Lecture 4: Kruskal-Wallis Test

Caden Hewlett

2024-01-17

The Kruskall-Wallis Test : Overview

The motivation for this test is comparison across multiple groups. These can arise from samples from multiple (3 or more) populations, and a study interested in comparing the means across populations. Similarly, we have an experiment in which responses are from multiple treatment groups, or a combination of factors and assignment conditions (blocks, etc.)

It is a similar procedure to the Wilcoxon Rank-Sum test, however, we do not have two groups anymore.

Motivating Example.

To test whether or not the alchohol, placebo, or motivation (monetary) had an impact, we could test the hypothesis:

$$H_0 : \mu_{\text{A}} = \mu_{\text{B}} = \mu_{\text{C}} = \mu_{\text{D}} = \mu$$

Against the alternative that at least one is not equal to the others, where μ is the overall mean. Under H_0, the group means should be near the grand mean.

You can just as easily phrase H_0 as "the distributions are the same for all three."

It is considered an alternative to ANOVA. However, ANOVA requires certain assumptions. 1. All Variances are the Same : *Homoskedasticity* 2. All distributions are Normal 3. All groups/observations are independent.

The Kruskal-Wallis Procedure.

Let's consider the data from the textbook example.

```
df <- data.frame(
  A = c(21, 23, 13, 19, 13, 19, 20, 21, 16),
  B = c(28, 30, 29, 24, 27, 30, 28, 28, 23),
  C = c(19, 28, 26, 26, 19, 24, 24, 22, 23),
  D = c(21, 14, 13, 19, 15, 15, 10, 18, 20)
)
```

In situations such as the above, for the Kruskal-Wallis test, it is recommended that you use a null hypothesis such as

$$H_0 : \text{The data in the groups are all from identical distributions.}$$

Primarily, this is because we can *no longer assume that the data are normal.*

Notably, if we were to use the language of STAT 404, letting τ_i be the i-th treatment effect, we could retain the structure of:

$$H_0 : \forall i \in [1, k], \tau_i = 0$$

Against the alternative:

$$H_A : \exists i \in [1, k] \text{ s.t. } \tau_i \neq 0$$

Regardless, we can employ similar hypotheses for Kruskal-Wallis.

Kruskal-Wallis Test: The Procedure.

Let's assume that we have k different groups, with sample sizes $n_1, n_2, \dots n_k$ such that $\sum_{i=1}^{k}(n_i) = N$, where N is the grand number of observations.

1.) The first step is to pool all of the observations together, while retaining "memory" of their original group. I'll denote this pooled vector as R.

```
names = unlist(lapply(colnames(df),
       function(x){
         rep(x, times = length(df[, x]))
       }))
pooled = data.frame(
  x = as.numeric(unlist(df)),
  name = names,
  rnk = rank(as.numeric(unlist(df)))
)
head(pooled)
```

```
##    x name rnk
## 1 21    A  18
## 2 23    A  22
## 3 13    A   3
## 4 19    A  12
## 5 13    A   3
## 6 19    A  12
```

2.) We then rank all of the observations, and calculate the mean of all the ranks, and denote it $\bar{R}$.

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i$$

Or, as `R` code:

```
R_bar = mean(pooled$rnk)
R_bar
```

```
## [1] 18.5
```

3.) Then, for each group $\omega \in \{A, B, C, \dots\}$ we calculate the *mean rank of that group.* This is because, if H_0 is true, the group-wise mean rank should be approximately equal to the grand mean rank for each group. In other words, we would expect:

$$H_0 \text{ is true.} \implies \forall i \in [1, k], \bar{R}_i \approx \bar{R}$$

So then our next step is to calculate the group-wise mean ranks $\bar{R}_i$ for treatments $1, \dots k$.

$$\bar{R}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} R_{i,j}$$

Where $R_{i,j}$ is the j-th observation in group i.

```
R_i = sapply(colnames(df),
             function(x){
               mean(pooled$rnk[pooled$name == x])
           })
round(R_i, 3)
```

```
##      A      B      C      D
## 12.389 30.611 22.500  8.500
```

4.) Then, we have the following test statistic.

$$H = \frac{12 \sum_{i=1}^{k} n_i (\bar{R}_i - \bar{R})^2}{N(N+1)} \sim \chi^2_{k-1}$$

In terms of ANOVA, think of this as "sum of squares rank."

5.) Now, we can calculate H_{obs} from our data.

```
N = length(unlist(df)); k = length(colnames(df))
# math for H looks a bit complex, but follows the formula
H = ( 12 * sum(as.numeric(sapply(colnames(df),
       function(x){ length(df[, x])* ( (R_i[x] - R_bar)^2 ) }))) /
        (N*(N+1)))
round(H, 3)
```

```
## [1] 24.326
```

6.) Now, we find the p-value as $P(H_{\text{obs}} > \chi^2_{k-1})$, where, here, $k = 4$.

```
pchisq(H, k - 1, lower.tail = FALSE)
```

```
## [1] 2.135241e-05
```

Built-In Functions

Due to internal rounding errors, our values of p and H_{obs} might be a bit different from the built-in R function. For completeness, we include the whole process here. It's best to go through the logic once on your own (and use the built-ins after that!)

```
kruskal.test(x ~ name, data = pooled)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  x by name
## Kruskal-Wallis chi-squared = 24.484, df = 3, p-value = 1.979e-05
```

Bonus: A Return to the In-Class Data

Combining all the R Code into one process…

```
df <- data.frame(
  AL = c(16, 10, 20, 29, -14),
  AR = c(51, 58, 52, 47, 32),
  PL = c(58, 12, 62, 43, 26)
)

names = unlist(lapply(colnames(df),
      function(x){
        rep(x, times = length(df[, x]))
      }))
pooled = data.frame(
  x = as.numeric(unlist(df)),
  name = names,
  rnk = rank(as.numeric(unlist(df)))
)

R_bar = mean(pooled$rnk)
print(paste("Mean Rank Overall:", R_bar))
```

```
## [1] "Mean Rank Overall: 8"
```

```
R_i = sapply(colnames(df),
            function(x){
              mean(pooled$rnk[pooled$name == x])
          })
print(R_i)
```

```
##   AL   AR   PL
##  3.8 10.9  9.3
```

```
N = length(unlist(df)); k = length(colnames(df))
# math for H looks a bit complex, but follows the formula
H = ( 12 * sum(as.numeric(sapply(colnames(df),
      function(x){ length(df[, x])* ( (R_i[x] - R_bar)^2 ) })))) /
        (N*(N+1)))
print(paste("Oberved Test Statistic (H Value):", round(H)))
```

```
## [1] "Oberved Test Statistic (H Value): 7"
```

```
pchisq(H, k - 1, lower.tail = FALSE)
```

```
## [1] 0.03119492
```