Regression Analysis HW2

Tuorui Peng, tuoruipeng2028@u.northwestern.edu

(a)

Since S_0 and S_1 are disjoint, we have independence between $\{X_1, Y_1\}$ and $\{X_2, Y_2\}$. We could obtin:

$$\begin{aligned}
\hat{\beta}_0 =&(X_0'X_0)^{-1}X_0'Y_0\\
\hat{\beta}_1 =&(X_1'X_1)^{-1}X_1'Y_1\\
\hat{\varepsilon}_0 =&Y_0 - X_0\hat{\beta}_0 := (I-H_0)Y_0\\
\hat{\varepsilon}_1 =&Y_1 - X_1\hat{\beta}_1 := (I-H_1)Y_1
\end{aligned}$$

Distribution under normality assumption:

$$\begin{aligned}
\Delta \sim& N(0, \sigma^2(X_0'X_0)^{-1} + \sigma^2(X_1'X_1)^{-1})\\
\delta \sim& N(0, \sigma^2(I-H_0) + \sigma^2(I-H_1))
\end{aligned}$$

Under normality assumption, it suffices to show the covariance between $\Delta = \hat{\beta}_0 - \hat{\beta}_1$ and $\delta = \hat{\varepsilon}_0 - \hat{\varepsilon}_1$ is 0, i.e.

$$\begin{aligned}
cov(\Delta, \delta) =& cov(\hat{\beta}_0 - \hat{\beta}_1, \hat{\varepsilon}_0 - \hat{\varepsilon}_1)\\
=& cov(\hat{\beta}_0, \hat{\varepsilon}_0) + cov(\hat{\beta}_1, \hat{\varepsilon}_1)\\
=& cov((X_0'X_0)^{-1}X_0'(X_0\beta + \varepsilon_0), (I-H_0)(X_0\beta + \varepsilon_0))\\
&+ cov((X_1'X_1)^{-1}X_1'(X_1\beta + \varepsilon_1), (I-H_1)(X_1\beta + \varepsilon_1))\\
=& (X_0'X_0)^{-1}X_0'\sigma^2 I(I-H_0) + (X_1'X_1)^{-1}X_1'\sigma^2 I(I-H_1) \overset{(i)}{=} 0
\end{aligned}$$

which proves $\Delta \perp\!\!\!\perp \delta$. Here (i) uses the fact that $(I-H_i)X_i = 0$, $i = 0, 1$.

(b)

Denote the QR decomposition of B as $B = QR$, in which $R \in \mathbb{R}^{n\times d}$ is an upper triangular matrix. The first column of B is $\mathbf{1}$ yields $q_1 := Q_{:,1} = \dfrac{1}{\sqrt{n}}\mathbf{1}_n$ and $R_{11} = \sqrt{n}$. Then we have:

$$\begin{aligned}
P =& B(B'B)^{-1}B' - \frac{1}{n}\mathbf{1}\mathbf{1}' = QR(R'R)^{-1}R'Q' - q_1q_1'\\
=& QRR^{-1}(R')^{-1}R'Q' - q_1q_1'\\
=& QQ' - q_1q_1'
\end{aligned}$$

Obviously P is symmetric. Further for idempotence, note that $q_1'Q = [1, 0, \dots, 0]$, we can check that

$$\begin{aligned}
P^2 =& (QQ' - q_1q_1')(QQ' - q_1q_1') \\
=& QQ' - QQ'q_1q_1' - q_1q_1'QQ' + q_1q_1'q_1q_1' \\
=& QQ' - q_1q_1' = P
\end{aligned}$$

Thus P is a projection matrix, and further

$$\frac{1}{\sqrt{n}}P\mathbf{1} = (QQ' - q_1q_1')q_1 = Q(Q' - q_1q_1')q_1 = Q\mathbf{0} = \mathbf{0}$$

(c)

Since we have distributions

$$\begin{aligned}
\Delta \sim& N(0, \sigma^2(X_0'X_0)^{-1} + \sigma^2(X_1'X_1)^{-1}) \\
\delta \sim& N(0, \sigma^2(I - H_0) + \sigma^2(I - H_1)) = N(0, \sigma^2(2I - H_0 - H_1))
\end{aligned}$$

For A, since $(X_0'X_0)^{-1} + (X_1'X_1)^{-1}$ has full rank, we can use the eigen decomposition $(X_0'X_0)^{-1} + (X_1'X_1)^{-1} := P_\Delta \Lambda_\Delta P_\Delta'$ to obtain A as follows:

$$A = P_\Delta \Lambda_\Delta^{-1/2} P_\Delta'$$

in which $\Lambda^{-1/2}$ is defined element-wise. In this way we can verify that

$$var(A\Delta) = Avar(\Delta)A' = P_\Delta \Lambda_\Delta^{-1/2} P_\Delta' \sigma^2 P_\Delta \Lambda_\Delta P_\Delta' P_\Delta \Lambda_\Delta^{-1/2} P_\Delta' = \sigma^2 I$$

For M we similarly use eigen decomposition $2I - H_0 - H_1 := P_\delta \Lambda_\delta P_\delta'$. Here notice that possibly we have some diagonal elements of Λ_δ being 0. WLOG say the last r ones are 0. In this case we make the following notation:

$$\tilde{\Lambda}_\delta = (\Lambda_\delta)_{1:(n-r),1:(n-r)}, \quad \tilde{P}_\delta = (P_\delta)_{:,1:(n-r)}$$

i.e. the eigenvalues-eigenvector paris of $2I - H_0 - H_1$ with none-zero eigenvalues. Then we can write M as

$$M = P_\delta' \tilde{P}_\delta \tilde{\Lambda}_\delta^{-1/2} \tilde{P}_\delta' P_\delta$$

in this way we can verify that

$$\begin{aligned}
var(M\delta) =& Mvar(\delta)M' \\
=& P_\delta' \tilde{P}_\delta \tilde{\Lambda}_\delta^{-1/2} \tilde{P}_\delta' P_\delta \sigma^2 \tilde{P}_\delta \tilde{\Lambda}_\delta \tilde{P}_\delta' P_\delta P_\delta' \tilde{P}_\delta \tilde{\Lambda}_\delta^{-1/2} \tilde{P}_\delta' P_\delta \\
=& \sigma^2 P_\delta' \tilde{P}_\delta \tilde{\Lambda}_\delta^{-1/2} \tilde{\Lambda}_\delta \tilde{\Lambda}_\delta^{-1/2} \tilde{P}_\delta' P_\delta \\
=& \sigma^2 P_\delta' \tilde{P}_\delta \tilde{P}_\delta' P_\delta \\
=& \sigma^2 \begin{bmatrix} I_{n-r} & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

(d)

Note that now we have

$$A\Delta \sim N(0, \sigma^2 I), \quad M\delta \sim N(0, \sigma^2 \mathrm{diag}(I_{n-r}, 0))$$

and further they are independent. We can construct

$$\hat{F} = \frac{\|A\Delta\|_2^2/d}{\|M\delta\|_2^2/(n-r)} \sim F_{d,n-r}$$

(e)

In this part we use α threshold for

$$\text{reject if } \hat{F} > F_{d,n-r}(\alpha)$$

w.r.t. H_0 : No heteroskedasticity.

because intuitively a large $\hat{F}$ means deviation in δ is small, suggesting a 'resonance' between two groups, i.e. heteroskedasticity behaviour.

```
dat <- read.csv('maybe-its-nonlinear.csv', header = FALSE, sep = ' ')
names(dat) <- c('x1', 'x2', 'y')
dat$intercept <- 1
dat <- dat[, c('intercept', 'x1', 'x2', 'y')]

mat_inverse_sqrt <- function(mat){
    a <- eigen(mat)
    idx <- which(a$value > 1e-8)
    return(a$vector[, idx] %*% diag(1 / sqrt(a$value[idx])) %*% t(a$vector[, idx]))
}
Fvalue <- function(S0, S1, dat){
    X0 <- as.matrix(dat[S0, 1:3])
    Y0 <- as.matrix(dat[S0, 4])
    X1 <- as.matrix(dat[S1, 1:3])
    Y1 <- as.matrix(dat[S1, 4])
    X0X0 <- t(X0) %*% X0
    X1X1 <- t(X1) %*% X1
    X0X0inv <- solve(X0X0)
    X1X1inv <- solve(X1X1)
    H0 <- X0 %*% X0X0inv %*% t(X0)
    H1 <- X1 %*% X1X1inv %*% t(X1)
    Delta <-  (X0X0inv %*% t(X0) %*% Y0 - X1X1inv %*% t(X1) %*% Y1)
    delta <- (Y0 - H0 %*% Y0) - (Y1 - H1 %*% Y1)
    # A=((X_0'X_0)^{-1}+ (X_1'X_1)^{-1})^{-1/2}
```

```
    A <- mat_inverse_sqrt(X0X0inv + X1X1inv)
    # M=(2I-(H_0+H_1))^{-1/2}
    M <- mat_inverse_sqrt(2 * diag(nrow(X0)) - H0 - H1)
    # Note: here the M matrix differs from my definition in (c) by a unitary transform $P_\delta$, bu

    dof1 <- ncol(X0)
    dof2 <- nrow(X0)-1 # bacause I included intercept in my regression model, so there would be (at l
    F <- dof2 / dof1 * sum((A %*% Delta)^2) / sum((M %*% delta)^2)
    return(list(F = F, dof1 = dof1, dof2 = dof2))
}
do_reject <- function(F, dof1, dof2, alpha = 0.05){
    return(F > qf(1-alpha, dof1, dof2))
}


#######################
## Simulation Begins ##

set.seed(42)
N <- 1e3

ireject <- 0
iireject <- 0
fi <- c()
fii <- c()
for(j in 1:N){
    ## (i)
    theta <- runif(1, 0, pi)
    phi <- runif(1, 0, 2 * pi)
    v <- c(sin(theta) * cos(phi), sin(theta) * sin(phi), cos(theta))
    # actually it's not a uniform distribution on the sphere, but it's not quite important here.
    xv <- as.matrix(dat[, 1:3]) %*% matrix(v)
    med_xv <- median(xv)
    S0i <- which(xv <= med_xv)
    S1i <- which(xv > med_xv)
    ## (ii)
    S0ii <- sample(1:nrow(dat), nrow(dat) / 2)
    S1ii <- setdiff(1:nrow(dat), S0ii)

    # use threshold alpha = 0.05
    reti <- Fvalue(S0i, S1i, dat)
```

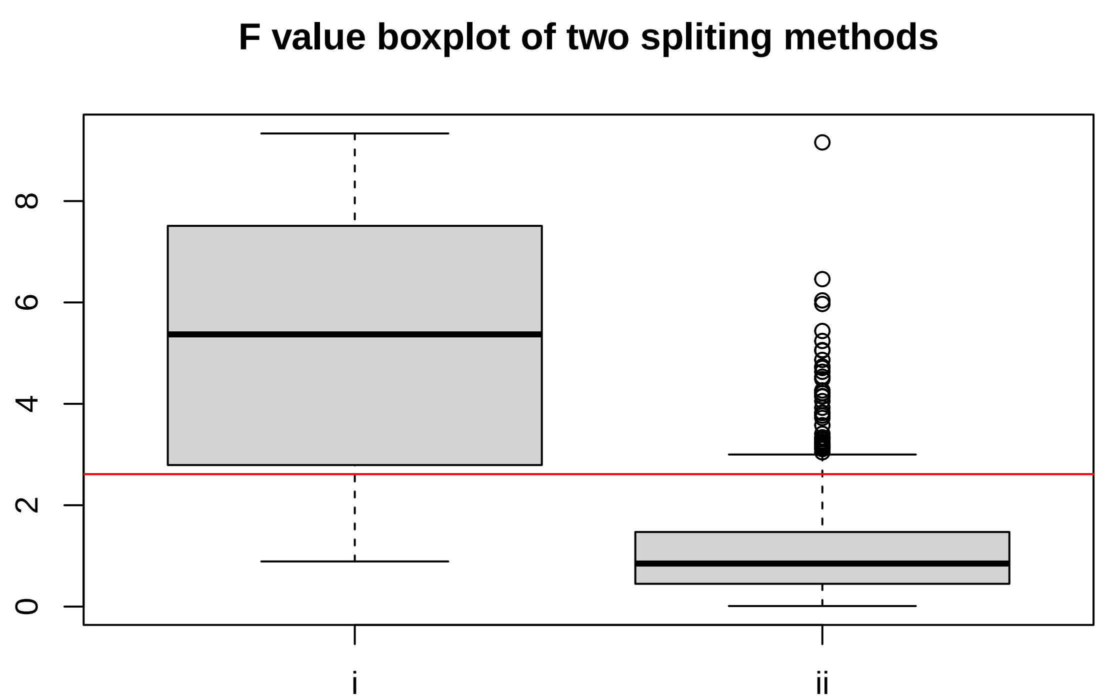
```
    retii <- Fvalue(S0ii, S1ii, dat)
    fi <- c(fi, reti$F)
    fii <- c(fii, retii$F)
    ireject <- ireject + do_reject(reti$F, reti$dof1, reti$dof2)
    iireject <- iireject + do_reject(retii$F, retii$dof1, retii$dof2)
}
write.csv(data.frame(fi, fii), 'Fvalue.csv', row.names = FALSE)


cat('The rejection rate for (i) is', ireject / N, '.')
```

```
## The rejection rate for (i) is 0.762 .
```

```
cat('The rejection rate for (ii) is', iireject / N, '.')
```

```
## The rejection rate for (ii) is 0.066 .
```

```
boxplot(fi, fii, names = c('i', 'ii'), main = 'F value boxplot of two spliting methods')
lines(c(0, 3), c(qf(0.95, 3, 999), qf(0.95, 3, 999)), col = 'red')
```



We can see that proportion of rejection in case (i) is much greater, suggesting a difference in variance between small $\|x\|$ and large $\|x\|$. Such behaviour can also been observed from general method of OLS, where we can observe larger variance for small fitted value.

```
par(mfrow = c(2, 2))
plot(lm(y ~ x1 + x2, data = dat))
```