Generalized Multilevel Models for Non-Normal Outcomes

- Topics:
  - 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 count outcomes
    - Models for not-normal but continuous outcomes

Dimensions for Organizing Models

- Outcome type: General (normal) vs. Generali*zed* (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)

Note: Least Squares is only for GLM

- **Generali*zed* Linear Models:** **any conditional outcome distribution**, **fixed** effects through **link functions**, no random effects (one dimension)
- **General Linear Mixed Models:** conditionally normal outcome distribution, **fixed and random effects** (identity link, but multiple sampling dimensions)
- **Generali*zed* 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 of DV in a linear combination: (effect*predictor) + (effect*predictor)…

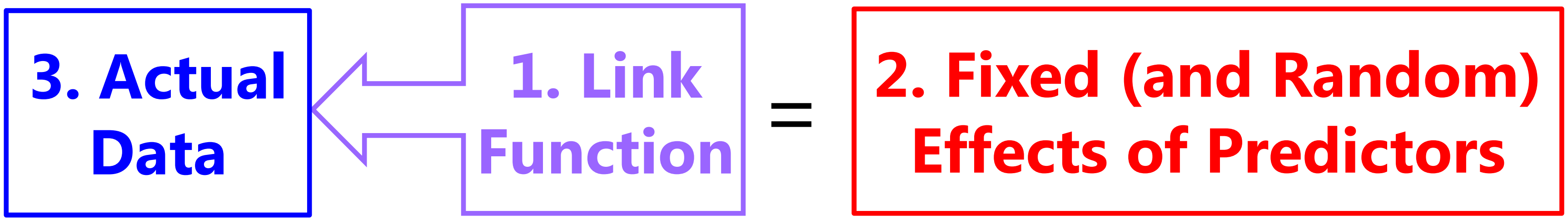
Generali*zed* Linear Models

- **Generali*zed* linear models:** link-transformed Y is predicted instead of actual Y; ML estimator uses not-normal distributions
  - **Single-level models →** residuals follow some not-normal distribution
  - **Multilevel models →** level-1 residuals follow some not-normal distribution, but level-2 random effects are almost always 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)

} These two are often called "multinomial" inconsistently

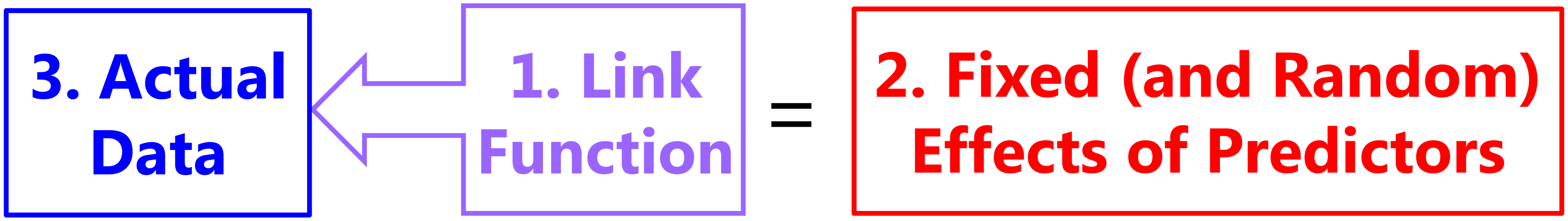
3 Parts of Generalized Multilevel Models



3. Actual Data ← **1. Link Function** = **2. Fixed (and Random) Effects of Predictors**

1. **Link Function** (different from general): How a non-normal outcome is transformed into an **unbounded** outcome that the model fixed and random effects can predict linearly
   - We can then convert the transformed prediction back into the Y scale
   - This way the predicted outcomes will stay within the sample space (boundaries) of the observed data (e.g., 0/1 for binary outcomes—the model should not predict −1 or 2, so linear slopes need to shut off)
   - Written as $g(\cdot)$ for link and $g^{-1}(\cdot)$ for inverse link (to go back to data)
   - For outcomes with residuals that are already normal, general linear models are just a special case with an "identity" link function (Y * 1)
     - So general linear models are a special case of generali*zed* linear models, and general linear mixed models are a special case of generali*zed* linear mixed models

3 Parts of Generalized Multilevel Models



3. Actual Data

1. Link Function

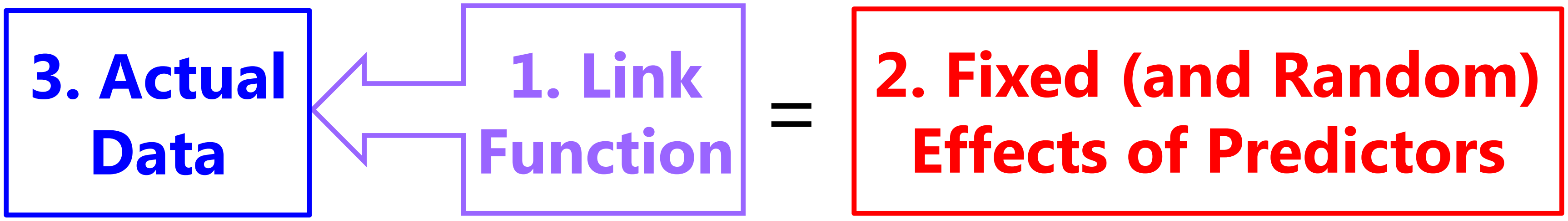
=

2. Fixed (and Random) Effects of Predictors

2. **Linear Predictor** (same as in general): How the model predictors linearly relate to the outcome conditional mean

- This works the same as usual, except the linear predictor model **directly predicts the link-transformed conditional mean**, which we then can convert (inverse link) back into the scale of original outcome
- That way we can still use the familiar "one-unit change" language to describe the effects of model predictors (on the 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

3 Parts of Generalized Multilevel Models



3. Actual Data ← **1. Link Function** = **2. Fixed (and Random) Effects of Predictors**

3. **Model for Level-1 Residuals** (different than general): how level-1 residuals should be distributed given the sample space of the actual outcome
   - Many alternative distributions that map onto what the distribution of **residuals** could possibly look like (and kept within sample space)
   - **Why?** To get the most correct **standard errors** for fixed effects
   - You can think of this as "model for the variance" still, but not all distributions will actually have an estimated residual variance

- Let's start with models for **binary data** to illustrate these 3 parts…

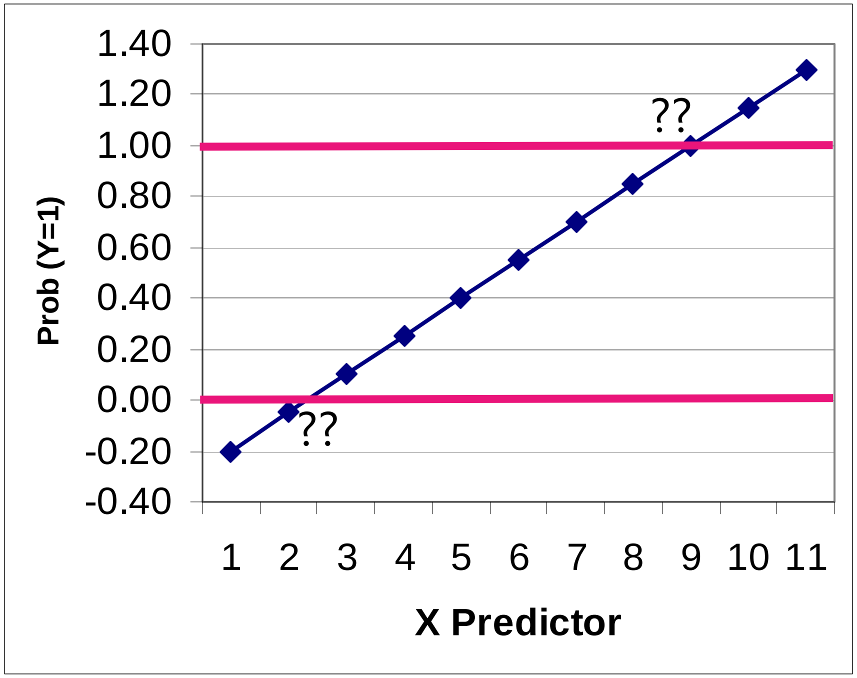
Normal GLM for Binary Outcomes?

- Let's say we have a single binary (0 or 1) outcome… *(concepts for multilevel data will proceed similarly)*
  - **Conditional mean** is proportion of people who have a 1, so the **probability of having a 1** is what we're trying to predict for each person, given the predictor values: $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1})$
  - 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 $\mathbf{y_i} = \mathbf{(predicted\ probability\ of\ 1)} + \mathbf{e_i}$
  - **What could possibly go wrong?**

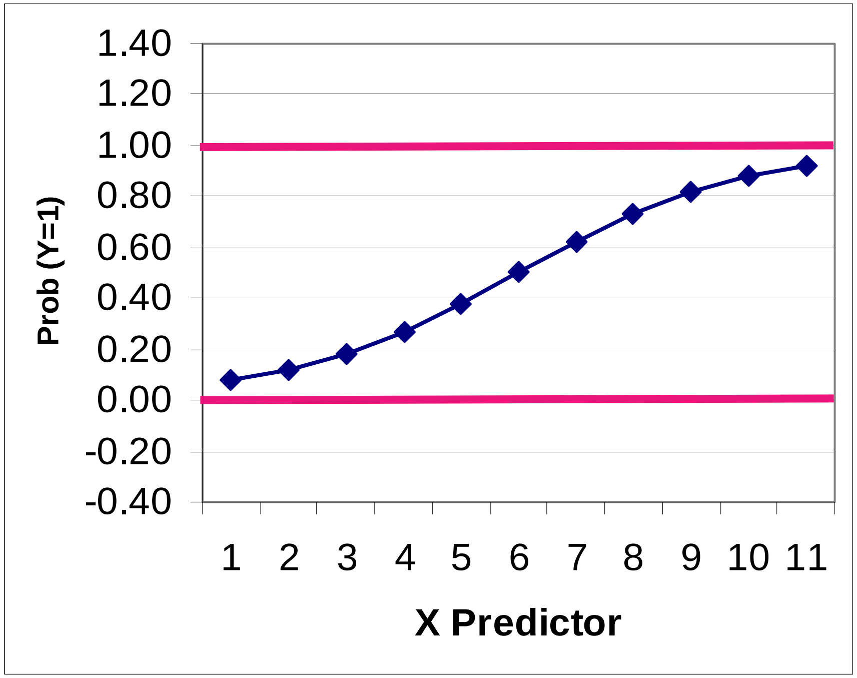
Normal GLM for Binary Outcomes?

- 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...

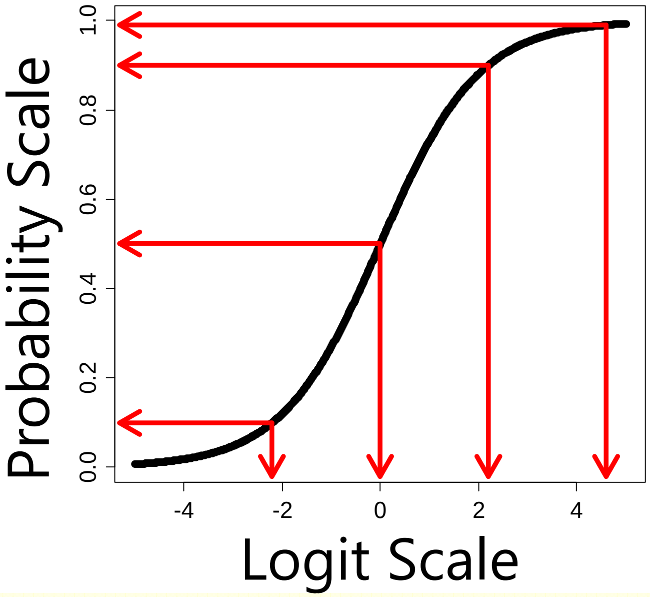


But we need this...



Generalized Models for Binary Outcomes

- Solution #1: Rather than predicting $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1})$ directly, we must transform it into an unbounded variable with a **link function**:
  - Transform **probability** into an **odds ratio**: $\frac{p}{1-p} = \frac{\text{prob(y=1)}}{\text{prob(y=0)}}$
    - If $p(\mathrm{y_i} = 1) = .7$ then $\mathrm{Odds}(1) = 2.33$; $\mathrm{Odds}(0) = .429$
    - But odds scale is skewed, asymmetric, and ranges from 0 to +∞ → Not helpful
  - **Take *natural log of odds ratio* → called "logit" link:** $\mathbf{Log}\left[\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}\right]$
    - If $p(\mathrm{y_i} = 1) = .7$, then $\mathrm{Logit}(1) = .846$; $\mathrm{Logit}(0) = -.846$
    - 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

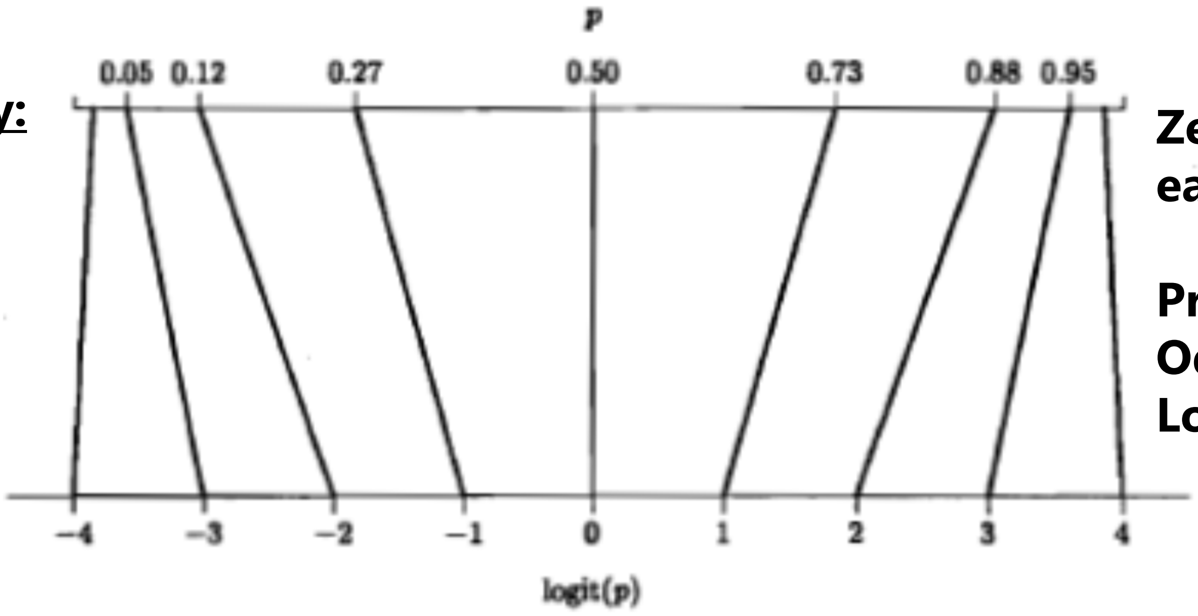
Can you guess what $p(.01)$ would be on the logit scale?

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 outcome conditional mean shuts off at 0 or 1 as needed**

Probability:
$\boldsymbol{p(\mathrm{y_i} = 1)}$

Logit:
$\mathbf{Log}\left[\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}\right]$



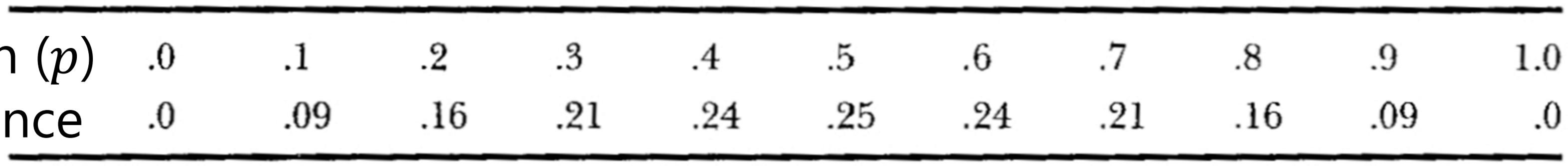
Zero-point on each scale:

Prob = .5
Odds = 1
Logit = 0

Normal GLM for Binary Outcomes?

- General linear model: $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1}) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}\mathbf{X_i} + \boldsymbol{\beta_2}\mathbf{Z_i} + \mathbf{e_i}$
- If $\mathbf{y_i}$ is binary, then $\mathbf{e_i}$ can only be 2 things: $\mathbf{e_i} = \mathbf{y_i} - \mathbf{\hat{y}_i}$
  - 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

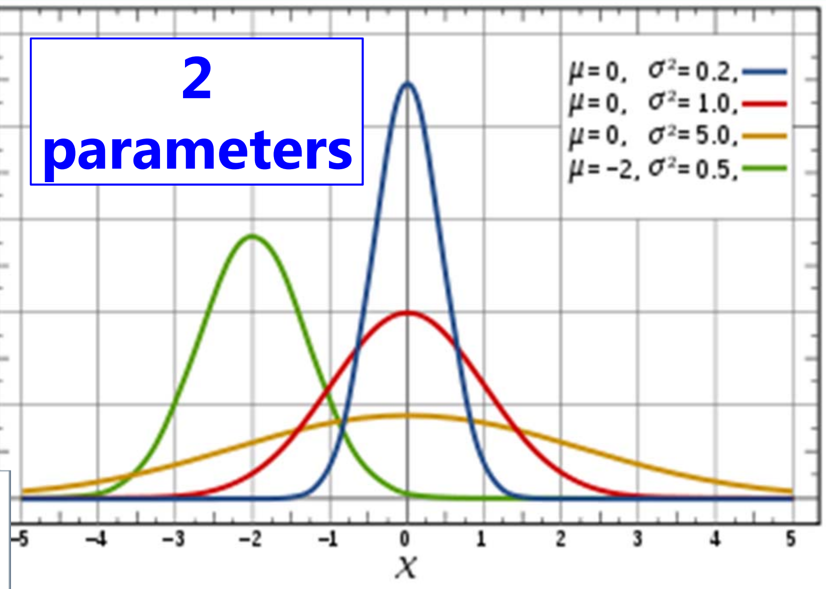
Solution #2: Bernoulli Distribution

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

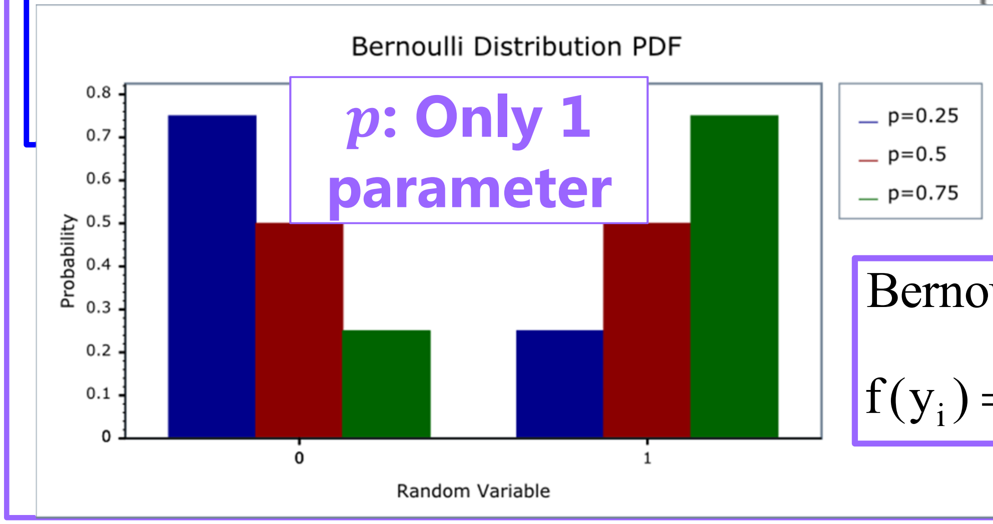
Univariate Normal PDF:

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

Likelihood (yᵢ)



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

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

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

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

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

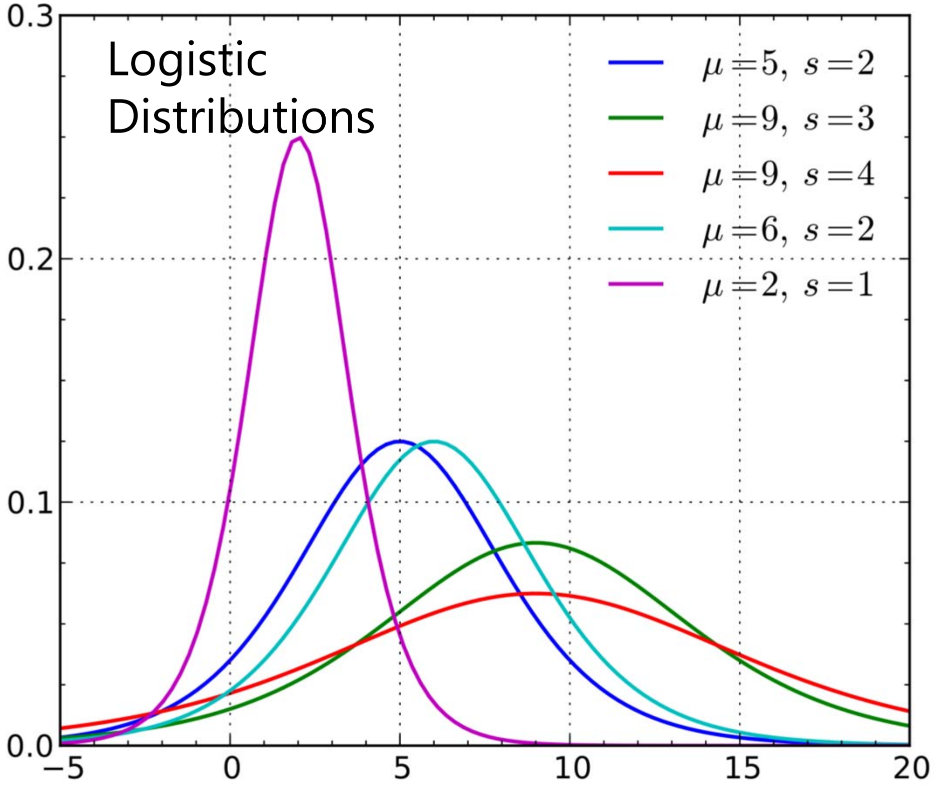
"Logistic Regression" for Binary Data

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

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

$threshold = \beta_0 * -1$ 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** $\boldsymbol{y_i^*}$, 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 logistic regression for identification)

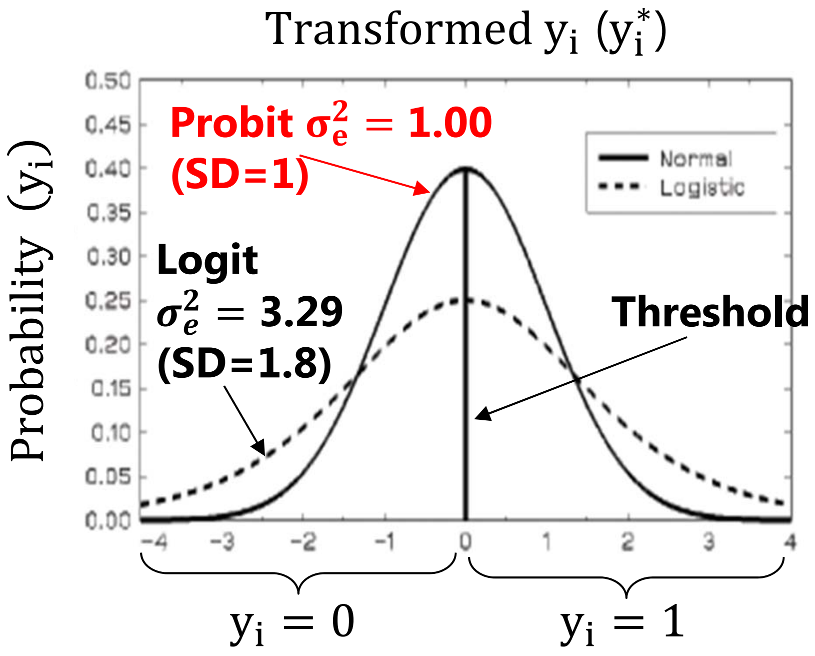
Other Models for Binary Data

- 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:

    $$\mathrm{Probit}(\mathrm{y_i} = 1) = \Phi^{-1}P(\mathrm{y_i} = 1) = \textit{your model} \longleftarrow \boxed{\mathbf{g(\cdot)}}$$

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

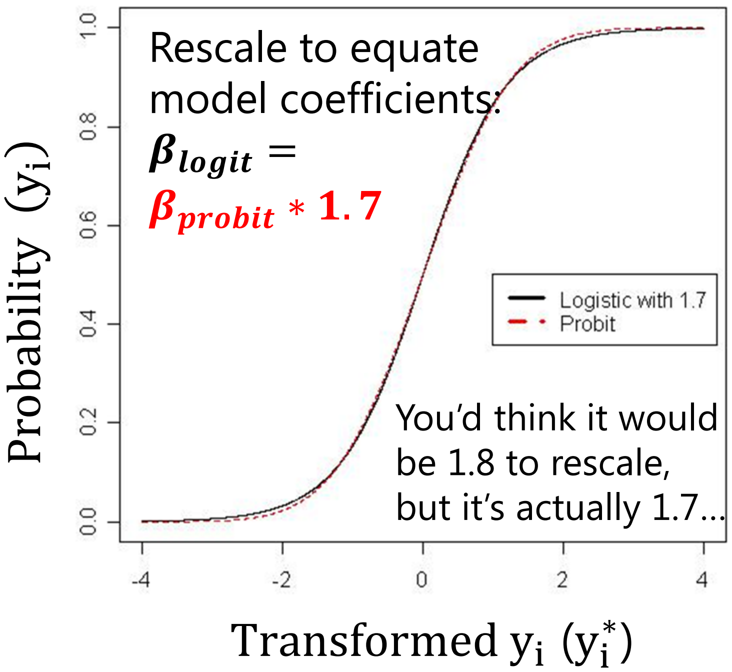
Probit vs. Logit: Should you care? Pry not.



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

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

Threshold



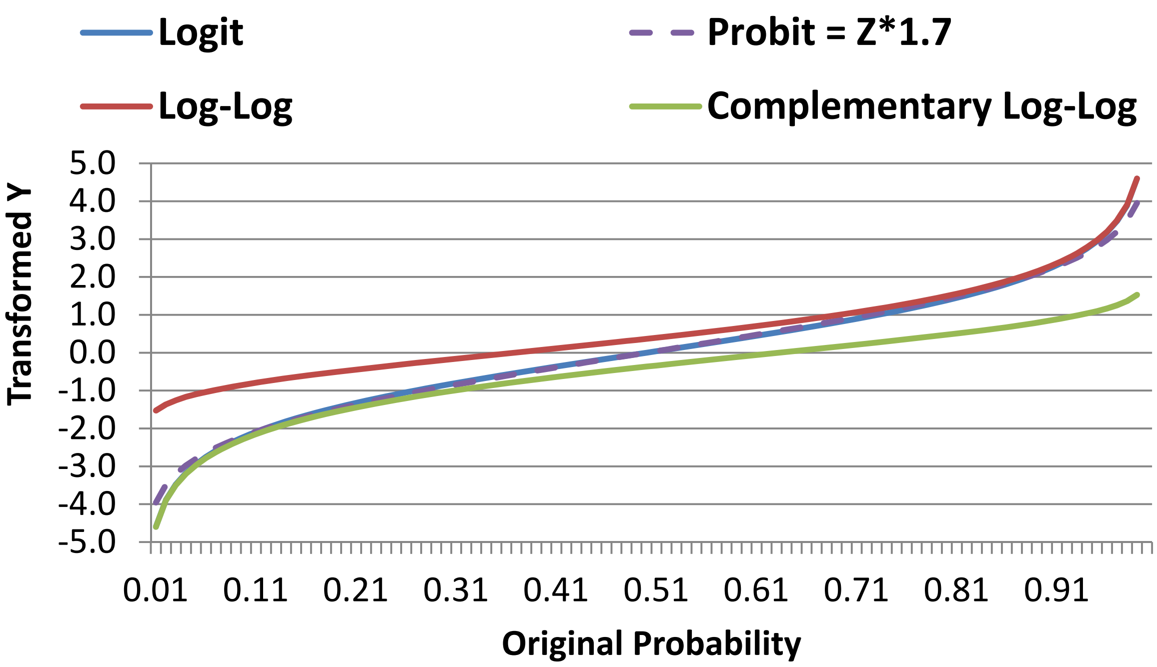
Rescale to equate model coefficients: $\boldsymbol{\beta_{logit}} =$ $\boldsymbol{\beta_{probit} * 1.7}$

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(·) for new y_i:	$\mathrm{Log}\left(\frac{p}{1-p}\right) = \mu$	$\Phi^{-1}(p) = \mu$	$-\mathrm{Log}[-\mathrm{Log}(p)] = \mu$	$\mathrm{Log}[-\mathrm{Log}(1-p)] = \mu$
$g^{-1}(\cdot)$ to get back to probability:	$p = \dfrac{\exp(\mu)}{1+\exp(\mu)}$	$p = \Phi(\mu)$	$p = \exp[-\exp(-\mu)]$ $e_i \sim \text{extreme value}\left(-\gamma?, \sigma_e^2 = \dfrac{\pi^2}{6}\right)$	$p = 1 - \exp[-\exp(\mu)]$
In SAS LINK=	LOGIT	PROBIT	LOGLOG	CLOGLOG

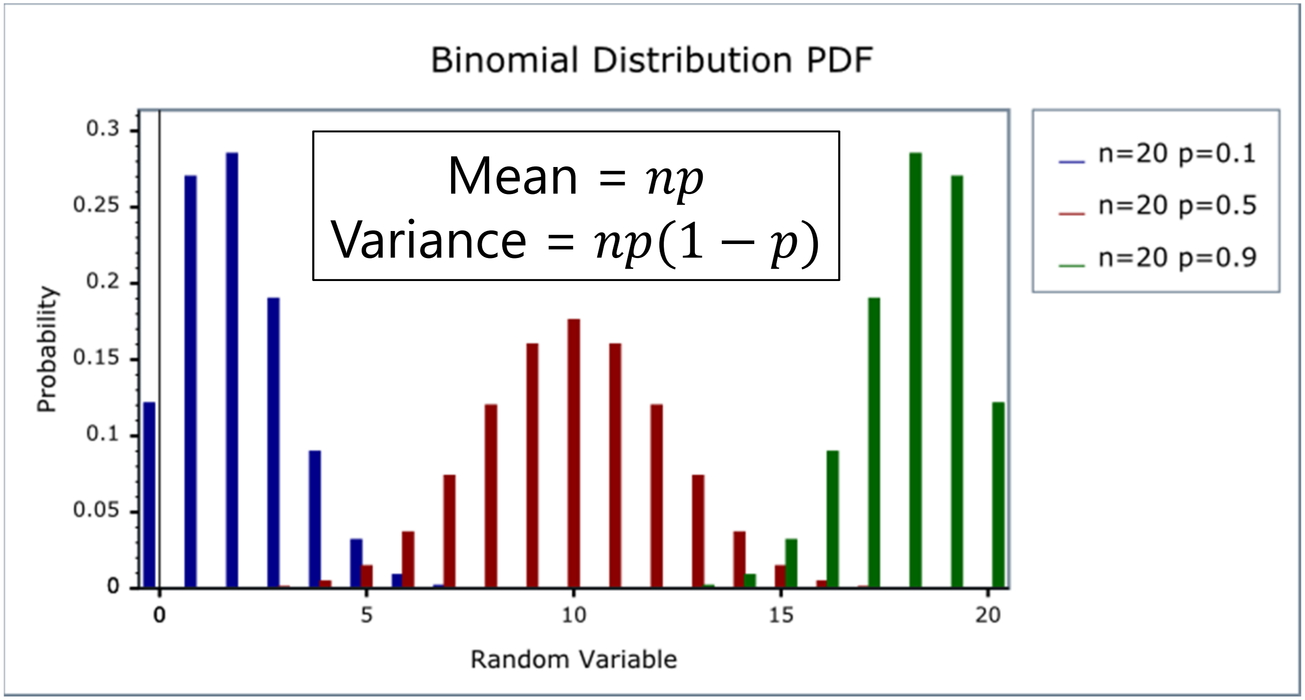
Too Logit to Quit: Predicting Proportions

- 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: → μ in logits $= \text{Log}\left(\frac{p}{1-p}\right)$ ← $\mathbf{g(\cdot)}$ **Link**
  - Model to data → $p = \frac{\exp(\mu)}{1+\exp(\mu)}$ ← $\mathbf{g^{-1}(\cdot)}$ **Inverse-Link**
- However, because the outcome values aren’t just 0 or 1, a Bernoulli residual 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
    - I *think* it’s ok to cheat by rescaling to fit between 0 and 1, though

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

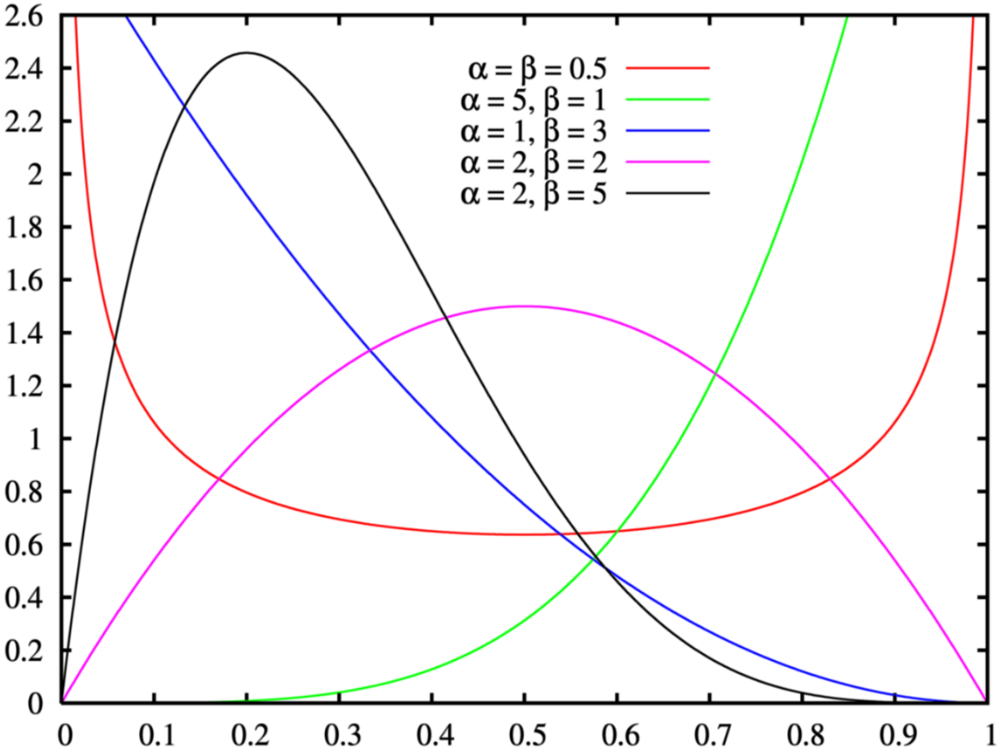
- SAS PROC GLIMMIX allows the outcome variable to be defined as ***#events/#trials*** on MODEL statement
  - LINK=LOGIT so that the conditional mean stays bounded between 0 and 1 as needed (or alternatively, CLOGLOG/LOGLOG)
  - DIST=BINOMIAL so variance (and SEs) are determined by that mean, as they should be assuming independent events

- 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

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

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

I've named these submodels based on what they predict, but SAS will name them its own way in the output.

- 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: $\text{Logit}(y_i > 0) = \beta_{01}$ → $P(y_i > 0) = \exp(\beta_{01})/[1 + \exp(\beta_{01})]$
- Submodel 2: $\text{Logit}(y_i > 1) = \beta_{02}$ → $P(y_i > 1) = \exp(\beta_{02})/[1 + \exp(\beta_{02})]$
- Submodel 3: $\text{Logit}(y_i > 2) = \beta_{03}$ → $P(y_i > 2) = \exp(\beta_{03})/[1 + \exp(\beta_{03})]$

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

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

→ Prob_1 → Prob_2 → Prob_3

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

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

- 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 = $1 - \text{Prob}_1$
Probability of 1 = $\text{Prob}_1 - \text{Prob}_2$
Probability of 2 = $\text{Prob}_2 - \text{Prob}_3$
Probability of 3 = $\text{Prob}_3 - 0$

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, respectively, in SAS PROC GLIMMIX

- Almost always assume **proportional odds**, that effects of predictors are the same across binary submodels—for example (subscripts = parm, submodel)
  - Submodel 1: $\mathrm{Logit}(\mathrm{y_i} > 0) = \boldsymbol{\beta_{01}} + \beta_1\mathrm{X_i} + \beta_2\mathrm{Z_i} + \beta_3\mathrm{X_i}\mathrm{Z_i}$
  - Submodel 2: $\mathrm{Logit}(\mathrm{y_i} > 1) = \boldsymbol{\beta_{02}} + \beta_1\mathrm{X_i} + \beta_2\mathrm{Z_i} + \beta_3\mathrm{X_i}\mathrm{Z_i}$
  - Submodel 3: $\mathrm{Logit}(\mathrm{y_i} > 2) = \boldsymbol{\beta_{03}} + \beta_1\mathrm{X_i} + \beta_2\mathrm{Z_i} + \beta_3\mathrm{X_i}\mathrm{Z_i}$

- Proportional odds essentially means no interaction between submodel and predictor effects, which greatly reduces the number of estimated parameters
  - Assumption for single-level data can be tested painlessly using PROC LOGISTIC, which provides a global SCORE test of equivalence of all slopes between submodels
  - If the proportional odds assumption fails and $C > 3$, you'll need to write your own model non-proportional odds ordinal model in PROC NLMIXED

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 $c = y_i$

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

Only p_{ic} for 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: $\sum_{\mathrm{c=1}}^{\mathrm{C}} \mathrm{p_{ic}} = 1$

- 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 $0 = \mathrm{ref}$):

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

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

One More Idea…

- Ordinal data can sometimes also be approximated with a logit link and binomial distribution instead
  - Example: Likert scale from $0\text{–}4$ → $\#\,\mathrm{trials} = 4, \#\,\mathrm{correct} = \mathrm{y_i}$
  - Model predicts p of binomial distribution, $p * \#trials\ =\ mean$
  - $p(\mathrm{y_i})$ = proportion of sample expected in that y_i response category
- Advantages:
  - Only estimates one parameter that creates a conditional mean for each response category, instead of $C - 1$ cumulative intercepts or thresholds
  - Can be used even if there is sparse data in some categories
  - Results may be easier to explain than if using cumulative sub-models
- Disadvantages:
  - # persons in each category will not be predicted perfectly to begin with, so it may not fit the data as well without the extra intercept parameters

Generalized Models: Summary

- Statistical models come from probability distributions
  - Specifically, residuals are assumed to have some 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. 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
  2. Linear predictor: how we predict that conditional mean
  3. Residuals model: what kind of distribution they follow

Multivariate Data in PROC GLIMMIX

- Multivariate models can be fitted in PROC GLIMMIX using stacked data, same as in MIXED… first, the bad news:
  - There is no **R** matrix in true ML, only **G**, and **V** can't be printed, either, which sometimes makes it hard to tell what structure is being predicted
  - There is no easy way to allow different scale factors given the same link and distribution across multivariate outcomes (as far as I know)
  - This means that a random intercept can be included to create constant covariance across outcomes, but that any differential variance (scale) or covariance must be included via RANDOM statement as well (to go in **G**)

- Now, the good news:
  - It allows different links and distributions across outcomes using LINK=BYOBS and DIST=BYOBS (Save new variables called "link" and "dist" to your data to tell GLIMMIX what to use per outcome)
  - It will do −2ΔLL tests for you using the COVTEST option! (not in MIXED)

From Single-Level to Multilevel…

- Multilevel generalized models have the same 3 parts as single-level generalized models:
  - Link function to transform bounded conditional mean into unbounded
  - Linear model directly predicts link-transformed conditional mean instead
  - Alternative distribution of level-1 residuals used (e.g., Bernoulli)

- But in adding random effects (i.e., additional piles of variance) to address dependency in longitudinal/clustered data:
  - Piles of variance are ADDED TO, not EXTRACTED FROM, the original residual variance pile when it is fixed to a known value (e.g., 3.29), which causes the model 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

Empty Multilevel Model for Binary Outcomes

- **Level 1:** $\mathbf{Logit(y_{ti}) = \beta_{0i}}$
- **Level 2:** $\mathbf{\beta_{0i} = \gamma_{00} + U_{0i}}$

Note what's NOT in level 1…

- **Composite:** $\mathbf{Logit(y_{ti}) = \gamma_{00} + U_{0i}}$

- σ_e^2 residual variance is not estimated → $\mathbf{\pi^2/3 = 3.29}$
  - (Known) residual is in model for actual Y, not prob(Y) or logit(Y)

- $\text{Logistic ICC} = \frac{\text{BP}}{\text{BP+WP}} = \frac{\boldsymbol{\tau}_{\mathbf{U_0}}^{\mathbf{2}}}{\boldsymbol{\tau}_{\mathbf{U_0}}^{\mathbf{2}}+\boldsymbol{\sigma}_{\mathbf{e}}^{\mathbf{2}}} = \frac{\boldsymbol{\tau}_{\mathbf{U_0}}^{\mathbf{2}}}{\boldsymbol{\tau}_{\mathbf{U_0}}^{\mathbf{2}}+\mathbf{3.29}}$

- Can do −2ΔLL test to see if $\boldsymbol{\tau}_{\mathbf{U_0}}^{\mathbf{2}} > 0$, although the ICC is somewhat problematic to interpret due to non-constant residual variance
  - Have not seen equivalent ICC formulas for other generalized models besides binary

Random Linear Time Model for **Binary** Outcomes

- **Level 1:** $\mathrm{Logit}(y_{ti}) = \beta_{0i} + \beta_{1i}(\mathrm{time}_{ti})$
- **Level 2:** $\beta_{0i} = \gamma_{00} + U_{0i}$
  $\beta_{1i} = \gamma_{10} + U_{1i}$

- **Combined:** $\mathrm{Logit}(y_{ti}) = (\gamma_{00} + U_{0i}) + (\gamma_{10} + U_{1i})(\mathrm{time}_{ti})$

- σ_e^2 residual variance is still not estimated → **$\pi^2/3 = 3.29$**
- Can test new fixed or random effects with $-2\Delta LL$ tests (or Wald test *p*-values for fixed effects as usual)

Random Linear Time Model for **Ordinal** Outcomes ($C = 3$)

- **L1:** **$\text{Logit}(y_{ti1}) = \beta_{0i1} + \beta_{1i1}(\text{time}_{ti})$**

  $\text{Logit}(y_{ti2}) = \beta_{0i2} + \beta_{1i2}(\text{time}_{ti})$

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

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

- 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 parameters can be constrained
  - σ_e^2 residual variance is still not estimated → **$\pi^2/3 = 3.29$**

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
  - $\mathbf{U_{0i}}$ is "individual deviation from sample average"

- What "average" means in general*ized* linear mixed models is different, because the natural log is a nonlinear function:
  - So 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*** $\boldsymbol{U_i = 0}$
    - Fixed effects are *conditional* on the random effects
    - This gets called a "unit-specific" or "subject-specific" model
    - This distinction does not exist for normally distributed outcomes

Comparing Results across Models

- NEW RULE: Coefficients cannot be compared across models, because they are not on the same scale! (see 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 residual variance** like usual, so all other models 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 is the same

A Little Bit about Estimation

- Goal: End up with maximum likelihood estimates for all model parameters (because they are consistent, efficient)
  - When we have a **V** matrix based on multivariate **normally** distributed $\mathbf{e}_{ti}$ residuals at level-1 and multivariate normally distributed $\mathbf{U}_i$ terms at level 2, ML is easy
  - When we have a **V** matrix based on multivariate **Bernoulli** distributed $\mathbf{e}_{ti}$ residuals at level-1 and multivariate normally distributed $\mathbf{U}_i$ terms at level 2, ML is much harder
    - Same with any other kind model for "not normal" level 1 residual
    - **ML does not assume normality unless you fit a "normal" model!**

- 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)

2 Main Types of Estimation

- **Quasi-Likelihood methods** → older methods
  - "Marginal QL" → approximation around fixed part of model
  - "Penalized QL" → approximation around fixed + random parts
  - These both underestimate variances (MQL more so than PQL)
  - 2nd-order PQL is supposed to be better than 1st-order MQL
  - QL methods DO NOT PERMIT MODEL −2ΔLL TESTS
  - HLM program adds Laplace approximation to QL, which then does permit valid −2ΔLL tests (also in SAS GLIMMIX and STATA xtmelogit)

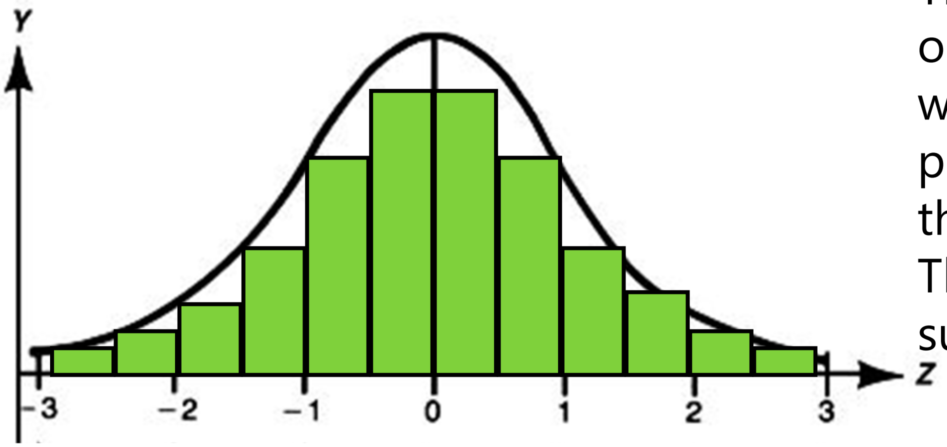
- **ML via Numerical Integration** → gold standard
  - Much better estimates and valid−2ΔLL tests, but can take for-freaking-ever (can use PQL methods to get good start values)
  - Will blow up with many random effects (which make the model exponentially more complex, especially in these models)
  - Relies on assumptions of local independence, like usual → all level-1 dependency has been modeled; level-2 units are independent

ML via Numerical 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 variance is 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 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")
  - Can either divide the whole distribution into rectangles, then repeat, 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: Generalized Multilevel Models

- Analyze link-transformed conditional mean of DV (e.g., via logit, log…)
  - ***Linear*** relationship between X's and ***transformed*** conditional mean of Y
  - ***Nonlinear*** relationship between X's and ***original*** Y
    - Original e_{ti} residuals are assumed to follow some non-normal distribution

- In models for binary or categorical data, level-1 residual variance is fixed
  - So it can't go down after adding level-1 predictors, which means that the scale of everything else 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 estimates may not be comparable across models as a result

- Estimation is trickier and takes longer
  - 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 positive infinity
    - 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 (with decimals)

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 the how much part)

Discrete 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

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

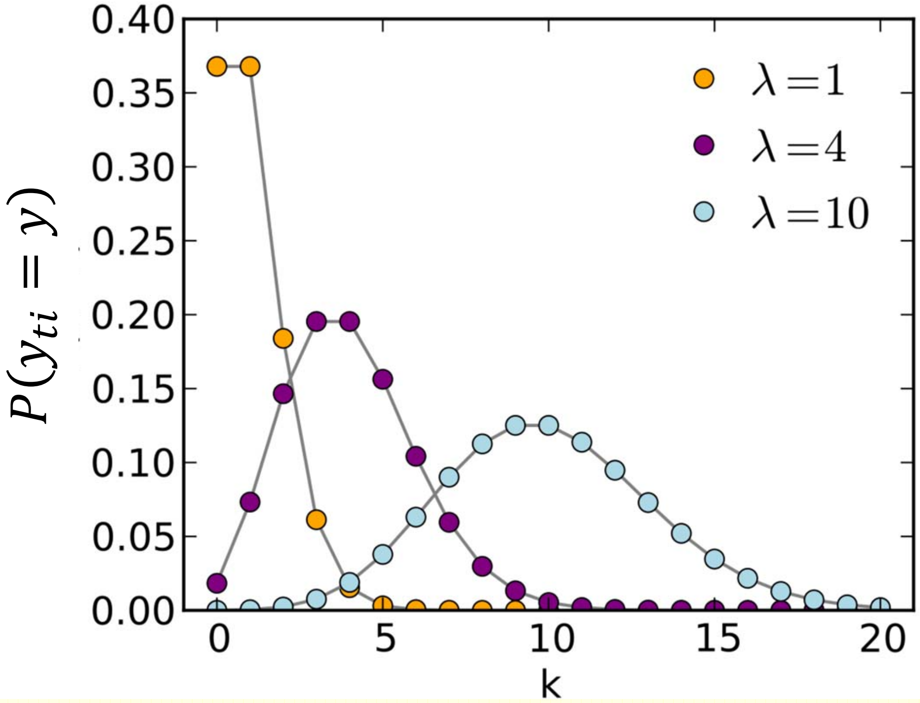
  - e.g., if $\mathrm{Log}(\mu_\mathrm{i}) = \mathrm{model}$ provides predicted $\mathrm{Log}(\mu_\mathrm{i}) = 1.098$, that translates to an actual predicted count of $\exp(1.098) = 3$
  - e.g., if $\mathrm{Log}(\mu_\mathrm{i}) = \mathrm{model}$ provides predicted $\mathrm{Log}(\mu_\mathrm{i}) = -5$,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

Poisson Distribution for Residuals

- 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 $\lambda > 6$ 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

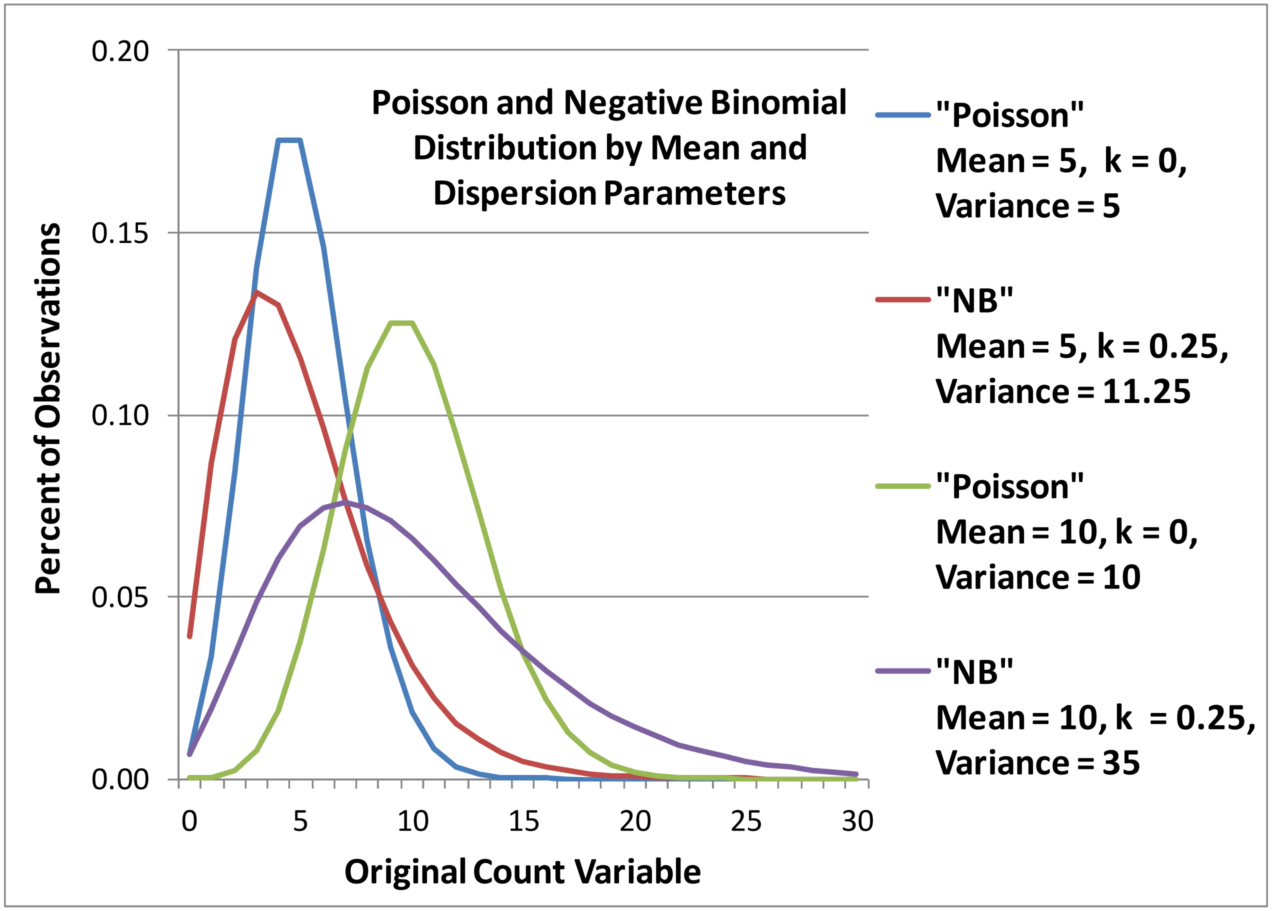
- Model: $\mathrm{Log}(y_i) = \mathrm{Log}(\mu_i) = \beta_0 + \beta_1 X_i + \beta_2 Z_i + e_i^G$
- Negative Binomial PDF with a new k dispersion parameter is now:
  - $\mathrm{Prob}(y_i = y|\beta_0, \beta_1, \beta_2) = \frac{\Gamma\left(y+\frac{1}{k}\right)}{\Gamma(y+1)*\Gamma\left(\frac{1}{k}\right)}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

DIST = NEGBIN in SAS

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 through user-defined functions)

Negative Binomial (NB) = “Stretchy” Poisson…



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

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

A Negative Binomial model can be useful for count residuals with extra skewness, but otherwise follow a Poisson 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

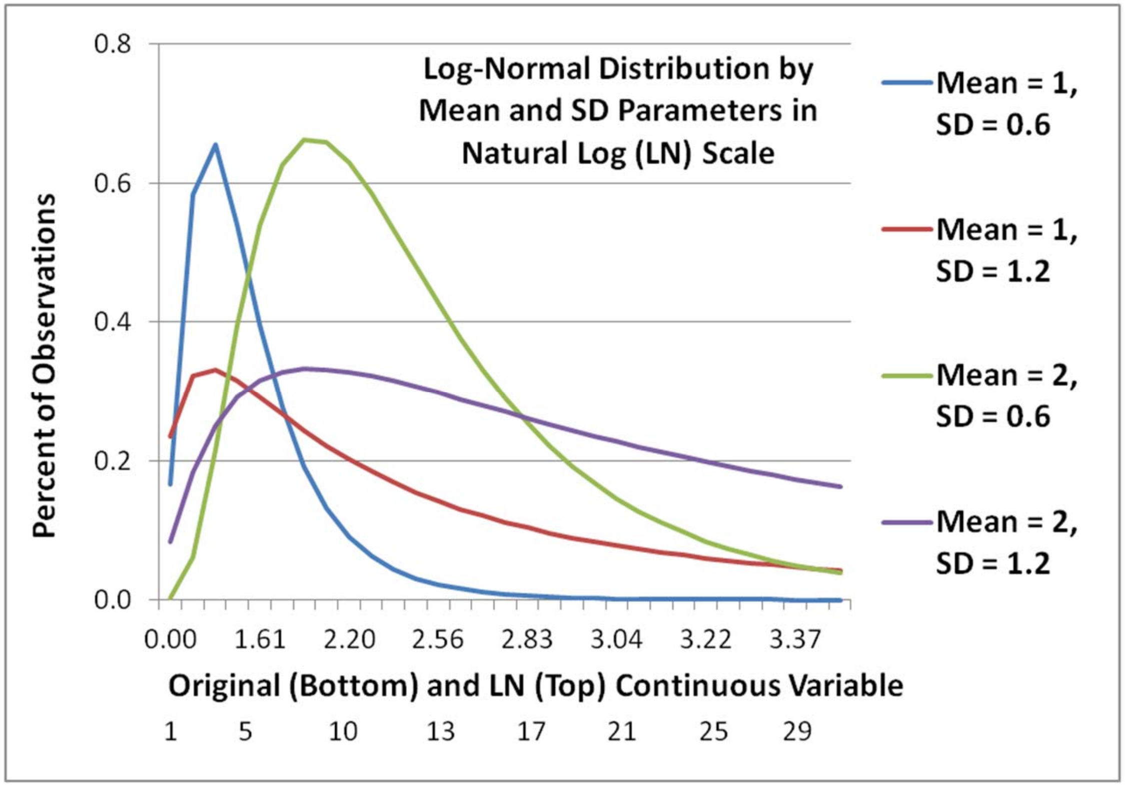
- "**Zero-Altered**" or "**Zero-Truncated**" Poisson or Negative Binomial: ZAP/ZANB or ZTP/ZTNB (used in hurdle models)
  - Is usual count distribution, just not allowing any 0 values
  - Poisson version is readily available within SAS PROC FMM using DIST=TRUNCPOISSON (next version should have TRUNCNEGBIN, too)
  - Could be fitted in SAS NLMIXED or Mplus, too

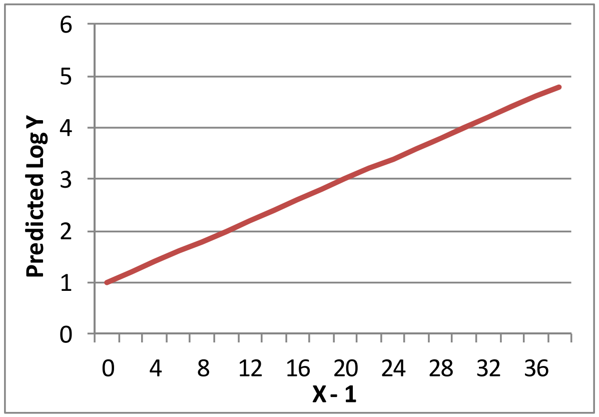
- Poisson PDF was: $\mathrm{Prob}(y_i = y|\mu_i\ ) = \frac{\mu_i^y * \exp(-\mu_i)}{y!}$
- Zero-Truncated Poisson PDF is:
  - $\mathrm{Prob}\big(y_i = y|\mu_{i,}y_i > 0\big) = \frac{\mu_i^y * \exp(-\mu_i)}{y![1-\exp(-\mu_i)]}$
  - $\mathrm{Prob}(y_i = 0) = \exp(-\mu_i)$, so $\mathrm{Prob}(y_i > 0) = 1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

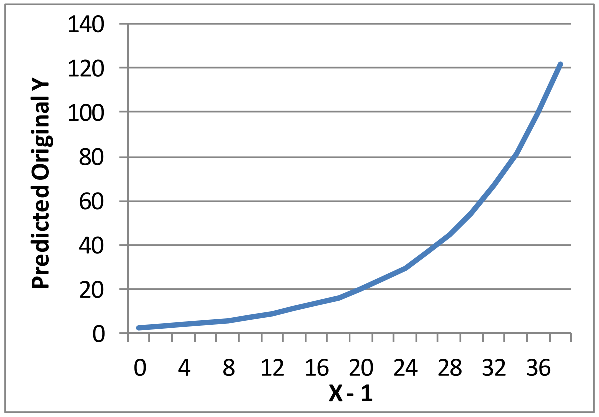
Software for Discrete Outcomes

- 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)

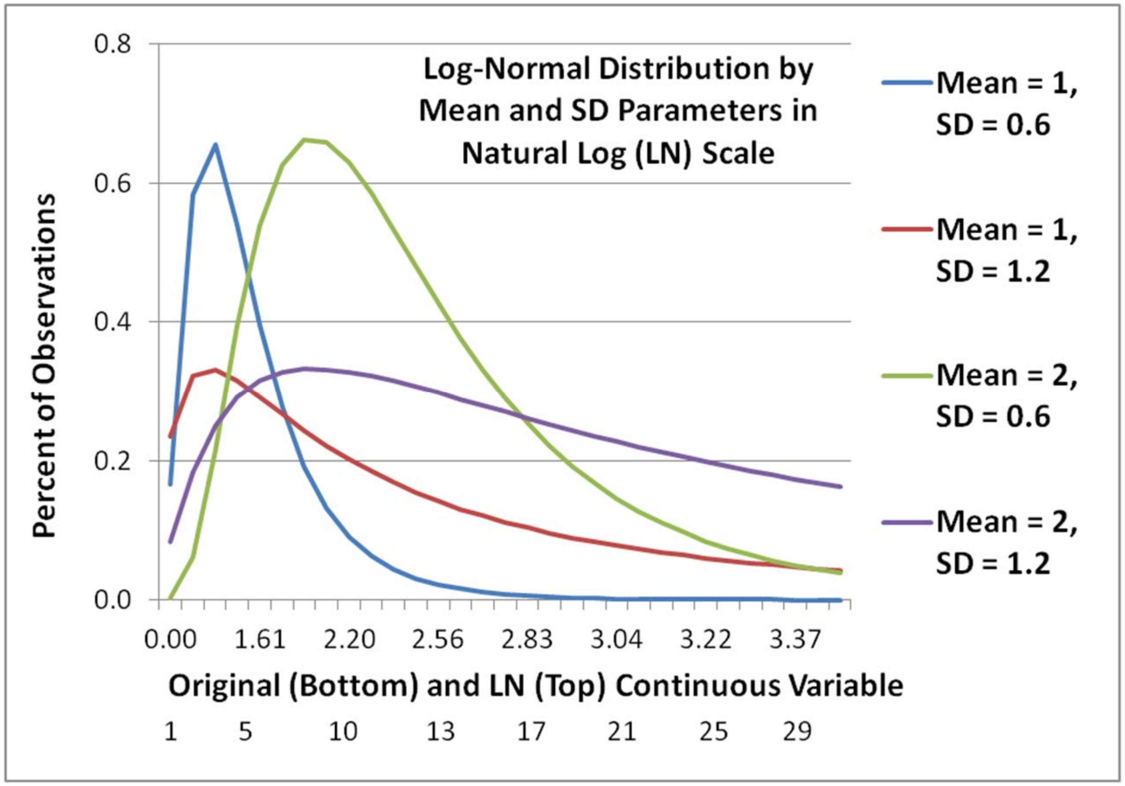


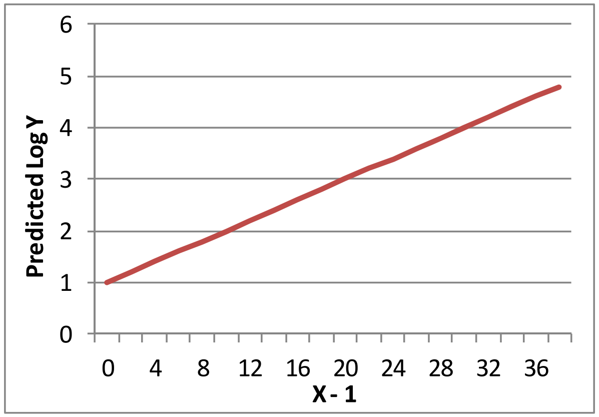


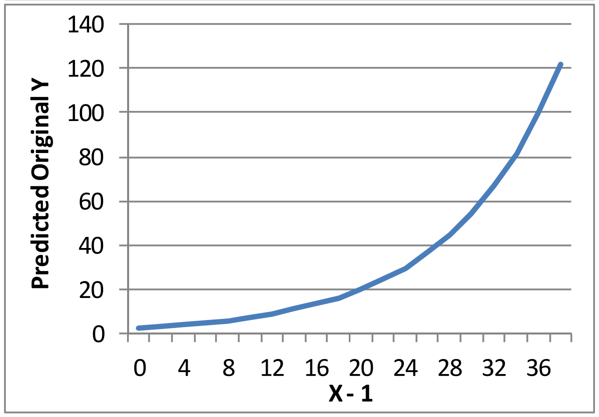


- $e_i \sim \text{LogNormal}(0, \sigma_e^2)$ → **log** of residuals is normal
  - Is same as log-transforming your outcome in this case…
  - The log link keeps the predicted values positive, but slopes then have an exponential (not linear) relation with original outcome

Log-Normal Distribution (Link=Identity)

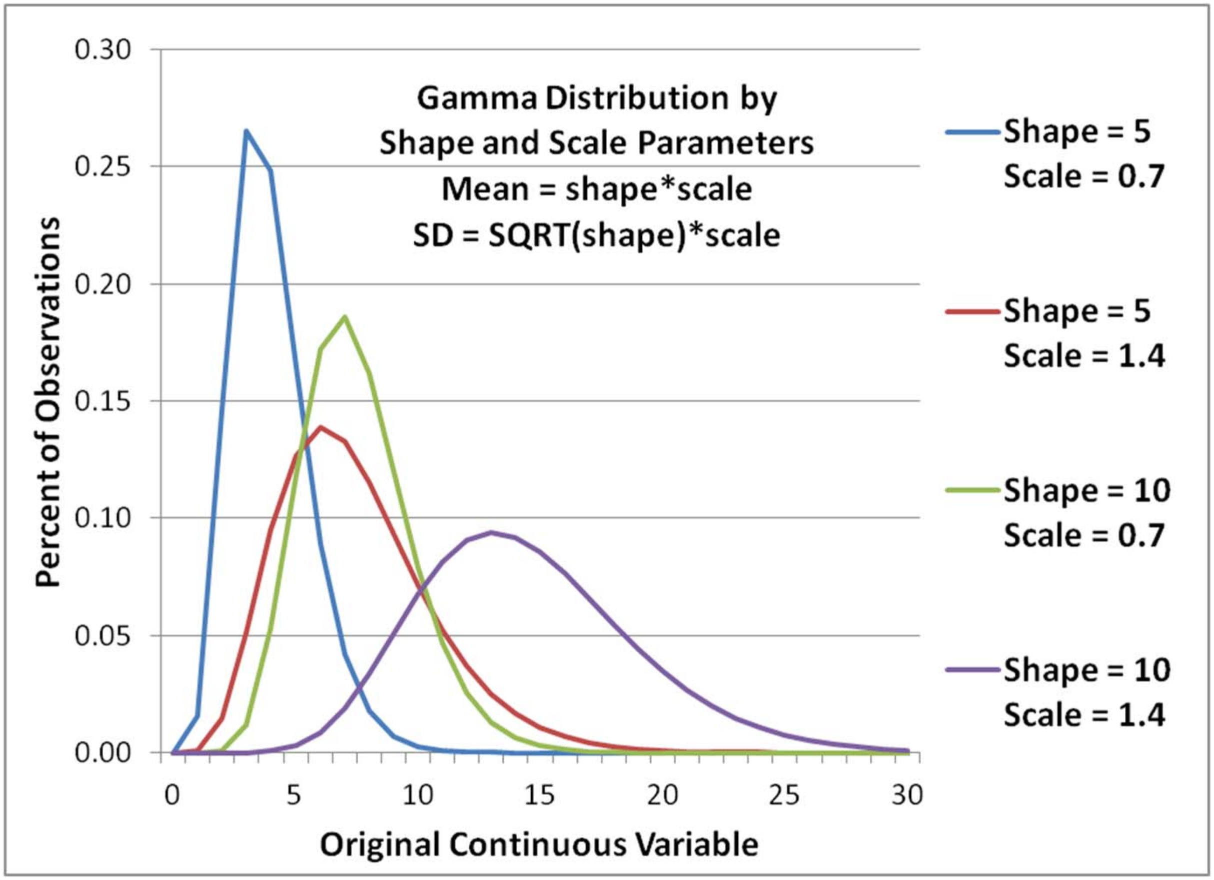


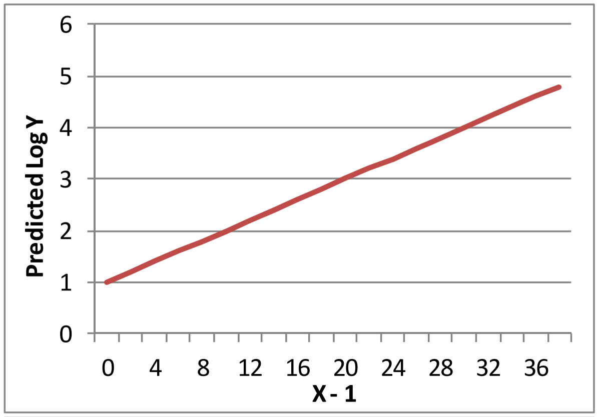


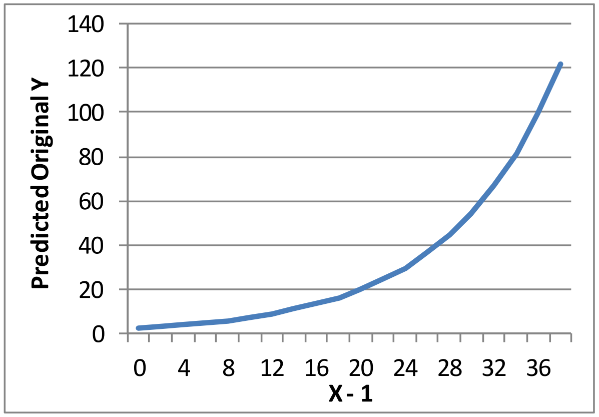


- GLIMMIX parameterization gives μ (= intercept) and $scale$ = (variance) to convert back into original data as follows:
  - $\text{Mean(Y)} = \exp(\mu) * \sqrt{\exp(scale)}$
  - $\text{Variance(Y)} = \exp(2\mu) * \exp(scale) * [\exp(scale) - 1]$

Gamma Response Distribution







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

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)
- **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 Part 1)
- 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 or NB, Hurdle Poisson or NB
  - Continuous: Two-Part (with normal or lognormal for how much)
  - Many of these can be estimated directly in Mplus or SAS GLIMMIX, but some will need to be programed in SAS GLIMMIX or NLMIXED
  - More options for single-level data in SAS PROC FMM

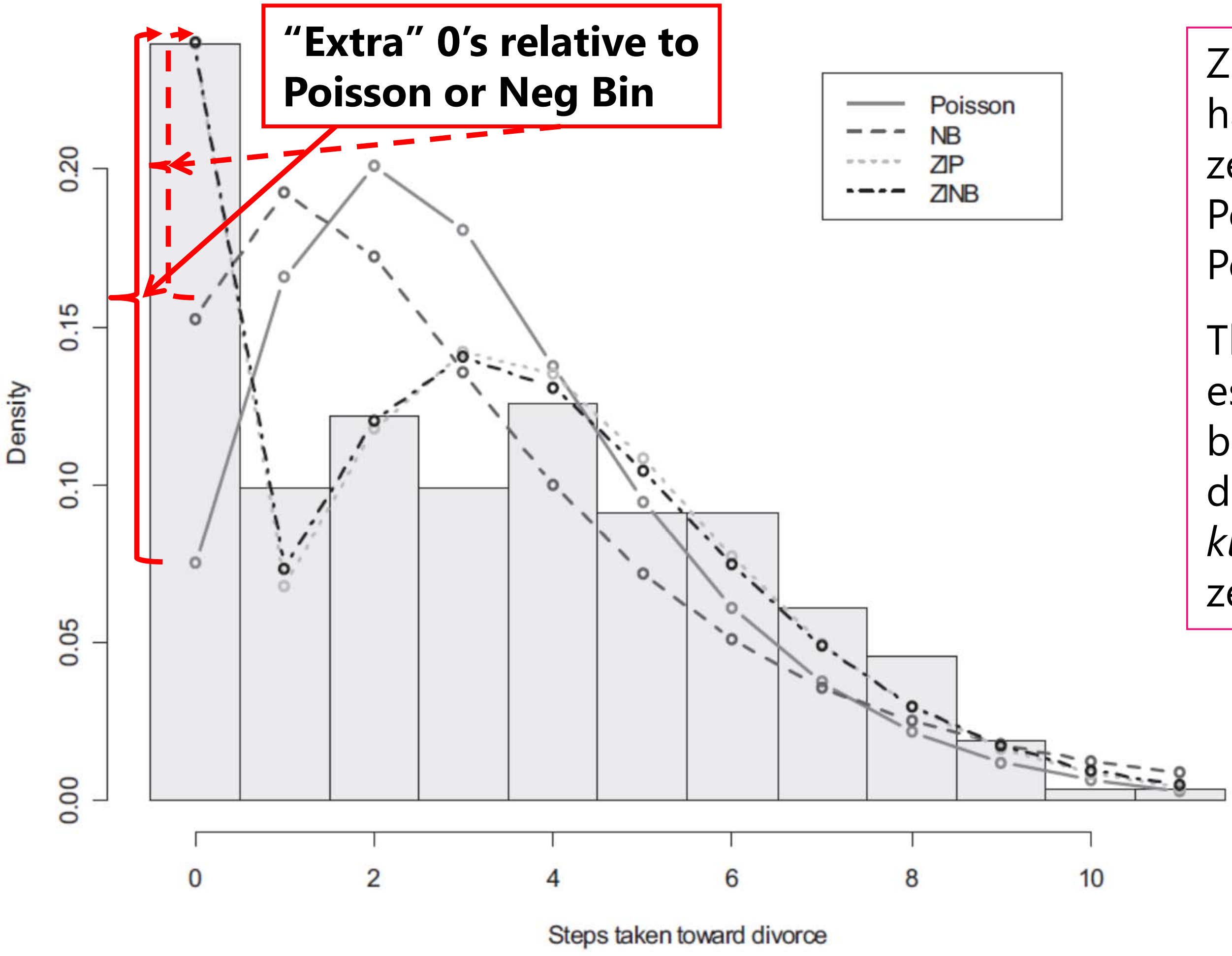
Problem #3: Too many 0 values, Option #1

- "**Zero-Inflated**" Poisson (DIST=ZIP) or Negative Binomial (DIST=ZINB); available within SAS PROC GENMOD (and Mplus)
  - Distinguishes **two kinds of 0 values**: **expected** 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: $\mathrm{Logit}(\mathrm{y_i} = \mathrm{extra\ 0}) = \ \beta_{01} + \beta_{11}\mathrm{X_i} + \beta_{21}\mathrm{Z_i}$
  - 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 (including 0's) 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:
  $$\omega_i = \frac{\exp[\text{Logit}(y_i = \text{extra } 0)]}{1 + \exp[\text{Logit}(y_i = \text{extra } 0)]}$$
  - μ_i is the predicted count for the rest of the distribution, from:
  $$\mu_i = \exp[\text{Log}(y_i)]$$
  - ZIP: $\text{Mean (original } y_i) = (1 - \omega_i)\mu_i$
  - ZIP: $\text{Variance(original } y_i) = \mu_i + \frac{\omega_i}{(1-\omega_i)}\mu_i^2$
  - ZINB: $\text{Mean (original } y_i) = (1 - \omega_i)\mu_i$
  - ZINB: $\text{Variance(original } y_i) = \mu_i + \left[\frac{\omega_i}{(1-\omega_i)} + \frac{k}{1-\omega_i}\right]\mu_i^2$

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: $\mathrm{Logit}(y_i = 0) = \beta_{01} + \beta_{11}X_i + \beta_{21}Z_i$
  - 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: $\mathrm{Log}(y_i | y_i > 0) = \beta_{02} + \beta_{12}X_i + \beta_{22}Z_i$
  - Predict rest of **positive** counts using Link = Log, Distribution = ZAP or ZANB

- These models are not readily available in SAS, but NBH is in Mplus
  - Could be fit in SAS NLMIXED (as could ZIP/ZINB)

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)
  - 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: $\mathrm{Logit}(\mathrm{y_i} > 0) = \ \upbeta_{01} + \upbeta_{11}\mathrm{X_i} + \upbeta_{21}\mathrm{Z_i}$
  - Predict being **<u>any not 0</u>** using Link = Logit, Distribution = Bernoulli
  - Usually do specify predictors for this part

- Submodel 2: $(\mathrm{y_i}|\mathrm{y_i} > 0) = \upbeta_{02} + \upbeta_{11}\mathrm{X_i} + \upbeta_{21}\mathrm{Z_i}$
  - 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 not readily available in SAS, but is in Mplus
  - Can be estimated as a multivariate model in SAS GLIMMIX or NLMIXED
  - 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 → Negative Binomial
  - Continuous: Normal, Log-Normal, Gamma

- What kind of **If 0** do you want to predict?
  - Discrete: Extra 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 Models Part 2: 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 random effects at each iteration)
  - But there are fewer ready-made options for modeling differential variance/covariance across DVs (no easy R matrix structures in true ML)

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