Assessing the effect of an exposure on multiple outcomes (with **R** code)

Brice Ozenne

March 8, 2021

Summary

We describe a strategy to assess the effect of an exposure (e.g. a disease, a genetic factor) on several outcomes (e.g. psychological outcomes, the binding potential measured in several brain regions) while accounting for possible risk factors and confounders. This strategy, called *multiple univariage regressions* strategy, models the relationship between each outcome and the exposure using a separate model. Once the models have been correctly fitted, a global test can be used to test whether there is any effect of the exposure on the outcomes. After that, multiple tests are performed to test outcome-specific effects of the exposure where the Dunnett adjustment is used to control the type 1 error (Pipper et al., 2012). An adjustment is used to improved the control of the type 1 error in small sample sizes (e.g. n<100). This adjustment has been shown to beneficial in several settings (using simulation studies) but does not always perfectly control the type 1 error rate. It is advised to check that validity of the adjustment when using very small samples or models with many parameters.

The proposed strategy can be used with any type of outcomes for which a can fit a model with asymptotically linear estimators (Tsiatis (2007), section 3). This includes generalized linear model or Cox models. It makes it very flexible and the strategy makes only few assumptions on the joint distribution. The drawback is that it is not the most efficient approach. For instance modelling the joint distribution of the outcomes, e.g. using a latent variable model / mixed model in the case of normally distributed outcomes, will be a more efficient strategy. Another limitation is that with the proposed approach a treatment effect specific to each outcome will be estimated, while in some context the investigator may want to constrain the treatment effect to be the same for some outcomes. Finally, to be feasible the strategy requires the number of outcomes to be not too large (<100) and smaller than the number of observations (low-dimensional setting).

This document we aim at giving a basic understanding of the strategy and how to implement it. In particular, we don't claim that the proposed strategy is valid or optimal results in every application. We start by simulating some data in section 1. Section 2 is a summary of important aspects in applied statistics. Finally section 3 describe the *multiple univariage regressions* strategy.

1 Simulation of the data

To be able to assess the validity of the proposed strategy, we will use simulated data containing:

- a variable identifying each patient: `Id`
- 10 outcomes per patient: `Y1` to `Y10`.
- 3 possible exposures per patient: `age` that is not related to the outcomes, `BMI` that has the same effect on all outcomes, and `MDI` that has a different effect per outcome.

We use the `lvm` function from the *lava* package to define these variables:

```
m.sim <- lava::lvm(Y1 ~ 0*age + 0.25*BMI + 0.1*MDI + 1*eta,
     Y2[0:2] ~ 0*age + 0.25*BMI + 0.2*MDI + 2*eta,
     Y3 ~ 0*age + 0.25*BMI + 0.15*MDI + 3*eta,
     Y4[0:0.5] ~ 0*age + 0.25*BMI + 0.175*MDI + 1*eta,
     Y5[0:3] ~ 0*age + 0.25*BMI + 0.075*MDI + 2*eta
     )
transform(m.sim, Id ~ eta) <- function(x){paste0("Subj",1:NROW(x))}
categorical(m.sim, labels = c("male","female")) <-  ~ Gender
distribution(m.sim, ~age) <-  gaussian.lvm(mean = 35, sd = 5)
distribution(m.sim, ~BMI) <-  gaussian.lvm(mean = 22, sd = 3)
distribution(m.sim, ~MDI) <-  gaussian.lvm(mean = 20, sd = 5)
latent(m.sim) <- ~eta
```

From the code above we can see that the variance of the outcomes differs between outcomes and that the correlation between pairs of outcomes is also variable. We now simulate data using `lava::sim`:

```
set.seed(10)
dfW <- lava::sim(m.sim, n = 50, latent = FALSE)
```

We round the values to 2 digits:

```
digit.cols <- c("age","BMI","MDI",paste0("Y",1:5))
dfW[,digit.cols] <- round(dfW[,digit.cols],2)
```

and re-order its columns:

```
dfW <- dfW[,c("Id","Gender",digit.cols)]
```

We can now display first lines of the dataset:

```
head(dfW)
```

```
     Id Gender   age   BMI   MDI   Y1    Y2    Y3    Y4   Y5
1 Subj1 female 30.57 21.76 25.82 7.64  8.73  7.72 10.42 8.44
2 Subj2 female 41.36 25.55 12.38 7.11  8.79  6.99  8.45 8.26
3 Subj3   male 26.97 28.56  7.41 7.88  9.89 13.51 10.79 7.90
4 Subj4 female 40.61 23.22 16.46 8.99 14.38 13.82 11.44 9.75
5 Subj5 female 45.79 19.78 18.56 7.60  8.77  8.38  7.94 6.17
6 Subj6 female 37.14 16.13 17.82 6.99  9.97  6.74  8.29 8.78
```

2 Statistics: definitions and notations

2.1 Variables

We can differentiate several types of random variables: outcomes, exposure, risk factors, confounders, and mediators. To explicit the difference between these types of variables we consider a set of random variables (Y, E, X_1, X_2, M) whose relationships are displayed on Figure 1:

- **outcome** (Y): random variables that are observed with noise. It can be for instance the 5HT-4 binding in a specific brain region. When considering several outcomes we will denote in bold variable that stands for a vector of random variables: $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$. This happens for instance when studying the binding in several brain regions. In such a case we expect the outcomes to be correlated.

- **exposure** (E): a variable that may affect the outcome or be associated with the outcome *and* we are interested in studying this effect/association. It can for instance be a genetic factor that is hypothesized to increase the 5HT-4 binding, or a disease like depression that is associated with a change in binding (we don't know whether one causes the other or whether they have a common cause, e.g. a genetic variant).

- **risk factor/confounder** (X_1, X_2): a variable that may affect the outcome or be associated with the outcomes *but* we are *not* interested in studying their effect/association. Risk factors (denoted by X_1) are only associated with the outcomes and confounders that are both associated with the outcome and the exposure. We usually need to account for confounders the statistical model in order to obtain unbiased estimates while accounting for risk factors only enables to obtain more precise estimates (at least in linear models).

- **mediator** (M): a variable that modulate the effect of the exposure, i.e. stands on the causal pathway between the exposure and the outcome. For instance, the permeability of the blood-brain barrier may modulate the response to drugs and can act as a mediator. It is important to keep in mind that when we are interested in the (total) effect of E on Y, we should *not* adjust the analysis on M[1]. Doing so we would remove the effect of E mediated by M and therefore bias the estimate of the total effect (we would only get the direct effect).

In the following we will assume that we do not measure any mediator variable and therefore ignore this type of variable. Also we will call **covariates** the variables E, X_1, X_2.

[1] This may not be true in specific types of confounding but we will ignore that.

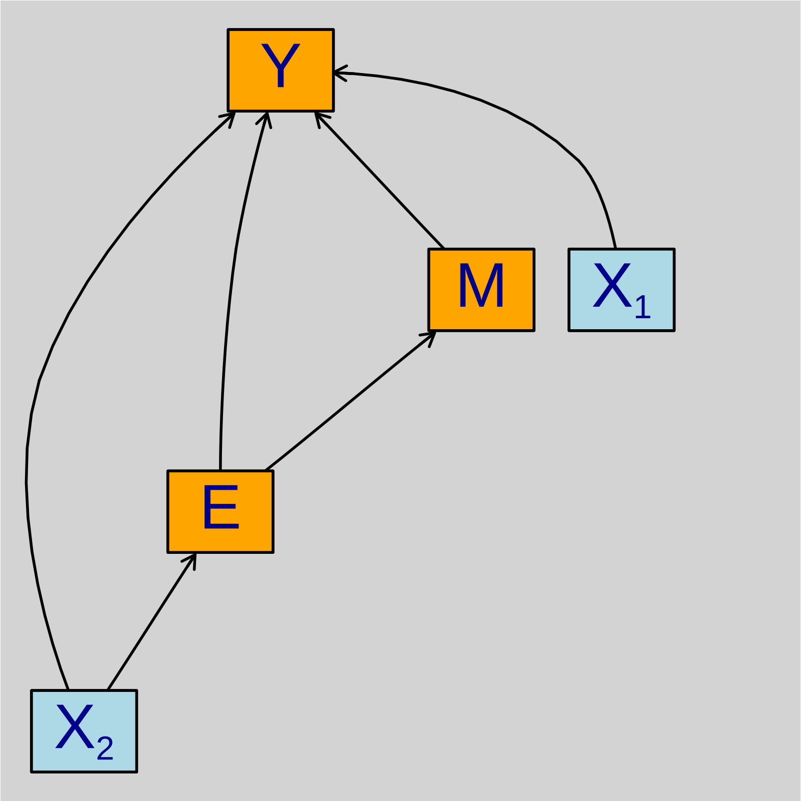

Figure 1: Path diagram relating the variables Y, E, M, X_1 and X_2

2.2 Assumptions

We can distinguish two types of assumptions:

- **causal assumptions**: saying which variables are related and in which direction. This can be done by drawing a path diagram similar to Figure 1. In simple univariate models it may seems unnecessary to draw the path diagram since the system of variables is very simple to visualize. In multivariate model, it is often very useful to draw it. Some of these assumptions are untestable, e.g. often we cannot decide whether it is E that impacts Y or whether it is Y that impacts E just based on the data.

- **modeling assumptions**: specifying the type of relationship between variables (e.g. linear) and the marginal or joint distribution (e.g. Gaussian). Often these assumptions can be tested and relaxed using a more flexible model. While appealing, there are some drawbacks with using a very flexible model: more data are needed to get precise estimates and the interpretation of the results is more complex.

2.3 Statistical model

A statistical model $\mathcal{M}$ is set of possible probability distributions. For instance when we fit a Gaussian linear model for Y_1 with just an intercept $\mathcal{M} = \{\mathcal{N}\left(\mu, \sigma^2\right); \mu \in \mathbb{R},\ \sigma^2 \in \mathbb{R}^+\}$: $\mathcal{M}$ is the set containing all possible univariate normal distributions.

2.4 Model parameters

The model parameters are the (non random) variables that enable the statistical model to "adapt" to different settings. They will be denoted Θ. They are the one that are estimated when we fit the statistical model using the data or that we specify when we simulate data. In the previous example, we could simulate data corresponding to a Gaussian linear model using the `rnorm` function in R:

```
rnorm
```

```
function (n, mean = 0, sd = 1)
.Call(C_rnorm, n, mean, sd)
<bytecode: 0x88d8050>
<environment: namespace:stats>
```

We would need to specify:

- n the sample size

- $\Theta = (\mu, \sigma^2)$ the model parameters, here μ corresponds to `mean` and σ to `sd`.

The true model parameters are the model parameters that have generated the observed data. They will be denoted Θ_0. For instance if in reality the binding potential is normally distributed with mean 5 and variance $2^2 = 4$, then $\Theta_0 = (\mu_0, \sigma_0^2) = (5, 4)$. Then doing our experiment we observed data such as:

```
set.seed(10)
Y_1.XP1 <- rnorm(10, mean = 5, sd = 2)
Y_1.XP1
```

```
[1] 5.037492 4.631495 2.257339 3.801665 5.589090 5.779589 2.583848 4.272648 1.746655 4.4870
```

If we were to re-do the experiment we would observe new data but Θ_0 would not change:

```
Y_1.XP2 <- rnorm(10, mean = 5, sd = 2)
Y_1.XP2
```

```
[1] 7.203559 6.511563 4.523533 6.974889 6.482780 5.178695 3.090112 4.609699 6.851043 5.9659
```

The estimated parameters are the parameters that we estimate when we fit the statistical model. They will be denoted $\hat{\Theta}$. We usually try to find parameters whose value maximize the chance of simulating the observed data under the estimated model (maximum likelihood estimation, MLE). For instance in the first experiment all values are positive so we would not estimate a negative mean value. In our example, $\hat{\mu}$ the MLE of μ reduces to the empirical average and $\hat{\sigma}^2$ the MLE of σ^2 to the empirical variance:

```
Theta_hat.XP1 <- c(mu_hat = mean(Y_1.XP1),
     sigma2_hat = var(Y_1.XP1))
Theta_hat.XP1
```

```
  mu_hat sigma2_hat
4.018686   1.959404
```

Clearly the estimated coefficients vary across experiments:

```
Theta_hat.XP2 <- c(mu_hat = mean(Y_1.XP2),
     sigma2_hat = var(Y_1.XP2))
Theta_hat.XP2
```

```
  mu_hat sigma2_hat
5.739183   1.799311
```

2.5 Parameter of interest

The statistical model may contain many parameters, most of them are often not of interest but are needed to obtain valid estimates (e.g. account for confounders). In most settings, the parameter of interest is one (or several) model parameter(s) - or simple transformation of them. For instance if we are interested in the average binding potential in the population our parameter of interest is μ.

Often, the aim of a study is to obtain the best estimate of the parameter of interest μ. Best means:

- **unbiased**: if we were able to replicate the study many times, i.e. get several estimates $\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_K$, the average estimate $< \hat{\mu} >= \frac{\hat{\mu}_1+\hat{\mu}_2+...+\hat{\mu}_K}{K}$ would coincide with the true one μ_0.
- **minimal variance**: if we were able to replicate the study many times, the variance of the estimates $\frac{(\hat{\mu}_1-<\hat{\mu}>)^2+...+(\hat{\mu}_K-<\hat{\mu}>)^2}{K-1}$ should be as low as possible.

There will often be a trade-off between these two objectives. A very flexible method is more likely to give an unbiased estimate (e.g. being able to model non-linear relationship) at the price of greater uncertainty about the estimates. Often we favor unbiasedness over minimal variance. Indeed, if several studies are published with the same parameter of interest, one can pool the results to obtain an estimate with lower variance. Note that we have no guarantee that it will reduce the bias.

2.6 Contrast matrix

Consider a linear model:

```
e.lm <- lm(Y1 ~ Gender + age + MDI, data = dfW)
e.lm
```

```
Call:
lm(formula = Y1 ~ Gender + age + MDI, data = dfW)

Coefficients:
 (Intercept)  Genderfemale           age           MDI  
     4.77626      -0.02049      -0.02030       0.16403  
```

Denote for the $i-th$ patient its outcome value by Y_i (can be any real number), its gender value by $Gender_i$ (can be "Male" or "Female"), its age value by age_i (can be 60, 35, or 26), and its BMI value by BMI_i. Mathematically, this linear model can be written:

$$Y_i = \alpha + \beta_{Gender} * \mathbb{1}_{Gender_i="Female"} + \beta_{Age} * Age_i + \beta_{MDI} * MDI_i + \varepsilon_i$$

When dealing with many parameters it is convenient to define the null hypothesis via a contrast matrix. An example of null hypothesis is:

$$(\mathcal{H}_0)\ \beta_{MDI,0} = 0$$

If we consider $\Theta = (\alpha, \beta_{Gender}, \beta_{age}, \beta_{MDI})$, this null hypothesis can be equivalently written:

$$c = [0\ 0\ 0\ 1]$$

such that:

$$(\mathcal{H}_0)\ c\Theta_0^\intercal = 0$$

Indeed

$$c\Theta_0^\intercal = 0 * \alpha_0 + 0 * \beta_{Gender,0} + 0 * \beta_{age,0} + 1 * \beta_{MDI,0} = \beta_{MDI,0}$$

An example where the contrast matrix is useful is

- when one wish to test linear combination of parameters, e.g. consider the null hypothesis where the added risk when being a female instead of a male is the same as being 5 years older:

$$(\mathcal{H}_0)\ 5\beta_{age,0} = \beta_{Gender,0}$$

Here the contrast matrix would be:

$$c = [0\ 5\ \ -1\ 0]$$

- when one wish to test several hypotheses simultaneously, e.g. consider the null hypothesis:

$$(\mathcal{H}_0)\ \beta_{age,0} = 0 \text{ or } \beta_{MDI,0} = 0$$

Here the contrast matrix would be:

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In **R**, the method `createContrast` from the *lavaSearch2* package helps to define the contrast matrix:

```
library(lavaSearch2)
Clin <- createContrast(e.lm, par = c("5*age - Genderfemale = 0"),
         add.variance = FALSE, rowname.rhs = FALSE)
Clin$contrast
```

```
                     (Intercept) Genderfemale age MDI
Genderfemale - 5*age           0            1  -5   0
```

```
Csim <- createContrast(e.lm, par = c("age = 0","MDI = 0"),
         add.variance = FALSE, rowname.rhs = FALSE)
Csim$contrast
```

```
    (Intercept) Genderfemale age MDI
age           0            0   1   0
MDI           0            0   0   1
```

Then the contrast matrix can be send to the function `glht` from the *multcomp* package to obtain p-values and confidence intervals:

```
library(multcomp)
elin.glht <- glht(e.lm, linfct = Clin$contrast)
summary(elin.glht)
```

```

         Simultaneous Tests for General Linear Hypotheses

Fit: lm(formula = Y1 ~ Gender + age + MDI, data = dfW)

Linear Hypotheses:
                          Estimate Std. Error t value Pr(>|t|)
Genderfemale - 5*age == 0   0.0810     0.5364   0.151    0.881
(Adjusted p values reported -- single-step method)
```

```
esim.glht <- glht(e.lm, linfct = Csim$contrast)
summary(esim.glht)
```

```

         Simultaneous Tests for General Linear Hypotheses

Fit: lm(formula = Y1 ~ Gender + age + MDI, data = dfW)

Linear Hypotheses:
         Estimate Std. Error t value Pr(>|t|)
age == 0 -0.02030    0.04250  -0.478  0.86315
MDI == 0  0.16403    0.04051   4.049  0.00039 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

3 Multivariate analysis using multiple univariate linear regressions

We want to simultaneously test the effect of MDI on five outcomes. To achieve it, we fit separately for each outcome a univariate linear regression. Mathematically the model can be written:

$$\begin{bmatrix} Y_1 & = \alpha_{Y_1} + \beta_{Y_1,age} age + \beta_{Y_1,BMI} BMI + \beta_{Y_1,MDI} MDI + \varepsilon_{Y_1} \\ Y_2 & = \alpha_{Y_2} + \beta_{Y_2,age} age + \beta_{Y_2,BMI} BMI + \beta_{Y_2,MDI} MDI + \varepsilon_{Y_2} \\ Y_3 & = \alpha_{Y_3} + \beta_{Y_3,age} age + \beta_{Y_3,BMI} BMI + \beta_{Y_3,MDI} MDI + \varepsilon_{Y_3} \\ Y_4 & = \alpha_{Y_4} + \beta_{Y_4,age} age + \beta_{Y_4,BMI} BMI + \beta_{Y_4,MDI} MDI + \varepsilon_{Y_4} \\ Y_5 & = \alpha_{Y_5} + \beta_{Y_5,age} age + \beta_{Y_5,BMI} BMI + \beta_{Y_5,MDI} MDI + \varepsilon_{Y_5} \end{bmatrix}$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5$ are the residual errors. The residuals are assumed to have zero mean and finite variance, respectively, $\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2$. Here we make no assumption on the correlation structure between the residuals.

3.1 Fitting multiple linear regression in R

We can estimate all the 5 models and store them into a list:

```
ls.lm <- list(Y1 = lm(Y1 ~ age + BMI + MDI, data = dfW),
       Y2 = lm(Y2 ~ age + BMI + MDI, data = dfW),
       Y3 = lm(Y3 ~ age + BMI + MDI, data = dfW),
       Y4 = lm(Y4 ~ age + BMI + MDI, data = dfW),
       Y5 = lm(Y5 ~ age + BMI + MDI, data = dfW)
       )
```

3.2 Interpretation of the regression coefficients

Same as in the univariate case (see https://bozenne.github.io/doc/2020-09-17-linearModel/post-linearModel.pdf).

3.3 Diagnostics tools for univariate linear regression in R

Same as in the univariate case (see https://bozenne.github.io/doc/2020-09-17-linearModel/post-linearModel.pdf). Model checking needs to be done for each outcome.

3.4 Hypothesis testing

We now want to test:

$(\mathcal{H}_0)\ \beta_{Y_1,MDI,0} = 0$ and $\beta_{Y_2,MDI,0} = 0$ and $\beta_{Y_3,MDI,0} = 0$ and $\beta_{Y_4,MDI,0} = 0$ and $\beta_{Y_5,MDI,0} = 0$

The p-values returned by `summary` are no more valid since we are performing multiple tests (here 5 tests). A basic solution would be to collect the p-values:

```
vec.p.value <- unlist(lapply(ls.lm, function(x){
    summary(x)$coef["MDI","Pr(>|t|)"]
}))
```

and adjust them for multiple comparisons using Bonferroni:

```
p.adjust(vec.p.value, method = "bonferroni")
```

```
          Y1           Y2           Y3           Y4           Y5 
3.299432e-04 4.218369e-02 3.552579e-01 2.276690e-07 8.565878e-01 
```

While easy to use this approach tends to be too conservative (i.e. give to large p-values) when the test statistics are correlated. This is usually the case when the outcomes are correlated. We will therefore use a more efficient correction called the Dunnett approach. First we need to define the null hypothesis that we want to test via a contrast matrix. For simple null hypotheses like the one we are considering in this example, we can use the function `createContrast` that will create the matrix for us:

```
resC <- createContrast(ls.lm, var.test = "MDI", add.variance = TRUE)
```

This function defines for each model the appropriate contrast matrix:

```
resC$mlf
```

```
$Y1
    (Intercept) age BMI MDI sigma2
MDI           0   0   0   1      0

$Y2
    (Intercept) age BMI MDI sigma2
MDI           0   0   0   1      0

$Y3
    (Intercept) age BMI MDI sigma2
MDI           0   0   0   1      0

$Y4
```

```
    (Intercept) age BMI MDI sigma2
MDI           0   0   0   1      0

$Y5
    (Intercept) age BMI MDI sigma2
MDI           0   0   0   1      0

attr(,"class")
[1] "mlf"
```

and right hand side of the null hypothesis:

```
resC$null
```

```
Y1: MDI Y2: MDI Y3: MDI Y4: MDI Y5: MDI
      0       0       0       0       0
```

We will now call `glht2` to perform the adjustment for multiple comparisons but first we need to convert the list into a `mmm` object:

```
class(ls.lm) <- "mmm"
e.glht_lm <- glht2(ls.lm, linfct = resC$contrast, rhs = resC$null)
e.glht_lm
```

```
         General Linear Hypotheses

Linear Hypotheses:
             Estimate
Y1: MDI == 0  0.15104
Y2: MDI == 0  0.16770
Y3: MDI == 0  0.14907
Y4: MDI == 0  0.19860
Y5: MDI == 0  0.09806
```

We can now correct for multiple comparisons using the (single-step) Dunnett approach:

```
summary(e.glht_lm, test = adjusted("single-step"))
```

```
         Simultaneous Tests for General Linear Hypotheses

Linear Hypotheses:
             Estimate Std. Error t value Pr(>|t|)
Y1: MDI == 0  0.15104    0.03441   4.389   <0.001 ***
Y2: MDI == 0  0.16770    0.06093   2.752   0.0286 *
```

```
Y3: MDI == 0  0.14907    0.08067   1.848   0.1996
Y4: MDI == 0  0.19860    0.03039   6.535   <0.001 ***
Y5: MDI == 0  0.09806    0.07057   1.390   0.4208
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
(Adjusted p values reported -- single-step method)
```

Note that the p-value for the global test equals to the smallest p-value. This means that we reject the global null hypothesis whenever we reject the null hypothesis for any of the outcome (after adjustment for multiple comparisons!).

For comparison one can change the argument in `adjust` to apply the Bonferroni adjustment:

```
summary(e.glht_lm, test = adjusted("bonferroni"))
```

```
         Simultaneous Tests for General Linear Hypotheses

Linear Hypotheses:
             Estimate Std. Error t value Pr(>|t|)
Y1: MDI == 0  0.15104    0.03441   4.389  0.00033 ***
Y2: MDI == 0  0.16770    0.06093   2.752  0.04218 *
Y3: MDI == 0  0.14907    0.08067   1.848  0.35526
Y4: MDI == 0  0.19860    0.03039   6.535 2.28e-07 ***
Y5: MDI == 0  0.09806    0.07057   1.390  0.85659
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
(Adjusted p values reported -- bonferroni method)
```

Finally, confidence intervals can be obtained using the `confint` function:

```
confint(e.glht_lm)
```

```
         Simultaneous Confidence Intervals

Fit: NULL

Quantile = 2.5215
95% family-wise confidence level


Linear Hypotheses:
             Estimate lwr      upr
Y1: MDI == 0  0.15104  0.06427  0.23782
Y2: MDI == 0  0.16770  0.01407  0.32133
Y3: MDI == 0  0.14907 -0.05434  0.35248
Y4: MDI == 0  0.19860  0.12197  0.27524
Y5: MDI == 0  0.09806 -0.07987  0.27599
```

Note that by default the `confint` function output confidence intervals using the (single-step) Dunnett approach.

4 References

Pipper, C. B., Ritz, C., and Bisgaard, H. (2012). A versatile method for confirmatory evaluation of the effects of a covariate in multiple models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(2):315–326.

Tsiatis, A. (2007). *Semiparametric theory and missing data*. Springer Science & Business Media.