Lab9_STA602

ElenaW.

11/15/2021

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
require(rstanarm)
```

```
## Loading required package: rstanarm
```

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

```
## This is rstanarm version 2.21.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
require(magrittr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(ggplot2)
library(mlmRev)
```

```
## Loading required package: lme4

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(tidybayes)
library(ggstance)
```

```
##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
library(dplyr)
library(modelr)
library(brms)
```

```
## Loading 'brms' package (version 2.16.1). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').

##
## Attaching package: 'brms'

## The following objects are masked from 'package:tidybayes':
##
##     dstudent_t, pstudent_t, qstudent_t, rstudent_t

## The following object is masked from 'package:lme4':
##
##     ngrps
```

```
## The following objects are masked from 'package:rstanarm':
##
##     dirichlet, exponential, get_y, lasso, ngrps

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

```
data(Gcsemv, package = "mlmRev")
dim(Gcsemv)
```

```
## [1] 1905    5
```

```
summary(Gcsemv)
```

```
##      school        student     gender      written          course
##  68137  : 104   77     :  14   F:1128   Min.   : 0.60   Min.   :  9.25
##  68411  :  84   83     :  14   M: 777   1st Qu.:37.00   1st Qu.: 62.90
##  68107  :  79   53     :  13            Median :46.00   Median : 75.90
##  68809  :  73   66     :  13            Mean   :46.37   Mean   : 73.39
##  22520  :  65   27     :  12            3rd Qu.:55.00   3rd Qu.: 86.10
##  60457  :  54   110    :  12            Max.   :90.00   Max.   :100.00
##  (Other):1446   (Other):1827            NA's   :202     NA's   :180
```

```
# Make Male the reference category and rename variable
Gcsemv$female <- relevel(Gcsemv$gender, "M")

# Use only total score on coursework paper
GCSE <- subset(x = Gcsemv,
               select = c(school, student, female, course))

# Count unique schools and students
m <- length(unique(GCSE$school))
N <- nrow(GCSE)
```

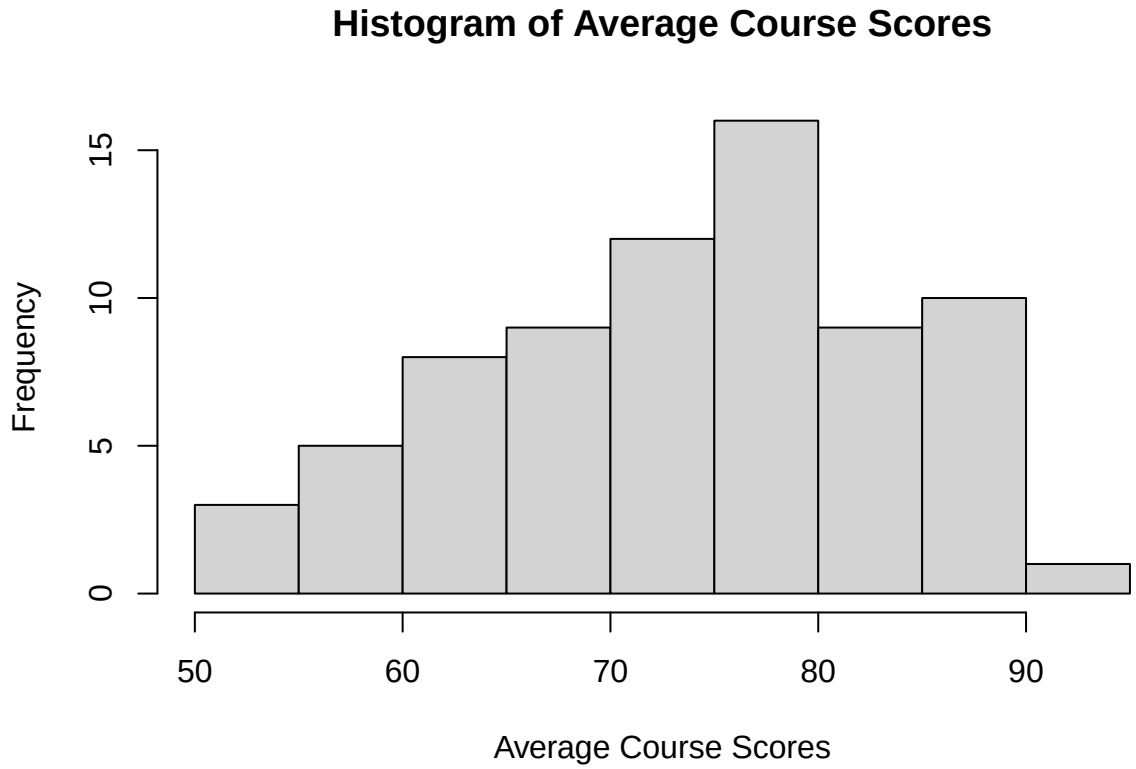
Exercise 1

```
GCSE = na.omit(GCSE)
average_course = GCSE %>%
  group_by(school) %>%
  summarise(average_course = mean(course))
average_course
```

```
## # A tibble: 73 x 2
##    school average_course
##    <fct>           <dbl>
##  1 20920            60
##  2 22520            56.4
##  3 22710            82.9
```

```
##  4 22738            72.9
##  5 22908            63.1
##  6 23208            79.8
##  7 25241            71.0
##  8 30474            82.3
##  9 35270            60.2
## 10 37224            66.4
## # ... with 63 more rows
```

```
hist(average_course$average_course,
     xlab = "Average Course Scores",
     main = "Histogram of Average Course Scores")
```



From the histogram above, we could see that the distribution is left skewed, and the overall average course scores of schools kind of spread off. Students in different schools performs differently. Thus, may it's not a good idea to do information share for mean in the hierarchical model.

```
pooled <- stan_glm(course ~ 1 + female, data = GCSE, refresh = 0)
unpooled <- stan_glm(course ~ -1 + school + female,data=GCSE, refresh = 0)
```

```
mod1 <- stan_lmer(formula = course ~ 1 + (1 | school),
                  data = GCSE,
                  seed = 349,
                  refresh = 0)
```

```
prior_summary(object = mod1)
```

```
## Priors for model 'mod1'
## ------
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 73, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 73, scale = 41)
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 0.061)
##
## Covariance
##  ~ decov(reg. = 1, conc. = 1, shape = 1, scale = 1)
## ------
## See help('prior_summary.stanreg') for more details
```

```
sd(GCSE$course, na.rm = T)
```

```
## [1] 16.32096
```

```
print(mod1, digits = 3)
```

```
## stan_lmer
##  family:       gaussian [identity]
##  formula:      course ~ 1 + (1 | school)
##  observations: 1725
## ------
##             Median MAD_SD
## (Intercept) 73.677  1.138
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 13.819  0.243
##
## Error terms:
##  Groups   Name        Std.Dev.
##  school   (Intercept)  8.869
##  Residual             13.819
## Num. levels: school 73
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
summary(mod1,
        pars = c("(Intercept)", "sigma", "Sigma[school:(Intercept),(Intercept)]"),
        probs = c(0.025, 0.975),
        digits = 3)
```

```
##
## Model Info:
##  function:     stan_lmer
##  family:       gaussian [identity]
##  formula:      course ~ 1 + (1 | school)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1725
##  groups:       school (73)
##
## Estimates:
##                                        mean    sd      2.5%    97.5%
## (Intercept)                           73.665   1.125  71.486  75.921
## sigma                                 13.819   0.239  13.361  14.298
## Sigma[school:(Intercept),(Intercept)]  78.667  15.550  53.464 114.098
##
## MCMC diagnostics
##                                       mcse  Rhat  n_eff
## (Intercept)                           0.047 1.006  572
## sigma                                 0.004 0.999 4579
## Sigma[school:(Intercept),(Intercept)] 0.614 1.002  641
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Exercise 2

From the code above, μ is 73.665, σ is 13.819, and τ^2 is 78.667.

```
mod1_sims <- as.matrix(mod1)
dim(mod1_sims)
```

```
## [1] 4000   76
```

```
par_names <- colnames(mod1_sims)
head(par_names)
```

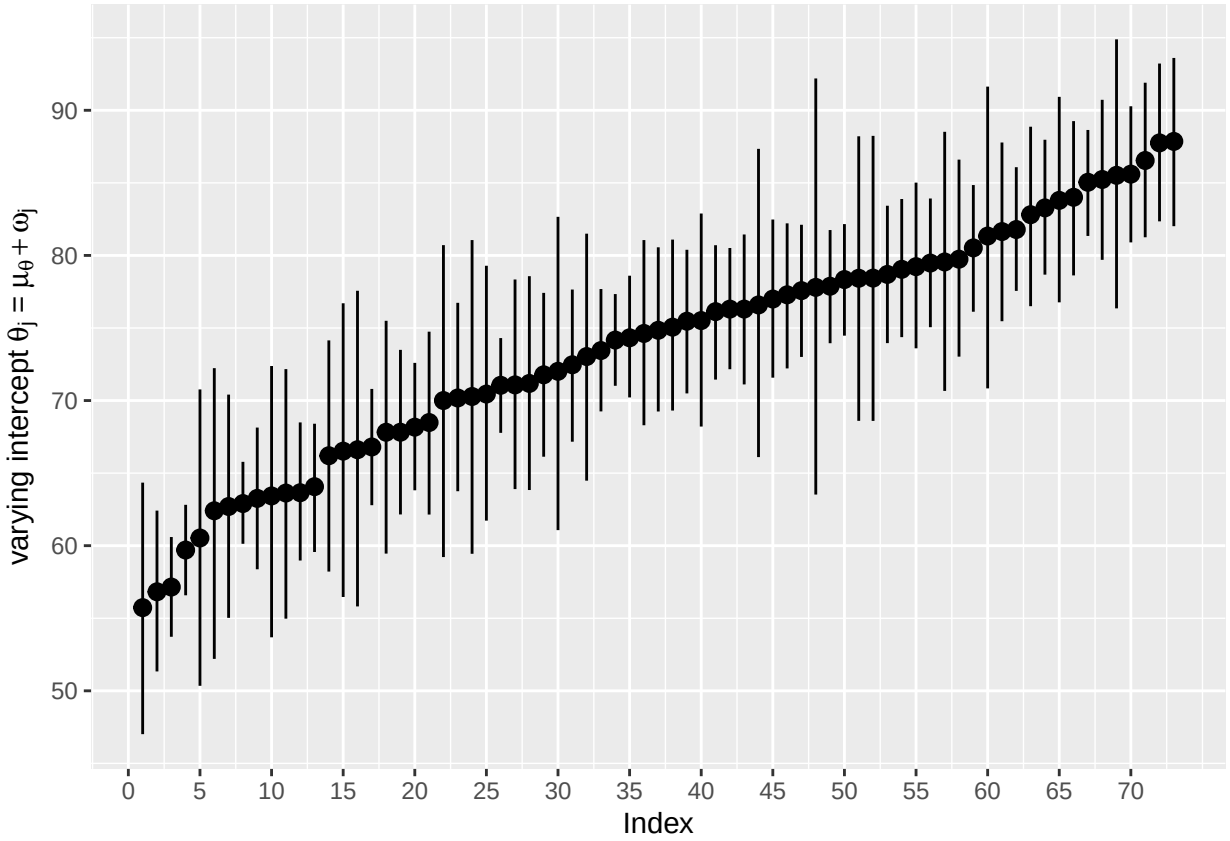
```
## [1] "(Intercept)"                 "b[(Intercept) school:20920]"
## [3] "b[(Intercept) school:22520]" "b[(Intercept) school:22710]"
## [5] "b[(Intercept) school:22738]" "b[(Intercept) school:22908]"
```

```
tail(par_names)
```

```
## [1] "b[(Intercept) school:76631]"
## [2] "b[(Intercept) school:77207]"
## [3] "b[(Intercept) school:84707]"
## [4] "b[(Intercept) school:84772]"
## [5] "sigma"
## [6] "Sigma[school:(Intercept),(Intercept)]"
```

```r
# obtain draws for mu_theta
mu_theta_sims <- as.matrix(mod1, pars = "(Intercept)")

# obtain draws for each school's contribution to intercept
omega_sim <- as.matrix(mod1,
                       regex_pars ="b\\[\\(Intercept\\) school\\:")

# to finish: obtain draws for sigma and tau^2
sig_sims <- as.matrix(mod1,
                      pars = "sigma")
tau2_sims <- as.matrix(mod1,
                       pars = "Sigma[school:(Intercept),(Intercept)]")
```

```r
# posterior samples of intercepts, which is overall intercept + school-specific intercepts
int_sims <- as.numeric(mu_theta_sims) + omega_sim

# posterior mean
int_mean <- apply(int_sims, MARGIN = 2, FUN = mean)

# credible interval
int_ci <- apply(int_sims, MARGIN = 2, FUN = quantile, probs = c(0.025, 0.975))
int_ci <- data.frame(t(int_ci))

# combine into a single df
int_df <- data.frame(int_mean, int_ci)
names(int_df) <- c("post_mean","Q2.5", "Q97.5")

# sort DF according to posterior mean
int_df <- int_df[order(int_df$post_mean),]

# create variable "index" to represent order
int_df <- int_df %>% mutate(index = row_number())

# plot posterior means of school-varying intercepts, along with 95 CIs
ggplot(data = int_df, aes(x = index, y = post_mean))+
  geom_pointrange(aes(ymin = Q2.5, ymax = Q97.5))+
  scale_x_continuous("Index", breaks = seq(0,m, 5)) +
  scale_y_continuous(expression(paste("varying intercept ", theta[j], " = ", mu[theta]+omega[j])))
```

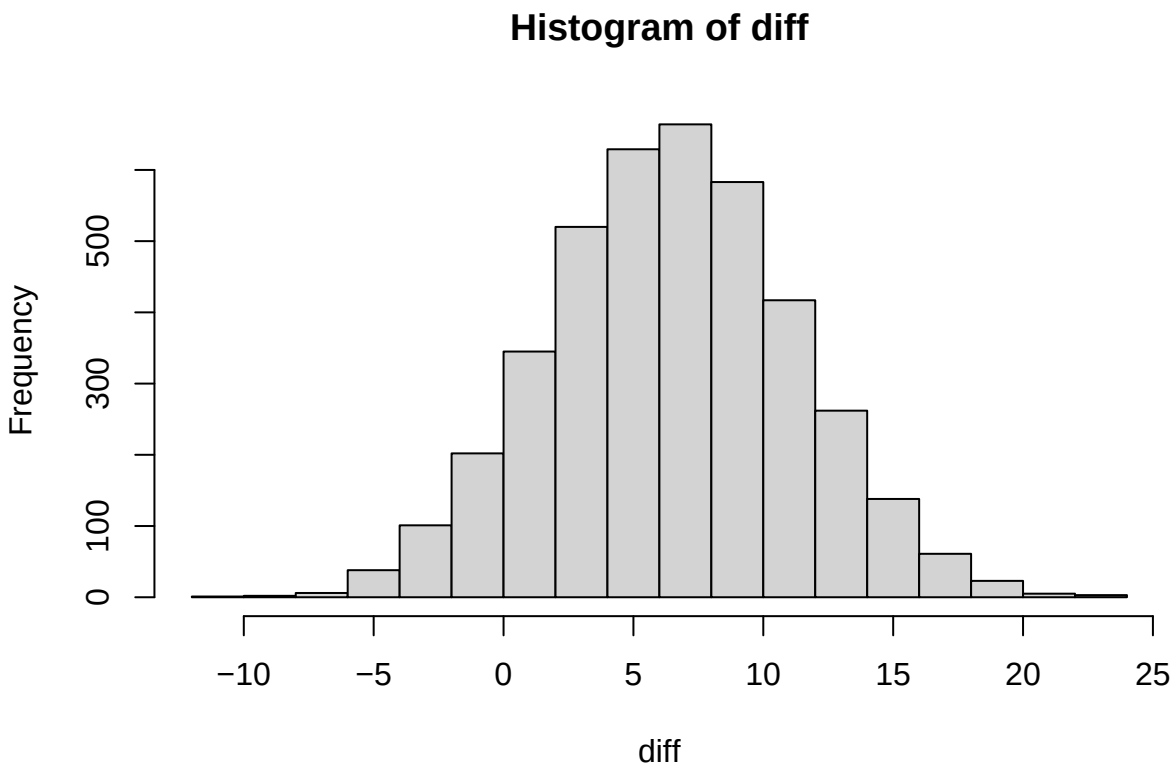


Exercise 3

```
school_20920 = as.matrix(mod1, pars = "b[(Intercept) school:20920]")
school_22520 = as.matrix(mod1, pars = "b[(Intercept) school:22520]")
diff = school_20920 - school_22520
summary(diff)
```

```
##  b[(Intercept) school:20920]
##  Min.   :-10.001
##  1st Qu.:  3.252
##  Median :  6.504
##  Mean   :  6.478
##  3rd Qu.:  9.697
##  Max.   : 23.687
```

```
hist(diff)
```



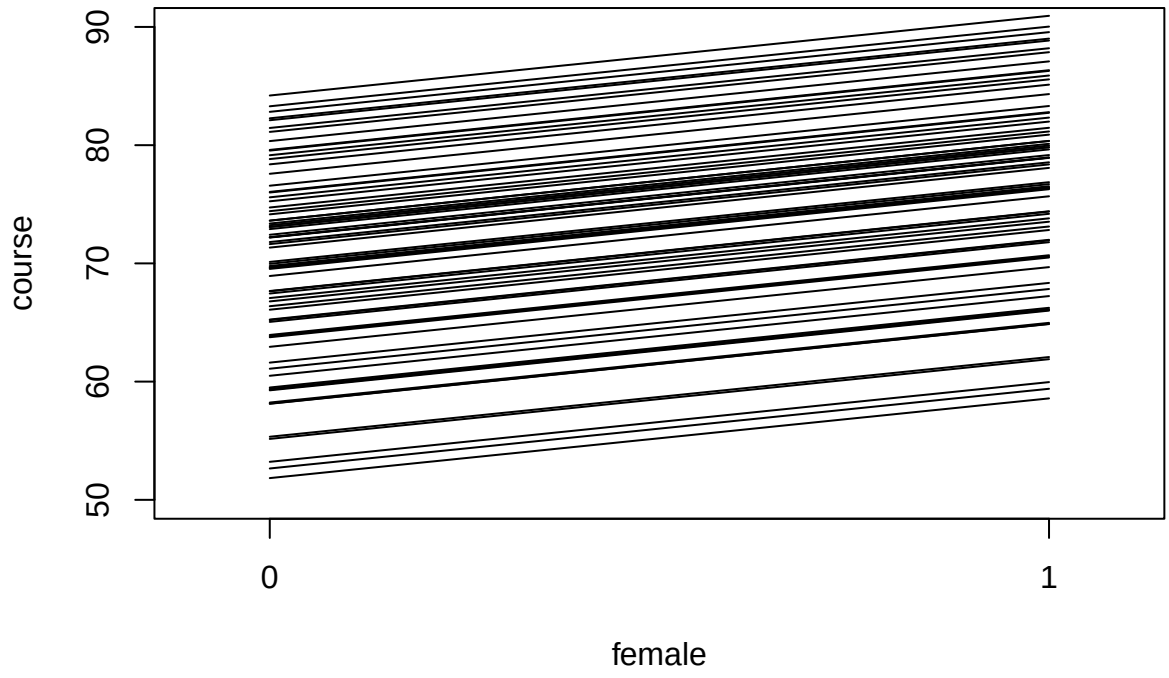
From the results above, we could see that the difference between school 20920 and school 22520 is from -10 to 23.7, and the center is around 6.5, which means that the posterior averages of this two school are different and we don't have strong evidence say that which one is higher than another.

Model 2: Varying intercept with a single individual-level predictor

```
mod2 <- stan_lmer(formula = course ~ 1 + female + (1 | school),
                  data = GCSE,
                  prior = normal(location = 0,
                                 scale = 100,
                                 autoscale = F),
                  prior_intercept = normal(location = 0,
                                 scale = 100,
                                 autoscale = F),
                  seed = 349,
                  refresh = 0)
```

```
# plot varying intercepts
mod2.sims <- as.matrix(mod2)
group_int <- mean(mod2.sims[,1])
mp <- mean(mod2.sims[,2])
bp <- apply(mod2.sims[, 3:75], 2, mean)
xvals <- seq(0,1,.01)
plot(x = xvals, y = rep(0, length(xvals)),
```

```
     ylim = c(50, 90), xlim = c(-0.1,1.1), xaxt = "n", xlab = "female", ylab = "course")
axis(side = 1, at = c(0,1))
for (bi in bp){
  lines(xvals, (group_int + bi)+xvals*mp)
}
```



Exercise 4

```
summary(mod2,
pars = c("(Intercept)", "sigma", "femaleF","Sigma[school:(Intercept),(Intercept)]"),
probs = c(0.025, 0.975),
digits = 3)
```

```
##
## Model Info:
##  function:     stan_lmer
##  family:       gaussian [identity]
##  formula:      course ~ 1 + female + (1 | school)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1725
##  groups:       school (73)
```

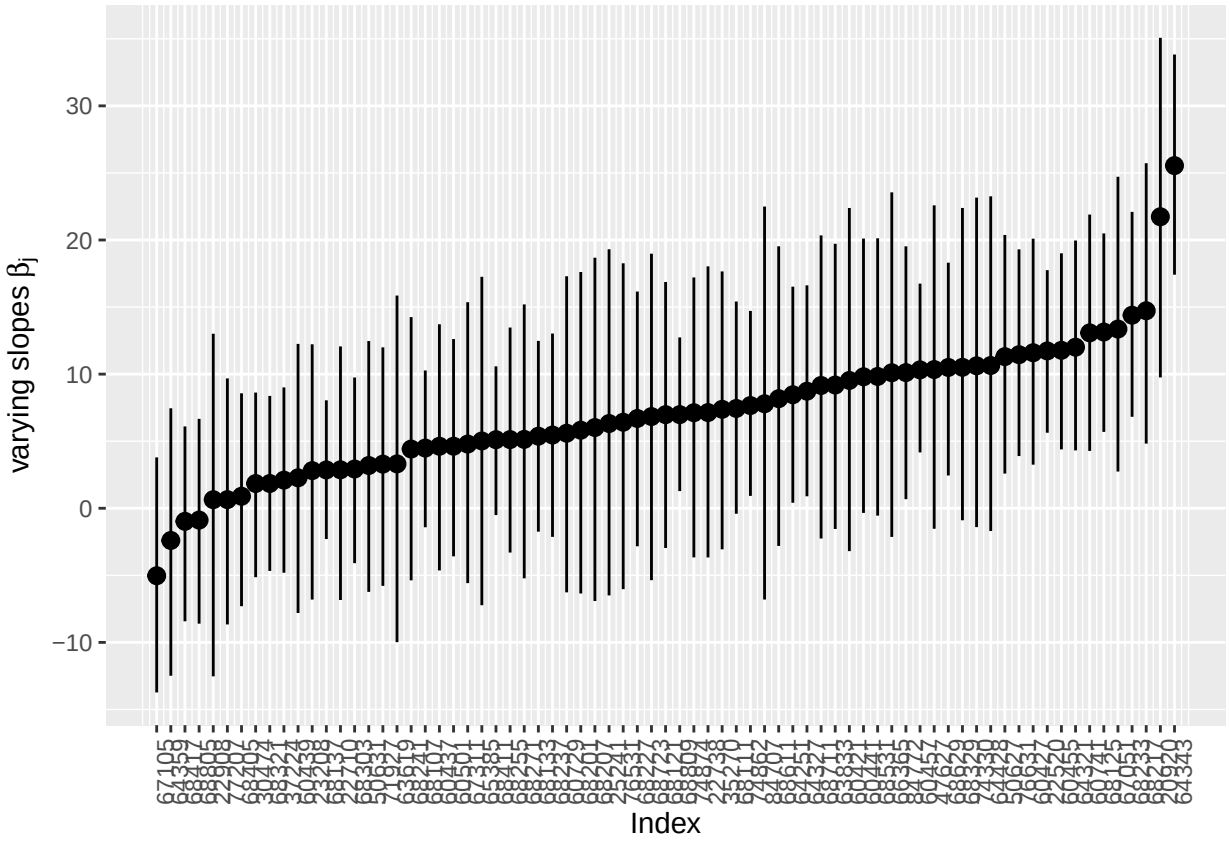
```
##
## Estimates:
##                                          mean    sd      2.5%    97.5%
## (Intercept)                             69.661   1.218  67.219  72.045
## femaleF                                  6.743   0.675   5.423   8.099
## sigma                                   13.427   0.239  12.968  13.906
## Sigma[school:(Intercept),(Intercept)]   80.900  16.258  54.571 115.802
##
## MCMC diagnostics
##                                        mcse  Rhat  n_eff
## (Intercept)                            0.057 1.005  455
## femaleF                                0.009 1.000 5634
## sigma                                  0.004 0.999 3550
## Sigma[school:(Intercept),(Intercept)] 0.628 1.000  671
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

From the results above, we could see that μ_θ is 69.661 α is 13.427 τ^2 is 80.900 and β is 6.743.

Model 3

```
mod3 <- stan_lmer(formula = course~ 1+ female + (1 + female | school),
                  data = GCSE,
                  seed = 349,
                  refresh = 0)
mod3_sims <- as.matrix(mod3)

# obtain draws for mu_theta
mu_theta_sims <- as.matrix(mod3, pars = "(Intercept)")

fem_sims <- as.matrix(mod3, pars = "femaleF")
# obtain draws for each school's contribution to intercept
omega_sims <- as.matrix(mod3,
                        regex_pars ="b\\[\\(Intercept\\) school\\:")
beta_sims <- as.matrix(mod3,
                       regex_pars ="b\\[femaleF school\\:")

int_sims <- as.numeric(mu_theta_sims) + omega_sims
slope_sims <- as.numeric(fem_sims) + beta_sims

# posterior mean
slope_mean <- apply(slope_sims, MARGIN = 2, FUN = mean)

# credible interval
slope_ci <- apply(slope_sims, MARGIN = 2, FUN = quantile, probs = c(0.025, 0.975))
slope_ci <- data.frame(t(slope_ci))

# combine into a single df
slope_df <- data.frame(slope_mean, slope_ci, levels(GCSE$school))
names(slope_df) <- c("post_mean","Q2.5", "Q97.5", "school")

# sort DF according to posterior mean
```

```
slope_df <- slope_df[order(slope_df$post_mean),]

# create variable "index" to represent order
slope_df <- slope_df %>% mutate(index = row_number())

# plot posterior means of school-varying slopes, along with 95% CIs
ggplot(data = slope_df, aes(x = index, y = post_mean))+
  geom_pointrange(aes(ymin = Q2.5, ymax = Q97.5))+
  scale_x_continuous("Index", breaks = seq(1,m, 1),
                     labels = slope_df$school) +
  scale_y_continuous(expression(paste("varying slopes ", beta[j])))+
  theme(axis.text.x = element_text(angle = 90))
```




```
loo1 <- loo(mod1)
loo2 <- loo(mod2)
loo3 <- loo(mod3)
```

```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argumen
```

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

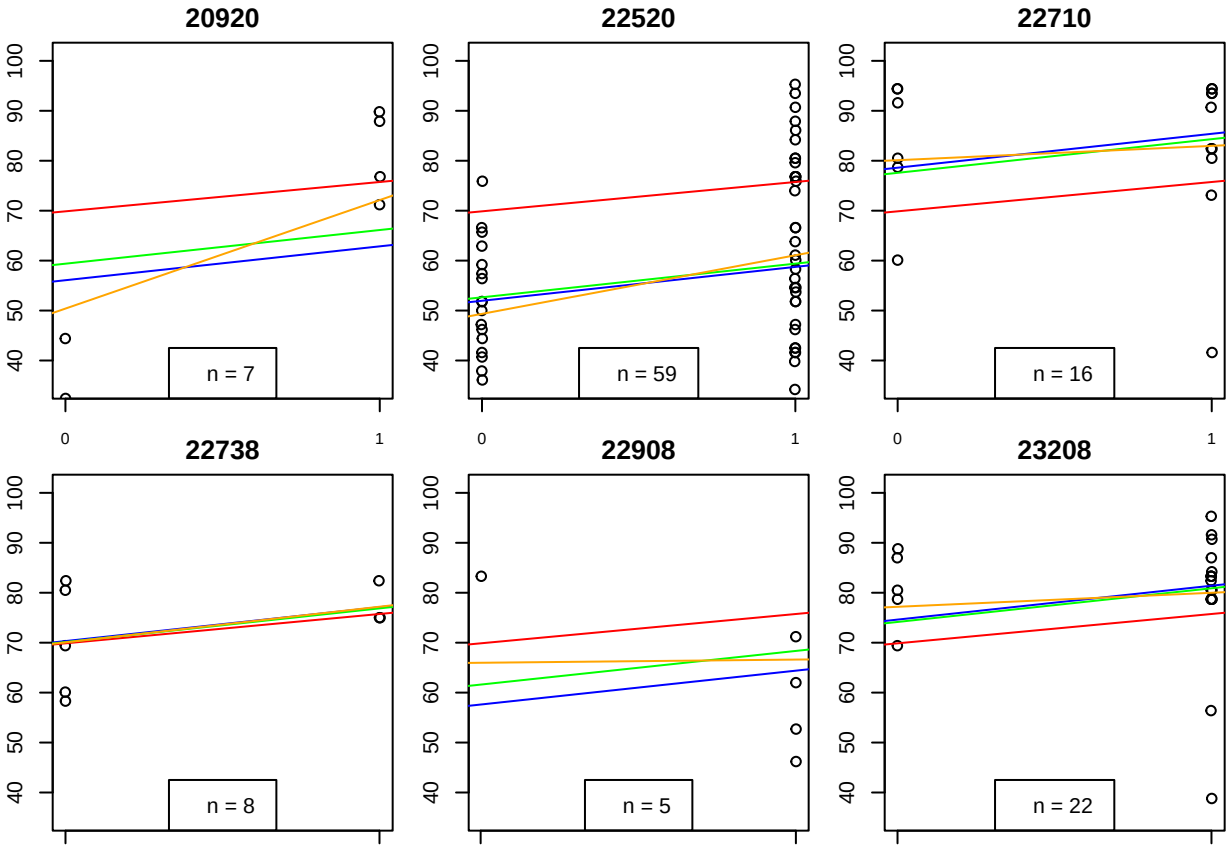
```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod2 -30.7       9.9
## mod1 -80.1      15.1
```

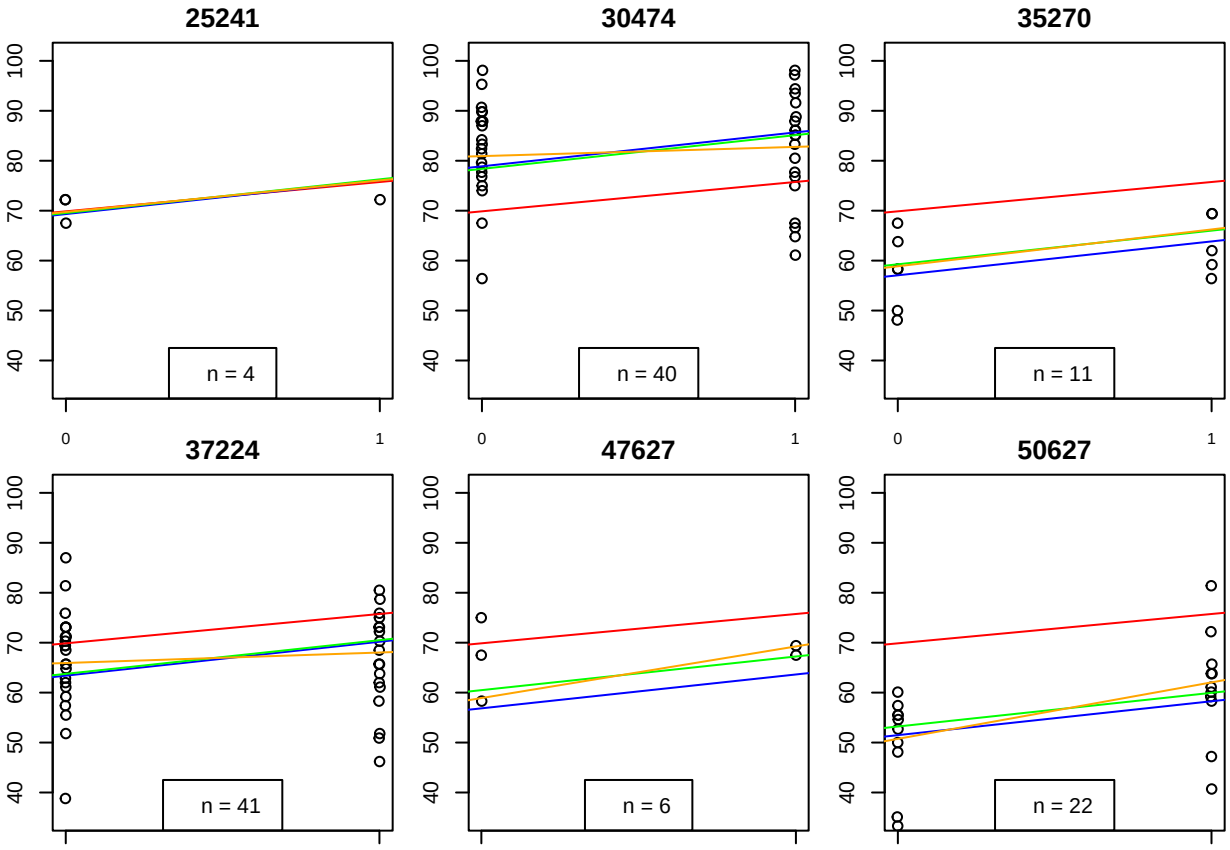
```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod2 -29.6       9.9
## mod1 -79.4      15.1
loo_compare(loo1, loo3)
```

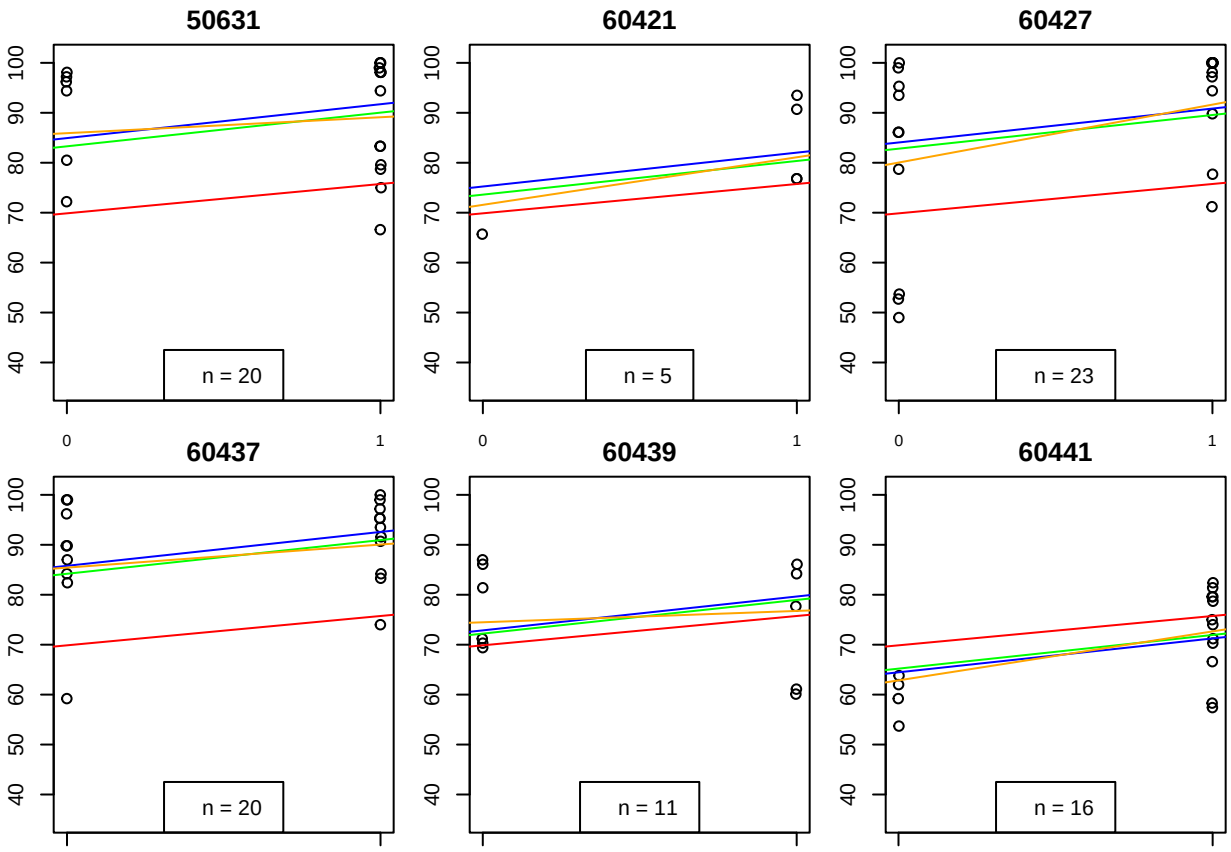
```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod1 -80.1      15.1
```

```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod1 -79.4      15.1
```

```
pooled.sim <- as.matrix(pooled)
unpooled.sim <- as.matrix(unpooled)
m1.sim <- as.matrix(mod1)
m2.sim <- as.matrix(mod2)
m3.sim <- as.matrix(mod3)
schools <- unique(GCSE$school)


alpha2 = mean(m2.sim[,1])
alpha3 <- mean(m3.sim[,1])

partial.fem2 <- mean(m2.sim[,2])
partial.fem3 <- mean(m3.sim[,2])
unpooled.fem <- mean(unpooled.sim[,74])

par(mfrow = c(2, 3), mar = c(1,2,2,1))
for (i in 1:18){
  temp = GCSE %>% filter(school == schools[i]) %>%
    na.omit()
  y <- temp$course
  x <- as.numeric(temp$female)-1
  plot(x + rnorm(length(x)) *0.001, y, ylim = c(35,101), xlab = "female",main =schools[i], xaxt = "n", y
  axis(1,c(0,1),cex.axis=0.8)

  # no pooling
  b = mean(unpooled.sim[,i])

  # plot lines and data
  xvals = seq(-0.1, 1.1, 0.01)
  lines(xvals, xvals * mean(pooled.sim[,2]) + mean(pooled.sim[,1]), col = "red") # pooled
  lines(xvals, xvals * unpooled.fem + b, col = "blue") # unpooled
  lines(xvals, xvals*partial.fem2 + (alpha2 + mean(m2.sim[,i+2])) , col = "green") # varying int
  lines(xvals, xvals*(partial.fem3 + mean(m3.sim[, 2 + i*2])) + (alpha3 + mean(m3.sim[, 1 + i*2])), col
  legend("bottom", legend = paste("n =", length(y), " "))
}
```


20920
22520
22710
22738
22908
23208
n = 7
n = 59
n = 16
n = 8
n = 5
n = 22
0
1
40
50
60
70
80
90
100


25241
n = 4
30474
n = 40
35270
n = 11
37224
n = 41
47627
n = 6
50627
n = 22
40
50
60
70
80
90
100
0
1



Exercise 5

From the regression lines above, we could see that the model 3 perform better than model 1 and 2 since the model 3 is more flexible. From Bayesian Shrinkage perspective, the green line (model2) is between red and blue lines. If the sample size is large, green would be closer to the blue line, and if the sample size is smaller, it would be closer to the red line, which means that model 2 will more depend on the sample size of schools. From loo_compare result, we could see that model1 and model2 have negative value compared to model 3, which means that the model 3 perform better.

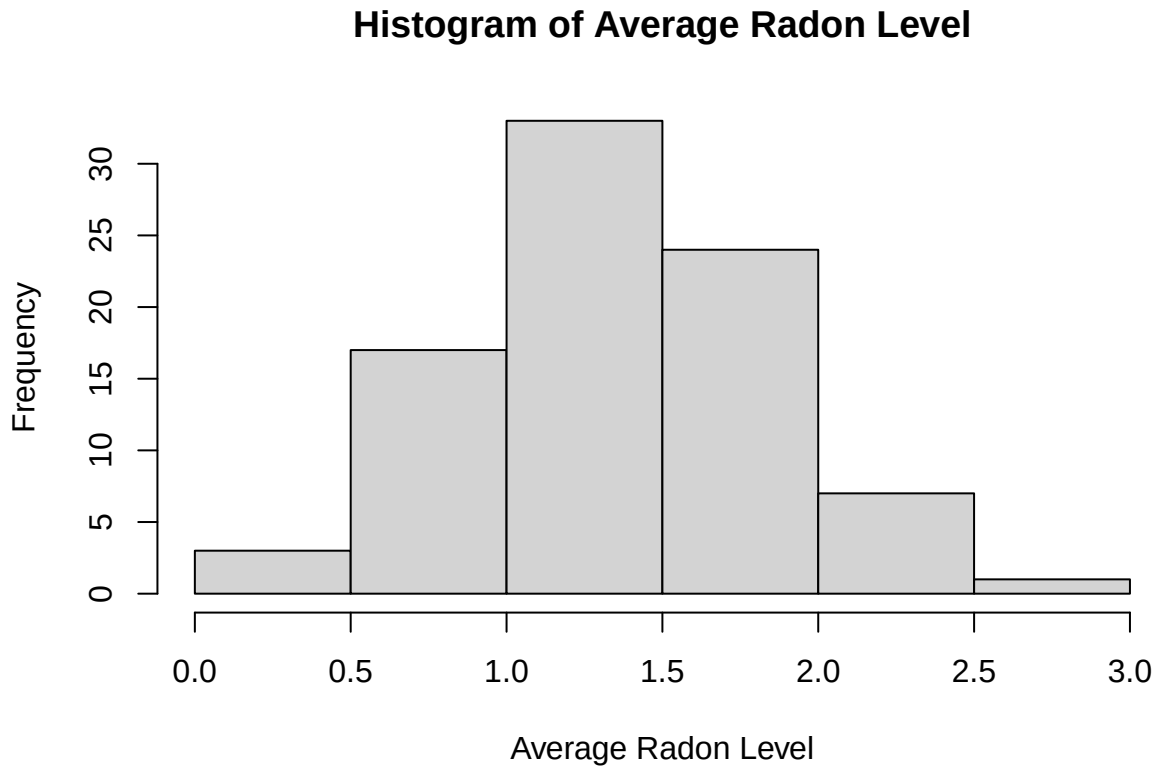
```
radon <- read.csv("radon.txt", header = T,sep="")
radon$county <- as.factor(radon$county)
```

```
average_radon = radon %>%
  group_by(county) %>%
  summarise(average_radon = mean(log_radon,na.rm = T))
average_radon
```

```
## # A tibble: 85 x 2
##    county average_radon
##    <fct>          <dbl>
##  1 1              0.660
##  2 2              0.833
##  3 3              1.05
##  4 4              1.14
```

```
##  5 5              1.25
##  6 6              1.51
##  7 7              1.91
##  8 8              1.63
##  9 9              0.931
## 10 10             1.20
## # ... with 75 more rows
```

```
hist(average_radon$average_radon,
     xlab = "Average Radon Level",
     main = "Histogram of Average Radon Level")
```



From the histogram above, we could see that the average of log_radon is different across the the counties, it's good idea to do the hierarchical model but the information may not be shared.

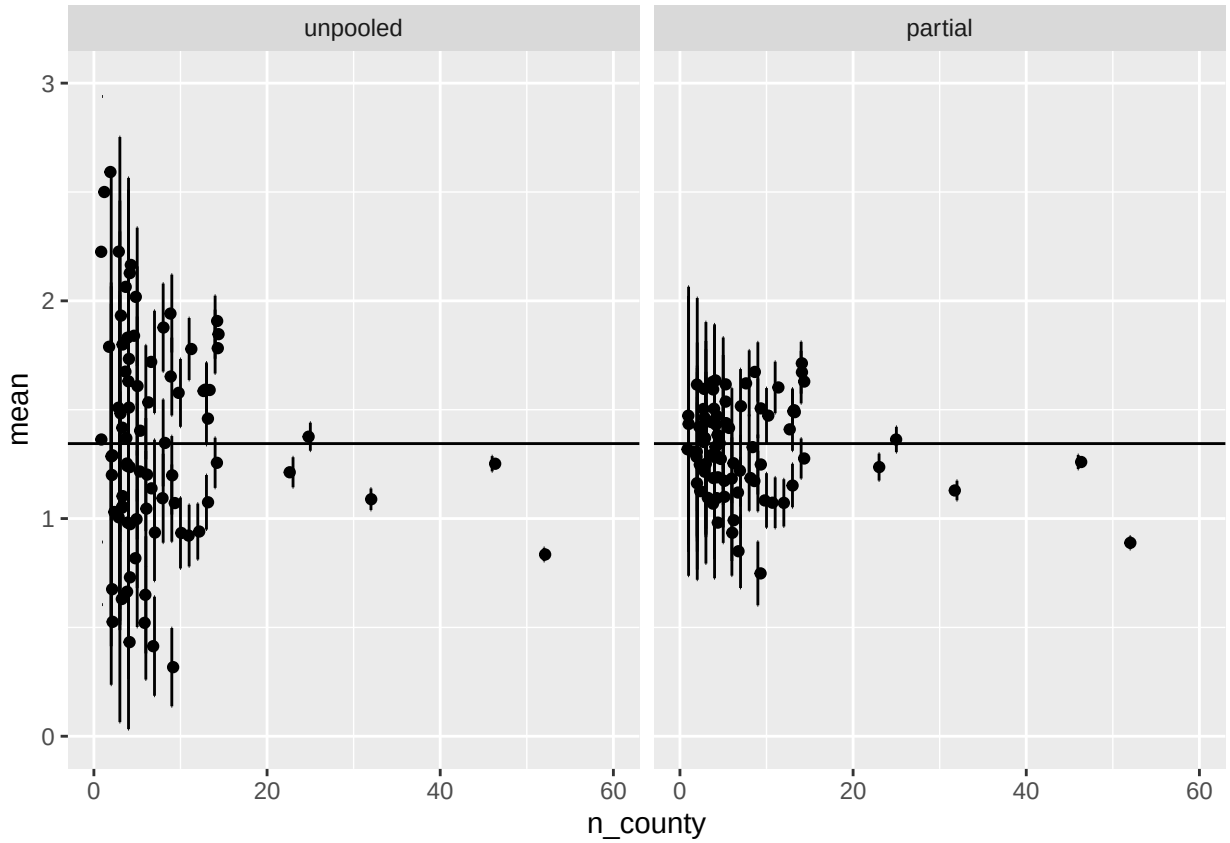
Exercise 7

```
radon.unpooled <- stan_glm(log_radon ~ -1 + county,data=radon, refresh = 0)
```

```
radon.mod1 <- stan_lmer(formula = log_radon ~ 1 + (1 | county),
data = radon,
seed = 349,
refresh = 0)
```

```
n_county <- as.numeric(table(radon$county))
create_df <- function(sim,model){
mean <- apply(sim,2,mean)
sd <- apply(sim,2,sd)
df <- cbind(n_county, mean, sd) %>%
as.data.frame()%>%
mutate(se = sd/ sqrt(n_county), model = model)
return(df)
}
unpooled.sim <- as.matrix(radon.unpooled)
unpooled.df <- create_df(unpooled.sim[,1:85], model = "unpooled")

mod1.sim <- as.matrix(radon.mod1)[,1:86]
mod1.sim <- (mod1.sim[,1] + mod1.sim)[,-1]
partial.df <- create_df(mod1.sim, model = "partial")
ggplot(rbind(unpooled.df, partial.df)%>% mutate(model = factor(model, levels = c("unpooled", "partial"))
#draws the means
geom_jitter() +
#draws the CI error bars
geom_errorbar(aes(ymin=mean-2*se, ymax= mean+2*se), width=.1)+
ylim(0,3)+
xlim(0,60)+
geom_hline(aes(yintercept= mean(coef(radon.unpooled))))+
facet_wrap(~model)
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



From the plots above, we could see that the Bayesian Shrinkage would be weaker with the larger sample size in both plots, and with the larger sample size, there is more uncertainty and larger posterior credible interval for mean.

Exercise 8

```
radon.mod2 <- stan_lmer(formula = log_radon ~ 1 + floor + (1 | county),
data = radon,
prior = normal(location = 0,
scale = 100,
autoscale = F),
prior_intercept = normal(location = 0,
scale = 100,
autoscale = F),
seed = 349,
refresh = 0)
radon.mod3 <- stan_lmer(formula = log_radon ~ 1+ floor + (1 + floor | county),
data = radon,
seed = 349,
refresh = 0)
radon.mod4 <- stan_lmer(formula = log_radon ~ 1 + floor + log_uranium + (1 | county),
data = radon,
prior = normal(location = 0,
scale = 100,
```

```
autoscale = F),
prior_intercept = normal(location = 0,
scale = 100,
autoscale = F),
seed = 349,
refresh = 0)
```

```
loo1 <- loo(radon.mod1)
loo2 <- loo(radon.mod2)
loo3 <- loo(radon.mod3)
```

```
## Warning: Found 2 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argumen
```

```
loo4 <- loo(radon.mod4)
loo_compare(loo1,loo2,loo3,loo4)
```

```
##            elpd_diff se_diff
## radon.mod4   0.0       0.0
## radon.mod2  -9.3       5.3
## radon.mod3 -10.3       5.8
## radon.mod1 -56.7      11.9
```

From the result above, we could see that the model 4 has the best performance compared to other negative models.

Exercise 9

With larger sample size, the Bayesian shrinkage would be less towards the other groups, which means that there would be less information borrowing/sharing. In other words, information sharing would be better to use for smaller sample size and modeling each group totally separately would be not a good idea in small sample size groups.