Introduction to Generalized Models

- Today's topics:
  - The big picture of generalized models
  - Review of maximum likelihood estimation
  - Models for binary outcomes
  - Models for proportion outcomes
  - Models for categorical outcomes

  - More models for non-normal data coming later, time permitting…

Why review Generalized Models?

Outcome Type → ***Model Family Name***	**Observed Predictor X**	**Latent Predictor X**
Continuous Y → *"General Linear Model"*	Linear Regression	Confirmatory Factor Models
Discrete/categorical Y → *"Generalized Linear Model"*	Logistic/Multinomial Regression	Item Response and Item Factor Models

- The basis of Item Response Theory (IRT) and Item Factor Analysis (IFA) lies in models for discrete outcomes, which are called "genera***lized***" models
- Thus, IRT and IFA will be easier to understand after reviewing concepts from generalized models…

The Two Sides of Any Model

- **Model for the Means:**
  - *Aka* **Fixed Effects**, Structural Part of Model
  - What you are used to **caring about for testing hypotheses**
  - How the expected outcome for a given observation varies as a function of values on predictor variables
  - People with the same values on the model predictors get the same predicted outcome (i.e., they share a "**conditional mean**")

- **Model for the Variance:**
  - *Aka* **Random Effects and Residuals**, Stochastic Part of Model
  - What you are used to **making assumptions about** instead
  - How residuals are distributed and related across observations (persons, groups, time, etc.) → these relationships are called "dependency" and ***this is how MLM differs from GLM***

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) → **MLM**

Note: Least Squares is only for GLM

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

The Two Sides of a General Model

$$y_i = \boxed{\beta_0 + \beta_1 X_i + \beta_2 Z_i + \beta_3 X_i Z_i} + \boxed{e_i}$$

Our focus this week

- **Model for the Means (Predicted Values):**
  - Each person's expected (predicted) outcome is a weighted linear function of his/her values on X and Z (and here, their interaction), each measured once per person (i.e., this is a general linear model)
  - Estimated parameters are called fixed effects (here, β_0, β_1, β_2, and β_3)

- **Model for the Variance ("Piles" of Variance):**
  - $e_i \sim N(0, \sigma_e^2)$→ ONE residual (unexplained) deviation
  - e_i has a mean of 0 with some estimated constant variance σ_e^2, is normally distributed, is unrelated to X and Z, and is unrelated across people (across all observations, just people here)
  - **Estimated parameter is residual variance only in above GLM, but the question is, what else could it be besides the usual e_i?**

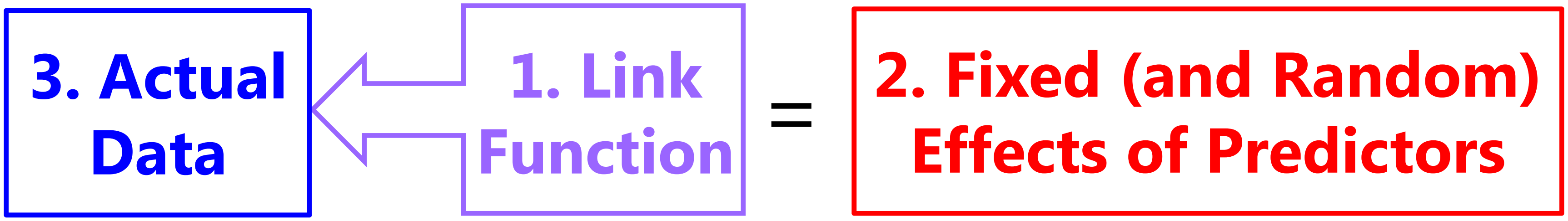
Genera*lized* Models

- **Genera*lized* linear models:** link-transformed conditional mean of Y is predicted instead of actual Y; ML uses not-normal distributions
    - **Single-level models →** residuals follow some not-normal distribution
    - **Multilevel/factor models →** level-1 residuals follow some not-normal distribution, but level-2 random effects or factor scores are almost always still multivariate normal

- Many kinds of non-normally distributed outcomes have some kind of generalized linear model for them using **maximum likelihood**:
    - 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

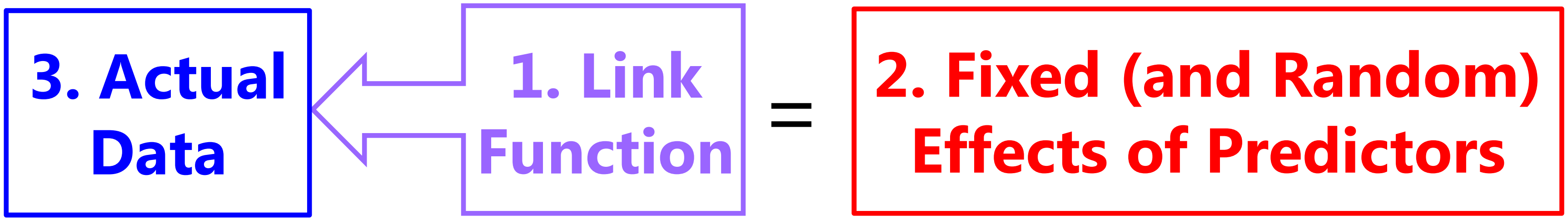
1. Link Function

=

2. Fixed (and Random) Effects of Predictors

1. **Link Function** (different from general): How the conditional mean of a non-normal outcome is made **unbounded** so that the model fixed and random effects can predict it 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 genera*lized* linear models, and general linear mixed models are a special case of genera*lized* linear mixed models

3 Parts of Generalized Multilevel Models



3. Actual Data

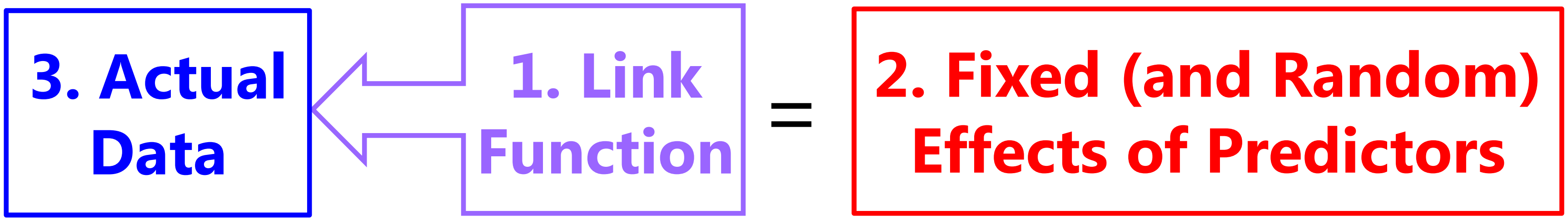
1. Link Function

=

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 can then convert back into the scale of the original outcome
    - That way we can still use the familiar "one-unit change" language to describe the effects of model predictors
    - You can think of this as "model for the means" still, but it would also include 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 the level-1 residuals should be distributed given the sample space (possible values) 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 review how ML would use a normal residual distribution, then examine models for **binary data** to illustrate these 3 parts…

End Goals of Maximum Likelihood Estimation

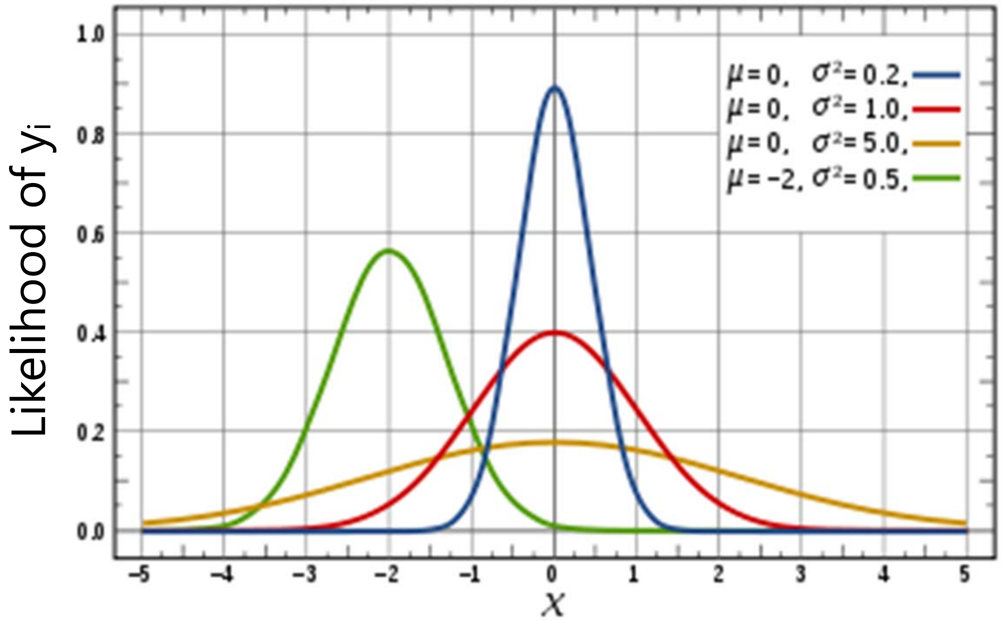
1. Obtain “most likely” values for each unknown model parameter (fixed effects, variances of residuals, and any random effects variances and covariances) → **the estimates**

2. Obtain an index as to how likely each parameter value actually is (i.e., “really likely” or pretty much just a guess?) → **the standard error (SE) of the estimates**

3. Obtain an index as to how well the model we’ve specified actually describes the data → **the model fit indices**

How does all of this happen? Probability distributions!
(i.e., probability density functions, or PDFs)

Univariate Normal Distribution



Univariate Normal PDF:

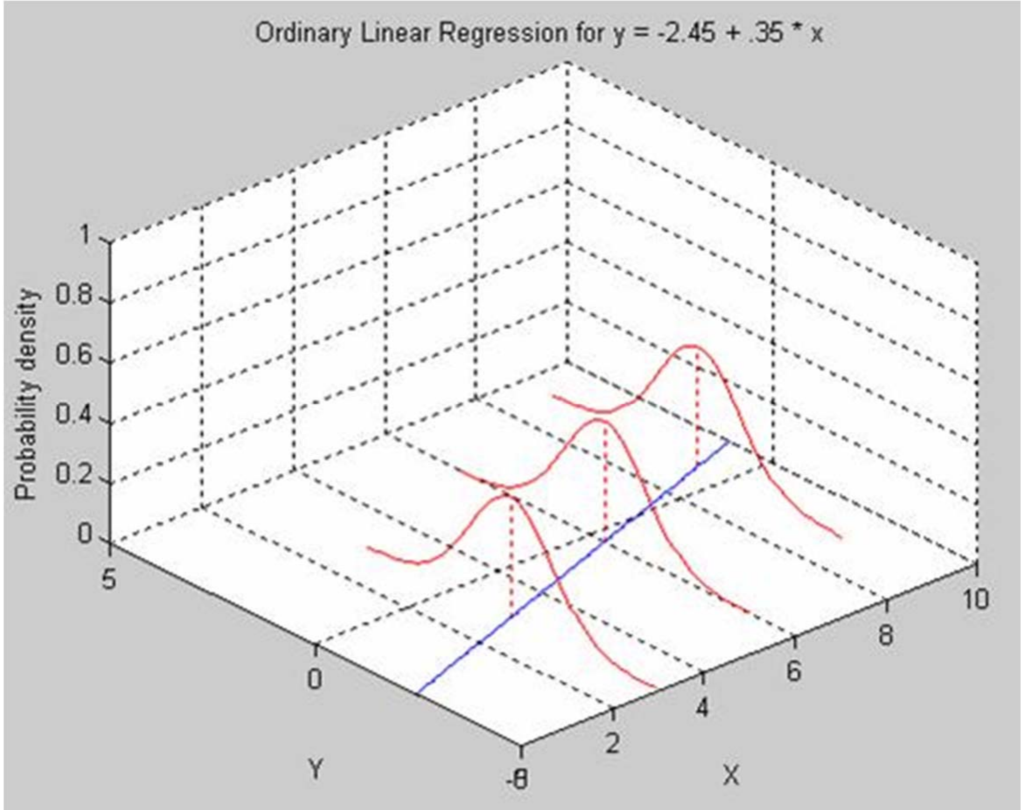
$$f(y_i) = \frac{1}{\sqrt{2\pi\sigma_e^2}} * \exp\left[-\frac{1}{2} * \frac{\left(y_i - \hat{y}_i\right)^2}{\sigma_e^2}\right]$$

Sum over persons for log of f(yi) =
Model Log-Likelihood → Model Fit

- This PDF tells us how **likely** any value of y_i is given two pieces of info:
  - Conditional mean $\hat{y}_i$
  - residual variance σ_e^2
- We can see this work using the NORMDIST function in excel!
  - Easiest for empty model:

    $$y_i = \beta_0 + e_i$$
- We can check our math via SAS PROC MIXED!

Conditional Univariate Normal



Univariate Normal PDF:

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

This function applies for any value of X, such as in regression:

- Fixed effects (intercept, predictor slopes) create a conditional mean for each person, $\hat{y}_i$
- We assume the same residual variance σ_e^2 holds for all values of $\hat{y}_i$

$$y_i = \beta_0 + \beta_1 X_i + e_i$$

$$\hat{y}_i = \beta_0 + \beta_1 X_i$$

$$e_i = y_i - \hat{y}_i \qquad \sigma_e^2 = \frac{\sum_{i=1}^{N} e_i^2}{N-2}$$

Try, try, then try again…

- The best possible answers for the model parameters (e.g., fixed effects and residual variance) can be calculated via least squares given certain ideal circumstances:
  - Complete data, normally distributed residuals with constant variance, and only one dimension of sampling (i.e., single-level, univariate model)
- For almost all other analyses, the best possible estimates of these parameters have to be searched for iteratively
  - Different algorithms are used to decide which values to try given that each parameter has its own distribution of possible values → like an uncharted mountain in which each parameter to find has its own dimension (partial)
  - Calculus helps the program scale this multidimensional mountain
    - At the top, all first partial derivatives (linear slopes at that point) ≈ 0
    - Positive first partial derivative? Too *low*, try again.
    - Negative first partial derivative? Too *high*, try again.
    - Matrix of partial first derivatives = "score function" = "gradient" (as in NLMIXED output for models with truly nonlinear effects)

End Goals 1 and 2: Model Estimates and SEs

- Process terminates (the model "converges") when the next set of tried parameter values don't improve the LL very much…
  - e.g., SAS default convergence criteria = .00000001
  - Those are the values for the parameters that, relative to the other possible values tried, are "most likely" → the estimates

- But we need to know how trustworthy those estimates are…
  - Precision is indexed by the steepness of the multidimensional mountain, where steepness → more negative partial second derivatives
  - Matrix of partial second derivatives = "Hessian matrix"
  - Hessian matrix *−1 = "information matrix"
  - So steeper function = more information = more precision = smaller SE

$$\text{Each parameter SE} = \frac{1}{\sqrt{\text{information}}}$$

End Goal #3: How well does the model fit?

- **Relative model fit** is indexed by a "**deviance**" statistic → **−2LL**
  - **−2LL indicates BADNESS of fit, so smaller values = better models**
  - Given as −2 log likelihood in SAS, SPSS, but given as LL instead in Mplus

- **Nested models are compared using their deviance values: −2ΔLL Test** (i.e., Likelihood Ratio Test, Deviance Difference Test)

1. Calculate −2ΔLL: $(-2LL_{fewer}) - (-2LL_{more})$
2. Calculate Δdf: $(\#\ Parms_{more}) - (\#\ Parms_{fewer})$
3. Compare −2ΔLL to χ^2 distribution with df = Δdf (use CHIDIST in excel for *p*-value)

1. & 2. must be positive values!

- Nested or non-nested models can also be compared by **Information Criteria** that reflect **−2LL** AND # parameters used and/or sample size
  - **AIC** = Akaike IC = **−2LL** + 2 *(#parameters)
  - **BIC** = Bayesian IC = **−2LL** + log(N)*(#parameters) → penalty for complexity
  - No significance tests or critical values, just "smaller is better"

Testing Significance of Model Effects

- For random effects (variances, covariances) you must use a −2LL (likelihood ratio) test to assess significance

- For <u>single</u> fixed effects, you can examine the *p*-value on the output created from the Wald test: $\text{test statistic} = \text{Est} / \text{SE}$
  - Test: SAS uses a *t*-distribution; Mplus uses *z* (infinite denominator df)

- For <u>multiple</u> fixed effects, you can compare nested models using −2LL (likelihood ratio) test
  - Add parameters? Model can get BETTER or NOT BETTER
  - Remove parameters? Model can get WORSE or NOT WORSE
  - You can also use the <u>CONTRAST</u> statement to provide a multivariate Wald test of multiple fixed effects (my favorite new trick—stay tuned!)

Software for Generalized Models

- SAS for single-level generalized models
  - PROC LOGISTIC or PROC PROBIT for binary data
  - PROC GENMOD for categorical and some continuous data
  - PROC FMM for lots of things (include hurdle models!)
  - PROC QLIM or PROC LIFEREG for censored data (tobit; cut-off data)
- SAS for multilevel multivariate generalized models
  - PROC GLIMMIX is newest and easiest to use
  - PROC NLMIXED allows user-defined custom models with lots of code
- Mplus for either type of generalized model
  - CATEGORICAL for binary/ordinal, NOMINAL for unordered categories, COUNT for discrete data, TWOPART for two-part models, CENSORED for cut-off data, DSURVIVAL for discrete-time survival data

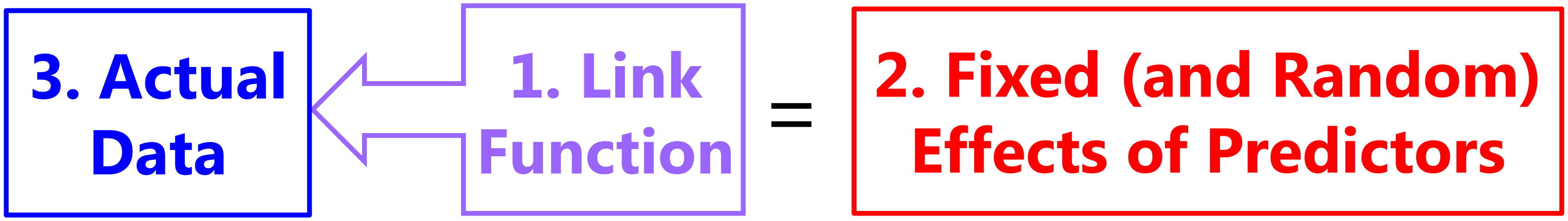
Estimation for Generalized Models

- Maximum likelihood estimation is the gold standard, but only recently has it become computationally feasible for all models
  - Previous approaches are band-aids at best and should not be used if possible, but they still exist in software so you should be careful

- Here is what you want to look for in the SAS PROCs:
  - GLIMMIX: **METHOD = QUAD, LAPLACE** → these are true maximum likelihood estimators that permit −2LL tests and give good estimates

- Here is what you want to avoid in the SAS PROCS:
  - GLIMMIX: METHOD = RSPL/MSPL (except for normal outcomes, which is then equivalent to REML/ML, respectively), RPML, MMPL
    - These are "quasi" or "pseudo" likelihoods, which are known to have biased variance components and for which −2LL tests are invalid
    - Always check your output to see what SAS did for you by default: For instance, Quasi/pseudo likelihood estimators get invoked if you use the _residual_ option in a RANDOM statement to induce a scale factor or structure an **R** matrix
  - GENMOD: using the REPEATED statement invokes GEE, which is also bad

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 the conditional mean of a non-normal outcome is made **unbounded** so that the model fixed and random effects can predict it linearly

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

3. **Model for Level-1 Residuals** (different than general): how the level-1 residuals should be distributed given the sample space (possible values) of the actual outcome

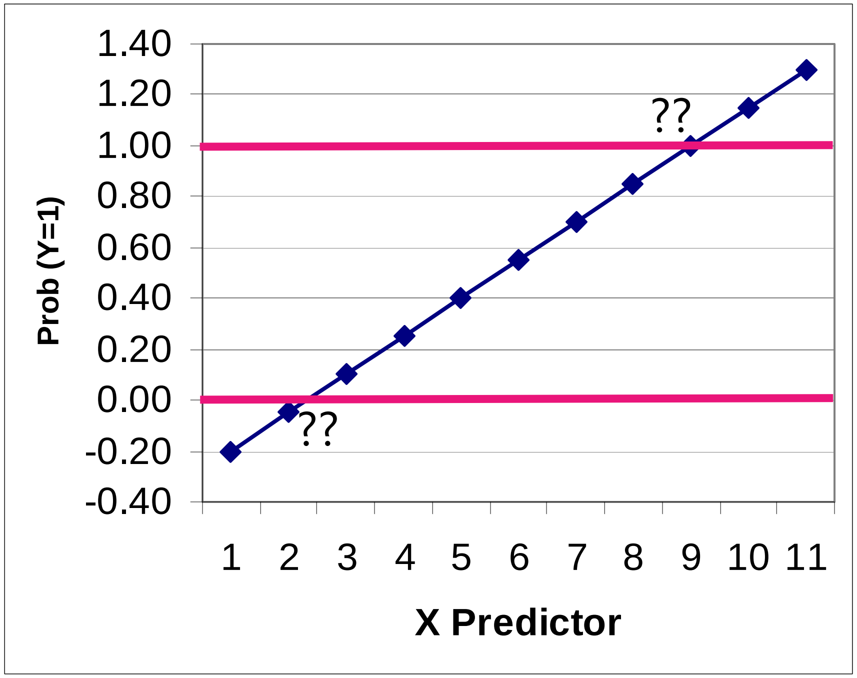
Normal GLM for Binary Outcomes?

- Let's say we have a single binary (0 or 1) outcome…
  - **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 $\mathbf{y_i} = \textbf{(predicted probability of 1)} + \mathbf{e_i}$
  - **What could possibly go wrong?**

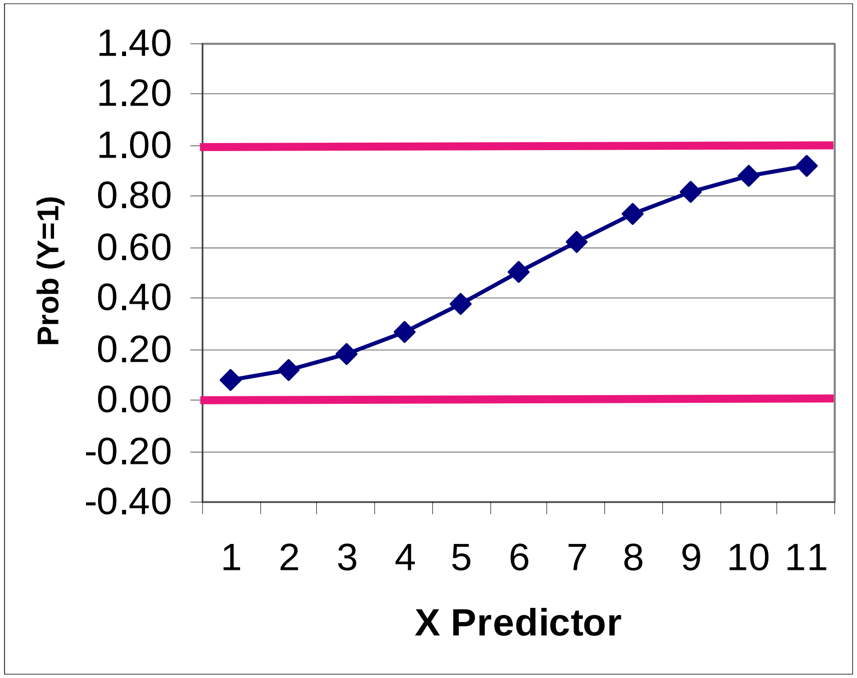
Normal GLM for Binary Outcomes?

- 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…

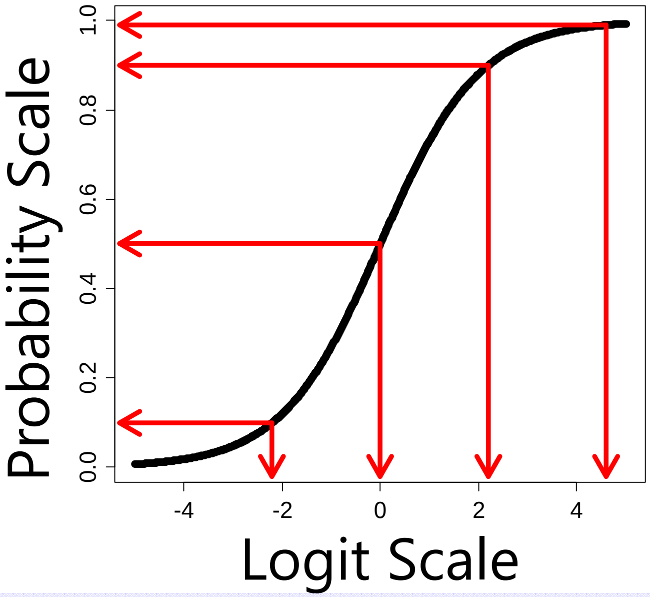


But we need this…



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 $p(\mathrm{y_i} = 1) = .7$ then $\text{Odds}(1) \ = \ 2.33$; $\text{Odds}(0) \ = \ .429$
    - 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 $\text{Logit}(1) \ = \ .846$; $\text{Logit}(0) = -.846$
    - 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

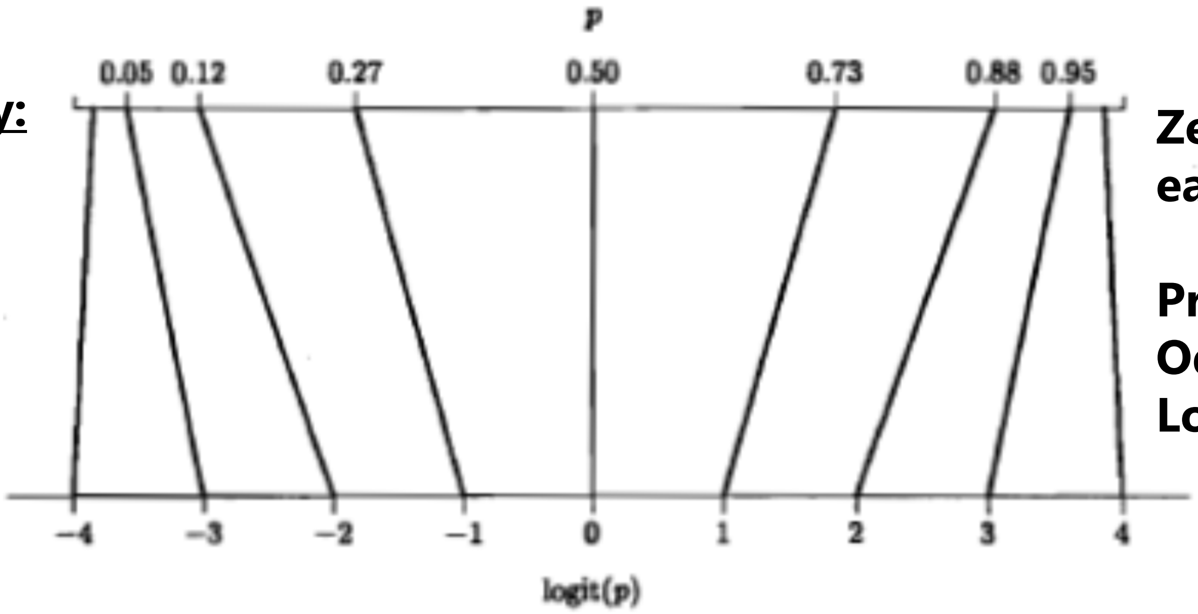
Can you guess what $p(.01)$ would be on the logit scale?

Solution to #1: Probability into Logits

- **A Logit link is a <u>nonlinear</u> transformation of probability:**
  - Equal intervals in logits are NOT equal intervals of probability
  - Logits range from ±∞ and are 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**

<u>Probability:</u>
$\boldsymbol{p(\mathrm{y_i}=1)}$

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



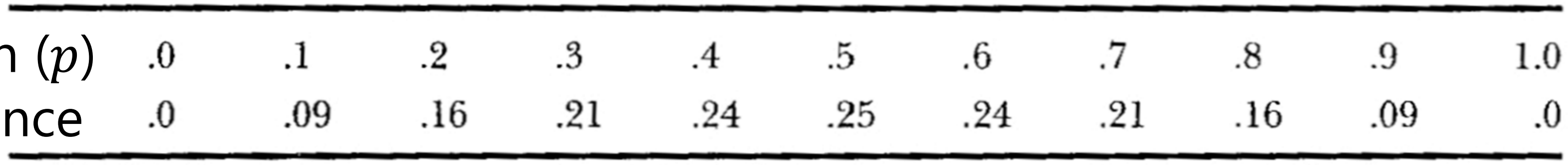
Zero-point on each scale:

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: $\mathbf{e_i} = \mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}$
  - 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



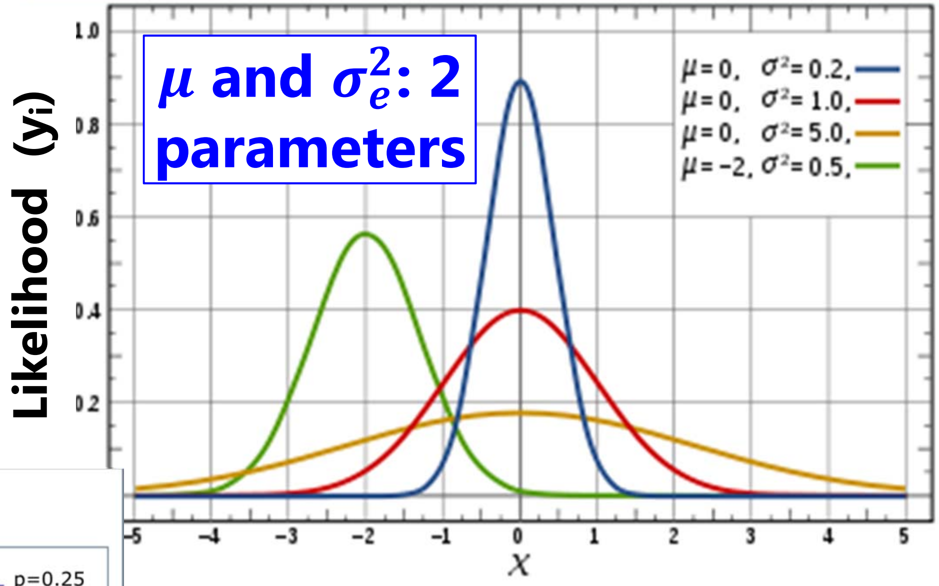
Mean (p)	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
Variance	.0	.09	.16	.21	.24	.25	.24	.21	.16	.09	.0

Solution to #2: Bernoulli Distribution

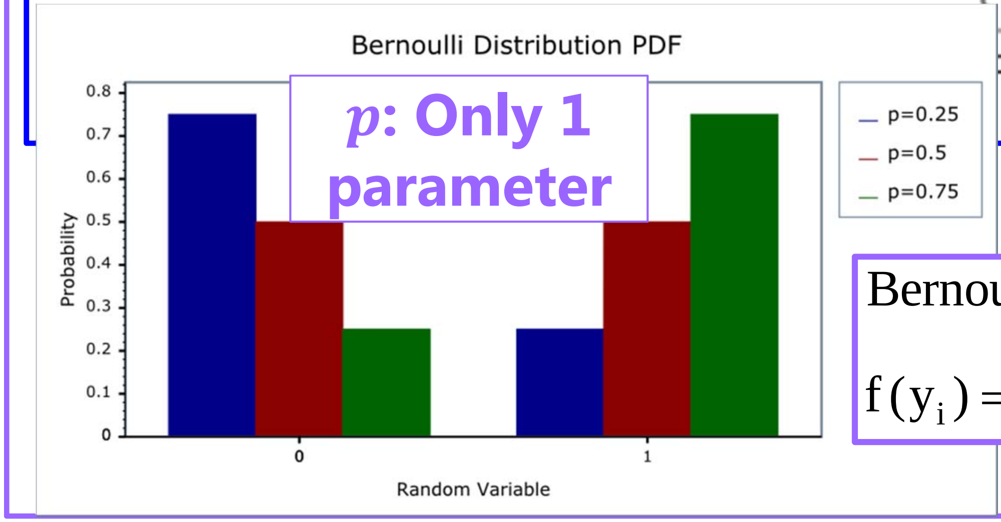
- Instead of a normal residual distribution, we will use a **Bernoulli distribution** → a special case of a binomial for only one outcome

Univariate Normal PDF:

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



μ and σ_e^2: 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:** $\mathbf{Log}\left[\frac{p}{1-p}\right] = \beta_0 + \beta_1 X_i + \beta_2 Z_i \quad \longleftarrow \boxed{g(\cdot)\ \textbf{link}}$
  - 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:** $p(y_i = 1) = \frac{\exp(\beta_0+\beta_1 X_i+\beta_2 Z_i)}{1+\exp(\beta_0+\beta_1 X_i+\beta_2 Z_i)} \quad \longleftarrow \boxed{g^{-1}(\cdot)\ \textbf{inverse link}}$

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

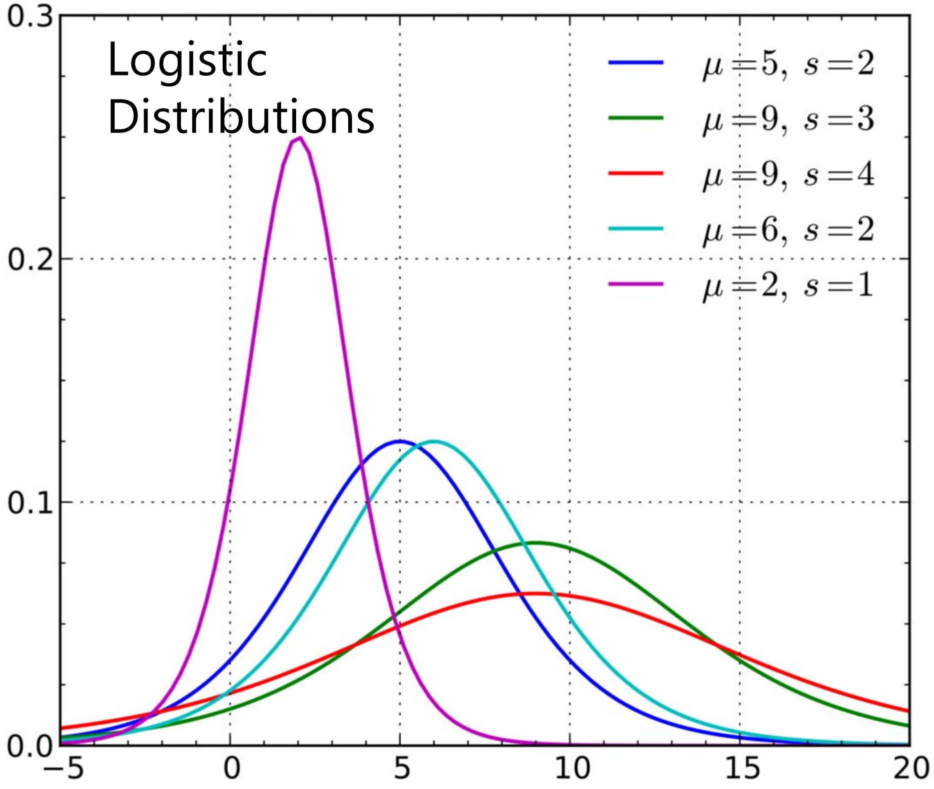
"Logistic Regression" for Binary Data

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

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

$threshold = \beta_0 * -1$ 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

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

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)

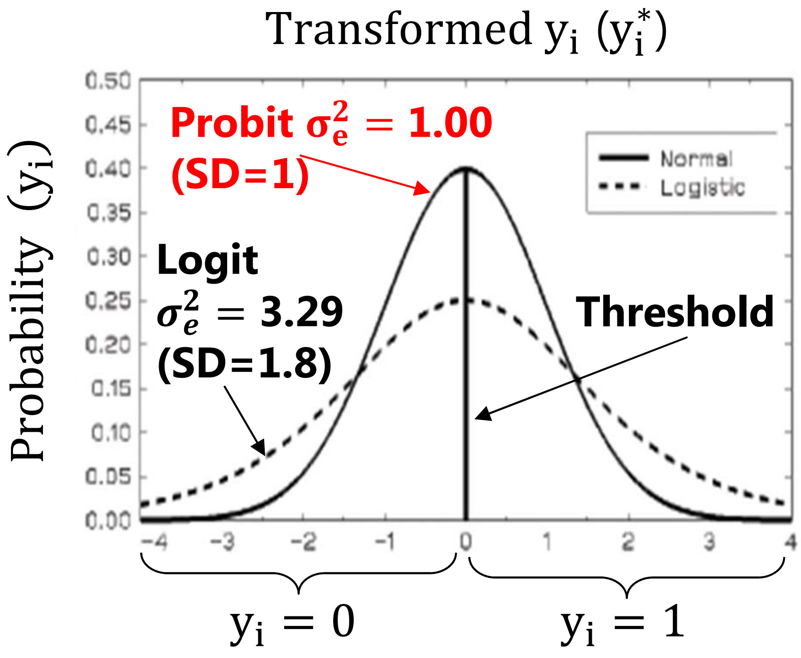
Other Models for Binary Data

- 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:

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

    - Where $\boldsymbol{\Phi}$ = 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 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: $\mathrm{y_i^*} = \mathrm{threshold} + \ \mathrm{your\ model} + \mathrm{e_i}$
    - But Probit says $\mathrm{e_i} \sim \mathrm{Normal}(0, \sigma_e^2 = 1.00)$, whereas 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…

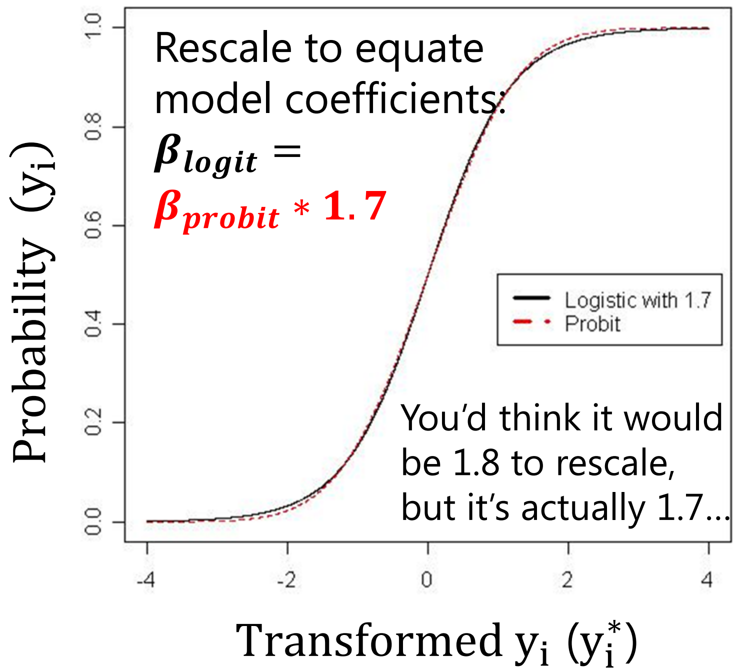
Probit vs. Logit: Should you care? Pry not.



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

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

Threshold



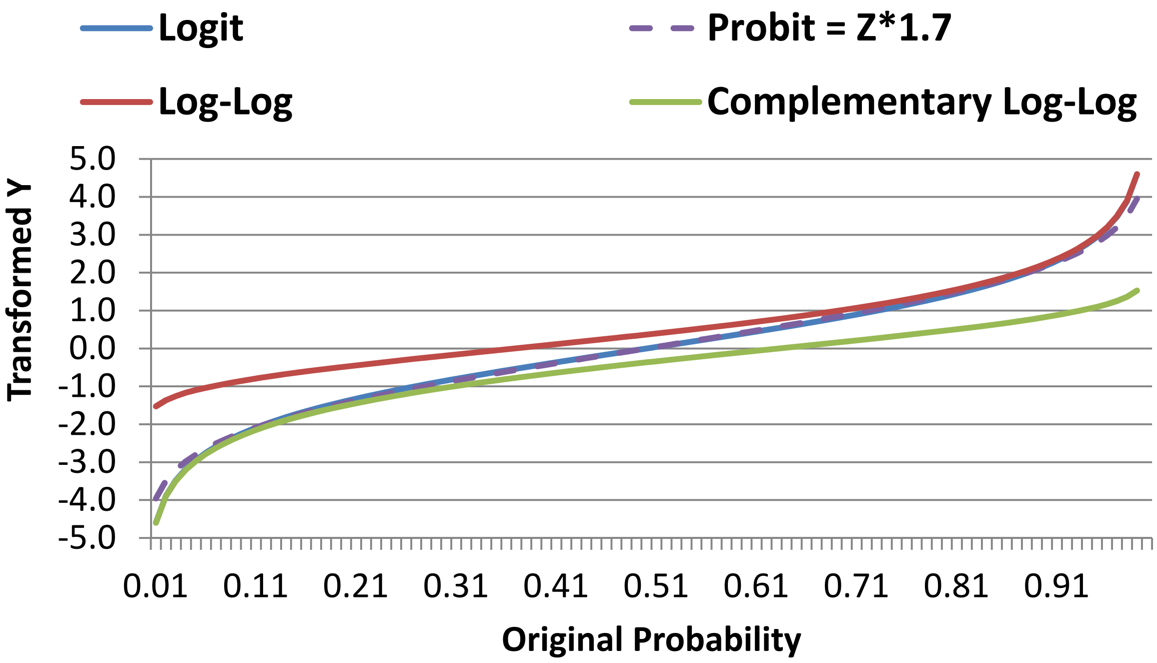
Rescale to equate model coefficients: $\beta_{logit} = \beta_{probit} * 1.7$

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

$\mu = \textbf{model}$	**Logit**	**Probit**	**Log-Log**	**Complement. Log-Log**
$g(\cdot)$ 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

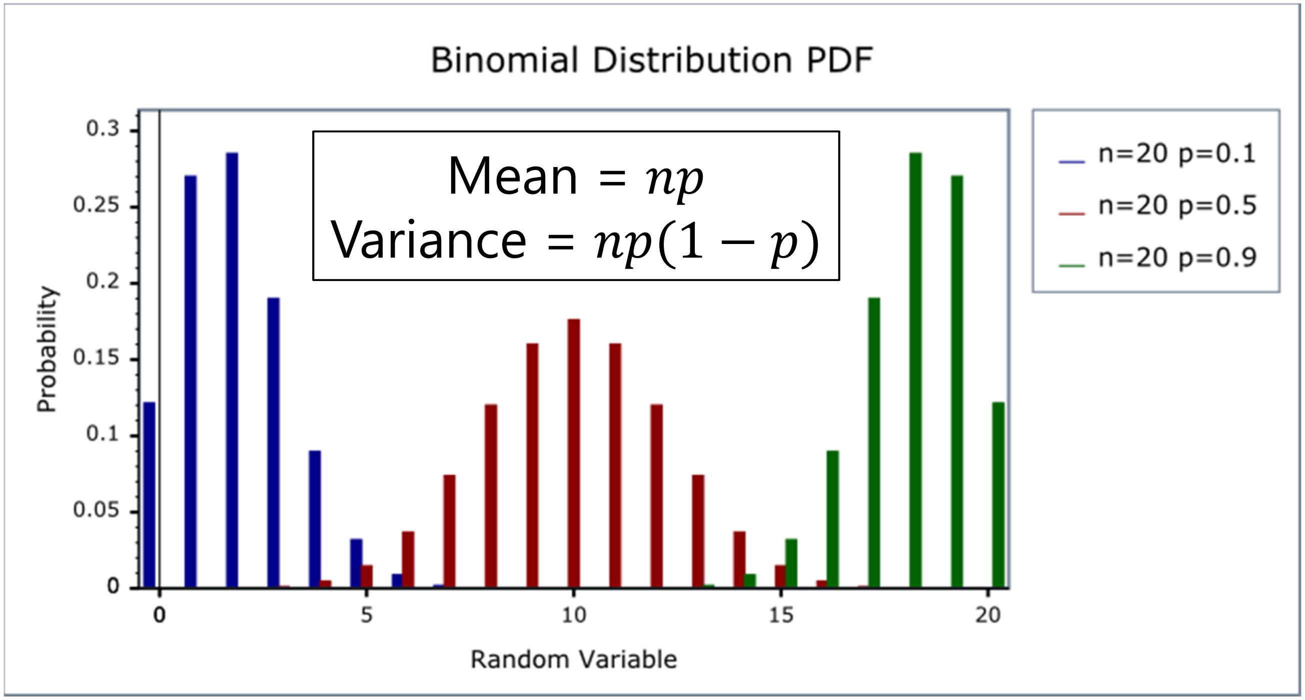
- The logit link can also be useful in predicting proportions:
  - Range between 0 and 1, so model needs to "shut off" predictions for conditional mean as they approach those ends, just as in binary data
  - Data to model: → μ in logits $= \text{Log}\left(\frac{p}{1-p}\right)$ ← $\text{g}(\cdot)$ **Link**
  - Model to data → $p = \frac{\exp(\mu)}{1+\exp(\mu)}$ ← $\text{g}^{-1}(\cdot)$ **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
    - (Not sure if it's ok to cheat by rescaling to fit between 0 and 1)

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

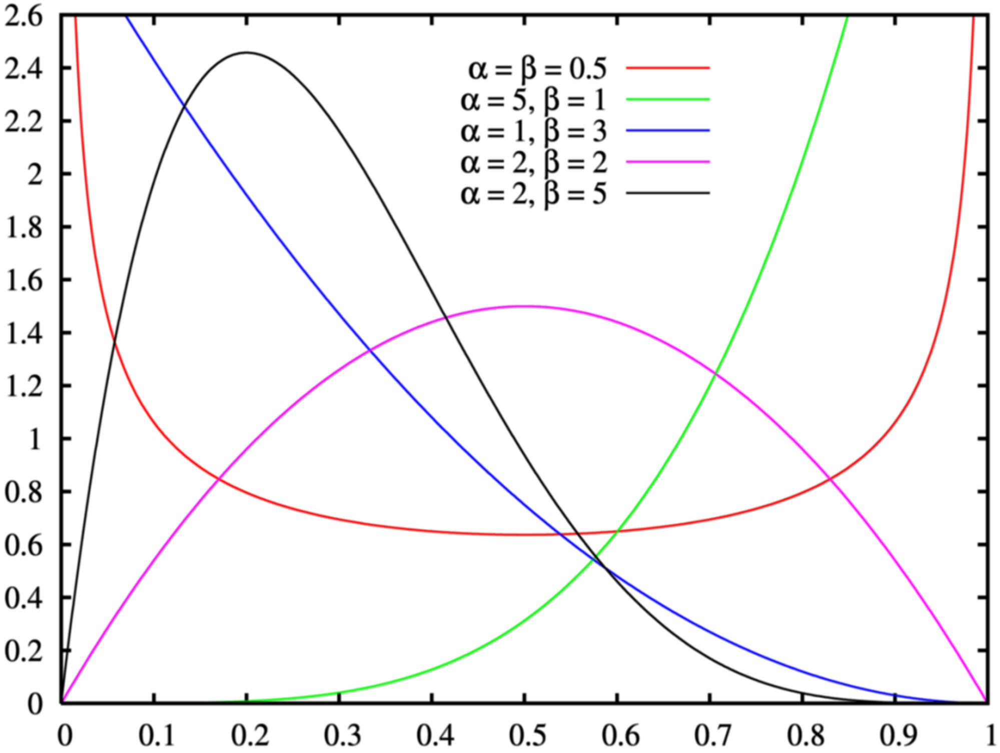
- SAS PROC GLIMMIX allows the outcome variable to be defined as ***#events/#trials*** on MODEL statement
  - LINK=LOGIT so that the conditional mean stays bounded between 0 and 1 as needed (or alternatively, CLOGLOG/LOGLOG)
  - DIST=BINOMIAL so variance (and SEs) are determined by that mean, as they 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 also the basis for many other generalized models for predicting categorical 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 $c = 0,1,2,3$):

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

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

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

$$\boxed{\begin{array}{l}\text{Logit}(y_i > 2) = \beta_{03} \\ \rightarrow p(y_i > 2) = \frac{\exp(\beta_{03})}{1+\exp(\beta_{03})}\end{array}}$$

- 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: $\mathrm{Logit}(\mathrm{y_i} > 0) = \boldsymbol{\beta_{01}} + \beta_1 \mathrm{X_i} + \beta_2 \mathrm{Z_i} + \beta_3 \mathrm{X_i Z_i}$
  - Submodel 2: $\mathrm{Logit}(\mathrm{y_i} > 1) = \boldsymbol{\beta_{02}} + \beta_1 \mathrm{X_i} + \beta_2 \mathrm{Z_i} + \beta_3 \mathrm{X_i Z_i}$
  - Submodel 3: $\mathrm{Logit}(\mathrm{y_i} > 2) = \boldsymbol{\beta_{03}} + \beta_1 \mathrm{X_i} + \beta_2 \mathrm{Z_i} + \beta_3 \mathrm{X_i Z_i}$

- 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 for residuals**, whose PDF for $C = 4$ categories of $c = 0,1,2,3$, an observed $y_i = c$, and indicators I if $c = y_i$

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

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 $0 = \mathrm{ref}$):

    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 0-4 → $\#\,\mathrm{trials} = 4, \#\,\mathrm{correct} = \mathrm{y_i}$
  - 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: we saw 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
     - 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