Lab 7

Frank Woodling

March 29, 2016

1.

```
movies <- read.table('moviesall.txt', header = T)
choose(4,2)
```

```
## [1] 6
```

We have to consider $\binom{K}{2}$. With groups of G, PG, PG-13, R we have 4 ratings so $\binom{4}{2}$ means we have 6 groups.

2.

```
data = rnorm(100, 10, 5)
t.test(data, mu = 10, alt = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  data
## t = 2.0726, df = 99, p-value = 0.0408
## alternative hypothesis: true mean is not equal to 10
## 95 percent confidence interval:
##  10.04578 12.10070
## sample estimates:
## mean of x
##  11.07324
```

The p-value is equal to 0.1218 so we fail to reject the null hypothesis at a significance level of 0.05. We can conclude that the mean is equal to 10.

3.

```
pval = rep(NA, 20)
for(i in 1:20)
{
  data = rnorm(100, 10, 5)
  pval[i] <- t.test(data, mu = 10, alt = "two.sided")$p.value
}
# pval
```

```
# sum(pval < .05)
# sum(pval < .05)/20

#  [1] 0.98098868 0.63974810 0.13797469 0.27056011 0.65848932 0.34852383
#  [7] 0.07258321 0.98215747 0.45292255 0.83558520 0.90717642 0.88130606
# [13] 0.45448398 0.21235961 0.02345634 0.85325499 0.48692802 0.17897105
# [19] 0.31703612 0.90003461
```

I was able to reject it 1 time when I ran it. One time out of 20 is a proportion of 0.05. The lab instructions say we should reject it 1 out of 20 which is exactly what we did.

4.

```
pval = rep(NA, 1000)
for(i in 1:1000) {
data = rnorm(100, 10, 5)
pval[i] = t.test(data, mu = 10, alternative = "two.sided")$p.value
}
sum(pval < .05)
```

```
## [1] 56
```

```
sum(pval < .05)/1000
```

```
## [1] 0.056
```

We rejected the null hypothesis 40 out of the 1000 times ran. The p-value is 0.04.

5.

With 6 combinations we should use a significance level of: $\frac{0.05}{6} = 0.008333333$.

6.

We should multiply the original p-values by 6.

7.

```
movies <- read.table('moviesall.txt', header = T)
attach(movies)
movies.pair = movies[rating=="G" | rating=="PG",]
detach(movies)
attach(movies.pair)
teststat.obs = mean(runtime[rating == "G"]) - mean(runtime[rating == "PG"])
teststat.obs
```

```
## [1] -12.58333
```

```
m = length(runtime[rating == "G"])
n = length(runtime[rating == "PG"])
teststat = rep(NA, 1000)

for(i in 1:1000)
{
  ### randomly "shuffle" the elements of the rainfall vector
  runtimeSHUFFLE = sample(runtime)

  ### assign the first m to the first group
  ### and the next n to the other group
  G = runtimeSHUFFLE[1:m]
  PG = runtimeSHUFFLE[(m+1):(m+n)]

  ### compute the test stat for the shuffled data
  teststat[i] = mean(G) - mean(PG)
}
### calculate the approximate p-value
pval = sum(teststat <= teststat.obs)/1000 + sum(teststat >= -teststat.obs)/1000
adjusted.pval = 0.044*6
```

The original p-value is 0.044 and after Bonferroni adjustment we have a p-value of 0.264. We fail to reject the null hypothesis, and conclude that there is not a difference in means between movies runtimes that are rated G and PG.

8.

```
movies <- read.table('moviesall.txt', header = T)
attach(movies)
```

```
## The following objects are masked from movies.pair:
##
##     genre, gross, rating, runtime, score
```

```
movies.pair = movies[rating=="G" | rating=="R",]
detach(movies)
attach(movies.pair)
```

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

```
teststat.obs = mean(runtime[rating == "G"]) - mean(runtime[rating == "R"])
teststat.obs
```

```
## [1] -27.85
```

```
m = length(runtime[rating == "G"])
n = length(runtime[rating == "R"])
teststat = rep(NA, 1000)

for(i in 1:1000)
{
  ### randomly "shuffle" the elements of the rainfall vector
  runtimeSHUFFLE = sample(runtime)

  ### assign the first m to the first group
  ### and the next n to the other group
  G = runtimeSHUFFLE[1:m]
  R = runtimeSHUFFLE[(m+1):(m+n)]

  ### compute the test stat for the shuffled data
  teststat[i] = mean(G) - mean(R)
}
### calculate the approximate p-value
pval = sum(teststat <= teststat.obs)/1000 + sum(teststat >= -teststat.obs)/1000
adjusted.pval = 0.006*6
```

The original p-value is 0.006 and after Bonferroni adjustment we have a p-value of 0.036. We can reject the null hypothesis, and conclude that there is a difference in means between movies runtimes that are rated G and R.

9.

```
movies <- read.table('moviesall.txt', header = T)
attach(movies)
```

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

## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score
```

```
movies.pair = movies[rating=="PG" | rating=="PG-13",]
detach(movies)
attach(movies.pair)
```

```
## The following objects are masked from movies.pair (pos = 3):
##
##     genre, gross, rating, runtime, score
##
## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score
```

```r
teststat.obs = mean(runtime[rating == "PG"]) - mean(runtime[rating == "PG-13"])
teststat.obs
```

```
## [1] -15.35897
```

```r
m = length(runtime[rating == "PG"])
n = length(runtime[rating == "PG-13"])
teststat = rep(NA, 1000)

for(i in 1:1000)
{
  ### randomly "shuffle" the elements of the rainfall vector
  runtimeSHUFFLE = sample(runtime)

  ### assign the first m to the first group
  ### and the next n to the other group
  G = runtimeSHUFFLE[1:m]
  PG13 = runtimeSHUFFLE[(m+1):(m+n)]

  ### compute the test stat for the shuffled data
  teststat[i] = mean(G) - mean(PG13)
}

### calculate the approximate p-value
pval = sum(teststat <= teststat.obs)/1000 + sum(teststat >= -teststat.obs)/1000
adjusted.pval = 0.65*6
```

The original p-value is 0.065 and after Bonferroni adjustment we have a p-value of 3.9. We fail to reject the null hypothesis, and conclude that there is not a difference in means between movies runtimes that are rated G and PG-13.

10.

```r
movies <- read.table('moviesall.txt', header = T)
attach(movies)
```

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

## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 5):
##
##     genre, gross, rating, runtime, score
```

```
movies.pair = movies[rating=="PG" | rating=="R",]
detach(movies)
attach(movies.pair)
```

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

## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 5):
##
##     genre, gross, rating, runtime, score
```

```
teststat.obs = mean(runtime[rating == "PG"]) - mean(runtime[rating == "R"])
teststat.obs
```

```
## [1] -15.26667
```

```
m = length(runtime[rating == "PG"])
n = length(runtime[rating == "R"])
teststat = rep(NA, 1000)

for(i in 1:1000)
{
  ### randomly "shuffle" the elements of the rainfall vector
  runtimeSHUFFLE = sample(runtime)

  ### assign the first m to the first group
  ### and the next n to the other group
  PG = runtimeSHUFFLE[1:m]
  R = runtimeSHUFFLE[(m+1):(m+n)]

  ### compute the test stat for the shuffled data
  teststat[i] = mean(PG) - mean(R)
}

### calculate the approximate p-value
pval = sum(teststat <= teststat.obs)/1000 + sum(teststat >= -teststat.obs)/1000
adjusted.pval = 0.01*6
```

The original p-value is 0.001 and after Bonferroni adjustment we have a p-value of 0.06. We can reject the null hypothesis, and conclude that there is a difference in means between movies runtimes that are rated PG and R.

11.

```r
movies <- read.table('moviesall.txt', header = T)
attach(movies)
```

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

## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 5):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 6):
##
##     genre, gross, rating, runtime, score
```

```r
movies.pair = movies[rating=="PG-13" | rating=="R",]
detach(movies)
attach(movies.pair)
```

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

## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 5):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 6):
##
##     genre, gross, rating, runtime, score
```

```r
teststat.obs = mean(runtime[rating == "PG-13"]) - mean(runtime[rating == "R"])
teststat.obs
```

```
## [1] 0.09230769
```

```r
m = length(runtime[rating == "PG-13"])
n = length(runtime[rating == "R"])
teststat = rep(NA, 1000)

for(i in 1:1000)
```

```
{
  ### randomly "shuffle" the elements of the rainfall vector
  runtimeSHUFFLE = sample(runtime)

  ### assign the first m to the first group
  ### and the next n to the other group
  PG13 = runtimeSHUFFLE[1:m]
  R = runtimeSHUFFLE[(m+1):(m+n)]

  ### compute the test stat for the shuffled data
  teststat[i] = mean(PG13) - mean(R)
}

### calculate the approximate p-value
pval = sum(teststat <= teststat.obs)/1000 + sum(teststat >= -teststat.obs)/1000
adjusted.pval = 1.018*6
```

The original p-value is 1.017 and after Bonferroni adjustment we have a p-value of 6.108. We fail to reject the null hypothesis, and conclude that there is not a difference in means between movies runtimes that are rated PG-13 and R.

12.

We were able to reject the null hypothesis on the test that was between PG and R, and also the test between G and R. It appears that there is a greater difference in mean runtimes as the ratings get further apart, or that higher rated movies are either substantially longer or shorter than lower rated movies.

Lab Summary

1.

```
movies <- read.table('moviesall.txt', header = T)
attach(movies)
```

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

## The following objects are masked from movies.pair (pos = 4):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 5):
##
##     genre, gross, rating, runtime, score
```

```
## The following objects are masked from movies.pair (pos = 6):
##
##     genre, gross, rating, runtime, score

## The following objects are masked from movies.pair (pos = 7):
##
##     genre, gross, rating, runtime, score
```

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

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

2.

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

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

```
movies.pair = movies[rating=="G" | rating=="PG",]
wilcox.test(gross~rating, data = movies.pair)
```

```
##
##  Wilcoxon rank sum test
##
## data:  gross by rating
## W = 49, p-value = 0.6422
## alternative hypothesis: true location shift is not equal to 0
```

```
movies.pair2 = movies[rating=="G" | rating=="PG-13",]
wilcox.test(gross~rating, data = movies.pair2)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gross by rating
## W = 165, p-value = 0.3757
## alternative hypothesis: true location shift is not equal to 0
```

```r
movies.pair3 = movies[rating=="G" | rating=="R",]
wilcox.test(gross ~ rating, data = movies.pair3)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gross by rating
## W = 144, p-value = 0.1508
## alternative hypothesis: true location shift is not equal to 0
```

```r
movies.pair4 = movies[rating=="PG" | rating=="PG-13",]
wilcox.test(gross ~ rating, data = movies.pair4)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gross by rating
## W = 790, p-value = 0.2821
## alternative hypothesis: true location shift is not equal to 0
```

```r
movies.pair5 = movies[rating=="PG-13" | rating=="R",]
wilcox.test(gross ~ rating, data = movies.pair5)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gross by rating
## W = 1939, p-value = 0.07694
## alternative hypothesis: true location shift is not equal to 0
```

```r
movies.pair6 = movies[rating=="PG" | rating=="R",]
wilcox.test(gross ~ rating, data = movies.pair6)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gross by rating
## W = 692, p-value = 0.03593
## alternative hypothesis: true location shift is not equal to 0
```