

Statistical Inference week4

Power

the probability of rejecting the null hypothesis when false 1 - beta
beta is 2nd error.

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

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

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

```
## Error in FUN("rs_createUUID"[[1L]], ...): no such symbol rs_createUUID
```

Usually you calculate for beta or n, to make a study, or make sure the study will be meaningful in a certain dataset.

t-test power just use power.t.test (non-central t distribution)

```
power.t.test(n = 16, delta = 2/4, sd = 1, type = "one.sample", alt = "one.sided")$power
```

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

```
power.t.test(n = 16, delta = 2, sd = 4, type = "one.sample", alt = "one.sided")$power
```

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

```
power.t.test(n = 16, delta = 100, sd = 200, type = "one.sample", alt = "one.sided")$power
```

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

```
power.t.test(power = 0.8, delta = 2/4, sd = 1, type = "one.sample", alt = "one.sided")$n
```

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

```
power.t.test(power = 0.8, delta = 2, sd = 4, type = "one.sample", alt = "one.sided")$n
```

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

```
power.t.test(power = 0.8, delta = 100, sd = 200, type = "one.sample", alt = "one.sided")$n
```

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

you need $n = 27$ to detect a 0.5 difference in the mean

Multiple comparisons

type1 : false positive

type2 : false negative

	$\beta = 0$	$\beta \neq 0$	HYPOTHESES
Claim $\beta = 0$	U	T	m-R
Claim $\beta \neq 0$	V	S	R
Claims	m_0	$m - m_0$	m

- false positive rate:

$$E \left[\frac{V}{m_0} \right]$$

- family wise error rate: $\Pr(V \geq 1)$ the probability of at least one false positive
- false discovery rate :

$$E \left[\frac{V}{R} \right]$$

controlling false positive

Bonferroni correction set

$$\alpha_{fwer} = \alpha/m$$

where m is the number of tests. This is easy to calculate, but may be very conservative.

Controlling false discovery rate

- most popular in genomics, etc.
- do m tests
- order p values from smallest to largest

$$P_{(1)}, \dots, P_{(m)}$$

- call any

$$P_{(i)} \leq \alpha \times \frac{i}{m}$$

significant

- this will allow for more false positives

Example with 10 p-values

Adjust the p values

- not p values anymore!
- easier than adjusting alpha
- eg. p values $P_{_1}, \dots, P_{_m}$

$$P_i^{fwer} = \max(m \times P_i, 1)$$

for each P-value.

- Then if you call all

$$P_i^{fwer} < \alpha$$

significant you will control the FWER.

```

set.seed(1010093)
pValues <- rep(NA, 1000)
for(i in 1:1000){
  y <- rnorm(20)
  x <- rnorm(20)
  pValues[i] <- summary(lm(y~x))$coeff[2,4]
}
# control false positive rate
sum(pValues<0.05)

```

```
## [1] 51
```

```
sum(p.adjust(pValues, method = "bonferroni")<0.05)
```

```
## [1] 0
```

```
sum(p.adjust(pValues, method = "BH")<0.05)
```

```
## [1] 0
```

```

set.seed(1010093)
pValues <- rep(NA,1000)
for(i in 1:1000){
  x <- rnorm(20)
  #first 500, beta = 0, last 500 beta = 2
  if(i<=500){y <- rnorm(20)}else{y <- rnorm(20,mean=2*x)}
  pValues[i] <- summary(lm(y~x))$coeff[2,4]
}
trueStatus <- rep(c("zero","not zero"), each = 500)
table(pValues < 0.05, trueStatus)

```

```

##      trueStatus
##      not zero zero
## FALSE      0  476
## TRUE      500   24

```

```
table(p.adjust(pValues, method = "bonferroni") < 0.05, trueStatus)
```

```

##      trueStatus
##      not zero zero
## FALSE      23  500
## TRUE      477    0

```

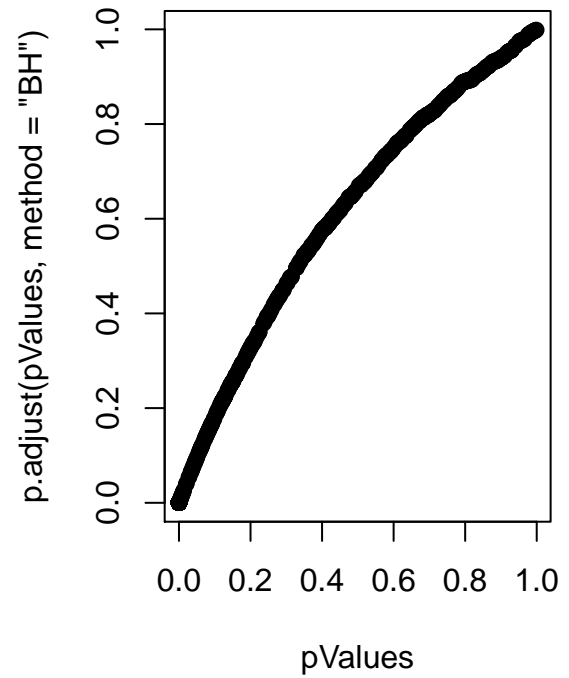
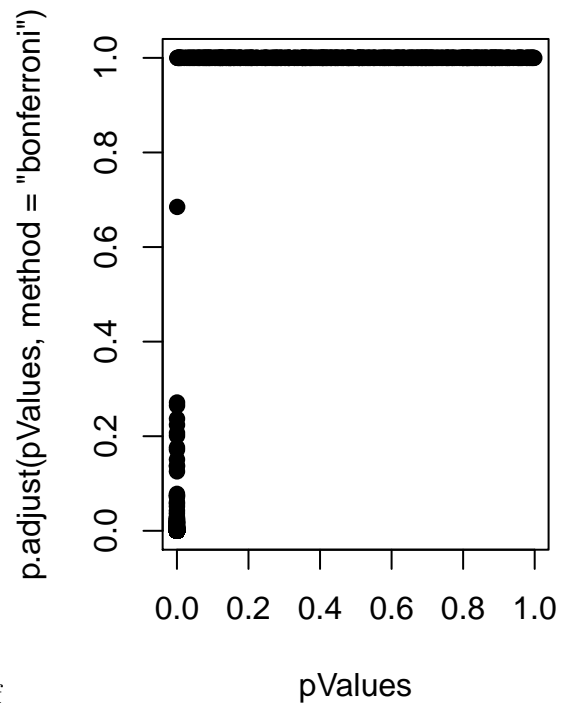
```
table(p.adjust(pValues, method = "BH")<0.05, trueStatus)
```

```

##      trueStatus
##      not zero zero
## FALSE      0  487
## TRUE      500   13

```

```
par(mfrow = c(1,2))
plot(pValues, p.adjust(pValues, method = "bonferroni"), pch = 19)
plot(pValues, p.adjust(pValues, method = "BH"), pch = 19)
```



p method does-1.pdf

maybe method = "BY" is good too