Independence tests using coin package in R

Jinheum Kim(Univ. Suwon), Jungdong Lee(Univ. Suwon, Master student)

2014.11.14

OUTLINE

Permutation test: review
Introduce 15 built-in functions in coin package
Explain how to use independent test function
Illustrative examples
Concluding remarks

Preliminary

- 검정통계량의 영가설 (null hypothesis) 분포는 모집단 분포에 의존하는데, 모집단의 분포를 모르면 결국 검정통계량의 분포도 알 수 없는데, 이때 모집단의 분포를 가정하여 영가설 분포를 직접 유도하거나 혹은 자료가 주어졌을 때 검정통계량의 조건부 분포를 영가설 분포로 대체하여 검정할 수 있음
- 전자의 방법을 무조건 검정이라고 하고, 후자의 방법을 조건부 검정 혹은 순열 검정 (permutation test)이라고 함 (Fisher, 1935)
- Strasser & Weber (1999)는 permutation test를 통합할 수 있는 이론적 근거를 마련함

Preliminary

- Hothorn et al. (2006, 2008)은 Strasser & Weber (1999)의 permutation test 이론을 coin 패키지로 구현
  - coin이란 이름은 conditional inference의 줄임말
  - 조건부 독립성 검정은 총괄적인 형태의 함수인 independence_test()를 통해서 할 수 있음
  - 잘 알려진 몇 가지 독립성 검정에 대해서는 사용자가 편리하도록 간편한 함수가 패키지에 포함되어 있음
- 본 논문에서는 이런 간편 함수에 대응하는 independece_test() 함수를 정의하고자 함
  - 관측변수의 적절한 변환과 관측값의 weight와 block값에 대한 정의가 필요

Permutation test: review

- Data: $\{(Y_i, X_i, w_i, b_i), i = 1, \ldots, n\}$,
  - X_i, Y_i: 표본공간 $\mathcal{X}, \mathcal{Y}$로부터 얻어진 i번째 관측값이고, data type은 numeric 또는 factor
  - w_i: i번째 관측값의 weight, default=1
  - b_i: i번째 관측값의 block 값, default=1
- j번째 block에 대한 영가설:

$$H_0: D(Y|X, j) = D(Y|j), j = 1, \ldots, k$$

Permutation test: review

- 영가설을 검정하기 위한 통계량:

$$T = \sum_{j=1}^{k} T_j \in R^{pq},$$

$$T_j = \mathrm{vec}\left(\sum_{i=1}^{n} I\left(b_i = j\right) w_i g(X_i) h(Y_i)' \right) \in R^{pq}, j = 1, \ldots, k$$

  - $g: \mathcal{X} \to R^p$: 관측값 X를 변환하는 함수
  - $h: \mathcal{Y} \to R^q$: 관측값 Y를 변환하는 함수
    - $h(Y_i) = h(Y_i, (Y_1, \ldots, Y_n))$: influence function이라고도 하는데, Y_i들의 관측값에는 의존하지만 Y_i 들의 배열에는 의존하지 않음

Permutation test: review

- $\mathcal{S}_j$: j번째 block에 속한 관측값들의 모든 순열들의 집합
- j번째 block에 대해, h의 조건부 평균 벡터와 공분산 행렬은,

$$E\left(h\middle|\mathcal{S}_j\right) = w_{\cdot j}^{-1}\sum_{i=1}^{n} I\left(b_i = j\right)w_i h(Y_i), j = 1, \dots, k,$$

$$Cov\left(h\middle|\mathcal{S}_j\right) =$$

$$w_{\cdot j}^{-1}\sum_{i=1}^{n} I\left(b_i = j\right)w_i\left(h(Y_i) - E\left(h|\mathcal{S}_j\right)\right)\left(h(Y_i) - E\left(h|\mathcal{S}_j\right)\right)',$$

$$j = 1, \dots, k$$

- $w_{\cdot j} = \sum_{i=1}^{n} I\left(b_i = j\right)w_i$: j번째 block에 속한 weight의 합

Permutation test: review

- j번째 block에 대해, T_j의 조건부 평균 벡터와 공분산 행렬은,

$$E(T_j|\mathcal{S}_j) = \mathrm{vec}\left(\left(\sum_{i=1}^{n} I\,(b_i = j) w_i g(X_i)\right) E(h|\mathcal{S}_j)'\right), j = 1, \ldots, k,$$

$$Cov(T_j|\mathcal{S}_j) = \frac{w_{\cdot j}}{w_{\cdot j} - 1} Cov(h|\mathcal{S}_j) \otimes \left(\sum_{i=1}^{n} I\,(b_i = j) w_i (g(X_i) \otimes g(X_i)')\right)$$

$$- \frac{1}{w_{\cdot j} - 1} Cov(h|\mathcal{S}_j) \otimes \left(\sum_{i=1}^{n} I\,(b_i = j) w_i g(X_i)\right) \otimes \left(\sum_{i=1}^{n} I\,(b_i = j) w_i g(X_i)\right)',$$

$$j = 1, \ldots, k$$

  - $\otimes$:Kroneker product

Permutation test: review

- 모든 순열들의 집합 $\mathcal{S}$가 주어졌을 때, 영가설 하에서, 통계량 T의 조건부 평균 벡터와 공분산 행렬은 k개 block의 결과를 합쳐 (Strasser & Weber, 1999),

$$\mu = E(T|\mathcal{S}) = \sum_{j=1}^{k} E\left(T_j|\mathcal{S}_j\right),$$

$$\Sigma = Cov(T|\mathcal{S}) = \sum_{j=1}^{k} C\,ov(T_j|\mathcal{S}_j)$$

Permutation test: review

- 통계량 $T \in R^{pq}$ 를 R로 보내는 단변량 통계량 중에서,
  - $pq = 1$이면, 검정통계량

    $c_{\text{scalar}}(T, \mu, \Sigma) = \text{diag}(\Sigma)^{-1/2}(T - \mu)$ 을 사용하고,
  - $pq > 1$이면, 검정통계량

    $c_{\max}(T, \mu, \Sigma) = \max \left| \text{diag}(\Sigma)^{-\frac{1}{2}}(T - \mu) \right|$ 또는

    $c_{\text{quad}}(T, \mu, \Sigma) = (T - \mu)' \Sigma^{+} (T - \mu)$ 을 사용

    - Σ^{+}: Σ의 Moore-Penrose inverse

Permutation test: review

- 영가설 하에서, 검정통계량 c의 조건부 분포는

$$\Pr(c(T, \mu, \Sigma) \leq z|\mathcal{S})$$

  - z를 초과하지 않는 순열의 개수를 전체 순열의 개수로 나눈 값으로 근사
  - z: 검정통계량의 c의 관측값

X:factor, Y: numeric 인 경우

		Test	p	q	Comments
Independent data	Location	W-M-W	1	1	#(B)=1, weight=1 #(G)=독립 모집단수
		Normal quantile	1	1	
		Median	1	1	
		Kruskal-Wallis	#(G)	1	
	Dispersion	Ansari-Bradley	1	1	
		Fligner-Killeen	#(G)	1	
Censored data		Log-rank	#(G)	1	Censoring 정보 필요 #(B)=1, weight=1 #(G)=독립 모집단수
Dependent data		Signed-rank	1	1	n =#(B)×#(G) [2×#(B)] weight=1 #(G)=독립 모집단수
		Friedman	#(G)	1	

X:numeric, Y: numeric 인 경우

Test	p	q	Comments
Spearman	1	1	#(B)=1, weight=1
Maximally selected statistic	#(G)-1	1	#(G)=서로 다른 X_i들의 개수 #(B)=1, weight=1

X:factor, Y: factor 인 경우

	Test	p	q	Comments
Independent pair data	Chi-square	#(R)	#(C)	#(R)=2차원 분할표의 행수 #(C)=2차원 분할표의 열수 n =#(R)×#(C) #(B)=1, weight=셀 빈도
	CMH	#(R)	#(C)	#(R)=2차원 부분분할표의 행수 #(C)=2차원 부분분할표의 열수 #(B)=제어변수의 수준수 n =#(B)×#(R)×#(C), weight=셀 빈도
	Linear-by-linear	1	1	#(B)=제어변수의 수준수 n =#(B)×#(R)×#(C), weight=셀 빈도
Matched-pair data	Marginal homogeneity	1	#(I)	#(B)=$I \times I$ 2차원 분할표의 총 셀 빈도 #(I)=$I \times I$ 2차원 분할표의 행수(열수) n =2×#(B), weight=1

X:factor, Y: numeric 인 경우

		Test	p	q	Comments
Independent data	Location	W-M-W	1	1	#(B)=1, weight=1 #(G)=독립 모집단수
		Normal quantile	1	1	
		Median	1	1	
		Kruskal-Wallis	#(G)	1	
	Dispersion	Ansari-Bradley	1	1	
		Fligner-Killeen	#(G)	1	
Censored data		Log-rank	#(G)	1	Censoring 정보 필요 #(B)=1, weight=1 #(G)=독립 모집단수
Dependent data		Signed-rank	1	1	n =#(B)×#(G) [2×#(B)] weight=1 #(G)=독립 모집단수
		Friedman	#(G)	1	

surv_test() 함수: censored data

- 두 개 이상의 모집단 ($p \geq 2$) 의 생존함수가 서로 동일한지를 검정하기 위한 log-rank test (Kalbfleisch & Prentice, 2002)
- Data: $\{(X_i, Y_i, \delta_i, w_i = 1, b_i = 1), i = 1, \ldots, n\}$
  - X_i: i번째 개체가 속한 그룹 ($X_i = 1, \ldots, p$)
  - Y_i: i번째 개체의 생존시간
  - δ_i: i번째 개체의 우중도절단여부를 나타내는 값 ($\delta_i = 0,1$)

surv_test() 함수: censored data

- $Y_{(1)} < \cdots < Y_{(c)}$: 관측된 서로 다른 생존시간
- $Y_{l1}, \ldots, Y_{lm_l}$: 구간 $[Y_{(l)}, Y_{(l+1)})$ 에서 중도절단된 시간, $l = 0, \ldots, c$
  - $Y_{(0)} = 0, Y_{(c+1)} = \infty$
- $s_l = 1 - \sum_{h=1}^{l} \frac{d_h}{n_h}$: $Y_{(l)}$에서 이벤트가 발생한 개체들의 스코어, $l = 1, \ldots, c$
- $S_l = -\sum_{h=1}^{l} \frac{d_h}{n_h}$: 구간 $[Y_{(l)}, Y_{(l+1)})$에서 중도절단된 개체들의 스코어, $l = 1, \ldots, c$
  - $S_0 = 0$

surv_test () 함수: censored data

- Transformations
  - $g(X_i) = \left(I(X_i = 1), \ldots, I(X_i = p)\right)', i = 1 \ldots, n$
  - $h(Y_i) =$
    $\sum_{l=1}^{c} \left\{ s_l I(Y_i = Y_l, \delta_i = 1) + S_l \sum_{j=1}^{m_l} I(Y_i = Y_{lj}, \delta_i = 0) \right\},$
    $i = 1, \ldots, n$
- R-code
  - > independence_test(Surv(time, event) ~ stadium, data = ocarcinoma, ytrafo = function(data) trafo(data, numeric_trafo = logrank_trafo))

X:factor, Y: numeric 인 경우

		Test	p	q	Comments
Independent data	Location	W-M-W	1	1	#(B)=1, weight=1 #(G)=독립 모집단수
		Normal quantile	1	1	
		Median	1	1	
		Kruskal-Wallis	#(G)	1	
	Dispersion	Ansari-Bradley	1	1	
		Fligner-Killeen	#(G)	1	
Censored data		Log-rank	#(G)	1	Censoring 정보 필요 #(B)=1, weight=1 #(G)=독립 모집단수
Dependent data		Signed-rank	1	1	n =#(B)×#(G) [2×#(B)] weight=1 #(G)=독립 모집단수
		Friedman	#(G)	1	

wilcoxsign_test () 함수: dependent data

- 서로 다른 k개의 block 내에 있는 두 모집단의 모평균이 서로 동일한지를 검정 (Wilcoxon, 1945)
- Data: $\{(X_i, Y_i, w_i = 1, b_i), i = 1, \ldots, n(= 2k)\}$
  - X_i: i번째 개체가 속한 그룹 ($X_i = 1,2$)
  - Y_i: i번째 개체의 관측 값
  - b_i: i번째 개체가 속한 block 값 ($b_i = 1, \ldots, k$)

wilcoxsign_test () 함수: dependent data

- Transformations
  - $g(X_i) = I(X_i = 2), i = 1 \ldots, n$
  - $h(Y_i) = R_i \sum_{l \neq i}^{n} I(b_l = b_i) I(Y_i < Y_l), i = 1, \ldots, n$
    - $D_j = \sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^{n} I\left(b_{i_1} = j = b_{i_2}\right) |Y_{i_1} - Y_{i_2}|$: j번째 block 내에 있는 두 관측 값의 절대 편차, $j = 1, \ldots, k$
    - $R_i = \sum_{j_1=1}^{k} \left\{ I(b_i = j_1) \sum_{j_2=1}^{k} I(D_{j_2} \leq D_{j_1}) \right\}, i = 1, \ldots, n$

wilcoxsign_test () 함수: dependent data

- R-code
  - > x = c(1.83, 0.50, 1.62, 2.48, 1.68, 1.88, 1.55, 3.06, 1.30)
  - > y = c(0.878, 0.647, 0.598, 2.05, 1.06, 1.29, 1.06, 3.14, 1.29)
  - > xydat = data.frame(y = c(y, x), x = gl(2, length(x)), block = factor(rep(1:length(x), 2)))
  - > a = as.numeric(rep(x > y, rep(2, length(x))))
  - > b = rep(c(0, 1), length(x))
  - > arank = as.numeric(a == b) * rep(rank(abs(x - y)), rep(2, length(x)))
  - > d = data.frame(d.x = rep(0:1, length(x)), d.y=c(x, y), block = factor(rep(1 : length(x), rep(2, length(x)))))
  - > independence_test(d.y ~ d.x | block, data = d, ytrafo = function(data) numeric_trafo = arank)

X:numeric, Y: numeric 인 경우

Test	p	q	Comments
Spearman	1	1	#(B)=1, weight=1
Maximally selected statistic	#(G)-1	1	#(G)=서로 다른 X_i들의 개수 #(B)=1, weight=1

max_stat() 함수: maximally selected statistic

- X의 가능한 모든 값을 기준으로 두 그룹으로 나눈 후, 통계량의 최대값으로 두 그룹의 모평균이 서로 동일한지를 검정하기 위한 최대 선택 통계량 검정 (Müller & Hothorn, 2004)
- Data: $\{(X_i, Y_i, w_i = 1, b_i = 1), i = 1, \ldots, n\}$
  - X_i, Y_i: i번째 개체의 관측값
- $X_{(1)} < \cdots < X_{(p)} < X_{(p+1)}$: 서로 다른 X의 값

max_stat() 함수: maximally selected statistic

- Transformations
  - $g(X_i) = \left(I\left(X_i \leq X_{(1)}\right), \ldots, I\left(X_i \leq X_{(p)}\right)\right)', i = 1 \ldots, n$
  - $h(Y_i) = Y_i, i = 1, \ldots, n$
- R-code
  - > independence_test(counts ~ coverstorey, data = treepipit, xtrafo = function(data) trafo(data, numeric_trafo = maxstat_trafo), teststat = "max")

X:factor, Y: factor 인 경우

	Test	p	q	Comments
Independent pair data	Chi-square	#(R)	#(C)	#(R)=2차원 분할표의 행수 #(C)=2차원 분할표의 열수 n =#(R)×#(C) #(B)=1, weight=셀 빈도
	CMH	#(R)	#(C)	#(R)=2차원 부분분할표의 행수 #(C)=2차원 부분분할표의 열수 #(B)=제어변수의 수준수 n =#(B)×#(R)×#(C), weight=셀 빈도
	Linear-by-linear	1	1	#(B)=제어변수의 수준수 n =#(B)×#(R)×#(C), weight=셀 빈도
Matched-pair data	Marginal homogeneity	1	#(I)	#(B)=$I \times I$ 2차원 분할표의 총 셀 빈도 #(I)=$I \times I$ 2차원 분할표의 행수(열수) n =2×#(B), weight=1

cmh_test() 함수: independent pair data

- 3 차원 분할표에서 두 범주형 변수의 조건부 독립성을 검정하기 위한 Cochran-Mantel-Haenszel 검정 (Cochran, 1954; Mantel & Haenszel, 1959)
- X, Y의 범주수가 각각 p, q개이고, 제어 변수 (block 변수)의 범주수는 k개라고 가정
- Data: $\{(X_i, Y_i, w_i, b_i), i = 1, \ldots, n(= p \times q \times k)\}$
  - X_i, Y_i: i번째 X, Y 범주쌍의 값($X_i = 1, \ldots, p; Y_i = 1, \ldots, q$)
  - w_i: i번째 X, Y 범주쌍의 관측 개체수
  - b_i:i번째X, Y 범주쌍의 속한 block 값($b_i = 1, \ldots, k$)

cmh_test() 함수: independent pair data

- Transformatrion
  - $g(X_i) = \left(I(X_i = 1), \ldots, I(X_i = p)\right)', i = 1, \ldots, n$
  - $h(Y_i) = \left(I(Y_i = 1), \ldots, I(Y_i = q)\right)', i = 1, \ldots, n$
- R-code
  - > independence_test(Job.Satisfaction ~ Income | Gender, data = jobsatisfaction, weights = ~ as.vector(jobsatisfaction), teststat = "quad")

X:factor, Y: factor 인 경우

	Test	p	q	Comments
Independent pair data	Chi-square	#(R)	#(C)	#(R)=2차원 분할표에서 행수 #(C)=2차원 분할표에서 열수 n =#(R)×#(C) #(B)=1, weight=셀빈도
	CMH	#(R)	#(C)	#(R)=2차원 부분분할표에서 행수 #(C)=2차원 부분분할표에서 열수 #(B)=제어변수의 수준수 n =#(B)×#(R)×#(C), weight=셀빈도
	Linear-by-linear	1	1	#(B)=제어변수의 수준수 n =#(B)×#(R)×#(C), weight=셀빈도
Matched-pair data	Marginal homogeneity	1	#(I)	#(B)=$I \times I$ 2차원 분할표의 총 셀빈도 #(I)=$I \times I$ 2차원 분할표 행수(열수) n =2×#(B), weight=1

mh_test () 함수: matched-pair data

- 대응비교처럼 한 쌍 (예, 대조군 대 시험군)으로부터 얻어진 관측값을 행과 열의 범주값으로 하는 $q \times q$ 2차원 분할표에서,
  - Original data: $\{(X_{i'}^*, Y_{i'}^*, w_{i'}, b_{i'} = 1), i' = 1, \ldots, n'(= q^2)\}$
    - $X_{i'}^*, Y_{i'}^*$: i'번째 X^*, Y^* 범주쌍의 값$(X_{i'}^*, Y_{i'}^* = 1, \ldots, q)$
    - $w_{i'}$: i'번째 X^*, Y^* 범주쌍의 관측 개체수

mh_test () 함수: matched-pair data

- 두 범주형 변수의 주변합 독립성 (혹은 주변 동질성)을 검정을 위해 $q \times q$ 2차원 분할표를 $k \times 2$ 2차원 분할표로 변형 (Stuart, 1955; Maxwell, 1970)
  - 변형된 분할표에서 행은 block을 나타내고 ($k = \sum_{i'=1}^{n'} w_{i'}$), 열은 서로 다른 두 처리를 나타냄
  - Transformed data: $\{(X_i, Y_i, w_i = 1, b_i), i = 1, \dots, n\ (= 2k)\}$
    - X_i: i번째 개체가 속한 그룹 ($X_{2j-1} = 1, X_{2j} = 2, j = 1, \dots, k$)
    - Y_i: i번째 개체의 관측값 ($Y_i = 1, \dots, q$)
    - b_i: i번째 개체가 속한 block 값 ($b_{2j-1} = j = b_{2j}, j = 1, \dots, k$)

mh_test () 함수: matched-pair data

- Transformation
  - $g(X_i) = I(X_i = 1), i = 1 \ldots, n$
  - $h(Y_i) = \left(I(Y_i = 1), \ldots, I(Y_i = q)\right)', i = 1, \ldots, n$

mh_test () 함수: matched-pair data

- R-code
  - > opinions = c("always wrong", "almost always wrong","wrong only sometimes", "not wrong at all")
  - > PreExSex = as.table(matrix(c(144, 33, 84, 126, 2, 4, 14, 29,0, 2, 6, 25, 0, 0, 1, 5), nrow = 4, dimnames = list(PremaritalSex = opinions, ExtramaritalSex = opinions)))
  - > cw = rep(names(margin.table(PreExSex, 2)), as.vector(margin.table(PreExSex, 2)))
  - > rw = rep(rep(rownames(PreExSex), times = dim(PreExSex)[2]), as.vector(PreExSex))
  - > y = factor(c(rw, cw), levels = rownames(PreExSex))
  - > x = c(rep(1, sum(PreExSex)), rep(0, sum(PreExSex)))
  - > block = factor(rep(1:sum(PreExSex), 2))
  - > mh.PreExSex = data.frame(x = x, y = y, block = block)
  - > independence_test(y ~ x | block, data = mh.PreExSex, teststat = "quad")

Illustrative examples

Test	Dataset	No. obs	X/Y
Log-rank	ocacinoma	35	f/n
Wilcoxon signed-rank	xydat	18	
Maximally selected statistic	treepipit	86	n/n→f/n
CMH	jobsatisfaction	104	f/f
Marginal homogeneity	preexsex	475	

- P값 비교
  - 점근분포에 의한 P값
  - Permutation test에 기초한 P값
    - 1,000번
    - 10,000번
    - 100,000번
  - Exact test에 의한 P값

Illustrative examples

Table 4.1 P-values of fifteen independence tests based on the asymptotic distribution, the number of permutations(1,000, 10,000, or 100,000), and exact distribution

Test	Data set	No. obs	Asymptotic-based	Permutation-based 1,000	Permutation-based 10,000	Permutation-based 100,000	Exact-based
			In case that X is `factor` and Y is `numeric`,				
Wilcoxon-Mann-Whitney	water_transfer	15	0.2207	0.2370	0.2555	0.2540	0.2544
Normal quantiles	water_transfer	15	0.2564	0.2750	0.2707	0.2662	0.2691
Median	water_transfer	15	0.1573	0.2820	0.2788	0.2828	0.2821
Ansari-Bradley	sid	40	0.1815	0.1890	0.1901	0.1879	0.1881
Kruskal-Wallis	YOY	40	4.34e-05	0.0000	0.0000	0.0000	-
Fligner-Killeen	sid	40	0.2237	0.2270	0.2267	0.2258	-
Log rank	ocarcinoma	35	0.0194	0.0120	0.0164	0.0188	0.0182
Wilcoxon signed rank	xydat	18	0.0382	0.0350	0.0424	0.0403	-
Friedman	RoundingTimes	66	0.0038	0.0060	0.0029	0.0032	-
			In case that both X and Y are `numeric`,				
Spearman	USJudgeRatings	43	0.2527	0.2710	0.2637	0.2585	-
Maximally selected statistic	treepipit	86	0.0001	0.0010	0.0007	0.0005	-
			In case both X and Y are `factor`,				
Pearson's chi-square	jobsatisfaction (female only)	64	0.6669	0.6840	0.6893	0.6880	-
Cochran-Mantel-Haenszel	jobsatisfaction	104	0.3345	0.3400	0.3337	0.3300	-
Linear-by-linear association	jobsatisfaction	104	0.0101	0.0140	0.0120	0.0109	-
Maxwell-Stuart	PreExSex	475	0.0000	0.0000	0.0000	0.0000	-

Concluding remarks

- coin 패키지에 내장된 독립성 검정을 위한 간편 함수를 independence_test() 함수로 표현
- 두 변수 X, Y를 적절히 변환하였으며, 관측값의 weight와 block 값에 대해 정의
- 정의한 independence_test() 함수를 써서 실제 자료의 점근 분포와 permutation test, exact test에 기초한 P값을 구하고 그 결과를 서로 비교
- Permutation test 방법은 검정통계량의 분포를 모를 때 유용한데, 독립성 검정에 대한 실제 자료 분석에서 살펴본 것처럼 permutation의 횟수가 증가함에 따라 exact test의 결과에 가까워질 뿐만 아니라 자료의 크기가 크면 점근 분포에 의한 결과와도 유사함을 알 수 있었음
- 본 논문에서 살펴본 15 개의 독립성 검정 이외 다른 독립성 검정 문제에 대해서도, 본 논문에서 한 것처럼,
  - 두 변수에 대해 적절한 변환을 정의하고,
  - weight와 block 값을 정의한 후에, permutation test 방법에 의해 영가설 분포를 모르는 검정통계량의 P값을 구할 수 있을 것으로 기대됨

THANK YOU!!!