hw8 for stat341

Zhihong Zhang

Feb 28th,2017

Q: 5E4 5H3

5E4. Suppose you have a single categorical predictor with 4 levels (unique values), labeled A, B, C and D. Let A_i be an indicator variable that is 1 where case i is in category A. Also suppose B_i, C_i , and $D_i$for the other categories. Now which of the following linear models are inferentially equivalent ways to include the categorical variable in a regression? Models are inferentially equivalent when it's possible to compute one posterior distribution from the posterior distribution of another model.

$(1)\mu_i = \alpha + \beta_A A_i + \beta_B B_i + \beta_D D_i$
$(2)\mu_i = \alpha + \beta_A A_i + \beta_B B_i + \beta_C C_i + \beta_D D_i$
$(3)\mu_i = \alpha + \beta_B B_i + \beta_C C_i + \beta_D D_i$
$(4)\mu_i = \alpha_A A_i + \alpha_B B_i + \alpha_C C_i + \alpha_D D_i$
$(5)\mu_i = \alpha_A(1 - B_i - C_i - D_i) + \alpha_B B_i + \alpha_C C_i + \alpha_D D_i$

Solution:

Except(4),all the others inferentially equivalent ways to include the categorical variable in a regression.

5H3. Finally, consider the avgfood variable. Fit two more multiple regressions: (1) body weight as an additive function of avgfood and groupsize , and (2) body weight as an additive function of all three variables, avgfood and groupsize and area . Compare the results of these models to the previous models you've fit, in the first two exercises. (a) Is avgfood or area a better predictor of body weight? If you had to choose one or the other to include in a model, which would it be? Support your assessment with any tables or plots you choose. (b) When both avgfood or area are in the same model, their effects are reduced (closer to zero) and their standard errors are larger than when they are included in separate models. Can you explain this result?

Solution:

two variable model with area and groupsize

```
twovarmodel <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + b*area+c*groupsize,
    a ~ dnorm(0.6, 10),
    b ~ dnorm( 0 , 3 ),
    c ~ dnorm( 0 , 4 ),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)
precis(twovarmodel)
```

```
##        Mean StdDev  5.5% 94.5%
## a      4.45   0.37  3.86  5.04
## b      0.62   0.20  0.30  0.94
## c     -0.43   0.12 -0.62 -0.24
## sigma  1.12   0.07  1.00  1.24
```

two variable model with avgfood and groupsize

```
twovarmodel <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + b*avgfood+c*groupsize,
    a ~ dnorm(0.6, 10),
    b ~ dnorm( 0 , 3 ),
    c ~ dnorm( 0 , 4 ),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)
precis(twovarmodel)
```

```
##        Mean StdDev  5.5% 94.5%
## a      4.25   0.42  3.58  4.92
## b      3.30   1.13  1.50  5.10
## c     -0.51   0.15 -0.74 -0.27
## sigma  1.12   0.07  1.00  1.23
```

Comparing these two models, avgfood is a better predictor of body weight. Since it has relatively large number of parameter to affect the weight. Also based on the parameters, the groupsize has less effect on both two models. Therefore for a two variable model, I may only consider avgfood and the area.

two variable models with constant avgfood and varied groupsize.

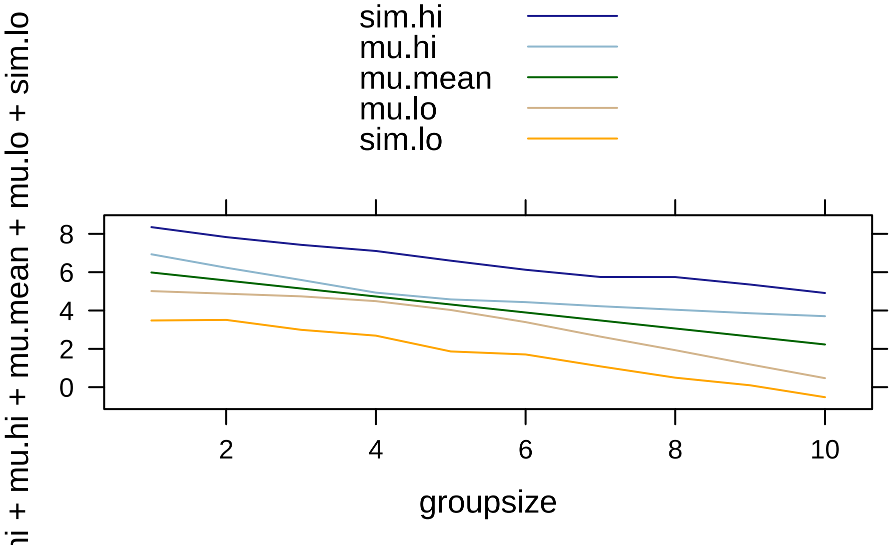
```
data("foxes")
linearmodel <- map(alist(
  weight ~ dnorm( mu , sigma ),
  mu <- a + b*avgfood+c*groupsize,
  a ~ dnorm( 10, 1 ),
  b ~ dnorm( 0 , 3 ),
  c ~ dnorm( 0 , 4 ),
  sigma ~ dunif( 0, 10 )
), data =foxes )

linearmodel.pred <-
  data_frame(
    groupsize = seq(from = 1, to = 10, by = 1),
    avgfood= mean(foxes$avgfood)
  )

mu <- link(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

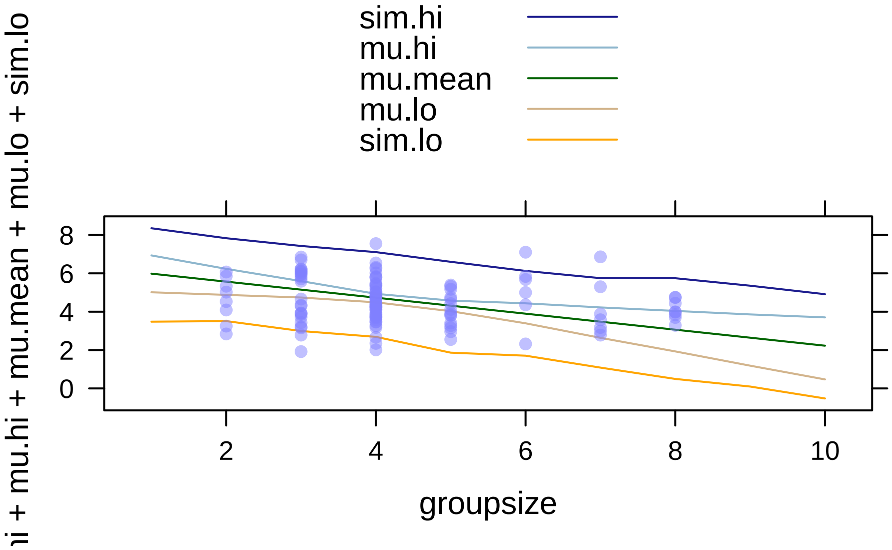
```
sim.groupsize <- sim(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
linearmodel.pred <-
  linearmodel.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.groupsize, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.groupsize, 2, HPDI,prob=0.95)[2,]
  )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ groupsize,
       data = linearmodel.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```



```
plotPoints(weight ~ groupsize, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```



two models with constant avgfood

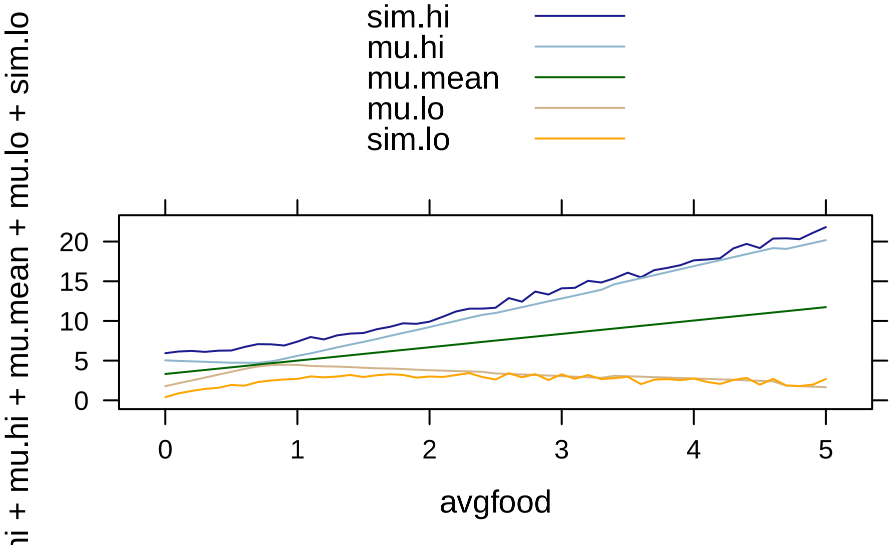
```r
data("foxes")
linearmodel <- map(alist(
weight ~ dnorm( mu , sigma ),
mu <- a + b*avgfood+c*groupsize,
a ~ dnorm( 10, 1 ),
b ~ dnorm( 0 , 3 ),
c ~ dnorm( 0 , 4 ),
sigma ~ dunif( 0, 10 )
), data =foxes )

linearmodel.pred <-
  data_frame(
  avgfood = seq(from = 0, to = 5, by = 0.1),
  groupsize= mean(foxes$groupsize)
  )

mu <- link(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

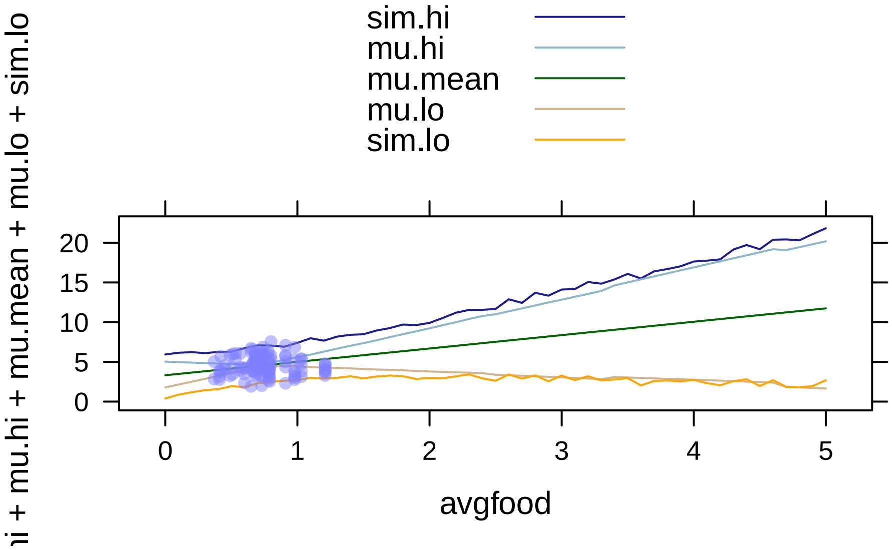
```r
sim.avgfood <- sim(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
linearmodel.pred <-
  linearmodel.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.avgfood, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.avgfood, 2, HPDI,prob=0.95)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ avgfood,
       data = linearmodel.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```



```
plotPoints(weight ~ avgfood, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```



For the three variable model, with constant groupsize and area, the graph is as follows:

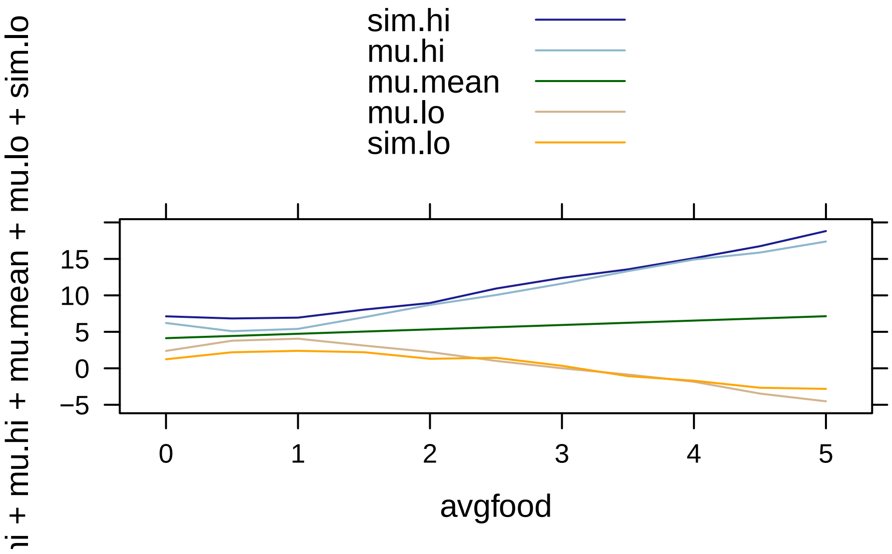
```
data("foxes")
linearmodel <- map(alist(
  weight ~ dnorm( mu , sigma ),
  mu <- a + b*avgfood+c*groupsize+d*area,
  a ~ dnorm( 10, 1 ),
  b ~ dnorm( 0 , 3 ),
  c ~ dnorm( 0 , 4 ),
  d ~ dnorm( 0 , 4 ),
  sigma ~ dunif( 0, 10 )
), data =foxes )

linearmodel.pred <-
  data_frame(
    avgfood = seq(from = 0, to = 5, by = 0.5),
    groupsize= mean(foxes$groupsize),
    area= mean(foxes$area)
  )

mu <- link(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

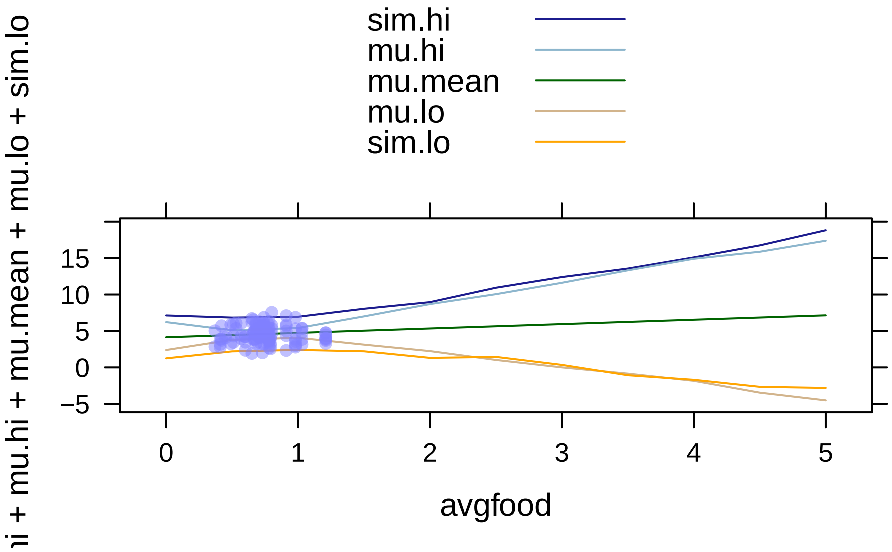
```
sim.avgfood <- sim(linearmodel, data = linearmodel.pred)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
linearmodel.pred <-
  linearmodel.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.95)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.95)[2,],
    sim.lo = apply(sim.avgfood, 2, HPDI,prob=0.95)[1,],
    sim.hi = apply(sim.avgfood, 2, HPDI,prob=0.95)[2,]
  )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ avgfood,
       data = linearmodel.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```



```r
plotPoints(weight ~ avgfood, data = foxes, col = rangi2, alpha = 0.5, add = TRUE)
```



three variable with mean and standard deviation

```
multivarmodel <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + b*avgfood+c*groupsize+d*area,
    a ~ dnorm(0.6, 10),
    b ~ dnorm( 0 , 3 ),
    c ~ dnorm( 0 , 4 ),
    d ~ dnorm( 0 , 4 ),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)
precis(multivarmodel)
```

```
##         Mean StdDev  5.5% 94.5%
## a       4.13   0.42  3.46  4.80
## b       2.04   1.31 -0.05  4.13
## c      -0.57   0.15 -0.81 -0.33
## d       0.43   0.23  0.06  0.80
## sigma  1.10   0.07  0.99  1.22
```

Based on this result, it actually make sense since the standard deviation should piled up due to increasing numbers of variable. And thus causing larger standard error. And the effect are reduced can be caused by that they both appears in the same model,and they cancel out their effect little bit.