Honours Lecture Practical

Dr. Julian D. Karch

Honours Lecture Practical

Objectives: Rough idea of

- What R is and how it works
- What simulation studies are and how they work

Small Print / Assumptions Two sample t-test

1. Normality: Errors are normally distributed (robust unless n very small)
2. Homogeneity of Variances: Variances of errors are equal across groups $\sigma_1^2 = \sigma_2^2$. (robust unless group sizes unequal $n_{\mathrm{max}}/n_{\mathrm{min}} > 1.5$)
3. Independence: Errors are independent from each other (not robust)

We are skeptical!

Research questions:

1. t-test valid when the assumptions are met?
2. What happens when the assumptions are violated?

Plan

Create artificial data and check what happens

Recap definition 95% confidence interval estimator:

- Math: $\mathbb{P}(\theta^* \in \delta(D)) = 95\%$
- English: The true parameter is within 95% of the generated intervals

–> Generate many intervals (data sets) and check

Detailed Plan / Pseudocode

- Repeat often (Repeat)

1. Generate data (Data)
2. Calculate confidence interval (CI)
3. True value in confidence interval? (Check)

- Count how often true value is in confidence interval (Count)

Assignments

```r
x <- 5
print(x)
```

```
## [1] 5
```

Vectors

```r
x <- c(1,2,3,4,5,100)
print(x)
```

```
## [1]   1   2   3   4   5 100
```

Functions

```r
print(x)
```

```
## [1]   1   2   3   4   5 100
```

```r
maxX <- max(x)
print(maxX)
```

```
## [1] 100
```

Data: Generating Artifical Data

5 Coin tosses from a fair coin

```
coinTosses <- rbinom(5,1,0.5)
print(coinTosses)
```

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

Data: Generating Normal Data

```r
set.seed(1903)
group1 <- rnorm(n=10,mean=20,sd=1)
group2 <- rnorm(n=10,mean=3,sd=1)
print(round(group1,2))
```

```
##  [1] 20.44 19.48 19.20 19.30 20.16 20.77 20.59 21.18 19.38 20.36
```

```r
print(round(group2,2))
```

```
##  [1] 2.31 2.01 3.61 1.87 4.02 4.29 3.48 3.03 2.92 2.97
```

CI: Performing the *t*-test

```r
tResult <- t.test(group1,group2,var.equal=TRUE)
print(tResult)
```

```
##
##  Two Sample t-test
##
## data:  group1 and group2
## t = 49.963, df = 18, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  16.31674 17.74919
## sample estimates:
## mean of x mean of y
## 20.085412  3.052448
```

CI: Extracting the Confidence Interval

```r
confInter <- tResult$conf.int
print(confInter)
```

```
## [1] 16.31674 17.74919
## attr(,"conf.level")
## [1] 0.95
```

Check: Check a value in CI

```
toCheck <- 0
if (toCheck > confInter[1] && toCheck < confInter[2]){
  isIn <- TRUE
}else{
  isIn <- FALSE
}
print(isIn)
```

```
## [1] FALSE
```

Repeat: Repeating a Part

```r
for (i in 1:3){
  variable <- rnorm(n=1,mean=20,sd=1)
  print(sprintf('Iteration: %d, Variable %.2f',i,variable))
}
```

```
## [1] "Iteration: 1, Variable 19.65"
## [1] "Iteration: 2, Variable 20.91"
## [1] "Iteration: 3, Variable 19.86"
```

Count: Check + Repeat

```
tails <- 0
for (i in 1:5){
  coinToss <- rbinom(1,1,0.5)
  if (coinToss==1){
    tails <- tails + 1
  }
}
```

Pair Exercise

Repeat often (for i in 1:100)

1. Generate data according to the assumptions (rnorm)
2. Calculate confidence interval (t.test)
3. Check and Count: True value in confidence interval? (if . . . ; count <- count+1)

You probably need the slides:

Solution

```r
inConf <- 0 #initialize counter
for (i in 1:1000){ #repeat 1000 times
  #generate data according to normal
  group1 <- rnorm(n=10,mean=20,sd=1)
  group2 <- rnorm(n=10,mean=3,sd=1)
  #get confidence interval from t-test
  tResult <- t.test(group1, group2, var.equal=TRUE)
  confInter <- tResult$conf.int
  #check if true mean difference (17) is in CI
  if (17 > confInter[1] && 17 < confInter[2]){
    inConf <- inConf + 1
  } #end of if
} #end of for
print(inConf/1000)
```

```
## [1] 0.952
```

Advanced Function

```r
source('http://bit.ly/checkttest')
print(sprintf('All Assumptions; Coverage: %.2f',
              checkTTest(100)))
```

```
## [1] "All Assumptions; Coverage: 0.95"
```

```r
str(checkTTest)
```

```
## function (n, normal = TRUE, homogeneity = TRUE, equalGroups = TRUE,
##     independence = TRUE)
```

Investigate the remaining conditions using the checkTTest function

Normality 1

```r
print(sprintf('No Normality, big n; Coverage: %.2f',
              checkTTest(100,normal=FALSE)))
```

```
## [1] "No Normality, big n; Coverage: 0.93"
```

```r
print(sprintf('No Normality; small n; Coverage: %.2f',
              checkTTest(6,normal=FALSE)))
```

```
## [1] "No Normality; small n; Coverage: 0.97"
```

Solution Homogeneity

```
print(sprintf('No Homogeneity, equal groups; Coverage: %.2f',
              checkTTest(100,homogeneity=FALSE)))
```

```
## [1] "No Homogeneity, equal groups; Coverage: 0.95"
```

```
print(sprintf('No Homogeneity, unequal groups; Coverage: %.2f',
              checkTTest(100,homogeneity=FALSE,equalGroups = FALSE)))
```

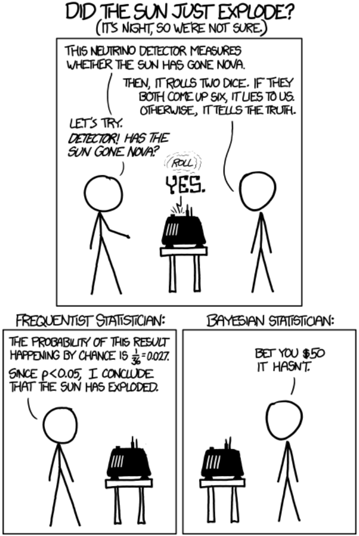
```
## [1] "No Homogeneity, unequal groups; Coverage: 0.43"
```

Solution Independence

```r
print(sprintf('No Independence; Coverage: %.2f',
              checkTTest(100,independence = FALSE)))
```

```
## [1] "No Independence; Coverage: 0.22"
```

The End


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY.
DETECTOR! HAS THE SUN GONE NOVA?
ROLL
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS 1/36 = 0.027.
SINCE p<0.05, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.