Module 7 (Bayesian) Exercise

Preliminaries

- Install RTools from here:
- Install and initialize the packages rstan, rstanarm, and shinystan.
  - Be sure to initialize any other packages you would normally use.
- In a new or existing RStudio Project, create an RMarkdown document.

Analysis of Well Switching

- Check out the descriptions of the datasets included with the rstanarm package here:
- Using the **wells** dataset, perform a Bayesian linear regression that uses arsenic level, distance from well, and years of education to predict well-switching.

```
wellsLM <- stan_glm(switch ~ arsenic +
    dist + educ,
    data = wells, family = gaussian)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
##
## Gradient evaluation took 0.001 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 1.209 seconds (Warm-up)
##                1.29 seconds (Sampling)
##                2.499 seconds (Total)
##
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
```

```
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 1.198 seconds (Warm-up)
##                1.266 seconds (Sampling)
##                2.464 seconds (Total)
##
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
##
## Gradient evaluation took 0.001 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 1.179 seconds (Warm-up)
##                1.336 seconds (Sampling)
##                2.515 seconds (Total)
##
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
```

```
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 1.153 seconds (Warm-up)
##                1.315 seconds (Sampling)
##                2.468 seconds (Total)
```

- View the summary of the regression output.

```
summary(wellsLM)
```

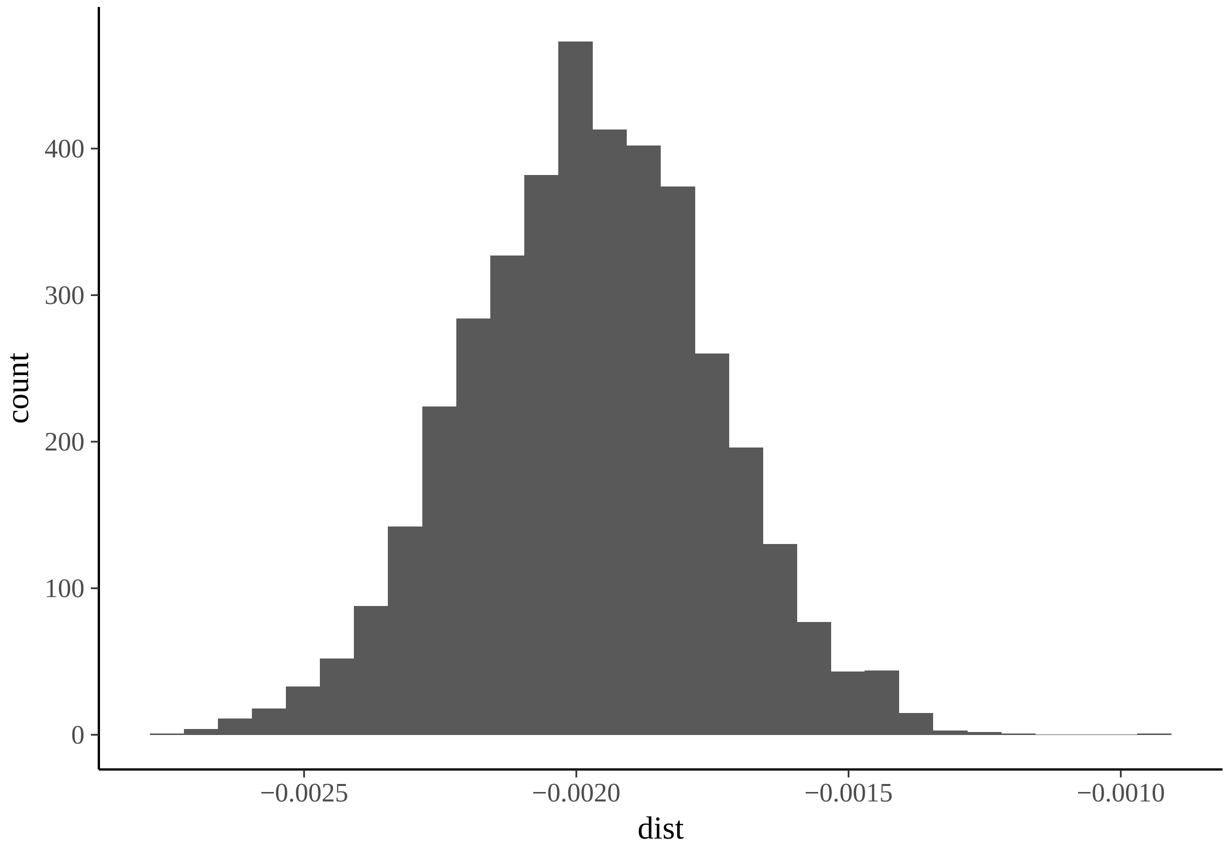
```
##
## Model Info:
##
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      switch ~ arsenic + dist + educ
##  algorithm:    sampling
##  priors:       see help('prior_summary')
##  sample:       4000 (posterior sample size)
##  observations: 3020
##  predictors:   4
##
## Estimates:
##                 mean    sd      2.5%    25%     50%     75%     97.5%
## (Intercept)       0.5     0.0     0.4     0.5     0.5     0.5     0.5
## arsenic           0.1     0.0     0.1     0.1     0.1     0.1     0.1
## dist              0.0     0.0     0.0     0.0     0.0     0.0     0.0
## educ              0.0     0.0     0.0     0.0     0.0     0.0     0.0
## sigma             0.5     0.0     0.5     0.5     0.5     0.5     0.5
## mean_PPD          0.6     0.0     0.6     0.6     0.6     0.6     0.6
## log-posterior -2067.2     1.6 -2071.1 -2068.1 -2066.9 -2066.0 -2065.2
##
## Diagnostics:
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  4000
## arsenic       0.0  1.0  4000
## dist          0.0  1.0  4000
## educ          0.0  1.0  4000
## sigma         0.0  1.0  4000
## mean_PPD      0.0  1.0  4000
## log-posterior 0.0  1.0  1861
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

- Create a histogram of the posterior distribution of the *dist* variable

```
wells_posterior <- wellsLM %>% as.tibble()

ggplot(wells_posterior, aes(x=dist)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Use the leave-one-out method of measuring out-of-sample predictivenes to compute the expected log predictive density.

```
loo(wellsLM)
```

```
## Computed from 4000 by 3020 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2063.0 15.0
## p_loo         4.0  0.1
## looic      4126.0 30.0
##
## All Pareto k estimates are good (k < 0.5)
## See help('pareto-k-diagnostic') for details.
```

Analysis of doctors visits

- If necessary, install the package "Ecdat" and initialize it.
- Using the dataset **Doctor**, perform a Bayesian Poisson regression that estimates the number of doctor visits as a function of health care access and health status.

```r
library(Ecdat)
doctordata <- Doctor

doctorvisits_poissonreg <- stan_glm(doctor ~ access +
    health,
    data = doctordata, family = poisson)
```

```
##
## SAMPLING FOR MODEL 'count' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.629 seconds (Warm-up)
##                0.615 seconds (Sampling)
##                1.244 seconds (Total)
##
##
## SAMPLING FOR MODEL 'count' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.609 seconds (Warm-up)
```

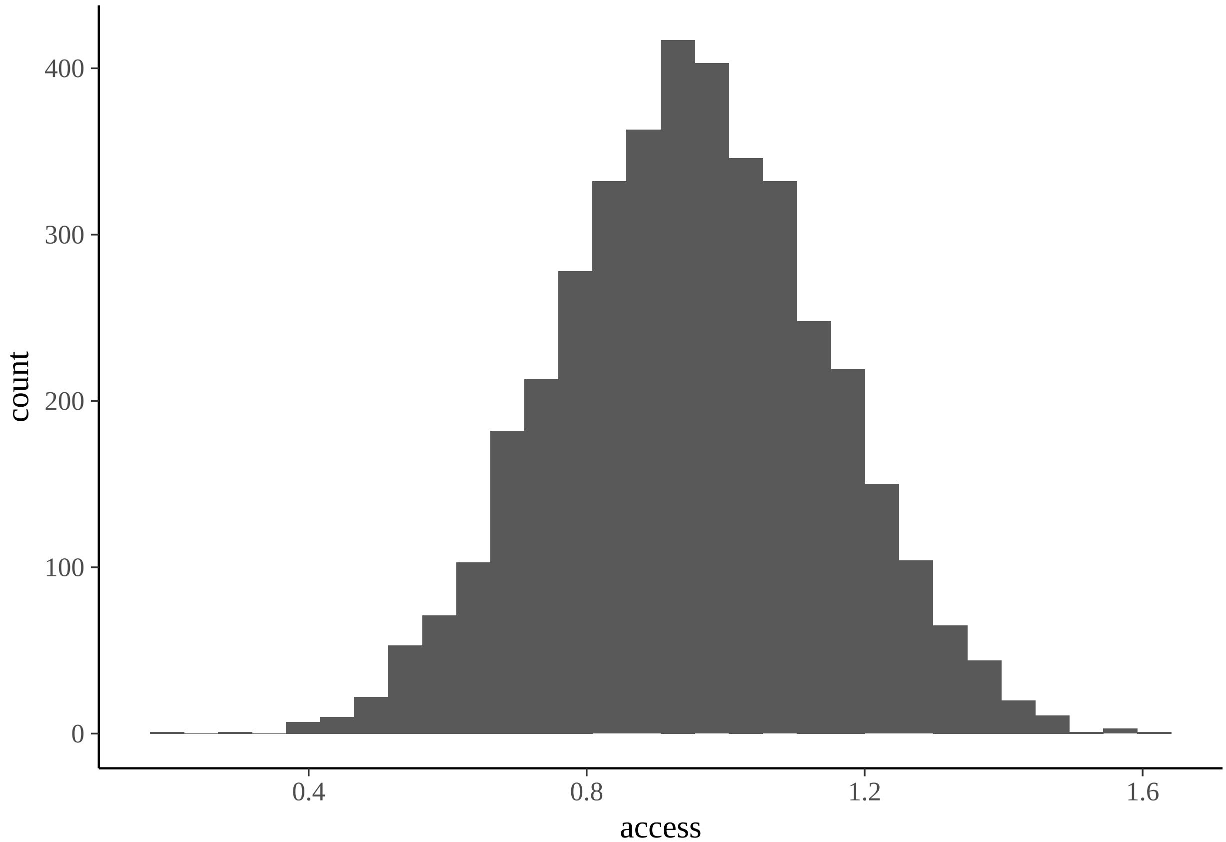
```
##                 0.587 seconds (Sampling)
##                 1.196 seconds (Total)
##
##
## SAMPLING FOR MODEL 'count' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.601 seconds (Warm-up)
##                0.62 seconds (Sampling)
##                1.221 seconds (Total)
##
##
## SAMPLING FOR MODEL 'count' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.632 seconds (Warm-up)
##                0.615 seconds (Sampling)
##                1.247 seconds (Total)
```

- Once again view the summary of the output but this time view a graphical plot of the credible intervals for *access*.

```
doctorvisits_posterior <- doctorvisits_poissonreg %>% as.tibble()

ggplot(doctorvisits_posterior, aes(x=access)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Use the Watanabe-Aikake information criterion to view the predictiveness of the Poisson model.

```
waic(doctorvisits_poissonreg)
```

```
## Warning: 6 (1.2%) p_waic estimates greater than 0.4.
## We recommend trying loo() instead.

## Computed from 4000 by 485 log-likelihood matrix
##
##           Estimate    SE
## elpd_waic  -1116.6 108.9
## p_waic        22.9  10.0
## waic        2233.2 217.9

## Warning: 6 (1.2%) p_waic estimates greater than 0.4.
## We recommend trying loo() instead.
```