Partial residuals with the package LMMstar

Brice Ozenne

May 12, 2024

This vignette details how partial residuals can be used to illustrate model fit in a linear regression and a linear mixed model when using the package LMMstar. We thus start by loading the necessary packages:

```
library(LMMstar)
library(ggplot2)
```

1 Univariate linear regression

To illustrate the use of partial residuals we will use the `state.x77` dataset:

```
df1 <- data.frame(lifeExp = state.x77[,4],
                  illiteracy = state.x77[,3],
                  income = state.x77[,2]/1000,
                  murder = state.x77[,5],
                  edu = cut(state.x77[,6],c(0,50,60,100)))
head(df1,4)
```

```
         lifeExp illiteracy income murder      edu
Alabama    69.05        2.1  3.624   15.1   (0,50]
Alaska     69.31        1.5  6.315   11.3 (60,100]
Arizona    70.55        1.8  4.530    7.8  (50,60]
Arkansas   70.66        1.9  3.378   10.1   (0,50]
```

which contains information about life expectancy (`lifeExp`), income (`income`), illeteracy (`illiteracy`), murder rate (`murder`), and the percentage of high-school graduates (as categorical variable) in various states in the USA. For later use we display a few descriptive for each covariate value:

```
summarize(lifeExp + illiteracy + income + murder + edu ~ 1, data = df1,
          columns = c("observed","missing","mean","min","max","sd"))
```

```
       outcome observed missing    mean    min    max      sd
1      lifeExp       50       0 70.8786 67.960 73.600 1.34239
2   illiteracy       50       0  1.1700  0.500  2.800 0.60953
3       income       50       0  4.4358  3.098  6.315 0.61447
4       murder       50       0  7.3780  1.400 15.100 3.69154
5  edu:(50,60]       50       0  0.5600  0.000  1.000 0.50143
6 edu:(60,100]       50       0  0.1600  0.000  1.000 0.37033
```

and check there are no missing values. Here for the categorical covariates the mean indicates the relative frequency of occurrence (56% and 16%) and other columns like `sd` should be ignored.

1.1 No interaction

Suppose we are interested in relating life expectancy (Y) to income (X). We cannot directly illustrate this relationship, as it could be confounded by other variables such as illiteracy (Z_1), murder rate (Z_2). and education (Z_3). We will therefore use a linear model to control for those variables ($\boldsymbol{Z} = (Z_1, Z_2, Z_3)$) where, for simplicity, we assume a linear effect for all variables:

$$Y = \alpha + \beta X + \gamma_1 Z_1 + \gamma_2 Z_2 + \gamma_2 Z_3 + \varepsilon$$

```
e.lm <- lmm(lifeExp ~ income + illiteracy + murder + edu, data = df1)
model.tables(e.lm)
```

```
            estimate       se     df    lower    upper    p.value
(Intercept) 71.57828 1.138488 44.009 69.28382 73.87274 0.0000e+00
income       0.19270 0.252649 44.009 -0.31648  0.70188 4.4971e-01
illiteracy   0.17590 0.320967 44.009 -0.47096  0.82276 5.8644e-01
murder      -0.27822 0.047855 44.009 -0.37467 -0.18178 6.3280e-07
edu(50,60]   0.30141 0.414015 44.009 -0.53298  1.13580 4.7046e-01
edu(60,100]  0.77306 0.513118 44.009 -0.26105  1.80718 1.3906e-01
```

Note that the estimates are nearly identical to the ones of the `lm` function:

```
coef(e.lm) - coef(lm(lifeExp ~ income + illiteracy + murder + edu, data = df1))
```

```
(Intercept)      income  illiteracy      murder  edu(50,60] edu(60,100]
-2.7001e-13  2.0345e-13 -2.9035e-13  6.3283e-15 -4.2277e-13 -5.3091e-13
```

A graphical display can now be obtained by modifying the original outcome, life expectancy, had every state had the same illiteracy and murder rate:

```
df1$pres <- residuals(e.lm, type = "partial", variable = c("(Intercept)","income"))
head(df1)
```

```
           lifeExp illiteracy income murder      edu   pres
Alabama      69.05        2.1  3.624   15.1   (0,50] 72.882
Alaska       69.31        1.5  6.315   11.3 (60,100] 71.417
Arizona      70.55        1.8  4.530    7.8  (50,60] 72.102
Arkansas     70.66        1.9  3.378   10.1   (0,50] 73.136
California   71.71        1.1  5.114   10.3 (60,100] 73.609
Colorado     72.06        0.7  4.884    6.8 (60,100] 73.056
```

By default, the partial residuals are computed substracting the effect of the covariates, i.e. had each state got no illiteracy, no murder, and the lowest education level:

```
c(69.05 - 0.17590 * 2.1 - (-0.27822) * 15.1,
  69.31 - 0.17590 * 1.5 - (-0.27822) * 11.3 - 0.77306)
```

```
[1] 72.882 71.417
```

The element `"(Intercept)"` was specified in the `variable` argument to avoid to substract the mean value from the outcome and keep a plausible range of values for the outcome.

One may wish to compute the life expectancy had every state got a specific illiteracy, murder rate, and education. Say the most common in the sample: 1.17, 7.378, and (50,60] based on the descriptive statistics. This can be obtained by specifying these values in the argument `at`:

```
df1$pres2 <- residuals(e.lm, type = "partial", variable = c("(Intercept)","income"),
                       at = data.frame(illiteracy = 1.17, murder = 7.378, edu = "(50,60]"))
head(df1)
```

```
           lifeExp illiteracy income murder      edu   pres  pres2
Alabama      69.05        2.1  3.624   15.1   (0,50] 72.882 71.336
Alaska       69.31        1.5  6.315   11.3 (60,100] 71.417 69.871
Arizona      70.55        1.8  4.530    7.8  (50,60] 72.102 70.557
Arkansas     70.66        1.9  3.378   10.1   (0,50] 73.136 71.590
California   71.71        1.1  5.114   10.3 (60,100] 73.609 72.064
Colorado     72.06        0.7  4.884    6.8 (60,100] 73.056 71.510
```

or doing the calcuation by hand:

```
c(69.05 - 0.17590 * (2.1-1.170) - (-0.27822) * (15.1-7.378) + 0.30141,
  69.31 - 0.17590 * (1.5-1.170) - (-0.27822) * (11.3-7.378) + 0.30141 - 0.77306)
```

```
[1] 71.336 69.871
```

Note that changing in counterfactual only shifts the partial residuals by a constant, here:

```
unique(df1$pres2 - df1$pres)
```

```
[1] -1.5455
```

so does not affect the relation between the counterfactual outcome (here `lifeExp`) and the exposure of interest (here `income`). One can then get a graphical display either manually using ggplot:

```
gg.pres <- ggplot(df1) + geom_point(aes(x=income, y=pres))
gg.pres <- gg.pres + geom_abline(intercept = coef(e.lm)["(Intercept)"],
                                 slope = coef(e.lm)["income"])
gg.pres <- gg.pres + ggtitle("(B) partial residuals")
gg.pres
```

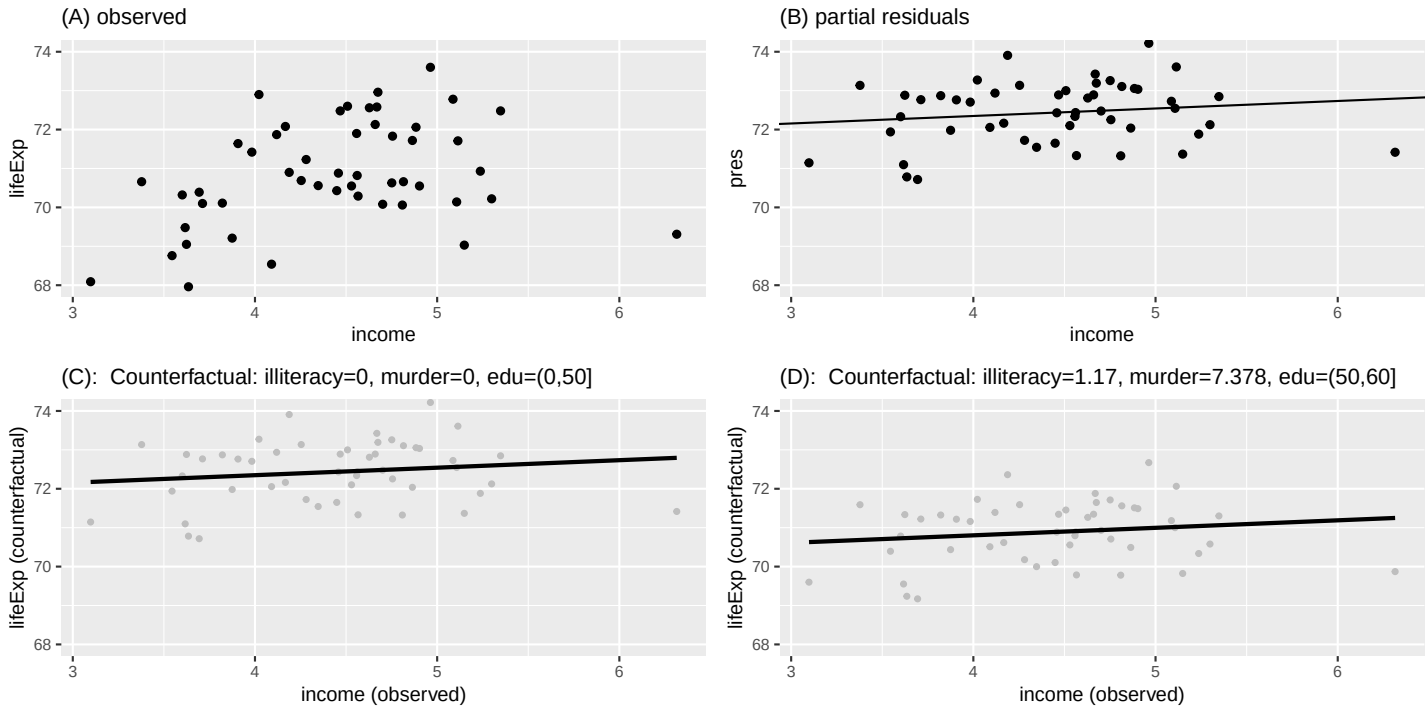
or directly via the plot function:

```
plot(e.lm, type = "partial", variable = c("(Intercept)","income")) # C
plot(e.lm, type = "partial", variable = c("(Intercept)","income"),
     at = data.frame(illiteracy = 1.17, murder = 7.378, edu = "(50,60]")) # D
```

These can be compared to displaying the observed outcome vs. income:

```
gg.obs <- ggplot(df1) + geom_point(aes(x=income, y=lifeExp))
gg.obs <- gg.obs + ggtitle("(A) observed")
gg.obs
```

where it is apparent that by using the partial residuals, the data has been normalized and exhibit less variability.



The output of the `plot` method is a list containing an element plot with the ggplot object and an element data with the dataset. To avoid actually displaying the graph one can use the method `autoplot` to only save the ggplot object:

```
ls.plot <- autoplot(e.lm, type = "partial", variable = c("(Intercept)","income"))
lapply(ls.plot, class)
```

```
$data
[1] "residuals_lmm" "data.frame"

$plot
[1] "gg"     "ggplot"
```

One can re-create the plot based on the data argument or modify the existing plot, e.g. displaying with the y axis between 68 and 74:

```
ls.plot$plot  + coord_cartesian(ylim=c(68,74))
```

1.2 What about confidence intervals?

A common question is whether one can display confidence intervals for the regression line. It is possible to add confidence intervals on the plot either via the argument `ci.alpha`:

```
plot(e.lm, type = "partial", variable = c("(Intercept)","income"), ci.alpha = 0.25) ## E
```

or by requesting confidence intervals for the fitted lines via the argument `pres.ci` when calling `residuals`:

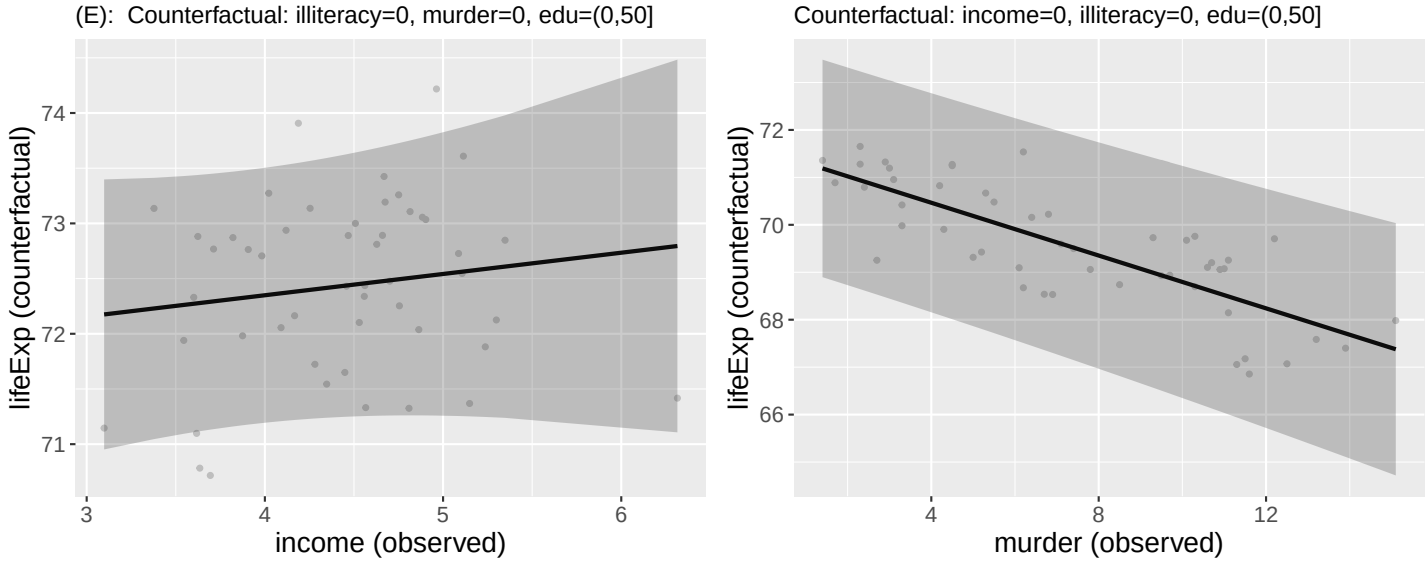
```
pres.ci <- residuals(e.lm, type = "partial", variable = c("(Intercept)","income"),
                     keep.data = TRUE, fitted.ci = TRUE)
head(pres.ci)
```

```
  lifeExp illiteracy income murder    edu fitted fitted.lower fitted.upper r.partial
1   69.05          0  3.624      0 (0,50] 72.277       71.115       73.439    72.882
2   69.31          0  6.315      0 (0,50] 72.795       71.107       74.483    71.417
3   70.55          0  4.530      0 (0,50] 72.451       71.253       73.649    72.102
4   70.66          0  3.378      0 (0,50] 72.229       71.046       73.413    73.136
5   71.71          0  5.114      0 (0,50] 72.564       71.254       73.874    73.609
6   72.06          0  4.884      0 (0,50] 72.519       71.261       73.778    73.056
```

which can be added to the previous graphical display, e.g.:

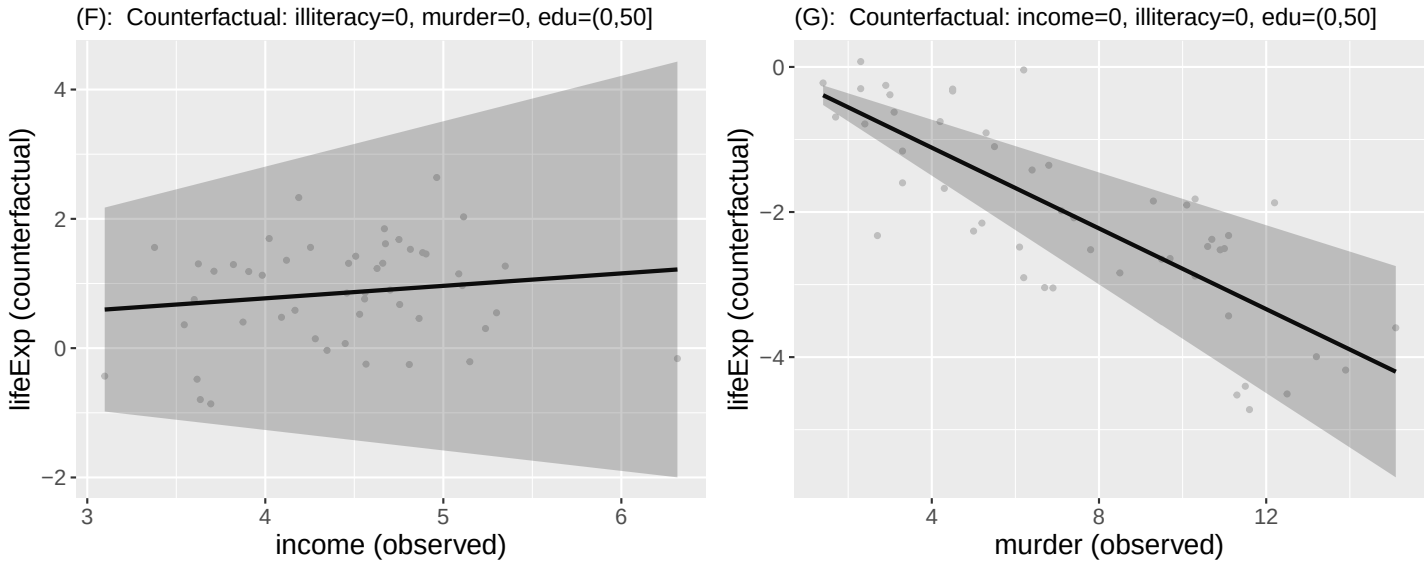
```
gg.pres + geom_ribbon(data = pres.ci, alpha = 0.25,
                      aes(ymin = fitted.lower, ymax = fitted.upper, x = income))
```

The first plot is displayed in the left panel of the figure below. A similar partial residual plot but now for the `murder` variable is displayed in the right panel.



In many case the uncertainty represented here is of little interest, since it is the uncertainty of the intercept plus the exposure effect. This is why even though the `murder` variable was highly significant ($p<0.001$) whereas the income variable was not significant ($p=0.45$) the confidence intervals looks large in both cases. To only capture the uncertainty relative to the `income` or `murder` variable one should remove the intercept value, e.g. by omitting `"(Intercept)"` from the `var` argument:

```
plot(e.lm, type = "partial", variable = "income", ci.alpha = 0.25) ## F
plot(e.lm, type = "partial", variable = "murder", ci.alpha = 0.25) ## G
```



The unpleasant side effect is that the range of values on the y-axis appears unrealistic now. The statistical uncertainty may therefore be better communicated otherwise, e.g. reporting confidence intervals or p-values related to the covariate effect and keeping the partial residual plot free of confidence intervals.

1.3 Interaction with a categorical variable

Suppose that we are now interested in relating life expectancy (Y) to both income (X_1) for various level of education ($X_2 \in \{a, b, c\}$), adjusting for other variables such as illiteracy (Z_1) and murder rate (Z_2). As before we assume a linear effect for all variables:

$$Y = \alpha + \beta_{1a} X_1 \mathbb{1}_{X_2=a} + \beta_{1b} X_1 \mathbb{1}_{X_2=b} + \beta_{1c} X_1 \mathbb{1}_{X_2=c} + \gamma_1 Z_1 + \gamma_2 Z_2 + \varepsilon$$

where $\mathbb{1}_x$ denotes the indicator variable taking value 1 when x is true and 0 otherwise. This model can be estimated with the following R code

```
e.lmI <- lmm(lifeExp ~ income:edu + illiteracy + murder, data = df1)
model.tables(e.lmI)
```

```
                  estimate       se     df    lower    upper    p.value
(Intercept)       71.78584 1.209517 44.009 69.34823 74.22344 0.0000e+00
illiteracy         0.12870 0.319145 44.009 -0.51449  0.77189 6.8871e-01
murder            -0.27940 0.048208 44.009 -0.37656 -0.18224 6.7276e-07
income:edu(0,50]   0.17147 0.297725 44.009 -0.42855  0.77149 5.6760e-01
income:edu(50,60]  0.22526 0.252110 44.009 -0.28284  0.73335 3.7646e-01
income:edu(60,100] 0.30377 0.236929 44.009 -0.17373  0.78126 2.0652e-01
```

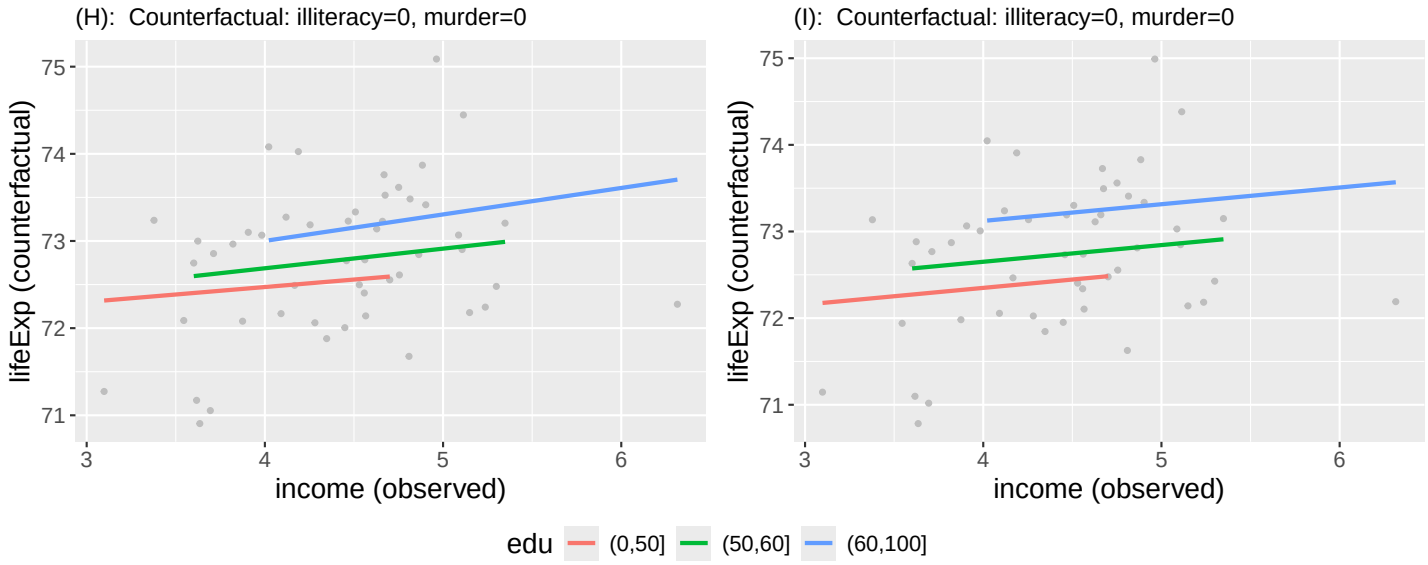
Note: this model is the same as `lmm(lifeExp ~ income*edu + illiteracy + murder, data = df1)` but uses a different parametrisation.

Similarly as before, we can use the `plot` function to display the partial residuals with respect to both `income` and `edu`:

```
plot(e.lmI, type = "partial", variable = c("(Intercept)","income","edu")) ## H
```

which can be compared to a plot assuming no interaction:

```
plot(e.lm, type = "partial", variable = c("(Intercept)","income","edu")) ## I
```



The partial residuals can also be output via the `residuals` method:

```
residuals(e.lmI, type = "partial", variable = c("(Intercept)","income","edu"))[1:5]
```

```
[1] 72.999 72.274 72.498 73.237 74.446
```

and one can check that they are evaluated by substracting the effect of the other variables (here `illiteracy` and `murder`), e.g.:

```
c(69.05 - 0.12870 * 2.1 - (-0.27940) * 15.1,
  69.31 - 0.12870 * 1.5 - (-0.27940) * 11.3)
```

```
[1] 72.999 72.274
```

Here we computed partial residuals representing the life expectancy in the states had there be no murder nor illiteracy. We could also consider the case of average murder rate and illiteracy:

```
residuals(e.lmI, type = "partial", variable = c("(Intercept)","income"),
          at = data.frame(illiteracy = 1.17, murder = 7.378))[1:5]
```

```
[1] 71.088 70.363 70.587 71.327 72.535
```

which we can also retrieve by hand:

```
c(69.05 - 0.12870 * (2.1-1.170) - (-0.27940) * (15.1-7.378),
  69.31 - 0.12870 * (1.5-1.170) - (-0.27940) * (11.3-7.378))
```

```
[1] 71.088 70.363
```

2 Linear mixed model

To illustrate the use of partial residuals we will use data from a two-arm randomized trial comparing the quality of the vision over time of patients under placebo vs. active drug. We first re-shape the data:

```
data(armd.wide, package = "nlmeU")
armd.long <- reshape(armd.wide, direction ="long",
                     varying = paste0("visual",c(0,4,12,24,52)), times = c(0,4,12,24,52),
                     timevar = "week.num", v.names = "visual")
armd.long$week <- as.factor(armd.long$week.num)
```

and notice that the outcome (`visual`) and the covariate `lesion` can be missing:

```
summarizeNA(armd.long)
```

frequency	missing.pattern	n.missing	subject	lesion	line0	treat.f	miss.pat	week.num	visual	id	week
1106	000000000	0	0	0	0	0	0	0	0	0	0
89	000000100	1	0	0	0	0	0	0	1	0	0
1	010000000	1	0	1	0	0	0	0	0	0	0
4	010000100	2	0	1	0	0	0	0	1	0	0

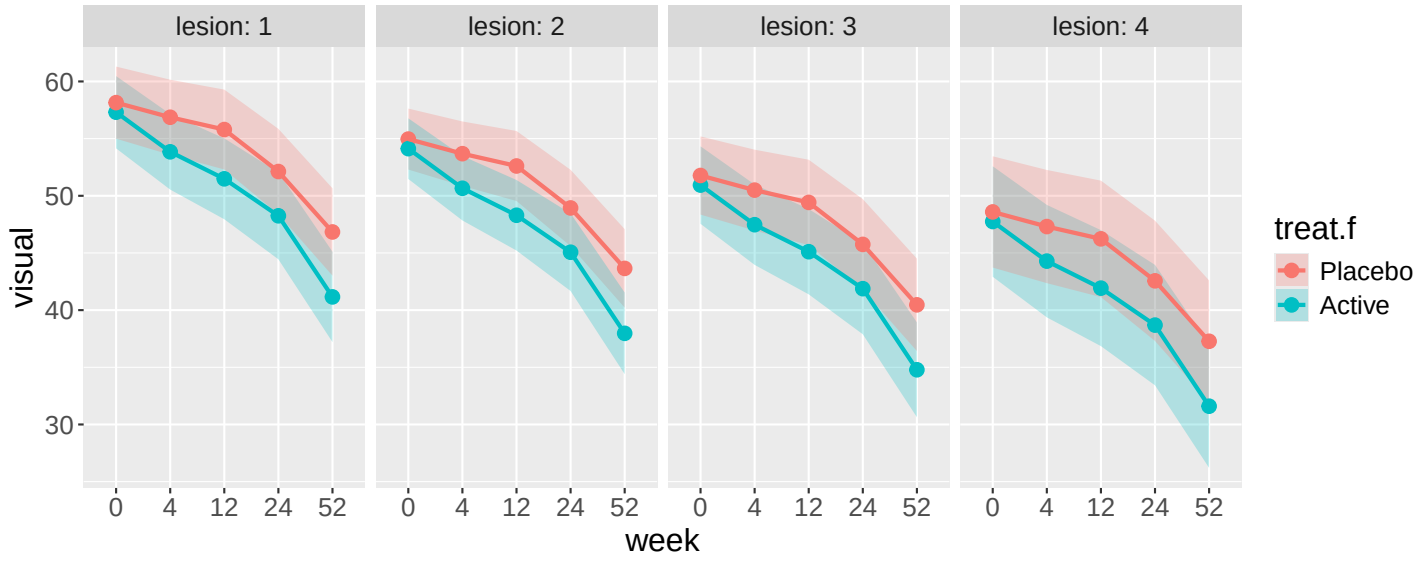
This is why a warning is displayed when fitting the linear mixed model:

```
e.lmm <- lmm(visual ~ week*treat.f + lesion, data = armd.long, repetition =~week|subject)
```

```
Warning message:
In .lmmNormalizeData(as.data.frame(data)[unique(stats::na.omit(var.all))],  :
  Can only handle missing values in the outcome variable visual.
  5 observations with missing values in "lesion" have been removed.
  1 cluster has been removed.
```

To visualize the model fit, we can display the fitted mean for each level of baseline lesion:

```
plot(e.lmm, facet = ~lesion, labeller = label_both, facet_ncol = 4)
```



We can retrive the fitted values from the estimated coefficients:

```
round(coef(e.lmm),2)
```

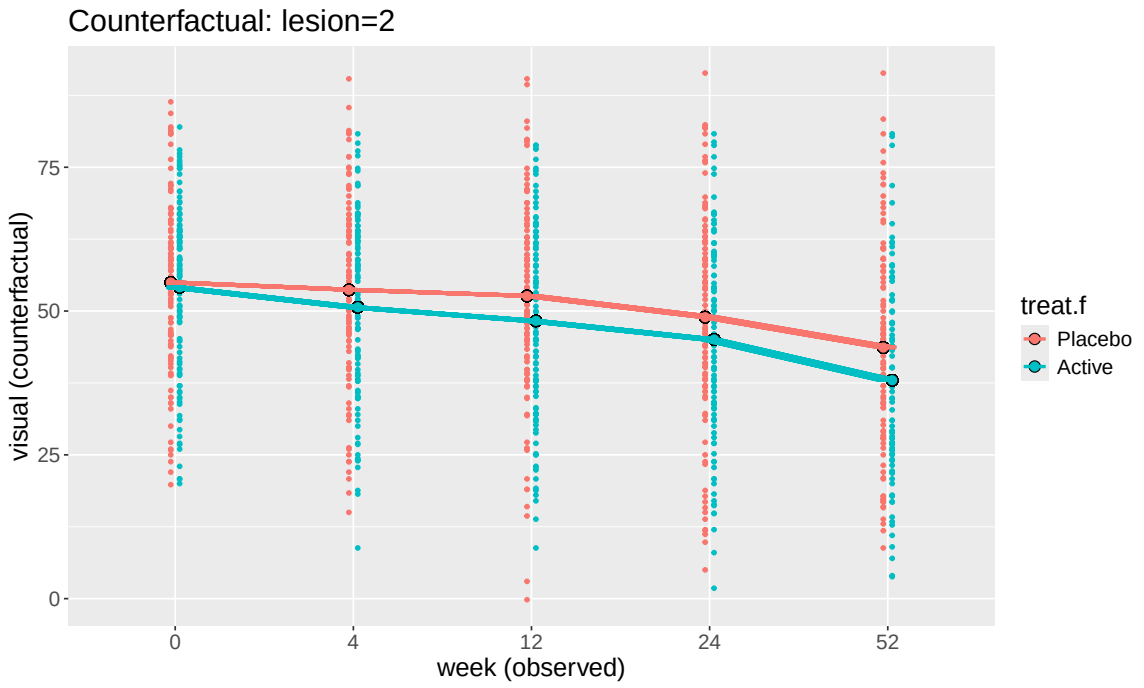
```
        (Intercept)               week4              week12              week24
              61.33               -1.28               -2.35               -6.03
             week52      treat.fActive              lesion week4:treat.fActive
             -11.31               -0.84               -3.19               -2.19
week12:treat.fActive week24:treat.fActive week52:treat.fActive
              -3.47               -3.03               -4.84
```

- in the Placebo group with lesion=1, the estimated average baseline mean is `(Intercept)+1*lesion`, i.e. 61.33-3.19=58.14. When lesion=4, the estimated average baseline mean is `(Intercept)+4*lesion`, i.e. 61.33-4*3.19=48.57.
- the estimated average baseline mean in the Active group is shifted by `treat.fActive` i.e. -0.84 from the Placebo group.
- in the Placebo group with lesion=1, the estimated average week 52 mean is `(Intercept)+week52+1*lesion`, i.e. 61.33-11.31-3.19=46.83.
- the estimated average week 52 mean in the Active group is shifted by `treat.fActive+week52:treat.fActive` i.e. -0.84-4.84=-5.68 from the Placebo group.

Unfortunately, the display of the fitted value becomes overwhelming with considering more covariates or more covariate levels. Instead can visualize the partial residuals, e.g. here the outcome and fitted values had there be no lesion:

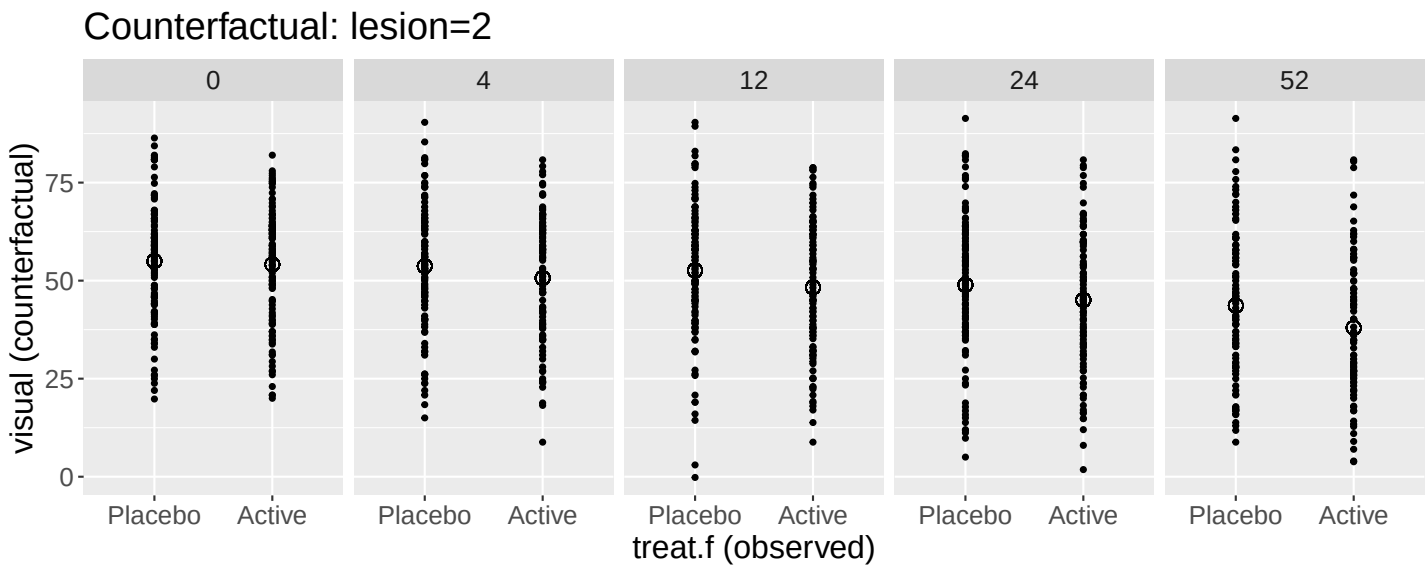
- on the same panel with a difference color for each treatment group:

```
plot(e.lmm, type = "partial", variable = c("(Intercept)","week","treat.f"),
     at = data.frame(lesion = 2))
```



- on a separate panel for each timepoint:

```
plot(e.lmm, type = "partial", variable = c("(Intercept)","week","treat.f"),
     facet =~week, facet_nrow = 1, time.var = "treat.f", color = FALSE,
     at = data.frame(lesion = 2))
```



The calculation of the partial residuals is similar to the univariate regression:

```
armd.long$pres <- residuals(e.lmm, type = "partial",
                            variable = c("(Intercept)","week","treat.f"),
                            at = data.frame(lesion = 2))
head(armd.long)
```

```
    subject lesion line0 treat.f miss.pat week.num visual id week   pres
1.0       1      3    12  Active     --XX        0     59  1    0 62.187
2.0       2      1    13  Active     ----        0     65  2    0 61.813
3.0       3      4     8 Placebo     ---X        0     40  3    0 46.373
4.0       4      2    13 Placebo     ----        0     67  4    0 67.000
5.0       5      1    14  Active     XXXX        0     70  5    0 66.813
6.0       6      3    12  Active     ----        0     59  6    0 62.187
```

here substract the estimated lesion effect from the observed outcome:

```
c(59 - (-3.19) * (3-2),
  65 - (-3.19) * (1-2))
```

```
[1] 62.19 61.81
```

In particular, the partial residuals for patient with lesion equal to two is the observed outcome.

3 R session

Details of the R session used to generate this document:

```
sessionInfo()
```

```
R version 4.3.3 (2024-02-29)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.4 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: Europe/Copenhagen
tzcode source: system (glibc)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] LMMstar_1.1.0 ggpubr_0.6.0  ggplot2_3.5.1

loaded via a namespace (and not attached):
 [1] utf8_1.2.4          future_1.33.2       generics_0.1.3      tidyr_1.3.1
 [5] rstatix_0.7.2       lattice_0.22-5      listenv_0.9.1       digest_0.6.35
 [9] magrittr_2.0.3      grid_4.3.3          Matrix_1.6-5        backports_1.4.1
[13] survival_3.5-8      gridExtra_2.3       purrr_1.0.2         fansi_1.0.6
[17] scales_1.3.0        codetools_0.2-19    numDeriv_2016.8-1.1 abind_1.4-5
[21] lava_1.8.0          cli_3.6.2           rlang_1.1.3         parallelly_1.37.1
[25] future.apply_1.11.2 cowplot_1.1.3       munsell_0.5.1       splines_4.3.3
[29] withr_3.0.0         tools_4.3.3         parallel_4.3.3      ggsignif_0.6.4
[33] dplyr_1.1.4         colorspace_2.1-0    globals_0.16.3      broom_1.0.5
[37] vctrs_0.6.5         R6_2.5.1            lifecycle_1.0.4     car_3.1-2
[41] pkgconfig_2.0.3     pillar_1.9.0        gtable_0.3.5        glue_1.7.0
[45] tibble_3.2.1        tidyselect_1.2.1    farver_2.1.1        nlme_3.1-163
[49] carData_3.0-5       labeling_0.4.3      compiler_4.3.3
```