STA104 - Homework 3

Xiaowei Zeng

2023-11-07

Problem 1

Load the dataset for problem 1.

```
library(readxl)
data1 = read_xlsx('data for problem 1 page 105.xlsx')
head(data1)
```

```
## # A tibble: 6 x 2
##   treat   count
##   <chr>   <dbl>
## 1 control  4.30
## 2 control  4.02
## 3 control  4.05
## 4 control  4.18
## 5 p1       2.02
## 6 p1       3.19
```

a. Apply the permutation F-test to the data.

Compute F_{obs} statistic for the data.

```
fit1 = lm(count ~ treat, data = data1)
F_obs = summary(fit1)[[10]][1]
F_obs
```

```
##   value
## 12.5708
```

The total number of permutation is

$$\frac{N!}{n_1!n_2!n_3!} = \frac{15!}{4!6!5!} = 630630,$$

which is too large for us to compute the exact permutation p-value. So we use simulation to conduct the permutation F-test.

```
permut_num = 10000
f = rep(0, permut_num)
set.seed(2023) # for reproducibility
for (i in 1:permut_num){
  permut = sample(data1$count)
  fit = lm(permut ~ data1$treat)
  f[i] = summary(fit)[[10]][1]
}
p_value1 = sum(f >= F_obs) / permut_num
p_value1
```

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

The approximate p-value of the permutation F-test is 0, indicating that we should reject the null hypothesis at any significance level.

b. Compare the results in part a with the results of the usual one-way analysis of variance.

Conduct the usual one-way anova test.

```
fit2 = aov(count ~ treat, data = data1)
p_value2 = summary(fit2)[[1]][['Pr(>F)']][1]
p_value2
```

```
## [1] 0.001137426
```

The p-value of anova is 0.00114. Suppose that the significance level is 0.05, so we should reject the null hypothesis, which is the same conclusion as the permutation test.

Problem 2

Load the dataset for problem 2.

```
data2 = read_xlsx('data for problem 2 page 105.xlsx', range = 'A1:B48')
colnames(data2) = c('weight', 'load')
data2$weight = factor(data2$weight)
head(data2)
```

```
## # A tibble: 6 x 2
##   weight  load
##   <fct>  <dbl>
## 1 1        574
## 2 1        926
## 3 1        789
## 4 1        805
## 5 1        361
## 6 1        529
```

a. Apply the permutation F-test and the ANOVA F-test to the data, and compare p-values.

Compute F_{obs} statistic for the data.

```r
fit1 = lm(load ~ weight, data = data2)
F_obs = summary(fit1)[[10]][1]
F_obs
```

```
##    value
## 0.948123
```

The total number of permutation is also too large for us to compute the exact permutation p-value. So we use simulation to conduct the permutation F-test.

```r
permut_num = 10000
f = rep(0, permut_num)
set.seed(2023) # for reproducibility
for (i in 1:permut_num){
  permut = sample(data2$load)
  fit = lm(permut ~ data2$weight)
  f[i] = summary(fit)[[10]][1]
}
p_value1 = sum(f >= F_obs) / permut_num
p_value1
```

```
## [1] 0.4445
```

Suppose that the significance level is 0.05. The approximate p-value of the permutation F-test is $0.4445 > 0.05$, indicating that we cannot reject the null hypothesis.

Then conduct the anova test.

```r
fit2 = aov(load ~ weight, data = data2)
p_value2 = summary(fit2)[[1]][['Pr(>F)']][1]
p_value2
```

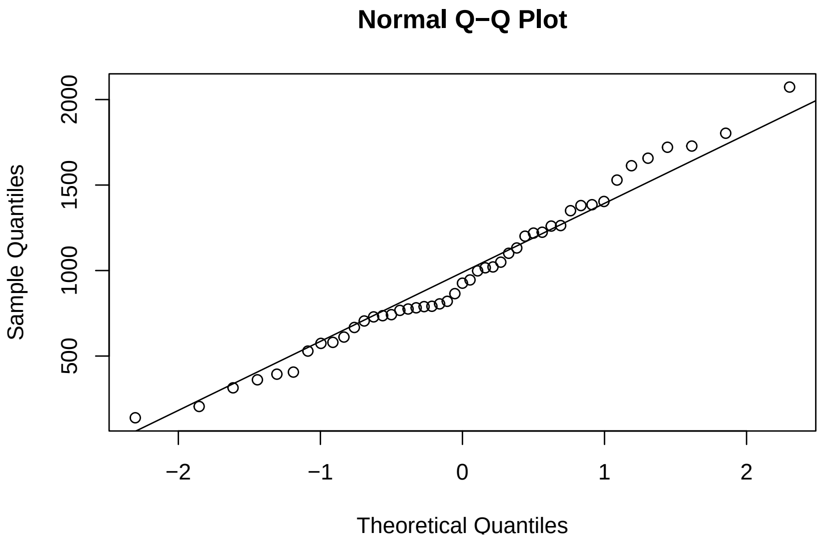
```
## [1] 0.4458401
```

The p-value of anova is $0.4458 > 0.05$, indicating that we cannot reject the null hypothesis, which is the same conclusion as the permutation test. The p-values of both tests are really close.

b. Does it appear that the data are normally distributed?

Since the p-values of anova and the permutation test don't differ much, we can assume that the data may be normally distributed. We can also check this assumption by Q-Q plot and Shapiro-Wilk normality test.

```r
qqnorm(data2$load); qqline(data2$load)
```



The data points almost form a straight line, suggesting that it approximately follows the normal distribution.

```
shapiro.test(data2$load)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data2$load
## W = 0.98059, p-value = 0.6167
```

The *p*-value is 0.6167, much larger than 0.05, so we cannot reject the normality of the data.

Problem 3

a. Apply the Kruskal-Wallis test to the data in Problem 1.

Define a function to calculate the ranks and the Kruskal-Wallis statistics.

```
library(dplyr)
calc_KW = function(N, treat, values){
  temp = data.frame(treat = treat, rank = rank(values))
  df_group = temp %>% group_by(treat) %>% summarise(n = length(rank), mr = mean(rank))
  KW = 12 / (N * (N + 1)) * sum(df_group$n * (df_group$mr - (N + 1) / 2) ^ 2)
  return(KW)
}
```

Obtain the observed KW statistics.

```
N = nrow(data1)
KW_obs = calc_KW(N, data1$treat, data1$count)
KW_obs
```

```
## [1] 12.375
```

The total number of permutation is also too large for us to compute the exact permutation p-value. So we use simulation to conduct the permutation Kruskal-Wallis test.

```
permut_num = 10000
kw = rep(0, permut_num)
set.seed(2023) # for reproducibility
for (i in 1:permut_num){
  permut = sample(data1$count)
  kw[i] = calc_KW(N, data1$treat, permut)
}
p_value3 = sum(kw >= KW_obs) / permut_num
p_value3
```

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

b. Compare the conclusions with those obtained in Problem 1.

Suppose that the significance level is 0.05. The approximate p-value of the permutation Kruskal-Wallis test is 0, the same as the permutation F-test; the p-value of the one-way anova test is 0.00114; all of the p-values are smaller than 0.05. So we can conduct the same conclusion that we should reject the null hypothesis.

Problem 4

Load the dataset for problem 6.

```
data6 = read_xlsx('data for problem 6 page 106.xlsx', range = 'A1:B71')
colnames(data6) = c('type', 'injury')
data6$type = factor(data6$type)
head(data6)
```

```
## # A tibble: 6 x 2
##   type  injury
##   <fct>  <dbl>
## 1 1        791
## 2 1        846
## 3 1       1024
## 4 1       1007
## 5 1       1399
## 6 1       1279
```

a. Test for differences among the groups using the Kruskal-Wallis test.

Use the function defined before. Obtain the observed KW statistics.

```
N = nrow(data6)
KW_obs = calc_KW(N, data6$type, data6$injury)
KW_obs
```

```
## [1] 14.80382
```

The total number of permutation is also too large for us to compute the exact permutation p-value. So we use simulation to conduct the permutation Kruskal-Wallis test.

```
permut_num = 10000
kw = rep(0, permut_num)
set.seed(2023) # for reproducibility
for (i in 1:permut_num){
  permut = sample(data6$injury)
  kw[i] = calc_KW(N, data6$type, permut)
}
p_value = sum(kw >= KW_obs) / permut_num
p_value
```

```
## [1] 0.0132
```

Suppose that the significance level is 0.05. The p-value of the permutation KW test is $0.0132 < 0.05$, indicating that we should reject the null hypothesis.

b. Separate means using the rank versions of the LSD and HSD criteria.

Since the result of the permutation K-W test is significant at level $\alpha = 0.05$, we can use ranked-based LSD and HSD criteria to conduct $7\mathbf{C}2 = 21$ pairwise comparisons.

i. Rank version of LSD criteria. We declare the distributions of treatments i and j to be different if

$$\left|\bar{R}_i - \bar{R}_j\right| \geq z_{1-\frac{\alpha}{2}} \sqrt{\frac{N(N+1)}{12}\left(\frac{1}{n_i} + \frac{1}{n_j}\right)}.$$

Since the number of samples in each group is the same in this problem, i.e. $n_i = n = 10, i = 1, \ldots, 7$, we can simplify this expression into

$$\left|\bar{R}_i - \bar{R}_j\right| \geq z_{1-\frac{\alpha}{2}} \sqrt{\frac{N(N+1)}{6n}}.$$

Compute the results.

```
data6$rank = rank(data6$injury)
df_group = data6 %>% group_by(type) %>% summarise(n = length(rank), mr = mean(rank))
k = length(df_group$type)
n = df_group$n[1]
tmp = matrix(rep(df_group$mr, k), nrow = k)
test_stat = abs(t(tmp) - tmp)
lsd = qnorm(1 - 0.05 / 2) * sqrt((N * (N + 1) / (6 * n)))
test_stat >= lsd
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## [3,] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
## [5,] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
## [6,] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [7,] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
```

The results show that six pairs, (type2, type3), (type2, type5), (type2, type6), (type2, type7), (type4, type5) and (type4, type7), are significantly different from each other.

ii. Rank version of HSD criteria. We declare the distributions of treatments i and j to be different if

$$\left|\bar{R}_i - \bar{R}_j\right| \geq q(\alpha, k, \infty)\sqrt{\frac{N(N+1)}{12n}}.$$

Compute the results.

```r
hsd = qtukey(1 - 0.05, k, Inf) * sqrt((N * (N + 1) / (12 * n)))
test_stat >= hsd
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [7,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

The results show that only one pair, (type2, type5), is significantly different from each other.

Problem 5

Obtain the upper 10% and 5% critical values of the permutation version of Tukey's HSD for the data in Problem 6.

The Tukey multiple comparison statistic is defined as

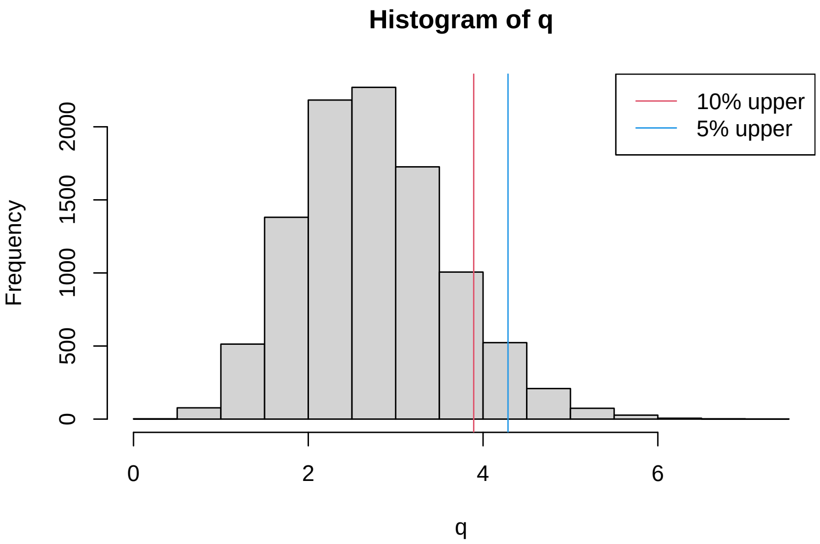
$$Q^* = \frac{\max(\bar{X}_i) - \min(\bar{X}_j)}{\sqrt{\text{MSE}/n}}.$$

Simulate the permutation distribution of Q^*.

```r
permut_num = 10000
q = rep(0, permut_num)
set.seed(2023) # for reproducibility
for (i in 1:permut_num){
  permut = sample(data6$injury)
  tmp = data.frame(type = data6$type, injury = permut)
  df_group = tmp %>% group_by(type) %>% summarise(mean = mean(injury), var = var(injury))
  mse = (n - 1) * sum(df_group$var) / (N - k)
  q[i] = (max(df_group$mean) - min(df_group$mean)) / sqrt(mse / n)
}
```

Plot the histogram of Q^* and the corresponding critical value.

```r
u.1 = quantile(q, 0.9)
u.05 = quantile(q, 0.95)
hist(q); abline(v = u.1, col = 2); abline(v = u.05, col = 4)
legend('topright', c('10% upper', '5% upper'), col = c(2, 4), lty = 1)
```



The upper 10% critical value is 3.89.

```
u.1
```

```
##      90%
## 3.892907
```

The upper 5% critical value is 4.29.

```
u.05
```

```
##      95%
## 4.285559
```