Final Lab

Kimberly Boswell

12/6/2023

Panel Data

How do we choose between a one way(individual or time only) versus a two-way fixed effect model?

```
data("EmplUK")
emppan <- pdata.frame(EmplUK, index=c("firm","year"), drop.index=TRUE, row.names=TRUE)

emppool <- plm(output~wage+capital+emp, data = EmplUK, model = "pooling")
empfixed.time<- plm(output~wage+capital+emp, data = EmplUK, model = "within", effect = "time")
pFtest( empfixed.time, emppool)
```

```
##
##  F test for time effects
##
## data:  output ~ wage + capital + emp
## F = 140.11, df1 = 8, df2 = 1019, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

You may need to change the method by which it estimates the parameters.

Qualitative Dependent Variable Models

Using the `TitanicSurvival` dataset, 1) Create a linear probability model 2) Correct for heteroskedasticity 3) What can be said about the likelihood of survival for older passengers versus younger. How about with respect to gender or passenger class? 4) What do you notice if you use margins to ascertain the marginal effect of the model? 5) Use a probit and logit model to evaluate and give the probability of survival of a 33 year old 3rd class woman on the sub. 6) Compare models

```
library(margins)

data("TitanicSurvival")
TitanicSurvival$survivednum <- as.numeric(TitanicSurvival$survived) - 1
titanreg.lpm <- lm(survivednum~ sex + age + passengerClass, data = TitanicSurvival)
coeftest(titanreg.lpm, vcov = hccm(titanreg.lpm,type="hc1"))
```

```
##
## t test of coefficients:
##
```

```
##                      Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)         1.1049549  0.0410275  26.9321 < 2.2e-16 ***
## sexmale            -0.4914131  0.0273469 -17.9696 < 2.2e-16 ***
## age                -0.0052695  0.0009500  -5.5468 3.689e-08 ***
## passengerClass2nd -0.2113738  0.0323328  -6.5374 9.784e-11 ***
## passengerClass3rd -0.3703874  0.0340949 -10.8634 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
predict(titanreg.lpm, data.frame(sex = "female", age = 33, passengerClass = "3rd"),
        type = "response")
```

```
##         1
## 0.5606755
```

```
predict(titanreg.lpm, data.frame(sex = "male", age = 33, passengerClass = "3rd"),
        type = "response")
```

```
##          1
## 0.06926235
```

```
margins(titanreg.lpm)
```

```
## Average marginal effects

## lm(formula = survivednum ~ sex + age + passengerClass, data = TitanicSurvival)

##        age sexmale passengerClass2nd passengerClass3rd
##  -0.005269 -0.4914           -0.2114           -0.3704
```

```
titanreg.probit <- glm(survivednum~ sex + age + passengerClass, data = TitanicSurvival,
                       family=binomial(link="probit"))

titanreg.logit <- glm(survivednum~ sex + age + passengerClass, data = TitanicSurvival,
                      family=binomial(link="logit"))

predvalp <- predict(titanreg.probit, data.frame(sex = "male", age = 33,
                                                passengerClass = "3rd"))
pnorm(predvalp)
```

```
##          1
## 0.08460383
```

```
predict(titanreg.probit, data.frame(sex = "male", age = 33,
                                    passengerClass = "3rd"),type = "response" )
```

```
##          1
## 0.08460383
```

```r
predict(titanreg.logit, data.frame(sex = "male", age = 33, passengerClass = "3rd"),
        type = "response")
```

```
##         1
## 0.0831416
```

```r
margins(titanreg.probit)
```

```
## Average marginal effects

## glm(formula = survivednum ~ sex + age + passengerClass, family = binomial(link = "probit"),     data

##        age sexmale passengerClass2nd passengerClass3rd
##  -0.005137  -0.491           -0.2282           -0.3759
```

In class code example - Coke v Pepsi

```r
data("coke")
coke.LPM <- lm(coke~pratio+disp_coke+disp_pepsi,
             data=coke)
#hcse for the lpm

hcErrors <- coeftest(coke.LPM,vcov.=hccm(coke.LPM,type="hc1"))


coke.probit <- glm(coke~pratio+disp_coke+disp_pepsi,
             data=coke, family=binomial(link="probit"))
coke.logit <- glm(coke~pratio+disp_coke+disp_pepsi,
             data=coke, family=binomial(link="logit"))

stargazer(hcErrors, coke.probit, coke.logit,
  header=FALSE,
  title="Three Binary Choice Models for the $coke$ Problem",
  type="text",
  keep.stat="n",digits=4, single.row=FALSE,
  intercept.bottom=FALSE,
  model.names=FALSE,
  column.labels=c("LPM","probit","logit"),
  omit.table.layout="n")
```

```
##
## Three Binary Choice Models for the coke Problem
## ==============================================
##                   Dependent variable:
##              --------------------------------
##                             coke
##                 LPM       probit     logit
##                 (1)        (2)        (3)
## ----------------------------------------------
```

```
## Constant     0.8902***  1.1080***  1.9230***
##               (0.0653)   (0.1925)   (0.3258)
##
## pratio       -0.4009*** -1.1459*** -1.9957***
##               (0.0604)   (0.1839)   (0.3146)
##
## disp_coke     0.0772**   0.2172**   0.3516**
##               (0.0339)   (0.0962)   (0.1585)
##
## disp_pepsi   -0.1657*** -0.4473*** -0.7310***
##               (0.0344)   (0.1010)   (0.1678)
##
## ---------------------------------------------
## Observations              1,140      1,140
## =============================================
```

```
hcErrors
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.890215   0.065301 13.6324 < 2.2e-16 ***
## pratio      -0.400861   0.060373 -6.6398 4.858e-11 ***
## disp_coke    0.077174   0.033932  2.2744   0.02313 *
## disp_pepsi  -0.165664   0.034365 -4.8207 1.624e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Extracting Probabilities

Now, if we want to find the probabilities at the representative points,

```
mod_est_lpm <- predict(coke.LPM, newdata=data.frame(pratio=1.2, disp_coke = 0, disp_pepsi=0) )
mod_est_lpm
```

```
##         1
## 0.4091814
```

```
mod_est_probit <- predict(coke.probit, newdata=data.frame(pratio=1.2, disp_coke = 0, disp_pepsi=0) )
pnorm(mod_est_probit)
```

```
##         1
## 0.3946997
```

```
mod_est_logit <- predict(coke.logit, newdata=data.frame(pratio=1.2, disp_coke = 0, disp_pepsi=0) )
plogis(mod_est_logit)
```

```
##         1
## 0.3841624
```

This way allows us to see the two part extraction of the probabilities 1. the linear portion 2. the corresponding probabilities using the standard normal cdf (probit) or logistic cdf (logit)

We could compute the probabilities, corresponding standard errors, degrees of freedom and residuals in one step:

```
predict(coke.LPM, data.frame(pratio=1.2, disp_coke = 0, disp_pepsi=0),
        type = "response", se.fit = TRUE)
```

```
## $fit
##         1
## 0.4091814
##
## $se.fit
## [1] 0.02218189
##
## $df
## [1] 1136
##
## $residual.scale
## [1] 0.4672405
```

```
predict(coke.probit, data.frame(pratio=1.2, disp_coke = 0, disp_pepsi=0),
        type = "response", se.fit = TRUE)
```

```
## $fit
##         1
## 0.3946997
##
## $se.fit
##          1
## 0.02450133
##
## $residual.scale
## [1] 1
```

```
predict(coke.logit, data.frame(pratio=1.2, disp_coke = 0, disp_pepsi=0),
        type = "response", se.fit = TRUE)
```

```
## $fit
##         1
## 0.3841624
##
## $se.fit
##          1
## 0.02517984
##
## $residual.scale
## [1] 1
```

Adding “response” gives us the response variable prediction rather than the result of the linear predictors.

Marginal Effects

Next we might be interested in the respective marginal effects, we will look at AER and MER. Using the `margins` package, we have:

```
#Average Marginal Effect
margins(coke.LPM)
```

```
## Average marginal effects

## lm(formula = coke ~ pratio + disp_coke + disp_pepsi, data = coke)

##   pratio disp_coke disp_pepsi
##  -0.4009   0.07717    -0.1657
```

```
margins(coke.probit)
```

```
## Average marginal effects

## glm(formula = coke ~ pratio + disp_coke + disp_pepsi, family = binomial(link = "probit"),     data =

##   pratio disp_coke disp_pepsi
##  -0.4097   0.07765    -0.1599
```

```
margins(coke.logit)
```

```
## Average marginal effects

## glm(formula = coke ~ pratio + disp_coke + disp_pepsi, family = binomial(link = "logit"),     data = 

##   pratio disp_coke disp_pepsi
##  -0.4333   0.07633    -0.1587
```

```
#At a representative point
margins(coke.probit, at = list(pratio=1.1, disp_coke = 0, disp_pepsi=0))
```

```
## Average marginal effects at specified values

## glm(formula = coke ~ pratio + disp_coke + disp_pepsi, family = binomial(link = "probit"),     data =

##  at(pratio) at(disp_coke) at(disp_pepsi)  pratio disp_coke disp_pepsi
##         1.1             0              0 -0.4519   0.08564    -0.1764
```

```
margins(coke.logit, at = list(pratio=1.2, disp_coke = 0, disp_pepsi=0))
```

```
## Average marginal effects at specified values

## glm(formula = coke ~ pratio + disp_coke + disp_pepsi, family = binomial(link = "logit"),     data = 

##  at(pratio) at(disp_coke) at(disp_pepsi)  pratio disp_coke disp_pepsi
##         1.2             0              0 -0.4722   0.08318    -0.1729
```