Many Variables (Part 1)

EES 4891-06/5891-01

Bayesian Statistical Methods

Jonathan Magnolia Gilligan

Class #6: Monday, February 02 2026

Outline of Regression Modeling

Outline of Regression Modeling

1. Develop a formula relating outcome variable (y) to predictor variables (x_1, x_2, ...)

$$y \sim \text{Normal}(\mu, \sigma)$$
$$\mu = \alpha + \beta x$$

   - You can use theory and/or exploratory analysis of your actual data.

2. Develop priors for parameters

$$\alpha \sim \text{Normal}(178, 20)$$
$$\beta \sim \text{Normal}(0, 10)$$
$$\sigma \sim \text{Exponential}(1)$$

   - Use theory
   - Check using *prior predictions*:
     - Does applying priors to formulas make reasonable predictions

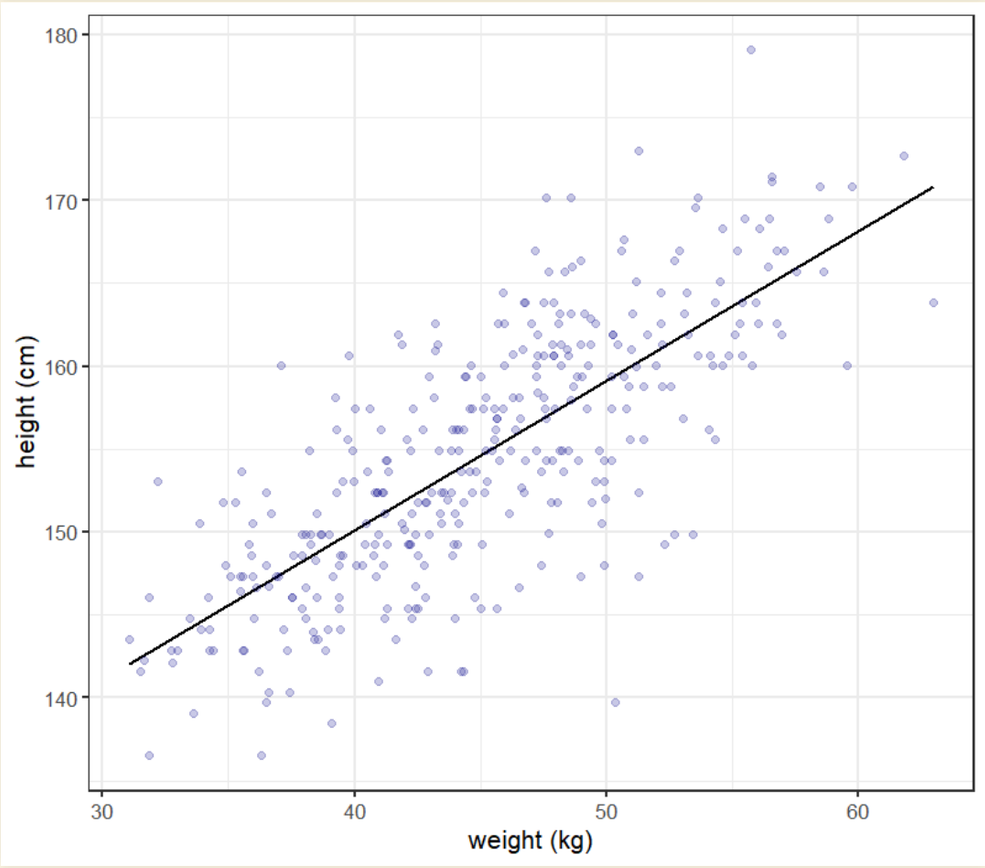
3. Fit model:
   - Combine *formulas*, *priors*, and *observed data* to produce *posterior* probability distributions of parameters
   - *Prior* represents what you knew about the parameters before getting the new data.
   - *Posterior* reflects your improved knowledge about parameters after incorporating new observations.
4. Check model:
   - Compare *posterior predictions* to your observed data.
   - Covariance among posteriors for different parameters may indicate a problem with your formulas.

Checking Models

Checking fit

- Sample *a* and *b* from the posterior
- Plot a line with the median slope and intercept
- Plot the original data

```
post <- extract.samples(lin_mdl)
map  <- summarize(post, a = median(a), b = median(b))
map_line <- tibble(weight = range(d2$weight),
                   height = map$a +
                            map$b * (weight - wbar))

ggplot(d2, aes(x = weight, y = height)) +
  geom_line(data = map_line, size = 1, color =
        "black") +
  geom_point(size = 2, color = "darkblue", alpha =
        0.2) +
  labs(x = "weight (kg)", y = "height (cm)")
```



What is the Posterior?

- The posterior is *full of lines*
  - Predict distribution of *average* heights

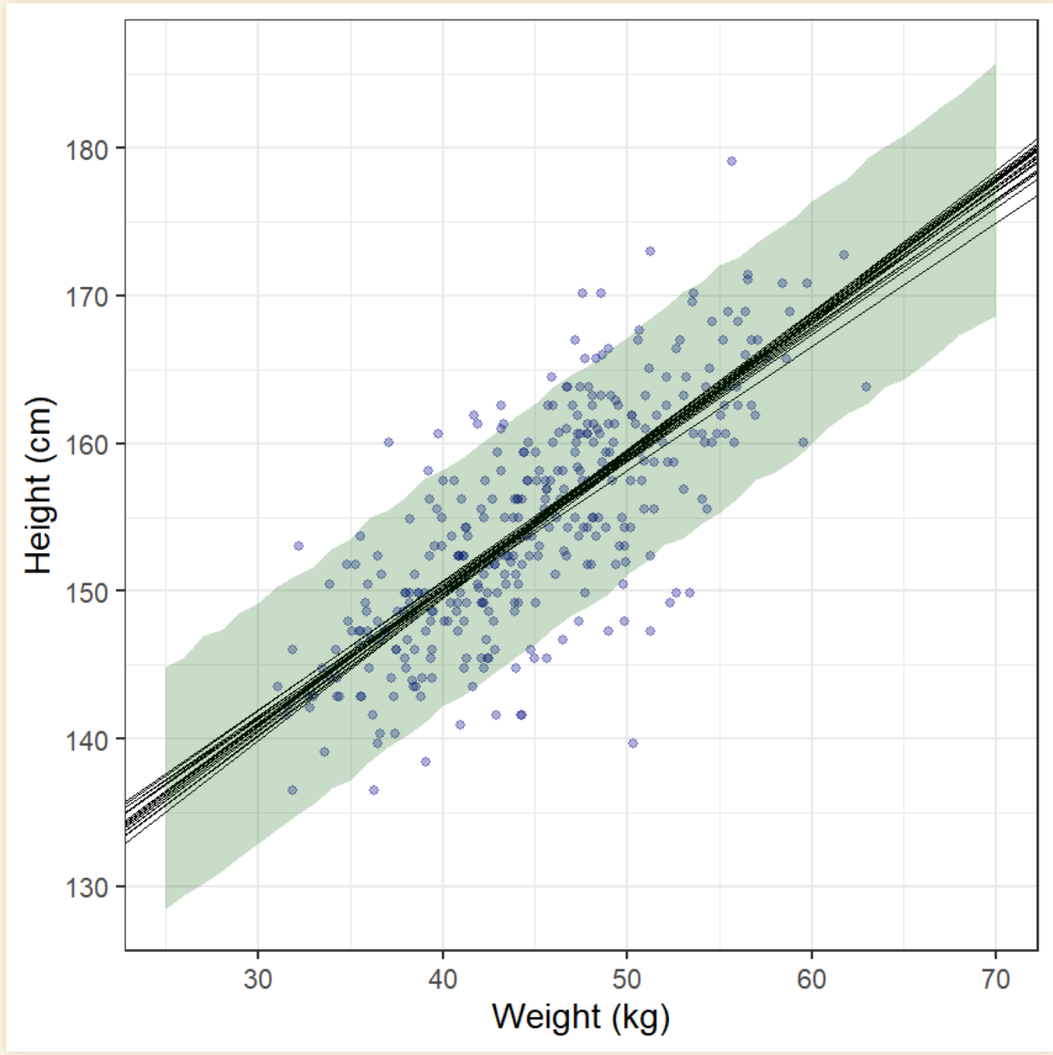
```
lines <- tidy_draws(lin_mdl, n = 20) |>
  mutate(a = a - b * wbar)
```

- The posterior is *full of people*
  - Predict distribution of *individual* heights

```
people <- predicted_draws(lin_mdl, new_data,
                          value = "height") |>
  summarize(lower = quantile(height, 0.055),
            upper = quantile(height, 0.945),
            height = median(height))
```

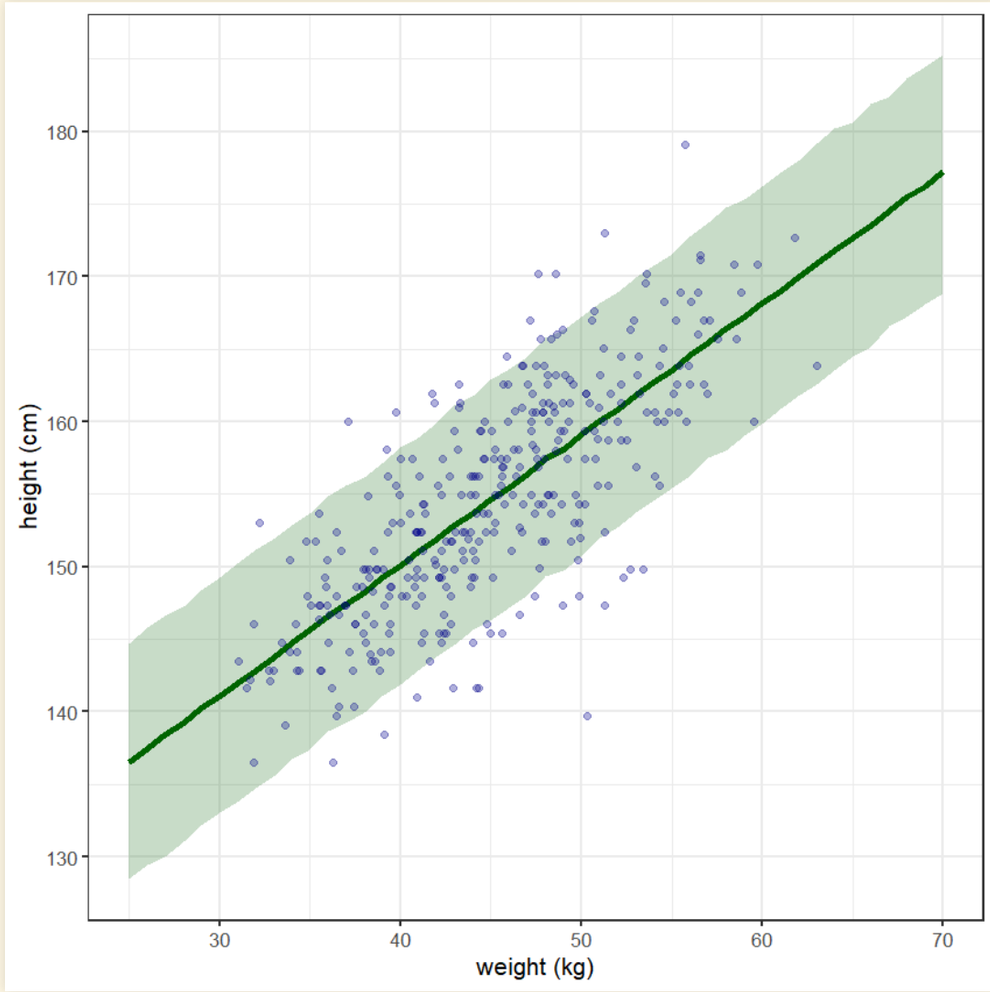
- Plot the result

```
ggplot(d2, aes(x = weight, y = height)) +
  geom_point(size = 2, color = "darkblue", alpha = 0.3)
        +
  geom_abline(data = lines, aes(slope = b, intercept =
        a),
              linewidth = 0.3) +
  geom_ribbon(data = people, aes(ymin = lower, ymax =
        upper),
              fill = "darkgreen", alpha = 0.2) +
  labs(x = "Weight (kg)", y = "Height (cm)")
```



Checking predictions

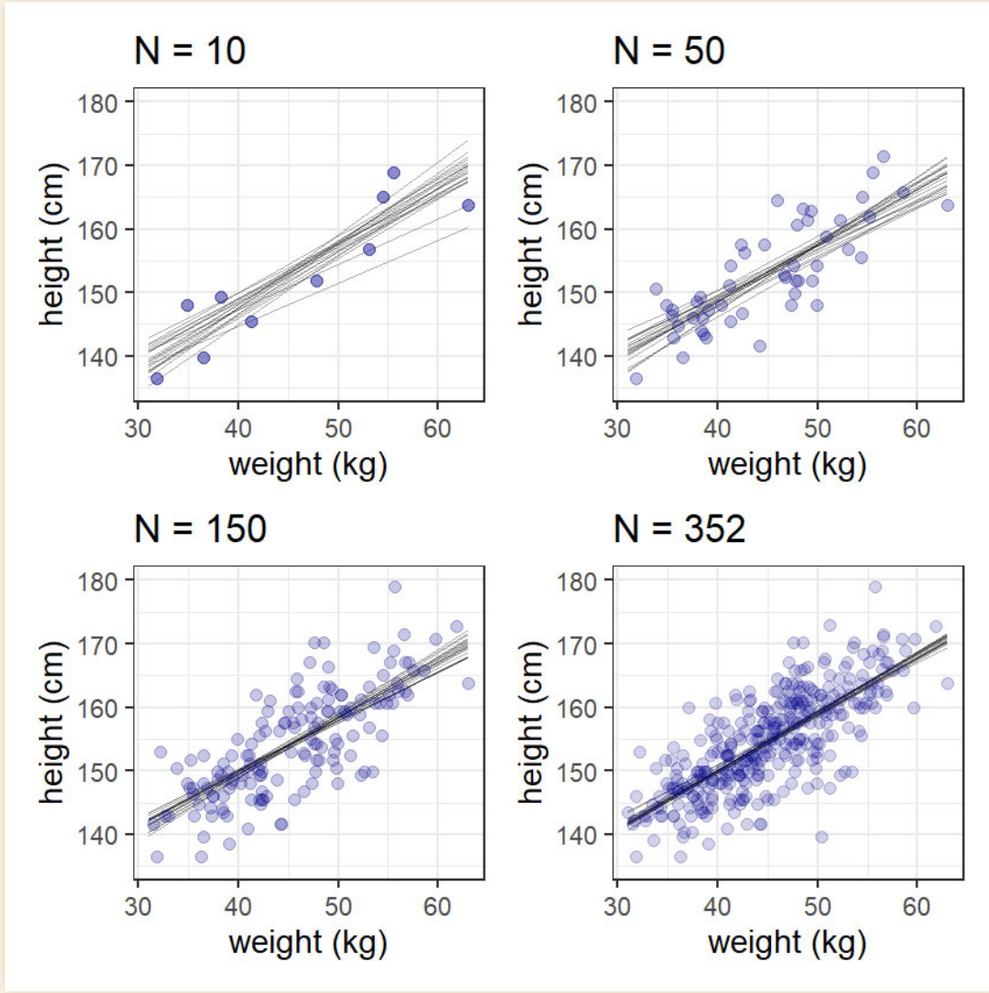
```
post <- predicted_draws(lin_mdl, new_data, value = "height")
post_sum <- summarize(post, lower = quantile(height, 0.055),
                      upper = quantile(height, 0.945),
                      height = mean(height))

p2 <- ggplot(d2, aes(x = weight, y = height)) +
  geom_ribbon(data = post_sum, aes(ymin = lower, ymax = upper),
              fill = "darkgreen", alpha = 0.2) +
  geom_line(data = post_sum, size = 2, color = "darkgreen") +
  geom_point(size = 2, color = "darkblue", alpha = 0.3) +
  labs(x = "weight (kg)", y = "height (cm)")

p2
```



- The graph shows the observed data and the 89% compatibility interval for predictions of where we will see data points.
  - 89% of the data should lie within this interval.
  - `linpred_draws` samples from posterior for $\mu = \alpha + \beta x$
  - `predicted_draws` samples from posterior for $y \sim \text{Normal}(\mu, \sigma)$

Fit Subsets of Data

```
f <- function(N) {
  dN <- slice_head(d2, n = N)
  mN <- quap(flist_lin, data = dN)
  post <- extract.samples(mN, n = 20)

  map_line <- post |> mutate(index = seq(n())) |>
    cross_join(tibble(weight = range(d2$weight))) |>
    mutate(height = a + b * (weight - wbar))

  ggplot(dN, aes(x = weight, y = height)) +
    geom_line(data = map_line, mapping = aes(group=index),
              color = "black", linewidth = 0.5,
              alpha = 0.3) +
    geom_point(size = 3, color = "darkblue", alpha = 1 / log(N)) +
    ylim(135, 180) +
    labs(x = "weight (kg)", y = "height (cm)",
         title = str_c("N = ", N))
}

library(patchwork)

p1 <- f(10)
p2 <- f(50)
p3 <- f(150)
p4 <- f(nrow(d2))

p1 + p2 + p3 + p4 + plot_layout(ncol = 2)
```



Beyond Linear Models

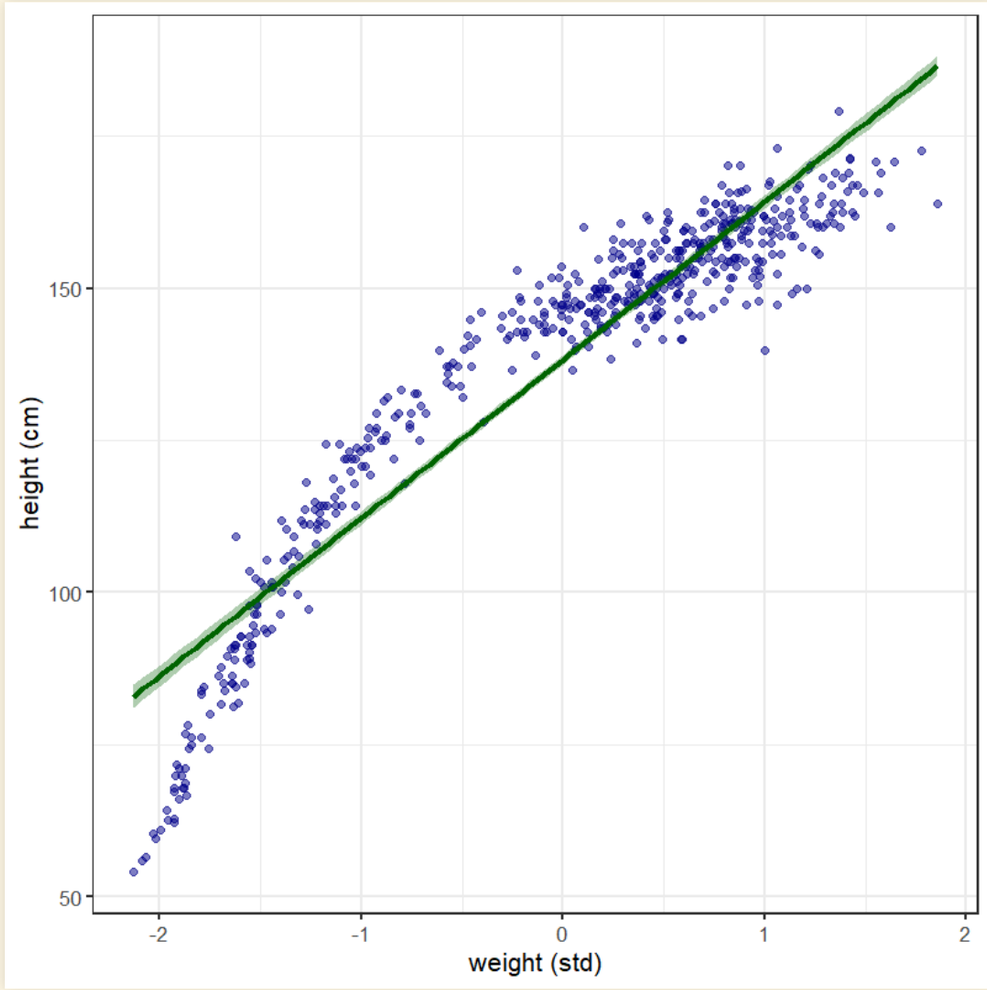
Beyond Linear Models

- Sometimes linear models aren't good enough.
  - If we include children, height is no longer a linear function of weight.
- Polynomial models (quadratic, cubic, etc.) may be better.
- High powers of variables can get large, so **standardize** your data

$$x_s = \frac{x - \bar{x}}{\sigma_x},$$

where $\bar{x}$ is the mean of x and σ_x is the standard deviation.

```
d_std <- mutate(d, ws = (weight - mean(weight)) / sd(weight))

ggplot(d_std, aes(x = ws, y = height)) +
  geom_point(size = 2, color = "darkblue", alpha = 0.5) +
  geom_smooth(method = "lm", color = "darkgreen", fill = "darkgreen",
              alpha = 0.3) +
  labs(x = "weight (std)", y = "height (cm)")
```



Polynomial Regression

- Model:

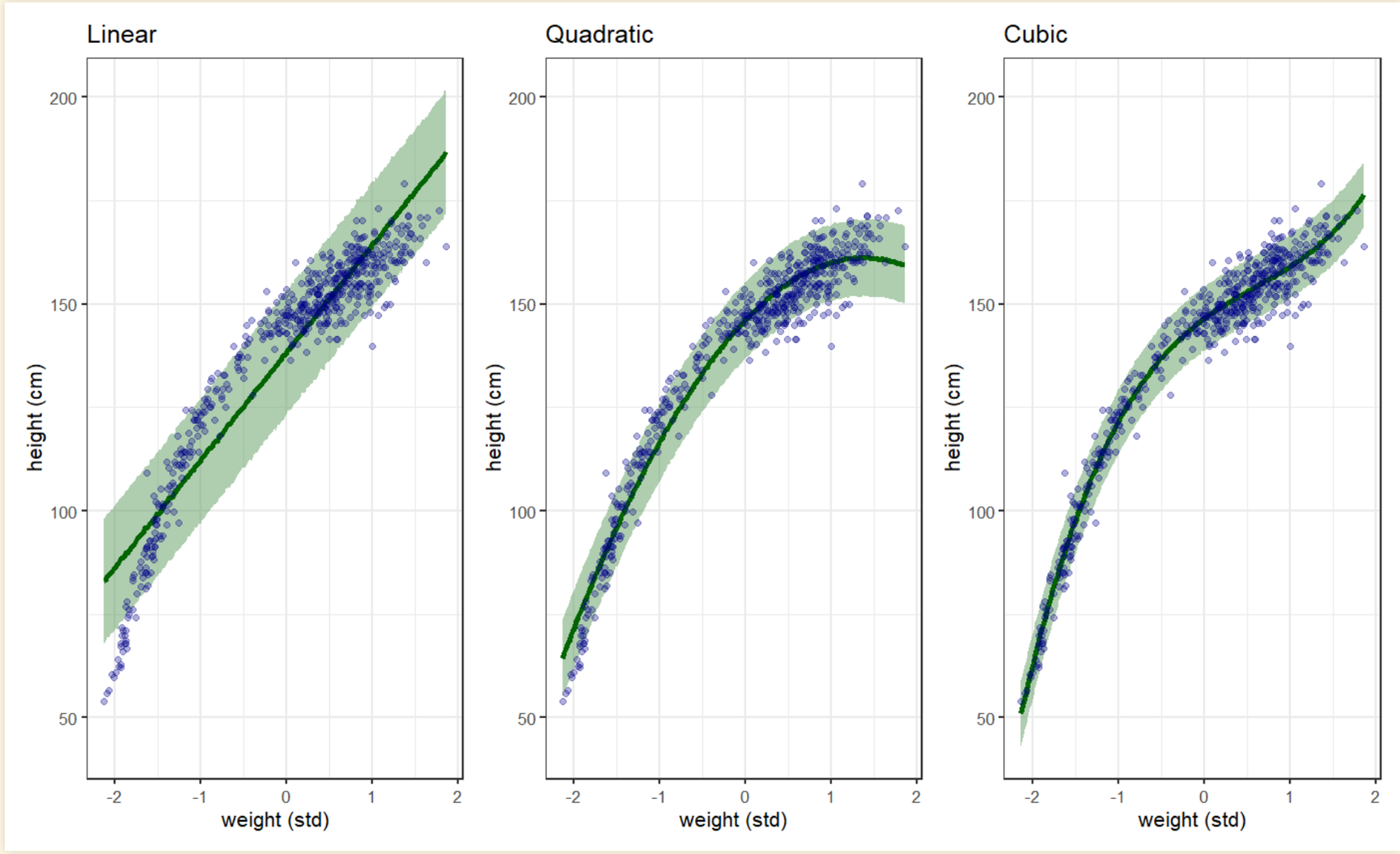
$$
\begin{aligned}
h &\sim \text{Normal}(\mu, \sigma) \\
\mu &= \alpha + \beta_1 w_s + \beta_2 w_s^2 \\
\alpha &\sim \text{Normal}(178, 20) \\
\beta_1 &\sim \text{Log-Normal}(0, 1) \\
\beta_2 &\sim \text{Normal}(0, 1) \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

  - $w_s =$ standardized weight
  - For computational efficiency, make a new variable $ws2 = ws^2$.

```
d_std <- mutate(d, ws = standardize(weight), ws2 = ws^2)

flist_lin_s <- alist(
  height ~ dnorm(mu, sigma),
  mu <- a + b * ws,
  a ~ dnorm(178, 20),
  b ~ dlnorm(0,1),
  sigma ~ dunif(0,50)
)
lin_mdl_s <- quap(flist_lin_s, data = d_std)

flist_quad_s <- alist(
  height ~ dnorm(mu, sigma),
  mu <- a + b1 * ws + b2 * ws2,
  a ~ dnorm(178, 20),
  b1 ~ dlnorm(0,1),
  b2 ~ dnorm(0,1),
  sigma ~ dunif(0,50)
)
quad_mdl_s <- quap(flist_quad_s, data = d_std)

precis(quad_mdl_s)
```

```
##             mean         sd       5.5%      94.5%
## a     146.057412 0.3689756 145.467718 146.647106
## b1     21.733063 0.2888891  21.271362  22.194763
## b2     -7.803267 0.2741839  -8.241466  -7.365068
## sigma   5.774476 0.1764652   5.492450   6.056501
```

Different polynomials



Nonparametric Models

Splines

- Originally from mechanical drafting splines
- Arbitrary smooth curve
- Complexity:
  - Physical splines: “ducks” or “whales”
  - Mathematical splines: “knots”
- Splines are a special case of a class of models called *generalized additive models* (GAMs).

(Photos: Rain Noe, When Splines Were Physical Objects)

Splines in Statistical Regression

- Cherry blossom data
  - Over 1000 years of historical records of cherry tree blossoming dates
    - Y. Aono & S. Saito, Int. J. Biometeorology **54**, 211 (2010).
  - No changes for most of history, but pronounced trend in 20th century (global warming).
- Spline regression:
  - *Basis splines*: for the *i*th point, x_i

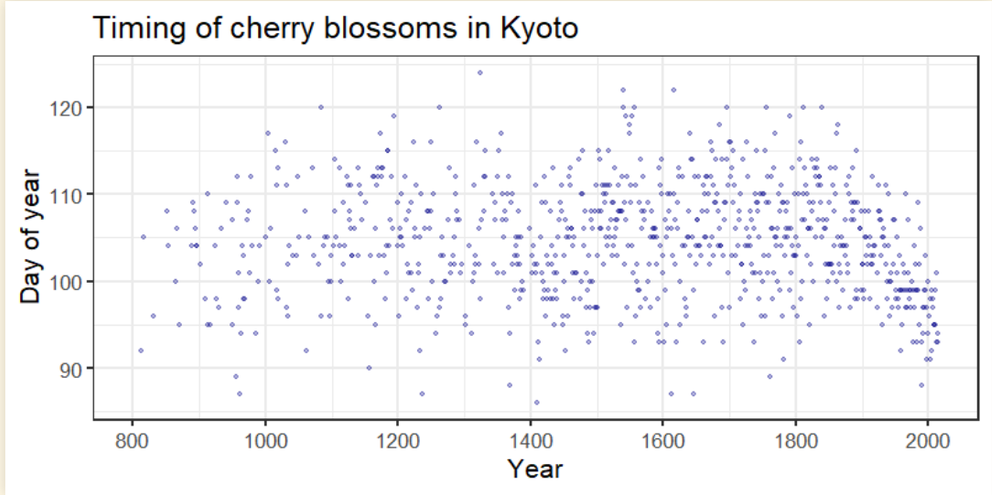
$$\mu_i = \alpha + \sum_{j=1}^{n_{\text{knots}}} w_j B_{i,j}$$

n_{knots} = # knots, w_j = weight for knot j, $B_{i,j}$ = ith row of jth basis function (matrix with one row for each x value, and n_{knots} columns).

```
data(cherry_blossoms)
d <- cherry_blossoms
precis_show(precis(d, digits = 2))
```

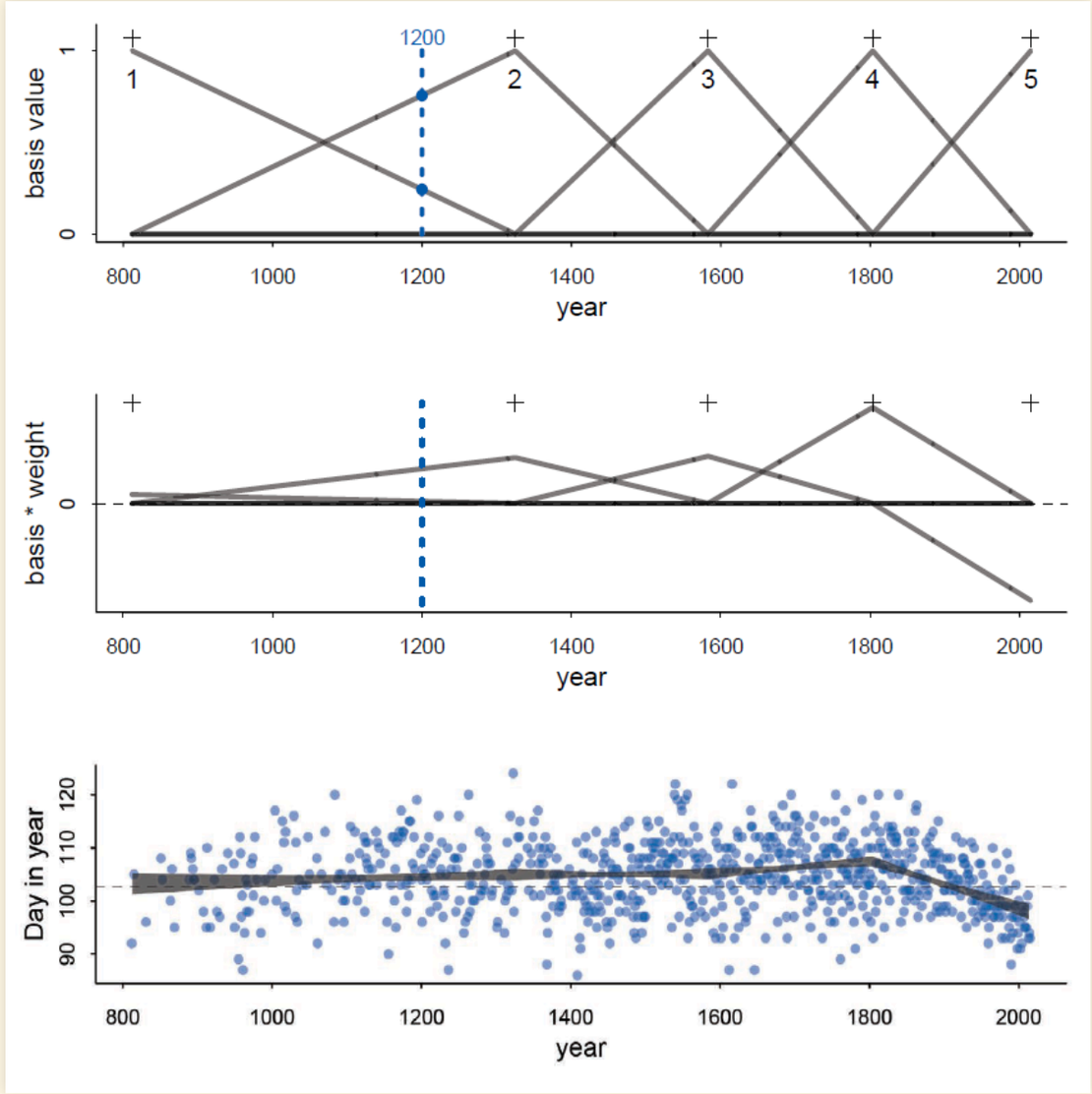
```
## 'data.frame': 1215 obs. of 5 variables:
##                mean     sd   5.5%   94.5%       histogram
## year        1408.00 350.88 867.77 1948.23
## doy          104.54   6.41  94.43  115.00
## temp           6.14   0.66   5.15    7.29
## temp_upper     7.19   0.99   5.90    8.90
## temp_lower     5.10   0.85   3.79    6.37
```

```
ggplot(d, aes(x = year, y = doy)) +
  geom_point(color = "darkblue", size = 1, alpha = 0.3) +
  scale_x_continuous(breaks = seq(600, 2200, by = 200)) +
  labs(x = "Year", y = "Day of year",
       title="Timing of cherry blossoms in Kyoto")
```



Linear Basis Spline

- Linear basis functions $B_j(x)$
  - 5 knots
  - Piecewise linear
  - At most 2 functions are nonzero for any x.
- Model fits weights w_j for each basis function

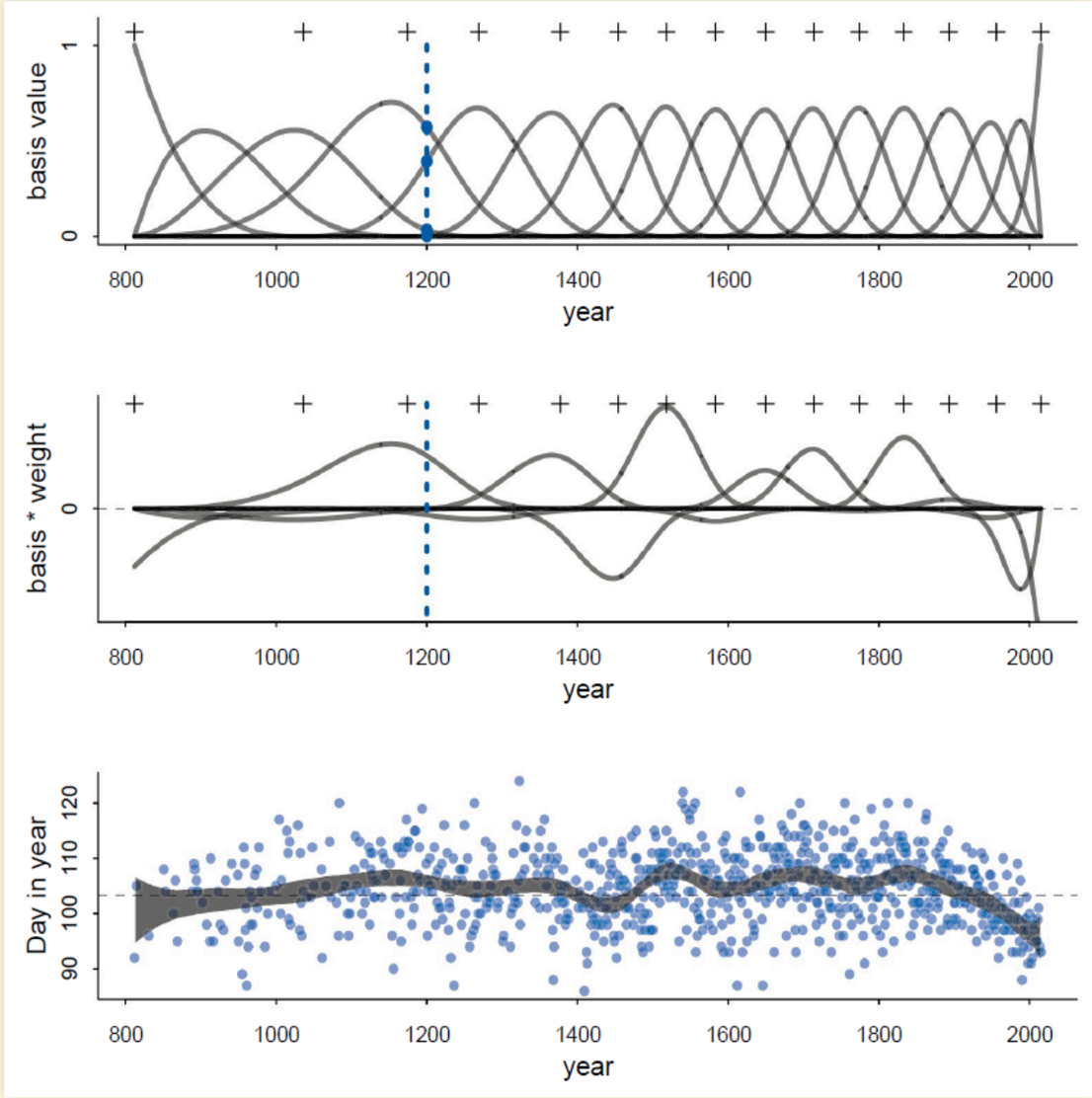


Cubic Basis Spline

- 15 knots
  - Equal # of years with data between knots.
- Cubic functions
- Only 3 have nonzero values for any *x*.

```
library(splines)
d2 <- filter(d, ! is.na(doy)) # omit missing values
n_knots <- 15
knot_list <- quantile(d2$year,
                      probs=seq(0,1, length.out = n_knots))

# Create basis function matrix
B <- bs(d2$year, knots = knot_list[-c(1,n_knots)],
        degree = 3, intercept = TRUE)

mdl <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a + B %*% w,
    a ~ dnorm(100, 10),
    w ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ),
  data = list(D = d2$doy, B = B),
  start = list(w = rep(0, ncol(B)))
)
```

- %∗% means matrix multiplication

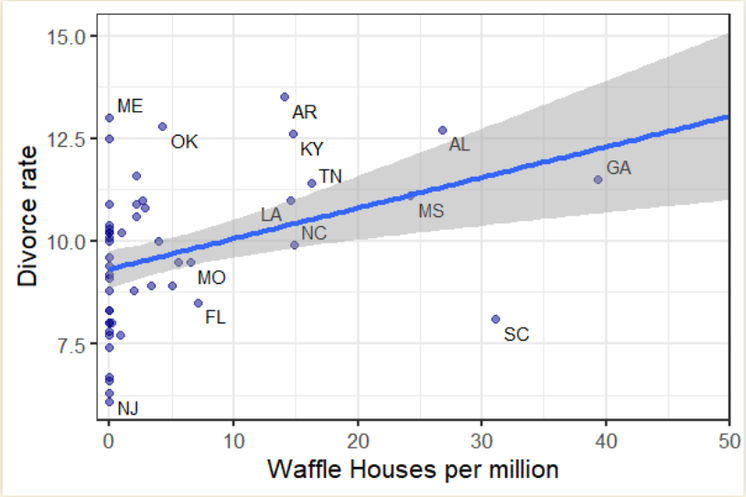


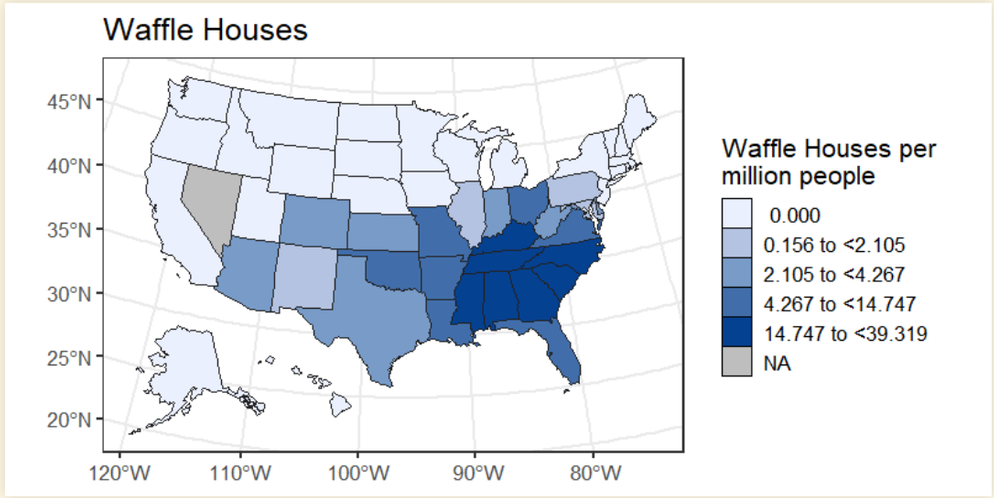
Multiple Regression

Multiple Regression

- Single Regression:
  - Dependent variable depends on one independent (predictor) variable
- Multiple Regression:
  - Dependent variable depends on multiple independent variables.
- Why use Multiple Regression?
  - Statistical controls for confounding effects
    - On average, men are taller than women
    - Studies of how income correlates with height should control for sex.
- Why use Multiple Regression? (cont.)
  - Multiple causes and complex causation
    - Earth's temperature depends on
      - Greenhouse gas concentrations
      - Albedo (reflection of sunlight)
      - Brightness of the sun
  - Interactions (Chapter 8)
    - Smoking increases risk of lung cancer
    - Exposure to radon increases risk of lung cancer
      - The effect of radon is much greater among smokers

Does Waffle House Drive Divorce Rates?





- The Waffle-House Index

US NEWS

Major sign of how bad winter storm will be: Waffle House locations close across the South

By Anna Young
Published Jan. 25, 2026, 2:28 a.m. ET

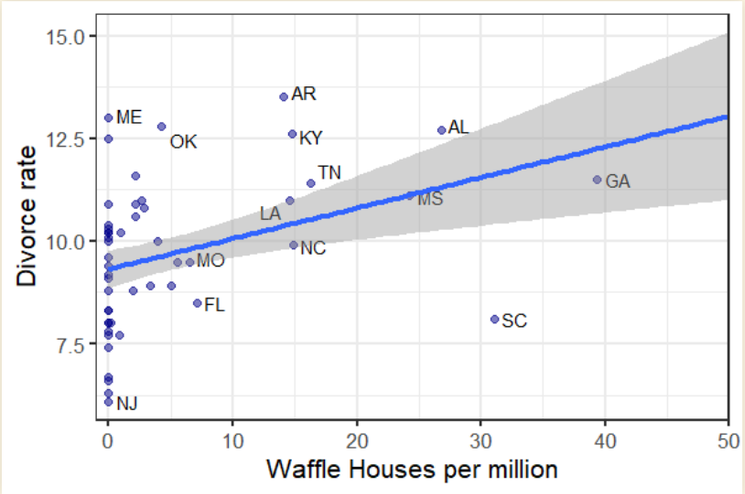
53 Comments

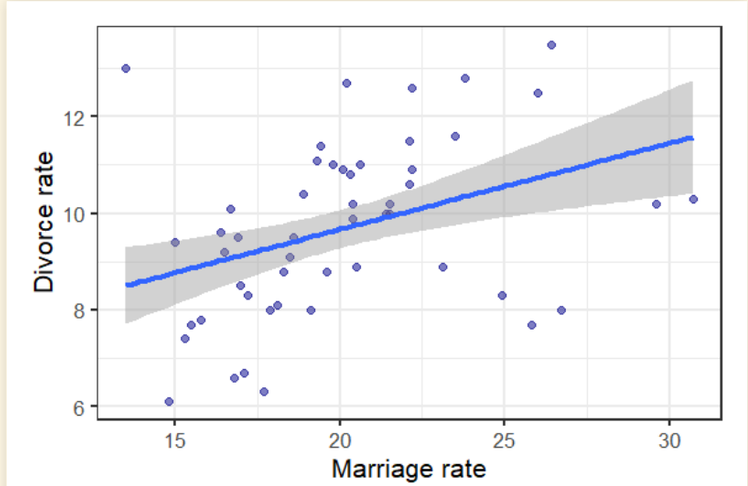
A Tennessee Waffle House has closed because of the winter storm. Here's what to know about the Waffle House Index

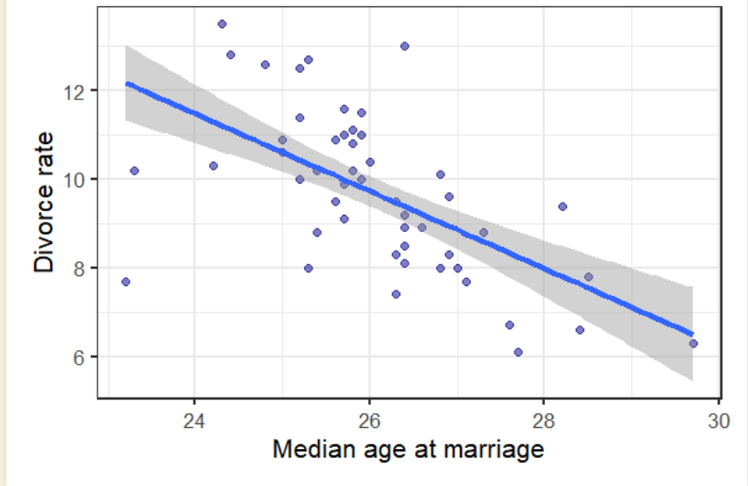
When a Waffle House location closes, it's become an unofficial way of scaling just how dangerous weather conditions are.

A Waffle House closed in Murfreesboro amid the weekend winter storm.

Spurious Association







- Causation vs. correlation

Standardize Data

- Standardizing makes it easier to compare effects of different variables

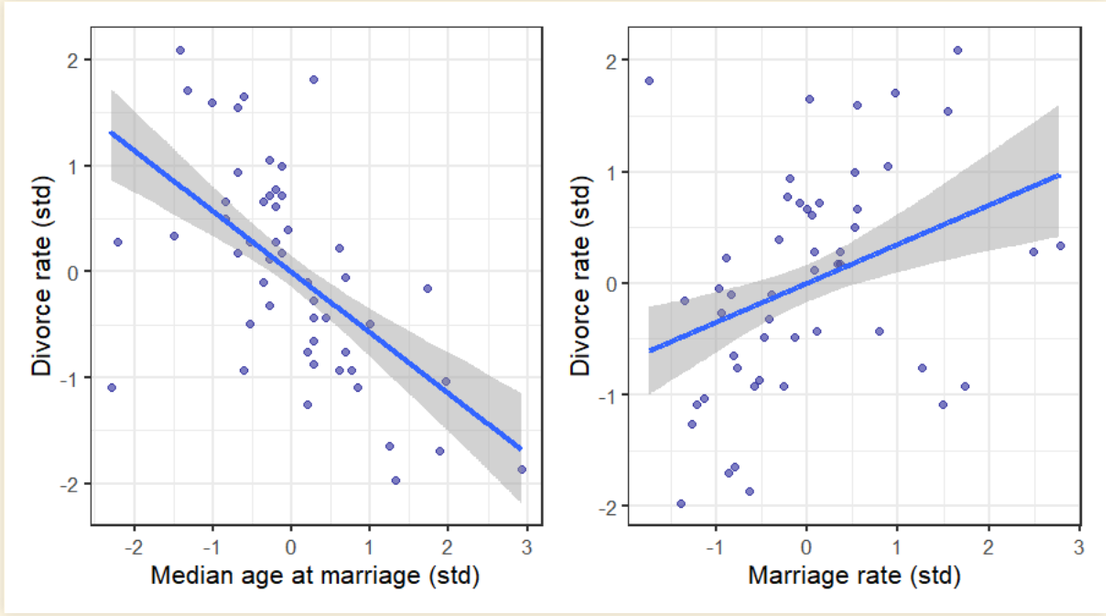
```
library(rethinking)
data(WaffleDivorce)
d <- WaffleDivorce |> mutate(
  D = standardize(Divorce),
  M = standardize(Marriage),
  A = standardize(MedianAgeMarriage)
)
```

- Model

$$
\begin{aligned}
D_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_A A_i \\
\alpha &\sim \text{Normal}(0, 0.2) \\
\beta_A &\sim \text{Normal}(0, 0.5) \\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

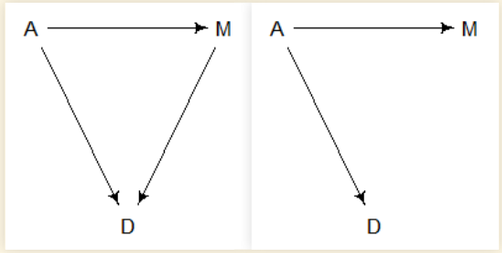
```
mdl_A <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a + bA * A,
    a ~ dnorm(0, 0.2),
    bA ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d)
```

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

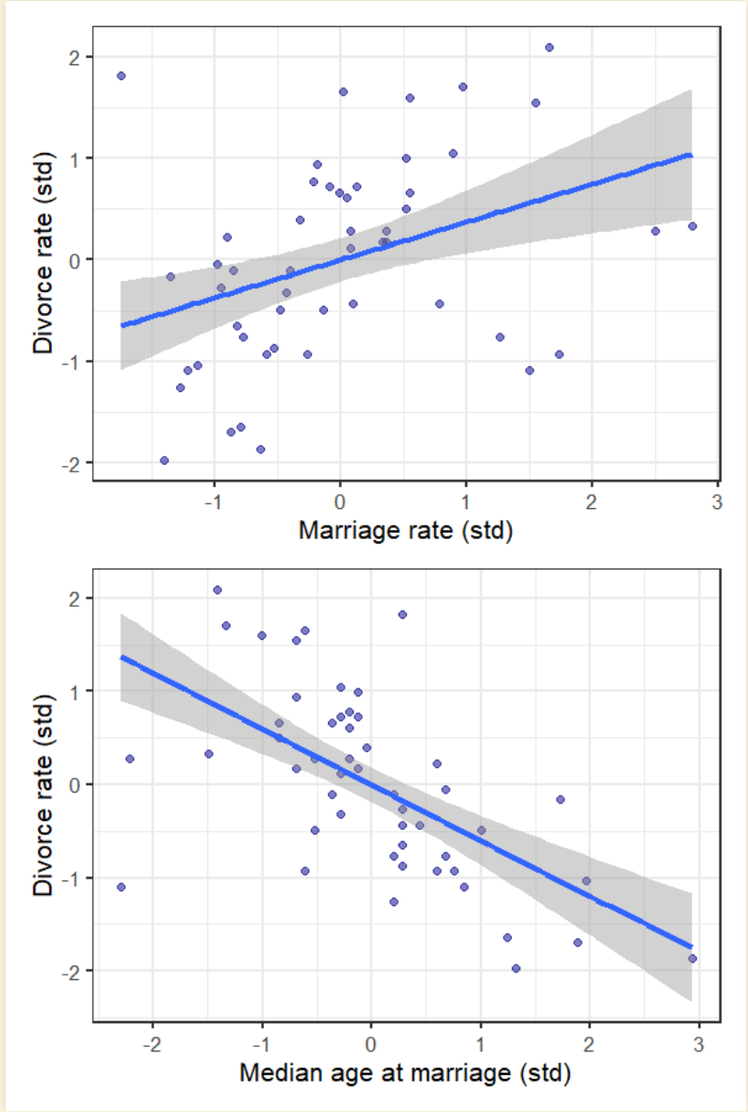


Causal Relationships

- **Think before you regress**
  - The pattern we see here is typical when one variable influences the dependent variable and also the other independent variable.
  - Two possible causal relationships

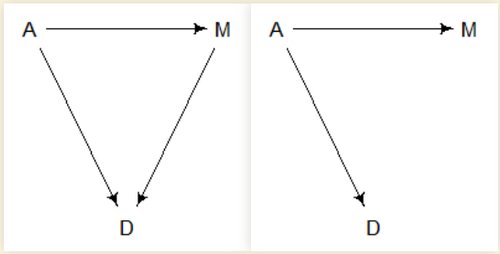


  - Which is it?



Testable Implications

- What are the possible *testable implications* of these DAGs?



- **Conditional independencies:**
  1. Which variables should be associated in the data?
  2. Which variables become dis-associated when we *condition* on another set of variables
     - **Conditioning:** *Y* is independent of *X*, conditioned on *Z* ($Y \perp\!\!\!\perp X | Z$) if *X* doesn't give us any additional information about *Y* when we already know *Z*.
     - Left DAG: There shouldn't be any independence among *D*, *A*, and *M*.
     - Right DAG: $D \perp\!\!\!\perp M | A$.

- Left DAG:

```
library(dagitty)
DMA_dag1 <- dagitty("dag{ D <- A -> M -> D }")
impliedConditionalIndependencies(DMA_dag1)
```

  - No conditional independencies
- Right DAG:

```
DMA_dag2 <- dagitty("dag{ D <- A -> M }")
impliedConditionalIndependencies(DMA_dag2)
```

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

Multiple Regression Models

Multiple Regression Notation

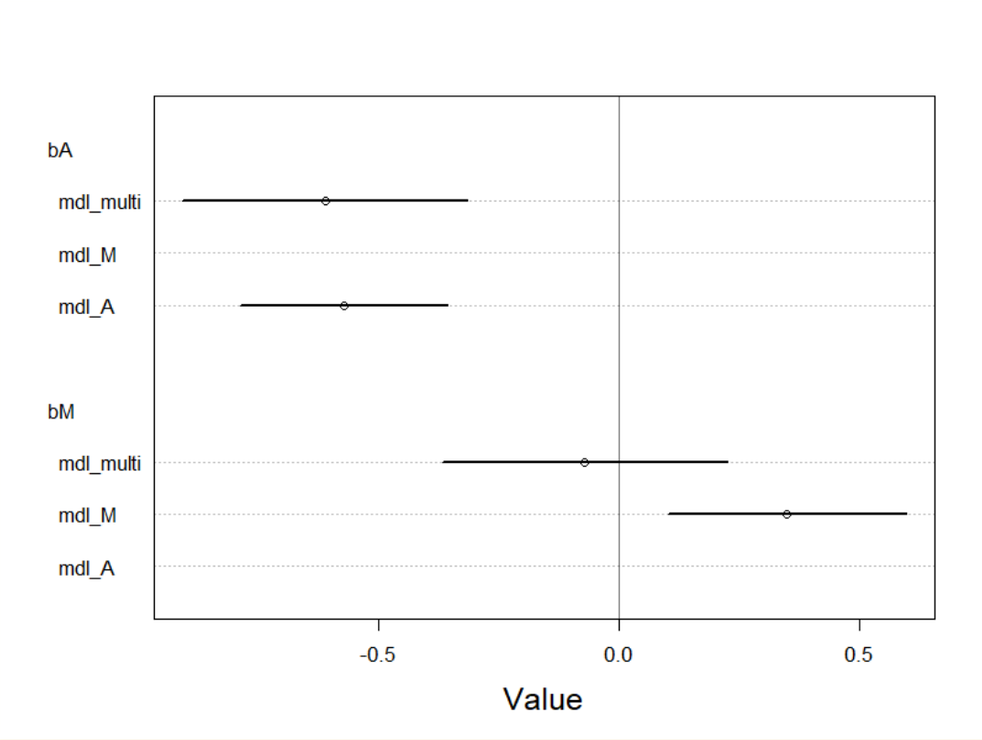
$$
\begin{aligned}
D_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_M M_i + \beta_A A_i \\
\alpha &\sim \text{Normal}(0, 0.2) \\
\beta_M &\sim \text{Normal}(0, 0.5) \\
\beta_A &\sim \text{Normal}(0, 0.5) \\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

```
mdl_multi <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a + bM * M + bA * A,
    a ~ dnorm(0, 0.2),
    bM ~ dnorm(0, 0.5),
    bA ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d
)
precis_show(precis(mdl_multi, digits=2))
```

```
##         mean   sd  5.5% 94.5%
## a       0.00 0.10 -0.16  0.16
## bM     -0.07 0.15 -0.31  0.18
## bA     -0.61 0.15 -0.85 -0.37
## sigma   0.79 0.08  0.66  0.91
```

Comparing Models

```
coeftab_plot(coeftab(mdl_A, mdl_M, mdl_multi), pars = c("bA", "bM"))
```

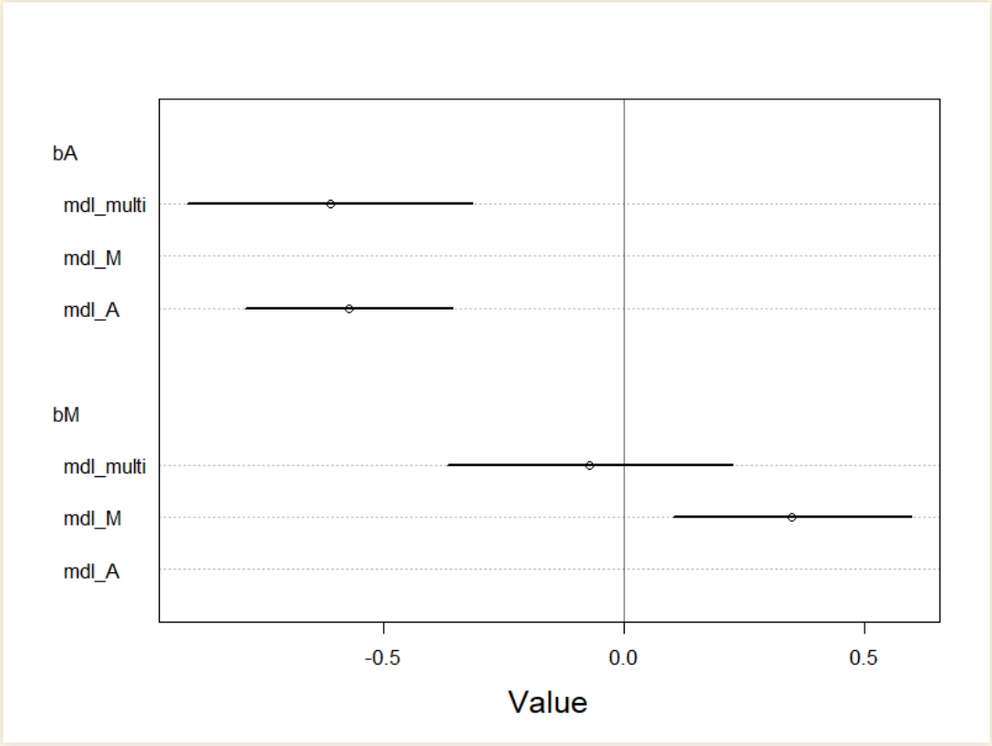


- In the multiple model, b_M is consistent with zero:
  - If we know the median age A,
  - we get no additional information from knowing the marriage rate M.
  - $D \perp\!\!\!\perp M|A$

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

Conclusion

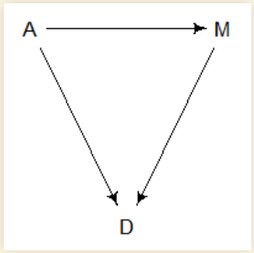
```
coeftab_plot(coeftab(mdl_A, mdl_M, mdl_multi), pars = c("bA", "bM"))
```



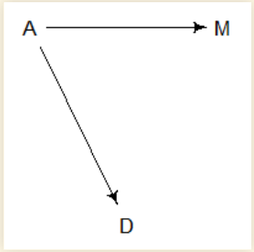
- In the multiple model, b_M is consistent with zero:
  - If we know the median age A,
  - we get no additional information from knowing the marriage rate M.
  - $D \perp\!\!\!\perp M|A$

- Therefore, DAG 2 is more consistent with the data

1. DAG 1 (No conditional independencies)



2. DAG 2 ($D \perp\!\!\!\perp M|A$)



Diagnostic Posterior Plots

Diagnostic Posterior Plots

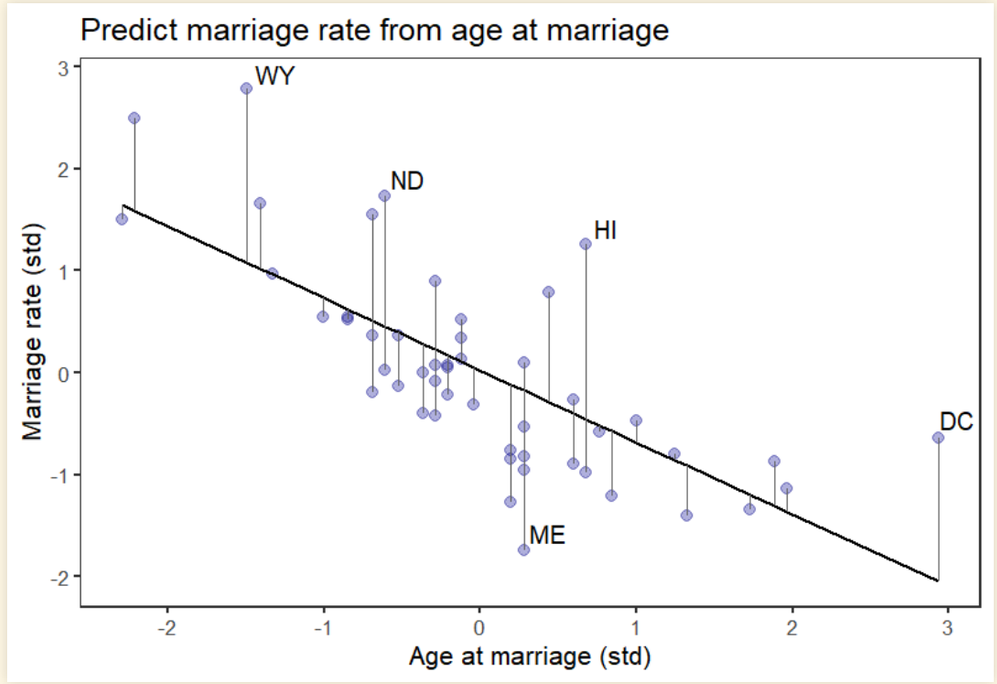
- We can also learn from plotting the posterior in different ways.
  1. **Predictor residual plots:**
     - *Data versus residual predictor values.*
     - These are only useful for understanding the model itself.
  2. **Posterior prediction plots:**
     - *Predictions versus raw data.*
     - Useful for checking how well the model fits the data.
     - Don’t tell us about causal relationships.
  3. **Counterfactual plots:**
     - *Implied predictions for imaginary experiments.*
     - Useful for exploring causal implications.

Predictor Residual Plots

- Predict *M* from *A*

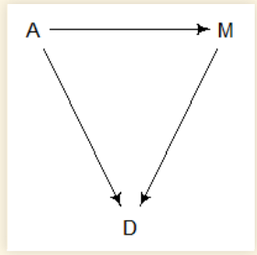
```
mdl_MA <- quap(
  alist(
    M ~ dnorm(mu, sigma),
    mu <- a + b * A,
    a ~ dnorm(0, 0.2),
    b ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d
)

new_data <- select(d, D, A, M, Loc)
notable_states_MA <- c("WY", "ND", "ME",
                       "HI", "DC")

post_MA <- linpred_draws(
    mdl_MA, new_data,
    value = "mu", ndraws = 100 ) |>
  ungroup() |>
  summarize(estimate = mean(mu),
            .by = c("Loc", "D", "A", "M"))
```

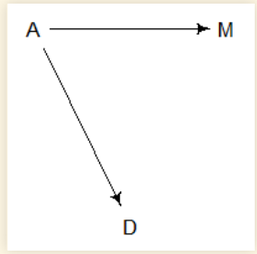
```
library(ggrepel)

ggplot(post_MA, aes(x = A, y = M)) +
  geom_point(size = 3, color = "darkblue", alpha = 0.3) +
  geom_line(aes(y = estimate), size = 1, color = "black") +
  geom_segment(aes(xend = A, yend = estimate), size = 0.25,
               color = "gray40") +
  geom_text_repel(data = . %>% filter(Loc %in% notable_states_MA),
                  aes(label = Loc), size = 5,
                  nudge_x = 0.01, nudge_y = 0.05) +
  labs(x = "Age at marriage (std)", y = "Marriage rate (std)",
       title = "Predict marriage rate from age at marriage") +
  theme_bw(base_size = 15) + theme(panel.grid = element_blank())
```



Which Model?

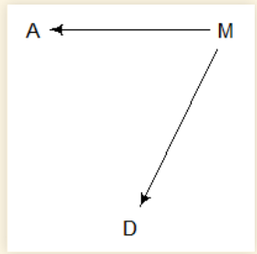
- We compared two models:

1. 

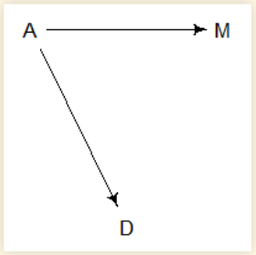
2. 

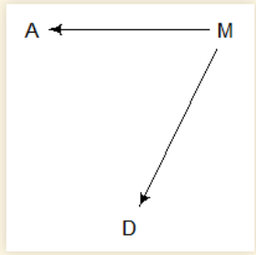
   - We ruled out (1).

- But what about this alternative?

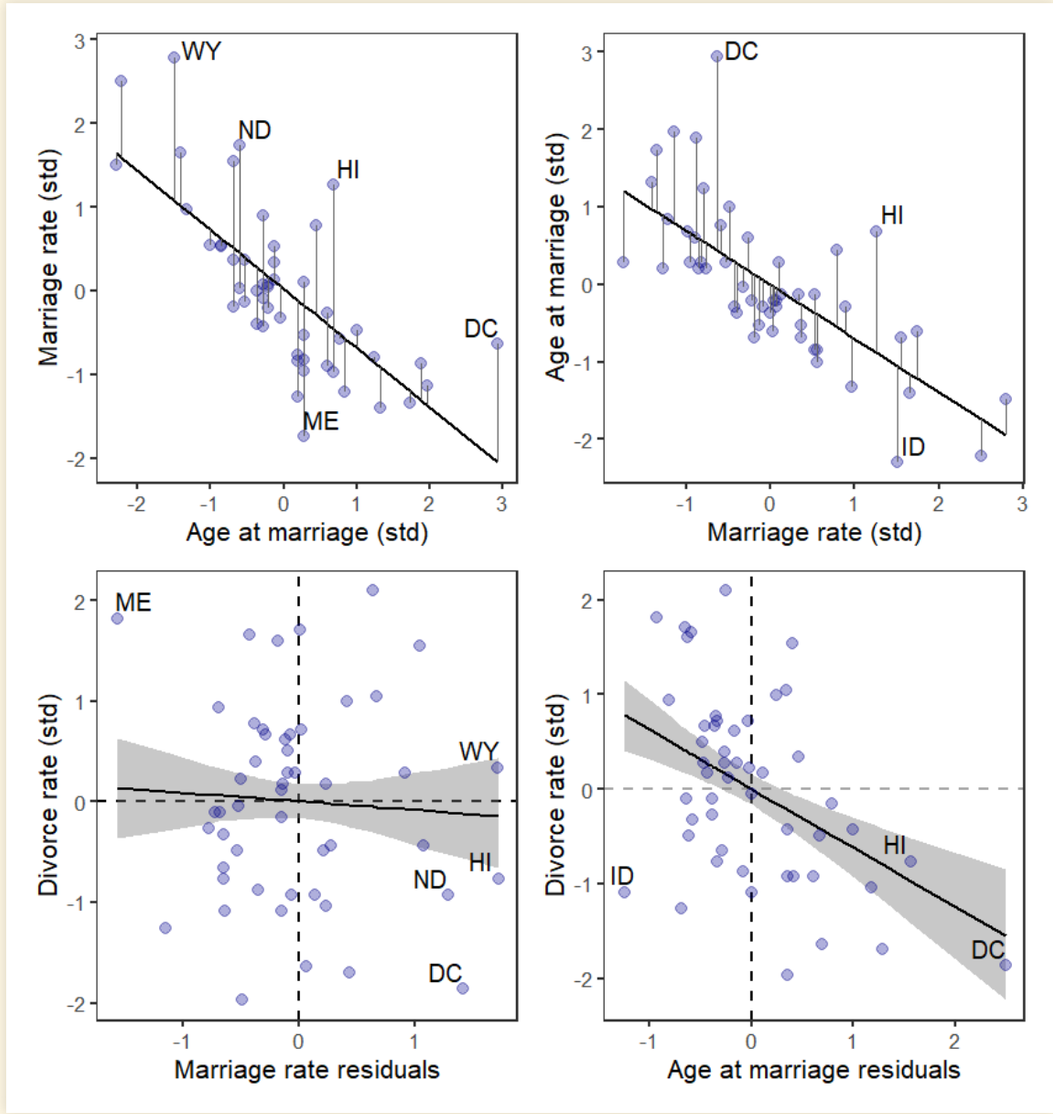
3. 

Compare Residual Plots

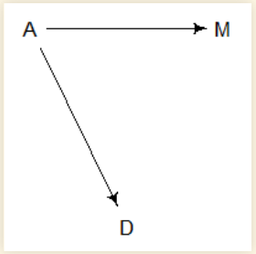
DAG 2 

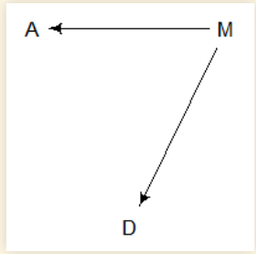
vs. DAG 3 


```
d2 <- post_MA |> mutate(resid = M - estimate)

mdl_MA_resid <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a + b_res * resid,
    a ~ dnorm(0, 0.2),
    b_res ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d2
)
post_MA_2 <- linpred_draws(mdl_MA_resid, select(d2,
        resid),
                           ndraws = 1000, value =
        "estimate") |>
  ungroup() |>
  summarize(mean = mean(estimate),
            lower = quantile(estimate, 0.055),
            upper = quantile(estimate, 0.945),
            .by = "resid")
```

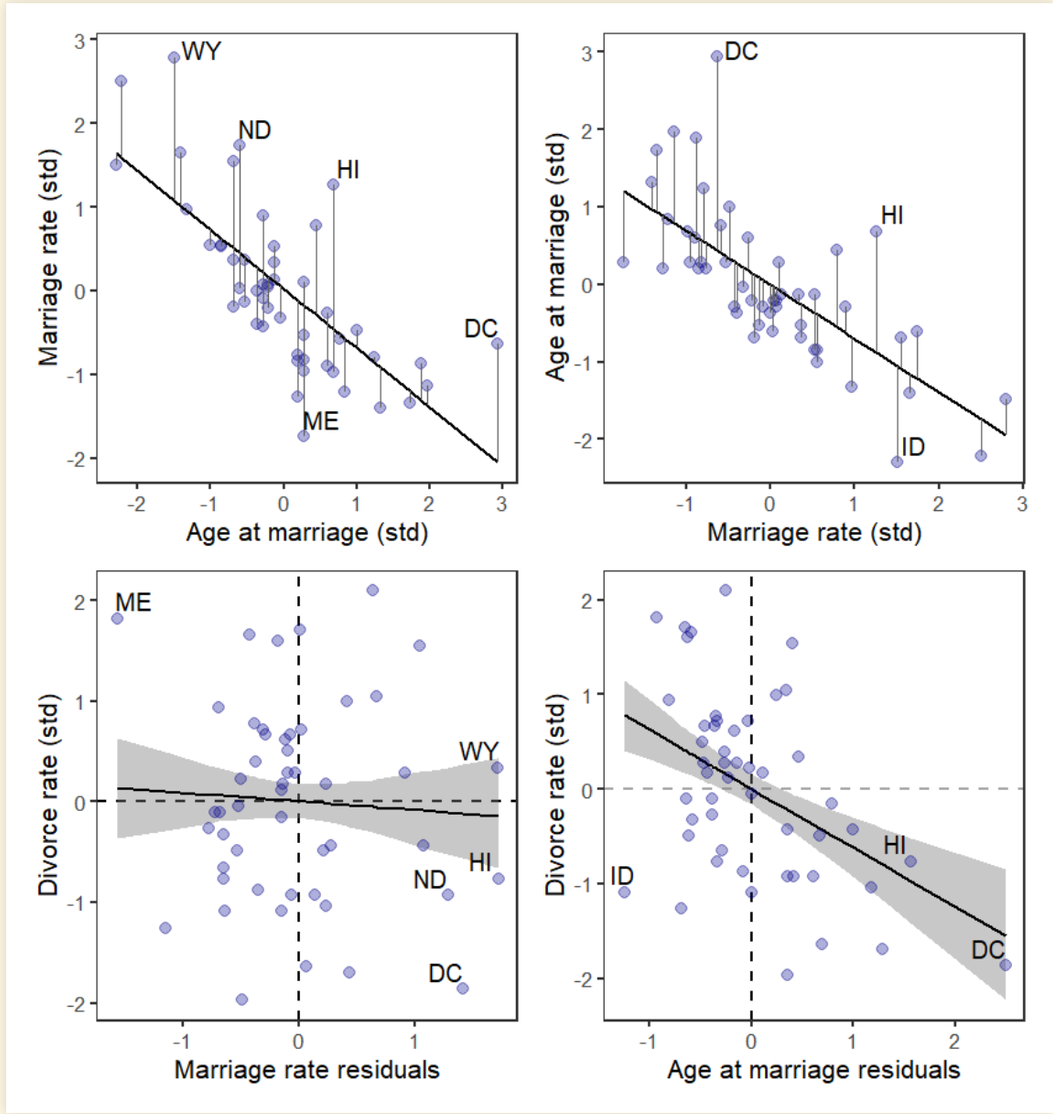


Compare Residual Plots

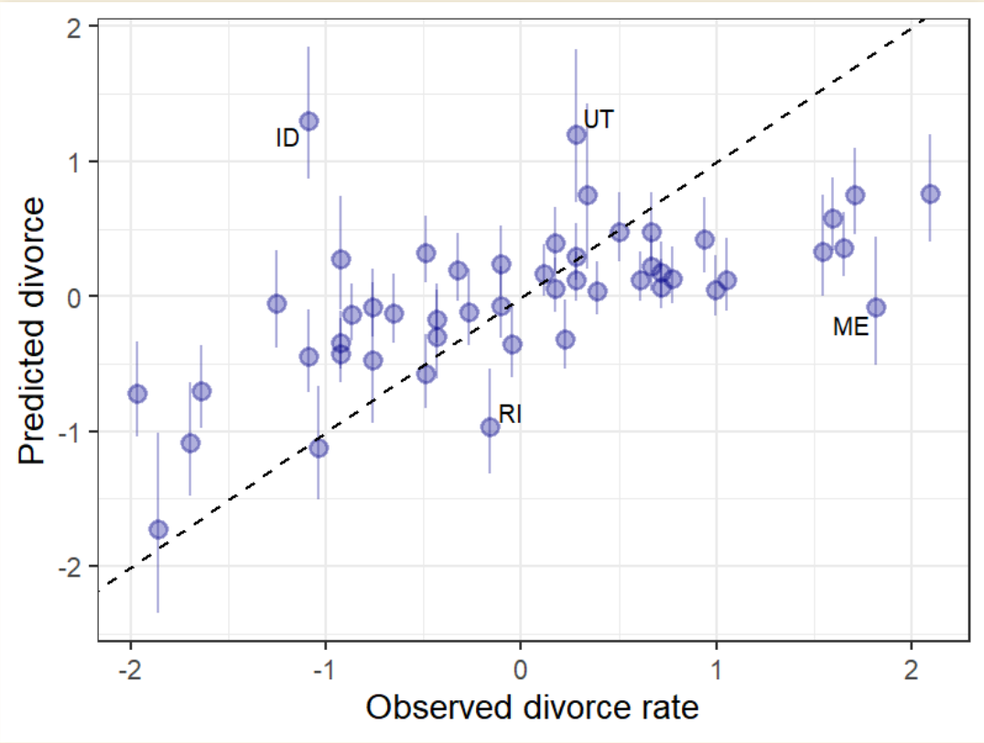
DAG 2 

vs. DAG 3 

- Look for structure in the residuals.
  - Is the trend line consistent with being flat?
    - DAG 2: Marriage rate residuals? (Yes)
    - DAG 3: Age at marriage residuals? (No)
      - Correlation implies structure the model doesn't account for.
      - A influences D, even after accounting for M
      - DAG (3) is ruled out.



Posterior Predictive Plots



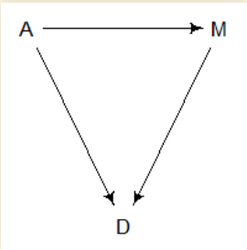
1. How well does the posterior distribution match the data?
2. When the model fails, ***how*** does it fail?

- Notice that I use `linpred_draws` (equivalent to `link`), not `predicted draws` (equivalent to `extract.samples`)

```
mu <- linpred_draws(mdl_multi, select(d, D, A, M, Loc),
                    ndraws = 1000, value = "mu")

d2 <- mu |> ungroup() |>
  summarize(estimate = mean(mu),
            lower = quantile(mu, 0.055),
            upper = quantile(mu, 0.975),
            .by = c("Loc", "D", "A", "M"))

notable_states <- c("ID", "UT", "RI", "ME")

ggplot(d2, aes(x = D, y = estimate,
               ymax = upper, ymin = lower)) +
  geom_pointrange(size = 1, color = "darkblue", alpha =
        0.3) +
  geom_text_repel(data = filter(d2, Loc %in%
        notable_states),
                  aes(label = Loc), size = 5) +
  geom_abline(slope = 1, intercept = 0, color = "black",
              linetype = "dashed") +
  labs(x = "Observed divorce rate",
       y = "Predicted divorce")
```

Counterfactual Predictions

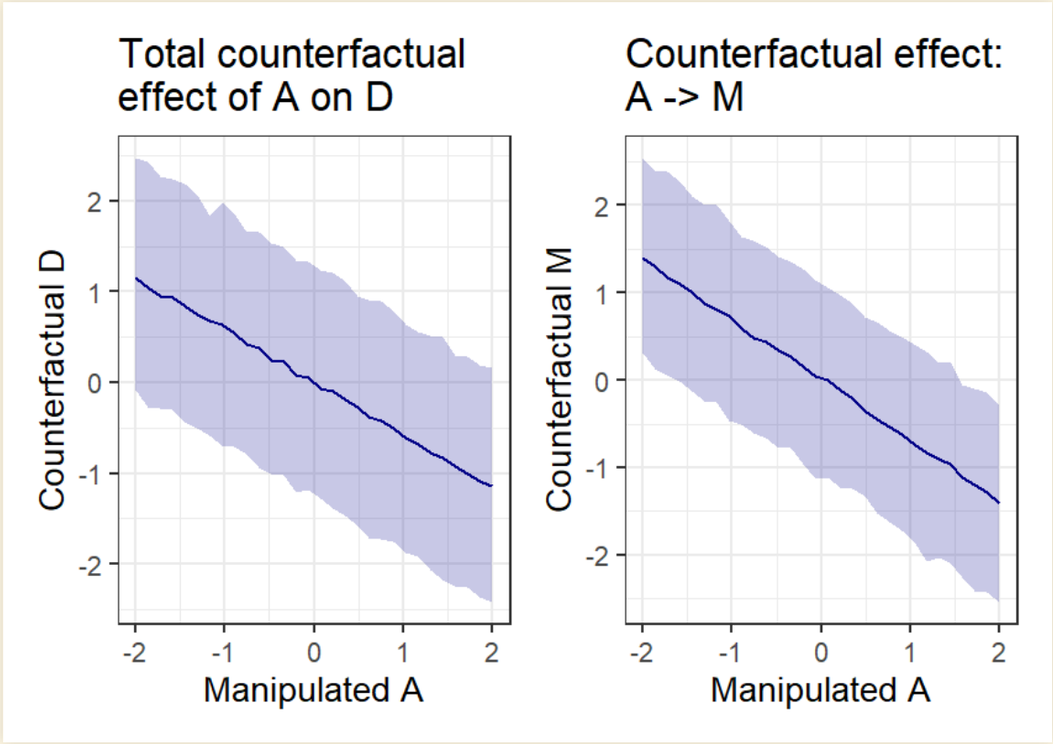
- No states have both a high median age of marriage and a high marriage rate.
- Utah: A_{std} = -2.21, M_{std} = 2.5, D_{std} = 0.28.
  - What would Utah's divorce rate be if its median age at marriage were high, and its marriage rate remained the same?
  - Use this causal model:



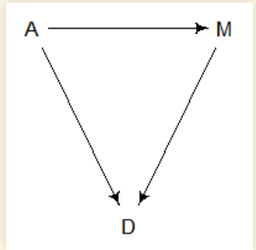
- Recipe:
  1. Pick a variable to manipulate: *intervention* variable.
  2. Define a range of values for the intervention variable.
  3. For each value of the intervention variable:
     - Sample parameters from the posterior.
     - For each sample of the posterior, use causal model to simulate the values of other variables, including the outcome.

Implementing Counterfactual Prediction

```
mdl_cfact <- quap( alist(
  ## A -> D <- M
  D ~ dnorm( mu, sigma),
  mu <- a + bM * M + bA * A,
  a ~ dnorm( 0, 0.2),
  bM ~ dnorm( 0, 0.5),
  bA ~ dnorm( 0, 0.5),
  sigma ~ dexp( 1),

  ## A -> M
  M ~ dnorm( mu_M, sigma_M),
  mu_M <- aM + bAM*A,
  aM ~ dnorm( 0, 0.2),
  bAM ~ dnorm( 0, 0.5),
  sigma_M ~ dexp( 1)
  ), data = d)
new_data <- tibble(A = seq(from = -2, to = 2, length.out =
        30))
cfact <- sim(mdl_cfact, data = new_data, vars = c("M",
        "D"))
```

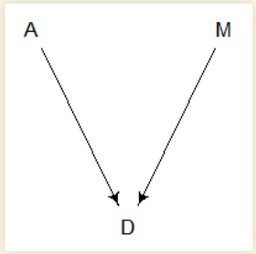


- Model DAG:

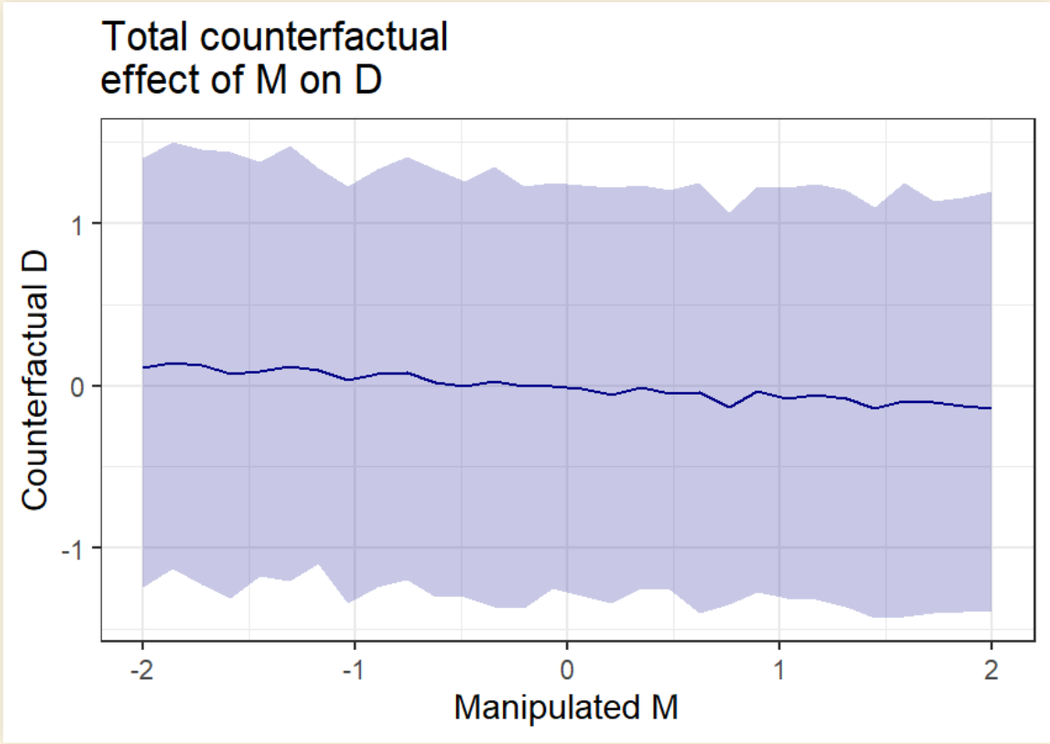


Different Counterfactual

- Now, suppose you want to explore the effect of changing M instead of A?
  - If we're going to change M, we need a different DAG:




```
new_data_2 <- tibble(M = seq(from = -2, to = 2, length.out
        = 30),
                     A = 0)
cfact_2 <- sim(mdl_cfact, data = new_data_2, vars =
        c("D"))
```



- We've simulated two controlled experiments:
  1. Vary A and account for effects of A on M.
  2. Vary M and keep A constant.