BIO5312: R Session 6
Statistical Hypothesis Testings

Yujin Chung

October 4th, 2016

Fall, 2016

Today's R session

- Using R for hypothesis testings
  - `t.test()`: one-sample t-test, paired t-test, two independent sample t-test
  - `var.test()`: two-sample variance test (F-test)
- `for()`: a loop statement
- `ifelse()`: a condition statement

Simulation: type-I error and power of a test

Consider a random sample of size $n = 50$. We want evaluate the type I error and the power of t-test for $H_0 : \mu = 0$ vs. $H_1 : \mu \neq 0$.

```
> n=50  # sample size
> x = rnorm(n) # a sample of size 50 from N(0,1)
> t.test(x) # one-sample t-test

One Sample t-test


data:  x
t = 0.067065, df = 49, p-value = 0.9468
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.2476242  0.2647226
sample estimates:
  mean of x
0.008549192
```

Simulation: type-I error of a test

```
t.test(x,
       alternative = c("two.sided", "less", "greater"),
       mu = 0,conf.level = 0.95, ...)
```

- `x`: a (non-empty) numeric vector of data values.
- `mu`: a number indicating the true value of the mean
- `alternative`: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
- `conf.level`: confidence level of the interval.

```
> t.test(x)$p.value  # access the p-value
[1] 0.9468029
```

Q) Generate 1000 sets and perform t-test. How many times the test reject the null?

for() loop statement

```
for(index_variable in start_value:end_value){
   <statements>
}
```

- `index_variable` changes from `start_value` to `end_value` increasing by 1
- For each value of `index_variable`, the statements are executed

```
> for(i in 1:5){
+   print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

`ifelse()` condition statement

```
ifelse(test, yes, no)
```

- `test`: an object which can be coerced to logical mode
- `yes`: return values for true elements of `test`.
- `no`: return values for false elements of `test`.

```
> ifelse(1==2, "apple","orange")
[1] "orange"
> ifelse(1!=2, "apple","orange")
[1] "apple"
```

Simulation: type-I error of a test

```
> # Generate 1000 sets and perform t-test
> # How many times the test reject the null
> count = 0
> for(i in 1:1000){
+   pval=t.test(rnorm(n))$p.value
+   count = count+ifelse(pval<0.05,1,0)
+ }
> count/1000
[1] 0.051
```

Simulation: power of a test

```
> mu=2 # true population mean
> x = rnorm(n,2) # random sample
> t.test(x) # t-test for H0:mu=0

One Sample t-test

data:  x
t = 13.441, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.599802 2.162279
sample estimates:
mean of x
  1.88104
```

Simulation: power of a test

```
> # Generate 1000 sets  and perform t-test
> # How many times the test reject the null
> count = 0
> for(i in 1:1000){
+   pval=t.test(rnorm(n,2))$p.value
+   count = count+ifelse(pval<0.05,1,0)
+ }
> count/1000
[1] 1
```

Paired t-test

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, conf.level = 0.95, ...)
```

- `x, y`: two samples
- `mu`: difference in means if you are performing a two sample test
- `paired`: a logical indicating whether you want a paired t-test.

Paired t-test: LEAD data

```
# lead data
>lead = read.table("LEAD.DAT.TXT",header=T)

> t.test(lead$fwt_r,lead$fwt_l,paired=T)

Paired t-test

data:  lead$fwt_r and lead$fwt_l
t = 9.8206, df = 123, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal
95 percent confidence interval:
 4.726245 7.112464
sample estimates:
mean of the differences
               5.919355
```

F-test for equal variances

```
var.test(x, y, ratio = 1,
         alternative = c("two.sided", "less", "greater"),
         conf.level = 0.95, ...)
```

- `x, y`: two independent samples
- **ratio**: the hypothesized ratio of the population variances of x and y.

F-test for equal variances

```
> fwt = lead$maxfwt
> fwt1 = fwt[lead$Group==1] # control
> fwt2 = fwt[lead$Group==2] # exposed
> var.test(fwt1,fwt2)

F test to compare two variances

data:  fwt1 and fwt2
F = 0.66384, num df = 77, denom df = 45, p-value = 0.1133
alternative hypothesis: true ratio of variances is not equal t
95 percent confidence interval:
 0.385337 1.102539
sample estimates:
ratio of variances
         0.6638375
```

t-test: two independent samples

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

- `x, y`: two samples
- `mu`: difference in means if you are performing a two sample test
- `paired`: a logical indicating whether you want a paired t-test.
- `var.equal`: Logical. `TRUE` if the equal variance is assumed; `FALSE` otherwise.

t-test: two independent samples (equal variance)

```
> t.test(fwt1,fwt2,var.equal = T)

Two Sample t-test

data:  fwt1 and fwt2
t = 0.64824, df = 122, p-value = 0.518
alternative hypothesis: true difference in means is not equal
95 percent confidence interval:
 -5.493961 10.844017
sample estimates:
mean of x mean of y
 62.43590  59.76087
```

t-test: two independent samples (unequal variance)

```
> t.test(fwt1,fwt2,var.equal = F)

Welch Two Sample t-test

data:  fwt1 and fwt2
t = 0.615, df = 79.969, p-value = 0.5403
alternative hypothesis: true difference in means is not equal
95 percent confidence interval:
 -5.981043 11.331098
sample estimates:
mean of x mean of y
 62.43590  59.76087
```