multivariate_lab_hw2

Na SeungChan

2024-12-13

1

Interpret the result of Box's M test. Can you assume equal covariance?

Box's M test의 H0는 공분산행렬이 동질적이라는 것이다. 즉 p-value가 유의수준 미만으로 나오면 귀무가설을 기각하여, 공분산행렬이 동질적이지 않고 그룹 간 차이가 있다고 결론을 내린다.

```
test_varequal <- BoxM(flea[,2:7], group = flea$species)
test_varequal$p.value
```

```
## [1] 0.2049986
```

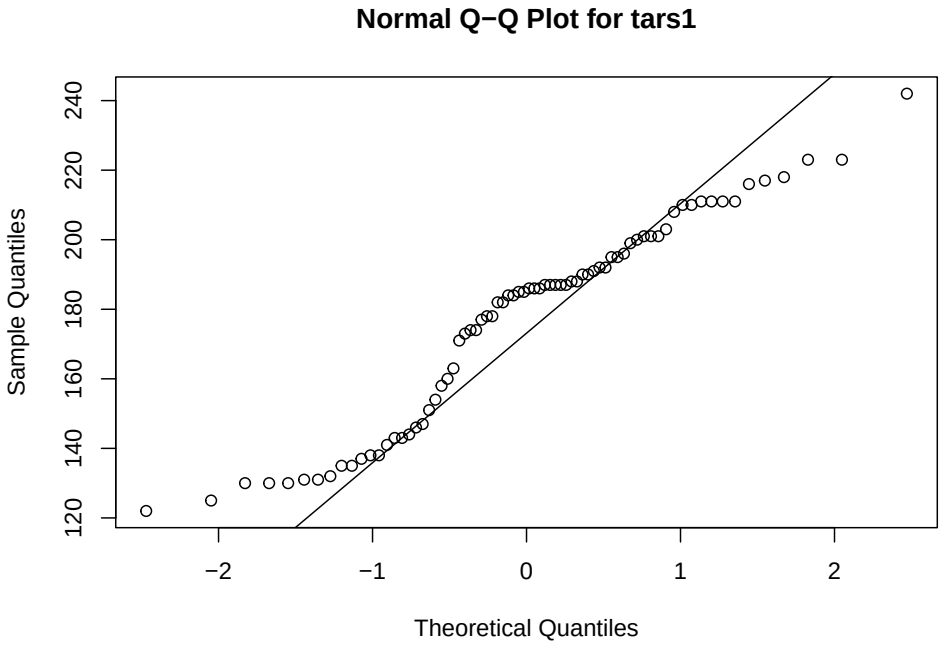
p.value = 0.2049986 > 0.05. flea beetles의 종에 따라 공분산행렬이 서로 다르다는 대립가설을 기각하지 못하며, 분산행렬이 서로 같다고 가정해도 될 것 같다. 이에 따라 MANOVA 등 모델을 사용할 수 있을 것이다.

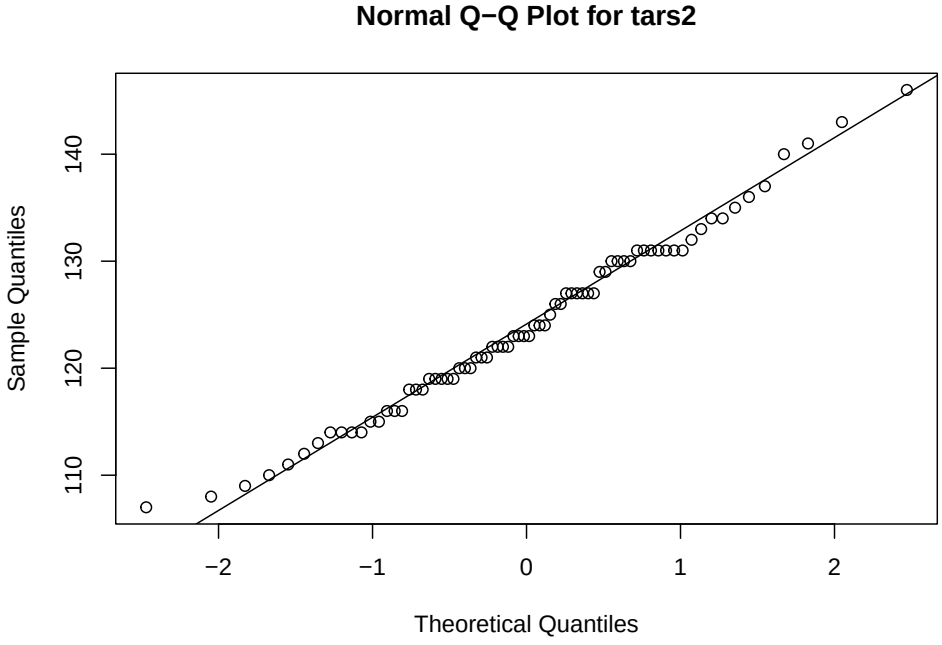
2

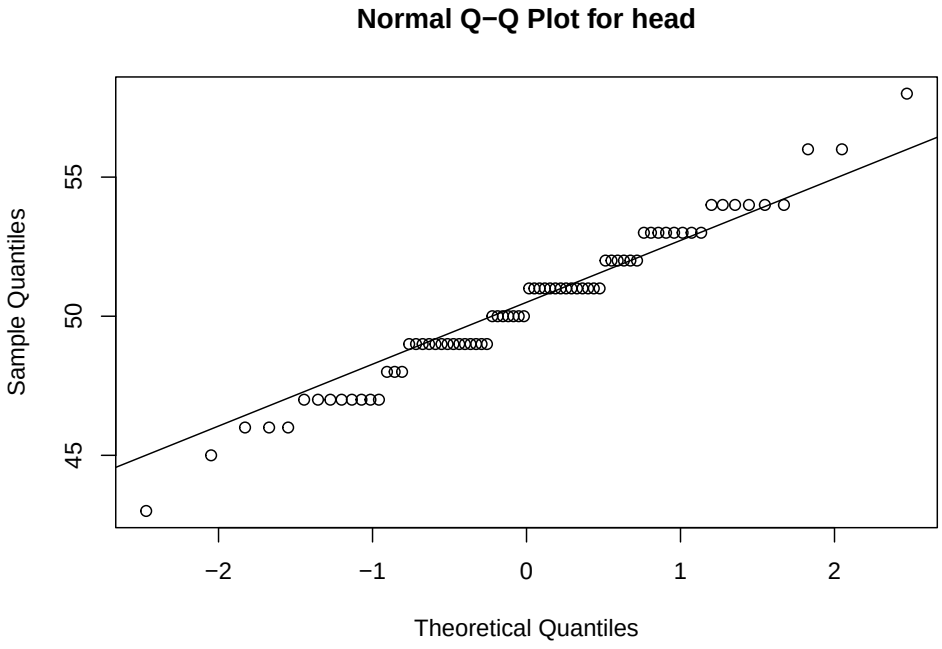
Box's M test assumes multivariate normality of each population. Is such an assumption valid? Can we assume normality for each population? (See Lab #2)

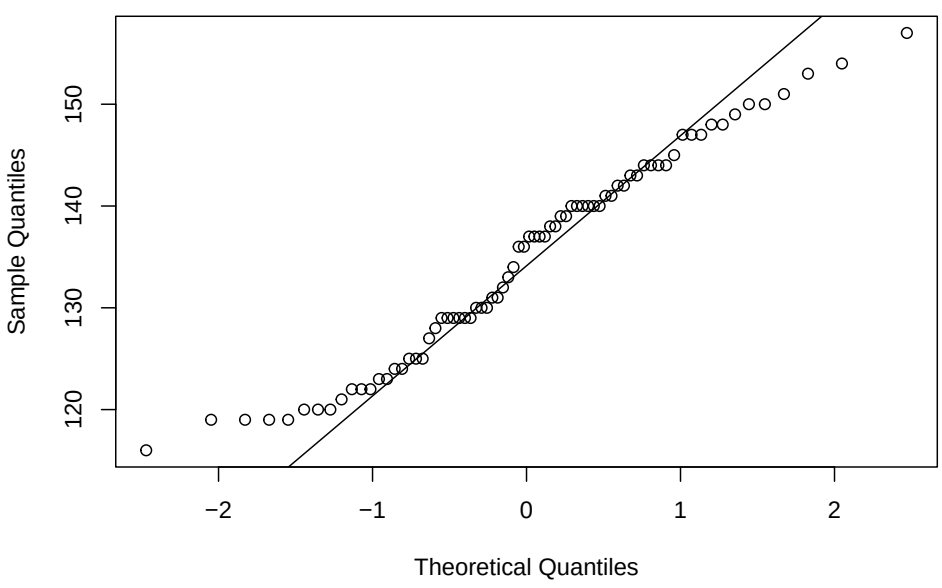
marginal normality : Q-Q plot for each variable

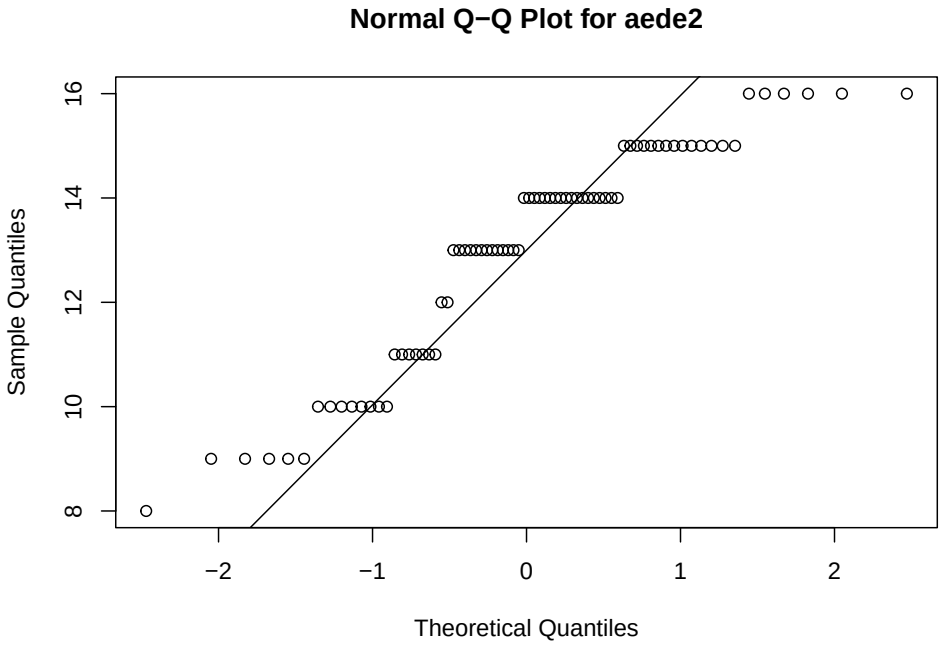
```
for (i in 2:7) {
  graph_name <- paste('Normal Q-Q Plot for', names(flea)[i])
  qqnorm(flea[,i], main = graph_name)
  qqline(flea[,i])
}
```

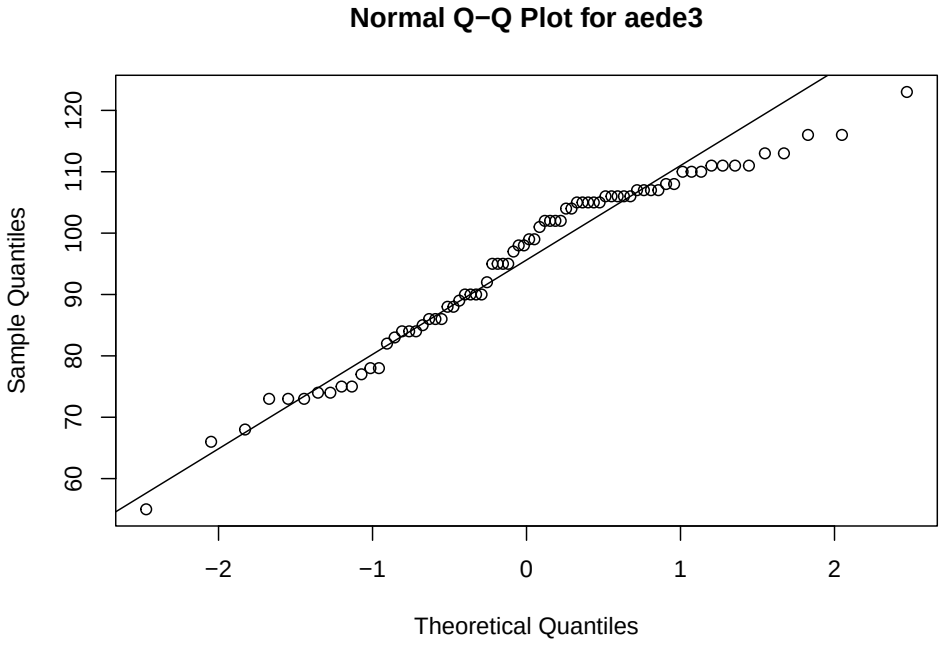

Normal Q−Q Plot for tars1
Sample Quantiles
120
140
160
180
200
220
240
−2
−1
0
1
2
Theoretical Quantiles


Normal Q−Q Plot for tars2
Sample Quantiles
110
120
130
140
−2
−1
0
1
2
Theoretical Quantiles


Normal Q−Q Plot for head
Sample Quantiles
45
50
55
−2
−1
0
1
2
Theoretical Quantiles


Normal Q−Q Plot for aede1
Sample Quantiles
120
130
140
150
−2
−1
0
1
2
Theoretical Quantiles



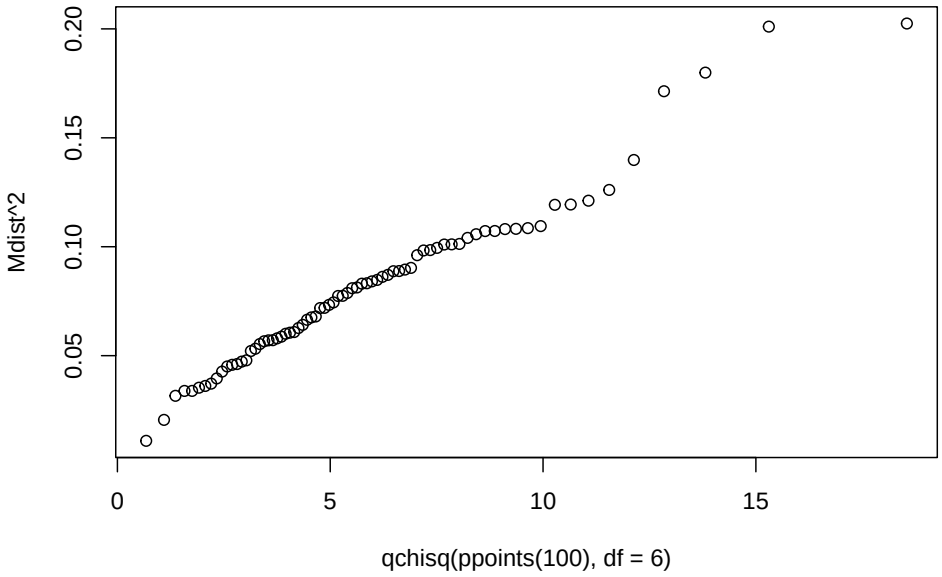


tars1, aede1, aede2 등에서 정규성 가정이 의심스러운 지점이 발견된다.

Joint normality : M-distance

```r
Xc <- scale(as.matrix(flea[, 2:7]), scale=FALSE)
S <- t(Xc) %*% Xc

Mdist <- sqrt(diag(Xc %*% solve(S) %*% t(Xc)))
qqplot(qchisq(ppoints(100), df=6), Mdist^2)
```



별로 직선에 가까워 보이지 않는다… 정규성 가정에 의심을 품고 검정을 진행해야 할 수 있다.

3

Manually compute the Between-variance matrix B from “mean_vectors”. Check whether your result is equal to B = Tot - W above.

Calculate B from B = total variance - W

```r
n <- nrow(flea)
p <- ncol(flea) - 1
G <- unique(flea$species)

# calculate W : within sum of squares
W <- matrix(0, nrow = p, ncol = p)
for (g in G) {
```

```
  gData <- as.matrix(flea[flea$species == g, 2:7])
  W <- W + t(scale(gData, scale=FALSE)) %*% (scale(gData, scale=FALSE))

}
Sp <- W / (n - length(G)) # pooled variance

# total variance = W + B = within variance + residual variance
Total_variance <- cov(flea[,2:7]) * (n-1) # total variance
B <- Total_variance - W # treatment variance
```

calculate B directly from mean vector

```
tmean <- c(colMeans(flea[, 2:7]))
B_new <- matrix(0, nrow = p, ncol = p)

for (g in G) {
  gData <- as.matrix(flea[flea$species == g, 2:7])
  n_g <- nrow(gData)
  gmean <- c(colMeans(gData))
  B_new <- B_new + n_g * outer(gmean-tmean, gmean-tmean)
}
```

B(tot - W)와 B_new(직접 계산) 비교

```
B
```

```
##         tars1     tars2     head     aede1     aede2     aede3
## tars1  51704.448 -3690.5371 -2045.2449 -9059.6608  3581.9018 -19048.520
## tars2  -3690.537  1367.3336   349.0408  2899.9057  -127.7174   3472.460
## head   -2045.245   349.0408   118.2533   772.8361  -118.1519   1142.130
## aede1  -9059.661  2899.9057   772.8361  6186.6528  -366.4563   7650.271
## aede2   3581.902  -127.7174  -118.1519  -366.4563   262.9717  -1074.726
## aede3 -19048.520  3472.4598  1142.1297  7650.2713 -1074.7255  11061.516
```

```
B_new
```

```
##         tars1     tars2     head     aede1     aede2     aede3
## tars1  51704.448 -3690.5371 -2045.2449 -9059.6608  3581.9018 -19048.520
## tars2  -3690.537  1367.3336   349.0408  2899.9057  -127.7174   3472.460
## head   -2045.245   349.0408   118.2533   772.8361  -118.1519   1142.130
## aede1  -9059.661  2899.9057   772.8361  6186.6528  -366.4563   7650.271
## aede2   3581.902  -127.7174  -118.1519  -366.4563   262.9717  -1074.726
## aede3 -19048.520  3472.4598  1142.1297  7650.2713 -1074.7255  11061.516
```

```
all.equal(B, B_new)
```

```
## [1] TRUE
```

두 행렬을 비교하면 서로 같음을 확인할 수 있다.

4

How many positive eigenvalues should you see in the above?

우선, 모델을 먼저 적합한다.

```
result_m <- manova(as.matrix(flea[,2:7]) ~ species, data = flea)
result_p <- summary(result_m, test = "Pillai") # default.
result_p
```

```
##           Df Pillai approx F num Df den Df    Pr(>F)
## species    2 1.7421   75.413     12    134 < 2.2e-16 ***
## Residuals 71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
result_w <- summary(result_m, test = "Wilks")
result_w
```

```
##           Df  Wilks approx F num Df den Df    Pr(>F)
## species    2 0.0109   94.359     12    132 < 2.2e-16 ***
## Residuals 71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
result_h <- summary(result_m, test = "Hotelling-Lawley")
result_r <- summary(result_m, test = "Roy")
```

다음으로, 비교 대상이 되는 행렬들을 확인하자. 부동소수점 오차를 고려하여 all.equal() 함수를 사용하여 B, W와 MANOVA 결과의 행렬들을 비교한다.

```
all.equal(result_p$SS$species, B)
```

```
## [1] TRUE
```

```
all.equal(result_p$SS$Residuals, W)
```

```
## [1] TRUE
```

```r
all.equal(result_w$SS$species, B)
```

```
## [1] TRUE
```

```r
all.equal(result_w$SS$Residuals, W)
```

```
## [1] TRUE
```

비교 결과, 앞서 계산했던 B, W와 manova() 함수를 활용해 계산한 B, W가 서로 같음을 확인할 수 있다. 그렇다면, 총 몇 개의 eignevalues가 양수여야 하는가? 당연히 B와 W 모두 적어도 Non-Negative matrix이므로 B, W 각각 6개씩 총 12개의 고윳값이 nonnegative여야 한다. $W^{-1} * B$ 역시 nonnegative여야 하는데, 이를 포함하면 6개가 추가되어 18개이다. 그런데!

```r
sum(eigen(B)$values > 0)
```

```
## [1] 4
```

```r
sum(eigen(W)$values > 0)
```

```
## [1] 6
```

B에서 고윳값이 음수이다. 왜인가? 수치선형대수상의 오차 때문이다. 특성다항식의 풀이는 오차가 꽤나 발생하는 풀이로, 특히 B와 같이 행렬을 구성하는 각 원소의 사이즈가 큰 경우에는 이와 같이 음수가 될 수 있다.

5

All of the MANOVA Tests above are not valid if normality is not assumed. For such cases, one may use permutation-based test. Inspect the following lines and report your conclusion.

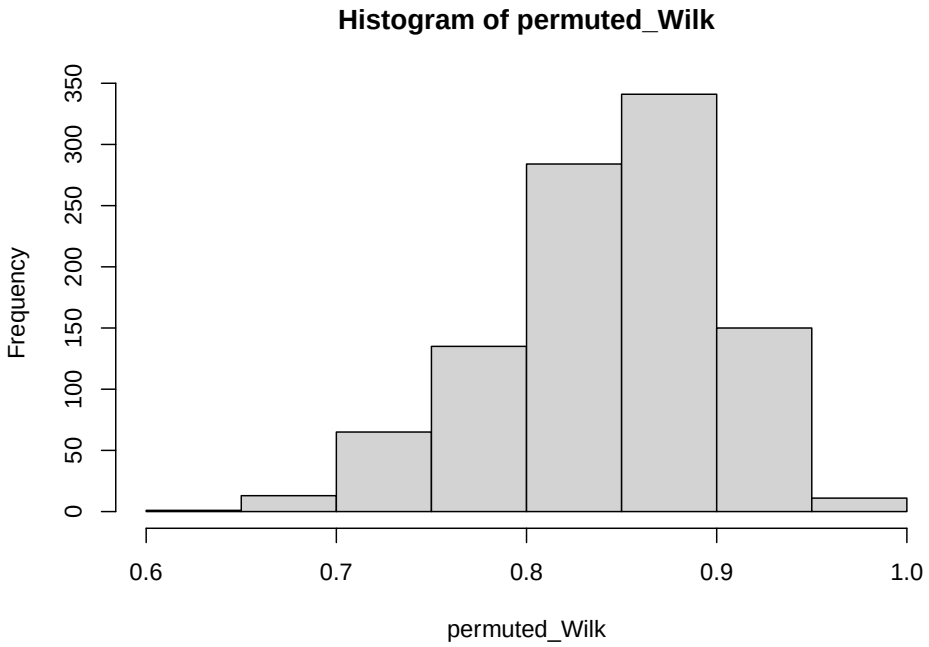
순열 검정의 귀무가설은 'null hypothesis of exchangeability'로, 집단 간 차이가 없다면 샘플 간 교환을 수행하였을 때(순열 함수를 적용하였을 때의 데이터)의 검정통계량이 원래 데이터의 통계량과 큰 차이가 없을 것이라는 것이다.

```r
#library(purrr)
library(modelr)
perms <- permute(flea, 1000, species)
permuted_Wilk <- map(perms$perm,
        ~ summary(manova(as.matrix(flea[,2:7]) ~ species, data = .),test = "Wilks")$stats[3])
permuted_Wilk <- unlist(permuted_Wilk)

actual_Wilk <- result_w$stats[3]
p_value <- sum(permuted_Wilk <= actual_Wilk)/1000
p_value
```

```
## [1] 0
```

```
hist(permuted_Wilk)
```



library(purrr) : map() 함수를 쓰기 위해 사용하였다. 위에서 tidyverse 라이브러리를 로드하였으므로 해당 블록에서는 주석 처리하였다.

library(modelr) : permute() 함수를 통해 순열 검정을 수행하기 위해 로드하였다.

perms : permute() 함수를 통해 순열 검정을 위한 데이터 섞기를 수행하였다. flea는 data, 1000은 만들 총 순열의 개수, species는 permute할 칼럼이다. 예시 코드의 n = 100과 달리 n = 1000으로 수행하였다.

map() + unlist() : map()은 dataframe 등에 적용되어 list 자료형을 결과로 return하는 함수이다. 구체적으로, map() 함수는 dataframe의 각 원소마다 특정한 함수를 적용하고, 그 함수의 결과를 list로 반환한다. 해당 코드에서는 perms list에 있는 각 계승마다 manova를 적용하고, 그로부터 Wilk 검정통계량을 계산하여 그 값을 리스트로 반환한다. 이후 unlist() 함수를 적용하는 것은 리스트를 다시 values sequence로 변환한 것이다.

actual_Wilk : 앞서 Wilk 검정통계량을 계산한 모델이 있으므로 이로부터 실제 검정통계량과 비교하여 p-value를 계산하기 위해 추가하였다.

히스토그램에 별도의 p-value를 표기하지 않았다. 0에 가까운 값이므로 표기하기 위해 축을 변형하는 것이 축을 오히려 왜곡하여 오도할 것으로 판단하였다.

p_value < 0.0001 수준의 매우 작은 값이므로 그룹 간 차이가 있다고 판단할 수 있다.