Applied Bayesian Modeling - module 7

Leontine Alkema

September 25, 2022

1 Read in radon data

Read in the radon data and process (copied from earlier module)

```
# house level data
d <- read.table(url("http://www.stat.columbia.edu/~gelman/arm/examples/radon/srrs2.dat"),
                header=T, sep=",")

# deal with zeros, select what we want, make a fips (county) variable to match on
d <- d %>%
  mutate(activity = ifelse(activity==0, 0.1, activity)) %>%
  mutate(fips = stfips * 1000 + cntyfips) %>%
  dplyr::select(fips, state, county, floor, activity)

# county level data
cty <- read.table(url("http://www.stat.columbia.edu/~gelman/arm/examples/radon/cty.dat"),
                  header = T, sep = ",")
cty <-
  cty %>%
  mutate(fips = 1000 * stfips + ctfips) %>%
  dplyr::select(fips, Uppm) %>%
  rename(ura_county = (Uppm))

dmn <- d %>%
  filter(state=="MN") %>% # Minnesota data only
  dplyr::select(fips, county, floor, activity) %>%
  left_join(cty)
```

1.1 More data processing for multilevel modeling

Some more data processing first, to produce a data set that has county info and to produce the plot with means from the slides. In the data set:

- y_i is log(activity)
- county gives county name (fips gives the unique county ID)
- x_i is floor
- u_i is log_ur = log(ura_county)

(the last two are added for module 8, when including predictors)

```r
dat <-
  dmn%>%
  mutate(y = log(activity), log_ur = log(ura_county))
head(dat)
```

```
##    fips                county floor activity ura_county         y     log_ur
## 1 27001 AITKIN                   1      2.2   0.502054 0.7884574 -0.6890476
## 2 27001 AITKIN                   0      2.2   0.502054 0.7884574 -0.6890476
## 3 27001 AITKIN                   0      2.9   0.502054 1.0647107 -0.6890476
## 4 27001 AITKIN                   0      1.0   0.502054 0.0000000 -0.6890476
## 5 27003 ANOKA                    0      3.1   0.428565 1.1314021 -0.8473129
## 6 27003 ANOKA                    0      2.5   0.428565 0.9162907 -0.8473129
```

Create summary data set with info for each county:

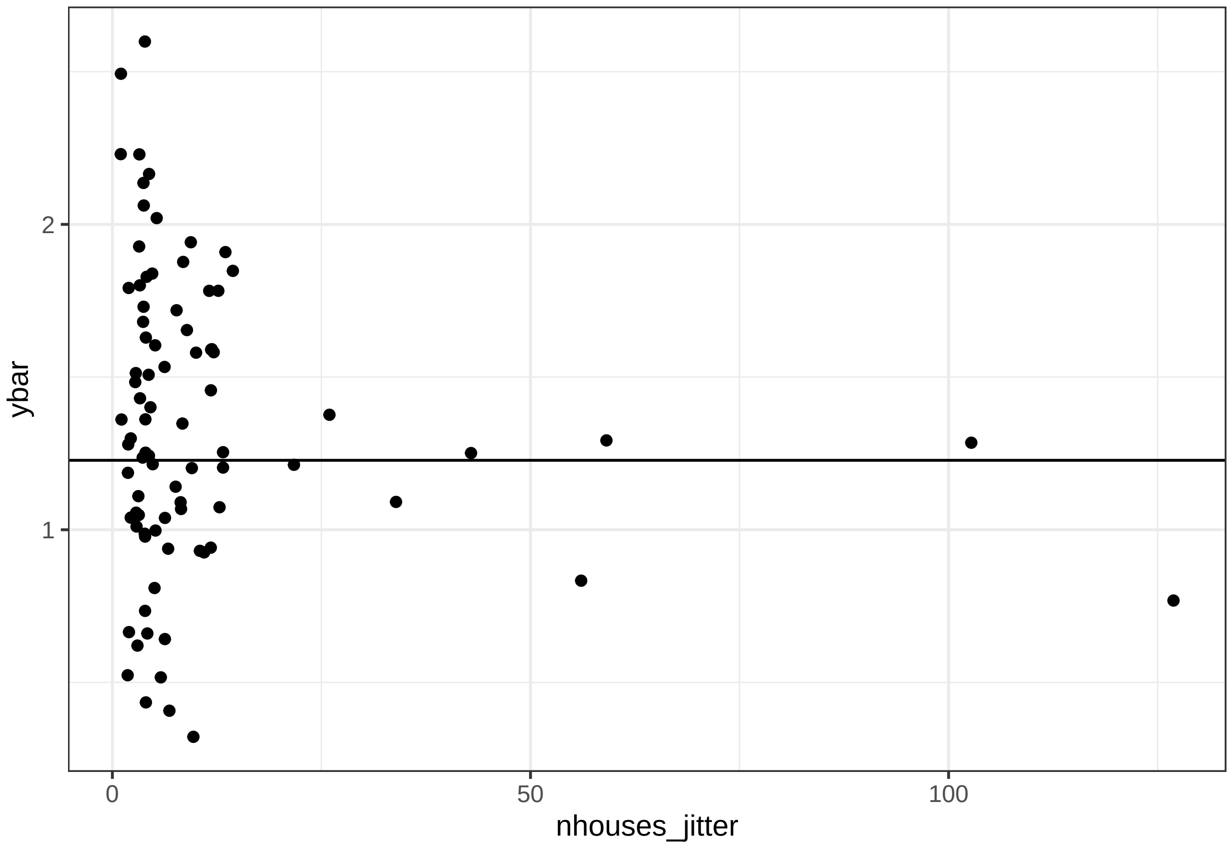
```r
# to plot observations and county means ~ sample sizes,
# easier to see if sample sizes are slighly jittered
set.seed(12345)

datcounty <- dat %>%
  group_by(fips) %>%
  summarize(nhouses = n(), ybar = mean(y), county = county[1], log_ur = log_ur[1]) %>%
  mutate(nhouses_jitter = nhouses*exp(runif (length(nhouses), -.1, .1)))
ngroups <- dim(datcounty)[1]
head(datcounty)
```

```
## # A tibble: 6 x 6
##    fips nhouses  ybar county                  log_ur nhouses_jitter
##   <dbl>   <int> <dbl> <chr>                    <dbl>          <dbl>
## 1 27001       4 0.660 "AITKIN               " -0.689           4.18
## 2 27003      52 0.833 "ANOKA                " -0.847          56.1
## 3 27005       3 1.05  "BECKER               " -0.113           3.16
## 4 27007       7 1.14  "BELTRAMI             " -0.593           7.56
## 5 27009       4 1.25  "BENTON               " -0.143           3.97
## 6 27011       3 1.51  "BIG STONE            "  0.387           2.81
```

```r
ybarbar <- mean(dat$y) # population (here state) mean
```

```r
datcounty %>%
  ggplot(aes(x = nhouses_jitter, y = ybar)) +
  geom_point() +
  geom_hline(mapping = aes(yintercept = ybarbar)) +
  theme_bw()
```



2 Model fitting

```
fit <- brm(y ~ (1|county),
        data = dat,
        iter = 1000,
        chains = 4,
        cores = getOption("mc.cores", 4))
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Users/lalkema/
## In file included from <built-in>:1:
## In file included from /Users/lalkema/Library/R/arm64/4.2/library/StanHeaders/include/stan/math/prim/
## In file included from /Users/lalkema/Library/R/arm64/4.2/library/RcppEigen/include/Eigen/Dense:1:
## In file included from /Users/lalkema/Library/R/arm64/4.2/library/RcppEigen/include/Eigen/Core:88:
## /Users/lalkema/Library/R/arm64/4.2/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:1: err
## namespace Eigen {
## ^
## /Users/lalkema/Library/R/arm64/4.2/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:16: er
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Users/lalkema/Library/R/arm64/4.2/library/StanHeaders/include/stan/math/prim/
## In file included from /Users/lalkema/Library/R/arm64/4.2/library/RcppEigen/include/Eigen/Dense:1:
```

```
## /Users/lalkema/Library/R/arm64/4.2/library/RcppEigen/include/Eigen/Core:96:10: fatal error: 'complex
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

Summary of model fit:

```
summary(fit)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ (1 | county)
##    Data: dat (Number of observations: 927)
##   Draws: 4 chains, each with iter = 1000; warmup = 500; thin = 1;
##          total post-warmup draws = 2000
##
## Group-Level Effects:
## ~county (Number of levels: 85)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.32      0.05     0.23     0.43 1.00      728     1033
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.32      0.05     1.22     1.42 1.00     1351     1544
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.81      0.02     0.77     0.85 1.00     3516     1372
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

3 Visualizing the group-level mean parameters

Coefficients can be obtained using coef(fit), you can get the help file here:

```
#?coef.brmsfit
```

Just showing some function calls here first, ie for mu_alpha:

```
fixef(fit)
```

```
##           Estimate  Est.Error     Q2.5    Q97.5
## Intercept 1.316091 0.04966739 1.215974 1.416438
```

eta = alpha - mu_alpha (as compared to notation in slides), labeled here as random effects

```
eta <- as_tibble(ranef(fit)$county[,,"Intercept"], rownames = "county")
head(eta)
```

```
## # A tibble: 6 x 5
##   county                 Estimate Est.Error   Q2.5  Q97.5
##   <chr>                     <dbl>     <dbl>  <dbl>  <dbl>
## 1 "AITKIN              "  -0.245      0.235 -0.722  0.218
## 2 "ANOKA               "  -0.425      0.113 -0.645 -0.209
## 3 "BECKER              "  -0.0833     0.261 -0.576  0.422
## 4 "BELTRAMI            "  -0.0928     0.213 -0.530  0.318
## 5 "BENTON              "  -0.0207     0.238 -0.494  0.430
## 6 "BIG STONE           "   0.0576     0.262 -0.459  0.565
```

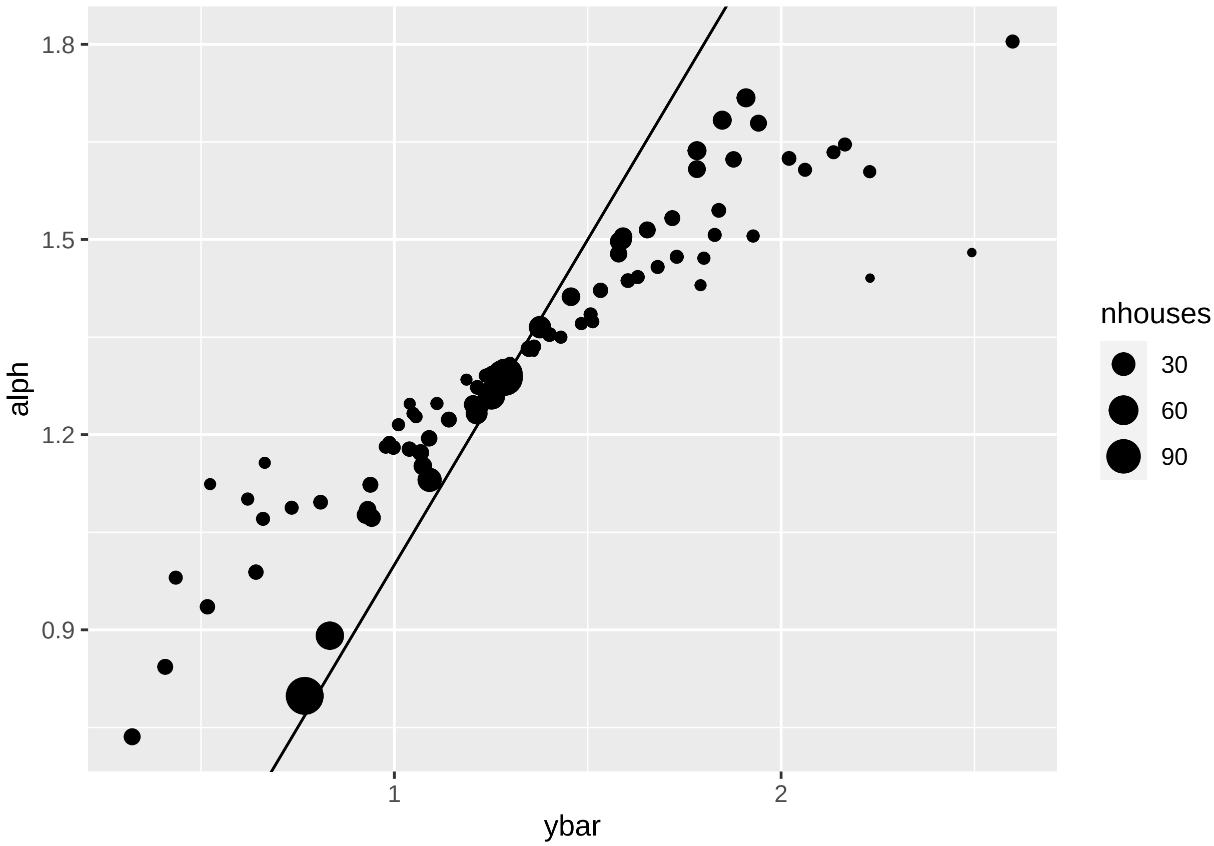
To get the alpha = eta + mu_alpha, we can use the following call

```
alphas <-
  coef(fit, summary = T)$county %>%
  as_tibble(rownames = "county") %>%
  rename(alph = Estimate.Intercept)
alphas
```

```
## # A tibble: 85 x 5
##    county                  alph Est.Error.Interc~ Q2.5.Intercept Q97.5.Intercept
##    <chr>                  <dbl>             <dbl>          <dbl>           <dbl>
##  1 "AITKIN              " 1.07              0.237          0.595            1.53
##  2 "ANOKA               " 0.891             0.104          0.689            1.10
##  3 "BECKER              " 1.23              0.263          0.723            1.75
##  4 "BELTRAMI            " 1.22              0.214          0.796            1.64
##  5 "BENTON              " 1.30              0.240          0.837            1.76
##  6 "BIG STONE           " 1.37              0.264          0.843            1.87
##  7 "BLUE EARTH          " 1.72              0.178          1.36             2.06
##  8 "BROWN               " 1.44              0.251          0.959            1.95
##  9 "CARLTON             " 1.08              0.205          0.684            1.48
## 10 "CARVER              " 1.25              0.194          0.864            1.61
## # ... with 75 more rows
```

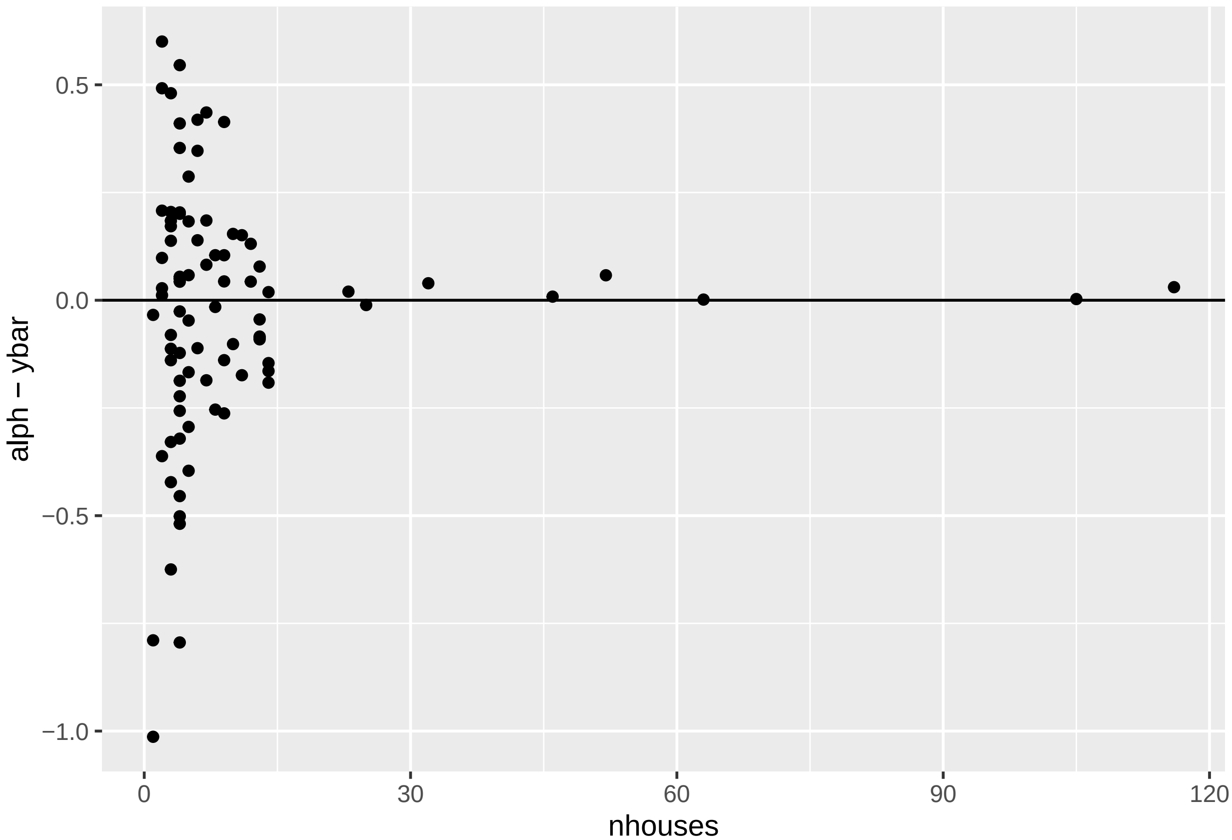
Make the plot of alpha ~ ybar

```
alphas %>%
  left_join(datcounty) %>%
  ggplot(aes(y = alph, x = ybar, size = nhouses)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0)
```



Plot of alpha - ybar

```
alphas %>%
  left_join(datcounty) %>%
  ggplot(aes(y = alph - ybar, x = nhouses)) +
  geom_point() +
  geom_hline(yintercept = 0)
```



0.5
0.0
−0.5
−1.0
alph − ybar
0
30
60
90
120
nhouses