

Lista10

Guilherme Pazian

19 de junho de 2017

1. Exercícios 8.1, 8.2, 8.3 e 10.1 de Stock-Watson. 8.1

a)

```
popNJ <- 8100000
coef_beertax <- -0.45
ep_coef_beertax <- 0.22
mortes1 <- 1*(-0.45)*popNJ/10000
ep_mortes1 <- ep_coef_beertax*popNJ/10000

LIIC_mortes1 <- mortes1 - qnorm(.975)*ep_mortes1

LSIC_mortes1 <- mortes1 + qnorm(.975)*ep_mortes1
LIIC_mortes1
```

```
## [1] -713.7656
```

```
LSIC_mortes1
```

```
## [1] -15.23442
```

b)

```
coef_idade18 <- 0.028
ep_coef_idade18 <- 0.066
mortes2 <- coef_idade18 * popNJ/10000
ep_mortes2 <- ep_coef_idade18 * popNJ/10000

LIIC_mortes2 <- mortes2 - qnorm(.975)*ep_mortes2

LSIC_mortes2 <- mortes2 + qnorm(.975)*ep_mortes2
LIIC_mortes2
```

```
## [1] -82.09967
```

```
LSIC_mortes2
```

```
## [1] 127.4597
```

c)

```
coef_lnRenda <- 1.81
ep_coef_lnRenda <- 0.47
mortes3 <- 0.01 * coef_lnRenda * popNJ/10000
ep_mortes3 <- ep_coef_lnRenda * popNJ/10000

LIIC_mortes3 <- mortes3 - qnorm(.95)*ep_mortes3

LSIC_mortes3 <- mortes3 + qnorm(.95)*ep_mortes3
LIIC_mortes3
```

```
## [1] -611.5348
```

```
LSIC_mortes3
```

```
## [1] 640.8568
```

- d) Sim, os efeitos temporais removem efeitos de variáveis não observáveis que são constantes em relação à estados mas mudam conforme o tempo, por isso, eliminamos possível vies de omissão de variáveis constantes nos estados (mas variam no tempo). O próprio modelo feito sem a inserção de efeitos temporais apresenta um coeficiente de determinação (R^2) e coeficiente de determinação ajustado (\bar{R}^2) menores em relação ao modelo que considera os efeitos temporais.
- e) Não necessariamente, pois se levarmos em consideração que os modelos estão ambos bem ajustados, não tendo nenhum viés de qualquer tipo, realmente podemos fazer o tipo de comparação, porém um viés nas estimativas de mínimos quadrados podem trazer conclusões totalmente erradas quanto ao problema, uma vez que a estimativa pode estar sobredeterminada no modelo (5) por exemplo, de maneira que o coeficiente estimado é maior em modulo do que o verdadeiro valor do parâmetro estimado.
- f)

2. Implementar um programa em R para reproduzir os resultados da Tabela 8.1 de Stock-Watson

<http://fmwww.bc.edu/ec-p/data/stockwatson/datasets.list.html>

```
library(foreign)
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: lmtest
```

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

```
##
```

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

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```
dados <- read.dta("fatality.dta")
dados$mrall <- dados$mrall * 10000
dados$mlda <- round(dados$mlda,0)
dados$mlda <- as.factor(dados$mlda)
dados$mlda <- relevel(dados$mlda,ref= 4)
```

```
fit_1 <- lm(mrall~beertax,data= dados)
#summary(fit_1)
epfit_1robusto <- vcovHC(fit_1, type = "HC1")
#epfit_1robusto
#diag(epfit_1robusto)
coef_fit_1 <- coeftest(fit_1, epfit_1robusto)
coef_fit_1[1:2,]
```

```
##              Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 1.8533079 0.04712975 39.32353 2.239345e-127
## beertax      0.3646054 0.05285240  6.89856 2.642889e-11
```

```
fit_2 <- lm(mrall ~ beertax + as.factor(state), data = dados)
#summary(fit_2)
epfit_2robusto <- vcovHC(fit_2, type = "HC1")
#epfit_2robusto
#diag(epfit_2robusto)
coef_fit_2 <- coeftest(fit_2, epfit_2robusto)
coef_fit_2[1:2,]
```

```
##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept)  3.4776300  0.3507844  9.913866 4.086456e-20
## beertax      -0.6558736  0.2032797 -3.226459 1.398373e-03
```

```
fit_3 <- lm(mrall ~ beertax + as.factor(state) + as.factor(year), data = dados)
#summary(fit_3)
epfit_3robusto <- vcovHC(fit_3, type = "HC1")
#epfit_3robusto
#diag(epfit_3robusto)
coef_fit_3 <- coeftest(fit_3, epfit_3robusto)
coef_fit_3[1:2,]
```

```
##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept)  3.5113747  0.4474138  7.848160 8.909071e-14
## beertax      -0.6399799  0.2547149 -2.512534 1.254701e-02
```

```
fit_4 <- lm(mrall ~ beertax + mllda + jaild + comserd + vmiles + unrate + log(perinc) + as.factor(state))
r2_fit4 <- summary(fit_4)$r.squared
r2_aj_fit4 <- summary(fit_4)$adj.r.squared
epfit_4robusto <- vcovHC(fit_4, type = "HC1")
#epfit_4robusto
#diag(epfit_4robusto)
coef_fit_4 <- coeftest(fit_4, epfit_4robusto)
coef_fit_4[1:10,]
```

```
##              Estimate   Std. Error   t value    Pr(>|t|)
## (Intercept) -1.255211e+01 4.530851e+00 -2.77036555 5.985138e-03
## beertax      -4.635768e-01 2.223838e-01 -2.08457983 3.804029e-02
## mllda18       3.330317e-02 6.424986e-02  0.51833834 6.046436e-01
## mllda19      -3.991231e-03 4.208546e-02 -0.09483634 9.245146e-01
## mllda20       6.251283e-03 3.929375e-02  0.15909102 8.737152e-01
## jaild        1.430645e-02 3.176753e-02  0.45034837 6.528180e-01
## comserd       3.443702e-02 1.147123e-01  0.30020335 7.642514e-01
## vmiles        8.922890e-06 8.102437e-06  1.10126002 2.717571e-01
## unrate       -6.295173e-02 1.163892e-02 -5.40872577 1.391263e-07
## log(perinc)  1.776883e+00 4.725621e-01  3.76010463 2.079350e-04
```

```
fit_5 <- lm(mrall ~ beertax + as.factor(mllda) + jaild + comserd + vmiles + unrate + log(perinc) + as.factor(year))
r2_fit5 <- summary(fit_5)$r.squared
r2_aj_fit5 <- summary(fit_5)$adj.r.squared
epfit_5robusto <- vcovHC(fit_5, type = "HC1")
#epfit_5robusto
#diag(epfit_5robusto)
coef_fit_5 <- coeftest(fit_5, epfit_5robusto)
coef_fit_5[1:10,]
```

```
##              Estimate   Std. Error   t value    Pr(>|t|)
## (Intercept) -1.969763e+00 4.123708e+00 -0.47766786 0.63326197
```

```
## beertax          -3.883038e-01 2.260654e-01 -1.71766136 0.08697197
## as.factor(mlda)18 1.322783e-01 5.399961e-02 2.44961583 0.01491802
## as.factor(mlda)19 5.565924e-02 4.984405e-02 1.11666768 0.26510064
## as.factor(mlda)20 4.450012e-03 3.918668e-02 0.11355929 0.90966912
## jaild           -2.008059e-02 3.260663e-02 -0.61584371 0.53850155
## comserd         -1.155504e-02 1.181534e-01 -0.09779688 0.92216406
## vmiles          -2.397921e-06 7.412143e-06 -0.32351251 0.74655036
## unrate          -2.117356e-02 1.007678e-02 -2.10122381 0.03652150
## log(perinc)      5.599802e-01 4.160097e-01 1.34607490 0.17937498
```

```
dados1=dados
dados1$mlda=as.factor(dados1$mlda)
dados1=within(dados1,mlda <- relevel(dados1$mlda, ref=4))

dados1$mlda <- relevel(dados1$mlda,ref= 4)
```