

Assignment2_Frank_Jiang

Frank Jiang

4/29/2020

1 The impact of Medicaid Expansion on Voter Participation

```
library(brms)
```

```
## Loading required package: Rcpp
```

```
## Loading 'brms' package (version 2.12.0). Useful instructions  
## can be found by typing help('brms'). A more detailed introduction  
## to the package is available through vignette('brms_overview').
```

```
##
```

```
## Attaching package: 'brms'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      ar
```

```
library(haven)  
unzip("100.00019026_supp.zip")  
oregon<- as_factor(read_dta(file.path('19026_supp', 'Data', 'individual_voting_data.dta')))  
table(oregon$treatment)
```

```
##
```

```
##      0      1
```

```
## 45088 29834
```

1.1 Priors and Priors Predictive Distribution with brms

For this part, we are focusing on the Intent to treat effect on being selected for the Medicaid lottery on the outcome variable of Registered as a republican (variable rep_1) using a Bernoulli models.

```
get_prior(rep_1~treatment+numhh_list,family='bernoulli',data=oregon)
```

```
##           prior      class                coef group  
## 1                        b  
## 2                        b numhh_listsignedselfupP1additionalperson  
## 3                        b numhh_listsignedselfupP2additionalpeople  
## 4                        b                        treatment  
## 5 student_t(3, 0, 10) Intercept
```

```

##   resp dpar nlpar bound
## 1
## 2
## 3
## 4
## 5

prior1<- c(set_prior('normal(0.2,0.5)',class='b',coef='treatment'),
           set_prior('normal(0,1)',class='b',coef='numhh_listsignedselfupP1additionalperson'),
           set_prior('normal(0,1)',class='b',coef='numhh_listsignedselfupP2additionalpeople'),
           set_prior('normal(0,5)',class='Intercept'))
ppd1<- brm(rep_1~treatment+numhh_list,family='bernoulli',data=oregon,prior=prior1,sample_prior = 'only')

## Compiling the C++ model

## Start sampling

##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.019 seconds (Warm-up)
## Chain 1:           0.019 seconds (Sampling)
## Chain 1:           0.038 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.02 seconds (Warm-up)
## Chain 2: 0.018 seconds (Sampling)
## Chain 2: 0.038 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.02 seconds (Warm-up)
## Chain 3: 0.021 seconds (Sampling)
## Chain 3: 0.041 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.02 seconds (Warm-up)
## Chain 4:           0.019 seconds (Sampling)
## Chain 4:           0.039 seconds (Total)
## Chain 4:
```

```
summary(ppd1)
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: rep_1 ~ treatment + numhh_list
## Data: oregon (Number of observations: 74922)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Population-Level Effects:
##
```

	Estimate	Est.Error	1-95% CI	u-95% CI
## Intercept	-0.09	4.98	-9.81	9.67
## treatment	0.21	0.50	-0.75	1.20
## numhh_listsignedselfupP1additionalperson	-0.00	1.00	-2.02	2.02
## numhh_listsignedselfupP2additionalpeople	-0.02	1.00	-2.02	1.92

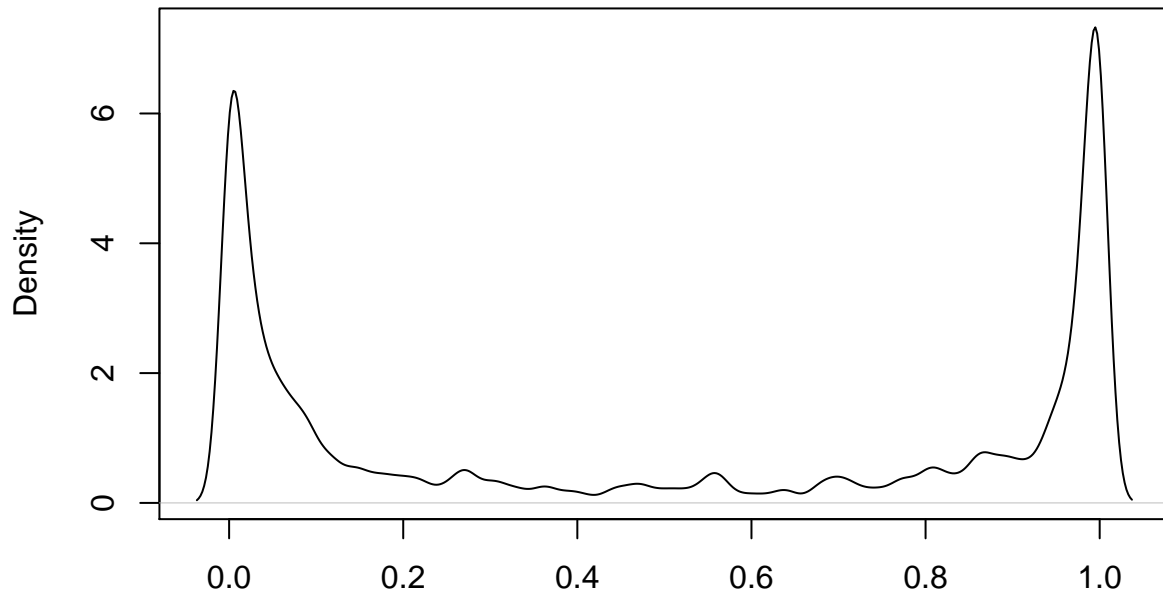
```
##
```

	Rhat	Bulk_ESS	Tail_ESS
## Intercept	1.00	4899	3069
## treatment	1.00	4006	3032
## numhh_listsignedselfupP1additionalperson	1.00	4054	2638
## numhh_listsignedselfupP2additionalpeople	1.00	3722	2783

```
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
ppe1<- pp_expect(ppd1,nsamples = 400)
plot(density(ppe1))
```

density.default(x = ppe1)



N = 29968800 Bandwidth = 0.01235

As the plot show, we can conclude that based on our prior distribution, the density plot of our probability that $y_n = 1$ is generally distributed around 0 and 1.

1.2 Posterior Distribution

```
post_1<- brm(rep_1~treatment+numhh_list,family='bernoulli',data=oregon,prior=prior1)
```

```
## Compiling the C++ model
```

```
## recompiling to avoid crashing R session
```

```
## Start sampling
```

```
##
```

```
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 1).
```

```
## Chain 1:
```

```
## Chain 1: Gradient evaluation took 0.004 seconds
```

```
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 40 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: 35.996 seconds (Warm-up)
## Chain 1: 25.362 seconds (Sampling)
## Chain 1: 61.358 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.003 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 30 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: 32.286 seconds (Warm-up)
## Chain 2: 29.295 seconds (Sampling)
## Chain 2: 61.581 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.003 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 30 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: 30.369 seconds (Warm-up)
## Chain 3: 30.231 seconds (Sampling)
## Chain 3: 60.6 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '5d7349a9ce52b76e555917b10230b739' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.004 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 40 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: 35.951 seconds (Warm-up)
## Chain 4: 25.903 seconds (Sampling)
## Chain 4: 61.854 seconds (Total)
## Chain 4:

```

```
summary(post_1)
```

```

## Family: bernoulli
## Links: mu = logit
## Formula: rep_1 ~ treatment + numhh_list
## Data: oregon (Number of observations: 74922)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup samples = 4000
##
## Population-Level Effects:
##
##               Estimate Est.Error 1-95% CI u-95% CI
## Intercept      -2.60      0.02    -2.63    -2.56
## treatment         0.03      0.03    -0.02     0.09
## numhh_listsignedselfupP1additionalperson  0.51      0.03     0.45     0.56
## numhh_listsignedselfupP2additionalpeople  0.11      0.27    -0.47     0.62
##
##               Rhat Bulk_ESS Tail_ESS
## Intercept      1.00    4056    3288
## treatment      1.00    3933    2879
## numhh_listsignedselfupP1additionalperson 1.00    3593    2924
## numhh_listsignedselfupP2additionalpeople 1.00    4290    3220

```

```
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
hypothesis(post_1, 'treatment>0')
```

```
## Hypothesis Tests for class b:
##      Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob
## 1 (treatment) > 0      0.03      0.03    -0.01     0.08       7.02      0.88
##      Star
## 1
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

The posterior probability that winning the lottery has a positive effect is 1. However, we cannot conclude that winning the lottery results in registered as a republican.

1.3 Alternative Model

```
get_prior(rep_1~treatment+young+old+female+male,family='bernoulli',data=oregon)
```

```
##           prior      class      coef group resp dpar nlpar bound
## 1                b
## 2                b    female
## 3                b      male
## 4                b      old
## 5                b treatment
## 6                b    young
## 7 student_t(3, 0, 10) Intercept
```

```
prior2<- c(set_prior('normal(0.2,0.5)',class='b',coef='treatment'),
           set_prior('normal(-0.3,0.5)',class='b',coef='young'),
           set_prior('normal(0.4,0.5)',class='b',coef='old'),
           set_prior('normal(0,0.5)',class='b',coef='female'),
           set_prior('normal(0.1,0.2)',class='b',coef='male'))
post_2<-brm(rep_1~treatment+young+old+female+male,family='bernoulli',data=oregon,prior=prior2)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL '4a7927c86d7d569c9bf0635319cb638f' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.004 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 40 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: 243.582 seconds (Warm-up)
## Chain 1:                329.394 seconds (Sampling)
## Chain 1:                572.976 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '4a7927c86d7d569c9bf0635319cb638f' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.004 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 40 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: 282.289 seconds (Warm-up)
## Chain 2:                338.221 seconds (Sampling)
## Chain 2:                620.51 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '4a7927c86d7d569c9bf0635319cb638f' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.004 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 40 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: 297.174 seconds (Warm-up)
## Chain 3: 402.243 seconds (Sampling)
## Chain 3: 699.417 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '4a7927c86d7d569c9bf0635319cb638f' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.004 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 40 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: 275.259 seconds (Warm-up)
## Chain 4: 301.457 seconds (Sampling)
## Chain 4: 576.716 seconds (Total)
## Chain 4:

```

```
summary(post_2)
```

```

## Family: bernoulli
## Links: mu = logit
## Formula: rep_1 ~ treatment + young + old + female + male
## Data: oregon (Number of observations: 74922)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup samples = 4000
##
## Population-Level Effects:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -2.55      0.41   -3.37   -1.76 1.00    1795    2187
## treatment     0.13      0.03    0.08    0.18 1.00    2786    2164

```

```
## young      -0.15      0.36     -0.85      0.56 1.00      1776      2145
## old         0.26      0.36     -0.45      0.97 1.00      1781      2163
## female      0.09      0.19     -0.26      0.47 1.00      1838      2023
## male        0.04      0.19     -0.31      0.42 1.00      1796      2109
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
result<- loo_subsample(post_1,post_2,compare=TRUE)
```

```
## Warning: Different subsamples in 'post_1' and 'post_2'. Naive diff SE is used.
```

```
result
```

```
## Output of model 'post_1':
##
## Computed from 4000 by 400 subsampled log-likelihood
## values from 74922 total observations.
##
##      Estimate      SE subsampling SE
## elpd_loo -20719.2 180.6           0.8
## p_loo      3.6    0.0           0.5
## looic     41438.4 361.2           1.6
## -----
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Output of model 'post_2':
##
## Computed from 4000 by 400 subsampled log-likelihood
## values from 74922 total observations.
##
##      Estimate      SE subsampling SE
## elpd_loo -20764.6 180.8          10.1
## p_loo      4.1    0.0           0.6
## looic     41529.2 361.6          20.1
## -----
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Model comparisons:
##      elpd_diff se_diff subsampling_se_diff
## post_1    0.0      0.0      0.0
## post_2  45.4    255.6    10.1
```

```
result$diffs
```

```
##           elpd_diff se_diff subsampling_se_diff
## post_1    0.0      0.0      0.0
## post_2  45.4     255.6     10.1
```

According to our ELPD calculation comparison done by `loo_subsample()`, we can have some confidence in concluding that the second model which includes male, female, old, and young as predictors improves the estimated Expected Log Predictive Density.

2. Coronavirus in NYC

```
ROOT <- "https://raw.githubusercontent.com/nychealth"
NYC <- readr::read_csv(paste0(ROOT, "/coronavirus-data/master/case-hosp-death.csv"))
```

```
## Parsed with column specification:
## cols(
##   DATE_OF_INTEREST = col_character(),
##   CASE_COUNT = col_double(),
##   HOSPITALIZED_COUNT = col_double(),
##   DEATH_COUNT = col_double()
## )
```

```
NYC$day<- 1:nrow(NYC)
```

2.1 Negative Binomial Model

```
get_prior(CASE_COUNT~poly(day,degree=2,row=FALSE),data=NYC,family='negbinomial')
```

```
##           prior      class      coef group resp dpar
## 1                b
## 2                b polydaydegreeEQ2rawEQFALSE1
## 3                b polydaydegreeEQ2rawEQFALSE2
## 4 student_t(3, 8, 10) Intercept
## 5   gamma(0.01, 0.01)      shape
## nlpar bound
## 1
## 2
## 3
## 4
## 5
```

```
prior3<- c(set_prior('normal(0,10)',class='b',coef='polydaydegreeEQ2rawEQFALSE1'),
           set_prior('normal(0,5)',class='b',coef='polydaydegreeEQ2rawEQFALSE2'),
           set_prior('normal(0,10)',class='shape'))
post_3<- brm(CASE_COUNT~poly(day,degree=2,row=FALSE),data=NYC,family='negbinomial',prior=prior3)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL '1c9a77b10939594117d23962b0adf385' NOW (CHAIN 1).
```

```

## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.185 seconds (Warm-up)
## Chain 1:                0.143 seconds (Sampling)
## Chain 1:                0.328 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '1c9a77b10939594117d23962b0adf385' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.18 seconds (Warm-up)
## Chain 2:                0.166 seconds (Sampling)
## Chain 2:                0.346 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '1c9a77b10939594117d23962b0adf385' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.206 seconds (Warm-up)
## Chain 3:                0.166 seconds (Sampling)
## Chain 3:                0.372 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '1c9a77b10939594117d23962b0adf385' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.244 seconds (Warm-up)
## Chain 4:                0.177 seconds (Sampling)
## Chain 4:                0.421 seconds (Total)
## Chain 4:

```

```
summary(post_3)
```

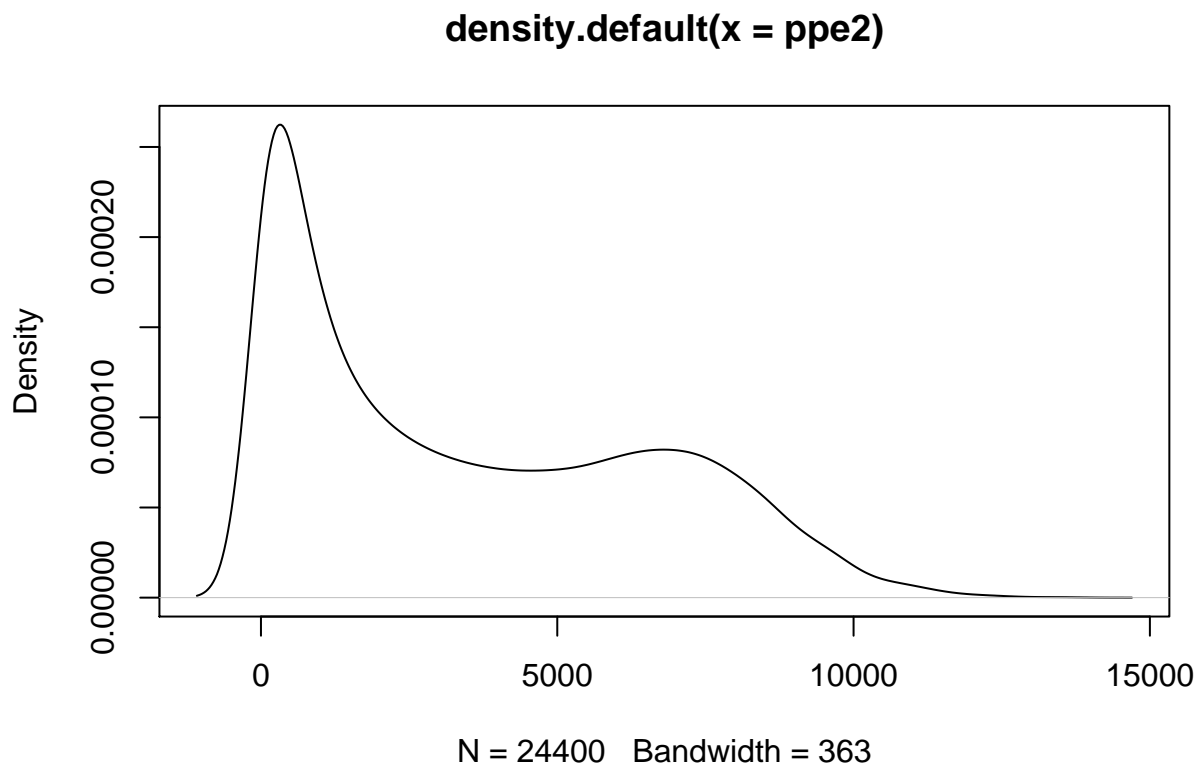
```

## Family: negbinomial
## Links: mu = log; shape = identity
## Formula: CASE_COUNT ~ poly(day, degree = 2, raw = FALSE)
## Data: NYC (Number of observations: 61)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##

```

```
## Population-Level Effects:
##               Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS
## Intercept           7.40      0.10   7.21   7.59 1.00    3048
## polydaydegreeEQ2rawEQFALSE1  7.08      0.85   5.43   8.74 1.00    4045
## polydaydegreeEQ2rawEQFALSE2 -10.45     0.93  -12.27  -8.65 1.00    3923
##               Tail_ESS
## Intercept           2585
## polydaydegreeEQ2rawEQFALSE1  3584
## polydaydegreeEQ2rawEQFALSE2  3222
##
## Family Specific Parameters:
##               Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape           1.81      0.32   1.25   2.50 1.00    3149    2846
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
ppe2<- pp_expect(post_3,nsample=400)
plot(density(ppe2))
```



2.2 Poisson Model

```
get_prior(CASE_COUNT~poly(day,degree=2,row=FALSE),data=NYC,family='poisson')
```

```
##               prior      class               coef group resp dpar
```

```

## 1                                b
## 2                                b polydaydegreeEQ2rawEQFALSE1
## 3                                b polydaydegreeEQ2rawEQFALSE2
## 4 student_t(3, 8, 10) Intercept
## nlpar bound
## 1
## 2
## 3
## 4

prior4<- c(set_prior('normal(0,10)',class='b',coef='polydaydegreeEQ2rawEQFALSE1'),
           set_prior('normal(0,5)',class='b',coef='polydaydegreeEQ2rawEQFALSE2'))
post_4<- brm(CASE_COUNT~poly(day,degree=2,raw=FALSE),data=NYC,prior=prior4,family='poisson')

## Compiling the C++ model

## Start sampling

##
## SAMPLING FOR MODEL '213f4616c9bf67ce378e13cd45fe067d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 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: 0.114 seconds (Warm-up)
## Chain 1:                0.076 seconds (Sampling)
## Chain 1:                0.19 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '213f4616c9bf67ce378e13cd45fe067d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.112 seconds (Warm-up)
## Chain 2: 0.078 seconds (Sampling)
## Chain 2: 0.19 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '213f4616c9bf67ce378e13cd45fe067d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.114 seconds (Warm-up)
## Chain 3: 0.073 seconds (Sampling)
## Chain 3: 0.187 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '213f4616c9bf67ce378e13cd45fe067d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.107 seconds (Warm-up)
## Chain 4: 0.076 seconds (Sampling)
## Chain 4: 0.183 seconds (Total)
## Chain 4:
```

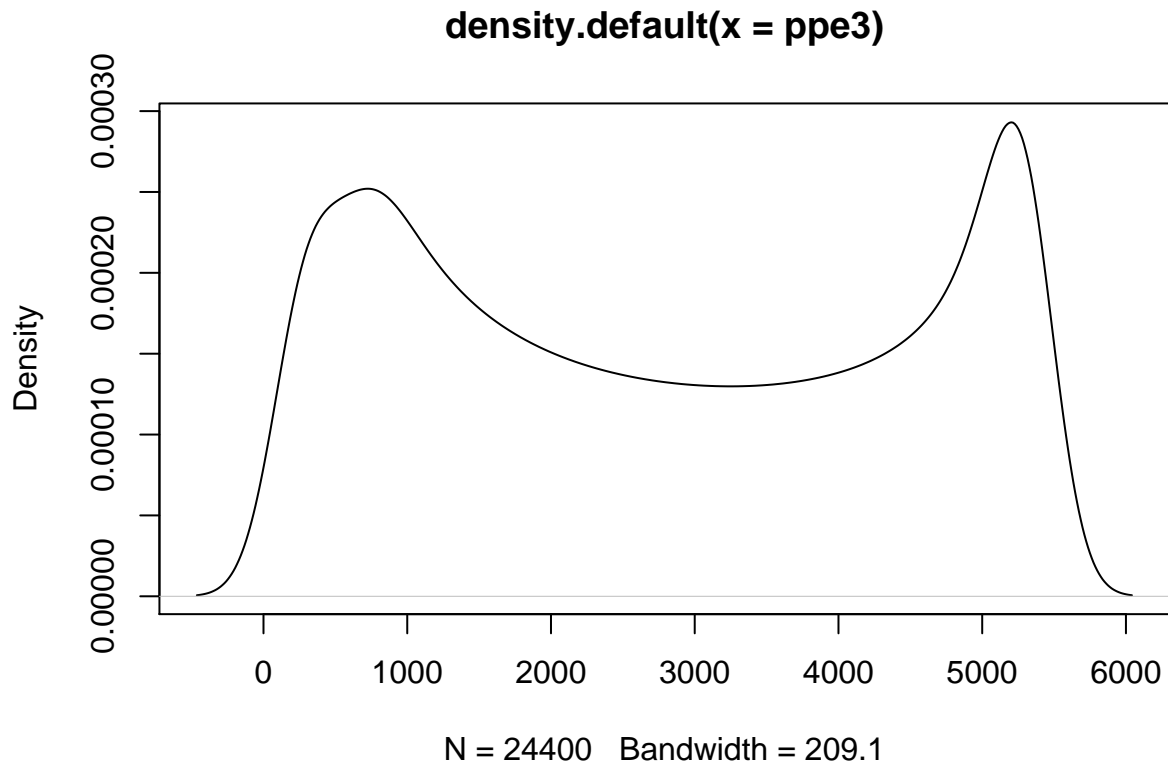
```
summary(post_4)
```

```
## Family: poisson
## Links: mu = log
## Formula: CASE_COUNT ~ poly(day, degree = 2, raw = FALSE)
## Data: NYC (Number of observations: 61)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup samples = 4000
##
## Population-Level Effects:
##
```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS
## Intercept	7.60	0.00	7.60	7.61	1.00	1969
## polydaydegreeEQ2rawEQFALSE1	3.09	0.03	3.04	3.15	1.00	1876
## polydaydegreeEQ2rawEQFALSE2	-6.58	0.03	-6.64	-6.52	1.00	1879

```
## Tail_ESS
## Intercept 2291
## polydaydegreeEQ2rawEQFALSE1 2219
## polydaydegreeEQ2rawEQFALSE2 2125
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
ppe3<- pp_expect(post_4,nsample=400)
plot(density(ppe3))
```



2.3 Model Comparison

```
loo(post_3,post_4,relloo=T)
```

```
## No problematic observations found. Returning the original 'loo' object.
```

```
## 42 problematic observation(s) found.
```

```
## The model will be refit 42 times.
```

```
##
```

```
## Fitting model 1 out of 42 (leaving out observation 2)
```

```
##
```

```
## Fitting model 2 out of 42 (leaving out observation 3)
```

```
##
```

```
## Fitting model 3 out of 42 (leaving out observation 4)
```

```
##
```

```
## Fitting model 4 out of 42 (leaving out observation 5)
```

```
##
```

```
## Fitting model 5 out of 42 (leaving out observation 6)
```

```
##
## Fitting model 6 out of 42 (leaving out observation 7)

##
## Fitting model 7 out of 42 (leaving out observation 8)

##
## Fitting model 8 out of 42 (leaving out observation 9)

##
## Fitting model 9 out of 42 (leaving out observation 10)

##
## Fitting model 10 out of 42 (leaving out observation 12)

##
## Fitting model 11 out of 42 (leaving out observation 14)

##
## Fitting model 12 out of 42 (leaving out observation 15)

##
## Fitting model 13 out of 42 (leaving out observation 16)

##
## Fitting model 14 out of 42 (leaving out observation 17)

##
## Fitting model 15 out of 42 (leaving out observation 18)

##
## Fitting model 16 out of 42 (leaving out observation 22)

##
## Fitting model 17 out of 42 (leaving out observation 23)

##
## Fitting model 18 out of 42 (leaving out observation 24)

##
## Fitting model 19 out of 42 (leaving out observation 25)

##
## Fitting model 20 out of 42 (leaving out observation 26)

##
## Fitting model 21 out of 42 (leaving out observation 27)

##
## Fitting model 22 out of 42 (leaving out observation 28)
```

```
##  
## Fitting model 23 out of 42 (leaving out observation 33)  
  
##  
## Fitting model 24 out of 42 (leaving out observation 34)  
  
##  
## Fitting model 25 out of 42 (leaving out observation 35)  
  
##  
## Fitting model 26 out of 42 (leaving out observation 39)  
  
##  
## Fitting model 27 out of 42 (leaving out observation 40)  
  
##  
## Fitting model 28 out of 42 (leaving out observation 41)  
  
##  
## Fitting model 29 out of 42 (leaving out observation 42)  
  
##  
## Fitting model 30 out of 42 (leaving out observation 47)  
  
##  
## Fitting model 31 out of 42 (leaving out observation 48)  
  
##  
## Fitting model 32 out of 42 (leaving out observation 49)  
  
##  
## Fitting model 33 out of 42 (leaving out observation 51)  
  
##  
## Fitting model 34 out of 42 (leaving out observation 52)  
  
##  
## Fitting model 35 out of 42 (leaving out observation 53)  
  
##  
## Fitting model 36 out of 42 (leaving out observation 55)  
  
##  
## Fitting model 37 out of 42 (leaving out observation 56)  
  
##  
## Fitting model 38 out of 42 (leaving out observation 57)  
  
##  
## Fitting model 39 out of 42 (leaving out observation 58)
```



```

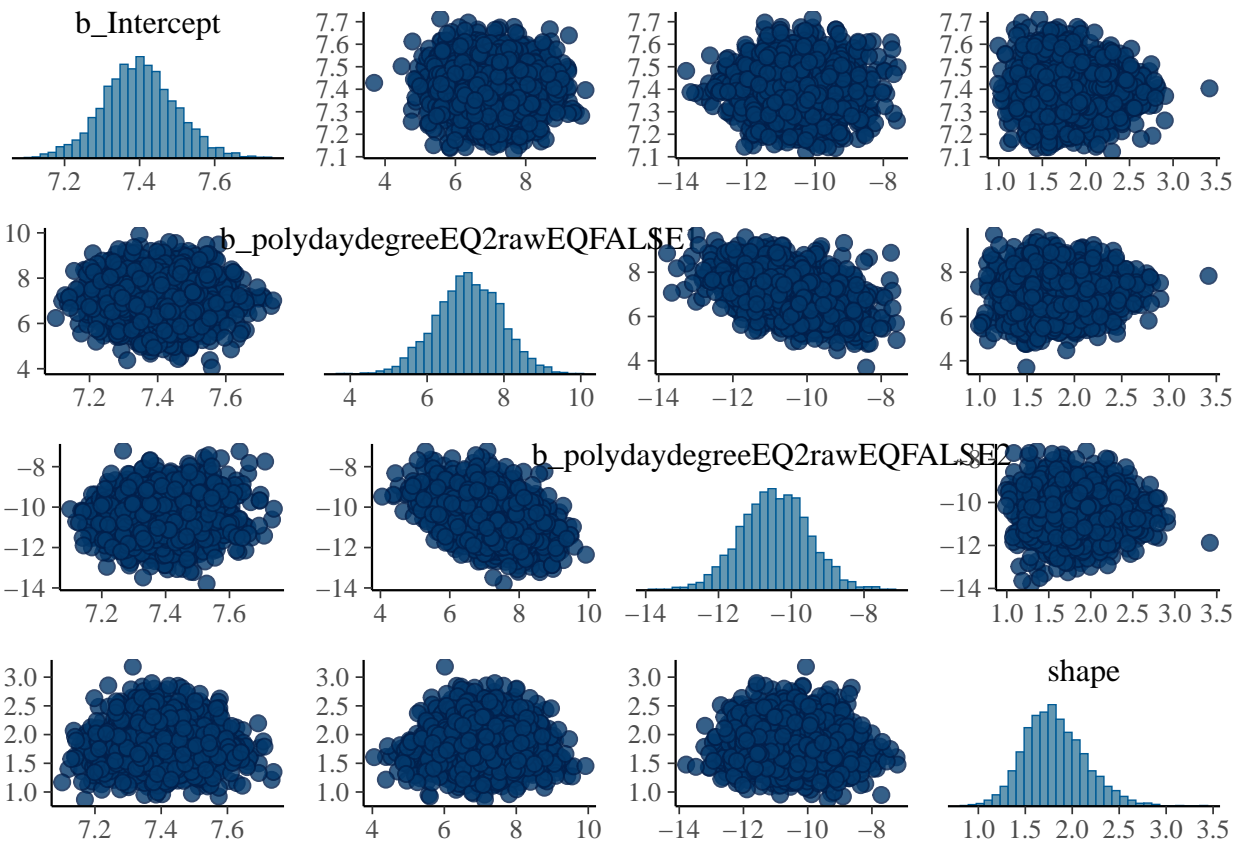
##
## Computed from 4000 by 61 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo   -510.9 12.4
## p_loo       4.9  1.0
## looic      1021.8 24.9
## -----
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Output of model 'post_4':
##
## Computed from 4000 by 61 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo -11219.8 1300.3
## p_loo      996.1 148.5
## looic     22439.6 2600.6
## -----
## Monte Carlo SE of elpd_loo is 2.8.
##
## Pareto k diagnostic values:
##           Count Pct.    Min. n_eff
## (-Inf, 0.5] (good)   55   90.2%     1
## (0.5, 0.7]  (ok)     6    9.8%    61
## (0.7, 1]    (bad)     0    0.0%   <NA>
## (1, Inf)    (very bad) 0    0.0%   <NA>
##
## All Pareto k estimates are ok (k < 0.7).
## See help('pareto-k-diagnostic') for details.
##
## Model comparisons:
##           elpd_diff se_diff
## post_3      0.0      0.0
## post_4 -10708.9 1302.0

```

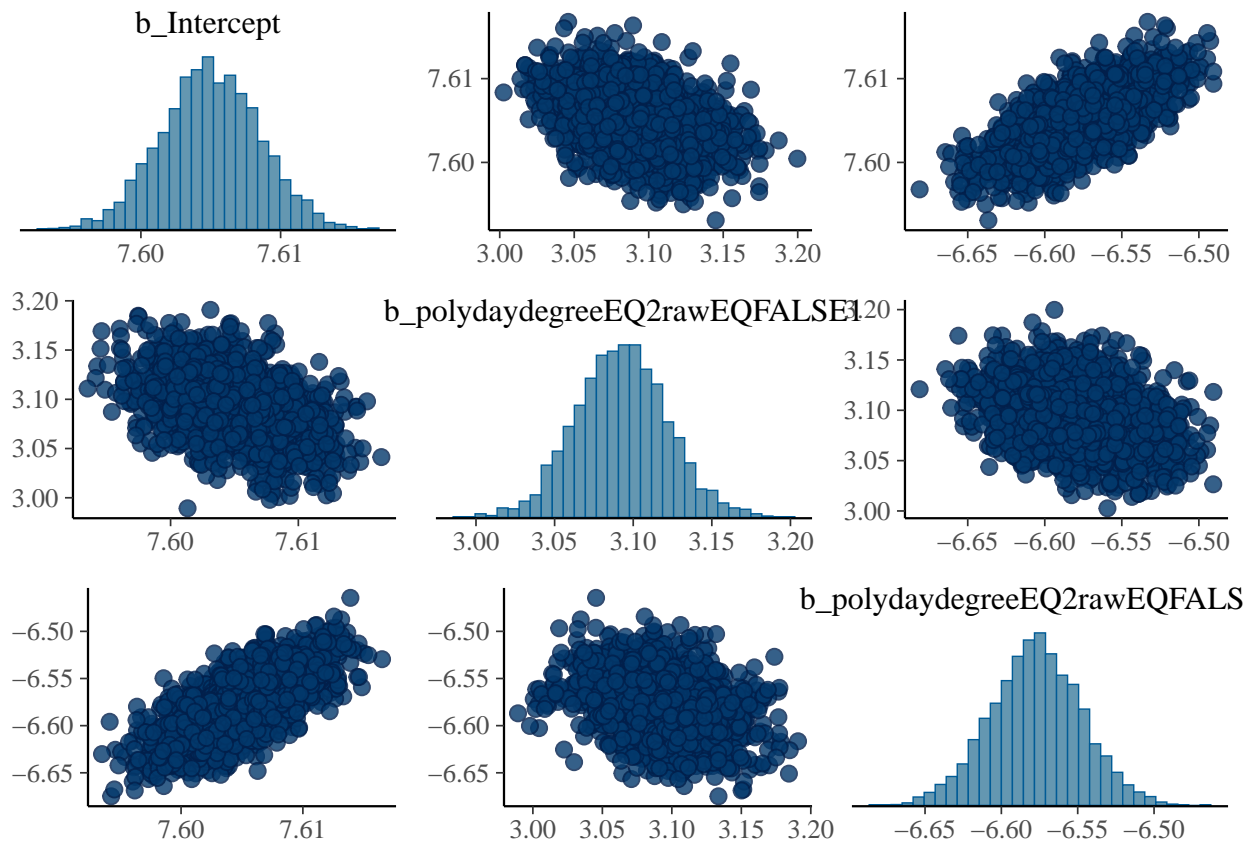
```

pairs(post_3)

```

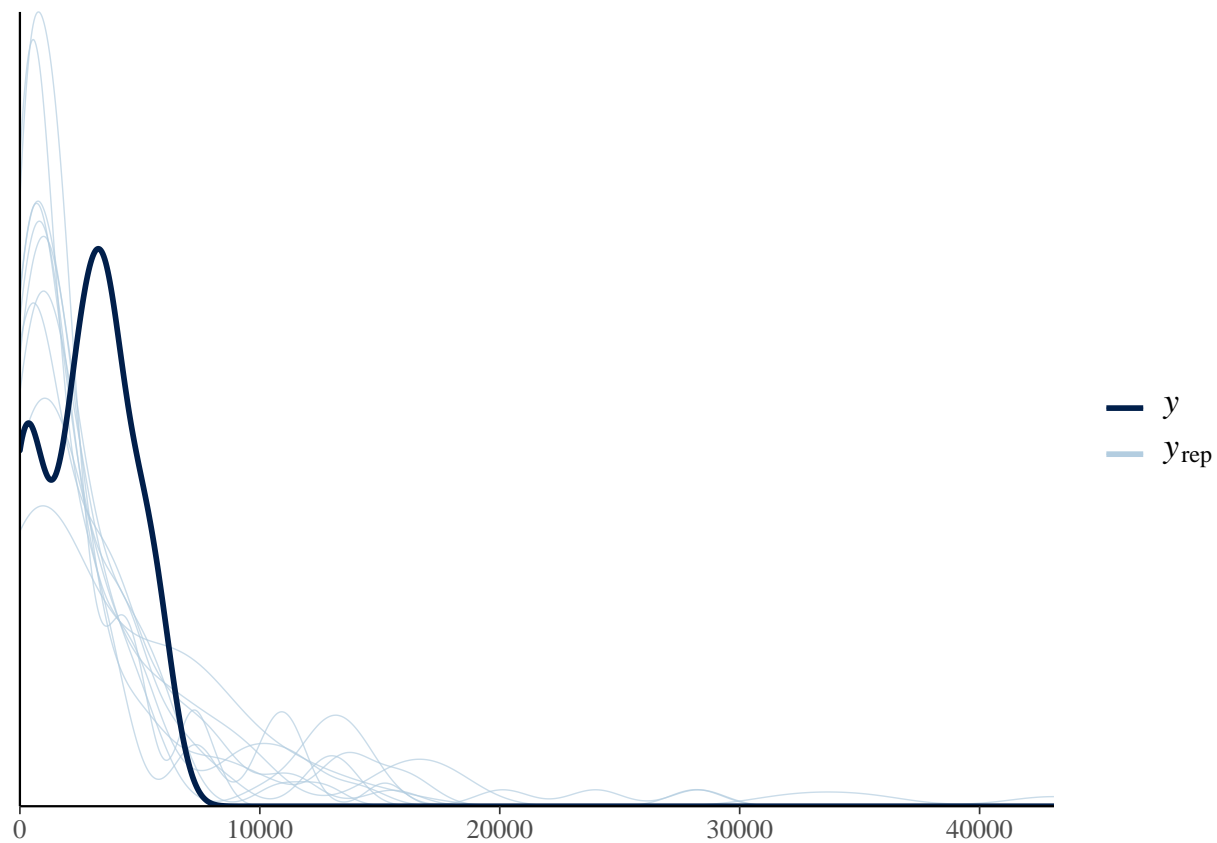


```
pairs(post_4)
```

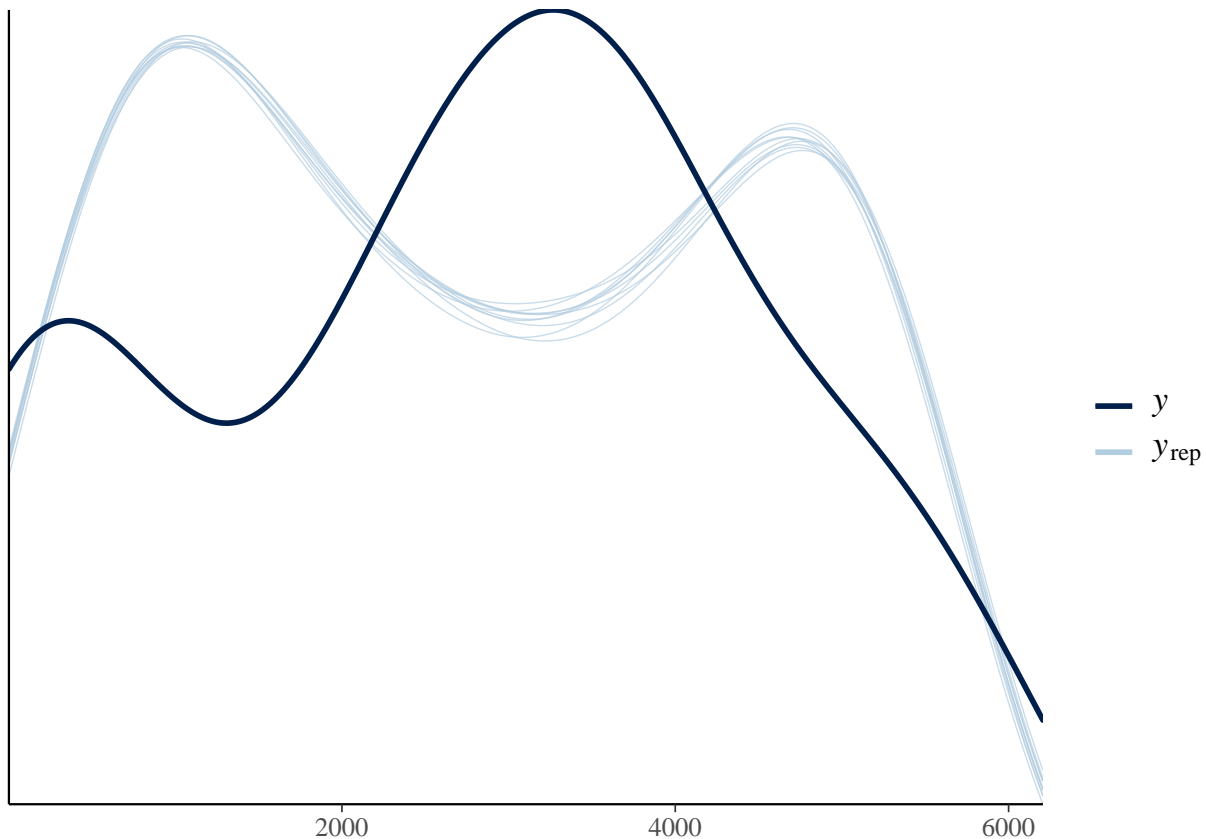
```
pp_check(post_3)
```

```
## Using 10 posterior samples for ppc type 'dens_overlay' by default.
```



```
pp_check(post_4)
```

```
## Using 10 posterior samples for ppc type 'dens_overlay' by default.
```



Based on the calculated ELPD difference of 10707.2 of the model comparison (which is multiple times higher than the standard error), we can conclude that the negative binomial model fits better than the poisson model. The pp_check shows the graph indicates that the negative binomial model fits better as well.

2.4 Posterior Prediction

```
extended<- as.data.frame(matrix(nrow=7,ncol=5))
colnames(extended)<-c('DATE_OF_INTEREST','CASE_COUNT','HOSPITALIZED_COUNT','DEATH_COUNT','day')
extended$day<- c(62:68)
NYC_new<- rbind(NYC,extended)
Pred<- posterior_predict(post_3,newdata=NYC_new)
Pred_result<- colMeans(Pred)
NYC_new$predicted_count<- Pred_result
```

Over the coming week, according to our model predictions, the new confirmed coronavirus cases are likely to slowly decay from 365 to 72. I would describe it as the explosion of the coronavirus is calming down. However, there are new cases that are arising every day, implying that there is still a significant amount of increase in new confirmed cases over the next week.