CS395T
Computational Statistics with Application to Bioinformatics

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

Lecture 20

To learn more, let's play with the first contingency table we looked at:

TABLE 1
Maternal drinking and congenital malformations

	Alcohol consumption (average no. of drinks/day)				
Malformation	0	< 1	1–2	3–5	≥ 6
Absent	17,066	14,464	788	126	37
Present	48	38	5	1	1

Source: Graubard and Korn (1987).

Different "standard methods" applied to this data get p-values ranging from 0.005 to 0.190. (Agresti 1992)

Fisher Exact Test done combinatorially is not a viable option (both because of computational workload and because we only derived the 2x2 case!)

So we'll try the (equivalent) permutation test.

Expand the table and generate 1000 permutations

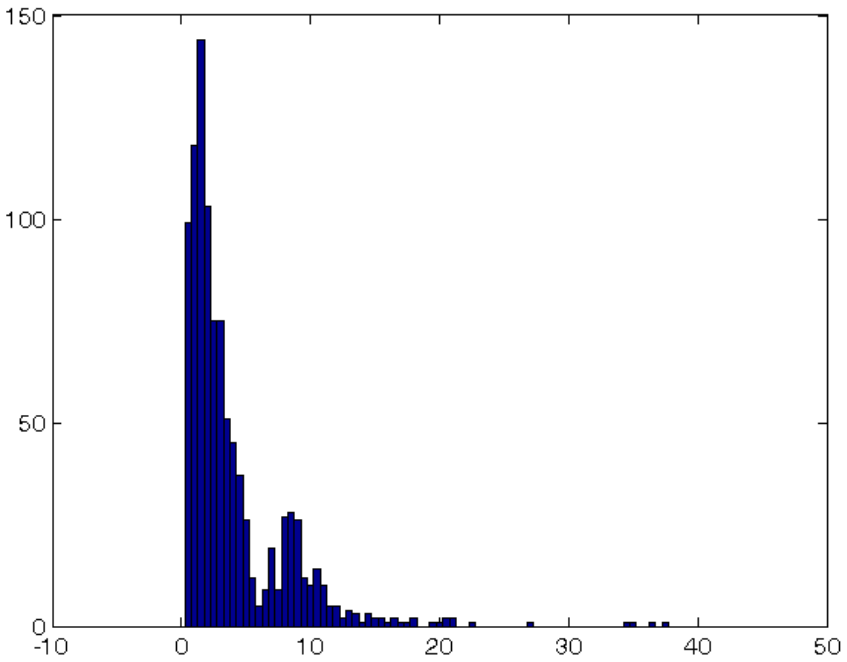
(Now takes ~1 min. Go figure out how to do the permutation test without expanding all the data!)

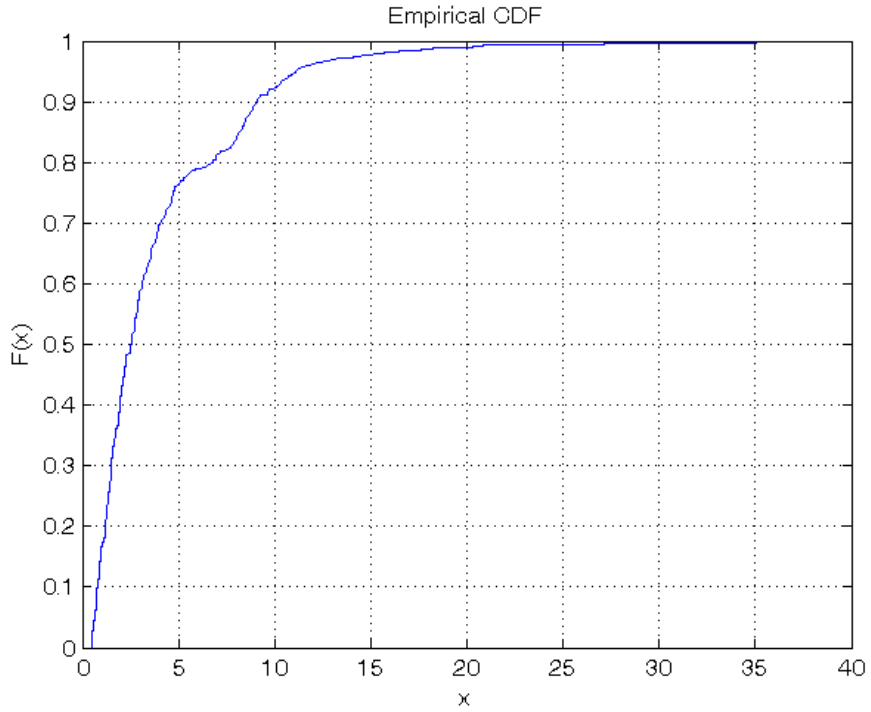
```
table = [17066 14464 788 126 37; 48 38 5 1 1]
table =
       17066       14464         788         126          37
          48          38           5           1           1

pearson(table)
ans =
   12.0821

[row col] = ndgrid(1:size(table,1),1:size(table,2));
d = [];
for k=1:numel(table); d = cat(1,d,repmat([row(k),col(k)],table(k),1)); end;
size(d)
ans =
       32574                    Yes, has the dimensions we expect.
           2
tablecheck = accumarray(d,1,size(table))                And we can reconstruct the original table.
tablecheck =
       17066       14464         788         126          37
          48          38           5           1           1

gen = @(x) pearson( accumarray([ d(randperm(size(d,1)),1) d(:,2)],1,size(table)));
gen(1)
ans =
    1.3378

perms = arrayfun(gen, 1:1000);
```

```
hist(perms,(0:.5:40))

cdfplot(perms)

pvalgt = numel(perms(perms>pearson(table)))/numel(perms)
pvalge = numel(perms(perms>=pearson(table)))/numel(perms)
pvalgt =
    0.0380
pvalge =
    0.0380
```

```
pearson(table)
ans =
   12.0821
```

our answer: the p-value





Two questions remain:

1. How good or bad an approximation was it to hold all marginals fixed?
2. Is there a more powerful statistical test for this data?

The more powerful statistical approach to the maternal drinking contingency table is to recognize that the table is ordinal, not just nominal

- Choose a test statistic that actually reflects your hypothesis!
  - the columns are ordered by an increasing independent variable
  - "more drinks lead to more abnormalities"
  - the obvious statistic is "difference of mean number of drinks between the two rows"
  - if a threshold effect is plausible, might also try "difference of mean of square"
    - we will discuss multiple hypothesis correction
- With this different statistic, we do a permutation test as before

TABLE 1
Maternal drinking and congenital malformations

	Alcohol consumption (average no. of drinks/day)				
Malformation	0	< 1	1–2	3–5	≥ 6
Absent	17,066	14,464	788	126	37
Present	48	38	5	1	1

Source: Graubard and Korn (1987).

Input the table and display the means and their differences:

```
table = [17066 14464 788 126 37; 48 38 5 1 1]
sum(table(:))
table =
       17066       14464         788         126          37
          48          38           5           1           1
ans =
       32574
drinks = [0 0.5 1.5 4. 6.];
drinksq = drinks.^2;
norm = sum(table,2);
mudrinks = (table * drinks')./norm
mudrinksq = (table * drinksq')./norm
mudrinks =
        0.2814
       0.39247
mudrinksq =
       0.26899
       0.78226
diff = [-1 1] * mudrinks
diffsq = [-1 1] * mudrinksq
diff =
       0.11108
diffsq =
       0.51327
```

reasonable quantification of the ordinal categories: exactness isn't important, since we get to define the statistic

These are our chosen "statistics". The question is: Are either of them statistically significant? We'll use the permutation test to find out.

TABLE 1
Maternal drinking and congenital malformations

Malformation	Alcohol consumption (average no. of drinks/day)				
	0	< 1	1–2	3–5	≥ 6
Absent	17,066	14,464	788	126	37
Present	48	38	5	1	1

Source: Graubard and Korn (1987).

Expand table back to dataset of length 32574:

```
[row col] = ndgrid(1:2,1:5)
row =
     1     1     1     1     1
     2     2     2     2     2
col =
     1     2     3     4     5
     1     2     3     4     5

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

size(d)
ans =
       32574           2

accumarray(d,1,[2,5])
ans =
       17066       14464         788         126          37
          48          38           5           1           1

mean(drinks(d(d(:,1)==2,2)))
ans =
      0.39247
```

This tells each cell its row and column number

Yes, has the dimensions we expect.

And we get the right mean, so it looks like we are good to go…

And we can reconstruct the original table.

TABLE 1
Maternal drinking and congenital malformations

	Alcohol consumption (average no. of drinks/day)				
Malformation	0	< 1	1–2	3–5	≥ 6
Absent	17,066	14,464	788	126	37
Present	48	38	5	1	1

Source: Graubard and Korn (1987).

Compute the statistic for the data and for 1000 permuations:

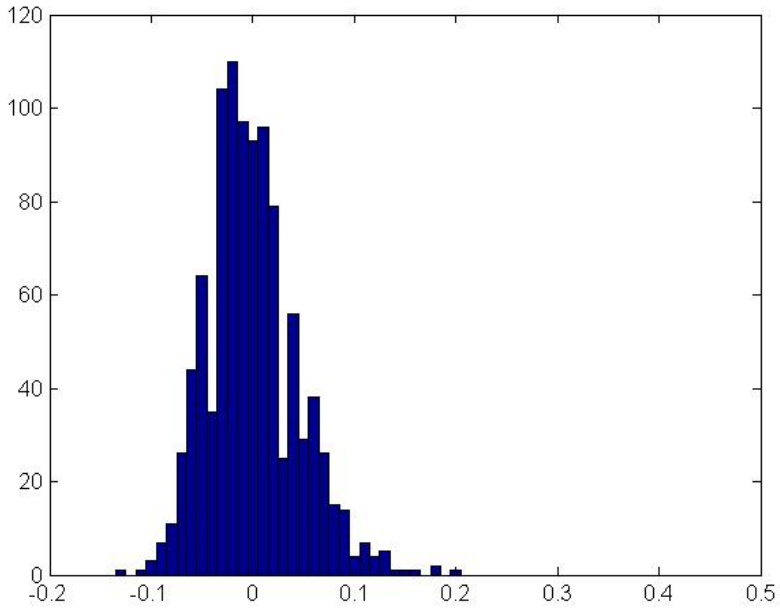
As before, the idea is to sample from the null hypothesis (no association) while keeping the distributions of each single variable unchanged. Do this by permuting a label that is irrelevant in the null hypothesis.

```
diffmean = @(d) mean(drinks(d(d(:,1)==2,2))) - mean(drinks(d(d(:,1)==1,2)));
diffmean(d)
ans =
      0.11108
diffmean([d(randperm(size(d,1)),1) d(:,2)])
ans =
     0.014027
perms = arrayfun(@(x) diffmean([d(randperm(size(d,1)),1) d(:,2)]), [1:1000]);

pval = numel(perms(perms>diffmean(d)))/numel(perms)
pval =
        0.015
hist(perms,(-.15:.01:.3))
```

Try one permutation just to see it work.

So, as a p-value, the association is now more than twice as significant as when we ignored the column ordering. We were throwing away useful information!

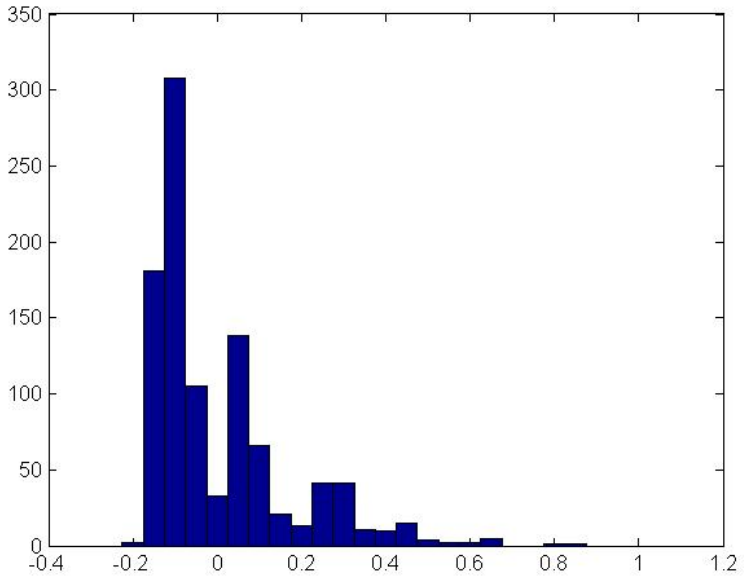
Reminder: p-value is a false positive rate.



Same analysis for the squared-drinks statistic:

```
diffmeansq = @(d) mean(drinksq(d(d(:,1)==2,2))) - mean(drinksq(d(d(:,1)==1,2)));
diffmeansq(d)
ans =
      0.51327
permsq = arrayfun(@(x) diffmeansq([d(randperm(size(d,1)),1) d(:,2)]), [1:1000]);
pval = numel(permsq(permsq>diffmeansq(d)))/numel(permsq)
pval =
        0.011
hist(permsq,(-.3:.05:1))
```

- Should we apply a multiple hypothesis correction to both pval's (mult x 2) ? Probably not.
  - mean and mean-of-squares highly correlated, and
  - the previous result was significant
  - we're not just shopping uniform p-values
- But, if your data can stand it, Bonferroni is the gold standard
- Is there a principled way to do multiple hypothesis correction with highly correlated tests?



The permutation test is not bootstrap resampling! Permutation test breaks the causal connection, giving the null hypothesis. Bootstrap doesn't, but tells us how much variation in the signal one might see in repeated identical experiments. Bootstrap might *possibly* be useful in understanding why another experiment didn't see the effect (false negative).

```
diffmean(d(randsample(size(d,1),size(d,1),true),:))
ans =
      0.20703
```

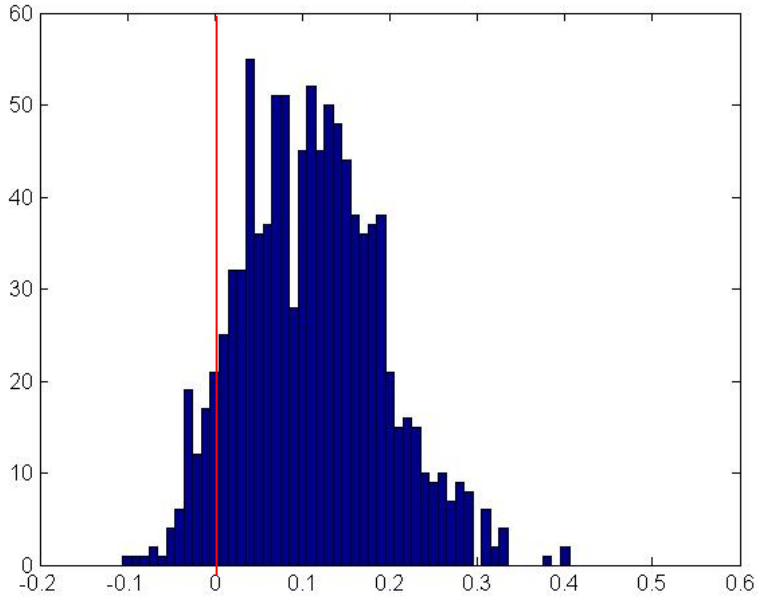
Try one resample just to see it work.

```
resamp = arrayfun(@(x) diffmean(d(randsample(size(d,1),size(d,1),true),:)), [1:1000]);

bias = mean(resamp) - diffmean(d)
resamp = resamp - bias;
pval = numel(resamp(resamp<0))/numel(resamp)
bias =
    0.0014876
pval =
       0.078

hist(resamp,[-.1:.01:.5])
```

If this is large, we should worry.

This isn't really a pval. (No null hypothesis.)

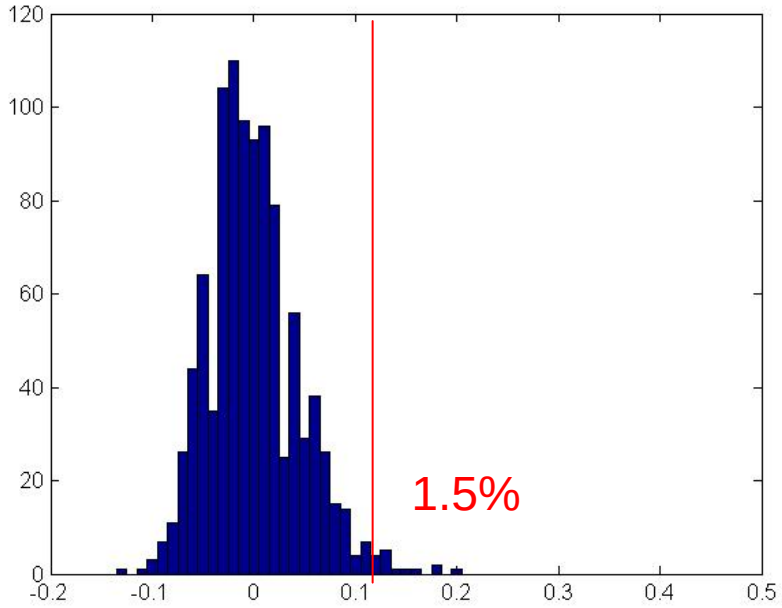
```
diffmean(d)
 = 0.11108
```

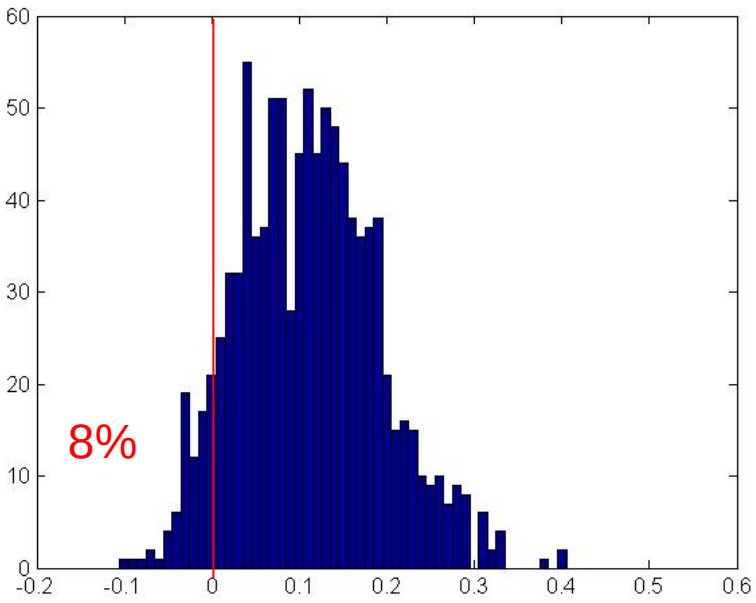


Now the "pval" is a false negative rate
How often would a repetition of the experiment show an effect with negative difference of the means?

So: Bootstrap resampling and sampling from the null hypothesis (e.g by permutation) are completely different things!

Distributions of the difference of mean drinks:





Permutation Test

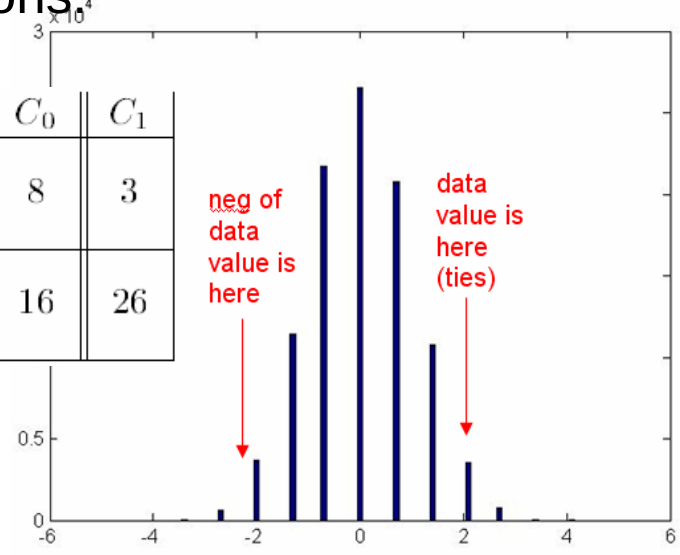
false positive rate,
i.e., significance

Bootstrap

false negative rate,
i.e. for other similar experiments

Summary: permutation tests (a.k.a. Fisher Exact) are easy to do and useful. But, if numbers of counts are small, these tests are less "exact" than they pretend to be, for several related reasons:

	C_0	C_1
f_0	8	3
f_1	16	26



- Because your data value <u>always</u> lands on a tie, it's either over-conservative or under-conservative
  - some people split the difference
- Because the negative of your data value (almost) <u>never</u> lands on a tie, the two-tailed test is fragile
  - might be virtually the same as one-tailed, as in our example
  - or might be hugely (>> x2) different
- In fact, the whole construct is fragile to irrelevant "number theoretical coincidences" about the values of the marginals
  - adding one data point, or using a slightly different statistic, could radically change p-values
- We've already seen what the fundamental problem is
  - real protocols <u>don't fix both sets of marginals</u>
  - Fisher's elegant elimination of the nuisance parameters p and/or q is a trap
- We actually need to estimate a distribution for the nuisance parameters (*p*'s and/or *q*'s) and marginalizing over them
  - this makes us Bayesians in a non-Bayesian (p-value) world
  - but we've already seen examples of this ("posterior predictive p-value")

How shall we estimate the nuisance parameters p and/or q?

Remember "conjugate distributions"? As before, we want to estimate parameters from observed counts. Beta is conjugate to Binomial:

$$P(n|N,q) = \binom{N}{n} q^n (1-q)^{N-n}$$

$$P(q|N,n) \propto q^n (1-q)^{N-n} P(q)$$ Bayes, with prior.

A "conjugate prior" is one that preserves the functional form of the distribution.

$$P(q) \propto q^\alpha (1-q)^\beta$$ ($\alpha = \beta = 0$ is a perfectly good choice: flat prior on q)

So the conjugate distribution is

$$\begin{aligned} P(q|N,n) &= \frac{q^{n+\alpha}(1-q)^{N-n+\beta}}{\int_0^1 q^{n+\alpha}(1-q)^{N-n+\beta} dq} \\ &= \frac{\Gamma(N+\alpha+\beta+2)}{\Gamma(n+\alpha+1)\Gamma(N-n+\beta+1)} q^{n+\alpha}(1-q)^{N-n+\beta} \\ &\sim \mathrm{Beta}(n+\alpha+1, N-n+\beta+1) \end{aligned}$$

This Beta distribution has

$$\mathrm{mean} = \frac{n+\alpha+1}{N+\alpha+\beta+1} \qquad \mathrm{var} = \frac{(n+\alpha+1)(N-n+\beta+1)}{(N+\alpha+\beta+2)^2(N+\alpha+\beta+3)}$$

Matlab, Mathematica, and NR3 all have methods for generating random Beta deviates