hw5

Kaicheng Luo

2019/11/11

Problem 4

```
## RDD numerical examples
set.seed(1000)
n   = 500
x   = rnorm(n)
y0  = x^2 + rnorm(n, 0, 0.5)
y1  = y0 + 5
z   = (x>=0)
y   = z*y1 + (1-z)*y0
png('plot0.png')
plot(y0 ~ x, col = "grey", pch = 19, cex = 0.1,
     ylim = c(min(y), max(y)),
     xlab = "X", ylab = "Y")
points(y1 ~ x, col = "grey", pch = 19, cex = 0.1)
points(y ~ x, col = "black", pch = 19, cex = 0.1)
abline(v = 0, lty = 2)
dev.off()
```

```
## pdf
##   2
```

```
png('plot1.png')
plot(rdd_data(x=x, y=y,cutpoint=0),
     xlab = "X", ylab = "Y", cex = 0.3)
dev.off()
```

```
## pdf
##   2
```

```
# The point estimate, variance estimate, and confidence interval for the causal effect
RDDest = rdrobust(y, x)
# Generally they're providing us with a good estimate of the causal effect (5)
cbind(RDDest$coef, RDDest$ci)
```

```
##                   Coeff CI Lower CI Upper
## Conventional   5.054218 4.848778 5.259658
## Bias-Corrected 5.049684 4.844244 5.255123
## Robust         5.049684 4.804010 5.295357
```

```
# Lin's estimator
Greg = lm(y ~ z + x + z*x)
cbind(coef(Greg)[2], confint(Greg, 'zTRUE'))
```

```
##                    2.5 %   97.5 %
## zTRUE 4.97995 4.790766 5.169134
```

```r
# Shrink the real effect
set.seed(1000)
n   = 500
x   = rnorm(n)
y0  = x^2 + rnorm(n, 0, 0.5)
y1  = y0 + 0.5
z   = (x>=0)
y   = z*y1 + (1-z)*y0
png("plot2.png")
plot(y0 ~ x, col = "grey", pch = 19, cex = 0.1,
     ylim = c(min(y), max(y)),
     xlab = "X", ylab = "Y")
points(y1 ~ x, col = "grey", pch = 19, cex = 0.1)
points(y ~ x, col = "black", pch = 19, cex = 0.1)
abline(v = 0, lty = 2)
dev.off()
```

```
## pdf
##   2
```

```r
png('plot3.png')
plot(rdd_data(x=x, y=y,cutpoint=0),
     xlab = "X", ylab = "Y", cex = 0.3)
dev.off()
```

```
## pdf
##   2
```

```r
# The point estimate, variance estimate, and confidence interval for the causal effect
RDDest = rdrobust(y, x)
# Stil, that's a pretty good estimate.
cbind(RDDest$coef, RDDest$ci)
```

```
##                    Coeff  CI Lower  CI Upper
## Conventional   0.5542181 0.3487782 0.7596579
## Bias-Corrected 0.5496835 0.3442436 0.7551234
## Robust         0.5496835 0.3040096 0.7953574
```

```r
# Lin's estimator (Note that the advantage of RDD is that it does not require unconfoundednenss to hold
Greg = lm(y ~ z + x + z*x)
cbind(coef(Greg)[2], confint(Greg, 'zTRUE'))
```

```
##                      2.5 %    97.5 %
## zTRUE 0.4799505 0.2907665 0.6691345
```

```r
y0  = x^2 + rnorm(n, 0, 0.5)
y1  = -1-0.5*x^2 + rnorm(n, 0, 0.5)
z   = (x>=0)
```

```r
y   = z*y1 + (1-z)*y0
png("plot4.png")
plot(y0 ~ x, col = "grey", pch = 19, cex = 0.1,
     ylim = c(min(y), max(y)))
points(y1 ~ x, col = "grey", pch = 19, cex = 0.1)
points(y ~ x, col = "black", pch = 19, cex = 0.1)
abline(v = 0, lty = 2)
dev.off()
```

```
## pdf
##   2
```

```r
png('plot5.png')
plot(rdd_data(x=x, y=y,cutpoint=0),
     xlab = "X", ylab = "Y", cex = 0.3)
dev.off()
```

```
## pdf
##   2
```

```r
# RDD is still robust assuming non-linear trend
RDDest = rdrobust(y, x)
cbind(RDDest$coef, RDDest$ci)
```

```
##                     Coeff  CI Lower   CI Upper
## Conventional   -0.9798502 -1.224561 -0.7351395
## Bias-Corrected -1.0308944 -1.275605 -0.7861837
## Robust         -1.0308944 -1.305329 -0.7564594
```

```r
# Lin's estimator loses it power
Greg = lm(y ~ z + x + z*x)
cbind(coef(Greg)[2], confint(Greg, 'zTRUE'))
```

```
##                         2.5 %      97.5 %
## zTRUE -0.1962193 -0.3689827 -0.02345585
```

Problem 5

```r
data <- matrix(c(0,0,1,1,1,1,
                 0,0,0,0,1,1,
                 0,1,0,1,0,1,
                 74,11514,34,2385,12,9663), nrow = 6,ncol = 4)
# Estimating ITT
ITT <- (data[5,4] + data[6,4])/(data[5,4] + data[6,4] + data[3,4] + data[4,4])
# Estimating the Local Average Treatment Effect
LATE <- (data[4,4] + data[6,4]) / sum(data[c(3,4,5,6),4]) - data[2,4] / sum(data[c(1,2),4])
tau <- LATE / ITT
print(paste("The Average Causal Effect (Estimated by IV) is ", tau, sep = ""))
```

```
## [1] "The Average Causal Effect (Estimated by IV) is 0.0032280386285733"
```

```
IV_Wald = function(Z, D, Y)
{
       tau_D = mean(D[Z==1]) - mean(D[Z==0])
       tau_Y = mean(Y[Z==1]) - mean(Y[Z==0])
       CACE  = tau_Y/tau_D

       return(list(tau_D = tau_D, tau_Y = tau_Y,
                   CACE  = CACE))
}

## IV se via the delta method
IV_Wald_delta = function(Z, D, Y)
{
       est          = IV_Wald(Z, D, Y)
       AdjustedY    = Y - D*est$CACE
       VarAdj       = var(AdjustedY[Z==1])/sum(Z) +
                          var(AdjustedY[Z==0])/sum(1 - Z)
       return(sqrt(VarAdj)/abs(est$tau_D))
}

##IV se via the bootstrap
IV_Wald_bootstrap = function(Z, D, Y, n.boot = 200)
{
       CACEboot  = replicate(n.boot,
                   {
                   bindex = sample(1:length(Z), replace = TRUE)
                   IV_Wald(Z[bindex], D[bindex], Y[bindex])$CACE
                   })

       return(sd(CACEboot))
}

## covariate adjustment in IV analysis
IV_Lin = function(Z, D, Y, X)
{
  X     = scale(as.matrix(X))
  tau_D = lm(D ~ Z + X + Z*X)$coef[2]
  tau_Y = lm(Y ~ Z + X + Z*X)$coef[2]
  names(tau_D) = NULL
  names(tau_Y) = NULL
  CACE  = tau_Y/tau_D

  return(list(tau_D = tau_D, tau_Y = tau_Y,
              CACE  = CACE))
}

## IV_adj se via the delta method
IV_Lin_delta = function(Z, D, Y, X)
{
  X      = scale(as.matrix(X))
  est    = IV_Lin(Z, D, Y, X)
```

```
  betaY1 = lm(Y ~ X, subset = (Z == 1))$coef[-1]
  betaY0 = lm(Y ~ X, subset = (Z == 0))$coef[-1]
  betaD1 = lm(D ~ X, subset = (Z == 1))$coef[-1]
  betaD0 = lm(D ~ X, subset = (Z == 0))$coef[-1]

  AdjustedY1   = Y - X%*%betaY1 -
                   (D - X%*%betaD1)*est$CACE
  AdjustedY0   = Y - X%*%betaY0 -
                   (D - X%*%betaD0)*est$CACE
  VarAdj       = var(AdjustedY1[Z==1])/sum(Z) +
                   var(AdjustedY0[Z==0])/sum(1 - Z)

  return(sqrt(VarAdj)/abs(est$tau_D))
}

##IV_adj se via the bootstrap
IV_Lin_bootstrap = function(Z, D, Y, X, n.boot = 200)
{
  X          = scale(as.matrix(X))
  CACEboot   = replicate(n.boot,
                         {
                           bindex = sample(1:length(Z), replace = TRUE)
                           IV_Lin(Z[bindex], D[bindex], Y[bindex], X[bindex])$CACE
                         })

  return(sqrt(var(CACEboot)))
}
```

Problem 6

```
data <- read.table("fludata.txt")
# Without Covariates
data %>% filter(assign == 1) %>% summarise(mean(receive))
```

```
##   mean(receive)
## 1     0.3077446
```

```
ITT <- data %>% filter(assign == 1) %>% summarise(mean(receive)) - data %>% filter(assign == 0) %>% summ
LATE <- data %>% filter(assign == 1) %>% summarise(mean(outcome)) - data %>% filter(assign == 0) %>% su
tau_IV <- LATE / ITT
# That's identical to the result by the function in class
IV_Wald(data$assign, data$receive, data$outcome)$CACE
```

```
## [1] -0.1245575
```

```
print(paste("The Average Causal Effect (Estimated by 2sls, without covariates) is ", tau_IV, sep = ""))
```

```
## [1] "The Average Causal Effect (Estimated by 2sls, without covariates) is -0.124557482825996"
```

```r
# This process can also be understood as a two-stage least square, which will offer identical results.
# Stage 1
model1 <- lm(receive ~ data$assign, data = data)$fitted.values
# Stage 2
model <- lm(data$outcome ~  model1)
model$coefficients[2]
```

```
##     model1
## -0.1245575
```

```r
# With Covariates (2SLS)
IV_Lin(data$assign, data$receive, data$outcome, data[,c(-1,-2,-3)])$CACE
```

```
## [1] -0.125214
```

```r
# Variance Estimation
IV_Lin_delta(data$assign, data$receive, data$outcome, data[,c(-1,-2,-3)])
```

```
## [1] 0.08844344
```

```r
# CI
print(paste("CI: [", IV_Lin(data$assign, data$receive, data$outcome, data[,c(-1,-2,-3)])$CACE - 1.96*IV_
```

```
## [1] "CI: [-0.298563111659852,0.0481351769249073]"
```

```r
# # Stage 1
# model1 <- lm(receive ~ .-outcome, data = data)
# # Stage 2
# model <- lm(data$outcome ~ data$age+ data$copd+data$dm+data$heartd+data$race+data$race+data$renal+dat
# print(paste("The Average Causal Effect (Estimated by IV, with covariates) is ", model$coefficients['m
```

```r
data <-  read.table("karolinska.txt", header = T)
Y <- as.numeric(data$YearsSurvivingAfterDiagnosis)
X <- as.numeric(data$HighVolTreatHosp)
IV <- as.numeric(data$HighVolDiagHosp)
covariates <- matrix(c(data$FromRuralArea, data$Male, data$AgeAtDiagnosis), ncol = 3)
# Lin's estimator (Unbiased)
IV_Lin(IV, X, Y, covariates)$CACE
```

```
## [1] 0.1849535
```

```r
# Variance estimation
IV_Lin_delta(IV, X, Y, covariates)^2
```

```
## [1] 0.04253986
```

```r
# The result is insignificant
print(paste("CI: [", IV_Lin(IV, X, Y, covariates)$CACE - 1.96*IV_Lin_delta(IV, X, Y, covariates) , ",",I
```

```
## [1] "CI: [-0.21930025888364,0.589207292815465]"
```

```r
# With Covariates (2SLS)
# Stage 1
model1 <- lm(X~IV+covariates + data$YearOfDiagnosis)
# Stage 2
model <- lm(Y~model1$fitted.values + covariates + data$YearOfDiagnosis)
print(paste("The Average Causal Effect (Estimated by 2SLS, with covariates) is ", model$coefficients['m
```

```
## [1] "The Average Causal Effect (Estimated by 2SLS, with covariates) is 0.174521858685822"
```

```r
# Note that 2SLS is in itself biased with even larger std error 0.24
temp <- summary(model)
temp$coefficients[2,2]
```

```
## [1] 0.2357129
```