# Homework 6

## Question 1

. The data in data(NWOGrants) are outcomes for scientific funding applications for the Netherlands Organization for Scientific Research (NWO) from 2010–2012 (see van der Lee and Ellemers doi:10.1073/pnas.1510159112). These data have a very similar structure to the UCBAdmit data discussed in Chapter 11. I want you to consider a similar question: What are the total and indirect causal effects of gender on grant awards? Consider a mediation path (a pipe) through discipline. Draw the corresponding DAG and then use one or more binomial GLMs to answer the question. What is your causal interpretation? If NWO's goal is to equalize rates of funding between the genders, what type of intervention would be most effective?

```
data(NWOGrants)

d <- NWOGrants

head(d)
```

```
##          discipline gender applications awards
## 1 Chemical sciences      m           83     22
## 2 Chemical sciences      f           39     10
## 3 Physical sciences      m          135     26
## 4 Physical sciences      f           39      9
## 5           Physics      m           67     18
## 6           Physics      f            9      2
```

```
d1 <- d %>%
  mutate(discipline=as.numeric(discipline), gender=as.numeric(gender))


dat_list1 <- list(
  discipline=d1$discipline,
  gender=d1$gender,
  applications=d1$applications,
  awards=d1$awards,
  N=18,
  K=9
)

head(dat_list1)
```

```
## $discipline
##  [1] 1 1 6 6 7 7 3 3 9 9 4 4 2 2 8 8 5 5
##
## $gender
##  [1] 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
```

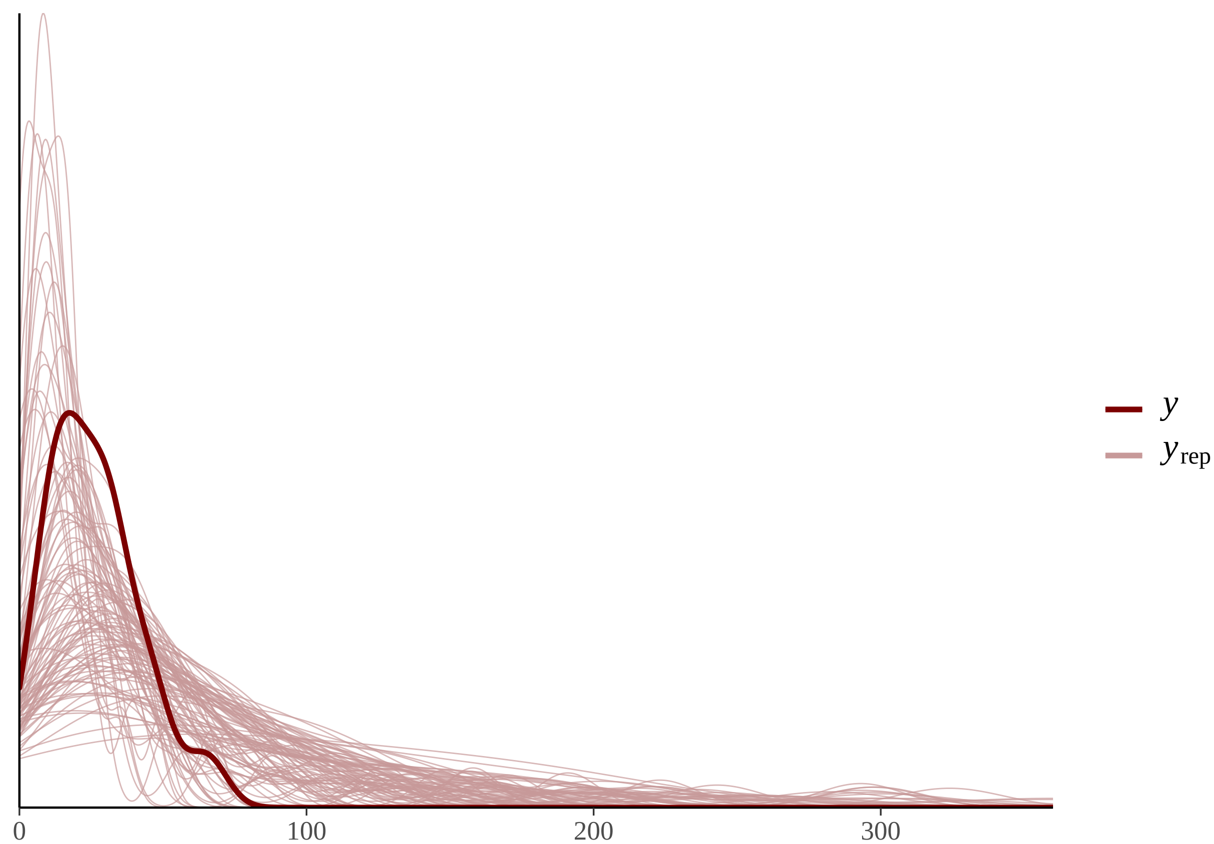
```
##
## $applications
##  [1]  83  39 135  39  67   9 230 166 189  62 105  78 156 126 425 409 245 260
##
## $awards
##  [1] 22 10 26  9 18  2 33 32 30 13 12 17 38 18 65 47 46 29
##
## $N
## [1] 18
##
## $K
## [1] 9
```

```
data {

  int N; // Number of individuals
  int K; // Number of careers

  int awards[N];
 // int discipline[N];
  int gender[N];
  int applications[N];

  int<lower=0, upper=1> PRIOR_ONLY;
}

parameters {
  real<lower=0> sigma;

  vector[2] gender_effect;
//  vector[K] discipline_effect;

}

model{
  vector[N] mu;

  for (i in 1:2) {
    gender_effect[i] ~ normal(-1, 1);
  }
  if (PRIOR_ONLY == 0) {
    for (i in 1:N) {
      mu[i] = gender_effect[gender[i]];
    }

    for (i in 1:N){
      awards[i] ~ binomial(applications[i], inv_logit(mu[i]));
    }
  }

}

generated quantities {
  vector[N] awards_sim;
```

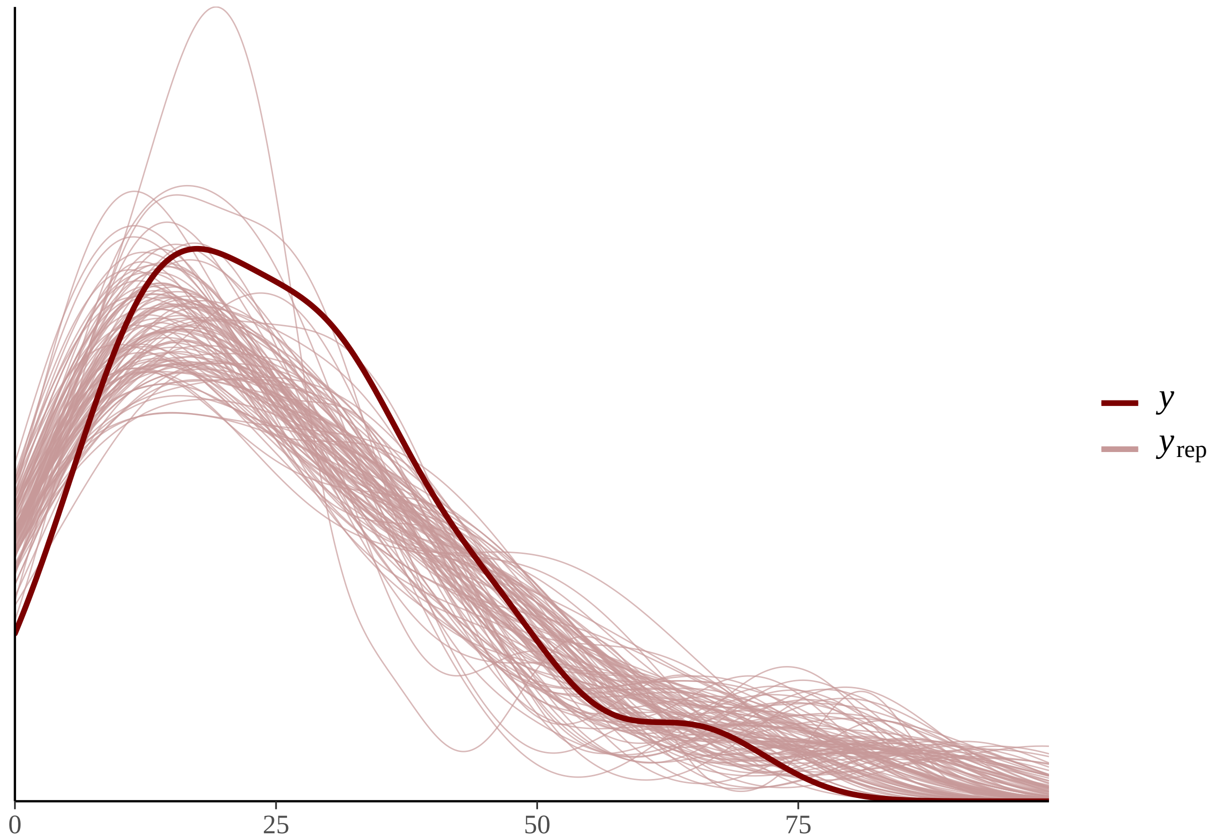
```
  vector[N] mu;

  for (i in 1:N) {
    mu[i] = gender_effect[gender[i]];
  }

  // prior predictive distributions for p patients:
  for(i in 1:N) {
    awards_sim[i] = binomial_rng(applications[i], inv_logit(mu[i]));
  }
}
```

Lets first check the prior predictive simulations

```
color_scheme_set("red")
ppc_dens_overlay(y = dat_list1$awards,
                 yrep = extract.samples(prior_mod)$awards_sim[1:100,])
```

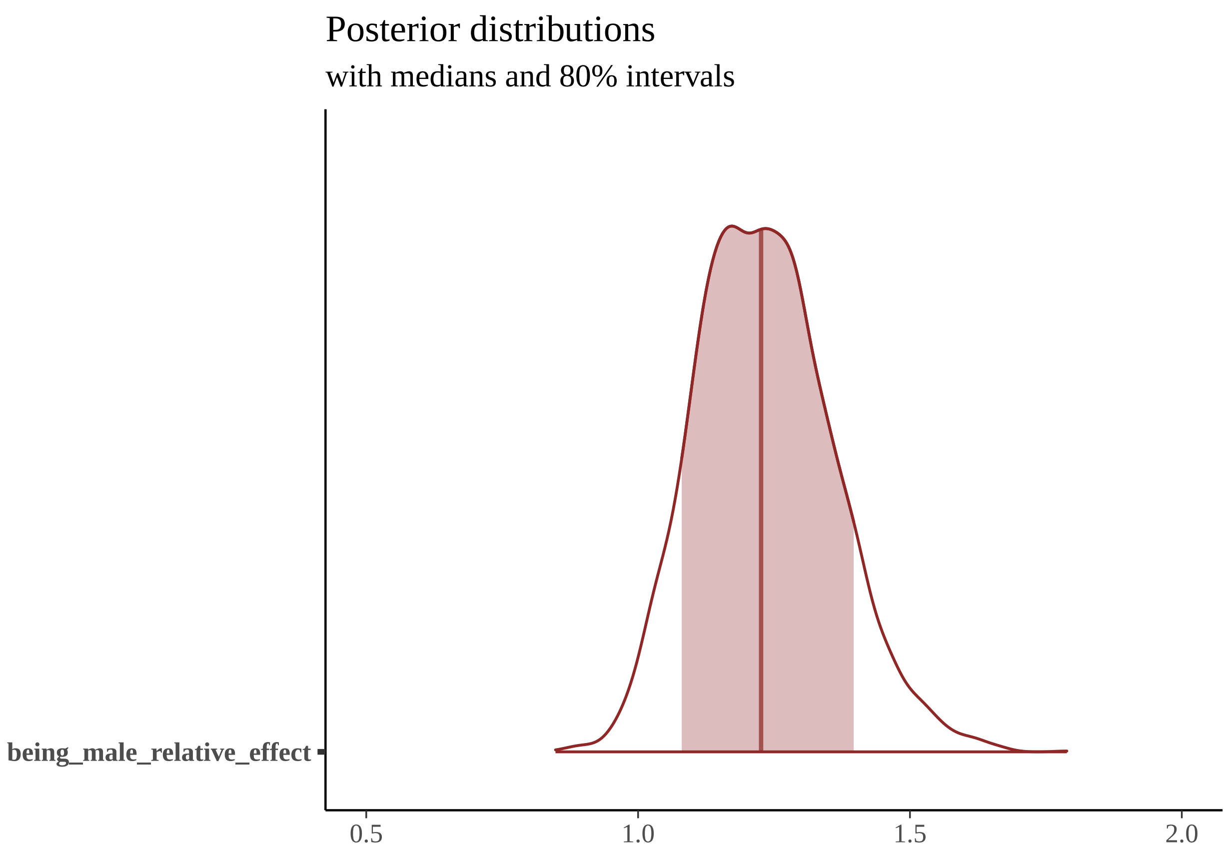



```
color_scheme_set("red")
ppc_dens_overlay(y = dat_list1$awards,
                 yrep = extract.samples(posterior_model1)$awards_sim[1:100,])
```




```
library(ggplot2)

posterior <- as.matrix(posterior_model1)

p1 <- exp(posterior[,'gender_effect[2]'] - posterior[,'gender_effect[1]'])

plot_title <- ggtitle("Posterior distributions",
                      "with medians and 80% intervals")
mcmc_areas(matrix(p1, dimnames = list(iterations=NULL, parameters='being_male_relative_effect')), pars=c
           prob = 0.8) + xlim(0.5,2) + plot_title
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```




```r
precis(p1, depth = 2)
```

```
##         mean        sd   5.5%   94.5%
## p1 1.233182 0.125166 1.0443 1.44169
##                                                                                             histogram
## p1 <U+2581><U+2581><U+2582><U+2587><U+2587><U+2585><U+2582><U+2581><U+2581><U+2581>
```

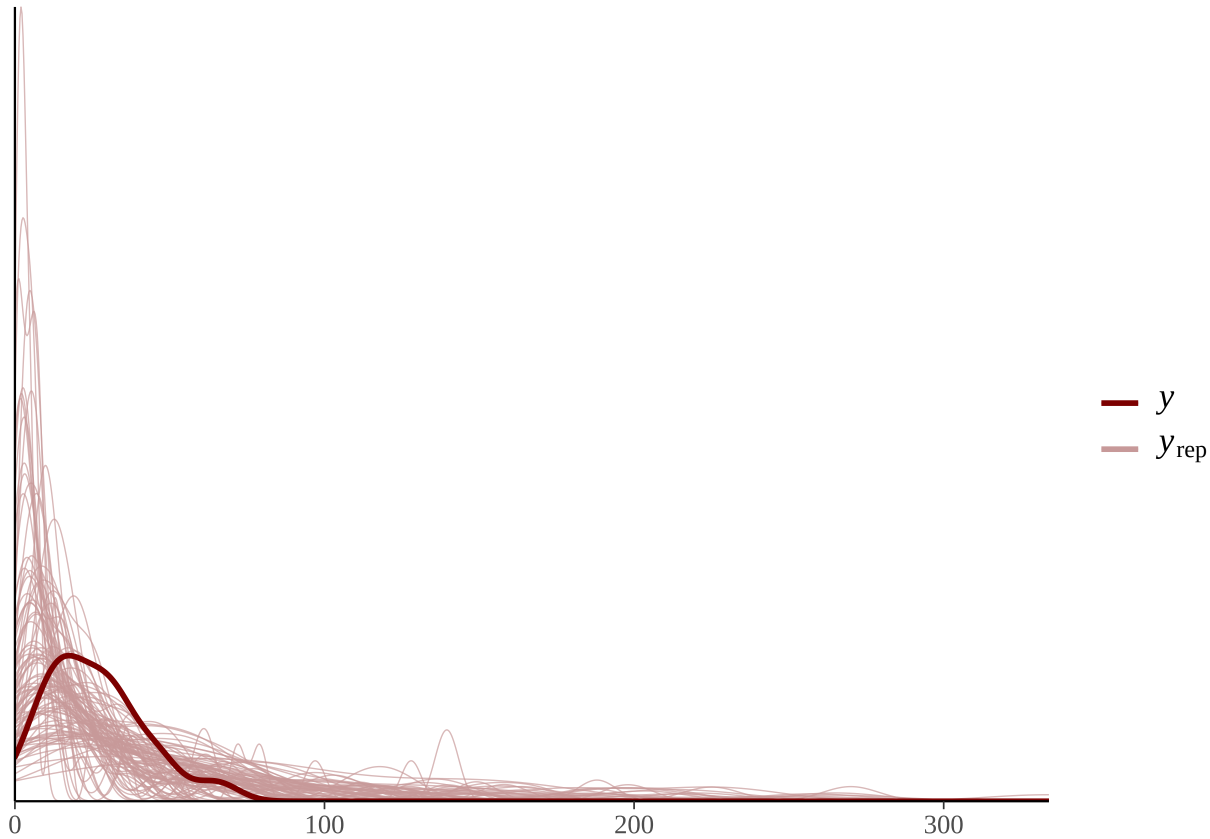
Okay lets consider discipline too

```
data {

  int N; // Number of individuals
  int K; // Number of careers

  int awards[N];
  int discipline[N];
  int gender[N];
  int applications[N];

  int<lower=0, upper=1> PRIOR_ONLY;
}

parameters {
  vector[2] gender_effect;
  vector[K] discipline_effect;
```

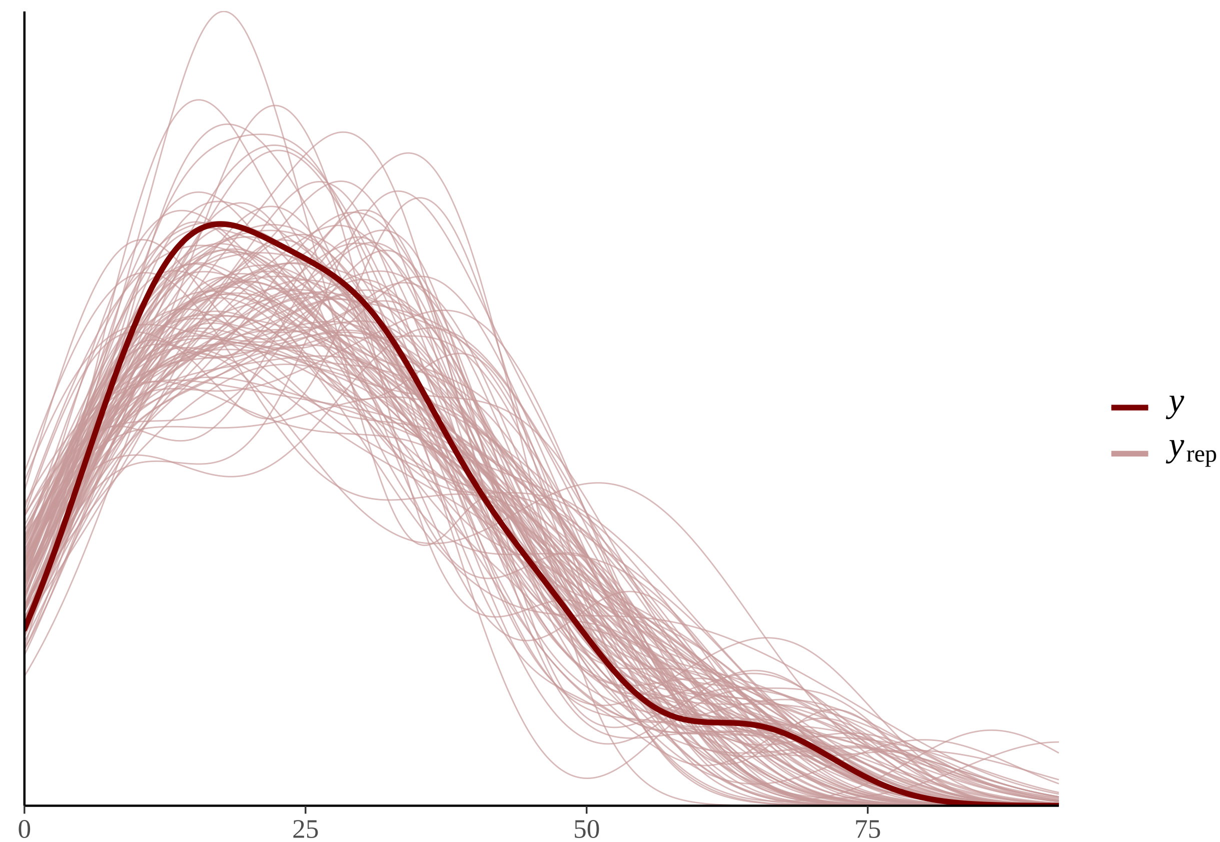
```
}

model{
  vector[N] mu;

  for (i in 1:2) {
    gender_effect[i] ~ normal(-1, 1);
  }

  for (i in 1:K) {
    discipline_effect[i] ~ normal(-1, 1);
  }

  if (PRIOR_ONLY == 0) {
    for (i in 1:N) {
      mu[i] = discipline_effect[discipline[i]] + gender_effect[gender[i]];
    }

    for (i in 1:N){
      awards[i] ~ binomial(applications[i], inv_logit(mu[i]));
    }
  }

}

generated quantities {
  vector[N] awards_sim;
  vector[N] mu;

  for (i in 1:N) {
    mu[i] = gender_effect[gender[i]] + discipline_effect[discipline[i]];
  }


  // prior predictive distributions for p patients:
  for(i in 1:N) {
    awards_sim[i] = binomial_rng(applications[i], inv_logit(mu[i]));
  }
}
```

Lets first check the prior predictive simulations

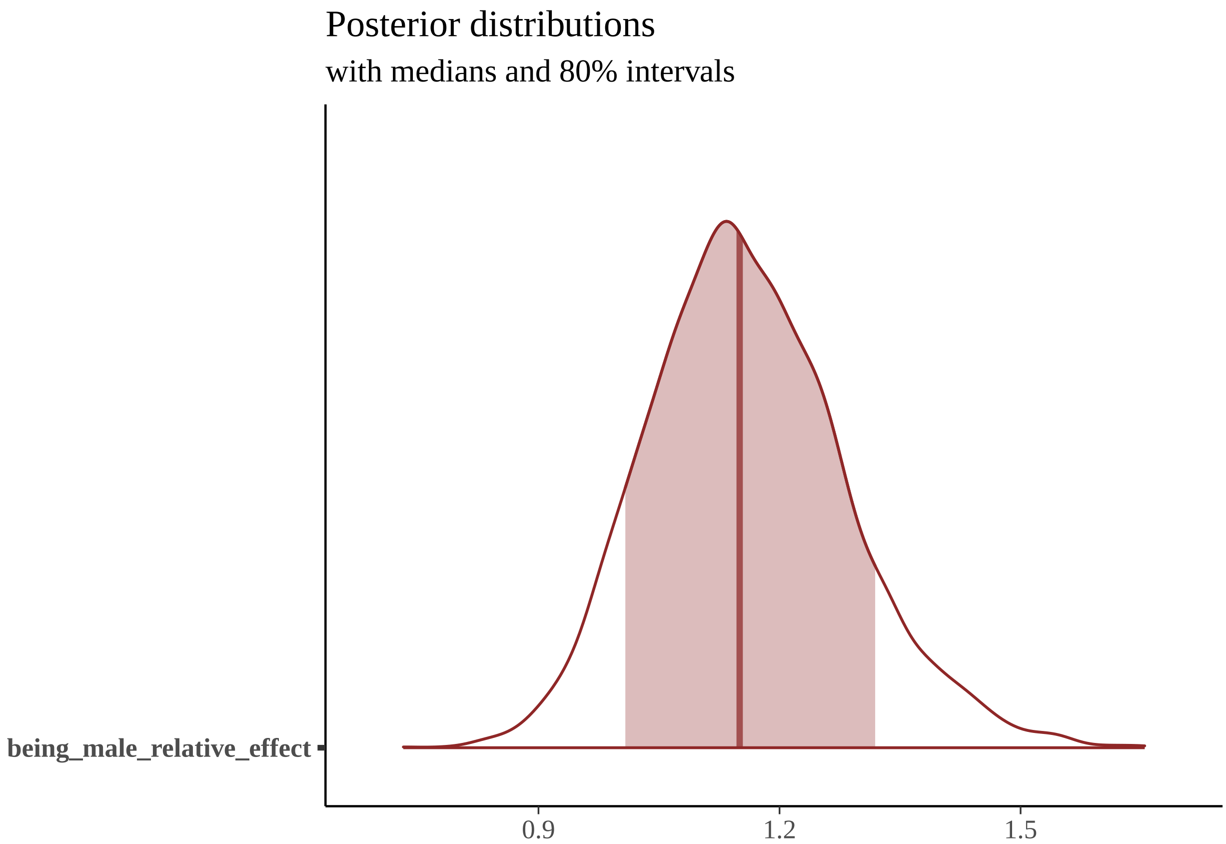
```
color_scheme_set("red")
ppc_dens_overlay(y = dat_list1$awards,
                 yrep = extract.samples(prior_mod)$awards_sim[1:100,])
```




```
color_scheme_set("red")
ppc_dens_overlay(y = dat_list1$awards,
                 yrep = extract.samples(posterior_model2)$awards_sim[1:100,])
```




```
library(ggplot2)

posterior <- as.matrix(posterior_model2)

p2 <- exp(posterior[,'gender_effect[2]'] - posterior[,'gender_effect[1]'])

plot_title <- ggtitle("Posterior distributions",
                      "with medians and 80% intervals")
mcmc_areas(matrix(p2, dimnames = list(iterations=NULL, parameters='being_male_relative_effect')), pars=c
           prob = 0.8)+ plot_title
```




```
precis(p2, depth = 2)
```

```
##        mean        sd      5.5%    94.5%
## p2 1.159701 0.1229276 0.9772295 1.369695
##                                                                                              histogram
## p2 <U+2581><U+2581><U+2582><U+2585><U+2587><U+2585><U+2582><U+2581><U+2581><U+2581>
```

### Question 1

The data in data(Primates301) were first introduced at the end of Chapter 7. In this problem, you will consider how brain size is associated with social learning. There are three parts. First, model the number of observations of social_learning for each species as a function of the log brain size. Use a Poisson distribution for the social_learning outcome variable. Interpret the resulting posterior. Second, some species are studied much more than others. So the number of reported instances of social_learning could be a product of research effort. Use the research_effort variable, specifically its logarithm, as an additional predictor variable. Interpret the coefficient for log research_effort. Does this model disagree with the previous one? Third, draw a DAG to represent how you think the variables social_learning, brain, and research_effort interact. Justify the DAG with the measured associations in the two models above (and any other models you used).

```
data(Primates301)

d <- Primates301

head(d)
```

```
##                          name          genus      species subspecies spp_id
## 1 Allenopithecus_nigroviridis Allenopithecus nigroviridis       <NA>      1
## 2         Allocebus_trichotis      Allocebus    trichotis       <NA>      2
## 3           Alouatta_belzebul       Alouatta     belzebul       <NA>      3
## 4             Alouatta_caraya       Alouatta       caraya       <NA>      4
## 5            Alouatta_guariba       Alouatta      guariba       <NA>      5
## 6           Alouatta_palliata       Alouatta     palliata       <NA>      6
##   genus_id social_learning research_effort brain    body group_size gestation
## 1        1               0               6 58.02 4655.00       40.0        NA
## 2        2               0               6    NA   78.09        1.0        NA
## 3        3               0              15 52.84 6395.00        7.4        NA
## 4        3               0              45 52.63 5383.00        8.9    185.92
## 5        3               0              37 51.70 5175.00        7.4        NA
## 6        3               3              79 49.88 6250.00       13.1    185.42
##   weaning longevity sex_maturity maternal_investment
## 1  106.15     276.0           NA                  NA
## 2      NA        NA           NA                  NA
## 3      NA        NA           NA                  NA
## 4  323.16     243.6      1276.72              509.08
## 5      NA        NA           NA                  NA
## 6  495.60     300.0      1578.42              681.02
```

```
d_q3 <- d %>% select(genus, brain, social_learning, research_effort) %>% drop_na()

head(d_q3)
```

```
##            genus brain social_learning research_effort
## 1 Allenopithecus 58.02               0               6
## 2       Alouatta 52.84               0              15
## 3       Alouatta 52.63               0              45
## 4       Alouatta 51.70               0              37
## 5       Alouatta 49.88               3              79
## 6       Alouatta 51.13               0              25
```

```
data {

  int N; // Number of individuals

  int social_learning[N];
  real brain[N];
  int<lower=0, upper=1> PRIOR_ONLY;
}

parameters {
  real intercept;
  real brain_effect;
}

model{
  brain_effect ~ normal(0, 0.5);
  intercept ~ normal(0, 1);

  if(PRIOR_ONLY == 0){
```

```
    for (i in 1:N) {
      social_learning[i] ~ poisson(exp( intercept + brain_effect * brain[i]));
    }
  }
}

generated quantities {
  int social_learning_sim[N];
  real log_lik[N];

  for(i in 1:N) {
    social_learning_sim[i] = poisson_rng(exp(intercept + brain_effect * brain[i]));
  }

  for(i in 1:N) {
    log_lik[i] = poisson_lpmf(social_learning[i] | exp(intercept + brain_effect * brain[i]));
  }
}
```

Lets first check the prior predictive simulations

```
dat_list_q3 <- list(
  brain=standardize( log(d_q3$brain)),
  social_learning=d_q3$social_learning,
  research_effort=log(d_q3$research_effort),
  N=150
)

head(dat_list_q3)
```

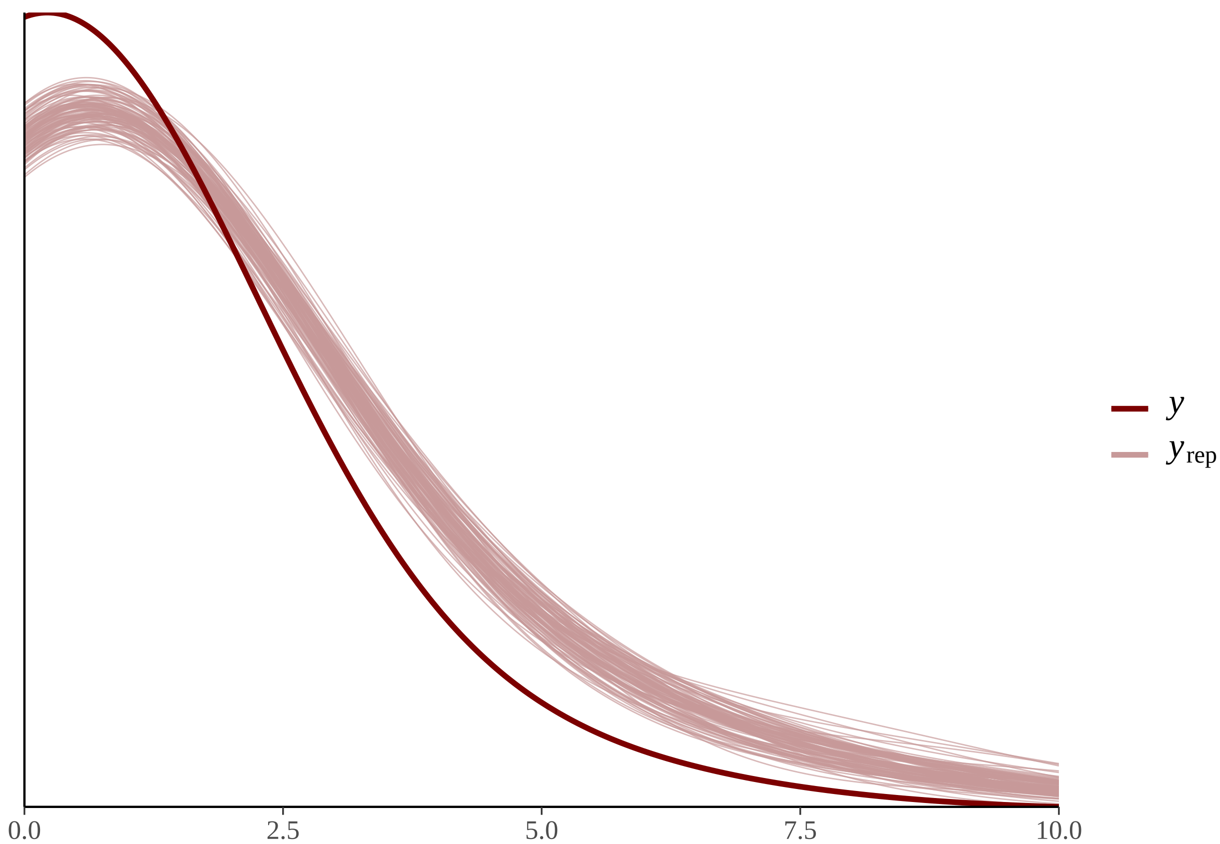
```
## $brain
##   [1]  0.35248309  0.27572868  0.27246037  0.25782792  0.22841475  0.24872896
##   [7]  0.36734220  0.31188751 -0.49459940 -0.68953856 -0.66230207 -1.39270130
##  [13]  0.92827437  0.90854118  0.84002277  0.83028102 -1.10210392 -1.27881949
##  [19]  0.88255812  0.57403641  0.49199120 -0.98083621 -1.27881949 -1.35559969
##  [25] -1.80840251  0.45138040  0.46604562  0.54019490  0.50466275  0.79160734
##  [31]  0.84702168  0.37425892  0.34352256  0.44899436  0.41497145  0.45075317
##  [37]  0.45700402  0.55436410  0.52198850  0.40481520  0.45787536  0.51968404
##  [43]  0.30980405  0.37398337  0.39962278 -1.53619437 -2.19612070  0.20250614
##  [49]  0.44571798  0.59219260  0.55646302  0.55026125  0.56207057  0.14117365
##  [55]  0.78043052 -0.49539391 -0.31360501 -0.35474823 -0.51469671 -0.29908396
##  [61] -1.57673265 -1.56934526 -1.90433324 -1.85081145 -2.18048722  2.10430063
##  [67] -0.80911922 -0.27109296  0.72331374  0.69426550  0.81448185  0.66714545
##  [73]  0.66289244 -0.06681568  0.77003546 -0.41051237 -0.95191180 -0.88600482
##  [79] -1.41921775 -1.35446686 -1.39865298 -1.11131163 -1.12746330 -1.24841849
##  [85]  0.74823069  0.81222288 -1.52776211  0.80500099  0.71698716  0.63640058
##  [91]  0.43273666  0.82289806  0.70344823  0.84391842  0.75631337  0.56174177
##  [97]  0.66589117  0.50301587  0.74147781  1.12103788  1.15301432 -2.57934689
## [103] -2.53523717 -1.53760821  0.73351376  0.94466181 -1.07993168 -1.35673408
## [109] -0.95608150 -0.97582518  1.80677723  1.85750512  1.22222885  1.20122591
## [115]  1.11082636  1.08995649  1.27252168 -0.91266068  0.42771844  0.34151798
## [121] -0.12925431  1.88927741  0.61920652  0.44382179  0.27199241  0.04313157
## [127] -0.29971000  0.72556146  0.93344810 -1.27985251 -1.11553135 -1.10879017
```

```
## [133] -1.00562045 -1.11047028 -0.33620727 -0.36723239  0.88440988  0.97250886
## [139] -2.03602798 -2.07867307 -1.98566044  1.03536562  0.35021665  0.62936424
## [145]  0.66201978  0.40852294  0.53918144  0.82861969  0.39748301 -0.13282383
## attr(,"scaled:center")
## [1] 3.631314
## attr(,"scaled:scale")
## [1] 1.218423
##
## $social_learning
##   [1]   0   0   0   0   3   0   0   0   0   0   0   0   0   0   2   0   0   0
##  [19]   0   0   0   0   0   2   0   1  17   5   0   0   1   1   0   0   1   0
##  [37]   0   0   0   0   0   0   0   0   0   0   0   0   5   0   0   0   0   0
##  [55]   2   0   1   0   0   0   0   0   0   0   0  13   0   0   0   0   0   0
##  [73]   0   0   0   4   0   0   0   0   0   0   0   0   0   0   0   1   0   0
##  [91]   7  45  15   3   0   0   1   0   0   0   3   0   0   0   0   0   0   0
## [109]   1   1   5 214   4   2   1   3   5   0   0   1   0  86   0   0   0   0
## [127]   1   0   0   2   0   0   0   0   1   1   2   0   0   0   0   0   0   0
## [145]   1   0   0   0   0   0
##
## $research_effort
##   [1] 1.7917595 2.7080502 3.8066625 3.6109179 4.3694479 3.2188758 1.3862944
##   [8] 4.4067192 3.0910425 2.7725887 4.0604430 0.0000000 2.4849066 1.3862944
##  [15] 4.0604430 3.4011974 2.3025851 1.7917595 3.1780538 2.3978953 2.0794415
##  [22] 3.7612001 2.7725887 5.0814044 3.5835189 2.5649494 5.5174529 4.0943446
##  [29] 2.8903718 2.9444390 3.4657359 3.2580965 2.3978953 2.0794415 3.3322045
##  [36] 1.0986123 1.3862944 1.9459101 4.0253517 2.0794415 2.8332133 1.9459101
##  [43] 1.6094379 2.0794415 1.9459101 1.0986123 2.5649494 3.0445224 4.5108595
##  [50] 2.7725887 3.7376696 2.8332133 2.3025851 3.9512437 3.4965076 2.3978953
##  [57] 4.3944492 3.4657359 2.5649494 2.5649494 0.0000000 0.6931472 2.6390573
##  [64] 2.9957323 1.6094379 6.2480429 3.6888795 2.0794415 2.7725887 1.3862944
##  [71] 4.4543473 1.6094379 2.7725887 2.0794415 3.5263605 4.6347290 3.8286414
##  [78] 4.4426513 0.0000000 1.6094379 0.6931472 0.0000000 1.6094379 0.6931472
##  [85] 3.5263605 1.7917595 2.6390573 3.8712010 2.8332133 2.4849066 5.1590553
##  [92] 5.5333895 5.6903595 3.9318256 3.2958369 3.5263605 3.8712010 2.4849066
##  [99] 4.2046926 2.8903718 3.4011974 4.1896547 2.0794415 1.0986123 2.8332133
## [106] 1.3862944 3.6109179 2.9444390 3.5835189 2.4849066 5.4161004 6.6267177
## [113] 3.7612001 4.7361984 4.3567088 2.0794415 3.0910425 2.3025851 3.9512437
## [120] 1.9459101 3.3322045 5.7714411 2.3978953 1.7917595 1.0986123 3.3322045
## [127] 3.7135721 3.2188758 3.5835189 4.3944492 1.0986123 2.8332133 3.8286414
## [134] 5.0304379 1.3862944 4.4886364 4.5849675 3.6888795 2.0794415 0.6931472
## [141] 2.3025851 3.5263605 2.0794415 1.9459101 2.1972246 1.7917595 2.7725887
## [148] 1.6094379 0.6931472 4.0430513
##
## $N
## [1] 150
```

```
color_scheme_set("red")
p <- ppc_dens_overlay(bw=2, y = dat_list_q3$social_learning,
                 yrep = extract.samples(posterior_q3_1)$social_learning_sim[1:100,])

p + xlim(0, 10)
```

```
## Warning: Removed 526 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 6 rows containing non-finite values (stat_density).
```

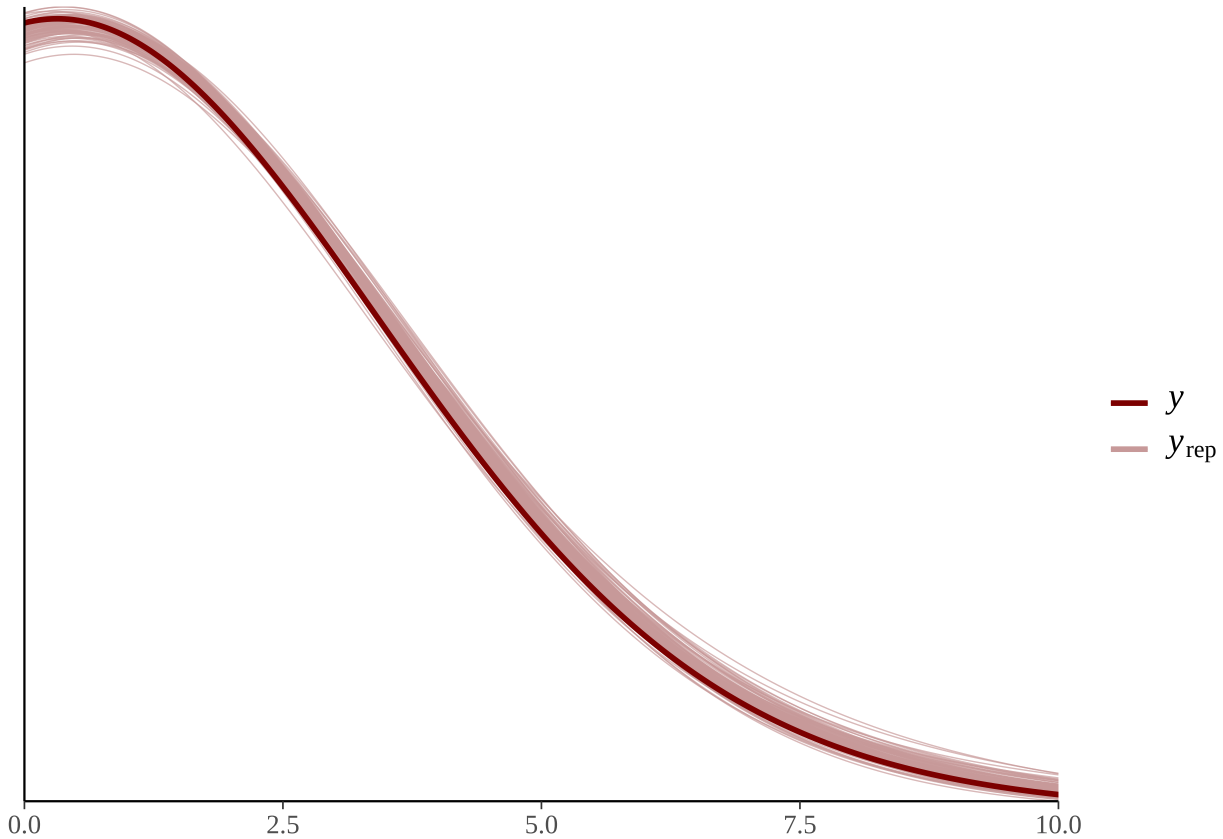



```
data {

  int N; // Number of individuals

  int social_learning[N];
  real brain[N];
  real research_effort[N];
  int<lower=0, upper=1> PRIOR_ONLY;
}

parameters {
  real intercept;
  real brain_effect;
  real research_effort_effect;
}

model{
  brain_effect ~ normal(0, 0.5);
  research_effort_effect ~ normal(0, 0.5);
  intercept ~ normal(0, 1);

  if(PRIOR_ONLY == 0){
    for (i in 1:N) {
```

```
      social_learning[i] ~ poisson(exp( intercept + brain_effect * brain[i] + research_effort_effect * r
    }
  }
}

generated quantities {
  int social_learning_sim[N];
  real log_lik[N];

  for(i in 1:N) {
    social_learning_sim[i] = poisson_rng(exp(intercept + brain_effect * brain[i]  + research_effort_effe
  }

  for(i in 1:N) {
    log_lik[i] = poisson_lpmf(social_learning[i] | exp(intercept + brain_effect * brain[i]  + research_e
  }

}
```

```
color_scheme_set("red")
p <- ppc_dens_overlay(bw=3, y = dat_list_q3$social_learning,
                 yrep = extract.samples(posterior_q3_2)$social_learning_sim[1:100,])

p + xlim(0, 10)
```

```
## Warning: Removed 696 rows containing non-finite values (stat_density).

## Warning: Removed 6 rows containing non-finite values (stat_density).
```



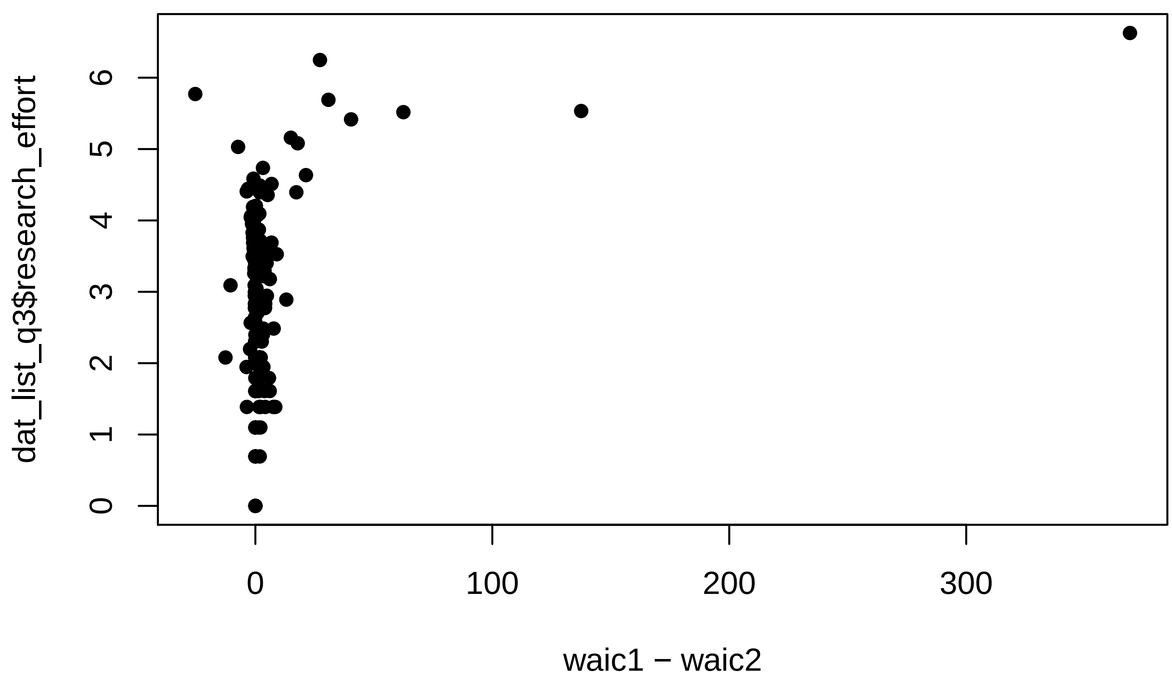

```
library(loo)
```

```
## Warning: package 'loo' was built under R version 4.0.5

## This is loo version 2.4.1

## - Online documentation and vignettes at mc-stan.org/loo

## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' arg

## - Windows 10 users: loo may be very slow if 'mc.cores' is set in your .Rprofile file (see https://gi

##
## Attaching package: 'loo'

## The following object is masked from 'package:rethinking':
##
##     compare

## The following object is masked from 'package:rstan':
##
##     loo
```

```r
loo_compare(loo(posterior_q3_1), loo(posterior_q3_2))
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

##        elpd_diff se_diff
## model2    0.0       0.0
## model1 -393.8     167.8
```

```r
waic1 <- WAIC( posterior_q3_1 , pointwise=TRUE )$WAIC
waic2 <- WAIC( posterior_q3_2 , pointwise=TRUE )$WAIC
plot( waic1 - waic2 , dat_list_q3$research_effort , pch=16)
identify( waic1-waic2 , dat_list_q3$log_effort , d_q3$genus , cex=0.8 )
```




```
## integer(0)
```