Power

Brian Caffo, Jeff Leek, Roger Peng

May 19, 2016

Power

- Power is the probability of rejecting the null hypothesis when it is false
- Ergo, power (as its name would suggest) is a good thing; you want more power
- A type II error (a bad thing, as its name would suggest) is failing to reject the null hypothesis when it's false; the probability of a type II error is usually called β
- Note Power $= 1 - \beta$

Notes

- Consider our previous example involving RDI
- $H_0 : \mu = 30$ versus $H_a : \mu > 30$
- Then power is

$$P\left(\frac{\bar{X} - 30}{s/\sqrt{n}} > t_{1-\alpha,n-1} \ ; \ \mu = \mu_a\right)$$

- Note that this is a function that depends on the specific value of μ_a!
- Notice as μ_a approaches 30 the power approaches α

Calculating power for Gaussian data

- We reject if $\frac{\bar{X}-30}{\sigma/\sqrt{n}} > z_{1-\alpha}$
  - Equivalently if $\bar{X} > 30 + Z_{1-\alpha}\frac{\sigma}{\sqrt{n}}$
- Under $H_0 : \bar{X} \sim N(\mu_0, \sigma^2/n)$
- Under $H_a : \bar{X} \sim N(\mu_a, \sigma^2/n)$
- So we want

```
alpha = 0.05; mu0 = 30; mua = 32; sigma = 4; n = 16
z = qnorm(1 - alpha)
pnorm(mu0 + z * sigma / sqrt(n), mean = mua, sd = sigma / s
      lower.tail = FALSE)
```

```
## [1] 0.63876
```

Example continued

- $\mu_a = 32$, $\mu_0 = 30$, $n = 16$, $\sigma = 4$

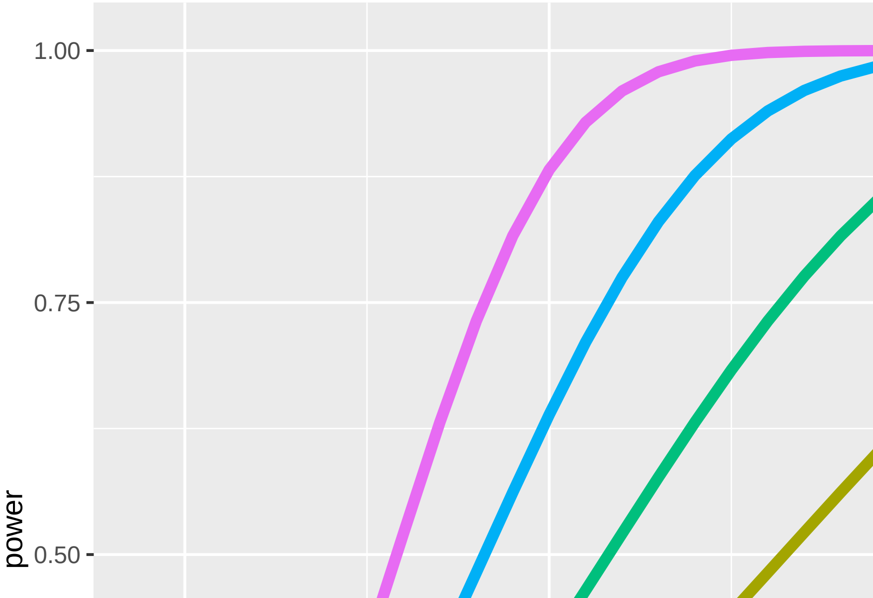
```
z = qnorm(1 - alpha)
pnorm(mu0 + z * sigma / sqrt(n), mean = mu0, sd = sigma / s
      lower.tail = FALSE)
```

```
## [1] 0.05
```

```
pnorm(mu0 + z * sigma / sqrt(n), mean = mua, sd = sigma / s
      lower.tail = FALSE)
```

```
## [1] 0.63876
```

Plotting the power curve



Graphical Depiction of Power

```
library(manipulate)
mu0 = 30
myplot <- function(sigma, mua, n, alpha){
    g = ggplot(data.frame(mu = c(27, 36)), aes(x = mu))
    g = g + stat_function(fun=dnorm, geom = "line",
                          args = list(mean = mu0, sd = sign
                          size = 2, col = "red")
    g = g + stat_function(fun=dnorm, geom = "line",
                          args = list(mean = mua, sd = sign
                          size = 2, col = "blue")
    xitc = mu0 + qnorm(1 - alpha) * sigma / sqrt(n)
    g = g + geom_vline(xintercept=xitc, size = 3)
    g
}
manipulate(
    myplot(sigma, mua, n, alpha),
    sigma = slider(1, 10, step = 1, initial = 4),
    mua = slider(30, 35, step = 1, initial = 32),
```

Question

- When testing $H_a : \mu > \mu_0$, notice if power is $1 - \beta$, then

$$1 - \beta = P\left(\bar{X} > \mu_0 + z_{1-\alpha}\frac{\sigma}{\sqrt{n}}; \mu = \mu_a\right)$$

- where $\bar{X} \sim N(\mu_a, \sigma^2/n)$
- Unknowns: μ_a, σ, n, β
- Knowns: μ_0, α
- Specify any 3 of the unknowns and you can solve for the remainder

Notes

- The calculation for $H_a : \mu < \mu_0$ is similar
- For $H_a : \mu \neq \mu_0$ calculate the one sided power using $\alpha / 2$ (this is only approximately right, it excludes the probability of getting a large TS in the opposite direction of the truth)
- Power goes up as α gets larger
- Power of a one sided test is greater than the power of the associated two sided test
- Power goes up as μ_1 gets further away from μ_0
- Power goes up as n goes up
- Power doesn't need μ_a, σ and n, instead only $\frac{\sqrt{n}(\mu_a - \mu_0)}{\sigma}$
- The quantity $\frac{\mu_a - \mu_0}{\sigma}$ is called the effect size, the difference in the means in standard deviation units.
- Being unit free, it has some hope of interpretability across settings

T-test power

- Consider calculating power for a Gossett's T test for our example
- The power is

$$P\left(\frac{\bar X - \mu_0}{S/\sqrt{n}} > t_{1-\alpha,n-1} ~;~ \mu = \mu_a\right)$$

- Calcuting this requires the non-central t distribution.
- `power.t.test` does this very well
- Omit one of the arguments and it solves for it

Example

```r
power.t.test(n = 16, delta = 2 / 4, sd=1, type = "one.sampl
```

```
## [1] 0.6040329
```

```r
power.t.test(n = 16, delta = 2, sd=4, type = "one.sample",
```

```
## [1] 0.6040329
```

```r
power.t.test(n = 16, delta = 100, sd=200, type = "one.sampl
```

```
## [1] 0.6040329
```

Example

```
power.t.test(power = .8, delta = 2 / 4, sd=1, type = "one.s
```

```
## [1] 26.13751
```

```
power.t.test(power = .8, delta = 2, sd=4, type = "one.sampl
```

```
## [1] 26.13751
```

```
power.t.test(power = .8, delta = 100, sd=200, type = "one.s
```

```
## [1] 26.13751
```