

ANOVA as Regression (For Illustration)

We consider fitting the ANOVA model 4 different ways!

1. Model1: Fit the default “effects” model using the `lm()` function. This will be our typical approach! Look at parameter (coefficient) estimates, model matrix, ANOVA table and `lsmeans`.
2. Model2: Fit the alternate “no intercept” or “means” model using the `lm()` function.
3. Model3: Fit the default model “by hand” by creating the 3 indicator variables. This model is overparameterized, for illustration only.
4. Model4: Fit the default model “by hand” using just 2 indicator variables. This is equivalent to Model1.
5. Model5: If we do not define `trt` (1, 2, 3) as factor, then we fit a regression model instead of an ANOVA model. This would not be appropriate, for illustration only.

```
library(emmeans)
InData <- read.csv("C:/hess/STAT512/RNotes/Intro and R/RegANOVA.csv")
str(InData)
```

```
## 'data.frame': 6 obs. of 5 variables:
## $ trt: int 1 1 2 2 3 3
## $ y : num 6.3 5.9 4.3 4.8 3.7 3.9
## $ x1: int 1 1 0 0 0 0
## $ x2: int 0 0 1 1 0 0
## $ x3: int 0 0 0 0 1 1
```

```
#Important: Need to redefine trt as factor!
InData$trt <- as.factor(InData$trt)
str(InData)
```

```
## 'data.frame': 6 obs. of 5 variables:
## $ trt: Factor w/ 3 levels "1","2","3": 1 1 2 2 3 3
## $ y : num 6.3 5.9 4.3 4.8 3.7 3.9
## $ x1: int 1 1 0 0 0 0
## $ x2: int 0 0 1 1 0 0
## $ x3: int 0 0 0 0 1 1
```

```
aggregate(y ~ trt, FUN = mean, data = InData)
```

```
## trt y
## 1 1 6.10
## 2 2 4.55
## 3 3 3.80
```

Approach1: one-way ANOVA

This is the standard approach corresponding to the “Effects Model”. Typical research questions are addressed using the ANOVA table and pairwise comparison of means. Note that the `lsmeans` are the same as the simple means.

```
Model1 <- lm(y ~ trt, data = InData)
anova(Model1)
```

```
## Analysis of Variance Table
##
## Response: y
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## trt         2 5.5033  2.7517  36.689 0.007785 **
## Residuals   3 0.2250  0.0750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
emmeans(Model1, pairwise ~ trt, adjust = "none")
```

```
## $emmeans
##   trt emmean      SE df lower.CL upper.CL
## 1     6.10 0.1936492  3 5.483722 6.716278
## 2     4.55 0.1936492  3 3.933722 5.166278
## 3     3.80 0.1936492  3 3.183722 4.416278
##
## Confidence level used: 0.95
##
## $contrasts
##   contrast estimate      SE df t.ratio p.value
## 1 - 2         1.55 0.2738613  3   5.660 0.0109
## 1 - 3         2.30 0.2738613  3   8.398 0.0035
## 2 - 3         0.75 0.2738613  3   2.739 0.0714
```

```
model.matrix(Model1)
```

```
##   (Intercept) trt2 trt3
## 1           1     0     0
## 2           1     0     0
## 3           1     1     0
## 4           1     1     0
## 5           1     0     1
## 6           1     0     1
## attr("assign")
## [1] 0 1 1
## attr("contrasts")
## attr("contrasts")$trt
## [1] "contr.treatment"
```

```
summary(Model1)
```

```
##
## Call:
## lm(formula = y ~ trt, data = InData)
##
## Residuals:
##      1      2      3      4      5      6
## 0.20 -0.20 -0.25  0.25 -0.10  0.10
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.1000     0.1936  31.500 7.03e-05 ***
## trt2          -1.5500     0.2739   -5.660 0.01092 *
## trt3          -2.3000     0.2739   -8.398 0.00354 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2739 on 3 degrees of freedom
```

```
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9345
## F-statistic: 36.69 on 2 and 3 DF,  p-value: 0.007785
```

Approach2: No intercept model

When we fit the model without the intercept, the parameters/coefficients correspond to the trt means! This is also called the “Means Model”.

```
Model2 <- lm(y ~ trt - 1, data = InData)
model.matrix(Model2)
```

```
##      trt1 trt2 trt3
## 1      1      0      0
## 2      1      0      0
## 3      0      1      0
## 4      0      1      0
## 5      0      0      1
## 6      0      0      1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$trt
## [1] "contr.treatment"
```

```
summary(Model2)
```

```
##
## Call:
## lm(formula = y ~ trt - 1, data = InData)
##
## Residuals:
##      1      2      3      4      5      6
##  0.20 -0.20 -0.25  0.25 -0.10  0.10
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## trt1    6.1000     0.1936   31.50 7.03e-05 ***
## trt2    4.5500     0.1936   23.50 0.000169 ***
## trt3    3.8000     0.1936   19.62 0.000289 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2739 on 3 degrees of freedom
## Multiple R-squared:  0.9984, Adjusted R-squared:  0.9969
## F-statistic: 643.1 on 3 and 3 DF,  p-value: 0.0001038
```

```
anova(Model2)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## trt         3 144.705   48.235   643.13 0.0001038 ***
## Residuals   3   0.225    0.075
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Approach3: Regression with Indicator Variables

This model is *overparameterized*. That is why the NA values appear. For Illustration only!

```
Model3 <- lm(y ~ x1 + x2 + x3, data = InData)
summary(Model3)

##
## Call:
## lm(formula = y ~ x1 + x2 + x3, data = InData)
##
## Residuals:
##      1      2      3      4      5      6
##  0.20 -0.20 -0.25  0.25 -0.10  0.10
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8000     0.1936   19.623 0.000289 ***
## x1            2.3000     0.2739    8.398 0.003541 **
## x2            0.7500     0.2739    2.739 0.071422 .
## x3              NA           NA      NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2739 on 3 degrees of freedom
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9345
## F-statistic: 36.69 on 2 and 3 DF,  p-value: 0.007785
```

Approach4: Regression with Indicator Variables

Note that x1 is not included in the model statement. This model is equivalent to the default one-way ANOVA model in R.

```
Model4 <- lm(y ~ x2 + x3, data = InData)
model.matrix(Model4)
```

```
##      (Intercept) x2 x3
## 1             1  0  0
## 2             1  0  0
## 3             1  1  0
## 4             1  1  0
## 5             1  0  1
## 6             1  0  1
## attr("assign")
## [1] 0 1 2
```

```
summary(Model4)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3, data = InData)
```

```
##
## Residuals:
##      1      2      3      4      5      6
##  0.20 -0.20 -0.25  0.25 -0.10  0.10
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.1000     0.1936  31.500 7.03e-05 ***
## x2            -1.5500     0.2739  -5.660 0.01092 *
## x3            -2.3000     0.2739  -8.398 0.00354 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2739 on 3 degrees of freedom
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9345
## F-statistic: 36.69 on 2 and 3 DF,  p-value: 0.007785
```

What happens if we don't define trt as a factor?

Since trt is coded as 1,2,3 it will be defined as a numerical variable by default. If we don't define trt as a factor, a regression model will be fit! This is NOT appropriate for this data.

```
InData <- read.csv("C:/hess/STAT512/RNotes/Intro and R/RegANOVA.csv")
str(InData)
```

```
## 'data.frame':   6 obs. of  5 variables:
## $ trt: int  1 1 2 2 3 3
## $ y : num  6.3 5.9 4.3 4.8 3.7 3.9
## $ x1 : int  1 1 0 0 0 0
## $ x2 : int  0 0 1 1 0 0
## $ x3 : int  0 0 0 0 1 1
```

```
Model5 <- lm(y ~ trt, data= InData)
summary(Model5)
```

```
##
## Call:
## lm(formula = y ~ trt, data = InData)
##
## Residuals:
##      1      2      3      4      5      6
##  0.33333 -0.06667 -0.51667 -0.01667  0.03333  0.23333
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.1167     0.3576  19.904 3.76e-05 ***
## trt          -1.1500     0.1655  -6.948 0.00225 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.331 on 4 degrees of freedom
## Multiple R-squared:  0.9235, Adjusted R-squared:  0.9043
## F-statistic: 48.27 on 1 and 4 DF,  p-value: 0.002254
```

```
model.matrix(Model5)
```

```
##      (Intercept) trt  
## 1             1    1  
## 2             1    1  
## 3             1    2  
## 4             1    2  
## 5             1    3  
## 6             1    3  
## attr(,"assign")  
## [1] 0 1
```