Lab 9

Roxanne Li

2024-03-17

```
library(ggplot2)
library(dplyr)
library(tidyverse)
library(rstan)
library(here)
library(bayesplot)
library(tidybayes)
library(loo)
```

Here is the lip cancer data that was used in the lecture.

- `aff.i` is proportion of male population working outside in each region
- `observe.i` is observed deaths in each region
- `expect.i` is expected deaths, based on region-specific age distribution and national-level age-specific mortality rates.

```
observe.i <- c(
  5,13,18,5,10,18,29,10,15,22,4,11,10,22,13,14,17,21,25,6,11,21,13,5,19,18,14,17,3,10,
  7,3,12,11,6,16,13,6,9,10,4,9,11,12,23,18,12,7,13,12,12,13,6,14,7,18,13,9,6,8,7,6,16,4,6,12,5,5,
  17,5,7,2,9,7,6,12,13,17,5,5,6,12,10,16,10,16,15,18,6,12,6,8,33,15,14,18,25,14,2,73,13,14,6,20,8,
  12,10,3,11,3,11,13,11,13,10,5,18,10,23,5,9,2,11,9,11,6,11,5,19,15,4,8,9,6,4,4,2,12,12,11,9,7,7,
  8,12,11,23,7,16,46,9,18,12,13,14,14,3,9,15,6,13,13,12,8,11,5,9,8,22,9,2,10,6,10,12,9,11,32,5,11,
  9,11,11,0,9,3,11,11,11,5,4,8,9,30,110)
expect.i <- c(
    6.17,8.44,7.23,5.62,4.18,29.35,11.79,12.35,7.28,9.40,3.77,3.41,8.70,9.57,8.18,4.35,
    4.91,10.66,16.99,2.94,3.07,5.50,6.47,4.85,9.85,6.95,5.74,5.70,2.22,3.46,4.40,4.05,5.74,6.36,5.13,
    16.99,6.19,5.56,11.69,4.69,6.25,10.84,8.40,13.19,9.25,16.98,8.39,2.86,9.70,12.12,12.94,9.77,
    10.34,5.09,3.29,17.19,5.42,11.39,8.33,4.97,7.14,6.74,17.01,5.80,4.84,12.00,4.50,4.39,16.35,6.02,
    6.42,5.26,4.59,11.86,4.05,5.48,13.13,8.72,2.87,2.13,4.48,5.85,6.67,6.11,5.78,12.31,10.56,10.23,
    2.52,6.22,14.29,5.71,37.93,7.81,9.86,11.61,18.52,12.28,5.41,61.96,8.55,12.07,4.29,19.42,8.25,
    12.90,4.76,5.56,11.11,4.76,10.48,13.13,12.94,14.61,9.26,6.94,16.82,33.49,20.91,5.32,6.77,8.70,
    12.94,16.07,8.87,7.79,14.60,5.10,24.42,17.78,4.04,7.84,9.89,8.45,5.06,4.49,6.25,9.16,12.37,8.40,
    9.57,5.83,9.21,9.64,9.09,12.94,17.42,10.29,7.14,92.50,14.29,15.61,6.00,8.55,15.22,18.42,5.77,
    18.37,13.16,7.69,14.61,15.85,12.77,7.41,14.86,6.94,5.66,9.88,102.16,7.63,5.13,7.58,8.00,12.82,
    18.75,12.33,5.88,64.64,8.62,12.09,11.11,14.10,10.48,7.00,10.23,6.82,15.71,9.65,8.59,8.33,6.06,
    12.31,8.91,50.10,288.00)

aff.i <- c(0.2415,0.2309,0.3999,0.2977,0.3264,0.3346,0.4150,0.4202,0.1023,0.1752,
        0.2548,0.3248,0.2287,0.2520,0.2058,0.2785,0.2528,0.1847,0.3736,0.2411,
        0.3700,0.2997,0.2883,0.2427,0.3782,0.1865,0.2633,0.2978,0.3541,0.4176,
        0.2910,0.3431,0.1168,0.2195,0.2911,0.4297,0.2119,0.2698,0.0874,0.3204,
```

```
        0.1839,0.1796,0.2471,0.2016,0.1560,0.3162,0.0732,0.1490,0.2283,0.1187,
        0.3500,0.2915,0.1339,0.0995,0.2355,0.2392,0.0877,0.3571,0.1014,0.0363,
        0.1665,0.1226,0.2186,0.1279,0.0842,0.0733,0.0377,0.2216,0.3062,0.0310,
        0.0755,0.0583,0.2546,0.2933,0.1682,0.2518,0.1971,0.1473,0.2311,0.2471,
        0.3063,0.1526,0.1487,0.3537,0.2753,0.0849,0.1013,0.1622,0.1267,0.2376,
        0.0737,0.2755,0.0152,0.1415,0.1344,0.1058,0.0545,0.1047,0.1335,0.3134,
        0.1326,0.1222,0.1992,0.0620,0.1313,0.0848,0.2687,0.1396,0.1234,0.0997,
        0.0694,0.1022,0.0779,0.0253,0.1012,0.0999,0.0828,0.2950,0.0778,0.1388,
        0.2449,0.0978,0.1144,0.1038,0.1613,0.1921,0.2714,0.1467,0.1783,0.1790,
        0.1482,0.1383,0.0805,0.0619,0.1934,0.1315,0.1050,0.0702,0.1002,0.1445,
        0.0353,0.0400,0.1385,0.0491,0.0520,0.0640,0.1017,0.0837,0.1462,0.0958,
        0.0745,0.2942,0.2278,0.1347,0.0907,0.1238,0.1773,0.0623,0.0742,0.1003,
        0.0590,0.0719,0.0652,0.1687,0.1199,0.1768,0.1638,0.1360,0.0832,0.2174,
        0.1662,0.2023,0.1319,0.0526,0.0287,0.0405,0.1616,0.0730,0.1005,0.0743,
        0.0577,0.0481,0.1002,0.0433,0.0838,0.1124,0.2265,0.0436,0.1402,0.0313,
        0.0359,0.0696,0.0618,0.0932,0.0097)
```

Make a dataset from the data given:

```
aff_i_centered <- aff.i - mean(aff.i)
lip <- data.frame(observe_i = observe.i,
                  expect_i = expect.i,
                  aff_i_c = aff_i_centered)
```

Question 1

Explain a bit more what the `expect.i` variable is. For example, if a particular area has an expected deaths of 16, what does this mean?

The `expect.i` is the prediction of a particular region's death given the region's age distribution. The prediction is obtained from a statistical model which uses age data to predict mortality rates across different regions in the nation. If a particular area has an expected deaths of 16, it means that based on the area's age distribution, the average deaths of this area is 16.

Question 2

Run four different models in Stan with three different set-ups for estimating θ_i, that is the relative risk of lip cancer in each region:

1. Intercept α_i is same in each region $= \alpha$.

```
stan_data1 <- list(N = nrow(lip),
                   aff_i_c = lip$aff_i_c,
                   observe_i = lip$observe_i,
                   expect_i = lip$expect_i
                  )

fit1 <- stan(file = 'code/models/lip_model1.stan',
             data = stan_data1,
             iter = 500,
             seed = 1234)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: 'Apple clang version 14.0.3 (clang-1403.0.22.14.1)'
## using SDK: 'MacOSX13.3.sdk'
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Framew
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Core
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Core
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/Core:96
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3.2e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.32 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 1: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 1: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 1: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 1: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 1: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 1: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 1: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 1: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 1: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 1: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 1: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.016 seconds (Warm-up)
## Chain 1:                0.008 seconds (Sampling)
## Chain 1:                0.024 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 7e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 2: Adjust your expectations accordingly!
```

```
## Chain 2:
## Chain 2:
## Chain 2: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 2: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 2: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 2: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 2: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 2: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 2: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 2: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 2: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 2: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 2: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 2: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.014 seconds (Warm-up)
## Chain 2:                0.009 seconds (Sampling)
## Chain 2:                0.023 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 7e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 3: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 3: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 3: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 3: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 3: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 3: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 3: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 3: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 3: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 3: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 3: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.014 seconds (Warm-up)
## Chain 3:                0.009 seconds (Sampling)
## Chain 3:                0.023 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 7e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 4: Iteration:  50 / 500 [ 10%]  (Warmup)
```

```
## Chain 4: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 4: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 4: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 4: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 4: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 4: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 4: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 4: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 4: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 4: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.015 seconds (Warm-up)
## Chain 4:                0.01 seconds (Sampling)
## Chain 4:                0.025 seconds (Total)
## Chain 4:
```

Print the estimates:

```
summary_fit1 <- summary(fit1)

# Extract estimates for alpha, beta
estimates <- summary_fit1$summary[c("alpha", "beta"), ]
print(estimates)
```

```
##               mean      se_mean         sd        2.5%         25%          50%
## alpha -0.009887678 0.0006555356 0.01912611 -0.04814999 -0.02202514 -0.009873148
## beta   2.420464211 0.0075741144 0.17668899  2.04882749  2.30460421  2.425966639
##               75%      97.5%    n_eff      Rhat
## alpha 0.003103827 0.02713516 851.2574 0.9999837
## beta  2.551998774 2.74027796 544.1959 1.0015729
```

2. Intercept α_i is different in each region and modeled separately.

```
stan_data2 <- list(N = nrow(lip),
                   aff_i_c = lip$aff_i_c,
                   observe_i = lip$observe_i,
                   expect_i = lip$expect_i
                  )

fit2 <- stan(file = 'code/models/lip_model2.stan',
             data = stan_data2,
             iter = 500,
             seed = 1234)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: 'Apple clang version 14.0.3 (clang-1403.0.22.14.1)'
## using SDK: 'MacOSX13.3.sdk'
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Framew
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
```

```
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Cor
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Cor
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/Core:96
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3.8e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.38 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 1: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 1: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 1: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 1: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 1: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 1: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 1: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 1: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 1: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 1: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 1: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.054 seconds (Warm-up)
## Chain 1:                0.043 seconds (Sampling)
## Chain 1:                0.097 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 9e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 2: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 2: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 2: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 2: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 2: Iteration: 250 / 500 [ 50%]  (Warmup)
```

```
## Chain 2: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 2: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 2: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 2: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 2: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 2: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.052 seconds (Warm-up)
## Chain 2:                0.043 seconds (Sampling)
## Chain 2:                0.095 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 9e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 3: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 3: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 3: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 3: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 3: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 3: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 3: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 3: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 3: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 3: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 3: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.049 seconds (Warm-up)
## Chain 3:                0.043 seconds (Sampling)
## Chain 3:                0.092 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 9e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 4: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 4: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 4: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 4: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 4: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 4: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 4: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 4: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 4: Iteration: 400 / 500 [ 80%]  (Sampling)
```

```
## Chain 4: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 4: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.057 seconds (Warm-up)
## Chain 4:                0.043 seconds (Sampling)
## Chain 4:                0.1 seconds (Total)
## Chain 4:
```

Print the estimates:

```
summary_fit2 <- summary(fit2)

# Extract estimates for alpha, beta
beta_estimate <- summary_fit2$summary[c("beta"), ]
alpha_i_estimates <- summary_fit2$summary[1:195, ]

# Print beta estimate
print(beta_estimate)
```

```
##         mean      se_mean           sd         2.5%          25%          50%
##   1.45359933   0.05195514   0.62137411   0.34934724   1.02640429   1.44338079
##          75%        97.5%        n_eff         Rhat
##   1.84265457   2.77277224 143.03729903   1.00954168
```

```
# Print the first 6 alpha estimates
print(head(alpha_i_estimates))
```

```
##                  mean     se_mean        sd        2.5%        25%        50%
## alpha_i[1] -0.3323638 0.009628493 0.4313670 -1.25931789 -0.6067789 -0.3080165
## alpha_i[2]  0.2845769 0.006701775 0.2782102 -0.27293519  0.1066799  0.2913914
## alpha_i[3]  0.5183827 0.012905032 0.2691130 -0.05312141  0.3404435  0.5359423
## alpha_i[4] -0.3198199 0.011717257 0.4064029 -1.19992965 -0.5745256 -0.2771357
## alpha_i[5]  0.5453578 0.009958614 0.3230166 -0.12915035  0.3292092  0.5507215
## alpha_i[6] -0.7161548 0.009334192 0.2380967 -1.15030145 -0.8783982 -0.7205239
##                   75%      97.5%     n_eff      Rhat
## alpha_i[1] -0.03693500  0.4633439 2007.1378 0.9970095
## alpha_i[2]  0.46527768  0.8115867 1723.3224 0.9968423
## alpha_i[3]  0.70730021  1.0270711  434.8618 0.9996371
## alpha_i[4] -0.03115945  0.3934131 1202.9891 0.9980521
## alpha_i[5]  0.75949762  1.1206057 1052.0876 0.9994352
## alpha_i[6] -0.54651730 -0.2516390  650.6587 0.9992266
```

3. Intercept α_i is different in each region and the intercept is modeled hierarchically

```
stan_data3 <- list(N = nrow(lip),
                   aff_i_c = lip$aff_i_c,
                   observe_i = lip$observe_i,
                   expect_i = lip$expect_i
                  )

fit3 <- stan(file = 'code/models/lip_model3.stan',
             data = stan_data3,
             iter = 500,
             seed = 1234)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: 'Apple clang version 14.0.3 (clang-1403.0.22.14.1)'
## using SDK: 'MacOSX13.3.sdk'
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Framew
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Core
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Core
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/Core:96
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 5.6e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.56 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 1: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 1: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 1: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 1: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 1: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 1: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 1: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 1: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 1: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 1: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 1: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.118 seconds (Warm-up)
## Chain 1:                0.067 seconds (Sampling)
## Chain 1:                0.185 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.6e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
## Chain 2: Adjust your expectations accordingly!
```

```
## Chain 2:
## Chain 2:
## Chain 2: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 2: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 2: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 2: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 2: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 2: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 2: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 2: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 2: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 2: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 2: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 2: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.117 seconds (Warm-up)
## Chain 2:                0.068 seconds (Sampling)
## Chain 2:                0.185 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.5e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 3: Iteration:  50 / 500 [ 10%]  (Warmup)
## Chain 3: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 3: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 3: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 3: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 3: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 3: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 3: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 3: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 3: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 3: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.114 seconds (Warm-up)
## Chain 3:                0.07 seconds (Sampling)
## Chain 3:                0.184 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:   1 / 500 [  0%]  (Warmup)
## Chain 4: Iteration:  50 / 500 [ 10%]  (Warmup)
```

```
## Chain 4: Iteration: 100 / 500 [ 20%]  (Warmup)
## Chain 4: Iteration: 150 / 500 [ 30%]  (Warmup)
## Chain 4: Iteration: 200 / 500 [ 40%]  (Warmup)
## Chain 4: Iteration: 250 / 500 [ 50%]  (Warmup)
## Chain 4: Iteration: 251 / 500 [ 50%]  (Sampling)
## Chain 4: Iteration: 300 / 500 [ 60%]  (Sampling)
## Chain 4: Iteration: 350 / 500 [ 70%]  (Sampling)
## Chain 4: Iteration: 400 / 500 [ 80%]  (Sampling)
## Chain 4: Iteration: 450 / 500 [ 90%]  (Sampling)
## Chain 4: Iteration: 500 / 500 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.125 seconds (Warm-up)
## Chain 4:                0.068 seconds (Sampling)
## Chain 4:                0.193 seconds (Total)
## Chain 4:
```

Print the estimates:

```
summary_fit3 <- summary(fit3)

# Extract estimates for alpha, beta, mu, sigma_mu
estimates <- summary_fit3$summary[c("beta", "mu", "sigma_mu"), ]
alpha_i_estimates <- summary_fit3$summary[3:197, ]

# Print estimates
print(estimates)
```

```
##                mean      se_mean          sd       2.5%        25%        50%
## beta     1.97237400 0.0119364370 0.32297242 1.30487900 1.76277881 1.95904064
## mu       0.08507193 0.0009649927 0.03451886 0.01935014 0.05997473 0.08440987
## sigma_mu 0.38695894 0.0012532105 0.03071642 0.33022164 0.36543784 0.38570608
##                75%     97.5%     n_eff      Rhat
## beta     2.1957741 2.5668197  732.1186 1.0007692
## mu       0.1092288 0.1544214 1279.5723 0.9985246
## sigma_mu 0.4068386 0.4474826  600.7492 1.0068994
```

```
# Print the first 6 alpha estimates
print(head(alpha_i_estimates))
```

```
##                   mean     se_mean        sd       2.5%        25%        50%
## alpha_i[1] -0.1303741 0.005239428 0.2605424 -0.6811382 -0.3025691 -0.1157886
## alpha_i[2]  0.2159121 0.005609777 0.2412127 -0.2579996  0.0610497  0.2257729
## alpha_i[3]  0.3278641 0.005208366 0.2238470 -0.1089172  0.1699953  0.3265033
## alpha_i[4] -0.1428303 0.006044368 0.2757279 -0.7276463 -0.3110593 -0.1388822
## alpha_i[5]  0.3375983 0.005690935 0.2491075 -0.1832512  0.1810608  0.3416583
## alpha_i[6] -0.6093793 0.004990923 0.1918578 -0.9982338 -0.7387010 -0.6046254
##                    75%      97.5%    n_eff      Rhat
## alpha_i[1]  0.03477060  0.3823261 2472.800 0.9975258
## alpha_i[2]  0.37023710  0.6767224 1848.882 0.9970753
## alpha_i[3]  0.48532334  0.7430057 1847.139 0.9983522
## alpha_i[4]  0.04459489  0.3882853 2080.940 0.9995189
## alpha_i[5]  0.50232164  0.8030162 1916.049 0.9988368
## alpha_i[6] -0.47758024 -0.2447936 1477.737 0.9997217
```

Question 3

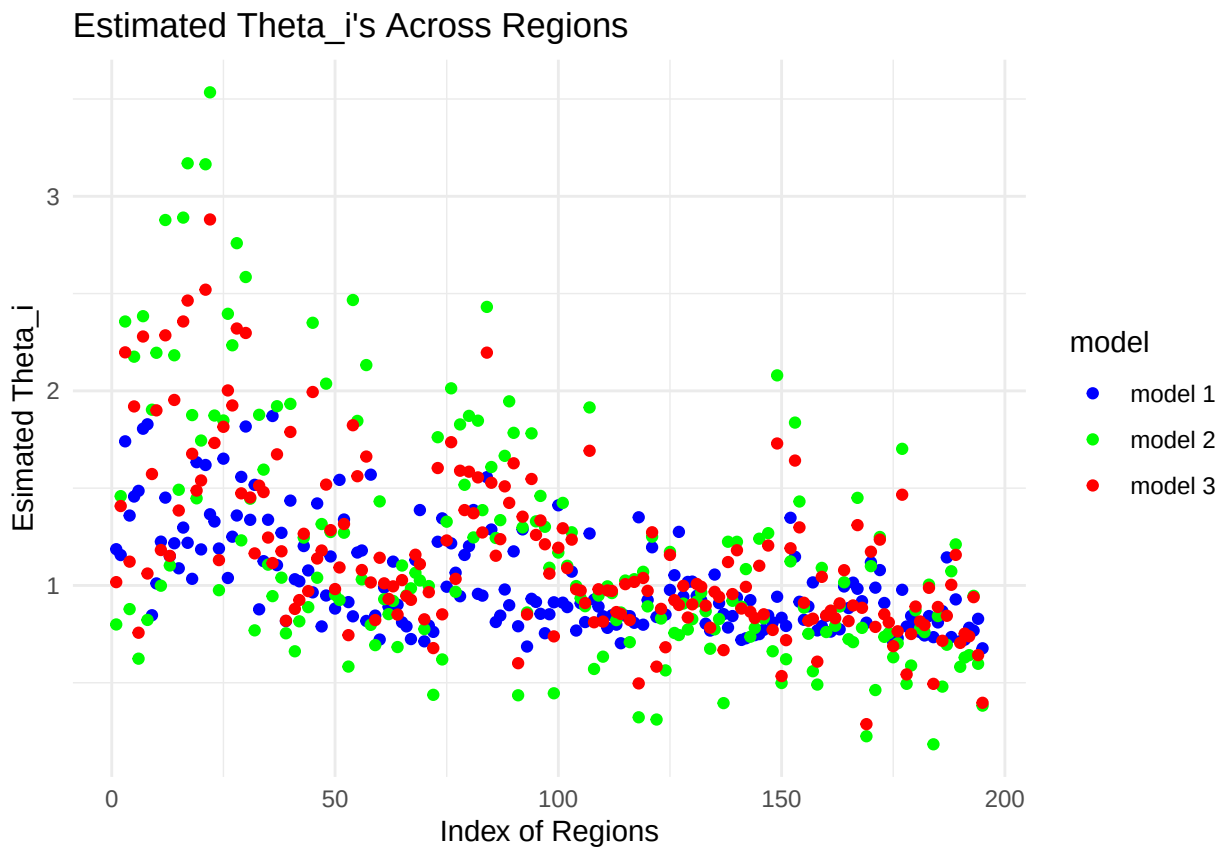
Make two plots (appropriately labeled and described) that illustrate the differences in estimated θ_i's across regions and the differences in θs across models.

Produce the estimated $theta_i$'s from 3 different models:

```
# model 1
alpha <- summary_fit1$summary[c("alpha"), "mean"]
beta <- summary_fit1$summary[c("beta"), "mean"]
theta_i_mod1 <- exp(alpha + beta * aff_i_centered)

# model 2
alpha_i <- summary_fit2$summary[1:195, "mean"]
beta <- summary_fit2$summary[c("beta"), "mean"]
theta_i_mod2 <- exp(alpha_i + beta * aff_i_centered)

# model 3
alpha_i <- summary_fit3$summary[3:197, "mean"]
beta <- summary_fit3$summary[c("beta"), "mean"]
theta_i_mod3 <- exp(alpha_i + beta * aff_i_centered)
```

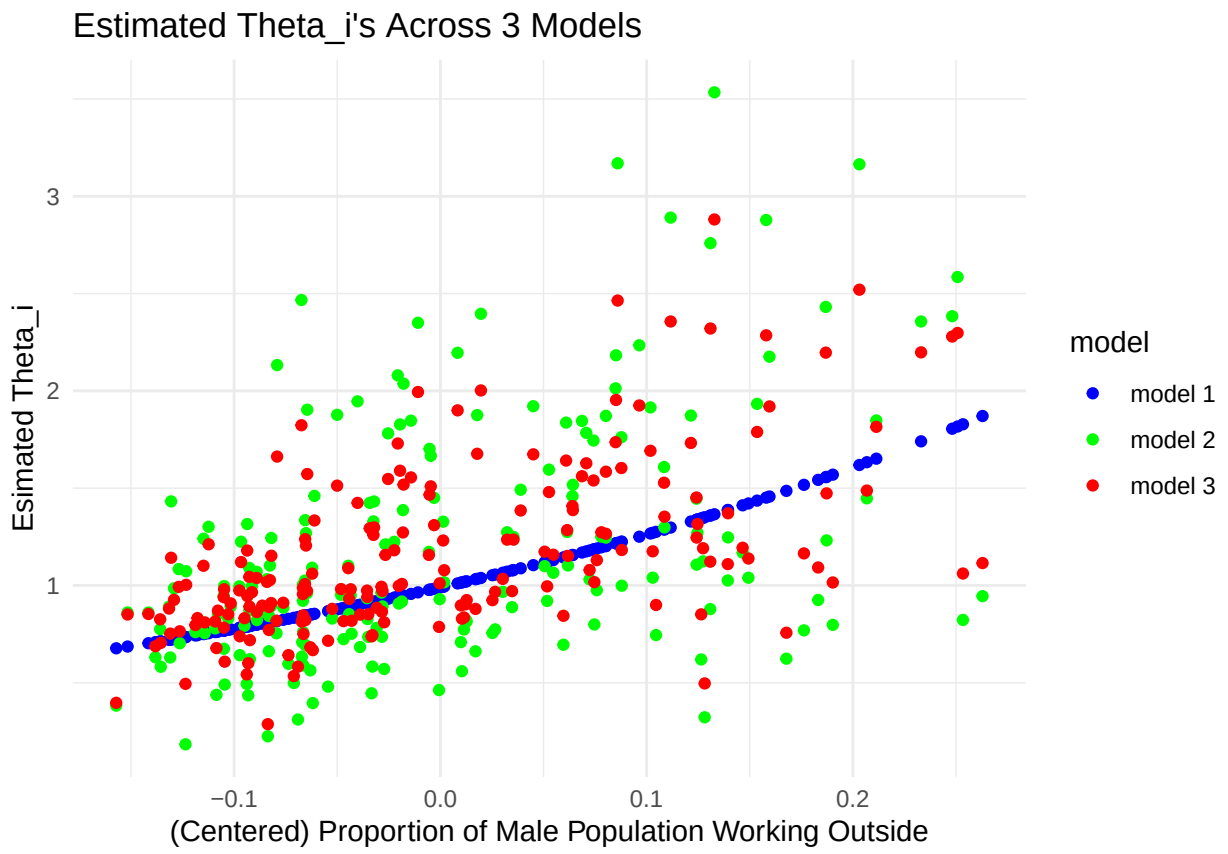
Plot the $theta_i$'s across regions:

```
theta_is <- c(theta_i_mod1, theta_i_mod2, theta_i_mod3)
models <- c(rep("model 1", 195), rep("model 2", 195), rep("model 3", 195))
regions <- rep(1:195, 3)

plot_data <- data.frame(theta_i = theta_is, model = models, region = regions)

ggplot(plot_data, aes(x = region, y = theta_i, color = model)) +
  geom_point() +
  scale_color_manual(values = c("blue", "green", "red")) +
  labs(title = "Estimated Theta_i's Across Regions",
       x = "Index of Regions",
       y = "Esimated Theta_i") +
  theme_minimal()
```



Plot the $theta_i$'s across models:

```
theta_is <- c(theta_i_mod1, theta_i_mod2, theta_i_mod3)
models <- c(rep("model 1", 195), rep("model 2", 195), rep("model 3", 195))
affs_is <- rep(aff_i_centered, 3)

plot_data <- data.frame(theta_i = theta_is, model = models, aff_i = affs_is)

ggplot(plot_data, aes(x = aff_i, y = theta_i, color = model)) +
  geom_point() +
  scale_color_manual(values = c("blue", "green", "red")) +
  labs(title = "Estimated Theta_i's Across 3 Models",
       x = "(Centered) Proportion of Male Population Working Outside",
       y = "Esimated Theta_i") +
  theme_minimal()
```



Question 4

Using tool of your choice, decide which model is the best, and justify your choice.

We extract the point-wise log likelihoods from the models:

```
log_lik_1 <- extract(fit1)[["log_lik"]]
loo1 <- loo(log_lik_1, save_psis = TRUE)

log_lik_2 <- extract(fit2)[["log_lik"]]
loo2 <- loo(log_lik_2, save_psis = TRUE)

log_lik_3 <- extract(fit3)[["log_lik"]]
loo3 <- loo(log_lik_3, save_psis = TRUE)
```

We then use loo_compare to compare the expected log predictive density of 3 models:

```
loo_compare(loo1,loo2,loo3)
```

```
##        elpd_diff se_diff
## model3    0.0       0.0
## model2   -9.4       7.9
## model1 -154.6      44.8
```

The result shows that model 3 has the highest expected log predictive density. Therefore, we choose model 3 as the best model.