Lab 13

Kevin Buck

4/19/2023

Today's objectives

- Evaluate a multiple regression model with and without t-tests, F-test, or p-values
- Evaluate using standard errors and effect sizes
- Evaluate predictive power using cross-validation
- Stabilize coefficient estimates using prior information
- Explore example code for building Bayesian models and leave-one-out cross-validation

Introduction

A common way of evaluating the information in a statistical model (and the approach highlighted in texts like W&S) is via the information shown in summary() function in R. For instance, if I write: model <- lm(y ~ x1+x2), summary(model) will provide coefficients and standard errors, as well as p-values and t-tests with significance levels for x1 and x2. Often those significance levels are used as a lexicographic decision rule for whether those coefficients should be included in the model.

For familiar reasons, Gelman, Hill and Vehtari (2021, RoS) write, "One thing we do *not* recommend is traditional null hypothesis significance tests." (RoS 10.8). However, the RoS authors recognize that the issue of uncertainty in parameter estimation is important and that it makes sense to ask whether a coefficient, "could just as well be zero, as far as the data are concerned." So, how do they recommend analysts evaluate coefficients?

In tutorial today you will use four different approaches to evaluate factors that might be associated with infection from the intestinal parasite whipworms in fecal samples from wild baboons collected in 2015 by the Amboseli Baboon Research Project (AMBR). In each case, our goals are to demonstrate the code needed to evaluate linear models, then to assess what each of these approaches tell us about factors that might explain (the log of) whipworm egg counts in fecal samples from baboons:
(A) Assessing coefficient estimates and uncertainties using the summary of stan_glm().
(B) Comparing this interpretation to the summary of the familiar lm() function.
(C) Evaluating whether including a noisy coefficient improves the ability of the model to predict whipworm eggs because of overfitting.
(D) Decreasing the noise of this coefficient using information from a Bayesian prior.

In class

Whipworms are parasitic roundworms that cause trichuriasis in people. They also infect other animals, including baboons. Across the year 2015, AMBR collected 91 fecal samples from male wild baboons in Kenya and estimated the parasitic whipworm load in these animals at the time of sample collection as the natural log of standardized egg counts, **ln_eggs** in our dataset. Both social and environmental variables could influence whipworm load, and we will model one of each as factors potentially associated with **ln_eggs**. **rain** is the total precipitation over the 3 months preceding the sample collection (cm). Baboons eat less (and lower quality) food during dry periods, which might increase their susceptibility to parasitism. **n_groups** is the number of "social groups" that each male lived in during the year preceding the sample collection. Exposure to a more diverse group of baboons might increase their exposure to parasites.

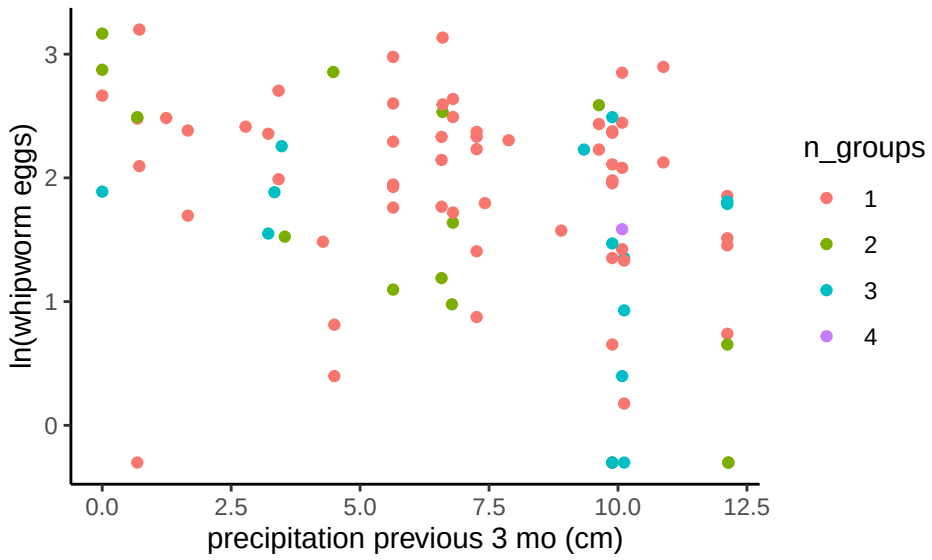
These hypotheses about how rain and n_groups might affect eggs are speculative. We'll use the dataset to assess them.

Let's start by reading in the data and looking at its structure.

```
# Load packages: ggplot2 for graphing.
# arm is an RoS package & rstanarm builds Bayesian linear models.
# loo is for cross validation.

library(tidyverse)
library("rstanarm")
library("arm")
library("loo")

# Read in data and store as object
parasite <- read_csv("parasite_2015.csv") #, header=TRUE)

# Look at the data
glimpse(parasite)
```

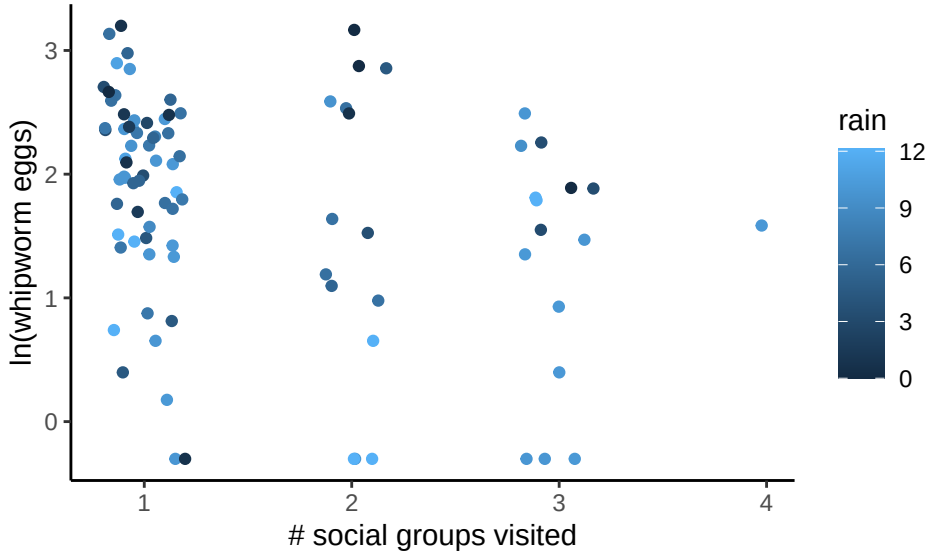
```
## Rows: 91
## Columns: 3
## $ n_groups <dbl> 3, 1, 3, 3, 1, 1, 3, 1, 1, 4, 1, 3, 1, 2, 1, 3, 1, 1, 1, 3, 1~
## $ rain     <dbl> 10.12, 10.12, 10.12, 10.12, 10.12, 10.08, 10.08, 10.08, 10.08~
## $ ln_eggs  <dbl> 1.352, 1.332, -0.301, 0.929, 0.176, 2.850, 0.398, 1.423, 2.08~
```

```
# Plot the predictors (2 graphs)...
parasite %>%
  # Set aesthetics including setting color to be designated by n_groups
  ggplot(aes(x = rain, y = ln_eggs, color = as.factor(n_groups))) +
  # Add scatterplot
  geom_point() +
  # Set theme
  theme_classic() +
  xlab("precipitation previous 3 mo (cm)") + ylab("ln(whipworm eggs)") + scale_color_discrete(name = "n_
```

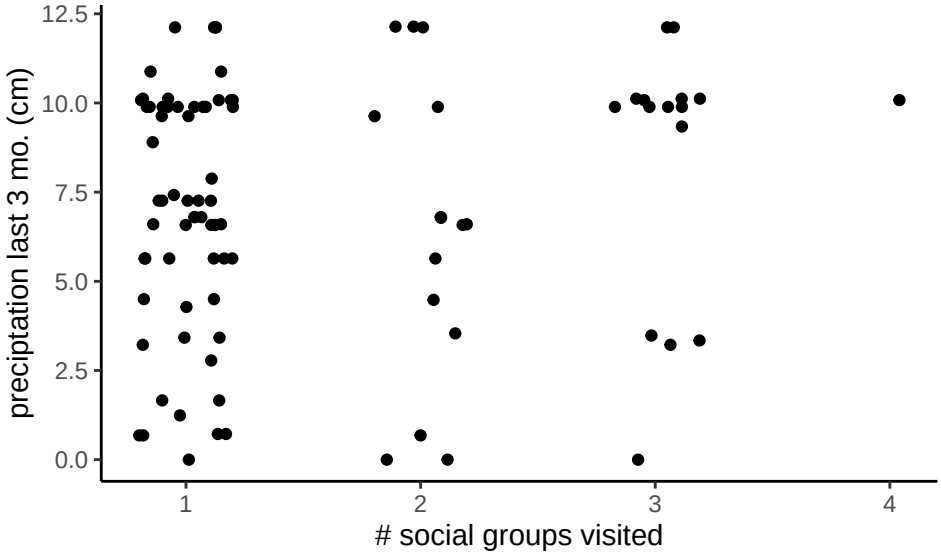



```
parasite %>%
  # Set aesthetics including setting color to be designated by rain
```

```
  ggplot(aes(x = jitter(n_groups), y = ln_eggs, color = rain)) +
  # Add scatterplot
  geom_point() +
  # Set theme
  theme_classic() +
  xlab("# social groups visited") + ylab("ln(whipworm eggs)")
```




```
# And a graph to assess colinearity between rain and n-groups...
parasite %>%
  ggplot(aes(x = jitter(n_groups), y = rain)) +
  # Add scatterplot
  geom_point() +
  # Set theme
  theme_classic() +
  xlab("# social groups visited") + ylab("preciptation last 3 mo. (cm)")
```



Question: Is there any visual indication that rain or n_groups might help explain variation in eggs? To me it appears that precipitation might explain variation in the eggs, but number of social groups

does not explain much variation. The n-groups vs eggs plot does not look to have much linear relatinship, but the precip and n social groups vs eggs plot does seem to have a decreasing linear trend, thus I think precip alone might have an effect.

Model evalutation A: Assess standard errors and parameter estimates

stan_glm() fits linear models using a Bayesian approach. However, as is often the case, Bayesian parameter estimates and uncertainties are very similar to those from least-squares estimates or estimates from likelihood approaches, like lm(), when the model priors are flat.

Uninformative flat priors are the default for stan_glm(), so let's fit a multiple regression examining the relationship between rain and n-groups and eggs, without an influence from prior information

```
# stan_glm() has similar syntax to the familiar lm().
# "refresh=0" disables a counter that shows model fitting progress.
# This is useful to see progress fitting complex models, but not needed for this simple model
model_A <- stan_glm(ln_eggs ~ rain + n_groups, data = parasite, refresh=0)

print(model_A)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      ln_eggs ~ rain + n_groups
##  observations: 91
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept)  2.8    0.3
## rain        -0.1    0.0
## n_groups    -0.3    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.8    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

So, how well does this model fit the data? What evidence is there for a relationship between ln_eggs and the two covariates?

The covariates don't predict a lot of variation in ln_eggs [sigma (median = 0.8) much larger than effect sizes of the covariates]. The effect size of n_groups (-0.3) is larger than that of rain (-0.01), but the negative sign of the n_groups coefficient is different from what we hypothesized (baboons who visit more social groups have *fewer* parasites). There is also a reasonable amount of noise in the n_groups parameter MAD_SD = 0.1.

Question: How would you assess the results of this model? It is suprising to see that number of social groups has a negative effect on number of eggs, but there is a lot of uncertainty with that pearameter, as with a mad_sd of 0.1 it has its uncertainty equal to about 1/3 its effect size. Rain has a small but predictably negative effect size of -0.1, with more rain increasing food quality and thus decreasing presence of parasites. Overall though, the 0.8 effect size of sigma shows that the model does not explain much of the variation in the data well.

Model evaluation B: Comparing summaries between stan_glm() and lm()

Let's compare the results above to a more familiar model summary

```
model_B <- lm(ln_eggs ~ rain + n_groups, data = parasite)
summary(model_B)
```

```
##
## Call:
## lm(formula = ln_eggs ~ rain + n_groups, data = parasite)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7544 -0.3842  0.1420  0.5722  1.3690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.77999    0.24408  11.390  < 2e-16 ***
## rain        -0.08639    0.02467  -3.502 0.000727 ***
## n_groups    -0.26787    0.10935  -2.450 0.016279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8316 on 88 degrees of freedom
## Multiple R-squared:  0.1898, Adjusted R-squared:  0.1714
## F-statistic: 10.31 on 2 and 88 DF,  p-value: 9.505e-05
```

Question: How do the models compare? The lm model output shows very similar results in terms of magnitude of the coefficients for each variable (rain is approx equal to negative 0.1 with not much variation, n_groups is approx -0.3 with a lot of variability). The model also still looks not very descriptive here with the adjusted r squared at 0.1714, showing it only explains around 17% of the variation int the data.

Since the results are similar, why use stan_glm() instead of lm()?
* lm() is awesome. We use it all the time!
* The streamlined results of stan_glm() avoid temptations of unreasonably precise estimates and the lure of lexicographic decision rules based on asterisks.
* Most importantly for us: stan_glm() allows easy sampling from posterior distributions of parameters, which allows us to do things like predictions, as well as using cross-validation and informative priors, which we will do next...

Model Evaluation C: Does including n_groups help prediction?

Extra terms can "fit the noise." This results in poor prediction (see regularization slides in lecture 18). Both terms seem to be reasonably informative, but n_groups is noisier and it has a large coef in the direction opposite than expected. Let's see whether including n_groups helps predict parasite burden...

We'll assess this model's ability to predict data using "leave-one-out" cross-validation, where each point in the dataset in turn is removed; the model is fit to the remaining data; we predict the that held-out point; then we compare that prediction to the real value of the point (RoS 11.8).

The function we'll use, loo(), rapidly computes leave-one-out validation. We'll focus on the metric of "expected log predictive density" (elpd, RoS 11.8) as our metric of the ability of the model to predict the held out data. A model with a lower elpd doesn't predict the held-out data as well as a model with higher elpd. However, like predicting coefficients: (A) a small difference in elpd means the models predict similarly well; and (B) the size of the elpd difference (elpd_diff) between models should be assessed with reference to the uncertainty in the elpd differences: In the case of an elpd_diff with a high standard error (elpd_se), the evidence for predictive improvement is uncertain.

Because n_groups was potentially an important covariate (large effect size! statistically significant!), but also a noisy covariate, let's see if the model_A predicts ln_eggs better than a "reduced" model, where the

"n_groups" covariate is dropped and ln_eggs is predicted only with "rain" only...

```
# First, we calculate the predictive power of the full model (with rain and n_groups as covariates)
loo_full <- loo(model_A)
# For details of this cross validation type: print(loo_full)

# Then we fit the reduced model, where ln_eggs is only estimated from rain
model_reduced <- stan_glm(ln_eggs ~ rain, data = parasite, refresh=0)

# Then perform leave-one-out cross-validation on this new model
loo_reduced<-loo(model_reduced)

# Then we compare the ability of the two models to predict held-out ln_eggs data
loo_compare(loo_full, loo_reduced)
```

```
##               elpd_diff se_diff
## model_A        0.0       0.0
## model_reduced -1.9       2.4
```

What does this mean? The full model has slightly more predictive power (model_A has an estimated log predictive score that is slightly higher than that of the reduced model). However, that difference in elpd (1.9) is small and also smaller than the uncertainty in the difference (2.4). So, there is no clear advantage to including n_groups for the purpose of predicting whipworm parasite burden. [See RoS p. 178]

There's also no sign of overfitting (the full model doesn't predict held-out data worse). So, cross-validation shows that including n_groups in the model doesn't hurt our model's predictive ability, but it also doesn't help.

It seems like we're getting mixed messages on whether n_groups is associated with ln_eggs or not. If only we had some more information to help us decide. And.... we do!

Model evaluation D: Reducing uncertainty by including prior information

The ABRP also estimated how n_group related to ln_eggs in 561 fecal samples spanning the period 2011-2014. This is a literal version of "prior information" because the same measurements were taken at the same place prior to the 2015 data we have been analyzing. So, this seems like information that might usefully improve our understanding of the impact of n_groups on ln_eggs, especially because the n_group coefficient was pretty noisy in previous models.

The median coefficient of n_group for explaining ln_eggs from the previous four years was -0.05 with a MAD_SD of 0.03. This is a **much** smaller effect size than that of model_A (Median 2.8, MAD_SD 0.2), so it seems like the data in 2015 were not typical. Because our scope of inference is more general than what specifically happened in 2015, let's run the model informed by this prior distribution and see how it affects our understanding of n_groups impact on parasite load.

```
# First, let's fit a simple regression modeling the relationship between n-groups and ln_eggs, with the

model_uninformed <- stan_glm(ln_eggs ~ n_groups, data = parasite, refresh=0)
print(model_uninformed)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      ln_eggs ~ n_groups
##  observations: 91
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  2.2    0.2
```

```
## n_groups    -0.3    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.9    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Then we'll fit the same model, but inform the slope parameter with the 2011-2014 data
model_informed <- stan_glm(ln_eggs ~ n_groups, data = parasite,
                           prior = normal(-0.05,0.03), refresh=0)
print(model_informed)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      ln_eggs ~ n_groups
##  observations: 91
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  1.9    0.1
## n_groups    -0.1    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.9    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

How did the prior information change our understanding of ln_eggs? The informed model doesn't explain much more of the variance in 2015 whipworm eggs (the sigmas of the two models are the same). But it did shrink the estimate of how much n_groups affect ln_eggs: Now it's a compromise between the information in the prior (saying the effect is small) and the info from 2015, where it was large. The added information also reduced the uncertainty on this estimate (MAD_SD has also shrunk).

Assignment

The code above should help you if you want to use R to understand these analyses. The HW questions ask you to interpret the results. The goal is to ask you to think through what and why we are doing this. You can work in pairs, if you'd like.

These are short answer questions (<100 words each, 2 points each). We're looking for your opinion, but based in sound reasoning. Keep your answer brief and to the point.

Don't worry if your opinion is different from Jason's. That could be a good thing!

Question 1: How much do you value the added information from the p-values and significance levels in model_B vs the results of model_A and why?

Answer 1: I do put a moderate stake in the p-values from the standard lm (model A) just because it is generally nice to know if the results were highly likely to happen from random chance.I am not advocating for lexicographically choosing whether to believe results based on p=.05 or something, but if p=.8 or some crazy large number, it is worth noting that any conclusions may be dubious because the result was very likely

to have occured due to random chance.

Question 2: In parts A and B, it seemed like n_groups was a useful predictor of ln_eggs, but the cross validation exercise in part C suggests that n_groups adds basically no value for predicting ln_eggs. Why?

Answer 2: Part C's tests show that leaving n_groups out keeps the model just about as accurate at predicting values as when it has the groups information. This is because the uncertainty in the effect size of n_groups is so high, that the predictions are not made much more accurate by including it.

Question 3: RoS say that informative priors "stabilize" coefficents. How does the analysis in part D demonstrate that the information from 2011-2014 help stabilize the coefficent for n_groups beyond the evidence provided by the 2015 data?

Answer 3: It appears as if the data from 2015 was unusual and an outlier year. When prior information from the previous four years was added, effect size drastically decreased, likely due to central limit theorem and the tendency of results to be more representative and include less outlines as number of samples increases (especially over time). Thus, the effect size from over 5 years of data (2015 and 4 years prior) is likely more accurate and informative, so the effect of number of social groups is probably truly lower than 2015 estimated.

Question 4: Which of the new approaches in this tutorial do you think is the most useful and why?

Answer 4: I really like the adding of prior information in part D. Knowing that doing science is an iterative process, and that real scientists might be building on a body of existing work and understanding, it is a great thing to have a way to add old understanding to new data in order to fit a model. It just makes sense to me!

Question 5: Given all the analyses in this tutorial, what is your assessment of the relationship between the number of social groups visited by a male baboon over the last year (n_groups) and the amount of whipworm eggs in male baboons (ln_eggs)? Would you keep n_groups in your model?

Answer 5: I personally think that the relationship between number of social groups and whipworm eggs is shaky at best. The effect size seems small, and including the n_groups variable in a model doesnt seem to add much certainty or help to predicting values. Thus, if I were to try to model whipworm egg counts in this species, I would not include n_groups as a predictive variable because it is not too informative