

Week 9: Hierarchical GLM

17/03/24

Lip cancer

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

```
library(rstan)
library(here)
library(tidyverse)
library(loo)
library(posterior)
#install.packages('tidybayes')
library(tidybayes)
```

- `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,
  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,
  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,
  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,
  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,
  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,
  10.34,5.09,3.29,17.19,5.42,11.39,8.33,4.97,7.14,6.74,17.01,5.80,4.84,12.00,4.50,4.39,1
```

```

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.
2.52,6.22,14.29,5.71,37.93,7.81,9.86,11.61,18.52,12.28,5.41,61.96,8.55,12.07,4.29,19.4
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.
12.94,16.07,8.87,7.79,14.60,5.10,24.42,17.78,4.04,7.84,9.89,8.45,5.06,4.49,6.25,9.16,1
9.57,5.83,9.21,9.64,9.09,12.94,17.42,10.29,7.14,92.50,14.29,15.61,6.00,8.55,15.22,18.4
18.37,13.16,7.69,14.61,15.85,12.77,7.41,14.86,6.94,5.66,9.88,102.16,7.63,5.13,7.58,8.0
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.
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)

```

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?

From the description, the `expect.i` represents the expected number of deaths from lip cancer in each region, based on the region-specific age distribution and the national-level age-specific mortality rates, which is not a observed value. To estimate `expect.i`, the entire population was divided into ten-year age groups. For each group, we determined the rate of lip cancer deaths among men. This rate was then applied to the number of men in each age group for every region. By adding up these , we got `expect.i`, which reflects the expected number of deaths if the risk of lip cancer mortality was consistent throughout all regions for men. If a particular area has an expected deaths value of 16, this implies that based on the age

distribution of the population in that area and national lip cancer death rates for those ages groups, we expect on average 16 deaths from lip cancer in that region in the given time if it followed the national average risk.

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 $= \alpha$
2. Intercept α_i is different in each region and modeled separately
3. Intercept α_i is different in each region and the intercept is modeled hierarchically

Note in all three cases, use the proportion of male population working outside in each region as a covariate.

```
data <- list(N = length(observe.i),
            obs = observe.i,
            exps= expect.i,
            aff_i= aff.i-mean(aff.i))

mod1 <- stan(data=data,file = here("~/Documents/mod0.stan"),
            iter = 1000,
            chain=4,
            seed = 100)
```

Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c

using C compiler: 'Apple clang version 14.0.0 (clang-1400.0.29.202)'

using SDK: 'MacOSX13.1.sdk'

clang -arch x86_64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Libra

In file included from <built-in>:1:

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/

/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen

namespace Eigen {

~

/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen

namespace Eigen {

~

;

In file included from <built-in>:1:

```

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen
#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 8.8e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.88 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 1000 [0%] (Warmup)

Chain 1: Iteration: 100 / 1000 [10%] (Warmup)

Chain 1: Iteration: 200 / 1000 [20%] (Warmup)

Chain 1: Iteration: 300 / 1000 [30%] (Warmup)

Chain 1: Iteration: 400 / 1000 [40%] (Warmup)

Chain 1: Iteration: 500 / 1000 [50%] (Warmup)

Chain 1: Iteration: 501 / 1000 [50%] (Sampling)

Chain 1: Iteration: 600 / 1000 [60%] (Sampling)

Chain 1: Iteration: 700 / 1000 [70%] (Sampling)

Chain 1: Iteration: 800 / 1000 [80%] (Sampling)

Chain 1: Iteration: 900 / 1000 [90%] (Sampling)

Chain 1: Iteration: 1000 / 1000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 0.067 seconds (Warm-up)

Chain 1: 0.057 seconds (Sampling)

Chain 1: 0.124 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 2.5e-05 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.25 seconds.

Chain 2: Adjust your expectations accordingly!

Chain 2:

Chain 2:

Chain 2: Iteration: 1 / 1000 [0%] (Warmup)

Chain 2: Iteration: 100 / 1000 [10%] (Warmup)

Chain 2: Iteration: 200 / 1000 [20%] (Warmup)

```

Chain 2: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 2: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 2: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 2: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 2: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 2: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 2: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 2: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 2: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.07 seconds (Warm-up)
Chain 2:           0.065 seconds (Sampling)
Chain 2:           0.135 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 3.7e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.37 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:   1 / 1000 [  0%] (Warmup)
Chain 3: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 3: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 3: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 3: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 3: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 3: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 3: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 3: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 3: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 3: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 3: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 0.071 seconds (Warm-up)
Chain 3:           0.067 seconds (Sampling)
Chain 3:           0.138 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 2.5e-05 seconds

```

Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.25 seconds.
Chain 4: Adjust your expectations accordingly!

Chain 4:

Chain 4:

Chain 4: Iteration: 1 / 1000 [0%] (Warmup)
Chain 4: Iteration: 100 / 1000 [10%] (Warmup)
Chain 4: Iteration: 200 / 1000 [20%] (Warmup)
Chain 4: Iteration: 300 / 1000 [30%] (Warmup)
Chain 4: Iteration: 400 / 1000 [40%] (Warmup)
Chain 4: Iteration: 500 / 1000 [50%] (Warmup)
Chain 4: Iteration: 501 / 1000 [50%] (Sampling)
Chain 4: Iteration: 600 / 1000 [60%] (Sampling)
Chain 4: Iteration: 700 / 1000 [70%] (Sampling)
Chain 4: Iteration: 800 / 1000 [80%] (Sampling)
Chain 4: Iteration: 900 / 1000 [90%] (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%] (Sampling)

Chain 4:

Chain 4: Elapsed Time: 0.07 seconds (Warm-up)
Chain 4: 0.06 seconds (Sampling)
Chain 4: 0.13 seconds (Total)
Chain 4:

```
mod2 <- stan(data=data,file = here("~/Documents/mod00.stan"),
             iter = 1000,
             chain=4,
             seed = 100)
```

Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c

using C compiler: 'Apple clang version 14.0.0 (clang-1400.0.29.202)'

using SDK: 'MacOSX13.1.sdk'

clang -arch x86_64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Libr

In file included from <built-in>:1:

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/

In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/

/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen

namespace Eigen {

~

/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen

namespace Eigen {

~

;

```

In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen
#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 0.000144 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.44 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 1000 [  0%] (Warmup)
Chain 1: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 1: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 1: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 1: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 1: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 1: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 1: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 1: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 1: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 1: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 1: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 0.326 seconds (Warm-up)
Chain 1:                0.249 seconds (Sampling)
Chain 1:                0.575 seconds (Total)
Chain 1:

```

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 3.5e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.35 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 1000 [  0%] (Warmup)
Chain 2: Iteration: 100 / 1000 [ 10%] (Warmup)

```

```

Chain 2: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 2: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 2: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 2: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 2: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 2: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 2: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 2: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 2: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 2: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.427 seconds (Warm-up)
Chain 2:           0.468 seconds (Sampling)
Chain 2:           0.895 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 3.4e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.34 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:   1 / 1000 [  0%] (Warmup)
Chain 3: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 3: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 3: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 3: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 3: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 3: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 3: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 3: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 3: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 3: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 3: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 0.368 seconds (Warm-up)
Chain 3:           0.334 seconds (Sampling)
Chain 3:           0.702 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).

```

Chain 4:

```



```

Chain 4: Gradient evaluation took 3.4e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.34 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:   1 / 1000 [  0%] (Warmup)
Chain 4: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 4: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 4: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 4: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 4: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 4: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 4: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 4: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 4: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 4: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 0.382 seconds (Warm-up)
Chain 4:                0.247 seconds (Sampling)
Chain 4:                0.629 seconds (Total)
Chain 4:

```

```

mod3 <- stan(data=data,file = here("~/Documents/mod000.stan"),
             iter = 1000,
             chain=4,
             seed = 100)

```

```

Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 14.0.0 (clang-1400.0.29.202)'
using SDK: 'MacOSX13.1.sdk'
clang -arch x86_64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen
namespace Eigen {
~
/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen
namespace Eigen {
~

```

```

;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
In file included from /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/
/Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library/RcppEigen/include/Eigen
#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 0.0001 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:   1 / 1000 [  0%] (Warmup)
Chain 1: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 1: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 1: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 1: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 1: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 1: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 1: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 1: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 1: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 1: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 1: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 0.34 seconds (Warm-up)
Chain 1:                0.259 seconds (Sampling)
Chain 1:                0.599 seconds (Total)
Chain 1:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).

```

Chain 2:
Chain 2: Gradient evaluation took 3.2e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.32 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:   1 / 1000 [  0%] (Warmup)

```

```

Chain 2: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 2: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 2: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 2: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 2: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 2: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 2: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 2: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 2: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 2: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 2: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.352 seconds (Warm-up)
Chain 2:           0.259 seconds (Sampling)
Chain 2:           0.611 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 3.2e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.32 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:   1 / 1000 [  0%] (Warmup)
Chain 3: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 3: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 3: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 3: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 3: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 3: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 3: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 3: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 3: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 3: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 3: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 0.376 seconds (Warm-up)
Chain 3:           0.259 seconds (Sampling)
Chain 3:           0.635 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).

```
Chain 4:
Chain 4: Gradient evaluation took 3.2e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.32 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration: 1 / 1000 [ 0%] (Warmup)
Chain 4: Iteration: 100 / 1000 [ 10%] (Warmup)
Chain 4: Iteration: 200 / 1000 [ 20%] (Warmup)
Chain 4: Iteration: 300 / 1000 [ 30%] (Warmup)
Chain 4: Iteration: 400 / 1000 [ 40%] (Warmup)
Chain 4: Iteration: 500 / 1000 [ 50%] (Warmup)
Chain 4: Iteration: 501 / 1000 [ 50%] (Sampling)
Chain 4: Iteration: 600 / 1000 [ 60%] (Sampling)
Chain 4: Iteration: 700 / 1000 [ 70%] (Sampling)
Chain 4: Iteration: 800 / 1000 [ 80%] (Sampling)
Chain 4: Iteration: 900 / 1000 [ 90%] (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 0.351 seconds (Warm-up)
Chain 4: 0.261 seconds (Sampling)
Chain 4: 0.612 seconds (Total)
Chain 4:
```

```
model1<-summary(mod1)
estimates <- model1$summary[c("beta", "alpha"), ]
estimates
```

	mean	se_mean	sd	2.5%	25%	50%
beta	2.424154201	0.0047179712	0.17143264	2.08324033	2.31099829	2.424352460
alpha	-0.008563316	0.0004420321	0.02055921	-0.04966362	-0.02258324	-0.008119303
	75%	97.5%	n_eff	Rhat		
beta	2.535899994	2.76198773	1320.312	1.0006412		
alpha	0.005338494	0.03079667	2163.243	0.9993581		

```
model2<-summary(mod2)
print(model2$summary[1:5, ])
```

	mean	se_mean	sd	2.5%	25%	50%
alpha[1]	-0.3382188	0.007625726	0.4080030	-1.18918114	-0.5980773	-0.3209879

```
alpha[2] 0.2794708 0.006133153 0.2702104 -0.26248677 0.1054159 0.2883014
alpha[3] 0.5075255 0.008941756 0.2719398 -0.07643664 0.3273175 0.5175620
alpha[4] -0.3344012 0.008064290 0.4021441 -1.14724780 -0.5854268 -0.3211764
alpha[5] 0.5322940 0.007058258 0.3351994 -0.15311426 0.3150246 0.5477097
      75%      97.5%      n_eff      Rhat
alpha[1] -0.04810027 0.3981653 2862.6245 0.9986805
alpha[2] 0.46710266 0.7682860 1941.0494 0.9989010
alpha[3] 0.68878252 1.0390609 924.9107 0.9995309
alpha[4] -0.05761974 0.3900022 2486.7439 1.0001007
alpha[5] 0.76084292 1.1881688 2255.3374 0.9995245
```

```
model3<-summary(mod3)
print(model3$summary[1:5, ])
```

```
      mean      se_mean      sd      2.5%      25%      50%
mu      0.08601303 0.0007246604 0.03709038 0.0129870 0.06215481 0.08672584
sigma    0.38764381 0.0010358622 0.03113813 0.3246353 0.36691721 0.38753402
alpha[1] -0.14421690 0.0041977511 0.27688361 -0.7095433 -0.33512742 -0.13148554
alpha[2] 0.20153958 0.0037346764 0.23668500 -0.2820922 0.04366859 0.20561439
alpha[3] 0.29233700 0.0033693317 0.21325001 -0.1308259 0.14397420 0.29949173
      75%      97.5%      n_eff      Rhat
mu      0.10950425 0.1578755 2619.7114 1.0011948
sigma    0.40829165 0.4494601 903.6102 1.0038974
alpha[1] 0.05154495 0.3589088 4350.7200 0.9993408
alpha[2] 0.35930374 0.6647686 4016.3865 0.9990445
alpha[3] 0.44053670 0.6869795 4005.8120 0.9990808
```

Question 3

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

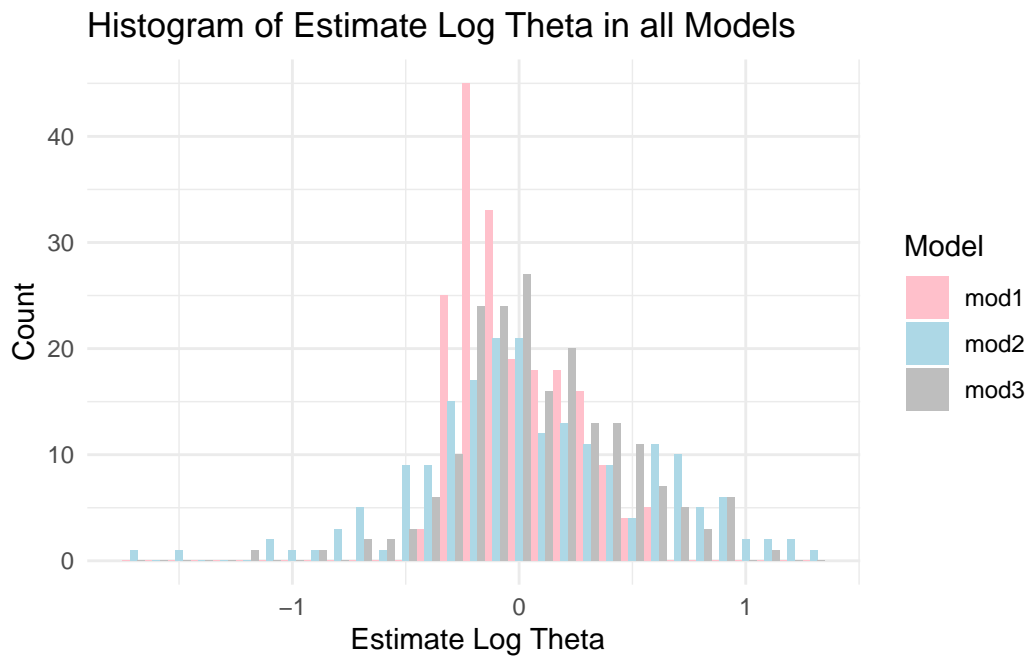
```
resfunction <- function(model, median_name, lower_name, upper_name) {
  model%>%
    gather_draws(theta[i]) %>%
    tidybayes::median_qi() %>%
    rename_with(~c(median_name, lower_name, upper_name), .cols = c(".value", ".lower", ".upper"))
    select(i, starts_with("median"), starts_with("lower"), starts_with("upper"))
}
res_mod1 <- resfunction(mod1, "median_mod1", "lower_mod1", "upper_mod1")
```

```

res_mod2 <- resfunction(mod2, "median_mod2", "lower_mod2", "upper_mod2")
res_mod3 <- resfunction(mod3, "median_mod3", "lower_mod3", "upper_mod3")
res1 <- list(res_mod1, res_mod2, res_mod3) %>%
  reduce(left_join, by = "i")

long_data <- res1 %>%
  pivot_longer(cols = starts_with('median'),
               names_to = 'model',
               values_to = 'theta') %>%
  mutate(model = str_remove(model, 'median_'))
ggplot(long_data, aes(x = theta, fill = model)) +
  geom_histogram(position = 'dodge', binwidth = 0.1) +
  scale_fill_manual(values=c("mod1" = "pink", "mod2" = "lightblue", "mod3" = "gray")) +
  labs(title = "Histogram of Estimate Log Theta in all Models ", x = " Estimate Log Theta",
       theme_minimal()

```



```

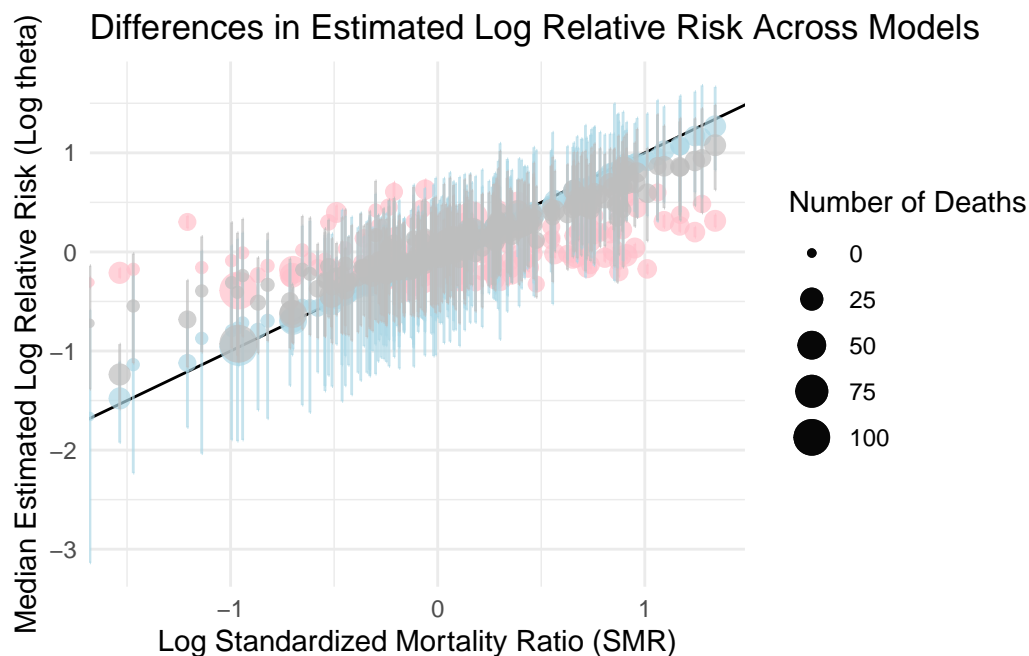
res1 %>%
  mutate(
    deaths = observe.i,
    log_smr = log(observe.i / expect.i)

```

```

) %>%
ggplot(aes(log_smr, median_mod1, color = "pink")) +
geom_point(aes(size = deaths), alpha = 0.7) +
geom_errorbar(aes(ymin = lower_mod1, ymax = upper_mod1, color = "pink"), alpha = 0.7) +
geom_abline(slope = 1, intercept = 0) +
geom_point(aes(log_smr, median_mod2, color = "lightblue", size = deaths), alpha = 0.7) +
geom_errorbar(aes(ymin = lower_mod2, ymax = upper_mod2, color = "lightblue"), alpha = 0.7) +
geom_point(aes(log_smr, median_mod3, color = "gray", size = deaths), alpha = 0.7) +
geom_errorbar(aes(ymin = lower_mod3, ymax = upper_mod3, color = "gray"), alpha = 0.7) +
labs(
  title = "Differences in Estimated Log Relative Risk Across Models",
  x = "Log Standardized Mortality Ratio (SMR)",
  y = "Median Estimated Log Relative Risk (Log theta)",
  size = "Number of Deaths"
) +
scale_color_identity() +
theme_minimal()

```



Question 4

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

```

loglik1 <- rstan::extract(mod1)[["log_lik"]]
loo1 <- loo(loglik1, save_psis = TRUE)
loglik2 <- rstan::extract(mod2)[["log_lik"]]
loo2 <- loo(loglik2, save_psis = TRUE)
loglik3 <- rstan::extract(mod3)[["log_lik"]]
loo3 <- loo(loglik3, save_psis = TRUE)
loo_compare(loo1,loo2,loo3)

```

	elpd_diff	se_diff
model3	0.0	0.0
model2	-17.7	8.1
model1	-151.5	44.8

To compare models, I use the loo package(loo_compare), the elpd_diff column shows the difference in expected log pointwise predictive density (ELPD) for each model compared to the top model (model 3 in this case), we know the higher ELPD value indicates a model has better predictive performance and fit the actual data well, compare to model 3, the elpd value of model 2 and model 1 is lower than model 3 (since elpd_diff is -17.7 and -151.3 respectively), so model 3 is the best model.