Computational Statistics & Probability

Problem Set 2 - Linear Models
Due: 23:59:59 19.nov.2025

Fall 2025

Instructions

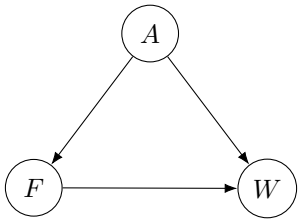
Assignments must be submitted through Canvas. See the course Canvas page for policies covering collaboration, acceptable file formats (`.Rmd` & `.pdf`), and late submissions. Completed assignments must include executable code (`.Rmd`) and a corresponding knitted markdown file (`.pdf`). An R Markdown cheat sheet is available.

1. Multiple Regression & Causal Models

The `foxes` dataset contains data on urban fox populations.

```
# First, loead the foxes dataset
library(rethinking)
data(foxes)
d <- foxes

# Set random seed
set.seed(390)
```

Consider the following hypothesized causal relationship between **territory size** and **body weight** in foxes.



where A, F and W represent random variables `area` (territory size), `avgfood`, and `weight`, respectively.

If this DAG correctly describes the causal relationships, it makes specific predictions about what we should observe in the data. Your task is to test whether the observed patterns match these predictions.

- Territory size (A) has a **direct** effect on weight (W): $A \rightarrow W$
- Food availability (F) has a **direct** effect on weight (W): $F \rightarrow W$
- Territory size (A) has an **indirect** effect on weight (W) through food (F): $A \rightarrow F \rightarrow W$

a) According to the DAG, territory size effects weight through two paths:

- Direct path: $A \rightarrow W$
- Indirect path: $A \rightarrow F \rightarrow W$

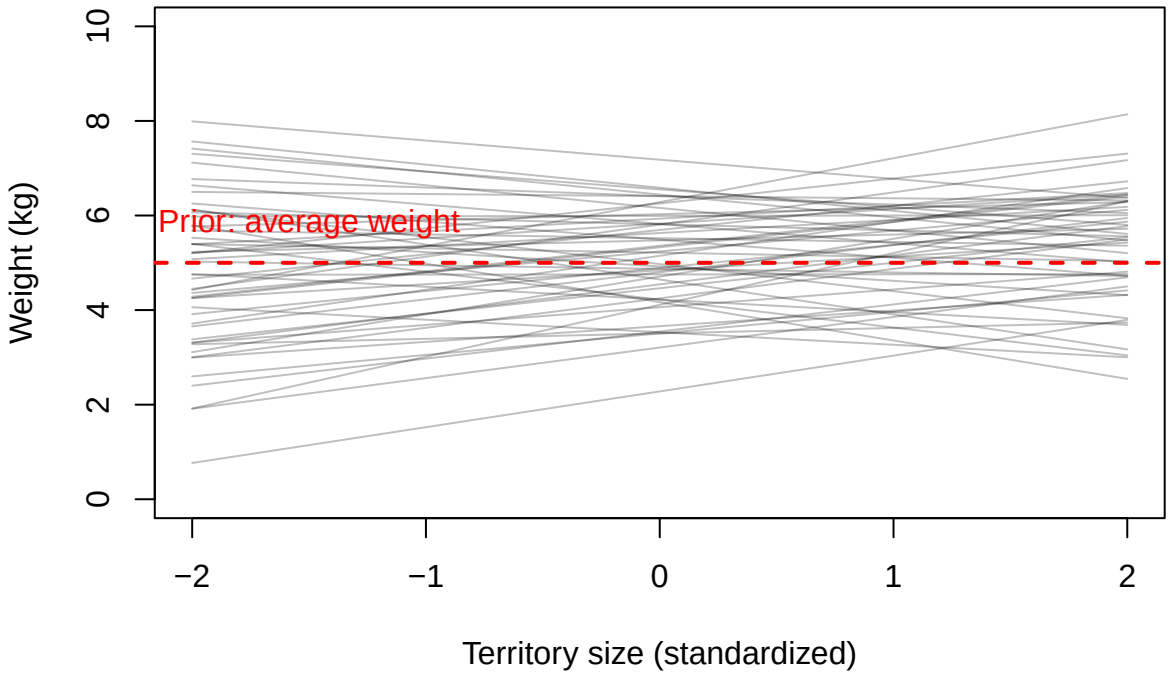
If we regress weight on territory size without including food, the coefficient should capture both pathways, the "total association" between A and W. Construct a linear regression (`m1a`) using `quap`. Urban foxes in

this population have an average weight of 5kg. Use prior predictive simulation to assess the implications of your priors. Standardize the predictor variable.

Question What association do you observe? What does your analysis suggest about how territory size relates to weight?

```
# ANSWER a)
# STEP 1. The first step is to standardize the predictor variable, which is the
# territory size, 'A'.
#
d$A_std <- (d$area - mean(d$area)) / sd(d$area)
```

```
# STEP 2. The second step is to set up priors and run a prior simulation. You
# should also visualize your priors.

# N = 50
n_lines <- 50

# Priors
alpha_prior <- rnorm(n_lines, 5, 1)
beta_A_prior <- rnorm(n_lines, 0, 0.5)
sigma_prior <- rexp(n_lines, 1)

# Sample territory sizes (standardized)
A_seq <- seq(from=-2, to=2, length.out=50)

plot(NULL, xlim=c(-2, 2), ylim=c(0, 10),
     xlab="Territory size (standardized)",
     ylab="Weight (kg)",
     main="Prior predictive simulation for m1a")

# Prior predictions
for (i in 1:n_lines) {
  mu <- alpha_prior[i] + beta_A_prior[i] * A_seq
  lines(A_seq, mu, col=col.alpha("black", 0.25))
}

# Plot
abline(h=5, lty=2, col="red", lwd=2)
text(-1.5, 5.25, "Prior: average weight", pos=3, col="red")
```



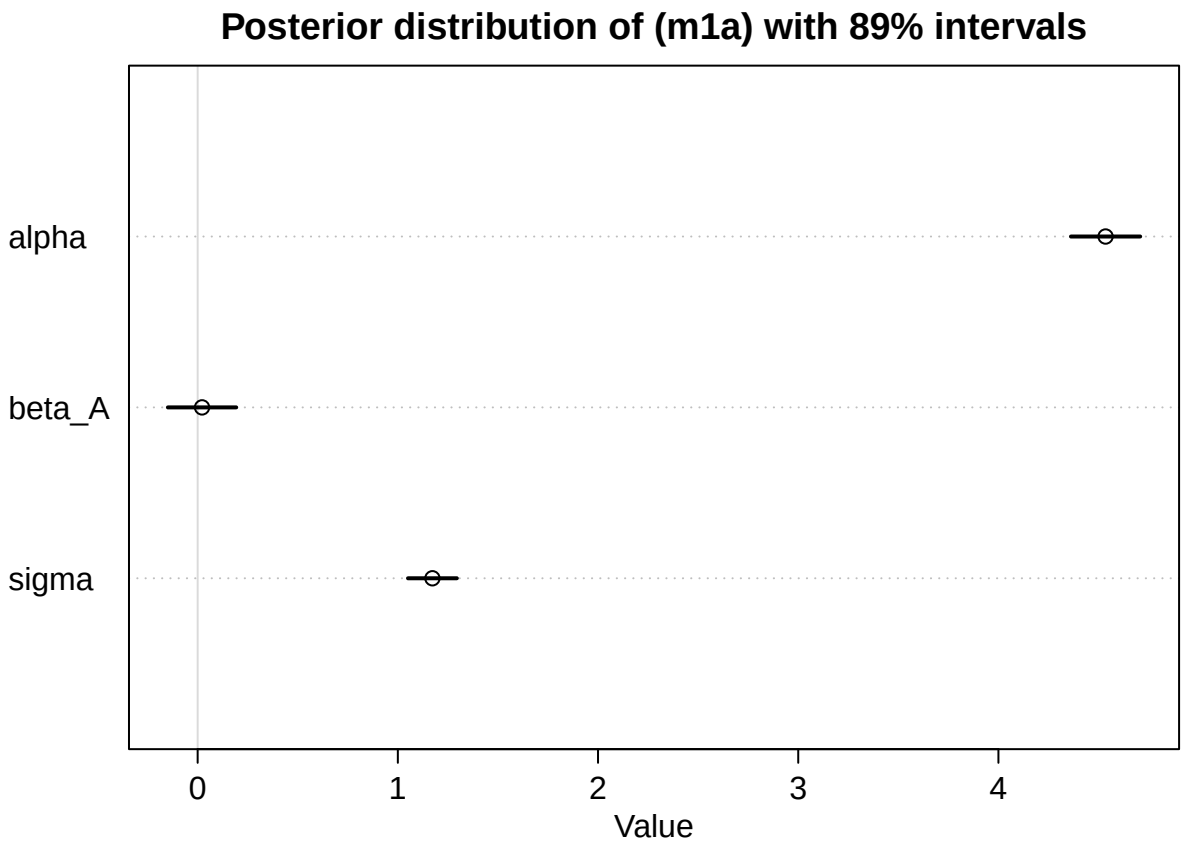

```
# NOTE: It is critical to use your external knowledge of the weight of urban
# foxes, NOT the sample mean.  You are told in the problem description in a)
# that the average weight of the fox population is 5kg. This is the correct
# prior to use, not the sample average of 4.53kg.

# Why? The reason is that using the sample average is "peaking" at the data.
# But your prior represents your knowledge BEFORE seeing the data you are analyzing.
# So, using `mean(d$weight)` to set your prior lets the data inform your prior, which
# you are using to analyze that same data. This is circular reasoning: you are letting
# the data speak twice, which can lead to over-confident priors and defeats the whole
# purpose of Bayesian updating.

# Similarly, you were instructed to ONLY standardize the predictor variable and
# not both the predictor and target variance, weight. Part (a) specifically asks you
# to use your knowledge that 'urban foxes have an average weight of 5kg' to inform
# your priors. This is only possible when weight remains on its natural scale.
# If instead you standardize, what you are saying is that you believe that the
# population mean is near the sample mean. This defeats the purpose of prior
# simulation as a tool for incorporating domain knowledge: on the contrary, you are
# explicitly ignoring the prior knowledge that the average weight of foxes is 5kg.

# Looking ahead, contrast with part (b), which introduces standardized outcomes for
# a specific reason: to interpret slopes as standardized effect sizes. In this case,
# you are comparing effect magnitudes.

# In Bayesian workflows, the sequence matters: first elicit priors using domain
# knowledge on natural scales, then transform if needed for computational or
# comparative purposes.
```

```r
# STEP 3. Fit the model with quap, using the priors you decided upon from your
# prior simulation in STEP 2.
m1a <- quap(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- alpha + beta_A * A_std,
    alpha ~ dnorm(5, 1),
    beta_A ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ),
  data = d
)

# STEP 4. Examine the posterior. Note that only at this stage do you inspect
# the distribution of the data.

# Summary table
precis(m1a)
```

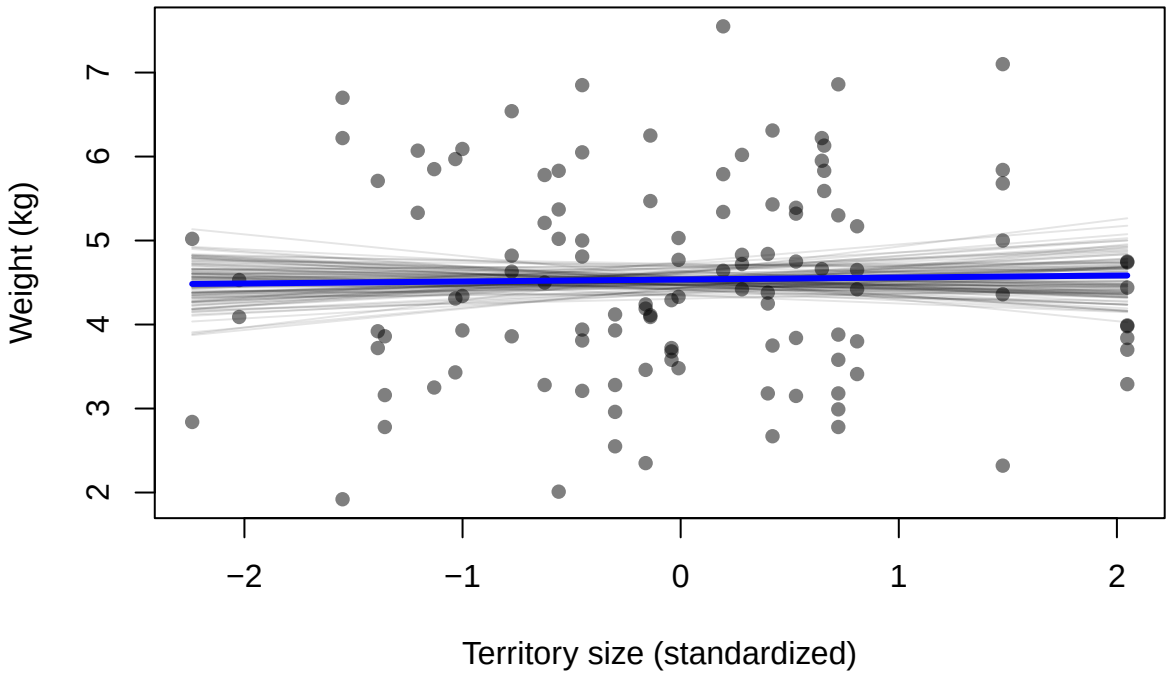
```
##              mean         sd       5.5%     94.5%
## alpha  4.53517829 0.10825055  4.3621730 4.7081836
## beta_A 0.02204517 0.10683495 -0.1486977 0.1927881
## sigma  1.17276021 0.07641734  1.0506305 1.2948899
```

```r
# Sample from the posterior distribution of m1a
post <-extract.samples(m1a)
```

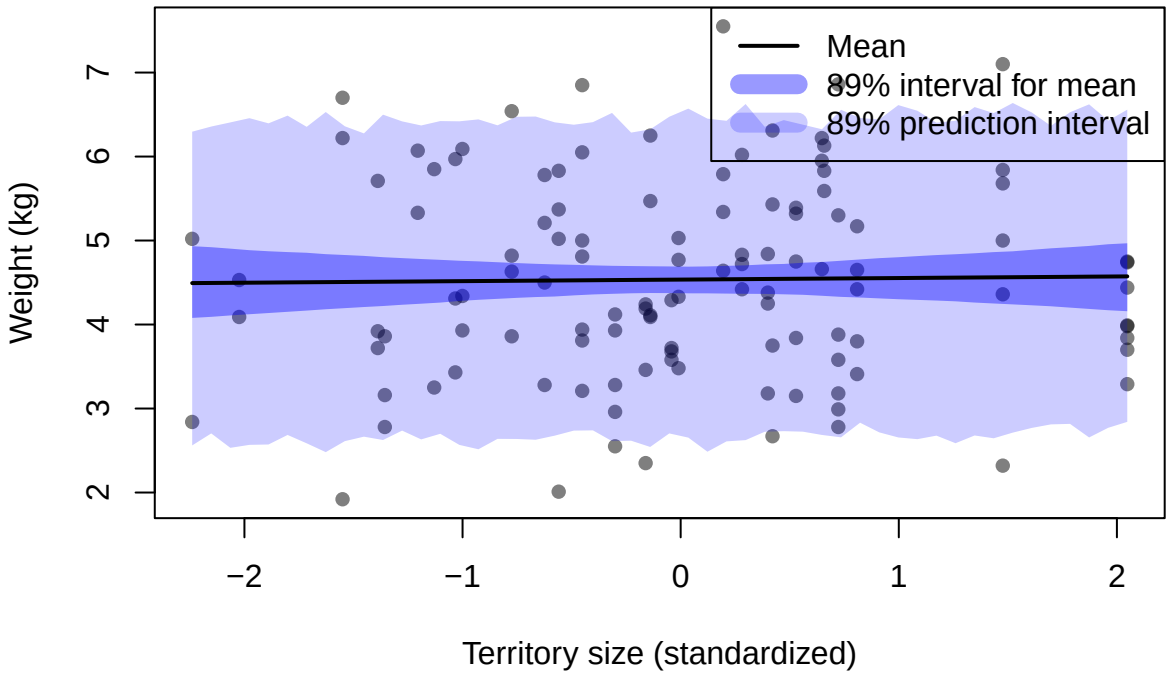
```r
# STEP 5. Assessing the causal influence of beta_A
plot(precis(m1a), main="Posterior distribution of (m1a) with 89% intervals")
```




```r
# INTERPRETATION: The causal effect of territorial area (beta_A) is effectively zero.
# The point estimate is close to 0, and both small positive and small negative values
# are included in 89% credibility interval of the posterior.

# Notice that the alpha intercept is updated from our prior of 5kg to a value (4.54kg)
# approaching the sample mean of approximately 4.53kg

# STEP 6. Plotting posterior predictions of (m1a) overlaying data.
plot(d$A_std, d$weight,
     xlab="Territory size (standardized)",
     ylab="Weight (kg)",
     main="Posterior predictions from m1a",
     col=col.alpha("black", 0.5), pch=16)

# Sequence for predictions
A_seq <- seq(from=min(d$A_std), to=max(d$A_std), length.out=50)

# Sample posterior regression lines
for (i in 1:100) {
  mu <- post$alpha[i] + post$beta_A[i] * A_seq
  lines(A_seq, mu, col=col.alpha("black", 0.1))
}

# Add mean regression line
mu <- link(m1a, data=list(A_std=A_seq))
mu_mean <- apply(mu, 2, mean)
lines(A_seq, mu_mean, lwd=3, col="blue")
```



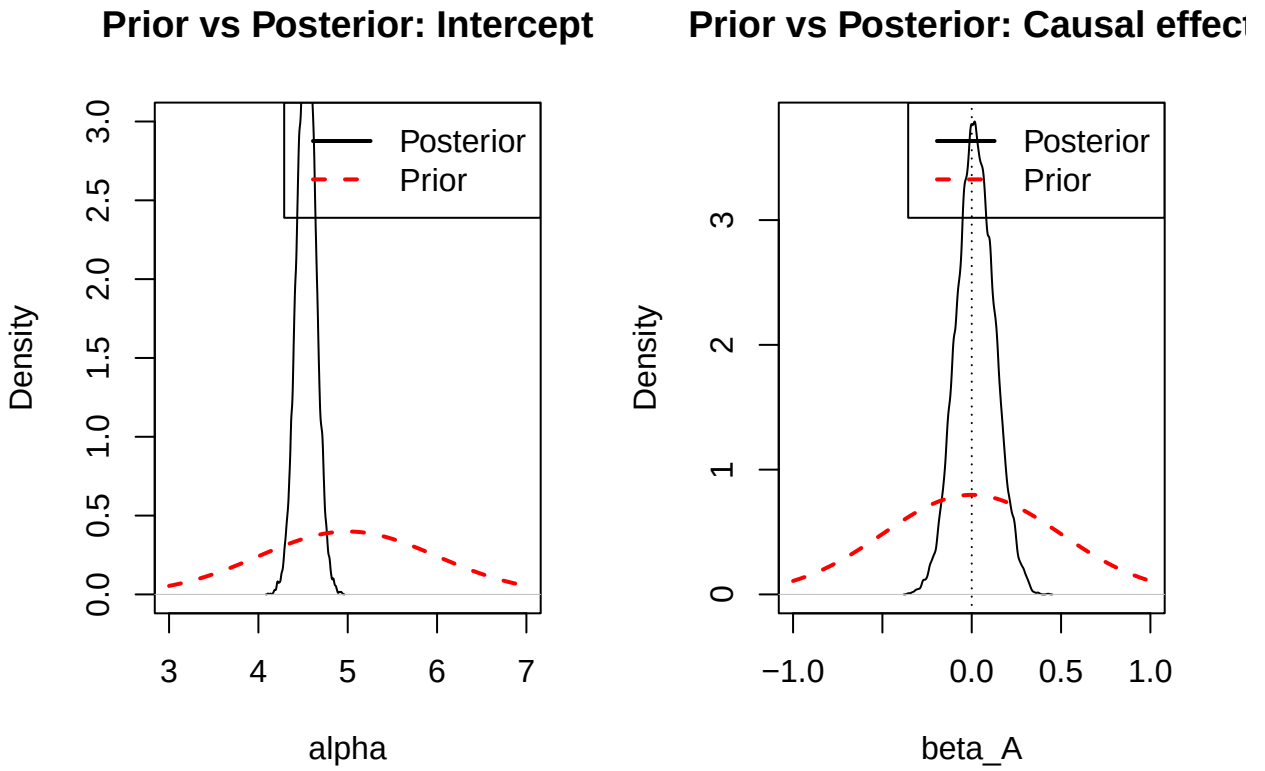

```r
# STEP 6 (continued): The next plot includes uncertainty

##
plot(d$A_std, d$weight,
     xlab="Territory size (standardized)",
     ylab="Weight (kg)",
     main="Posterior predictions with uncertainty",
     col=col.alpha("black", 0.5), pch=16)

# Mean and interval for mu (mean weight)
mu <- link(m1a, data=list(A_std=A_seq))
mu_mean <- apply(mu, 2, mean)
mu_PI <- apply(mu, 2, PI, prob=0.89)

# Prediction interval (includes sigma - sampling variability)
weight_sim <- sim(m1a, data=list(A_std=A_seq))
weight_PI <- apply(weight_sim, 2, PI, prob=0.89)

# Plot intervals
shade(weight_PI, A_seq, col=col.alpha("blue", 0.2))  # Prediction interval
shade(mu_PI, A_seq, col=col.alpha("blue", 0.4))       # Uncertainty in mean
lines(A_seq, mu_mean, lwd=2)                          # Mean line

legend("topright",
       legend=c("Mean", "89% interval for mean", "89% prediction interval"),
       lwd=c(2, 10, 10),
       col=c("black", col.alpha("blue", 0.4), col.alpha("blue", 0.2)))
```




```
# NOTE: The difference between `link()`` and `sim()`:

# link() - Uncertainty about the MEAN
mu <- link(m1a, data=list(A_std=A_seq))
# This samples: mu[i] = alpha[i] + beta_A[i] * A_std
# Only parameter uncertainty, no sigma

# sim() - Uncertainty about INDIVIDUAL OBSERVATIONS
weight_sim <- sim(m1a, data=list(A_std=A_seq))
# This samples: mu[i] = alpha[i] + beta_A[i] * A_std
#               weight[i] ~ Normal(mu[i], sigma[i])
# Parameter uncertainty PLUS sampling variability
```

```
# BONUS: The next plot helps to solidify intuitions about the results of m1a by
# plotting the comparison of prior to posterior for the intercept term, which
# encodes an estimate of the *population* weight, and the slope term, which
# encodes the influence of territory size on the weight of foxes.

par(mfrow=c(1,2))
# Alpha comparison
dens(post$alpha, xlab="alpha", main="Prior vs Posterior: Intercept",
     xlim=c(3, 7), ylim=c(0, 3))
curve(dnorm(x, 5, 1), add=TRUE, col="red", lwd=2, lty=2)
legend("topright", legend=c("Posterior", "Prior"),
       col=c("black", "red"), lwd=2, lty=c(1,2))

# Beta_A comparison
dens(post$beta_A, xlab="beta_A",
     main="Prior vs Posterior: Causal effect",
     xlim=c(-1, 1))
```

```r
curve(dnorm(x, 0, 0.5), add=TRUE, col="red", lwd=2, lty=2)
abline(v=0, lty=3)
legend("topright", legend=c("Posterior", "Prior"),
       col=c("black", "red"), lwd=2, lty=c(1,2))
```



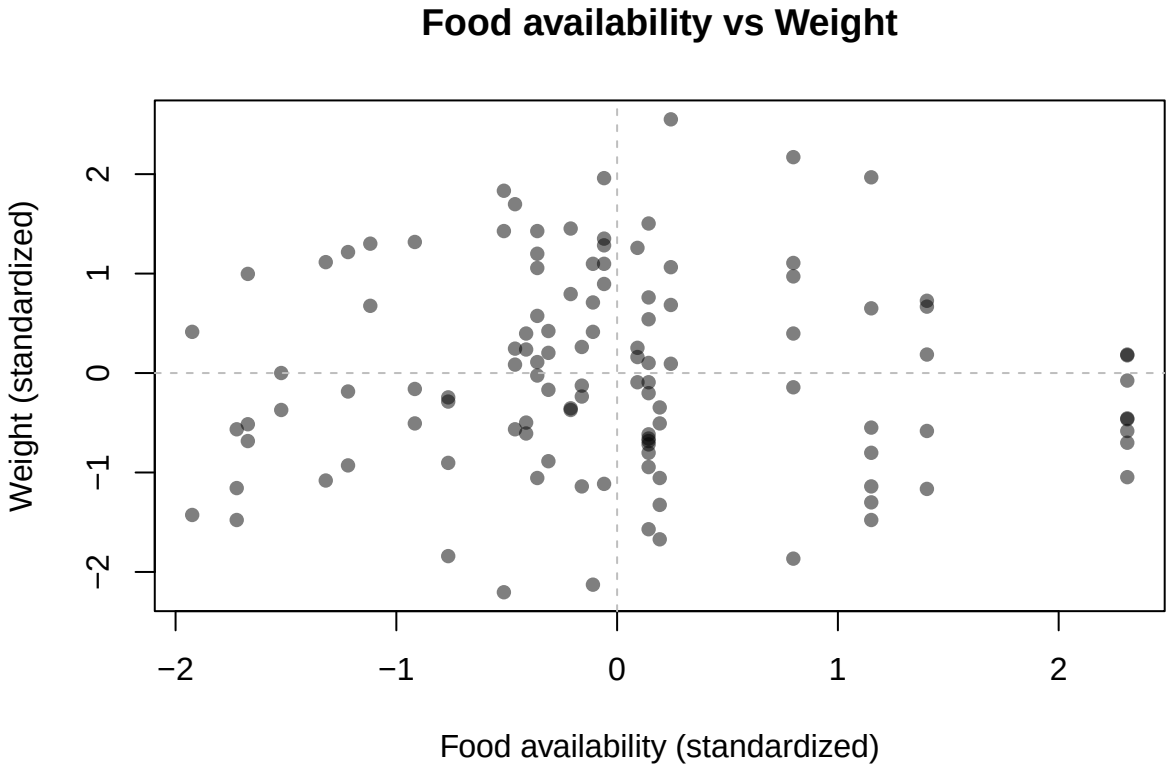

```r
par(mfrow=c(1,1))
```

b) Regress weight on food availability. That is, construct a `quap` linear regression (`m1b`) to estimate the association of food availability and fox weight. *Before fitting the model*, standardize both `avgfood` and `weight` to have mean 0 and standard deviation 1.

Hint: *With standardized variables, regression slopes represent standardized effect sizes. A slope of 1.0 would indicate a perfect positive relationship, while slopes >2 would be implausibly large for most ecological relationships.*

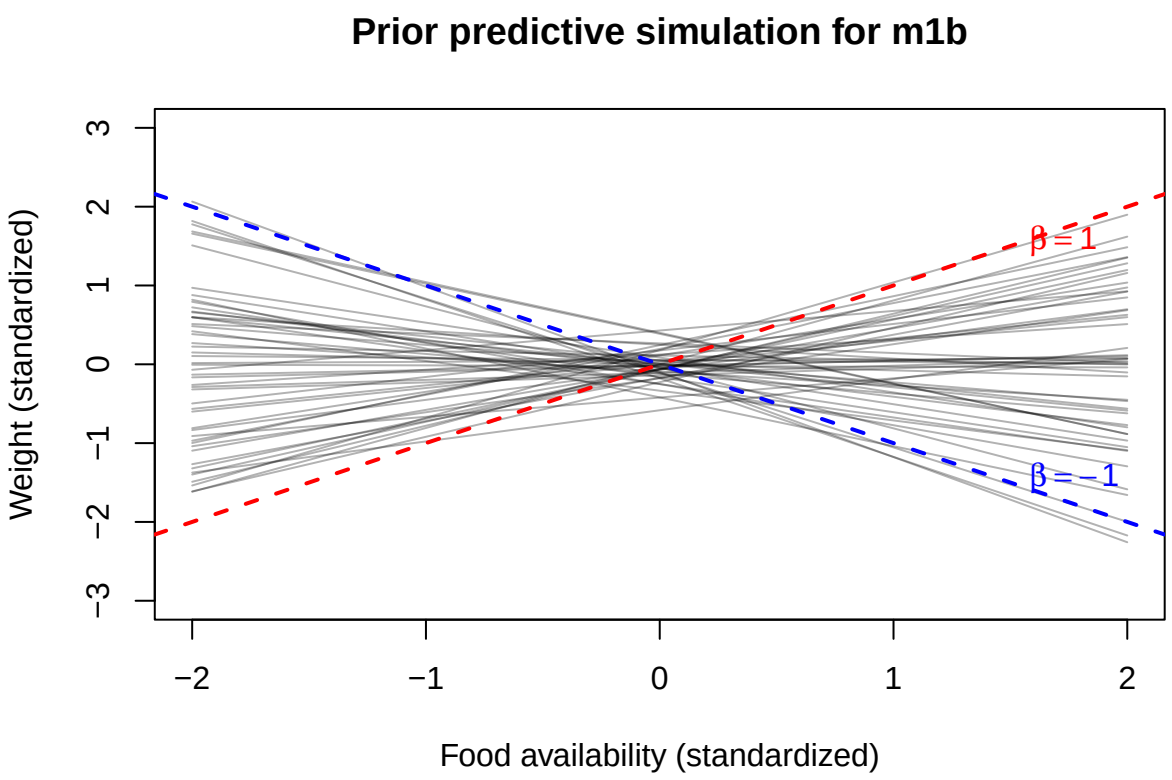
Use prior predictive simulation to assess the implication of your priors. Write 1-2 sentences to justify your priors.

```r
# STEP 1. Standardize both weight and average food availability
d$F_std <- (d$avgfood - mean(d$avgfood)) / sd(d$avgfood)
d$W_std <- (d$weight - mean(d$weight)) / sd(d$weight)

# STEP 2. Visualize

plot(d$F_std, d$W_std,
     xlab="Food availability (standardized)",
     ylab="Weight (standardized)",
     main="Food availability vs Weight",
     col=col.alpha("black", 0.5), pch=16)
abline(h=0, v=0, lty=2, col="gray")
```




```
# STEP 2. The second step is to set up priors and run a prior simulation. You
# should also visualize your priors.

# Prior predictive check:
# Most prior regression lines should fall between perfect positive (beta = 1) and
# perfect negative (beta = -1) correlations.

# N = 50
n_lines <- 50

# Priors
alpha_prior <- rnorm(n_lines, 0, 0.2)
beta_F_prior <- rnorm(n_lines, 0, 0.5)
sigma_prior <- rexp(n_lines, 1)

# Sample territory sizes (standardized)
F_seq <- seq(from=-2, to=2, length.out=50)

plot(NULL, xlim=c(-2, 2), ylim=c(-3, 3),
     xlab="Food availability (standardized)",
     ylab="Weight (standardized)",
     main="Prior predictive simulation for m1b")

# Prior predictions
for (i in 1:n_lines) {
  mu <- alpha_prior[i] + beta_F_prior[i] * F_seq
  lines(F_seq, mu, col=col.alpha("black", 0.3))
}
```

```r
# Plot
abline(a=0, b=1, col="red", lwd=2, lty=2)  # Perfect positive correlation
abline(a=0, b=-1, col="blue", lwd=2, lty=2) # Perfect negative correlation
text(1.5, 1.5, expression(beta == 1), col="red", pos=4)
text(1.5, -1.5, expression(beta == -1), col="blue", pos=4)
```




```r
# Check plausibility of prior predictions
prior_sims <- sapply(1:1000, function(i) {
  a <- rnorm(1, 0, 0.2)
  b <- rnorm(1, 0, 0.5)
  # Weight at +2 SD food availability
  a + b * 2
})

# At +2 SD food availability:
cat("  Mean predicted weight:", round(mean(prior_sims), 2), "SD\n")
```

```
##   Mean predicted weight: -0.03 SD
```

```r
cat("  89% PI: [", round(PI(prior_sims, 0.89)[1], 2), ",",
    round(PI(prior_sims, 0.89)[2], 2), "]\n\n")
```

```
##   89% PI: [ -1.62 , 1.53 ]
```

```r
# DISCUSSION: The prior predictive check displays what your priors imply BEFORE
# SEEING THE DATA. The plot shows a central tendency around zero: at food = 0,
# most lines intersect at (0,0).

# Mean prediction at +2 SD food = -0.06 SD weight, which indicates that your prior
# has essentially no directional bias.  Your prior is NOT assuming food increases
# or decreases weight a prior.
```

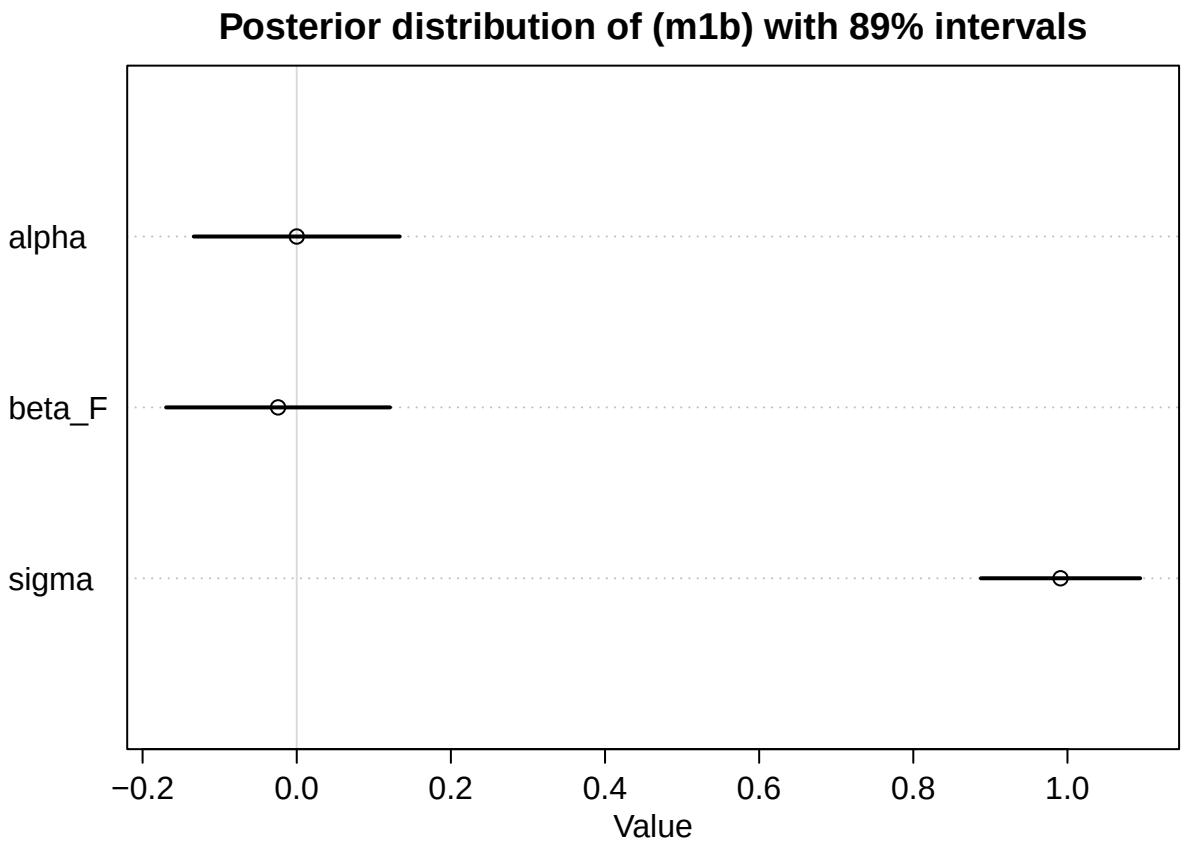
```r
# At +2 SD food, the prior allows weights to range -1.76 to +1.76 SD -- a ~3.4 SD
# range of plausible outcomes.

# Most lines fall between beta +/- 1, which means that my prior is skeptical of
# perfect or near perfect correlations, while still allowing strong effects in either
# direction.

# The upshot is that this is wide enough to be flexible, but narrow enough to be
# sensible.
```

```r
# STEP 3. Fit the model with quap, using the priors you decided upon from your
# prior simulation in STEP 2.

m1b <- quap(
  alist(
    W_std ~ dnorm(mu, sigma),
    mu <- alpha + beta_F * F_std,
    alpha ~ dnorm(0, 0.2),
    beta_F ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ),
  data = d
)

# STEP 4. Examine the posterior. Note that only at this stage do you inspect
# the distribution of the data.

# Summary table
precis(m1b)
```

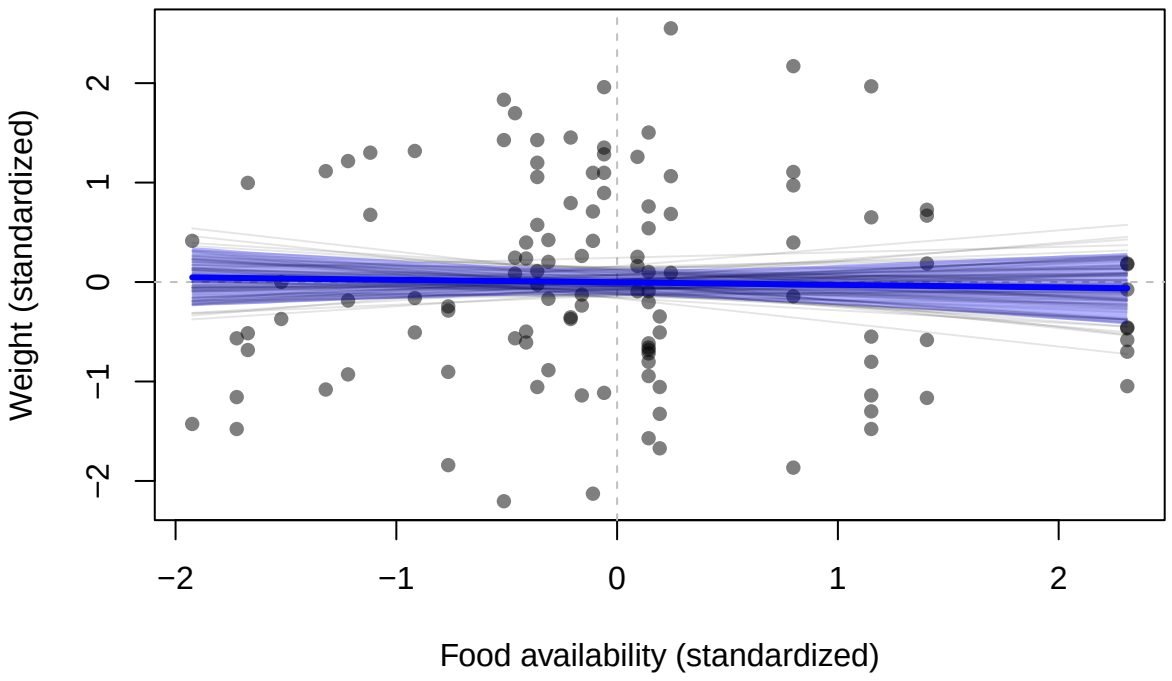
```
##                mean         sd       5.5%     94.5%
## alpha   5.878621e-06 0.08358981 -0.1335868 0.1335985
## beta_F -2.421319e-02 0.09087182 -0.1694439 0.1210175
## sigma   9.909951e-01 0.06463437  0.8876969 1.0942933
```

```r
# Sample from the posterior distribution of m1a
post <-extract.samples(m1b)
```

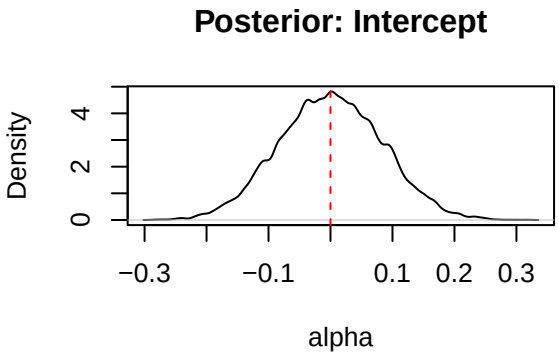
```r
# STEP 5. Assessing the causal influence of beta_A
plot(precis(m1b), main="Posterior distribution of (m1b) with 89% intervals")
```

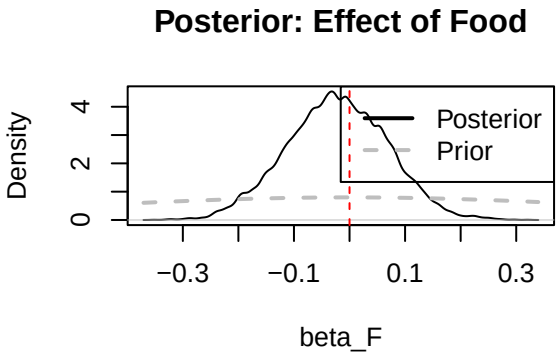


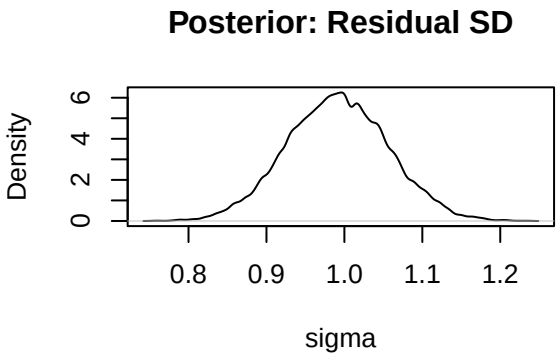

```
# INTERPRETATION: The causal effect of food availability (beta_F) is effectively zero.
# The point estimate, while negative, includes both small positive and small negative
# values in the 89% credibility interval of the posterior.

# STEP 6. Plotting posterior predictions of (m1a) overlaying dat
plot(d$F_std, d$W_std,
     xlab="Food availability (standardized)",
     ylab="Weight (standardized)",
     main="Model m1b: Total causal effect of food on weight",
     col=col.alpha("black", 0.5), pch=16)
abline(h=0, v=0, lty=2, col="gray")

# Posterior regression lines (sample 100)
F_seq <- seq(from=min(d$F_std), to=max(d$F_std), length.out=50)
for (i in 1:100) {
  mu <- post$alpha[i] + post$beta_F[i] * F_seq
  lines(F_seq, mu, col=col.alpha("black", 0.1))
}

# Add mean line and intervals
mu <- link(m1b, data=list(F_std=F_seq))
mu_mean <- apply(mu, 2, mean)
mu_PI <- apply(mu, 2, PI, prob=0.89)

lines(F_seq, mu_mean, lwd=3, col="blue")
shade(mu_PI, F_seq, col=col.alpha("blue", 0.3))
```




```
# BONUS: The next plots compare prior to posterior for the intercept all terms.

post <- extract.samples(m1b)

par(mfrow=c(2,2))

dens(post$alpha, xlab="alpha", main="Posterior: Intercept")
abline(v=0, col="red", lty=2)

dens(post$beta_F, xlab="beta_F", main="Posterior: Effect of Food")
abline(v=0, col="red", lty=2)
# Add prior for comparison
curve(dnorm(x, 0, 0.5), add=TRUE, col="gray", lwd=2, lty=2)
legend("topright", legend=c("Posterior", "Prior"),
       col=c("black", "gray"), lwd=2, lty=c(1,2))

dens(post$sigma, xlab="sigma", main="Posterior: Residual SD")

par(mfrow=c(1,1))
```








```
# TOTAL CAUSAL EFFECT OF FOOD AVAILABILITY ON WEIGHT\n")

cat("Posterior mean for beta_F:", round(mean(post$beta_F), 3), "\n")
```

```
## Posterior mean for beta_F: -0.023
```

```
cat("89% CI: [", round(PI(post$beta_F, prob=0.89)[1], 3), ",",
    round(PI(post$beta_F, prob=0.89)[2], 3), "]\n")
```

```
## 89% CI: [ -0.17 , 0.121 ]
```

```
cat("95% CI: [", round(PI(post$beta_F, prob=0.95)[1], 3), ",",
    round(PI(post$beta_F, prob=0.95)[2], 3), "]\n\n")
```

```
## 95% CI: [ -0.2 , 0.153 ]
```

```
# Probability of positive vs negative effect
prob_positive <- sum(post$beta_F > 0) / length(post$beta_F)
prob_negative <- sum(post$beta_F < 0) / length(post$beta_F)

cat("Probability of positive effect:", round(prob_positive, 3), "\n")
```

```
## Probability of positive effect: 0.402
```

```
cat("Probability of negative effect:", round(prob_negative, 3), "\n\n")
```

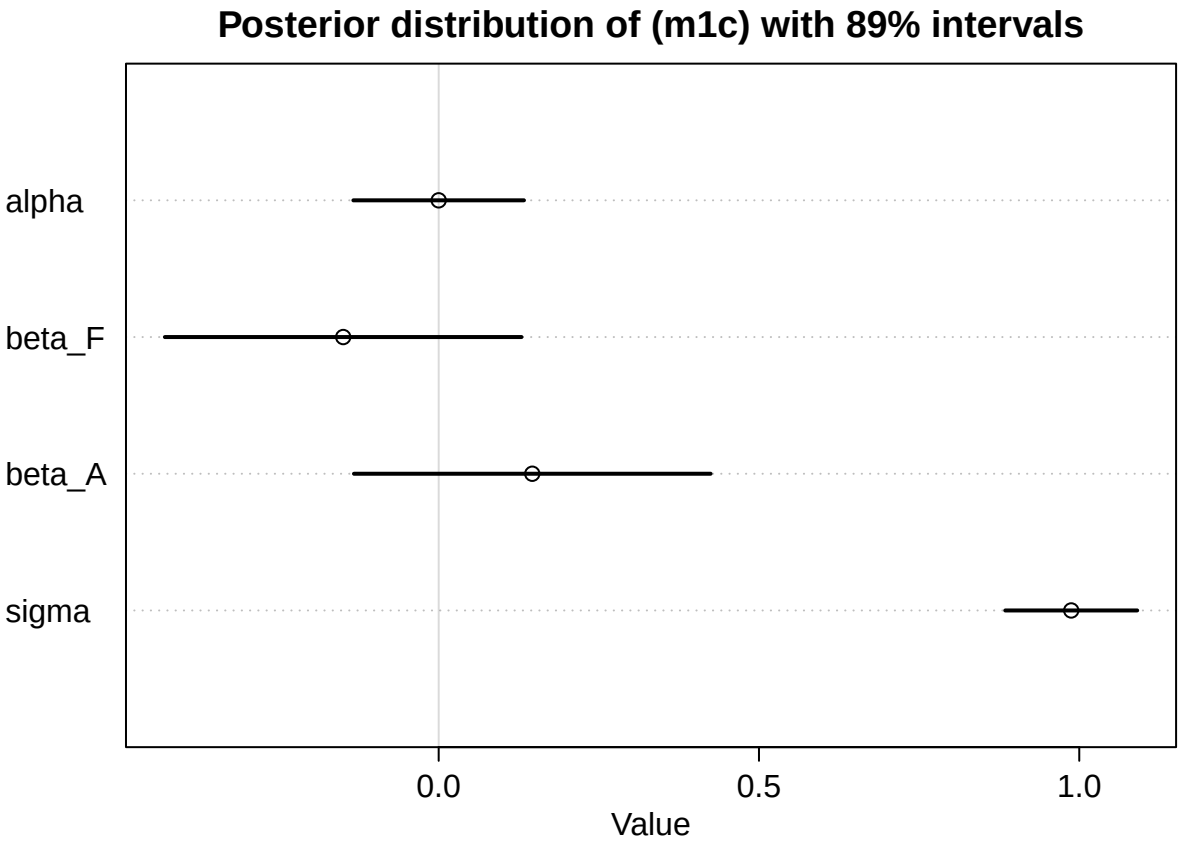
```
## Probability of negative effect: 0.598
```

```
# INTERPRETATION:
# The 89% credible interval includes zero [-0.173, 0.122], indicating
# we cannot confidently conclude that food availability has any effect
# on weight in either direction. The posterior is fairly evenly split
# between positive (39%) and negative (61%) effects, with the point
```

```
# estimate essentially at zero (-0.025).

# This is a 'precise null' - the narrow credible interval tells us we
# have adequate data, but the effect (if any) is too small to distinguish
# from zero. We can confidently say that any effect of food on weight
# is small (< 0.2 SD), but we cannot determine its direction.
```

c) Now regress weight on *both* territory size and food availability. Construct a `quap` model (`m1c`) that includes both predictors. Use the standardized variables. Explain your findings with 3-4 sentences and appropriate plots.

```
# STEP 1. Standardize variables.  All variables have been standardized and are
# in the global environment:

# A_std
# F_std
# W_std

# STEP 2. The second step is to set up priors and run a prior simulation. We will
# use the priors we have simulated and selected earlier.

# STEP 3. Fit the model with quap, using the priors you decided upon from your
# prior simulation in STEP 2.

m1c <- quap(
  alist(
    W_std ~ dnorm(mu, sigma),
    mu <- alpha + beta_A * A_std + beta_F * F_std,
    alpha ~ dnorm(0, 0.2),
    beta_F ~ dnorm(0, 0.5),
    beta_A ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ),
  data = d
)

plot(precis(m1c),  main="Posterior distribution of (m1c) with 89% intervals")
```




```
# Compare all three models:
precis(m1a)  # beta_A approx 0.02 (essentially zero)
```

```
##              mean         sd       5.5%     94.5%
## alpha  4.53517829 0.10825055  4.3621730 4.7081836
## beta_A 0.02204517 0.10683495 -0.1486977 0.1927881
## sigma  1.17276021 0.07641734  1.0506305 1.2948899
```

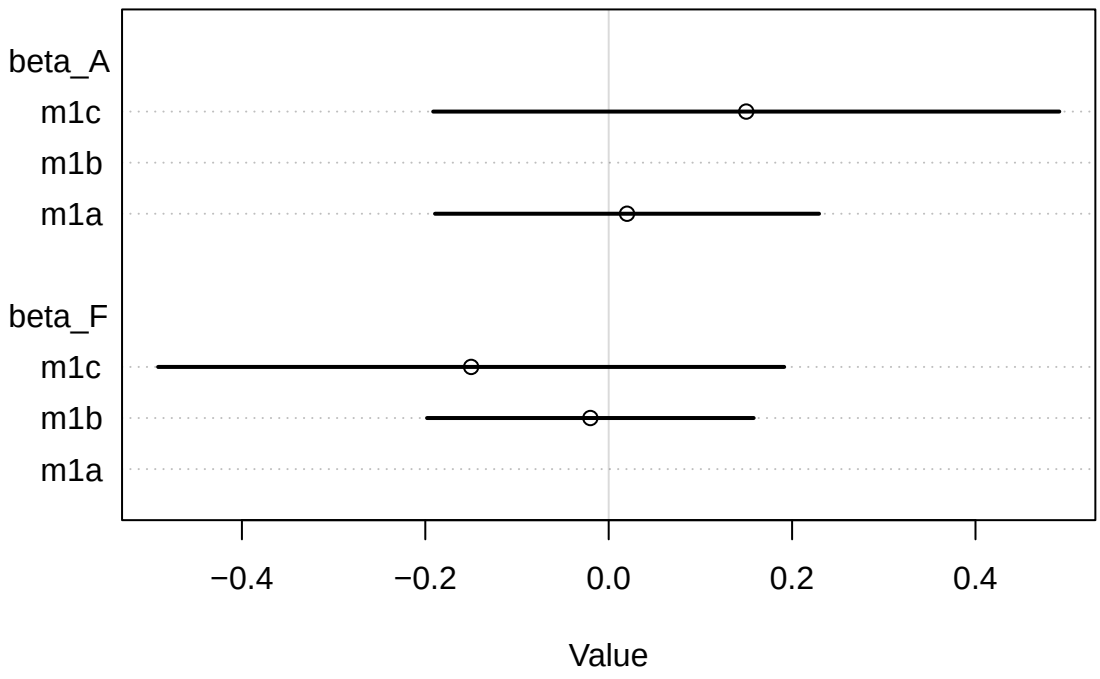
```
precis(m1b)  # beta_F approx -0.02 (essentially zero)
```

```
##                 mean         sd       5.5%     94.5%
## alpha   5.878621e-06 0.08358981 -0.1335868 0.1335985
## beta_F -2.421319e-02 0.09087182 -0.1694439 0.1210175
## sigma   9.909951e-01 0.06463437  0.8876969 1.0942933
```

```
precis(m1c)  # beta_A approx 0.15 (positive!), beta_F approx -0.15 (slightly positive)
```

```
##                 mean         sd       5.5%     94.5%
## alpha  -8.045762e-08 0.08334407 -0.1332000 0.1331998
## beta_F -1.490385e-01 0.17418850 -0.4274254 0.1293484
## beta_A  1.461374e-01 0.17418834 -0.1322492 0.4245240
## sigma   9.874684e-01 0.06444176  0.8844780 1.0904588
```

```
# Visualize the comparison
plot(coeftab(m1a, m1b, m1c), pars=c("beta_A", "beta_F"),  main="All three posterior distributions with 8
```



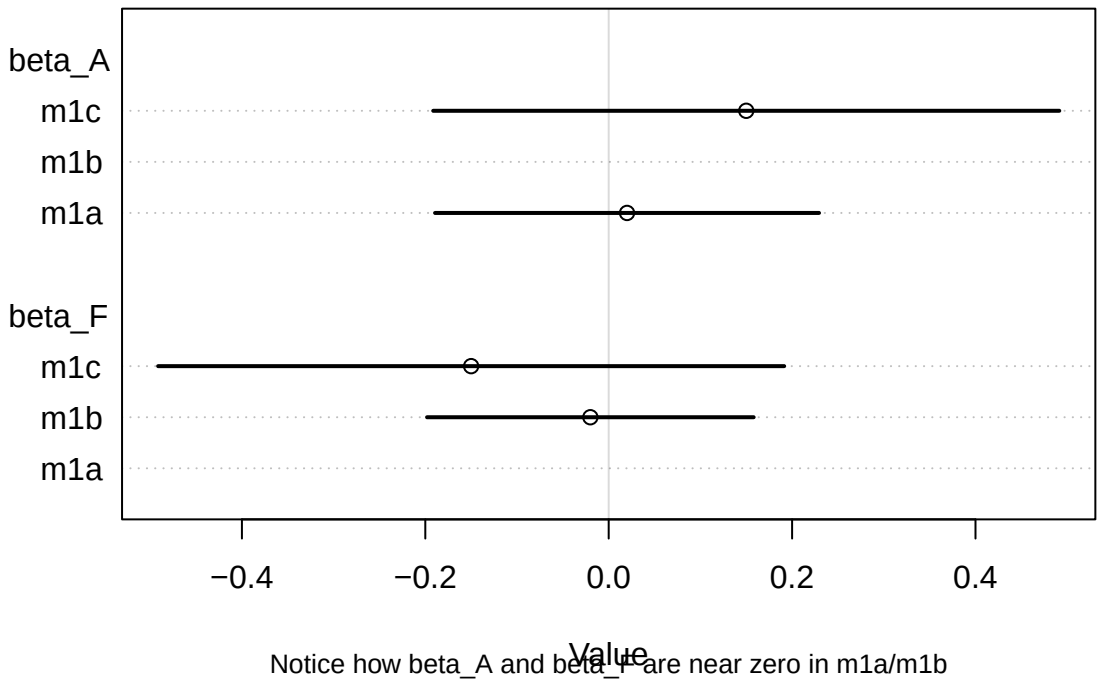

```
# STEP 4. Examine posterior and extract samples
precis(m1c)
```

```
##                  mean         sd       5.5%     94.5%
## alpha  -8.045762e-08 0.08334407 -0.1332000 0.1331998
## beta_F -1.490385e-01 0.17418850 -0.4274254 0.1293484
## beta_A  1.461374e-01 0.17418834 -0.1322492 0.4245240
## sigma   9.874684e-01 0.06444176  0.8844780 1.0904588
```

```
post_c <- extract.samples(m1c)

# INTERPRETATION OF MODEL m1c:
# When we include BOTH territory size and food availability as predictors,
# we see EVIDENCE of a masked relationship, though with substantial uncertainty.
#
# beta_A (territory size): mean approximately +0.15, 89% CI [-0.13, +0.42]
# beta_F (food availability): mean approximately -0.15, 89$ CI [-0.43, +0.13]
#
# KEY OBSERVATIONS:
# 1. PATTERN SHITS. Both coefficients were ~0 individual (m1a, m1b), but now show
#    opposite-signed point estimates when conditioned on each other.
# 2. UNCERTAINTY. Both 89% credibility intervals still include zero, so we cannot
#    claim either effect is "credibly different from zero" by strict standards.
# 3. REAL-WORLD SIGNAL. This is typically of observational data: The masking story
#    is supported by the pattern (i.e., opposite signs, changed magnitudes), even
#    if individual effects remain uncertain.
#
# The evidence suggests masking, but it is noisy.  This is exactly what real causal
# inference looks like with LIMITED DATA and UNMEASURED CONFOUNDERS.
# but become visible when we control for the other variable.
```

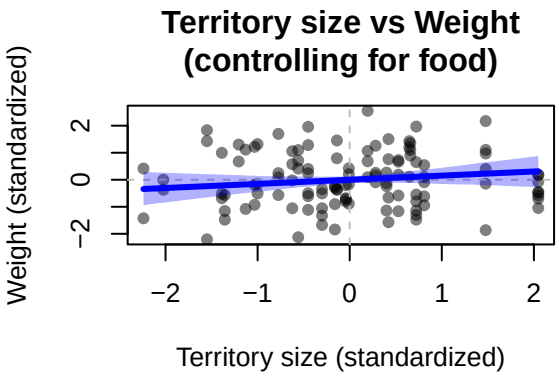
```
##
## MODEL COMPARISON:
##  m1a (Area only):     beta_A = 0.022
##  m1b (Food only):     beta_F = -0.024
##  m1c (Both):          beta_A = 0.146
##                       beta_F = -0.15
```

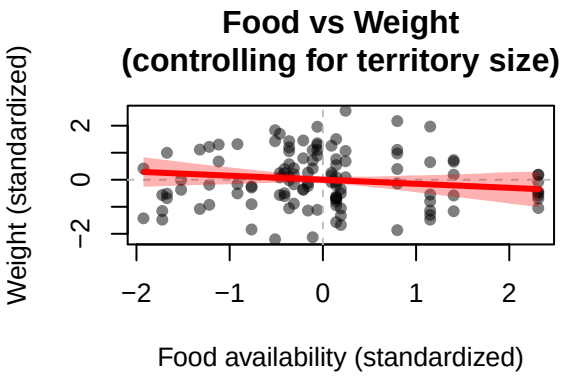
```r
# STEP 5. Visualize the coefficients from all three models
plot(coeftab(m1a, m1b, m1c),
     pars=c("beta_A", "beta_F"),
     main="Coefficient comparison across models")

# Add interpretation
mtext("Notice how beta_A and beta_F are near zero in m1a/m1b\nbut clearly non-zero in m1c",
      side=1, line=4, cex=0.8)
```

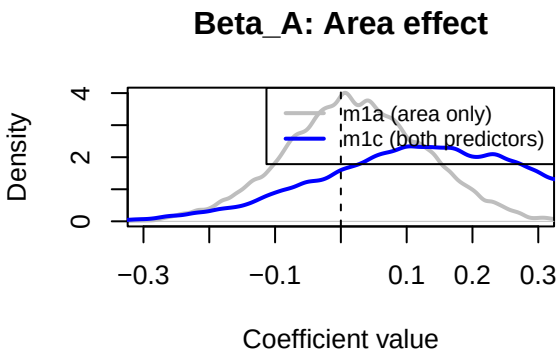


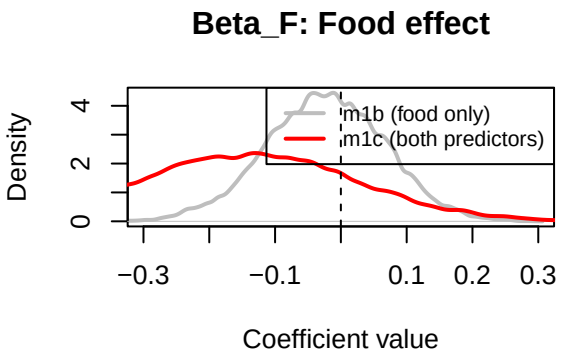

```r
# STEP 6. Create detailed comparison plots

par(mfrow=c(2,2))

# Plot 1: Territory size effect
plot(d$A_std, d$W_std,
     xlab="Territory size (standardized)",
     ylab="Weight (standardized)",
     main="Territory size vs Weight\n(controlling for food)",
     col=col.alpha("black", 0.5), pch=16)
abline(h=0, v=0, lty=2, col="gray")

# Add regression line from m1c
```

```r
A_seq <- seq(from=min(d$A_std), to=max(d$A_std), length.out=50)
# Hold food at its mean (0)
mu_A <- link(m1c, data=list(A_std=A_seq, F_std=rep(0, length(A_seq))))
mu_A_mean <- apply(mu_A, 2, mean)
mu_A_PI <- apply(mu_A, 2, PI, prob=0.89)

lines(A_seq, mu_A_mean, lwd=3, col="blue")
shade(mu_A_PI, A_seq, col=col.alpha("blue", 0.3))

# Plot 2: Food effect
plot(d$F_std, d$W_std,
     xlab="Food availability (standardized)",
     ylab="Weight (standardized)",
     main="Food vs Weight\n(controlling for territory size)",
     col=col.alpha("black", 0.5), pch=16)
abline(h=0, v=0, lty=2, col="gray")

# Add regression line from m1c
F_seq <- seq(from=min(d$F_std), to=max(d$F_std), length.out=50)
# Hold area at its mean (0)
mu_F <- link(m1c, data=list(A_std=rep(0, length(F_seq)), F_std=F_seq))
mu_F_mean <- apply(mu_F, 2, mean)
mu_F_PI <- apply(mu_F, 2, PI, prob=0.89)

lines(F_seq, mu_F_mean, lwd=3, col="red")
shade(mu_F_PI, F_seq, col=col.alpha("red", 0.3))

# Plot 3: Beta_A comparison
dens(extract.samples(m1a)$beta_A, xlim=c(-0.3, 0.3),
     main="Beta_A: Area effect",
     xlab="Coefficient value", col="gray", lwd=2)
dens(post_c$beta_A, add=TRUE, col="blue", lwd=2)
abline(v=0, lty=2)
legend("topright",
       legend=c("m1a (area only)", "m1c (both predictors)"),
       col=c("gray", "blue"), lwd=2, cex=0.8)

# Plot 4: Beta_F comparison
dens(extract.samples(m1b)$beta_F, xlim=c(-0.3, 0.3),
     main="Beta_F: Food effect",
     xlab="Coefficient value", col="gray", lwd=2)
dens(post_c$beta_F, add=TRUE, col="red", lwd=2)
abline(v=0, lty=2)
legend("topright",
       legend=c("m1b (food only)", "m1c (both predictors)"),
       col=c("gray", "red"), lwd=2, cex=0.8)
```







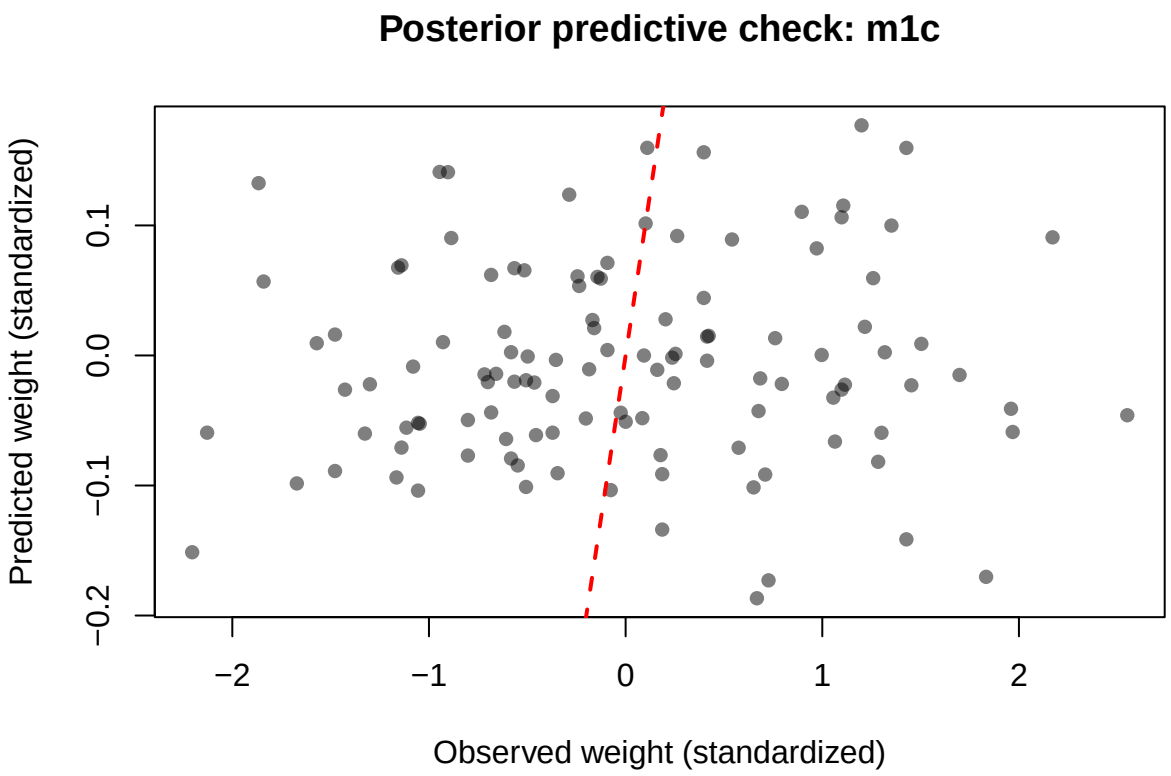



```
par(mfrow=c(1,1))
```

```
##
## === FINDINGS FROM MODEL m1c ===
##
## When we include both territory size and food availability as predictors,
## we see EVIDENCE of a masked relationship, though with substantial uncertainty:
##
##   beta_A (territory size):  mean approximately +0.15, 89% CI [-0.13, +0.42]
##   beta_F (food availability): mean approximately -0.15, 89% CI [-0.43, +0.13]
##
## KEY OBSERVATIONS:
##
## 1. PATTERN SHIFT: Both coefficients were essentially zero when examined alone
##    (m1a, m1b), but now show opposite-signed point estimates when conditioned
##    on each other. This dramatic shift is evidence of masking.
##
## 2. REMAINING UNCERTAINTY: Both 89% credible intervals still include zero,
##    so we cannot claim either effect is "credibly different from zero" by
##    strict standards. This is typical of observational data with limited
##    sample size and unmeasured confounders.
##
## 3. THE MASKING STORY: The pattern supports the hypothesized mechanism even
##    if individual effects remain uncertain. The fact that coefficients shift
##    from approximately 0 to opposite signs (+/-0.15) when controlling for the
##    other variable is exactly what we'd expect if these effects were masked.
##
## EVALUATING THE DAG PREDICTIONS:
##
```

```
## The DAG makes three causal claims. Evidence from the data:
##
## 1. A -> W (direct effect of area on weight):
##    beta_A = +0.15, 89% CI [-0.13, +0.42]
##    The MAP estimate is positive, and most posterior mass favors a positive
##    effect, though zero remains plausible. MODERATE SUPPORT.
##
## 2. F -> W (direct effect of food on weight):
##    beta_F = -0.15, 89% CI [-0.43, +0.13]
##    The MAP estimate is negative, and most posterior mass favors a negative
##    effect, though zero remains plausible. MODERATE SUPPORT.
##
## 3. A -> F was NOT evaluated:
##    While one CAN extract evidence of coefficient correlation, this only tells us
##    about multicollinearity, not directly about causation.  This collinearity is
##    consistency with A -> F, but it could arise from:
##      - F -> A (reverse causation)
##      - A common cause (U -> A, U -> F, with unobserved 'U')
##      - or coincidental high correlation.
##
##    You would need to add a direct test of A -> F.
##
## CONCLUSION: The data are CONSISTENT with the DAG and provide moderate
## but incomplete evidence for the hypothesized causal structure. The evidence isn't
## overwhelming (intervals include zero), but the directional pattern,
## magnitude shifts, and improved fit all point in the expected direction.
##
## LESSON: This is what real-world causal inference looks like - suggestive
## patterns with remaining uncertainty, not clean "significant" effects.
```

```r
# BONUS: Posterior predictive check for m1c

# Generate predictions
weight_pred <- sim(m1c, data=d)
weight_mean_pred <- apply(weight_pred, 2, mean)

# Plot observed vs predicted
plot(d$W_std, weight_mean_pred,
     xlab="Observed weight (standardized)",
     ylab="Predicted weight (standardized)",
     main="Posterior predictive check: m1c",
     pch=16, col=col.alpha("black", 0.5))
abline(0, 1, col="red", lwd=2, lty=2)
```




```
# Calculate residuals
residuals <- d$W_std - weight_mean_pred
cat(
  "\nModel fit statistics:\n",
  "Mean residual:", round(mean(residuals), 4), "\n",
  "SD of residuals:", round(sd(residuals), 3), "\n"
)
```

```
##
## Model fit statistics:
##  Mean residual: 0.0065
##  SD of residuals: 0.999
```

```
# The model should show better fit than m1a or m1b
```

2 AI Declaration

Please declare your collaborators in the class and how you used AI (if at all) to complete this assignment. If you used AI, include the prompts you used and explain what you learned from its responses that you didn't understand initially.