Lab 6

Frank Woodling

March 29, 2016

1.

```
list <- c(92, 123, 114, 72, 117, 111, 129, 113, 102, 98, 94, 96)
names(list) <- c("PG-13", "PG-13", "PG-13", "G", "PG-13", "R", "R", "PG","R","PG","PG", "R")

list.sorted <- sort(list)
rbind(list.sorted, 1:12)
```

```
##              G PG-13 PG  R PG   R   R  PG PG-13 PG-13 PG-13   R
## list.sorted 72    92 94 96 98 102 111 113   114   117   123 129
##              1     2  3  4  5   6   7   8     9    10    11  12
```

```
g.ranks <- c(1)
pg.ranks <- c(3, 5, 8)
pg13.ranks <- c(2, 9, 10, 11)
r.ranks <- c(4, 6, 7, 12)

# group          ranks         sample size         mean rank
#
# G               12                1                 12
#
# PG            3, 5, 8             3                 5.33
#
# PG-13      2, 9, 10, 11           4                 8
#
# R          4, 6, 7, 12            4                 7.25

kw.1 <- 1*(12-13/2)^2
kw.2 <- 3*(5.33-13/2)^2
kw.3 <- 4*(8-13/2)^2
kw.4 <- 4*(7.25-13/2)^2

kw <- (12/(12*13))*(kw.1+kw.2+kw.3+kw.4)
kw
```

```
## [1] 3.508208
```

The Kruskal-Wallis statistic is 3.50641.

2.

```r
expsum.g <- sum(1:12)*1/12
expsum.pg <- sum(1:12)*3/12
expsum.pg13 <- sum(1:12)*4/12
expsum.r <- sum(1:12)*4/12

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

kw2.1 <- ((12-6.5)^2)/6.5
kw2.2 <- ((16-19.5)^2)/19.5
kw2.3 <- ((32-26)^2)/26
kw2.4 <- ((29-26)^2)/26

kw2 <- 6/12*(kw2.1 + kw2.2 + kw2.3 + kw2.4)
kw2
```

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

The Kruskal-Wallis statistic is 3.50641 in this test as well.

3.

```r
list.2 <- c(95, 97, 100, 95, 96, 101, 102, 100, 141, 87, 154, 107)
names(list.2) <- c("PG-13", "R", "G", "PG", "PG-13", "PG-13", "R", "PG-13", "PG-13", "PG", "R", "PG-13"

list.sorted2 <- sort(list.2)
rbind(list.sorted2, 1:12)
```

```
##              PG PG-13 PG PG-13  R   G PG-13 PG-13   R PG-13 PG-13   R
## list.sorted2 87    95 95    96 97 100   100   101 102   107   141 154
##               1     2  3     4  5   6     7     8   9    10    11  12
```

```r
# group     ranks                       sample size         mean rank
#
# G         6.5                            1                   6.5
#
# PG        1, 2.5                         2                   1.75
#
# PG-13     2.5, 4, 6.5, 8, 10, 11         6                   7
#
# R         5, 9, 12                       3                   8.666667
```

```
kw3.1 <- 1*(6.5-13/2)^2
kw3.2 <- 2*(1.75-13/2)^2
kw3.3 <- 6*(7-13/2)^2
kw3.4 <- 3*(8.66667-13/2)^2
kw3 <- (12/(12*13))*(kw3.1+kw3.2+kw3.3+kw3.4)
kw3
```

```
## [1] 4.669875
```

```
# there are 2 ties, 95 and 100 with two a piece
kwties.denom <- 1-(2*(2^3-2))/(12^2-12)
kw3.ties <- kw3/kwties.denom
kw3.ties
```

```
## [1] 5.136863
```

4.

```
movies <- read.table("moviesall.txt", header=T)
attach(movies)
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[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
```

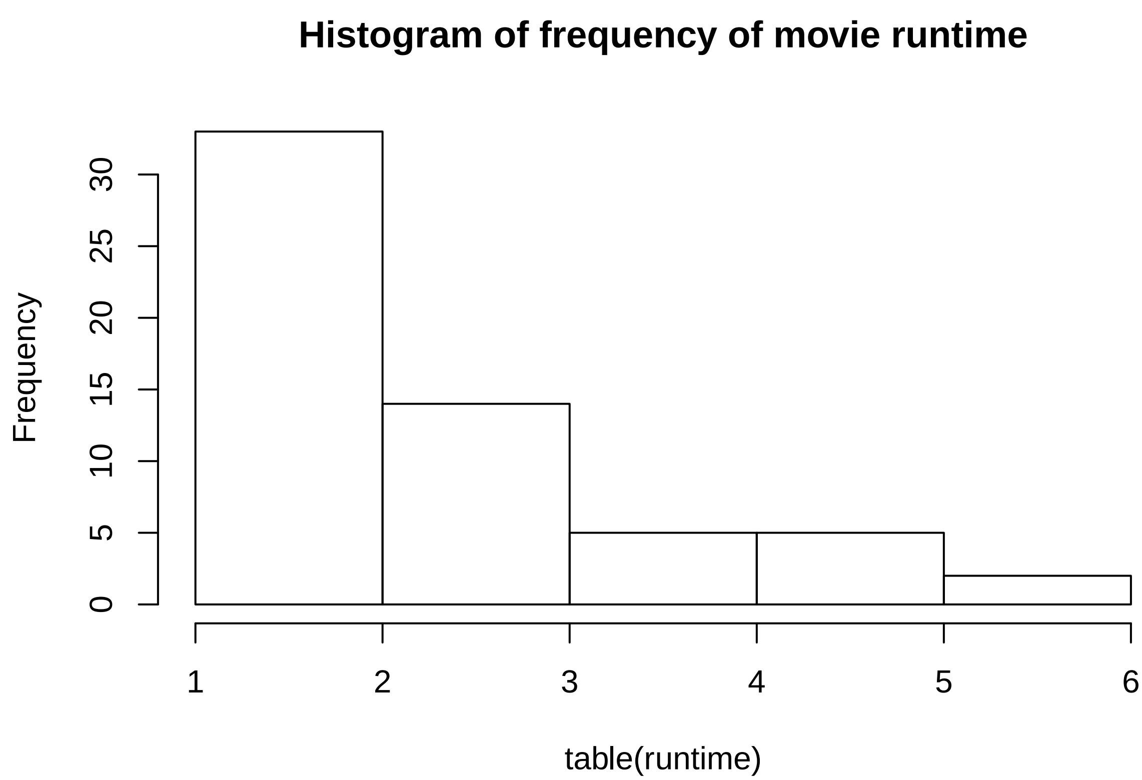
```r
R.i = c(mean(ranks[ratingSHUFFLE=="G"]), mean(ranks[ratingSHUFFLE=="PG"]),
        mean(ranks[ratingSHUFFLE=="PG-13"]), mean(ranks[ratingSHUFFLE=="R"]))
KW.noties = 12/(N*(N+1)) * sum( n.i*(R.i - (N+1)/2)^2 )
teststat[i] = KW.noties/( 1 - sum( t.j^3 - t.j )/(N^3 - N) )
}

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

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

```r
teststat.obs
```

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

```r
hist(table(runtime), main="Histogram of frequency of movie runtime")
```



Looking at the table and histogram we can see that 6 is the maximum number of tied observations on 84 minutes and 105 minutes. It seems like the longer a movie is the less chance of a tie, and most of the ties are for lower runtimes.

5.

6.

```r
sum(teststat >= teststat.obs)/1000
```

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

```r
teststat.obs
```

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

The p-value is 0 and the observed test statistic is 19.67098. I seem to get a different number each time I try this. I also got 0.001 and 6700 for the test statistic.

7.

```r
kruskal.test(runtime~rating, data = movies)
```

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

This value lines up with the lines of code in part 4.

Apply what you have learned here

1.

```r
movies <- read.table("moviesall.txt", header=T)
attach(movies)
```

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

```r
### 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(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)/1000
```

```
## [1] 0.519
```

```
teststat.obs
```

```
## [1] 2.221713
```

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

```
##
##  Kruskal-Wallis rank sum test
##
## data:  score by rating
## Kruskal-Wallis chi-squared = 2.2204, df = 3, p-value = 0.5279
```

We fail to reject the null hypothesis with a p-value of 0.541 (the built in test says 0.5279). We can conclude that there is a difference in means between the scores given for a movie and the rating (G, PG, PG-13, R).

2.

```
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
```

The null hypothesis is that there is not a difference of means between the different ratings and their box office gross. In this case we fail to reject the null hypothesis with a p-value of 0.07781. We can conclude that the is not a significant difference between the mean box office gross throughout the different ratings.

Test

```r
# 1
tlist <- c(92, 123, 114, 72, 117, 111, 129, 113, 102, 98, 94, 96)
tnames <- c("PG-13", "PG-13", "PG-13", "G", "PG-13", "R", "R", "PG","R","PG","PG", "R")
tgroups <- c(3,3,3,1,3,4,4,2,4,2,2,4)
kruskal.test(tlist, tgroups)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  tlist and tgroups
## Kruskal-Wallis chi-squared = 3.5064, df = 3, p-value = 0.3199
```

```r
# 3
tlist.3 <- c(95, 97, 100, 95, 96, 101, 102, 100, 141, 87, 154, 107)
tgroups.3 <- c(3, 4, 1, 2, 3, 3, 4, 3, 3, 2, 4, 3)
kruskal.test(tlist.3, tgroups.3)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  tlist.3 and tgroups.3
## Kruskal-Wallis chi-squared = 4.7028, df = 3, p-value = 0.1949
```