Another multi-level credit-assignment problem

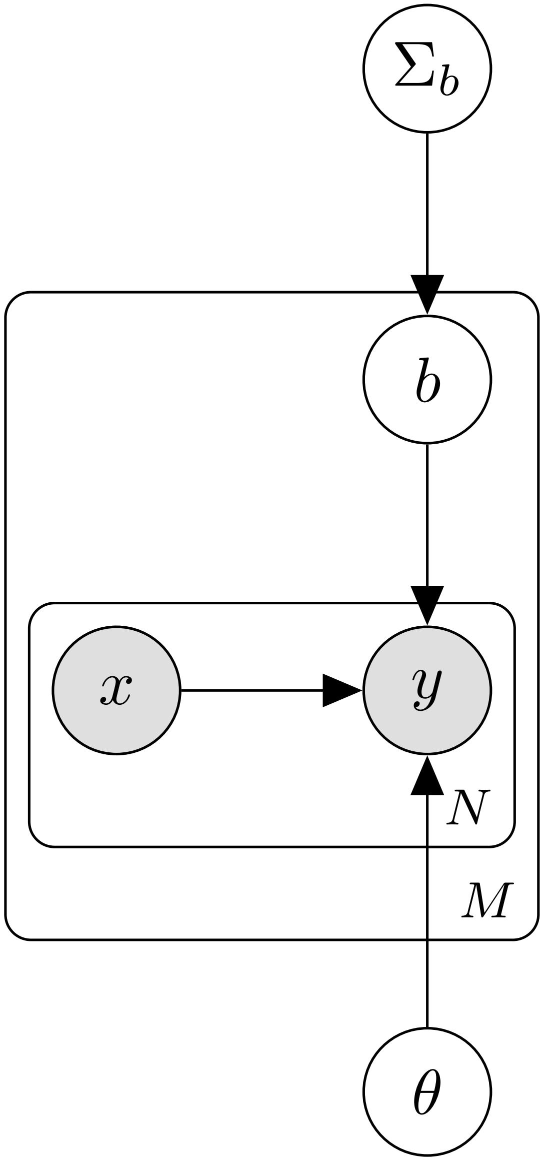
- Scenario: a new tutoring method is being compared with an older tutoring method
- Experiment design:
  - 6 tutoring sites
  - 4 tutored students at each site
  - No repeated measures from either students or tutors
  - Student:tutor pairs randomly assigned to tutoring method
  - Dependent variable: does the student pass a test?

R implementation of experiment design

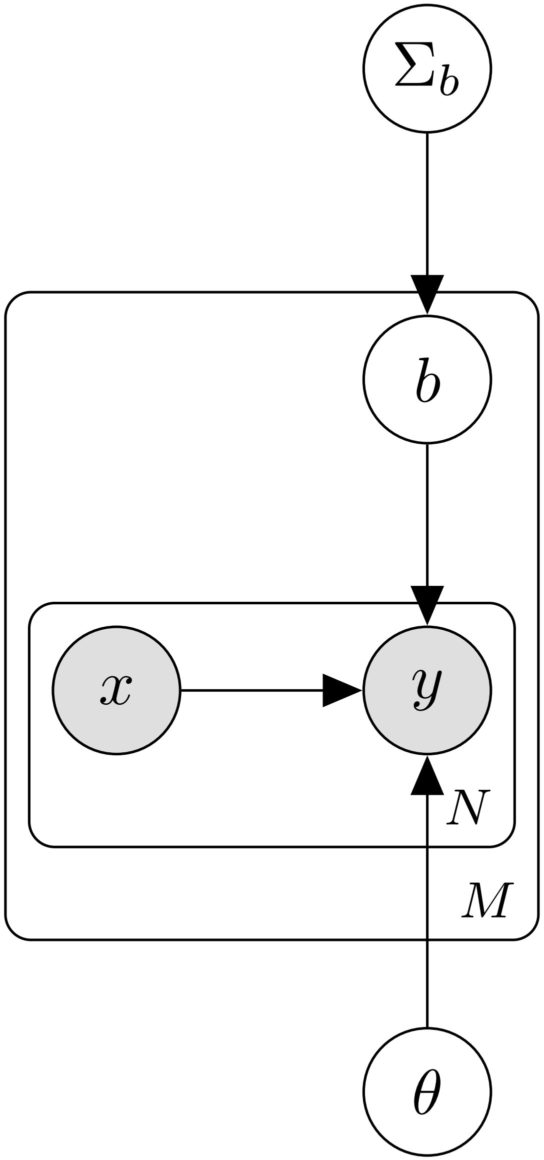
```r
library(dplyr) # Load necessary libraries

set.seed(10) # Set seed for reproducibility

# Number of sites and students per site
n_sites <- 6   # Small number of sites to induce variance uncertainty
n_students_per_site <- 4  # Each student is assigned to ONE teaching method

# Generate student-level data
dat <- expand.grid(site_id = 1:n_sites, student_id = 1:n_students_per_site)

# Assign each student to a teaching method (ensuring between-subjects design)
dat <- dat %>%
  group_by(site_id) %>%
  mutate(method = sample(0:1, size = n(), replace = TRUE)) %>%
  ungroup()
```

```
> head(dat,n=8)
# A tibble: 8 × 7
  site_id student_id method
    <int>      <int>  <int>
1       1          1      0
2       2          1      1
3       3          1      0
4       4          1      0
5       5          1      1
6       6          1      0
7       1          2      0
8       2          2      0
```

Mixed logit model assumption

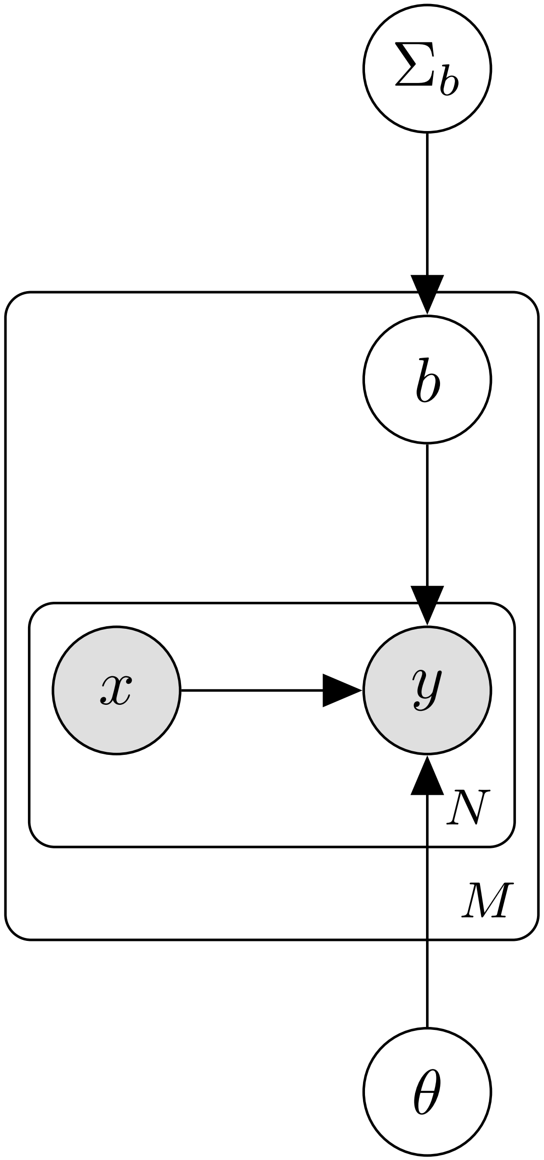


Mixed logit model assumption



$$b \sim N(0,\Sigma_b)$$

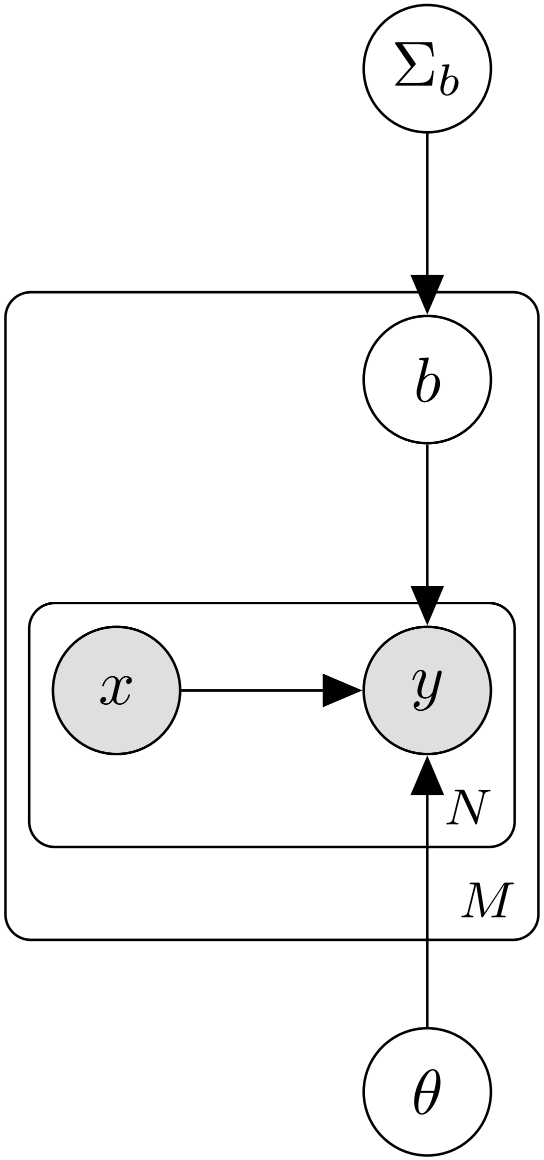
Mixed logit model assumption



$$b \sim N(0, \Sigma_b)$$

$$\eta = \left(\beta + b\right) x$$

Mixed logit model assumption

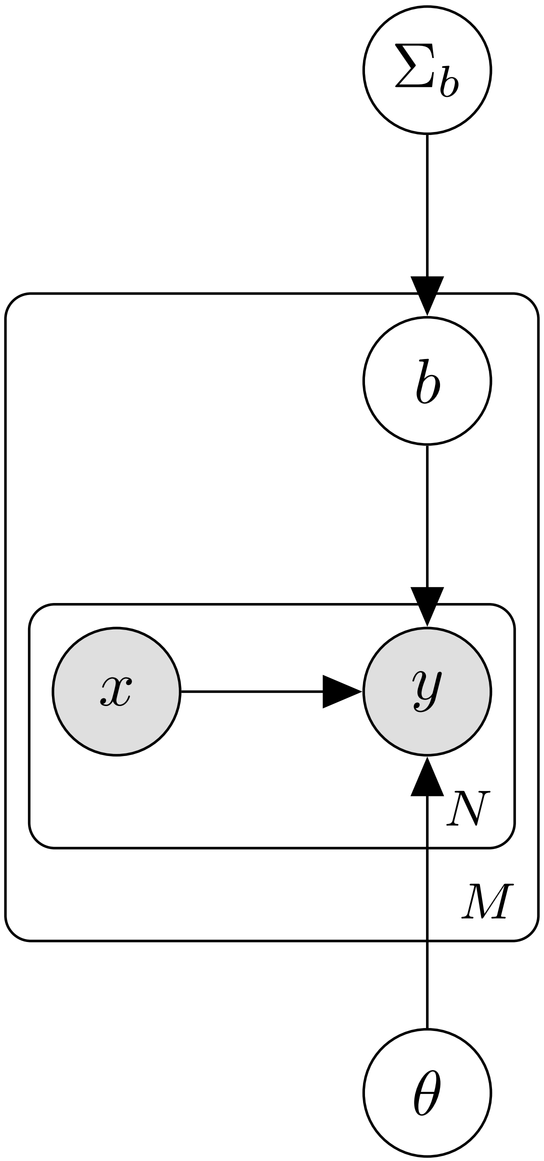


$$b \sim N(0, \Sigma_b)$$

$$\eta = \left(\beta + b\right) x$$

$$\hat{y} = l^{-1}(\eta) = \frac{e^{\eta}}{1 + e^{\eta}}$$

Mixed logit model assumption



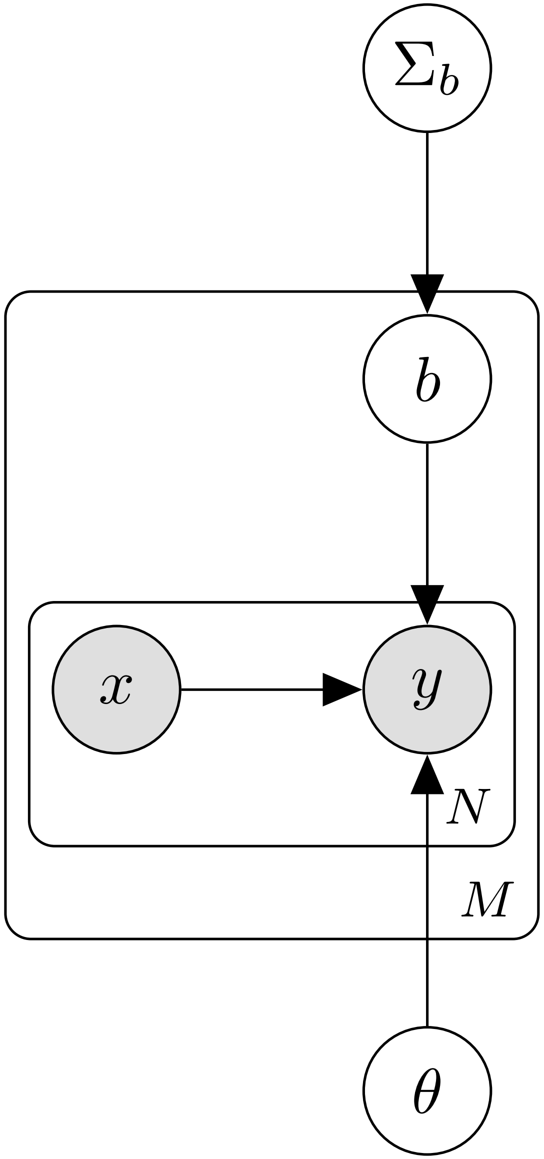
$$b \sim N(0, \Sigma_b)$$

$$\eta = \left(\beta + b\right) x$$

$$\hat{y} = l^{-1}(\eta) = \frac{e^{\eta}}{1 + e^{\eta}}$$

$$P(y \mid \hat{y}) = \begin{cases} \hat{y} & y = 1 \\ 1 - \hat{y} & y = 0 \\ 0 & \text{otherwise} \end{cases}$$

Mixed logit model assumption



$$b \sim N(0, \Sigma_b)$$

$$\eta = \left(\beta + b\right) x$$

$$\hat{y} = l^{-1}(\eta) = \frac{e^{\eta}}{1 + e^{\eta}}$$

$$P(y \mid \hat{y}) = \begin{cases} \hat{y} & y = 1 \\ 1 - \hat{y} & y = 0 \\ 0 & \text{otherwise} \end{cases}$$

For our case, $\eta = \alpha + \beta X + b$, where X is a {0,1} dummy variable

R implementation of data generation

```r
# True parameters
beta_0 <-      [redacted]# Intercept (baseline pass probability)
beta_method <- [redacted]# Positive effect of the new teaching method
sigma_site <-  [redacted]# Variance of site random effects

# Generate site-level random intercepts
sites <- data.frame(site_id = 1:n_sites,
                    u = rnorm(n_sites, mean = 0, sd = sigma_site))

# Merge with site random effects
dat <- left_join(dat, sites, by = "site_id")

# Compute log-odds of passing
dat$logit_p <- beta_0 + beta_method * dat$method + dat$u
dat$p <- exp(dat$logit_p) / (1 + exp(dat$logit_p))  # Log-odds -> probability

# Simulate pass/fail outcome
dat$passed <- rbinom(nrow(data), 1, dat$p)
```

R implementation of data generation

```r
# True parameters
beta_0 <-      [redacted] # Intercept (baseline pass probability)
beta_method <- [redacted] # Positive effect of the new teaching method
sigma_site <-  [redacted] # Variance of site random effects

# Generate site-level random intercepts
sites <- data.frame(site_id = 1:n_sites,
                    u = rnorm(n_sites, mean = 0, sd = sigma_site))

# Merge with site random effects
dat <- left_join(dat, sites, by = "site_id")

# Compute log-odds of passing
dat$logit_p <- beta_0 + beta_method * dat$method + dat$u
dat$p <- exp(dat$logit_p) / (1 + exp(dat$logit_p))  # Log-odds -> probability

# Simulate pass/fail outcome
dat$passed <- rbinom(nrow(data), 1, dat$p)
```

```
> head(dat,n=8)
# A tibble: 8 × 7
  site_id student_id method      u logit_p     p passed
    <int>      <int>  <int>  <dbl>   <dbl> <dbl>  <int>
1       1          1      0 -0.476 -0.176  0.456      1
2       2          1      1  1.97   2.27   0.907      0
3       3          1      0  1.48   1.78   0.856      1
4       4          1      0  0.179  0.479  0.617      1
5       5          1      1 -1.91  -1.61   0.167      0
6       6          1      0 -0.390 -0.0903 0.477     0
7       1          2      0 -0.476 -0.176  0.456      0
8       2          2      0  1.97   2.27   0.907      1
```

Some summary statistics

The *"raw statistics"* suggest that the new teaching method ($X = 1$) leads to a lower pass rate:

```
> dat %>%
+   group_by(method) %>%
+   summarize(mean_passed=mean(passed))
# A tibble: 2 × 2
  method mean_passed
   <int>       <dbl>
1      0       0.667
2      1       0.25
```

Some summary statistics

The *"raw statistics"* suggest that the new teaching method ($X = 1$) leads to a lower pass rate:

```
> dat %>%
+   group_by(method) %>%
+   summarize(mean_passed=mean(passed))
# A tibble: 2 × 2
  method mean_passed
   <int>       <dbl>
1      0       0.667
2      1       0.25
```

But there is also site-level variation in both rate of use of the new teaching method and pass rate:

```
> dat1 <- dat %>%
+   group_by(site_id) %>%
+   summarize(mean_new_method=mean(method),mean_passed=mean(passed))
> print(dat1)
# A tibble: 6 × 3
  site_id mean_new_method mean_passed
    <int>           <dbl>       <dbl>
1       1            0.5         0.25
2       2            0.75        0.75
3       3            0.25        1
4       4            0.25        0.75
5       5            0.75        0
6       6            0.5         0
```

Some summary statistics

The *"raw statistics"* suggest that the new teaching method ($X = 1$) leads to a lower pass rate:

```
> dat %>%
+   group_by(method) %>%
+   summarize(mean_passed=mean(passed))
# A tibble: 2 x 2
  method mean_passed
   <int>       <dbl>
1      0       0.667
2      1       0.25
```

But there is also site-level variation in both rate of use of the new teaching method and pass rate:

```
> dat1 <- dat %>%
+   group_by(site_id) %>%
+   summarize(mean_new_method=mean(method),mean_passed=mean(passed))
> print(dat1)
# A tibble: 6 x 3
  site_id mean_new_method mean_passed
    <int>           <dbl>       <dbl>
1       1            0.5         0.25
2       2            0.75        0.75
3       3            0.25        1
4       4            0.25        0.75
5       5            0.75        0
6       6            0.5         0
```

This is *exactly* the case where we want to fit a multi-level model to work out credit assignment

MLE fitting the mixed-logit model

```r
# Fit mixed logit model (random intercept for site)
library(lme4)
model <- glmer(passed ~ method + (1 | site_id), data = dat, family = binomial)
```

```
> summary(model)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: passed ~ method + (1 | site_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
    28.6     32.2    -11.3     22.6       21

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.18452 -0.25342 -0.08034  0.35771  0.97355

Random effects:
 Groups  Name        Variance Std.Dev.
 site_id (Intercept) 9.343    3.057
Number of obs: 24, groups:  site_id, 6

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.031      1.689    0.61    0.542
method        -3.261      2.039   -1.60    0.110

Correlation of Fixed Effects:
       (Intr)
method -0.427
```

Problem with interpretation of ML fit!

```
Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.031      1.689    0.61    0.542
method        -3.261      2.039   -1.60    0.110
```

- The Wald Z is $\widehat{\beta}/\mathrm{SE}(\widehat{\beta})$, and this is computed **using a point estimate of the random effects covariance matrix** $\widehat{\Sigma}_b$**!**
- The data leave us uncertain about not only β but also Σ_b, but this uncertainty is not taken into account by the Wald Z statistic
- This motivates the use of inferential methods that **take into account the collective uncertainty about all model parameters**
- One powerful technique to do this is **Bayesian inference**

Unnormalizable posteriors

*For now, θ denotes **all** model parameters, not just the fixed effects*

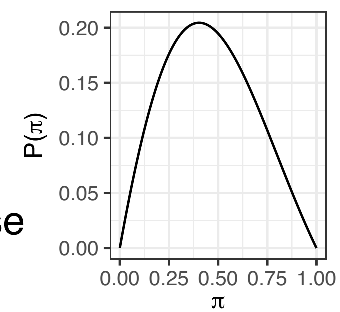
- Our motivation: Bayesian posterior inference

$$P(\theta \mid \mathbf{y}, I) = \frac{P(\mathbf{y} \mid \theta, I) P(\theta \mid I)}{P(\mathbf{y} \mid I)}$$

- Sometimes $P(\mathbf{y} \mid I)$ can't be calculated exactly. Example

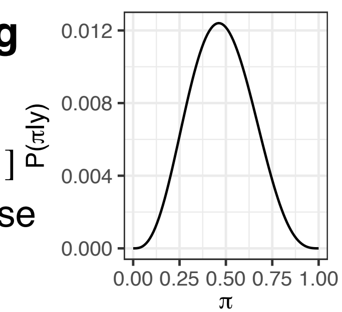
Bernoulli data with non-conjugate prior:

$$P(\pi) \propto \begin{cases} \pi(1-\pi)e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

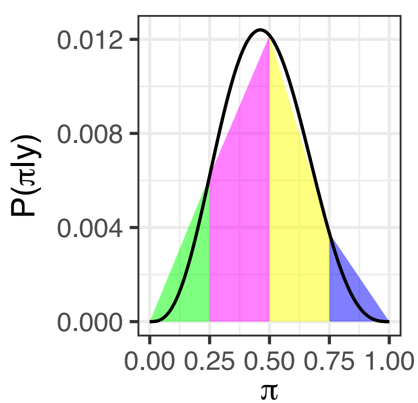


Posterior after observing 2 heads, 2 tails:

$$P(\pi) \propto \begin{cases} \pi^3(1-\pi)^3 e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$



- In simple cases like this, we can numerically approximate the integral:



- But in high dimension and/or unbounded ranges, difficult or even impossible!

Unnormalizable posteriors

*For now, θ denotes **all** model parameters, not just the fixed effects*

- Our motivation: Bayesian posterior inference

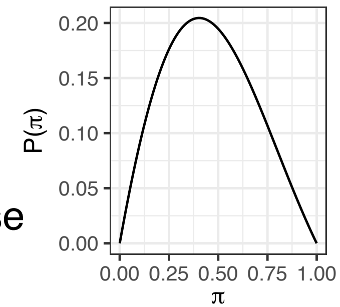
$$P(\theta \mid \mathbf{y}, I) = \frac{P(\mathbf{y} \mid \theta, I) P(\theta \mid I)}{P(\mathbf{y} \mid I)}$$

Model parameters

- Sometimes $P(\mathbf{y} \mid I)$ can't be calculated exactly. Example

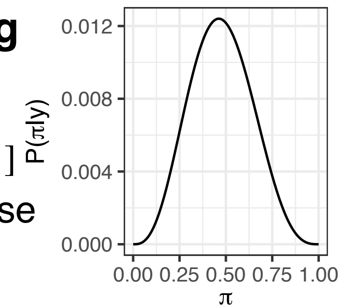
Bernoulli data with non-conjugate prior:

$$P(\pi) \propto \begin{cases} \pi(1-\pi)e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

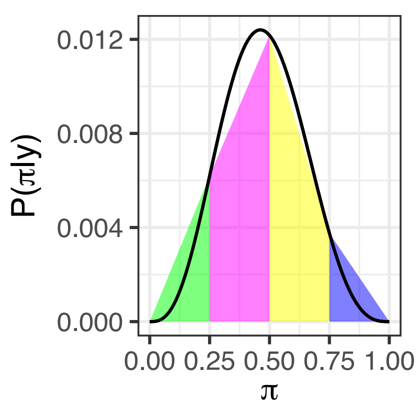


Posterior after observing 2 heads, 2 tails:

$$P(\pi) \propto \begin{cases} \pi^3(1-\pi)^3 e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$



- In simple cases like this, we can numerically approximate the integral:



- But in high dimension and/or unbounded ranges, difficult or even impossible!

Unnormalizable posteriors

*For now, θ denotes **all** model parameters, not just the fixed effects*

- Our motivation: Bayesian posterior inference

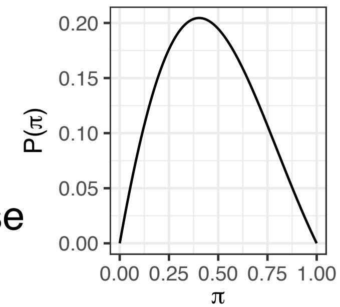
$$P(\theta \mid \mathbf{y}, I) = \frac{P(\mathbf{y} \mid \theta, I) P(\theta \mid I)}{P(\mathbf{y} \mid I)}$$

Model parameters ***Background knowledge***

- Sometimes $P(\mathbf{y} \mid I)$ can't be calculated exactly. Example

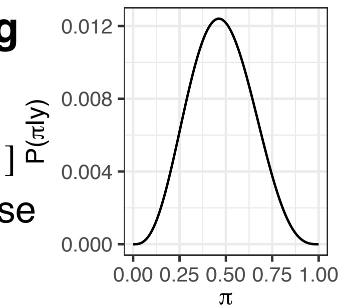
Bernoulli data with non-conjugate prior:

$$P(\pi) \propto \begin{cases} \pi(1-\pi)e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

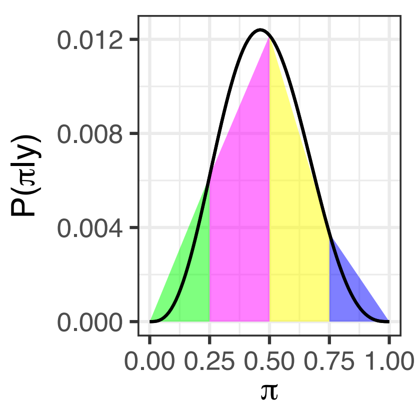


Posterior after observing 2 heads, 2 tails:

$$P(\pi) \propto \begin{cases} \pi^3(1-\pi)^3 e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$



- In simple cases like this, we can numerically approximate the integral:



- But in high dimension and/or unbounded ranges, difficult or even impossible!

Unnormalizable posteriors

*For now, θ denotes **all** model parameters, not just the fixed effects*

- Our motivation: Bayesian posterior inference

Observed data

$$P(\theta \mid \mathbf{y}, I) = \frac{P(\mathbf{y} \mid \theta, I)P(\theta \mid I)}{P(\mathbf{y} \mid I)}$$

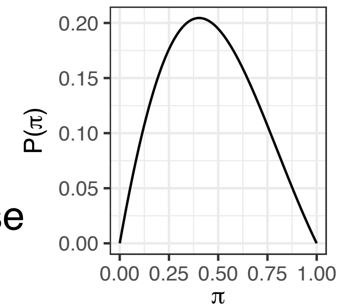
Model parameters

Background knowledge

- Sometimes $P(\mathbf{y} \mid I)$ can't be calculated exactly. Example

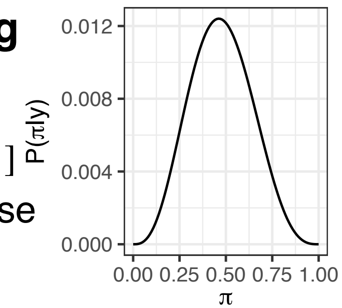
Bernoulli data with non-conjugate prior:

$$P(\pi) \propto \begin{cases} \pi(1-\pi)e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

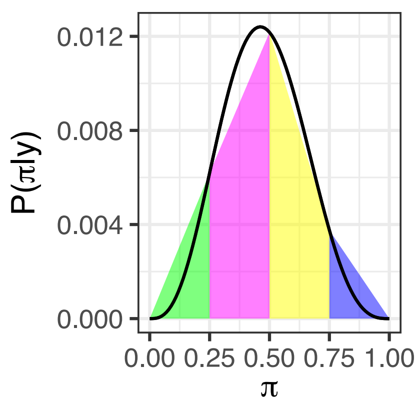


Posterior after observing 2 heads, 2 tails:

$$P(\pi) \propto \begin{cases} \pi^3(1-\pi)^3e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$



- In simple cases like this, we can numerically approximate the integral:



- But in high dimension and/or unbounded ranges, difficult or even impossible!

Unnormalizable posteriors

*For now, θ denotes **all** model parameters, not just the fixed effects*

- Our motivation: Bayesian posterior inference

Observed data

$$P(\theta \mid \mathbf{y}, I) = \frac{P(\mathbf{y} \mid \theta, I)P(\theta \mid I)}{P(\mathbf{y} \mid I)}$$

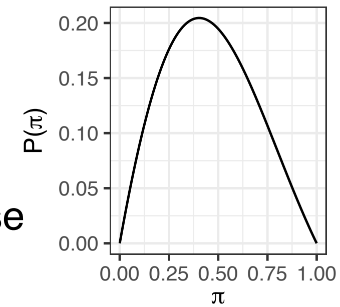
Model parameters

Background knowledge

- Sometimes $P(\mathbf{y} \mid I)$ can't be calculated exactly. Example

Bernoulli data with non-conjugate prior:

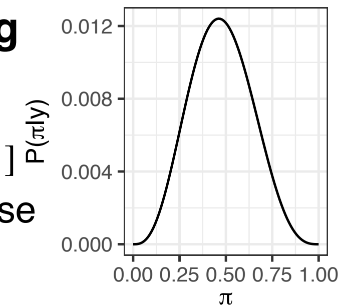
$$P(\pi) \propto \begin{cases} \pi(1-\pi)e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$



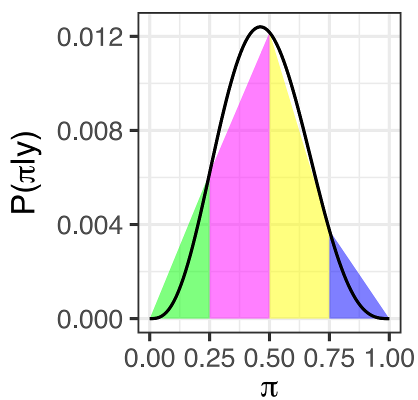
Posterior after observing 2 heads, 2 tails:

$$P(\pi) \propto \begin{cases} \pi^3(1-\pi)^3e^{-\pi^2} & \pi \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

No closed form!



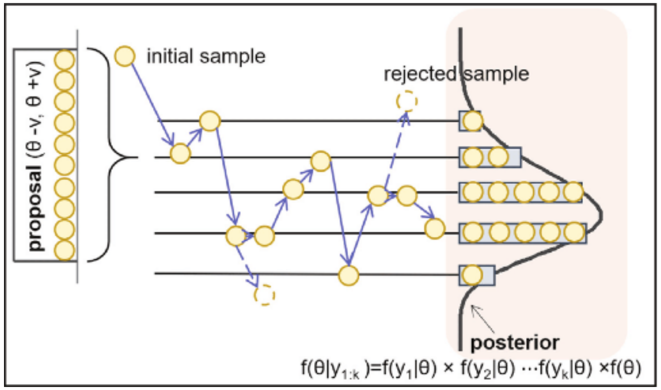
- In simple cases like this, we can numerically approximate the integral:



- But in high dimension and/or unbounded ranges, difficult or even impossible!

Markov chain Monte Carlo

- However, we can often **take samples from the posterior** even when we can't compute normalized probabilities
- One general and widely used approach: **Markov chain Monte Carlo (MCMC)**
- MCMC is a mathematically principled random walk on a non-negative function, directed toward regions where the function takes on a larger value



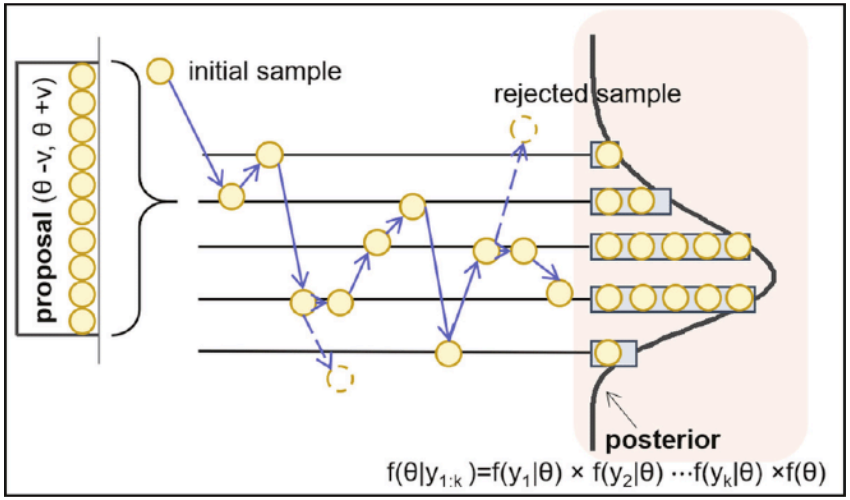
- Asymptotically, the random walk gives us samples from in proportion to the height of the function

Image credit: Dong et al., 2019, http://dx.doi.org/10.5772/intechopen.82781

MCMC for posterior sampling

- We use the *unnormalized* form of the posterior:

$$P(\theta \,|\, \mathbf{y}, I) \propto P(\mathbf{y} \,|\, \theta, I) P(\theta \,|\, I)$$

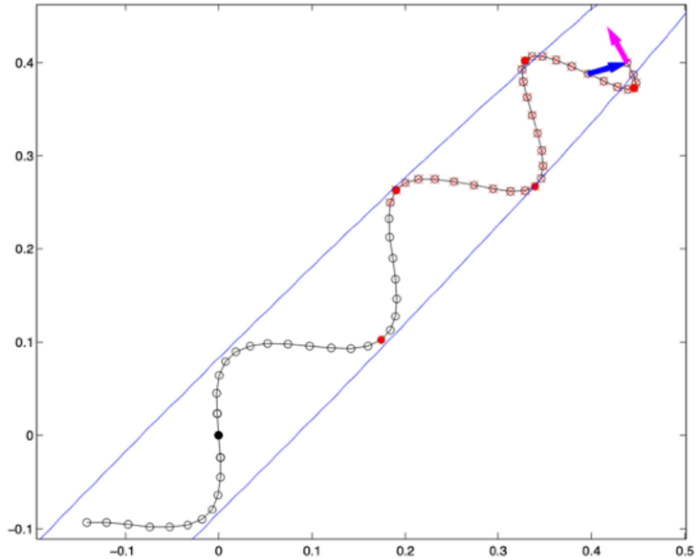


- We run MCMC and then treat the chain of values as samples from the posterior
- The full set of samples is not iid (nearby values on the chain are correlated), but methods exist for estimating "effectively" how many independent samples we have

Image credit: Dong et al., 2019, http://dx.doi.org/10.5772/intechopen.82781

Stan, HMC, and NUTS

- There are many different MCMC algorithms (e.g., Metropolis, Gibbs Sampling)
- We will use the probabilistic programming language **Stan** for Bayesian inference about model parameters
- Stan uses an algorithm called **Hamiltonian Monte Carlo** (**HMC**) with the **No U-Turn Sampler (NUTS)**



(Hoffman & Gelman, 2014)

- This algorithm tends to be particularly efficient for many problems we'll face

Bayesian posterior inference with Stan

Bayesian posterior inference with Stan

1. Define the generative model you assume underlies the data you want to analyze

Bayesian posterior inference with Stan

1. Define the generative model you assume underlies the data you want to analyze
2. Choose a prior distribution for your model parameters θ

Bayesian posterior inference with Stan

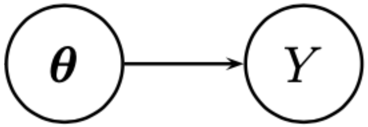
1. Define the generative model you assume underlies the data you want to analyze
2. Choose a prior distribution for your model parameters θ
3. Encode the model structure and prior in a Stan program

Bayesian posterior inference with Stan

1. Define the generative model you assume underlies the data you want to analyze
2. Choose a prior distribution for your model parameters θ
3. Encode the model structure and prior in a Stan program
4. Provide the data Y you want to analyze, and ask Stan to sample from the posterior $P(\theta \mid Y, I) \propto P(Y \mid \theta) P(\theta \mid I)$ (often written as $P(\theta \mid Y) \propto P(Y \mid \theta) P(\theta)$, i.e. eliding I)

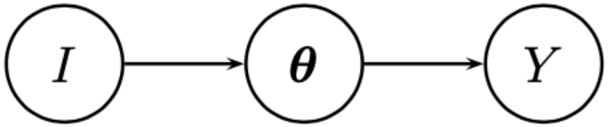
Bayesian posterior inference with Stan

1. Define the generative model you assume underlies the data you want to analyze
2. Choose a prior distribution for your model parameters θ
3. Encode the model structure and prior in a Stan program
4. Provide the data Y you want to analyze, and ask Stan to sample from the posterior $P(\theta \mid Y, I) \propto P(Y \mid \theta) P(\theta \mid I)$ (often written as $P(\theta \mid Y) \propto P(Y \mid \theta) P(\theta)$, i.e. eliding I)



Bayesian posterior inference with Stan

1. Define the generative model you assume underlies the data you want to analyze
2. Choose a prior distribution for your model parameters θ
3. Encode the model structure and prior in a Stan program
4. Provide the data Y you want to analyze, and ask Stan to sample from the posterior $P(\theta \mid Y, I) \propto P(Y \mid \theta)P(\theta \mid I)$ (often written as $P(\theta \mid Y) \propto P(Y \mid \theta)P(\theta)$, i.e. eliding I)



A Stan model

```
data {
  int<lower = 0> n;  // Total number of trials
  int<lower = 0> r;  // number of successes
}
parameters {
  real<lower = 0, upper = 1> p;
}
model {
  // Prior:
  p ~ beta(1,1);
  // Likelihood
  r ~ binomial(n,p);
}
```

Likelihood ratio test more robust than Wald Z

```
> model <- glmer(passed ~ method + (1 | site_id), data = dat, family = binomial)
> m0 <- glmer(passed ~ 1 + (1 | site_id), data = dat, family = binomial)
> anova(m0,model)
Data: dat
Models:
m0: passed ~ 1 + (1 | site_id)
model: passed ~ method + (1 | site_id)
      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
m0       2 31.482 33.838 -13.741   27.482
model    3 28.636 32.170 -11.318   22.636 4.8455  1    0.02772 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```