

מודלים סטטיסטיים ויישומיהם 52518 תשע"ח – פתרון תרגיל 5

1. להלן נתונים שלוש קבוצות:

	2.9	3	2.6	3.2	קבוצה 1
	3.8	2.7	4	2.4	קבוצה 2
	3.8	3.4	3.7	2.2	קבוצה 3

- א. חשבו ידנית את סטטיסטי Kruskal-Wallis.
- ב. חשבו את מובוקות הסטטיסטי בהתבסס על קירוב אסימפטוטי להתפלגות חי-בריבוע.
- ג. כתבו פונקציה ב-R אשר מקבלת אוסף נתונים ומחשבת עבורו את מובוקות מבחן Kruskal-Wallis בהתבסס על סימולציות מונטה קרלו. הריצו את הפונקציה על הנתונים לעיל.
- ד. השתמשו בפונקציה `KruskalWallis.test` מתוך החבילת KSample ב-R כדי לבצע את מבחן Kruskal-Wallis, עם שלושת האופציות הקיימות עבור הparameter `method` (הריצו 3 פעמים).
- ה. השוו בין תוצאות סעיפים ב', ג' ו-ד'.

$$n_1 = n_2 = 4, n_3 = 5, N = \sum_i n_i = 13, I = 3$$

א. דרגות נתונים (סטטיסטי סדר):

1	2	3	4	5	6	7	8	9	10	11.5	11.5	13
2	2.2	2.4	2.6	2.7	2.9	3	3.2	3.4	3.7	3.8	3.8	4
y₃₁	y₃₂	y₂₁	y₁₂	y₂₃	y₁₄	y₁₃	y₁₁	y₃₄	y₃₃	y₂₄	y₃₅	y₂₃

$$\bar{R}_{i.} = \begin{bmatrix} 6.25 \\ 8.125 \\ 6.7 \end{bmatrix}, \bar{R}_{..} = 7, n_i(\bar{R}_{i.} - \bar{R}_{..})^2 = \begin{bmatrix} 4 \cdot 0.5625 \\ 4 \cdot 1.266 \\ 5 \cdot 0.09 \end{bmatrix} = \begin{bmatrix} 2.25 \\ 5.0625 \\ 0.45 \end{bmatrix} \rightarrow \sum_{i=1}^I n_i(\bar{R}_{i.} - \bar{R}_{..})^2 = 7.7625$$

y_{ij}	R_{ij}	$(R_{ij} - \bar{R}_{i.})^2$
3.2	8	3.0625
2.6	4	5.0625
3	7	0.5625
2.9	6	0.0625
2.4	3	26.2656
4	13	23.7656
2.7	5	9.7656
3.8	11.3	11.3906
2	1	32.49
2.2	2	22.09
3.7	10	10.89
3.4	9	5.29
3.8	11.5	23.04

$$\sum_{j=1}^{n_i} (R_{ij} - \bar{R}_{i.})^2 = \begin{bmatrix} 8.75 \\ 71.1875 \\ 93.8 \end{bmatrix} \rightarrow \sum_{i=1}^I \sum_{j=1}^{n_i} (R_{ij} - \bar{R}_{i.})^2 = 173.7375$$

$$\rightarrow H = (N - 1) \frac{\sum_{i=1}^I n_i (\bar{R}_{i.} - \bar{R}_{..})^2}{\sum_i \sum_j (R_{ij} - \bar{R}_{i.})^2} = 12 \cdot \frac{7.7625}{173.7375} \approx 0.5362$$

$$P(\chi^2_2 > H) = \underbrace{pchisq(H, 2, lower.tail = F)}_{@R} = 0.764849 .$$

. R קוד :

```
library("multicool")
KW <- function(y,I){ #y are samples, I is factor vector
  R <- rank(y)
  return((length(y) - 1) * (sum((ave(R, I) - mean(R)) ^ 2)) / (sum((R - ave(R, I)) ^ 2)))
}
```

```
MCKW <- function(y,I){
  H <- KW(y,I)
  facts <- allPerm(initMC(I))
  perms <- nrow(facts) #number of possible permutations
  pH <- rep(0,perms)
  for(i in 1:perms){
    pH[i] <- KW(y, as.factor(facts[i,]))
  }
  return(mean(0 + (pH > H)))
}
```

y <- c(3.2,2.6,3,2.9,2.4,4,2.7,3.8,2,2.2,3.7,3.4,3.8)

I <- as.factor(c(rep(1,4),rep(2,4),rep(3,5)))

MCKW(y,I)

הערך המתקיים הוא 0.791564

. T

*qn.test(y ~ I, method = "asymptotic") → H = 0.5132, Pval = 0.7737
 qn.test(y ~ I, method = "simulated") → H = 0.5132, Pval = 0.8001
 qn.test(y ~ I, method = "exact", Nsim = 1e9) → H = 0.5132, Pval = 0.7951*

ה. התוצאות שהתקבלו לסעיפים השונים זהות כמעט לחלוטין

2. בהתייחס לנחוני הקובץ ex4a.txt:

- א. כתבו פונקציה ב-R לביצוע מבחן Levene והריצו אותה על הנתוני הקובץ
- ב. כתבו פונקציה ב-R לביצוע מבחן Welch והריצו אותה על הנתוני הקובץ.

. R

```
levene <- function(y,G){
  m <- ave(y,G)
  x <- abs(y - m)
  N <- length(y)
  I <- length(levels(G))
  numer <- sum((ave(x, G) - mean(x)) ^ 2)
  denom <- sum((x - ave(x, G)) ^ 2)
  Fstat <- ((N - I) * (numer)) / ((I - 1) * (denom))
```

```

    return(pf(Fstat, (I - 1), (N - I), lower.tail = F))
}
D <- read.table('ex4a.txt')[2:3]
G <- as.factor(D[,1])
y <- D[,2]
levene(y, G)

```

הערך המתתקבל הוא 8.43×10^{-5} ולכן, בהינתן אלטרנטיבת בעצמה מספקת, נדחה את H_0 כמעט בכל רמת מובהקות.

.ב.

```

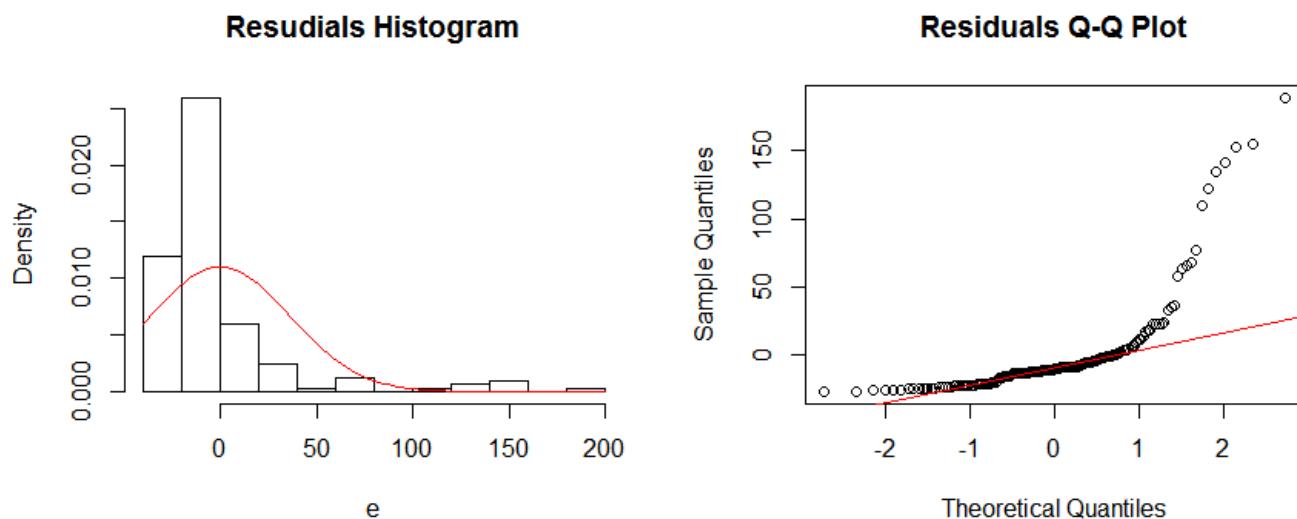
welch <- function(y, G) {
  sigma <- tapply(y, G, FUN=sd)
  n <- tapply(y, G, FUN=length)
  w <- n / (sigma) ^ 2
  W <- sum(w)
  yi <- tapply(y, G, FUN=mean)
  yw <- sum(w * yi) / W
  SSBW <- sum(w * (yi - yw) ^ 2)
  pii <- w / W
  I <- length(levels(G))
  P <- sum(((1 - pii) ^ 2) / (n - 1))
  C <- 1 + ((2 * (I + 2)) / (I ^ 2 - 1)) * P
  d <- 1 / ((3 / (I ^ 2 - 1)) * P)
  Fstat <- SSBW / ((I - 1) * C)
  return(pf(Fstat, (I - 1), d, lower.tail = F))
}
D <- read.table('ex4a.txt')[2:3]
G <- as.factor(D[,1])
y <- D[,2]
welch(y, G)

```

הערך המתתקבל הוא 2.446×10^{-4} ולכן, בהינתן אלטרנטיבת בעצמה מספקת, נדחה את H_0 כמעט בכל רמת מובהקות.

3. עבור הנתוני הקובץ **ex4b.txt**, השתמשו בכלי שנלמדו בכיתה כדי לבדוק את הנחות המודל של ANOVA חד-בוננית. נסzo לשפר את התאמת הנתונים להנחה באמצעות טרנספורמציה.

נגיד $e_{ij} = \mu_i + \text{מוצעי הקבוצות} + \text{שאריות} \quad \mu_i - y_{ij} = e_{ij}$. בעת נביט ב-**QQPlot** ובהיסטוגרמה של השאריות:

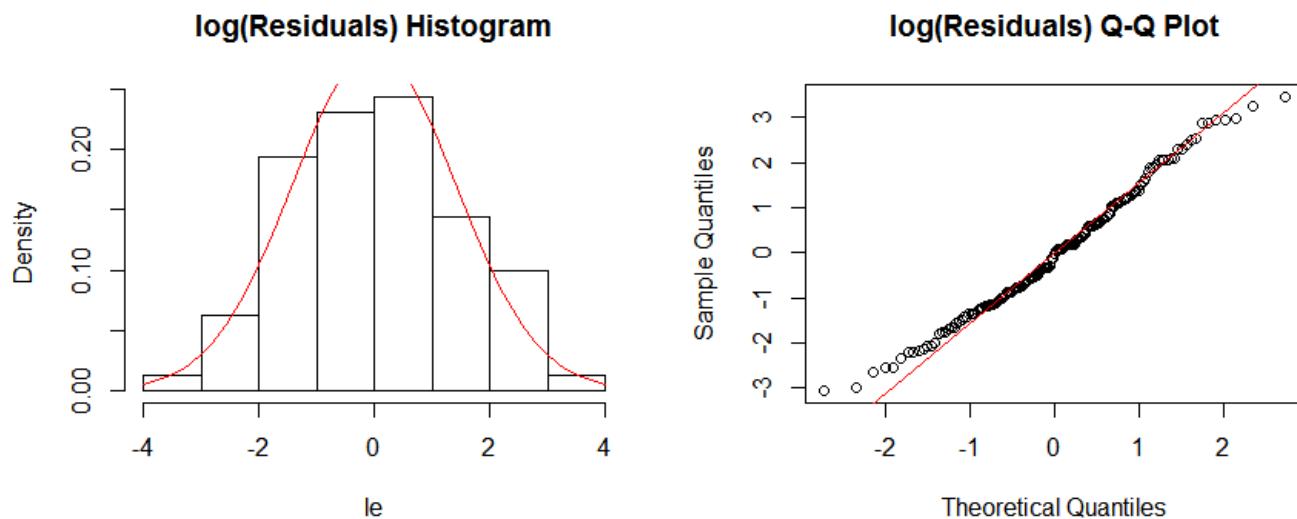


ניתן להתרשם כי זו התרגלות עם זנב ימני ארוך. גם לפי מבחן קולמוגורוב-סmirnov, נדחה את השערת האפס לפיה השאריות מפולגות נורמלית:

Lilliefors (kolmogorov-smirnov) normality test

```
data: e
D = 0.26734, p-value < 2.2e-16
```

כיוון שזו התרגלות עם זנב ימני ארוך, נרצה להפעיל טרנספורמציה לוגריתמית, $\tilde{y}_{ij} = \log(y_{ij})$.



וניתן לראות נורמליות, גם לפי קולמוגורוב-סmirnov:

Lilliefors (kolmogorov-smirnov) normality test

```
data: le
D = 0.064132, p-value = 0.1109
```

קוד R לשאלת זו:

```
library(nortest)
D <- read.table("ex4b.txt")
mu <- ave(D$V2,D$V1)
e <- D$V2 - mu
```

```

#qq plot
qqnorm(e, main = "Residuals Q-Q Plot")
qqline(e, col = 2)
#histogram
hist(e, probability = T, main = "Residuals Histogram")
curve(dnorm(x, mean(e), sd(e)), add = T, col = 2)
#kolmogorov-smirnoff
lillie.test(e)

#log transformation
ly <- log(D$V2)
mu.log <- ave(ly,D$V1)
le <- ly - mu.log
qqnorm(le, main = "log(Residuals) Q-Q Plot")
qqline(le, col = 2)
hist(le, probability = T, main = "log(Residuals) Histogram")
curve(dnorm(x, mean(le), sd(le)), add = T, col = 2)
lillie.test(le)

```

4. נניח כי אנחנו מבצעים ANOVA חד-כוננית ורוצים לבדוק האם השאריות מפולגות נורמלית באמצעות סטטיסטי קולמוגורוב-סmirנוב, $D = \max_{t \in [-\infty, \infty]} \left| \hat{F}_n(t) - \Phi\left(\frac{t-\bar{x}}{s}\right) \right|$. כתבו פונקציה ב-R אשר מקבלת כקלט את הערך d ומחזירה קירוב עבור ההסתברות $P(D \geq d)$ במקרה בו השגיאות ϵ_{ij} אכן מתפלגות נורמלית עם שונות זהה. הריצו את הפונקציה במקרים הבאים:

- א. שתי קבוצות בעלות 15 תצפיות כל אחת, $d = 0.5$
- ב. שלוש קבוצות בעלות 10 תצפיות כל אחת, $d = 0.5$

```

source("lilli.r")
generate_anova_residuals <- function(I, n_i){
  mu_i <- runif(I,0,100)
  x <- NULL
  for(i in 1:I){
    x <- c(x, rnorm(n_i, mu_i[i], 1))
  }
  df <- data.frame(group = sort(rep(1:I,n_i)), x = x)
  e <- df$x - ave(df$x, df$group)
  return(e)
}
lilli.pval.anova <- function(d.obs, I, n_i, nsim) {
  llsm <- NULL
  for (s in 1:nsim) {
    x <- generate_anova_residuals(I, n_i)
    d <- my.lilli(x)
    llsm <- c(llsm,d)
  }
  pval <- mean(llsm >= d.obs)
  return(pval)
}
d <- 0.5
print(lilli.pval.anova(d, 2, 15, 100000))
print(lilli.pval.anova(d, 3, 10, 100000))

```