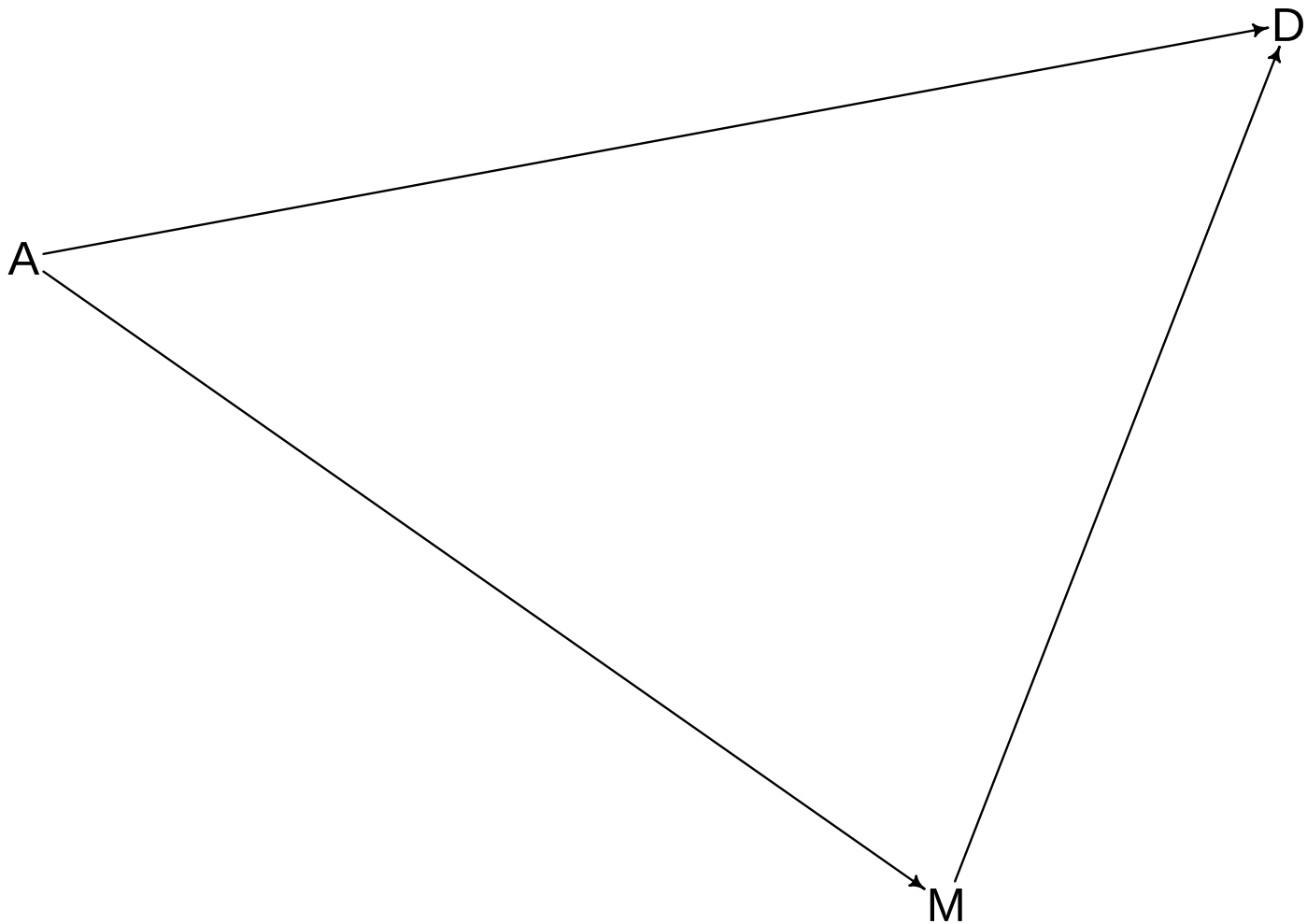
Causal models and multivariate regression

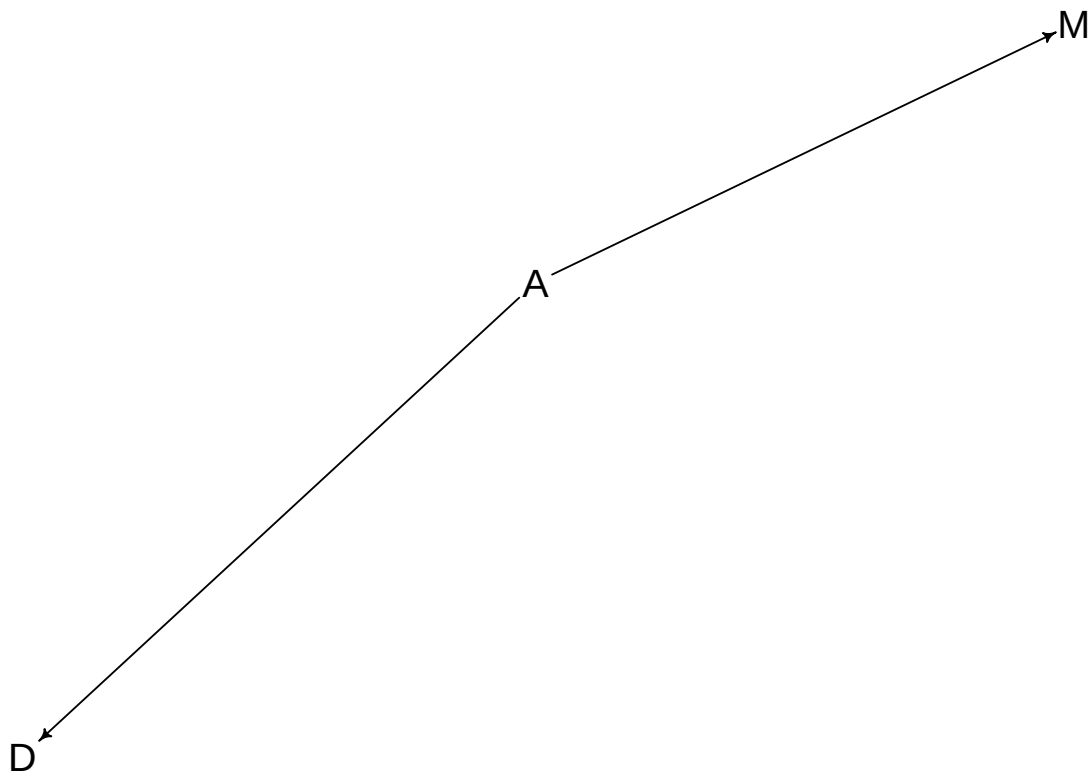
Rafał Urbaniak, Nikodem Lewandowski
(LoPSE research group, University of Gdansk)

DAG and divorce rate



DAG and divorce rate

```
dagWaffles2 <- dagitty(
  "dag{
  A -> D; A -> M
  }"
)

drawdag(dagWaffles2, goodarrow = TRUE, cex = 2, radius = 3)
```



DAG and divorce rate

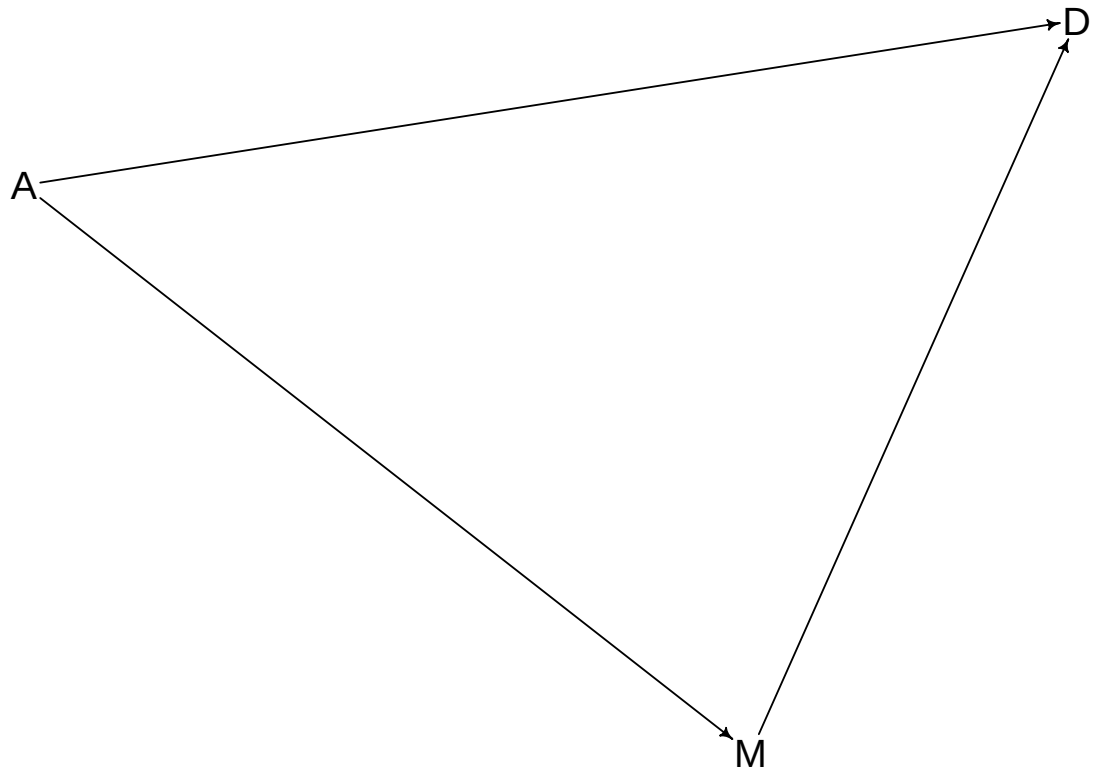
```
precis(ageModelNarrow)
```

```
##                  mean         sd       5.5%      94.5%
## a      1.140547e-07 0.10921783 -0.1745511  0.1745513
## bA    -5.681891e-01 0.11041016 -0.7446458 -0.3917323
## sigma  7.913979e-01 0.07877158  0.6655057  0.9172901
```

```
precis(marriageModelNarrow)
```

```
##                  mean         sd       5.5%     94.5%
## m     -2.274637e-06 0.12518997 -0.2000800 0.2000755
## bM     3.497872e-01 0.12645865  0.1476818 0.5518925
## sigma  9.143510e-01 0.09087051  0.7691223 1.0595796
```

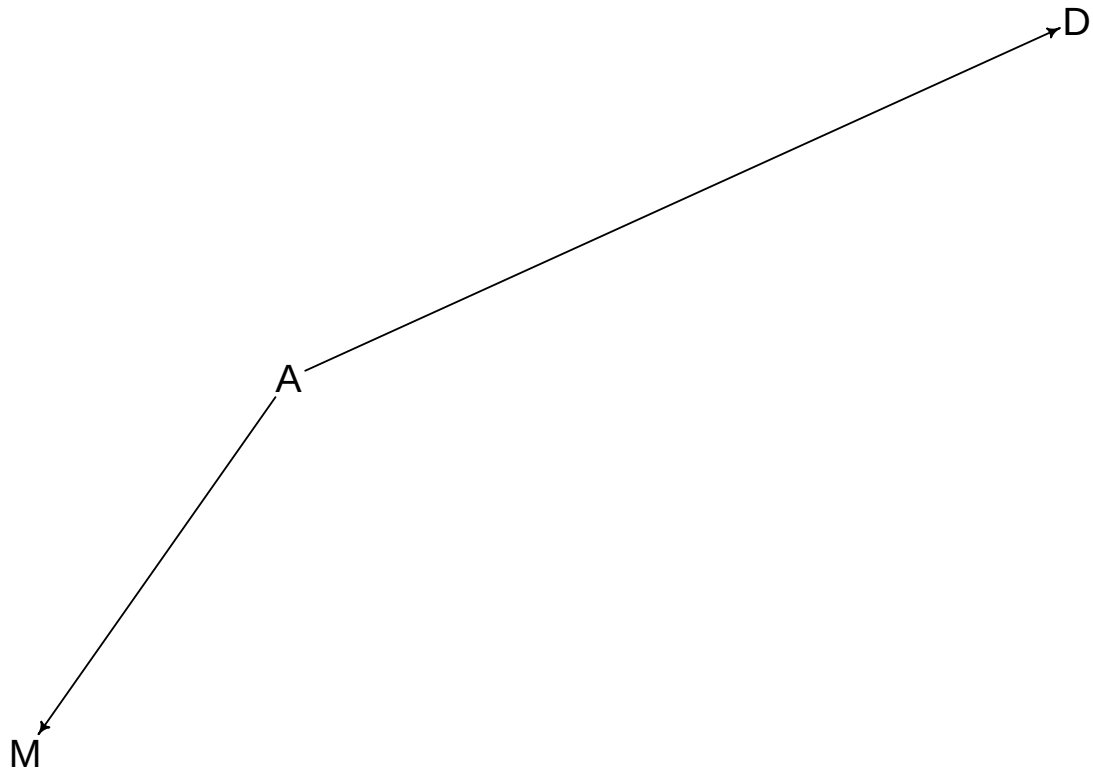
Figuring out independencies



Everything associated with everything else

$$\neg I(D, A), \neg I(D, M), \neg I(A, M)$$

Figuring out independencies



All information relevant for D is already in A

$$I(D, M)|A$$

Figuring out independencies

```
impliedConditionalIndependencies(dagWaffles1)
impliedConditionalIndependencies(dagWaffles2)
```

```
## D _||_ M | A
```

Guided multiple regression

```
marriageAgeModelNarrow <- quap(
  alist(
    D ~ dnorm(mu, sigma ) ,
    mu <- a + bA * A + bM * M,
    a ~ dnorm(0, .5),
    bA ~ dnorm( 0, .5),
    bM ~ dnorm( 0, .5),
    sigma ~ dexp( .5 )
  ), data = d
)
```

```
round(precis(marriageAgeModelNarrow),3)
```

```
##           mean    sd   5.5%  94.5%
## a        0.000 0.109 -0.174  0.174
## bA      -0.613 0.152 -0.855 -0.371
## bM      -0.065 0.151 -0.307  0.177
## sigma    0.788 0.079  0.663  0.914
```

Visualizing residuals

```
mu_m <- link(marriageModelNarrow)
mu_m_mean <- apply(mu_m, 2, mean)
mu_m_hpdi <- data.frame(t(apply(mu_m, 2, HPDI)))
mu_m_res <- mu_m_mean - d$D

mu_a <- link(ageModelNarrow)
mu_a_mean <- apply(mu_a, 2, mean)
mu_a_hpdi <- data.frame(t(apply(mu_a, 2, HPDI)))
mu_a_res <- mu_a_mean - d$D

mu_ma <- link(marriageAgeModelNarrow)
mu_ma_mean <- apply(mu_ma, 2, mean)
mu_ma_hpdi <- data.frame(t(apply(mu_ma, 2, HPDI)))
mu_ma_res <- mu_ma_mean - d$D

str(mu_m_mean)
```

```
##  num [1:50] 0.00894 0.54768 0.01823 0.58483 -0.09323 ...
```

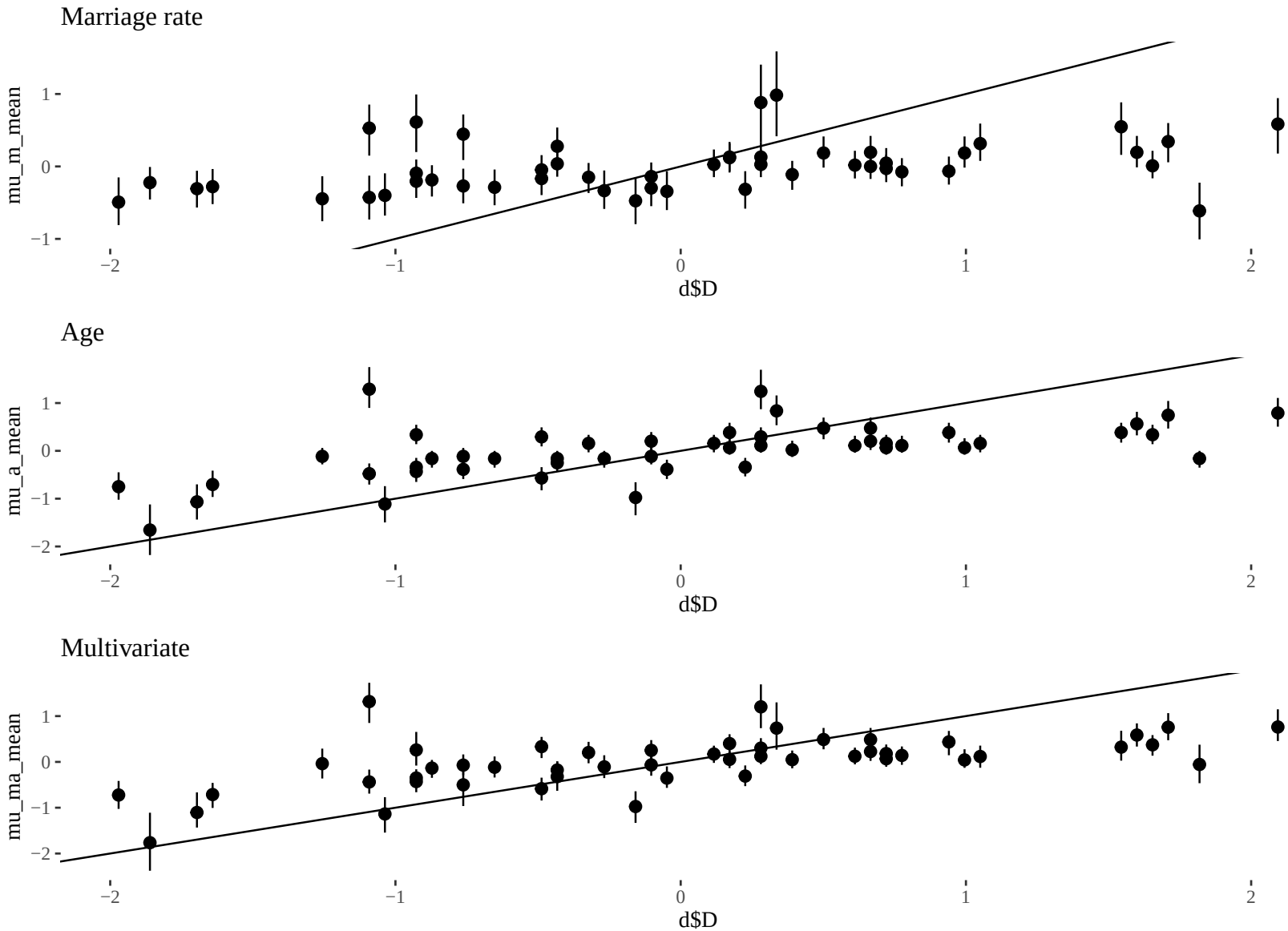
```
str(mu_m_hpdi)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ X.0.89: num  -0.164 0.16 -0.166 0.178 -0.297 ...
##  $ X0.89.: num  0.2157 0.8845 0.2161 0.9431 0.0956 ...
```

Predicted means, three models

```
plot_m <- ggplot()+geom_pointrange(aes(x = d$D, y = mu_m_mean,
                                   ymin = mu_m_hpdi[,1],
                                   ymax = mu_m_hpdi[,2]))+
                    geom_abline(intercept = 0, slope = 1)+
                    theme_tufte(base_size = 12)+
                    ggtitle("Marriage rate")

plot_a <- ggplot()+geom_pointrange(aes(x = d$D, y = mu_a_mean,
                                     ymin = mu_a_hpdi[,1],
                                     ymax = mu_a_hpdi[,2]))+
                    geom_abline(intercept = 0, slope = 1)+
                    theme_tufte(base_size = 12)+ggtitle("Age")

plot_ma <- ggplot()+geom_pointrange(aes(x = d$D, y = mu_ma_mean,
                                   ymin = mu_ma_hpdi[,1],
                                   ymax = mu_ma_hpdi[,2]))+
                    geom_abline(intercept = 0, slope = 1)+
                    theme_tufte(base_size = 12)+
                    ggtitle("Multivariate")
```

Predicted means, three models



Residuals, three models

```
df <- data.frame(m = mu_m_res, a = mu_a_res, ma = mu_ma_res )
head(df, n = 5)
```

```
##            m          a         ma
## 1 -1.6452638 -1.3162777 -1.2810417
## 2 -0.9966882 -1.1611292 -1.2216831
## 3 -0.5924859 -0.4993261 -0.4870045
## 4 -1.5087390 -1.3025662 -1.3318769
## 5  0.8338256  0.5853722  0.5760096
```

```
dfLong <- melt(df)
```

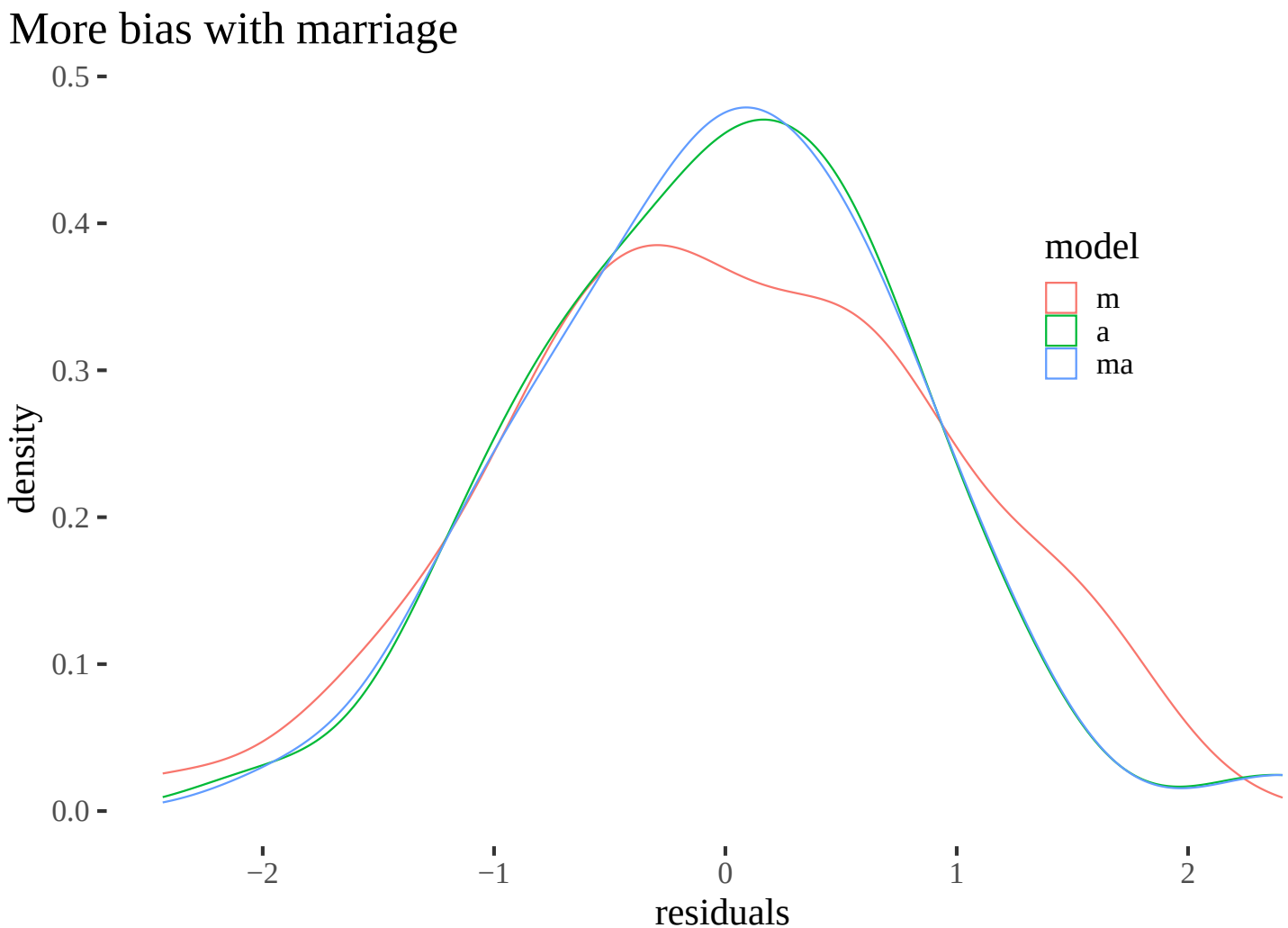
```
## No id variables; using all as measure variables
```

```
colnames(dfLong) <- c("model", "residuals")
head(dfLong, n = 5 )
```

```
##   model  residuals
## 1     m -1.6452638
## 2     m -0.9966882
## 3     m -0.5924859
## 4     m -1.5087390
## 5     m  0.8338256
```

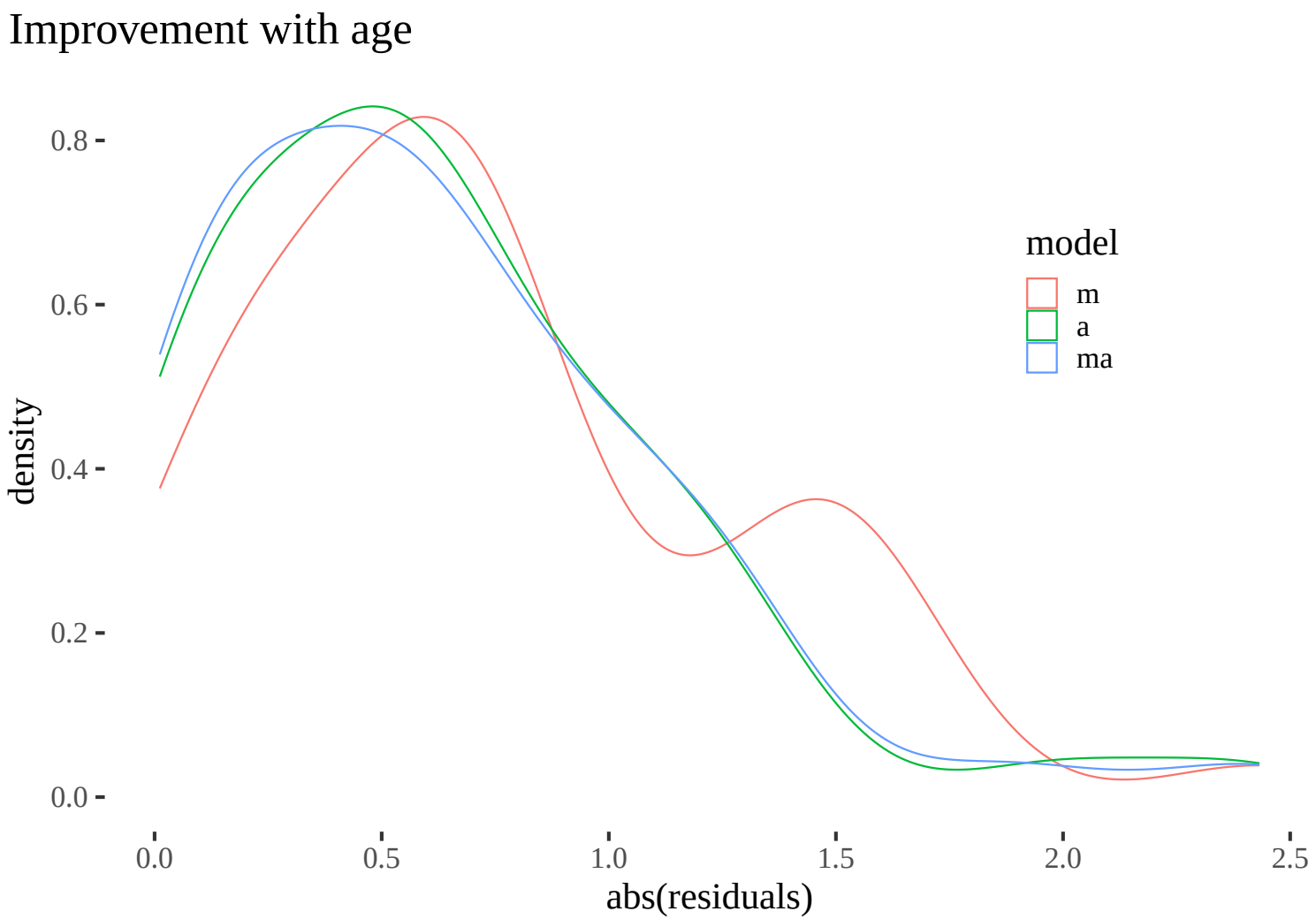
Residuals, three models

```
ggplot(dfLong)+geom_density(aes(x = residuals, color= model),
                            alpha = .2)+th +ggtitle("More bias with marriage")+
  theme(legend.position =  c(.8, .7))
```



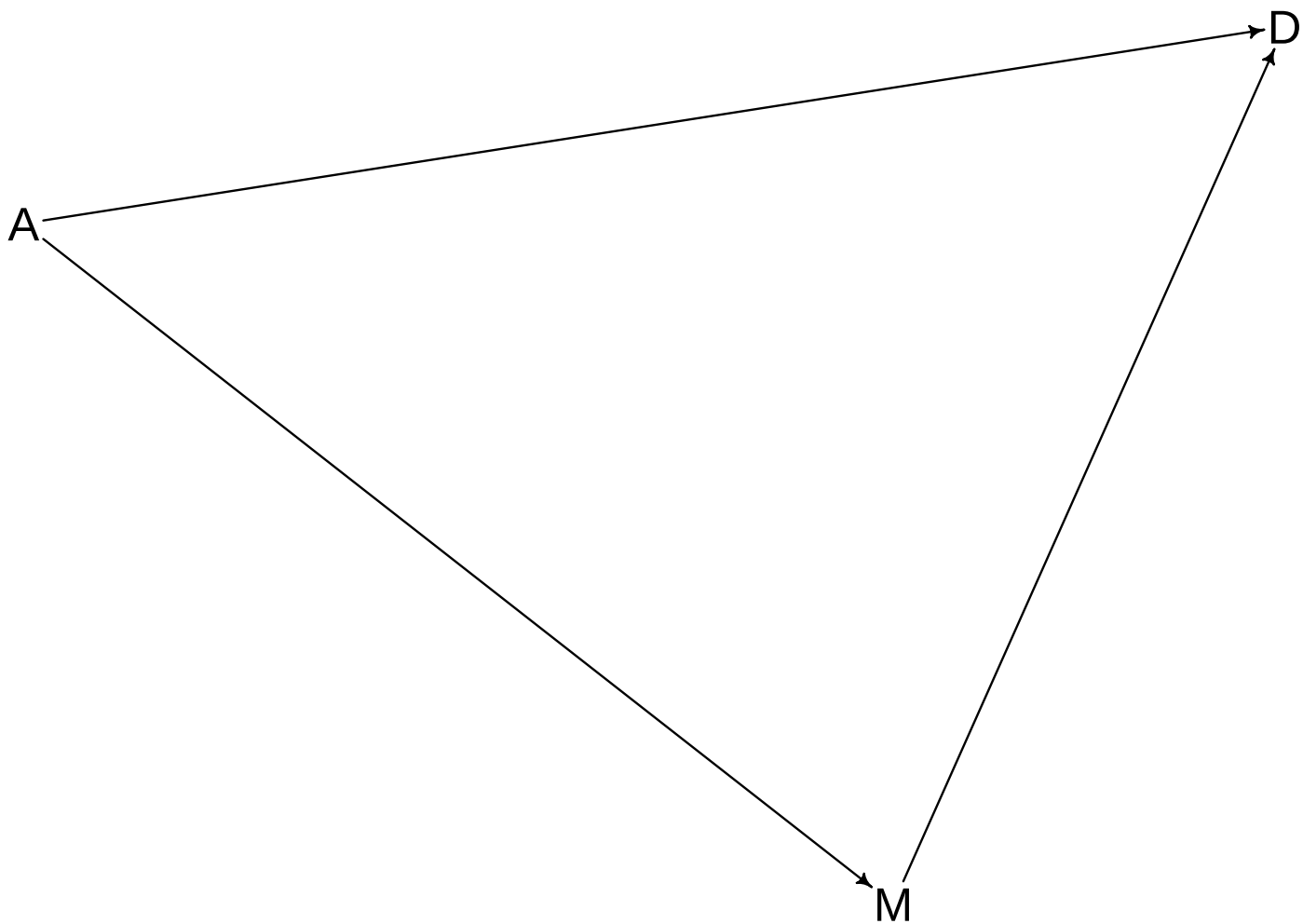
Residuals, three models

```
ggplot(dfLong)+geom_density(aes(x = abs(residuals), color= model),
                            alpha = .2)+th + ggtitle("Improvement with age")+
  theme(legend.position =  c(.8, .7))
```



Counterfactual plots

- Pick intervention variable and a range for it
- for each sample from the posterior, simulate the values of other variables



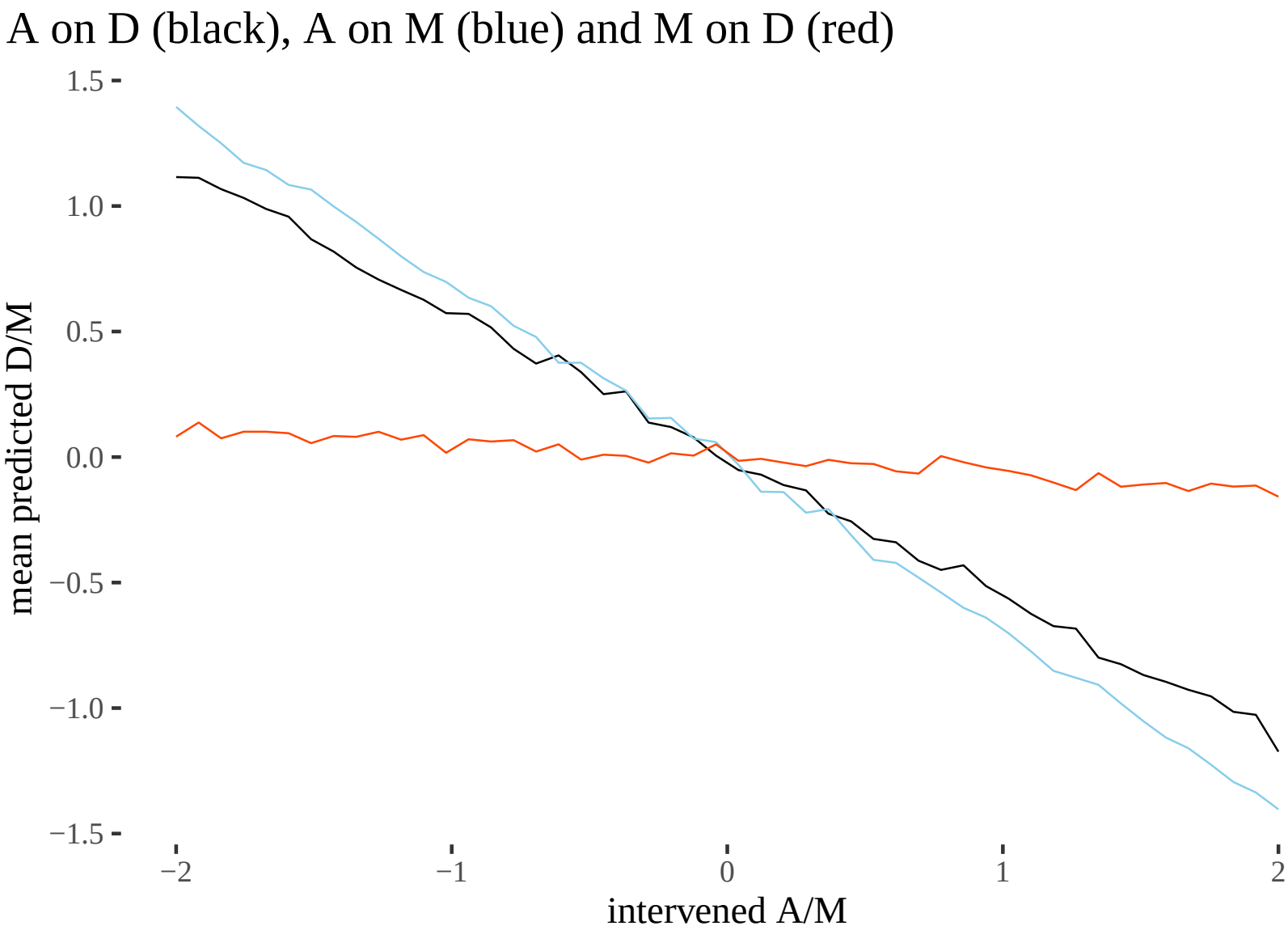
Counterfactual plots

```
DAG_Model <- quap(
  alist(
    ## A -> D <- M
    D ~ dnorm(mu, sigma ) ,
    mu <- a + bA * A + bM * M,
    a ~ dnorm(0, .5),
    bA ~ dnorm( 0, .5),
    bM ~ dnorm( 0, .5),
    sigma ~ dexp( .5 ),
    # A -> M
    M ~ dnorm(mu_M, sigma_M),
    mu_M <- aM + bAM * A,
    aM ~ dnorm( 0, .5),
    bAM ~ dnorm( 0, .5),
    sigma_M ~ dexp(.5)
  ) , data = d
)
```

Counterfactual plots

```
A_seq <- seq(-2,2, length.out = 50)
dag_sim <- sim(DAG_Model,
               data = data.frame(A = A_seq),
               vars = c("M", "D")
               )

M_seq <- seq(-2,2, length.out = 50)
dag_sim_M <- sim(DAG_Model,
               data = data.frame(M = M_seq, A = 0),
               vars = c("D")
)
```

Counterfactual plots



Masking

Hypothesis

Primates with larger brains produce more energetic milk.

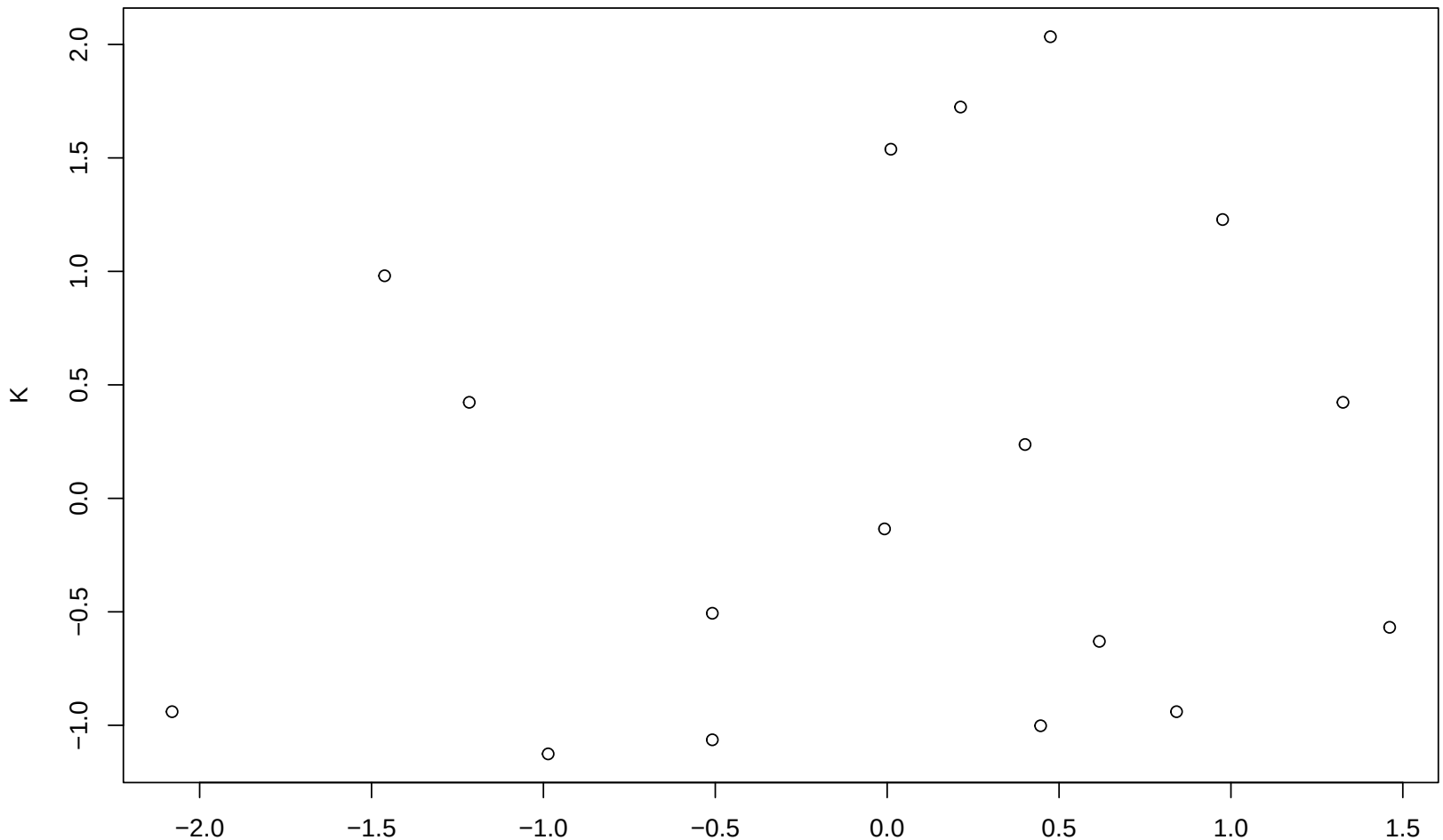
```
##                clade              species kcal.per.g perc.fat perc.protein
## 1     Strepsirrhine      Eulemur fulvus       0.49    16.60        15.42
## 2     Strepsirrhine           E macaco       0.51    19.27        16.91
## 3     Strepsirrhine           E mongoz       0.46    14.11        16.85
## 4     Strepsirrhine       E rubriventer      0.48    14.91        13.18
## 5     Strepsirrhine         Lemur catta      0.60    27.28        19.50
## 6 New World Monkey Alouatta seniculus       0.47    21.22        23.58
##   perc.lactose mass neocortex.perc
## 1        67.98 1.95          55.16
## 2        63.82 2.09             NA
## 3        69.04 2.51             NA
## 4        71.91 1.62             NA
## 5        53.22 2.19             NA
## 6        55.20 5.25          64.54
```

```
d$K <- standardize(d$kcal.per.g)
d$N <- standardize(d$neocortex.perc)
d$M <- standardize(log(d$mass))
```

Masking

Note small size

```
plot(K ~ N, data = d)
```



Masking

Model error, vimin not finite

```
milk_try <- quap(
  alist(
    K ~ dnorm( mu, sigma),
    mu <- a + bN * N,
    a ~ dnorm(0, 1),
    bN ~ dnorm( 0, 1),
    sigma ~ dexp(1)
  ), data = d
)
```

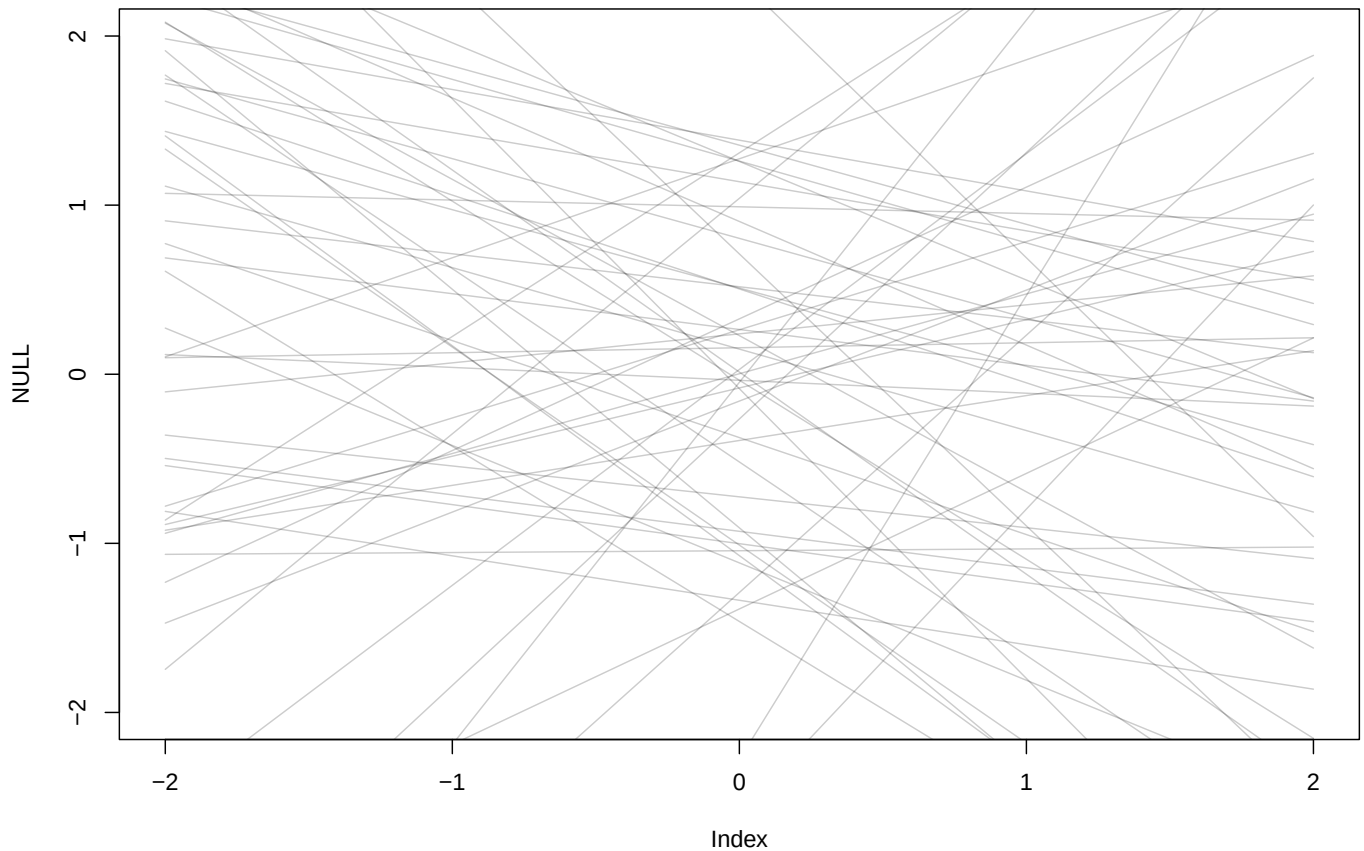
```
dc <- d[complete.cases(d$K, d$N, d$M), ]
```

Masking

```
milk_try2 <- quap(
  alist(
    K ~ dnorm( mu, sigma),
    mu <- a + bN * N,
    a ~ dnorm(0, 1),
    bN ~ dnorm( 0, 1),
    sigma ~ dexp(1)
  ), data = dc
)
```

Check your priors!

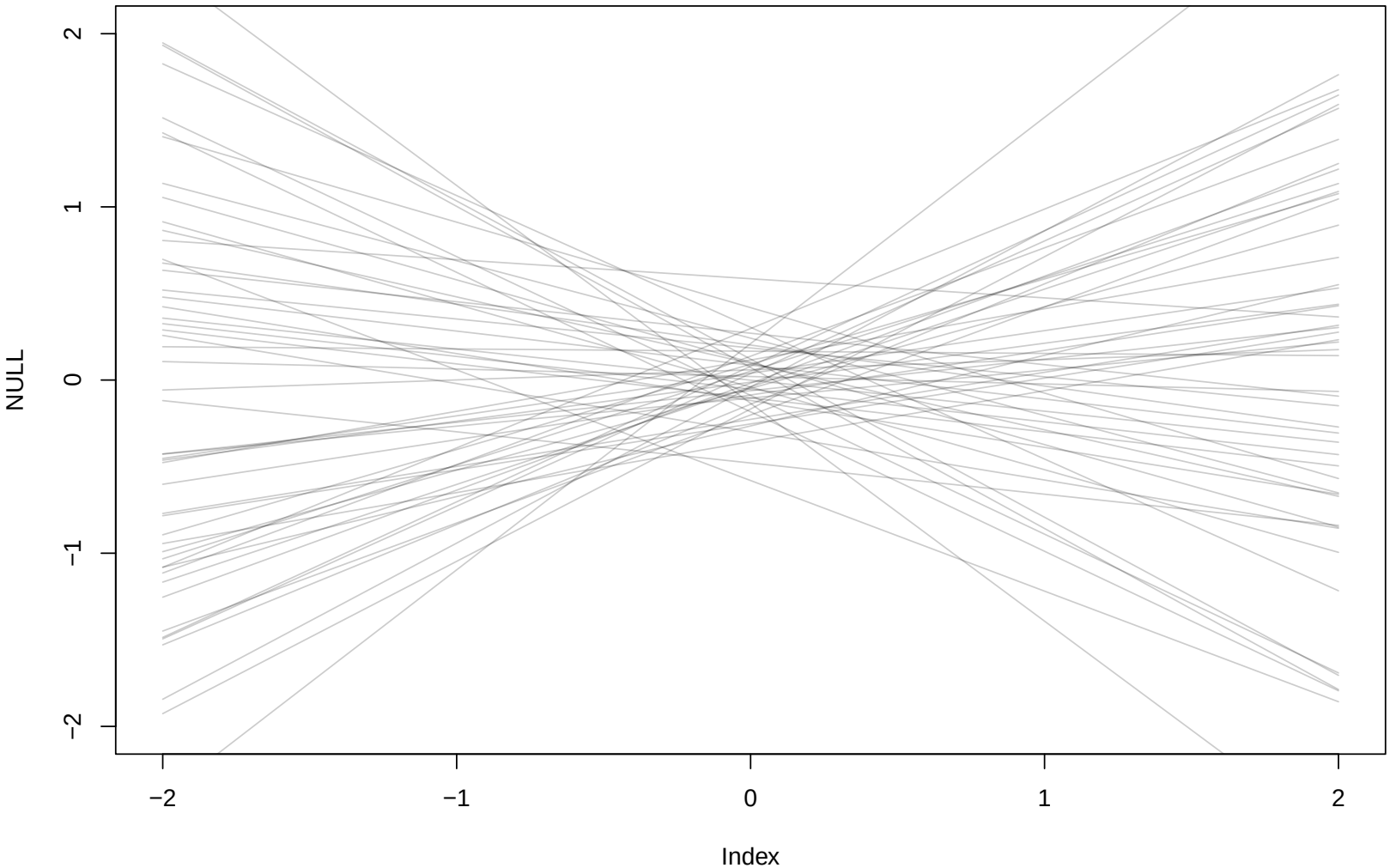
```
prior <- extract.prior(milk_try2)
xseq <- seq(-2,2,length.out = 30)
mu <- link(milk_try2, post = prior, data = list(N = xseq))

plot( NULL, xlim = c(-2,2), ylim = c(-2,2))
for (i in 1:50 ) lines (xseq, mu[i,], col = col.alpha("black", .2))
```



Check your priors!

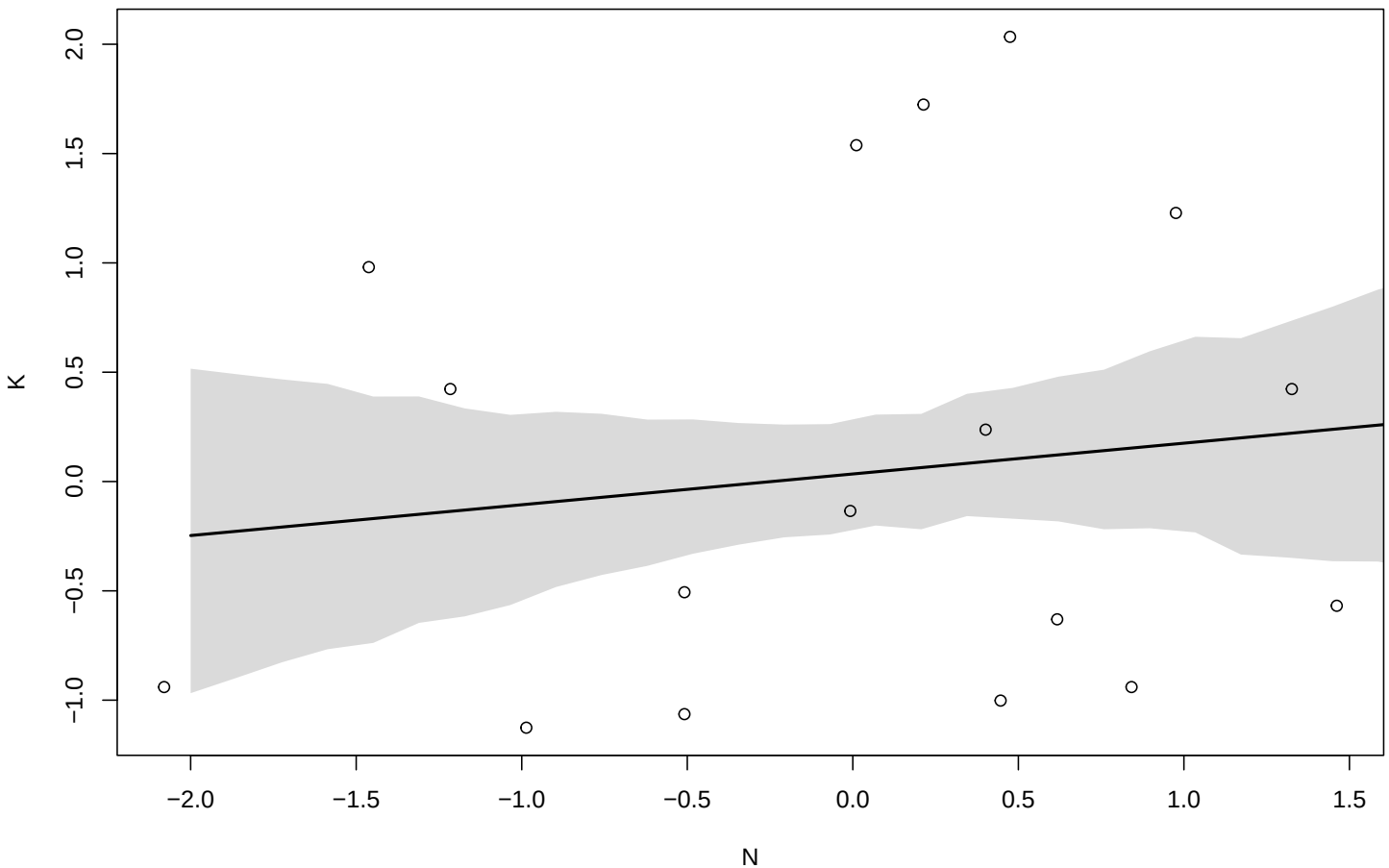
```
milk_n <- quap(
  alist(
    K ~ dnorm( mu, sigma),
    mu <- a + bN * N,
    a ~ dnorm(0, .2),
    bN ~ dnorm( 0, .5),
    sigma ~ dexp(1)
  ), data = dc
)
```

Check your priors!



Posterior for neocortex percentage

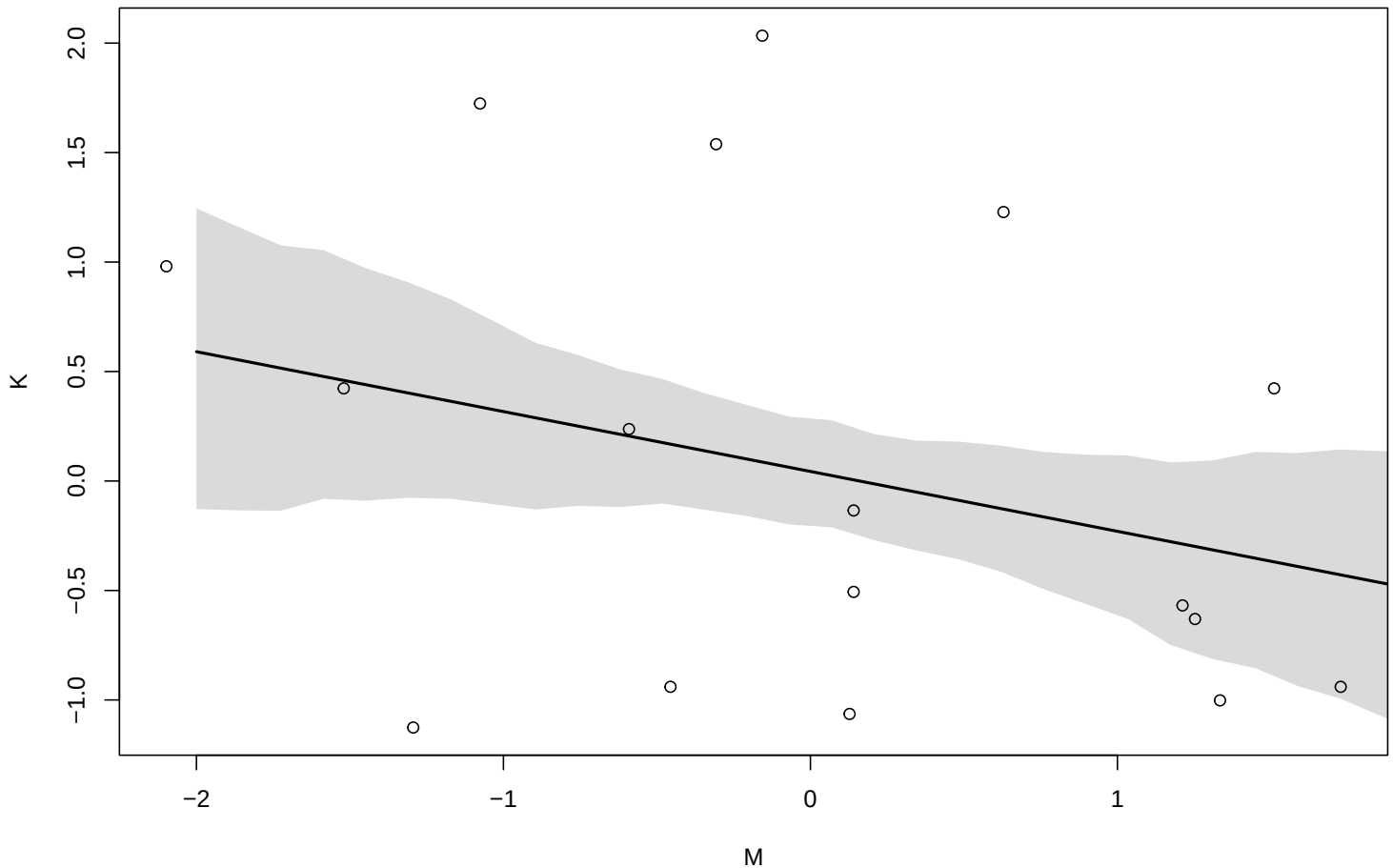
```
##              mean         sd       5.5%     94.5%
## a      0.03993969 0.1544908 -0.2069664 0.2868458
## bN     0.13323453 0.2237469 -0.2243563 0.4908253
## sigma  0.99982066 0.1647082  0.7365852 1.2630562
```



How about mass?

```
milk_m <- quap(
  alist(
    K ~ dnorm( mu, sigma),
    mu <- a + bM * M,
    a ~ dnorm(0, .2),
    bM ~ dnorm( 0, .5),
    sigma ~ dexp(1)
  ), data = dc
)
```

How about mass?



How about mass?

```
##                mean         sd       5.5%      94.5%
## a       0.04654135 0.1512801 -0.1952334 0.28831610
## bM     -0.28253582 0.1928818 -0.5907983 0.02572663
## sigma   0.94927974 0.1570617  0.6982649 1.20029461
```

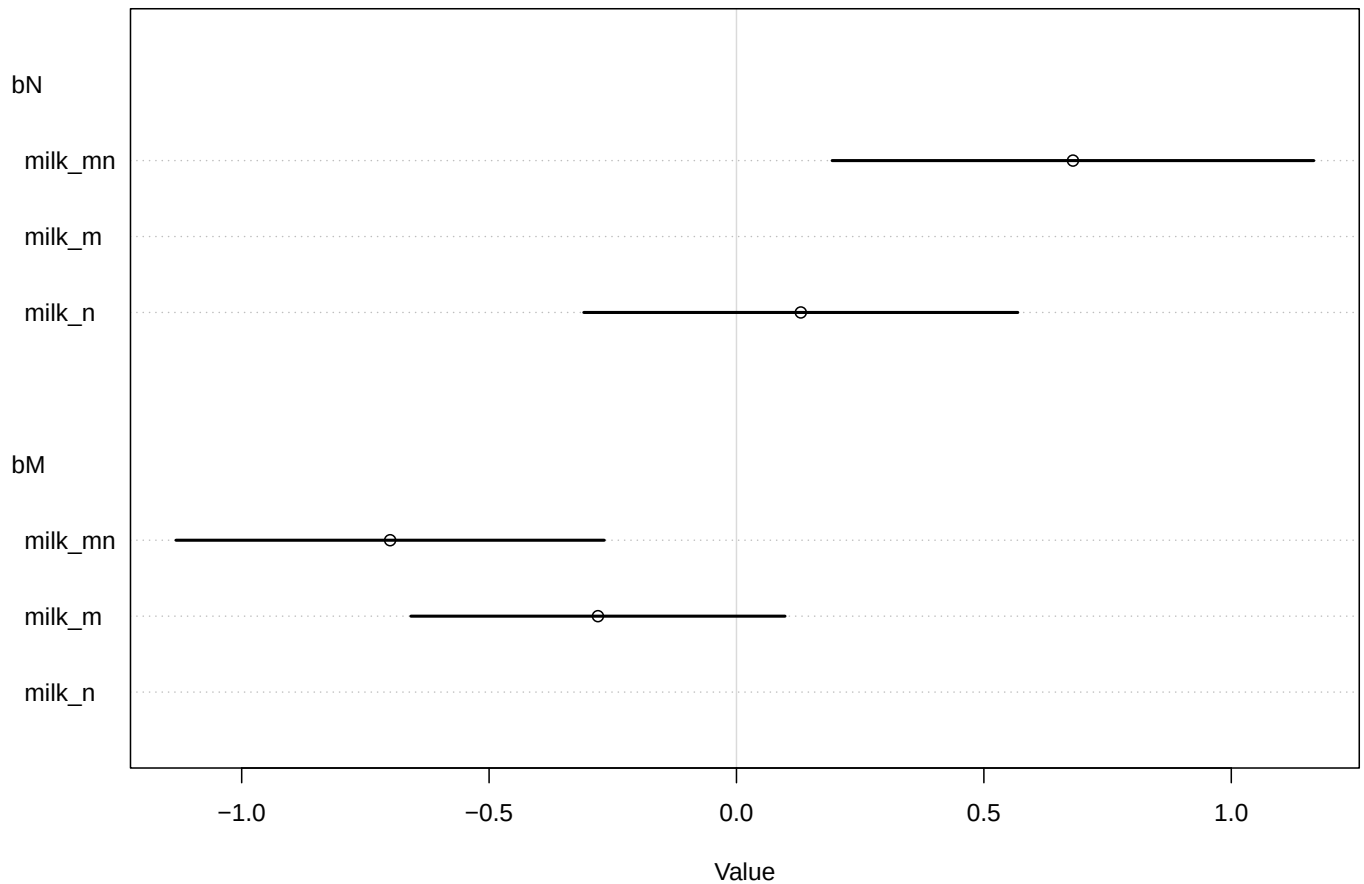
Now with both predictors

```
milk_mn <- quap(
  alist(
    K ~ dnorm( mu, sigma),
    mu <- a + bN * N + bM * M,
    a ~ dnorm(0, .2),
    bM ~ dnorm( 0, .5),
    bN ~ dnorm( 0, .5),
sigma ~ dexp(1)
  ), data = dc
)
```

```
##              mean        sd       5.5%      94.5%
## a       0.0679926 0.1339987 -0.1461632  0.2821484
## bM     -0.7029909 0.2207871 -1.0558514 -0.3501304
## bN      0.6751191 0.2482986  0.2782900  1.0719482
## sigma   0.7380148 0.1324621  0.5263148  0.9497147
```

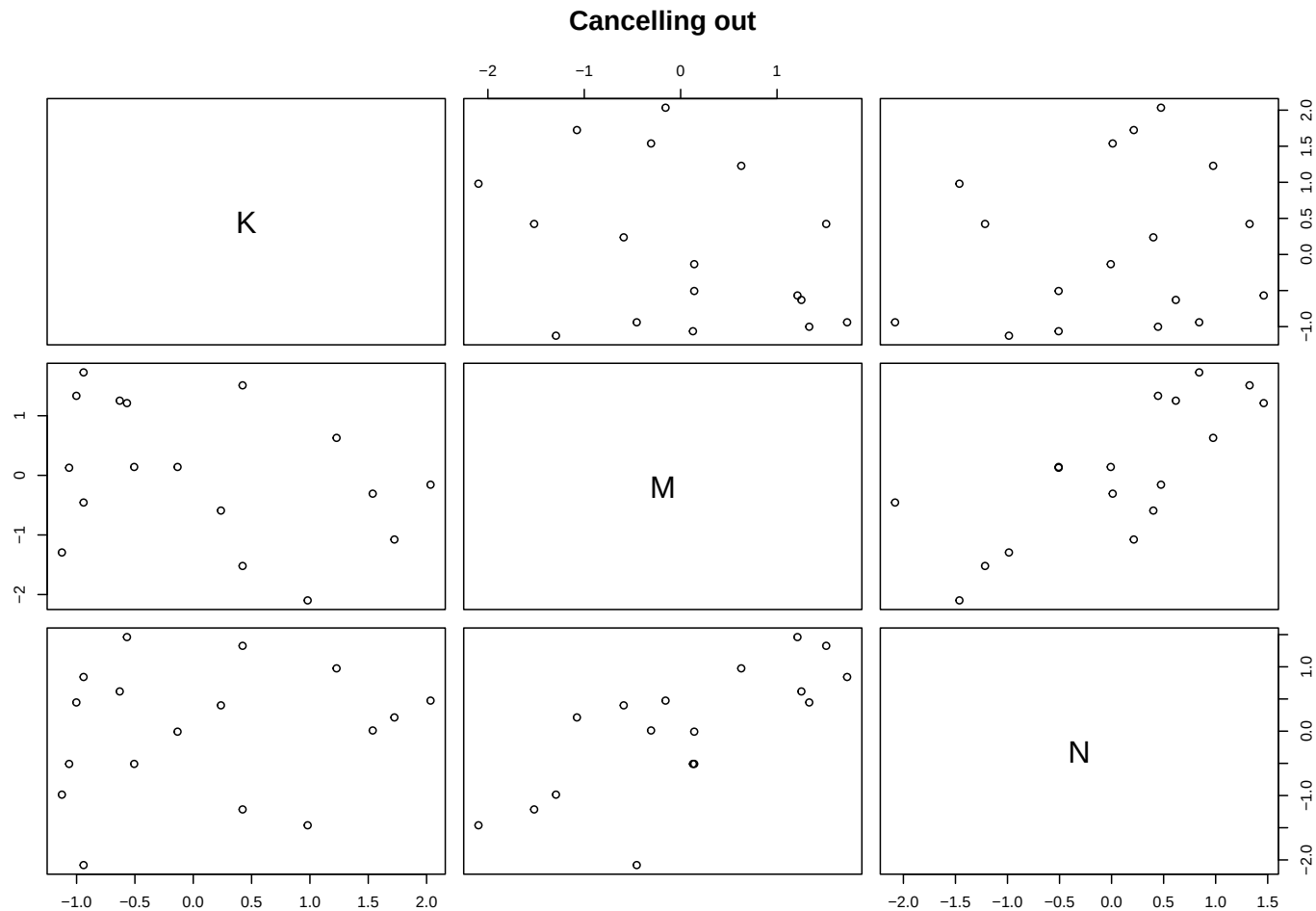
Now with both predictors

```
plot(coeftab(milk_n, milk_m, milk_mn), pars = c("bN", "bM"))
```



Now with both predictors

```
pairs( ~K + M + N , dc, main = "Cancelling out" )
```



Now with both predictors

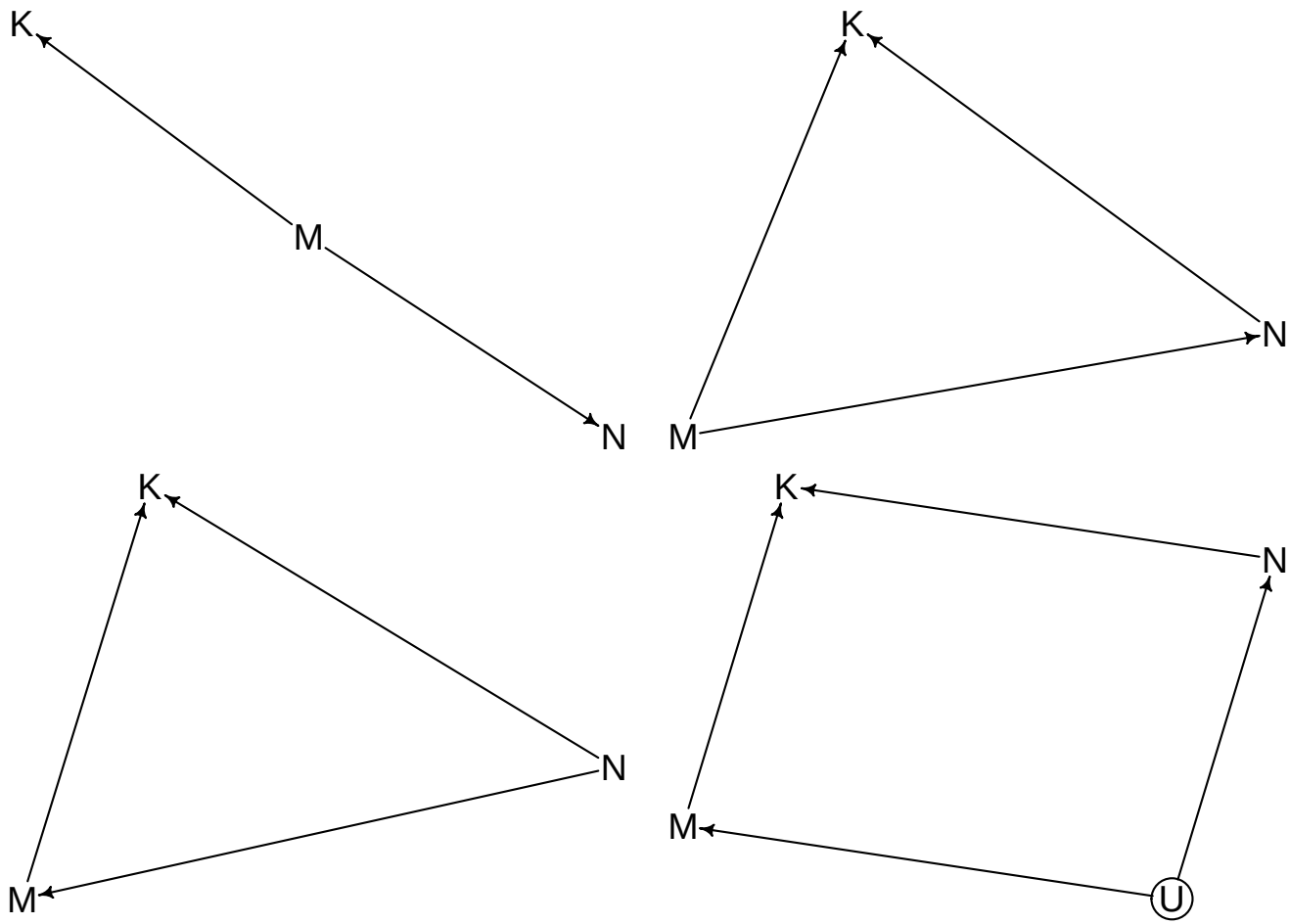
- predictors are positively correlated
- each has impact on the outcome variable
- those outcomes are opposite

Now with DAGs

```
#first three chunks
milkDAG1a <- dagitty("dag {
K <-  M -> N
} " )

milkDAG1 <- dagitty("dag {
K <-  M -> N
N -> K
} " )

milkDAG2 <- dagitty("dag {
 N -> M -> K
 N -> K
} " )

milkDAG3 <- dagitty("dag {
 U [unobserved]
 U -> M
 U -> N
 M -> K
 N -> K
} " )
```

Now with DAGs

```
par(mfrow = c(2, 2))
drawdag(milkDAG1a, cex = 2, radius = 5)
drawdag(milkDAG1, cex = 2, radius = 5)
drawdag(milkDAG2, cex = 2, radius = 5)
drawdag(milkDAG3, cex = 2, radius = 5)
```



Markov equivalence

```
#output for the first one only
impliedConditionalIndependencies( milkDAG1a)
impliedConditionalIndependencies( milkDAG1 )
impliedConditionalIndependencies( milkDAG2 )
impliedConditionalIndependencies( milkDAG3 )
```

```
## K _||_ N | M
```

Binary categorical predictors

```
#chunk with output
data(Howell1)
d <- Howell1
str(d)
```

```
## 'data.frame':    544 obs. of  4 variables:
##  $ height: num  152 140 137 157 145 ...
##  $ weight: num  47.8 36.5 31.9 53 41.3 ...
##  $ age   : num  63 63 65 41 51 35 32 27 19 54 ...
##  $ male  : int  1 0 0 1 0 1 0 1 0 1 ...
```

How not to do it

$$
\begin{aligned}
h_i &\sim N(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta m_i \\
\alpha &\sim N(178, 20) \\
\beta_m &\sim N(0, 10) \\
\sigma &\sim \text{Unif}(0, 50)
\end{aligned}
$$

α is the average **female** height.

How not to do it

Results in more uncertainty about males

$$\alpha \sim N(178, 20)$$
$$\beta_m \sim N(0, 10)$$

```
mu_female <- rnorm(1e4,178,20)
mu_male <- rnorm(1e4,178,20) + rnorm(1e4,0,10)
mu_malfemDF <- data.frame( mu_female , mu_male )
precis(mu_malfemDF)[,-5]
```

```
##                mean       sd     5.5%    94.5%
## mu_female 178.2234 20.22667 146.1867 210.6365
## mu_male   177.9701 22.38307 142.8147 213.8185
```

Proper way of dealing with binary predictors

```
d$sex <- ifelse( d$male==1 , 2 , 1 )
str( d$sex )
```

```
##  num [1:544] 2 1 1 2 1 2 1 2 1 2 ...
```

```
heightByGender <- quap(
  alist(
    height ~ dnorm( mu , sigma ) ,
    mu <- a[sex] ,
    a[sex] ~ dnorm( 178 , 20 ) ,
    sigma ~ dunif( 0 , 50 )
  ) , data=d )

heightByGenderWrong <- quap(
  alist(
    height ~ dnorm( mu , sigma ) ,
    mu <- a + b * male ,
    a ~ dnorm( 178 , 20 ) ,
    b ~ dnorm( 0 , 10 ) ,
    sigma ~ dunif( 0 , 50 )
  ) , data=d )
```

Proper way of dealing with binary predictors

```
precis( heightByGender , depth=2 )[,-5]
```

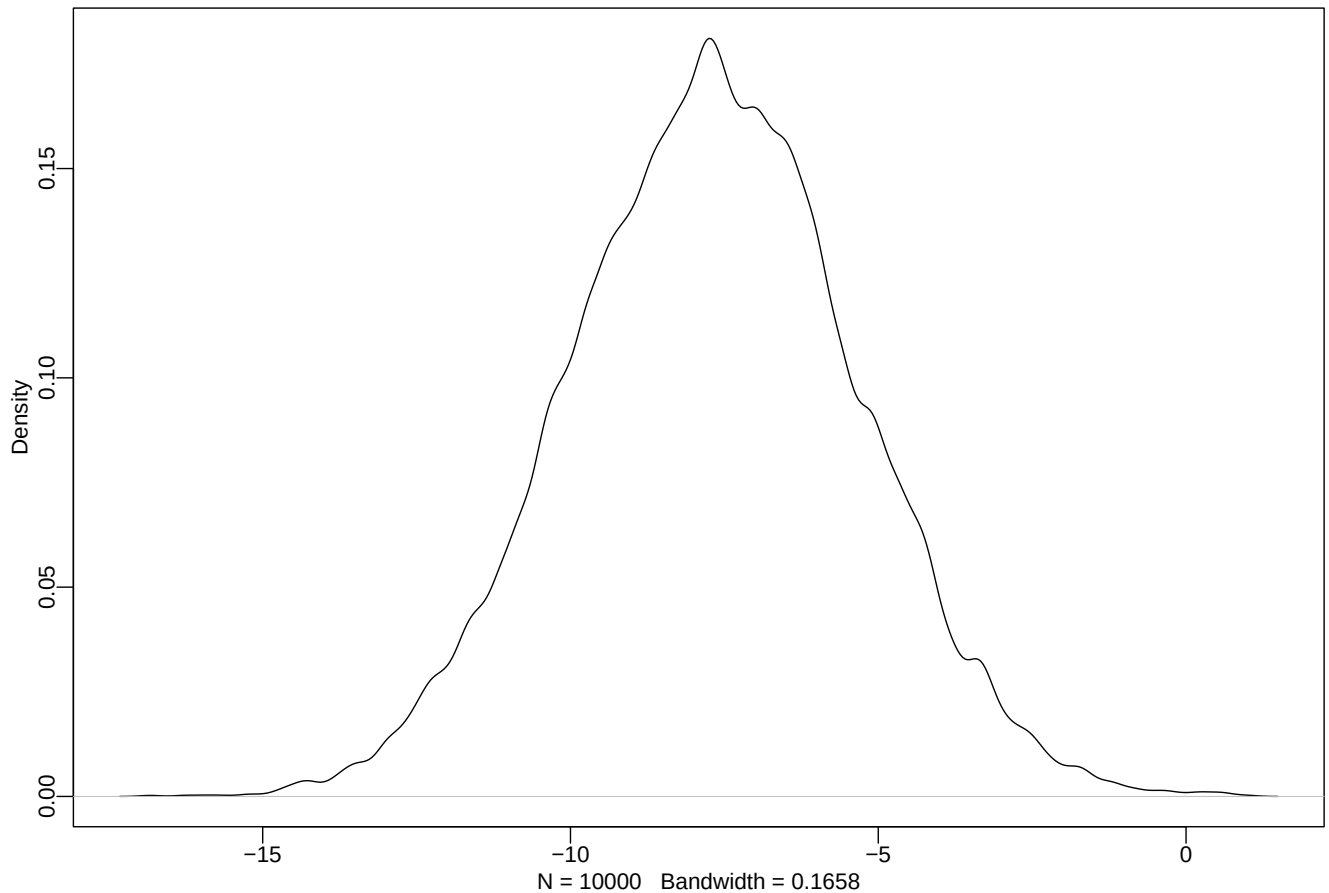
```
##             mean         sd     5.5%     94.5%
## a[1]  134.91020 1.6069069 132.3421 137.47835
## a[2]  142.57824 1.6974451 139.8654 145.29108
## sigma  27.30952 0.8280084  25.9862  28.63283
```

```
precis( heightByGenderWrong , depth=2 )[,-5]
```

```
##             mean         sd       5.5%     94.5%
## a     135.090924 1.587314 132.554091 137.62776
## b       7.026389 2.280473   3.381753  10.67102
## sigma  27.310490 0.828092  25.987039  28.63394
```

Proper way of dealing with binary predictors

```
post <- extract.samples(heightByGender)
post$diff_fm <- post$a[,1] - post$a[,2]
dens ( post$diff_fm  )
```



Proper way of dealing with binary predictors

```
precis( post$diff_fm)[,-5]
```

```
##                      mean       sd      5.5%     94.5%
## post.diff_fm -7.693857 2.324466 -11.44997 -4.033423
```

Multiple predictors

```
data(milk)
m <- milk
unique(m$clade)
```

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

```
m$cladeID <- as.integer( m$clade )
m$K <- standardize( m$kcal.per.g )

str(m)
```

```
## 'data.frame':    29 obs. of  10 variables:
##  $ clade         : Factor w/ 4 levels "Ape","New World Monkey",..: 4 4 4 4 4
##  $ species       : Factor w/ 29 levels "A palliata","Alouatta seniculus",..:
##  $ 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 ...
##  $ cladeID       : int  4 4 4 4 4 2 2 2 2 2 ...
##  $ K             : num  -0.94 -0.816 -1.126 -1.002 -0.259 ...
##   ..- attr(*, "scaled:center")= num 0.642
##   ..- attr(*, "scaled:scale")= num 0.161
```

Multiple predictors

```
dat <- list(k = m$K, cladeID = m$cladeID)

str(dat)
```

```
## List of 2
##  $ k       : num [1:29] -0.94 -0.816 -1.126 -1.002 -0.259 ...
##   ..- attr(*, "scaled:center")= num 0.642
##   ..- attr(*, "scaled:scale")= num 0.161
##  $ cladeID: int [1:29] 4 4 4 4 4 2 2 2 2 2 ...
```

```
caloriesByClade <- quap(
  alist(
    k ~ dnorm( mu , sigma ) ,
    mu <- a[cladeID],
    a[cladeID] ~ dnorm( 0 , .5 ) ,
    sigma ~ dexp( 1 )
  ) , data=dat )
```

Multiple predictors

```
calByClade <- data.frame(precis( caloriesByClade , depth=2 , pars="a" ))
rownames(calByClade) <- paste( "a[" , 1:4 , "]:" , levels(m$clade) , sep="" )
calByClade
```

```
##                              mean        sd       X5.5.      X94.5.
## a[1]:Ape                -0.4843349 0.2176367 -0.83216031 -0.1365094
## a[2]:New World Monkey    0.3662394 0.2170543  0.01934468  0.7131341
## a[3]:Old World Monkey    0.6751813 0.2575302  0.26359832  1.0867643
## a[4]:Strepsirrhine      -0.5858279 0.2745030 -1.02453681 -0.1471191
```

Multiple predictors