Example of a three-level model of clustered data

Miratrix

November 5, 2015

This will illustrate fitting a three level model (with clusters inside of clusters) and extracting the various components from it. This is a rough document, but hopefully will be useful.

Load the data

We first load the data. This is a dataset extensively discussed in Rabe-Hesketh and Skrondal. I am replicating the model they propose in chapter 8.4. This story is as follows: the data set is a collection of measurements for a test-retest of two peak expiratory flow measurement devices (in English, patients were told to exhale into a device to measure their lung capacity, and they did so twice for two different measurement devices). We want to understand whether the types of meter are different, and also understand variation in subjects lung capacities, and variation in the measurement error of the meters.

In the following we load the data and look at the first few lines. We see that each subject had two measurements from the standard and the mini Wright flow meter.

```
pefr = read.dta( "pefr.dta" )

head( pefr )
```

```
##   id wp1 wp2 wm1 wm2
## 1  1 494 490 512 525
## 2  2 395 397 430 415
## 3  3 516 512 520 508
## 4  4 434 401 428 444
## 5  5 476 470 500 500
## 6  6 557 611 600 625
```

We are going to view this as three-level data. We have multiple measurements nested inside device type nested inside subject. We might imagine that different subjects have different lung capacities. We also might imagine that different subjects are going to have different biases when using the two different meters. The two observations for each meter allows us to understand the variability of measurements for a single meter for a given subject, and looking at how these vary across subjects allows us to understand how much the biases move across individuals.

Reshape the data (Optional section)

This section illustrates some advanced reshaping techniques. In particular we reshape the data twice to deal with the time and the device as different levels.

Here we go:

```
dat = reshape( pefr, direction="long", idvar = "id",
               varying=list( c("wp1","wm1"), c("wp2","wm2") ),
               times=c("wp","wm"),
               timevar="device",
               v.names=c("time1","time2") )
```

Let's see what we got:

```
head( dat )
```

```
##      id device time1 time2
## 1.wp  1     wp   494   490
## 2.wp  2     wp   395   397
## 3.wp  3     wp   516   512
## 4.wp  4     wp   434   401
## 5.wp  5     wp   476   470
## 6.wp  6     wp   557   611
```

```
subset( dat, id==1 )
```

```
##      id device time1 time2
## 1.wp  1     wp   494   490
## 1.wm  1     wm   512   525
```

The second line above shows us our first person now as two new lines, one for each device. We see the measurements correspond to the first row of the original `pefr` data.

Now we have a row for each person for each device. Next we unstack the time (and then look at what we got):

```
dat = reshape( dat, direction="long", idvar=c("id","device"),
               varying=list( c("time1","time2") ),
               v.names=c("flow") )
head( dat )
```

```
##        id device time flow
## 1.wp.1  1     wp    1  494
## 2.wp.1  2     wp    1  395
## 3.wp.1  3     wp    1  516
## 4.wp.1  4     wp    1  434
## 5.wp.1  5     wp    1  476
## 6.wp.1  6     wp    1  557
```

We look at our second person to see if the measurements have the appropriate labels. They do.

```
subset( dat, id==2 )
```

```
##        id device time flow
## 2.wp.1  2     wp    1  395
## 2.wm.1  2     wm    1  430
## 2.wp.2  2     wp    2  397
## 2.wm.2  2     wm    2  415
```

```
subset( pefr, id==2 )
```

```
##   id wp1 wp2 wm1 wm2
## 2  2 395 397 430 415
```

Another sanity check:

```
table( dat$id )
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
##  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
```

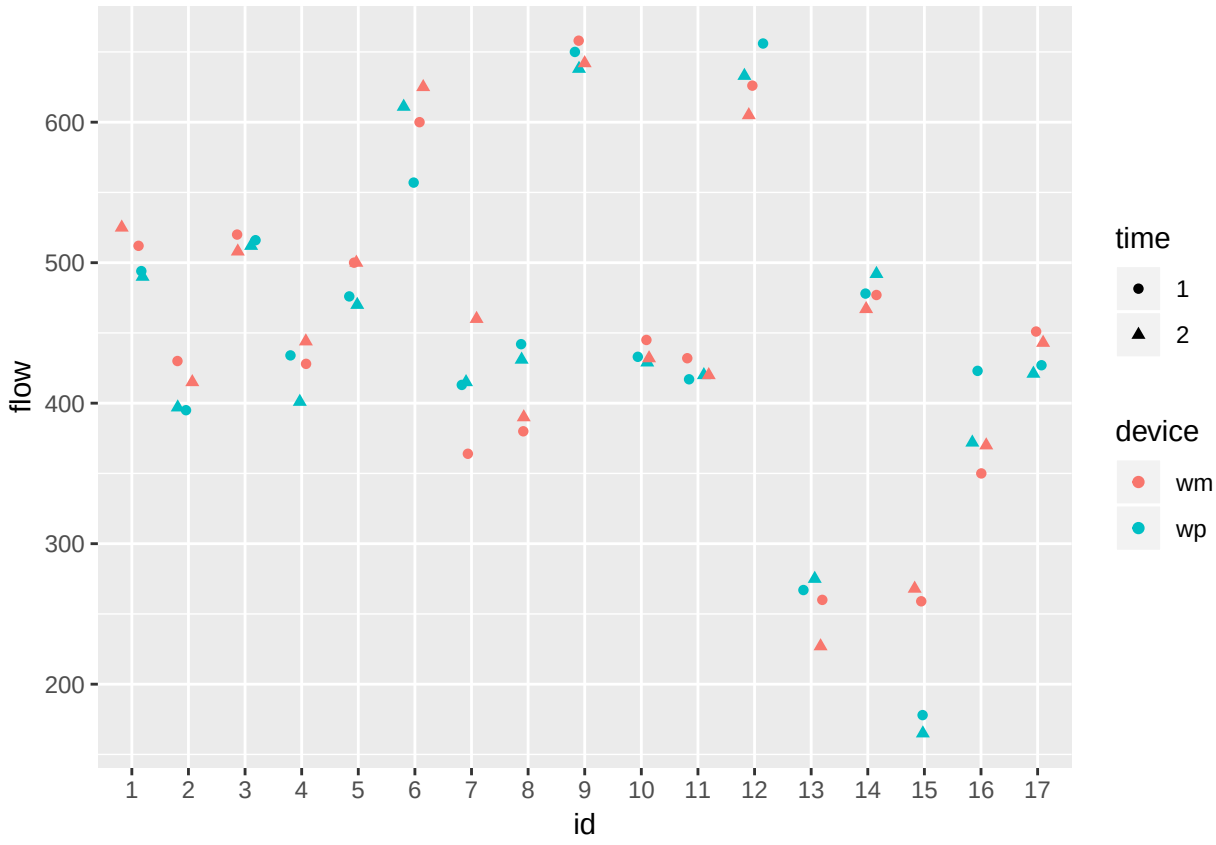
We have four measurements, still, for each person.

When reshaping data, one typically has to fiddle with all of the commands and check the results a few times to get it right.

Plot the data

We can look at the data. The following illustrates getting different colors and symbols depending on covariate information:

```
dat$id = as.factor( dat$id )
dat$device = as.factor( dat$device )
dat$time = as.factor( dat$time )
ggplot( data=dat, aes( x=id, y=flow, col=device, pch=time ) ) +
    geom_jitter( width=0.2, height=0 )
```



The mathematical model

Level 1: We have for individual i using machine j at time t:

$$Y_{ijt} = \beta_{0ij} + \beta_1 t + \epsilon_{ijt}$$

The β_1 allows for a time effect of the second measurement being systematically lower or higher than the first. We pool this across all subjects and machines.

Level 2: Our machine-level intercepts for each subject are

$$\beta_{0ij} = \gamma_{0i} + \gamma_1 D_j + u_{ij}$$

with $D_j = 1\{j = wp\}$ being an indicator (dummy variable) for the second machine. The γ_1 allows a systematic bias for the two machines (so the wp machine could tend to give larger readings than the wm machine, for example). Overall, the above says each machine expected reading varies around the subject's lung capacity, but that these expected readings will vary around the subjects true capacity by the u_{ij}. Actual readings for

subject i on machine j will hover around β_{ij} if we had the subject test over and over, according to our model (not including fatigue captured by the time coefficient).

Level 3: Finally our subject intercepts are

$$\gamma_{0i} = \mu + w_i.$$

The overall population lung capacity is μ. Subjects have larger or smaller lung capacity depending on their w_i.

The u_{ij} and w_i are each normally distributed, and independent from each other.

The w_i are how the subjects vary (i.e., their different lung capacities). The u_{ij} are the individual biases of a machine for a given subject. Looking at our plot, we see that subjects vary a lot, and machines vary sometimes within a subject (the centers of the pairs of colored points tend to be close, but not always), and the residual variance tends to be small (colored points are close together).

Fit the model

```
library( lme4 )
M1 = lmer( flow ~ device + time + (1|id) + (1|device:id), data=dat )
display( M1 )
```

```
## lmer(formula = flow ~ device + time + (1 | id) + (1 | device:id),
##     data = dat)
##             coef.est coef.se
## (Intercept) 454.43    27.84
## devicewp     -6.03     8.05
## time2        -1.03     4.37
##
## Error terms:
##  Groups    Name        Std.Dev.
##  device:id (Intercept)  19.72
##  id        (Intercept) 111.99
##  Residual               18.01
## ---
## number of obs: 68, groups: device:id, 34; id, 17
## AIC = 682.8, DIC = 709.1
## deviance = 689.9
```

Now let's connect some pieces:

- The main effects estimate $\mu = 455.46$ and $\gamma_1 = -6.03$ and $\beta_1 = -1.03$.
- The z-score of $z = -6.03/8.05 < 1$ means there is no evidence of systematic bias of one machine compared to the other.
- The estimated standard deviation of actual lung capacity is 112.
- The estimated standard deviation of how two different machines will measure the same person is 19.72. Different machines will tend to give different average measurements for the same subject.
- The estimated standard deviation of how much a repeated measurement of the same machine on the same person will vary is 18. The machines are relatively precise, given the variation in the population.
- The amount of variance explained by lung variation is $112^2/(19.72^2 + 111.99^2 + 18.01^2) = 0.94636$, i.e., most of it.

Appendix (Optional): base plot package

Here is how to build the plot without ggplot:

```
plot( flow ~ as.numeric(id), data=dat, pch=ifelse( time==1, 21, 22 ),
      col=ifelse( device=="wp", "red", "green" ) )
legend( "bottomleft", legend=c("WP-1", "WP-2","WM-1", "WM-2"),
        pch=c(21,22,21,22),
        col=c("red","red","green","green") )
```