

# milk2

Chris Parrish

June 20, 2016

## Contents

<b>milk</b>	<b>1</b>
data	1
exploratory data analysis	2
model	4
map	4
parameter distributions	5
plot parameter distribution	5
map	6
plot parameter distribution	7
map	8
plot parameter distributions	9
correlated predictor variables	11

milk2

references:

- McElreath, Statistical Rethinking, chap 5, pp.119-164
- cowplot vignette
- plotting multiple figures

## milk

```
library(rethinking)
library(ggplot2)
library(cowplot)
theme_set(theme_gray())
```

## data

```
## R code 5.35
```

```
data(milk)
d <- milk
str(d)
```

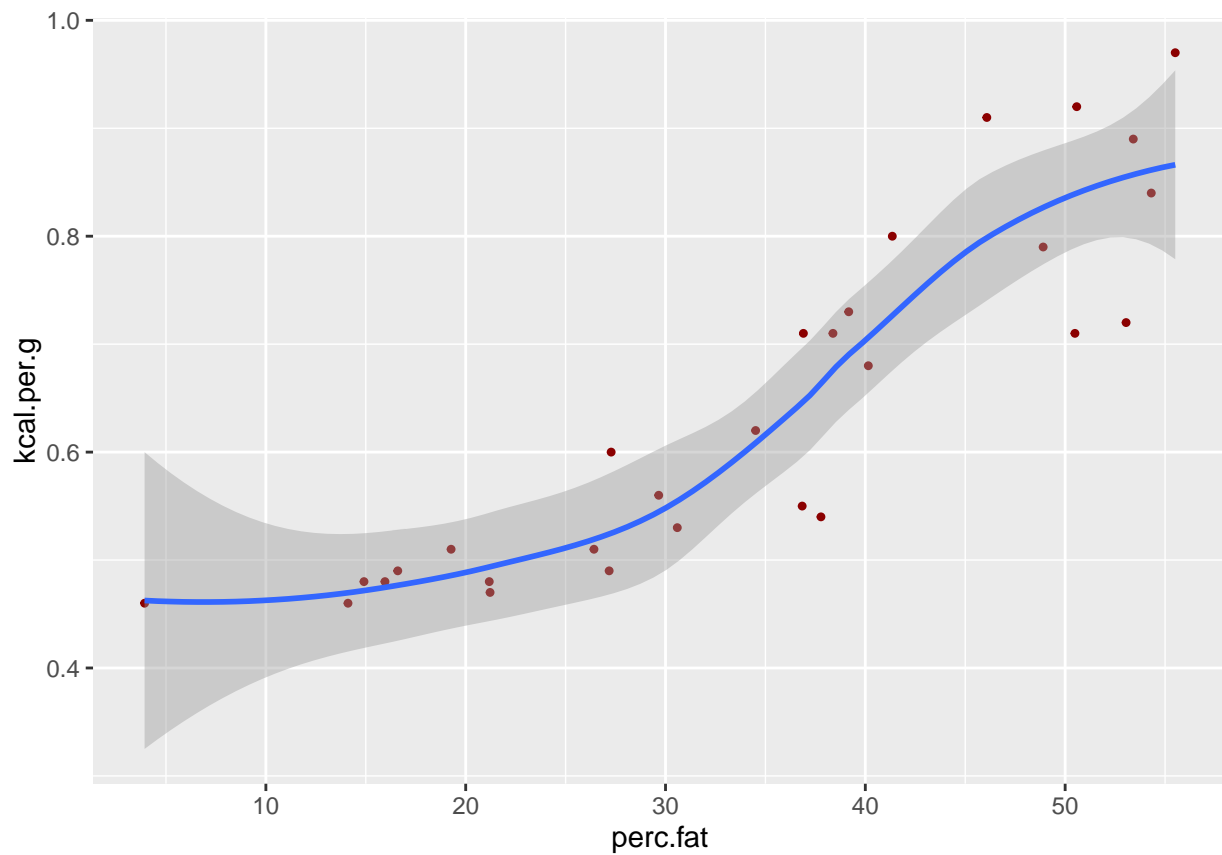
```
## 'data.frame':   29 obs. of  8 variables:
## $ clade         : Factor w/ 4 levels "Ape","New World Monkey",...: 4 4 4 4 4 2 2 2 2 ...
## $ species       : Factor w/ 29 levels "A palliata","Alouatta seniculus",...: 11 8 9 10 16 2 1 6 28 2
## $ kcal.per.g    : num  0.49 0.51 0.46 0.48 0.6 0.47 0.56 0.89 0.91 0.92 ...
## $ perc.fat      : num  16.6 19.3 14.1 14.9 27.3 ...
## $ perc.protein  : num  15.4 16.9 16.9 13.2 19.5 ...
## $ perc.lactose  : num  68 63.8 69 71.9 53.2 ...
## $ mass         : num  1.95 2.09 2.51 1.62 2.19 5.25 5.37 2.51 0.71 0.68 ...
```

```
## $ neocortex.perc: num 55.2 NA NA NA NA ...
```

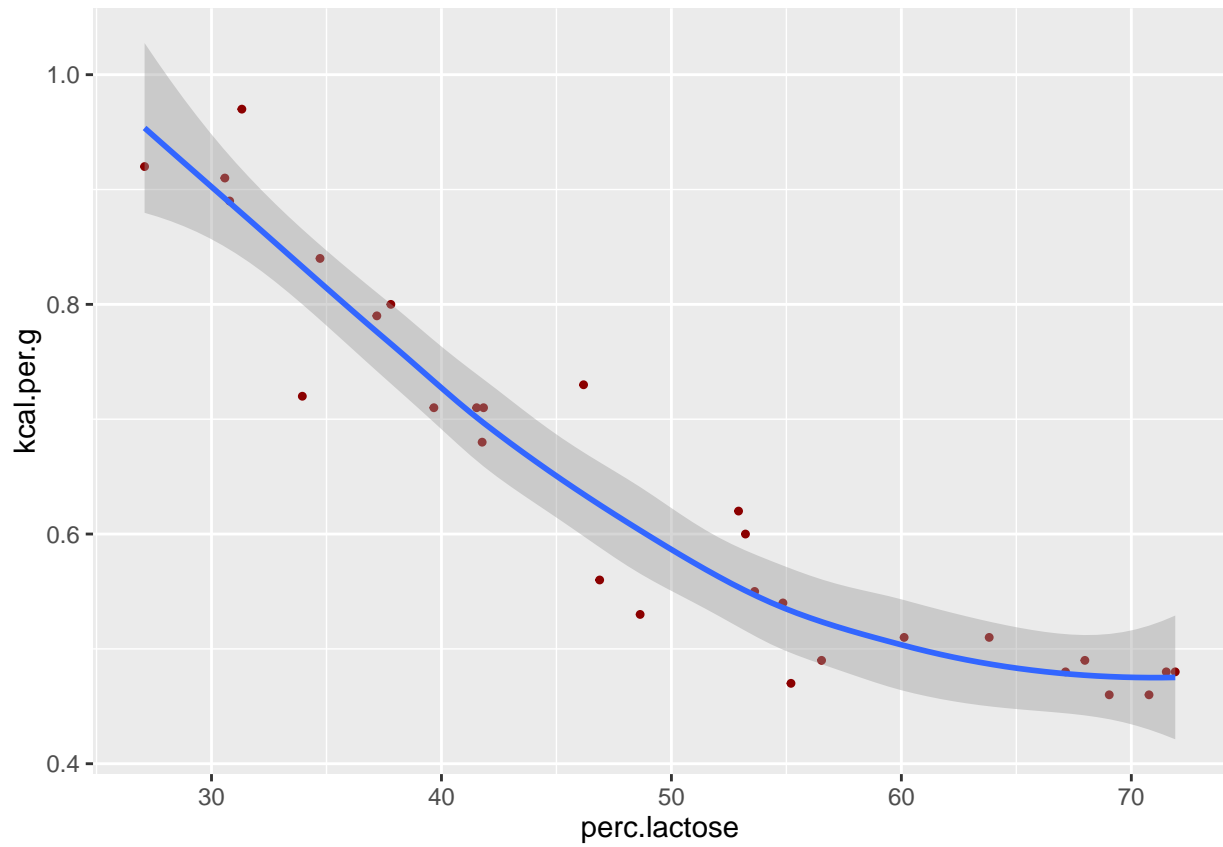
## exploratory data analysis

Two predictor variables are associated.

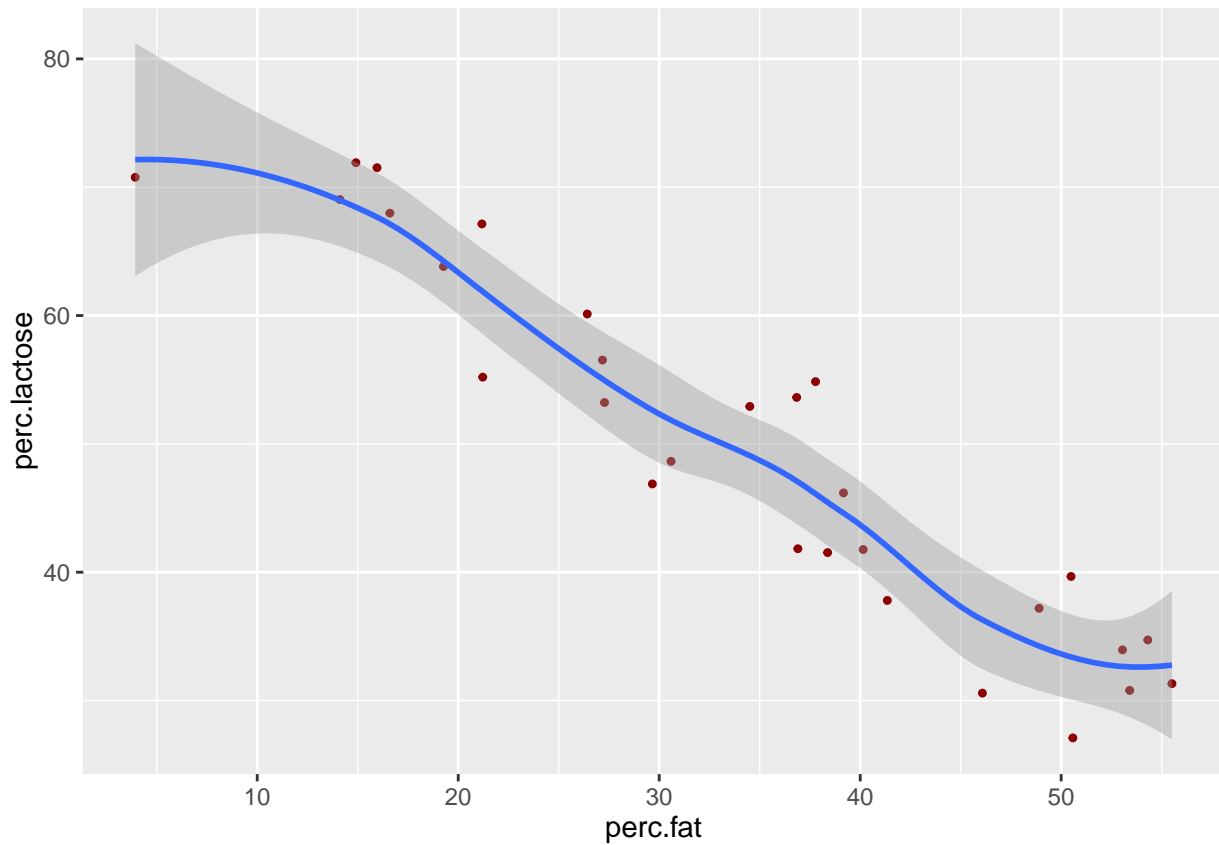
```
ggplot(d, aes(perc.fat, kcal.per.g)) +  
  geom_point(shape = 20, color = "darkred") +  
  geom_smooth()
```



```
ggplot(d, aes(perc.lactose, kcal.per.g)) +  
  geom_point(shape = 20, color = "darkred") +  
  geom_smooth()
```



```
ggplot(d, aes(perc.fat, perc.lactose)) +  
  geom_point(shape = 20, color = "darkred") +  
  geom_smooth()
```



model

$$E_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_f \text{perc.fat} + \beta_l \text{perc.lactose}$$

$$\alpha \sim \text{Normal}(0.6, 10)$$

$$\beta_f \sim \text{Normal}(0, 1)$$

$$\beta_l \sim \text{Normal}(0, 1)$$

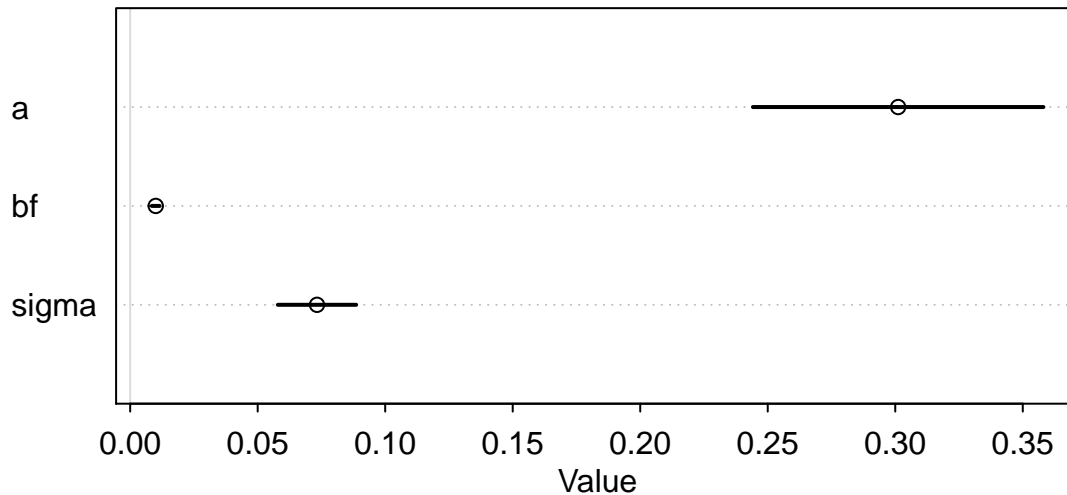
$$\sigma \sim \text{Uniform}(0, 10)$$

map

```
## R code 5.36
# kcal.per.g regressed on perc.fat
m5.10 <- map(
  alist(
    kcal.per.g ~ dnorm( mu , sigma ) ,
    mu <- a + bf*perc.fat ,
    a ~ dnorm( 0.6 , 10 ) ,
    bf ~ dnorm( 0 , 1 ) ,
    sigma ~ dunif( 0 , 10 )
  ) ,
  data=d )
precis( m5.10 , digits=3 )
```

```
##      Mean StdDev  5.5% 94.5%
## a    0.301  0.036 0.244 0.358
## bf   0.010  0.001 0.008 0.012
## sigma 0.073  0.010 0.058 0.089
```

```
plot(precis( m5.10 , digits=3 ))
```



parameter distributions

plot parameter distribution

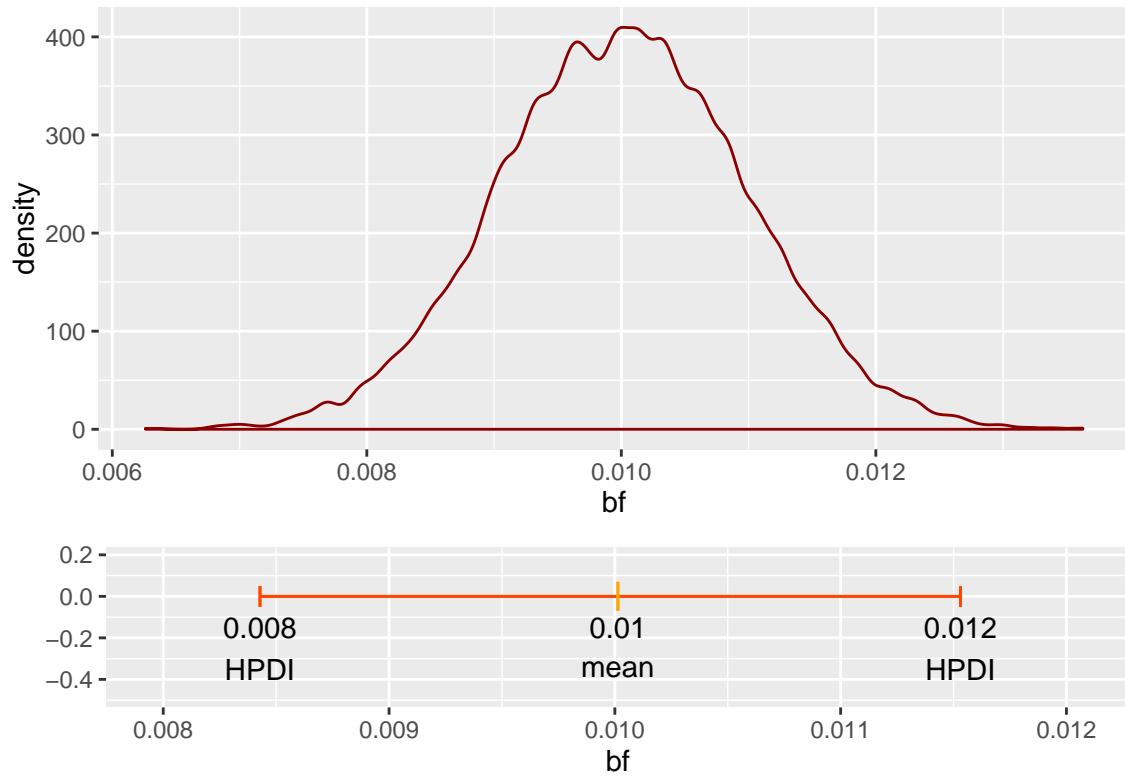
```
# extract samples
```

```
post <- extract.samples( m5.10 )
str(post)
```

```
## 'data.frame':  10000 obs. of  3 variables:
## $ a      : num  0.312 0.342 0.265 0.328 0.322 ...
## $ bf     : num  0.00966 0.00849 0.01097 0.00941 0.00892 ...
## $ sigma : num  0.0965 0.0782 0.0693 0.0853 0.0763 ...
```

```
# plot parameter distribution
```

```
bf.plot <- parameter.dist(parameter = "bf", values = post$bf)
bf.plot
```



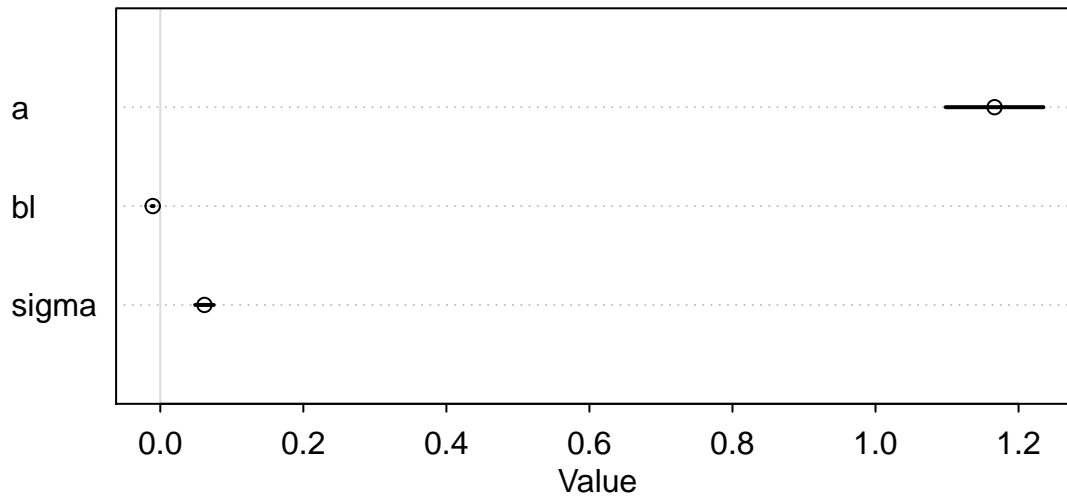
## map

Start value (1)

```
# kcal.per.g regressed on perc.lactose
m5.11 <- map(
  alist(
    kcal.per.g ~ dnorm( mu , sigma ) ,
    mu <- a + bl*perc.lactose ,
    a ~ dnorm( 0.6 , 10 ) ,
    bl ~ dnorm( 0 , 1 ) ,
    sigma ~ dunif( 0 , 10 )
  ) ,
  data=d )
precis( m5.11 , digits=3 )
```

```
##      Mean StdDev  5.5% 94.5%
## a      1.166  0.043  1.098  1.235
## bl     -0.011  0.001 -0.012 -0.009
## sigma  0.062  0.008  0.049  0.075
```

```
plot( precis( m5.11 , digits=3 ) )
```

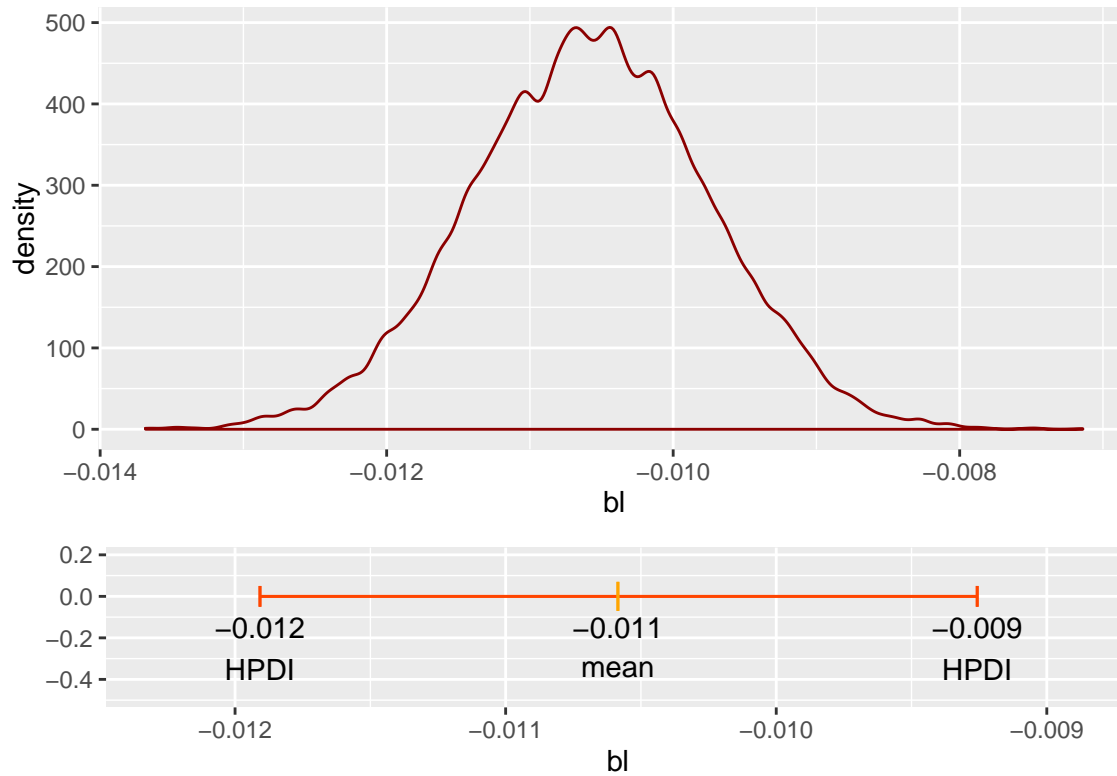


### plot parameter distribution

```
# extract samples
post <- extract.samples( m5.11 )
str(post)

## 'data.frame':  10000 obs. of  3 variables:
## $ a      : num  1.09 1.18 1.12 1.2 1.15 ...
## $ bl     : num -0.00897 -0.01067 -0.01009 -0.01108 -0.01025 ...
## $ sigma: num  0.065 0.0667 0.0671 0.0613 0.0589 ...

# plot parameter distribution
bl.plot <- parameter.dist(parameter = "bl", values = post$bl)
bl.plot
```



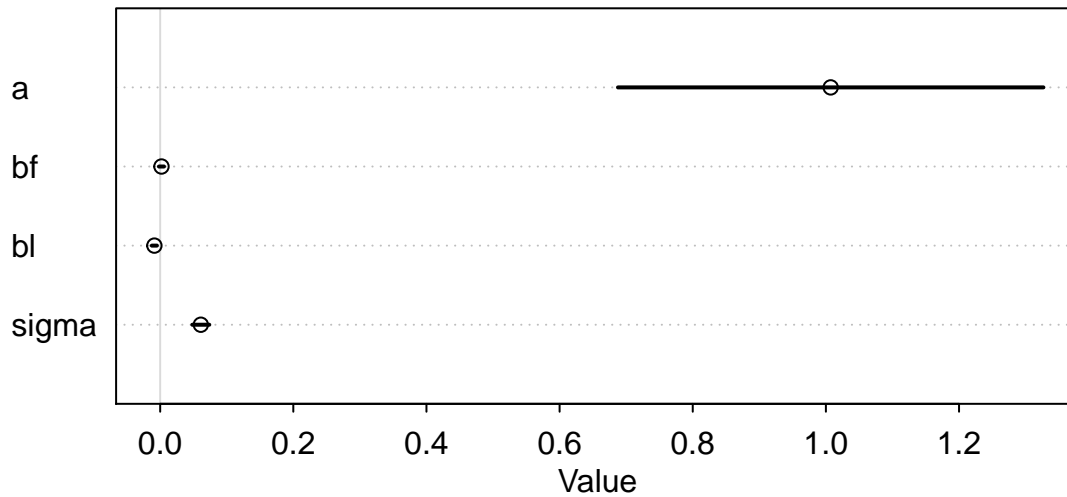
## map

```
## R code 5.37
m5.12 <- map(
  alist(
    kcal.per.g ~ dnorm( mu , sigma ) ,
    mu <- a + bf*perc.fat + bl*perc.lactose ,
    a ~ dnorm( 0.6 , 10 ) ,
    bf ~ dnorm( 0 , 1 ) ,
    bl ~ dnorm( 0 , 1 ) ,
    sigma ~ dunif( 0 , 10 )
  ) ,
  data=d )
precis( m5.12 , digits=3 )

##      Mean StdDev  5.5% 94.5%
## a      1.007  0.200  0.688  1.327
## bf      0.002  0.002 -0.002  0.006
## bl     -0.009  0.002 -0.013 -0.005
## sigma  0.061  0.008  0.048  0.074

plot( precis( m5.12 , digits=3 ) )
```



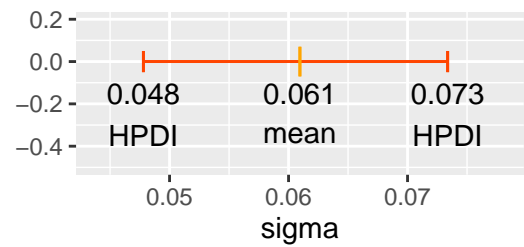
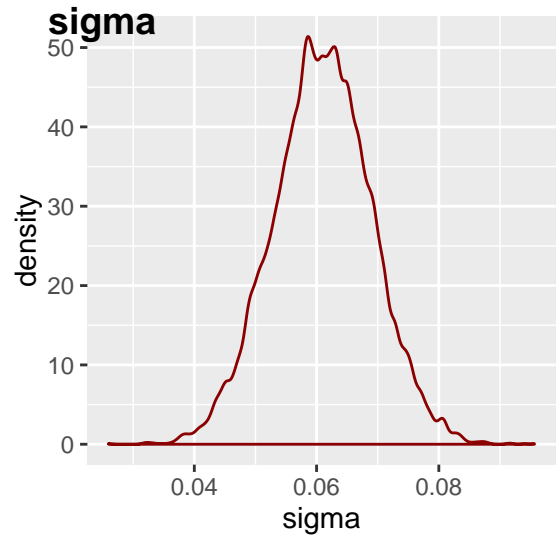
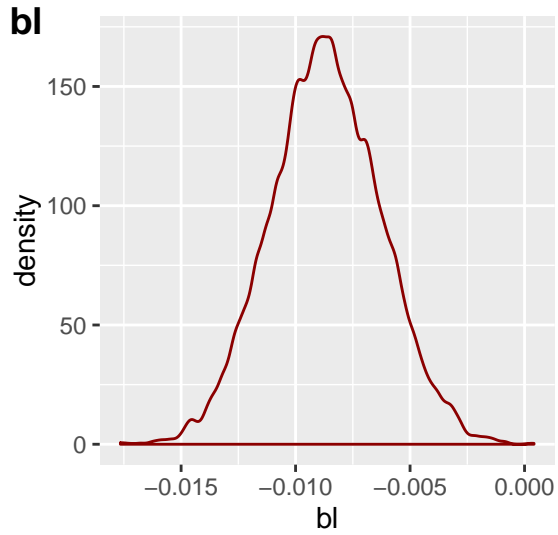
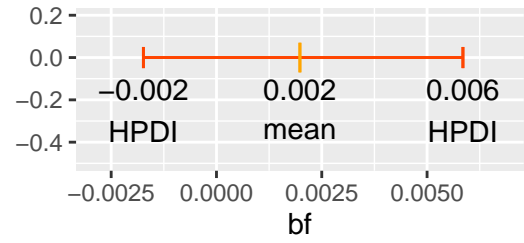
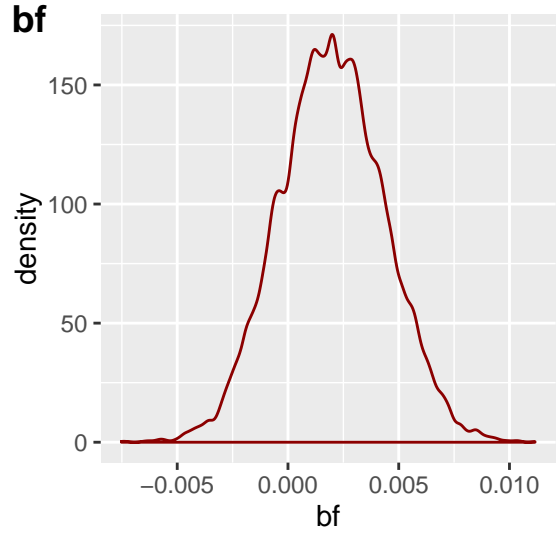
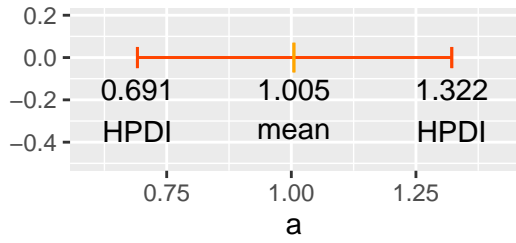
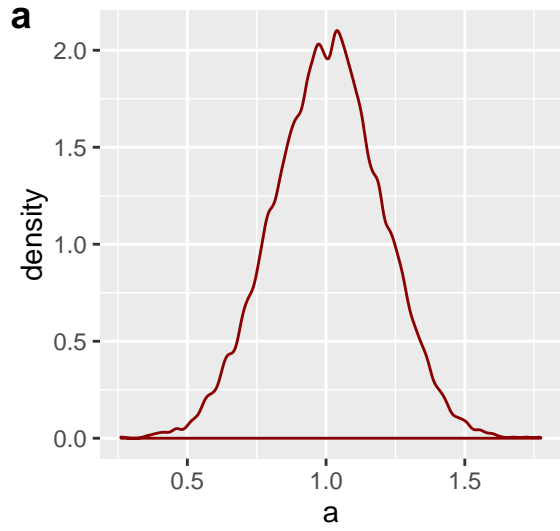


plot parameter distributions

```
# extract samples
post <- extract.samples( m5.12 )
str(post)

## 'data.frame': 10000 obs. of 4 variables:
## $ a : num 0.795 0.986 0.87 0.942 0.845 ...
## $ bf : num 0.00393 0.00232 0.0029 0.00198 0.0039 ...
## $ bl : num -0.0058 -0.00865 -0.00665 -0.00735 -0.00664 ...
## $ sigma: num 0.0579 0.0557 0.0533 0.0586 0.0511 ...

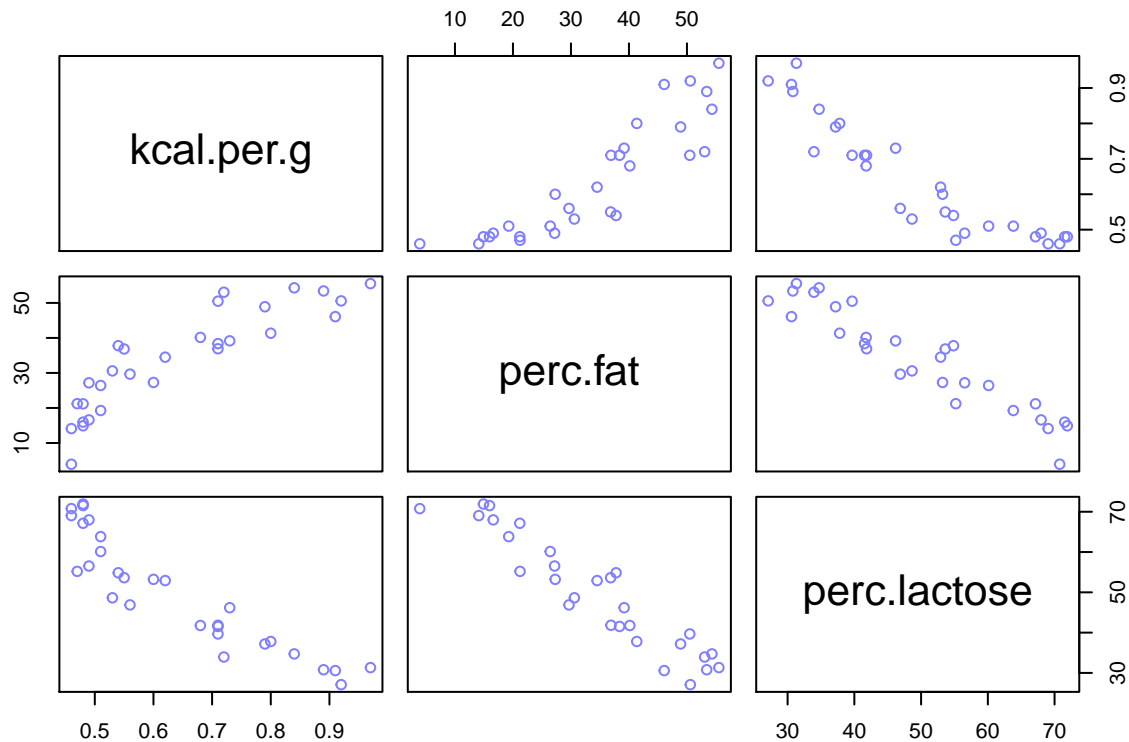
# plot each parameter distribution
a.plot <- parameter.dist(parameter = "a", values = post$a)
bf.plot <- parameter.dist(parameter = "bf", values = post$bf)
bl.plot <- parameter.dist(parameter = "bl", values = post$bl)
sigma.plot <- parameter.dist(parameter = "sigma", values = post$sigma)
# display parameter distributions
plot_grid(a.plot, bf.plot, bl.plot, sigma.plot,
          labels=c("a", "bf", "bl", "sigma"), ncol = 2, nrow = 2)
```



## correlated predictor variables

*perc.fat* and *perc.lactose* are inversely related

```
## R code 5.38
pairs( ~ kcal.per.g + perc.fat + perc.lactose ,
      data=d , col=rangl2 )
```



```
## R code 5.39
cor( d$perc.fat , d$perc.lactose )
```

```
## [1] -0.9416373
```

Simulation

```
## R code 5.40
data(milk)
d <- milk
sim.coll <- function( r=0.9 ) {
  d$x <- rnorm( nrow(d) , mean=r*d$perc.fat ,
               sd=sqrt( (1-r^2)*var(d$perc.fat) ) )
  m <- lm( kcal.per.g ~ perc.fat + x , data=d )
  sqrt( diag( vcov(m) ) ) [2] # stddev of parameter
}
rep.sim.coll <- function( r=0.9 , n=100 ) {
  stddev <- replicate( n , sim.coll(r) )
  mean(stddev)
}
r.seq <- seq(from=0,to=0.99,by=0.01)
stddev <- sapply( r.seq , function(z) rep.sim.coll(r=z,n=100) )
plot( stddev ~ r.seq , type="l" , col=rangl2 , lwd=2 , xlab="correlation" )
```

