Lab 1: Mixed Models

Georges Monette

Lab 1: Mixed Models

ICPSR Summer Program at York University

Georges Monette

May 2018

- Mixed Models Exercises Lab 1 —-
  - Setup —-

- Introduction to Lab 1 —-
  - Data: Math Achievement and SES —-
    - Exploring data —-
  - Data: Levels and structure —-
- Selecting a random subset of clusters —-
- First look at variables —-
  - Looking at Level 2 variables (invariant within schools) —-
  - Creating additional Level 2 (and Level 1) variables with 'capply' —-
  - Tranformations of Level 1 variables within groups —-
- Looking at data in 3D —-
  - Looking at Level 1 and Level 2 data using Lattice graphics —-
  - Visualizing fitted lines in beta space —-
    - Looking at between group effect —-
  - Fitting a mixed model —-
  - Convergence problems —-
    - In passing: handling NAs —-
  - Hausman Test: Is the between effect different from the within effect? —-
  - Fitting a model with a contextual mean —-
  - Role of contextual variable for ses —-
  - Interpreting the model with contextual effect —-
  - Estimating the compositional effect (between effect) —-

- Visualizing the fitted model —-
  - Effect plots with the effects package
  - Handmade prediction plot
- Plotting error bars —-
  - Estimating the gap

- EXERCISES —-
- Using CWG instead of raw SES —-
- CWG vs raw in RE model —-

- EXERCISES —-
- REML vs ML —-
- Notes on testing: ML vs REML —-

- Diagnostics —-
- Diagnostics with Level 1 residuals —-
- Scale - location plot: —-
- Diagnostics with Level 2 residuals —-
- Influence diagnostics – drop one row or one cluster at a time —-
- Exercises —-

(Updated: May 27 2018 20:44)

Mixed Models Exercises Lab 1 —-

Setup —-

To install packages visit John Fox's http://tinyurl.com/York-R-Course

If necessary, update spida2:

```
#
# devtools::install_github('gmonette/spida2')
#
# OR:
#
# On Windows
# intall.packages('http://blackwell.math.yorku.ca/R/spida2.zip', repos=NULL, typ
#
# On a Mac
# install.packages("http://blackwell.math.yorku.ca/R/spida2.tgz", repos=NULL, ty
#
library(spida2)
library(p3d)
```

```
Loading required package: rgl


Attaching package: 'rgl'

The following object is masked from 'package:spida2':

    %>%


Attaching package: 'p3d'

The following objects are masked from 'package:spida2':

    cell, center, ConjComp, dell, disp, ell, ell.conj, ellbox,
    ellplus, ellpt, ellptc, ellpts, ellptsc, elltan, elltanc,
    na.include, uv
```

```
library(car)
```

```
Loading required package: carData


Attaching package: 'car'
```

```
   The following object is masked from 'package:p3d':

       Identify3d
```

```
library(effects)
```

```
   lattice theme set by effectsTheme()
   See ?effectsTheme for details.
```

```
library(nlme)
```

```

   Attaching package: 'nlme'
```

```
   The following object is masked from 'package:spida2':

       getData
```

```
library(lattice)
library(latticeExtra)
```

```
   Loading required package: RColorBrewer
```

Introduction to Lab 1 —-

At first, it's easier to learn by following a textbook example where nothing goes wrong. However, when you analyze real data lots of things tend to go wrong – or, more positively, many interesting things happen – and you wish you had experience with more realistic examples. These examples try to do both.

The first example is a gentle analysis in which we use the basic ideas of multilevel models with mixed models. We learn how to formulate and test general linear hypotheses that allow us to ask a much wider range of questions than those accessible through standard output.

The second example, using a different model with the same data, illustrates what to do when 'everything' goes wrong, oops, I meant "when a lot of interesting things happen."

In the second example, the first model does not converge and we have to figure out what to do. We need to test random effects variance parameters that are not tested validly with the usual methods and we learn how to use simulation for this purpose.

Our analysis of BLUPS reveals problems with the model whose correction we consider.

#

The second sample analysis is quite long but it is amply annotated so that you can follow much of it on your own.

Data: Math Achievement and SES —-

Math Achievement and SES in a sample of Public and Catholic high schools

Goal: Study the relationship between SES and Mathach in Public vs Catholic schools

Exploring data —-

```
?hsfull
```

```
   starting httpd help server ... done
```

```
dim( hsfull )
```

```
   [1] 7185    9
```

```
head(hsfull)
```

```
     school mathach    ses    Sex Minority Size Sector PRACAD DISCLIM
   1   1224   5.876 -1.528 Female       No  842 Public   0.35   1.597
   2   1224  19.708 -0.588 Female       No  842 Public   0.35   1.597
   3   1224  20.349 -0.528   Male       No  842 Public   0.35   1.597
   4   1224   8.781 -0.668   Male       No  842 Public   0.35   1.597
   5   1224  17.898 -0.158   Male       No  842 Public   0.35   1.597
```

```
   6   1224   4.583  0.022   Male       No  842 Public   0.35   1.597
```

```
hsfull %>% head  # same as head(hsfull)
```

```
     school mathach    ses    Sex Minority Size Sector PRACAD DISCLIM
   1   1224   5.876 -1.528 Female       No  842 Public   0.35   1.597
   2   1224  19.708 -0.588 Female       No  842 Public   0.35   1.597
   3   1224  20.349 -0.528   Male       No  842 Public   0.35   1.597
   4   1224   8.781 -0.668   Male       No  842 Public   0.35   1.597
   5   1224  17.898 -0.158   Male       No  842 Public   0.35   1.597
   6   1224   4.583  0.022   Male       No  842 Public   0.35   1.597
```

```
hsfull %>% some  # a random selection of rows
```

```
       school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM
   285    1436   9.561  0.212   Male       No  185 Catholic   1.00  -1.141
   1148   2526   9.843  0.762 Female       No  572 Catholic   0.97  -2.154
   1425   2658  16.587  0.512 Female       No  780 Catholic   0.79  -0.961
   1496   2768  12.812  0.092   Male       No 1680   Public   0.31   0.086
   2255   3533  10.121 -0.418 Female       No 1015 Catholic   0.54   0.022
   3019   4292   7.128 -1.148   Male      Yes 1328 Catholic   0.76  -0.674
   3392   4530   8.898 -1.848 Female      Yes  435 Catholic   0.60  -0.245
   4351   6170  10.991 -0.948   Male       No 2126   Public   0.25   0.432
```

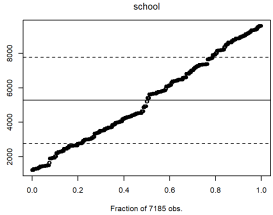
```
   4978   6990   4.756  1.292 Female      Yes 1650   Public   0.59   0.475
   5192   7276   4.027 -0.438 Female       No 1630   Public   0.30   0.109
```

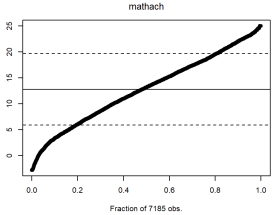
```
hsfull %>% tail
```

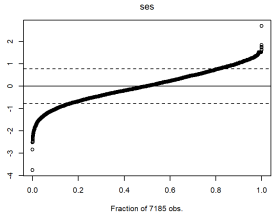
```
        school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM
   7180   9586  20.967  1.612 Female      Yes  262 Catholic      1  -2.416
   7181   9586  20.402  1.512 Female       No  262 Catholic      1  -2.416
   7182   9586  14.794 -0.038 Female       No  262 Catholic      1  -2.416
   7183   9586  19.641  1.332 Female       No  262 Catholic      1  -2.416
   7184   9586  16.241 -0.008 Female       No  262 Catholic      1  -2.416
   7185   9586  22.733  0.792 Female       No  262 Catholic      1  -2.416
```

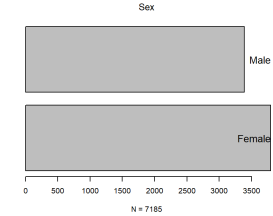
A 'uniform quantile' plot. Data is lined up from shortest to tallest.

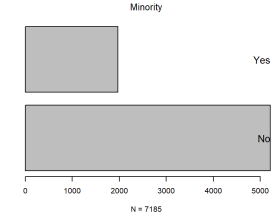
```
xqplot( hsfull )  # gives an idea of distribution, outliers, granularity, etc. a
```

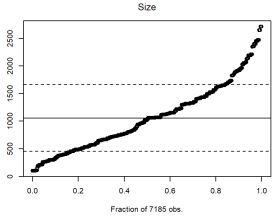

school
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 7185 obs.

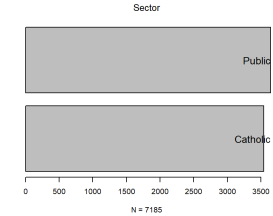

mathach
0
5
10
15
20
25
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 7185 obs.

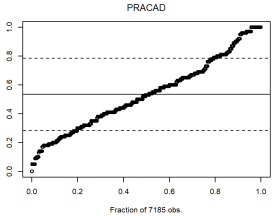

ses
-4
-3
-2
-1
0
1
2
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 7185 obs.


Sex
Male
Female
0
500
1000
1500
2000
2500
3000
3500
N = 7185


Minority
Yes
No
0
1000
2000
3000
4000
5000
N = 7185


Size
0
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 7185 obs.


Sector
Public
Catholic
0
500
1000
1500
2000
2500
3000
3500
N = 7185


PRACAD
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 7185 obs.

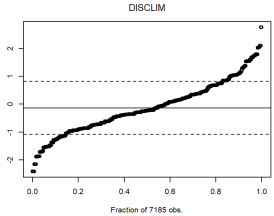

DISCLIM
-2
-1
0
1
2
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 7185 obs.

Figure: A **uniform quantile** plot. Data are lined up from shortest to tallest uniformly spaced on the x-axis. If a variable is close to uniform, its quantile plot will be close to a diagonal.

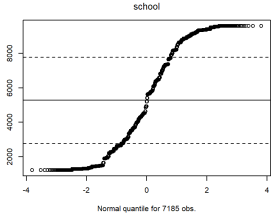
Think of a classroom photo with kids lined up from shortest to tallest

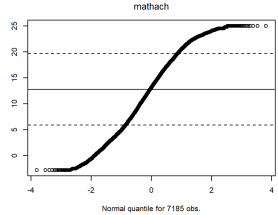
Solid line at mean, dashed lines at +/- one standard deviation

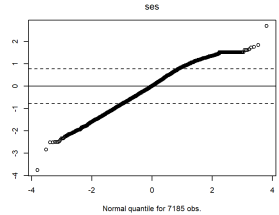
Find quantiles by selecting a proportion on x-axis and read of height of graph on y-axis (use the edge of sheet of paper) e.g. .8-quantile = 80th percentile of ses is ~ 0.8

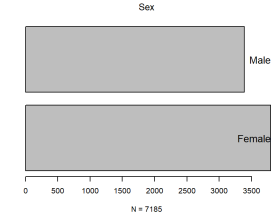
Note: with an approximately normal distribution we expect mean -/+ std. dev. to be at approx the 32nd and 68th percentiles.

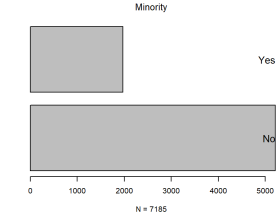
```
xqplot( hsfull , ptype='normal')
```

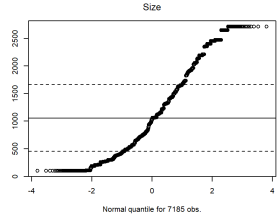

school
Normal quantile for 7185 obs.

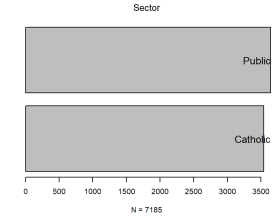

mathach
Normal quantile for 7185 obs.

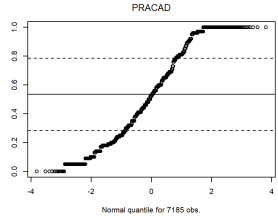

ses
Normal quantile for 7185 obs.


Sex
Male
Female
N = 7185


Minority
Yes
No
N = 7185


Size
Normal quantile for 7185 obs.


Sector
Public
Catholic
N = 7185


PRACAD
Normal quantile for 7185 obs.

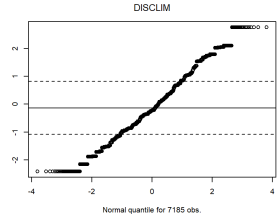

DISCLIM
Normal quantile for 7185 obs.

Figure: A **normal quantile** plot. Data are lined up from shortest to tallest and x-axis is stretched at the ends to have shape of ideal normal. If a variable is close to normal, its quantile plot will be close to a diagonal

Data: Levels and structure —-

Data structures, part 1: moving up one level

'hsfull' has one student per row.

It is a **long** file combining variables of all levels. If we want information on schools, we can create a data set that has one row per school, keeping only those variable that are invariant within schools, i.e. Level 2 variables

```
hsup <- up( hsfull, ~ school )  # one row per school with Level 2 variables
some( hsup )
```

```
      school Size   Sector PRACAD DISCLIM
3013   3013  760   Public   0.56  -0.213
3351   3351 1125   Public   0.35   0.383
3498   3498  388 Catholic   0.92  -1.556
4350   4350 1661   Public   0.62   0.512
4931   4931  603 Catholic   0.79  -1.172
```

```
   5762   5762 1826   Public   0.24   0.364
   7232   7232 1154   Public   0.20   0.975
   7919   7919 1451   Public   0.50  -0.402
   8775   8775  667   Public   0.28   1.783
   9021   9021  472 Catholic   0.82  -0.817
```

```
dim( hsup )
```

```
   [1] 160   5
```

```
tab(hsup, ~ Sector)
```

```
   Sector
   Catholic   Public    Total
         70       90      160
```

gicc: levels of variables in hsfull

Generalized intraclass correlation:

- intraclass correlation (variance between / total variance) of ranks for numerical variables
- Goodman-Kruskal's tau for factor and character variables

Interpretation:

- 1 means it's a level-2 or school-level variable,
- 0 means that it's mean is the same for each school, e.g. if it's balanced

```
gicc( hsfull, ~ school, type = 'raw' )  # uses variance between / total variance
```

```
      school   mathach       ses       Sex  Minority      Size    Sector
   1.0000000 0.1955041 0.2749462 0.2869238 0.4644279 1.0000000 1.0000000
      PRACAD   DISCLIM
   1.0000000 1.0000000
```

```
gicc( hsfull, ~ school )  # uses ranks by default
```

```
      school   mathach       ses       Sex  Minority      Size    Sector
   1.0000000 0.1955041 0.2749462 0.2869238 0.4644279 1.0000000 1.0000000
      PRACAD   DISCLIM
   1.0000000 1.0000000
```

```
tab_( hsfull, ~ school )          # school size: 'tab_' is a variant of 'tab' wit
```

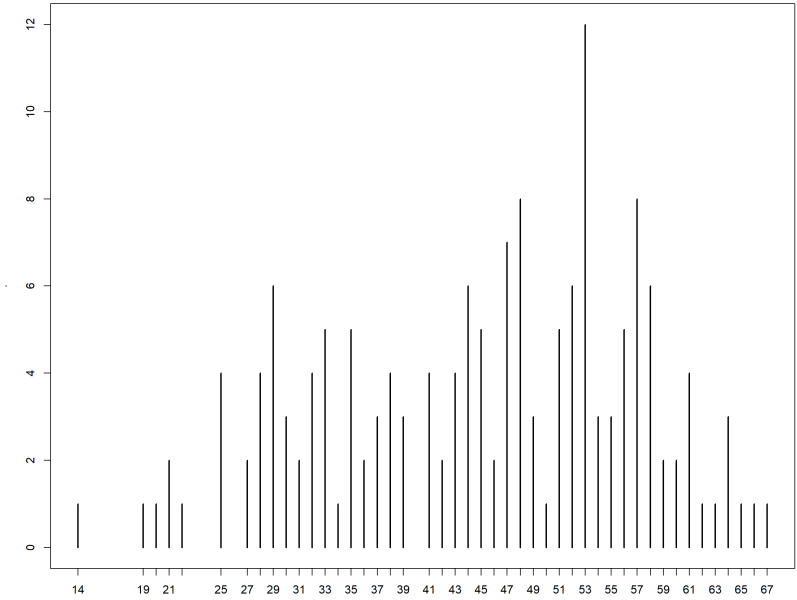
```
   school
   1224 1288 1296 1308 1317 1358 1374 1433 1436 1461 1462 1477 1499 1637 1906
     47   25   48   20   48   30   28   35   44   33   57   62   53   27   53
   1909 1942 1946 2030 2208 2277 2305 2336 2458 2467 2526 2626 2629 2639 2651
     28   29   39   47   60   61   67   47   57   52   57   38   57   42   38
```

```
2655 2658 2755 2768 2771 2818 2917 2990 2995 3013 3020 3039 3088 3152 3332
  52   45   47   25   55   42   43   48   46   53   59   21   39   52   38
3351 3377 3427 3498 3499 3533 3610 3657 3688 3705 3716 3838 3881 3967 3992
  39   45   49   53   38   48   64   51   43   45   41   54   41   52   53
3999 4042 4173 4223 4253 4292 4325 4350 4383 4410 4420 4458 4511 4523 4530
  46   64   44   45   58   65   53   33   25   41   32   48   58   47   63
4642 4868 4931 5192 5404 5619 5640 5650 5667 5720 5761 5762 5783 5815 5819
  61   34   58   28   57   66   57   45   61   53   52   37   29   25   50
5838 5937 6074 6089 6144 6170 6291 6366 6397 6415 6443 6464 6469 6484 6578
  31   29   56   33   43   21   35   58   60   54   30   29   57   35   56
6600 6808 6816 6897 6990 7011 7101 7172 7232 7276 7332 7341 7342 7345 7364
  56   44   55   49   53   33   28   44   52   53   48   51   58   56   44
7635 7688 7697 7734 7890 7919 8009 8150 8165 8175 8188 8193 8202 8357 8367
  51   54   32   22   51   37   47   44   49   33   30   43   35   27   14
8477 8531 8627 8628 8707 8775 8800 8854 8857 8874 8946 8983 9021 9104 9158
  37   41   53   61   48   48   32   32   64   36   58   51   56   55   53
9198 9225 9292 9340 9347 9359 9397 9508 9550 9586
  31   36   19   29   57   53   47   35   29   59
```

```
tab_( hsfull, ~ school ) %>% c %>% tab # distribution of school sizes
```

14	19	20	21	22	25	27	28	29	30	31	32
1	1	1	2	1	4	2	4	6	3	2	4
33	34	35	36	37	38	39	41	42	43	44	45
5	1	5	2	3	4	3	4	2	4	6	5
46	47	48	49	50	51	52	53	54	55	56	57
2	7	8	3	1	5	6	12	3	3	5	8
58	59	60	61	62	63	64	65	66	67	Total	
6	2	2	4	1	1	3	1	1	1	160	

```
tab_( hsfull, ~ school ) %>% c %>% tab_ %>% plot
tab_( hsfull, ~ school ) %>% as.vector %>% tab_ %>% plot
```


0
2
4
6
8
10
12
14
19
21
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
59
61
63
65
67

Level-2 variables

```
hsup <- up( hsfull, ~ school ) # 'up' one level in the multilevel hierarchy
```

To include summaries of selected Level-1 variables.

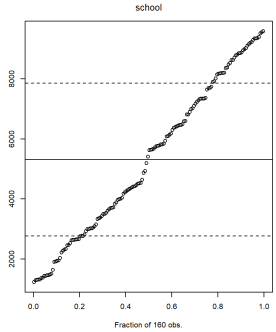
For factors, this produces the mean of the incidence matrix within each school, i.e. the proportion of students in each category of the factor.

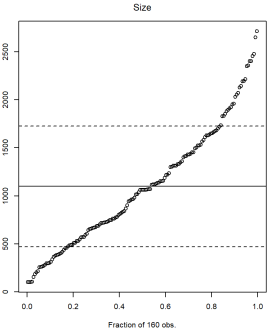
For each factor, these proportions must sum to 1.

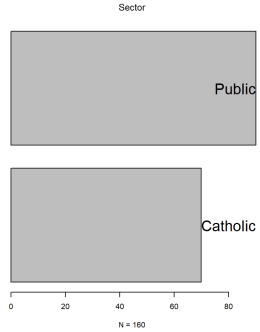
```
hsfull %>%
  up( ~ school, agg = ~ Sex + ses ) %>%
  head
```

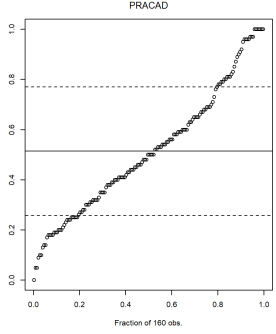
	school	Size	Sector	PRACAD	DISCLIM	Sex_Female	Sex_Male	ses
1224	1224	842	Public	0.35	1.597	0.5957447	0.4042553	-0.43438298
1288	1288	1855	Public	0.27	0.174	0.4400000	0.5600000	0.12160000
1296	1296	1719	Public	0.32	-0.137	0.6458333	0.3541667	-0.42550000
1308	1308	716	Catholic	0.96	-0.622	0.0000000	1.0000000	0.52800000
1317	1317	455	Catholic	0.95	-1.694	1.0000000	0.0000000	0.34533333
1358	1358	1430	Public	0.25	1.535	0.3666667	0.6333333	-0.01966667

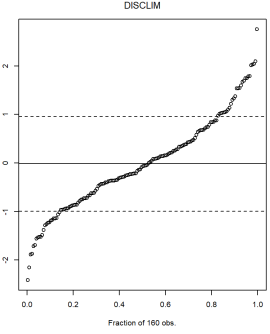
```
xqplot( hsup )
```


school
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 160 obs.


Size
0
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 160 obs.


Sector
Public
Catholic
0
20
40
60
80
N = 160


PRACAD
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 160 obs.


DISCLIM
-2
-1
0
1
2
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 160 obs.

To speed up calculations, we will use a (pre-)randomly selected quarter of the schools.

However, since it's a good idea to know that it's easy to select a random subset of schools, **not students**, here's how it's done.

Selecting a random subset of clusters —-

We can't randomly sample from the long file if we want each cluster (school) to be either all in or all out.

There are a few ways of selecting all the observations from a sample of clusters. The following seems fairly intuitive.

We first create the school level file and take a sample of school from that file. We then merge the sample file with the longfile. Merge will match with variables that have the same name. By default, it only uses records that match in both files, so it produces the result we want.

```
hsup <- up ( hsfull, ~ school)  # as above
dim(hsup)                       # one row per school
```

```
[1] 160   5
```

We'll use a randomly selected subset of half the schools

This has another interesting consequence: we can do a **split-half analysis** in which the model developed on one half of the clusters is validated with the other half

The following code shows how to split the clusters into two halves although we'll be using two 'preselected' halves

How to select a random sample of 5 numbers from the numbers 1 to 10:

```
sample(1:10, 5)
```

```
[1] 5 9 7 3 6
```

Select a sample of half the row indices:

```
selected_rows <- sample( 1:nrow(hsup), round( nrow(hsup)/2 ))
selected_rows
```

```
 [1] 113  47  36  59 126 138 145  28  57  46 129  62  60  19  90  38 150
[18]  31 149 106  30  61  54 102  91   6  32 157 139  67  68   9 140  11
[35]  82  72  27 128  78  87  56  13  85  58 124 101 121 155  97 119 100
[52]  49  79 103  39  99  84  76  40 156  75  88 123  69  33   4  80 125
[69] 111   8  66 152  77 110  55  18  53   3  71 122
```

```
hsu1 <- hsup[selected_rows, ]      # first half of schools
hsu2 <- hsup[ - selected_rows, ]    # other half
```

```
hs1 <- merge( hsu1, hsfull )    # long data for 1st half of schools
hs2 <- merge( hsu2, hsfull )    # long data for other half

dim(hs1)

   [1] 3687    9

dim(hs2)

   [1] 3498    9

length(unique(hs1$school))

   [1] 80

length(unique(hs2$school))

   [1] 80
```

But we will use an even smaller preselected subset of 40 schools

```
data(hs)
dd <- hs              # easier to type repeatedly

dim( dd )
```

```
[1] 1977    9
```

```
some(dd)
```

```
      school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM
58      1906  14.171  0.602   Male       No  400 Catholic   0.87  -0.939
268     2629  14.691 -0.768   Male       No 1314 Catholic   0.81  -0.613
341     2639   6.166 -0.888 Female      Yes 2713   Public   0.14  -0.282
557     3610  13.942  0.602   Male       No 1431 Catholic   0.80  -0.621
915     5640  23.048  0.632   Male       No 1152   Public   0.41   0.256
916     5640  18.066 -0.038 Female       No 1152   Public   0.41   0.256
926     5640  18.225  0.332   Male       No 1152   Public   0.41   0.256
987     5650  20.907 -0.768 Female      Yes  720 Catholic   0.60  -0.070
1734    8627  13.946 -0.018   Male       No 2452   Public   0.25   0.742
1811    8707   8.432 -0.238   Male      Yes 1133   Public   0.48   1.542
```

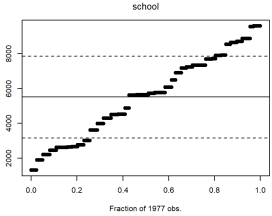
First look at variables —-

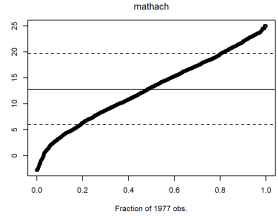
```
summary( dd )
```

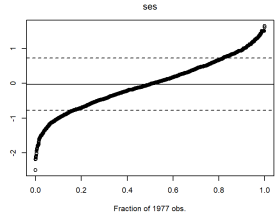
```
       school        mathach            ses               Sex
```

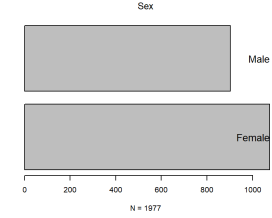
```
    Min.   :1317   Min.   :-2.832   Min.   :-2.49800   Female:1074
    1st Qu.:3013   1st Qu.: 7.529   1st Qu.:-0.55800   Male  : 903
    Median :5650   Median :13.095   Median :-0.02800
    Mean   :5507   Mean   :12.783   Mean   :-0.02684
    3rd Qu.:7345   3rd Qu.:18.336   3rd Qu.: 0.52200
    Max.   :9586   Max.   :24.993   Max.   : 1.65200
    Minority        Size           Sector         PRACAD
    No :1433   Min.   : 215   Catholic:1144   Min.   :0.0500
    Yes: 544   1st Qu.: 545   Public  : 833   1st Qu.:0.3200
               Median :1114                   Median :0.5800
               Mean   :1106                   Mean   :0.5513
               3rd Qu.:1415                   3rd Qu.:0.7600
               Max.   :2713                   Max.   :1.0000
       DISCLIM
    Min.   :-2.4160
    1st Qu.:-0.9390
    Median :-0.2820
    Mean   :-0.3022
    3rd Qu.: 0.3360
    Max.   : 1.7420
```

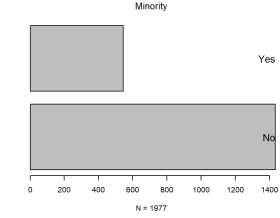
```
xqplot( dd )
```

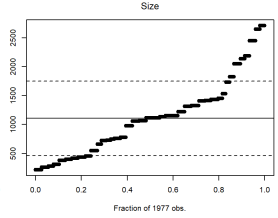

school
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 1977 obs.

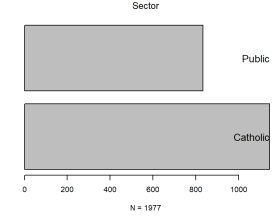

mathach
0
5
10
15
20
25
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 1977 obs.

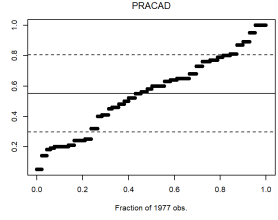

ses
-2
-1
0
1
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 1977 obs.

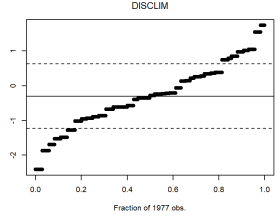

Sex
Male
Female
0
200
400
600
800
1000
N = 1977


Minority
Yes
No
0
200
400
600
800
1000
1200
1400
N = 1977


Size
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 1977 obs.


Sector
Public
Catholic
0
200
400
600
800
1000
N = 1977


PRACAD
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 1977 obs.


DISCLIM
-2
-1
0
1
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 1977 obs.

```
library(Hmisc, pos = 1000) # to avoid masking functions in other packages
   Loading required package: survival
   Loading required package: Formula
   Loading required package: ggplot2

   Attaching package: 'ggplot2'
   The following object is masked from 'package:latticeExtra':

       layer
   The following object is masked from 'package:spida2':

       labs

   Attaching package: 'Hmisc'
   The following object is masked _by_ 'package:p3d':

       na.include
```

```
The following objects are masked _by_ 'package:spida2':

    fillin, Lag, na.include

The following objects are masked from 'package:base':

    format.pval, units
```

```
describe(dd)
```

```
dd

 9  Variables      1977  Observations
--------------------------------------------------------------------------------
school
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1977        0       40    0.999     5507     2695     1906     2458
     .25      .50      .75      .90      .95
    3013     5650     7345     8707     8874

lowest : 1317 1906 2208 2458 2626, highest: 8707 8854 8874 9550 9586
--------------------------------------------------------------------------------
mathach
```

```
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1977        0     1860        1    12.78    7.863    1.363    3.369
     .25      .50      .75      .90      .95
   7.529   13.095   18.336   21.892   23.220

lowest : -2.832 -2.721 -2.658 -2.557 -2.550, highest: 24.488 24.545 24.707 24
---------------------------------------------------------------------------
ses
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1977        0      323        1 -0.02684   0.8577   -1.270   -0.998
     .25      .50      .75      .90      .95
  -0.558   -0.028    0.522    0.952    1.202

lowest : -2.498 -2.188 -2.178 -2.128 -2.108, highest:  1.452  1.462  1.512  1
---------------------------------------------------------------------------
Sex
       n  missing distinct
    1977        0        2

Value      Female   Male
Frequency    1074    903
```

```
Proportion  0.543  0.457
-------------------------------------------------------------------------------
Minority
       n  missing distinct
    1977        0        2

Value          No   Yes
Frequency    1433   544
Proportion 0.725 0.275
-------------------------------------------------------------------------------
Size
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1977        0       40    0.999     1106    714.7      262      311
     .25      .50      .75      .90      .95
     545     1114     1415     2142     2452

lowest :  215  262  280  311  381, highest: 2142 2190 2452 2650 2713
-------------------------------------------------------------------------------
Sector
       n  missing distinct
    1977        0        2
```

Value	Catholic	Public
Frequency	1144	833
Proportion	0.579	0.421

PRACAD

n	missing	distinct	Info	Mean	Gmd	.05	.10
1977	0	34	0.999	0.5513	0.2903	0.180	0.200

.25	.50	.75	.90	.95
0.320	0.580	0.760	0.878	0.950

lowest : 0.05 0.14 0.18 0.19 0.20, highest: 0.81 0.87 0.89 0.95 1.00

DISCLIM

n	missing	distinct	Info	Mean	Gmd	.05	.10
1977	0	40	0.999	-0.3022	1.054	-1.872	-1.534

.25	.50	.75	.90	.95
-0.939	-0.282	0.336	0.975	1.048

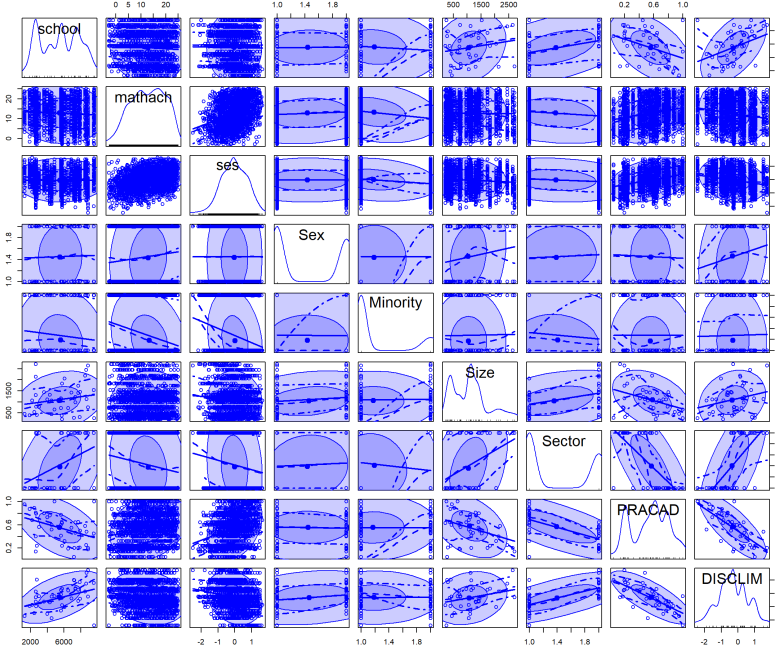
lowest : -2.416 -1.872 -1.694 -1.534 -1.484, highest: 0.975 1.013 1.048 1

QUESTIONS: looking at univariate distributions for this data:

Any problems with: NAs? Highly skewed distributions? Possible univariate outliers? Skewed factors: i.e. very rare category

Any actions to take?

```
library(car)
scatterplotMatrix( dd , ellipse = TRUE)    # 2-dim summary
```


school
mathach
ses
Sex
Minority
Size
Sector
PRACAD
DISCLIM

QUESTIONS: looking at bivariate relationships

```
Hint: focus on one column or one row at a time

Bivariate relationships that stand out?
Bivariate outliers?
Curvilinear bivariate relationships?
Which pair has the 'strongest' bivariate relationship?
```

Looking at Level 2 variables (invariant within schools) —-

```
ddu <- up( dd, ~ school)  # only variables invariant within schools
dim(ddu)
```

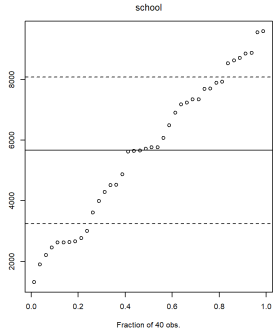
```
[1] 40  5
```

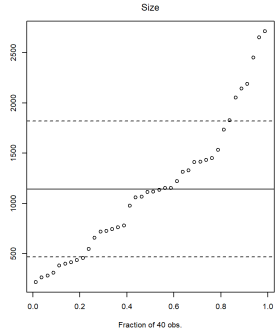
```
some(ddu)
```

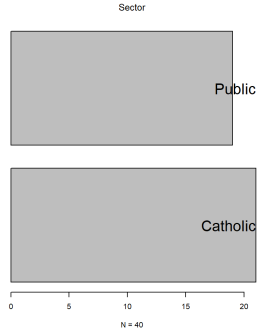
```
      school Size   Sector PRACAD DISCLIM
2658   2658  780 Catholic   0.79  -0.961
2771   2771  415   Public   0.24   1.048
5650   5650  720 Catholic   0.60  -0.070
5761   5761  215 Catholic   0.63  -0.892
```

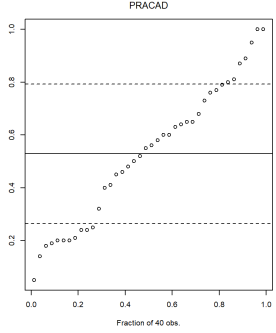
```
   5762   5762 1826   Public   0.24   0.364
   6074   6074 2051 Catholic   0.32  -1.018
   7232   7232 1154   Public   0.20   0.975
   7342   7342 1220 Catholic   0.46   0.380
   8854   8854  745   Public   0.18  -0.228
   9586   9586  262 Catholic   1.00  -2.416
```

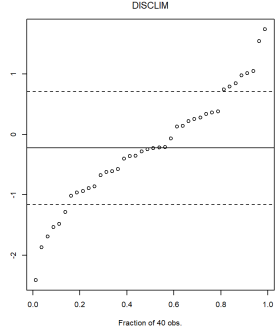
```
xqplot( ddu )    # shows just variables invariant within schools
```


school
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.

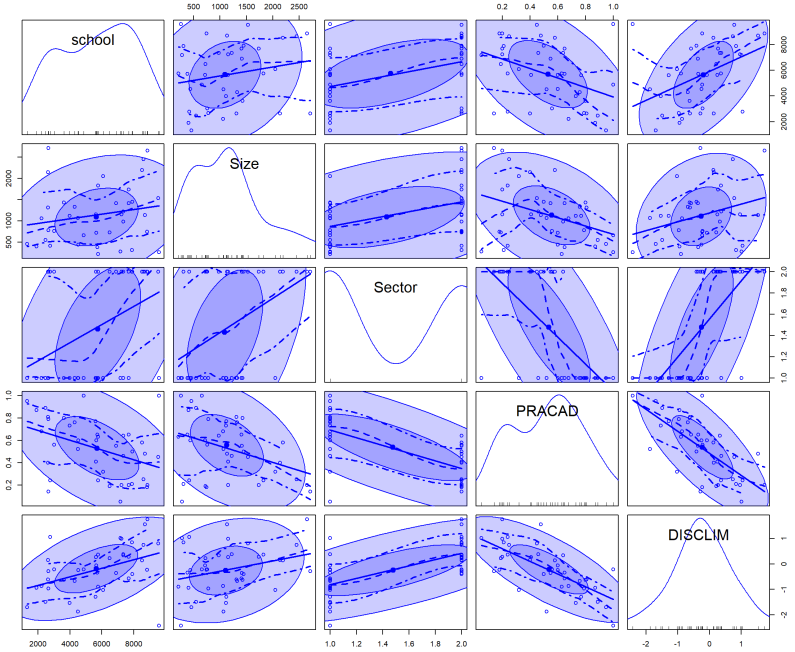

Size
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.


Sector
Public
Catholic
0
5
10
15
20
N = 40


PRACAD
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.


DISCLIM
-2
-1
0
1
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.

```r
scatterplotMatrix( ddu , ell = T)    # between school variables
```


school
Size
Sector
PRACAD
DISCLIM

Include derived Level 2 summaries of Level 1 variables

- Level 2 means of Level 1 numerical variables
- Level 2 **modes** of Level 1 categorical variables (not very useful for analysis)
- Recall that **agg** returns the mean of incidence matrix

```
up(dd, ~school) %>% head
```

```
      school Size   Sector PRACAD DISCLIM
1317   1317  455 Catholic   0.95  -1.694
1906   1906  400 Catholic   0.87  -0.939
2208   2208 1061 Catholic   0.68  -0.864
2458   2458  545 Catholic   0.89  -1.484
2626   2626 2142   Public   0.40   0.142
2629   2629 1314 Catholic   0.81  -0.613
```

```
ddall <- up (dd, ~school, all = T)
head(ddall)
```

```
      school  mathach         ses    Sex Minority Size   Sector PRACAD
1317   1317 13.17769  0.34533333 Female      Yes  455 Catholic   0.95
1906   1906 15.98317  0.51162264 Female       No  400 Catholic   0.87
2208   2208 15.40467  0.42316667 Female       No 1061 Catholic   0.68
2458   2458 13.98568  0.22778947 Female      Yes  545 Catholic   0.89
```

```
2626   2626 13.39661 -0.06484211   Male       No 2142   Public   0.40
2629   2629 14.90777 -0.13764912   Male       No 1314 Catholic   0.81
     DISCLIM
1317  -1.694
1906  -0.939
2208  -0.864
2458  -1.484
2626   0.142
2629  -0.613
```

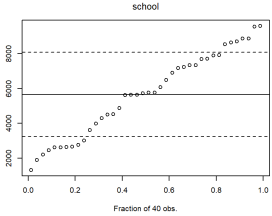
To get distribution of proportions for factors – not just modes

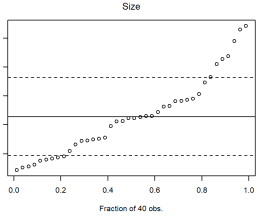
```
ddagg <- up(dd, ~school, agg = ~ Minority + Sex)
head(ddagg)
```

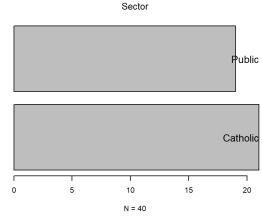
```
     school Size   Sector PRACAD DISCLIM Minority_No Minority_Yes
1317   1317  455 Catholic   0.95  -1.694   0.2708333   0.72916667
1906   1906  400 Catholic   0.87  -0.939   0.9056604   0.09433962
2208   2208 1061 Catholic   0.68  -0.864   1.0000000   0.00000000
2458   2458  545 Catholic   0.89  -1.484   0.3859649   0.61403509
2626   2626 2142   Public   0.40   0.142   1.0000000   0.00000000
2629   2629 1314 Catholic   0.81  -0.613   0.7894737   0.21052632
     Sex_Female  Sex_Male
```

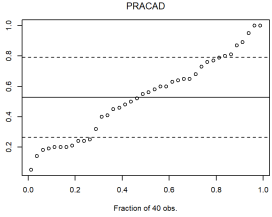
```
   1317  1.0000000 0.0000000
   1906  0.5094340 0.4905660
   2208  0.5833333 0.4166667
   2458  1.0000000 0.0000000
   2626  0.4736842 0.5263158
   2629  0.0000000 1.0000000
```

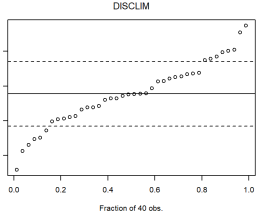
```
xqplot( ddagg )
```

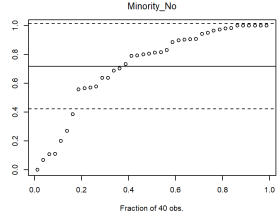

school
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.

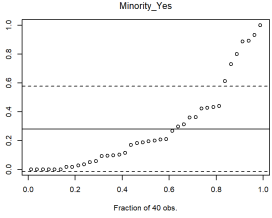

Size
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.

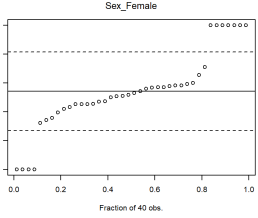

Sector
Public
Catholic
0
5
10
15
20
N = 40

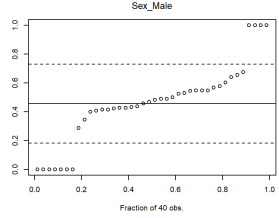

PRACAD
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.


DISCLIM
-2
-1
0
1
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.


Minority_No
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.


Minority_Yes
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.

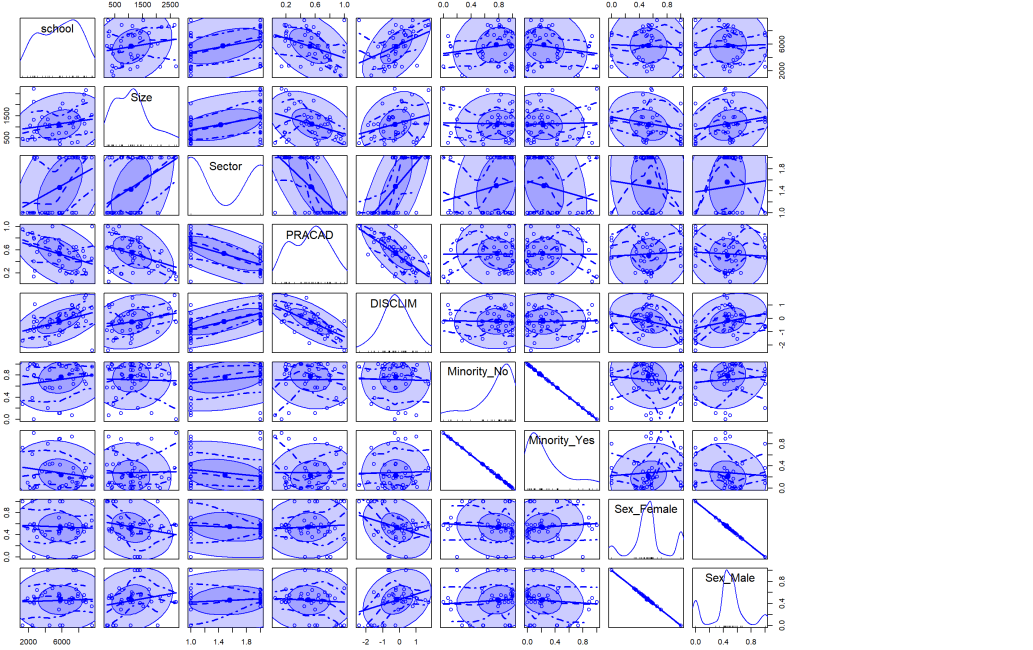

Sex_Female
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.


Sex_Male
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 40 obs.

```
spm( ddagg, ell = T  )  # spm = scatterplotMatrix for slow typists
```

```
Warning in MASS::cov.trob(cbind(x[use], y[use]), wt = weights[use]):
Probable convergence failure

Warning in chol.default(shape, pivot = TRUE): the matrix is either rank-
deficient or indefinite

Warning in chol.default(shape, pivot = TRUE): the matrix is either rank-
deficient or indefinite

Warning in MASS::cov.trob(cbind(x[use], y[use]), wt = weights[use]):
Probable convergence failure

Warning in MASS::cov.trob(cbind(x[use], y[use]), wt = weights[use]):
Probable convergence failure

Warning in chol.default(shape, pivot = TRUE): the matrix is either rank-
deficient or indefinite

Warning in chol.default(shape, pivot = TRUE): the matrix is either rank-
deficient or indefinite

Warning in MASS::cov.trob(cbind(x[use], y[use]), wt = weights[use]):
```

```
Probable convergence failure

Warning in chol.default(shape, pivot = TRUE): the matrix is either rank-
deficient or indefinite

Warning in chol.default(shape, pivot = TRUE): the matrix is either rank-
deficient or indefinite
```


school
Size
Sector
PRACAD
DISCLIM
Minority_No
Minority_Yes
Sex_Female
Sex_Male

QUESTIONS:

- Note that there is a relationship between minority and ses as well as between minority and mathach
- Does anything stand out in 'between-school' relationships?
- Classify variables:
  - Level 1 variables
  - Level 2 variables:
    - contextual variables: natural (properties of cluster) and
    - compositional: derived cluster values from individual values
- Does this suggest any multilevel questions?
- Are there derived level 2 (compositional) variables that could be relevant?

Creating additional Level 2 (and Level 1) variables with 'capply' —-

An important part of modeling is being able to easily create the variables that capture the phenomena you want to study. Often, these variables are not in the raw data set.

Examples:

Sample size:

capply(y, cluster, fun)

1. divides 'y' into chunks for each value of 'cluster'
2. applies the function 'fun' to each chunk
3. a. if 'fun' returns a vector of the same length as the chunk of 'y' then 'capply' returns that vector in the positions corresponding to the values of 'y'

   b. if 'fun' returns a single value, that value gets returned in every position of 'y'

How many observations within each school?

```
dd$n <- capply( dd$school, dd$school, length)
head( dd )  # to see that new variable is constant in first school
```

```
     school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
  1    1317  12.862  0.882 Female       No  455 Catholic   0.95  -1.694 48
  2    1317   8.961  0.932 Female      Yes  455 Catholic   0.95  -1.694 48
  3    1317   4.756 -0.158 Female      Yes  455 Catholic   0.95  -1.694 48
  4    1317  21.405  0.362 Female      Yes  455 Catholic   0.95  -1.694 48
  5    1317  20.748  1.372 Female       No  455 Catholic   0.95  -1.694 48
  6    1317  18.362  0.132 Female      Yes  455 Catholic   0.95  -1.694 48
```

```
some( dd )
```

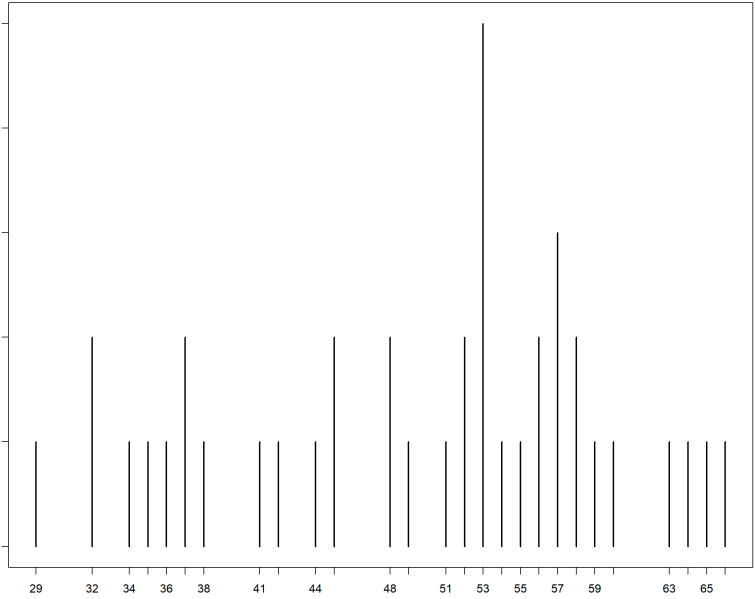
```
       school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
```

449	2771	9.589	0.302	Male	No	415	Public	0.24	1.048	55
486	3013	22.377	-0.768	Male	No	760	Public	0.56	-0.213	53
572	3610	19.128	-0.808	Female	Yes	1431	Catholic	0.80	-0.621	64
905	5619	23.062	0.932	Male	No	1118	Catholic	0.77	-1.286	66
1370	7232	6.258	-1.418	Female	No	1154	Public	0.20	0.975	52
1409	7342	13.848	-0.718	Male	No	1220	Catholic	0.46	0.380	58
1529	7688	19.060	0.792	Male	No	1410	Catholic	0.65	-0.575	54
1812	8707	12.167	-0.858	Female	Yes	1133	Public	0.48	1.542	48
1863	8874	15.577	0.652	Female	Yes	2650	Public	0.20	1.742	36
1919	9586	14.076	0.852	Female	No	262	Catholic	1.00	-2.416	59

```
dd %>%
  {up(., ~school)$n} %>%    # piping to an expression and using '.' for piped ar
  tab
```

29	32	34	35	36	37	38	41	42	44	45	48
1	2	1	1	1	2	1	1	1	1	2	2
49	51	52	53	54	55	56	57	58	59	60	63
1	1	2	5	1	1	2	3	2	1	1	1
64	65	66	Total								
1	1	1	40								

```
dd %>%
  {up(., ~school)$n} %>%
  tab_ %>%
  plot
```


0
1
2
3
4
5
29
32
34
36
38
41
44
48
51
53
55
57
59
63
65

mean ses by school (sample compositional ses)

```
dd$ses.m <- with( dd, capply(ses, school, mean, na.rm = TRUE ))
head( dd )
```

```
     school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
  1   1317  12.862  0.882 Female       No  455 Catholic   0.95  -1.694 48
  2   1317   8.961  0.932 Female      Yes  455 Catholic   0.95  -1.694 48
  3   1317   4.756 -0.158 Female      Yes  455 Catholic   0.95  -1.694 48
  4   1317  21.405  0.362 Female      Yes  455 Catholic   0.95  -1.694 48
  5   1317  20.748  1.372 Female       No  455 Catholic   0.95  -1.694 48
  6   1317  18.362  0.132 Female      Yes  455 Catholic   0.95  -1.694 48
        ses.m
  1 0.3453333
  2 0.3453333
  3 0.3453333
  4 0.3453333
  5 0.3453333
  6 0.3453333
```

ses centered within school (CWG = centered with groups) This is the deviation of a student from the mean of the sample in her school

```
dd$ses.cwg <- dd$ses - dd$ses.m
head( dd )
```

```
     school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
   1   1317  12.862  0.882 Female       No  455 Catholic   0.95  -1.694 48
   2   1317   8.961  0.932 Female      Yes  455 Catholic   0.95  -1.694 48
   3   1317   4.756 -0.158 Female      Yes  455 Catholic   0.95  -1.694 48
   4   1317  21.405  0.362 Female      Yes  455 Catholic   0.95  -1.694 48
   5   1317  20.748  1.372 Female       No  455 Catholic   0.95  -1.694 48
   6   1317  18.362  0.132 Female      Yes  455 Catholic   0.95  -1.694 48
         ses.m     ses.cwg
   1 0.3453333  0.53666667
   2 0.3453333  0.58666667
   3 0.3453333 -0.50333333
   4 0.3453333  0.01666667
   5 0.3453333  1.02666667
   6 0.3453333 -0.21333333
```

```
layer <- latticeExtra::layer
xyplot(ses.cwg ~ ses.m, dd) + layer(panel.dell(...,col='red',lwd=3))
```

ses.cwg
ses.m
2
1
0
-1
-2
-1.0
-0.5
0.0
0.5

Note that ses.cwg and ses.m are uncorrelated

ses heterogeneity as measured by the IQR within schools

```
dd$ses.iqr <- capply( dd$ses, dd$school,
                      function ( x ) {
                        qs <- quantile(x, c(25,75)/100)
                        qs[2] - qs[1]
                      }
)
```

Note: There is an "IQR" function to compute the inter-quartile range but this does it from scratch to illustrate the use of an 'anonymous' function, i.e. a function defined on the fly.

We could have done:

```
dd$ses.iqr <- capply(dd$ses, dd$school, IQR, na.rm = FALSE)
```

but we could modify the anomymous function it to

```
dd$ses.trange <- capply( dd$ses, dd$school,   # where 'trange' stands for
                                              # 'truncated range'
                         function ( x ) {
                           qs <- quantile(x, c(10,90)/100, na.rm = FALSE)
                           qs[2] - qs[1]
```

```
                        }
)
```

consider variations like

```
dd$ses.sd <- capply( dd$ses, dd$school, sd, na.rm = FALSE)

some( dd )
```

```
      school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
663     4292  12.956  0.852   Male      Yes 1328 Catholic   0.76  -0.674 65
677     4292   7.128 -1.148   Male      Yes 1328 Catholic   0.76  -0.674 65
763     4530   6.295 -0.738 Female      Yes  435 Catholic   0.60  -0.245 63
943     5640   4.037 -0.818 Female       No 1152   Public   0.41   0.256 57
1001    5650   8.507  1.262 Female       No  720 Catholic   0.60  -0.070 45
1107    5761  14.872 -0.278 Female      Yes  215 Catholic   0.63  -0.892 52
1121    5762  10.810 -1.498   Male      Yes 1826   Public   0.24   0.364 37
1163    6074  15.573 -0.438 Female       No 2051 Catholic   0.32  -1.018 56
1820    8707  18.470  1.512 Female      Yes 1133   Public   0.48   1.542 48
1905    9550  22.802  0.302   Male       No 1532   Public   0.45   0.791 29
           ses.m    ses.cwg ses.iqr ses.trange    ses.sd
663  -0.48615385  1.3381538  0.8600      1.708 0.6511382
677  -0.48615385 -0.6618462  0.8600      1.708 0.6511382
```

```
763  -0.59688889 -0.1411111  0.8850      1.544 0.6210062
943  -0.17659649 -0.6414035  0.8500      1.492 0.5830261
1001  0.02244444  1.2395556  1.0600      2.114 0.7777414
1107 -0.32300000  0.0450000  0.8000      1.917 0.7122389
1121 -1.19394595 -0.3040541  0.6500      1.272 0.5154149
1163 -0.27675000 -0.1612500  0.8175      1.440 0.6271235
1820  0.15512500  1.3568750  1.1350      2.169 0.8042577
1905  0.05303448  0.2489655  1.0200      1.824 0.7847035
```

```
head( dd )
```

```
  school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
1   1317  12.862  0.882 Female       No  455 Catholic   0.95  -1.694 48
2   1317   8.961  0.932 Female      Yes  455 Catholic   0.95  -1.694 48
3   1317   4.756 -0.158 Female      Yes  455 Catholic   0.95  -1.694 48
4   1317  21.405  0.362 Female      Yes  455 Catholic   0.95  -1.694 48
5   1317  20.748  1.372 Female       No  455 Catholic   0.95  -1.694 48
6   1317  18.362  0.132 Female      Yes  455 Catholic   0.95  -1.694 48
      ses.m     ses.cwg ses.iqr ses.trange    ses.sd
1 0.3453333  0.53666667  0.7825      1.166 0.5561583
2 0.3453333  0.58666667  0.7825      1.166 0.5561583
3 0.3453333 -0.50333333  0.7825      1.166 0.5561583
```

```
4 0.3453333  0.01666667  0.7825      1.166 0.5561583
5 0.3453333  1.02666667  0.7825      1.166 0.5561583
6 0.3453333 -0.21333333  0.7825      1.166 0.5561583
```

Tranformations of Level 1 variables within groups —-

capply will also create a Level 1 variable if the FUN returns a a vector of length greater than 1. The vector gets recycled to match the length of the argument, i.e. the number of the rows corresponding to a particular id.

The following example shows the calculation of ses rank within schools:

```
dd$ses.rank <- capply( dd$ses, dd$school, rank , na.last = 'keep' )
some( dd )
```

```
      school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
12      1317  23.355  0.502 Female       No  455 Catholic   0.95  -1.694 48
62      1906  21.122  0.382 Female       No  400 Catholic   0.87  -0.939 53
722     4511  20.566 -0.258 Female       No 1068 Catholic   0.52  -1.872 58
946     5640  17.085  0.012 Female       No 1152   Public   0.41   0.256 57
1027    5720  15.896 -0.078   Male       No  381 Catholic   0.65  -0.352 53
1150    5762   2.617 -1.368   Male       No 1826   Public   0.24   0.364 37
```

```
   1216   6484  15.296  1.122   Male       No  726   Public   0.19   0.218 35
   1394   7342  23.271 -0.748   Male       No 1220 Catholic   0.46   0.380 58
   1479   7345   5.926 -1.048 Female       No  978   Public   0.64   0.336 56
   1827   8854  -2.832 -0.408   Male       No  745   Public   0.18  -0.228 32
             ses.m    ses.cwg ses.iqr ses.trange    ses.sd ses.rank
   12    0.34533333  0.1566667  0.7825      1.166 0.5561583     30.0
   62    0.51162264 -0.1296226  1.0100      1.586 0.6135833     24.5
   722  -0.10713793 -0.1508621  0.8875      1.476 0.5813363     26.0
   946  -0.17659649  0.1885965  0.8500      1.492 0.5830261     37.5
   1027  0.03256604 -0.1105660  1.0100      1.730 0.6641693     23.0
   1150 -1.19394595 -0.1740541  0.6500      1.272 0.5154149     15.0
   1216 -0.18514286  1.3071429  0.7400      1.932 0.6958345     33.0
   1394 -0.44782759 -0.3001724  0.7125      1.323 0.5648459     19.0
   1479  0.03325000 -1.0812500  1.3550      1.930 0.8257296      6.0
   1827 -0.75675000  0.3487500  0.8125      1.732 0.8036439     26.0
```

```
head( dd )
```

```
     school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
   1   1317  12.862  0.882 Female       No  455 Catholic   0.95  -1.694 48
   2   1317   8.961  0.932 Female      Yes  455 Catholic   0.95  -1.694 48
   3   1317   4.756 -0.158 Female      Yes  455 Catholic   0.95  -1.694 48
```

```
4   1317  21.405  0.362 Female      Yes  455 Catholic   0.95  -1.694 48
5   1317  20.748  1.372 Female       No  455 Catholic   0.95  -1.694 48
6   1317  18.362  0.132 Female      Yes  455 Catholic   0.95  -1.694 48
      ses.m     ses.cwg ses.iqr ses.trange    ses.sd ses.rank
1 0.3453333  0.53666667  0.7825      1.166 0.5561583       39
2 0.3453333  0.58666667  0.7825      1.166 0.5561583       40
3 0.3453333 -0.50333333  0.7825      1.166 0.5561583        6
4 0.3453333  0.01666667  0.7825      1.166 0.5561583       25
5 0.3453333  1.02666667  0.7825      1.166 0.5561583       47
6 0.3453333 -0.21333333  0.7825      1.166 0.5561583       20
```

Using the rank instead of the raw value could be useful for highly skewed variables where you don't want extreme values to have too much influence

The following example shows the use of 'capply' to compute a value that depends on more than one variable in the data frame. If the first argument is a data frame, then 'capply' splits the data frame into data frames with just the rows belonging to each id. Using the 'with' function on these data frames allows you to write an expression as the fourth argument which becomes the second argument of with.

ses discrepancy of Minority in school (requires more than one variable)

```
dd$minority.diff <- capply( dd, ~ school, with,
```

```
                                    mean( ses[ Minority == "Yes" ] , na.rm = T) -
                                      mean( ses[ Minority == "No" ] , na.rm = T) )
```

Beware: this way of using 'capply' can be slow with very large files such as the NPHS long file.

```
some( dd )
```

```
      school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
95      1906  -1.603  0.282   Male      Yes  400 Catholic   0.87  -0.939 53
141     2208   6.078  0.772 Female       No 1061 Catholic   0.68  -0.864 60
200     2458  20.689  0.502 Female       No  545 Catholic   0.89  -1.484 57
278     2629  13.913  0.352   Male       No 1314 Catholic   0.81  -0.613 57
281     2629  19.111 -0.768   Male       No 1314 Catholic   0.81  -0.613 57
1358    7232  17.001  0.332 Female       No 1154   Public   0.20   0.975 52
1469    7345  16.652  0.292   Male       No  978   Public   0.64   0.336 56
1509    7688  20.928  0.222   Male       No 1410 Catholic   0.65  -0.575 54
1556    7688  19.567 -1.278   Male       No 1410 Catholic   0.65  -0.575 54
1754    8627  16.660 -0.648   Male       No 2452   Public   0.25   0.742 53
          ses.m     ses.cwg ses.iqr ses.trange    ses.sd ses.rank
95   0.51162264 -0.22962264  1.0100      1.586 0.6135833     21.0
141  0.42316667  0.34883333  0.9825      1.505 0.5981188     38.5
200  0.22778947  0.27421053  0.7900      1.948 0.6584097     41.0
278 -0.13764912  0.48964912  1.0600      1.728 0.7063209     43.0
```

```
   281  -0.13764912 -0.63035088  1.0600      1.728 0.7063209     11.5
   1358 -0.09011538  0.42211538  0.7975      1.327 0.5743482     40.0
   1469  0.03325000  0.25875000  1.3550      1.930 0.8257296     29.0
   1509  0.18588889  0.03611111  0.8275      1.378 0.5644347     28.0
   1556  0.18588889 -1.46388889  0.8275      1.378 0.5644347      1.0
   1754  0.10483019 -0.75283019  0.8400      1.788 0.7077276      7.0
        minority.diff
   95      -0.3661667
   141           NaN
   200     -0.5820390
   278      0.7141667
   281      0.7141667
   1358    -0.0107971
   1469    -1.0707154
   1509    -0.1387347
   1556    -0.1387347
   1754    -0.7930556
```

```
head( dd )
```

```
     school mathach    ses    Sex Minority Size   Sector PRACAD DISCLIM  n
   1   1317  12.862  0.882 Female       No  455 Catholic   0.95  -1.694 48
```

```
2   1317   8.961  0.932 Female      Yes  455 Catholic   0.95  -1.694 48
3   1317   4.756 -0.158 Female      Yes  455 Catholic   0.95  -1.694 48
4   1317  21.405  0.362 Female      Yes  455 Catholic   0.95  -1.694 48
5   1317  20.748  1.372 Female       No  455 Catholic   0.95  -1.694 48
6   1317  18.362  0.132 Female      Yes  455 Catholic   0.95  -1.694 48
      ses.m     ses.cwg ses.iqr ses.trange    ses.sd ses.rank
1 0.3453333  0.53666667  0.7825      1.166 0.5561583       39
2 0.3453333  0.58666667  0.7825      1.166 0.5561583       40
3 0.3453333 -0.50333333  0.7825      1.166 0.5561583        6
4 0.3453333  0.01666667  0.7825      1.166 0.5561583       25
5 0.3453333  1.02666667  0.7825      1.166 0.5561583       47
6 0.3453333 -0.21333333  0.7825      1.166 0.5561583       20
  minority.diff
1    -0.2676044
2    -0.2676044
3    -0.2676044
4    -0.2676044
5    -0.2676044
6    -0.2676044
```

Looking at data in 3D —-

Visualizing the school level data again

```
if(interactive) {
  up( dd, ~ school) %>% xqplot
  up( dd, ~ school) %>% spm
}
```

QUESTIONS:

1. Would sample size be a reasonable proxy for school size if we did not have school size?
2. Can you think of other ways of summarizing or visualizing the data: what do you see?

1d - 2d -> 3d

We can see 3 continuous variable + 1 categorical variable (really 4d)

```
if(interactive){
  library( p3d )
  Init3d(family = 'serif', cex = 1.2)
  dda <- up(dd, ~ school, ~ mathach + ses + Sex)
  head(dda)
  Plot3d ( mathach ~ ses + Sex_Female | Sector, dda)
```

```
  fg()
}
```

Note gender composition of Catholic schools

```
# just for fun
if(interactive) {
  fit <- lm( mathach ~ ses * Sex_Female * Sector +
               I(ses*(Sector == 'Catholic')*Sex_Female^2), dda)
  summary(fit)
  Fit3d ( fit )
  Id3d(pad=2)
}
```

EXERCISE:

Try other possibilities. What do you find?

Looking at Level 1 and Level 2 data using Lattice graphics —-

Preparing for multilevel modelling:

```
Visualizing data at Level 1 and Level 2
```

From the mixed model to the hierarchical models:

With software like HLM, MlWin, you need to use separate data sets for the different levels.

With nlme and SAS PROC MIXED, we use the combined long data set with all the variables together.

We first need to separate the Levels of the model.

1. Level 1 model:

   ```
   mathach ~ 1 + ses | school
   ```

2. Level 2 model:

   - Using coefficients of Level 1 Model:
     - `B.0 ~ 1 + Sector`
     - `B.ses ~ 1 + Sector`
   - Combining with Level 1:
     - `~ 1 * ( 1 + Sector ) + ses * ( 1 + Sector )`

3. Mixed model: Simplify combined model

   ```
   mathach ~ 1 + ses + Sector + ses:Sector
   ```

   OR (using R's rules for generating marginal effects)

`mathach ~ ses * Sector`

4. Random effects model:

   `~ 1 + ses | id` – all terms usually contained in Level 1 model

   Often, a simpler model is used, e.g. '$\sim 1 \mid$ id"

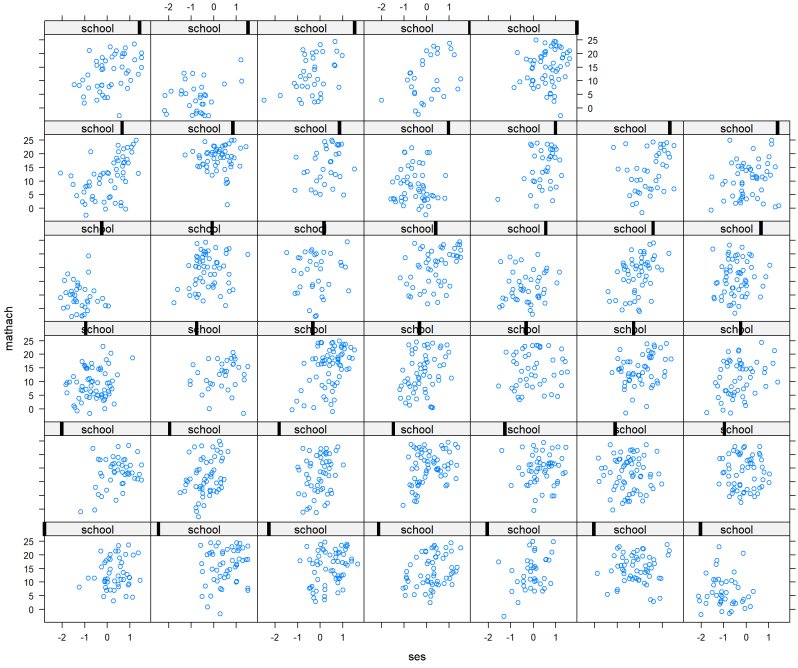
5. Compositional variable:

   Mean ses | school i.e `cvar( ses , school)`

   Mixed model with compositional variable:

   `mathach ~ 1 + ses + Sector + ses:Sector + cvar( ses, school) + ...`

```
xyplot( mathach ~ ses | school,  dd )
```


school
mathach
ses
-2 -1 0 1
0 5 10 15 20 25

```
# Make an informative label and ordering for schools

dd$id <- factor( paste( substring( dd$Sector, 1,1), dd$school, sep =''))
some( dd )
```

	school	mathach	ses	Sex	Minority	Size	Sector	PRACAD	DISCLIM	n
147	2208	20.628	0.792	Male	No	1061	Catholic	0.68	-0.864	60
362	2658	12.860	0.952	Female	No	780	Catholic	0.79	-0.961	45
842	4868	9.368	-0.158	Male	Yes	657	Catholic	1.00	-0.219	34
886	5619	1.701	0.472	Female	No	1118	Catholic	0.77	-1.286	66
1210	6074	19.882	1.512	Female	No	2051	Catholic	0.32	-1.018	56
1484	7345	9.098	-0.358	Female	Yes	978	Public	0.64	0.336	56
1495	7345	23.134	0.472	Female	No	978	Public	0.64	0.336	56
1584	7697	11.620	0.872	Male	No	1734	Public	0.20	0.279	32
1910	9550	-1.005	-0.508	Female	Yes	1532	Public	0.45	0.791	29
1958	9586	24.292	1.002	Female	No	262	Catholic	1.00	-2.416	59

	ses.m	ses.cwg	ses.iqr	ses.trange	ses.sd	ses.rank
147	0.42316667	0.36883333	0.9825	1.505	0.5981188	41.5
362	0.43844444	0.51355556	0.9100	1.698	0.6402846	37.0
842	0.36111765	-0.51911765	1.0475	1.920	0.7100080	10.0
886	0.42033333	0.05166667	0.7550	1.405	0.5972748	31.0
1210	-0.27675000	1.78875000	0.8175	1.440	0.6271235	56.0

```
   1484  0.03325000 -0.39125000  1.3550      1.930 0.8257296     18.0
   1495  0.03325000  0.43875000  1.3550      1.930 0.8257296     35.5
   1584  0.25825000  0.61375000  0.8625      1.438 0.6133712     29.0
   1910  0.05303448 -0.56103448  1.0200      1.824 0.7847035      9.0
   1958  0.62115254  0.38084746  0.8550      1.390 0.5949914     43.0
        minority.diff    id
   147            NaN C2208
   362    -0.18305556 C2658
   842    -0.77505263 C4868
   886            NaN C5619
   1210   -0.32796296 C6074
   1484   -1.07071545 P7345
   1495   -1.07071545 P7345
   1584           NaN P7697
   1910   -0.56239130 P9550
   1958    0.08517857 C9586
```

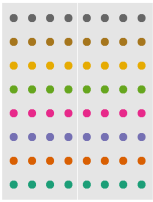
```
# ordering used for panels in xyplot:

dd$id <- reorder( dd$id, dd$ses)    # or

dd$id.sec <- reorder( dd$id, dd$ses + 1000* (dd$Sector == "Catholic"))
```

Note: If you are irritated by caps in variable names use:

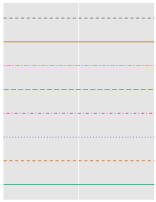
```
names(dd) <- tolower( names(dd) )
```

Make sure this won't create duplicates. Variable names are case sensitive in R

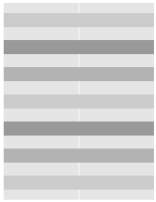
```
gd() # sets 8 default colours for multiple groups using colour-blind friendly co
show.settings() # first two panels show 'superpose' colours for multiple groups
```



superpose.symbol

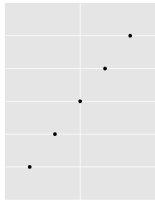


superpose.line

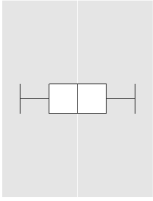


strip.background

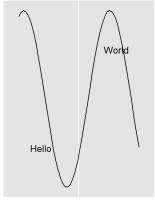
strip.shingle



dot.[symbol, line]

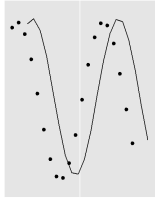


box.[dot, rectangle, umbrella]



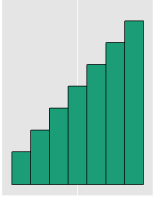
add.[line, text]

reference.line

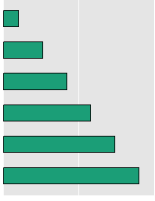


plot.[symbol, line]

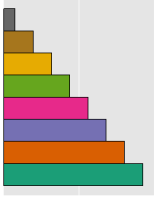
plot.shingle[plot.polygon]



histogram[plot.polygon]



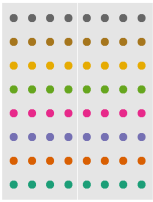
barchart[plot.polygon]



superpose.polygon

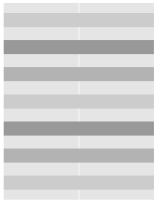
regions

```
gd(lwd=c(3,1), lty = c(1,1)) # if arguments have length > 1, superpose options a
show.settings() # first two panels show 'superpose' colours for multiple groups
```



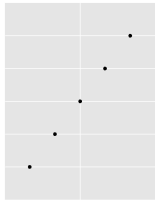
superpose.symbol

superpose.line

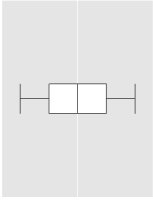


strip.background

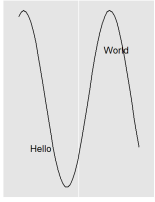
strip.shingle



dot.[symbol, line]

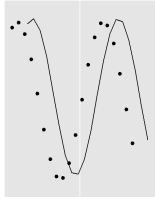


box.[dot, rectangle, umbrella]

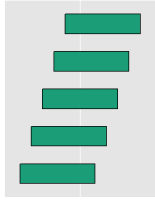


add.[line, text]

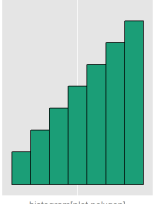
reference.line



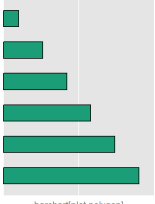
plot.[symbol, line]



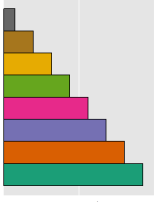
plot.shingle[plot.polygon]



histogram[plot.polygon]



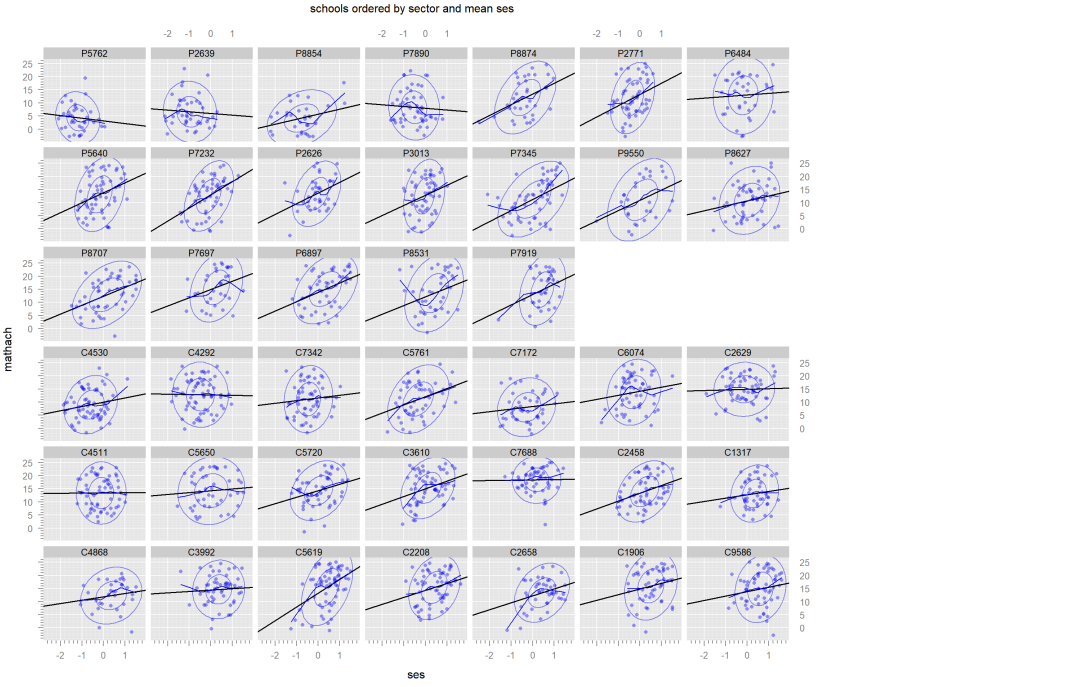
barchart[plot.polygon]



superpose.polygon

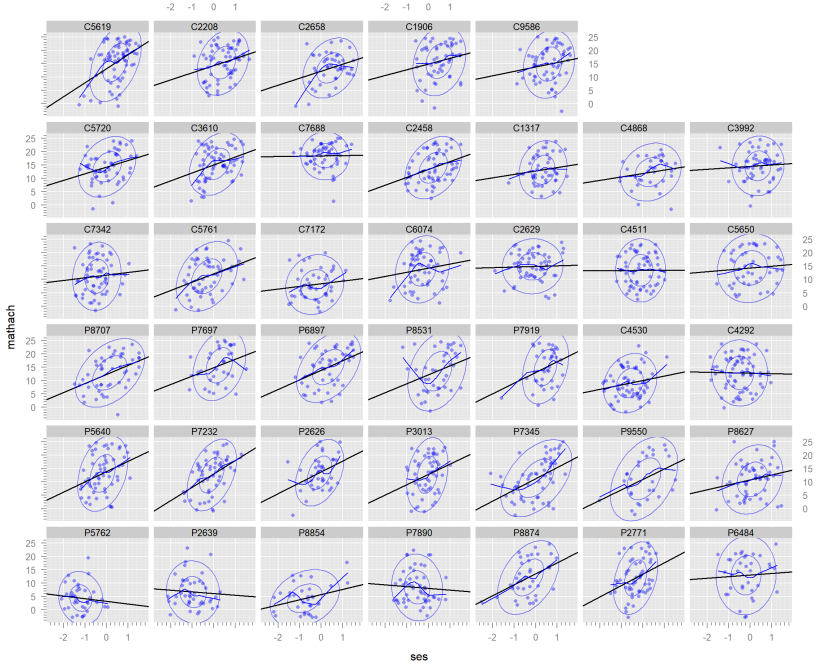
regions

```
gd(col = 'blue', pch = 16, alpha = 1)     # options for single group
xyplot( mathach ~ ses | id.sec, dd, layout = c(7,6),
        main = list('schools ordered by sector and mean ses', cex =1, font = 1),
        skip = c(rep(FALSE, 19), TRUE, TRUE, rep(FALSE, 21) ),
        as.table = TRUE,
        panel = function( x,y ,..., type, alpha, lwd) {
          panel.xyplot( x, y, ... , alpha = .4)
          panel.lines( dell( x, y, radius = 1:2), type = 'l', alpha = .5,...)
          panel.lmline( x, y, ..., lwd = 1.5, alpha = 1)
          panel.lines( lowess( x, y),..., type = 'l')
        }
)
```


schools ordered by sector and mean ses
P5762
P2639
P8854
P7890
P8874
P2771
P6484
P5640
P7232
P2626
P3013
P7345
P9550
P8627
P8707
P7697
P6897
P8531
P7919
C4530
C4292
C7342
C5761
C7172
C6074
C2629
C4511
C5650
C5720
C3610
C7688
C2458
C1317
C4868
C3992
C5619
C2208
C2658
C1906
C9586
mathach
ses

If you like a particular panel function, you can make it yours: R is easy to customize:

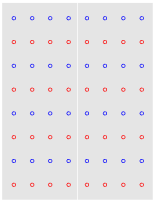
```
mypanel = function( x,y ,..., type, alpha, lwd) {
    panel.xyplot( x, y, ... , alpha = .4)
    panel.lines( dell( x, y, radius = 1:2), type = 'l', alpha = .5,...)
    panel.lmline( x, y, ..., lwd = 1.5, alpha = 1)
    panel.lines( lowess( x, y),..., type = 'l')
}

xyplot( mathach ~ ses | id.sec, dd,
        panel = mypanel)
```


C5619
C2208
C2658
C1906
C9586
C5720
C3610
C7688
C2458
C1317
C4868
C3992
C7342
C5761
C7172
C6074
C2629
C4511
C5650
P8707
P7697
P6897
P8531
P7919
C4530
C4292
P5640
P7232
P2626
P3013
P7345
P9550
P8627
P5762
P2639
P8854
P7890
P8874
P2771
P6484
mathach
ses

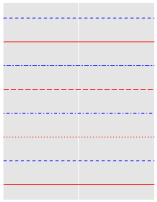
Use ‘groups’ to have more than one group per panel

Set options for groups by specifying vectors of length > 1, short vectors get recycled

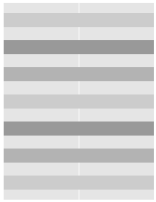
```
gd(col = c('red','blue'), pch = c(1,1), cex = c(.7,.7))
show.settings()
```



superpose.symbol

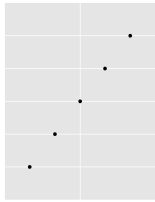


superpose.line

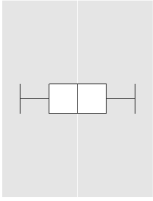


strip.background

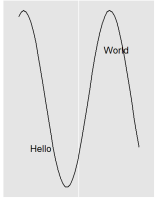
strip.shingle



dot.[symbol, line]

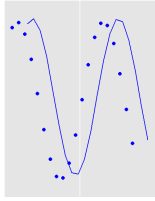


box.[dot, rectangle, umbrella]



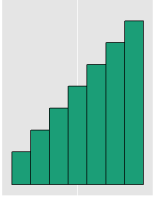
add.[line, text]

reference.line

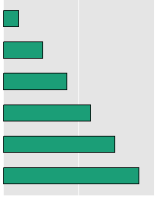


plot.[symbol, line]

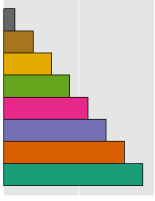
plot.shingle[plot.polygon]



histogram[plot.polygon]



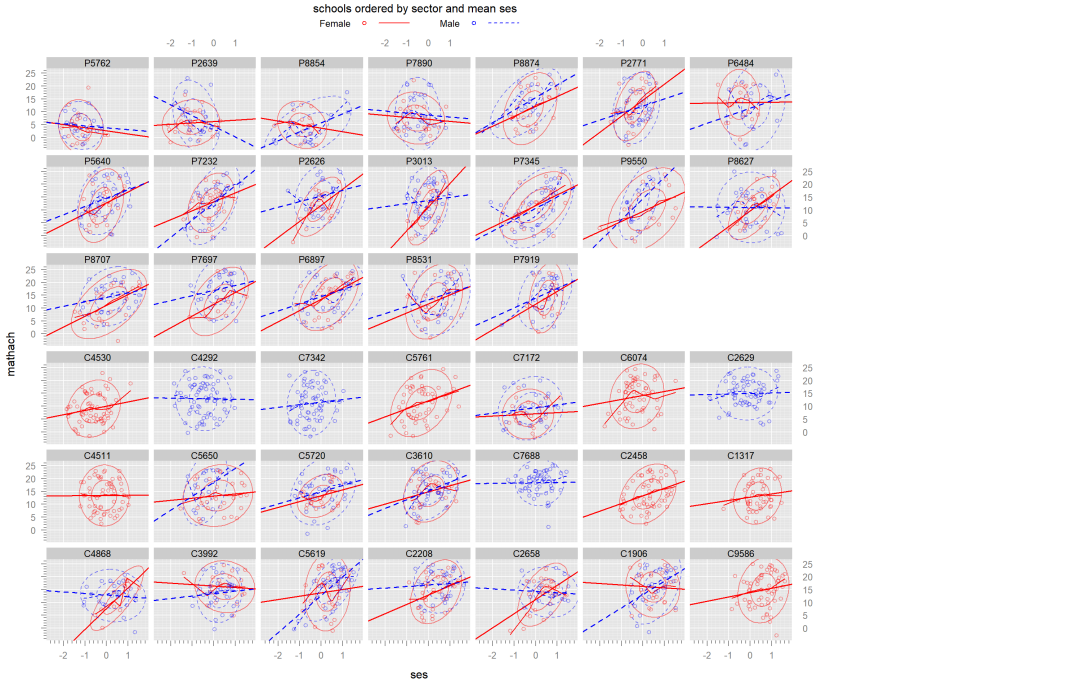
barchart[plot.polygon]



superpose.polygon

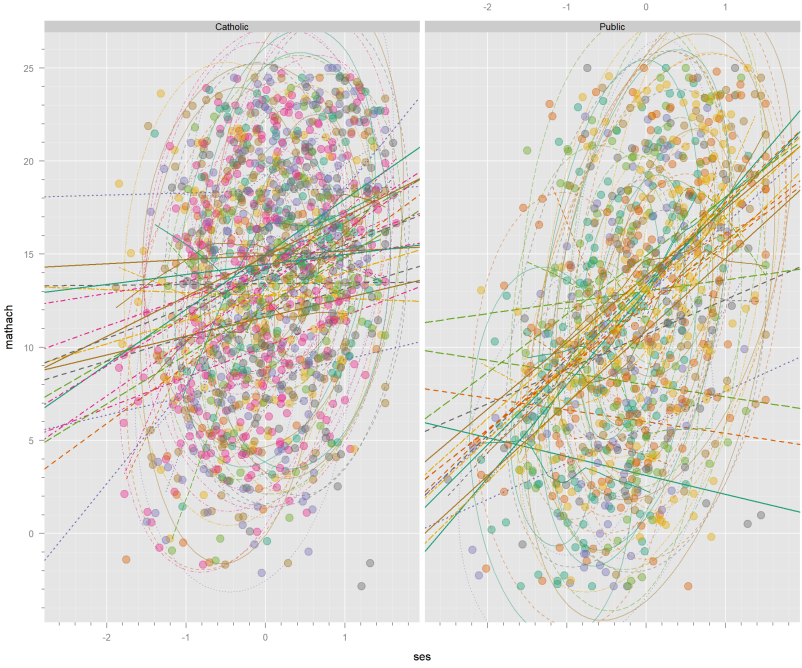
regions

```
xyplot( mathach ~ ses | id.sec,  dd, layout = c(7,6),
        groups = Sex,
        main = list('schools ordered by sector and mean ses', cex =1, font = 1),
        skip = c(rep(FALSE, 19), TRUE, TRUE, rep(FALSE, 21) ),
        as.table = TRUE,
        panel = panel.superpose,
        panel.groups = mypanel,
        auto.key = list(columns = 2, lines = T)
)
```


schools ordered by sector and mean ses
Female
Male
P5762
P2639
P8854
P7890
P8874
P2771
P6484
P5640
P7232
P2626
P3013
P7345
P9550
P8627
P8707
P7697
P6897
P8531
P7919
C4530
C4292
C7342
C5761
C7172
C6074
C2629
C4511
C5650
C5720
C3610
C7688
C2458
C1317
C4868
C3992
C5619
C2208
C2658
C1906
C9586
mathach
ses

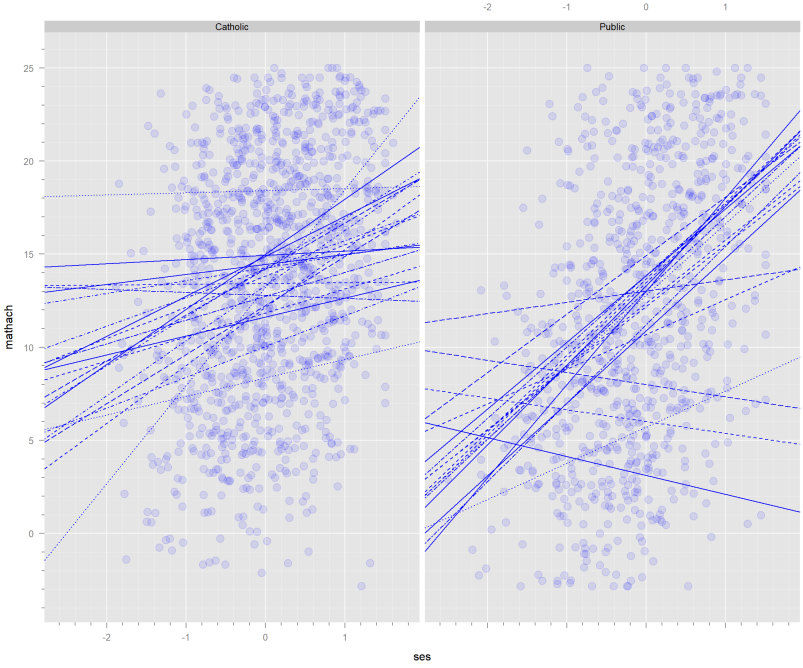
Combine panels and groups to see data in different ways:

```
gd()
xyplot( mathach ~ ses | Sector, dd,   groups = id,
        panel = panel.superpose,
        panel.groups = mypanel)
```


Catholic
Public
mathach
ses
25
20
15
10
5
0
-2
-1
0
1

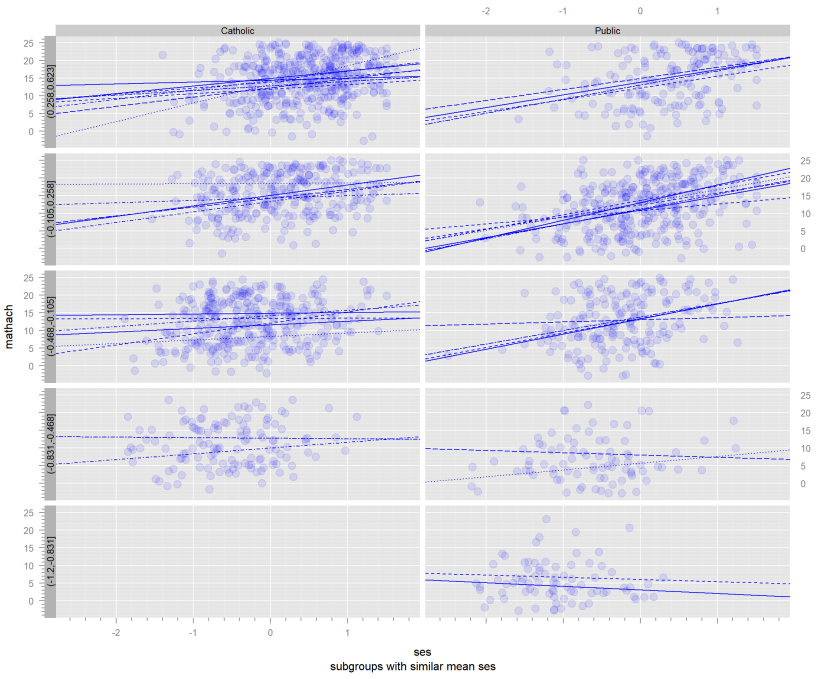
too much! Just data and fitted line:

```
mypanel2 <- function( x,y ,..., type, alpha, col) {
  panel.xyplot( x, y, ... ,alpha = alpha)
  panel.lmline( x, y, ..., alpha = 1, superpose.col = 'black')
}

gd(col=c('blue','blue'),alpha=c(.1,.1))
xyplot( mathach ~ ses | Sector, dd,   groups = id,
        panel = panel.superpose,
        panel.groups = mypanel2)
```


Catholic
Public
mathach
ses
25
20
15
10
5
0
-2
-1
0
1

fitted lines generally higher and flatter in Catholic sector

```
xyplot( mathach ~ ses | Sector * cut(ses.m, 5), dd, groups = id,
        sub = list('subgroups with similar mean ses', font=1, cex = 1),
        panel = panel.superpose,
        panel.groups = mypanel2) %>%
  useOuterStrips
```

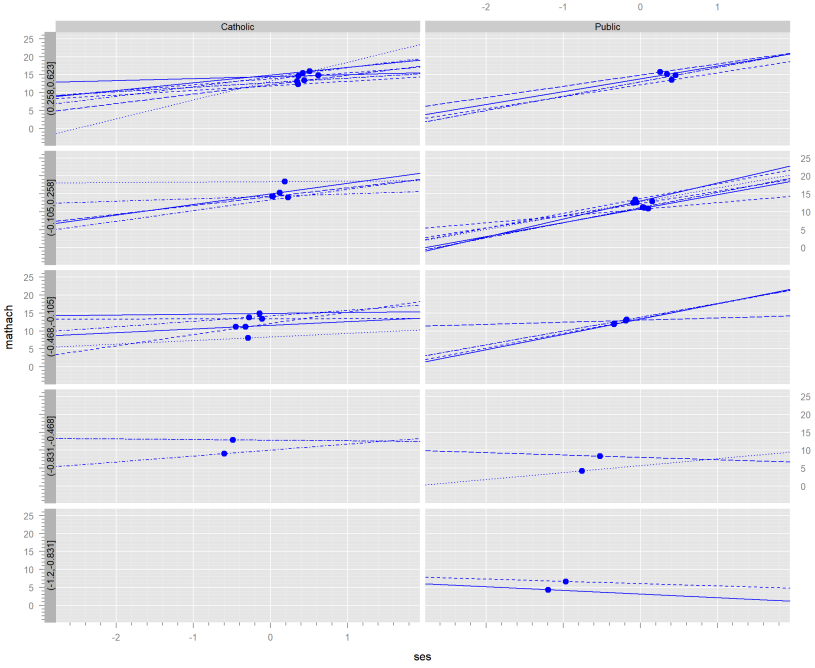

Catholic
Public
(0.258,0.623]
(-0.105,0.258]
(-0.468,-0.105]
(-0.831,-0.468]
(-1.2,-0.831]
mathach
ses
subgroups with similar mean ses

Note:

- fitted lines generally higher and flatter in Catholic sector

Center point and fitted lines

```
mypanel3 <- function( x,y ,..., type) {
  panel.xyplot( mean(x), mean(y), ... )
  panel.lmline( x, y, ...)
}
gd(col=c('blue','blue'), alpha = 1, cex = 1)
xyplot( mathach ~ ses | Sector * cut( ses.m, 5), dd,   groups = id,
        panel = panel.superpose,
        panel.groups = mypanel3) %>%
  useOuterStrips
```


Catholic
Public
(0.258,0.623]
(-0.105,0.258]
(-0.468,-0.105]
(-0.831,-0.468]
(-1.2,-0.831]
mathach
ses

Visualizing fitted lines in beta space —-

Fit an OLS model in each school

```
# fit.list <- lmList( mathach ~ ses | id, dd)

# may produce an error due to missing values in some variables in 'dd' that
# 'lmList' does not use. To work around this problem, you can use:

fit.list <- lmList( mathach ~ ses | id, dd[,c('mathach','ses','id')])
fit.list
```

```
   Call:
     Model: mathach ~ ses | id
      Data: dd[, c("mathach", "ses", "id")]

   Coefficients:
         (Intercept)         ses
   P5762    3.114085 -1.01409923
   P2639    6.007785 -0.63010463
   P8854    5.707002  1.93884461
   C4530   10.039029  1.64742597
```

P7890	7.998220	-0.65596791
C4292	12.786274	-0.16060800
C7342	11.619820	1.01245787
P8874	13.450957	4.09630389
P2771	13.292965	4.26818800
C5761	12.141945	3.10801055
C7172	8.355037	0.99448053
C6074	14.202264	1.52908771
P6484	13.024537	0.60567730
P5640	13.836071	3.82774230
C2629	14.938378	0.22234915
C4511	13.413589	0.04251038
P7232	12.993356	5.00160032
P2626	13.662437	4.09967961
P3013	12.780651	3.83979913
C5650	14.258257	0.68061888
C5720	14.201984	2.46630669
P7345	11.198507	4.21192377
P9550	10.882731	3.89193777
P8627	10.687731	1.86955959
C3610	14.999882	2.95584910

```
   P8707   12.357826  3.39153230
   C7688   18.400688  0.11634493
   C2458   13.312180  2.95669443
   P7697   14.911853  3.13621972
   C1317   12.737763  1.27391282
   P6897   13.846070  3.58048769
   C4868   11.845609  1.28647122
   C3992   14.448670  0.53787533
   P8531   12.174522  3.31822792
   C5619   13.206326  5.25753341
   C2208   14.288928  2.63664069
   C2658   12.243090  2.62990138
   P7919   13.024135  3.98937031
   C1906   14.885481  2.14550546
   C9586   13.825077  1.67208118

   Degrees of freedom: 1977 total; 1897 residual
   Residual standard error: 6.106947
```

```
levels( dd$id )
```

```
    [1] "P5762" "P2639" "P8854" "C4530" "P7890" "C4292" "C7342" "P8874"
```

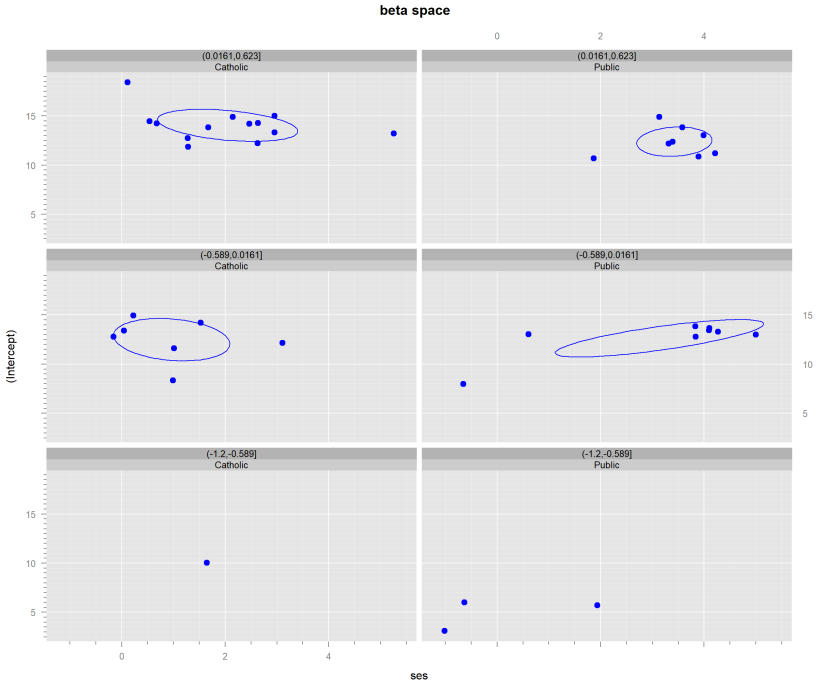
```
 [9] "P2771" "C5761" "C7172" "C6074" "P6484" "P5640" "C2629" "C4511"
[17] "P7232" "P2626" "P3013" "C5650" "C5720" "P7345" "P9550" "P8627"
[25] "C3610" "P8707" "C7688" "C2458" "P7697" "C1317" "P6897" "C4868"
[33] "C3992" "P8531" "C5619" "C2208" "C2658" "P7919" "C1906" "C9586"
```

```
beta <- expand.grid( id = levels( dd$id ))
beta <- cbind( beta, coef( fit.list))
some ( beta )
```

```
           id (Intercept)       ses
C7342 C7342   11.619820 1.0124579
P8874 P8874   13.450957 4.0963039
C5761 C5761   12.141945 3.1080106
C7172 C7172    8.355037 0.9944805
C2629 C2629   14.938378 0.2223492
C5720 C5720   14.201984 2.4663067
C3610 C3610   14.999882 2.9558491
C2458 C2458   13.312180 2.9566944
C1317 C1317   12.737763 1.2739128
C2658 C2658   12.243090 2.6299014
```

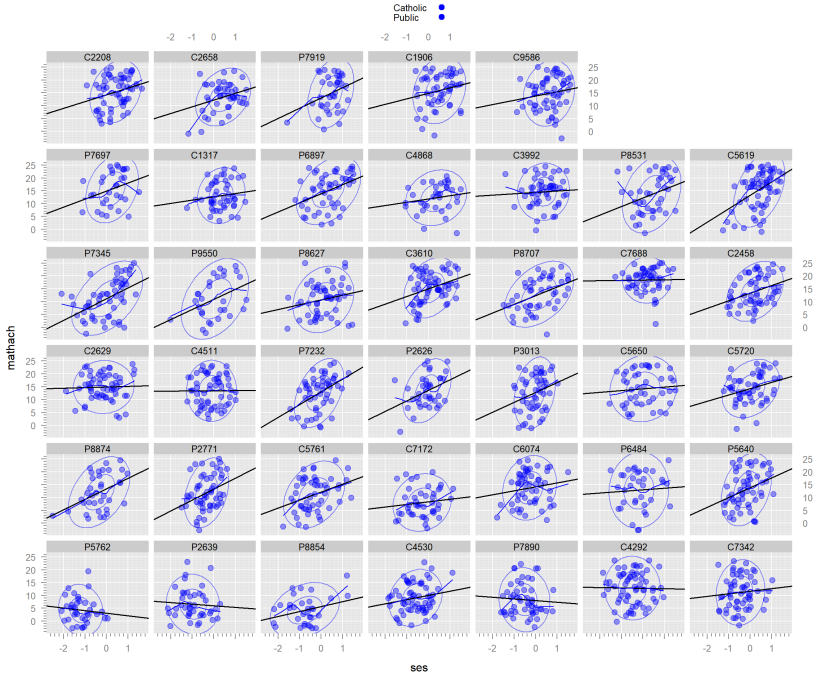
```
beta <- merge( beta, up(dd, ~ id))
```

```
xyplot( `(Intercept)` ~ ses | Sector * cut( ses.m, 3), beta,  groups = Sector,
        panel = function(x, y, ..., type){
          panel.xyplot( x,y , ...)
          if ( length(x) > 3)panel.lines( dell( x, y), ..., type = 'l')
        },
        main = 'beta space'
)
```


beta space
(0.0161,0.623]
Catholic
(0.0161,0.623]
Public
(-0.589,0.0161]
Catholic
(-0.589,0.0161]
Public
(-1.2,-0.589]
Catholic
(-1.2,-0.589]
Public
(Intercept)
ses
0
2
4
5
10
15

Looking at between group effect —-

```
xyplot( mathach ~ ses | id, dd,  layout = c(7,6),  groups = Sector,
        panel = mypanel, auto.key = T)
```

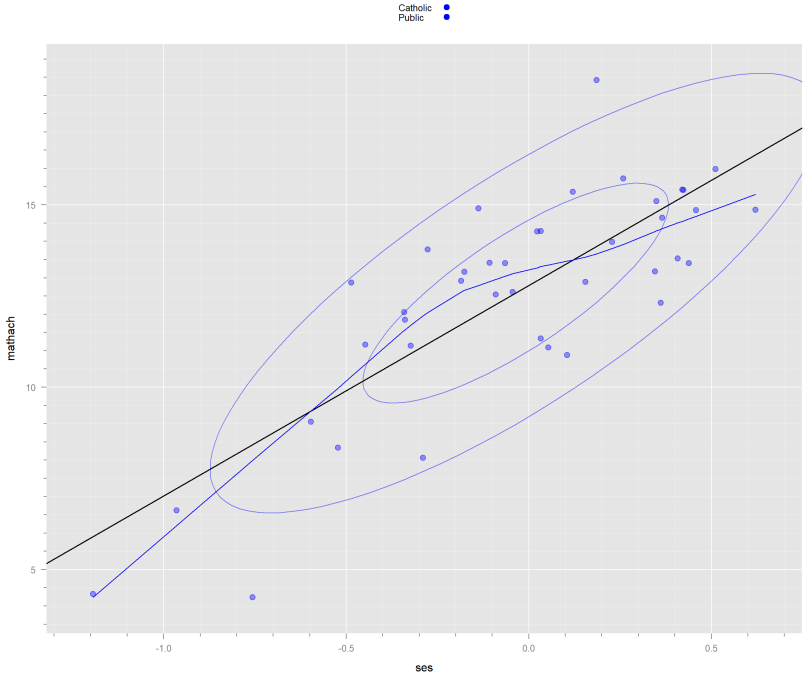

Catholic
Public
C2208
C2658
P7919
C1906
C9586
P7697
C1317
P6897
C4868
C3992
P8531
C5619
P7345
P9550
P8627
C3610
P8707
C7688
C2458
C2629
C4511
P7232
P2626
P3013
C5650
C5720
P8874
P2771
C5761
C7172
C6074
P6484
P5640
P5762
P2639
P8854
C4530
P7890
C4292
C7342
mathach
ses

```
dda <- up(dd, ~ id, all = T)  # means of numerical variables, modes of factors
some(dda)
```

```
       school   mathach         ses    Sex Minority Size   Sector PRACAD
C7342    7342 11.166414 -0.44782759   Male       No 1220 Catholic   0.46
C5761    5761 11.138058 -0.32300000 Female       No  215 Catholic   0.63
C7172    7172  8.066818 -0.28981818 Female      Yes  280 Catholic   0.05
P2626    2626 13.396605 -0.06484211   Male       No 2142   Public   0.40
P3013    3013 12.610830 -0.04422642   Male       No  760   Public   0.56
P8627    8627 10.883717  0.10483019   Male       No 2452   Public   0.25
P8707    8707 12.883938  0.15512500 Female       No 1133   Public   0.48
C4868    4868 12.310176  0.36111765   Male       No  657 Catholic   1.00
P8531    8531 13.528683  0.40809756 Female       No 2190   Public   0.58
P7919    7919 14.849973  0.45767568   Male       No 1451   Public   0.50
       DISCLIM  n       ses.m       ses.cwg ses.iqr ses.trange    ses.sd
C7342    0.380 58 -0.44782759 -4.974128e-19  0.7125      1.323 0.5648459
C5761   -0.892 52 -0.32300000 -5.358461e-18  0.8000      1.917 0.7122389
C7172    1.013 44 -0.28981818 -3.027573e-17  1.0275      1.686 0.6764417
P2626    0.142 38 -0.06484211  6.211337e-18  0.5975      1.384 0.5601067
P3013   -0.213 53 -0.04422642  1.112228e-17  0.7300      1.168 0.4799328
P8627    0.742 53  0.10483019 -6.563514e-18  0.8400      1.788 0.7077276
P8707    1.542 48  0.15512500  8.673617e-19  1.1350      2.169 0.8042577
```

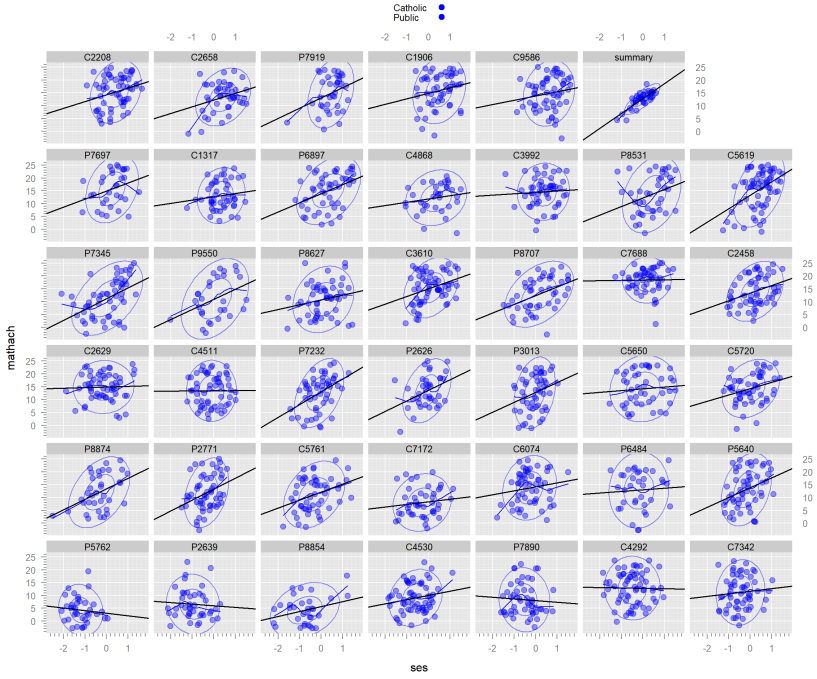
```
   C4868  -0.219 34  0.36111765 -6.555484e-18  1.0475      1.920 0.7100080
   P8531   0.132 41  0.40809756 -1.152428e-17  1.0600      1.650 0.6829747
   P7919  -0.402 37  0.45767568 -1.351883e-17  0.6700      1.070 0.5367005
         ses.rank minority.diff    id id.sec
   C7342     29.5     0.1027413 C7342  C7342
   C5761     26.5     0.1197576 C5761  C5761
   C7172     22.5     0.3915447 C7172  C7172
   P2626     19.5           NaN P2626  P2626
   P3013     27.0    -0.6765385 P3013  P3013
   P8627     27.0    -0.7930556 P8627  P8627
   P8707     24.5    -0.5873333 P8707  P8707
   C4868     17.5    -0.7750526 C4868  C4868
   P8531     21.0    -0.4734848 P8531  P8531
   P7919     19.0     0.3127778 P7919  P7919
```

```
xyplot( mathach ~ ses, dda ,   groups = Sector,
       panel = mypanel, auto.key = T)
```


Catholic
Public
15
10
5
mathach
-1.0
-0.5
0.0
0.5
ses

Seeing both together:

```
dda$id <- 'summary'
gd(pch='.', cex =2)
xyplot( mathach ~ ses | id, rbind(dd,dda),  layout = c(7,6),
        groups = Sector,
        panel = mypanel, auto.key = T)
```


Catholic
Public
C2208
C2658
P7919
C1906
C9586
summary
P7697
C1317
P6897
C4868
C3992
P8531
C5619
P7345
P9550
P8627
C3610
P8707
C7688
C2458
C2629
C4511
P7232
P2626
P3013
C5650
C5720
P8874
P2771
C5761
C7172
C6074
P6484
P5640
P5762
P2639
P8854
C4530
P7890
C4292
C7342
mathach
ses

EXERCISE:

Try groups = Sex or Minority and try '| id.sec

Do these plots reveal anything useful?

Fitting a mixed model —-

Question to investigate:

```
mathach ~ ses  in differenct Sectors
```

Level 1 model:

```
mathach ~ 1 + ses
```

Level 1 + 2:

```
mathach ~ 1 + ses + Sector + Sector:ses
```

Full random model:

```
~ 1 + ses | id
```

```
dd$id <- factor(dd$school)
fit <- lme( mathach ~ ses * Sector, dd, random = ~ 1 + ses | id)
```

If this fails to converge, increase the number of iteration

Convergence problems —-

```
fit <- lme( mathach ~ ses * Sector, dd, random = ~ 1 + ses | id,
            control = list(msMaxIter=200, msVerbose=T))
```

```
     0:     11105.191:  1.22722  2.27855 -3.36583
     1:     11105.172:  1.23706  3.08949 -3.39049
     2:     11105.171:  1.19748  3.08072 -3.39188
     3:     11105.161:  1.21737  3.07678 -3.39235
     4:     11105.096:  1.22151  2.70896 -3.41493
     5:     11105.089:  1.22011  2.64468 -3.43555
     6:     11105.085:  1.22389  2.55034 -3.50479
     7:     11105.083:  1.22385  2.57106 -3.54636
     8:     11105.076:  1.22534  2.60016 -3.72983
     9:     11105.057:  1.23381  2.66214 -4.47027
    10:     11105.033:  1.25234  2.72319 -5.87758
    11:     11105.019:  1.27696  2.75593 -7.28573
    12:     11105.013:  1.30246  2.78019 -8.50888
    13:     11105.011:  1.31965  2.79844 -9.36213
```

```
    14:     11105.011:  1.33116  2.81675 -9.93866
    15:     11105.011:  1.33714  2.82776 -10.3398
    16:     11105.011:  1.33993  2.83733 -10.5086
    17:     11105.011:  1.33984  2.84079 -10.5414
    18:     11105.011:  1.33908  2.84283 -10.5393
    19:     11105.011:  1.33903  2.84278 -10.5371
```

```
print(fit)
```

```
  Linear mixed-effects model fit by REML
    Data: dd
    Log-restricted-likelihood: -6419.695
    Fixed: mathach ~ ses * Sector
       (Intercept)              ses     SectorPublic ses:SectorPublic
         13.479564         1.805347        -1.599373         1.388797

  Random effects:
   Formula: ~1 + ses | id
   Structure: General positive-definite, Log-Cholesky parametrization
               StdDev    Corr
  (Intercept) 1.8816109 (Intr)
  ses         0.3564559 0.523
```

```
   Residual    6.1180179

   Number of Observations: 1977
   Number of Groups: 40
```

Things we can do with a model

```
summary( fit )
```

```
   Linear mixed-effects model fit by REML
    Data: dd
          AIC      BIC    logLik
     12855.39 12900.09 -6419.695

   Random effects:
    Formula: ~1 + ses | id
    Structure: General positive-definite, Log-Cholesky parametrization
                StdDev    Corr
   (Intercept) 1.8816109 (Intr)
   ses         0.3564559 0.523
   Residual    6.1180179

   Fixed effects: mathach ~ ses * Sector
```

```
                       Value Std.Error   DF   t-value p-value
(Intercept)       13.479564 0.4502225 1935 29.939782  0.0000
ses                1.805347 0.2905759 1935  6.212995  0.0000
SectorPublic      -1.599373 0.6628232   38 -2.412971  0.0208
ses:SectorPublic   1.388797 0.4359721 1935  3.185518  0.0015
 Correlation:
                  (Intr) ses    SctrPb
ses                0.094
SectorPublic      -0.679 -0.064
ses:SectorPublic -0.063 -0.667  0.163

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.06175080 -0.74559046  0.03508812  0.78804832  2.73786882

Number of Observations: 1977
Number of Groups: 40
```

Sequential tests

```
anova(fit)
```

```
             numDF denDF   F-value p-value
```

```
(Intercept)     1  1935 1376.8723  <.0001
ses             1  1935  129.5570  <.0001
Sector          1    38    8.8271  0.0051
ses:Sector      1  1935   10.1475  0.0015
```

Type II tests

- Each term added last EXCLUDING interactions that include it
- Test that respect the principle of marginality

```
Anova(fit)
```

```
Analysis of Deviance Table (Type II tests)

Response: mathach
              Chisq Df Pr(>Chisq)
ses        125.0347  1  < 2.2e-16 ***
Sector       8.8271  1   0.002968 **
ses:Sector  10.1475  1   0.001445 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Overall test for a variable including all its interactions

- Quick and rough: using spida2::wald
- Conservative denominator DFs

```
wald(fit, 'ses')      # overall test for 'ses'
```

```
       numDF denDF  F.value p.value
   ses     2  1935 67.59113 <.00001

   Coefficients       Estimate Std.Error   DF  t-value p-value Lower 0.95
     ses              1.805347  0.290576 1935 6.212995 <.00001   1.235472
     ses:SectorPublic 1.388797  0.435972 1935 3.185518 0.00147   0.533773

   Coefficients       Upper 0.95
     ses                2.375222
     ses:SectorPublic   2.243822
```

```
wald(fit, 'Sector')   # overall test for 'Sector'
```

```
          numDF denDF  F.value p.value
   Sector     2    38 9.487323 0.00045

   Coefficients        Estimate Std.Error   DF   t-value p-value Lower 0.95
     SectorPublic     -1.599373  0.662823   38 -2.412971 0.02075  -2.941188
```

```
    ses:SectorPublic  1.388797  0.435972 1935  3.185518 0.00147   0.533773

  Coefficients       Upper 0.95
    SectorPublic      -0.257558
    ses:SectorPublic   2.243822
```

Using car::lht

```
fixef(fit)
```

```
       (Intercept)              ses     SectorPublic ses:SectorPublic
         13.479564         1.805347        -1.599373         1.388797
```

```
coefs <- names(fixef(fit))
coefs
```

```
  [1] "(Intercept)"      "ses"              "SectorPublic"
  [4] "ses:SectorPublic"
```

```
lht(fit, coefs[c(2,4)])
```

```
  Linear hypothesis test

  Hypothesis:
  ses = 0
```

```
  ses:SectorPublic = 0

  Model 1: restricted model
  Model 2: mathach ~ ses * Sector

    Df  Chisq Pr(>Chisq)
  1
  2  2 135.18  < 2.2e-16 ***
  ---
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lht(fit, coefs[c(3,4)])
```

```
  Linear hypothesis test

  Hypothesis:
  SectorPublic = 0
  ses:SectorPublic = 0

  Model 1: restricted model
  Model 2: mathach ~ ses * Sector
```

```
  Df  Chisq Pr(>Chisq)
1
2  2 18.975  7.581e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that this model tacitly assumes that the between-school effect of ses is the same as the within-school effect of ses

In passing: handling NAs —-

This is a big topic but note that deleting rows with NAs does not delete entire clusters, only units within clusters. This is appropriate under a wider range of assumptions than deleting clusters – as one would have to do with a repeated measures analysis. More on this in Lab 4.R

Syntax

```
fit <- lme( mathach ~ ses * Sector, dd, random = ~ 1 + ses | id,
            na.action = na.exclude,
            control = list(msMaxIter=200, msVerbose=T))
```

```
     0:      11105.191:  1.22722  2.27855 -3.36583
     1:      11105.172:  1.23706  3.08949 -3.39049
```

```
 2:     11105.171:  1.19748  3.08072 -3.39188
 3:     11105.161:  1.21737  3.07678 -3.39235
 4:     11105.096:  1.22151  2.70896 -3.41493
 5:     11105.089:  1.22011  2.64468 -3.43555
 6:     11105.085:  1.22389  2.55034 -3.50479
 7:     11105.083:  1.22385  2.57106 -3.54636
 8:     11105.076:  1.22534  2.60016 -3.72983
 9:     11105.057:  1.23381  2.66214 -4.47027
10:     11105.033:  1.25234  2.72319 -5.87758
11:     11105.019:  1.27696  2.75593 -7.28573
12:     11105.013:  1.30246  2.78019 -8.50888
13:     11105.011:  1.31965  2.79844 -9.36213
14:     11105.011:  1.33116  2.81675 -9.93866
15:     11105.011:  1.33714  2.82776 -10.3398
16:     11105.011:  1.33993  2.83733 -10.5086
17:     11105.011:  1.33984  2.84079 -10.5414
18:     11105.011:  1.33908  2.84283 -10.5393
19:     11105.011:  1.33903  2.84278 -10.5371
```

Note: generally 'na.exclude' is preferable to 'na.omit' It allows residuals and predicted values to match the original data frame, not just the observations that were not missing.

In passing: What to do if the model does not converge

We will revisit this when we need it:

Summary:

1. If Iteration limit reached: try increase iterations.

```
?lmeControl
lmeControl

fit <- lme( mathach ~ ses * Sector, dd, random = ~ 1 + ses | id,
    control = list( maxIter = 200, msMaxIter = 200, niterEM = 100,
                msVerbose = TRUE , returnObject = TRUE ))
summary( fit )
```

2. If singular convergence, the frequent cause is that the random model is more complex than the cluster size or the variability between clusters can support. I.e. the variability from cluster to cluster that you are trying to account for with the random part of the model is actually accounted for by the 'epsilons', the level 1 or within-cluster variability. Consequently a variance is estimated to be 0 or the G matris is non-singular, i.e. it's a flat pancake.

   As a rule of thumb, if the random effects model has K terms, you need, for a balanced

within cluster design, a good deal more than K clusters with at least K observations and some with more than K observations. Otherwise, the G matrix and the within-cluster variance may not be identifiable.

Also, the G matrix might be relatively flat because of scaling OR centering. For example, one dimension is so much more stretched than another.

Try:

1. Rescale variables so they have similar variability or similar variability in slopes.
2. Global center x's in RE model (not necessarily at the mean but that's a good first try. The 'ideal' place is the point of minimum variance of the response. Plot OLS lines to get an idea.
3. Also try centering within clusters (CWC or CWG). The RE model is not equivalent but it could often be better depending on the process. Use AIC to compare.,
4. If nothing works, start with simple RE model and add components using a manual forward stepwise approach. Use 'anova' and 'simulate' to test additional components. If a model does not converge, use ..., control = (...., returnObject = TRUE, msVerbose = TRUE) to judge whether the likelihood ratio converges even though the parameters don't. In this case, anova can give a usable p-value that can be adjusted with 'simulate'.

```
‘fit.big <- lme( ...., control = list( maxIter = 200,
‘                msMaxIter = 200, niterEM = 100,
‘                msVerbose = TRUE , returnObject = TRUE ))‘
‘anova ( fit.simple, fit.big )‘
‘zsim <- simulate( fit.simple, nsim = 1000, m2 = fit.big)‘
‘plot( zsim )‘
```

to test whether ‘fit.simple’ is adequate.

Handling non-convergence is discussed in greater detail in Lab Session 3

Hausman Test: Is the between effect different from the within effect?

—-

First diagnostics: Hausman Test

After fitting a model: diagnostics

- Historically: The Hausman test asks whether the mixed model correctly estimates the effect of ses
- But in fact:
  - Do we need a contextual variable? i.e.

- Is the between-school effect of ses not significantly different from the within-school effect of ses
- If so, then we should include a separate effect for the between-school effect of ses and we will be able to estimate the within school effect correctly.

Model with contextual variable for ses, i.e. each school's mean ses: Note: we can use the cvar function for convenience or we can create a variable in the data frame, i.e. ses.m above. We will use cvar because it's always easy to use even if the variable has not been defined.

Moreover, cvar works correctly with factors generating mean values of indicators for each level of the factor omitting the reference level. We will illustrate this later.

Fitting a model with a contextual mean —-

```
fitc <- lme( mathach ~ (ses + cvar(ses,id))* Sector, dd,
             random = ~ 1 + ses | id )
summary( fitc )
```

```
   Linear mixed-effects model fit by REML
    Data: dd
         AIC      BIC    logLik
     12839.6 12895.47 -6409.801
```

```
Random effects:
 Formula: ~1 + ses | id
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.4561519 (Intr)
ses         0.5190639 -0.217
Residual    6.1140010

Fixed effects: mathach ~ (ses + cvar(ses, id)) * Sector
                              Value Std.Error   DF  t-value p-value
(Intercept)                13.356053 0.3753744 1935 35.58062  0.0000
ses                         1.699397 0.3095543 1935  5.48982  0.0000
cvar(ses, id)               2.407785 1.0629298   36  2.26523  0.0296
SectorPublic               -0.934345 0.5628780   36 -1.65994  0.1056
ses:SectorPublic            1.173092 0.4677016 1935  2.50821  0.0122
cvar(ses, id):SectorPublic  1.414896 1.4652174   36  0.96566  0.3407
 Correlation:
                           (Intr) ses    cv(,i) SctrPb ss:ScP
ses                        -0.071
cvar(ses, id)              -0.186 -0.248
```

```
   SectorPublic                -0.667  0.047  0.124
   ses:SectorPublic             0.047 -0.662  0.164 -0.065
   cvar(ses, id):SectorPublic   0.135  0.180 -0.725  0.056 -0.284

   Standardized Within-Group Residuals:
           Min           Q1          Med           Q3          Max
   -3.09882513 -0.75336271  0.03194452  0.78173508  2.79671295

   Number of Observations: 1977
   Number of Groups: 40
```

```
wald( fitc )
```

```
    numDF denDF  F.value p.value
        6    36 407.3664 <.00001

   Coefficients                  Estimate Std.Error   DF   t-value p-value
     (Intercept)                13.356053  0.375374 1935 35.580620 <.00001
     ses                         1.699397  0.309554 1935  5.489821 <.00001
     cvar(ses, id)               2.407785  1.062930   36  2.265235 0.02961
     SectorPublic               -0.934345  0.562878   36 -1.659942 0.10561
     ses:SectorPublic            1.173092  0.467702 1935  2.508206 0.01222
```

```
                              cvar(ses, id):SectorPublic  1.414896  1.465217   36  0.965656 0.34066

  Coefficients                Lower 0.95 Upper 0.95
    (Intercept)                12.619872  14.092234
    ses                         1.092302   2.306492
    cvar(ses, id)               0.252064   4.563507
    SectorPublic               -2.075914   0.207225
    ses:SectorPublic            0.255840   2.090344
    cvar(ses, id):SectorPublic -1.556703   4.386494
```

```
wald( fitc, -1)      # overall test of FE model
```

```
    numDF denDF  F.value p.value
1       5    36 33.98978 <.00001

                                Estimate Std.Error   DF   t-value p-value
    ses                         1.699397  0.309554 1935  5.489821 <.00001
    cvar(ses, id)               2.407785  1.062930   36  2.265235 0.02961
    SectorPublic               -0.934345  0.562878   36 -1.659942 0.10561
    ses:SectorPublic            1.173092  0.467702 1935  2.508206 0.01222
    cvar(ses, id):SectorPublic  1.414896  1.465217   36  0.965656 0.34066
```

```
                                Lower 0.95 Upper 0.95
    ses                           1.092302   2.306492
    cvar(ses, id)                 0.252064   4.563507
    SectorPublic                 -2.075914   0.207225
    ses:SectorPublic              0.255840   2.090344
    cvar(ses, id):SectorPublic   -1.556703   4.386494
```

```
wald( fitc, 'cvar')
```

```
       numDF denDF  F.value p.value
  cvar     2    36 9.749657 0.00041

  Coefficients                  Estimate Std.Error DF  t-value p-value
    cvar(ses, id)               2.407785  1.062930 36 2.265235 0.02961
    cvar(ses, id):SectorPublic 1.414896  1.465217 36 0.965656 0.34066

  Coefficients                  Lower 0.95 Upper 0.95
    cvar(ses, id)                 0.252064   4.563507
    cvar(ses, id):SectorPublic   -1.556703   4.386494
```

```
wald( fitc, 'ses')  # overall test of all 'ses' effects
```

```
      numDF denDF  F.value p.value
```

```
   ses     4    36 39.23784 <.00001

   Coefficients                  Estimate Std.Error   DF  t-value p-value
     ses                         1.699397  0.309554 1935 5.489821 <.00001
     cvar(ses, id)               2.407785  1.062930   36 2.265235 0.02961
     ses:SectorPublic            1.173092  0.467702 1935 2.508206 0.01222
     cvar(ses, id):SectorPublic  1.414896  1.465217   36 0.965656 0.34066

   Coefficients                  Lower 0.95 Upper 0.95
     ses                           1.092302   2.306492
     cvar(ses, id)                 0.252064   4.563507
     ses:SectorPublic              0.255840   2.090344
     cvar(ses, id):SectorPublic   -1.556703   4.386494
```

```
wald( fitc, 'Sector')  # overall test of all 'Sector' effects
```

```
          numDF denDF  F.value p.value
   Sector     3    36 3.939573 0.01576

   Coefficients                   Estimate Std.Error   DF   t-value p-value
     SectorPublic                -0.934345  0.562878   36 -1.659942 0.10561
     ses:SectorPublic             1.173092  0.467702 1935  2.508206 0.01222
```

```
    cvar(ses, id):SectorPublic  1.414896  1.465217   36  0.965656 0.34066

  Coefficients                 Lower 0.95 Upper 0.95
    SectorPublic                -2.075914   0.207225
    ses:SectorPublic             0.255840   2.090344
    cvar(ses, id):SectorPublic  -1.556703   4.386494
```

```
wald( fitc, ':')  # overall test of all interaction effects
```

```
    numDF denDF  F.value p.value
:       2    36 4.675875 0.01566

  Coefficients                 Estimate Std.Error   DF  t-value p-value
    ses:SectorPublic           1.173092  0.467702 1935 2.508206 0.01222
    cvar(ses, id):SectorPublic 1.414896  1.465217   36 0.965656 0.34066

  Coefficients                 Lower 0.95 Upper 0.95
    ses:SectorPublic             0.255840   2.090344
    cvar(ses, id):SectorPublic  -1.556703   4.386494
```

```
                # i.e. are lines parallel between sectors
```

Role of contextual variable for ses —-

Including cvar(ses, id) has two important consequences:

1) It unbiases the estimated effect of ses so it estimates
   the WITHIN-SCHOOL effect of ses unbiased by the between
   school effect of ses

2) It provides an estimate of the 'contextual effect' of ses
   and of the BETWEEN-SCHOOL (= compositional) effect of ses which
   is equal to the sum of the contextual and the within-effect

Interpreting the model with contextual effect —-

```
wald( fitc )
```

```
    numDF denDF  F.value p.value
        6    36 407.3664 <.00001

   Coefficients                  Estimate Std.Error   DF   t-value p-value
     (Intercept)                13.356053  0.375374 1935 35.580620 <.00001
```

```
    ses                          1.699397  0.309554 1935  5.489821 <.00001
    cvar(ses, id)                2.407785  1.062930   36  2.265235 0.02961
    SectorPublic                -0.934345  0.562878   36 -1.659942 0.10561
    ses:SectorPublic             1.173092  0.467702 1935  2.508206 0.01222
    cvar(ses, id):SectorPublic   1.414896  1.465217   36  0.965656 0.34066

  Coefficients                  Lower 0.95 Upper 0.95
    (Intercept)                  12.619872  14.092234
    ses                           1.092302   2.306492
    cvar(ses, id)                 0.252064   4.563507
    SectorPublic                 -2.075914   0.207225
    ses:SectorPublic              0.255840   2.090344
    cvar(ses, id):SectorPublic   -1.556703   4.386494
```

```
wald( fitc, 'cvar')
```

```
       numDF denDF  F.value p.value
  cvar     2    36 9.749657 0.00041

  Coefficients                  Estimate Std.Error DF  t-value p-value
    cvar(ses, id)               2.407785  1.062930 36 2.265235 0.02961
    cvar(ses, id):SectorPublic 1.414896  1.465217 36 0.965656 0.34066
```

```
Coefficients                  Lower 0.95 Upper 0.95
  cvar(ses, id)                 0.252064   4.563507
  cvar(ses, id):SectorPublic   -1.556703   4.386494
```

```
Anova(fitc)
```

```
Analysis of Deviance Table (Type II tests)

Response: mathach
                       Chisq Df Pr(>Chisq)
ses                  90.9731  1  < 2.2e-16 ***
cvar(ses, id)        18.5668  1  1.641e-05 ***
Sector                2.4670  1    0.11626
ses:Sector            6.2911  1    0.01213 *
cvar(ses, id):Sector  0.9325  1    0.33422
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that although neither 'cvar' component in the model seems overwhelmingly significant, the joint test that both are 0 is much more significant. This is a common kind of occurence and illustrates how individual component p-values should not be used individually to judge the overall contribution of a variable.

Interpretation of the coefficients of in this model:

```
> wald( fitc )
numDF denDF  F.value p.value
    6    76 555.2492 <.00001
    -- This is a test that ALL parameters (including the intercept)
       are equal to 0. It is generally of no interest. The
       command 'wald(fitc, -1)' tests all parameters except the
       intercept and provides a test that model as a whole is
       not predictive.

numDF denDF  F.value p.value
    6    36 407.3664 <.00001

Coefficients                Estimate Std.Error   DF   t-value p-value
(Intercept)                13.356053  0.375374 1935 35.580620 <.00001

            -- E(Y) when ses = 0 for a Catholic school
               mean ses = 0

ses                         1.699397  0.309554 1935  5.489821 <.00001
```

```
            -- Expected difference in E(Y) associated with
               a difference of 1 unit of individual ses
               within a given Catholic school. Often called the
               'WITHIN-SCHOOL' effect of ses.

cvar(ses, id)              2.407785  1.062930   36  2.265235 0.02961

            -- Expected different in E(Y) associated with
               a difference in a Catholic school mean ses of 1 unit
               with individual ses constant. Often call the
               'CONTEXTUAL' effect of ses.

               The sum of the 'WITHIN-SCHOOL' and the 'CONTEXTUAL'
               effects is the 'COMPOSITIONAL' or 'BETWEEN-SCHOOL' effect

SectorPublic               -0.934345  0.562878   36 -1.659942 0.10561

            -- Public intercept minus Catholic intercept

ses:SectorPublic            1.173092  0.467702 1935  2.508206 0.01222
```

```
            -- coef of 'ses' in Public schools minus value in
               Catholic schools

cvar(ses, id):SectorPublic  1.414896  1.465217   36  0.965656 0.34066

            -- ditto for coefficient of 'cvar(ses, id)'
```

Estimating the compositional effect (between effect) —-

```
wald( fitc )
```

```
    numDF denDF  F.value p.value
       6    36 407.3664 <.00001

   Coefficients                 Estimate Std.Error   DF   t-value p-value
     (Intercept)               13.356053  0.375374 1935 35.580620 <.00001
     ses                        1.699397  0.309554 1935  5.489821 <.00001
     cvar(ses, id)              2.407785  1.062930   36  2.265235 0.02961
     SectorPublic              -0.934345  0.562878   36 -1.659942 0.10561
     ses:SectorPublic           1.173092  0.467702 1935  2.508206 0.01222
     cvar(ses, id):SectorPublic 1.414896  1.465217   36  0.965656 0.34066
```

```
Coefficients                 Lower 0.95 Upper 0.95
  (Intercept)                 12.619872  14.092234
  ses                          1.092302   2.306492
  cvar(ses, id)                0.252064   4.563507
  SectorPublic                -2.075914   0.207225
  ses:SectorPublic             0.255840   2.090344
  cvar(ses, id):SectorPublic  -1.556703   4.386494
```

All 'possible' effects of ses:

They're actually 'infinite' consisting of ALL linear combinations of the row space of the following matrix which has dimension 4.

SO: infinite in number but finite in dimension

```
L <- rbind(
  'Within-school effect of ses | Cath'  = c(0, 1, 0,   0, 0, 0),
  'Contextual effect of ses | Cath'     = c(0, 0, 1,   0, 0, 0),
  'Between-school effect of ses | Cath' = c(0, 1, 1,   0, 0, 0),
  'Within-school effect of ses | Pub'   = c(0, 1, 0,   0, 1, 0),
  'Contextual effect of ses | Pub'      = c(0, 0, 1,   0, 0, 1),
  'Between-school effect of ses | Pub'  = c(0, 1, 1,   0, 1, 1),
```

```
  'Within-school effect of ses | Pub - Cath'   = c(0, 0, 0,   0, 1, 0),
  'Contextual effect of ses | Pub - Cath'      = c(0, 0, 0,   0, 0, 1),
  'Between-school effect of ses | Pub - Cath'  = c(0, 0, 0,   0, 1, 1))
wald( fitc, L)
```

```
    numDF denDF  F.value p.value
 1      4    36 39.23784 <.00001
```

```
                                                Estimate Std.Error   DF
   Within-school effect of ses | Cath           1.699397  0.309554 1935
   Contextual effect of ses | Cath              2.407785  1.062930   36
   Between-school effect of ses | Cath          4.107183  1.030638   36
   Within-school effect of ses | Pub            2.872489  0.350601 1935
   Contextual effect of ses | Pub               3.822681  1.008485   36
   Between-school effect of ses | Pub           6.695170  0.956243   36
   Within-school effect of ses | Pub - Cath     1.173092  0.467702 1935
   Contextual effect of ses | Pub - Cath        1.414896  1.465217   36
   Between-school effect of ses | Pub - Cath    2.587987  1.405921   36

                                                 t-value p-value Lower 0.95
   Within-school effect of ses | Cath           5.489821 <.00001   1.092302
   Contextual effect of ses | Cath              2.265235 0.02961   0.252064
```

```
Between-school effect of ses | Cath       3.985090 0.00031   2.016953
Within-school effect of ses | Pub         8.193048 <.00001   2.184894
Contextual effect of ses | Pub            3.790518 0.00055   1.777378
Between-school effect of ses | Pub        7.001537 <.00001   4.755820
Within-school effect of ses | Pub - Cath  2.508206 0.01222   0.255840
Contextual effect of ses | Pub - Cath     0.965656 0.34066  -1.556703
Between-school effect of ses | Pub - Cath 1.840777 0.07391  -0.263353

                                          Upper 0.95
Within-school effect of ses | Cath          2.306492
Contextual effect of ses | Cath             4.563507
Between-school effect of ses | Cath         6.197413
Within-school effect of ses | Pub           3.560084
Contextual effect of ses | Pub              5.867984
Between-school effect of ses | Pub          8.634521
Within-school effect of ses | Pub - Cath    2.090344
Contextual effect of ses | Pub - Cath       4.386494
Between-school effect of ses | Pub - Cath   5.439328
```

We can ask 'subquestions' with multiple dfs

Question: Is there evidence of an effect of ses in Catholic schools?

```
wald(fitc, list('ses in Catholic schools' = L[1:3,]))
```

```
                          numDF denDF  F.value p.value
   ses in Catholic schools     2    36 22.08524 <.00001

                                      Estimate Std.Error   DF  t-value
     Within-school effect of ses | Cath 1.699397  0.309554 1935 5.489821
     Contextual effect of ses | Cath    2.407785  1.062930   36 2.265235
     Between-school effect of ses | Cath 4.107183  1.030638   36 3.985090

                                      p-value Lower 0.95 Upper 0.95
     Within-school effect of ses | Cath  <.00001   1.092302   2.306492
     Contextual effect of ses | Cath     0.02961   0.252064   4.563507
     Between-school effect of ses | Cath 0.00031   2.016953   6.197413
```

Question: Is there evidence of an effect of ses in Public schools?

```
wald(fitc, list('ses in Public schools' = L[4:6,]))
```

```
                         numDF denDF  F.value p.value
   ses in Public schools     2    36 56.39045 <.00001

                                     Estimate Std.Error   DF  t-value
```

```
                                                  Within-school effect of ses | Pub  2.872489  0.350601 1935 8.193048
                                                  Contextual effect of ses | Pub     3.822681  1.008485   36 3.790518
                                                  Between-school effect of ses | Pub 6.695170  0.956243   36 7.001537

                                                                                     p-value Lower 0.95 Upper 0.95
                                                  Within-school effect of ses | Pub  <.00001   2.184894   3.560084
                                                  Contextual effect of ses | Pub     0.00055   1.777378   5.867984
                                                  Between-school effect of ses | Pub <.00001   4.755820   8.634521
```

Question: Is there evidence of a difference between Public and Catholic?

```
wald(fitc, list('Difference between effect of ses between Sectors' = L[7:9,]))
```

```
                                                                 numDF denDF  F.value
                                                Difference between effect of ses between Sectors     2    36 4.675875
                                                                 p.value
                                                Difference between effect of ses between Sectors 0.01566

                                                                                    Estimate Std.Error   DF
                                                  Within-school effect of ses | Pub - Cath  1.173092  0.467702 1935
                                                  Contextual effect of ses | Pub - Cath     1.414896  1.465217   36
                                                  Between-school effect of ses | Pub - Cath 2.587987  1.405921   36
```

```
                                                t-value p-value Lower 0.95
    Within-school effect of ses | Pub - Cath  2.508206 0.01222   0.255840
    Contextual effect of ses | Pub - Cath     0.965656 0.34066  -1.556703
    Between-school effect of ses | Pub - Cath 1.840777 0.07391  -0.263353

                                               Upper 0.95
    Within-school effect of ses | Pub - Cath    2.090344
    Contextual effect of ses | Pub - Cath       4.386494
    Between-school effect of ses | Pub - Cath   5.439328
```

or

```
wald(fitc, ':')
```

```
    numDF denDF  F.value p.value
:       2    36 4.675875 0.01566

  Coefficients                  Estimate Std.Error   DF  t-value p-value
    ses:SectorPublic            1.173092  0.467702 1935 2.508206 0.01222
    cvar(ses, id):SectorPublic 1.414896  1.465217   36 0.965656 0.34066

  Coefficients                  Lower 0.95 Upper 0.95
    ses:SectorPublic              0.255840   2.090344
```

```
cvar(ses, id):SectorPublic  -1.556703   4.386494
```

Note: that the estimated different in contextual effect is larger than the difference in within effect, yet the p-values are 'reversed'.

This illustrates the fact that there is generally more information – i.e. smaller SEs – for within effects than for contextual effects – although not always.

Although the interaction in not significant, there is much less power to detect between-school effects than within-school effects and depending on the purposes of the analysis it might not be reasonable to assume that the slopes are equal just because the p-value does not provide evidence that they are not.

QUESTIONS:

1. How would you test whether there is ANY difference between the two sectors – not just in the effect of SES.
2. Why does the second to last F test have 2 numDFs although there are three lines in the L matrix?

Visualizing the fitted model —-

Effect plots with the effects package

The `effects` package, provides an excellent way of visualizing the effects of terms in a model. If a model has

- categorical variables with more than 2 levels or
- interactions

the table of estimated coefficients and their standard errors give a very incomplete picture of the model.

Effect plots as implemented in the 'effects' package let you see the effect of categorical variables at all levels and the conditional effect of interacting variables at various levels of the other variables with which they interact. They convey a great deal more than the numerical summary of estimates and standard deviations and they should be used routinely.

There is one caveat in using effect plots with mixed models in that they may, when looking at effects of level-1 variables, give the impression that an effect is less important than it actually is. The effect plot shows error bands around the predicted value of the response. With mixed models, the there could be a large standard error around predicted values but a relatively small standard error around the difference of two predicted values.

Thus, testing specific hypotheses with linear hypothesis matrices or constructing graphical estimates of quantities other than predicted values can play a role in the analysis and reporting of results.

```
library(effects)
```

We need to refit with a computed variable for cvar(ses, id)

```
dd$ses.cvar <- with(dd, cvar(ses, id))
fitc2 <- update(fitc, . ~ (ses + ses.cvar) * Sector)
```

Range of predicted values given levels of interacting variables

```
predictorEffects(fitc2, 'ses')
```

```
 ses predictor effect

 ses*Sector effect
      Sector
ses     Catholic    Public
  -2    9.892629  6.574122
  -1   11.592026  9.446611
  -0.4 12.611665 11.170105
```

```
     0.6  14.311062 14.042594
     2    16.690219 18.064079
```

```
predictorEffects(fitc2, 'ses.cvar')
```

```
    ses.cvar predictor effect

    ses.cvar*Sector effect
           Sector
   ses.cvar Catholic    Public
      -1   10.90265  8.521924
      -0.7 11.62499  9.668729
      -0.3 12.58810 11.197801
      0.2  13.79200 13.109142
      0.6  14.75511 14.638214
```

```
predictorEffects(fitc2, 'Sector')
```

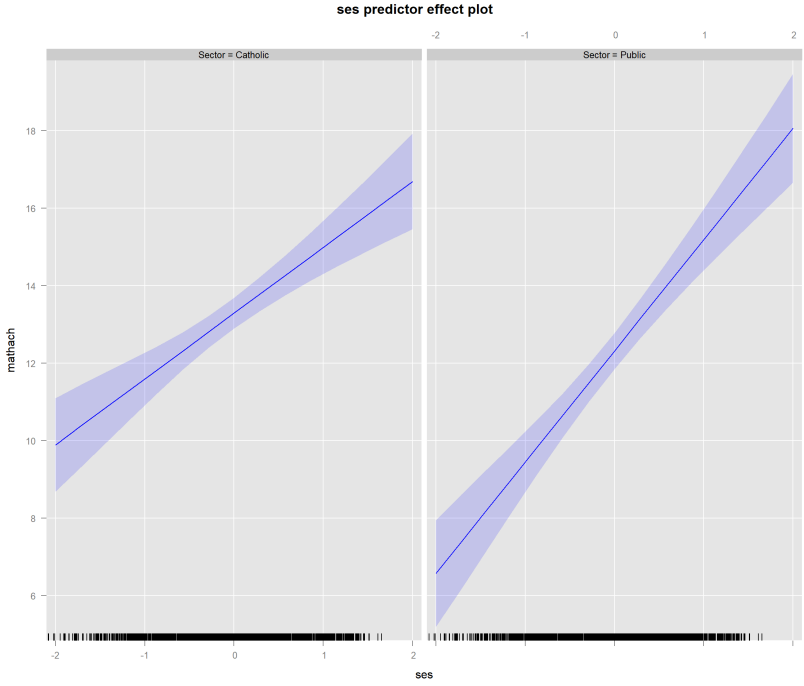
```
    Sector predictor effect

    Sector*ses*ses.cvar effect
```

```
, , ses.cvar = -1

          ses
Sector           -2       -1      -0.4      0.6        2
  Catholic 7.549473 9.248870 10.268509 11.96791 14.34706
  Public   2.854048 5.726538  7.450031 10.32252 14.34401

, , ses.cvar = -0.7

          ses
Sector           -2       -1      -0.4      0.6        2
  Catholic 8.271808 9.971206 10.990844 12.69024 15.06940
  Public   4.000853 6.873342  8.596835 11.46932 15.49081

, , ses.cvar = -0.3

          ses
Sector           -2       -1      -0.4      0.6        2
  Catholic 9.234922 10.934320 11.95396 13.65336 16.03251
  Public   5.529925  8.402414 10.12591 12.99840 17.01988
```

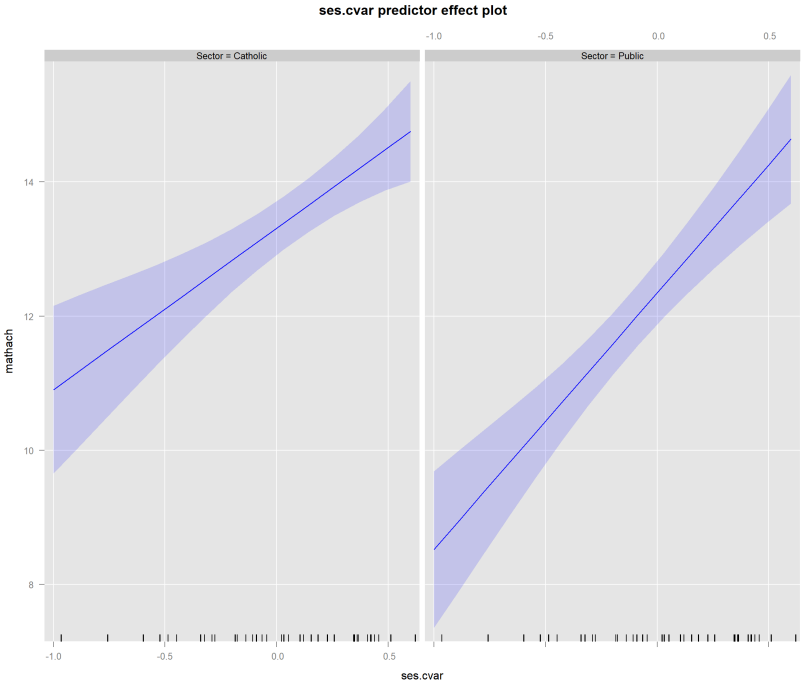
```
, , ses.cvar = 0.2

          ses
Sector           -2       -1     -0.4      0.6        2
  Catholic 10.438815 12.13821 13.15785 14.85725 17.23640
  Public    7.441266 10.31375 12.03725 14.90974 18.93122

, , ses.cvar = 0.6

          ses
Sector           -2       -1     -0.4      0.6        2
  Catholic 11.401929 13.10133 14.12097 15.82036 18.19952
  Public    8.970338 11.84283 13.56632 16.43881 20.46029
```

Plot:

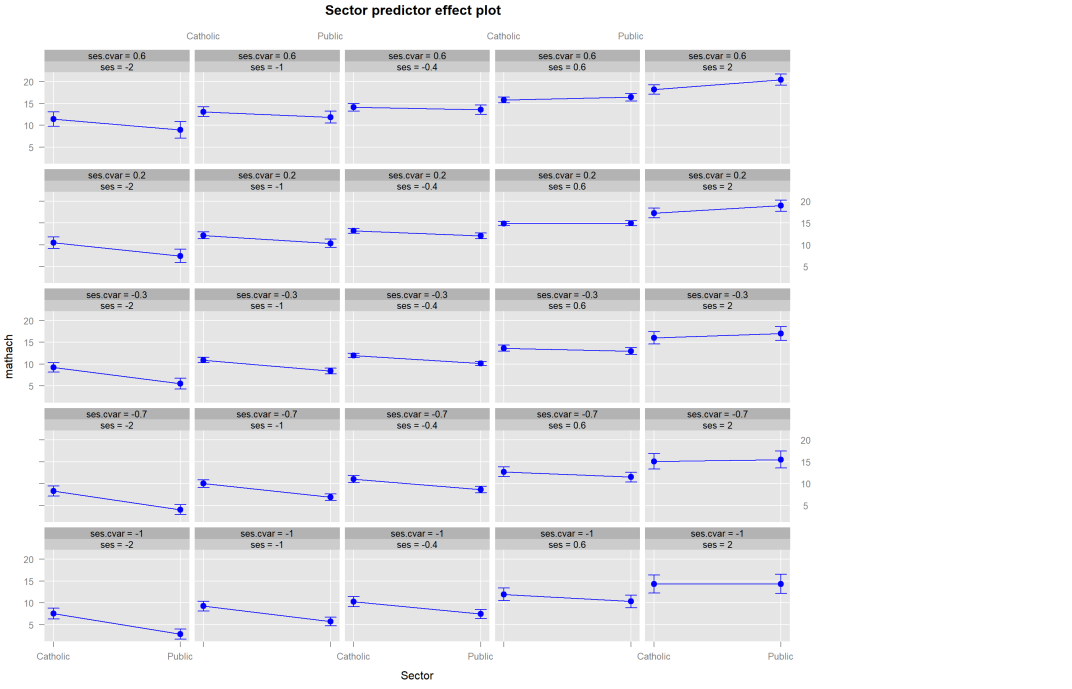
```
predictorEffects(fitc2, 'ses') %>% plot
```


ses predictor effect plot
Sector = Catholic
Sector = Public
mathach
ses
18
16
14
12
10
8
6
-2
-1
0
1
2

```
predictorEffects(fitc2, 'ses.cvar') %>% plot
```


ses.cvar predictor effect plot
Sector = Catholic
Sector = Public
mathach
ses.cvar
-1.0
-0.5
0.0
0.5
8
10
12
14

```
predictorEffects(fitc2, 'Sector') %>% plot
```


Sector predictor effect plot
Catholic
Public
Catholic
Public
ses.cvar = 0.6
ses = -2
ses.cvar = 0.6
ses = -1
ses.cvar = 0.6
ses = -0.4
ses.cvar = 0.6
ses = 0.6
ses.cvar = 0.6
ses = 2
ses.cvar = 0.2
ses = -2
ses.cvar = 0.2
ses = -1
ses.cvar = 0.2
ses = -0.4
ses.cvar = 0.2
ses = 0.6
ses.cvar = 0.2
ses = 2
ses.cvar = -0.3
ses = -2
ses.cvar = -0.3
ses = -1
ses.cvar = -0.3
ses = -0.4
ses.cvar = -0.3
ses = 0.6
ses.cvar = -0.3
ses = 2
ses.cvar = -0.7
ses = -2
ses.cvar = -0.7
ses = -1
ses.cvar = -0.7
ses = -0.4
ses.cvar = -0.7
ses = 0.6
ses.cvar = -0.7
ses = 2
ses.cvar = -1
ses = -2
ses.cvar = -1
ses = -1
ses.cvar = -1
ses = -0.4
ses.cvar = -1
ses = 0.6
ses.cvar = -1
ses = 2
mathach
Sector
Catholic
Public

Handmade prediction plot

We create a data frame with predictor values for which we want to visualize the predicted values for the model.

Since the data have two levels, we create a data set with school characteristics and deviations of students within schools

```
pred <- expand.grid(ses.mean = c(-1,0,1),
                    Sector = levels(dd$Sector), # this ensure that the factor is
                    ses.dev = seq(-1,1,.1))     # properly
pred$ses <- with(pred, ses.mean + ses.dev)
pred$m.ses <- with(pred, paste0(Sector," mean ses: ", ses.mean))
some(pred)
```

```
      ses.mean   Sector ses.dev  ses                 m.ses
2            0 Catholic    -1.0 -1.0  Catholic mean ses: 0
17           0   Public    -0.8 -0.8    Public mean ses: 0
30           1   Public    -0.6  0.4    Public mean ses: 1
31          -1 Catholic    -0.5 -1.5 Catholic mean ses: -1
32           0 Catholic    -0.5 -0.5  Catholic mean ses: 0
42           1   Public    -0.4  0.6    Public mean ses: 1
67          -1 Catholic     0.1 -0.9 Catholic mean ses: -1
```

```
78          1   Public     0.2  1.2    Public mean ses: 1
92          0 Catholic     0.5  0.5  Catholic mean ses: 0
107         0   Public     0.7  0.7    Public mean ses: 0
```

We can now use the wald function to generate predicted values and standard errors

The 'getX' function generates the design matrix over the data frame pred

Note that it is possible to use the 'predict' method but it does not provide standard errors

```
round( head( getX(fitc, pred)),2)
```

```
   Error in capply.default(x, id, mean, na.rm = na.rm, ...): object 'id' not fou
```

```
ww <- as.data.frame( wald( fitc, getX(fitc, pred)) )
```

```
   Error in capply.default(x, id, mean, na.rm = na.rm, ...): object 'id' not fou
```

```
head(ww)
```

```
   Error in head(ww): object 'ww' not found
```

```
xyplot( coef ~ ses | m.ses, ww, type = 'l')
```

```
   Error in eval(substitute(groups), data, environment(x)): object 'ww' not foun
```

```
gd(6, lwd = 2)  # select colours
xyplot( coef ~ ses | Sector, ww, groups = m.ses, type = 'l')
```

```
   Error in eval(substitute(groups), data, environment(x)): object 'ww' not foun
```

```
xyplot( coef ~ ses | Sector, ww, groups = m.ses,
        auto.key = list(columns = 2, lines = T, points = F), type = 'l')
```

```
   Error in eval(substitute(groups), data, environment(x)): object 'ww' not foun
```

```
xyplot( coef ~ ses , ww, groups = m.ses,
        auto.key = list(columns = 2, lines = T, points = F), type = 'l')
```

```
   Error in eval(substitute(groups), data, environment(x)): object 'ww' not foun
```

```
gd(col=rep(c('red','blue'),each = 3), lty = rep(1:3,2), lwd = 2)
xyplot( coef ~ ses , ww, groups = m.ses,
        ylab = 'predicted math achievement',
        auto.key = list(columns = 2, lines = T, points = F), type = 'l') +
  layer(panel.abline(v=c(-1,0,1), lty = 2))
```

```
   Error in eval(substitute(groups), data, environment(x)): object 'ww' not foun
```

EXERCISE:

- Show how the estimated coefficients are related to the last plot.

Plotting error bars —-

The data frame created by wald includes the information necessary to plot error bounds on the predicted lines

The 'panel function' panel.fit will produce the error bars. The easiest way to use it is with 'layers' provided by the 'latticeExtra' package

In the call to xyplot, provide arguments for the following parameters:

- fit: the fitted value for which a line or curve will be plotted
- upper: the upper boundary of the error bar
- lower: the lower boundary
- subscripts = T (so xyplot will pass the necessary information to the panel function)

Then you need to add a '+' at the end of the 'xyplot' command and follow that with:

layer(panel.fit(...)) if you did not use groups, and

glayer(panel.fit(...)) if you did use groups

```
head(ww)
```

```
Error in head(ww): object 'ww' not found
```

```
xyplot( coef ~ ses | Sector,
        ww, groups = m.ses,
        ylab = 'predicted math achievement',
        fit = ww$coef,
        upper = ww$U2, lower = ww$L2,
        subscripts = T,
        sub = 'predicted mean with appx. 95% error bars (NOT prediction bars)',
        auto.key = list(columns = 2, lines = T, points = F), type = 'l') +
  glayer(panel.fit(...))
```

```
Error in xyplot.formula(coef ~ ses | Sector, ww, groups = m.ses, ylab = "pred
```

```
xyplot( coef ~ ses | ses.mean,
        ww, groups = m.ses,
        ylab = 'predicted math achievement',
        fit = ww$coef,
        upper = ww$U2, lower = ww$L2,
        subscripts = T,
        layout = c(1,3),
        sub = 'predicted mean with appx. 95% error bars (NOT prediction bars)',
        auto.key = list(columns = 2, lines = T, points = F), type = 'l') +
  glayer(panel.fit(...))
```

```
   Error in xyplot.formula(coef ~ ses | ses.mean, ww, groups = m.ses, ylab = "pr
```

```
xyplot( coef ~ ses | paste0('Mean school ses = ',ses.mean),
        ww, groups = m.ses,
        ylab = 'predicted math achievement',
        fit = ww$coef,
        upper = ww$U2, lower = ww$L2,
        subscripts = T,
        layout = c(1,3),
        sub = 'predicted mean with appx. 95% error bars (NOT prediction bars)',
        auto.key = list(columns = 2, lines = T, points = F), type = 'l') +
  glayer(panel.fit(...))
```

```
   Error in xyplot.formula(coef ~ ses | paste0("Mean school ses = ", ses.mean),
```

Estimating the gap

We would like to estimate the intersector gap at the same value of ses.mean and ses.

```
head(ww)
```

```
   Error in head(ww): object 'ww' not found
```

```
vals <- ww[, c('ses','m.ses','ses.mean','Sector')]
```

```
   Error in eval(expr, envir, enclos): object 'ww' not found
```

Construct an L matrix to estimate the gap

This is like estimating the 'derivative' of each term wrt Sector

```
wald(fitc)
```

```
    numDF denDF  F.value p.value
        6    36 407.3664 <.00001

   Coefficients                   Estimate Std.Error   DF   t-value p-value
     (Intercept)                 13.356053  0.375374 1935 35.580620 <.00001
     ses                          1.699397  0.309554 1935  5.489821 <.00001
     cvar(ses, id)                2.407785  1.062930   36  2.265235 0.02961
     SectorPublic                -0.934345  0.562878   36 -1.659942 0.10561
     ses:SectorPublic             1.173092  0.467702 1935  2.508206 0.01222
     cvar(ses, id):SectorPublic   1.414896  1.465217   36  0.965656 0.34066

   Coefficients                  Lower 0.95 Upper 0.95
     (Intercept)                  12.619872  14.092234
     ses                           1.092302   2.306492
```

```
     cvar(ses, id)                0.252064   4.563507
     SectorPublic                -2.075914   0.207225
     ses:SectorPublic             0.255840   2.090344
     cvar(ses, id):SectorPublic  -1.556703   4.386494
```

```
Lgap <- with(vals,
             cbind(0, 0, 0, 1, ses, ses.mean)
)
```

```
   Error in with(vals, cbind(0, 0, 0, 1, ses, ses.mean)): object 'vals' not foun
```

```
some(Lgap)
```

```
   Error in some(Lgap): object 'Lgap' not found
```

```
ww <- as.data.frame(wald(fitc, Lgap, data = vals ))
```

```
   Error in wald(fitc, Lgap, data = vals): object 'Lgap' not found
```

```
some(ww)
```

```
   Error in some(ww): object 'ww' not found
```

```
gd(col = 'blue')
xyplot( coef ~ ses | paste0('Mean school ses = ',ses.mean),
       ww,
```

```
       ylim = c(-8,8), xlim = c(-2,2),
       ylab = 'predicted math achievement',
       fit = ww$coef,
       upper = ww$U2, lower = ww$L2,
       subscripts = T,
       layout = c(1,3),
       type = 'l',
       sub = 'estimated difference (Public - Catholic) with appx. 95% error bars
  layer(panel.abline(h = 0))+
  layer(panel.fit(...))
```

```
   Error in xyplot.formula(coef ~ ses | paste0("Mean school ses = ", ses.mean),
```

```
xyplot( coef ~ ses | paste0('Mean school ses = ',ses.mean),
       ww,
       ylim = c(-8,8), xlim = c(-2,2),
       ylab = 'predicted math achievement',
       fit = ww$coef,
       upper = ww$U2, lower = ww$L2,
       subscripts = T,
       layout = c(3,1),
       type = 'l',
```

```
        sub = 'estimated difference (Public - Catholic) with appx. 95% error bar
  layer(panel.abline(h = 0, lwd = 2 ))+
  layer(panel.fit(...))
```

```
   Error in xyplot.formula(coef ~ ses | paste0("Mean school ses = ", ses.mean),
```

We can combine the bands in one panel, although it isn't very effective here.

Note the changes:

- the '| panel' argument becomes a 'groups' argument
- remove the 'layout' argument
- add 'auto.key'
- 'layer' become 'glayer'
- everything else can stay the same

```
gd(3)
xyplot( coef ~ ses ,
        ww,
        groups = paste0('Mean school ses = ',ses.mean),
        ylim = c(-8,8), xlim = c(-2,2),
        ylab = 'predicted math achievement',
        fit = ww$coef,
        upper = ww$U2, lower = ww$L2,
```

```
        subscripts = T,
        type = 'l',
        auto.key = T,
        sub = 'estimated difference (Public - Catholic) with appx. 95% error bar
  glayer(panel.abline(h = 0, lwd = 2 ))+
  glayer(panel.fit(...))
```

```
 Error in xyplot.formula(coef ~ ses, ww, groups = paste0("Mean school ses = ",
```

Figure: Combining bands in one panel. Tastes may vary!

EXERCISES —-

1. Starting with the model above, test whether Sex improves prediction in the model. Consider including a contextual variable for Sex. What is its interpretation?
2. Is is reasonable to use the contextual variable for Sex as the only regressor for the contextual effect of Sex or should other variables be used as well? Can you create these variables and include them in your analysis? What do they reveal?

See Visualizing the coeficients

Using CWG instead of raw SES —-

```
fitcd <- update( fitc, . ~ (cvar(ses,id) + dvar(ses,id)*Sector))
summary( fitcd )
```

```
   Linear mixed-effects model fit by REML
    Data: dd
         AIC      BIC  logLik
     12843.4 12893.68 -6412.7

   Random effects:
    Formula: ~1 + ses | id
    Structure: General positive-definite, Log-Cholesky parametrization
                StdDev    Corr
   (Intercept) 1.5088268 (Intr)
   ses         0.4215203 -0.202
   Residual    6.1173112

   Fixed effects: mathach ~ cvar(ses, id) + dvar(ses, id) + Sector + dvar(ses,
                                  Value Std.Error   DF  t-value p-value
   (Intercept)                13.257428 0.3803559 1935 34.85533  0.0000
   cvar(ses, id)               5.477723 0.7136326   37  7.67583  0.0000
```

```
   dvar(ses, id)                1.711278 0.3023289 1935  5.66032  0.0000
   SectorPublic                -0.981203 0.5765720   37 -1.70179  0.0972
   dvar(ses, id):SectorPublic   1.161565 0.4571695 1935  2.54078  0.0111
    Correlation:
                               (Intr) cv(,i) dv(,i) SctrPb
   cvar(ses, id)               -0.140
   dvar(ses, id)               -0.052  0.020
   SectorPublic                -0.682  0.251  0.038
   dvar(ses, id):SectorPublic   0.033 -0.003 -0.661 -0.052

   Standardized Within-Group Residuals:
           Min           Q1          Med           Q3          Max
   -3.12476554 -0.74613310  0.02214392  0.78204903  2.76676038

   Number of Observations: 1977
   Number of Groups: 40
```

```
summary( fitc )
```

```
   Linear mixed-effects model fit by REML
    Data: dd
         AIC      BIC    logLik
```

```
  12839.6 12895.47 -6409.801

Random effects:
 Formula: ~1 + ses | id
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.4561519 (Intr)
ses         0.5190639 -0.217
Residual    6.1140010

Fixed effects: mathach ~ (ses + cvar(ses, id)) * Sector
                               Value Std.Error   DF  t-value p-value
(Intercept)                13.356053 0.3753744 1935 35.58062  0.0000
ses                         1.699397 0.3095543 1935  5.48982  0.0000
cvar(ses, id)               2.407785 1.0629298   36  2.26523  0.0296
SectorPublic               -0.934345 0.5628780   36 -1.65994  0.1056
ses:SectorPublic            1.173092 0.4677016 1935  2.50821  0.0122
cvar(ses, id):SectorPublic  1.414896 1.4652174   36  0.96566  0.3407
 Correlation:
                           (Intr) ses    cv(,i) SctrPb ss:ScP
ses                        -0.071
```

```
   cvar(ses, id)               -0.186 -0.248
   SectorPublic                -0.667  0.047  0.124
   ses:SectorPublic             0.047 -0.662  0.164 -0.065
   cvar(ses, id):SectorPublic   0.135  0.180 -0.725  0.056 -0.284

   Standardized Within-Group Residuals:
           Min           Q1          Med           Q3          Max
   -3.09882513 -0.75336271  0.03194452  0.78173508  2.79671295

   Number of Observations: 1977
   Number of Groups: 40
```

```
wald(fitc, L)
```

```
     numDF denDF  F.value p.value
   1     4    36 39.23784 <.00001

                                          Estimate Std.Error   DF
     Within-school effect of ses | Cath   1.699397  0.309554 1935
     Contextual effect of ses | Cath      2.407785  1.062930   36
     Between-school effect of ses | Cath  4.107183  1.030638   36
     Within-school effect of ses | Pub    2.872489  0.350601 1935
```

```
Contextual effect of ses | Pub            3.822681  1.008485   36
Between-school effect of ses | Pub        6.695170  0.956243   36
Within-school effect of ses | Pub - Cath  1.173092  0.467702 1935
Contextual effect of ses | Pub - Cath     1.414896  1.465217   36
Between-school effect of ses | Pub - Cath 2.587987  1.405921   36
```

```
                                    t-value p-value Lower 0.95
Within-school effect of ses | Cath 5.489821 <.00001   1.092302
Contextual effect of ses | Cath     2.265235 0.02961   0.252064
Between-school effect of ses | Cath 3.985090 0.00031   2.016953
Within-school effect of ses | Pub  8.193048 <.00001   2.184894
Contextual effect of ses | Pub      3.790518 0.00055   1.777378
Between-school effect of ses | Pub  7.001537 <.00001   4.755820
Within-school effect of ses | Pub - Cath  2.508206 0.01222   0.255840
Contextual effect of ses | Pub - Cath     0.965656 0.34066  -1.556703
Between-school effect of ses | Pub - Cath 1.840777 0.07391  -0.263353
```

```
                                    Upper 0.95
Within-school effect of ses | Cath   2.306492
Contextual effect of ses | Cath      4.563507
Between-school effect of ses | Cath  6.197413
```

```
Within-school effect of ses | Pub           3.560084
Contextual effect of ses | Pub              5.867984
Between-school effect of ses | Pub          8.634521
Within-school effect of ses | Pub - Cath    2.090344
Contextual effect of ses | Pub - Cath       4.386494
Between-school effect of ses | Pub - Cath   5.439328
```

QUESTIONS:

1. what's the same and what's different?
2. How are the various effects of ses related
3. Create an L matrix to estimate the 3 effects of ses with fitcd as in the previous example

Note: replacing ses with dvar(ses,id) in the RE model does give an **mathematically** equivalent model but **NOT** a **numerically** equivalent model.

That is, with calculations using infinite precision the the two models would give equivalent results but *with finite precision* the 'dvar(ses,id)' model will, in general, converge more easily.

CWG vs raw in RE model —-

Alternative but non-equivalent RE parametrization

Using ‘dvar(ses,id)’ or raw ‘ses’ in the FE model produces an equivalent model.

This is not so in the RE model. The

‘random = ~ 1 + ses | id’ (1)

model is essentially different from the

‘random = ~ 1 + dvar(ses,id) | id’ (2)

model.

In model (1) the ‘point of minimum variance’ of school regression lines corresponds to a *common* value of ses for all schools.

In model (2) the ‘point of minimum variance’ of school regression lines has a consistent value *relative* to each school’s mean ses.

```
fitca <- update( fitc, random = ~ 1 + dvar( ses, id ) | id)
summary( fitca )
```

```
   Linear mixed-effects model fit by REML
    Data: dd
          AIC      BIC    logLik
     12839.22 12895.09 -6409.612
```

```
Random effects:
 Formula: ~1 + dvar(ses, id) | id
 Structure: General positive-definite, Log-Cholesky parametrization
             StdDev    Corr
(Intercept)   1.4625200 (Intr)
dvar(ses, id) 0.6802261 -0.071
Residual      6.1084863

Fixed effects: mathach ~ (ses + cvar(ses, id)) * Sector
                              Value Std.Error   DF  t-value p-value
(Intercept)                13.374349 0.3732778 1935 35.82948  0.0000
ses                         1.695976 0.3241495 1935  5.23208  0.0000
cvar(ses, id)               2.398991 1.0734653   36  2.23481  0.0317
SectorPublic               -0.939334 0.5625274   36 -1.66985  0.1036
ses:SectorPublic            1.171880 0.4888990 1935  2.39698  0.0166
cvar(ses, id):SectorPublic  1.238194 1.4581049   36  0.84918  0.4014
 Correlation:
                           (Intr) ses    cv(,i) SctrPb ss:ScP
ses                        -0.028
cvar(ses, id)              -0.159 -0.300
SectorPublic               -0.664  0.019  0.106
```

```
   ses:SectorPublic             0.019 -0.663  0.199 -0.026
   cvar(ses, id):SectorPublic  0.117  0.221 -0.736  0.078 -0.336

   Standardized Within-Group Residuals:
           Min           Q1          Med           Q3          Max
   -3.10199875 -0.75105077  0.03298316  0.78240512  2.80416484

   Number of Observations: 1977
   Number of Groups: 40
```

```
summary( fitc )
```

```
   Linear mixed-effects model fit by REML
    Data: dd
         AIC      BIC    logLik
     12839.6 12895.47 -6409.801

   Random effects:
    Formula: ~1 + ses | id
    Structure: General positive-definite, Log-Cholesky parametrization
                StdDev    Corr
   (Intercept) 1.4561519 (Intr)
```

```
ses         0.5190639 -0.217
Residual    6.1140010

Fixed effects: mathach ~ (ses + cvar(ses, id)) * Sector
                               Value Std.Error   DF  t-value p-value
(Intercept)                13.356053 0.3753744 1935 35.58062  0.0000
ses                         1.699397 0.3095543 1935  5.48982  0.0000
cvar(ses, id)               2.407785 1.0629298   36  2.26523  0.0296
SectorPublic               -0.934345 0.5628780   36 -1.65994  0.1056
ses:SectorPublic            1.173092 0.4677016 1935  2.50821  0.0122
cvar(ses, id):SectorPublic  1.414896 1.4652174   36  0.96566  0.3407
 Correlation:
                           (Intr) ses    cv(,i) SctrPb ss:ScP
ses                        -0.071
cvar(ses, id)              -0.186 -0.248
SectorPublic               -0.667  0.047  0.124
ses:SectorPublic            0.047 -0.662  0.164 -0.065
cvar(ses, id):SectorPublic  0.135  0.180 -0.725  0.056 -0.284

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
```

```
-3.09882513 -0.75336271  0.03194452  0.78173508  2.79671295

Number of Observations: 1977
Number of Groups: 40
```

Which is better? It will depend on the true state of nature.

What do the data tell us?

Neither model is nested in the other – in fact they have the same DFs – so we can't get a p-value for a test. We can compare with AIC:

```
anova(fitca, fitc)
```

```
       Model df      AIC      BIC    logLik
fitca      1 10 12839.22 12895.09 -6409.612
fitc       2 10 12839.60 12895.47 -6409.801
```

fitca looks better, barely!

More importantly, it seems to make more sense.

EXERCISES —-

1. Now that we have another Level 2 variable (cvar(ses, id)), we could consider whether there are interesting interactions between it and other variables.

Fit a model with interactions between cvar(ses,id) and other variables.

2. Is there evidence that these interactions are non-zero in the population?
    - BEWARE: Do not merely look at the individual p-value Either remove effects one at a time or test effects simultaneously with 'wald' or anova (need to refit with method = "ML")
    - What do you conclude wrt interactions?

REML vs ML —-

Previous fits with REML (default)

Here we use a 'sub-par' model (no contextual effect) to illustrate convergence issues.

FE: ~ 1 + ses + Sector + ses:Sector

RE: ~ 1 + ses | id

Fitting with 'REML'

Maximum likelihood optimization is done only on the RE model, i.e. the G matrix and sigma – or G and R. The FE model is then estimated using GLS. This makes optimization easier because there are fewer parameters – in this case 4:

1. 2 SD parameters for REs: 'beta_1' and 'beta_ses' (diagonal of G)
2. covariance parameter for REs: between 'beta_1', and 'beta_ses' (off-diagonal of G)
3. the SD parameter for individual error

Fit with 'ML'

Maximum likelihood optimization done on all paramters together. Generally more numerically difficult.

Pros and Cons:

REML: better estimates of variances, good for Wald tests. But, can use Likelihood-Ratio Tests (LRT with anova) only to compare models with identical FEs and nested RE models. However, beware that LRTs for RE models are problematic near the 'edge' of the model, e.g.

- if the null model is on the 'edge' of the full model
- if an estimated G matrix is nearly singular

ML: Not so good estimates of variances, not so good for Wald tests. But LRTs can compare models that are nested wrt to FEs and or REs.

REML fit (default):

```
fit <- lme( mathach ~ ses * Sector, dd,
            random = ~ 1 + ses | id,
            na.action = na.exclude,
            control = list(msMaxIter=200, msVerbose=T))
```

```
     0:     11105.191:  1.22722  2.27855 -3.36583
     1:     11105.172:  1.23706  3.08949 -3.39049
     2:     11105.171:  1.19748  3.08072 -3.39188
     3:     11105.161:  1.21737  3.07678 -3.39235
     4:     11105.096:  1.22151  2.70896 -3.41493
     5:     11105.089:  1.22011  2.64468 -3.43555
     6:     11105.085:  1.22389  2.55034 -3.50479
     7:     11105.083:  1.22385  2.57106 -3.54636
     8:     11105.076:  1.22534  2.60016 -3.72983
     9:     11105.057:  1.23381  2.66214 -4.47027
    10:     11105.033:  1.25234  2.72319 -5.87758
    11:     11105.019:  1.27696  2.75593 -7.28573
    12:     11105.013:  1.30246  2.78019 -8.50888
```

```
    13:     11105.011:  1.31965  2.79844 -9.36213
    14:     11105.011:  1.33116  2.81675 -9.93866
    15:     11105.011:  1.33714  2.82776 -10.3398
    16:     11105.011:  1.33993  2.83733 -10.5086
    17:     11105.011:  1.33984  2.84079 -10.5414
    18:     11105.011:  1.33908  2.84283 -10.5393
    19:     11105.011:  1.33903  2.84278 -10.5371
```

ML fit:

```
#  fit.ml <- lme( mathach ~ ses * Sector, dd,
#                 random = ~ 1 + ses | id,
#                 na.action = na.exclude,
#                 method = "ML",
#                 control = list(msMaxIter=200, msVerbose=T))
```

equivalent:

```
fit.ml <- update(fit, method = 'ML')
```

```
     0:     11116.488:  1.26387  2.36435 -3.50300
     1:     11116.369:  1.27189  3.26343 -3.52690
     2:     11116.362:  1.25371  3.26152 -3.52738
     3:     11116.357:  1.26159  3.19060 -3.54350
```

```
 4:     11116.344:  1.26820  3.00294 -3.58152
 5:     11116.336:  1.27032  2.81688 -3.62716
 6:     11116.332:  1.26359  2.81941 -3.65385
 7:     11116.330:  1.25804  2.81817 -3.71167
 8:     11116.329:  1.25586  2.81904 -3.76973
 9:     11116.324:  1.25192  2.81640 -4.00208
10:     11116.314:  1.24864  2.82902 -4.61237
11:     11116.304:  1.25175  2.85724 -5.22213
12:     11116.277:  1.27266  2.93789 -7.00650
13:     11116.252:  1.30765  3.05638 -9.17520
14:     11116.240:  1.34226  3.01159 -11.3467
15:     11116.229:  1.37860  3.10898 -13.5164
16:     11116.224:  1.41455  3.14663 -15.6880
17:     11116.221:  1.45113  3.15186 -17.8599
18:     11116.220:  1.48586  3.18124 -20.0316
19:     11116.219:  1.51994  3.20406 -22.2035
20:     11116.218:  1.55064  3.22386 -24.2324
21:     11116.218:  1.57972  3.24631 -26.2613
22:     11116.217:  1.60724  3.26437 -28.2902
23:     11116.217:  1.63097  3.28564 -30.0739
24:     11116.217:  1.65191  3.30750 -31.8577
```

```
25:     11116.216:  1.67476  3.34690 -34.5078
26:     11116.215:  1.68926  3.38050 -36.5249
27:     11116.215:  1.68666  3.39839 -37.2087
28:     11116.215:  1.69364  3.40859 -37.8926
29:     11116.215:  1.69991  3.41753 -38.5765
30:     11116.215:  1.70677  3.42223 -39.1666
31:     11116.215:  1.73242  3.43322 -41.1845
32:     11116.215:  1.74434  3.43038 -41.8849
33:     11116.215:  1.76468  3.43539 -43.4545
34:     11116.214:  1.78907  3.44632 -45.5732
35:     11116.214:  1.81430  3.45958 -47.9453
36:     11116.214:  1.82514  3.46566 -48.9606
37:     11116.214:  1.86235  3.48610 -52.5741
38:     11116.214:  1.85876  3.48527 -52.3932
39:     11116.214:  1.87752  3.49341 -54.1692
40:     11116.214:  1.88572  3.49664 -54.9450
41:     11116.214:  1.90242  3.50224 -56.4968
42:     11116.214:  1.91953  3.50653 -58.0486
43:     11116.214:  1.93706  3.50907 -59.6003
44:     11116.214:  1.95337  3.51374 -61.1521
45:     11116.214:  1.96914  3.51822 -62.7039
```

```
46:     11116.214:  1.98756  3.52451 -64.6271
47:     11116.214:  2.00629  3.53088 -66.5502
48:     11116.214:  2.01839  3.53556 -67.9328
49:     11116.214:  2.03148  3.53962 -69.3154
50:     11116.214:  2.04463  3.54299 -70.6979
51:     11116.214:  2.06259  3.54637 -72.5577
52:     11116.214:  2.08052  3.55034 -74.4175
53:     11116.214:  2.08561  3.55078 -74.9460
54:     11116.214:  2.10570  3.55444 -77.0600
55:     11116.214:  2.12038  3.55726 -78.6633
56:     11116.214:  2.13299  3.56012 -80.1147
57:     11116.214:  2.14597  3.56298 -81.5662
58:     11116.214:  2.15858  3.56589 -83.0177
59:     11116.214:  2.16736  3.56817 -84.0919
60:     11116.214:  2.17658  3.57007 -85.1662
61:     11116.214:  2.19976  3.57416 -87.8635
62:     11116.214:  2.21172  3.57556 -89.2594
63:     11116.214:  2.22380  3.57736 -90.6552
64:     11116.214:  2.23579  3.57908 -92.0511
65:     11116.214:  2.23585  3.57893 -92.0595
```

Here ML takes more iterations but **used to** give up with 'singular convergence' perhaps

because the estimated RE variance is closer to singular, which, combined with the larger number of parameter, leads 'nlminb' to declare singular convergence: the peak is too flat.

Note: increasing msMaxIter won't help with 'singular convergence'

REML maximizes only over RE parameters, while ML maximizes over all parameters, hence more challenging optimization

To see the fit at the last iteration, use 'returnObject = TRUE' in the 'control' list:

```
# fit.ml <- lme( mathach ~ ses * Sector, dd, random = ~ 1 + ses | id,
#                na.action = na.exclude,
#                method = "ML",
#                control = list(msMaxIter=200,
#                               msVerbose=T,
#                               returnObject = TRUE))
summary(fit.ml)
```

```
   Linear mixed-effects model fit by maximum likelihood
    Data: dd
          AIC      BIC    logLik
     12854.79 12899.51 -6419.397

   Random effects:
```

```
 Formula: ~1 + ses | id
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.8032583 (Intr)
ses         0.1707538 0.932
Residual    6.1189789

Fixed effects: mathach ~ ses * Sector
                    Value Std.Error   DF   t-value p-value
(Intercept)      13.489366 0.4345233 1935 31.044061  0.0000
ses               1.815089 0.2821217 1935  6.433711  0.0000
SectorPublic     -1.590550 0.6400743   38 -2.484946  0.0175
ses:SectorPublic  1.398797 0.4234030 1935  3.303700  0.0010
 Correlation:
                  (Intr) ses    SctrPb
ses                0.076
SectorPublic      -0.679 -0.052
ses:SectorPublic -0.051 -0.666  0.149

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
```

```
-3.06217668 -0.74613002  0.03736602  0.78804322  2.74091602

Number of Observations: 1977
Number of Groups: 40
```

Shows high correlation in the RE model between b_1 and b_ses. This suggests that the 'point of minimum variance' is far from the observed values of ses and that the random true school regression lines seem close to parallel.

Possible remedies for non-convergence:

Alas now 'academic' here because the model now converges

- drop random slope
- recenter ses
- add a contextual effect and CWG random effect: ~ 1 + dvar(ses,id) | id
- explore different models
- change the criterion for lme to declare singularity, i.e. excessive flatness near the likelihood peak with 'sing.tol'

```
fit.ml <- lme( mathach ~ ses * Sector + Sex, dd,
               random = ~ 1 + ses + Sex | id,
               na.action = na.exclude,
               method = "ML",
```

```
                control = list(msMaxIter=200, msVerbose=T,
                               sing.tol=1e-20))
 0:     11109.615:  1.41833  2.41824  1.88935 -3.96692 -0.973608 -1.36731
 1:     11109.477:  1.44252  2.70884  2.15340 -3.97460 -0.990959 -1.37504
 2:     11109.400:  1.35156  2.71273  2.13702 -3.97739 -0.998252 -1.37783
 3:     11109.376:  1.36793  2.79269  2.14010 -3.99193 -1.03470 -1.39820
 4:     11109.362:  1.40683  2.80828  2.01850 -4.03354 -1.14572 -1.46064
 5:     11109.320:  1.42023  2.99532  2.08100 -4.13910 -1.43142 -1.62290
 6:     11109.285:  1.43114  3.10075  2.08836 -4.24835 -1.75102 -1.80401
 7:     11109.185:  1.48413  3.36028  2.12353 -4.68678 -3.05622 -2.55469
 8:     11109.123:  1.53966  3.48380  2.21745 -5.12894 -4.36430 -3.32730
 9:     11109.076:  1.59585  3.55853  2.24802 -5.56928 -5.67298 -4.11118
10:     11109.021:  1.71368  3.55344  2.39068 -6.49460 -8.40163 -5.77104
11:     11108.954:  1.81242  3.40201  2.40801 -7.25146 -10.6234 -7.15512
12:     11108.937:  1.90548  3.27424  2.47237 -8.01006 -12.8474 -8.53624
13:     11108.932:  1.90942  3.30006  2.46637 -8.09936 -13.0958 -8.69635
14:     11108.923:  1.93768  3.37975  2.48650 -8.44823 -14.1038 -9.32179
15:     11108.912:  2.04458  3.52641  2.55840 -9.63736 -17.5463 -11.4635
16:     11108.905:  2.05586  3.54962  2.56007 -9.83938 -18.1168 -11.8326
17:     11108.899:  2.11950  3.62610  2.59167 -10.6409 -20.4159 -13.2863
18:     11108.894:  2.18970  3.69114  2.62024 -11.5585 -23.0377 -14.9574
```

```
19:     11108.889:  2.24429  3.73449  2.63533 -12.3256 -25.2168 -16.3618
20:     11108.887:  2.30215  3.79270  2.65515 -13.0915 -27.4009 -17.7587
21:     11108.885:  2.34188  3.83961  2.66230 -13.6605 -29.0131 -18.8015
22:     11108.883:  2.39691  3.90956  2.66522 -14.4695 -31.2977 -20.2890
23:     11108.882:  2.45426  3.97424  2.67403 -15.2774 -33.5868 -21.7705
24:     11108.880:  2.50176  4.02582  2.67524 -15.9609 -35.5165 -23.0288
25:     11108.879:  2.55864  4.08471  2.66951 -16.7983 -37.8728 -24.5771
26:     11108.878:  2.61557  4.14578  2.67017 -17.6350 -40.2328 -26.1198
27:     11108.878:  2.64907  4.18248  2.66874 -18.1610 -41.7119 -27.0935
28:     11108.877:  2.69285  4.23159  2.66654 -18.9217 -43.8454 -28.5079
29:     11108.876:  2.73814  4.28140  2.66685 -19.6820 -45.9818 -29.9180
30:     11108.876:  2.77826  4.32515  2.66935 -20.4426 -48.1145 -31.3339
31:     11108.875:  2.81108  4.36009  2.67593 -21.2034 -50.2423 -32.7574
32:     11108.874:  2.86012  4.41024  2.68995 -22.4337 -53.6809 -35.0625
33:     11108.874:  2.87997  4.42669  2.71002 -23.3420 -56.2078 -36.7800
34:     11108.873:  2.91449  4.46188  2.72112 -24.2499 -58.7441 -38.4832
35:     11108.873:  2.93380  4.48241  2.72586 -24.7517 -60.1447 -39.4251
36:     11108.873:  2.93473  4.48449  2.72325 -24.7335 -60.0938 -39.3907
37:     11108.873:  2.97031  4.52788  2.71283 -25.3758 -61.8861 -40.5955
38:     11108.873:  3.03861  4.60421  2.70580 -26.8211 -65.9184 -43.3097
39:     11108.873:  3.03513  4.59758  2.70602 -26.7677 -65.7680 -43.2114
```

40:	11108.873:	3.03679	4.59937	2.70571	-26.8211	-65.9168	-43.3123
41:	11108.872:	3.05342	4.61500	2.70623	-27.2488	-67.1083	-44.1186
42:	11108.872:	3.11825	4.67979	2.70999	-28.9597	-71.8741	-47.3440
43:	11108.872:	3.11142	4.67338	2.71520	-28.9620	-71.8786	-47.3515
44:	11108.872:	3.11117	4.67349	2.71485	-28.9660	-71.8897	-47.3590
45:	11108.872:	3.13222	4.69632	2.71875	-29.6371	-73.7580	-48.6261
46:	11108.872:	3.15772	4.72426	2.72096	-30.3932	-75.8633	-50.0525
47:	11108.872:	3.18344	4.75266	2.72209	-31.1492	-77.9688	-51.4787
48:	11108.872:	3.20928	4.78158	2.72176	-31.9053	-80.0744	-52.9047
49:	11108.871:	3.24659	4.82278	2.71914	-32.9989	-83.1206	-54.9671
50:	11108.871:	3.28239	4.85920	2.71907	-34.0924	-86.1663	-57.0305
51:	11108.871:	3.31604	4.89126	2.71866	-35.1859	-89.2115	-59.0947
52:	11108.871:	3.34442	4.91913	2.72002	-36.1901	-92.0077	-60.9915
53:	11108.871:	3.36973	4.94546	2.72113	-37.0776	-94.4788	-62.6675
54:	11108.871:	3.39259	4.96998	2.72239	-37.9650	-96.9493	-64.3443
55:	11108.871:	3.41683	4.99579	2.72332	-38.8525	-99.4202	-66.0207
56:	11108.871:	3.44020	5.02127	2.72401	-39.7399	-101.891	-67.6972
57:	11108.871:	3.47088	5.05549	2.72465	-40.9619	-105.293	-70.0064
58:	11108.871:	3.50130	5.08730	2.72501	-42.1741	-108.668	-72.2971
59:	11108.870:	3.54083	5.12735	2.72505	-43.8354	-113.293	-75.4378
60:	11108.870:	3.58166	5.16897	2.72549	-45.4969	-117.919	-78.5780

61:	11108.870:	3.57103	5.15775	2.72541	-45.0932	-116.795	-77.8153
62:	11108.870:	3.58030	5.16713	2.72553	-45.4968	-117.919	-78.5785
63:	11108.870:	3.61731	5.20512	2.72598	-47.1114	-122.414	-81.6312
64:	11108.870:	3.63540	5.22371	2.72626	-47.9463	-124.738	-83.2102
65:	11108.870:	3.65738	5.24653	2.72665	-49.0452	-127.797	-85.2895
66:	11108.870:	3.68094	5.27097	2.72697	-50.1442	-130.856	-87.3681
67:	11108.870:	3.70419	5.29512	2.72724	-51.2432	-133.916	-89.4468
68:	11108.870:	3.70382	5.29489	2.72714	-51.2391	-133.904	-89.4392
69:	11108.870:	3.70372	5.29509	2.72680	-51.2499	-133.934	-89.4596
70:	11108.870:	3.72505	5.31734	2.72698	-52.3046	-136.870	-91.4551
71:	11108.870:	3.73523	5.32795	2.72702	-52.8152	-138.291	-92.4212
72:	11108.870:	3.75534	5.34876	2.72713	-53.8363	-141.134	-94.3533
73:	11108.870:	3.77507	5.36911	2.72724	-54.8574	-143.976	-96.2855
74:	11108.870:	3.79415	5.38864	2.72735	-55.8785	-146.819	-98.2178
75:	11108.870:	3.81913	5.41386	2.72772	-57.2752	-150.706	-100.861
76:	11108.870:	3.84466	5.43989	2.72809	-58.6719	-154.594	-103.505
77:	11108.870:	3.85358	5.44893	2.72833	-59.2238	-156.130	-104.549
78:	11108.870:	3.89183	5.48813	2.72887	-61.4314	-162.275	-108.728
79:	11108.870:	3.89233	5.48890	2.72878	-61.5124	-162.500	-108.882
80:	11108.870:	3.91404	5.51138	2.72888	-62.8092	-166.109	-111.337
81:	11108.870:	3.92430	5.52201	2.72888	-63.4305	-167.839	-112.513

82:	11108.870:	3.94457	5.54292	2.72886	-64.6732	-171.297	-114.865
83:	11108.870:	3.96442	5.56338	2.72879	-65.9159	-174.756	-117.218
84:	11108.870:	3.99363	5.59335	2.72864	-67.8061	-180.017	-120.797
85:	11108.870:	4.02246	5.62257	2.72898	-69.6964	-185.278	-124.376
86:	11108.870:	4.04654	5.64655	2.72954	-71.4205	-190.076	-127.642
87:	11108.870:	4.07169	5.67218	2.72980	-73.1446	-194.874	-130.907
88:	11108.870:	4.06953	5.67031	2.72970	-73.0381	-194.578	-130.705
89:	11108.870:	4.08436	5.68574	2.72974	-74.1011	-197.536	-132.719
90:	11108.870:	4.09548	5.69720	2.72976	-74.8991	-199.757	-134.230
91:	11108.870:	4.11748	5.71981	2.72981	-76.4953	-204.199	-137.253
92:	11108.870:	4.13498	5.73779	2.72978	-77.8038	-207.841	-139.732
93:	11108.870:	4.15109	5.75413	2.72975	-79.0754	-211.380	-142.141
94:	11108.870:	4.16779	5.77108	2.72982	-80.3471	-214.919	-144.549
95:	11108.870:	4.18428	5.78781	2.72992	-81.6187	-218.457	-146.958
96:	11108.870:	4.18484	5.78823	2.73003	-81.6855	-218.643	-147.085
97:	11108.870:	4.18716	5.79064	2.72995	-81.8857	-219.200	-147.464
98:	11108.870:	4.20695	5.81048	2.73027	-83.4878	-223.659	-150.499
99:	11108.870:	4.21345	5.81712	2.73028	-84.0168	-225.131	-151.502
100:	11108.870:	4.22635	5.83037	2.73025	-85.0748	-228.075	-153.506
101:	11108.870:	4.25192	5.85664	2.73020	-87.1908	-233.963	-157.515
102:	11108.870:	4.27673	5.88252	2.72986	-89.3067	-239.852	-161.524

```
103:     11108.870:  4.28524  5.89112  2.72998 -90.1149 -242.100 -163.055
104:     11108.870:  4.31403  5.92022  2.73029 -92.6455 -249.142 -167.850
105:     11108.870:  4.32456  5.93065  2.73053 -93.6039 -251.809 -169.666
106:     11108.870:  4.33436  5.93987  2.73109 -94.5623 -254.476 -171.483
107:     11108.870:  4.35277  5.95848  2.73121 -96.2805 -259.257 -174.739
108:     11108.870:  4.37086  5.97729  2.73092 -97.9988 -264.039 -177.995
109:     11108.870:  4.38919  5.99615  2.73079 -99.7171 -268.820 -181.251
110:     11108.870:  4.39342  6.00062  2.73068 -100.139 -269.994 -182.050
111:     11108.870:  4.42375  6.03185  2.73045 -103.092 -278.210 -187.646
112:     11108.870:  4.42003  6.02787  2.73060 -102.791 -277.374 -187.077
113:     11108.870:  4.43195  6.03990  2.73068 -103.993 -280.718 -189.355
114:     11108.870:  4.43902  6.04707  2.73071 -104.711 -282.716 -190.716
115:     11108.870:  4.45305  6.06125  2.73079 -106.147 -286.711 -193.437
116:     11108.870:  4.46681  6.07505  2.73093 -107.583 -290.706 -196.158
117:     11108.870:  4.47692  6.08503  2.73113 -108.687 -293.779 -198.251
118:     11108.870:  4.49525  6.10365  2.73116 -110.647 -299.232 -201.966
119:     11108.870:  4.51301  6.12197  2.73102 -112.606 -304.684 -205.680
120:     11108.870:  4.53107  6.14042  2.73098 -114.566 -310.137 -209.394
121:     11108.870:  4.53070  6.14001  2.73098 -114.554 -310.105 -209.373
122:     11108.870:  4.53074  6.14011  2.73092 -114.566 -310.137 -209.395
123:     11108.870:  4.54388  6.15342  2.73096 -116.044 -314.248 -212.195
```

```
   124:     11108.870:  4.54910  6.15872  2.73095 -116.635 -315.893 -213.316
   125:     11108.870:  4.55943  6.16922  2.73092 -117.817 -319.182 -215.556
   126:     11108.870:  4.57993  6.19001  2.73092 -120.181 -325.760 -220.038
   127:     11108.870:  4.59575  6.20608  2.73083 -122.064 -331.000 -223.608
   128:     11108.870:  4.60914  6.21952  2.73098 -123.696 -335.540 -226.701
   129:     11108.870:  4.62805  6.23841  2.73134 -126.095 -342.214 -231.249
   130:     11108.870:  4.64748  6.25788  2.73166 -128.494 -348.890 -235.798
   131:     11108.870:  4.65512  6.26588  2.73141 -129.446 -351.538 -237.602
   132:     11108.870:  4.67425  6.28550  2.73119 -131.846 -358.215 -242.152
   133:     11108.870:  4.69170  6.30388  2.73035 -134.085 -364.445 -246.396
   134:     11108.870:  4.70907  6.32157  2.73021 -136.324 -370.674 -250.641
   135:     11108.870:  4.70524  6.31710  2.73082 -135.913 -369.533 -249.863
   136:     11108.870:  4.71413  6.32547  2.73148 -137.181 -373.060 -252.267
   137:     11108.870:  4.72446  6.33566  2.73181 -138.605 -377.022 -254.968
   138:     11108.870:  4.73464  6.34583  2.73208 -140.029 -380.985 -257.668
   139:     11108.870:  4.74476  6.35618  2.73214 -141.454 -384.947 -260.368
   140:     11108.870:  4.76631  6.37842  2.73178 -144.393 -393.126 -265.942
   141:     11108.869:  4.76523  6.37779  2.73120 -144.227 -392.664 -265.627
   142:     11108.869:  4.76643  6.37901  2.73119 -144.393 -393.126 -265.942
```

```
summary(fit.ml)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: dd
       AIC      BIC    logLik
  12848.11 12915.18 -6412.053

Random effects:
 Formula: ~1 + ses + Sex | id
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.5507024 (Intr) ses
ses         0.1796016 0.998
SexMale     0.3973486 0.999  0.998
Residual    6.0997280

Fixed effects: mathach ~ ses * Sector + Sex
                    Value Std.Error   DF   t-value p-value
(Intercept)      12.951009 0.4218649 1934 30.699421  0.0000
ses               1.830092 0.2807716 1934  6.518079  0.0000
SectorPublic     -1.618970 0.6103963   38 -2.652326  0.0116
SexMale           1.173682 0.3321072 1934  3.534045  0.0004
ses:SectorPublic  1.344947 0.4212879 1934  3.192466  0.0014
```

```
 Correlation:
                  (Intr) ses    SctrPb SexMal
ses                0.079
SectorPublic      -0.655 -0.058
SexMale           -0.220  0.019 -0.012
ses:SectorPublic -0.048 -0.667  0.169 -0.032

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.06009142 -0.75212643  0.04604295  0.79450300  2.66713374

Number of Observations: 1977
Number of Groups: 40
```

Notes on testing: ML vs REML —-

The default for lme is fitting with 'Residual Maximum Likelihood' (REML) – it has other interpretations but everyone agrees on 'REML' – With REML, maximum likelihood is applied ONLY to the parameters for random effects, i.e. the G matrix and sigma (or R). The Betas play no direct role, they are only estimated with Generalized Least Squares as an afterthought. This means that the likelihood with REML is only informative for the the RE model: the G

and R matrices. You can't compare two models that have different FE models.

Full maximum likelihood (ML) does look at Betas, G and R together. So the likelihood with ML can be used to test two models with nested FE models.

The consequence is that:

If two models have been fitted with REML, the must have identical FE models to compare them with 'anova': p-values, AIC, BIC.

If two models have been fitted with ML, you can use 'anova' regardless of the FE or the RE models — PROVIDED they have the same response variable. This includes requiring that they have the same set of observations, hence the same omitted rows due to missing values.

Wald tests can be used for both ML or REML fits. They might be more accurate with REML fits. However, Wald test tend to useful mainly for the FE part of the model. 'wald' in 'spida2' only works for the FE model. To compare two RE models (with identical FE models) LRT is generally better than Wald. However, classic p-values tend to be too large in many cases and can be adjusted through simulation.

Summary:

- Wald is ok for the FE model whether ML or REML
- anova (with simulation adjustment) is ok for the RE model provided the FE models are

identical

- anova can be used to test different FE models or models that differ in both FEs and REs ONLY IF THE MODELS ARE fitted with ML.

Diagnostics —-

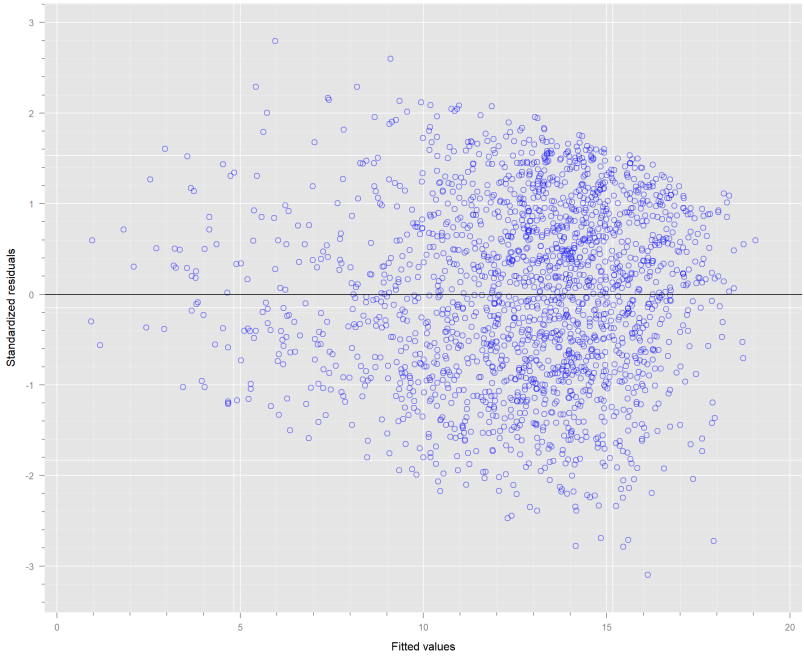
The 'car' package has methods to conduct influence diagnostics with a wide range of models include 'lme' models. I plan to incorporate a discussion of these methods soon.

Diagnostics with Level 1 residuals —-

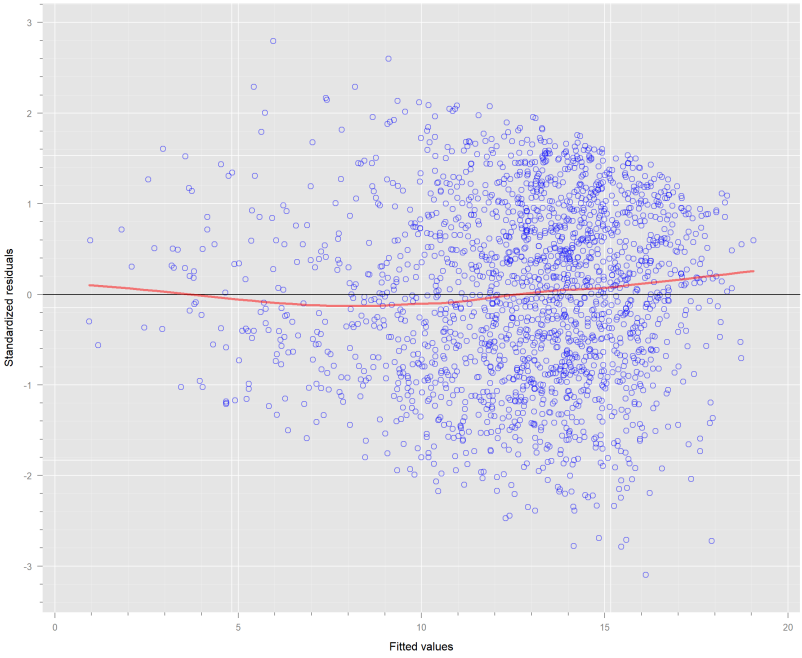
Level 1 residuals

Level 1 standardized residuals versus yhat:

```
gd(pch=1, alpha = .5)
plot(fitc)
```


Standardized residuals
3
2
1
0
-1
-2
-3
0
5
10
15
20
Fitted values

```
plot(fitc) + layer(panel.loess(x,y, lwd=3,col='red'))
```


Standardized residuals
3
2
1
0
-1
-2
-3
0
5
10
15
20
Fitted values

Not as generous as plot for 'lm'.

In contrast with least-squares, it is possible to have a non-flat least-squares regression here.

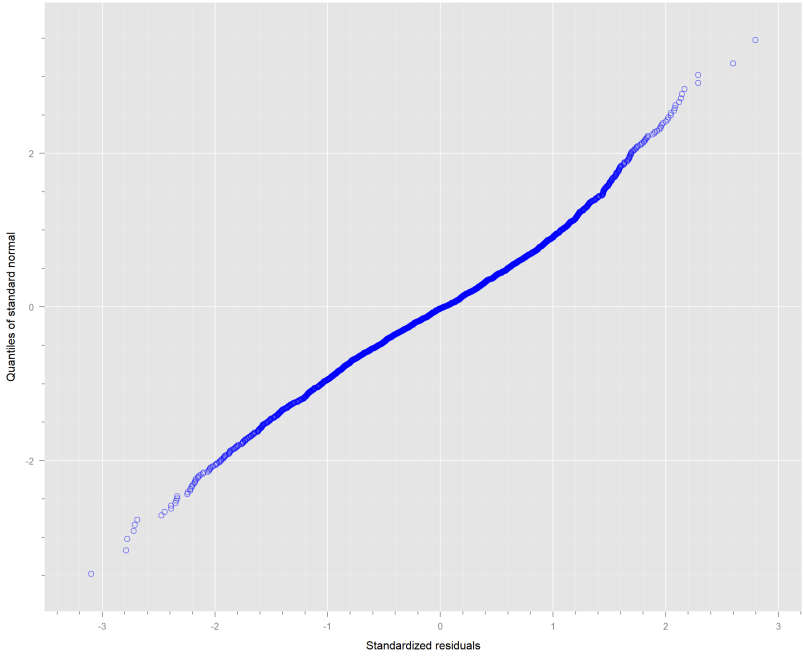
```
diag.df <- data.frame( resid = resid(fitc), fitted = fitted(fitc))
summary( lm( resid ~ fitted, diag.df))
```

```
Call:
lm(formula = resid ~ fitted, data = diag.df)

Residuals:
     Min       1Q   Median       3Q      Max
-19.1659  -4.5557   0.1715   4.7476  17.5492

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.84287    0.60009  -1.405    0.160
fitted       0.06594    0.04572   1.442    0.149

Residual standard error: 6.056 on 1975 degrees of freedom
Multiple R-squared:  0.001052,   Adjusted R-squared:  0.0005463
F-statistic:  2.08 on 1 and 1975 DF,  p-value: 0.1494
```

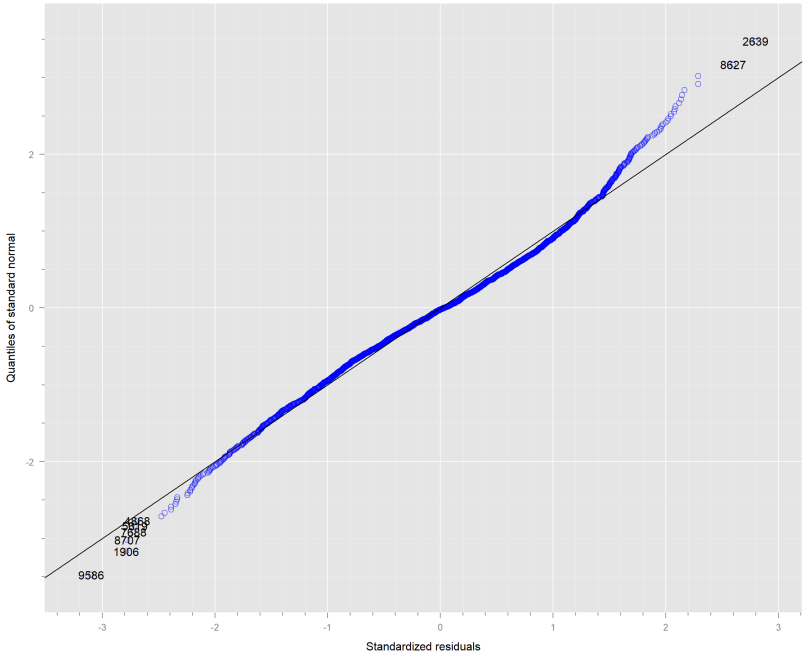
It could be a symptom of lack of independence between school-level random effects and level 1 errors or between school-level random effects and other predictors. One possible cure is adding a contextual effect if there isn't one.

Normality:

```
qqnorm( fitc  )
```


Quantiles of standard normal
2
0
-2
-3
-2
-1
0
1
2
3
Standardized residuals

```
qqnorm( fitc , abline = c(0,1), id = .01 )
```

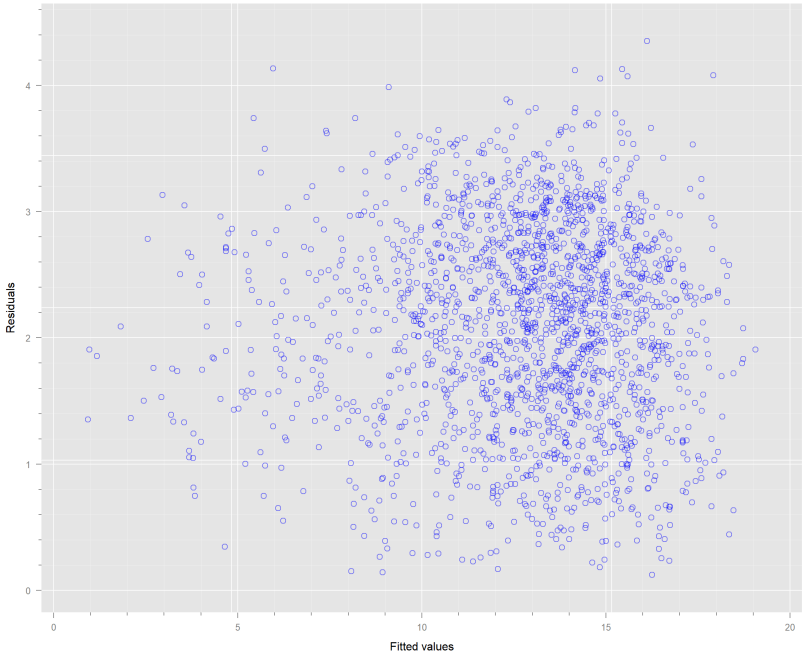

2639
8627
4488
5858
7858
8707
1906
9586
Quantiles of standard normal
Standardized residuals
2
0
-2
-3
-2
-1
0
1
2
3

BEWARE: data on horizontal axis SO distribution is slightly platykurtic. Variance might be overestimated with extreme values thereby overajusting.

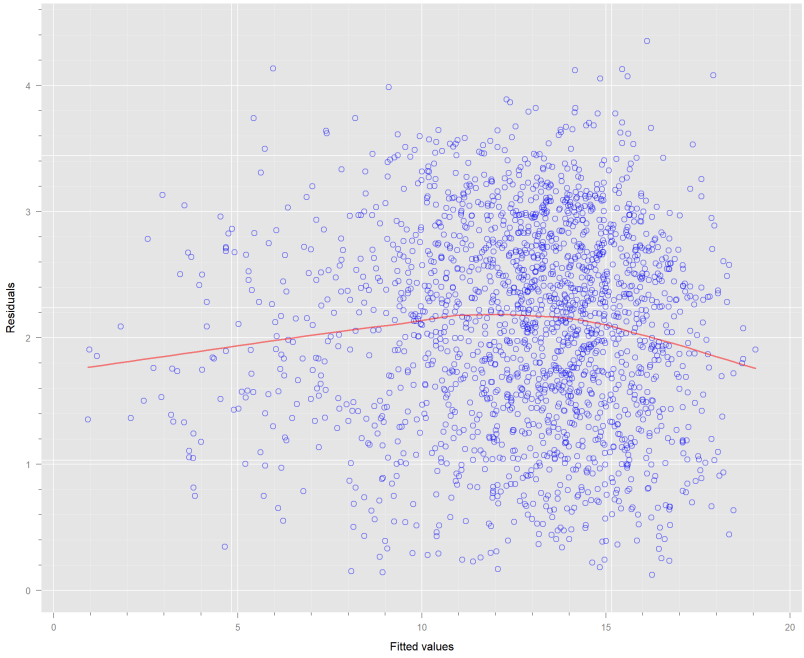
Scale - location plot: —-

Using sqrt(abs(resid)) produces an approximate normal if everything is ok

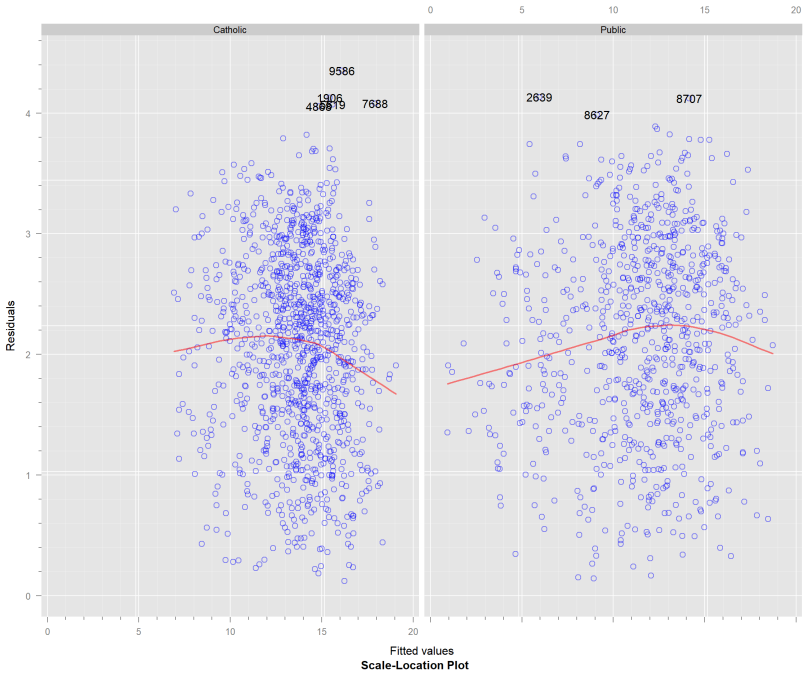
```
plot( fitc , sqrt( abs( resid(.))) ~ fitted(.))
```


Residuals
0
1
2
3
4
0
5
10
15
20
Fitted values

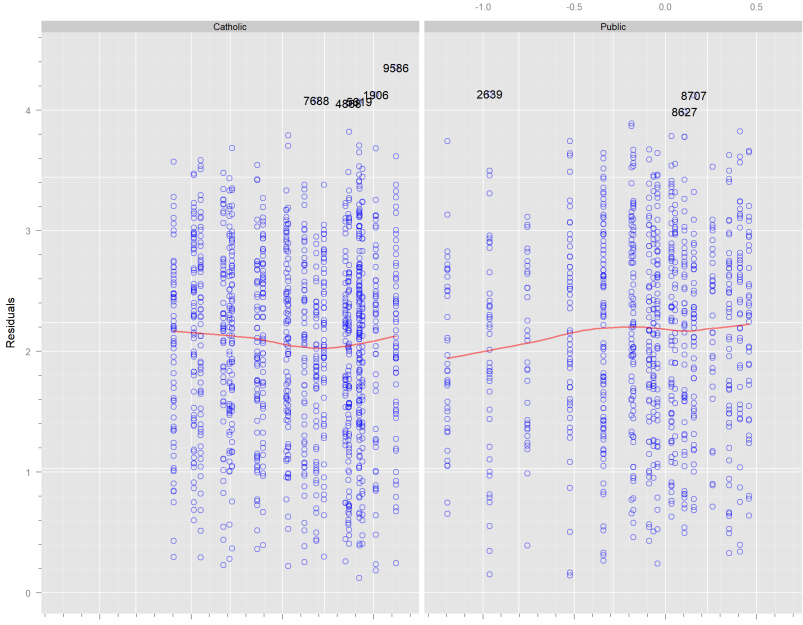
```
plot( fitc , sqrt( abs( resid(.))) ~ fitted(.)) +
  layer(panel.loess(x, y, lwd = 2, col = 'red'))
```


Residuals
Fitted values
0
1
2
3
4
0
5
10
15
20

```
plot( fitc ,
      sqrt( abs( resid(.))) ~ fitted(.)| Sector,
      id = .01, sub = "Scale-Location Plot") +
  layer(panel.loess(x, y, lwd = 2, col = 'red'))
```


Catholic
Public
9586
1906
7688
2639
8707
8627
Residuals
Fitted values
Scale-Location Plot

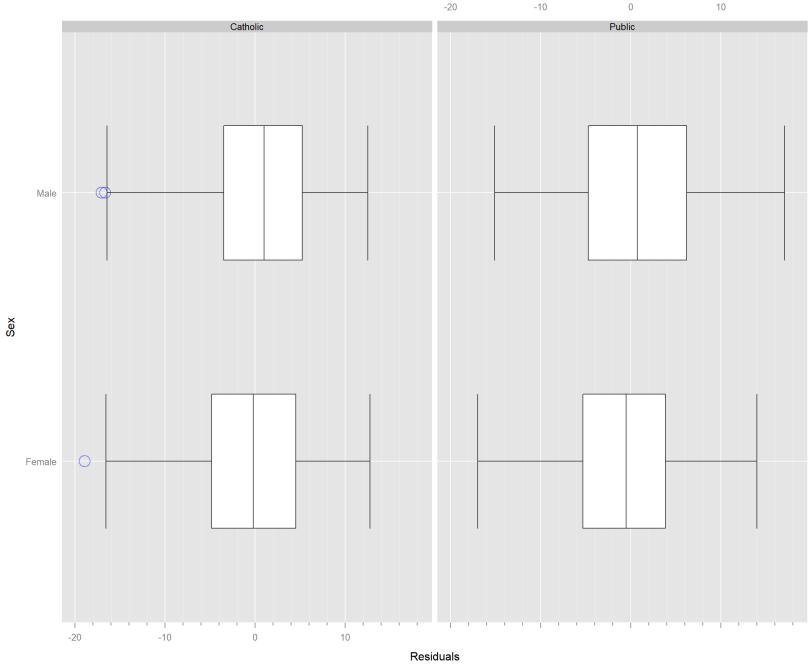
```
plot( fitc , sqrt( abs( resid(.))) ~ cvar(ses,school)| Sector,
      id = .01, sub = "Scale-Location Plot") +
  layer(panel.loess(x, y, lwd = 2, col = 'red'))
```



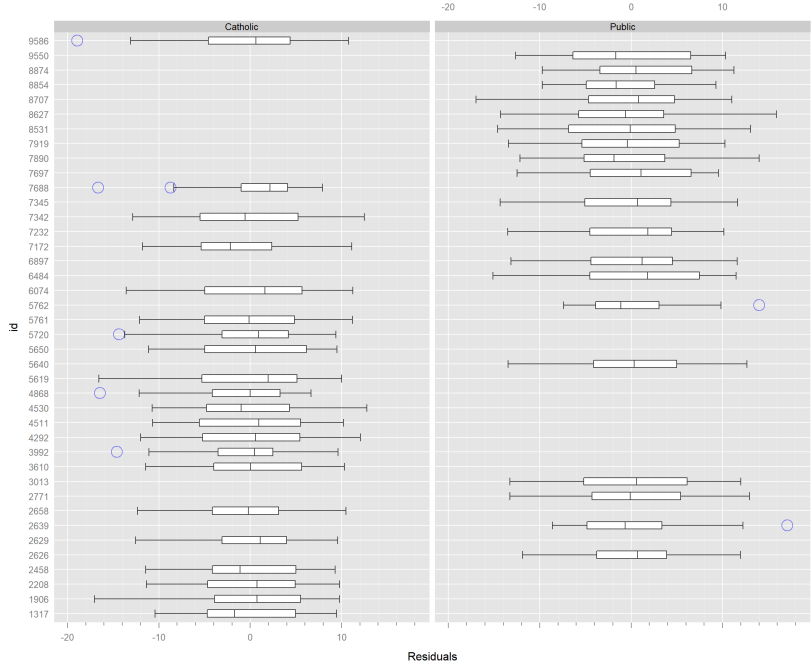
Scale-Location Plot

other views of residuals:

```
plot( fitc, Sex ~ resid(.) | Sector)
```

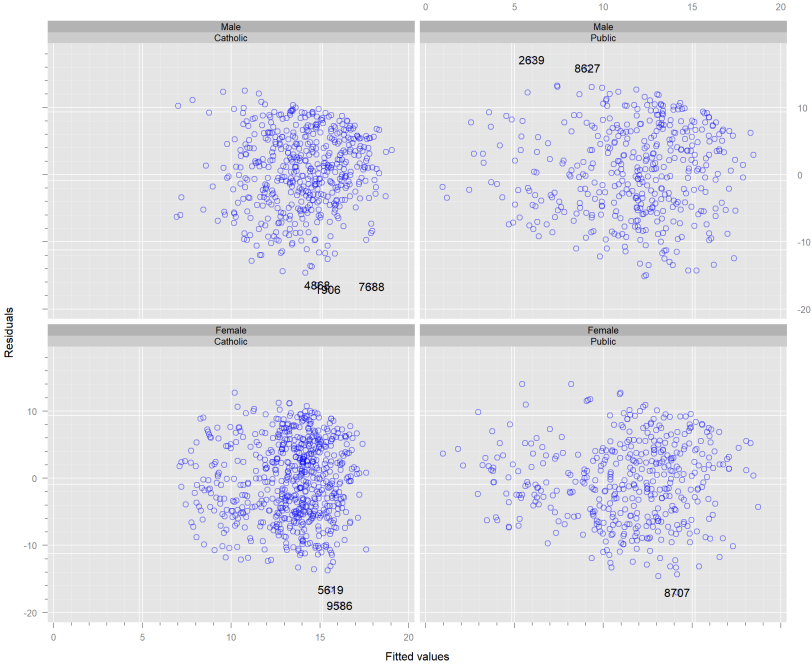

Catholic
Public
-20
-10
0
10
Male
Female
Sex
-20
-10
0
10
Residuals

```
plot( fitc, id ~ resid(.) | Sector)
```


Catholic
Public
id
9586
9550
8874
8854
8707
8627
8531
7919
7890
7697
7688
7345
7342
7232
7172
6897
6484
6074
5762
5761
5720
5650
5640
5619
4868
4530
4511
4292
3992
3610
3013
2771
2658
2639
2629
2626
2458
2208
1906
1317
-20
-10
0
10
Residuals

Omitted variables:

```
plot( fitc , resid(.) ~ fitted(.)| Sector*Sex, id = .01)
```


Male
Catholic
Male
Public
Female
Catholic
Female
Public
2639
8627
4868
1906
7688
5619
9586
8707
Residuals
Fitted values
0
5
10
15
20
-20
-10
0
10

Why MM residual plots don't look like OLS residual plots:

Question: Performing OLS regression diagnostics, you find that the residuals are correlated with the predicted values. What would this signify and what action should you take?

Answer: at end of script.

With MM residuals:

Why are residuals not uncorrelated with predicted values (a canon of OLS)?

Reason: if we were using OLS on the pooled data, resids would be uncorrelated.

But MMs is roughly like doing OLS on pooled data, then adding an additional fit at the cluster level.

If unaccounted between cluster effects are in the same direction as within cluster effect then higher residuals with higher fitted values will get a second fit, This shrinks the high positive OLS residuals and stretches the corresponding fitted values creating a negative slope in the residual vs fitted plot.

Diagnostics with Level 2 residuals —-

Make cvar(ses,id) a variable in dd

```
dd$ses.m <- with( dd, cvar( ses, id))
fitc <- lme( mathach ~ (ses + ses.m) * Sector, dd,
             random = ~ 1 + ses | id )
```

- 'ranef(fitc)' produces Level 2 errors: u in each school
- 'coef(fitc)' produces Level 2 BLUPS: beta = gamma + u

```
some( coef ( fitc ) )      # BLUP in each cluster
```

```
      (Intercept)      ses    ses.m SectorPublic ses:SectorPublic
2208    13.53819 1.817657 2.407785    -0.934345         1.173092
2626    14.30123 1.714866 2.407785    -0.934345         1.173092
3610    14.49506 1.824884 2.407785    -0.934345         1.173092
3992    13.25713 1.564306 2.407785    -0.934345         1.173092
4292    14.42899 1.294163 2.407785    -0.934345         1.173092
5762    13.17761 1.514336 2.407785    -0.934345         1.173092
6074    14.52082 1.565613 2.407785    -0.934345         1.173092
6484    14.51768 1.368009 2.407785    -0.934345         1.173092
7890    12.86207 1.393502 2.407785    -0.934345         1.173092
9586    12.56447 1.694839 2.407785    -0.934345         1.173092
      ses.m:SectorPublic
2208           1.414896
2626           1.414896
```

```
   3610           1.414896
   3992           1.414896
   4292           1.414896
   5762           1.414896
   6074           1.414896
   6484           1.414896
   7890           1.414896
   9586           1.414896
```

```
some( ranef ( fitc ) )    # RE in each cluster
```

```
      (Intercept)          ses
   1906  0.36252477  0.050644237
   2208  0.18213282  0.118259831
   3013  0.35388133  0.046187778
   3610  1.13900591  0.125486781
   4868 -1.63966343  0.004294988
   5619  0.02874504  0.502254674
   5761 -0.63626758  0.270788614
   6484  1.16162333 -0.331388396
   7172 -2.88069356  0.202229471
   8531 -1.15715031  0.079146909
```

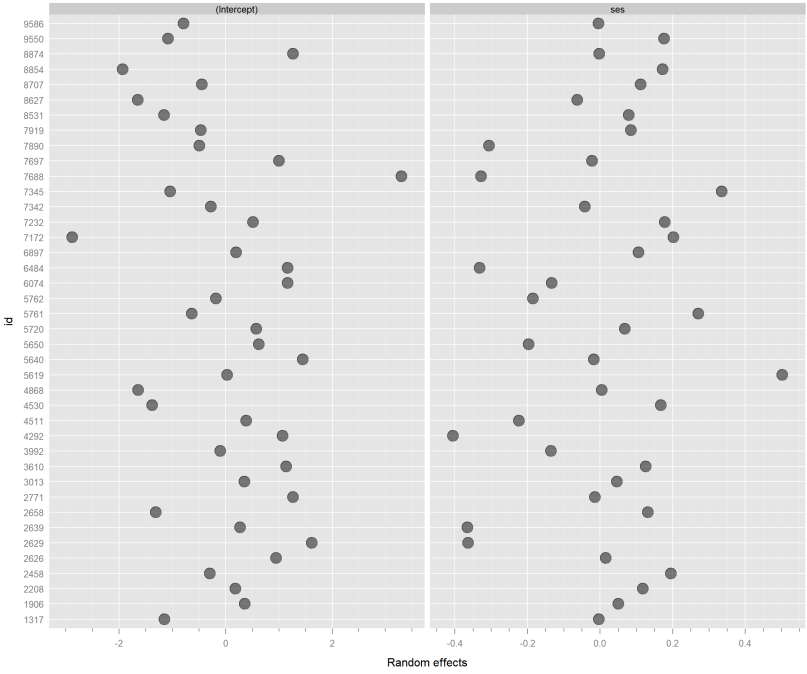
Note: coef(fitc) = fixed.effect + random.effect pairs(cbind(coef(fitc), ranef(fitc))) # can you interpret what this says?

```
re <- ranef( fitc, aug = T)   # creates data frame with up( dd, ~ school) variab
some( re )
```

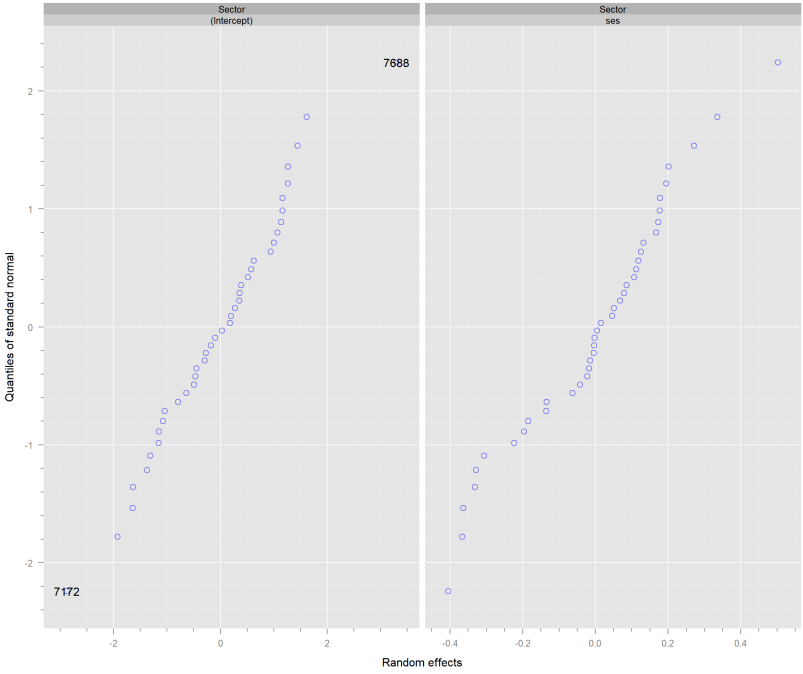
```
      (Intercept)          ses school   mathach    Sex Minority Size
1317  -1.1509678 -0.003523084   1317 13.177687 Female      Yes  455
2658  -1.3121006  0.131984540   2658 13.396156 Female       No  780
4292   1.0729371 -0.405233961   4292 12.864354   Male      Yes 1328
6074   1.1647641 -0.133784197   6074 13.779089 Female       No 2051
6484   1.1616233 -0.331388396   6484 12.912400 Female       No  726
6897   0.1989012  0.106243527   6897 15.097633 Female       No 1415
7172  -2.8806936  0.202229471   7172  8.066818 Female      Yes  280
7342  -0.2737076 -0.042255077   7342 11.166414   Male       No 1220
8707  -0.4475834  0.112155798   8707 12.883938 Female       No 1133
9550  -1.0779843  0.176773458   9550 11.089138 Female       No 1532
       Sector PRACAD DISCLIM  n       ses.m       ses.cwg ses.iqr
1317 Catholic   0.95  -1.694 48  0.34533333  1.502975e-17  0.7825
2658 Catholic   0.79  -0.961 45  0.43844444 -9.250654e-18  0.9100
4292 Catholic   0.76  -0.674 65 -0.48615385 -5.096167e-18  0.8600
6074 Catholic   0.32  -1.018 56 -0.27675000 -1.487390e-18  0.8175
```

```
   6484   Public   0.19   0.218 35 -0.18514286 -2.762392e-18  0.7400
   6897   Public   0.55  -0.361 49  0.34955102 -1.359125e-17  1.0700
   7172 Catholic   0.05   1.013 44 -0.28981818 -3.027573e-17  1.0275
   7342 Catholic   0.46   0.380 58 -0.44782759 -4.974128e-19  0.7125
   8707   Public   0.48   1.542 48  0.15512500  8.673617e-19  1.1350
   9550   Public   0.45   0.791 29  0.05303448  9.077389e-18  1.0200
        ses.trange    ses.sd ses.rank minority.diff id.sec    ses.cvar
   1317      1.166 0.5561583     24.5    -0.2676044  C1317  0.34533333
   2658      1.698 0.6402846     23.0    -0.1830556  C2658  0.43844444
   4292      1.708 0.6511382     33.0    -0.0150000  C4292 -0.48615385
   6074      1.440 0.6271235     28.5    -0.3279630  C6074 -0.27675000
   6484      1.932 0.6958345     18.0    -0.5227273  P6484 -0.18514286
   6897      2.090 0.7445231     25.0    -0.7075000  P6897  0.34955102
   7172      1.686 0.6764417     22.5     0.3915447  C7172 -0.28981818
   7342      1.323 0.5648459     29.5     0.1027413  C7342 -0.44782759
   8707      2.169 0.8042577     24.5    -0.5873333  P8707  0.15512500
   9550      1.824 0.7847035     15.0    -0.5623913  P9550  0.05303448
```

```
plot( re )     # special plot method
```


(Intercept)
ses
id
9586
9550
8874
8854
8707
8627
8531
7919
7890
7697
7688
7345
7342
7232
7172
6897
6484
6074
5762
5761
5720
5650
5640
5619
4868
4530
4511
4292
3992
3610
3013
2771
2658
2639
2629
2626
2458
2208
1906
1317
-2
0
2
-0.4
-0.2
0.0
0.2
0.4
Random effects

```
qqnorm( fitc, ~ ranef(.) | Sector, id = .05)
```


Sector
(Intercept)
Sector
ses
7688
7172
Quantiles of standard normal
Random effects
2
1
0
-1
-2
-2
0
2
-0.4
-0.2
0.0
0.2
0.4

```
if(interactive) {
  Init3d()
  Plot3d( `(Intercept)` ~ ses + ses.m | Sector, re)  # note funny names in backq
  Ell3d()
}
```

EXERCISE:

- Plot random effects against variables you feel might reveal a pattern
- Experiment with the alternative RE model.

Influence diagnostics – drop one row or one cluster at a time —-

Here's a quick dropone function

```
dropone <- function(fit) {
  data <- getData(fit)
  index <- 1:nrow(data)
  lapply(index, function(i) fixef(update(fit,data=data[-i,])))
}
```

Here's a fancier one that will drop clusters and return NA if there's an error in fitting

```r
dropone <- function(fit, form = NULL, FUN = fixef, data = getData(fit),...) {
  if(is.null(form)) {
    by <- factor(1:nrow(data))
    data$by <- by
    dframe <- data
  } else {
    by <- model.frame(form, data)
    by <- do.call(paste, c(by, sep='/'))
    data$by <- factor(by)  # bad if by already exists in data
    dframe <- up(data, form)
  }
  values <- dframe$by
  names(values) <- values
  ret <- lapply(values, function(v) {
    ret <- try(update(fit, data = data[by != v, ,drop=FALSE],...))
    if(class(ret) == 'try-error') NA else FUN(ret)
  })
  ret <- do.call(rbind,ret)
  colnames(ret) <- paste0('b_',colnames(ret))
  cbind(dframe, ret)
}
```

Ideas for improvement:

- optionally standardize relative to SEs

The following takes about 5 minutes so it's been precooked so we can take it out of the oven.

```
# system.time(
#    fits.dropone <- dropone(fitc)
# )
```

Take fits.dropone out of the oven:

```
con <- url('http://blackwell.math.yorku.ca/ICPSR/fits.dropone.rda')
load(con, verbose = T)
```

```
   Loading objects:
     fits.dropone
```

```
close(con)
```

```
str(fits.dropone)
```

```
   'data.frame':    1977 obs. of  26 variables:
    $ school              : int  1317 1317 1317 1317 1317 1317 1317 1317 1317 13
    $ mathach             : num  12.86 8.96 4.76 21.41 20.75 ...
    $ ses                 : num  0.882 0.932 -0.158 0.362 1.372 ...
```

```
$ Sex                : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1
$ Minority           : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 2 1 2 2 2 .
$ Size               : int  455 455 455 455 455 455 455 455 455 455 ...
$ Sector             : Factor w/ 2 levels "Catholic","Public": 1 1 1 1 1 1
$ PRACAD             : num  0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.
$ DISCLIM            : num  -1.69 -1.69 -1.69 -1.69 -1.69 ...
$ n                  : int  48 48 48 48 48 48 48 48 48 48 ...
$ ses.m              : num  0.345 0.345 0.345 0.345 0.345 ...
$ ses.cwg            : num  0.5367 0.5867 -0.5033 0.0167 1.0267 ...
$ ses.iqr            : num  0.782 0.782 0.782 0.782 0.782 ...
$ ses.trange         : num  1.17 1.17 1.17 1.17 1.17 ...
$ ses.sd             : num  0.556 0.556 0.556 0.556 0.556 ...
$ ses.rank           : num  39 40 6 25 47 20 20 14.5 8 33 ...
$ minority.diff      : num  -0.268 -0.268 -0.268 -0.268 -0.268 ...
$ id                 : int  1317 1317 1317 1317 1317 1317 1317 1317 1317 13
$ id.sec             : Factor w/ 40 levels "P5762","P2639",..: 33 33 33 33
 ..- attr(*, "scores")= num [1:40(1d)] -1.194 -0.964 -0.757 999.403 -0.523 .
 .. ..- attr(*, "dimnames")=List of 1
 .. .. ..$ : chr  "P5762" "P2639" "P8854" "C4530" ...
$ by                 : Factor w/ 1977 levels "1","2","3","4",..: 1 2 3 4 5
$ b_(Intercept)      : num  13.4 13.4 13.4 13.3 13.4 ...
```

```
    $ b_ses                : num  1.7 1.71 1.69 1.7 1.69 ...
    $ b_ses.m              : num  2.41 2.42 2.43 2.39 2.4 ...
    $ b_SectorPublic       : num  -0.935 -0.937 -0.942 -0.928 -0.933 ...
    $ b_ses:SectorPublic   : num  1.17 1.16 1.18 1.17 1.19 ...
    $ b_ses.m:SectorPublic: num  1.41 1.41 1.39 1.43 1.42 ...
```

```
# scatterplotMatrix(fits.dropone[,grep('^b_',names(fits.dropone))], id = TRUE)
```

Dropping one school at a time

```
system.time(
  fits.dropschool <- dropone(fitc, ~ id) # only 6 seconds
)
```

```
   user  system elapsed
   4.67    0.06    5.16
```

- If some models don't converge, rerun fitc with larger msMaxIter, returnObject = T, sing.tol = 10e-20 to see if all drops will converge.

```
fitc$call
```

```
lme.formula(fixed = mathach ~ (ses + ses.m) * Sector, data = dd,
    random = ~1 + ses | id)
```

```
fitc <- update(fitc, control = list(returnObject = TRUE, msMaxIter = 1000) )
fitc$call
```

```
   lme.formula(fixed = mathach ~ (ses + ses.m) * Sector, data = dd,
       random = ~1 + ses | id, control = list(returnObject = TRUE,
           msMaxIter = 1000))
```

```
system.time(
  fits.dropschool <- dropone(fitc, ~ id) # only 6 seconds
)
```

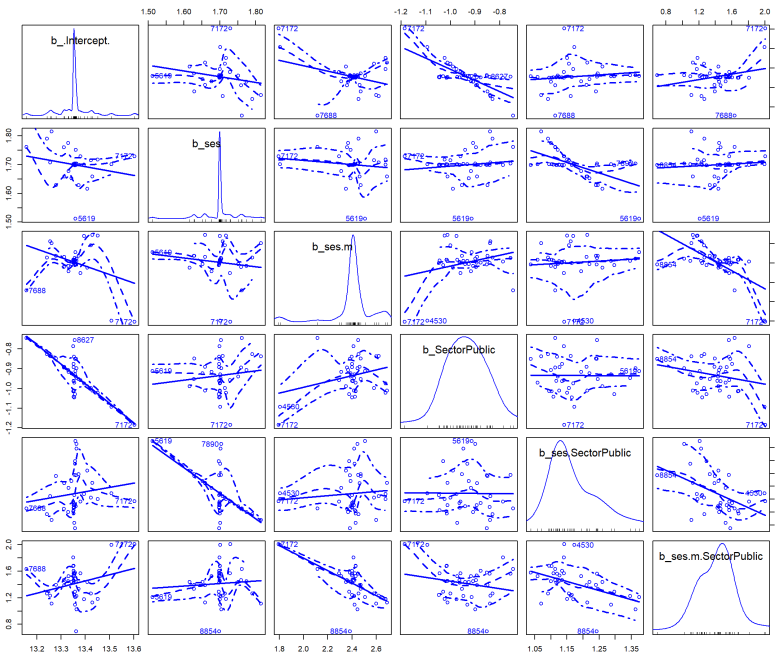
```
      user  system elapsed
      4.06    0.00    4.30
```

```
str(fits.dropschool)
```

```
   'data.frame':    40 obs. of  21 variables:
    $ school              : int  1317 1906 2208 2458 2626 2629 2639 2658 2771 30
    $ Size                : int  455 400 1061 545 2142 1314 2713 780 415 760 ...
    $ Sector              : Factor w/ 2 levels "Catholic","Public": 1 1 1 1 2 1
    $ PRACAD              : num  0.95 0.87 0.68 0.89 0.4 0.81 0.14 0.79 0.24 0.5
    $ DISCLIM             : num  -1.694 -0.939 -0.864 -1.484 0.142 ...
    $ n                   : int  48 53 60 57 38 57 42 45 55 53 ...
    $ ses.m               : num  0.3453 0.5116 0.4232 0.2278 -0.0648 ...
```

```
    $ ses.iqr             : num  0.782 1.01 0.983 0.79 0.597 ...
    $ ses.trange          : num  1.17 1.59 1.5 1.95 1.38 ...
    $ ses.sd              : num  0.556 0.614 0.598 0.658 0.56 ...
    $ minority.diff       : num  -0.268 -0.366 NaN -0.582 NaN ...
    $ id                  : Factor w/ 40 levels "1317","1906",..: 1 2 3 4 5 6 7
    $ id.sec              : Factor w/ 40 levels "P5762","P2639",..: 33 39 37 32
    $ ses.cvar            : num  0.3453 0.5116 0.4232 0.2278 -0.0648 ...
    $ by                  : Factor w/ 40 levels "1317","1906",..: 1 2 3 4 5 6 7
    $ b_(Intercept)       : num  13.4 13.3 13.3 13.4 13.4 ...
    $ b_ses               : num  1.71 1.68 1.65 1.63 1.7 ...
    $ b_ses.m             : num  2.58 2.31 2.39 2.47 2.42 ...
    $ b_SectorPublic      : num  -0.999 -0.917 -0.925 -0.954 -1.012 ...
    $ b_ses:SectorPublic  : num  1.16 1.19 1.22 1.25 1.13 ...
    $ b_ses.m:SectorPublic: num  1.26 1.52 1.44 1.33 1.45 ...
```

```
scatterplotMatrix(fits.dropschool[,grep('^b_',names(fits.dropschool))], id = TRU
```


b_.Intercept.
b_ses
b_ses.m
b_SectorPublic
b_ses.SectorPublic
b_ses.m.SectorPublic

What does this show?

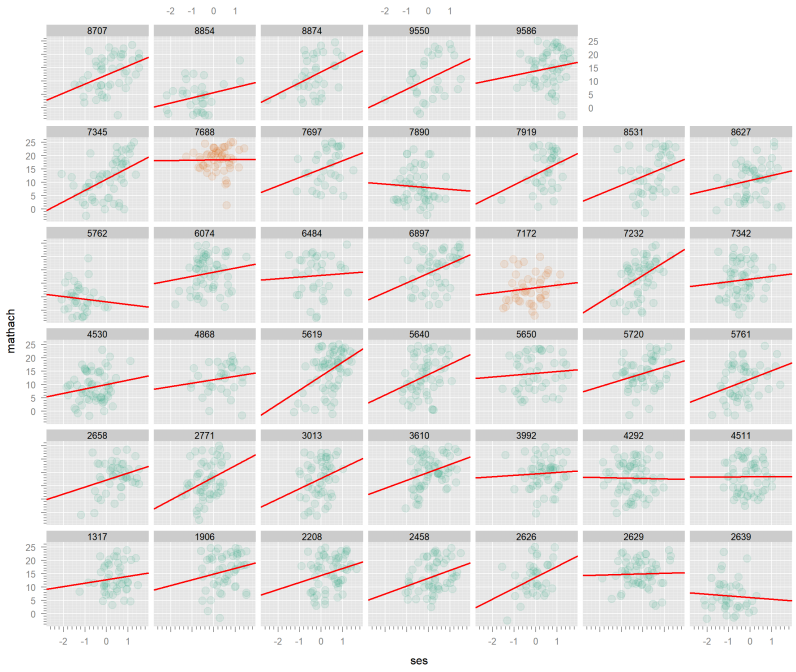
Which points represent Catholic schools and which Public schools?

```
if(interactive) {
  names(fits.dropschool)
  Init3d()
  Plot3d(`b_(Intercept)` ~ b_ses + `b_ses:SectorPublic` | Sector,
         fits.dropschool, sphere.size = 1.5)
  Id3d() # to identify schools
}
id_miss <- is.na(fits.dropschool$b_ses)
categ <- ifelse(dd$id %in% c('7688','7172'), 'out', 'ok')
subset( fits.dropschool, id %in% c('7688','7172'))
```

```
      school Size   Sector PRACAD DISCLIM  n      ses.m ses.iqr ses.trange
7172    7172  280 Catholic   0.05   1.013 44 -0.2898182  1.0275      1.686
7688    7688 1410 Catholic   0.65  -0.575 54  0.1858889  0.8275      1.378
        ses.sd minority.diff   id id.sec   ses.cvar   by b_(Intercept)
7172 0.6764417     0.3915447 7172  C7172 -0.2898182 7172      13.60385
7688 0.5644347    -0.1387347 7688  C7688  0.1858889 7688      13.15648
        b_ses  b_ses.m b_SectorPublic b_ses:SectorPublic
7172 1.729063 1.792919      -1.185338           1.141840
```

```
   7688 1.761263 2.115580      -0.745562           1.114109
        b_ses.m:SectorPublic
   7172             2.005783
   7688             1.635031
```

```
gd(alpha=c(.1,.1))
xyplot(mathach ~ ses | paste(id), dd, groups = categ) +
  glayer(panel.lmline(x,y, lwd = 2, col = 'red'))
```


8707
8854
8874
9550
9586
7345
7688
7697
7890
7919
8531
8627
5762
6074
6484
6897
7172
7232
7342
4530
4868
5619
5640
5650
5720
5761
2658
2771
3013
3610
3992
4292
4511
1317
1906
2208
2458
2626
2629
2639
mathach
ses

Not surprising!

Exercises —-

```
# Choose the problem that seems most appropriate:
#
# 1)  Add the variable 'Sex' to analysis above. Note that Sex has
#     a contextual variable: Sex composition, a variable with
#     interestingly different features in Catholic vs Public school.
#
#     If you wish to gain some experience with convergence problems
#     use the full Level 1 model ( ~ ses*Sex | id ) for your RE model.
#     If this fails to converge, try to use the advice given above to
#     correct the problem. If you prefer not to bother for now,
#     just use the additive model:
#
#             random =   ~1 + ses + Sector |id
#
#     Try to estimate the Gender Gap under various circumstances:
#     ses and Sector if the effect of Sex interacts with these variables.
#
```

```
#     Which circumstances are associated with large gender gaps,
#     with small gender gaps?
#
#     Make graphs of the gender gap as a function of its moderators,
#     preferably with confidence bounds.
#
# 2)  Analyze the data in 'bdf' in 'nlme':
#     > data (bdf)
#     > ?bdf
#     on Language Scores of 8-Graders in The Netherlands
#
#     This is similar to the 'hs' dataset but different enough
#     to be interesting.
#
#     The response of interest is 'langPOST' a measure of language
#     achievement at the end of the school year.
#
#     Potential predictors include: ses, sex, denominat (Sector),
#     Minority, IQ.verb as well as a pretest: langPRE.
#
#     This raises the interesting question of the best use for the
```

```
#     pretest. Should it be used as a covariate or should you analyze
#     the gain score: langPOST - langPRE as the response.
#
#     As usual, the best course of action may depend on the questions
#     and intended interpretation of the results.
#
# 3)  Study the role of 'Minority' in the 'hs' data following the
#     script below
#
# 4)  Validate the model above with the other half of the data 'hs2'
#     What differences do you note?
#
```