Generalized Linear Models for Non-Normal Data

- Today's Class:
  - **3 parts of a generalized model**
  - Models for binary outcomes
  - Complications for generalized multivariate or multilevel models

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)

- **General Linear Models:** conditionally normal outcome distribution, **fixed effects** (identity link; only one dimension of sampling)

Note: Least Squares is only for GLM

- **General*ized* 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)
- **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 of DV in a linear combination of (effect*predictor) + (effect*predictor)…

Generalized Linear Models

- **Generali*zed* linear models:** link-transformed conditional mean of y_{ti} is predicted instead; ML estimator uses not-normal distributions to calculate the likelihood of the outcome data
  - **Level-1** conditional outcomes follow some not-normal distribution that may not have a residual variance, but level-2 random effects are 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)

    (These two may get grouped together as "multinomial")
  - 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_{ti} ⟵ **2. Link Function** = **3. Linear Predictor of Fixed and Random Effects**

1. <u>Non-normal conditional distribution of y_{ti}:</u>

- General MLM uses a *normal* conditional distribution to describe the y_{ti} 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 y_{ti}, so the 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 residual variance (e.g., binary → Bernoulli, count → Poisson)

3 Parts of Generalized (Multilevel) Models

1. Non-Normal Conditional Distribution of y_{ti} ⇐ **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_{ti} scale
   - Predicted outcomes (found via inverse link) will then stay within bounds
   - e.g., binary outcome: conditional mean to be predicted is probability of a 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_{ti} * 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_{ti} ⟵ **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

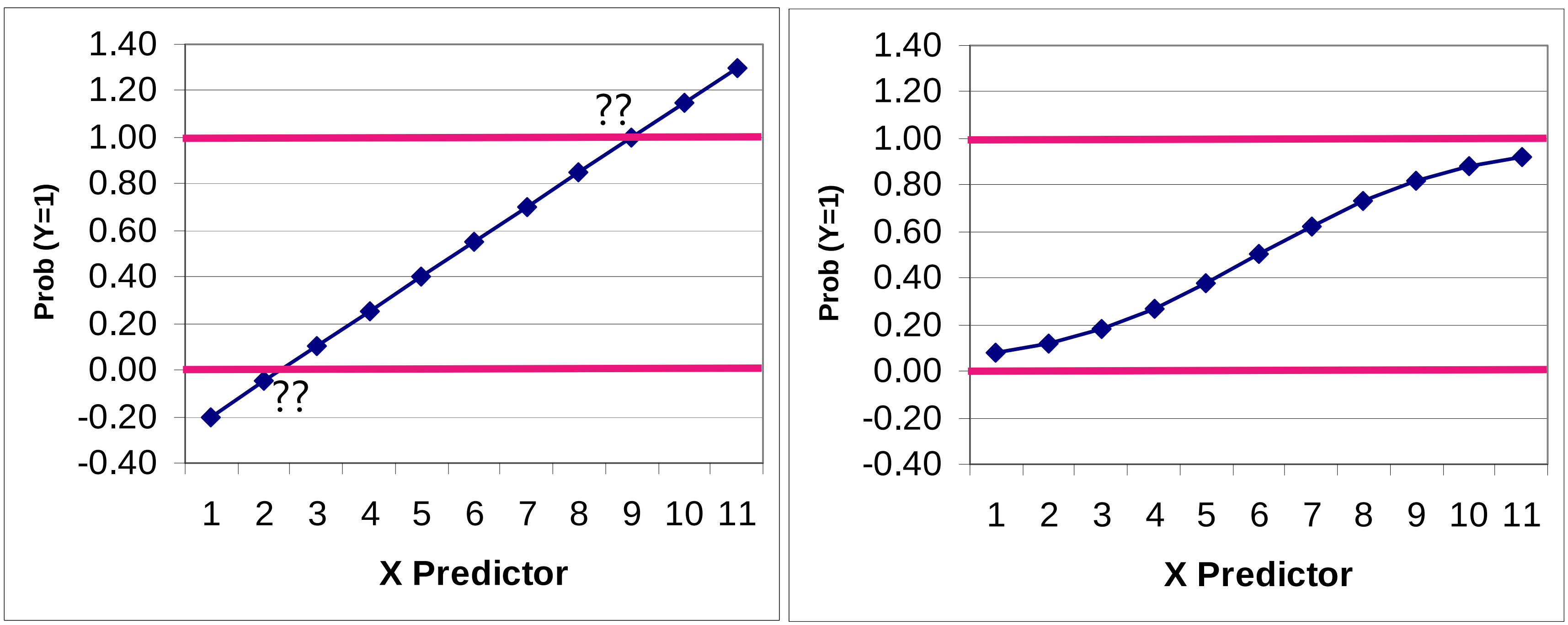
Normal GLM for Binary Outcomes?

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

Normal GLM for Binary Outcomes?

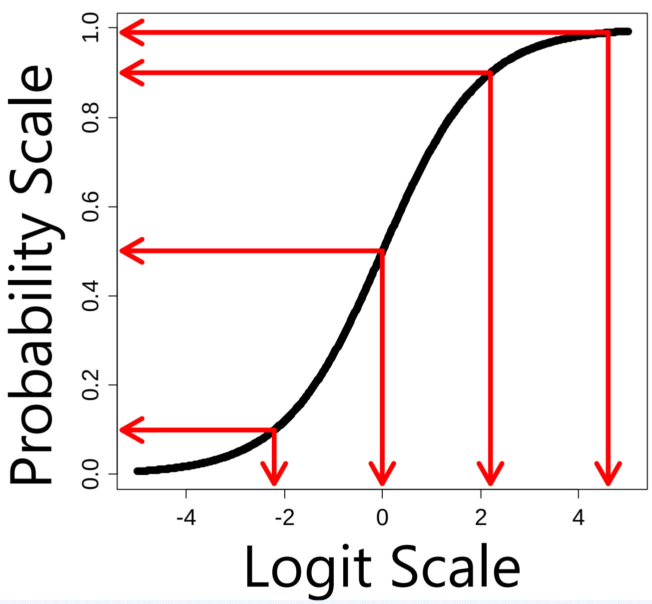
- Problem #1: A **linear** relationship between X and Y???
- Probability of a 1 is bounded between 0 and 1, but predicted probabilities from a linear model aren't going to be bounded
- Linear relationship needs to shut off → made nonlinear

We have this... **But we need this...**



Generalized Models for Binary Outcomes

- Solution to #1: Rather than predicting $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1})$ 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 $p(\text{y}_\text{i} = 1) = .7$ then $\text{Odds}(1) = 2.33$; $\text{Odds}(0) = 0.429$
    - But odds scale is skewed, asymmetric, and ranges from 0 to +∞ → Not helpful
  - **Take *natural log of odds ratio* → called "logit" link:** $\mathbf{Log}\left[\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}\right]$
    - If $p(\text{y}_\text{i} = 1) = .7$, then $\text{Logit}(1) = 0.846$; $\text{Logit}(0) = -0.846$
    - 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?

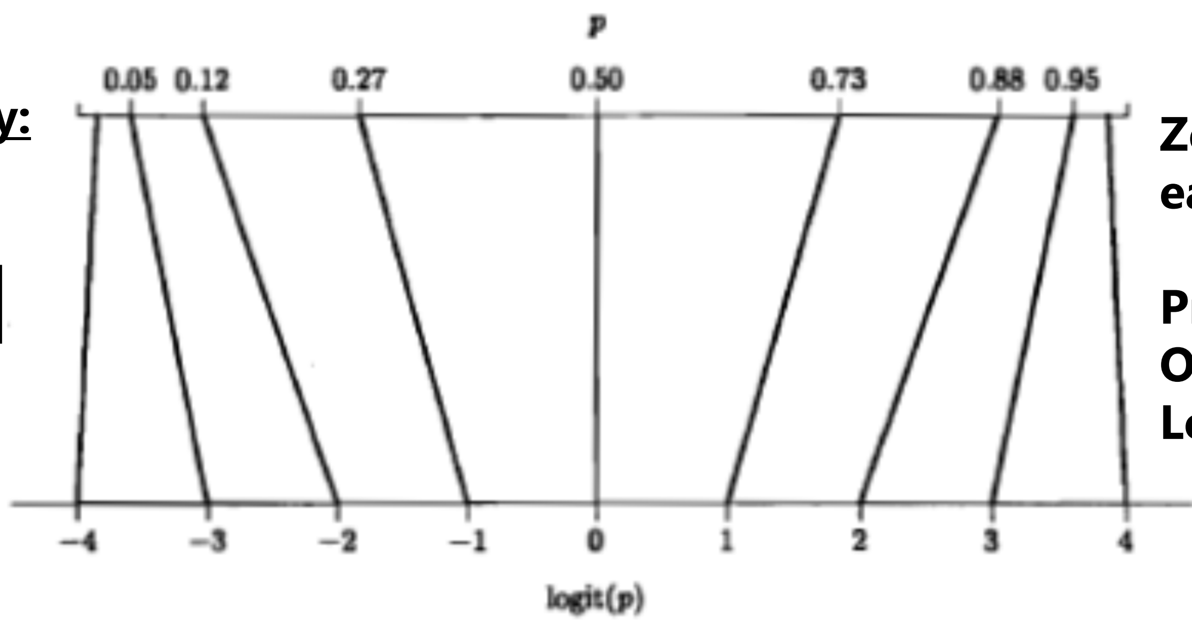
Solution #1: Probability into Logits

- **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: $p(\mathbf{y_i} = 1)$

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

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



Zero-point on each scale:

Prob = .5
Odds = 1
Logit = 0

Normal GLM for Binary Outcomes?

- General linear model: $\boldsymbol{p}(\mathbf{y_i} = \mathbf{1}) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}\mathbf{X_i} + \boldsymbol{\beta_2}\mathbf{Z_i} + \mathbf{e_i}$
- 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)
- <u>Problem #2a</u>: So the residuals can't be normally distributed
- <u>Problem #2b</u>: 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

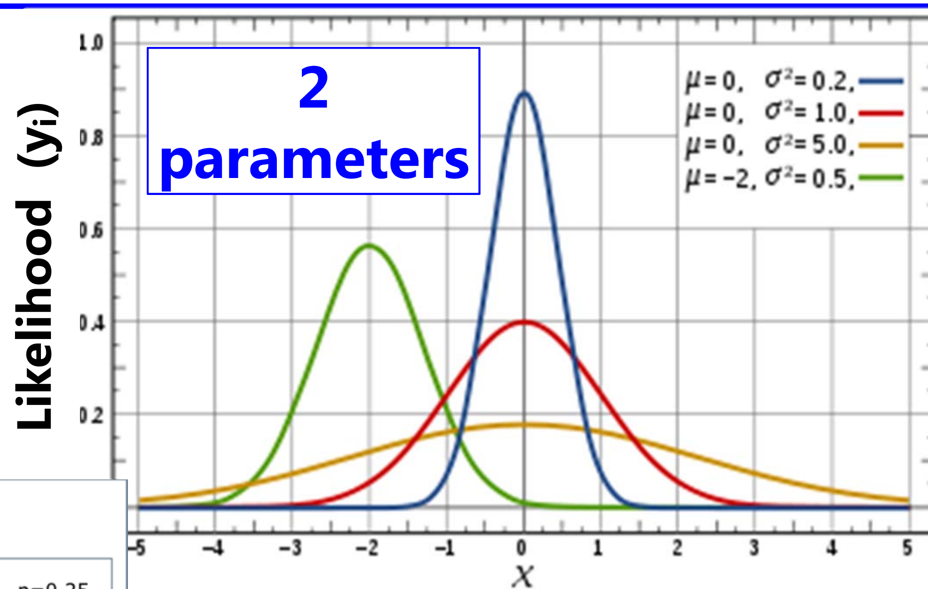
Mean (p)	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
Variance	.0	.09	.16	.21	.24	.25	.24	.21	.16	.09	.0

Solution to #2: Bernoulli Distribution

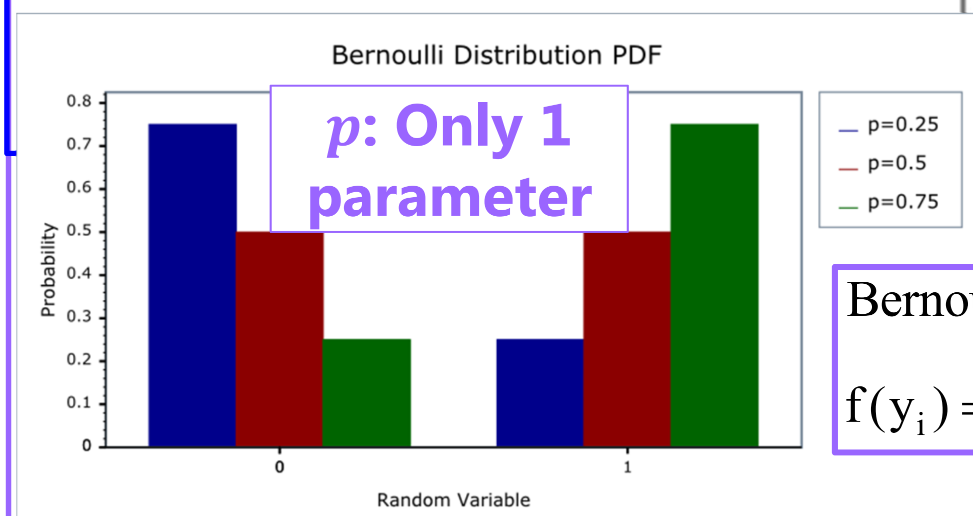
- Rather than using a **normal** conditional outcome distribution, we will use a **Bernoulli distribution** → a special case of a binomial distribution for only one binary outcome

Univariate Normal PDF:

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



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:** $\mathbf{Log}\left[\frac{p(y_i=1)}{1-p(y_i=1)}\right] = \beta_0 + \beta_1 X_i + \beta_2 Z_i$ ← **g(·) 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:** $p(y_i = 1) = \frac{\exp(\beta_0+\beta_1 X_i+\beta_2 Z_i)}{1+\exp(\beta_0+\beta_1 X_i+\beta_2 Z_i)}$ ← **$g^{-1}(\cdot)$ inverse link**

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

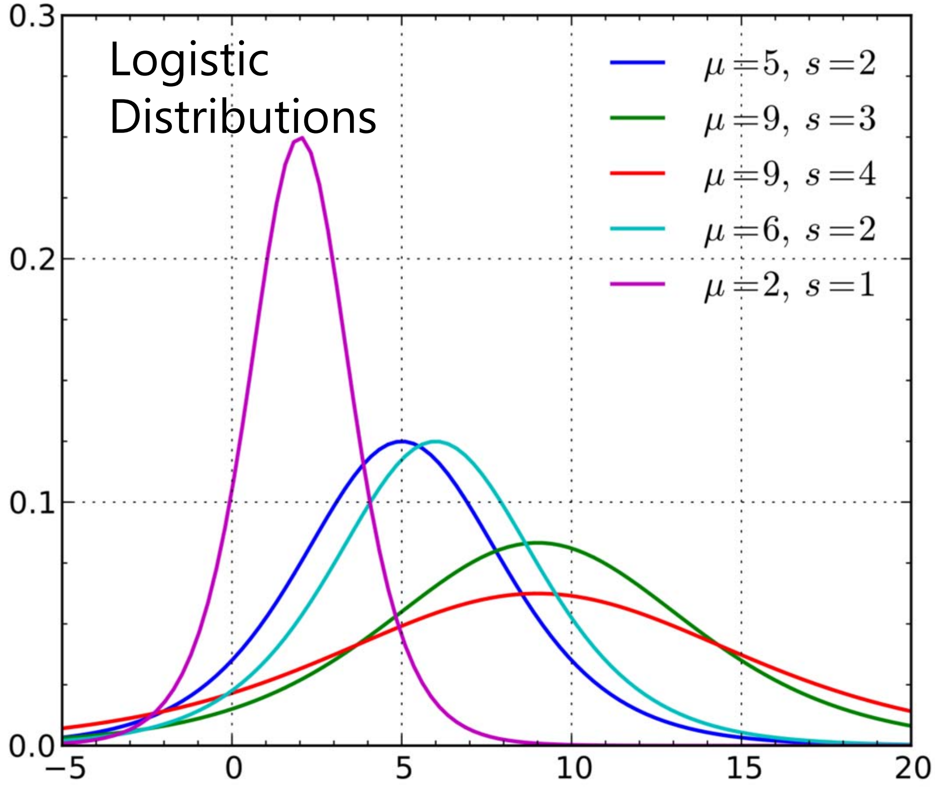
"Logistic Regression" for Binary Data

- 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 $\mathbf{y_i^*}$**, then

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

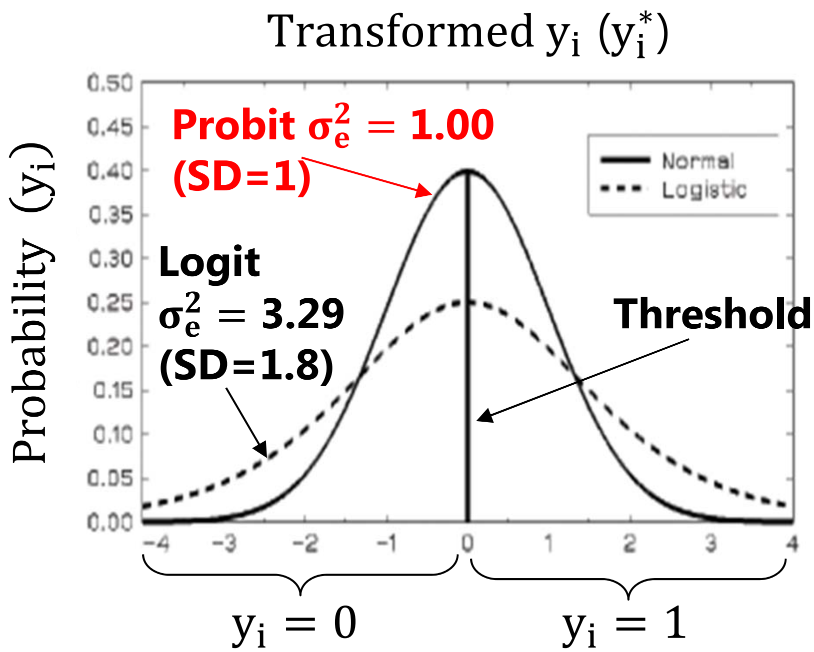
<u>Logistic Distribution:</u>

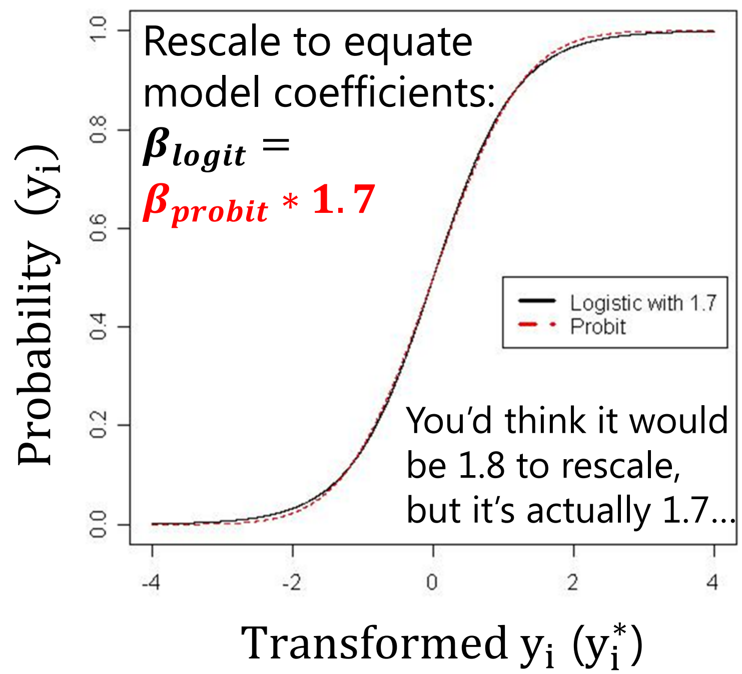
Mean = μ, Variance = $\frac{\pi^2}{3}s^2$, where *s* = scale factor that allows for "over-dispersion" (must be fixed to 1 in binary outcomes for identification)

Other Link Functions for Binary Data

- The idea that a "latent" continuous variable underlies an observed binary response also appears in a **Probit Regression** model:
  - A ***probit*** link, such that now your model predicts a different transformed Y_p:
    $\text{Probit}(\text{y}_\text{i} = 1) = \Phi^{-1}[p(\text{y}_\text{i} = 1)] = \textit{your model}$ ⟵ $\mathbf{g}(\cdot)$
    - Where Φ = standard normal cumulative distribution function, so the transformed y_i is the **z-score** that corresponds to the value of cumulative standard normal distribution **below** which the conditional mean probability is found
    - Inverse link requires integration to find probability → $p(\text{y}_\text{i} = 1) = \Phi^{-1}(z)$
  - Same Bernoulli distribution for the conditional binary outcomes, in which residual variance cannot be separately estimated (so no e_i in the model)
    - Probit also predicts "latent" response: $\text{y}_\text{i}^* = \text{threshold} + \text{ your model} + \text{e}_\text{i}$
    - But Probit says $\text{e}_\text{i} \sim \text{Normal}(0, \sigma_\text{e}^2 = 1.00)$, whereas Logit $\sigma_\text{e}^2 = \frac{\pi^2}{3} = 3.29$
  - 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.

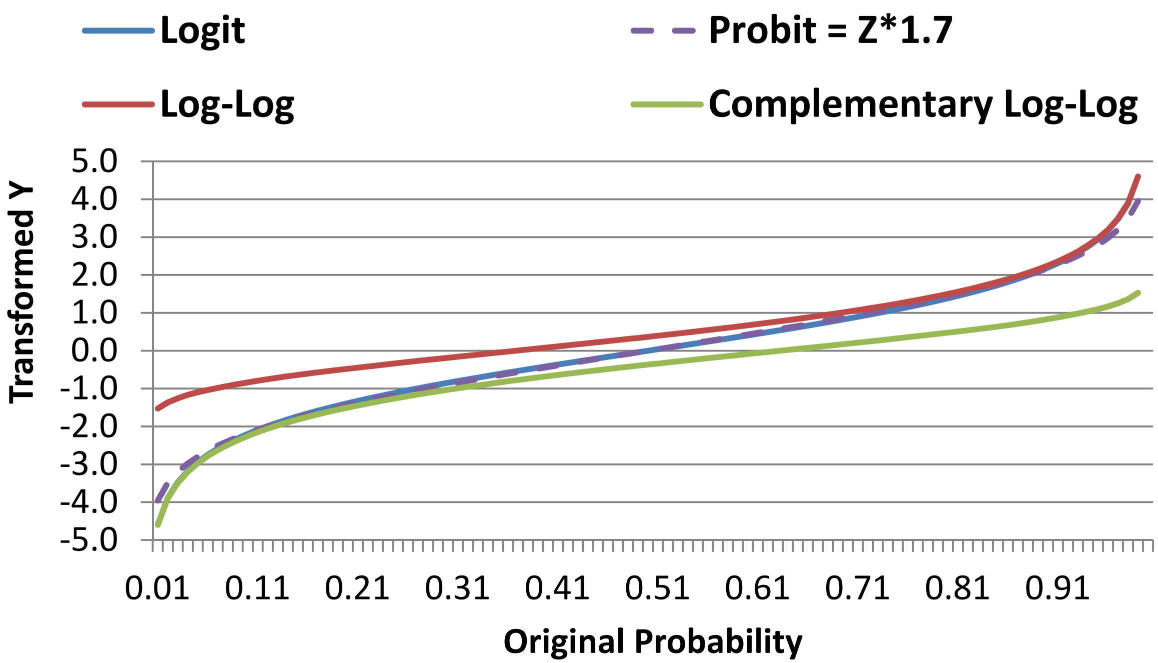




- Other fun facts about probit:
  - Probit = "ogive" in the Item Response Theory (IRT) world
  - Probit has no odds ratios (because it's not based on odds)

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

Other Models 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):	$p = \frac{\exp(\mu)}{1+\exp(\mu)}$	$p = \Phi^{-1}(\mu)$	$p = \exp[-\exp(-\mu)]$	$p = 1 - \exp[-\exp(\mu)]$
			$e_i \sim \text{log-Weibull extreme value}\left(0.577, \sigma_e^2 = \frac{\pi^2}{6}\right)$	
In SAS LINK=	LOGIT	PROBIT	LOGLOG	CLOGLOG

Generalized MLM: Summary

- Statistical models come from probability distributions
  - Conditional outcomes are assumed to have some distribution
  - The normal distribution is one choice, but there are lots of others: so far we've seen Bernoulli (and mentioned log-Weibull)
  - ML estimation tries to maximize the height of the data using that chosen 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_{ti} 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

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:
  - Alternative conditional outcome distribution used (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 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:** **Logit [*p*(y_{ti} = 1)] = β_{0i}**
- **Level 2:** **$\beta_{0i} = \gamma_{00} + U_{0i}$**
- **Composite:** **Logit [*p*(y_{ti} = 1)] = $\gamma_{00} + U_{0i}$**

Notice what's NOT in level 1…

- σ_e^2 residual variance is not estimated → **$\pi^2/3 = 3.29$**
  - (Known) residual is in model for actual y_{ti}, so $\sigma_e^2 = 3.29$ 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, not estimated residual variance
- Have not seen equivalent ICC formulas for other outcomes besides binary!

Random Linear Time Model for Binary Outcomes

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

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

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 for normal conditional 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

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

  - **Level-1 predictors cannot decrease the residual variance** like usual, so all other models estimates have to go up 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

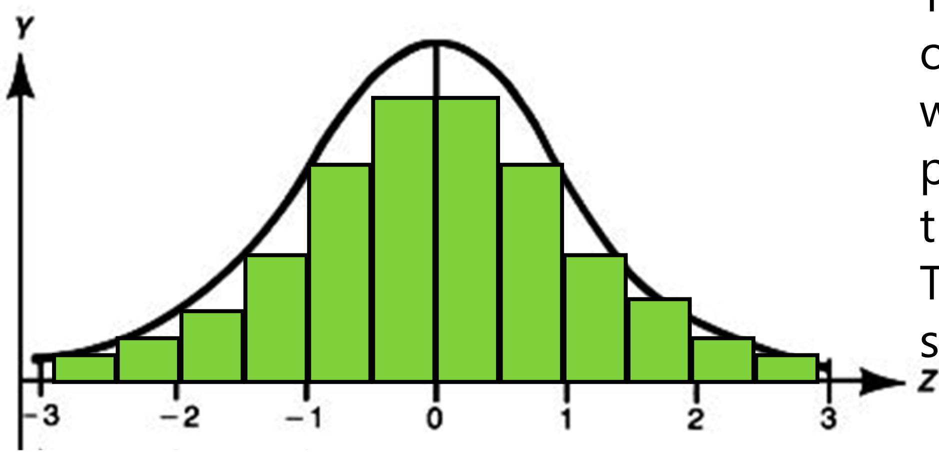
- **Quasi-Likelihood methods** → older methods
  - "Marginal QL" → approximation around fixed part of model
  - "Penalized QL" → approximation around fixed + random parts
  - These both underestimate variances (MQL more so than PQL)
  - 2nd-order PQL is supposed to be better than 1st-order MQL
  - QL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

	U_{0i} = -2	**Logit(y_{ti})**	**IF y_{ti}=1 Prob**	**IF y_{ti}=0 1-Prob**	**Likelihood if both y=1**	**Theta prob**	**Theta width**	**Product 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 (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 set
  - 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 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)