

Chapter 5

Fitting the distribution

I showed this slide, not expecting you to understand but rather to give an idea what going on underneath (the math).

- Observing a height h_1 , the posterior distribution is

$$\begin{aligned} p(\mu, \sigma | h_1) &= \frac{N(h_1 | \mu, \sigma) p(\mu) p(\sigma)}{\int \int N(h_1 | \tilde{\mu}, \tilde{\sigma}) p(\tilde{\mu}) p(\tilde{\sigma}) d\tilde{\mu} d\tilde{\sigma}} \\ &\propto N(h_1 | \mu, \sigma) p(\mu) p(\sigma) \end{aligned}$$

- Observing two heights h_1, h_2 , the posterior distribution is

$$\begin{aligned} p(\mu, \sigma | h_1, h_2) &= \frac{N(h_1 | \mu, \sigma) N(h_2 | \mu, \sigma) p(\mu) p(\sigma)}{\int \int N(h_1 | \tilde{\mu}, \tilde{\sigma}) N(h_2 | \tilde{\mu}, \tilde{\sigma}) p(\tilde{\mu}) p(\tilde{\sigma}) d\tilde{\mu} d\tilde{\sigma}} \\ &\propto N(h_1 | \mu, \sigma) N(h_2 | \mu, \sigma) p(\mu) p(\sigma) \end{aligned}$$

Repetition

Started exploring linear Gaussian (Normal) models:

$$y_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + x_i \beta$$

$$\beta \sim N(\mu_\beta, \sigma_\beta)$$

$$\alpha \sim N(\mu_\alpha, \sigma_\alpha)$$

$$\sigma \sim U[a_\sigma, b_\sigma]$$

Fake data

Simulating fake data:

```
alpha_true <- 0
beta_true <- 1
sigma_true <- 0.1
x      <- 4 * runif(n = n)
y      <- alpha_true + beta_true * x + sigma_true * rnorm(n = n)
data = list(x = x, y = y)
```

Fake data

estimating the data:

```
model <- map( alist(
y ~ dnorm(mu, sigma),
mu <- a + b * x,
a ~ dnorm(0,100),
b ~ dnorm(0, 100),
sigma ~ dunif(0.,50)),
data = data,
start = list(a=0,b=0,sigma=1))
samples <- extract.samples(model)
```

Contraction of the posterior distribution

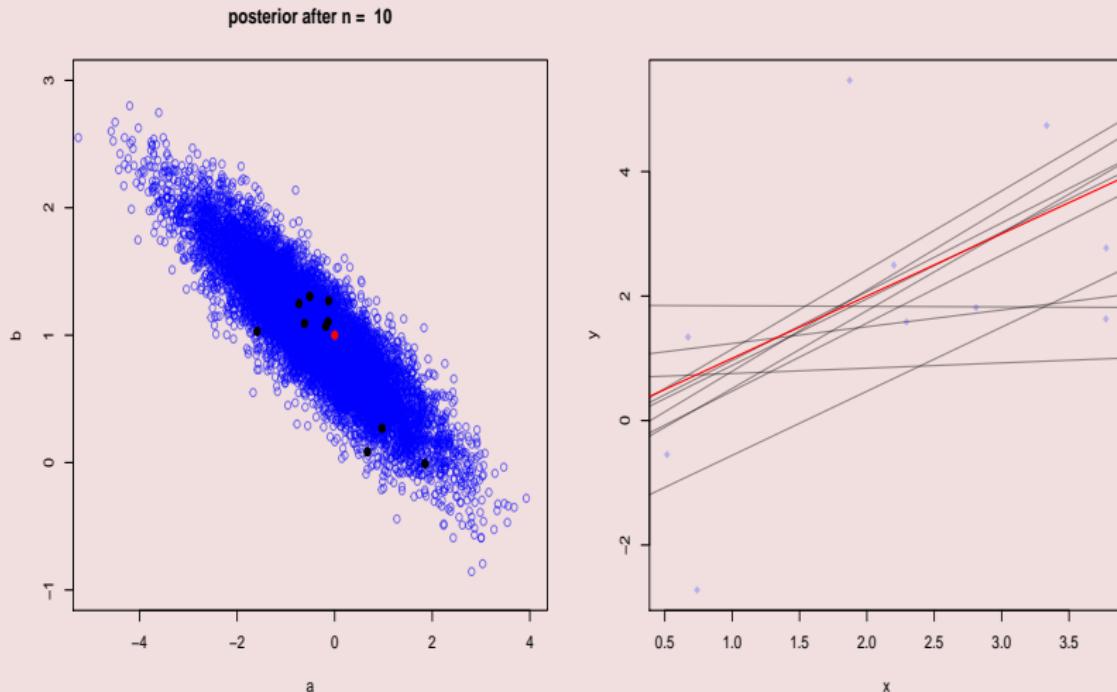


Figure : samples from $p(a, b|x_1, \dots, x_n, y_1, \dots, y_n)$

Contraction of the posterior distribution

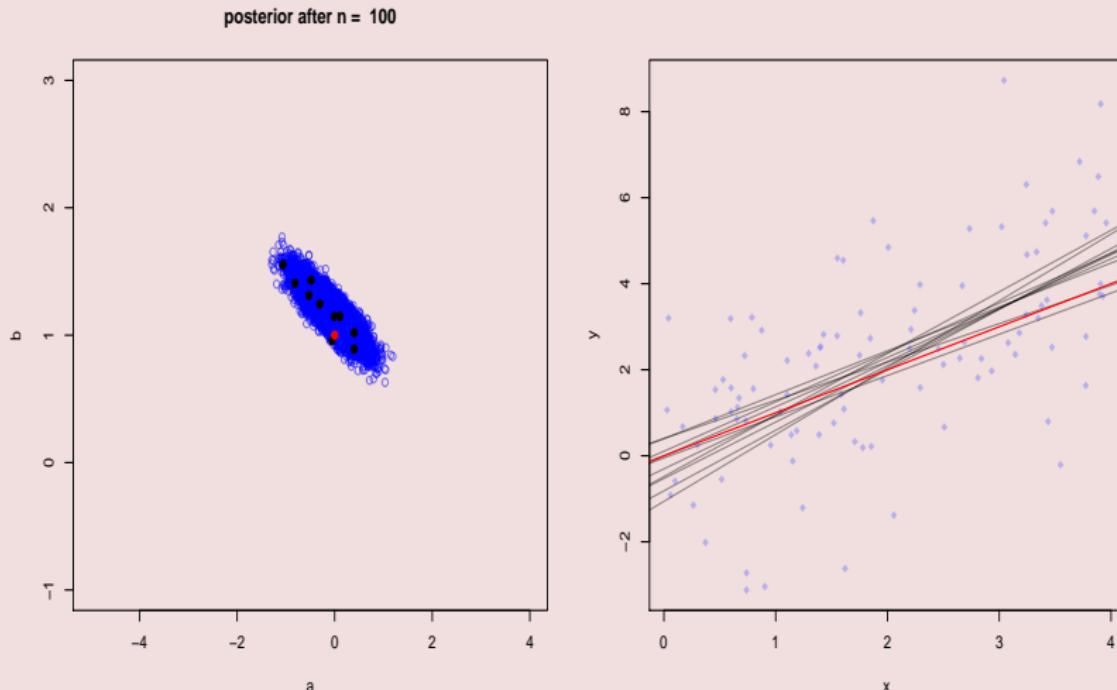


Figure : samples from $p(a, b|x_1, \dots, x_n, y_1, \dots, y_n)$

Contraction of the posterior distribution

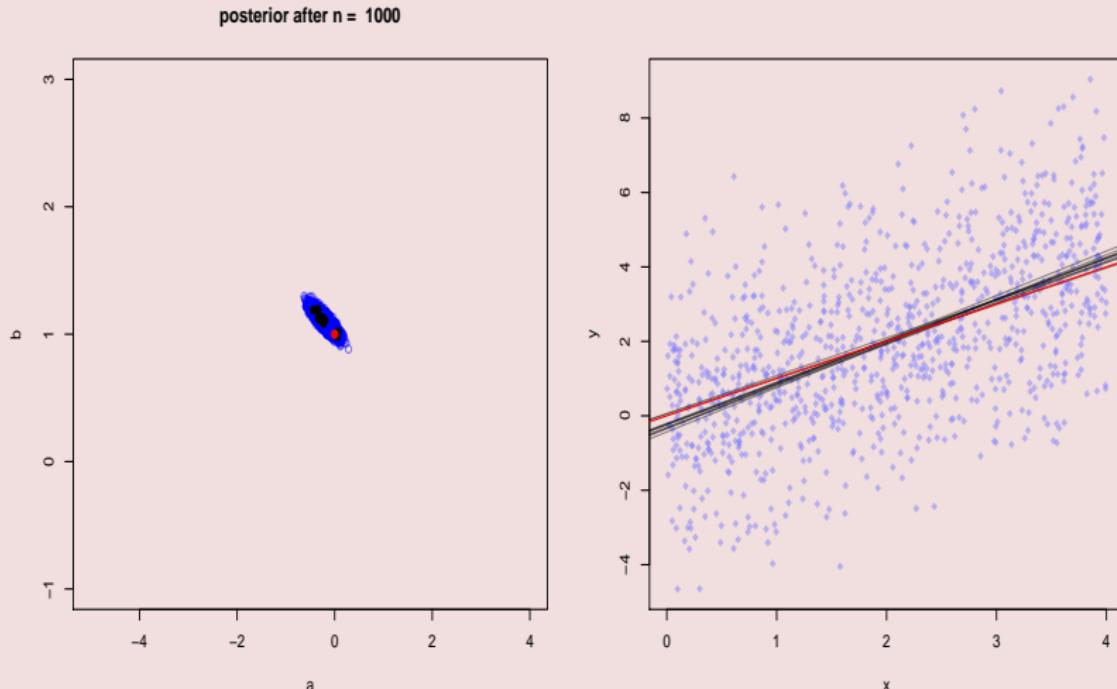


Figure : samples from $p(a, b|x_1, \dots, x_n, y_1, \dots, y_n)$

Contraction of the posterior distribution

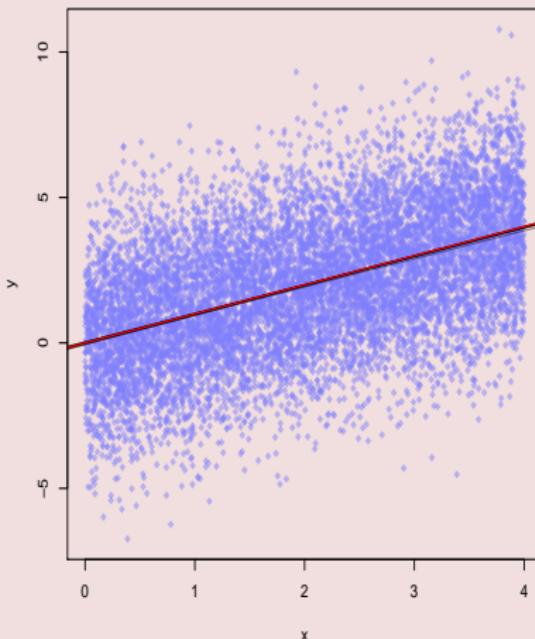
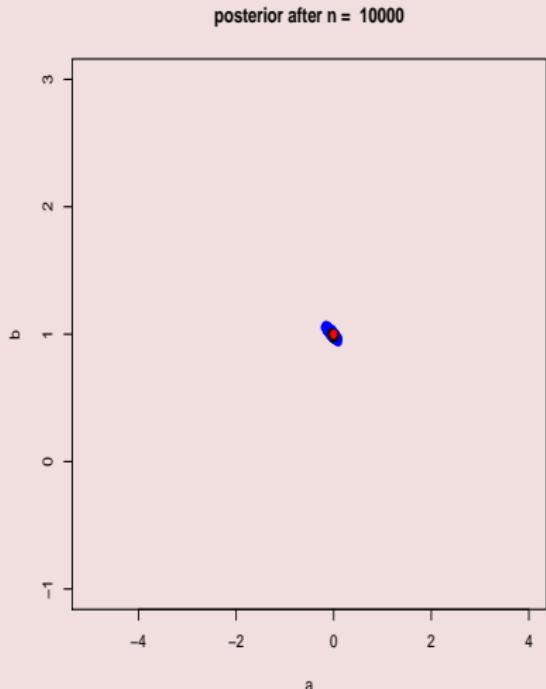


Figure : samples from $p(a, b|x_1, \dots, x_n, y_1, \dots, y_n)$

Quantiles from samples

Generate percentile interval:

```
PI95_b <- quantiles(samples$b, c(0.025,0.975))
```

N = 3		PI95_a = (-0.146 , 3.4262)
N = 10		PI95_a = (0.0612 , 2.0160)
N = 100		PI95_a = (0.8930 , 1.4837)
N = 1000		PI95_a = (0.9988 , 1.1973)
N = 10000		PI95_a = (0.9714 , 1.0391)

Generating samples

Generate percentile interval:

```
precis(model)
```

	Mean	StdDev	5.5%	94.5%
a	-0.05	0.04	-0.12	0.01
b	1.02	0.02	0.99	1.05
sigma	2.02	0.01	2.00	2.04

Example data

Divorce rate in the US Model:

$$d_i \sim N(\mu_i, \sigma), \\ \mu_i = \alpha + r_i \beta_r + a_i \beta_a$$

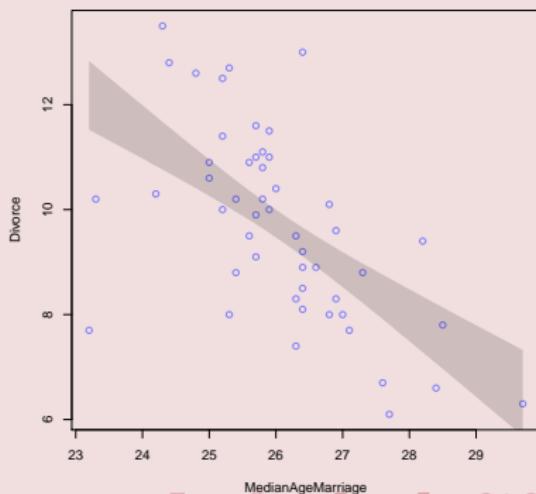
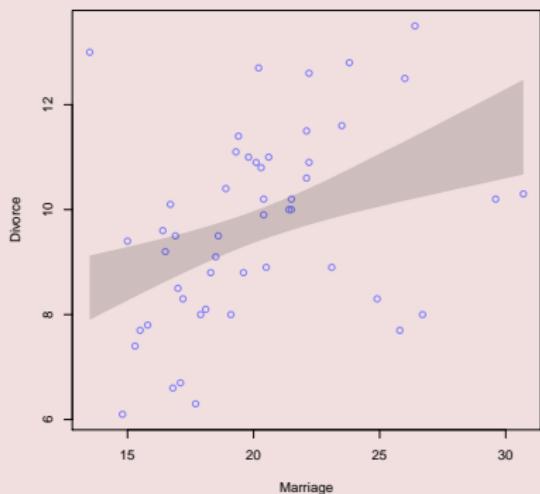
- d_i - divorce rate by state
- r_i - marriage rate by state
- a_i - median age at marriage

Priors will be chosen non-informative.

Fitting the models separately

$$d_i \sim N(\mu_i, \sigma), \\ \mu_i = \alpha + r_i \beta_r$$

$$d_i \sim N(\mu_i, \sigma), \\ \mu_i = \alpha + a_i \beta_a$$



fitting model

$$d_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + r_i \beta_r + a_i \beta_a$$

$$\beta_r \sim N(0, 100)$$

$$\beta_a \sim N(0, 100)$$

$$\alpha \sim N(0, 100)$$

$$\sigma \sim U(0, 50)$$

```
library(rethinking)
model <- map(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <- alpha + MedianAgeMarriage * beta_a
      + Marriage * beta_r,
    alpha ~ dnorm(0, 100),
    beta_a ~ dnorm(0, 100),
    beta_r ~ dnorm(0, 100),
    sigma ~ dunif(0, 50)),
  start = list(alpha = 0, beta_a = 0, beta_r = 0,
    sigma = 1),
  data = WaffleDivorce)
```

Table output

```
precis(model)
```

	Mean	StdDev	5.5%	94.5%
alpha	36.69	7.41	24.85	48.53
beta_a	-0.99	0.24	-1.37	-0.61
beta_r	-0.06	0.08	-0.18	0.07
sigma	1.44	0.14	1.21	1.67

Figure table output

```
plot(precis(model))
plot(precis(model), pars = c('beta_r', 'beta_a'))
```

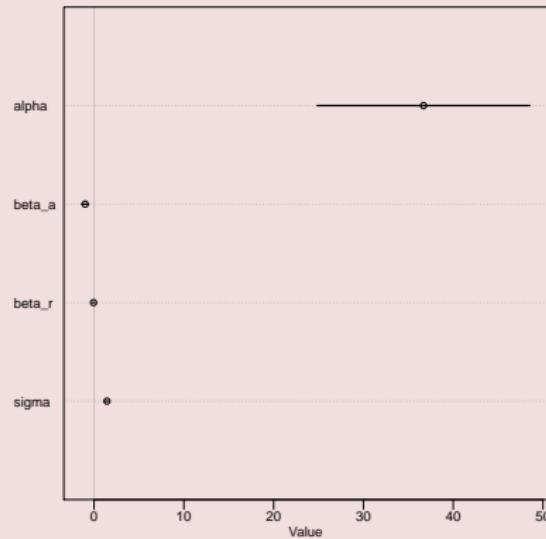
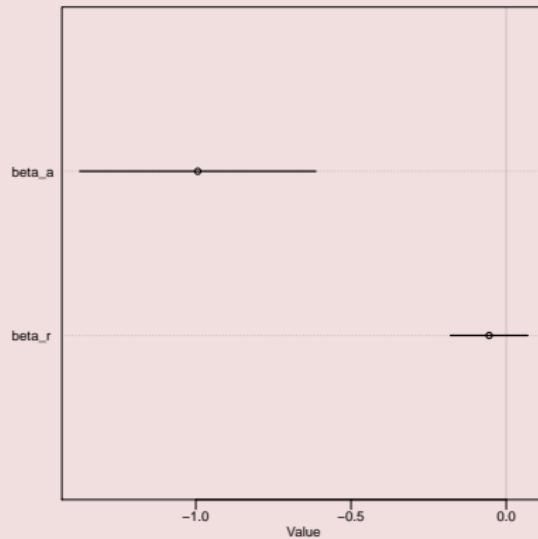


Figure table output

```
plot(precis(model))
plot(precis(model), pars = c('beta_r', 'beta_a'))
```



Counterfactual plots

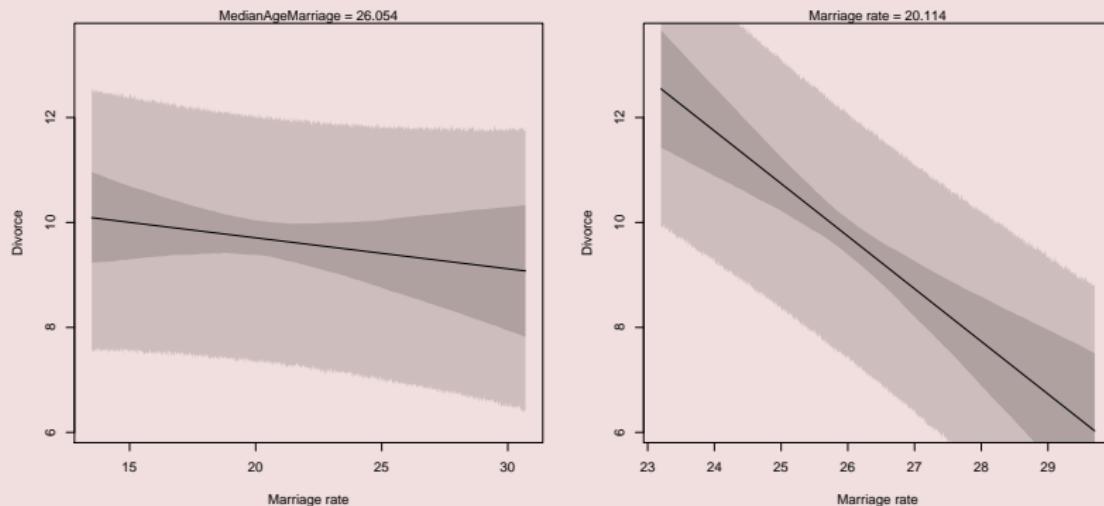


Figure : Keeping one variable fixed varying the other

Posterior prediction, residuals

- Check model fit.
- Find model failures, use to improve model.
- Non Bayesian $y_i - \hat{\mu}_i$
- Bayesian $pr(y_i - \mu_i | y_1, \dots, y_n)$ for $i = 1, \dots, n$.

residual

simple model

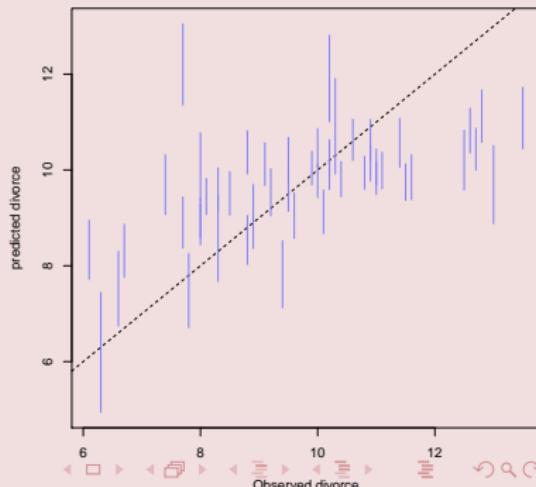
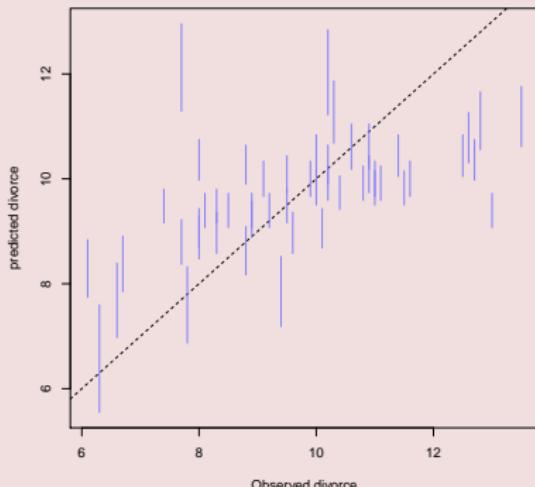
$$d_i \sim N(\mu_i, \sigma),$$

$$\mu_i = \alpha + a_i \beta_a$$

full model

$$d_i \sim N(\mu_i, \sigma),$$

$$\mu_i = \alpha + r_i \beta_r + a_i \beta_a$$



milk and brains

Model:

$$k_i \sim N(\mu_i, \sigma),$$

$$\mu_i = \alpha + n_i \beta_n + \log(m_i) \beta_m$$

- k_i - calories per gram milk.
- m_i - weight of the ape.
- n_i - neocortex percentage.

Priors see book.

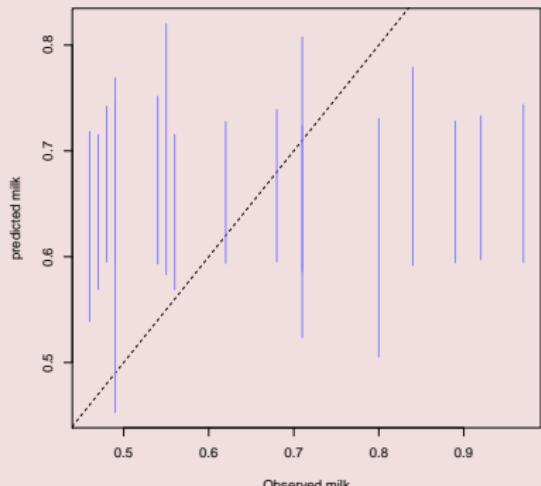


0.89 kcal/g
68% neocortex

residual

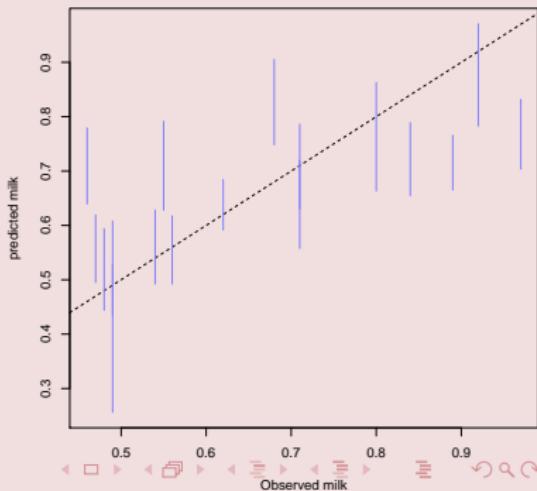
simple model

$$k_i \sim N(\mu_i, \sigma), \\ \mu_i = \alpha + n_i \beta_n$$

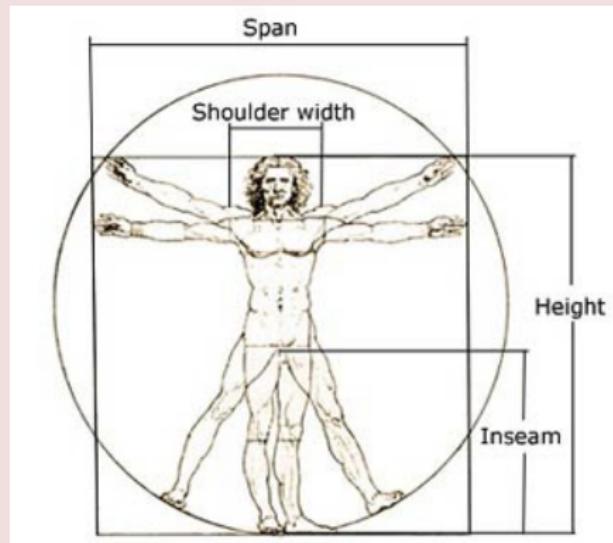


full model

$$d_i \sim N(\mu_i, \sigma), \\ \mu_i = \alpha + n_i \beta_n + \log(m_i) \beta_m$$



Body proportion



Simulating data

Simulating a data set.

```
N      <- 100 # number of individuals
height <- rnorm(N, 10, 2)
leg_prop <- runif(0,0.4, 0.5)
leg_left <- leg_prop * height + rnorm(0, 0, 0.02)
leg_right <- leg_prop * height + rnorm(0, 0, 0.02)
d       <- data.frame( height, leg_left, leg_right)
```

	height	leg_left	leg_right
1	8.747092	3.741833	3.735553
2	10.367287	4.373219	4.381734
3	8.328743	3.755563	3.760531
4	13.190562	5.612398	5.626032
5	10.659016	4.426965	4.470625
6	8.359063	3.755603	3.800031

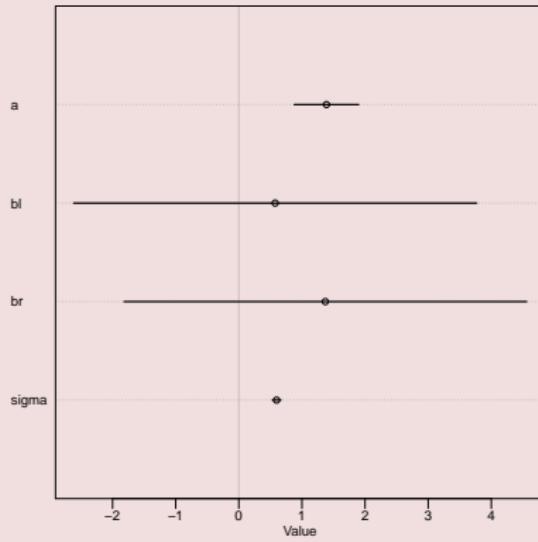
the leg model

Building and estimating a Bayesian model:

```
model <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- alpha + leg_left * beta1 + leg_right * beta2,
    alpha ~ dnorm(10, 100),
    beta1 ~ dnorm(2, 10),
    beta2 ~ dnorm(2, 10),
    sigma ~ duniform(0, 10)
  ),
  data = d)
precis(model)
```

	Mean	StdDev	5.5%	94.5%
alpha	1.60	0.34	1.06	2.13
beta1	2.31	2.34	-1.43	6.05
beta2	-0.44	2.35	-4.19	3.32
sigma	0.65	0.05	0.58	0.73

Fitting the distribution

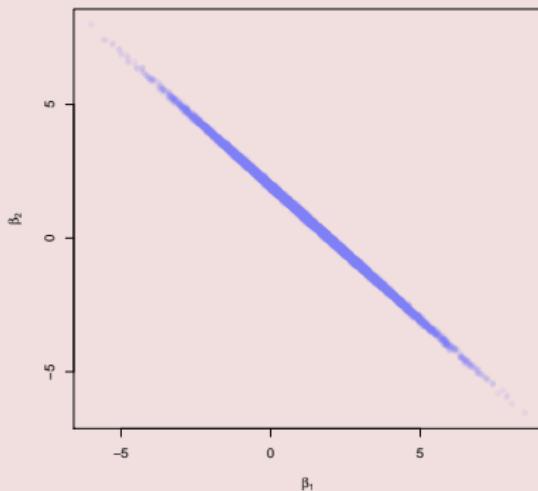


Fitting the distribution

- Note that
 $\text{left_leg} \approx \text{right_leg}$.

Fitting the distribution

- Note that $\text{left_leg} \approx \text{right_leg}$.
- Thus the β_1 and β_2 are very dependent.

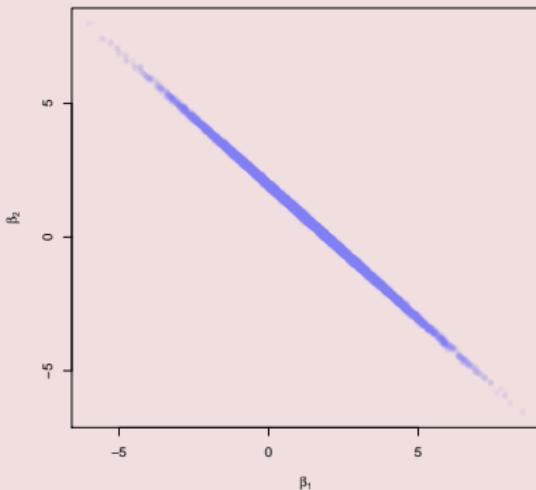


Fitting the distribution

- Note that $\text{left_leg} \approx \text{right_leg}$.
- Thus the β_1 and β_2 are very dependent.
- The model is almost:

$$\text{height}_i \sim N(\mu_i, \sigma),$$

$$\mu_i = \alpha + \text{leg}_i(\beta_1 + \beta_2)$$



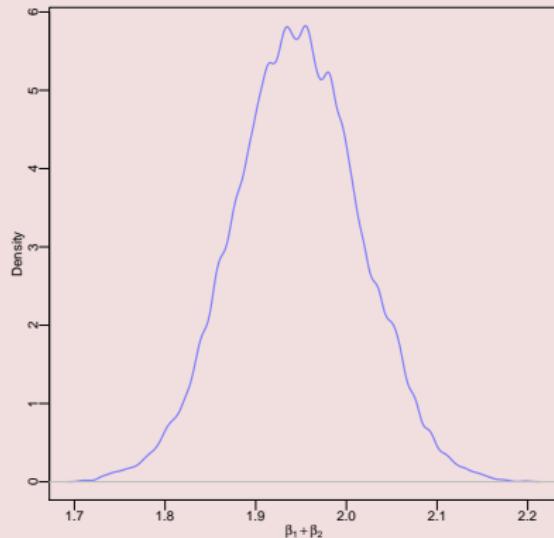
Fitting the distribution

- Note that $\text{left_leg} \approx \text{right_leg}$.
- Thus the β_1 and β_2 are very dependent.
- The model is almost:

$$\text{height}_i \sim N(\mu_i, \sigma),$$

$$\mu_i = \alpha + \text{leg}_i(\beta_1 + \beta_2)$$

- Examining posterior distribution of $\beta_1 + \beta_2$



Category model

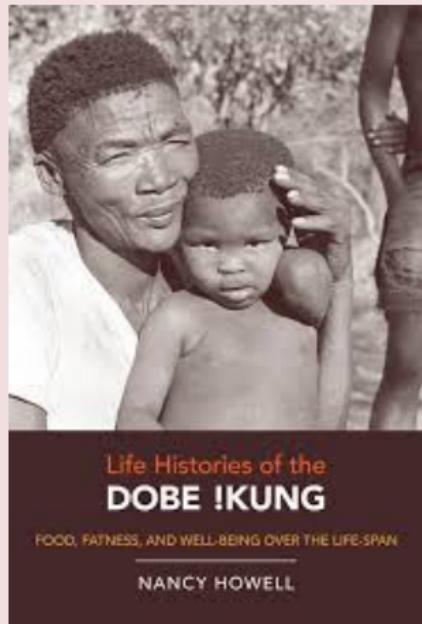
- Binary variables (or dummy variables) allow each category to have its own unique intercept.

$$h_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + m\beta_m$$

- For simple model, better interpretation we think the categories have the same standard deviation.

height data again



```
library(rethinking)
data(Howell1)
head(Howell1)
```

	height	male
1	151.765	1
2	139.700	0
3	136.525	0
4	156.845	1
5	145.415	0
6	163.830	1

difference in gender

```
precis(model)
```

	Mean	StdDev	5.5%	94.5%
alpha	149.52	0.40	148.87	150.16
beta	10.84	0.59	9.90	11.79
sigma	5.52	0.21	5.19	5.85

Table output

```
post <- extract.samples(model)
alpha.male <- post$alpha + post$beta
PI(alpha.male)
```

5% 94%
159.6768 161.0520

The two models

difference prior

$$h_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta m_i$$

$$\alpha \sim N(178, 100)$$

$$\beta \sim N(0, 10)$$

$$\sigma \sim U(0, 50)$$

independent prior

$$h_i \sim N(\mu_i, \sigma)$$

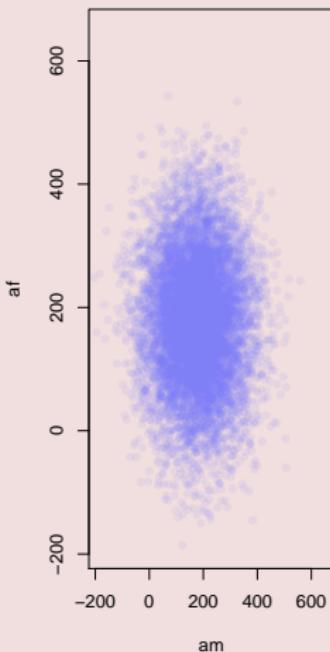
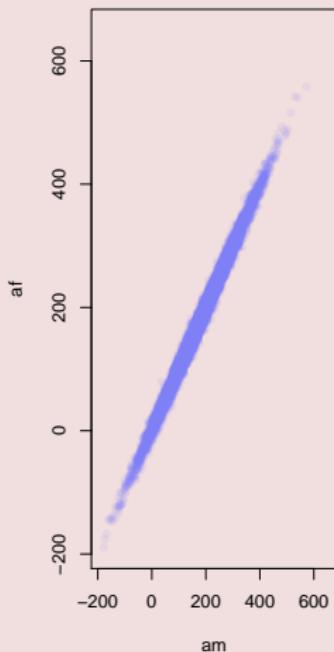
$$\mu_i = \alpha_f(1 - m_i) + \alpha_m m_i$$

$$\alpha_m \sim N(178, 100)$$

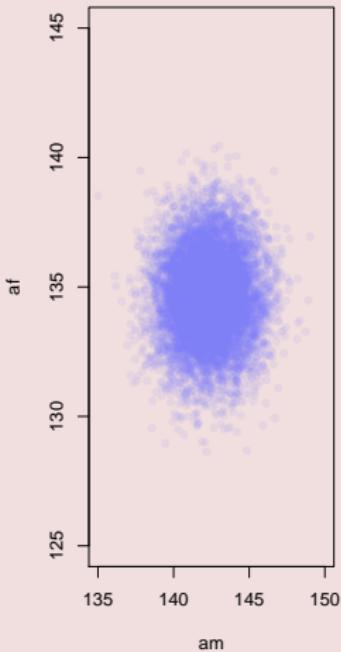
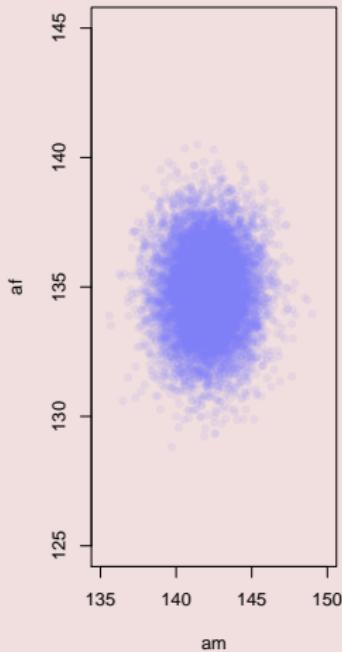
$$\alpha_f \sim N(178, 100)$$

$$\sigma \sim U(0, 50)$$

Sampling from prior



Sampling from posterior



The two priors

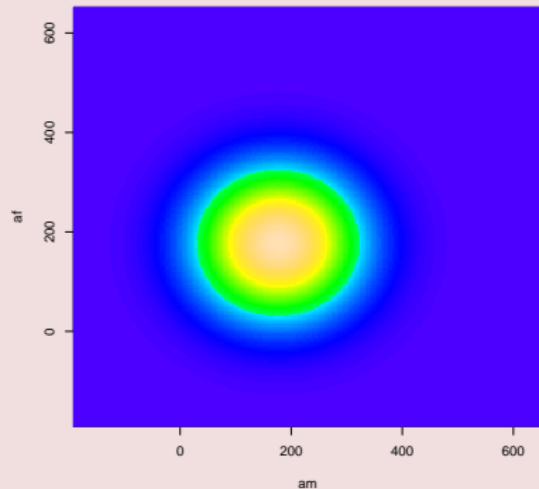
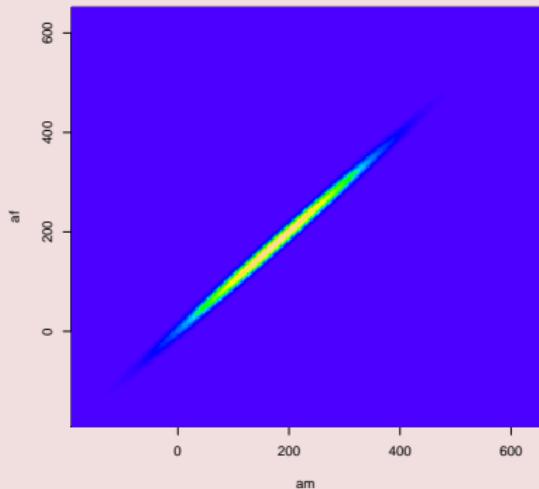


Figure : The two priors on α_f and α_m

The likelihood

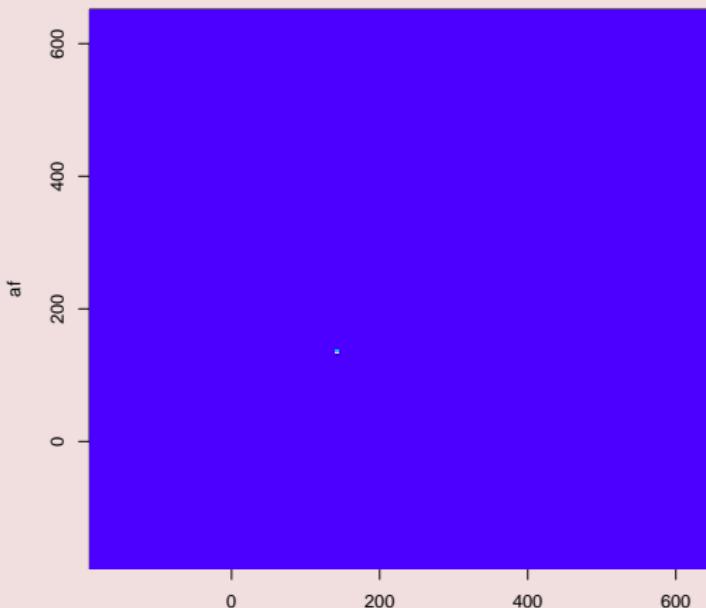


Figure : $p(h_1, \dots, h_n | \alpha_m, \alpha_f, \sigma = \hat{\sigma})$

Two posteriors

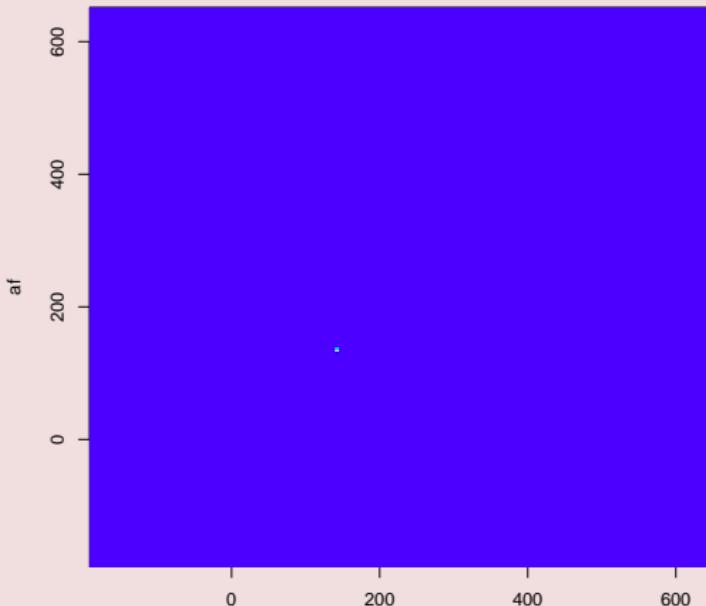


Figure : $p(h_1, \dots, h_n | \alpha_m, \alpha_f, \sigma = \hat{\sigma})$

The two posteriors

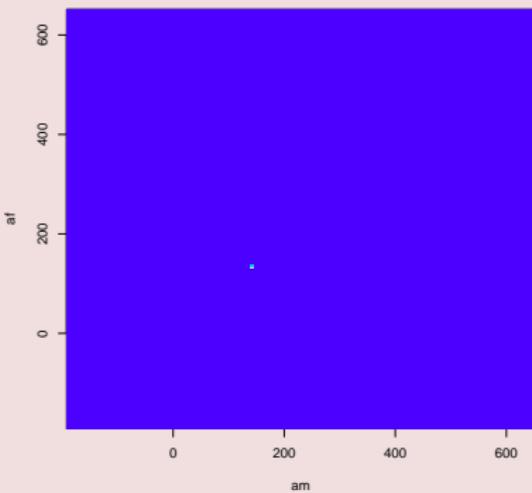
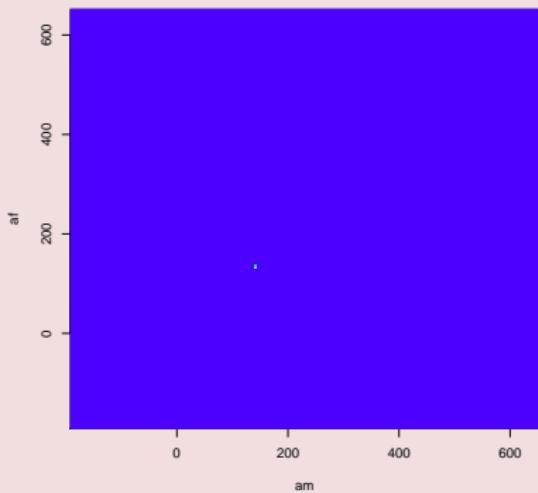


Figure : The two posteriors

The two posteriors

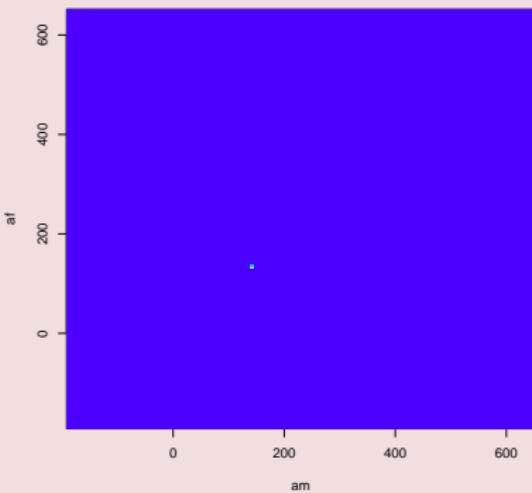
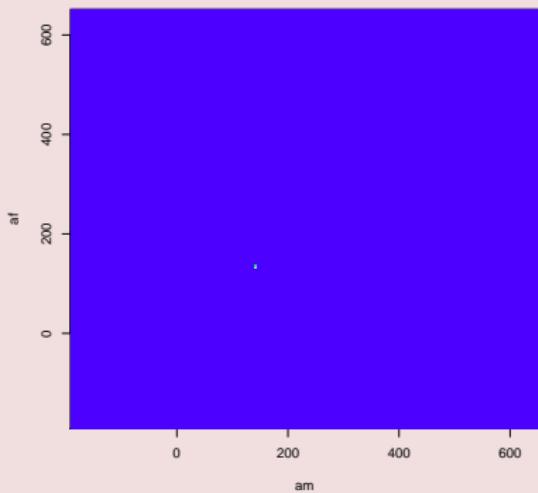


Figure : The two posteriors

Several categories

- For k categories, we need $k - 1$ dummy variables.

Building category variables

```
data(milk)
unique(milk$clade)
```

```
[1] Strepsirrhine      New World Monkey Old World Monkey Ape
Levels: Ape New World Monkey Old World Monkey Strepsirrhine
```

Building category variables

```
milk$d.NMM <- ifelse(milk$clade == "New\u20d5World\u20d5Monkey", 1, 0)
milk$d.OMM <- ifelse(milk$clade == "Old\u20d5World\u20d5Monkey", 1, 0)
milk$d.S  <- ifelse(milk$clade == "Strepsirrhine", 1, 0)
head(milk[,c("clade", "d.NMM", "d.OMM", "d.S")])
```

	clade	d.NMM	d.OMM	d.S
1	Strepsirrhine	0	0	1
6	New World Monkey	1	0	0
7	New World Monkey	1	0	0
8	New World Monkey	1	0	0
10	New World Monkey	1	0	0
11	New World Monkey	1	0	0
12	New World Monkey	1	0	0
13	New World Monkey	1	0	0
16	Old World Monkey	0	1	0
18	Old World Monkey	0	1	0

Building category variables by intercept

$$kcal_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha_{j[i]}$$

$$\alpha_j \sim N(0.6, 10), j = 1, 2, 3, \dots, J$$

$$\sigma \sim U(0, 10)$$

- $j[i]$ means persons i category.
- Example if person $i = 1$ belongs to category 2, then $\mu_1 = \alpha_2$.

Building category variables

```
milk$clade_id <- coerce_index(milk$clade)
head(milk[,c("clade", "clade_id")],5)
```

	clade	clade_id
1	Strepsirrhine	4
6	New World Monkey	2
7	New World Monkey	2
8	New World Monkey	2
10	New World Monkey	2

Building category variables

```
data(milk)
milk <- milk[complete.cases(milk),]
milk$clade_id <- coerce_index(milk$clade)
model <- map(
  alist(
    kcal.per.g ~ dnorm(mu, sigma),
    mu ~ alpha[clade_id],
    alpha[clade_id] ~ dnorm(0.6, 10),
    sigma ~ dunif(0, 10)
  ),
  data = milk)
precis(model, depth = 2)
```

	Mean	StdDev	5.5%	94.5%
alpha[1]	0.57	0.05	0.48	0.65
alpha[2]	0.69	0.05	0.61	0.77
alpha[3]	0.83	0.08	0.71	0.95
alpha[4]	0.49	0.13	0.28	0.70
sigma	0.13	0.02	0.10	0.17