CS395T
Computational Statistics with Application to Bioinformatics

Prof. William H. Press
Spring Term, 2011
The University of Texas at Austin

Lecture 19

Protocol 1: Retrospective analysis or “case/control study”

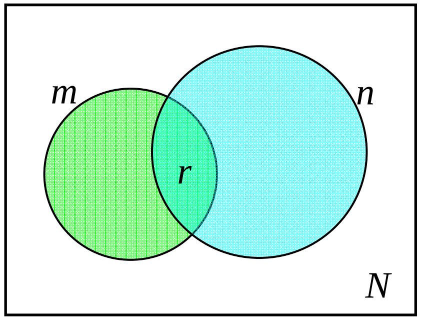
C_1 already has the disease. We retrospectively look at their factors.

In the null hypothesis, both columns share row probabilities q and (1-q). But we don’t know q. It’s a “nuisance parameter”.

	C_0	C_1	
q	$\text{bin}_q(n_{.0}, n_{00})$	$\text{bin}_q(n_{.1}, n_{01})$	$n_{0.}$
$(1-q)$	$\surd$	$\surd$	$n_{1.}$
	$n_{.0}$ (fixed)	$n_{.1}$ (fixed)	$n_{..}$ (fixed)

	C_0	C_1	
f_0	n_{00}	n_{01}	$n_{0.}$
f_1	n_{10}	n_{11}	$n_{1.}$
	$n_{.0}$	$n_{.1}$	$n_{..}$

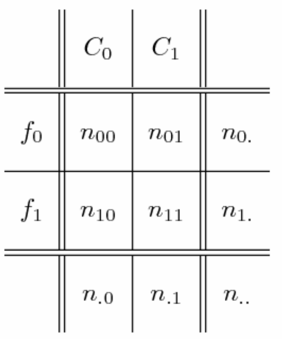
$$
\begin{aligned}
P(\text{table}) &= \text{bin}_q(n_{.0}, n_{00})\text{bin}_q(n_{.1}, n_{01}) \\
&= \binom{n_{.0}}{n_{00}} q^{n_{00}}(1-q)^{n_{10}} \binom{n_{.1}}{n_{01}} q^{n_{01}}(1-q)^{n_{11}} \\
&= \frac{n_{.0}!n_{.1}!}{n_{00}!n_{01}!n_{10}!n_{11}!} q^{n_{0.}}(1-q)^{n_{1.}} \\
&= \text{bin}_q(n_{..}, n_{0.}) \times \frac{n_{0.}!\,n_{1.}!\,n_{.0}!\,n_{.1}!}{n_{..}!\,n_{00}!\,n_{01}!\,n_{10}!\,n_{11}!} \\
&\equiv \text{bin}_q(n_{..}, n_{0.}) \times \text{hyper}(n_{00}; n_{..}, n_{.0}, n_{0.}) \\
&= P(n_{0.}\,|\,n_{..}, q) \times P(\text{table}\,|\,n_{0.}, n_{..})
\end{aligned}
$$



$$
\text{hyper}(n_{00}; n_{..}, n_{.0}, n_{0.}) = \frac{\binom{n_{.0}}{n_{00}}\binom{n_{.1}}{n_{01}}}{\binom{n_{..}}{n_{0.}}}
$$

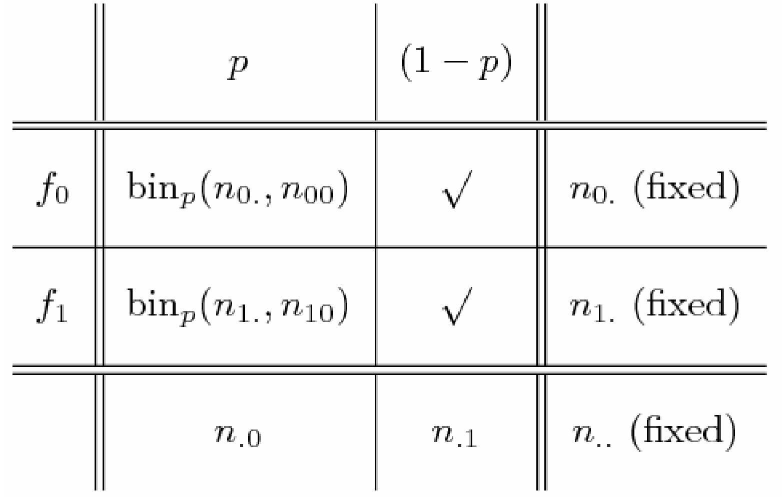
Protocol 2: Prospective experiment or “longitudinal study”

Identify samples with the factors, then watch to see who gets the disease



	C_0	C_1	
f_0	n_{00}	n_{01}	$n_{0.}$
f_1	n_{10}	n_{11}	$n_{1.}$
	$n_{.0}$	$n_{.1}$	$n_{..}$

In the null hypothesis, both rows share row probabilities p and (1-p). But we don’t know p. It’s now the nuisance parameter.



	p	$(1-p)$	
f_0	$\text{bin}_p(n_{0.}, n_{00})$	$\surd$	$n_{0.}$ (fixed)
f_1	$\text{bin}_p(n_{1.}, n_{10})$	$\surd$	$n_{1.}$ (fixed)
	$n_{.0}$	$n_{.1}$	$n_{..}$ (fixed)

$$\begin{aligned} P(\text{table}) &= \text{bin}_p(n_{0.}, n_{00})\text{bin}_q(n_{1.}, n_{10}) \\ &= \text{bin}_p(n_{..}, n_{.0}) \times \frac{n_{0.}!\, n_{1.}!\, n_{.0}!\, n_{.1}!}{n_{..}!\, n_{00}!\, n_{01}!\, n_{10}!\, n_{11}!} \\ &= P(n_{.0}\,|\,n_{..}, p) \times P(\text{table}\,|\,n_{.0}, n_{..}) \end{aligned}$$

Protocol 3: Cross-sectional or snapshot study (no fixed marginals)

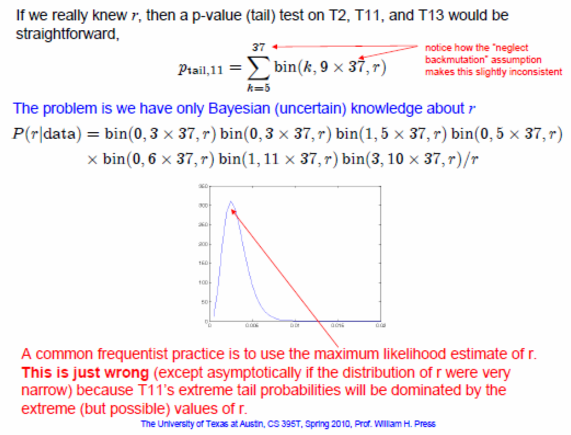
E.g., test all Austin residents for both disease and factors.

	C_0	C_1	
f_0	n_{00}	n_{01}	$n_{0.}$
f_1	n_{10}	n_{11}	$n_{1.}$
	$n_{.0}$	$n_{.1}$	$n_{..}$

multinomial distribution

$$\begin{aligned}P(\text{table}) &= \frac{n_{..}!}{n_{00}!\,n_{01}!\,n_{10}!\,n_{11}!}[pq]^{n_{00}}[(1-p)q]^{n_{01}}[(1-q)p]^{n_{10}}[(1-p)(1-q)]^{n_{11}}\\ &= \text{bin}_p(n_{..},n_{.0})\text{bin}_q(n_{..},n_{0.})\times\frac{n_{0.}!\,n_{1.}!\,n_{.0}!\,n_{.1}!}{n_{..}!\,n_{00}!\,n_{01}!\,n_{10}!\,n_{11}!}\\ &= P(n_{.0}\,|\,n_{..},p)P(n_{0.}\,|\,n_{..},q)\times P(\text{table}\,|\,n_{.0},n_{0.}n_{..})\end{aligned}$$

Asymptotic methods (e.g. chi-square) are typically equivalent to making point ML estimates of p,q, and thus the nuisance factors, from the data itself. Remember when we encountered this before?



If we really knew r, then a p-value (tail) test on T2, T11, and T13 would be straightforward,

$$p_{\text{tail},11} = \sum_{k=5}^{37} \text{bin}(k, 9\times 37, r)$$

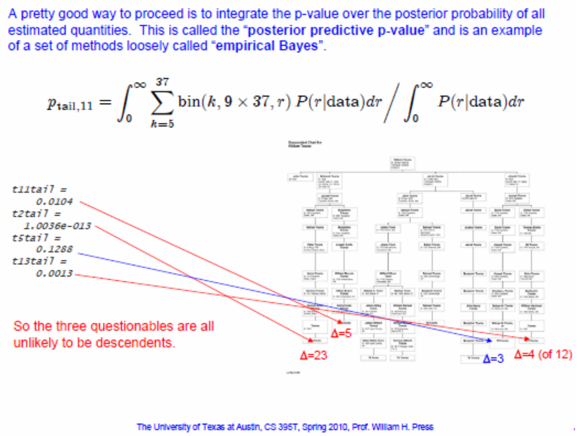
notice how the "neglect backmutation" assumption makes this slightly inconsistent

The problem is we have only Bayesian (uncertain) knowledge about r

$$P(r|\text{data}) = \text{bin}(0,3\times 37,r)\,\text{bin}(0,3\times 37,r)\,\text{bin}(1,5\times 37,r)\,\text{bin}(0,5\times 37,r)$$
$$\times\,\text{bin}(0,6\times 37,r)\,\text{bin}(1,11\times 37,r)\,\text{bin}(3,10\times 37,r)/r$$

A common frequentist practice is to use the maximum likelihood estimate of r. **This is just wrong** (except asymptotically if the distribution of r were very narrow) because T11's extreme tail probabilities will be dominated by the extreme (but possible) values of r.

The University of Texas at Austin, CS 395T, Spring 2010, Prof. William H. Press



A pretty good way to proceed is to integrate the p-value over the posterior probability of all estimated quantities. This is called the "**posterior predictive p-value**" and is an example of a set of methods loosely called "**empirical Bayes**".

$$p_{\text{tail},11} = \int_0^\infty \sum_{k=5}^{37} \text{bin}(k, 9\times 37, r)\, P(r|\text{data})dr \Big/ \int_0^\infty P(r|\text{data})dr$$

```
t11tail =
    0.0104
t2tail =
  1.0036e-013
t5tail =
    0.1288
t13tail =
    0.0013
```

So the three questionables are all unlikely to be descendents.

The University of Texas at Austin, CS 395T, Spring 2010, Prof. William H. Press

Digression on the multinomial distribution

On each i.i.d. try, exactly one of *K* outcomes occurs, with probabilities

$$p_1, p_2, \ldots, p_K \qquad \sum_{i=1}^{K} p_i = 1$$

For *N* tries, the probability of seeing exactly the outcome

$$n_1, n_2, \ldots, n_K \qquad \sum_{i=1}^{K} n_i = N$$

is

$$P(n_1, \ldots, n_K | N, p_1, \ldots, p_K) = \underbrace{\frac{N!}{n_1! \cdots n_K!}}_{\text{number of equivalent arrangements}} \overbrace{p_1^{n_1} p_2^{n_2} \cdots p_K^{n_K}}^{\text{probability of one specific outcome}}$$

N=26: `abcde fgh ijklmnop q rs tuvwxyz` *N!* arrangements

`(12345)(123)(12345678)(1)(12)(1234567)` partition into the observed n_i's

$n_1 = 5$ $n_2 = 3$ $n_6 = 7$

So, in all three cases we got a product of “nuisance” probabilities (depending on unknown p or q or both) and a “sufficient statistic” conditioned on all the marginals.

Fisher’s Exact Test just throws away the nuisance factors and uses the sufficient statistic:

	C_0	C_1	
f_0	n_{00}	n_{01}	$n_{0.}$
f_1	n_{10}	n_{11}	$n_{1.}$
	$n_{.0}$	$n_{.1}$	$n_{..}$

$$P(\text{table}\,|\,n_{0.},n_{.0},n_{..}) = \frac{n_{0.}!\,n_{1.}!\,n_{.0}!\,n_{.1}!}{n_{..}!\;n_{00}!\,n_{01}!\,n_{10}!\,n_{11}!}$$

This can also be seen to be the (purely combinatorial) probability of the table with all marginals fixed:

$$P(k\,|\,n_{0.},n_{.0},n_{..}) = \frac{\binom{n_{..}}{n_{.0}}\binom{n_{.0}}{k}\binom{n_{.1}}{n_{0.}-k}}{\binom{n_{..}}{n_{.0}}\sum_k\binom{n_{.0}}{k}\binom{n_{.1}}{n_{0.}-k}} = \frac{\binom{n_{.0}}{k}\binom{n_{.1}}{n_{0.}-k}}{\binom{n_{..}}{n_{0.}}}$$

table is fully determined by k alone

Numerator: number of partitions with n_{00}=k
Denominator: sum numerator over k
With all marginals fixed, n_{00} determines the whole table.

Vandermonde’s identity:

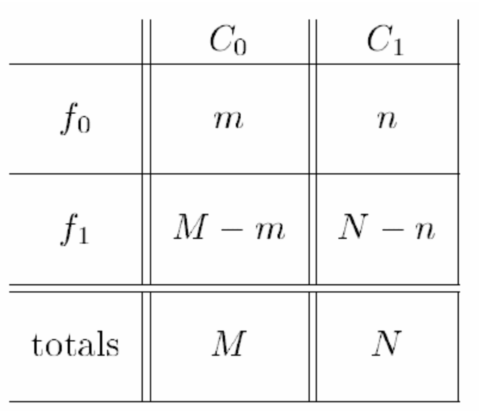
$$\binom{n+m}{r} = \sum_{k=0}^{r}\binom{n}{k}\binom{m}{r-k}.$$

Proof: How many ways can you choose a subcommittee of size r from a committee with n Democrats and m Republicans?

How many Democrats on the subcommittee?

A statistic is sufficient "when no other statistic which can be calculated from the same sample provides any additional information as to the value of the parameter".

That was all about the distribution of tables in the **null hypothesis**.
Now we complete the rest of the tail test paradigm:
The most popular choice for a statistic for 2x2 tables is the “Wald statistic”:



	C_0	C_1
f_0	m	n
f_1	$M-m$	$N-n$
totals	M	N

(sorry for the slight change in notation!)

$$T = \frac{\widehat{p}_1 - \widehat{p}_2}{\sqrt{\widehat{p}(1-\widehat{p})(M^{-1}+N^{-1})}}$$

This is constructed so that it will asymptotically became a true t-value.

$$\widehat{p}_1 \equiv m/M, \quad \widehat{p}_2 \equiv n/N, \quad \widehat{p} \equiv (m+n)/(M+N)$$

Notice that this is monotonic with m when all marginals are fixed.

You could instead use the Pearson (chi-square) statistic,
but <u>not</u> the assumption that it is chi-square distributed.

So, the Fisher Exact Test looks like this:

- Compute the statistic for the data
- Loop over all possible contingency tables with the same marginals
  - for 2x2 there is just one free parameter
- Compute the statistic for each table in the loop
- Accumulate weight (by hypergeometric probability) of statistic <, =, > the data statistic
- Output the p-value (or, because of discreteness effects, the range)

Is this table a significant result?

	C_0	C_1
f_0	8	3
f_1	16	26

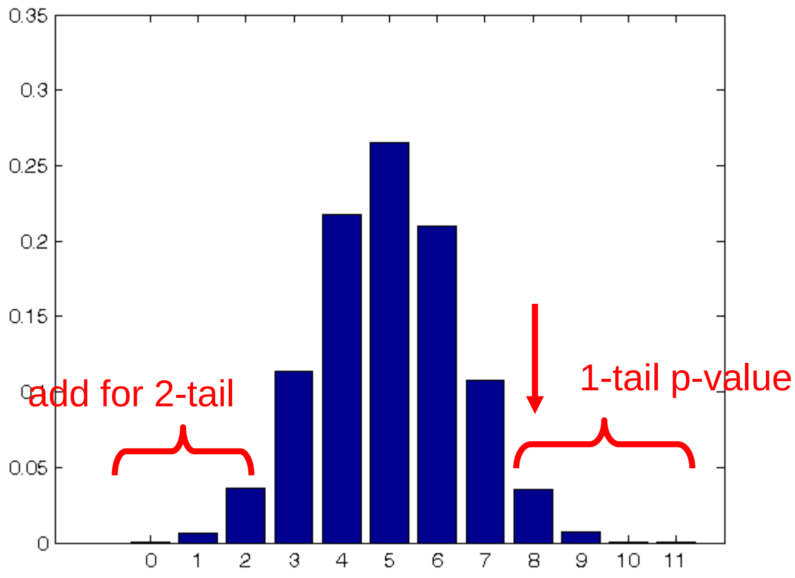
	C_0	C_1
f_0	m	$11-m$
f_1	$24-m$	$18+m$

$$P(m) = \binom{24}{m}\binom{29}{11-m} \Big/ \binom{53}{11}, \qquad 0 \le m \le 11$$

Actually, here in the 2x2 case, all statistics monotonic in m are equivalent (except for some two-tail issues)!
So the test statistic only matters in the case of larger tables, when there is more than one degree of freedom (with fixed marginals).

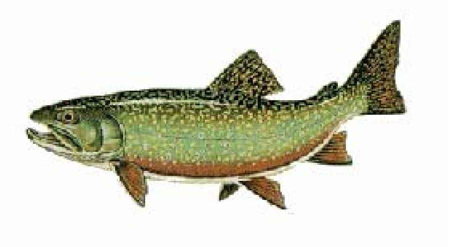
Compute Fisher Exact Test for our table

```
myprob = @(m) nchoosek(24,m)*nchoosek(29,11-m)/nchoosek(53,11);
ms = 0:11
ps = arrayfun(myprob,ms)
ms =
     0     1     2     3     4     5     6     7     8     9    10    11
ps =
0.0005    0.0063    0.0363    0.1140    0.2176    0.2649    0.2097    0.1078    0.0353
0.0070    0.0007    0.0000
[sum(ps(1:8)) ps(9) sum(ps(10:12))]
ans =
    0.9570    0.0353    0.0077
sum(ps(9:12))
ans =
    0.0430
bar(ms,ps)
```

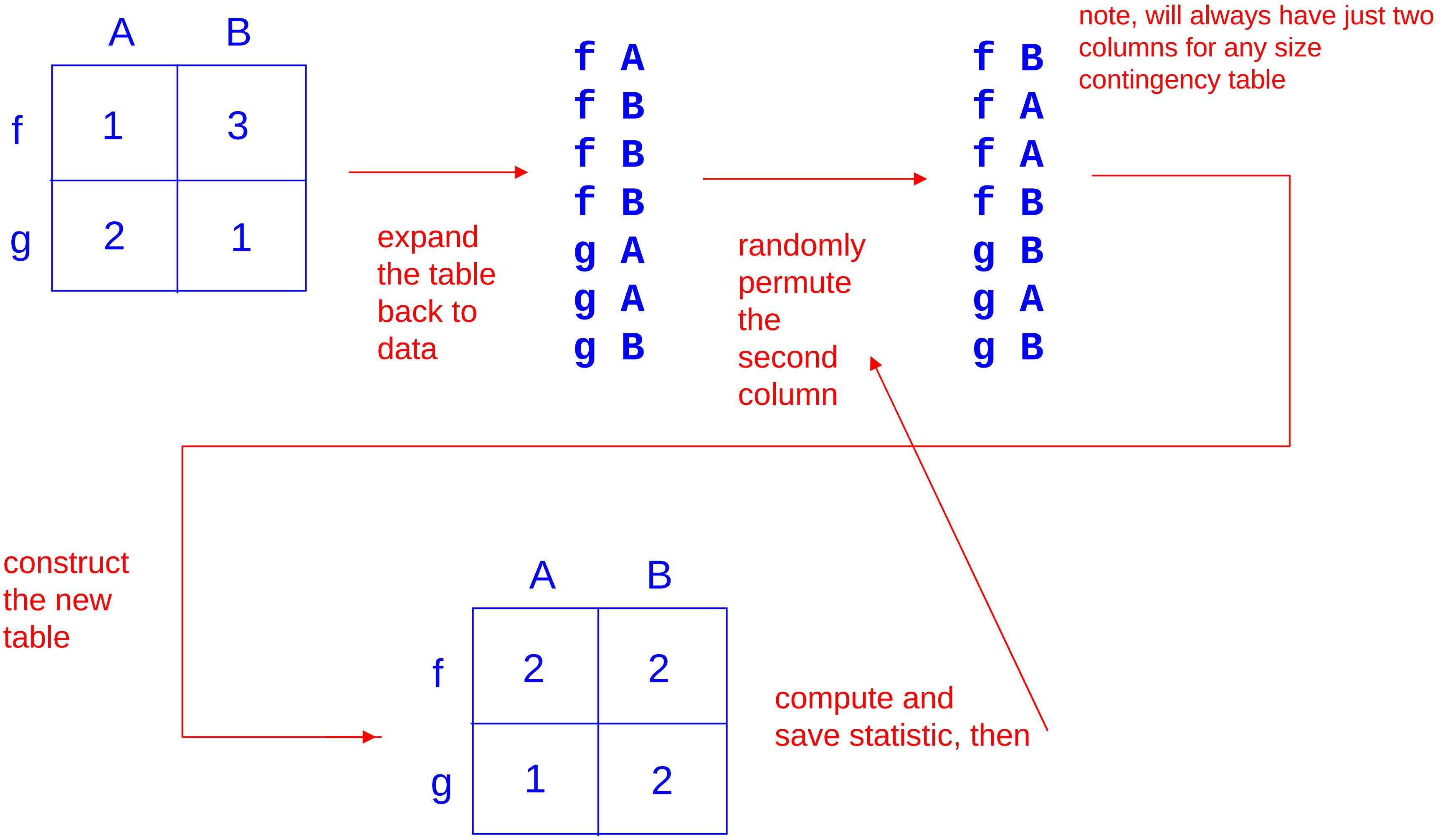


Editorial: We will next learn an efficient way to compute the Fisher Exact test. But despite the words "Fisher" (true) and "exact" (questionable) in its name, this test isn't conceptually well grounded, since virtually <u>never</u> are all marginals held fixed (none of Protocols 1,2,3 above)! At best it is an approximation that ignores the nuisance parameters (p and/or q).

I don't understand why Fisher Exact is so widely used. I think it is historical accident, due to outdated frequentist worship of sufficient statistics!



An computational alternative to the Fisher Exact Test is the Permuation Test. The idea is to break any association between the row and column variables by shuffling. This is allowed under the null hypothesis of no association!



	A	B
f	1	3
g	2	1

expand the table back to data

f	A
f	B
f	B
f	B
g	A
g	A
g	B

randomly permute the second column

f	B
f	A
f	A
f	B
g	B
g	A
g	B

note, will always have just two columns for any size contingency table

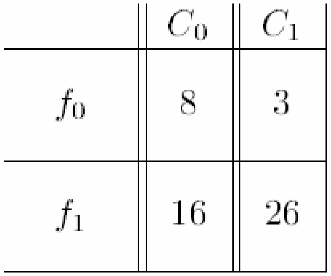
construct the new table

	A	B
f	2	2
g	1	2

compute and save statistic, then

(notice that all marginals are preserved – will come back to this point)

Aha! The permutation preserves all marginals. In fact, it is a Monte Carlo calculation of the Fisher Exact Test. And it is easy to compute for any size table!



	C_0	C_1
f_0	8	3
f_1	16	26

```
function t = wald(tab)
m = tab(1,1);
n = tab(1,2);
mm = m + tab(2,1);
nn = n + tab(2,2);
p1 = m/mm;
p2 = n/nn;
p = (m+n)/(mm+nn);
t = (p1-p2)/sqrt(p*(1-p)*(1/mm+1/nn));
```

Code up the Wald statistic.

```
table = [8 3; 16 26;]
table =
     8     3
    16    26

tdata = wald(table)
tdata =
       2.0542
```

The data show about a 2 standard deviation effect, except that they’re not really standard deviations because of the small counts!

In scientific papers, people can equally well say, “Fisher Exact test” or “Permutation test”. You might think that the former sounds more learned, but to me it sounds like they don’t know exactly what their test actually did!

Expand the table and generate permutations:

```
[row col] = ndgrid(1:2,1:2);
```

This tells each cell its row and column number

```
d = [];
for k=1:numel(table); d = cat(1,d,repmat([row(k),col(k)],table(k),1)); end;
```

(Darn it, I couldn't think of a way to do this in Matlab without an explicit loop, thus spoiling my no-loop record)*

```
size(d)
ans =
    53     2
accumarray(d,1,[2,2])
```

Check that we recover the original table.

```
ans =
     8     3
    16    26

gen = @(x) wald(accumarray( [d(randperm(size(d,1)),1) d(:,2)] ,1,[2,2]));

gen(1)
ans =
      -0.6676
```

Try one permutation just to see it work.

```
perms = arrayfun(gen,1:100000);
```

It's fast, so can easily do lots of permuations.

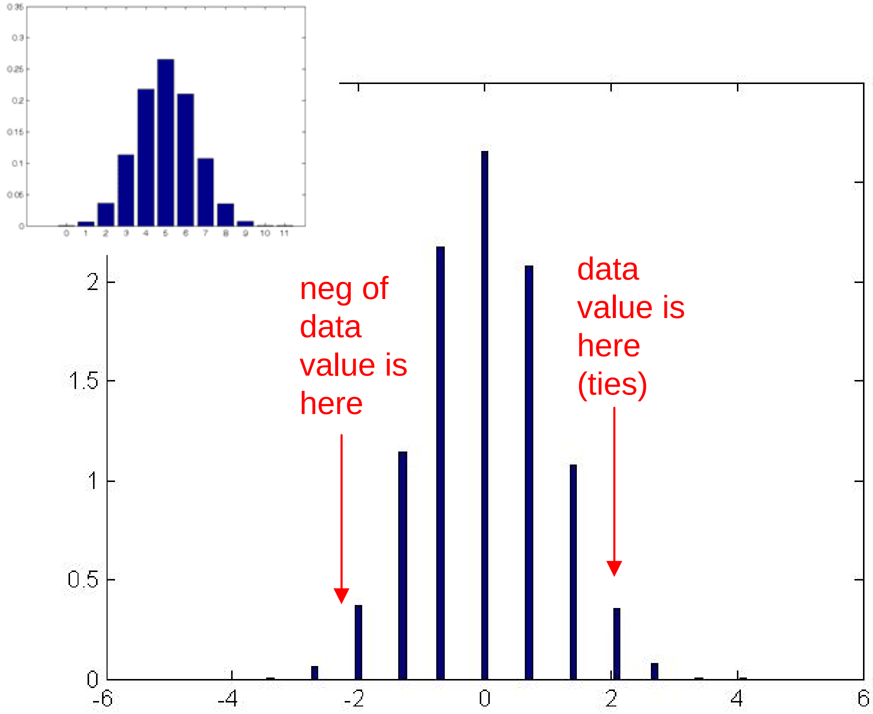
```
hist(perms,(-4:.1:5))
cdfplot(perms)
```

*Peter Perkins (MathWorks) suggests the wonderfully obscure

```
d = [rldecode(table,row) rldecode(table,col)];
```

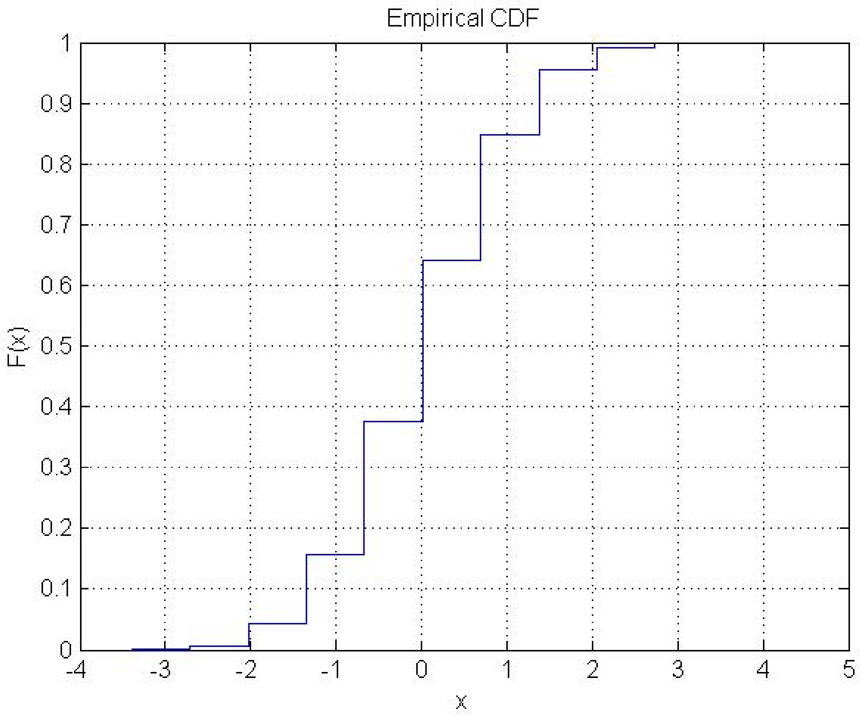
where rldecode is one of Peter Acklam's Matlab Tips and Tricks.

We get discrete values because only a few discrete tables are possible.



neg of data value is here

data value is here (ties)



```
pvalgt = numel(perms(perms>wald(table)))/numel(perms)
pvalge = numel(perms(perms>=wald(table)))/numel(perms)
pvaltt = (numel(perms(perms > wald(table)))+numel(perms(perms < -wald(table))))/numel(perms)
pvaltte = (numel(perms(perms >= wald(table)))+numel(perms(perms <= -wald(table))))/numel(perms)
pvalwrongtail = numel(perms(perms> -wald(table)))/numel(perms)
```

```
pvalgt =  0.00812
pvalge =  0.04382
pvaltt =  0.01498
pvaltte =  0.05068
pvalwrongtail =  0.99314
```

We reproduce values from Fisher Exact test done combinatorially