STA 602 Lab 9

Yicheng Shen

21 November, 2022

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

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

```r
summary(Gcsemv)
```

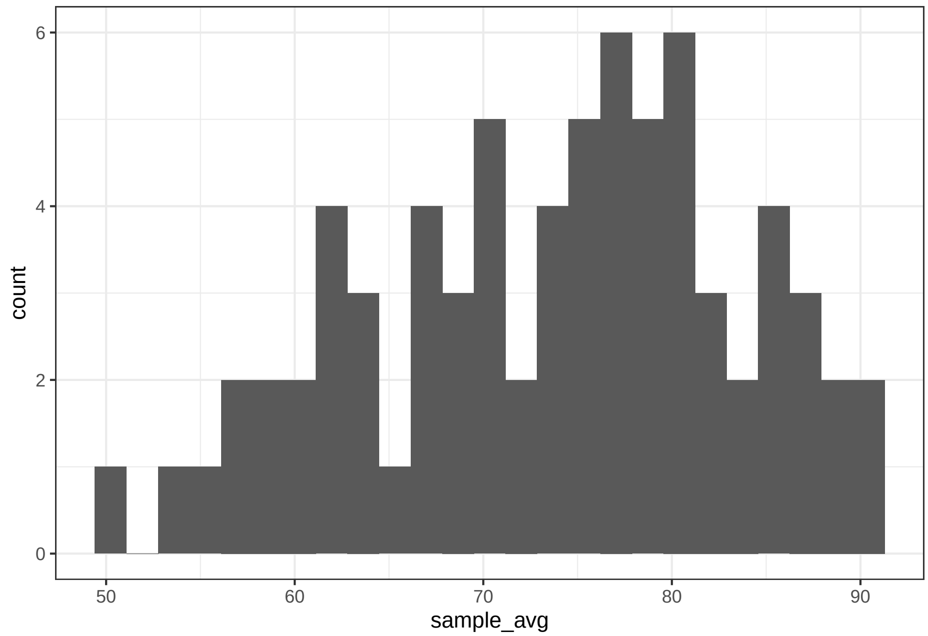
```
##      school        student     gender     written          course      
##  68137  : 104   77     :  14   F:1128   Min.   : 0.60   Min.   :  9.25  
##  68411  :  84   83     :  14   M: 777   1st Qu.:37.00   1st Qu.: 62.90  
##  68107  :  79   53     :  13            Median :46.00   Median : 75.90  
##  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     
```

```r
# 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)
```

Ex.1

The histogram shows that depending on the school, the average `course` scores can vary a lot, with a slightly left skewed distribution.

```r
GCSE %>% group_by(school) %>% summarise(sample_avg = mean(course, na.rm=T)) %>%
  ggplot() + geom_histogram(aes(x=sample_avg), bins = 25)
```



Ex.2

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

```r
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
```

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

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

$\mu_\theta = 73.78, \tau = 8.888, \sigma = 13.821$

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

```
## stan_lmer
##  family:       gaussian [identity]
##  formula:      course ~ 1 + (1 | school)
##  observations: 1725
## ------
##             Median MAD_SD
## (Intercept) 73.780  1.124
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 13.818  0.240
##
## Error terms:
##  Groups   Name        Std.Dev.
##  school   (Intercept)  8.888
##  Residual             13.821
## Num. levels: school 73
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

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

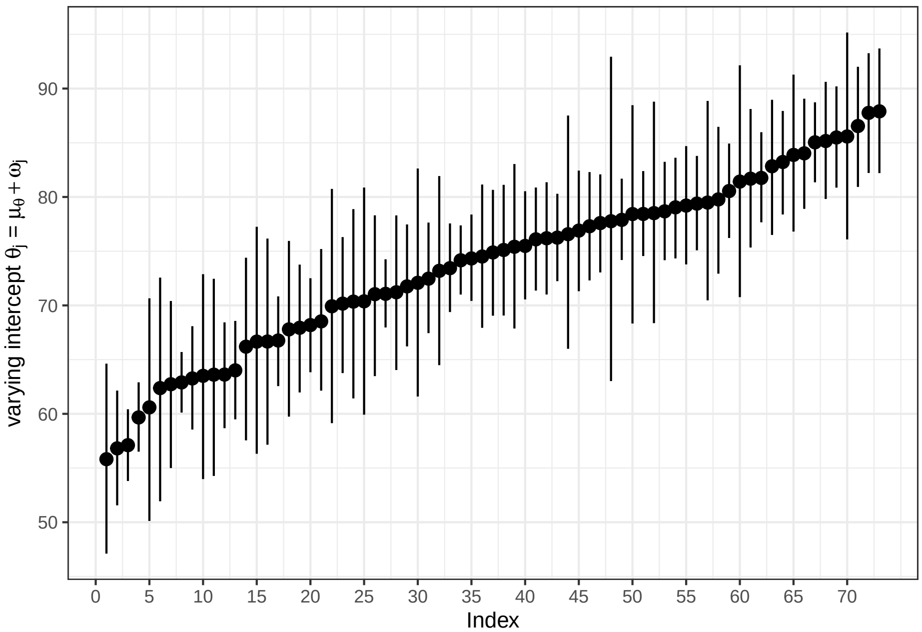
```
##
## Model Info:
##  function:     stan_lmer
##  family:       gaussian [identity]
##  formula:      course ~ 1 + (1 | school)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1725
##  groups:       school (73)
##
## Estimates:
##                                         mean    sd      2.5%    97.5%
## (Intercept)                            73.777   1.136  71.447  75.941
## sigma                                  13.821   0.241  13.362  14.299
## Sigma[school:(Intercept),(Intercept)]  79.004  16.416  52.086 114.869
```

```
##
## MCMC diagnostics
##                                        mcse  Rhat  n_eff
## (Intercept)                            0.041 1.002  756
## sigma                                  0.003 1.002 5455
## Sigma[school:(Intercept),(Intercept)] 0.671 1.011  598
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

The posterior estimates are $\mu_\theta = 73.78, \tau^2 = 79.004, \sigma = 13.821$

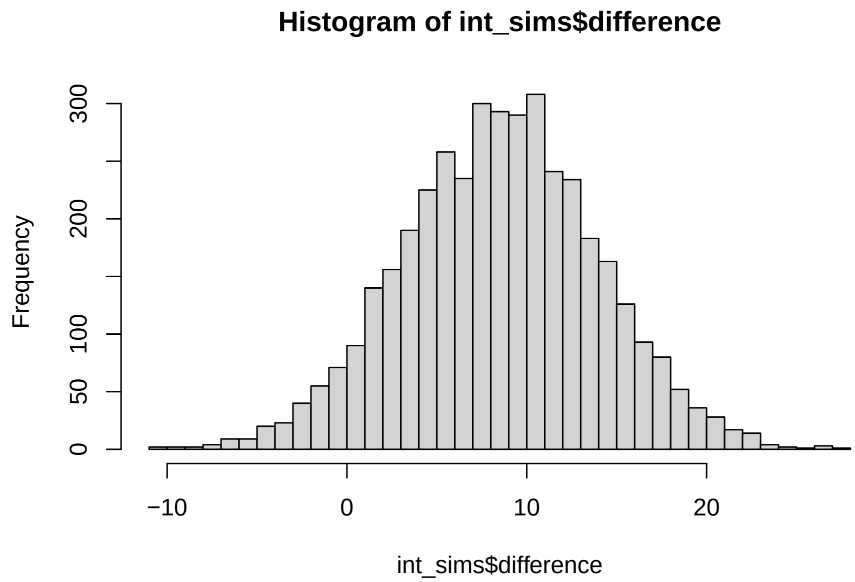
Ex.3

```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])))
```



Choose two schools, extract out the posterior samples of their average scores, and report on their difference in average scores with descriptive statistics, a histogram, and interpretation.

```
mod1_sims <- as.matrix(mod1)
mu_theta_sims <- as.matrix(mod1, pars = "(Intercept)")
omega_sim <- as.matrix(mod1,
                        regex_pars ="b\\[\\(Intercept\\) school\\:227")

int_sims <- as.numeric(mu_theta_sims) + omega_sim

int_sims <- int_sims %>% as.data.frame() %>%
  mutate(difference = `b[(Intercept) school:22710]` - `b[(Intercept) school:22738]`)

hist(int_sims$difference, breaks = 30)
```




```
summary(int_sims$difference)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -10.834    4.835    8.565    8.498   12.121   27.838
```

```
quantile(int_sims$difference, c(0.025, 0.975))
```

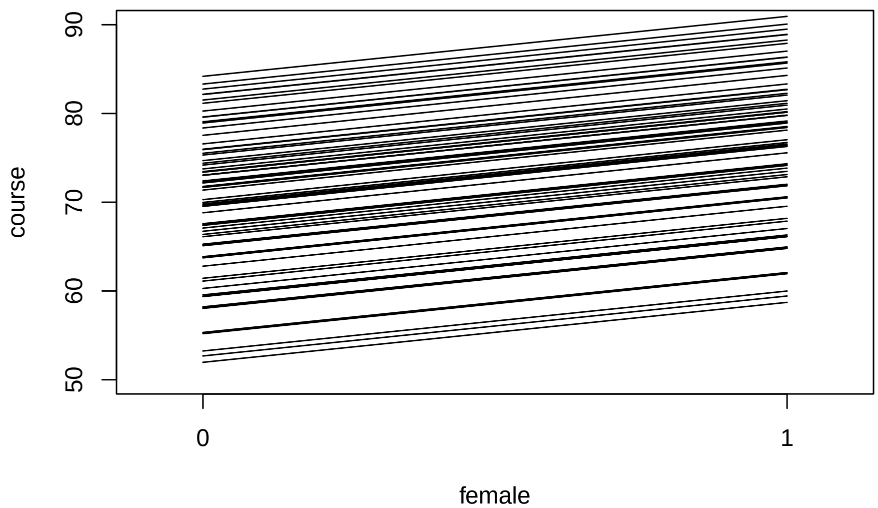
```
##      2.5%     97.5%
## -2.326096 19.187440
```

I am looking at school 22710 and 22738. It seems that the score of school 22710 is usually higher, although the difference of their 95% CI contains zero, so the difference is not significant.

Ex.4

```
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])
```

```r
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)
}
```




```r
summary(mod2,
        pars = c("(Intercept)", "femaleF", "sigma", "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)                            69.730  1.188  67.366  72.018
## femaleF                                 6.754  0.684   5.397   8.069
## sigma                                  13.420  0.237  12.959  13.906
```

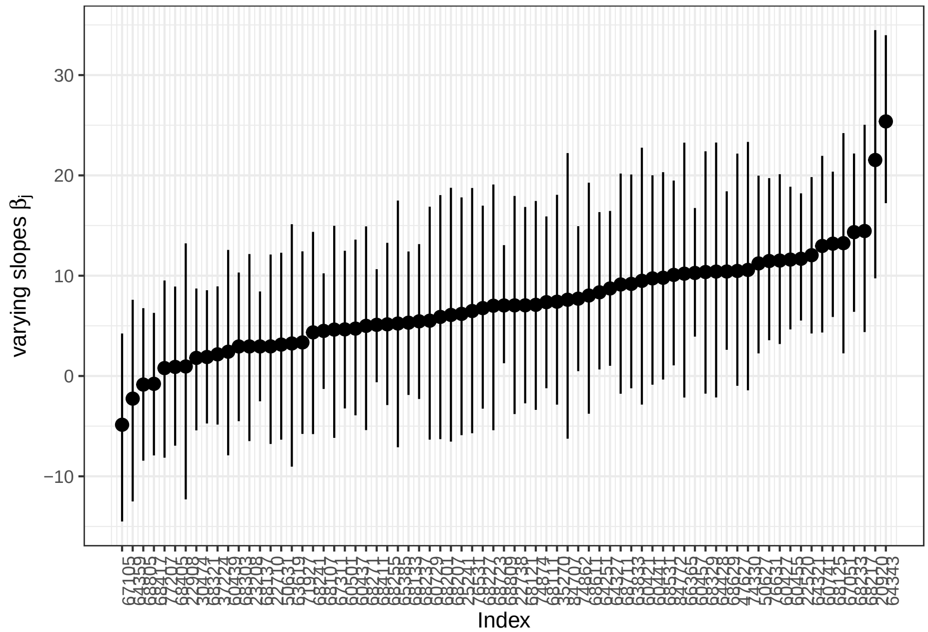
```
## Sigma[school:(Intercept),(Intercept)]  81.438  16.492  53.845 117.378
##
## MCMC diagnostics
##                                      mcse  Rhat  n_eff
## (Intercept)                          0.044 1.001  723
## femaleF                              0.009 1.000 5424
## sigma                                0.004 1.000 4301
## Sigma[school:(Intercept),(Intercept)] 0.646 1.002  653
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

The posterior estimates are $\mu_\theta = 69.730, \beta = 6.754, \tau^2 = 81.438, \sigma = 13.420$.

Ex.5

```
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
```

```r
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))
```



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

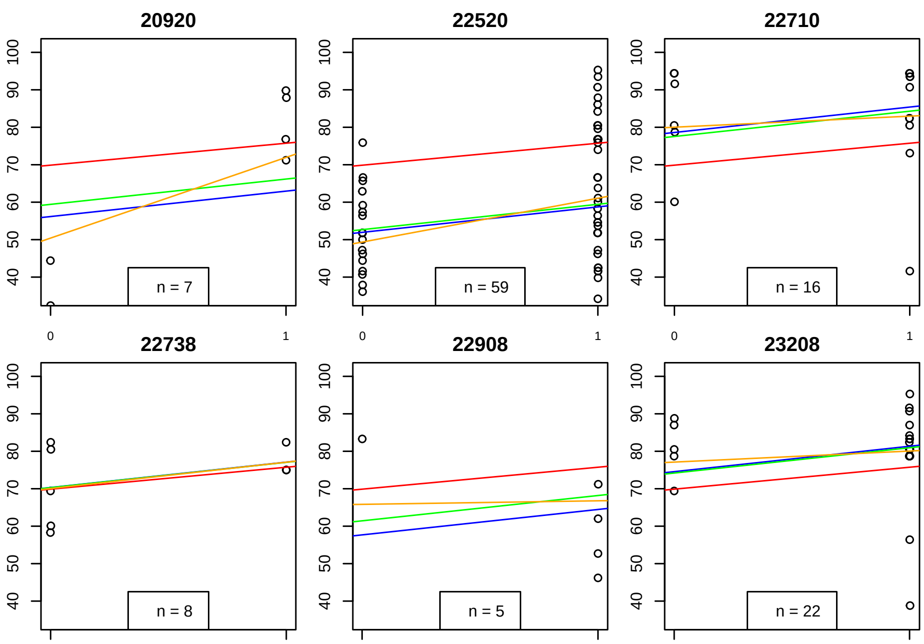
```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod2 -29.4       9.8
## mod1 -78.9      15.1
```

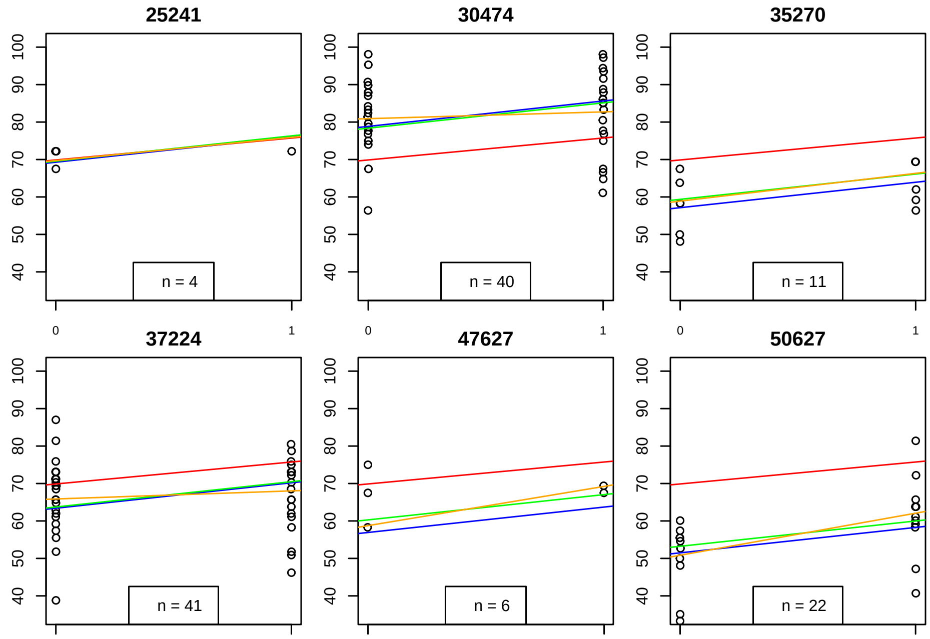
```r
loo_compare(loo1, loo3)
```

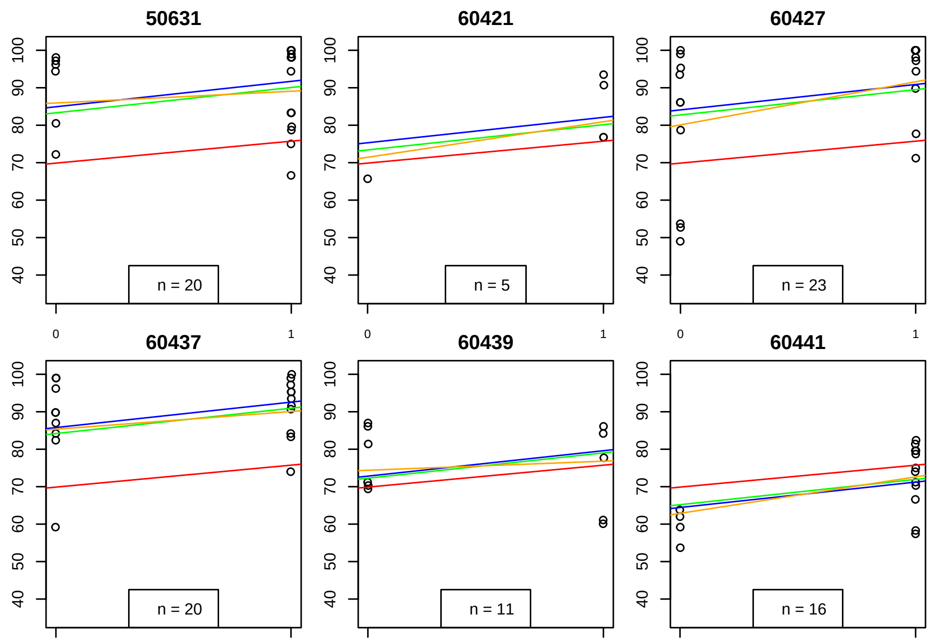
```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod1 -78.9      15.1
```

```r
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)
```

```r
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), " "))
}
```





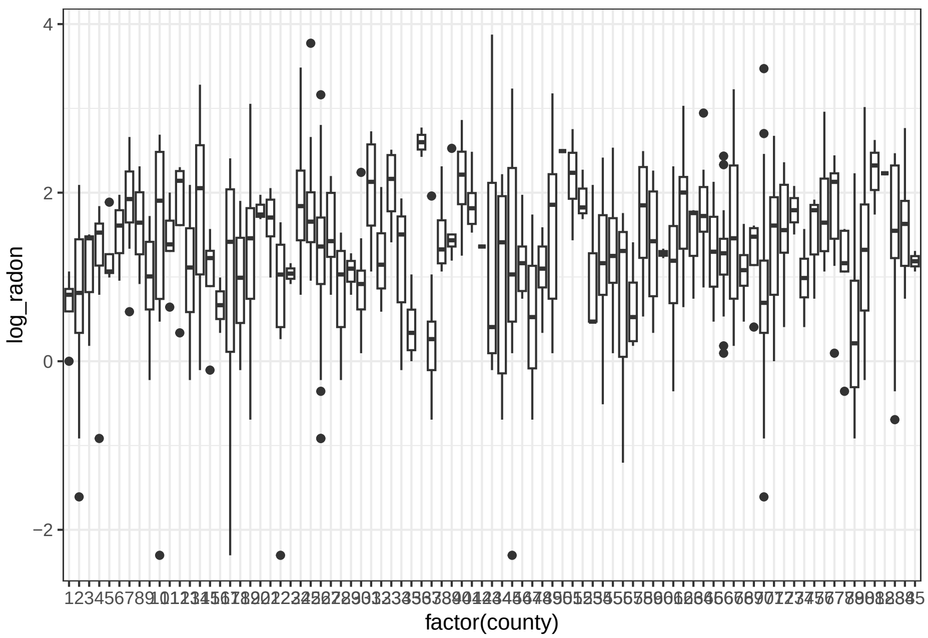


It seems that we want to prefer the random intercept and random slope model.

Ex.6

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

```r
ggplot(radon) + geom_boxplot(aes(factor(county), log_radon))
```



Yes, a hierarchical model here makes sense, with the county as the grouping variable. From the EDA we can see that the `log_radon` differs quite a lot across counties.

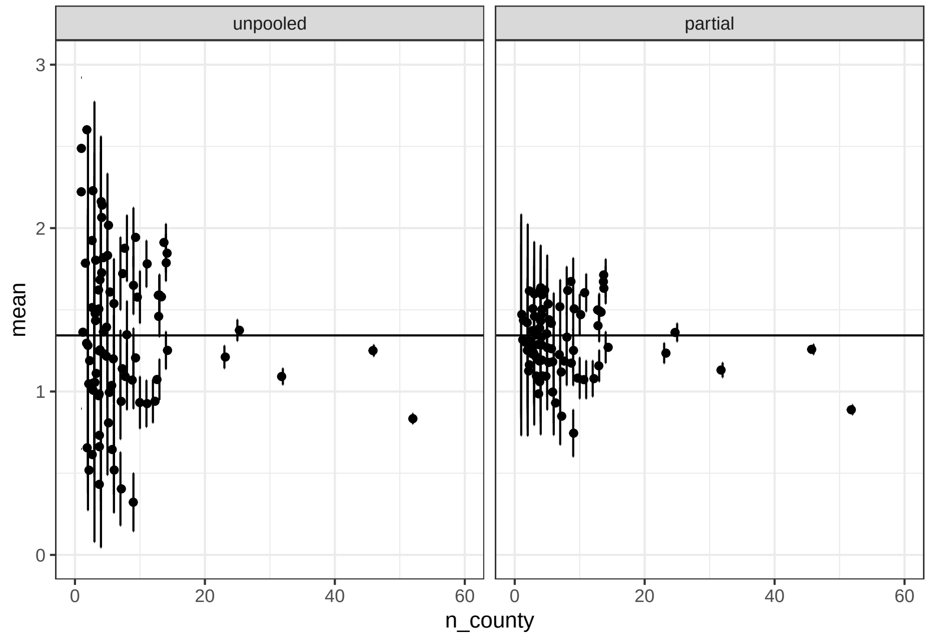
Ex.7

```r
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 = 8848,
                  refresh = 0)
```

```r
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)
}
```

```r
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")
```

```r
ggplot(rbind(unpooled.df, partial.df)%>%
         mutate(model = factor(model, levels = c("unpooled", "partial"))),
       aes(x= n_county, y = mean)) +
      geom_jitter() +
      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)
```



Ex.8

```r
radon.mod2 <- stan_lmer(formula = log_radon ~ 1 +floor + (1 | county),
                  data = radon,
                  seed = 8848,
                  refresh = 0)
radon.mod3 <- stan_lmer(formula = log_radon ~ 1 +floor + (1 + floor | county),
                  data = radon,
                  seed = 8848,
                  refresh = 0)
radon.mod4 <- stan_lmer(formula = log_radon ~ 1 +floor + log_uranium + (1 | county),
                  data = radon,
                  seed = 8848,
                  refresh = 0)
```

```r
loo_compare(
  loo(radon.unpooled),
  loo(radon.mod1),
  loo(radon.mod2),
```

```
  loo(radon.mod3),
  loo(radon.mod4)
)
```

```
##                 elpd_diff se_diff
## radon.mod4       0.0       0.0  
## radon.mod2      -9.4       5.3  
## radon.mod3     -11.0       5.7  
## radon.mod1     -56.6      11.9  
## radon.unpooled -85.1      14.3
```

According to the predictive accuracy by the difference, I'd prefer mod 4, which is a random intercept model with `floor` and `log_uranium` as the fixed effects.

Ex.9

Some groups (counties, schools, etc.) have quite small sample sizes (only 2 or 3 observations) in the data. If we purely rely on the samples from those small groups, our estimates will have very high variances due to low sample sizes. Therefore a hierarchical structure allows us to borrow information from the overall estimates (shrinkage) and trade a bit bias for reduction of variance. If there are lots of observations in a group, then the shrinkage would be very small, allowing more sufficient data to dominate the posterior estimates.