Multiple Testing

Josh Jarvey

07/09/2020

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

## 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 -|-|-|-|-|-|-|-|-|-|-|-

#counting the number of times the pnorm function return a value under 0.05. These are our discoveries.  
sum(P<0.05)

## [1] 129

If the tests were working perfectly, we would expect 100 discoveries made. 900 from the mean 0 population would not be rejected, whereas the 100 rejections would be from the mean 3 population (since the null is mean=0)

### 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 -|-|-|-|-|-|-|-|-|-|-|-

#PCER - FALSE means reject null (a discovery)  
 #there is no correction to this P value of 0.05.  
test = P > 0.05  
test0 = test[1:T0]  
test1 = test[(T0+1):(T0+T1)]  
 #count the number TRUE and FALSE (i.e. discoveries) in each population: 40 in population 1, 89 in population 2 (now there are some type II errors occurring in the 2nd population sample)  
summary(test0)

## Mode FALSE TRUE   
## logical 40 860

summary(test1)

## Mode FALSE TRUE   
## logical 89 11

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

## [1] 0.04444444

#type II error rate  
sum(test1==FALSE)/T1

## [1] 0.89

#False discovery rate (FDR)  
sum(test0==FALSE)/(sum(test0==FALSE)+sum(test1==FALSE))

## [1] 0.3100775

The type I error rate (~4%) is as expected per the P-value we set at the beginning of the code block of 0.05 (5%). This means we should expect type I error rate to be at or below this 5% threshold.

### 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 -|-|-|-|-|-|-|-|-|-|-|-

#PCER - FALSE means reject null (a discovery)  
 #there is no correction to this P value of 0.05.  
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

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

## [1] 0

##type II error rate  
sum(btest1==FALSE)/T1

## [1] 0.17

##False discovery rate (FDR)  
sum(btest0==FALSE)/(sum(btest0==FALSE)+sum(btest1==FALSE))

## [1] 0

The bonferroni correction has accounted for all Type I errors with its extreme conservative approach (pvalue is much smaller to pass the test). However, this has lead to an overcompensation and increased the number of false discoveries in the second population with mean 3 - there are now 83 in the TRUE category.

### 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 -|-|-|-|-|-|-|-|-|-|-|-

bhtest = p.adjust(P,method = "holm") > 0.05  
bhtest0 = bhtest[1:T0]  
bhtest1 = bhtest[(T0+1):(T0+T1)]  
summary(bhtest0)

## Mode TRUE   
## logical 900

summary(bhtest1)

## Mode FALSE TRUE   
## logical 17 83

#type I error rate  
sum(bhtest0==FALSE)/T0

## [1] 0

##type II error rate  
sum(bhtest1==FALSE)/T1

## [1] 0.17

##False discovery rate (FDR)  
sum(bhtest0==FALSE)/(sum(bhtest0==FALSE)+sum(bhtest1==FALSE))

## [1] 0

There is no difference between the normal bonferroni and the bonferroni-holm method (however i would reccomend the latter since it is more powerful than the plain bonferroni).

### 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 -|-|-|-|-|-|-|-|-|-|-|-

If we were conducting exploratory analysis on the dataset, I would not recommend bonferroni because it is too conservative, and we may miss significant results causing a Type II error of not rejecting a false null (and further investigating the alternative for that variable).

### 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 -|-|-|-|-|-|-|-|-|-|-|-

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

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

## [1] 0.003333333

##type II error rate  
sum(fdrt1==FALSE)/T1

## [1] 0.58

##False discovery rate (FDR)  
sum(fdrt0==FALSE)/(sum(fdrt0==FALSE)+sum(fdrt1==FALSE))

## [1] 0.04918033

In this example with the Benjamin-Hochberg correction vs. the Bonferroni, we “trade” a few type I errors (0 to 3 in this example) for a reduction of type II errors (83 to 42 in this example). It does a nice job controlling our overall false discovery rate (FDR) by keeping it under the 5% specified (~4.9%)

## 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 -|-|-|-|-|-|-|-|-|-|-|-

Not correcting the p-value here would allow for too many type I errors (which would waste time), yet because this is exploratory, using a Bonferroni is too conservative and may cause too many type II errors (actually missing compounds that could be important) - so my final answer is going to be C) FDR, or the benjamin-hochberg correction!

## 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:

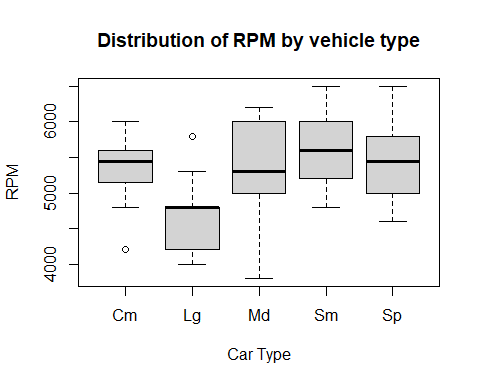
Throughout this exercise we’ll compare population mean engine revolutions per minute at maximum horsepower (RPM) of the different types of cars (Type).

### 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 -|-|-|-|-|-|-|-|-|-|-|-

#create a boxplot of the car RPM by type.  
boxplot(RPM~Type, data = Cars93, xlab = "Car Type", main = "Distribution of RPM by vehicle type")



While there do appear to be some categories of car type that have normally shaped distributions, this does not appear be the case for all types. The type “Md” is skewed to the left, type “Lg” is also skewed to the left with an outlier to the right. “Cm” also has an outlier to the left.

They generally all share the same variance, however type “Md” does appear to have a larger variance than the others.

### 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 significantly different mean RPM ().

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

#using the Cars93 dataset, conduct pairwise t-tests with no correction and pooled variance.   
 #Use a 5% significance level for testing between pairs.  
with(Cars93, pairwise.t.test(RPM,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

There are 4 significant differences out of 10 possible pairs, but Type I errors may be present because no correction method was applied to the p-values. Mean RPM for vehicle type “Lg” is significantly different than for “Cm”, “Md”, “Sm”, “Sp”.

### Part 3c

Repeat 3b, but this time use Bonferroni correction.

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

#using the Cars93 dataset, conduct pairwise t-tests with no correction and pooled variance.   
 #Use a 5% significance level for testing between pairs.  
with(Cars93, pairwise.t.test(RPM,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

There are 4 significant differences out of 10 possible pairs, but Type I errors are now, on average, 5% or less because the Bonferonni correction method was applied to the p-values. Mean RPM for vehicle type “Lg” is significantly different than for “Cm”, “Md”, “Sm”, “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 -|-|-|-|-|-|-|-|-|-|-|-

library(DS705data)  
onewayComp(RPM~Type,data = Cars93,var.equal = TRUE,adjust = "bonferroni")$comp[,c(2,3,6,7)]

## lwr upr p adj rej H\_0  
## Lg-Cm -1291.04250 -88.50295 1.393808e-02 1  
## Md-Cm -530.52384 478.25111 1.000000e+00 0  
## Sm-Cm -238.58556 780.25223 1.000000e+00 0  
## Sp-Cm -531.44119 592.15547 1.000000e+00 0  
## Md-Lg 96.75378 1230.51894 1.125310e-02 1  
## Sm-Lg 389.24213 1531.97000 5.951613e-05 1  
## Sp-Lg 101.61027 1338.64947 1.193175e-02 1  
## Sm-Md -171.36579 765.30518 7.082930e-01 0  
## Sp-Md -468.33778 581.32479 1.000000e+00 0  
## Sp-Sm -770.14473 289.19235 1.000000e+00 0

We are 95% confident that the population mean RPM for “Lg” is less than the population mean RPM for “Cm”, “Md”, “Sm”, “Sp” by 88.50 to 1291.04, 96.75 to 1230.51, 389.24 to 1531.97, and 101.61 to 1338.64 RPMs respectively.

### 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[,c(2,3,6,7)]

## lwr upr p adj rej H\_0  
## Lg-Cm -1271.0080 -108.5374 1.182831e-02 1  
## Md-Cm -513.7175 461.4448 9.998848e-01 0  
## Sm-Cm -221.6116 763.2783 5.429018e-01 0  
## Sp-Cm -512.7219 573.4362 9.998639e-01 0  
## Md-Lg 115.6425 1211.6303 9.651980e-03 1  
## Sm-Lg 408.2801 1512.9320 5.744492e-05 1  
## Sp-Lg 122.2195 1318.0402 1.020512e-02 1  
## Sm-Md -155.7607 749.7001 3.632245e-01 0  
## Sp-Md -450.8503 563.8373 9.979305e-01 0  
## Sp-Sm -752.4960 271.5437 6.851085e-01 0

We are 95% confident that the population mean RPM for “Lg” is less than the population mean RPM for “Cm”, “Md”, “Sm”, “Sp” by 108.53 to 1271.00, 115.64 to 1211.63, 408.28 to 1512.93, and 122.21 to 1318.04 RPMs respectively.

### 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 Tukey-Kramer has tighter confidence intervals when compared to the Bonferroni corrected CI’s, therefore I would suggest the Tukey-Kramer since it has more accuracy (not as conservative with guarding against the Type I error) in identifying the range for the true difference in population means.

### 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 approximate 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 -|-|-|-|-|-|-|-|-|-|-|-

#using onewaycomp() to conduct a bootstrapped Tukey-Kramer test (adjust=one.step)  
onewayComp(RPM~Type,data = Cars93,var.equal = TRUE,adjust = "one.step", nboot = 10000)$comp[,c(2,3,6,7)]

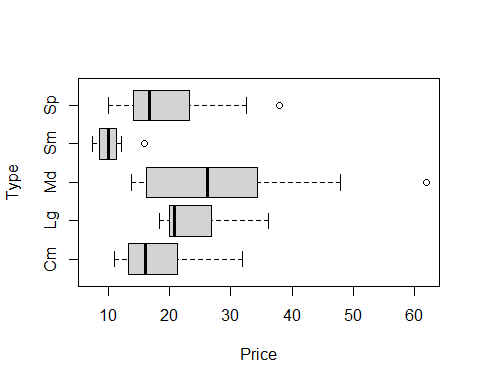
## lwr upr p adj rej H\_0  
## Lg-Cm -1270.9485 -108.5970 0.0130 1  
## Md-Cm -513.6675 461.3948 1.0000 0  
## Sm-Cm -221.5611 763.2278 0.5347 0  
## Sp-Cm -512.6663 573.3806 1.0000 0  
## Md-Lg 115.6986 1211.5741 0.0105 1  
## Sm-Lg 408.3367 1512.8754 0.0001 1  
## Sp-Lg 122.2808 1317.9790 0.0109 1  
## Sm-Md -155.7143 749.6537 0.3545 0  
## Sp-Md -450.7983 563.7853 0.9977 0  
## Sp-Sm -752.4436 271.4912 0.6730 0

After running the bootstrapped version of the Tukey-Kramer test and comparing the confidence intervals to the theoretically derived Tukey-Kramer results from the previous step, there is nominal difference between the two (they are different by roughly less than 1 RPM in most cases). Since they are close (and bootstrapping accounts for normality), I feel more confident with the theoretical 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 adjust = ‘one.step’). Use the CI’s to identify and interpret the largest significant difference in population mean prices.

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

#using the onewaycomp() function, complete a games-howell (var.equals=FALSE)test to generate pairwise confidence intervals for car prices-type  
onewayComp(Price~Type,data = Cars93, var.equal = FALSE,adjust = "one.step")$comp[,c(2,3,6,7)]

## lwr upr p adj rej H\_0  
## Lg-Cm -1.43351665 13.608517 1.266833e-01 0  
## Md-Cm 0.06552497 17.945839 3.759910e-02 1  
## Sm-Cm -13.29611220 -2.795554 1.192465e-04 1  
## Sp-Cm -6.76232482 9.123039 9.923626e-01 0  
## Md-Lg -6.45877344 12.295137 8.956675e-01 0  
## Sm-Lg -20.46424531 -7.802421 2.789471e-09 1  
## Sp-Lg -13.36906316 3.554777 4.312457e-01 0  
## Sm-Md -24.90872496 -9.194305 8.678284e-08 1  
## Sp-Md -17.53995770 1.889308 1.495367e-01 0  
## Sp-Sm 2.45658636 15.995795 5.542557e-04 1

We are 90% confident that the largest mean price difference in vehicles is between “Sm” and “Md”, where “Sm” is 9.19 to 24.90 thousand dollars less than “Md”.

### 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 -|-|-|-|-|-|-|-|-|-|-|-

#using the onewaycomp() function, complete a bootstrapped version of the games-howell (var.equals=FALSE) test to generate confidence intervals  
onewayComp(Price~Type,data = Cars93, var.equal = FALSE,adjust = "one.step", nboot = 10000)$comp[,c(2,3,6,7)]

## lwr upr p adj rej H\_0  
## Lg-Cm -3.7199761 15.894976 0.2063 0  
## Md-Cm -2.9831705 20.994534 0.1189 0  
## Sm-Cm -14.7101509 -1.381516 0.0267 1  
## Sp-Cm -9.2832327 11.643947 0.9922 0  
## Md-Lg -9.5924165 15.428780 0.8892 0  
## Sm-Lg -21.6961117 -6.570555 0.0028 1  
## Sp-Lg -15.9646833 6.150398 0.4657 0  
## Sm-Md -27.2858500 -6.817180 0.0061 1  
## Sp-Md -20.8566406 5.205991 0.2265 0  
## Sp-Sm 0.8300163 17.622365 0.0372 1

We are 90% confident that the largest mean price difference in vehicles is between “Sm” and “Md”, where “Sm” is 6.81 to 27.28 thousand dollars less than “Md”.

### 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 -|-|-|-|-|-|-|-|-|-|-|-

## The results in 4b (bootstrapped version) seem a bit more trustworthy even though the confidence interval is a bit larger than the non-bootstrapped version. The reason im going with the bootstrapped version is because per the boxplots, these population’s sample price do not look normally distributed and therefore would skew the results of the standard. Both the Tukey and Games tests require normality as a condition.

### 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 -|-|-|-|-|-|-|-|-|-|-|-

#load the bootstrap package  
library(boot)  
  
 #creating the auxiliary function - it requires a dataframe "d", where the quantitative variable of interest is in column1  
 #column2 contains the factor variable  
bootMedDiff <- function(d,i){  
 #use tapply to apply the median() calculation to the values (re)sampled from the bootstrap.   
 #again, values are in the column 1 of the current data frame, and factor is in column 2. tapply applies based on these factors  
 meds = tapply(d[i,1],d[,2],median)  
   
 #now, calculate differences in median per this bootstrap iteration. We are only focused on sporty cars, so commenting out the other pairs.  
 c(meds[5]-meds[1],#Sp-Cm  
 meds[5]-meds[2],#Sp-Lg  
 meds[5]-meds[3],#SP-Md  
 meds[5]-meds[4]#,#Sp-Sm  
 #meds[4]-meds[3],#Sm-Md  
 #meds[4]-meds[2],#Sm-Lg  
 #meds[4]-meds[1],#Sm-Cm  
 #meds[3]-meds[2],#Md-Lg  
 #meds[3]-meds[1],#Md-Cm  
 #meds[2]-meds[1] #Lg-Cm  
 )  
}  
  
 #create the bootstrap object using just the Price & Type from Cars93 dataset.   
boot.object = boot(Cars93[,c("Price","Type")],bootMedDiff,R=5000,strata = Cars93$Type)  
  
#Calculate the various pairs of confidence intervals. Because we're completing 4 sets of tests, m=4.  
  
 #Sporty Cars vs. other  
 #Comparison: Sp-Cm  
boot.ci(boot.object, conf = 1 - 0.10/4, type = "bca", index = 1)$bca[4:5]

## [1] -5.45 8.45

#Comparison: Sp-Lg  
boot.ci(boot.object, conf = 1 - 0.10/4, type = "bca", index = 2)$bca[4:5]

## [1] -10.22892 5.10000

#Comparison: SP-Md  
boot.ci(boot.object, conf = 1 - 0.10/4, type = "bca", index = 3)$bca[4:5]#<--this median appears to be shifted 19.69 to .62 < than Medium cars.

## [1] -19.42089 -0.40000

#Comparison: Sp-Sm  
boot.ci(boot.object, conf = 1 - 0.10/4, type = "bca", index = 4)$bca[4:5]#<--this median appears to be shifted 3.50 to 13.63 > than Small cars.

## [1] 3.45 14.10

##### instructions say these additional CI's are not needed #####  
  
 #Small Cars vs. other (sporty already in above section)  
 #Comparison: Sm-Md  
#boot.ci(boot.object, conf = 1 - 0.10/10, type = "bca", index = 5)$bca[4:5]  
 #Comparison: Sm-Lg  
#boot.ci(boot.object, conf = 1 - 0.10/10, type = "bca", index = 6)$bca[4:5]  
 #Comparison: Sm-Cm  
#boot.ci(boot.object, conf = 1 - 0.10/10, type = "bca", index = 7)$bca[4:5]  
  
 #Medium Cars vs. other (sporty & small already in the above section)  
 #Comparison: Md-Lg  
#boot.ci(boot.object, conf = 1 - 0.10/10, type = "bca", index = 8)$bca[4:5]  
 #Comparison: Md-Cm  
#boot.ci(boot.object, conf = 1 - 0.10/10, type = "bca", index = 9)$bca[4:5]  
  
 #Large Cars vs. other (sporty & small & medium already in the above section)  
 #Comparison: Lg-Cm  
#boot.ci(boot.object, conf = 1 - 0.10/10, type = "bca", index = 10)$bca[4:5]