P values

P values and effect sizes

What is a P value?

This simple question is surprisingly difficult to nail down exactly.

"Informally, a p-value is the probability under a specified statistical model that a statistical summary of the data (e.g., the sample mean difference between two compared groups) would be equal to or more extreme than its observed value."

The ASA's Statement on p-Values: Context, Process, and Purpose, Wasserstein and Lazar, The American Statistician, 2016

So to break this down a bit. A p-value is only valid if the statistical model is valid, if we are using an appropriate summary of the data. A p-value is not a test of whether the hypothesis is true, nor is it a measure of magnitude or importance, nor is it by itself, a good indicator of evidence for or against a model.

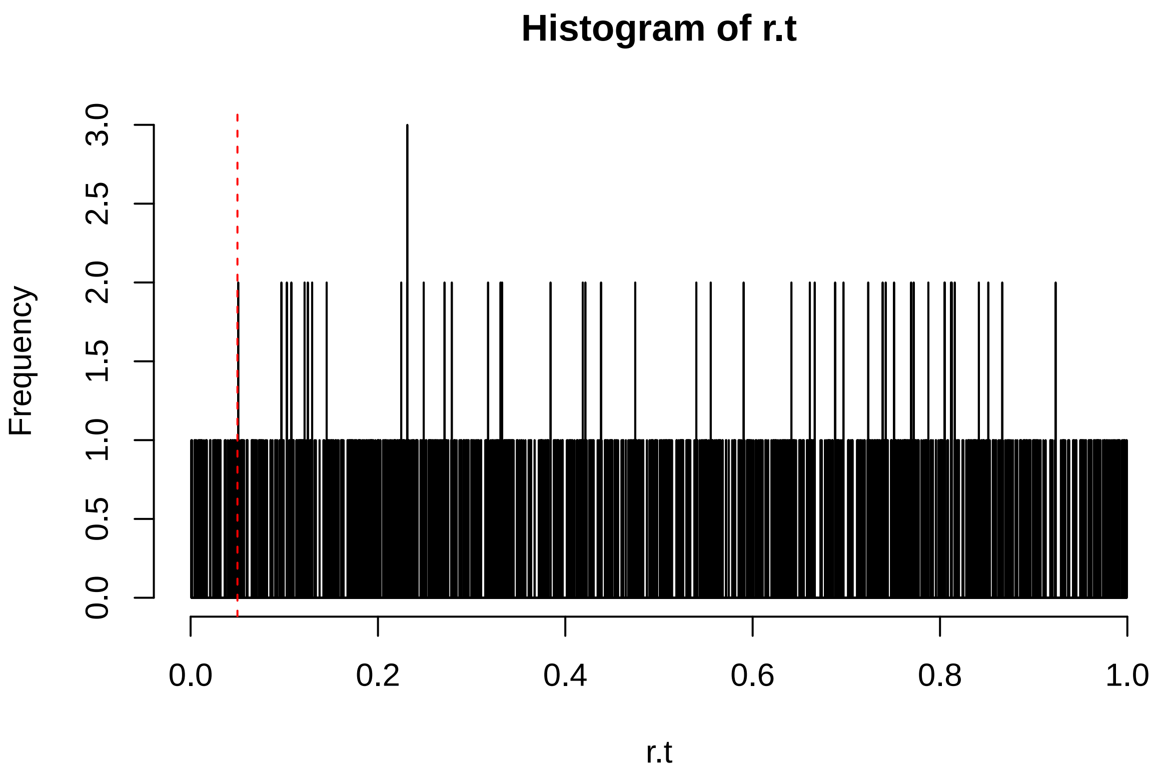
In addition, p-values have the problem of not being reproducible when sample sizes are small The fickle P value generates irreproducible results , Halsey et. al 2015. Nat. Methods.

Lets explore the behaviour of p-values

First, the expected distribution of p-value is random uniform in non-informative data where every value is independent. That is, a flat line between 0 and 1. This is an important concept, because most, if not all, analysis tools expect the majority of things not to change between experimental replicates. For example if you knock out a gene, it is not expected to change the expression of a majority of the remaining genes in the system.

Play with the n.samples to convince yourself that having different numbers of samples does not change the distribution.

```r
# toy example
# first case
# change the number of samples
n.samples <- 1000
n.tests = 1000
breaks <- as.integer(n.samples/2)
first.mean <- 100

# generate random data
r <- matrix(data=NA, nrow=n.samples, ncol=n.tests)
for(i in 1:n.samples){ r[i,] <- rnorm(n.tests, mean=first.mean, sd=5) }

# and the t-tests
r.t <- apply(r, 2, function(x) as.numeric(t.test(x[1:breaks], x[(breaks+1):n.samples])[3]) )

hist(r.t, breaks=10000)
abline(v=0.05, col="red", lty=2)
```

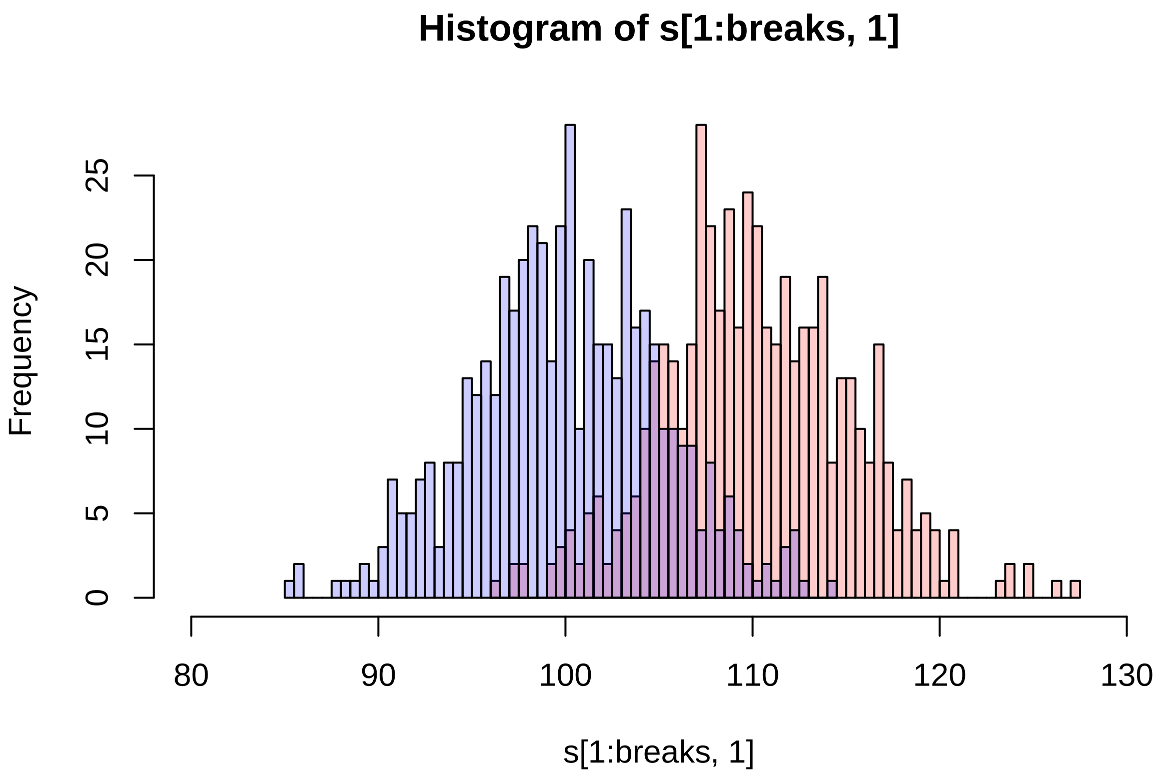


Now we can add in a small number of true differences.

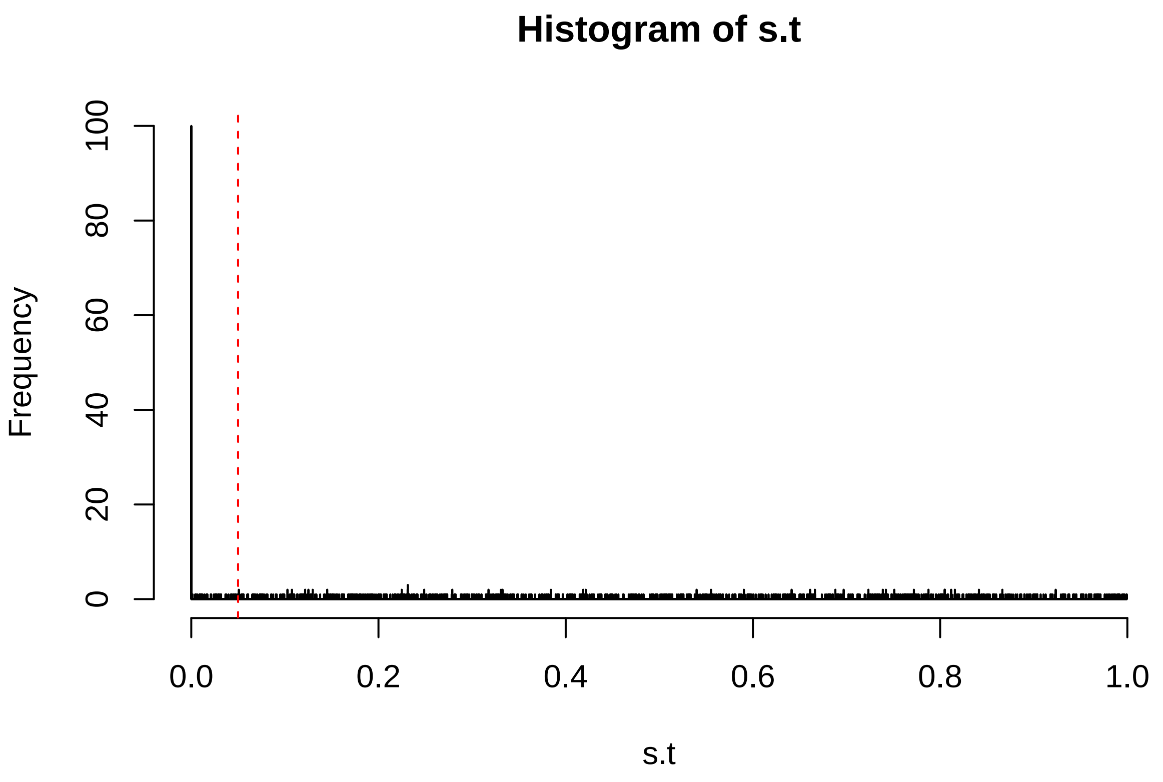
How does the distribution change?

What is the relationship between the data points?

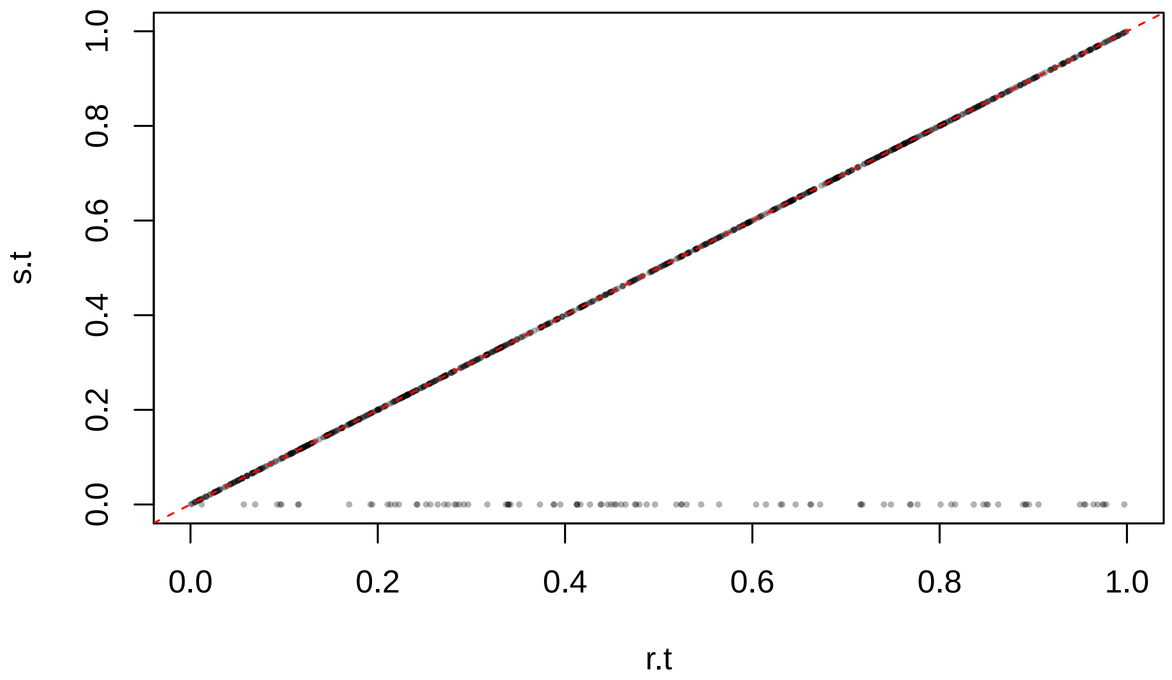
```
n.sig <- 100

# copy r to s
s <- r
second.mean <- 110

for(i in 1:breaks){ s[i,1:n.sig] <- rnorm(n.sig, mean=second.mean, sd=5) }

# change the mean, and sd and observe how the distributions change
hist(s[1:breaks, 1], breaks=100, xlim=c((first.mean - 20),(second.mean + 20)), col=rgb(1,0,0,0.2))
hist(s[(breaks+1):n.samples, 1], breaks=100, col=rgb(0,0,1,0.2), add=T)
```



```r
s.t <- apply(s, 2, function(x) as.numeric(t.test(x[1:breaks], x[(breaks+1):n.samples])[3]))

hist(s.t, breaks=10000)
abline(v=0.05, col="red", lty=2)
```



```r
plot(r.t, s.t, pch=19, cex=0.3, col=rgb(0,0,0,0.3))
abline(0,1, col="red", lty=2)
```



So this is all good. We have a null hypothesis that assumes that almost nothing changes in the data, except for a few values where we have increased the mean of the data from 100. So far so good!

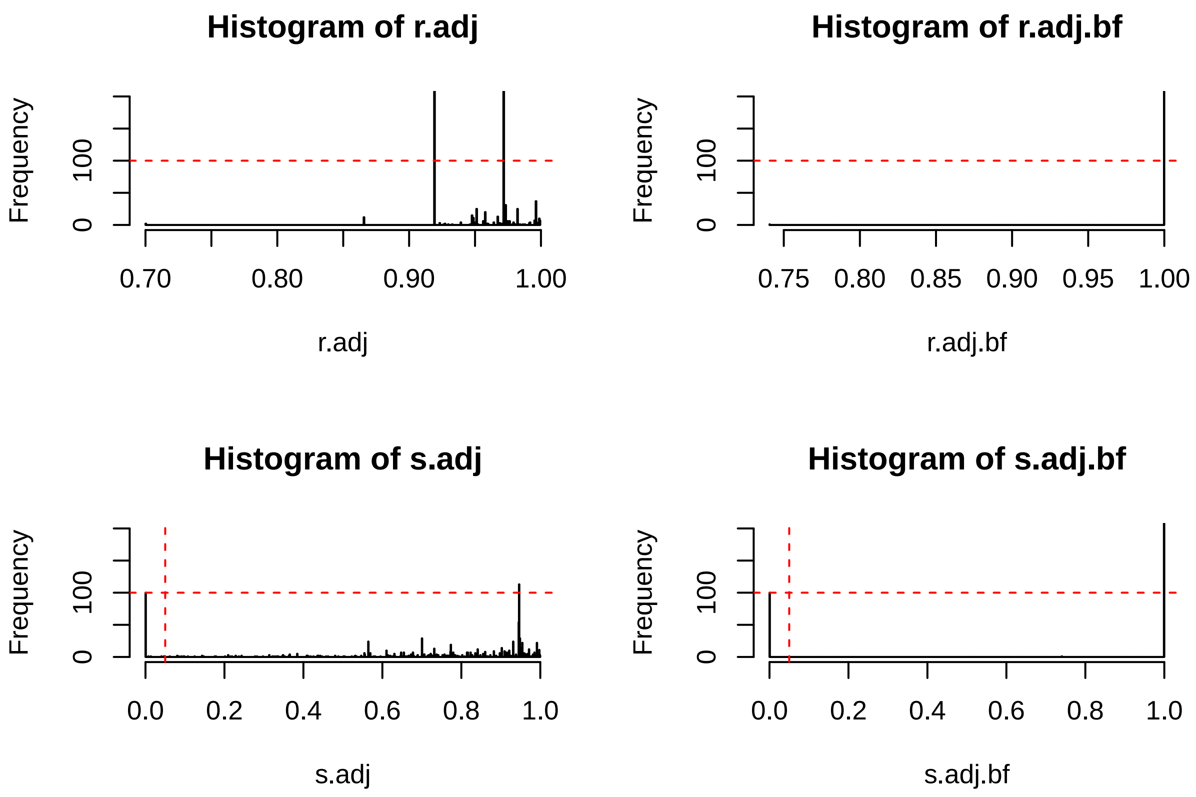
When conducting multiple statistical tests, we assume independence of the tests. There are two main ways of correcting for multiple hypothesis tests.

The first and simplest is the Bonferroni method. This is referred to as the family wide error rate. The idea here is that no adjusted p-value has a false positive likelihood greater than the chosen critical value. Basically, the p-value is multipled by the number of hypothesis tests, and if the value is greater than 1, it is set to 1.

The second is the Benjamini-Hochberg method. This is referred to as a false positive probability. The idea here is that each value has a likelihood of being a false positive that corresponds to its adjusted p-value (or q score).

```r
r.adj <- p.adjust(r.t, method="BH")
s.adj <- p.adjust(s.t, method="BH")
r.adj.bf <- p.adjust(r.t, method="bonferroni")
s.adj.bf <- p.adjust(s.t, method="bonferroni")

par(mfrow=c(2,2))
hist(r.adj, breaks=1000, ylim=c(0,(n.sig*2)))
abline(v=0.05, col="red", lty=2)
abline(h=n.sig, col="red", lty=2)

hist(r.adj.bf, breaks=1000, ylim=c(0,(n.sig*2)))
abline(v=0.05, col="red", lty=2)
abline(h=n.sig, col="red", lty=2)

hist(s.adj, breaks=1000, ylim=c(0,(n.sig*2)))
abline(v=0.05, col="red", lty=2)
abline(h=n.sig, col="red", lty=2)
```

```
hist(s.adj.bf, breaks=1000, ylim=c(0,(n.sig*2)))
abline(v=0.05, col="red", lty=2)
abline(h=n.sig, col="red", lty=2)
```



Which method is most effective for this data?

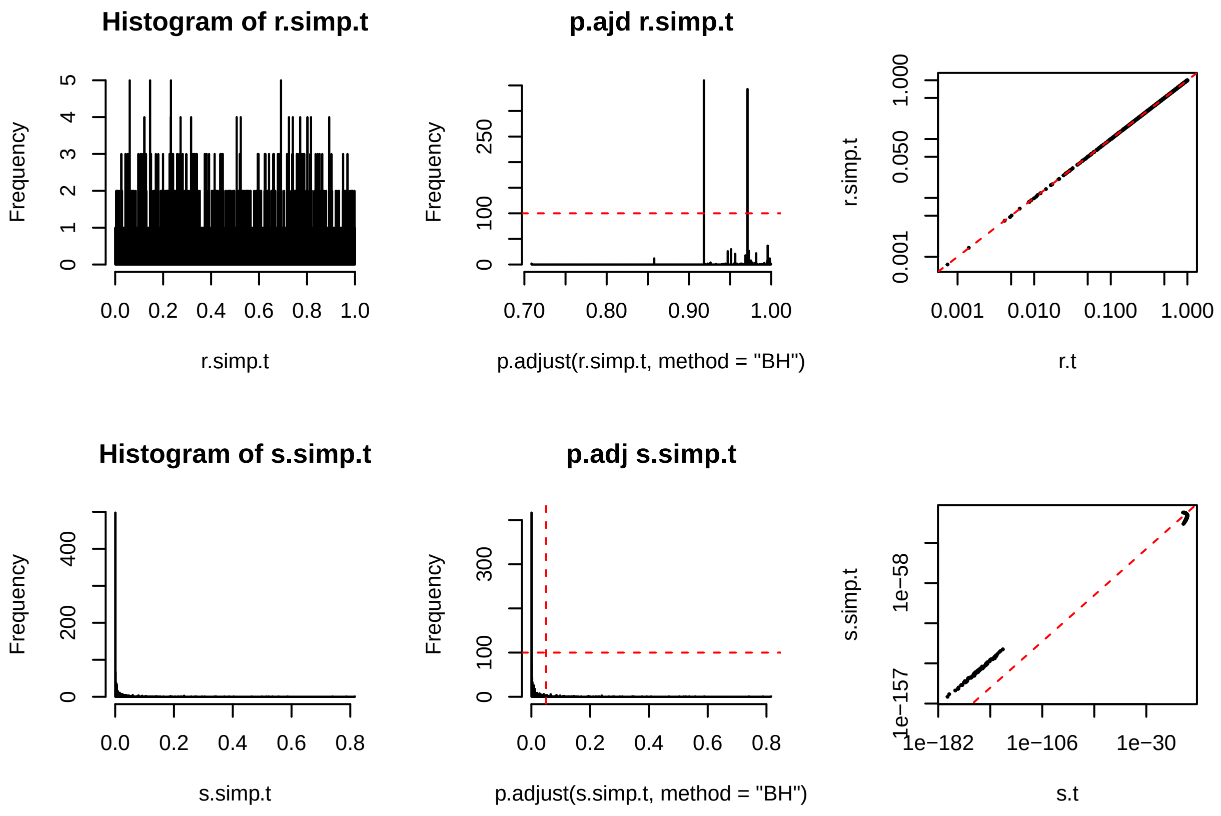
Which method would you use for your analyses? Why?

Non-independence in high throughput sequencing.

It is common, in fact mandated, in high throughput sequencing to normalize the number of read counts in each sample to a constant number. In RNA-seq this is called 'count normalization' in 16S rRNA gene sequencing this is usually done by count normalization, rarefaction (subsampling without replacement) or by converting the counts to proportions.

```
# the effect of non-independence on P values in high throughput sequencing

r.simp <- t(apply(r, 1, function(x) x/sum(x)))

r.simp.t <- apply(r.simp, 2, function(x) as.numeric(t.test(x[1:breaks], x[(breaks+1):n.samples])[3]) )

s.simp <- t(apply(s, 1, function(x) x/sum(x)))

s.simp.t <- apply(s.simp, 2, function(x) as.numeric(t.test(x[1:breaks], x[(breaks+1):n.samples])[3]) )

par(mfrow=c(2,3))
hist(r.simp.t, breaks=1000)
```

```r
hist(p.adjust(r.simp.t, method="BH"), breaks=1000, main="p.ajd r.simp.t")
abline(v=0.05, col="red", lty=2)
abline(h=n.sig, col="red", lty=2)
plot(r.t, r.simp.t, pch=19, cex=0.2, log="xy")
abline(0,1, col="red", lty=2)

hist(s.simp.t, breaks=1000)
hist(p.adjust(s.simp.t, method="BH"), breaks=1000, main="p.adj s.simp.t")
abline(v=0.05, col="red", lty=2)
abline(h=n.sig, col="red", lty=2)
plot(s.t, s.simp.t, pch=19, cex=0.2, log="xy")
abline(0,1, col="red", lty=2)
```



There is now no longer a relationship between the constant sum data and the count data it was derived from when some of the values have real changes. What has happened here???

Play with the values for n.sig, and second.mean to see what is going on.