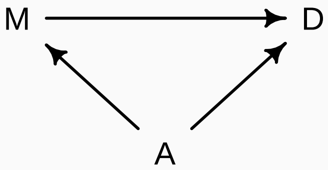
Multiple regression 2

Frank Edwards

2/21/2020

What does it mean to condition on a variable?

Let's return to the divorce model with this DAG



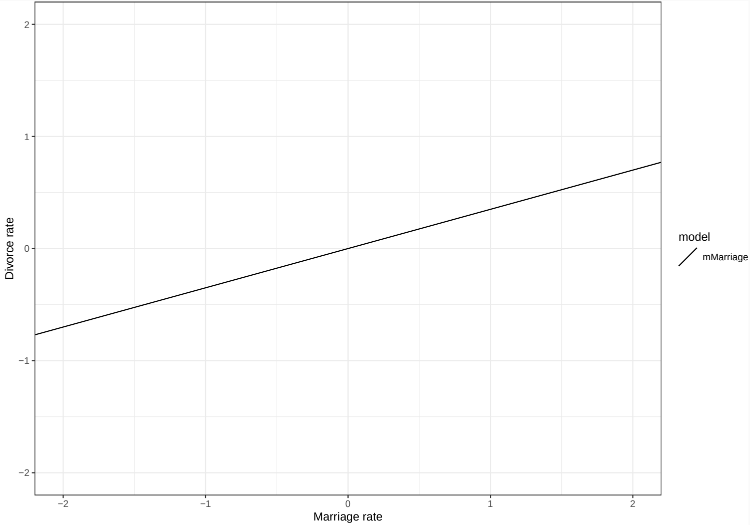
We want to know how much we learn about divorce rates by knowing another variable if:

- We already know marriage rates
- We already know the median age at first marriage

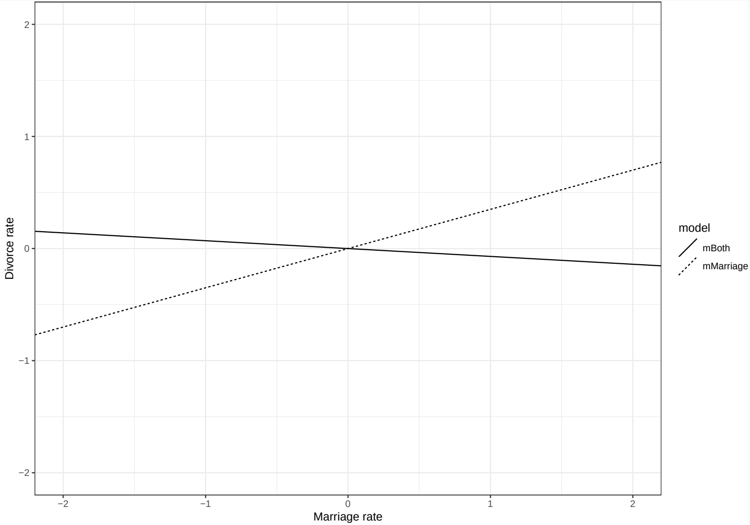
The single and multiple regressions

```
mAge<-quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu<-a + bA * A,
    a ~ dnorm(0, 0.2),
    bA ~ dnorm(0, 0.5),
    sigma ~ dexp(1)),
  data = WaffleDivorce
)
mMarriage<-quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu<-a + bM * M,
    a ~ dnorm(0, 0.2),
    bM ~ dnorm(0, 0.5),
    sigma ~ dexp(1)),
  data = WaffleDivorce
)
mBoth<-quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu<-a + bA * A + bM * M,
    a ~ dnorm(0, 0.2),
    bA ~ dnorm(0, 0.5),
    bM ~ dnorm(0, 0.5),
    sigma ~ dexp(1)),
  data = WaffleDivorce
)
```

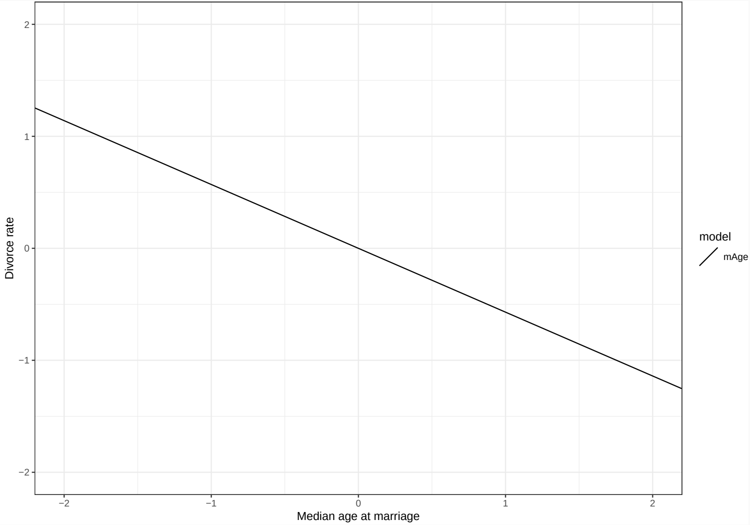
The relationship between divorce and marriage



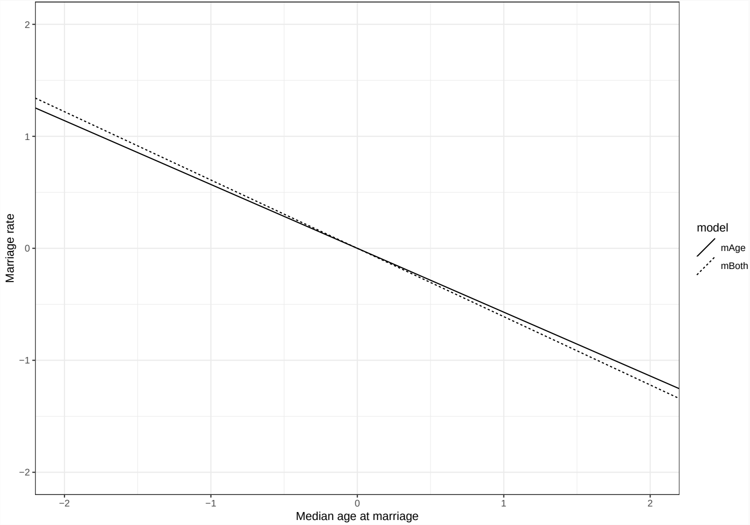
The relationship between divorce and marriage



The relationship between age and marriage



The relationship between age and marriage



Regressions as conditional expectation engines

- mMarriage tells us $\mathrm{E}(D|M)$
- mBoth tells us $\mathrm{E}(D|A, M)$

Regressions as conditional expectation engines

- mMarriage tells us $\mathrm{E}(D|M)$
- mBoth tells us $\mathrm{E}(D|A, M)$
- mAge tells us $\mathrm{E}(D|A)$
- mBoth tells us $\mathrm{E}(D|A, M)$

Regressions as conditional expectation engines

- mMarriage tells us $\mathrm{E}(D|M)$
- mBoth tells us $\mathrm{E}(D|A, M)$
- mAge tells us $\mathrm{E}(D|A)$
- mBoth tells us $\mathrm{E}(D|A, M)$
- Once we know the median age at first marriage, the marriage rate provides little additional information about divorce rates.

Regressions as conditional expectation engines

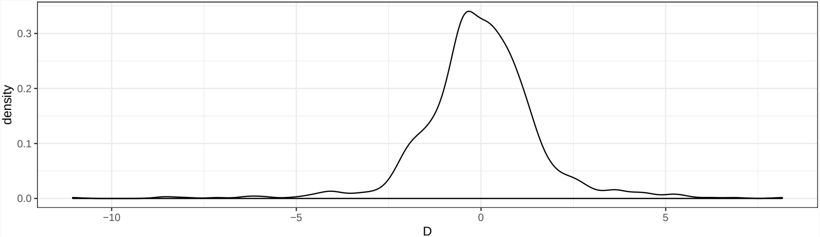
- mMarriage tells us $\mathrm{E}(D|M)$
- mBoth tells us $\mathrm{E}(D|A, M)$
- mAge tells us $\mathrm{E}(D|A)$
- mBoth tells us $\mathrm{E}(D|A, M)$
- Once we know the median age at first marriage, the marriage rate provides little additional information about divorce rates.
- The association between marriage rates and divorce rates is *spurious*, driven by the underlying $D \leftarrow A \rightarrow M$ relationship

Visualizing models with more than one predictor

1. **Prior prediction plots**: What are plausible datasets we could observe?
2. Posterior prediction plots: Does the model fit the observed data?
3. Counterfactual plots: What is the effect of X on Y?

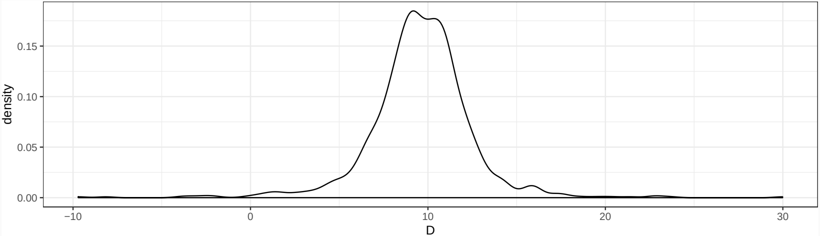
Prior prediction on the standardized scale for one value of the predictors

```r
p<-extract.prior(mBoth)
### make into a data frame
p<-bind_cols(p)
### simulate from 1 SD below the mean for A, M
A<- -1
M<- -1
p<-p %>%
  mutate(mu = a + bA * A + bM * M, sigma,
         D = rnorm(1000, mu, sigma))

ggplot(p, aes(x = D)) +
  geom_density()
```

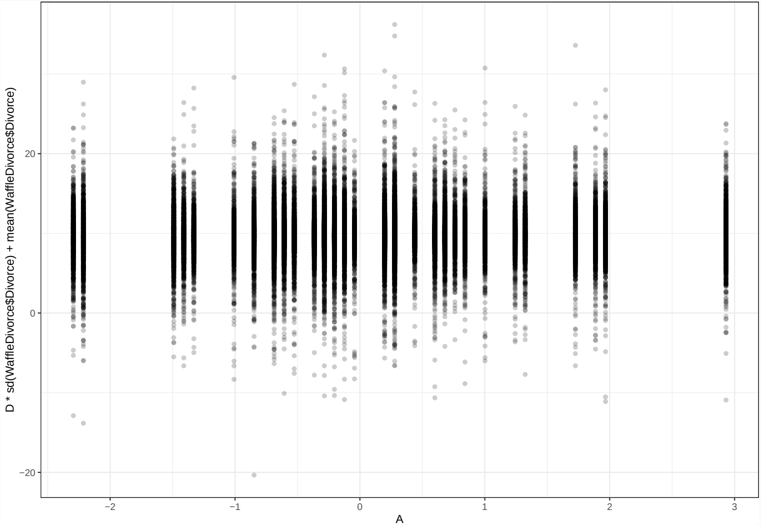


Prior prediction on the original scale for one value of the predictors

```
p<-p %>%
  mutate(mu = a + bA * A + bM * M, sigma,
         D = rnorm(1000, mu, sigma),
         D = D * sd(WaffleDivorce$Divorce) + mean(WaffleDivorce$Divorce))

ggplot(p, aes(x = D)) +
  geom_density()
```



Prior prediction for all observed values

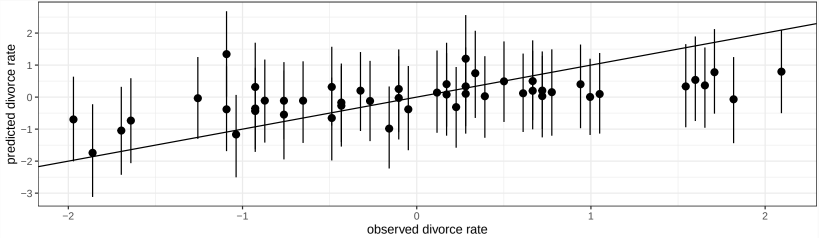


Visualizing models with more than one predictor

1. Prior prediction plots: What are plausible datasets we could observe?
2. **Posterior prediction plots**: Does the model fit the observed data?
3. Counterfactual plots: What is the effect of X on Y?

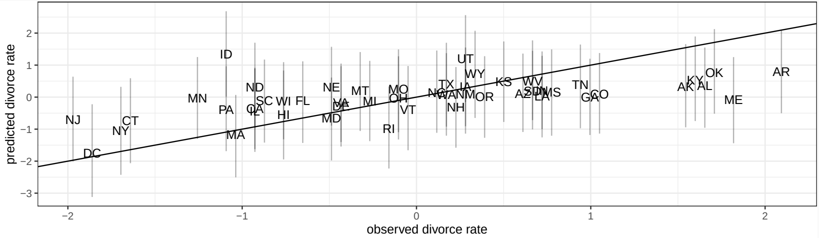
Posterior prediction plot

```
d_post<-sim(mBoth)
mu_post<-apply(d_post, 2, mean)
pi_d_post<-apply(d_post, 2, PI)
plot_dat<-data.frame(mu = mu_post,
                     d_upr = pi_d_post[2,],
                     d_lwr = pi_d_post[1,],
                     d_obs = WaffleDivorce$D,
                     state = WaffleDivorce$Loc)

ggplot(plot_dat, aes(x = d_obs,
                     y = mu,
                     ymin = d_lwr,
                     ymax = d_upr)) +
  geom_pointrange() +
  geom_abline(intercept = 0, slope = 1) +
  xlab(”observed divorce rate”) +
  ylab(”predicted divorce rate”)
```



Posterior prediction plot with labels

```
ggplot(plot_dat, aes(x = d_obs,
                     y = mu,
                     ymin = d_lwr,
                     ymax = d_upr,
                     label = state)) +
  geom_linerange(alpha = 0.3) +
  geom_text() +
  geom_abline(intercept = 0, slope = 1)+
  xlab("observed divorce rate") +
  ylab("predicted divorce rate")
```



Visualizing models with more than one predictor

1. Prior prediction plots: What are plausible datasets we could observe?
2. Posterior prediction plots: Does the model fit the observed data?
3. **Counterfactual plots**: What is the effect of X on Y?

Counterfactual plots

Algorithm:

1. Pick a variable to manipulate
2. Choose a range or set of values for that predictor
3. Simulate each of the other variables in the model for each value of the predictor and each posterior sample

What is the impact of age at first marriage on divorce?

1. Pick a variable to manipulate: Age
2. Choose a range for that predictor: [-2, 2]

```r
## scenarios: age varies from -2 to +2 SD of observed
a_cfact<-seq(-2, 2, length.out = 100)
```

What is the impact of age at first marriage on divorce?

1. Pick a variable to manipulate: Age
2. Choose a range for that predictor: [-2, 2]

```r
## scenarios: age varies from -2 to +2 SD of observed
a_cfact<-seq(-2, 2, length.out = 100)
```

3. Simulate each of the other variables in the model for each value of the predictor

What is the impact of age at first marriage on divorce?

1. Pick a variable to manipulate: Age
2. Choose a range for that predictor: [-2, 2]

```r
## scenarios: age varies from -2 to +2 SD of observed
a_cfact<-seq(-2, 2, length.out = 100)
```

3. Simulate each of the other variables in the model for each value of the predictor

What is the impact of age at first marriage on divorce?

1. Pick a variable to manipulate: Age
2. Choose a range for that predictor: [-2, 2]

```
## scenarios: age varies from -2 to +2 SD of observed
a_cfact<-seq(-2, 2, length.out = 100)
```

3. Simulate each of the other variables in the model for each value of the predictor

Recall that our DAG thinks that $A \longrightarrow D$, $A \longrightarrow M$, and $M \longrightarrow D$. If we want to understand what will happen with A changes, we need to allow M to move as A moves.

Simulate other variables in the model

Model the relationship between A and M on D, and the relationship between A and M

```r
mBothCFact<-quap(
  alist(
    ## A -> D <- M
    D ~ dnorm(mu, sigma),
    mu<-a + bA * A + bM * M,
    a ~ dnorm(0, 0.2),
    bA ~ dnorm(0, 0.5),
    bM ~ dnorm(0, 0.5),
    sigma ~ dexp(1),
    ## A -> M
    M ~ dnorm(mu_m, sigma_m),
    mu_m <- a_m + bAm * A,
    a_m ~ dnorm(0, 0.2),
    bAm ~ dnorm(0, 0.5),
    sigma_m ~ dexp(1)),
  data = WaffleDivorce)
```

Simulate other variables in the model

Our predictions should account for the expected changes in M when we counterfactually “manipulate” A

```
summary(mBothCFact)
```

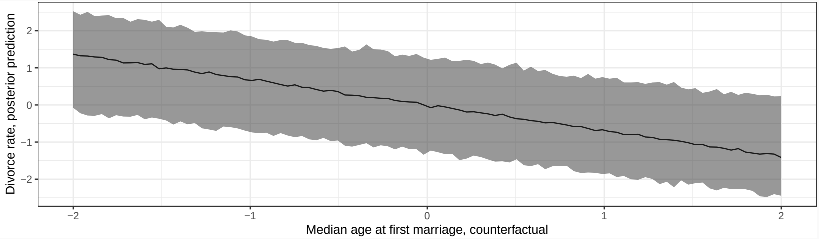
```
##                    mean         sd       5.5%      94.5%
## a       -5.396811e-09 0.09707604 -0.1551463  0.1551463
## bA      -6.135133e-01 0.15098361 -0.8548143 -0.3722123
## bM      -6.538064e-02 0.15077308 -0.3063451  0.1755839
## sigma    7.851181e-01 0.07784342  0.6607093  0.9095269
## a_m      1.841448e-08 0.08684782 -0.1387996  0.1387996
## bAm     -6.947376e-01 0.09572691 -0.8477277 -0.5417475
## sigma_m  6.817367e-01 0.06758002  0.5737308  0.7897427
```

Generate predictions

```r
sim_dat<-data.frame(A = seq(-2, 2, length.out = 100))
## A is fixed, simulate M then D from posterior
D_cfact<-sim(mBothCFact, data = sim_dat, vars = c(”M”, ”D”))
## compute mean and PI for both M and D predictions
mu_M <-apply(D_cfact$M, 2, mean)
mu_D <- apply(D_cfact$M, 2, mean)
pi_M <-apply(D_cfact$M, 2, PI)
pi_D <- apply(D_cfact$D, 2, PI)
### put it all together for plotting
sim_dat<-sim_dat %>%
  mutate(mu_M = mu_M,
         mu_D = mu_D,
         D_lwr = pi_D[1,],
         D_upr = pi_D[2,],
         M_lwr = pi_M[1,],
         M_upr = pi_M[2,])
```

Plot expected changes in D for changes in A

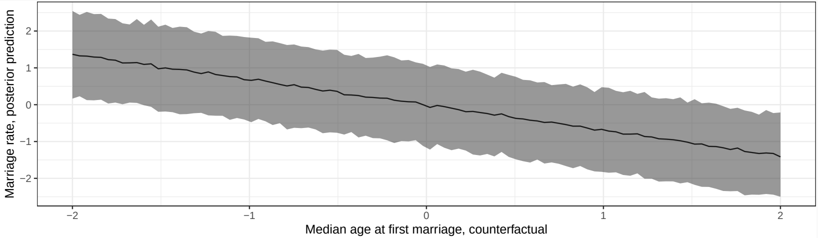
```r
ggplot(sim_dat,
       aes(x = A,
           y = mu_D,
           ymin = D_lwr,
           ymax = D_upr)) +
  geom_line() +
  geom_ribbon(alpha = 0.5) +
  ylab("Divorce rate, posterior prediction") +
  xlab("Median age at first marriage, counterfactual")
```



Plot expected changes in M for changes in A (for illustration)

Recall that A is negatively associated with M, but once we condition on A, M is not clearly associated with D. We allow A to change M, and M *and* A to change D in this approach

```
ggplot(sim_dat,
       aes(x = A,
           y = mu_M,
           ymin = M_lwr,
           ymax = M_upr)) +
  geom_line() +
  geom_ribbon(alpha = 0.5) +
  ylab("Marriage rate, posterior prediction") +
  xlab("Median age at first marriage, counterfactual")
```



Categorical variables

Kinds of categorical variables

- Binary [T,F]

Kinds of categorical variables

- Binary [T,F]
- Qualitative differences

Kinds of categorical variables

- Binary [T,F]
- Qualitative differences
- Ranked qualitative differences

In R

Categorical variables typically take on three formats:

- factor (a linked pair of integers and labels, can be finicky)
- character (label only vectors)
- integers (where labels are implicit)

In R

Categorical variables typically take on three formats:

- factor (a linked pair of integers and labels, can be finicky)
- character (label only vectors)
- integers (where labels are implicit)

I typically prefer to work with character vectors (as.character()), but there are cases where each approach has advantages

Returning to the height data

```r
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 ...
```

```r
d<-d %>%
  mutate(h = as.vector(scale(height)),
         w = as.vector(scale(weight)))
```

Returning to the height data

```
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 ...
```

```
d<-d %>%
  mutate(h = as.vector(scale(height)),
         w = as.vector(scale(weight)))
```

any categoricals?

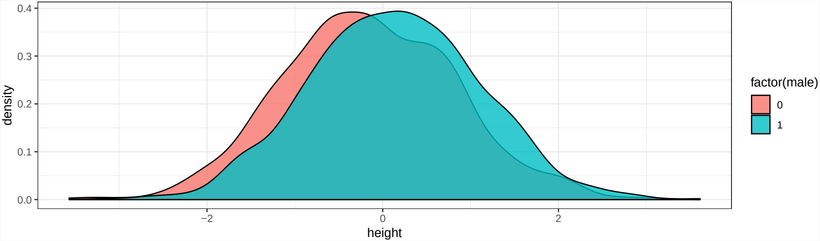
Estimating a model with sex as a predictor

```
m_h1<-quap(alist(
  h ~ dnorm(mu, sigma),
  mu <- a + b * male,
  a ~ dnorm(0, 1),
  b ~ dnorm(0, 1),
  sigma ~ dexp(1)),
  data = d)

summary(m_h1)
```

```
##             mean         sd       5.5%       94.5%
## a     -0.1302467 0.05814938 -0.2231806 -0.03731275
## b      0.2761924 0.08451697  0.1411180  0.41126683
## sigma  0.9884536 0.02992610  0.9406260  1.03628133
```

What does this model suggest?

```
new_data<-data.frame(male = c(0,1))
sim_post<-sim(m_h1, data = new_data)
plot_dat<-data.frame(height = c(sim_post[,1], sim_post[,2]),
                     male = rep(c(0, 1), each = 1000))

ggplot(plot_dat, aes(x = height, fill = factor(male))) +
  geom_density(alpha = 0.8)
```



The linear model explained

$$E(height_i) = \alpha + \beta \times Male_i$$

The linear model explained

$$E(height_i) = \alpha + \beta \times Male_i$$

For males:

$$E(height_{male}) = \alpha + \beta \times 1$$

The linear model explained

$$E(height_i) = \alpha + \beta \times Male_i$$

For males:

$$E(height_{male}) = \alpha + \beta \times 1$$

For females:

$$E(height_{female}) = \alpha + \beta \times 0$$

The linear model explained

$$E(height_i) = \alpha + \beta \times Male_i$$

For males:

$$E(height_{male}) = \alpha + \beta \times 1$$

For females:

$$E(height_{female}) = \alpha + \beta \times 0$$

The intercept α then becomes the expected height for females, and $\alpha + \beta$ is the expected height for males.

An alternative parameterization

```r
### set sex = 1 for F, = 2 for M
d<-d %>%
  mutate(sex = case_when(male == 0 ~ 1,
                         male == 1 ~ 2))

m_h2 <- quap(alist(
  h ~ dnorm(mu , sigma),
  mu <- a[sex],
  a[sex] ~ dnorm(0 ,1),
  sigma ~ dexp(1)),
  data=d)

precis(m_h2, depth = 2)
```

```
##              mean         sd        5.5%      94.5%
## a[1]  -0.1311841 0.05824742 -0.22427476 -0.0380935
## a[2]   0.1464392 0.06154108  0.04808464  0.2447937
## sigma  0.9884520 0.02992596  0.94062453  1.0362795
```

Extending to multiple category variables

```
data(iris)
head(iris)
```

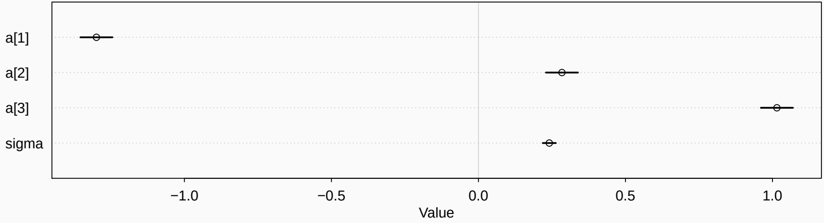
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

What is the expected petal length for each species?

```
iris<- iris %>%
  mutate(p.l = as.vector(scale(Petal.Length)))

m_iris<-quap(alist(
  p.l <- dnorm(mu, sigma),
  mu<- a[Species],
  a[Species] ~ dnorm(0, 1),
  sigma ~ dexp(1)),
  data = iris)

levels(iris$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

```
plot(precis(m_iris, depth = 2))
```



Masked relationships

What is the relationship between milk nutrients and brain size in primates?

You thought we were studying criminal justice?

```r
data(milk)
d <- milk
d<-d %>%
  mutate(K = as.vector(scale(kcal.per.g)),
         N = as.vector(scale(neocortex.perc)),
         M = as.vector(scale(log(mass)))) %>%
  filter(!(is.na(K) | is.na(N) | is.na(M))) # remove missings
glimpse(d)
```

```
## Observations: 17
## Variables: 11
## $ clade          <fct> Strepsirrhine, New World Monkey, New World Monkey, N...
## $ species        <fct> Eulemur fulvus, Alouatta seniculus, A palliata, Cebu...
## $ kcal.per.g     <dbl> 0.49, 0.47, 0.56, 0.89, 0.92, 0.80, 0.46, 0.71, 0.68...
## $ perc.fat       <dbl> 16.60, 21.22, 29.66, 53.41, 50.58, 41.35, 3.93, 38.3...
## $ perc.protein   <dbl> 15.42, 23.58, 23.46, 15.80, 22.33, 20.85, 25.30, 20....
## $ perc.lactose   <dbl> 67.98, 55.20, 46.88, 30.79, 27.09, 37.80, 70.77, 41....
## $ mass           <dbl> 1.95, 5.25, 5.37, 2.51, 0.68, 0.12, 0.47, 0.32, 1.55...
## $ neocortex.perc <dbl> 55.16, 64.54, 64.54, 67.64, 68.85, 58.85, 61.69, 60....
## $ K              <dbl> -0.9400408, -1.0639553, -0.5063402, 1.5382486, 1.724...
## $ N              <dbl> -2.080196025, -0.508641289, -0.508641289, 0.01074247...
## $ M              <dbl> -0.4558357, 0.1274408, 0.1407505, -0.3071581, -1.076...
```

Is milk nutrient density related to percent of the brain that is neocortex? to body weight?

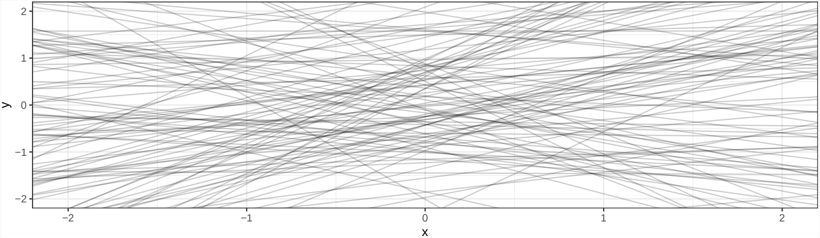
Two initial models, one for brain composition, and one for body mass:

$$
\begin{aligned}
K_i &\sim N(\mu, \sigma) \\
\mu_i &= \alpha + \beta_N N_i \\
\alpha &\sim N(0, 1) \\
\beta_N &\sim N(0, 1) \\
\sigma &\sim Exp(1)
\end{aligned}
$$

$$
\begin{aligned}
K_i &\sim N(\mu, \sigma) \\
\mu_i &= \alpha + \beta_M M_i \\
\alpha &\sim N(0, 1) \\
\beta_M &\sim N(0, 1) \\
\sigma &\sim Exp(1)
\end{aligned}
$$

Check the priors

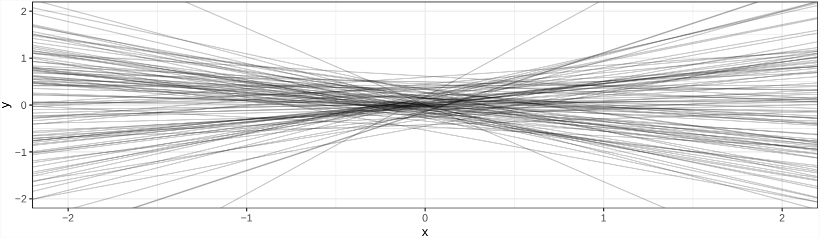
```
plot_dat<-data.frame(a=rnorm(100, 0, 1),
                     bn = rnorm(100, 0 , 1),
                     x = seq(-2, 2, length.out = 100),
                     y = seq(-2, 2, length.out = 100))

ggplot(plot_dat, aes(x,y)) +
  geom_blank() +
  geom_abline(aes(intercept = a, slope = bn), alpha = 0.2)
```



Maybe rein those in a bit?

Try $\alpha \sim N(0, 0.2)$, $\beta_N \sim N(0, 0.5)$

```
plot_dat<-data.frame(a=rnorm(100, 0, 0.2),
                     bn = rnorm(100, 0 , 0.5),
                     x = seq(-2, 2, length.out = 100),
                     y = seq(-2, 2, length.out = 100))

ggplot(plot_dat, aes(x,y)) +
  geom_blank() +
  geom_abline(aes(intercept = a, slope = bn), alpha = 0.2)
```



Fit the models

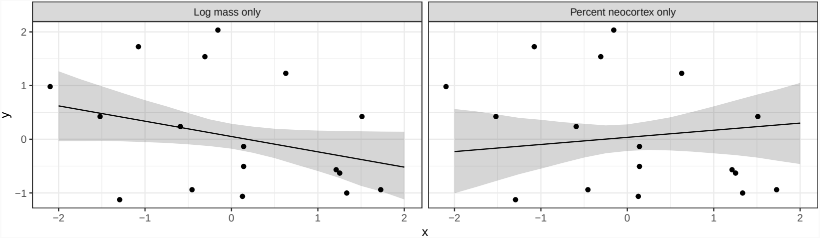
```
mN<-quap(alist(
  K ~ dnorm(mu, sigma),
  mu <- a + bn * N,
  a ~ dnorm(0, 0.2),
  bn ~ dnorm(0, 0.5),
  sigma ~ dexp(1)
), data = d)

mM<-quap(alist(
  K ~ dnorm(mu, sigma),
  mu <- a + bm * M,
  a ~ dnorm(0, 0.2),
  bm ~ dnorm(0, 0.5),
  sigma ~ dexp(1)
), data = d)
```

Format for visualization

```
### draw from the posterior for both models: M
sim_seq<-seq(-2, 2, length.out=17)
mu_m<-link(mM, data = list(M = sim_seq))
mu_m_mn<-apply(mu_m, 2, mean)
mu_m_pi<-apply(mu_m, 2, PI)
### N
mu_n<-link(mN, data = list(N = sim_seq))
mu_n_mn<-apply(mu_n, 2, mean)
mu_n_pi<-apply(mu_n, 2, PI)
### format for plotting, -2,2 sequence for M, N, mu and PI, along with observed data. Stack for facets
plot_dat<-data.frame(x = sim_seq,
                     y = mu_m_mn,
                     obs_x = d$M,
                     obs_y = d$K,
                     y_upr = mu_m_pi[2,],
                     y_lwr = mu_m_pi[1,],
                     model = ”Log mass only”)
plot_dat<-plot_dat%>%
  bind_rows(data.frame(x = sim_seq,
                       y = mu_n_mn,
                       obs_x = d$M,
                       obs_y = d$K,
                       y_upr = mu_n_pi[2,],
                       y_lwr = mu_n_pi[1,],
                       model = ”Percent neocortex only”))
```

Plot it

```
ggplot(plot_dat, aes(x = x, y = y, ymin = y_lwr, ymax = y_upr)) +
  geom_line() +
  geom_ribbon(alpha = 0.2) +
  geom_point(aes(x = obs_x, y = obs_y)) +
  facet_wrap(~model)
```



Masking

```
d %>%
  summarise(K_N = cor(K, N),
            N_M = cor(N, M),
            K_M = cor(K, M))
```

```
##         K_N       N_M        K_M
## 1 0.1554576 0.7503758 -0.3542636
```

Masking

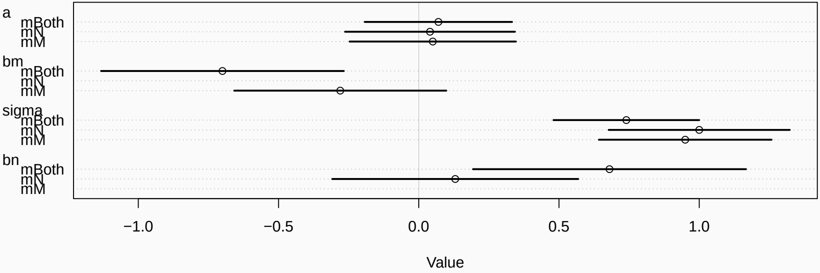
```
d %>%
  summarise(K_N = cor(K, N),
            N_M = cor(N, M),
            K_M = cor(K, M))
```

```
##         K_N       N_M        K_M
## 1 0.1554576 0.7503758 -0.3542636
```

When two predictor variables are correlated with each other, and have opposite sign correlations with the outcome, excluding one can *mask* an underlying relationship.

Fit a model with both mass and percent neocortex

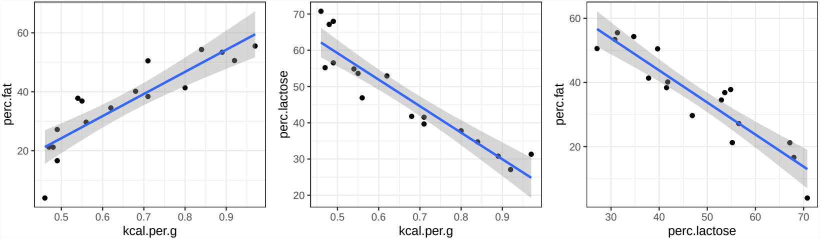
```
mBoth<-quap(alist(
  K ~ dnorm(mu, sigma),
  mu <- a + bn * N + bm * M,
  a ~ dnorm(0, 0.2),
  bn ~ dnorm(0, 0.5),
  bm ~ dnorm(0, 0.5),
  sigma ~ dexp(1)
), data = d)

plot(coeftab(mM, mN, mBoth))
```



Multicollinearity

- Sometimes, adding more predictors *unmasks* relationships
- Sometimes, adding more predictors *masks* relationships

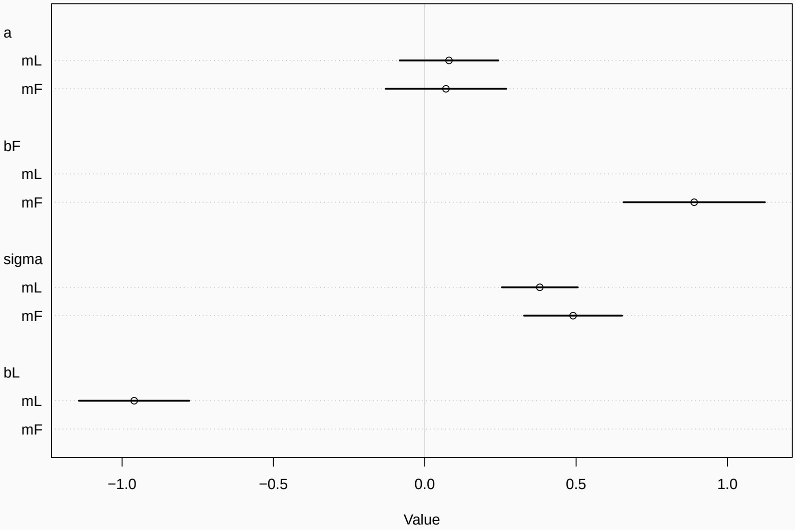
```
library(gridExtra)
p1<-ggplot(d, aes(x = kcal.per.g, y = perc.fat)) + geom_point() + geom_smooth(method = ”lm”)
p2<-ggplot(d, aes(x = kcal.per.g, y = perc.lactose)) + geom_point() + geom_smooth(method = ”lm”)
p3<-ggplot(d, aes(x = perc.lactose, y = perc.fat)) + geom_point() + geom_smooth(method = ”lm”)
grid.arrange(p1, p2, p3, ncol = 3)
```



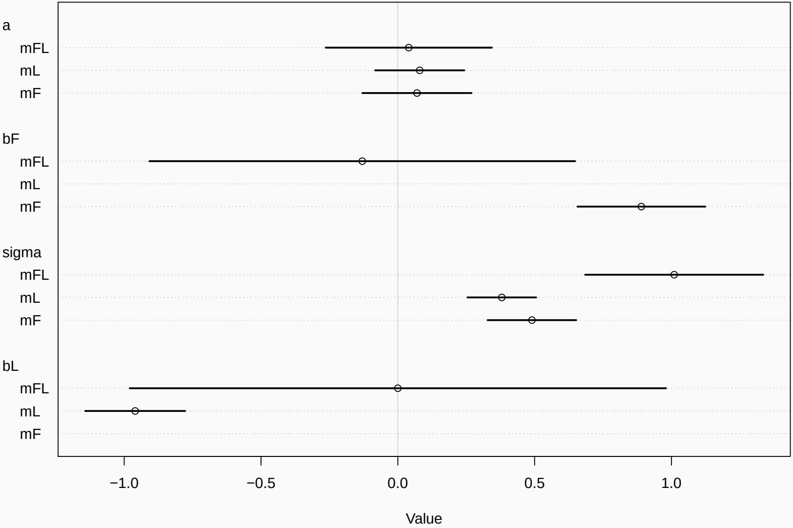
What happens when we put strongly correlated variables in a model?

```r
d<-d %>% mutate(F = as.vector(scale(perc.fat)), L = as.vector(scale(perc.lactose)))
mF<-quap(alist(
    K ~ dnorm(mu, sigma),
    mu <- a + bF * F,
    a ~ dnorm(0, 0.2),
    bF ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d)

mL<-quap(alist(
    K ~ dnorm(mu, sigma),
    mu <- a + bL * L,
    a ~ dnorm(0, 0.2),
    bL ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d)
```

Results from one variable regressions



With both variables as predictors


a
mFL
mL
mF
bF
mFL
mL
mF
sigma
mFL
mL
mF
bL
mFL
mL
mF
Value