PS5

Tate Mason

Question 1

```r
library(haven)
library(dplyr)
library(magrittr)
```

Part A

```r
data <- read_dta("cps09mar.dta")
df <- data %>% filter(female == 0) %>% mutate(black = as.integer(race == 2))
X <- model.matrix(~ age + education + black + hisp, data = df)
Y <- df$union
n <- nrow(X)
```

```r
log_likelihood <- function(beta, X, Y) {
  xb <- X %*% beta
  p <- pnorm(xb)
  loglik <- sum(Y*log(p) + (1-Y)*log(1-p))
  return(-loglik)
}

init <- rep(0, ncol(X))
fit <- optim(
  init,
  log_likelihood,
  X = X,
  Y = Y,
  hessian = TRUE,
```

```r
  method = "BFGS"
)

beta_hat <- fit$par
vcov_b <- solve(fit$hessian)
se_b <- sqrt(diag(vcov_b))

tibble(
  Variable = colnames(X),
  Estimate = beta_hat,
  Std_Error = se_b
)
```

```
# A tibble: 5 x 3
  Variable    Estimate Std_Error
  <chr>          <dbl>     <dbl>
1 (Intercept) -1.95      0.108
2 age          0.00778   0.00143
3 education   -0.0256    0.00631
4 black       -0.0542    0.0600
5 hisp        -0.300     0.0565
```

```r
glm <- glm(Y~age+education+black+hisp, family=binomial(link="probit"), data = df)
summary(glm)$coefficients %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Variable") %>%
  rename(
    Estimate = Estimate,
    Std_Error = 'Std. Error',
    z_value = 'z value',
    p_value = 'Pr(>|z|)'
  )
```

```
     Variable     Estimate   Std_Error     z_value      p_value
1 (Intercept) -1.953818176 0.107176068 -18.2299856 2.983996e-74
2         age  0.007925402 0.001419363   5.5837737 2.353549e-08
3   education -0.025504590 0.006221006  -4.0997536 4.135902e-05
4       black -0.054082971 0.060004104  -0.9013212 3.674176e-01
5        hisp -0.297744914 0.056864898  -5.2360054 1.640891e-07
```

Part B

```r
phi <- dnorm(X%*%beta_hat)
APE <- colMeans(X*as.vector(phi))
APE
```

```
(Intercept)         age   education       black        hisp
0.053127456 2.350458351 0.725066488 0.004330912 0.005199222
```

```r
tibble(
  Variable = colnames(X),
  APE = APE
)
```

```
# A tibble: 5 x 2
  Variable        APE
  <chr>         <dbl>
1 (Intercept) 0.0531
2 age         2.35
3 education   0.725
4 black       0.00433
5 hisp        0.00520
```

Part C

```r
G <- t(X)%*%(X * as.vector(phi))/n
ape_var <- G%*%vcov_b%*%t(G)
ape_se <- sqrt(diag(ape_var))

tibble(
  Variable = colnames(X),
  APE = APE,
  Analytical_SE = ape_se
)
```

```
# A tibble: 5 x 3
  Variable        APE Analytical_SE
  <chr>         <dbl>         <dbl>
```

```
1 (Intercept) 0.0531      0.000869
2 age         2.35        0.0400
3 education   0.725       0.0121
4 black       0.00433     0.000247
5 hisp        0.00520     0.000262
```

Part D

```
boot_fn <- function(data, indices) {
  d <- data[indices, ] %>%
    mutate(Y = ifelse(union==1, 1, 0))
  X_b <- model.matrix(~ age+education+black+hisp, data=d)
  Y_b <- d$Y

  loglik_b <- function(beta) {
    xb_b <- X_b %*% beta
    p_b <- pnorm(xb_b)
    -sum(Y_b*log(p_b) + (1-Y_b)*log(1-p_b))
  }

  fit_b <- optim(
    rep(0, ncol(X_b)),
    loglik_b,
    hessian = FALSE
  )
  beta_b <- fit_b$par
  phi_b <- dnorm(X_b%*%beta_b)
  ape_b <- colMeans(X_b*as.vector(phi_b))
  return(ape_b)
}

set.seed(19)
R <- 500
boot_apes <- matrix(NA, nrow = R, ncol=ncol(X))

for (r in 1:R) {
  sample_idx <- sample(1:nrow(df), replace = TRUE)
  boot_data <- df[sample_idx, ] %>%
    mutate(Y = ifelse(union == 1, 1, 0))

  X_b <- model.matrix(~ age+education+black+hisp, data = boot_data)
```

```r
  Y_b <- boot_data$Y

  loglik <- function(beta) {
    xb <- X_b %*% beta
    p <- pnorm(xb)
    -sum(Y_b*log(p) + (1-Y_b)*log(1-p))
  }
  opt <- optim(
    rep(0, ncol(X_b)),
    loglik,
    hessian = FALSE,
    method = "BFGS"
  )
  beta_b <- opt$par

  phi_b <- dnorm(X_b %*%beta_b)
  ape_b <- colMeans(X_b*as.vector(phi_b))
  boot_apes[r, ] <- ape_b
}

boot_se <- apply(boot_apes, 2, sd)

tibble(
  Variable = colnames(X),
  APE = APE,
  Analytical_SE = ape_se,
  Bootstrap_SE = boot_se
)
```

```
# A tibble: 5 x 4
  Variable        APE Analytical_SE Bootstrap_SE
  <chr>         <dbl>         <dbl>        <dbl>
1 (Intercept) 0.0531       0.000869     0.00170
2 age         2.35         0.0400       0.0765
3 education   0.725        0.0121       0.0240
4 black       0.00433      0.000247     0.000494
5 hisp        0.00520      0.000262     0.000616
```

Question 2

```r
library(haven)
library(dplyr)
library(ranger)
library(tidyr)
```

```r
df <- read_dta("~/Downloads/jtrain_observational.dta") %>%
  mutate(across(c(train, black, hisp, married), as.numeric)) %>%
  drop_na(re78, train, age, educ, black, hisp, married, re75, unem75)
vars <- c("age", "educ", "black", "hisp", "married", "re75", "unem75")
x_matrix <- function(df) {
  cbind(1, as.matrix(df %>% select(all_of(vars))))
}
```

Part A

```r
ols <- function(X, Y) {
  beta_hat <- solve(t(X)%*%X) %*% t(X)%*%Y
  return(beta_hat)
}

X0 <- x_matrix(df %>% filter(train==0))
Y0 <- df %>% filter(train==0) %>% pull(re78)
beta_0 <- ols(X0, Y0)

X1 <- x_matrix(df %>% filter(train == 1))
Y1 <- df %>% filter(train == 1) %>% pull(re78)
y0_hat <- X1 %*% beta_0

att_hat <- mean(Y1) - mean(y0_hat)
att_hat
```

```
[1] 0.8588455
```

Part B

```r
resid <- Y1 - y0_hat
n1 <- length(Y1)
att_se <- sd(resid)/sqrt(n1)
att_se
```

```
[1] 0.597038
```

Part C

```r
set.seed(19)
R <- 500
boot_att <- numeric(R)

for (r in 1:R) {
  samp_idx <- sample(1:nrow(df), replace=TRUE)
  d_b <- df[samp_idx, ]

  X0_b <- x_matrix(d_b %>% filter(train==0))
  Y0_b <- d_b %>% filter(train == 0) %>% pull(re78)
  beta_0b <- ols(X0_b, Y0_b)

  X1_b <- x_matrix(d_b %>% filter(train==1))
  Y1_b <- d_b %>% filter(train==1) %>% pull(re78)
  y0_hat_b <- X1_b %*% beta_0b

  boot_att[r] <- mean(Y1_b) - mean(y0_hat_b)
}

boot_se <- sd(boot_att)
boot_se
```

```
[1] 0.8859035
```

Part D

```r
X_full <- cbind(1, as.matrix(df %>% select(train, all_of(vars))))
Y_full <- df$re78
beta_full <- ols(X_full, Y_full)
att_ols <- beta_full[2]
att_ols
```

```
[1] 0.5249489
```

Part E

```r
ps <- glm(train ~ age+educ+black+hisp+married+re75+unem75, data=df, family=binomial)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
ps_hat <- predict(ps, type = "response")

w <- ps_hat/(1-ps_hat)
treated <- df$train==1

att_ps <- mean(df$re78[treated]) - weighted.mean(df$re78[!treated], w=w[!treated])
att_ps
```

```
[1] 0.5695314
```

Part F

```r
mod_y1 <- glm(re78 ~ age + educ + black + hisp + married + re75 + unem75,
              data = df %>% filter(train == 1))
mod_y0 <- glm(re78 ~ age + educ + black + hisp + married + re75 + unem75,
              data = df %>% filter(train == 0))

m1_hat <- predict(mod_y1, newdata = df)
m0_hat <- predict(mod_y0, newdata = df)

aipw <- mean(df$train * (df$re78 - m0_hat) / ps_hat + m1_hat - m0_hat)
aipw
```

```
[1] -69.8642
```

Part G

```
ml_ps <- ranger(train ~ ., data = df %>% select(train, all_of(vars)), probability=TRUE)
ml_ps_hat <- predict(ml_ps, data=df)$predictions[, 2]

ml_y1 <- ranger(re78 ~., data = df %>% filter(train == 1) %>% select(re78, all_of(vars)))
ml_y0 <- ranger(re78 ~., data = df %>% filter(train == 0) %>% select(re78, all_of(vars)))

m1_ml <- predict(ml_y1, data=df)$predictions
m0_ml <- predict(ml_y0, data=df)$predictions

att_ml <- mean(df$train*(df$re78-m0_ml)/ml_ps_hat+m1_ml-m0_ml)
att_ml
```

```
[1] -10.15842
```