Sums of Squares in R

uoepsy.github.io

11/11/2021

Some Data

```
library(tidyverse)
usmrsurv2 <- read_csv("https://uoepsy.github.io/data/usmrsurvey2.csv")
names(usmrsurv2)
```

```
##  [1] "id"                "pseudonym"         "catdog"
##  [4] "gender"            "height"            "optimism"
##  [7] "spirituality"      "ampm"              "extraversion"
## [10] "agreeableness"     "conscientiousness" "emotional_stability"
## [13] "imagination"       "internal_control"
```

```
names(usmrsurv2)[9:14]<-c("E","A","C","ES","I","LOC")
```

A model

Here is a model with two predictors, with the order of the predictors differing between the two models:

```
mymod1 <- lm(LOC ~ ES + optimism, data = usmrsurv2)
mymod2 <- lm(LOC ~ optimism + ES, data = usmrsurv2)
```

Type 1 of Sums of Squares

Type 1 Sums of Squares is the "incremental" or "sequential" sums of squares.
If we have a model $Y \sim A + B$, this method tests:

- the main effect of A
- the main effect of B after the main effect of A
- *Interactions (which come in Week 9 of the course) are tested after the main effects*[1].

Because this is sequential, the order matters.

We can get the Type 1 SS in R using the function `anova()`.
As you will see, the order in which the predictors are entered in the model influences the results.
This is because:

- for `mymod1` (`lm(LOC ~ ES + optimism)`) we are testing the main effect of `ES`, followed by the main effect of `optimism` *after* accounting for effects of `ES`.

- for `mymod2` (`lm(LOC ~ optimism + ES)`) it is the other way around: we test the main effect of `optimism`, followed by the main effect of `ES` *after* accounting for effects of `optimism`.

```
# mymod1 <- lm(LOC ~ ES + optimism, data = usmrsurv2)
anova(mymod1)
```

```
## Analysis of Variance Table
##
## Response: LOC
##           Df Sum Sq Mean Sq F value   Pr(>F)
## ES         1 114.66 114.661  7.3357 0.009036 **
## optimism   1  16.80  16.797  1.0746 0.304527
## Residuals 54 844.05  15.631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# mymod2 <- lm(LOC ~ optimism + ES, data = usmrsurv2)
anova(mymod2)
```

```
## Analysis of Variance Table
##
## Response: LOC
##           Df Sum Sq Mean Sq F value  Pr(>F)
## optimism   1  31.81  31.813  2.0353 0.15944
## ES         1  99.64  99.644  6.3749 0.01454 *
## Residuals 54 844.05  15.631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[1] So a model $Y \sim A + B + A : B$ will (in addition to the two above) test 1) the main effect of A 2) the main effect of B after the main effect of A and 3) the effect of the interaction A:B *after* the main effects of A and B.

Type 3 of Sums of Squares

Type 3 Sums of Squares is the "partial" sums of squares.
If we have a model $Y \sim A + B$, this method tests:

- the main effect of A after the main effect of B
- the main effect of B after the main effect of A

So the Type 3 will be equivalent to Type 1 *only for the final predictor in the model.*

Martin showed us one approach in the Live R session, using the `drop1()` function. We can also get the same using the `Anova()` function (capital **A**) from the **car** package.

```
# mymod1 <- lm(LOC ~ ES + optimism, data = usmrsurv2)
drop1(mymod1, test = "F")
```

```
## Single term deletions
##
## Model:
## LOC ~ ES + optimism
##          Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                844.05 159.62
## ES        1    99.644 943.70 163.99  6.3749 0.01454 *
## optimism  1    16.797 860.85 158.75  1.0746 0.30453
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that these results are the same as the Type 3 for `optimism` in the model `lm(LOC ~ ES + optimism)`, and the Type 3 for `ES` in the model `lm(LOC ~ optimism + ES)`. Take a look at the previous page for confirmation of this.

Note that Type 3 SS are **invariant to the order of predictors:** We get the same when we switch around our predictors:

```
# mymod2 <- lm(LOC ~ optimism + ES, data = usmrsurv2)
drop1(mymod2, test = "F")
```

```
## Single term deletions
##
## Model:
## LOC ~ optimism + ES
##          Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                844.05 159.62
## optimism  1    16.797 860.85 158.75  1.0746 0.30453
## ES        1    99.644 943.70 163.99  6.3749 0.01454 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The summary() function

Remember that we mentioned in the Week 8 lab that in the simple regression model (one predictor), the t-statistic for the coefficient test is the square root of the F-statistic for the test of the overall reduction in the residual sums of squares?

Well this does still hold for the multiple regression model, but it is a little more complicated.
For a given coefficient t-statistic in a multiple regression model, the associated F-statistic is the one corresponding to the reduction in residual sums of squares that is attributable to that predictor only. Or, in other words, the Type 3 F-statistic.

Here are our model coefficients and t-statistics:

```
# mymod1 <- lm(LOC ~ optimism + ES, data = usmrsurv2)
summary(mymod1)$coefficients
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 15.7589278 2.32636156 6.774066 9.587579e-09
## ES           0.1904987 0.07544910 2.524863 1.454432e-02
## optimism     0.0223095 0.02152114 1.036632 3.045273e-01
```

Here are our Type 3 SS F-statistics:

```
drop1(mymod1, test = "F")
```

```
## Single term deletions
##
## Model:
## LOC ~ ES + optimism
##          Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                844.05 159.62
## ES        1    99.644 943.70 163.99  6.3749 0.01454 *
## optimism  1    16.797 860.85 158.75  1.0746 0.30453
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can square-root them to get back to the t-statistic:

```
sqrt(drop1(mymod1, test = "F")$`F value`)
```

```
## [1]       NA 2.524863 1.036632
```

What this means is that just like the `drop1()` F test for reduction in residual sums of squares uses Type 3 SS, the t tests for the coefficients produced in `summary()` for a linear model are also *invariant to the order of predictors.*