Partial correlation in linear models

Brice Ozenne

August 30, 2022

1 Summary

This document starts by presenting how to extract from a (univariate) linear regression model partial correlation coefficients. It also precise what type of "partial" (i.e. adjusted on which covariate) we get. When having multiple measurements of pairs of variables, various technics to estimate (partial) correlations are being compared.

2 Illustration

For illustration we will use the following packages:

```
library(LMMstar);library(mvtnorm);library(ggplot2);library(nlme)
LMMstar.options(method.numDeriv = "Richardson",
  columns.confint = c("estimate","se","statistic","df","p.value"))
```

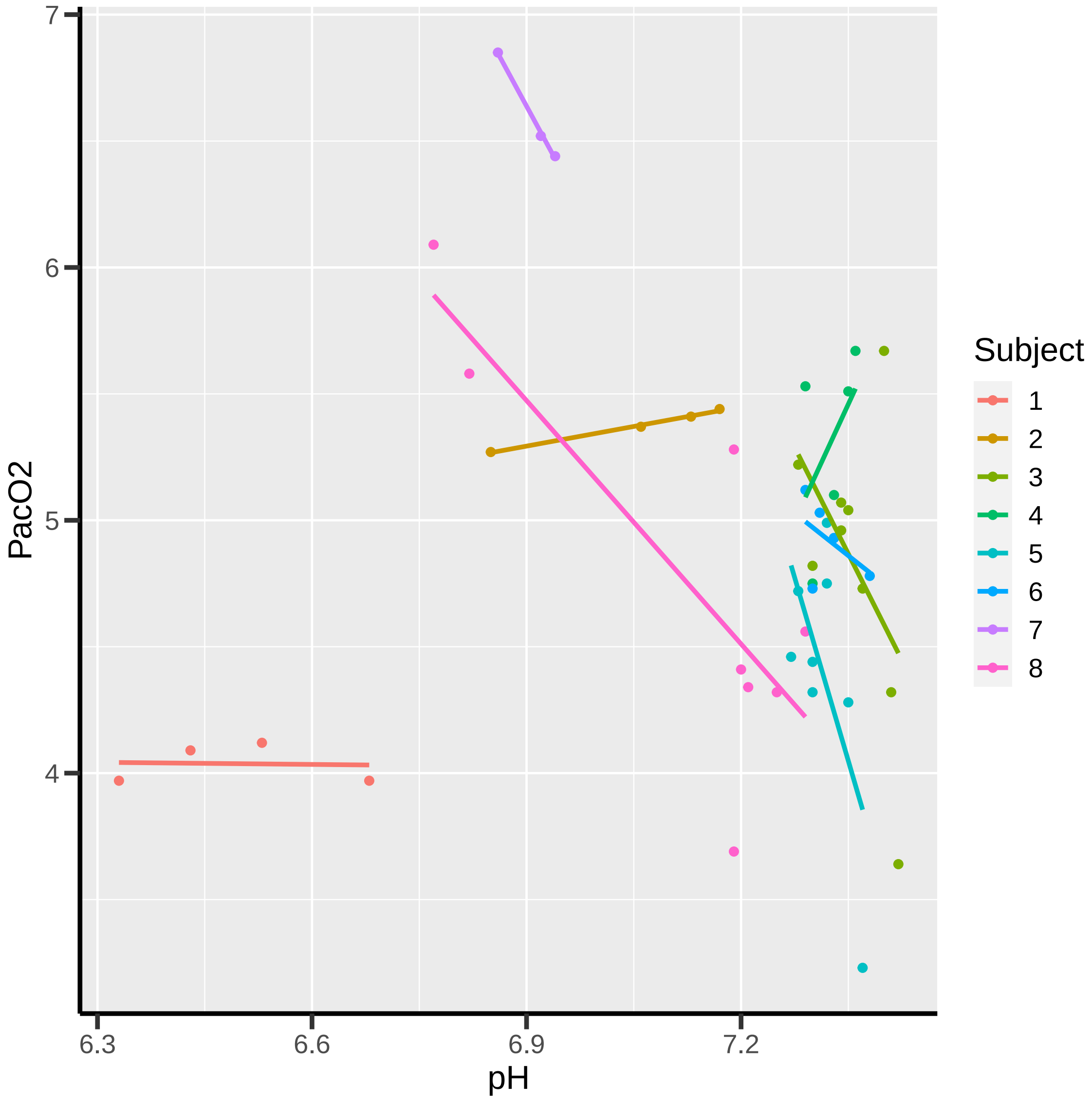
and dataset (Bland and Altman, 1995):

```
data("bland1995", package = "rmcorr")
bland1995$Subject <- as.factor(bland1995$Subject)
head(bland1995)
```

```
  Subject   pH PacO2
1       1 6.68  3.97
2       1 6.53  4.12
3       1 6.43  4.09
4       1 6.33  3.97
5       2 6.85  5.27
6       2 7.06  5.37
```

The aim is to relate intramural pH and PaCO2 using eight subjects:

```
gg <- ggplot(bland1995, aes(x = pH, y = PacO2,
       group = Subject, color = Subject))
gg <- gg + geom_point() + geom_smooth(method = "lm", se = FALSE)
gg
```


7
6
5
4
PacO2
6.3
6.6
6.9
7.2
pH
Subject
1
2
3
4
5
6
7
8

3 Partial partial in multiple linear regression

Consider the linear model:

```
e.lmm <- lmm(pH ~ Subject + PacO2, data = bland1995)
eTable.lmm <- model.tables(e.lmm)
eTable.lmm
```

```
              estimate         se df      lower       upper      p.value
(Intercept)  6.9298543 0.12946898 38  6.6677580  7.19195056 0.000000e+00
Subject2     0.7046113 0.07735488 38  0.5480145  0.86120804 4.277623e-11
Subject3     0.9500127 0.06109545 38  0.8263314  1.07369394 0.000000e+00
Subject4     0.9715577 0.07350906 38  0.8227464  1.12036905 8.881784e-16
Subject5     0.8603817 0.05839543 38  0.7421663  0.97859708 0.000000e+00
Subject6     0.9264284 0.06599450 38  0.7928295  1.06002730 0.000000e+00
Subject7     0.6921056 0.10490935 38  0.4797277  0.90448342 8.670218e-08
Subject8     0.7033361 0.06157141 38  0.5786913  0.82798087 7.460699e-14
PacO2       -0.1083230 0.02989281 38 -0.1688379 -0.04780822 8.471081e-04
```

The F-statistic testing the effect of each factor:

```
anova(e.lmm)
```

```
             Multivariate Wald test

  F-statistic       df p.value
1      48.247 (7,38.0)  <0.001 ***
2      13.131 (1,38.0)  <0.001 ***
```

equal the Wald-statistic squared (divided by 1, the number of parameters)

```
Wald <- eTable.lmm["PacO2","estimate"]/eTable.lmm["PacO2","se"]
Wald^2
```

```
[1] 13.13132
```

This F-statistic is also related to the sum of squares (ANOVA). Consider a model with a single regressor:

$$Y = X\beta + \varepsilon, \, \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

where we would have centered the outcome Y. Here we denote by X the design matrix, n the number of observations and $p = 1$ the number of coefficients, $H =$

$X(XX^\intercal)^{-1}X^\intercal$ the hat matrix and $\widehat{\beta} = (XX^\intercal)^{-1}X^\intercal Y$ the OLS estimator of the regression coefficients.

$$\begin{aligned}
\mathbb{V}ar(Y) = YY^\intercal =& YHY^\intercal + Y(1-H)Y^\intercal \\
SST =& SSR + SSE \\
=& \hat{\beta}(XX^\intercal)\hat{\beta}^\intercal + Y(1-H)Y^\intercal \\
=& \sigma^2(\hat{\beta}\Sigma_{\hat{\beta}}^{-1}\hat{\beta}^\intercal + n - p)
\end{aligned}$$

Introducing $MSSR = SSR/1$ and $MSSE = SSE/(n-p)$, we obtain that:

$$\frac{MSSR}{MSSE} = \frac{\hat{\beta}^2}{\Sigma_{\hat{\beta}}} = Wald^2$$

So the F-statistic equals the ratio of the residual sum of squared (normalized by their degrees of freedom). We can check that this extends to multiple regression using the usual anova table:

```
anova(lm(pH ~ Subject + PacO2, data = bland1995))
```

```
Analysis of Variance Table

Response: pH
          Df  Sum Sq Mean Sq F value    Pr(>F)    
Subject    7 2.86484 0.40926  46.600 < 2.2e-16 ***
PacO2      1 0.11532 0.11532  13.131 0.0008471 ***
Residuals 38 0.33373 0.00878                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

⚠ Since **R** output type 1 anova only the last and second to last line are relevant. The first line (`Subject`) is for a model without `PacO2` so it should be expected that the F-value does not match with the one of `Subject` in a model with `PacO2`.

```
sigma2 <- as.double(sigma(e.lmm))
beta <- eTable.lmm["PacO2","estimate"]
sigma_beta <- eTable.lmm["PacO2","se"]
c(MSSE = sigma2, MSSR = sigma2 * beta^2 /sigma_beta^2)
```

```
       MSSE        MSSR 
0.008782435 0.115324959 
```

Finally the R^2 is defined as the proportion of variance explained, i.e.:

$$
\begin{aligned}
R^2 =& \frac{SSR}{SSR + SSE} \\
=& \frac{1}{1 + SSE/SSR} \\
=& \frac{1}{1 + (n-p)/(\beta^2/\sigma^2_\beta)}
\end{aligned}
$$

$$= \frac{Wald^2}{Wald^2 + n - p}$$

So the partial correlation coefficient is the square root of that quantity, with sign the sign of the test statistic:

```
df <- eTable.lmm["PacO2","df"]
sign(Wald)*sqrt(Wald^2/(Wald^2+df))
```

```
[1] -0.5067697
```

which matches exactly the partial correlation coefficient when **both** outcome are adjusted for `Subject`:

```
e.partialCor <- partialCor(list(pH ~ Subject, PacO2 ~ Subject),
      data = bland1995)
print(e.partialCor, digit = 5)
```

```
                Partial correlation

              estimate    se   df lower  upper p.value
rho(pH,PacO2)   -0.507 0.125 25.7 -0.71 -0.225 0.00178

   Note: estimate, standard error, confidence interval have been back-transformed (rho para
```

Similar values can be obtained using dedicated packages, e.g.:

```
library(rmcorr)
rmcorr(Subject, PacO2, pH, bland1995)$r
```

```
[1] -0.5067697
```

4 Partial partial with repeated measurements

There are several references on the subject (Bland and Altman, 1995; Lipsitz et al., 2001; Bakdash and Marusich, 2017; Shan et al., 2020). We will focus on the mixed model approach. The idea is to jointly model the variance and covariance of all measurements under appropriate constrains. For instance denoting one measurement X and the other measurement Y, both indexed by time t, our target parameter may be $\rho = cor(X(t), Y(t))$ (marginal) assumed independent of t while X and Y may or may not be stationnary. Another target parameter could be the correlation between a de-noised version of X and Y, where we have for instance removed individual-specific variations (conditional).

4.1 Illustration

Let's illustrate ρ and r on an example with 3 timepoints, $r = 0.8$, and 250 individuals:

```
n.time <- 3
n.id <- 250
Sigma <- matrix(c(1,0.8,0.8,1),2,2)
Sigma
```

```
     [,1] [,2]
[1,]  1.0  0.8
[2,]  0.8  1.0
```

```
set.seed(11)
df.W <- data.frame(id = unlist(lapply(1:n.id, rep, n.time)),
     time = rep(1:n.time,n.id),
     rmvnorm(n.time*n.id, mean = c(3,3), sigma = Sigma)
     )
head(df.W)
```

We introduce random effects to impose a constant correlation within-individuals for X and for Y:

```
sd.id <- 1.5
df.W$X1 <- df.W$X1 + rnorm(n.id, sd = sd.id/4)[df.W$id]
df.W$X2 <- df.W$X2 + rnorm(n.id, sd = sd.id)[df.W$id]
df.W$id <- as.factor(df.W$id)
df.L <- reshape2::melt(df.W, id.vars = c("id","time"))
df.L$time2 <- as.factor(as.numeric(as.factor(paste(df.L$variable,df.L$time
    ,sep="."))))
```

This will lead to the following correlation structure:

```
Sigma.GS <- as.matrix(bdiag(Sigma,Sigma,Sigma))[c(1,3,5,2,4,6),c
    (1,3,5,2,4,6)]
Sigma.GS[1:3,1:3] <- Sigma.GS[1:3,1:3] + (sd.id/4)^2
Sigma.GS[4:6,4:6] <- Sigma.GS[4:6,4:6] + sd.id^2
cov2cor(Sigma.GS)
```

```
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 1.0000000 0.1232877 0.1232877 0.4155056 0.0000000 0.0000000
[2,] 0.1232877 1.0000000 0.1232877 0.0000000 0.4155056 0.0000000
[3,] 0.1232877 0.1232877 1.0000000 0.0000000 0.0000000 0.4155056
[4,] 0.4155056 0.0000000 0.0000000 1.0000000 0.6923077 0.6923077
[5,] 0.0000000 0.4155056 0.0000000 0.6923077 1.0000000 0.6923077
[6,] 0.0000000 0.0000000 0.4155056 0.6923077 0.6923077 1.0000000
```

We can now estimate two types of correlation: marginal and conditional

```
e.LMMstar <- partialCor(c(X1,X2) ~ 1, repetition = ~ time|id, data = df.W
    , heterogeneous = 0.5)
e.LMMstar
```

```
                Partial correlation


               estimate     se   df lower upper p.value
rho(1.X1,1.X2)    0.427 0.0346 34.8 0.356 0.493 6.5e-13
r(1.X1,1.X2)      0.790 0.0672 66.4 0.656 0.925 0.0e+00
        ----------------------------------------------------
        rho: marginal correlation
        r  : correlation conditional on the individual
```

The conditional is very close to what other packages output:

```
rmcorr(id, X1, X2, df.W)$r
```

```
[1] 0.7941749
```

Here the modeled correlation matrix is:

```
Sigma <- sigma(attr(e.LMMstar,"lmm"))
Rho <- cov2cor(Sigma)
Rho
```

```
            1.X1        2.X1        3.X1        1.X2        2.X2        3.X2
1.X1  1.00000000  0.06545230  0.06545230  0.42652595 -0.00432106 -0.00432106
2.X1  0.06545230  1.00000000  0.06545230 -0.00432106  0.42652595 -0.00432106
3.X1  0.06545230  0.06545230  1.00000000 -0.00432106 -0.00432106  0.42652595
1.X2  0.42652595 -0.00432106 -0.00432106  1.00000000  0.68836567  0.68836567
2.X2 -0.00432106  0.42652595 -0.00432106  0.68836567  1.00000000  0.68836567
3.X2 -0.00432106 -0.00432106  0.42652595  0.68836567  0.68836567  1.00000000
```

From which the conditional correlation can be deduced:

```
Rho[1,4]/sqrt((1-Rho[1,2])*(1-Rho[4,5]))
```

```
[1] 0.7903548
```

or equivalently:

```
Sigma[1,4]/sqrt((Sigma[1,1]-Sigma[1,2])*(Sigma[4,4]-Sigma[4,5]))
```

```
[1] 0.7903548
```

4.2 Simulation study

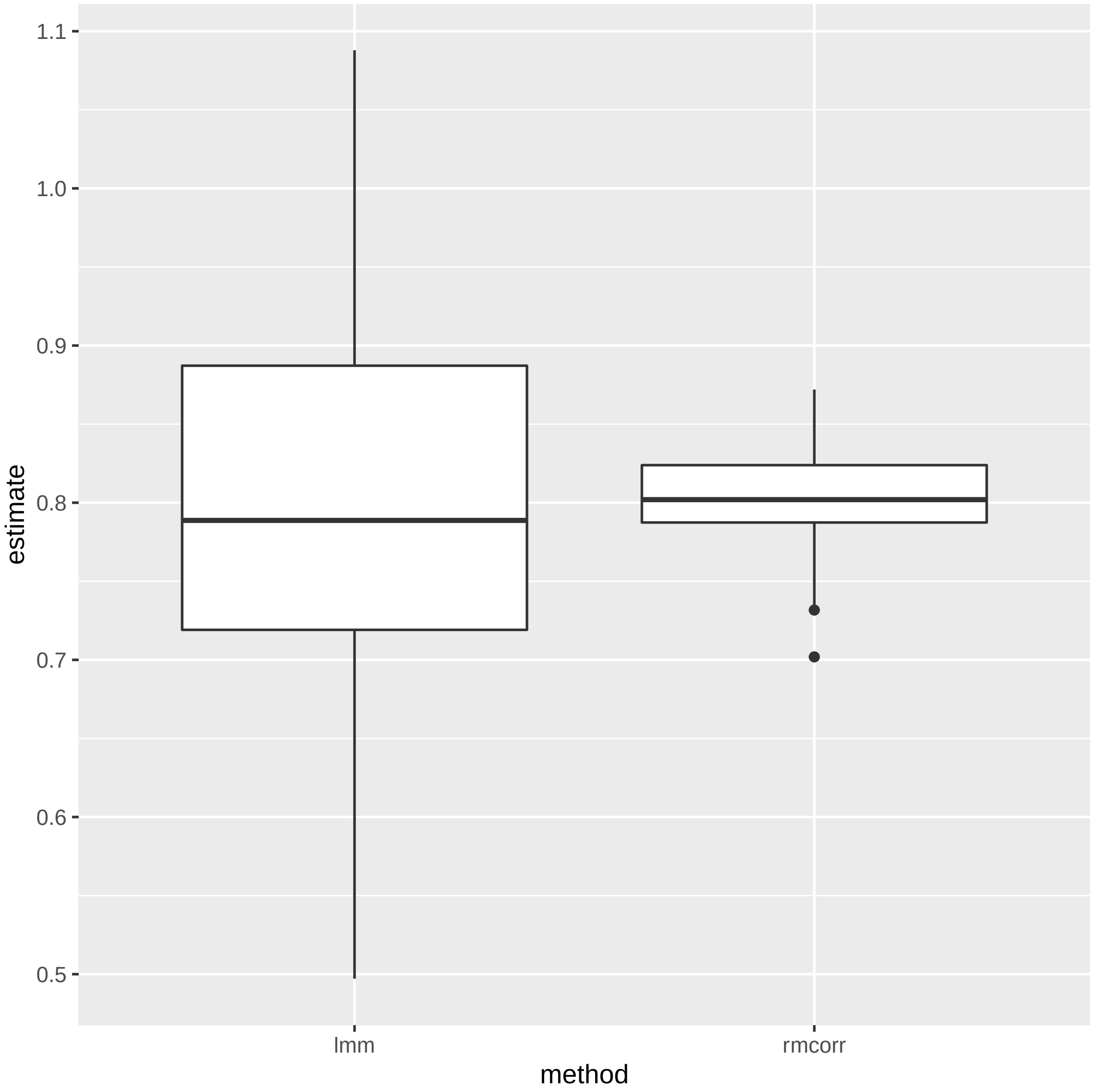
```
library(data.table)

n.id <- 100
warper <- function(n){
  df.W <- data.frame(id = unlist(lapply(1:n.id, rep, n.time)),
         time = rep(1:n.time,n.id),
         rmvnorm(n.time*n.id, mean = c(3,3), sigma = Sigma)
         )
  df.W$X1 <- df.W$X1 + rnorm(n.id, sd = sd.id/4)[df.W$id]
  df.W$X2 <- df.W$X2 + rnorm(n.id, sd = sd.id)[df.W$id]
  df.W$id <- as.factor(df.W$id)

  res1 <- setNames(c(rmcorr(id, X1, X2, df.W)$r, rmcorr(id, X1, X2, df.W)$
    CI), c("estimate","lower","upper"))
  res2 <- partialCor(c(X1,X2) ∼ 1, repetition = ∼ time|id, data = df.W,
    heterogeneous = 0.5)
  return(rbind(cbind(as.data.frame(as.list(res1)), se = NA, method = "
    rmcorr"),
         cbind(res2[2,c("estimate","lower","upper","se")],method="lmm")))
}

ls.res <- pbapply::pblapply(1:101,function(iSim){
  cbind(sim = iSim, warper(100))
})
dt.res <- as.data.table(do.call(rbind, ls.res))
```

```
ggplot(dt.res,aes(x=method,y=estimate))+ geom_boxplot()
```



5 Reference

Bakdash, J. Z. and Marusich, L. R. (2017). Repeated measures correlation. *Frontiers in psychology*, 8:456.

Bland, J. M. and Altman, D. G. (1995). Calculating correlation coefficients with repeated observations: Part 2—correlation between subjects. *Bmj*, 310(6980):633.

Lipsitz, S. R., Leong, T., Ibrahim, J., and Lipshultz, S. (2001). A partial correlation coefficient and coefficient of determination for multivariate normal repeated measures data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 50(1):87–95.

Shan, G., Zhang, H., and Jiang, T. (2020). Correlation coefficients for a study with repeated measures. *Computational and mathematical methods in medicine*, 2020.