Methods 4 – Portfolio Assignment 1

Pandemic Exercises

```
pacman::p_load(ggplot2, dplyr, purrr, tidyr, rethinking, dagitty)
```

1) Testing Efficiency

Imagine there was a global pandemic.

It’s a bit difficult, I know.

Maybe a new version of the old SARS-CoV turns out to be really infectious, or something like that.

A test is developed that is cheap and quick to use, and the government asks you to determine its efficiency.

To do this, they find X people that they know for sure are infected, and X people that they know for sure are not infected. *NB: This is not always possible. For example, there is an ongoing global pandemic in the real world - maybe you heard of it -where a 100% sure test doesn’t exist, as far as I know. But let’s ignore that. The government finds a wizard who can tell for sure, but he wants a lot of money and he’s really slow too.*

Okay, so X infected people take the test, and X uninfected people take the test. See the results below. P means positive, N means negative.

- Infected:

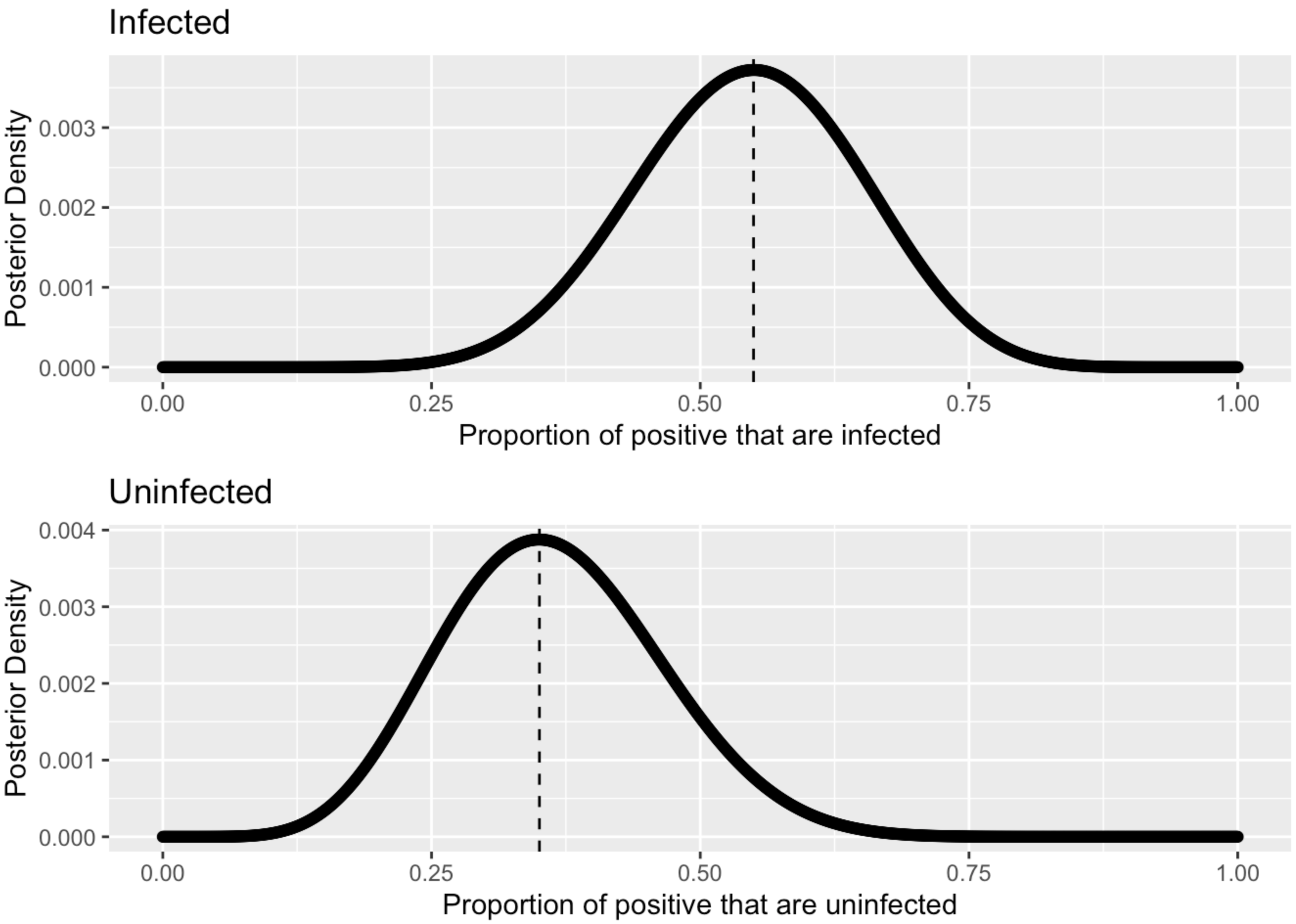
$$P, N, P, P, N, P, P, N, N, N, P, P, N, P, P, N, N, P, N, P$$

- Uninfected:

$$P, N, N, P, N, P, P, N, N, N, P, N, N, N, N, P, P, N, N, N$$

A) (SM) Estimate the probabilities of testing positive given that you’re infected, and given that you’re not infected. Use the grid approximation method as in the book. Use a prior you can defend using. Report the full posterior probability distribution for each case (we can do better than just a single value!).

```
# Infected
infected <- tibble(p_grid = seq(from = 0, to = 1, length.out = 1000), # Building the
grid
               prior = rep(1, times = 1000)) %>% # Setting the prior
              mutate(likelihood = dbinom(11, size = 20, prob = p_grid))

infected <- infected %>% mutate(unstd_posterior = likelihood * prior)
infected <- infected %>% mutate(posterior = unstd_posterior / sum(unstd_posterior))


infected_plot <-ggplot(infected, aes(x = p_grid, y = posterior)) +
  geom_line() +
  geom_point() +
  labs(x = "Proportion of positive that are infected", y = "Posterior Density")+ geom
_vline(aes(xintercept = p_grid[which.max(posterior)]),
             linetype="dashed") + ggtitle("Infected")

# Uninfected
uninfected <- tibble(p_grid = seq(from = 0, to = 1, length.out = 1000), # Building th
e grid
               prior = rep(1, times = 1000)) %>% # Setting the prior
              mutate(likelihood = dbinom(7, size = 20, prob = p_grid))

uninfected <- uninfected %>% mutate(unstd_posterior = likelihood * prior)
uninfected <- uninfected %>% mutate(posterior = unstd_posterior / sum(unstd_posterio
r))


uninfected_plot <- ggplot(uninfected, aes(x = p_grid, y = posterior)) +
  geom_line() +
  geom_point() +
  labs(x = "Proportion of positive that are uninfected", y = "Posterior Density")+ ge
om_vline(aes(xintercept = p_grid[which.max(posterior)]),
             linetype="dashed") + ggtitle("Uninfected")

gridExtra::grid.arrange(infected_plot, uninfected_plot)
```



We chose the prior to be 1, which is a neutral prior that will not change the posterior (posterior = likelihood * prior). Therefore, the posterior is only based on the likelihood. This decision is made, since we are given no previous knowledge on the topic.

```
# Finding the maximum posterior probability of the proportion of positive for the inf
ected
infected$p_grid[which.max(infected$posterior)]
```

```
## [1] 0.5495495
```

```
# Finding the maximum posterior probability of the proportion of positive for the uni
nfected
uninfected$p_grid[which.max(uninfected$posterior)]
```

```
## [1] 0.3503504
```

B) (LR) The government says that they find probability distributions difficult to use. They ask you to provide them with a confidence interval of 95% within which the ‘real’ probability can be found. Do it.

```
# Confidence interval of 95 % for infected
samples_infected <- sample( infected$p_grid , prob=infected$posterior , size=1e4 , replace=TRUE )

PI(samples_infected, prob = 0.95)
```

```
##        3%       98% 
## 0.3343093 0.7437688
```

```
# Confidence interval of 95 % for uninfected
samples_uninfected <- sample( uninfected$p_grid , prob=uninfected$posterior , size=1e4 , replace=TRUE )

PI(samples_uninfected, prob = 0.95)
```

```
##        3%       98% 
## 0.1811562 0.5695696
```

C) (MST) The government says that their voters find confidence intervals difficult to read. In addition, they are so wide that it looks like the government doesn't know what they're doing. They want a point estimate instead. Give them one.

```
infected_loss <- sapply( infected$p_grid , function(d) sum( infected$posterior*abs( d - infected$p_grid ) ) )

infected$p_grid[ which.min(infected_loss) ]
```

```
## [1] 0.5465465
```

```
uninfected_loss <- sapply( uninfected$p_grid , function(d) sum( uninfected$posterior*abs( d - uninfected$p_grid ) ) )

uninfected$p_grid[ which.min(uninfected_loss) ]
```

```
## [1] 0.3593594
```

The point estimate of testing positive given that you are infected is 54.7 % using a loss function. The point estimate of testing positive given that you are uninfected is 36 % using a loss function.

2) Dark Cellars

Months pass. Thousands of people are tested by the wizards of the world governments. A fancy company analyses the data, and determine, with very high confidence they say, the probability of testing positive with the current test. They give the following point estimates:

- A 53% chance of testing positive if you are infected.

- A 45% chance of testing positive if you are not infected.

NB: These numbers also happen to be real estimates for the efficiency of the COVID kviktest[1]. Remember that the actual Danish government doesn't have any wizards, though.

A) (TI) You are sitting in your dark cellar room, writing an apology to the Danish government, when you receive a positive test result on your phone. Oh, that party last weekend. In order to fight the boredom of isolation life, you start doing statistical inference. Using Bayes theorem, estimate the probability that you are infected, given that it is *a priori* equally likely to be infected or not to be.

```
pr_positive_infected <- 0.53
pr_positive_uninfected <- 0.45
prior_infected <- 0.5
pr_positive <- pr_positive_infected * prior_infected +
               pr_positive_uninfected * ( 1 - prior_infected )

(prob_infected <- pr_positive_infected*prior_infected / pr_positive)
```

```
## [1] 0.5408163
```

```
# Or in a more intuitive way
53/(45+53) # number of real positives divided by all the positives (including false p
ositives). This is of course only possible since the prior is 0.5
```

```
## [1] 0.5408163
```

The probability that you are infected, given the prior, is 54 %.

B) (SM) A quick Google search tells you that about 546.000[2] people in Denmark are infected right now. Use this for a prior instead.

```
# People in Denmark according to google = 5.850.000
prior_infected_b <- 546000/5850000 # What percentage of the population is 546.000?
pr_positive_b <- pr_positive_infected * prior_infected_b +
               pr_positive_uninfected * ( 1 - prior_infected_b )

( prob_infected_b <- pr_positive_infected*prior_infected_b / pr_positive_b )
```

```
## [1] 0.1081317
```

The probability that the positive test is accurate is 10.8 % given the new prior.

C) (LR) A friend calls and says that they have been determined by a wizard to be infected. You and your friend danced tango together at the party last weekend. It has been estimated that dancing tango with an infected person leads to an infection 32% of the time[3]. Use this information to construct a prior instead.

```
prior_infected_c <- 0.32

pr_positive_c <- pr_positive_infected * prior_infected_c +
               pr_positive_uninfected * ( 1 - prior_infected_c )

( prob_infected_c <- pr_positive_infected*prior_infected_c / pr_positive_c )
```

```
## [1] 0.3566022
```

The probability that the positive test is accurate is 35.7 % given the new prior.

D) (MST) You quickly run and get two more tests. One is negative, the other positive. Update your estimate.

```
#We already that the probability when we have one test, which is now our new prior
prior_infected_d <- prob_infected_c

#We know get a negative test. We now want to look at the probabilities that the test
is negative
pr_negative_infected <- 0.47
pr_negative_uninfected <- 0.55

pr_negative_d <- pr_negative_infected * prior_infected_d +
               pr_negative_uninfected * ( 1 - prior_infected_d )

( prob_infected_d <- pr_negative_infected*prior_infected_d / pr_negative_d )
```

```
## [1] 0.3214038
```

```
#We use this probability as our prior for the third positive test
prior_infected_e <- prob_infected_d

pr_positive_e <- pr_positive_infected * prior_infected_e +
               pr_positive_uninfected * ( 1 - prior_infected_e )

( prob_infected_e <- pr_positive_infected*prior_infected_e / pr_positive_e )
```

```
## [1] 0.358082
```

35 % probability that you are actually infected given the three tests

E) (TI) In a questionnaire someone sent out for their exam project, you have to answer if you think you are infected. You can only answer yes or no (a bit like making a point estimate). What do you answer?

```
if(prob_infected_e>50){
print("Yes")
} else {
print("No")
}
```

```
## [1] "No"
```

Since our point estimate is 35 %, it is most likely that we are not infected and the answer would be “No”

F) (SM & LR) You are invited to a party. They ask if you are infected. They also think that if you are in doubt, staying home is safer. Because they studied statistics, they formulate this as preferring that you used an asymmetric loss function when making your decision: it is three times worse to falsely answer not infected and come sick to the party, as compared to to falsely answering infected and staying home while healthy. What do you answer?

```
prob_not_infected_e <- 1-prob_infected_e

if(3*prob_infected_e > prob_not_infected_e){
print("You are infected")
} else {
print("You are not infected")
}
```

```
## [1] "You are infected"
```

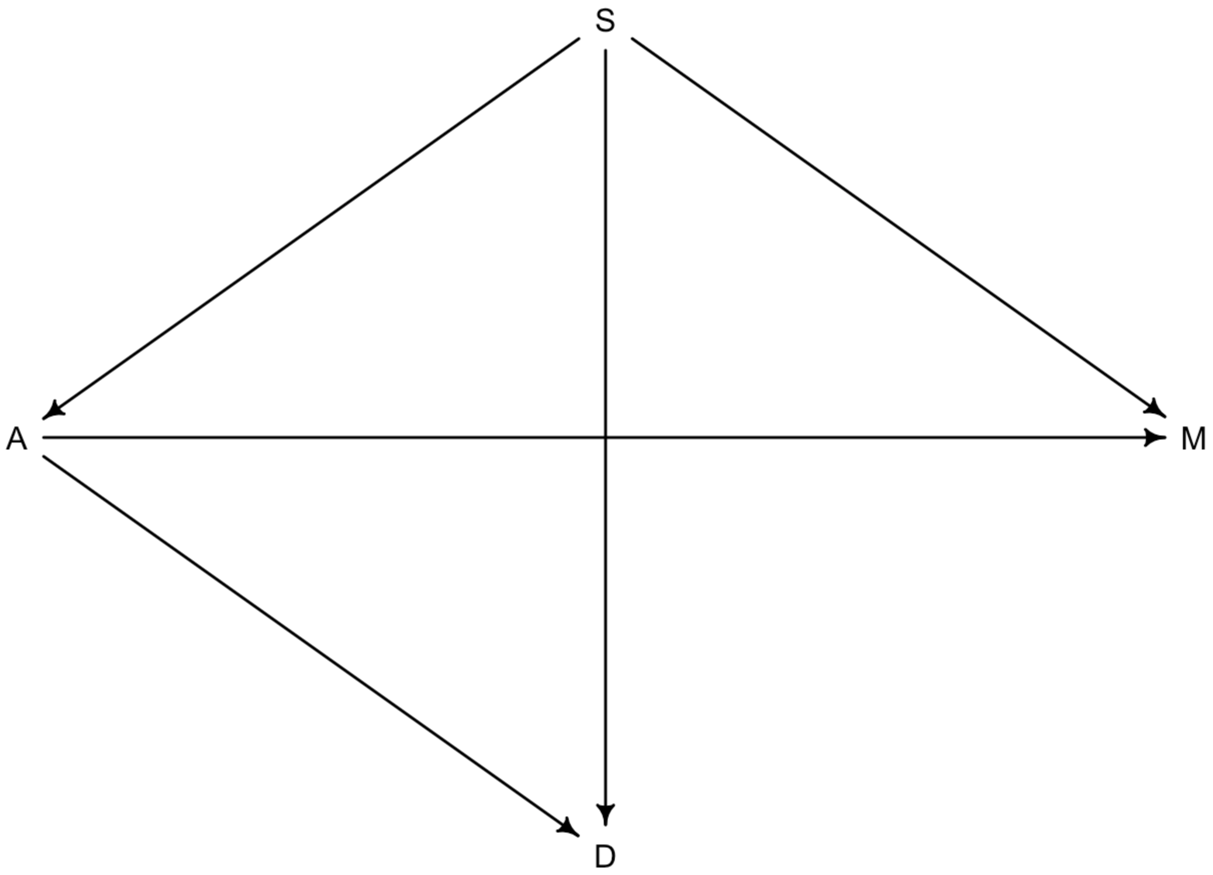
3) Causal Models (TI & MST)

A problem from our textbook *Statistical Rethinking (2nd ed.)* (p. 160):

5H4. Here is an open practice problem to engage your imagination. In the divorce data, states in the southern United States have many of the highest divorce rates. Add the `South` indicator variable to the analysis. First, draw one or more DAGs that represent your ideas for how Southern American culture might influence any of the other three variables (D, M, or A). Then list the testable implications of your DAGs, if there are any, and fit one or more models to evaluate the implications. What do you think the influence of “Southernness” is?

DAG models

```
DAG <- dagitty('dag{S -> M; S -> D; S -> A; A -> D; A -> M}')

coordinates(DAG) <- list(x=c(A=-1, D=0, M=1, S=0), y=c(A=1, D=2, M=1, S=0))

drawdag(DAG)
```



In our proposed DAG model we hypothesize that Southerness influences the median age at marriage, the marriage rate and divorce rate.

Testing implications of DAG model

```
impliedConditionalIndependencies(DAG)
```

```
## D _||_ M | A, S
```

Divorce rate is independent of marriage rate, conditional on median age at marriage and whether or not you are from the South.

Fit model to evaluate implications

```
data("WaffleDivorce")
d <- WaffleDivorce
d_n <- d %>%
  as_tibble() %>%
  select(D = Divorce, A = MedianAgeMarriage, M = Marriage, S = South)
d_n$D <- standardize(d_n$D)
d_n$A <- standardize(d_n$A)
d_n$M <- standardize(d_n$M)
d_n$S <- as.factor(d_n$S)



m1 <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a[S] + bM[S]*M + bA[S]*A,
    a[S] ~ dnorm(0,0.2),
    bM[S] ~ dnorm(0,0.5),
    bA[S] ~ dnorm(0,0.5),
    sigma ~ dexp(1)),
    data = d_n)
precis(m1, depth = 2)
```

	mean <dbl>	sd <dbl>	5.5% <dbl>	94.5% <dbl>
a[1]	-0.1005832	0.10419478	-0.26710655	0.06594023
a[2]	0.1056980	0.15096425	-0.13557206	0.34696799
bM[1]	-0.1013165	0.14738940	-0.33687322	0.13424023
bM[2]	0.4489774	0.31731219	-0.05814872	0.95610362
bA[1]	-0.5293613	0.14934044	-0.76803614	-0.29068642
bA[2]	-0.7246914	0.32908858	-1.25063855	-0.19874434
sigma	0.7229209	0.07263505	0.60683610	0.83900578

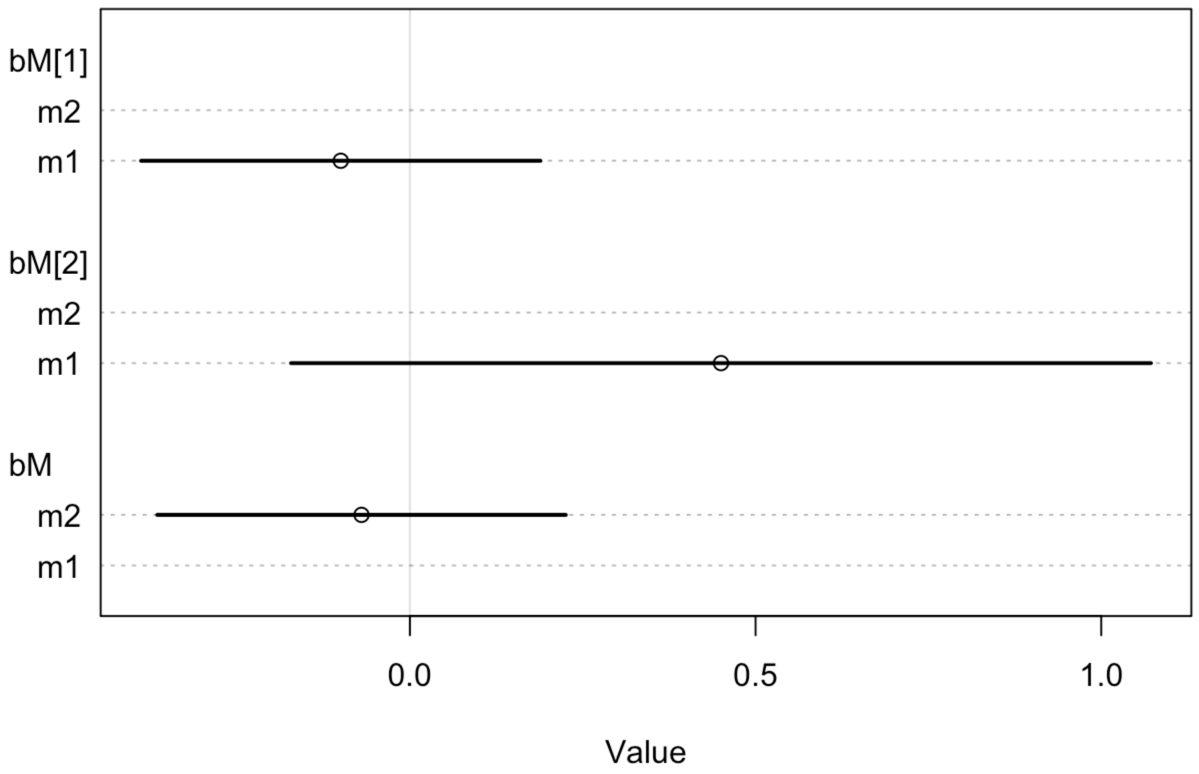
7 rows

```
m2 <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a + bM*M + bA*A,
    a ~ dnorm(0,0.2),
    bM ~ dnorm(0,0.5),
    bA ~ dnorm(0,0.5),
    sigma ~ dexp(1)),
    data = d_n)
precis(m2, depth = 2)
```

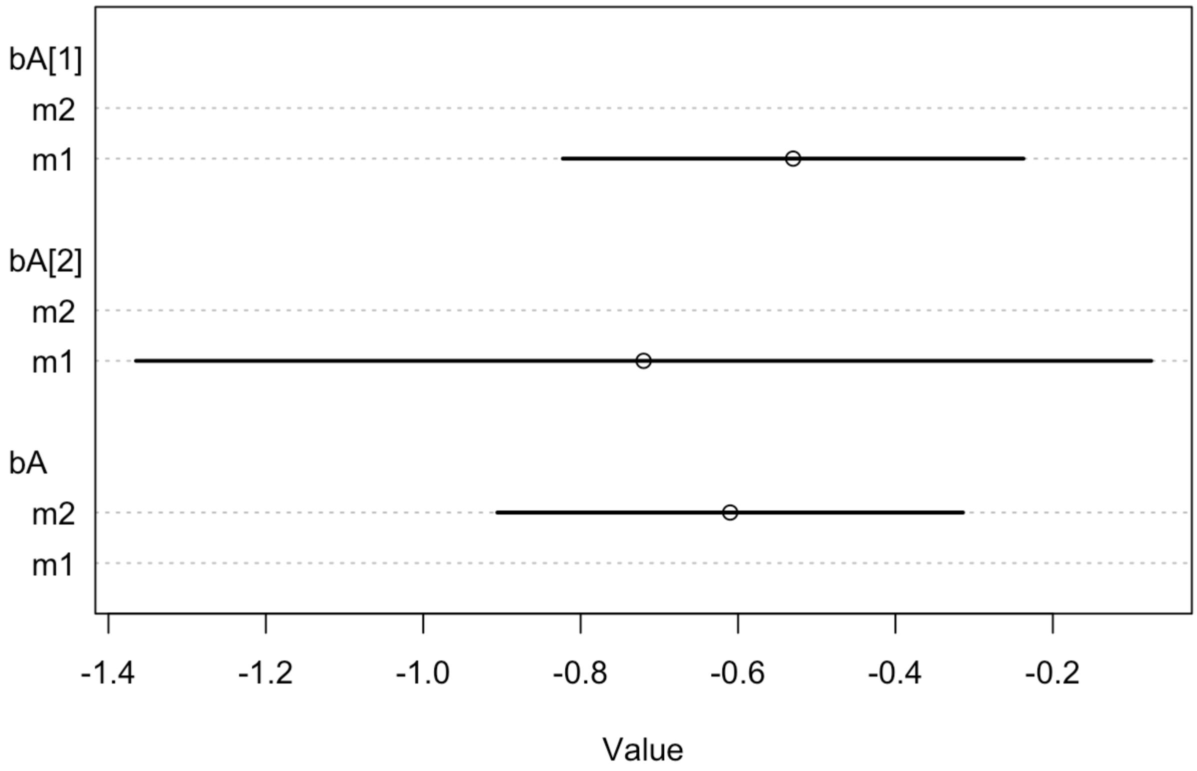
	mean	**sd**	**5.5%**	**94.5%**
	<dbl>	<dbl>	<dbl>	<dbl>
a	7.345618e-06	0.09707473	-0.1551368	0.1551515
bM	-6.544232e-02	0.15077015	-0.3064021	0.1755175
bA	-6.135671e-01	0.15098059	-0.8548633	-0.3722710
sigma	7.851042e-01	0.07783995	0.6607009	0.9095075

4 rows

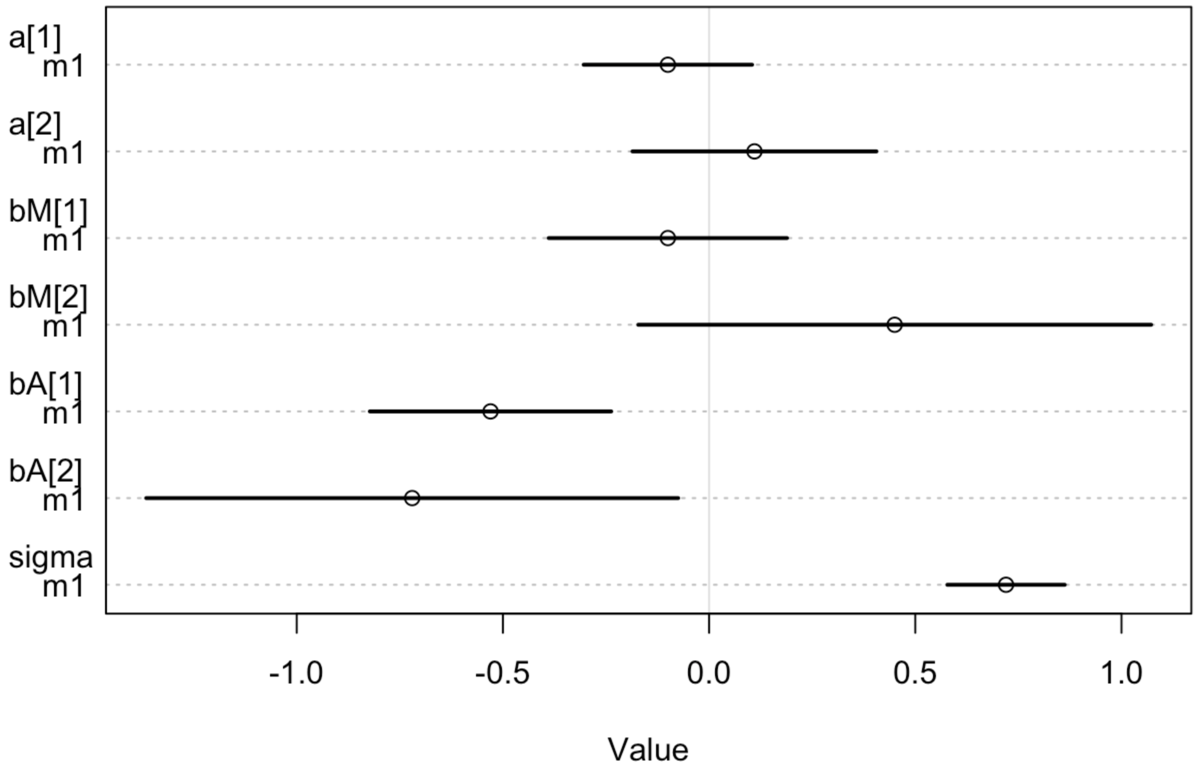
```
plot(coeftab(m1, m2), par = c("bM[1]", "bM[2]", "bM"))
```



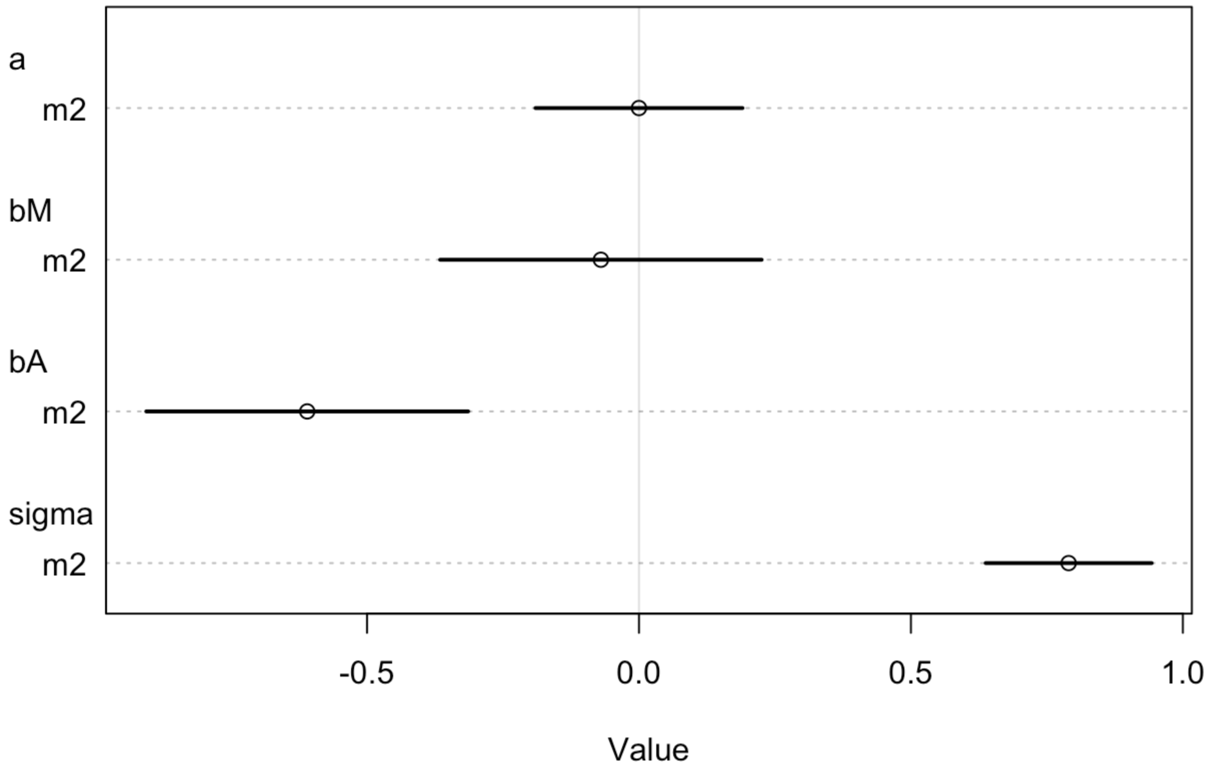
```
plot(coeftab(m1, m2), par = c("bA[1]", "bA[2]", "bA"))
```



```
plot(coeftab(m1))
```



```
plot(coeftab(m2))
```



Conclusion:

The plots showing the different parameters from the two models indicate an effect of southerness on both the influence of marriage rate on divorce rate and the influence of median age of marriage on divorce rate. These effects are however small, and as the data includes less southern states than non-southern states there is much more variance in the parameter estimates, which results in an overlap of all the parameter estimates. However looking at what the models indicate, it seems that southerness affects the correlation between marriage rate and divorce rate (i.e. increased marriage rate results in increased divorce rate, with a steeper slope for southern states) and affects the correlation between median age of marriage and divorce rate (i.e. lower median age of marriage results in increased divorce rates, with a steeper slope for southern states).

1. I was lazy and just used this source:

   https://www.ssi.dk/aktuelt/nyheder/2021/antigentest-gav-47-falsk-negative-svar (https://www.ssi.dk/aktuelt/nyheder/2021/antigentest-gav-47-falsk-negative-svar)↩

2. https://www.worldometers.info/coronavirus/#countries (https://www.worldometers.info/coronavirus/#countries)↩

3. That one I just made up.↩