

HW 6

Nathan Hawkins

3/2/2021

```
library(brms)

crack <- read.table("C:/Users/nateh/Downloads/cracker.dat", header = TRUE)
```

1. Run 3 slope model in using brm

```
crack$tmtf <- as.factor(crack$tmt)
fit1 <- brm(formula = sales ~ -1 + tmtf + tmtf:presales, data = crack, family
= "gaussian", prior = c(set_prior("normal(0,100)", class = "b")),
  warmup = 1000, iter = 3000, chains = 4,
  control = list(adapt_delta = .98))
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
##
```

```
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
```

```
## Chain 1: Iteration:   300 / 3000 [ 10%] (Warmup)
```

```
## Chain 1: Iteration:   600 / 3000 [ 20%] (Warmup)
```

```
## Chain 1: Iteration:   900 / 3000 [ 30%] (Warmup)
```

```
## Chain 1: Iteration:  1001 / 3000 [ 33%] (Sampling)
```

```
## Chain 1: Iteration:  1300 / 3000 [ 43%] (Sampling)
```

```
## Chain 1: Iteration:  1600 / 3000 [ 53%] (Sampling)
```

```
## Chain 1: Iteration:  1900 / 3000 [ 63%] (Sampling)
```

```
## Chain 1: Iteration:  2200 / 3000 [ 73%] (Sampling)
```

```
## Chain 1: Iteration:  2500 / 3000 [ 83%] (Sampling)
```

```
## Chain 1: Iteration:  2800 / 3000 [ 93%] (Sampling)
```

```
## Chain 1: Iteration:  3000 / 3000 [100%] (Sampling)
```

```
## Chain 1:
```

```
## Chain 1: Elapsed Time: 0.496 seconds (Warm-up)
```

```
## Chain 1:           1.129 seconds (Sampling)
```

```
## Chain 1:           1.625 seconds (Total)
```

```
## Chain 1:
```

```
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 2: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 2: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 2: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 2: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 2: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 2: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 2: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 2: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 2: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 2: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 2: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 2:
## Chain 2: Elapsed Time: 0.644 seconds (Warm-up)
## Chain 2:                0.951 seconds (Sampling)
## Chain 2:                1.595 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 3: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 3: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 3: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 3: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 3: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 3: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 3: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 3: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 3: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 3: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 3: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 3:
## Chain 3: Elapsed Time: 0.597 seconds (Warm-up)
## Chain 3:                0.699 seconds (Sampling)
```

```

## Chain 3:                1.296 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 4: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 4: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 4: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 4: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 4: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 4: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 4: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 4: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 4: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 4: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 4: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 4:
## Chain 4: Elapsed Time: 0.53 seconds (Warm-up)
## Chain 4:                0.749 seconds (Sampling)
## Chain 4:                1.279 seconds (Total)
## Chain 4:

```

```
summary(fit1)
```

```

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: sales ~ -1 + tmtf + tmtf:presales
## Data: crack (Number of observations: 14)
## Samples: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup samples = 8000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## tmtf1           12.69      7.35   -2.36   27.31 1.00    3631    2974
## tmtf2            9.46      5.34   -1.21   19.82 1.00    4150    3087
## tmtf3            7.75      5.06   -2.33   17.65 1.00    3874    3835
## tmtf1:presales    1.10      0.31    0.47    1.75 1.00    3608    2942
## tmtf2:presales    1.01      0.20    0.62    1.42 1.00    4127    3306
## tmtf3:presales    0.78      0.20    0.39    1.17 1.00    3880    3865
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma           2.16      0.66    1.30    3.82 1.00    2270    2451

```

```
##
## 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).

library(shinystan)

#Launch_shinystan(fit1)
library(coda)
```

2. Look for differences in slopes

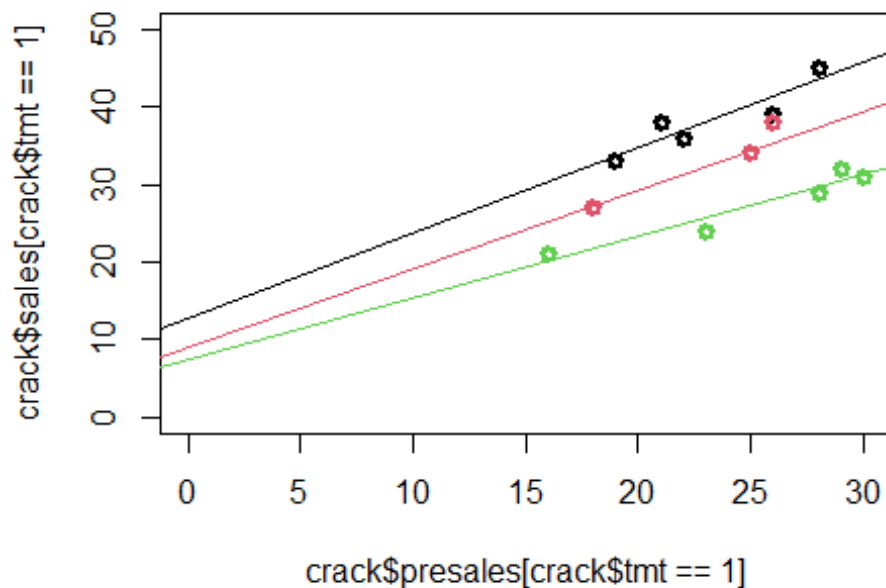
It doesn't appear that there is a significant difference in any of the slopes.

```
sims <- as.mcmc(fit1)
chains <- as.matrix(sims)
sims <- as.mcmc(chains)
colnames(chains)

## [1] "b_tmtf1"          "b_tmtf2"          "b_tmtf3"
## [5] "b_tmtf1:presales" "b_tmtf2:presales" "b_tmtf3:presales" "sigma"
## [9] "lp__"

slope1 = chains[,4]
slope2 = chains[,5]
slope3 = chains[,6]

# Plots look like the slopes are different
plot(crack$presales[crack$tmt == 1], crack$sales[crack$tmt ==1], col = 1,
xlim = c(0,30), ylim = c(0,50), lwd = 3)
points(crack$presales[crack$tmt == 2], crack$sales[crack$tmt ==2], col = 2,
lwd = 3)
points(crack$presales[crack$tmt == 3], crack$sales[crack$tmt ==3], col = 3,
lwd = 3)
abline(a = 12.71, b = 1.10, col = 1)
abline(a = 9.56, b = 1.01, col = 2)
abline(a = 7.51, b = 0.79, col = 3)
```



There doesn't seem to be a significant difference in the slopes

```
quantile(slope1- slope2, probs = c(.025, .975))
```

```
##          2.5%          97.5%
## -0.6663939   0.8192417
```

```
quantile(slope1- slope3, probs = c(.025, .975))
```

```
##          2.5%          97.5%
## -0.3957255   1.0595054
```

```
quantile(slope2- slope3, probs = c(.025, .975))
```

```
##          2.5%          97.5%
## -0.3269125   0.7837421
```

3 Three intercepts and one slope

```
fit2 <- brm(formula = sales ~ -1 + tmtf + presales, data = crack, family =
"gaussian", prior = c(set_prior("normal(0,100)", class = "b")),
  warmup = 1000, iter = 3000, chains = 4,
  control = list(adapt_delta = .98))
```

```
## Compiling Stan program...
```

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

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 1: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 1: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 1: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 1: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 1: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 1: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 1: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 1: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 1: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 1: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 1: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 1:
## Chain 1: Elapsed Time: 0.294 seconds (Warm-up)
## Chain 1:                0.45 seconds (Sampling)
## Chain 1:                0.744 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 2: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 2: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 2: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 2: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 2: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 2: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 2: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 2: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 2: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 2: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 2: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 2:
## Chain 2: Elapsed Time: 0.265 seconds (Warm-up)
## Chain 2:                0.362 seconds (Sampling)
```

```
## Chain 2:                0.627 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 3: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 3: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 3: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 3: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 3: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 3: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 3: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 3: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 3: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 3: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 3: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 3:
## Chain 3: Elapsed Time: 0.288 seconds (Warm-up)
## Chain 3:                0.358 seconds (Sampling)
## Chain 3:                0.646 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'e8a275f9b809a54de3d3e43c082a898f' 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 / 3000 [  0%] (Warmup)
## Chain 4: Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 4: Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 4: Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 4: Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 4: Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 4: Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 4: Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 4: Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 4: Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 4: Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 4: Iteration:  3000 / 3000 [100%] (Sampling)
## Chain 4:
```

```

## Chain 4: Elapsed Time: 0.321 seconds (Warm-up)
## Chain 4: 0.263 seconds (Sampling)
## Chain 4: 0.584 seconds (Total)
## Chain 4:

summary(fit2)

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: sales ~ -1 + tmtf + presales
## Data: crack (Number of observations: 14)
## Samples: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
## total post-warmup samples = 8000
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## tmtf1      16.60      2.93   10.53   22.38 1.00    1639    1575
## tmtf2      11.52      3.26    4.92   18.09 1.00    1621    1729
## tmtf3       3.97      3.14   -2.42   10.34 1.00    1626    1660
## presales    0.93      0.12    0.69    1.18 1.00    1546    1589
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      2.04      0.54    1.28    3.37 1.00    1899    2726
##
## 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).

sims2 <- as.mcmc(fit2)
chains2 <- as.matrix(sims2)
sims2 <- as.mcmc(chains2)

```

Loo 1 is a better fit. It has more “good” predictions than loo 2. The model with 3 slopes is the better model.

```

library(loo)

loo1 = loo(fit1, save_psis = TRUE, moment_match = TRUE)

## Warning: Some Pareto k diagnostic values are slightly high. See
## help('pareto-k-diagnostic') for details.

loo2 = loo(fit2, save_psis = TRUE, moment_match = TRUE)

## Warning: Some Pareto k diagnostic values are slightly high. See
## help('pareto-k-diagnostic') for details.

loo1

##
## Computed from 8000 by 14 log-likelihood matrix

```



```
##
##           Estimate SE
## elpd_loo    -33.6 1.5
## p_loo        5.1 0.7
## looic        67.3 2.9
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## Pareto k diagnostic values:
##           Count Pct.    Min. n_eff
## (-Inf, 0.5] (good)     8    57.1%   1289
## (0.5, 0.7] (ok)        6    42.9%    158
## (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.

loo2

##
## Computed from 8000 by 14 log-likelihood matrix
##
##           Estimate SE
## elpd_loo    -31.6 1.6
## p_loo        3.7 0.6
## looic        63.3 3.1
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## Pareto k diagnostic values:
##           Count Pct.    Min. n_eff
## (-Inf, 0.5] (good)    11    78.6%   1330
## (0.5, 0.7] (ok)        3    21.4%    630
## (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.
```