t-Test

André Meichtry

Contents

One-sample t-test

One-sample t-tests look at just one parameter of interest μ (in addition to σ) as data-generation mechanism. (see shinyApp).

Statistical model

- $Y_i = \mu + \epsilon_i; \quad \epsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$ (or with ϵ_i i.i.d. and "large" sample size), $\quad i = 1, ..., n$
- $Y_i \overset{i.i.d.}{\sim} N(\mu, \sigma^2)$ (equivalent)

This is the simplest example of $Y_i = \beta_1 + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$, the **only intercept model** with $\beta_1 = \mu$ (see IntroLinearModels).

Intro in sample size calculation

Let us first ask the **how many do I need**-question. There are two kinds of *sample size calculations*, the power approach and the precision approach.

Consider the problem $H_0 : \mu = \mu_0$ versus $H_1 : \mu \neq \mu_0$.

Power approach

A priori sample size calculation, see pages 77-78 AMT. Let β be the Type II error (The probability of **not** rejectig H_0 **if** a specified alternative H_1 is true. $1 - \beta$ is the probability of the complement, of rejectig H_0 **if** a specified alternative H_1 is true).

Question: What n we need to assure

- to reject $H_0 : \mu = \mu_0$
- with probability $1 - \beta$
- with false-positive rate α
- **if** the specified alternative $H_1 : \mu = \mu_1$ is true?

Assume $\mu_0 = 0$, thus $\delta = \mu_1 - \mu_0 = \mu_1$. One can show hat

$$n \geq \frac{(z_{1-\alpha/2} + z_{1-\beta})^2}{(\delta/\sigma)^2}$$

The approximation $n \approx 8/(\delta/\sigma)^2$ holds for $\alpha = 0.05$ and $\beta = 0.2$.

Assume we estimate or "guess" σ from another study with $\hat{\sigma} = 22$ and we want that H_0 is rejected with probability 0.8 **if** $\mu_1 = 13$ is true.

```
(qnorm(.975)+qnorm(.8))^2/((mu1-mu0)/sigmahat)^2
```

```
[1] 22.478
```

In R, you can use the *exact* version, `power.t.test()`. Read the help file `help(power.t.test)`. You have to specify δ, σ, power and the type of test:

```
power.t.test(delta=mu1-mu0,sd=sigmahat,power=0.8,type="one.sample")
```

```
     One-sample t test power calculation

              n = 24.469
          delta = 13
             sd = 22
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

Precision approach

In this approach, we do not need a Type II error. We have **not** to specify the alternative, which **most often makes more sense**, since there are **many (even infinite) options for the alternative** $\mu \neq \mu_0$.

We estimate the sample size by specifying the precision we want for the estimation, that is, we specify a priori the maximal width of the $100 \times (1-\alpha)\%$ CI.

An approximative $100 \times (1-\alpha)\%$ CI is given by $\bar{x} \pm \delta$, with $\delta = z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}$. Solving for n gives

$$n \geq \frac{z_{1-\alpha/2}^2}{(\delta/\sigma)^2}$$

.

Using again $\hat{\sigma} = 22$:

```
delta<-c(2,3,4,5,8)
alpha<-0.05
data.frame(delta=delta,n=qnorm(1-alpha/2)^2*(sigmahat/delta)^2)
```

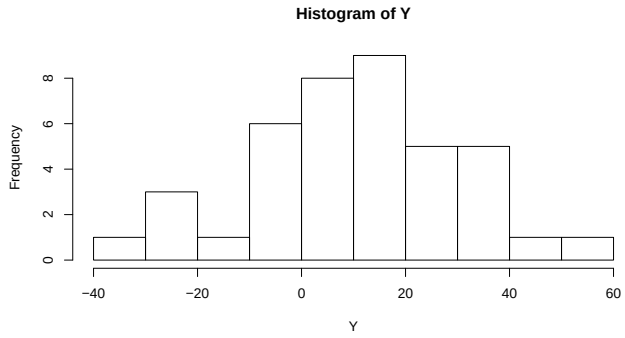
```
  delta       n
1     2 464.817
2     3 206.585
3     4 116.204
4     5  74.371
5     8  29.051
```

Simulation of example data

Let us simulate some data from population with **assumed known parameters** μ and σ^2. We simulate to understand the emergence of data. Of course, in reality, we do not know the data-generating process.

```
set.seed(55)
mu<-10
sigma<-20
n<-40
Y<-rnorm(n,mu,sigma)
```

```
hist(Y)
```




```
summary(Y)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-31.505  -0.214  11.220  11.309  25.104  57.107
```

Analysis

As one-sample t-Test

```
t.test(Y)
```

```
    One Sample t-test

data:  Y
t = 3.66, df = 39, p-value = 0.00074
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  5.0604 17.5574
sample estimates:
mean of x
   11.309
```

As linear model

This is equivalent with a linear model with only an intercept as parameter.

```
summary(mod<-lm(Y~1))
```

```
Call:
lm(formula = Y ~ 1)

Residuals:
   Min     1Q Median     3Q    Max
-42.81 -11.52  -0.09  13.80  45.80

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    11.31       3.09    3.66  0.00074

Residual standard error: 19.5 on 39 degrees of freedom
```

Of course, the results are the same. The t-test **is** a simple linear model.

Estimation of σ "by hand" (this quantity is given in the output above):

```
sqrt(sum(mod$residuals^2)/(n-1))
```

```
[1] 19.538
```

Confidence interval for μ:

```
confint(mod)
```

```
             2.5 % 97.5 %
(Intercept) 5.0604 17.557
```

Small samples without normality

In the example above, the assumptions are met since we simulated data from a normal distribution. In the absence of normality, this assumption of the t-test is not met. We can then perform a non-parametric test.

- The `wilcox.test` is the non-parametric alternative to the t-test.

- With the actual data, results are similar since we have simulated from a normal distribution.

```
wilcox.test(Y,conf.int=TRUE)
```

```
    Wilcoxon signed rank test

data:  Y
V = 655, p-value = 0.00068
alternative hypothesis: true location is not equal to 0
95 percent confidence interval:
  5.141 17.842
sample estimates:
(pseudo)median
        11.487
```

Two-sample t-Test

(see shinyApp).

Statistical model

1. Means parameterization:
   - $Y_{ij} = \mu_i + \epsilon_{ij}, \quad i = 1, 2; \quad j = 1, ..., n_i; \quad \epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2)$
   - $Y_{ij} \overset{i.i.d.}{\sim} N(\mu_i, \sigma^2)$ (equivalent)
2. Effects parameterization:
   - $Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$
   - $Y_{ij} \overset{i.i.d.}{\sim} N(\mu + \alpha_i, \sigma^2)$ (equivalent)
3. In R regression language:
   - $Y_i = \beta_0 + \beta_1 I_{group_i=2} + \epsilon_i$, with
     - $i = 1, ..., n$
     - $\beta_0 = \mu_1, \quad \beta_1 = \mu_2 - \mu_1$
     - $I_{group_i=2} = 1$, $if\ group_i = 2; \quad I_{group_i=2} = 0$, $if\ group_i = 1$ (="Dummy" variable)
   - (see IntroLinearModels).

Simulation of example data

We simulate to understand the emergence of data. Of course, in reality, we do not know the data-generating process.

```
n<-30 ##n per group
muInt<-21 ##True mean Int
muCont<-20 ##True mean Cont
Delta<-muInt-muCont ##True mean difference
Delta
```

```
[1] 1
```

```
sigma<-2 ##True standard deviation
set.seed(10)
dataInt <- rnorm(n=n,mean=muInt,sd=sigma) ##Sample from Int
dataCont <- rnorm(n=n,mean=muCont,sd=sigma) ##Sample from Cont
group<-gl(n=2,k=n,labels=c("Cont","Int"))##grouping variable
```

```r
response<-c(dataCont,dataInt)##the outcome
mydata <- data.frame(response=response,group=group)##the data.frame
str(mydata)
```

```
'data.frame':   60 obs. of  2 variables:
 $ response: num  16.3 19.8 21.9 20.4 17.2 ...
 $ group   : Factor w/ 2 levels "Cont","Int": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
head(mydata)
```

```
  response group
1   16.293  Cont
2   19.844  Cont
3   21.937  Cont
4   20.370  Cont
5   17.240  Cont
6   17.129  Cont
```

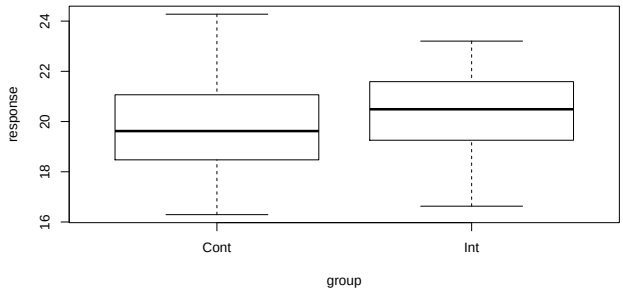
```r
summary(mydata)
```

```
    response     group
 Min.   :16.3   Cont:30
 1st Qu.:18.7   Int :30
 Median :20.3
 Mean   :20.0
 3rd Qu.:21.2
 Max.   :24.3
```

```r
by(mydata,mydata$group,summary)
```

```
mydata$group: Cont
    response     group
 Min.   :16.3   Cont:30
 1st Qu.:18.5   Int : 0
 Median :19.6
 Mean   :19.8
 3rd Qu.:21.1
 Max.   :24.3
------------------------------------------------------------------------------------------
mydata$group: Int
    response     group
 Min.   :16.6   Cont: 0
 1st Qu.:19.3   Int :30
 Median :20.5
 Mean   :20.3
 3rd Qu.:21.5
 Max.   :23.2
```

```r
boxplot(response~group,mydata)
```



Analysis

Hypothesis

H_0: $\mu_I - \mu_C = 0$ vs. $H_1 : \mu_I - \mu_C \neq 0$

```
test <- t.test(x=dataInt,y=dataCont,var.equal=TRUE)
test
```

```
    Two Sample t-test

data:  dataInt and dataCont
t = 1.14, df = 58, p-value = 0.26
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.40249  1.47525
sample estimates:
mean of x mean of y
   20.311    19.774
```

```
test$estimate ##Estimate of muInt and muCont
```

Extract information from object test

```
mean of x mean of y
   20.311    19.774
```

```
test$statistic ##t-statistic
```

```
     t
1.1436
```

```
test$stderr ##standard error
```

```
[1] 0.46903
```

```
((test$estimate[1]-test$estimate[2])-0)/test$stderr ##t-statistic "by hand"
```

```
mean of x
   1.1436
```

```
test$p.value
```

```
[1] 0.25749
```

Other Hypotheses

- H_0: $\mu_I - \mu_C \leq 0$ vs. $H_1 : \mu_I - \mu_C > 0$

```
test2 <- t.test(x=dataInt,y=dataCont,alternative="greater",var.equal=TRUE)
test2
```

```
    Two Sample t-test

data:  dataInt and dataCont
t = 1.14, df = 58, p-value = 0.13
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -0.24763      Inf
sample estimates:
mean of x mean of y
   20.311    19.774
```

- H_0: $\mu_I - \mu_C \leq -1$ vs. $H_1 : \mu_I - \mu_C > -1$

```
test3 <- t.test(x=dataInt,y=dataCont,mu=-1,alternative="greater",var.equal=TRUE)
test3
```

```
    Two Sample t-test

data:  dataInt and dataCont
t = 3.28, df = 58, p-value = 0.00089
alternative hypothesis: true difference in means is greater than -1
95 percent confidence interval:
 -0.24763      Inf
sample estimates:
mean of x mean of y
   20.311    19.774
```

Formula version

Back to the problem H_0: $\mu_I - \mu_C = 0$ vs. $H_1 : \mu_I - \mu_C \neq 0$

```
test<- t.test(response~group,data=mydata,var.equal=TRUE)
test
```

```
    Two Sample t-test

data:  response by group
t = -1.14, df = 58, p-value = 0.26
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.47525  0.40249
sample estimates:
mean in group Cont  mean in group Int
            19.774             20.311
```

As linear model

```r
mod<-lm(response~group,data=mydata)
summary(mod)
```

```
Call:
lm(formula = response ~ group, data = mydata)

Residuals:
   Min     1Q Median     3Q    Max
-3.681 -1.152  0.119  1.282  4.501

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   19.774      0.332   59.62   <2e-16
groupInt       0.536      0.469    1.14     0.26

Residual standard error: 1.82 on 58 degrees of freedom
Multiple R-squared:  0.0221,    Adjusted R-squared:  0.00519
F-statistic: 1.31 on 1 and 58 DF,  p-value: 0.257
```

```r
confint(mod)
```

```
               2.5 %  97.5 %
(Intercept) 19.11038 20.4381
groupInt    -0.40249  1.4753
```

What is the F-statistic

In the output of the model summary, there is an F-value. We will introduce the F-statistic in the context of ANOVA.

The aim is to test the null that we can omit `group` from the model. In this simple case, that is the same as the t-test for `group` ($\mu_I = \mu_C$). Very generally, we use F-tests as tests in the context of **model comparison**.

```r
mod2<-lm(response~1,data=mydata) ## Intercept-only model
anova(mod,mod2) ## tests the null: "Effect of group is absent"
```

```
Analysis of Variance Table

Model 1: response ~ group
Model 2: response ~ 1
  Res.Df RSS Df Sum of Sq    F Pr(>F)
1     58 191
2     59 196 -1     -4.32 1.31   0.26
```

Introduction in residual analysis

The **assumptions of linear models** are (see IntroLinearModels)

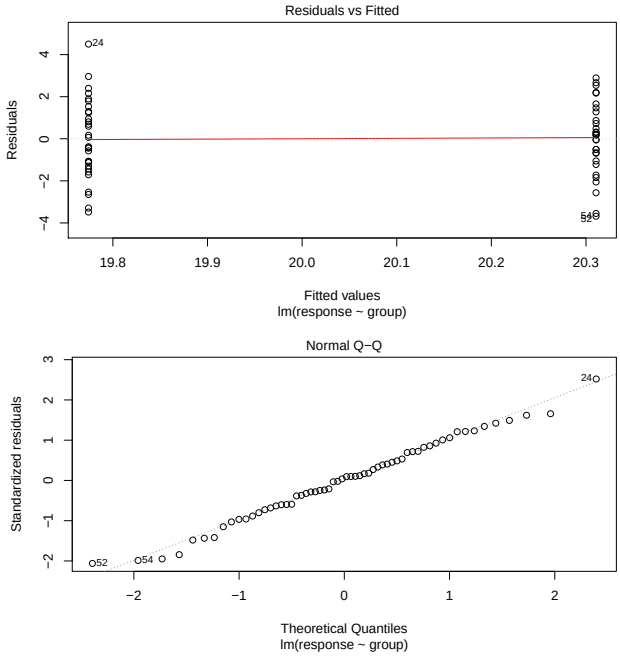
- ϵ_i are independent and identically distributed, i.i.d.
- $E(\epsilon_i) = 0$ for all i.
- $Var(\epsilon_i) = \sigma^2$ constant for all i.

For the t-test, we have to assume **in addition**

- ϵ_i i.i.d. $\sim N(0, \sigma^2)$

In R, we can just call the `plot` function for the model object to check the assumptions.

```
plot(mod,which=c(1,2))
```



Small samples without normality

In the example above, the assumptions are met since we simulated data from a normal distribution. Again, in the absence of normality, this assumption of the t-test is not met. We can then perform a non-parametric test.

- The Mann-Whitney Test (`wilcox.test` in R) is the non-parametric alternative to the two-sample t-test.
- With the actual data, results are similar since we have simulated from a normal distribution.

```
wilcox.test(dataInt,dataCont,conf.int=TRUE)
```

```
    Wilcoxon rank sum test

data:  dataInt and dataCont
W = 530, p-value = 0.24
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -0.38743  1.56444
sample estimates:
difference in location
               0.57931
```

Equivalence testing

Philosophical background

Very often, it makes not much sense to test nulls such as $H_0 : \mu = 0$ versus $H_1 : \mu \neq 0$. Assume a theory predicts a range for μ, i.e. that μ lies in a region $[-\epsilon, +\epsilon]$.

You then have the following test situation:

$H_0 : \mu \leq -\epsilon$ OR $\mu \geq +\epsilon$ versus $H_1 : -\epsilon < \mu < \epsilon$, meaning that

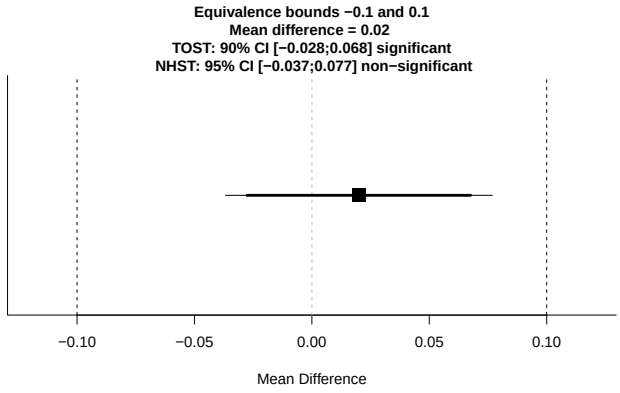
- Null: The true parameter is outside a tolerance region, "irrelevance".
- Alternative: The true parameter is inside a tolerance region, "relevance".

  Rejecting H_0 now really means rejecting irrelevance (defined by the margins ϵ).

Implementation

This analysis can be performed with TOST (Two One-sided t-Tests), this is implemented in the package `TOSTER`. For the one-sample situation, we have the function `TOSTone.raw()`.

```
library(TOSTER)
## Test observed mean of 0.52 and standard deviation of 0.5 in sample of 300 participants
## against 0.5 given equivalence bounds in raw units of -0.1 and 0.1, with an alpha = 0.05.
TOSTone.raw(m=0.52,mu=0.5,sd=0.5,n=300,low_eqbound=-0.1, high_eqbound=0.1, alpha=0.05)
```




```
TOST results:
t-value lower bound: 4.16   p-value lower bound: 0.00002
t-value upper bound: -2.77  p-value upper bound: 0.003
degrees of freedom : 299

Equivalence bounds (raw scores):
low eqbound: -0.1
high eqbound: 0.1

TOST confidence interval:
lower bound 90% CI: -0.028
upper bound 90% CI:  0.068

NHST confidence interval:
lower bound 95% CI: -0.037
upper bound 95% CI:  0.077

Equivalence Test Result:
The equivalence test was significant, t(299) = -2.771, p = 0.00297, given equivalence bounds of -0.100 a
Null Hypothesis Test Result:
The null hypothesis test was non-significant, t(299) = 0.693, p = 0.489, given an alpha of 0.05.
Based on the equivalence test and the null-hypothesis test combined, we can conclude that the observed e
```

Package TOSTER functions

If you need such analysis in the future, look at `help(package="TOSTER")` for other functions in the package, such as functions for sample size estimation.