Applied Examples: Count Data and Binary Outcomes

Francis J. DiTraglia

1st-Year MPhil Econometrics - HT 2021

Count Data: Example 18.1 from Wooldridge (2010)

1988 Demographic and Health Survey in Botswana

```
# Load packages for robust standard errors (install them first!)
library(sandwich)
library(lmtest)

# Load the data from the wooldridge package (install it first!)
library(wooldridge)

# View the names of the columns
names(fertil2)
```

```
##  [1] "mnthborn" "yearborn" "age"       "electric" "radio"    "tv"
##  [7] "bicycle"  "educ"     "ceb"       "agefbrth" "children" "knowmeth"
## [13] "usemeth"  "monthfm"  "yearfm"    "agefm"    "idlnchld" "heduc"
## [19] "agesq"    "urban"    "urb_educ" "spirit"   "protest"  "catholic"
## [25] "frsthalf" "educ0"    "evermarr"
```

Description of Variables: `fertil2`

```r
# Specify x'beta
fertility_model <- children ~ educ + age + agesq + evermarr + urban +
  electric + tv
```

- `children` number of living children
- `educ` years of education
- `age` age in years
- `agesq` age squared
- `evermarr` equals 1 if ever married
- `urban` equals 1 if live in urban area
- `electric` equals 1 if has electricity
- `tv` equals 1 if has tv

Fit Poisson Regression and OLS

```r
pois_reg <- glm(fertility_model, family = poisson(link = 'log'),
                data = fertil2)
quasipois <- glm(fertility_model, family = quasipoisson(link = 'log'),
                 data = fertil2)
ols <- lm(fertility_model, data = fertil2)
```

Results: Poisson Regression with Poisson Variance Assumption

```r
coeftest(pois_reg)
```

```
##
## z test of coefficients:
##
##               Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept) -5.37482940  0.16286713 -33.0013 < 2.2e-16 ***
## educ        -0.02166447  0.00291310  -7.4369 1.031e-13 ***
## age          0.33733082  0.00993651  33.9486 < 2.2e-16 ***
## agesq       -0.00411583  0.00014528 -28.3308 < 2.2e-16 ***
## evermarr     0.31475104  0.02444729  12.8747 < 2.2e-16 ***
## urban       -0.08605490  0.02164866  -3.9751 7.036e-05 ***
## electric    -0.12053472  0.03883902  -3.1034  0.001913 **
## tv          -0.14470460  0.04738751  -3.0536  0.002261 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results: Poisson Regression with Quasi-Poisson Variance Assumption

```r
coeftest(quasipois)
```

```
##
## z test of coefficients:
##
##               Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept) -5.37482940  0.14111992 -38.0870 < 2.2e-16 ***
## educ        -0.02166447  0.00252412  -8.5830 < 2.2e-16 ***
## age          0.33733082  0.00860971  39.1803 < 2.2e-16 ***
## agesq       -0.00411583  0.00012588 -32.6967 < 2.2e-16 ***
## evermarr     0.31475104  0.02118291  14.8587 < 2.2e-16 ***
## urban       -0.08605490  0.01875798  -4.5876 4.483e-06 ***
## electric    -0.12053472  0.03365295  -3.5817 0.0003414 ***
## tv          -0.14470460  0.04105998  -3.5242 0.0004247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results: OLS with Robust SE

```
coeftest(ols, vcov. = vcovHC, type = 'HC0')
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept) -3.39383962  0.25132813 -13.5036 < 2.2e-16 ***
## educ        -0.06440859  0.00634670 -10.1484 < 2.2e-16 ***
## age          0.27247359  0.01983018  13.7404 < 2.2e-16 ***
## agesq       -0.00190671  0.00035513  -5.3690 8.331e-08 ***
## evermarr     0.68227245  0.05261334  12.9677 < 2.2e-16 ***
## urban       -0.22789330  0.04474180  -5.0935 3.664e-07 ***
## electric    -0.26173944  0.07292374  -3.5892 0.0003353 ***
## tv          -0.24995092  0.08207145  -3.0455 0.0023366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Overdispersion or Underdispersion?

```r
# Extract the estimate of sigma-squared
summary(quasipois)$dispersion
```

```
## [1] 0.7507749
```

```r
# Now do it "by hand"
yhat <- predict(pois_reg, type = 'response')
uhat <- residuals(pois_reg, type = 'response')
mean(uhat^2 / yhat)
```

```
## [1] 0.7493959
```

Robust “Sandwich” SEs for Poisson Regression

```r
coeftest(pois_reg, vcov. = vcovHC, type = 'HC0')
```

```
##
## z test of coefficients:
##
##               Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept) -5.37482940  0.14774627 -36.3788 < 2.2e-16 ***
## educ        -0.02166447  0.00259146  -8.3600 < 2.2e-16 ***
## age          0.33733082  0.00944626  35.7105 < 2.2e-16 ***
## agesq       -0.00411583  0.00014403 -28.5769 < 2.2e-16 ***
## evermarr     0.31475104  0.02320899  13.5616 < 2.2e-16 ***
## urban       -0.08605490  0.02004479  -4.2931 1.762e-05 ***
## electric    -0.12053472  0.03728819  -3.2325 0.0012270 **
## tv          -0.14470460  0.04380043  -3.3037 0.0009541 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing OLS and Poisson Estimates

```r
ybar <- mean(fertil2$children)
OLS_est <- coefficients(ols)[-1]
pois_est <- coefficients(pois_reg)[-1]
cbind(OLS = OLS_est, Poisson_APE = ybar * pois_est, Poisson = pois_est)
```

```
##                   OLS  Poisson_APE      Poisson
## educ     -0.064408588 -0.049131299 -0.021664469
## age       0.272473589  0.765008442  0.337330821
## agesq    -0.001906707 -0.009333995 -0.004115829
## evermarr  0.682272453  0.713801366  0.314751037
## urban    -0.227893299 -0.195157760 -0.086054903
## electric -0.261739443 -0.273352064 -0.120534717
## tv       -0.249950916 -0.328165205 -0.144704596
```

Note the age and agesq entries are not partial effects. Why not?

Binary Outcome: Example 15.1 from Wooldridge (2010)

Mroz (1987, Econometrica)

```r
# Load packages for robust standard errors (install them first!)
library(sandwich)
library(lmtest)

# Load the data from the wooldridge package (install it first!)
library(wooldridge)

# View the names of the columns
names(mroz)
```

```
##  [1] "inlf"     "hours"    "kidslt6"  "kidsge6"  "age"      "educ"
##  [7] "wage"     "repwage"  "hushrs"   "husage"   "huseduc"  "huswage"
## [13] "faminc"   "mtr"      "motheduc" "fatheduc" "unem"     "city"
## [19] "exper"    "nwifeinc" "lwage"    "expersq"
```

Description of Variables: `mroz`

```
# Specify the linear index
labor_model <- inlf ~ nwifeinc + educ + exper + expersq + age +
  kidslt6 + kidsge6
```

- `inlf` equals 1 if in labor force, 1975
- `nwifeinc` non-wife income in $1000
- `educ` years of schooling
- `exper` actual labor market experience
- `expersq` square of `exper`
- `age` woman's age in years
- `kidslt6` number of kids $<$ 6 years
- `kidsge6` number of kids 6-18

Fit LPM, Logit, and Probit

```r
lpm <- lm(labor_model, data = mroz)
logit <- glm(labor_model, family = binomial(link = 'logit'),
             data = mroz)
probit <- glm(labor_model, family = binomial(link = 'probit'),
              data = mroz)
```

LPM Results - Robust SE

```r
coeftest(lpm, vcov. = vcovHC, type = 'HC0')
```

```
##
## t test of coefficients:
##
##                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  0.58551922  0.15144889  3.8661 0.0001202 ***
## nwifeinc    -0.00340517  0.00151681 -2.2450 0.0250635 *
## educ         0.03799530  0.00722734  5.2572 1.913e-07 ***
## exper        0.03949239  0.00577907  6.8337 1.722e-11 ***
## expersq     -0.00059631  0.00018899 -3.1552 0.0016683 **
## age         -0.01609081  0.00238623 -6.7432 3.108e-11 ***
## kidslt6     -0.26181047  0.03161391 -8.2815 5.626e-16 ***
## kidsge6      0.01301223  0.01346085  0.9667 0.3340215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Logistic Regression Results

```r
coeftest(logit)
```

```
## 
## z test of coefficients:
## 
##               Estimate Std. Error z value  Pr(>|z|)    
## (Intercept)  0.4254524  0.8603645  0.4945  0.620951    
## nwifeinc    -0.0213452  0.0084214 -2.5346  0.011256 *  
## educ         0.2211704  0.0434393  5.0915 3.553e-07 ***
## exper        0.2058695  0.0320567  6.4220 1.345e-10 ***
## expersq     -0.0031541  0.0010161 -3.1041  0.001909 ** 
## age         -0.0880244  0.0145729 -6.0403 1.538e-09 ***
## kidslt6     -1.4433541  0.2035828 -7.0898 1.343e-12 ***
## kidsge6      0.0601122  0.0747893  0.8038  0.421539    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Probit Regression Results

```
coeftest(probit)
```

```
##
## z test of coefficients:
##
##                Estimate  Std. Error z value  Pr(>|z|)
## (Intercept)  0.27007357  0.50807817  0.5316  0.595031
## nwifeinc    -0.01202364  0.00493917 -2.4343  0.014919 *
## educ         0.13090397  0.02539873  5.1540 2.550e-07 ***
## exper        0.12334717  0.01875869  6.5755 4.850e-11 ***
## expersq     -0.00188707  0.00059993 -3.1455  0.001658 **
## age         -0.05285244  0.00846236 -6.2456 4.222e-10 ***
## kidslt6     -0.86832468  0.11837727 -7.3352 2.213e-13 ***
## kidsge6      0.03600561  0.04403026  0.8177  0.413502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Calculating Pseudo-R^2

```r
# Fit models with only an intercept
model0 <- inlf ~ 1
logit0 <- glm(model0, family = binomial(link = 'logit'), data = mroz)
probit0 <- glm(model0, family = binomial(link = 'probit'), data = mroz)

# Pseudo R-squared for Logit
1 - logLik(logit) / logLik(logit0)
```

```
## 'log Lik.' 0.2196814 (df=8)
```

```r
# Pseudo R-squared for Probit
1 - logLik(probit) / logLik(probit0)
```

```
## 'log Lik.' 0.2205805 (df=8)
```

Calculating Average Partial Effects

```r
# Average of g(x'beta_hat) where g is dlogis
# (predict defaults to the scale of x'beta_hat)
logit_APE_factor <- mean(dlogis(predict(logit)))
logit_APE_factor
```

```
## [1] 0.1785796
```

```r
# Average of g(x'beta_hat) where g is dnorm
# (predict defaults to the scale of x'beta_hat)
probit_APE_factor <- mean(dnorm(predict(probit)))
probit_APE_factor
```

```
## [1] 0.3007555
```

Comparison of APEs - LPM, Logit & Probit

```r
# Extract estimated coefficients, excluding the first (the constant)
lpm_est <- coefficients(lpm)[-1]
logit_est <- coefficients(logit)[-1]
probit_est <- coefficients(probit)[-1]

# Rescale the logit and probit estimates to obtain APEs
cbind(lpm = lpm_est, logit_APE = logit_APE_factor * logit_est,
      probit_APE = probit_APE_factor * probit_est)
```

```
##                    lpm     logit_APE   probit_APE
## nwifeinc -0.0034051689 -0.0038118135 -0.003616176
## educ      0.0379953030  0.0394965238  0.039370095
## exper     0.0394923895  0.0367641056  0.037097345
## expersq  -0.0005963119 -0.0005632587 -0.000567546
## age      -0.0160908061 -0.0157193606 -0.015895665
## kidslt6  -0.2618104667 -0.2577536551 -0.261153464
## kidsge6   0.0130122346  0.0107348186  0.010828887
```