Generalized Multilevel Models for Non-Normal Longitudinal Data

- Today's class:
  - Clarifying distribution terminology
  - 3 parts of a generalized (multilevel) model
  - Models for binary, proportion, and categorical outcomes
  - Complications for generalized multilevel models
  - A brief tour of other generalized models:
    - Models for discrete count or continuous skewed outcomes
    - Models for two-part discrete or continuous outcomes

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: Holds variances and covariances of **level-2 random effects** (denoted with U), which are assumed multivariate normal
  - **R** matrix: Holds variances and covariances of **level-1 residuals** (denoted with e), which are also assumed multivariate normal

- e.g., a random linear time model for four occasions:

Level 1: $\mathbf{y_{ti} = \beta_{0i} + \beta_{1i}(Time_{ti}) + e_{ti}}$

Level 2: $\mathbf{\beta_{0i} = \gamma_{00} + U_{0i}}$

$\mathbf{\beta_{1i} = \gamma_{10} + U_{1i}}$

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

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

Level-1 **R** matrix: REPEATED TYPE=VC

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

Composite: $\mathbf{y_{ti} = (\gamma_{00} + U_{0i}) + (\gamma_{10} + U_{1i})(Time_{ti}) + e_{ti}}$

The SAME Random Linear Time Model written another, more combined way

- Scalar “mixed” model equation per person:

$$\mathbf{Y}_i = \boxed{\mathbf{X}_i * \boldsymbol{\gamma}} + \boxed{\mathbf{Z}_i * \mathbf{U}_i + \mathbf{E}_i}$$

$$\begin{bmatrix} y_{0i} \\ y_{1i} \\ y_{2i} \\ y_{3i} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \gamma_{00} \\ \gamma_{10} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} U_{0i} \\ U_{1i} \end{bmatrix} + \begin{bmatrix} e_{0i} \\ e_{1i} \\ e_{2i} \\ e_{3i} \end{bmatrix}$$

$$\begin{bmatrix} y_{0i} \\ y_{1i} \\ y_{2i} \\ y_{3i} \end{bmatrix} = \begin{bmatrix} \gamma_{00} + \gamma_{10}(0) \\ \gamma_{00} + \gamma_{10}(1) \\ \gamma_{00} + \gamma_{10}(2) \\ \gamma_{00} + \gamma_{10}(3) \end{bmatrix} + \begin{bmatrix} U_{0i} + U_{1i}(0) \\ U_{0i} + U_{1i}(1) \\ U_{0i} + U_{1i}(2) \\ U_{0i} + U_{1i}(3) \end{bmatrix} + \begin{bmatrix} e_{0i} \\ e_{1i} \\ e_{2i} \\ e_{3i} \end{bmatrix}$$

$$\begin{bmatrix} y_{0i} \\ y_{1i} \\ y_{2i} \\ y_{3i} \end{bmatrix} = \begin{bmatrix} \gamma_{00} + \gamma_{10}(0) + U_{0i} + U_{1i}(0) + e_{0i} \\ \gamma_{00} + \gamma_{10}(1) + U_{0i} + U_{1i}(1) + e_{1i} \\ \gamma_{00} + \gamma_{10}(2) + U_{0i} + U_{1i}(2) + e_{2i} \\ \gamma_{00} + \gamma_{10}(3) + U_{0i} + U_{1i}(3) + e_{3i} \end{bmatrix}$$

$\mathbf{X}_i$ = *n* x *k* values of **predictors with fixed effects**, so can differ per person (*k* = 2: intercept, linear time)

$\boldsymbol{\gamma}$ = *k* x 1 estimated **fixed effects**, so will be the same for all persons (γ_{00} = intercept, γ_{10} = linear time)

$\mathbf{Z}_i$ = *n* x *u* values of level-1 predictors with level-2 random effects, so can differ per person (*u* = 2: intercept, linear time)

$\mathbf{U}_i$ = *u* x 2 estimated individual level-2 **random effects**, so can differ per person

$\mathbf{E}_i$ = *n* x *n* time-specific level-1 residuals, so can differ per person

Clarifying Distribution Terminology

Level 1: $\mathbf{y_{ti} = \beta_{0i} + \beta_{1i}(Time_{ti}) + e_{ti}}$

Level 2: $\mathbf{\beta_{0i} = \gamma_{00} + U_{0i}}$

$\mathbf{\beta_{1i} = \gamma_{10} + U_{1i}}$

$$\mathbf{Y_i} = \boxed{\mathbf{X_i}\gamma} + \boxed{\mathbf{Z_i U_i} + \mathbf{E_i}}$$

$\boldsymbol{\mu}_i = \mathbf{X}_i\gamma$ where $\boldsymbol{\mu}_i$ = **Conditional Mean** created by **fixed effects** in the model for means

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

$$\mathbf{V_i} = \mathbf{Z_i} * \mathbf{G_i} * \mathbf{Z_i^T} + \mathbf{R_i}$$

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

- This model says the "**marginal**" distribution of the total column of $\mathbf{Y}$ outcomes is: $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\gamma}, \mathbf{V})$
- This model says the "**conditional**" distribution of the total column of $\mathbf{Y}$ outcomes is: $\mathbf{Y}|\mathbf{U} \sim N(\mathbf{X}\boldsymbol{\gamma} + \mathbf{ZU}, \mathbf{R})$
  - Conditional = after controlling for fixed and random effects
  - Marginal and conditional "general" models both have same normal distribution (which makes ML estimation relatively straightforward)

Clarifying Terminology

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

- Thus far in "general" linear mixed models, 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
  - "Conditional distribution" is assumed multivariate normal

- This may not be the case for outcomes with non-normal distributions (and thus, non-normal *conditional* distributions)
  - Level-1 residual variance may not be estimated, so there may not be such a thing as a separately calculated "level-1 residual", 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

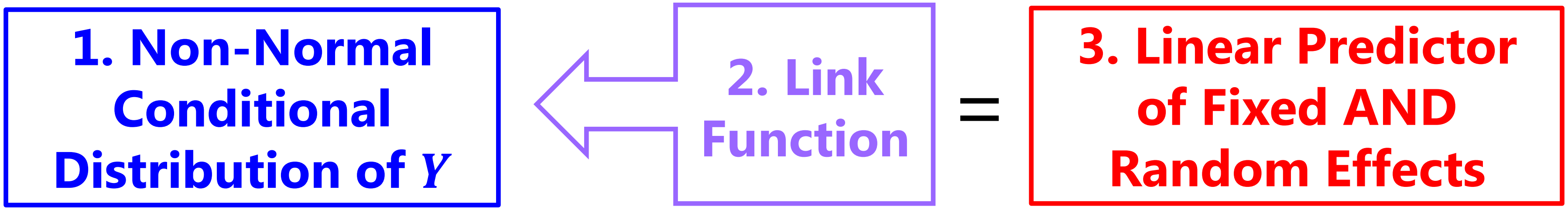
- **General Linear Mixed Models:** conditionally normal outcome distribution, **fixed and random effects** (identity link, but multiple sampling dimensions)
- **General*ized* Linear Mixed Models:** **any conditional outcome distribution**, **fixed and random effects** through **link functions** (multiple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

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

General*ized* Linear Models

- **General*ized* 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
- Many kinds of non-normally distributed outcomes have some kind of generalized linear model to go with them via ML:
  - Binary (dichotomous)
  - Unordered categorical (nominal)
  - Ordered categorical (ordinal)
  - 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 data (long tail)

3 Parts of Generalized Multilevel Models



1. Non-Normal Conditional Distribution of Y

2. Link Function

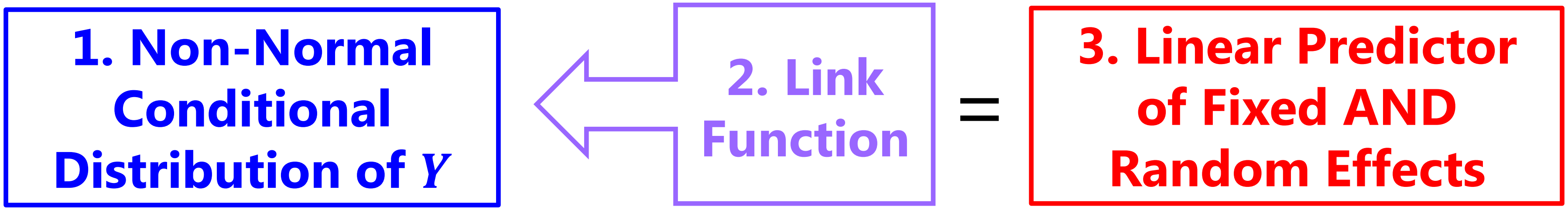
=

3. Linear Predictor of Fixed AND Random Effects

1. Non-normal conditional distribution of Y:

- General MLM uses a *normal* conditional distribution to describe the Y variance remaining after fixed + random effects → we called this the level-1 residual variance, which is estimated separately and usually assumed constant across observations (unless modeled otherwise)
- Other distributions will be more plausible for bounded/skewed outcomes, so ML function maximizes the likelihood using those instead
- **Why?** To get the most correct **standard errors** for fixed effects
- Although you can still think of this as *model for the variance*, not all conditional distributions will actually have a separately estimated level-1 residual variance (e.g., binary → Bernoulli, count → Poisson)

3 Parts of Generalized Multilevel Models



1. Non-Normal Conditional Distribution of Y

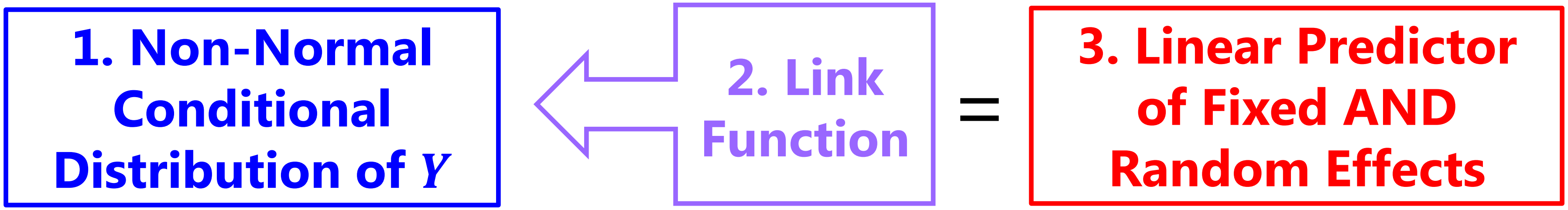
2. Link Function

=

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 Y scale
   - Predicted outcomes (found via inverse link) will then stay within bounds
   - e.g., binary outcome: conditional mean to be predicted is probability of $Y = 1$, so the model predicts a linked version (when inverse-linked, the predicted outcome will stay between a probability of 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., for normal outcome: an "identity" link function (Y * 1) is used given that the conditional mean to be predicted is already unbounded…

3 Parts of Generalized Multilevel Models



1. Non-Normal Conditional Distribution of *Y*

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 original conditional mean
   - That way we can still use the familiar "one-unit change" language to describe effects of model predictors (on the linked conditional mean)
   - You can think of this as "model for the means" still, but it also includes the level-2 random effects for dependency of level-1 observations
   - Fixed effects are no longer determined: they now have to be found through the ML algorithm, the same as the variance parameters

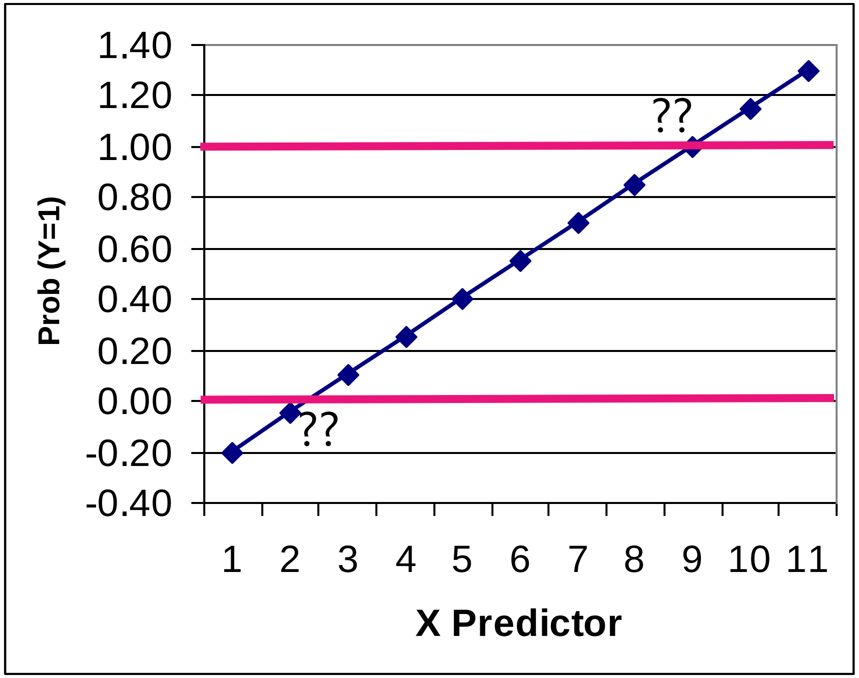
Normal GLM for Binary Outcomes?

- Let's say we have a single binary (0 or 1) outcome… *(concepts for longitudinal data will proceed similarly)*
  - Expected mean is proportion of people who have a 1, so the **probability of having a 1** is the conditional mean we're trying to predict for each person: $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1})$
  - General linear model: $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1}) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}\mathbf{X_i} + \boldsymbol{\beta_2}\mathbf{Z_i} + \mathbf{e_i}$
    - $\boldsymbol{\beta_0}$ = expected probability when all predictors are 0
    - $\boldsymbol{\beta}$'s = expected change in $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1})$ for a one-unit Δ in predictor
    - $\mathbf{e_i}$ = difference between observed and predicted binary values
  - Model becomes $\mathbf{y_i} = (\textbf{predicted probability of 1}) + \mathbf{e_i}$
  - **What could possibly go wrong?**

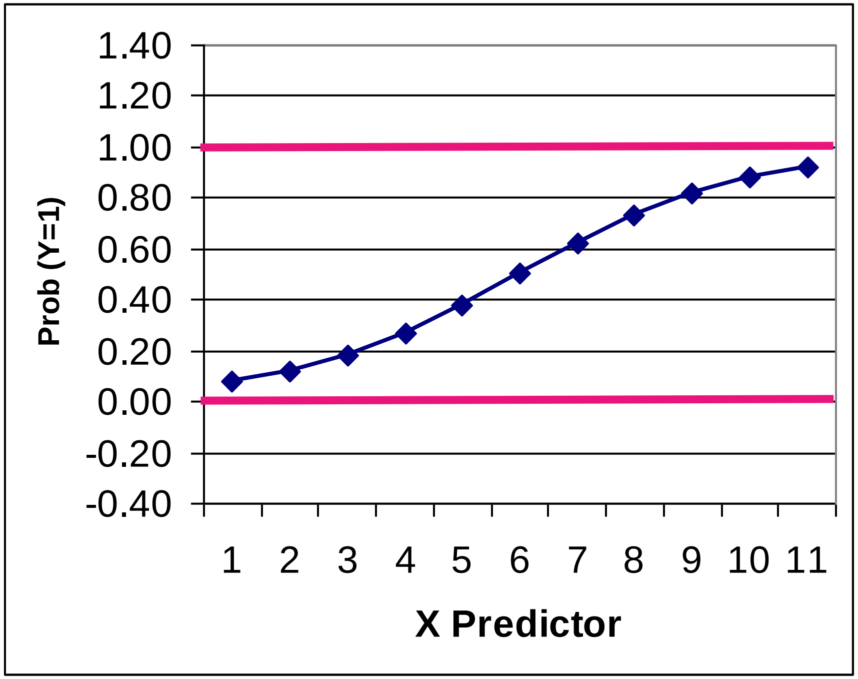
Normal GLM for Binary Outcomes?

- Problem #1: A **linear** relationship between X and Y???
- 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...

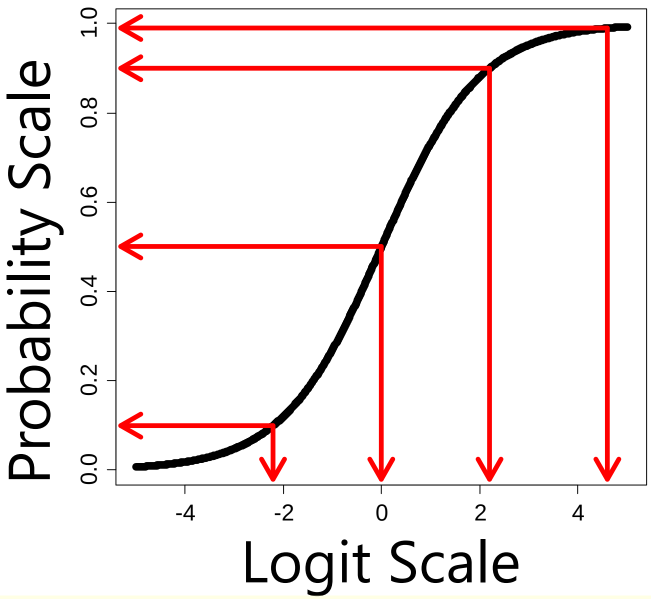


But we need this...



Generalized Models for Binary Outcomes

- Solution to #1: Rather than predicting $\boldsymbol{p(\mathrm{y_i}=1)}$ directly, we must transform it into an unbounded variable with a **link function**:
  - Transform **probability** into an **odds ratio**: $\frac{p}{1-p}=\frac{\mathrm{prob(y=1)}}{\mathrm{prob(y=0)}}$
    - If $p(\mathrm{y_i}=1)=.7$ then $\mathrm{Odds}(1)=2.33$; $\mathrm{Odds}(0)=0.429$
    - But odds scale is skewed, asymmetric, and ranges from 0 to +∞ → Not helpful
  - **Take *natural log of odds ratio* → called "logit" link:** $\boldsymbol{\mathrm{Log}\left[\frac{p}{1-p}\right]}$
    - If $p(\mathrm{y_i}=1)=.7$, then $\mathrm{Logit}(1)=0.846$; $\mathrm{Logit}(0)=-0.846$
    - Logit scale is now symmetric about 0, range is ±∞ → DING



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?

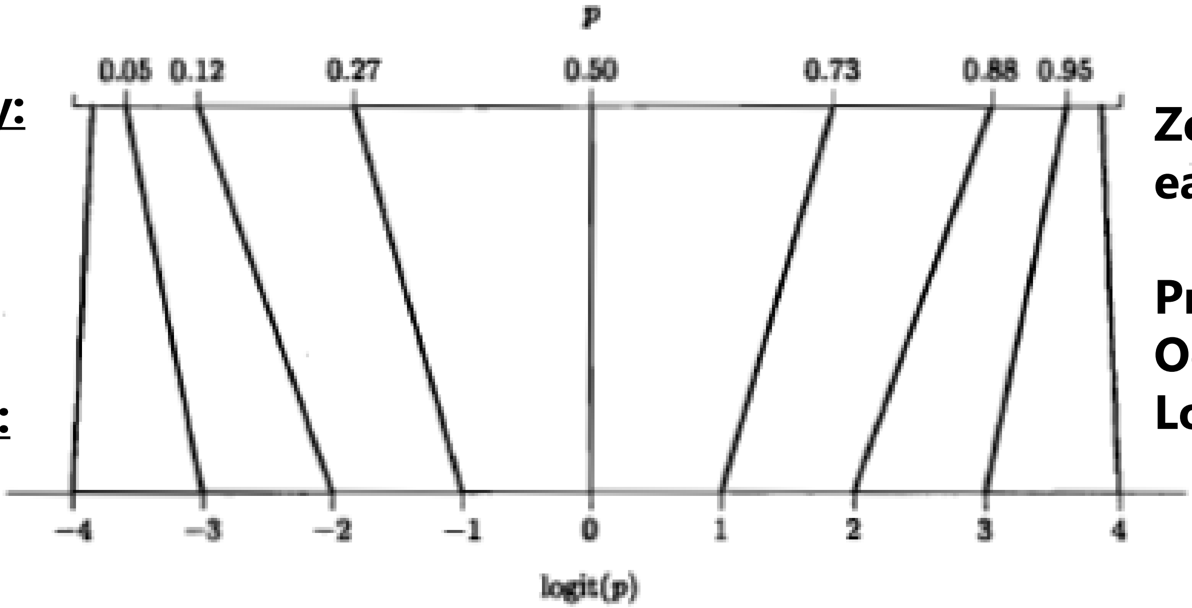
Solution #1: Probability into Logits

- **A Logit link is a nonlinear transformation of probability:**
  - Equal intervals in logits are NOT equal intervals of probability
  - The logit goes from ±∞ and is symmetric about 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 conditional mean outcome shuts off at 0 or 1 as needed**

Probability: $p(y_i = 1)$

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

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

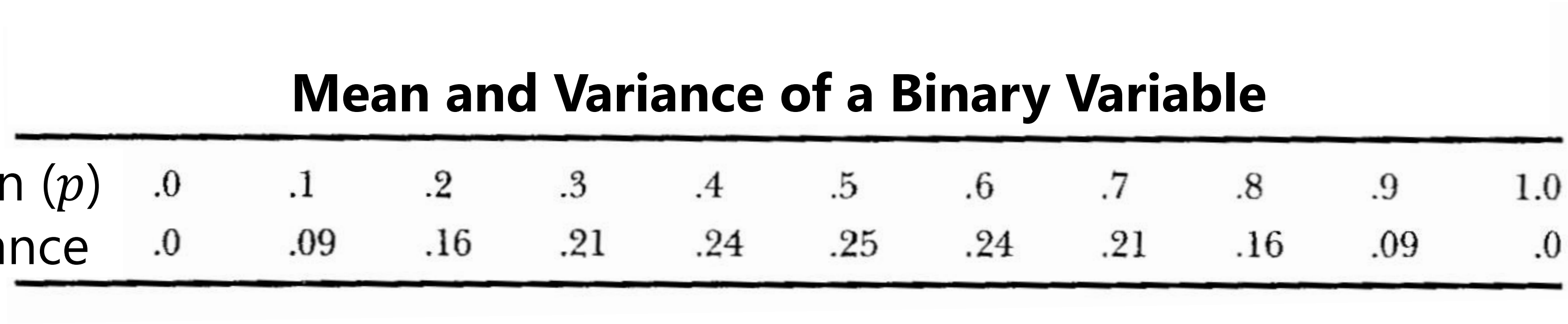


Zero-point on each scale:

Prob = .5
Odds = 1
Logit = 0

Normal GLM for Binary Outcomes?

- General linear model: $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1}) = \boldsymbol{\beta}_\mathbf{0} + \boldsymbol{\beta}_\mathbf{1}\mathbf{X_i} + \boldsymbol{\beta}_\mathbf{2}\mathbf{Z_i} + \mathbf{e_i}$
- If $\mathbf{y_i}$ is binary, then $\mathbf{e_i}$ can only be 2 things: $\mathbf{e_i} = \mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}$
  - If $\mathbf{y_i} = 0$ then $\mathbf{e_i}$ = (0 − predicted probability)
  - If $\mathbf{y_i} = 1$ then $\mathbf{e_i}$= (1 − predicted probability)
- Problem #2a: So the residuals can't be normally distributed
- Problem #2b: The residual variance can't be constant over X as in GLM because the **mean and variance are dependent**
  - Variance of binary variable: $\mathbf{Var}(\mathbf{y_i}) = \boldsymbol{p} * (\mathbf{1} - \boldsymbol{p})$



Mean and Variance of a Binary Variable

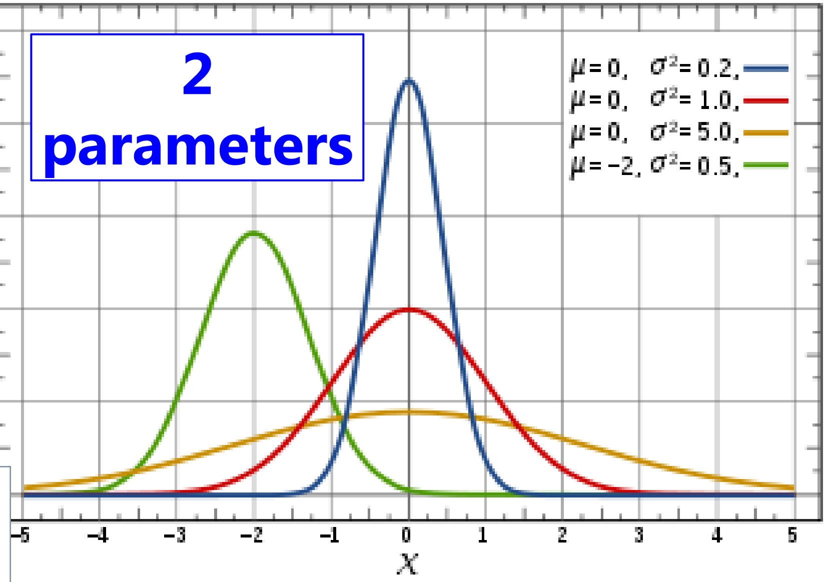
Mean (p)	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
Variance	.0	.09	.16	.21	.24	.25	.24	.21	.16	.09	.0

Solution to #2: Bernoulli Distribution

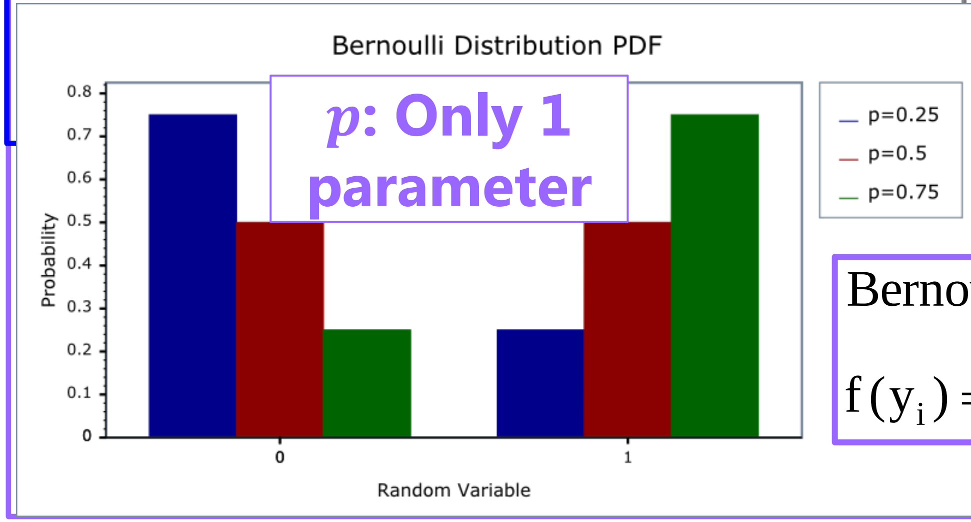
- Rather than using a **normal conditional distribution** for the outcome, we will use a **Bernoulli conditional distribution** → a special case of a binomial distribution for only one binary outcome

Univariate Normal PDF:

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



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

Predicted Binary Outcomes

- **Logit:** $\mathbf{Log}\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = \beta_0 + \beta_1 X_i + \beta_2 Z_i \leftarrow \boxed{g(\cdot)\ \textbf{link}}$
  - Predictor effects are linear and additive like in GLM, but β = change in **logit** per one-unit change in predictor

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

  or $\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = \exp(\beta_0 + \beta_1 X_i + \beta_2 Z_i\ )$

- **Probability:** $p(y_i = 1) = \frac{\exp(\beta_0+\beta_1 X_i+\beta_2 Z_i)}{1+\exp(\beta_0+\beta_1 X_i+\beta_2 Z_i)} \leftarrow \boxed{g^{-1}(\cdot)\ \textbf{inverse link}}$

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

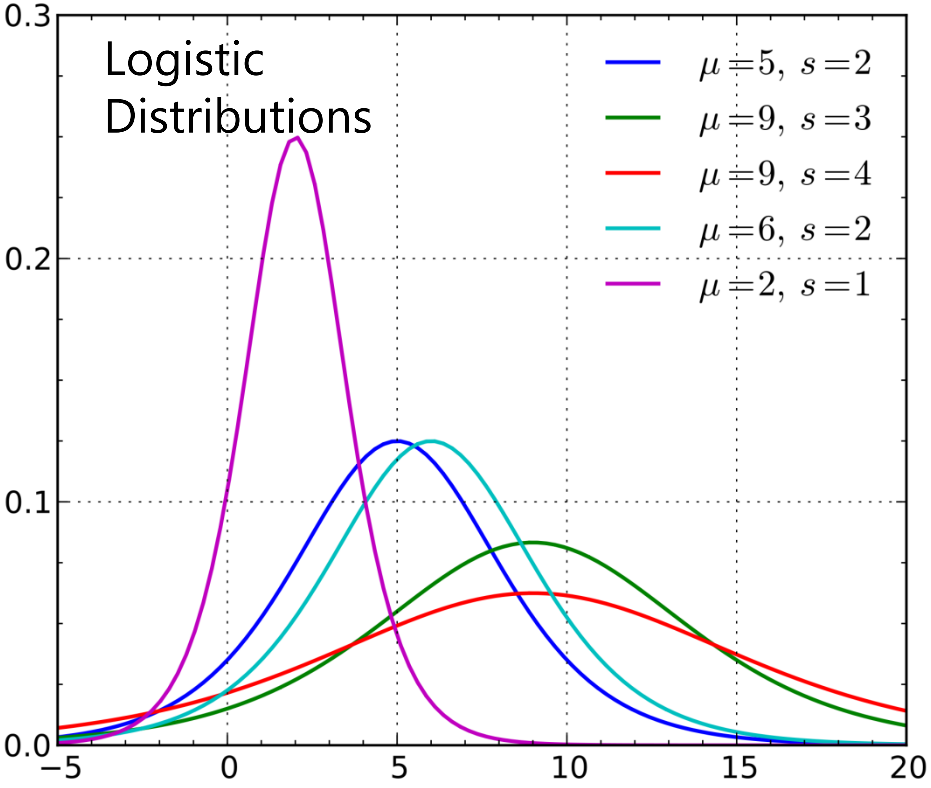
“Logistic Regression” for Binary Data

- This model is sometimes expressed by calling the logit(y_i) a underlying continuous (“latent”) response of $\mathbf{y_i^*}$ instead:

$$\mathbf{y_i^*} = \boldsymbol{\mathit{threshold}} + \textbf{your model} + \mathbf{e_i}$$

$threshold = \beta_0 * -1$ is given in Mplus, not intercept

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



So **if predicting $\mathbf{y_i^*}$** instead, then $e_i \sim \text{Logistic}(0, \sigma_e^2 = 3.29)$

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)

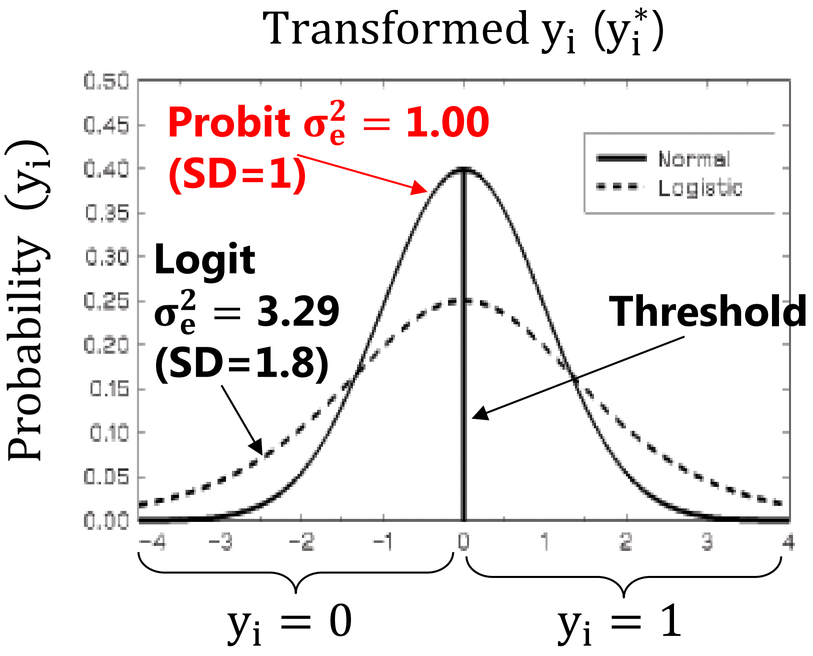
Other Models for Binary Data

- The idea that a “latent” continuous variable underlies an observed binary response also appears in a **Probit Regression** model:
  - A ***probit*** link, such that now your model predicts a different transformed Y_p:

    $$\mathrm{Probit}(\mathrm{y_i}=1)=\Phi^{-1}[p(\mathrm{y_i}=1)]=\mathit{your\ model} \longleftarrow \boxed{\mathbf{g}(\cdot)}$$

    - Where Φ = standard normal cumulative distribution function, so the transformed $\mathrm{y_i}$ is the **z-score** that corresponds to the value of standard normal curve **below** which conditional mean probability is found (requires integration to inverse link)
  - Same Bernoulli distribution for the conditional binary outcomes, in which residual variance cannot be separately estimated (so no $\mathrm{e_i}$ in the model)
    - Probit also predicts “latent” response: $\mathrm{y_i^*} = \mathrm{threshold} + \ \mathrm{your\ model} + \mathrm{e_i}$
    - But Probit says $\mathrm{e_i} \sim \mathrm{Normal}(0, \sigma_e^2 = 1.00)$, whereas $\mathrm{Logit}\ \sigma_e^2 = \frac{\pi^2}{3} = 3.29$
  - So given this difference in variance, probit estimates are on a different scale than logit estimates, and so their estimates won’t match… however…

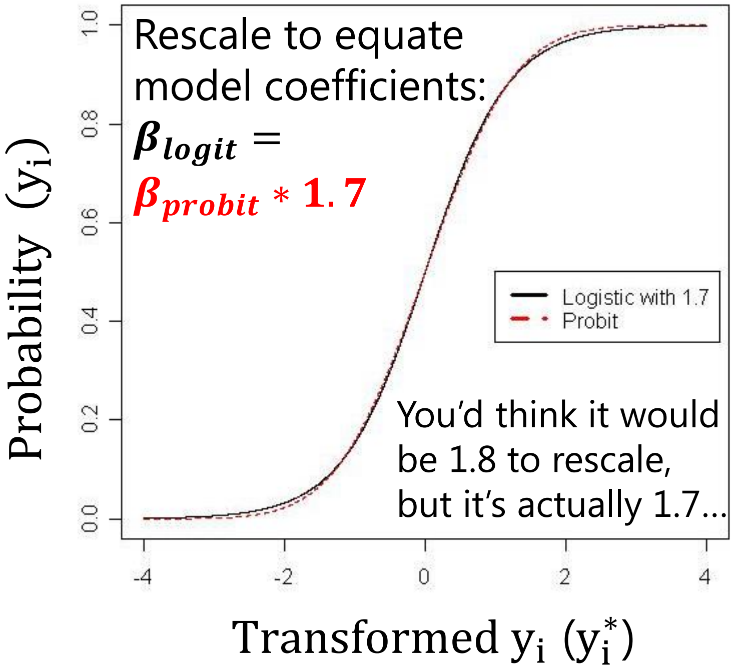
Probit vs. Logit: Should you care? Pry not.



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

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

Threshold



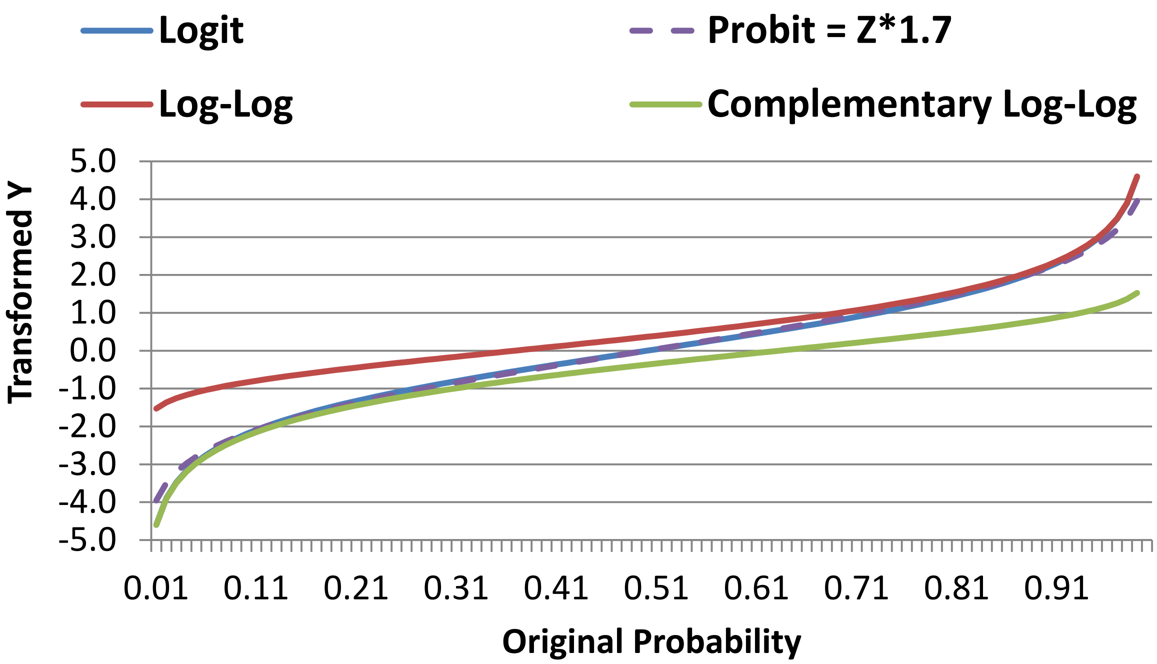
Rescale to equate model coefficients: $\boldsymbol{\beta_{logit}} =$ $\boldsymbol{\beta_{probit} * 1.7}$

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 based on odds)

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

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

μ = model	Logit	Probit	Log-Log	Complement. Log-Log
g(·) link	$\text{Log}\left(\frac{p}{1-p}\right) = \mu$	$\Phi^{-1}(p) = \mu$	$-\text{Log}[-\text{Log}(p)] = \mu$	$\text{Log}[-\text{Log}(1-p)] = \mu$
$g^{-1}(\cdot)$ inverse link (go back to probability):	$p = \frac{\exp(\mu)}{1+\exp(\mu)}$	$p = \Phi^{-1}(\mu)$	$p = \exp[-\exp(-\mu)]$ $e_i \sim \text{log} - \text{Weibull extreme value}\left(0.577, \sigma_e^2 = \frac{\pi^2}{6}\right)$	$p = 1 - \exp[-\exp(\mu)]$
In SAS LINK=	LOGIT	PROBIT	LOGLOG	CLOGLOG

Too Logit to Quit: Predicting Proportions

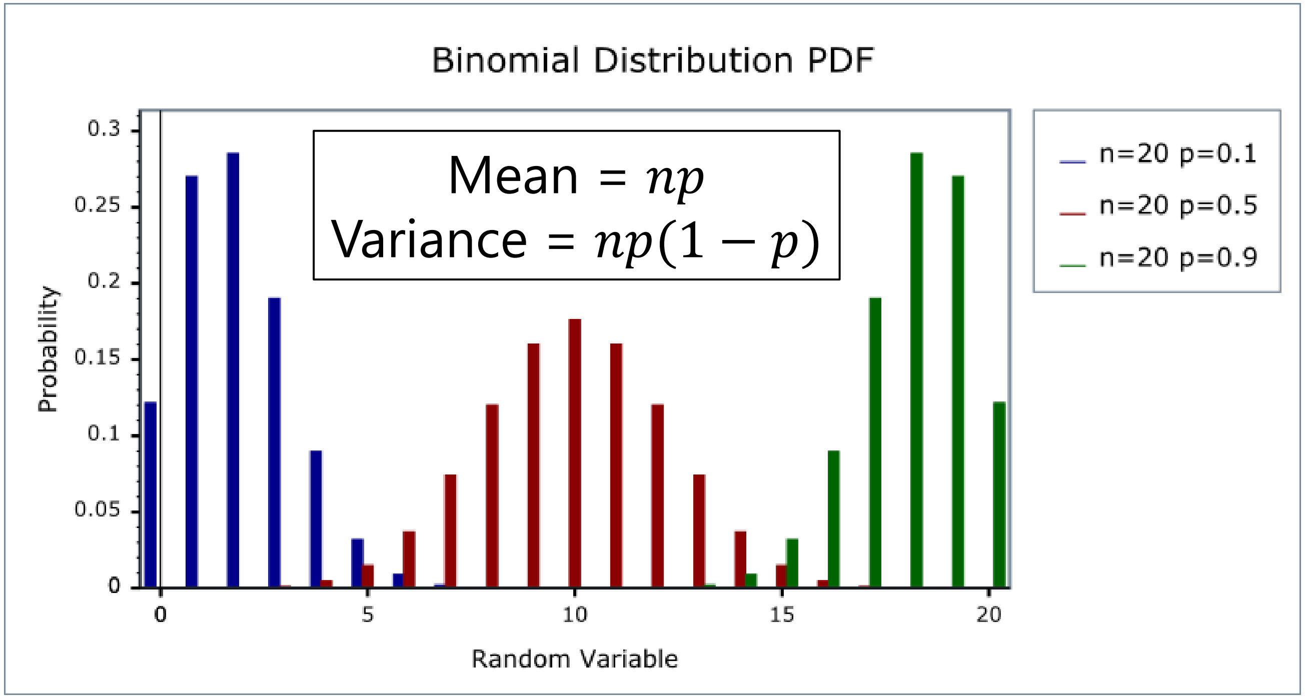
- The logit link can also be useful in predicting proportions:
  - Range between 0 and 1, so model needs to “shut off” predictions for conditional mean as they approach those ends, just as in binary data
  - Data to model: → $\mu \text{ in logits} = \text{Log}\left(\frac{p}{1-p}\right) \longleftarrow \boxed{\mathbf{g(\cdot)\ Link}}$
  - Model to data → $p = \frac{\exp(\mu)}{1+\exp(\mu)} \longleftarrow \boxed{\mathbf{g^{-1}(\cdot)\ Inverse\text{-}Link}}$

- However, because the outcome values aren’t just 0 or 1, a Bernoulli conditional distribution won’t work for proportions
- Two distributions: **Binomial** (discrete) vs. **Beta** (continuous)
  - Binomial: Less flexible (just one hump), but can include 0 and 1 values
  - Beta: Way more flexible (????), but cannot directly include 0 or 1 values
    - There are “zero-inflated” and/or “one-inflated” versions for these cases

Binomial Distribution for Proportions

- The discrete **binomial** distribution can be used to predict $\boldsymbol{c}$ correct responses given $\boldsymbol{n}$ trials
  - Bernoulli for binary = special case of binomial when n=1
  - $Prob(y = c) \; = \frac{n!}{c!(n-c)!} p^c (1-p)^{n-c}$

p = probability of 1



Mean = np
Variance = $np(1-p)$

As p gets closer to .5 and n gets larger, the binomial pdf will look more like a normal distribution.

But if many people show floor/ceiling effects, a normal distribution is not likely to work well… so use a binomial!

Binomial Distribution for Proportions

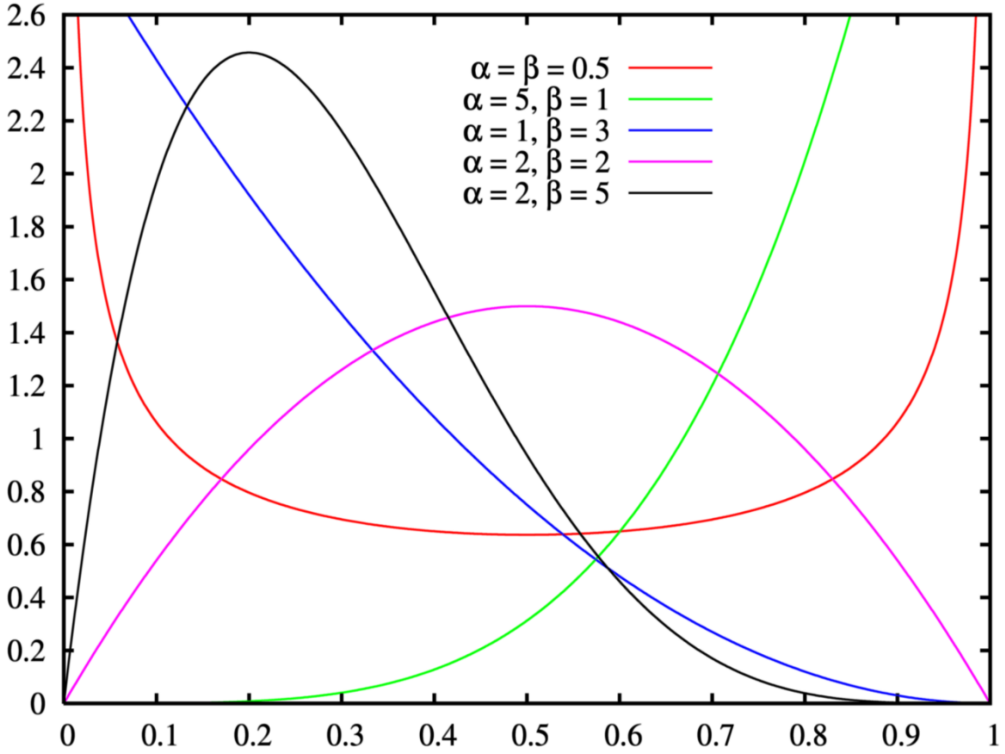
- SAS PROC GLIMMIX allows the outcome variable to be defined as ***#events/#trials*** on MODEL statement
  - LINK=LOGIT so that the conditional mean stays bounded between 0 and 1 as needed (or alternatively, CLOGLOG/LOGLOG)
  - DIST=BINOMIAL so variance (and SEs) are determined by that mean, as they should be assuming independent events
- STATA MELOGIT does the same with this option after ||:
  - Binomial(*VarforNtrials*); outcome then has number of events

- Be careful of **overdispersion**
  - Overdispersion = more variability than the mean would predict (cannot happen in binary outcomes, but it can for binomial)
  - Indicated by Pearson $\chi^2/\mathrm{df} > 1$ in SAS output
  - Can be caused by an improperly specified linear predictor model (e.g., forgot some interaction terms) or correlated observations (i.e., due to nesting, clustering, multivariate, and so forth)

Beta Distribution for Proportions

- The continuous **beta** distribution (LINK=LOGIT, DIST=BETA) can predict percentage correct $\boldsymbol{p}$ (must be $0 < \boldsymbol{p} < 1$)
  - $F(y|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \mathrm{y}^{\alpha-1}\,(1-\mathrm{y})^{\beta-1}$

α and β are "shape" parameters (> 0)



Mean = $\mu = \frac{\alpha}{\alpha+\beta}$

"Scale" = $\phi = \alpha + \beta$

Variance = $\frac{\mu(1-\mu)}{1+\phi}$

SAS GLIMMIX will provide a fixed intercept as logit(μ) and the "scale" ϕ

Beta Distribution for Proportions

- STATA does not appear to have a mixed effects version…?
- SAS NLMIXED can be used for beta mixed models

- The beta distribution is extremely flexible (i.e., can take on many shapes), but outcomes must be $0 < \boldsymbol{y} < 1$
  - If have 0's, need to add "zero-inflation" factor: → predicts logit of 0, then beta after 0 in two simultaneous models
  - If have 1's, need to add "one-inflation" factor: → predicts beta, then logit of 1 via two simultaneous models
  - Need both inflation factors if you have 0s and 1s (3 models)
  - Can be used with outcomes that have other ranges of possible values if they are rescaled into 0 to 1

Too Logit to Quit… http://www.youtube.com/watch?v=Cdk1gwWH-Cg

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

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

- Types of categorical outcomes:
  - Definitely ordered categories: "**cumulative logit**"
  - Maybe ordered categories: "**adjacent category logit**" (not used much)
  - Definitely NOT ordered categories: "**generalized logit**"

Logit-Based Models for *C* Ordinal Categories

- Known as "**cumulative logit**" or "**proportional odds**" model in generalized models; known as "graded response model" in IRT
  - LINK=CLOGIT, (DIST=MULT) in SAS GLIMMIX; MEOLOGIT or MEGLM in STATA
- Models the probability of **lower vs. higher** cumulative categories via $C - 1$ submodels (e.g., if $C = 4$ possible responses of $c = 0,1,2,3$):

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

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

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

Logit-Based 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 $c = 0,1,2,3$):

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

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

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

- In SAS, what the binary submodels predict depends on whether the model is predicting **DOWN ($y_i = 0$, the default)** or **UP ($y_i = 1$) cumulatively**
  - 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-Based Models for *C* Ordinal Categories

- Ordinal models usually use a logit link transformation, but they can also use cumulative log-log or cumulative complementary log-log links
  - LINK= CUMLOGLOG or CUMCLL in SAS GLIMMIX; CLOGLOG link in MEGLM in STATA

- Almost always assume **proportional odds**, that effects of predictors are the same across binary submodels—for example (subscripts = parm, submodel)
  - Submodel 1: $\text{Logit}[p(\text{y}_\text{i} > 0)] = \boldsymbol{\beta_{01}} + \beta_1\text{X}_\text{i} + \beta_2\text{Z}_\text{i} + \beta_3\text{X}_\text{i}\text{Z}_\text{i}$
  - Submodel 2: $\text{Logit}[p(\text{y}_\text{i} > 1)] = \boldsymbol{\beta_{02}} + \beta_1\text{X}_\text{i} + \beta_2\text{Z}_\text{i} + \beta_3\text{X}_\text{i}\text{Z}_\text{i}$
  - Submodel 3: $\text{Logit}[p(\text{y}_\text{i} > 2)] = \boldsymbol{\beta_{03}} + \beta_1\text{X}_\text{i} + \beta_2\text{Z}_\text{i} + \beta_3\text{X}_\text{i}\text{Z}_\text{i}$

- Proportional odds essentially means no interaction between submodel and predictor effects, which greatly reduces the number of estimated parameters
  - Despite the importance of this assumption, there appears to be no way to test it directly in most software packages for mixed effects models (except SAS NLMIXED)
  - 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, whose PDF for $C = 4$ categories of $c = 0{,}1{,}2{,}3$, an observed $y_i = c$, and indicators I if $c = y_i$

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

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

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

- Other models for categorical data that use the multinomial:
  - Adjacent category logit (partial credit): Models the 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): Models the probability of **reference vs. other** category via $C - 1$ submodels (e.g., if $C = 4$ and $0 = \text{ref}$):

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

In **nominal** models, all parameters are estimated **separately** per submodel

Generalized MLM: Intermediate Summary

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

- Generalized models have three parts:
  1. Non-normal conditional outcome distribution
  2. Link function: how bounded conditional mean of Y 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

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 longitudinal data:
  - Piles of variance will appear to be ADDED TO, not EXTRACTED FROM, the original residual variance when fixed to a known value (e.g., 3.29), 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

Empty Multilevel Model for Binary Outcomes

- **Level 1:** **Logit [*p*(y$_{ti}$ = 1)] = $\boldsymbol{\beta_{0i}}$**
- **Level 2:** $\boldsymbol{\beta_{0i} = \gamma_{00} + U_{0i}}$
- **Composite:** **Logit [*p*(y$_{ti}$ = 1)] = $\boldsymbol{\gamma_{00} + U_{0i}}$**

Notice what's NOT in level 1…

- σ_e^2 residual variance is not estimated → $\boldsymbol{\pi^2/3 = 3.29}$
  - (Known) residual is in model for actual y_{ti}, so σ_e^2 = 3.29 is for logit(y_{ti})

- $\text{Logistic ICC} = \frac{\text{BP}}{\text{BP+WP}} = \frac{\boldsymbol{\tau_{U_0}^2}}{\boldsymbol{\tau_{U_0}^2 + \sigma_e^2}} = \frac{\boldsymbol{\tau_{U_0}^2}}{\boldsymbol{\tau_{U_0}^2 + 3.29}}$

- Can do -2ΔLL test to see if $\boldsymbol{\tau_{U_0}^2} > 0$, although the ICC is problematic to interpret due to non-constant and not estimated residual variance
- ICC formulas for other outcomes besides binary vary widely
  - Some replace residual variance with 1; others use a function of the mean when the variance is dependent (e.g., Poisson) – I'm still not sure which is more correct

Example Random Linear Time Model for Binary Outcomes

- **Level 1:** $\textbf{Logit}\ [p(y_{ti} = 1)] = \beta_{0i} + \beta_{1i}(\text{time}_{ti})$
- **Level 2:** $\beta_{0i} = \gamma_{00} + U_{0i}$
  $\beta_{1i} = \gamma_{10} + U_{1i}$
- **Combined:**
  $\textbf{Logit}\ [p(y_{ti} = 1)] = (\gamma_{00} + U_{0i}) + (\gamma_{10} + U_{1i})(\text{time}_{ti})$

- σ_e^2 residual variance is still not estimated → $\boldsymbol{\pi^2/3 = 3.29}$
- Can test new fixed OR random effects with −2ΔLL tests when using true ML estimation (or use univariate or multivariate Wald test *p*-values for fixed effects as usual)

Example Random Linear Time Model for **Ordinal** Outcomes $(y_{ti} = 0, 1, \text{or } 2)$

- **L1:** $\text{Logit}\,[p(y_{ti} > 0)] = \beta_{0i1} + \beta_{1i1}(\text{time}_{ti})$

  $\text{Logit}\,[p(y_{ti} > 1)] = \beta_{0i2} + \beta_{1i2}(\text{time}_{ti})$

- **L2:** $\beta_{0i1} = \gamma_{001} + U_{0i1} \quad \beta_{1i1} = \gamma_{101} + U_{1i1}$

  $\beta_{0i2} = \gamma_{002} + U_{0i2} \quad \beta_{1i2} = \gamma_{102} + U_{1i2}$

- Assumes proportional odds → $\gamma_{001} \neq \gamma_{002}$ and $\gamma_{101} = \gamma_{102}$ and $U_{0i1} = U_{0i2}$ and $U_{1i1} = U_{1i2}$
  - Testable via nominal model (all unequal) or using NLMIXED to write a custom model in which some can be constrained
  - σ_e^2 residual variance is still not estimated → **$\pi^2/3 = 3.29$** (if link=logit)

New Interpretation of Fixed Effects

- In general linear mixed models, the fixed effects are interpreted as the "average" effect for the sample
  - γ_{00} is "sample average" intercept
  - U_{0i} is "individual deviation from sample average"

- What "average" means in generali*zed* linear mixed models is different, because of the use of nonlinear link functions:
  - e.g., the mean of the logs ≠ log of the means
  - Therefore, the fixed effects are not the "sample average" effect, they are the effect for ***specifically for*** $U_i = 0$
    - So fixed effects are *conditional* on the random effects
    - This gets called a "**unit-specific**" or "**subject-specific**" model
    - This distinction does not exist when using a normal conditional distribution

Comparing Results across Models

- NEW RULE: Coefficients cannot be compared directly across models, because they are not on the same scale! (Bauer, 2009)
- e.g., if residual variance = 3.29 in binary 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

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

  - **Level-1 predictors cannot decrease the level-1 variance** like usual, so all other model estimates have to increase to compensate
    - If X_{ti} is uncorrelated with other X's and is a pure level-1 variable (ICC ≈ 0), then fixed and SD(U_{0i}) will increase by same factor
  - **Random effects variances can decrease**, though, so level-2 effects should be on the same scale across models if level-1 model is the same

A Little Bit about Estimation

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

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

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 −2ΔLL TESTS
    - Modern software may also add a Laplace approximation to QL, which then does permit −2ΔLL tests (also in SAS GLIMMIX and STATA melogit)

Marginal Maximum Likelihood Estimation

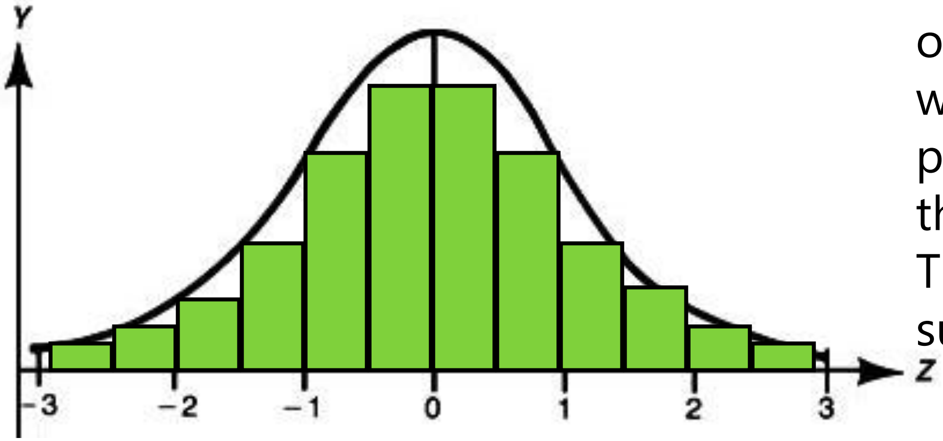
- **ML via Numeric(al) Integration** → gold standard
  - Synonyms: (adaptive) Gaussian quadrature
  - Provides much better estimates and valid −2ΔLL tests (ML flavor only)
  - Can take forever or not converge at all in models with many random effects; not available for models with crossed random effects
    - "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
  - So no such thing as an **R** matrix structure possible, so any differences in variance or additional sources of covariance must be specified 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
  - Multivariate outcomes can have different links and distributions in SAS GLIMMIX using LINK=BYOBS and DIST=BYOBS (Save new variables called "link" and "dist" to your data to tell GLIMMIX what to use for each)

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 gives link-predicted outcome given parameter estimates, but the U's themselves are not parameters—their variances and covariances are instead
  - But so long as we can assume the **U**'s are MVN, we can still proceed…
  - Computing the likelihood for each set of possible parameters requires *removing* the contribution of the individual **U** values from the model equation—by ***integrating*** across possible **U** 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 you have the right answers, which occurs when the log-likelihood changes very little across iterations (i.e., it converges)
- **Step 4:** If you aren't converged, choose new parameters values
  - Newton-Rhapson or Fisher Scoring (calculus), EM algorithm (U's =missing data)

ML via Numerical Integration

- More on Step 2: Divide the U distribution into rectangles
  - → "Gaussian Quadrature" (# rectangles = # "quadrature points")
  - First divide the whole U 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 (SAS will pick how many)



The likelihood of each level-2 unit's outcomes at each **U** 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 Linear Time, Random Intercept Model

1. Start with values for fixed effects: intercept: γ_{00} = 0.5, time: γ_{10} = 1.5,
2. Compute likelihood for real data based on fixed effects and plausible U_{0i} (-2,0,2) using model: Logit(y_{ti}=1) = γ_{00} + γ_{10}(time$_{ti}$) + U_{0i}
   - Here for one person at two occasions with y_{ti}=1 at both occasions

			IF y_{ti}=1	**IF y_{ti}=0**	**Likelihood**	**Theta**	**Theta**	**Product**
	U_{0i} = -2	**Logit(y_{ti})**	**Prob**	**1-Prob**	**if both y=1**	**prob**	**width**	**per Theta**
Time 0	0.5 + 1.5(0) **- 2**	-1.5	0.18	0.82	0.091213	0.05	2	0.00912
Time 1	0.5 + 1.5(1) **- 2**	0.0	0.50	0.50				
	U_{0i} = 0	**Logit(y_{ti})**	**Prob**	**1-Prob**				
Time 0	0.5 + 1.5(0) **+ 0**	0.5	0.62	0.38	0.54826	0.40	2	0.43861
Time 1	0.5 + 1.5(1) **+ 0**	2.0	0.88	0.12				
	U_{0i} = 2	**Logit(y_{ti})**	**Prob**	**1-Prob**				
Time 0	0.5 + 1.5(0) **+ 2**	2.5	0.92	0.08	0.90752	0.05	2	0.09075
Time 1	0.5 + 1.5(1) **+ 2**	4.0	0.98	0.02				
Overall Likelihood (Sum of Products over All Thetas):								**0.53848**

(do this for each occasion, then multiply this whole thing over all people)

(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, log, log-log…)
  - ***Linear*** relationship between X's and ***transformed*** conditional mean outcome
  - ***Nonlinear*** relationship between X's and ***original*** conditional mean outcome
    - Conditional outcomes then follow some non-normal distribution
- In models for binary or categorical data, level-1 residual variance is fixed and varies with the conditional mean (smaller at bounds)
  - So it can't go decrease after being explained by level-1 predictors, which means that the scale of all model parameters has to 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 amounts
    - Binomial (2 options) or multinomial (3+ options) distributions
    - Question: Are the values ordered → which link?
  - **Count of things that happened**, so values < 0 cannot exist
    - Sample space goes from 0 to +∞
    - Poisson or Negative Binomial distributions (usually)
    - Log link (usually) so predicted outcomes can't go below 0
    - Question: Are there *extra* 0 values? What to do about them?

- **"Continuous" outcomes**—responses can be **any** number
  - Question: What does the residual distribution look like?
    - Normal-ish? Skewed? Cut off? Mixture of different distributions?

A Revised Taxonomy

- Rather than just separating into discrete vs. continuous, think about models based on their shape AND kinds of data they fit
  - Note: You can use continuous models for discrete data (that only have integers), but not discrete models for continuous data (non-integers)

1. Skewed-looking distributions
   - Discrete: Poisson, Generalized Poisson, Negative Binomial (NB)
   - Continuous: Log-Normal, Beta, Gamma

2. Skewed with a pile of 0's: Becomes **If 0** and **How Much**
   - These models will differ in how they define the "If 0" part
   - Discrete: Zero-Inflated Poisson or NB, Hurdle Poisson or NB
   - Continuous: "Two-Part" (with normal or lognormal for how much part)
     - Better: **Gamma** for the how much part because it only includes values > 0

Models for Count Outcomes

- Counts: non-negative integer unbounded responses
  - e.g., how many cigarettes did you smoke this week?
  - Traditionally uses natural log link so that predicted outcomes stay ≥ 0

- $\mathrm{g}(\bullet)$ $\quad \mathrm{Log}[\mathrm{E}(\mathrm{y_i})] = \mathrm{Log}(\mu_\mathrm{i}) = \mathrm{model}$ → predicts mean of $\mathrm{y_i}$
- $\mathrm{g}^{-1}(\bullet)$ $\mathrm{E}(\mathrm{y_i}) = \exp(\mathrm{model})$ → to un-log it, use $\exp(\mathrm{model})$

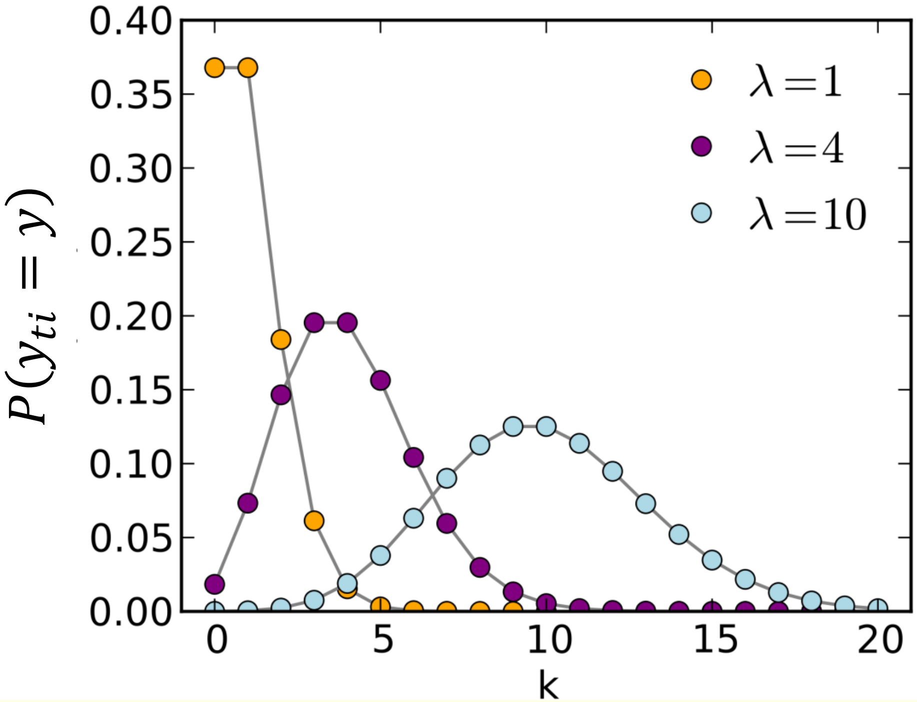
  - e.g., if $\mathrm{Log}(\mu_\mathrm{i}) = \mathrm{model}$ provides predicted $\mathrm{Log}(\mu_\mathrm{i}) = 1.098$,
    that translates to an actual predicted count of $\exp(1.098) = 3$
  - e.g., if $\mathrm{Log}(\mu_\mathrm{i}) = \mathrm{model}$ provides predicted $\mathrm{Log}(\mu_\mathrm{i}) = -5$,
    that translates to an actual predicted count of $\exp(-5) = 0.006738$

- So that's how linear model predicts μ_i, the conditional mean for $\mathrm{y_i}$, but what about the conditional (residual) variance?

Poisson Conditional Distribution

- Poisson distribution has one parameter, λ, which is both its mean and its variance (so λ = mean = variance in Poisson)
- $f(\mathrm{y_i}|\lambda) = \mathrm{Prob}(\mathrm{y_i} = \mathrm{y}) = \frac{\lambda^{\mathrm{y}} * \exp(-\lambda)}{\mathrm{y}!}$
- PDF: $\mathrm{Prob}(\mathrm{y_i} = \mathrm{y}|\beta_0, \beta_1, \beta_2) = \frac{\mu_i^{\mathrm{y}} * \exp(-\mu_i)}{\mathrm{y}!}$

y! is factorial of y



The dots indicate that only integer values are observed.

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

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

3 potential problems for Poisson…

- The standard Poisson distribution is rarely sufficient, though

- **Problem #1: When mean ≠ variance**
  - If variance < mean, this leads to "under-dispersion" (not that likely)
  - If variance > mean, this leads to "over-dispersion" (happens frequently)

- **Problem #2: When there are *no* 0 values**
  - Some 0 values are expected from count models, but in some contexts $y_i > 0$ always (but subtracting 1 won't fix it; need to adjust the model)

- **Problem #3: When there are *too many* 0 values**
  - Some 0 values are expected from the Poisson and Negative Binomial models already, but many times there are even more 0 values observed than that
  - To fix it, there are two main options, depending on what you do to the 0's

- Each of these problems requires a model adjustment to fix it…

Problem #1: Variance > mean = over-dispersion

- To fix it, we must add another parameter that allows the variance to exceed the mean… becomes a **Negative Binomial** distribution
  - Says residuals are a mixture of Poisson and gamma distributions, such that λ itself is a random variable with a gamma distribution
  - So expected mean is still given by λ, but the variance will differ from Poisson

- Model: $\mathrm{Log}[\mathrm{E}(\mathrm{y_i})] = \mathrm{Log}(\mu_\mathrm{i}) = \beta_0 + \beta_1 \mathrm{X_i} + \beta_2 \mathrm{Z_i} + \mathrm{e_i^G}$
- Negative Binomial PDF with a new k dispersion parameter is now:
  - $\mathrm{Prob}(\mathrm{y_i} = \mathrm{y} | \beta_0, \beta_1, \beta_2) = \frac{\Gamma\left(\mathrm{y}+\frac{1}{k}\right)}{\Gamma(\mathrm{y}+1) * \Gamma\left(\frac{1}{k}\right)} * \frac{(k\mu_\mathrm{i})^\mathrm{y}}{(1+k\mu_\mathrm{i})^{\mathrm{y}+\frac{1}{k}}}$
  - $\boldsymbol{k}$ is dispersion, such that $\mathrm{Var}(\mathrm{y_i}) = \mu_\mathrm{i} + \boldsymbol{k}\mu_\mathrm{i}^2$
  - Can test whether $k > 0$ via −2LL test, although LL for $k = 0$ is undefined

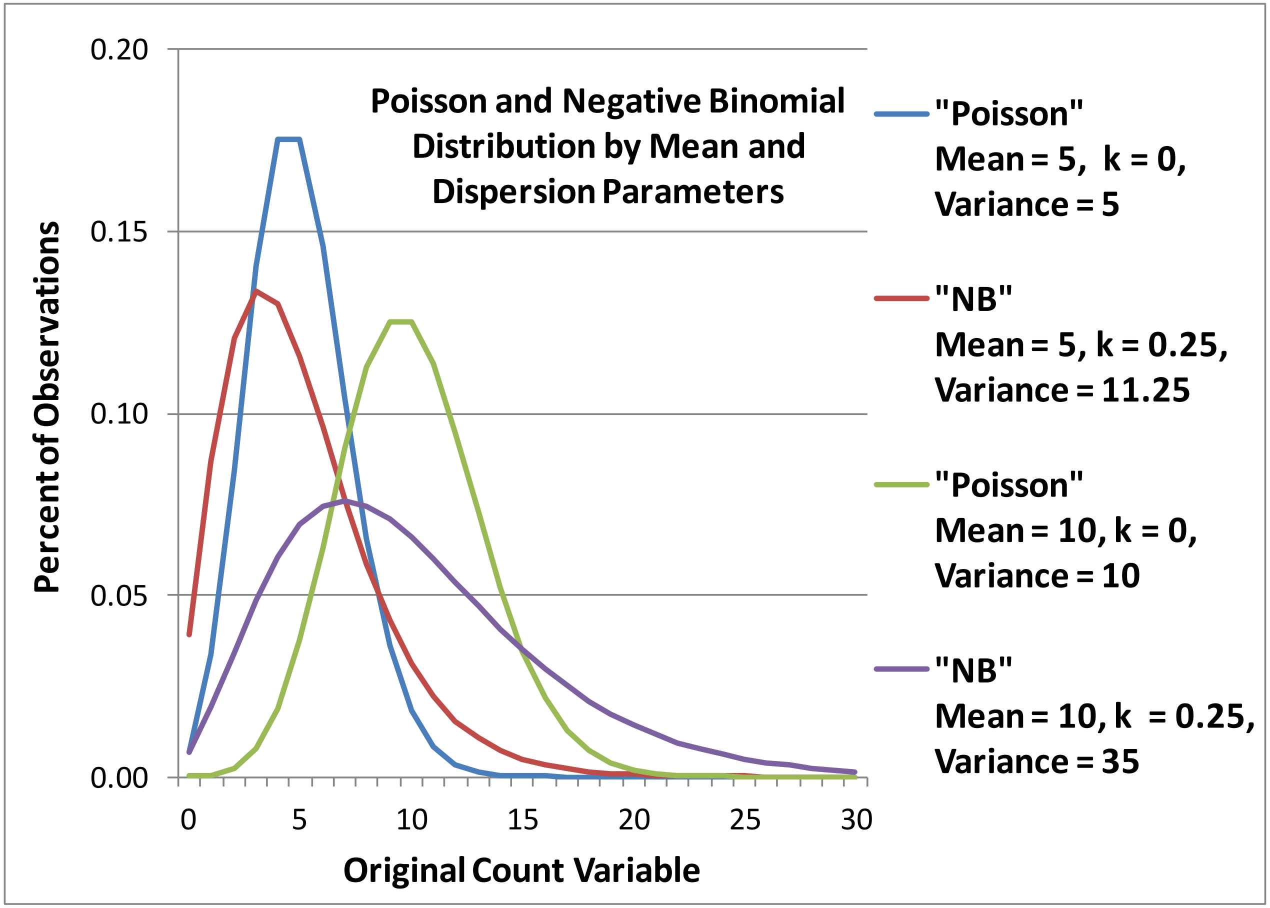
DIST = NEGBIN in SAS; **MENBREG** in STATA

So ≈ Poisson if $k = 0$

- An alternative model with the same idea is the **generalized Poisson**:
  - Mean: $\frac{\lambda}{1-k}$, Variance: $\frac{\mu}{(1-k)^2}$, that way LL is defined for $k = 0$
  - Is in SAS FMM (and in GLIMMIX via user-defined functions)

GPOISSON in STATA

Negative Binomial (NB) = "Stretchy" Poisson…



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

$$\text{Var}(y_i) = \lambda + k\lambda^2$$

A Negative Binomial model can be useful for count outcomes with extra skewness, but that otherwise follow a Poisson conditional distribution.

- 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 $p < .05$, keep NB

Problem #2: There are no 0 values

- "**Zero-Altered**" or "**Zero-Truncated**" Poisson or Negative Binomial: ZAP/ZANB or ZTP/ZTNB (used in hurdle models)
  - Is usual count distribution, just not allowing any 0 values
  - Single-level models are in SAS PROC FMM using DIST=TRUNCPOISSON for ZTP or DIST=TRUNCNEGBIN for ZTNB
  - Single-level TPOISSON (for ZTP) and TNBREG (for ZTNB) in STATA
  - Multivariate versions could be fitted in SAS NLMIXED or Mplus, too

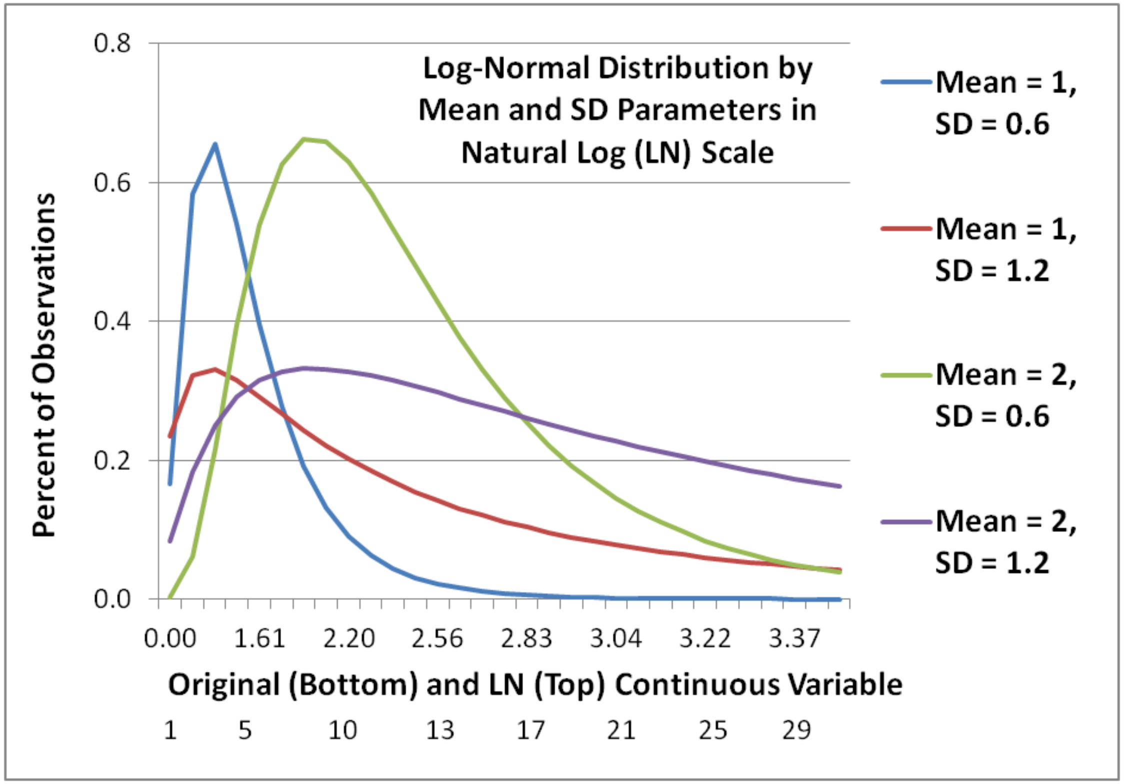
- Poisson PDF was: $\text{Prob}(y_i = y|\mu_i\ ) = \frac{\mu_i^y * \exp(-\mu_i)}{y!}$
- Zero-Truncated Poisson PDF is:
  - $\text{Prob}\big(y_i = y|\mu_{i,}y_i > 0\big) = \frac{\mu_i^y * \exp(-\mu_i)}{y![1-\exp(-\mu_i)]}$
  - $\text{Prob}(y_i = 0) = \exp(-\mu_i)$, so $\text{Prob}(y_i > 0) = 1 - \exp(-\mu_i)$
  - Divides by probability of non-0 outcomes so probability still sums to 1

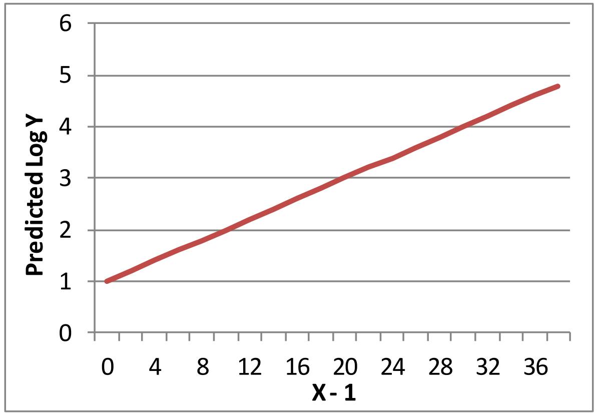
Software for Discrete Outcomes

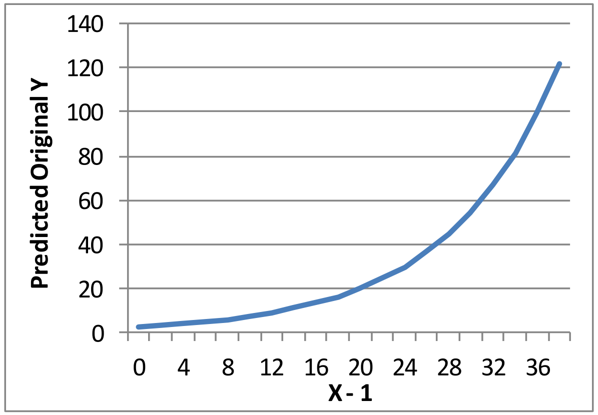
- There are many choices for modeling not-normal ***discrete*** outcomes (that include integer values only); most use either an identity or log link
- **Single-level, univariate generalized models in SAS:**
  - GENMOD: DIST= (and default link): Binomial (Logit), Poisson (Log), Zero-Inflated Poisson (Log), Negative Binomial (Log), Zero-Inflated Negative Binomial (Log)
  - FMM: DIST= (and default link): Binomial (Logit), Poisson (Log), Generalized Poisson (Log), Truncated Poisson (Log), Negative Binomial (Log), Uniform
- **Multilevel, multivariate generalized models in SAS through GLIMMIX:**
  - Binomial (Logit), Poisson (Log), Negative Binomial (Log)
  - BYOBS, which allows multivariate models by which you specify DV-specific link functions and distributions estimated simultaneously
  - User-defined variance functions for special cases (e.g., generalized Poisson)
- NLMIXED can also be used to fit any user-defined model

- **Up next: models for skewed continuous outcomes…**

Log-Normal Distribution (Link=Identity)

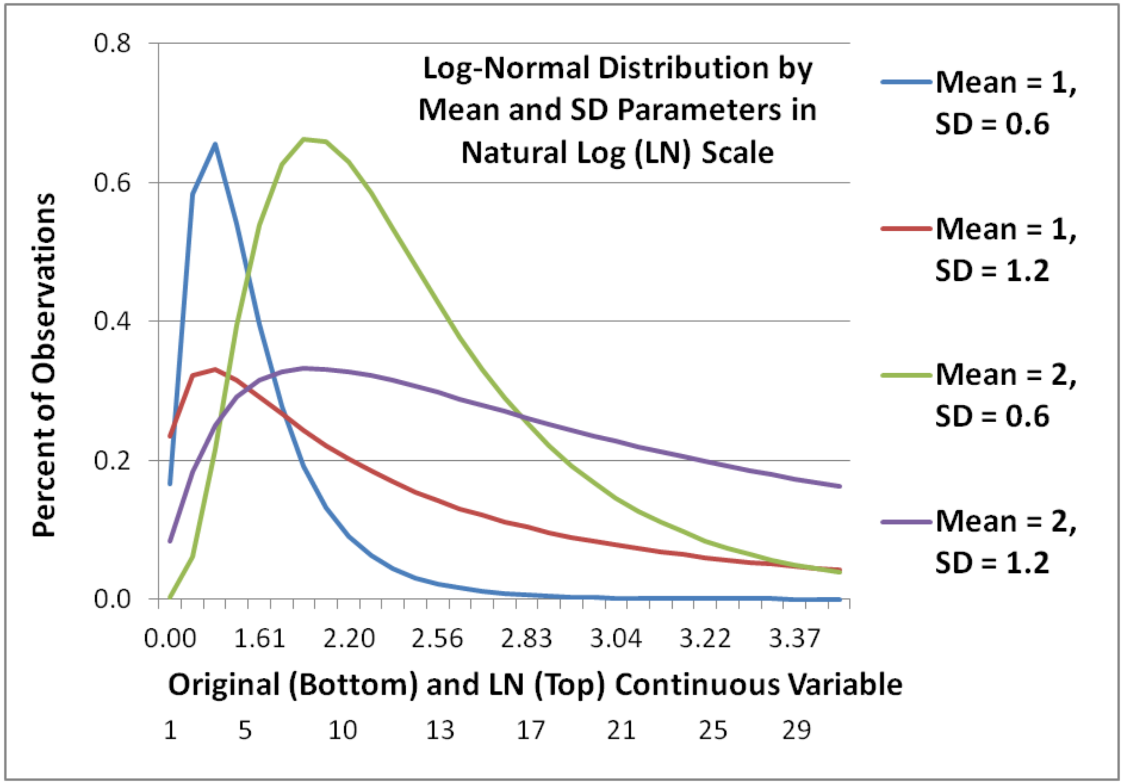


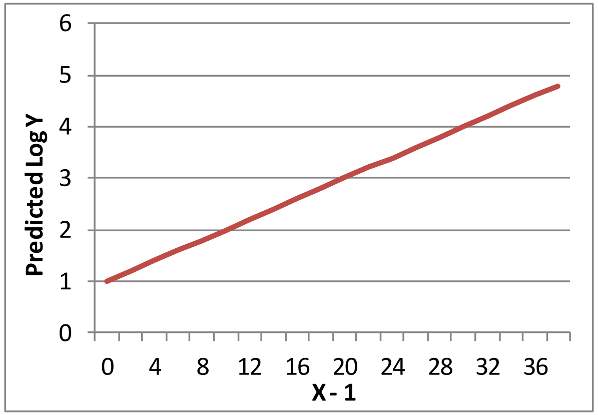


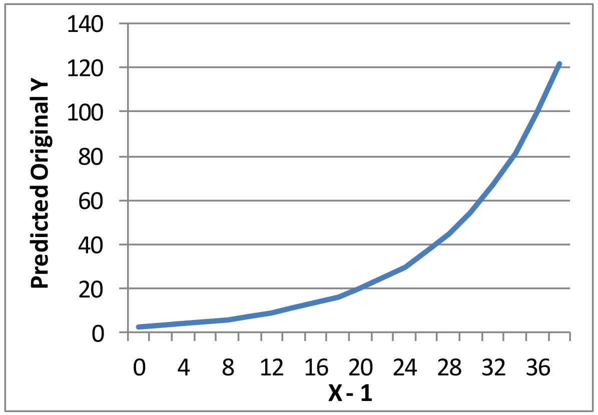


- $e_i \sim \text{LogNormal}(0, \sigma_e^2)$ → **log** of residuals is normal
  - Is same as log-transforming your outcome in this case…
  - The log link keeps the predicted values positive, but slopes then have an exponential (not linear) relation with original outcome

Log-Normal Distribution (Link=Identity)

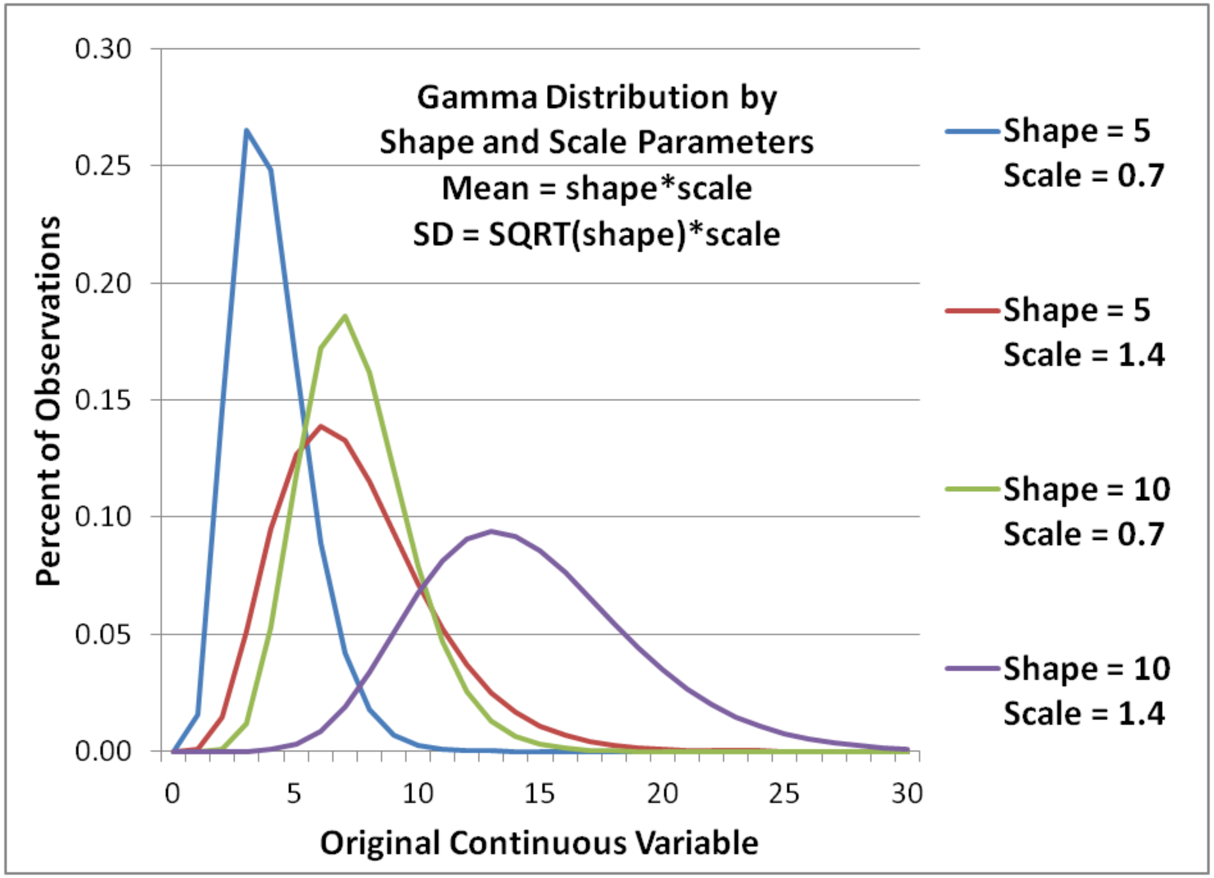


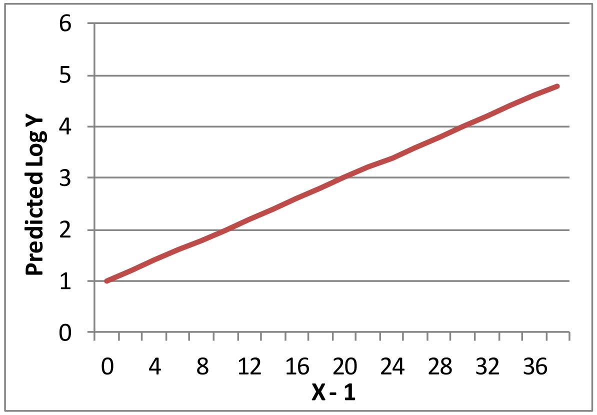


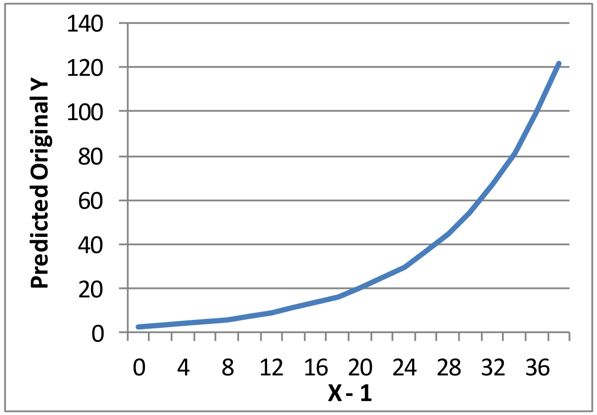


- GLIMMIX parameterization gives μ (= intercept) and $scale$ = (variance) to convert back into original data as follows:
  - $\text{Mean(Y)} = \exp(\mu) * \sqrt{\exp(scale)}$
  - $\text{Variance(Y)} = \exp(2\mu) * \exp(scale) * [\exp(scale) - 1]$

Gamma Response Distribution (y>0)







- GLIMMIX parameterization with LINK=LOG gives μ (= intercept) and $scale$ = (dispersion) to convert into original data as follows:
  - $\mathrm{Mean(Y)} = \exp(\mu) \approx (\mathrm{shape{*}scale})$
  - $\mathrm{Variance(Y)} = \exp(\mu)^2 * dispersion \approx (\mathrm{shape} * \mathrm{scale}^2)$

Software for Continuous Outcomes

- There are many choices for modeling not-normal ***continuous*** outcomes (that can include non-integer values); most use either an identity or log link
- **Single-level, univariate generalized models in SAS (not in Mplus):**
  - GENMOD: DIST= (and default link): Gamma (Inverse), Geometric (Log), Inverse Gaussian (Inverse^2), Normal (Identity)
  - FMM: DIST= (and default link): Beta (Logit), Betabinomial (Logit), Exponential (Log), Gamma (Log), Normal (Identity), Geometric (Log), Inverse Gaussian (Inverse^2), LogNormal (Identity), TCentral (Identity), Weibull (Log)
- **GLM in STATA** has gamma and inverse Gaussian distributions
- **Multilevel or multivariate generalized models in SAS via GLIMMIX:**
  - Beta (Logit), Exponential (Log), Gamma (Log), Geometric (Log), Inverse Gaussian (Inverse^2), Normal (Identity), LogNormal (Identity), TCentral (Identity)
  - BYOBS, which allows multivariate models by which you specify DV-specific link functions and distributions estimated simultaneously (e.g., two-part)
- NLMIXED can also be used to fit any user-defined model
- **Up next: models for zero-inflated discrete or continuous outcomes…**

Modeling Not-Normal Outcomes

- Previously we examined models for skewed distributions
  - Discrete: Poisson, Generalized Poisson, Negative Binomial (NB)
  - Continuous: Log-Normal, Gamma (also Beta from before)

- Now we will see additions to these models when the outcome also has a pile of 0's: Model becomes **If 0** and **How Much**
  - These models will differ in how they define the "If 0" part
  - Discrete → Zero-Inflated: Poisson, Generalized Poisson, or NB;
Hurdle: Poisson, Generalized Poisson, or NB
  - Continuous → Two-Part (with normal, lognormal, gamma for how much)
  - Many of these can be estimated directly in Mplus or SAS GLIMMIX, but some will need to be programmed in SAS GLIMMIX or NLMIXED
  - More options for single-level data in SAS PROC FMM and in STATA

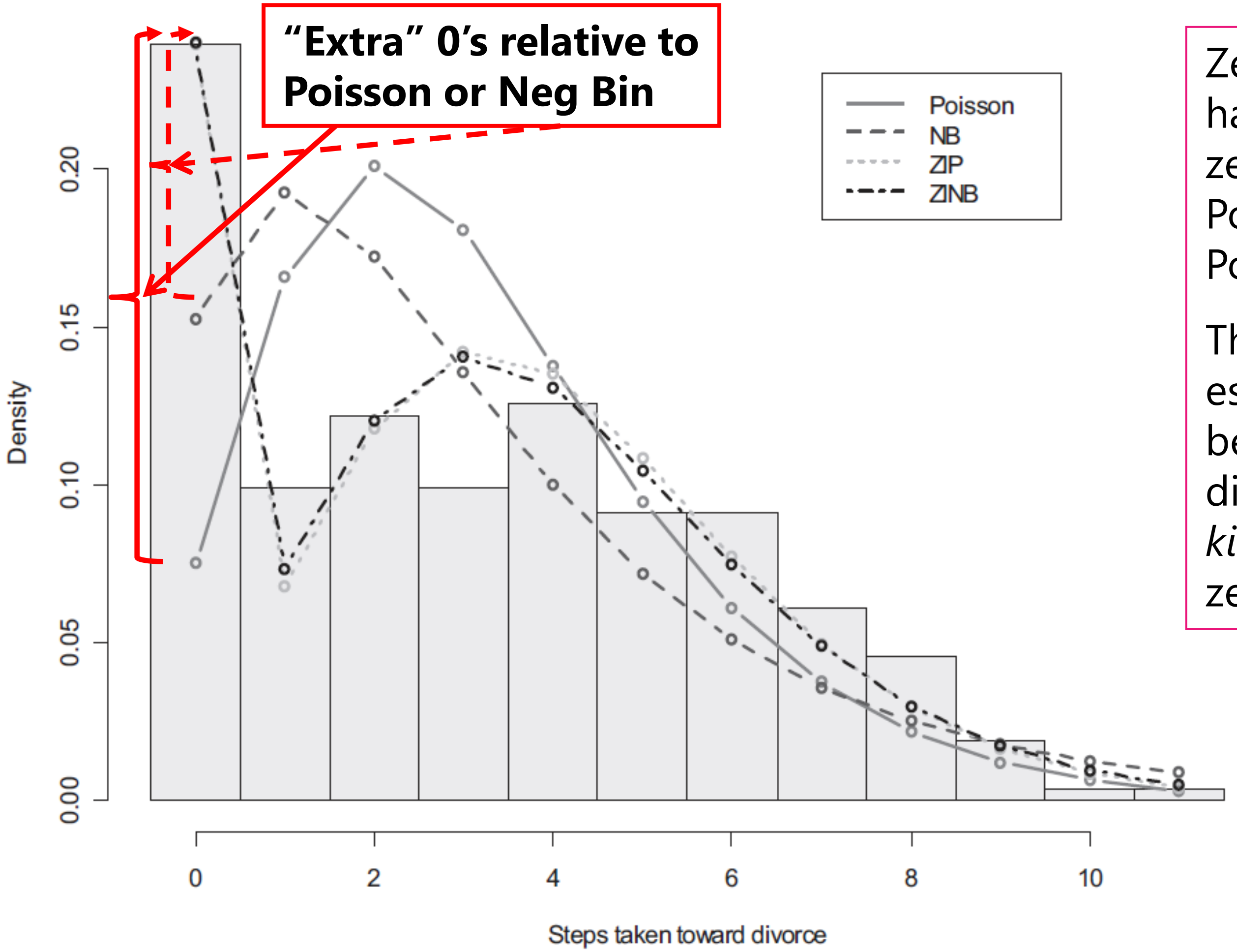
Problem #3: Too many 0 values, Option #1

- "**Zero-Inflated**" Poisson (DIST=ZIP) or NB(DIST=ZINB) in SAS GENMOD or Mplus; ZIP/ZI Generalized Poisson (ZIGP) in STATA
  - Distinguishes **two kinds of 0 values**: **expected/structural** and **inflated** ("structural") through a mixture of distributions (Bernoulli + Poisson/NB)
  - Creates two submodels to predict "if *extra* 0" and "if not, how much"?
    - Does not readily map onto most hypotheses (in my opinion)
    - But a ZIP example would look like this… (ZINB would add *k* dispersion, too)

- Submodel 1: $\mathrm{Logit}[p(\mathrm{y_i} = \mathrm{extra}\ 0)] = \beta_{01} + \beta_{11}\mathrm{X_i} + \beta_{21}\mathrm{Z_i}$
  - Predict being an extra 0 using Link = Logit, Distribution = Bernoulli
  - Don't have to specify predictors for this part, can simply allow an intercept (but need ZEROMODEL option to include predictors in SAS GENMOD)

- Submodel 2: $\mathrm{Log}[\mathrm{E}(\mathrm{y_i})] = \beta_{02} + \beta_{12}\mathrm{X_i} + \beta_{22}\mathrm{Z_i}$
  - Predict rest of counts <u>(including 0's)</u> using Link = Log, Distribution = Poisson

Example of Zero-Inflated Outcomes



"Extra" 0's relative to Poisson or Neg Bin

Figure 1. Histogram of Marital Status Inventory with predicted probabilities from regressions. NB = negative binomial; ZIP = zero-inflated Poisson; ZINB = zero-inflated negative binomial.

Zero-inflated distributions have extra "structural zeros" not expected from Poisson or NB ("stretched Poisson") distributions.

This can be tricky to estimate and interpret because the model distinguishes between *kinds of zeros* rather than zero or not...

Image borrowed from Atkins & Gallop, 2007

Problem #3: Too many 0 values, Option #1

- The Zero-Inflated models get put back together as follows:
  - ω_i is the predicted probability of being an extra 0, from:
  $$\omega_\mathrm{i} = \frac{\exp[\mathrm{Logit}[p(\mathrm{y_i} = \mathrm{extra}\ 0)]]}{1 + \exp[\mathrm{Logit}[p(\mathrm{y_i} = \mathrm{extra}\ 0)]]}$$
  - μ_i is the predicted count for the rest of the distribution, from:
  $$\mu_\mathrm{i} = \exp[\mathrm{Log}(\mathrm{y_i})]$$
  - ZIP: $\mathrm{Mean\ (original\ y_i)} = (1 - \omega_\mathrm{i})\mu_\mathrm{i}$
  - ZIP: $\mathrm{Variance(original\ y_i)} = \mu_\mathrm{i} + \frac{\omega_\mathrm{i}}{(1-\omega_\mathrm{i})}\mu_\mathrm{i}^2$
  - ZINB: $\mathrm{Mean\ (original\ y_i)} = (1 - \omega_\mathrm{i})\mu_\mathrm{i}$
  - ZINB: $\mathrm{Variance(original\ y_i)} = \mu_\mathrm{i} + \left[\frac{\omega_\mathrm{i}}{(1-\omega_\mathrm{i})} + \frac{\mathrm{k}}{1-\omega_\mathrm{i}}\right]\mu_\mathrm{i}^2$

Problem #3: Too many 0 values, Option #2

- "**Hurdle**" models for Poisson or Negative Binomial
  - PH or NBH: Explicitly **separates 0 from non-0 values** through a mixture of distributions (Bernoulli + Zero-Altered Poisson/NB)
  - Creates two submodels to predict "if any 0" and "if not 0, how much"?
    - Easier to think about in terms of prediction (in my opinion)

- Submodel 1: $\mathrm{Logit}[p(\mathrm{y_i} = 0)] = \beta_{01} + \beta_{11}\mathrm{X_i} + \beta_{21}\mathrm{Z_i}$
  - Predict being **<u>any 0</u>** using Link = Logit, Distribution = Bernoulli
  - Don't have to specify predictors for this part, can simply allow it to exist

- Submodel 2: $\mathrm{Log}[\mathrm{E}(\mathrm{y_i})|\mathrm{y_i} > 0] = \beta_{02} + \beta_{12}\mathrm{X_i} + \beta_{22}\mathrm{Z_i}$
  - Predict rest of **<u>positive counts</u>** using Link = Log, Distribution = ZAP/ZANB

- These models are not readily available in SAS, but NBH is in Mplus
  - Could be fit in SAS NLMIXED (as could ZIP/ZINB)
  - Can also split DV into each submodel and estimate separately (in STATA)

Two-Part Models for Continuous Outcomes

- A two-part model is an analog to hurdle models for zero-inflated count outcomes (and could be used with count outcomes, too)
  - Explicitly **separates 0 from non-0 values** through a mixture of distributions (Bernoulli + Normal or LogNormal or Gamma)
  - Creates two submodels to predict "if any not 0" and "if not 0, how much"?
    - Easier to think about in terms of prediction (in my opinion)

- Submodel 1: $\mathrm{Logit}[p(\mathrm{y_i} > 0)] = \beta_{01} + \beta_{11}\mathrm{X_i} + \beta_{21}\mathrm{Z_i}$
  - Predict being ***any not 0*** using Link = Logit, Distribution = Bernoulli
  - Usually do specify predictors for this part

- Submodel 2: $(\mathrm{y_i}|\mathrm{y_i} > 0) = \beta_{02} + \beta_{11}\mathrm{X_i} + \beta_{21}\mathrm{Z_i}$
  - Predict rest of ***positive amount*** using Link = Identity, Distribution = Normal or Log-Normal (often rest of distribution is skewed, so log works better)

- Two-part is in Mplus, but parts can be estimated separately in SAS/STATA
  - Logit of 0/1 for "if part" + log-transformed DV for "how much" part
  - Is related to "tobit" models for censored outcomes (for floor/ceiling effects)

Pile of 0's Taxonomy

- What kind of **amount** do you want to predict?
  - Discrete: Count → Poisson
  - Stretchy Count → Generalized Poisson or Negative Binomial
  - Continuous: Normal, Log-Normal, Gamma

- What kind of **If 0** do you want to predict?
  - Discrete: Extra "structural" 0 beyond predicted by amount?
    → Zero-inflated Poisson or Zero-inflated Negative Binomial
  - Discrete: Any 0 at all?
    → Hurdle Poisson or Hurdle Negative Binomial
  - Continuous: Any 0 at all?
    → Two-Part with Continuous Amount (see above)
  - Note: Given the same amount distribution, these alternative ways of predicting 0 will result in the same empty model fit

Comparing Generalized Models

- Whether or not a dispersion parameter is needed (to distinguish Poisson and NB) can be answered via a likelihood ratio test
  - For the most fair comparison, keep the linear predictor model the same

- Whether or not a zero-inflation model is needed should, in theory, also be answerable via a likelihood ratio test…
  - But people disagree about this
  - Problem? Zero-inflation probability can't be negative, so is bounded at 0
  - Other tests have been proposed (e.g., Vuong test—see SAS macro online)
  - Can always check AIC and BIC (smaller is better)

- In general, models with the same distribution and different links can be compared via AIC and BIC, but one cannot use AIC and BIC to compare across alternative distributions (e.g., normal or not?)
  - Log-Likelihoods are not on the same scale due to using different PDFs
  - You can compute predicted values under different models to see how reasonably they approximate the data for some unofficial guidance

Generalized MLM: Summary

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

- Multivariate and multilevel versions of all the generalized models we covered *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 (fewer R matrix structures in true ML)

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