Multiple Testing

A Long Presentation

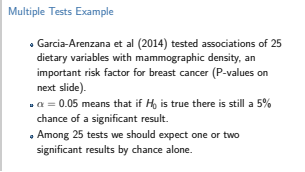
- Multiple Testing is a big topic and a subject of active research.
- Blend of classical and current approaches - the material on the False Discovery Rate isn't in your textbook, but is definitely applicable to big data.
- The right approach depends on your application.
- Many of the slides show the same examples with different approaches so you can flip through many of these quickly.

A Long Presentation
- Multiple Testing is a big topic and a subject of active research.
- Blend of classical and current approaches - the material on the False Discovery Rate isn't in your textbook, but is definitely applicable to big data.
- The right approach depends on your application.
- Many of the slides show the same examples with different approaches so you can flip through many of these quickly.

- audio01.mp3
- The presentation this week is one of our longer ones. There are two reasons for that
- first multiple testing is a big topic for which there are entire books and some of the procedures we cover aren't addressed in the textbook.
- Most notably we present material on the False Discovery Rate which has only been around since 1995 and grew out of the field of genomics where it's standard to have thousands of simultaneous comparisons.
- The second reason is that we probably went a little over the top with many examples in R to show you how to implement various procedures.
- Many of the R slides are similar in nature so we think you'll be able to flip through

Multiple Tests Example

- Garcia-Arenzana et al (2014) tested associations of 25 dietary variables with mammographic density, an important risk factor for breast cancer (P-values on next slide).
- $\alpha = 0.05$ means that if H_0 is true there is still a 5% chance of a significant result.
- Among 25 tests we should expect one or two significant results by chance alone.



- no audio
- add reference below slide: García-Arenzana, N., E.M. Navarrete-Muñoz, V. Lope, P. Moreo, S. Laso-Pablos, N. Ascunce, F. Casanova-Gómez, C. Sánchez-Contador, C. Santamariña, N. Aragonés, B.P. Gómez, J. Vioque, and M. Pollán. 2014. Calorie intake, olive oil consumption and mammographic density among Spanish women. International journal of cancer 134: 1916-1925.

Multiple P-values

Dietary Variable	P	Dietary Variable	P
Total Calories	<0.001	Eggs 0.275	
Olive oil	0.008	Blue fish	0.34
Whole milk	0.039	Legumes	0.341
White meat	0.041	Carbohydrates	0.384
Proteins	0.042	Potatoes	0.569
Nuts	0.06	Bread	0.594
Cereals and pasta	0.074	Fats	0.696
White Fish	0.205	Sweets	0.762
Butter	0.212	Dairy products	0.94
Vegetables	0.216	Semi-skimmed milk	0.942
Skimmed Milk	0.222	Total meat	0.975
Red Meat	0.251	Processed meat	0.986
Fruit	0.269		

Multiple P-values

Dietary Variable	P	Dietary Variable	P
Total Calories	<0.001	Eggs	0.275
Olive oil	0.008	Blue fish	0.34
Whole milk	0.039	Legumes	0.341
White meat	0.041	Carbohydrates	0.384
Proteins	0.042	Potatoes	0.569
Nuts	0.06	Bread	0.594
Cereals and pasta	0.074	Fats	0.696
White Fish	0.205	Sweets	0.762
Butter	0.212	Dairy products	0.94
Vegetables	0.216	Semi-skimmed milk	0.942
Skimmed Milk	0.222	Total meat	0.975
Red Meat	0.251	Processed meat	0.986
Fruit	0.269		

- audio02.mp3
- If we're testing at the 5% level, then there are 5 significant results here. If there weren't truly any significant results we'd expect to find 1 or 2 false positives by chance alone.
- add below this slide: The idea to use this example cam from `http://www.biostathandbook.com/multiplecomparisons.html` which is really good and quick read on this subject.

Multiple Statistical Tests

- **Possible Problem:** As the number of tests increases so does the fraction of the tests that may be wrong due to random chance alone.
- Depending on application we might have to apply a *multiple tests correction* to reduce the chance of Type I errors and/or Type II errors.

Multiple Statistical Tests

- **Possible Problem:** As the number of tests increases so does the fraction of the tests that may be wrong due to random chance alone.
- Depending on application we might have to apply a *multiple tests correction* to reduce the chance of Type I errors and/or Type II errors.

- no audio

The Multiple Testing Problem

- perform m simultaneous hypothesis tests with a common procedure

	H_0 retained (test non-significant)	H_0 rejected (test significant)	Total
H_0 true	TN	FD	T_0
H_a true	FN	TD	T_1
Total	N	D	m

- T/F = True/False, D/N = Discovery/Nondiscovery
- can only observe N, D and m
- FD = False Discovery = Type I error
- FN = False Nondiscovery = Type II error

The Multiple Testing Problem

- perform m simultaneous hypothesis tests with a common procedure

	H_0 retained (test non-significant)	H_0 rejected (test significant)	Total
H_0 true	TN	FD	T_0
H_a true	FN	TD	T_1
Total	N	D	m

- T/F — True/False, D/N — Discovery/Nondiscovery
- can only observe N, D and m
- FD — False Discovery — Type I error
- FN — False Nondiscovery — Type II error

- audio03.mp3
- we'll say that significant tests in which the null hypothesis is rejected represent discoveries
- a Type I error is a false discovery, represented by FD in the table
- a Type II error is a false nondiscovery, that is there truly is a significant effect but the test failed to reveal it, we call this FN in the table.
- the more simultaneous tests we do, the more opportunities for these errors occur due to chance, but in the end after we've done m tests all we know if how many nulls we've retained and how many we've rejected.
- Statisticians have developed many, many procedures to allow the user to try to

Do we need multiple tests correction?

- If false discoveries are really bad, then yes. What if you're comparing multiple new medical treatments to an existing treatment?
- If false nondiscoveries are really bad, then no. What if you're just looking for possible dietary factors that might be linked to breast cancer?
- Compromise - find more true discoveries by allowing for some false discoveries.

Do we need multiple tests correction?

- If false discoveries are really bad, then yes. What if you're comparing multiple new medical treatments to an existing treatment?
- If false nondiscoveries are really bad, then no. What if you're just looking for possible dietary factors that might be linked to breast cancer?
- Compromise - find more true discoveries by allowing for some false discoveries.

- audio04.mp3
- Whenever we try to reduce one kind of error, we'll make more of the other and we'll explore this in what follows.
- In the next few slides we'll define three types of error control.

Per-Comparison Error Control

- PCER = Per Comparison type I Error Rate
- uncorrected testing
- each individual test uses a significance level of α
- probability of Type I error for each test is $\leq \alpha$
- many Type I errors = many false discoveries

Per-Comparison Error Control

- PCER = Per Comparison type I Error Rate
- uncorrected testing
- each individual test uses a significance level of α
- probability of Type I error for each test is $\leq \alpha$
- many Type I errors = many false discoveries

- no audio

Familywise Error Control

- FWER = FamilyWise Error Rate
- control overall rate of Type I error.
- e.g. Bonferonni correction - use a per-comparison significance level of α/m
- guarantees the probability of one or more Type I errors is $\leq \alpha$
- many Type II errors = many false nondiscoveries

Familywise Error Control

- FWER = FamilyWise Error Rate
- control overall rate of Type I error.
- e.g. Bonferonni correction - use a per-comparison significance level of α/m
- guarantees the probability of one or more Type I errors is $\leq \alpha$
- many Type II errors = many false nondiscoveries

- audio05.mp3
- Procedures to control the familywise error rate have been around since the 1950's.
- These procedures tend to be very conservative. Since the focus is on avoiding false discoveries we can end up missing significant effects.

Control False Discovery Rate

- False Discovery Rate = control, on average, FD/D which is the proportion of false discoveries out all discoveries
- in other words, out of all the significant tests we control the fraction that are truly not significant

Control False Discovery Rate

- False Discovery Rate = control, on average, FD/D which is the proportion of false discoveries out all discoveries
- in other words, out of all the significant tests we control the fraction that are truly not significant

- audio06.mp3
- work on the false discovery rate began in 1995 with a paper due to Benjamin and Hochberg.
- Procedures that control the false discovery rate don't tell you the maximum probability of a Type I or a Type II error.
- Instead they say that if we repeated the experiment many times then this is the average proportion of false discoveries out of all discoveries.
- This is exactly the sort of tool we need in big data where we might be sifting through thousands of results to try to find some that are worthy of further exploration.

An Exploratory Example - Setup 1

- We'll examine the three types of error control in R for a synthetic example.
- We'll generate some random data and test each value:
  - H_0 : value is from a normal distribution with $\mu = 0$
  - H_a : value is from a normal distribution with $\mu > 0$

An Exploratory Example - Setup 1

- We'll examine the three types of error control in R for a synthetic example.
- We'll generate some random data and test each value:
  - H_0 : value is from a normal distribution with $\mu = 0$
  - H_a : value is from a normal distribution with $\mu > 0$

- no audio

An Exploratory Example - Setup 2

```
T0 = 900; T1 = 100;
x = c( rnorm(T0), rnorm(100, mean = 3) )
P = pnorm( x, lower.tail = FALSE )
sum( P < 0.05 )
```

```
## [1] 142
```

- Number of discoveries is $D = 142$.

An Exploratory Example - Setup 2

```
T0 = 900; T1 = 100;
x = c( rnorm(T0), rnorm(100, mean = 3) )
P = pnorm( x, lower.tail = FALSE )
sum( P < 0.05 )
```

```
## [1] 142
```

- Number of discoveries is $D = 142$.

- below slide - the file ErrorExperiments.R in the download pack will let you explore this series of experiments yourself.
- audio07.mp3
- our data consists of random observations from two normal distributions.
- 900 are from the standard normal with mean 0 and standard deviation 1. For these 900 the null is true so we ideally we'd get 900 True Nondiscoveries.
- 100 are from a right-shifted normal with mean 3 and standard deviation 1. For these 100 the alternative is true so ideally we'd have 100 True Disoveries.
- To get the P-value for each test, we assume the observation is from the standard normal and compute the right-tail probability. At the 5% significance level we'll

An Exploratory Example - No Corrections 1

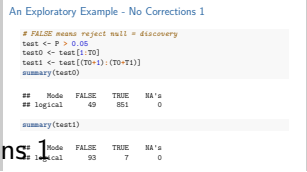
```
# FALSE means reject null = discovery
test <- P > 0.05
test0 <- test[1:T0]
test1 <- test[(T0+1):(T0+T1)]
summary(test0)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      49     851       0
```

```
summary(test1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      93       7       0
```

An Exploratory Example - No Corrections 1



- audio08.mp3
- here is what happens when we use a Per Comparison Error Rate of 5% and we've added no corrections to account for multiple comparisons.
- We'll explain the results in detail on this slide, but for the other methods of error control on the upcoming slides we'll just give shorter summaries.
- the test variable is FALSE for all significant tests
- For the first 900 observations we'd like to see 900 TRUE values indicating that we retain the null hypothesis, but instead we see 49 FALSE values indicating we have 49 significant results that are in error. These are Type I errors or False Discoveries.
- For the last 100 observations we'd liek to see 100 FALSE values indicating that we

An Exploratory Example - No Corrections 2

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

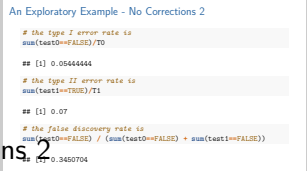
```
## [1] 0.05444444
```

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

```
## [1] 0.07
```

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

```
## [1] 0.3450704
```


An Exploratory Example - No Corrections 2

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

## [1] 0.05444444

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

## [1] 0.07

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

## [1] 0.3450704
```

- audio09.mp3
- the Type I error rate is the number of False Discoveries out of the 900 tests where we knew the null was true.
- Notice that the type I error rate is exactly what we'd expect since we set the significance level to be 5% meaning there is a 5% chance of rejecting a true null due to random variation of the data.
- We have a lot of False Discoveries here with almost 35% of all our discoveries being false, but the Type II error rate is low showing that we've managed to find most of the truly significant results.

Bonferonni Correction for FWER

- Controls FWER.
- Reject H_0 if $P < \alpha/m$
- In R use `p.adjust( P, method = 'bonf')` and compare the adjusted p-values to α.
  - Reject H_0 if $\tilde{p} = mp < \alpha$
- Pros: simple, any hypothesis tests (or CI's)
- Cons: super conservative / low power so that many effects may be missed.

Bonferonni Correction for FWER

- Controls FWER.
- Reject H_0 if $P < \alpha/m$
- In R use `p.adjust( P, method = 'bonf')` and compare the adjusted p-values to α.
  - Reject H_0 if $\tilde{p} = mp < \alpha$
- Pros: simple, any hypothesis tests (or CI's)
- Cons: super conservative / low power so that many effects may be missed.

- audio10.mp3
- The Bonferonni correction is often explained by saying that we compare each individual P value to the corrected significance level alpha over m.
- However, in practice and in software we multiply the original p-values by m and compare these to the family wise error rate alpha. These new p-values are often called adjusted or corrected p-values.
- R uses adjusted P-values.
- On the next couple of slides we'll apply the Bonferonni correction to our 1000 hypothesis tests.

An Exploratory Example - Bonferonni Correction 1

```
# 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)
```

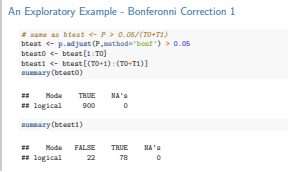
```
##    Mode    TRUE    NA's
## logical     900       0
```

```
summary(btest1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      22      78       0
```

Multiple Testing

An Exploratory Example - Bonferonni Correction 1



```
# 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    NA's
## logical     900       0
```

```
summary(btest1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      22      78       0
```

- no audio

An Exploratory Example - Bonferroni Correction 2

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

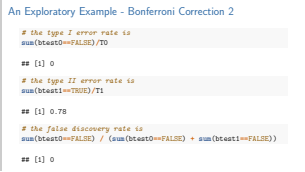
```
## [1] 0
```

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

```
## [1] 0.78
```

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

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


An Exploratory Example - Bonferroni Correction 2

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

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

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

```
## [1] 0.78
```

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

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

- audio11.mp3
- Bonferonni was wildly successful at controlling the Type I error rate, which is now 0, because no false disoveries were made at all. This also makes the false discovery rate 0 as well.
- However, the Type II error rate has skyrocketed as we've now missed a whole bunch of significant results.
- controlling errors is always a balancing act.

Bonferroni-Holm Step-down Procedure for FWER

- sequential correction
- Compare smallest p-value to α/m
- Second smallest p-value to $\alpha/(m-1)$, etc.
- Stop at first non-rejection and do not reject any remaining hypotheses.
- Pros: fairly simple, controls FWER, slightly more power than Bonferroni
- Cons: still conservative, can't use for simultaneous confidence intervals

Always use instead of Bonferonni for multiple hypothesis tests, but stick to Bonferonni if CI's are needed.

Bonferroni-Holm Step-down Procedure for FWER

- sequential correction
- Compare smallest p-value to α/m
- Second smallest p-value to $\alpha/(m-1)$, etc.
- Stop at first non-rejection and do not reject any remaining hypotheses.
- Pros: fairly simple, controls FWER, slightly more power than Bonferroni
- Cons: still conservative, can't use for simultaneous confidence intervals

Always use instead of Bonferonni for multiple hypothesis tests, but stick to Bonferonni if CI's are needed.

- audio11a.mp3
- this is a sequential procedure that reduces the amount of correction to account for the number of tests remaining.
- it's uniformly more powerful than Bonferroni though the difference isn't usually large.
- So always use this for simultaneous hypothesis tests instead of plain Bonferonni. Sequential adjustments don't make sense for confidence intervals so for simultaneous confidence intervals use plain Bonferonni.

An Exploratory Example - Bonferonni-Holm 1

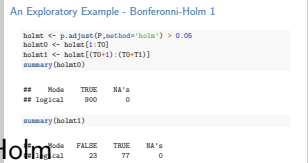
```
holmt <- p.adjust(P,method='holm') > 0.05
holmt0 <- holmt[1:T0]
holmt1 <- holmt[(T0+1):(T0+T1)]
summary(holmt0)
```

```
##    Mode    TRUE    NA's
## logical     900       0
```

```
summary(holmt1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      23      77       0
```

An Exploratory Example - Bonferonni-Holm 1



- no audio

An Exploratory Example - Bonferonni-Holm 2

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

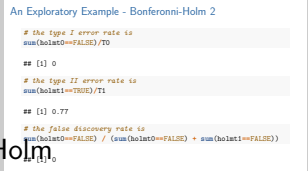
```
## [1] 0
```

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

```
## [1] 0.77
```

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

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


An Exploratory Example - Bonferonni-Holm 2

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

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

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

```
## [1] 0.77
```

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

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

- audio12.mp3
- the only difference between the Bonferroni correction and the Bonferonni-Holm sequential correction is that the number of Type II errors has been reduced by one reflecting the slightly higher power of the sequential procedure.

FDR Control - Benjamin and Hochberg Procedure

- α is now the target average false discovery rate
- use `p.adjust( p, method = 'BH' )` to compute adjusted p-values in R and compare to α
- the adjusted p-values are sometimes called q-values
- details of sequential procedure on Wikipedia
- widely used in genomics and medical imaging with thousands of simultaneous tests

FDR Control - Benjamin and Hochberg Procedure

- α is now the target average false discovery rate
- use `p.adjust( p, method = 'BH' )` to compute adjusted p-values in R and compare to α
- the adjusted p-values are sometimes called q-values
- details of sequential procedure on Wikipedia
- widely used in genomics and medical imaging with thousands of simultaneous tests

- audio13.mp3
- We could show you how to sequentially adjust the P-values, but it wouldn't really give you any insight into why this procedure works.
- You can find the reference to the original 1995 paper on Wikipedia if you want to read more about it.
- Notice that alpha doesn't have anything to do with the probability of Type I errors here, instead it's the desired average False Discovery Rate.

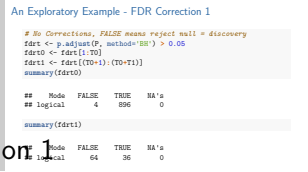
An Exploratory Example - FDR Correction 1

```
# 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    NA's
## logical       4     896       0
```

```
summary(fdrt1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      64      36       0
```



- audio14.mp3
- now we've made 4 False Disoveries or Type I errors which is way less conservative than the Bonferroni correction
- the tradeoff is we've made far fewer Type II errors

An Exploratory Example - FDR Correction 2

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

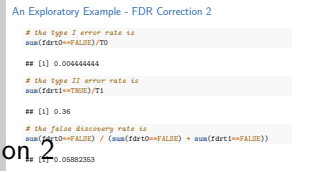
```
## [1] 0.004444444
```

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

```
## [1] 0.36
```

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

```
## [1] 0.05882353
```


An Exploratory Example - FDR Correction 2

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

```
## [1] 0.004444444
```

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

```
## [1] 0.36
```

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

```
## [1] 0.05882353
```

- audio15.mp3
- The Type I error rate is still quite small and we've made far fewer Type II errors.
- Notice that the False Discovery Rate is about 6% meaning of all our significant tests or Discoveries only about 6% are wrong. This is pretty close to the 5% target we set.

An Exploratory Example - FDR Correction 3

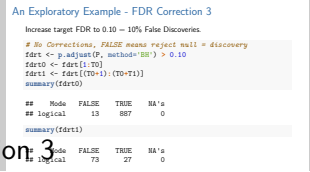
Increase target FDR to 0.10 = 10% False Discoveries.

```
# No Corrections, FALSE means reject null = discovery
fdrt <- p.adjust(P, method='BH') > 0.10
fdrt0 <- fdrt[1:T0]
fdrt1 <- fdrt[(T0+1):(T0+T1)]
summary(fdrt0)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      13     887       0
```

```
summary(fdrt1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      73      27       0
```



An Exploratory Example - FDR Correction 3

Increase target FDR to 0.10 — 10% False Discoveries.

```
# No Corrections, FALSE means reject null = discovery
fdrt <- p.adjust(P, method='BH') > 0.10
fdrt0 <- fdrt[1:T0]
fdrt1 <- fdrt[(T0+1):(T0+T1)]
summary(fdrt0)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      13     887       0
```

```
summary(fdrt1)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      73      27       0
```

-

An Exploratory Example - FDR Correction 4

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

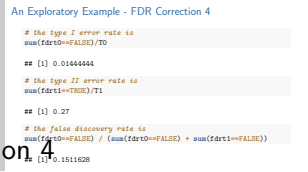
```
## [1] 0.01444444
```

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

```
## [1] 0.27
```

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

```
## [1] 0.1511628
```



- audio15a.mp3
- as the target False Discovery Rate increases notice that we make more discoveries, in this case there are 86 discoveries of which 13 are false
- so our actual FDR is about 15% which is in the neighborhood of our 10% target
- Note that we've decreased the Type II error rate while increasing the Type I error rate. Changing the desired FDR allows us to change the balance between Type I and Type II errors.

Revisiting the Dietary Example

```
# no corrections
rej <- P<.05; rej[1:10]
```

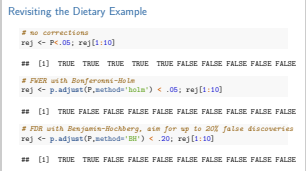
```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```
# FWER with Bonferonni-Holm
rej <- p.adjust(P,method='holm') < .05; rej[1:10]
```

```
##  [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
# FDR with Benjamin-Hochberg, aim for up to 20% false discoveries
rej <- p.adjust(P,method='BH') < .20; rej[1:10]
```

```
##  [1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```



- audio16.mp3 -lets revisit the dietary example from the beginning
- if we do no multiple test correction there appear to be 5 significant results, but since we expect at least 1 or 2 false positives it's hard to tell what's really significant and what isn't
- if we control the family wise error rate with Bonferonni-Holm then there is only 1 significant result. That's great if false discoveries are a problem, but the researchers in this case were really just trying to find possible associations for further research
- so they actually chose to control the false discovery rate with the Benjamin Hochberg procedure and found two significant associations. To be accurate here, I'm not actually sure what target false discovery rate they used.

Bonferroni Correction for CI's

- For 4 simultaneous CI's.
- want familywise error rate $\alpha_E = 0.05$
- familywise confidence level $1 - \alpha_E = 0.95$
- individual comparison error rate $\alpha_I = 0.05/4 = 0.0125$
- individual comparison confidence level $1 - \alpha_I = 0.9875$

Generally: familywise confidence level $1 - \alpha$ use individual confidence level $1 - \alpha/m$.

Bonferroni Correction for CI's

- For 4 simultaneous CI's.
  - want familywise error rate $\alpha_E = 0.05$
  - familywise confidence level $1 - \alpha_E = 0.95$
  - individual comparison error rate $\alpha_I = 0.05/4 = 0.0125$
  - individual comparison confidence level $1 - \alpha_I = 0.9875$

Generally: familywise confidence level $1 - \alpha$ use individual confidence level $1 - \alpha/m$.

- audio17.mp3
- we'll see an example of using this later to estimate differences between population means, but this correction can be used for any family of simulataneous confidence intervals which is why we've introduced it here.
- if we have overall 95% confidence level for a whole family of intervals we can say that we are 95% confident that the collection of intervals doesn't contain any intervals that fail to contain the estimated parameter

Which kind of Error Control?

Type I Error = False Discovery, Type II Error = False Nondiscovery

Error Control	Type I Errors	Type II Errors	When to use
PCER	Many	Few	When it's important not to miss any discoveries. Exploratory Only.
FWER	Very few	Many	When false discoveries are bad and need to be controlled.
FDR	Few	Controlled	Exploratory Analysis. Don't want to miss discoveries while keeping false discoveries controlled.

Which kind of Error Control?

Type I Error = False Discovery, Type II Error = False Nondiscovery

Error Control	Type I Errors	Type II Errors	When to use
PCER	Many	Few	When it's important not to miss any discoveries. Exploratory Only.
FWER	Very few	Many	When false discoveries are bad and need to be controlled.
FDR	Few	Controlled	Exploratory Analysis. Don't want to miss discoveries while keeping false discoveries controlled.

- audio18.mp3
- The kind of error control you choose for multiple tests really depends on the application
- If you're comparing several new but expensive medical treatments to an existing one, it might make sense to control the family wise error rate to avoid making a potentially expensive Type I error.
- If you're looking for associations between one variable and many others and you plan to do further research into the significant associations then using FDR or possibly no corrections at all makes sense.
- If you've got thousands or even tens of thousands of tests then you really have to do something like FDR control so that you have a reasonable chance of discovering

Comparing Population Means

- The methods above apply to any family of hypothesis tests:
  - Simultanous t-tests for some effect
  - Testing multiple correlations
  - Testing multiple regression coefficients
  - many others
- Below we study comparing multiple population means
  - Multiple two-sample t-tests for each pair of means.
  - Tukey-Kramer test for pairwise means comparison.

Comparing Population Means

- The methods above apply to any family of hypothesis tests:
  - Simultanous t-tests for some effect
  - Testing multiple correlations
  - Testing multiple regression coefficients
  - many others
- Below we study comparing multiple population means
  - Multiple two-sample t-tests for each pair of means.
  - Tukey-Kramer test for pairwise means comparison.

-audio19.mp3 - the error control procedures we've met so far apply to any family of hypothesis tests, or in the case of Bonferonni we can also correct the confidence levels for simultaneous confidence intervals - in what follows we'll focus on the problem of comparing multiple population means

ANOVA and Kruskal-Wallis Tests

- Reject H_0 and conclude there is at least one significant difference between means, but which?
- Find pairwise differences: multiple pairwise tests with error control or use a specialized procedure like Tukey-Kramer.
- Good idea to do ANOVA first especially for controlling Type I errors, but not required.

ANOVA and Kruskal-Wallis Tests

- Reject H_0 and conclude there is at least one significant difference between means, but which?
- Find pairwise differences: multiple pairwise tests with error control or use a specialized procedure like Tukey-Kramer.
- Good idea to do ANOVA first especially for controlling Type I errors, but not required.

- audio20.mp3
- It's conventional to start with an ANOVA to see if there are significant differences among the means, but all an ANOVA (or Kruskal-Wallis) can tell us is that there is at least one difference, but we won't know hwere it is.
- So we usually follow up with multiple pairwise tests or pairwise confidence interval estimates of the differences in means which requires us to think about error control.
- If you're not worried about Type I errors you really don't even need to do ANOVA to begin with and can jump straight to the pairwise comparisons.

Comparing Multiple Population Means

- How many pairs of means?

k means	$m = \frac{k(k-1)}{2}$ pairs
3	3
4	6
5	10
6	15
7	21
$\vdots$	$\vdots$

The number grows quadratically with the number of means.

Comparing Multiple Population Means

- How many pairs of means?

k means	$m = \frac{k(k-1)}{2}$ pairs
3	3
4	6
5	10
6	15
7	21
⋮	⋮

The number grows quadratically with the number of means.

- no audio

Testing for significant differences among means

- PCER control
  - pairwise t-tests with no correction
- FWER control
  - pairwise t-tests with Bonferonni-Holm
  - "one-step" procedure like Tukey-Kramer which is more powerful than Bonferonni
- FDR control
  - pairwise t-tests with Benjamin-Hochberg

Any of these methods can be bootstrapped if the conditions aren't met.

Testing for significant differences among means

- PCER control
  - pairwise t-tests with no correction
- FWER control
  - pairwise t-tests with Bonferonni-Holm
  - "one-step" procedure like Tukey-Kramer which is more powerful than Bonferonni
- FDR control
  - pairwise t-tests with Benjamin-Hochberg

Any of these methods can be bootstrapped if the conditions aren't met.

- audio21.mp3
- these are just a few of the many different procedures which have been developed for comparing population means, but these are some of the main ones that are used in practice.
- the pairwise t-tests we're talking about here are essentially the same as the independent samples t-test for the difference of two means that you learned earlier in the course.
- All of these are based on doing one t-test or t-interval for each pair of means except for the Tukey-Kramer procedure which uses a t-test statistic that is based on the maximum difference between groups so the requirements are similar to those of the t-test.

Requirements for pairwise t-tests

- Similar to independent samples t-test.
- Requires normal distributions or each sample size ≥ 30.
- Generally use unequal variances tests without checking for equal variances.
- If samples are small it can be helpful to use equal variances versions of tests as long as the samples have comparable variances.
- If the distributions really non-normal and the samples aren't too small, then bootstrap the t-tests using onewayComp() from DS705data package.

Requirements for pairwise t-tests
- Similar to independent samples t-test.
- Requires normal distributions or each sample size ≥ 30.
- Generally use unequal variances tests without checking for equal variances.
- If samples are small it can be helpful to use equal variances versions of tests as long as the samples have comparable variances.
- If the distributions really non-normal and the samples aren't too small, then bootstrap the t-tests using onewayComp() from DS705data package.

- no audio

Morphine Tolerance Example

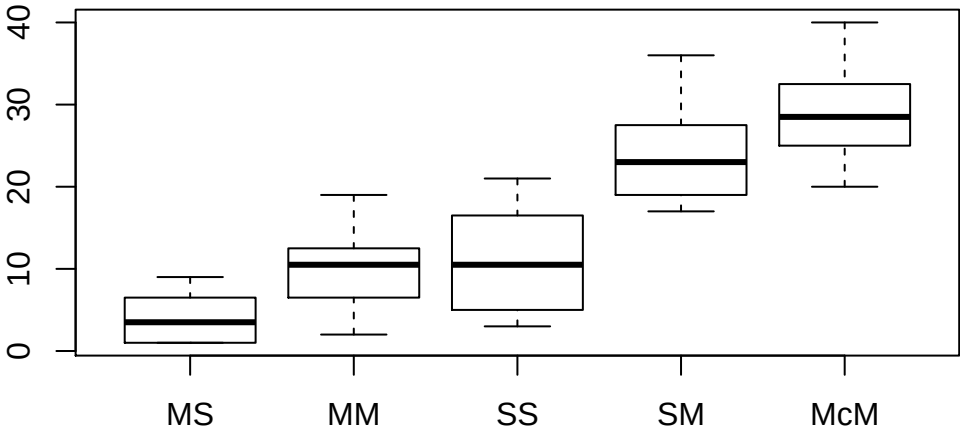
- Record pain sensitivity after rats developed morphine tolerance
- 5 treatment groups: MS, MM, SS, SM, McM
- example found in David Howell's book: *Statistical Methods for Psychology* - Chapter 12 (included with download)
- original study: Siegel, Shepard. "Evidence from rats that morphine tolerance is a learned response." Journal of comparative and physiological psychology 89.5 (1975): 498.

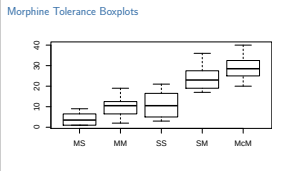
Morphine Tolerance Example

- Record pain sensitivity after rats developed morphine tolerance
- 5 treatment groups: MS, MM, SS, SM, McM
- example found in David Howell's book: *Statistical Methods for Psychology* - Chapter 12 (included with download)
- original study: Siegel, Shepard. "Evidence from rats that morphine tolerance is a learned response." Journal of comparative and physiological psychology 89.5 (1975): 498.

- audio22.mp3
- The details of the experiment aren't important for the purpose of demonstrating our statistical procedures, but domain knowledge is always important for data scientist.
- If you are are curious, the M's are for morphine and S's for Saline, so MS is morphine followed by Saline etc.

Morphine Tolerance Boxplots





- audio23.mp3
- based on the boxplot it seems reasonable to say that the samples come from normal distributions with similar variances.
- for our pairwise t-tests we'll go ahead an use the equal variances assumption since the slight boost in power might be helpful to compensate for the small sample size of 8 for each sample.
- visually we can see that some of the samples are shifted from relative to the others so we'll likely see some different means

Morphine Tolerance - PCER 1

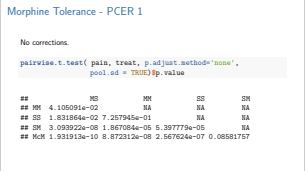
No corrections.

```
pairwise.t.test( pain, treat, p.adjust.method='none',
                 pool.sd = TRUE)$p.value
```

```
##                MS           MM           SS          SM
## MM  4.105091e-02           NA           NA          NA
## SS  1.831864e-02 7.257945e-01           NA          NA
## SM  3.093922e-08 1.867084e-05 5.397779e-05          NA
## McM 1.931913e-10 8.872312e-08 2.567624e-07 0.08581757
```

Multiple Testing

Morphine Tolerance - PCER 1



Morphine Tolerance - PCER 1

No corrections.

```
pairwise.t.test( pain, treat, p.adjust.method='none',
                 pool.sd = TRUE)$p.value
```

```
##              MS           MM           SS           SM
## MM  4.105091e-02           NA           NA           NA
## SS  1.831864e-02 7.257945e-01           NA           NA
## SM  3.093922e-06 1.867084e-05 5.397779e-05           NA
## MoM 1.931913e-10 8.872312e-08 2.567624e-07 0.08581757
```

- no audio

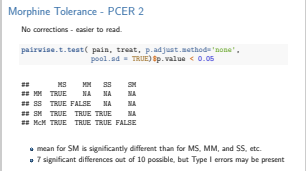
Morphine Tolerance - PCER 2

No corrections - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='none',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##        MS    MM    SS    SM
## MM   TRUE    NA    NA    NA
## SS   TRUE FALSE    NA    NA
## SM   TRUE  TRUE  TRUE    NA
## McM  TRUE  TRUE  TRUE FALSE
```

- mean for SM is significantly different than for MS, MM, and SS, etc.
- 7 significant differences out of 10 possible, but Type I errors may be present


Morphine Tolerance - PCER 2

No corrections - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='none',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##       MS    MM    SS    SM
## MM  TRUE    NA    NA    NA
## SS  TRUE FALSE    NA    NA
## SM  TRUE  TRUE  TRUE    NA
## McM TRUE  TRUE  TRUE FALSE
```

- mean for SM is significantly different than for MS, MM, and SS, etc.
- 7 significant differences out of 10 possible, but Type I errors may be present

- audio24.mp3
- it's easier to read the output if we use the 5% threshold to return TRUE if we should reject the hypothesis that the pair of means is the same
- these are two-tailed tests unless otherwise specified.
- for reporting these results in homework you should should write a summary that describes which pairs of means are different

Morphine Tolerance - PCER 3

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='none')$comp[,c(2,3,5,6)]
```

```
##                lwr       upr            p rej H_0
## MM-MS    0.2579877 11.742012 4.105091e-02       1
## SS-MS    1.2579877 12.742012 1.831864e-02       1
## SM-MS   14.2579877 25.742012 3.093922e-08       1
## McM-MS 19.2579877 30.742012 1.931913e-10       1
## SS-MM  -4.7420123  6.742012 7.257945e-01       0
## SM-MM   8.2579877 19.742012 1.867084e-05       1
## McM-MM 13.2579877 24.742012 8.872312e-08       1
## SM-SS   7.2579877 18.742012 5.397779e-05       1
## McM-SS 12.2579877 23.742012 2.567624e-07       1
## McM-SM -0.7420123 10.742012 8.581757e-02       0
```

Morphine Tolerance - PCER 3

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='none')$comp[,c(2,3,5,6)]
```

```
##              lwr       upr            p rej H_0
## MM-MS   0.2579877 11.742012 4.105091e-02   1
## SS-MS   1.2579877 12.742012 1.831864e-02   1
## SM-MS  14.2579877 25.742012 3.093922e-08   1
## McM-MS 19.2579877 30.742012 1.931913e-10   1
## SS-MM  -4.7420123  6.742012 7.257945e-01   0
## SM-MM   8.2579877 19.742012 1.867084e-05   1
## McM-MM 13.2579877 24.742012 8.872312e-08   1
## SM-SS   7.2579877 18.742012 5.397779e-05   1
## McM-SS 12.2579877 23.742012 2.567624e-07   1
## McM-SM -0.7420123 10.742012 8.581757e-02   0
```

- audio25.mp3
- to get the onewayComp function you'll have to load the DS705data package
- this function will do a lot and you can look at the help page for the function to learn more
- we're displaying some of the columns of output to keep this readable. You can see that we asked for columns 2,3,5 and 6 to be displayed.
- in this case it includes a 95% confidence interval for the difference of each pair of means, these confidence levels haven't been corrected to allow for multiple comparison

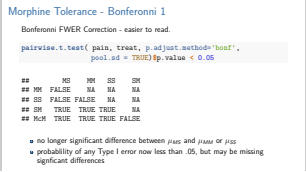
Morphine Tolerance - Bonferonni 1

Bonferonni FWER Correction - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='bonf',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##         MS    MM    SS    SM
## MM   FALSE    NA    NA    NA
## SS   FALSE FALSE    NA    NA
## SM    TRUE  TRUE TRUE    NA
## McM   TRUE  TRUE TRUE FALSE
```

- no longer significant difference between μ_{MS} and μ_{MM} or μ_{SS}
- probablility of any Type I error now less than .05, but may be missing signficant differences


Morphine Tolerance - Bonferonni 1

Bonferonni FWER Correction - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='bonf',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##        MS    MM    SS    SM
## MM  FALSE    NA    NA    NA
## SS  FALSE FALSE    NA    NA
## SM   TRUE  TRUE  TRUE    NA
## McM  TRUE  TRUE  TRUE FALSE
```

- no longer significant difference between μ_{MS} and μ_{MM} or μ_{SS}
- probability of any Type I error now less than .05, but may be missing significant differences

- no audio

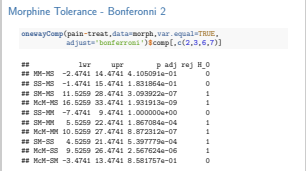
Morphine Tolerance - Bonferonni 2

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='bonferroni')$comp[,c(2,3,6,7)]
```

```
##             lwr     upr         p adj rej H_0
## MM-MS  -2.4741 14.4741 4.105091e-01       0
## SS-MS  -1.4741 15.4741 1.831864e-01       0
## SM-MS  11.5259 28.4741 3.093922e-07       1
## McM-MS 16.5259 33.4741 1.931913e-09       1
## SS-MM  -7.4741  9.4741 1.000000e+00       0
## SM-MM   5.5259 22.4741 1.867084e-04       1
## McM-MM 10.5259 27.4741 8.872312e-07       1
## SM-SS   4.5259 21.4741 5.397779e-04       1
## McM-SS  9.5259 26.4741 2.567624e-06       1
## McM-SM -3.4741 13.4741 8.581757e-01       0
```

Multiple Testing

Morphine Tolerance - Bonferonni 2



Morphine Tolerance - Bonferonni 2

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='bonferroni')$comp[,c(2,3,6,7)]
```

```
##            lwr      upr         p adj rej H_0
## MM-MS  -2.4741 14.4741 4.105091e-01       0
## SS-MS  -1.4741 15.4741 1.531864e-01       0
## SM-MS  11.5259 28.4741 3.093922e-07       1
## McM-MS 16.5259 33.4741 1.931913e-09       1
## SS-MM  -7.4741  9.4741 1.000000e+00       0
## SM-MM   5.5259 22.4741 1.867084e-04       1
## McM-MM 10.5259 27.4741 8.872312e-07       1
## SM-SS   4.5259 21.4741 5.397779e-04       1
## McM-SS  9.5259 26.4741 2.567624e-06       1
## McM-SM -3.4741 13.4741 8.581757e-01       0
```

- no audio

Morphine Tolerance - Bonferonni 3

Simultaneous Hypothesis Test Conclusion: At the 5% significance level we can say the population mean pain sensitivities for SM and MCM are different than those of MS, MM, and SS. There are no other significant differences.

Family of confidence interval interpretation: We are 95% confident that the population mean pain sensitivity is greater for SM than for MS, MM, and SS by 11.5 to 28.5, 5.5 to 22.5, and 4.5 to 21.5, respectively. While the pain sensitity for McM is greater than for MS, MM, and SS by 16.5 to 33.5, 10.5 to 27.5, and 9.5 to 26.5, respectively.

Morphine Tolerance - Bonferonni 3

Simultaneous Hypothesis Test Conclusion: At the 5% significance level we can say the population mean pain sensitivities for SM and MCM are different than those of MS, MM, and SS. There are no other significant differences.

Family of confidence interval interpretation: We are 95% confident that the population mean pain sensitivity is greater for SM than for MS, MM, and SS by 11.5 to 28.5, 5.5 to 22.5, and 4.5 to 21.5, respectively. While the pain sensitity for McM is greater than for MS, MM, and SS by 16.5 to 33.5, 10.5 to 27.5, and 9.5 to 26.5, respectively.

- audio26.mp3
- since we are controlling the familywise error rate here we can assert a conclusion or interpretation about the whole family of comparison in much the same way as we would for a single test or confidence interval.
- if we are not using corrections or are controlling the false discovery rate, then we need to explain that with our conlusions as well.

Morphine Tolerance - Bonferonni-Holm 1

Bonferonni-Holm FWER Correction - easier to read.

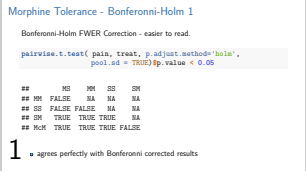
```
pairwise.t.test( pain, treat, p.adjust.method='holm',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##         MS    MM    SS    SM
## MM   FALSE    NA    NA    NA
## SS   FALSE FALSE    NA    NA
## SM    TRUE  TRUE TRUE    NA
## McM   TRUE  TRUE TRUE FALSE
```

- agrees perfectly with Bonferonni corrected results

Multiple Testing

Morphine Tolerance - Bonferonni-Holm 1



Morphine Tolerance - Bonferonni-Holm 1

Bonferonni-Holm FWER Correction - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='holm',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##       MS    MM    SS    SM
## MM  FALSE    NA    NA    NA
## SS  FALSE FALSE    NA    NA
## SM   TRUE  TRUE  TRUE    NA
## McM  TRUE  TRUE  TRUE FALSE
```

- agrees perfectly with Bonferonni corrected results

- no audio

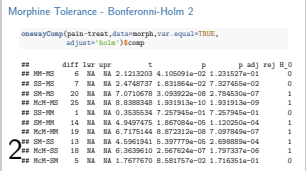
Morphine Tolerance - Bonferonni-Holm 2

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='holm')$comp
```

```
##         diff lwr upr         t             p         p adj rej H_0
## MM-MS      6  NA  NA 2.1213203 4.105091e-02 1.231527e-01       0
## SS-MS      7  NA  NA 2.4748737 1.831864e-02 7.327455e-02       0
## SM-MS     20  NA  NA 7.0710678 3.093922e-08 2.784530e-07       1
## McM-MS    25  NA  NA 8.8388348 1.931913e-10 1.931913e-09       1
## SS-MM      1  NA  NA 0.3535534 7.257945e-01 7.257945e-01       0
## SM-MM     14  NA  NA 4.9497475 1.867084e-05 1.120250e-04       1
## McM-MM    19  NA  NA 6.7175144 8.872312e-08 7.097849e-07       1
## SM-SS     13  NA  NA 4.5961941 5.397779e-05 2.698889e-04       1
## McM-SS    18  NA  NA 6.3639610 2.567624e-07 1.797337e-06       1
## McM-SM     5  NA  NA 1.7677670 8.581757e-02 1.716351e-01       0
```

Multiple Testing

Morphine Tolerance - Bonferonni-Holm 2



Morphine Tolerance - Bonferonni-Holm 2

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='holm')$comp
```

```
##        diff lwr upr           t           p         p adj rej H_0
## MM-MS     6  NA  NA 2.1213203 4.105091e-02 1.231527e-01       0
## SS-MS     7  NA  NA 2.4748737 1.831864e-02 7.327455e-02       0
## SM-MS    20  NA  NA 7.0710678 3.093922e-08 2.784530e-07       1
## McM-MS   25  NA  NA 8.8388348 1.931913e-10 1.931913e-09       1
## SS-MM     1  NA  NA 0.3535534 7.257945e-01 7.257945e-01       0
## SM-MM    14  NA  NA 4.9497475 1.867084e-05 1.120250e-04       1
## McM-MM   19  NA  NA 6.7175144 8.872312e-08 7.097849e-07       1
## SM-SS    13  NA  NA 4.5961941 5.397779e-05 2.698889e-04       1
## McM-SS   18  NA  NA 6.3639610 2.567624e-07 1.797337e-06       1
## McM-SM    5  NA  NA 1.7677670 8.581757e-02 1.716351e-01       0
```

- no audio

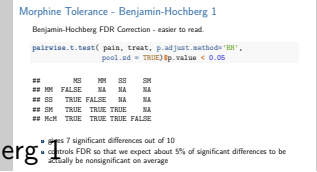
Morphine Tolerance - Benjamin-Hochberg 1

Benjamin-Hochberg FDR Correction - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='BH',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##         MS    MM    SS    SM
## MM   FALSE    NA    NA    NA
## SS    TRUE FALSE    NA    NA
## SM    TRUE  TRUE  TRUE    NA
## McM   TRUE  TRUE  TRUE FALSE
```

- gives 7 significant differences out of 10
- controls FDR so that we expect about 5% of significant differences to be actually be nonsignificant on average


Morphine Tolerance - Benjamin-Hochberg 1

Benjamin-Hochberg FDR Correction - easier to read.

```
pairwise.t.test( pain, treat, p.adjust.method='BH',
                 pool.sd = TRUE)$p.value < 0.05
```

```
##        MS    MM    SS    SM
## MM  FALSE    NA    NA    NA
## SS   TRUE FALSE    NA    NA
## SM   TRUE  TRUE  TRUE    NA
## McM  TRUE  TRUE  TRUE FALSE
```

- gives 7 significant differences out of 10
- controls FDR so that we expect about 5% of significant differences to be actually be nonsignificant on average

- audio27.mp3
- When you explain these results you should no longer claim that these differences are significant at the 5% significance level
- Rather you should say something like we've discovered 7 significant differences by a method which gets about 5% of significant differences wrong on average.

Morphine Tolerance - Benjamin-Hochberg 2

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='BH')$comp
```

```
##         diff lwr upr         t             p         p adj
## MM-MS      6  NA  NA 2.1213203 4.105091e-02 5.131363e-02
## SS-MS      7  NA  NA 2.4748737 1.831864e-02 2.616948e-02
## SM-MS     20  NA  NA 7.0710678 3.093922e-08 1.546961e-07
## McM-MS    25  NA  NA 8.8388348 1.931913e-10 1.931913e-09
## SS-MM      1  NA  NA 0.3535534 7.257945e-01 7.257945e-01
## SM-MM     14  NA  NA 4.9497475 1.867084e-05 3.734167e-05
## McM-MM    19  NA  NA 6.7175144 8.872312e-08 2.957437e-07
## SM-SS     13  NA  NA 4.5961941 5.397779e-05 8.996298e-05
## McM-SS    18  NA  NA 6.3639610 2.567624e-07 6.419060e-07
## McM-SM     5  NA  NA 1.7677670 8.581757e-02 9.535286e-02
```

Multiple Testing

Morphine Tolerance - Benjamin-Hochberg 2

Morphine Tolerance - Benjamin-Hochberg 2

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='BH')$comp
```

```
##        diff lwr upr          t            p        p adj
## MM-MS     6  NA  NA 2.1213203 4.105091e-02 5.131363e-02
## SS-MS     7  NA  NA 2.4748737 1.831864e-02 2.616948e-02
## SM-MS    20  NA  NA 7.0710678 3.093922e-08 1.546961e-07
## McM-MS   25  NA  NA 8.8388348 1.931913e-10 1.931913e-09
## SS-MM     1  NA  NA 0.3535534 7.257945e-01 7.257945e-01
## SM-MM    14  NA  NA 4.9497475 1.867084e-05 3.734167e-05
## McM-MM   19  NA  NA 6.7175144 8.872312e-08 2.957437e-07
## SM-SS    13  NA  NA 4.5961941 5.397779e-05 8.996298e-05
## McM-SS   18  NA  NA 6.3639610 2.567624e-07 6.419060e-07
## McM-SM    5  NA  NA 1.7677670 8.581757e-02 9.535286e-02
```

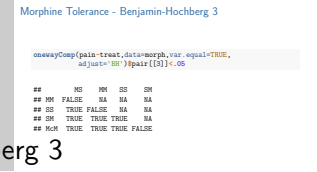
- no audio

Morphine Tolerance - Benjamin-Hochberg 3

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='BH')$pair[[3]]<.05
```

```
##         MS    MM    SS    SM
## MM   FALSE    NA    NA    NA
## SS    TRUE FALSE    NA    NA
## SM    TRUE  TRUE  TRUE    NA
## McM   TRUE  TRUE  TRUE FALSE
```

Morphine Tolerance - Benjamin-Hochberg 3



- audio28.mp3
- we just threw in this slide so you can see how you can get the matrix of p-values from the onewayComp function.

Less Conservative FWER Control for Comparing Means

- Bonferroni correction usually overly conservative possibly producing many Type II errors
- Less conservative for samples from normal distributions use Tukey-Kramer or Games-Howell.
  - more details on next 3 slides (also page 460 in Ott)
  - sometimes called one-step procedures since they compare all pairs at once
  - preferred to Bonferonni for FWER control when applicable
  - in R use TukeyHSD() or onewayComp()

Less Conservative FWER Control for Comparing Means

- Bonferroni correction usually overly conservative possibly producing many Type II errors
- Less conservative for samples from normal distributions use Tukey-Kramer or Games-Howell
  - more details on next 3 slides (also page 460 in Ott)
  - sometimes called one-step procedures since they compare all pairs at once
  - preferred to Bonferonni for FWER control when applicable
  - in R use TukeyHSD() or onewayComp()

- audio29.mp3
- Tukey-Kramer is a widely used procedure but needs the samples to be from normal distributions, to have approximately equal variances, and similar sample sizes. Games-Howell is less well known, but uses a Welch-like correction to allow for unequal variances and unbalanced sample sizes.
- Even when the variances and sample sizes are similar the Games-Howell procedure usually produces results that are very close to those from Tukey-Kramer, but since Games-Howell isn't as widely known it's probably best to use Tukey-Kramer when possible.
- In the onewayComp function just set the var.equal argument to FALSE to use Games-Howell.

Tukey-Kramer

$$\overline{x}_i - \overline{x}_j \pm q_{\text{crit}} s_p \sqrt{\frac{\frac{1}{n_i} + \frac{1}{n_j}}{2}}$$

q_{crit} is the upper-tail critical value of the Studentized range distribution (`qtukey()` in R).

$s_p = \sqrt{MSE}, df = N - k$

exact control of FWER if samples balanced and population variances equal.

Multiple Testing

Tukey-Kramer

Tukey-Kramer

$$\bar{x}_i - \bar{x}_j \pm q_{\text{crit}} s_p \sqrt{\frac{\frac{1}{n_i} + \frac{1}{n_j}}{2}}$$

q_{crit} is the upper-tail critical value of the Studentized range distribution (`qtukey()` in R).

$s_p = \sqrt{MSE}$, $df = N - k$

exact control of FWER if samples balanced and population variances equal.

no audio

Games-Howell

$$\overline{x}_i - \overline{x}_j \pm q_{\text{crit}} \sqrt{\frac{\frac{s_i^2}{n_i} + \frac{s_j^2}{n_j}}{2}}$$

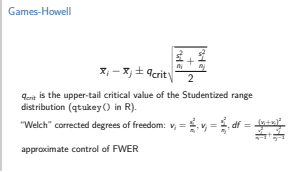
q_{crit} is the upper-tail critical value of the Studentized range distribution (`qtukey()` in R).

"Welch" corrected degrees of freedom: $v_i = \frac{s_i^2}{n_i}$, $v_j = \frac{s_j^2}{n_j}$, $df = \frac{(v_i + v_j)^2}{\frac{v_i^2}{n_i - 1} + \frac{v_j^2}{n_j - 1}}$

approximate control of FWER

Multiple Testing

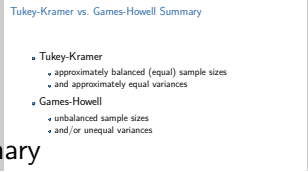
Games-Howell



no audio

Tukey-Kramer vs. Games-Howell Summary

- Tukey-Kramer
  - approximately balanced (equal) sample sizes
  - and approximately equal variances
- Games-Howell
  - unbalanced sample sizes
  - and/or unequal variances


Tukey-Kramer vs. Games-Howell Summary
Tukey-Kramer
approximately balanced (equal) sample sizes
and approximately equal variances
Games-Howell
unbalanced sample sizes
and/or unequal variances

- no audio

Morphine Tolerance - Tukey-Kramer 1

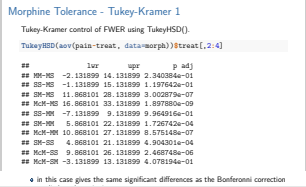
Tukey-Kramer control of FWER using TukeyHSD().

```
TukeyHSD(aov(pain~treat, data=morph))$treat[,2:4]
```

```
##                lwr       upr        p adj
## MM-MS  -2.131899 14.131899 2.340384e-01
## SS-MS  -1.131899 15.131899 1.197642e-01
## SM-MS  11.868101 28.131899 3.002879e-07
## McM-MS 16.868101 33.131899 1.897880e-09
## SS-MM  -7.131899  9.131899 9.964916e-01
## SM-MM   5.868101 22.131899 1.726742e-04
## McM-MM 10.868101 27.131899 8.575148e-07
## SM-SS   4.868101 21.131899 4.904301e-04
## McM-SS  9.868101 26.131899 2.468748e-06
## McM-SM -3.131899 13.131899 4.078194e-01
```

Multiple Testing

Morphine Tolerance - Tukey-Kramer 1



Morphine Tolerance - Tukey-Kramer 1

Tukey-Kramer control of FWER using TukeyHSD().

```
TukeyHSD(aov(pain~treat, data=morph))$treat[,2:4]
```

```
##              lwr       upr        p adj
## MM-MS  -2.131899 14.131899 2.340384e-01
## SS-MS  -1.131899 15.131899 1.197642e-01
## SM-MS  11.868101 28.131899 3.002879e-07
## McM-MS 16.868101 33.131899 1.897880e-09
## SS-MM  -7.131899  9.131899 9.964916e-01
## SM-MM   5.868101 22.131899 1.726742e-04
## McM-MM 10.868101 27.131899 8.575148e-07
## SM-SS   4.868101 21.131899 4.904301e-04
## McM-SS  9.868101 26.131899 2.468748e-06
## McM-SM -3.131899 13.131899 4.078194e-01
```

- in this case gives the same significant differences as the Bonferonni correction

- no audio

Morphine Tolerance - Tukey-Kramer 2

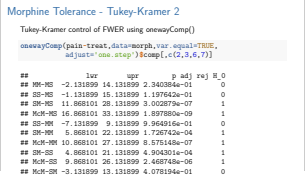
Tukey-Kramer control of FWER using onewayComp()

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='one.step')$comp[,c(2,3,6,7)]
```

```
##               lwr       upr        p adj rej H_0
## MM-MS  -2.131899 14.131899 2.340384e-01       0
## SS-MS  -1.131899 15.131899 1.197642e-01       0
## SM-MS  11.868101 28.131899 3.002879e-07       1
## McM-MS 16.868101 33.131899 1.897880e-09       1
## SS-MM  -7.131899  9.131899 9.964916e-01       0
## SM-MM   5.868101 22.131899 1.726742e-04       1
## McM-MM 10.868101 27.131899 8.575148e-07       1
## SM-SS   4.868101 21.131899 4.904301e-04       1
## McM-SS  9.868101 26.131899 2.468748e-06       1
```

Multiple Testing

Morphine Tolerance - Tukey-Kramer 2



Morphine Tolerance - Tukey-Kramer 2

Tukey-Kramer control of FWER using onewayComp()

```
onewayComp(pain~treat,data=morph,var.equal=TRUE,
           adjust='one.step')$comp[,c(2,3,6,7)]
```

```
##              lwr       upr          p adj rej H_0
## MM-MS  -2.131899 14.131899 2.340384e-01       0
## SS-MS  -1.131899 15.131899 1.197642e-01       0
## SM-MS  11.868101 28.131899 3.002879e-07       1
## McM-MS 16.868101 33.131899 1.897880e-09       1
## SS-MM  -7.131899  9.131899 9.964916e-01       0
## SM-MM   5.868101 22.131899 1.726742e-04       1
## McM-MM 10.868101 27.131899 8.575148e-07       1
## SM-SS   6.868101 21.131899 4.904301e-04       1
## McM-SS  9.868101 26.131899 2.468748e-06       1
## McM-SM -3.131899 13.131899 4.078194e-01       0
```

- no audio

Morphine Tolerance - Games-Howell

Games-Howell control of FWER using onewayComp()

```
onewayComp(pain~treat,data=morph,var.equal=FALSE,
           adjust='one.step')$comp[,c(2,3,6,7)]
```

```
##                lwr      upr        p adj rej H_0
## MM-MS   -0.8206702 12.82067 5.709969e-02       0
## SS-MS   -1.6478242 15.64782 7.996763e-02       0
## SM-MS   11.7646823 28.23532 2.289236e-08       1
## McM-MS 17.0045033 32.99550 4.880729e-11       1
## SS-MM   -8.3992351 10.39924 9.971646e-01       0
## SM-MM    4.9361870 23.06381 2.371557e-04       1
## McM-MM 10.1272247 27.87278 8.965799e-07       1
## SM-SS    2.7967438 23.20326 2.940156e-03       1
## McM-SS   7.9447706 28.05523 2.585832e-05       1
```

Morphine Tolerance - Games-Howell

Games-Howell control of FWER using onewayComp()

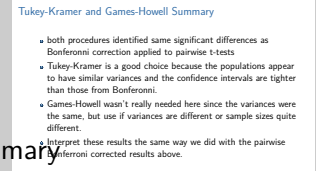
```
onewayComp(pain~treat,data=morph,var.equal=FALSE,
           adjust='one.step')$comp[,c(2,3,6,7)]
```

```
##              lwr       upr          p adj rej H_0
## MM-MS  -0.8206702 12.820267 5.709969e-02       0
## SS-MS  -1.6478242 15.647824 7.996763e-02       0
## SM-MS  11.7646823 28.235323 2.289236e-08       1
## McM-MS 17.0045033 32.995550 4.880729e-11       1
## SS-MM  -8.3992351 10.399924 9.971646e-01       0
## SM-MM   4.9361870 23.063813 2.371557e-04       1
## McM-MM 10.1272247 27.872778 8.965799e-07       1
## SM-SS   2.7967438 23.203256 2.940156e-03       1
## McM-SS  7.9447706 28.055223 2.588832e-05       1
## McM-SM -4.7665239 14.766552 5.101111e-01       0
```

- no audio

Tukey-Kramer and Games-Howell Summary

- both procedures identified same significant differences as Bonferonni correction applied to pairwise t-tests
- Tukey-Kramer is a good choice because the populations appear to have similar variances and the confidence intervals are tighter than those from Bonferonni.
- Games-Howell wasn't really needed here since the variances were the same, but use if variances are different or sample sizes quite different.
- Interpret these results the same way we did with the pairwise Bonferroni corrected results above.


Tukey-Kramer and Games-Howell Summary
both procedures identified same significant differences as Bonferonni correction applied to pairwise t-tests
Tukey-Kramer is a good choice because the populations appear to have similar variances and the confidence intervals are tighter than those from Bonferonni.
Games-Howell wasn't really needed here since the variances were the same, but use if variances are different or sample sizes quite different.
Interpret these results the same way we did with the pairwise Bonferroni corrected results above.

- no audio

Bootstrapping for pairwise means

- Any of the procedures in onewayComp() can be bootstrapped, by specifying nboot $= 1000$ say.
- Good to do if populations clearly aren't normally distributed.

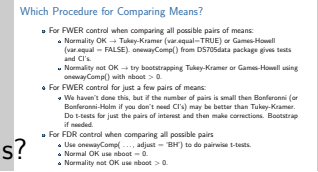
Bootstrapping for pairwise means

- Any of the procedures in onewayComp() can be bootstrapped, by specifying nboot = 1000 say.
- Good to do if populations clearly aren't normally distributed.

- audio30.mp3
- Some people like to use bootstrapping to validate results.
- Say you've elected to use Tukey-Kramer becaue the samples appear to come from normal distributions, etc.
- Use onewayComp to compute the results and the use onewayComp again with nboot = 5000 say to get bootstrapped results. If the results are similar then great, but if the results are quite different perhaps you should explore a bit further to see if the conditions are really met.
- If it's clear that the samples aren't from normal distributions, then using bootstrapping is a good thing to try.

Which Procedure for Comparing Means?

- For FWER control when comparing all possible pairs of means:
  - Normality OK $\rightarrow$ Tukey-Kramer (var.equal=TRUE) or Games-Howell (var.equal = FALSE). onewayComp() from DS705data package gives tests and CI's.
  - Normality not OK $\rightarrow$ try bootstrapping Tukey-Kramer or Games-Howell using onewayComp() with nboot > 0.
- For FWER control for just a few pairs of means:
  - We haven't done this, but if the number of pairs is small then Bonferonni (or Bonferonni-Holm if you don't need CI's) may be better than Tukey-Kramer. Do t-tests for just the pairs of interest and then make corrections. Bootstrap if needed.
- For FDR control when comparing all possible pairs
  - Use onewayComp(..., adjust = 'BH') to do pairwise t-tests.
  - Normal OK use nboot $= 0$.
  - Normality not OK use nboot > 0.

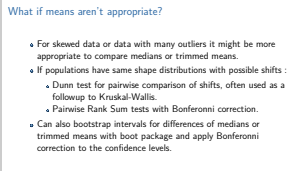

Which Procedure for Comparing Means?

- For FWER control when comparing all possible pairs of means:
  - Normality OK → Tukey-Kramer (var.equal=TRUE) or Games-Howell (var.equal = FALSE). onewayComp() from DS705data package gives tests and CI's.
  - Normality not OK → try bootstrapping Tukey-Kramer or Games-Howell using onewayComp() with nboot > 0.
- For FWER control for just a few pairs of means:
  - We haven't done this, but if the number of pairs is small then Bonferonmi (or Bonferonmi-Holm if you don't need CI's) may be better than Tukey-Kramer. Do t-tests for just the pairs of interest and then make corrections. Bootstrap if needed.
- For FDR control when comparing all possible pairs
  - Use onewayComp(. . . , adjust = 'BH') to do pairwise t-tests.
  - Normal OK use nboot = 0.
  - Normality not OK use nboot > 0.

- no audio

What if means aren't appropriate?

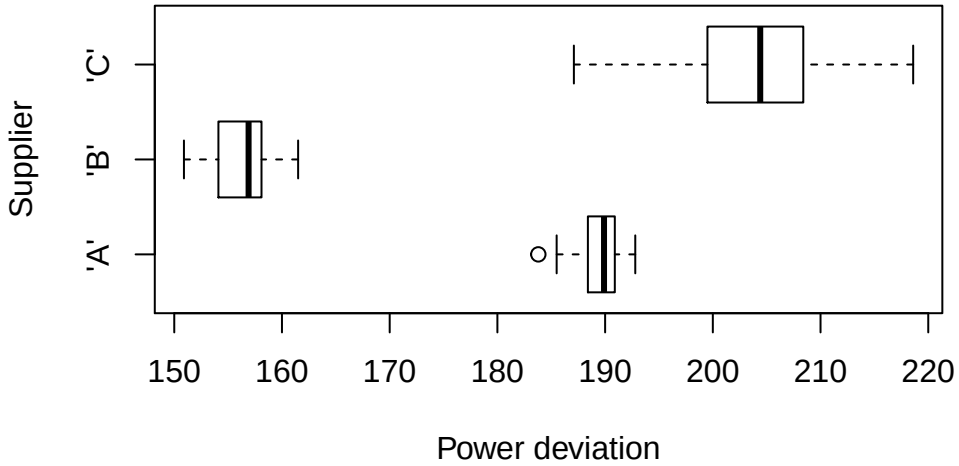
- For skewed data or data with many outliers it might be more appropriate to compare medians or trimmed means.
- If populations have same shape distributions with possible shifts :
  - Dunn test for pairwise comparison of shifts, often used as a followup to Kruskal-Wallis.
  - Pairwise Rank Sum tests with Bonferonni correction.
- Can also bootstrap intervals for differences of medians or trimmed means with boot package and apply Bonferonni correction to the confidence levels.



- audio31.mp3
- Since there's already plenty of material to learn in this lesson we won't cover all of these alternative procedures here
- but we will conclude with a bootstrapping example that shows you how to bootstrap a family of confidence intervals, in this case, for differences of medians.

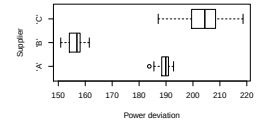
Contact Lenses Example

See Problem 8.27 in Ott (page 442).



Contact Lenses Example

See Problem 8.27 in Ott (page 442).



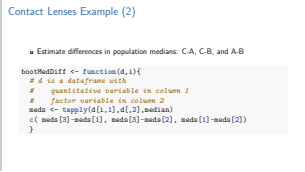

```
## Loading required package: boot
```

- audio32.mp3
- The response variable, recorded for each of the suppliers, is the deviation between the actual power of a lens and the reported or labeled power of a lens.
- because of outliers and potential skewness, we elect to estimate the differences in the population medians instead of the means.

Contact Lenses Example (2)

- Estimate differences in population medians: C-A, C-B, and A-B

```
bootMedDiff <- function(d,i){
  # d is a dataframe with
  #    quantitative variable in column 1
  #    factor variable in column 2
  meds <- tapply(d[i,1],d[,2],median)
  c( meds[3]-meds[1], meds[3]-meds[2], meds[1]-meds[2])
  }
```



- audio33.mp3
- Here is our helper function for computing the three differences of medians
- we will pass this function to the boot() function and use the strata option so that the resampling occurs within each sample.

Contact Lenses Example (3)

```
boot.object <- boot(contacts, bootMedDiff, R = 5000,
                    strata = contacts$Supplier)
# med_C - med_A
boot.ci(boot.object,conf = 1 - .05/3, type='bca', index=1)$bca[4:5]
```

```
## [1]  6.7 22.1
```

```
# med_C - med_B
boot.ci(boot.object,conf = 1 - .05/3, type='bca', index=2)$bca[4:5]
```

```
## [1] 39.2 54.5
```

```
# med_A - med_B (= 6)
boot.ci(boot.object,conf = 1 - .05/3, type='bca', index=3)$bca[4:5]
```

```
## [1] 27.8 37.8
```

Contact Lenses Example (3)

- audio34.mp3
- notice we're using the bca intervals that we've used throughout the course and
- that we've also use a Bonferonni correction for the three intervals so that we can be 95% confident in the entire family of intervals.

Fast Facts: Bonferonni Correction for Multiple Tests

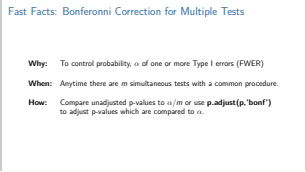
Why: To control probability, α of one or more Type I errors (FWER)

When: Anytime there are *m* simultaneous tests with a common procedure.

How: Compare unadjusted p-values to α/m or use **p.adjust(p,'bonf')** to adjust p-values which are compared to α.

Multiple Testing

Fast Facts: Bonferonni Correction for Multiple Tests



- No audio.

Fast Facts: Bonferonni Correction for Multiple Confidence Intervals

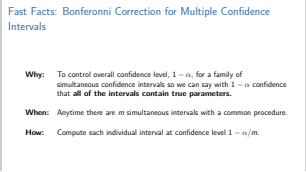
Why: To control overall confidence level, $1 - \alpha$, for a family of simultaneous confidence intervals so we can say with $1 - \alpha$ confidence that **all of the intervals contain true parameters.**

When: Anytime there are m simultaneous intervals with a common procedure.

How: Compute each individual interval at confidence level $1 - \alpha/m$.

Multiple Testing

Fast Facts: Bonferonni Correction for Multiple Confidence Intervals



- No audio.

Fast Facts: False Discovery Rate for Multiple Tests

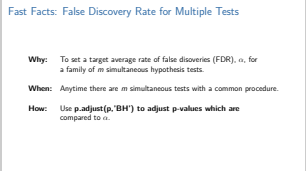
Why: To set a target average rate of false disoveries (FDR), α, for a family of *m* simultaneous hypothesis tests.

When: Anytime there are *m* simultaneous tests with a common procedure.

How: Use **p.adjust(p,'BH') to adjust p-values which are** compared to α.

Multiple Testing

Fast Facts: False Discovery Rate for Multiple Tests



- No audio.

Our 2 Cents

- Multiple comparisons is a huge topic so we've just given you a survey of the some the most widely used tools.
- Use FWER when it's important to not make Type I errors.
- Use FDR when it's more important to make discoveries for further research, but still want to keep a handle on False Discoveries.
- Only use PCER (no corrections) when it's very important not to make Type II errors or when you're just exploring.
- Bonferonni assumes the tests are perfectly independent so it's unnecessarily conservative in many circumstances.
- Beware of data dredging. Don't do many comparisons without correction and report only the significant ones, that's bad science.

Our 2 Cents

- Multiple comparisons is a huge topic so we've just given you a survey of the some the most widely used tools.
- Use FWER when it's important to not make Type I errors.
- Use FDR when it's more important to make discoveries for further research, but still want to keep a handle on False Discoveries.
- Only use PCER (no corrections) when it's very important not to make Type II errors or when you're just exploring.
- Bonferonni assumes the tests are perfectly independent so it's unnecessarily conservative in many circumstances.
- Beware of data dredging. Don't do many comparisons without correction and report only the significant ones, that's bad science.

- No audio.

Our 2 Cents (Continued)

- For pairwise comparisons of means the Tukey-Kramer/Games-Howell procedures are preferable to Bonferonni for FWER control.
- The Dunn test is a good followup for a significant result in Kruskal-Wallis and there is an R-package - `dunn.test`
- onewayComp() is a versatile tool which you're welcome to use after the class ends.
- onewayComp() also supports arbitrary linear contrasts which is something we haven't covered but is in the textbook.
- Try doing onewayComp() with nboot $= 0$ and with nboot $= 1000$ or more and if the results are similar then assumptions about normality and such are probably fine.

Multiple Testing

Our 2 Cents (Continued)

Our 2 Cents (Continued)

- For pairwise comparisons of means the Tukey-Kramer/Games-Howell procedures are preferable to Bonferonni for FWER control.
- The Dunn test is a good followup for a significant result in Kruskal-Wallis and there is an R-package - `dunn.test`
- onewayComp() is a versatile tool which you're welcome to use after the class ends.
- onewayComp() also supports arbitrary linear contrasts which is something we haven't covered but is in the textbook.
- Try doing onewayComp() with nboot = 0 and with nboot = 1000 or more and if the results are similar then assumptions about normality and such are probably fine.

- No audio.