**Multiple comparisons**

If the hypothesis test for equality of means results in a “reject Ho”, there is at least one pair of means that are different. This section shows how to determine which pair(s) of means is(are) different. Thus, we will perform “multiple comparisons” for differences of mean pairs.

How can we examine mean pairs?

* CI for μi – μi′ for i ≠ i′
* Hypothesis test for μi – μi′ for i ≠ i′

How many comparisons are possible?



For example, the Wheaties example has t = 4 leading to 6 possible comparisons:

1. μ1 – μ2
2. μ1 – μ3
3. μ1 – μ4
4. μ2 – μ3
5. μ2 – μ4
6. μ3 – μ4

Suppose each comparison uses a level of α for the type I error rate in a hypothesis test. What is the probability of making AT LEAST one type I error for all multiple comparisons?

NOT α

The probability of making at least one type I error is referred to as the experimentwise type I error rate. Some people refer to it as the familywise type I error rate too. We will denote this error rate as αE. The individual type I error rate used for each test is denoted by αI to help differentiate it.

To see what the experimentwise type I error rate could be, consider the setting of 2 multiple comparisons. Let Aj be the event that a type I error occurs for the jth comparison of means. Then



Because  ≥ 0, we have the following inequality:



If , then

.

In general for m multiple comparisons, we have

.

Thus, αE ≤ mαI, so that the probability of at least one type I error could be as large as mαI!.

There have been MANY different multiple comparison methods proposed in order to control the experimentwise error rate at a pre-set level. In fact, whole courses are taught on this topic at some universities! The methods we will discuss are:

1. Fisher’s protected least significant differences (LSD) method
2. Bonferroni method
3. Tukey's honest significant differences (HSD) method

One book on this topic is “Multiple Comparisons Using R” by Bretz, Horthorn, and Westfall if you want to investigate these methods beyond those discussed in our class.

Fisher’s protected LSD

This procedure is VERY similar to finding a CI and/or performing a hypothesis test for the difference of two means as seen earlier in the course. The confidence interval is



The hypothesis test for Ho: μi – μi′ = 0 vs. Ho: μi – μi′ ≠ 0 uses a test statistic of



with critical values of . Equivalently, we could look at  for each (i, i′) pair and compare it to

.

The “protected” part of the name comes from only performing these comparisons if Ho: μ1 = … = μt is rejected. The LSD part of the name comes from



being the smallest (least) possible value where a “significant” difference in population means is found.

Unfortunately, this does not give us a measure of the experimentwise error rate.

Bonferroni

Because αE ≤ mαI, we could adjust the value of αI to achieve a desired experimentwise error level. Thus, for an experimentwise error rate of no more than αE, use a αI equal to αE/m. For example, if we want αE = 0.05 and m = 2, then use αI = 0.05/2 = 0.025 to have an experimentwise error rate no larger than 0.05.

For m possible comparisons, the CI becomes



The critical values are . The p-value needs to be < αE/m to reject Ho: μ1 – μ2 = 0 using the projected LSD test statistic. Sometimes, p-values will be automatically adjusted for you (R does this) so you can compare the p-value directly to αE. The p-value adjustment is simply min(p-value × m, 1). For example, if a p-value from the projected LSD test was 0.12. The adjusted p-value for a Bonferroni test would be the smallest of 0.12×m or 1.

Tukey's HSD

This procedure uses a different probability distribution to help make its comparisons. Specifically, we “could” use the test statistic



where

*  is the largest sample mean among the t treatments
*  is the smallest sample mean among the t treatments.

This test statistic looks very similar to the statistic that we first used for testing the difference between two means. However, because we have to wait to determine which is the largest and the smallest sample mean, the mathematics does not work out for the statistic to have a t distribution. Instead, it has a studentized range probability distribution. R has functions ptukey() and qtukey() that calculate probabilities and quantiles from this probability distribution, but there is no dtukey() function that would allow me to show you a plot of the distribution.

Below are the important items from using this procedure:

* CI: 

where  is the 1 – αE quantile from a studentized range probability distribution where t is the total number of treatments and N – t is the degrees of freedom associated with MSE.

* Test statistic: 
* P-value: 

where X has the appropriate studentized range probability distribution

The experimentwise error rate is αE.

Comparisons of the multiple comparisons procedures

Control of the experimentwise error rate

* LSD: None
* Bonferroni: ≤ αE
* HSD: αE

Of course, the above values are conditional on all of the assumptions behind ANOVA being correct.

Typically, LSD is often called a “liberal” procedure because it may find more population mean differences than those that actually exist. Bonferroni is often called a “conservative” procedure because it may find less population mean differences than those that actually exist.

Displaying the results

Once multiple comparisons are completed, there is a type of “plot” that is used to summarize this information:

1. Order the treatment “symbols” by their sample means (from smallest to largest).
2. Place bars over treatments where there is not sufficient evidence to show that their means are different.

For example, suppose there are four treatments (A, B, C, and D) with sample means:

|  |  |
| --- | --- |
| **Treatment** | **Sample Mean** |
| A | 1 |
| B | 4 |
| C | 3 |
| D | 2 |

Then



denotes

* A’s population mean is different than C and B’s population mean.
* A’s population mean is not found to be different than D’s population mean.
* D, C, and B’s population means are not found to be different.

Often this is worded as, “A is different from B and C. A is not significantly different than D. D, C, and B are not significantly different.”

Some books will draw the horizontal lines below the letters instead of above the letters.

Example: Wheaties Cereal (wheaties.R, wheaties.csv)

Which design type means are different? We need to compare 6 different mean pairs using

Ho: μi - μi′= 0

Ha: μi - μi′ ≠ 0

for i, i′ = 1, 2, 3, 4 and i < i′ with αE = 0.05. Below is the R code and output where mod.fit contains the results from aov(). First, we examine the results from LSD:

> aggregate(x = Response ~ Design, data = wheaties,

Added after video recording: R has changed the syntax for aggregate(). In the video, I show formula = Response ~ Design. Now, the proper syntax is x = Response ~ Design. I made the correction here and in the program.

FUN = mean)

Design Response

1 1 15

2 2 13

3 3 19

4 4 27

> pairwise.t.test(x = wheaties$Response, g =

wheaties$Design, p.adjust.method = "none", alternative

= "two.sided")

Pairwise comparisons using t tests with pooled SD

data: wheaties$Response and wheaties$Design

1 2 3

2 0.4589 - -

3 0.1646 0.0378 -

4 0.0049 0.0015 0.0194

P value adjustment method: none

The table above gives p-values. Using a αI = 0.05 level, we have:



If we used αI = 0.01 instead, notice the plot changes to



Using Bonferroni:

> pairwise.t.test(x = wheaties$Response, g =

wheaties$Design, p.adjust.method = "bonferroni",

alternative = "two.sided")

Pairwise comparisons using t tests with pooled SD

data: wheaties$Response and wheaties$Design

1 2 3

2 1.0000 - -

3 0.9877 0.2270 -

4 0.0294 0.0088 0.1166

P value adjustment method: bonferroni

Using an experimentwise error rate of αE = 0.05, we can compare 0.05 to the p-values in the above table to determine which means are different. This leads to:



Notice that we obtain the same p-values if the LSD p-values are multiplied by 6 (the number of comparisons):

>LSD <- pairwise.t.test(x = wheaties$Response, g =

wheaties$Design, p.adjust.method = "none",

alternative = "two.sided")

> names(LSD)

[1] "method" "data.name" "p.value"

"p.adjust.method"

> LSD$p.value

1 2 3

2 0.458921723 NA NA

3 0.164619744 0.037828182 NA

4 0.004907837 0.001461194 0.01944082

> LSD$p.value\*6

1 2 3

2 2.75353034 NA NA

3 0.98771846 0.226969090 NA

4 0.02944702 0.008767167 0.1166449

where a maximum value of 1 was given by pairwise.t.test() when the multiplication results in a value greater than 1. Equivalently, we could use αE/6 with the LSD p-values to decide which mean pairs are different.

Using HSD:

> TukeyHSD(x = mod.fit, conf.level = 0.95)

Tukey multiple comparisons of means

95% family-wise confidence level

Fit: aov(formula = Response ~ factor(Design), data = wheaties)

$`factor(Design)`

diff lwr upr p adj

2-1 -2 -10.7499087 6.749909 0.8561049

3-1 4 -4.7499087 12.749909 0.4522099

4-1 12 2.4149552 21.585045 0.0190946

3-2 6 -1.8261563 13.826156 0.1304578

4-2 14 5.2500913 22.749909 0.0058766

4-3 8 -0.7499087 16.749909 0.0707158

Using an experimentwise error rate of αE = 0.05 level, we can compare 0.05 to the p-values in the above table to determine which means are different. This leads to:



Summary of the results:

# Design type 1 and 2 are different from design type 4. Because the confidence intervals are all positive for μ4 – μ1 and μ4 – μ2, design type 4’s mean sales are larger than design type 1 and 2’s mean sales.

# There is not sufficient evidence to indicate a difference between design types 3 and 4 means.

# There is not sufficient evidence to indicate a difference between design types 1, 2, and 3 means.

How could you determine that design type 4’s population mean is **greater** than design type 2’s population mean using the output from pairwise.t.test()?

Only the p-values are given so you can not directly. However, you can look at the sample means. Because  AND the p-value is small, we know that .

Which package design should General Mills use?

There are packages specifically designed for multiple comparisons:

* agricolae

> library(package = agricolae)

> save.LSD <- LSD.test(y = mod.fit, trt =

"factor(Design)", alpha = 0.05, group = FALSE, p.adj

= "none", main = "Wheaties box design")

> names(save.LSD)

[1] "statistics" "parameters" "means" "comparison"

"groups"

> save.LSD

$statistics

MSerror Df Mean CV

7.666667 6 18 15.38264

$parameters

test p.ajusted name.t ntr alpha

Fisher-LSD none factor(Design) 4 0.05

$means

Response std r LCL UCL Min Max Q25 Q50 Q75

1 15 4.242641 2 10.209216 19.79078 12 18 13.5 15 16.5

2 13 1.000000 3 9.088341 16.91166 12 14 12.5 13 13.5

3 19 2.000000 3 15.088341 22.91166 17 21 18.0 19 20.0

4 27 4.242641 2 22.209216 31.79078 24 30 25.5 27 28.5

$comparison

difference pvalue signif. LCL UCL

1 - 2 2 0.4589 -4.184876 8.1848759

1 - 3 -4 0.1646 -10.184876 2.1848759

1 - 4 -12 0.0049 \*\* -18.775192 -5.2248079

2 - 3 -6 0.0378 \* -11.531921 -0.4680788

2 - 4 -14 0.0015 \*\* -20.184876 -7.8151241

3 - 4 -8 0.0194 \* -14.184876 -1.8151241

$groups

NULL

attr(,"class")

[1] "group"

> save.Bon <- LSD.test(y = mod.fit, trt =

"factor(Design)", alpha = 0.05, group = FALSE, p.adj

= "bonferroni", main = "Wheaties box design")

> save.Bon

$statistics

MSerror Df Mean CV

7.666667 6 18 15.38264

$parameters

test p.ajusted name.t ntr alpha

Fisher-LSD bonferroni factor(Design) 4 0.05

$means

Response std r LCL UCL Min Max Q25 Q50 Q75

1 15 4.242641 2 10.209216 19.79078 12 18 13.5 15 16.5

2 13 1.000000 3 9.088341 16.91166 12 14 12.5 13 13.5

3 19 2.000000 3 15.088341 22.91166 17 21 18.0 19 20.0

4 27 4.242641 2 22.209216 31.79078 24 30 25.5 27 28.5

$comparison

difference pvalue signif. LCL UCL

1 - 2 2 1.0000 -7.764192 11.764192

1 - 3 -4 0.9877 -13.764192 5.764192

1 - 4 -12 0.0294 \* -22.696137 -1.303863

2 - 3 -6 0.2270 -14.733359 2.733359

2 - 4 -14 0.0088 \*\* -23.764192 -4.235808

3 - 4 -8 0.1166 -17.764192 1.764192

$groups

NULL

attr(,"class")

[1] "group"

Notice how the factor was specified in the LSD.test() function. This needs to be the name as displayed by summary(object = mod.fit). Again, factor() would be unnecessary if the factor levels were not coded strictly as numeric.

There are no options in this function for HSD. However, there is a HSD.test() function available for it.

> save.HSD <- HSD.test(y = mod.fit, trt =

"factor(Design)", alpha = 0.05, group = FALSE, main =

"Wheaties box design")

> save.HSD

$statistics

MSerror Df Mean CV

7.666667 6 18 15.38264

$parameters

test name.t ntr StudentizedRange alpha

Tukey factor(Design) 4 4.895599 0.05

$means

Response std r Min Max Q25 Q50 Q75

1 15 4.242641 2 12 18 13.5 15 16.5

2 13 1.000000 3 12 14 12.5 13 13.5

3 19 2.000000 3 17 21 18.0 19 20.0

4 27 4.242641 2 24 30 25.5 27 28.5

$comparison

difference pvalue signif. LCL UCL

1 - 2 2 0.8561 -6.749909 10.7499087

1 - 3 -4 0.4522 -12.749909 4.7499087

1 - 4 -12 0.0191 \* -21.585045 -2.4149552

2 - 3 -6 0.1305 -13.826156 1.8261563

2 - 4 -14 0.0059 \*\* -22.749909 -5.2500913

3 - 4 -8 0.0707 . -16.749909 0.7499087

$groups

NULL

attr(,"class")

[1] "group"

* multcomp – This is the most advanced package in R for multiple comparisons, but it often requires an experimental design course understanding for multiple comparisons. Note that this package corresponds to the Bretz, Horthorn, and Westfall (2011) book.

Multiple comparisons using HSD:

> library(package = multcomp)

> HSD <- glht(model = mod.fit, linfct =

mcp("factor(Design)" = "Tukey"))

> summary(object = HSD)

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = Response ~ factor(Design), data = wheaties)

Linear Hypotheses:

Estimate Std. Error t value Pr(>|t|)

2 - 1 == 0 -2.000 2.528 -0.791 0.85554

3 - 1 == 0 4.000 2.528 1.583 0.45121

4 - 1 == 0 12.000 2.769 4.334 0.01887 \*

3 - 2 == 0 6.000 2.261 2.654 0.13021

4 - 2 == 0 14.000 2.528 5.539 0.00608 \*\*

4 - 3 == 0 8.000 2.528 3.165 0.07028 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Adjusted p values reported -- single-step method)

> confint(object = HSD, level = 0.95)

Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = Response ~ factor(Design), data = wheaties)

Quantile = 3.459

95% family-wise confidence level

Linear Hypotheses:

Estimate lwr upr

2 - 1 == 0 -2.0000 -10.7430 6.7430

3 - 1 == 0 4.0000 -4.7430 12.7430

4 - 1 == 0 12.0000 2.4225 21.5775

3 - 2 == 0 6.0000 -1.8200 13.8200

4 - 2 == 0 14.0000 5.2570 22.7430

4 - 3 == 0 8.0000 -0.7430 16.7430

> plot(HSD)



Please see the corresponding program for how to do the multiple comparisons using LSD and Bonferroni.