waffle

Chris Parrish

June 18, 2016

Contents

waffle

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

Waffle House

```
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.9.0-3, packaged: 2016-02-11 15:54:41 UTC, GitRev: 05c3d0058b6a)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())
```

```
## Loading required package: parallel
```

```
## rethinking (Version 1.58)
```

```
library(ggplot2)
```

data

```
## R code 5.1
# load data
data(WaffleDivorce)
d <- WaffleDivorce
str(d)
```

```
## 'data.frame':    50 obs. of  13 variables:
##  $ Location         : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Loc              : Factor w/ 50 levels "AK","AL","AR",..: 2 1 4 3 5 6 7 9 8 10 ...
##  $ Population       : num  4.78 0.71 6.33 2.92 37.25 ...
##  $ MedianAgeMarriage: num  25.3 25.2 25.8 24.3 26.8 25.7 27.6 26.6 29.7 26.4 ...
##  $ Marriage         : num  20.2 26 20.3 26.4 19.1 23.5 17.1 23.1 17.7 17 ...
##  $ Marriage.SE      : num  1.27 2.93 0.98 1.7 0.39 1.24 1.06 2.89 2.53 0.58 ...
##  $ Divorce          : num  12.7 12.5 10.8 13.5 8 11.6 6.7 8.9 6.3 8.5 ...
##  $ Divorce.SE       : num  0.79 2.05 0.74 1.22 0.24 0.94 0.77 1.39 1.89 0.32 ...
##  $ WaffleHouses     : int  128 0 18 41 0 11 0 3 0 133 ...
##  $ South            : int  1 0 0 1 0 0 0 0 0 1 ...
##  $ Slaves1860       : int  435080 0 0 111115 0 0 0 1798 0 61745 ...
##  $ Population1860   : int  964201 0 0 435450 379994 34277 460147 112216 75080 140424 ...
##  $ PropSlaves1860   : num  0.45 0 0 0.26 0 0 0 0.016 0 0.44 ...
```

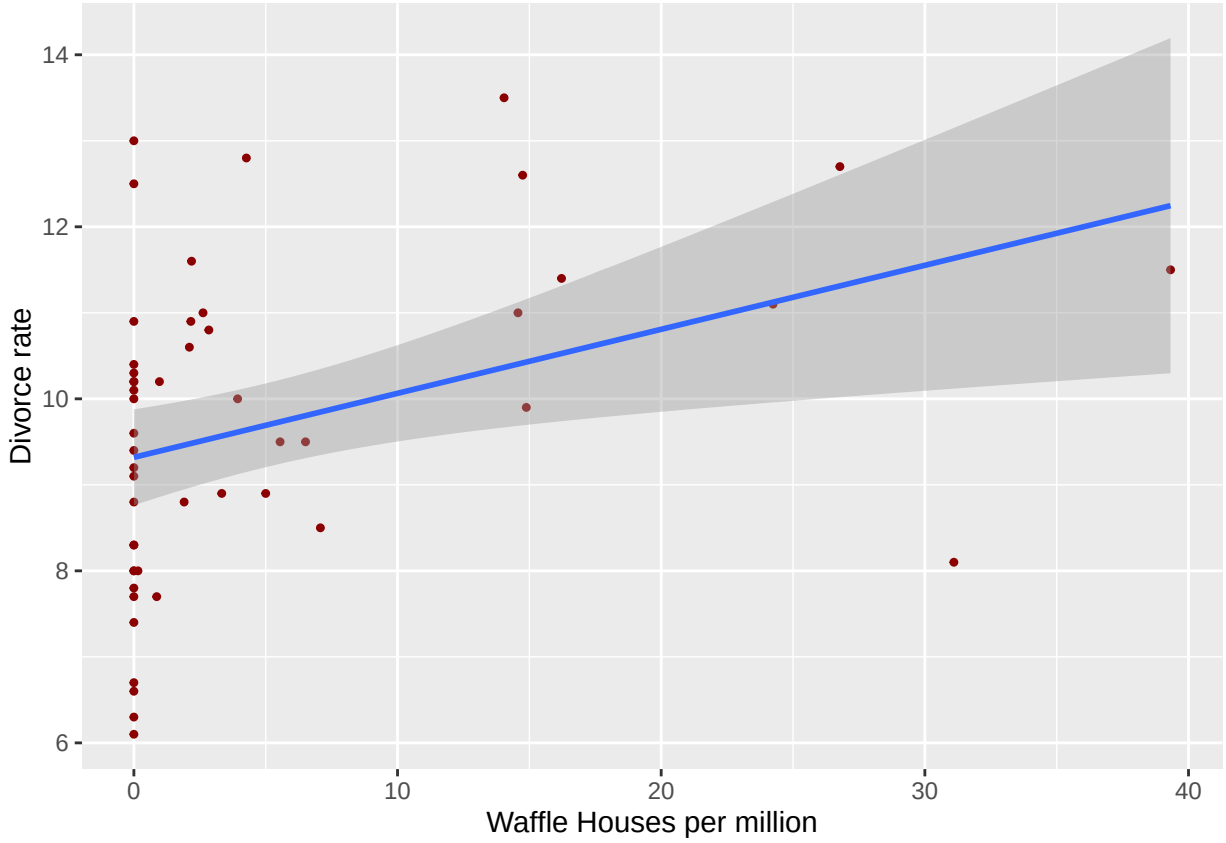
```
head(d)
```

```
##     Location Loc Population MedianAgeMarriage Marriage Marriage.SE Divorce
## 1    Alabama  AL       4.78              25.3     20.2        1.27    12.7
## 2     Alaska  AK       0.71              25.2     26.0        2.93    12.5
## 3    Arizona  AZ       6.33              25.8     20.3        0.98    10.8
## 4   Arkansas  AR       2.92              24.3     26.4        1.70    13.5
## 5 California  CA      37.25              26.8     19.1        0.39     8.0
## 6   Colorado  CO       5.03              25.7     23.5        1.24    11.6
##   Divorce.SE WaffleHouses South Slaves1860 Population1860 PropSlaves1860
## 1       0.79          128     1     435080         964201           0.45
## 2       2.05            0     0          0              0           0.00
## 3       0.74           18     0          0              0           0.00
## 4       1.22           41     1     111115         435450           0.26
## 5       0.24            0     0          0         379994           0.00
## 6       0.94           11     0          0          34277           0.00
```

exploratory data analysis

Waffle Houses

```
ggplot(d, aes(x = WaffleHouses / Population, y = Divorce)) +
  geom_point(shape = 20, color = "darkred") +
  geom_smooth(method = "lm") +
  labs(x = "Waffle Houses per million", y = "Divorce rate")
```

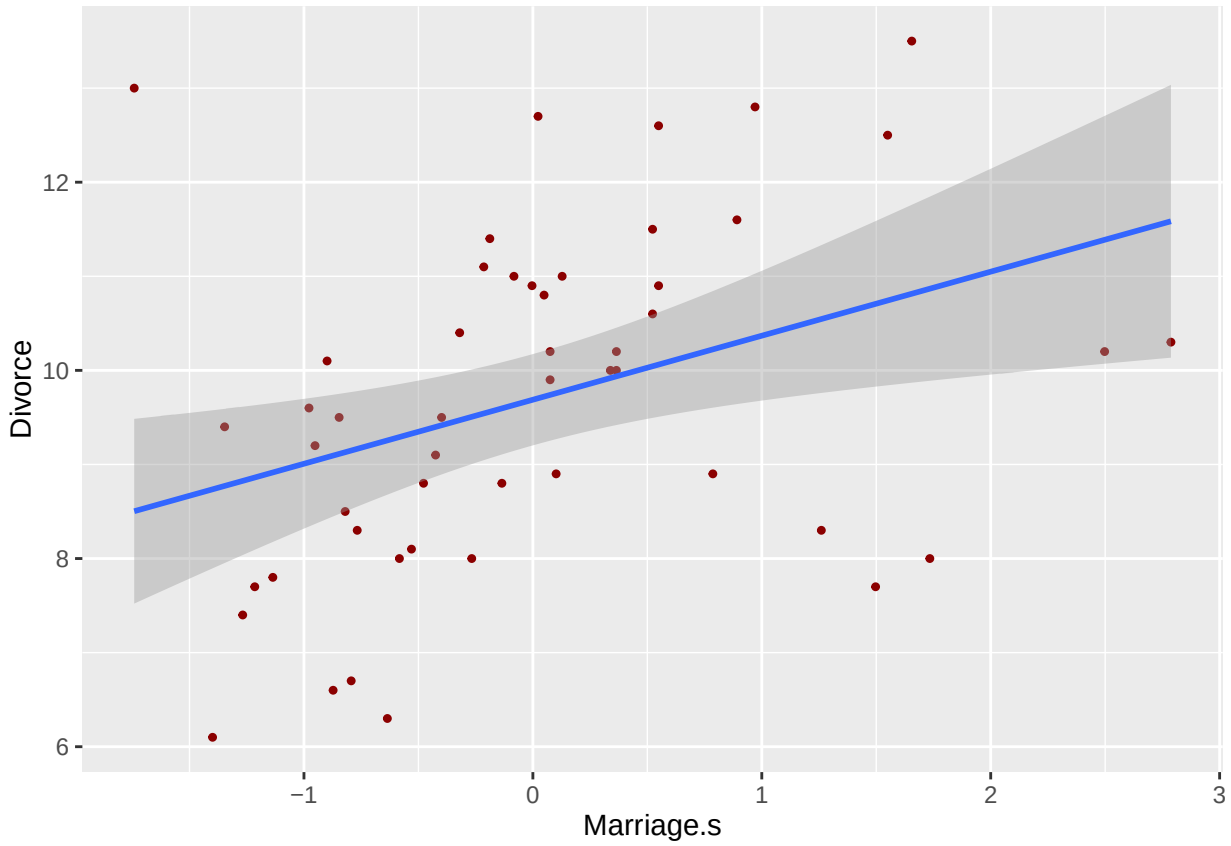


Marriage.s

Standardize marriage.

```
d$Marriage.s <- (d$Marriage - mean(d$Marriage))/sd(d$Marriage)
```

```
ggplot(d, aes(Marriage.s, Divorce)) +
  geom_point(aes(x = Marriage.s, y = Divorce),
             shape = 20, color = "darkred") +
  geom_smooth(method = "lm") +
  labs(x = "Marriage.s", y = "Divorce")
```

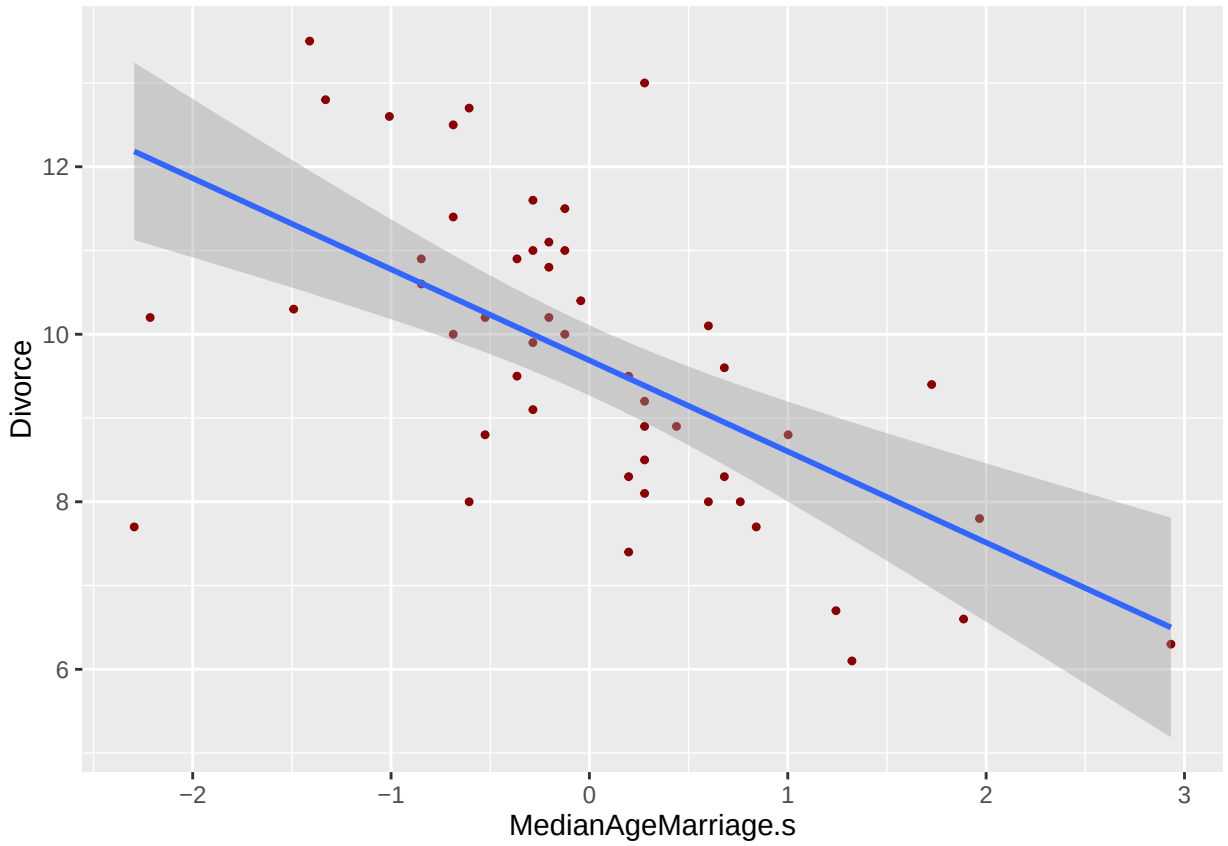


MedianAgeMarriage.s

Standardize median age at marriage.

```
# standardize predictor
d$MedianAgeMarriage.s <- (d$MedianAgeMarriage-mean(d$MedianAgeMarriage))/
    sd(d$MedianAgeMarriage)
```

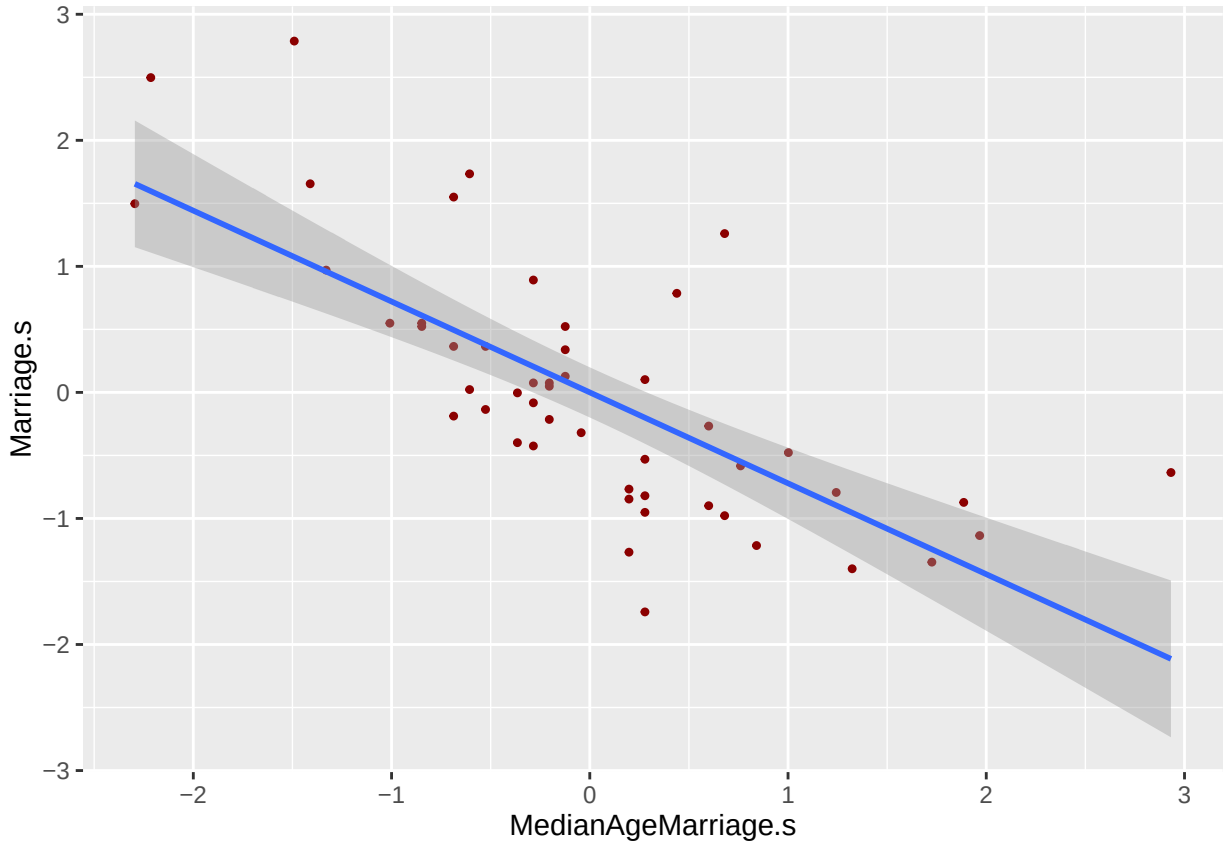
```
ggplot(d, aes(MedianAgeMarriage.s, Divorce)) +
  geom_point(aes(x = MedianAgeMarriage.s, y = Divorce),
             shape = 20, color = "darkred") +
  geom_smooth(method = "lm") +
  labs(x = "MedianAgeMarriage.s", y = "Divorce")
```



MedianAgeMarriage.s **and** *Marriage.s*

How are marriage rate and median age at marriage related?

```
ggplot(d, aes(MedianAgeMarriage.s, Marriage.s)) +
  geom_point(shape = 20, color = "darkred") +
  geom_smooth(method = "lm") +
  labs(x = "MedianAgeMarriage.s", y = "Marriage.s")
```



model divorce rate ~ age at marriage.

$$
\begin{aligned}
D_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_A A_i \\
\alpha &\sim Normal(10, 10) \\
\beta_A &\sim Normal(0, 1) \\
\sigma &\sim Uniform(0, 10)
\end{aligned}
$$

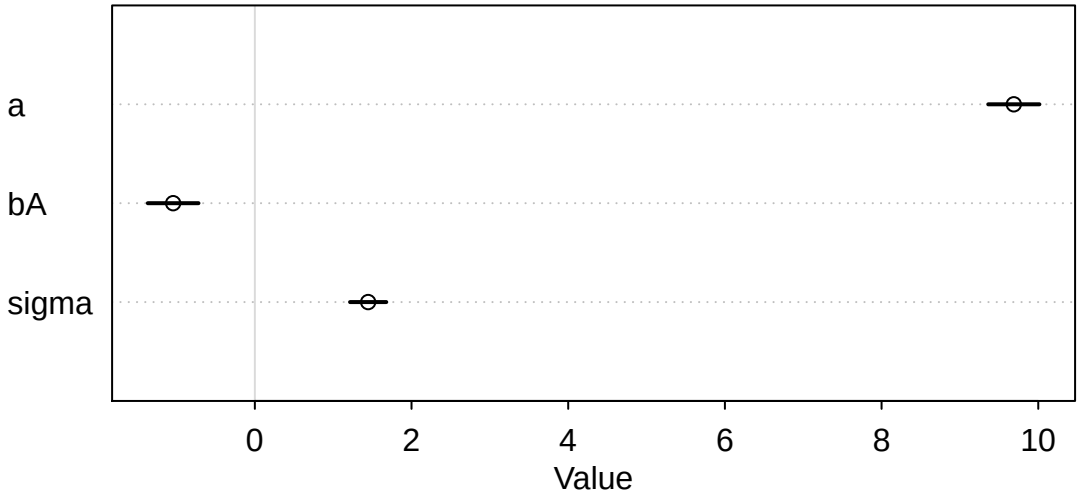
map $m5.1$

```
# fit model
m5.1 <- map(
    alist(
        Divorce ~ dnorm( mu , sigma ) ,
        mu <- a + bA * MedianAgeMarriage.s ,
        a ~ dnorm( 10 , 10 ) ,
        bA ~ dnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 10 )
    ) , data = d )
precis(m5.1)
```

```
##         Mean StdDev  5.5% 94.5%
## a       9.69   0.20  9.36 10.01
```

```
## bA    -1.04   0.20 -1.37 -0.72
## sigma  1.45   0.14  1.22  1.68
```

```
plot( precis(m5.1) )
```



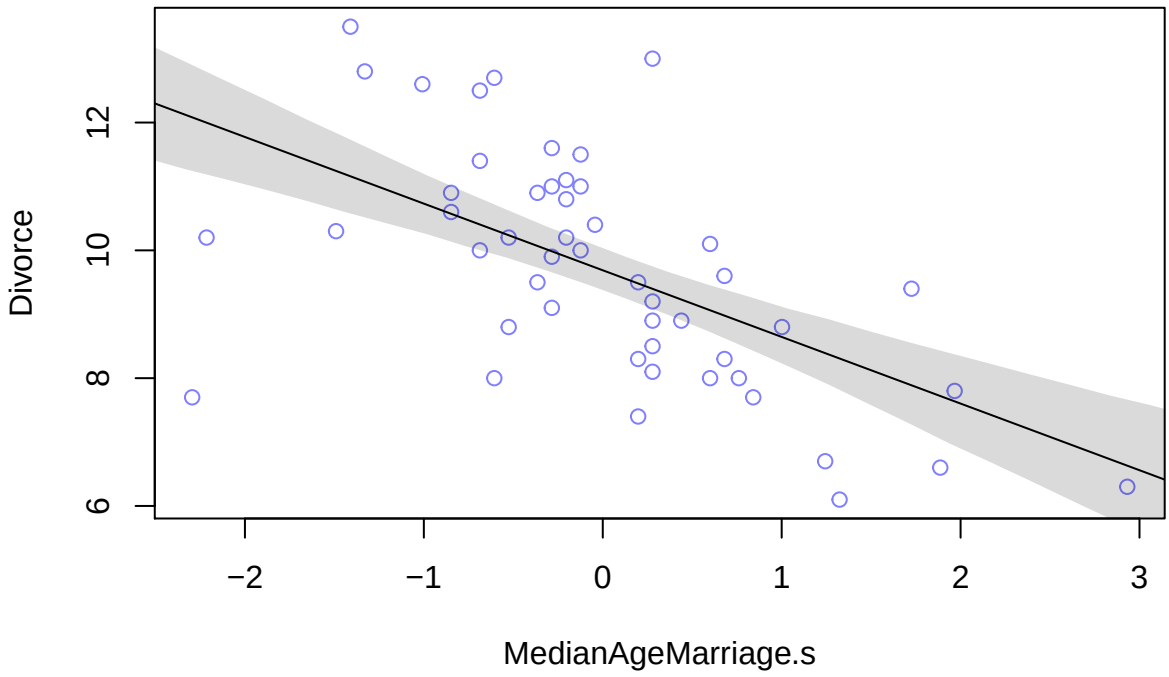

```
## R code 5.2
# compute percentile interval of mean
MAM.seq <- seq( from=-3 , to=3.5 , length.out=30 )
mu <- link( m5.1 , data=data.frame(MedianAgeMarriage.s=MAM.seq) )
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

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

# plot it all
plot( Divorce ~ MedianAgeMarriage.s , data=d , col=rangi2 )
abline( m5.1 )
```

```
## Warning in abline(m5.1): only using the first two of 3 regression
## coefficients
```

```
shade( mu.PI , MAM.seq )
```



model divorce rate ~ marriage rate.

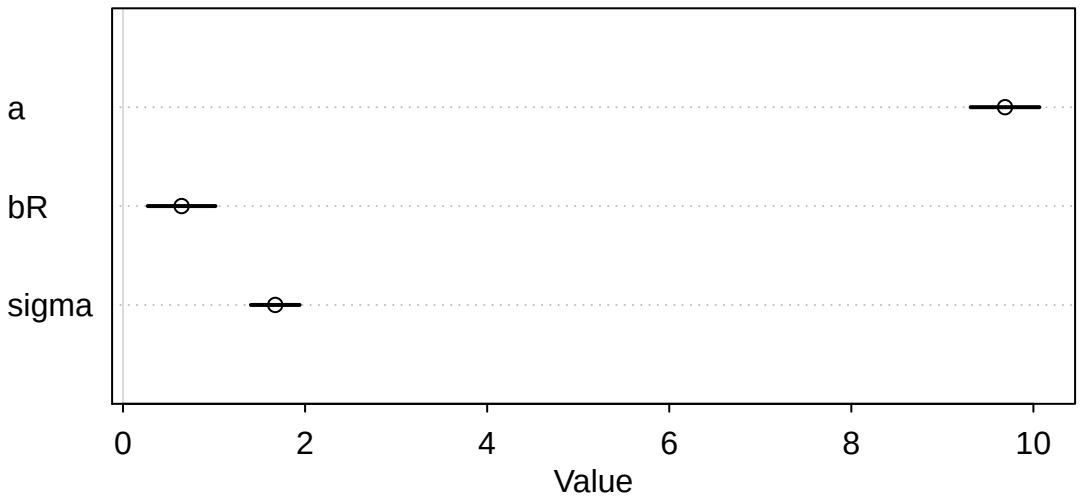
$$
\begin{aligned}
D_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_R R_i \\
\alpha &\sim Normal(10, 10) \\
\beta_R &\sim Normal(0, 1) \\
\sigma &\sim Uniform(0, 10)
\end{aligned}
$$

map $m5.2$

```
## R code 5.3
d$Marriage.s <- (d$Marriage - mean(d$Marriage))/sd(d$Marriage)
m5.2 <- map(
    alist(
        Divorce ~ dnorm( mu , sigma ) ,
        mu <- a + bR * Marriage.s ,
        a ~ dnorm( 10 , 10 ) ,
        bR ~ dnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 10 )
    ) , data = d )
precis(m5.2)
```

```
##       Mean StdDev 5.5% 94.5%
## a     9.69   0.24 9.31 10.07
## bR    0.64   0.23 0.27  1.02
## sigma 1.67   0.17 1.40  1.94
```

```
plot( precis(m5.2) )
```



model divorce rate ~ age at marriage and marriage rate.

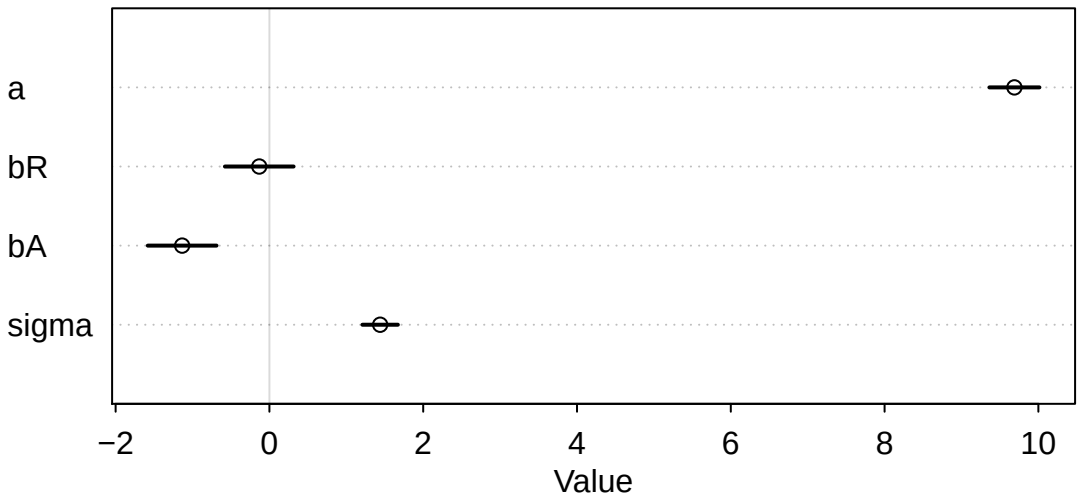
$$
\begin{aligned}
D_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_R R_i + \beta_A A_i \\
\alpha &\sim Normal(10, 10) \\
\beta_R &\sim Normal(0, 1) \\
\beta_A &\sim Normal(0, 1) \\
\sigma &\sim Uniform(0, 10)
\end{aligned}
$$

map $m5.3$

```
## R code 5.4
m5.3 <- map(
    alist(
        Divorce ~ dnorm( mu , sigma ) ,
        mu <- a + bR*Marriage.s + bA*MedianAgeMarriage.s ,
        a ~ dnorm( 10 , 10 ) ,
        bR ~ dnorm( 0 , 1 ) ,
        bA ~ dnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 10 )
    ) ,
    data = d )
precis( m5.3 )
```

```
##         Mean StdDev  5.5% 94.5%
## a       9.69   0.20  9.36 10.01
## bR     -0.13   0.28 -0.58  0.31
## bA     -1.13   0.28 -1.58 -0.69
## sigma   1.44   0.14  1.21  1.67
```

```
## R code 5.5
plot( precis(m5.3) )
```



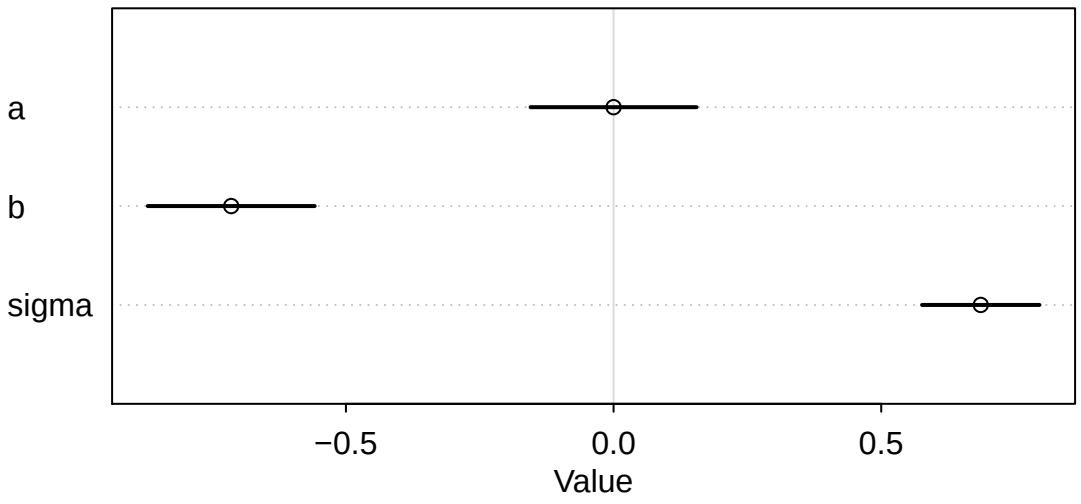
model marriage rate ~ age at marriage

map $m5.4$

```
## R code 5.6
m5.4 <- map(
    alist(
        Marriage.s ~ dnorm( mu , sigma ) ,
        mu <- a + b*MedianAgeMarriage.s ,
        a ~ dnorm( 0 , 10 ) ,
        b ~ dnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 10 )
    ) ,
    data = d )
precis(m5.4)
```

```
##        Mean StdDev  5.5% 94.5%
## a      0.00   0.10 -0.16  0.16
## b     -0.71   0.10 -0.87 -0.56
## sigma  0.69   0.07  0.58  0.80
```

```
plot( precis(m5.4) )
```



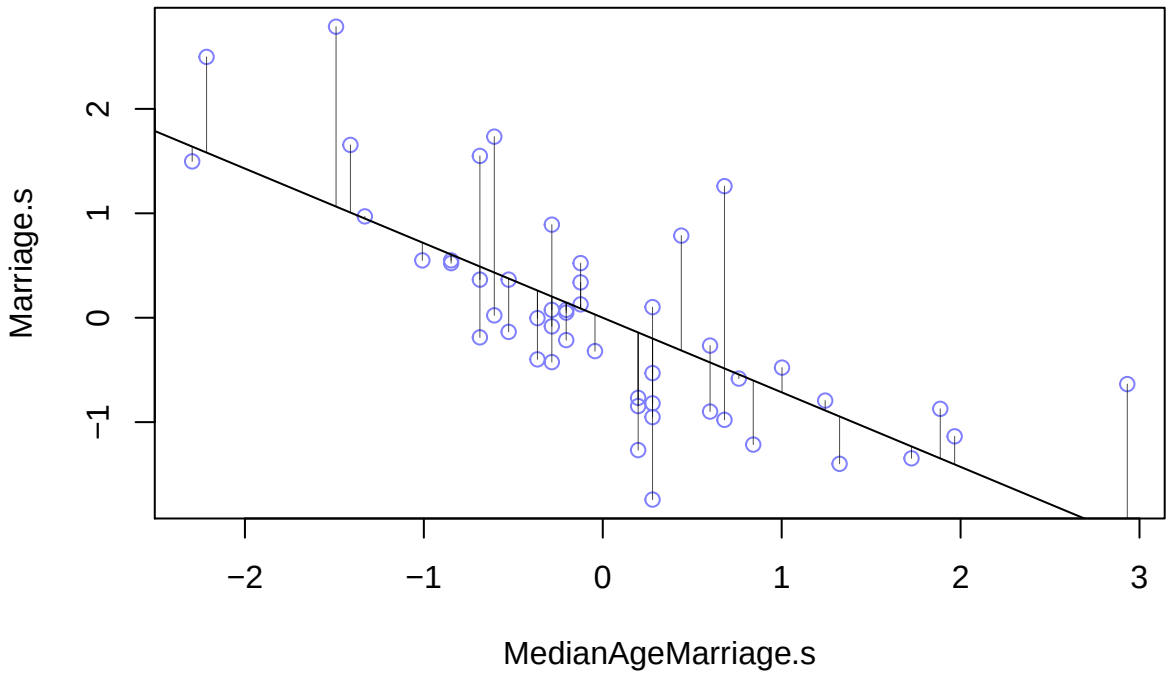
plotting multivariate posteriors

predictor residual plots

```
## R code 5.7
# compute expected value at MAP, for each State
mu <- coef(m5.4)['a'] + coef(m5.4)['b']*d$MedianAgeMarriage.s
# compute residual for each State
m.resid <- d$Marriage.s - mu

## R code 5.8
plot( Marriage.s ~ MedianAgeMarriage.s , d , col=rangi2 )
abline( m5.4 )
```

```
## Warning in abline(m5.4): only using the first two of 3 regression
## coefficients
```

```
# loop over States
for ( i in 1:length(m.resid) ) {
    x <- d$MedianAgeMarriage.s[i] # x location of line segment
    y <- d$Marriage.s[i] # observed endpoint of line segment
    # draw the line segment
    lines( c(x,x) , c(mu[i],y) , lwd=0.5 , col=col.alpha("black",0.7) )
}
```



counterfactual plots

```
## R code 5.9
# prepare new counterfactual data
A.avg <- mean( d$MedianAgeMarriage.s )
R.seq <- seq( from=-3 , to=3 , length.out=30 )
pred.data <- data.frame(
    Marriage.s=R.seq,
    MedianAgeMarriage.s=A.avg
)

# compute counterfactual mean divorce (mu)
mu <- link( m5.3 , data=pred.data )
```

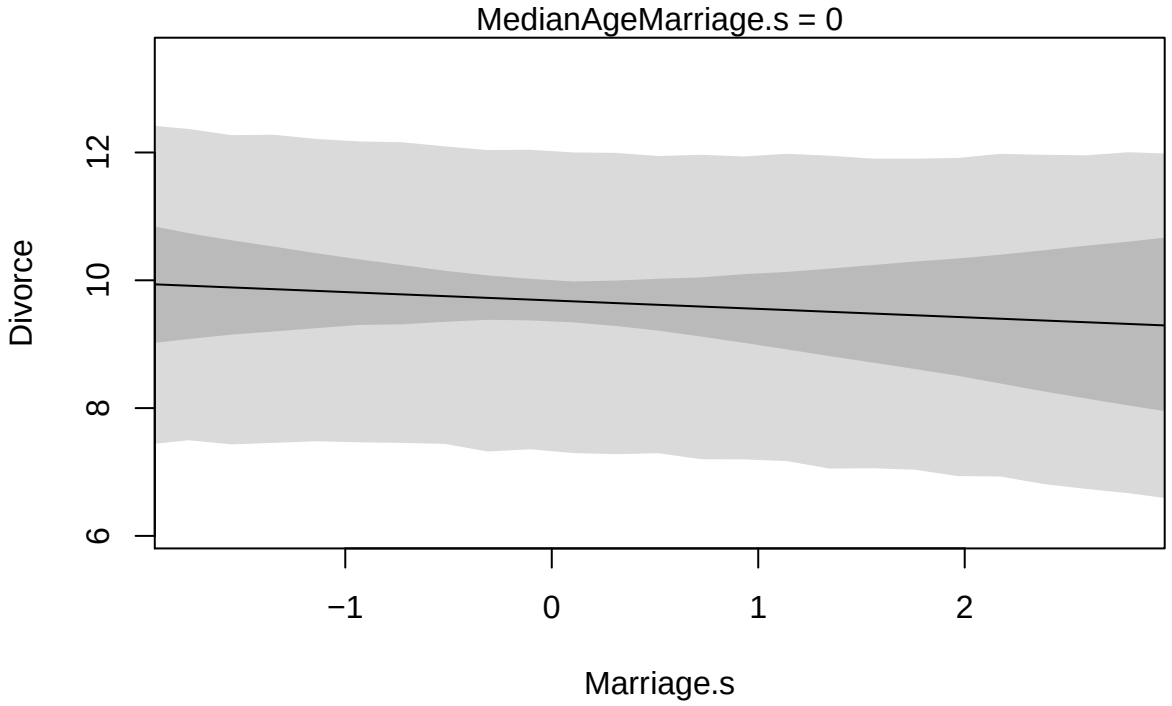
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

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

# simulate counterfactual divorce outcomes
R.sim <- sim( m5.3 , 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 ]
```

```r
R.PI <- apply( R.sim , 2 , PI )

# display predictions, hiding raw data with type="n"
plot( Divorce ~ Marriage.s , data=d , type="n" )
mtext( "MedianAgeMarriage.s = 0" )
lines( R.seq , mu.mean )
shade( mu.PI , R.seq )
shade( R.PI , R.seq )
```




```r
## R code 5.10
R.avg <- mean( d$Marriage.s )
A.seq <- seq( from=-3 , to=3.5 , length.out=30 )
pred.data2 <- data.frame(
    Marriage.s=R.avg,
    MedianAgeMarriage.s=A.seq
)

mu <- link( m5.3 , data=pred.data2 )
```

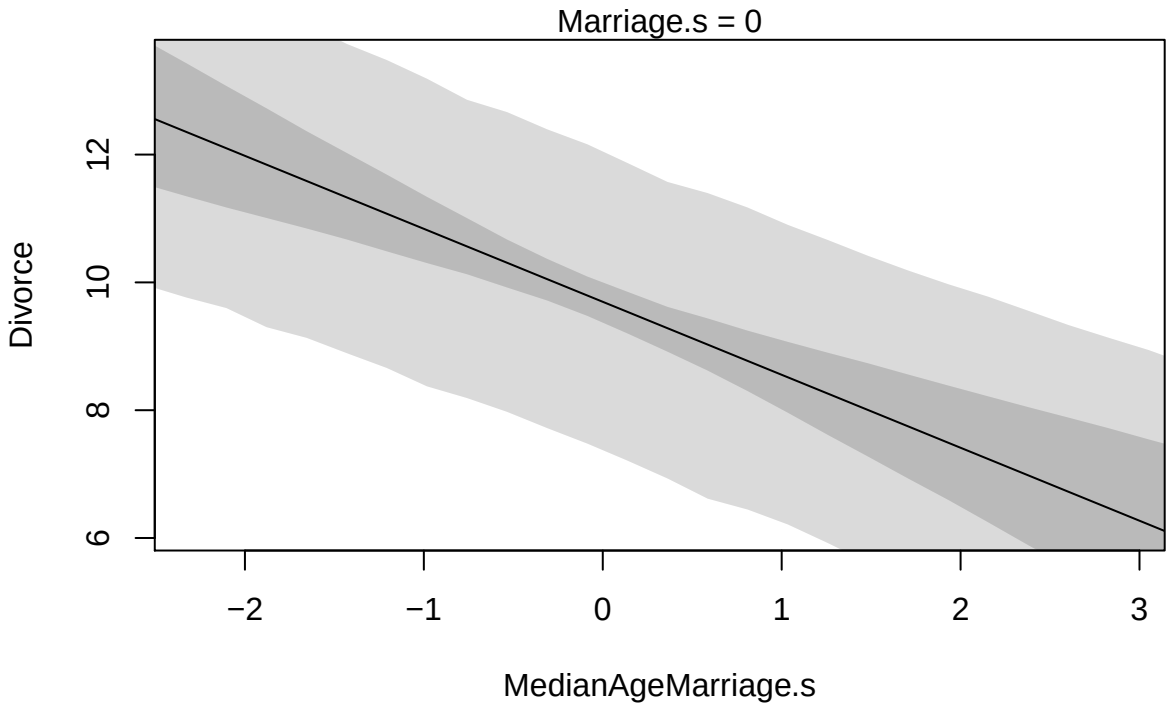
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
```

```
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

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

A.sim <- sim( m5.3 , data=pred.data2 , n=1e4 )
```

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

```
A.PI <- apply( A.sim , 2 , PI )

plot( Divorce ~ MedianAgeMarriage.s , data=d , type="n" )
mtext( "Marriage.s = 0" )
lines( A.seq , mu.mean )
shade( mu.PI , A.seq )
shade( A.PI , A.seq )
```



posterior prediction plots

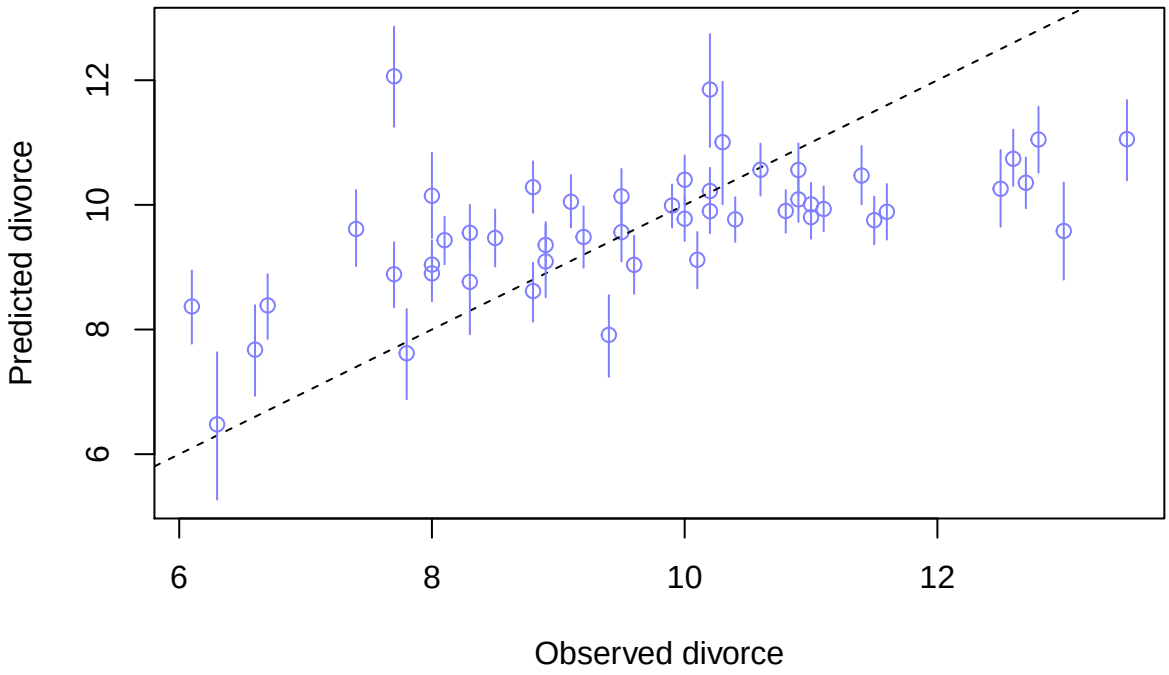
```
## R code 5.11
# call link without specifying new data
# so it uses original data
mu <- link( m5.3 )
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
# summarize samples across cases
mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI )

# simulate observations
# again no new data, so uses original data
divorce.sim <- sim( m5.3 , n=1e4 )
```

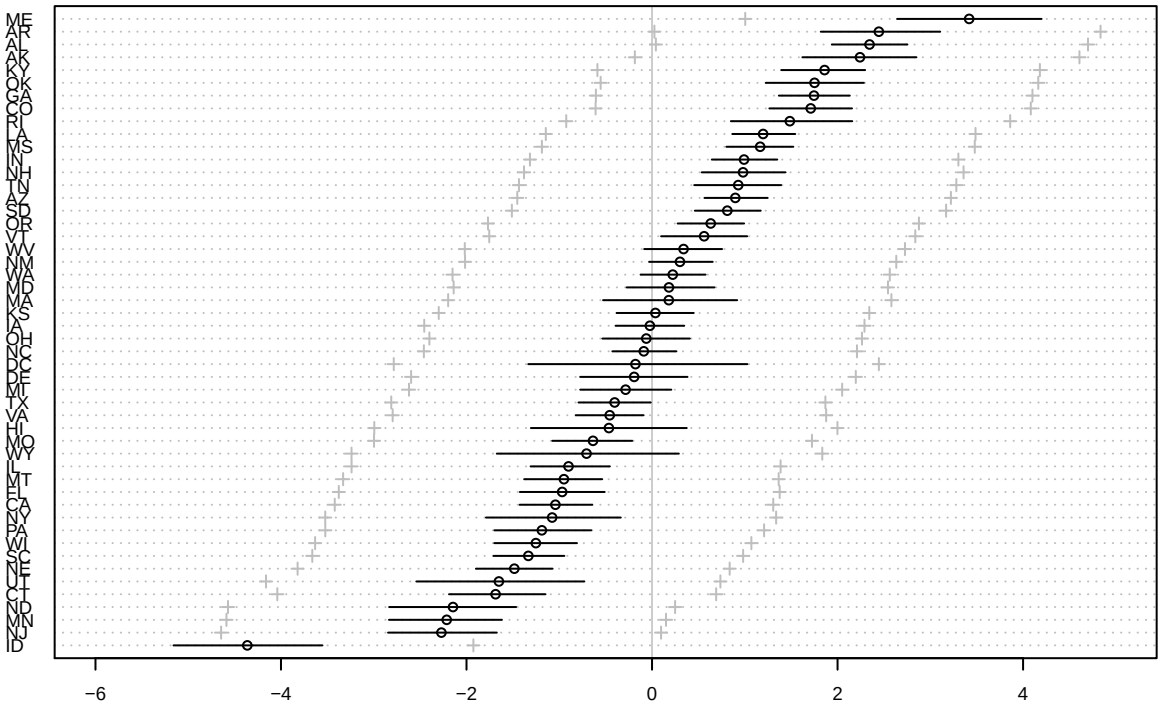
```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```
divorce.PI <- apply( divorce.sim , 2 , PI )

## R code 5.12
plot( mu.mean ~ d$Divorce , col=rangi2 , ylim=range(mu.PI) ,
    xlab="Observed divorce" , ylab="Predicted divorce" )
abline( a=0 , b=1 , lty=2 )
for ( i in 1:nrow(d) )
    lines( rep(d$Divorce[i],2) , c(mu.PI[1,i],mu.PI[2,i]) ,
        col=rangi2 )

## R code 5.13
identify( x=d$Divorce , y=mu.mean , labels=d$Loc , cex=0.8 )
```




```
## integer(0)
```

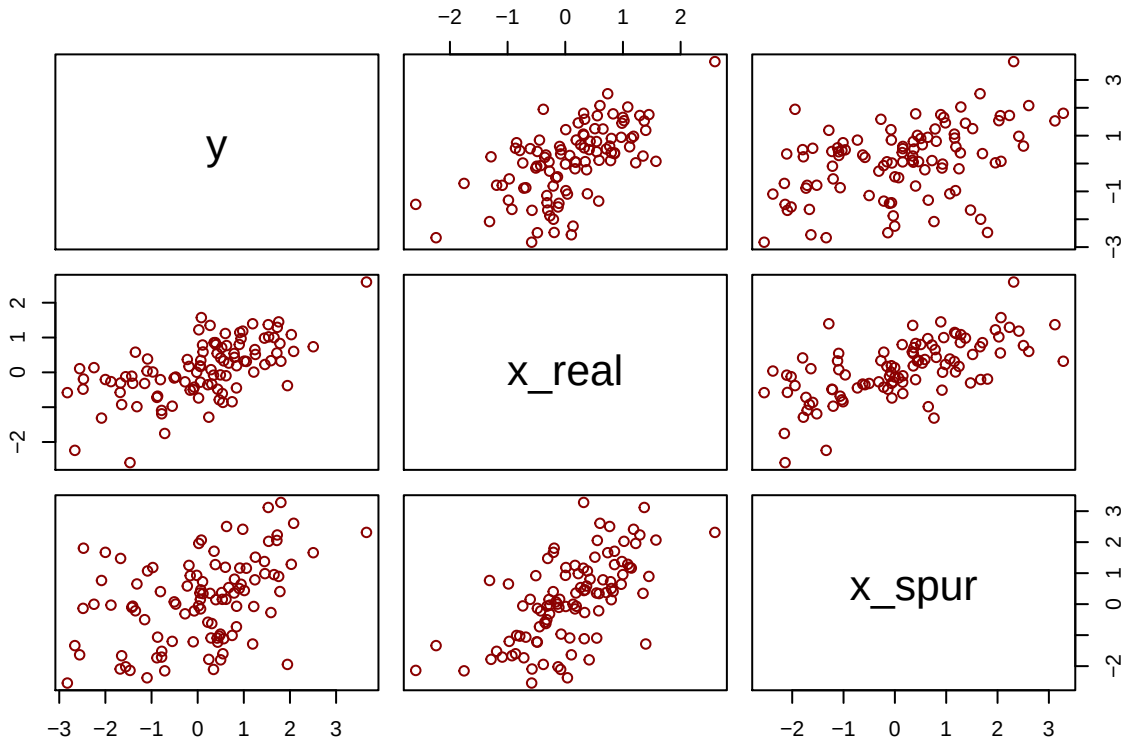
```
## R code 5.14
# compute residuals
divorce.resid <- d$Divorce - mu.mean
# get ordering by divorce rate
o <- order(divorce.resid)
# make the plot
dotchart( divorce.resid[o] , labels=d$Loc[o] , xlim=c(-6,5) , cex=0.6 )
abline( v=0 , col=col.alpha("black",0.2) )
for ( i in 1:nrow(d) ) {
    j <- o[i] # which State in order
    lines( d$Divorce[j]-c(mu.PI[1,j],mu.PI[2,j]) , rep(i,2) )
    points( d$Divorce[j]-c(divorce.PI[1,j],divorce.PI[2,j]) , rep(i,2),
        pch=3 , cex=0.6 , col="gray" )
}
```



simulating spurious association

```
## R code 5.15
N <- 100                          # number of cases
x_real <- rnorm( N )              # x_real as Gaussian with mean 0 and stddev 1
x_spur <- rnorm( N , x_real )     # x_spur as Gaussian with mean=x_real
y <- rnorm( N , x_real )          # y as Gaussian with mean=x_real
d <- data.frame(y,x_real,x_spur) # bind all together in data frame
```

```
pairs(~ y + x_real + x_spur, data=d,
      col="darkred")
```




```
demo.lm <- lm(y ~ x_real + x_spur, data=d)
options(show.signif.stars=FALSE)
summary(demo.lm)
```

```
##
## Call:
## lm(formula = y ~ x_real + x_spur, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5484 -0.5838  0.1744  0.6885  2.3940
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.003954   0.103260  -0.038    0.970
## x_real       0.864603   0.159877   5.408 4.58e-07
## x_spur       0.059178   0.098300   0.602    0.549
##
## Residual standard error: 1.025 on 97 degrees of freedom
## Multiple R-squared:  0.368,  Adjusted R-squared:  0.355
## F-statistic: 28.24 on 2 and 97 DF,  p-value: 2.158e-10
```