Statistical Rethinking
2023

19. Generalized Linear Madness


GLM
Generalized Linear Models

GLMM
Generalized Linear
Mixed Models

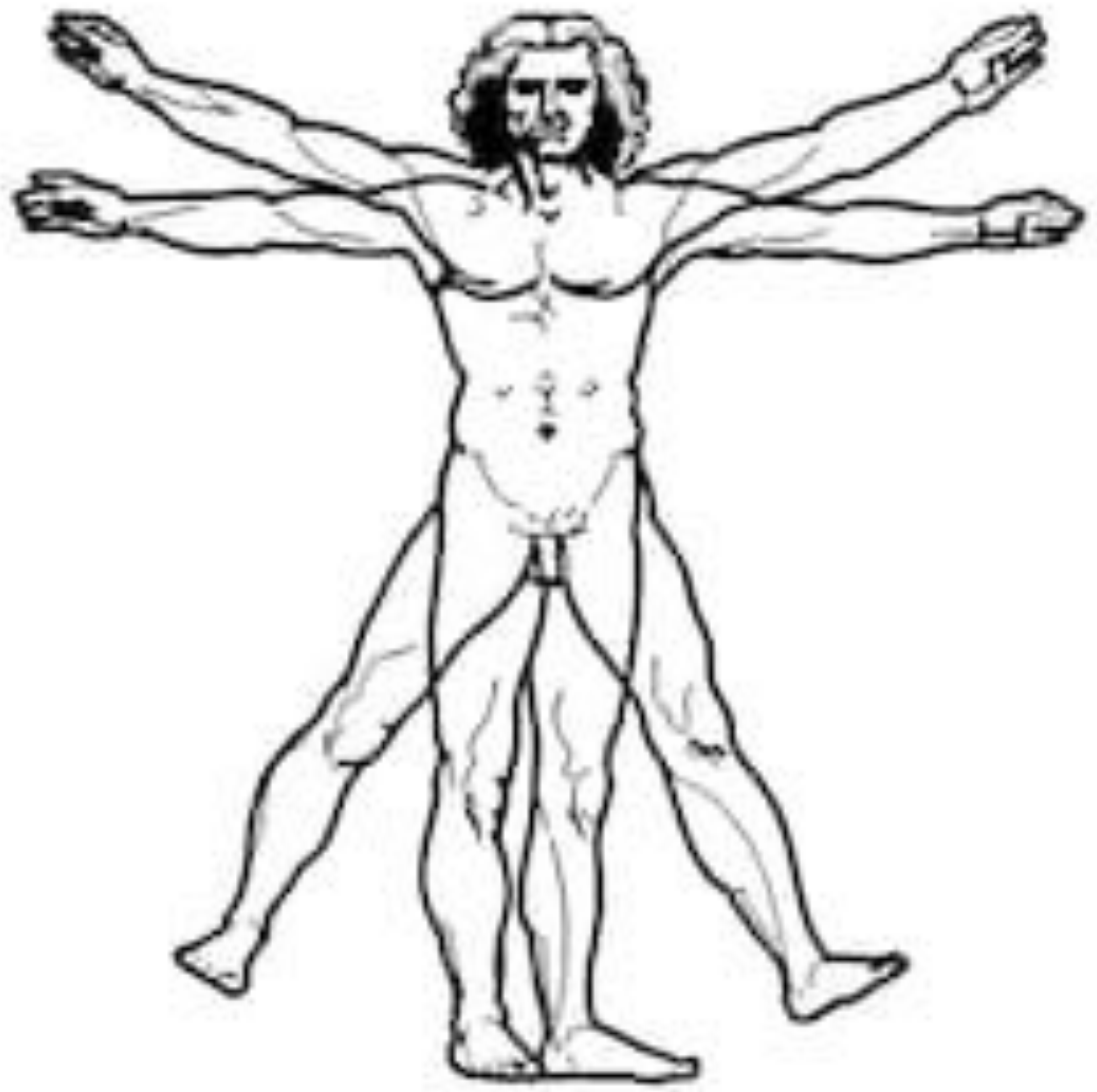
Generalized Linear Habits

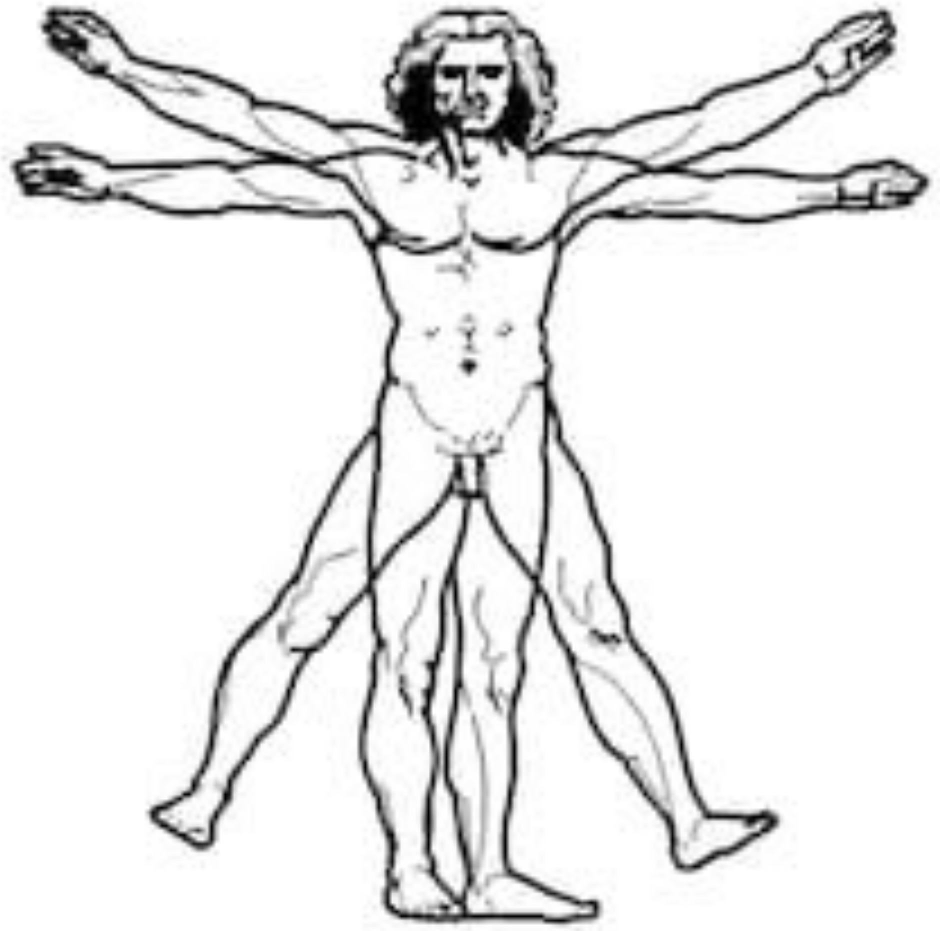
GLMs and GLMMs: Flexible association description machines

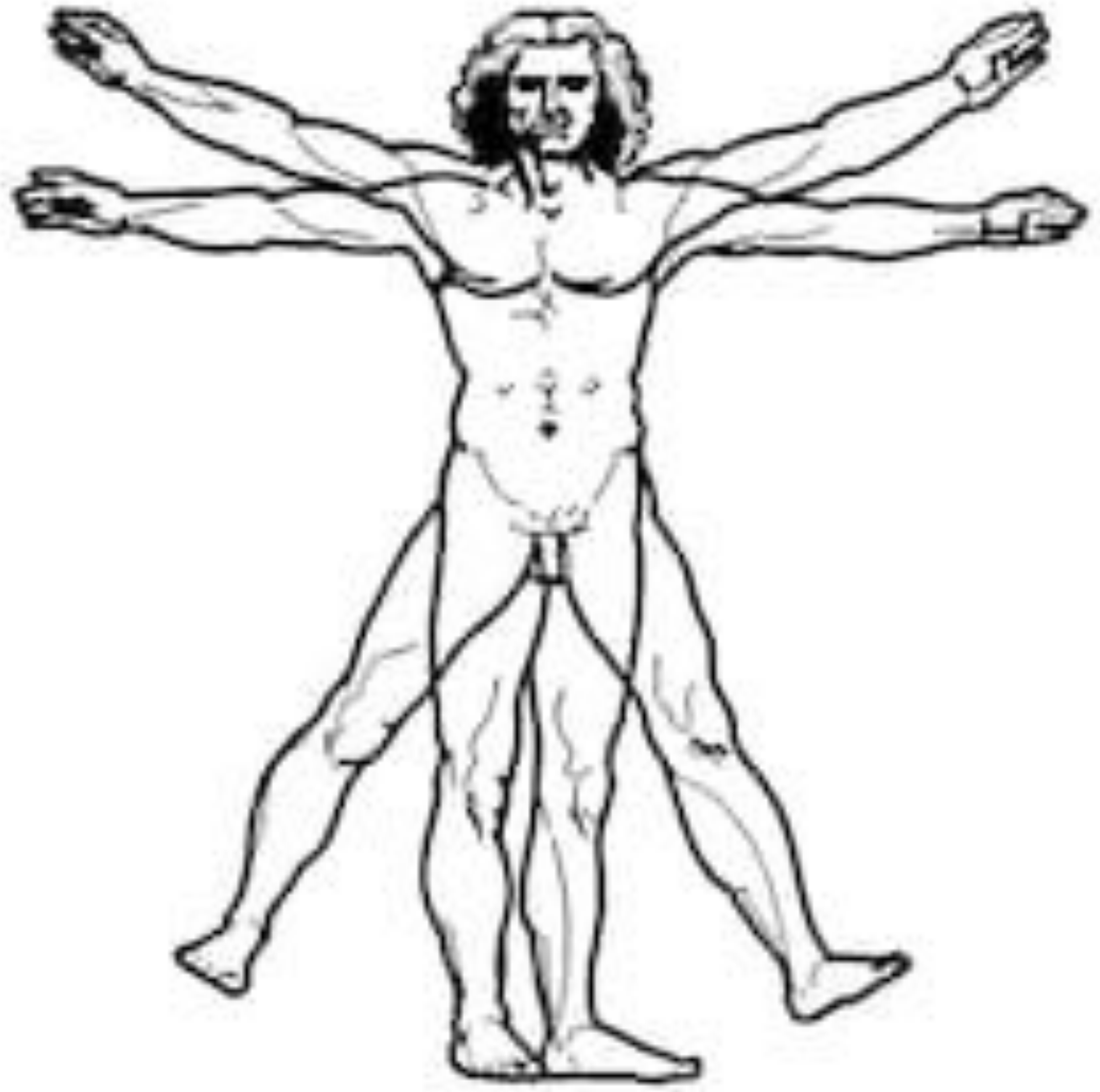
With external causal model, causal interpretation possible

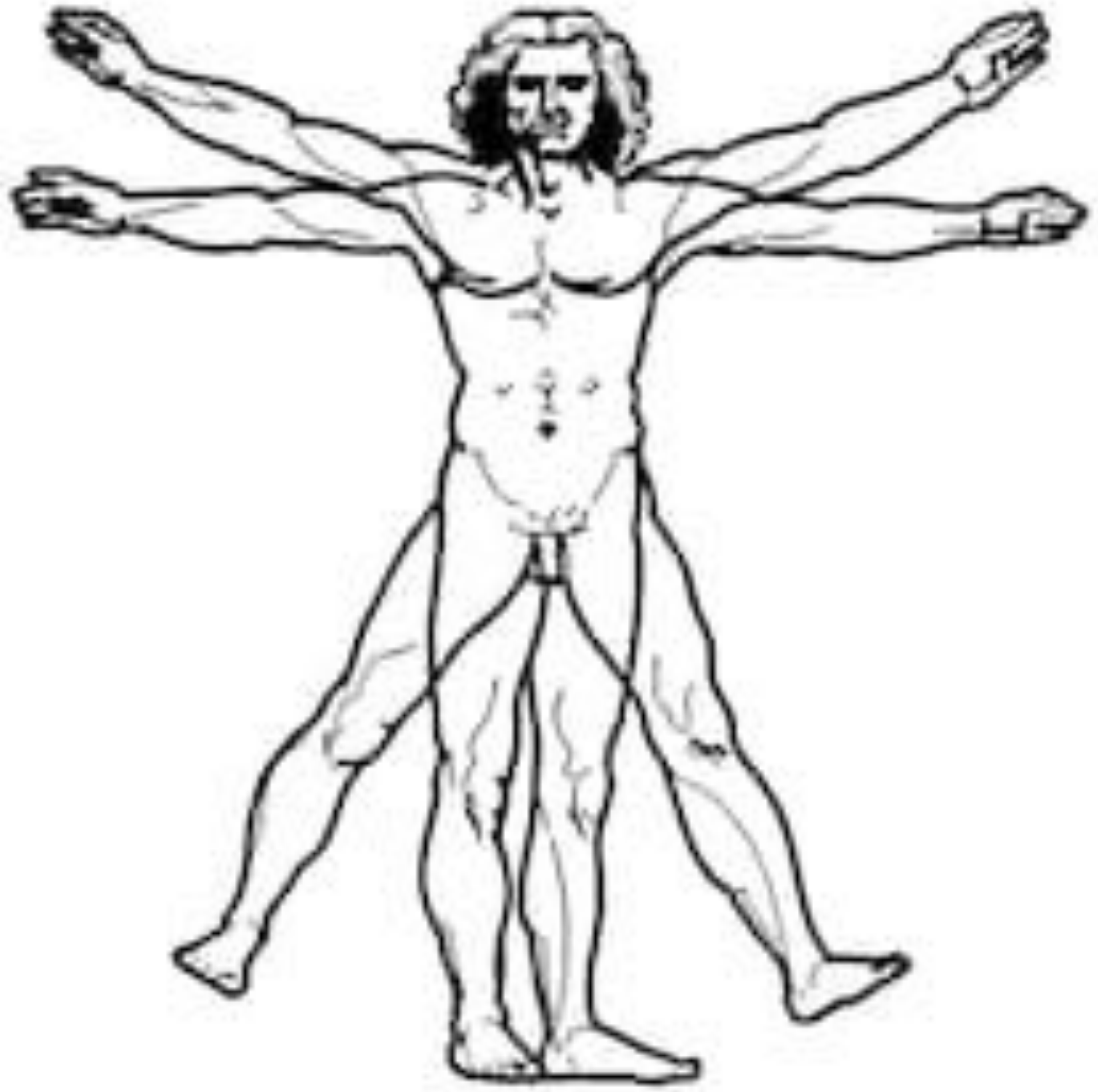
But only a fraction of scientific phenomena expressible as GLM(M)s

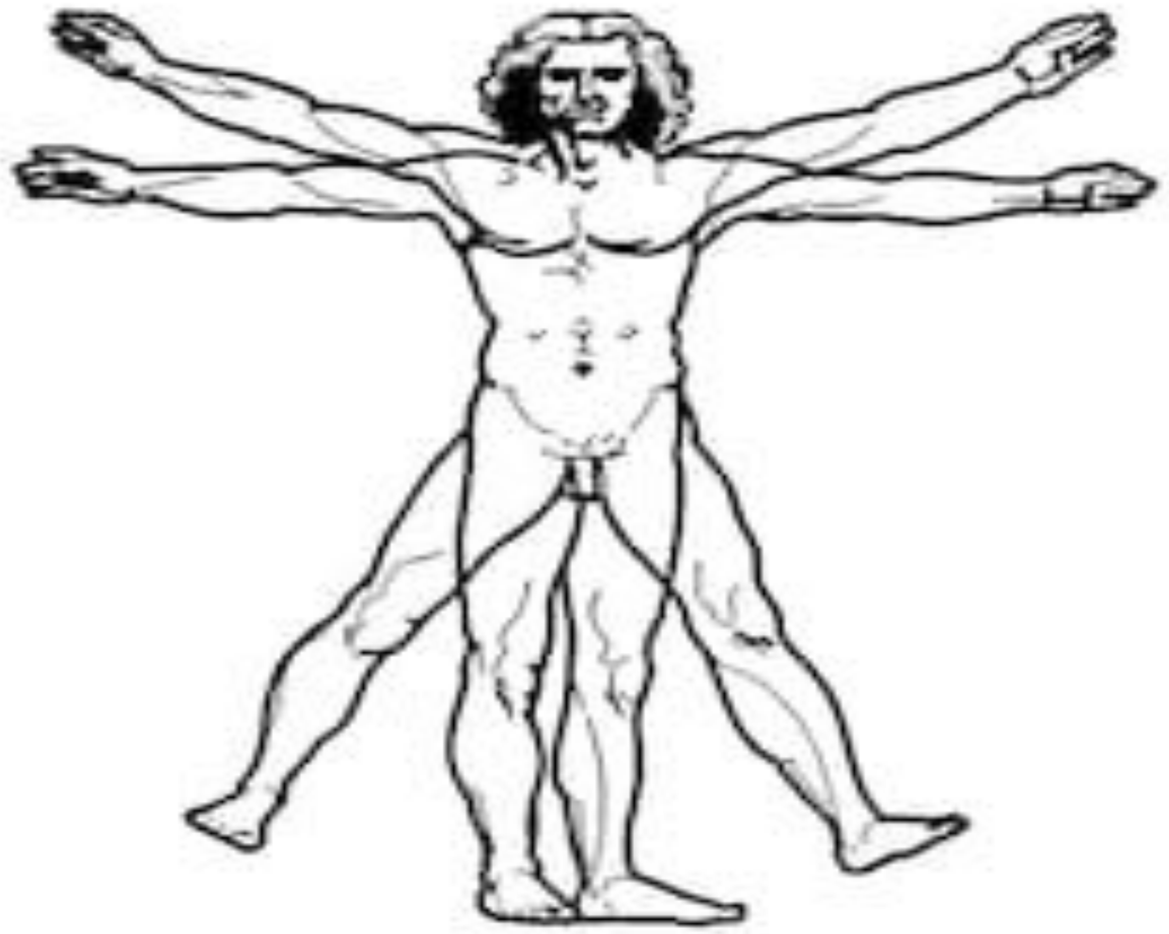
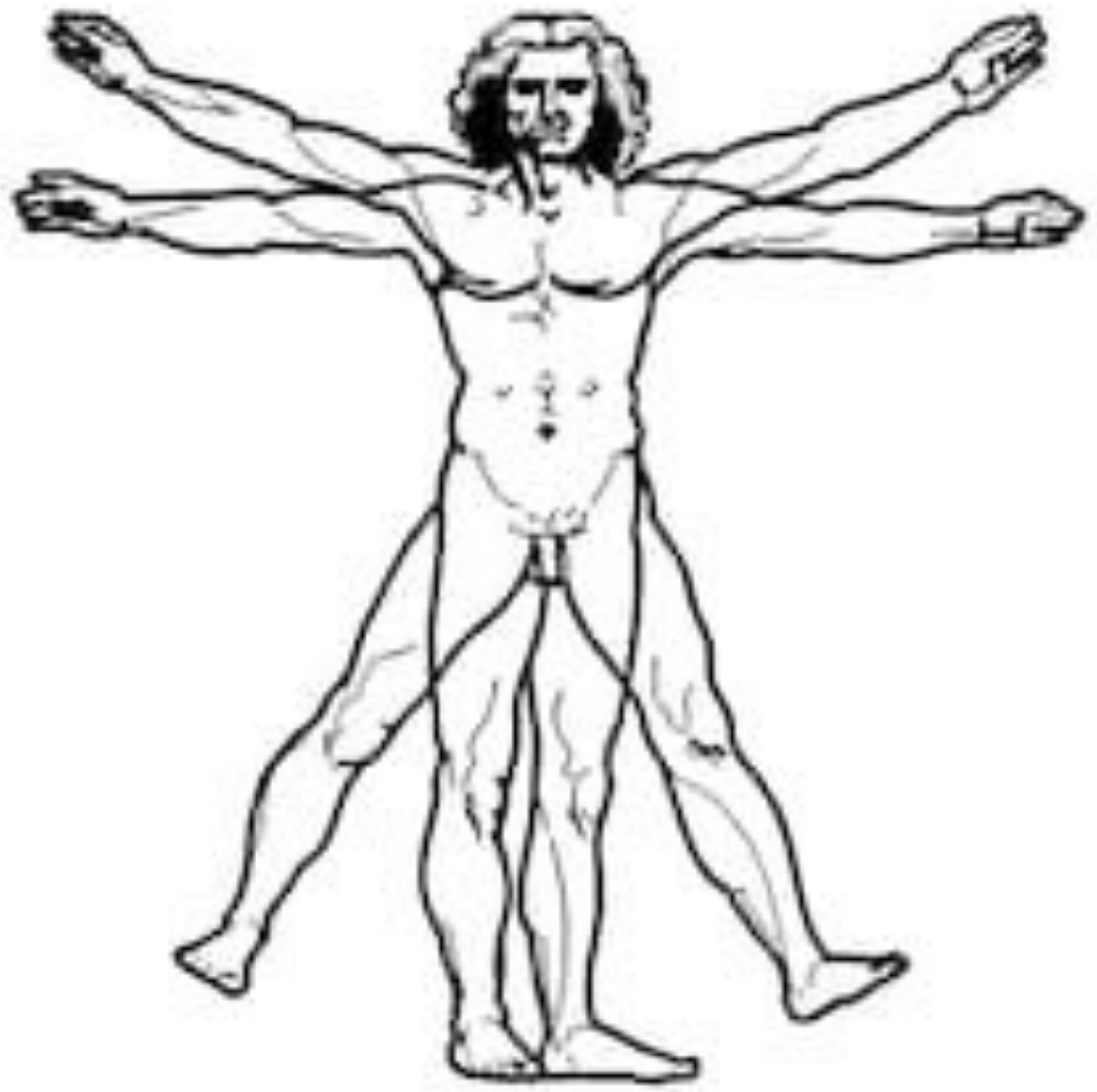
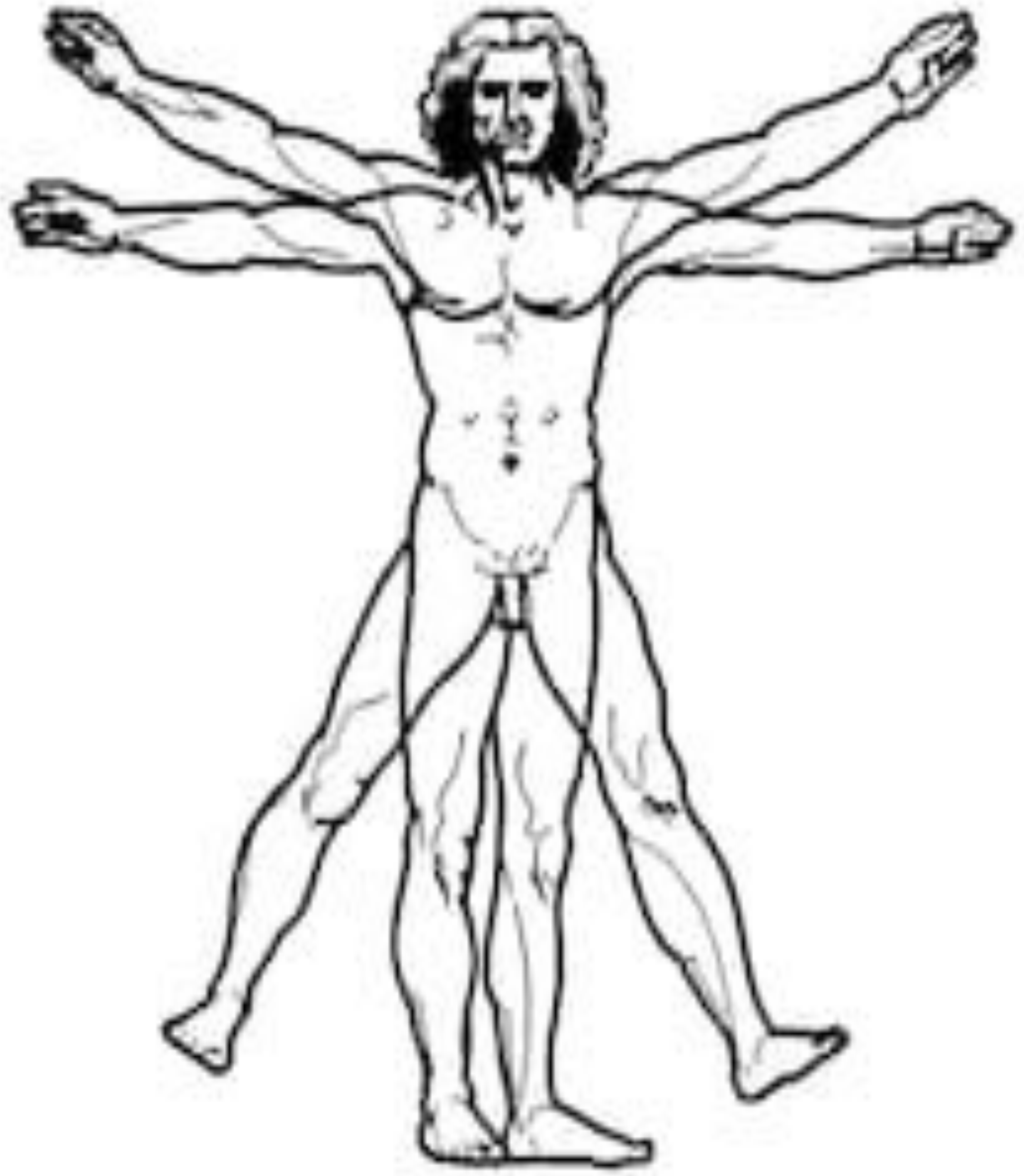
Even when GLM(M)s sufficient, starting with theory solves empirical problems

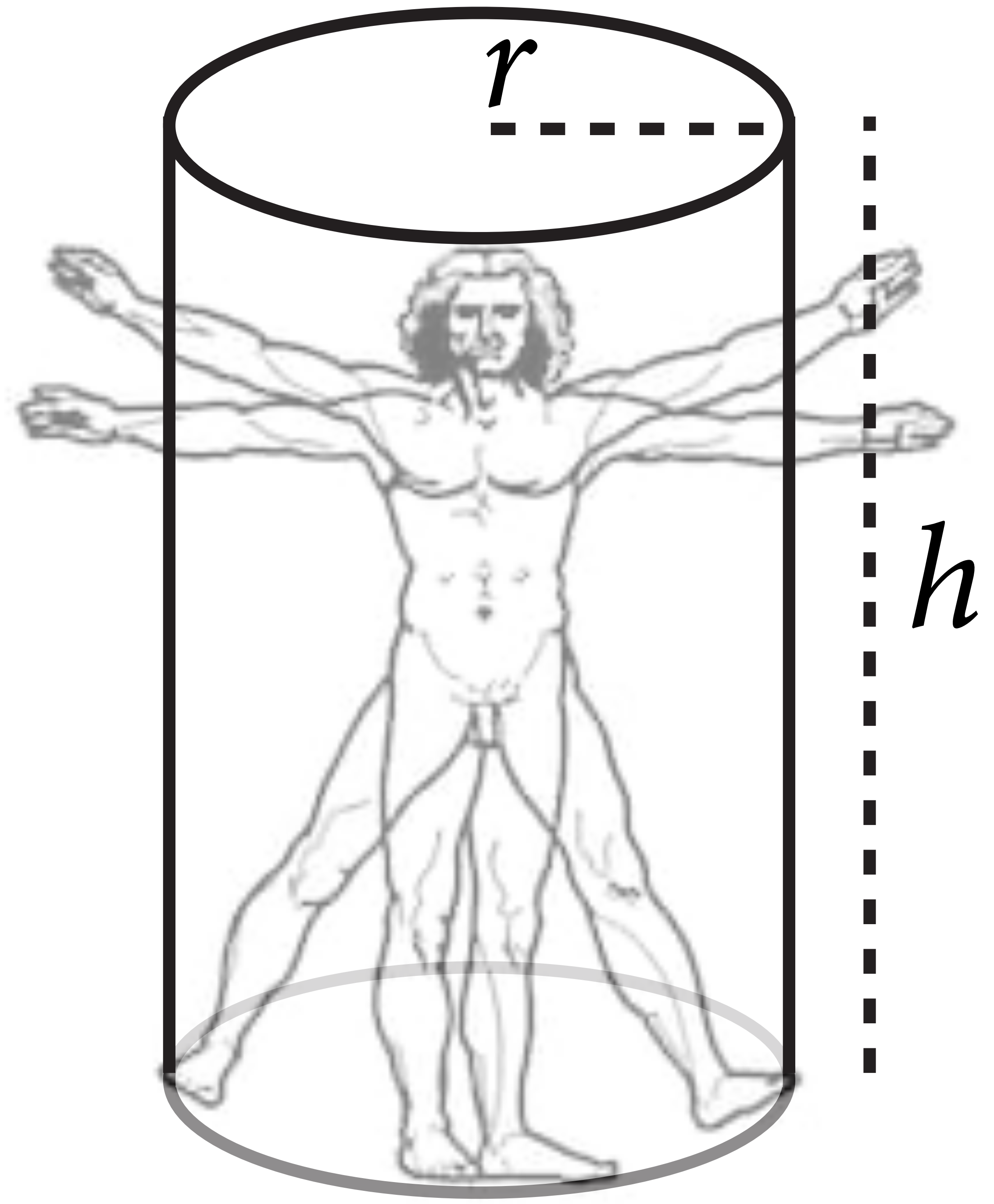








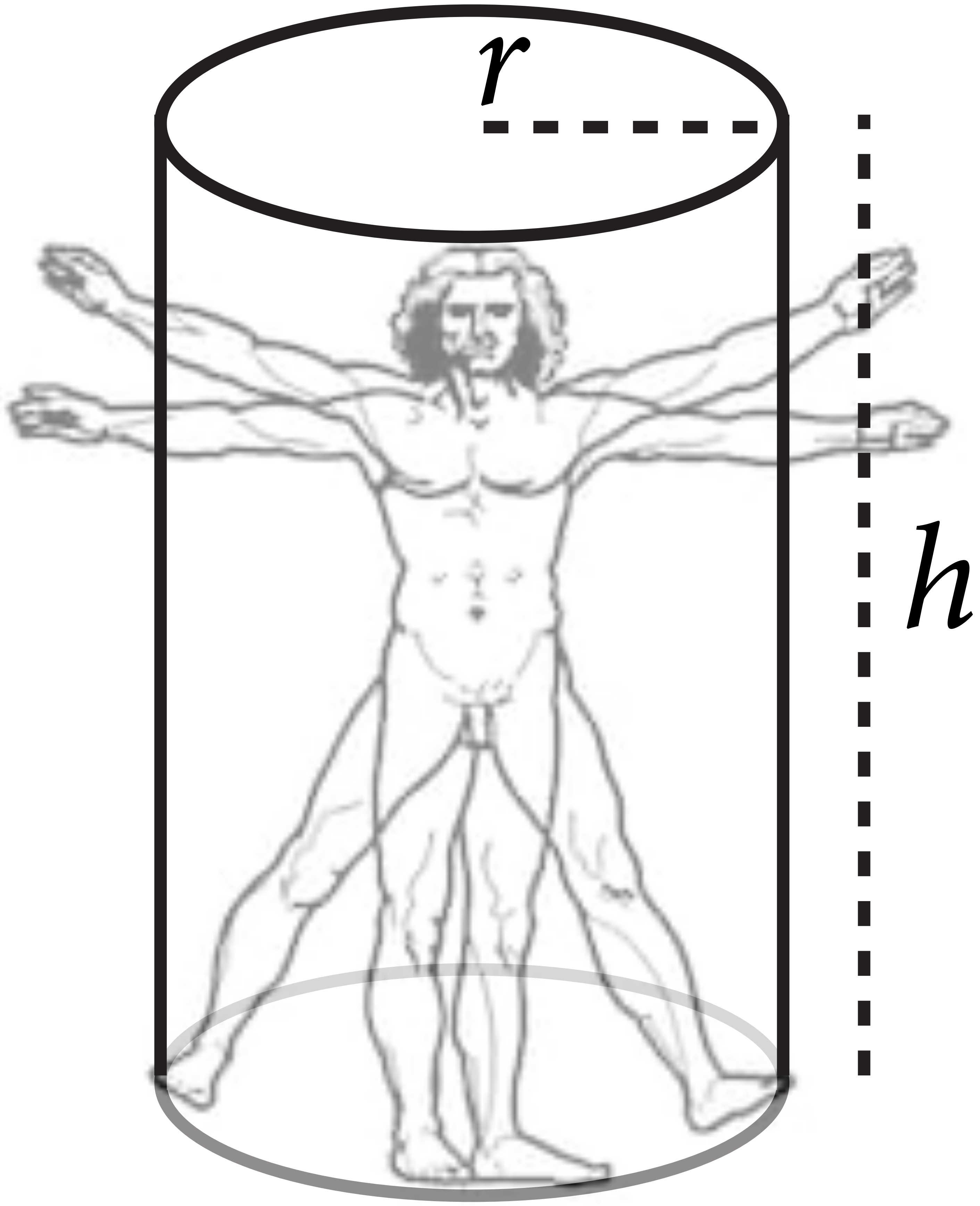

r
h

$$V = \pi r^2 h$$

volume

radius

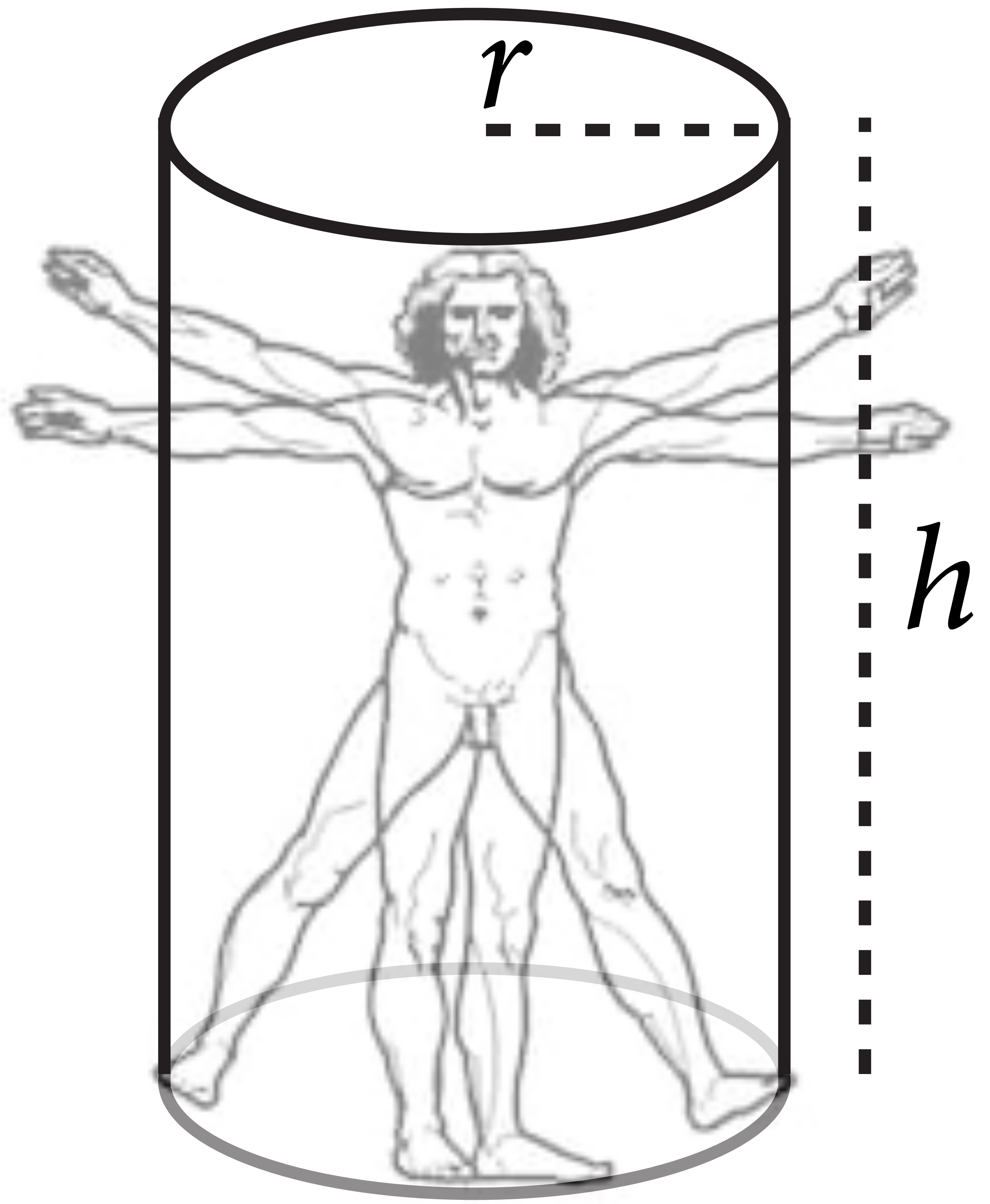
height



$$V = \pi r^2 h$$

$$V = \pi(ph)^2 h$$

radius as proportion of height

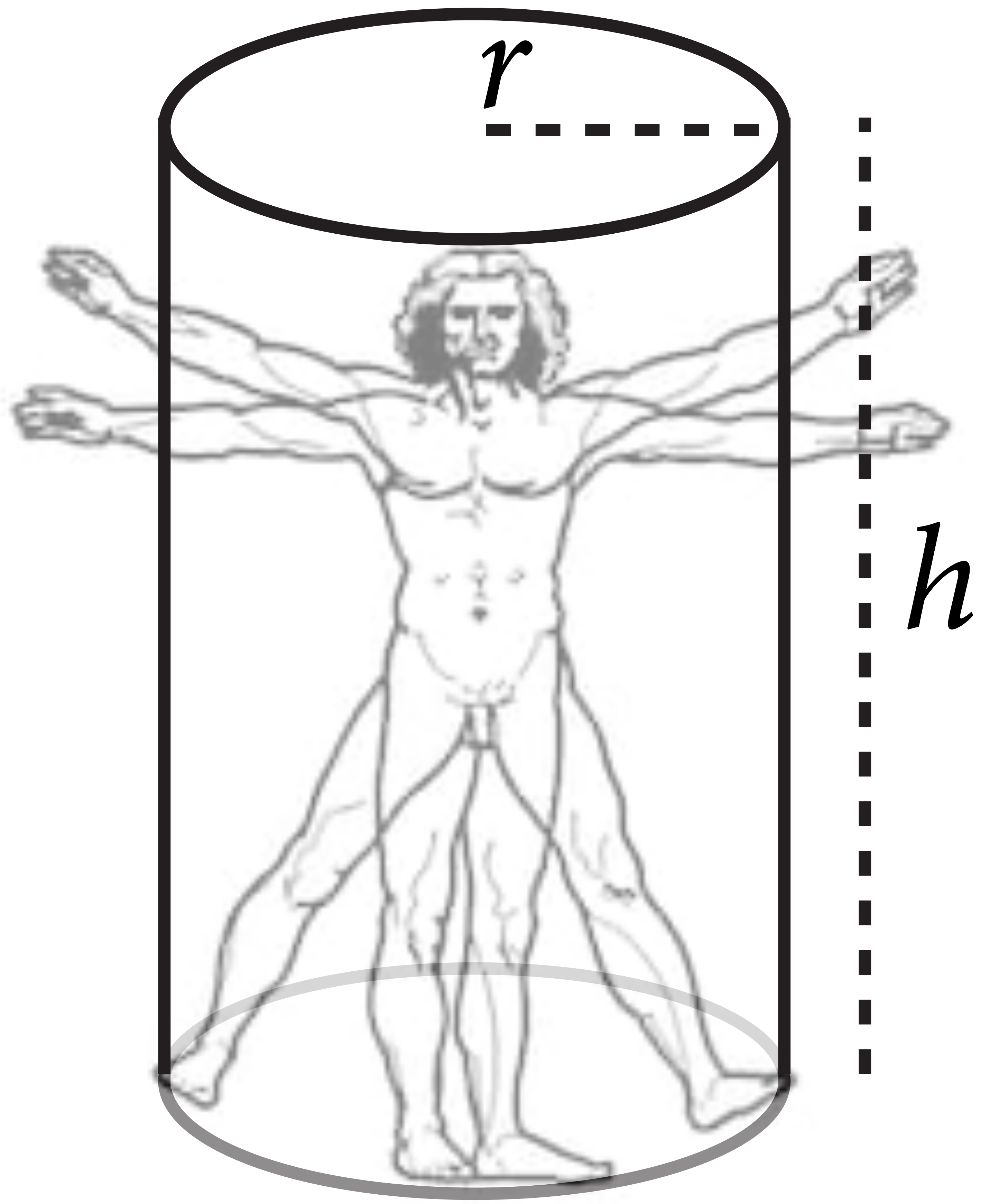


$$V = \pi r^2 h$$

$$V = \pi (ph)^2 h$$

$$W = kV = k\pi (ph)^2 h$$

weight “*density*”

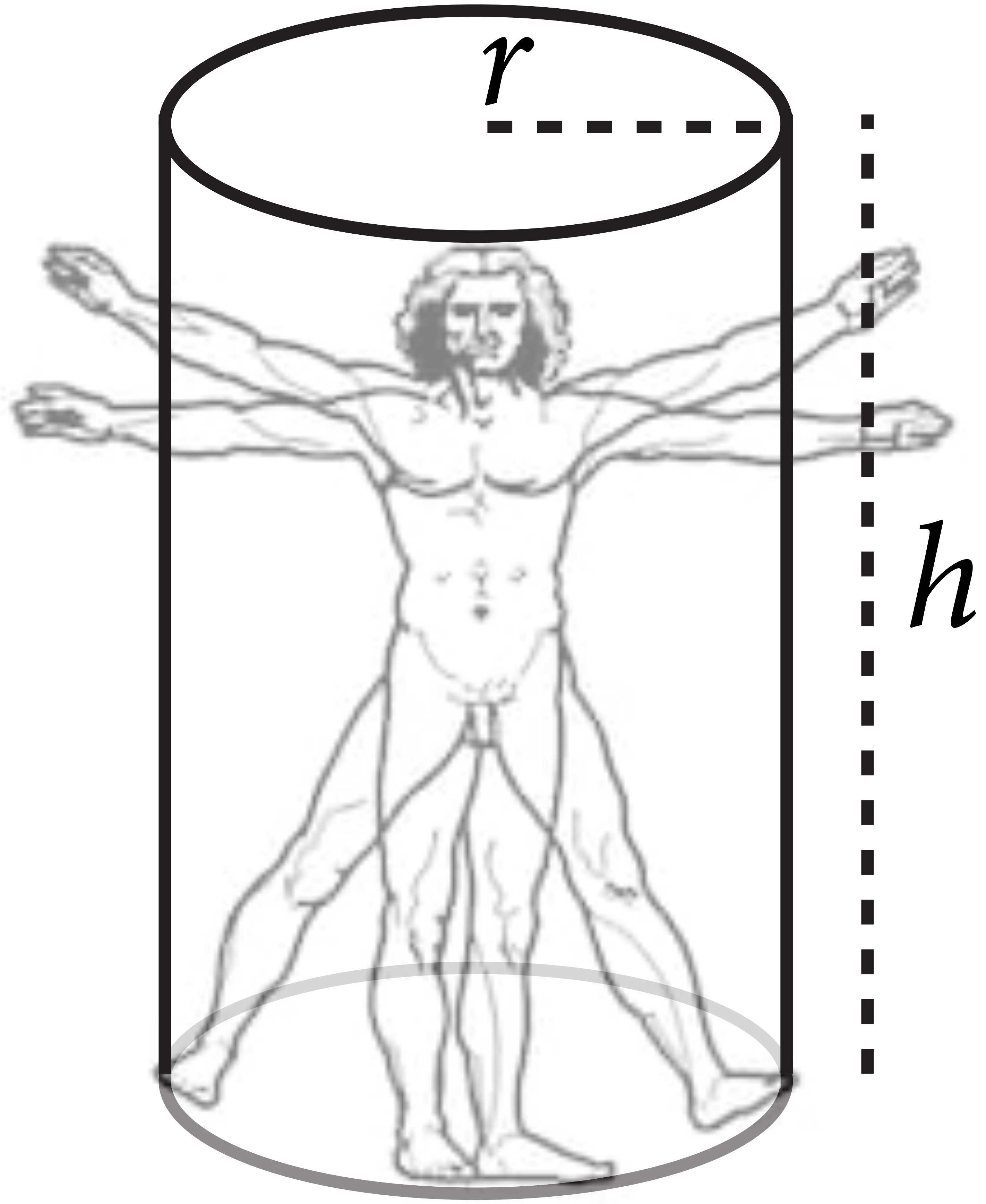


$$V = \pi r^2 h$$

$$V = \pi (ph)^2 h$$

$$W = kV = k\pi (ph)^2 h$$

$$W = k\pi p^2 h^3$$



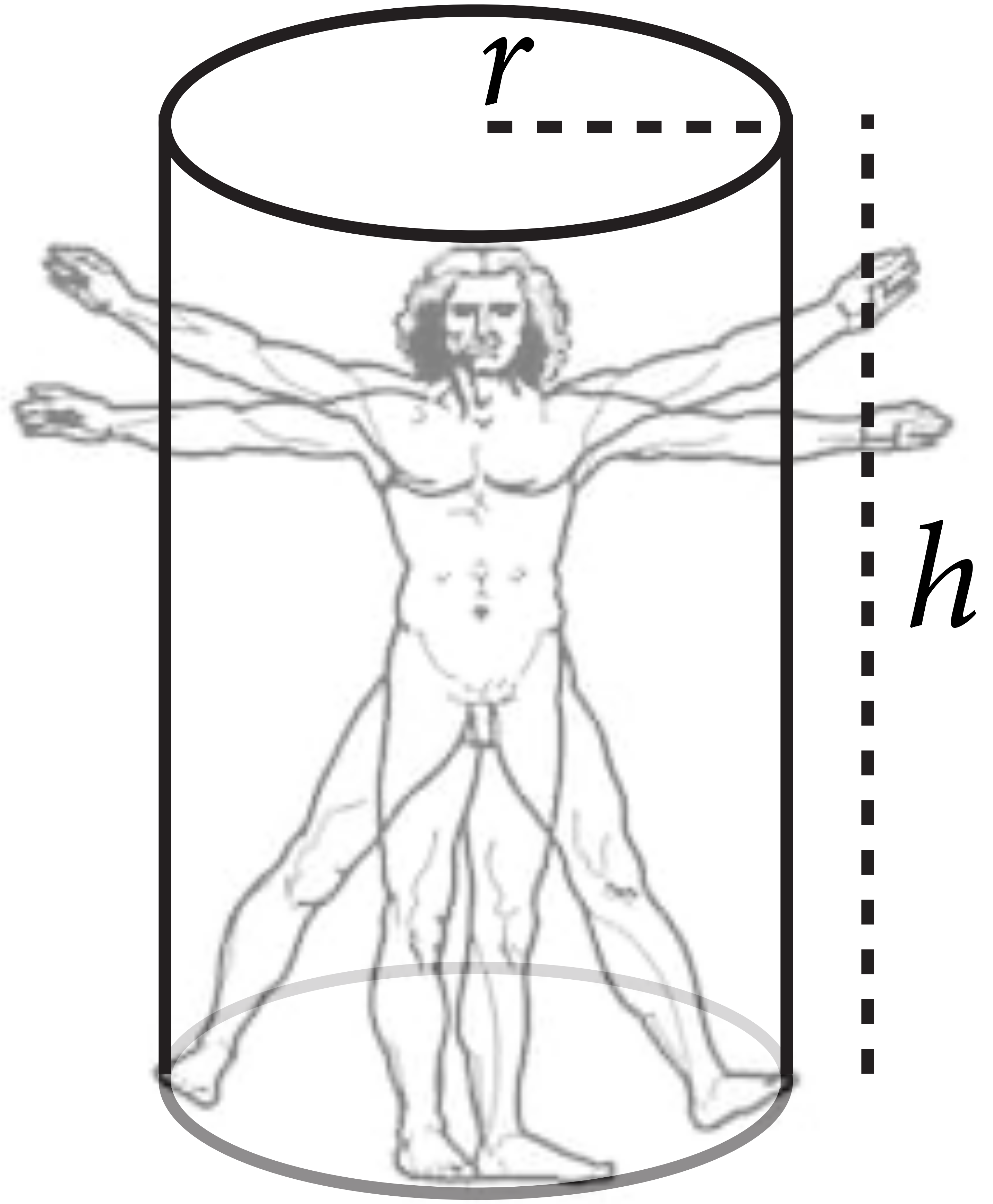
weight (data)

height (data)

$$W = k\pi p^2 h^3$$

density

proportionality



$$W_i \sim \text{Distribution}(\mu_i, \ldots)$$ *"error" distribution for W*

$$\mu_i = k\pi p^2 H_i^3$$ *expected W for H*

$$p \sim \text{Distribution}(\ldots)$$ *prior for proportionality*

$$k \sim \text{Distribution}(\ldots)$$ *prior for density*

How to set these priors?
(1) Choose measurement scales
(2) Simulate
(3) Think

$$p \sim \text{Distribution}(\ldots)$$ *prior for proportionality*

$$k \sim \text{Distribution}(\ldots)$$ *prior for density*

How to set these priors?

(1) Choose measurement scales

(2) Simulate

(3) Think

unitless ratios

$$\mu_i = k\pi p^2 H_i^3$$

kg *kg/cm³* *cm³*

How to set these priors?

(1) Choose measurement scales

(2) Simulate

(3) Think

$$\mu_i = k \pi p^2 H_i^3$$

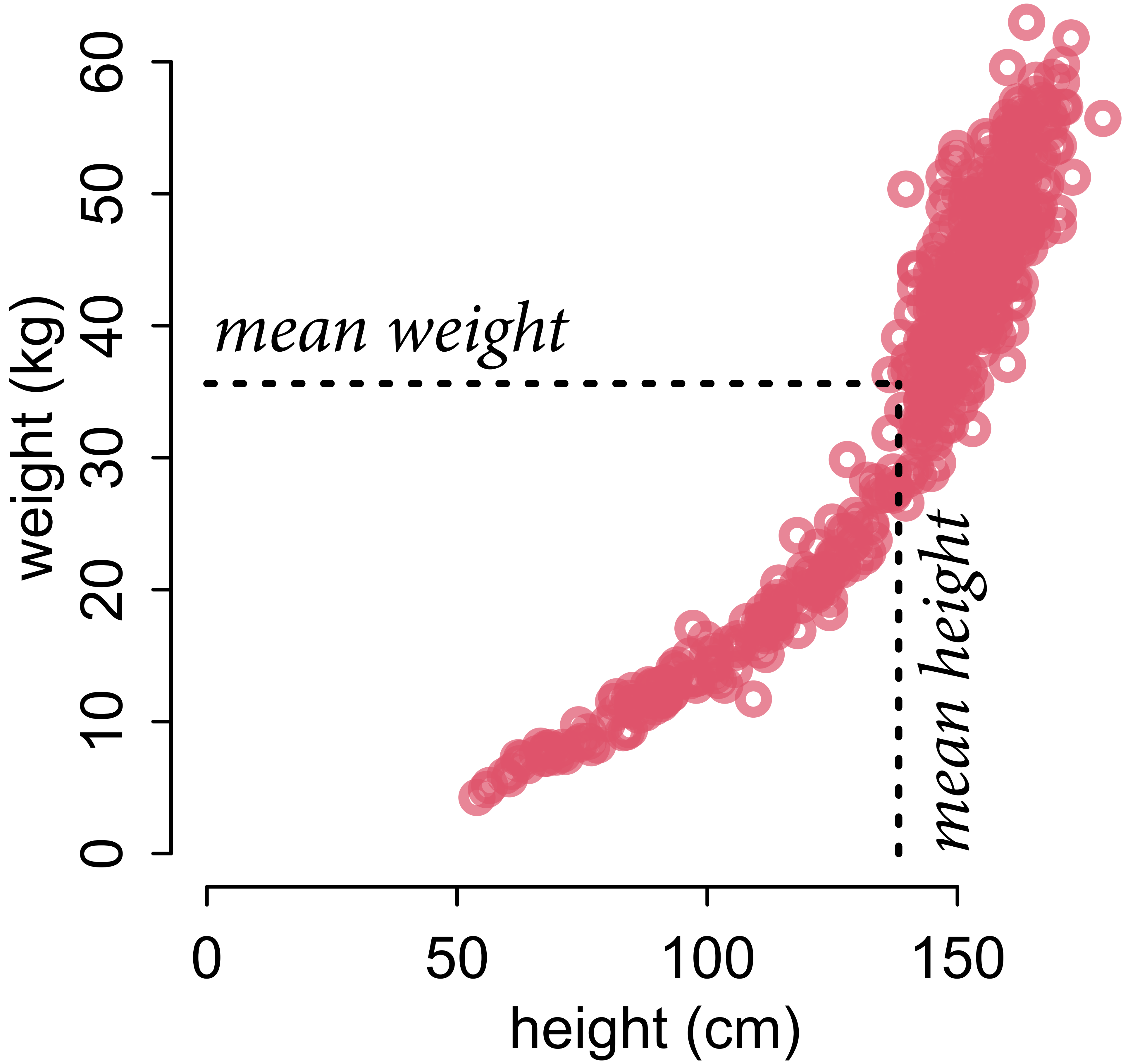
$$kg \quad = \quad kg/cm^3 \times cm^3$$

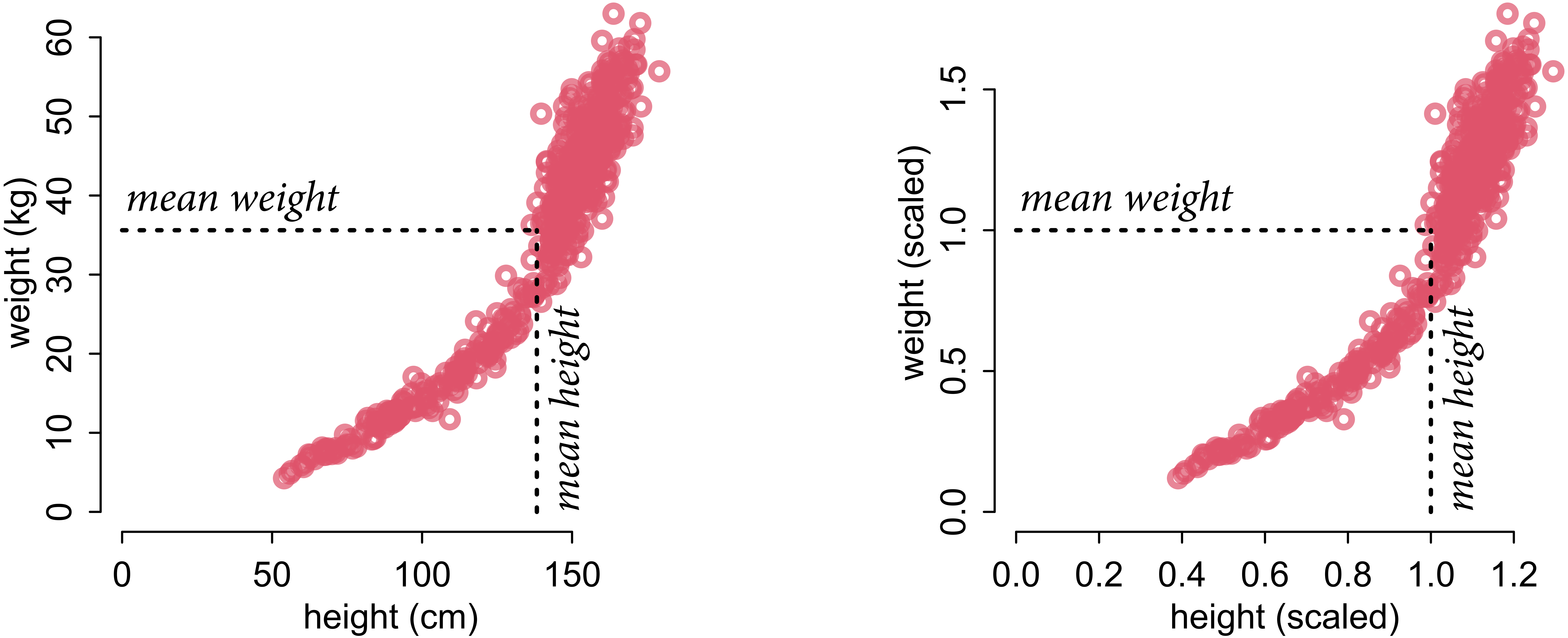
$$\mu_i = k\pi p^2 H_i^3$$

kg = kg/cm³ × cm³

Measurement scales are artifice

If you can divide out all measurement units (kg, cm), often easier





mean weight
mean height
weight (kg)
height (cm)
0
10
20
30
40
50
60
0
50
100
150
mean weight
mean height
weight (scaled)
height (scaled)
0.0
0.5
1.0
1.5
0.0
0.2
0.4
0.6
0.8
1.0
1.2

How to set these priors?
(1) Choose measurement scales
(2) Simulate
(3) Think

$$p \sim \text{Distribution}(\ldots)$$

between 0–1, < 0.5

$$k \sim \text{Distribution}(\ldots)$$

positive real, > 1

How to set these priors?

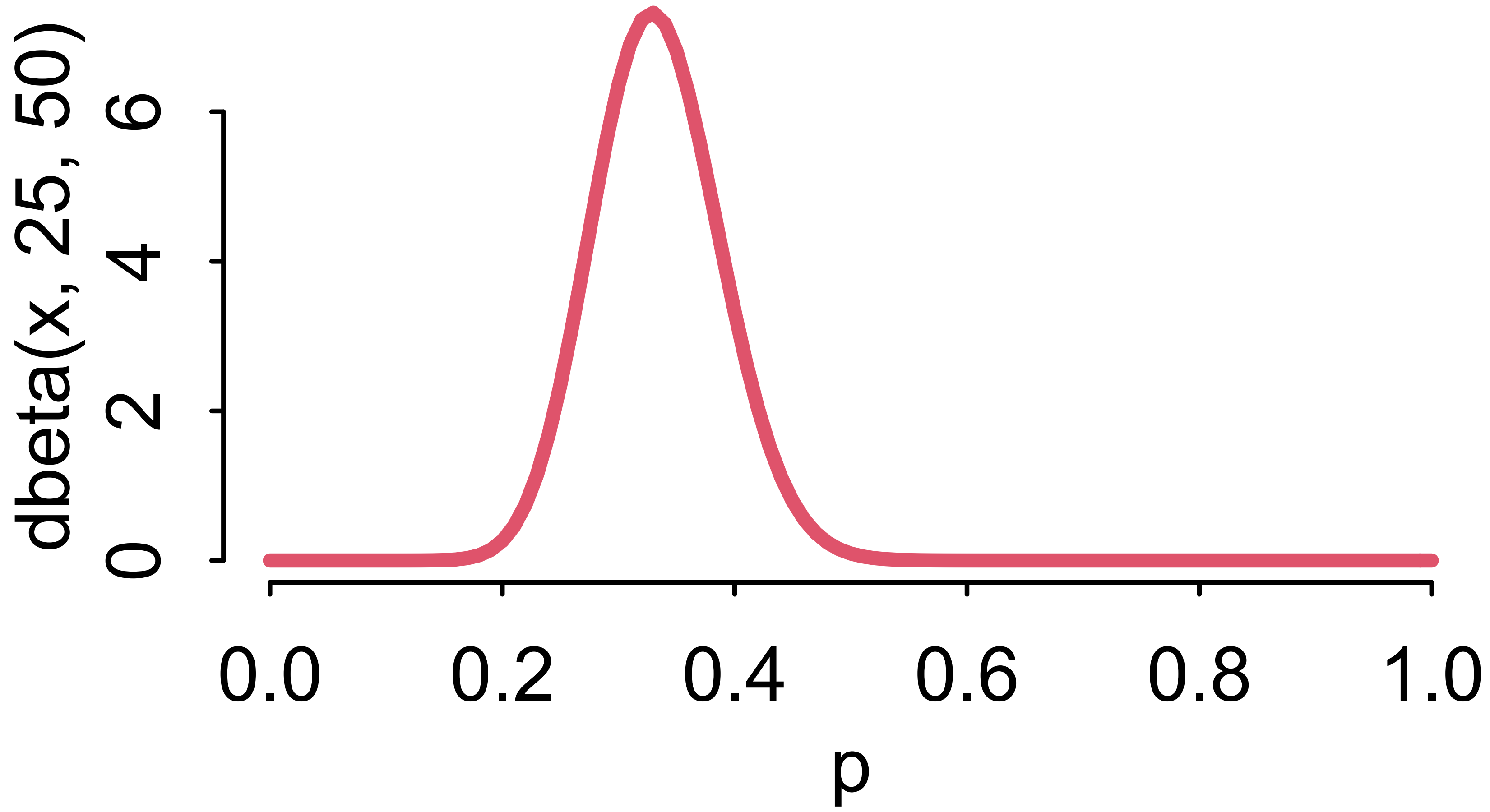
(1) Choose measurement scales

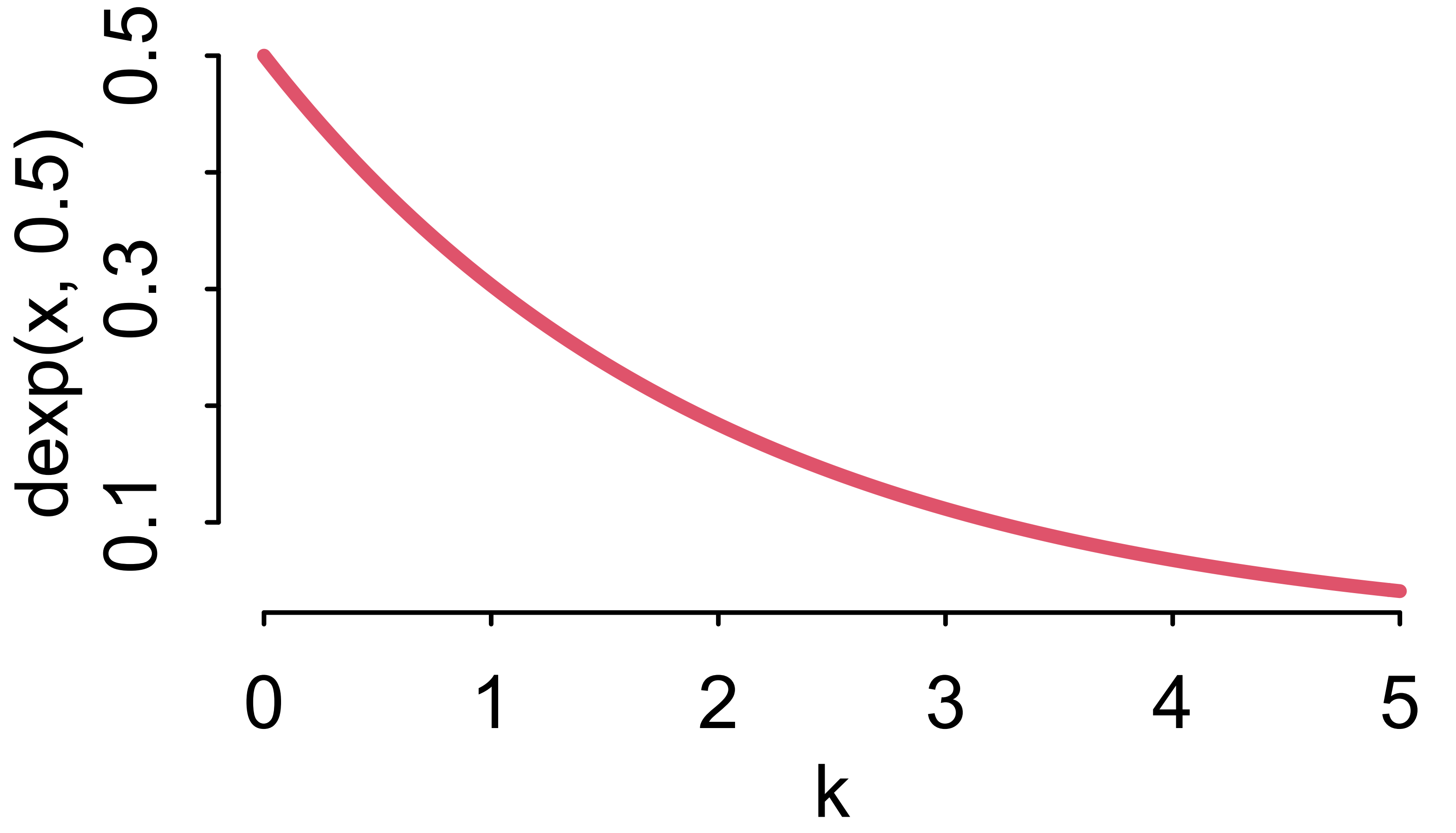
(2) Simulate

(3) Think

$$p \sim \text{Beta}(25,50)$$

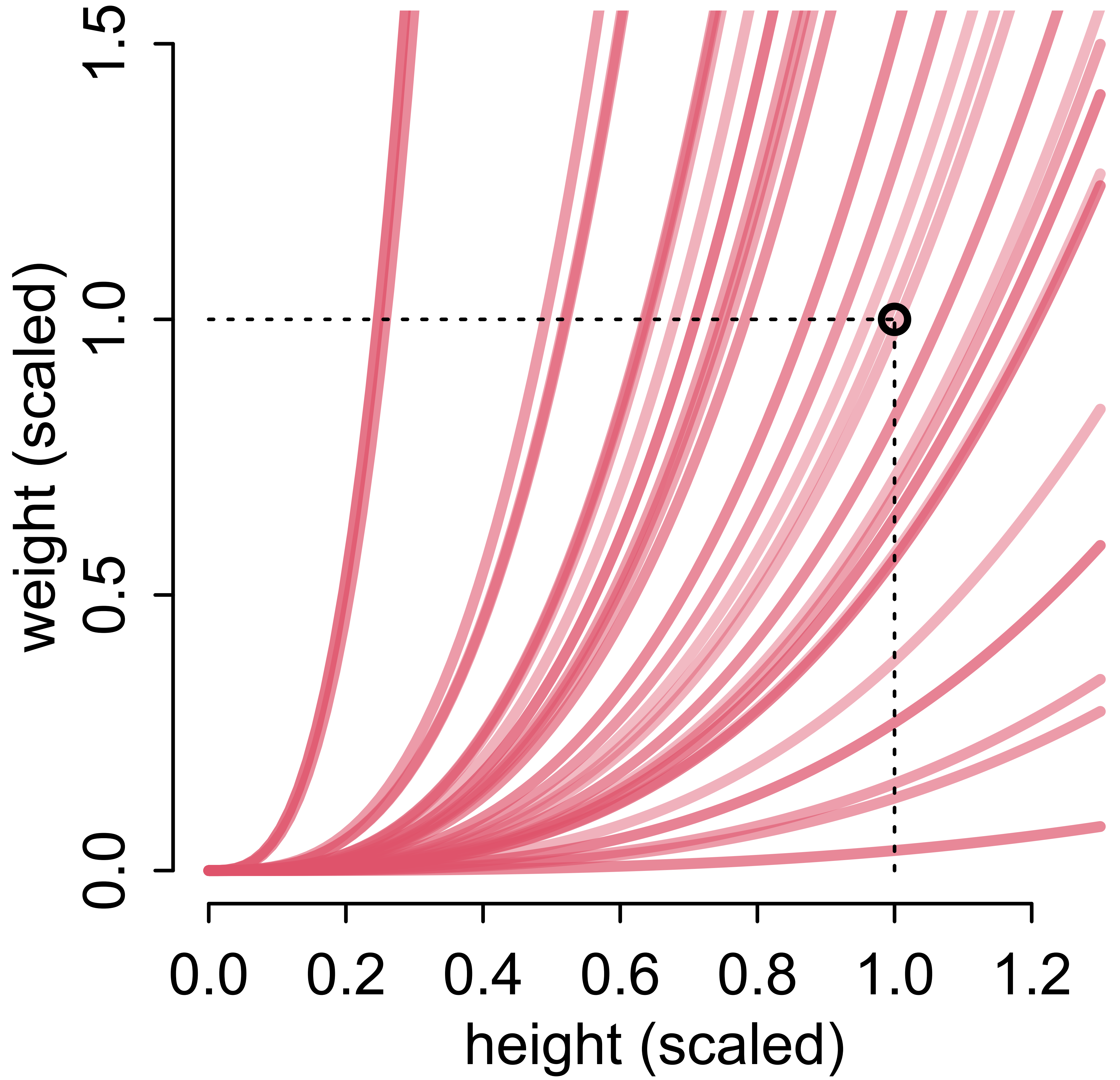
$$k \sim \text{Exponential}(0.5)$$





Prior predictive simulation

```r
# prior sim
n <- 30
p <- rbeta(n,25,50)
k <- rexp(n,0.5)
sigma <- rexp(n,1)

xseq <- seq(from=0,to=1.3,len=100)
plot(NULL,xlim=c(0,1.3),ylim=c(0,1.5))
for ( i in 1:n ) {
    mu <- log( pi * k[i] * p[i]^2 * xseq^3 )
    lines( xseq , exp(mu + sigma[i]^2/2) ,
lwd=3 , col=col.alpha(2,runif(1,0.4,0.8)) )
}
```



$$W_i \sim \text{Distribution}(\mu_i, \ldots)$$

positive real,
variance scales with mean

$$\mu_i = k\pi p^2 H_i^3$$

$$p \sim \text{Beta}(25{,}50)$$

$$k \sim \text{Exponential}(0.5)$$

$$W_i \sim \mathrm{LogNormal}(\mu_i, \sigma)$$
$$\exp(\mu_i) = k\pi p^2 H_i^3$$
$$p \sim \mathrm{Beta}(25, 50)$$
$$k \sim \mathrm{Exponential}(0.5)$$
$$\sigma \sim \mathrm{Exponential}(1)$$

Growth is multiplicative,
log-normal is natural choice

mu in log-normal is mean of log, not mean of observed

$$W_i \sim \text{LogNormal}(\mu_i, \sigma)$$

$$\exp(\mu_i) = k\pi p^2 H_i^3$$

$$p \sim \text{Beta}(25, 50)$$

$$k \sim \text{Exponential}(0.5)$$

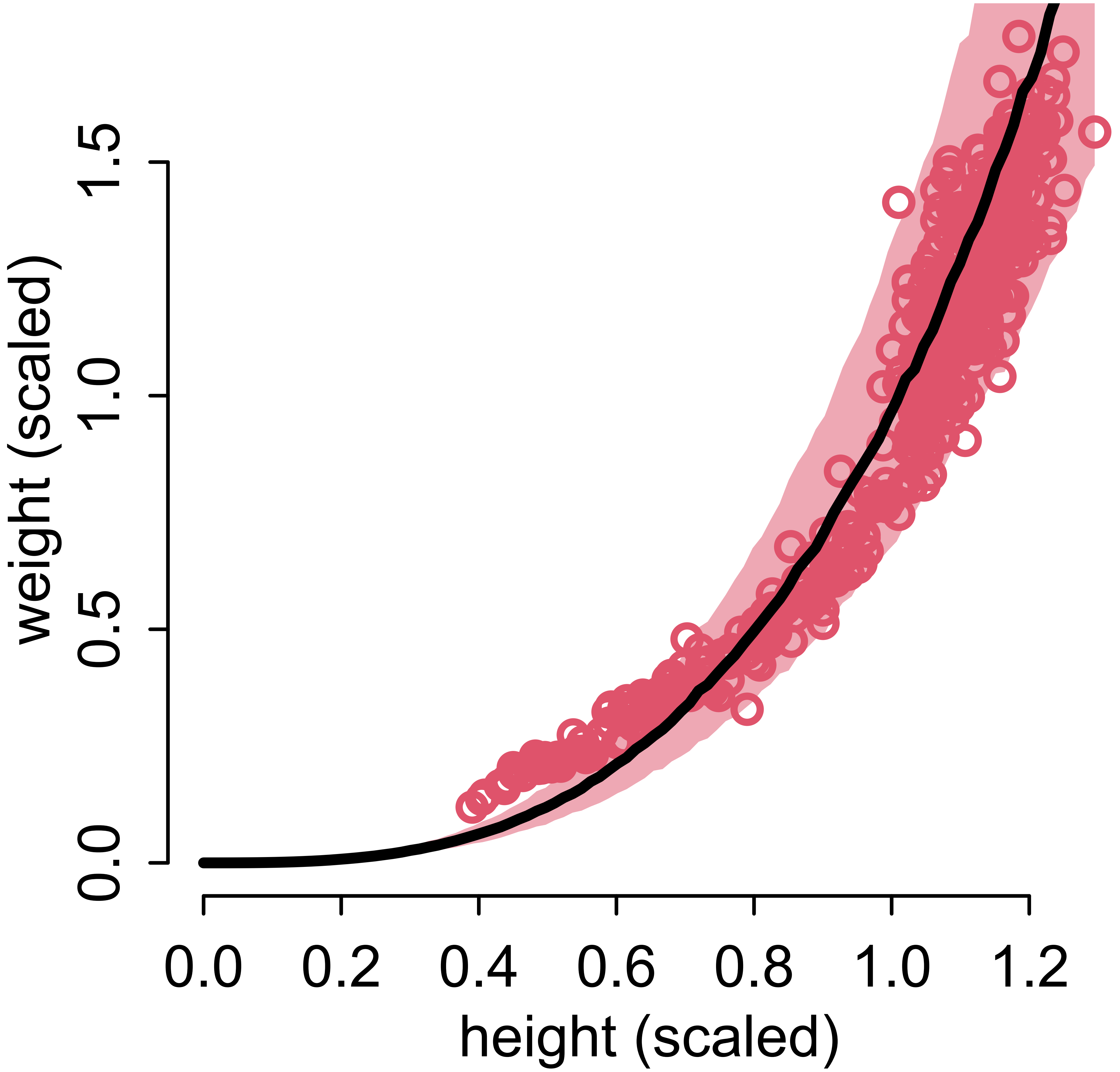
$$\sigma \sim \text{Exponential}(1)$$

Growth is multiplicative, log-normal is natural choice

```r
## R code 16.2
dat <- list(W=d$w,H=d$h)
m16.1 <- ulam(
    alist(
        W ~ dlnorm( mu , sigma ),
        exp(mu) <- 3.141593 * k * p^2 * H^3,
        p ~ beta( 25 , 50 ),
        k ~ exponential( 0.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 )
```

$$W_i \sim \text{LogNormal}(\mu_i, \sigma)$$
$$\exp(\mu_i) = k\pi p^2 H_i^3$$
$$p \sim \text{Beta}(25,50)$$
$$k \sim \text{Exponential}(0.5)$$
$$\sigma \sim \text{Exponential}(1)$$

```
## R code 16.2
dat <- list(W=d$w,H=d$h)
m16.1 <- ulam(
    alist(
        W ~ dlnorm( mu , sigma ),
        exp(mu) <- 3.141593 * k * p^2 * H^3,
        p ~ beta( 25 , 50 ),
        k ~ exponential( 0.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 )
```



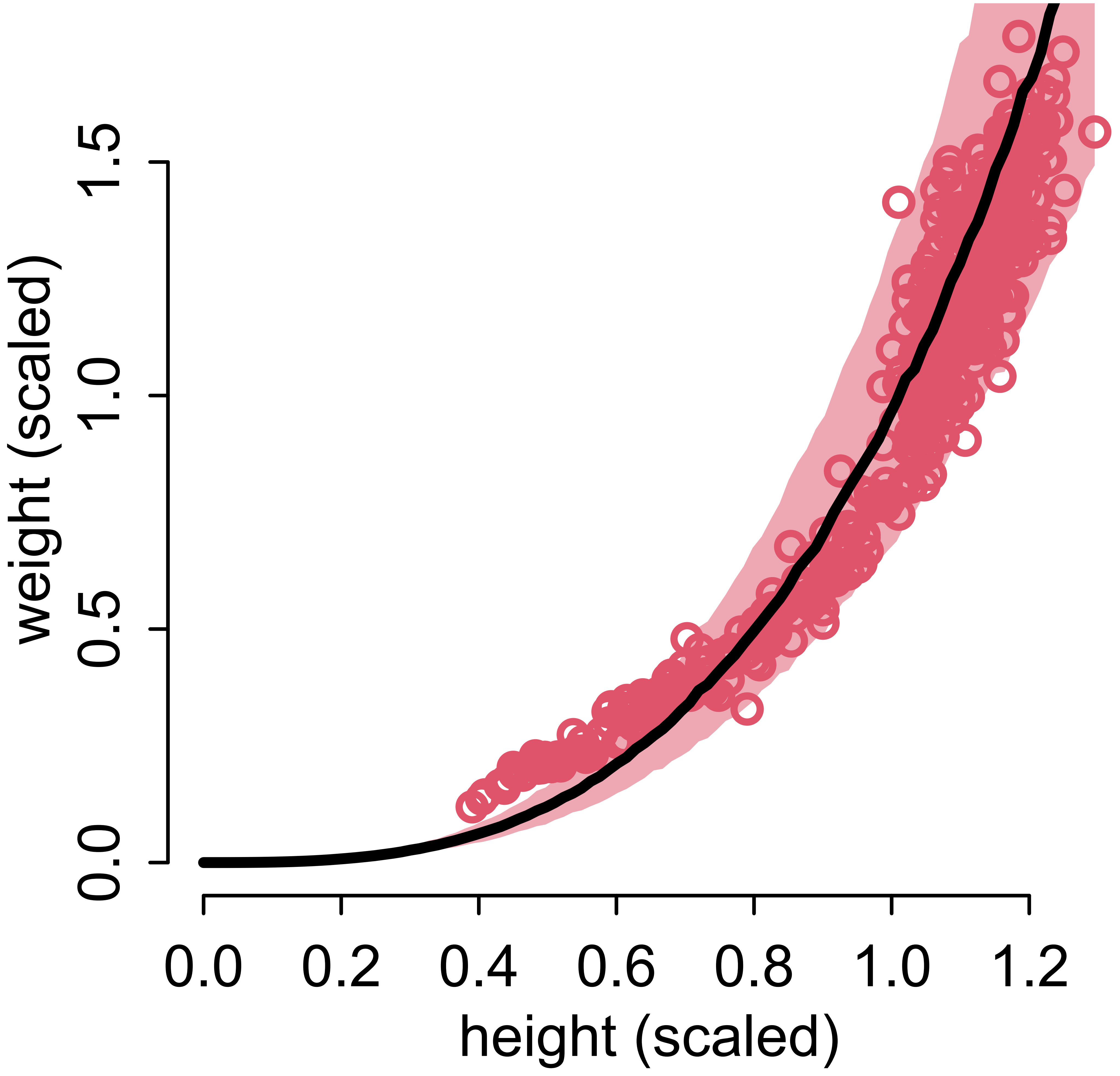
Insightful errors

Not bad for a cylinder

Poor fit for children

In scientific model, poor fit is informative — p different for kids

Bad epicycles harder to read



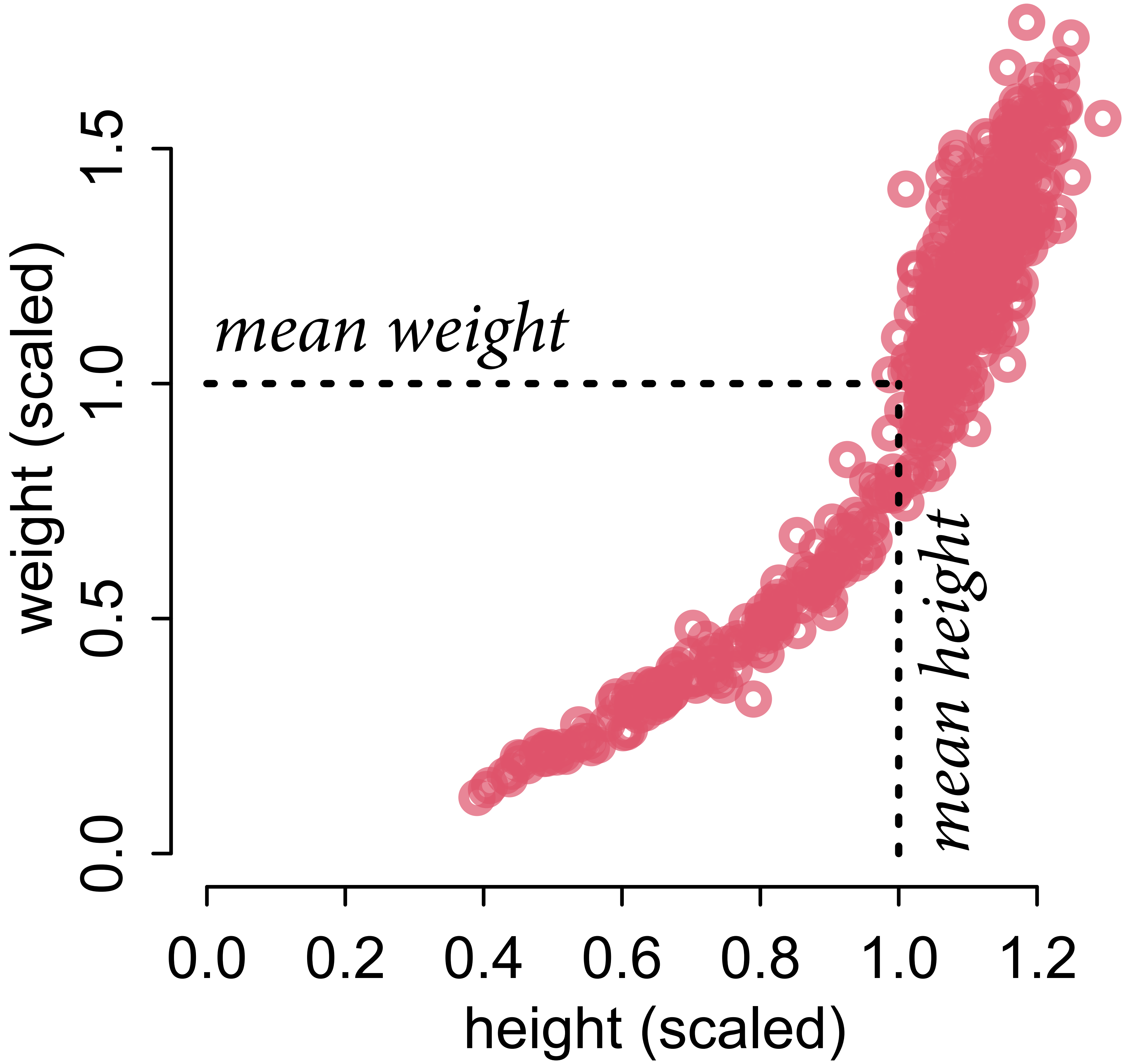
How to set these priors?

(1) Choose measurement scales

(2) Simulate

(3) Think

$$\mu_i = k\pi p^2 H_i^3$$



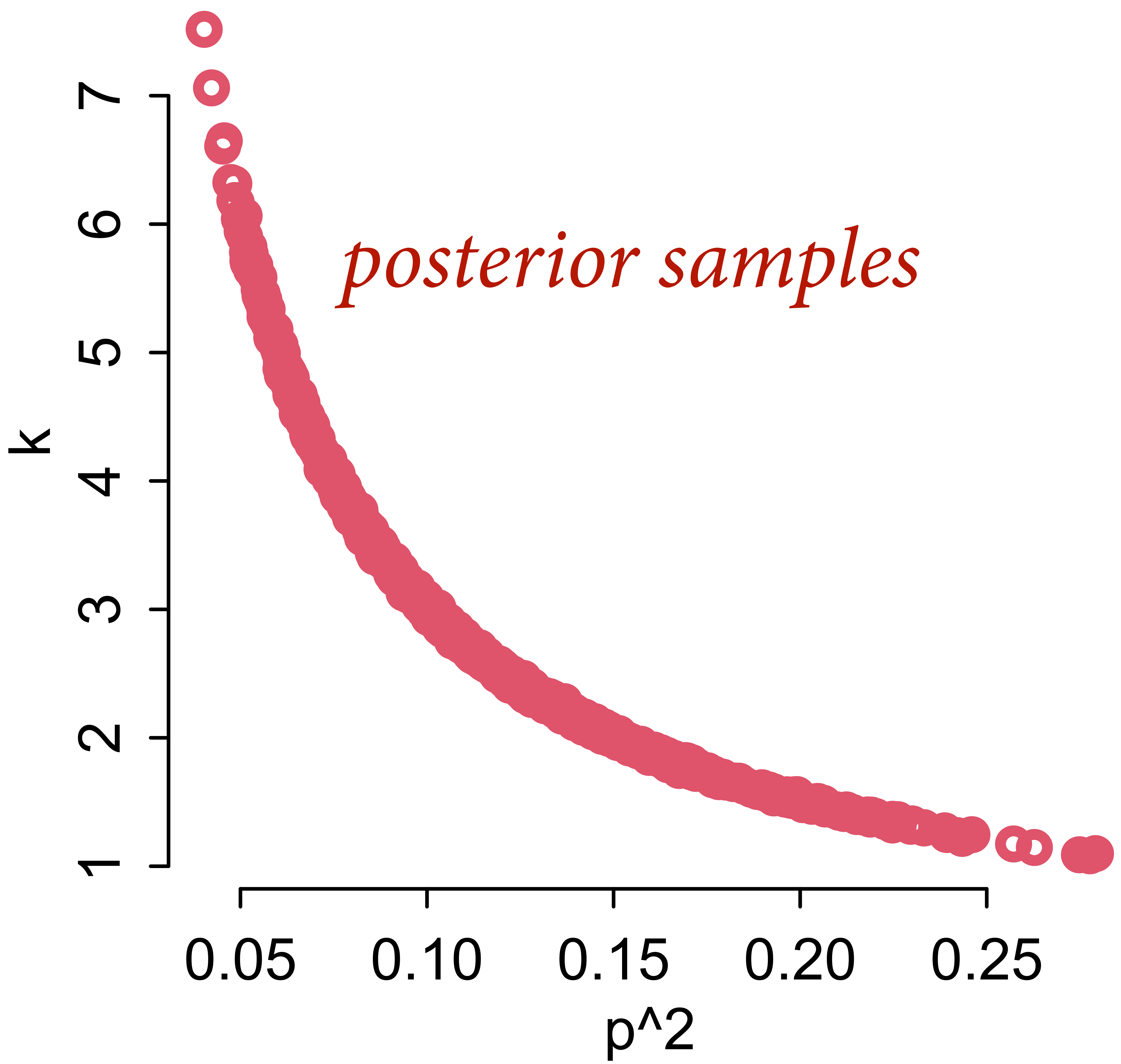
How to set these priors?

(1) Choose measurement scales

(2) Simulate

(3) Think

$$\mu_i = k\pi p^2 H_i^3$$



How to set these priors?

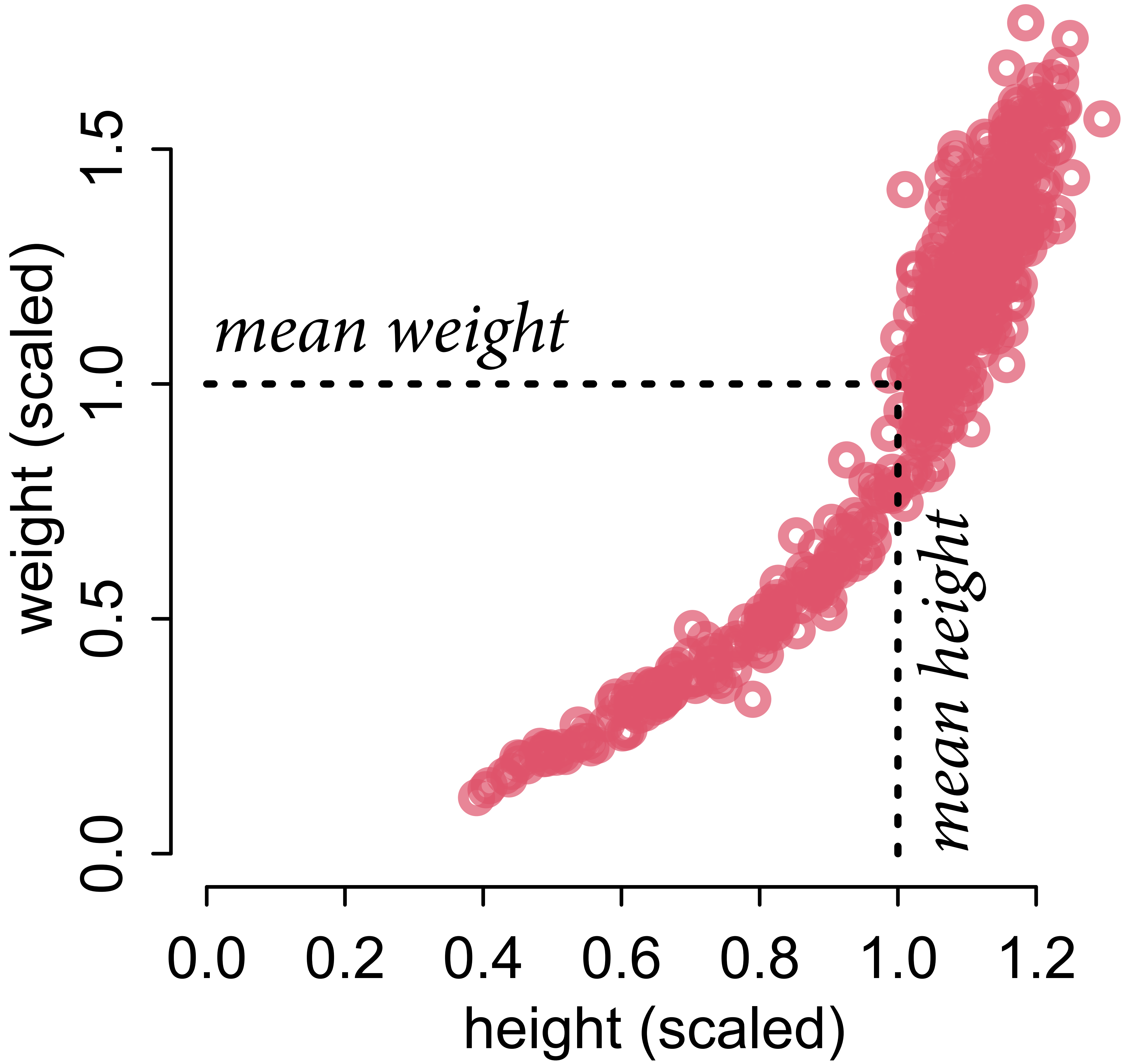
(1) Choose measurement scales

(2) Simulate

(3) Think

$$\mu_i = k\pi p^2 H_i^3$$

$$(1) = k\pi p^2 (1)^3$$



How to set these priors?

(1) Choose measurement scales

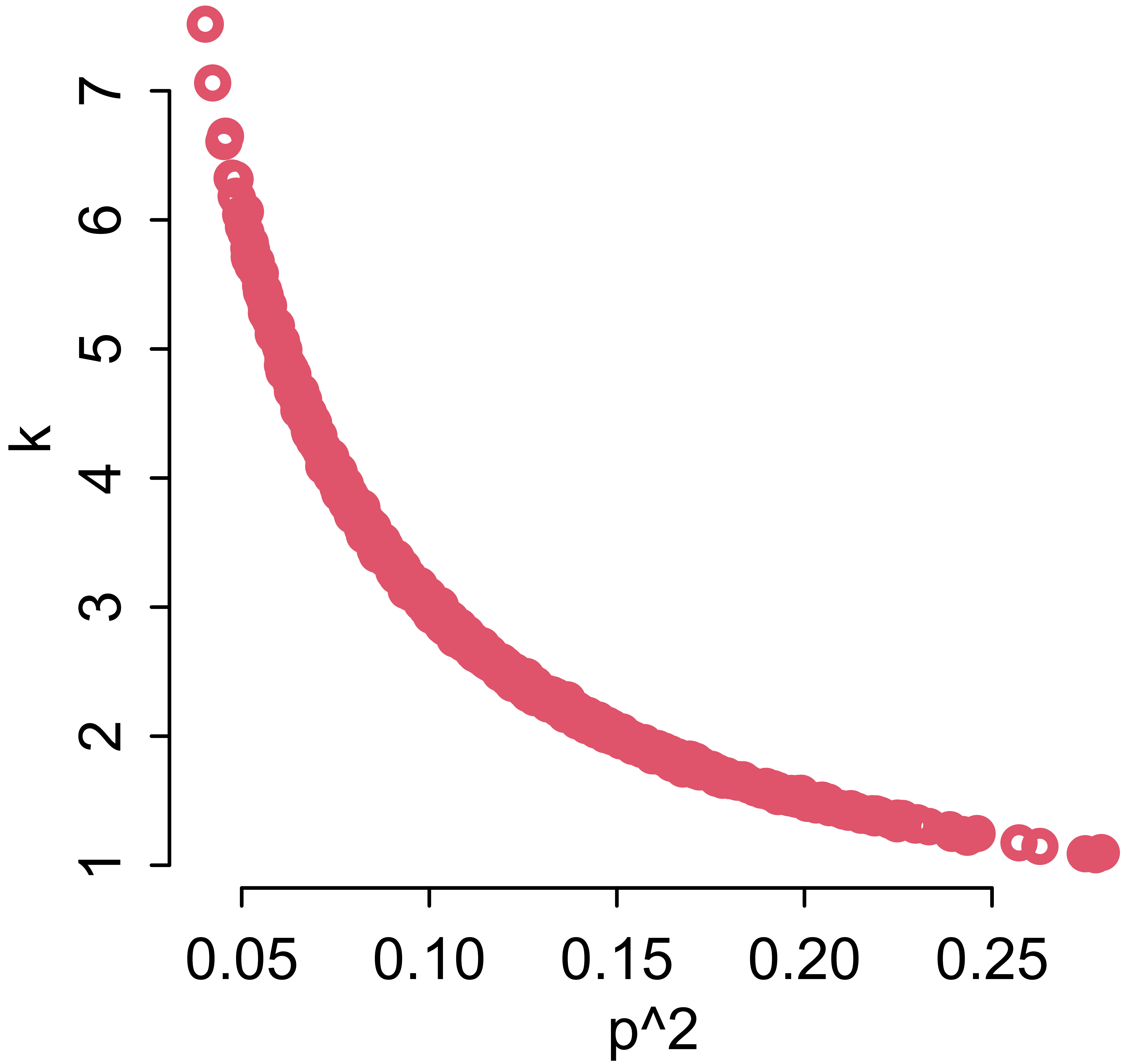
(2) Simulate

(3) Think

$$\mu_i = k\pi p^2 H_i^3$$

$$(1) = k\pi p^2 (1)^3$$

$$k = \frac{1}{\pi p^2}$$



How to set these priors?

(1) Choose measurement scales

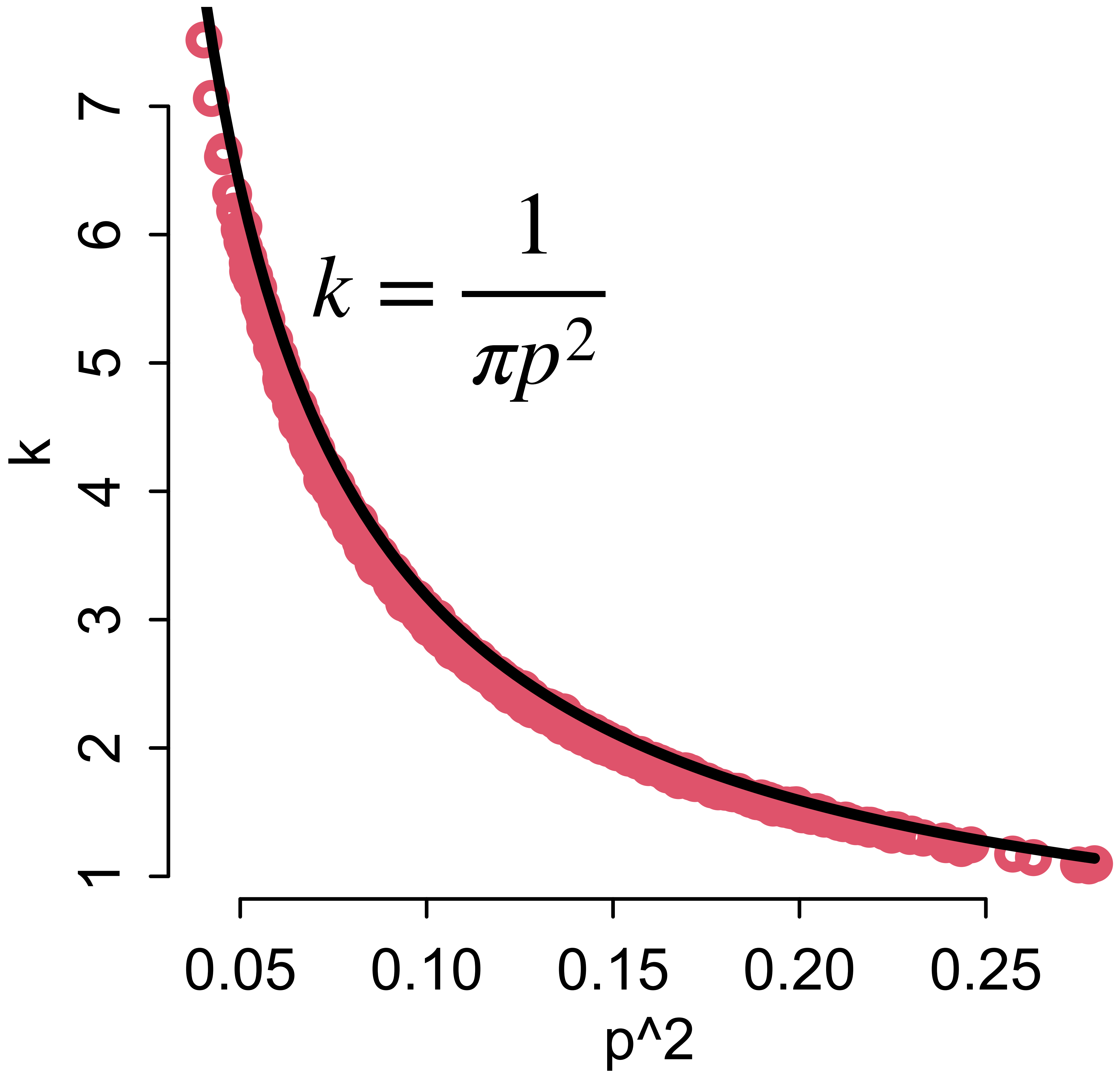
(2) Simulate

(3) Think

$$\mu_i = k\pi p^2 H_i^3$$

$$(1) = k\pi p^2 (1)^3$$

$$k = \frac{1}{\pi p^2}$$



How to set these priors?

(1) Choose measurement scales

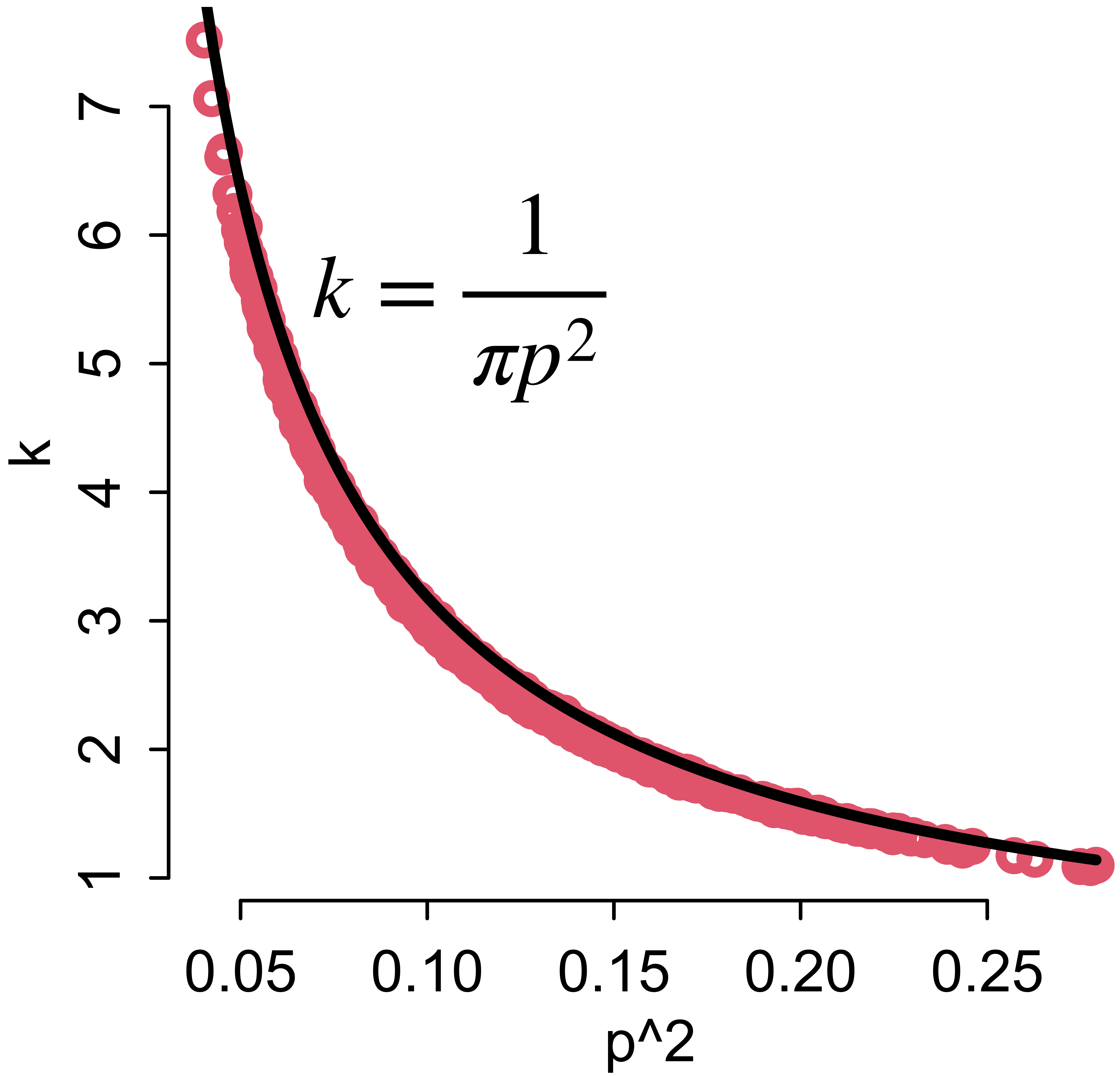
(2) Simulate

(3) Think

$$(1) = k \pi p^2 (1)^3$$

$$(1) = \pi \theta (1)^3$$

$$\theta \approx \pi^{-1}$$



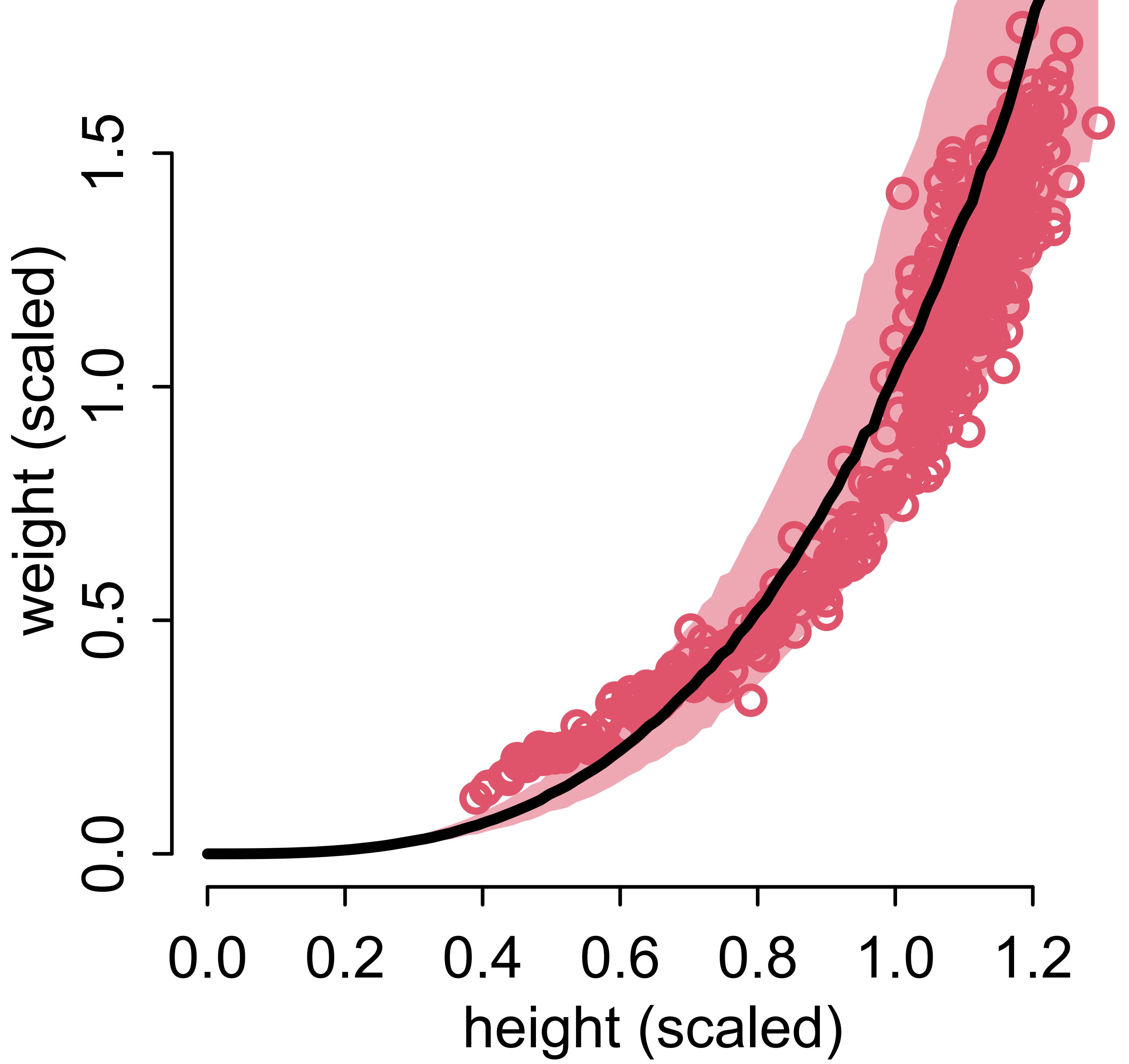
```
mWH2 <- ulam(
    alist(
        W ~ dlnorm( mu , sigma ),
        exp(mu) <- H^3 ,
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 )
```

$$W_i \sim \text{LogNormal}(\mu_i, \sigma)$$
$$\exp(\mu_i) = H_i^3$$
$$\sigma \sim \text{Exponential}(1)$$

In dimensionless model, W is H^3

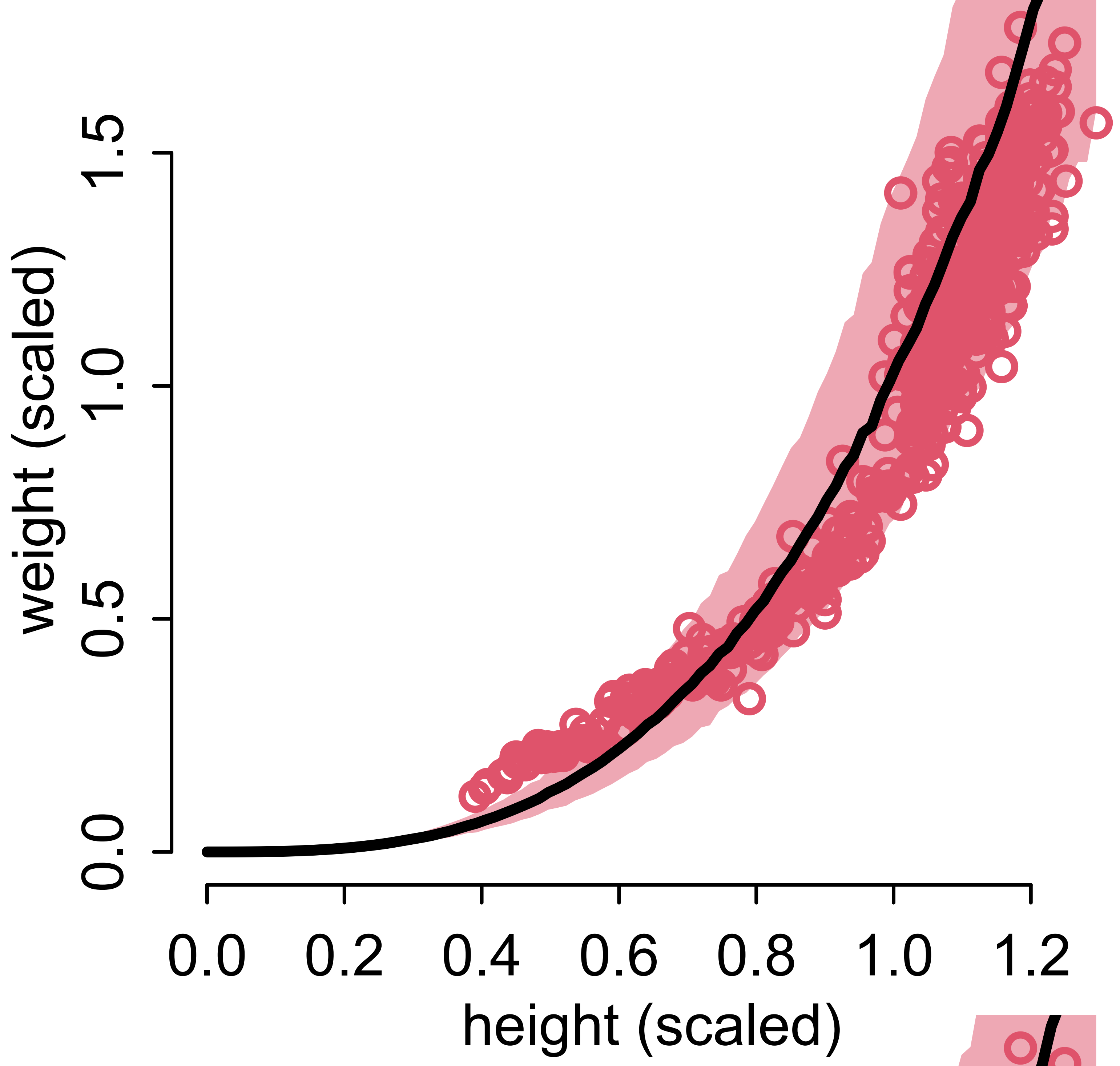
```r
mWH2 <- ulam(
    alist(
        W ~ dlnorm( mu , sigma ),
        exp(mu) <- H^3 ,
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 )
```

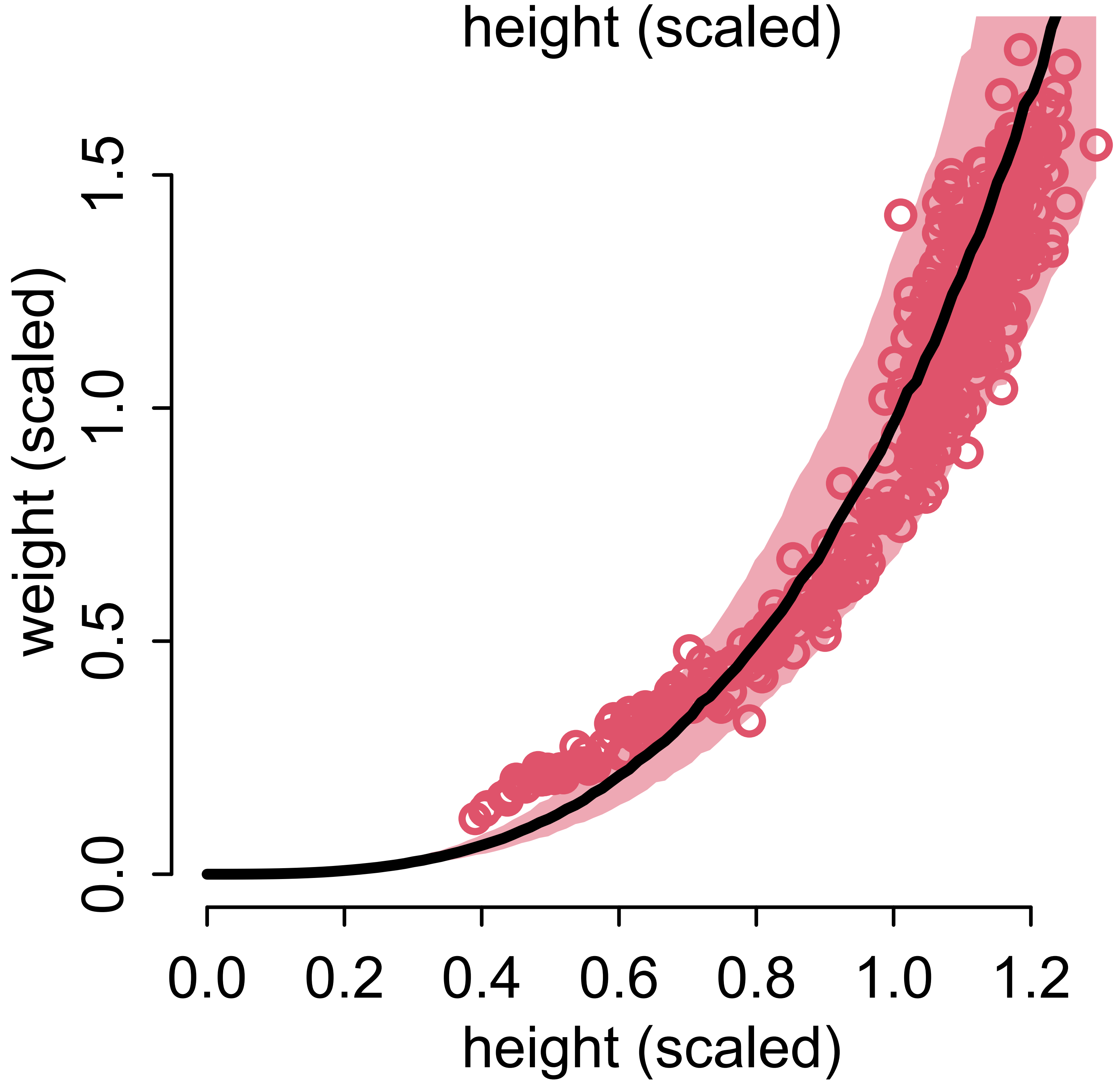
In dimensionless model, W is H^3



```
mWH2 <- ulam(
    alist(
        W ~ dlnorm( mu , sigma ),
        exp(mu) <- H^3 ,
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 )
```

```
## R code 16.2
dat <- list(W=d$w,H=d$h)
m16.1 <- ulam(
    alist(
        W ~ dlnorm( mu , sigma ),
        exp(mu) <- 3.141593 * k * p^2 * H^3,
        p ~ beta( 25 , 50 ),
        k ~ exponential( 0.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 )
```





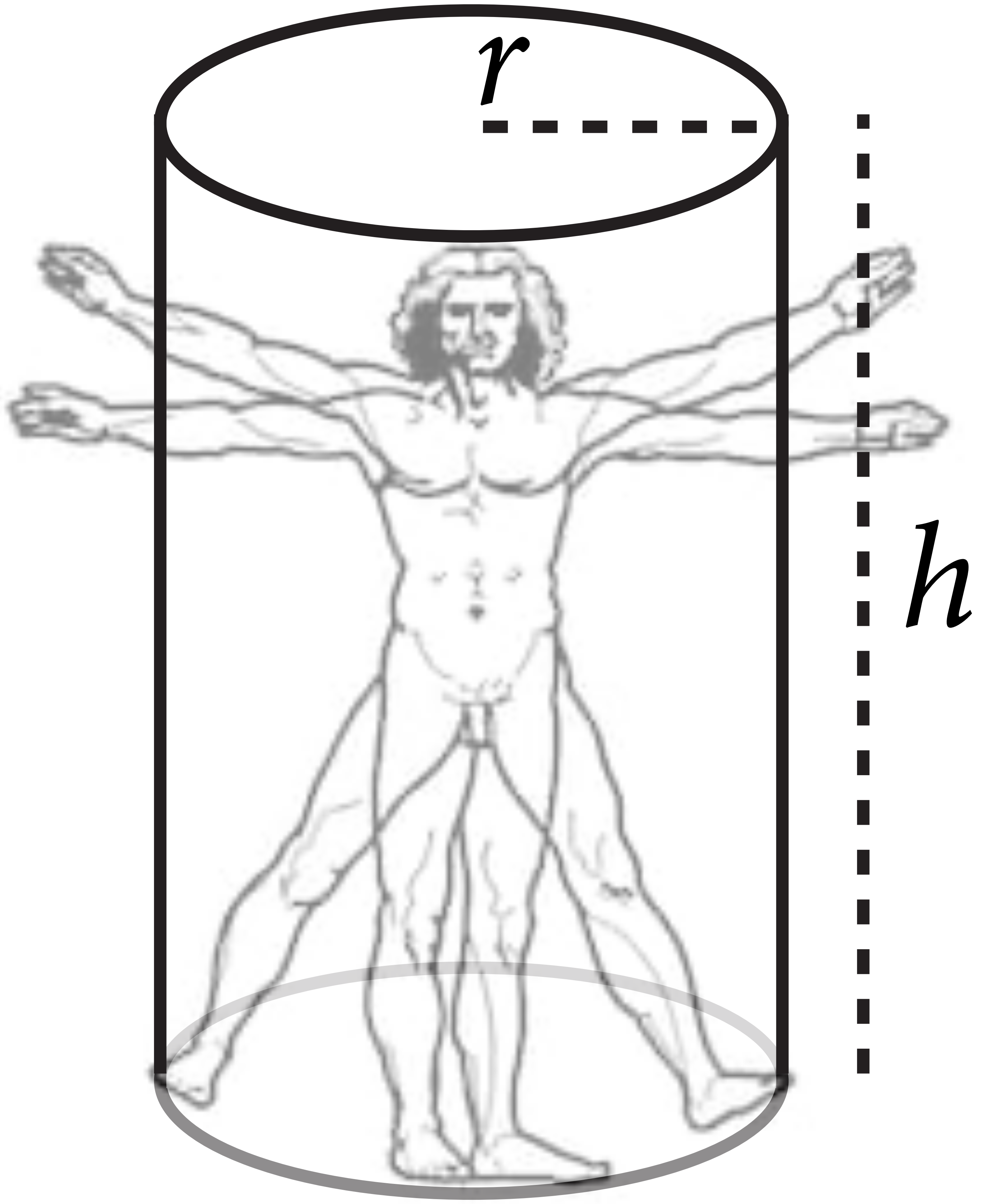
Geometric People

Most of the relationship *H* –> *W* is just relationship between length and volume

Changes in body shape explain poor fit for children?

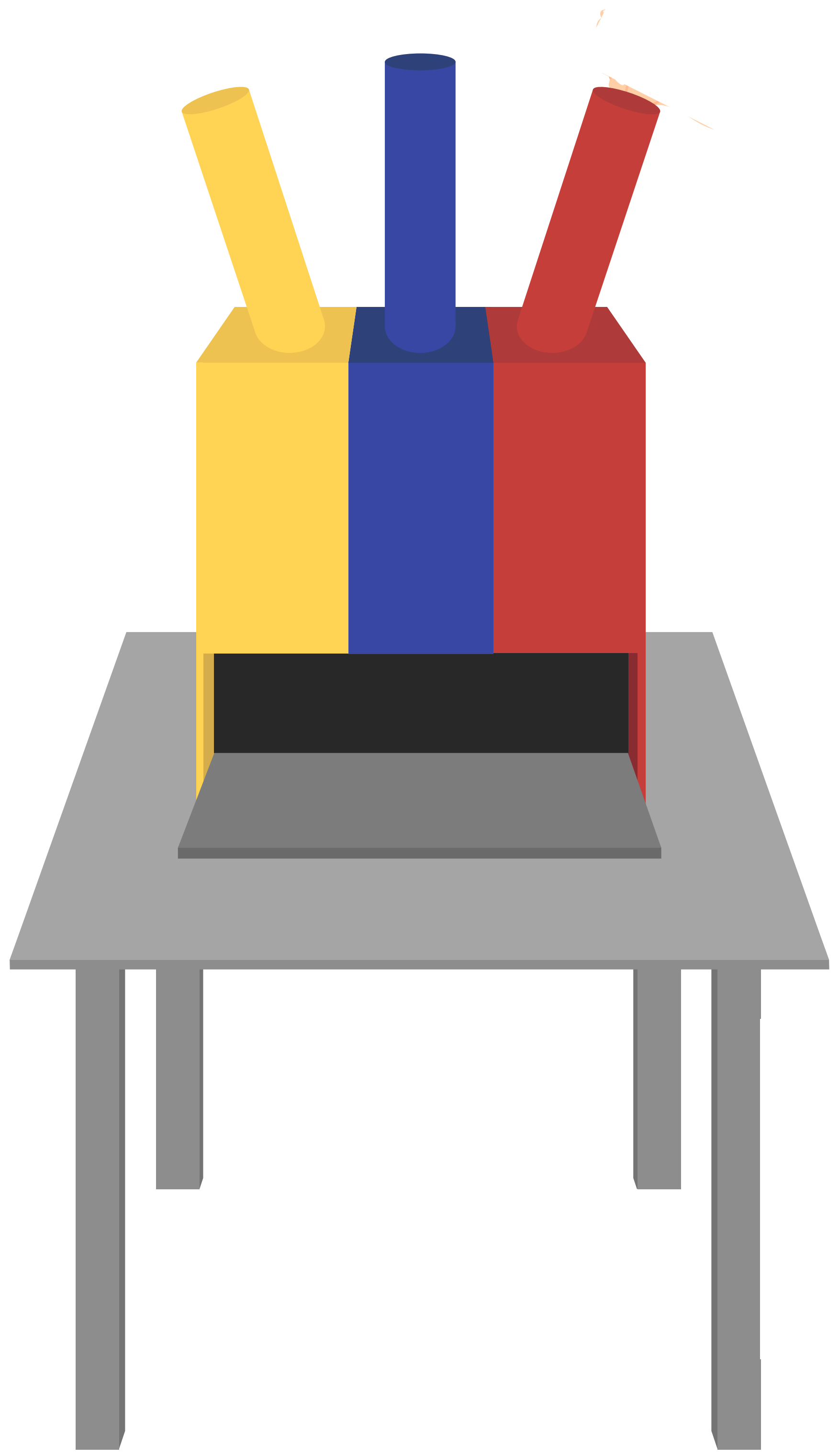
Problems provide insight when model is scientific instead of purely statistical

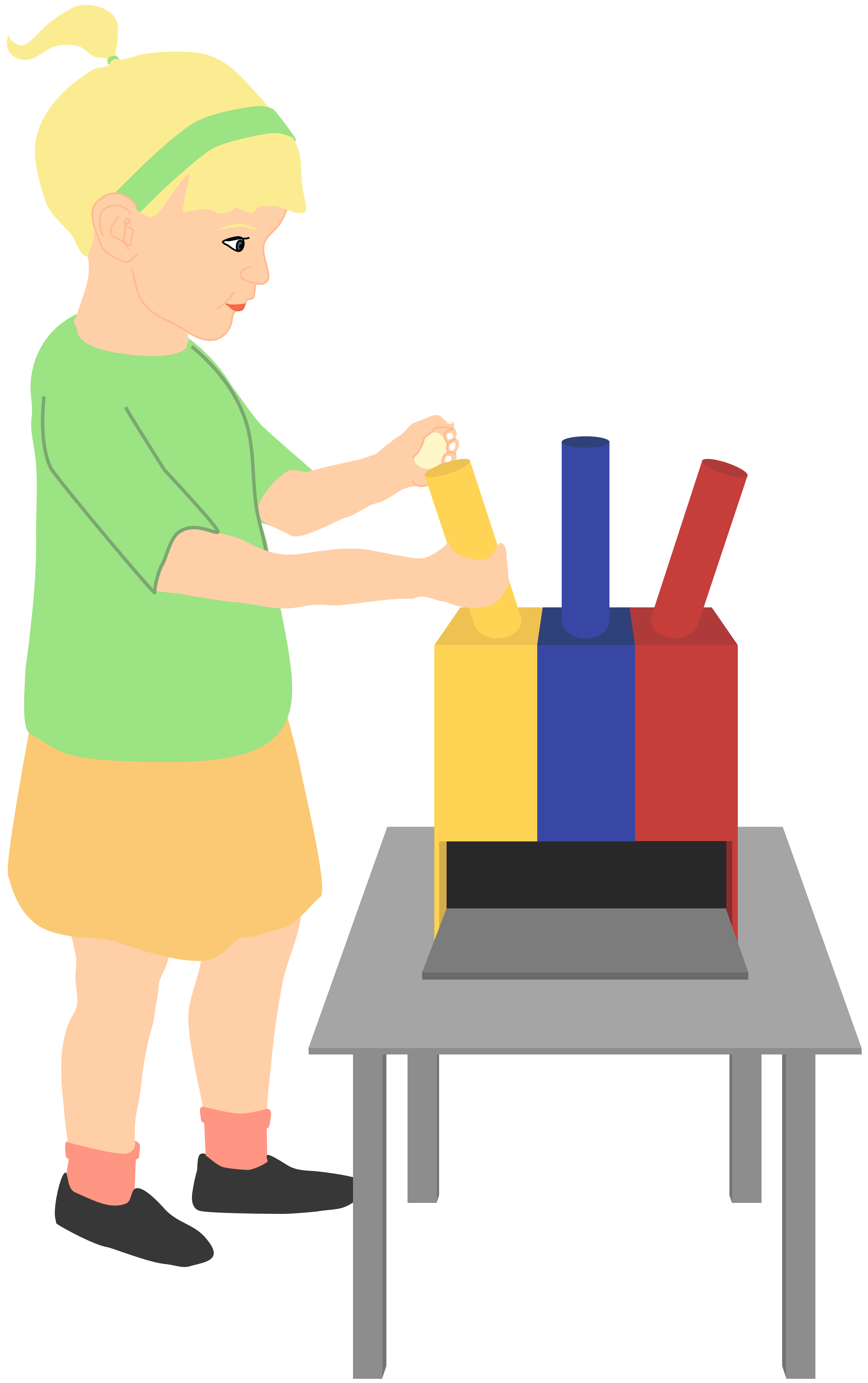
There is no empiricism without theory

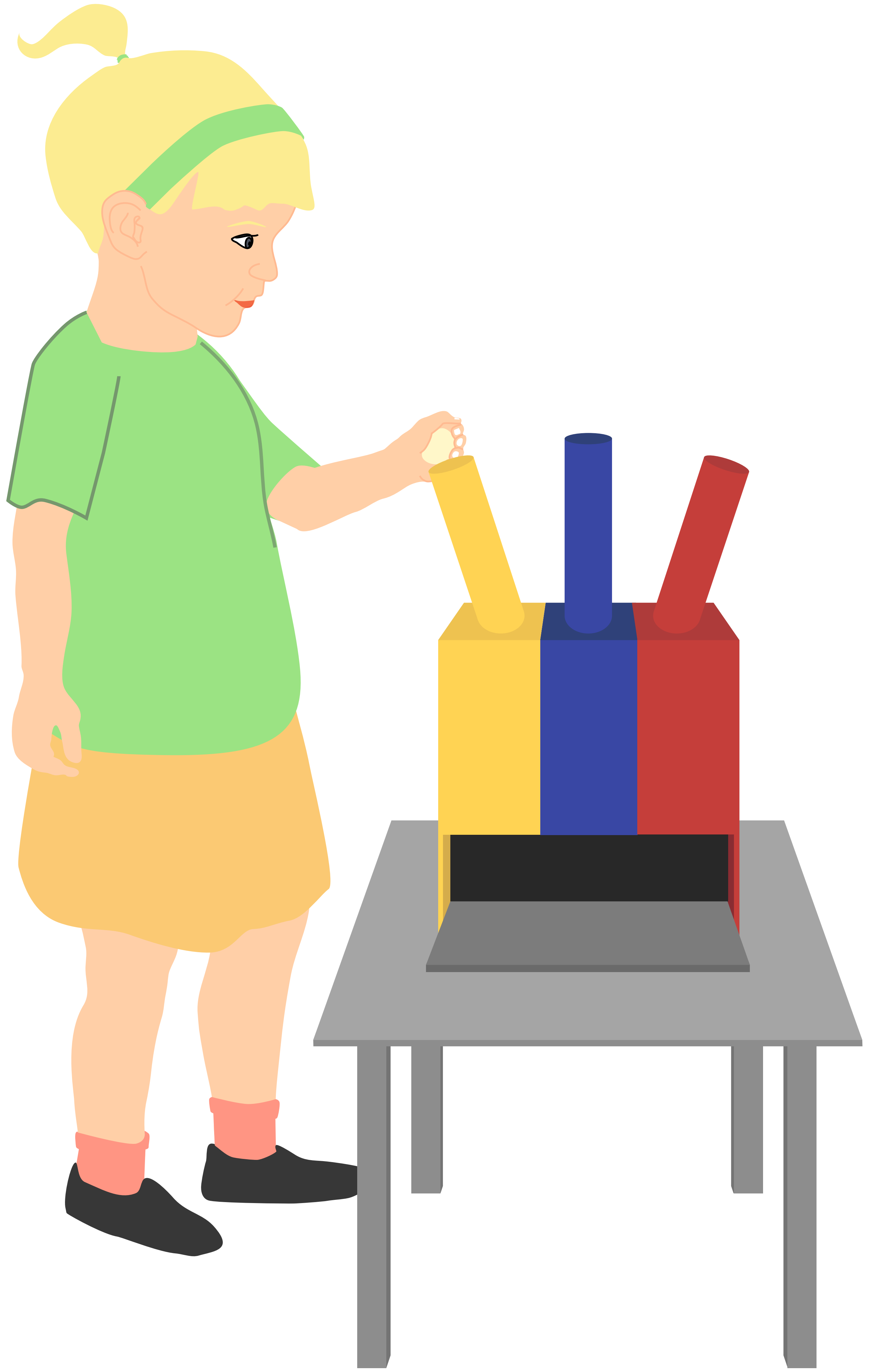


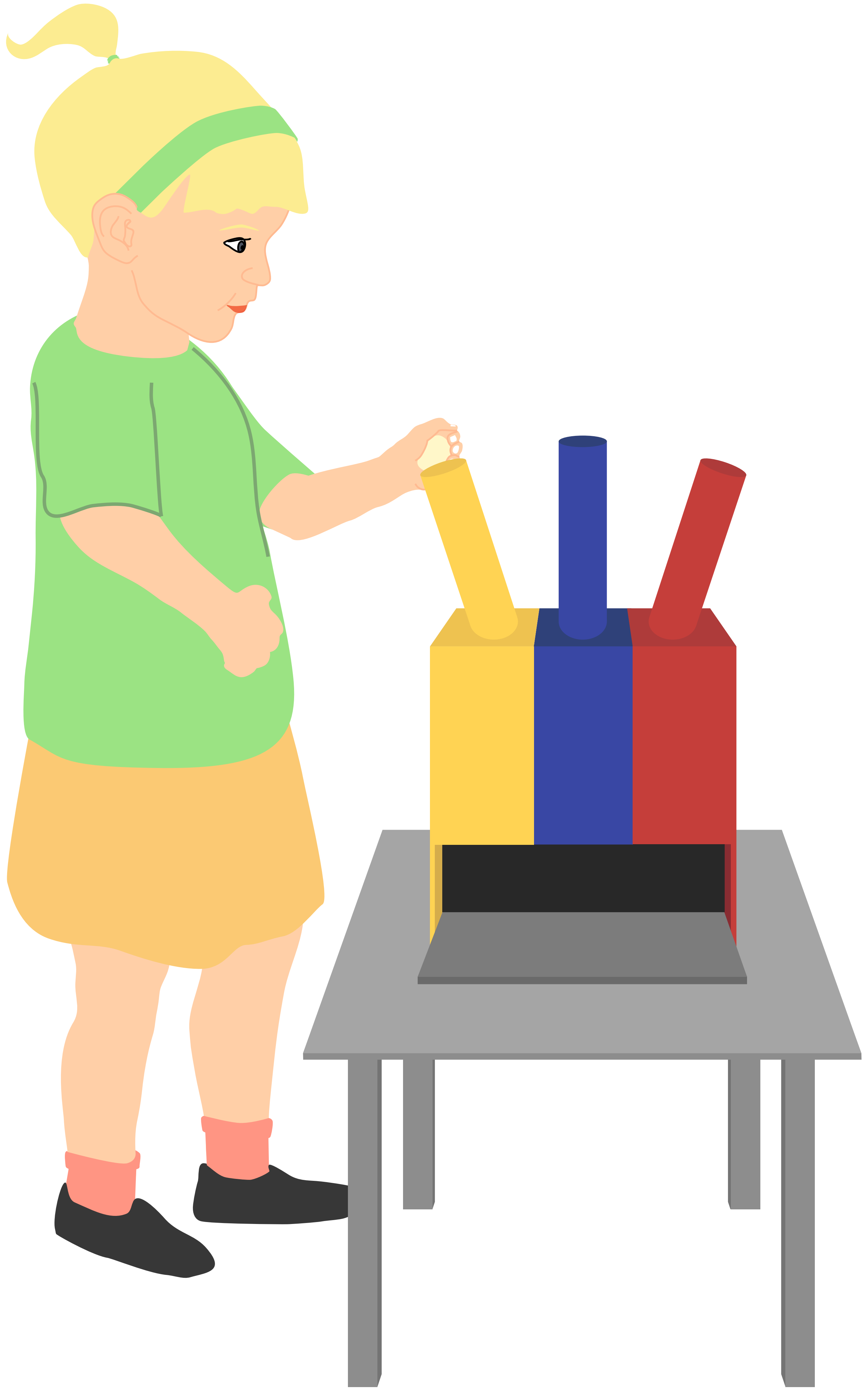
$$W = k\pi p^2 h^3$$

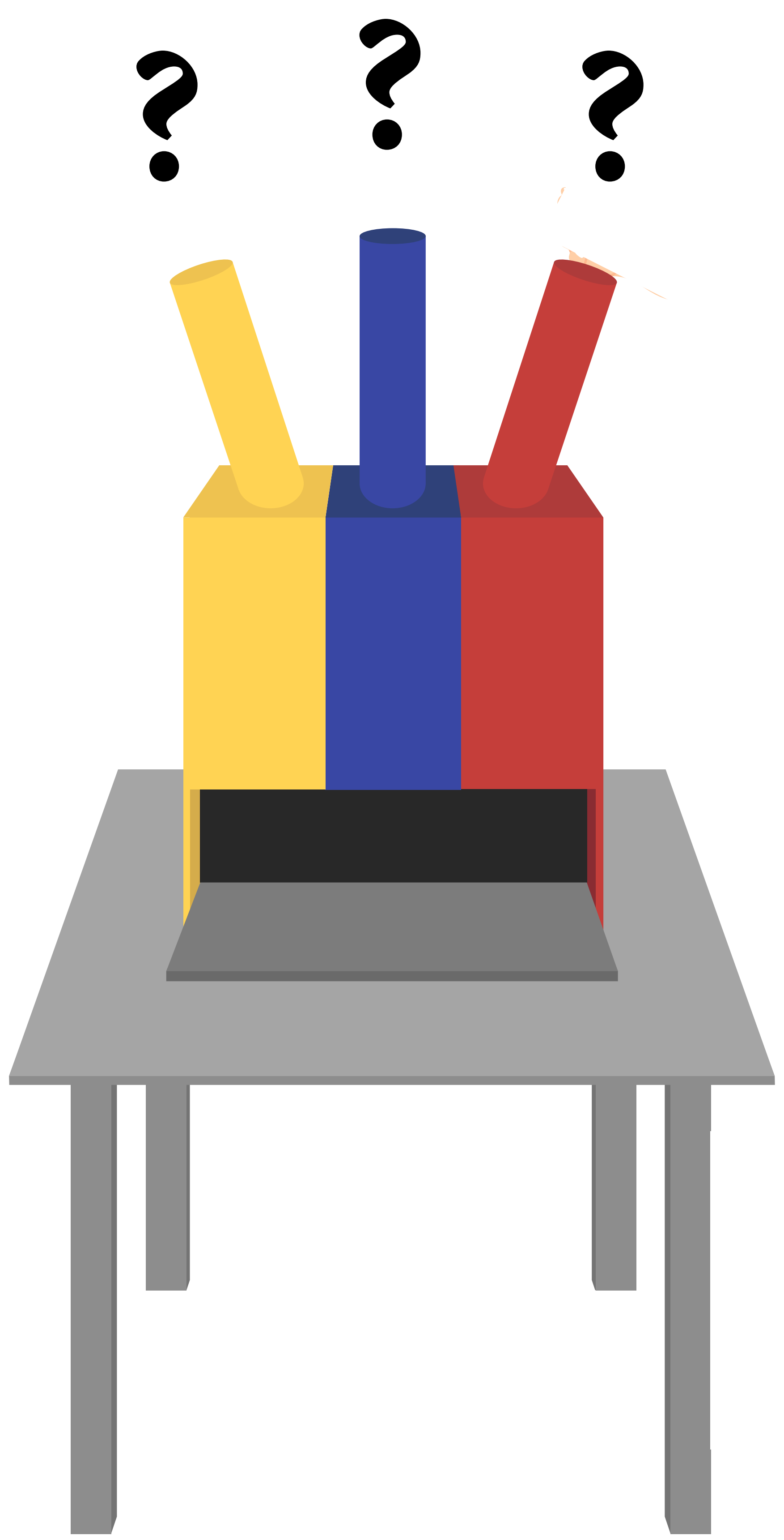
PAUSE

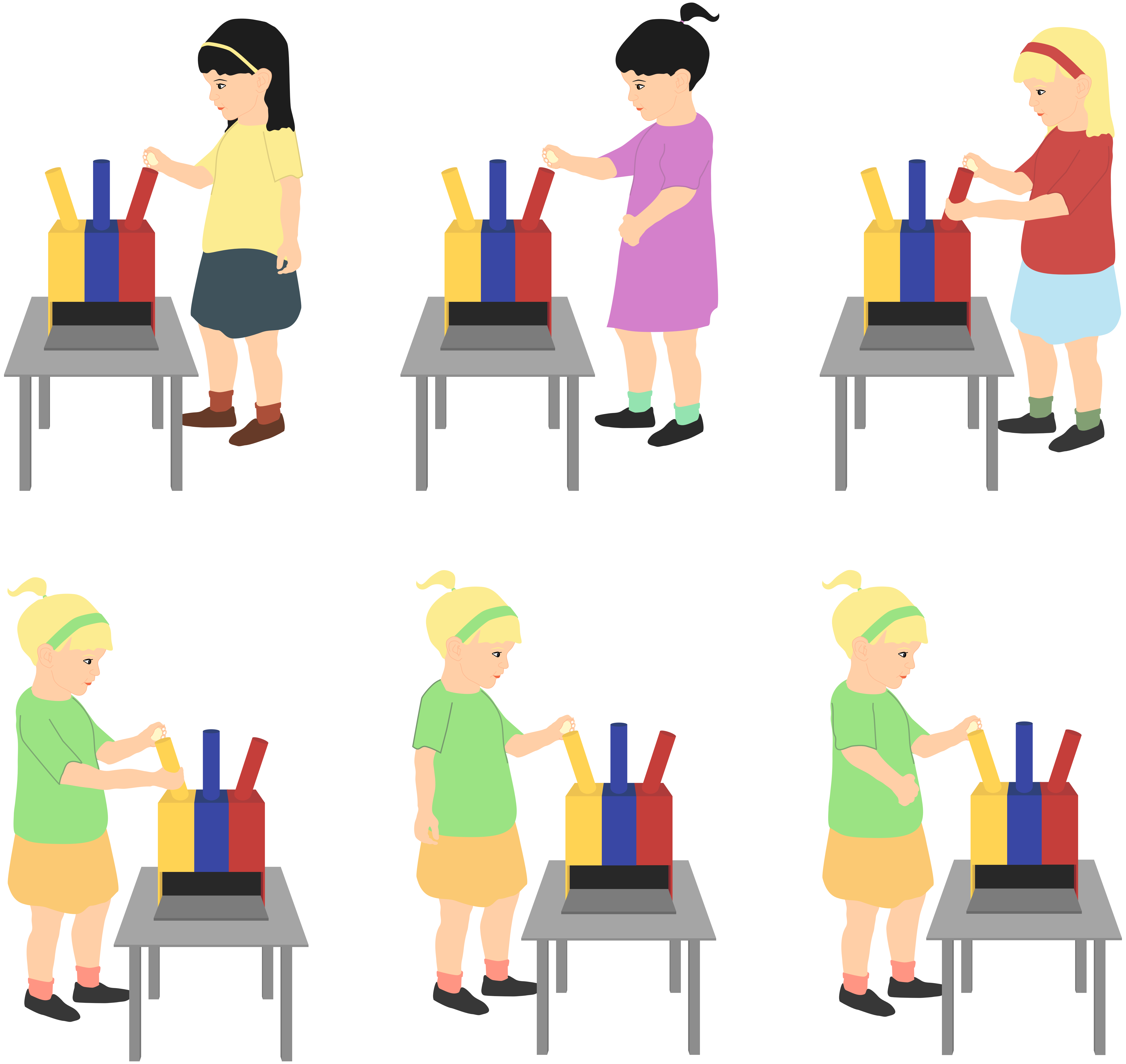


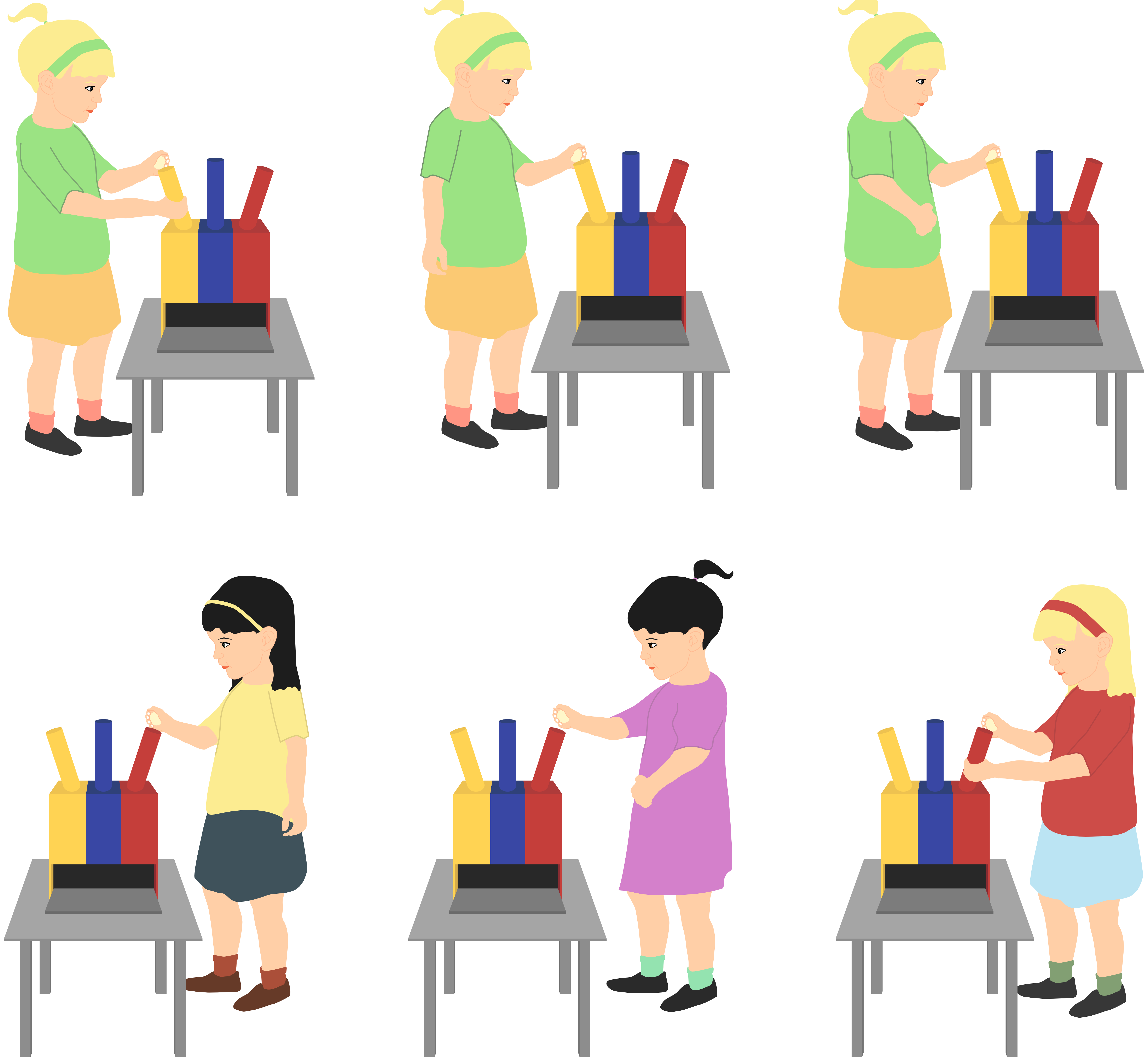


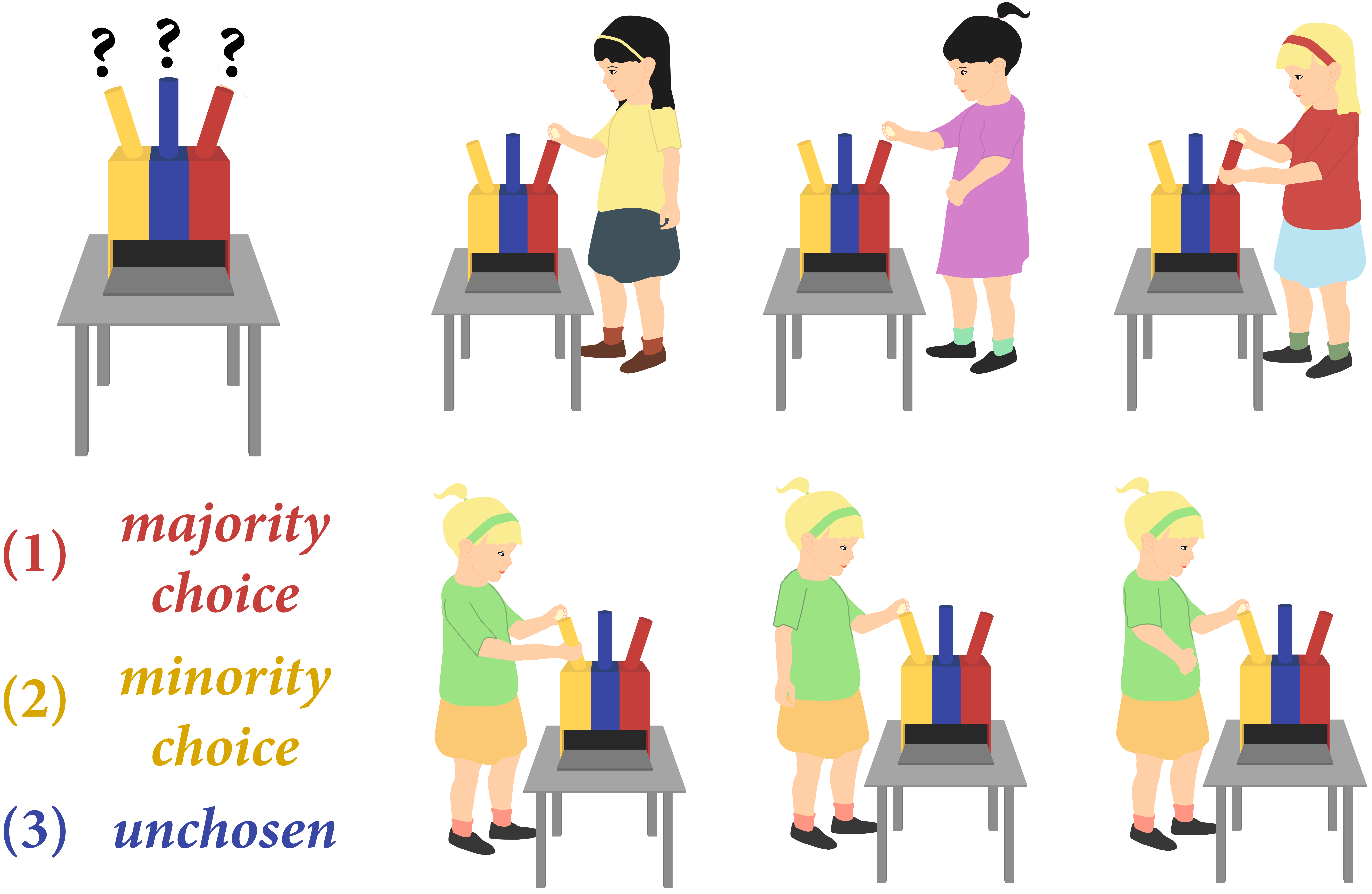










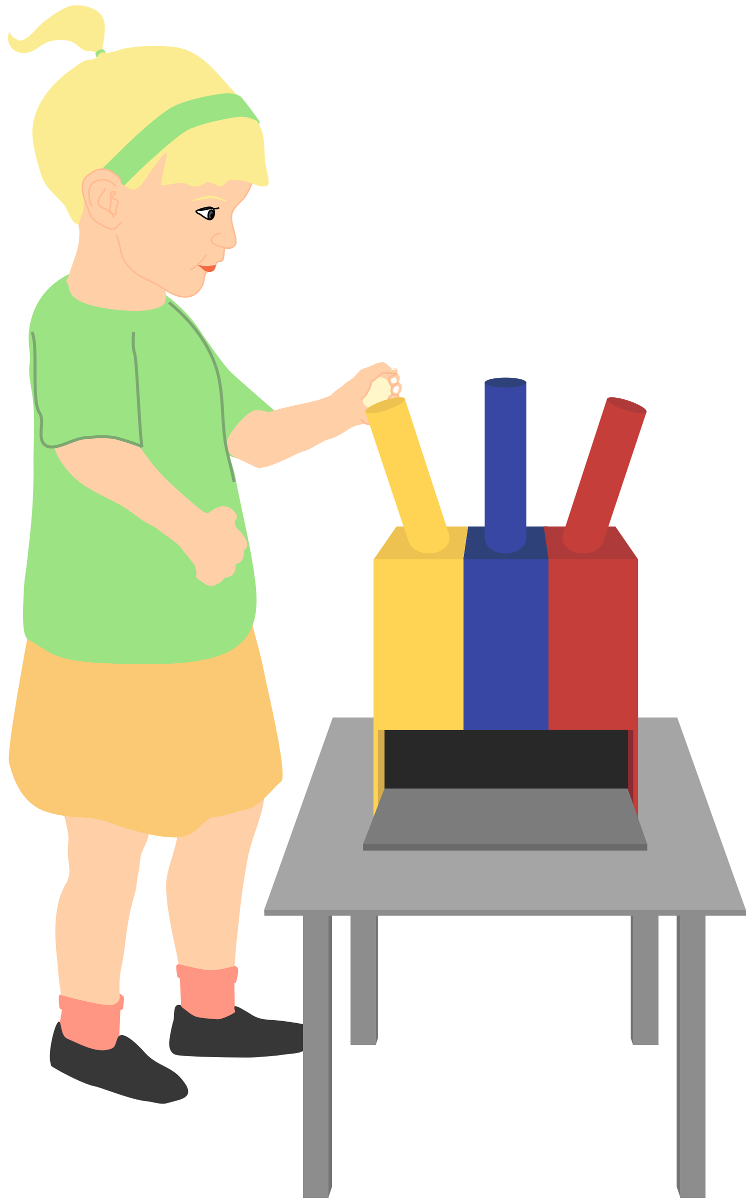

(1) majority choice
(2) minority choice
(3) unchosen

Social Conformity

Do children copy the majority? If so, how does this develop?

Problem: Cannot see strategy, only choice

Majority choice consistent with many strategies



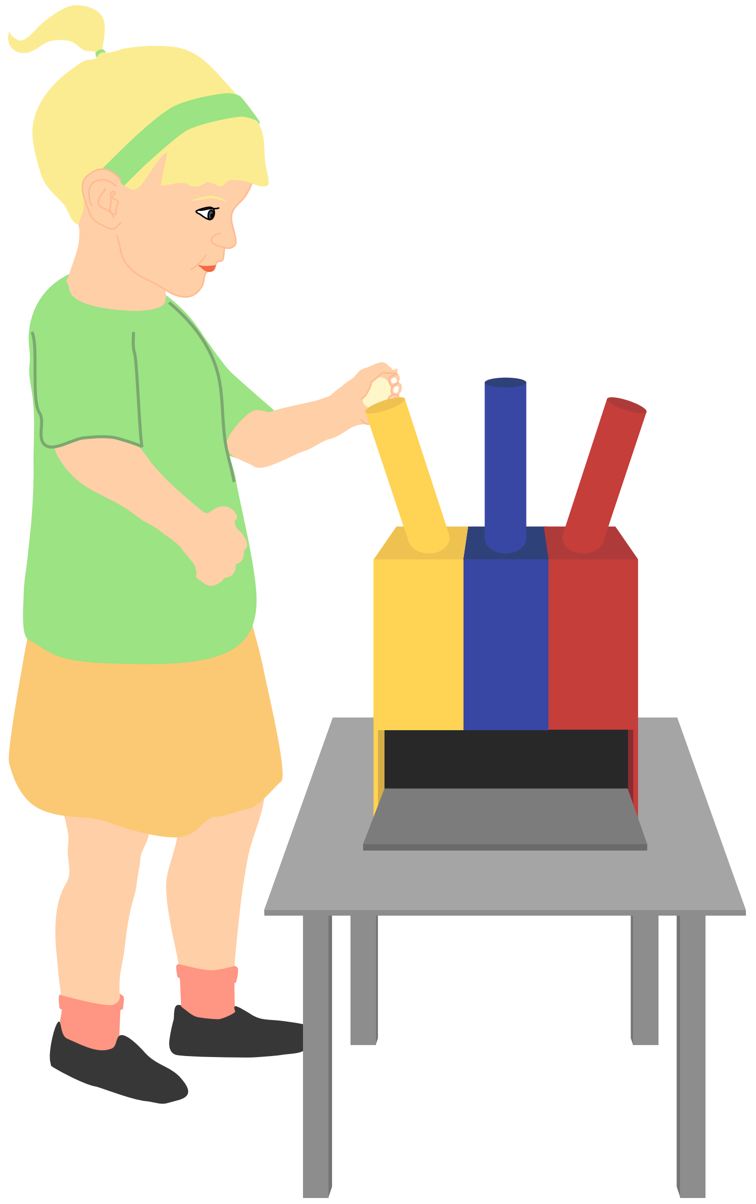
Social Conformity

Majority choice consistent with many strategies

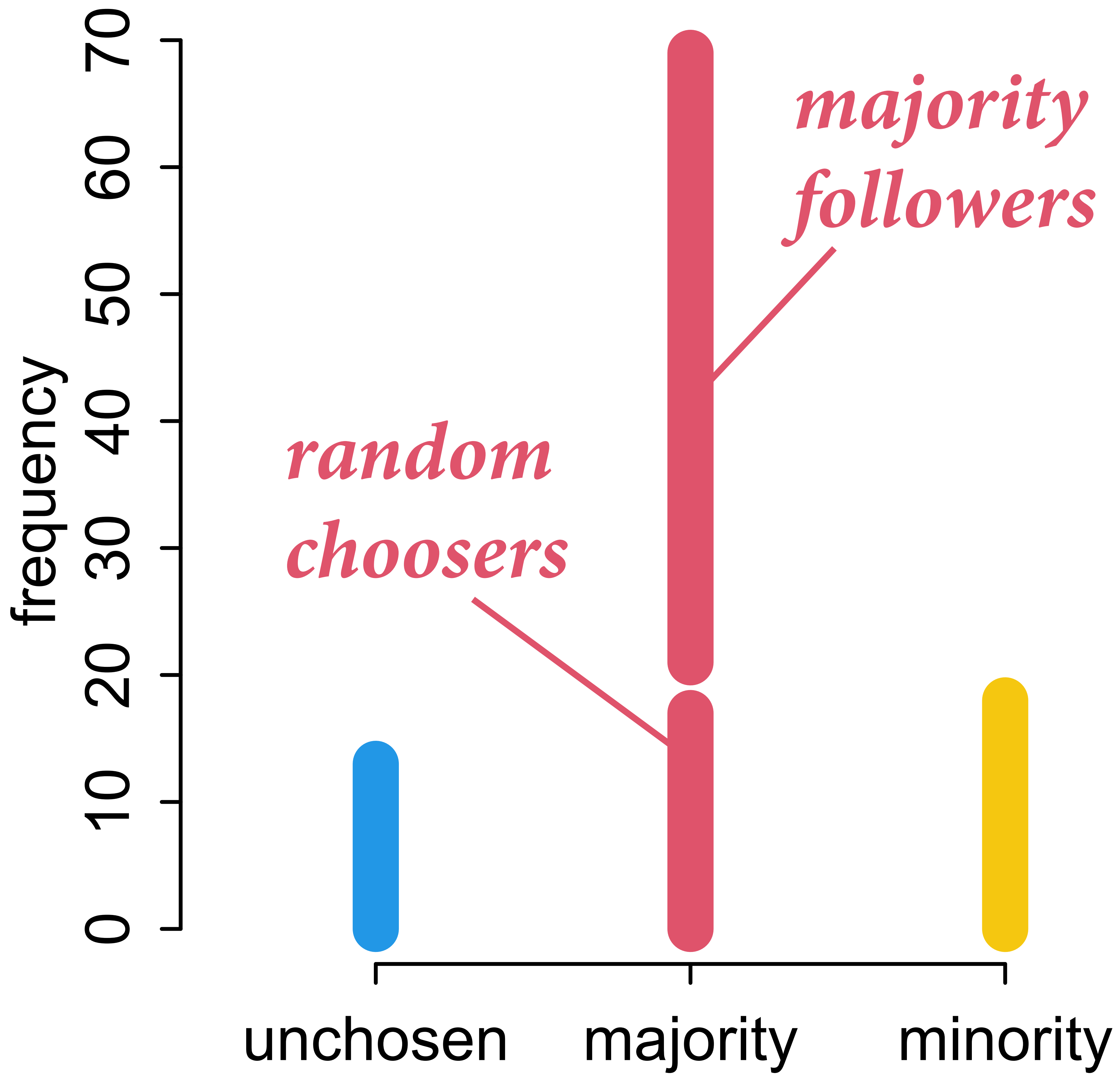
Random color: Choose majority 1/3 of time

Random demonstrator: 3/4 of time

Random demonstration: 1/2 of time



```r
N <- 100 # number of children

# half choose random color
# sample from 1,2,3 at random for each
y1 <- sample( 1:3 , size=N/2 , replace=TRUE )

# half follow majority
y2 <- rep( 2 , N/2 )

# combine and shuffle y1 and y2
y <- sample( c(y1,y2) )
```

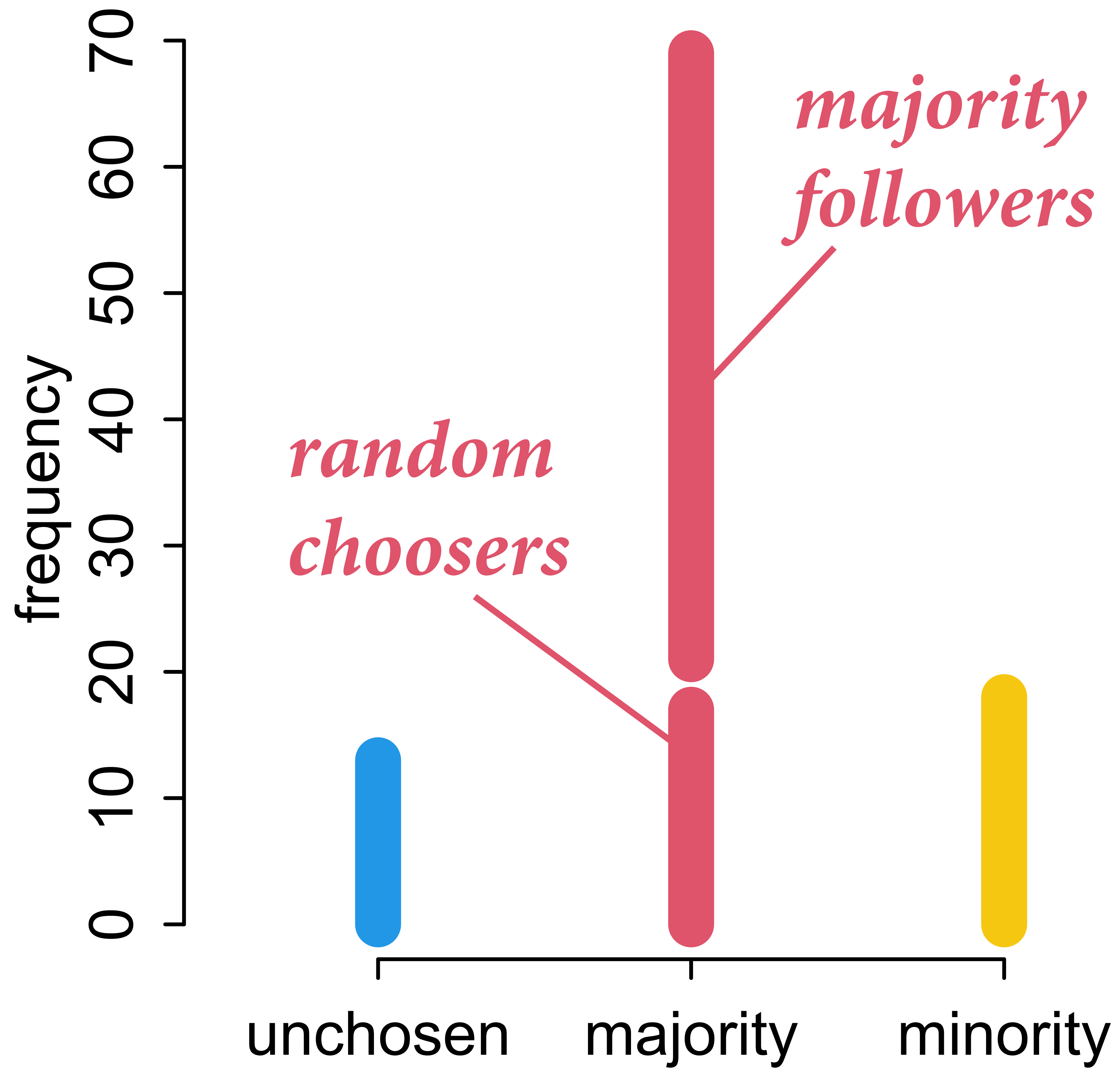


State-Based Model

Majority choice does not indicate majority preference

Instead infer the unobserved strategy (state) of each child

Strategy space:
(1) **Majority** (2) **Minority**
(3) **Maverick** (4) Random Color
(5) Follow First



$$Y_i \sim \text{Categorical}(\theta)$$

vector with probability of each choice

Probability of (1) unchosen, (2) majority, (3) minority

$$Y_i \sim \text{Categorical}(\theta)$$

Probability of (1) unchosen, (2) majority, (3) minority

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j \mid S)$$

Probability of choice j

average over strategies

prior probability strategy S

probability choice j assuming strategy S

$$Y_i \sim \text{Categorical}(\theta)$$

Probability of (1) unchosen, (2) majority, (3) minority

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j \mid S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

Prior for strategy space

```
data{
    int N;
    int y[N];
    int majority_first[N];
}
parameters{
    simplex[5] p;
}
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
data{
    int N;
    int y[N];
    int majority_first[N];
}
parameters{
    simplex[5] p;
}
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
data{
    int N;
    int y[N];
    int majority_first[N];
}
parameters{
    simplex[5] p;
}
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;

        // compute log( p_S * Pr(y_i|S) )
        for ( S in 1:5 )
            theta_j[S] = log(p[S]) + log(theta_j[S]);

        // compute average log-probability of y_i
        target += log_sum_exp( theta_j );
    }
}
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;

        // compute log( p_S * Pr(y_i|S) )
        for ( S in 1:5 )
            theta_j[S] = log(p[S]) + log(theta_j[S]);

        // compute average log-probability of y_i
        target += log_sum_exp( theta_j );
    }
}
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;

        // compute log( p_S * Pr(y_i|S) )
        for ( S in 1:5 )
            theta_j[S] = log(p[S]) + log(theta_j[S]);

        // compute average log-probability of y_i
        target += log_sum_exp( theta_j );
    }
}
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;

        // compute log( p_S * Pr(y_i|S) )
        for ( S in 1:5 )
            theta_j[S] = log(p[S]) + log(theta_j[S]);

        // compute average log-probability of y_i
        target += log_sum_exp( theta_j );
    }
}
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;

        // compute log( p_S * Pr(y_i|S) )
        for ( S in 1:5 )
            theta_j[S] = log(p[S]) + log(theta_j[S]);

        // compute average log-probability of y_i
        target += log_sum_exp( theta_j );
    }
}
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j \mid S)$$

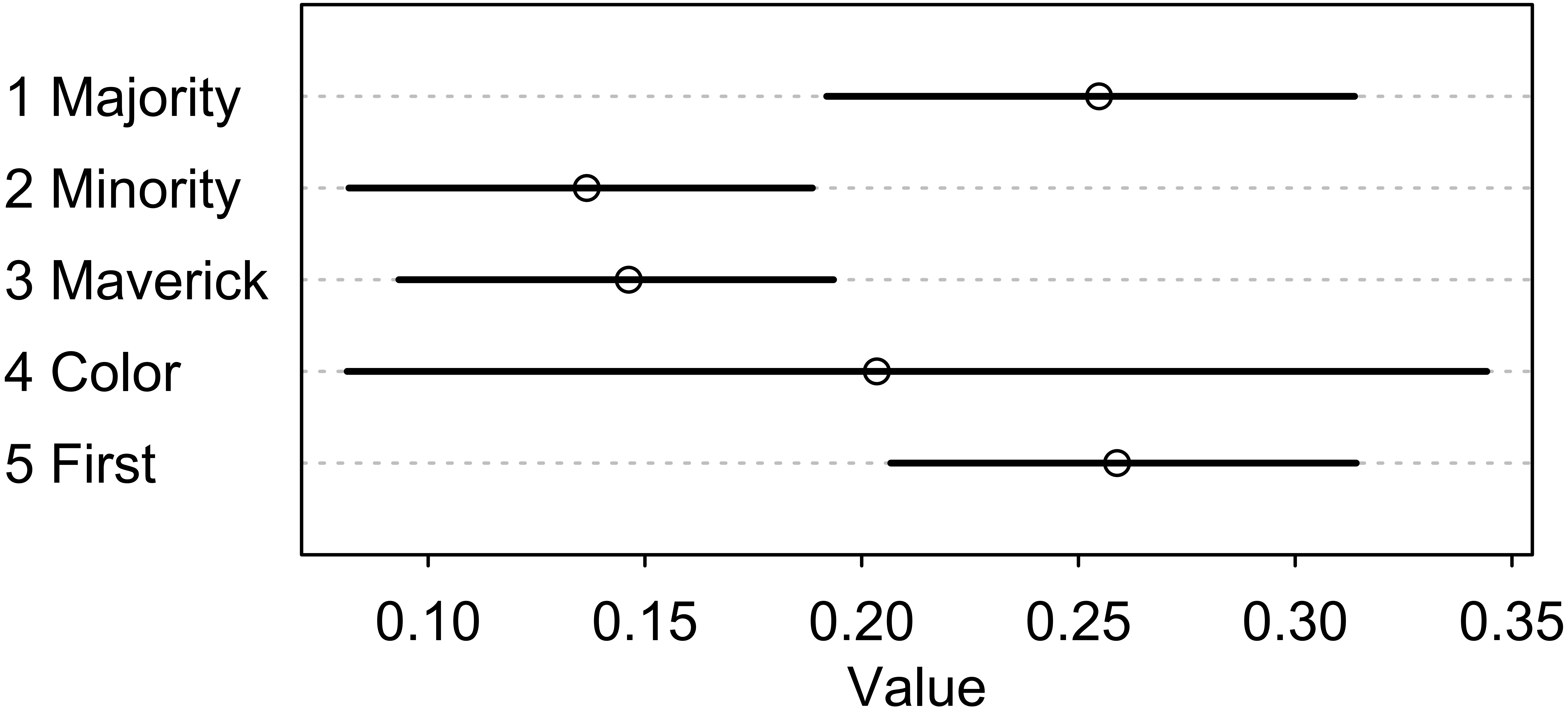
$$p \sim \text{Dirichlet}([4,4,4,4,4])$$

```
model{
    vector[5] theta_j;

    // prior
    p ~ dirichlet( rep_vector(4,5) );

    // probability of data
    for ( i in 1:N ) {
        theta_j = rep_vector(0,5); // clear it out
        if ( y[i]==2 ) theta_j[1]=1; // majority
        if ( y[i]==3 ) theta_j[2]=1; // minority
        if ( y[i]==1 ) theta_j[3]=1; // maverick
        theta_j[4]=1.0/3.0;          // random color
        if ( majority_first[i]==1 )  // follow first
            if ( y[i]==2 ) theta_j[5]=1;
        else
            if ( y[i]==3 ) theta_j[5]=1;

        // compute log( p_S * Pr(y_i|S) )
        for ( S in 1:5 )
            theta_j[S] = log(p[S]) + log(theta_j[S]);

        // compute average log-probability of y_i
        target += log_sum_exp( theta_j );
    }
}
```

$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j | S)$$

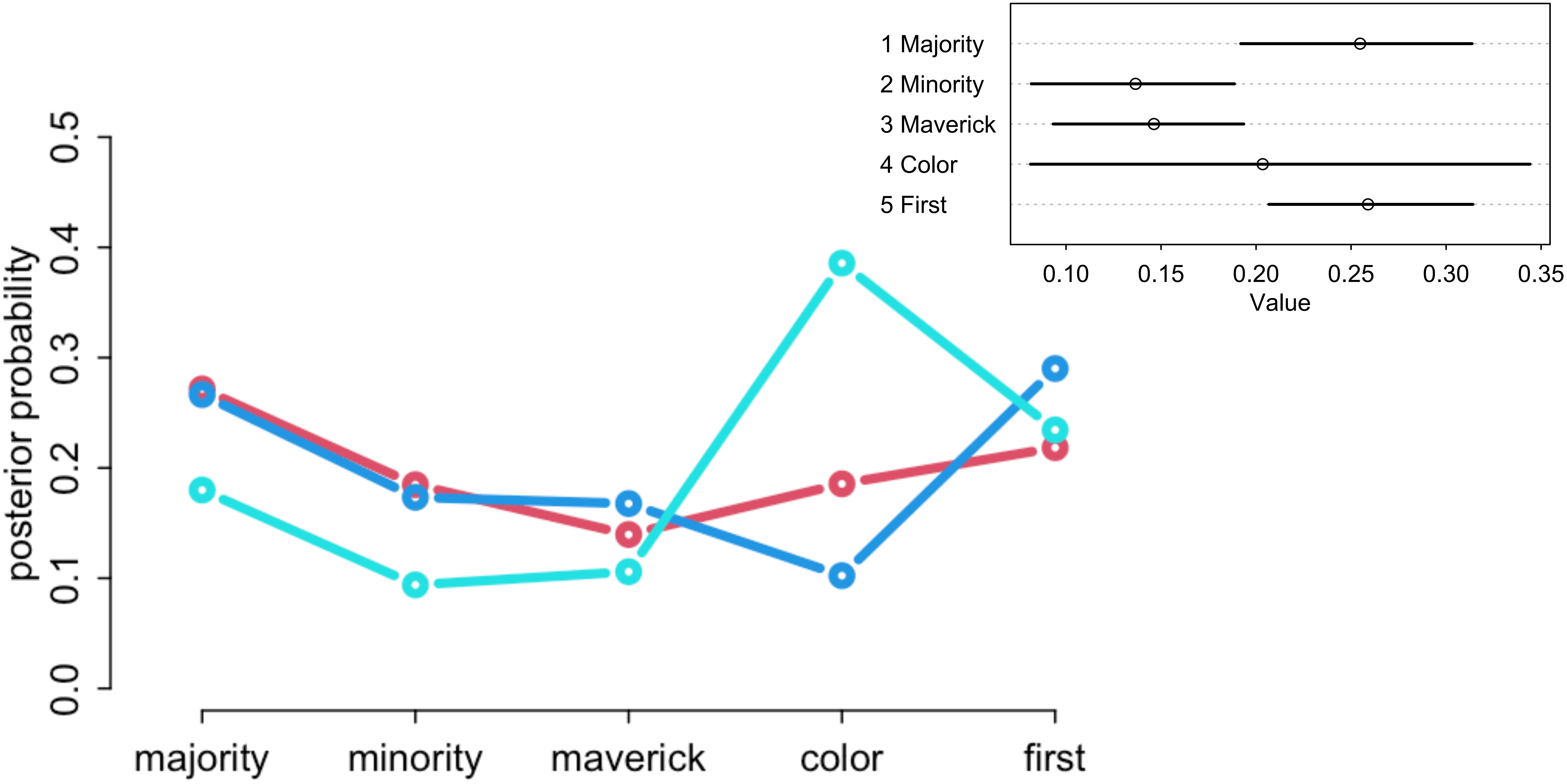
$$p \sim \text{Dirichlet}([4,4,4,4,4])$$



$$Y_i \sim \text{Categorical}(\theta)$$

$$\theta_j = \sum_{S=1}^{5} p_S \Pr(Y = j \,|\, S)$$

$$p \sim \text{Dirichlet}([4,4,4,4,4])$$



1 Majority
2 Minority
3 Maverick
4 Color
5 First
0.10
0.15
0.20
0.25
0.30
0.35
Value
posterior probability
0.0
0.1
0.2
0.3
0.4
0.5
majority
minority
maverick
color
first

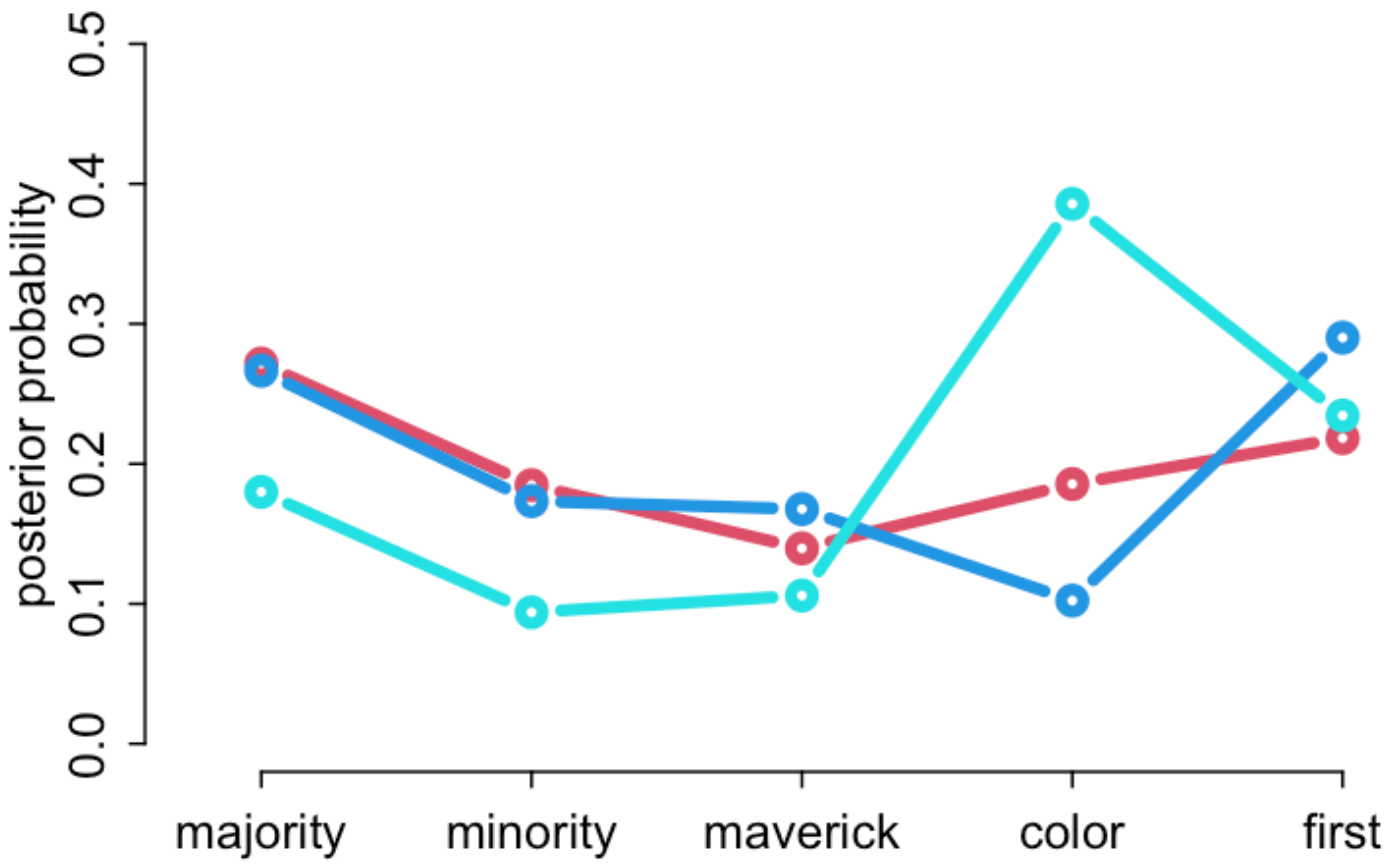
State-Based Models

What we want: Latent states

What we have: Emissions

Typically lots of uncertainty, but being honest is only ethical choice

Large family: Movement, learning, population dynamics, international relations, family planning, …



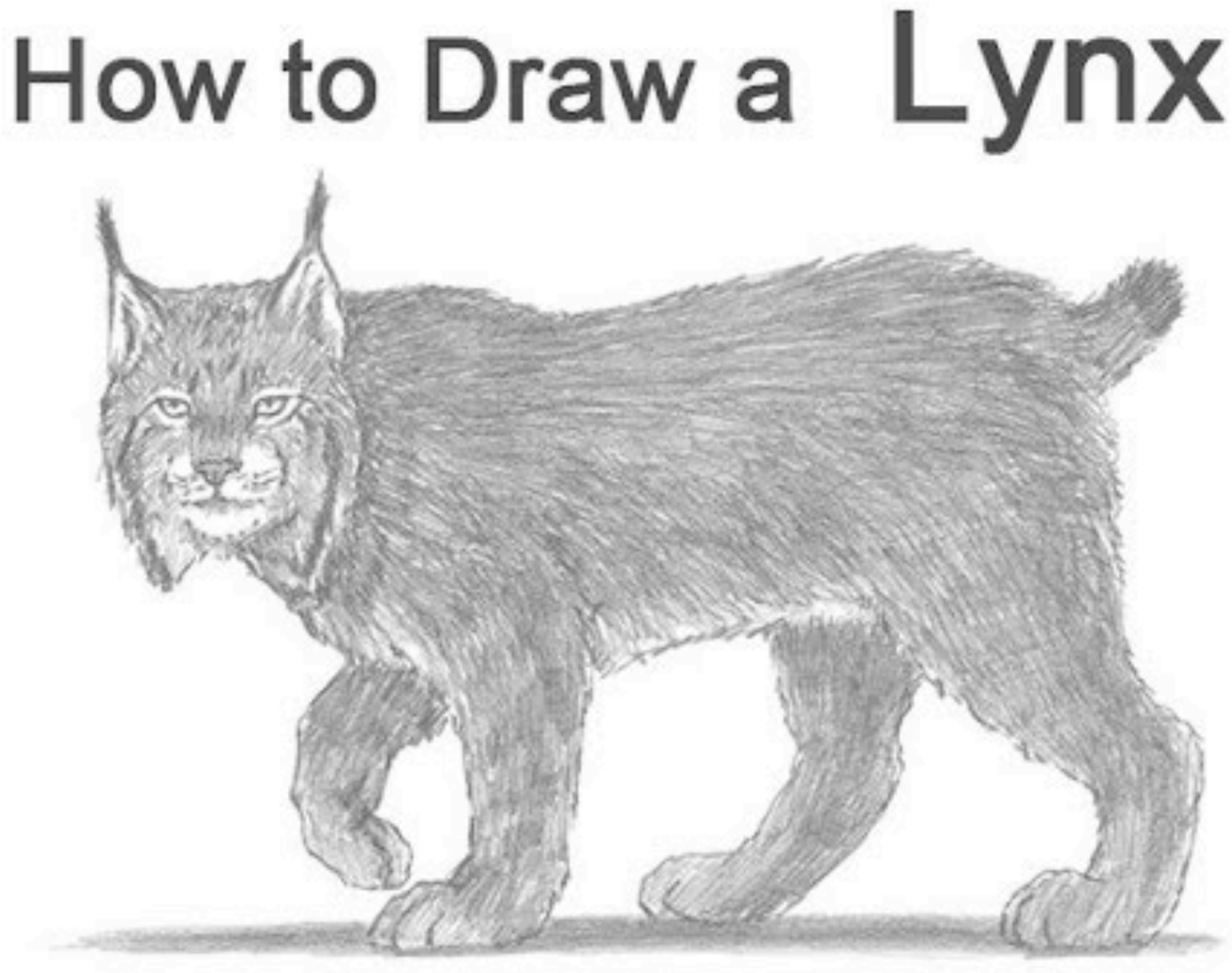
PAUSE

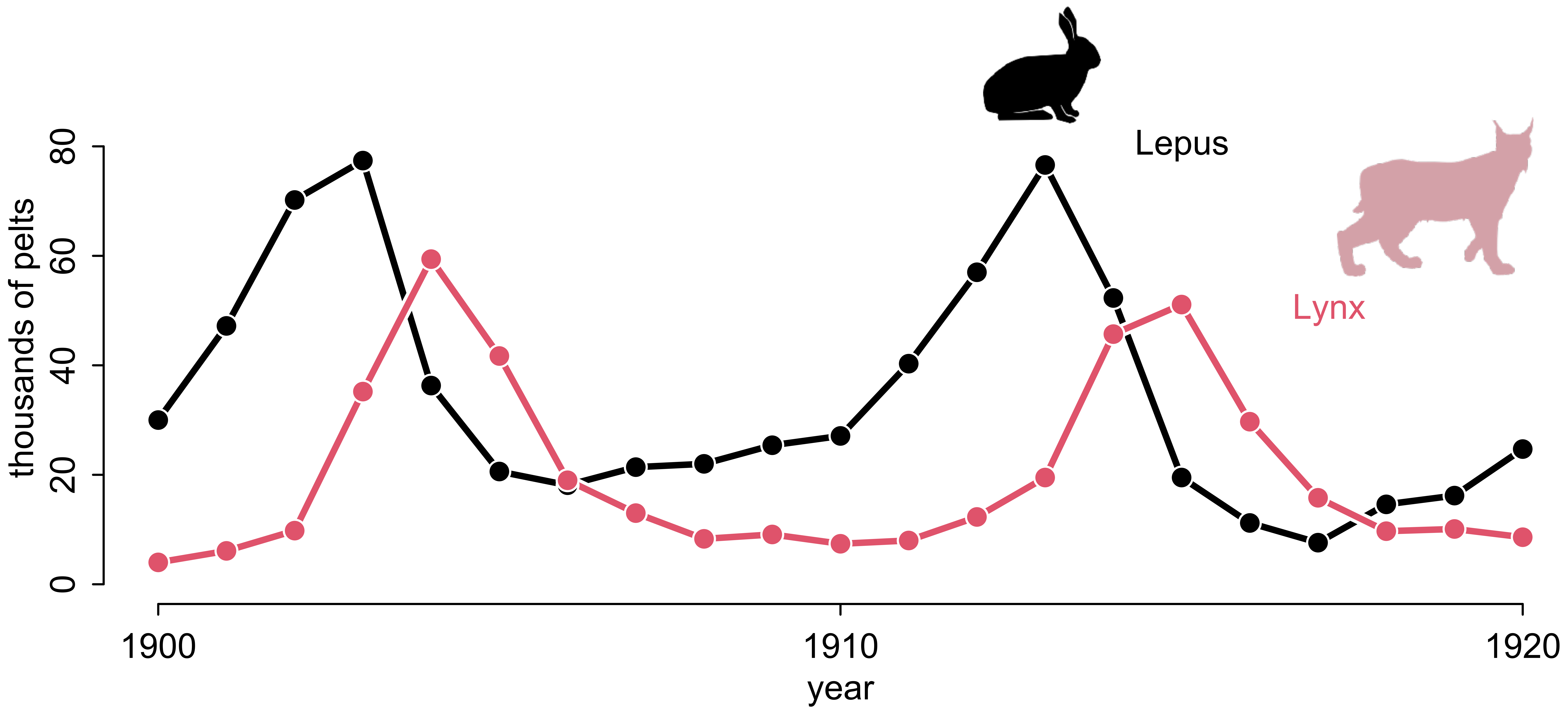
Population Dynamics

Latent states can be time varying

Example: Ecological dynamics, numbers of different species over time

Estimand: How do different species interact; how do interactions influence population dynamics

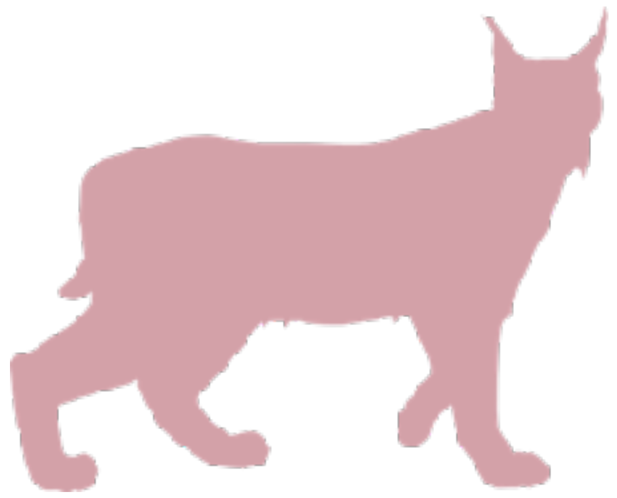




Lepus
Lynx
thousands of pels
0
20
40
60
80
1900
1910
1920
year

$$\frac{dH}{dt} = H_t \times (\text{birth rate}) - H_t \times (\text{death rate})$$

$$\frac{dL}{dt} = L_t \times (\text{birth rate}) - L_t \times (\text{death rate})$$



$$\frac{dH}{dt} = H_t b_H - H_t(L_t m_H)$$

birth rate of hares

impact of lynx on hares

$$\frac{dL}{dt} = L_t \times (\text{birth rate}) - L_t \times (\text{death rate})$$

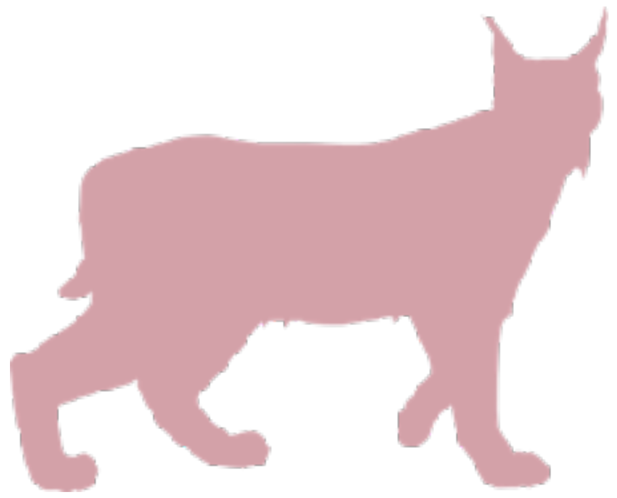
$$\frac{dH}{dt} = H_t b_H - H_t(L_t m_H)$$

birth rate of hares

impact of lynx on hares

$$\frac{dL}{dt} = L_t(H_t b_L) - L_t m_L$$

birth rate of lynx depends upon hares



$$h_t \sim \text{LogNormal}\big(\log(p_H H_t), \sigma_H\big)$$

$$l_t \sim \text{LogNormal}\big(\log(p_L L_t), \sigma_L\big)$$

$$\frac{dH}{dt} = H_t b_H - H_t(L_t m_H)$$

$$\frac{dL}{dt} = L_t(H_t b_L) - L_t m_L$$

$$H_T = H_1 + \int_1^T \frac{dH}{dt} dt$$

$$L_T = L_1 + \int_1^T \frac{dL}{dt} dt$$

observed hare pelts

observed lynx pelts

$$h_t \sim \text{LogNormal}\big(\log(p_H H_t), \sigma_H\big)$$

$$l_t \sim \text{LogNormal}\big(\log(p_L L_t), \sigma_L\big)$$

$$\frac{dH}{dt} = H_t b_H - H_t(L_t m_H)$$

$$\frac{dL}{dt} = L_t(H_t b_L) - L_t m_L$$

$$H_T = H_1 + \int_1^T \frac{dH}{dt} dt$$

$$L_T = L_1 + \int_1^T \frac{dL}{dt} dt$$

observed hare pelts

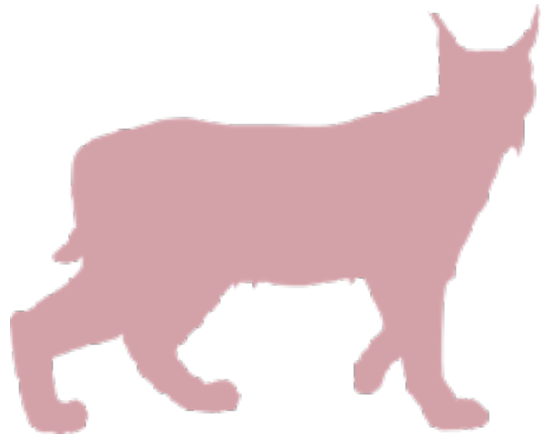
$$h_t \sim \text{LogNormal}\big(\log(p_H H_t), \sigma_H\big)$$

$$\frac{dH}{dt} = H_t b_H - H_t(L_t m_H)$$

$$H_T = H_1 + \underbrace{\int_1^T \frac{dH}{dt} dt}$$

cumulative changes in H until time T

observed lynx pelts



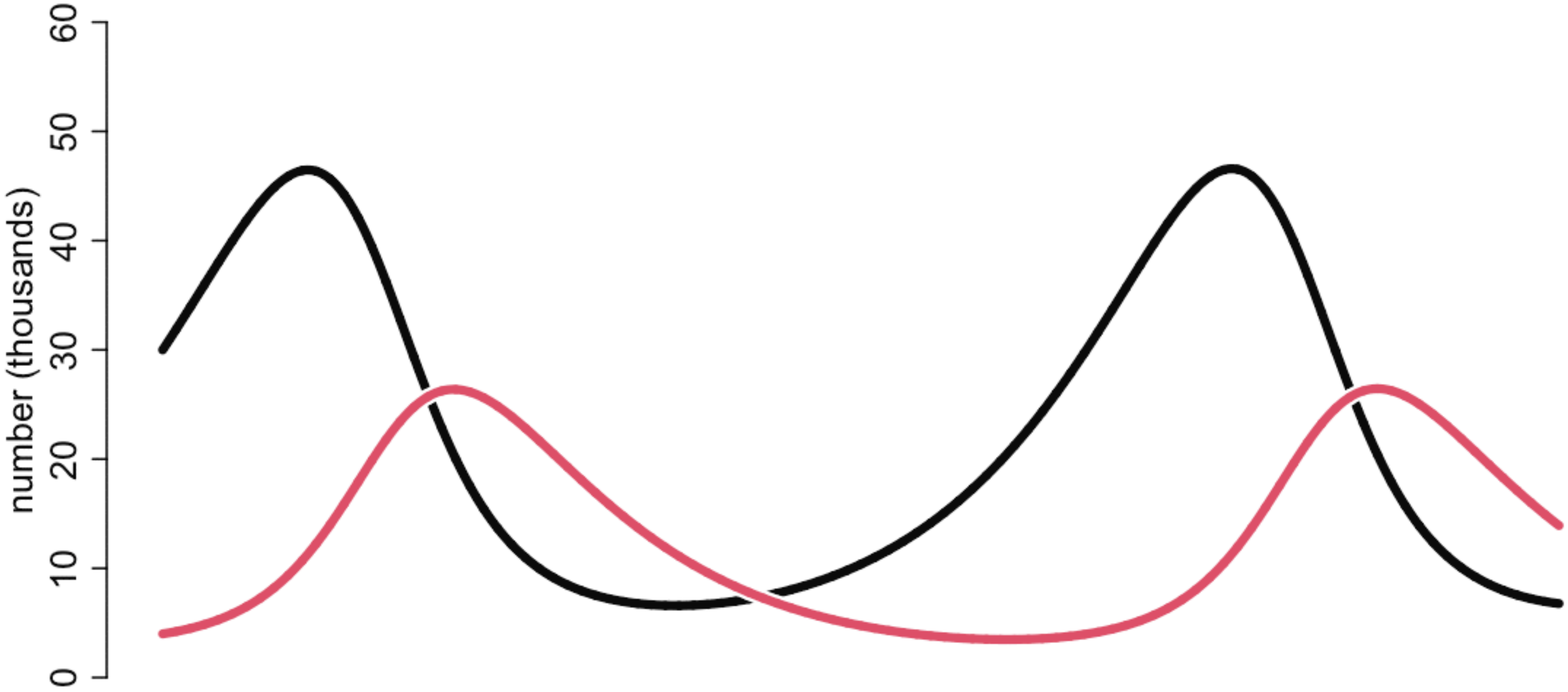
$$l_t \sim \text{LogNormal}\big(\log(p_L L_t), \sigma_L\big)$$

$$\frac{dL}{dt} = L_t(H_t b_L) - L_t m_L$$

$$L_T = L_1 + \underbrace{\int_1^T \frac{dL}{dt} dt}$$

cumulative changes in L until time T

Prior Simulation



```
functions {
  real[] dpop_dt( real t,                        // time
                  real[] pop_init,               // initial state {lynx, hares}
                  real[] theta,                  // parameters
                  real[] x_r, int[] x_i) {       // unused
    real L = pop_init[1];
    real H = pop_init[2];
    real bh = theta[1];
    real mh = theta[2];
    real ml = theta[3];
    real bl = theta[4];
    // differential equations
    real dH_dt = (bh - mh * L) * H;
    real dL_dt = (bl * H - ml) * L;
    return { dL_dt , dH_dt };
  }
}
data {
  int<lower=0> N;               // number of measurement times
  real<lower=0> pelts[N,2];     // measured populations
}
transformed data{
  real times_measured[N-1];     // N-1 because first time is initial state
  for ( i in 2:N ) times_measured[i-1] = i;
}
parameters {
```

```
functions {
  real[] dpop_dt( real t,                    // time
                  real[] pop_init,           // initial state {lynx, hares}
                  real[] theta,              // parameters
                  real[] x_r, int[] x_i) {  // unused
    real L = pop_init[1];
    real H = pop_init[2];
    real bh = theta[1];
    real mh = theta[2];
    real ml = theta[3];
    real bl = theta[4];
    // differential equations
    real dH_dt = (bh - mh * L) * H;
    real dL_dt = (bl * H - ml) * L;
    return { dL_dt , dH_dt };
  }
}
data {
  int<lower=0> N;              // number of measurement times
  real<lower=0> pelts[N,2];    // measured populations
}
transformed data{
  real times_measured[N-1];    // N-1 because first time is initial state
  for ( i in 2:N ) times_measured[i-1] = i;
}
parameters {
```

Computes cumulative change to time t

```
functions {
  real[] dpop_dt( real t,                 // time
                  real[] pop_init,        // initial state {lynx, hares}
                  real[] theta,           // parameters
                  real[] x_r, int[] x_i) { // unused
    real L = pop_init[1];
    real H = pop_init[2];
    real bh = theta[1];
    real mh = theta[2];
    real ml = theta[3];
    real bl = theta[4];
    // differential equations
    real dH_dt = (bh - mh * L) * H;
    real dL_dt = (bl * H - ml) * L;
    return { dL_dt , dH_dt };
  }
}
data {
  int<lower=0> N;              // number of measurement times
  real<lower=0> pelts[N,2];    // measured populations
}
transformed data{
  real times_measured[N-1];    // N-1 because first time is initial state
  for ( i in 2:N ) times_measured[i-1] = i;
}
parameters {
```

Computes cumulative change to time t

$$\frac{dH}{dt} = H_t b_H - H_t(L_t m_H)$$

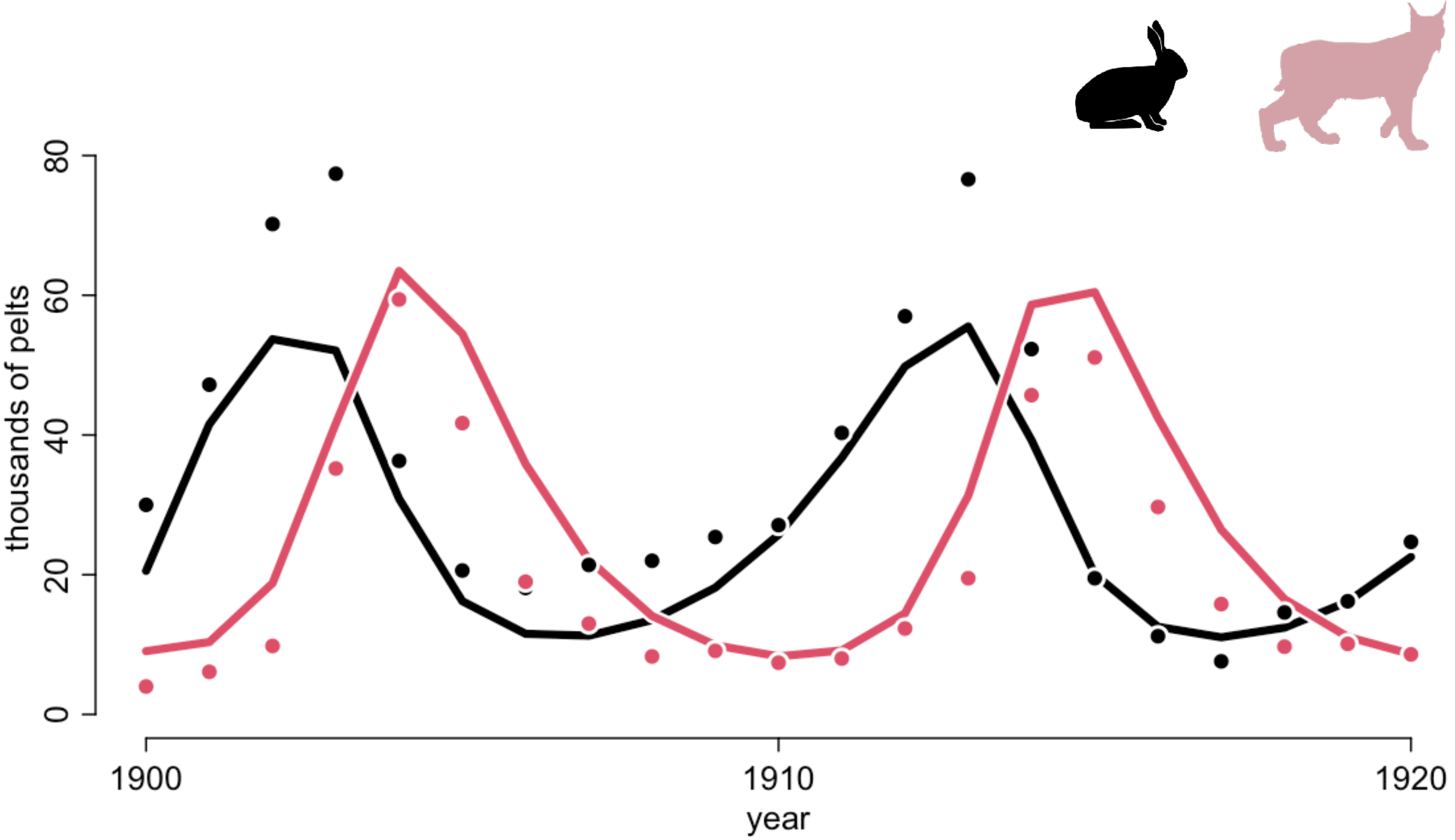
$$\frac{dL}{dt} = L_t(H_t b_L) - L_t m_L$$

```
}
parameters {
  real<lower=0> theta[4];      // { bh, mh, ml, bl }
  real<lower=0> pop_init[2];   // initial population state
  real<lower=0> sigma[2];      // measurement errors
  real<lower=0,upper=1> p[2];  // trap rate
}
transformed parameters {
  real pop[N, 2];
  pop[1,1] = pop_init[1];
  pop[1,2] = pop_init[2];
  pop[2:N,1:2] = integrate_ode_rk45(
    dpop_dt, pop_init, 0, times_measured, theta,
    rep_array(0.0, 0), rep_array(0, 0),
    1e-5, 1e-3, 5e2);
}
model {
  // priors
  theta[{1,3}] ~ normal( 1 , 0.5 );    // bh,ml
  theta[{2,4}] ~ normal( 0.05, 0.05 ); // mh,bl
  sigma ~ exponential( 1 );
  pop_init ~ lognormal( log(10) , 1 );
  p ~ beta(40,200);
  // observation model
  // connect latent population state to observed pelts
  for ( t in 1:N )
    for ( k in 1:2 )
      pelts[t,k] ~ lognormal( log(pop[t,k]*p[k]) , sigma[k] );
```

Compute population state for each time

```
  pop[1,1] = pop_init[1];
  pop[1,2] = pop_init[2];
  pop[2:N,1:2] = integrate_ode_rk45(
    dpop_dt, pop_init, 0, times_measured, theta,
    rep_array(0.0, 0), rep_array(0, 0),
    1e-5, 1e-3, 5e2);
}
model {
  // priors
  theta[{1,3}] ~ normal( 1 , 0.5 );     // bh,ml
  theta[{2,4}] ~ normal( 0.05, 0.05 ); // mh,bl
  sigma ~ exponential( 1 );
  pop_init ~ lognormal( log(10) , 1 );
  p ~ beta(40,200);
  // observation model
  // connect latent population state to observed pelts
  for ( t in 1:N )
    for ( k in 1:2 )
      pelts[t,k] ~ lognormal( log(pop[t,k]*p[k]) , sigma[k] );
}
generated quantities {
  real pelts_pred[N,2];
  for ( t in 1:N )
    for ( k in 1:2 )
      pelts_pred[t,k] = lognormal_rng( log(pop[t,k]*p[k]) , sigma[k] );
}
```

Probability of data, given latent population



thousands of pelts
0
20
40
60
80
1900
1910
1920
year

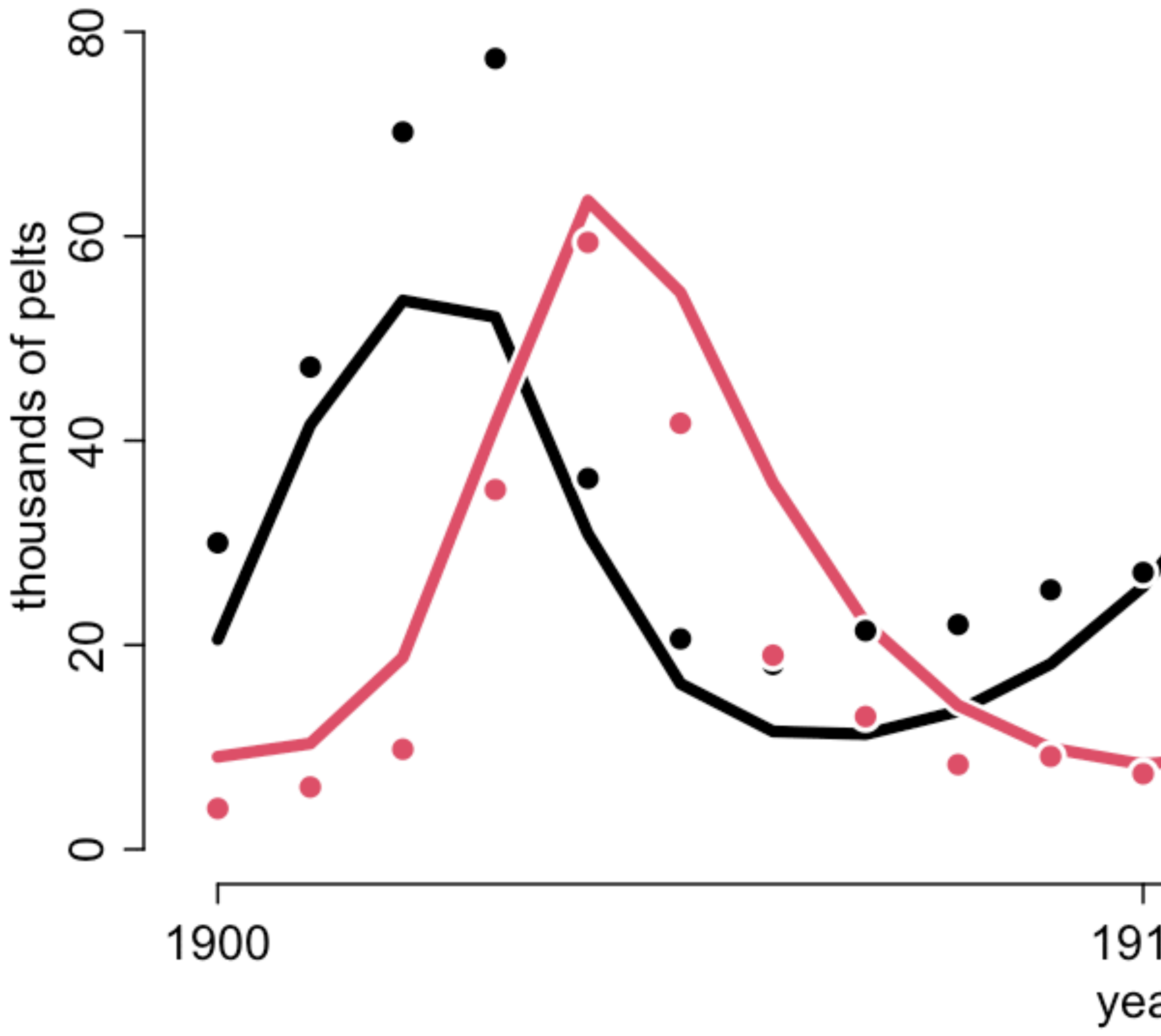
Population Dynamics

Ecologies much more complex

Other animals prey on hare

Without causal model, little hope to understand interventions

Same framework very successful in fisheries management



Science Before Statistics

Epicycles get you only so far

Scientific models also flawed, but flaws are more productive

Theory necessary for empiricism

Be patient; mastery takes time; experts learn safe habits

Student learning differential equations

Course Schedule

Week 1	Bayesian inference	Chapters 1, 2, 3
Week 2	Linear models & Causal Inference	Chapter 4
Week 3	Causes, Confounds & Colliders	Chapters 5 & 6
Week 4	Overfitting / MCMC	Chapters 7, 8, 9
Week 5	Generalized Linear Models	Chapters 10, 11
Week 6	Ordered categories & Multilevel models	Chapters 12 & 13
Week 7	More Multilevel models	Chapters 13 & 14
Week 8	Social Networks & Gaussian Processes	Chapter 14
Week 9	Measurement & Missingness	Chapter 15
Week 10	Generalized Linear Madness	Chapter 16

https://github.com/rmcelreath/stat_rethinking_2023