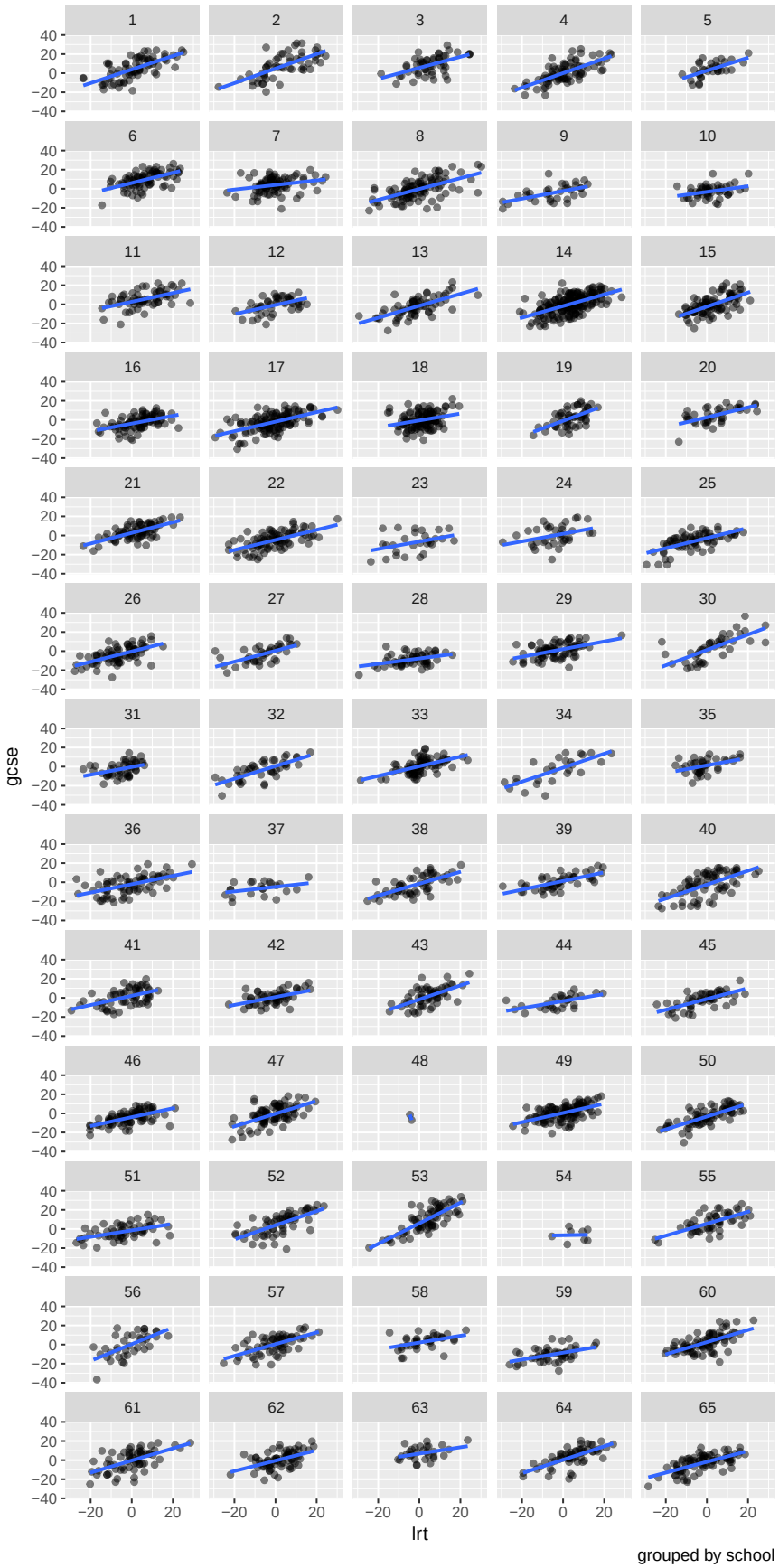
Multilevel Modeling ggplot2 Examples

Jake Thompson

2017-03-01

```
library(haven)
gcse <- read_dta("data/gcse.dta")
```

```
library(ggplot2)

ggplot(data = gcse, mapping = aes(x = lrt, y = gcse)) +
  facet_wrap(~ school, ncol = 5) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(caption = "grouped by school")
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
gcse
lrt
grouped by school

Time series plots

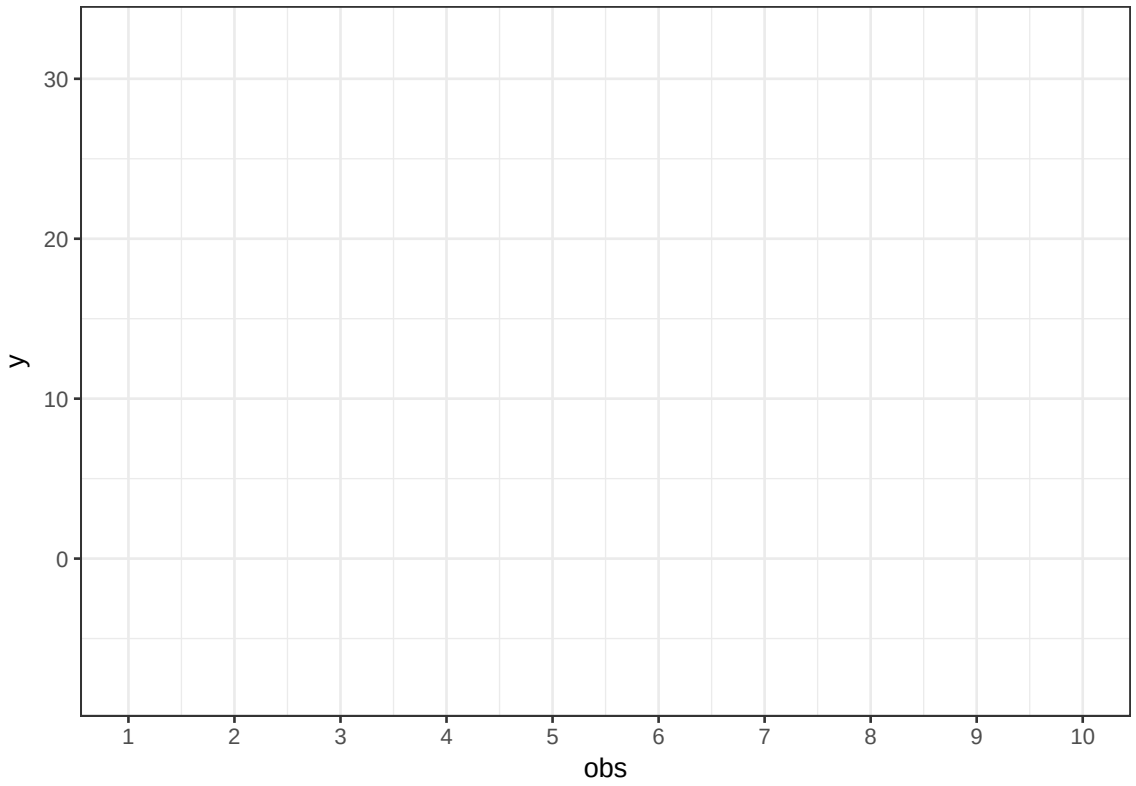
First we generate some longitudinal data (adapted from here).

```
library(MASS)
library(nlme)
library(purrr)

### set number of individuals
n <- 200

### average intercept and slope
beta0 <- 1.0
beta1 <- 6.0

### true autocorrelation
ar.val <- .4

### true error SD, intercept SD, slope SD, and intercept-slope cor
sigma <- 1.5
tau0  <- 2.5
tau1  <- 2.0
tau01 <- 0.3

### maximum number of possible observations
m <- 10

### simulate number of observations for each individual
p <- round(runif(n, 4, m))

### set up data frame
dat <- map2_df(.x = 1:n, .y = p, .f = function(x, y, m) {
  data.frame(
    id = rep(x, times = y),
    group = factor(rbinom(n = 1, size = 3, prob = 0.5), levels = 0:3),
    obs = c(1, sort(sample(2:m, y - 1, replace = FALSE)))
  )
}, m = m)

### simulate (correlated) random effects for intercepts and slopes
mu  <- c(0, 0)
S   <- matrix(c(1, tau01, tau01, 1), nrow = 2)
tau <- c(tau0, tau1)
S   <- diag(tau) %*% S %*% diag(tau)
U   <- mvrnorm(n, mu=mu, Sigma=S)

### simulate AR(1) errors and then the actual outcomes
dat$e <- unlist(sapply(p, function(x) arima.sim(model=list(ar=ar.val), n=x) * sqrt(1-ar.val^2) * sigma))
dat$y <- (beta0 + rep(U[,1], times = p)) + (beta1 + rep(U[, 2], times = p)) * log(dat$obs) + dat$e
```

First we create a blank template plot with some formatting

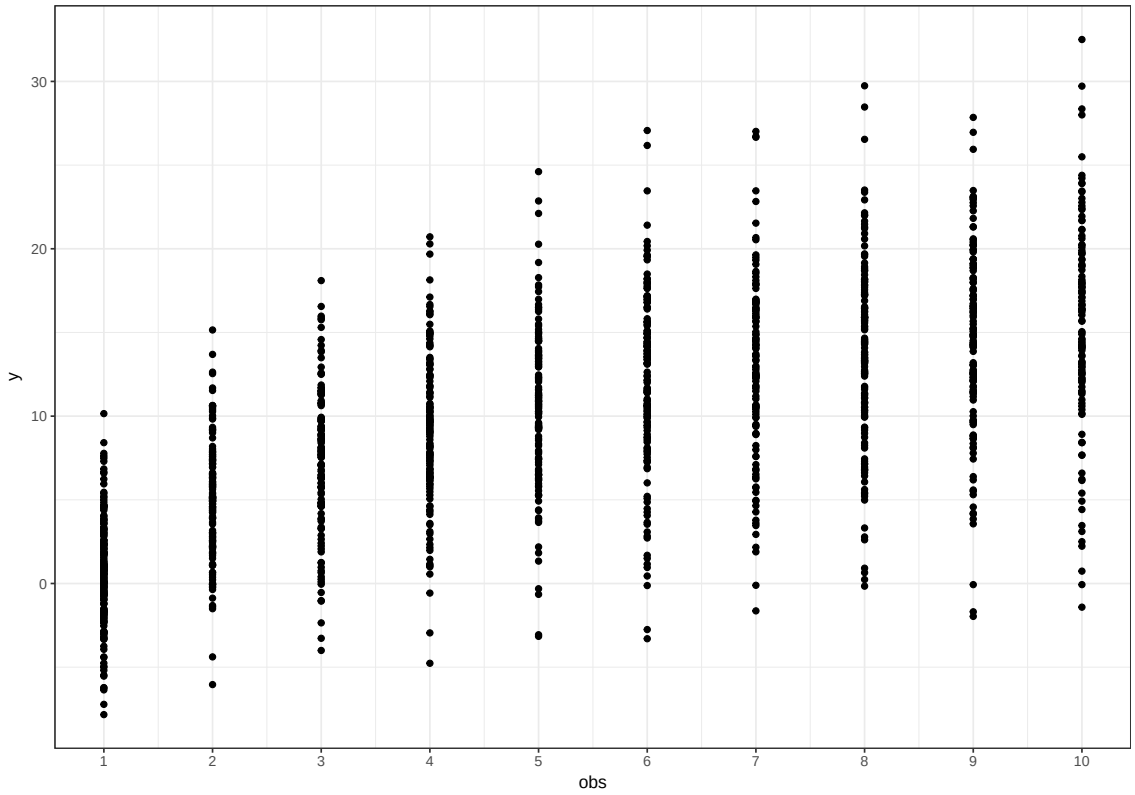
```
p <- ggplot(data = dat, mapping = aes(x = obs, y = y)) +
  scale_x_continuous(limits = c(1, 10), breaks = seq(1, 10, 1)) +
  theme_bw()
```

```
p
```



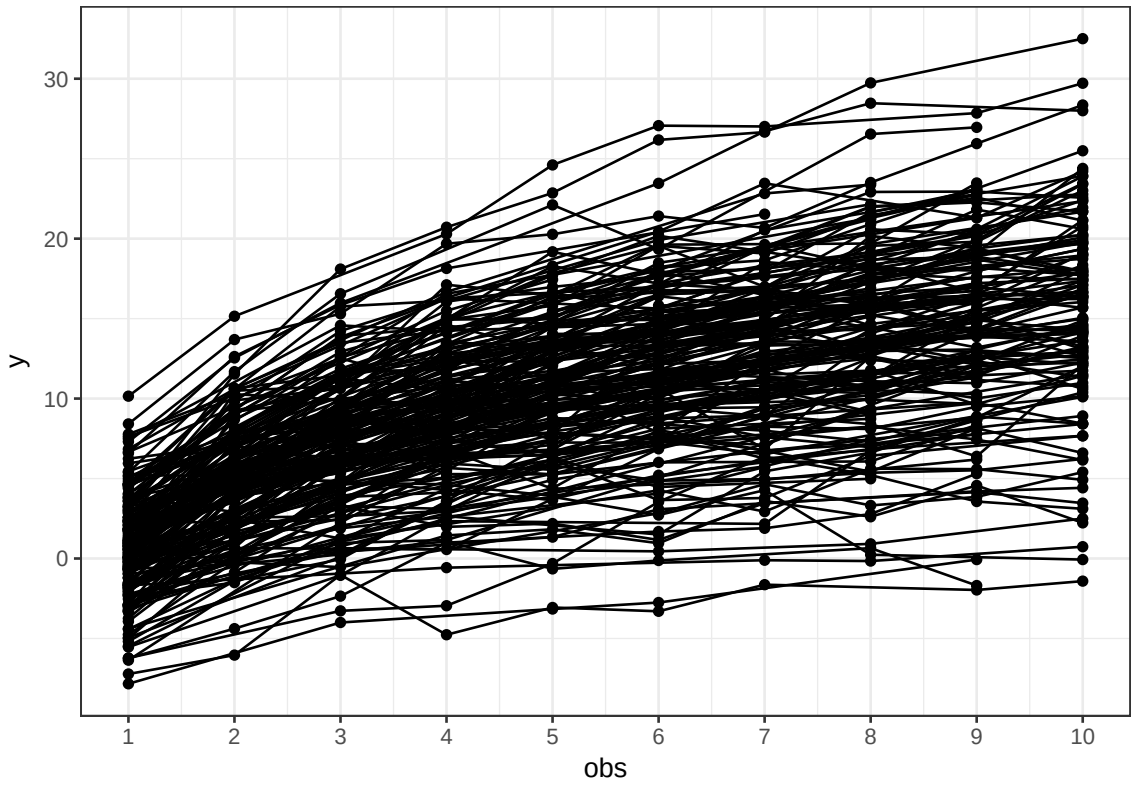
Now, we create our initial scatter plot.

```
p + geom_point()
```



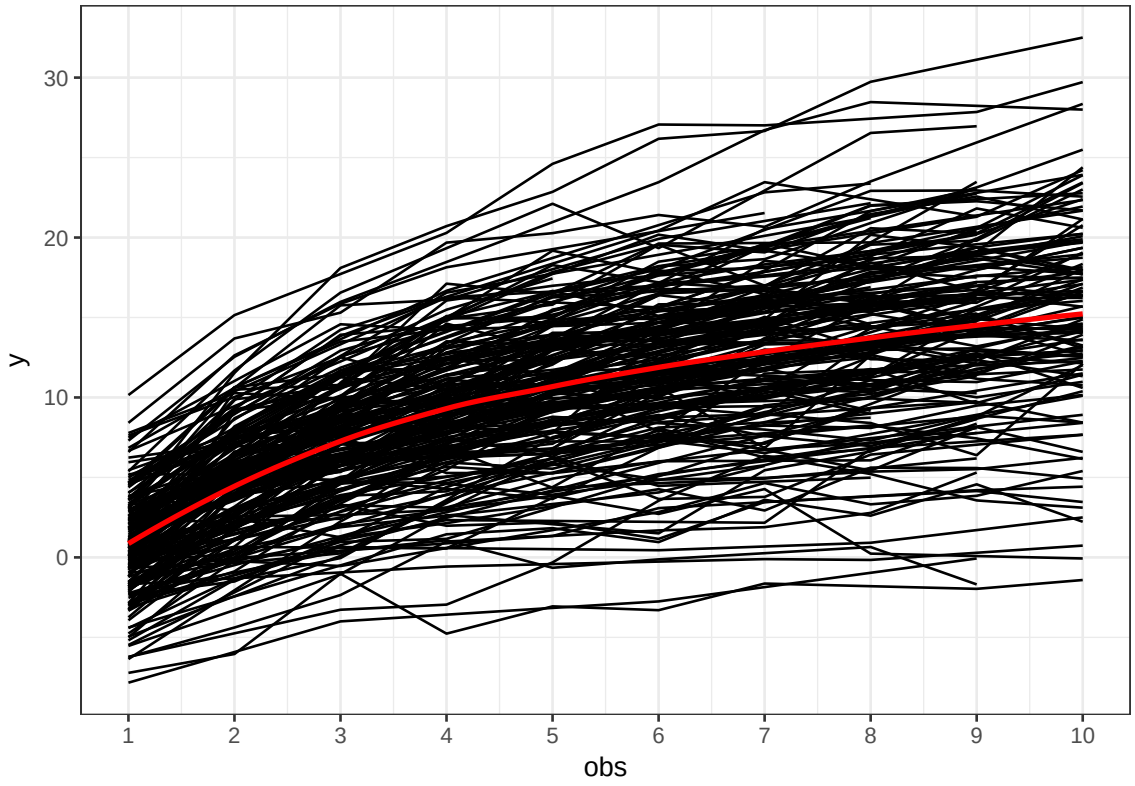
Draw a line for each individual by assigning group to the person identifier.

```
p + geom_point() +
  geom_line(aes(group = id))
```



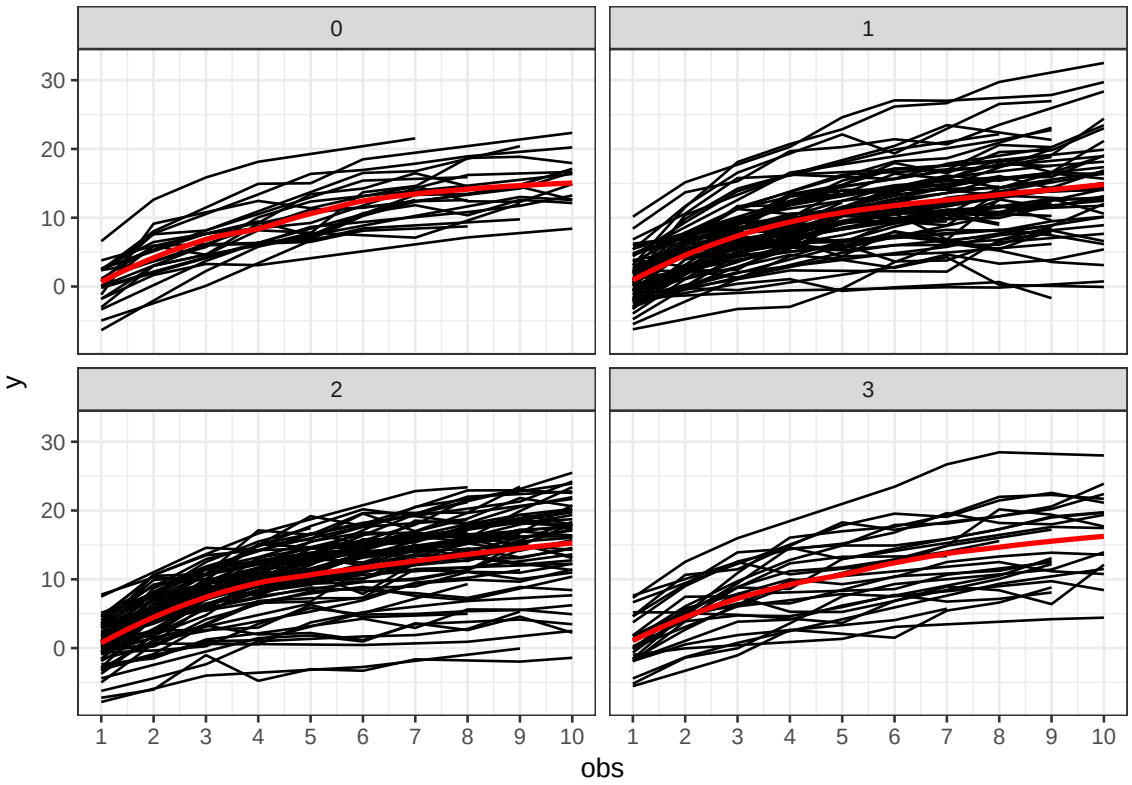
Remove the points and add an overall loess trend line.

```
p + geom_line(aes(group = id)) +
  geom_smooth(method = "loess", color = "red", se = FALSE)
```



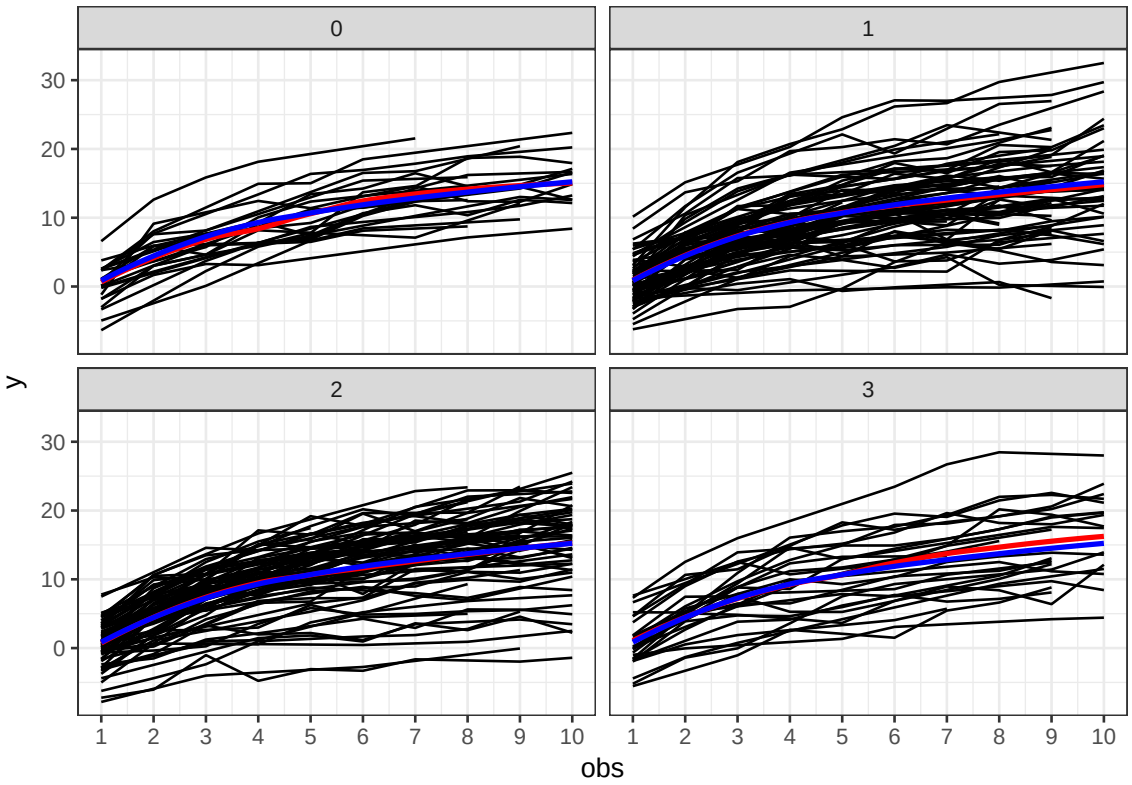
Split out by group, drawing a trend line for each group individually.

```
p + geom_line(aes(group = id)) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  facet_wrap(~ group)
```



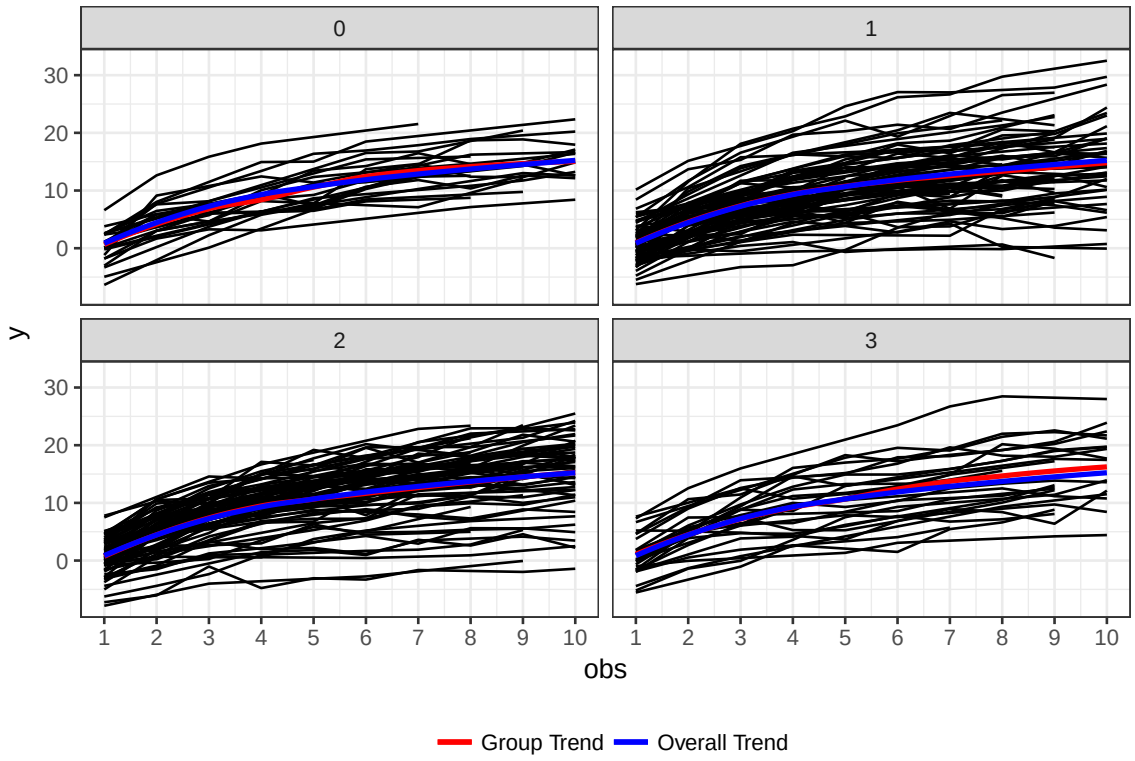
We can add the overall trend line by adding a second `geom_smooth` layer that uses the same data, but without the facetting variable.

```
library(dplyr)

p + geom_line(aes(group = id)) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  geom_smooth(data = select(dat, -group), method = "loess", color = "blue",
    se = FALSE) +
  facet_wrap(~ group)
```



Finally, add a color aesthetic so a legend will be created.

```
p + geom_line(aes(group = id)) +
  geom_smooth(aes(color = "Group Trend"), method = "loess", se = FALSE) +
  geom_smooth(data = select(dat, -group), aes(color = "Overall Trend"),
    method = "loess", se = FALSE) +
  facet_wrap(~ group) +
  scale_color_manual(name = NULL, values = c("red", "blue")) +
  theme(
    legend.position = "bottom"
  )
```


0
1
2
3
y
obs
Group Trend
Overall Trend