General Linear Model

André Meichtry

Contents

```
library(psych)
```

Analysis of covariance

We have looked at linear models with **categorical independent variables**, often called ANOVA. Remember though that we looked at ANOVA from a more global perspective, the perspective of model comparison.

Now, we look at a problem where we have one **continuous predictor** in addition to a categorical predictor. Such models are also called **ANCOVA or analysis of covariance**.

In our journey, an ANCOVA is (just) one further special case of the General linear Model (LM) which allows multiple categorical and continuous predictors.

Statistical model

Remember that a model equation includes fixed and random quantities,

$$\underbrace{Y_i}_{response} = \underbrace{\mu_i}_{deterministic} + \underbrace{\epsilon_i}_{stochastic}, \quad i = 1, \ldots, n.$$

Assume we have

- one categorical predictor **group** with 2 values A and B
- one **continuous** predictor X.
- We then need 4 (5) parameters: **intercept** and **slope** for each group plus error standard deviation.
- The **Effects parameterization** of the interaction-effects model is

$Y_i = \underbrace{\beta_1 + \beta_2 I_{B(i)} + \beta_3 \, x_i + \beta_4 \, x_i \, I_{B(i)}}_{\mu_i} + \epsilon_i$ with indicator variable

$$I_{B(i)} = \begin{cases} 0, & B(i) = A \\ 1, & B(i) = B \end{cases}$$

The expectations ("Long run average"):

- $E(Y_i; group_i = A, X_i = 0) = \beta_1$
- $E(Y_i; group_i = B, X_i = 0) = \beta_1 + \beta_2$
- $E(Y_i; group_i = A, X_i = x_i) = \beta_1 + \beta_3 x_i$
- $E(Y_i; group_i = B, X_i = x_i) = \beta_1 + \beta_2 + (\beta_3 + \beta_4) x_i$

The quantity of interest are the parameters of the model, i.e. β_4, the **difference in slopes between groups**.

Simulation of data*

It is not mandatory to understand the data simulation code. But there is considerable conceptual value in studying the code if you have time.

```
set.seed(10)
n.groups <- 2
n.sample <- 30
n <- n.groups*n.sample ##sample size
ind <- rep(1:n.groups, each=n.sample) ##Indicator for group
group <- factor(ind, labels = c("A", "B"))
height <- rnorm(n, mean=165, sd=11.4)
covariates<-data.frame(group,height)
Xeffects <- model.matrix(~group*height)
Xmeans <- model.matrix(~group*height-height-1)
sigma <-2
betaMeans <- c(muA<--36.475,muB<--45.5,slopeA<-0.615,slopeB<-0.7)
betaEffects <- c(muA,muB-muA,slopeA,slopeB-slopeA)
lin.pred <- Xeffects %*% betaEffects
lin.pred2 <- Xmeans %*% betaMeans
#all.equal(lin.pred,lin.pred2) ## should be same of course
eps <- rnorm(n = n, mean = 0, sd = sigma) ## add noise
weight <- lin.pred + eps ## response
df <- data.frame(group,height,weight)
```

Data

```
str(df)
```

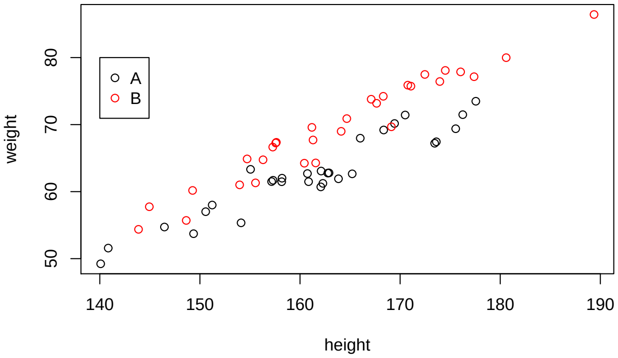
```
'data.frame':   60 obs. of  3 variables:
 $ group : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
 $ height: num  165 163 149 158 168 ...
 $ weight: num  62.7 62.8 53.7 61.5 69.2 ...
```

```
headTail(df)
```

```
    group height weight
1       A 165.21  62.66
2       A  162.9   62.8
```

```
3        A 149.37  53.72
4        A 158.17  61.48
...  <NA>     ...    ...
57       B 154.71  64.88
58       B 171.08  75.74
59       B 157.64  67.36
60       B 168.32  74.22
```

```
plot(weight~height,data=df,col=as.numeric(group))
legend(140,80,legend=levels(group),col=c(1,2),pch=21)
```



Analysis

Description

```
by(df[,-1],df$group,describe)
```

```
df$group: A
       vars  n mean   sd median trimmed mad min max range  skew kurtosis  se
height    1 30  161 9.9    162     161 9.9 140 178    37 -0.28    -0.60 1.8
weight    2 30   63 6.0     62      63 6.8  49  73    24 -0.21    -0.57 1.1
------------------------------------------------------------------------------------------
df$group: B
       vars  n mean   sd median trimmed  mad min max range  skew kurtosis  se
height    1 30  164 10.8    163     164 11.3 144 189    46  0.16    -0.53 2.0
weight    2 30   69  7.8     69      70  8.5  54  86    32 -0.04    -0.75 1.4
```

```
cor(weight,height,method="pearson")
```

```
     [,1]
[1,] 0.91
```

```
cor(weight,height,method="spearman")
```

```
     [,1]
[1,] 0.88
```

```
by(df[,-1],df$group,cor)
```

```
df$group: A
       height weight
height   1.00   0.95
weight   0.95   1.00
------------------------------------------------------------------------------------------
```

```
df$group: B
       height weight
height   1.00   0.97
weight   0.97   1.00
```

Fitting a linear model

We now fit a linear model to the data, of course again with `lm()`

```
mod<-lm(weight~group*height,df)
mod
```

```

Call:
lm(formula = weight ~ group * height, data = df)

Coefficients:
  (Intercept)         groupB         height  groupB:height  
      -30.329        -14.531          0.577          0.121  
```

The output show the point estimates $\hat{\beta}$. For further information, we already know the `summary()` function.

```
summary(mod)
```

```

Call:
lm(formula = weight ~ group * height, data = df)

Residuals:
   Min     1Q Median     3Q    Max 
-3.649 -1.546  0.413  1.420  4.236 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -30.3289     5.8828   -5.16  3.4e-06
groupB        -14.5310     8.0328   -1.81    0.076
height          0.5767     0.0365   15.82  < 2e-16
groupB:height   0.1214     0.0494    2.46    0.017

Residual standard error: 1.9 on 56 degrees of freedom
Multiple R-squared:  0.94,  Adjusted R-squared:  0.937 
F-statistic:  292 on 3 and 56 DF,  p-value: <2e-16
```

Note that the term **residual standard error** is a misnomer with a long tradition, since *standard error* for an estimated parameter θ usually means $\sqrt{Var(\hat{\theta})}$. The correct term would be **residual standard deviation**.

The true beta values and the true sigma are

```
betaEffects
```

```
[1] -36.475  -9.025   0.615   0.085
```

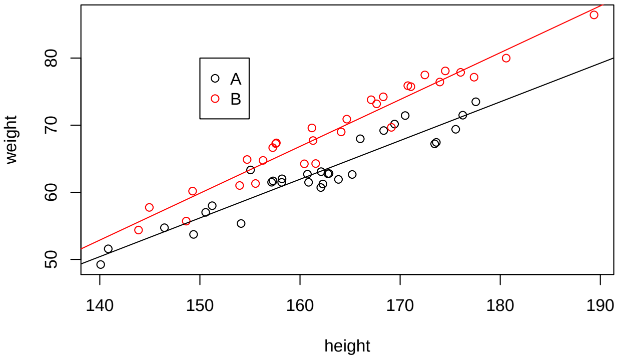
```
sigma
```

```
[1] 2
```

Of course, in real life, you will not know the true values.

We can now add the model fit to the data:

```r
plot(weight~height,data=df,col=as.numeric(group))
legend(150,80,legend=levels(group),col=c(1,2),pch=21)
abline(mod$coef[1],mod$coef[3],col=1)
abline(mod$coef[1]+mod$coef[2],mod$coef[3]+mod$coef[4],col=2)
```



Let us construct confidence intervals for the true effects.

```r
confint(mod,level=0.95)
```

```
                 2.5 % 97.5 %
(Intercept)    -42.114 -18.54
groupB         -30.623   1.56
height           0.504   0.65
groupB:height    0.022   0.22
```

β will be overlapped by a $(1-\alpha)$-CI with a “long-run-probability” of $(1-\alpha)$.

If we want a enhanced output, we can use the `kable()` function:

```r
knitr::kable(cbind(summary(mod)$coef,confint(mod)),digits=3)
```

	Estimate	Std. Error	t value	Pr(>|t|)	2.5 %	97.5 %
(Intercept)	-30.33	5.883	-5.2	0.000	-42.114	-18.54
groupB	-14.53	8.033	-1.8	0.076	-30.623	1.56
height	0.58	0.036	15.8	0.000	0.504	0.65
groupB:height	0.12	0.049	2.5	0.017	0.022	0.22

and for the `anova()` results

```r
knitr::kable(anova(mod),digits=3)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
group	1	708	708.0	188	0.000
height	1	2569	2569.4	683	0.000
group:height	1	23	22.7	6	0.017
Residuals	56	211	3.8	NA	NA

Interpretation

The estimated **increase in weight by unit change on height** is larger in group B than in group A, the **difference in slopes** is $\hat{\beta} = 0.1214138$ (with 95% CI: 0.0224471, 0.2203805). We can reject the null model of no-interaction.

Principle of marginality

Take care: F-tests in R (`anova()` and `aov()`) are **sequential** (so-called Type I sum of squares), the order the terms enter the model does matter! The t-tests in `lm()` on the contrary are marginal (impact of the variables, given the presence of **all the other variables** in the model).

Note the difference between the summary output and the anova output. In the former, we see marginal tests, in the latter, we see – the correct – sequential tests. Sequential tests (also called Type I tests) respect the the principle of **marginality**. As example, it makes no sense that a main effect is controlled for the interaction effect. **Do not interpret main effects in the presence of interaction effects!**

If we wanted to reproduce the senseless p-value for the group main effect in the summary output (0.0758265), we compare the interaction model with the same model without the main effect of group:

```
modNoMainG<-lm(weight~height*group-group) #equals weight~height+height:group
anova(mod,modNoMainG)
```

```
Analysis of Variance Table

Model 1: weight ~ group * height
Model 2: weight ~ height * group - group
  Res.Df RSS Df Sum of Sq    F Pr(>F)
1     56 211
2     57 223 -1     -12.3 3.27  0.076
```

Marginal tests (often used in SPSS, so called Type III tests) are implemented also in the `Anova()` function of the `car` package, but **in general, you are more safe with sequential tests**.

```
car::Anova(mod,type=3)
```

```
Anova Table (Type III tests)

Response: weight
             Sum Sq Df F value  Pr(>F)
(Intercept)     100  1   26.58 3.4e-06
group            12  1    3.27   0.076
height          942  1  250.22 < 2e-16
group:height     23  1    6.04   0.017
Residuals       211 56
```

Global F-test

The global F-test in the summary output tells us if the models explains anything at all, this can be reproduced by

```
mod0<-lm(weight~1)
anova(mod0,mod)
```

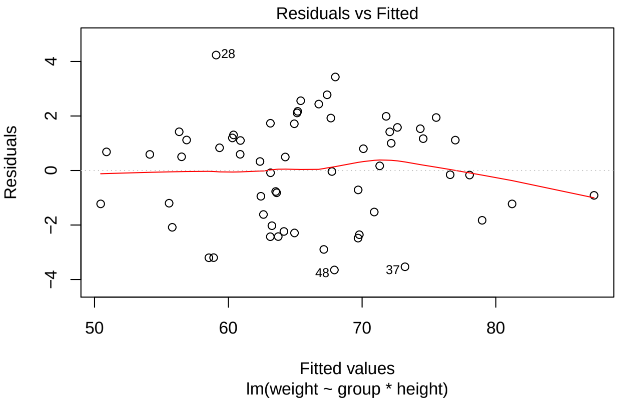
```
Analysis of Variance Table

Model 1: weight ~ 1
Model 2: weight ~ group * height
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1     59 3511
```

```
2      56   211   3      3300 292 <2e-16
```

Residual analysis

```
plot(mod,which=1)
```



Prediction

At the end of the day, we also want to predict new observations from a fitted model. This is implemented in the `predict()` function. There are two kinds of predictions, mean predictions or individual predictions.

The difference between **confidence bound for the expected value** given predictor value versus **prediction bound for future observation** given predictor value is visualized in shinyApp.

Assume we want to predict new observations for different arbitrary combinations on group and height:

```
new<-data.frame(group=c("A","B","A"),height=c(170,180,190))
new
```

```
  group height
1     A    170
2     B    180
3     A    190
```

- Uncertainty for new observation $Y_{new} \mid X_{new} = x_{new}$

```
pred<-predict(mod,newdata=new,interval="prediction")
cbind(new,pred)
```

```
  group height fit lwr upr
1     A    170  68  64  72
2     B    180  81  77  85
3     A    190  79  75  84
```

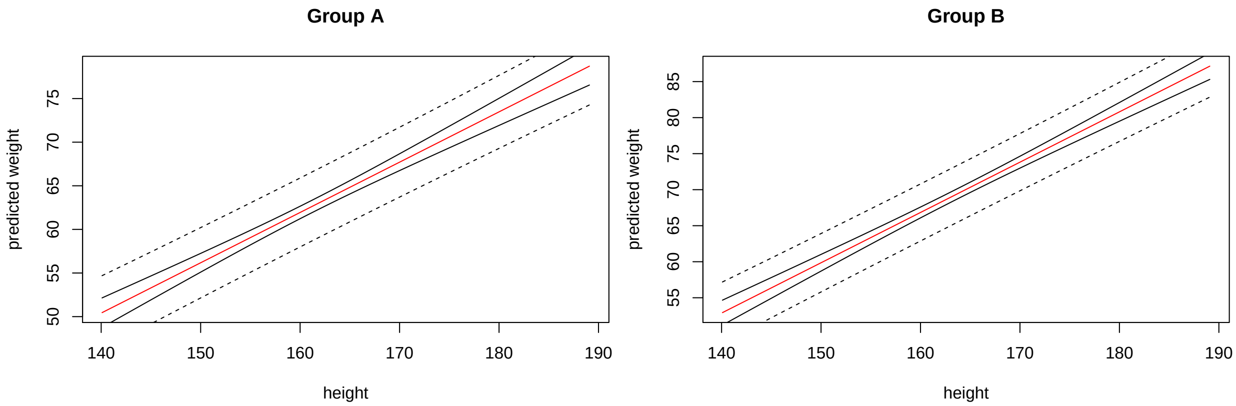
- Uncertainty for the conditional mean $E(Y_{new} \mid X_{new} = x_{new})$

```
pred2<-predict(mod,newdata=new,interval="confidence")
cbind(new,pred2)
```

```
  group height fit lwr upr
1     A    170  68  67  69
2     B    180  81  79  82
3     A    190  79  77  81
```

Let us draw confidence and prediction bounds for each possible length and for each group:

```
pred.frame<-data.frame(group="A",height=seq(min(df$height),max(df$height)))
pc<-data.frame(predict(mod,newdata=pred.frame,interval="confidence"))
pp<-data.frame(predict(mod,newdata=pred.frame,interval="prediction"))
plot(pred.frame$height,pc[,1],col=2,type="l",xlab="height",ylab="predicted weight",main="Group A")
lines(pred.frame$height,pc[,2])
lines(pred.frame$height,pc[,3])
lines(pred.frame$height,pp[,2],lty=2)
lines(pred.frame$height,pp[,3],lty=2)
pred.frame<-data.frame(group="B",height=seq(min(df$height),max(df$height)))
pc<-data.frame(predict(mod,newdata=pred.frame,interval="confidence"))
pp<-data.frame(predict(mod,newdata=pred.frame,interval="prediction"))
plot(pred.frame$height,pc[,1],col=2,type="l",xlab="height",ylab="predicted weight",main="Group B")
lines(pred.frame$height,pc[,2])
lines(pred.frame$height,pc[,3])
lines(pred.frame$height,pp[,2],lty=2)
lines(pred.frame$height,pp[,3],lty=2)
```



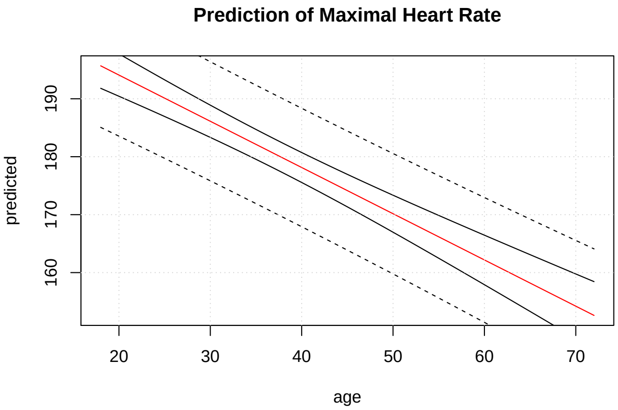
Of course, predictions for a new observation (dashed lines) have more uncertainty (larger prediction bounds) than predictions of an expected value (solid lines, called confidence bounds).

Uncertainty of predictions can be (very) large. In statistics, it is crucial to quantify the corresponding uncertainty. We have seen in this section that we cannot only quantify uncertainty for estimates of parameters β, but also for future unobserved responses $\hat{Y}$!

As example, assume a study with running athletes to predict Maximal Heart Rate as a function of age with data.

```
Age <- c(18,23,25,35,65,54,34,56,72,19,23,42,18,39,37)
HR <- c(202,186,187,180,156,169,174,172,153,199,193,174,198,183,178)
```

```
modHR <- lm(HR~Age)
pred.frame<-data.frame(Age=18:72)
pp<-predict(modHR,interval="prediction",newdata=pred.frame)
pc<-predict(modHR,interval="confidence",newdata=pred.frame)
pred.age<-pred.frame$Age
plot(pred.age,pp[,1],lty=1,type="l",main="Prediction of Maximal Heart Rate",ylab="predicted",col=2,xlab=
lines(pred.age,pc[,2],lty=1)
lines(pred.age,pc[,3],lty=1)
lines(pred.age,pp[,2],lty=2)
lines(pred.age,pp[,3],lty=2)
grid()
```



We see that the uncertainty is high and that simple formulas (such as 220-age, etc.) do not work for most people.