Solution to the exercises day 8

Basic Statistics for health researchers 2021

22 November 2021

1 Exercise A: what to adjust on?

To obtain a formal argument, we will denote by Y the PET signal, by X the genetic factor(s), and by $Z = (Z_1, Z_2, Z_3)$ the age, scanner type, and radioactive dose. We can express the PET signal as a function of the covariate at baseline (t_0) and follow-up (t_1). Under a linear model, we have for a generic individual:

$$\begin{aligned} Y(t_0) &= \alpha(t_0) + \beta X + \gamma_1 Z_1(t_0) + \gamma_2 Z_2(t_0) + \gamma_3 Z_3(t_0) + \varepsilon(t_0) \\ Y(t_1) &= \alpha(t_1) + \beta X + \gamma_1 Z_1(t_1) + \gamma_2 Z_2(t_1) + \gamma_3 Z_3(t_1) + \varepsilon(t_1) \end{aligned}$$

So the difference is:

$$\begin{aligned} Y(t_1) - Y(t_0) &= \alpha(t_1) - \alpha(t_0) + \gamma_1(Z_1(t_1) - Z_1(t_0)) + \gamma_2(Z_2(t_1) - Z_2(t_0)) + \gamma_3(Z_3(t_1) - Z_3(t_0)) \\ &\quad + \varepsilon(t_1) - \varepsilon(t_0) \\ &= \alpha^* + \gamma_1 \Delta Z_1 + \gamma_2 \Delta Z_2 + \gamma_3 \Delta Z_3 + \varepsilon^* \end{aligned}$$

where $\Delta Z = (\Delta Z_1, \Delta Z_2, \Delta Z_3)$ is the change in age, scanner type and radioactive dose between baseline and follow-up.

1. Genetic factors are time-independent covariate and will therefore affect in the same way the baseline and follow-up measurement. Their effect will therefore cancel out when computing the change score $Y(t_1) - Y(t_0)$ so there is no need to adjust for them.
   Age is technically a time-varying covariate but its variation is small between baseline and follow-up and its effect on the change score could be neglected.
   To test the treatment effect, we could do a two sample t-test comparing the change score of the two groups.

2. The variables scanner type and radioactive dose are time dependent and it is their difference between baseline and follow-up that we should adjust for (see the expression of $Y(t_1) - Y(t_0)$). We could test the treatment effect using a linear regression with the change score as an outcome, group, scanner type, and radioactive dose as regressors and extract the p-value corresponding to the group effect. Note that compared to a t-test, using `lm` in **R** will assume same

variance between treatment groups [1].

In a randomized experiment, this would reduce the residual variance and would therefore lead to a gain in power. We can see that in the expression of $Y(t_1) - Y(t_0)$ as, without adjustment, the residual would be $\xi^* = \gamma_1 \Delta Z_1 + \gamma_2 \Delta Z_2 + \gamma_3 \Delta Z_3 + \varepsilon^*$ instead of ε^*. Since ε^* is independent of Z, the variance of ξ is greater than the variance of ε^*.

In a non-randomized experiment this can lead to a gain in power as well as a reduction in bias, as it will remove any confounding effect from scanner type and radioactive dose (if their effect is linear).

3. Adjusting on post-randomized variables can bias the treatment effect, e.g. if the variable is a mediator of the treatment effect.

   It is very unlikely to be the case here as the production of the radioactive tracer and the choice of the scanner are logistic/technical choices that should be independent of the treatment group. It could be a problem if, for instance, more depressed patients take (much more) time to get into scanner leading to a lower radioactive dose. If the treatment is effective against depression, we will see a larger PET signal in the treated group even if the treatment would not affect the brain serotonergic system.

[1] this can be relaxed using `LMMstar::lmm` or `nlme::gls`

2 Exercise B: Analyzing a longitudinal study

1. The `str` function reveals the presence of missing values in the dataset. We can also visualize see them when looking at the first rows of the dataset:

```
head(armdW)
```

```
  subject lesion line0 visual0 visual4 visual12 visual24 visual52 treat.f miss.pat
1       1      3    12      59      55       45       NA       NA  Active     --XX
2       2      1    13      65      70       65       65       55  Active     ----
3       3      4     8      40      40       37       17       NA Placebo     ---X
4       4      2    13      67      64       64       64       68 Placebo     ----
5       5      1    14      70      NA       NA       NA       NA  Active     XXXX
6       6      3    12      59      53       52       53       42  Active     ----
```

The `miss.pat` variable indicates the missing data pattern: "-" for observed data and "X" for missing data. The `treat.f` contain the randomization group and not the treatment given at a patient at a given timepoint. Indeed at baseline none of the subjects are treated.

2. Using the wide format, we can compute a summary statistic (e.g. the mean) in each group at a specific timepoint using `tapply`. For instance:

```
tapply(X = armd.wide$visual0, ## values
       INDEX = armd.wide$treat.f, ## group
       FUN = mean, ## function to apply to values per group
       na.rm  = TRUE) ## additional argument for FUN
```

```
 Placebo   Active 
55.33613 54.57851
```

Alternatively we can apply the `summarize` function to the long format of the data to compute most of the summary statistics at once. We should add the argument `na.rm` to indicate the missing values should be removed when computing the summary statistics:

```
library(LMMstar)
armd.sum <- summarize(visual ~ week * treat.f, ## value ~ group
        data = armd.long, ## dataset
        na.rm = TRUE) ## additional argument
armd.sum
```

```
   outcome week treat.f observed missing     mean       sd min median max
1   visual    0 Placebo      119       0 55.33613 15.00129  22   56.0  85
2   visual    4 Placebo      117       2 53.96581 15.90973  12   54.0  84
3   visual   12 Placebo      117       2 52.87179 17.20091   3   53.0  85
4   visual   24 Placebo      112       7 49.33036 18.51242   5   50.5  85
5   visual   52 Placebo      105      14 44.43810 18.53683  11   44.0  85
6   visual    0  Active      121       0 54.57851 14.82270  20   57.0  82
7   visual    4  Active      114       7 50.91228 15.81114  12   52.0  84
8   visual   12  Active      110      11 48.67273 17.47665  12   49.5  82
9   visual   24  Active      102      19 45.46078 18.08050   5   45.0  84
10  visual   52  Active       90      31 39.10000 18.40069   4   37.0  84
```

To compute the correlation per group, it is easier to use the wide format:

```
armd.visual <- armd.wide[,paste0("visual",c(0,4,12,24,52))]
cor(armd.visual, use = "pairwise")
```

```
           visual0   visual4  visual12  visual24  visual52
visual0  1.0000000 0.8543813 0.7442610 0.6611932 0.5593174
visual4  0.8543813 1.0000000 0.8425869 0.7387614 0.6135206
visual12 0.7442610 0.8425869 1.0000000 0.8220768 0.7021200
visual24 0.6611932 0.7387614 0.8220768 1.0000000 0.8355586
visual52 0.5593174 0.6135206 0.7021200 0.8355586 1.0000000
```

We could also compute the correlation within group, e.g.:

```
cor(armd.visual[armd.wide$treat.f=="Active",], use = "pairwise")
```

```
           visual0   visual4  visual12  visual24  visual52
visual0  1.0000000 0.8331308 0.7527332 0.6446344 0.5842411
visual4  0.8331308 1.0000000 0.8500741 0.7490043 0.6228478
visual12 0.7527332 0.8500741 1.0000000 0.8711227 0.7253615
visual24 0.6446344 0.7490043 0.8711227 1.0000000 0.8279954
visual52 0.5842411 0.6228478 0.7253615 0.8279954 1.0000000
```

3. A boxplot of the data can be obtained using the `boxplot` function or the geometry `geom_boxplot` from the ggplot2 package:

```
library(ggplot2)
gg.box <- ggplot(armd.long, aes(x = week, y = visual, fill = treat.f))
gg.box <- gg.box + geom_boxplot()
gg.box
```

A spaghetti plot can be obtained using the `matplot` function or the geometry `geom_line` and `geom_point()` from the ggplot2 package:

```
gg.spa <- ggplot(armd.long, aes(x = week, y = visual,
    group = subject, color = treat.f))
gg.spa <- gg.spa + geom_point() + geom_line()
gg.spa
```

The mean plot is similar to a spaghetti plot except it is performed on the (previously computed) summary statistics instead of the original data:

```
gg.mean <- ggplot(armd.sum, aes(x = week, y = mean,
    group = treat.f, color = treat.f))
gg.mean <- gg.mean + geom_point() + geom_line() + ylab("visual")
gg.mean
```

It is possible to combine plots by defining several layers for the graph. Each layer has a specific geometry and dataset. Transparency (argument `alpha`) is used to focus the graph on the mean instead of the individual trajectories:

```
gg.spa2 <- ggplot(mapping = aes(x = week, color = treat.f))
gg.spa2 <- gg.spa2 + geom_line(data = armd.long, alpha = 0.3,
          aes(y = visual, group=subject))
gg.spa2 <- gg.spa2 + geom_point(data = armd.sum, aes(y = mean), size = 3)
gg.spa2 <- gg.spa2 + geom_line(data = armd.sum, aes(y = mean, group =
    treat.f), size = 1.5)
gg.spa2
```

A boxplot gives a very concise and readable representation of the data, even when with many timepoints and with a large number of observations. One can quickly identify trends in mean and variance over repetitions. One may also be able to identify certain deviation to normality (e.g. asymmetry). It, however, does give any information about the correlation between measurements. So in some sense it may exaggerate the variability. It is also not well suited to identify subgroups (e.g. half of the people respond to the treatment and the other half do not).
Spaghetti plots are well suited when there is an ordering of the repetitions (e.g. over time, this is not the case when looking at several brain regions though). They can be used to visually assess the correlation over time and detect groups of observations (e.g. some go up and some go down). However when the sample increase, they become hard to read and one should consider displaying subsets of the observations.
We could have also used a scatterplot of visual at week 52 vs week 0. It gives the full picture of the data when having only 2 measurements but becomes hard to read with more timepoints as there are many pairs of variables.

4. We compute the percentage of missing value at a specific timepoint using the wide format, e.g:

```
100*tapply(is.na(armd.wide$visual52), armd.wide$treat.f, mean)
```

```
 Placebo   Active 
11.76471 25.61983 
```

For the graphical display, we would need to repeat the operation at the various timepoints and gather the results. Instead of doing that, we can reuse the `armd.sum` object, as the percentage of missing values can be deduced from the number of observed and missing values:

```
armd.sum$pc.missing <- armd.sum$missing/(armd.sum$observed+armd.sum$
   missing)
armd.sum[,c("week","treat.f","pc.missing")]
```

```
   week treat.f pc.missing
1     0 Placebo 0.00000000
2     4 Placebo 0.01680672
3    12 Placebo 0.01680672
4    24 Placebo 0.05882353
5    52 Placebo 0.11764706
6     0  Active 0.00000000
7     4  Active 0.05785124
8    12  Active 0.09090909
9    24  Active 0.15702479
10   52  Active 0.25619835
```

We can then use:

```
gg.mis <- ggplot(armd.sum, aes(x=week, y=pc.missing, group=treat.f, color=
   treat.f))
gg.mis <- gg.mis + geom_point(size = 3) + geom_line(, size = 1.25)
gg.mis <- gg.mis + ylab("Percentage of missing data") + scale_y_continuous
   (labels = scales::percent)
gg.mis ## same as figure on slide 16
```

The number of missing values seems to increase over time, especially in the active group. In a real analysis, this would be concerning. Indeed, if patients with bad vision are more likely to drop out the mean computed at the later timepoints will be biased and not in a conservative way. It could also reflect side effects of the treatment that are so serious that the patients choose/have to leave study. However, for simplicity, we will ignore this problem in the latter questions and act as if censoring was independent of the outcome and of the treatment.

The figure obtained via the mice package displays the difference missing patterns. The left numbers correspond to the number of subjects for a specific pattern and the right numbers the number of missing data per pattern. For instance there are 188 subjects with full data and 6 subjects with only baseline data.

5. We can select individual with no missing data at baseline nor at week 52 doing:

```
armd.wideCC <- armd.wide[is.na(armd.wide$visual0)+is.na(armd.wide$visual52
   )==0,]
```

and then compute the change:

```
armd.wideCC$change <- armd.wideCC$visual52 - armd.wideCC$visual0
```

We then obtain a histogram using:

```
gg.dens <- ggplot(armd.wideCC, aes(change, color = treat.f, fill = treat.f
   ))
gg.dens <- gg.dens + geom_histogram(alpha = 0.45, aes(y=..density..),
   position = "identity")
gg.dens <- gg.dens + xlab("\u0394 Y (52)")
gg.dens
```

```
‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

6. With a t-test:

```
e.tt <- t.test(change ~ treat.f, data = armd.wideCC)
e.tt
```

```

        Welch Two Sample t-test

data:  change by treat.f
t = 1.8842, df = 191.47, p-value = 0.06106
alternative hypothesis: true difference in means between group Placebo and group Active is
95 percent confidence interval:
 -0.2013017  8.7949525
sample estimates:
mean in group Placebo  mean in group Active
            -11.18095             -15.47778
```

we obtain an estimated effect of:

```
diff(e.tt$estimate)
```

```
mean in group Active
           -4.296825
```

i.e. a faster decrease in vision in the active group. The corresponding p-value and confidence intervals are displayed the output of the t-test object.
We can try to retrieve this result by comparing the change in each group:

```
armd.sum052 <- armd.sum[armd.sum$week %in% c("0","52"),]
diff.active <- diff(armd.sum052[armd.sum052$treat.f=="Active","mean"])
diff.placebo <- diff(armd.sum052[armd.sum052$treat.f=="Placebo","mean"])
c(diff.active,diff.placebo,diff.active-diff.placebo)
```

```
[1] -15.478512 -10.898039  -4.580473
```

We don't get exactly the same result because `armd.sum` was computed on the entire population while the t-test was performed only among those with complete data.

7. When fitting a linear regression using `lm`, we assume that the residual variance is the same in both groups which is not the case with a t-test. If we were to assume same variance when doing a t-test:

   ```
   t.test(change ~ treat.f, data = armd.wideCC, var.equal = TRUE)
   ```

   ```

   	Two Sample t-test

   data:  change by treat.f
   t = 1.8746, df = 193, p-value = 0.06235
   alternative hypothesis: true difference in means between group Placebo and group Activ
   95 percent confidence interval:
    -0.2239352  8.8175860
   sample estimates:
   mean in group Placebo  mean in group Active
               -11.18095             -15.47778
   ```

   we would get the same as the linear regression.

8. With the complete case approach we discarded all data from individual who had a missing value at week 52. So even if a subject had full data until week 24, he was not included in the analysis. We have seen that there was a strong correlation between timepoints (e.g. >0.8 between week 24 and 52) so it is not optimal to not exploit early measurements of the outcome.

9. To interpret the coefficients it can be useful:

   - to look at graphical display - e.g. see slide 32 of the lecture slides
   - to know the reference level:

```
levels(e052.lmm)$reference
```

```
  treat.f      week 
"Placebo"       "0" 
```

Then
`(Intercept)` is the average vision in the control group at week 0.
`treat.fActive` is the difference in vision between groups at baseline.
`week52` is the time effect in the control group.
`treat.fActive:week52` is the difference in time effect between groups, i.e. the treatment effect.

The estimate differs (slightly) from part 2 because we now take into account all individuals, and not only those with complete data. This will typically result in a more precise estimate and possibly less bias. So this estimate is more reliable.

10. We can use the formula from slide 34 to deduce the estimated vision at each timepoint:

```
c(placebo.0 = as.double(coef(e052.lmm)["(Intercept)"]),
  placebo.52 = sum(coef(e052.lmm)[c("(Intercept)","week52")]),
  active.0 = sum(coef(e052.lmm)[c("(Intercept)","treat.fActive")]),
  active.52 = sum(coef(e052.lmm)))
```

```
placebo.0 placebo.52   active.0  active.52 
 55.33613   44.24126   54.57851   39.10051 
```

We can compare our results with:

```
dummy.coef(e052.lmm)
```

```
  treat.f week estimate       se       df    lower    upper
1 Placebo    0 55.33613 1.366936 238.0491 52.64330 58.02897
2  Active    0 54.57851 1.355592 238.0488 51.90802 57.24900
3 Placebo   52 44.24126 1.770454 205.8764 40.75071 47.73180
4  Active   52 39.10051 1.868344 216.5233 35.41804 42.78298
```

11. We first gather all relevant quantities:

```
sigma0 <- coef(e052.lmm, effects="variance")["sigma"]
sigma52 <- sigma0*coef(e052.lmm, effects="variance")["k.52"]
rho <- coef(e052.lmm, effects="correlation")["rho(0,52)"]
alpha0 <- coef(e052.lmm)["(Intercept)"]
alpha52 <- alpha0 + coef(e052.lmm)["week52"]
```

and then apply the formula:

```
unname(alpha52 + rho * sigma52/sigma0 * (armd.114$visual[1]-alpha0))
```

```
[1] 37.04983
```

Note that because individual 114 has a baseline value below the baseline mean (and the correlation is positive), its predicted value is below the follow-up mean.

12. First note that the reference group has changed:

```
levels(eLin.lmm)$reference
```

```
week   treat.f
 "0" "Placebo"
```

The `(Intercept)` is the average vision at baseline in the active group. The `week4`, `week8`, `week24`, and `week52` coefficients are the change in vision from baseline in the active group. The `week.num:treat.fPlacebo` is the amount of decrease in vision per week due to the treatment.
With this model we can summarize the treatment effect in this single number as we assume a linear treatment effect (i.e. proportional to the number of weeks from baseline).
This assumption makes it easier to communicate the treatment effect (as it is not time specific) and can also lead to a gain in statistical power if the linearity assumption is reasonable.

13. The interpretation of the first five coefficients (`(Intercept)`, `week4`, `week8`, `week24`, and `week52`) is the same as in the previous model. The coefficient `treat.fActive` is the group difference at baseline while the remaining coefficients (`week4:treat.fActive`, `week8:treat.fActive`, `week24:treat.fActive`, and `week52:treat.fActive`) indicate the difference in change in vision between the two groups (i.e. the treatment effect at each follow-up time).
    So the coefficient `week52:treat.fActive` as the same interpretation as the one found in question 9. However its value differ because we use a different model to handle the missing data. Intuitively, we now use values from timepoint 0, 4, 12, 24 (instead of only timepoint 0) to "guess" the missing value. This new estimate should be more precise and less biased.

14. From the graph it seems that we do not miss an important feature of the treatment effect over time by assuming a proportional effect. Note that this assumption enforces that the two group have identical average baseline value, which is reasonable in a randomized study.

    `eFlex.lmm` would be recommended if one is only interested in the long-term treatment effect, as this model makes no assumption on the treatment effect over time. When one is interested in describing the time dynamic of the treatment effect, an approach like `eLin.lmm` can be more useful. Note in between approaches could be considered, e.g. adding non-linear interaction terms such as `I(week.num^2):treat.f`.