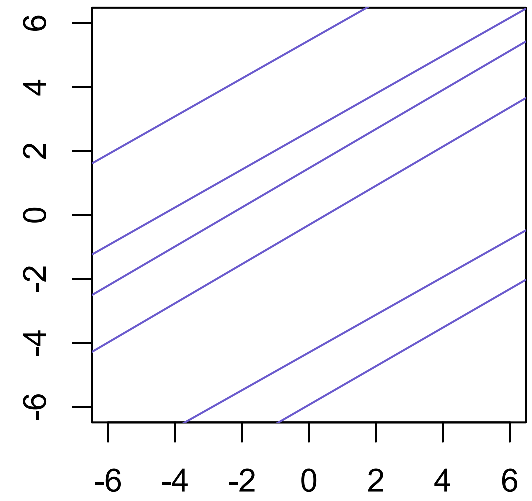
Statistical Rethinking
Winter 2019

Lecture 17 / Week 9

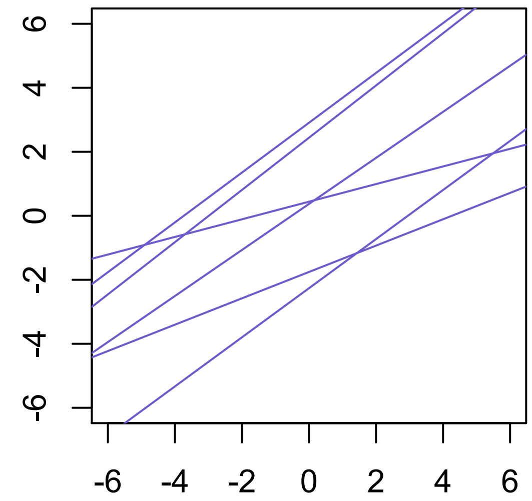
Adventures in Covariance

Kinds of varying effects

- *Varying intercepts*: means differ by cluster



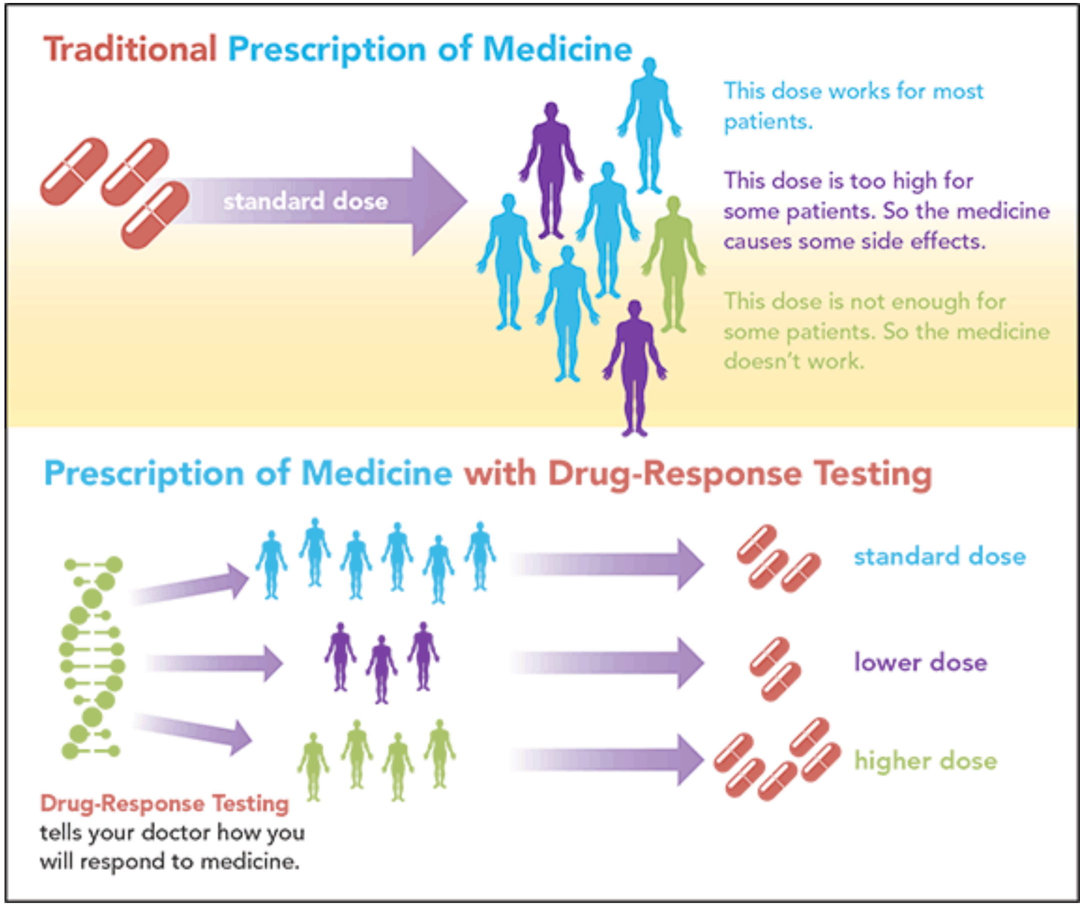
- *Varying slopes*: effects of predictors vary by cluster



- Any parameter can be made into a varying effect
  - (1) split into vector of parameters by cluster
  - (2) define population distribution

Varying slopes

- Why varying slopes?
  - drugs affect people differently
  - after school programs don't work for everyone
  - not every unit has same relationship to predictor
  - variation is important, whether for intervention or inference
- *Average* effect misleading?
- Pooling, shrinkage, mnesia

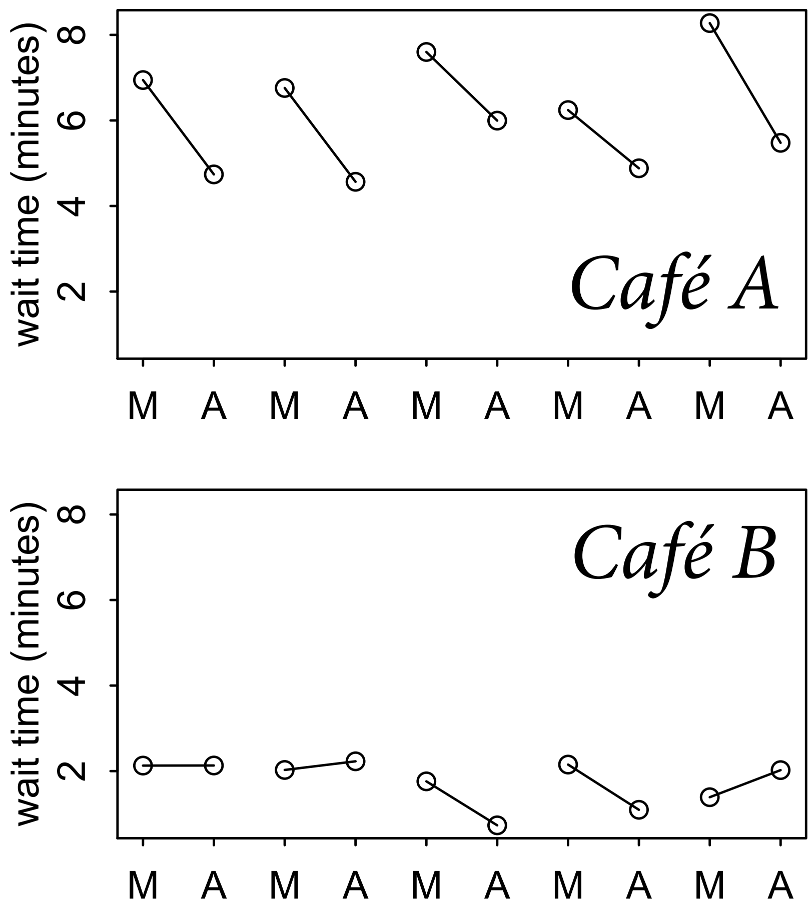


Population pooling

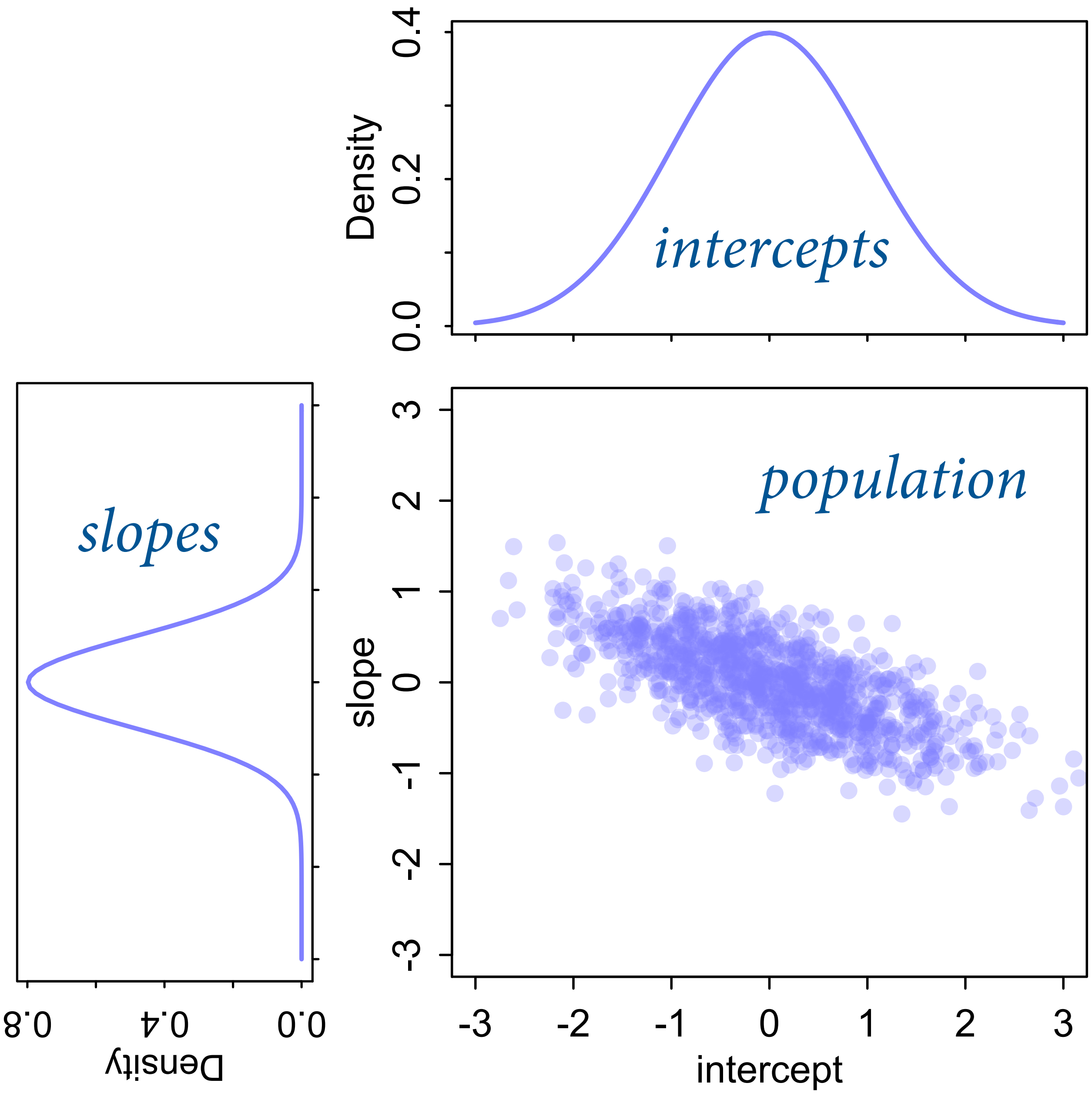
- Major innovation with varying slopes is **pooling across parameters (intercepts & slopes)**
- Features of units have correlation structure
- Learn one feature —> info about other features
- e.g. Higher intercepts associated with smaller slopes

Café Robot

- Robot programmed to visit cafés, order coffee, record wait time
- Visits in *morning* and *afternoon*
- Intercepts: avg morning wait
- Slopes: avg difference btw afternoon and morning
- Are intercepts and slopes related?
    - Yes => pooling across parameter types!



Population of Cafés



Population of Cafés

- 2-dimensional Gaussian distribution
  - vector of means
  - variance-covariance matrix

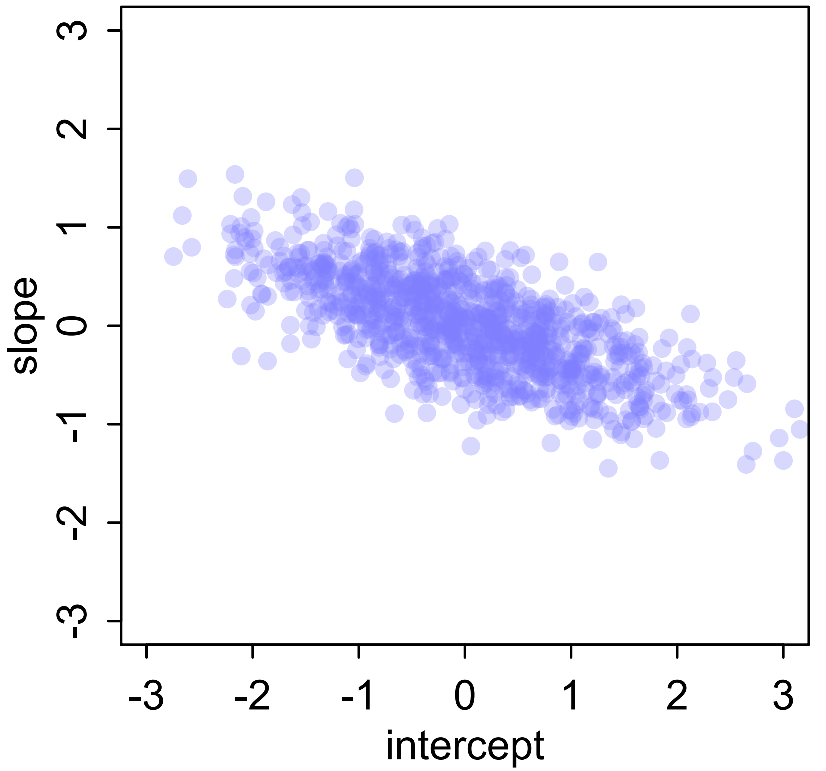
intercepts variance

$$\begin{pmatrix} \sigma_\alpha^2 & \sigma_\alpha \sigma_\beta \rho \\ \boxed{\sigma_\alpha \sigma_\beta \rho} & \sigma_\beta^2 \end{pmatrix}$$

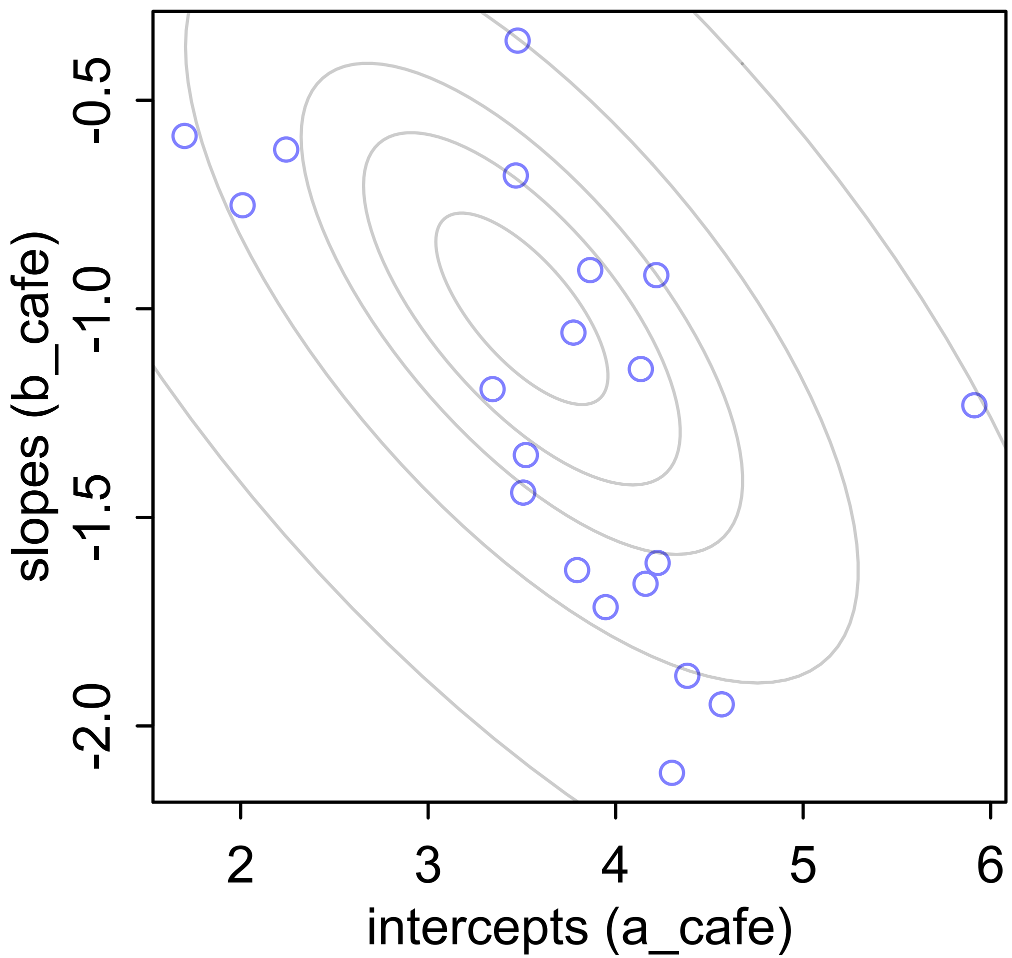
covariance

correlation

slopes variance



Simulated Cafés



20 cafés

5 days
morning & afternoon

200 observations

Varying slopes model

$$
\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \\
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}
$$

$$
\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \\
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}
$$

varying intercepts

varying slopes

$$
\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \\
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}
$$

multivariate prior

covariance matrix

pop avg intercept

$$\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right)
\end{aligned}$$

pop avg slope

$$\begin{aligned}
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}$$

Covariance matrix shuffle

- m-by-m covariance matrix requires estimating
  - m standard deviations (or variances)
  - $(m^2 - m)/2$ correlations (for covariances)
  - total of $m(m + 1)/2$ parameters

Covariance matrix shuffle

- m-by-m covariance matrix requires estimating
  - m standard deviations (or variances)
  - $(m^2 - m)/2$ correlations (for covariances)
  - total of $m(m + 1)/2$ parameters
- Several ways specify priors
  - Conjugate: inverse-Wishart (`inv_wishart`)
  - inverse-Wishart cannot pull apart stddev and correlations
  - Better to decompose:

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \underbrace{\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}}_{\mathbf{R}} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

Matrixes are nice

- Matrix algebra just shortcuts for working with lists of numbers
- A few simple rules
- Can you make an omelet?
  You can multiply matrixes.

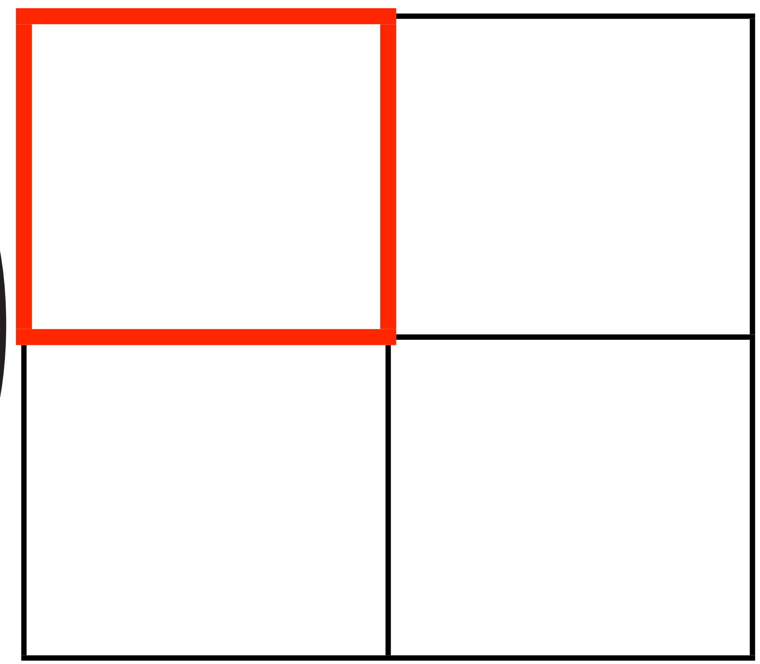
Case	Singular	Plural
nominative	**mātrīx**	mātrīcēs
genitive	mātrīcis	mātrīcum
dative	mātrīcī	mātrīcibus
accusative	mātrīcem	mātrīcēs
ablative	mātrīce	mātrīcibus
vocative	**mātrīx**	mātrīcēs

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

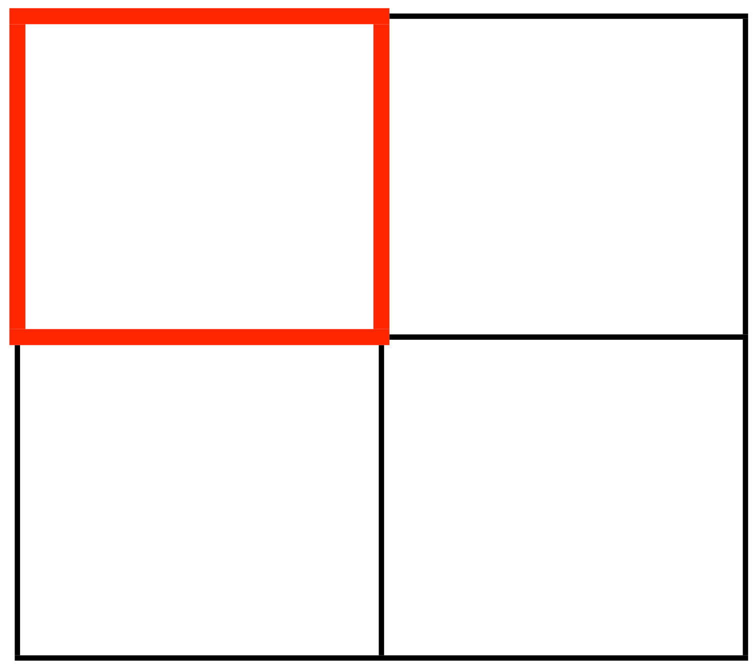
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

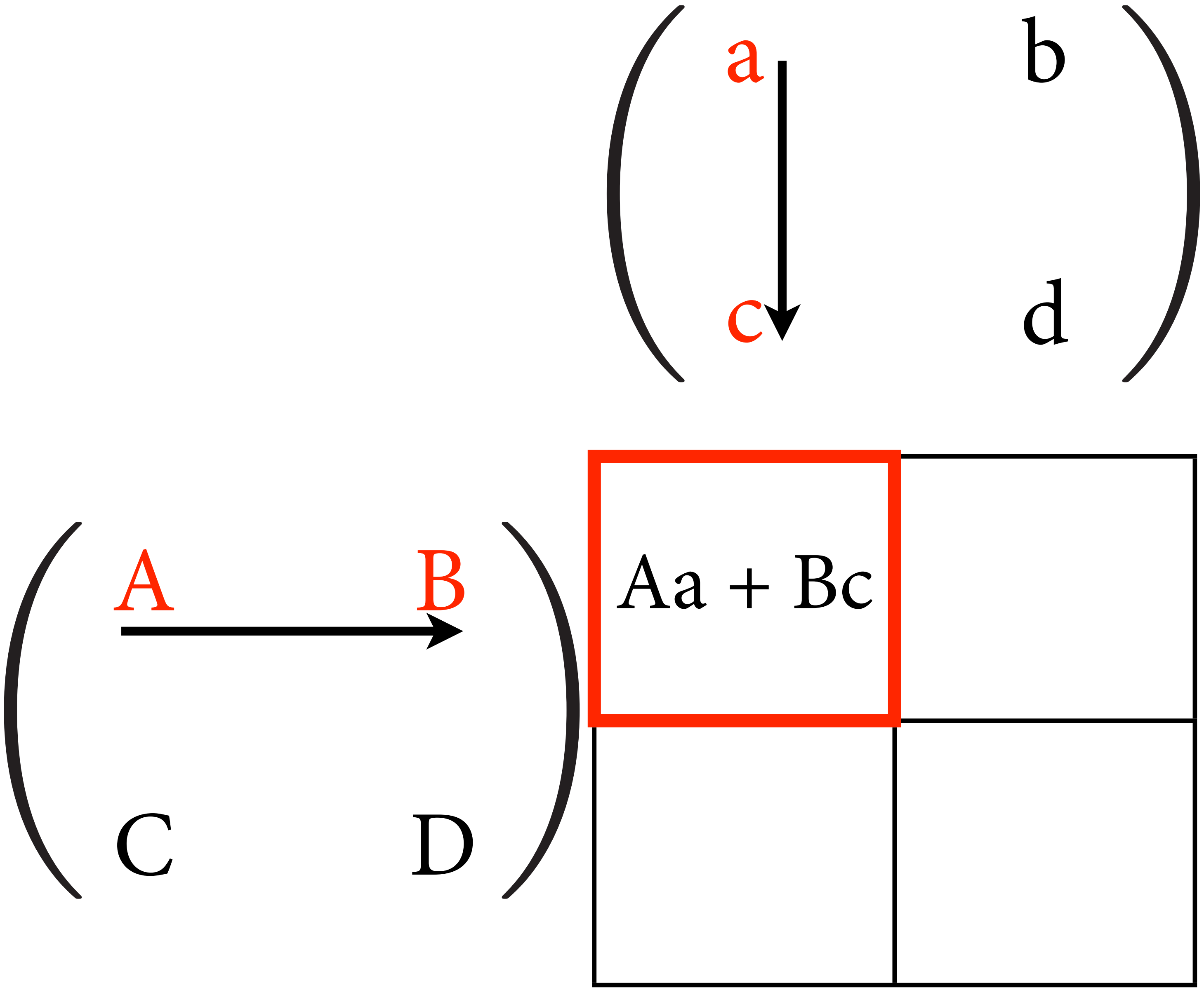
$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$



$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$





a
b
c
d
A
B
C
D
Aa + Bc

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

Aa + Bc	Ab + Bd
Ca + Dc	Cb + Dd

Matrixes are nice

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} = \mathbf{SRS}$$

Matrixes are nice

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} = \mathbf{SRS}$$

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

$$\begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

?

Matrixes are nice

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} = \mathbf{SRS}$$

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

$$\begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \quad \begin{pmatrix} \sigma_\alpha & \rho\sigma_\alpha \\ \rho\sigma_\beta & \sigma_\beta \end{pmatrix}$$

Matrixes are nice

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} = \mathbf{SRS}$$

$$\begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

$$\begin{pmatrix} \sigma_\alpha & \rho\sigma_\alpha \\ \rho\sigma_\beta & \sigma_\beta \end{pmatrix}$$

?

Matrixes are nice

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} = \mathbf{SRS}$$

$$\begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

$$\begin{pmatrix} \sigma_\alpha & \rho\sigma_\alpha \\ \rho\sigma_\beta & \sigma_\beta \end{pmatrix} \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix}$$

$$
\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \\
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}
$$

build cov matrix

$$
\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \\
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}
$$

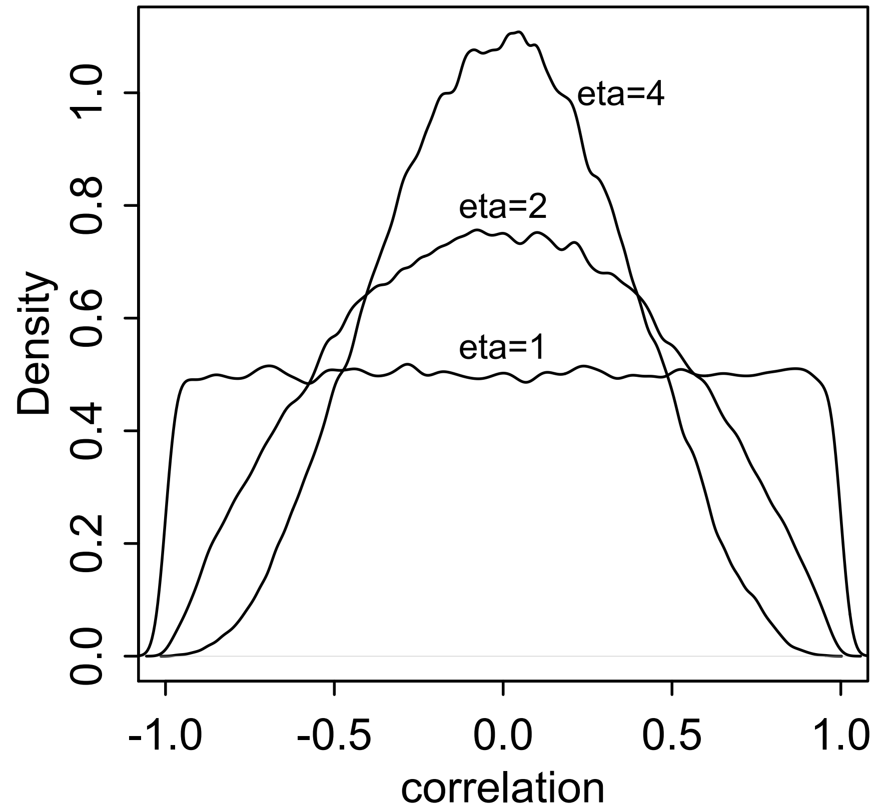
fixed (non-adaptive) priors

$$
\begin{aligned}
W_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\
\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} &\sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \\
\mathbf{S} &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\alpha &\sim \text{Normal}(5, 2) \\
\beta &\sim \text{Normal}(-1, 0.5) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\mathbf{R} &\sim \text{LKJcorr}(2)
\end{aligned}
$$

correction matrix prior

LKJ Correlation prior

- After Lewandowski, Kurowicka, and Joe (LKJ) 2009
- One parameter, *eta*, specifies concentration or dispersion from *identity matrix* (zero correlations)
    - *eta* = 1, uniform correlation matrices
    - *eta* > 1, stomps on extreme correlations
    - *eta* < 1, elevates extreme correlations



Varying slopes estimation

```
m14.1 <- ulam(
    alist(
        wait ~ normal( mu , sigma ),
        mu <- a_cafe[cafe] + b_cafe[cafe]*afternoon,
        c(a_cafe,b_cafe)[cafe] ~ multi_normal( c(a,b) , Rho , sigma_cafe ),
        a ~ normal(5,2),
        b ~ normal(-1,0.5),
        sigma_cafe ~ exponential(1),
        sigma ~ exponential(1),
        Rho ~ lkj_corr(2)
    ) , data=d , chains=4 , cores=4 )
```

Varying slopes estimation

```
m14.1 <- ulam(
    alist(
        wait ~ normal( mu , sigma ),
        mu <- a_cafe[cafe] + b_cafe[cafe]*afternoon,
        c(a_cafe,b_cafe)[cafe] ~ multi_normal( c(a,b) , Rho , sigma_cafe ),
        a ~ normal(5,2),
        b ~ normal(-1,0.5),
        sigma_cafe ~ exponential(1),
        sigma ~ exponential(1),
        Rho ~ lkj_corr(2)
    ) , data=d , chains=4 , cores=4 )
```

Varying slopes estimation

```
m14.1 <- ulam(
    alist(
        wait ~ normal( mu , sigma ),
        mu <- a_cafe[cafe] + b_cafe[cafe]*afternoon,
        c(a_cafe,b_cafe)[cafe] ~ multi_normal( c(a,b) , Rho , sigma_cafe ),
        a ~ normal(5,2),
        b ~ normal(-1,0.5),
        sigma_cafe ~ exponential(1),
        sigma ~ exponential(1),
        Rho ~ lkj_corr(2)
    ) , data=d , chains=4 , cores=4 )
```

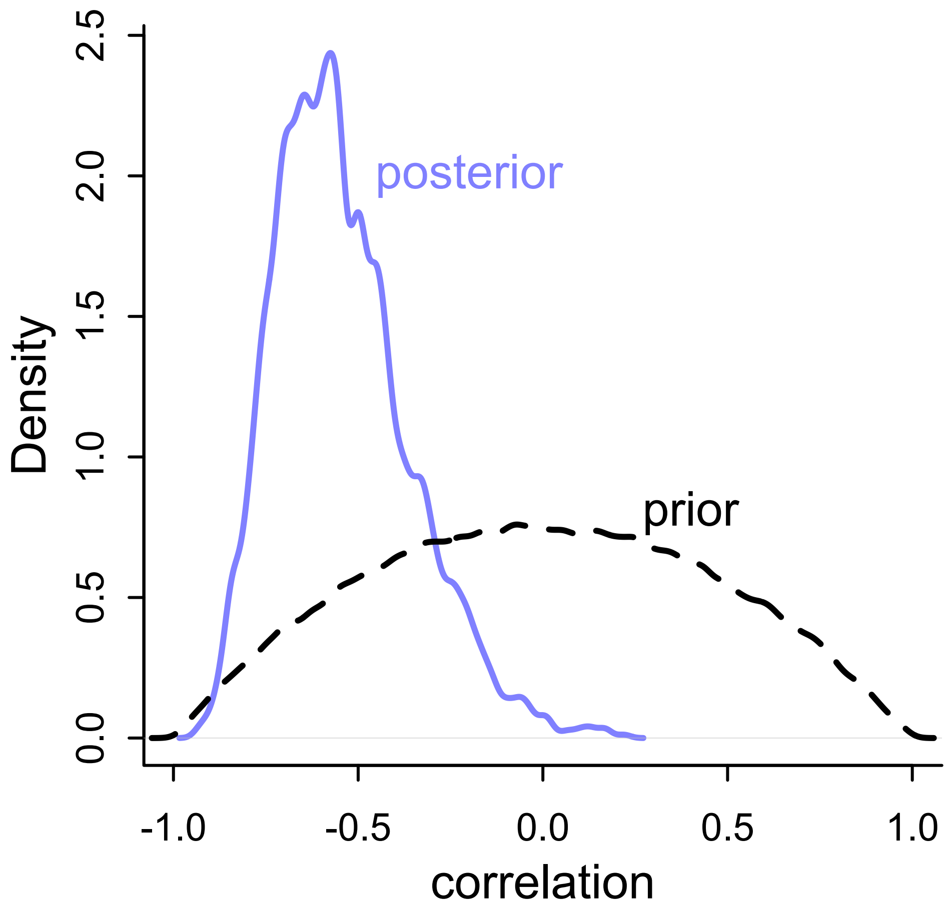
Varying slopes estimation

```
m14.1 <- ulam(
    alist(
        wait ~ normal( mu , sigma ),
        mu <- a_cafe[cafe] + b_cafe[cafe]*afternoon,
        c(a_cafe,b_cafe)[cafe] ~ multi_normal( c(a,b) , Rho , sigma_cafe ),
        a ~ normal(5,2),
        b ~ normal(-1,0.5),
        sigma_cafe ~ exponential(1),
        sigma ~ exponential(1),
        Rho ~ lkj_corr(2)
    ) , data=d , chains=4 , cores=4 )
```

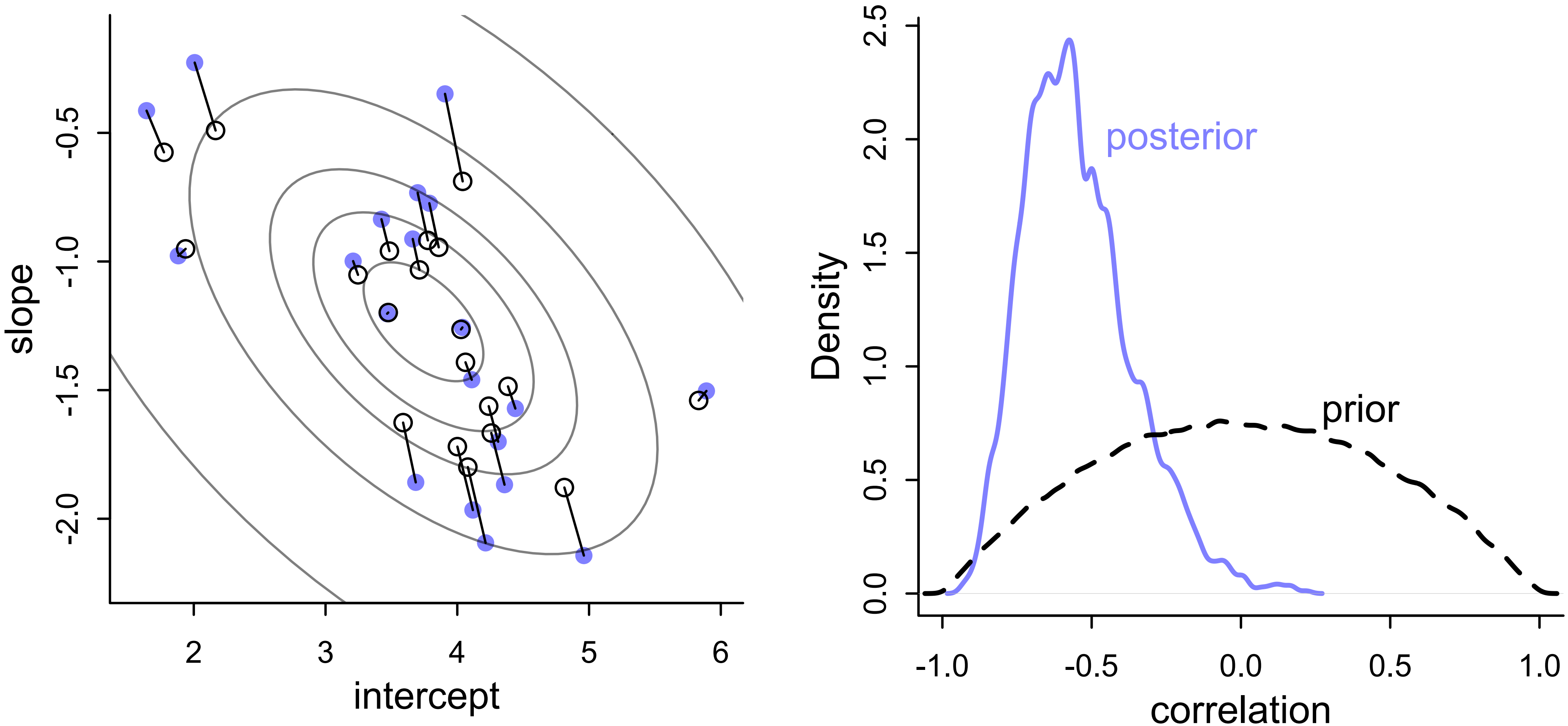
Posterior correlation

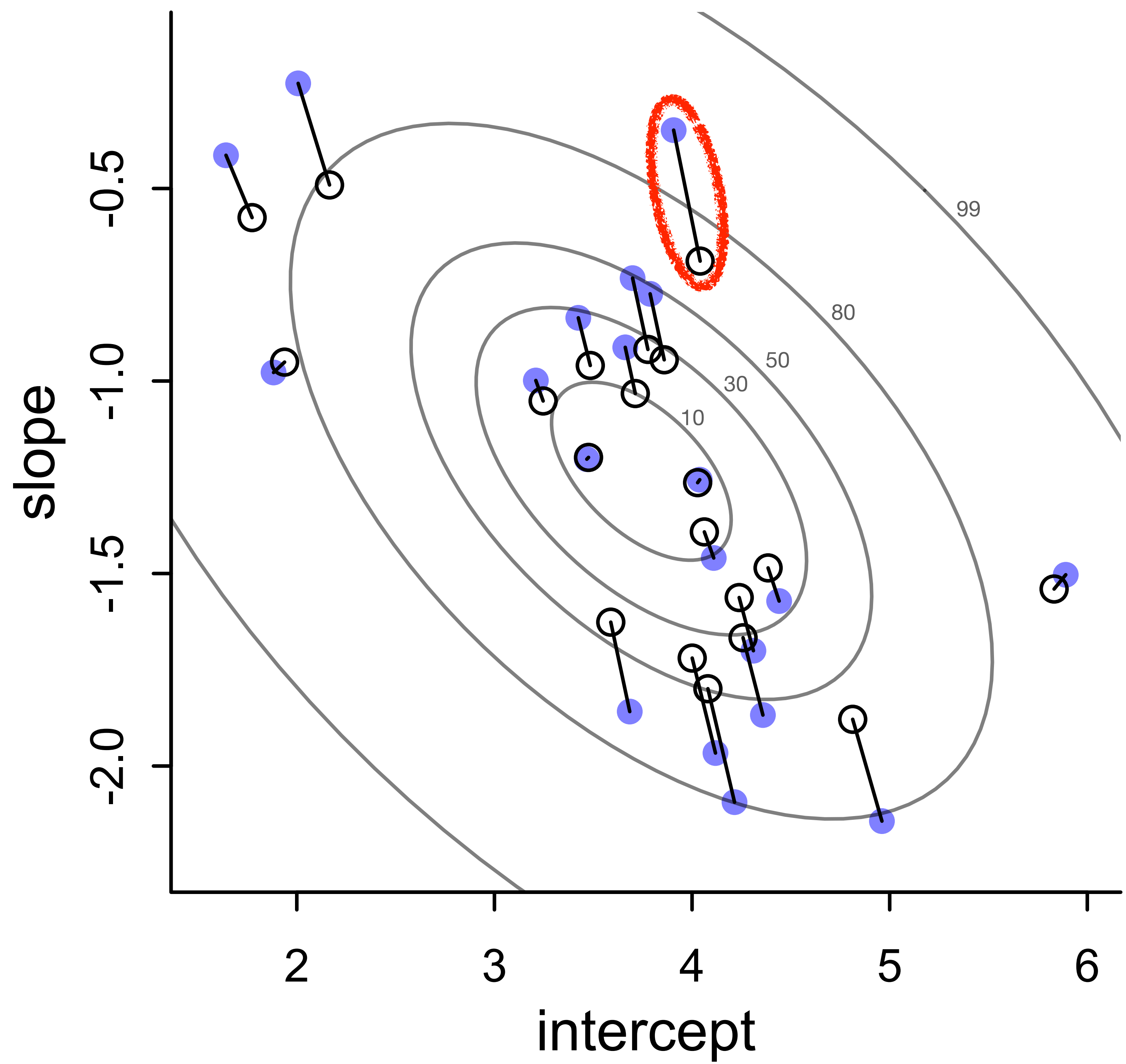
R code
14.13

```
post <- extract.samples(m14.1)
dens( post$Rho[,1,2] )
```

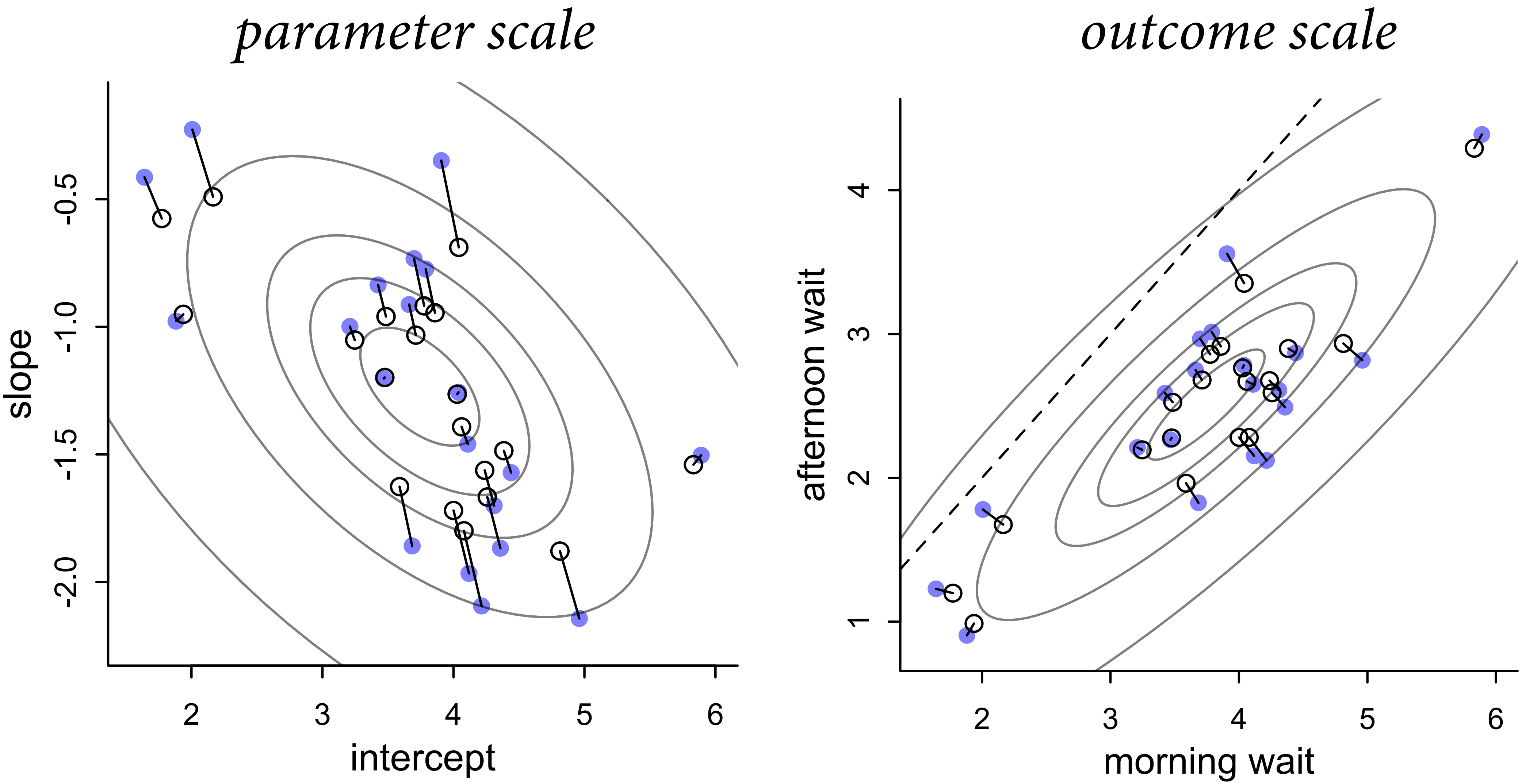


Posterior shrinkage





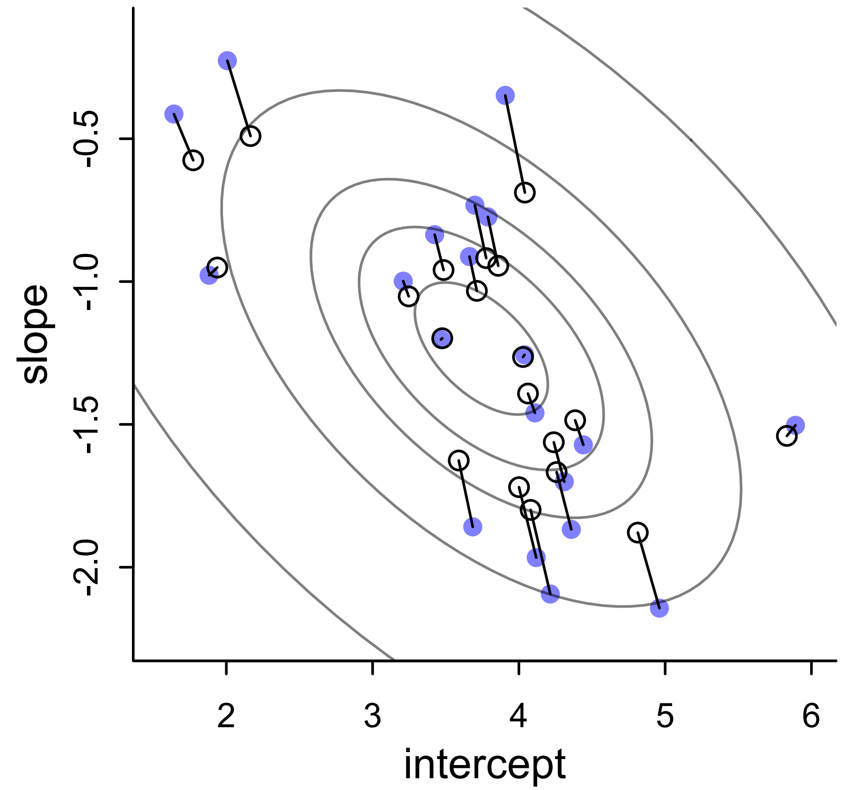
slope
-0.5
-1.0
-1.5
-2.0
intercept
2
3
4
5
6
10
30
50
80
99



parameter scale
outcome scale
slope
intercept
-0.5
-1.0
-1.5
-2.0
2
3
4
5
6
afternoon wait
morning wait
1
2
3
4
2
3
4
5
6

Multi-dimensional shrinkage

- Joint distribution of varying effects pools information across intercepts & slopes
- Correlation btw effects => shrinkage in one dimension induces shrinkage in others
- Improved accuracy, just like varying intercepts



Many effects, many clusters

- Let's try something more ambitious
- Chimpanzees data
  - 4 treatments
  - Each can vary by **actor** or **block**

$$L_i \sim \text{Binomial}(1, p_i)$$

$$\text{logit}(p_i) = \gamma_{\text{TID}[i]} + \alpha_{\text{ACTOR}[i],\text{TID}[i]} + \beta_{\text{BLOCK}[i],\text{TID}[i]}$$

Mean treatment effects

Each actor in each treatment

Each block in each treatment

Covariance matrixes

- One matrix for each cluster (actor, block)
- 7*4 = 28 actor parameters
- 6*4 = 24 block parameters
- Each covariance matrix: 6 correlations + 4 sigmas
- Total: 28 + 24 + 20 + 4 = 76 parameters

$$\begin{bmatrix} \alpha_{j,1} \\ \alpha_{j,2} \\ \alpha_{j,3} \\ \alpha_{j,4} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{S}_{\text{ACTOR}} \right)$$

$$\begin{bmatrix} \beta_{j,1} \\ \beta_{j,2} \\ \beta_{j,3} \\ \beta_{j,4} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{S}_{\text{BLOCK}} \right)$$

1	ρ_{12}	ρ_{13}	ρ_{14}
ρ_{12}	1	ρ_{23}	ρ_{24}
ρ_{13}	ρ_{23}	1	ρ_{34}
ρ_{14}	ρ_{24}	ρ_{34}	1

Maximally random chimps

```
m14.2 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors
        vector[4]:alpha[actor] ~ multi_normal(0,Rho_actor,sigma_actor),
        vector[4]:beta[block_id] ~ multi_normal(0,Rho_block,sigma_block),

        # fixed priors
        g[tid] ~ dnorm(0,1),
        sigma_actor ~ dexp(1),
        Rho_actor ~ dlkjcorr(4),
        sigma_block ~ dexp(1),
        Rho_block ~ dlkjcorr(4)
    ) , data=dat , chains=4 , cores=4 )
```

Maximally random chimps

```
m14.2 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors
        vector[4]:alpha[actor] ~ multi_normal(0,Rho_actor,sigma_actor),
        vector[4]:beta[block_id] ~ multi_normal(0,Rho_block,sigma_block),

        # fixed priors
        g[tid] ~ dnorm(0,1),
        sigma_actor ~ dexp(1),
        Rho_actor ~ dlkjcorr(4),
        sigma_block ~ dexp(1),
        Rho_block ~ dlkjcorr(4)
    ) , data=dat , chains=4 , cores=4 )
```

Maximally random chimps

```
m14.2 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors
        vector[4]:alpha[actor] ~ multi_normal(0,Rho_actor,sigma_actor),
        vector[4]:beta[block_id] ~ multi_normal(0,Rho_block,sigma_block),

        # fixed priors
        g[tid] ~ dnorm(0,1),
        sigma_actor ~ dexp(1),
        Rho_actor ~ dlkjcorr(4),
        sigma_block ~ dexp(1),
        Rho_block ~ dlkjcorr(4)
    ) , data=dat , chains=4 , cores=4 )
```

Divergence, my old friend

!rethinking package — R — 80×21

```
Chain 4:                29.2344 seconds (Total)
Chain 4: 
Chain 2: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 2: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 2: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 2: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 2: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 2: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 2: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 2: 
Chain 2:  Elapsed Time: 30.4394 seconds (Warm-up)
Chain 2:                4.42052 seconds (Sampling)
Chain 2:                34.86 seconds (Total)
Chain 2: 
Warning messages:
1: There were 96 divergent transitions after warmup. Increasing adapt_delta abov
e 0.95 may help. See
http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
2: Examine the pairs() plot to diagnose sampling problems

> 
```

Non-centered form

- Non-centered easy for uni-variate priors: Just factor out sigma
- But now need to factor correlation matrix out of the prior and smuggle into linear model
- Can be done: Cholesky factor

André-Louis Cholesky (1875–1918)

I. — NOTICES SCIENTIFIQUES

Commandant BENOIT[1].

NOTE SUR UNE MÉTHODE DE RÉSOLUTION DES ÉQUATIONS NORMALES PROVENANT DE L'APPLICATION DE LA MÉTHODE DES MOINDRES CARRÉS A UN SYSTÈME D'ÉQUATIONS LI CELUI DES INCO THODE A LA R D'ÉQUATIONS LI

(Procédé

Le Commandant d'Artillerie Cholesky, du Service géographique de l'Armée, tué pendant la grande guerre, a imaginé, au cours de recherches sur la compensation des réseaux géodésiques, un procédé très ingénieux de résolution des équations dites *normales*, obtenues par application de la méthode des moindres carrés à des équations linéaires en nombre inférieur à celui des inconnues. Il en a conclu une méthode générale de résolution des équations linéaires.

Nous suivrons, pour la démonstration de cette méthode, la progression même qui a servi au Commandant Cholesky pour l'imaginer.

1. De l'Artillerie coloniale, ancien officier géodésien au Service géographique de l'Armée et au Service géographique de l'Indo-Chine, Membre du Comité national français de Géodésie et Géophysique.

2. Sur le Commandant Cholesky, tué à l'ennemi le 31 août 1918, voir la notice biographique insérée dans le volume du *Bulletin géodésique* de 1922 intitulé : *Union géodésique et géophysique internationale, Première Assemblée générale, Rome, mai 1922, Section de Géodésie*, Toulouse, Privat, 1922, in-8°, 241 p., pp. 159 à 161.

Cholesky magic

```
N <- 1e4
sigma1 <- 2
sigma2 <- 0.5
rho <- 0.6
z1 <- rnorm( N )
z2 <- rnorm( N )
a1 <- z1 * sigma1
a2 <- ( rho*z1 + sqrt( 1-rho^2 )*z2 )*sigma2
```

```
> cor(z1,z2)
[1] -0.0005542644
> cor(a1,a2)
[1] 0.5999334
> sd(a1)
[1] 1.997036
> sd(a2)
[1] 0.4989456
```

Non-centered random chimps

```
m14.3 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors - non-centered
        transpars> matrix[actor,4]:alpha <-
                compose_noncentered( sigma_actor , L_Rho_actor , z_actor ),
        transpars> matrix[block_id,4]:beta <-
                compose_noncentered( sigma_block , L_Rho_block , z_block ),
        matrix[4,actor]:z_actor ~ normal( 0 , 1 ),
        matrix[4,block_id]:z_block ~ normal( 0 , 1 ),

        # fixed priors
        g[tid] ~ normal(0,1),
        vector[4]:sigma_actor ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_actor ~ lkj_corr_cholesky( 2 ),
        vector[4]:sigma_block ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_block ~ lkj_corr_cholesky( 2 )
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

Non-centered random chimps

```
m14.3 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors - non-centered
        transpars> matrix[actor,4]:alpha <-
                compose_noncentered( sigma_actor , L_Rho_actor , z_actor ),
        transpars> matrix[block_id,4]:beta <-
                compose_noncentered( sigma_block , L_Rho_block , z_block ),
        matrix[4,actor]:z_actor ~ normal( 0 , 1 ),
        matrix[4,block_id]:z_block ~ normal( 0 , 1 ),

        # fixed priors
        g[tid] ~ normal(0,1),
        vector[4]:sigma_actor ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_actor ~ lkj_corr_cholesky( 2 ),
        vector[4]:sigma_block ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_block ~ lkj_corr_cholesky( 2 )
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

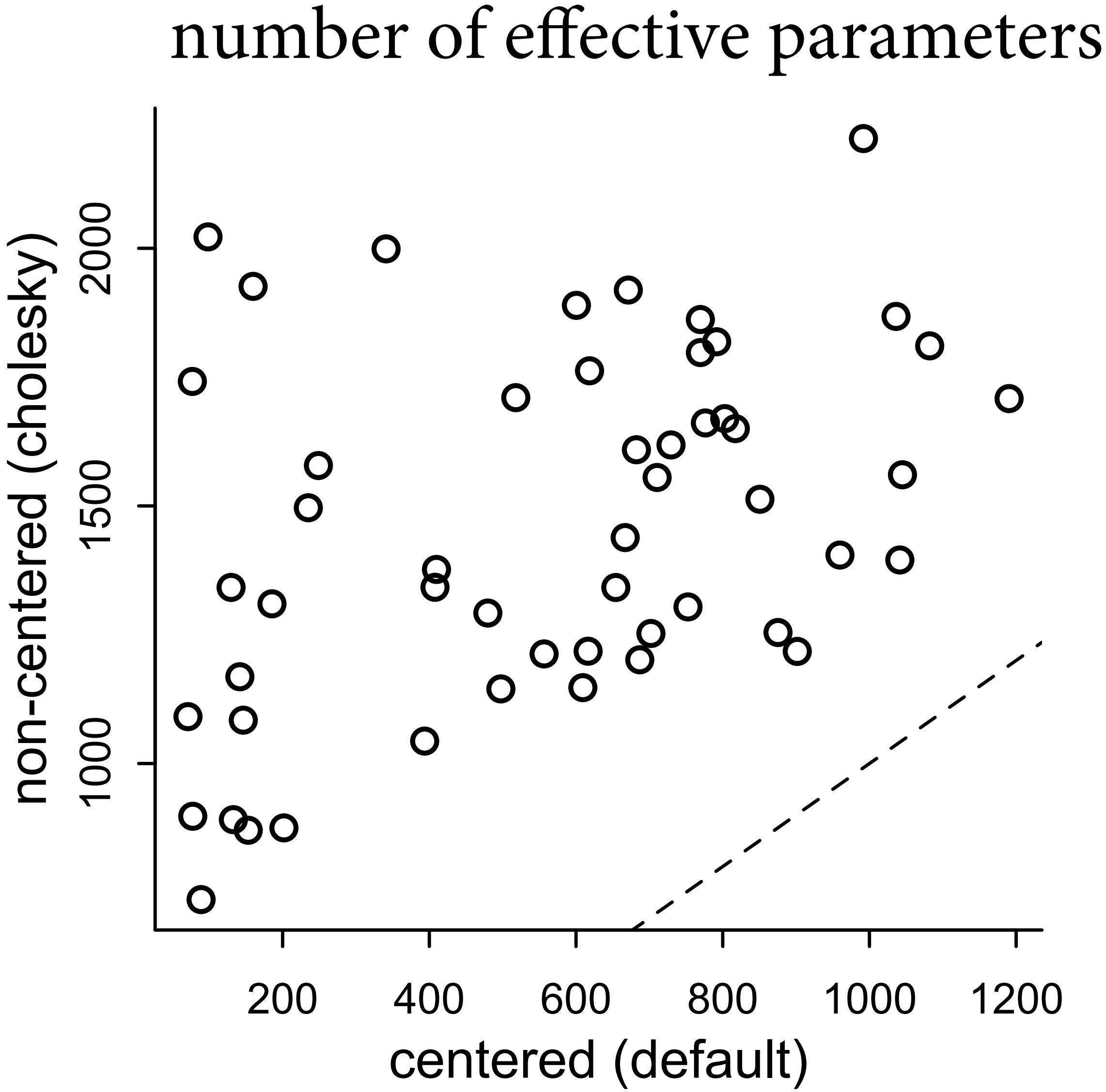
Non-centered random chimps

```
m14.3 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors - non-centered
        transpars> matrix[actor,4]:alpha <-
                compose_noncentered( sigma_actor , L_Rho_actor , z_actor ),
        transpars> matrix[block_id,4]:beta <-
                compose_noncentered( sigma_block , L_Rho_block , z_block ),
        matrix[4,actor]:z_actor ~ normal( 0 , 1 ),
        matrix[4,block_id]:z_block ~ normal( 0 , 1 ),

        # fixed priors
        g[tid] ~ normal(0,1),
        vector[4]:sigma_actor ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_actor ~ lkj_corr_cholesky( 2 ),
        vector[4]:sigma_block ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_block ~ lkj_corr_cholesky( 2 )
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

Non-centered random chimps



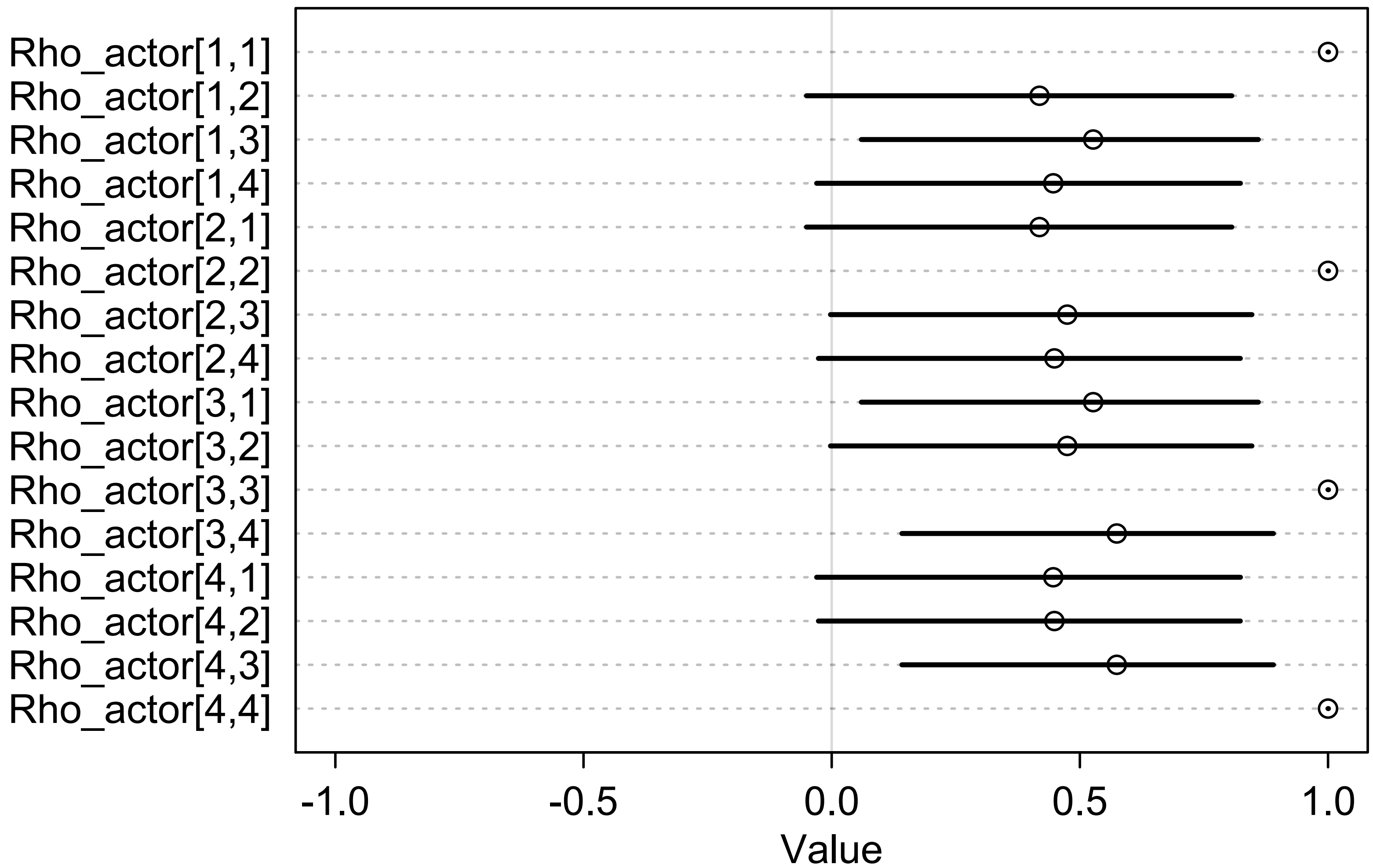
Random chimpanzees

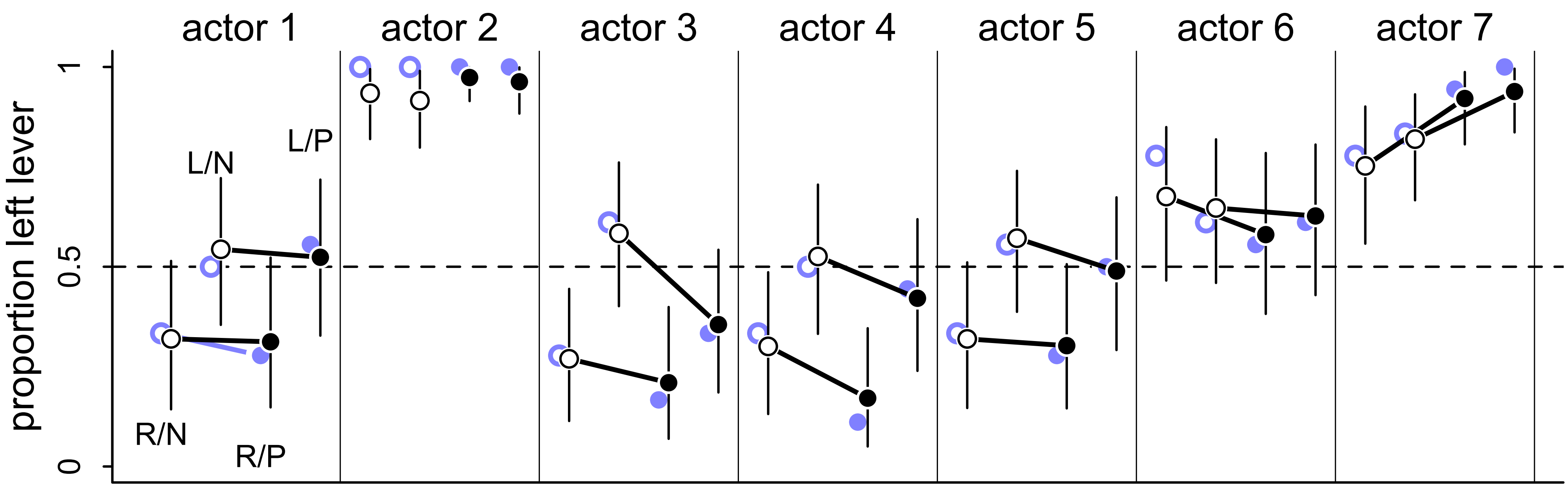
```
precis( m14.3 , depth=2 , pars=c("sigma_actor","sigma_block") )
```

R code 14.21

```
                mean   sd 5.5% 94.5% n_eff Rhat
sigma_actor[1] 1.39 0.49 0.80  2.24   906    1
sigma_actor[2] 0.92 0.38 0.44  1.64  1060    1
sigma_actor[3] 1.86 0.57 1.14  2.89  1191    1
sigma_actor[4] 1.59 0.66 0.86  2.81  1148    1
sigma_block[1] 0.40 0.32 0.03  0.98  1033    1
sigma_block[2] 0.44 0.36 0.04  1.10   944    1
sigma_block[3] 0.30 0.27 0.02  0.79  1606    1
sigma_block[4] 0.47 0.38 0.03  1.15  1073    1
```

Correlations






```
                  mean   sd 5.5% 94.5% n_eff Rhat
sigma_actor[1] 1.39 0.49 0.80  2.24   906    1
sigma_actor[2] 0.92 0.38 0.44  1.64  1060    1
sigma_actor[3] 1.86 0.57 1.14  2.89  1191    1
sigma_actor[4] 1.59 0.66 0.86  2.81  1148    1
```