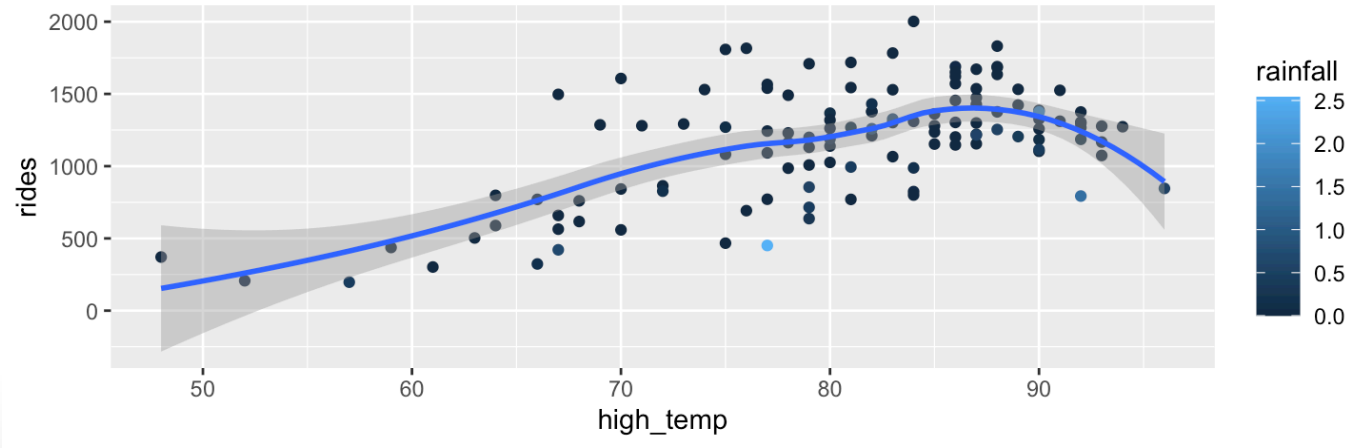
Model Comparison: Implementation

Bayesian Data Analysis

Steve Buyske

Working example

- We will look again at the Citibike usage in 2020.
- This time, though, I've expanded the data from May 1 to August 31, and included variables for the average number of Covid-19 cases in the city over the previous 7 days, for the high temperature, and for the daily rainfall.
  - After looking at the data, I defined a new variable, `lrainfall = log(rainfall + 1)`.
- A plot of rides temperature seems to suggest including a quadratic term for temperature.



First question: group terms

- Let's start with four models, each with different groupings for group-level intercepts.

```
citibike3_fit1a <-
  stan_glmer(
    rides ~ high_temp + I(high_temp^2) + lrainfall + covid_cases +
      (1 | day_of_the_week) + (1 | week_of_the_year),
    data = citibike3 %>% filter(year == "2020"),
    adapt_delta = 0.999,
    chains = 8,
    prior_covariance = decov(2)
  )

citibike3_fit1b <-
  stan_glmer(
    rides ~ high_temp + I(high_temp^2) + lrainfall + covid_cases +
      (1 | week_of_the_year),
    data = citibike3 %>% filter(year == "2020"),
    adapt_delta = 0.999,
    chains = 8,
    prior_covariance = decov(2)
  )
```

```
citibike3_fit1c <-
  stan_glmer(
    rides ~ high_temp + I(high_temp^2) + lrainfall + covid_cases +
      (1 | day_of_the_week),
    data = citibike3 %>% filter(year == "2020"),
    adapt_delta = 0.999,
    chains = 8,
    prior_covariance = decov(2)
  )

citibike3_fit1d <-
  stan_glm(
    rides ~ high_temp + I(high_temp^2) + lrainfall + covid_cases,
    data = citibike3 %>% filter(year == "2020"),
    adapt_delta = 0.999,
    chains = 8
  )
```

The `loo()` function

- The `loo()` function will calculate the PSIS LOO-CV estimate of elppd, and will warn if there is a problem.

```
citi_loo_1a <- loo(citibike3_fit1a)
```

```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argument 'k_threshold = 0.7' in or
```

- You can address the warning by adding the `k_threshold = 0.7` argument; the function will use actual loo-cv for the troublesome observations.

```
citi_loo_1a <- loo(citibike3_fit1a, k_threshold = 0.7)
```

```
## 1 problematic observation(s) found.
## Model will be refit 1 times.


##
## Fitting model 1 out of 1 (leaving out observation 108)
```

```
citi_loo_1b <-   loo(citibike3_fit1b)
citi_loo_1c <-   loo(citibike3_fit1c)
citi_loo_1d <-   loo(citibike3_fit1d)
```

- We use `loo_compare()` to compare the estimated elppd's

```
loo_compare(citi_loo_1a,
            citi_loo_1b,
            citi_loo_1c,
            citi_loo_1d)
```

```
##                 elpd_diff se_diff
## citibike3_fit1c   0.0       0.0
## citibike3_fit1a  -9.6       3.5
## citibike3_fit1d -31.7       7.9
## citibike3_fit1b -32.4       8.0
```

- The `elpd_diff` shows the differences in the estimate elppd from the best model on the top row, while `se_diff` shows an estimate standard error for that difference.

First decision

```
loo_compare(citi_loo_1a,
            citi_loo_1b,
            citi_loo_1c,
            citi_loo_1d)
```

```
##                 elpd_diff se_diff
## citibike3_fit1c   0.0       0.0
## citibike3_fit1a  -9.6       3.5
## citibike3_fit1d -31.7       7.9
## citibike3_fit1b -32.4       8.0
```

- It looks like `citibike3_fit1c` is really much better than the other models.

Second choice: covariates

- Next let's try models with different population-level covariates.

```
citibike3_fit2 <- stan_glmer(rides ~ high_temp + I(high_temp^2) + lrainfall + (1 | day_of_the_week),
                             data = citibike3 %>% filter(year == "2020"), adapt_delta = 0.999,
                             chains = 8, prior_covariance = decov(2))

citibike3_fit3 <- stan_glmer(rides ~ lrainfall + (1 | day_of_the_week),
                             data = citibike3 %>% filter(year == "2020"), adapt_delta = 0.999,
                             chains = 8, prior_covariance = decov(2))

citibike3_fit4 <- stan_glmer(rides ~ high_temp + I(high_temp^2)  + (1 | day_of_the_week),
                             data = citibike3 %>% filter(year == "2020"), adapt_delta = 0.999,
                             chains = 8, prior_covariance = decov(2))

citibike3_fit5 <- stan_glmer(rides ~ (high_temp + I(high_temp^2) * lrainfall + (1 | day_of_the_week),
                             data = citibike3 %>% filter(year == "2020"), adapt_delta = 0.999,
                             chains = 8, prior_covariance = decov(2))
```

- How do they compare?

```
loo_compare(
  citi_loo_1c,
  citi_loo_2,
  citi_loo_3,
  citi_loo_4,
  citi_loo_5
)
```

```
##                 elpd_diff se_diff
## citibike3_fit1c   0.0       0.0
## citibike3_fit5  -11.0       6.7
## citibike3_fit2  -11.9       5.3
## citibike3_fit4  -20.1       7.7
## citibike3_fit3  -64.0      10.9
```

- It looks like `citibike3_fit1c`, `citibike3_fit2`, and `citibike3_fit5` are all worth considering.

Big models

- Since `citibike3_fit1c` has all the covariates and is readily interpretable, let's take a final look at that.

```
describe_posterior(citibike3_fit1c, centrality = "mean", ci = 0.9, rope_ci = 0.9)
```

```
## # Description of Posterior Distributions
##
## Parameter      |      Mean |                 90% CI |      pd |          90% ROPE | % in ROPE |  Rhat |      ESS
## ------------------------------------------------------------------------------------------------------------
## (Intercept)    | -3390.926 | [-4858.713, -1920.231] |  99.98% | [-38.549, 38.549] |         0 | 1.002 | 4689.525
## high_temp      |   110.296 | [   74.187,   149.099] | 100.00% | [-38.549, 38.549] |         0 | 1.002 | 4677.819
## I(high_temp^2) |    -0.624 | [   -0.872,    -0.390] |  99.98% | [-38.549, 38.549] |       100 | 1.002 | 4734.162
## lrainfall      |  -497.167 | [ -648.160,  -343.499] | 100.00% | [-38.549, 38.549] |         0 | 1.001 | 7398.357
## covid_cases    |    -0.345 | [   -0.453,    -0.235] | 100.00% | [-38.549, 38.549] |       100 | 1.000 | 7879.726
```

- It looks like `high_temp`, `lrainfall`, and `covid_cases` all have non-negligible effects.
- Keep in mind, though, that the scales are different.

```
citibike3 %>%
  filter(year == "2020") %>%
  select(high_temp, lrainfall, covid_cases) %>%
  summarize_all(quantile, prob = c(0.25, 0.75))
```

```
## # A tibble: 2 x 3
##   high_temp lrainfall covid_cases
##       <dbl>     <dbl>       <dbl>
## 1      76.5    0             254
## 2      87      0.0535        606.
```

- A little arithmetic shows that the difference from the 1st quartile to the 3rd changes the number of rides by about 24, -27, and 95, respectively, for `high_temp`, `lrainfall`, and `covid_cases`.

- By way of comparison, look how much larger the day of the week effect is:

```
describe_posterior(citibike3_fit1c, centrality = "mean", ci = 0.9, rope_ci = 0.9, effect = "random")
```

```
## # Description of Posterior Distributions
##
## Parameter                                        |      Mean |                90% CI |      pd |          90% ROPE | % in ROPE |
## ------------------------------------------------------------------------------------------------------------------------------
## day_of_the_week:Sun                              |   203.585 | [  53.904,   359.774] |  98.49% | [-38.549, 38.549] |     0.000 |
## day_of_the_week:Mon                              |  -130.529 | [-282.844,    19.251] |  92.51% | [-38.549, 38.549] |    11.137 |
## day_of_the_week:Tue                              |  -104.897 | [-255.989,    41.654] |  88.22% | [-38.549, 38.549] |    19.289 |
## day_of_the_week:Wed                              |   -53.254 | [-198.431,   102.706] |  72.84% | [-38.549, 38.549] |    30.274 |
## day_of_the_week:Thu                              |  -129.006 | [-280.403,    21.762] |  92.39% | [-38.549, 38.549] |    12.498 |
## day_of_the_week:Fri                              |   -99.071 | [-250.648,    48.098] |  86.60% | [-38.549, 38.549] |    20.608 |
## day_of_the_week:Sat                              |   301.341 | [ 150.216,   451.866] |  99.89% | [-38.549, 38.549] |     0.000 |
## Sigma[day_of_the_week:(Intercept),(Intercept)] | 47001.916 | [9283.020, 87495.973] | 100.00% | [-38.549, 38.549] |     0.000 |
```