

milk

Chris Parrish

June 18, 2016

Contents

milk	1
data	1
map	1
m5.5 with NA	1
m5.5 complete cases	2
m5.6	3
m5.7	4

milk

reference:

- McElreath, Statistical Rethinking, chap 5, pp.119-164

milk

data

```
## R code 5.16
library(rethinking)
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 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 ...
```

map

m5.5 with NA

Problem here with incomplete cases.

```
## R code 5.17
# m5.5 <- map(
#   alist(
#     kcal.per.g ~ dnorm(mu, sigma),
```

```

#      mu <- a + bn*neocortex.perc ,
#      a ~ dnorm( 0 , 100 ) ,
#      bn ~ dnorm( 0 , 1 ) ,
#      sigma ~ dunif( 0 , 1 )
#    ) ,
#    data=d )

## R code 5.18
d$neocortex.perc

## [1] 55.16    NA    NA    NA    NA 64.54 64.54 67.64    NA 68.85 58.85
## [12] 61.69 60.32    NA    NA 69.97    NA 70.41    NA 73.40    NA 67.53
## [23]    NA 71.26 72.60    NA 70.24 76.30 75.49

```

m5.5 complete cases

```

## R code 5.19
dcc <- d[ complete.cases(d) , ]

a.start <- mean(dcc$kcal.per.g)
bn.start <- 0

## R code 5.20
m5.5 <- map(
  alist(
    kcal.per.g ~ dnorm( mu , sigma ) ,
    mu <- a + bn*neocortex.perc ,
    a ~ dnorm( 0 , 100 ) ,
    bn ~ dnorm( 0 , 1 ) ,
    sigma ~ dunif( 0 , 1 )
  ) ,
  data = dcc,
  start = list(a=a.start, bn=bn.start))

## R code 5.21
precis( m5.5 , digits=3 )

##      Mean StdDev  5.5% 94.5%
## a      0.353  0.471 -0.399 1.106
## bn     0.005  0.007 -0.007 0.016
## sigma 0.166  0.028  0.120 0.211

## R code 5.22
coef(m5.5)["bn"] * ( 76 - 55 )

##      bn
## 0.09456716

## R code 5.23
np.seq <- 0:100
pred.data <- data.frame( neocortex.perc=np.seq )

mu <- link( m5.5 , data=pred.data , n=1e4 )

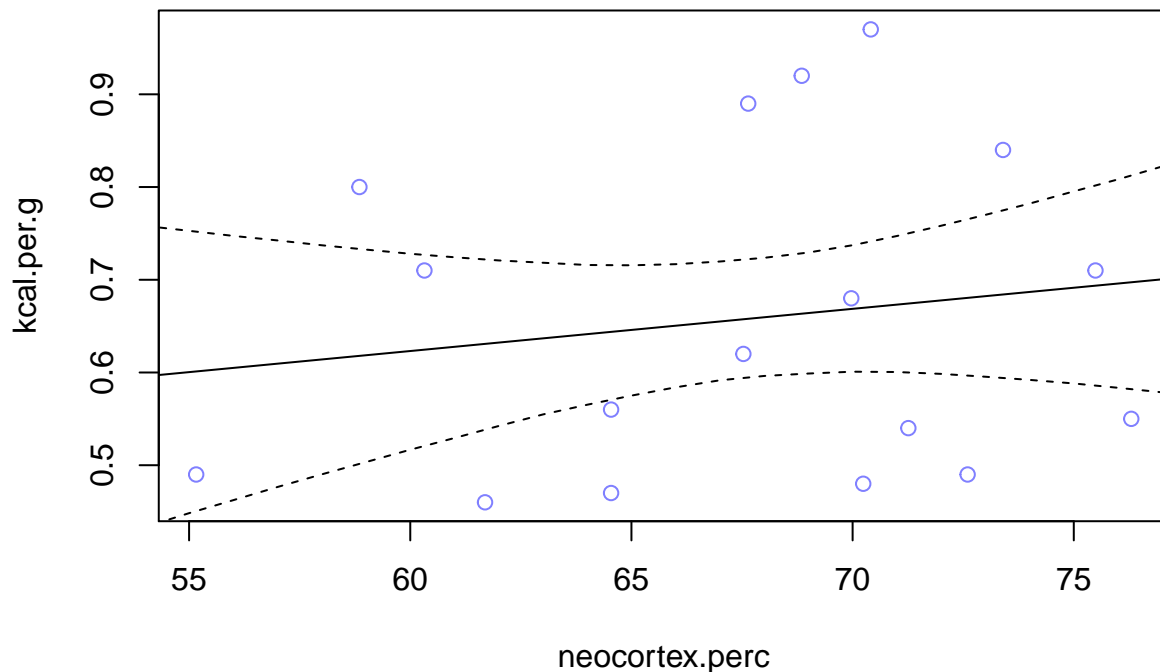
## [ 1000 / 10000 ]

```

```
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```
mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI )
```

```
plot( kcal.per.g ~ neocortex.perc , data=dcc , col=rangi2 )
lines( np.seq , mu.mean )
lines( np.seq , mu.PI[1,] , lty=2 )
lines( np.seq , mu.PI[2,] , lty=2 )
```



```
## R code 5.24
dcc$log.mass <- log(dcc$mass)
```

m5.6

```
## R code 5.25
m5.6 <- map(
  alist(
    kcal.per.g ~ dnorm( mu , sigma ) ,
    mu <- a + bm*log.mass ,
    a ~ dnorm( 0 , 100 ) ,
    bm ~ dnorm( 0 , 1 ) ,
    sigma ~ dunif( 0 , 1 )
  ) ,
```

```

data=dcc )
precis(m5.6)

##      Mean StdDev 5.5% 94.5%
## a      0.71  0.05  0.63  0.78
## bm     -0.03  0.02 -0.06  0.00
## sigma  0.16  0.03  0.11  0.20

```

m5.7

Bad start value (1)

```

## R code 5.26
m5.7 <- map(
  alist(
    kcal.per.g ~ dnorm( mu , sigma ) ,
    mu <- a + bn*neocortex.perc + bm*log.mass ,
    a ~ dnorm( 0 , 100 ) ,
    bn ~ dnorm( 0 , 1 ) ,
    bm ~ dnorm( 0 , 1 ) ,
    sigma ~ dunif( 0 , 1 )
  ) ,
  data=dcc )
precis(m5.7)

```

```

##      Mean StdDev 5.5% 94.5%
## a     -1.08  0.47 -1.83 -0.34
## bn      0.03  0.01  0.02  0.04
## bm     -0.10  0.02 -0.13 -0.06
## sigma  0.11  0.02  0.08  0.15

```

```

## R code 5.27
mean.log.mass <- mean( log(dcc$mass) )
np.seq <- 0:100
pred.data <- data.frame(
  neocortex.perc=np.seq,
  log.mass=mean.log.mass
)

mu <- link( m5.7 , data=pred.data , n=1e4 )

```

```

## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]

```

```

mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI )

```

```
plot( kcal.per.g ~ neocortex.perc , data=dcc , type="n" )  
lines( np.seq , mu.mean )  
lines( np.seq , mu.PI[1,] , lty=2 )  
lines( np.seq , mu.PI[2,] , lty=2 )
```

