중간고사

```
group = rep(1:5,each=4)

y = c(2.4, 2.7, 3.1, 3.1, 0.7, 1.6, 1.7, 1.8, 2.4, 3.1,
      5.4, 6.1, 0.3, 0.3, 2.4, 2.7, 0.5, 0.9, 1.4, 2.0)

sol = cbind(group, y)
group = as.factor(group) # 숫자 자체의 의미 x
aov1 = aov(y~group) # y라는 데이터가 grouyp에 따라 차이가 있냐?
summary(aov1) # 분산분석
```

가설 설정

$$H_0 : \tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5, \quad H_1 : \text{not } H_0$$

1 way ANOVA table

```
> summary(aov1)
            Df Sum Sq Mean Sq F value  Pr(>F)
group        4  27.01   6.752   5.966 0.00444 **
Residuals   15  16.98   1.132
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- 유의수준 0.05에서 p-value < 0.05이므로 귀무가설 기각 => 적어도 한 그룹의 효과는 다른 것이다

f-value로 p-value 구하기

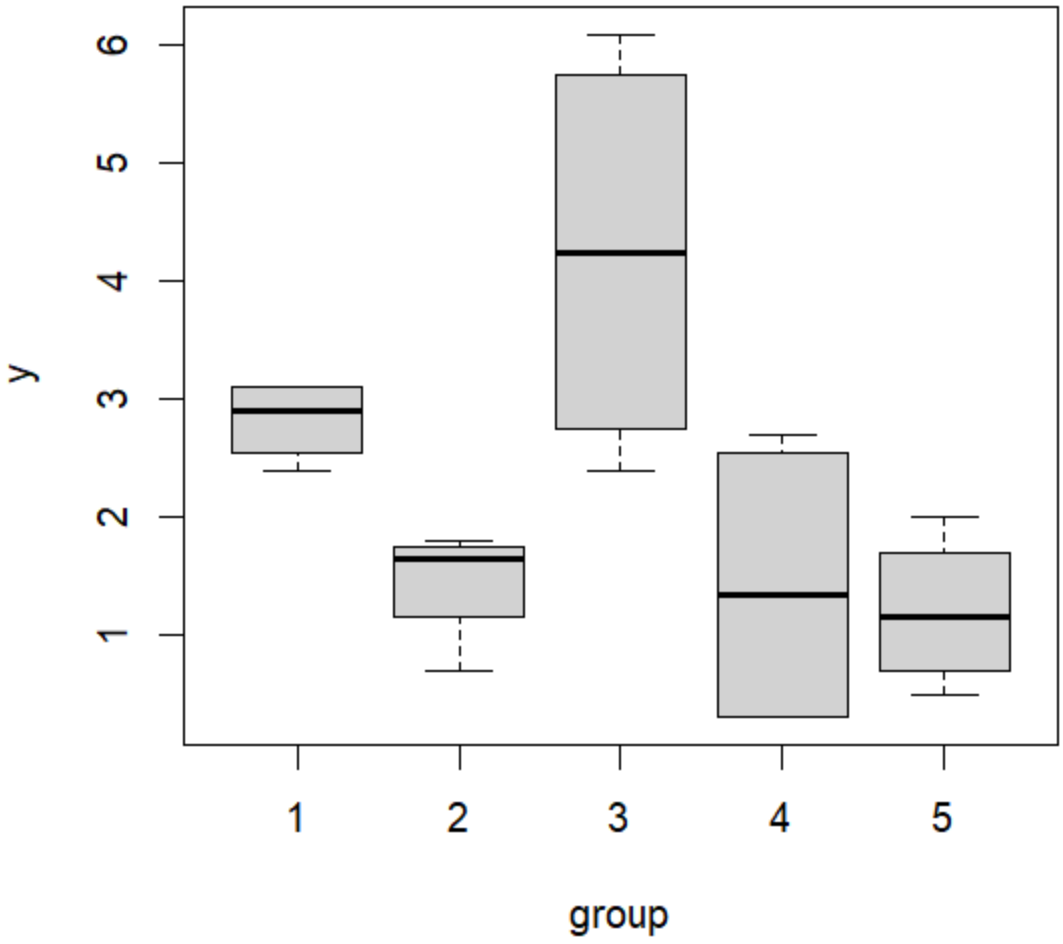
```
1 - pf(5.966, 4, 15)
```

유의수준 0.05에서 기각역 구하기

```
qf(0.95, 4, 15)
```

factor 별로 상자 그림 출력

```
plot(y~group)
```



- group5는 다른 그룹보다 `중앙값`이 높음 (`평균`이 아님)
- 1, 3, 5의 그룹 분포가 달라보임

- 표본 크기가 작거나, 오차 가정을 지키지 못한다면 귀무가설을 기각하지 못할 수 있음
- `다중비교`로 어떻게 다른지 확인해야 함

회귀분석 형태로 확인

```
> summary.lm(aov1)

Call:
aov(formula = y ~ group)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8500 -0.7125  0.1750  0.4625  1.8500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.8250     0.5319   5.311 8.72e-05 ***
group2       -1.3750     0.7522  -1.828   0.0875 .
group3        1.4250     0.7522   1.894   0.0776 .
group4       -1.4000     0.7522  -1.861   0.0824 .
group5       -1.6250     0.7522  -2.160   0.0473 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.064 on 15 degrees of freedom
Multiple R-squared:  0.614,     Adjusted R-squared:  0.5111
F-statistic: 5.966 on 4 and 15 DF,  p-value: 0.004442
```

- group 1을 기준으로 비교
- group5는 유의수준 0.05에서 유의하다. -> 음수이므로 group1보다 평균이 유의하게 낮음
- 다른 그룹들도 0.01에서는 유의성을 보일 수 있으나 엄밀하게 통계적으로 결정짓기는 어려움

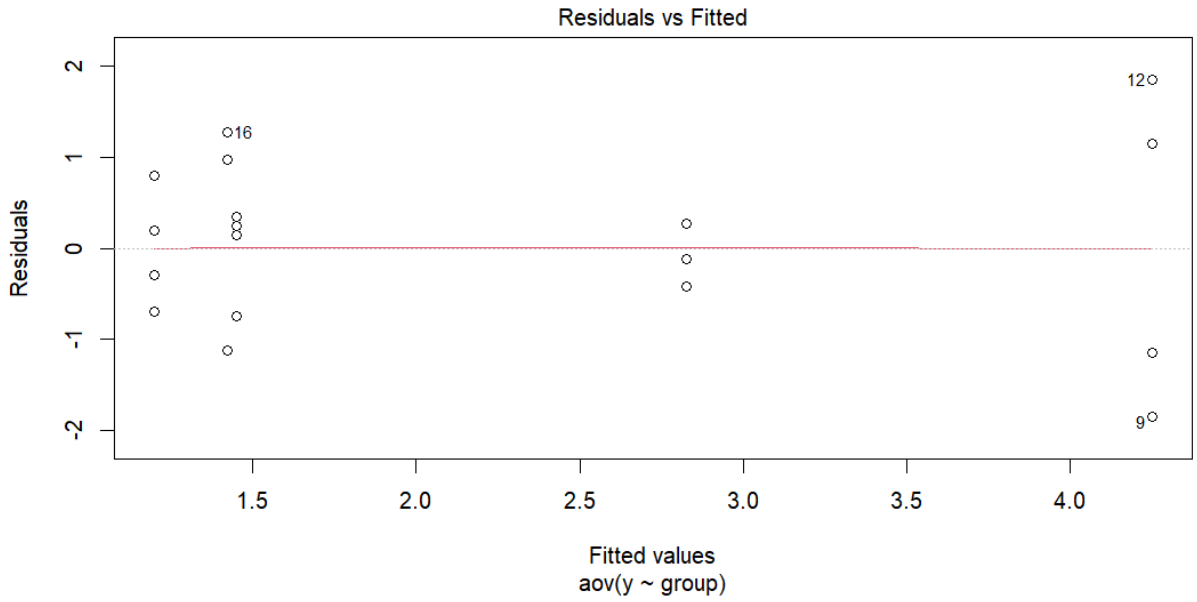
모형에 대한 가정

```
plot(aov1)
```

$$\epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2)$$

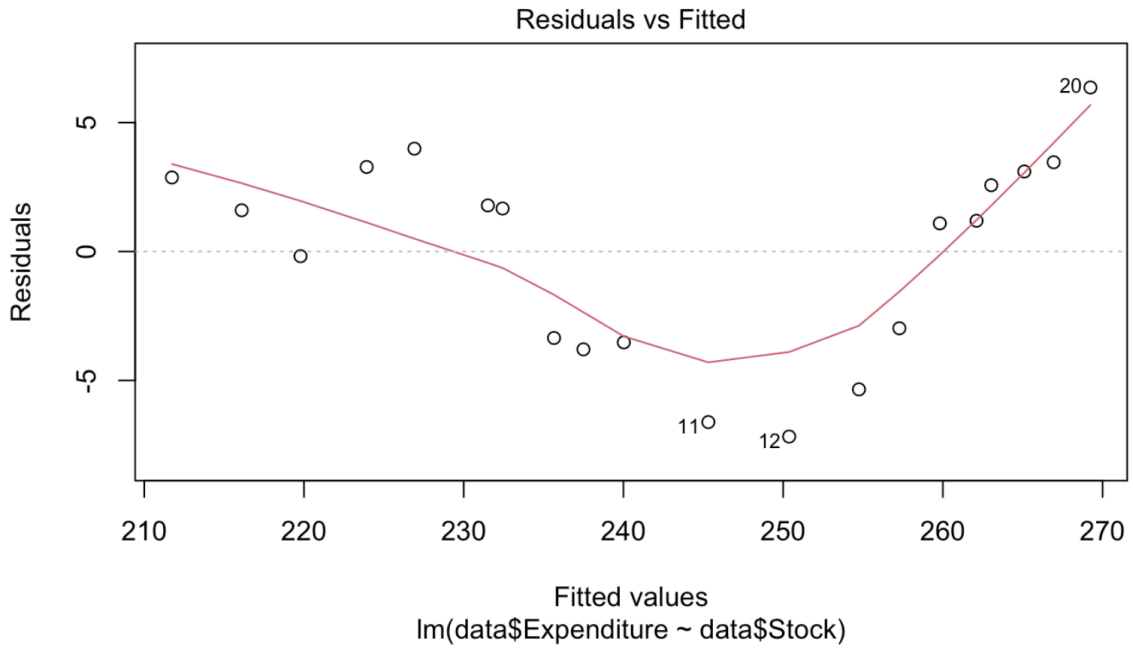
- 독립
- 정규분포
- 등분산성

잔차 그림



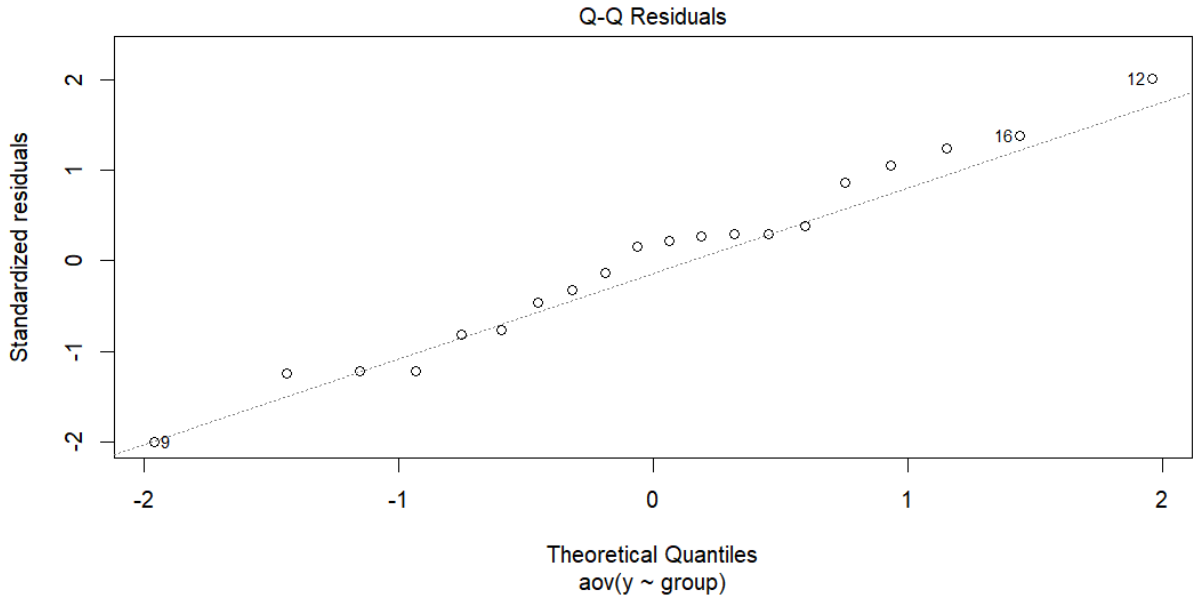
- 잔차 그림의 경우, 모형의 오차에 대한 가정 중 독립성, 등분산성을 만족하는 경우에 각각의 점으로 표현되는 잔차들이 y축 기준으로 0을 기준으로 위, 아래로 대체로 대칭적이며 특별한 패턴이 없이 나옴
- 하지만 실제 데이터가 이러한 오차에 대한 가정을 만족하지 못한다면 어떠한 패턴을 나타냄
- 따라서 이 잔차 그림을 통해 잔차들이 0을 기준으로 랜덤하게 나타나는지 확인하함으로써 오차에 대한 가정을 확인할 수 있음

- 그리고 이 때 빨간색 선은 잔차들의 평균을 직선으로 표현한 것. 따라서 이 직선이 y=0의 가로선과 비슷 하다면, 잔차의 평균이 0과 비슷하고 볼 수 있으며, 특정한 패턴이 없다고 판단할 수 있음



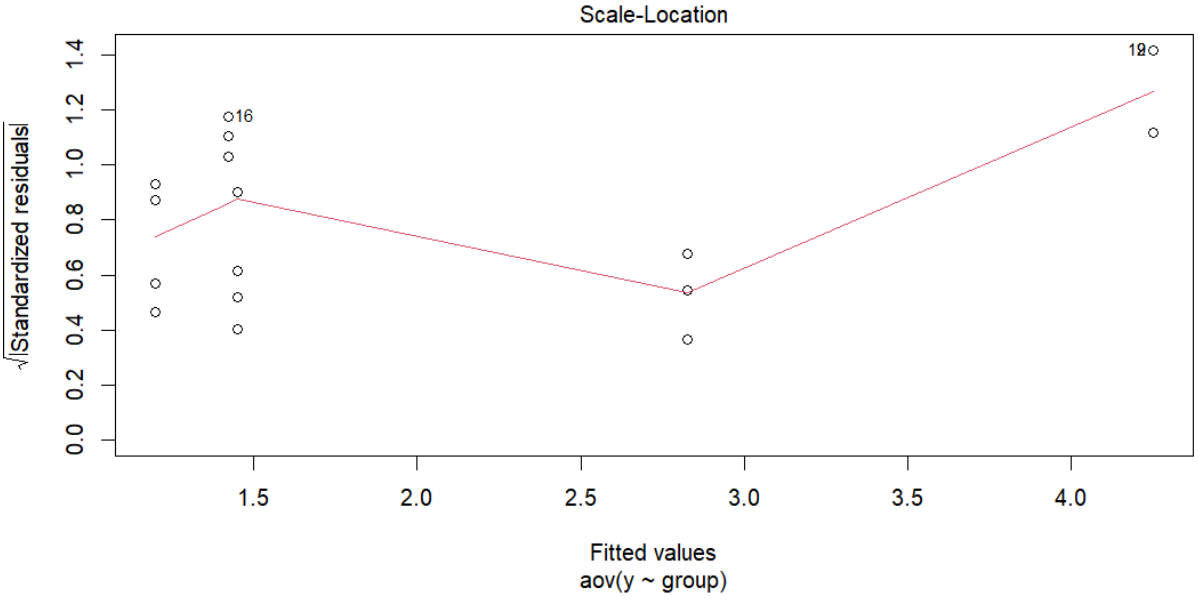
- 독립성 에 어긋나는 경우

Q-Q 그림



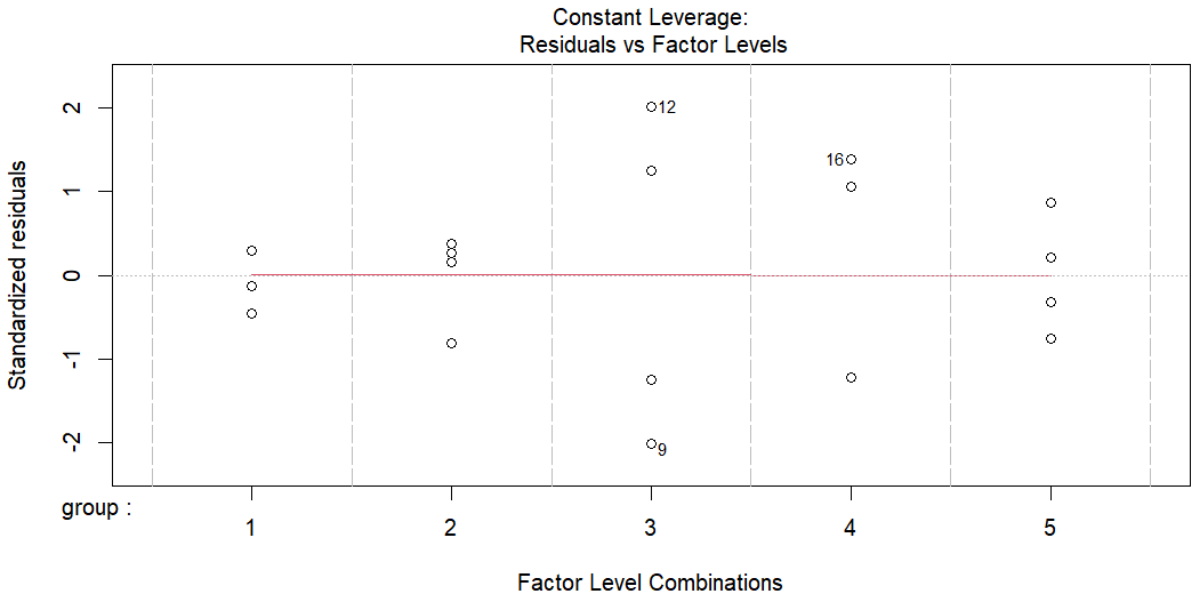
- 대각선 위에 모든 점이 위치해있다면, 정확한 정규분포 를 따름
- 많은 점들이 아니라 몇몇의 점들이 대각선에서 조금씩 떨어져 있더라고 정규분포를 따른다는 가정에서 크게 어긋나지 않음

표준화 잔차 그림

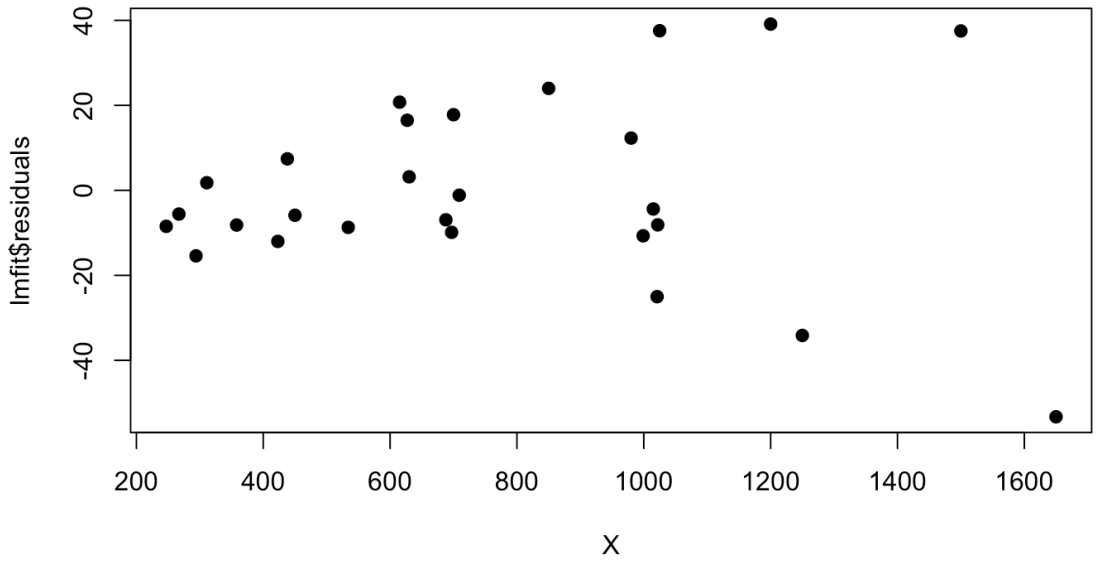


- 잔차들이 0-2 사이에 놓여있다면 데이터에 특별히 이상한 값 없이 정규분포를 따르는 데이터

그룹별 잔차



- 처리요인의 각 수준 마다 해당되는 잔차를 표현한 그림
- 각 처리 수준마다 비슷한 퍼짐을 나타낸다면, 오차의 분산이 처리 수준에 영향을 받지 않고 따라서 모든 수준에서 동일한 분포를 가짐
- 이 경우는 처리 수준에 대해서도 등분산성을 만족



- 점차 퍼져나가는 어떠한 패턴이 존재하는 경우는 등분산성 에 어긋나는 형태

등분산성 확인

바틀렛 검정

가설

$$H_0 : \sigma_{10}^2 = \sigma_{30}^2 = \sigma_{50}^2 = \sigma_{70}^2$$

$$H_1 : \text{적어도 한 개의 } \sigma_i^2\text{는 나머지와 다르다.}$$

- 여기서 1, 2, 3, 4, 5 5개 처리 수준으로 변경한..

```
bartlett.test(y~group)
```

```
> bartlett.test(y~group)

        Bartlett test of homogeneity of variances

data:  y by group
Bartlett's K-squared = 8.7824, df = 4, p-value = 0.06677
```

- 유의수준 0.05에서 p-value < 0.05이므로 귀무가설을 기각. 등분산성을 만족함

레빈 검정

```
install.packages("lawstat")
library(lawstat)
# levene.test(y, group, option=c("mean", "median", "trim.mean"), trim.alpha=
1)
levene.test(y, group, location = "mean")      # 평균 기반
levene.test(y, group, location = "median")    # 중앙값 기반
levene.test(y, group, location = "trimmed", trim.alpha = 0.1)  # 절사평균 기반
```

잔차

잔차 출력

`aov.sum$residuals` = `residuals(aov.sum)` =

처리효과 출력하기

`tapply(y, trt, mean) - mean(y)` = `model.tables(aov.sum)`

적합값 출력하기

`predict(aov.sum)` = `aov.sum$fitted`

다중 비교

분산분석에서 처리들의 효과가 모두 같다는 귀무가설이 기각됨. 구체적으로 어떤 처리가 다른 효과를 나타내는지 검정하기 위해 다중 비교 진행.

`tapply`로 각 그룹 평균 구하기

```
tapply(y, group, mean)
```

- group 5가 가장 평균이 낮고, group 3이 가장 평균이 높군!

LSD

0.05를 기준으로 p-value를 사용함

```
pairwise.t.test(y, group, p.adjust="none", pool.sd=T)
# none : LSD
# pool.sd=T 등분산을 가정함
```

```
> pairwise.t.test(y, group, p.adjust="none", pool.sd=T)

        Pairwise comparisons using t tests with pooled SD

data:  y and group

  1      2      3      4
2 0.0875 -      -      -
3 0.0776 0.0020 -      -
4 0.0824 0.9739 0.0019 -
5 0.0473 0.7442 0.0010 0.7690

P value adjustment method: none
```

5 4 2 1 3

(3,1) (4,1,2) (5,2,1)

- 평균을 오름차순으로 정렬
- 평균차가 유의하지 않으면 밑줄을 그림 (p-value > 0.05인 값들)
- 묶을 수 있는 그룹은 (5, 4, 2) (1, 3)
- 혹은 (5), (4, 2, 1), (3)도 가능

Bonferroni (pool.sd=T)

```
pairwise.t.test(y, group, p.adjust="bonferroni", pool.sd=T)
```

```
> pairwise.t.test(y, group, p.adjust="bonferroni", pool.sd=T)

        Pairwise comparisons using t tests with pooled SD

data:  y and group

  1     2     3     4
2 0.875 -     -     -
3 0.776 0.020 -     -
4 0.824 1.000 0.019 -
5 0.473 1.000 0.010 1.000

P value adjustment method: bonferroni
```

- 한 개의 그룹밖에 안 나옴
- LSD 방식에 비해 매우 보수적으로 처리평균 간 차이가 유의하다는 결론을 쉽게 내리지 않음
- LSD의 p-value에 비교 횟수만큼 p-value에 곱함 (기각값/검정개수) 5C2

Bonferroni (pool.sd=F)

```
pairwise.t.test(y, group, p.adjust="bonferroni", pool.sd=F)
```

Tukey HSD

```
a.tukey ← TukeyHSD(aov1)
a.tukey
plot(a.tukey) # 신뢰구간 출력
```

```
> a.tukey <- TukeyHSD(aov1)
> a.tukey
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = y ~ group)

$group
      diff        lwr        upr     p adj
2-1 -1.375 -3.6977953  0.9477953 0.3944760
3-1  1.425 -0.8977953  3.7477953 0.3612849
4-1 -1.400 -3.7227953  0.9227953 0.3776700
5-1 -1.625 -3.9477953  0.6977953 0.2466201
3-2  2.800  0.4772047  5.1227953 0.0149150
4-2 -0.025 -2.3477953  2.2977953 0.9999997
5-2 -0.250 -2.5727953  2.0727953 0.9970635
4-3 -2.825 -5.1477953 -0.5022047 0.0139855
5-3 -3.050 -5.3727953 -0.7272047 0.0078279
5-4 -0.225 -2.5477953  2.0977953 0.9980497
```

- group 3이 다름

- 95% 신뢰구간이 0을 포함하면 귀무가설을 기각하지 못함!

agricolae 패키지를 이용한 다중 비교

LSD

```
(LSD.test(aov1, "group", p.adj="none"))
```

```
> (LSD.test(aov1, "group", p.adj="none"))
$statistics
   MSerror Df Mean       CV t.value      LSD
  1.131667 15 2.23 47.70396 2.13145 1.603317

$parameters
        test p.ajusted name.t ntr alpha
  Fisher-LSD      none  group   5  0.05

$means
      y       std r        se        LCL      UCL Min Max   Q25  Q50   Q75
1 2.825 0.3403430 4 0.5318991 1.69128388 3.958716 2.4 3.1 2.625 2.90 3.100
2 1.450 0.5066228 4 0.5318991 0.31628388 2.583716 0.7 1.8 1.375 1.65 1.725
3 4.250 1.7785762 4 0.5318991 3.11628388 5.383716 2.4 6.1 2.925 4.25 5.575
4 1.425 1.3047988 4 0.5318991 0.29128388 2.558716 0.3 2.7 0.300 1.35 2.475
5 1.200 0.6480741 4 0.5318991 0.06628388 2.333716 0.5 2.0 0.800 1.15 1.550

$comparison
NULL

$groups
      y groups
3 4.250      a
1 2.825     ab
2 1.450     bc
4 1.425     bc
5 1.200      c

attr(,"class")
[1] "group"
```

a b c
3
1 1
2 2
4 4
5

- (1, 3) (2, 4, 5)로 그룹화됨

HSD

```
(HSD.test(aov1, "group", group=T))
```

```
> (HSD.test(aov1, "group", group=T))
$statistics
   MSerror Df Mean       CV      MSD
  1.131667 15 2.23 47.70396 2.322795

$parameters
   test name.t ntr StudentizedRange alpha
  Tukey  group   5         4.366985  0.05

$means
      y       std r        se Min Max   Q25  Q50   Q75
1 2.825 0.3403430 4 0.5318991 2.4 3.1 2.625 2.90 3.100
2 1.450 0.5066228 4 0.5318991 0.7 1.8 1.375 1.65 1.725
3 4.250 1.7785762 4 0.5318991 2.4 6.1 2.925 4.25 5.575
4 1.425 1.3047988 4 0.5318991 0.3 2.7 0.300 1.35 2.475
5 1.200 0.6480741 4 0.5318991 0.5 2.0 0.800 1.15 1.550

$comparison
NULL

$groups
      y groups
3 4.250      a
1 2.825     ab
2 1.450      b
4 1.425      b
5 1.200      b

attr(,"class")
[1] "group"
```

Scheffe

p-value 보정 동시 비교

```
(scheffe.test(aov1, "group", group=T))
```

```
> (scheffe.test(aov1, "group", group=T))
$statistics
   MSerror Df        F Mean       CV  Scheffe CriticalDifference
  1.131667 15 3.055568 2.23 47.70396 3.496037           2.629785

$parameters
     test name.t ntr alpha
  Scheffe  group   5  0.05

$means
      y       std r        se Min Max   Q25  Q50   Q75
1 2.825 0.3403430 4 0.5318991 2.4 3.1 2.625 2.90 3.100
2 1.450 0.5066228 4 0.5318991 0.7 1.8 1.375 1.65 1.725
3 4.250 1.7785762 4 0.5318991 2.4 6.1 2.925 4.25 5.575
4 1.425 1.3047988 4 0.5318991 0.3 2.7 0.300 1.35 2.475
5 1.200 0.6480741 4 0.5318991 0.5 2.0 0.800 1.15 1.550

$comparison
NULL

$groups
      y groups
3 4.250      a
1 2.825     ab
2 1.450      b
4 1.425      b
5 1.200      b

attr(,"class")
[1] "group"
```

대비

base에 대한 대비 만들기

```
contr.treatment(5, base=1, contrasts = T)
# 첫 번째 인자 -> 처리 수
# 두 번째 인자 -> 기준점
# 행렬 형태로 출력됨
```

```
contrasts(group) ← contr.treatment(5, base=2, contrasts = T)
summary.lm(aov(y~group))
```

```
> contrasts(group) <- contr.treatment(5, base=2, contrasts = T)
> summary.lm(aov(y~group))

Call:
aov(formula = y ~ group)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8500 -0.7125  0.1750  0.4625  1.8500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.4500     0.5319   2.726  0.01562 *
group1        1.3750     0.7522   1.828  0.08752 .
group3        2.8000     0.7522   3.722  0.00204 **
group4       -0.0250     0.7522  -0.033  0.97393
group5       -0.2500     0.7522  -0.332  0.74422
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.064 on 15 degrees of freedom
Multiple R-squared:  0.614,     Adjusted R-squared:  0.5111
F-statistic: 5.966 on 4 and 15 DF,  p-value: 0.004442
```

- group 3에 대해서 유의함

직교다항대비

```
contrasts(group) ← contr.poly(levels(group))
summary.lm(aov(y~group))
```

```
> contr.poly(levels(group))
                .L         .Q             .C         ^4
[1,] -6.324555e-01  0.5345225 -3.162278e-01  0.1195229
[2,] -3.162278e-01 -0.2672612  6.324555e-01 -0.4780914
[3,] -3.510833e-17 -0.5345225  1.755417e-16  0.7171372
[4,]  3.162278e-01 -0.2672612 -6.324555e-01 -0.4780914
[5,]  6.324555e-01  0.5345225  3.162278e-01  0.1195229
> contrasts(group) <- contr.poly(levels(group))
> summary.lm(aov(y~group))

Call:
aov(formula = y ~ group)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8500 -0.7125  0.1750  0.4625  1.8500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.2300     0.2379   9.375 1.16e-07 ***
group.L      -1.0356     0.5319  -1.947  0.07050 .
group.Q      -0.8886     0.5319  -1.671  0.11551
group.C      -0.4981     0.5319  -0.936  0.36391
group^4       2.1544     0.5319   4.050  0.00105 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.064 on 15 degrees of freedom
Multiple R-squared:  0.614,     Adjusted R-squared:  0.5111
F-statistic: 5.966 on 4 and 15 DF,  p-value: 0.004442
```

- 첫 번째 대비 기각 못 함
- 마지막 대비 유의수준보다 작음 -> 귀무가설 기각.

constrastmatrix에 직접 값을 저장하여 직교대비를 직접 생성

- k = 5이므로, 총 4개의 직교대비를 만들어야 함, 각 직교대비에는 5개의 값을 가지도록

```
contrastmatrix = cbind( c(-2, -1, 0, 1, 2),
                        c(2, -1, -2, -1, -2),
                        c(-1, 2, 0, -2, 1),
```

```
                c(1, -4, 6, -4, 1)
)

contrasts(group) = contrastmatrix
contrasts(group)
summary.lm(aov(y~group))
```

```
> contrasts(group)
  [,1] [,2] [,3] [,4]
1   -2    2   -1    1
2   -1   -1    2   -4
3    0   -2    0    6
4    1   -1   -2   -4
5    2   -2    1    1
> summary.lm(aov(y~group))

Call:
aov(formula = y ~ group)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8500 -0.7125  0.1750  0.4625  1.8500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.78667    0.35637   5.014 0.000154 ***
group1      -0.77083    0.31418  -2.454 0.026856 *
group2      -0.55417    0.33170  -1.671 0.115509
group3      -0.37917    0.21423  -1.770 0.097064 .
group4       0.22583    0.06634   3.404 0.003924 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.064 on 15 degrees of freedom
Multiple R-squared:  0.614,     Adjusted R-squared:  0.5111
F-statistic: 5.966 on 4 and 15 DF,  p-value: 0.004442
```

하나의 대비 구하기

구체적인 가설에 대한 F 검정

$$H_0 : \frac{\mu_1 + \mu_2 + \mu_3}{3} = \frac{\mu_4 + \mu_5}{2}$$

$$H_0 : 2\mu_1 + 2\mu_2 + 2\mu_3 - 3\mu_4 - 3\mu_5 = 0$$

$$t_L = \frac{\sum_{i=1}^{k} c_i \bar{Y}_{i\cdot}}{\sqrt{MSe \sum_{i=1}^{k} \frac{c_i^2}{n_i}}} \overset{H_0}{\sim} F_{1,N-k}$$

```
N = length(y)
k = 5
m = tapply(y, group, mean)
ni = tapply(y, group, length)
c1 = c(2,2,2,-3,-3)
summ.aov = summary(aov1)
mse = summ.aov[[1]]$`Mean Sq`[2]
f = sum(c1*m)^2 / (mse*sum(c1^2/ni))
1 - pf(f, 1, N-k) # 자유도가 1임!!
```

- 위 대비의 SSt 구하기

```
sst = sum(c1*m)^2 / (sum(c1^2/ni))
```

- 위 직교대비에서 가장 SSt 값이 많은 것은?

```
N = length(y)
k = 5
m = tapply(y, group, mean)
ni = tapply(y, group, length)
sst_list ← rep(0, k-1)
for(i in 1:(k-1)) {
  c ← contrasts(group)[,i] # contr.poly로 직교행렬 넣어줌, 위는 직교 행렬이 아니라
```

```
SSt 값이 일치하지 않음
  sst_list[i] ←  sum(c*m)^2 / (sum(c^2/ni))
}

sum(sst_list)
sst_list
```

완전 Tip

직교 대비는 행렬 관점에서 보면 직교 행렬임 (TT(t) = 단위행렬)

```
t(행렬) %*% 행렬
```

따라서 직교 대비인지 확인하기 위해서 저렇게 %*%를 해주자!

```
            .L           .Q           .C           ^4
.L  1.000000e+00  0.000000e+00 -2.775558e-17  5.551115e-17
.Q  0.000000e+00  1.000000e+00 -2.775558e-17 -1.110223e-16
.C -2.775558e-17 -2.775558e-17  1.000000e+00  0.000000e+00
^4  5.551115e-17 -1.110223e-16  0.000000e+00  1.000000e+00
```

<hr>

4장

```
# 처리별 평균 (bar(yi))
tapply(y, trt, mean)
# 블록별 평균
tapply(y, block, mean)

# 처리효과
tapply(y, trt, mean) - mean(y)
# 블록효과
tapply(y, block, mean) - mean(y)
```

```
ft = summ.aov[[1]]$`F value`[2]
# p-value 구하기
1 - pf(ft, 2, 6)
# 기각역 구하기
qf(0.95, 2, 6)

# 분산 = MSe
mse
```

LSD 직접 구하기

```
# LSD
pairwise.t.test(y, trt, p.adjust="none")

# LSD 값 직접 구하기
mse = summ.aov[[1]]$`Mean Sq`[3]
t ← qt(0.975, 6) # 주의 t 분포이고, 자유도는 오차자유도를 사용함
se ← sqrt(2 * mse / 4) # 블록 개수로 나눔 (반복수로 나눔)
lsd ← t * se

m ← tapply(y, trt, mean)
m[1] - m[2]
m[2] - m[3]
m[1] - m[3]
```

적합값, 잔차 출력

- 실제 값 = y
- 적합값 (00 평균) = `predict(aov)`, `aov$fitted`
- 오차 추정값 = `aov$residuals`, `y - predict(aov)`, `y - aov$fitted`

```
# 적합값 출력
aov.out$fitted
```

```
# 잔차 출력
aov.out$residuals
```