Example 2-3

September 12, 2020

[ ]:
```
# first install the following packages and library
install.packages("pder")
install.packages("plm")
library("plm")

# import the data, frame the data, and create the model Qeq
data("TobinQ", package = "pder")
pTobinQ <- pdata.frame(TobinQ)
Qeq <- ikn ~ qn

# estimate the pooling, within, between, and random effects estimators
Q.pooling <- plm(Qeq, pTobinQ, model = "pooling")
Q.within <- update(Q.pooling, model = "within")
Q.between <- update(Q.pooling, model = "between")
Q.swar <- plm(Qeq, pTobinQ, model = "random", random.method = "swar")
```

[3]:
```
##---------------------------------Block 1---------------------------------

#### Example 2-3 ####

## ------------------------------------------------------------------------

# this command pulls the coefficients for the pooled, within, between, and␣
 ↪random effects models
sapply(list(pooling = Q.pooling, within = Q.within,
            between = Q.between, swar = Q.swar),
       function(x) coef(summary(x))["qn", c("Estimate", "Std. Error")])

## ------------------------------------------------------------------------

# computes the shares of the variances for the covariate qn
summary(pTobinQ$qn)
```

	pooling	within	between	swar
Estimate	0.00439197	0.0037919483	0.0051847368	0.0038622017
Std. Error	0.00015294	0.0001726447	0.0007490711	0.0001682634

total 314349.912127245 **between_id** 135423.856864277 **between_time** 29528.4571149111

[4]:
```
##---------------------------------Block 2---------------------------------

# calcuate the sum of squares for the within and between models and the total␣
 ↪sum of squares
SxxW <- sum(Within(pTobinQ$qn) ^ 2)
SxxB <- sum((Between(pTobinQ$qn) - mean(pTobinQ$qn)) ^ 2)
SxxTot <- sum( (pTobinQ$qn - mean(pTobinQ$qn)) ^ 2)

# pondW is the weight of the within model
pondW <- SxxW / SxxTot
pondW

# calculates the OLS estimator
pondW * coef(Q.within)[["qn"]] +
  (1 - pondW) * coef(Q.between)[["qn"]]
```

0.569193908953731

0.00439197005507543

[5]:
```
##---------------------------------Block 3---------------------------------

# to estimate the GLS estimator, we need to first estimate the parameter phi
# using the residuals of both the between and within estimators
T <- 35
N <- 188
smxt2 <- deviance(Q.between) * T / (N - 2)
sidios2 <- deviance(Q.within) / (N * (T - 1) - 1)
phi <- sqrt(sidios2 / smxt2)

## ------------------------------------------------------------------------
# again pondW is the weight of the within estimator
pondW <- SxxW / (SxxW + phi^2 * SxxB)
pondW

#calculates the GLS estimator
pondW * coef(Q.within)[["qn"]] +
  (1 - pondW) * coef(Q.between)[["qn"]]
```

0.949559133562537

0.00386220173915685