Solution to the exercises day 8

Basic Statistics for health researchers

20 November 2023

Exercise A: what to adjust on?

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 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. We could test the treatment effect using a linear regression with the change score as an outcome, group, change in radioactive dose and scanner type 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.
   In a randomized experiment, this adjustment would reduce the residual variance and would therefore lead to a gain in power.
   In an observational study, this adjustment would will 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.

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.

Exercise B: Analyzing a longitudinal study

Part 1: descriptive statistics

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(armd.wide)
```

```
  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` contains 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. The output display the number of observed outcome value, missing outcome value, and a number of summary statistics of the outcome distribution (mean, standard deviation) for each treatment group at each timepoint.
   Summary statistics for the whole cohort can be obtained using:

```
summarize(visual ~ week, na.rm = TRUE, data = armd.long)
```

```
  week observed missing     mean       sd min   q1 median   q3 max
1    0      240       0 54.95417 14.88512  20 45.0   56.5 66.0  85
2    4      231       9 52.45887 15.90042  12 42.0   53.0 64.0  84
3   12      227      13 50.83700 17.42404   3 39.5   52.0 64.5  85
4   24      214      26 47.48598 18.36733   5 36.0   47.0 62.0  85
5   52      195      45 41.97436 18.61865   4 27.0   40.0 56.5  85
```

Note: adding `|subject` in the formula will also display the correlation.

3. What is the best graphical representation depends on the aspect(s) of the data we want to visualize and the number of observations and timepoints.

   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.

   Information about missing data and within-subject correlation are missing or not easily visible on the previous displays.

4. The first figure displays the percentage of missing value at a specific timepoint. The second figure displays the missing data patterns (the left numbers correspond to the number of subjects for a specific pattern). For instance there are 188 subjects with full data and 6 subjects with only baseline data.

   From the first figure, 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.

Part 2: Univariate approach

5. It selects individuals with complete data at baseline and at week 52

6. Using the `t.test` function lead to the following results:

```
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 
```

The estimated effect:

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

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

indicates 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. Note that estimate differs from the one from part one:

```
diff(armd.s[armd.s$week == "0","mean"]-armd.s[armd.s$week == "52","mean"])
```

```
[1] 4.580473
```

As some individuals have been excluded in this part even though they had a non-missing baseline value, e.g.:

```
armd.wide[1,]
```

```
  subject lesion line0 visual0 visual4 visual12 visual24 visual52 treat.f miss.pat
1       1      3    12      59      55       45       NA       NA  Active     --XX
```

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 Active
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. Instead we could use the following syntax to match the result of the Welch t-test in a regression framework:

```
e.tt <- lmm(change ~ treat.f, structure = IND(~treat.f),
            data = armd.wideCC)
model.tables(e.tt)
```

```
                estimate       se       df      lower     upper      p.value
(Intercept)   -11.180952 1.603326 104.0208 -14.360402 -8.001503 2.940177e-10
treat.fActive  -4.296825 2.280499 191.5078  -8.794947  0.201296 6.105844e-02
```

8. We can repeat exactly the same steps with `visual24` instead of `visual52` to study the treatment effect at 24 weeks. Alternatively we can also trust the software to handle correctly the missing values and directly calling `t.test`:

```
t.test(visual52 - visual0 ~ treat.f, data = armd.wide)
```

```

	Welch Two Sample t-test

data:  visual52 - visual0 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 
```

```
t.test(visual24 - visual0 ~ treat.f, data = armd.wide)
```

```

        Welch Two Sample t-test

data:  visual24 - visual0 by treat.f
t = 1.7599, df = 209.36, p-value = 0.07989
alternative hypothesis: true difference in means between group Placebo and group Active is
95 percent confidence interval:
 -0.4041554  7.1306960
sample estimates:
mean in group Placebo  mean in group Active 
            -5.705357             -9.068627 
```

With this (complete case) approach we discarded all data from individuals 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.

Part 3: Multivariate approach

9. To interpret the coefficients it can be useful to know the reference level:

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

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

`(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 and p-value exactly the t-test under equal variance. We can use the formula from the lecture (slide 37) 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
```

which matches the output of `dummy.coef`:

```
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
```

We could retrieve the results of the t-test with heterogeneous variance using:

```
e052.lmm2 <- lmm(visual ~ treat.f*week,
                repetition = treat.f~week|subject,
                data = dfCC)
model.tables(e052.lmm2)
```

```
                       estimate       se       df      lower      upper      p.value
(Intercept)           55.619048 1.487209 104.0217  52.669863 58.5682321 0.000000e+00
treat.fActive         -1.041270 2.128823 191.0550  -5.240284  3.1577441 6.253111e-01
week52               -11.180952 1.603326 103.9912 -14.360412 -8.0014926 2.943139e-10
treat.fActive:week52  -4.296825 2.280499 191.4544  -8.794955  0.2013039 6.105887e-02
```

10. The mixed model e52.lmm is the same as e052.lmm except it includes individual with missing data only at follow-up. Their final value is predicted based on their baseline value which provides some protection against informative dropout due to poor outcome. The mixed model e52.lmm is fitted using data from all timepoints. The final value of patients with missing data is predicted based on their baseline and early follow-up information. This provides substantially more protection against informative dropout due to poor outcome, as there is a strong correlation over time.
    The estimates (between -4.29 and -4.86) and the p-values (between 0.062 and 0.037) do not differ substantially, even though for a clinical trial it would lead to a different conclusion as the p-value is below 0.05 in the last, most reliable, approach.

13. 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):

```
armd.long$week.num <- as.numeric(as.character(armd.long$week))

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

```
Design matrix for the mean structure is singular.
Coefficient "week.num:treat.fPlacebo" has been removed.
                           estimate         se       df       lower        upper      p.val
(Intercept)             54.95416667 0.96083065 239.0200  53.0613893 56.846944015 0.000000e+
week4                   -2.20654872 0.55201346 242.6356  -3.2938989 -1.119198564 8.505743e-
week12                  -3.58487586 0.81927366 258.5491  -5.1981745 -1.971577178 1.757722e-
week24                  -6.56331226 1.05848300 279.3102  -8.6469293 -4.479695245 2.015522e-
week52                 -11.60066367 1.53164482 203.2552 -14.6206139 -8.580713446 1.248779e-
week.num:treat.fActive  -0.08299735 0.04090117 187.3855  -0.1636833 -0.002311424 4.385081e-
```

The coefficient (Intercept) is the average vision at baseline (in both groups). The week4, week8, week24, and week52 coefficients are the change in vision from baseline in the placebo group. The week.num:treat.fActive is the amount of decrease in vision per week due to the treatment.
The message we get when fitting the model comes from the interaction we implicitely asked for a parameter modeling the group difference at baseline. Such parameter cannot be estimated with the "proportional" parametrisation since it enforces no treatment effect at baseline.

The treatment effect at week 52 would be 52 times the `week.num:treat.fActive`:

```
52*coef(eLin.lmm)["week.num:treat.fActive"]
```

```
week.num:treat.fActive
             -4.315862
```

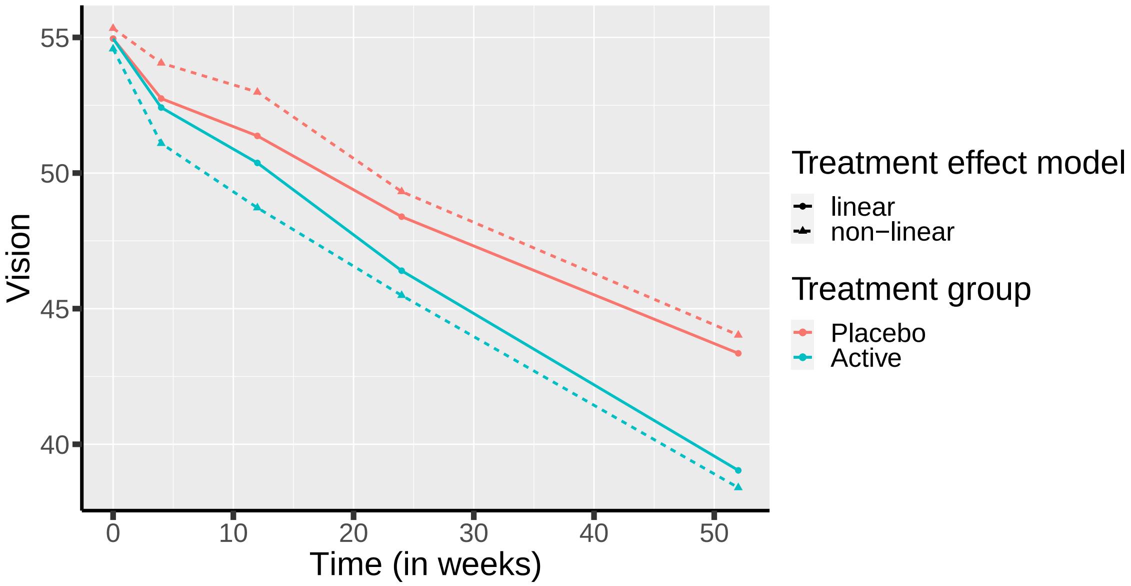
This exactly matches the difference between the two groups in predicted change in vision from baseline to week 52:

```
grid <- unique(armd.long[,c("week","week.num","treat.f")],)
gridA <- predict(eLin.lmm, newdata = grid, keep.newdata = TRUE)

mu0_A <- gridA[gridA$treat.f == "Active" & gridA$week==0,"estimate"]
mu52_A <- gridA[gridA$treat.f == "Active" & gridA$week==52,"estimate"]
mu0_P <- gridA[gridA$treat.f == "Placebo" & gridA$week==0,"estimate"]
mu52_P <- gridA[gridA$treat.f == "Placebo" & gridA$week==52,"estimate"]

(mu52_A - mu0_A) - (mu52_P - mu0_P)
```

```
[1] -4.315862
```

Note: The linearity 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. This can be investigated by comparing the fit with the more flexible model:

```
gridB <- predict(e.lmm, newdata = grid, keep.newdata = TRUE)

gg.fit <- ggplot(mapping = aes(x = week.num, y = estimate,
                               color = treat.f, group = treat.f))
gg.fit <- gg.fit + geom_point(data = gridA, aes(shape = "linear"),
                              size = 2)
gg.fit <- gg.fit + geom_line(data = gridA, aes(linetype = "linear"),
                             size = 1)
gg.fit <- gg.fit + geom_point(data = gridB, aes(shape = "non-linear"),
                              size = 2)
gg.fit <- gg.fit + geom_line(data = gridB, aes(linetype = "non-linear"),
                             size = 1)
gg.fit <- gg.fit + labs(x = "Time (in weeks)", y = "Vision",
                        shape = "Treatment effect model",
                        linetype = "Treatment effect model",
                        color = "Treatment group")
gg.fit
```



Vision
55
50
45
40
0
10
20
30
40
50
Time (in weeks)
Treatment effect model
linear
non−linear
Treatment group
Placebo
Active