Multiple Testing

Matt Allen

08/01/2018

Create a Word document from this R Markdown file for the following exercises. Submit the R markdown file and the knitted Word document via D2L Dropbox.

## Exercise 1

For this exercise we’re going to follow along with the first part of the presentation and compute the error rates for three types of multiple testing correction. Like the presentation, in this example there are 1000 tests and we know when the null and alternative hypotheses are true. For each test we have: - value is from a normal distribution with - value is from a normal distribution with

Here is the data and p-values:

# Do not change this chunk  
set.seed(124)  
T0 = 900;  
T1 = 100;  
x = c( rnorm(T0), rnorm(T1, mean = 3))  
P = pnorm(x, lower.tail = FALSE )

For the first 900 tests is true while for the last 100 tests is true. Throughout this problem you should test with .

### Part 1a

Using the p-values from above how many discoveries are made? If the testing were working perfectly how many discoveries should have been made?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 1a -|-|-|-|-|-|-|-|-|-|-|-

sum(P < .05)

## [1] 129

129 discoveries were made, but if the testing were working perfectly there would be 100 true discoveries.

### Part 1b

If no correction (PCER) is made for multiple testing, then compute the Type I error rate, Type II error rate, and False Discovery Rate. Is the Type I error rate similar to what you would expect? Explain.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 1b -|-|-|-|-|-|-|-|-|-|-|-

# FALSE means reject null = discovery  
test <- P > 0.05  
test0 <- test[1:T0]  
test1 <- test[(T0+1):(T0+T1)]  
summary(test0)

## Mode FALSE TRUE   
## logical 40 860

summary(test1)

## Mode FALSE TRUE   
## logical 89 11

# the type I error rate is  
sum(test0==FALSE)/T0

## [1] 0.04444444

# the type II error rate is  
sum(test1==TRUE)/T1

## [1] 0.11

# the false discovery rate is  
sum(test0==FALSE) / (sum(test0==FALSE) + sum(test1==FALSE))

## [1] 0.3100775

The Type I error rate is 4.4%, which is close to the 5% level of significance that would be expected. There is a 5% chance of rejecting the null hypothesis by random chance.

### Part 1c

Now attempt to control the family-wise error rate (FWER) using Bonferroni correction. Compute the Type I error rate, Type II error rate, and False Discovery Rate. How do these results compare to using no correction as in Part 1b?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 1c -|-|-|-|-|-|-|-|-|-|-|-

# same as btest <- P > 0.05/(T0+T1)  
btest <- p.adjust(P,method='bonf') > 0.05  
btest0 <- btest[1:T0]  
btest1 <- btest[(T0+1):(T0+T1)]  
summary(btest0)

## Mode TRUE   
## logical 900

summary(btest1)

## Mode FALSE TRUE   
## logical 17 83

# the type I error rate is  
sum(btest0==FALSE)/T0

## [1] 0

# the type II error rate is  
sum(btest1==TRUE)/T1

## [1] 0.83

# the false discovery rate is  
sum(btest0==FALSE) / (sum(btest0==FALSE) + sum(btest1==FALSE))

## [1] 0

The type I error rate and false discovery rate have gone to zero. However, the the type II error rate has increased substantially from 11% to 83%.

### Part 1d

Repeat Part 1c using the Bonferroni-Holm Step-Down procedure to control FWER. Are the results any different than when using the ordinary Bonferroni correction?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 1d -|-|-|-|-|-|-|-|-|-|-|-

holmt <- p.adjust(P,method='holm') > 0.05  
holmt0 <- holmt[1:T0]  
holmt1 <- holmt[(T0+1):(T0+T1)]  
summary(holmt0)

## Mode TRUE   
## logical 900

summary(holmt1)

## Mode FALSE TRUE   
## logical 17 83

# the type I error rate is  
sum(holmt0==FALSE)/T0

## [1] 0

# the type II error rate is  
sum(holmt1==TRUE)/T1

## [1] 0.83

# the false discovery rate is  
sum(holmt0==FALSE) / (sum(holmt0==FALSE) + sum(holmt1==FALSE))

## [1] 0

The results are no different. The type II error is still 83%, and the type I error and the false discovery rate are still both zero.

### Part 1e

Would either of the Bonferroni correction methods be recommended if you were trying to discover possibly significant results for conducting further research into those results? Explain.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 1e -|-|-|-|-|-|-|-|-|-|-|-

It would not be recommended to use either of the Bonferroni correction methods, because the type II error is so high. Having such a high type II error would cause many significant results to appear to be null results.

### Part 1f

Now apply the Benjamin-Hochberg procedure to achieve a target average False Discovery Rate (FDR) of 5%. Compute the Type I error rate, Type II error rate, and False Discovery Rate. Write a brief summary comparing these results to those above.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 1f -|-|-|-|-|-|-|-|-|-|-|-

# No Corrections, FALSE means reject null = discovery  
fdrt <- p.adjust(P, method='BH') > 0.05  
fdrt0 <- fdrt[1:T0]  
fdrt1 <- fdrt[(T0+1):(T0+T1)]  
summary(fdrt0)

## Mode FALSE TRUE   
## logical 3 897

summary(fdrt1)

## Mode FALSE TRUE   
## logical 58 42

# the type I error rate is  
sum(fdrt0==FALSE)/T0

## [1] 0.003333333

# the type II error rate is  
sum(fdrt1==TRUE)/T1

## [1] 0.42

# the false discovery rate is  
sum(fdrt0==FALSE) / (sum(fdrt0==FALSE) + sum(fdrt1==FALSE))

## [1] 0.04918033

The type I error has increased a small amount from 0 to .3%, and the false discovery rate has increased from 0 to 4.9%. The tradeoff is that the type II error has decreased from 83% to 42%.

## Exercise 2

A pharmaceutical company is doing preliminary hypothesis testing of hundreds of compounds to see which ones might be useful in treating a rare form of cancer. What form of multiple testing correction should the company use (none, Bonferroni, or FDR)? Explain your reasoning.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 2 -|-|-|-|-|-|-|-|-|-|-|-

The company should use FDR. The false discovery rate is not as big a problem when in the preliminary phase, because the company is looking for candidate compounds for further research. It also gives you a better chance compared to no correction at all of finding significant discoveries while keeping the False Discovery Rate down.

## Exercise 3

Cars were selected at random from among 1993 passenger car models that were listed in both the Consumer Reports issue and the PACE Buying Guide. Pickup trucks and Sport/Utility vehicles were eliminated due to incomplete information in the Consumer Reports source. Duplicate models (e.g., Dodge Shadow and Plymouth Sundance) were listed at most once. Use the data set Cars93 to do the following. (Type ?Cars93 to learn more about the data.)

For the next two exercises we are going to use the Cars93 data set from the MASS package. We’ll delete the data having to do with vans so that we are only dealing with cars. The code to load and prepare the data is here:

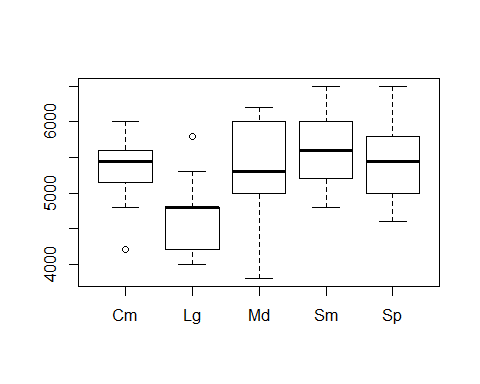
Throughout this exercise we’ll compare population mean engine revolutions per minute at maximum horsepower (RPM) of the different types of cars (Type).

### Part 3a

Make a boxplot of RPM~Type. Is it reasonable to assume the RPM distributions are normal and have equal variances for the different types of cars?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3a -|-|-|-|-|-|-|-|-|-|-|-

boxplot(Cars93$RPM~Cars93$Type)



Md and Lg appear skewed. Use the shapiro test to validate normality.

#'Cm','Lg','Md','Sm','Sp'  
 shapiro.test(Cars93[Cars93$Type == 'Cm',]$RPM)

##   
## Shapiro-Wilk normality test  
##   
## data: Cars93[Cars93$Type == "Cm", ]$RPM  
## W = 0.93324, p-value = 0.2742

shapiro.test(Cars93[Cars93$Type == 'Lg',]$RPM)

##   
## Shapiro-Wilk normality test  
##   
## data: Cars93[Cars93$Type == "Lg", ]$RPM  
## W = 0.90962, p-value = 0.2414

shapiro.test(Cars93[Cars93$Type == 'Md',]$RPM)

##   
## Shapiro-Wilk normality test  
##   
## data: Cars93[Cars93$Type == "Md", ]$RPM  
## W = 0.93456, p-value = 0.1527

shapiro.test(Cars93[Cars93$Type == 'Sm',]$RPM)

##   
## Shapiro-Wilk normality test  
##   
## data: Cars93[Cars93$Type == "Sm", ]$RPM  
## W = 0.94576, p-value = 0.2826

shapiro.test(Cars93[Cars93$Type == 'Sp',]$RPM)

##   
## Shapiro-Wilk normality test  
##   
## data: Cars93[Cars93$Type == "Sp", ]$RPM  
## W = 0.94089, p-value = 0.4298

Based on the Shapiro Test, it is reasonable to assume all the RPM distributions are normal. Visually the variance looks like it could be the same for the distributions of RPMs.

### Part 3b

Conduct pairwise t-tests with no correction for multiple testing to see which mean RPM’s are different from each other. Summarize your findings including a brief description of which types of cars have significanlt different mean RPM ().

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3b -|-|-|-|-|-|-|-|-|-|-|-

pairwise.t.test( Cars93$RPM, Cars93$Type, p.adjust.method='none',  
 pool.sd = TRUE)$p.value < 0.05

## Cm Lg Md Sm  
## Lg TRUE NA NA NA  
## Md FALSE TRUE NA NA  
## Sm FALSE TRUE FALSE NA  
## Sp FALSE TRUE FALSE FALSE

The mean RPM for Lg is significantly different than for Cm, Md, Sm, and Sp.

### Part 3c

Repeat 3b, but this time use Bonferroni correction.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3c -|-|-|-|-|-|-|-|-|-|-|-

pairwise.t.test( Cars93$RPM, Cars93$Type, p.adjust.method='bonf',  
 pool.sd = TRUE)$p.value < 0.05

## Cm Lg Md Sm  
## Lg TRUE NA NA NA  
## Md FALSE TRUE NA NA  
## Sm FALSE TRUE FALSE NA  
## Sp FALSE TRUE FALSE FALSE

The mean RPM for Lg is significantly different than for Cm, Md, Sm, and Sp.

### Part 3d

Now suppose we actually need to estimate the differences in the population mean RPM types while controlling for Type I errors using the Bonferroni correction. Use the onewayComp() function from the DS705data package with adjust = ‘bonferroni’ to compute the CI’s with 95% overall confidence. How much larger is the mean RPM for small cars than for large cars according to your estimates?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3d -|-|-|-|-|-|-|-|-|-|-|-

onewayComp(RPM~Type,data=Cars93,var.equal=TRUE,  
 adjust='bonferroni')$comp

## diff lwr upr t p  
## Lg-Cm -689.77273 -1291.04250 -88.50295 -3.3131069 1.393808e-03  
## Md-Cm -26.13636 -530.52384 478.25111 -0.1496510 8.814214e-01  
## Sm-Cm 270.83333 -238.58556 780.25223 1.5354153 1.286774e-01  
## Sp-Cm 30.35714 -531.44119 592.15547 0.1560556 8.763873e-01  
## Md-Lg 663.63636 96.75378 1230.51894 3.3809275 1.125310e-03  
## Sm-Lg 960.60606 389.24213 1531.97000 4.8554702 5.951613e-06  
## Sp-Lg 720.12987 101.61027 1338.64947 3.3624525 1.193175e-03  
## Sm-Md 296.96970 -171.36579 765.30518 1.8312763 7.082930e-02  
## Sp-Md 56.49351 -468.33778 581.32479 0.3108691 7.567189e-01  
## Sp-Sm -240.47619 -770.14473 289.19235 -1.3111932 1.935909e-01  
## p adj rej H\_0  
## Lg-Cm 1.393808e-02 1  
## Md-Cm 1.000000e+00 0  
## Sm-Cm 1.000000e+00 0  
## Sp-Cm 1.000000e+00 0  
## Md-Lg 1.125310e-02 1  
## Sm-Lg 5.951613e-05 1  
## Sp-Lg 1.193175e-02 1  
## Sm-Md 7.082930e-01 0  
## Sp-Md 1.000000e+00 0  
## Sp-Sm 1.000000e+00 0

The mean RPM for small cars is 961 RPMs larger than for large cars.

### Part 3e

Repeat 3d, but this time use the Tukey-Kramer procedure (use onewayComp() with adust = ‘one.step’ and var.equal=TRUE ). Again, how much larger is the mean RPM for small cars than for large cars according to your estimates?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3e -|-|-|-|-|-|-|-|-|-|-|-

onewayComp(RPM~Type,data=Cars93,var.equal=TRUE,  
 adjust='one.step')$comp

## diff lwr upr t p p adj  
## Lg-Cm -689.77273 -1271.0080 -108.5374 -3.3131069 1.393808e-03 1.182831e-02  
## Md-Cm -26.13636 -513.7175 461.4448 -0.1496510 8.814214e-01 9.998848e-01  
## Sm-Cm 270.83333 -221.6116 763.2783 1.5354153 1.286774e-01 5.429018e-01  
## Sp-Cm 30.35714 -512.7219 573.4362 0.1560556 8.763873e-01 9.998639e-01  
## Md-Lg 663.63636 115.6425 1211.6303 3.3809275 1.125310e-03 9.651980e-03  
## Sm-Lg 960.60606 408.2801 1512.9320 4.8554702 5.951613e-06 5.744492e-05  
## Sp-Lg 720.12987 122.2195 1318.0402 3.3624525 1.193175e-03 1.020512e-02  
## Sm-Md 296.96970 -155.7607 749.7001 1.8312763 7.082930e-02 3.632245e-01  
## Sp-Md 56.49351 -450.8503 563.8373 0.3108691 7.567189e-01 9.979305e-01  
## Sp-Sm -240.47619 -752.4960 271.5437 -1.3111932 1.935909e-01 6.851085e-01  
## rej H\_0  
## Lg-Cm 1  
## Md-Cm 0  
## Sm-Cm 0  
## Sp-Cm 0  
## Md-Lg 1  
## Sm-Lg 1  
## Sp-Lg 1  
## Sm-Md 0  
## Sp-Md 0  
## Sp-Sm 0

The mean RPM for small cars is 961 RPMs larger than for large cars.

### Part 3f

Simultaneous confidence intervals increase the width of the individual intervals to limit the probability that one or more of the intervals are wrong. Both Bonferroni and Tukey-Kramer can provide the family of simultaneous confidence intervals and maintain an overall confidence level of 95%. Compare your results from 3d and 3e. Which set of intervals do you think is better? Why?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3f -|-|-|-|-|-|-|-|-|-|-|-

The populations appear to have similar variances, so I think the Tukey-Kramer confidence intervals are better, because they are tighter.

### Part 3g

Even when you’re using a parametric procedure (one that assumes normality for instance), it can be useful to bootstrap the results to validate the choice of parametric procedure. Use onewayComp() to get the Tukey-Kramer confidence intervals with 95% confidence, but add nboot=10000 to have the code appoximate the critical values used to construct the intervals using bootstrapping. How do the results compare the theoretically derived Tukey-Kramer results from 3e? Does this increase your belief in the validity of the theoretical Tukey-Kramer results?

### -|-|-|-|-|-|-|-|-|-|-|- Answer 3g -|-|-|-|-|-|-|-|-|-|-|-

onewayComp(RPM~Type,data=Cars93,var.equal=TRUE,  
 adjust='one.step', nboot=10000)$comp

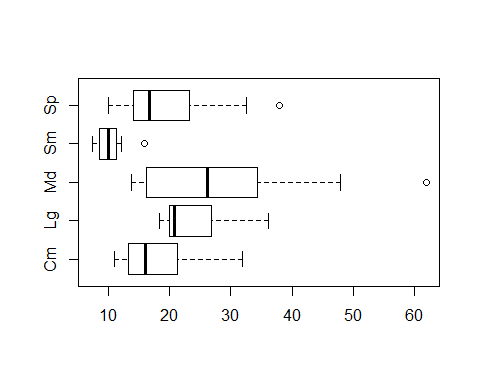
## diff lwr upr t p p adj rej H\_0  
## Lg-Cm -689.77273 -1275.4905 -104.0550 -3.3131069 0.0016 0.0122 1  
## Md-Cm -26.13636 -517.4777 465.2050 -0.1496510 0.8688 0.9998 0  
## Sm-Cm 270.83333 -225.4093 767.0760 1.5354153 0.0834 0.5394 0  
## Sp-Cm 30.35714 -516.9101 577.6244 0.1560556 0.8592 0.9998 0  
## Md-Lg 663.63636 111.4164 1215.8564 3.3809275 0.0018 0.0098 1  
## Sm-Lg 960.60606 404.0206 1517.1915 4.8554702 0.0000 0.0000 1  
## Sp-Lg 720.12987 117.6085 1322.6513 3.3624525 0.0012 0.0103 1  
## Sm-Md 296.96970 -159.2522 753.1916 1.8312763 0.0702 0.3626 0  
## Sp-Md 56.49351 -454.7629 567.7499 0.3108691 0.7584 0.9980 0  
## Sp-Sm -240.47619 -756.4447 275.4923 -1.3111932 0.2080 0.6849 0

The confidence intervals are similar to Tukey-Kramer. Tukey-Kramer confidence intervals appear to be tigther than the bootstrap confidence intervals. This increases my belief in the validity of the Tukey-Kramer results.

## Exercise 4

Now we are going to analyze differences in prices for different types of cars in the Cars93 data set. The boxplot below shows that the prices are skewed and variances are different.

boxplot(Price~Type,horizontal=TRUE,data=Cars93)



### Part 4a

It should be fairly clear that the price data is not from normal distributions, at least for several of the car types, but ignore that for now and use the Games-Howell procedure with confidence level 90% to do simultaneous comparisons (if interpreting the -values use ). (You can use onewayComp() with var.equal=FALSE and ajust = ‘one.step’). Use the CI’s to identify and interpret the largest significant difference in population mean prices.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 4a -|-|-|-|-|-|-|-|-|-|-|-

games.howell <- onewayComp(Price~Type,data=Cars93,var.equal=FALSE,  
 adjust='one.step')  
  
games.howell$pair[[3]]<.1

## Cm Lg Md Sm  
## Lg FALSE NA NA NA  
## Md TRUE FALSE NA NA  
## Sm TRUE TRUE TRUE NA  
## Sp FALSE FALSE FALSE TRUE

games.howell$comp[,1]

## Lg-Cm Md-Cm Sm-Cm Sp-Cm Md-Lg Sm-Lg   
## 6.087500 9.005682 -8.045833 1.180357 2.918182 -14.133333   
## Sp-Lg Sm-Md Sp-Md Sp-Sm   
## -4.907143 -17.051515 -7.825325 9.226190

The largest significant difference in price is between Small and Midsize cars. The Small Midrange price is $17,052 less than Midsize Midrange price.

### Part 4b

Now repeat 4a, but since the price distributions are skewed use bootstrapping by specifying nboot=10000 in onewayComp(). Summarize how these results are different than in 4a.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 4b -|-|-|-|-|-|-|-|-|-|-|-

games.howell <- onewayComp(Price~Type,data=Cars93,var.equal=FALSE, nboot=10000,  
 adjust='one.step')  
  
games.howell$pair[[3]]<.1

## Cm Lg Md Sm  
## Lg FALSE NA NA NA  
## Md FALSE FALSE NA NA  
## Sm TRUE TRUE TRUE NA  
## Sp FALSE FALSE FALSE TRUE

games.howell$comp[,1]

## Lg-Cm Md-Cm Sm-Cm Sp-Cm Md-Lg Sm-Lg   
## 6.087500 9.005682 -8.045833 1.180357 2.918182 -14.133333   
## Sp-Lg Sm-Md Sp-Md Sp-Sm   
## -4.907143 -17.051515 -7.825325 9.226190

They seem to be the same.

### Part 4c

Are the results in 4a and 4b very different? Which results seem more trustworthy, those in 4a or in 4b? Explain.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 4c -|-|-|-|-|-|-|-|-|-|-|-

They seem to be the same. I would trust the bootstrap method more, because the distributions of Price are not normal.

### Part 4d

Since the prices are skewed it might be better to report differences in medians than in means. Use the boot package and Bonferroni correction to bootstrap 4 simultaneous confidence intervals with overall confidence level 90% for the difference in population median price between Sporty Cars and each of the other four car types. You don’t need to interpret the results.

### -|-|-|-|-|-|-|-|-|-|-|- Answer 4d -|-|-|-|-|-|-|-|-|-|-|-

bootMedDiff <- function(d,i){  
 meds <- tapply(d[i,"Price"],d[,"Type"],median)  
 c( meds[5]-meds[1], meds[5]-meds[2], meds[5]-meds[3], meds[5]-meds[4])  
 }

boot.object <- boot(Cars93[,c("Type","Price")], bootMedDiff, R = 5000,   
 strata = Cars93$Type)  
  
# Cm = Compact  
# Lg = Large  
# Md = Midsize  
# Sm = Small  
# Sp = Sporty  
  
# med\_Sp - med\_Cm   
boot.ci(boot.object,conf = 1 - .1/4, type='bca', index=1)$bca[4:5]

## [1] -5.519961 8.489695

# med\_Sp - med\_Lg  
boot.ci(boot.object,conf = 1 - .1/4, type='bca', index=2)$bca[4:5]

## [1] -10.15 4.90

# med\_Sp - med\_Md  
boot.ci(boot.object,conf = 1 - .1/4, type='bca', index=3)$bca[4:5]

## [1] -19.35 -0.25

# med\_Sp - med\_Sm  
boot.ci(boot.object,conf = 1 - .1/4, type='bca', index=4)$bca[4:5]

## [1] 3.5 14.1