Bayesian Causal Inference

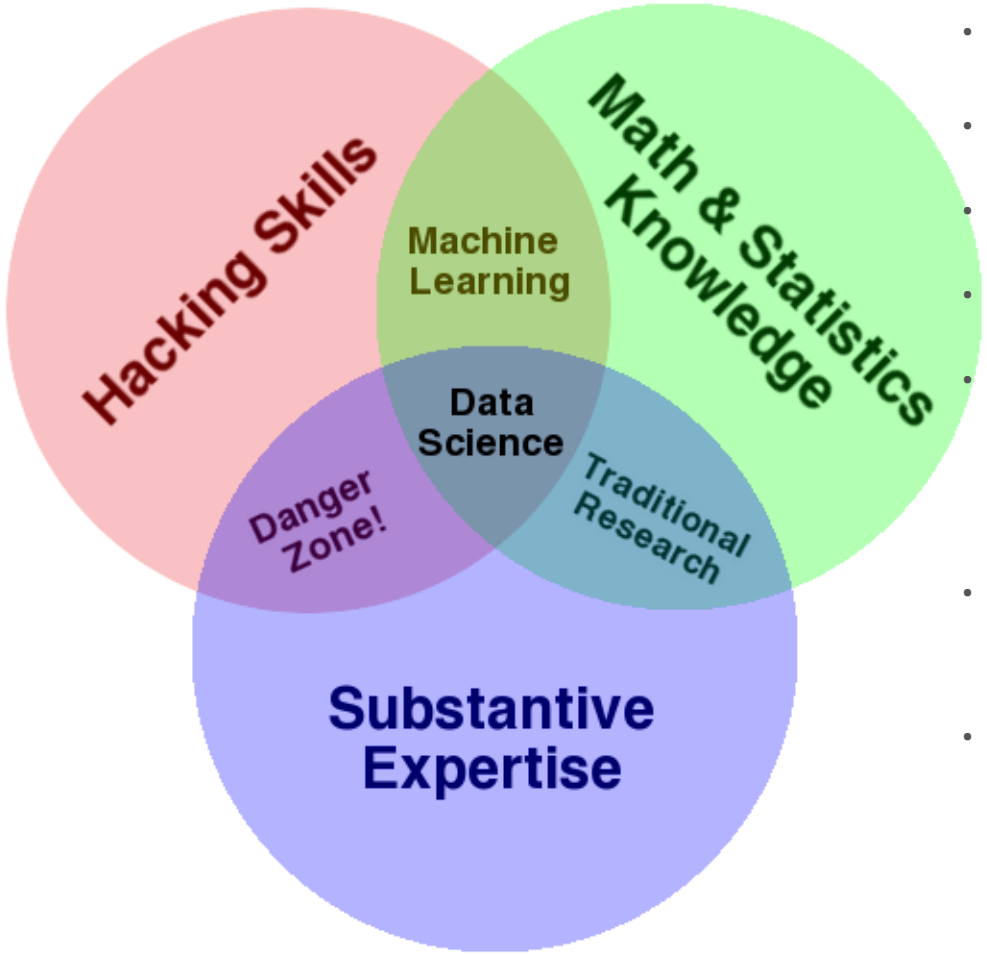
Ben Goodrich

May 09, 2022

HW1 Again

- What was estimated GDP growth in the first quarter?
- [T]he conduct of monetary policy in the United States has come to involve, at its core, crucial elements of risk management. This conceptual framework emphasizes understanding as much as possible the many sources of risk and uncertainty that policymakers face, quantifying those risks when possible, and assessing the costs associated with each of the risks. In essence, the risk-management approach to monetary policymaking is an application of Bayesian decision-making.
- Alan Greenspan, 2004, while chairing the Federal Reserve Board
- Why is the Bayesian approach not taught in economics, public policy, etc.?
- Why, if you search for “Bayesian” in the course descriptions on Albert, does little turn up?

HW2, Question 1



- What do you get if you equate . . .
- Probability
- Priors, DAGs, and Model Building
- Markov Chain Monte Carlo
- Since at least 1990, Bayesian estimation has been what data science purported to be but isn't
- Data science came along later but largely ignored Bayesian approaches
- In reality, data science programs are asymmetric with more emphasis on hacking skills and less on the others

HW2, Question 2

- Could largely use negative binomial example from Week4
- With GLMs, it is very easy to put positive prior probability on extreme events
- Drawing from the prior predictive distribution has a tendency to overflow, in which case you can do something like this in `generated quantities`

```
vector[N] eta = alpha + offset + X * beta;
for (n in 1:N) {
  real inv_mu = exp(-eta[n]);
  real lambda = gamma_rng(phi, phi * inv_mu);
  if (lambda > poisson_max) { // 2^30
    y_rep[n] = 2 * (poisson_max - 1);
  } else {
    y_rep[n] = poisson_rng(lambda);
  }
}
```

- More scientifically accurate videos have fewer views

HW2, Question 3

- Could model vaccination with a normal (percentage), beta (proportion), or binomial (count) distribution
- Should allow heterogeneity in the data-generating process by state
- $\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma)$ works better — in this case — than $\alpha_j \equiv \mu_\alpha + \sigma a_j$ with $a_j \sim \mathcal{N}(0, 1)$ because the state-wise heterogeneity is large relative to the county-level error

Graphical Causal Models Framework

- Potential Outcomes framework is widespread in economics & political science and is fine for obtaining point estimates of Average Causal Effects (ACEs) in simple experiments. It only involves probability theory and algebra / calculus.
- Graphical Causal Models framework is widespread in epidemiology and to a lesser extent, in sociology and psychology but is overkill for estimating ACEs in simple experiments
- A theorem in one framework implies a theorem in the other framework
- Directed Acyclic Graphs can serve multiple purposes:

  1. A language to communicate theories
  2. Identification Analysis: Whether a theory implies the ACE could be calculated in a population
  3. Testable Implications: What observable variables are conditionally independent, which can be investigated with data

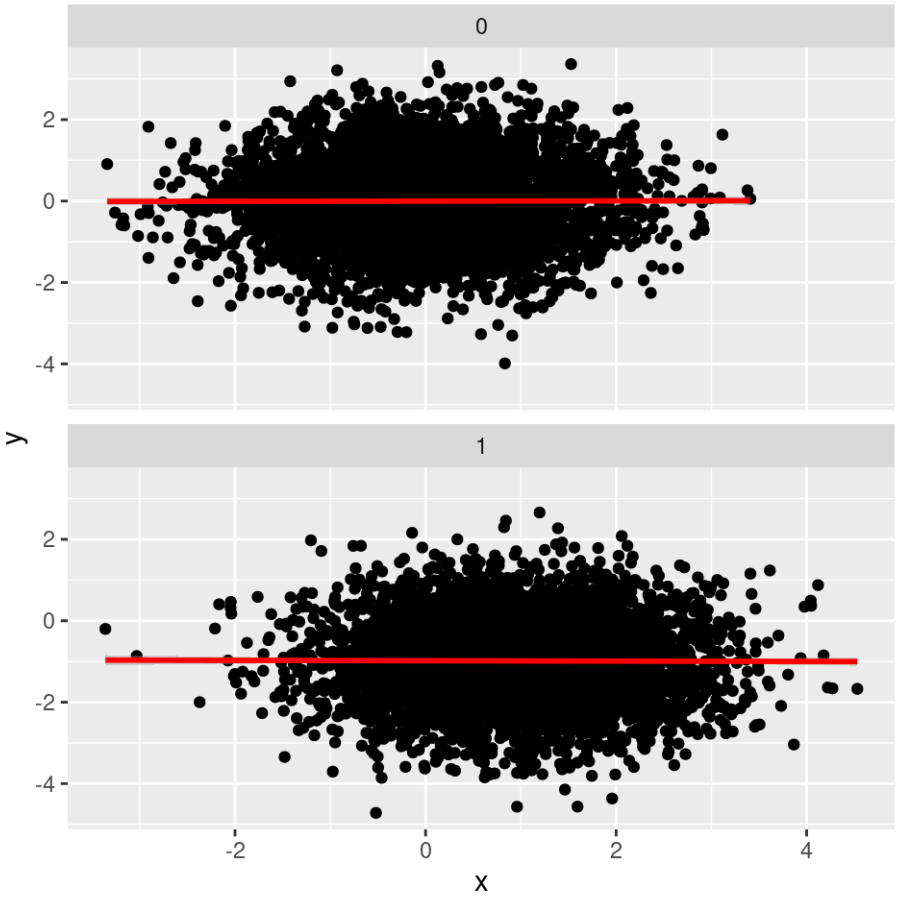
Directed Acyclic Graphs (DAGs)

- Three elements to a DAG:
  1. Variables / Nodes
  2. Arrows from an earlier (in time) node to a later node
  3. Absence of arrows between nodes, which implies an ACE is zero
- $A \rightarrow B$ means that if A were experimentally manipulated there MAY be a non-zero ACE that is assumed to be unmediated by any other variable
- Cannot start at any node, follow arrows, and get back to where you started
- DAGs are usually written w/o distributional or functional form assumptions

Fork

- $X \leftarrow \boxed{Z} \rightarrow Y$
- Like GDP* driving both GDI and GDP
- Stratifying on Z implies $X \perp Y \mid Z$
- Not stratifying on Z implies X and Y are marginally dependent, i.e $f\left(\not{\perp} \bigcap x \bigcap y\right) \neq f\left(x\right) \times f\left(y\right)$

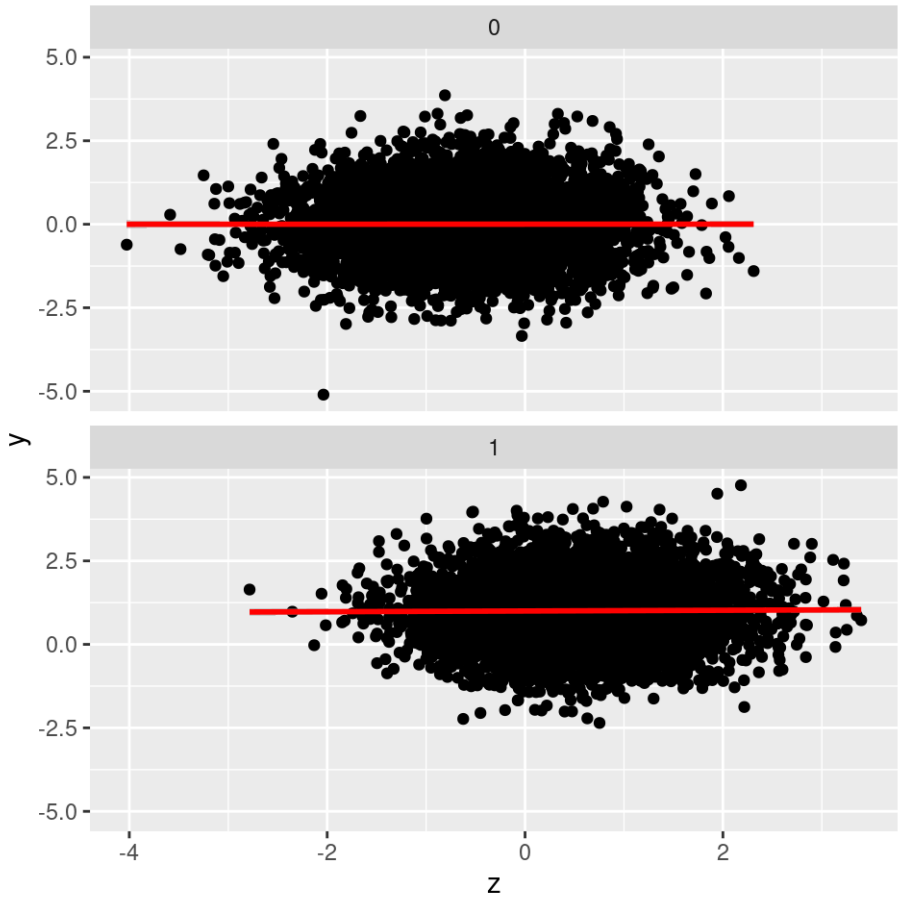
```
tibble(z = rbinom(10^4, size = 1,
                  prob = 0.5),
       x = rnorm( 10^4, mean =  z),
       y = rnorm( 10^4, mean = -z)) %>%
  ggplot(aes(x, y)) + geom_point() +
  facet_wrap(~as.factor(z), ncol = 1) +
  geom_smooth(method = "lm", color = "red")
```



Pipe (or chain)

- $Z \overset{\beta_1}{\rightarrow} X \overset{\beta_2}{\rightarrow} Y$
- Stratifying on X implies $Z \perp Y \mid X$
- Not stratifying on X implies Z and Y are marginally dependent, i.e $f\left(z \bigcap \not{x} \bigcap y\right) \neq f\left(z\right) \times f\left(y\right)$
- Iff the relationships are linear the marginal effect of Z on Y is $\beta_1 \beta_2$

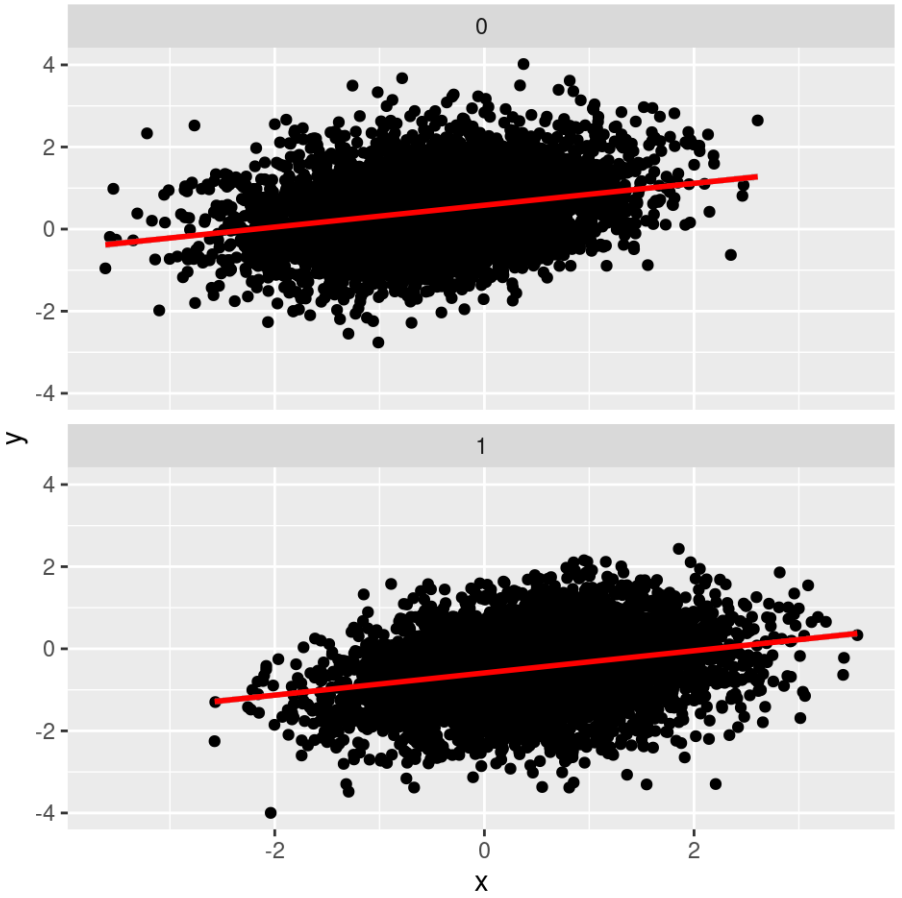
```
tibble(z = rnorm( 10^4),
       x = rbinom(10^4, size = 1,
                  prob = pnorm(z)),
       y = rnorm( 10^4, mean = x)) %>%
  ggplot(aes(z, y)) + geom_point() +
  facet_wrap(~as.factor(x), ncol = 1) +
  geom_smooth(method = "lm", color = "red")
```



Collider

- $X \rightarrow Z \leftarrow Y$
- Not stratifying on Z implies X and Y are marginally independent, i.e $f\left(x \bigcap \not\searrow \bigcap y\right) = f\left(x\right) \times f\left(y\right)$
- Stratifying on Z implies X and Y are conditionally dependent, i.e $f\left(x \bigcap y \mid z\right) \neq f\left(x \mid z\right) \times f\left(y \mid z\right)$

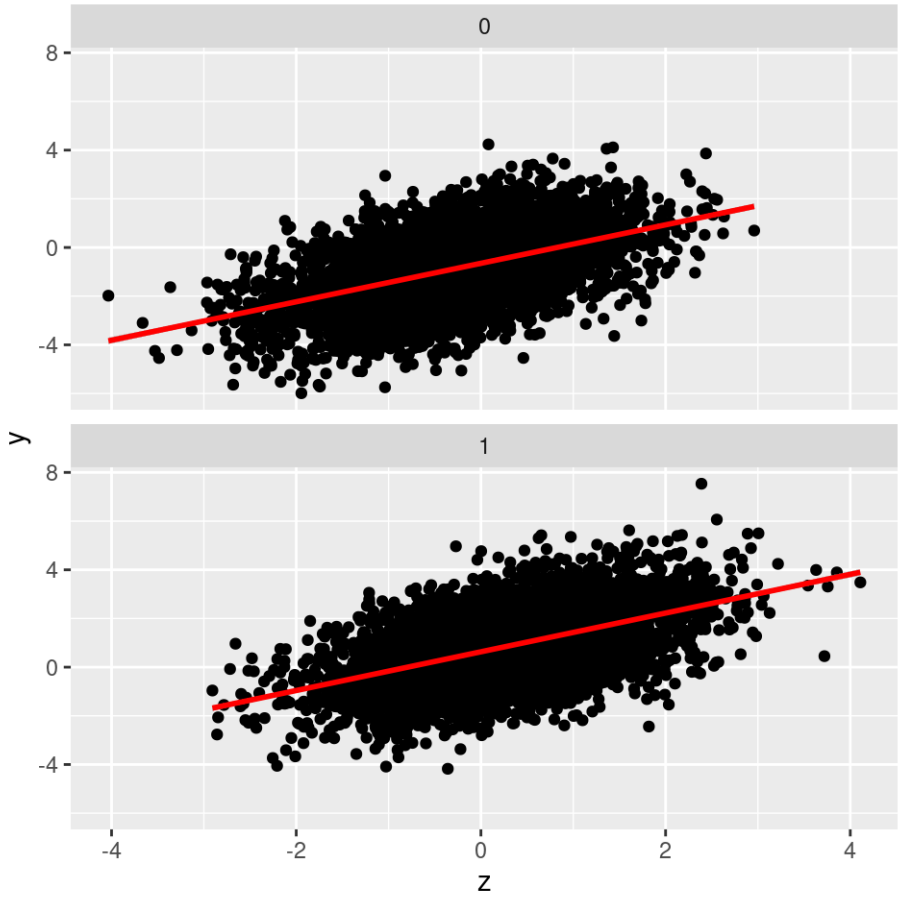
```
tibble(x = rnorm( 10^4),
       y = rnorm( 10^4),
       z = rbinom(10^4, size = 1,
                  prob = pnorm(x - y))) %>%
  ggplot(aes(x, y)) + geom_point() +
  facet_wrap(~as.factor(z), ncol = 1) +
  geom_smooth(method = "lm", color = "red")
```



Descendant

- $Z \overset{\beta_1}{\to} X \overset{\beta_2}{\to} Y \to R$
- Not stratifying on R implies effect of Z on Y is $\beta_1 \beta_2$ (assuming linearity)
- Stratifying on R prevents you from recovering $\beta_1 \beta_2$ via regression

```
tibble(z = rnorm( 10^4),
       x = rnorm( 10^4, mean = z),
       y = rnorm( 10^4, mean = x),
       r = rbinom(10^4, size = 1,
                  prob = pnorm(y / 2))) %>%
  ggplot(aes(z, y)) + geom_point() +
  facet_wrap(~as.factor(r), ncol = 1) +
  geom_smooth(method = "lm", color = "red")
```



Blocking / Closing a Path and d-separation

- A path is a sequence of connected nodes, regardless of the arrows' direction
  1. Causal Path: A path that exclusively follows the arrows
  2. Non-causal Path: Any other path
- To block a path, either
  1. Condition on a noncollider along a path: $A \leftarrow \boxed{\text{C}} \rightarrow B$
  2. Refrain from conditioning on a (descendant of a) collider along a path: $A \rightarrow C \leftarrow B$
- Two variables are "d-separated" iff all paths between them are blocked by conditioning on a possibly empty set of variables $\{Z\}$, in which case they are conditionally independent
- Two variables that are not "d-separated" are not conditionally independent

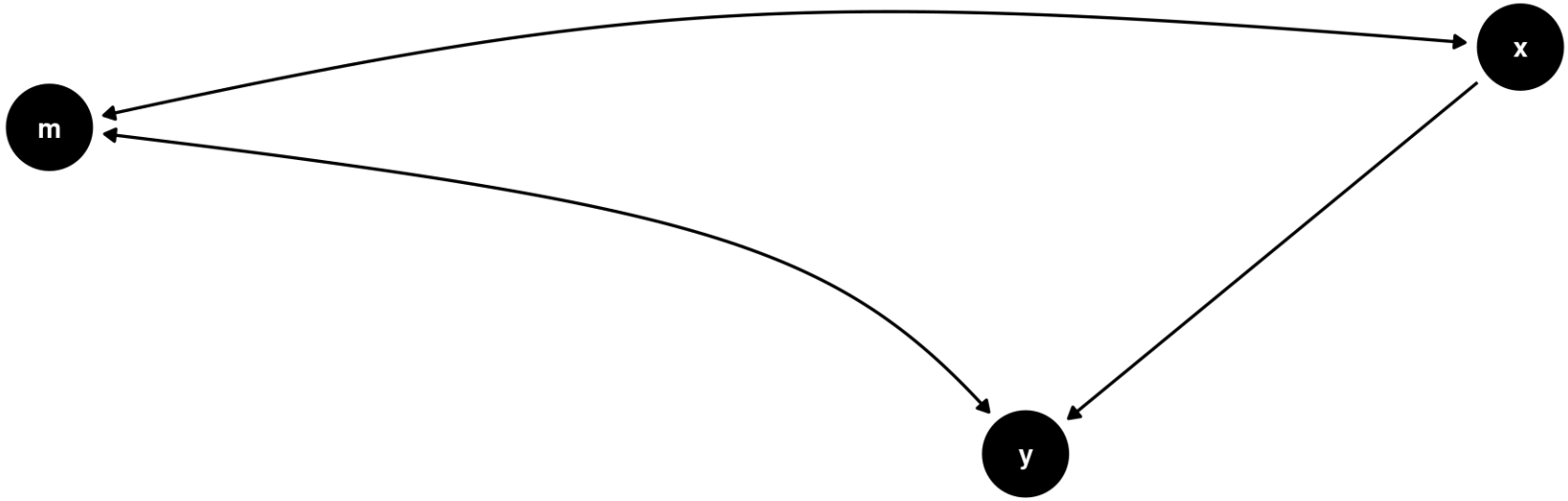
Adjustment and Backdoor Criteria

- Adjustment Criterion is satisfied iff
  1. All non-causal paths from X to Y blocked by a (possibly empty) set of variables $\{Z\}$
  2. No variable in $\{Z\}$ lies on or descends from any causal path from X to Y
- If the adjustment criterion is satisfied, ACE of X on Y is identified and can be consistently estimated with a (possibly weighted) difference in means
- Backdoor Criterion is satisfied iff
  1. No element of $\{Z\}$ is a descendant of Y
  2. Some element of $\{Z\}$ blocks all "backdoor" paths, i.e. those starting with $\rightarrow X$
- Backdoor Criterion implies Adjustment Criterion but not vice versa

Dagitty

- There is a website, that implements the most common and useful identification strategies for DAGs and a similar R package, dagitty:

```
library(dagitty); g <- dagitty( "dag{ x -> y ; x <-> m <-> y }" )
library(ggdag); ggdag(g) + theme_void()
```

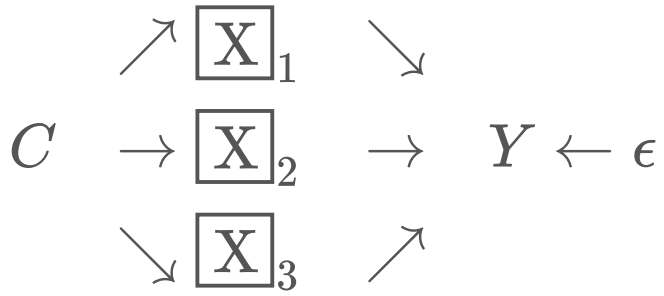


Double-headed Arrows

- The dagitty R package and a lot of authors use $A \leftrightarrow B$ as a shorthand for a $A \leftarrow U \rightarrow B$ with U unobserved or for $\begin{array}{ccc} X & \leftrightarrow & Y \\ \downarrow & \swarrow & \\ \boxed{\text{C}} & & \end{array}$ due to conditioning
- Any path involving a $\leftrightarrow$ is a non-causal path

Contrasts with Common Practice

- Supervised learning generally does not utilize DAGs and is not capable of estimating causal effects. Models are scored on how well they predict the outcome (in the testing set), which is better if you condition on descendants of the outcome, mediators, some colliders, etc.
- Regressions generally do not estimate the causal effect of all covariates and rarely estimate the causal effect of any covariate. They implicitly correspond to a DAG like (although the error term is often omitted):



- Non-Bayesians tend to write DAGs without distributions, parameters, & priors

CausalQueries R Package

- All DAGs where all observed nodes are binary have a (multinomial) likelihood function that can be combined with priors on the “type” parameters to perform Bayesian inference
- The CausalQueries R Package takes a DAG, priors, and data and uses Stan to draw from the posterior distribution of the parameters given the data & priors
- Useful for describing your beliefs about (not necessarily average) causal effects in many situations that are not simple experiments

Dataset from Bertrand and Mullainathan (2004)

- Resumes were created for a fictitious person applying for an entry-level job but the name at the top of the resume was randomized to make the company think the applicant was probably black / white / male / female. The outcome is whether the company called the applicant to schedule an interview, etc.

```
data(resume, package = "qss") # remotes::install_github("kosukeimai/qss-package")
resume <- mutate(resume, white = race == "white", female = sex == "female")
resume_grouped <- group_by(resume, white, female, call) %>%
  summarize(n = n())
glimpse(resume_grouped)
```

```
## Rows: 8
## Columns: 4
## Groups: white, female [4]
## $ white  <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE
## $ female <lgl> FALSE, FALSE, TRUE, TRUE, FALSE, FALSE, TRUE, TRUE
## $ call   <int> 0, 1, 0, 1, 0, 1, 0, 1
## $ n      <int> 517, 32, 1761, 125, 524, 51, 1676, 184
```

Basic Frequentist Inference

```
prop.test(matrix(rev(resume_grouped$n), ncol = 2, byrow = TRUE,
                 dimnames = list(c("white_female", "white_male",
                                   "black_female", "black_male"),
                                 c("call", "no call"))))
```

```
##
##  4-sample test for equality of proportions without continuity correction
##
## data:  matrix(rev(resume_grouped$n), ncol = 2, byrow = TRUE, dimnames = list(c("white_femal
## X-squared = 17.867, df = 3, p-value = 0.0004686
## alternative hypothesis: two.sided
## sample estimates:
##     prop 1     prop 2     prop 3     prop 4
## 0.09892473 0.08869565 0.06627784 0.05828780
```

- Could get point estimates, but many questions prohibited by Frequentism, e.g. “How sure are you that companies favor white / male over black / female?”

Principal (not principled) Stratification

- If X and $Y(X)$ are both binary, then there are $2^2 = 4$ types of observations:

Function	Rubin	Synonym	This Example
$Y(x) \neq x$	Defiers	**A**dverse	Interview iff white
$Y(x) = x$	Compliers	**B**eneficial	Interview iff black
$Y(x) = 0 \forall x$	Never takers	**C**hronic	No interview regardless
$Y(x) = 1 \forall x$	Always takers	**D**estined	Interview regardless

↓ What We Observe →	No interview	Interview
White	$\Longrightarrow$ **B** or **C**	$\Longrightarrow$ **A** or **D**
Black	$\Longrightarrow$ **A** or **C**	$\Longrightarrow$ **B** or **D**

Dirichlet Distribution

- Dirichlet distribution is over the parameter space of PMFs — i.e. $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$ — and the Dirichlet PDF is $f\left(\boldsymbol{\pi} \mid \boldsymbol{\alpha}\right) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$ where $\alpha_k \geq 0 \, \forall k$ and the multivariate Beta function is $B\left(\boldsymbol{\alpha}\right) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma\left(\prod_{k=1}^{K} \alpha_k\right)}$ where $\Gamma\left(z\right) = \frac{1}{z} \prod_{n=1}^{\infty} \frac{\left(1 + \frac{1}{n}\right)^n}{1 + \frac{z}{n}} = \int_0^{\infty} u^{z-1} e^{-u} du$ is the Gamma function
- $\mathbb{E}\pi_i = \frac{\alpha_i}{\sum_{k=1}^{K} \alpha_k} \, \forall i$ and the mode of π_i is $\frac{\alpha_i - 1}{-1 + \sum_{k=1}^{K} \alpha_k}$ if $\alpha_i > 1$
- Iff $\alpha_k = 1 \, \forall k$, $f\left(\boldsymbol{\pi} \middle| \boldsymbol{\alpha} = \mathbf{1}\right)$ is constant over Θ (simplexes)
- Beta distribution is a special case of the Dirichlet where $K = 2$
- Marginal and conditional distributions for subsets of $\boldsymbol{\pi}$ are also Dirichlet

Conditioning on the Resume Data

```
library(CausalQueries)
model <- make_model("white -> call <- female")
model %>% get_parameter_matrix %>% nrow # default priors are flat Dirichlet
```

```
## [1] 20
```

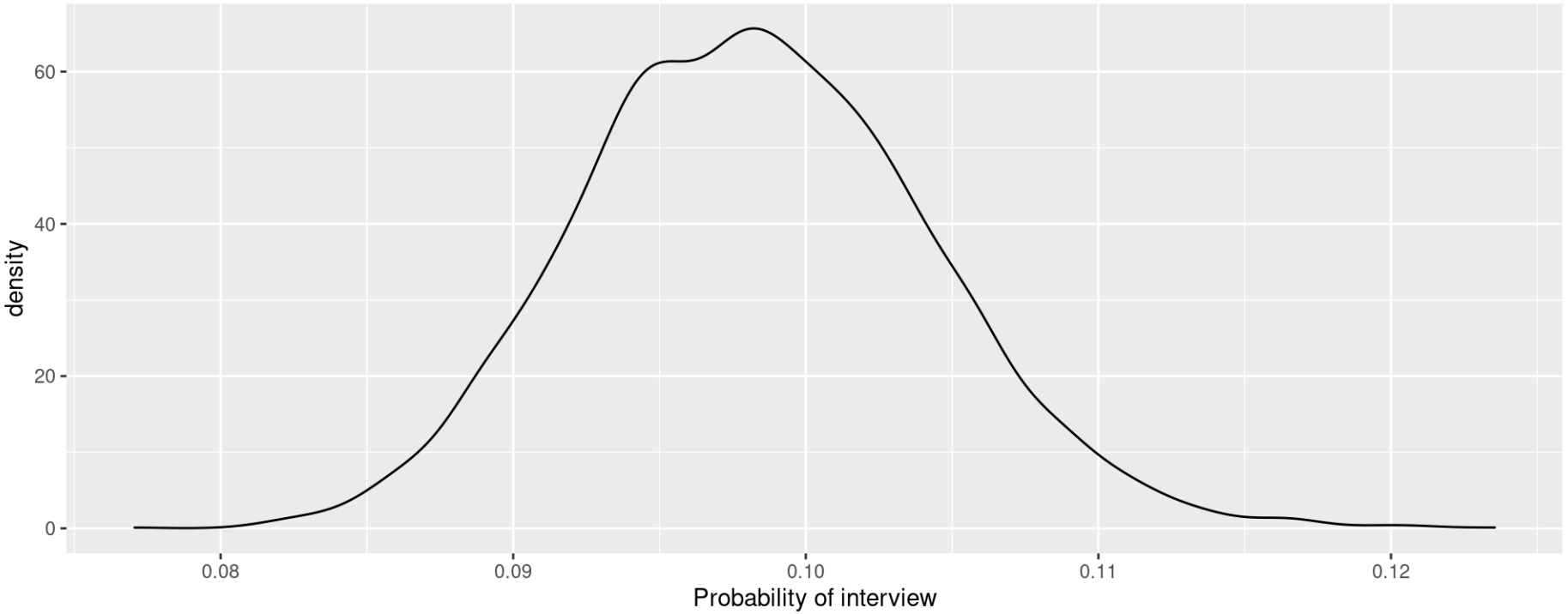
```
model <- update_model(model, seed = 12345, data = resume)
```

```
query_model(model, using = "posteriors", queries = # can also ask about priors
              c(race_ACE = "call[white  = 0] - call[white  = 1]",
                sex_ACE  = "call[female = 0] - call[female = 1]",
                Pr = "call[white = 0, female = 1] > call[white = 1, female = 1]"))
```

```
##      Query Given      Using   mean    sd
## 1 race_ACE     - posteriors -0.030 0.008
## 2  sex_ACE     - posteriors -0.004 0.009
## 3       Pr     - posteriors  0.036 0.012
```

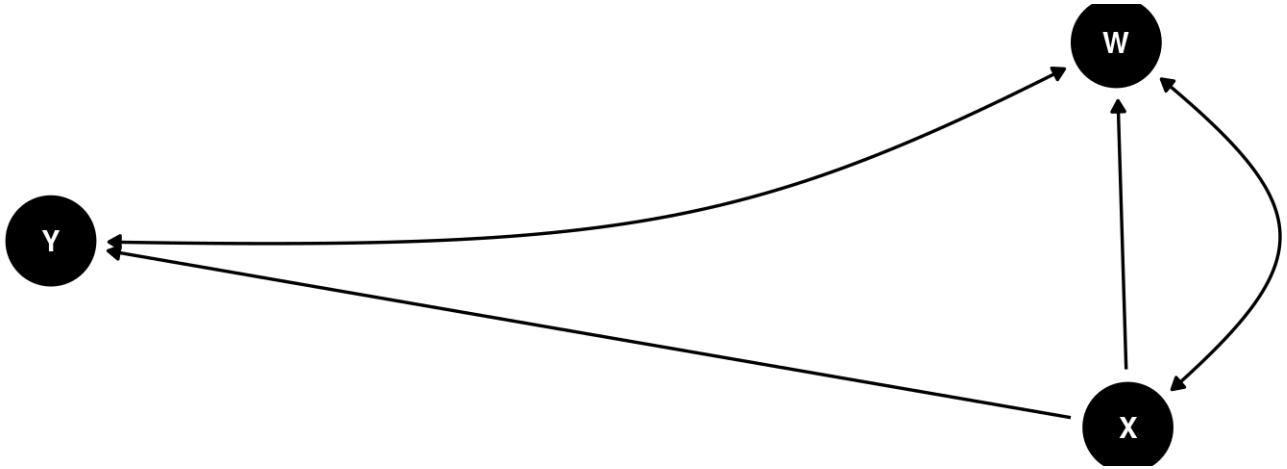
Counterfactual Distributional Queries

```
query_distribution(model, using = "posteriors", given = "white == 0", # condition
                   query = "call[white = 1]") %>% as.data.frame %>%   # counterfactual
  ggplot() + geom_density(aes(`.`)) + xlab("Probability of interview")
```



A Model with Confounding

```
model <- make_model("W <- X -> Y") %>% set_confound(list("W <-> X", "W <-> Y"))
CausalQueries:::translate_dagitty(model) %>% ggdag + theme_void()
```




```
adjustmentSets(CausalQueries:::translate_dagitty(model), exposure = "X", outcome = "Y")
```

```
##  {}
```

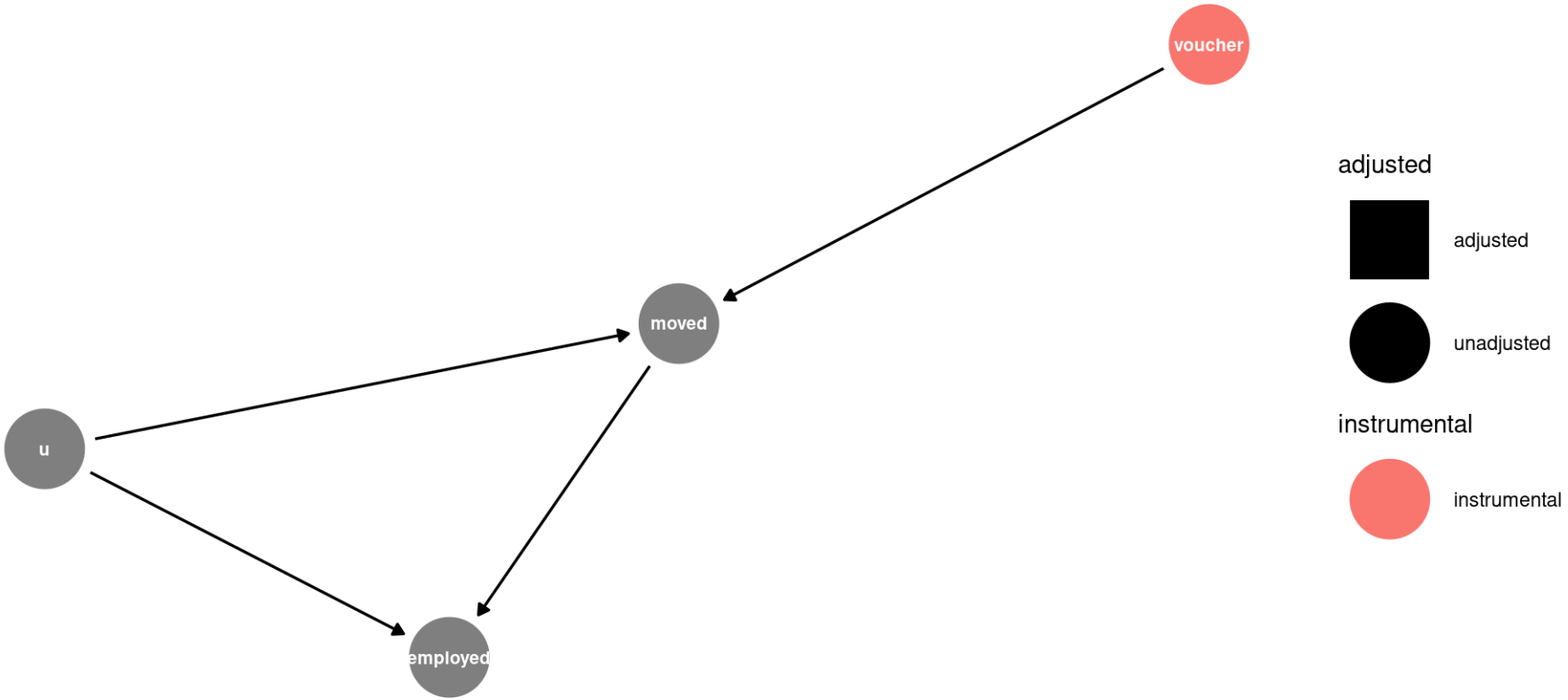
Conditional vs. Joint Modeling

- If you do not condition on W in the previous model, the ACE of X on Y is identified and easy to estimate via regression, differences in means, etc.
- If you do condition on W, the ACE of X on Y is not identified and cannot be estimated correctly via regression, differences in means, etc.
- If you use CausalQueries, the model is of W, X, and Y jointly, which does not condition on the collider, W
- You can subsequently do things like

```
query_distribution(model, using = "posteriors", given = "W == 0", # condition
                   query = "Y[X = 1] - Y[X = 0]") # conditional ACE
```

Instrumental Variables with MTO Study

```
dagify(employed ~ moved + u, moved ~ voucher + u, exposure = "moved", outcome = "employed",
       latent = "u") %>%  ggdag_instrumental(text_size = 2.75) + theme_void()
```



Frequentist Instrumental Variable Estimation

- When everything is binary, can point estimate the (local) average treatment effect of moving (because of the MTO voucher) as

$$\widehat{\beta} = \frac{\text{cov}\left(\text{voucher}, \text{employed}\right)}{\text{cov}\left(\text{voucher}, \text{moved}\right)}$$

- The sampling distribution of this Wald estimator across datasets of size N can be terrible because both the numerator and the denominator are both real random variables
- Cauchy distribution is the ratio of standard normal variates and has no expectation; Frequentist instrumental variable estimators are similar
- Major problem if you want to pretend MLEs characterize a normal posterior distribution
- Actual posterior distribution does not involve any ratios, although it may be far from normal

Angrist and Kreuger (1991) Data

```
ROOT <- "http://higheredbcs.wiley.com/legacy/college/"
PATH <- "lancaster/1405117206/datasets/AKdata.zip"
if (!file.exists("AKdata.zip")) {
  download.file(paste0(ROOT, PATH), destfile = "AKdata.zip")
  unzip("AKdata.zip")
}
AKdata <- read.table("AKdata.txt", header = FALSE, skip = 4,
                     col.names = c("ID", "log_wage", "schooling", "birth_quarter", "age"))
glimpse(AKdata)
```

```
## Rows: 35,805
## Columns: 5
## $ ID            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,…
## $ log_wage      <dbl> 5.9525, 5.3549, 5.6649, 5.7900, 5.0366, 5.4309, 6.4380, 5.9012, 5.…
## $ schooling     <int> 12, 12, 11, 9, 16, 7, 14, 12, 8, 12, 13, 12, 11, 0, 8, 8, 12, 16, …
## $ birth_quarter <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ age           <int> 42, 45, 45, 45, 37, 45, 39, 13, 37, 45, 45, 13, 45, 45, 37, 1, 45,…
```

Generative Model from McElreath section 14.3

- “The answer is actually pretty simple. We just use the generative model.”

$$q_n \sim \mathcal{C}\left(\left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right]^\top\right)$$

$$[\lambda_1, \lambda_2, \lambda_3, \lambda_4]^\top \sim ???$$

$$\sigma_s \sim ???$$

$$s_n \sim \mathcal{N}\left(\lambda_{q_n}, \sigma_s\right)$$

$$\sigma_u \sim ???$$

$$u_n \sim \mathcal{N}\left(0, \sigma_u\right)$$

$$\alpha \sim ???$$

$$\beta \sim ???$$

$$\mu_n \equiv \alpha + \beta s_n + u_n$$

$$\sigma_y \sim ???$$

$$y_n \sim \mathcal{N}\left(\mu_n, \sigma_y\right)$$

Stan Program

```
data {              // saved as "iv.stan"
  int<lower = 0> N; // number of observations
  int<lower = 1, upper = 4> q[N];       // quarter
  vector[N] s;                          // schooling
  vector[N] y;                          // outcome
  int<lower = 0, upper = 1> prior_only;
  vector[3] m;
  vector<lower = 0>[3] scale;
  real<lower = 0> r;
}
parameters {
  vector[4] lambda;
  vector<lower = 0>[2] sigma;
  real<lower = -1, upper = 1> rho;
  real alpha;
  real beta;
}
```

```
model {
  if (!prior_only) {
    vector[N] s_ = lambda[q];
    vector[N] y_ = alpha + beta * s
                + sigma[2] / sigma[1] * rho * (s - s_);
    target += normal_lpdf(s | s_, sigma[1]);
    target += normal_lpdf(y | y_,
                          sigma[2] * sqrt(1 - square(rho)));
  }
  target += normal_lpdf(lambda | m[1], scale[1]);
  target += normal_lpdf(alpha  | m[2], scale[2]);
  target += normal_lpdf(beta   | m[3], scale[3]);
  target += exponential_lpdf(sigma | r);
  // implicit: rho ~ uniform(0,1)
}
```

Calling the Stan Program

```
library(rstan)
stan_data <- with(AKdata, list(N = nrow(AKdata), q = birth_quarter,
                               s = schooling - mean(schooling), y = log_wage,
                               prior_only = FALSE, m = c(0, log(400), 0),
                               scale = c(1, 3, 1), r = 1))

post_iv <- stan("iv.stan", data = stan_data, seed = 20220504)
```

Results

```
print(post_iv, pars = "lp__", include = FALSE)
```

```
## Inference for Stan model: iv.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##            mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## lambda[1] -0.03    0.00 0.03 -0.09 -0.05 -0.03 -0.01  0.03  2866    1
## lambda[2] -0.01    0.00 0.03 -0.07 -0.03  0.00  0.02  0.06  2155    1
## lambda[3] -0.04    0.00 0.03 -0.10 -0.06 -0.04 -0.02  0.02  2717    1
## lambda[4]  0.08    0.00 0.03  0.01  0.06  0.08  0.10  0.14  1830    1
## sigma[1]   3.13    0.00 0.01  3.10  3.12  3.13  3.13  3.15  5521    1
## sigma[2]   0.68    0.00 0.08  0.62  0.63  0.65  0.70  0.93   473    1
## rho       -0.24    0.01 0.29 -0.74 -0.45 -0.25 -0.05  0.35   691    1
## alpha      5.90    0.00 0.00  5.89  5.89  5.90  5.90  5.90  3954    1
## beta       0.13    0.00 0.07 -0.01  0.08  0.12  0.17  0.29   576    1
##
## Samples were drawn using NUTS(diag_e) at Wed May  4 14:40:17 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```