

Statistical Rethinking

Winter 2019

Lecture 13 / Week 7

Monsters & Mixtures

How to draw an owl

1.



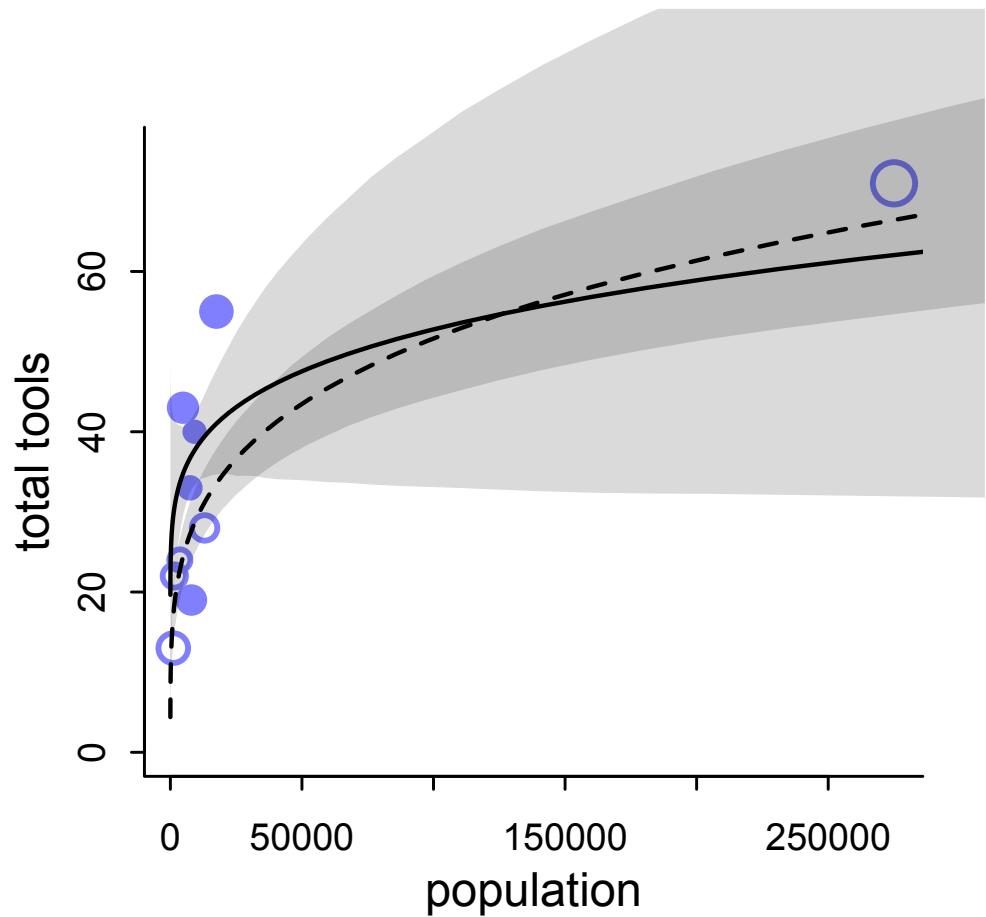
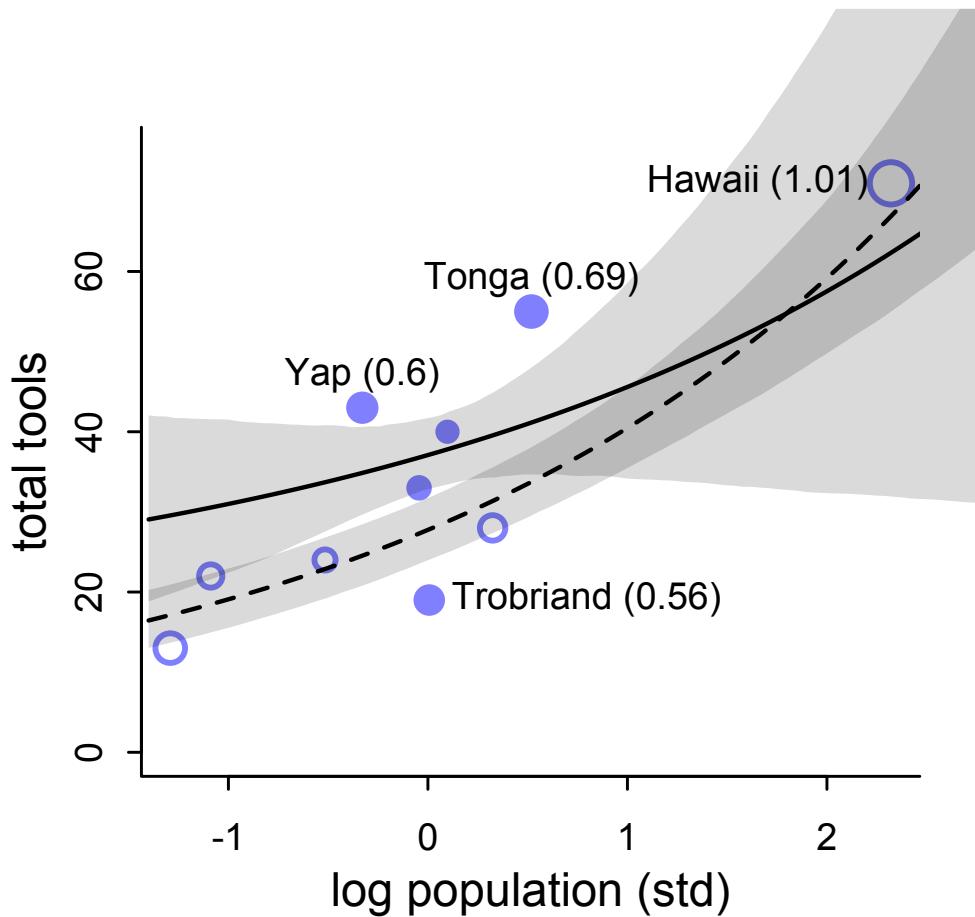
2.



1. Draw some circles

2. Draw the rest of the owl

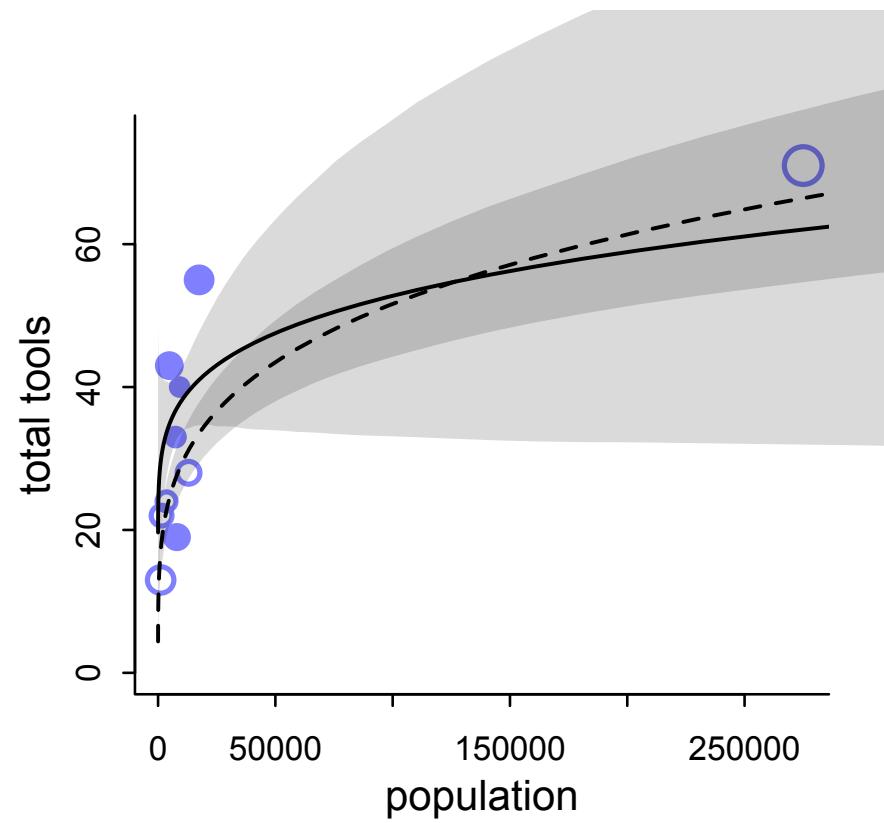
Hawaii has leverage



- Point size proportional to Pareto- k diagnostic value

Generalized Linear Madness

- This model is terrible:
- Intercepts don't pass through origin
- Zero population = zero tools
- We can do better by thinking *scientifically* instead of *statistically*



Scientific model

- Change in tools per unit time:

$$\Delta T =$$

Scientific model

- Change in tools per unit time:

Diminishing returns
("elasticity")

Innovation rate

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

Population

The diagram illustrates a scientific model. At the top right, the text "Diminishing returns ("elasticity")" is displayed. Below it, the term "Innovation rate" is written. A dashed blue line connects this text to the left side of the central equation. The central equation is $\Delta T = \alpha P^\beta$. To the left of the equation, the word "Population" is written, and a dashed blue line connects it to the right side of the equation. The entire equation is centered under the heading "Diminishing returns ("elasticity")".

Scientific model

- Change in tools per unit time:

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

Diminishing returns
("elasticity")

The diagram illustrates the components of the equation $\Delta T = \alpha P^\beta - \gamma T$. It features a central equation with three dashed arrows pointing to its terms. The first arrow, from the left, points to the term αP^β and is labeled "Innovation rate". The second arrow, from the bottom, points to the term T and is labeled "Population". The third arrow, from the right, points to the term γT and is labeled "Loss rate".

Scientific model

- Solve for steady state expected number of tools
- Where $\Delta T = 0$

$$\hat{T} = \frac{\alpha P^\beta}{\gamma}$$

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

$$\lambda_i = \alpha P_i^\beta / \gamma$$

No *ad hoc* link function!

Scientific model

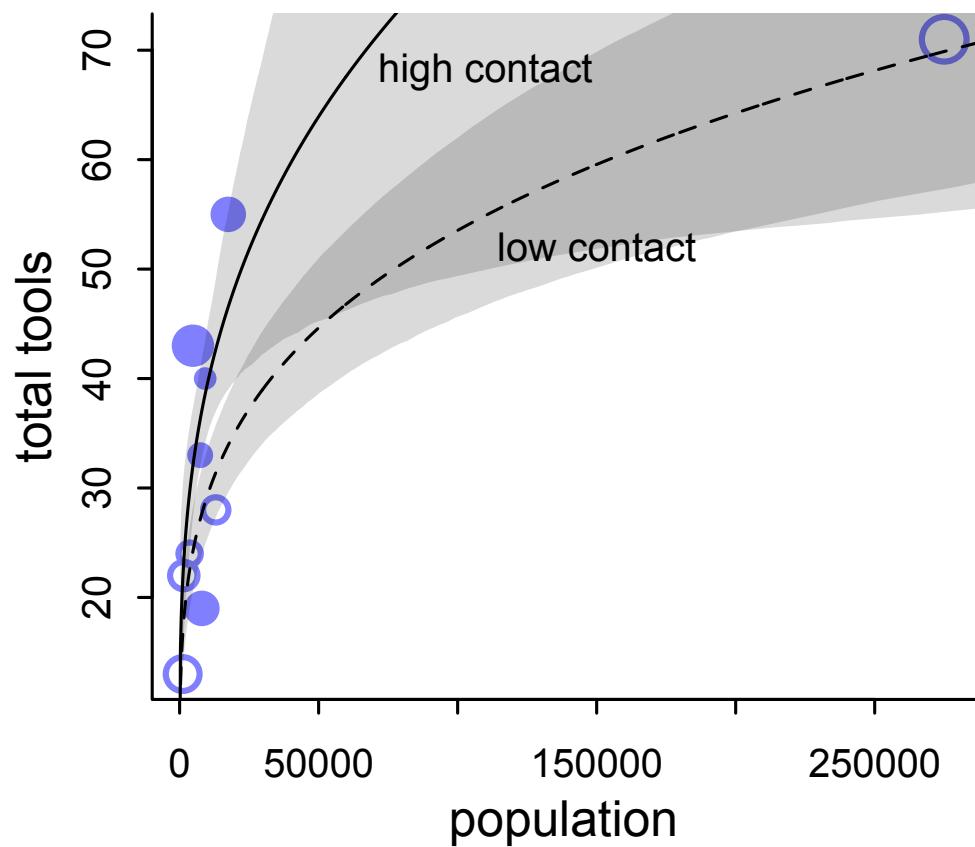
$$T_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = \alpha P_i^\beta / \gamma$$

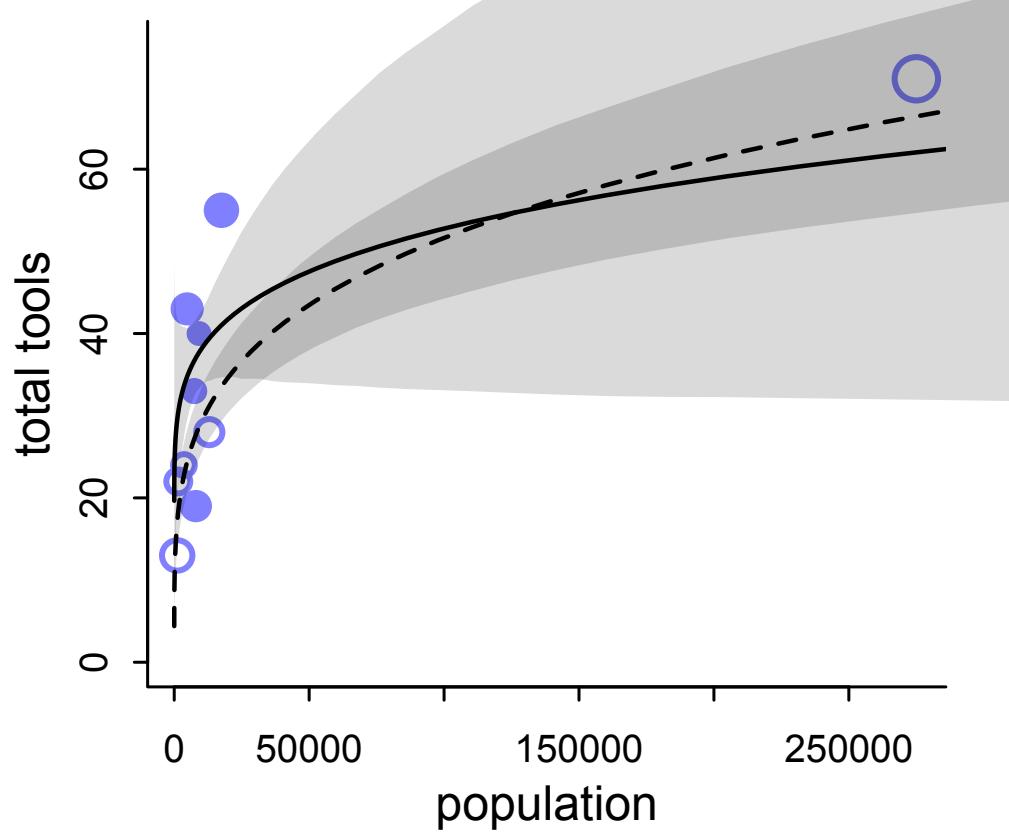
```
dat2 <- list( T=d$total_tools, P=d$population, cid=d$contact_id )
m11.11 <- ulam(
  alist(
    T ~ dpois( lambda ),
    lambda <- exp(a[cid])*P^b[cid]/g,
    a[cid] ~ dnorm(1,1),
    b[cid] ~ dexp(1),
    g ~ dexp(1)
  ), data=dat2 , chains=4 , log_lik=TRUE )
```

R code
11.52

Scientific model



Statistical model



Model violations now mean something.
Parameters now mean something.

Poisson exposure (offsets)

- Poisson outcome: events per unit time/distance
- Q: What if time/distance varies across cases?
- A: Use an *exposure*, aka *offset*

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

$$\log \lambda_i = \log \frac{\mu_i}{\tau_i} = \alpha + \beta x_i$$

exposure *expected count*

$$y_i \sim \text{Poisson}(\mu_i)$$

$$\log \mu_i = \log \tau_i + \alpha + \beta x_i$$

Additional count distributions

- Multinomial/categorical: generalized binomial, more than 2 un-ordered outcomes
- Geometric: number of trials until specific event
- Mixtures, coping with heterogeneity:
 - Beta-binomial: varying probabilities
 - gamma-Poisson: aka negative-Binomial, varying rates
 - others (e.g. Dirichlet-multinomial)

Survival Analysis

- Count models are fundamentally about rates
 - Rate of heads per coin toss
 - Rate of tools per person
- Can also estimate rates by modeling time-to-event
- Tricky, because cannot ignore *censored* cases
 - Left-censored: Don't know when time started
 - Right-censored: Something cut observation off before event occurred
- Ignoring censored cases leads to inferential error
 - Imagine estimating time-to-PhD but ignoring people who drop out
 - Time in program before dropping out is info about rate

Survival Analysis

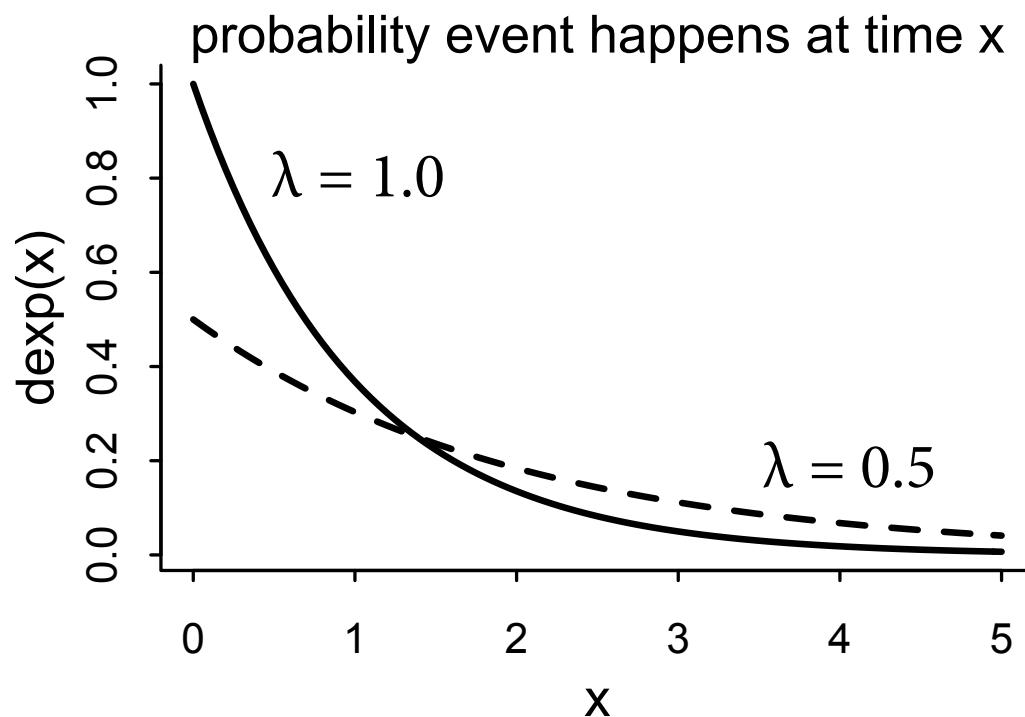
- Example: Cat adoptions
- `data(AustinCats)`
 - 20-thousand cats
 - time-to-event
 - Event either: (1) adopted or (2) something else
 - Something else could be: death, escape, **censored**



Un-censored observations

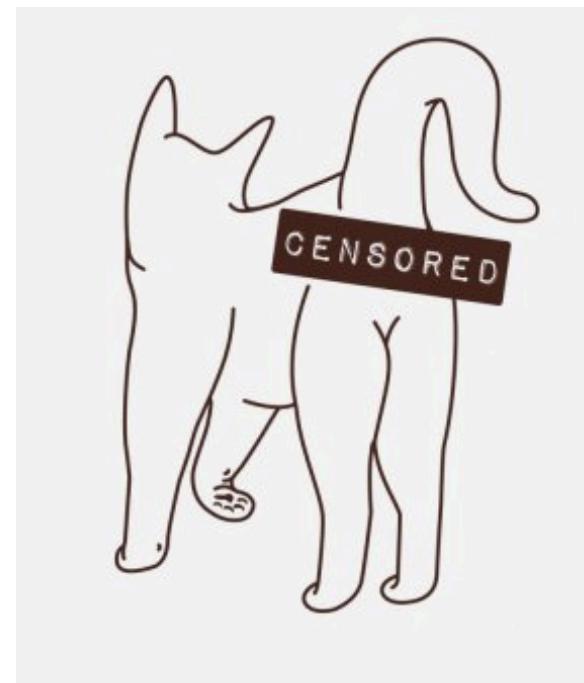
- For observed adoptions, just need:

$$D_i \sim \text{Exponential}(\lambda_i) \quad p(D_i | \lambda_i) = \lambda_i \exp(-\lambda_i D_i)$$

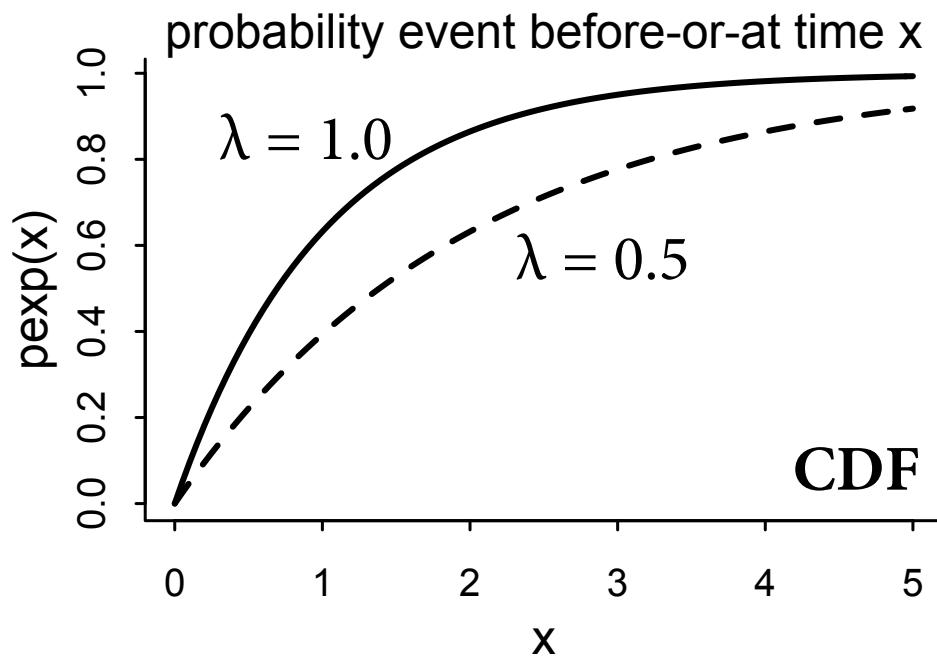


Censored cats

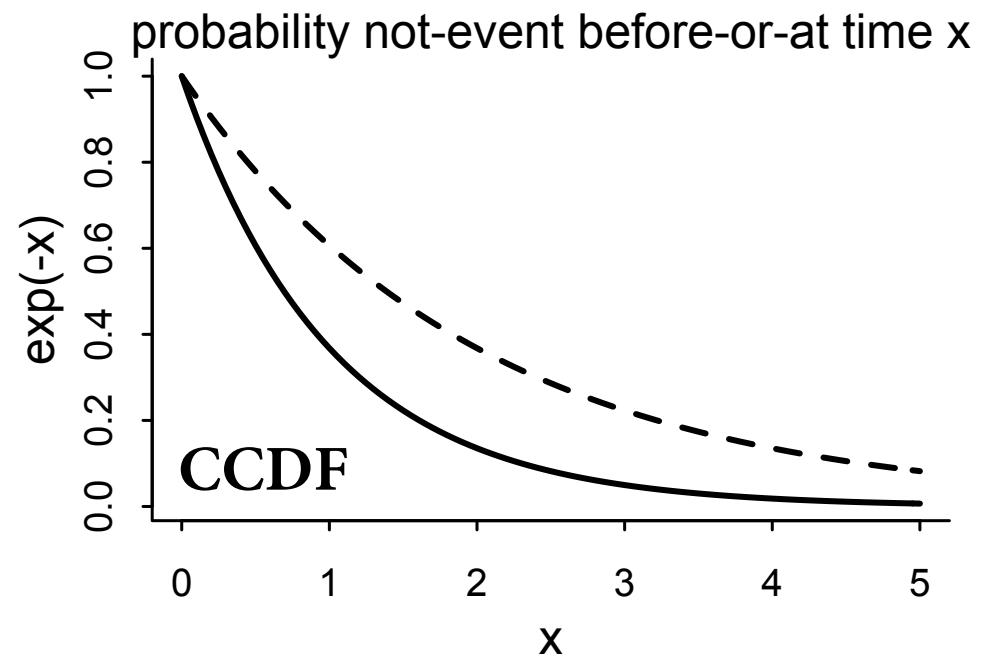
- Cumulative distribution (CDF): Probability event before-or-at time x
- Complementary cumulative distribution (CCDF): Probability not-event-yet



$$\Pr(D_i|\lambda_i) = 1 - \exp(-\lambda_i D_i)$$



$$\Pr(D_i|\lambda_i) = \exp(-\lambda_i D_i)$$



Cat code

$$D_i | A_i = 1 \sim \text{Exponential}(\lambda_i)$$

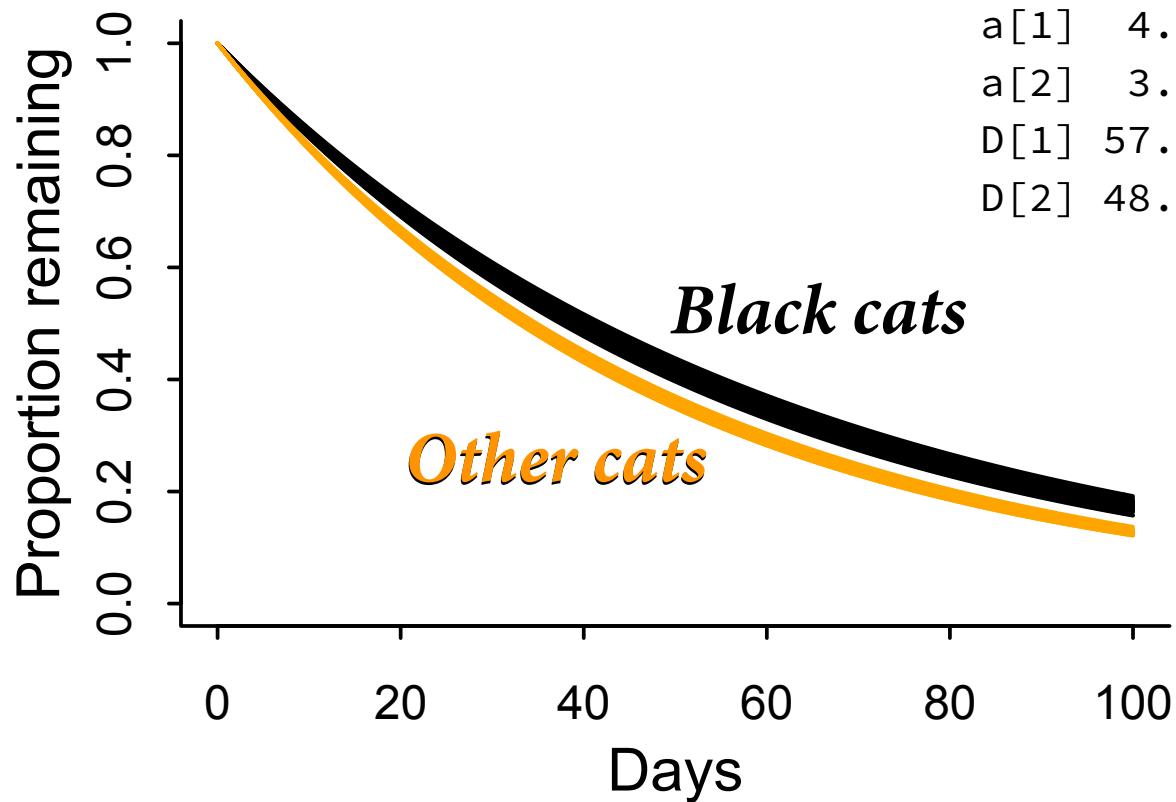
$$D_i | A_i = 0 \sim \text{Exponential-CCDF}(\lambda_i)$$

$$\lambda_i = 1/\mu_i$$

$$\log \mu_i = \alpha_{\text{CID}[i]}$$

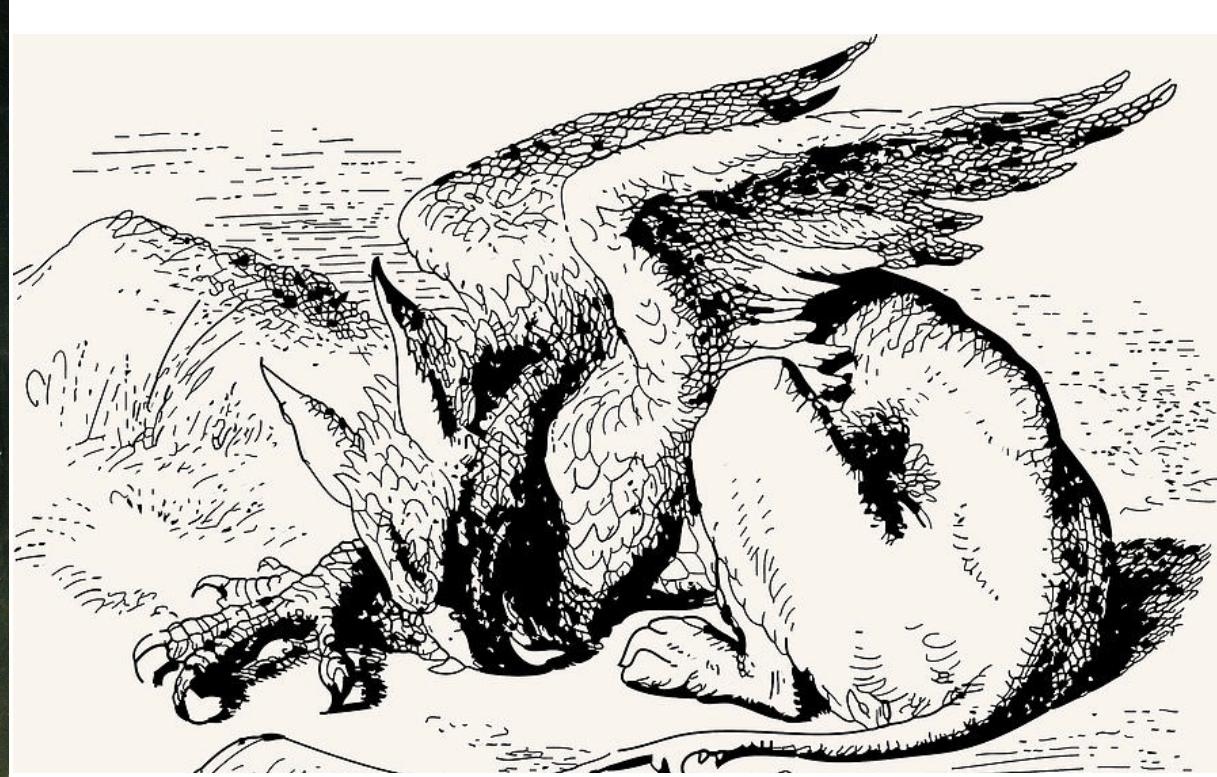
```
m11.14 <- ulam(  
  alist(  
    days_to_event|adopted==1 ~ exponential( lambda ),  
    days_to_event|adopted==0 ~ custom(exponential_lccdf( !Y | lambda )),  
    lambda <- 1.0/mu,  
    log(mu) <- a[color_id],  
    a[color_id] ~ normal(0,1)  
, data=dat , chains=4 , cores=4 )
```

Posterior survival curves



```
post <- extract.samples( m11.14 )
post$D <- exp(post$a)
precis( post , 2 )
```

```
'data.frame': 2000 obs. of 4 variables:
               mean     sd   5.5% 94.5% histogram
a[1]    4.05 0.03  4.01  4.09
a[2]    3.88 0.01  3.87  3.90
D[1]   57.44 1.47 55.11 59.77
D[2]   48.44 0.49 47.71 49.22
```





Do not attempt at home (or anywhere near me)

Monsters and mixtures

- More complicated GLMs:
 - *Monsters*: Specialized, complex distributions
 - ordered categories, ranks
 - *Mixtures*: Blends of stochastic processes
 - Varying means, probabilities, rates
 - Varying process: zero-inflation, hurdles



Mixtures

- Some outcomes mix different processes
 - replace parameter of likelihood with distribution of its own
- Over-dispersion: counts often more variable than expected, because probabilities/rates are variable
 - beta-binomial, gamma-Poisson (negative-binomial)
- Zero-inflated mixtures

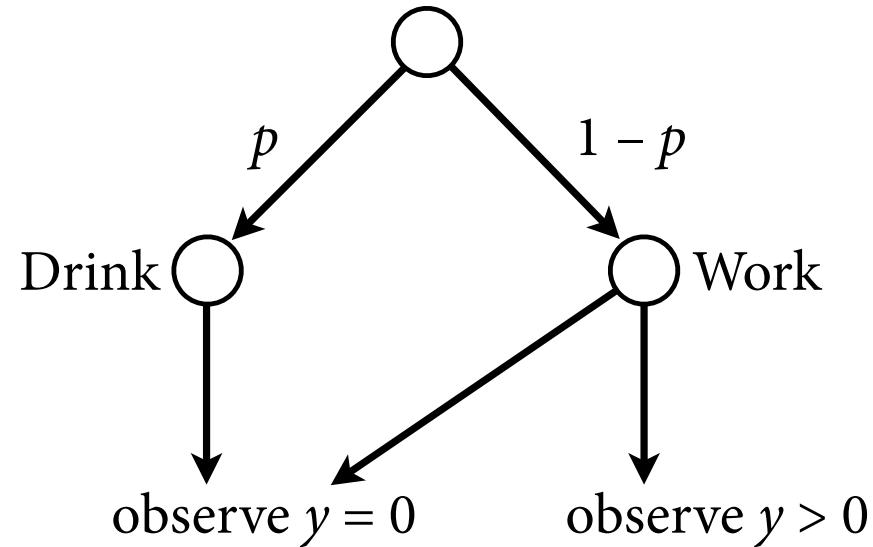
Monastery Mystery

- Monks copy manuscripts
- They also get drunk
- Data: num manuscripts completed each day
- Can infer number of days they got drunk?

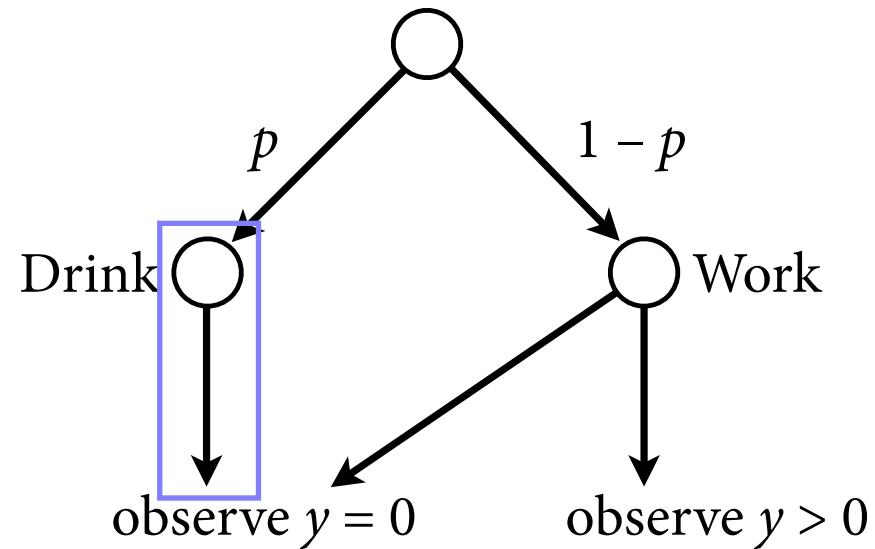
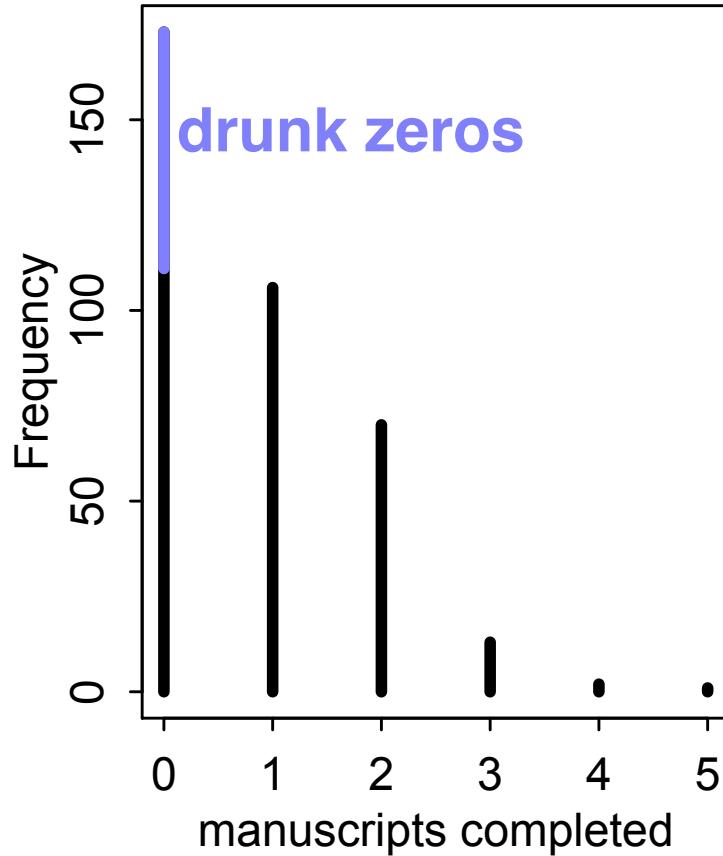


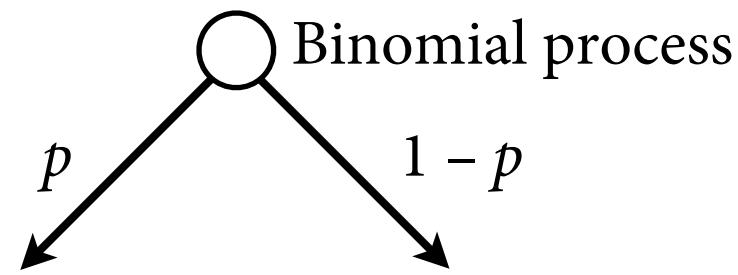
Analyze?

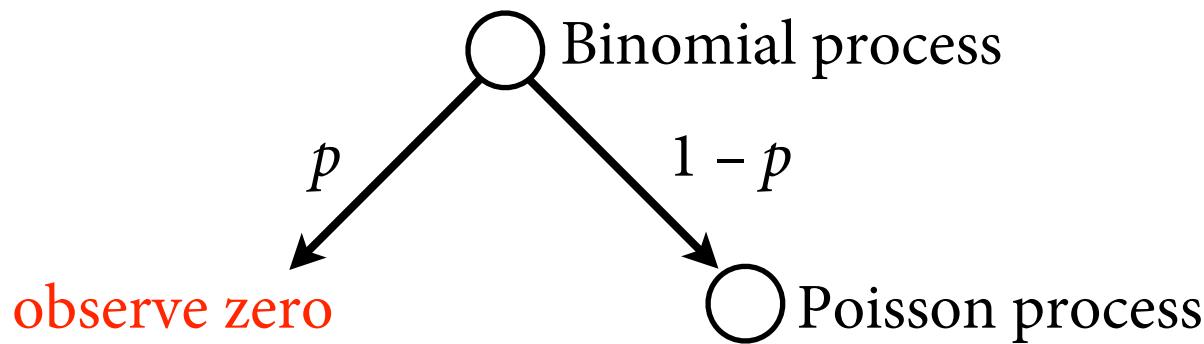
- Zero-inflated Poisson observations
- Hidden state: drunk or sober
- Can estimate probability of drinking and rate of production when sober
- Must build a new likelihood, a mixture of stochastic processes

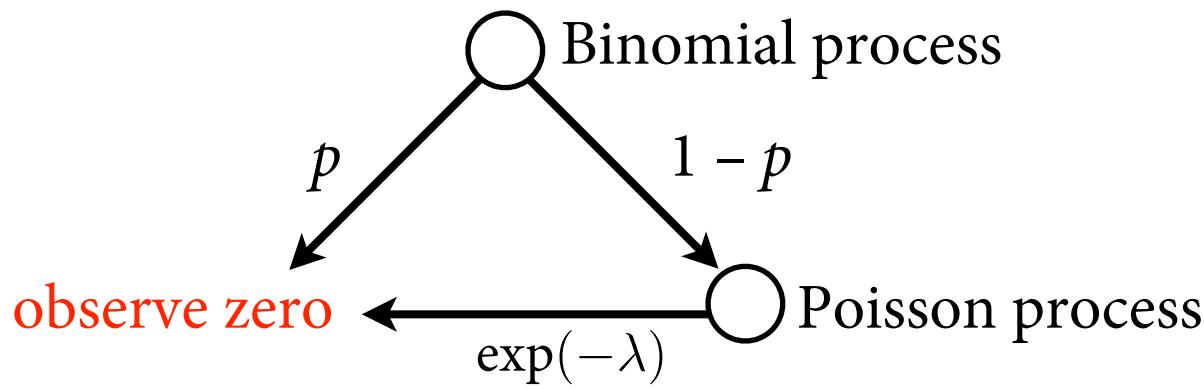


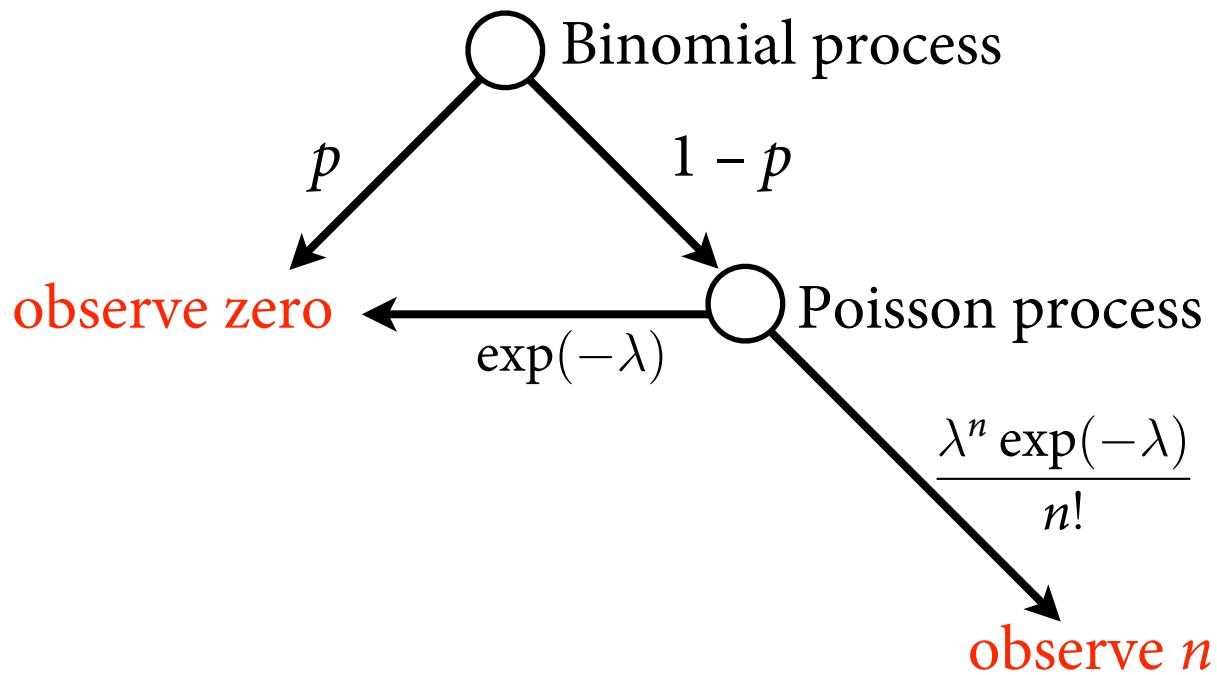
Analyze?

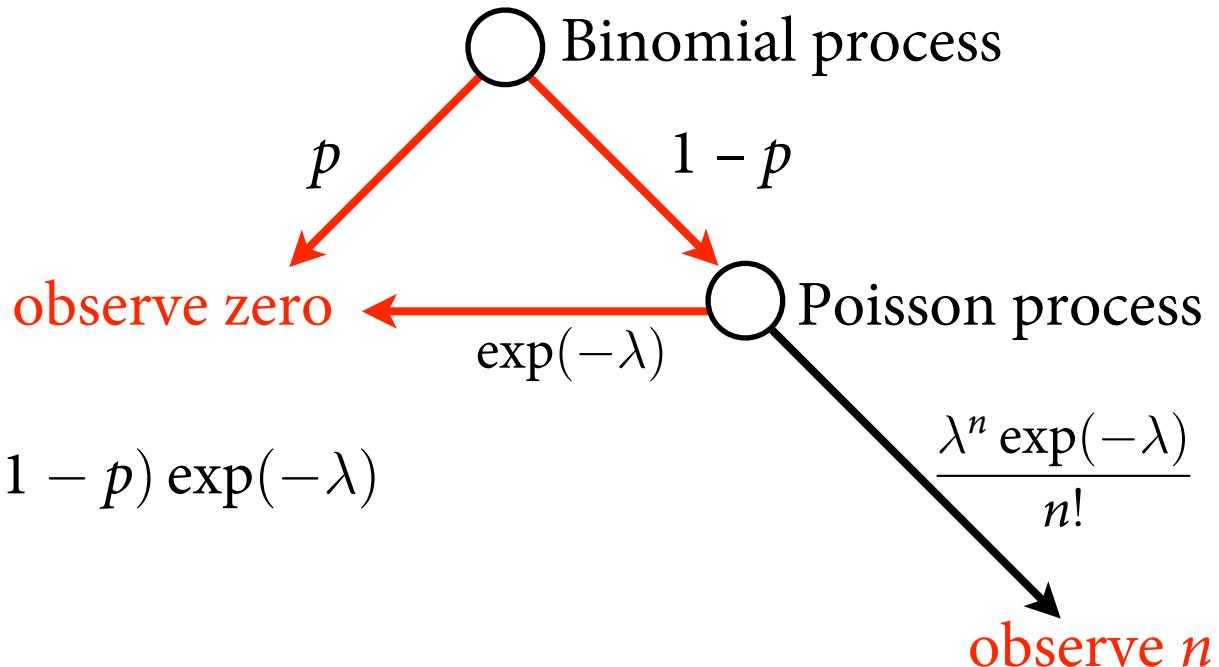




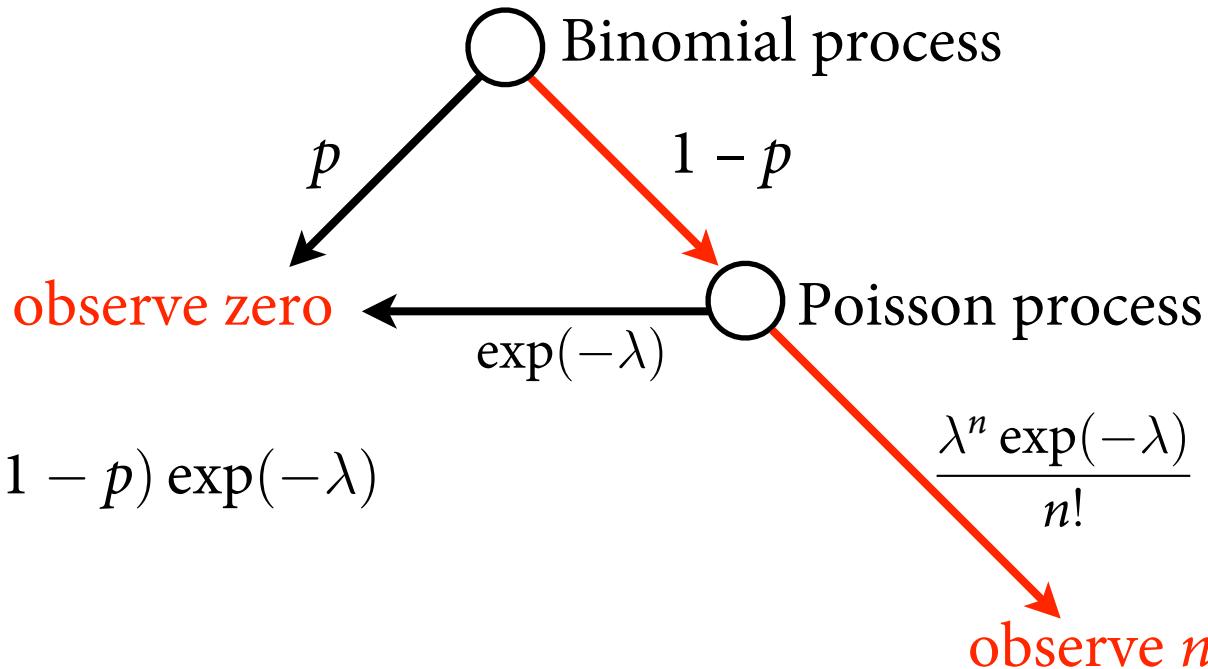








$$\Pr(0|p, \lambda) = p + (1 - p) \exp(-\lambda)$$



$$\Pr(0|p, \lambda) = p + (1 - p) \exp(-\lambda)$$

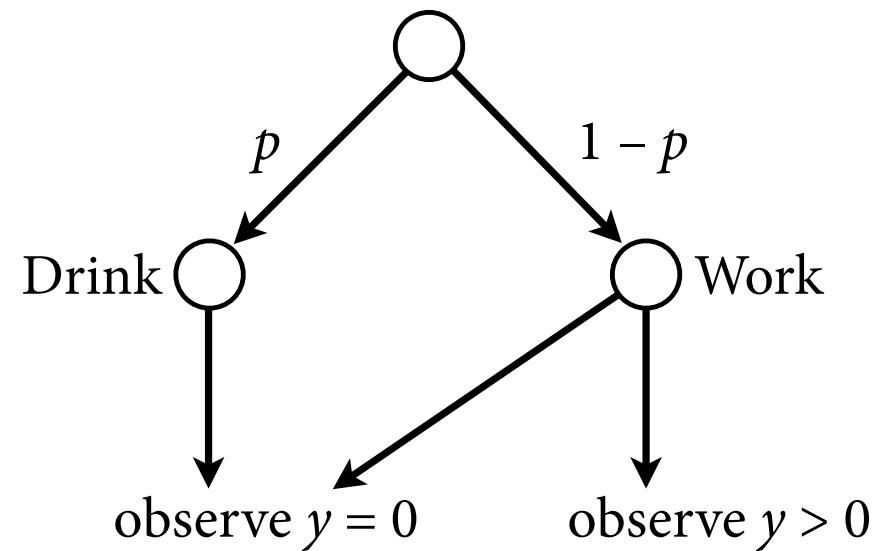
$$\Pr(n|p, \lambda) = (1 - p) \frac{\lambda^n \exp(-\lambda)}{n!}$$

Zero-inflated Poisson model

$$y_i \sim \text{ZIPoisson}(p_i, \lambda_i)$$

$$\text{logit}(p_i) = \alpha_p + \beta_p x_i$$

*Linear models
are independent*



Simulate, validate, cromulate

- As models get more complicated, no guarantees you can
 - specify model correctly
 - estimate actual process reliably
 - Bayes not magic, just logic
- Simulate “dummy data”
 - recover estimates
 - understand the model
- Try parameter combinations hostile to estimation, so you know limits of the golem



Simulated manuscripts

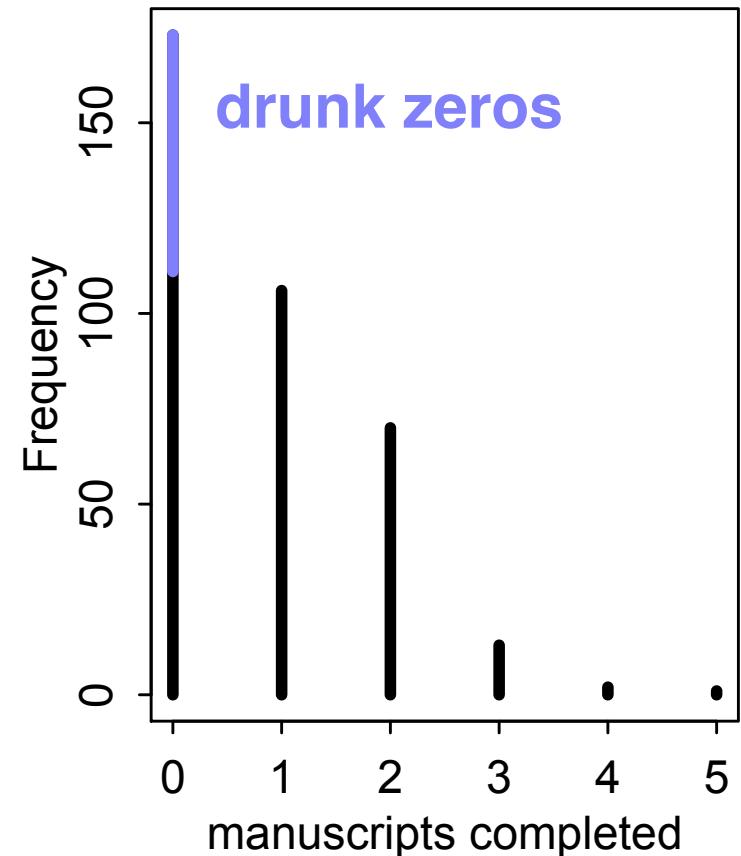
code
12.7

```
# define parameters
prob_drink <- 0.2 # 20% of days
rate_work <- 1      # average 1 manuscript per day

# sample one year of production
N <- 365

# simulate days monks drink
set.seed(365)
drink <- rbinom( N , 1 , prob_drink )

# simulate manuscripts completed
y <- (1-drink)*rpois( N , rate_work )
```



$$y_i \sim \text{ZIPoisson}(p_i, \lambda_i)$$

$$\text{logit}(p_i) = \alpha_p$$

$$\log(\lambda_i) = \alpha_\lambda$$

```
m12.4 <- ulam(  
  alist(  
    y ~ dzipois( p , lambda ),  
    logit(p) <- ap,  
    log(lambda) <- al,  
    ap ~ dnorm( -1.5 , 1 ),  
    al ~ dnorm( 1 , 0.5 )  
  ) , data=list(y=as.integer(y)) , chains=4 )  
precis( m12.4 )
```

	mean	sd	5.5%	94.5%	n_eff	Rhat
ap	-1.28	0.35	-1.90	-0.80	626	1.01
al	0.01	0.09	-0.13	0.15	618	1.00

```
inv_logit(-1.28) # probability drink  
exp(0.01)         # rate finish manuscripts, when not drinking
```

```
[1] 0.2175502
```

```
[1] 1.01005
```

Under the hood

- Same model, raw coding
- See Overthinking box on page 383

```
m12.4_alt <- ulam(  
  alist(  
    y|y>0 ~ custom( log1m(p) + poisson_lpmf(y|lambda) ) ,  
    y|y==0 ~ custom( log_mix( p , 0 , poisson_lpmf(0|lambda) ) ) ,  
    logit(p) <- ap ,  
    log(lambda) <- al ,  
    ap ~ dnorm(-1.5,1) ,  
    al ~ dnorm(1,0.5)  
  ) , data=list(y=as.integer(y)) , chains=4 )
```

Other mixtures

- Can ZIBinomial, too
- Also “hurdle” models, aka zero-augmented
- Continuous mixtures for overdispersed counts
 - beta-binomial
 - gamma-Poisson
- Multilevel models are a sort of mixture

Ordered categories

- How much do you like this class? (1–7)
- How important is income of a potential spouse? (1–10)
- How often do you see bats?
(never, sometimes, frequently)
- Depth harbor seals dive? (shallow, middle, deep)