Regression Analysis HW5

Tuorui Peng, tuoruipeng2028@u.northwestern.edu

Question 2.1

(a)

Using any test function $g \in \mathcal{L}^2(\mathbb{R}^d)$ with $t \in \mathbb{R}$ to denote $f^*(x) = \mathbb{E}[Y|X = x] = (f - tg)(x)$, i.e. $f(x) = \mathbb{E}[Y|X = x] + tg(x)$

$$\begin{aligned}
\mathbb{E}[(Y - f(X))^2] =&\mathbb{E}[(Y - (f^* + tg)(X))^2]\\
=&\mathbb{E}[(Y - f^*(X))^2] - 2t\mathbb{E}[(Y - f^*(X))(g(X))] + t^2\mathbb{E}[(g(X))^2]\\
=&\mathbb{E}[(Y - f^*(X))^2] - 2t\mathbb{E}_X\mathbb{E}_Y[(Y - f^*(X))g(X)|X] + t^2\mathbb{E}[(g(X))^2]\\
=&\mathbb{E}[(Y - f^*(X))^2] + t^2\mathbb{E}[(g(X))^2]\\
\geq&\mathbb{E}[(Y - f^*(X))^2], \quad \forall t \in \mathbb{R},\, g \in \mathcal{L}^2(\mathbb{R}^d)
\end{aligned}$$

which proves that $f^*(x) = \mathbb{E}[Y|X = x]$ is the minimizer of $\mathbb{E}[(Y - f(X))^2]$.

(b)

Minimization of the expected squared loss yields

$$\begin{aligned}
0 = \frac{\partial}{\partial b}L(b) =&\frac{\partial}{\partial b}\mathbb{E}[(Y - X'b)^2] = \mathbb{E}[\frac{\partial}{\partial b}(Y - X'b)^2]\\
=&\mathbb{E}[-2(Y - X'b)X] = \mathbb{E}[2XX'b - 2XY]
\end{aligned}$$

So here again we have the solution as $\hat{\beta} = \mathbb{E}[XX']^{-1}mathbbE[XY]$. To verify its optimality, we denote any elements in linear space as $b = \hat{\beta} + ta$, $t \in \mathbb{R}$, $a \in \mathbb{R}^n$, and follow similar steps as in (a):

$$\begin{aligned}
L(b) = L(\hat{\beta} + ta) =&\mathbb{E}[(Y - X'(\hat{\beta} + ta))^2]\\
=&\mathbb{E}[(Y - X'\hat{\beta})^2] - 2t\mathbb{E}_X\mathbb{E}_Y[(Y - \hat{\beta}'X)(X'a)|X] + t^2\mathbb{E}[(X'a)^2]\\
=&\mathbb{E}[(Y - X'\hat{\beta})^2] - 2t\mathbb{E}_X[\mathbb{E}\left[Y|X\right]\left(I - X'\mathbb{E}\left[X'X\right]X\right)X'a] + t^2\mathbb{E}[(X'a)^2]\\
=&\mathbb{E}[(Y - X'\hat{\beta})^2] + t^2\mathbb{E}[(X'a)^2]\\
\geq&\mathbb{E}[(Y - X'\hat{\beta})^2], \quad \forall t \in \mathbb{R},\, a \in \mathbb{R}^n
\end{aligned}$$

which proves that $\hat{\beta} = \mathbb{E}[XX']^{-1}mathbbE[XY]$ is the minimizer of $\mathbb{E}[(Y - X'b)^2]$.

(c)

Here we use $\{\mathbf{X}, \mathbf{Y}\}$ to represent the sample. From OLS estimator we know that $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. We can see that on population level

$$\beta^* = \mathbb{E}\left[(\mathbf{X}'\mathbf{X})^{-1}\right]\mathbb{E}\left[\mathbf{X}'\mathbf{Y}\right] = \mathbb{E}[\mathbf{X}\mathbf{X}']^{-1}\mathbb{E}[\mathbf{X}\mathbf{Y}]$$

while the estimator

$$\mathbb{E}\left[\hat{\beta}\right] = \mathbb{E}\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}\right] \neq \mathbb{E}[\mathbf{X}\mathbf{X}']^{-1}\mathbb{E}[\mathbf{X}\mathbf{Y}], \quad \text{generally speaking.}$$

Question 2.2

Using the QR decomposition of X, say $X = QR$, we can express

$$Y = X\beta + \epsilon = QR\beta + \epsilon$$

Multiply both sides by Q', and we make some now notation

$$Z = Q'Y = Q'QR\beta + Q'\epsilon := R\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I)$$

And since Q as a unitary matrix, we can solve the problem for Z, R here. With R being upper triangular, and WLOG take $j = d$ we obtain:

$$\begin{aligned}
\hat{\beta}_d &= \frac{z_d}{R_{dd}} \\
var(\hat{\beta}_d) &= \sigma^2 (R'R)^{-1}_{dd} \underset{\text{cramer's rule}}{=} \sigma^2 \frac{\det(R'R)^*_{dd}}{\det(R'R)} = \frac{\sigma^2}{R^2_{dd}} \\
v\hat{a}r(\hat{\beta}_d) &= \frac{\hat{\sigma}^2}{R^2_{dd}} \\
&\Rightarrow \begin{cases} Hz = R\hat{\beta} \\ H_d z = R_{\wedge d}\hat{\beta}_{\wedge d} \end{cases} \quad \Rightarrow (H - H_d)z = R(\hat{\beta} - \hat{\beta}_{\wedge d}) = R_{dd}\frac{z_d}{R_{dd}} = z_d
\end{aligned}$$

with the above results we obtain that

$$t_j^2 = \frac{(\hat{\beta}_d)^2}{\hat{s}e(\hat{\beta}_j)} = \frac{(z_d/R_{dd})^2}{\hat{\sigma}^2/R^2_{dd}} = \frac{\|(H - H_d)z\|_2^2}{\text{MSE}} = F_j, \, \forall j \in [d]$$

Question 2.3

(a)

Note that $(\frac{1}{n}\mathbf{1}\mathbf{1}')^2 = \frac{1}{n}\mathbf{1}\mathbf{1}'$ and that $I' = I$, $(\mathbf{1}\mathbf{1}')' = \mathbf{1}\mathbf{1}'$. Verify directly:

$$(I - \frac{1}{n}\mathbf{1}\mathbf{1}')Z \sim N(0, (I - \frac{1}{n}\mathbf{1}\mathbf{1}')'C_\rho(I - \frac{1}{n}\mathbf{1}\mathbf{1}')) = N(0, (1-\rho)(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}'))$$

(b)

Using the result from the previous question, we have

$$Y_i - \bar{Y}_i\mathbb{1} \sim N(0, (1-\rho)(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}')), \quad i = 1, 2$$

degree of freedom of squared value:

$$\text{dof}\, \frac{(Y_i - \bar{Y}_i\mathbb{1})'(Y_i - \bar{Y}_i\mathbb{1})}{1-\rho} = \text{rank}(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}') = n-1$$

then we have

$$\begin{aligned}\frac{1}{1-\rho}S_n^2 =& \frac{1}{1-\rho}\frac{\sum_{i=1}^n (Y_{1i} - \bar{Y}_1)^2 + \sum_{i=1}^n (Y_{2i} - \bar{Y}_2)^2}{2(n-1)} \\ \sim& \frac{\sigma^2}{2(n-1)}\chi^2_{2n-2}\end{aligned}$$

(c)

Note that

$$\bar{Y}_i = \frac{1}{n}\mathbf{1}'Y_i \sim N(\mu + \alpha_i, \frac{\sigma^2(1-\rho)}{n} + \sigma^2\rho)$$

then we have

$$\bar{Y}_1 - \bar{Y}_2 \sim N(0, 2\sigma^2(\frac{1-\rho}{n} + \rho))$$

(d)

Since both $\bar{Y}_1 - \bar{Y}_2$ and $Y_{ij} - \bar{Y}_i$ $\forall i \in \{1,2\}, 1 \le j \le n$ are normal, and S_n^2 is function of $Y_{ij} - \bar{Y}_i$, it suffices to show that

$$cov(\bar{Y}_1 - \bar{Y}_2, Y_{ij} - \bar{Y}_i) = 0, \quad \forall i \in \{1,2\}, 1 \le j \le n$$

We can obtain that

$$\begin{aligned}cov(\bar{Y}_1 - \bar{Y}_2, Y_{ij} - \bar{Y}_i) =& cov(\varepsilon_1'\frac{1}{n}\mathbf{1} - \varepsilon_2'\frac{1}{n}\mathbf{1}, \varepsilon_{1j} - \varepsilon_1'\frac{1}{n}\mathbf{1}) \\ =& cov(\frac{1}{n}\mathbf{1}'\varepsilon_i, \varepsilon_{1j} - \frac{1}{n}\varepsilon_i'\mathbf{1}) \\ =& \frac{1}{n}(1 + (n-1)\rho) - \frac{1}{n^2}(n(1-\rho) + n^2\rho) = 0\end{aligned}$$

thus we have the independence relation $\bar{Y}_1 - \bar{Y}_2 \perp\!\!\!\perp S_n^2$

(e)

Using the distribution of $\bar{Y}_1 - \bar{Y}_2$, we have

$$\frac{(\bar{Y}_1 - \bar{Y}_2)^2}{2\sigma^2(\frac{1-\rho}{n} + \rho)} \sim \chi_1^2$$

Then

$$\frac{\frac{n}{2(1-\rho)+2\rho n}(\bar{Y}_1-\bar{Y}_2)^2}{\frac{1}{1-\rho}S_n^2} \sim F_{1,2n-2}$$

(f)

For $n \to \infty$, notice that

$$\hat{F} = \frac{1-\rho+\rho n}{1-\rho}\hat{F}_\rho \to \infty$$

which causes frequent rejection of H_0.

Question 2.4

(a)

A direct result by left multiply $\Sigma^{-1/2}$:

$$\Sigma^{-1/2}y = \Sigma^{-1/2}X\beta + \Sigma^{-1/2}\varepsilon = \Sigma^{-1/2}X\beta + \xi$$

in which $\xi = \Sigma^{-1/2}\varepsilon \sim N(0, I)$.

(b)(c)

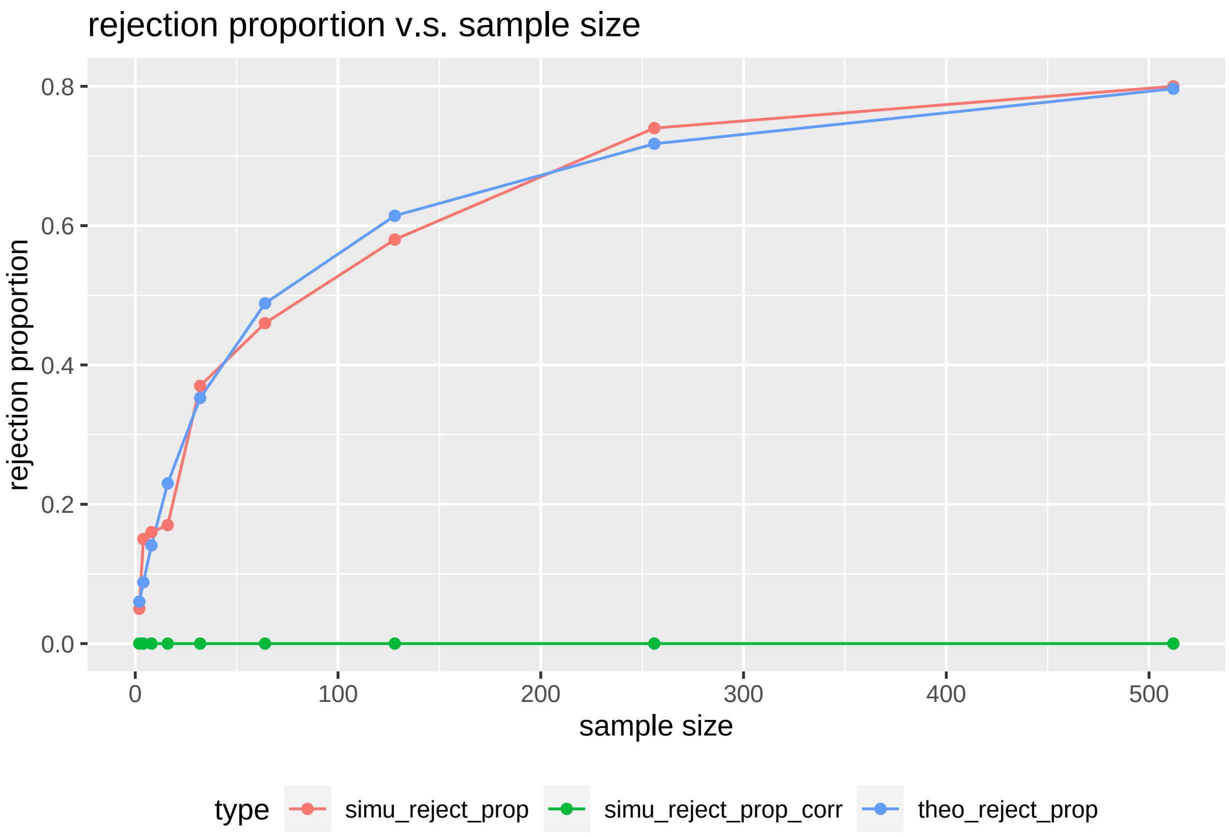
Simulation with correction to correlation matrix:

```r
library('mvtnorm')
library('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
set.seed(42)


N <- 1e2
nsamples <- 2^(1:9)
rho <- 0.1
```

```
f.stat.mat <- matrix(0, nrow = N, ncol = length(nsamples))
f.stat.mat.corr <- matrix(0, nrow = N, ncol = length(nsamples))

for(nsample in nsamples){
    for(n in 1:N){
        Sigma <- matrix(rho, nsample, nsample) + diag(1 - rho, nsample)
        X_1 <- rmvnorm(1, rep(0, nsample), Sigma) %>% t()
        X_2 <- rmvnorm(1, rep(0, nsample), Sigma) %>% t()
        f.stat <- (t.test(X_1, X_2, var.equal = TRUE)$statistic)^2
        hatsigma2 <- (apply(as.matrix(X_1), 1, function(x) (x-X_2)^2) %>% sum()) / (2*nsample^2)
        Sn2 <- (var(X_1)+var(X_2))/2
        hatrho <- max(1 - Sn2/hatsigma2, 0)
        f.stat.mat[n, which(nsamples == nsample)] <- f.stat
        f.stat.mat.corr[n, which(nsamples == nsample)] <- f.stat * (1 - hatrho) / (1 - hatrho + hatrh
    }
}

# calculate rejection proportion
reject_prop <- apply(f.stat.mat, 1, function(x)x > qf(0.95, 1, 2 * nsamples - 2)) %>% t() %>% apply(2
reject_prop.corr <- apply(f.stat.mat.corr, 1, function(x)x > qf(0.95, 1, 2 * nsamples - 2)) %>% t() %

# theoretical rejection proportion
theo_reject_prop <- 1 - pf(qf(0.95, 1, 2 * nsamples - 2) * (1 - rho) / (1 - rho + rho*nsamples), 1, 2

# plotting of rejection proportion v.s. sample size
plot_dat <- data.frame(nsamples = nsamples, simu_reject_prop = reject_prop, theo_reject_prop = theo_r
ggplot(plot_dat, aes(x = nsamples, y = rejection_prop, color = type)) + geom_line() + geom_point() +
```

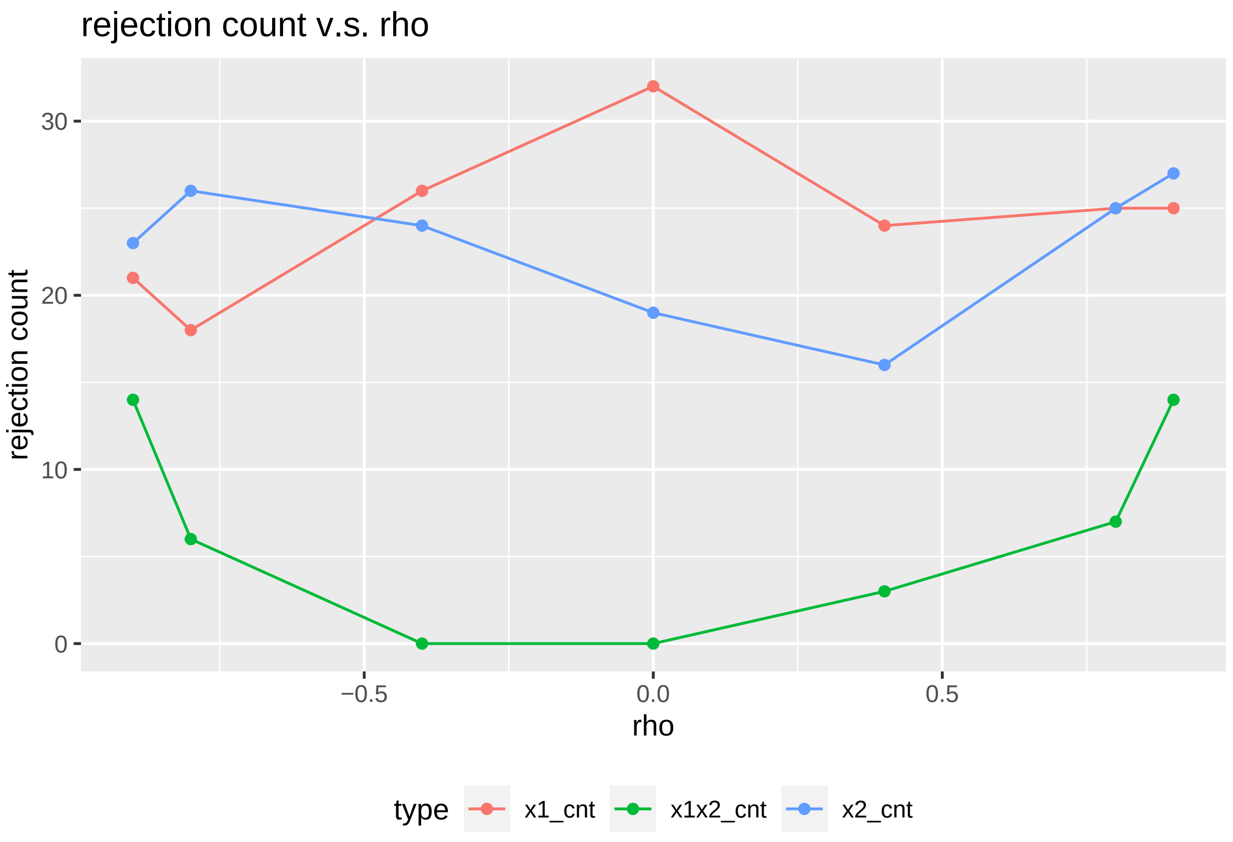


Question 2.5

The result is just as expected. The t.test on $x^{(1)}$ or $x^{(2)}$ individually performs 'homogeneously' across different ρ. However when we look at false discovery of $(x^{(1)}, x^{(2)})$ jointly we see that when $\rho \to \pm 1$, say, in our experiment $\rho = \pm.9, \pm.8$, we can see more false discovery of them together.

```
set.seed(42)
library('tidyverse')
library('mvtnorm')
aba <- read.csv('abalone.data', header = FALSE)
names(aba) <- c('Sex','Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight', 'Viscera weig
df <- data.frame(aba[,2:9])
df$isM <- ifelse(aba[,1] == 'M', 1, 0)
df$isF <- ifelse(aba[,1] == 'F', 1, 0)
# add extra multivariate noise
N <- 1e3
alpha = 0.05
rhos <- c(-0.9, -0.8, -0.4, 0, 0.4, 0.8, 0.9)
fd.cnt <- matrix(0, nrow = 3, ncol = length(rhos))
```

```r
for(rho in rhos){
    x1.cnt <- 0
    x2.cnt <- 0
    x1x2.cnt <- 0
    for(i in 1:N){
        x1x2 <- rmvnorm(nrow(df), c(0,0), matrix(c(1,rho,rho,1), 2, 2))
        df$x1 <- x1x2[,1]
        df$x2 <- x1x2[,2]
        lm.fit.rho <- lm(Rings ~ ., data = df)
        p.value <- summary(lm.fit.rho)$coefficients[c('x1','x2'),4]
        if(p.value[1] < alpha/2){
            x1.cnt <- x1.cnt + 1
        }
        if(p.value[2] < alpha/2){
            x2.cnt <- x2.cnt + 1
        }
        if(p.value[1] < alpha/2 & p.value[2] < alpha/2){
            x1x2.cnt <- x1x2.cnt + 1
        }
    }
    fd.cnt[,which(rhos == rho)] <- c(x1.cnt, x2.cnt, x1x2.cnt)
}

# plot fd.cnt v.s. rho
plot_dat <- data.frame(rho = rhos, x1_cnt = fd.cnt[1,], x2_cnt = fd.cnt[2,], x1x2_cnt = fd.cnt[3,]) %
ggplot(plot_dat, aes(x = rho, y = cnt, color = type)) + geom_line() + geom_point() + labs(x = "rho",
```



Question 2.6

(a)

A main problem of the figure is that it actually only test the $m = 5$ methods on $p = 1$ dataset (problem), and show the training process. Actually if we want to illustrate the 'generalization ability' of the methods, we should test the methods on multiple datasets.

(b)

First definitely runtime $R \geq 0$, so our normality assumption might not be that precise, and might cause confusion in understanding the quantative relations between α, β and runtime.

(c)

The transformation

$$Y_{ijk} = \phi(R_{ijk}) = \mu + \alpha_i + \beta_j + \varepsilon_{ijk}$$

should satisfy

$$\phi(2r) = 1 + \phi(r) \Rightarrow \phi(\,\cdot\,) = \log_2(\,\cdot\,)$$

(d)

The minimizer

$$\begin{aligned}(\hat{\mu}, \hat{\alpha}_{i=1}^m, \hat{\beta}_{j=1}^p) =& \underset{(\mu, \alpha_{i=1}^m, \beta_{j=1}^p)}{\arg\min} \sum_{i,j,k=1}^{m,p,n} \left(Y_{ijk} - \mu - \alpha_i - \beta_j\right)^2 \\ :=& \underset{(\mu, \alpha_{i=1}^m, \beta_{j=1}^p)}{\arg\min} \mathcal{L}(\mu, \alpha_{i=1}^m, \beta_{j=1}^p), \quad w.r.t. \sum_{i=1}^m \alpha_i = \sum_{j=1}^p \beta_j = 0\end{aligned}$$

satisties

$$\begin{cases} 0 = \frac{\partial \mathcal{L}}{\partial \mu} = -\sum_{i,j,k=1}^{m,p,n} 2(Y_{ijk} - \mu - \alpha_i - \beta_j) & \Rightarrow \hat{\mu} = \dfrac{\sum_{i,j,k=1}^{m,p,n} Y_{ijk}}{mnp} = \bar{Y}_{\cdots} \\ 0 = \frac{\partial \mathcal{L}}{\partial \alpha_i} = -\sum_{j,k=1}^{p,n} 2(Y_{ijk} - \mu - \alpha_i - \beta_j) & \Rightarrow \hat{\alpha}_i = \dfrac{\sum_{j,k=1}^{p,n} Y_{ijk}}{pn} - \bar{Y}_{\cdots} = \bar{Y}_{i\cdot\cdot} - \bar{Y}_{\cdots}, \quad \forall i \in [m] \\ 0 = \frac{\partial \mathcal{L}}{\partial \beta_j} = -\sum_{i,k=1}^{m,n} 2(Y_{ijk} - \mu - \alpha_i - \beta_j) & \Rightarrow \hat{\beta}_j = \dfrac{\sum_{i,k=1}^{m,n} Y_{ijk}}{mn} - \bar{Y}_{\cdots} = \bar{Y}_{\cdot j \cdot} - \bar{Y}_{\cdots}, \quad \forall j \in [p] \end{cases}$$

(e)

We have

$$\hat{\alpha}_1 - \hat{\alpha}_i = \bar{Y}_{1\cdot\cdot} - \bar{Y}_{i\cdot\cdot} \sim N(\alpha_1 - \alpha_i, \frac{2\sigma^2}{np}), \quad \forall i \in \{2, \cdots, m\}$$

and each $\hat{\alpha}_1 - \hat{\alpha}_i$ and $\hat{\alpha}_1 - \hat{\alpha}_j$ are independent if $i \neq j$. For each individual test

$$H_{0i} : \alpha_1 \geq \alpha_i$$

we can use the one-sided t-test

$$T_i = \frac{\hat{\alpha}_1 - \hat{\alpha}_i}{\sqrt{\dfrac{2\hat{\sigma}^2}{np}}} \sim t_{N-m-p+1}, \qquad \hat{\sigma}^2 = S^2 = \frac{1}{N-m-p+1} \sum_{i,j,k=1}^{m,p,n} (Y_{ijk} - \bar{Y}_{ij\cdot})^2$$

reject H_{0i} if $T_i \leq t_{N-m-p+1,-a}$.

Note: here I think there's a mistake in the HW material, in which $\hat{\sigma}^2$ should have degree of freedom $N-m-p+1$ if we are using model without interaction term.

(f)

If $H_{0i} : \alpha_1 \geq \alpha_i$ holds, then

$$\begin{aligned}\mathbb{P}\left(R_{1jk} \leq R_{ijk}\right) =& \mathbb{P}\left(Y_{1jk} \leq Y_{ijk}\right) \\ =& \mathbb{P}\left(\mu + \alpha_1 + \beta_j + \varepsilon_{ijk} \leq \mu + \alpha_i + \beta_j + \varepsilon_{ijk}\right) \\ =& \mathbb{P}\left(\alpha_1 + \varepsilon_{1jk} \leq \alpha_i + \varepsilon_{ijk}\right) \\ =& \mathbb{P}\left(2N(0, \sigma^2) \leq \alpha_i - \alpha_1\right) \leq \frac{1}{2}, \quad \forall i \in \{2, \cdots, m\},\ j \in [p],\ k \in [n]\end{aligned}$$

(g)

Consider for given $i \geq 2$, for each $j \in [p]$, we denote

$$B_{jk} = \mathbf{1}_{R_{1jk} \leq R_{ijk}} \sim \text{Bernoulli}(\rho_j), \quad k \in [n]$$

In this way null hypotheis is $H'_{0i} : \mathbb{P}\left(R_{1jk} \leq R_{ijk}\right) = \mathbb{E}\left[B_{jk}\right] = \rho_j \leq \frac{1}{2}, \forall j \in [p], k \in [n]$.

On the other hand, notice that for given t,

$$\mathbb{P}_{\rho_{j=1}^p}\left(\sum_{j=1}^p \sum_{k=1}^n B_{jk} \geq t\right)$$

is monotone increasing w.r.t. ρ_j $\forall j \in [p]$, and note that if null H'_{0i} holds we have

$$\sum_{j=1}^p \sum_{k=1}^n B_{jk} \sim \text{Binom}(\frac{1}{2}, np)$$

then we can obtain p-value as

$$\hat{p} = \mathbb{P}_\rho\left(\sum_{j=1}^p \sum_{k=1}^n B_{jk} \geq t\right) \leq \mathbb{P}_{\{1/2\}^p}\left(\sum_{j=1}^p \sum_{k=1}^n B \geq t\right) = \mathbb{P}\left(\text{Binom}(np, 1/2) \geq t\right) := \tilde{p}$$

since here $\hat{p} \leq \tilde{p}$, then we can examine threshold on $\tilde{p}? \leq a$, if so then we can definitely reject H'_{0i}.

(h)

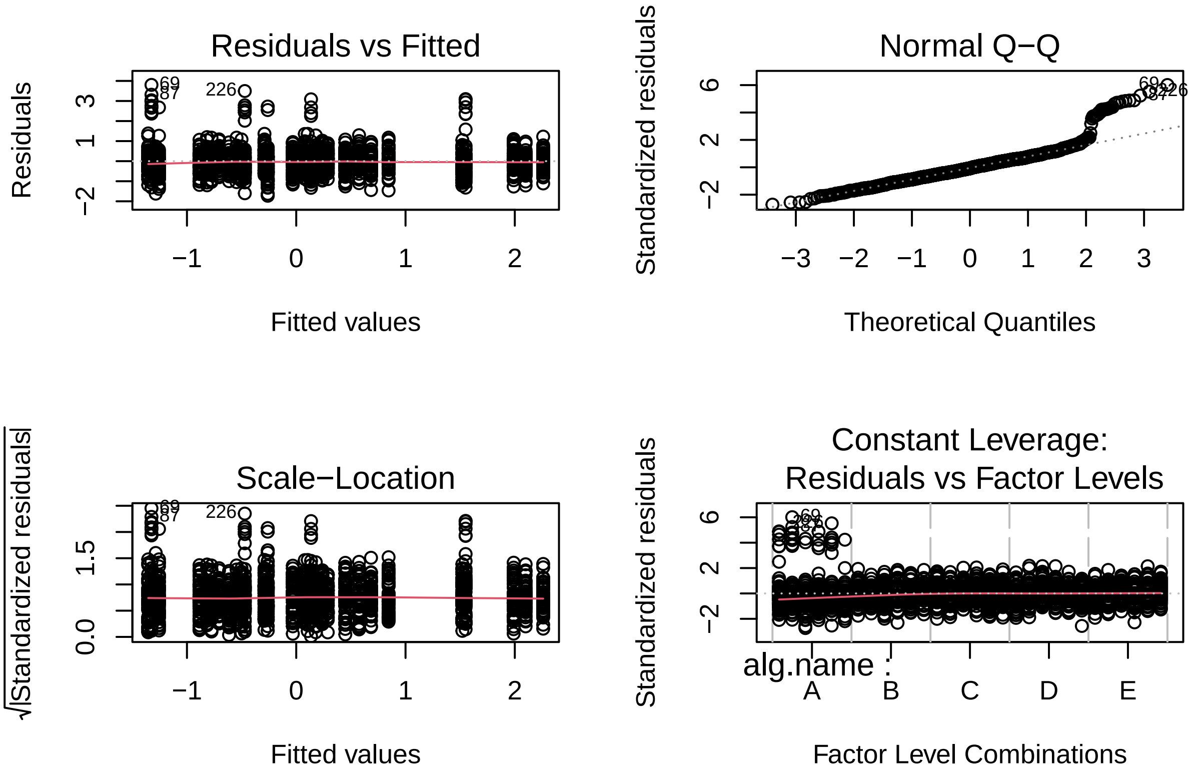
```
runtimes <- read.csv("runtimes.csv", sep = ",")

## data processing
runtimes$y <- log2(runtimes$runtime)
runtimes$alg.name <- as.factor(runtimes$alg.name)
runtimes$prob.ind <- as.factor(runtimes$prob.ind)
N <- nrow(runtimes)
m <- length(unique(runtimes$alg.name))
p <- length(unique(runtimes$prob.ind))
n <- N / (m * p)

## Simulation for $H_{0i}: \alpha _1 \leq \alpha _i, \forall i\geq 2$
aovfit <- aov(y ~ alg.name + prob.ind, data = runtimes)
S2 <- (aovfit$residuals)^2 %>% sum() / (N - m - p + 1)
p_values <- c()
for(alg in c('alg.nameB', 'alg.nameC', 'alg.nameD', 'alg.nameE')){
    alpha_diff <- -aovfit$coefficients[alg]
    t_stat <- alpha_diff / sqrt(2 * S2 / (n * p))
    p_values <- c(p_values, pt(t_stat, N - m - p + 1))
}
p_values_H <- p_values

## Simulation for $H_{0i}': \rho _j \leq 1/2, \forall j\in[p]$
p_values <- c()
for(alg in c('B', 'C', 'D', 'E')){
    ## first construct $B_{jk}$ matrix
    diff_1i <- runtimes$y[runtimes$alg.name == 'A'] - runtimes$y[runtimes$alg.name == alg]
    B_mat <- matrix(diff_1i<=0, nrow = n, ncol = p)
    ## then calculate p-value
    p_values <- c(p_values, pbinom(sum(B_mat), n * p, 1/2, lower.tail = FALSE))
}
p_values_H_prime <- p_values

print(data.frame(p_values_H, p_values_H_prime, row.names = c('B', 'C', 'D', 'E')))
```

```
##     p_values_H p_values_H_prime
## B 1.354876e-40     1.376150e-35
## C 2.682117e-17     2.051051e-22
## D 7.179311e-01     1.209099e-02
## E 1.228562e-25     4.414478e-30
```

Here we choose threshold $a = 0.05$. If we are using tests H_{0i}, we will accept H_{0D}, for which p-value is 0.72; while if we are using tests H'_{0i}, we will reject all H'_{0i}.

Observing from the following diagonisis figures, we see that actually the model does **not** fit the normality assumption quite well, which might cause the problem that the p-values are not that precise if we use H_{0i}. So I hereby choose to use tests H'_{0i} and believe that algorithm A is the best.

```
par(mfrow = c(2, 2))
plot(aovfit)
```


Residuals vs Fitted
Residuals
−2
1
3
69
87
226
−1
0
1
2
Fitted values
Normal Q−Q
Standardized residuals
−2
2
6
69
87
226
−3
−2
−1
0
1
2
3
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
0.0
1.5
69
87
226
−1
0
1
2
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
−2
2
6
69
alg.name :
A
B
C
D
E
Factor Level Combinations