Example 9-2

September 11, 2020

[ ]:

```
# install the following packages and libraries
install.packages("plm")
install.packages("pglm")

library("pglm")
library("plm")
library("MASS")
library("maxLik")
library("texreg")
```

[3]:

```
# code to create a table for our results

extract.maxLik <- function (model, include.nobs = TRUE, ...){
    s <- summary(model, ...)
    names <- rownames(s$estimate)
    class(names) <- "character"
    co <- s$estimate[, 1]
    se <- s$estimate[, 2]
    pval <- s$estimate[, 4]
    class(co) <- class(se) <- class(pval) <- "numeric"
    n <- nrow(model$gradientObs)
    lik <- logLik(model)
    gof <- numeric()
    gof.names <- character()
    gof.decimal <- logical()
    gof <- c(gof, n, lik)
    gof.names <- c(gof.names, "Num. obs.", "Log Likelihood")
    gof.decimal <- c(gof.decimal, FALSE, TRUE)
    tr <- createTexreg(coef.names = names, coef = co, se = se, pvalues = pval,
                       gof.names = gof.names, gof = gof, gof.decimal = gof.
 ↪decimal)
    return(tr)
}
setMethod("extract", signature = className("maxLik", "maxLik"), definition =␣
 ↪extract.maxLik)
```

[4]:

```
##--------------------------------Block 1---------------------------------

#### Example 9-2 ####

## -----------------------------------------------------------------------
data("Fairness", package = "pglm")

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

# the polr() function from the MASS package used to estimate the ordered logit
# and ordered probit models.
parking.ol <- polr(answer ~ recurring + driving + education + rule,
                   data = Fairness, subset = good == "parking",
                   Hess = TRUE, method = "logistic")
parking.op <- update(parking.ol, method = "probit")

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

# estimation of the ordered probit random effects model
parking.opp <- pglm(as.numeric(answer) ~ recurring + driving + education + rule,
                    data = Fairness, subset = good == 'parking',
                    family = ordinal(link = 'probit'), R = 10, index = 'id',
                    model = "random")
parking.olp <- update(parking.opp, family = ordinal(link = 'probit'))

## -----------------------------------------------------------------------
screenreg(list(ologit = parking.ol, oprobit = parking.op,
               pologit = parking.olp, poprobit = parking.opp),
          digits = 3)
```

```
'\n=====================================================================
ologit oprobit pologit poprobit \n———————————————————————————————————————-
\nrecurringyes -0.120 -0.070 -0.077 -0.077 \n (0.075) (0.044) (0.059) (0.059) \ndrivingno 0.413 ***
0.237 *** 0.255 ** 0.255 ** \n (0.101) (0.060) (0.080) (0.080) \neducationno -0.480 *** -0.280
*** -0.309 ** -0.309 ** \n (0.138) (0.079) (0.105) (0.105) \nruleadmin -0.133 -0.061 -0.066 -0.066
\n (0.144) (0.086) (0.088) (0.088) \nrulelottery 0.330 * 0.217 * 0.238 ** 0.238 ** \n (0.141)
(0.085) (0.086) (0.086) \nruleaddsupply 1.892 *** 1.141 *** 1.221 *** 1.221 ***\n (0.143) (0.083)
(0.085) (0.085) \nrulequeuing 2.973 *** 1.731 *** 1.848 *** 1.848 ***\n (0.152) (0.086) (0.089)
(0.089) \nrulemoral 4.597 *** 2.656 *** 2.837 *** 2.837 ***\n (0.166) (0.093) (0.098) (0.098)
\nrulecompensation 4.231 *** 2.458 *** 2.622 *** 2.622 ***\n (0.162) (0.091) (0.096) (0.096)
\n0|1 0.369 *** 0.249 *** \n (0.108) (0.066) \n1|2 2.034 *** 1.199 *** \n (0.119) (0.069) \n2|3
4.527 *** 2.613 *** \n (0.144) (0.080) \n(Intercept) -0.269 *** -0.269 ***\n (0.072) (0.072)
\nmu_1 1.019 *** 1.019 ***\n (0.038) (0.038) \nmu_2 2.515 *** 2.515 ***\n (0.059) (0.059)
\nsigma 0.529 *** 0.529 ***\n (0.050) (0.050) \n——————————————————————————————
————————-\nAIC 5482.722 5490.689 \nBIC 5553.360 5561.326 \nLog Likelihood -2729.361
-2733.344 -2705.814 -2705.814 \nDeviance 5458.722 5466.689 \nNum. obs. 2661 2661 2661 2661
\n=====================================================================
```

p < 0.001; ** p < 0.01; * p < 0.05\n’