Lab09

Marc Brooks

11/2/2020

```
library(tidyverse)
library(rstanarm)
library(magrittr)
library(ggplot2)
library(mlmRev)
library(tidybayes)
library(ggstance)
library(dplyr)
library(modelr)
```

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

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

```
summary(Gcsemv)
```

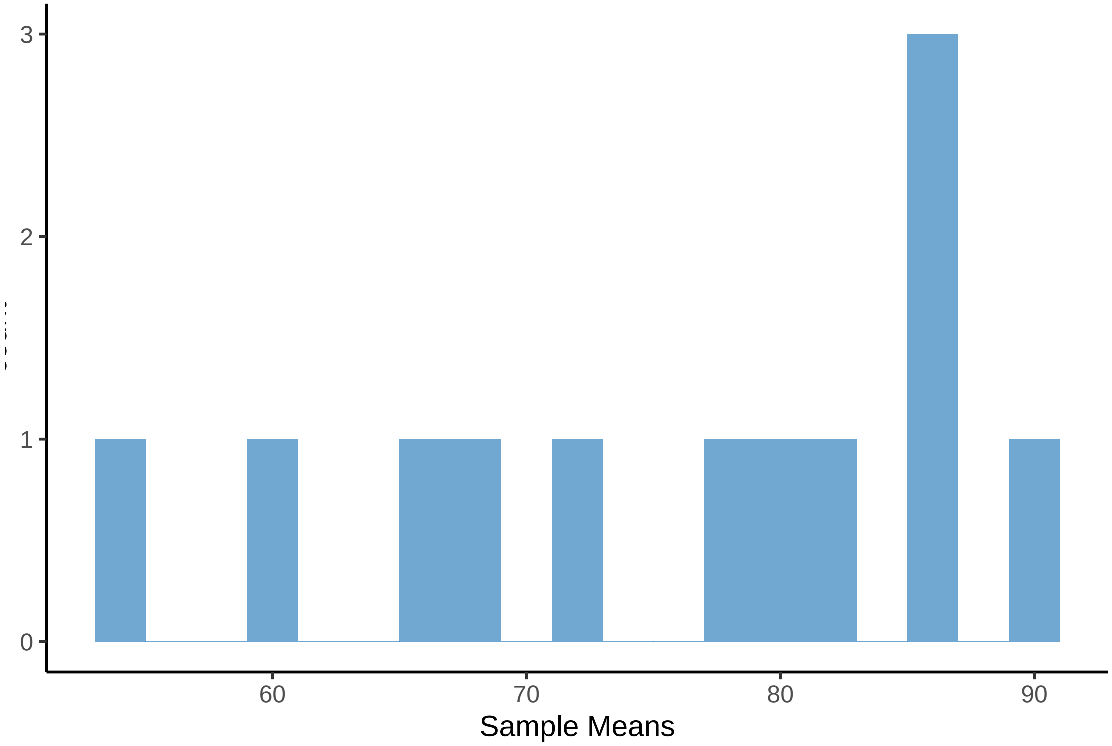
```
##      school        student     gender      written          course      
##  68137  : 104   77     :  14   F:1128   Min.   : 0.60   Min.   :  9.25  
##  68411  :  84   83     :  14   M: 777   1st Qu.:37.00   1st Qu.: 62.90  
##  68107  :  79   53     :  13            Median :46.00   Median : 75.90  
##  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 %>%
  group_by(school) %>%
  summarise(mean_score = mean(course)) %>%
```

```r
ggplot(aes(x=mean_score)) +
geom_histogram(binwidth = 2, fill = "#3182bd", alpha=.7) +
theme_classic() +
labs(x="Sample Means")
```




```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)
```

```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)                            73.704   1.139  71.531  75.935
## sigma                                  13.816   0.238  13.351  14.308
## Sigma[school:(Intercept),(Intercept)]  79.674  15.574  54.390 115.156
##
## MCMC diagnostics
##                                       mcse  Rhat  n_eff
## (Intercept)                           0.046 1.006  624
## sigma                                 0.004 0.999 4330
## Sigma[school:(Intercept),(Intercept)] 0.554 1.004  789
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Exercise 2

The posterior mean for μ_θ is 73.704 and the 95% Credible Interval is $[71.531, 75.935]$. The posterior mean for σ is 13.816 and the 95% Credible Interval is $[13.351, 14.308]$. The posterior mean for τ^2 is 79.674 and he 95% Credible Interval is $[54.390, 115.156]$.

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

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

```r
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]"
```

```r
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)]"
```

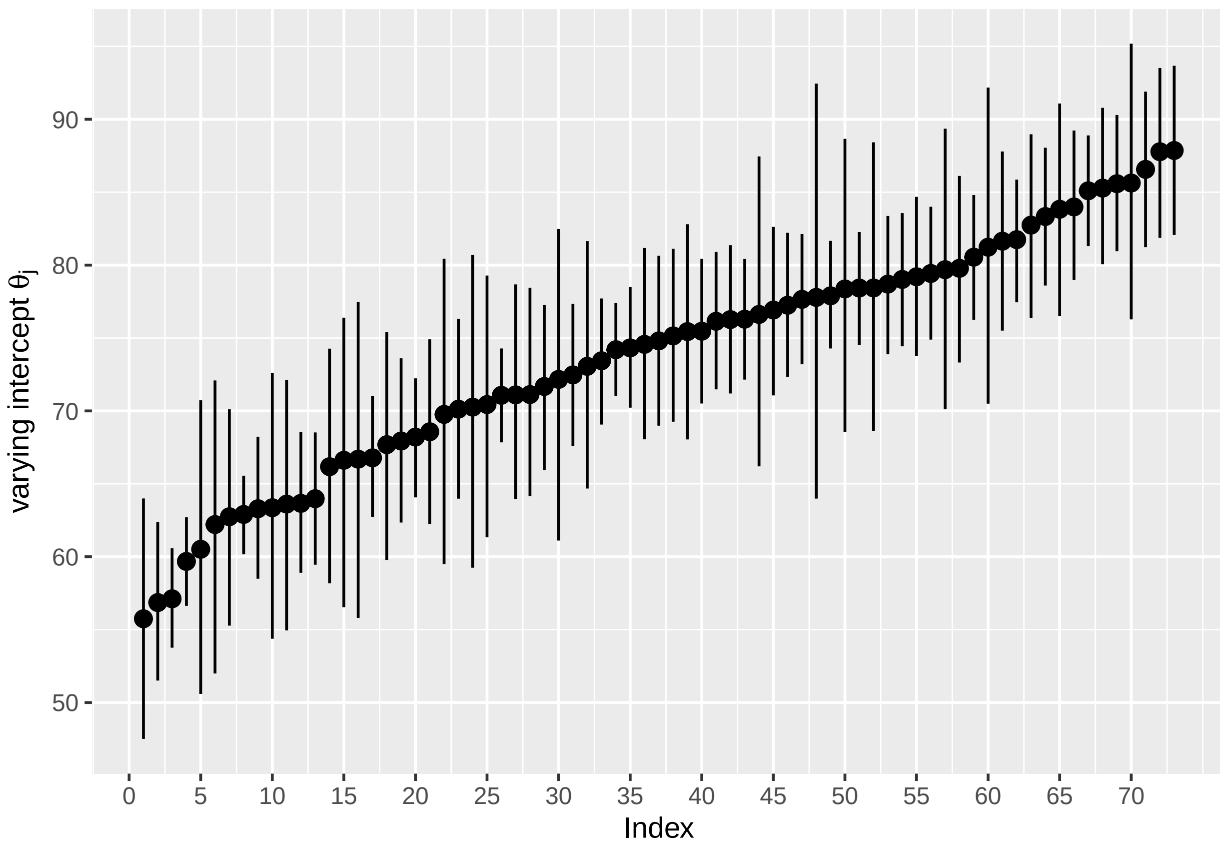
```r
# obtain draws for mu_theta
mu_theta_sims <- as.matrix(mod1, pars = "(Intercept)")
```

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

```r
# 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
int_sims <- as.numeric(mu_theta_sims) + theta_sims

# posterior mean
```

```r
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])))
```



Let's look at two schools with small sample sizes.

```r
GCSE %>%
  group_by(school) %>%
  summarise(count  = n()) %>%
```

```r
  arrange(count) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 2
##    school count
##    <fct>  <int>
##  1 84707      2
##  2 63619      4
##  3 65385      4
##  4 68201      4
##  5 25241      5
##  6 60421      5
##  7 64428      5
##  8 68207      5
##  9 22908      6
## 10 47627      6
```

```r
GCSE %>%
  group_by(school) %>%
  summarise(count  = n()) %>%
  arrange(desc(count)) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 2
##    school count
##    <fct>  <int>
##  1 68137    104
##  2 68411     84
##  3 68107     79
##  4 68809     73
##  5 22520     65
##  6 60457     54
##  7 68321     52
##  8 68125     50
##  9 68133     47
## 10 68417     47
```

```r
theta_simsComp <- as.matrix(mod1,
                        regex_pars ="b\\[\\(Intercept\\) school\\:(60421|68417)")
```

```r
GCSE %>%
  filter(school %in% c(60421,68417)) %>%
  group_by(school) %>%
  summarise(mean_score = mean(course, na.rm=T), .groups="drop")
```

```
## # A tibble: 2 x 2
##   school mean_score
##   <fct>       <dbl>
## 1 60421        80.7
## 2 68417        74.4
```

```r
apply(theta_simsComp + as.numeric(mu_theta_sims), 2, mean)
```

```
## b[(Intercept) school:60421] b[(Intercept) school:68417]
##                    78.36424                    74.32290
```

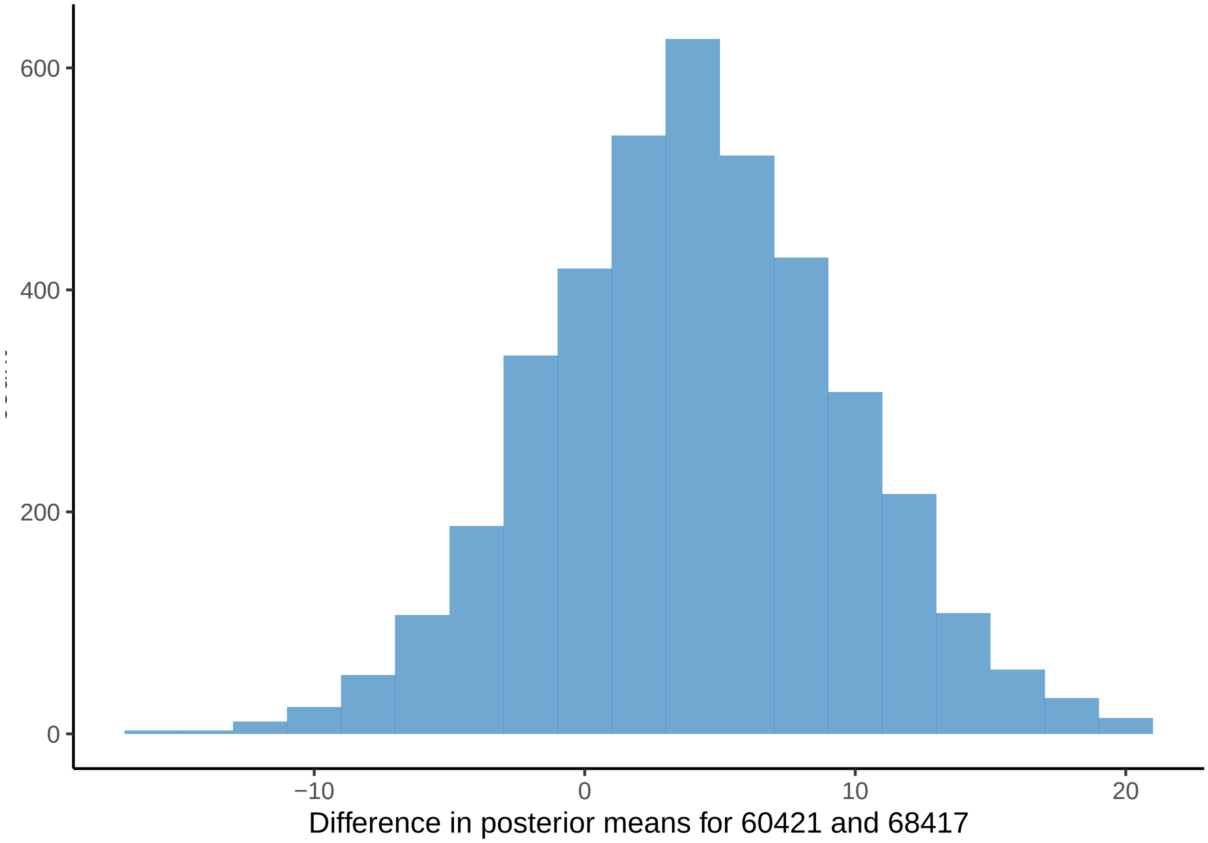
```r
print(paste("Posterior mean difference of average scores =  = ", round(mean(theta_simsComp[,1] - theta_s
```

```
## [1] "Posterior mean difference of average scores =  =  4.0413"
```

```r
CI <- quantile(round(theta_simsComp[,1] - theta_simsComp[,2],4), c(0.025, .975))
print(paste("95% CI = ", '[',  str_c(CI, collapse=","), ']'))
```

```
## [1] "95% CI =  [ -6.8851625,15.037025 ]"
```

```r
data.frame(diff = theta_simsComp[,1] - theta_simsComp[,2]) %>%
  ggplot(aes(x=diff)) +
  geom_histogram(binwidth = 2, fill = "#3182bd", alpha=.7) +
  theme_classic() +
  labs(x="Difference in posterior means for 60421 and 68417")
```

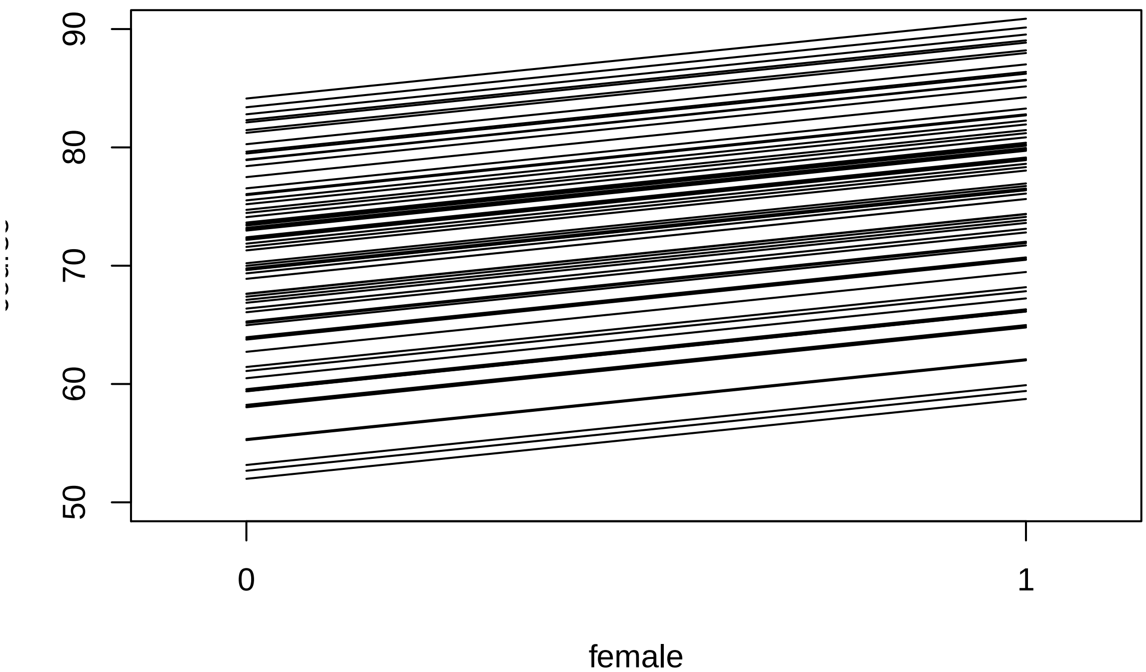


For the school with the smaller sample size, it looks like it was almost compeletly shrunk to the global mean. The school with th larger sample size differs more from the global mean. The differences between the school averages are on average smaller than the difference between their sample average and as we can see from the histogram there is a high density around 0 for the the difference of their averages.

```r
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)", "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)                          69.669   1.211  67.322  72.056
## femaleF                               6.744   0.677   5.410   8.043
## sigma                                13.424   0.236  12.965  13.905
## Sigma[school:(Intercept),(Intercept)]  80.663  16.466  54.072 118.776
##
## MCMC diagnostics
##                                     mcse  Rhat  n_eff
## (Intercept)                         0.049 1.000  612
## femaleF                             0.009 1.000 5273
## sigma                               0.003 0.999 5127
## Sigma[school:(Intercept),(Intercept)] 0.615 1.003  717
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

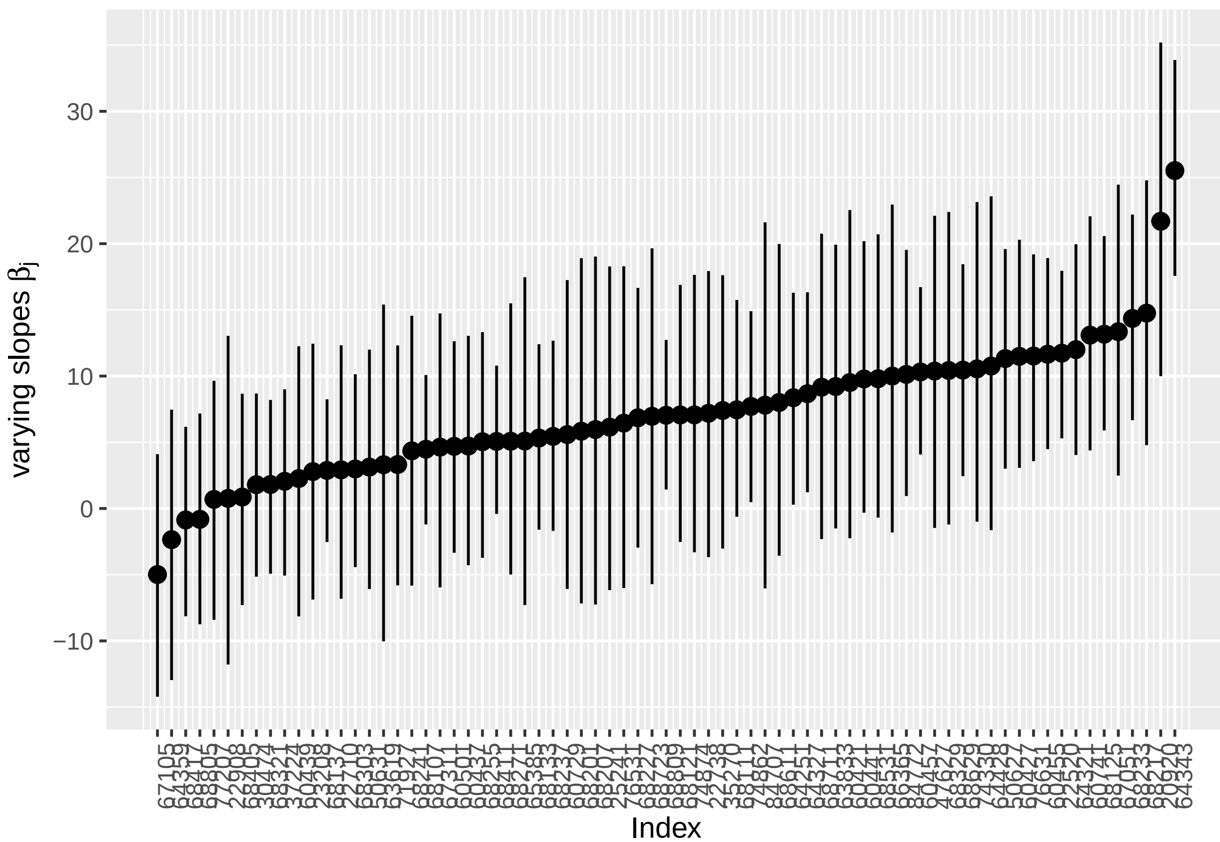
The posterior mean for μ_θ is 69.669 and the 95% Credible Interval is $[67.322, 72.056]$. The posterior mean for β is 6.744 and the 95% Credible Interval is $[5.410, 8.043]$. The posterior mean for σ is 13.424 and the 95% Credible Interval is $[12.965, 13.905]$. The posterior mean for τ^2 is 80.663 and he 95% Credible Interval is $[54.072, 118.776]$.

Model 3

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

mod3_sims <- as.matrix(mod3)
```

```r
# 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
theta_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) + theta_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))
```

```r
# 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))
```



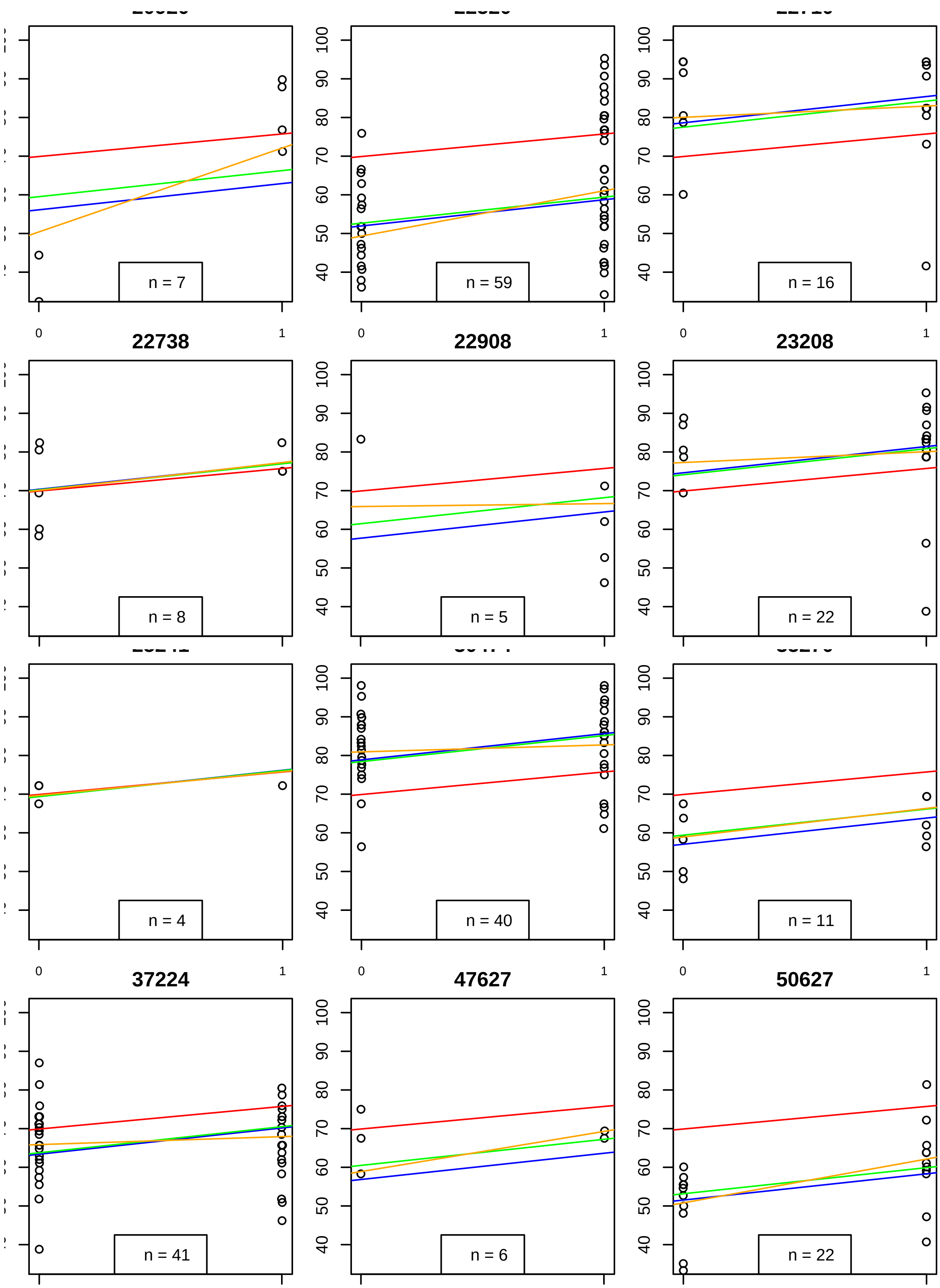
Model Comparison

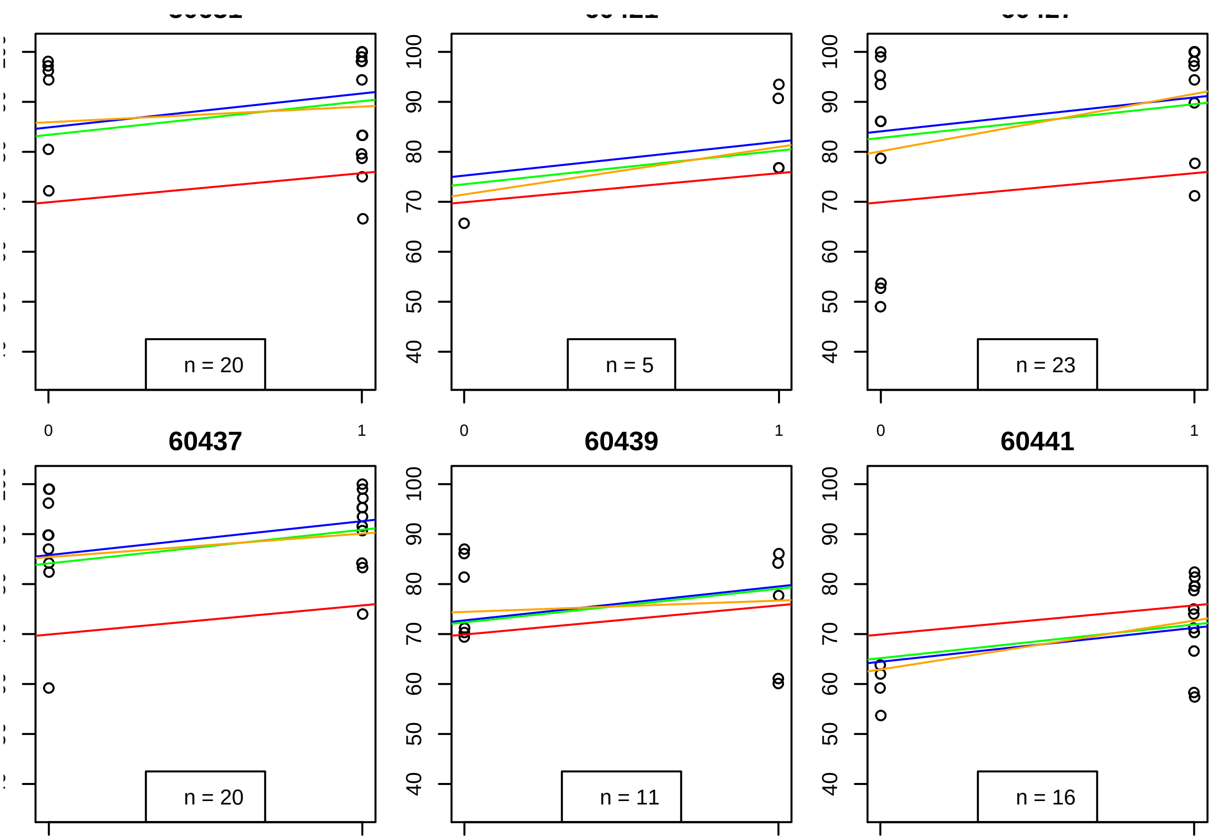
```r
loo1 <- loo(mod1)
loo2 <- loo(mod2)
loo3 <- loo(mod3)
```

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

```
##      elpd_diff se_diff
## mod3   0.0       0.0
## mod2 -29.9       9.9
## mod1 -78.2      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)


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), " "))
}
```


n = 7
n = 59
n = 16
22738
22908
23208
n = 8
n = 5
n = 22
n = 4
n = 40
n = 11
37224
47627
50627
n = 41
n = 6
n = 22



Exercise 5

The red line represents the pooled model it represents one intercept and a fixed slope. It does not change across schools. The blue line represents our second model, which has a varying intercept (by scools) but fixed slope. Therefore, as we can see, the blue line is parallel the red line and deviates by a constant. The green line represents the hierarchical model where the slope varies. In this case the green line is close to the blue line, but there is a pulling effect that is stronger for schools with small sample sizes. For these schools the intercept gets pulled to the global mean and we can see in the plot that for schools with small samples sizes the green line lies between the red and blue line. The yellow line represents our varying slope varying intercept model. The slope of the line changes for each school and again there is a pulling effect that will shift lines representing schools with small sizes from the blue line to the level of the red line.

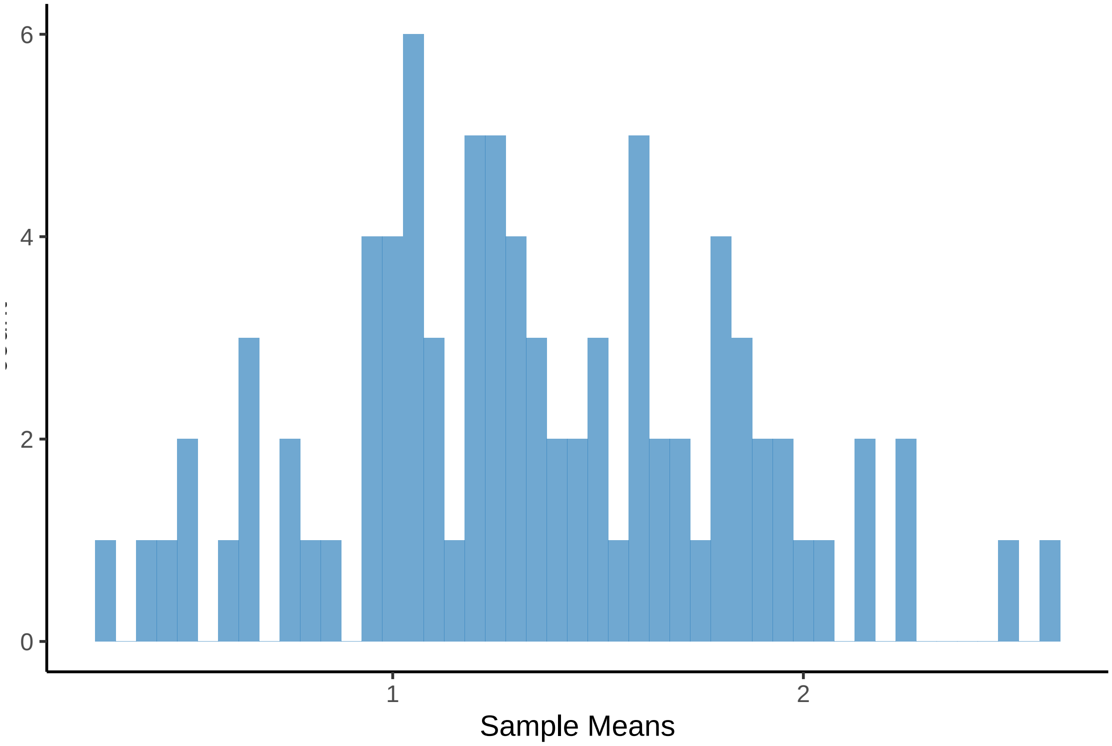
According to the model comparison I would recommend model 3, which has the varying slope and intercept with a single predictor as it seems to have the highest likelihood.

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

Exercise 6

```
radon %>%
  group_by(county) %>%
  summarise(mean_radon = mean(log_radon),
            .groups = "drop") %>%
```

```
  ggplot(aes(x=mean_radon)) +
  geom_histogram(binwidth = .05, fill = "#3182bd", alpha=.7) +
  theme_classic() +
  labs(x="Sample Means")
```



The samples means of log radon across counties are spead out between 0 and 3, indicating the the means of radon are different across counties (especially considering this is the log of radon).

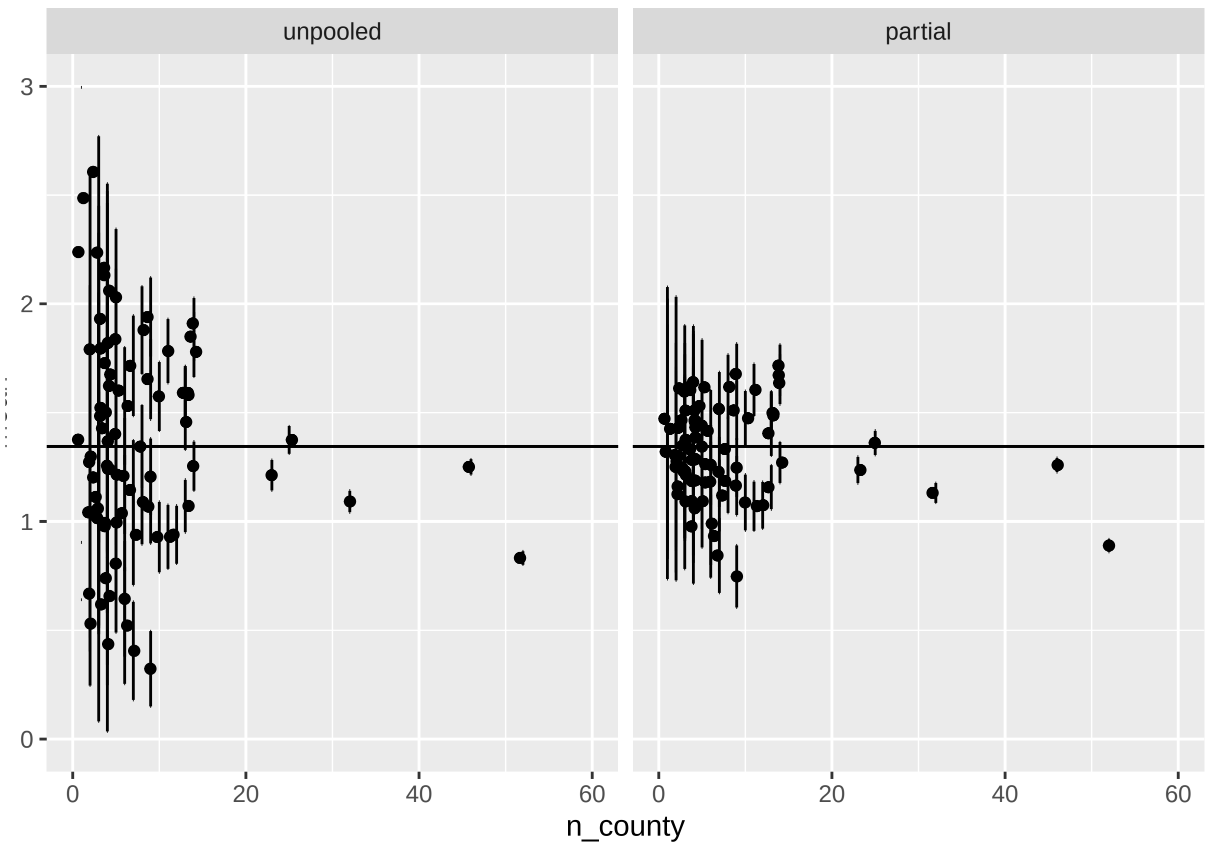
Exercise 7

```
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)
```



Exercise 8

```
radon.mod2 <- stan_lmer(formula = log_radon ~ 1 + floor + (1 | county),
                  data = radon,
                  seed = 349,
                  refresh = 0)
```

```r
radon.mod3 <- stan_lmer(formula = log_radon ~ 1 + floor + (1 + floor | county),
                  data = radon,
                  seed = 349,
                  refresh = 0)
```

```r
radon.mod4 <- stan_lmer(formula = log_radon ~ 1 + floor + log_uranium + (1 | county),
                  data = radon,
                  seed = 349,
                  refresh = 0)
```

Comparing all 5 models

```r
loo1 <- loo(radon.unpooled)
loo2 <- loo(radon.mod1)
loo3 <- loo(radon.mod2)
loo4 <- loo(radon.mod3)
loo5 <- loo(radon.mod4)
```

```r
loo_compare(loo1,loo2,loo3,loo4,loo5)
```

```
##                elpd_diff se_diff
## radon.mod4       0.0       0.0
## radon.mod2      -9.3       5.2
## radon.mod3     -11.1       5.6
## radon.mod1     -56.6      11.9
## radon.unpooled -84.7      14.2
```

According to our comparison the last model we fit with the varying slope by counties and floor / log uranium covariates has the highest likelihood and shown to be the best.