Week 3: Model Specification, Training Exercises

Coursera/Erasmus U., Econometric Methods and Applications

Anthony Nguyen

Training Exercise 3.5

Notes:

- This exercise uses the datafile `TrainExer35`and requires a computer.
- The dataset `TrainExer35` is available on the website.

Questions

(a) Replicate the R^2 values of slide 7 from lecture 3.5. In particular, show that a regression of the log equity premium (the variable *LogEqPrem* in the data file) on a constant and all five explanatory variables gives an R^2 of 10.8%, and that a regression of the log equity premium on a constant and only book-to market gives an R^2 of 6.3%. Then, based on these values, argue whether the additional four variables are significant when comparing the full with the book-to-market only model.

(b) Replicate the RESET statistic of slide 8 of Lecture 3.5. Proceed in the following steps. First regress the log equity premium on a constant and the book-to-market ratio. Then store the fitted log equity premium based on the output from this regression. Finally, regress the log equity premium on a constant, the book-to-market ration, and the square of the fitted log equity premium that was stored in the previous step. The RESET test statistic is the statistic of an F-test on the fitted log equity premium parameter.

(c) Replicate the Chow break statistic of slide 8 of Lecture 3.5. Proceed in the following steps. First regress the log equity premium on a constant and the book-to-market ratio and store the sum of squared residuals. Then perform the same regression for both the subsample of observations over 1927-1979, and the subsample of oberservations over 1980-2013. For both regressions, store the sum of squared residuals. Use these sum of squared residuals to calculate the Show break statistic.

(d) Replicate the Chow forecast statistic of slide 8 of Lecture 3.5. No new regression is required, you should be able to base this result on the regressions you have run so far.

Answers

(a) Replicate the R^2 values of slide 7 from lecture 3.5. In particular, show that a regression of the log equity premium (the variable *LogEqPrem* in the data file) on a constant and all five explanatory variables gives an R^2 of 10.8%, and that a regression of the log equity premium on a constant and only book-to market gives an R^2 of 6.3%. Then, based on these values, argue whether the additional four variables are significant when comparing the full with the book-to-market only model.

First we regress `LogEqPrem` on all explanatory variables as described:

```r
#regress LogEqPrem~BookMarket+NTIS+DivPrice+EarnPrice+Inflation
mod1 <- lm(LogEqPrem~BookMarket+NTIS+DivPrice+EarnPrice+Inflation, data = TrainExer35)
summary(mod1)
```

```
##
## Call:
## lm(formula = LogEqPrem ~ BookMarket + NTIS + DivPrice + EarnPrice +
##     Inflation, data = TrainExer35)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48999 -0.10794 -0.00693  0.11799  0.48550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.23441    0.38451   0.610   0.5438
## BookMarket  -0.17608    0.15410  -1.143   0.2566
## NTIS        -0.14563    0.81903  -0.178   0.8593
## DivPrice    -0.11985    0.09832  -1.219   0.2264
## EarnPrice    0.16715    0.08485   1.970   0.0523 .
## Inflation   -0.56691    0.58733  -0.965   0.3373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1875 on 81 degrees of freedom
## Multiple R-squared:  0.1084, Adjusted R-squared:  0.05335
## F-statistic: 1.969 on 5 and 81 DF,  p-value: 0.092
```

Next, we regress `LogEqPrem` only on `BookMarket`:

```r
#regress LogEqPrem~BookMarket
mod2 <-  lm(LogEqPrem~BookMarket, data = TrainExer35)
summary(mod2)
```

```
##
## Call:
## lm(formula = LogEqPrem ~ BookMarket, data = TrainExer35)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55549 -0.09729  0.01231  0.13419  0.40972
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.16593    0.04864   3.411 0.000991 ***
## BookMarket  -0.18508    0.07719  -2.398 0.018691 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1876 on 85 degrees of freedom
## Multiple R-squared:  0.06335,    Adjusted R-squared:  0.05233 
## F-statistic: 5.749 on 1 and 85 DF,  p-value: 0.01869
```

Looking at the results of our regression, for our first model, we have an R-squared value: R_1^2=0.1083879

In our second model, ignoring all of the other variables apart from `BookMarket`, we have an R-squared value: R_0^2=0.0633476
To compare the two models, we can use an F-test, which can be calculated as:

$$F = \frac{(SSR_R - SSR_{UR})/q}{(SSR_{UR})/(n-k)} \tag{1}$$

Or in it's alternative form, using R-squared:

$$F = \frac{(R_{UR}^2 - R_R^2)/q}{(1 - R_{UR}^2)/(n-k)} \tag{2}$$

Using the alternative form (2) and substituting in our values for R_1^2 and R_0^2, we find that:

$$F = \frac{(0.1084 - 0.06335)/4}{(1 - 0.1084)/(87 - 6)} = \frac{0.0112625}{0.01100741} = 1.023174$$

This gives us an F-statistic value of: 1.023174

To calculate the p-value, we can use the `pf()` function in `R`, which takes the form: `pf(q, df1, df2, ncp, lower.tail = TRUE, log.p = FALSE)`, where `df1` is `q` and `df2` is `n-k`. Substituting in our values for F, q, and n-k, we find that:

```r
pval <- pf(1.023174, 4, 81, lower.tail = FALSE)

paste("p-value =", round(pval,3))
```

```
## [1] "p-value = 0.4"
```

In this case, with our p-value of 0.4004259 being so large, we cannot reject the null hypothesis (H_0), which states that the effect of the extra regressors we added to our model (i.e. the unrestricted model) is zero. Given this, we should stick with the restricted version of our model, which only uses a single regressor, `BookMarket`.

Lastly, we can check all of this quickly by using the `linearHypothesis()` function in the `car` package:

```r
Hnull <- c("NTIS=0", "DivPrice=0", "EarnPrice=0", "Inflation=0")

linearHypothesis(mod1, Hnull)
```

```
## Linear hypothesis test
##
## Hypothesis:
## NTIS = 0
## DivPrice = 0
## EarnPrice = 0
## Inflation = 0
##
## Model 1: restricted model
## Model 2: LogEqPrem ~ BookMarket + NTIS + DivPrice + EarnPrice + Inflation
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     85 2.9917
## 2     81 2.8479  4   0.14386 1.0229 0.4005
```

(b) Replicate the RESET statistic of slide 8 of Lecture 3.5. Proceed in the following steps. First regress the log equity premium on a constant and the book-to-market ratio. Then store the fitted log equity premium based on the output from this regression. Finally, regress the log equity premium on a constant, the book-to-market ratio, and the square of the fitted log equity premium that was stored in the previous step. The RESET test statistic is the statistic of an F-test on the fitted log equity premium parameter.

To quickly summarize, the Ramsey RESET test can be described as such:

> In statistics, the Ramsey Regression Equation Specification Error Test (RESET) test is a general specification test for the linear regression model. More specifically, it tests whether non-linear combinations of the fitted values help explain the response variable. The intuition behind the test is that if non-linear combinations of the explanatory variables have any power in explaining the response variable, the model is misspecified in the sense that the data generating process might be better approximated by a polynomial or another non-linear functional form.

Thus, in order to recreate the RESET statistic from slide 8 of Lecture 3.5, we will run the regression of our various models and use their respective sum of squared residuals (SSR) to calculate an F-statistic using formula (1) from above.

Following the steps in this question, first we regress `LogEqPrem` by `BookMarket` and store the fitted values:

```r
#regress LeEqPrem~BookMarket
mod3 <- lm(LogEqPrem~BookMarket, data = TrainExer35)

#extract fitted values and save to vector
mod3_fitted <- predict(mod3)
```

Next, we regress `LogEqPrem` by `BookMarket` and the square of the fitted value from the previous regression:

```r
#add fitted values to data table
TrainExer35 <- cbind(TrainExer35, mod3_fitted)

#transform fitted values column
TrainExer35 <-TrainExer35 %>% mutate(mod3_fitted2 = mod3_fitted^2)

#re-run regression including tranformed column
mod4 <- lm(LogEqPrem~BookMarket+mod3_fitted2, data = TrainExer35)
```

The sum of squared residuals from our respective models can be calculated using matrix algebra, where $SSR = e'e$ or using the `sum()` function over the form $\sum e^2$:

```
SSR_R <- sum(residuals(mod3)^2)
SSR_UR <- sum(residuals(mod4)^2)
```

We can now use these values to calculate our F-statistic by hand, subbing in to get:

$$F = \frac{(SSR_R - SSR_{UR})/q}{(SSR_{UR})/(n-k)} = \frac{(2.991714 - 2.87383)/1}{2.87383/(87-3)} = 3.445665$$

Using this F-value, we can calculate the p-value of the RESET statistic:

```
p_RESET <- pf(3.445665, 1, 84, lower.tail = F)

paste("p_RESET =", round(p_RESET,3))
```

```
## [1] "p_RESET = 0.067"
```

Or more simply, we could have just used the `linearHypothesis()` function once again on our unrestricted model as such:

```
linearHypothesis(mod4, "mod3_fitted2=0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## mod3_fitted2 = 0
##
## Model 1: restricted model
## Model 2: LogEqPrem ~ BookMarket + mod3_fitted2
##
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     85 2.9917
## 2     84 2.8738  1   0.11789 3.4457 0.06692 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Replicate the Chow break statistic of slide 8 of Lecture 3.5. Proceed in the following steps. First regress the log equity premium on a constant and the book-to-market ratio and store the sum of squared residuals. Then perform the same regression for both the subsample of observations over 1927-1979, and the subsample of oberservations over 1980-2013. For both regressions, store the sum of squared residuals. Use these sum of squared residuals to calculate the Chow break statistic.

Again, to quote Wiki, the Chow break test can be described as follows:

> The Chow test, proposed by econometrician Gregory Chow in 1960, is a test of whether the true coefficients in two linear regressions on different data sets are equal. In econometrics, it is most commonly used in time series analysis to test for the presence of a structural break at a period which can be assumed to be known a priori (for instance, a major historical event such as a war). In program evaluation, the Chow test is often used to determine whether the independent variables have different impacts on different subgroups of the population.

The Chow test statistic is given by:

$$F_{Chow} = \frac{(S_C - (S_1 + S_2))/k}{(S_1 + S_2)/(N_1 + N_2 - 2k)}$$

Where, S_C is the sum of squared residuals from the combined data, S_1 is the sum of squared residuals from the first group, and S_2 is the sum of squared residuals from the second group. N_1 and N_2 are the number of observations in each group and k is the total number of parameters.

Following the steps outlined in the question, first we regress `LogEqPrem` on `BookMarket` and store the SSR:

```r
mod5 <- lm(LogEqPrem~BookMarket ,data = TrainExer35)

SSR_C <- sum(residuals(mod5)^2)
```

Next, we perform the same regression for both subamples of observations from 1927-1979 and 1980-2013 and store their respective SSRs:

```r
#create two subsets of TrainExer35 according to year ranges
TrainExer35_1 <- TrainExer35 %>% filter(Year <= 1979)
TrainExer35_2 <- TrainExer35 %>% filter(Year >= 1980)

#run same regression on subsets
mod5_1 <- lm(LogEqPrem~BookMarket ,data = TrainExer35_1)
mod5_2 <- lm(LogEqPrem~BookMarket ,data = TrainExer35_2)

#save SSRs for subsets
SSR_1 <- sum(residuals(mod5_1)^2)
SSR_2 <- sum(residuals(mod5_2)^2)
```

Finally, we can use all of our stored SSRs to calculate the Chow break statistic, where $N_1 = 53$, $N_2 = 34$, and $k = 2$:

```r
#\frac{(S_C - (S_1 + S_2))/k}{(S_1 + S_2)/(N_1 +N_2 - 2k)
F_Chow <- ((SSR_C - (SSR_1 + SSR_2))/2) / ((SSR_1 + SSR_2)/(53 + 34 - 2*2))

paste("F_Chow =", round(F_Chow,3))
```

```
## [1] "F_Chow = 2.269"
```

Using this value, we can calculate the p-value using the `pf()` function as we did before:

```r
p_Chow <- pf(2.268756, 2, 83, lower.tail = F)

paste("p_Chow =", round(p_Chow,3))
```

```
## [1] "p_Chow = 0.11"
```

(d) Replicate the Chow forecast statistic of slide 8 of Lecture 3.5. No new regression is required, you should be able to base this result on the regressions you have run so far.

The F-statistic for the Chow forecast test can be derived using:

$$F_{ChowForecast} = \frac{(SSR_C - SSR_1)/q}{SSR_1/(N_1 - k)}$$

Using the values we calculated from the previous regressions, we can substitute everything back so that:

```
#\frac{(SSR_C - SSR_1)/q}{SSR_1/(N_1 - k)}
F_ChowForecast <- ((SSR_C - SSR_1)/34)/(SSR_1/(53 - 2))

paste("F_ChowForecast =", round(F_ChowForecast,3))
```

```
## [1] "F_ChowForecast = 0.765"
```

With this F-stat, we show the p-value for the Chow forecast to be:

```
p_ChowForecast <- pf(F_ChowForecast, 34, 51, lower.tail = F)

paste("p_ChowForecast", round(p_ChowForecast, 3))
```

```
## [1] "p_ChowForecast 0.794"
```