STA365 Assignment 4

Raj Patel

Multilevel Regression and Poststratification

```
model_code <- "
data {
  int<lower = 0> n_survey;
  int<lower = 0> n_age;
  int<lower = 0> n_eth;
  int<lower = 0> n_income;
  int<lower = 0> n_state;
  int cat[n_survey];
  vector[n_survey] male;
  int age[n_survey];
  int eth[n_survey];
  int income[n_survey];
  int state[n_survey];
  // Bit for poststratification
  int<lower = 0> n_pred;
  vector[n_pred] male_pred;
  int age_pred[n_pred];
  int eth_pred[n_pred];
  int income_pred[n_pred];
  int state_pred[n_pred];
  int N_in_cell_pred[n_pred];
}
parameters {
  real mu;
  real beta;
  vector[n_age] z_age;
  vector[n_eth] z_eth;
  vector[n_income] z_income;
  vector[n_state] z_state;
  real<lower= 0> tau_age;
  real<lower= 0> tau_eth;
  real<lower= 0> tau_income;
  real<lower= 0> tau_state;
}
transformed parameters {
  vector[n_age] alpha_age = tau_age * z_age;
  vector[n_eth] alpha_eth = tau_eth * z_eth;
  vector[n_income] alpha_income = tau_income * z_income;
  vector[n_state] alpha_state = tau_state * z_state;
}
model {
  cat ~ binomial_logit(1, mu + beta*male + alpha_age[age] +
alpha_eth[eth] + alpha_income[income] + alpha_state[state]);
  z_age ~ normal(0,1);
  z_eth ~ normal(0,1);
  z_income ~ normal(0,1);
```

```
  z_state ~ normal(0,1);
  tau_age ~ normal(0,1);
  tau_eth ~ normal(0,1);
  tau_income ~ normal(0,1);
  tau_state ~ normal(0,1);
  mu ~ normal(0,1);
  beta ~ normal(0,1);
}
generated quantities {
  int cat_pred[n_pred];
  for (n in 1:n_pred) {
    cat_pred[n] = binomial_rng(N_in_cell_pred[n],
   1.0/(1.0 + exp(-(mu +
                    beta*male_pred[n] +
                    alpha_age[age_pred[n]] +
                    alpha_eth[eth_pred[n]] +
                    alpha_income[income_pred[n]] +
                    alpha_state[state_pred[n]] ))));
  }
}
"
```

```
# Input Data for the Stan Model
stan_data <- list(
  n_survey = length(survey$cat_pref),
  n_age = length(unique(poststrat$age)),
  n_eth = length(unique(poststrat$eth)),
  n_income = length(unique(poststrat$income)),
  n_state = length(unique(poststrat$state)),
  cat = survey$cat_pref,
  male = survey$male,
  age = survey$age,
  eth = survey$eth,
  income = survey$income,
  state = survey$state,
  n_pred = length(poststrat$male),
  male_pred = poststrat$male,
  age_pred = poststrat$age,
  eth_pred = poststrat$eth,
  income_pred = poststrat$income,
  state_pred = poststrat$state,
  N_in_cell_pred = poststrat$N
)
fit <- stan(model_code = model_code, data = stan_data, control=list(adapt_delta=0.95))
```

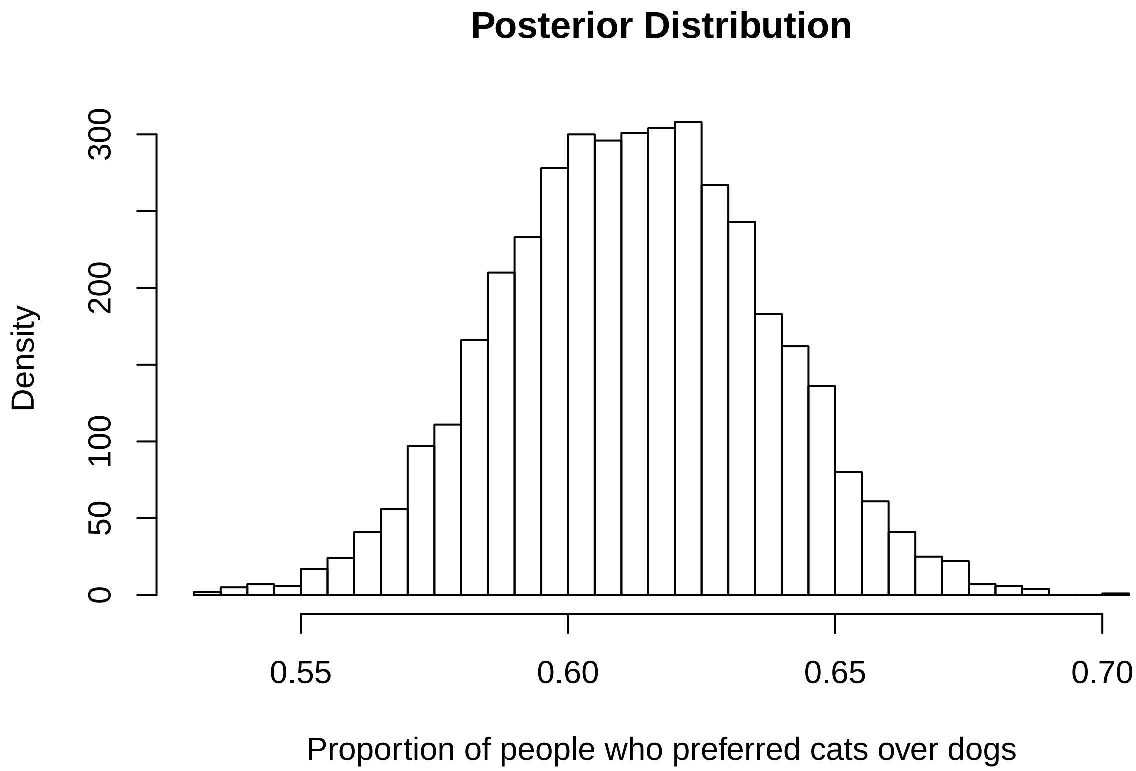
```
##
## SAMPLING FOR MODEL 'cadf6f7f9788b36b707aa786816ac6df' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000364 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 3.64 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
```

```
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 11.3857 seconds (Warm-up)
## Chain 1:                10.5183 seconds (Sampling)
## Chain 1:                21.904 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'cadf6f7f9788b36b707aa786816ac6df' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000211 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 2.11 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 11.9192 seconds (Warm-up)
## Chain 2:                10.7755 seconds (Sampling)
## Chain 2:                22.6947 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'cadf6f7f9788b36b707aa786816ac6df' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000206 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.06 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 11.0571 seconds (Warm-up)
## Chain 3:                10.6694 seconds (Sampling)
## Chain 3:                21.7265 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'cadf6f7f9788b36b707aa786816ac6df' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000168 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.68 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 10.5463 seconds (Warm-up)
## Chain 4:                9.61034 seconds (Sampling)
## Chain 4:                20.1566 seconds (Total)
## Chain 4:
```

```r
## Propotion of people liking cat over dogs
n_pop <- sum(poststrat$N)
cat <- rstan::extract(fit,"cat_pred")


# This gives a 4000x6300 matrix so each row
# is a sample from the posterior predictive!
prop <- rowSums(cat$cat_pred)/n_pop
hist(prop,breaks=30,xlab = 'Proportion of people who preferred cats over dogs',
     ylab='Density', main='Posterior Distribution')
```


Posterior Distribution
Density
0
50
100
200
300
0.55
0.60
0.65
0.70
Proportion of people who preferred cats over dogs