Plotting random effects

John Poulsen

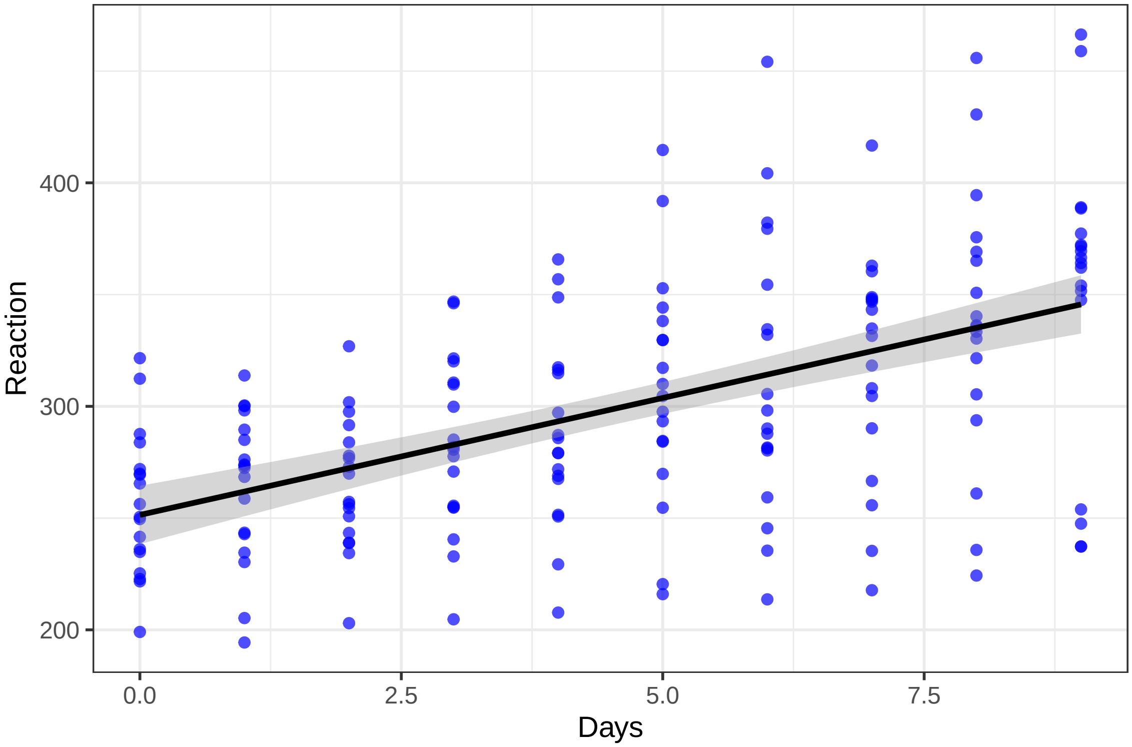
11/11/2020

Plotting random effects

This is a barebones demonstration of plotting random effects from multilevel models. I use the `sleepstudy` dataset from the `lme4` package.

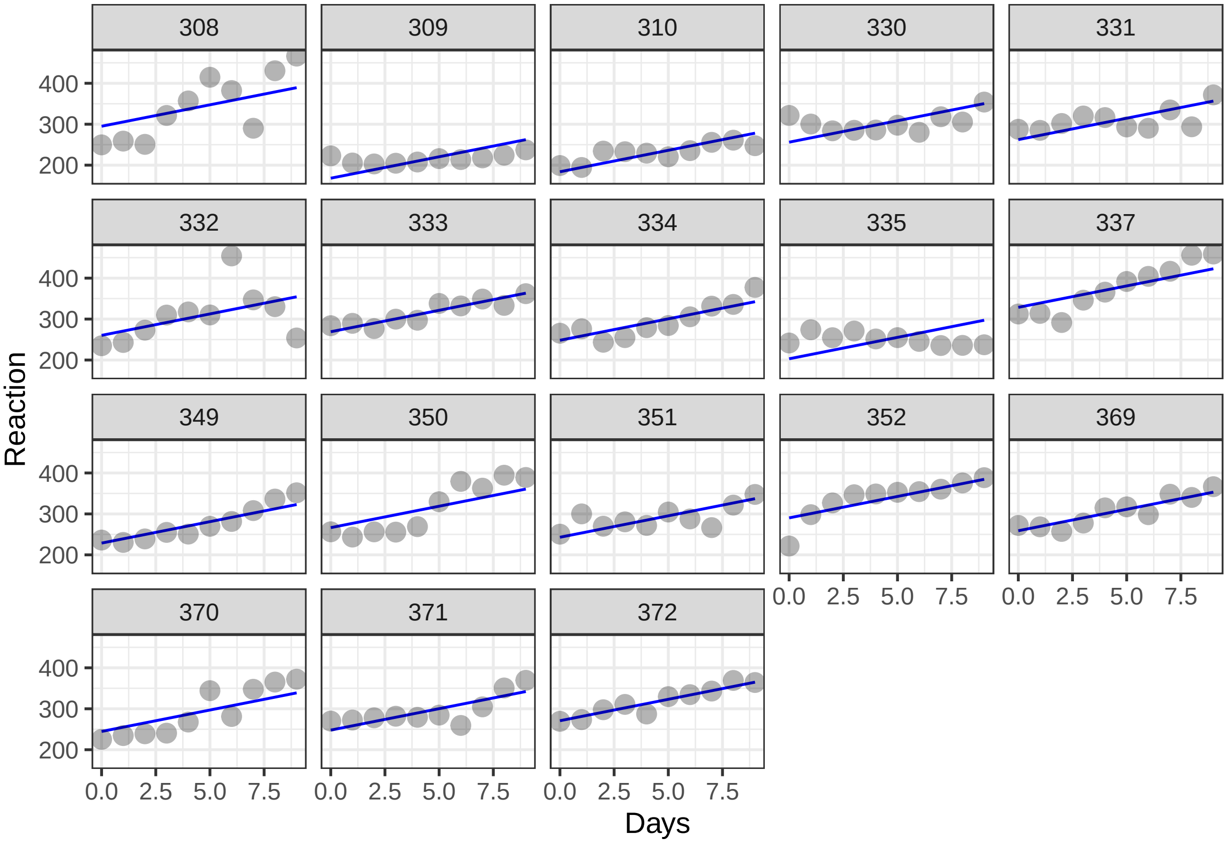
```
sleepstudy <- sleepstudy
 mod1 <- lm(Reaction ~ Days, sleepstudy)

gg <- ggplot(sleepstudy, aes(x = Days, y = Reaction)) +
       geom_point(color = "blue", alpha = 0.7) +
        geom_smooth(method = "lm", color = "black") +
       theme_bw()
gg
```

```
## `geom_smooth()` using formula 'y ~ x'
```

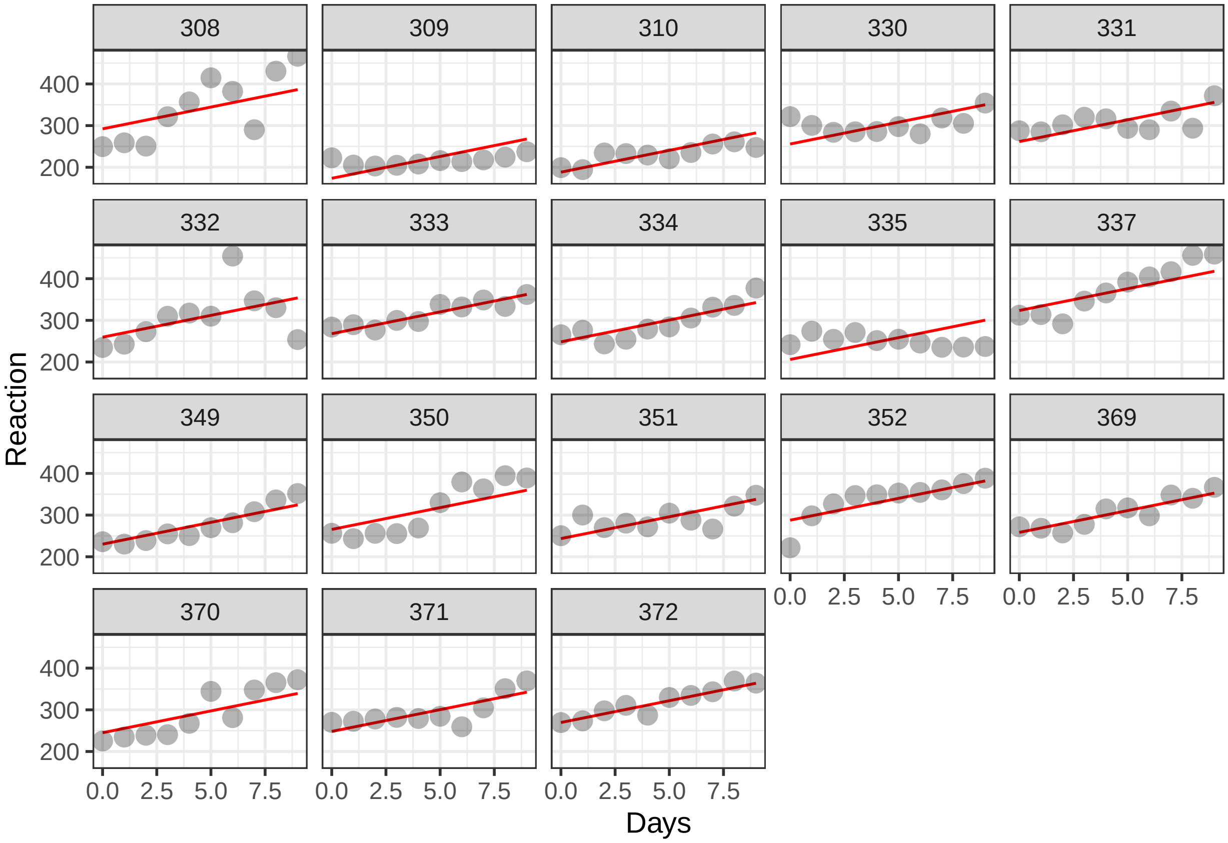


```
mod2 <- lm(Reaction ~ Days + Subject, sleepstudy)
 sleepstudy$UP <- fitted(mod2)
```

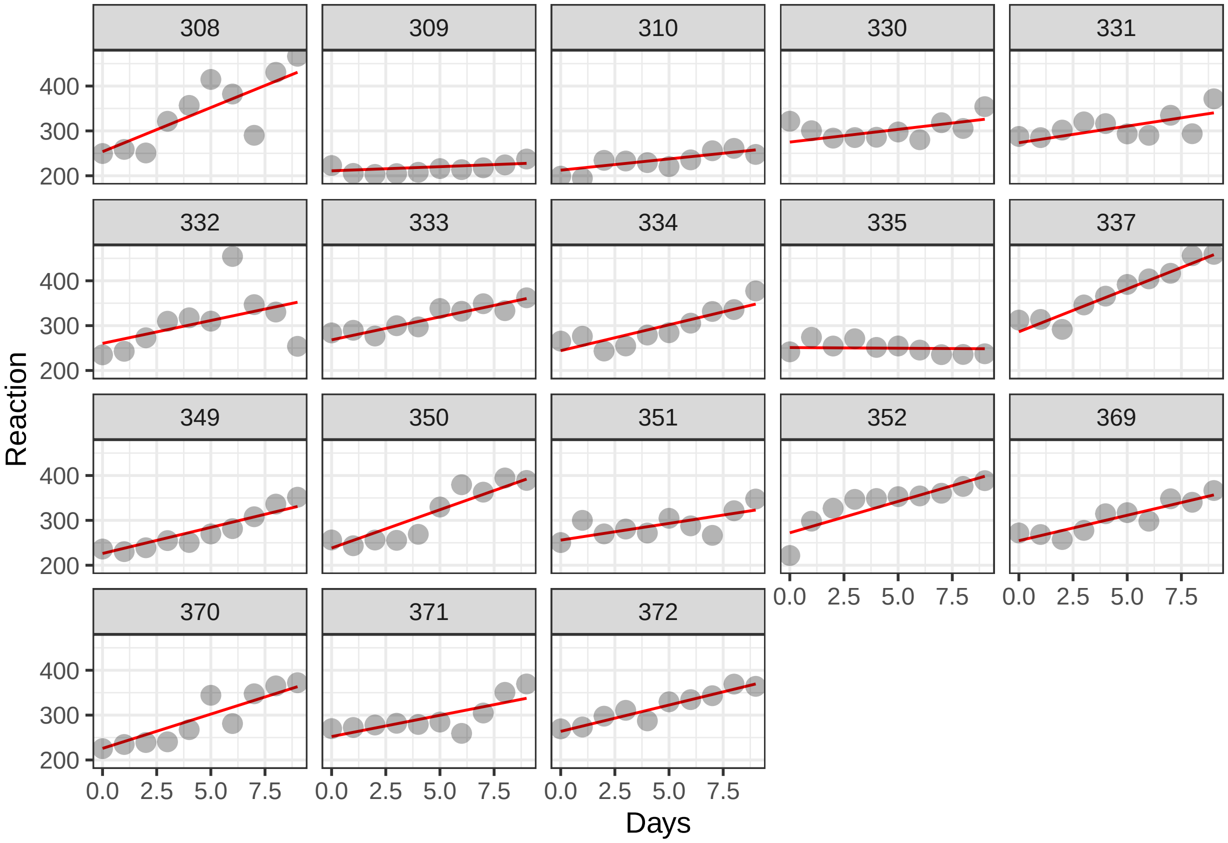
```
gg2 <- ggplot(sleepstudy, aes(x = Days, y = Reaction,
                              group = Subject)) +
      geom_line(aes(y = UP), color = "blue") +
      geom_point(alpha = 0.3, size = 3) +
      facet_wrap(~Subject) +
      theme_bw()
gg2
```



```
mod3 <- lmer(Reaction ~ Days + (1|Subject), sleepstudy)
 sleepstudy$VI <- fitted(mod3)

gg3 <- ggplot(sleepstudy, aes(x = Days, y = Reaction,
                              group = Subject)) +
      geom_line(aes(y = VI), color = "red") +
      geom_point(alpha = 0.3, size = 3) +
      facet_wrap(~Subject) +
      theme_bw()

gg3
```



```
mod4 <- lmer(Reaction ~ Days + (1 + Days|Subject), sleepstudy)
 sleepstudy$VIS <- fitted(mod4)

gg4 <- ggplot(sleepstudy, aes(x = Days, y = Reaction,
                              group = Subject)) +
      geom_line(aes(y = VIS), color = "red") +
      geom_point(alpha = 0.3, size = 3) +
      facet_wrap(~Subject) +
      theme_bw()

gg4
```

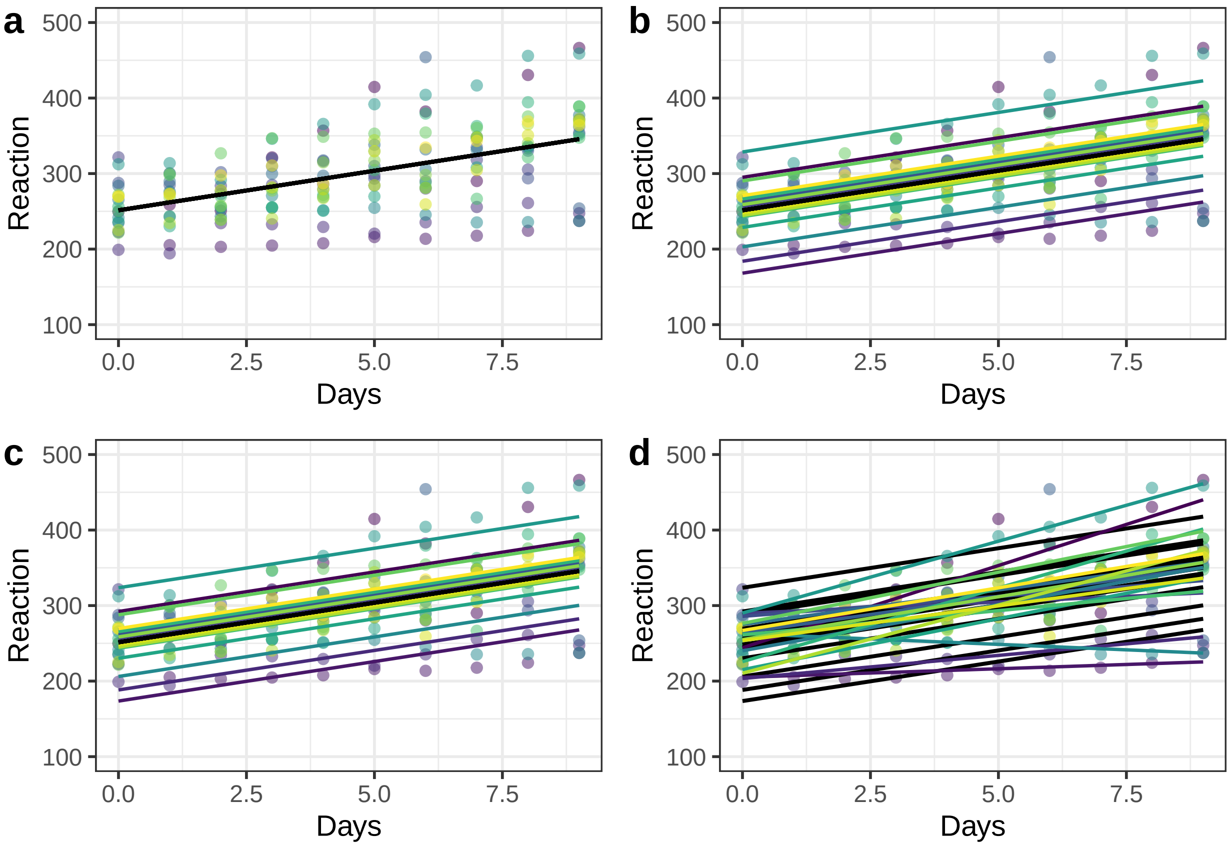


Now that the models have been run, plot the results of each model type.

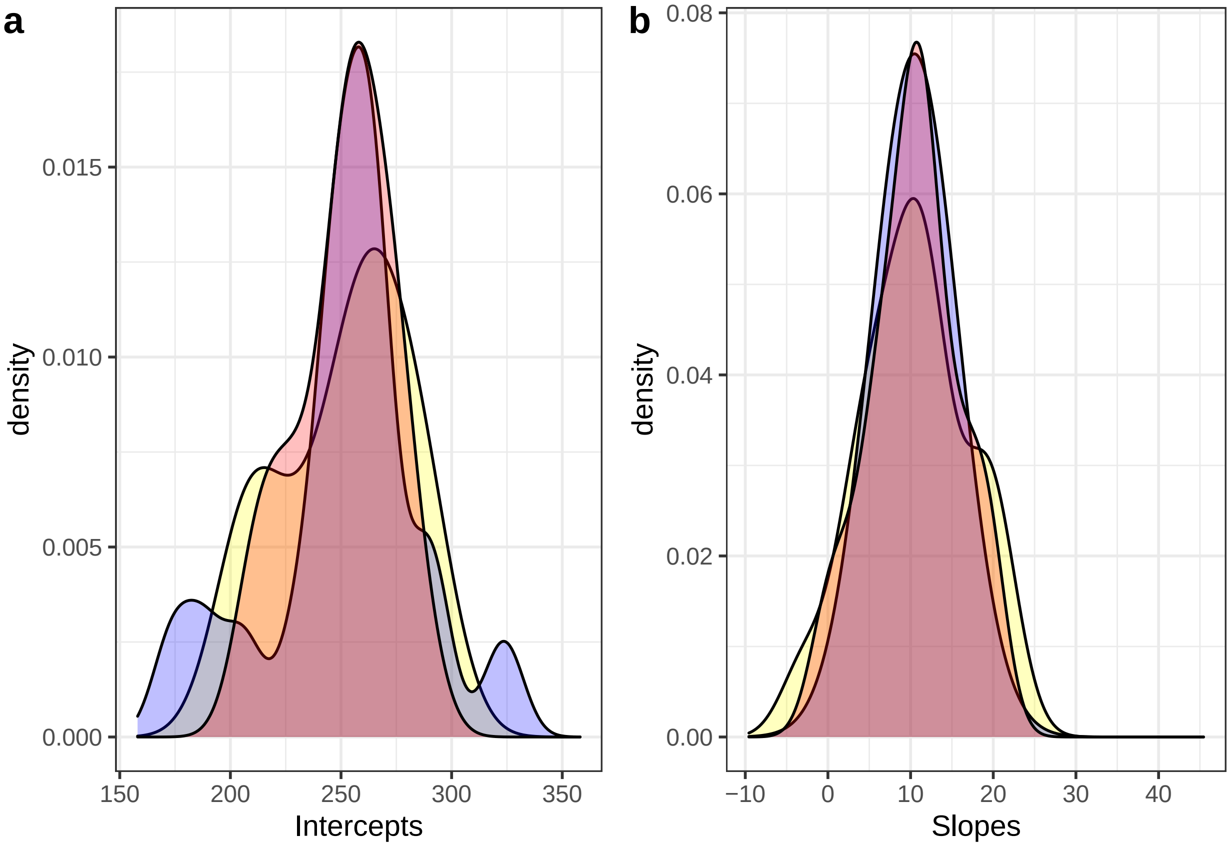
```
df <- sleepstudy

df$pred.VI <- predict(mod4)          # varying intercept model, prediction
df$pred.VS <- predict(mod3)          # varying slope model, prediction
df$pred.Pool <- predict(mod1)        # pooled model, prediction
df$pred.np <- predict(mod2)


 # assemble parameters
parms <- data.frame(mod3.re = ranef(mod3)$Subject)  # varying intercept model, re
 names(parms) <- "mod3.re"
parms$mod3.int <- fixef(mod3)[1]      # varying int. model, intercept
parms$mod3.slope <- fixef(mod3)[2]    # varying int. model, slope
slint <- data.frame(ranef(mod4)$Subject)   # varying slope model, random int & slope
slint$int <- fixef(mod4)[1]
slint$slope <- fixef(mod4)[2]
 colnames(slint) <- c("mod4.rint", "mod4.rslope", "mod4.int", "mod4.slope")
parms <- data.frame(cbind(parms, slint))

n <- length(unique(sleepstudy$Subject))
subs <- unique(sleepstudy$Subject)
np.mod <- data.frame()
   for (i in 1:n){
        npm <- lm(Reaction ~ Days, data = sleepstudy[sleepstudy$Subject == subs[i],])
         np.mod <- rbind(np.mod, coef(npm))
```

```r
    }
   names(np.mod) <- c("Int", "Slope")

parms$nopool.int <- np.mod$Int            # no pooled model, intercept
 parms$nopool.slope <- np.mod$Slope

 # varying intercept model
 slp1 <- ggplot(data = df, aes(x = Days, y = Reaction, group = Subject,
                               color = factor(Subject))) +
               geom_point(alpha = 0.5) +
                geom_line(aes(y = VI), size=0.6) +
                 geom_line(aes(y = pred.Pool), size = 0.7, color = "black") +
                 theme_bw() + scale_color_viridis_d() +
                      ylim(100, 500) +
                theme(legend.position = 'none')

 # no pooling model
 slp2 <- ggplot(data = df, aes(x = Days, y = Reaction, group = Subject,
                               color = factor(Subject))) +
               geom_point(alpha = 0.5) +
                geom_line(aes(y = pred.np), size=0.6) +
                 geom_line(aes(y = pred.Pool), size = 0.7, color = "black") +
                 theme_bw() + scale_color_viridis_d() +
                      ylim(100, 500) +
                theme(legend.position = 'none')


  # varying slope, varying intercept
  slp3 <- ggplot(data = df, aes(x = Days, y = Reaction, group = Subject,
                               color = factor(Subject))) +
                geom_point(alpha = 0.5) +
                 geom_line(aes(y = pred.VS), size = 0.7, color = "black") +
                  geom_smooth(method="lm", se = F, size = 0.6) +
                  theme_bw() + scale_color_viridis_d() +
                       ylim(100, 500) +
                 theme(legend.position = 'none')

  # pooled model
  slp4 <- ggplot(data = df, aes(x = Days, y = Reaction, group = Subject,
                               color = factor(Subject))) +
                geom_point(alpha = 0.5) +
                 geom_line(aes(y = pred.Pool), size = 0.7, color = "black") +
                 theme_bw() + scale_color_viridis_d() +
                   ylim(100, 500) +
                theme(legend.position = 'none')

  plot_grid(slp4, slp2, slp1, slp3, ncol = 2, labels = c("a", "b", "c", "d"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
# density of intercepts
xrange <- c(median(parms$nopool.int) - 100,
            median(parms$nopool.int) + 100)

dd1 <- ggplot(parms, aes(x = nopool.int)) +
        geom_density(fill = "yellow", alpha = .25) +
         geom_density(aes(mod3.int + mod3.re), fill = "blue", alpha = 0.25) +
        geom_density(aes(mod4.int + mod4.rint), fill = "red", alpha = 0.25) +
             xlab("Intercepts") +  xlim(xrange[1], xrange[2]) +
       theme_bw()


# density of slopes
xxrange <- c(median(parms$nopool.slope) - 20,
            median(parms$nopool.slope) + 35)
dd2 <- ggplot(parms, aes(x = nopool.slope) ) +
        geom_density(fill = "yellow", alpha = .25) +
          geom_density(aes(mod3.slope), fill = "blue", alpha = .25) +
         geom_density(aes(mod4.slope + mod4.rslope), fill = "red", alpha = .25) +
        xlim(xxrange[1], xxrange[2]) + xlab("Slopes") + theme_bw()

plot_grid(dd1, dd2, ncol = 2, labels = c("a", "b"))
```


a
density
0.000
0.005
0.010
0.015
150
200
250
300
350
Intercepts
b
density
0.00
0.02
0.04
0.06
0.08
−10
0
10
20
30
40
Slopes