Regression Analysis HW4

Tuorui Peng, tuoruipeng2028@u.northwestern.edu

(a)

- [i] $e = \log 1/p$: we have

$$\mathbb{E}\left[e\right] = \mathbb{E}\left[\log\frac{1}{p}\right] = \int_{t=0}^{\infty}\mathbb{P}\left(\log\frac{1}{p}\geq t\right)\,\mathrm{d}t = \int_{t=0}^{\infty}\mathbb{P}\left(p\leq e^{-t}\right)\,\mathrm{d}t \leq \int_{t=0}^{\infty} e^{-t}\,\mathrm{d}t = 1$$

- [ii] $e = 1/2\sqrt{p}$: note that in this case $e\in[1/2,\infty]$, we have

$$\mathbb{E}\left[e\right] = \int_{t=1/2}^{\infty}\mathbb{P}\left(\frac{1}{2\sqrt{p}}\geq t\right)\,\mathrm{d}t = \int_{t=1/2}^{\infty}\mathbb{P}\left(p\leq\frac{1}{4t^2}\right)\,\mathrm{d}t = \int_{t=1/2}^{\infty}\frac{1}{4t^2}\,\mathrm{d}t = \frac{1}{2} < 1$$

(b)

For OLS estimation, we know that

$$T_j = \frac{\hat{\beta}_j}{s_n\sqrt{[(X'X)^{-1}]_{jj}}} \sim t_{n-d} \Rightarrow T_j^2 \sim F_{1,n-d}$$

In this way we have

$$\begin{aligned}
\mathbb{E}\left[cM_j(m)\right] =& c\mathbb{E}\left[(T_j^2)^m\right]\\
=& c(n-d)^m\frac{\Gamma(\frac{1}{2}+m)\Gamma(\frac{n-d}{2}-m)}{\Gamma(\frac{1}{2}\Gamma(\frac{n-d}{2}))} \leq 1\\
\Rightarrow c \leq& (n-d)^{-m}\frac{\Gamma(\frac{1}{2}\Gamma(\frac{n-d}{2}))}{\Gamma(\frac{1}{2}+m)\Gamma(\frac{n-d}{2}-m)},\quad m\leq\frac{n-d}{2}
\end{aligned}$$

(c)

We have

$$\mathbb{P}\left(\text{falsely rejection}\right) = \mathbb{P}\left(E\geq\frac{1}{\alpha}\right) \overset{(i)}{\leq} \frac{\mathbb{E}\left[E\right]}{1/\alpha} \overset{(ii)}{\leq} \alpha$$

in which (i) uses Markov's inequality and (ii) uses the fact that $\mathbb{E}\left[E\right]\leq 1$.

(d)

- [(i)] Clearly we have

$$\text{card}(\mathcal{N} \cap \mathcal{R}) = \#\{j : j \in \mathcal{N} \& j \in \mathcal{R}\} = \sum_{j \in \mathcal{N}} \mathbf{1}(j \in \mathcal{R}) \Rightarrow \frac{\text{card}(\mathcal{N} \cap \mathcal{R})}{\max\{R, 1\}} \leq \sum_{j \in \mathcal{N}} \frac{\mathbf{1}(j \in \mathcal{R})}{\max\{R, 1\}}$$

- [(ii)] With our test being the e-value test, we have

$$\begin{aligned} \mathbf{1}(j \in \mathcal{R}) =& \mathbf{1}(E_j \geq \frac{N}{R\alpha}) \leq \mathbf{1}(E_j \geq \frac{N}{R\alpha}) \frac{R\alpha E_j}{N} \\ \Rightarrow \sum_{j \in \mathcal{N}} \frac{\mathbf{1}(j \in \mathcal{R})}{\max\{R, 1\}} \leq& \sum_{j \in \mathcal{N}} \frac{\mathbf{1}(E_j \geq \frac{N}{R\alpha})}{\max\{R, 1\}} \cdot \frac{R\alpha E_j}{N} \end{aligned}$$

- [(iii)] We have

$$\frac{\mathbf{1}(j \in \mathcal{R})R}{\max\{R, 1\}} \leq \frac{R}{\max\{R, 1\}} \leq 1 \Rightarrow \sum_{j \in \mathcal{N}} \frac{\mathbf{1}(j \in \mathcal{R})}{\max\{R, 1\}} \cdot \frac{R\alpha E_j}{N} \leq \sum_{j \in \mathcal{N}} 1 \cdot \frac{\alpha E_j}{N} = \frac{\alpha}{N} \sum_{j \in \mathcal{N}} E_j$$

(e)

Using the result in (d): FDP $= \text{card}(\mathcal{N} \cap \mathcal{R}) \leq \frac{\alpha}{N} \sum_{j \in \mathcal{N}} E_j$, we have

$$\text{FDR} = \mathbb{E}\left[\text{FDP}\right] = \mathbb{E}\left[\text{card}(\mathcal{N} \cap \mathcal{R})\right] \leq \mathbb{E}\left[\frac{\alpha}{N} \sum_{j \in \mathcal{N}} E_j\right] = \frac{\alpha}{N} \sum_{j \in \mathcal{N}} \mathbb{E}\left[E_j\right] \leq \frac{\text{card}\mathcal{N}}{N} \alpha$$

(f)

Simulation study on multiple regression model: from the figure below we can see that

- Bonferroni's method seems to keep making false discovery in this case, probably due to the noise effect in the data (note that our sd of signal is 0.1 while the sd of noise is 1);
- e-value method with too low m, say $m = 1$ seems to be too conservative to make any rejection, we can see that number of rejection for it is always 0;
- BY's method and e-value methods with mediate m value (e.g. $m = 4, 8$) are giving similar results, and they are both better than Bonferroni's method.
- And we see that with higher m value, we are having higher FDP. Intuitively, higher m value yields heavier tailed $M_j(m)$ distribution (concentrate at 0), which makes it more likely to make false discovery.

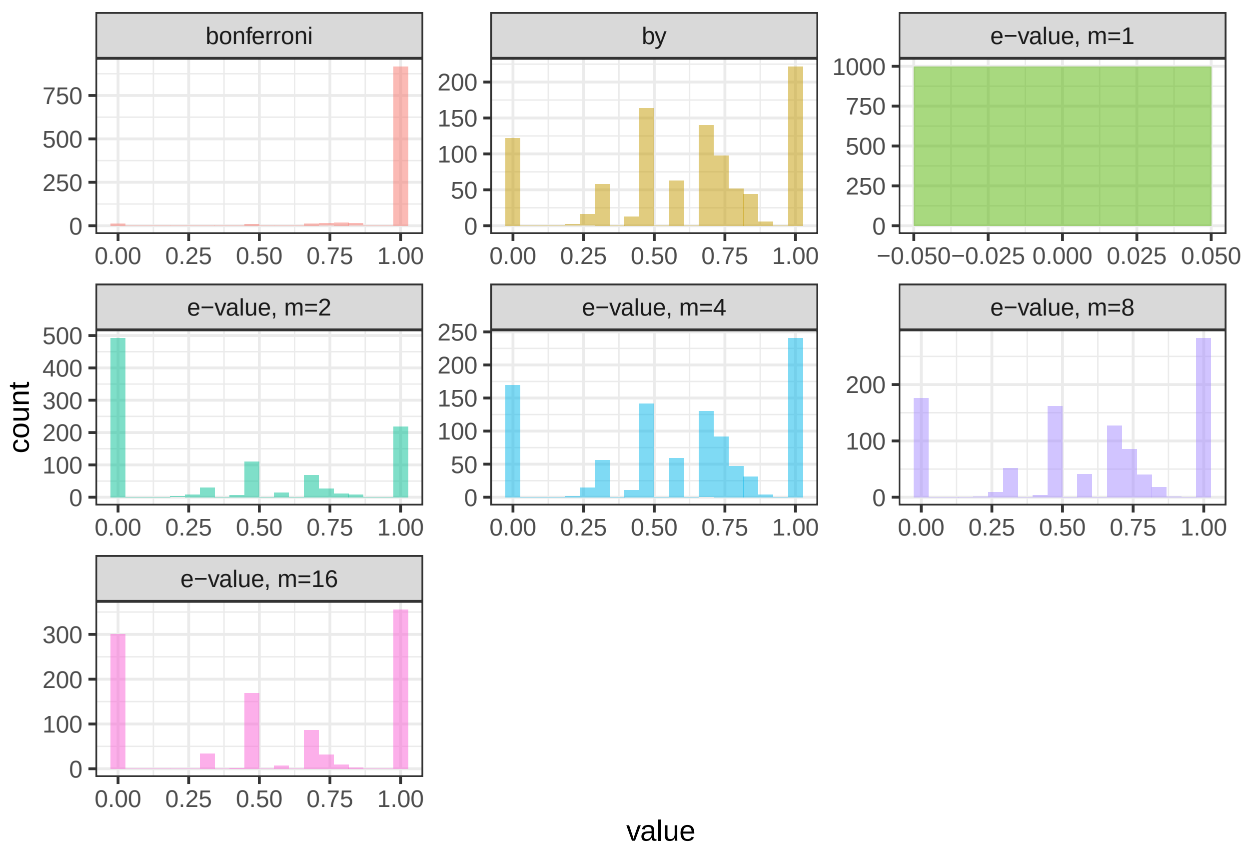
```
library(ggplot2)
library(reshape2)


N <- 1e3
```

```r
alpha <- 0.1
m_seq <- 2^(0:4)
fdp.mat <- matrix(0, nrow = N, ncol = 2 + length(m_seq))
num_rej.mat <- matrix(0, nrow = N, ncol = 2 + length(m_seq))
c_m <- function(m,n,d){
    exp( lgamma(1/2)+lgamma((n-d)/2)-lgamma(1/2+m)-lgamma((n-d)/2-m) )/((n-d)^m)
}
for(i in 1:N){
    set.seed(i)
    # construct data
    n <- 900
    d <- 30
    X <- matrix(rnorm(n*d), nrow = n, ncol = d)
    beta.true <- c(rnorm(10,0,0.1), rep(0,20))
    y <- X %*% beta.true + rnorm(n,0,1)
    lm.fit <- lm(y~0+X)
    p <- summary(lm.fit)$coefficients[,4]
    ## method (i): bonferroni correction
    R <- sum(p < alpha / d)
    fdp.bonf <- sum(p[1:10] < alpha / d) / max(R,1)

    fdp.mat[i,1] <- fdp.bonf
    num_rej.mat[i,1] <- R
    ## method (ii): benjamini-yekutieli: find largest (k) s.t. p_{(k)} \leq k\alpha / c_d*d and rejec
    c_d <- sum(1 / (1:d))
    order.p <- order(p)
    leq_idx_in_ord <- p[order.p] <= (1:d) * alpha / c_d / d
    k <- ifelse(sum(leq_idx_in_ord == TRUE) == 0, 0, max(which(leq_idx_in_ord)))
    R <- k
    null_idx <- order.p[1:10]
    fdp.by <- length(intersect( which(leq_idx_in_ord), null_idx)) / max(R,1)

    fdp.mat[i,2] <- fdp.by
    num_rej.mat[i,2] <- R
    ## method (iii): e-value
    t_j <- lm.fit$coefficients / summary(lm.fit)$coefficients[,2]
    for(m in m_seq){
        c.m <- c_m(m,n,d)
        cM_jm <- c.m * t_j^(2*m)
        E_j <- cM_jm
        order.e <- order(E_j, decreasing = TRUE)
```

```
        geq_idx_in_ord <- E_j[order.e] >= d / (1:d) / alpha
        k <- ifelse(sum(geq_idx_in_ord == TRUE) == 0, 0, max(which(geq_idx_in_ord)))
        R <- k
        null_idx <- order.e[1:10]
        fdp.e <- length(intersect( which(geq_idx_in_ord), null_idx)) / max(R,1)

        col_idx <- which(m_seq == m) + 2
        fdp.mat[i,col_idx] <- fdp.e
        num_rej.mat[i,col_idx] <- R
    }
}

## plot histogram of FDP, for each columns in fdp.mat
fdp.df <- data.frame(fdp.mat)
colnames(fdp.df) <- c("bonferroni", "by", paste0("e-value, m=", m_seq))

fdp.df.melt <- melt(fdp.df)
```

```
## No id variables; using all as measure variables
```

```
ggplot(fdp.df.melt, aes(x = value, fill = variable)) + geom_histogram(position = "identity", alpha =
```



```r
summary(data.frame(num_rej.mat))
```

```
##        X1              X2              X3          X4              X5       
##  Min.   :0.000   Min.   :0.000   Min.   :0   Min.   :0.000   Min.   :0.000  
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:0   1st Qu.:0.000   1st Qu.:2.000  
##  Median :3.000   Median :3.000   Median :0   Median :1.000   Median :3.000  
##  Mean   :3.462   Mean   :3.457   Mean   :0   Mean   :1.551   Mean   :3.122  
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:0   3rd Qu.:3.000   3rd Qu.:4.000  
##  Max.   :8.000   Max.   :9.000   Max.   :0   Max.   :7.000   Max.   :8.000  
##        X6              X7       
##  Min.   :0.000   Min.   :0.000  
##  1st Qu.:2.000   1st Qu.:1.000  
##  Median :3.000   Median :2.000  
##  Mean   :2.823   Mean   :1.864  
##  3rd Qu.:4.000   3rd Qu.:3.000  
##  Max.   :8.000   Max.   :6.000
```

```r
apply(fdp.mat, 2, mean)
```

```
## [1] 0.9685929 0.6208119 0.0000000 0.3801357 0.5968905 0.6070107 0.5486667
```