Selection: 1

| Please choose a lesson, or type 0 to return to course menu.

1: Introduction 2: Probability1 3: Probability2

4: ConditionalProbability 5: Expectations 6: Variance

7: CommonDistros 8: Asymptotics 9: T Confidence Intervals

10: Hypothesis Testing 11: P Values 12: Power

13: Multiple Testing 14: Resampling

Selection: 13

| Attempting to load lesson dependencies...

| Package ‘jpeg’ loaded correctly!

| | | 0%

| Multiple\_Testing. (Slides for this and other Data Science courses may be found at

| github https://github.com/DataScienceSpecialization/courses/. If you care to use them,

| they must be downloaded as a zip file and viewed locally. This lesson corresponds to

| 06\_Statistical\_Inference/12\_MultipleTesting.)

...

| |= | 2%

| In this lesson, we'll discuss multiple testing. You might ask, "What's that?"

...

| |=== | 3%

| Given that data is valuable and we'd like to get the most out of it, we might use it to

| test several hypotheses. If we have an alpha level of .05 and we test 20 hypotheses,

| then on average, we expect one error, just by chance.

...

| |==== | 5%

| Another potential problem is that after running several tests, only the lowest p-value

| might be reported OR all p-values under some threshold might be considered significant.

| Undoubtedly, some of these would be false.

...

| |===== | 7%

| Luckily, we have clever ways of minimizing errors in this situation. That's what we'll

| address. We'll define specific error measures and then statistical ways of correcting

| or limiting them.

...

| |======= | 8%

| Multiple testing is particularly relevant now in this age of BIG data. Statisticians

| are tasked with questions such as "Which variables matter among the thousands

| measured?" and "How do you relate unrelated information?"

...

| |======== | 10%

| Since multiple testing addresses compensating for errors let's review what we know

| about them. A Type I error is

1: rejecting a true hypothesis

2: rejecting a false hypothesis

3: failing to reject a false hypothesis

4: failing to reject a true hypothesis

Selection: 1

| You're the best!

| |========= | 11%

| In an American court, an example of a Type I error is

1: convicting an innocent person

2: letting the indicted off on a technicality

3: acquitting a guilty person

Selection: 1

| That's correct!

| |========== | 13%

| A Type II error is

1: rejecting a true hypothesis

2: failing to reject a false hypothesis

3: failing to reject a true hypothesis

4: rejecting a false hypothesis

Selection: 2

| Your dedication is inspiring!

| |============ | 15%

| In an American court, an example of a Type II error is

1: letting the indicted off on a technicality

2: acquitting a guilty person

3: convicting an innocent person

Selection: 2

| You are amazing!

| |============= | 16%

| Good. Let's continue reviewing. The null hypothesis

1: represents the status\_quo and is assumed true

2: is a big nothing that statisticians like to gossip about

3: tells us the origins of the number 0

4: is never true

Selection: 1

| All that hard work is paying off!

| |============== | 18%

| The p-value is "the probability under the null hypothesis of obtaining evidence as or

| more extreme than your test statistic (obtained from your observed data) in the

| direction of the alternative hypothesis." Of course p-values are related to

| significance or alpha levels, which are set before the test is conducted (often at

| 0.05).

...

| |================ | 20%

| If a p-value is found to be less than alpha (say 0.05), then the test result is

| considered statistically significant, i.e., surprising and unusual, and the null

| hypothesis (the status quo) is ?

1: rejected

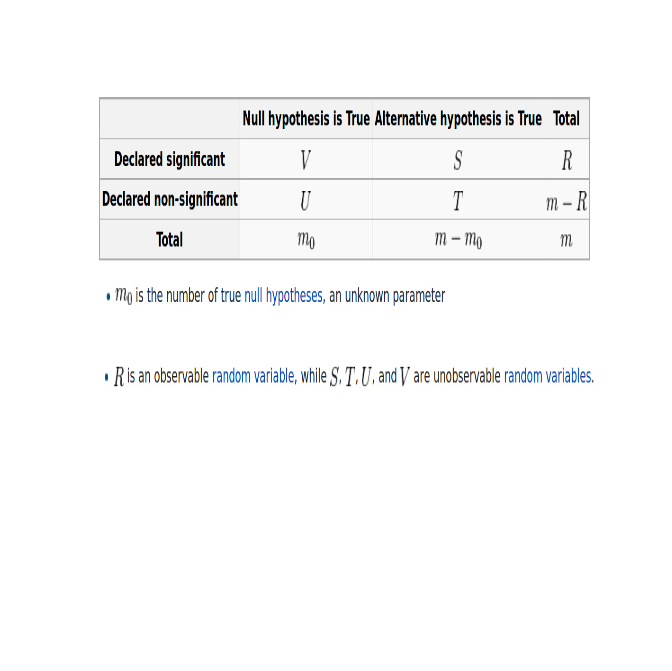
2: revised

3: accepted

4: renamed the aleph null hypothesis

Selection: 1

| You got it right!



| |================= | 21%

| Now consider this chart copied from http://en.wikipedia.org/wiki/Familywise\_error\_rate.

| Suppose we've tested m null hypotheses, m\_0 of which are actually true, and m-m\_0 are

| actually false. Out of the m tests R have been declared significant, that is, the

| associated p-values were less than alpha, and m-R were nonsignificant, or boring

| results.

...

| |================== | 23%

| Looking at the chart, which variables are known?

1: A,B,C

2: S,T,U,V

3: m\_0, and m

4: m and R

Selection: 4

| All that practice is paying off!

| |==================== | 25%

| In testing the m\_0 true null hypotheses, V results were declared significant, that is,

| these tests favored the alternative hypothesis. What type of error does this represent?

1: Type I

2: a serious one

3: Type III

4: Type II

Selection: 1

| Great job!

| |===================== | 26%

| Another name for a Type I error is False Positive, since it is falsely claiming a

| significant (positive) result.

...

| |====================== | 28%

| Of the m-m\_0 false null hypotheses, T were declared nonsignificant. This means that

| these T null hypotheses were accepted (failed to be rejected). What type of error does

| this represent?

1: Type II

2: a serious one

3: Type III

4: Type I

Selection: 1

| You're the best!

| |======================== | 30%

| Another name for a Type II error is False Negative, since it is falsely claiming a

| nonsignificant (negative) result.

...

| |========================= | 31%

| A rose by any other name, right? Consider the fraction V/R.

...

| |========================== | 33%

| The observed R represents the number of test results declared significant. These are

| 'discoveries', something different from the status quo. V is the number of those

| falsely declared significant, so V/R is the ratio of FALSE discoveries. Since V is a

| random variable (i.e., unknown until we do an experiment) we call the expected value of

| the ratio, E(V/R), the False Discovery Rate (FDR).

...

| |============================ | 34%

| A rose by any other name, right? How about the fraction V/m\_0? From the chart, m\_0

| represents the number of true H\_0's and m\_0 is unknown. V is the number of those

| falsely declared significant, so V/m\_0 is the ratio of FALSE positives. Since V is a

| random variable (i.e., unknown until we do an experiment) we call the expected value of

| the ratio, E(V/m\_0), the FALSE POSITIVE rate.

...

| |============================= | 36%

| Another good name for the false positive rate would be

1: a thorn

2: a rose

3: the Type II rate

4: false alarm rate

Selection: 4

| Excellent job!

| |============================== | 38%

| The false positive rate would be closely related to

1: a thorny rose

2: the Type I error rate

3: the Type II error rate

Selection: 2

| You're the best!

| |=============================== | 39%

| We call the probability of at least one false positive, Pr(V >= 1) the Family Wise

| Error Rate (FWER).

...

| |================================= | 41%

| So how do we control the False Positive Rate?

...

| |================================== | 43%

| Suppose we're really smart, calculate our p-values correctly, and declare all tests

| with p < alpha as significant. This means that our false positive rate is at most

| alpha, on average.

...

| |=================================== | 44%

| Suppose we perform 10,000 tests and alpha = .05. How many false positives do we expect

| on average?

1: 5000

2: 500

3: 50

4: 50000

Selection: 2

| That's a job well done!

| |===================================== | 46%

| You got it! 500 false positives seems like a lot. How do we avoid so many?

...

| |====================================== | 48%

| We can try to control the family-wise error rate (FWER), the probability of at least

| one false positive, with the Bonferroni correction, the oldest multiple testing

| correction.

...

| |======================================= | 49%

| It's very straightforward. We do m tests and want to control the FWER at level alpha so

| that Pr(V >= 1) < alpha. We simply reduce alpha dramatically. Set alpha\_fwer to be

| alpha/m. We'll only call a test result significant if its p-value < alpha\_fwer.

...

| |========================================= | 51%

| Sounds good, right? Easy to calculate. What would be a drawback with this method?

1: too many results will pass

2: too many results will fail

3: requires too much math

Selection: 2

| You are doing so well!

| |========================================== | 52%

| Another way to limit the false positive rate is to control the false discovery rate

| (FDR). Recall this is E(V/R). This is the most popular correction when performing lots

| of tests. It's used in lots of areas such as genomics, imaging, astronomy, and other

| signal-processing disciplines.

...

| |=========================================== | 54%

| Again, we'll do m tests but now we'll set the FDR, or E(V/R) at level alpha. We'll

| calculate the p-values as usual and order them from smallest to largest, p\_1,

| p\_2,...p\_m. We'll call significant any result with p\_i <= (alpha\*i)/m. This is the

| Benjamini-Hochberg method (BH). A p-value is compared to a value that depends on its

| ranking.

...

| |============================================= | 56%

| This is equivalent to finding the largest k such that p\_k <= (k \* alpha)/m, (for a

| given alpha) and then rejecting all the null hypotheses for i=1,...,k.

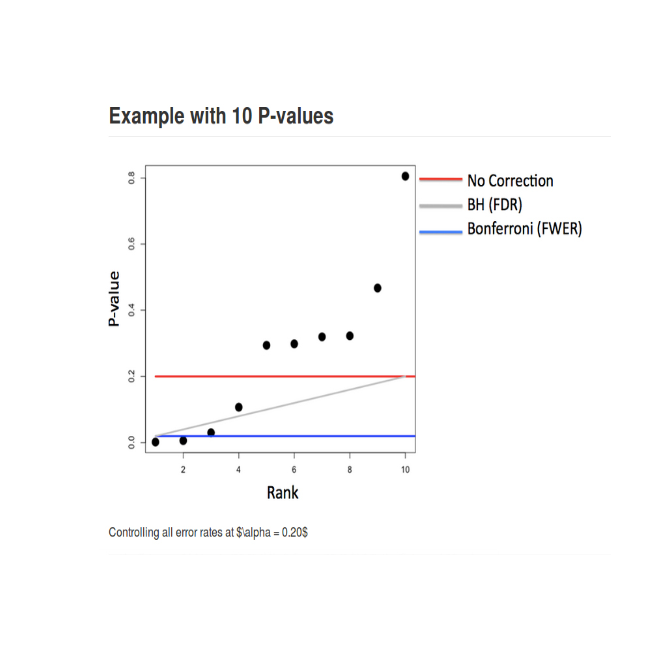
...

| |============================================== | 57%

| Like the Bonferroni correction, this is easy to calculate and it's much less

| conservative. It might let more false positives through and it may behave strangely if

| the tests aren't independent.



...

| |=============================================== | 59%

| Now consider this chart copied from the slides. It shows the p-values for 10 tests

| performed at the alpha=.2 level and three cutoff lines. The p-values are shown in order

| from left to right along the x-axis. The red line is the threshold for No Corrections

| (p-values are compared to alpha=.2), the blue line is the Bonferroni threshold,

| alpha=.2/10 = .02, and the gray line shows the BH correction. Note that it is not

| horizontal but has a positive slope as we expect.

...

| |================================================= | 61%

| With no correction, how many results are declared significant?

1: 6

2: 2

3: 8

4: 4

Selection: 1

| Not quite! Try again.

| How many points fall below the red line?

1: 8

2: 6

3: 2

4: 4

Selection: 4

| That's a job well done!

| |================================================== | 62%

| With the Bonferroni correction, how many tests are declared significant?

1: 4

2: 2

3: 6

4: 8

Selection: 2

| You're the best!

| |=================================================== | 64%

| So the Bonferroni passed only half the results that the No Correction (comparing

| p-values to alpha) method passed. Now look at the BH correction. How many tests are

| significant with this scale?

1: 7

2: 3

3: 5

4: 1

Selection: 2

| Perseverance, that's the answer.

| |==================================================== | 66%

| So the BH correction which limits the FWER is between the No Correction and the

| Bonferroni. It's more conservative (fewer significant results) than the No Correction

| but less conservative (more significant results) than the Bonferroni. Note that with

| this method the threshold is proportional to the ranking of the values so it slopes

| positively while the other two thresholds are flat.

...

| |====================================================== | 67%

| Notice how both the Bonferroni and BH methods adjusted the threshold (alpha) level of

| rejecting the null hypotheses. Another equivalent corrective approach is to adjust the

| p-values, so they're not classical p-values anymore, but they can be compared directly

| to the original alpha.

...

| |======================================================= | 69%

| Suppose the p-values are p\_1, ... , p\_m. With the Bonferroni method you would adjust

| these by setting p'\_i = max(m \* p\_i, 1) for each p-value. Then if you call all p'\_i <

| alpha significant you will control the FWER.

...

| |======================================================== | 70%

| To demonstrate some of these concepts, we've created an array of p-values for you. It

| is 1000-long and the result of a linear regression performed on random normal x,y pairs

| so there is no true significant relationship between the x's and y's.

...

| |========================================================== | 72%

| Use the R command head to see the first few entries of the array pValues.

> head(pValues)

[1] 0.5334915 0.2765785 0.8380943 0.6721730 0.8122037 0.4078675

| Great job!

| |=========================================================== | 74%

| Now count the number of entries in the array that are less than the value .05. Use the

| R command sum, and the appropriate Boolean expression.

> sum(pValues < 0.05)

[1] 51

| Your dedication is inspiring!

| |============================================================ | 75%

| So we got around 50 false positives, just as we expected (.05\*1000=50). The beauty of R

| is that it provides a lot of built-in statistical functionality. The function p.adjust

| is one example. The first argument is the array of pValues. Another argument is the

| method of adjustment. Once again, use the R function sum and a boolean expression using

| p.adjust with method="bonferroni" to control the FWER.

> p.adjust(pValues, method="bonferroni")

[1] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[8] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[15] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[22] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[29] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[36] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[43] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[50] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[57] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[64] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[71] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[78] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[85] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[92] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[99] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[106] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[113] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[120] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[127] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[134] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[141] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[148] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[155] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[162] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[169] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[176] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[183] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[190] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[197] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[204] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[211] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[218] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[225] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[232] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[239] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[246] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[253] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[260] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[267] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[274] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[281] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[288] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[295] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[302] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[309] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[316] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[323] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[330] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[337] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[344] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[351] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[358] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[365] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[372] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[379] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[386] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[393] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[400] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[407] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[414] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[421] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[428] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[435] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[442] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[449] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[456] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[463] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[470] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[477] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[484] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[491] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[498] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[505] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[512] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[519] 0.56288445 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[526] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[533] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[540] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[547] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[554] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[561] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[568] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[575] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[582] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[589] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[596] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[603] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 0.09482691

[610] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[617] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[624] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[631] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[638] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[645] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[652] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[659] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[666] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[673] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[680] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[687] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[694] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[701] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[708] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[715] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[722] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[729] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[736] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[743] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[750] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[757] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[764] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[771] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[778] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[785] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[792] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[799] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[806] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[813] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[820] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[827] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[834] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[841] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[848] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[855] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[862] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[869] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[876] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[883] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[890] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[897] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[904] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[911] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[918] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[925] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[932] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[939] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[946] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[953] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[960] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[967] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[974] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[981] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[988] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

[995] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

| Give it another try. Or, type info() for more options.

| Type sum(p.adjust(pValues,method="bonferroni") < 0.05) at the command prompt.

> sum(p.adjust(pValues,method="bonferroni") < 0.05)

[1] 0

| All that hard work is paying off!

| |============================================================== | 77%

| So the correction eliminated all the false positives that had passed the uncorrected

| alpha test. Repeat the same experiment, this time using the method "BH" to control the

| FDR.

> sum(p.adjust(pValues,method="BH") < 0.05)

[1] 0

| Your dedication is inspiring!

| |=============================================================== | 79%

| So the BH method also eliminated all the false positives. Now we've generated another

| 1000-long array of p-values, this one called pValues2. In this data, the first half (

| 500 x/y pairs) contains x and y values that are random and the second half contain x

| and y pairs that are related, so running a linear regression model on the 1000 pairs

| should find some significant (not random) relationship.

...

| |================================================================ | 80%

| We also created a 1000-long array of character strings, trueStatus. The first 500

| entries are "zero" and the last are "not zero". Use the R function tail to look at the

| end of trueStatus.

> tail(trueStatus)

[1] "not zero" "not zero" "not zero" "not zero" "not zero" "not zero"

| Great job!

| |================================================================== | 82%

| Once again we can use R's greatness to count and tabulate for us. We can call the R

| function table with two arguments, a boolean such as pValues2<.05, and the array

| trueStatus. The boolean obviously has two outcomes and each entry of trueStatus has one

| of two possible values. The function table aligns the two arguments and counts how many

| of each combination (TRUE,"zero"), (TRUE,"not zero"), (FALSE,"zero"), and (FALSE,"not

| zero") appear. Try it now.

> table(pValues2<.05, trueStatus)

trueStatus

not zero zero

FALSE 0 476

TRUE 500 24

| Your dedication is inspiring!

| |=================================================================== | 84%

| We see that without any correction all 500 of the truly significant (nonrandom) tests

| were correctly identified in the "not zero" column. In the zero column (the truly

| random tests), however, 24 results were flagged as significant.

...

| |==================================================================== | 85%

| What is the percentage of false positives in this test?

> 24/500

[1] 0.048

| You are doing so well!

| |====================================================================== | 87%

| Just as we expected - around 5% or .05\*100.

...

| |======================================================================= | 89%

| Now run the same table function, however, this time use the call to p.adjust with the

| "bonferroni" method in the boolean expression. This will control the FWER.

> table(p.adjust(pValues2<.05, trueStatus)

+ )

Error in match.arg(method) : 'arg' must be of length 1

> table(p.adjust(pValues2<.05, trueStatus))

Error in match.arg(method) : 'arg' must be of length 1

> table(p.adjust(pValues2)<.05, trueStatus)

trueStatus

not zero zero

FALSE 16 500

TRUE 484 0

| Give it another try. Or, type info() for more options.

| Type table(p.adjust(pValues2,method="bonferroni") < 0.05, trueStatus) at the command

| prompt.

> table(p.adjust(pValues2,method="bonferroni") < 0.05, trueStatus)

trueStatus

not zero zero

FALSE 23 500

TRUE 477 0

| Your dedication is inspiring!

| |======================================================================== | 90%

| Since the Bonferroni correction method is more conservative than just comparing

| p-values to alpha all the truly random tests are correctly identified in the zero

| column. In other words, we have no false positives. However, the threshold has been

| adjusted so much that 23 of the truly significant results have been misidentified in

| the not zero column.

...

| |========================================================================= | 92%

| Now run the same table function one final time. Use the call to p.adjust with "BH"

| method in the boolean expression. This will control the false discovery rate.

> table(p.adjust(pValues2,method="BH") < 0.05, trueStatus)

trueStatus

not zero zero

FALSE 0 487

TRUE 500 13

| Perseverance, that's the answer.

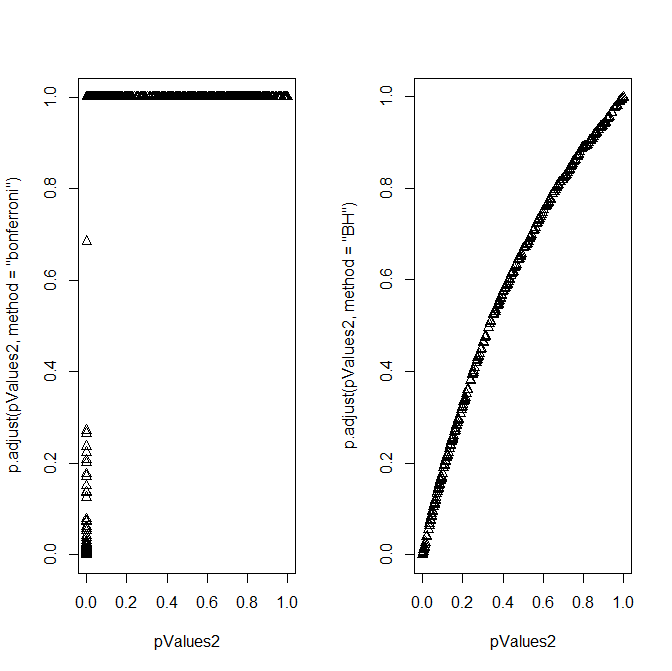
| |=========================================================================== | 93%

| Again, the results are a compromise between the No Corrections and the Bonferroni. All

| the significant results were correctly identified in the "not zero" column but in the

| random ("zero") column 13 results were incorrectly identified. These are the false

| positives. This is roughly half the number of errors in the other two runs.



...

| |============================================================================ | 95%

| Here's a plot of the two sets of adjusted p-values, Bonferroni on the left and BH on

| the right. The x-axis indicates the original p-values. For the Bonferroni, (adjusting

| by multiplying by 1000, the number of tests), only a few of the adjusted values are

| below 1. For the BH, the adjusted values are slightly larger than the original values.

...

| |============================================================================= | 97%

| We'll conclude by saying that multiple testing is an entire subfield of statistical

| inference. Usually a basic Bonferroni/BH correction is good enough to eliminate false

| positives, but if there is strong dependence between tests there may be problems.

| Another correction method to consider is "BY".

...

| |=============================================================================== | 98%

| Congrats! We hope you liked the multiple concepts and questions you saw in this lesson.

...

| |================================================================================| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: Yes

2: No

Selection: 1

What is your email address? sweeyean@gmail.com

What is your assignment token? zuV7NVgRkDrU6ISk

Grade submission succeeded!

| All that hard work is paying off!

| You've reached the end of this lesson! Returning to the main menu...

| Please choose a course, or type 0 to exit swirl.

1: Statistical Inference

2: Take me to the swirl course repository!

Selection: