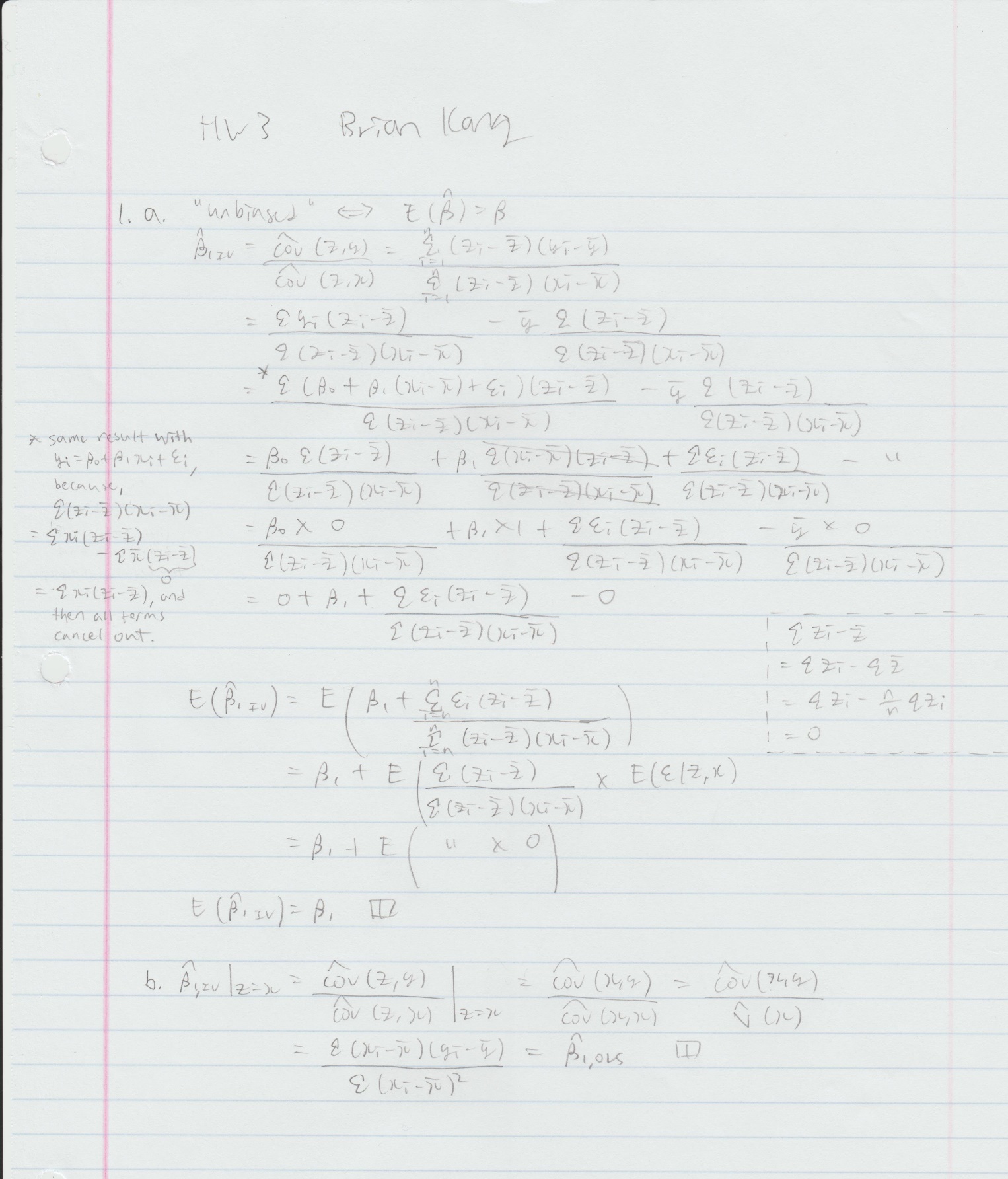
hw03

Brian Kang

Wed Feb 06 13:44:11 2019

THIS IS THE NUMERICAL PART



THIS IS THE COMPUTATIONAL PART

# hw3  
# 1)  
n <- 500  
# i)  
set.seed(123456)  
xi <- runif(n,0,10)  
mean(xi)

## [1] 4.946002

sd(xi)

## [1] 2.83055

# ii)  
set.seed(123456)  
ui <- rnorm(n,0,sqrt(36))  
mean(ui)

## [1] 0.07399147

# The sample average of ui will not be zero unless all  
# 500 generated error numbers equals to zero OR the  
# samples are exactly symmetric with zero being the center.  
# Since we generated with mu=zero with sd=6, it is extremely   
# unlikely that we will get a sample mean exactly equal to 0.  
sd(ui)

## [1] 5.985243

# iii)  
yi <- 1+2\*xi+ui  
lm.0 <- lm(yi~xi)  
summary(lm.0)

##   
## Call:  
## lm(formula = yi ~ xi)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -22.7300 -3.9472 0.2413 4.1648 16.0338   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.99934 0.53982 1.851 0.0647 .   
## xi 2.01509 0.09475 21.267 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.991 on 498 degrees of freedom  
## Multiple R-squared: 0.476, Adjusted R-squared: 0.4749   
## F-statistic: 452.3 on 1 and 498 DF, p-value: < 2.2e-16

# Intercept: 0.99934  
# Slope: 2.01509  
# Both estimated intercept and slope are pretty close to the  
# population intercept and slope. They are not equal due to  
# the random generation of numbers and addition of random  
# error from a normal distribution to a term from a uniform   
# distribution.  
  
# iv)  
resids <- lm.0$residuals  
sum(resids)

## [1] 5.329071e-15

sum(xi\*resids)

## [1] -2.163963e-13

# Both sums are extremely close to zero.  
# Thus, verifying two of our assumptions.  
  
# v)  
sum(ui)

## [1] 36.99574

sum(xi\*ui)

## [1] 243.3267

# These sums are nowhere near zero. Clarifying that there is  
# a difference between errors and residuals.  
  
# vi)  
# vi) i)  
set.seed(567890)  
xi2 <- runif(n,0,10)  
mean(xi2)

## [1] 4.936884

sd(xi2)

## [1] 2.756792

# vi) ii)  
set.seed(567890)  
ui2 <- rnorm(n,0,sqrt(36))  
mean(ui2)

## [1] -0.2989511

# Again, the sample average of ui2 will not be zero unless all  
# 500 generated error numbers equals to zero OR the  
# samples are exactly symmetric with zero being the center.  
# Since we generated with mu=zero with sd=6, it is extremely   
# unlikely that we will get a sample mean exactly equal to 0.  
sd(ui2)

## [1] 5.705766

# vi) iii)  
yi2 <- 1+2\*xi2+ui2  
lm.02 <- lm(yi2~xi2)  
summary(lm.02)

##   
## Call:  
## lm(formula = yi2 ~ xi2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.399 -3.788 -0.249 3.615 16.680   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.15412 0.52355 0.294 0.769   
## xi2 2.11078 0.09261 22.791 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.703 on 498 degrees of freedom  
## Multiple R-squared: 0.5105, Adjusted R-squared: 0.5096   
## F-statistic: 519.4 on 1 and 498 DF, p-value: < 2.2e-16

# Intercept: 0.15412 (Beta0\_hat)  
# Slope: 2.11078 (Beta1\_hat)  
# This time, our intercept estimate is quite off and our slope  
# estimate is a bit off than before from the population values.  
# Our estimators' values are different from before's because  
# we are using different samples. And with the variations  
# caused by the random samples, our Beta0\_hat and Beta1\_hat  
# values will vary almost everytime.  
  
# 2)  
n <- 1000  
set.seed(123456)  
x <- runif(n,0,1) # x val  
e <- rnorm(n,0,1) # error  
v <- rnorm(n,1,1)  
z <- v\*x # iv val  
y <- 4+0.3\*x+e # true val  
# a)  
lm.1 <- lm(y~x)  
# b)  
# install.packages("AER")  
library(AER)

## Warning: package 'AER' was built under R version 3.5.2

## Loading required package: car

## Loading required package: carData

## Loading required package: lmtest

## Loading required package: zoo

##   
## Attaching package: 'zoo'

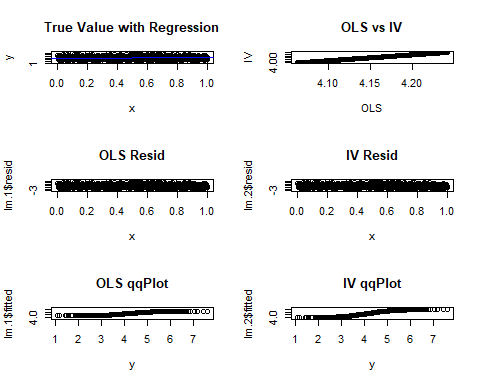
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

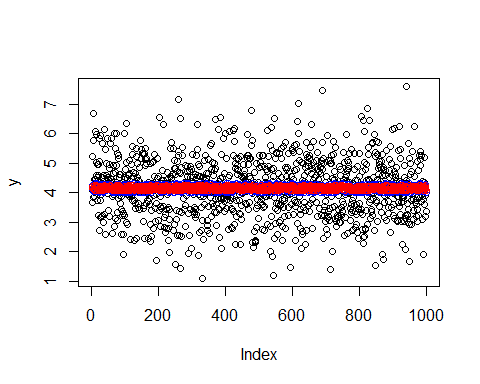
## Loading required package: survival

## Warning: package 'survival' was built under R version 3.5.2

lm.2 <- ivreg(y~x|z)  
# plots to compare  
par(mfrow = c(3,2))  
plot(x,y,main = "True Value with Regression")  
abline(lm.1, col = 'red')  
abline(lm.2, col = 'blue')  
plot(lm.1$fitted,lm.2$fitted, main = "OLS vs IV",   
 xlab = "OLS", ylab = "IV")  
# residual plots  
plot(x,lm.1$resid, main = "OLS Resid")  
plot(x,lm.2$resid, main = "IV Resid")  
# qqplots  
qqplot(y,lm.1$fitted, ylim = c(3.95,4.35), main = "OLS qqPlot")  
qqplot(y,lm.2$fitted, ylim = c(3.95,4.35), main = "IV qqPlot")



par(mfrow = c(1,1))  
plot(y)  
points(lm.2$fitted, col = "blue")  
points(lm.1$fitted, col = "red")



# summaries  
summary(lm.1)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.1444 -0.6895 0.0168 0.6665 3.5379   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.06372 0.06307 64.433 <2e-16 \*\*\*  
## x 0.17777 0.10994 1.617 0.106   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.001 on 998 degrees of freedom  
## Multiple R-squared: 0.002613, Adjusted R-squared: 0.001613   
## F-statistic: 2.614 on 1 and 998 DF, p-value: 0.1062

summary(lm.2)

##   
## Call:  
## ivreg(formula = y ~ x | z)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.209882 -0.690039 0.008174 0.668678 3.597661   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.9971 0.1290 30.990 <2e-16 \*\*\*  
## x 0.3121 0.2520 1.238 0.216   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.002 on 998 degrees of freedom  
## Multiple R-Squared: 0.001122, Adjusted R-squared: 0.0001209   
## Wald test: 1.533 on 1 and 998 DF, p-value: 0.2159

# summary(lm.1)$sigma  
# summary(lm.2)$sigma  
# summary(lm.1)$r.squared  
# summary(lm.2)$r.squared  
  
# i)  
# Comparing the OLS estimates and the true values, we can see  
# that the redisuals are quite large with a R-squared value of  
# 0.002613. The residual SE is 1.001. The linear model's  
# intercept is 4.06372 with a slope 0.17777. Looking at the  
# qqplot we can clearly tell that the generated data comes from  
# a normal distribution. Overall, we can say that the OLS  
# regression captures the general trend of the true values.  
  
# ii)  
# Comparing the IV estimates and the true values, we can see  
# that the residuals are again quite large with a R-squared  
# value of 0.001122. The residual SE is 1.002. The model's  
# intercept is 3.9971 with a slope 0.3121. From the qqplot we  
# can again see that the data comes from a normal distribution.  
# Here as well, we can say that the IV regression captures the  
# general trend of the true values.  
  
# iii)  
# The interesting facts come from comparing OLS and IV. Even   
# though they give similar estimates, they are different. The  
# IV's R-squared value is smaller. Its residual SE is larger.  
# Its intercept is close enough but has a steeper slope. The  
# min and max values of the residuals are a slightly more  
# extreme than the OLS's. I would say that these two estimates  
# give mostly similar results from our simple generated true  
# data, but the IV estimate is just slightly more "spread out."