Building up models

Sean Raleigh

Tuesday, January 27, 2015

Introduction

In this document, we build up models for the NATSAP data, starting from very simple models and gradually getting more and more complex.

Read in data

Code to read and clean the data.

```r
library(lme4)
library(rstan)
library(dplyr)
library(reshape2)
library(ggplot2)

## Import Data
natsap <- read.csv("NewNATSAP.csv")
dose <- read.csv("NATSAPDoseData.csv")

## Get rid of program with no NatsapID
dose <- dose[!is.na(dose$NatsapId),]

## Select only wanted variables and create diff
natsap_tidy <- natsap %>%
    select(ID = NatsapId,
           sex = GenderNumeric,
           admission_OQ = AdmissionTotalScore,
           discharge_OQ = DischargeTotalScore) %>%
    mutate(diff = admission_OQ - discharge_OQ)

natsap_tidy <- natsap_tidy[complete.cases(natsap_tidy),]


dose_tidy <- dose %>%
    select(rtc_vs_obh = RTCvsOBH,
           ID = NatsapId,
           minutes_ind_therapy = Mode.minutes.of.Inidividual.Therapy,
           minutes_group_therapy = Mode.minutes.of.Group.Therapy)

## Creates new program IDs incrementing from 1 for loops in Stan
## lookup is the intersection of ID from dose_tidy and natsap_tidy
natsap_tidy_ID <- select(natsap_tidy, ID)
dose_tidy_ID <- select(dose_tidy, ID)
lookup <- semi_join(dose_tidy_ID, natsap_tidy_ID)
lookup <- cbind(lookup, new_ID = 1:length(lookup$ID))
```

```
## Selects only the cases in the dataframes that have IDs in Lookup
## and adds a column including the new indices for the NatsapIds
natsap_tidy <- natsap_tidy %>%
    inner_join(lookup, by = "ID") %>%
    arrange(new_ID)
dose_tidy <- dose_tidy %>%
    inner_join(lookup, by = "ID") %>%
    arrange(new_ID)

## Add sample sizes for each program
n_by_program <- natsap_tidy %>%
    group_by(new_ID) %>%
    summarize(n = n())

dose_tidy <- cbind(dose_tidy, n = n_by_program$n)

## Defines Variables to be passed to Stan
## IPred and GPred have a column of 1's representing the constant term
n_subj <- nrow(natsap_tidy)
n_prog <- nrow(dose_tidy)
sex <- select(natsap_tidy, sex)
ind_pred <- cbind(rep(1, n_subj), sex)
minutes_ind_therapy <- select(dose_tidy, minutes_ind_therapy)
minutes_group_therapy <- select(dose_tidy, minutes_group_therapy)
rtc_vs_obh <- select(dose_tidy, rtc_vs_obh)
group_pred <- cbind(rep(1, n_prog), minutes_ind_therapy, minutes_group_therapy)
diff <- natsap_tidy$diff
ID = select(natsap_tidy, ID)


## Put data in a list for Stan
data_list <- list(n_subj = n_subj,
                  n_prog = n_prog,
                  n_ind_pred = ncol(ind_pred),
                  n_group_pred = ncol(group_pred),
                  diff = diff,
                  ID = ID,
                  ind_pred = ind_pred,
                  group_pred = group_pred)

## Wihtout having to load the arm package, we can still use the handy
## functions se.fixef and se.ranef
se.fixef <- function (object)
{
    fcoef.name <- names(fixef(object))
    corF <- vcov(object)@factors$correlation
    ses <- corF@sd
    names(ses) <- fcoef.name
    return(ses)
}

se.ranef <- function (object)
{
```

```r
    se.bygroup <- ranef(object, condVar = TRUE)
    n.groupings <- length(se.bygroup)
    for (m in 1:n.groupings) {
        vars.m <- attr(se.bygroup[[m]], "postVar")
        K <- dim(vars.m)[1]
        J <- dim(vars.m)[3]
        names.full <- dimnames(se.bygroup[[m]])
        se.bygroup[[m]] <- array(NA, c(J, K))
        for (j in 1:J) {
            se.bygroup[[m]][j, ] <- sqrt(diag(as.matrix(vars.m[,
                , j])))
        }
        dimnames(se.bygroup[[m]]) <- list(names.full[[1]], names.full[[2]])
    }
    return(se.bygroup)
}
```

Simple linear models

We need new_ID and sex to be factor variables.

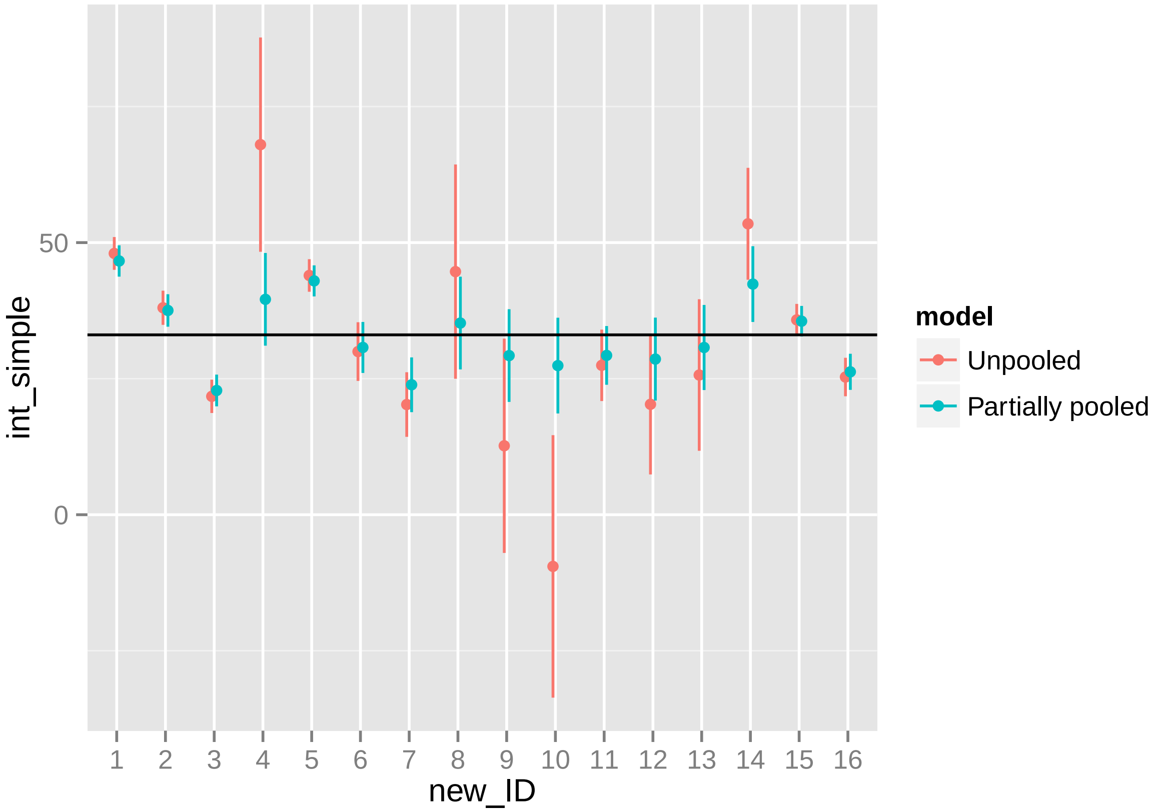
```r
natsap_tidy <- natsap_tidy %>%
    mutate(new_ID = as.factor(new_ID),
           sex = as.factor(sex))
```

First, let's only consider group differences.

```r
fit_simple1 <- lm(diff ~ 1, data = natsap_tidy)
fit_simple2 <- lm(diff ~ new_ID - 1, data = natsap_tidy)
fit_simple3 <- lmer(diff ~ (1|new_ID), data = natsap_tidy)

## Extract coeffiencts and st errors
int_simple1 <- summary(fit_simple1)$coef[,1]
int_simple2 <- data.frame(int_simple = summary(fit_simple2)$coef[,1])
se_int_simple2 <- data.frame(se_int_simple = summary(fit_simple2)$coef[,2])
int_simple3 <-
    data.frame(int_simple =
                   unname(fixef(fit_simple3)) + ranef(fit_simple3)$new_ID[,1])
se_int_simple3 <-
    data.frame(se_int_simple =
                   se.ranef(fit_simple3)$new_ID[,1])
int_simple_a <-
    cbind(new_ID = as.factor(dose_tidy$new_ID),
          model = rep("Unpooled", length(int_simple2)),
          int_simple = int_simple2,
          se_int_simple = se_int_simple2)
rownames(int_simple_a) <- NULL
int_simple_b <-
    cbind(new_ID = as.factor(dose_tidy$new_ID),
          model = rep("Partially pooled", length(int_simple3)),
          int_simple = int_simple3,
          se_int_simple = se_int_simple3)
```

```
int_simple <- rbind(int_simple_a, int_simple_b)
limits_simple <- aes(ymax = int_simple + se_int_simple,
    ymin = int_simple - se_int_simple)
ggplot(int_simple, aes(x = new_ID, y = int_simple, color = model)) +
    geom_pointrange(limits_simple, position = position_dodge(width = 0.2)) +
    geom_hline(yintercept = fixef(fit_simple3))
```



Let's look only at `diff` by `sex`. This is what Gelman and Hill call "complete pooling".

```
fit_pooled <- lm(diff ~ sex - 1, data = natsap_tidy)
summary(fit_pooled)
```

```
##
## Call:
## lm(formula = diff ~ sex - 1, data = natsap_tidy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -127.401  -24.164   -2.927   22.599  125.073
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## sex0   27.927      1.629   17.14   <2e-16 ***
## sex1   42.401      1.700   24.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.56 on 861 degrees of freedom
## Multiple R-squared:  0.5154, Adjusted R-squared:  0.5142
## F-statistic: 457.8 on 2 and 861 DF,  p-value: < 2.2e-16
```

Contrast this with no pooling.

```
fit_unpooled <- lm(diff ~ sex + new_ID - 1, data = natsap_tidy)
summary(fit_unpooled)
```

```
##
## Call:
## lm(formula = diff ~ sex + new_ID - 1, data = natsap_tidy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -128.96  -22.54   -2.00   22.32  132.53
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## sex0       39.025      4.649   8.393  < 2e-16 ***
## sex1       48.008      3.005  15.977  < 2e-16 ***
## new_ID2    -3.740      4.990  -0.749 0.453803
## new_ID3   -17.275      5.562  -3.106 0.001961 **
## new_ID4    19.992     19.855   1.007 0.314277
## new_ID5    -4.049      4.233  -0.957 0.339031
## new_ID6   -12.643      6.515  -1.940 0.052659 .
## new_ID7   -20.960      7.161  -2.927 0.003512 **
## new_ID8     5.642     20.170   0.280 0.779754
## new_ID9   -35.341     19.855  -1.780 0.075448 .
## new_ID10  -57.508     24.225  -2.374 0.017823 *
## new_ID11  -11.580      8.026  -1.443 0.149448
## new_ID12  -21.305     13.437  -1.586 0.113197
## new_ID13  -13.358     14.636  -0.913 0.361687
## new_ID14   12.797     11.069   1.156 0.247956
## new_ID15   -5.420      4.977  -1.089 0.276467
## new_ID16  -18.553      4.914  -3.776 0.000171 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.99 on 846 degrees of freedom
## Multiple R-squared:  0.5392, Adjusted R-squared:  0.5299
## F-statistic: 58.23 on 17 and 846 DF,  p-value: < 2.2e-16
```

We use `lmer` from the `lme4` package to create a varying intercept model.

```
fit_vint <- lmer(diff ~ sex + (1 | new_ID), data = natsap_tidy)
summary(fit_vint)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: diff ~ sex + (1 | new_ID)
##    Data: natsap_tidy
##
## REML criterion at convergence: 8543.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7631 -0.6638 -0.0647  0.6633  3.8548
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  new_ID   (Intercept)   47.93   6.923
##  Residual             1160.66  34.068
## Number of obs: 863, groups:  new_ID, 16
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   28.908      2.815  10.268
## sex1          10.251      3.147   3.257
##
## Correlation of Fixed Effects:
##      (Intr)
## sex1 -0.480
```

```
coef(fit_vint)
```

```
## $new_ID
##    (Intercept)     sex1
## 1     36.34895 10.25091
## 2     33.87791 10.25091
## 3     22.91950 10.25091
## 4     32.08647 10.25091
## 5     32.95377 10.25091
## 6     27.01840 10.25091
## 7     22.47676 10.25091
## 8     30.64451 10.25091
## 9     25.98757 10.25091
## 10    25.19569 10.25091
## 11    28.13625 10.25091
## 12    26.31753 10.25091
## 13    28.26403 10.25091
## 14    35.99240 10.25091
## 15    32.62698 10.25091
## 16    21.67361 10.25091
##
## attr(,"class")
## [1] "coef.mer"
```

```
fixef(fit_vint)
```

```
## (Intercept)        sex1
##    28.90752    10.25091
```

```
ranef(fit_vint)
```

```
## $new_ID
##    (Intercept)
## 1    7.4414340
## 2    4.9703946
## 3   -5.9880203
## 4    3.1789487
## 5    4.0462530
## 6   -1.8891183
## 7   -6.4307621
## 8    1.7369903
## 9   -2.9199515
## 10  -3.7118311
## 11  -0.7712724
## 12  -2.5899934
## 13  -0.6434942
## 14   7.0848797
## 15   3.7194570
## 16  -7.2339141
```

Hierarchical models

Now we add a group-level predictor.

```
## We need to grab the minutes of individual and group therapy for each individual
## as well as the value of rtc_vs_obh
minutes_ind_therapy_full <- minutes_ind_therapy[natsap_tidy$new_ID,]
minutes_group_therapy_full <- minutes_group_therapy[natsap_tidy$new_ID,]
rtc_vs_obh_full <- rtc_vs_obh[natsap_tidy$new_ID,]


## The model for individual therapy
fit_hier_vint_ind <- lmer(diff ~ sex + minutes_ind_therapy_full + (1 | new_ID),
    data = natsap_tidy)
summary(fit_hier_vint_ind)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: diff ~ sex + minutes_ind_therapy_full + (1 | new_ID)
##    Data: natsap_tidy
##
## REML criterion at convergence: 8543
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7689 -0.6642 -0.0800  0.6597  3.8790
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  new_ID   (Intercept)   41.38   6.433
##  Residual             1158.80  34.041
## Number of obs: 863, groups:  new_ID, 16
```

```
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)               11.4863     9.7812   1.174
## sex1                       9.8440     3.1264   3.149
## minutes_ind_therapy_full   0.2101     0.1136   1.849
##
## Correlation of Fixed Effects:
##             (Intr) sex1
## sex1        -0.037
## mnts_nd_th_ -0.961 -0.104
```

```
coef(fit_hier_vint_ind)
```

```
## $new_ID
##    (Intercept)    sex1 minutes_ind_therapy_full
## 1    12.692015 9.84404                0.2100665
## 2    20.321421 9.84404                0.2100665
## 3     4.436198 9.84404                0.2100665
## 4    14.782601 9.84404                0.2100665
## 5    14.548760 9.84404                0.2100665
## 6    12.631598 9.84404                0.2100665
## 7     4.704809 9.84404                0.2100665
## 8    12.257755 9.84404                0.2100665
## 9     8.818625 9.84404                0.2100665
## 10    8.591075 9.84404                0.2100665
## 11   10.039295 9.84404                0.2100665
## 12   10.163037 9.84404                0.2100665
## 13   11.764479 9.84404                0.2100665
## 14   15.708342 9.84404                0.2100665
## 15   13.972065 9.84404                0.2100665
## 16    8.349225 9.84404                0.2100665
##
## attr(,"class")
## [1] "coef.mer"
```

```
fixef(fit_hier_vint_ind)
```

```
##              (Intercept)                     sex1 minutes_ind_therapy_full
##               11.4863312                9.8440396                0.2100665
```

```
ranef(fit_hier_vint_ind)
```

```
## $new_ID
##    (Intercept)
## 1    1.2056835
## 2    8.8350894
## 3   -7.0501332
## 4    3.2962695
## 5    3.0624292
## 6    1.1452673
## 7   -6.7815226
## 8    0.7714235
## 9   -2.6677063
## 10  -2.8952561
## 11  -1.4470364
## 12  -1.3232937
## 13   0.2781475
## 14   4.2220106
## 15   2.4857339
## 16  -3.1371062
```

```
## The model for group therapy
fit_hier_vint_group <- lmer(diff ~ sex + minutes_group_therapy_full + (1 | new_ID),
    data = natsap_tidy)
summary(fit_hier_vint_group)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: diff ~ sex + minutes_group_therapy_full + (1 | new_ID)
##    Data: natsap_tidy
##
## REML criterion at convergence: 8549.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7945 -0.6550 -0.0777  0.6561  3.8525
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  new_ID   (Intercept)   47.67   6.904
##  Residual             1160.05  34.060
## Number of obs: 863, groups:  new_ID, 16
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                26.03484    3.66702   7.100
## sex1                        9.17485    3.26793   2.808
## minutes_group_therapy_full  0.01301    0.01067   1.219
##
## Correlation of Fixed Effects:
##             (Intr) sex1
## sex1        -0.180
## mnts_grp_t_ -0.642 -0.272
```

```
coef(fit_hier_vint_group)
```

```
## $new_ID
##    (Intercept)     sex1 minutes_group_therapy_full
## 1     32.19759 9.174851                 0.01300868
## 2     33.65476 9.174851                 0.01300868
## 3     19.84313 9.174851                 0.01300868
## 4     29.24788 9.174851                 0.01300868
## 5     24.52858 9.174851                 0.01300868
## 6     25.23230 9.174851                 0.01300868
## 7     20.74705 9.174851                 0.01300868
## 8     27.82258 9.174851                 0.01300868
## 9     23.17545 9.174851                 0.01300868
## 10    22.28396 9.174851                 0.01300868
## 11    25.95519 9.174851                 0.01300868
## 12    23.64197 9.174851                 0.01300868
## 13    24.41831 9.174851                 0.01300868
## 14    32.10732 9.174851                 0.01300868
## 15    31.08569 9.174851                 0.01300868
## 16    20.61569 9.174851                 0.01300868
##
## attr(,"class")
## [1] "coef.mer"
```

```
fixef(fit_hier_vint_group)
```

```
##                 (Intercept)                       sex1
##                 26.03484110                 9.17485096
## minutes_group_therapy_full
##                  0.01300868
```

```
ranef(fit_hier_vint_group)
```

```
## $new_ID
##    (Intercept)
## 1   6.16275086
## 2   7.61992062
## 3  -6.19171601
## 4   3.21304313
## 5  -1.50626568
## 6  -0.80254248
## 7  -5.28779125
## 8   1.78773763
## 9  -2.85938662
## 10 -3.75087803
## 11 -0.07964723
## 12 -2.39286836
## 13 -1.61653492
## 14  6.07248049
## 15  5.05085118
## 16 -5.41915332
```

```
## The model for rtc_vs_obh
fit_hier_vint_rtc_vs_obh <-
    lmer(diff ~ sex + rtc_vs_obh_full + (1 | new_ID),
    data = natsap_tidy)
summary(fit_hier_vint_rtc_vs_obh)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: diff ~ sex + rtc_vs_obh_full + (1 | new_ID)
##    Data: natsap_tidy
##
## REML criterion at convergence: 8538.4
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7707 -0.6643 -0.0708  0.6585  3.8636
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  new_ID   (Intercept)   55.09   7.422
##  Residual             1160.24  34.062
## Number of obs: 863, groups:  new_ID, 16
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)          27.371      3.966   6.902
## sex1                  9.779      3.219   3.038
## rtc_vs_obh_fullRTC    3.091      5.273   0.586
##
## Correlation of Fixed Effects:
##             (Intr) sex1
## sex1        -0.232
## rtc_vs__RTC -0.675 -0.158
```

```r
coef(fit_hier_vint_rtc_vs_obh)
```

```
## $new_ID
##    (Intercept)     sex1 rtc_vs_obh_fullRTC
## 1     34.04058 9.779137           3.090692
## 2     33.88019 9.779137           3.090692
## 3     19.92421 9.779137           3.090692
## 4     30.83223 9.779137           3.090692
## 5     30.57035 9.779137           3.090692
## 6     24.48985 9.779137           3.090692
## 7     21.57263 9.779137           3.090692
## 8     29.14231 9.779137           3.090692
## 9     23.93332 9.779137           3.090692
## 10    23.05755 9.779137           3.090692
## 11    27.41238 9.779137           3.090692
## 12    24.13584 9.779137           3.090692
## 13    26.30809 9.779137           3.090692
## 14    34.64973 9.779137           3.090692
## 15    32.59463 9.779137           3.090692
## 16    21.39657 9.779137           3.090692
##
## attr(,"class")
## [1] "coef.mer"
```

```r
fixef(fit_hier_vint_rtc_vs_obh)
```

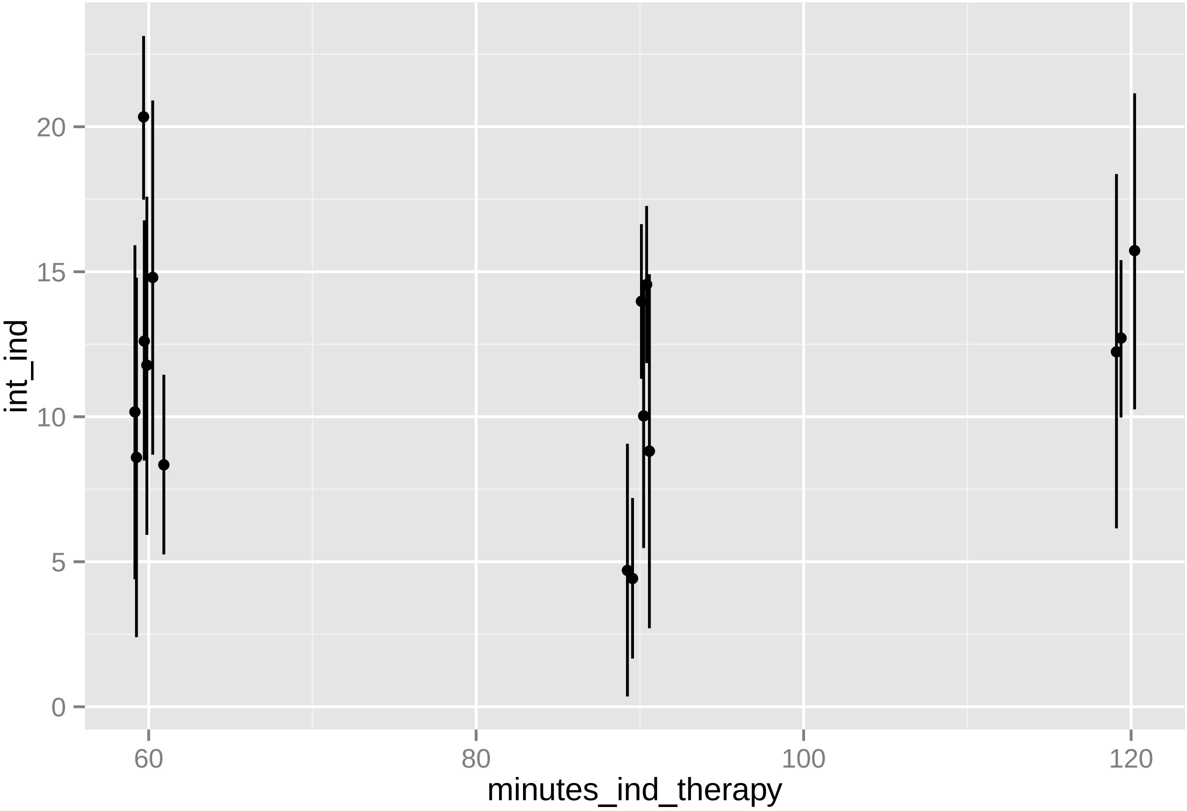
```
##        (Intercept)               sex1 rtc_vs_obh_fullRTC
##          27.371280           9.779137           3.090692
```

```
ranef(fit_hier_vint_rtc_vs_obh)
```

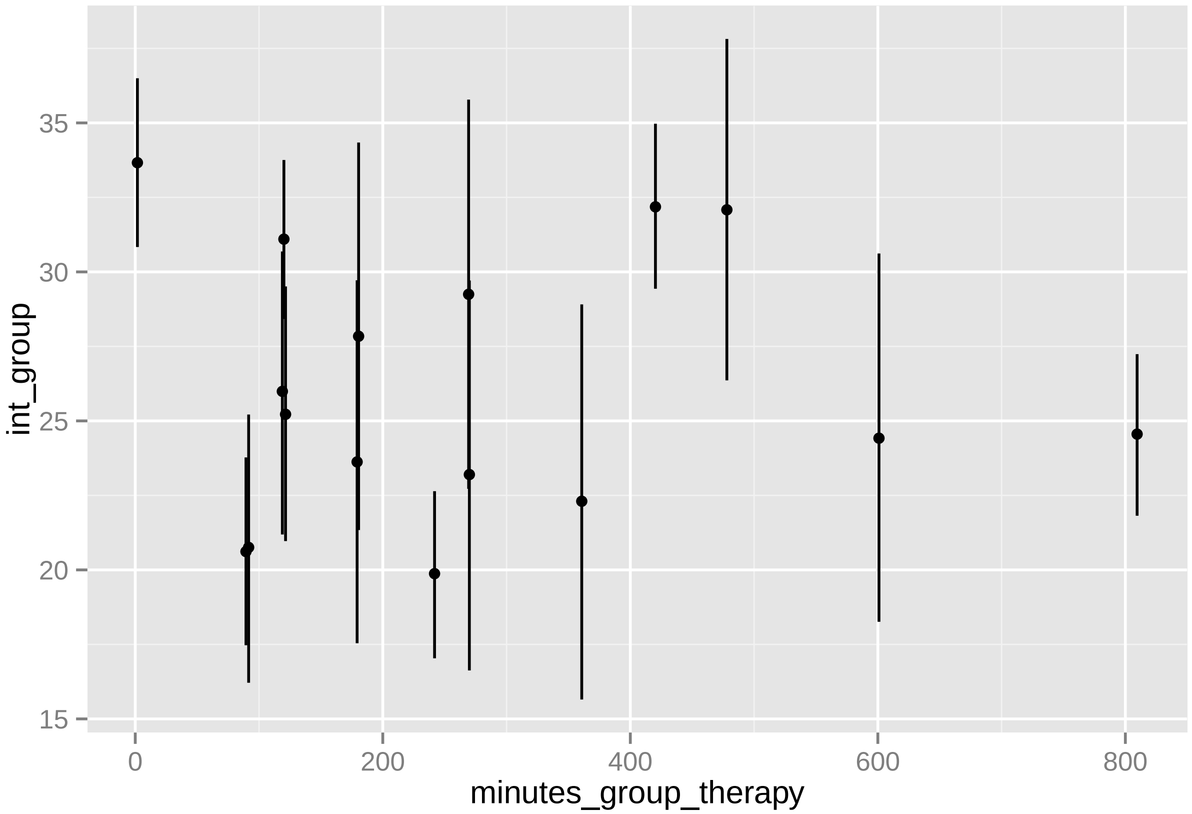
```
## $new_ID
##    (Intercept)
## 1   6.66930227
## 2   6.50890624
## 3  -7.44706594
## 4   3.46095417
## 5   3.19907154
## 6  -2.88142799
## 7  -5.79865148
## 8   1.77102889
## 9  -3.43795740
## 10 -4.31372934
## 11  0.04110205
## 12 -3.23543706
## 13 -1.06319122
## 14  7.27845209
## 15  5.22335270
## 16 -5.97470951
```

Let's try to plot something.

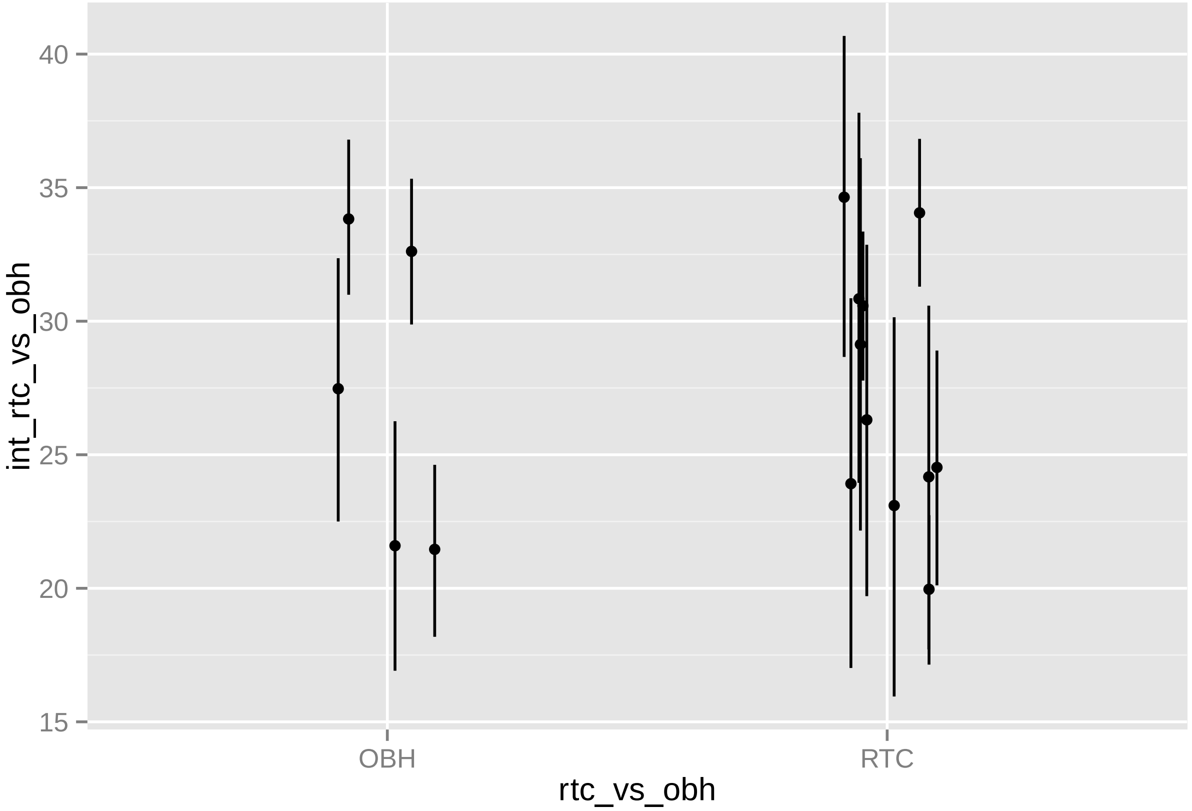
```
## Extract coefficients for minutes of individual therapy
int_ind <-coef(fit_hier_vint_ind)$new_ID[,1]
se_int_ind <- se.ranef(fit_hier_vint_ind)$new_ID[,1]
int_by_ind <-
    data.frame(dose_tidy$new_ID, minutes_ind_therapy,int_ind, se_int_ind)
limits_ind <- aes(ymax = int_ind + se_int_ind, ymin = int_ind - se_int_ind)
ggplot(int_by_ind, aes(x = minutes_ind_therapy, y = int_ind)) +
    geom_pointrange(limits_ind, position = position_jitter(width = 1))
```



```
## Extract coefficients for minutes of group therapy
int_group <-coef(fit_hier_vint_group)$new_ID[,1]
se_int_group <- se.ranef(fit_hier_vint_group)$new_ID[,1]
int_by_group <-
    data.frame(dose_tidy$new_ID, minutes_group_therapy,int_group, se_int_group)
limits_group <- aes(ymax = int_group + se_int_group, ymin = int_group - se_int_group)
ggplot(int_by_group, aes(x = minutes_group_therapy, y = int_group)) +
    geom_pointrange(limits_group, position = position_jitter(width = 2))
```

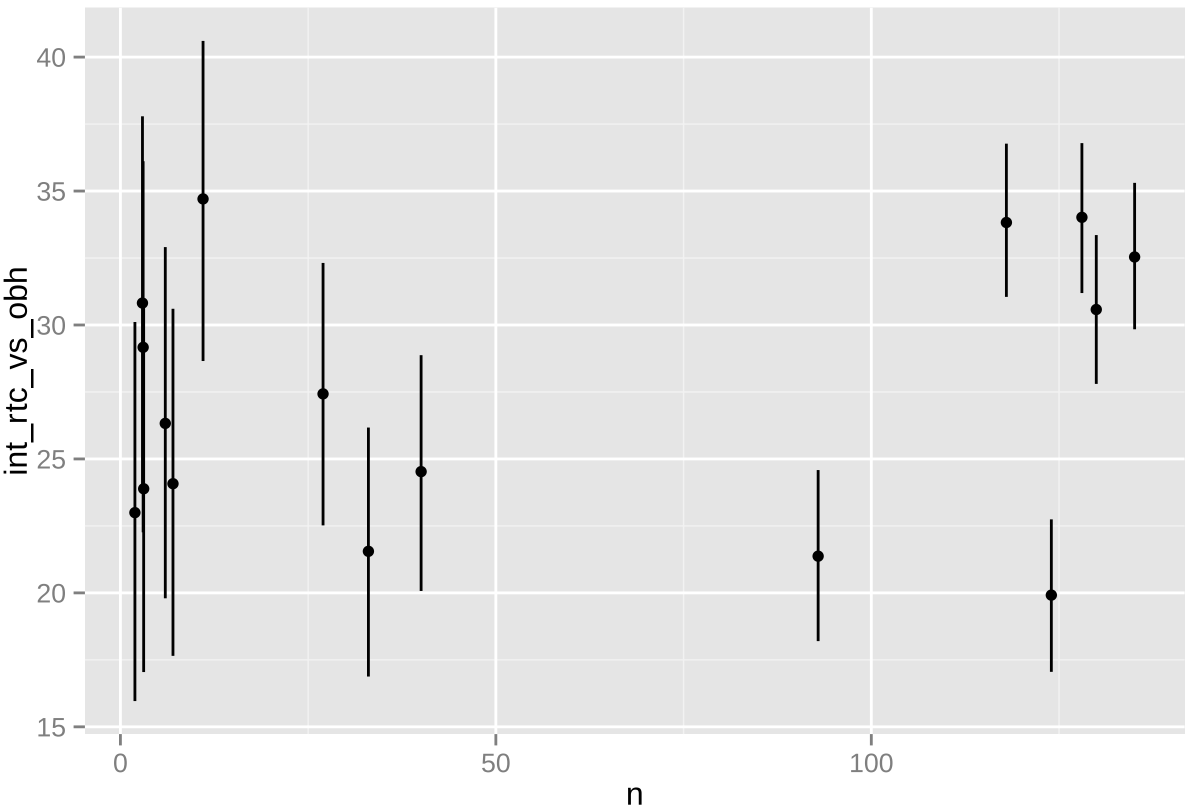


```
## Extract coefficients for rtc_vs_obh
int_rtc_vs_obh <-coef(fit_hier_vint_rtc_vs_obh)$new_ID[,1]
se_int_rtc_vs_obh <- se.ranef(fit_hier_vint_rtc_vs_obh)$new_ID[,1]
int_by_rtc_vs_obh <-
    data.frame(dose_tidy$new_ID, rtc_vs_obh,int_rtc_vs_obh, se_int_rtc_vs_obh)
limits_rtc_vs_obh <- aes(ymax = int_rtc_vs_obh + se_int_rtc_vs_obh,
    ymin = int_rtc_vs_obh - se_int_rtc_vs_obh)
ggplot(int_by_rtc_vs_obh, aes(x = rtc_vs_obh, y = int_rtc_vs_obh)) +
    geom_pointrange(limits_rtc_vs_obh, position = position_jitter(width = 0.1))
```



This all looks fine, but we perform a check to make sure that errors bars really are correlated with sample size.

```
## Sample size check
sample_check <- data.frame(dose_tidy$new_ID, n = dose_tidy$n,
                           rtc_vs_obh,int_rtc_vs_obh, se_int_rtc_vs_obh)
ggplot(sample_check, aes(x = n, y = int_rtc_vs_obh)) +
    geom_pointrange(limits_rtc_vs_obh, position = position_jitter(width = 0.1))
```


int_rtc_vs_obh
40
35
30
25
20
15
0
50
100
n