Linear regression with multiple predictors (multiple regression)

Frank Edwards

2/14/2020

The linear and stochastic components of the model

Understanding a linear model with one predictor

Let's say that wages are related to years of education according to this linear model:

$$wages_i \sim Normal(\mu, \sigma)$$

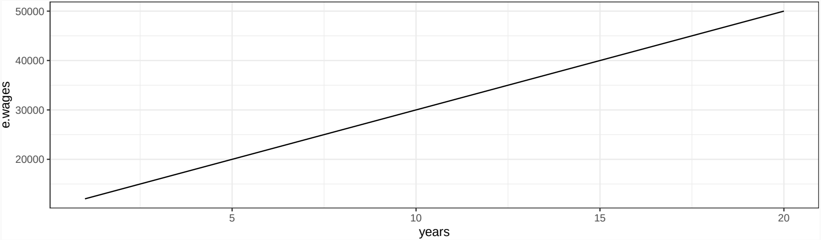
$$E(wages_i) = \mu_i = 10000 + 2000 \times years_i$$

The linear component of the model

$$E(wages_i) = \mu_i = 10000 + 2000 \times years_i$$

This defines the expected value of wages, conditional on knowing someone's years of education.

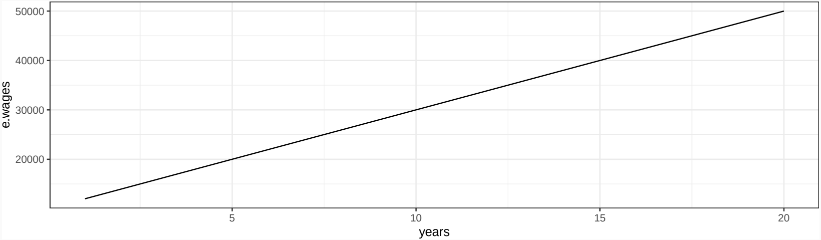
```
### make sequence of years to describe wages over
plot_dat<-data.frame(years = seq(1, 20, length.out = 100))
### apply the linear component to estimate expected wages
plot_dat<-plot_dat %>%
  mutate(e.wages = 10000 + 2000 * years)
ggplot(plot_dat, aes(x = years, y = e.wages)) +
  geom_line()
```



The stochastic component of the model

This is not a deterministic (perfectly accurate) prediction. It is just accurate on average. The stochastic component of the model describes (sd = σ) how different people are from the expected value (μ_i).

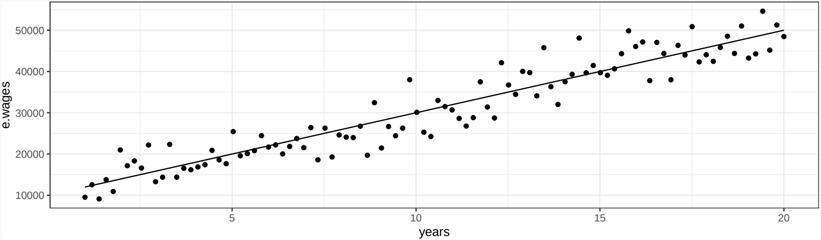
```r
plot_dat<-plot_dat %>%
  mutate(sim_wages = rnorm(n = 100, mean = plot_dat$e.wages, sd = 4000))
### sigma = 4000
ggplot(plot_dat, aes(x = years, y = e.wages)) +
  geom_line()
```



The stochastic component of the model

This is not a deterministic (perfectly accurate) prediction. It is just accurate on average. The stochastic component of the model describes how different people are from the expected value (μ_i).

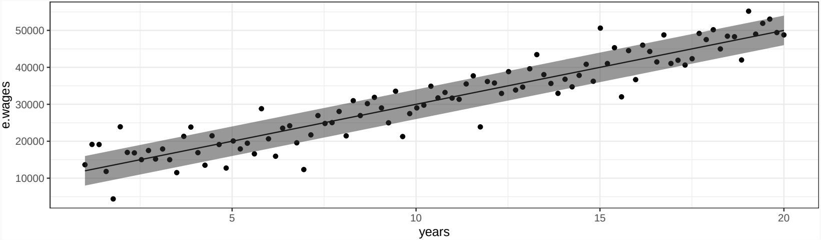
```r
plot_dat<-plot_dat %>%
  mutate(sim_wages = rnorm(n = 100, mean = plot_dat$e.wages, sd = 4000))
### sigma = 4000
ggplot(plot_dat, aes(x = years, y = e.wages)) +
  geom_line() +
  geom_point(aes(y = sim_wages))
```



The stochastic component of the model

This is not a deterministic (perfectly accurate) prediction. It is just accurate on average. The stochastic component of the model describes how different people are from the expected value (μ_i).

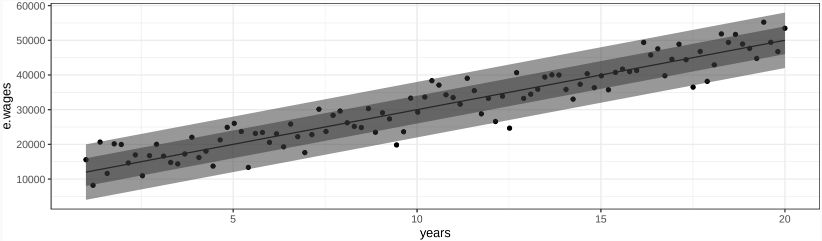
```
plot_dat<-plot_dat %>%
  mutate(sim_wages = rnorm(n = 100, mean = plot_dat$e.wages, sd = 4000))
### sigma = 4000
ggplot(plot_dat, aes(x = years, y = e.wages)) +
  geom_line() +
  geom_point(aes(y = sim_wages)) +
  geom_ribbon(aes(ymax = e.wages + 4000, ymin =e.wages - 4000), alpha = 0.5) # 1 SD
```



The stochastic component of the model

This is not a deterministic (perfectly accurate) prediction. It is just accurate on average. The stochastic component of the model describes how different people are from the expected value (μ_i).

```r
plot_dat<-plot_dat %>%
  mutate(sim_wages = rnorm(n = 100, mean = plot_dat$e.wages, sd = 4000))
### sigma = 4000
ggplot(plot_dat, aes(x = years, y = e.wages)) +
  geom_line() +
  geom_point(aes(y = sim_wages)) +
  geom_ribbon(aes(ymax = e.wages + 4000, ymin =e.wages - 4000), alpha = 0.5) + # 1 SD
  geom_ribbon(aes(ymax = e.wages + 2*4000, ymin =e.wages - 2*4000), alpha = 0.5) ## 2 SD
```



Returning to the height model

The anatomy of a linear model

Likelihood: $h_i \sim Normal(\mu_i, \sigma)$

Linear model: $\mu_i = \alpha + \beta x_i$

Prior: $\alpha \sim Normal(150, 25)$

Prior: $\beta \sim Uniform(0, 5)$

Prior: $\sigma \sim Uniform(0, 10)$

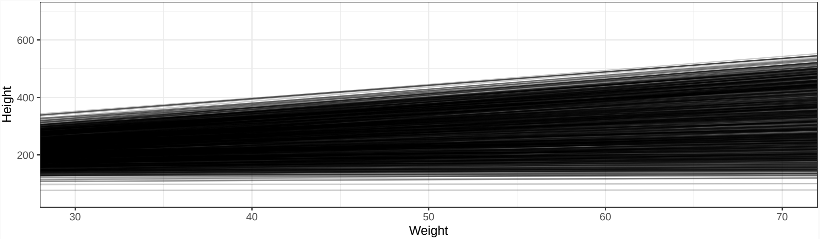
Estimating this model

```r
data(Howell1)
d<-Howell1 %>%
  filter(age>=18)
m0<-quap(
  flist = alist(
    height ~ dnorm(mu, sigma),
    mu<-a + b * weight,
    a ~ dnorm(150, 25),
    b ~ dunif(0, 5),
    sigma ~ dunif(0,10)
  ),
  data = d
)
```

Generate prior predictions

Prior predictions let us confirm that our priors make logical sense for our question

```r
prior_dist<-extract.prior(m0)
plot_dat<-as.data.frame(prior_dist)
ggplot(plot_dat) +
  geom_blank()+
  xlim(30, 70) +
  ylim(50, 700) +
  geom_abline(aes(intercept = a, slope = b),
              alpha = 0.2) +
  xlab("Weight") +
  ylab("Height")
```

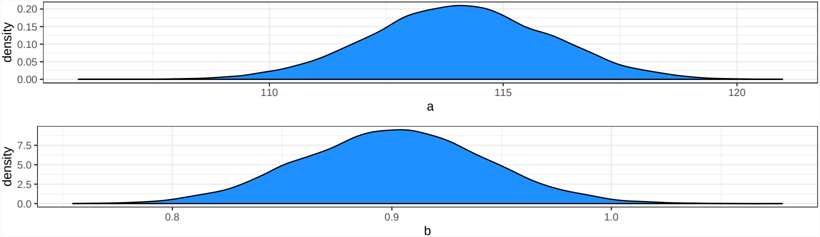


Draw posterior samples and visualize parameters

```
summary(m0)
```

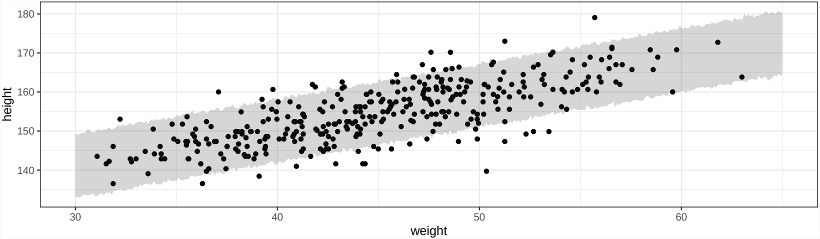
```
##              mean         sd        5.5%       94.5%
## a     114.0880506 1.90021304 111.0511432 117.1249581
## b       0.9004846 0.04181116   0.8336623   0.9673069
## sigma   5.0719529 0.19116289   4.7664377   5.3774682
```

```
post_m0<-extract.samples(m0)
a<-ggplot(post_m0, aes(x = a)) + geom_density(fill = ”dodgerblue”)
b<-ggplot(post_m0, aes(x = b)) + geom_density(fill = ”dodgerblue”)
grid.arrange(a, b)
```



Predict from the posterior and compare to observed

```r
sim_dat<-data.frame(weight = seq(30, 65, length.out = nrow(d))) # generate weights to predict at
sims<-sim(m0, data = sim_dat) ## draw posterior predictions using defined weights
sims_pi<-apply(sims, 2, PI) ## construct 89% PI
sim_dat$sim_upr<-sims_pi[2,] ## attach PI to plotting data.frame
sim_dat$sim_lwr<-sims_pi[1,] ## attach PI to plotting data.frame
ggplot(d, aes(x = weight, y = height)) +
  geom_point() + ## add scatterplot
  geom_ribbon(aes( ## add PI from posterior predictions
    x = sim_dat$weight,
    ymin = sim_dat$sim_lwr,
    ymax = sim_dat$sim_upr),
    alpha = 0.2)
```

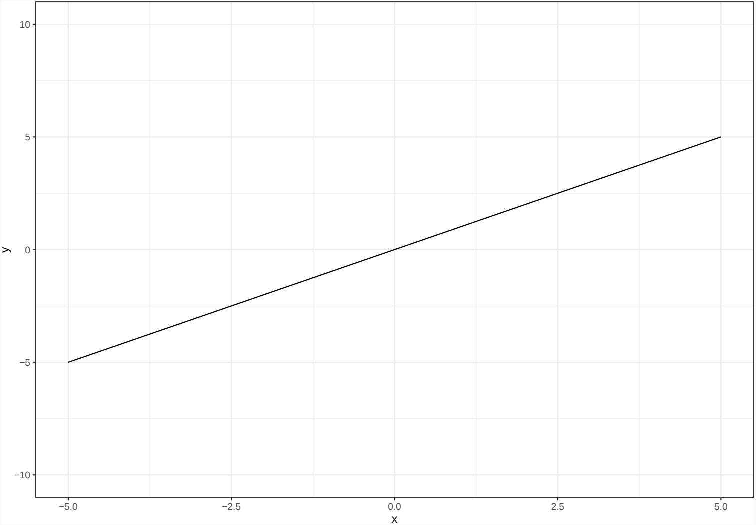


The basic process

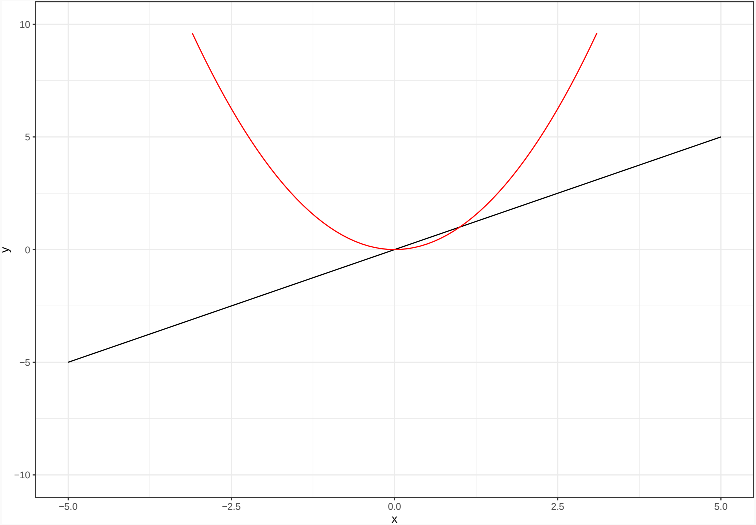
1. Define a model
2. Evaluate / critique your priors
3. Fit the model
4. Evaluate fit / critique model
5. Repeat

Fitting curves in linear regression models

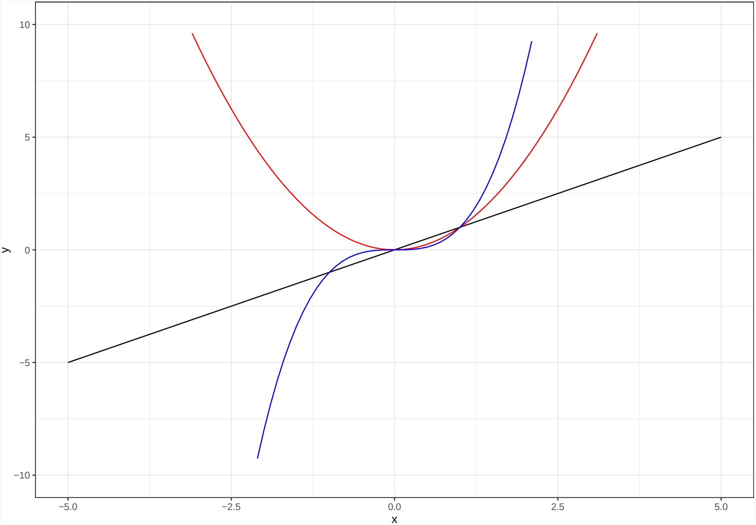
Polynomials: linear



Polynomials: quadratic

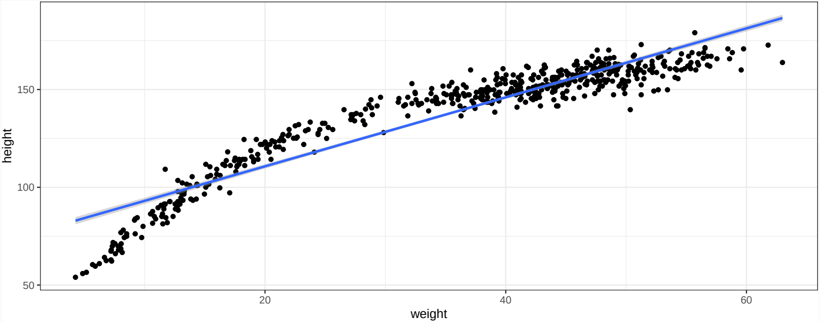


Polynomials: cubic



When do we want a polynomial?

```
d2<-Howell1
ggplot(d2, aes(x = weight, y = height)) +
  geom_point() +
  geom_smooth(method = ”lm”)
```



Defining a model with a polynomial term

$$h_i \sim Normal(\mu, \sigma)$$

$$\mu = \alpha + \beta_1 x_i + \beta_2 x_i^2$$

$$\alpha \sim Normal(0, 5)$$

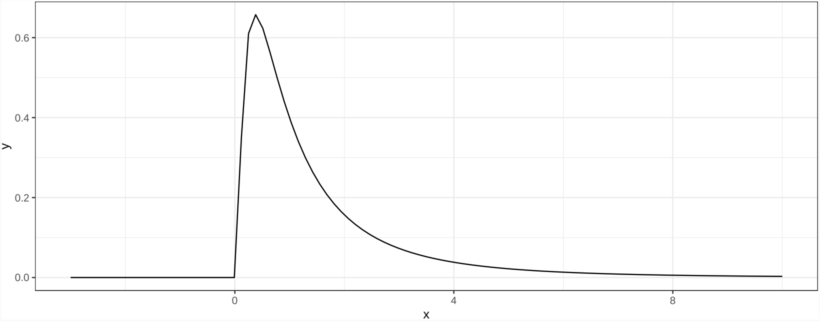
$$\beta_1 \sim LogNormal(0, 1)$$

$$\beta_2 \sim Normal(0, 1)$$

$$\sigma \sim Exponential(1)$$

Why use the log-Normal distribution?

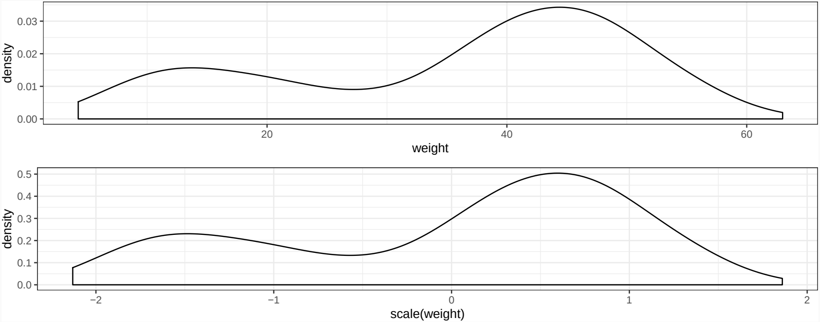
```
ggplot(data = data.frame(x = c(-3, 10)),
       aes(x = x)) +
  stat_function(fun = dlnorm)
```



Scaling variables

Rescaling variables: $\frac{x_i - \overline{x}}{sd(x)}$ doesn't change the shape of a variable's distribution. Priors are easier to define and models easier to fit.

```
p_original<-ggplot(d2, aes(x = weight)) +
  geom_density()
p_scaled<-ggplot(d2, aes(x = scale(weight))) +
  geom_density()
grid.arrange(p_original, p_scaled)
```

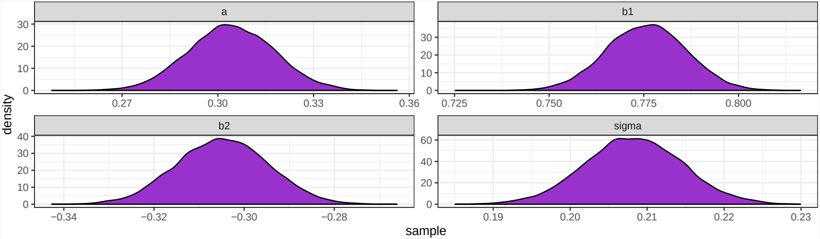


Fitting the model

```
d2<-d2 %>%
  mutate(height.s = scale(height),
         weight.s = scale(weight))
m_quad<-quap(
  alist(height.s ~ dnorm(mu, sigma),
        mu<- a + b1 * weight.s + b2 * weight.s^2,
        a ~ dnorm(0, 5),
        b1 ~ dlnorm(0, 1),
        b2 ~ dnorm(0,1),
        sigma ~ dexp(1)),
  data = d2
)
```

Evaluating the posterior for each parameter

```r
## extract posterior samples
m_quad_post<-extract.samples(m_quad)
## format for plotting with pivot_longer()
m_quad_plot<-m_quad_post %>%
  pivot_longer(cols = everything(),
               names_to = "parameter",
               values_to = "sample")
## plot with facet_wrap
ggplot(m_quad_plot, aes(x = sample)) +
  geom_density(fill = "darkorchid") +
  facet_wrap(~parameter, scales = "free")
```



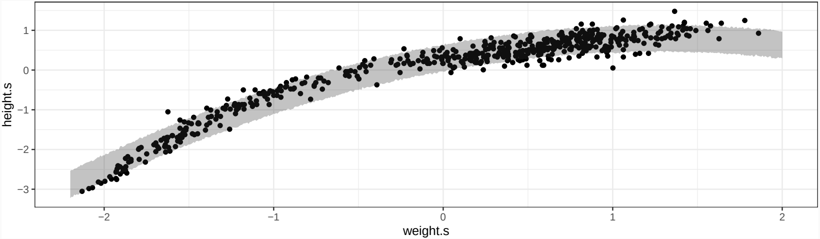
Compare to MAP and posterior credible intervals (89%)

```r
summary(m_quad)
```

```
##              mean          sd       5.5%      94.5%
## a       0.3042272 0.013537975  0.2825909  0.3258635
## b1      0.7757469 0.010503956  0.7589596  0.7925343
## b2     -0.3047910 0.010194448 -0.3210837 -0.2884983
## sigma   0.2082656 0.006311462  0.1981787  0.2183525
```

Visualize the fit

```
sim_dat<-data.frame(weight.s = seq(-2.2, 2, length.out = nrow(d2)))
mu_post<-sim(m_quad, data =sim_dat)
mu_post_PI<-apply(mu_post, 2, PI)
sim_dat<-sim_dat %>%
  mutate(lwr = mu_post_PI[1,],
         upr = mu_post_PI[2,])
ggplot(d2,
       aes(x = weight.s, y = height.s)) +
  geom_point() +
  geom_ribbon(aes(x = sim_dat$weight.s,
                  ymax = sim_dat$upr,
                  ymin = sim_dat$lwr),
              alpha = 0.3)
```

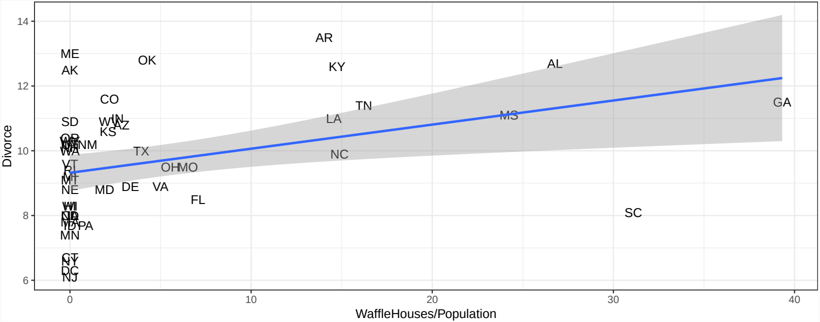


Multiple regression

I have an irrational affection for this place

Waffle house and divorce rates

```r
data("WaffleDivorce")
ggplot(WaffleDivorce,
       aes(x = WaffleHouses / Population,
           y = Divorce,
           label = Loc)) +
  geom_text() +
  geom_smooth(method = "lm")
```



Why add variables to a regression?

When we aim to estimate a causal relationship:

1. Confounding
2. Multiple causation
3. Interactions

When we are not estimating a causal relationship:

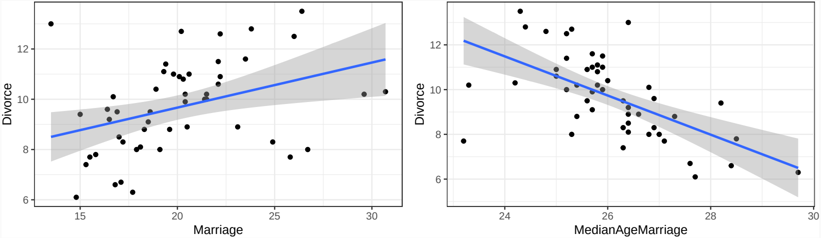
1. Predictive accuracy

More plausible causes

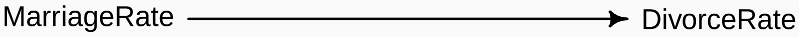
```
p1<-ggplot(WaffleDivorce,
           aes(x = Marriage,
               y = Divorce)) +
  geom_point()+
  geom_smooth(method = ”lm”)

p2<-ggplot(WaffleDivorce,
           aes(x = MedianAgeMarriage,
               y = Divorce)) +
  geom_point()+
  geom_smooth(method = ”lm”)

grid.arrange(p1, p2, ncol=2)
```

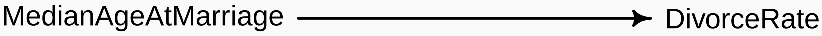


Two theories for the causes of divorce rates



MarriageRate

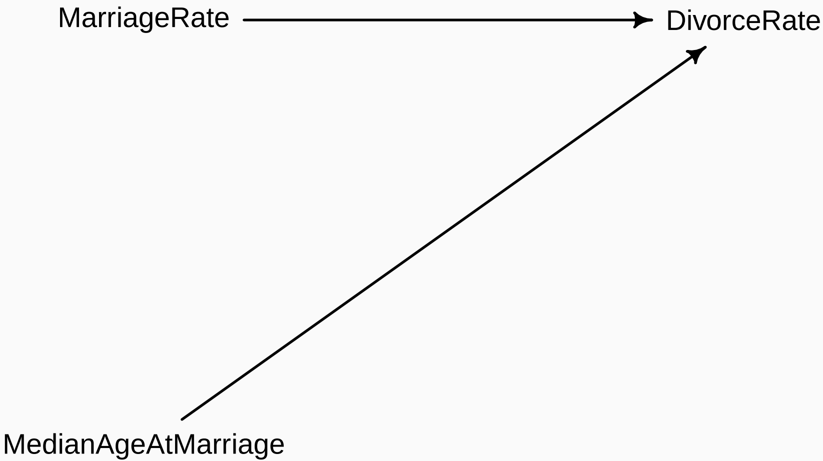
DivorceRate



MedianAgeAtMarriage

DivorceRate

An equivalent statement



Propose a model for divorce rates with age at first marriage as a predictor

$$D_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_A A_i$$

$$\alpha \sim Normal(0, 0.2)$$

$$\beta_A \sim Normal(0, 0.5)$$

$$\sigma \sim Exponential(1)$$

Load and scale variables

Recall that scaling variables to $\bar{x} = 0, sd = 1$ makes defining priors and fitting complex models *much* easier.

```r
WaffleDivorce<- WaffleDivorce %>%
  mutate(A = scale(MedianAgeMarriage),
         D = scale(Divorce),
         M = scale(Marriage))
```

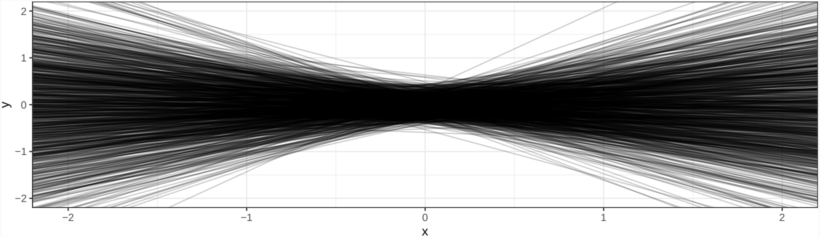
Fit the model(s)

```
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
)
```

Evaluate the priors

```r
prior<-as.data.frame(extract.prior(mAge))
axis_scales<-data.frame(x = c(-2,2),
                        y = c(-2,2))
ggplot(axis_scales,
       aes(x = x, y = y)) +
  geom_blank() +
  geom_abline(data = prior,
              aes(intercept = a,
                  slope = bA),
              alpha = 0.2)
```



Evaluate the posterior

```r
summary(mAge)
```

```
##                mean         sd       5.5%      94.5%
## a     -9.422234e-08 0.09737869 -0.1556301  0.1556299
## bA    -5.684027e-01 0.10999970 -0.7442035 -0.3926020
## sigma  7.883249e-01 0.07801114  0.6636480  0.9130018
```

```r
summary(mMarriage)
```

```
##               mean        sd       5.5%     94.5%
## a     7.884148e-07 0.1082465 -0.1729980 0.1729996
## bM    3.500548e-01 0.1259275  0.1487983 0.5513114
## sigma 9.102662e-01 0.0898626  0.7666484 1.0538840
```

Adding a second predictor

Perhaps age at first marriage and overall divorce rate both impact divorce rates.

$$D_i \sim \mathit{Normal}(\mu, \sigma)$$

$$\mu_i = \alpha + \beta_M M_i + \beta_A A_i$$

$$\alpha \sim \mathit{Normal}(0, 0.2)$$

$$\beta_M \sim \mathit{Normal}(0, 0.5)$$

$$\beta_A \sim \mathit{Normal}(0, 0.5)$$

$$\sigma \sim \mathit{Exponential}(1)$$

Note: we'll consider DAGs in detail next week

Estimating the model

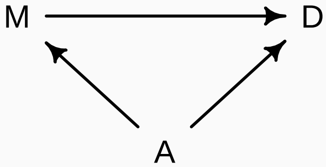
```
mBoth<-quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu<-a + bA * A + bM * M,
    a ~ dnorm(0, 0.2),
    bA ~ dnorm(0, 0.5),
    bM ~ dnorm(0, 0.5),
    sigma ~ dexp(1)),
  data = WaffleDivorce
)

summary(mBoth)
```

```
##                  mean         sd       5.5%      94.5%
## a      1.462063e-06 0.09707596 -0.1551447  0.1551476
## bA    -6.135125e-01 0.15098348 -0.8548132 -0.3722117
## bM    -6.537987e-02 0.15077294 -0.3063441  0.1755844
## sigma  7.851172e-01 0.07784320  0.6607087  0.9095257
```

What’s going on here?

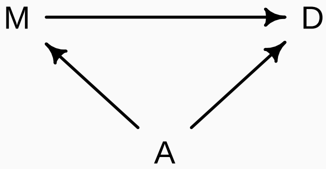
Do both age at marriage and overall marriage rate contribute to the divorce rate? Do both A and M have a causal impact on D?



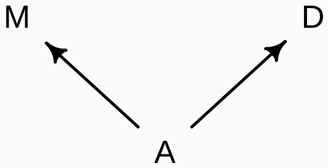
Causal graphs (DAGs)

This DAG assumes:

- A lower age at first marriage (A) leads to higher divorce rates (D)
- More marriages (M) could mean either more divorces (opportunities) or less divorces (stronger norms)
- A lower age at first marriage probably leads to more marriages
- Age at first marriage affects divorce both directly and indirectly through its effect on overall marriage rates
- A, D, and M are all correlated with each other



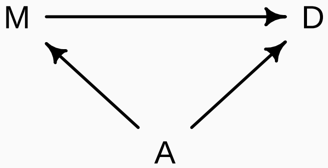
An alternative model: spurious association



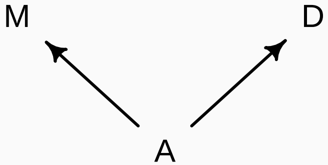
- This model suggests that D and M are only associated with each other because of their relationship with A.
- Another way of saying this: Conditional on A, D is independent of M: $D \perp\!\!\!\perp M|A$

Comparing models

Recall that DAG 1 implies that A, D, and M are all associated with each other

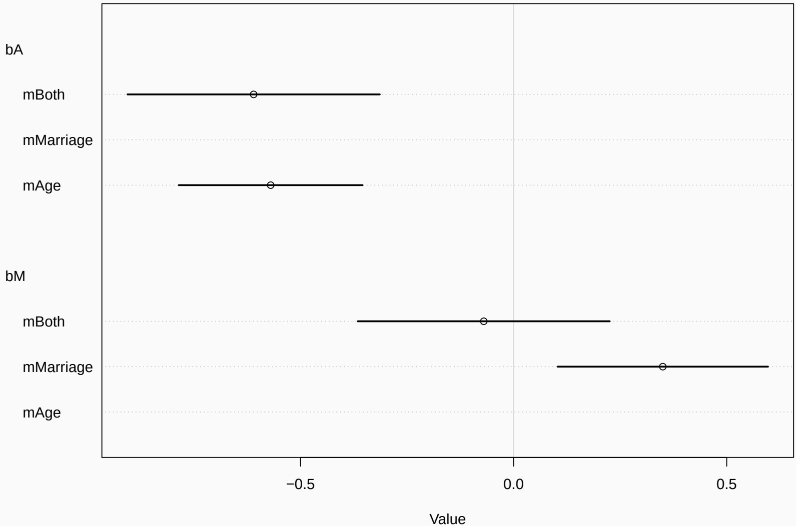


And that DAG 2 implies that D and M are only associated because of their relationship with A.



Comparing results

```
plot(coeftab(mAge, mMarriage, mBoth),
     par = c("bA", "bM"))
```



Next week

- Much more multiple regression
- Visualizing multiple regression
- More DAGs and causality
- HW 4 is posted