Methods 4 – Portfolio Assignment 3

- *Type:* Individual assignment
- *Due:* 1 May 2022, 23:59

Hey again CogSci's :)

So now for the last of the three portfolios :)

This time it's an individual one. We will build a workflow and use it to analyze a new dataset.

There are seven tasks below. As usual, handing in as a markdown is nice :)

1. Get familiar with the data

This dataset contains information about passengers aboard the Titanic.

```
df_train <- read_csv("data/titanic_train.csv")
```

```
## Rows: 891 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_test <- read_csv("data/titanic_test.csv")
```

```
## Rows: 418 Columns: 11

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

The dataset includes 12 variables (as described in the Kaggle entry):

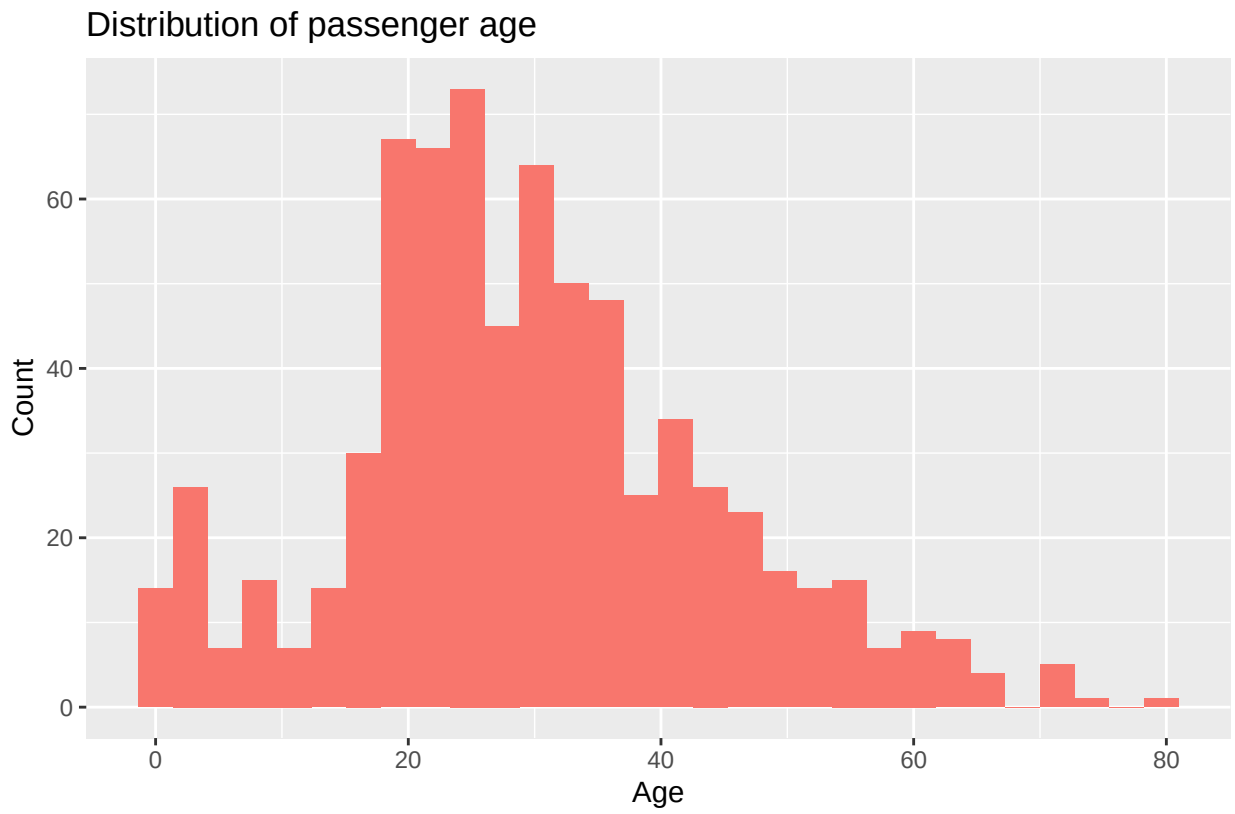
Survival: Survival (0 = No; 1 = Yes) Pclass: Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd) Name: Name of passenger Sex: Sex of passenger Age: Age of passenger Sibsp: Number of Siblings/Spouses Aboard for passenger Parch: Number of Parents/Children Aboard for passenger Ticket: Ticket number Fare: Passenger fare (price paid by passenger, can include multiple tickets) Cabin: Cabin no. (a lot of NA's in this variable) Embarked: Port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Plotting some variables of interest

```
df_train %>%
  ggplot()+
  geom_histogram(aes(Age, fill = "red"))+
  xlab("Age")+
  ylab("Count")+
  labs(title = "Distribution of passenger age", caption = "We see that age is somewhat normally distribu
  theme(legend.position = "none")
```

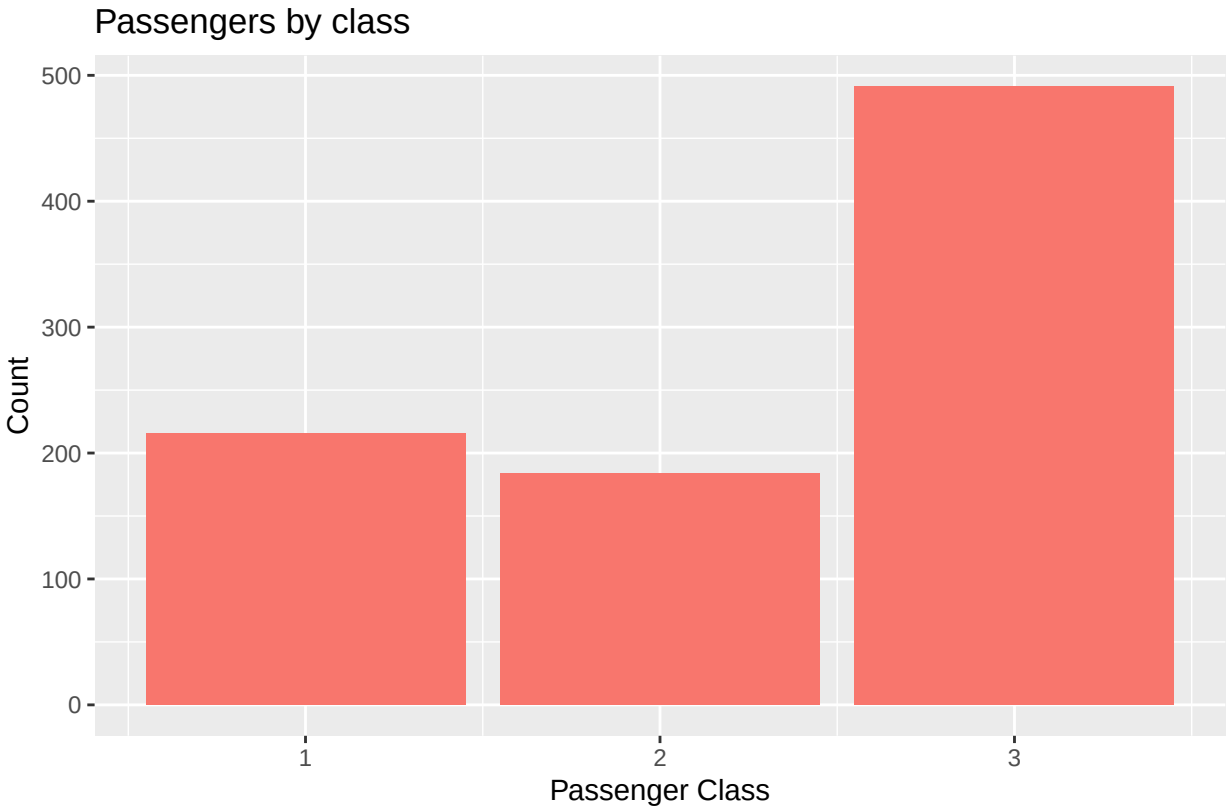
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 177 rows containing non-finite values (stat_bin).
```



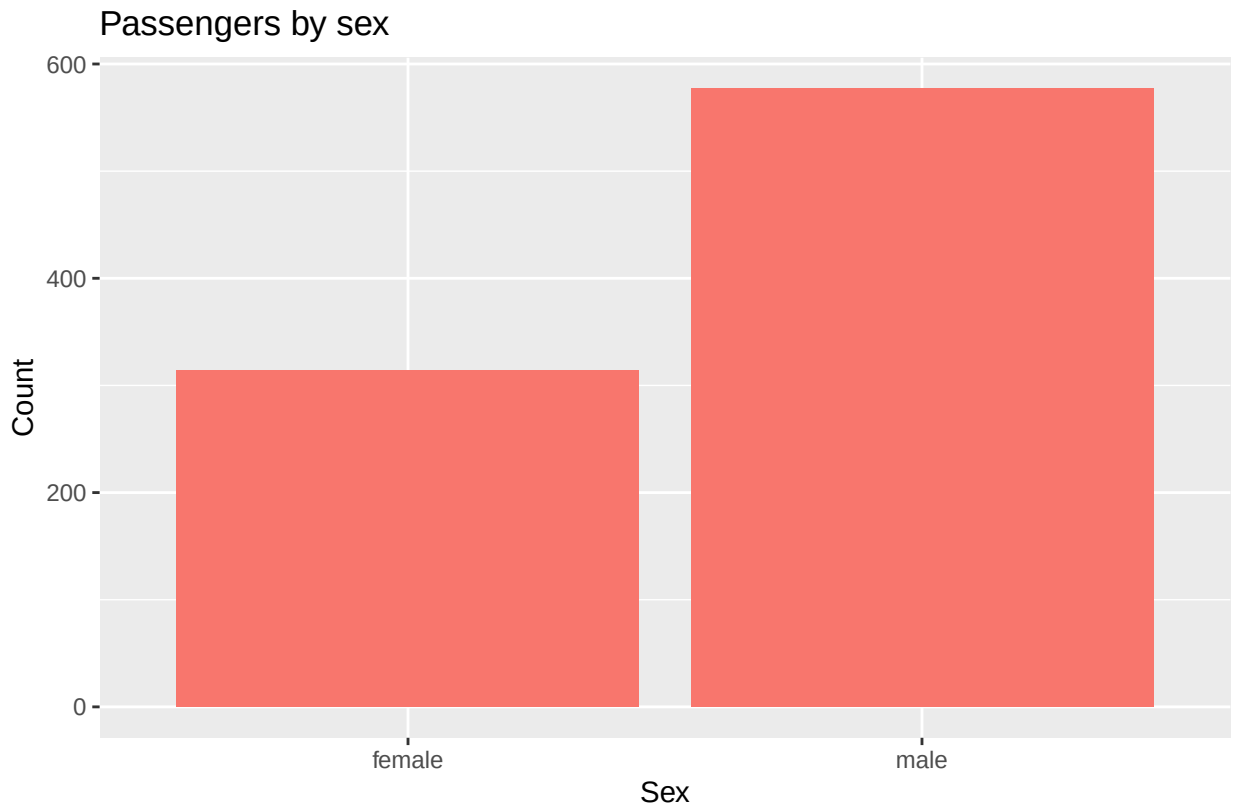
We see that age is somewhat normally distributed, skewed a bit towards younger passengers.

```
df_train %>%
  ggplot()+
  geom_bar(aes(Pclass, fill = "red"))+
  xlab("Passenger Class")+
  ylab("Count")+
  labs(title = "Passengers by class", caption = "We see that there are more on 3rd class than 1st and 2
  theme(legend.position = "none")
```



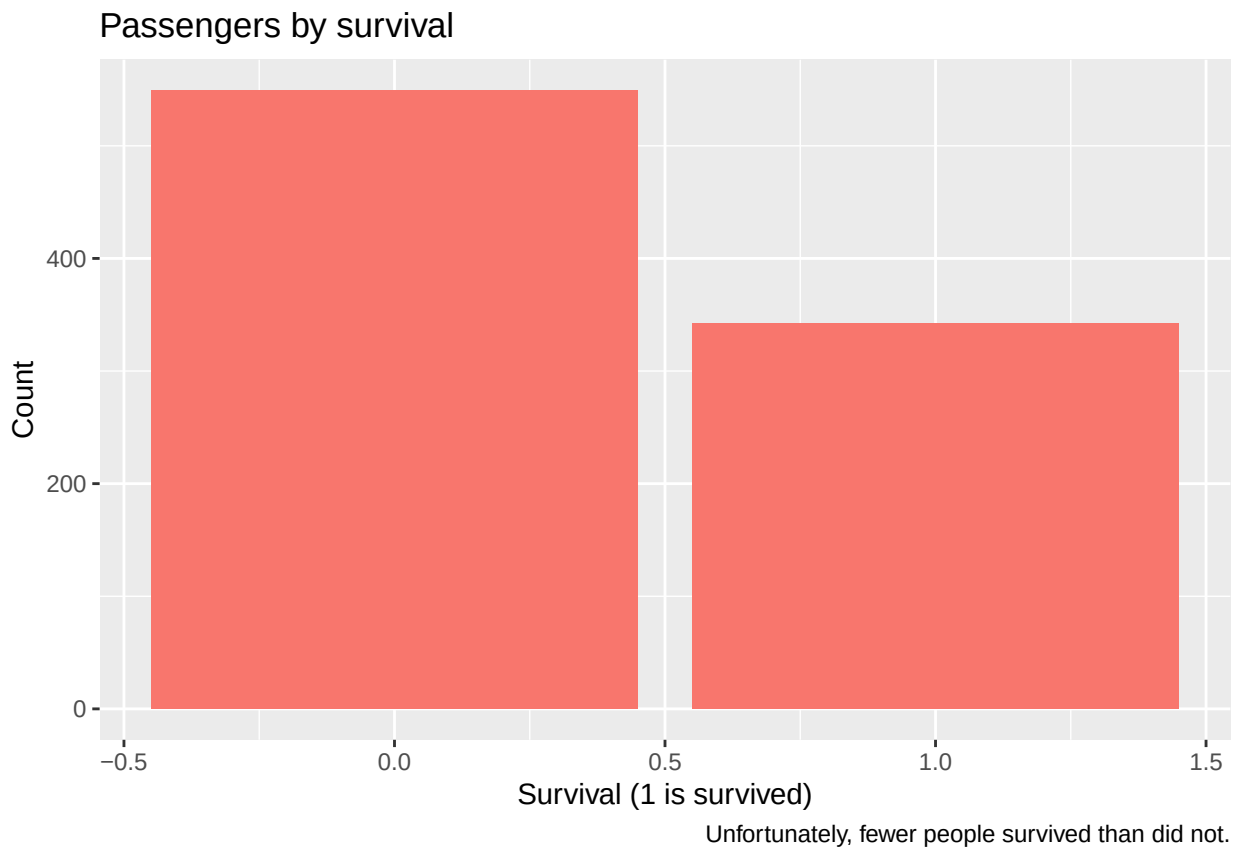
We see that there are more on 3rd class than 1st and 2nd combined.

```
df_train %>%
  ggplot()+
  geom_bar(aes(Sex, fill = "red"))+
  xlab("Sex")+
  ylab("Count")+
  labs(title = "Passengers by sex", caption = "Generally more men were aboard the Titanic than women, al
  theme(legend.position = "none")
```



Generally more men were aboard the Titanic than women, almost double the amount.

```
df_train %>%
  ggplot()+
  geom_bar(aes(Survived, fill = "red"))+
  xlab("Survival (1 is survived)")+
  ylab("Count")+
  labs(title = "Passengers by survival", caption = "Unfortunately, fewer people survived than did not.")
  theme(legend.position = "none")
```



2. Choose an estimand / outcome

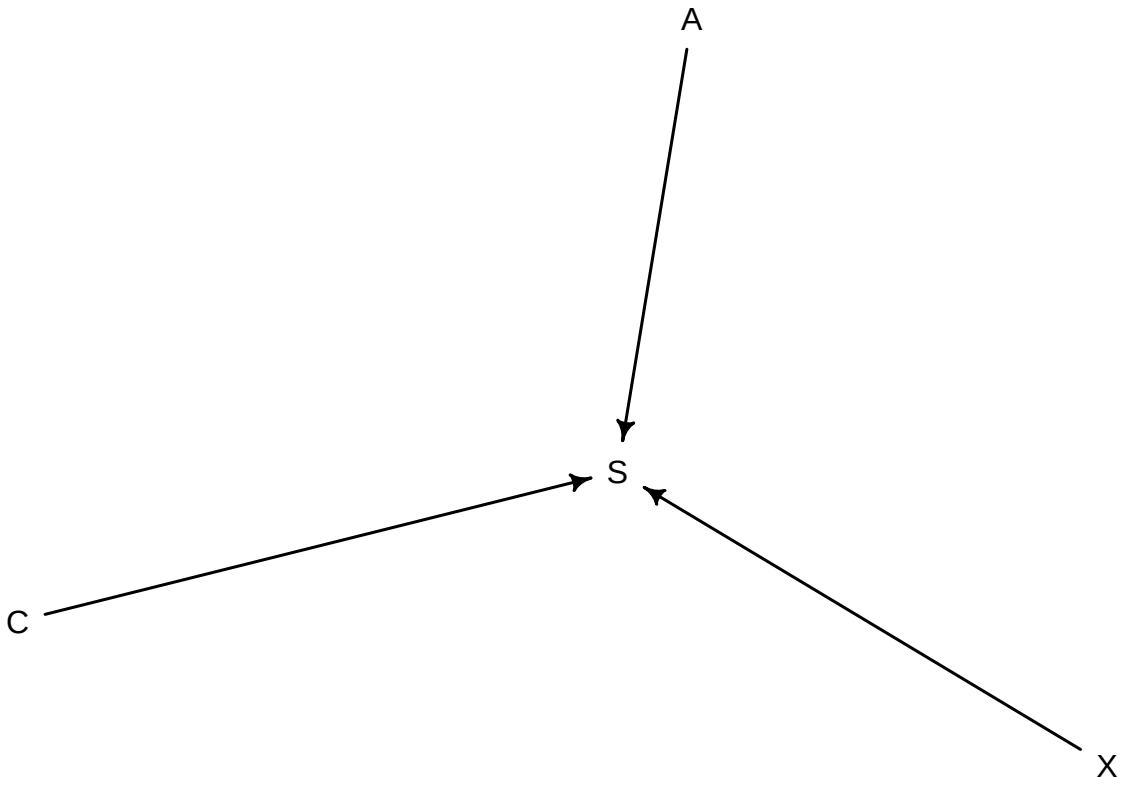
- **I will do as recommended and choose survival as my outcome variable. :)**

3. Make a scientific model (i.e., a DAG)

Make a DAG that seems theoretically reasonable, and that includes some of the variables in the dataset. It can include unobserved variables too, if you want, but then you have to come up with them.

You might have to return to this point later on ;)

```
dag <- dagitty( "dag {
    A -> S
    C -> S
    X -> S
}")

drawdag(dag)
```

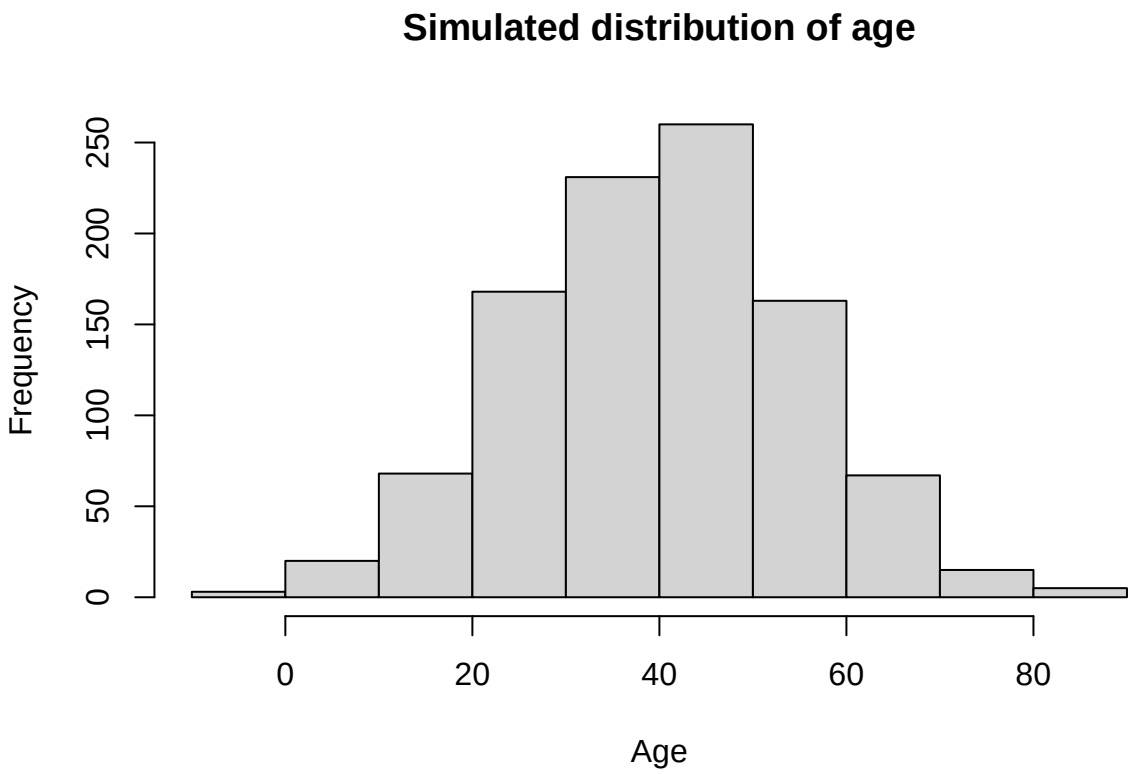


Considering the variables available, i would expect survival to be predicted by sex, age and class. This is a very simple DAG, but it is based on my ignorance of the famous ship

4. Simulate data from the DAG

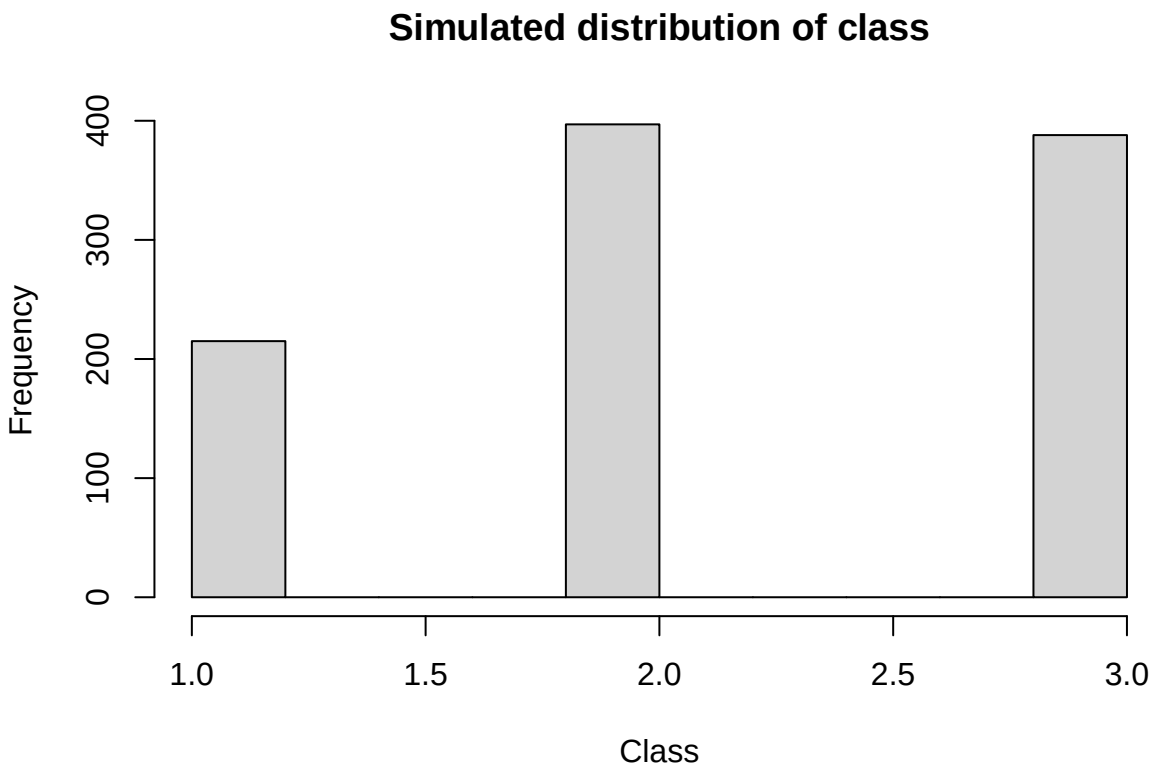
Age: I would expect a mean age of 40 and a std of 15. When sampling, some values might dip below 0, which of course isn't interpretable when dealing with age, but when simulating it should be okay. There were roughly 2200 people aboard the Titanic, but for simplicity I will sample 1000.

```
A = rnorm(1000, 40, 15)
hist(A, main = paste("Simulated distribution of age"), xlab = "Age")
```



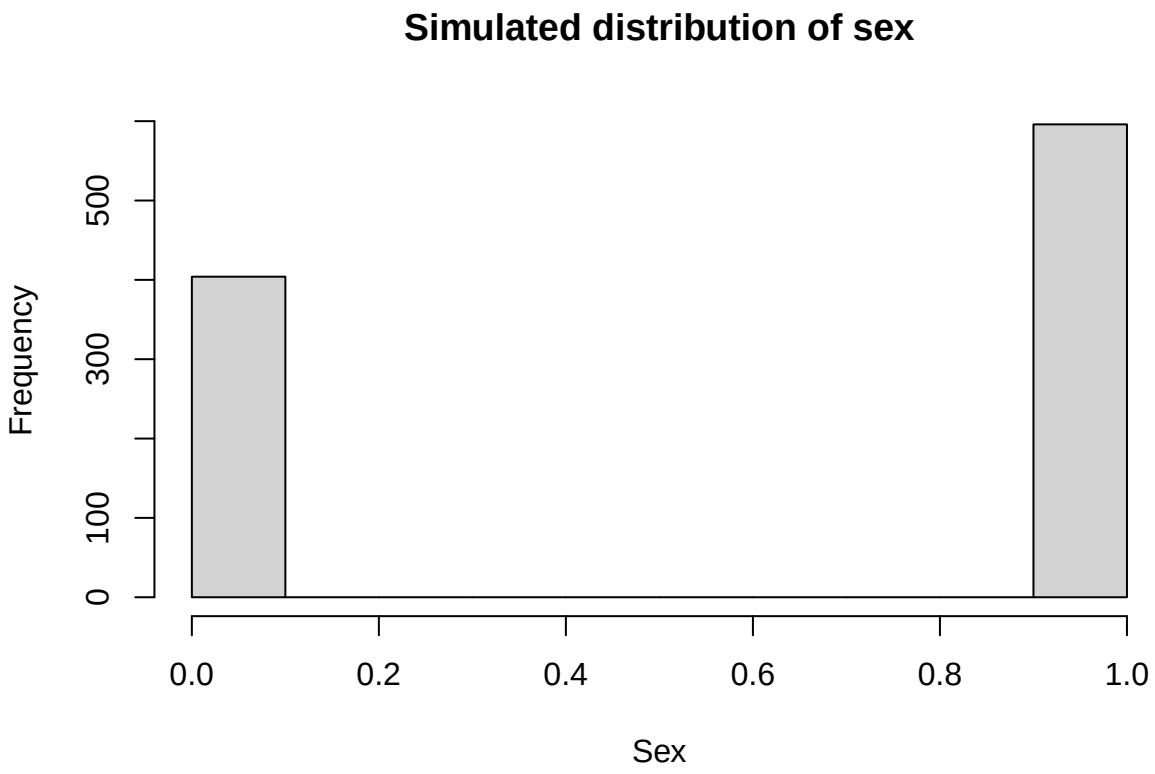
Class: I would expect less 1st class than 2nd and 3rd, as 1st class probably requires more space and requires more service.

```
C = sample(1:3, 1000, prob = c(0.2, 0.4, 0.4), replace=TRUE)
hist(C, main = paste("Simulated distribution of class"), xlab = "Class")
```



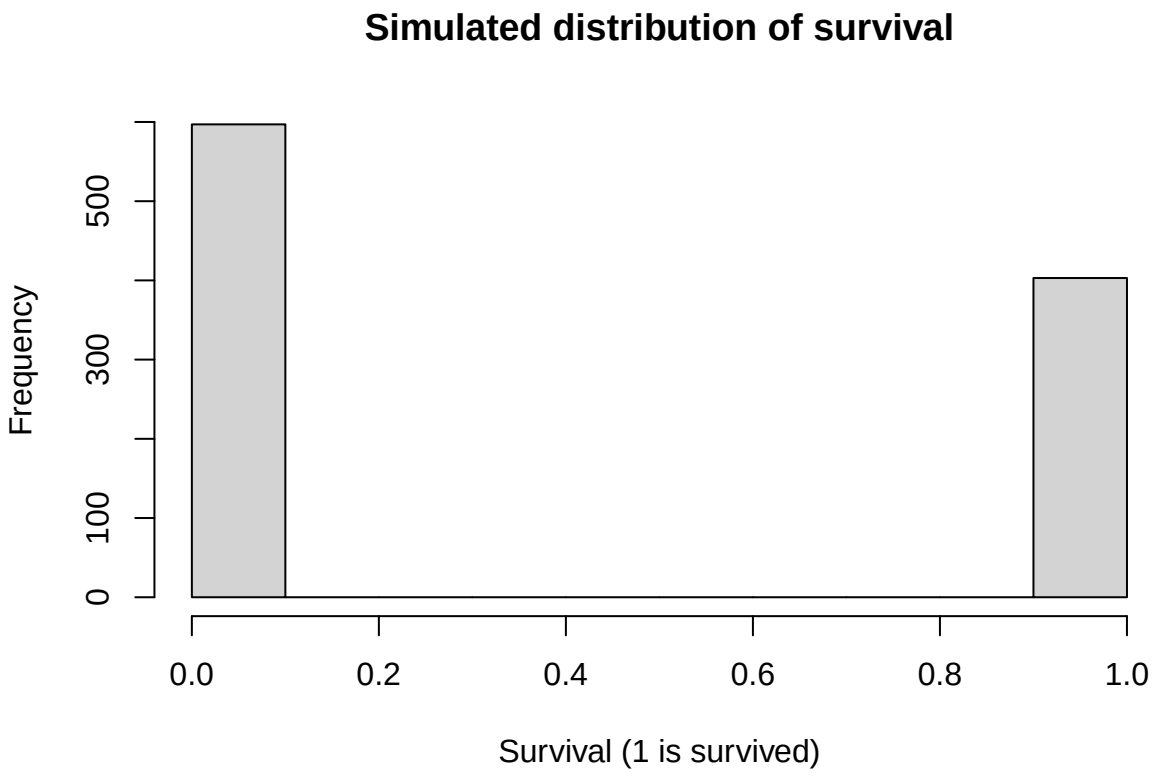
Sex: Based on the movie and stereotypes, I'd expect more men aboard than women due to the expensive tickets, and assuming that crew members are largely men. 1 = mean, 0 = women

```
X = rbinom(1000, 1, 0.6)
hist(X, main = paste("Simulated distribution of sex"), xlab = "Sex")
```



Survived: The "base" survival rate is sort of arbitrary, but i'd imagine less people to survive than not. Then again, i don't know how likely people are to survive a sinking ship. The weightings are based on the following: Age is scaled and weighted so that younger people are more likely to survive, higher class are more likely to survive and women are more likely to survive.

```
S = rbinom(1000, 1, prob = 0.4 + (-0.05)*scale(A) + (-0.05)*scale(C) + (-0.1)*scale(X))
hist(S, main = paste("Simulated distribution of survival"), xlab = "Survival (1 is survived)")
```




```
table(is.na(S))
```

```
##
## FALSE
##  1000
```

```
df_sim <- data.frame(A, C, X, S)
```

Based on the plots above, i'd say my simulated data isn't that far off from the real data.

5. Make a statistical model

The relevant predictors for predicting survival, as implied by the DAG, are age, sex and class, therefore S ~ A + X + C. In the model, survived is a binary variable (although treated as an integer), so the posterior is a binomial distribution. There are random slopes for sex and for class, and age is a fixed effect. Priors are chosen based on prior predictive checks. Interactions will be explored when modeling the real data later on.

6. Test the statistical model on the simulated data

```
# Standardizing
dat_sim = list(
```

```
  A = scale(df_sim$A),
  C = as.factor(df_sim$C),
  X = as.factor(df_sim$X),
  S = as.integer(df_sim$S)
)

# Using a modest prior
m1_1 <- ulam(
  alist(

    ## S ~ A + X + C

    S ~ dbinom(1, p),
    logit(p) <- aX[X] + aC[C] + bA*A,
    aX[X] ~ dnorm(0, 0.5),
    aC[C] ~ dnorm(0, 0.5),
    bA ~ dnorm(0, 0.5)

  ), data = dat_sim, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 5.7 seconds.
```

```
# Very narrow priors
m1_2 <- ulam(
  alist(

    ## S ~ A + X + C

    S ~ dbinom(1, p),
    logit(p) <- aX[X] + aC[C] + bA*A,
    aX[X] ~ dnorm(0, 0.0001),
    aC[C] ~ dnorm(0, 0.0001),
    bA ~ dnorm(0, 0.0001)

  ), data = dat_sim, log_lik = TRUE, refresh = 0
)
```

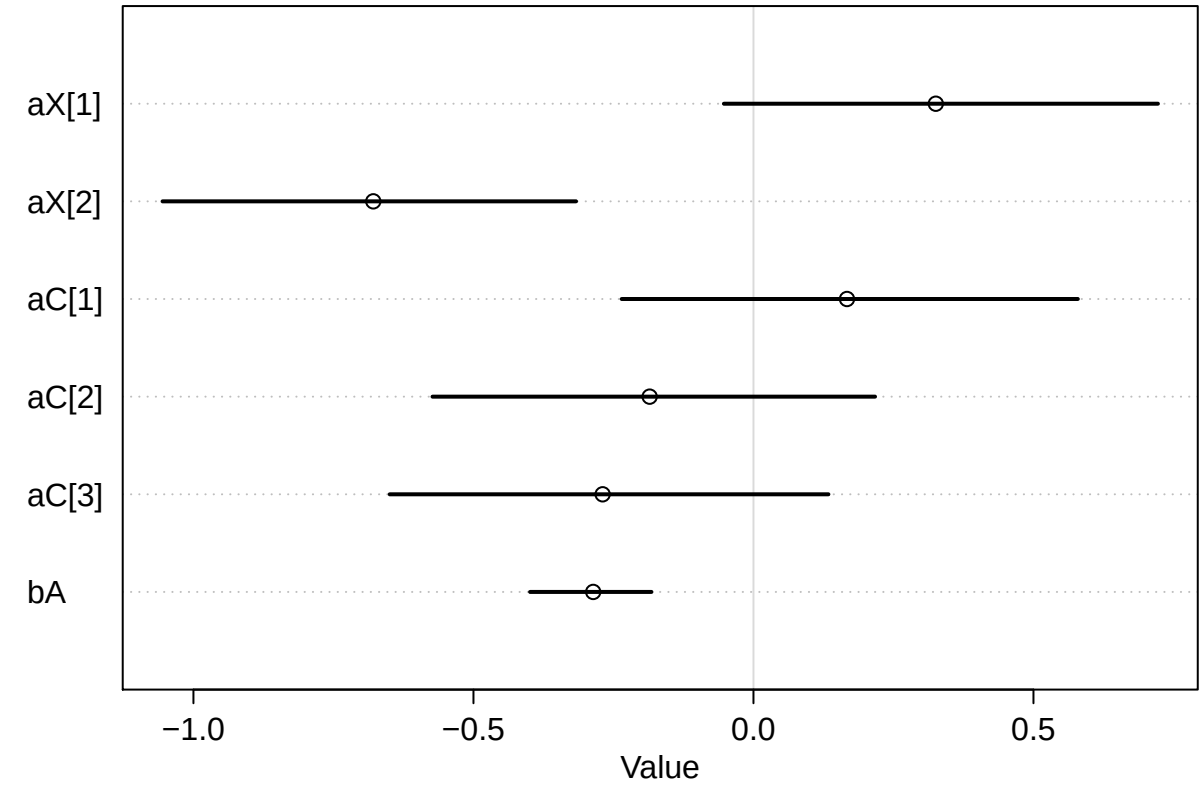
```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 3.7 seconds.
```

```
# Very large priors
m1_3 <- ulam(
  alist(

    ## S ~ A + X + C

    S ~ dbinom(1, p),
    logit(p) <- aX[X] + aC[C] + bA*A,
```

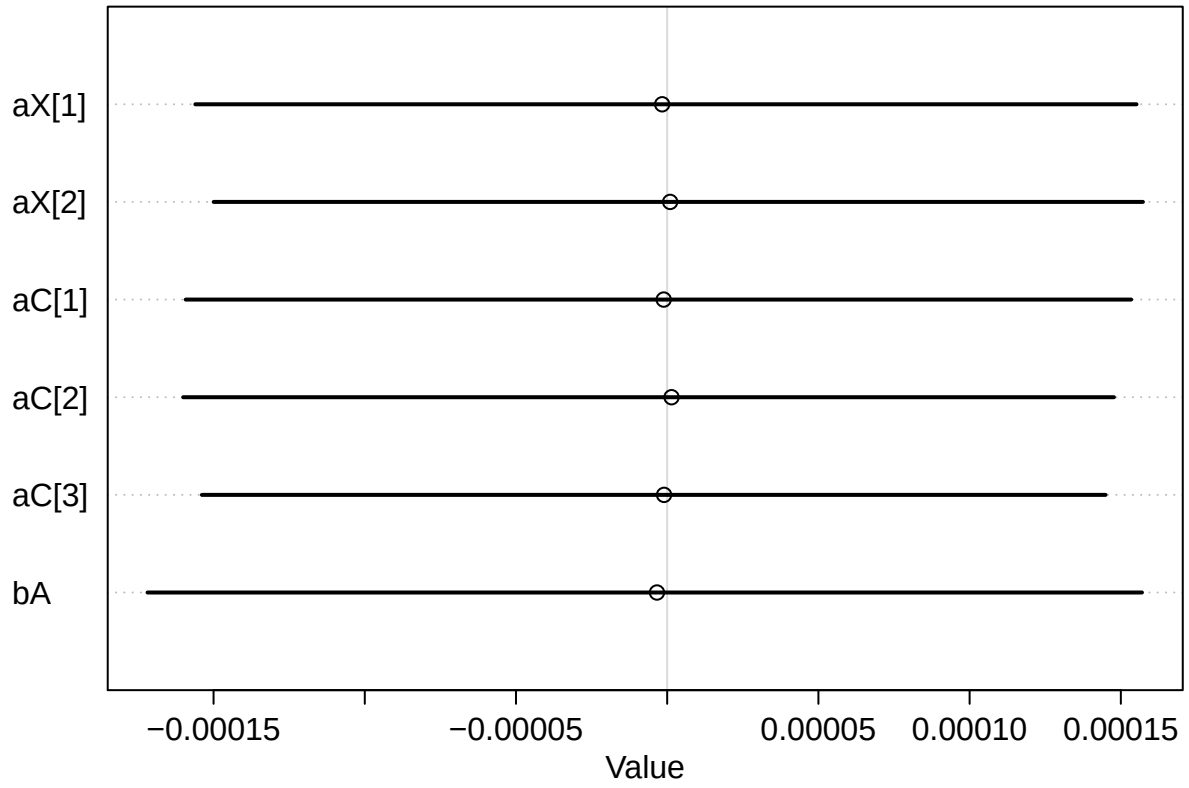
```
    aX[X] ~ dnorm(0, 10),
    aC[C] ~ dnorm(0, 10),
    bA ~ dnorm(0, 10)

  ), data = dat_sim, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 25.1 seconds.
```

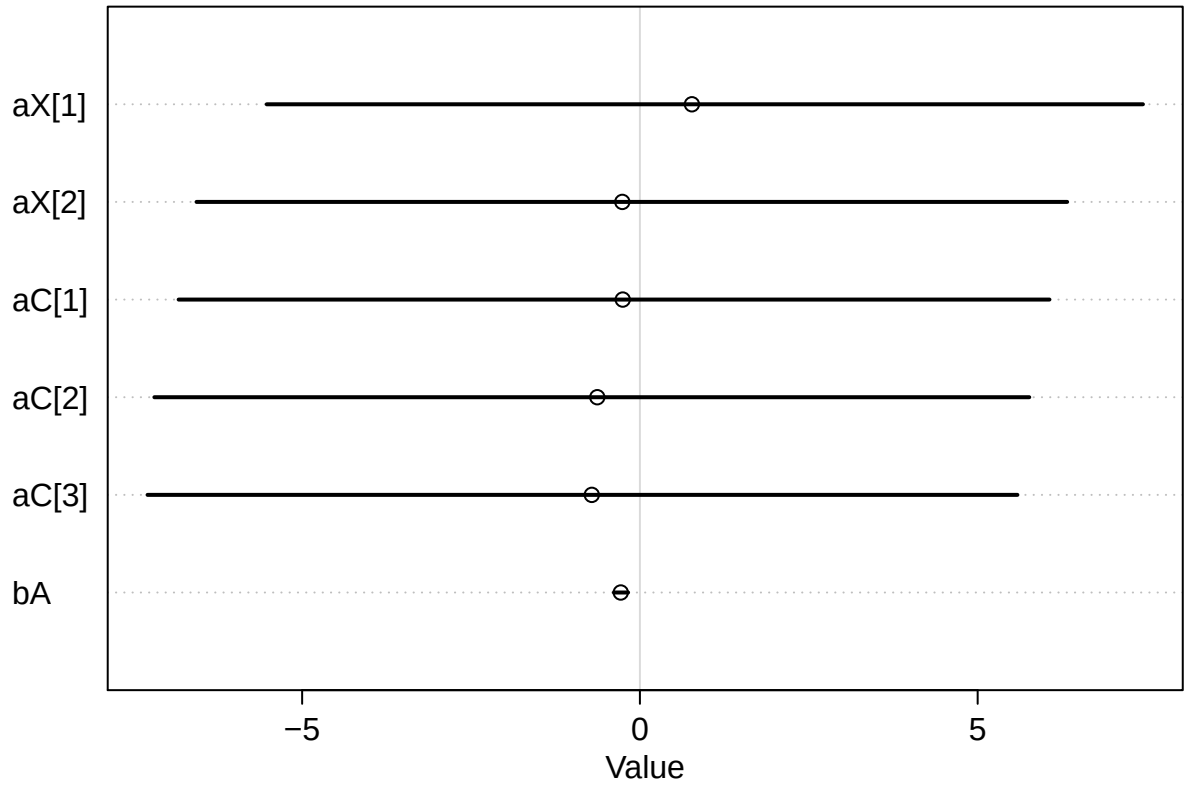
```
precis_plot(precis(m1_1, depth = 2))
```




```
precis_plot(precis(m1_2, depth = 2))
```




```
precis_plot(precis(m1_3, depth = 2))
```



This is the statistical model including all variables, congruent with the DAG. This shows indeed that sex has an effect on survival with men dying more than women. Class also affects whether passengers survive or not, with 1st class being most safe, then 2nd and then 3rd. The slope parameter for age shows that older passengers are less likely to survive. The model reflects the intuitions which the data was simulated on. The priors I've chosen for all three parameters are (0, 0.5), since these look most reasonable in regards to my expectations of the values. The too narrow priors make the effects crazy small, and increases uncertainty wildly. Too broad priors have the same effect but on a larger scale. I will stick with priors of mean 0 and std 0.5.

A note on priors

```
# Extracting priors
prior1 <- extract.prior(m1_1)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1 Iteration:  100 / 2000 [  5%]  (Warmup)
## Chain 1 Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1 Iteration:  300 / 2000 [ 15%]  (Warmup)
## Chain 1 Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
## Chain 1 Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 1 Iteration:  700 / 2000 [ 35%]  (Warmup)
## Chain 1 Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1 Iteration:  900 / 2000 [ 45%]  (Warmup)
## Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1 Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1 finished in 2.8 seconds.
```

```
prior2 <- extract.prior(m1_2)
```

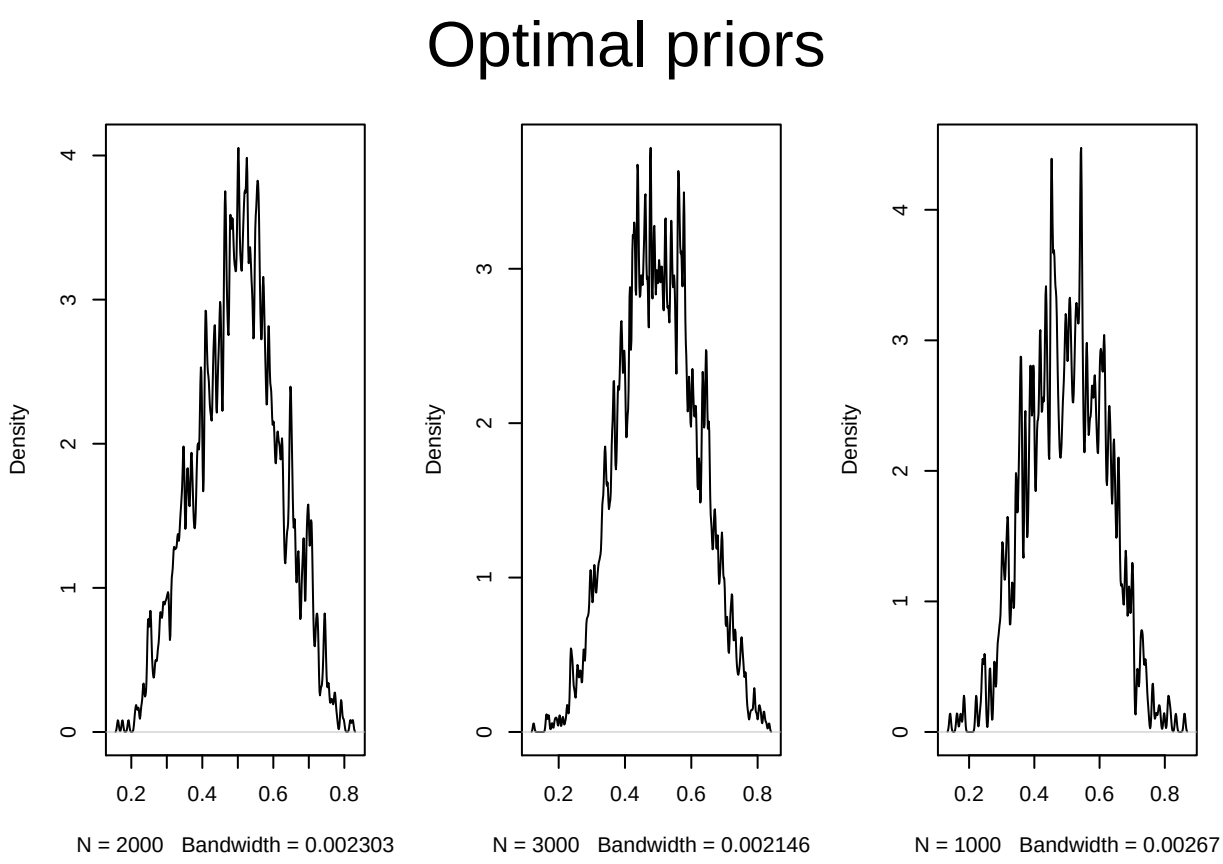
```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1 Iteration:  100 / 2000 [  5%]  (Warmup)
## Chain 1 Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1 Iteration:  300 / 2000 [ 15%]  (Warmup)
## Chain 1 Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
## Chain 1 Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1 Iteration:  700 / 2000 [ 35%]  (Warmup)
## Chain 1 Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1 Iteration:  900 / 2000 [ 45%]  (Warmup)
## Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1 Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1 finished in 3.8 seconds.
```

```
prior3 <- extract.prior(m1_3)
```

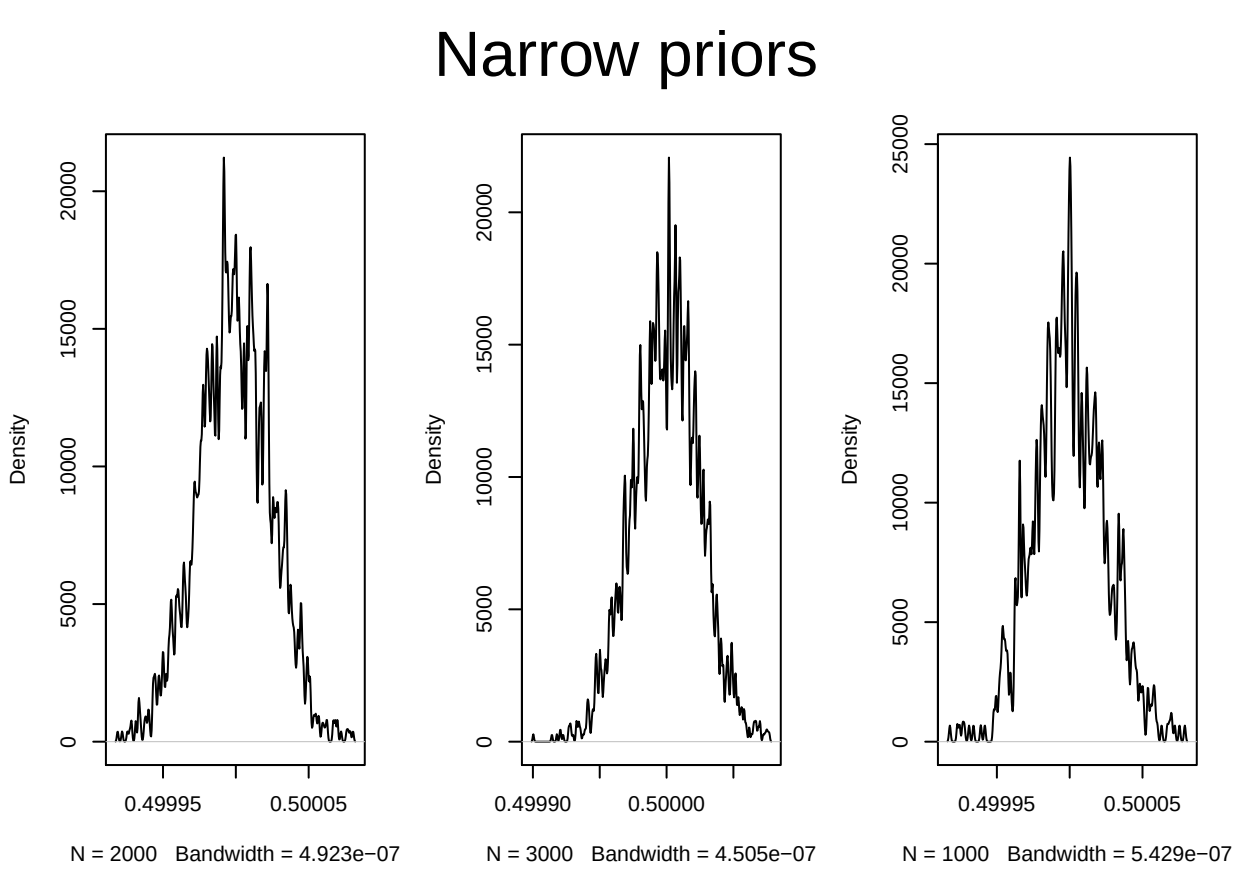
```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1 Iteration:  100 / 2000 [  5%]  (Warmup)
## Chain 1 Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1 Iteration:  300 / 2000 [ 15%]  (Warmup)
```

```
## Chain 1 Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
## Chain 1 Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1 Iteration:  700 / 2000 [ 35%]  (Warmup)
## Chain 1 Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1 Iteration:  900 / 2000 [ 45%]  (Warmup)
## Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1 Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1 finished in 2.8 seconds.
```

```
par(mfrow=c(1,3))
dens(inv_logit(prior1$aX), adj=0.1)
dens(inv_logit(prior1$aC), adj=0.1)
dens(inv_logit(prior1$bA), adj=0.1)
mtext("Optimal priors", side=3, line=-2, cex=2, outer = TRUE)
```

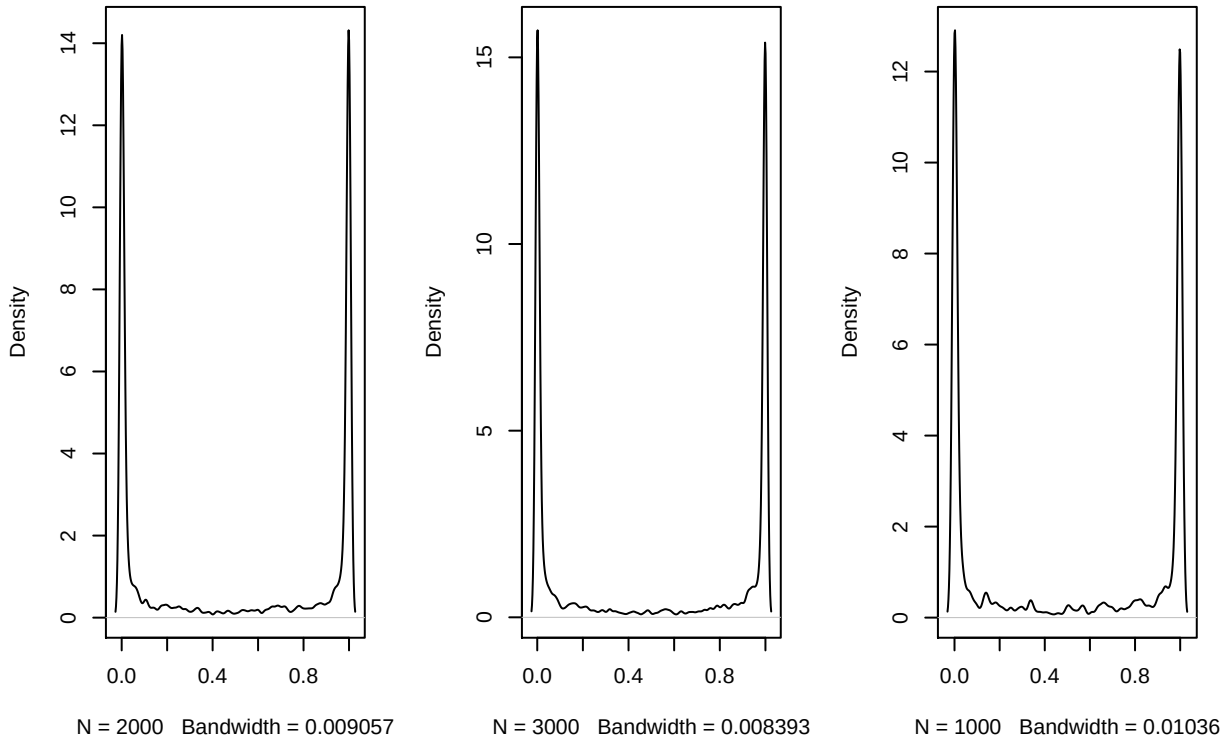


```
par(mfrow=c(1,3))
dens(inv_logit(prior2$aX), adj=0.1)
dens(inv_logit(prior2$aC), adj=0.1)
dens(inv_logit(prior2$bA), adj=0.1)
mtext("Narrow priors", side=3, line=-2, cex=2, outer = TRUE)
```




```
par(mfrow=c(1,3))
dens(inv_logit(prior3$aX), adj=0.1)
dens(inv_logit(prior3$aC), adj=0.1)
dens(inv_logit(prior3$bA), adj=0.1)
mtext("Wide priors", side=3, line=-2, cex=2, outer = TRUE)
```

Wide priors



These density plots confirm that my choice of priors (optimal) are at least reasonable. When using priors with a std of 10, the model is very sure of either surival or not, which i know isn't the case in the data. When using very narrow priors, it is highly unlikely that the real values are in this distribution space.

Testing conditional independencies

```
impliedConditionalIndependencies(dag)
```

```
## A _||_ C
## A _||_ X
## C _||_ X
```

```
# A _||_ C
m3 <- ulam(
  alist(

    ## A ~ C

    A ~ dnorm(mu, sigma),
    mu <- aC[C],
    aC[C] ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
```

```
  ), data = dat_sim, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 2.2 seconds.
```

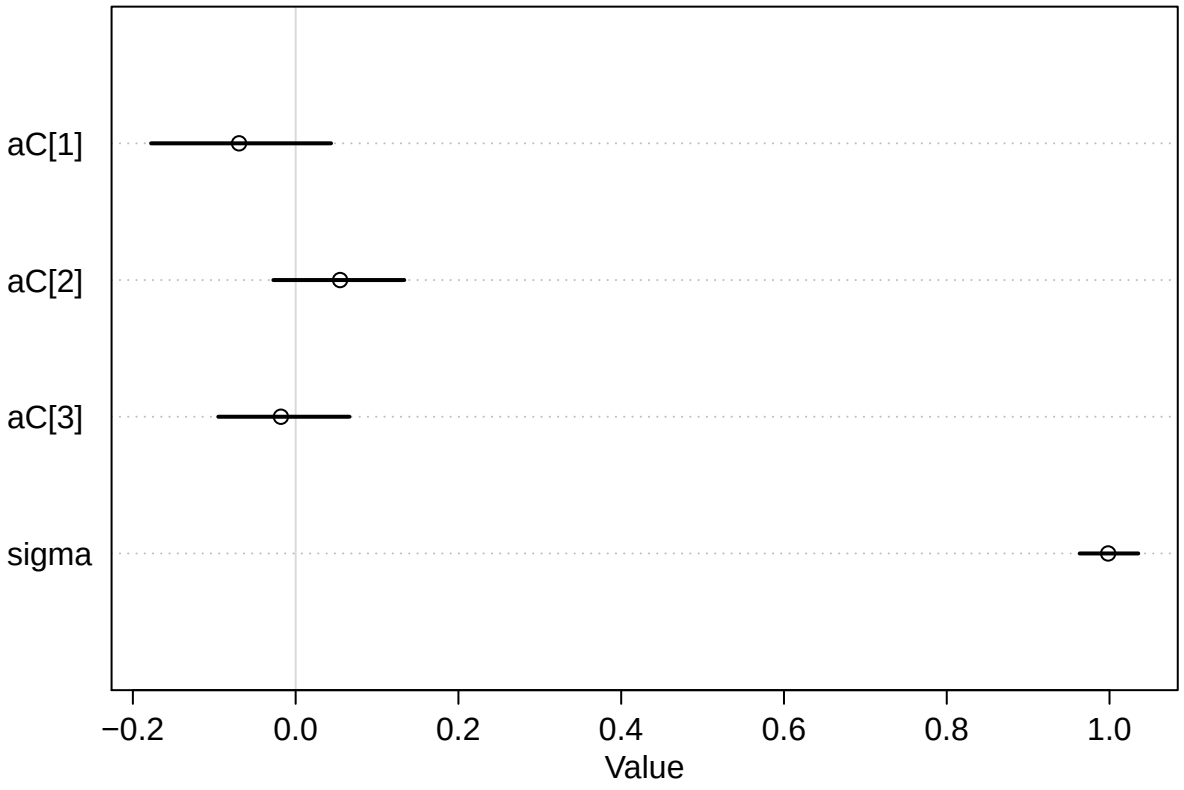
```
# A _||_ X
# I need X as integer for it to be the outcome, otherwise i get an error
dat_sim_X <- dat_sim
dat_sim_X$X <- as.integer(dat_sim_X$X)-1

m4 <- ulam(
  alist(

    ## X ~ A

    X ~ dbinom(1, p),
    logit(p) <- a + bA*A,
    a ~ dnorm(0, 0.5),
    bA ~ dnorm(0, 0.5)

  ), data = dat_sim_X, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 3.5 seconds.
```

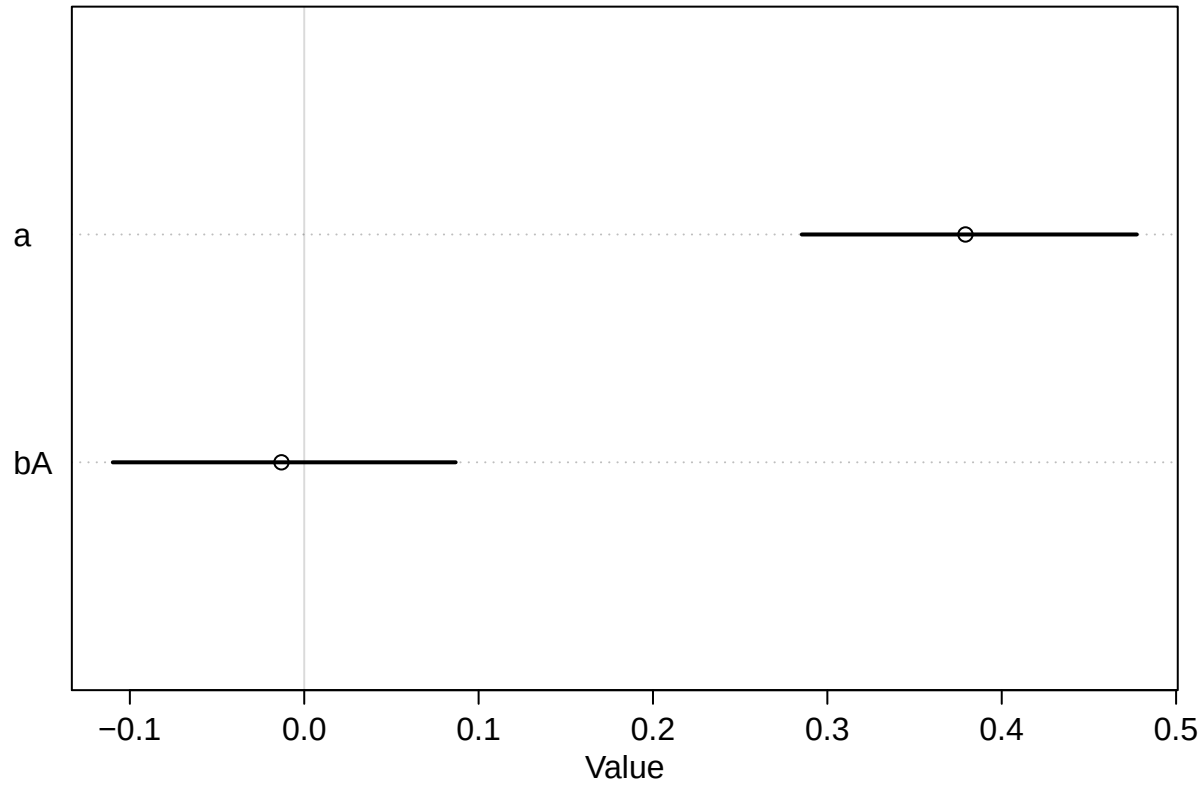
```
# C _||_ X
m5 <- ulam(
  alist(

    ## X ~ C

    X ~ dbinom(1, p),
    logit(p) <- aC[C],
    aC[C] ~ dnorm(0, 0.5)

  ), data = dat_sim_X, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 3.6 seconds.
```

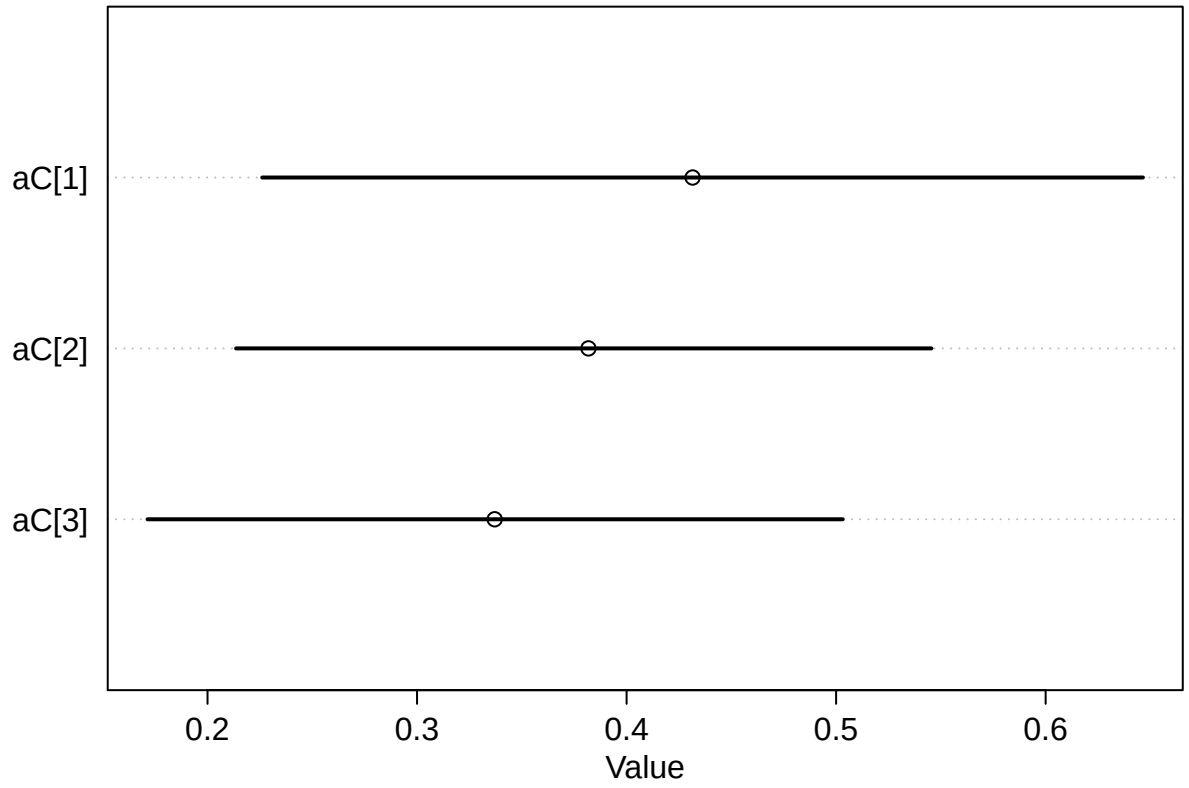
```
precis_plot(precis(m3, depth = 2))
```




```
precis_plot(precis(m4, depth = 2))
```




```
precis_plot(precis(m5, depth = 2))
```



Since all my predictors are completely independent of each other (i.e. there is no other variables which must be conditioned upon when modeling), I've made three models, to assess whether these three are truly independent. If so, there should be little to no predictive power of modeling one variable by another. This seems to be true for sex on class and sex on age, but there seems to be some effect of age on class. Since i have simulated the data, I know that these to variables are independent of each other, so I must conclude that the correlation is merely spurious.

7. Assess whether the DAG is compatible with the data

```
adjustmentSets(dag, exposure = "X", outcome = "S")
```

```
##  {}
```

I do not need to stratify on any variables to get the individual effects of my variables. Below is the total effects model using the real data.

```
# Preprocessing data
df_train <- df_train %>%
  select(Survived, Pclass, Age, Sex)

# Omitting NAs
df_train <- na.omit(df_train)
```

```
# Standardizing
dat = list(

  A = scale(df_train$Age),
  C = as.factor(df_train$Pclass),
  X = as.factor(df_train$Sex),
  S = as.integer(df_train$Survived)
)

# Total effects
m1_real <- ulam(
  alist(

    ## S ~ A + X + C

    S ~ dbinom(1, p),
    logit(p) <- aX[X] + aC[C] + bA*A,
    aX[X] ~ dnorm(0, 0.5),
    aC[C] ~ dnorm(0, 0.5),
    bA ~ dnorm(0, 0.5)

  ), data = dat, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 4.0 seconds.
```

Running the model with the real data yields quite similar parameter values as when using my simulated data. So far so good.

```
impliedConditionalIndependencies(dag)
```

```
## A _||_ C
## A _||_ X
## C _||_ X
```

```
# Testing conditional independencies

# A _||_ C
m6 <- ulam(
  alist(

    ## A ~ C

    A ~ dnorm(mu, sigma),
    mu <- aC[C],
    aC[C] ~ dnorm(0, 0.5),
    sigma ~ dexp(1)

  ), data = dat, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 1.4 seconds.
```

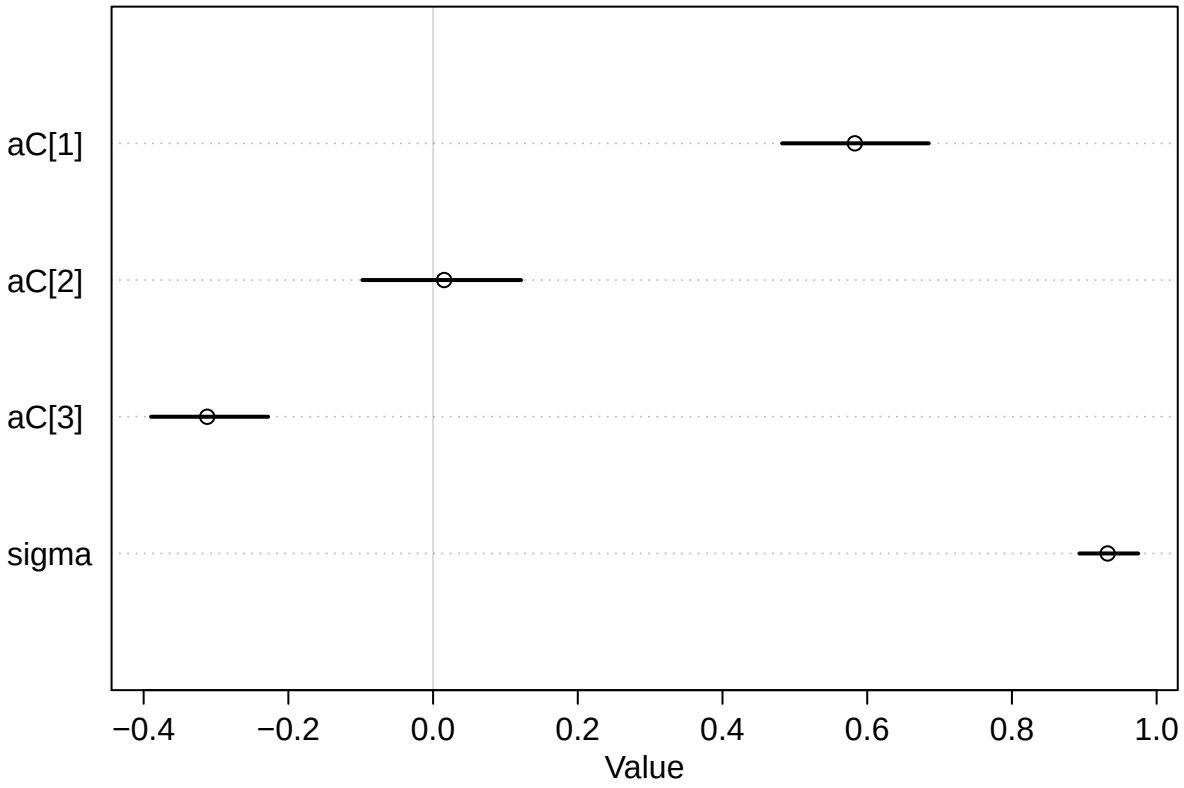
```
# A _||_ X
# I need X as integer for it to be the outcome, otherwise i get an error
dat_X <- dat
dat_X$X <- as.integer(dat_X$X)-1

m7 <- ulam(
  alist(

    ## X ~ A

    X ~ dbinom(1, p),
    logit(p) <- a + bA*A,
    a ~ dnorm(0, 0.2),
    bA ~ dnorm(0, 0.5)

  ), data = dat_X, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 2.1 seconds.
```

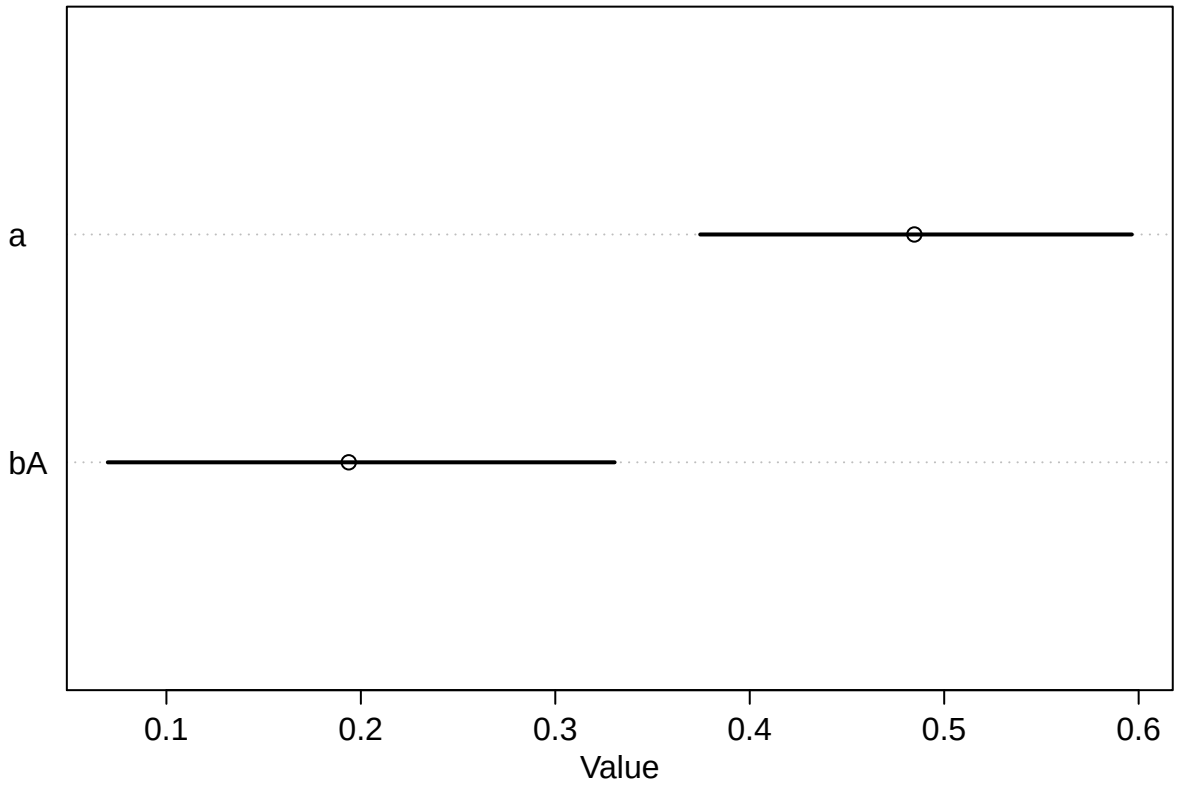
```
# C _||_ X
m8 <- ulam(
  alist(

    ## X ~ C

    X ~ dbinom(1, p),
    logit(p) <- aC[C],
    aC[C] ~ dnorm(0, 0.5)

  ), data = dat_X, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 2.0 seconds.
```

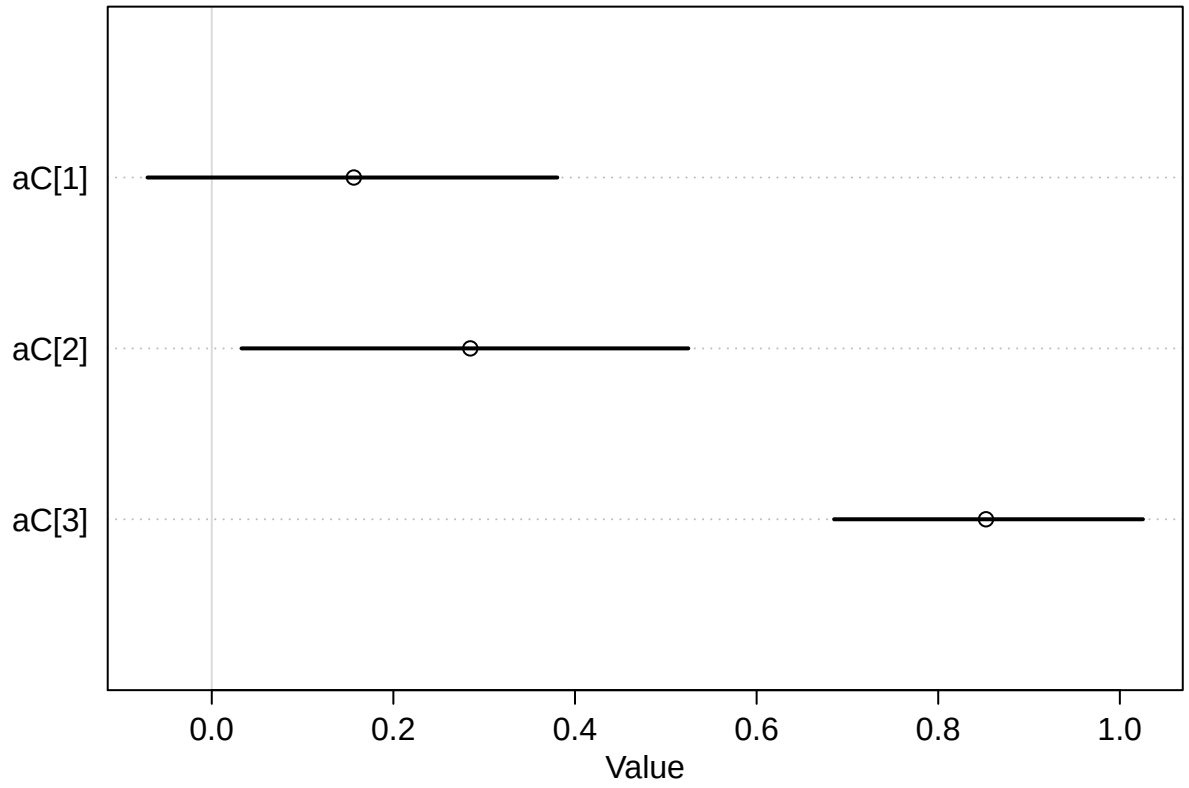
```
precis_plot(precis(m6, depth = 2))
```




```
precis_plot(precis(m7, depth = 2))
```

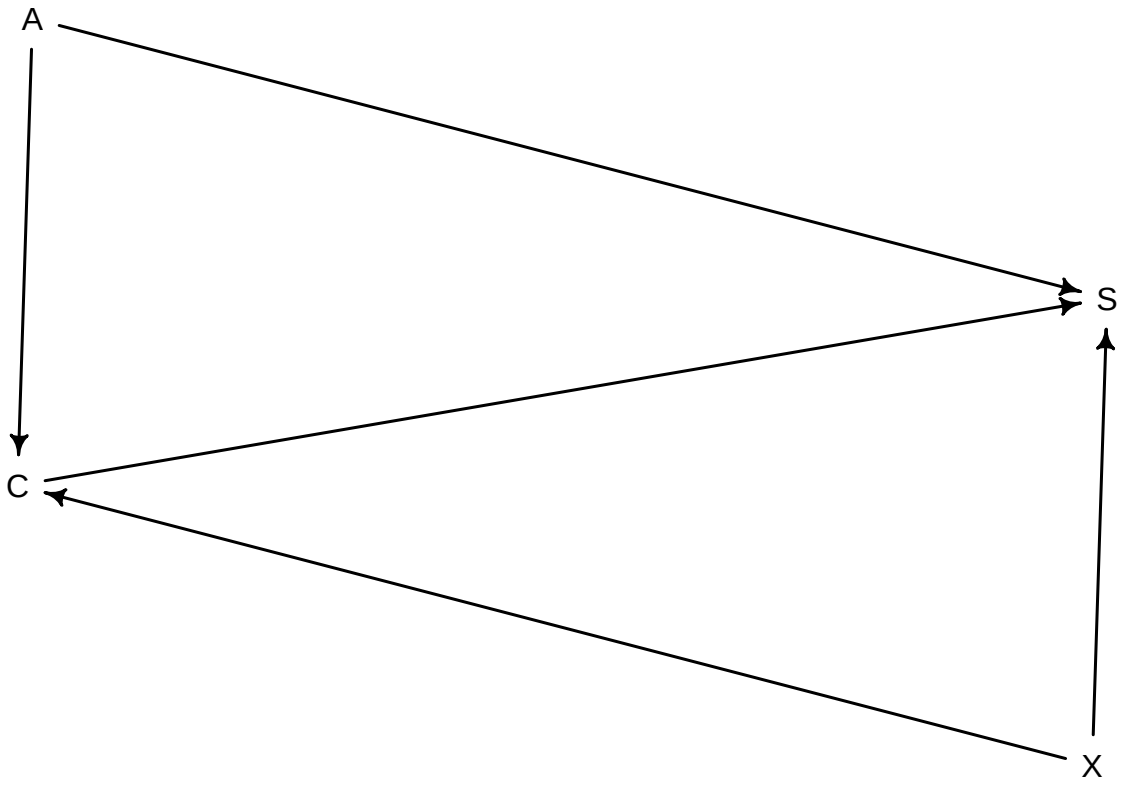



```
precis_plot(precis(m8, depth = 2))
```



It appears that i was too naive. In the real data, both age and sex seems to predict class in some way. I'll incorporate this in a new DAG.

```
dag2 <- dagitty( "dag {
    A -> S
    A -> C
    C -> S
    X -> S
    X -> C
}")

drawdag(dag2)
```



This also makes for a much more interesting DAG. I'll check the conditional independencies

```
impliedConditionalIndependencies(dag2)
```

```
## A _||_ X
```

```
adjustmentSets(dag2, exposure = "A", outcome = "S")
```

```
##  {}
```

```
adjustmentSets(dag2, exposure = "C", outcome = "S")
```

```
## { A, X }
```

```
adjustmentSets(dag2, exposure = "X", outcome = "S")
```

```
##  {}
```

This doesn't change much regarding the statistical model structure, as I now know that i must include all three variables to not have any backdoors open. Therefore i will continue to include all three predictors in my model.

8. Do model comparison

Comparing my models with different priors

```
# Modeling the real data with different priors

# Very narrow priors
m2_real <- ulam(
  alist(

    ## S ~ A + X + C

    S ~ dbinom(1, p),
    logit(p) <- aX[X] + aC[C] + bA*A,
    aX[X] ~ dnorm(0, 0.0001),
    aC[C] ~ dnorm(0, 0.0001),
    bA ~ dnorm(0, 0.0001)

  ), data = dat, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 2.8 seconds.
```

```
# Very broad priors
m3_real <- ulam(
  alist(

    ## S ~ A + X + C

    S ~ dbinom(1, p),
    logit(p) <- aX[X] + aC[C] + bA*A,
    aX[X] ~ dnorm(0, 10),
    aC[C] ~ dnorm(0, 10),
    bA ~ dnorm(0, 10)

  ), data = dat, log_lik = TRUE, refresh = 0
)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 14.0 seconds.
```

```
# Comparing model with different priors

compare(m1_real, m2_real, m3_real)
```

```
##             WAIC           SE       dWAIC       dSE        pWAIC       weight
## m3_real 657.7938 3.137852e+01   0.0000000        NA 5.164043e+00 5.278468e-01
## m1_real 658.0168 2.840432e+01   0.2230049  3.353378 3.916188e+00 4.721532e-01
## m2_real 989.8152 1.108017e-04 332.0214070 31.400565 5.183112e-06 4.216726e-73
```

```
compare(m1_real, m2_real, m3_real, func = PSIS)
```

```
##             PSIS           SE       dPSIS       dSE        pPSIS       weight
## m3_real 657.8901 3.140563e+01   0.0000000        NA 5.212205e+00 5.246856e-01
## m1_real 658.0877 2.842773e+01   0.1976451  3.354852 3.951670e+00 4.753144e-01
## m2_real 989.8152 1.108797e-04 331.9250840 31.405676 5.219084e-06 4.398280e-73
```

Even though the model with too broad priors has lower WAIC and PSIS score than my optimal-prior model, I know that this model doesn't make sense with my knowledge of the data.

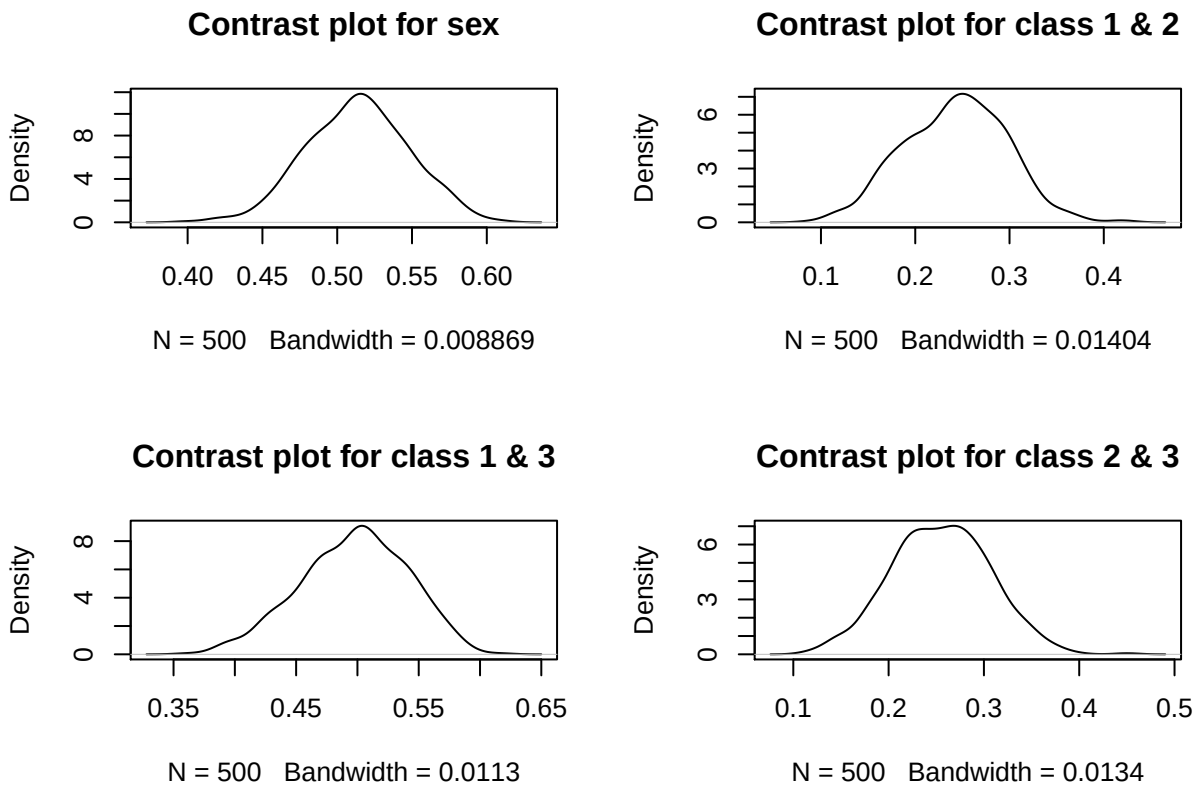
Making contrast plots for the model with optimal priors

```
# Contrast plots
posterior1 <- extract.samples(m1_real)
```

```
precis(m1_real, depth = 3)
```

```
##              mean         sd       5.5%      94.5%    n_eff     Rhat4
## aX[1]  1.19732218 0.23380651  0.8474721  1.5754515 192.6123 0.9997944
## aX[2] -1.10271850 0.22654923 -1.4429459 -0.7313372 152.7867 0.9987861
## aC[1]  1.12221616 0.24673582  0.7624285  1.5230088 162.0975 1.0045081
## aC[2]  0.03744118 0.26092560 -0.3609912  0.4723566 159.4691 0.9983063
## aC[3] -1.08966322 0.24374227 -1.4732511 -0.6996403 172.6997 0.9999650
## bA    -0.45877469 0.09627833 -0.6157826 -0.2965598 284.3390 1.0037926
```

```
par(mfrow=c(2,2))
plot(density(inv_logit(posterior1$aX[,1]) - inv_logit(posterior1$aX[,2])), main="Contrast plot for sex"
plot(density(inv_logit(posterior1$aC[,1]) - inv_logit(posterior1$aC[,2])), main="Contrast plot for class
plot(density(inv_logit(posterior1$aC[,1]) - inv_logit(posterior1$aC[,3])), main="Contrast plot for class
plot(density(inv_logit(posterior1$aC[,2]) - inv_logit(posterior1$aC[,3])), main="Contrast plot for class
```



The contrast plot for sex shows that women have somewhere between 50-55% higher chance of surviving than men. Passengers traveling on 1st class are somewhere between 20-25% more likely to survive than passengers on 2nd class, and somewhere between 47-52% more likely to survive than passengers on 3rd class. Passengers on 2nd class are somewhere between 26-29% more likely to survive than passengers on 3rd class.

Trying out a model with an interaction between sex and class

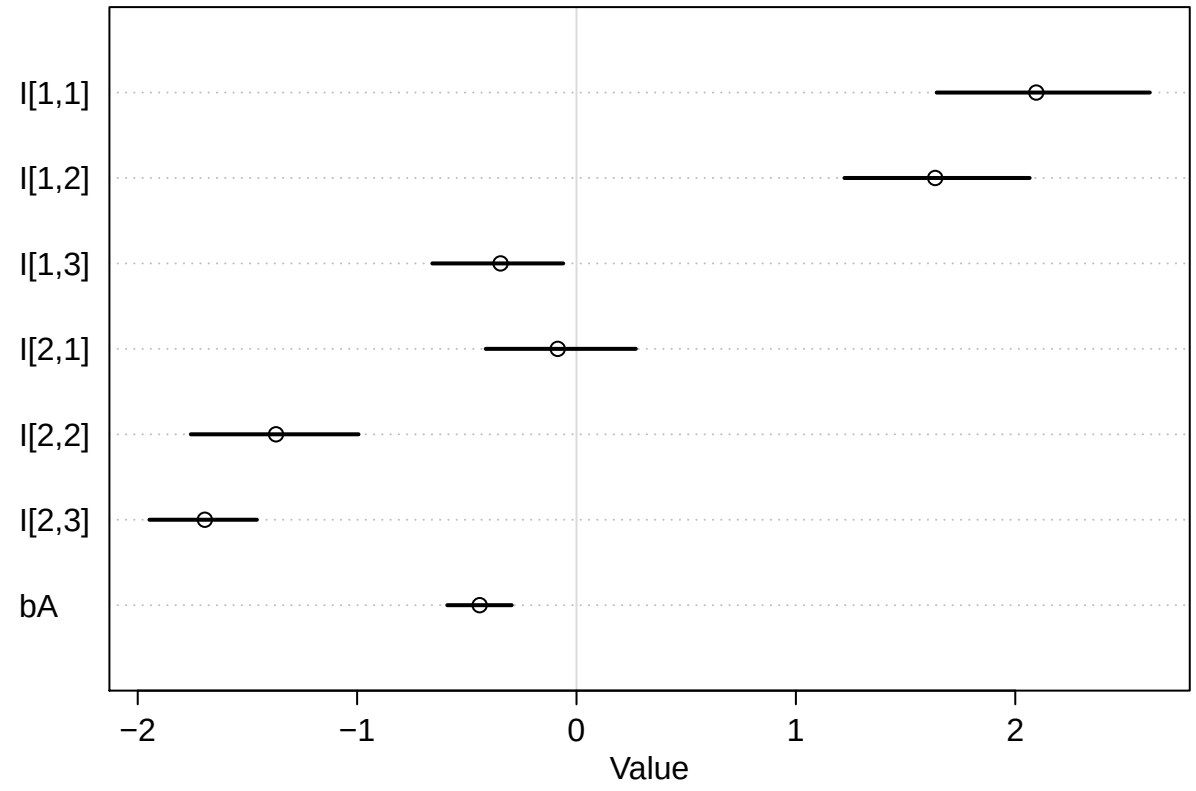
```
# Interaction between sex and class
m1_real_int <- ulam(
  alist(

    #

    S ~ dbinom(1, p),
    logit(p) <- I[X,C]+ bA*A,
    matrix[X,C]:I ~ normal(0, 0.5),
    bA ~ dnorm(0, 0.5)
    ), data=dat, cores = 4, log_lik = TRUE, refresh = 0)
```

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...
##
## Chain 1 finished in 2.4 seconds.
```

```
# Checking the parameter estimates
precis(m1_real_int, depth=3)
```

```
##               mean         sd       5.5%       94.5%    n_eff     Rhat4
## I[1,1]  2.09573442 0.28804123  1.6418713  2.61295635 946.4364 1.0084985
## I[1,2]  1.63500270 0.26523794  1.2212565  2.06553130 620.1547 0.9988637
## I[1,3] -0.34608945 0.18947653 -0.6573058 -0.06086797 516.7923 0.9980583
## I[2,1] -0.08556839 0.20766112 -0.4136982  0.27041044 604.3273 0.9979988
## I[2,2] -1.36954964 0.22995236 -1.7581423 -0.99358299 629.3549 0.9984277
## I[2,3] -1.69413646 0.15272819 -1.9466989 -1.45697130 519.5613 0.9982090
## bA     -0.44141394 0.09813678 -0.5889760 -0.29520627 527.3032 0.9986198
```

```
precis_plot(precis(m1_real_int, 3))
```



The interaction model shows that for both sexes, 1st class is the safest. Though for females, 1st and 2nd class are close to each other, with 3rd class being less safe. For males, both 2nd and 3rd class are relatively unsafe compared to 1st class.

```
# Comparing interaction model to no interaction model
compare(m1_real, m1_real_int)
```

```
##                  WAIC       SE    dWAIC      dSE    pWAIC      weight
## m1_real_int 643.4044 24.72788  0.00000       NA 4.683829 0.999329062
## m1_real     658.0168 28.40432 14.61233 7.935132 3.916188 0.000670938
```

The interaction model also has a slightly lower WAIC score, and a smaller standard error as well.

9. Use the statistical model to do inference

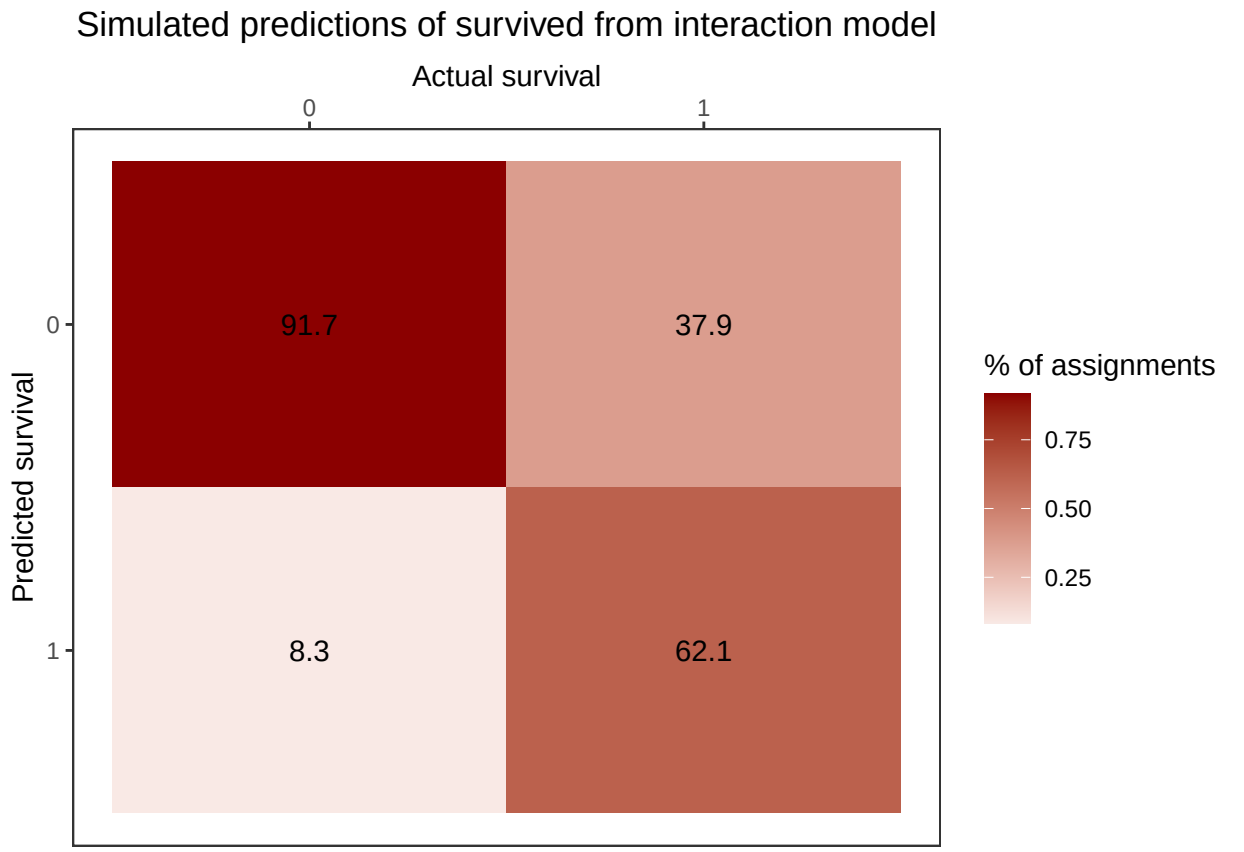
```
# Effect sizes and posterior predictions
post_int <- extract.samples(m1_real_int)
post_precis <- precis(post_int, depth=3)
survive_probability <- inv_logit(post_precis$mean)
Category <- c("Woman 1. Class", "Woman 2. Class", "Woman 3. Class", "Men 1. Class", "Men 2. Class", "Me

df <- tibble(Category, survive_probability)
df <- df %>%
  filter(Category != "Age")

df
```

```
## # A tibble: 6 x 2
##   Category       survive_probability
##   <chr>                        <dbl>
## 1 Woman 1. Class               0.890
## 2 Woman 2. Class               0.479
## 3 Woman 3. Class               0.837
## 4 Men 1. Class                 0.203
## 5 Men 2. Class                 0.414
## 6 Men 3. Class                 0.155
```

```
# Using model to predict survival on new data
survival_pred <- sim(m1_real_int, data=dat)
prediction <- round(colMeans(survival_pred), 0)
survived <- dat$S
survived_df <- data.frame(survived, prediction)
```

```
# Plotting confusion matrix to show predictions
pacman::p_load(caret, scales)
survived_df %>%
  count(survived, prediction) %>%
  mutate(across(c(survived, prediction), ~str_wrap(., 20))) %>%
  group_by(survived) %>%
  mutate(percent = n / sum(n)) %>%
  ggplot(aes(x = survived, y = reorder(prediction, desc(prediction)), fill = percent)) +
  geom_tile() +
  scale_fill_gradient2(high = "darkred") +
  geom_text(aes(label = round(percent*100, digits = 1))) +
  scale_x_discrete(position = "top") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5),
        panel.grid = element_blank(),
        plot.title = element_text(hjust = 0.5)) +
  labs(x = "Actual survival",
       y = "Predicted survival",
```

```
       fill = "% of assignments",
       title = "Simulated predictions of survived from interaction model")
```



The data frame shows the average predictions for survival based on the interaction of class and sex with age as a fixed effect.

Aboard the Titanic, women travelling on 1st and 3rd class had highest probability of surviving. Both men and women on 2nd class were around chance-level of surviving, a bit lower for men though. Ultimately, according to the model, men on 1st and 3rd class had a very small chance of survival. The estimated parameter for age shows that as age increases, people are less likely to survive, although the effect isn't that large.

Unfortunately, my DAG does not fit the data. Through further exploration of the data, better inference could be made.

The table below shows the simulated predictions for survival based on the interaction model. It has high accuracy predicting deaths, but my model doesn't predict survival that well (many false alarms / type 2 errors). This is expected as my data doesn't fit the DAG.