

# INLArethinking\_HW3

Ania Kawiecki

7/30/2020

Statistical rethinking homework solutions link: [https://github.com/rmcelreath/statrethinking\\_winter2019/tree/master/homework](https://github.com/rmcelreath/statrethinking_winter2019/tree/master/homework)

INLA book: <https://becarioprecario.bitbucket.io/inla-gitbook/ch-intro.html>

```
library(tidyverse)
library(rethinking)
library(dagitty)
library(INLA)
```

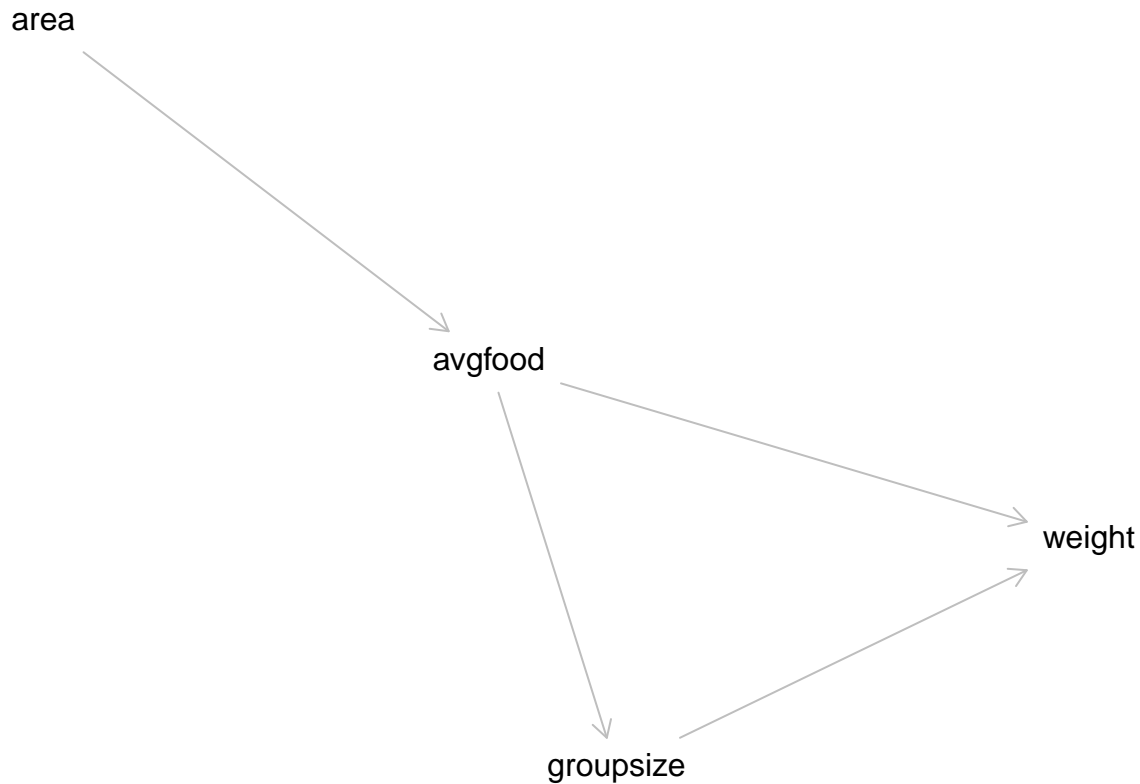
## HOMEWORK 3

All three problems below are based on the same data. The data in `data(foxes)` are 116 foxes from 30 different urban groups in England. These foxes are like street gangs. Group size varies from 2 to 8 individuals. Each group maintains its own (almost exclusive) urban territory. Some territories are larger than others. The `area` variable encodes this information. Some territories also have more `avgfood` than others. We want to model the weight of each fox. For the problems below, assume this DAG:

```
hw3dag <- dagitty("dag{
    avgfood <- area
    weight <- avgfood
    weight <- groupsize
    groupsize <- avgfood
}")

plot(hw3dag)
```

## Plot coordinates for graph not supplied! Generating coordinates, see `?coordinates` for how to set your



1. Use a model to infer the total causal influence of area on weight. Would increasing the area available to each fox make it heavier (healthier)? You might want to standardize the variables. Regardless, use prior predictive simulation to show that your model's prior predictions stay within the possible outcome range.

SOL: Because there are no back-door paths from area to weight, we only need to include area. No other variables are needed. Territory size seems to have no total causal influence on weight, at least not in this sample.

```
data(foxes)
f <- foxes

f$A <- scale( f$area )
f$F <- scale( f$avgfood )
f$G <- scale( f$groupsize )
f$W <- scale( f$weight )

# area on weight
m1 <- quap(
  alist(
    W ~ dnorm( mu , sigma ) ,
    mu <- a + bA*A ,
    a ~ dnorm( 0 , 0.2 ) ,
    bA ~ dnorm( 0 , 0.5 ) ,
    sigma ~ dexp( 1 )
```

```
) , data=f )

precis(m1)

##              mean          sd      5.5%      94.5%
## a      6.963538e-05 0.08361042 -0.1335560 0.1336952
## bA     1.884078e-02 0.09089805 -0.1264319 0.1641134
## sigma 9.912912e-01 0.06467057 0.8879352 1.0946473
# the area has no influence on the weight
```

## default priors

By default, the intercept has a Gaussian prior with mean and precision equal to zero. Coefficients of the fixed effects also have a Gaussian prior by default with zero mean and precision equal to 0.001. The prior on the precision of the error term is, by default, a Gamma distribution with parameters 1 and 0.00005 (shape and rate, respectively) ( so this is different from the statistical rethinking book's prior on the variance which is  $\exp(1)$ )

```
#using default priors

m1.inla <- inla(W~A, data= f)
summary(m1.inla)

##
## Call:
##      "inla(formula = W ~ A, data = f)"
## Time used:
##      Pre = 1.53, Running = 0.241, Post = 0.207, Total = 1.98
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant  mode kld
## (Intercept) 0.000 0.093    -0.183    0.000    0.183 0.000  0
## A           0.019 0.094    -0.165    0.019    0.203 0.019  0
##
## Model hyperparameters:
##                                mean      sd 0.025quant 0.5quant
## Precision for the Gaussian observations 1.01 0.132    0.767    1.00
##                                0.975quant  mode
## Precision for the Gaussian observations    1.28 0.992
##
## Expected number of effective parameters(stdev): 2.00(0.00)
## Number of equivalent replicates : 57.98
##
## Marginal log-Likelihood: -182.37

names(inla.models())$prior

## [1] "normal"          "gaussian"         "wishart1d"
## [4] "wishart2d"       "wishart3d"        "wishart4d"
## [7] "wishart5d"       "loggamma"         "gamma"
## [10] "minuslogsqrtruncnormal" "logtnormal"      "logtgaussian"
## [13] "flat"           "logflat"          "logiflat"
## [16] "mvnorm"         "pc.alphaw"        "pc.ar"
## [19] "dirichlet"      "none"             "invalid"
## [22] "betacorrelation" "logitbeta"        "pc.prec"
```

```
## [25] "pc.dof"          "pc.cor0"          "pc.cor1"
## [28] "pc.fgnh"         "pc.spde.GA"       "pc.matern"
## [31] "pc.range"        "pc.sn"            "pc.gamma"
## [34] "pc.mgamma"       "pc.gammacount"    "pc.gevtail"
## [37] "pc"              "ref.ar"           "pom"
## [40] "jeffreystdf"     "expression:"      "table:"

#inla.doc("normal")

m1.inla.prior <- inla(W~A, data= f, control.fixed = list(
  mean= 0,
  prec= 0.5,
  mean.intercept= 0,
  prec.intercept= 0.2
))

#m1.inla.prior = inla(W ~ A,family = c("gaussian"), data = f,control.family = list( hyper = list(prec =
summary(m1.inla.prior)

##
## Call:
## c("inla(formula = W ~ A, data = f, control.fixed = list(mean = 0, ", "
## prec = 0.5, mean.intercept = 0, prec.intercept = 0.2)))")
## Time used:
## Pre = 1.35, Running = 0.137, Post = 0.193, Total = 1.68
## Fixed effects:
##          mean      sd 0.025quant 0.5quant 0.975quant  mode kld
## (Intercept) 0.000 0.093    -0.183    0.000    0.183 0.000  0
## A           0.019 0.093    -0.164    0.019    0.203 0.019  0
##
## Model hyperparameters:
##                mean      sd 0.025quant 0.5quant
## Precision for the Gaussian observations 1.01 0.132    0.767    1.00
##                0.975quant  mode
## Precision for the Gaussian observations      1.28 0.992
##
## Expected number of effective parameters(stddev): 2.00(0.001)
## Number of equivalent replicates : 58.16
##
## Marginal log-Likelihood: -180.99
```

**2. Now infer the causal impact of adding food to a territory. Would this make foxes heavier? Which covariates do you need to adjust for to estimate the total causal influence of food?**

SOL: To infer the causal influence of avg food on weight, we need to close any back-door paths. There are no back-door paths in the DAG. So again, just use a model with a single predictor. If you include groupsize, to block the indirect path, then you won't get the total causal influence of food. You'll just get the direct influence. But I asked for the effect of adding food, and that would mean through all forward paths.

Again nothing. Adding food does not change weight. This shouldn't surprise you, if the DAG is correct, because area is upstream of avgfood.

```
# food on weight
m2 <- quap(
  alist(
    W ~ dnorm( mu , sigma ) ,
    mu <- a + bF*F,
    a ~ dnorm( 0 , 0.2 ) ,
    bF ~ dnorm( 0 , 0.5 ) ,
    sigma ~ dexp( 1 )
  ) , data=f )

precis(m2)
```

```
##              mean      sd      5.5%      94.5%
## a      8.085385e-07 0.08360015 -0.1336084 0.1336100
## bF     -2.421257e-02 0.09088499 -0.1694643 0.1210392
## sigma  9.911437e-01 0.06465854  0.8878069 1.0944805
```

```
# the food has no infuence on the weight
```

```
#using default priors
```

```
m2.inla <- inla(W~F, data= f)
summary(m2.inla)
```

```
##
## Call:
##   "inla(formula = W ~ F, data = f)"
## Time used:
##   Pre = 1.53, Running = 0.135, Post = 0.141, Total = 1.81
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)  0.000 0.093   -0.183    0.000    0.183  0.000   0
## F           -0.025 0.094   -0.209   -0.025    0.159 -0.025   0
##
## Model hyperparameters:
##              mean      sd 0.025quant 0.5quant
## Precision for the Gaussian observations 1.01 0.132    0.767    1.00
##              0.975quant   mode
## Precision for the Gaussian observations    1.28 0.992
##
## Expected number of effective parameters(stddev): 2.00(0.00)
## Number of equivalent replicates : 57.98
##
## Marginal log-Likelihood: -182.36
```

```
#using custom priors
```

```
m2.prec.prior <-
m2.inla.prior <- inla(W~F, data= f, control.fixed = list(
  mean= 0,
  prec= 0.5,
  mean.intercept= 0,
  prec.intercept= 0.2
))
```

```
summary(m2.inla.prior)
```

```
##
## Call:
## c("inla(formula = W ~ F, data = f, control.fixed = list(mean = 0, ", "
## prec = 0.5, mean.intercept = 0, prec.intercept = 0.2))")
## Time used:
## Pre = 1.23, Running = 0.138, Post = 0.155, Total = 1.52
## Fixed effects:
##          mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)  0.000 0.093    -0.183   0.000    0.183  0.000  0
## F            -0.025 0.093    -0.209  -0.025    0.159 -0.025  0
##
## Model hyperparameters:
##                                mean      sd 0.025quant 0.5quant
## Precision for the Gaussian observations 1.01 0.132    0.767    1.00
##                                0.975quant   mode
## Precision for the Gaussian observations    1.28 0.992
##
## Expected number of effective parameters(stdev): 2.00(0.001)
## Number of equivalent replicates : 58.16
##
## Marginal log-Likelihood: -180.98
```

**3. Now infer the causal impact of groupsize. Which covariates do you need to adjust for? Looking at the posterior distribution of the resulting model, what do you think explains these data? That is, can you explain the estimates for all three problems? How do they go together?**

SOL: The variable groupsize does have a back-door path, passing through avgfood. So to infer the causal influence of groupsize, we need to close that path. This implies a model with both groupsize and avgfood as predictors.

It looks like group size is negatively associated with weight, controlling for food. Similarly, food is positively associated with weight, controlling for group size. So the causal influence of group size is to reduce weight—less food for each fox. And the direct causal influence of food is positive, of course. But the total causal influence of food is still nothing, since it causes larger groups. This is a masking effect, like in the milk energy example. But the causal explanation here is that more foxes move into a territory until the food available to each is no better than the food in a neighboring territory. Every territory ends up equally good/bad on average. This is known in behavioral ecology as an ideal free distribution.

```
# groupsize on weight
#need to include avgfood
m3 <- quap(
  alist(
    W ~ dnorm( mu , sigma ) ,
    mu <- a + bF*F + bG*G,
    a ~ dnorm( 0 , 0.2 ) ,
    bF ~ dnorm( 0 , 0.5 ) ,
    bG ~ dnorm( 0 , 0.5 ) ,
    sigma ~ dexp( 1 )
  ) , data=f )
```

```
precis(m3)
```

```
##              mean          sd        5.5%        94.5%
## a      2.426187e-09 0.08013807 -0.1280761  0.1280761
## bF      4.772531e-01 0.17912322  0.1909796  0.7635266
## bG     -5.735258e-01 0.17914172 -0.8598289 -0.2872228
## sigma  9.420439e-01 0.06175256  0.8433514  1.0407365
```

```
#using default priors
```

```
m3.inla <- inla(W~ F + G,family = c("gaussian"), data= f)
summary(m3.inla)
```

```
##
## Call:
##   inla(formula = W ~ F + G, family = c("\gaussian\"), data = f)"
## Time used:
##   Pre = 1.31, Running = 0.139, Post = 0.159, Total = 1.61
## Fixed effects:
##              mean          sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)  0.000 0.089      -0.174    0.000      0.174  0.000  0
## F              0.641 0.206       0.236    0.641      1.045  0.641  0
## G             -0.739 0.206      -1.144   -0.739     -0.335 -0.739  0
##
## Model hyperparameters:
##              mean          sd 0.025quant 0.5quant
## Precision for the Gaussian observations 1.11 0.147      0.845      1.11
##              0.975quant mode
## Precision for the Gaussian observations      1.42 1.09
##
## Expected number of effective parameters(stdev): 3.00(0.00)
## Number of equivalent replicates : 38.66
##
## Marginal log-Likelihood: -181.14
```

Prior mean for all fixed effects except the intercept. Alternatively, a named list with specific means where name=default applies to unmatched names. For example `control.fixed=list(mean=list(a=1, b=2, default=0))` assign 'mean=1' to fixed effect 'a', 'mean=2' to effect 'b' and 'mean=0' to all others. (default 0.0)

```
#inla.models()
```

```
#inla.set.control.fixed.default()
```

```
#using custom priors
```

```
m3.inla.prior <- inla(W~F+G, data= f, control.fixed = list(
  mean= list(F=0, G=0),
  prec= list(F=0.5, G=0.5),
  mean.intercept= 0,
  prec.intercept= 0.2
))
summary(m3.inla.prior)
```

```
##
## Call:
##   c("inla(formula = W ~ F + G, data = f, control.fixed = list(mean =
##   list(F = 0, ", " G = 0), prec = list(F = 0.5, G = 0.5), mean.intercept
##   = 0, ", " prec.intercept = 0.2))")
```

```

## Time used:
##      Pre = 1.34, Running = 0.147, Post = 0.158, Total = 1.64
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) 0.000 0.089   -0.174   0.000    0.174 0.000  0
## F           0.615 0.202    0.217   0.615    1.011 0.615  0
## G          -0.712 0.202   -1.108  -0.712   -0.315 -0.713  0
##
## Model hyperparameters:
##                               mean      sd 0.025quant 0.5quant
## Precision for the Gaussian observations 1.11 0.147    0.846    1.11
##                               0.975quant mode
## Precision for the Gaussian observations    1.42 1.10
##
## Expected number of effective parameters(stdev): 2.96(0.006)
## Number of equivalent replicates : 39.21
##
## Marginal log-Likelihood: -176.90

```