hw3-8

Yifan Zheng

3/4/2020

8, King and Roberts (2015) gave three examples where the EHW standard errors differ from the OLS standard error. I have replicated one example in Section 4.4. Replicate another one.

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(sandwich)
# load data
load("DreherandJensenJLEreplication.RData")
```

```
# delete na and keep useful columns
data.0 = na.omit(x[,c("sum","un_per_l", "election_av", "un_el_av", "gdp_r_wk",
                      "gg_oecd_l","libor_l", "mg_l",
                      "imf_c_liab_l", "country","year")])

# replicate glm and get t statistic
model.0 = glm(sum ~ un_per_l*election_av + gdp_r_wk +
                gg_oecd_l + libor_l +mg_l+
                factor(country) + factor(year),
              data = data.0, family = "poisson")
hc0.0 = sqrt(diag(vcovHC(model.0, type = "HC0")))
hc1.0 = sqrt(diag(vcovHC(model.0, type = "HC1")))
covar.0 = summary(model.0)$coef
```

```
# show standard errors
se.0 = cbind(covar.0[1:7,2], hc0.0[1:7],hc1.0[1:7])
colnames(se.0) = c("ols","hc0","hc1")
round(se.0[2,],2)
```

```
##  ols  hc0  hc1
## 3.74 6.29 8.08
```

"For their coefficient on U.S.support -9.55, the classical standard error is 3.73, whereas the robust standard error is larger, at 6.28, a difference of substantive importance." (King and Roberts (2015))

"We fix the first problem by switching from a Poisson to a negative binomial distribution and the second by truncating it." (King and Roberts (2015))

"The result is a 0-to-4 truncated negative binomial regression model, paralleling our simulation on the effects of changing to a better-fitting distribution" (King and Roberts (2015))

```
library(maxLik)
```

```
## Loading required package: miscTools
```

```
##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. C
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum o
## https://r-forge.r-project.org/projects/maxlik/
```

```
# construct a 0 truncated nb ll
trunc.0.ll <- function(par, y, X, cut){
  end <- (length(par))-1
  theta <- par[1:end]
  alpha <- exp(par[length(par)])
  lambda <- exp(drop(X %*% theta))
  zeros <- pnbinom(cut, size = alpha, mu = lambda,
                   lower.tail = F, log.p = T)
  ll <- sum(dnbinom(y, size = alpha, mu = lambda, log = T) - zeros)
  return(ll)
}


# get x and y
X <- cbind(model.matrix(model.0)[,!is.na(as.vector(model.0$coefficients))])
y <- data.0$sum

# optimize
outcome <- optim(c(rep(0.01, ncol(X)), 0.01), trunc.0.ll,
                 y = y, X = X, cut = 4,
                 control = list(fnscale = -1, maxit = 10000),
                 method = "BFGS",hessian = T)
bread <- solve(-outcome$hessian)
halfmeat <- apply(cbind(y, X), 1,
              function(x) numericGradient(trunc.0.ll, outcome$par, y=x[1],
                                          X = x[2:(length(x))], cut = 4))
# get sandwich
meat <- halfmeat %*% t(halfmeat)
sw <- bread %*% meat %*% bread

# check standard errors
```

```r
se.1 = cbind(sqrt(diag(sw))[1:7], sqrt(diag(bread))[1:7])
colnames(se.1) = c("ols.new","hc0")
round(se.1[2,], 2)
```

```
## ols.new     hc0
##    6.80    6.12
```

In this new model, the robust standard error is nearly what the paper reported, that is, 6.76.