Kaicheng Luo

Assignment 2

1. Neymanian inference and OLS

Equivalence of the regression coefficient and tau-hat

For a regression problem

$$Y_i = \alpha + \beta z_i + \epsilon_i \tag{1}$$

We have point estimation

$$\hat{\beta} = \frac{\sum (z_i - \bar{z})(Y_i - \bar{Y})}{\sum (z_i - \bar{z})^2} \tag{2}$$

where

$$\bar{z} = \frac{n_1}{n_0} \qquad 1 - \bar{z} = \frac{n_0}{n_1} \tag{3}$$

So we can rewrite $\hat{\beta}$ as

$$\begin{aligned}\hat{\beta} &= \frac{\frac{n_0}{n_1} Y_i(1) - \frac{n_1}{n_0} Y_i(0)}{\frac{n_0 n_1 (n_0 + n_1)}{n^2}} \\ &= \frac{1}{n_1} Y_i(1) - \frac{1}{n_0} Y_i(0) \\ &= \hat{\tau}\end{aligned} \tag{4}$$

The inconsistency of normal variance estimator

In an OLS estimation, we have

$$var(\hat{\beta}) = \frac{\sigma^2}{\sum (z_i - \bar{z})^2} = \frac{\sum \epsilon^2}{(n-2) \sum (z_i - \bar{z})^2} \tag{5}$$

Due to the OLS property,

$$\sum \epsilon_i^2 = \sum (Y_i - \hat{\alpha} - \hat{\beta} z_i)^2 = \sum (Y_i(1) - \bar{Y(1)})^2 + \sum (Y_i(0) - \bar{Y(0)})^2 \tag{6}$$

With a large sample size,

$$\lim_{n \to \infty} var(\hat{\beta}) = \frac{n_1 \hat{S}_1^2 + n_0 \hat{S}_0^2}{n_0 n_1} \frac{n}{n-2} = \frac{\hat{S}_1^2}{n_0} + \frac{\hat{S}_0^2}{n_1} \tag{7}$$

$$\lim_{n \to \infty} var(\hat{\beta}) \neq \lim_{n \to \infty} \hat{V} = \frac{\hat{S}_1^2}{n_1} + \frac{\hat{S}_0^2}{n_0} \tag{8}$$

Unless $n_0 \equiv n_1$

The consistency of the White estimator

$$v\tilde{a}r(\hat{\beta}) = \frac{\sum(z_i - \bar{z})^2(Y_i - \hat{\alpha} - \hat{\beta}z_i)^2}{(\sum(z_i - \bar{z})^2)^2} \tag{9}$$

Split the summation into treatment and control group, similarly

$$\epsilon_i^2 = (Y_i - \hat{\alpha} - \hat{\beta}z_i)^2 = z_i(Y_i(1) - Y\bar{(1)})^2 + (1 - z_i)(Y_i(0) - Y\bar{(0)})^2 \tag{10}$$

$$\lim_{n\to\infty} v\tilde{a}r(\hat{\beta}) = \frac{\frac{n_0 n_1^2}{n^2}\hat{S}_0^2 + \frac{n_1 n_0^2}{n^2}\hat{S}_1^2}{(\frac{n_0 n_1}{n})^2} = \frac{n_1\hat{S}_0^2 + n_0\hat{S}_1^2}{n_0 n_1} = \hat{V} \tag{11}$$

Problem 2

Playing with a toy model of SRE

```
# FRT for the Project STAR data in the Imbens-Rubin book
# Note that here we collect the image manually
data <- read_excel("STAR.xlsx")

# Preparations
# Step 1: Write a function that gives you all the statistics you want in SRE
stat_SRE <- function(stratum, treatment, y){
  # Assume in our case that the stratum in arranged and indexed.
  # If not, then re-code it to an index.
  number = length(unique(stratum))
  tau = 0
  wil = 0
  r = 0
  # Calculate the three statistics as defined
  for (i in 1:number){
    tempy = y[stratum == i]
    tempt = treatment[stratum == i]
    n = length(tempy)
    pi = n/length(y)
    tau = tau + pi*(mean(tempy[tempt == 1] - mean(tempy[tempt == 0])))
    wil = wil + wilcox.test(tempy[tempt == 1], tempy[tempt == 0])$statistic / (n+1)
    tempy = tempy - mean(tempy)
  }
  y <- rank(y)
  for (i in 1:length(y)){
```

```
    if (treatment[i] == 1){
      r = r + y[i]
    }
  }
  return(c(taus = tau, wilcoxon = wil, alignedRank = r))
}
# Then we can calculate the observed value
obsValue <- stat_SRE(data$Stratum, data$Treatment, data$Y)
obsValue
```

```
##          taus   wilcoxon.W  alignedRank
##     0.2278897    8.3000000 1362.5000000
```

```
# Step 2: Write a function that permutes your data in strata
permute <- function(stratum, treatment){
  ptreat <- vector()
  for (i in 1:length(unique(stratum))){
    ptreat <- c(ptreat, sample(treatment[stratum == i]))
  }
  return(ptreat)
}
```

```
# Step 3: Carry out Stratified Randomization Test
MC = 2000
extreme = rep(0,3)
for (i in 1:MC){
  mcStat = stat_SRE(data$Stratum, permute(data$Stratum, data$Treatment), data$Y)
  for (j in 1:3){
    if (abs(mcStat[j]) > abs(obsValue[j])){
      extreme[j] = extreme[j] + 1
    }
  }
}
# Tidy display of our result
display <- data.frame("Taus" = extreme[1]/MC, "V" = extreme[2]/MC, "Aligned Rank" = extr
display
```

```
##     Taus     V Aligned.Rank
## 1 0.0325 0.086       0.0235
```

```
# At 95% significance level, we rejeect the sharp null hypothesis that there's no sign
```

Problem 3

```
# Baseline Model with NO Strata
# Compare it with the normal complete randomized experiment
# Part 1: FRT
# Step 4: Compare our results with the CRE
library(Matching)
data("lalonde")
z <- lalonde$treat
y <- lalonde$re78

# Monte-Carlo Simulation of data
MC = 2000
Tauhat   = rep(0, MC)
Student  = rep(0, MC)
Wilcox   = rep(0, MC)
Ks       = rep(0, MC)
tau = t.test(y ~ z, var.equal = TRUE)$statistic
t = t.test(y ~ z, var.equal = FALSE)$statistic
w = wilcox.test(y ~ z)$statistic
ks = ks.test(y[z == 1], y[z == 0])$statistic

extreme_tau = 0
extreme_t = 0
extreme_w = 0
extreme_ks = 0
for(mc in 1:MC){
   zperm = sample(z)
   temptau = t.test(y ~ zperm, var.equal = TRUE)$statistic
   tempt = t.test(y ~ zperm, var.equal = FALSE)$statistic
   tempw = wilcox.test(y ~ zperm)$statistic
   tempks = ks.test(y[zperm == 1], y[zperm == 0])$statistic
   if (abs(temptau) > abs(tau)){
     extreme_tau <- extreme_tau + 1
   }
   if (abs(tempt) > abs(t)){
     extreme_t <- extreme_t + 1
   }
   if (abs(tempw) < abs(w)){
     extreme_w <- extreme_w + 1
   }
```

```
  if (abs(tempks) > abs(ks)){
    extreme_ks <- extreme_ks + 1
  }
}
# Tidy display of our result
display_CRE <- data.frame("Tau" = extreme_tau/MC, "t" = extreme_t/MC, "Wilcoxon" = extre
display_CRE
```

```
##      Tau     t Wilcoxon    KS
## 1 0.0055 0.009   0.0055 0.039
```

```
# Part 2: Neymanian Inference
library(Matching)
data(lalonde)
```

```
z = lalonde$treat
y = lalonde$re78

## Neymanian inference
n1= sum(z)
n0= length(z) - n1
tauhat = mean(y[z==1]) - mean(y[z==0])
vhat   = var(y[z==1])/n1 + var(y[z==0])/n0
sehat  = sqrt(vhat)
tauhat
```

```
## [1] 1794.343
```

```
sehat
```

```
## [1] 670.9967
```

```
# Step 0: Some data-cleaning presumed here as I'm implementing my own function of SRE
library(Matching)
data(lalonde)
data <- lalonde
data <- data %>%
  mutate(race = ifelse(black==1, 1, 0)) %>%
  mutate(race = ifelse(hisp == 1, 2, race))
data <- data[,c(-3,-4)]
data$race <- data$race + 1
data <- data %>%
  arrange(by = race)
```

```
# Step 1: Pretend that the SRE is done by blocking race
# Part 1: Fisher Randomization test
MC = 2000
extreme = rep(0,3)
obsValue <- stat_SRE(data$race, data$treat, data$re78)
obsValue
```

```
##        taus  wilcoxon.W alignedRank
##  1794.96905    60.50269 44607.50000
```

```
for (i in 1:MC){
  mcStat = stat_SRE(data$race, permute(data$race, data$treat), data$re78)
  for (j in 1:3){
    if (abs(mcStat[j]) > abs(obsValue[j])){
      extreme[j] = extreme[j] + 1
    }
  }
}
# Tidy display of our result
display1 <- data.frame("Taus" = extreme[1]/MC, "V" = extreme[2]/MC, "Aligned Rank" = ext
display1
```

```
##    Taus      V Aligned.Rank
## 1 0.005 0.0045        0.004
```

```
# Step 1: Pretend that the SRE is done by blocking race
# Part 2: Neymanian Inference
print(c ("The point estimator is", obsValue[1]))
```

```
##                                              taus
## "The point estimator is"       "1794.96904513932"
```

```
var_neyman <- function(stratum, treatment, y){
  V = 0
  for(i in 1:length(unique(stratum))){
    tempy = y[stratum == i]
    tempt = treatment[stratum == i]
    n = length(tempy)
    y0 = tempy[tempt == 0]
    y1 = tempy[tempt == 1]
    V = V + (length(y0)/n)^2 * (sd(y0)/length(y0) + sd(y1)/length(y1))
  }
  return(V)
}
```

```
SRE_race <- var_neyman(data$race, data$treat, data$re78)
SRE_race
```

```
## [1] 610.8122
```

```
# Step 2: Pretend that the SRE is done by blocking marital status
# Part 1: FRT
data$married <- data$married + 1
data <- data %>% arrange(by=married)
MC = 2000
extreme = rep(0,3)
obsValue <- stat_SRE(data$married, data$treat, data$re78)
obsValue
```

```
##        taus  wilcoxon.W alignedRank
##  1767.17517    61.02082 44607.50000
```

```
for (i in 1:MC){
  mcStat = stat_SRE(data$married, permute(data$married, data$treat), data$re78)
  for (j in 1:3){
    if (abs(mcStat[j]) > abs(obsValue[j])){
      extreme[j] = extreme[j] + 1
    }
  }
}
# Tidy display of our result
display2 <- data.frame("Taus" = extreme[1]/MC, "V" = extreme[2]/MC, "Aligned Rank" = ext
display2
```

```
##     Taus     V Aligned.Rank
## 1 0.0035 0.004        0.004
```

```
# Step 2: Pretend that the SRE is done by blocking marital status
# Part 2: Neymanian Inference
SRE_marriage <- var_neyman(data$married, data$treat, data$re78)
SRE_marriage
```

```
## [1] 127.7127
```

```
# Step 3: Pretend that the SRE is done by blocking nodegr
# Part 1: FRT
data$nodegr = data$nodegr + 1
data <- data %>% arrange(by = nodegr)
MC = 2000
extreme = rep(0,3)
```

```
obsValue <- stat_SRE(data$nodegr, data$treat, data$re78)
obsValue
```

```
##        taus  wilcoxon.W alignedRank
##  1598.28122    59.17541 44607.50000
```

```
for (i in 1:MC){
  mcStat = stat_SRE(data$nodegr, permute(data$nodegr, data$treat), data$re78)
  for (j in 1:3){
    if (abs(mcStat[j]) > abs(obsValue[j])){
      extreme[j] = extreme[j] + 1
    }
  }
}
# Tidy display of our result
display3 <- data.frame("Taus" = extreme[1]/MC, "V" = extreme[2]/MC, "Aligned Rank" = ext
display3
```

```
##     Taus     V Aligned.Rank
## 1 0.0135 0.014       0.0125
```

```
# Step 3: Pretend that the SRE is done by blocking nodegr
# Part 2: Neymanian Inference
SRE_edu <- var_neyman(data$nodegr, data$treat, data$re78)
SRE_edu
```

```
## [1] 88.97333
```

3.2 Regression adjusments for Penn

```
penndata = read.table("Penn46_ascii.txt")
```

```
z = penndata$treatment
penndata$duration = log(penndata$duration)
y = lm(duration ~ .-treatment, data = penndata)$residuals
penndata <- penndata %>%
  mutate(quarter = quarter + 1) %>%
  arrange(by = quarter)
obsValue = stat_SRE(penndata$quarter, penndata$treatment, y)
# The point estimator
obsValue[1]
```

```
##        taus
```

```
## -0.01150982
```

```r
SRE_adjusted <- var_neyman(penndata$quarter, penndata$treatment, y)
SRE_adjusted
```

```
## [1] 0.02450576
```

```r
# Interval estimation
print(paste("[",obsValue[1] - SRE_adjusted*1.96,",",obsValue[1] + SRE_adjusted*1.96,"]"))
```

```
## [1] "[ -0.0595410962445224 , 0.0365214636256135 ]"
```

```r
Neyman_SRE = function(z, y, x)
{
       xlevels = unique(x)
       K       = length(xlevels)
       PiK     = rep(0, K)
       TauK    = rep(0, K)
       varK    = rep(0, K)
       for(k in 1:K)
       {
             xk         = xlevels[k]
             zk         = z[x == xk]
             yk         = y[x == xk]
             PiK[k]     = length(zk)/length(z)
             TauK[k]    = mean(yk[zk==1]) - mean(yk[zk==0])
             varK[k]    = var(yk[zk==1])/sum(zk) +
                               var(yk[zk==0])/sum(1 - zk)
       }

       return(c(sum(PiK*TauK), sum(PiK^2*varK)))
}

## pennsylvania re-employment bonus experiment
## description of the DATA:
## Koenker and Xiao 2002 Econometrica
## "Inference on the Quantile Regression Process"

penndata = read.table("Penn46_ascii.txt")
```

```r
z = penndata$treatment
y = log(penndata$duration)
block = penndata$quarter
est = Neyman_SRE(z, y, block)
```

```r
est[1]
```

```
## [1] -0.08990646
```

```r
sqrt(est[2])
```

```
## [1] 0.03079775
```

```r
print(paste("[",est[1]-1.96*sqrt(est[2]),",",est[1]+sqrt(est[2]),"]"))
```

```
## [1] "[ -0.150270048386588 , -0.0591087093146378 ]"
```

4. Regression adjustment / post-stratification of CRE

Proof :

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 z_i + \beta_2 X_i + \beta_3 z_i X_i \\ Y_i(1) &= \beta_0 + \beta_1 + (\beta_2 + \beta_3) X_i \\ Y_i(0) &= \beta_0 + \beta_2 X_i \end{aligned} \tag{12}$$

Denote γ_j as the estimated intercept given $z_i = j$, j = 1,2

$$\gamma_1 - \gamma_2 = (\bar{Y}_i(1) - \hat{\alpha}_1 \bar{X}_i) - (\bar{Y}_i(0) - \hat{\alpha}_2 \bar{X}_i) \tag{13}$$

Note that

$$X_i = \delta_{1i} - \pi_{[i]} \qquad \bar{X}_i = 0 \tag{14}$$

$$\begin{aligned} \tau_{PS} &= \frac{1}{\pi[1]}[\bar{Y}(1)_{x=1} - \bar{Y}(0)_{x=1}] + \frac{1}{\pi[0]}[\bar{Y}(1)_{x=0} - \bar{Y}(0)_{x=0}] \\ &= \frac{1}{\pi[1]}\bar{Y}(1)_{x=1} + \frac{1}{\pi[0]}\bar{Y}(1)_{x=0} - \frac{1}{\pi[1]}\bar{Y}(0)_{x=1} - \frac{1}{\pi[0]}\bar{Y}(0)_{x=0} \end{aligned} \tag{15}$$

Consider the normal equations of the regression

$$\sum Y_i = n_1 \gamma_1 + \sum \beta_1 X_i \tag{16}$$

$$\sum X_i Y_i = \gamma_1 \sum X_i + \sum \beta_1 X_i^2 \quad where\, X_i = X_i^2 \tag{17}$$

Subtract the latter by the former, we have $\gamma_1 = \frac{1}{\pi[1]}\bar{Y}(1)_{x=1} + \frac{1}{\pi[0]}\bar{Y}(1)_{x=0}$, indicating that the intercept is simply the weighted average of the subset Y.
Similarly, for γ_0 we have $\gamma_0 = \frac{1}{\pi[1]}\bar{Y}(0)_{x=1} + \frac{1}{\pi[0]}\bar{Y}(0)_{x=0}$

$$\tau_L = \gamma_1 - \gamma_0 = \tau_{PS} \tag{18}$$

5. Additional comments on the Neymanian inference under an SRE

By the definition of $e_{[k]} \equiv e$, we have

$$\mathbb{P}(z_{[K]i} = 1 | K = k) = const \tag{19}$$

So $\frac{n_{[k]1}}{n_1} = \pi_k$, $\frac{n_{[k]0}}{n_0} = \pi_k$

$$\begin{aligned}
\hat{\tau}_s &= \sum_{i=1}^{n} \pi_i \hat{\tau}_{[i]} \\
&= [\sum_{l=1}^{n} \frac{n_{[k]1}}{n_1} \frac{1}{n_{[k]1}} Y_{[k]}(1) - \sum_{k=1}^{n} \frac{n_{[k]0}}{n_0} \frac{1}{n_{[k]0}} Y_{[k]}(0)] \\
&= \frac{1}{n_1} \sum Y_{[k]}(1) - \frac{1}{n_0} \sum Y_{[k]}(0) \\
&= \frac{1}{n_1} \sum Y(1) - \frac{1}{n_0} \sum Y(0) \\
&= \hat{\tau}
\end{aligned} \tag{20}$$