F-tests of linear restrictions: examples

Heikki Korpela

November 7, 2024

For those interested, the examples here demonstrate three different ways to perform Wald- and F-tests in R. It is perfectly fine to use a package for this, and in particular, the function `linearHypothesis()` from the `car` package. In some applications, you may need to work with matrices in R; these examples illustrate how this would be done.

First, we read in the data:

```
suppressPackageStartupMessages({
  library(car)
  library(data.table)
})

# Set working directory to wherever you have the dataset
setwd('G:/My Drive/opinnot/2024-2025/Econometrics I/Tutorials/Tutorial-
I/')

# Read the data
NERLOVE116 <- fread('NERLOVE116.TXT')
setnames(NERLOVE116, c('totalCost', 'output',
                       'labourPrice', 'fuelPrice', 'capitalPrice'))

# Transform variables to logs (hint: a mistake has been left here)
transToLogs <- c('output', 'labourPrice', 'capitalPrice')

# The following is a data.table specific method for transforming variables.
NERLOVE116[,(transToLogs) := lapply(.SD,log),.SDcols=transToLogs]
```

Next, suppose we are given the model

$$\mathbf{y} = \mathbf{X}\beta, \mathbf{X} = \begin{bmatrix}\mathbf{1} & \mathbf{x_2} & \mathbf{x_2} & \mathbf{x_3} & \mathbf{x_4} & \mathbf{x_5}\end{bmatrix}, \beta = \begin{bmatrix}\beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5\end{bmatrix}^\mathsf{T}, \tag{1}$$

with x_2 standing for total output and x_3 through x_5 for prices of labour, fuel and capital respectively.

The null hypothesis is given as

$$\beta_3 + \beta_4 + \beta_5 = 1. \tag{2}$$

The first thing to do is to estimate the model. We can do this as usual with `lm`. (Of course, one may also do this with R primitives.)

Lecture slides (Week 1 / Testing hypotheses under the normality assumption / slide 7 on "Test of J linear restrictions") tell us that the restrictions can be written in a matrix form. This form has not been developed to make life more complicated for students, but because they can be used to write an estimator that has a known distribution under the null (see further below). The restrictions are:

$$\mathbf{R}\beta = \mathbf{q}$$

Each row of the left-hand side matrices should match the left-hand side of the restriction equations, and similarly for the right-hand side. β is specified by the model (1), and restrictions were given by (2).

As long as you remember your matrix multiplication, you should be able to simply see the type of $\mathbf{R}$ matrix we need:

$$\mathbf{R} = \left[\underbrace{0}_{\text{multiplies } \beta_1} \; \underbrace{0}_{\text{multiplies } \beta_2} \; \underbrace{1}_{\text{multiplies } \beta_3} \; \underbrace{1}_{\text{multiplies } \beta_4} \; \underbrace{1}_{\text{multiplies } \beta_5}\right], \mathbf{q} = \left[1\right],$$

where β_1 and β_5 did not appear in (2); to prevent them from appearing in the restriction equations, multiply them by zero.

Next, we are told that the F-test statistic has the structure

$$F = \frac{(\mathbf{Rb}-\mathbf{q})^\intercal \left(\mathbf{R}(\mathbf{X}^\intercal\mathbf{X})\mathbf{R}^\intercal\right)^{-1}(\mathbf{Rb}-\mathbf{q})}{Js^2}, \tag{3}$$

where J is the number of restrictions and s^2 is the estimator for the error term's variance, defined in the book on page 18 as the bias-adjusted sum of squared residuals:

$$s^2 = \frac{1}{N-K}\sum_{i=1}^{N} e_i^2, \tag{4}$$

where N is the number of observations and K is the number of regressors including the constant. Under the null hypothesis (model assumptions plus linear restriction), F has a F-distribution with J and $N-K$ degrees of freedom. Note that the asymptotic Wald-statistic also has an F-form, which asymptotically follows a $F_{J,\infty}$ distribution as $N\to\infty$.

Let's calculate the statistic ourselves in R.

```r
# Create the restriction matrix R
Rmatrix <- as.matrix(
  data.table(
    '(Intercept)' = 0L, 'output' = 0L, 'labourPrice' = 1L,
    'fuelPrice' = 1L, 'capitalPrice' = 1L))
# Create the b-estimate vector. coef() returns the point estimates. By default,
# as.matrix turns a vector of values into an explicit row vector.
bvector <- as.matrix(coef(modelResults))
# The RHS of Rb = q
qvector <- as.matrix(1L)
# The X matrix can be grabbed from the model with the following
# function (it can also be constructed directly from the data).
```

```r
Xmatrix <- model.matrix(modelResults)
# (X'X)^(-1) does the following:
# Multiply by transpose. t() does a transpose, while matrix multiplication
# is done by %*%. (Do not use A*B for matrix multiplication; it multiplies
# each element of A by the corresponding element of B!)
XtX <- t(Xmatrix) %*% Xmatrix
# Invert the multiplication result. This is done (for example) by
# solve():
XtXinverse <- solve(XtX)
# s (estimate for the distribution's sigma) is available as:
sigmaEstimate <- summary(modelResults)$sigma
# We have one restriction equation
JNofRestrictions <- 1L

# Number of coefficients
KNoOfCoefficients <- length(coef(modelResults))
# Number of observations
NNoOfObservations <- NERLOVE116[,.N]

Fstatistic <-
        t(Rmatrix %*% bvector - qvector) %*%
        solve( Rmatrix %*% XtXinverse %*% t(Rmatrix) ) %*%
        (Rmatrix %*% bvector - qvector)/(JNofRestrictions * sigmaEstimate^2)
# The result is a 1x1 matrix, which we equate with a scalar
Fstatistic <- Fstatistic[1,1]

# pf gives the distribution function for the F-distribution
# lower.tail = F means we ask for P(Fvariable > Fstatistic) rather than
# P(Fvariable < Fstatistic), the default.
FTestResult <-
        pf(Fstatistic,
                 df1 = JNofRestrictions,
                 df2 = NNoOfObservations - KNoOfCoefficients,
                 lower.tail = F)

# The asymptotic version (the Wald test) may be valid even if the exact
# small-sample counterpart using F-distribution is not. In this case, the
# F-statistic is the same, but the distribution used is different.
waldTestResult <-
        pchisq(Fstatistic, df = JNofRestrictions, lower.tail = F)

# Note that the asymptotic F-distribution is different
FAsymptoticTestResult <-
        pf(Fstatistic,
                 df1 = JNofRestrictions,
                 df2 = Inf,
                 lower.tail = F)
FTestResult - FAsymptoticTestResult
```

```
## [1] 0.002671
```

We can actually see the restriction matrices also by performing `linearHypothesis` with the *verbose* option.

```
FTestResultCheck <-
        linearHypothesis(modelResults,
    'labourPrice + fuelPrice + capitalPrice = 1', verbose = T,
    test = 'F')

##
## Hypothesis matrix:
##                                              (Intercept) output labourPrice
## labourPrice + fuelPrice + capitalPrice = 1           0      0           1
##                                              fuelPrice capitalPrice
## labourPrice + fuelPrice + capitalPrice = 1         1            1
##
## Right-hand-side vector:
## *rhs*
##     1
##
## Estimated linear function (hypothesis.matrix %*% coef - rhs)
## labourPrice + fuelPrice + capitalPrice = 1
##                                         26.4
##
##
## Estimated variance of linear function
## [1] 424

# The F-statistics match
all.equal( FTestResultCheck$F[2], Fstatistic )

## [1] TRUE

# The p-values match
all.equal( FTestResultCheck$`Pr(>F)`[2], FTestResult )

## [1] TRUE

waldTestResultCheck <-
        linearHypothesis(modelResults,
    'labourPrice + fuelPrice + capitalPrice = 1', verbose = T,
    test = 'Chisq')

##
## Hypothesis matrix:
##                                              (Intercept) output labourPrice
## labourPrice + fuelPrice + capitalPrice = 1           0      0           1
##                                              fuelPrice capitalPrice
```

```
## labourPrice + fuelPrice + capitalPrice = 1          1             1
##
## Right-hand-side vector:
## *rhs*
##     1
##
## Estimated linear function (hypothesis.matrix %*% coef - rhs)
## labourPrice + fuelPrice + capitalPrice = 1
##                                       26.4
##
##
## Estimated variance of linear function
## [1] 424

# The F-statistics match
all.equal( waldTestResultCheck$F[2], Fstatistic )

## [1] "target is NULL, current is numeric"

# The p-values match
all.equal( waldTestResultCheck$`Pr(>Chisq)`[2], waldTestResult )

## [1] TRUE
```

The lecture slides and the textbook (p. 27) both mention that you can also estimate a restricted model, and compare it to the unrestricted model, to get F-statistics:

$$F = \frac{(S_0 - S_1)/J}{S_1/(N-K)}, \tag{5}$$

where S_0 and S_1 are the sums of squared residuals of the restricted (null hypothesis) model and unrestricted model. How does this work?

Note that (2) can be plugged in the model equation (1) as follows:

$$\begin{aligned} \beta_3 + \beta_4 + \beta_5 = 1 \Rightarrow \beta_3 = 1 - \beta_4 - \beta_5 \Rightarrow \\ y_i = \beta_1 + \beta_2 x_{2i} + (1 - \beta_4 - \beta_5) x_{3i} + \beta_4 x_{4i} + \beta_5 x_{5i} \Leftrightarrow \\ y_i - x_{3,i} = \beta_1 + \beta_2 x_{2,i} + \beta_4 (x_{3i} - x_{4i}) + \beta_5 (x_{3i} - x_{5i}), \end{aligned}$$

and in fact each applicable restriction equation will allow you to delete coefficients needed to be estimated like this.

To do this in R:

```
# Use the I(x1 - x2) syntax to use the difference x1 - x2 as a variable
# Simply typing x1 - x2 inside a formula without I() adds the variable x1 and
# silently removes x1.
restrictedResults <- lm(
  I(totalCost - labourPrice) ~
    output +
```

```r
    I(fuelPrice - labourPrice) +
    I(capitalPrice - labourPrice),
  data = NERLOVE116)
S_0 <- sum(restrictedResults$residuals^2)
S_1 <- sum(modelResults$residuals^2)
FstatisticCheck <-
  ((S_0 - S_1)/JNofRestrictions)/
  (S_1/(NNoOfObservations - KNoOfCoefficients))
all.equal( FstatisticCheck, Fstatistic )

## [1] TRUE
```

Note that you can also calculate residuals as follows (no need to go through summary.lm()):

```r
sigmaEstimateCheck <- sqrt(sum(modelResults$residuals^2)/
        (NNoOfObservations - KNoOfCoefficients))
all.equal(sigmaEstimateCheck, sigmaEstimate)

## [1] TRUE
```