Introduction to Multivariate General*ized* Linear Models

- Topics:
  - Taxonomy of multivariate dependency: directly vs. indirectly
  - Indirect induction of residual correlation via random intercepts
  - Caveats about fitting multivariate generalized linear models using univariate software (e.g., SAS GLIMMIX, STATA GLM)

3 Parts of Generalized Linear Models

C. Actual Data ← **A. Link Function** = **B. Same Linear Predictive Model**

A. **Link Function**: Transformation of conditional mean to keep *predicted outcomes* within the bounds of the outcome

B. **Same Linear Predictor**: How the model linearly predicts the *link-transformed* conditional mean of the outcome
   - Btw, I call this as the "**model for the means**" more generally

C. **Conditional Distribution**: How the outcome residuals could be distributed given the possible values of the outcome

- **Now we need to consider how the model needs to adapt when residuals are correlated → capture "dependency"**
  - Btw, I call this as the "**model for the variance**" more generally

Estimating (Balanced) Multivariate Models

- Multivariate models can be estimated by **"tricking" univariate software** for general(*ized*) linear models (e.g., SAS MIXED, STATA MIXED) if each **variable is either a predictor OR an outcome**, not both, such as when:
  - You want to examine **mean differences** across the outcomes (e.g., over time or across conditions, as in traditional Repeated Measures ANOVA)
  - You want to test **differences in the effects of predictors** across outcomes (i.e., as in traditional MANOVA)
  - In this case we can build correlations (directly or indirectly) into the model between outcomes from the same person

- Multivariate models will need to be estimated in **"truly" multivariate software** (i.e., as path analysis models or structural equation models) if some **variables are both predictors and outcomes**, such as in mediation
  - e.g., X → M → Y, in which M is both an outcome of X and a predictor of Y
  - This involves regressions instead of correlations between outcomes
  - Otherwise correlations can be built in directly or indirectly… **more on that**:

Taxonomy of Multivariate Dependency

- Dependency = correlated outcome residuals from same sampling unit
- Here is a taxonomy of how **residual correlation** can be included in models for multivariate outcomes when using **univariate software**:

	Normal Conditional Distribution	**Non-Normal Conditional Distribution* using true ML (not pseudo-ML)**
Balanced Design (same possible distinct outcomes per sampling unit)	**R**-Only Pattern –OR– **V** Pattern from **G** random effects & **R**	**G** random effects only
Unbalanced Design (different possible outcomes across sampling units)	**V** Pattern from **G** random effects & **R**	**G** random effects only

* Multiple outcomes per sampling unit will require a multivariate likelihood version of whatever kind of conditional distribution…

Multivariate Dependency, R-Only Style

- For balanced multivariate sampling designs with plausibly normal residuals, multivariate dependency can be specified directly as a **chosen pattern in the R matrix** for each sampling unit:
  - **UN**structured: a separate residual variance for each outcome and for each pair of residual covariances are estimated (fits perfectly)
  - **C**ompound **S**ymmetry **H**eterogeneous: still a separate residual variance for each outcome, but a common (constrained) residual correlation
  - **C**ompound **S**ymmetry: two parameters: a common residual variance and a common residual covariance/correlation across outcomes

- **CS** can be specified an equivalent way using **two matrices** instead of one: **G** & **R**, which creates a combination **V** matrix
  - This strategy is used instead in unbalanced multivariate sampling designs and all multivariate models with non-normal distributions

Introducing **G** & **R** → **V** (Person is Unit)

- e.g., For three outcomes per person, a **Compound Symmetry R** matrix would have this pattern:

$$\mathbf{R}_i = \begin{bmatrix} CS+\sigma_e^2 & CS & CS \\ CS & CS+\sigma_e^2 & CS \\ CS & CS & CS+\sigma_e^2 \end{bmatrix}$$

  - Shown below is how CS can be produced an equivalent way, in which "CS" is the same thing as "random intercept variance" ($\tau_{U_0}^2$) which is distinguished from "residual variance" (σ_e^2)

Random effect source(s) of person dependency are moved to **G** Matrix (dimensions are NOT person-specific)

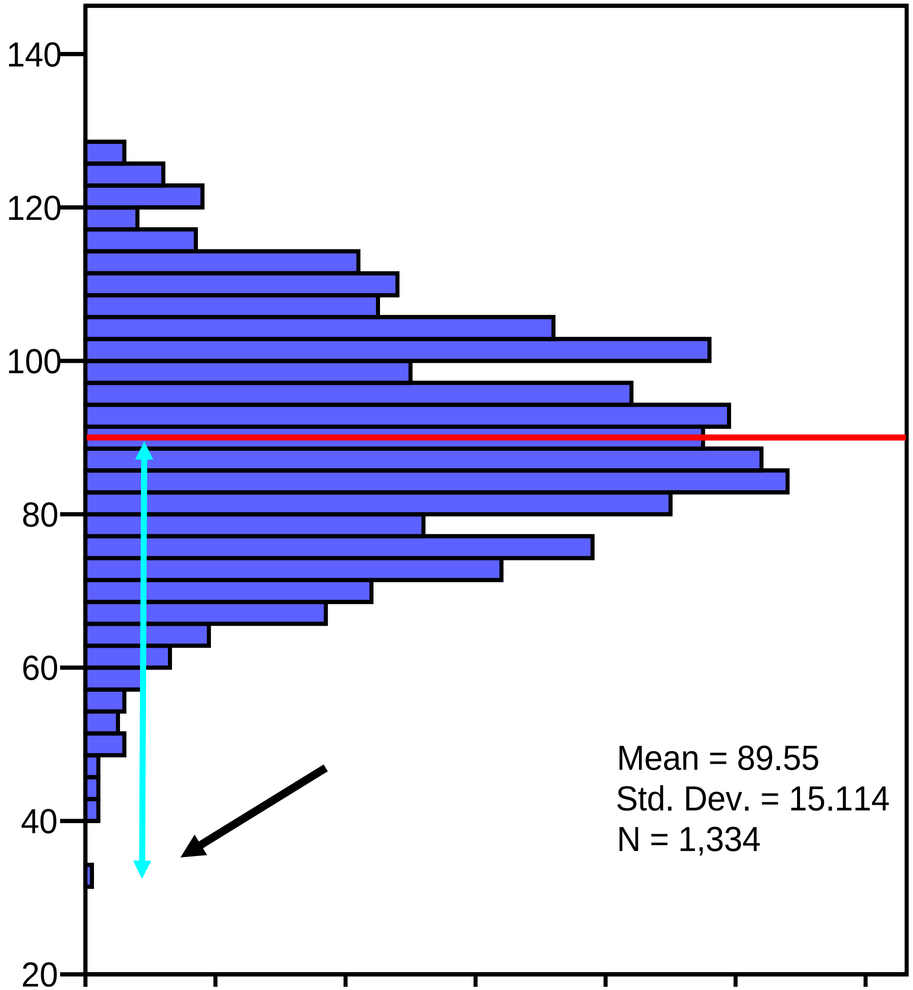
Remaining within-person variance and covariance is in **R** matrix (dimensions are person-specific)

Total Predicted Residual Variance-Covariance Matrix is called **V** Matrix (dimensions are person-specific)

$$\mathbf{G} = \begin{bmatrix} \tau_{U_0}^2 \end{bmatrix} \quad \text{"+"} \quad \mathbf{R}_i = \begin{bmatrix} \sigma_e^2 & 0 & 0 \\ 0 & \sigma_e^2 & 0 \\ 0 & 0 & \sigma_e^2 \end{bmatrix} \quad \text{"="} \quad \mathbf{V}_i = \begin{bmatrix} \tau_{U_0}^2+\sigma_e^2 & \tau_{U_0}^2 & \tau_{U_0}^2 \\ \tau_{U_0}^2 & \tau_{U_0}^2+\sigma_e^2 & \tau_{U_0}^2 \\ \tau_{U_0}^2 & \tau_{U_0}^2 & \tau_{U_0}^2+\sigma_e^2 \end{bmatrix}$$

But what is a "U" anyway ????

Let's Go Way Back: An Empty Univariate GLM



$$y_i = \beta_0 + e_i$$

Filling in values:

$$32 = 90 + -58$$

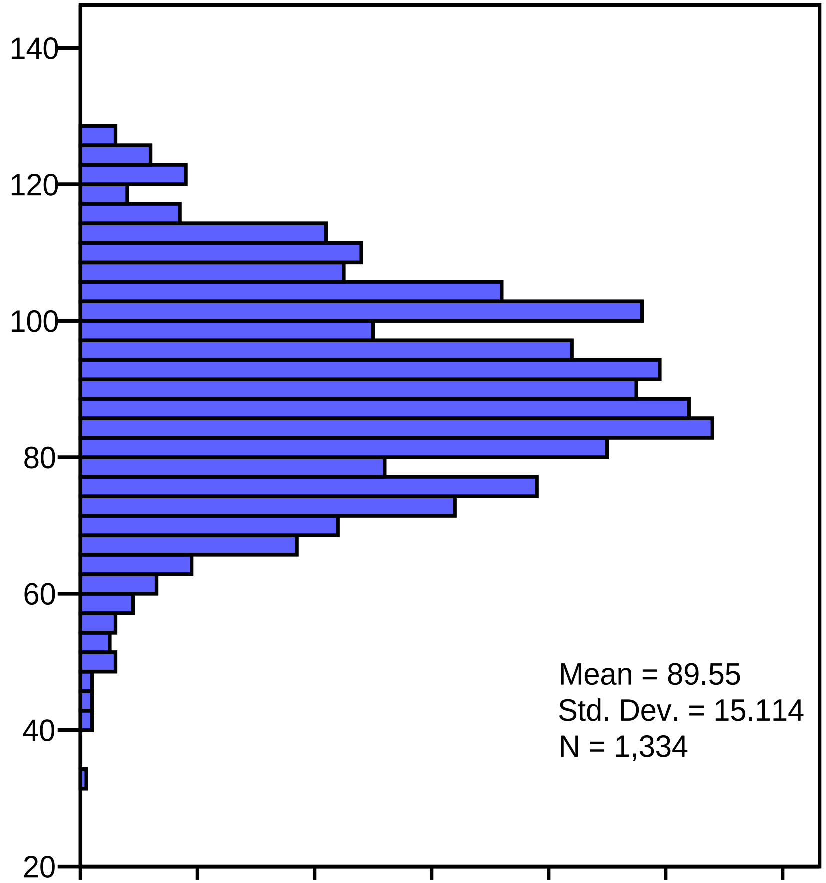
$$\hat{y}_i = 90$$

All the variance in y_i:

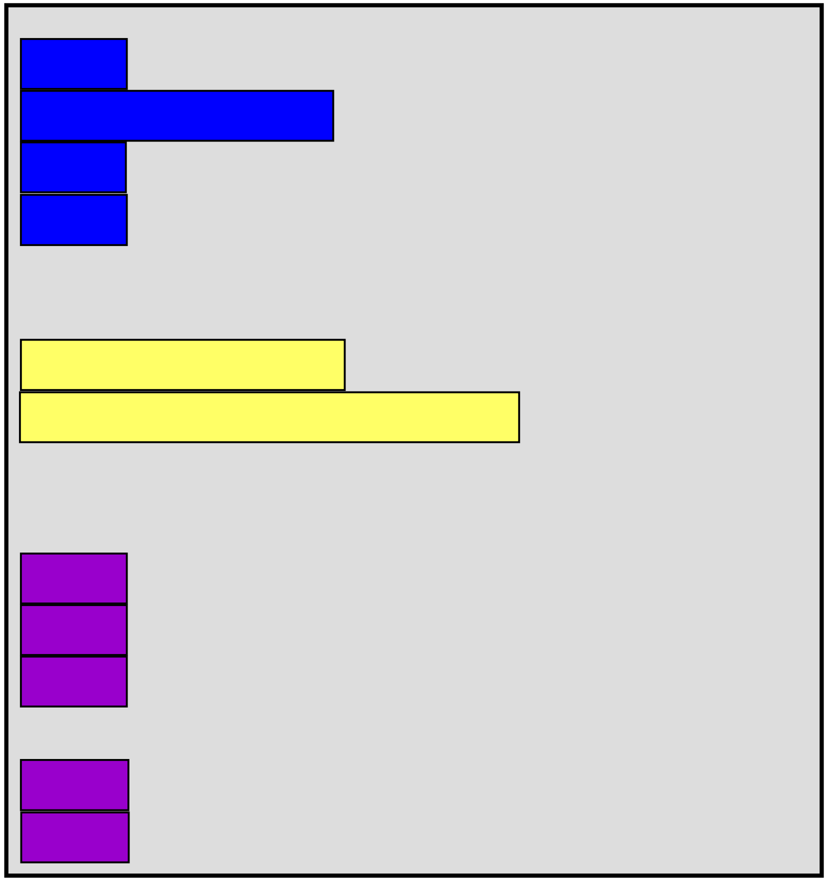
$$\sigma_e^2 = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N-1}$$

Adding 4 Outcomes (t) Per Person… (i.e., to become a Multivariate Model)

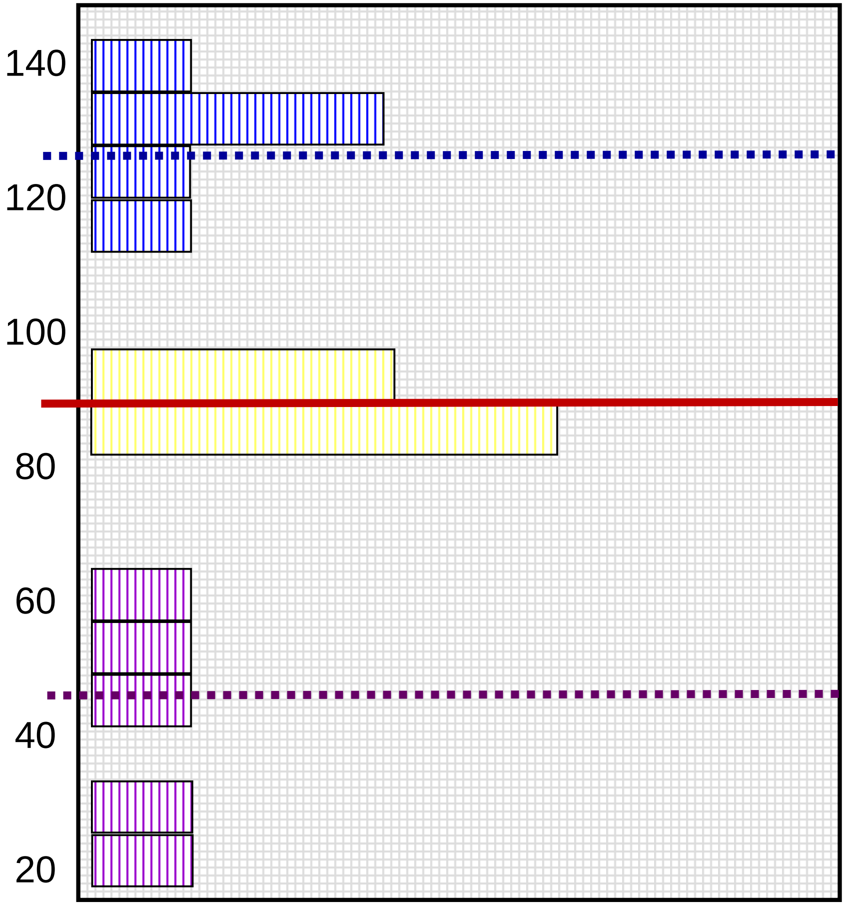
Full Sample Distribution



3 People, 5 Outcomes each



Empty Means Multivariate Model



$$y_{it} = \beta_{00} + e_{it}$$

Start off with Mean of y_{it} as "best guess" for any $\hat{y}_{it}$:

= Grand Mean

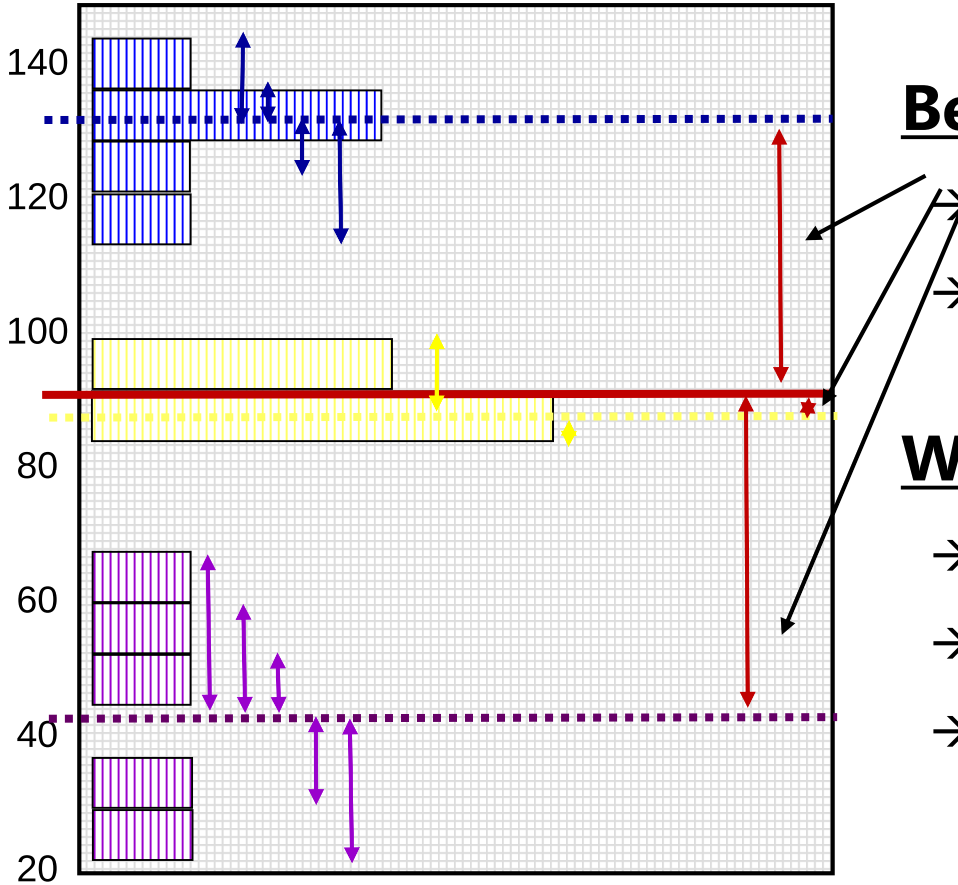
= Fixed Intercept

Can make better guess by taking advantage of repeated observations:

= Person Mean

→ Random Intercept

Empty Means Multivariate Model



Variance of y_{it} → 2 sources:

Between-Person (BP) Variance:

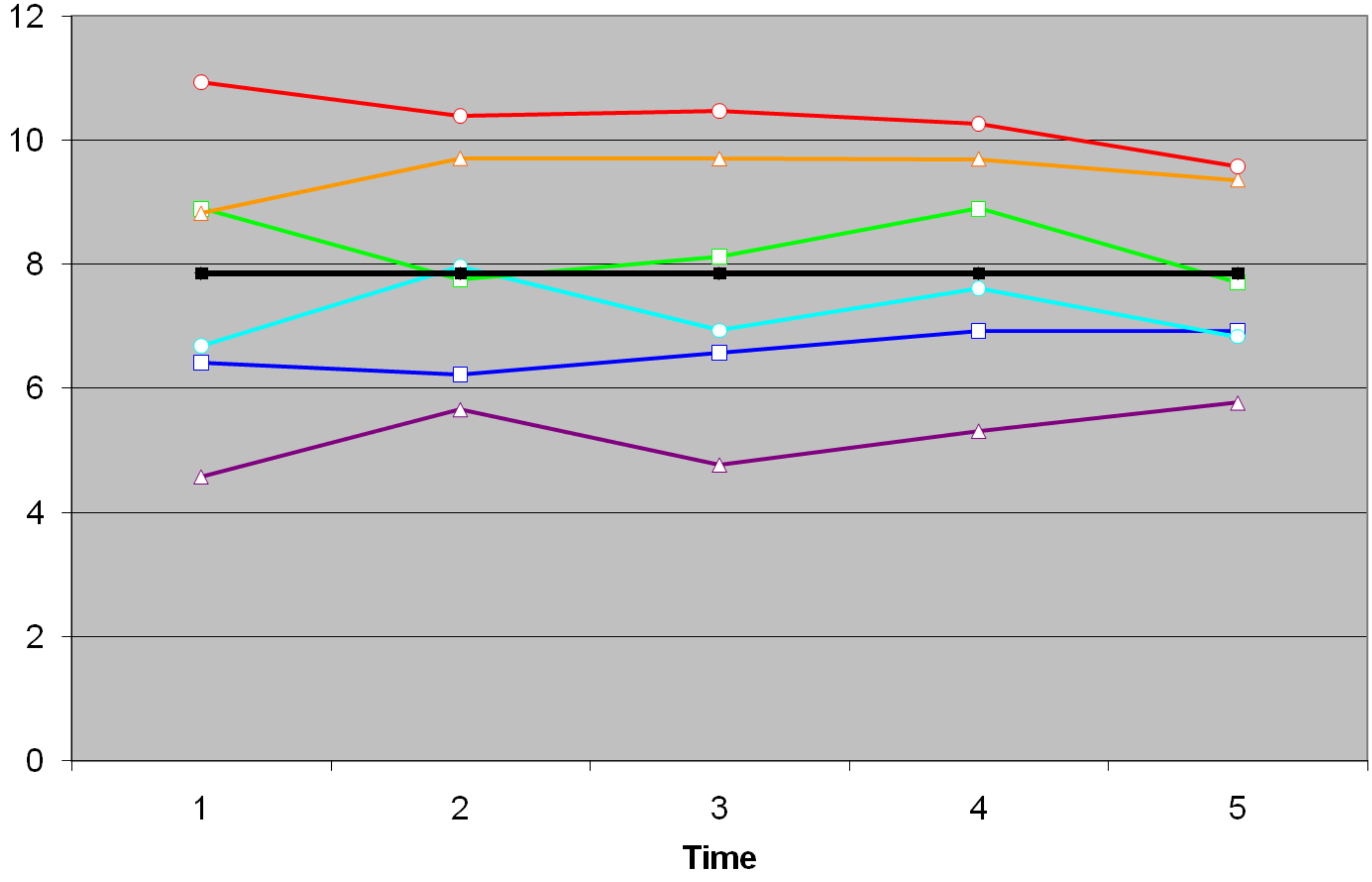
- → Differences from **GRAND** mean
- → **INTER**-Individual Differences

Within-Person (WP) Variance:

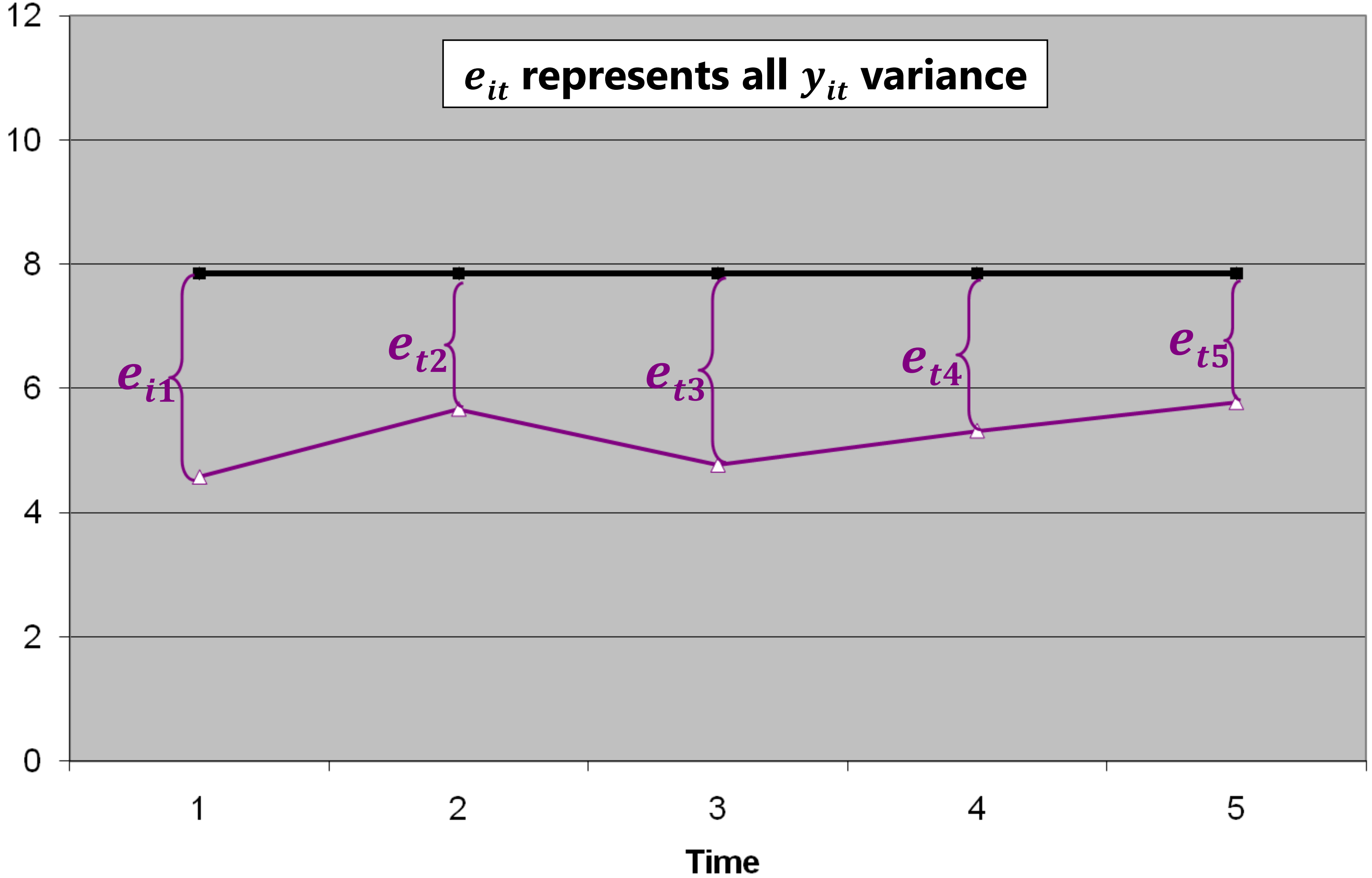
- → Differences from **OWN** mean
- → **INTRA**-Individual Differences
- → This part is only observable through multivariate data.

Now we have 2 piles of variance in y_i to predict.

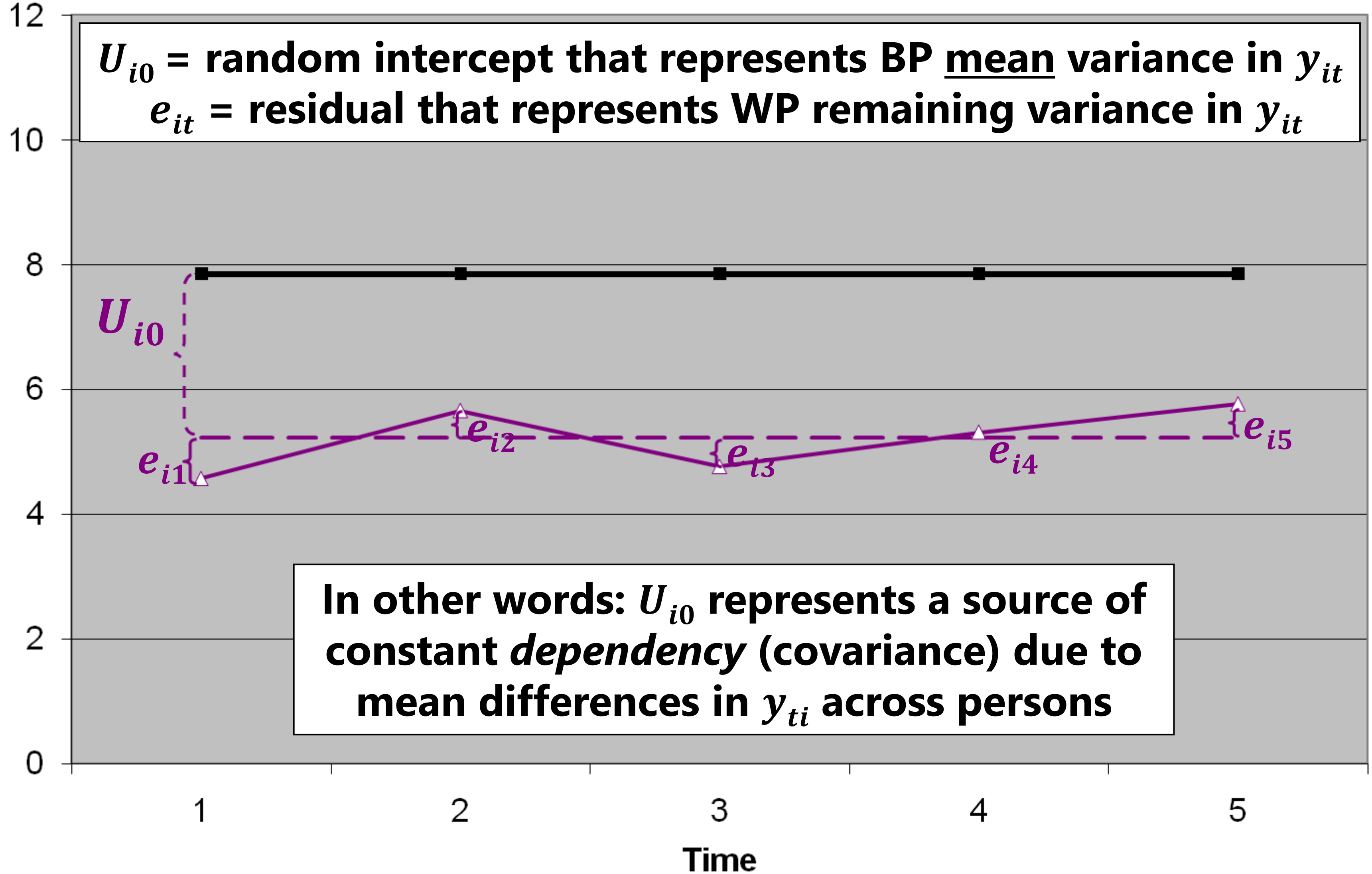
Hypothetical Longitudinal Data



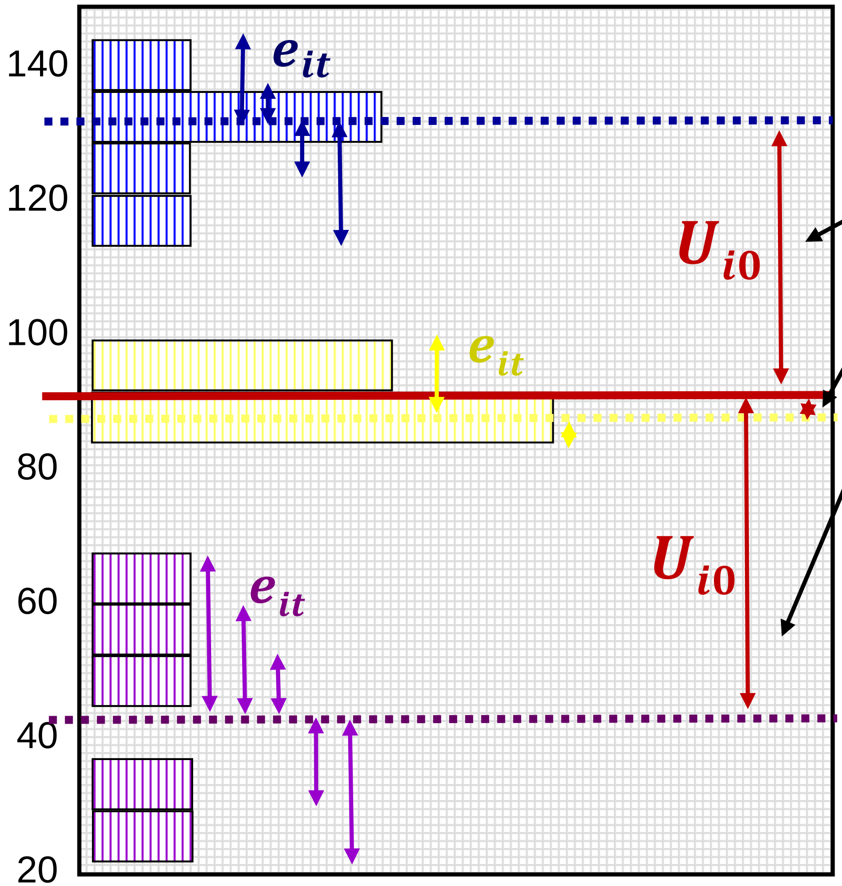
"Error" in a Univariate GLM



Sources of "Error" in a Multivariate GLM



Empty Means Multivariate Model



y_{it} variance → 2 sources:

"Random Intercept" Variance:

- → **of U_{i0} → $\tau^2_{U_0}$**
- → **Between**-Person Variance
- → Differences from **GRAND** mean

"Residual" Variance:

- → **of e_{it} → σ^2_e**
- → **Within**-Person Variance
- → Differences from **OWN** mean

Univariate vs. Multivariate Empty Models

- Empty **Univariate** Model (used for 1 outcome):

$$y_i = \beta_0 + e_i$$

  - β_0 = fixed intercept = grand mean
  - e_{it} = residual deviation from GRAND mean

- Empty **Multivariate** Model (for >1 outcomes):

$$y_{it} = \beta_{00} + U_{i0} + e_{it}$$

Outside of longitudinal data, this model would also include a separate mean per outcome

  - β_{00} = fixed intercept = grand mean
  - U_{i0} = random intercept = individual deviation from GRAND mean
  - e_{it} = outcome-specific residual deviation from OWN mean

Dependency via a Random Intercept

- A scalar example model with $n = 3$ outcomes (A, B, and C):

$$\boldsymbol{y_{it}} = \boldsymbol{\beta_{00}} + \boldsymbol{\beta_{01}}(\boldsymbol{dvA_{it}}) + \boldsymbol{\beta_{02}}(\boldsymbol{dvC_{it}}) + \boldsymbol{U_{i0}} + \boldsymbol{e_{it}}$$

- In matrix notation, this becomes $\boldsymbol{Y_i} = \boldsymbol{X_i}\boldsymbol{\beta} + \boldsymbol{Z_i}\boldsymbol{U_i} + \boldsymbol{E_i}$

$\boldsymbol{Y_i}$	$\boldsymbol{X_i}$	$\boldsymbol{\beta}$	$\boldsymbol{Z_i}$	$\boldsymbol{U_i}$	$\boldsymbol{E_i}$

$$\begin{bmatrix} y_{t1} \\ y_{t2} \\ y_{t3} \end{bmatrix} = \begin{bmatrix} 1 & dvA_{t1} & dvB_{t1} \\ 1 & dvA_{t2} & dvB_{t2} \\ 1 & dvA_{t3} & dvB_{t3} \end{bmatrix} \begin{bmatrix} \beta_{00} \\ \beta_{01} \\ \beta_{02} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [U_{i0}] + \begin{bmatrix} e_{t1} \\ e_{t2} \\ e_{t3} \end{bmatrix}$$

$\boldsymbol{Y_i} = n * 1$ outcome vector
$\boldsymbol{X_i} = n * k$ matrix for predictors that have fixed effects
$\boldsymbol{\beta} = k * 1$ fixed effects vector
$\boldsymbol{Z_i} = n * u$ matrix for predictors that have random effects

n = # outcomes for person i
k = # model fixed effects
u = # model random effects

$\boldsymbol{U_i} = u * 1$ random effects vector
$\boldsymbol{E_i} = n * 1$ residual vector

Predicted V: Total Variance and Covariance across $n = 3$ Outcomes for Person i

$$\mathbf{V}_i = \mathbf{Z}_i * \mathbf{G}_i * \mathbf{Z}_i^T + \mathbf{R}_i$$

$$\mathbf{V}_i = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} \tau^2_{U_0} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} \sigma^2_e & 0 & 0 \\ 0 & \sigma^2_e & 0 \\ 0 & 0 & \sigma^2_e \end{bmatrix}$$

$$\mathbf{V}_i = \begin{bmatrix} \tau^2_{U_0} + \sigma^2_e & \tau^2_{U_0} & \tau^2_{U_0} \\ \tau^2_{U_0} & \tau^2_{U_0} + \sigma^2_e & \tau^2_{U_0} \\ \tau^2_{U_0} & \tau^2_{U_0} & \tau^2_{U_0} + \sigma^2_e \end{bmatrix}$$

$$\mathbf{V}_i = \mathbf{R}_i = \begin{bmatrix} CS + \sigma^2_e & CS & CS \\ CS & CS + \sigma^2_e & CS \\ CS & CS & CS + \sigma^2_e \end{bmatrix}$$

Same result: compound symmetry, either indirectly (G&R→V) or directly (CS for R)

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

$\boldsymbol{Z}_i^T = u\ x\ n$ values of predictors with random effects (just $\boldsymbol{Z}_i$ transposed)

$\boldsymbol{G}_i = u\ x\ u$ estimated **random effects variances and covariances**, so will be the same for all persons (τ^2_{U0} = intercept variance)

$\boldsymbol{R}_i = n\ x\ n$ **outcome-specific residual variances and covariances**, so will be same for all persons (here, just diagonal σ^2_e, although it's possible to add heterogeneous variances and/or covariances)

Distribution Terminology for MVN

- Scalar: $y_{it} = \beta_{00} + \beta_{01}(dvA_{it}) + \beta_{02}(dvC_{it}) + U_{i0} + e_{it}$
  Matrix: $Y_i = X_i\beta + Z_iU_i + E_i$

$\hat{Y}_i = X_i\beta$ where $\hat{Y}_i$ **is the conditional Mean** created by **fixed effects** in the model for means

Model for the Variance creates V_i

$$V_i = Z_i * G_i * Z_i^T + R_i$$

$$V_i = \begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}\tau^2_{U_0}\end{bmatrix}\begin{bmatrix}1 & 1 & 1\end{bmatrix} + \begin{bmatrix}\sigma^2_e & 0 & 0\\ 0 & \sigma^2_e & 0\\ 0 & 0 & \sigma^2_e\end{bmatrix}$$

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

Conditional Distributions for Generalized

- Conditional distribution in multivariate general linear models: $\boldsymbol{Y}|\boldsymbol{U} \sim \mathrm{N}(\boldsymbol{X\beta} + \boldsymbol{ZU}, \boldsymbol{R})$
- But $\boldsymbol{R}$ and choices for its patterns doesn't exist for generalized model variants (when using true maximum likelihood; stay tuned)
  - No separately estimated residual variance (e.g., in Bernoulli, multinomial, Poisson, or binomial) means no direct residual covariances are possible for multivariate models in any software
  - Univariate software (SAS GLIMMIX or STATA GLM) does not fit separate "stretchy" factors for negative binomial, beta-binomial, or gamma (and still no direct residual covariances are possible in any software)
  - So to maintain independent observations in the conditional distribution, all multivariate outcome relationships must be modeled indirectly in the linear predictor using regressions among outcomes OR random effects
    - In univariate software, can use random effects only
    - In software for path analysis or structural equation models (SEM), can use regressions between outcomes OR random effects
  - Estimation becomes harder when including random effects…

A Little Bit about Estimation

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

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

Quasi-Likelihood Estimation

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

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

Marginal Maximum Likelihood Estimation

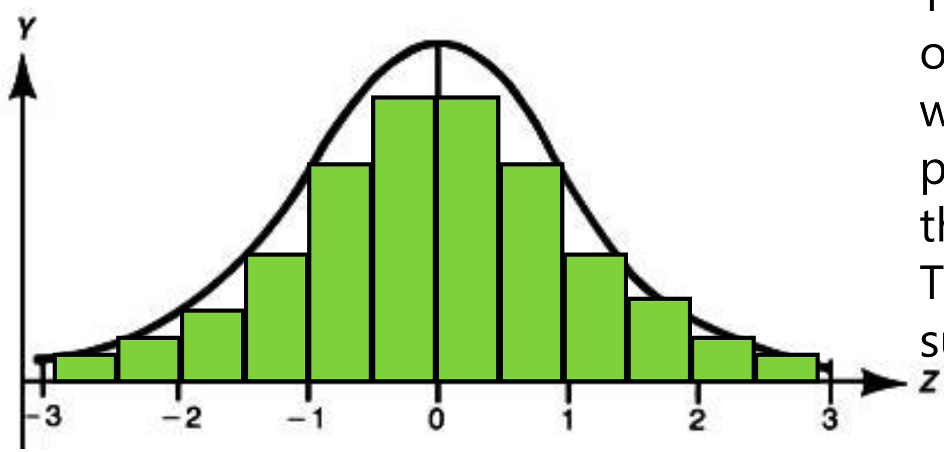
- **ML via Numeric(al) Integration** → gold standard
  - Synonyms: (adaptive) Gaussian quadrature
  - Provides much better variance estimates and valid −2ΔLL tests (ML only; no REML) in "large enough" samples
  - Can take forever or not converge at all in models with many random effects; not often available for models with crossed random effects
    - "Laplace" approximation is used, which is equivalent to 1 integration point (???)
  - Start values can help speed estimation (i.e., from QL methods)
  - Relies on assumptions of local independence → all outcome dependency has been modeled; sampling units are independent

ML via Numeric(al) Integration

- **Step 1**: Select **starting values** for all fixed effects
- **Step 2**: Compute the **likelihood** of each observation given by the *current* parameter values using chosen distribution of residuals
  - Model gives link-predicted outcome given parameter estimates, but the U's themselves are not parameters—their variances and covariances are instead
  - But so long as we can assume the **U**'s are MVN, we can still proceed…
  - Computing the likelihood for each set of possible parameters requires *removing* the contribution of the individual **U** values from the model equation—by ***integrating*** across possible **U** values for each sampling unit (person here)
  - Integration is accomplished by "Gaussian Quadrature" → summing up rectangles that approximate the integral (area under the curve) for each sampling 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, Two Outcomes, Random Intercept Model

1. Start with values for fixed effects: intercept: $\beta_{00} = 0.5$, $\beta_{01} = 1.5$,
2. Compute likelihood for real data based on fixed effects and plausible U_{i0} (-2, 0, or 2) using model: $\text{Logit}(y_{it} = 1) = \beta_{00} + \beta_{01}(x_{it}) + U_{i0}$
   - Here for one person for two outcomes with $y_{it} = 1$ for both outcomes

			IF y=1	IF y=0	Likelihood	U0	U0	Product
	U0 = -2	**Logit**	**Prob**	**1-Prob**	**if both y=1**	**prob**	**width**	**per U0**
x=0	(0.5 -2)	-1.5	0.18	0.82	0.091213	0.05	2	0.00912
x=1	(0.5+1.5-2)	0.0	0.50	0.50				
	U0 = 0	**Logit**	**Prob**	**1-Prob**				
x=0	(0.5-0)	0.5	0.62	0.38	0.54826	0.40	2	0.43861
x=1	(0.5+1.5-0)	2.0	0.88	0.12				
	U0 = +2	**Logit**	**Prob**	**1-Prob**				
x=0	(0.5+2)	2.5	0.92	0.08	0.897053	0.05	2	0.08971
x=1	(0.5+1.5+2)	3.5	0.97	0.03				
Overall Likelihood (Sum of Products over All U0 Values):								**0.53743**
(do this for each person, then multiply this whole thing over all persons)								
(repeat with new values of fixed effects until find highest overall likelihood)								

Univariate software restricts what outcomes can be predicted together

- SAS GLIMMIX has a "byobs" option that directs the LINK= and DIST= options to variables that specify what should be used for each row (no analog in STATA I could find)
  - e.g., could predict separate binary and count outcomes to get a joint test for each predictor
  - Does not include multinomial (so no ordinal or nominal outcomes)

- If-and-How-Much models are harder to fit in univariate software, too
  - SAS GLIMMIX does not have zero-truncated counts, so it cannot fit hurdle models for discrete "how much", but it could be used for continuous "how much" part (e.g., lognormal or gamma)
  - STATA NLMIXED can do almost anything—IF you can figure out how to program it!
  - STATA has hurdle models for a single outcome but they have no way to add the random effects needed in order to model multiple sets of hurdle outcomes simultaneously
  - STATA's GLLAMM can do continuous "how much" using gamma (other families available too)

- "Truly" multivariate software (for path analysis or SEM) affords greater flexibility for multivariate dependency, but fewer choices for an outcome's conditional distribution
  - Mplus: Normal, Bernoulli, multinomial, Poisson, negative binomial (and zero-truncated for each)
  - STATA GSEM: Normal, Bernoulli, multinomial (ordinal or nominal specifically), Poisson, negative binomial , and gamma

Wrapping Up…

- When each sampling unit has >1outcome → **multivariate models**
  - Our model needs to capture **dependency** (correlated residuals)
  - For plausibly normal outcomes, **dependency can be modeled directly**: we can allow same or different residual variances and covariances across outcomes (in a **person-specific R matrix** of type UN, CSH, or CS)
  - For **other outcome types**, dependency must be modeled indirectly by including **random effects** (which means more challenging estimation)
- For convenience, **fixed effects** can be specified in 2 different ways
  - Single general intercept → DV terms reflect DV **differences**
  - Multiple DV-specific intercepts → DV terms are **switches** for own effects
- Univariate software for multivariate generalized linear models is less flexible than "truly" multivariate software—**so onto path models**!!