Displays for Statistics 5303

Lecture 15

October 7, 2002

Christopher Bingham, Instructor

612-625-7023 (St. Paul)
612-625-1024 (Minneapolis)

Class Web Page
http://www.stat.umn.edu/~kb/classes/5303

Since power is the probability of obtaining a large F-statistic when H_0 is false, you use the non-central F distribution to calculate power.

Example: α = .01, g = 6, n = 4 and $\zeta_1 = \sum_i \alpha_i^2/\sigma^2 = 5$, when testing $H_0: \alpha_1 = \ldots = \alpha_g$

```
Cmd> g <- 6; n <- 4; df_error <- g*(n-1); df_error
(1)         18      Error DF

Cmd> alpha <- .01  # type I error probability

Cmd> F_alpha <- invF(1 - alpha, g-1, g*(n-1)); F_alpha
(1)      4.2479      Rejection cut-point for F-test

Cmd> zeta1 <- 5   #  n=1 non-centrality parameter
```

cumF() with 4 arguments computes $P(F_{noncen} \leq F_\alpha)$:

```
Cmd> 1 - cumF(F_alpha,g-1,g*(n-1),n*zeta1)
(1)     0.61812
```

power2() avoids finding F_α

```
Cmd> power2(n*zeta1,g-1,alpha,g*(n-1))
(1)     0.61812
```

power() is a short cut to **power()** for CRD and RCB.

```
Cmd> power(zeta1,g,alpha,n) # Power for CRD, the default
(1)     0.61812
```

For Randomized Complete Block design (RCB), $df_{error} = (g - 1)(n - 1)$.

```
Cmd> power2(n*zeta1,g-1,alpha,(g-1)*(n-1))
(1)     0.56874

Cmd> power(zeta1,g,alpha,n,design:"rcb")
(1)     0.56874
```


In $n_1 = n_2 = \ldots = n_g = n$ case, non-central F depends on $\zeta_1 = \sum\alpha_i^2/\sigma^2$.

Before you can choose a sample size n, you need somehow to come up with values for $\sum\alpha_i^2$ and σ^2.

You pick a value for σ^2 the same way you pick a value for MS_E when the goal is a C.I. width.

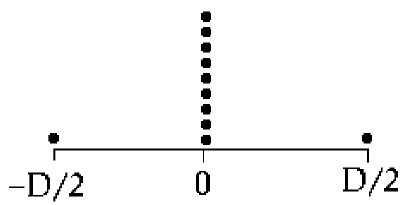
Picking $\sum\alpha_i^2$ often seems impossible.

It is simpler when you can come up with a difference D of two effects that is important to *discover* with high probability, that is reject H_0 with high probability.

You then can look at various cases involving at least one pair with $|\alpha_i - \alpha_j| = D$.

Pessimistic ζ_1

The most conservative or pessimistic is to plan for the *smallest* $\zeta_1 = \sum\alpha_i^2$ that can occur when at least α_i and α_j with $|\alpha_i - \alpha_j| = D$.



This guarantees at least the desired power with one or more $|\alpha_i - \alpha_j| \geq D$

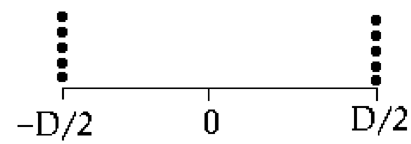
This worst case is when all $\alpha_i = 0$ except two which have values $\pm D/2$. In this case

$$\sum\alpha_i^2 = D^2/2 \text{ and } \zeta_1 = D^2/(2\sigma^2).$$

Optimistic ζ_1

Or you could be an optimist and plan for the *largest* possible $\sum\alpha_i^2$ with at least 1 pair with $|\alpha_i - \alpha_j| = D$. This case has the lowest power of alternatives with $|\alpha_i - \alpha_j| = D$ for some i, j.

With *even* g, this happens when half the α_i's are $+D/2$ and the other half are $-D/2$.



In this case

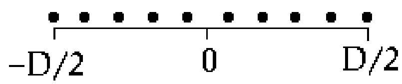
$$\sum\alpha_i^2 = g(D/2)^2 \text{ and } \zeta_1 = gD^2/(4\sigma^2).$$

With *odd* g, say $g = 2h + 1$, the best case is with h α_i's $= -(D/2)(-1 - 1/g)$ and h α_i's $= (D/2)*(1 - 1/g)$. With these α_i's

$$\sum\alpha_i^2 = (D/2)^2(g^2-1)/g$$

$$\zeta_1 = (g^2-1)D^2/(4g\sigma^2)$$

Somewhere in the middle is an alternative in which you assume the α_i's are **equally spaced** between $\min_i \alpha_i = -D/2$ and $\max_i \alpha_i = +D/2$.



In this case

$$\zeta_1 = g(g+1)D^2/(12(g-1)\sigma^2).$$

```
Cmd> sigmasq <- 1.26

Cmd> D <- 3

Cmd> power(D^2/(4*sigmasq),g,.01,n) # pessimistic
(1)     0.15842        Lowest  power

Cmd> power(g*(g+1)*D^2/(12*(g-1)*sigmasq),g,.01,n) # in middle
(1)     0.61812        Intermediate  power

Cmd> power(g*D^2/(4*sigmasq),g,.01,n) # optimistic
(1)     0.96745        Largest  power
```

The differing powers reflect the fact that the pessimistic case has smallest ζ_1, the intermediate case has second smallest ζ_1, and the optimistic case has the largest ζ_1.

You can use **power()** to find a sample size directly by **trial and error**.

Consider the intermediate case: Compute power for n = 2, 3, ..., 10

```
Cmd> N <- run(2,10) # range of sample sizes ≥ 2

Cmd> power(g*(g+1)*D^2/(12*(g-1)*sigmasq),g,.01,N)
(1)     0.10337     0.34759     0.61812     0.81296     0.92051
(6)     0.96987     0.98961     0.99669     0.99901
```

Power = .10337 goes with n = 2. The first power $\geq$ .9 is .92051 for n = 6 so you need n $\geq$ 6 for power $\geq$ .9; similarly for power > .95 you need n $\geq$ 7, etc.

Alternatively, you could use **power2()**

```
Cmd> power2(N*g*(g+1)*D^2/(12*(g-1)*sigmasq),g-1,.01,g*(N-1))
(1)     0.10337     0.34759     0.61812     0.81296     0.92051
(6)     0.96987     0.98961     0.99669     0.99901
```

Or you can use **samplesize()**. This is used almost like **power()** except the last argument is the power you want.

```
Cmd> samplesize(g*(g+1)*D^2/(12*(g-1)*sigmasq),g,.01,.9)
(1)           6

Cmd> samplesize(g*(g+1)*D^2/(12*(g-1)*sigmasq),g,.01,.95)
(1)           7
```

.9 and .95 are the desired powers. .01 is the significance level α of the test.

`samplesize()` is somewhat computationally intensive. By default it quits once it sees that $n > 256$ and reports 256 as the answer. It warns you when that happens.

Here's an example we want power = .9 but we have a tiny n = 1 non-centrality parameter ($\zeta_1 = .04$). This requires a large sample size.

```
Cmd> samplesize(.04,2,.05,.9) # g = 2, alpha = .05,power=.9
WARNING: samplesize() truncated at  256
(1)        256
```

To try harder, allowing answers ≥ 256, use keyword phrase **maxn:N** as an argument to set the truncation point to N. Here I tried again, allowing $N \leq 1000$.

```
Cmd> samplesize(.04,2,.05,.9,maxn:1000)
(1)        264
```

This is only slightly more than the default truncation point, and there's not much gain in power:

```
Cmd> power(.04,2,.05,vector(256,264))
(1)     0.8914     0.90038 8 more cases gains little
```


Another comparison of the three choices.

Suppose the threshold for "interesting" is D = 3 and you best guess is $\sigma^2 = 4$.

Then the lower bound (pessimistic) n = 1 non-centrality parameter is

$$\zeta_1 = D^2/(2\sigma^2) = 3^2/(2\times 4) = 1.125.$$

```
Cmd> samplesize(3^2/(2*4),6,.05,.95) # g = 6
(1)         19  Required sample size for each of 6 groups

Cmd> power(3^2/(2*4),6,.05,run(18,20))#Powers for n=18,19,20
(1)     0.94311     0.95545     0.96531
```

The intermediate sample size comes when

$$\zeta_1 = g(g+1)D^2/(12(g-1)\sigma^2)$$

```
Cmd> samplesize(6*(6+1)*3^2/(12*(6-1)*4),g,.05,.95)
(1)         14

Cmd> power(6*(6+1)*3^2/(12*(6-1)*4),g,.05,run(13,15))
(1)     0.94039     0.95768     0.97029 Powers for n=13,14,15
```

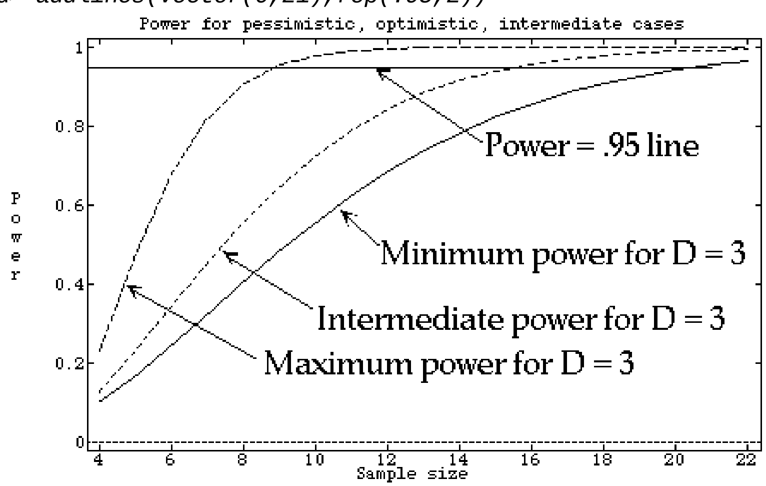
The optimistic sample size is

```
Cmd> samplesize(6*3^2/(4*4),g,.05,.95)
(1)          7

Cmd> power(6*3^2/(4*4),g,.05,run(6,8))#Powers for n=6,7,8
(1)     0.9078     0.95507     0.97919
```

Here I compute and the power for the 3 alternatives for n = 2, 3, ..., 20.

```
Cmd> D <- 3; sigmasq <- 4; g <- 6
Cmd> N <- run(2,20) # range of sample sizes
Cmd> pwr1 <- power(D^2/(2*sigmasq), g, .05, N) # pessimistic
Cmd> pwr2 <- power(g*D^2/(4*sigmasq), g, .05, N) #optimistic
Cmd> pwr3 <- power(g*(g+1)*D^2/(12*(g-1)*sigmasq), g, .05, N)
Cmd> lineplot(n,hconcat(pwr1,pwr2,pwr3),xlab:"Sample size",\
 ylab:"Power",ymin:0,\
 title:"Power for pessimistic, optimistic, intermediate cases")
Cmd> addlines(vector(0,21),rep(.95,2))
```



From this plot you can determine the required sample sizes from the points where the curves cross the power = .95 line.

Power for a contrast.

To test H_0: $\sum_i w_i \alpha_i = 0$, you normally use the t-statistic

$$t = \sum_i w_i \overline{y}_{i\bullet} / \hat{SE}[\sum_i w_i \overline{y}_{i\bullet}]$$
$$= \sum_i w_i \overline{y}_{i\bullet} / \sqrt{\{MS_E \times \sum_i w_i^2/n\}}$$

with d.f. = $df_{error} = g(n-1)$

Since $t_{df}^2 = F_{1,df}$, you can use **power2()** to compute power.

You *can't* use **power()** to compute power and you *can't* use **samplesize()** to find a sample size.

The n = 1 non-centrality parameter is

$$\zeta_1 = (\sum_i w_i \alpha_i)^2 / \{\sigma^2 \times \sum_i w_i^2\}$$

Suppose you want to compare the average effects of treates 1, 2 and 3 with the average effects of treatments 4, 5, an 6, and a difference of D = 1.5 is important to detect with high probability. You guess σ^2 = 1.26 and want power = .95

The contrast weights are

$$\{1/3, 1/3, 1/3, -1/3, -1/3, -1/3\}$$

```
Cmd> w <- vector(rep(1/3,3),rep(-1/3,3)); w
(1)     0.33333     0.33333     0.33333    -0.33333    -0.33333
(6)    -0.33333

Cmd> sigmasq <- 1.26 # Hoped for variance

Cmd> D <- 1.5

Cmd> zeta1 <- D^2/(sum(w^2)*sigmasq); zeta1
(1)      2.6786

Cmd> power2(5*zeta1,1,.01,g*(5-1)) # power for n = 5
(1)     0.79612

Cmd> N <- run(2,20) # range of sample sizes

Cmd> power2(N*zeta1,1,.01,g*(N-1)) # power for n = 2, ..., 20
 (1)      0.19702     0.44891     0.65329     0.79612     0.88649
 (6)       0.9396     0.96906     0.98466     0.99261     0.99653
(11)       0.9984     0.99928     0.99968     0.99986     0.99994
(16)      0.99997     0.99999           1           1
```

From this output, the first power $\geq$ .95 is .96906 corresponding to n = 8.

Non-central t

You should the non-central F distribution with numerator d.f. = 1 to find the power of a t-test of a contrast only when you plan a *two*-tail test. Although this is probably most common, sometimes your alternative to

H_0: $\sum_i w_i \alpha_i = 0$ is

- H_a: $\sum_i w_i \alpha_i > 0$ (reject for $t > t_\alpha$)

or

- H_a: $\sum_i w_i \alpha_i < 0$ (reject for $t < -t_\alpha$)

When H_0 is false, t has what is known as the **non-central t-distribution** on df_{error} degrees of freedom and non-centrality parameter $\delta = \sqrt{n}\sum w_i \alpha_i / (\sigma\sqrt{\{\sum w_i^2\}})$ so that $\delta^2 = \zeta$. $\delta = 0$ corresponds to ordinary (central) t.

When $\delta \neq 0$, t does not have 0 mean and is non-symmetric about its mean.

In MacAnova, you use **cumstu()** with δ as argument 3 to compute non-central t probabilities.

Find power of 1% two-tail test for the previous example:

```
Cmd> t_005 <- invstu(1 - .01/2, g*(5-1))#two tail 1% cutpoint
```

Now compute $P(|t_{noncentral}| \geq t_{.005}) =$

$$P(t_{noncentral} \leq -t_{.005}) + P(t_{noncentral} \geq +t_{.005})$$

```
Cmd> cumstu(-t_005,g*(5-1),sqrt(5*zeta1)) +\
    1 - cumstu(t_005,g*(5-1),sqrt(5*zeta1))
(1)     0.79612
```

This matches the power computed using **power2()**.

```
Cmd> power2(5*zeta1, 1, .01, g*(5-1)) # power for n = 5
(1)      0.79612
```

Find one-tail power when $\sum_i w_i \alpha_i = D = 1.5$

```
Cmd> t_01 <- invstu(1 - .01,g*5-1);t_01# 1-tail cutpoint
(1)        2.462

Cmd> delta1 <- D/sqrt(sum(w^2)*sigmasq)

Cmd> 1-cumstu(t_01,g*(5-1),sqrt(5)*delta1)
(1)      0.87552
```

As you should expect, the power of the one-tail test is larger than the power of the two-tail test.