Statistical Reasoning 1

Peter Nelson and Abbie Ward

load libraries

```r
library(brms) # for statistics
library(tidyverse)
library(ggeffects) # for the prediction plot
library(lterdatasampler) # for built-in datasets
```

Load data

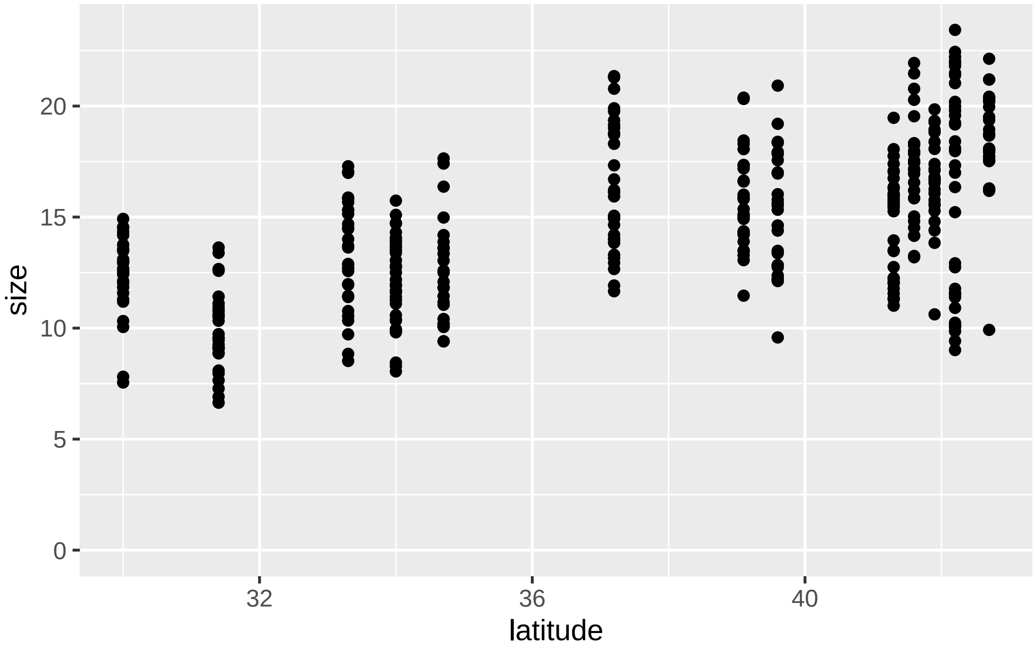
```r
head(pie_crab)
```

```
# A tibble: 6 x 9
  date       latitude site   size air_temp air_temp_sd water_temp water_temp_sd
  <date>        <dbl> <chr> <dbl>    <dbl>       <dbl>      <dbl>         <dbl>
1 2016-07-24       30 GTM    12.4     21.8        6.39       24.5          6.12
2 2016-07-24       30 GTM    14.2     21.8        6.39       24.5          6.12
3 2016-07-24       30 GTM    14.5     21.8        6.39       24.5          6.12
4 2016-07-24       30 GTM    12.9     21.8        6.39       24.5          6.12
5 2016-07-24       30 GTM    12.4     21.8        6.39       24.5          6.12
6 2016-07-24       30 GTM    13.0     21.8        6.39       24.5          6.12
# i 1 more variable: name <chr>
```

```r
view(pie_crab)
```

1.1 Plot data, pick the model

```r
pie_crab %>%
  ggplot(aes(x = latitude, y = size)) +
  geom_point() +
  # Make the y-axis include 0
  ylim(0, NA)
```

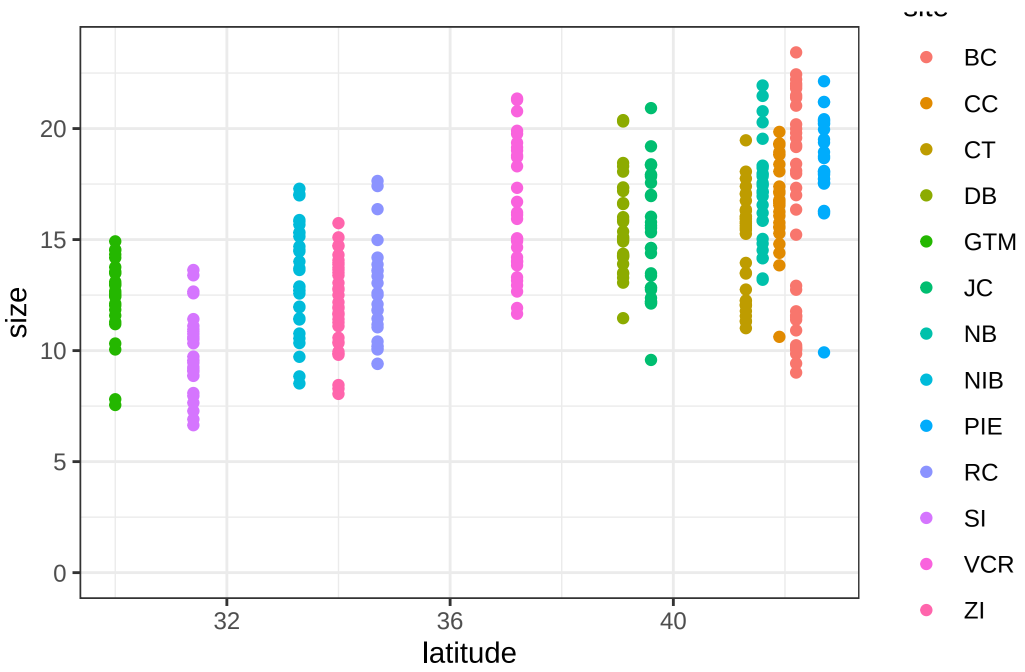


Q1.1 Interpret the graph

Seems to be a correlation between size and latitude, consistent with Bergman's Rule. There's a lot of variation at every site, but still there's a pretty solid positive slope. So there are certainly going to be multiple factors affecting regional crab size.

Q1.2 Beautify this graph

```r
p <-
  pie_crab %>%
  ggplot(aes(x = latitude, y = size)) +
  geom_point() +
  # Make the y-axis include 0
  ylim(0, NA)

p + aes(color = site) + theme_bw()
```



Now, let's model these data…

$$size = intercept + slope*latitude$

```
mod1 <- lm(size ~ latitude, data = pie_crab)
summary(mod1)
```

```


Call:
lm(formula = size ~ latitude, data = pie_crab)

Residuals:
    Min      1Q  Median      3Q     Max
-7.8376 -1.8797  0.1144  1.9484  6.9280

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.62442    1.27405  -2.845  0.00468 **
```

```
latitude     0.48512    0.03359  14.441   < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.832 on 390 degrees of freedom
Multiple R-squared:  0.3484,    Adjusted R-squared:  0.3467
F-statistic: 208.5 on 1 and 390 DF,  p-value: < 2.2e-16
```

1.2 Fit linear regression with `brms`

```r
m.crab.lat <-
  brm(data = pie_crab, # Give the model the pie_crab data
      # Choose a gaussian (normal) distribution
      family = gaussian,
      # Specify the model here.
      size ~ latitude,
      # Here's where you specify parameters for executing the Markov chains
      # We're using similar to the defaults, except we set cores to 4 so the analysis runs fa
      iter = 2000, warmup = 1000, chains = 4, cores = 4,
      # Setting the "seed" determines which random numbers will get sampled.
      # In this case, it makes the randomness of the Markov chain runs reproducible
      # (so that both of us get the exact same results when running the model)
      seed = 4,
      # Save the fitted model object as output - helpful for reloading in the output later
      file = "output/m.crab.lat")
```

Q1.3 What does the "iter" argument do?

Navigate to the `brm` help page to answer: What does the `iter =` argument do?

This term sets the total number of *iterations* per chain, including the warmup.

1.3 Assess model

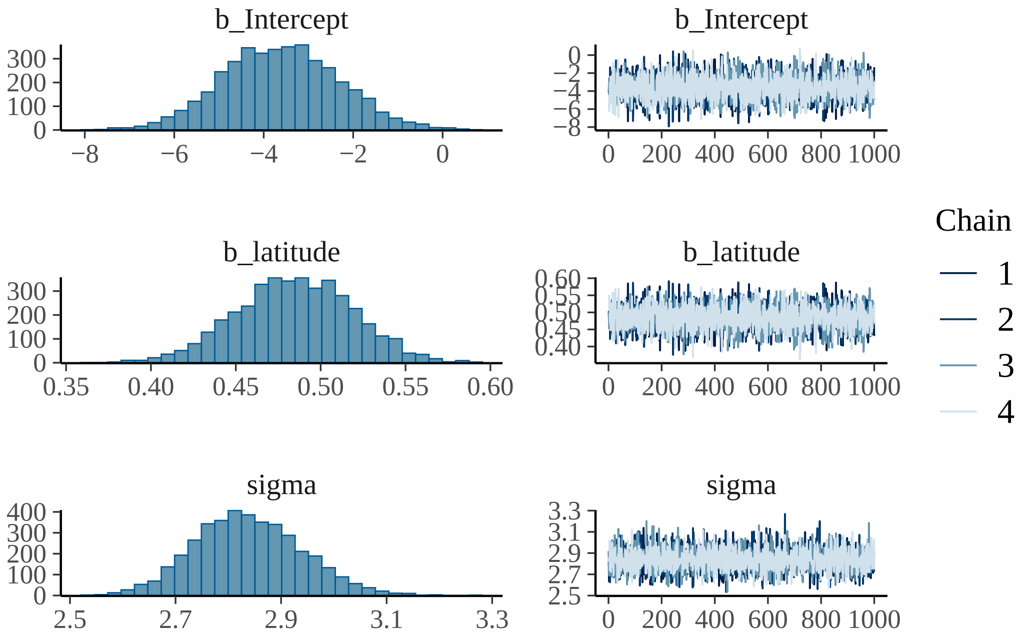
```r
summary(m.crab.lat)
```

```
 Family: gaussian
  Links: mu = identity
```

```
Formula: size ~ latitude
   Data: pie_crab (Number of observations: 392)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -3.61      1.30    -6.09    -1.01 1.00     4116     3192
latitude      0.48      0.03     0.42     0.55 1.00     4108     3140

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     2.84      0.10     2.65     3.04 1.00     3758     2852

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Plot the model output.

```r
plot(m.crab.lat) # show posteriors and chains
```



```r
summary(m.crab.lat)
```

```
 Family: gaussian
  Links: mu = identity
Formula: size ~ latitude
   Data: pie_crab (Number of observations: 392)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -3.61      1.30    -6.09    -1.01 1.00     4116     3192
latitude      0.48      0.03     0.42     0.55 1.00     4108     3140

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     2.84      0.10     2.65     3.04 1.00     3758     2852

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Output looks good. Estimate column indicates effect of latitude on crab size. Estimate looks solid and error is small. Based on the credible intervals, too, we have some real confidence that the slope is positive, but let's calculate the probability of slope = 0.

```r
as_draws_df(m.crab.lat)  %>%  # extract the posterior samples from the model estimate
  select(b_latitude)  %>%  # pull out the latitude samples from all 4 chains. we'll get a wa
  summarize(p_slope_lessthanorequalto_zero = sum(b_latitude <= 0)/length(b_latitude))
```

```
Warning: Dropping 'draws_df' class as required metadata was removed.
```

```
# A tibble: 1 x 1
  p_slope_lessthanorequalto_zero
                           <dbl>
1                              0
```

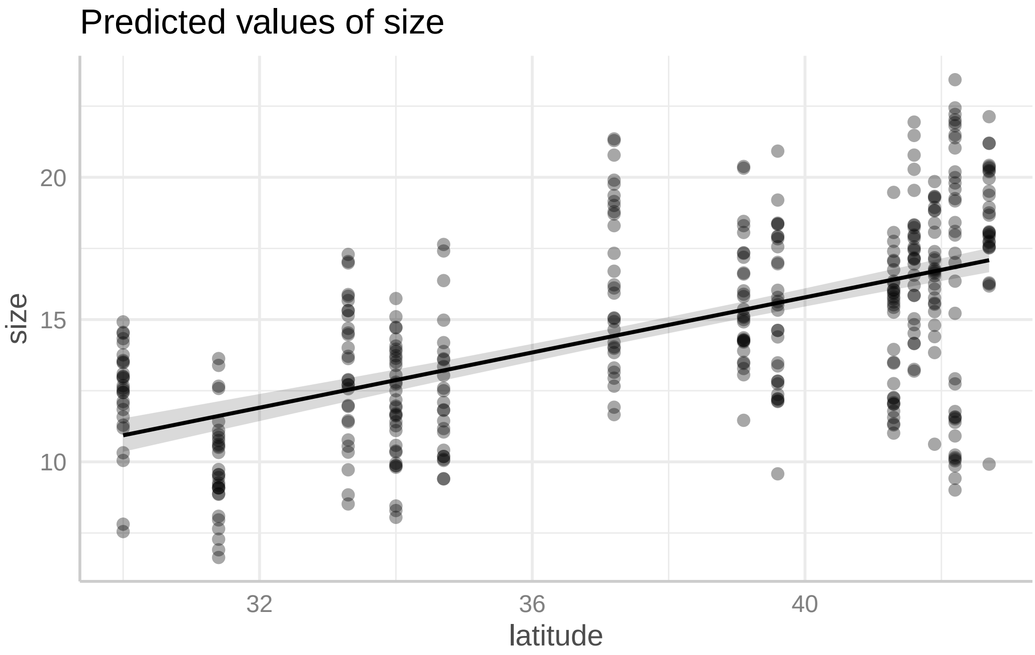
Great!

1.5 Plot model on the data

Start with the compatibility interval:

```
confm.crab.lat <- predict_response(m.crab.lat)
plot(confm.crab.lat, show_data = TRUE)
```

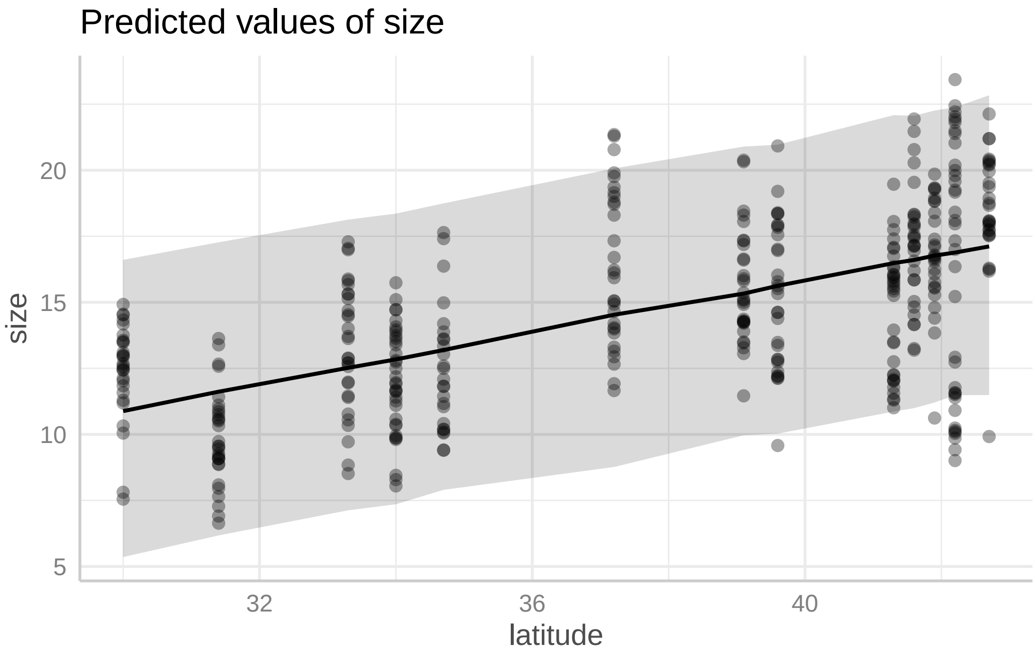
```
Data points may overlap. Use the `jitter` argument to add some amount of
  random variation to the location of data points and avoid overplotting.
```



Prediction interval:

```
confm.crab.lat <- predict_response(m.crab.lat, interval = 'prediction')
plot(confm.crab.lat, show_data = TRUE)
```

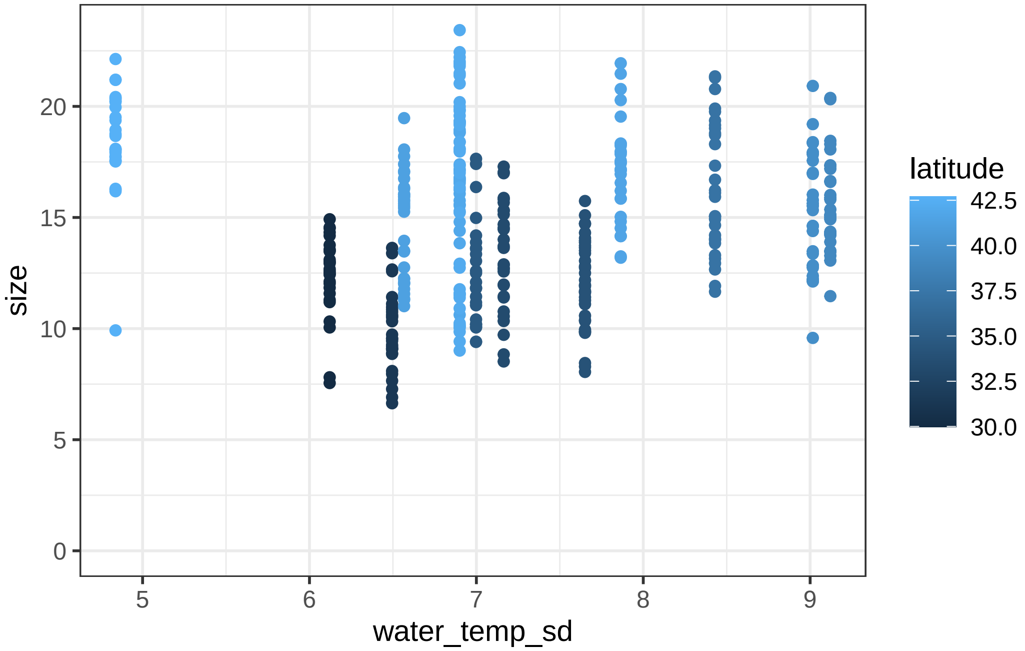
```
Data points may overlap. Use the `jitter` argument to add some amount of
  random variation to the location of data points and avoid overplotting.
```



1.6 Repeat with a new variable: water temp sd

Abbie thinks the northern portion of the fiddler crab range (more temperate) is going to experience greater variability in water temperatures. We definitely saw larger crabs at higher latitudes, so…we should see a positive relationship between water temp sd and body size. Peter thought the opposite…and now thinks he's wrong…

```
p2 <-
  pie_crab %>%
  ggplot(aes(x = water_temp_sd, y = size)) +
  geom_point() +
  # Make the y-axis include 0
  ylim(0, NA)

p2 + aes(color = latitude) + theme_bw()
```



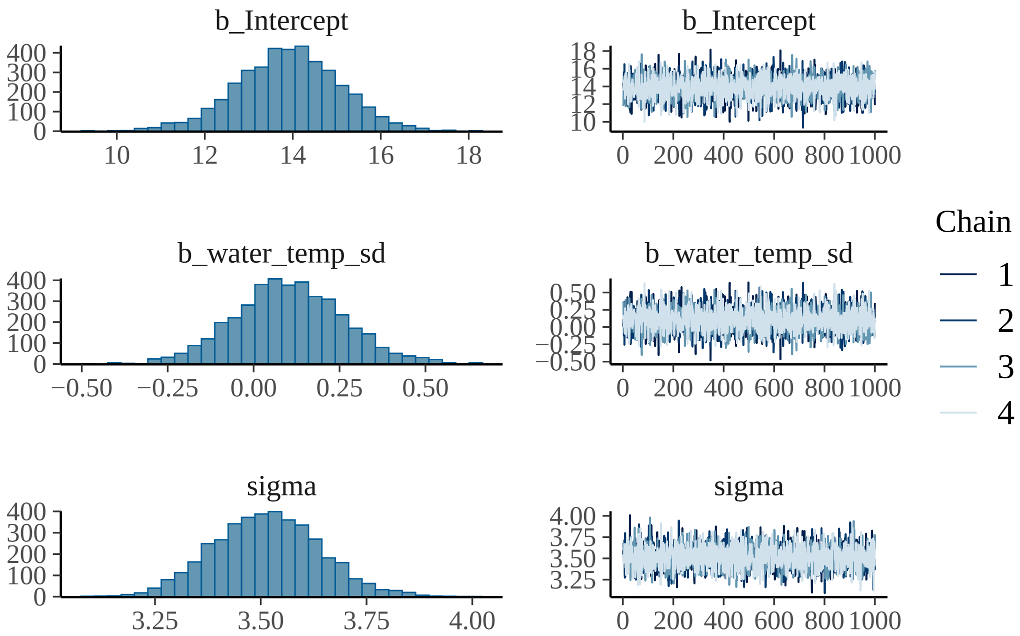
NOT what we expected! Changing careers now to furniture making...

Q1.7 Set up and run a model with this new relationship

```
m.crab.water.sd <-
  brm(data = pie_crab, # Give the model the pie_crab data
      # Choose a gaussian (normal) distribution
      family = gaussian,
      # Specify the model here.
      size ~ water_temp_sd,
      # Here's where you specify parameters for executing the Markov chains
      # We're using similar to the defaults, except we set cores to 4 so the analysis runs fa
      iter = 2000, warmup = 1000, chains = 4, cores = 4,
      # Setting the "seed" determines which random numbers will get sampled.
      # In this case, it makes the randomness of the Markov chain runs reproducible
      # (so that both of us get the exact same results when running the model)
      seed = 4,
      # Save the fitted model object as output - helpful for reloading in the output later
      file = "output/m.crab.water.sd")
```

Q1.8 Assess the model

```r
plot(m.crab.water.sd)
```



The model ran correctly. rhat looks good (=1). Plots look excellent. Unimodal distributions. Chains aren't overlapping.

```r
summary(m.crab.water.sd)
```

```
 Family: gaussian
  Links: mu = identity
Formula: size ~ water_temp_sd
   Data: pie_crab (Number of observations: 392)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept        13.90      1.17    11.47    16.15 1.00     3770     2942
water_temp_sd     0.10      0.16    -0.21     0.44 1.00     3797     2725
```

```
Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     3.51      0.13     3.27     3.77 1.00     3544     3064

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Q 1.8 Interpret the model

The effect of the predictor shows that 0.1 increase in body size with the standard deviation of the water temperature. But the credible intervals include 0; the effect is not reasonably different from zero. Not surprising considering our plot of the relationship. Also the size of the estimate relative to the error…

Back to Pikas

Look at the data again.

```r
head(nwt_pikas)
```

```
# A tibble: 6 x 8
  date       site      station utm_easting utm_northing sex   concentration_pg_g
  <date>     <fct>     <fct>         <dbl>        <dbl> <fct>              <dbl>
1 2018-06-08 Cable Ga~ Cable ~      451373      4432963 male              11563.
2 2018-06-08 Cable Ga~ Cable ~      451411      4432985 male              10629.
3 2018-06-08 Cable Ga~ Cable ~      451462      4432991 male              10924.
4 2018-06-13 West Kno~ West K~      449317      4434093 male              10414.
5 2018-06-13 West Kno~ West K~      449342      4434141 male              13531.
6 2018-06-13 West Kno~ West K~      449323      4434273 <NA>               7799.
# i 1 more variable: elev_m <dbl>
```

```r
nwt_pikas_doy <- nwt_pikas %>%
  # Add a new column called day_of_year
  # yday extracts the day of year from the date column
  mutate(day_of_year = yday(date)) %>%
  # relocate the day_of_year column after the date column
  relocate(day_of_year, .after = date)

head(nwt_pikas_doy)
```

```
# A tibble: 6 x 9
  date       day_of_year site       station      utm_easting utm_northing sex
  <date>           <dbl> <fct>      <fct>              <dbl>        <dbl> <fct>
1 2018-06-08         159 Cable Gate Cable Gate 1      451373      4432963 male
2 2018-06-08         159 Cable Gate Cable Gate 2      451411      4432985 male
3 2018-06-08         159 Cable Gate Cable Gate 3      451462      4432991 male
4 2018-06-13         164 West Knoll West Knoll 3      449317      4434093 male
5 2018-06-13         164 West Knoll West Knoll 4      449342      4434141 male
6 2018-06-13         164 West Knoll West Knoll 5      449323      4434273 <NA>
# i 2 more variables: concentration_pg_g <dbl>, elev_m <dbl>
```

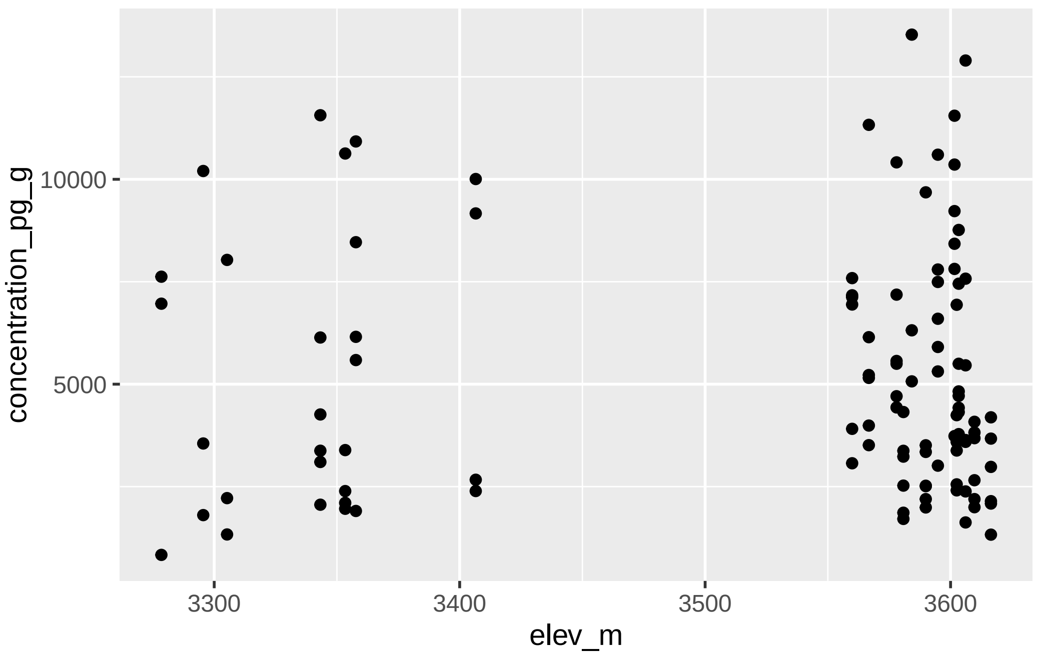
Q 2.1 Make a question

Let's look at the relationship between stress and elevation.

Q 2.2 Make a hypothesis

We expect that stress levels will be inversely related to elevation. At lower elevations, perhaps pikas are exposed to greater predation rates.

```r
ggplot(nwt_pikas_doy,
       aes(x = elev_m, y = concentration_pg_g)) +
  geom_point()
```



I have no idea what's going on, so obviously we have to do statistics.

```
m.pika.elev <-
  brm(data = nwt_pikas_doy, # Give the model the pika data
      # Choose a gaussian (normal) distribution
      family = gaussian,
      # Specify the model here.
      concentration_pg_g ~ elev_m,
      # Here's where you specify parameters for executing the Markov chains
      # We're using similar to the defaults, except we set cores to 4 so the analysis runs fa
      iter = 2000, warmup = 1000, chains = 4, cores = 4,
      # Setting the "seed" determines which random numbers will get sampled.
      # In this case, it makes the randomness of the Markov chain runs reproducible
      # (so that both of us get the exact same results when running the model)
      seed = 4,
      # Save the fitted model object as output - helpful for reloading in the output later
      file = "output/m.pika.elev")
```

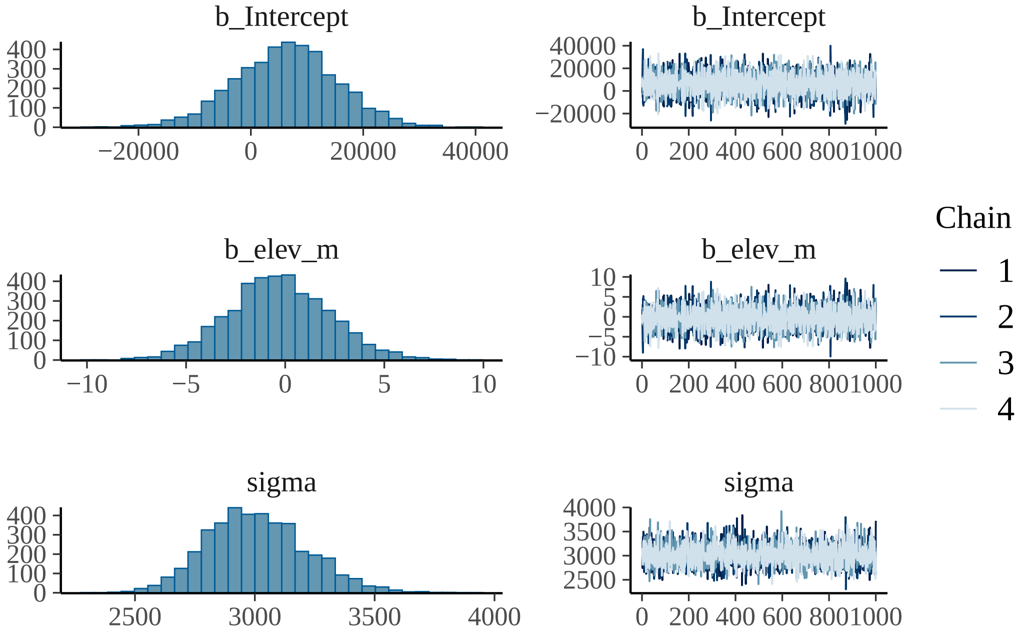
Q 2.5 Assess the model

```
summary(m.pika.elev)
```

```
 Family: gaussian
  Links: mu = identity
Formula: concentration_pg_g ~ elev_m
   Data: nwt_pikas_doy (Number of observations: 109)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
          Estimate Est.Error  l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept  6316.20   9153.40 -12514.07 23986.05 1.00     3948     2928
elev_m       -0.33      2.59     -5.36     5.04 1.00     3953     2918

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma  3009.80    209.45  2630.66  3443.94 1.00     3017     2749

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

rhat looks good (=1), but the CIs do not look good! These include 0 and the error is an order of magnitude greater than the estimate.

```r
plot(m.pika.elev)
```



Nothing wrong with the model…the data simply aren't cooperating (;-).

Q 2.6 Interpret the model

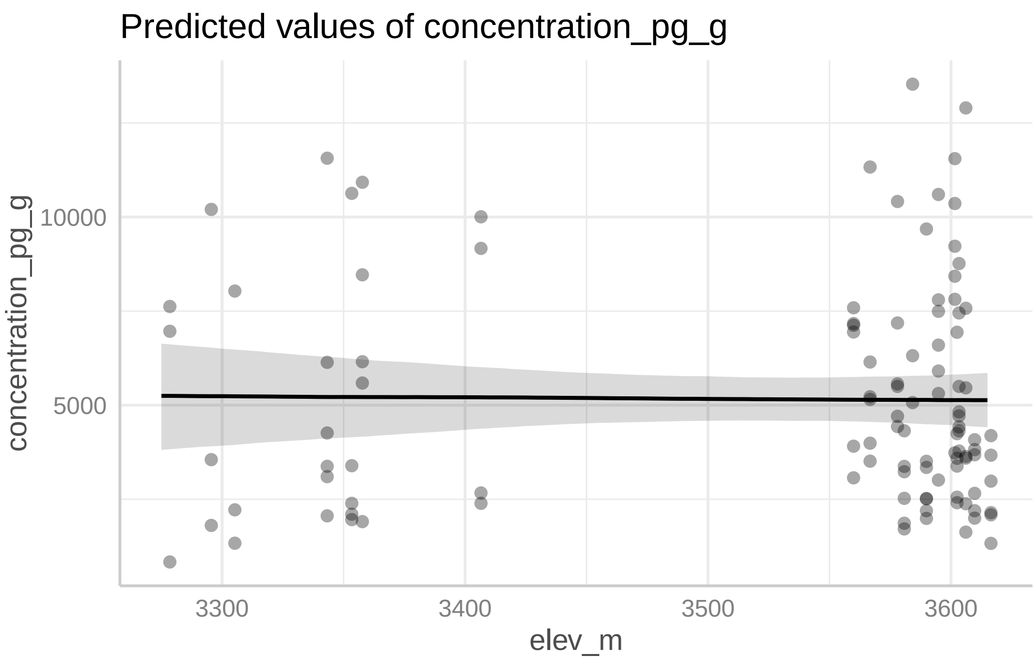
Elevation is not a good indicator of stress as shown by our estimate (model results). Our CI values show that the effect is not reasonably different from 0.

Q 2.7 Plot the model on the data

Compatibility interval:

```
confm.pika.elev <- predict_response(m.pika.elev)
plot(confm.pika.elev, show_data = TRUE)
```

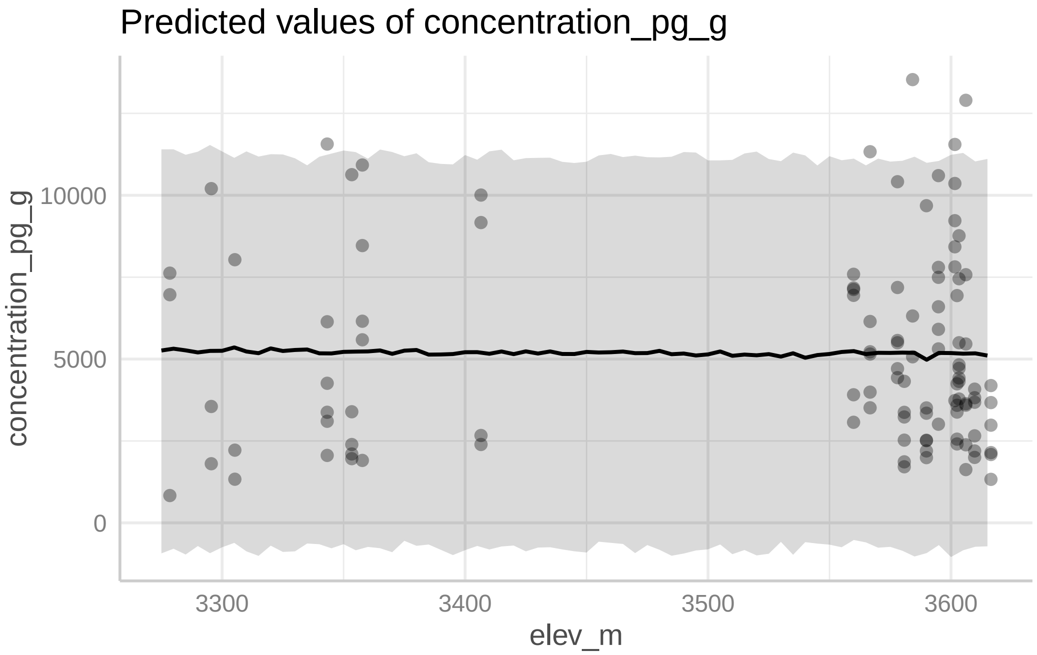
```
Data points may overlap. Use the `jitter` argument to add some amount of
  random variation to the location of data points and avoid overplotting.
```



Plot prediction interval:

```
confm.pika.elev <- predict_response(m.pika.elev, interval = "prediction")
plot(confm.pika.elev, show_data = TRUE)
```

```
Data points may overlap. Use the `jitter` argument to add some amount of
  random variation to the location of data points and avoid overplotting.
```



Q 2.8 Results

We found no evidence of a relationship between elevation and stress levels in pikas. The models seemed to run well. The CIs included 0, strongly suggesting that there was no difference from 0. And our estimates were consistently low and the associated errors were high.