$ is undefined
  - In SAS GLIMMIX DIST = NEGBIN (as k = "scale"); STATA NBREG or GLM (as k = "alpha"); R VGAM, MASS (as θ), or PSCL (as θ); more about R here

- An alternative model with the same idea is **generalized Poisson**:
  - Mean: $\frac{\mu}{1-k}$, Variance: $\frac{1}{(1-k)^3}$, so LL is actually defined for $k = 0$
  - Much less commonly used, though

Negative Binomial (NB) = "Stretchy" Poisson…



$\text{Mean} = \mu$
$\text{Dispersion} = k$

$$Var(y_i) = \mu + k\mu^2$$

A Negative Binomial model can be useful for count outcomes with extra skewness.

- Because its k dispersion parameter is fixed to 0, the Poisson model is nested within the Negative Binomial model—to test improvement in fit:
- Is $-2\left(LL_{Poisson} - LL_{NegBin}\right) > 3.84$ for $DF = 1$? Then if $p < .05$, keep NB
- If using a mixture of $DF = 0$ and $DF = 1$, use $-2\Delta LL > 2.71$ instead

More on Generalized MLM: Summary

- There are many options for "amount" variables whose residuals may not be normally distributed
  - Discrete Counts: Poisson, Negative Binomial
  - Continuous Amounts: Lognormal, Gamma, Beta
  - Too many 0's: Zero-inflated or hurdle for discrete; two-part for continuous

- Multilevel versions of most generalized models *can* be estimated…
  - But it's harder to do and takes longer due to numeric integration (trying on all combinations of random effects at each iteration)
  - But there are fewer ready-made options for modeling differential variance/covariance across DVs (no R matrix structures in true ML)

- Program documentation will always be your friend to determine exactly what a given model is doing!