20160922_Thompson_MLE_HW-02

Nick Thompson
9/16/2016

Homework 2 - GVPT 729A

Answer the following questions. Include your code, and report all the results you used to answer the questions.

https://raw.githubusercontent.com/Neilblund/729A/master/data/world.csv

The dataset at the link above contains information on cross national voter turnout (votevap) among the voting eligible population and per-capita GDP (gdppcap08).

1. Estimate the effect of per-capita GDP on voter turnout using OLS.

```
ols_01 <- lm(wdata$vote~wdata$gdppcap08); ols_01
```

```
##
## Call:
## lm(formula = wdata$vote ~ wdata$gdppcap08)
##
## Coefficients:
##     (Intercept)  wdata$gdppcap08
##      61.7294466        0.0001971
```

```
stargazer(ols_01, type = "text")
```

```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                 vote
## ------------------------------------------------
## gdppcap08                     0.0002**
##                               (0.0001)
##
## Constant                      61.729***
##                                (1.723)
##
## ------------------------------------------------
## Observations                     156
## R2                              0.037
## Adjusted R2                     0.031
## Residual Std. Error      16.448 (df = 154)
## F Statistic            5.991** (df = 1; 154)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

- On average a one unit change in per-capita GDP correlates to a 0.000197 unit change in voter turnout. Statistically significant at $p<0.05$.

2. Estimate the effect of per-capita GDP on voter turnout using the maximum likelihood estimator.

```
attach(wdata)
X <- cbind(1,gdppcap08)
y <- votevap

ols.lf<-function(theta,y,X){
  n<-nrow(X)
  k<-ncol(X)
  beta<-theta[1:k]
  sigma2<-theta[k+1]
  e<-y-X%*%beta
  logl<- -.5*n*log(2*pi)-.5*n*log(sigma2)-
    ((t(e)%*%e)/(2*sigma2))
  return(-logl)
}

p<-optim(c(1,1,1),ols.lf,method="BFGS",hessian=T,y=y,X=X)
p
```

```
## $par
## [1]  61.746656966   0.000196625 267.139491144
##
## $value
## [1] 657.1776
##
## $counts
## function gradient
##      139       36
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##                    [,1]                [,2]            [,3]
## [1,]    0.58396457803      8057.02290883 -0.00002233946
## [2,] 8057.02290882948 267281416.72829446 -0.01242283076
## [3,]   -0.00002233946        -0.01242283  0.00109241682
```

```
Fisher.p<-solve(p$hessian)
Fisher.p
```

```
##                  [,1]                [,2]              [,3]
## [1,]  2.93177351355 -0.000088376385904   0.058948527196
## [2,] -0.00008837639  0.000000006405423  -0.000001734418
## [3,]  0.05894852720 -0.000001734418154 915.402686655178
```

```
sqrt(diag(Fisher.p))
```

```
## [1]  1.71224224733  0.00008003389 30.25562239742
```

3. Take random samples of size 40, 30, and 20 from your dataset. Use the same model you used to answer questions 1 and 2, and compare your results.

```r
# Attach Data for Random Sample of 40
set.seed(42)
sample.40 <- wdata[sample(1:nrow(wdata), 40,
    replace=FALSE),]
attach(sample.40)

# Run OLS model for Random Sample of 40.
ols_40 <- lm(sample.40$vote~sample.40$gdppcap08)
stargazer(ols_40, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                vote
## -----------------------------------------------
## gdppcap08                     0.00002
##                              (0.0001)
##
## Constant                     66.450***
##                               (3.571)
##
## -----------------------------------------------
## Observations                    40
## R2                             0.001
## Adjusted R2                   -0.026
## Residual Std. Error      16.033 (df = 38)
## F Statistic             0.026 (df = 1; 38)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```r
# Run MLE model for Random Sample of 40.
# Name variables
X <- cbind(1,gdppcap08)
y <- votevap

ols.lf<-function(theta,y,X){
  n<-nrow(X)
  k<-ncol(X)
  beta<-theta[1:k]
  sigma2<-theta[k+1]
  e<-y-X%*%beta
  logl<- -.5*n*log(2*pi)-.5*n*log(sigma2)-
    ((t(e)%*%e)/(2*sigma2))
  return(-logl)
}


p.mle.40<-optim(c(1,1,1),ols.lf,method="BFGS",hessian=T,y=y,X=X)
p.mle.40
```

```
## $par
## [1]  66.5147810863   0.0000219116 312.8153039381
##
```

```
## $value
## [1] 167.2835
##
## $counts
## function gradient
##      209       64
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##                    [,1]              [,2]           [,3]
## [1,]    0.12787098314     2190.45229365 -0.00001284306
## [2,] 2190.45229365378 75633243.63976781  0.01875974576
## [3,]   -0.00001284306        0.01875975  0.00011473134
```

```
Fisher.p.mle.40<-solve(p.mle.40$hessian)
Fisher.p.mle.40
```

```
##                 [,1]              [,2]             [,3]
## [1,] 15.5205674273 -0.00044949938487    1.81087469350
## [2,] -0.0004494994  0.00000002623989   -0.00005460759
## [3,]  1.8108746935 -0.00005460758545 8716.22627647590
```

```
sqrt(diag(Fisher.p.mle.40))
```

```
## [1]  3.9396151370  0.0001619873 93.3607319834
```

```
set.seed(43)
sample.30 <- wdata[sample(1:nrow(wdata), 30,
    replace=FALSE),]
attach(sample.30)

ols_30 <- lm(sample.30$vote~sample.30$gdppcap08)
stargazer(ols_30, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                vote
## -----------------------------------------------
## gdppcap08                    0.0005**
##                              (0.0002)
##
## Constant                     57.013***
##                               (3.730)
##
## -----------------------------------------------
## Observations                    30
## R2                              0.165
## Adjusted R2                     0.135
## Residual Std. Error      15.512 (df = 28)
## F Statistic             5.518** (df = 1; 28)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
X <- cbind(1,gdppcap08)
y <- votevap

ols.lf<-function(theta,y,X){
  n<-nrow(X)
  k<-ncol(X)
  beta<-theta[1:k]
  sigma2<-theta[k+1]
  e<-y-X%*%beta
  logl<- -.5*n*log(2*pi)-.5*n*log(sigma2)-
    ((t(e)%*%e)/(2*sigma2))
  return(-logl)
}

p.mle.30<-optim(c(1,1,1),ols.lf,method="BFGS",hessian=T,y=y,X=X)
p.mle.30
```

```
## $par
## [1]  57.0395920356   0.0004753561 252.1881760265
##
## $value
## [1] 123.8794
##
## $counts
## function gradient
##       87       22
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##                   [,1]              [,2]            [,3]
## [1,]    0.118958787709     1423.980321590 -0.000007105427
## [2,] 1423.980321590079 40251581.485464193  0.002858541848
## [3,]   -0.000007105427        0.002858542  0.000184229521
```

```
Fisher.p.mle.30<-solve(p.mle.30$hessian)
Fisher.p.mle.30
```

```
##               [,1]              [,2]             [,3]
## [1,] 14.5810110871 -0.00051583251443    0.57036919820
## [2,] -0.0005158325  0.00000004309235   -0.00002056343
## [3,]  0.5703691982 -0.00002056343493 5428.03405574199
```

```
sqrt(diag(Fisher.p.mle.30))
```

```
## [1]  3.818509014  0.000207587 73.675192947
```

```
set.seed(44)
sample.20 <- wdata[sample(1:nrow(wdata), 20,
    replace=FALSE),]
attach(sample.20)

ols_20 <- lm(sample.20$vote~sample.20$gdppcap08)
stargazer(ols_20, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                vote
## ------------------------------------------------
## gdppcap08                     0.0002
##                              (0.0003)
##
## Constant                     61.860***
##                               (3.762)
##
## ------------------------------------------------
## Observations                    20
## R2                             0.032
## Adjusted R2                   -0.022
## Residual Std. Error      12.280 (df = 18)
## F Statistic             0.591 (df = 1; 18)
## ================================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
X <- cbind(1,gdppcap08)
y <- votevap

ols.lf<-function(theta,y,X){
  n<-nrow(X)
  k<-ncol(X)
  beta<-theta[1:k]
  sigma2<-theta[k+1]
  e<-y-X%*%beta
  logl<- -.5*n*log(2*pi)-.5*n*log(sigma2)-
    ((t(e)%*%e)/(2*sigma2))
  return(-logl)
}

p.mle.20<-optim(c(1,1,1),ols.lf,method="BFGS",hessian=T,y=y,X=X)
p.mle.20
```

```
## $par
## [1]  61.9430460487   0.0001954333 129.1009135791
##
```

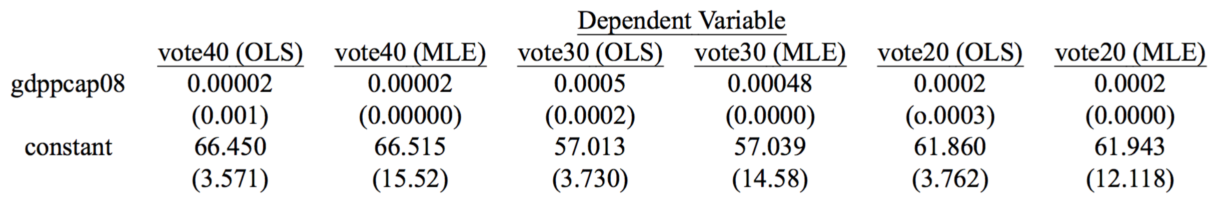
```
## $value
## [1] 77.49843
##
## $counts
## function gradient
##      293      100
##
## $convergence
## [1] 1
##
## $message
## NULL
##
## $hessian
##                    [,1]             [,2]            [,3]
## [1,]     0.15491757566     1525.2177118 -0.00003510436
## [2,] 1525.21771179615 32133660.5449883  0.37499552619
## [3,]   -0.00003510436        0.3749955  0.00066163608
```

```
Fisher.p.mle.20<-solve(p.mle.20$hessian)
Fisher.p.mle.20
```

```
##                 [,1]               [,2]             [,3]
## [1,] 12.1183926176 -0.00057520836739    0.96897531490
## [2,] -0.0005752084  0.00000005842291   -0.00006363113
## [3,]  0.9689753149 -0.00006363113164 1511.49235660433
```

```
sqrt(diag(Fisher.p.mle.20))
```

```
## [1]  3.4811481752  0.0002417083 38.8779160527
```



	Dependent Variable					
	vote40 (OLS)	vote40 (MLE)	vote30 (OLS)	vote30 (MLE)	vote20 (OLS)	vote20 (MLE)
gdppcap08	0.00002	0.00002	0.0005	0.00048	0.0002	0.0002
	(0.001)	(0.00000)	(0.0002)	(0.0000)	(o.0003)	(0.0000)
constant	66.450	66.515	57.013	57.039	61.860	61.943
	(3.571)	(15.52)	(3.730)	(14.58)	(3.762)	(12.118)

Table 1: Side-by-Side Comparisons

Figure 1:

- What differences do you notice between the OLS and MLE results?

  Table 1 shows a side by side comparison of all the results. The standard errors for the variable gdppcap08 (OLS) vary slightly (within 0.0002). The standard errors for the constant (OLS) vary more, but remain within 0.25 of one another. One the other hand, the standard errors for the MLE are inconsistent. In this case they reduce, but not in a uniform way.

- What characteristics of MLE and OLS estimators explain these differences?

  Asymptotic consistency is responsible for the variation in standard errors.

notes

- Turning off scientific notation can make it a little easier to make comparisons between your results. Use the command:

  To turn off scientific notation. Turn it back on by resetting ”scipen=0”

- You can format your tables however you see fit, but make sure you include the relevant information.
- http://www.statmethods.net/management/subset.html has code for taking random subsets from a data frame.
- If you want to reproduce your results later, you can set R’s random number seed with this command:

  #the number itself doesn’t matter

```
set.seed(1000)
```