STAT 3480 Lab6

Shaoran Sun

March 29, 2016

1

group	ranks	sample size	mean rank
G	1	1	1
PG	3, 5, 8	3	5.333333
PG-13	2, 9, 10, 11	4	8
R	4, 6, 7, 12	4	7.25

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

=$\frac{12}{12(12+1)} \sum_{i=1}^{K} n_i(\bar{R}_i - \frac{12+1}{2})$

=3.50641

2

group	ranks	observed rank-sum	expected rank-sum
G	1	1	6.5
PG	3, 5, 8	16	19.5
PG-13	2, 9, 10, 11	32	26
R	4, 6, 7, 12	29	26

KW = $\frac{6}{N} \sum_{i=1}^{K} \frac{(obsRS_i - expRS_i)^2}{expRS_i}$

=$\frac{6}{12} \sum_{i=1}^{K} \frac{(obsRS_i - expRS_i)^2}{expRS_i}$

=3.50641

Two method computed the same answer of $KW = 3.51$.

3

group	ranks	sample size	mean rank
G	6.5	1	6.5
PG	1,2.5	2	1.166667
PG-13	2.5, 4, 6.5, 8, 10, 11	6	7
R	5, 9, 12	3	8.666667

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

=$\frac{12}{12(12+1)} \sum_{i=1}^{K} n_i(\bar{R}_i - \frac{12+1}{2})$

=4.669

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

=$\frac{4.669}{1 - \frac{\sum_{j=1}^{g}(t_j^3 - t_j)}{12^3 - 12}}$

=4.702

4

t_j are the counts of each duplicate element in the array. In total, there are 34 of them, in another words, there are 34 duplicate elements.

5

We are calculating the p-value by sum(teststat >= teststat.obs)/1000 because we want to see if out of 1000 random shuffles, which ones have higher teststat than the observed teststat.

The number 1000 might change if we are looking for more or less runs of random shuffling.

The greater or equal sign might change if other relations may cause being more extreme, such as less than, or not equal.

6

After running the code, observed teststat is 19.671, p-value is 0. This is less than 0.05. Hence we reject null hypothesis at 0.05 siginificance level, and conclude that 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.

This is different with our exact test of result 0.008. Though both of the tests reject the null hypothesis, the sampling method is looking at $\frac{1}{3}$ of all the data.

7

The test statistic from the command is 19.671, with p-value of 0.000199. This is less than 0.05. Hence we reject null hypothesis at 0.05 siginificance level, and conclude that 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.

This result match up with the one in #6.

1

The hypotheses for a Kruskal-Wallis test are the same as for a permutation F -test: $H_0 : F_1(x) = F_2(x) =$ ... $= F_K(x)$ $H_1 : F_i(x) \leq F_j(x)$ or $F_i(x) geq F_j(x)$ for at least one pair (i, j), with strict inequality for at least one

The test statistic from the following command is 2.220, with p-value of 0.0508. This is greater than 0.05. Hence we fail to reject null hypothesis at 0.05 siginificance level, and conclude that there is not enough eveidence to show that 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.

Under the context of the problem, the scores and the ratings are not related.

```r
moviesall <- read.delim("~/Desktop/moviesall.txt")
attach(moviesall)
data1 <- table(score)
data1
```

```
## score
## 29.2 29.3 29.5 29.8 30.4 31.1 32.6 33.2 34.8   35 35.1 36.2 36.3 37.8 37.9
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 38.1 38.5 39.5 40.5 40.8 41.5   42 42.2 42.5 42.9 43.2 43.6 43.7 44.2 44.8
##    1    1    1    1    1    1    1    1    1    2    1    1    1    1    3
## 45.3 45.4 45.8 46.1 46.2 46.9 47.3 47.9 48.1 48.3 48.8 48.9 49.7 49.9 50.2
##    1    1    1    1    2    1    2    1    1    1    2    1    1    1    1
## 50.3 50.6 50.7 50.8 51.3 52.1 52.6 52.9 53.1 53.3 53.6   54 54.2 54.3 54.7
##    1    1    1    1    1    1    1    1    2    1    2    1    1    1    1
## 54.8 55.2 55.3 55.4 55.6 55.8 56.4 56.7 57.1 57.2 57.7 57.9 58.7 59.3 59.6
##    1    1    1    1    2    1    1    1    1    1    1    3    1    1    1
##   60 60.3 60.4 60.6   61 61.8 62.1 62.2 62.6 63.1 63.3 63.6 63.7 64.6 64.9
##    1    1    1    1    1    1    2    1    2    1    3    1    1    1    1
##   65 65.4 65.6 65.8 66.5   67 67.1 67.3 67.5 67.8 68.9 69.5 70.1 70.3 70.5
##    1    1    1    1    1    1    1    1    1    2    1    1    1    1    1
## 71.3 71.6 71.9 72.5 74.9 75.5 75.8   78 78.2 79.9 81.3 84.1 86.4 87.6 89.9
##    1    1    1    1    1    1    1    2    1    1    1    1    1    1    1
## 91.2 92.2 94.6
##    1    1    1
```

```
### calculate the observed KW statistic
t.j = data1[data1>1]
n.i = c(4, 21, 65, 50); N = sum(n.i)
ranks = rank(score) ### 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)/10000
```

```
## [1] 0.0543
```

```
teststat.obs
```

```
## [1] 2.220445
```

2

The hypotheses for a Kruskal-Wallis test are the same as for a permutation F -test: $H_0 : F_1(x) = F_2(x) = ... = F_K(x)$ $H_1 : F_i(x) \leq F_j(x)$ or $F_i(x) geq F_j(x)$ for at least one pair (i, j), with strict inequality for at least one

The test statistic from the following command is 6.822, with p-value of 0.07781. This is greater than 0.05. Hence we fail to reject null hypothesis at 0.05 siginificance level, and conclude that there is not enough eveidence to show that $F_i(\text{x}) \leq F_j\ (\text{x})$ or $F_i(\text{x}) \geq F_j\ (\text{x})$ for at least one pair (i, j), with strict inequality for at least one x.

Under the context of the problem, the box office gross and ratings are not related.

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

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gross by rating
## Kruskal-Wallis chi-squared = 6.8215, df = 3, p-value = 0.07781
```

Appendix

1

```
(3+5+8)/3
```

```
## [1] 5.333333
```

```
(2+9+10+11)/4
```

```
## [1] 8
```

```
(4+6+7+12)/4
```

```
## [1] 7.25
```

```
12/(12*13)*(1*(1-6.5)^2+3*(5.333333-6.5)^2+4*(8-6.5)^2+4*(7.25-6.5)^2)
```

```
## [1] 3.50641
```

2

```
6/12*((1-6.5)^2/6.5+(16-19.5)^2/19.5+(32-26)^2/26+(29-26)^2/26)
```

```
## [1] 3.50641
```

3

```
3.5/3
```

```
## [1] 1.166667
```

```
(2.5+ 4+ 6.5+ 8+ 10+ 11)/6
```

```
## [1] 7
```

```
(5+9+12)/3
```

```
## [1] 8.666667
```

```
12/(12*13)*(1*(6.5-6.5)^2+2*(1.166667-6.5)^2+6*(7-6.5)^2+3*(8.666667-6.5)^2)
```

```
## [1] 5.574786
```

4

```
moviesall <- read.delim("~/Desktop/moviesall.txt")
attach(moviesall)
```

```
## The following objects are masked from moviesall (pos = 3):
##
##     genre, gross, rating, runtime, score
```

```
table(runtime)
```

```
## runtime
##  72  75  81  82  84  85  86  87  88  89  90  91  92  93  94  95  96  97
##   1   1   2   1   6   1   3   3   2   3   5   2   3   1   4   5   3   3
##  98  99 100 101 102 103 104 105 106 107 108 109 110 111 113 114 115 116
##   5   1   4   5   4   1   3   6   2   1   3   4   3   3   3   1   1   4
## 117 118 119 121 123 125 127 128 129 130 133 135 136 137 138 139 141 143
##   5   3   2   3   1   1   2   1   1   1   1   2   1   2   3   1   1   1
## 147 152 154 201 231
##   1   1   1   1   1
```

```
### 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[ra
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=="P
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
```

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

```r
teststat.obs
```

```
## [1] 19.67098
```

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

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  runtime by rating
## Kruskal-Wallis chi-squared = 19.671, df = 3, p-value = 0.0001986
```