

# Methods 4 - 10

**Chris Mathys**



BSc Programme in Cognitive Science

Spring 2022

# Generalized Linear Models

**Linear Models:** Expected value is additive (“linear”) combination of parameters

$$Y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_X X_i + \beta_Z Z_i$$

**Generalized Linear Models:**  
Expected value is **some function** of an additive combination of parameters

$$Y_i \sim \text{Bernoulli}(p_i)$$
$$f(p_i) = \alpha + \beta_X X_i + \beta_Z Z_i$$

## Generalized Linear Models:

Expected value is **some function** of an additive combination of parameters

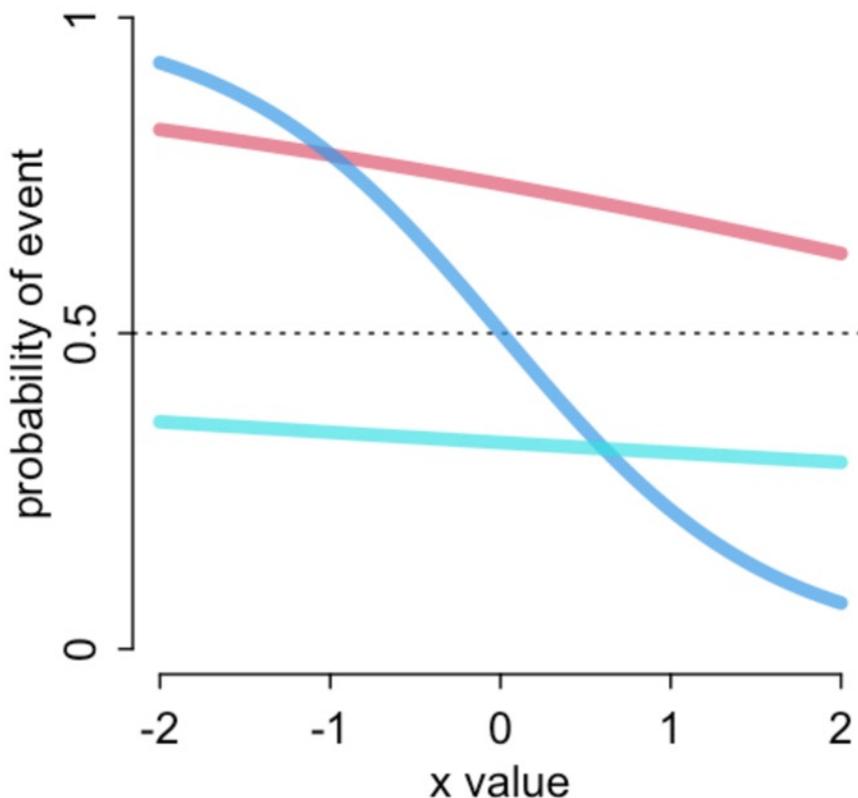
Uniform changes in predictor not uniform changes in prediction

All predictor variables interact, moderate one another

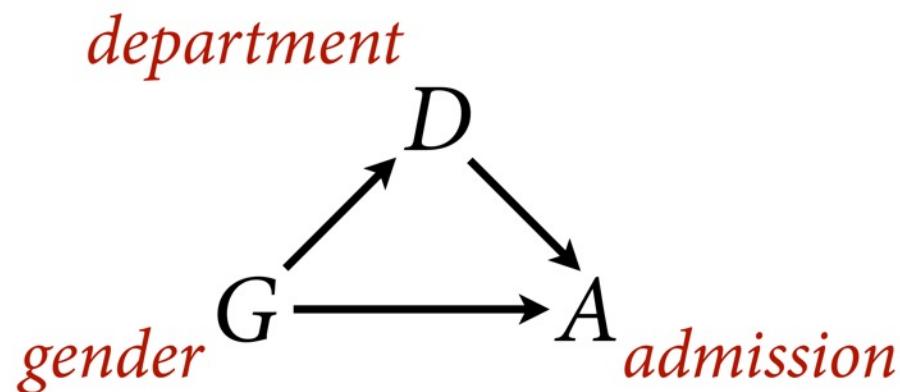
Influences predictions & uncertainty of predictions

$$\text{logit}(p_i) = \alpha + \beta x_i$$

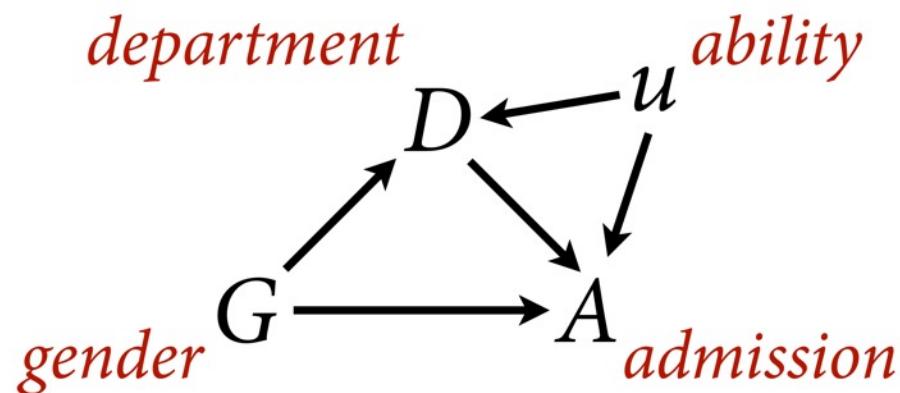
$$n = 0$$



# Confounded Admissions



# Confounded Admissions

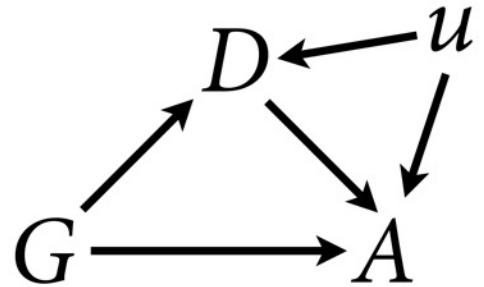


If we stratify by D, we open the non-causal path from G-D-u-A which is a huge problem

```

set.seed(17)
N <- 2000 # number of applicants
# even gender distribution
G <- sample( 1:2 , size=N , replace=TRUE )
# sample ability, high (1) to average (0)
u <- rbern(N,0.1)
# gender 1 tends to apply to department 1, 2 to 2
# and G=1 with greater ability tend to apply to 2 as well
D <- rbern( N , ifelse( G==1 , u*0.5 , 0.8 ) ) + 1
# matrix of acceptance rates [dept,gender]
accept_rate_u0 <- matrix( c(0.1,0.1,0.1,0.3) , nrow=2 )
accept_rate_u1 <- matrix( c(0.2,0.3,0.2,0.5) , nrow=2 )
# simulate acceptance
p <- sapply( 1:N , function(i)
  ifelse( u[i]==0 , accept_rate_u0[D[i],G[i]] ,
  accept_rate_u1[D[i],G[i]] ) )
A <- rbern( N , p )

```

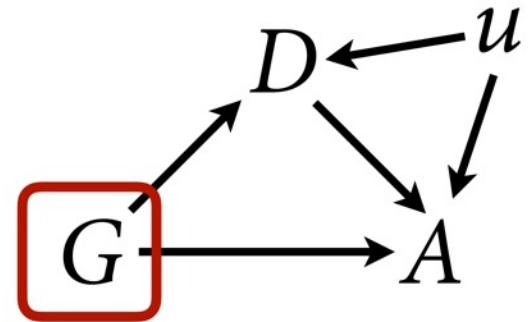


how to simulate: start with the points that has no arrows going into them. In this case: G and u.  
u is defined as a bernouilli distribution meaning that there's a 10% chance vs 90% that something happens.

```

set.seed(17)
N <- 2000 # number of applicants
# even gender distribution
G <- sample( 1:2 , size=N , replace=TRUE )
# sample ability, high (1) to average (0)
u <- rbern(N,0.1)
# gender 1 tends to apply to department 1, 2 to 2
# and G=1 with greater ability tend to apply to 2 as well
D <- rbern( N , ifelse( G==1 , u*0.5 , 0.8 ) ) + 1
# matrix of acceptance rates [dept,gender]
accept_rate_u0 <- matrix( c(0.1,0.1,0.1,0.3) , nrow=2 )
accept_rate_u1 <- matrix( c(0.2,0.3,0.2,0.5) , nrow=2 )
# simulate acceptance
p <- sapply( 1:N , function(i)
  ifelse( u[i]==0 , accept_rate_u0[D[i],G[i]] ,
  accept_rate_u1[D[i],G[i]] ) )
A <- rbern( N , p )

```



We define  $G$  as above:

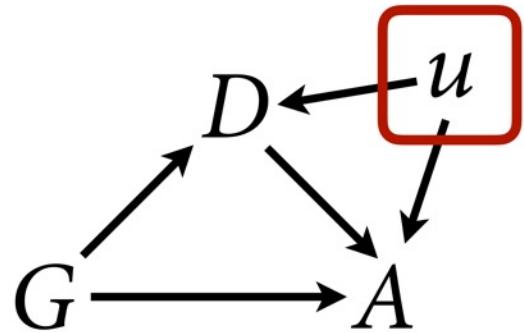
$u$  has a differential effect depending on  $G$ .

The outcome of the bernoulli is 0 and 1.

```

set.seed(17)
N <- 2000 # number of applicants
# even gender distribution
G <- sample( 1:2 , size=N , replace=TRUE )
# sample ability, high (1) to average (0)
u <- rbern(N,0.1)
## gender 1 tends to apply to department 1, 2 to 2
# and G=1 with greater ability tend to apply to 2 as well
D <- rbern( N , ifelse( G==1 , u*0.5 , 0.8 ) ) + 1
# matrix of acceptance rates [dept,gender]
accept_rate_u0 <- matrix( c(0.1,0.1,0.1,0.3) , nrow=2 )
accept_rate_u1 <- matrix( c(0.2,0.3,0.2,0.5) , nrow=2 )
# simulate acceptance
p <- sapply( 1:N , function(i)
  ifelse( u[i]==0 , accept_rate_u0[D[i],G[i]] ,
  accept_rate_u1[D[i],G[i]] ) )
A <- rbern( N , p )

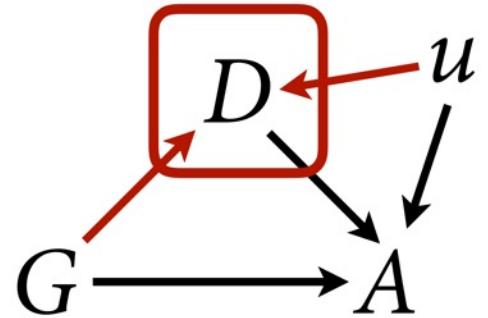
```



```

set.seed(17)
N <- 2000 # number of applicants
# even gender distribution
G <- sample( 1:2 , size=N , replace=TRUE )
# sample ability, high (1) to average (0)
u <- rbern(N,0.1)
# gender 1 tends to apply to department 1, 2 to 2
# and G-1 with greater ability tend to apply to 2 as well
D <- rbern( N , ifelse( G==1 , u*0.5 , 0.8 ) ) + 1
# matrix of acceptance rates [dept,gender]
accept_rate_u0 <- matrix( c(0.1,0.1,0.1,0.3) , nrow=2 )
accept_rate_u1 <- matrix( c(0.2,0.3,0.2,0.5) , nrow=2 )
# simulate acceptance
p <- sapply( 1:N , function(i)
  ifelse( u[i]==0 , accept_rate_u0[D[i],G[i]] ,
  accept_rate_u1[D[i],G[i]] ) )
A <- rbern( N , p )

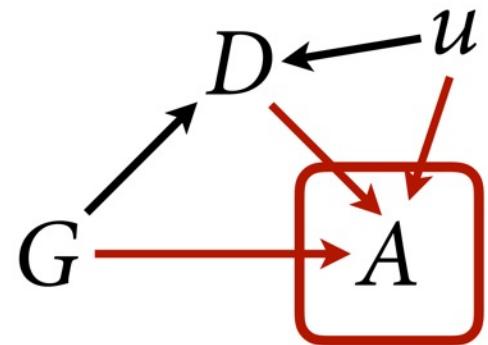
```



```

set.seed(17)
N <- 2000 # number of applicants
# even gender distribution
G <- sample( 1:2 , size=N , replace=TRUE )
# sample ability, high (1) to average (0)
u <- rbern(N,0.1)
# gender 1 tends to apply to department 1, 2 to 2
# and G=1 with greater ability tend to apply to 2 as well
D <- rbern( N . ifelse( G==1 . u*0.5 . 0.8 ) ) + 1
# matrix of acceptance rates [dept,gender]
accept_rate_u0 <- matrix( c(0.1,0.1,0.1,0.3) , nrow=2 )
accept_rate_u1 <- matrix( c(0.2,0.3,0.2,0.5) , nrow=2 )
## Simulate acceptance
p <- sapply( 1:N , function(i)
  ifelse( u[i]==0 , accept_rate_u0[D[i],G[i]] ,
  accept_rate_u1[D[i],G[i]] ) )
A <- rbern( N , p )

```



```

> accept_rate_u0
      [,1] [,2]
[1,]  0.1  0.1
[2,]  0.1  0.3

```

```

> accept_rate_u1
      [,1] [,2]
[1,]  0.2  0.2
[2,]  0.3  0.5

```

Acceptance rates:

Two different accept rates matices meaning that  $u$  influences the acceptance rates differentially.

```

dat_sim <- list( A=A , D=D , G=G )

# total effect gender
m1 <- ulam(
  alist(
    A ~ bernoulli(p),
    logit(p) <- a[G],
    a[G] ~ normal(0,1)
  ), data=dat_sim , chains=4 , cores=4 )

# direct effects - now confounded!
m2 <- ulam(
  alist(
    A ~ bernoulli(p),
    logit(p) <- a[G,D],
    matrix[G,D]:a ~ normal(0,1)
  ), data=dat_sim , chains=4 , cores=4 )

```

Text

*total effect shows disadvantage for gender*

```

> precis(m1,depth=2)
      mean   sd  5.5% 94.5% n_eff Rhat4
a[1] -2.10 0.10 -2.26 -1.93  1297    1
a[2] -0.86 0.07 -0.97 -0.76  1008    1

```

*direct effect confounded*

```

> precis(m2,depth=3)
      mean   sd  5.5% 94.5% n_eff Rhat4
a[1,1] -2.18 0.11 -2.35 -2.01  2083    1
a[1,2] -0.99 0.30 -1.49 -0.51  2408    1
a[2,1] -1.97 0.21 -2.31 -1.65  2335    1
a[2,2] -0.65 0.07 -0.77 -0.53  2260    1

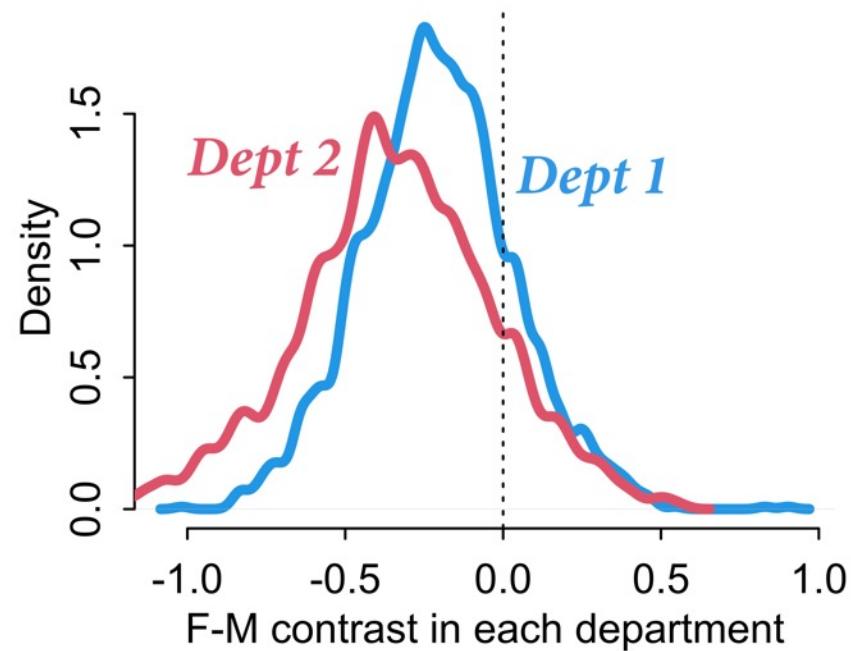
```

Let's look at the contrasts...

```
post2 <- extract.samples(m2)

post2$fm_contrast_D1 <-
  post2$a[,1,1] - post2$a[,2,1]

post2$fm_contrast_D2 <-
  post2$a[,1,2] - post2$a[,2,2]
```



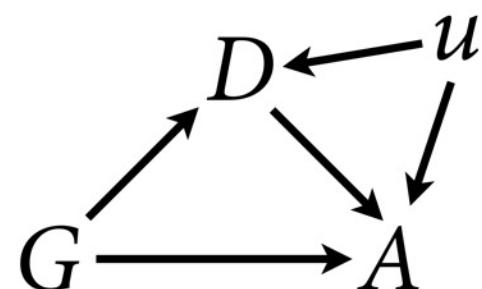
## How can the confound hide discrimination?

# You guessed it: Collider bias

Stratifying by  $D$  opens non-causal path through  $u$

Can estimate **total** causal effect of  $G$ ,  
but this isn't what we want

Cannot estimate **direct** effect of  $D$  or  $G$



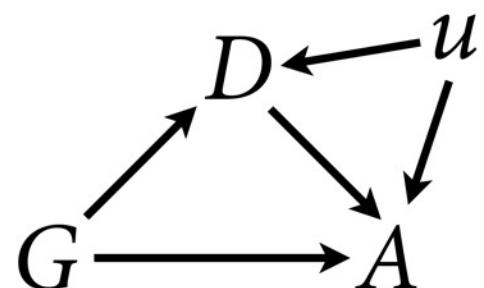
# You guessed it: Collider bias

More intuitive explanation:

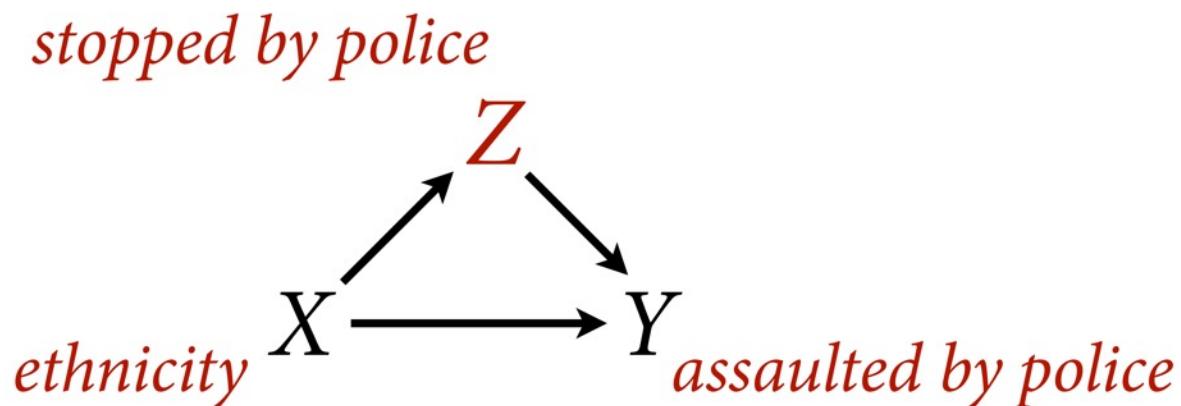
High ability G1s apply to discriminatory department anyway

G1s in that department are higher ability on average than G2s

High ability compensates for discrimination => masks evidence

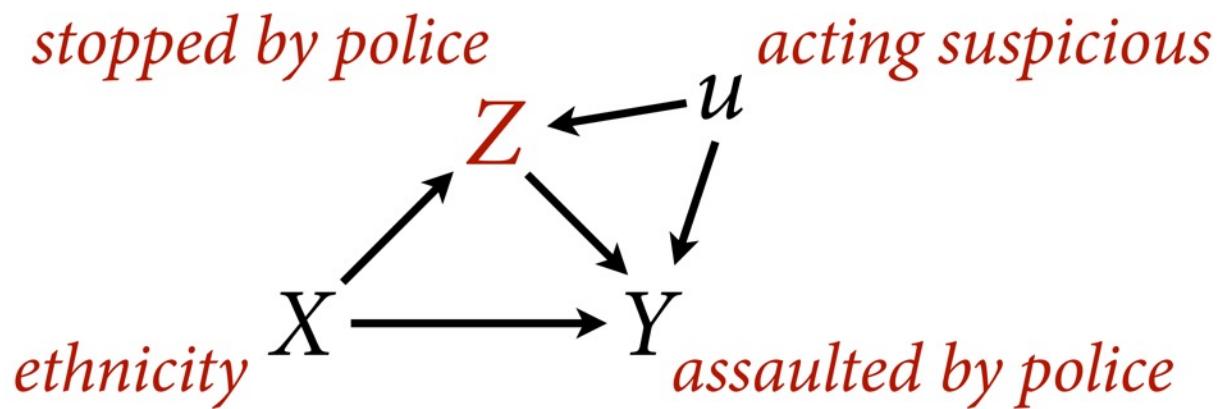


# Policing confounds



Knox Lowe & Mummolo 2020 Administrative Records Mask Racially Biased Policing

# Policing confounds



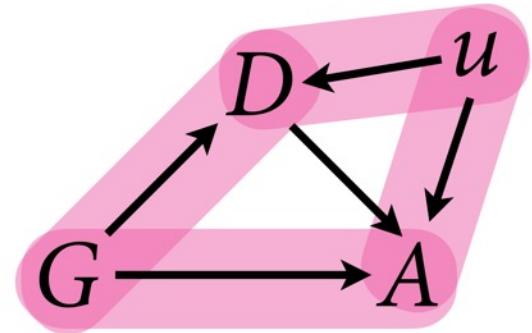
Data on police stops are confounded by lack of data on who wasn't stopped: forced conditioning on  $Z$

Knox Lowe & Mummolo 2020 Administrative Records Mask Racially Biased Policing

# Imagine we could measure the confound

```
# if we could measure u
dat_sim$u <- u
m3 <- ulam(
  alist(
    A ~ bernoulli(p),
    logit(p) <- a[G,D] + buA*u,
    matrix[G,D]:a ~ normal(0,1),
    buA ~ normal(0,1)
  ), data=dat_sim ,
constraints=list(buA="lower=0") ,
  chains=4 , cores=4 )

post3 <- extract.samples(m3)
post3$fm_contrast_D1 <-
  post3$a[,1,1] - post3$a[,2,1]
post3$fm_contrast_D2 <-
  post3$a[,1,2] - post3$a[,2,2]
```



Need to block non-causal  
path through  $u$

# Imagine we could measure the confound

```
# if we could measure u
dat_sim$u <- u
m3 <- ulam(
  alist(
    ^ ~ bernoulli(p),
    logit(p) ~ a[G,D] + buA*u,
    matrix[C,D]:u ~ normal(0,1),
    buA ~ normal(0,1)
  ), data=dat_sim ,
  constraints=list(buA="lower=0") ,
  chains=4 , cores=4 )

post3 <- extract.samples(m3)
post3$fm_contrast_D1 <-
  post3$a[,1,1] - post3$a[,2,1]
post3$fm_contrast_D2 <-
  post3$a[,1,2] - post3$a[,2,2]
```

Add  $u$  to linear model

# Imagine we could measure the confound

```
# if we could measure u
dat_sim$u <- u
m3 <- ulam(
  alist(
    A ~ bernoulli(p),
    logit(p) <- a[G,D] + buA*u,
    matrix[G,D]:a ~ normal(0,1),
    buA ~ normal(0,1)
  ), data=dat_sim ,
  constraints=list(buA="lower=0") ,
  chains=1 , cores=1 )

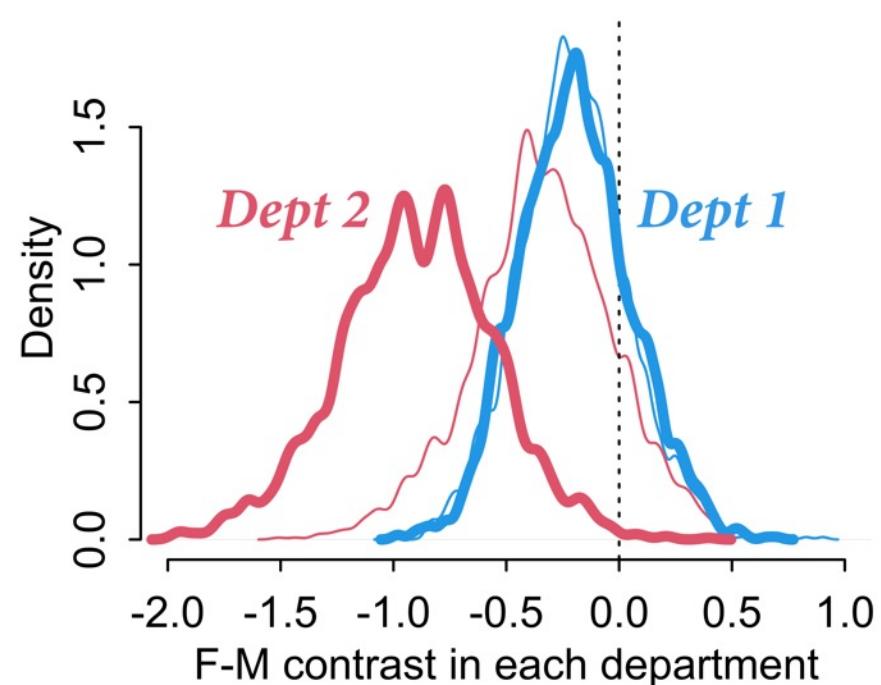
post3 <- extract.samples(m3)
post3$fm_contrast_D1 <-
  post3$a[,1,1] - post3$a[,2,1]
post3$fm_contrast_D2 <-
  post3$a[,1,2] - post3$a[,2,2]
```

Constrain effect of  $u$  to +

# Imagine we could measure the confound

```
# if we could measure u
dat_sim$u <- u
m3 <- ulam(
  alist(
    A ~ bernoulli(p),
    logit(p) <- a[G,D] + buA*u,
    matrix[G,D]:a ~ normal(0,1),
    buA ~ normal(0,1)
  ), data=dat_sim ,
constraints=list(buA="lower=0") ,
  chains=4 , cores=4 )

post3 <- extract.samples(m3)
post3$fm_contrast_D1 <-
  post3$a[,1,1] - post3$a[,2,1]
post3$fm_contrast_D2 <-
  post3$a[,1,2] - post3$a[,2,2]
```



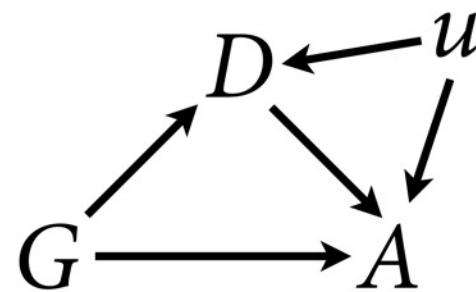
# De-confounding

What can be done?

Experiments

Sensitivity analysis

Measure proxies of confound



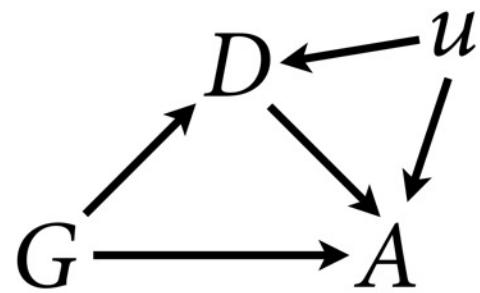
# Sensitivity analysis

Text

**What are the implications of what we don't know?**

Assume confound exists, model its consequences for different strengths/kinds of influence

How strong must the confound be to change conclusions?

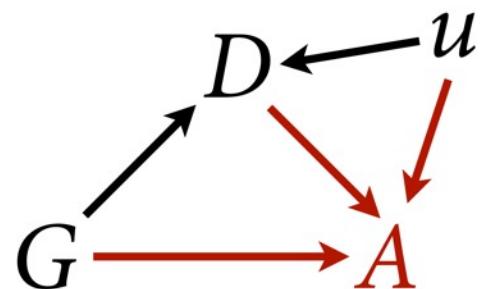


# Sensitivity analysis

$$A_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha[G_i, D_i] + \beta_{G[i]} u_i$$

Text



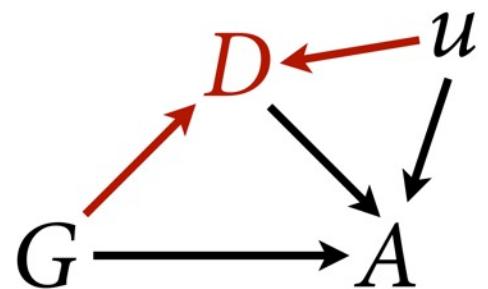
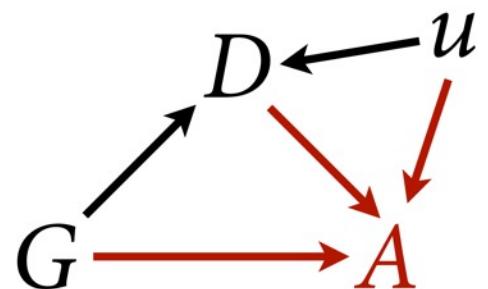
# Sensitivity analysis

$$A_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha[G_i, D_i] + \beta_{G[i]} u_i$$

$$(D_i = 2) \sim \text{Bernoulli}(q_i)$$

$$\text{logit}(q_i) = \delta[G_i] + \gamma_{G[i]} u_i$$



```

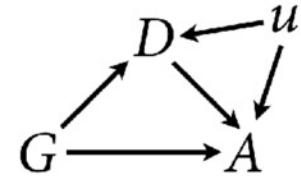
datl$D1 <- ifelse(datl$D==1,1,0)
datl$N <- length(datl$D)
datl$b <- c(1,1)
datl$g <- c(1,0)

mGDU <- ulam(
  alist(
    # A model
    A ~ bernoulli(p),
    logit(p) <- a[G,D] + b[G]*u[i],
    matrix[G,D]:a ~ normal(0,1),

    # D model
    D1 ~ bernoulli(q),
    logit(q) <- delta[G] + g[G]*u[i],
    delta[G] ~ normal(0,1),

    # declare unobserved u
    vector[N]:u ~ normal(0,1)
  ), data=datl , chains=4 , cores=4 )

```



$$A_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha[G_i, D_i] + \beta_{G[i]} u_i$$

$$(D_i = 2) \sim \text{Bernoulli}(q_i)$$

$$\text{logit}(q_i) = \delta[G_i] + \gamma_{G[i]} u_i$$

$$u_j \sim \text{Normal}(0,1)$$

```

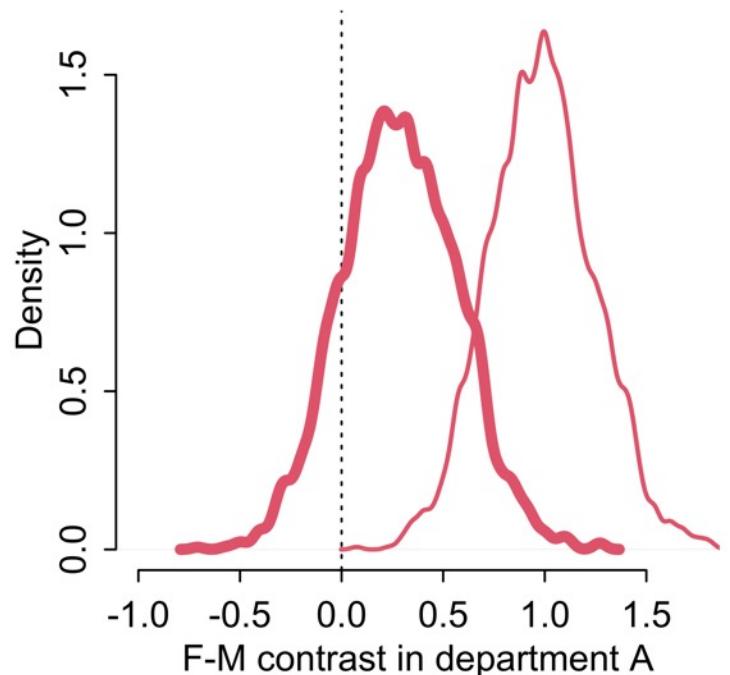
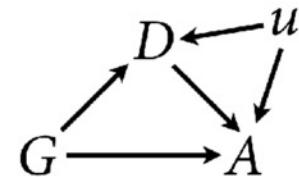
datl$D1 <- ifelse(datl$D==1,1,0)
datl$N <- length(datl$D)
datl$b <- c(1,1)
datl$g <- c(1,0)

mGDU <- ulam(
  alist(
    # A model
    A ~ bernoulli(p),
    logit(p) <- a[G,D] + b[G]*u[i],
    matrix[G,D]:a ~ normal(0,1),

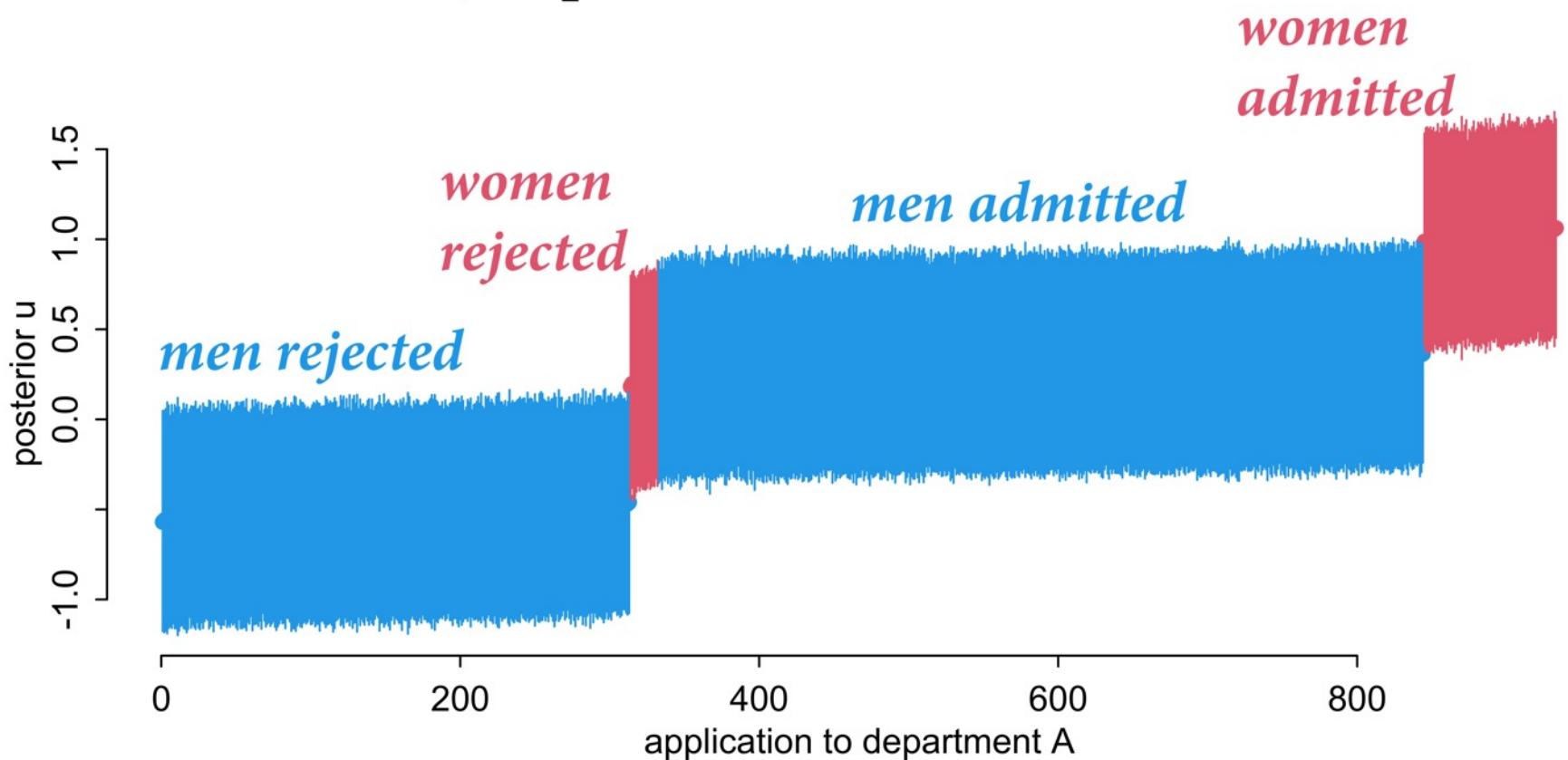
    # D model
    D1 ~ bernoulli(q),
    logit(q) <- delta[G] + g[G]*u[i],
    delta[G] ~ normal(0,1),

    # declare unobserved u
    vector[N]:u ~ normal(0,1)
  ), data=datl , chains=4 , cores=4 )

```



## Posterior $u$ values, Department A



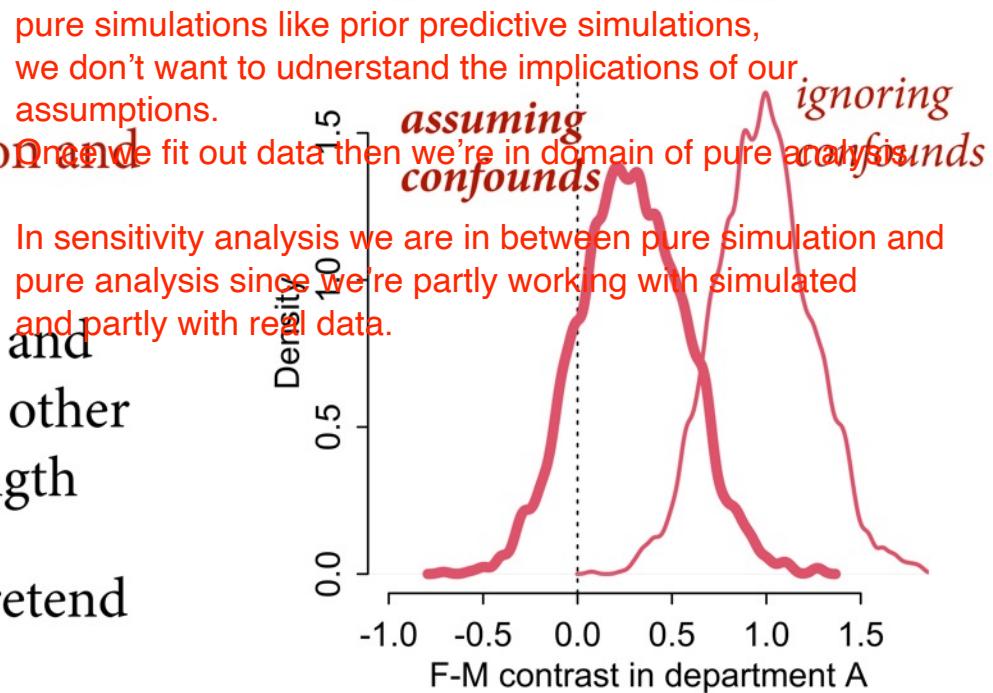
# Sensitivity analysis

What are the implications of what we don't know?

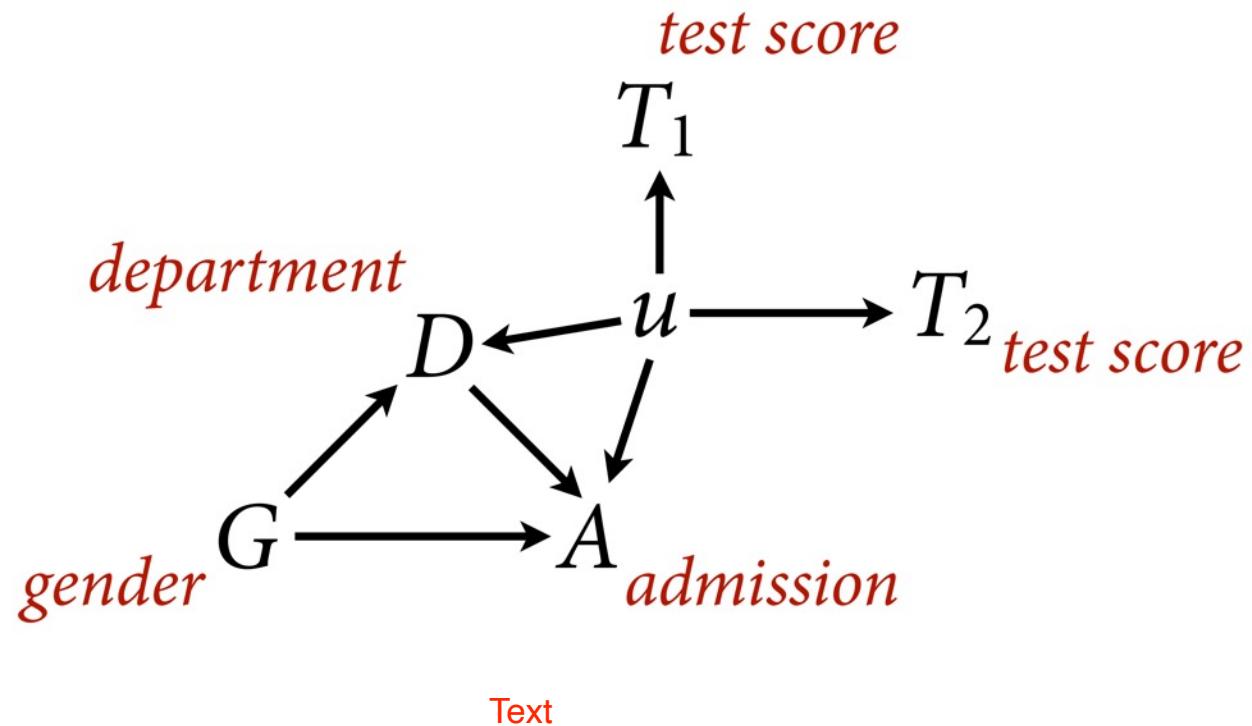
Somewhere between pure simulation and pure analysis

Vary confound strength over range and show how results change –or– vary other effects and estimate confound strength

Confound often persist — don't pretend



# De-confounding



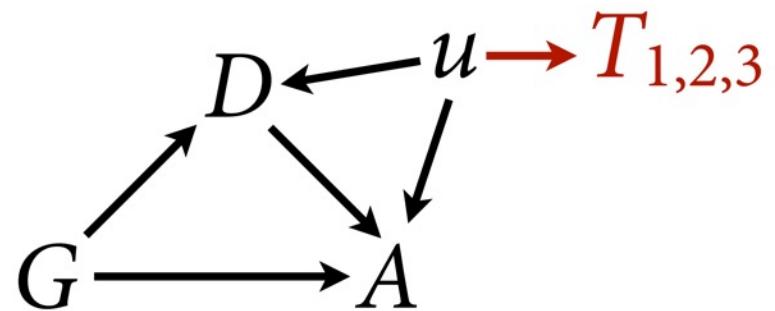
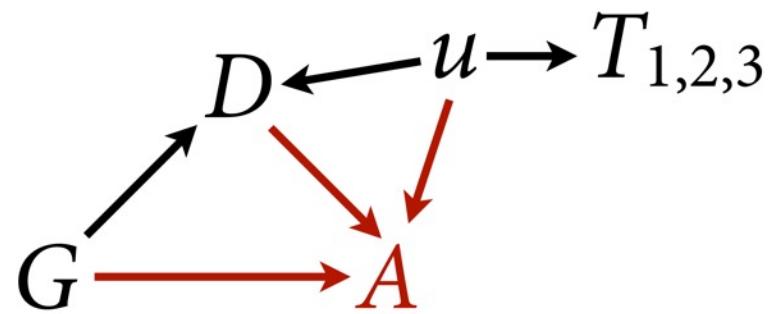
# De-confounding

$$A_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha[G_i, D_i] + \beta_{G[i]} u_i$$

$$T_{i,j} \sim \text{Normal}(u_i, \tau_j)$$

$$u_k \sim \text{Normal}(0, 1)$$

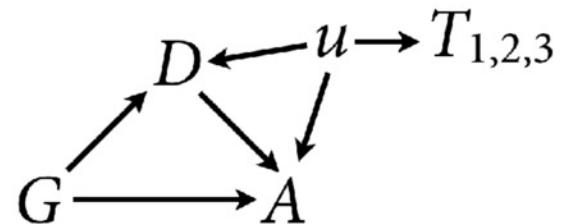


```

T1 <- rnorm(N,u,0.1)
T2 <- rnorm(N,u,0.5)
T3 <- rnorm(N,u,0.25)

m4 <- ulam(
  alist(
    # A model
    A ~ bernoulli(p),
    logit(p) <- a[G,D] + b*u[i],
    matrix[G,D]:a ~ normal(0,1),
    b ~ normal(0,1),
    # u and T model
    vector[N]:u ~ normal(0,1),
    T1 ~ normal(u,tau[1]),
    T2 ~ normal(u,tau[2]),
    T3 ~ normal(u,tau[3]),
    vector[3]:tau ~ exponential(1)
  ),
  data=dat_sim , chains=4 , cores=4 ,
  constraints=list(b="lower=0") )

```



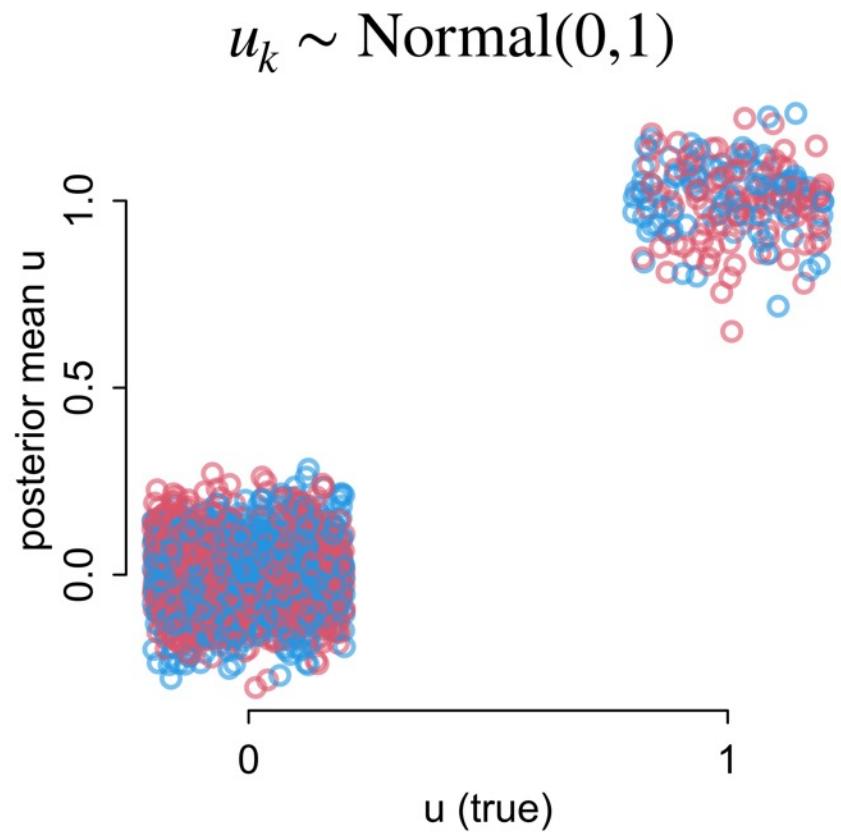
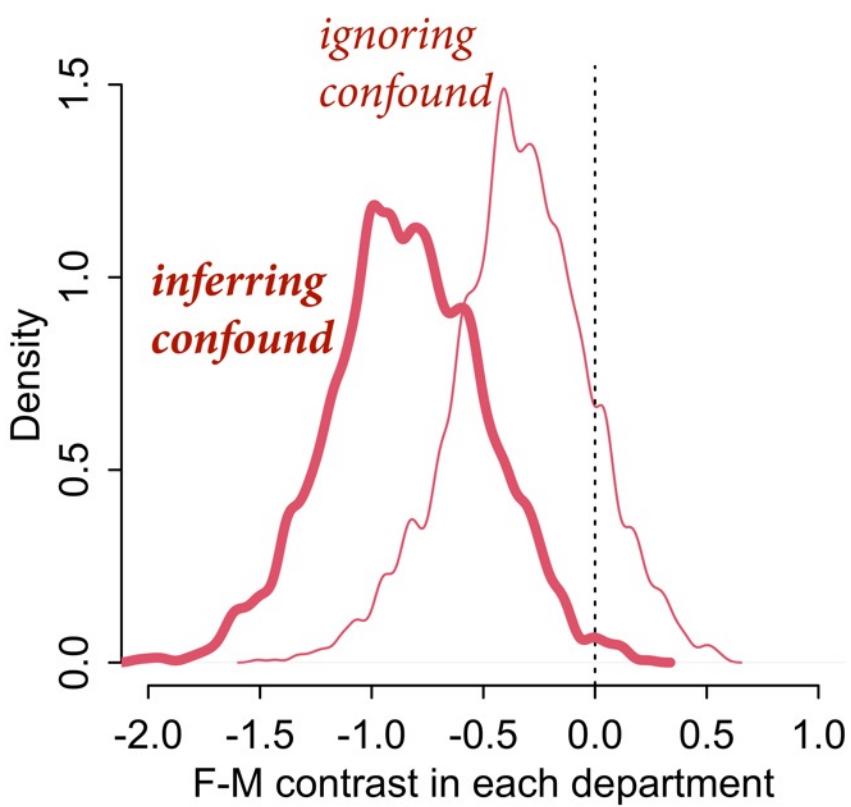
$$A_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha[G_i, D_i] + \beta_{G[i]} u_i$$

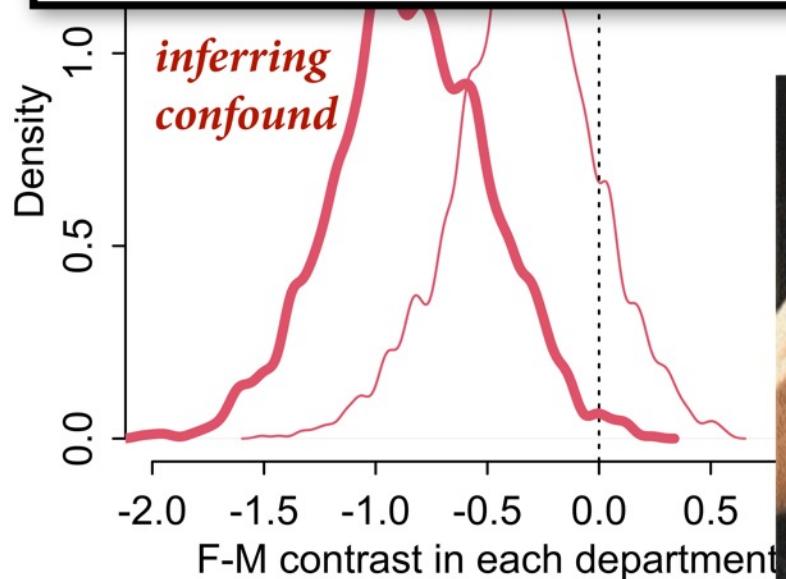
$$u_k \sim \text{Normal}(0,1)$$

$$T_{i,j} \sim \text{Normal}(u_i, \tau_j)$$

\*This model samples inefficiently; learn to fix later in course

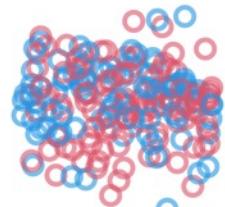


# *More parameters (2008) than observations (2000)!*



an u

Normal(0,1)



How can we have more parameters than observations?

# Oceanic tool complexity

How is technological complexity related to population size?

To social structure?

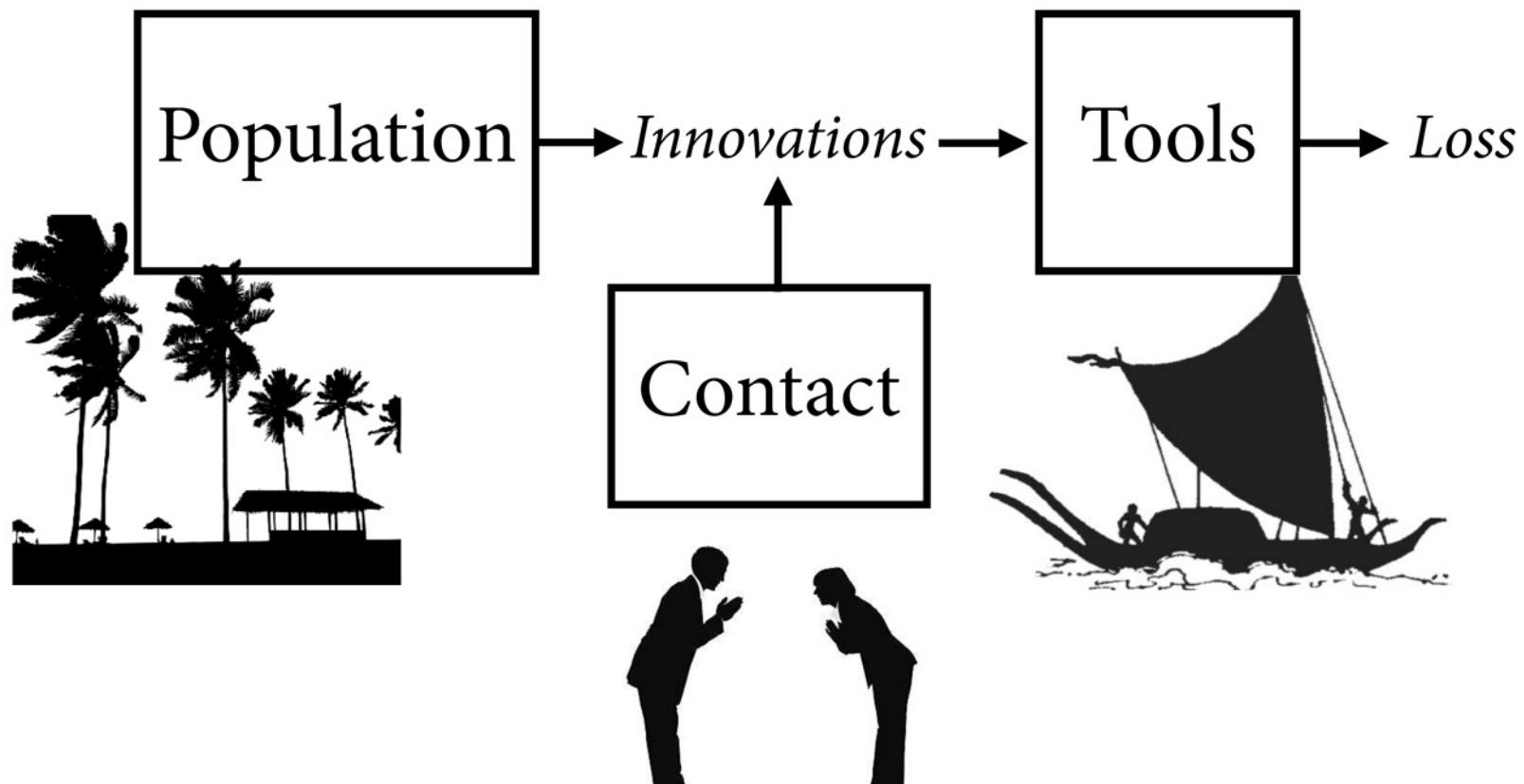
data(Kline)

	culture	population	contact	total_tools	mean_TU
1	Malekula	1100	low	13	3.2
2	Tikopia	1500	low	22	4.7
3	Santa Cruz	3600	low	24	4.0
4	Yap	4791	high	43	5.0
5	Lau Fiji	7400	high	33	5.0
6	Trobriand	8000	high	19	4.0
7	Chuuk	9200	high	40	3.8
8	Manus	13000	low	28	6.6
9	Tonga	17500	high	55	5.4
10	Hawaii	275000	low	71	6.6

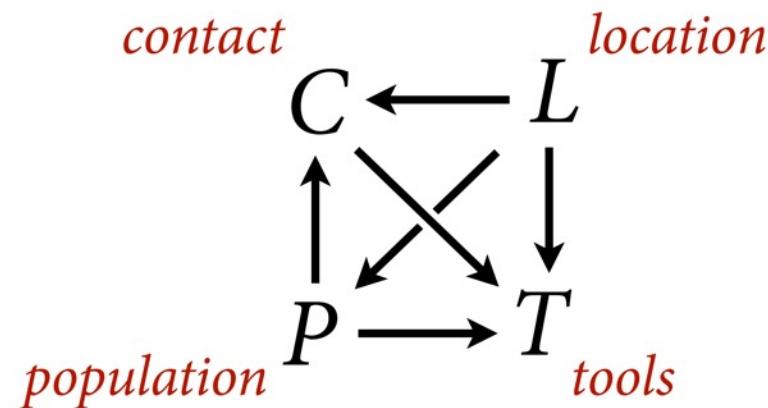
Estimand: Causal influence of **population size** and **contact** on **total tools**

Kline & Boyd 2010 Population size predicts technological complexity in Oceania

# Technological complexity

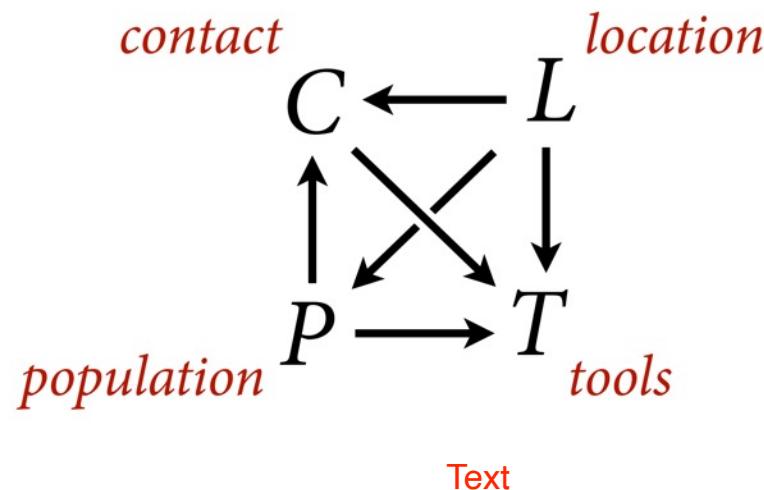


# Technological complexity



we are interested in the effect of population  
on tools, and contact and location  
as confounders.

# Technological complexity



Adjustment set for  $P$ :  $L$

Also want to stratify by  $C$ ,  
to study interaction

# Modeling tools

Tool count is not binomial: No maximum

**Poisson distribution:** Very high maximum and very low probability of each success

Here: Many many possible technologies, very few realized in any one place

# Poisson link is log

Poisson distribution takes shape from expected value

Must be positive

Exponential scaling can be surprising!

number of traffic accidents (outcome)

$$Y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \alpha + \beta x_i$$

$$\lambda_i = \exp(\alpha + \beta x_i)$$

# Poisson (poison) priors

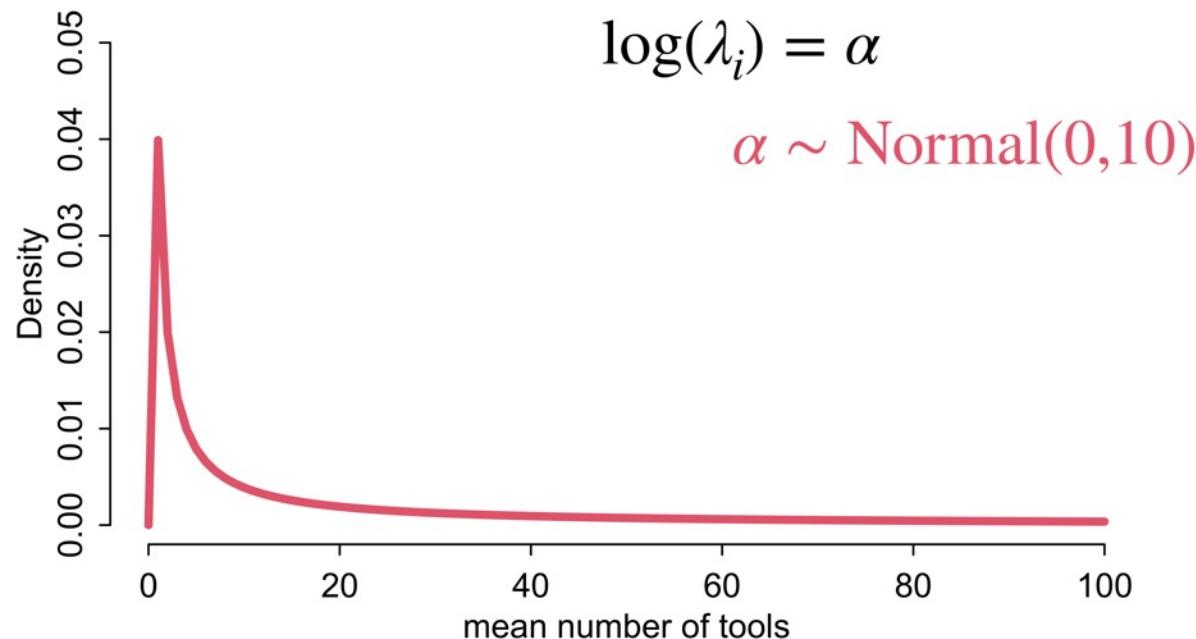
Exponential scaling can be surprising

$$\log(\lambda_i) = \alpha$$

$$\alpha \sim \text{Normal}(0, 10)$$

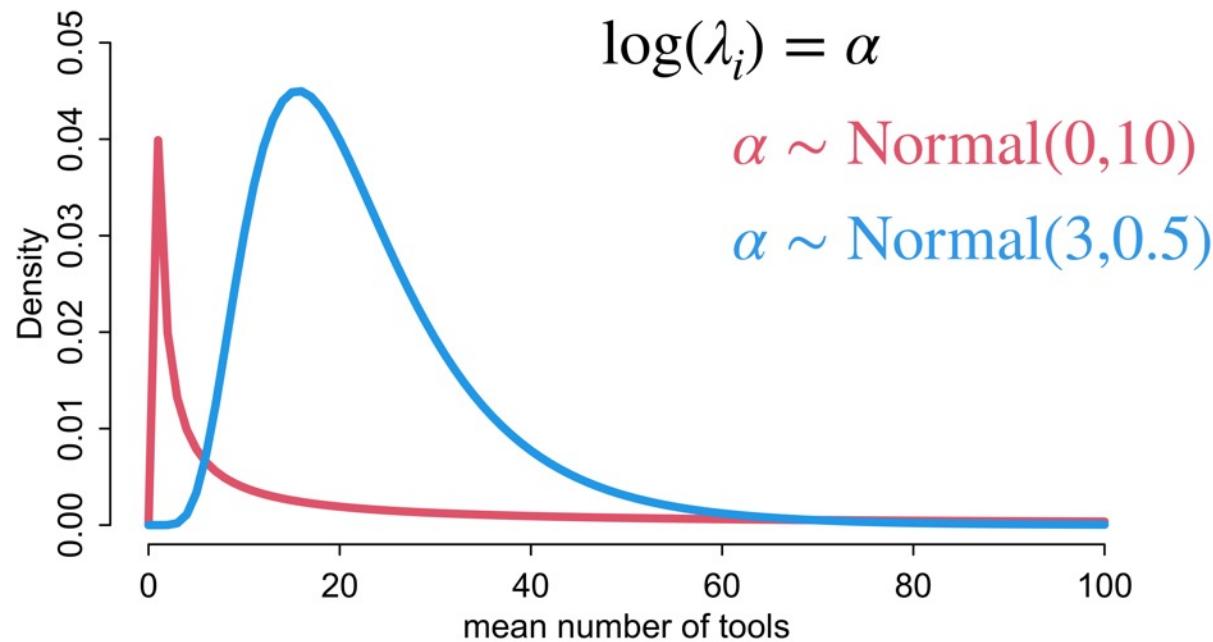
# Poisson (poison) priors

Exponential scaling can be surprising



# Poisson (poison) priors

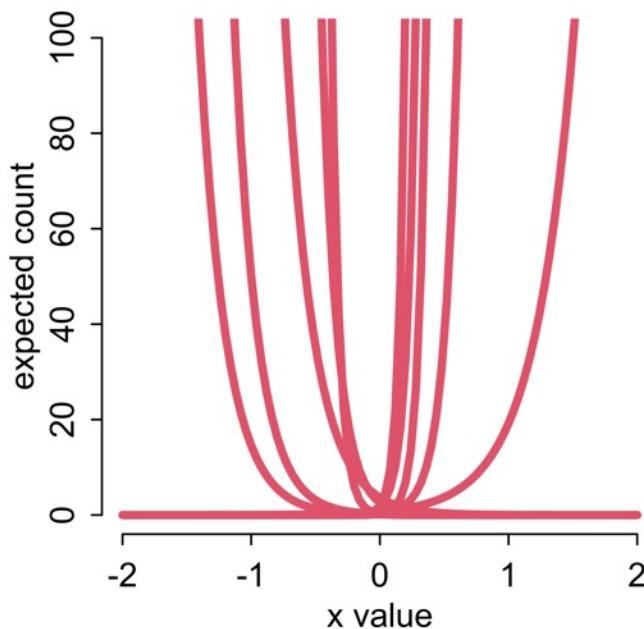
Exponential scaling can be surprising



# Poisson priors

$\alpha \sim \text{Normal}(0,1)$

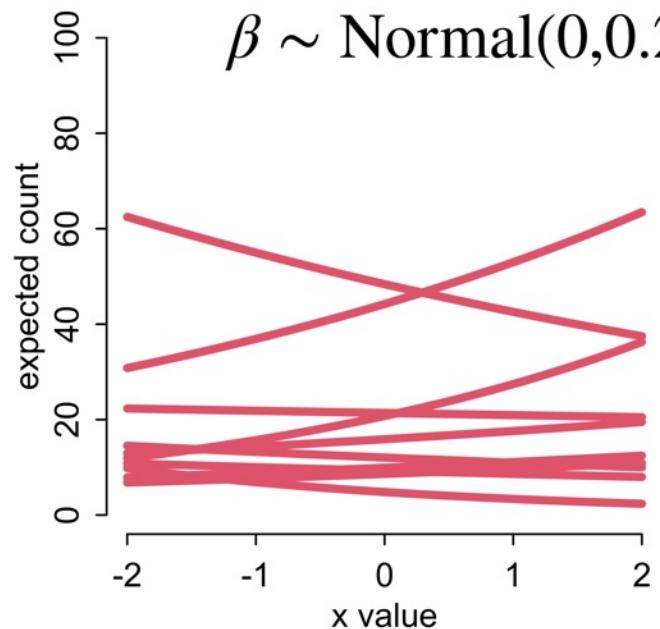
$\beta \sim \text{Normal}(0,10)$



$$\log(\lambda_i) = \alpha + \beta x_i$$

$\alpha \sim \text{Normal}(3,0.5)$

$\beta \sim \text{Normal}(0,0.2)$



```

data(Kline)
d <- Kline
d$P <- scale( log(d$population) )
d$contact_id <- ifelse( d$contact=="high" , 2 , 1 )
dat <- list(
  T = d$total_tools ,
  P = d$P ,
  C = d$contact_id )

# intercept only
m11.9 <- ulam(
  alist(
    T ~ dpois( lambda ),
    log(lambda) <- a,
    a ~ dnorm( 3 , 0.5 )
  ), data=dat , chains=4 , log_lik=TRUE )
# interaction model
m11.10 <- ulam(
  alist(
    T ~ dpois( lambda ),
    log(lambda) <- a[C] + b[C]*P,
    a[C] ~ dnorm( 3 , 0.5 ),
    b[C] ~ dnorm( 0 , 0.2 )
  ), data=dat , chains=4 , log_lik=TRUE )

compare( m11.9 , m11.10 , func=PSIS )

```

$$Y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \alpha_{C[i]} + \beta_{C[i]} \log(P_i)$$

$$\alpha_j \sim \text{Normal}(3,0.5)$$

$$\beta_j \sim \text{Normal}(0,0.2)$$

```

data(Kline)
d <- Kline
d$P <- scale( log(d$population) )
d$contact_id <- ifelse( d$contact=="high" , 2 , 1 )
dat <- list(
  T = d$total_tools ,
  P = d$P ,
  C = d$contact_id )

# intercept only
m11.9 <- ulam(
  alist(
    T ~ dpois( lambda ),
    log(lambda) <- a,
    a ~ dnorm( 3 , 0.5 )
  ), data=dat , chains=4 , log_lik=TRUE )
# interaction model
m11.10 > compare( m11.9 , m11.10 , func=PSIS )
Some Pareto k values are high (>0.5). Set pointwise=TRUE to inspect individual points.
Some Pareto k values are high (>0.5). Set pointwise=TRUE to inspect individual points.
      PSIS      SE dPSIS   dSE pPSIS weight
m11.10  85.9 13.50    0.0    NA    7.3      1
m11.9   141.3 33.69   55.4 33.13    8.0      0
), update=update , chains=4 , log_lik=TRUE )

compare( m11.9 , m11.10 , func=PSIS )

```

$$Y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \alpha_{C[i]} + \beta_{C[i]} \log(P_i)$$

$$\alpha_j \sim \text{Normal}(3,0.5)$$

$$\beta_j \sim \text{Normal}(0,0.2)$$

```

k <- PSIS( m11.10 , pointwise=TRUE )$k
plot( dat$P , dat$T , xlab="log population (std)" ,
ylab="total tools" ,
col=ifelse( dat$C==1 , 4 , 2 ) , lwd=4+4*normalize(k) ,
ylim=c(0,75) , cex=1+normalize(k) )

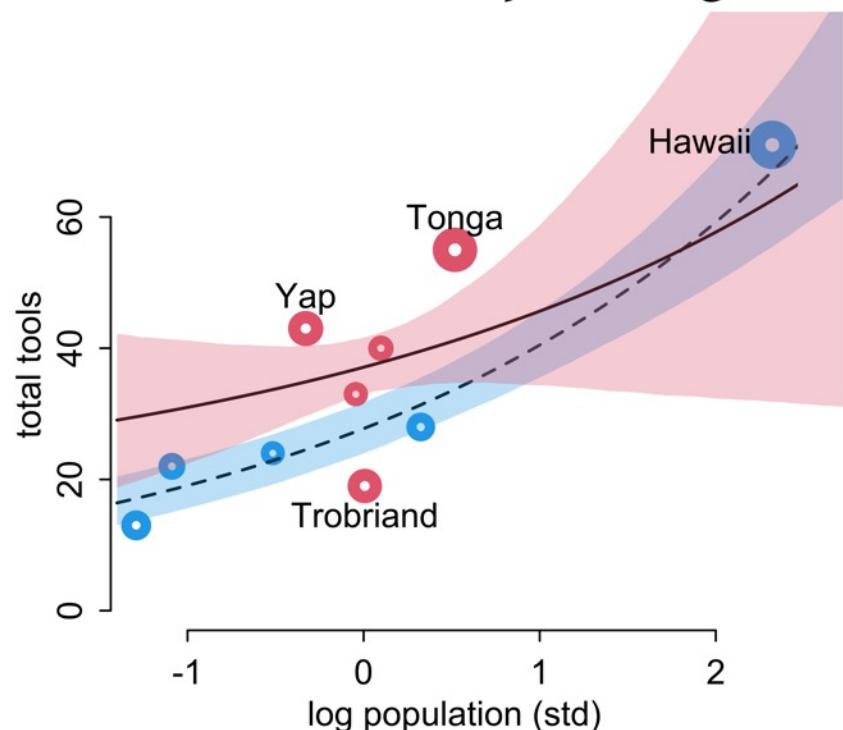
# set up the horizontal axis values to compute predictions
at
P_seq <- seq( from=-1.4 , to=3 , len=100 )

# predictions for C=1 (low contact)
lambda <- link( m11.10 , data=data.frame( P=P_seq , C=1 ) )
lmu <- apply( lambda , 2 , mean )
lci <- apply( lambda , 2 , PI )
lines( P_seq , lmu , lty=2 , lwd=1.5 )
shade( lci , P_seq , xpd=TRUE , col=col.alpha(4,0.3) )

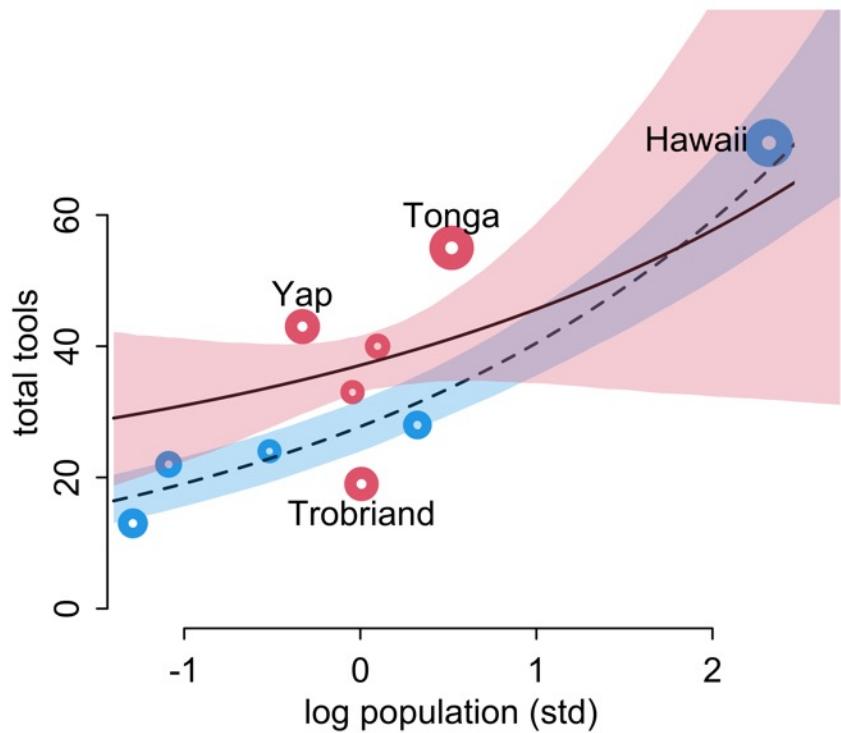
# predictions for C=2 (high contact)
lambda <- link( m11.10 , data=data.frame( P=P_seq , C=2 ) )
lmu <- apply( lambda , 2 , mean )
lci <- apply( lambda , 2 , PI )
lines( P_seq , lmu , lty=1 , lwd=1.5 )
shade( lci , P_seq , xpd=TRUE , col=col.alpha(2,0.3) )

```

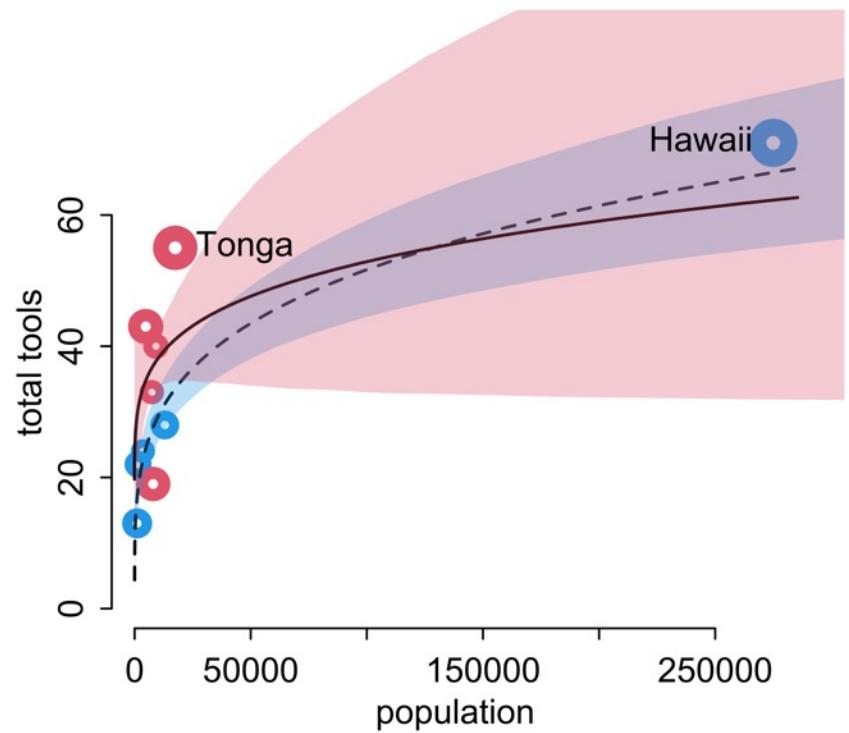
*Points scaled by leverage*

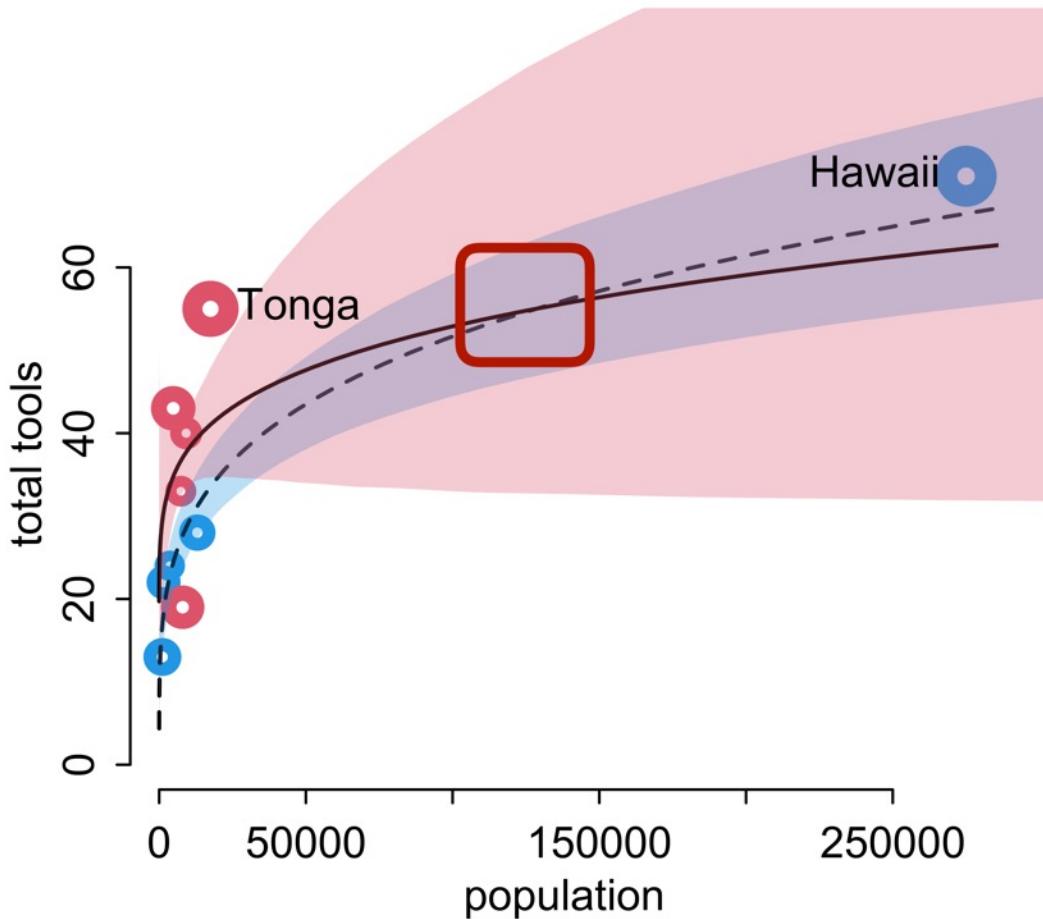


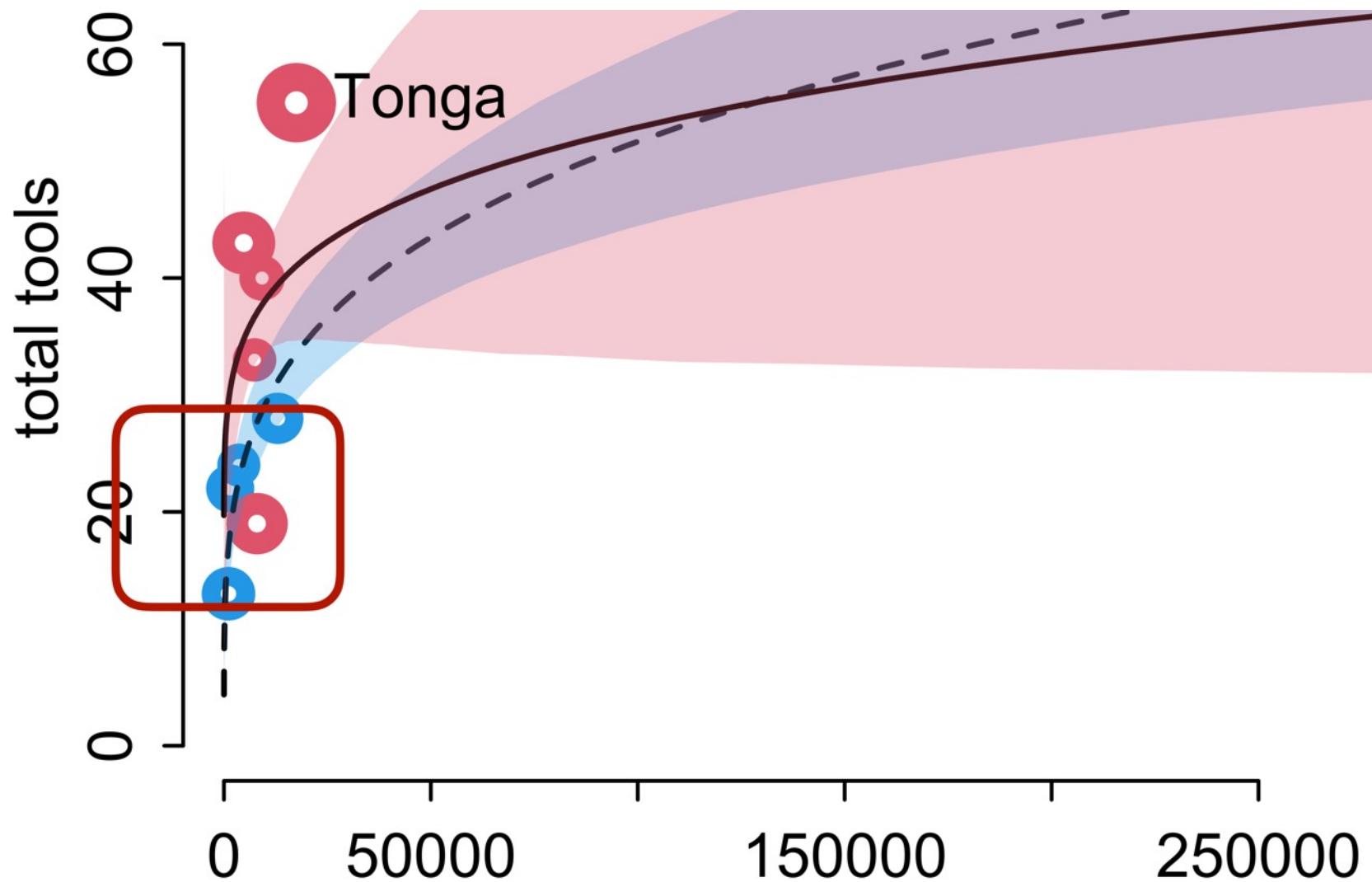
*log scale*



*Natural scale*



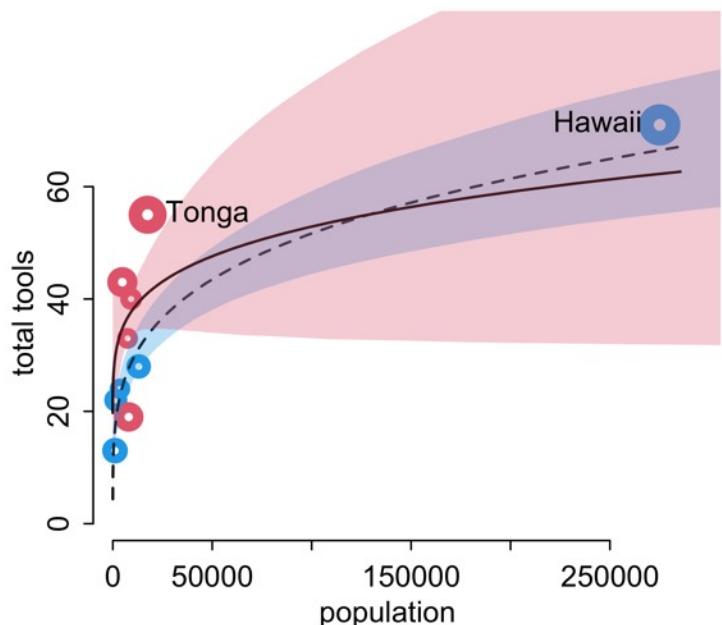




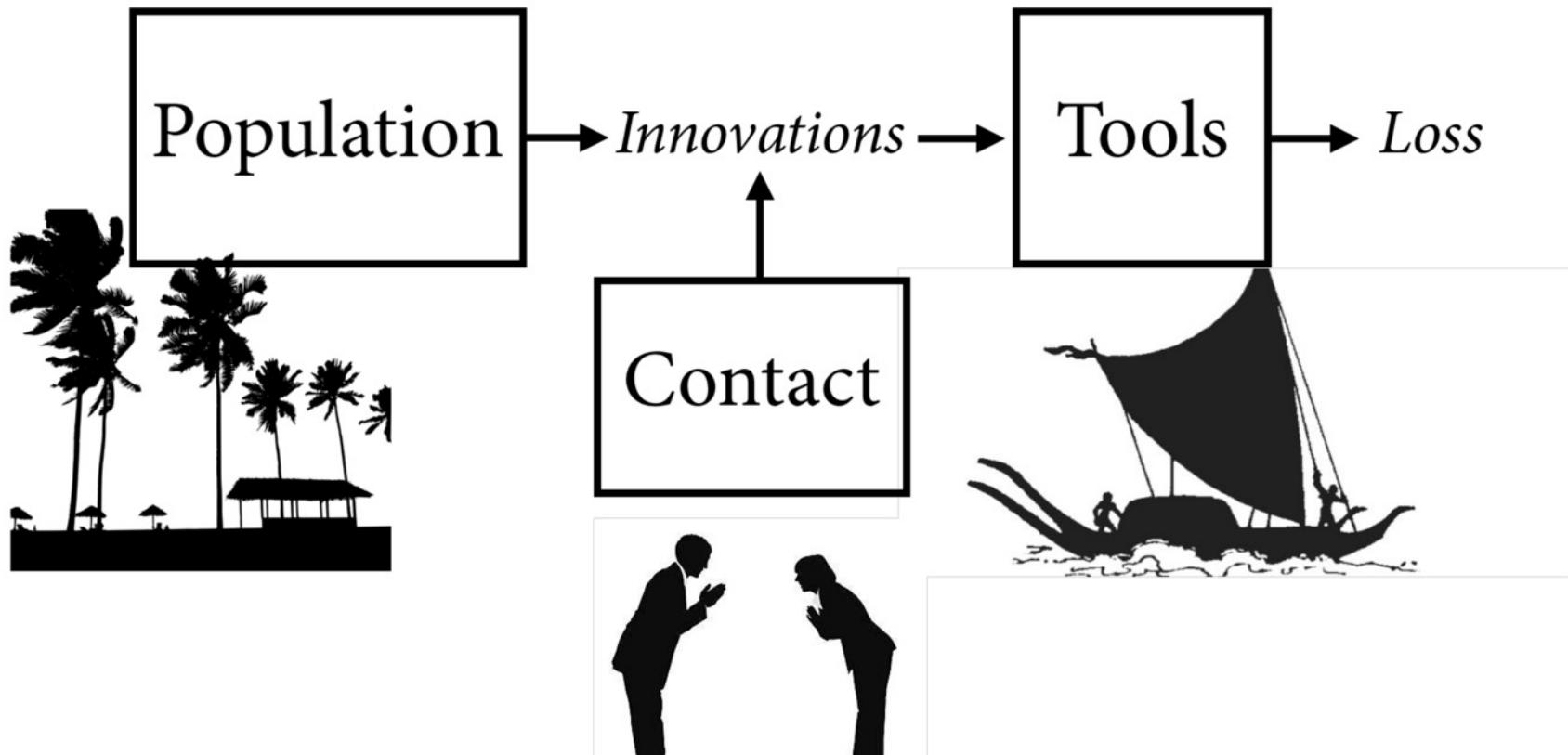
# Oceanic tools

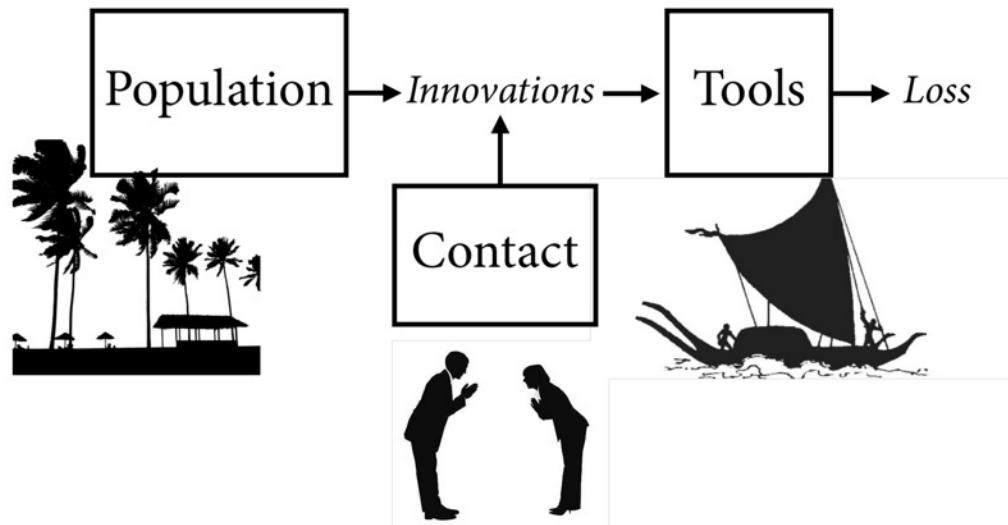
Two immediate ways to improve the model

- (1) Use a robust model:  
gamma-Poisson (neg-binomial)
- (2) Use a more principled scientific  
model



# Technological complexity





$$\Delta T = \alpha P^\beta - \gamma T$$

$$\Delta T = \alpha P^\beta - \gamma T$$

Diagram illustrating the components of the equation:

- change in tools* points to the term  $\alpha P^\beta$ .
- innovation rate* points to the term  $\alpha P^\beta$ .
- diminishing returns (elasticity)* points to the term  $\gamma T$ .
- rate of loss* points to the term  $\gamma T$ .

*diminishing returns  
depend upon contact*

$$\Delta T = \alpha_C P^{\beta_C} - \gamma T$$

*innovation depends  
upon contact*

$$\Delta T = \alpha_C P^{\beta_C} - \gamma T = 0$$

*Solve for T*

$$\Delta T = \alpha_C P^{\beta_C} - \gamma T = 0$$

*Solve for T*

$$\hat{T} = \frac{\alpha_C P^{\beta_C}}{\gamma}$$

$$\hat{T} = \frac{\alpha_C P^{\beta_C}}{\gamma}$$

$$T_i \sim \text{Poisson}(\lambda_i)$$
$$\lambda_i = \hat{T}$$

```
# innovation/loss model

dat2 <- list( T=d$total_tools, P=d$population,
C=d$contact_id )

m11.11 <- ulam(
  alist(
    T ~ dpois( lambda ),
    lambda <- exp(a[C])*P^b[C]/g,
    a[C] ~ dnorm(1,1),
    b[C] ~ dexp(1),
    g ~ dexp(1)
  ), data=dat2 , chains=4 , cores=4 )
```

All parameters must be positive

Two ways to do this

(1) use `exp()`

(2) use appropriate prior

```

# innovation/loss model

dat2 <- list( T=d$total_tools, P=d$population,
C=d$contact_id )

m11.11 <- ulam(
  alist(
    T ~ dpois( lambda ),
    lambda <- exp(a[C])*P^b[C]/g,
    a[C] ~ dnorm(1,1),
    b[C] ~ dexp(1),
    g ~ dexp(1)
  ), data=dat2 , chains=4
)

```

All parameters must be positive

Two ways to do this

(1) use `exp()`

```

> precis(m11.11,2)
      mean   sd  5.5% 94.5% n_eff Rhat4
a[1] 0.85 0.68 -0.26  1.90   698     1
a[2] 0.93 0.83 -0.39  2.31   902     1
b[1] 0.26 0.03  0.21  0.32  1149     1
b[2] 0.29 0.10  0.12  0.45   711     1
g    1.11 0.70  0.32  2.43   862     1

```

appropriate prior

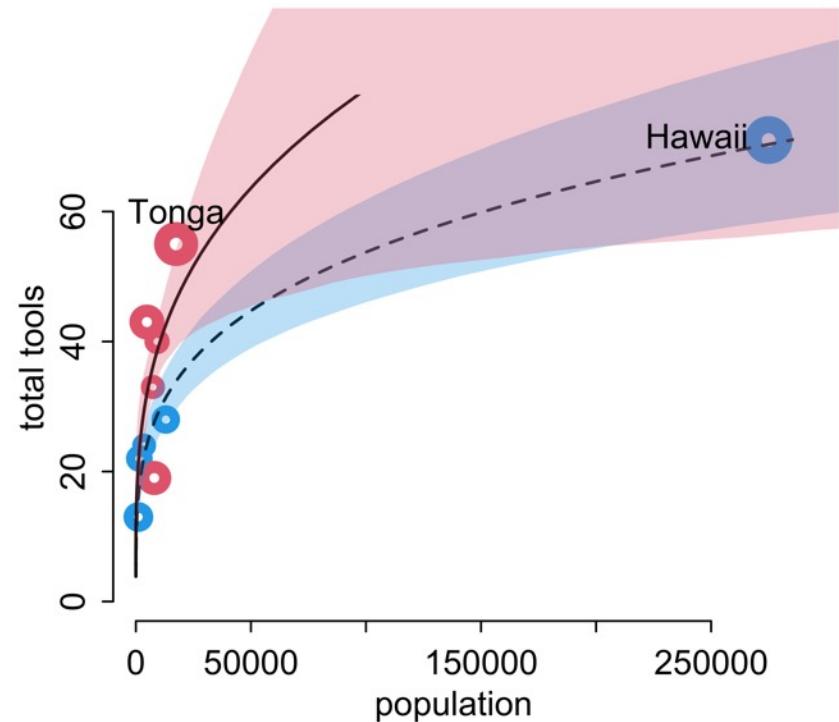
```

# innovation/loss model

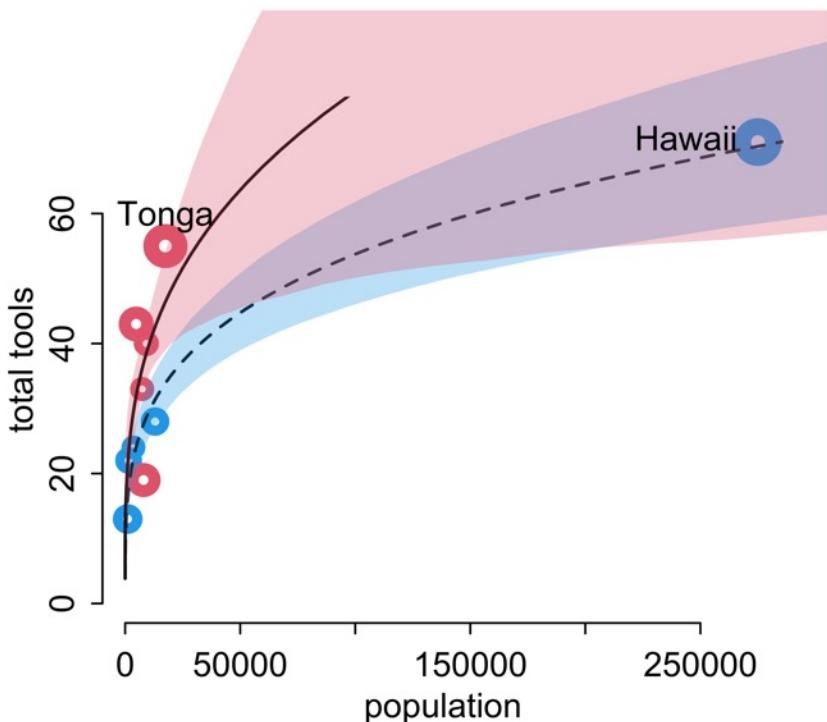
dat2 <- list( T=d$total_tools, P=d$population,
C=d$contact_id )

m11.11 <- ulam(
  alist(
    T ~ dpois( lambda ),
    lambda <- exp(a[C])*P^b[C]/g,
    a[C] ~ dnorm(1,1),
    b[C] ~ dexp(1),
    g ~ dexp(1)
  ), data=dat2 , chains=4 , cores=4 )

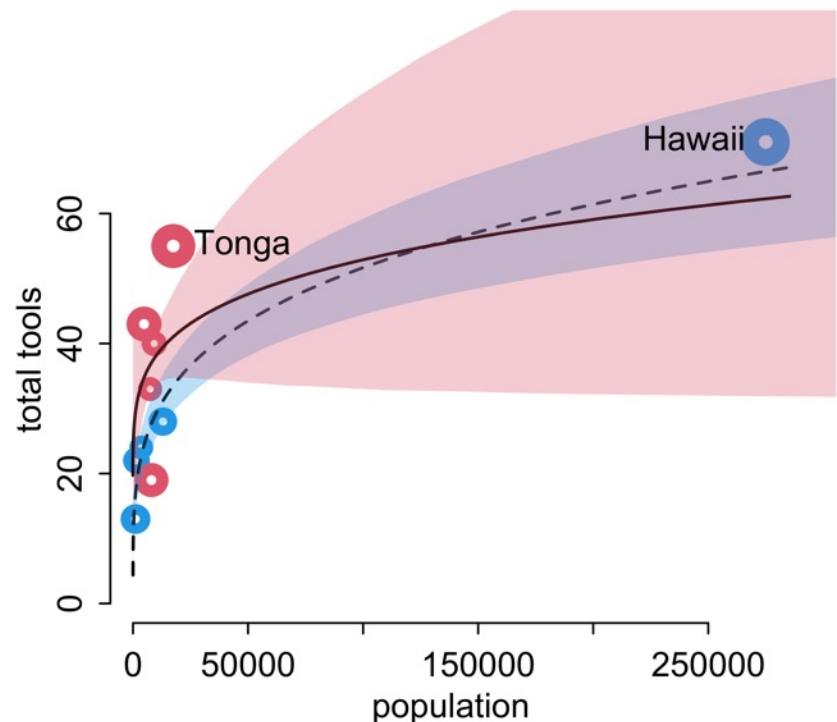
```



Innovation/loss model



Generalized linear model



Still have to deal with location as confound

# Count GLMs

Before you see the values, you know a count is zero or positive

Maximum entropy priors: Binomial, Poisson, and extensions

More event types: Multinomial and categorical

Robust regressions:  
Beta-binomial, gamma-Poisson (neg-binomial)

Examples to come



# Trolley Problems

data(Trolley)

331 individuals (age, gender, edu)

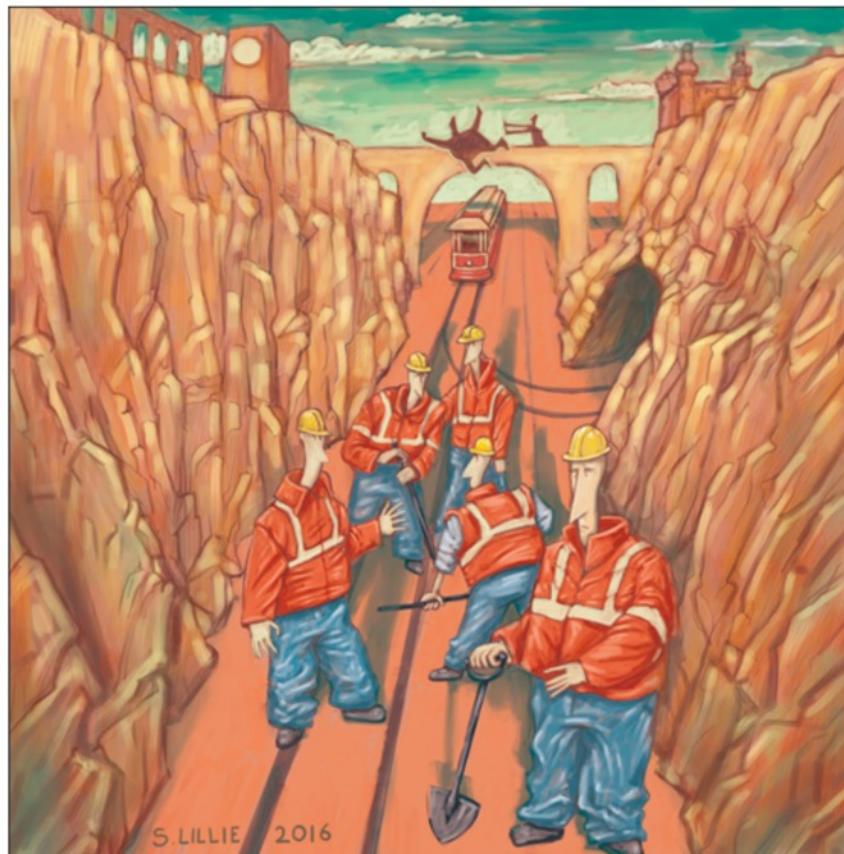
Voluntary participation (online)

30 different trolley problems

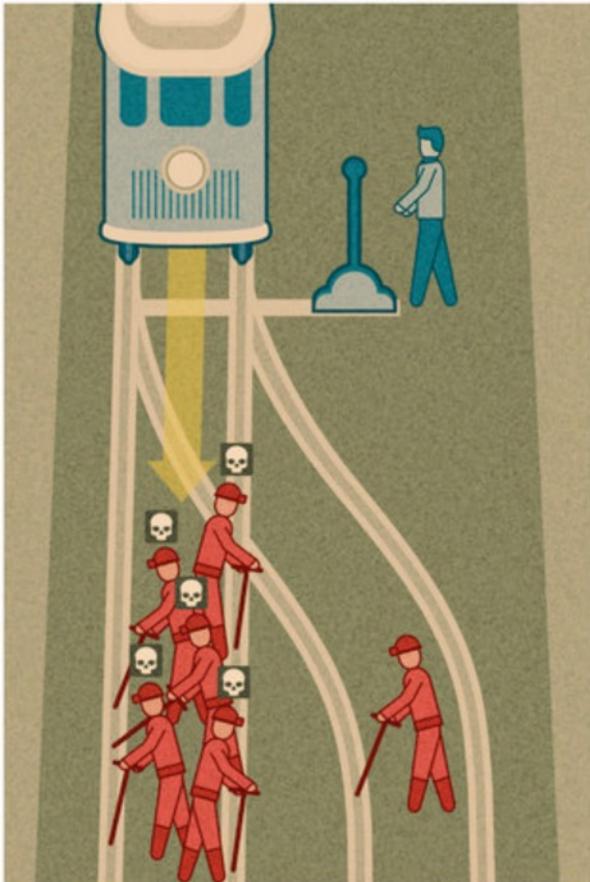
action / intention / contact

9930 responses:

*How appropriate (from 1 to 7)?*



# *Action*



# Trolley Problems

data(Trolley)

331 individuals (age, gender, edu)

Voluntary participation (online)

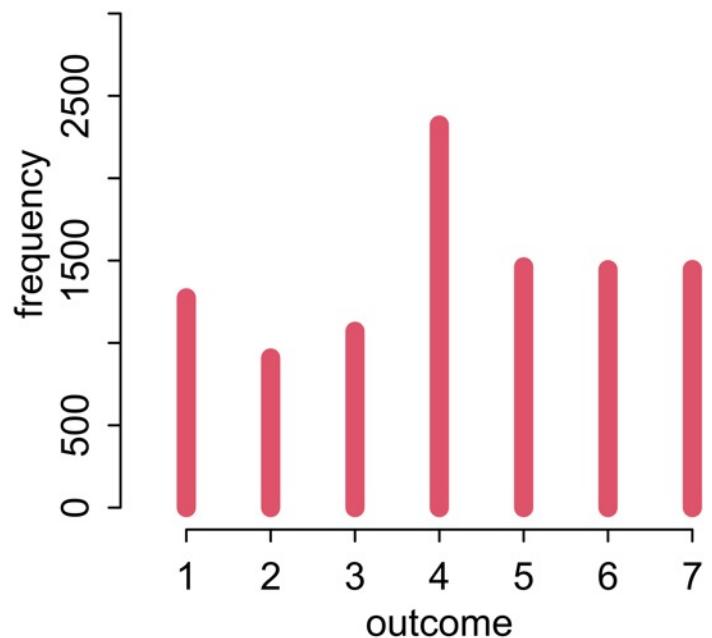
30 different trolley problems

action / intention / contact

9930 responses:

*How appropriate (from 1 to 7)?*

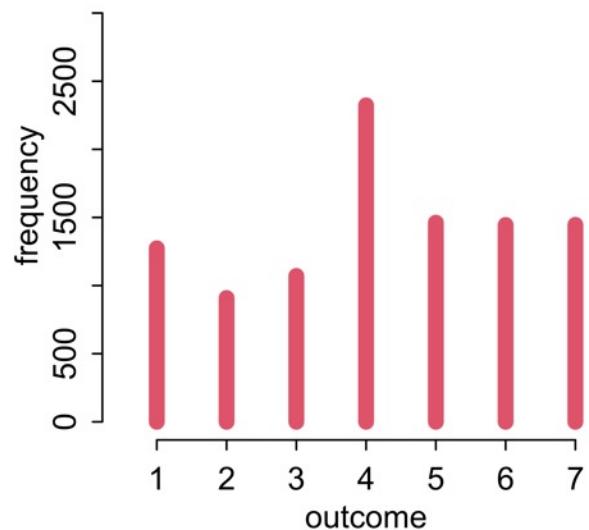
*Ordered categorical*



Estimand: How do **action**, **intention**, **contact** influence **response** to a trolley story?

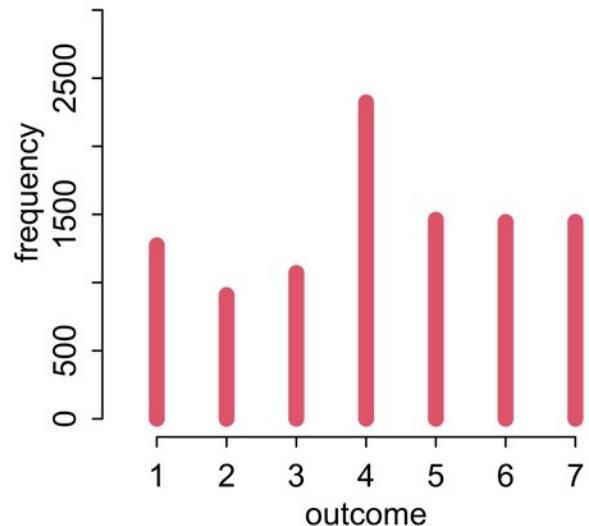
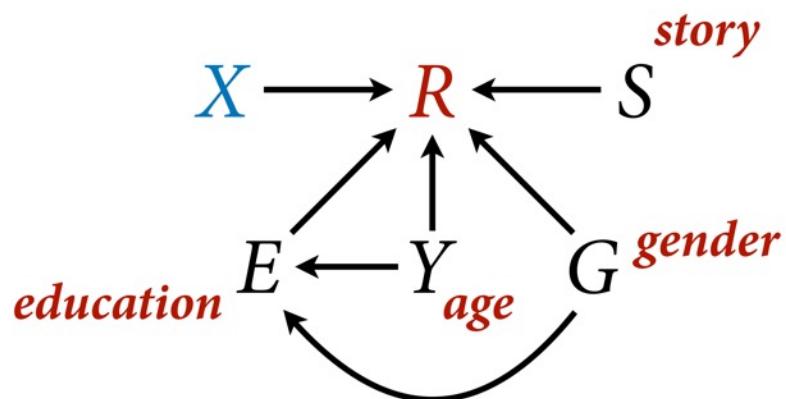
*treatment*

$X \longrightarrow R$   
*response*



Estimand: How do **action**, **intention**, **contact** influence **response** to a trolley story?

How are influences of A/I/C associated with other variables?



# Ordered categories

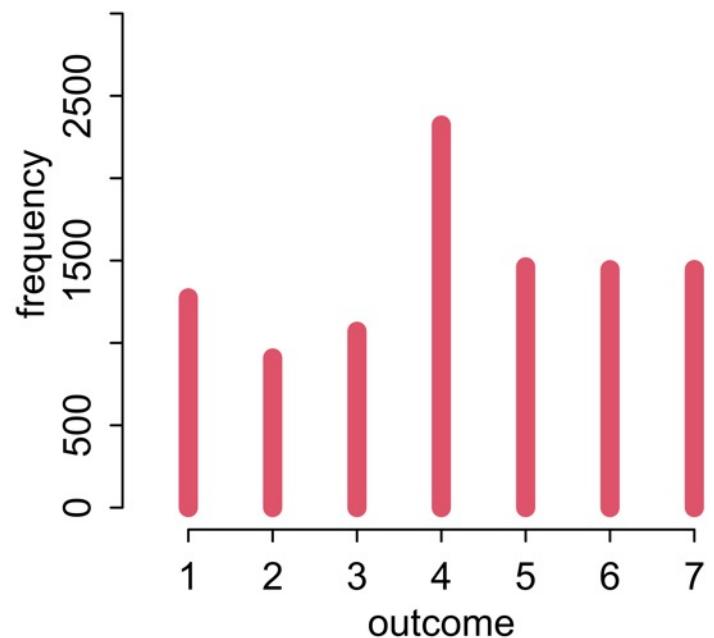
Categories: Discrete types

cat, dog, chicken

Ordered categories: Discrete types  
with ordered relationships

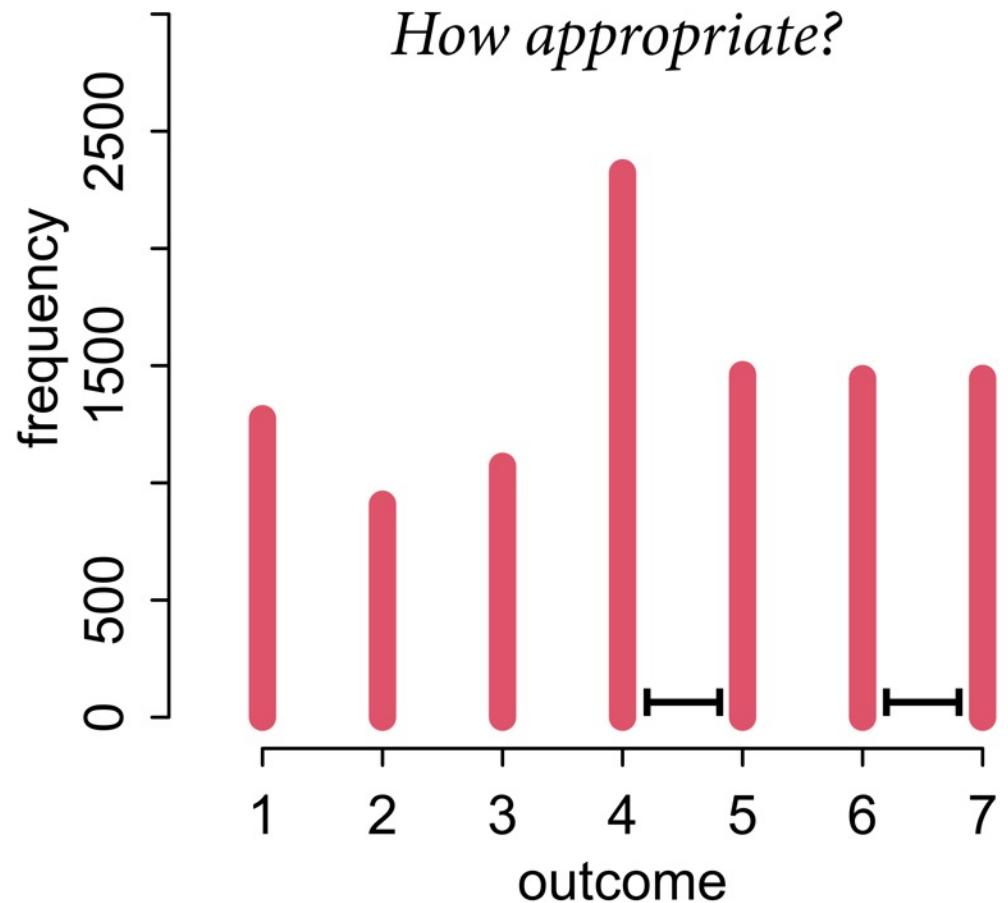
bad, good, excellent

*Ordered categorical*



Distance between values  
not constant

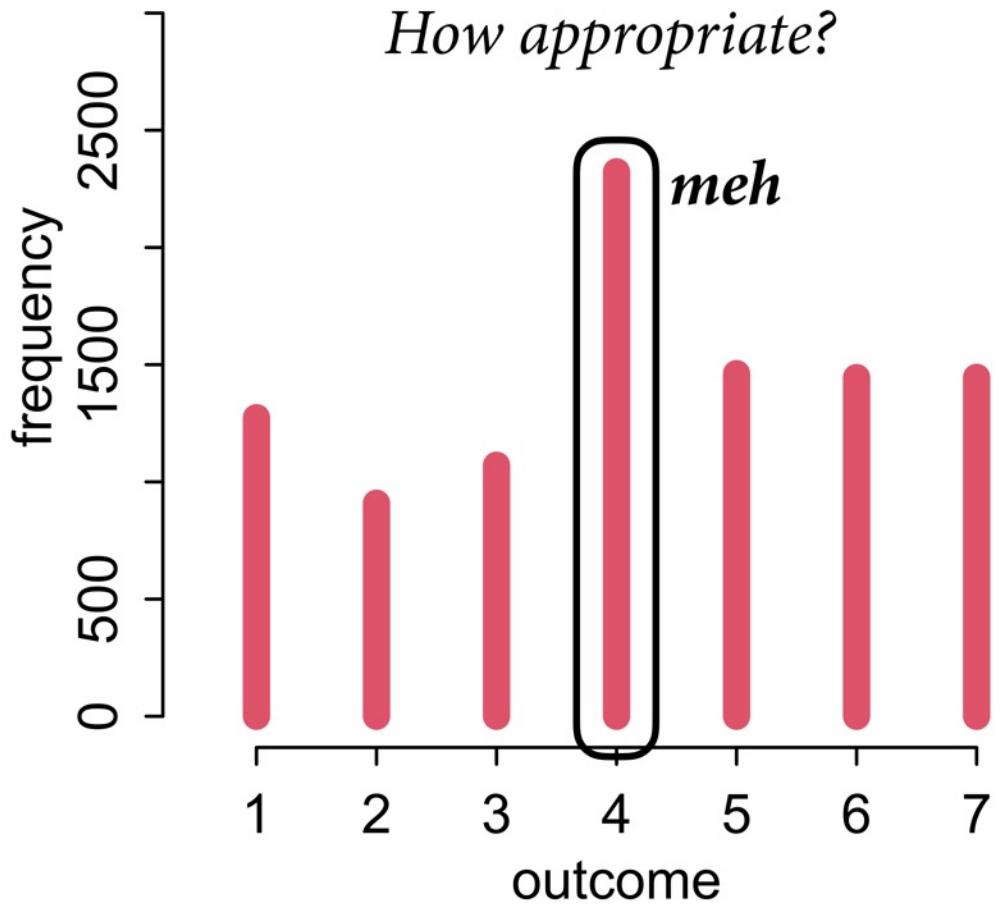
Probably much easier to  
go from 4 to 5 than from  
6 to 7



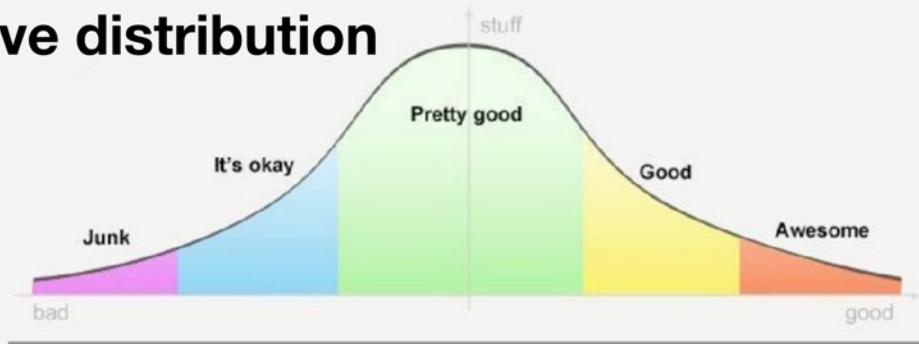
Anchor points common

Not everyone shares the same anchor points

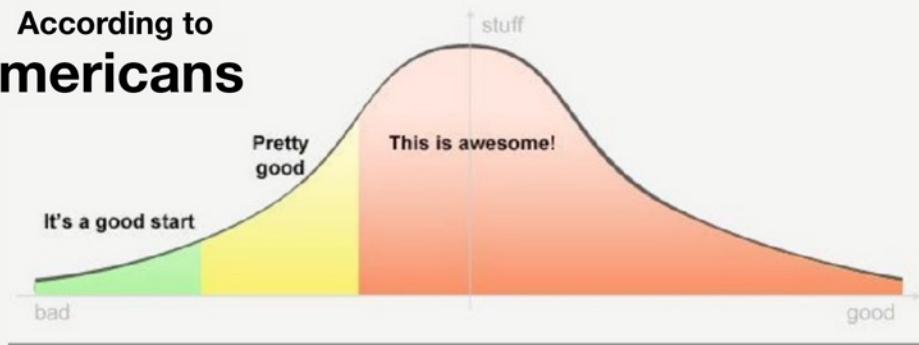
*How appropriate?*



# Objective distribution

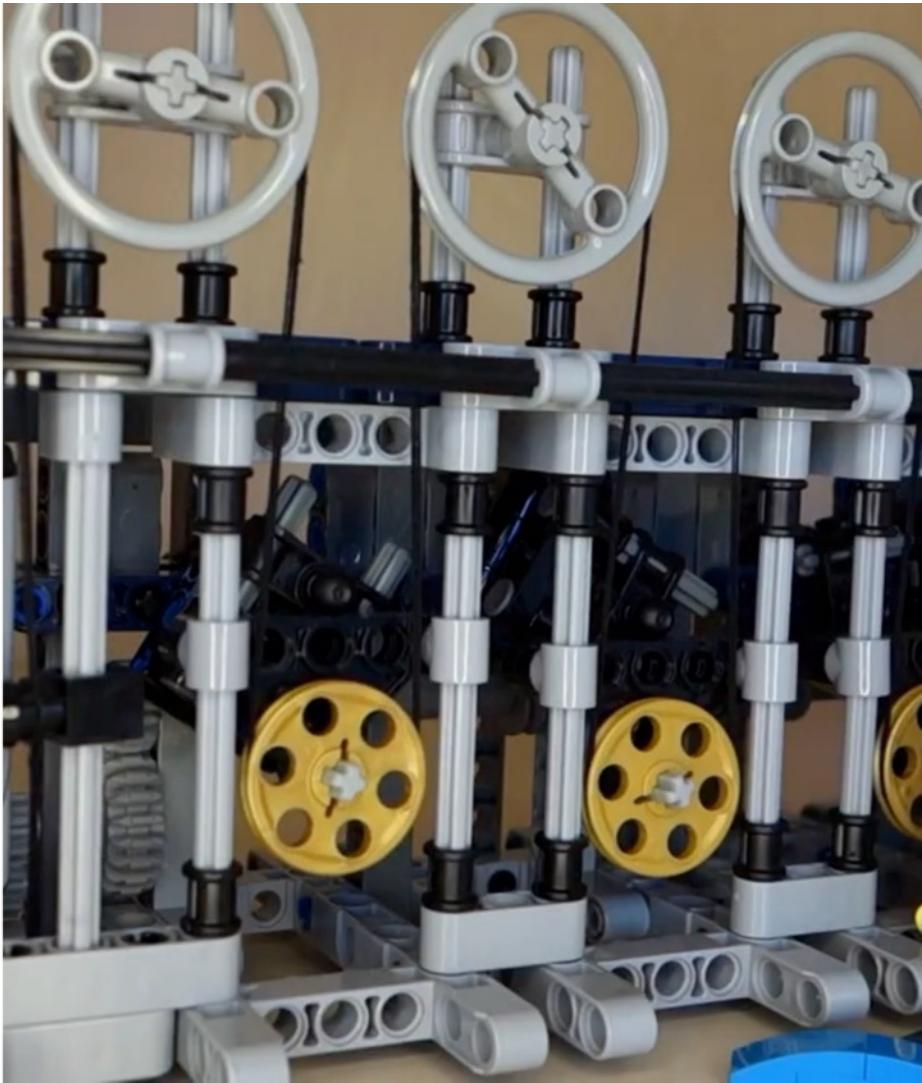
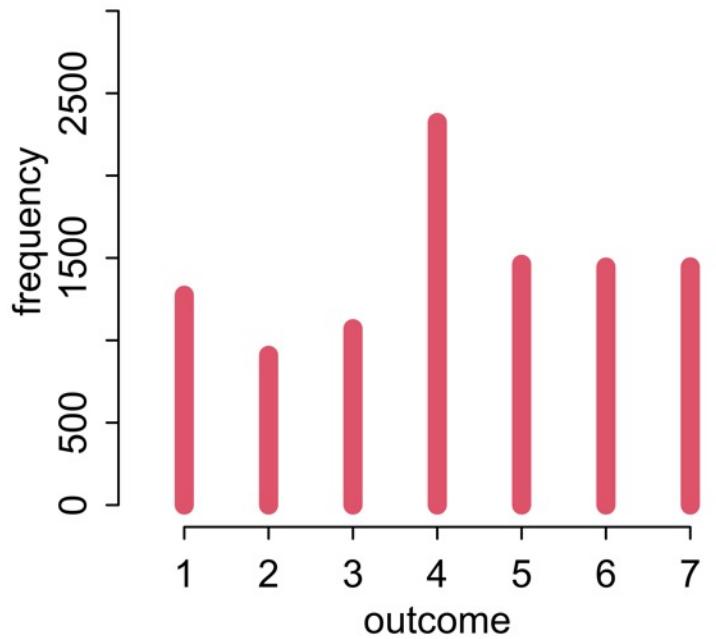


## According to Americans

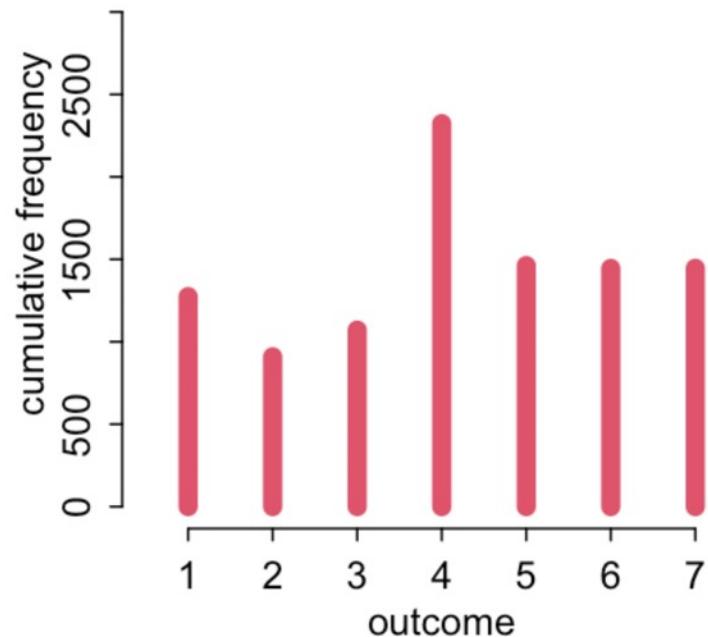
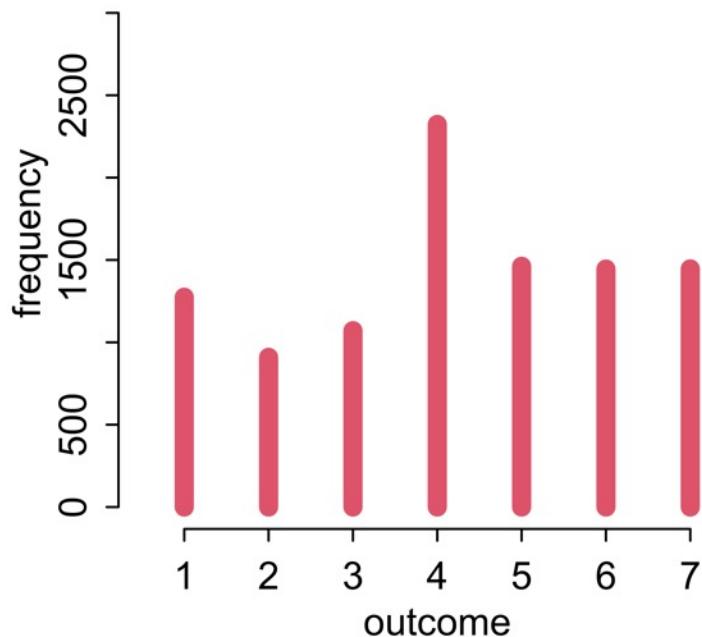


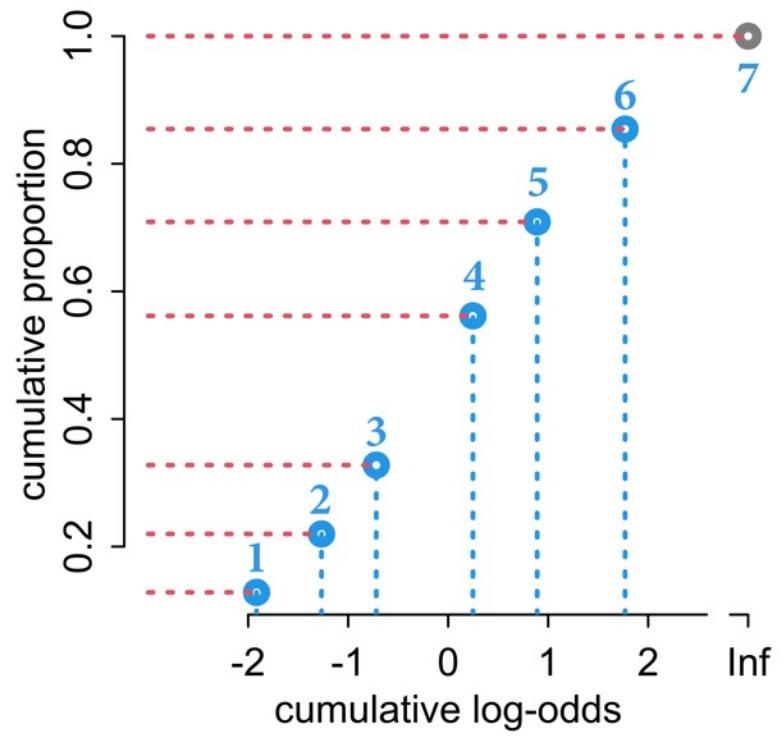
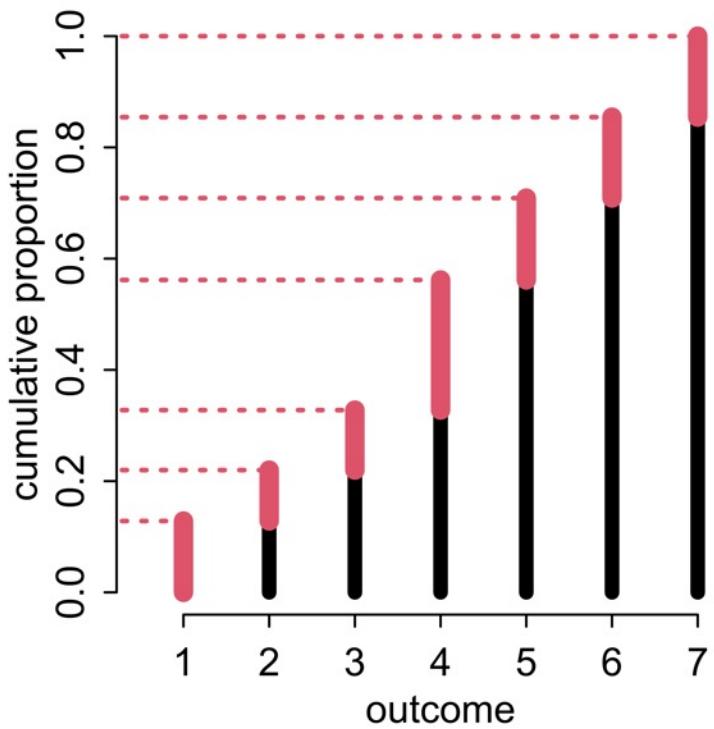
## According to Eastern Europeans

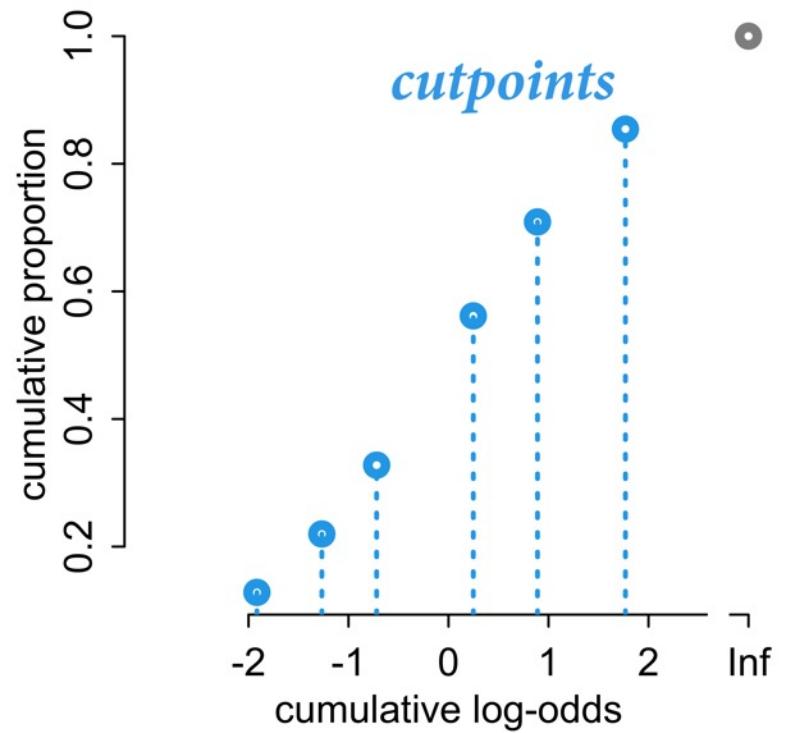
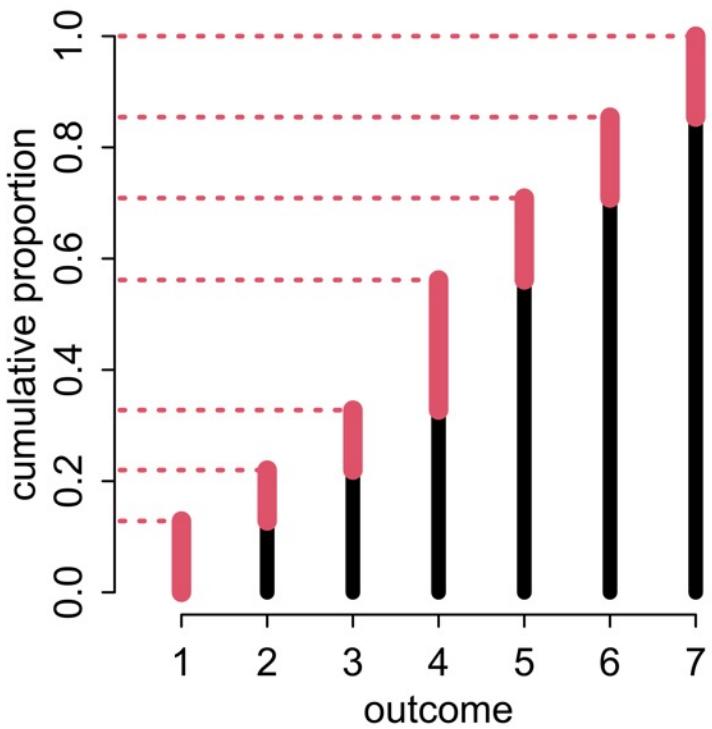


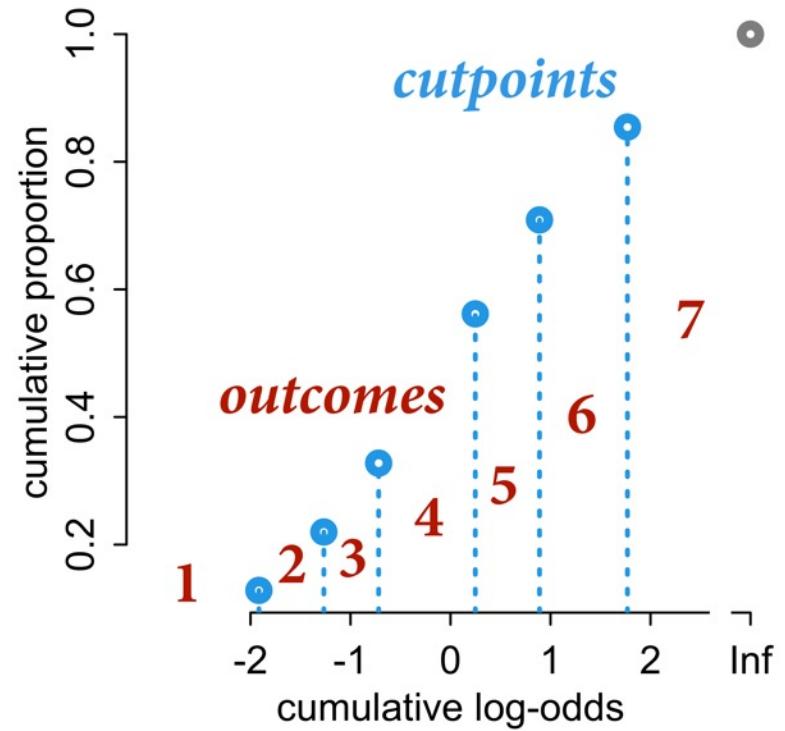
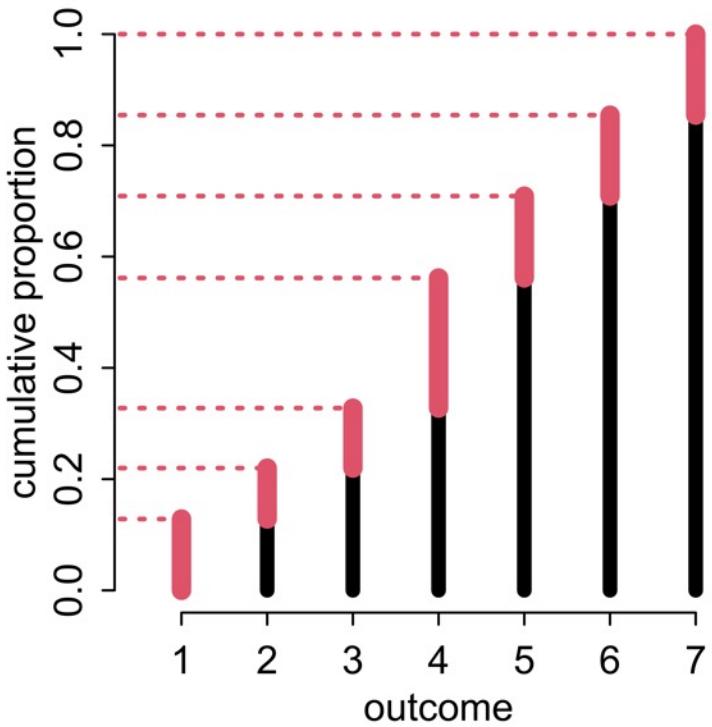


# Ordered = Cumulative

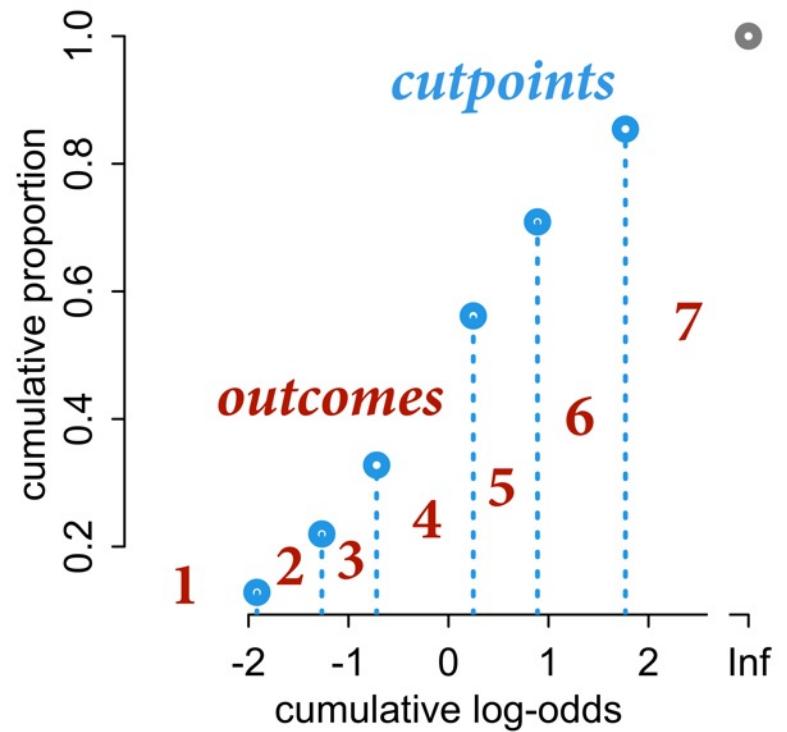




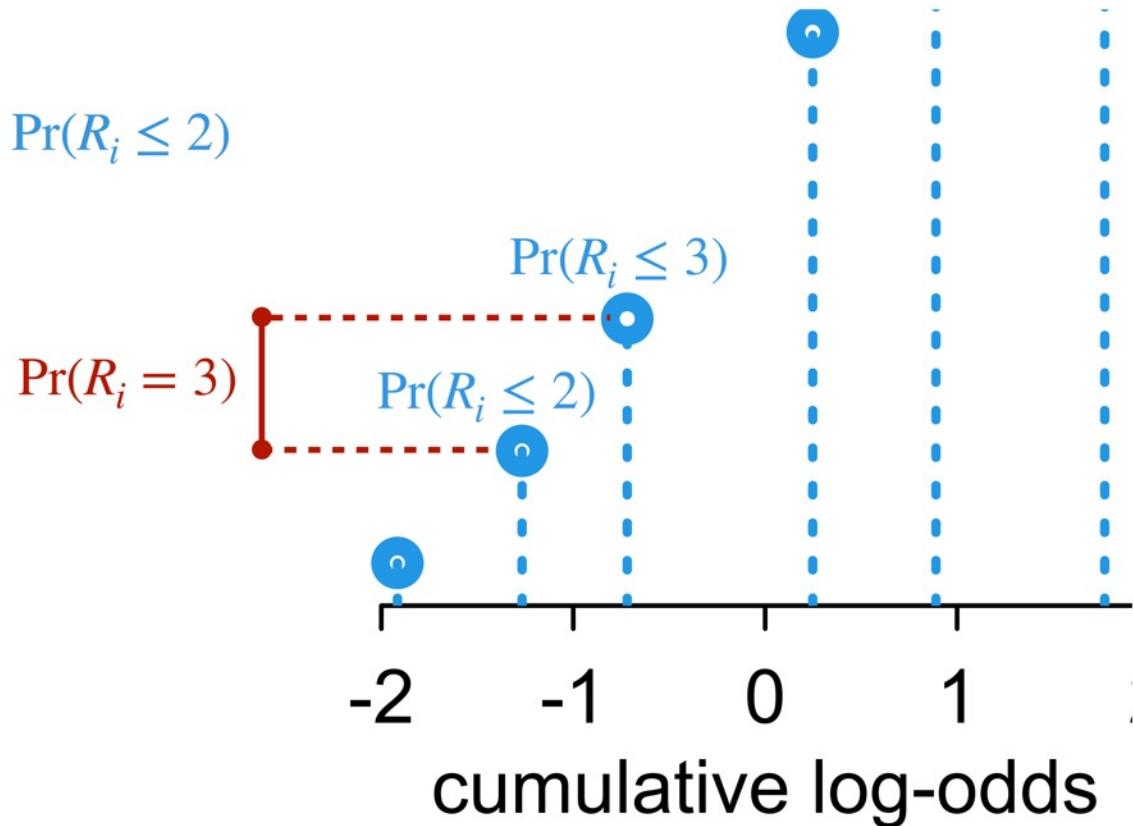




$$\Pr(R_i = k) = \Pr(R_i \leq k) - \Pr(R_i \leq k - 1)$$



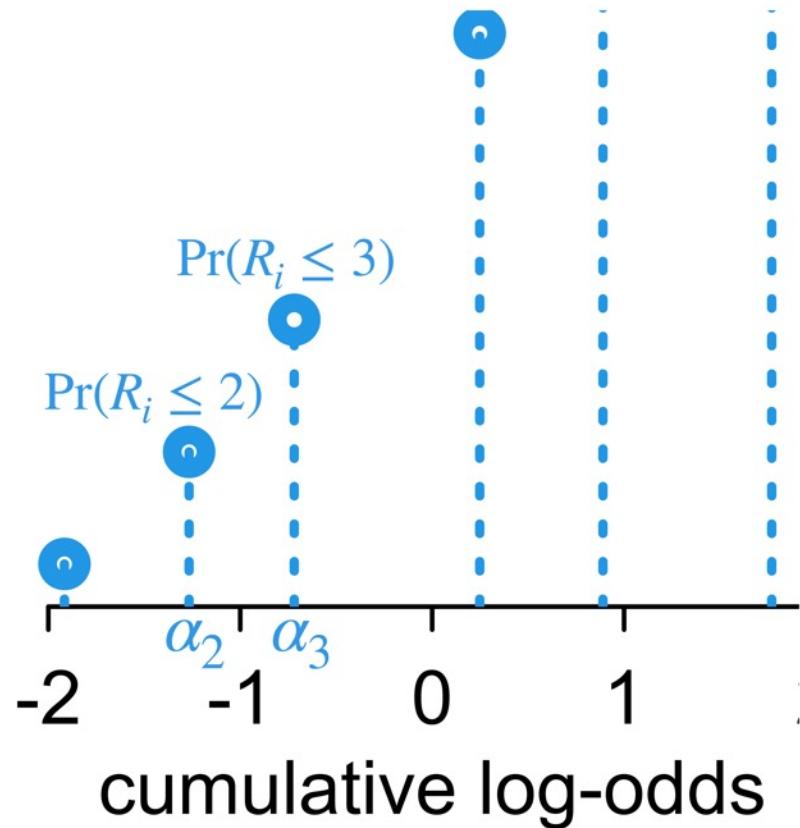
$$\Pr(R_i = 3) = \Pr(R_i \leq 3) - \Pr(R_i \leq 2)$$



$$\Pr(R_i = 3) = \Pr(R_i \leq 3) - \Pr(R_i \leq 2)$$

$$\frac{\log \frac{\Pr(R_i \leq k)}{1 - \Pr(R_i \leq k)}}{\text{cumulative log-odds}} = \alpha_k$$

*cutpoint  
(to estimate)*



# Where's the GLM?

So far just estimating the histogram

How to make it a function of variables?

(1) Stratify cutpoints

(2) Offset each cutpoint by value of linear model  $\phi_i$

# Where's the GLM?

So far just estimating the histogram

How to make it a function of variables?

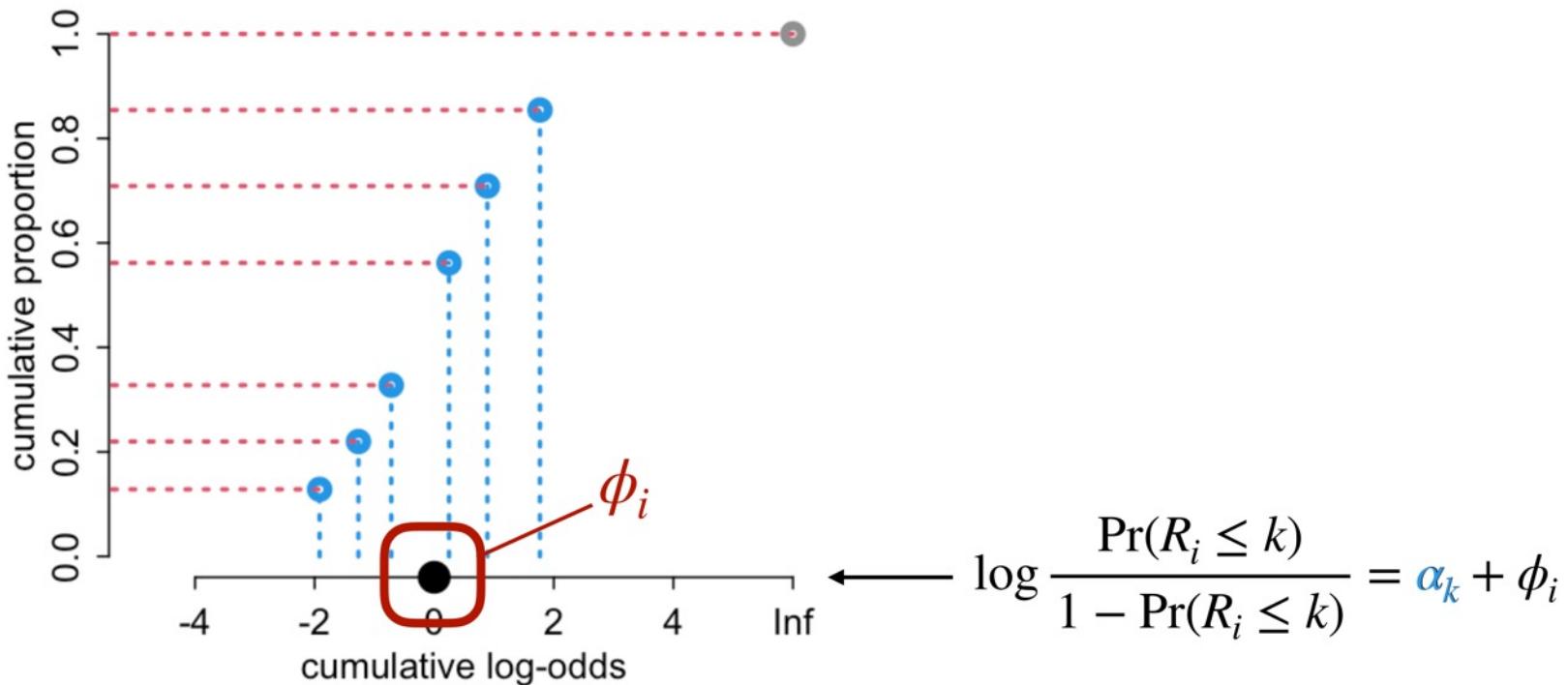
(1) Stratify cutpoints

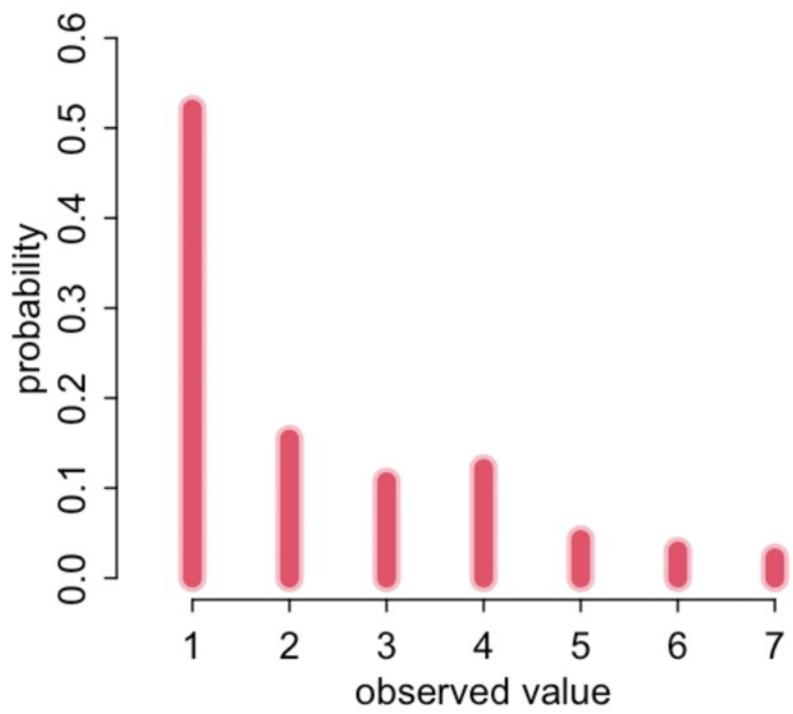
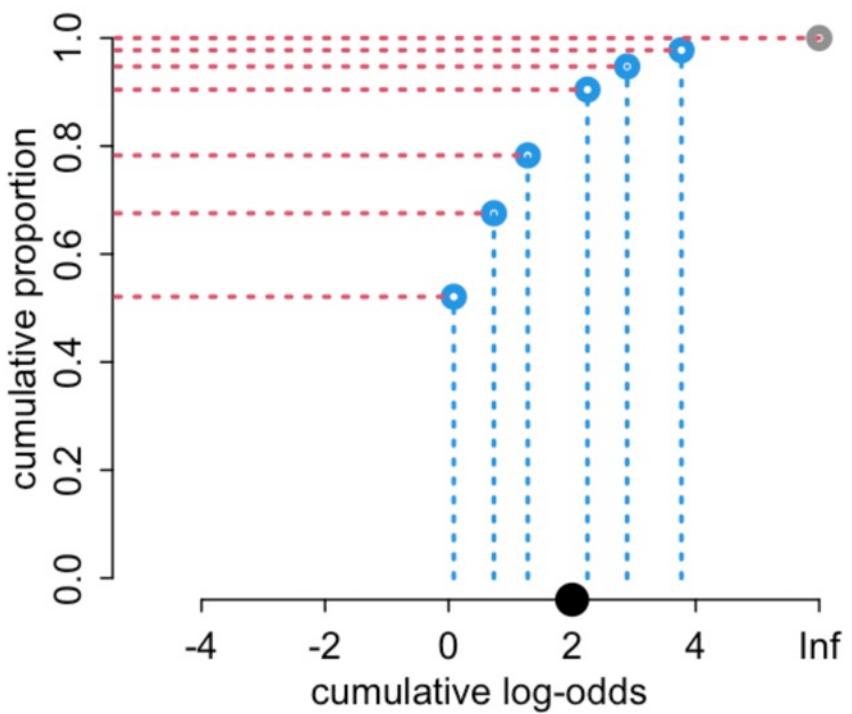
(2) Offset each cutpoint by value of linear model  $\phi_i$

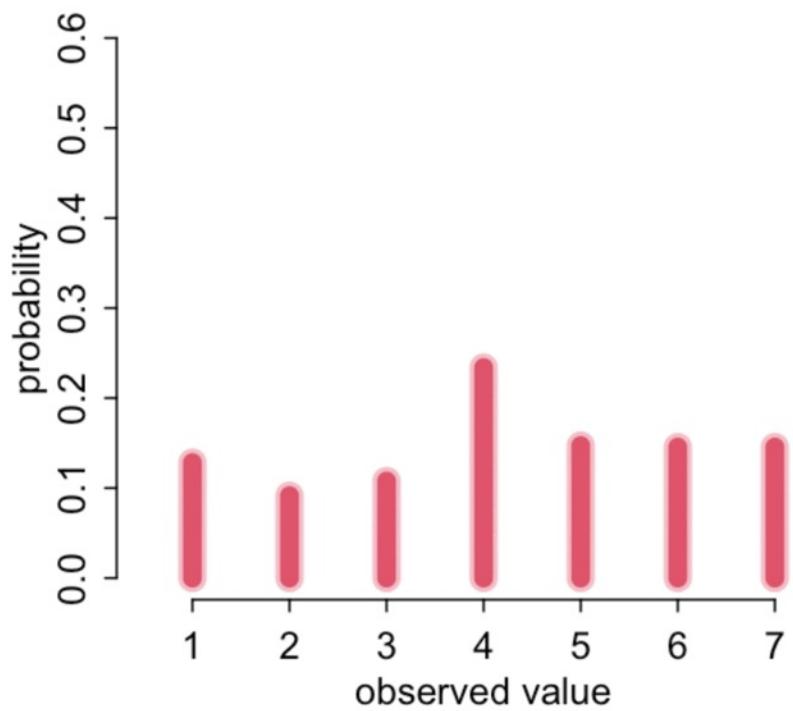
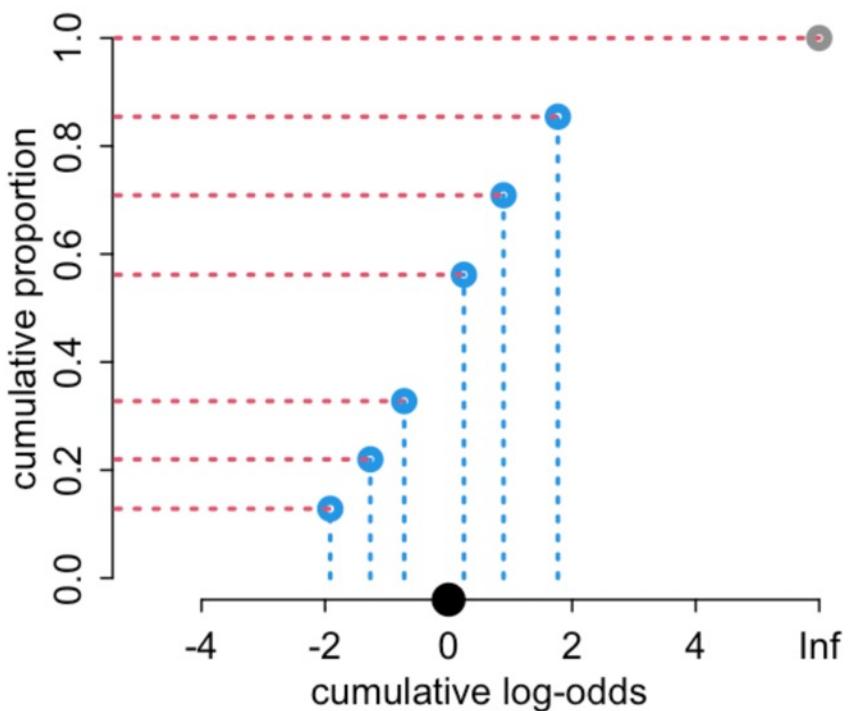
$$\phi_i = \beta x_i$$

$$\log \frac{\Pr(R_i \leq k)}{1 - \Pr(R_i \leq k)} = \alpha_k + \phi_i$$

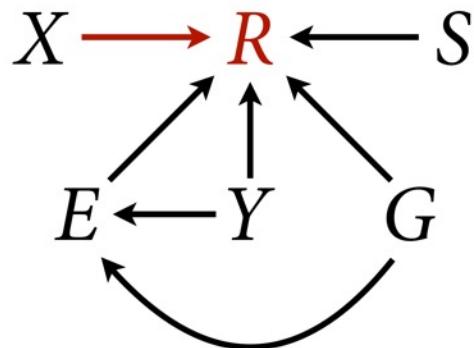
$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$







Start off easy:



$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_A A_i + \beta_C C_i + \beta_I I_i$$

$$\beta_- \sim \text{Normal}(0, 0.5)$$

$$\alpha_j \sim \text{Normal}(0, 1)$$

```

data(Trolley)
d <- Trolley
dat <- list(
  R = d$response,
  A = d$action,
  I = d$intention,
  C = d$contact
)
mRX <- ulam(
  alist(
    R ~ dordlogit(phi,alpha),
    phi <- bA*A + bI*I + bC*C,
    c(bA,bI,bC) ~ normal(0,0.5),
    alpha ~ normal(0,1)
  ) , data=dat , chains=4 , cores=4 )

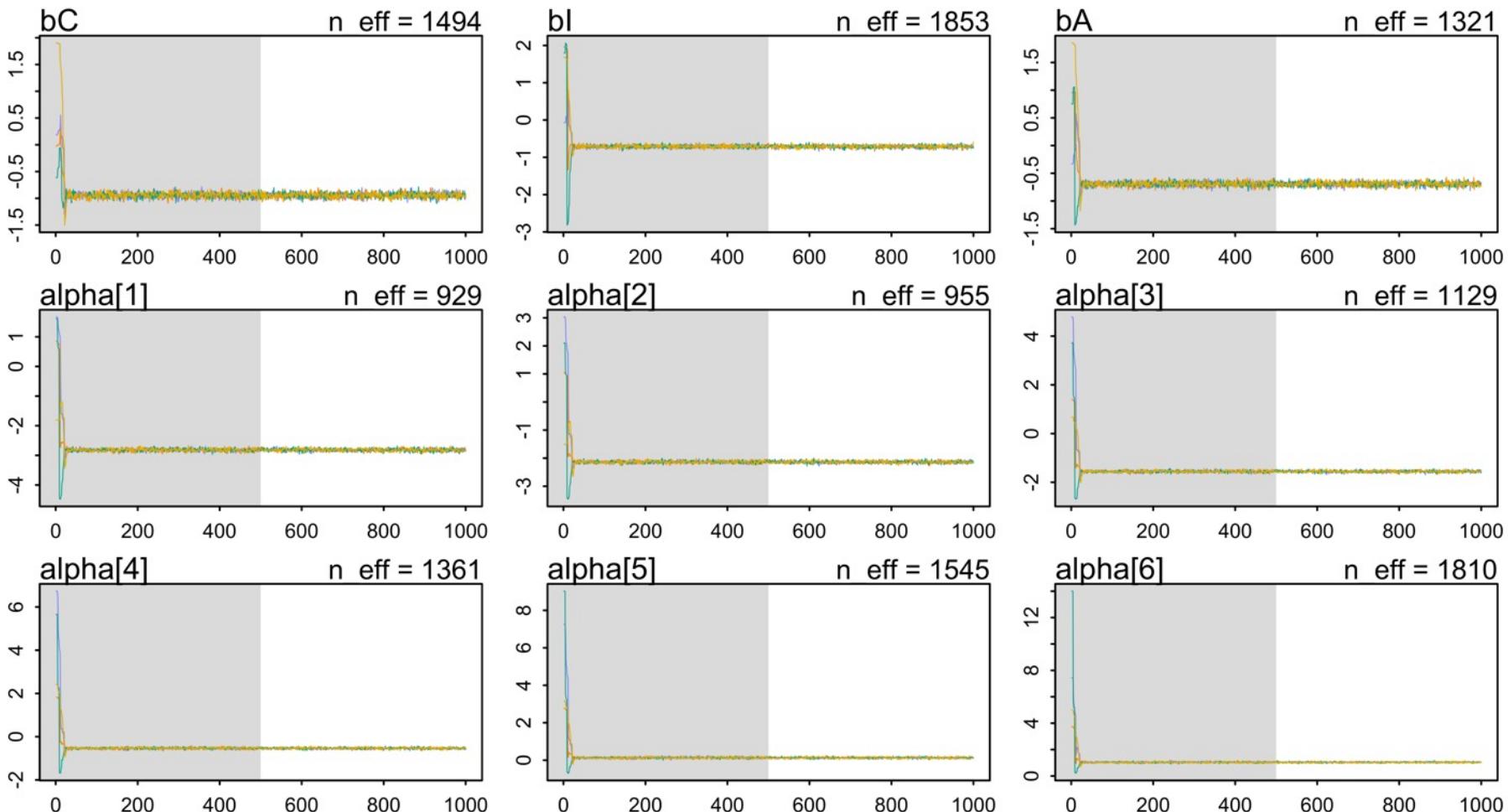
```

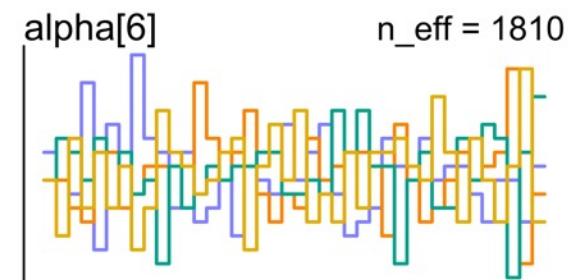
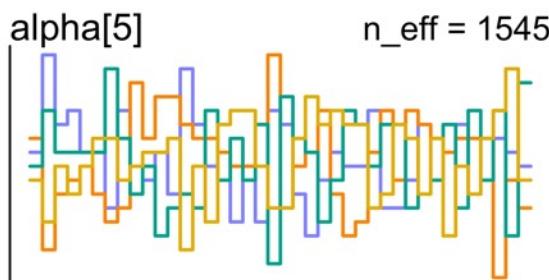
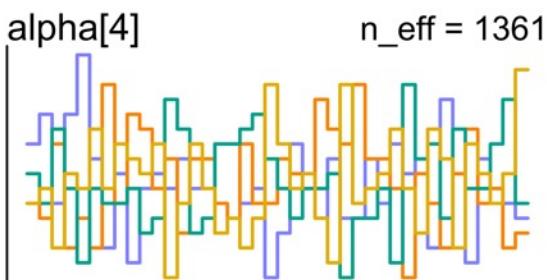
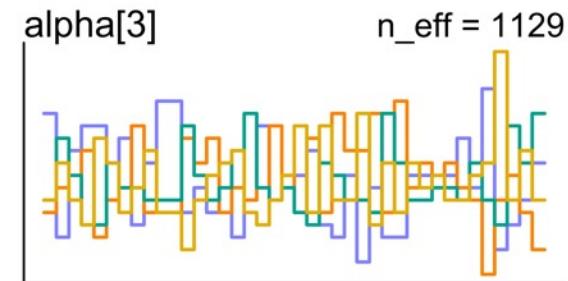
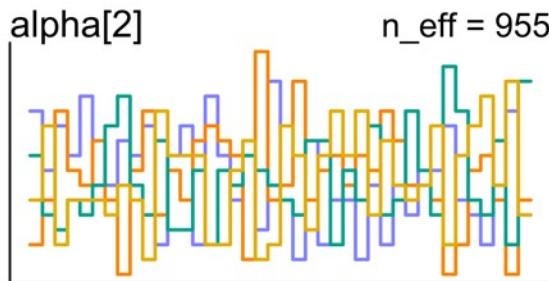
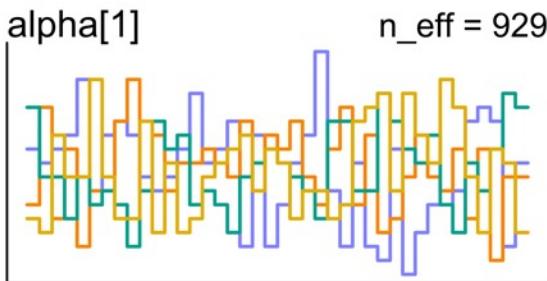
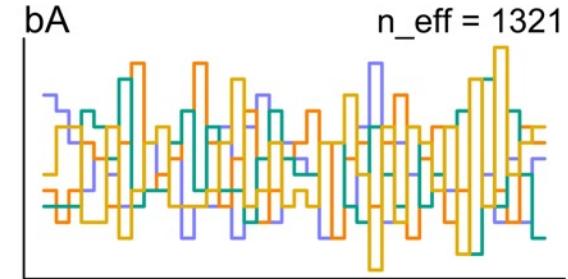
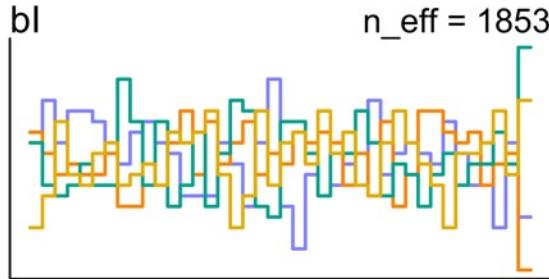
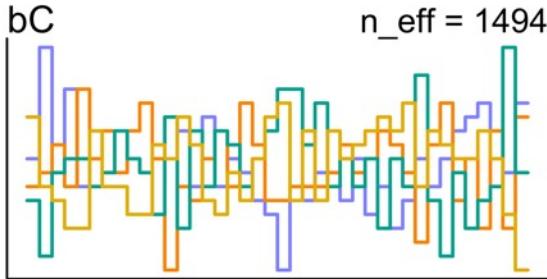
$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_A A_i + \beta_C C_i + \beta_I I_i$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

$$\alpha_j \sim \text{Normal}(0,1)$$





```

data(Trolley)
d <- Trolley
dat <- list(
  R = d$response,
  A = d$action,
  I = d$intention,
  C = d$contact
)

mRX <- ulam(
  alist(
    R ~ dordlogit(phi,alpha),
    phi <- bA*A + bI*I + bC*C,
    c(bA,bI,bC) ~ normal(0,0.5),
    alpha ~ normal(0,1)

  ) , data=dat , chains=4 , cores=4 )

```

```

> precis(mRX,2)
      mean   sd  5.5% 94.5% n_eff Rhat4
bC     -0.94 0.05 -1.02 -0.87  1494    1
bI     -0.71 0.04 -0.77 -0.65  1853    1
bA     -0.69 0.04 -0.76 -0.63  1321    1
alpha[1] -2.82 0.05 -2.89 -2.74   929    1
alpha[2] -2.14 0.04 -2.20 -2.07   955    1
alpha[3] -1.56 0.04 -1.62 -1.49  1129    1
alpha[4] -0.54 0.04 -0.59 -0.48  1361    1
alpha[5]  0.13 0.04  0.07  0.19  1545    1
alpha[6]  1.04 0.04  0.97  1.10  1810    1

```

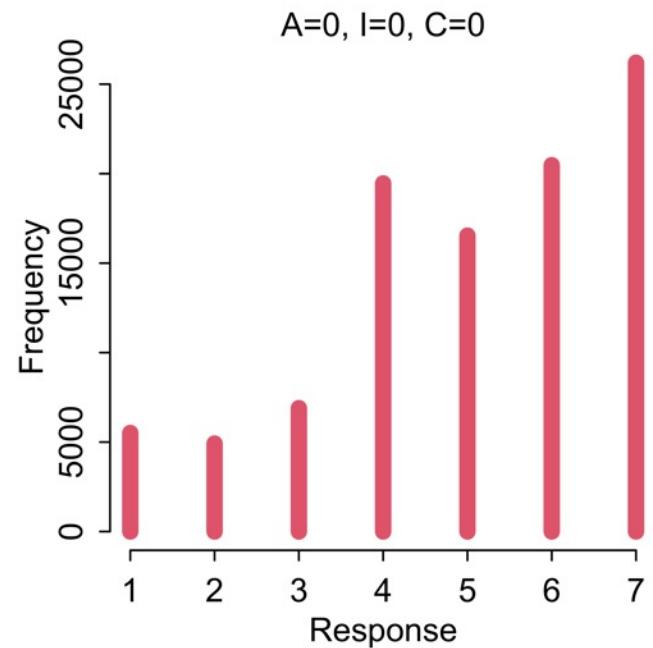
```

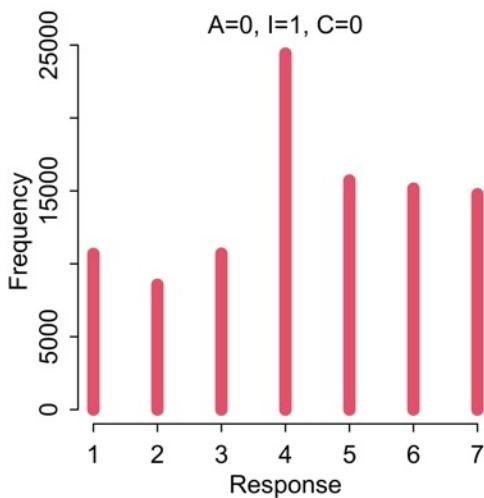
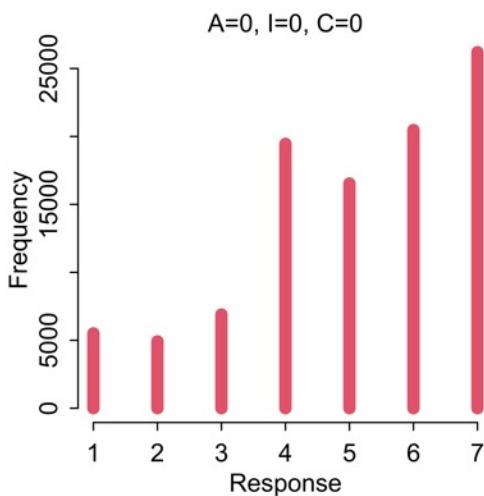
# plot predictive distributions for each treatment

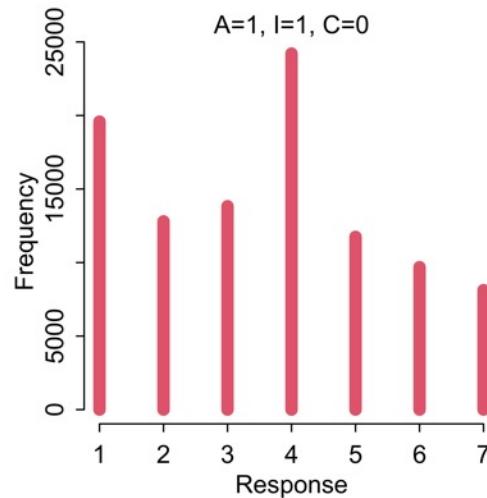
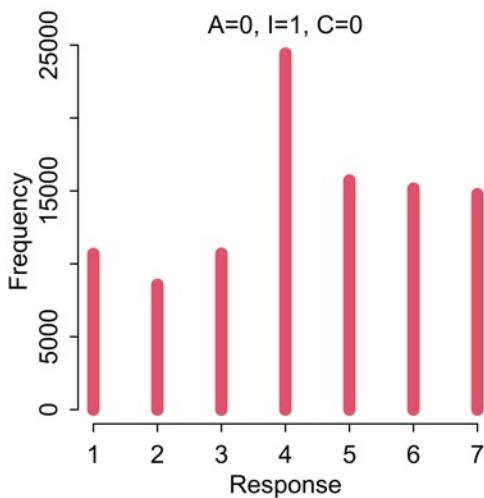
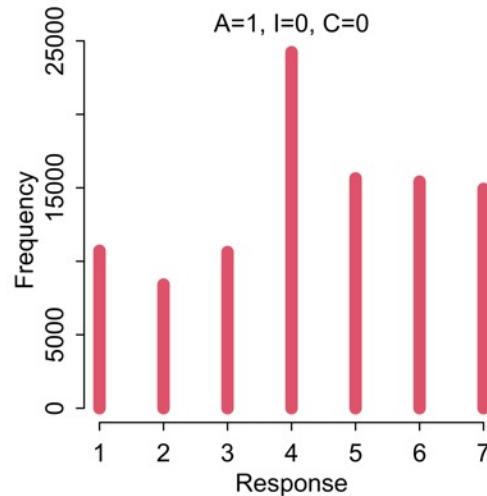
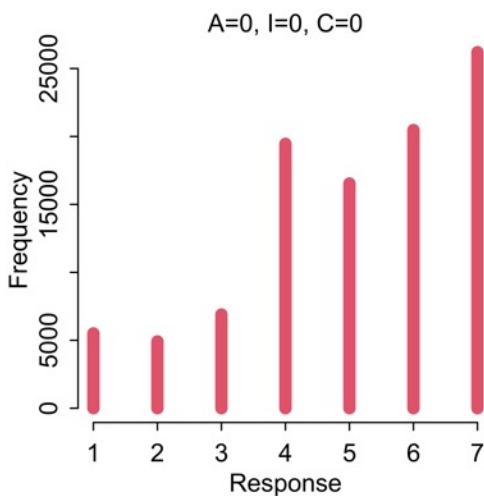
vals <- c(0,0,0)
Rsim <- mcreplicate( 100 ,
sim(mRX,data=list(A=vals[1],I=vals[2],C=vals[3])) ,
mc.cores=6 )

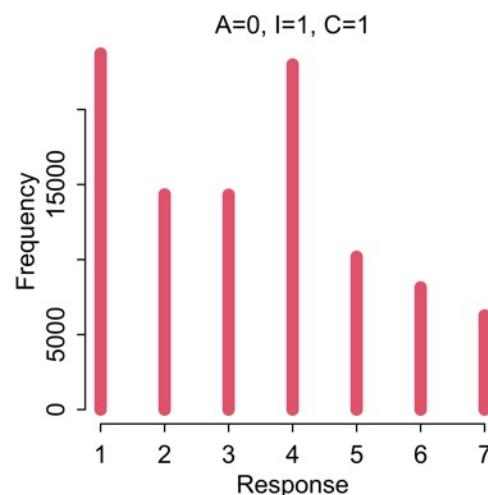
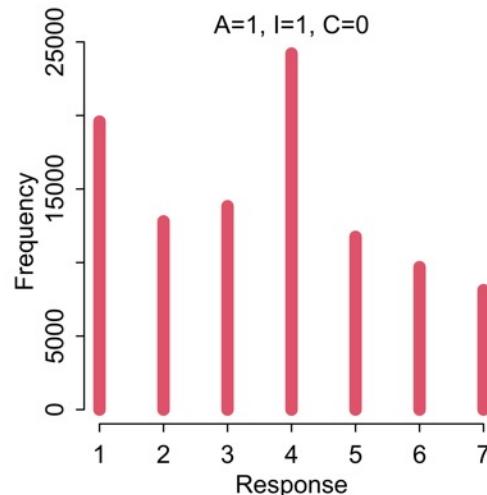
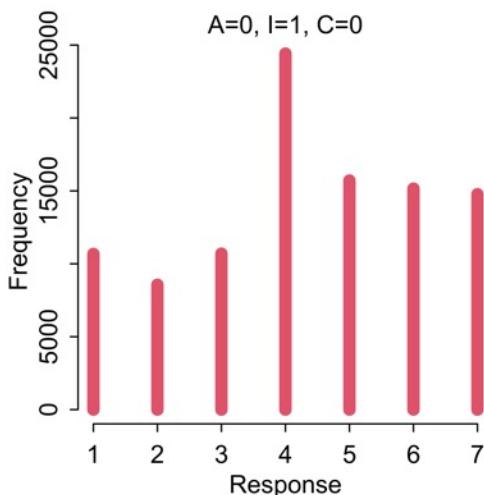
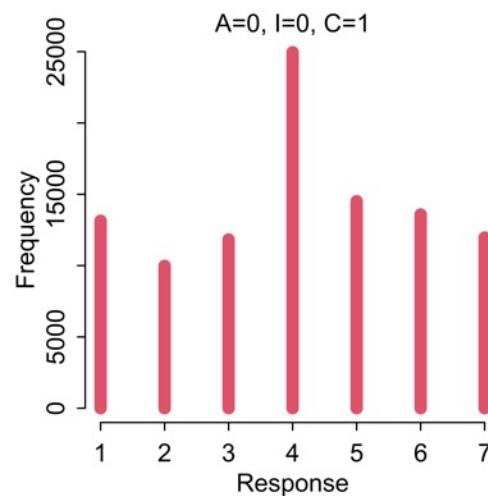
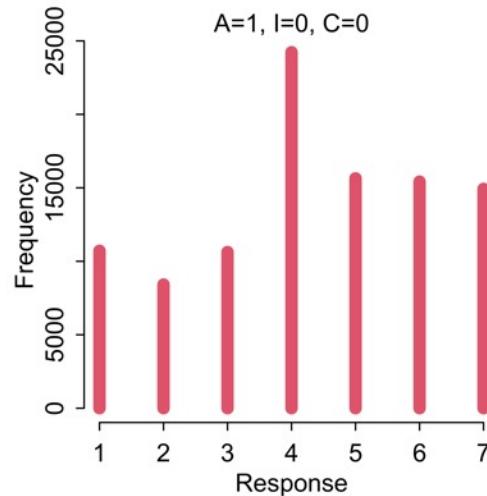
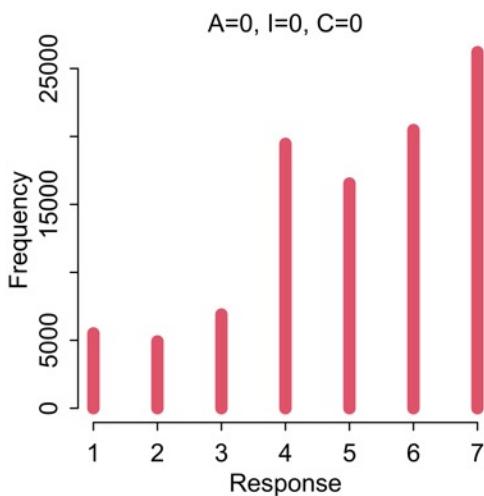
simplehist(as.vector(Rsim),lwd=8,col=2,xlab="Response")
mtext(concat("A=",vals[1]," I=",vals[2]," C=",
C=vals[3]))

```

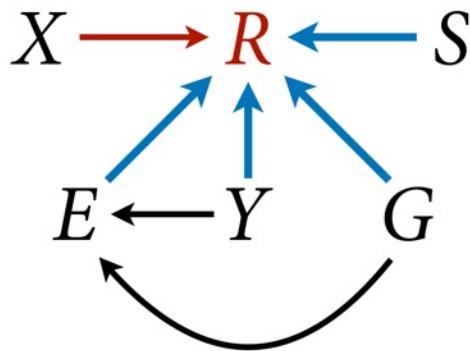








# What about the competing causes?



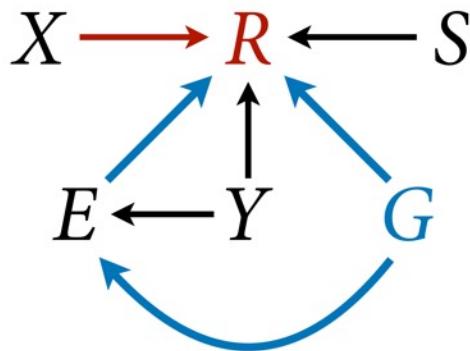
$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_A A_i + \beta_C C_i + \beta_I I_i$$

$$\beta \sim \text{Normal}(0, 0.5)$$

$$\alpha_j \sim \text{Normal}(0, 1)$$

# Total effect of gender:



$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_{A,G[i]} A_i + \beta_{C,G[i]} C_i + \beta_{I,G[i]} I_i$$

$$\beta_- \sim \text{Normal}(0, 0.5)$$

$$\alpha_j \sim \text{Normal}(0, 1)$$

```

# total effect of gender
dat$G <- ifelse(d$male==1,2,1)
mRXG <- ulam(
  alist(
    R ~ dordlogit(phi,alpha),
    phi <- bA[G]*A + bI[G]*I + bC[G]*C,
    bA[G] ~ normal(0,0.5),
    bI[G] ~ normal(0,0.5),
    bC[G] ~ normal(0,0.5),
    alpha ~ normal(0,1)
  ) , data=dat , chains=4 , cores=4 )

```

$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_{A,G[i]} A_i + \beta_{C,G[i]} C_i + \beta_{I,G[i]} I_i$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

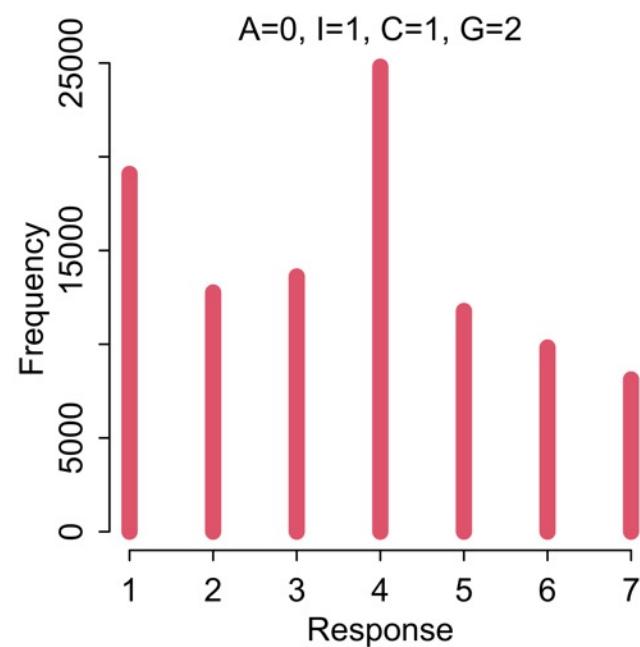
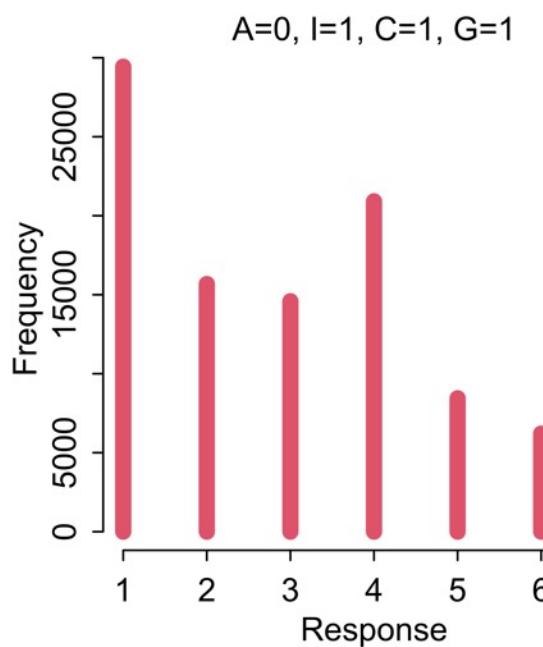
$$\alpha_j \sim \text{Normal}(0,1)$$

```

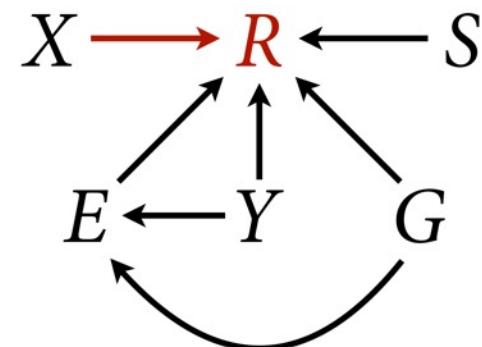
# total effect of gender
dat$G <- ifelse(d$male==1,2,1)
mRXG <- ulam(
  alist(
    R ~ dordlogit(phi,alpha),
    phi <- bA[G]*A + bI[G]*I + bC[G]*C,
    bA[G] ~ normal(0,0.5),
    bI[G] ~ normal(0,0.5),
    bC[G] ~ normal(0,0.5),
    alpha ~ normal(0,1)
  ) , data=dat , chains=4 , cores=4 )

```

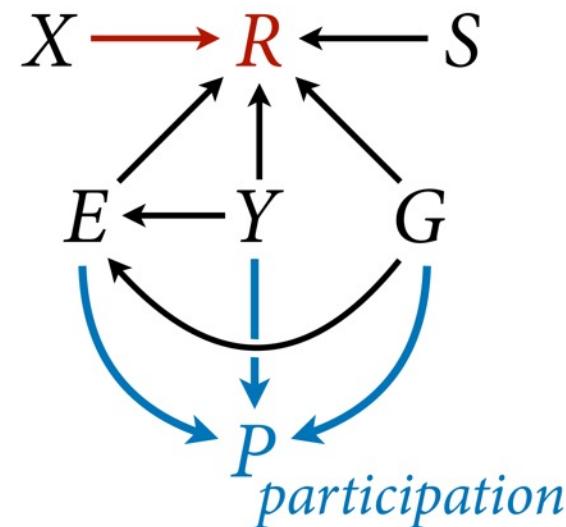
	> precis(mRXG,2)	mean	sd	5.5%	94.5%	n_eff	Rhat4
bA[1]	-0.88	0.05	-0.96	-0.80	1858	1.00	
bA[2]	-0.53	0.05	-0.61	-0.45	1724	1.00	
bI[1]	-0.90	0.05	-0.97	-0.82	2189	1.00	
bI[2]	-0.55	0.05	-0.63	-0.48	2382	1.00	
bC[1]	-1.06	0.07	-1.17	-0.95	2298	1.00	
bC[2]	-0.84	0.06	-0.94	-0.74	2000	1.00	
alpha[1]	-2.83	0.05	-2.90	-2.75	1054	1.01	
alpha[2]	-2.15	0.04	-2.21	-2.08	1104	1.00	
alpha[3]	-1.56	0.04	-1.62	-1.50	1076	1.00	
alpha[4]	-0.53	0.04	-0.59	-0.47	1080	1.00	
alpha[5]	0.14	0.04	0.09	0.20	1216	1.00	
alpha[6]	1.06	0.04	1.00	1.12	1532	1.00	



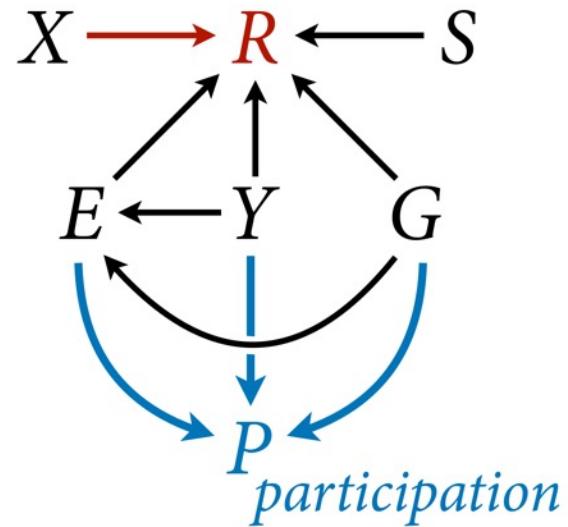
Hang on! This is a voluntary sample



Hang on! This is a **voluntary** sample



Hang on! This is a **voluntary** sample



*Conditioning on  $P$  makes  $E, Y, G$  covary in sample*

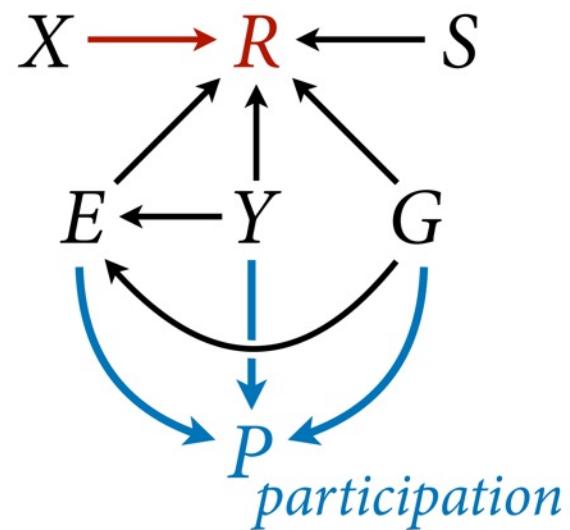
# Endogenous selection

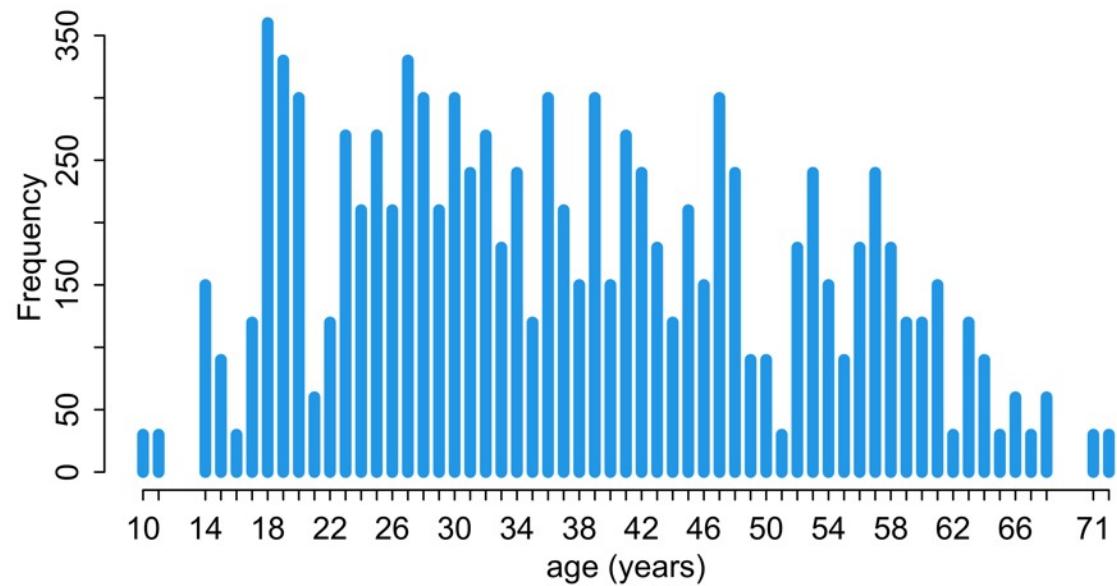
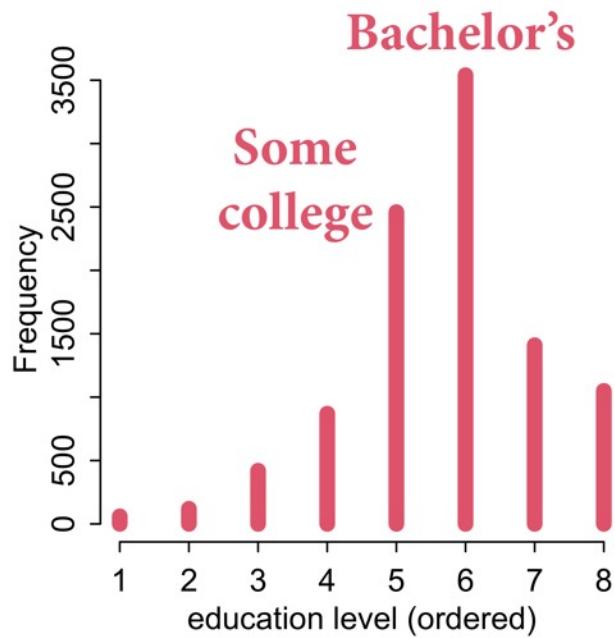
Sample is selected on a collider

Induces misleading associations among variables

Not possible here to estimate total effect of  $G$ , BUT can get direct effect

Need to stratify by  $E$  and  $Y$  and  $G$





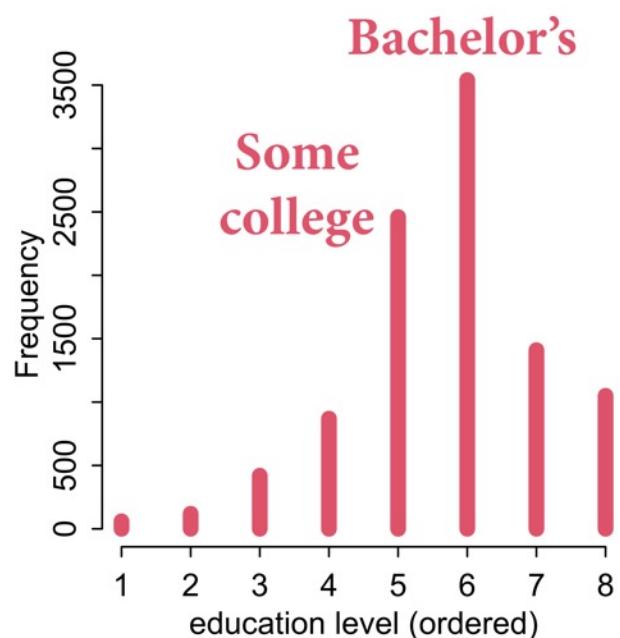
# Ordered monotonic predictors

Education is an ordered category

Unlikely that each level has same effect

Want a parameter for each level

But how to enforce ordering, so that each level has larger (or smaller) effect than previous?



# Ordered monotonic predictors

1 (elementary)  $\phi_i = 0$

2 (middle school)  $\phi_i = \delta_1$

3 (some high school)  $\phi_i = \delta_1 + \delta_2$

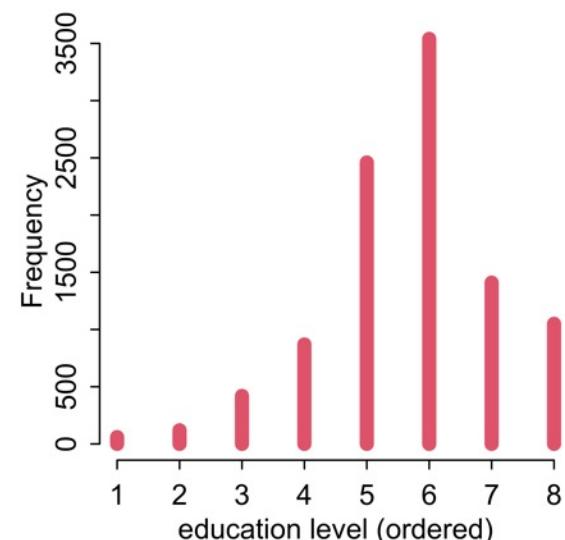
4 (high school)  $\phi_i = \delta_1 + \delta_2 + \delta_3$

5 (some college)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4$

6 (college)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5$

7 (master's)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_6$

8 (doctorate)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_6 + \delta_7$



# Ordered monotonic predictors

1 (elementary)  $\phi_i = 0$

2 (middle school)  $\phi_i = \delta_1$

3 (some high school)  $\phi_i = \delta_1 + \delta_2$

4 (high school)  $\phi_i = \delta_1 + \delta_2 + \delta_3$

5 (some college)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4$

6 (college)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5$

7 (master's)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_6$

8 (doctorate)  $\phi_i = \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_6 + \delta_7 = \beta_E$

*maximum effect  
of education*

# Ordered monotonic predictors

1 (elementary)

2 (middle school)

3 (some high school)

4 (high school)

5 (some college)

6 (college)

7 (master's)

8 (doctorate)

$$\delta_0 = 0$$

$$\sum_{j=0}^7 \delta_j = 1$$

# Ordered monotonic predictors

1 (elementary)

2 (middle school)

3 (some high school)

4 (high school)

5 (some college)

6 (college)

7 (master's)

8 (doctorate)

$$\phi_i = \beta_E \sum_{j=0}^{E_i-1} \delta_j$$

*education level*

*maximum effect*

*proportion of maximum effect*

# Ordered monotonic *priors*

How do we set priors for the delta parameters?

delta parameters form a **simplex**

*Simplex*: vector that sums to 1

$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_E \sum_{j=0}^{E_i-1} \delta_j + \dots$$

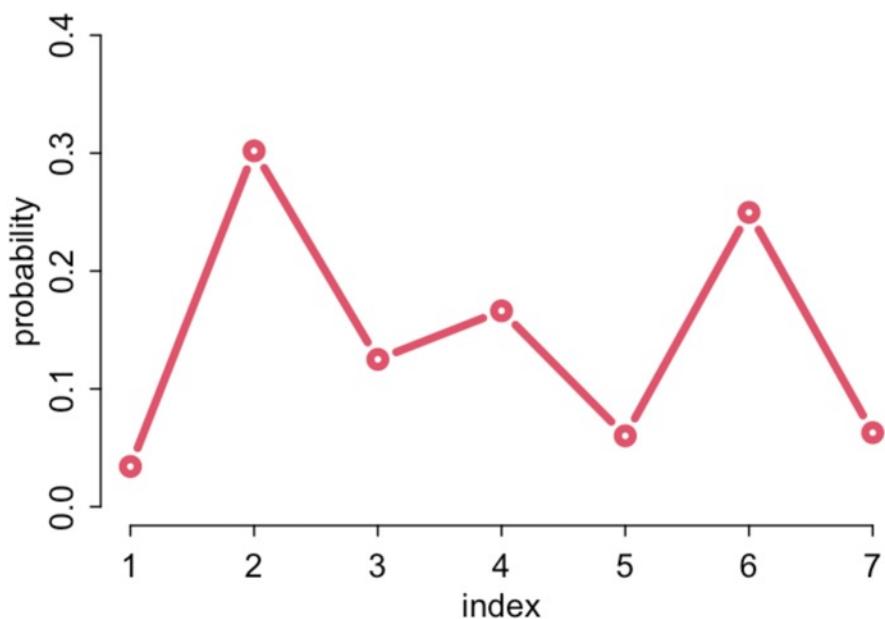
$$\alpha_j \sim \text{Normal}(0,1)$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

$$\delta_j \sim ?$$

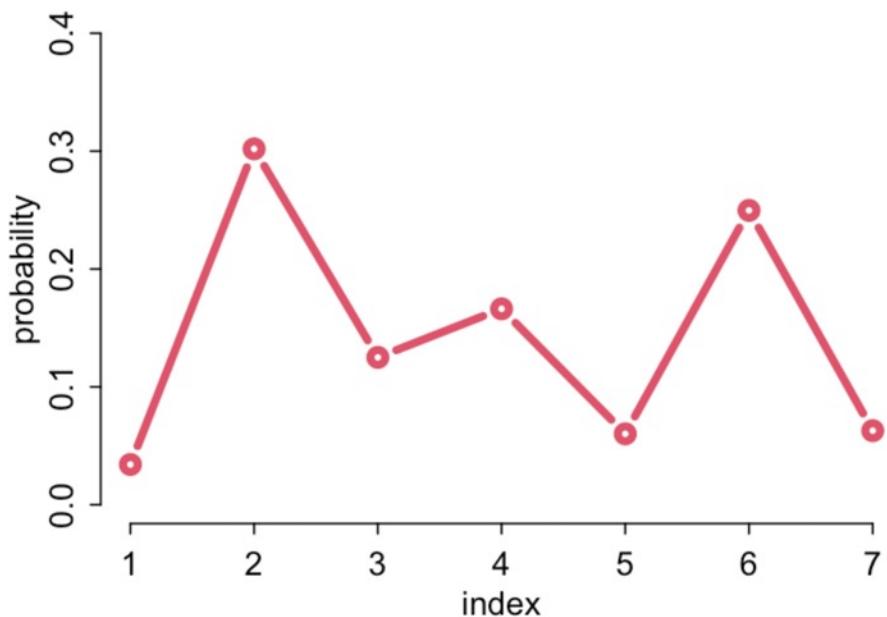
$\delta \sim \text{Dirichlet}(a)$

$a = [2,2,2,2,2,2,2]$



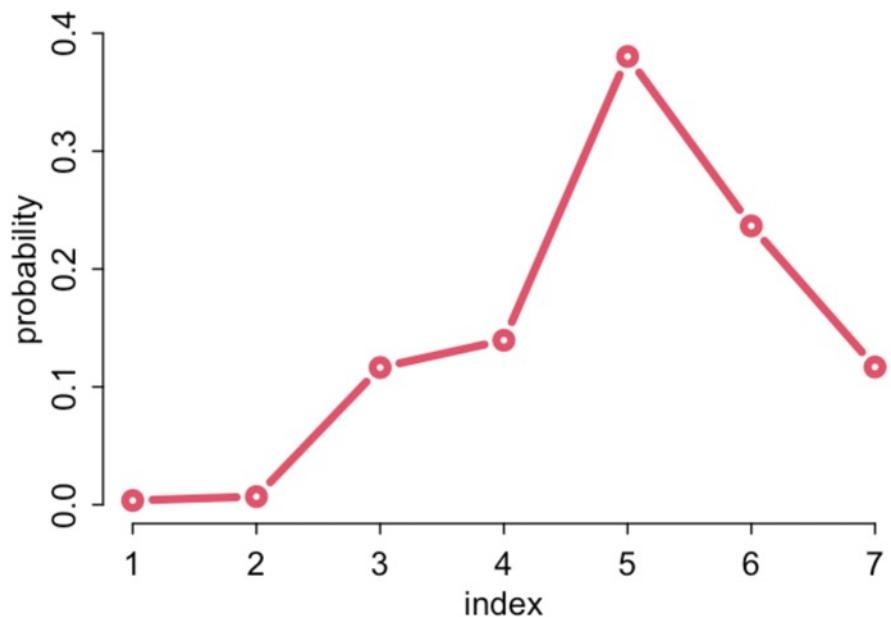
$\delta \sim \text{Dirichlet}(a)$

$a = [2,2,2,2,2,2,2]$



$\delta \sim \text{Dirichlet}(a)$

$a = [1,2,3,4,5,6,7]$



```

edu_levels <- c( 6 , 1 , 8 , 4 , 7 , 2 , 5 , 3 )
edu_new <- edu_levels[ d$edu ]

dat$E <- edu_new
dat$a <- rep(2,7) # dirichlet prior

mRXE <- ulam(
  alist(
    R ~ ordered_logistic( phi , alpha ),
    phi <- bE*sum( delta_j[1:E] ) +
      bA*A + bI*I + bC*C,
    alpha ~ normal( 0 , 1 ),
    c(bA,bI,bC,bE) ~ normal( 0 , 0.5 ),
    vector[8]: delta_j <- append_row( 0 , delta ),
    simplex[7]: delta ~ dirichlet( a )
  ), data=dat , chains=4 , cores=4 )

```

$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_E \sum_{j=0}^{E_i-1} \delta_j + \dots$$

$$\alpha_j \sim \text{Normal}(0,1)$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

$$\delta \sim \text{Dirichlet}(a)$$

```

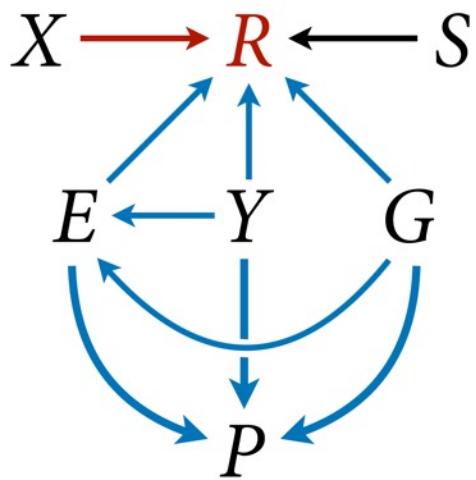
edu_levels <- c( 6 , 1 , 8 , 4 , 7 , 2 , 5 , 3 )
edu_new <- edu_levels[ d$edu ]

dat$E <- edu_new
dat$a <- rep(2,7) # dirichlet prior

mRXE <- ulam(
  alist(
    R ~ ordered_logistic( phi , alpha ),
    phi <- bE*sum( delta_j[1:E] ) +
      bA*A + bI*I + bC*C,
    alpha ~ normal( 0 , 1 ),
    c(bA,bI,bC,bE) ~ normal( 0 , 0.5 ),
    vector[8]: delta_j <- append_row( 0 ,
      simplex[7]: delta ~ dirichlet( a )
  ), data=dat , chains=4 , cores=4 )

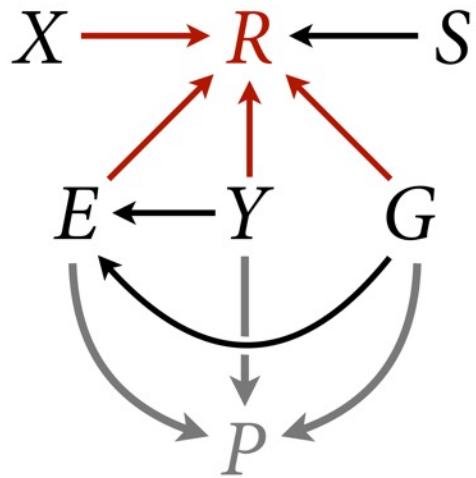
```

	> precis(mRXE,2)	mean	sd	5.5%	94.5%	n_eff	Rhat4
alpha[1]	-3.07	0.14	-3.32	-2.86	793	1	
alpha[2]	-2.39	0.14	-2.63	-2.17	804	1	
alpha[3]	-1.81	0.14	-2.05	-1.60	811	1	
alpha[4]	-0.79	0.14	-1.03	-0.57	799	1	
alpha[5]	-0.12	0.14	-0.36	0.10	804	1	
alpha[6]	0.79	0.14	0.54	1.00	831	1	
bE	-0.31	0.16	-0.57	-0.06	838	1	
bC	-0.96	0.05	-1.04	-0.88	1757	1	
bI	-0.72	0.04	-0.77	-0.66	1982	1	
bA	-0.70	0.04	-0.77	-0.64	1779	1	
delta[1]	0.22	0.13	0.05	0.47	1227	1	
delta[2]	0.14	0.09	0.03	0.31	2258	1	
delta[3]	0.20	0.11	0.05	0.38	2256	1	
delta[4]	0.17	0.09	0.04	0.34	1926	1	
delta[5]	0.04	0.05	0.01	0.12	945	1	
delta[6]	0.10	0.07	0.02	0.23	1870	1	
delta[7]	0.13	0.08	0.03	0.27	2335	1	



*bE not interpretable*

	mean	sd	5.5%	94.5%	n_eff	Rhat4
alpha[1]	-3.07	0.14	-3.32	-2.86	793	1
alpha[2]	-2.39	0.14	-2.63	-2.17	804	1
alpha[3]	-1.81	0.14	-2.05	-1.60	811	1
alpha[4]	-0.79	0.14	-1.03	-0.57	799	1
alpha[5]	-0.12	0.14	-0.36	0.10	804	1
alpha[6]	0.79	0.14	0.54	1.00	831	1
bE	-0.31	0.16	-0.57	-0.06	838	1
bC	-0.96	0.05	-1.04	-0.88	1757	1
bI	-0.72	0.04	-0.77	-0.66	1982	1
bA	-0.70	0.04	-0.77	-0.64	1779	1
delta[1]	0.22	0.13	0.05	0.47	1227	1
delta[2]	0.14	0.09	0.03	0.31	2258	1
delta[3]	0.20	0.11	0.05	0.38	2256	1
delta[4]	0.17	0.09	0.04	0.34	1926	1
delta[5]	0.04	0.05	0.01	0.12	945	1
delta[6]	0.10	0.07	0.02	0.23	1870	1
delta[7]	0.13	0.08	0.03	0.27	2335	1



$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\begin{aligned}\phi_i = & \beta_{E,G[i]} \sum_{j=0}^{E_i-1} \delta_j + \\ & \beta_{A,G[i]} A_i + \beta_{I,G[i]} I_i + \beta_{C,G[i]} C_i + \\ & \beta_{Y,G[i]} Y_i\end{aligned}$$

$$\alpha_j \sim \text{Normal}(0,1)$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

$$\delta \sim \text{Dirichlet}(a)$$

```

dat$Y <- standardize(d$age)

mRXEYGt <- ulam(
  alist(
    R ~ ordered_logistic( phi , alpha ),
    phi <- bE[G]*sum( delta_j[1:E] ) +
      bA[G]*A + bI[G]*I + bC[G]*C +
      bY[G]*Y,
    alpha ~ normal( 0 , 1 ),
    bA[G] ~ normal( 0 , 0.5 ),
    bI[G] ~ normal( 0 , 0.5 ),
    bC[G] ~ normal( 0 , 0.5 ),
    bE[G] ~ normal( 0 , 0.5 ),
    bY[G] ~ normal( 0 , 0.5 ),
    vector[8]: delta_j <- append_row( 0 , delta ),
    simplex[7]: delta ~ dirichlet( a )
  ), data=dat , chains=4 , cores=4 , threads=2 )

```

$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_{E,G[i]} \sum_{j=0}^{E_i-1} \delta_j +$$

$$\beta_{A,G[i],i} A_i + \beta_{I,G[i]} I_i + \beta_{C,G[i]} C_i +$$

$$\beta_{Y,G[i]} Y_i$$

$$\alpha_j \sim \text{Normal}(0,1)$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

$$\delta \sim \text{Dirichlet}(a)$$

```

dat$Y <- standardize(d$age)

mRXEYgt <- ulam(
  alist(
    R ~ ordered_logistic( phi , alpha ),
    phi <- bE[G]*sum( delta_j[1:E] ) +
      bA[G]*A + bI[G]*I + bC[G]*C +
      bY[G]*Y,
    alpha ~ normal( 0 , 1 ),
    bA[G] ~ normal( 0 , 0.5 ),
    bI[G] ~ normal( 0 , 0.5 ),
    bC[G] ~ normal( 0 , 0.5 ),
    bE[G] ~ normal( 0 , 0.5 ),
    bY[G] ~ normal( 0 , 0.5 ),
    vector[8]: delta_j <- append_row( 0 , delta ),
    simplex[7]: delta ~ dirichlet( a )
  ), data=dat , chains=4 , cores=4 , threads=2 )

```

*4 chains times 2 threads each = 8 cores*

$$R_i \sim \text{OrderedLogit}(\phi_i, \alpha)$$

$$\phi_i = \beta_{E,G[i]} \sum_{j=0}^{E_i-1} \delta_j +$$

$$\beta_{A,G[i],i} A_i + \beta_{I,G[i]} I_i + \beta_{C,G[i]} C_i +$$

$$\beta_{Y,G[i]} Y_i$$

$$\alpha_j \sim \text{Normal}(0,1)$$

$$\beta_- \sim \text{Normal}(0,0.5)$$

$$\delta \sim \text{Dirichlet}(a)$$

```

dat$Y <- standardize(d$age)

mRXEYGt <- ulam(
  alist(
    R ~ ordered_logistic( phi , alpha ),
    phi <- bE[G]*sum( delta_j[1:E] ) +
      bA[G]*A + bI[G]*I + bC[G]*C +
      bY[G]*Y,
    alpha ~ normal( 0 , 1 ),
    bA[G] ~ normal( 0 , 0.5 ),
    bI[G] ~ normal( 0 , 0.5 ),
    bC[G] ~ normal( 0 , 0.5 ),
    bE[G] ~ normal( 0 , 0.5 ),
    bY[G] ~ normal( 0 , 0.5 ),
    vector[8]: delta_j <- append_row( 0 , delta ),
    simplex[7]: delta ~ dirichlet( a )
  ), data=dat , chains=4 , cores=4 , threads=2 )

```

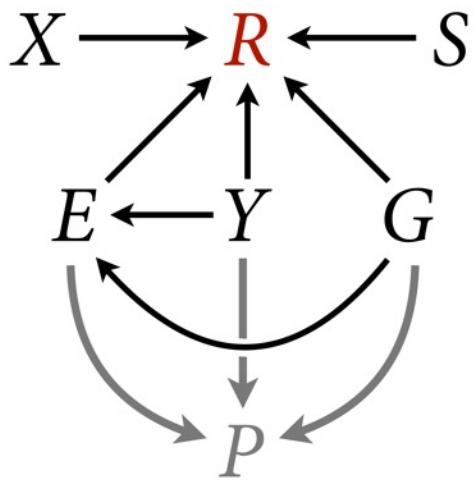
*4 chains times 2 threads each = 8 cores*

*1 thread each*

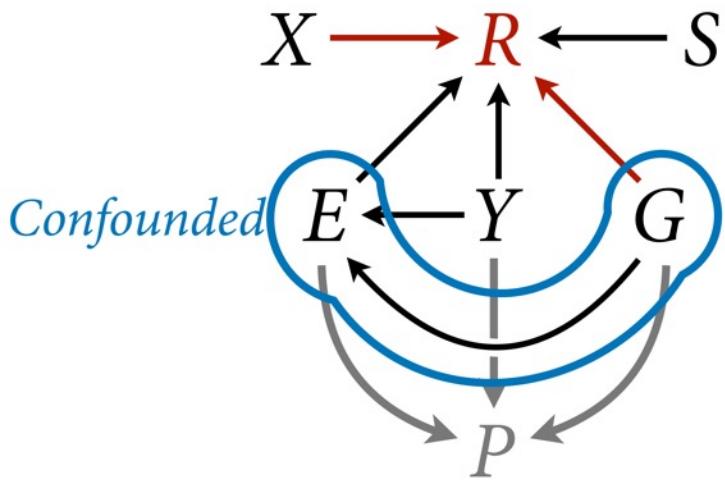
Sampling durations (minutes):			
	warmup	sample	total
chain:1	6.53	3.99	10.52
chain:2	7.33	2.66	9.99
chain:3	6.88	3.70	10.58
chain:4	6.40	2.63	9.03

*2 threads each*

Sampling durations (minutes):			
	warmup	sample	total
chain:1	4.41	1.80	6.21
chain:2	4.69	1.87	6.56
chain:3	5.14	1.56	6.70
chain:4	4.21	1.84	6.05

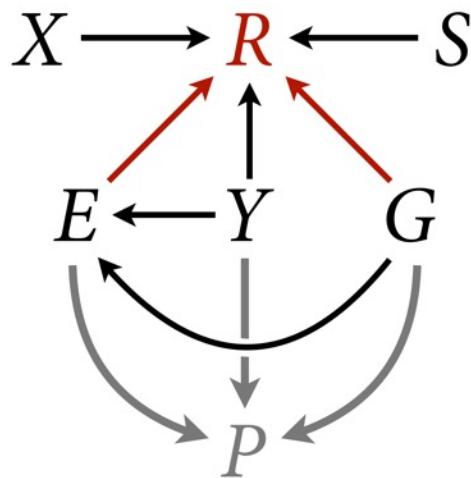


	mean	sd	5.5%	94.5%	n_eff	Rhat4
alpha[1]	-2.89	0.10	-3.06	-2.73	729	1
alpha[2]	-2.21	0.10	-2.37	-2.06	728	1
alpha[3]	-1.62	0.10	-1.78	-1.47	724	1
alpha[4]	-0.58	0.10	-0.74	-0.43	729	1
alpha[5]	0.11	0.10	-0.05	0.26	726	1
alpha[6]	1.03	0.10	0.87	1.18	746	1
bA[1]	-0.56	0.06	-0.65	-0.47	1932	1
bA[2]	-0.81	0.05	-0.90	-0.73	2013	1
bI[1]	-0.66	0.05	-0.74	-0.58	2539	1
bI[2]	-0.76	0.05	-0.84	-0.68	2283	1
bC[1]	-0.77	0.07	-0.88	-0.65	2029	1
bC[2]	-1.09	0.07	-1.20	-0.99	2012	1
bE[1]	-0.63	0.14	-0.85	-0.42	810	1
bE[2]	0.41	0.14	0.19	0.62	795	1
bY[1]	0.00	0.03	-0.05	0.05	2740	1
bY[2]	-0.13	0.03	-0.18	-0.09	1426	1
delta[1]	0.15	0.08	0.04	0.31	1759	1
delta[2]	0.15	0.09	0.04	0.30	2440	1
delta[3]	0.29	0.11	0.11	0.46	2001	1
delta[4]	0.08	0.05	0.02	0.17	2414	1
delta[5]	0.06	0.04	0.01	0.14	1087	1
delta[6]	0.24	0.07	0.13	0.34	2301	1
delta[7]	0.04	0.02	0.01	0.08	2755	1

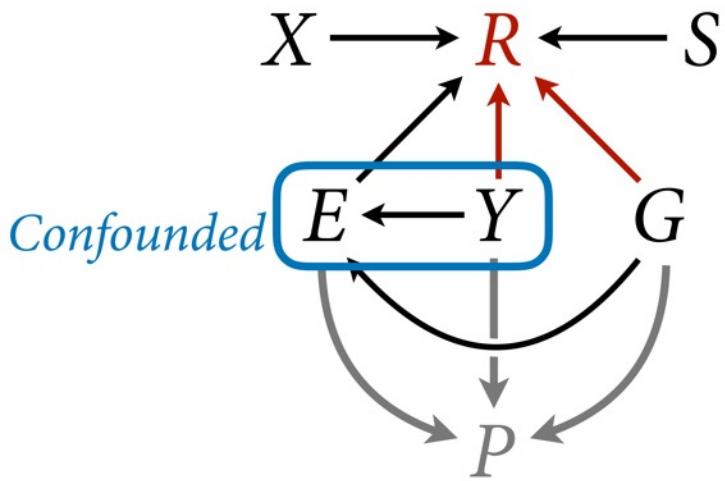


*Only  
direct  
effect G*

```
> precis(mRXYGt,2)
      mean   sd  5.5% 94.5% n_eff Rhat4
alpha[1] -2.89 0.10 -3.06 -2.73    729     1
alpha[2] -2.21 0.10 -2.37 -2.06    728     1
alpha[3] -1.62 0.10 -1.78 -1.47    724     1
alpha[4] -0.58 0.10 -0.74 -0.43    729     1
alpha[5]  0.11 0.10 -0.05  0.26    726     1
alpha[6]  1.03 0.10  0.87  1.18    746     1
bA[1]   -0.56 0.06 -0.65 -0.47   1932     1
bA[2]   -0.81 0.05 -0.90 -0.73  2013     1
bI[1]   -0.66 0.05 -0.74 -0.58  2539     1
bI[2]   -0.76 0.05 -0.84 -0.68  2283     1
bC[1]   -0.77 0.07 -0.88 -0.65  2029     1
bC[2]   -1.09 0.07 -1.20 -0.99  2012     1
bE[1]   -0.63 0.14 -0.85 -0.42   810     1
bE[2]    0.41 0.14  0.19  0.62   795     1
bY[1]    0.00 0.03 -0.05  0.05  2740     1
bY[2]   -0.13 0.03 -0.18 -0.09  1426     1
delta[1]  0.15 0.08  0.04  0.31  1759     1
delta[2]  0.15 0.09  0.04  0.30  2440     1
delta[3]  0.29 0.11  0.11  0.46  2001     1
delta[4]  0.08 0.05  0.02  0.17  2414     1
delta[5]  0.06 0.04  0.01  0.14  1087     1
delta[6]  0.24 0.07  0.13  0.34  2301     1
delta[7]  0.04 0.02  0.01  0.08  2755     1
```

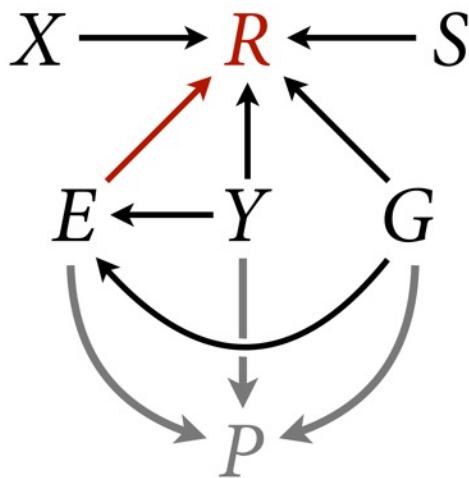


	mean	sd	5.5%	94.5%	n_eff	Rhat4
alpha[1]	-2.89	0.10	-3.06	-2.73	729	1
alpha[2]	-2.21	0.10	-2.37	-2.06	728	1
alpha[3]	-1.62	0.10	-1.78	-1.47	724	1
alpha[4]	-0.58	0.10	-0.74	-0.43	729	1
alpha[5]	0.11	0.10	-0.05	0.26	726	1
alpha[6]	1.03	0.10	0.87	1.18	746	1
bA[1]	-0.56	0.06	-0.65	-0.47	1932	1
bA[2]	-0.81	0.05	-0.90	-0.73	2013	1
bI[1]	-0.66	0.05	-0.74	-0.58	2539	1
bI[2]	-0.76	0.05	-0.84	-0.68	2283	1
bC[1]	-0.77	0.07	-0.88	-0.65	2029	1
bC[2]	-1.09	0.07	-1.20	-0.99	2012	1
bE[1]	-0.63	0.14	-0.85	-0.42	810	1
bE[2]	0.41	0.14	0.19	0.62	795	1
bY[1]	0.00	0.03	-0.05	0.05	2740	1
bY[2]	-0.13	0.03	-0.18	-0.09	1426	1
delta[1]	0.15	0.08	0.04	0.31	1759	1
delta[2]	0.15	0.09	0.04	0.30	2440	1
delta[3]	0.29	0.11	0.11	0.46	2001	1
delta[4]	0.08	0.05	0.02	0.17	2414	1
delta[5]	0.06	0.04	0.01	0.14	1087	1
delta[6]	0.24	0.07	0.13	0.34	2301	1
delta[7]	0.04	0.02	0.01	0.08	2755	1



*Only  
direct  
effect*

	mean	sd	5.5%	94.5%	n_eff	Rhat4
alpha[1]	-2.89	0.10	-3.06	-2.73	729	1
alpha[2]	-2.21	0.10	-2.37	-2.06	728	1
alpha[3]	-1.62	0.10	-1.78	-1.47	724	1
alpha[4]	-0.58	0.10	-0.74	-0.43	729	1
alpha[5]	0.11	0.10	-0.05	0.26	726	1
alpha[6]	1.03	0.10	0.87	1.18	746	1
bA[1]	-0.56	0.06	-0.65	-0.47	1932	1
bA[2]	-0.81	0.05	-0.90	-0.73	2013	1
bI[1]	-0.66	0.05	-0.74	-0.58	2539	1
bI[2]	-0.76	0.05	-0.84	-0.68	2283	1
bC[1]	-0.77	0.07	-0.88	-0.65	2029	1
bC[2]	-1.09	0.07	-1.20	-0.99	2012	1
bE[1]	-0.63	0.14	-0.85	-0.42	810	1
bE[2]	0.41	0.14	0.19	0.62	795	1
bY[1]	0.00	0.03	-0.05	0.05	2740	1
bY[2]	-0.13	0.03	-0.18	-0.09	1426	1
delta[1]	0.15	0.08	0.04	0.31	1759	1
delta[2]	0.15	0.09	0.04	0.30	2440	1
delta[3]	0.29	0.11	0.11	0.46	2001	1
delta[4]	0.08	0.05	0.02	0.17	2414	1
delta[5]	0.06	0.04	0.01	0.14	1087	1
delta[6]	0.24	0.07	0.13	0.34	2301	1
delta[7]	0.04	0.02	0.01	0.08	2755	1



*Model mRXEYG2t stratifies by G, in Lecture 11 script*

	mean	sd	5.5%	94.5%	n_eff	Rhat4
alpha[1]	-2.89	0.10	-3.06	-2.73	729	1
alpha[2]	-2.21	0.10	-2.37	-2.06	728	1
alpha[3]	-1.62	0.10	-1.78	-1.47	724	1
alpha[4]	-0.58	0.10	-0.74	-0.43	729	1
alpha[5]	0.11	0.10	-0.05	0.26	726	1
alpha[6]	1.03	0.10	0.87	1.18	746	1
bA[1]	-0.56	0.06	-0.65	-0.47	1932	1
bA[2]	-0.81	0.05	-0.90	-0.73	2013	1
bI[1]	-0.66	0.05	-0.74	-0.58	2539	1
bI[2]	-0.76	0.05	-0.84	-0.68	2283	1
bC[1]	-0.77	0.07	-0.88	-0.65	2029	1
bC[2]	-1.09	0.07	-1.20	-0.99	2012	1
bE[1]	-0.63	0.14	-0.85	-0.42	810	1
bE[2]	0.41	0.14	0.19	0.62	795	1
bY[1]	0.00	0.03	-0.05	0.05	2740	1
bY[2]	-0.13	0.03	-0.18	-0.09	1426	1
delta[1]	0.15	0.08	0.04	0.31	1759	1
delta[2]	0.15	0.09	0.04	0.30	2440	1
delta[3]	0.29	0.11	0.11	0.46	2001	1
delta[4]	0.08	0.05	0.02	0.17	2414	1
delta[5]	0.06	0.04	0.01	0.14	1087	1
delta[6]	0.24	0.07	0.13	0.34	2301	1
delta[7]	0.04	0.02	0.01	0.08	2755	1

# Complex causal effects

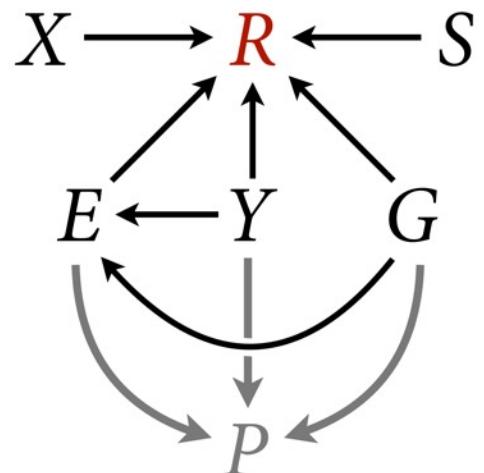
Causal effects (predicted consequences of intervention) require **marginalization**

Example: Causal effect of  $E$  requires distribution of  $Y$  and  $G$  to average over

Problem 1: Should not marginalize over **this** sample—**cursed  $P$ !** Post-stratify to new target.

Problem 2: Should not set all  $Y$  to same  $E$

Example: Causal effect of  $Y$  requires effect of  $Y$  on  $E$ , which we cannot estimate ( $P$  again!)



# Complex causal effects

Causal effects (predicted consequences of intervention)

Example of  $Y$  and

Problem sample

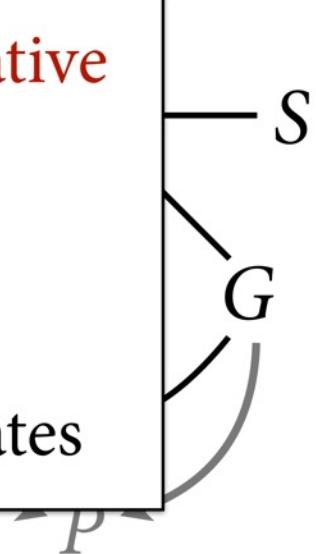
Problem

No matter how complex, still just a **generative simulation using posterior samples**

Need generative model to plan estimation

Need generative model to compute estimates

Example: Causal effect of  $Y$  requires effect of  $Y$  on  $E$ , which we cannot estimate ( $P$  again!)

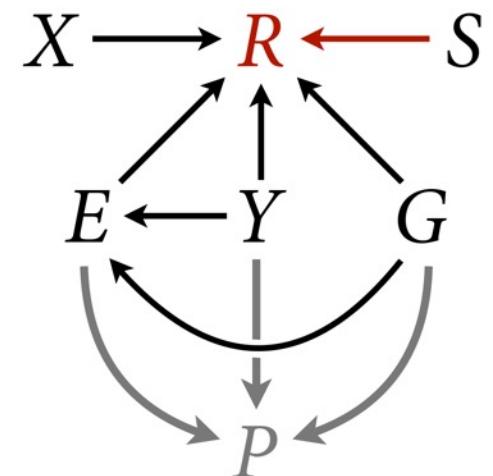


# Repeat observations

30 stories ( $S$ )

```
> table(d$story)
```

aqu	boa	box	bur	car	che	pon	rub	sha	shi	spe	swi
662	662	1324	1324	662	662	662	662	662	662	993	993



# Repeat observations

30 stories ( $S$ )

```
> table(d$story)
```

aqu	boa	box	bur	car	che	pon	rub	sha	shi	spe	swi
662	662	1324	1324	662	662	662	662	662	662	993	993

331 individuals ( $U$ )

```
> table(d$id)
```

96;434	96;445	96;451	96;456	96;458	96;466	96;467	96;474	96;480	96;481	96;497
30	30	30	30	30	30	30	30	30	30	30
96;498	96;502	96;505	96;511	96;512	96;518	96;519	96;531	96;533	96;538	96;547
30	30	30	30	30	30	30	30	30	30	30
96;550	96;553	96;555	96;558	96;560	96;562	96;566	96;570	96;581	96;586	96;591
30	30	30	30	30	30	30	30	30	30	30

