

Small hierarchical example

georges

2022-01-28

For some information on using Rmarkdown in R scripts like this one (Roxygen style) see 3.3 Render an R script to a report.

```
library(spida2)
setwd_here()      # set working directory to directory containing this file
                  # so executing line-by-line and knitting (Alt F C)
                  # will have the same effect.
# plot(c(0,20), c(0,30), type = 'n')
# xy <- locator(24)
# plot(xy)
# save(xy, file = 'xy2.rda') # save version in a permanent file
load('xy.rda', verbose = T) # load version saved in 'xy.rda'
```

```
## Loading objects:
```

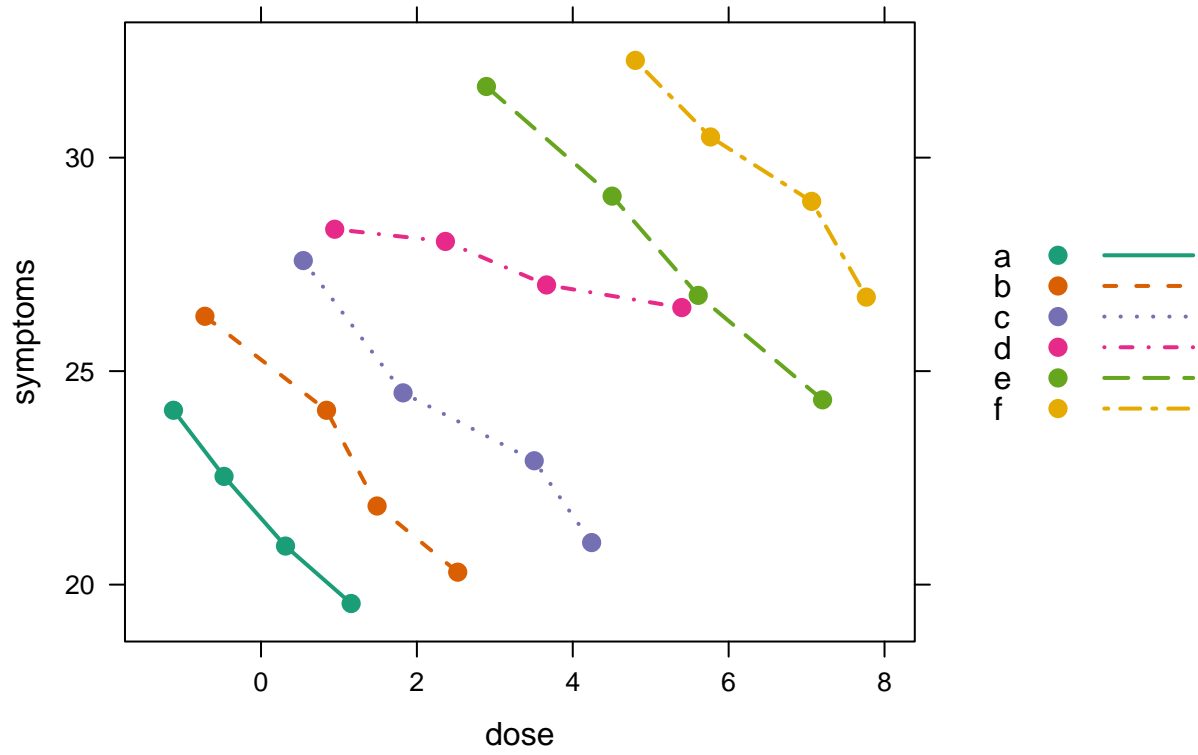
```
## xy
```

```
dd <- data.frame(id = rep(letters[1:6], each = 4), xy)
names(dd)[c(2,3)] <- c('dose', 'symptoms')
```

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

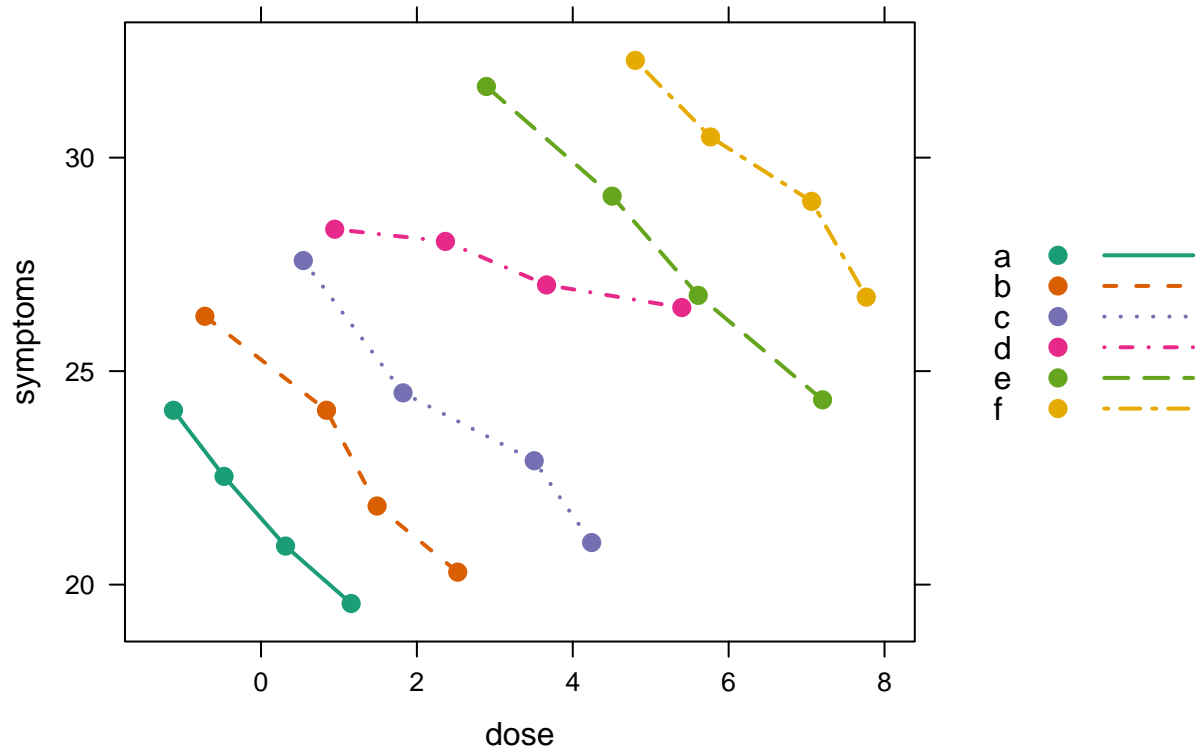
```
td(lwd = 2, pch = 16, cex = 1.2) # sets graphical parameters
```

```
xyplot(symptoms ~ dose, dd,
       groups = id,
       type = 'b',
       auto.key = list(space = 'right', lines = T))
```



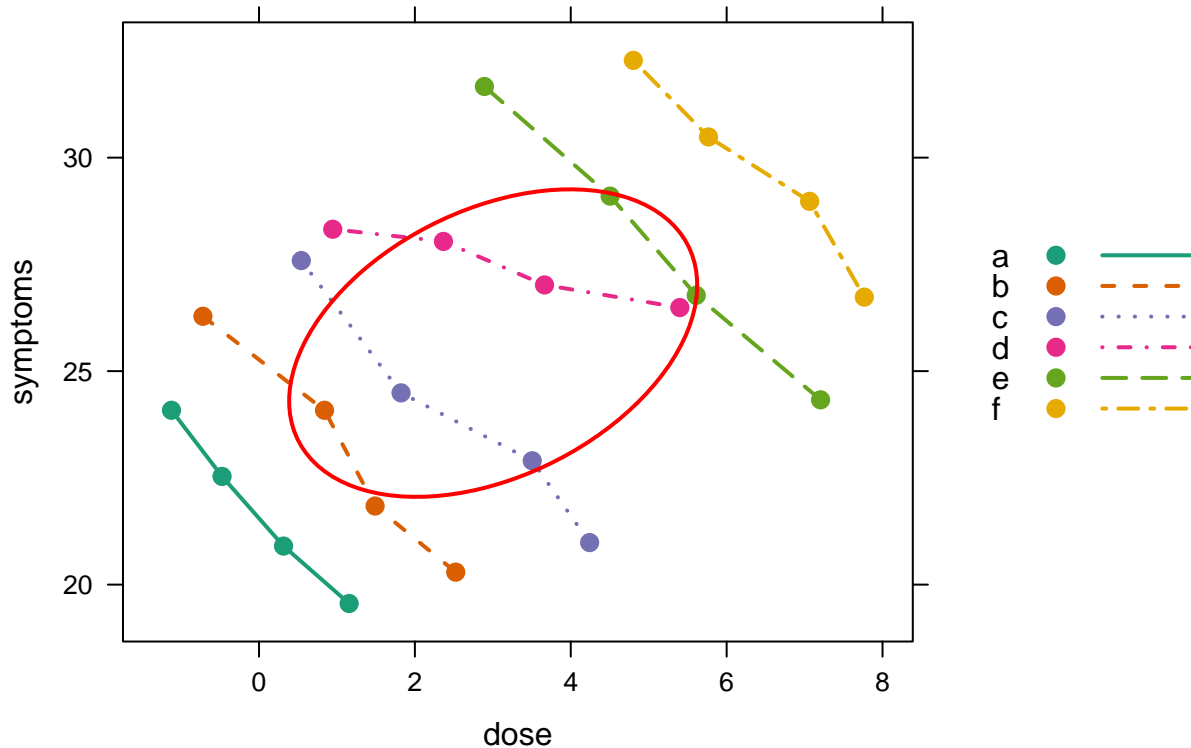
Plotting data

```
xyplot(symptoms ~ dose, dd,
  groups = id,
  type = 'b',
  auto.key = list(space = 'right', lines = T))
```



With marginal ellipse

```
xyplot(symptoms ~ dose, dd,
  groups = id,
  type = 'b',
  auto.key = list(space = 'right', lines = T)) +
  layer(panel.dell(..., col = 'red', lwd = 2))
```



With marginal ellipse and pooled regression line

```
fit.pooled <- lm(symptoms ~ dose, dd)
head(dd)
```

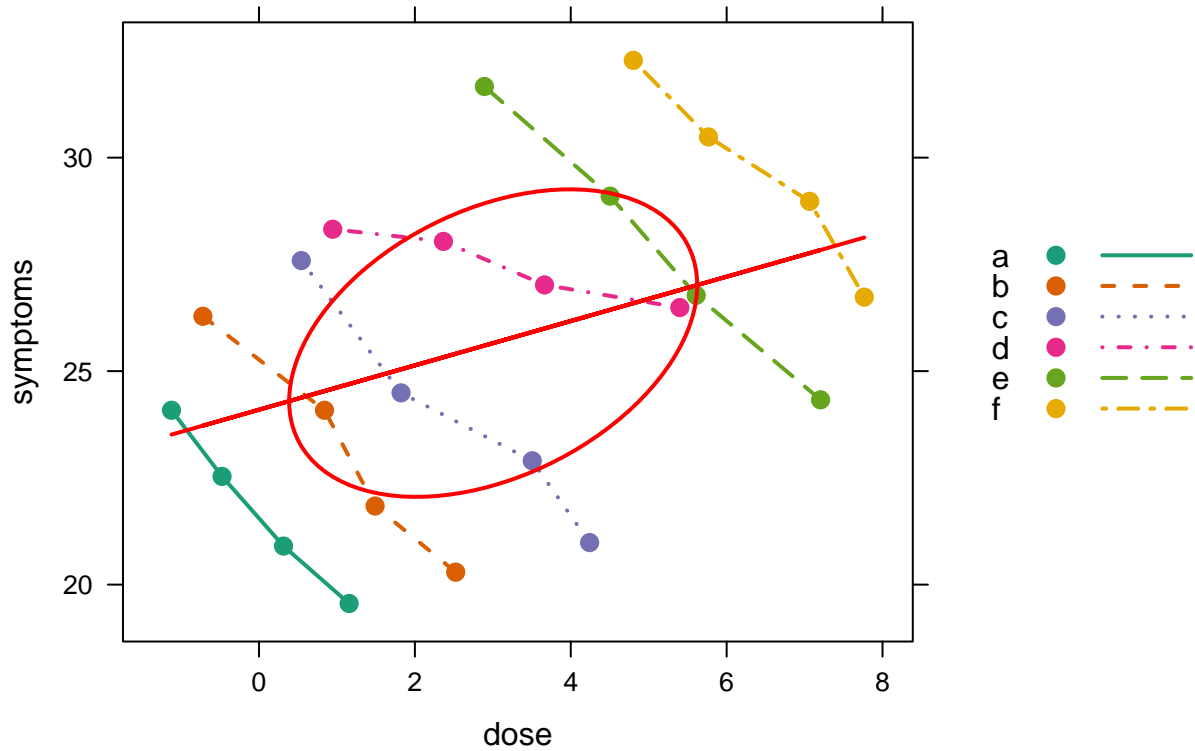
```
##   id      dose symptoms
## 1  a -1.1233197 24.08335
## 2  a -0.4745763 22.53413
## 3  a  0.3144360 20.90336
## 4  a  1.1560491 19.55798
## 5  b -0.7200468 26.28488
## 6  b  0.8404442 24.08335
```

```
dd$fit.pooled <- predict(fit.pooled)
```

Easier way to accumulating plots so:

- no cutting and pasting
- changes devolve to later plots

```
p <- xyplot(symptoms ~ dose, dd,
            groups = id,
            type = 'b',
            auto.key = list(space = 'right', lines = T)) +
  layer(panel.dell(..., col = 'red', lwd = 2)) +
  xyplot(fit.pooled ~ dose, dd, type = 'l', lwd = 2, col = 'red')
p # this does the actual plotting
```

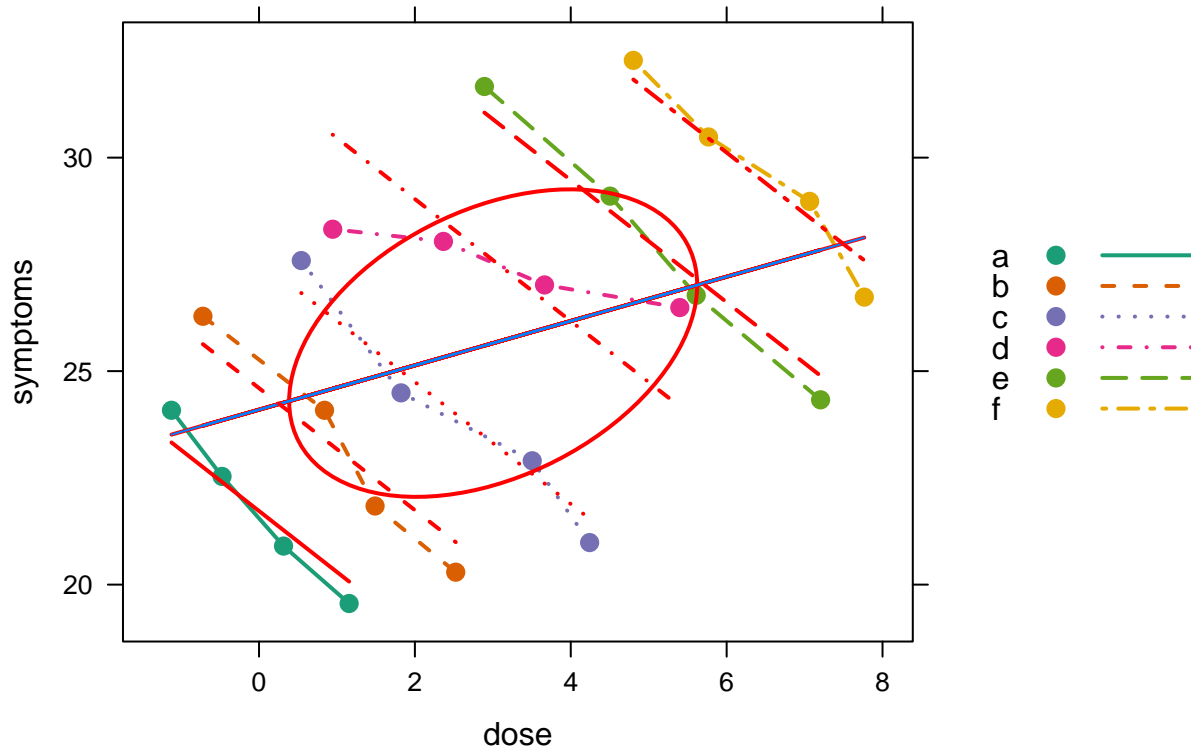


With conditional additive model

```
fit.additive <- lm(symptoms ~ dose + id, dd)
dd$fit.additive <- predict(fit.additive)
head(dd)
```

```
##   id      dose symptoms fit.pooled fit.additive
## 1  a -1.1233197 24.08335  23.51356  23.32960
## 2  a -0.4745763 22.53413  23.85034  22.40243
## 3  a  0.3144360 20.90336  24.25994  21.27480
## 4  a  1.1560491 19.55798  24.69684  20.07199
## 5  b -0.7200468 26.28488  23.72291  25.63114
## 6  b  0.8404442 24.08335  24.53300  23.40092
```

```
p2 <- p +
  xyplot(fit.pooled ~ dose, dd, type = 'l') +
  xyplot(fit.additive ~ dose, dd, groups = id, type = 'l', col = 'red', lwd = 2)
p2
```



Between model

- Note use of 'capply' for 'contextual apply'

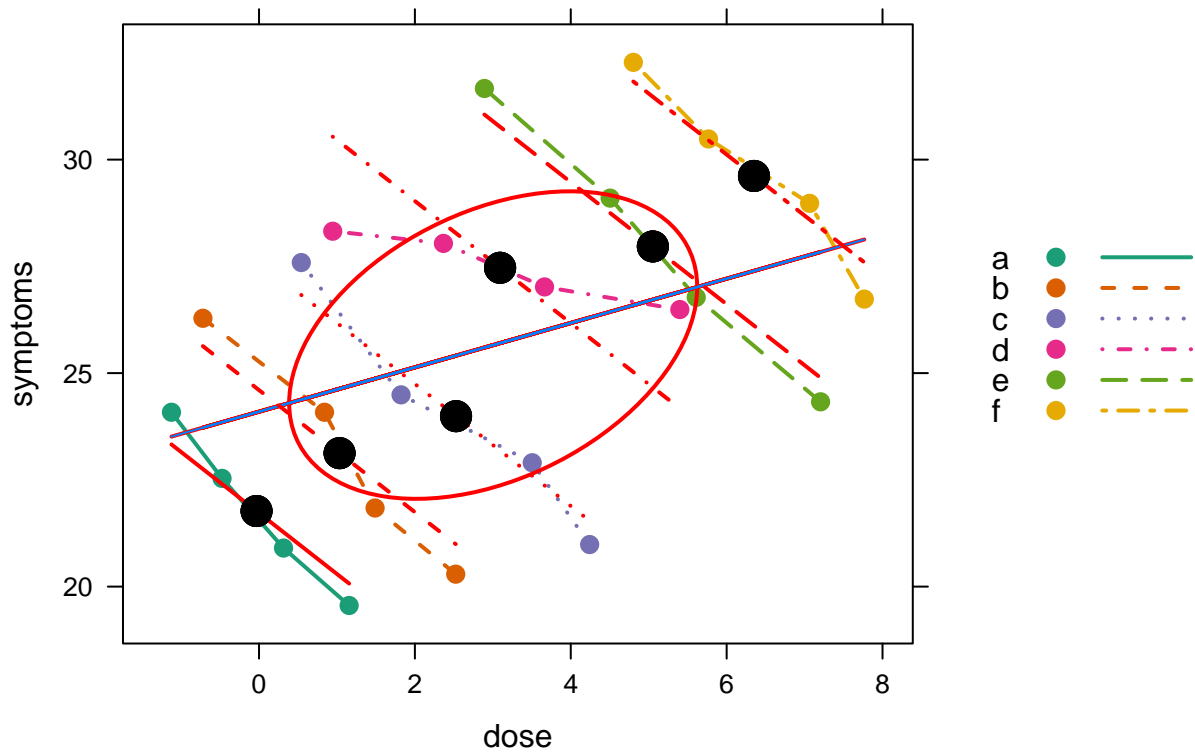
```
dd <- within(
  dd,
  {
    dose_m <- capply(dose, id, mean)
    symptoms_m <- capply(symptoms, id, mean)
  }
)
head(dd)
```

```
##   id      dose symptoms fit.pooled fit.additive symptoms_m      dose_m
## 1  a -1.1233197 24.08335  23.51356   23.32960  21.76971 -0.03185272
## 2  a -0.4745763 22.53413  23.85034   22.40243  21.76971 -0.03185272
## 3  a  0.3144360 20.90336  24.25994   21.27480  21.76971 -0.03185272
## 4  a  1.1560491 19.55798  24.69684   20.07199  21.76971 -0.03185272
## 5  b -0.7200468 26.28488  23.72291   25.63114  23.12528  1.03331385
## 6  b  0.8404442 24.08335  24.53300   23.40092  23.12528  1.03331385
```

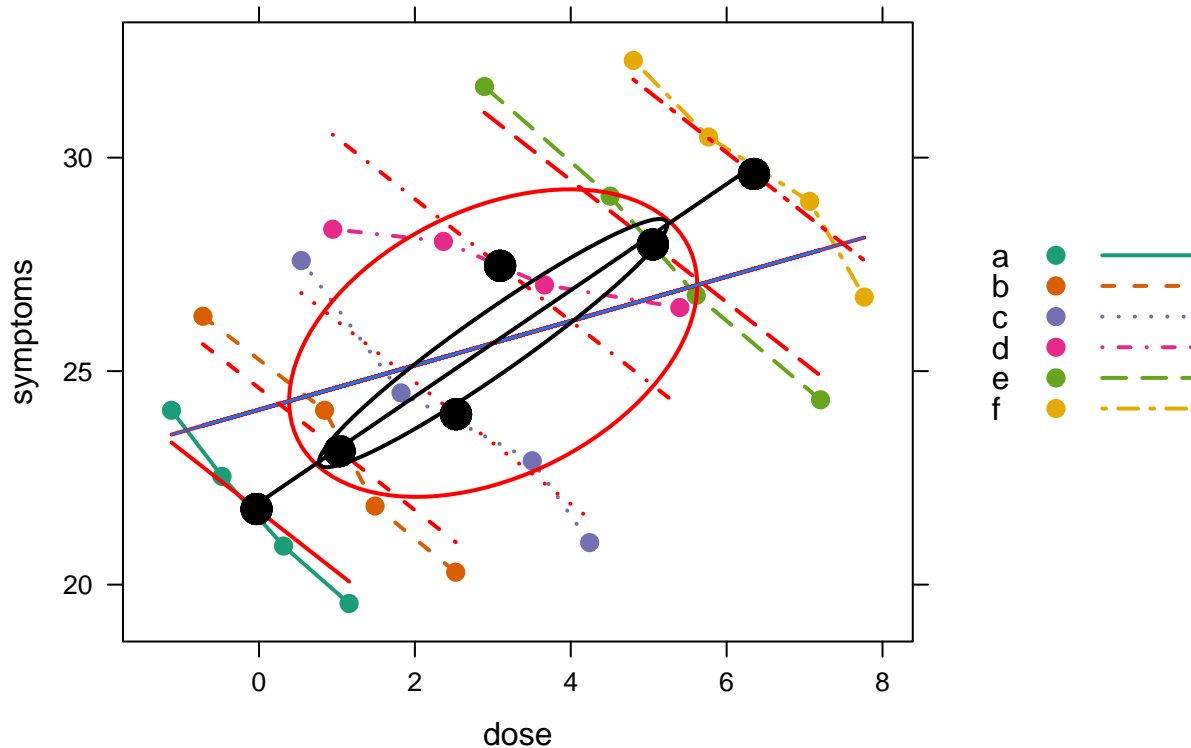
```
fit.between <- lm(symptoms_m ~ dose_m, dd) # note that each point is repeated
# as many times as there are
# observations in each group
```

```
dd$fit.between <- predict(fit.between)
```

```
p3 <- p2 +
  xyplot(symptoms_m ~ dose_m, dd, type = 'p', cex = 2, col = 'black', pch = 16)
```



```
(
  p4 <- p3 +
  (xyplot(symptoms_m ~ dose_m, dd, type = 'p', cex = 2, col = 'black', pch = 16) +
  layer(panel.dell(..., col = 'black', lwd = 2))) +
  xyplot(fit.between ~ dose_m, dd, type = 'l', col = 'black', lwd = 2)
)
```



Explore conditional models:

```
fit.conditional <- lm(symptoms ~ id * dose, dd)
summary(fit.conditional) # can you interpret each parameter by showing
```

```
##
## Call:
## lm(formula = symptoms ~ id * dose, data = dd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6621 -0.1809 -0.0358  0.1572  0.6052
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.70650   0.21713  99.971 < 2e-16 ***
## idb           3.38322   0.36127   9.365 7.24e-07 ***
## idc           6.44577   0.48850  13.195 1.67e-08 ***
## idd           7.12785   0.51122  13.943 8.94e-09 ***
## ide          14.98226   0.76110  19.685 1.68e-10 ***
## idf          18.99750   1.24135  15.304 3.09e-09 ***
## dose        -1.98436   0.25402  -7.812 4.79e-06 ***
## idb:dose     0.08325   0.31387   0.265 0.795336
## idc:dose     0.33857   0.29517   1.147 0.273725
## idd:dose     1.54241   0.28634   5.387 0.000164 ***
## ide:dose     0.25819   0.28901   0.893 0.389236
## idf:dose     0.23859   0.31686   0.753 0.465974
```



```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.434 on 12 degrees of freedom
## Multiple R-squared:  0.9924, Adjusted R-squared:  0.9855
## F-statistic: 142.9 on 11 and 12 DF,  p-value: 5.789e-11
```

where it goes on the plot below

Same model, different parametrization

```
fit.conditional2 <- lm(symptoms ~ id / dose - 1, dd)
summary(fit.conditional2) # can you interpret each parameter by showing
```

```
##
## Call:
## lm(formula = symptoms ~ id/dose - 1, data = dd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6621 -0.1809 -0.0358  0.1572  0.6052
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## ida             21.7065     0.2171  99.971 < 2e-16 ***
## idb             25.0897     0.2887  86.894 < 2e-16 ***
## idc             28.1523     0.4376  64.334 < 2e-16 ***
## idd             28.8343     0.4628  62.302 < 2e-16 ***
## ide             36.6888     0.7295  50.295 2.50e-15 ***
## idf             40.7040     1.2222  33.303 3.41e-13 ***
## ida:dose       -1.9844     0.2540  -7.812 4.79e-06 ***
## idb:dose       -1.9011     0.1844 -10.312 2.57e-07 ***
## idc:dose       -1.6458     0.1503 -10.949 1.33e-07 ***
## idd:dose       -0.4419     0.1321  -3.344 0.00584 **
## ide:dose       -1.7262     0.1378 -12.524 3.00e-08 ***
## idf:dose       -1.7458     0.1894  -9.217 8.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.434 on 12 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9997
## F-statistic: 7122 on 12 and 12 DF,  p-value: < 2.2e-16
```

where it goes on the plot below

Adding the conditional fits

```
dd$fit.conditional <- predict(fit.conditional)
(
  p5 <- p4 +
  xyplot(fit.conditional ~ dose, data = dd, groups = id, type = 'l', col = 'blue', lty = 1)
)
```

