

CSP Problem Set 3

Nicolas Kepper

23 11 2021

Exercise 1

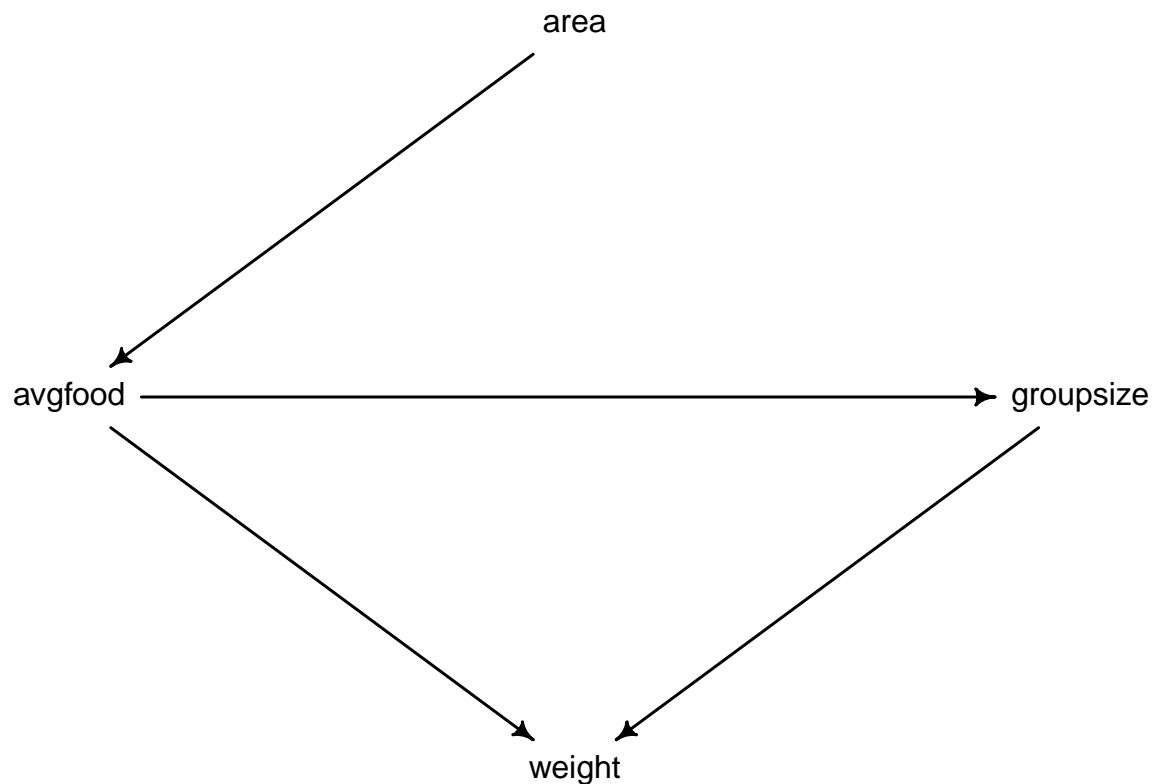
a)

Let's do some preparation first

```
data(foxes)
d <- foxes

d$AREA <- standardize(d$area)
d$AVGFOOD <- standardize(d$avgfood)
d$GROUPSIZE <- standardize(d$groupsize)
d$WEIGHT <- standardize(d$weight)

dag <- dagitty("dag{
    area -> avgfood;
    avgfood -> groupsize -> weight;
    avgfood -> weight
}")
coordinates(dag) <- list(x=c(area=1, avgfood=0, groupsize=2, weight=1)
    , y=c(area=0, avgfood=1, groupsize=1, weight=2))
drawdag(dag)
```



Inspecting the DAG we can see that there are no backdoor paths from area to weight. Thus, we can use area as the single predictor for weight in our model. We can use the following function to show us what variables we have to condition on in order to close backdoor paths if doing it manually by looking at the dag seems to be too much work.

```
adjustmentSets(dag, exposure="area", outcome="weight")
```

```
## {}
```

Well then, let's fit a model

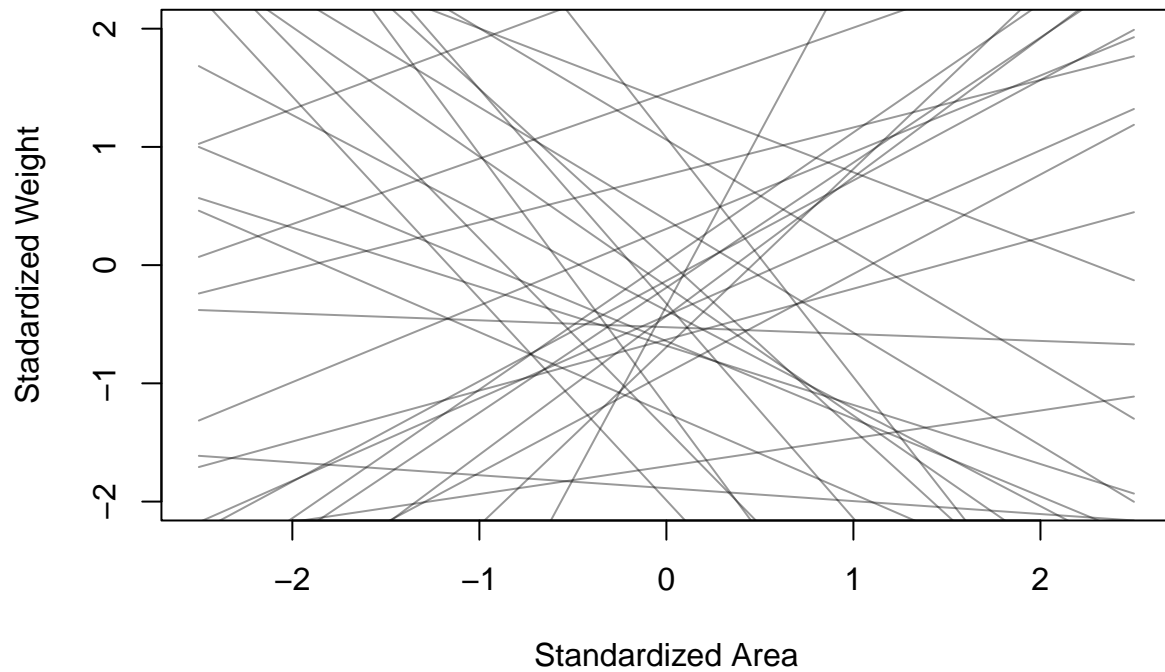
```
m_w_ar <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAR * AREA,
    a <- dnorm(0, 1),
    bAR <- dnorm(0, 1),
    sigma <- dexp(1)
  ), data=d
)
```

Let's have a look at potential regressions with these priors

```
prior <- extract.prior(m_w_ar)
area.seq <- seq(from=-2.5, to=2.5, length.out=30)

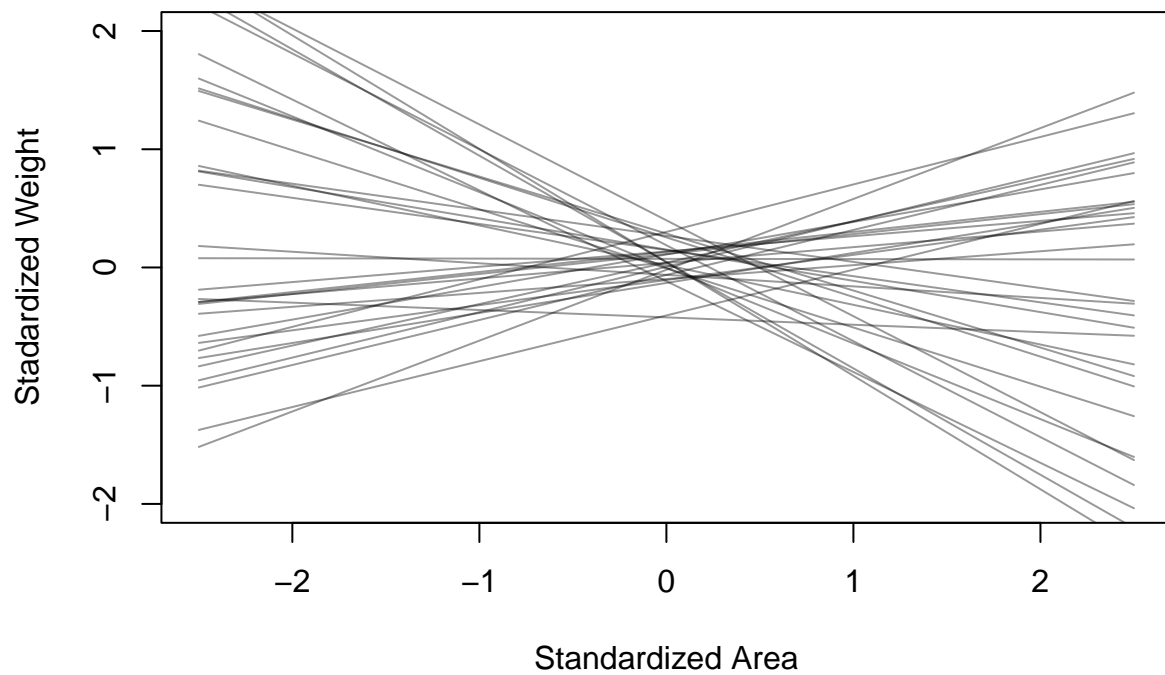
mu <- link(m_w_ar, post=prior, data=list(AREA=area.seq))
plot(NULL, xlim=range(area.seq), ylim=c(-2, 2), xlab="Standardized Area",
      ylab="Standardized Weight")

for(i in 1:30) lines(area.seq, mu[i, ], col=col.alpha("black", 0.4))
```



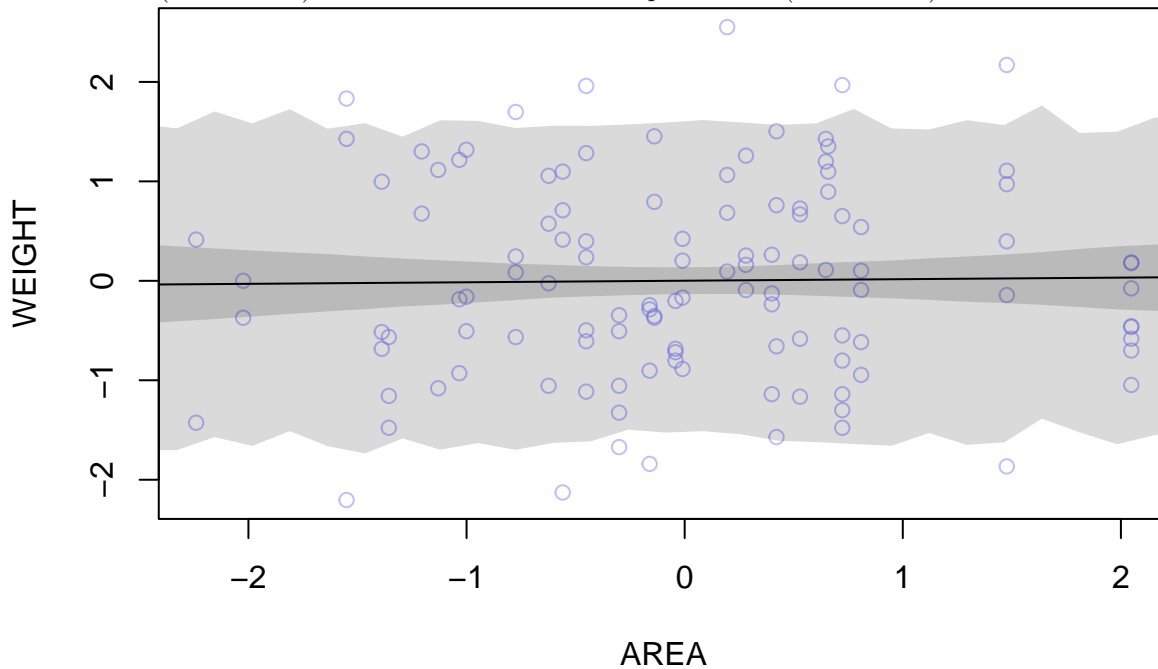
This looks a bit wild, so let's change the priors a little for them to better match reality

```
m_w_ar <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAR * AREA,
    a <- dnorm(0, 0.2),
    bAR <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)
```



Much better.

However, looking at `precis(m_w_ar)` as well as a little regression graph, we can see clearly that area size doesn't really have any casual influence on weight of the foxes. The shades in the graph show the 89% interval of the mean (inner shade) and the 89% interval of the predictions (outer shade).



```
##           mean      sd      5.5%    94.5%
## a      -9.750725e-08 0.08360865 -0.1336229 0.1336227
## bAR     1.883351e-02 0.09089579 -0.1264355 0.1641025
## sigma   9.912657e-01 0.06466643  0.8879163 1.0946152
```

b)

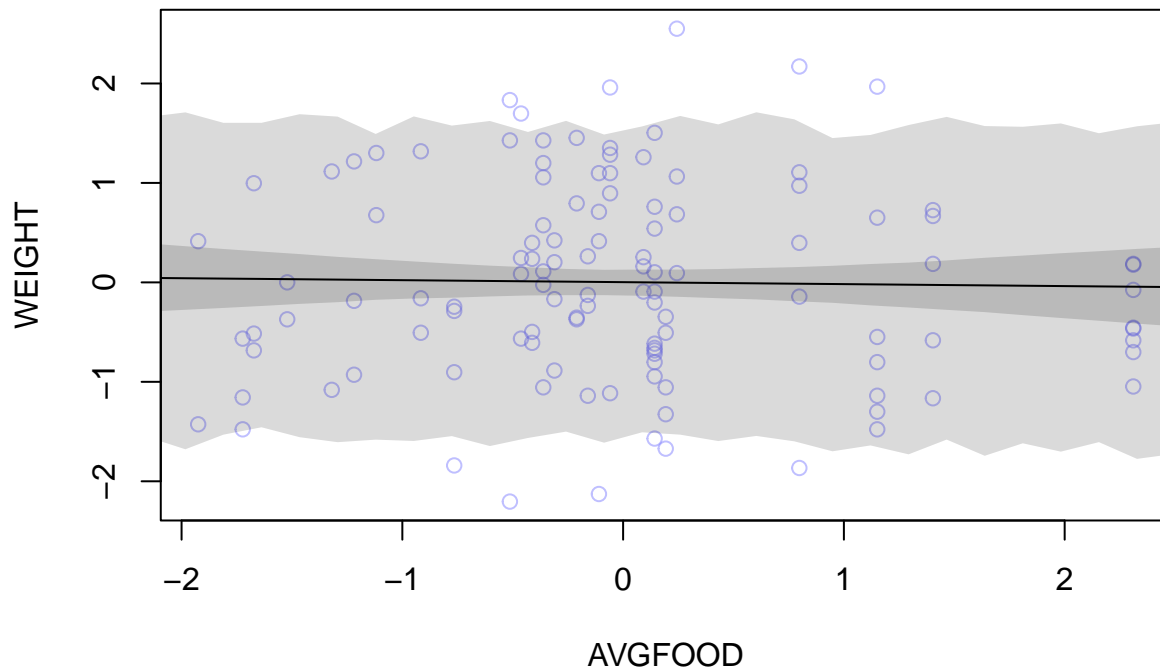
Once again, we need to check for backdoor paths first.

```
adjustmentSets(dag, exposure="avgfood", outcome="weight")
```

```
## {}
```

And once again we can see that there are no backdoor paths, so we again use a single predictor model. Including groupsize into the model would block the indirect path from avgfood to weight but then the result would not be the **total** casual influence. Then it would be the **direct** casual influence. “The effect of adding food” however means that we have to take all paths from food to weight into consideration.

```
m_w_af <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAF * AVGFOOD,
    a <- dnorm(0, 0.2),
    bAF <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)
```



```
##               mean      sd      5.5%    94.5%
## a      0.0001849539 0.08359533 -0.1334165 0.1337864
## bAF    -0.0241711949 0.09087885 -0.1694132 0.1210708
## sigma  0.9910743863 0.06464727  0.8877556 1.0943932
```

As we can see, we see nothing. This makes quite a bit of sense though, since - on our DAG - avgfood is directly influenced only by territory size, which evidently doesn't influence weight.

c)

Checking for backdoor paths again...

```
adjustmentSets(dag, exposure="groupsize", outcome="weight")
```

```
## { avgfood }
```

... we can see that we actually have to close a backdoor this time. So, let's build a model with groupsize and avgfood as predictors.

```
m_w_gs <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAF * AVGFOOD + bGS * GROUPSIZE,
    a <- dnorm(0, 0.2),
    bAF <- dnorm(0, 0.5),
    bGS <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)
```

Let's have a look at how each predictor influences the weight now.

##		mean	sd	5.5%	94.5%
## a		7.974701e-05	0.08014174	-0.1280022	0.1281617
## bAF		4.772998e-01	0.17913040	0.1910148	0.7635847
## bGS		-5.735674e-01	0.17914895	-0.8598820	-0.2872528
## sigma		9.420954e-01	0.06176098	0.8433894	1.0408014

Using this model, it seems that group size negatively impacts weight, while food positively impacts it. This makes sense. More foxes in a group with the same amount of food means every fox gets a smaller share and thus weighs less - more average food means every fox gets more food, so they weigh more. However, we didn't see a casual effect of neither area, nor avgfood on weight earlier so this looks a lot like a masking effect. What really happens when there's excess food is that foxes from other territories move here to get more food, increasing the group size but decreasing the avgfood. If there's too little food in one territory foxes will move to a different territory that has more food. So we can increase/decrease parameters arbitrarily, in reality the foxes will find an equilibrium balance. And this explains why the total casual influence of food is 0.

Exercise 2

Model selection is a process in which you look at different models using techniques like cross-validation or widely applicable information criterion and then **select** the model with the best score. The other models are completely discarded. In model comparison, on the other hand, we keep all models in order to compare how different variables influence the outcome. This also helps with understanding casual influences of the variables.

Under model selection we lose a lot of information about relative model accuracy. We lose the differences in CV/PSIS/WAIC values. And we also lose the possibility to choose a different model for a different goal, i.e. looking at the data from a slightly different point of view, focussing more on casual influences, etc.

Exercise 3

```
m_3_1 <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAF * AVGFOOD + bGS * GROUPSIZE + bAR * AREA,
    a <- dnorm(0, 0.2),
    c(bAF, bGS, bAR) <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)

m_3_2 <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAF * AVGFOOD + bGS * GROUPSIZE,
    a <- dnorm(0, 0.2),
    c(bAF, bGS) <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)

m_3_3 <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
```

```

    mu <- a + bAF * AVGFOOD + bAR * AREA,
    a <- dnorm(0, 0.2),
    c(bAF, bAR) <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)

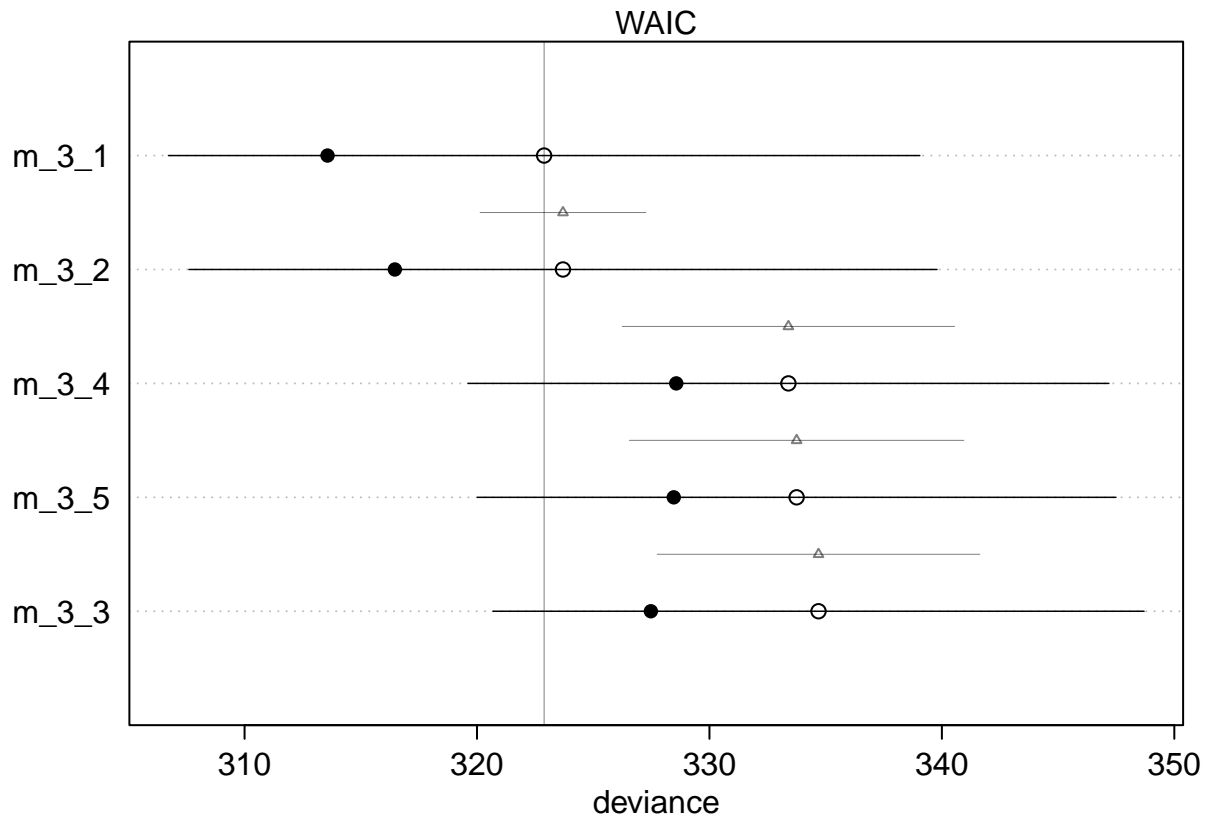
m_3_4 <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAF * AVGFOOD,
    a <- dnorm(0, 0.2),
    bAF <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)

m_3_5 <- quap(
  alist(
    WEIGHT ~ dnorm(mu, sigma),
    mu <- a + bAR * AREA,
    a <- dnorm(0, 0.2),
    bAR <- dnorm(0, 0.5),
    sigma <- dexp(1)
  ), data=d
)

compare(m_3_1, m_3_2, m_3_3, m_3_4, m_3_5, func=WAIC)

```

##		WAIC	SE	dWAIC	dSE	pWAIC	weight
##	m_3_1	323.6309	16.32855	0.0000000	NA	4.986664	0.562679913
##	m_3_2	324.1795	16.23732	0.5485723	3.617411	3.895542	0.427701173
##	m_3_4	333.5753	13.88450	9.9444074	7.171392	2.482712	0.003898170
##	m_3_5	333.8500	13.81930	10.2190943	7.225733	2.712409	0.003397921
##	m_3_3	334.6108	14.11326	10.9798560	6.961315	3.614222	0.002322822



We can kind of observe two groups looking at out-of-sample deviance. `m_3_1` and `m_3_2` seem almost identical while the other three build their own small group of similarness.

Let's talk about (4) and (5) first. They only use one predictor variable each - one uses area the other avgfood. Looking at the DAG we can recall, that they're nearly identical in isolation since all information of area has to pass through avgfood on its way to weight.

We can use the same reasoning to explain why (3) is in the same group. This model uses the two predictors area and avgfood which - as I just said - are basically the same thing.

Similarly, it doesn't matter whether we use all three predictors or just groupsize and avgfood, because they essentially say the same thing due to area being fully routed through avgfood.

However, adding groupsize to the model does make the difference of closing a backdoor, or rather splitting up the total casual influence into two direct influences, which is why the two models that do use groupsize as a predictor perform better than the other three models.