

Assignment 3

Jasmin Capka

11/21/2021

Exercise 1

Loading and inspecting the data:

```
data(foxes)
d <- foxes
precis(d)
```

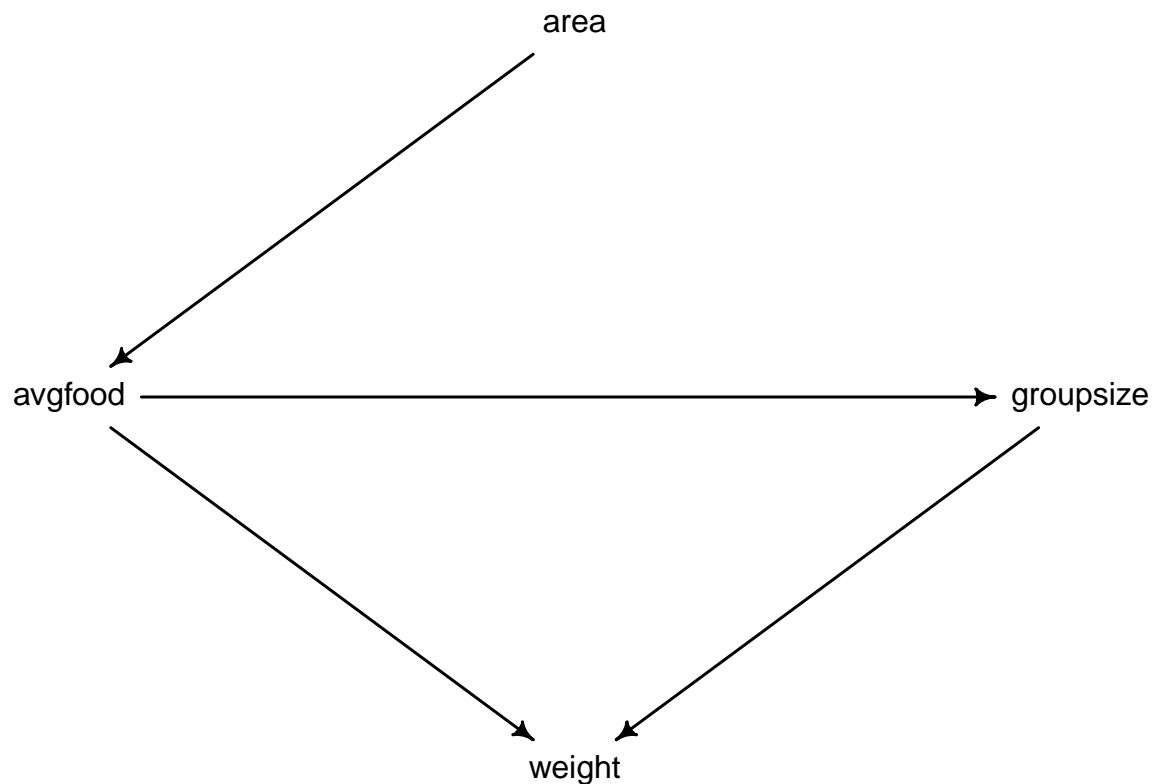
##	mean	sd	5.5%	94.5%	histogram
## group	17.2068966	8.0027357	4.00000	28.6750	
## avgfood	0.7517241	0.1983158	0.42000	1.2100	
## groupsize	4.3448276	1.5385111	2.00000	8.0000	
## area	3.1691379	0.9283539	1.77875	5.0700	
## weight	4.5296552	1.1840226	2.78000	6.2905	

Given causal DAG:

```
# define DAG
dag_foxes <- dagitty("dag{
  area -> avgfood
  avgfood -> groupsize -> weight
  avgfood -> weight
}")

# set coordinates
coordinates(dag_foxes) <- list(x = c(area = 1, avgfood = 0, groupsize = 2, weight = 1),
                               y = c(area = 0, avgfood = 1, groupsize = 1, weight = 2))

# draw
drawdag(dag_foxes)
```



Question 1.a):

The first step for all following models is standardizing all variables:

```
# standardize data
d$area_std = standardize(d$area)
d$avgfood_std = standardize(d$avgfood)
d$weight_std = standardize(d$weight)
d$groupsize_std = standardize(d$groupsize)
```

In order to construct a quap model to infer the total causal influence of area on weight, I have to check whether there are covariates I need to adjust:

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

```
## {}
```

There are no variables I need to adjust for as the code above shows. This makes sense as area has two indirect paths linking it to weight and no other variables are influencing area. Therefore, both indirect paths need to be included when modeling the total causal influence of area on weight and no backdoors need to be closed.

In the next step I use prior predictive simulation to show that my model's predictions will stay within the possible outcome range. As all variables are standardized with a mean of 0 and a standard deviation of 1, I use the following priors:

$$\begin{aligned}
 weight_i &\sim Normal(\mu_i, \sigma) \\
 \mu_i &= \alpha + \beta area_std_i \\
 \alpha &\sim Normal(0, 0.5)
 \end{aligned}$$

$$\beta \sim \text{Normal}(0, 0.5)$$

$$\sigma \sim \text{Log} - \text{Normal}(0, 1)$$

Using prior predictive simulation from 4.3.2 (p.82) of the book with a density plot, shows that most of the predictions stay within the possible outcomes, meaning the minimal value of weight (displayed via the green line) and the max value of weight (displayed via the red line):

```
N = 116

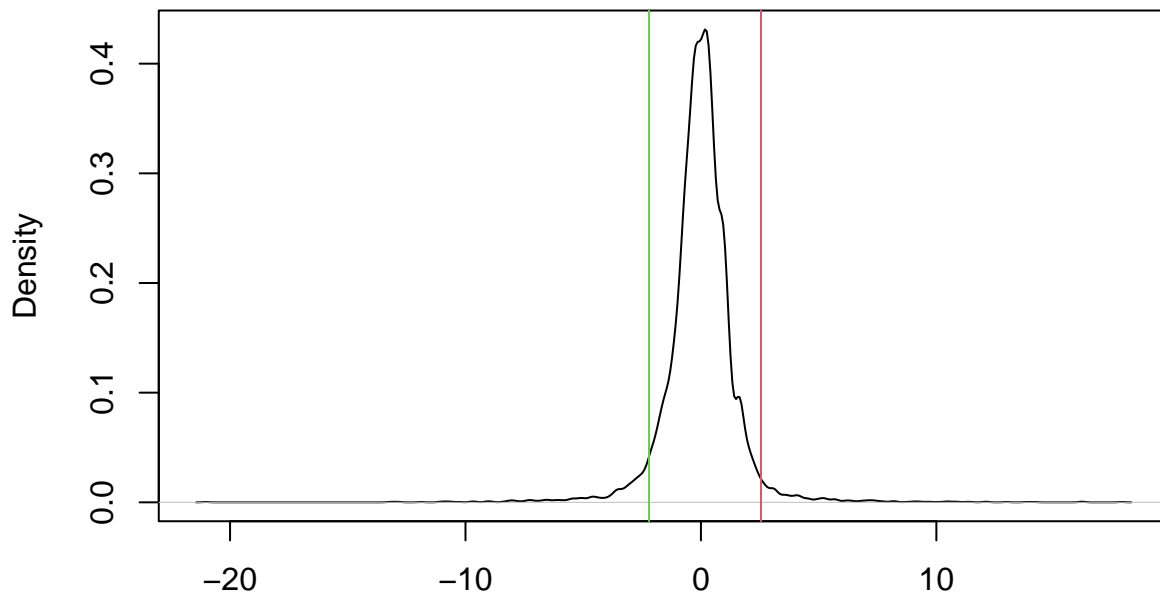
# define distribution of a and bAR
a <- rnorm(N, 0, 0.5)
bAR <- rnorm(N, 0, 0.5)

# sigma and mu
sample_sigma <- rexp(N, 1)
sample_mu <- a + bAR * d$area_std

# plot prior predictive simulation
prior_w <- rnorm(1e4, sample_mu, sample_sigma)
plot(density(prior_w))

# plot min and max line
abline(v = max(d$weight_std), col = 2)
abline(v = min(d$weight_std), col = 3)
```

density.default(x = prior_w)



N = 10000 Bandwidth = 0.1351

Additionally, using prior predictive simulation from page 95 of the book, with individual lines of possible models shows as well that most of the predictions stay within possible outcomes.

```
# create graph
plot(NULL, xlim = range(d$area_std), ylim = range(-3, 3),
```

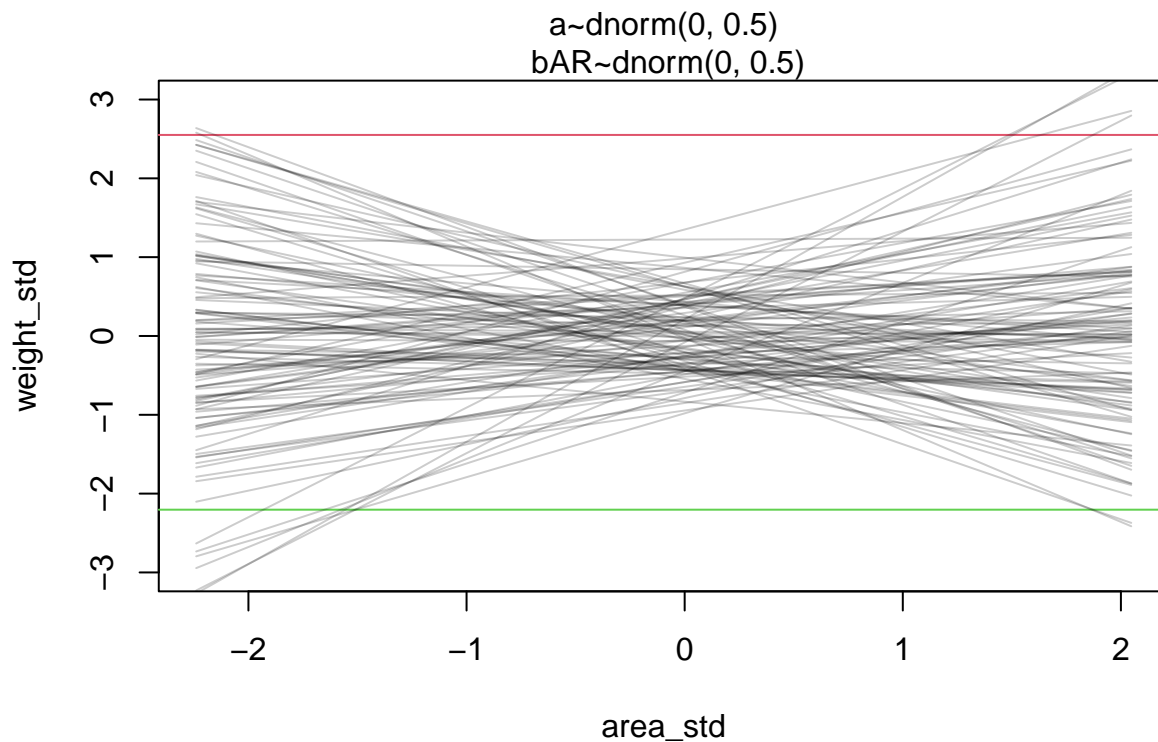
```

      xlab = "area_std", ylab = "weight_std")

# plot min and max line
abline(h = max(d$weight_std), col = 2)
abline(h = min(d$weight_std), col = 3)

# plot individual lines
mtext("a~dnorm(0, 0.5)\n bAR~dnorm(0, 0.5)")
for (i in 1:N) curve((a[i] + bAR[i] * x),
                      from = min(d$area_std), to = max(d$area_std),
                      add = TRUE,
                      col = col.alpha("black", 0.2))

```



Applying those priors, the model can be constructed:

```

m1a <- quap(
  alist(
    weight_std ~ dnorm(mu, sigma),
    mu <- a + bAR * area_std,
    a ~ dnorm(0, 1),
    bAR ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data = d)

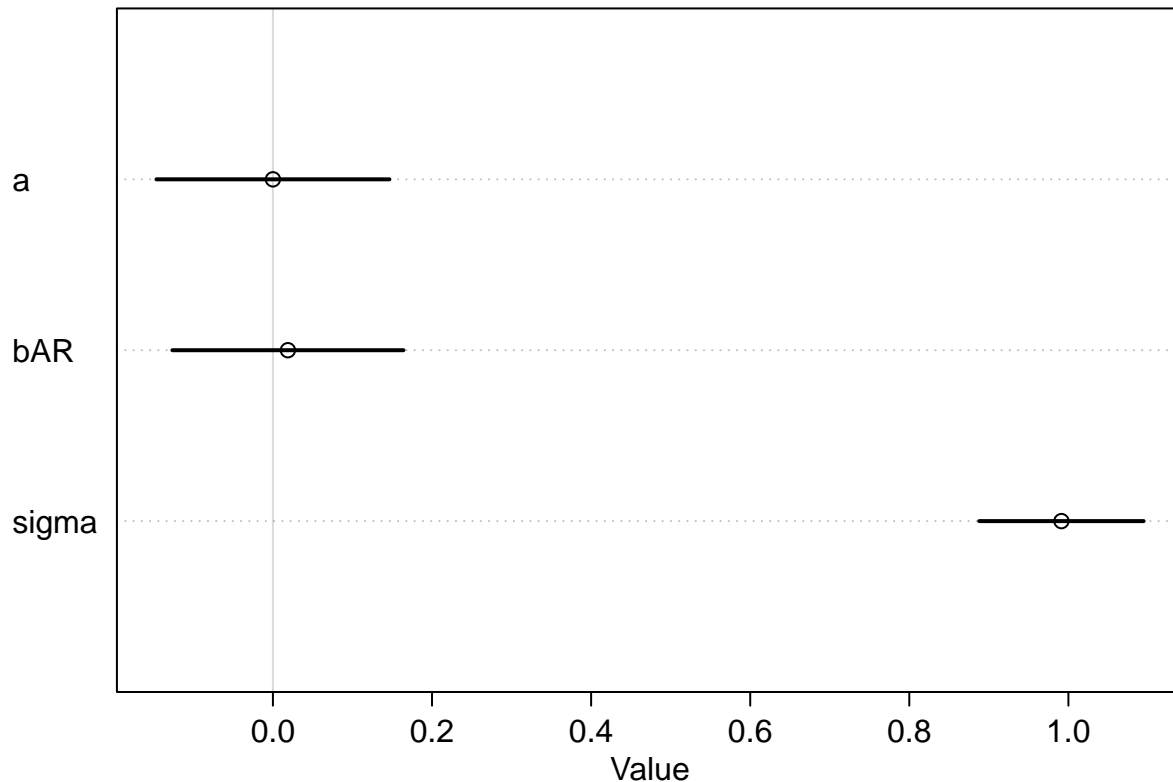
precis(m1a, depth = 2)

```

##		mean	sd	5.5%	94.5%
##	a	-5.277651e-07	0.09164934	-0.1464739	0.1464728
##	bAR	1.883364e-02	0.09089577	-0.1264354	0.1641026
##	sigma	9.912655e-01	0.06466639	0.8879161	1.0946149

Looking at this outcome, territory size (area) seems to have no causal influence on the weight of foxes. The mean of bAR is slightly above zero, but looking at the standard deviation and the percentile intervals, bAR is distributed above 0 as well as below. This can be seen in the following plot as well:

```
plot(precis(m1a, depth = 2))
```



Therefore, increasing the area available to each fox has no visible influence on its health (weight).

Question 1.b):

To estimate the total causal influence of food no covariates need to be adjusted as well. This code below shows this as well:

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

```
## {}
```

There are two paths connecting avgfood with weight. The first one is a direct path. The second path 'avgfood -> groupsize -> weight' is a pipe but, nevertheless, acts as an indirect causal effect for avgfood on weight. Therefore, with the same explanation as for the model in 1.a), when estimating the total causal effect of avgfood on weight, it should be included.

The following model can be constructed (using the same priors as for m1a because of standardization):

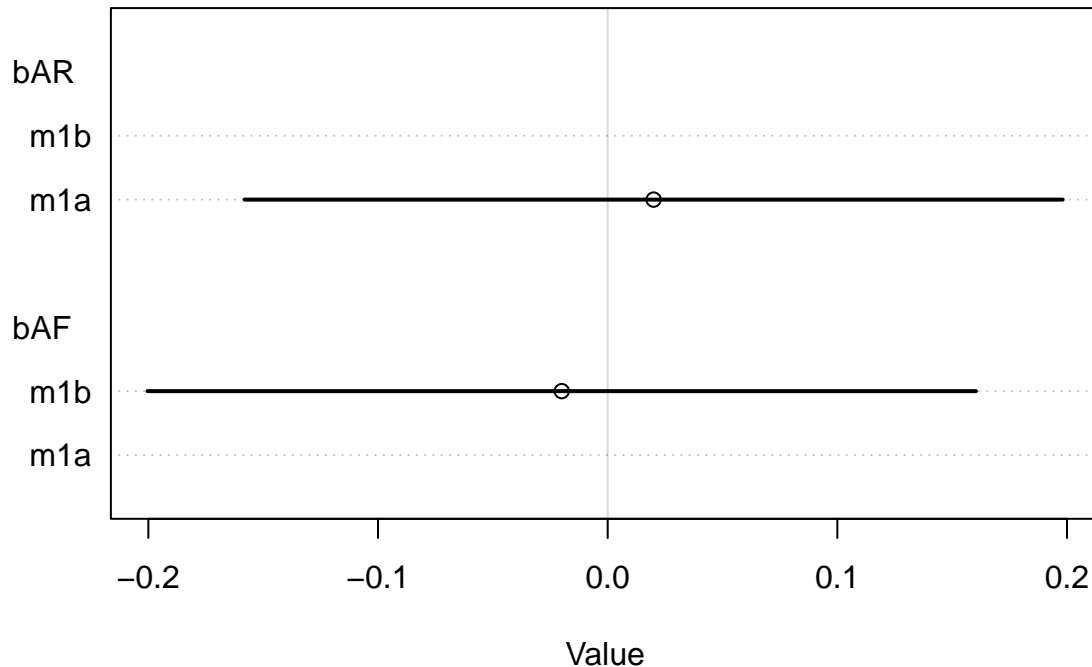
```
m1b <- quap(
  alist(
    weight_std ~ dnorm(mu, sigma),
    mu <- a + bAF * avgfood_std,
    a ~ dnorm(0, 1),
    bAF ~ dnorm(0, 1),
    sigma ~ dexp(1)
  ), data = d)
```

```
precis(m1b, depth = 2)
```

```
##           mean          sd      5.5%      94.5%
## a      -2.661338e-06 0.09163490 -0.1464529 0.1464476
## bAF    -2.482399e-02 0.09202909 -0.1719043 0.1222563
## sigma  9.911080e-01 0.06465272  0.8877804 1.0944355
```

The distribution of bAF is almost identical to the negative distribution of bAR before. This can be seen in the plot below, as well:

```
plot(coeftab(m1a, m1b), par = c("bAR", "bAF"))
```



This means, that adding food (avgfood) will not make foxes heavier. Avgfood seems to have no visible impact on weight as well.

Question 1.c):

To estimate the causal influence of groupsize on weight, there are covariates that need to be adjusted:

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

```
## { avgfood }
```

The model needs to be conditioned on avgfood. This is because the path 'groupsize <- avgfood -> weight' is a fork. Conditioning on avgfood closes the backdoor.

The following model can be constructed (using the same priors as for m1a because of standardization):

```
m1c <- quap(
  alist(
    weight_std ~ dnorm(mu, sigma),
    mu <- a + bAF * avgfood_std + bG * groupsize_std,
    a ~ dnorm(0, 1),
    bAF ~ dnorm(0, 1),
    bG ~ dnorm(0, 1),
```

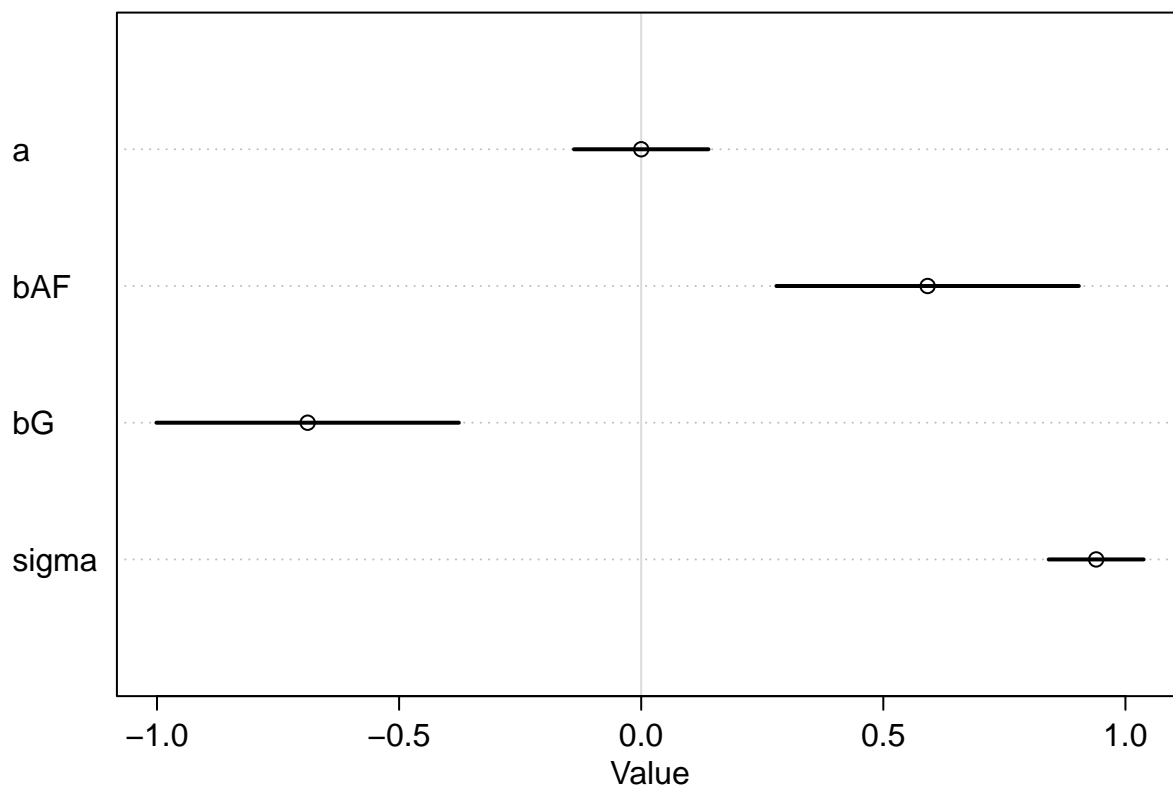
```
sigma ~ dexp(1)
), data = d)

precis(m1c, depth = 2)
```

##		mean	sd	5.5%	94.5%
## a		9.807250e-06	0.08690759	-0.1388853	0.1389049
## bAF		5.914564e-01	0.19542485	0.2791297	0.9037830
## bG		-6.888734e-01	0.19542653	-1.0012027	-0.3765440
## sigma		9.395784e-01	0.06135160	0.8415267	1.0376301

Looking at the posterior distribution, groupsize seems to have a negative effect on weight, because bG has a value of -0.69. Looking at the plot below, even considering the standard deviation, bG is clearly negative when conditioning on avgfood.

```
plot(precis(m1c, depth = 2))
```



Therefore, increasing groupsize decreases weight. This makes sense, because in a larger group of foxes a single fox will get less food, keeping the avgfood constant.

Additionally, bAF is now positive. Thus, increasing the average food seems to increase the weight of a fox, if group size is kept constant. This makes sense as well, when more food is available, a fox can eat more and gain weight, but it contradicts the result of m1b. In m1b, avgfood had no significant effect on weight. This effect arises, because (as can be seen in the DAG) avgfood also has an influence on groupsize. The more food is available, the larger becomes a group, which decreases weight.

This means, in general, the direct effect of avgfood on weight is to increase weight as can be seen in model m1c. Nevertheless, it also affects groupsize and decreases weight over that indirect path. This results in a non-significant total effect from model m1b. This effect arises, because groupsize is a post-treatment variable. The variable groupsize is a consequence of avgfood.

Furthermore, considering the estimates of the effects of area and avgfood from the models before and looking at the data, area and avgfood seem to contain the same information. They both have a similar effect on weight and area has a direct influence on avgfood. This is because an increase in the territory of a group means that average amount of food increases as well.

Exercise 2

The book provides information about both terms on page 226:

Model selection is the process of picking the model with the lowest criterion value compared to other models. It only focuses on that single selected model and pays no attention to the other possible models anymore. Model comparison in contrast means using multiple models to better understand how included variables influence predictions and detect causal relationships through implied conditional independencies in a causal model.

Therefore, under model selection, the information about the relative model accuracy gets lost. To get this relative model accuracy other models and their different WAIC/CV/PSIS values are needed. The relative model accuracy indicates how confident one can be about a model conditional on a set of models that are compared. With this information included, model comparison is a more holistic approach allowing to distinguish between different models and their results as well as implications on the initial data.

Exercise 3

In the first step, all five models are constructed. Model 2, 4 and 5 can be taken from exercise 1.

```
m3.1 <- quap(
  alist(
    weight_std ~ dnorm(mu, sigma),
    mu <- a +
      bAF * avgfood_std +
      bG * groupsize_std +
      bAR * area_std,
    a ~ dnorm(0, 1),
    bAF ~ dnorm(0, 1),
    bG ~ dnorm(0, 1),
    bAR ~ dnorm(0, 1),
    sigma ~ dexp(1)
  ), data = d)

m3.2 = m1c

m3.3 <- quap(
  alist(
    weight_std ~ dnorm(mu, sigma),
    mu <- a + bAF * avgfood_std + bAR * area_std,
    a ~ dnorm(0, 1),
    bAF ~ dnorm(0, 1),
    bAR ~ dnorm(0, 1),
    sigma ~ dexp(1)
  ), data = d)

m3.4 = m1b

m3.5 = m1a
```

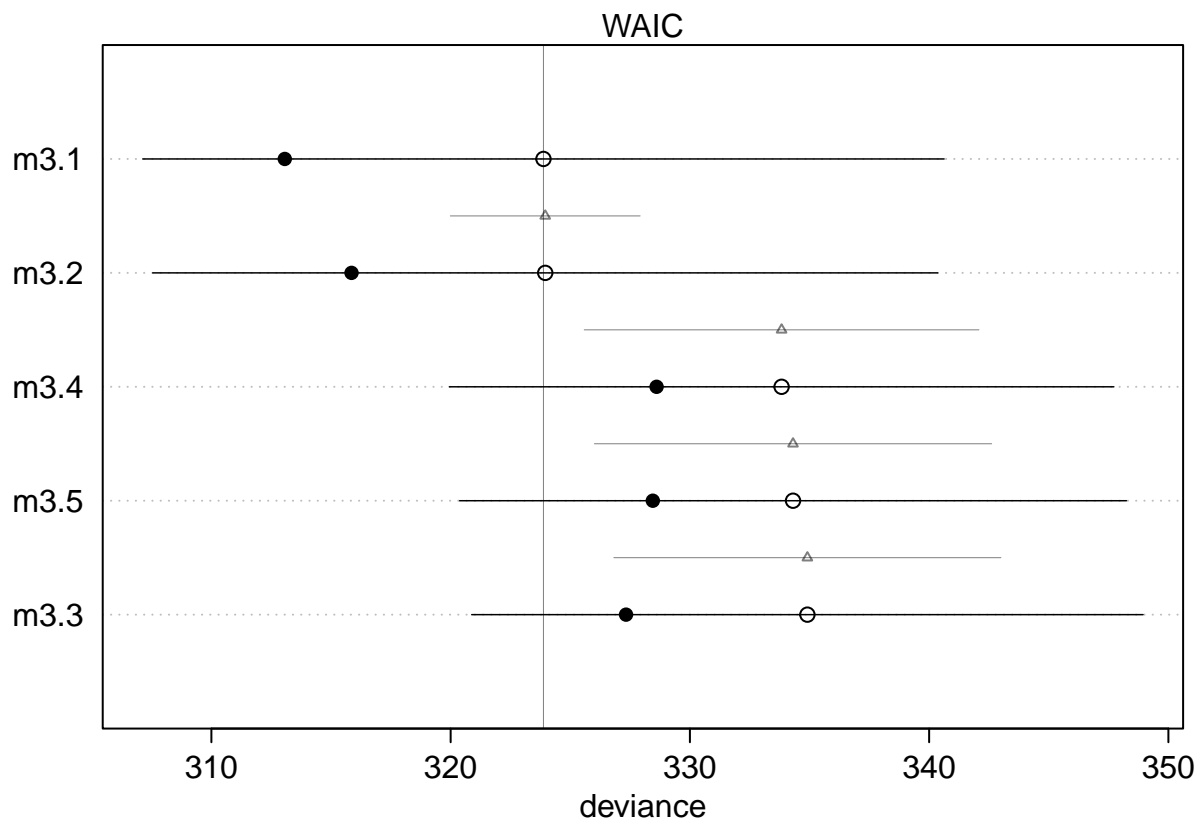

Then I use the WAIC comparison:

```
set.seed(11)
compare(m3.1, m3.2, m3.3, m3.4, m3.5, func = WAIC)
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
## m3.1	323.4903	16.70205	0.0000000	NA	5.250546	0.563862375
## m3.2	324.0364	16.59146	0.5461525	3.829789	4.148096	0.429118868
## m3.5	334.0725	13.81297	10.5822334	8.267835	2.822805	0.002839686
## m3.4	334.1924	13.86955	10.7020949	8.227563	2.777439	0.002674501
## m3.3	335.3429	14.22137	11.8526066	8.111745	3.996276	0.001504570

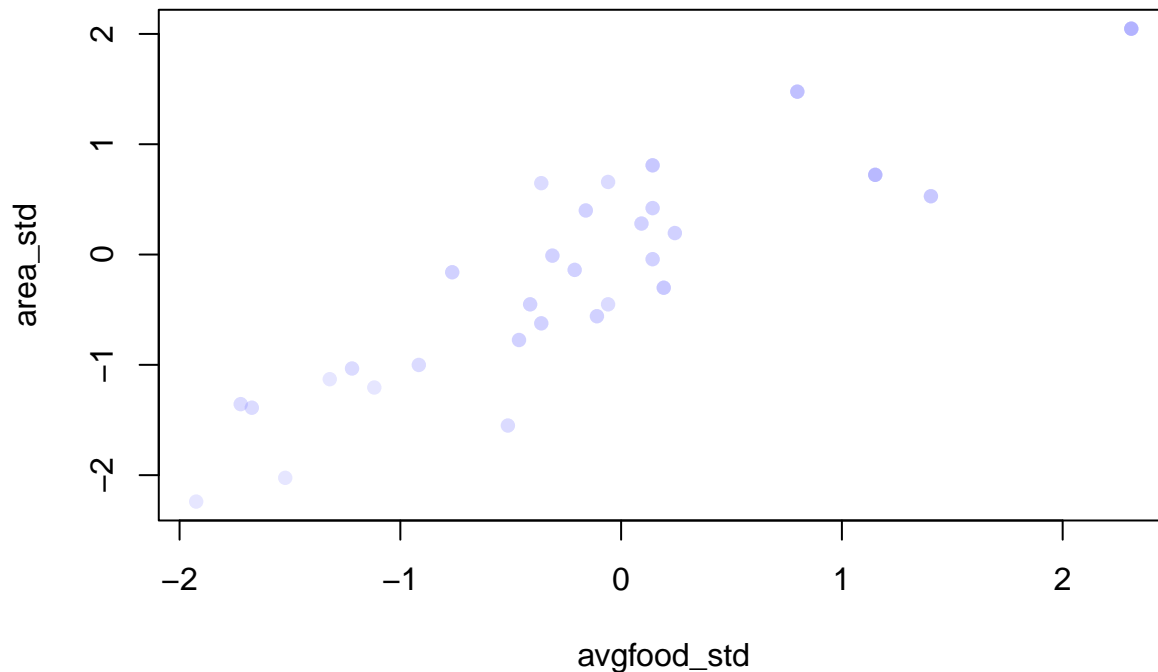
Looking at the differences in WAIC scores, the first two models and the last three models are pretty similar. This gets even more obvious when looking at the graph below.

```
plot(compare(m3.1, m3.2, m3.3, m3.4, m3.5, func = WAIC))
```



The difference in WAIC scores between m3.1 and m3.2 is very small with 0.5. Taking the standard error of this difference, which has a value of 3.83 into consideration, they perform similarly. This can be explained with the DAG above, because as area only directly influences avgfood, no further information is added, when adding area to the model. The two variables are strongly correlated and nearly redundant as can be seen in the plot below. Therefore, both models can be treated equally regarding their accuracy.

```
plot(area_std ~ avgfood_std, d, col = col.alpha(rangi2, 0.1), pch = 16)
```



The models m3.4 and m3.5 are about 11 units of deviance smaller model m3.1. They are still having a standard error of this difference of around 8.25. These models also perform similarly, because as mentioned before area and avgfood contain the same information. These two models perform worse than models m3.1 and m3.2 before, because the information about groupsize is missing. Groupsize has a significant negative impact on weight and is able to add further information as can be seen from the DAG, because it has a direct impact on weight.

Model m3.3 has a slightly higher difference in WAIC with the value of 12 than the models m3.4 and m3.5, but an equal standard error in this difference. Again the information about groupsize is missing resulting in a similar accuracy as models m3.4 and m3.5. This models contains the redundant variables avgfood and area. Model m3.3 has a difference in WAIC of 1.2 to model m3.4 and of value 1.3 to model m3.5. Nevertheless, looking at the table below, the respective errors are 2.26 and 2.08 for models m3.4 and m3.5. Therefore, these three models can be grouped together which is also supported by the WAIC comparison graph.

```
compare(m3.1, m3.2, m3.3, m3.4, m3.5, func = WAIC)@dSE
```

	m3.1	m3.2	m3.3	m3.4	m3.5
m3.1	NA	3.848323	8.124635	8.2517688	8.3163713
m3.2	3.848323	NA	8.814267	7.9765935	8.1664188
m3.3	8.124635	8.814267	NA	2.2677082	2.0946232
m3.4	8.251769	7.976594	2.267708	NA	0.9343175
m3.5	8.316371	8.166419	2.094623	0.9343175	NA

In conclusion, this split in two groups of models results from the inclusion of groupsize as variable in the first group of the models and omission of it in the second group.