Exercises day 8

Basic Statistics for health researchers 2021

22 November 2021

Exercise A: what to adjust on?

In the lecture it was mentioned that working on the change between baseline and follow-up provides a natural adjustment for certain but not all covariates. We will examplify this via the following example:

> Investigators are planning study where they want to assess the impact of a treatment against depression (SSRI) on the brain serotonergic system. They will include two groups (placebo and SSRI) with a baseline and a follow-up measurement a week after. At each timepoint, a PET scan is performed to quantify the availability of serotonin receptors in the brain, which involves the injection of a radioactive contrast agent to the patient. The investigators are planning to use the change in PET signal from baseline to assess the treatment effect.

Based on the existing literature, the PET signal can be influenced by:

- genetic polymorphisms (e.g. 5-HTTLPR)
- age (decline of 10% per decade)
- scanner type (binary variable, only 2 scanner types)
- radioactive dose in the contrast agent: this will typically vary from patient to patient and is measured before the patient undergo the scan.

We will assume that these variables have a linear relationship with the outcome.

1. Which variable are "naturally" adjusted for when computed the change score? How would you test the treatment effect if there were no other variables to control for?

2. How would you control for the other variables?
   What would be the benefit(s) of this adjustment?
   (consider the case of a randomized study and a non-randomized study)

3. In randomized experiment, adjusting for post-randomization variables is generally not recommended. Why?
   Is that problematic in this example?

Exercise B: analyzing a longitudinal study

In this exercise, we will reproduce the graphics and results presented during the lecture. The R code given at the end of the lecture notes (section 6) can be help to answer some of the questions. To load the data in **R** use [1]:

```
## requires the nlmeU package to be installed
data(armd.wide, package = "nlmeU")
```

The following code converts the data from the wide to the long format:

```
library(reshape2)
armd.long <- melt(armd.wide,
    measure.vars = paste0("visual",c(0,4,12,24,52)),
    id.var = c("subject","lesion","treat.f","miss.pat"),
    variable.name = "week",
    value.name = "visual")

armd.long$week <- factor(armd.long$week,
    level = paste0("visual",c(0,4,12,24,52)),
    labels = c(0,4,12,24,52))
```

Part 1: descriptive statistics

In this first part we will replicate the descriptive statistics presented during the lecture (slides 13-18).

1. we can display the dataset in the wide format using `str`. What is the meaning of the values in the columns `treat.f` and `miss.pat`?

```
str(armd.wide)
```

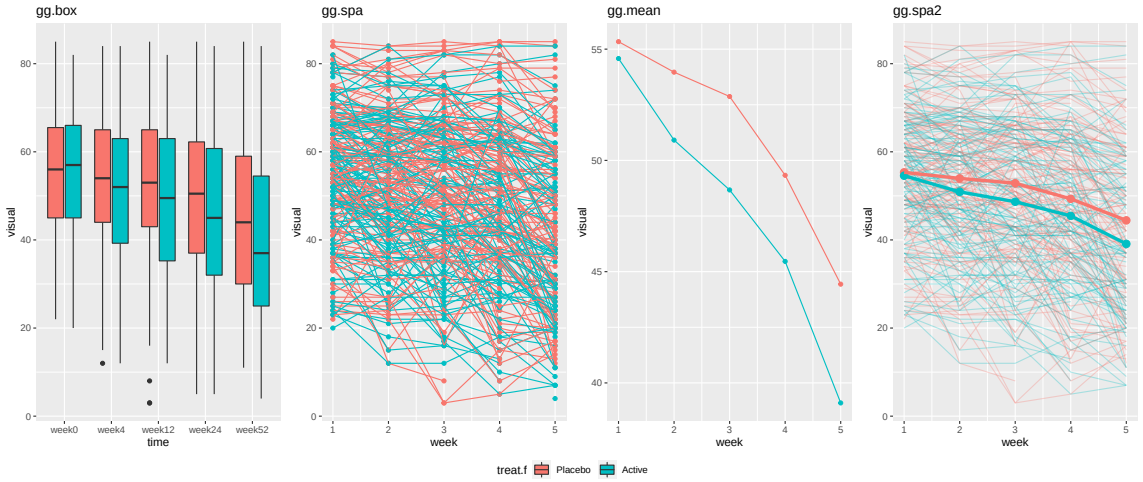
```
'data.frame':        240 obs. of  10 variables:
 $ subject : Factor w/ 240 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ lesion  : int  3 1 4 2 1 3 1 3 2 1 ...
 $ line0   : int  12 13 8 13 14 12 13 8 12 10 ...
 $ visual0 : int  59 65 40 67 70 59 64 39 59 49 ...
 $ visual4 : int  55 70 40 64 NA 53 68 37 58 51 ...
 $ visual12: int  45 65 37 64 NA 52 74 43 49 71 ...
 $ visual24: int  NA 65 17 64 NA 53 72 37 54 71 ...
 $ visual52: int  NA 55 NA 68 NA 42 65 37 58 NA ...
 $ treat.f : Factor w/ 2 levels "Placebo","Active": 2 2 1 1 2 2 1 1 2 1 ...
 $ miss.pat: Factor w/ 9 levels "----","---X",..: 4 1 2 1 9 1 1 1 1 2 ...
```

[1] for non R users, the file `armd.txt` available on the course webpage contains the data

2. Compute summary statistics (mean, variance, correlation) of the dataset over time and in each treatment group.

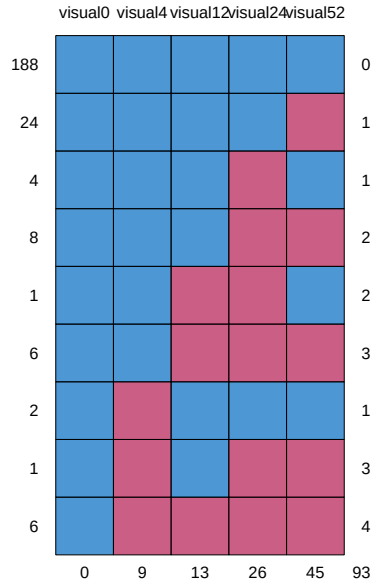
3. Make different graphical representations of the data (see figure below) and discuss the pro- and cons- of each type of display:

   - a boxplot of the values per group and per time
   - a spaghetti plot
   - a mean plot, i.e. mean value in each group over time
   - combine the mean plot and the spaghetti plot



4. Compute the percentage of missing values in each group at each timepoint and provide a graphical representation of it.
   What type of information is provided by the following figure:

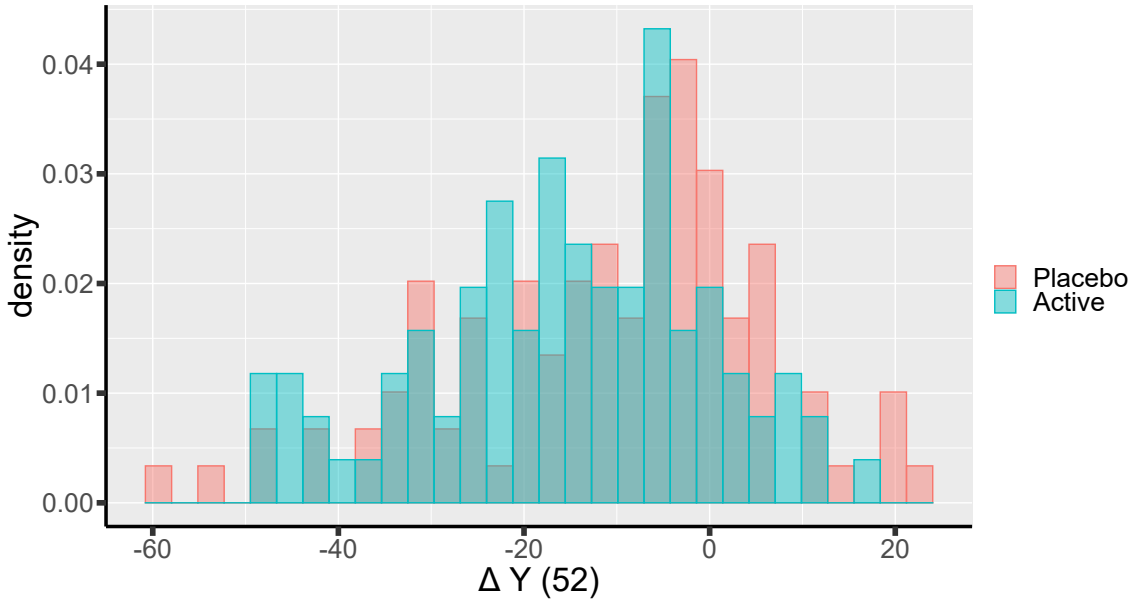
```
library(mice)
md.pattern(armd.wide[,paste0("visual",c(0,4,12,24,52))])
```



Part 2: change from baseline (univariate)

In this second part, we will replicate the univariate analysis presented during the lecture (slides 21-25).

5. Create a new data, `armd.wideCC`, by restricting the dataset `armd.wide` to individuals with complete data at week 0 and 52. Compute the change in outcome from baseline and create an histogram of the change per group.



6. Assess the treatment effect by comparing the change between the two groups using a t-test. Extract the estimated effect, its confidence interval, and p-value. Can you retrieve the estimated treatment effect from the mean values computed in Part 1?

7. Compare the result with fitting a univariate linear regression. Why do we get a (slightly) different p.value?

```
e.lm <- lm(change ~ treat.f, data = armd.wideCC)
summary(e.lm)$coef
```

```
                Estimate Std. Error   t value     Pr(>|t|)
(Intercept)   -11.180952   1.557168 -7.180312 1.466539e-11
treat.fActive  -4.296825   2.292089 -1.874633 6.235402e-02
```

8. What information/data have we discarded in this approach?

Part 3: change from baseline (multivariate)

In this third part, we will replicate the multivariate analysis presented during the lecture (slides 27-35). You will need to load the `LMMstar` package:

```
library(LMMstar)
```

9. Consider the following mixed model:

```
e052.lmm <- lmm(visual ~ treat.f*week,
  repetition = ~week|subject,
  data = armd.long[armd.long$week %in% c("0","52"),])
model.tables(e052.lmm)
```

```
                     estimate   se  df  lower  upper  p.value
(Intercept)            55.336 1.37 238  52.64 58.029 0.00e+00
treat.fActive          -0.758 1.93 238  -4.55  3.035 6.94e-01
week52                -11.095 1.55 196 -14.15 -8.038 1.61e-11
treat.fActive:week52   -4.383 2.27 198  -8.87  0.103 5.54e-02
```

What is the interpretation of each coefficient?
Why does the estimation of the treatment effect differs from the one of part 2?
Which one would you trust most?

10. Can you deduce from the coefficients the estimated average vision at which timepoint? You can check your calculation with the output of `dummy.coef`.

11. It was mentioned during the lecture that the mixed model could be seen as way to "guess" missing values. We now illustrate this point with individual 114:

```
armd.114 <- armd.long[armd.long$subject=="114" & armd.long$week %in% c("0"
    ,"52"),]
armd.114
```

```
     subject lesion treat.f miss.pat week visual
114      114      4 Placebo     --XX    0     45
1074     114      4 Placebo     --XX   52     NA
```

Using the estimated mean coefficients (see previous question) and variance/correlation coefficients:

```
coef(e052.lmm, effects=c("correlation","variance"))
## note: variance at week 52 is sigma^2 k.52^2
```

```
     sigma       k.52  rho(0,52)
14.9115119  1.2397277  0.5612167
```

apply the following formula:

$$\widehat{Y}_{114}(52) = \alpha(52) + \rho(0,52)\frac{\sigma(52)}{\sigma(0)}\left(Y_{114}(0) - \alpha(0)\right)$$

to retrieve the predicted value at week 52 for individual 114 given its baseline value:

```
predict(e052.lmm, newdata = armd.114, type = "dynamic")
```

```
  estimate       se  df    lower    upper
1 37.04983 1.799755 Inf 33.52238 40.57728
```

(in the formula $Y_{114}(t)$ denotes the observed vision at time t for individual 114 and $\alpha(t)$ the modeled mean in the placebo group at time t)

Part 4: longitudinal analysis (multivariate)

In this last part, we will discuss the mixed model presented at the end of the lecture (slides 36-37). We will now work on the entire dataset (i.e. week 0, 4, 12, 24, 52).

12. Create a numeric time variable `week.num` indicating the number of weeks since baseline. Fit a mixed model including in the mean structure the categorical time variable and an interaction between the continuous time variable and the treatment variable:

```
eLin.lmm <- lmm(visual ~ week + week.num:treat.f,
  repetition = ~ week | subject, structure = "UN",
  data = armd.long)
model.tables(eLin.lmm)
```

```
Singular design matrix, coefficient "week.num:treat.fPlacebo" has been removed.
                       estimate     se  df   lower    upper  p.value
(Intercept)              54.954 0.9608 239  53.061 56.84693 0.00e+00
week4                    -2.207 0.5520 243  -3.294 -1.11919 8.51e-05
week12                   -3.585 0.8193 259  -5.198 -1.97156 1.76e-05
week24                   -6.563 1.0585 279  -8.647 -4.47968 2.02e-09
week52                  -11.601 1.5316 203 -14.621 -8.58070 1.25e-12
week.num:treat.fActive   -0.083 0.0409 187  -0.164 -0.00231 4.39e-02
```

What is the interpretation of each coefficient?
What assumption are we making about the treatment effect?
What are the possible benefits of such an assumption?

13. To check this assumption, we will fit a more flexible mixed model:

```
eFlex.lmm <- lmm(visual ~ week*treat.f,
   repetition = ~ week | subject, structure = "UN",
   data = armd.long)
model.tables(eFlex.lmm)
```

```
                    estimate    se  df  lower   upper  p.value
(Intercept)           55.336 1.367 238  52.64 58.0289 0.00e+00
week4                 -1.281 0.765 231  -2.79  0.2254 9.52e-02
week12                -2.352 1.091 220  -4.50 -0.2007 3.23e-02
week24                -6.020 1.318 212  -8.62 -3.4211 8.42e-06
week52               -11.311 1.599 193 -14.46 -8.1576 2.70e-11
treat.fActive         -0.758 1.925 238  -4.55  3.0348 6.94e-01
week4:treat.fActive   -2.204 1.087 232  -4.35 -0.0617 4.38e-02
week12:treat.fActive  -3.508 1.560 222  -6.58 -0.4330 2.55e-02
week24:treat.fActive  -3.070 1.895 216  -6.81  0.6661 1.07e-01
week52:treat.fActive  -4.866 2.317 199  -9.44 -0.2963 3.70e-02
```

What is the interpretation of each coefficient?
Compare the estimated treatment effect to the one computed in part 3. Why does it differ?

14. To visualize both models on the same plot, we will create a dataset containing all combinations of time and treatment:

```
    week week.num treat.f             week week.num treat.f
1      0        0  Active          483   12       12 Placebo
3      0        0 Placebo          721   24       24  Active
241    4        4  Active          723   24       24 Placebo
243    4        4 Placebo          961   52       52  Active
481   12       12  Active          963   52       52 Placebo
```

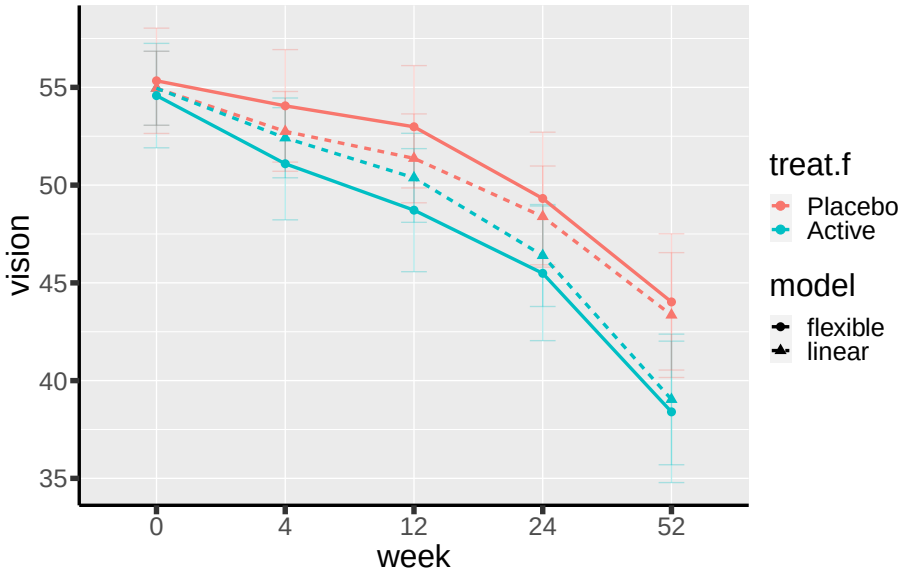
add the predicted value by each model using the `predict` function:

```
pLin <- predict(eLin.lmm, newdata = armdU, keep.newdata = TRUE)
pFlex <- predict(eFlex.lmm, newdata = armdU, keep.newdata = TRUE)
armdU <- rbind(cbind(pLin, model = "linear"),
        cbind(pFlex, model = "flexible"))
armdU[1:4,]
```

```
  week week.num treat.f estimate        se       df    lower    upper  model
1    0        0  Active 54.95417 0.9608237 239.0248 53.06140 56.84693 linear
2    0        0 Placebo 54.95417 0.9608237 239.0248 53.06140 56.84693 linear
3    4        4  Active 52.41563 1.0359495 240.5175 50.37494 54.45632 linear
4    4        4 Placebo 52.74762 1.0358886 240.4368 50.70704 54.78819 linear
```

and make an appropriate graphical display:

```
gg.comp <- ggplot(armdU, aes(x = week, y = estimate, color = treat.f,
        group =  interaction(treat.f,model)))
gg.comp <- gg.comp + geom_errorbar(aes(ymin = lower, ymax = upper),
       width = 0.2, alpha = 0.3)
gg.comp <- gg.comp + geom_point(aes(shape = model), size = 3)
gg.comp <- gg.comp + geom_line(aes(linetype = model), size = 1.25)
gg.comp + ylab("vision")
```



Does `eLin.lmm` provide a reasonable summary of the treatment effect over time? When would you use `eLin.lmm` and when would `eFlex.lmm` be more appropriate?