Hierarchical Models with Stan

Ben Goodrich

April 20, 2022

HW1, Question 1

- Empirical “statistical” practice historically is based on two conventions:

  1. Reject the null hypothesis that $\theta = \theta_0$ if and only if the realization of the p-value is less than 0.05.

  2. If you fail to reject the null hypothesis that $\theta = \theta_0$, proceed as if $\theta = \theta_0$. If you reject the null hypothesis that $\theta = \theta_0$ in favor of the alternative hypothesis that $\theta \neq \theta_0$, either proceed as if

     - $\theta = \hat{\theta}$

     - $\theta \sim \mathcal{N}\left(\hat{\theta}, \widehat{\mathrm{se}}\right)$

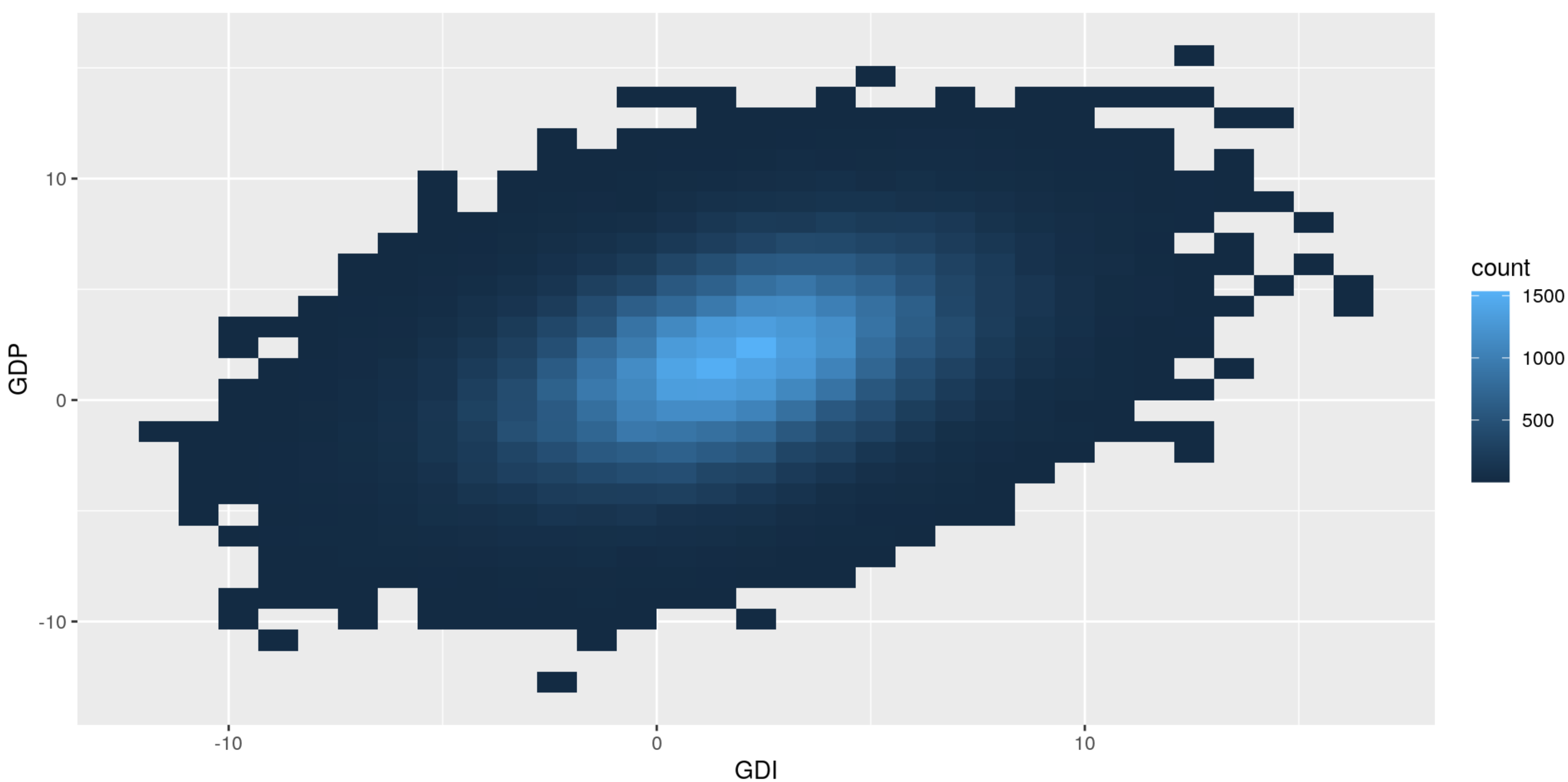
- Many criticisms can (and have, mostly by Bayesians, for decades) be made against these two conventions, but the conventions are not Frequentist

HW1, Question 2

- From a Frequentist or supervised learning perspective, it does not make sense to think about μ for one period of time, like the first quarter of 2022
- Nevertheless, tons of people are thinking about exactly that, albeit not from a Bayesian perspective. See the Survey of Professional Forecasters
- Choosing a normal prior for μ, like $\mathcal{N}\left(1.8, 2.15\right)$ is not difficult, which can then be used to draw from the prior predictive distribution

```
library(dplyr); library(ggplot2)
m <- 1.8; s <- 2.15; sigma <- 7 / 3; rho <- -1 / 10
tibble(mu = rnorm(10^5, mean = m, sd = s),
       GDI = rnorm(10^5, mean = mu, sd = sigma),
       GDP = rnorm(10^5, mean = mu + rho * (GDI - mu),
                   sd = sigma * sqrt(1 - rho^2))) %>%
  ggplot() + geom_bin_2d(aes(x = GDI, y = GDP))
```

Plot from Previous Slide



HW1, Question 2: Posterior Distribution

- You can numerically evaluate the denominator of Bayes Rule

$$f\left(\cancel{\mu} \bigcap \mathrm{GDP} \bigcap \mathrm{GDI} \mid m, s, \sigma, \rho\right) =$$

$$\int_{-\infty}^{\infty} \frac{e^{-\frac{1}{2}\left(\frac{\mu-m}{s}\right)^2}}{s\sqrt{2\pi}} \frac{e^{-\frac{1}{2}\left(\frac{\mathrm{GDP}-\mu}{\sigma}\right)^2}}{\sigma\sqrt{2\pi}} \frac{e^{-\frac{1}{2}\left(\frac{\mathrm{GDI}-(\mu+\rho(\mathrm{GDP}-\mu))}{\sigma\sqrt{1-\rho^2}}\right)^2}}{\sigma\sqrt{1-\rho^2}\sqrt{2\pi}} d\mu$$

- However, in this case, the integrals can be evaluated “analytically”
- Conditional on GDP alone, the posterior distribution of μ is $\mathcal{N}\left(m^*, s^*\right)$ where $m^* = m\frac{\sigma^2}{s^2+\sigma^2} + \mathrm{GDP}\frac{s^2}{s^2+\sigma^2}$ and $s^* = s\sigma\sqrt{\frac{1}{s^2+\sigma^2}}$. Conditional on both GDP and GDI, $m^* = m\frac{(1+\rho)\sigma^2}{2s^2+(1+\rho)\sigma^2} + \mathrm{GDO}\frac{2s^2}{2s^2+(1+\rho)\sigma^2}$ and $s^* = s\sigma\sqrt{\frac{1+\rho}{2s^2+(1+\rho)\sigma^2}}$. As $s \uparrow \infty, m^* \rightarrow \mathrm{GDO} = \frac{\mathrm{GDP}+\mathrm{GDI}}{2}$.

HW1, Question 3

- You already had Stan code for this problem because it is the same as in the vaccination / Trump model. You just need to choose different GLD priors that are appropriate for the individual stock you choose, here GameStop.

```
R_i <- tq_get("GME", from = "2020-06-01", to = "2022-04-01") %>%
  filter(weekdays(date) == "Wednesday") %>%
  transmute(R_i = (adjusted - lag(adjusted)) / lag(adjusted)) %>%
  na.omit %>%
  pull
```

- α should have a prior median of about zero
- On average across all companies, $\beta = 1$ under the CAPM, but that might not hold for individual companies. It is hard to justify $\beta < 0$ for a long investment, but the right tail is long for “meme stocks”.
- The marginal standard deviation of R_i can be used as an upper bound on the standard deviation of the errors, which would be achieved if $\beta = 0$

What Are Hierarchical Models

- In Bayesian terms, a hierarchical model is nothing more than a model where the prior distribution of some parameter depends on another parameter
- In other words, it is just another application of the Multiplication Rule

$$f(\boldsymbol{\theta}) = \int f(\boldsymbol{\theta} \mid \boldsymbol{\phi}) f(\boldsymbol{\phi}) \, d\phi_1 \dots d\phi_K$$

- But most of the discussion of “hierarchical models” refers to the very narrow circumstances in which they can be estimated via Frequentist methods
- From a Frequentist perspective, a hierarchical model is appropriate for cluster random sampling designs, inappropriate for stratified random sample designs, and hard to justify for other sampling designs

Prior Predictive Distribution of Hierarchical Model

- Here is how McElreath does many hierarchical binomial models
- Suppose a categorical predictor x_k has K levels

$$\sigma \sim \mathcal{E}\left(r\right)$$
$$\forall k: \beta_k \sim \mathcal{N}\left(m_k, \sigma\right)$$
$$\forall k: \mu_k = \frac{1}{1 + e^{-\beta_k}}$$
$$\forall k: y_k \sim \text{Binomial}\left(n_k, \mu_k\right)$$

- Aggregating Bernoulli random variables with a common success probability to binomial random variables is much more computationally efficient

Cluster Sampling vs. Stratified Sampling

- For cluster random sampling, you
  - Sample J large units (such as schools) from their population
  - Sample N_j small units (such as students) from the j-th large unit
- If you replicate such a study, you get different realizations of the large units
- For stratified random sampling, you
  - Divide the population of large units into J mutually exclusive and exhaustive groups (like states)
  - Sample N_j small units (such as voters) from the j-th large unit
- If you replicate such a study, you would use the same large units and only get different realizations of the small units

Why Bayesians Should Use Hierarchical Models

- Suppose you estimated a Bayesian model on people in New York
- Next, you are going to collect data on people who live in Connecticut
- Intuitively, the New York posterior should influence the Connecticut prior
- But it is unlikely that the data-generating processes in Connecticut is exactly the same as in New York
- Hierarchical models apply when you have data from New York, Connecticut, and other states at the same time
- Posterior distribution in any one state is not independent of other states
- Posterior distribution in any one state are not the same as in other states
- McElreath argues hierarchical models should be the default and “flat” models should be the rare exception only when justified by the data
- With more data, there is always more heterogeneity in the data-generating processes that a generative model should be allowing for

Models with Group-Specific Intercepts

- Let α be the common intercept and $\boldsymbol{\beta}$ be the common coefficients while a_j is the deviation from the common intercept in the j-th group. Write a model as:

$$y_{ij} = \overbrace{\underbrace{\alpha + \sum_{k=1}^{K} \beta_k x_{ik}}_{\text{Frequentist } \boldsymbol{\mu}|\mathbf{x}} + a_j}^{\text{Bayesian } \boldsymbol{\mu}|\mathbf{x},j} + \boldsymbol{\epsilon} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik} + \underbrace{a_j + \overbrace{\boldsymbol{\epsilon}}^{\text{Bayesian error}}}_{\text{Frequentist error}}$$

- The same holds in GLMs where $\eta_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik} + a_j$ or $\eta_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik}$ depending on whether you are Bayesian or Frequentist
- Many people write $\alpha_j \equiv \alpha + a_j$

Models with Group-Specific Slopes and Intercepts

- Let α be the common intercept and $\boldsymbol{\beta}$ be the common coefficients while a_j and $\mathbf{b}_j$ are the deviations from the common intercept and slope respectively:

$$y_{ij} = \overbrace{\underbrace{\alpha + \sum_{k=1}^{K} \beta_k x_{ik}}_{\text{Frequentist } \boldsymbol{\mu}|\mathbf{x}} + a_j + \sum_{k=1}^{K} b_{jk} x_{ik}}^{\text{Bayesian } \boldsymbol{\mu}|\mathbf{x},j} + \boldsymbol{\epsilon} =$$

$$\alpha + \sum_{k=1}^{K} \beta_k x_{ik} + \underbrace{a_j + \sum_{k=1}^{K} b_{jk} x_{ik} + \overbrace{\boldsymbol{\epsilon}}^{\text{Bayesian error}}}_{\text{Frequentist error}}$$

- And similarly for GLMs, but you need a joint prior on a_j, b_j

Data for a Binomial GLM

```
funding <-
  tibble(
    discipline   = rep(c("Chemical sciences", "Physical sciences", "Physics", "Humanities",
                         "Technical sciences",  "Interdisciplinary", "Earth/life sciences",
                         "Social sciences", "Medical sciences"),
                     each = 2) %>% as.factor,
    female       = rep(0:1, times = 9),
    applications = c(83, 39, 135, 39, 67, 9, 230, 166, 189,
                     62, 105, 78, 156, 126, 425, 409, 245, 260),
    awards       = c(22, 10, 26, 9, 18, 2, 33, 32, 30,
                     13, 12, 17, 38, 18, 65, 47, 46, 29)
  )
stan_data <- with(funding, list(N = nrow(funding), J = nlevels(discipline),
                                discipline = as.integer(discipline), female = female,
                                applications = applications, awards = awards,
                                prior_only = 0, m = 0, s = 1))
```

- How would we write the Stan program?

Data Block of a Stan Program

```
data {
  int<lower = 0> N; // number of observations
  int<lower = 1> J; // number of disciplines
  int<lower = 1, upper = J> discipline[N];
  vector<lower = 0, upper = 1>[N] female;
  int<lower = 0> applications[N];
  int<lower = 0, upper = max(applications)> awards[N];

  int<lower = 0, upper = 1> prior_only;
  real m;            // prior me{di}an
  real<lower = 0> s; // prior standard deviation
}
```

Special Matrices

- A square matrix has the same number of rows as columns
- A square matrix $\mathbf{X}$ is symmetric iff $\mathbf{X} = \mathbf{X}^\top$
- Triangular matrices are square matrices such that
  - Lower triangular matrix has $X_{kp} = 0 \, \forall k < p$
  - Upper triangular matrix has $X_{kp} = 0 \, \forall k > p$
- Diagonal matrix is a square matrix that is simultaneously lower and upper triangular and thus has $X_{kp} = 0 \, \forall k \neq p$
- The identity matrix, $\mathbf{I}$, is the diagonal matrix with only ones on its diagonal — i.e. $I_{kp} = \begin{cases} 1 & \text{if } k = p \\ 0 & \text{if } k \neq p \end{cases}$ — and is the matrix analogue of the scalar 1
- If $\mathbf{X}$ is square, then $\mathbf{XI} = \mathbf{X} = \mathbf{IX}$
- A square orthogonal matrix $\mathbf{Q}$ is such that $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I} = \mathbf{Q}\mathbf{Q}^\top$, but sometimes we refer to a rectangular matrix as having orthogonal columns if $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$

Matrix Inversion

- If $\mathbf{X}$ is a square matrix, then the inverse of $\mathbf{X}$ — if it exists — is denoted $\mathbf{X}^{-1}$ and is the unique matrix of the same size such that $\mathbf{X}\mathbf{X}^{-1} = \mathbf{I} = \mathbf{X}^{-1}\mathbf{X}$
- Don't worry about how software finds the elements of $\mathbf{X}^{-1}$, just use `solve` in R or various functions in Stan
  - But if $\mathbf{X}$ is diagonal, then $\left[\mathbf{X}^{-1}\right]_{kp} = \begin{cases} \frac{1}{X_{kp}} & \text{if } k = p \\ 0 & \text{if } k \neq p \end{cases}$
  - If $\mathbf{X}$ is only triangular, $\mathbf{X}^{-1}$ is also triangular and easy to find
- There is no vector or matrix "division" but multiplying $\mathbf{X}$ by $\mathbf{X}^{-1}$ is the matrix analogue of scalar multiplying a by $\frac{1}{a}$. Also, $(\mathbf{X}a)^{-1} = \frac{1}{a}\mathbf{X}^{-1}$.
- An inverse of a product of square matrices equals the product of the inverses in reverse order: $(\mathbf{X}\mathbf{Y})^{-1} = \mathbf{Y}^{-1}\mathbf{X}^{-1}$. Also, the inverse of a transpose of a square matrix is the transpose of the inverse: $\left(\mathbf{X}^{\top}\right)^{-1} = \left(\mathbf{X}^{-1}\right)^{\top}$

Covariance and Correlation Matrices

- Recall that if $g(X_i, X_j) = (X_i - \mu_i)(X_j - \mu_j)$, then

$$\mathbb{E}g(X_i, X_j) = \int_{\Omega_{X_j}} \int_{\Omega_{X_i}} (x_i - \mu_i)(x_j - \mu_j) f(x_i, x_j)\, dx_i dx_j = \sigma_{ij}$$

is the covariance between X_i and X_j, while $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \in [-1, 1]$ is their correlation, which is a measure of LINEAR dependence

- Let $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$ be $K \times K$, such that $\Sigma_{ij} = \sigma_{ij}\, \forall i, j$ and $\Lambda_{ij} = \rho_{ij}\, \forall i \neq j$
    - Since $\sigma_{ij} = \sigma_{ji}\, \forall i, j$, $\mathbf{\Sigma} = \mathbf{\Sigma}^\top$ is symmetric
    - Since $\sigma_{ij} = \sigma_i^2$ iff $i = j$, $\Sigma_{ii} = \sigma_i^2 > 0$
    - Hence, $\mathbf{\Sigma} = \mathbb{E}(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top$ is the variance-covariance matrix of $\mathbf{x}$
    - $\mathbf{\Sigma} = \mathbf{\Delta}\mathbf{\Lambda}\mathbf{\Delta}$ where $\mathbf{\Delta}$ is a diagonal matrix of standard deviations

Multivariate CDFs, PDFs, and Expectations

- If $\mathbf{x}$ is a K-vector of continuous random variables

$$F\left(\mathbf{x}\right)=\Pr\left(X_1\leq x_1\bigcap X_2\leq x_2\bigcap\cdots\bigcap X_K\leq x_K\right)$$

$$f\left(\mathbf{x}\right)=\frac{\partial^K F\left(\mathbf{x}\right)}{\partial x_1\partial x_2\cdots\partial x_K}=f_1\left(x_1\right)\prod_{k=2}^{K}f_k\left(x_k|\,x_1,\ldots,x_{k-1}\right)$$

$$F\left(\mathbf{x}\right)=\int_{-\infty}^{x_k}\cdots\int_{-\infty}^{x_2}\int_{-\infty}^{x_1}f\left(\mathbf{x}\right)dx_1dx_2\cdots dx_K$$

$$\mathbb{E}g\left(\mathbf{x}\right)=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}g\left(\mathbf{x}\right)f\left(\mathbf{x}\right)dx_1dx_2\cdots dx_K$$

$$\boldsymbol{\mu}^\top=\mathbb{E}\mathbf{x}^\top=\begin{bmatrix}\mathbb{E}X_1 & \mathbb{E}X_2 & \cdots & \mathbb{E}X_K\end{bmatrix}$$

$$\boldsymbol{\Sigma}^\top=\boldsymbol{\Sigma}=\mathbb{E}\left[\left(\mathbf{x}-\boldsymbol{\mu}\right)\left(\mathbf{x}-\boldsymbol{\mu}\right)^\top\right]=\begin{bmatrix}\sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1K}\\ \sigma_{12} & \sigma_2^2 & \cdots & \vdots\\ \vdots & \cdots & \ddots & \sigma_{(K-1)K}\\ \sigma_{1K} & \cdots & \sigma_{(K-1)K} & \sigma_K^2\end{bmatrix}$$

Determinants

- A determinant is “like” a multivariate version of the absolute value operation and is denoted with the same symbol, $|\mathbf{X}|$
- Iff $|\mathbf{X}| \neq 0$, then $\mathbf{X}^{-1}$ exists and $\left|\mathbf{X}^{-1}\right| = \frac{1}{|\mathbf{X}|}$
- All you need to know about how determinants are calculated:
  - Any square matrix $\mathbf{X}$ can be factored as $\mathbf{X} = \dot{\mathbf{L}}\mathbf{U}$ where $\dot{\mathbf{L}}$ is unit lower triangular and $\mathbf{U}$ is upper triangular. For covariance matrices, there are further computational shortcuts.
  - Determinant of a product of square matrices is equal to the product of their determinants
  - Determinant of a triangular matrix is the product of its diagonal elements
  - Thus, $|\mathbf{X}| = \left|\dot{\mathbf{L}}\right| \times |\mathbf{U}| = \prod_{k=1}^{K} U_{kk}$

Multivariate Transformations

- Most multivariate distributions are generated by transforming independent random variables from some distribution

- If $\mathbf{z}$ is a K-vector with PDF $f\left(\mathbf{z}\right) = \frac{\partial^K F(\mathbf{z})}{\partial z_1 \partial z_2 \cdots \partial z_K}$ and $\mathbf{x}\left(\mathbf{z}\right)$ is an bijective $\mathbb{R}^K \mapsto \mathbb{R}^K$ function of $\mathbf{z}$, what is the PDF of $\mathbf{x}$?

- $f\left(\mathbf{x} | \cdot\right) = \frac{\partial^K F(\mathbf{z})}{\partial z_1 \partial z_2 \cdots \partial z_K} \times \mathrm{ChainRule}\left(\mathbf{x} \mapsto \mathbf{z}\right) = f\left(\mathbf{z}\left(\mathbf{x}\right) | \cdot\right) \times \left|\mathbf{J}_{\mathbf{x} \mapsto \mathbf{z}}\right|$

  where the Jacobian matrix is $\mathbf{J}_{\mathbf{x} \mapsto \mathbf{z}} = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_K} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_2}{\partial x_K} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial z_K}{\partial x_1} & \frac{\partial z_K}{\partial x_2} & \cdots & \frac{\partial z_K}{\partial x_K} \end{bmatrix}$

Bivariate Normal Distribution with Linear Algebra

- Let $\mathbf{L} = \begin{bmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sigma_2\sqrt{1-\rho^2} \end{bmatrix}$ and let Z_1 and Z_2 be iid standard normal
- If $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ and $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{x}(\mathbf{z}) = \boldsymbol{\mu} + \mathbf{L}\mathbf{z}$, what is the distribution of $\mathbf{x}$?
- $$\begin{bmatrix} x_1(\mathbf{z}) \\ x_2(\mathbf{z}) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \sigma_1 z_1 + 0 z_2 \\ \rho\sigma_2 z_1 + \sigma_2\sqrt{1-\rho^2}\, z_2 \end{bmatrix} \implies \begin{bmatrix} z_1(\mathbf{x}) \\ z_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{x_1-\mu_1}{\sigma_1} \\ \frac{x_2-\mu_2-\rho\sigma_2\left(\frac{x_1-\mu_1}{\sigma_1}\right)}{\sigma_2\sqrt{1-\rho^2}} \end{bmatrix}$$
- $$\mathbf{J} = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 \\ -\frac{\rho\frac{\sigma_2}{\sigma_1}}{\sigma_2\sqrt{1-\rho^2}} & \frac{1}{\sigma_2\sqrt{1-\rho^2}} \end{bmatrix}$$ so $|\mathbf{J}| = \frac{1}{\sigma_1\sigma_2\sqrt{1-\rho^2}}$
- $$f(\mathbf{x}|\,\mu_1,\mu_2,\sigma_1,\sigma_2,\rho) = \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2} \times \frac{1}{\sigma_2\sqrt{1-\rho^2}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_2-\left(\mu_2+\rho\frac{\sigma_2}{\sigma_1}(x_1-\mu_1)\right)}{\sigma_2\sqrt{1-\rho^2}}\right)^2}$$, which is the PDF of the bivariate normal distribution we learned before, written as a product of a marginal normal PDF for x_1 and a conditional normal PDF for $x_2|\,x_1$

Multivariate Normal Distribution

- If Z_k is iid standard normal for all k and $\mathbf{x}\left(\mathbf{z}\right)=\boldsymbol{\mu}+\mathbf{L}\mathbf{z}$ with $L_{kk}>0\,\forall k$ and $L_{ij}=0\,\forall j>i$, what is the distribution of $\mathbf{x}$?

- Step 1: $\mathbf{z}\left(\mathbf{x}\right)=\mathbf{L}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)$ so $z_i\left(\mathbf{x}\right)=\sum_{k=1}^{i}L_{ij}^{-1}\left(x_j-\mu_j\right)$
- Step 2: $\frac{\partial z_i}{\partial x_j}=L_{ij}^{-1}\,\forall i,j$ so $\mathbf{J}_{\mathbf{x}\mapsto\mathbf{z}}=\mathbf{L}^{-1}$ and $\left|\mathbf{J}_{\mathbf{x}\mapsto\mathbf{z}}\right|=\prod_{k=1}^{K}\frac{1}{L_{kk}}=\frac{1}{|\mathbf{L}|}$
- Step 3: $f\left(\mathbf{x}\right|\boldsymbol{\mu},\mathbf{L})=f\left(\mathbf{z}\left(\mathbf{x}\right)\right)\times\left|\mathbf{L}^{-1}\right|=\frac{f(\mathbf{z}(\mathbf{x}))}{|\mathbf{L}|}$
- Step 4: $f\left(\mathbf{z}\right)=\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z_k^2}=\frac{1}{\left(2\pi\right)^{\frac{K}{2}}}e^{-\frac{1}{2}\sum_{k=1}^{K}z_k^2}=\frac{1}{\left(2\pi\right)^{\frac{K}{2}}}e^{-\frac{1}{2}\mathbf{z}^\top\mathbf{z}}$
- Step 5: Substituting for $\mathbf{z}\left(\mathbf{x}\right)$, $f\left(\mathbf{x}\right|\boldsymbol{\mu},\mathbf{L})=\frac{e^{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^\top\left(\mathbf{L}^{-1}\right)^\top\mathbf{L}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)}}{\left(2\pi\right)^{\frac{K}{2}}|\mathbf{L}|}$ and substituting $\boldsymbol{\Sigma}=\mathbf{L}\mathbf{L}^\top$, $f\left(\mathbf{x}\right|\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{e^{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^\top\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)}}{\left(2\pi\right)^{\frac{K}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}$

Cholesky Factors and Positive Definiteness

- Let $\mathbf{L}$ be lower triangular w/ positive diagonal entries such that $\mathbf{L}\mathbf{L}^\top = \boldsymbol{\Sigma}$, which is a Cholesky factor of $\boldsymbol{\Sigma}$ and can uniquely be defined via recursion:

$$L_{ij} = \begin{cases} \sqrt[+]{\Sigma_{jj} - \sum_{k=1}^{j-1} L_{kj}^2} & \text{if } i = j \\ \frac{1}{L_{jj}} \left(\Sigma_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk}\right) & \text{if } i > j \\ 0 & \text{if } i < j \end{cases}$$

- Positive definiteness of $\boldsymbol{\Sigma}$ implies L_{jj} is real and positive for all j and implies the existence of $\boldsymbol{\Sigma}^{-1} = \mathbf{L}^{-1}\left(\mathbf{L}^{-1}\right)^\top$, which is called a "precision matrix". But not all symmetric matrices are positive definite, so $\Theta \subset \mathbb{R}^{K + \binom{K}{2}}$ in this case
- The `cholesky_decompose` function in Stan outputs $\mathbf{L}$, while the `chol` function in R outputs $\mathbf{L}^\top$ instead

The LKJ Distribution for Correlation Matrices

- Let $\boldsymbol{\Delta}$ be a $K \times K$ diagonal matrix such that Δ_{kk} is the k-th standard deviation, σ_k, and let $\boldsymbol{\Lambda}$ be a correlation matrix
- Formulating a prior for $\boldsymbol{\Sigma} = \boldsymbol{\Delta\Lambda\Delta}$ is harder than putting a prior on $\boldsymbol{\Delta}$ & $\boldsymbol{\Lambda}$
- LKJ PDF is $f\left(\boldsymbol{\Lambda}\middle|\,\eta\right) = \frac{1}{c(K,\eta)}\left|\boldsymbol{\Lambda}\right|^{\eta-1} = \left|\mathbf{L}\right|^{2(\eta-1)}$ where $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^\top$ with $\mathbf{L}$ a Cholesky factor and $c\left(K,\eta\right)$ is the normalizing constant that forces the PDF to integrate to 1 over the space of correlation matrices
  - Iff $\eta = 1$, $f\left(\boldsymbol{\Lambda}\middle|\,\eta\right) = \frac{1}{c(K,\eta)}$ is constant
  - If $\eta > 1$, the mode of $f\left(\boldsymbol{\Lambda}\middle|\,\eta\right)$ is at $\mathbf{I}$ and as $\eta \uparrow \infty$, $\boldsymbol{\Lambda} \rightarrow \mathbf{I}$
  - If $0 < \eta < 1$, trough of $f\left(\boldsymbol{\Lambda}\middle|\,\eta\right)$ is at $\mathbf{I}$, which is an odd thing to believe
- Can also derive the distribution of the Cholesky factor $\mathbf{L}$ such that $\mathbf{L}\mathbf{L}^\top$ is a correlation matrix with an LKJ(η) distribution

Frequentist Estimation of Multilevel Models

- Frequentists assume that a_j and b_j deviate from the common parameters according to a (multivariate) normal distribution, whose (co)variances are common parameters to be estimated
- To Frequentists, a_j and b_j are not parameters because parameters must remained fixed in repeated sampling of observations from some population
- Since a_j and b_j are not parameters, they can't be "estimated" only "predicted"
- Since a_j and b_j aren't estimated, they must be integrated out of the likelihood function, leaving an integrated likelihood function of the common parameters
- After obtaining maximum likelihood estimates of the common parameters, each a_j and b_j can be predicted from the residuals via a regression
- Estimated standard errors produced by frequentist software are too small
- There are no standard errors etc. for the a_j and b_j
- Maximum likelihood estimation often results in a corner solution

Frequentist Example

```
poll <- readRDS("GooglePoll.rds") # WantToWin is coded as 1 for Romney and 0 for Obama
poll$Income[poll$Income == "150,000+"] <- "100,000-149,999"
library(dplyr)
collapsed <- filter(poll, !is.na(WantToWin)) %>%
             group_by(Region, Gender, Urban_Density, Age, Income) %>%
             summarize(Romney = sum(grepl("Romney", WantToWin)), Obama = n() - Romney) %>%
             na.omit
```

```
mle <- lme4::glmer(cbind(Romney, Obama) ~ Gender + Urban_Density + Age + Income +
                     (Gender + Urban_Density + Age + Income | Region),
                   data = collapsed, family = binomial(link = "logit"))
```

```
## boundary (singular) fit: see ?isSingular
```

- For models that are more complicated than `(1 + x | g)`, the MLE of $\boldsymbol{\Sigma}$ usually implies that $\widehat{\boldsymbol{\Sigma}}^{-1}$ does not exist. How can we do it with Stan?

Stuff for the Data Block

```
library(lme4)
X <- model.matrix(mle)[ , -1]
Z <- getME(mle, name = "Z")
class(Z)
```

```
## [1] "dgCMatrix"
## attr(,"package")
## [1] "Matrix"
```

```
parts <- extract_sparse_parts(Z)
str(parts)
```

```
## List of 3
##  $ w: num [1:2005] 1 1 1 1 1 1 1 1 1 1 ...
##  $ v: int [1:2005] 1 10 1 11 1 5 10 1 5 11 ...
##  $ u: int [1:514] 1 3 5 8 11 14 17 20 23 25 ...
```

Data for Hierarchical Model of Bowling

```r
ROOT <- "https://www.cs.rpi.edu/academics/courses/fall14/csci1200/"
US_Open2010 <- readLines(paste0(ROOT, "hw/02_bowling_classes/2010_US_Open.txt"))
x1_x2 <- lapply(US_Open2010, FUN = function(x) {
  pins <- scan(what = integer(), sep = " ", quiet = TRUE,
               text = sub("^[a-zA-Z_ \']+(.*$)", "\\1", x))
  results <- matrix(NA_integer_, 10, 2)
  pos <- 1
  for (f in 1:10) {
    x1 <- pins[pos]
    if (x1 == 10) results[f, ] <- c(x1, 0L)
    else {
      pos <- pos + 1
      x2 <- pins[pos]
      results[f, ] <- c(x1, x2)
    }
    pos <- pos + 1
  }
  return(results)
}) # 30 element list each with a 10x2 integer array of pins knocked down
```

Dirichlet Distribution

- Dirichlet distribution is over the parameter space of PMFs — i.e. $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$ — and the Dirichlet PDF is $f\left(\boldsymbol{\pi} \mid \boldsymbol{\alpha}\right) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$ where $\alpha_k \geq 0 \, \forall k$ and the multivariate Beta function is $B\left(\boldsymbol{\alpha}\right) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma\left(\prod_{k=1}^{K} \alpha_k\right)}$ where $\Gamma\left(z\right) = \frac{1}{z} \prod_{n=1}^{\infty} \frac{\left(1+\frac{1}{n}\right)^n}{1+\frac{z}{n}} = \int_0^\infty u^{z-1} e^{-u} du$ is the Gamma function
- $\mathbb{E}\pi_i = \frac{\alpha_i}{\sum_{k=1}^{K} \alpha_k} \forall i$ and the mode of π_i is $\frac{\alpha_i - 1}{-1+\sum_{k=1}^{K} \alpha_k}$ if $\alpha_i > 1$
- Iff $\alpha_k = 1 \, \forall k$, $f\left(\boldsymbol{\pi} \middle| \, \boldsymbol{\alpha} = \mathbf{1}\right)$ is constant over Θ (simplexes)
- Beta distribution is a special case of the Dirichlet where $K = 2$
- Marginal and conditional distributions for subsets of $\boldsymbol{\pi}$ are also Dirichlet

Multilevel Stan Program for Bowling

```
#include bowling_kernel.stan
data { // exogenous and endogenous knowns
  int<lower = 0> J;                             // number of bowlers
  int<lower = 0, upper = 10> x1_x2[J, 10, 2]; // results of each bowler's frames
  vector<lower = 0>[11] a;                      // shapes for Dirichlet prior on mu
  real<lower = 0> s;                            // scale factor on top of theta
}
parameters { // exogenous unknowns
  simplex[11] mu;        // overall probability of knocking down 0:10 pins
  real<lower = 0> theta; // concentration parameter across bowlers
  simplex[11] pi[J];     // bowler's probability of knocking down 0:10 pins
}
model { // target becomes the log-numerator of Bayes Rule
  vector[11] mu_theta = mu * theta * s;                  // not saved in results
  for (j in 1:J) // bowling_kernel() is defined in the functions block
    target += bowling_kernel(pi[j], mu_theta, x1_x2[j]); // note indexing
  target += dirichlet_lpdf(mu | a);                      // prior on mu
  target += exponential_lpdf(theta | 1);                 // prior on theta
}
```

Multilevel Posterior Distribution

```r
post_mlm <- stan("bowling_mlm.stan", control = list(adapt_delta = 0.85), refresh = 0,
                 data = list(J = length(x1_x2), x1_x2 = x1_x2, a = 1:11, s = 10))
```

```r
print(post_mlm, pars = "pi", include = FALSE, digits = 2)
```

```
## Inference for Stan model: bowling_mlm.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##            mean se_mean    sd     2.5%     25%     50%     75%   97.5% n_eff Rhat
## mu[1]      0.00     0.0  0.00     0.00    0.00    0.00    0.00    0.00   235 1.01
## mu[2]      0.01     0.0  0.00     0.01    0.01    0.01    0.01    0.02   448 1.00
## mu[3]      0.02     0.0  0.00     0.01    0.01    0.02    0.02    0.03   744 1.00
## mu[4]      0.02     0.0  0.00     0.01    0.01    0.02    0.02    0.03   605 1.01
## mu[5]      0.02     0.0  0.01     0.01    0.01    0.02    0.02    0.03   442 1.00
## mu[6]      0.03     0.0  0.01     0.01    0.02    0.02    0.03    0.04   555 1.01
## mu[7]      0.04     0.0  0.01     0.02    0.03    0.04    0.05    0.07   607 1.01
## mu[8]      0.08     0.0  0.01     0.06    0.07    0.08    0.09    0.11   682 1.00
## mu[9]      0.13     0.0  0.02     0.10    0.12    0.13    0.14    0.17   448 1.01
## mu[10]     0.24     0.0  0.02     0.19    0.22    0.24    0.25    0.28   623 1.00
## mu[11]     0.42     0.0  0.03     0.37    0.40    0.42    0.44    0.47   604 1.01
## theta      7.36     0.2  2.30     3.74    5.74    7.02    8.68   12.82   131 1.02
## lp__    -918.82     5.7 51.38 -1029.02 -951.29 -917.53 -883.48 -823.13    81 1.03
```

Pairs Plot

```
pairs(post_mlm, pars = c("mu", "pi"), include = FALSE)
```