SCI 2025: Homework 2

Setup

In this homework, we will work with data from:

Nettle, D. (1998). Explaining global patterns of language diversity. *Journal of Anthropological Archaeology*, 17:354–74.

First, load the data into your R session.

```
library(rethinking)
data(nettle)
head(nettle)
```

```
     country num.lang    area  k.pop num.stations mean.growing.season
1    Algeria       18 2381741  25660          102                6.60
2     Angola       42 1246700  10303           50                6.22
3  Australia      234 7713364  17336          134                6.00
4 Bangladesh       37  143998 118745           20                7.40
5      Benin       52  112622   4889            7                7.14
6    Bolivia       38 1098581   7612           48                6.92
  sd.growing.season
1              2.29
2              1.87
3              4.17
```

4	0.73
5	0.99
6	2.50

The meaning of each column in the dataset is given below:

(1) country: Name of the country
(2) num.lang: Number of recognized languages spoken
(3) area: Area in square kilometers
(4) k.pop: Population, in thousands
(5) num.stations: Number of weather stations that provided data for the next two columns
(6) mean.growing.season: Average length of growing season,in months
(7) sd.growing.season: Standard deviation of length of growing season,in months

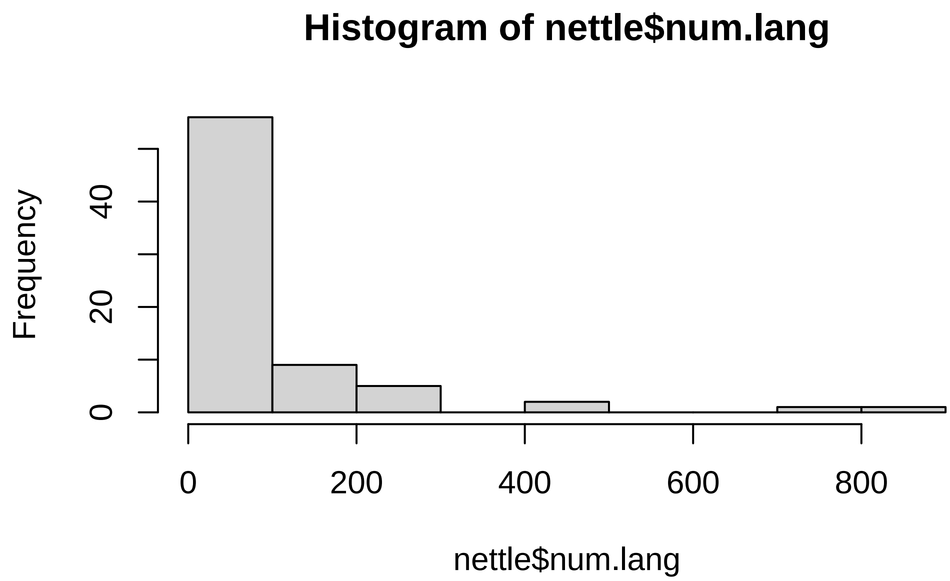
You should use quadratic approximation via `rethinking::quap()` for all model fitting.

Question 1

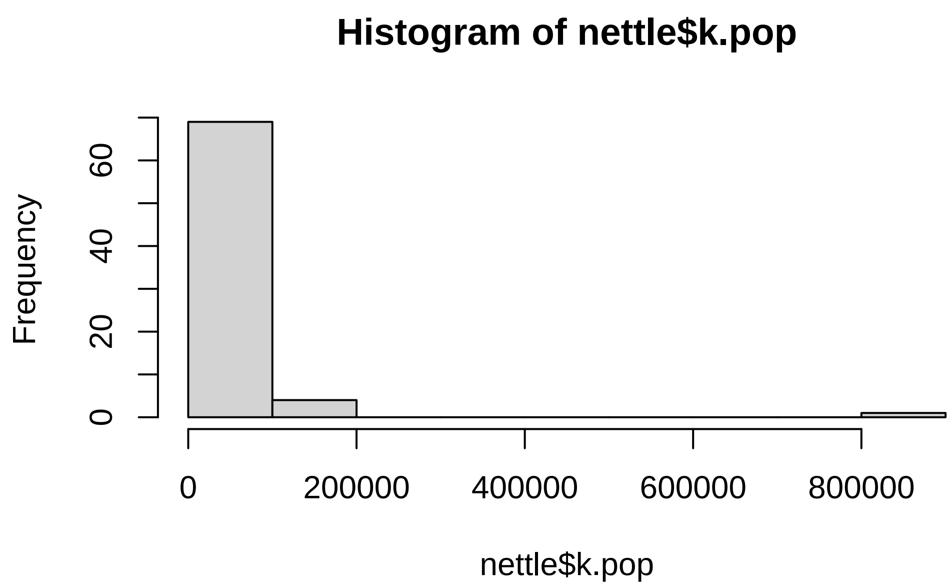
Write down a mathematical model that describes a linear regression of the number of languages spoken (`num.lang`) as a function of the population of the country (`k.pop`). Use similar notation to the textbook chapter. Be sure to include prior definitions for all parameters. You may apply any transformations to the data that you think are appropriate.

First, let's get a sense of the scale of the data.

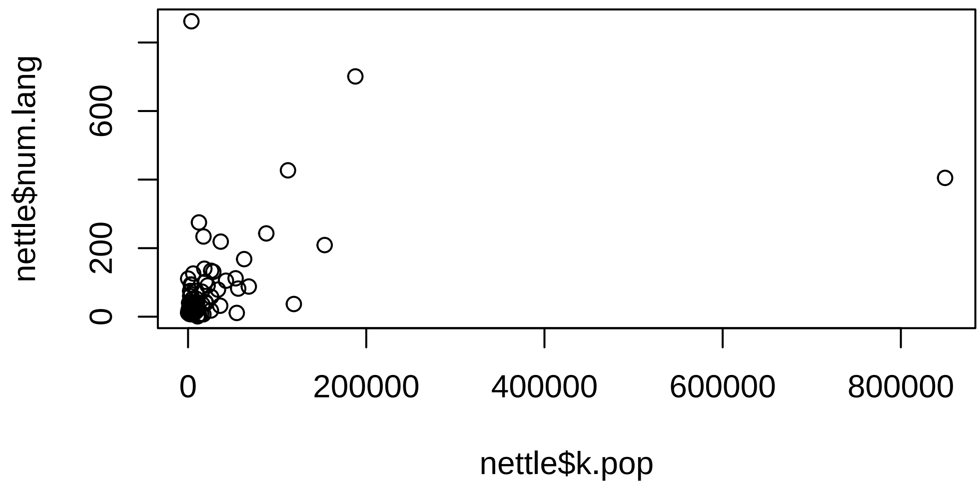
```
options(scipen=999) # disable scientific notation
hist(nettle$num.lang)
```



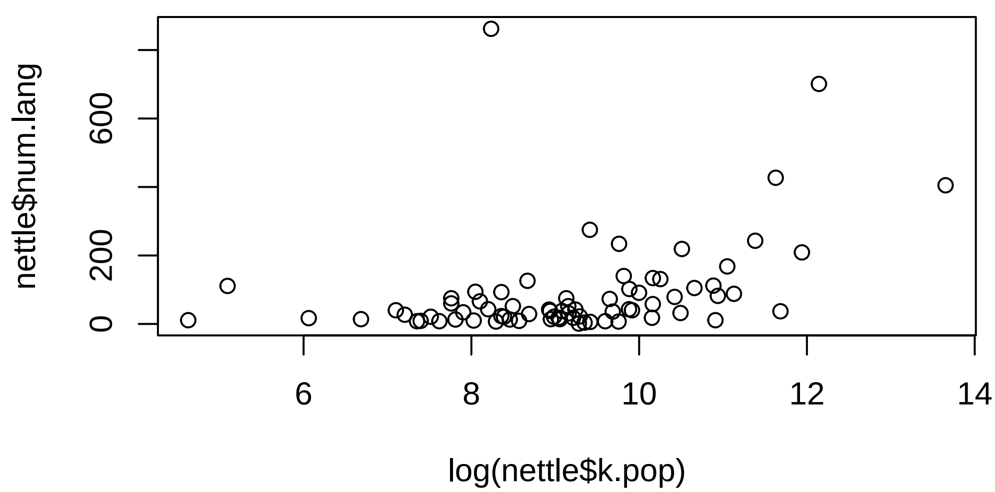

```
hist(nettle$k.pop)
```




```
plot(nettle$num.lang ~ nettle$k.pop)
```

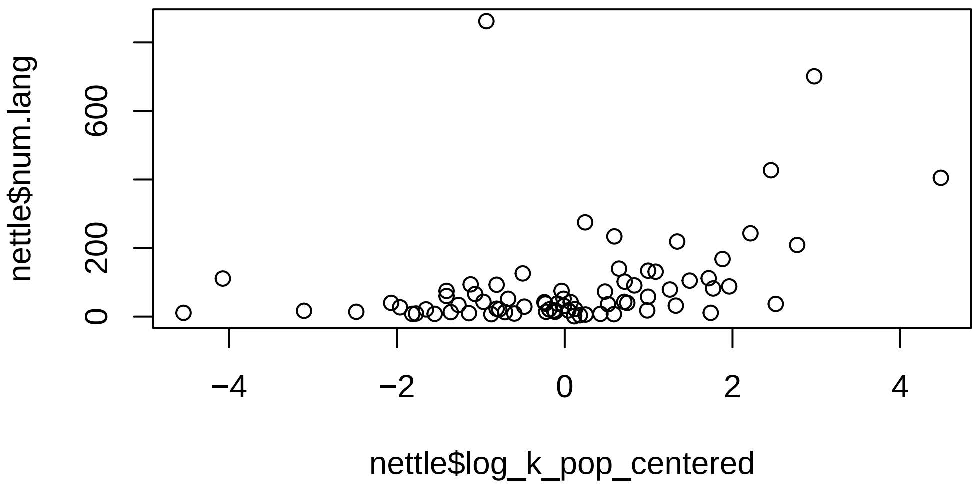



```
# log transform makes the relationship more linear
plot(nettle$num.lang ~ log(nettle$k.pop))
```



I don't think I can set a sensible intercept when log(k.pop) is 0. Instead, I will center the log(k.pop) variable using the mean.

```r
nettle$log_k_pop <- log(nettle$k.pop)
nettle$log_k_pop_centered <- nettle$log_k_pop - mean(nettle$log_k_pop)

plot(nettle$num.lang ~ nettle$log_k_pop_centered)
```



For each country i:

$$num_l ang_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta * (log(k.pop_i) - log(\bar{k}.pop))$$

$$\alpha \sim \text{Normal}((num\bar{.}lang), 100)$$

$$\beta \sim \text{Normal}(50, 50)$$

$$\sigma \sim \text{Uniform}(0, 200)$$

Where $log(\bar{k}.pop)$ denotes the sample mean of the log-transformed population size and $num\bar{.}lang$ denotes the sample mean of the number of languages spoken.

Question 2

Implement the model you wrote down in Question 1 using `rethinking::quap()`. Print the model summary.

```
round(mean(nettle$num.lang), 2)
```

```
[1] 89.73
```

```
m1 <- quap(
  alist(
    num.lang ~ dnorm(mu, sigma),
    mu <- a + b * (log_k_pop_centered),
    a ~ dnorm(89.73, 100),
    b ~ dnorm(50, 50),
    sigma ~ dunif(0, 200)
  ),
  data = nettle
)


precis(m1)
```

```
           mean        sd      5.5%    94.5%
a      89.99427 15.252427  65.61795 114.3706
b      35.56609  9.798508  19.90618  51.2260
sigma 132.75942 10.924385 115.30015 150.2187
```

Question 3

Perform a posterior predictive check on the model you fit in Question 2. You should plot the posterior function relating the number of languages spoken to the population of the country. Represent uncertainty either by drawing lines from the posterior or by plotting a credible/highest posterior density interval. Be sure to also plot the raw data.

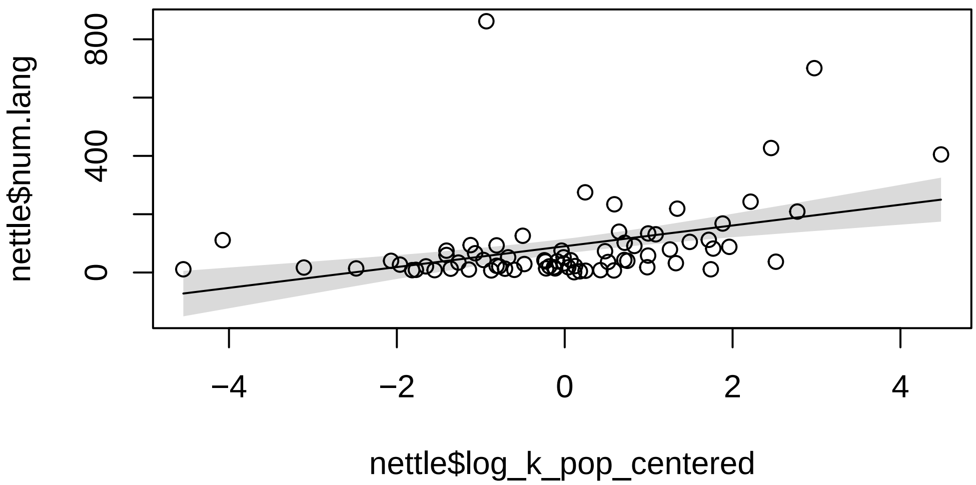
```r
post <- extract.samples(m1)

log_pop_seq <- seq(
    from = min(nettle$log_k_pop_centered),
    to = max(nettle$log_k_pop_centered),
    length.out = 30)

mu_num_lang <- sapply(log_pop_seq, function(x) post$a + post$b * x)

# Or, more conveniently:
mu_num_lang2 <- link(m1, data = data.frame(log_k_pop_centered = log_pop_seq))

str(mu_num_lang)
```

```
 num [1:10000, 1:30] -122 -56.5 -59.9 -74.1 -73.5 ...
```

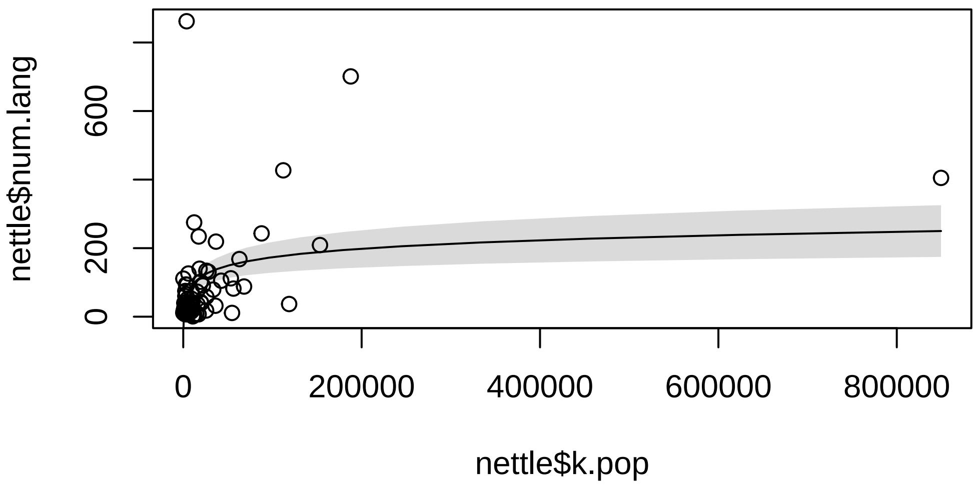
```r
str(mu_num_lang2)
```

```
 num [1:1000, 1:30] -70.73 -101.74 -63.79 -117.86 5.76 ...
```

```r
mu_mean <- apply(mu_num_lang, 2, mean)
mu_CI <- apply(mu_num_lang, 2, PI, prob = 0.90)

plot(nettle$num.lang ~ nettle$log_k_pop_centered,
 ylim = range(c(nettle$num.lang, mu_CI)))
lines(log_pop_seq, mu_mean)
shade(mu_CI, log_pop_seq)
```



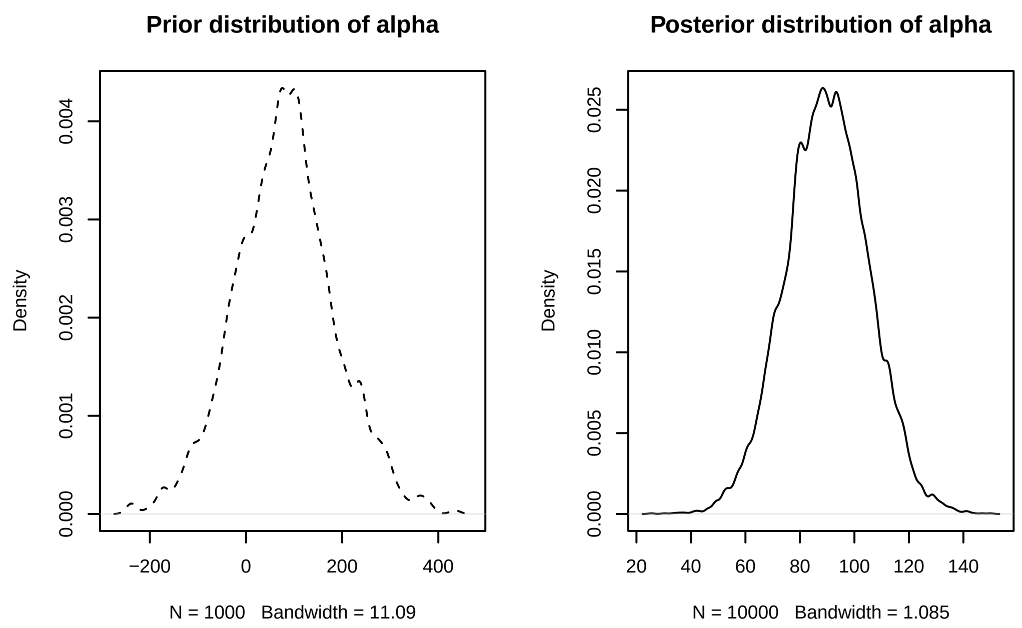

```
# what if we plot the relationship on the original scale?
pop_seq <- exp(log_pop_seq + mean(nettle$log_k_pop))


plot(nettle$num.lang ~ nettle$k.pop)
lines(pop_seq, mu_mean)
shade(mu_CI, pop_seq)
```



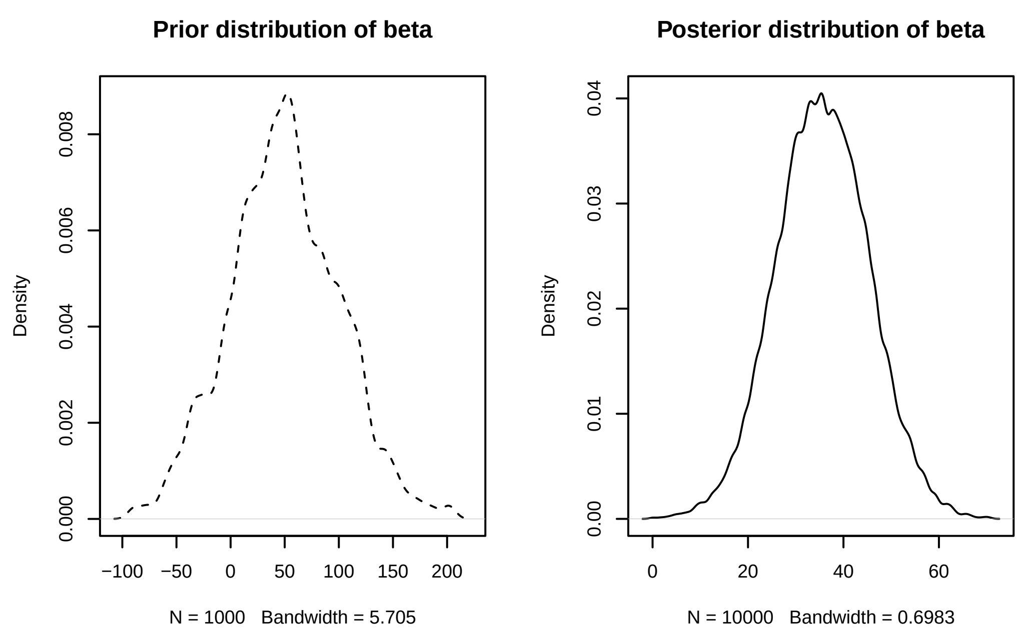
Question 4

Visually compare the prior and posterior distributions of the *parameters* from the model you fit in Question 2.

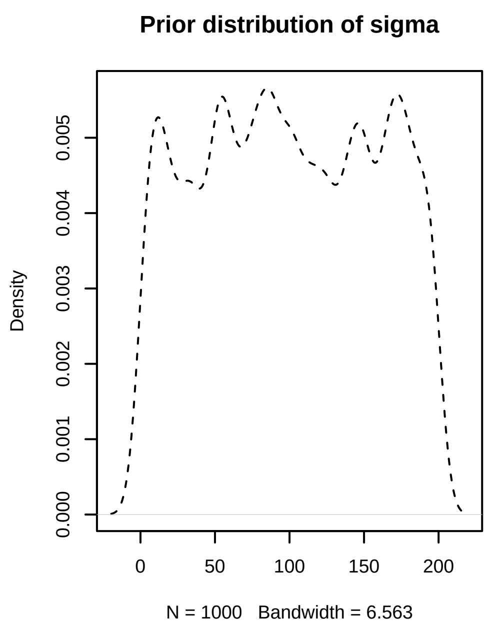
```
prior <- extract.prior(m1)


par(mfrow = c(1, 2), cex = 0.6)
dens(prior$a, lty="dashed", main = "Prior distribution of alpha")
dens(post$a, main = "Posterior distribution of alpha")
```

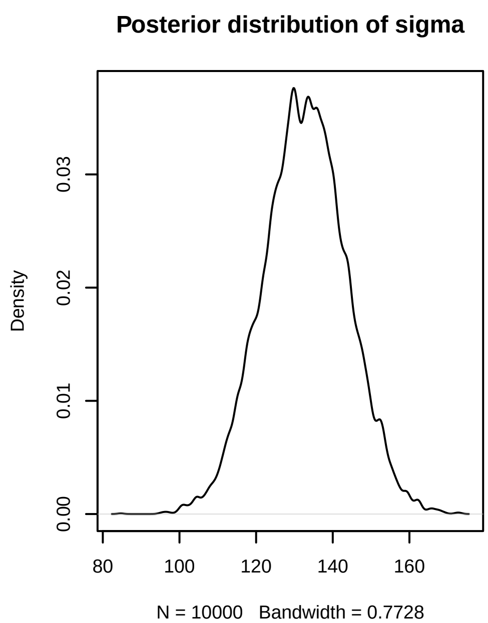



```
dens(prior$b, lty="dashed", main = "Prior distribution of beta")
dens(post$b, main = "Posterior distribution of beta")
```



```
dens(prior$sigma, lty="dashed", main = "Prior distribution of sigma")
dens(post$sigma, main = "Posterior distribution of sigma")
```





Question 5

Using insights from Questions 3-4, try to improve upon the model you fit in Question 2. Justify your changes, fit the new model, and perform a new posterior predictive check.

A few changes I'd like to make:

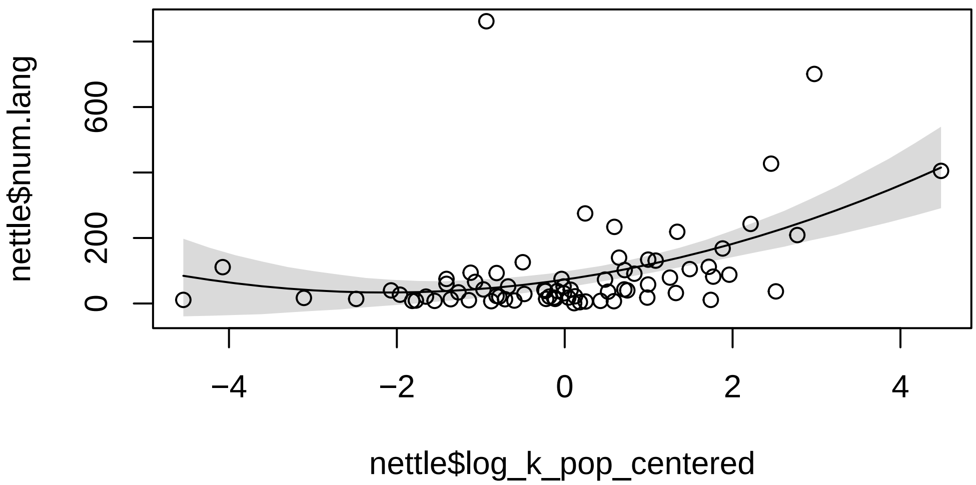
1. I'd like to add a quadratic term, allowing the relationship
2. I'd like to make the priors on the intercept and sigma more realistic.

```
m2 <- quap(
  alist(
    num.lang ~ dnorm(mu, sigma),
    mu <- a + b * (log_k_pop_centered) + b2 * (log_k_pop_centered)^2,
    a ~ dnorm(89.73, 30),
```

```r
    b ~ dnorm(50, 50),
    b2 ~ dnorm(0, 10),
    sigma ~ dunif(50, 200)
  ),
  data = nettle
)

precis(m2)
```

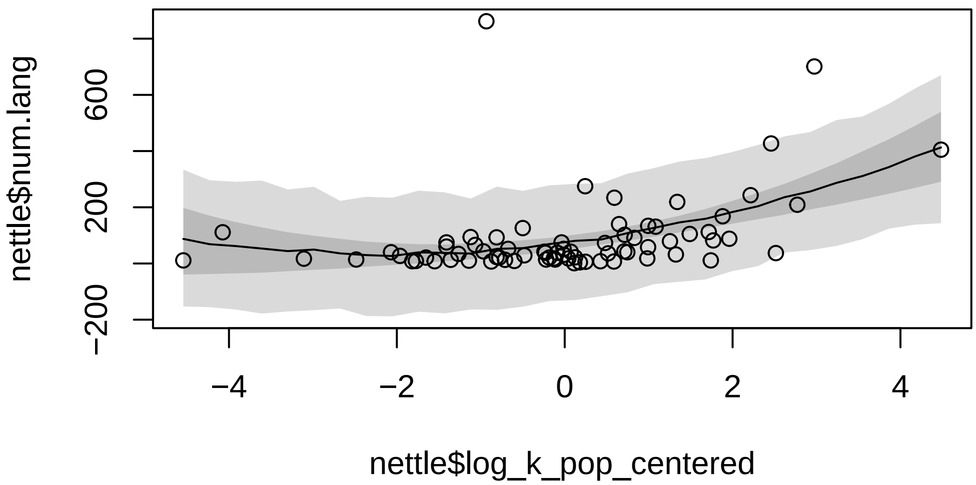
```
           mean        sd       5.5%     94.5%
a      73.161767 14.666664  49.721606  96.60193
b      37.200134  9.347660  22.260767  52.13950
b2      8.717198  3.317204   3.415666  14.01873
sigma 126.112615 10.414443 109.468324 142.75691
```

```r
log_pop_seq <- seq(
    from = min(nettle$log_k_pop_centered),
    to = max(nettle$log_k_pop_centered),
    length.out = 30)

mu_num_lang <- link(m2, data = data.frame(log_k_pop_centered = log_pop_seq))

mu_mean <- apply(mu_num_lang, 2, mean)
mu_CI <- apply(mu_num_lang, 2, PI, prob = 0.90)

plot(nettle$num.lang ~ nettle$log_k_pop_centered,
 ylim = range(c(nettle$num.lang, mu_CI)))
lines(log_pop_seq, mu_mean)
shade(mu_CI, log_pop_seq)
```



Or, using the full predictive distribution:

```
sim_num_lang <- sim(m2, data = data.frame(log_k_pop_centered = log_pop_seq))


sim_mean <- apply(sim_num_lang, 2, mean)
sim_CI <- apply(sim_num_lang, 2, PI, prob = 0.90)


plot(nettle$num.lang ~ nettle$log_k_pop_centered,
 ylim = range(c(nettle$num.lang, sim_CI)))
lines(log_pop_seq, sim_mean)
shade(sim_CI, log_pop_seq)
shade(mu_CI, log_pop_seq) # superimpose the mean CI
```


nettle$num.lang
−200
200
600
−4
−2
0
2
4
nettle$log_k_pop_centered