

# Week 9: Hierarchical GLM

Fred

15/03/23

## Lip cancer

Here is the lip cancer data given to you in terribly unreproducible and error-prone format.

- `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 6, what does this mean?

### Answer:

Expected deaths is the implied number of lip cancer deaths for a particular region given that region's age structure and the national level age-specific mortality rates for lip cancer. For example, an expected number of deaths of 6 would mean that for particular region, we would expect 6 lip cancer deaths if this region were to experience the same age specific mortality rates as at the national level.

## Question 2

Run three different models in Stan with three different set-up's for estimating  $\theta_i$ , that is the relative risk of lip cancer in each region:

1. Intercept  $\alpha_i$  is same in each region =  $\alpha$  (with covariate)
2.  $\alpha_i$  is different in each region and modeled separately (with covariate)

3.  $\alpha_i$  is different in each region and the intercept is modeled hierarchically (with covariate)

## Answer

$$y_i | \theta_i \sim \text{Poisson}(\theta_i \cdot e_i)$$

Look at three models for  $\log \theta_i$

Model 1:

$$\log \theta_i = \alpha + \beta x_i$$

Model 2:

$$\log \theta_i = \alpha_i + \beta x_i$$

Model 3:

$$\log \theta_i = \alpha_i + \beta x_i$$

with

$$\alpha_i \sim N(\mu, \sigma^2)$$

## Model 1:

```
library(tidyverse)
library(rstan)
library(tidybayes)

stan_data <- list(y = observe.i,
                 log_e = log(expect.i),
                 N = length(observe.i),
                 x = aff.i - mean(aff.i))

mod1 <- stan(data = stan_data, file = 'modell1.stan')
```

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

clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S

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

In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S

In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R

In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R

```

/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
^

/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
^
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R:
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
#include <complex>
^~~~~~
3 errors generated.
make: *** [foo.o] Error 1

```

SAMPLING FOR MODEL 'model1' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 3.1e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.31 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: 0.042314 seconds (Warm-up)

Chain 1: 0.035222 seconds (Sampling)

Chain 1: 0.077536 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'model1' 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 / 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: 0.039396 seconds (Warm-up)  
Chain 2: 0.036004 seconds (Sampling)  
Chain 2: 0.0754 seconds (Total)  
Chain 2:

SAMPLING FOR MODEL 'model1' NOW (CHAIN 3).

Chain 3:  
Chain 3: Gradient evaluation took 1e-05 seconds  
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.1 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: 0.039404 seconds (Warm-up)
Chain 3:           0.038217 seconds (Sampling)
Chain 3:           0.077621 seconds (Total)
Chain 3:
```

SAMPLING FOR MODEL 'model1' NOW (CHAIN 4).

```
Chain 4:
Chain 4: Gradient evaluation took 1.1e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 0.041231 seconds (Warm-up)
Chain 4:           0.032489 seconds (Sampling)
Chain 4:           0.07372 seconds (Total)
Chain 4:
```

## Model 2:

```
library(tidyverse)
library(rstan)
library(tidybayes)

stan_data <- list(y = observe.i,
                  log_e = log(expect.i),
                  N = length(observe.i),
                  x = aff.i - mean(aff.i))
```

```
mod2 <- stan(data = stan_data, file = 'model2.stan')
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
~
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
~
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
#include <complex>
~~~~~~
3 errors generated.
make: *** [foo.o] Error 1
```

```
SAMPLING FOR MODEL 'model2' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 2.8e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.28 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: 0.191548 seconds (Warm-up)  
Chain 1: 0.181881 seconds (Sampling)  
Chain 1: 0.373429 seconds (Total)  
Chain 1:

SAMPLING FOR MODEL 'model2' NOW (CHAIN 2).

Chain 2:  
Chain 2: Gradient evaluation took 1.1e-05 seconds  
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.  
Chain 2: Adjust your expectations accordingly!  
Chain 2:  
Chain 2:  
Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
Chain 2:  
Chain 2: Elapsed Time: 0.195079 seconds (Warm-up)  
Chain 2: 0.184297 seconds (Sampling)  
Chain 2: 0.379376 seconds (Total)  
Chain 2:

SAMPLING FOR MODEL 'model2' NOW (CHAIN 3).

Chain 3:  
Chain 3: Gradient evaluation took 1.1e-05 seconds  
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 0.196424 seconds (Warm-up)
Chain 3: 0.183607 seconds (Sampling)
Chain 3: 0.380031 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'model2' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 1.1e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 0.1948 seconds (Warm-up)
Chain 4: 0.189488 seconds (Sampling)
Chain 4: 0.384288 seconds (Total)
Chain 4:

```

### Model 3:

```
library(tidyverse)
library(rstan)
library(tidybayes)

stan_data <- list(y = observe.i,
                 log_e = log(expect.i),
                 N = length(observe.i),
                 x = aff.i - mean(aff.i))

mod3 <- stan(data = stan_data, file = 'model3.stan')
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
~
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
~
~
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
#include <complex>
~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'model3' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3.4e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.34 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: 0.211887 seconds (Warm-up)
Chain 1:                0.190782 seconds (Sampling)
Chain 1:                0.402669 seconds (Total)
Chain 1:

```

SAMPLING FOR MODEL 'model3' NOW (CHAIN 2).

```

Chain 2:
Chain 2: Gradient evaluation took 1.3e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.13 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: 0.21772 seconds (Warm-up)
Chain 2:                0.197975 seconds (Sampling)
Chain 2:                0.415695 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'model3' NOW (CHAIN 3).

Chain 3:

Chain 3: Gradient evaluation took 1.3e-05 seconds

Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.13 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: 0.215159 seconds (Warm-up)

Chain 3: 0.193594 seconds (Sampling)

Chain 3: 0.408753 seconds (Total)

Chain 3:

SAMPLING FOR MODEL 'model3' NOW (CHAIN 4).

Chain 4:

Chain 4: Gradient evaluation took 1.7e-05 seconds

Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.17 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: 0.212929 seconds (Warm-up)
Chain 4: 0.193254 seconds (Sampling)
Chain 4: 0.406183 seconds (Total)
Chain 4:
```

### Question 3

Make two plots (appropriately labeled and described) that illustrate the differences in estimated  $\theta_i$ 's across regions and the differences in  $\theta$ s across models.

```
res_mod1 <- mod1 |>
  gather_draws(log_theta[i]) |>
  median_qi() |>
  rename(median_mod1 = .value,
         lower_mod1 = .lower,
         upper_mod1 = .upper) |>
  dplyr::select(i, median_mod1:upper_mod1)

res_mod2 <- mod2 |>
  gather_draws(log_theta[i]) |>
  median_qi() |>
  rename(median_mod2 = .value,
         lower_mod2 = .lower,
         upper_mod2 = .upper) |>
  dplyr::select(i, median_mod2:upper_mod2)

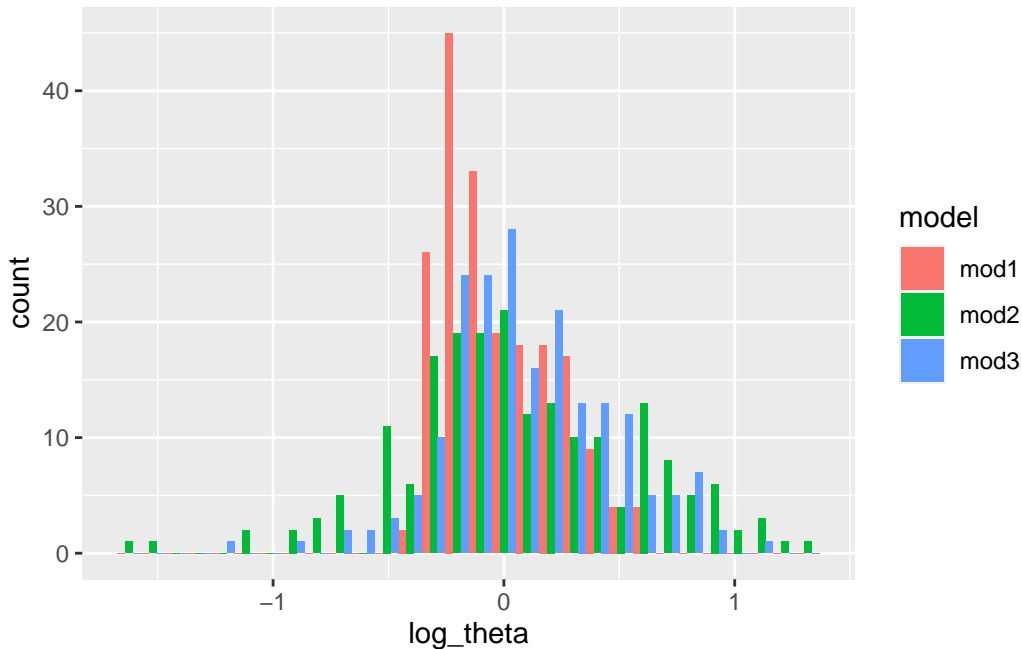
res_mod3 <- mod3 |>
  gather_draws(log_theta[i]) |>
  median_qi() |>
  rename(median_mod3 = .value,
         lower_mod3 = .lower,
         upper_mod3 = .upper) |>
  dplyr::select(i, median_mod3:upper_mod3)

res <- res_mod1 |>
  left_join(res_mod2) |>
  left_join(res_mod3)
```

```

res |>
  select(median_mod1,
         median_mod2,
         median_mod3) |>
  pivot_longer(median_mod1:median_mod3, names_to = "model", values_to = "log_theta") |>
  mutate(model = str_remove(model, "median_")) |>
  ggplot(aes(log_theta, fill = model)) +
  geom_histogram(position = 'dodge')

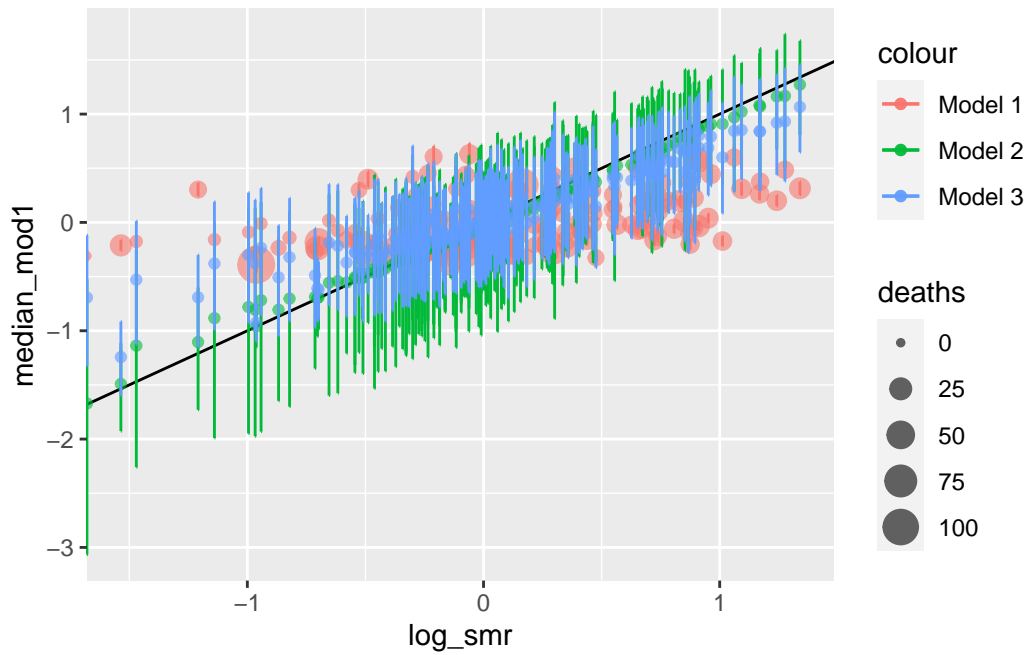
```



```

res |>
  mutate(deaths = observe.i) |>
  mutate(log_smr = log(observe.i/expect.i)) |>
  ggplot(aes(log_smr, median_mod1, color = 'Model 1')) +
  geom_point(aes(size = deaths), alpha = 0.6) +
  geom_errorbar(aes(ymin = lower_mod1, ymax = upper_mod1, color = 'Model 1'))+
  geom_abline(slope = 1, intercept = 0)+
  geom_point(aes(log_smr, median_mod2, color = 'Model 2'), alpha = 0.6)+
  geom_errorbar(aes(ymin = lower_mod2, ymax = upper_mod2, color = 'Model 2'))+
  geom_point(aes(log_smr, median_mod3, color = 'Model 3'), alpha = 0.6)+
  geom_errorbar(aes(ymin = lower_mod3, ymax = upper_mod3, color = 'Model 3'))

```



the uncertainty level fro model 1 is less but it is more truncated towards points.

P.S. I sent the code to Philip because he forgot to bring his laptop...just in case our answers look very similar :D