LMM vs T-Test(slopes)

Lukas Graz

2023-05-31

Contents

1 TLDR

Given a dataset as shown in the last plot, we consider the following analyses:
1. For each subject extract the slope and perform a simple t-test on the slopes
2. Fit `lmer(salery ~ slope + (1|subject))` and test if $\beta_{slope} = 0$
3. Fit `lmer(salery ~ slope + (slope|subject))` and test if $\beta_{slope} = 0$
Then: 1 & 3 are valid approaches and 2 is only valid if there is no individual slope

2 Simulate Data

Salary of subject s at time t:

$$salary_t^{(s)} = t \cdot (slope + d_s) + intercept_s + \varepsilon_{t,s}$$

- $d_s \sim \mathcal{N}(0, subjSlopeSD^2)$
- $intercept_s \sim \mathcal{N}(0, subjSD^2)$
- $\varepsilon_{t,s} \sim \mathcal{N}(0, obsSD^2)$

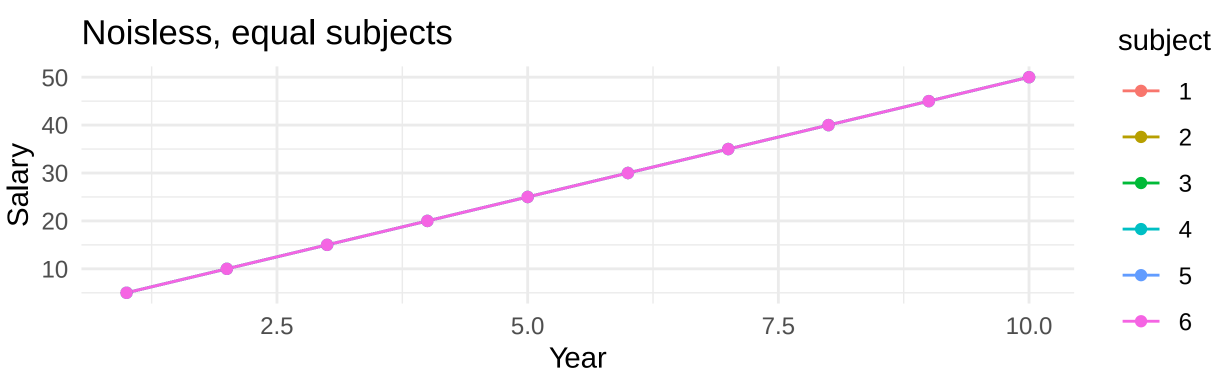
```
library(mcreplicate)
library(ggplot2)
library(lmerTest)

#' `nsub`: how many subjects (default: 6)
#' `nyears`: how many years(default: 10)
#' `obsSD`: standard deviation of noise (observation-level) (default: 15)
#' `subjSD`: standard deviation of individual effect (default: 4)
#' `slope`: shared increase of income per year (default: 5)
#' `subjSlopeSD`: subject specific standard deviation from slope (default: 2)
```

```
get_data <- function(nsub=6, nyears=10,
                     obsSD=15, subjSD=4,
                     slope=5, subjSlopeSD=2){
  subj_intercept <- rep(rnorm(nsub, 0, subjSD), each=nyears)
  subj_slope <- rep(rnorm(nsub, slope, subjSlopeSD), each=nyears)
  data.frame(
    subject = as.factor(rep(1:nsub, each = nyears)),
    year = rep(1:nyears, times = nsub),
    salary =  subj_intercept +  # subject effect
              subj_slope*(1:nyears) + # individual slope
              rnorm(nyears*nsub, 0, obsSD) # obsSD
  )
}
```

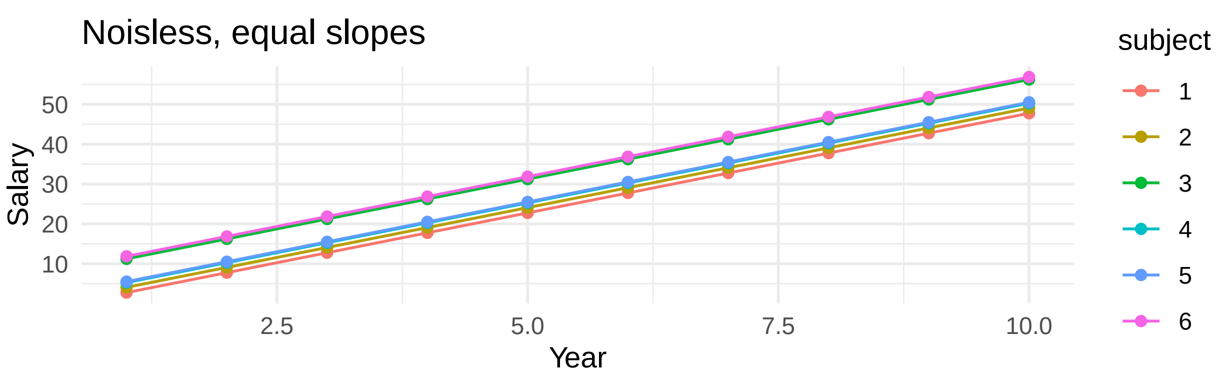
3 Plot Data

```
plot_data <- function(main, ...){
  ggplot(get_data(...), aes(x = year, y = salary, group = subject, color = subject)) +
    geom_line() +
    geom_point() +
    labs(x = "Year", y = "Salary") +
    theme_minimal() +
    ggtitle(main)
}
```

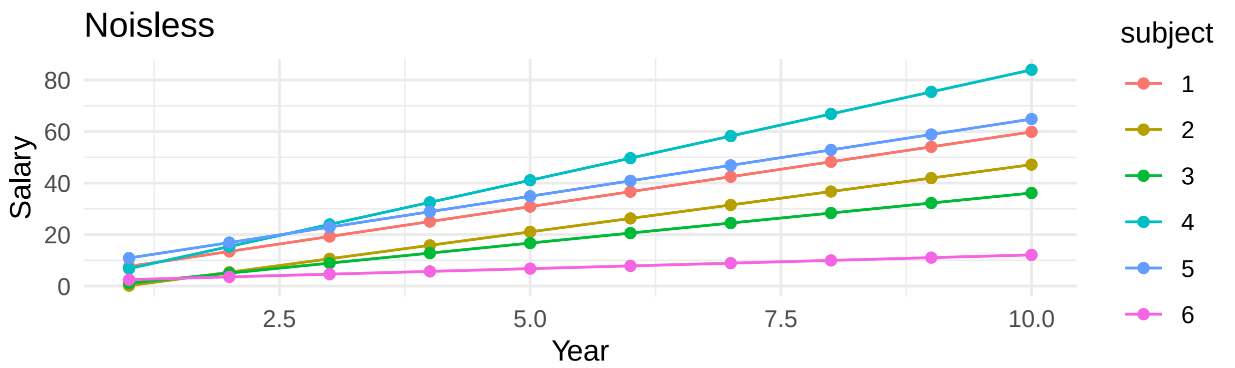
```
plot_data("Noisless, equal subjects", obsSD=0, subjSD=0, subjSlopeSD=0)
```



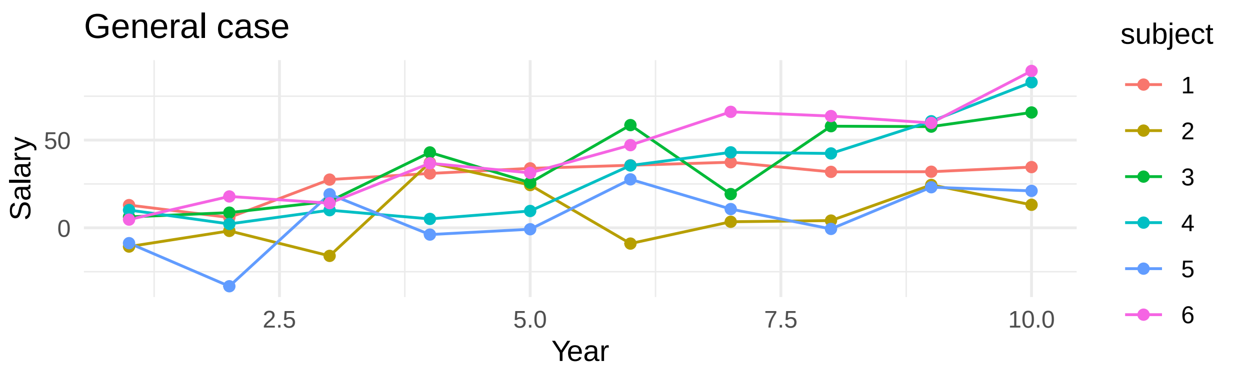

```
plot_data("Noisless, equal slopes", obsSD=0, subjSlopeSD=0)
```



```
plot_data("Noisless", obsSD=0)
```




```
plot_data("General case")
```



4 Analysis Methods

```
slopeTTest <- function(data){
  fits <- lmList(salary ~ year | subject, data)
  slopes <- coef(fits)[,"year"]
  t_test <- t.test(slopes, mu = 0)
  t_test$p.value
}

lmmRandItcpt <- function(data){
  lmm <- lmer(salary ~ year + (1|subject), data = data)
    summary(lmm)$coefficients[2, "Pr(>|t|)"]
}

lmmRandSlope <- function(data){
  lmm <- lmer(salary ~ year + (year|subject), data = data)
    summary(lmm)$coefficients[2, "Pr(>|t|)"]
}

p_values <- function(...){
  data <- get_data(...)
  c(
    slopeTTest = slopeTTest(data),
    lmmRandItcpt = lmmRandItcpt(data),
    lmmRandSlope = lmmRandSlope(data)
```

```
  ) # if you cange the amount of arguments, change also the object "Power"
}

get_power <- function(...){
  args <- list(...)
  PVALS <- as.data.frame(t(
  mc_replicate(50000, do.call(p_values, args))
  ))
  # POWER:
  sapply(lapply(PVALS, function(x) x<0.05), mean)
}
```

5 Power Calculations

```
set.seed(123)
# - `nsub`: how many subjects (default: 6)
# - `nyears`: how many years(default: 10)
# - `obsSD`: standard deviation of noise (observation-level) (default: 15)
# - `subjSD`: standard deviation of individual effect (default: 4)
# - `slope`: shared increase of income per year (default: 5)
# - `subjSlopeSD`: subject specific standard deviation from slope (default: 2)


ARGS <- as.data.frame(rbind(
  # NULL:
  expand.grid(
    nsub=6,
    nyears=10,
    obsSD=5,
    subjSD=4,
    slope=0,
    subjSlopeSD=c(0,4)),
  # Change standard deviations and effects:
  expand.grid(
    nsub=6,
    nyears=10,
    obsSD=c(5,15),
    subjSD=c(4, 12),
    slope=c(2,4),
    subjSlopeSD=c(1,4)),
  # Change samplesize (allocation):
  expand.grid(
    nsub=c(6,10,20),
    nyears=c(10,20),
    obsSD=5,
    subjSD=4,
    slope=2,
    subjSlopeSD=1)[-1,]
))
rownames(ARGS) <- NULL

Power <- matrix(NA, nrow=nrow(ARGS), ncol=3)
```

```r
colnames(Power) <- c("slopeTTest",
  "lmmRandItcpt", "lmmRandSlope")

for (i in 1:nrow(ARGS)){
  args <- as.list(ARGS[i,])
  Power[i,] <- do.call(get_power, args)
}
```

6 Show Results:

```r
as.data.frame(cbind(ARGS, Power))
```

nsub	nyears	obsSD	subjSD	slope	subjSlopeSD	slopeTTest	lmmRandItcpt	lmmRandSlope
6	10	5	4	0	0	0.04888	0.04880	0.02142
6	10	5	4	0	4	0.05092	0.52992	0.05080
6	10	5	4	2	1	0.92300	0.99944	0.92158
6	10	15	4	2	1	0.53466	0.78824	0.51126
6	10	5	12	2	1	0.92602	0.99916	0.92568
6	10	15	12	2	1	0.53688	0.78708	0.51792
6	10	5	4	4	1	1.00000	1.00000	0.99996
6	10	15	4	4	1	0.97850	0.99954	0.97682
6	10	5	12	4	1	1.00000	1.00000	1.00000
6	10	15	12	4	1	0.97792	0.99958	0.97736
6	10	5	4	2	4	0.16922	0.74490	0.16850
6	10	15	4	2	4	0.15234	0.59184	0.14504
6	10	5	12	2	4	0.17018	0.74732	0.17012
6	10	15	12	2	4	0.15202	0.59228	0.14812
6	10	5	4	4	4	0.49952	0.96212	0.49918
6	10	15	4	4	4	0.44912	0.90606	0.43974
6	10	5	12	4	4	0.49566	0.96068	0.49564
6	10	15	12	4	4	0.44628	0.90346	0.44230
10	10	5	4	2	1	0.99844	1.00000	0.99842
20	10	5	4	2	1	1.00000	1.00000	1.00000
6	20	5	4	2	1	0.96588	0.99998	0.96574
10	20	5	4	2	1	0.99970	1.00000	0.99970
20	20	5	4	2	1	1.00000	1.00000	1.00000