Kruskal-Wallis Test

This lab will reinforce concepts regarding the Kruskal-Wallis test. You can think of the Kruskal-Wallis test as the K-sample version of the rank-sum test. Or you can think of the Kruskal-Wallis test as the equivalent of the F-test for ranks.

Movie ratings

The file moviesall.txt contains various information on movies released in the year 2003. Altough it is not really a random sample of movies, we will treat it as a random sample of movies for the purpose of this lab. You may think of it as representative of movies in general in the past decade.

For each movie, we have it's rating (G, PG, PG-13, R), it's genre, it's box-office gross (in millions of dollars), it's run time (in minutes), and it's score on rottentomatoes.com (higher scores mean better movies). For this lab, you will investigate differences in run time between the movie ratings.

Kruskal-Wallis test

The Kruskal-Wallis test, like the permutation F-test, tests for differences between two or more groups. However, unlike the F-test, the Kruskal-Wallis test works with the ranks of the observations rather than the actual observations themselves. In this way, it is similar to the Wilcoxon rank-sum test from Chapter 2.

The hypotheses for a Kruskal-Wallis test are the same as for a permutation F-test:

$H_0 : F_1(x) = F_2(x) = \cdots = F_K(x)$

$H_1 : F_i(x) \leq F_j(x)$ or $F_i(x) \geq F_j(x)$ for at least one pair (i, j), with strict inequality for at least one x.

Kruskal-Wallis statistic

Your book definites the Kruskal-Wallis statistic as

$$KW = \frac{12}{N(N+1)} \sum_{i=1}^{K} n_i \left(\overline{R}_i - \frac{N+1}{2} \right)^2$$

Here $\overline{R}_i$ is the mean rank for group i, n_i is the number of observations in group i, K is the number of groups, and $N = n_1 + \ldots + n_K$ is the total number of observations.

Consider the following subset of the entire movie dataset. Below are the runtimes and ratings for 12 movies.

rating	runtime
PG-13	92
PG-13	123
PG-13	114
G	72
PG-13	117
R	111
R	129
PG	113
R	102
PG	98
PG	94
R	96

1. Compute the Kruskal-Wallis statistic for this subset of the movie data. You might find it helpful to summarize the data in the following table:

group	ranks	sample size	mean rank
G			
PG			
PG-13			
R			

An alternative way to define the Kruskal-Wallis statistic is

$$\text{KW} = \frac{6}{N} \sum_{i=1}^{K} \frac{(\text{obsRS}_i - \text{expRS}_i)^2}{\text{expRS}_i},$$

where obsRS_i is the observed rank-sum in group i and expRS_i is the expected rank sum in group i. The expected rank-sum is the rank-sum you would expect to see if the null hypothesis were true.

2. Compute the Kruskal-Wallis statistic for this subset of the movie data using this new definition of the statistic. Verify that these two methods give you the same answer. You might find it helpful to summarize the data in the following table:

group	ranks	observed rank-sum	expected rank-sum
G			
PG			
PG-13			
R			

The Kruskal-Wallis statistic with ties

When there are ties in our data, we have to make an adjustment to the Kruskal-Wallis statistic. First, we adjust the ranks by assigning the average rank of the tied observations to the tied observations, just was we did for the Wilcoxon rank-sum test. The second adjustment we make is in the statistic itself:

$$KW_{ties} = \frac{KW}{1 - \frac{\sum_{j=1}^{g}(t_j^3 - t_j)}{N^3 - N}}$$

Here KW is the originally definied Kruskal-Wallis statistic applied to the adjusted ranks and t_j is the number of tied observations for each set of ties.

Consider a different subset of the movie data, as shown below.

rating	runtime
PG-13	95
R	97
G	100
PG	95
PG-13	96
PG-13	101
R	102
PG-13	100
PG-13	141
PG	87
R	154
PG-13	107

In this case, we have two sets of ties, since there are two 95's and two 100's. So we have $t_1 = 2$ and $t_2 = 2$.

3. Compute the Kruskal-Wallis statistic for this subset of the movie data. Do this by first finding the adjusted ranks, then computing the original statistic for these adjusted ranks, and then adjusting the statistic for ties with the formula above.

A Kruskal-Wallis permutation test

We can find a p-value for the Kruskal-Wallis test by performing a permutation test. You will be permforming a Kruskal-Wallis test on the entire movies data set now.

The entire movies data set has multiple tied observations that we will have to account for in our calculation of the KW statistic.

4. Run the following command, which tallies up how many observations there are at each movie run time.

   What values of t_j are of note here? [Hint, you will need to examine t_1 through t_{34}.]

   ```
   table(runtime)
   ```

We we will primarily focus on random sampling the permutations when performing a Kruskal-Wallis test, since the number of possible group assignments is quite large. In this way, we are computing an approximate p-value instead of an exact one. Consider the following code. Be sure you understand how the KW statistic is calculated.

```
### calculate the observed KW statistic
t.j = c(2,6,3,3,2,3,5,2,3,4,5,3,3,5,4,5,4,3,6,2,3,4,3,3,3,4,5,3,2,3,2,2,2,3)
n.i = c(4, 21, 65, 50); N = sum(n.i)
```

```
ranks = rank(runtime)   ### rank the data
R.i = c(mean(ranks[rating=="G"]), mean(ranks[rating=="PG"]), mean(ranks[rating=="PG-13"]),
                                                                    mean(ranks[rating=="R"]))
KW.noties = 12/(N*(N+1)) * sum( n.i*(R.i - (N+1)/2)^2 )
teststat.obs = KW.noties/( 1 - sum( t.j^3 - t.j )/(N^3 - N) )

teststat = rep(NA, 1000)
for(i in 1:1000) {

### randomly "shuffle" the rating labels for the movies
ratingSHUFFLE = sample(rating)

### compute the KW statistic for the shuffled data
R.i = c(mean(ranks[ratingSHUFFLE=="G"]), mean(ranks[ratingSHUFFLE=="PG"]),
                        mean(ranks[ratingSHUFFLE=="PG-13"]), mean(ranks[ratingSHUFFLE=="R"]))
KW.noties = 12/(N*(N+1)) * sum( n.i*(R.i - (N+1)/2)^2 )
teststat[i] = KW.noties/( 1 - sum( t.j^3 - t.j )/(N^3 - N) )

}
### calculate the approximate p-value
sum(teststat >= teststat.obs)/1000
```

5. Explain why we calculate the p-value by sum(teststat >= teststat.obs)/1000. Will this always be the case, or are there circumstances where this might change?

6. Run the entire chunk of code to calculate a p-value. Report your observed test statistic and your p-value. Explain anything odd you encounter here and why it is possible. Explain how this contrasts to the results for an exact test (where you examine all the permutations, not just a sample of them).

Kruskal-Wallis with R

The function kruskal.test() will perform a Kruskal-Wallis test in R. This function does not use permutations to compute a p-value – instead it uses an approximation based on the chi-square distribution.

7. Perform a Kruskal-Wallis test of the run time data using the command below. Report your test statistic and your p-value. Verify that this matches up with your results in #6.

   ```
   kruskal.test(runtime ~ rating)
   ```

Apply what you have learned here

Please write clearly and in complete sentences. Include any R code at the end of your write-up, after you have written up your solution.

1. Perform a Kruskal-Wallis test to determine whether the distribution of scores on rottentomatoes.com is the same for movies with different ratings. Code and use the full Kruskal-Wallis permutation test similar to the code preceeding question 5. State your hypotheses and report you test statistic and p-value. State a conclusion relating to the context of the question of interest.

2. Perform a Kruskal-Wallis test to determine whether box office gross is the same for movies with different ratings. You may use either the full permutation code or the kruskal.test function. State your hypotheses and report you test statistic and p-value. State a conclusion relating to the context of the question of interest.