Exploration librairie swirl : Multiple Testing

> swirl()

Welcome to swirl! Please sign in. If you've been here before, use the same name as you did then. If you are new, call yourself something unique.

What shall I call you? ***jlbellier***

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

1: Statistical Inference
2: Take me to the swirl course repository!

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: failing to reject a true hypothesis
2: rejecting a false hypothesis
3: rejecting a true hypothesis
4: failing to reject a false hypothesis

Selection: ***3***

Nice work!

========== ***11%***

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

1: acquitting a guilty person
2: letting the indicted off on a technicality
3: convicting an innocent person

Selection: ***3***

Excellent job!

========== ***13%***

A Type II error is

1: rejecting a true hypothesis
2: failing to reject a true hypothesis
3: failing to reject a false hypothesis
4: rejecting a false hypothesis

Selection: ***3***

That's a job well done!

========== ***15%***

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

1: letting the indicted off on a technicality

2: convicting an innocent person
3: acquitting a guilty person

Selection: ***3***

You got it!

========== ***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: is never true
4: tells us the origins of the number 0

Selection: ***1***

Keep up the great work!

========== ***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: accepted
2: rejected
3: renamed the aleph null hypothesis
4: revised

Selection: ***1***

Not quite, but you're learning! Try again.
Accepted (failed to be rejected) or rejected are the only real choices here. A low p-value is a low probability. This means your data is unusual and is closer to the alternative hypothesis than the null.

1: renamed the aleph null hypothesis
2: rejected
3: revised
4: accepted

Selection: ***2***
Keep up the great work!

========== ***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: ***2***

That's not exactly what I'm looking for. Try again.
The number of hypotheses tested (m) and the number declared significant (R) are known. The variable m_0 represents the unknowable, the number of true hypotheses. S, T, U, and V are unobservable random variables.

1: S,T,U,V
2: A,B,C
3: m and R
4: m_0, and m

Selection: ***1***

You almost had it, but not quite. Try again.
The number of hypotheses tested (m) and the number declared significant (R) are known. The variable m_0 represents the unknowable, the number of true hypotheses. S, T, U, and V are unobservable random variables.

1: m and R
2: m_0, and m
3: S,T,U,V
4: A,B,C

Selection: ***1***

Nice work!

========== ***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: a serious one
2: Type I
3: Type III
4: Type II

*Selection: **2***

You are amazing!

========== ***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 I
2: a serious one
3: Type III
4: Type II

*Selection: **4***

Your dedication is inspiring!

========== ***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 rose
2: false alarm rate
3: the Type II rate
4: a thorn

*Selection: **1***

Not quite right, but keep trying.

False positives are Type I errors so one of the only two sensible answers is incorrect.

1: a thorn
2: the Type II rate
3: a rose
4: false alarm rate

*Selection: **4***

You nailed it! Good job!

========== ***38%***

The false positive rate would be closely related to

1: the Type I error rate
2: a thorny rose
3: the Type II error rate

*Selection: **1***

That's the answer I was looking for.

========== ***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: 50000
2: 50
3: 5000
4: 500

*Selection: **4***

All that practice is paying off!

========== ***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: ***1***

Try again. Getting it right on the first try is boring anyway!
Dividing alpha by m makes your cutoff value very small so you might not get any significant results, much less false ones.

1: too many results will fail
2: too many results will pass
3: requires too much math

Selection: ***1***

Nice work!

========== ***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: 8
2: 6
3: 4
4: 2

*Selection: **3***

That's a job well done!

========== ***62%***

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

1: 2
2: 8
3: 6
4: 4

*Selection: **2***

Not exactly. Give it another go.
How many points fall below the blue line?

1: 2
2: 8
3: 6
4: 4

*Selection: **1***

That's correct!

========== ***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: 3
2: 7
3: 1
4: 5

*Selection: **1***

You got it!

========== ***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.*

```
> sum(p.adjust(pValues,method="bonferroni"))
[1] 998.6577
```

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<0.05)
FALSE  TRUE
  476   524
```

That's not the answer I was looking for, but try again. Or, type info() for more options.

Type table(pValues2 < 0.05, trueStatus) at the command prompt.

```
> table(pValues2<0.05,trueStatus)
       trueStatus
        not zero zero
  FALSE        0  476
  TRUE       500   24
```

You are doing so well!

========== 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?

*> **47.6***
[1] 47.6

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

Divide 24 by 500 to get the percentage.

*> **24/500***
[1] 0.048

Your dedication is inspiring!

========== ***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,method="bonferroni")<0.05,trueStatus)***

```
      trueStatus
       not zero zero
 FALSE       23  500
 TRUE       477    0
```

Perseverance, that's the answer.

========== ***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
```

You nailed it! Good job!

========== ***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%