

Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Seminar 2

1 Synopsis.

What this seminar is about. In the previous seminar you saw how to estimate things using the Monte Carlo method. We will learn to use this technique again in conjunction with the theory to assess the effect of violation of the normality and equal variances assumptions on some common hypothesis tests.

What you need. You need to be fully acquainted with the content of previous seminars and practice assignment. You will also need access to a computer equipped with R.

2 Background

2.1 Classical normal theory tests

We have seen by now several classic tests (on the mean of a single population, on the difference of means of two populations, on the difference of variances. . .) In all cases, the normality assumption on the “mother” population (or the mother populations, where there are two) enables us to derive the distribution of the test statistic under the null hypothesis, and hence the significance level associated to a critical region. For instance, if we deal with a normal population $N(m_0, \sigma^2)$, we know that the test statistic

$$Z = \frac{\bar{X} - m_0}{s} \sqrt{n - 1} \quad (1)$$

is distributed as Student’s t with $(n - 1)$ degrees of freedom.

2.2 Significance level and power

If, for instance, the degrees of freedom are $n - 1 = 9$, we know that when indeed the distribution is normal and the mean is truly m_0 , the probability that $Z > 2.82$ is just 0.01. Thus, if we reject the null hypothesis $H_0 : m = m_0$ whenever $|Z| > 2.82$, the probability of unduly rejecting H_0 is 0.01. We call this probability the *significance level* α or probability of type I error.

Other than the significance level we are interested in the probability of type II error or β , the probability of not rejecting H_0 when in fact we should, because the alternative is true. Equivalently, we are interested in the *power* or $1 - \beta$, probability of rejecting H_0 when it is in fact false. Clearly, this probability of rejection will depend on the alternative: $H_0 : m = 0$ may be difficult to reject if the true distribution has $m = 0.01$, but far more easy in $m = 12$, for instance. Thus, the power has to be calculated for different values of m (or whichever other parameter is involved in our testing).

Now, if we compute the probability of rejection for all values of m we will have the *power curve*: for $m = m_0$, this probability of rejection is the significance level, for $m \neq m_0$ it will be the power. A single curve contains all the information about the performance of a test.

2.3 Significance level and power: nominal and real

Now, distributions are theoretical models which will rarely be more than approximations to the prevailing real distribution. This means that nominal and real significance levels will ordinarily differ. By *nominal significance level* we mean the theoretical significance level defined in the previous section, which can be calculated under prescribed assumptions (such as normality of the population). The *real significance level* is the true probability of rejecting H_0 in the situation at hand, were the distribution may not be the one we are assuming.

Ideally, we would like to use tests which are not overly sensitive to distributional assumptions. In other words, we would like that even if the mother distribution is not really normal, the real significance level be not far off the nominal significance.

We will learn in the sequel how to compute approximate real significance levels using the Monte Carlo method introduced in Seminar 1. This will provide an easy method for checking sensitivity of our test procedures to violation of the assumptions.

Even more interesting, we will learn how to calculate the approximate real power for any alternative, under given conditions. This is important, because this is usually the only practical way to approximate the power: even under precise distributional assumptions, power is in general not easy to compute.

3 Monte Carlo approximations

3.1 Of the real significance level.

Consider the test statistic Z in (1). We have seen that if we are indeed taking a sample of size $n = 10$ from a normal population with mean $m = m_0$, then $P(|Z| > 2.82) = 0.01$.

Lets check that in this case Z takes values in the critical region $[2.82, \infty)$ with probability as prescribed by the theory. To this effect, we will generate a large number of samples from the distribution assumed by H_0 , compute the value of Z for each, and count how many fall in $[2.82, \infty)$.

Since we will need to compute Z many times for different samples and values of m_0 we define a function¹ which takes a sample and value of m_0 and returns the value of Z :

```
> t.stat <- function(sample, m0) {
  n     <- length(sample)
  xbar  <- mean(sample)
  s2    <- sum( (sample-xbar)^2 ) / n
```

¹We do this for didactical purposes. You have pre-defined functions in R for every conceivable purpose. In the case at hand, rather than defining your own function, you can use the R pre-defined function `t.test`. We will see how make use of it for Activity 2 in Section 4 below.

```

s    <- sqrt(s2)
Z    <- (xbar-m0) * sqrt(n-1) / s
return(Z)
}

```

Observe now the following code. We will start defining everything that can change at the top, then inside a loop generate random normal samples, compute values of the test statistic and check how many fall inside the critical region. We will set the true mean equal to the hypothetical mean and the critical region equal to the value which leaves a right tail of size α .

```

> N    <- 10000          # Number of simulations
> n    <- 10             # Sample size
> m0   <- 0              # Hypothetical mean
> m    <- 0              # True mean
> alfa <- 0.01          # Desired alpha
> cr   <- qt(1-alfa,df=n-1) # Start of critical region (one sided)
> cr

```

```
[1] 2.821438
```

Now we are all set; can write a loop to do repetitive work:

```

> Hits <- 0
> for (i in 1:N) {
  sample <- rnorm(n,mean=m)      # Generate random sample
  if (t.stat(sample, m0) > cr)  # Check whether Z in critical region
    Hits <- Hits + 1            # If so, increase count
}
> prob <- Hits / N
> prob

```

```
[1] 0.0111
```

In `prob` we have computed the fraction of times the test statistic falls in the (nominal) size $\alpha = 0.01$ critical region. The value in `prob` is a Monte Carlo estimate of the *real significance level*. Since the distributional assumptions are fulfilled, we should expect `prob` to be quite close to 0.01, and this is indeed the case.

What would happen if the true distribution, rather than normal, were $U(-2, 2)$? Let's try:

```

> Hits <- 0
> for (i in 1:N) {
  sample <- runif(n,min=-2, max=2) # Generate random sample
  if (t.stat(sample, m0) > cr)    # Check whether Z in critical region
    Hits <- Hits + 1              # If so, increase count
}
> prob <- Hits / N
> prob

```

```
[1] 0.0142
```

The value of `prob` is somewhat larger than 1%, but not much. One of the nice things of the one-sample t -test is that it is fairly insensitive to lack of normality: it can be used in rather general situations, and the real significance level of the test will not be far off the theoretical or nominal significance level.

3.2 Of the power

Not only can we approximate the type I (or α) error, but also the power of the test. Sometimes the power can be computed analytically², but in real situations this is cumbersome. All we have to do in the code above is replace `m` by a value different from the (null hypothesis) `m0` and count hits in the critical region. For instance, how often will the t -test detect that a $N(m = 0.5, \sigma = 1)$ is *not* a $N(m = 0, \sigma = 1)$? Let's try (all omitted values are the same as in the previous simulation):

```
> m      <- 0.5                # True mean
> Hits <- 0
> for (i in 1:N) {
  sample <- rnorm(n,mean=m)    # Generate random sample
  if (t.stat(sample, m0) > cr) # Check whether Z in critical region
    Hits <- Hits + 1          # If so, increase count
}
> prob <- Hits / N
> prob

[1] 0.1634
```

So with only $n = 10$ observations, we can hardly tell apart a $N(m = 0, \sigma = 1)$ and a $N(m = 0.5, \sigma = 1)$: only about 16.34% will the difference be detected at the $\alpha = 0.01$ significance level. As you may guess, increasing n (or making m “more different” from m_0) will increase the power. We will use $n = 50$ in the next simulations.

3.3 Of the power curve

All we need to compute an approximation of the power curve is a second loop over values of `m`.

```
> n      <- 50                # Sample size
> m.min <- 0.0 ; m.max <- 1.0 # Min and max
> vals  <- seq(from=m.min,    # Values of m to try; a vector.
               to=m.max,
               by=0.05)
> prob  <- 0 * vals          # Vector of the same dimension, to
>                                             # keep values of OC curve
>
> for (k in 1:length(vals)) { # Loop over means
  Hits <- 0 ; m <- vals[k]
  for (i in 1:N) {           # Loop over samples
    sample <- rnorm(n,mean=m) # Generate random sample
    if (t.stat(sample, m0) > cr) # Check whether Z in critical region
```

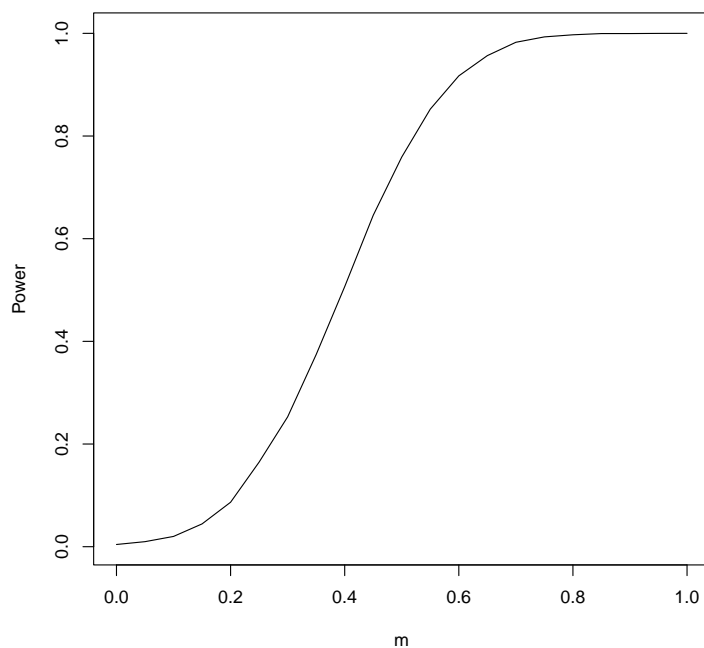
²The noncentral distributions that were briefly mentioned in class play a key role in power computations.

```

        Hits <- Hits + 1          # If so, increase count
    }
    prob[k] <- Hits / N
  }
> plot(vals, prob,
      type="l",
      xlab="m",
      ylab="Power")

```

Figure 1: Power curve for the test of $H_0 : m = 0$ versus $H_a : m > 0$ for different values of m and normal populations. The height of the curve above $m = 0$ is an estimate of the real significance level (here coincident with the nominal significance level, as the samples are normal).



The code above produces the graph in Figure 1. If the true distribution, instead of normal, were centered at $m_0 = 0$ but Cauchy with location parameter m ,

$$f_X(x) = \frac{1}{\pi} \frac{1}{1 + (x - m)^2}$$

using the t -test for testing $H_0 : m = 0$ versus $H_a : m > 0$ would give misleading results. To estimate the power curve we only need to perform the same simulation aboved with the line

```
> sample <- rnorm(n,mean=m)          # Generate random sample
```

replaced by:

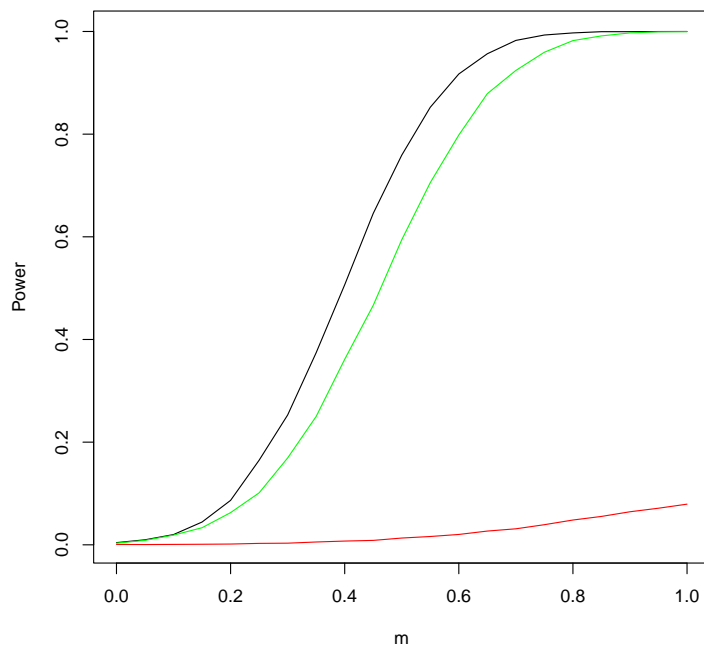
```
> sample <- rcauchy(n, location=m)    # Generate random sample
```

Similarly, to simulate from uniforms $U(m - 2, m + 2)$ we could replace the same line by :

```
> sample <- runif(n, min=m-2, max=m+2) # Generate random sample
```

Figure 2 (code not shown) displays a comparison of the power in the normal, $U(m - 2, m + 2)$ and the Cauchy case. It is apparent that there is a modest loss in power for the uniform distribution. However, the lost of power is disastrous if the distribution is Cauchy. The t -test, although fairly robust against non-normality, breaks in extreme cases of fat-tailed distributions.

Figure 2: Power curve for the rest of $H_0 : m = 0$ versus $H_a : m > 0$ for different values of mean of a normal distribution (black), uniform $U(m - 2, m + 2)$ (green) and location m of a Cauchy distribution (red). The height of the curve above $m = 0$ is an estimate of the real significance level. Notice the much reduced power with the Cauchy.



4 Practical details

4.1 Use of `t.test`, one sample problems.

For the simulations above we have coded `t.stat` and set the critical region ourselves. We can use the standard R function `t.test`. The full description can be obtained with

```
> help(t.test)
```

We give some details here, however. You can invoke `t.test` with one or two arguments, depending on whether you want a one population or a two population test. These arguments are respectively `x` and `y`.

For a one-sample t -test you have to provide only x . Compare here the results of our function `t.stat` and `t.test` in a simple example in which $H_0 : m_0 = 0$ is tested against a two sided alternative. Using our `t.stat`:

```
> sample1 <- rnorm(50,mean=0, sd=1)
> Z <- t.stat(sample1, m0=0)
> Z
```

```
[1] -1.335097
```

```
> 2 * (1 - pt(abs(Z), df=49))
```

```
[1] 0.1880144
```

Using the standard `t.test` we obtain:

```
> t.test(x=sample1, mu=0, alternative="two.sided")
```

```
One Sample t-test
```

```
data: sample1
t = -1.3351, df = 49, p-value = 0.188
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.4571451  0.0921867
sample estimates:
mean of x
-0.1824792
```

The results reported are the same, but `t.test` gives more detailed information, including degrees of freedom, alternative tested, and p -value.

One advantage of using `t.test` which makes our life easier is that we need not explicitly set the critical region: the function does it automatically, from the value of the argument `alternative`. In order to check whether the test statistic falls in the critical region of size α we only have to check whether the returned p -value is smaller than α .

In the case above, if we wanted to check whether the result is significant at the $\alpha = 0.05$ level (and increase `Hits` if it is), we would code:

```
> result <- t.test(x=sample1, mu=0, alternative="two.sided")
> if (result$p.value < 0.05) {
  Hits <- Hits + 1
}
```

(All the information computed by `t.test` can be accessed in a like manner; if you want to see the structure of the object returned by `t.test` above, you can type: `str(result)`.)

4.2 Use of `t.test`, two sample problems.

Function `t.test` and also be used for a two sample test. For instance, if you want to test $H_0 : m_x = m_y$ versus $H_a : m_x < m_y$ you can invoke:

```
> sample1 <- rnorm(50,mean=0, sd=1) # Sample first population
> sample2 <- rnorm(50,mean=0.3,sd=1) # Sample second population
> t.test(x=sample1, y=sample2, mu=0,
        var.equal=TRUE,
        alternative="less")
```

Two Sample t-test

```
data: sample1 and sample2
t = -2.9213, df = 98, p-value = 0.002163
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -0.2386522
sample estimates:
 mean of x mean of y
0.07667092 0.62964592
```

The output contains a wealth of information: the value of the t statistic for a two sample test, the estimated means, and the p -value under the assumption of equal variances³.

As before, checking whether the test statistic is inside the critical region is equivalent to checking whether the p -value is smaller than the chosen significance level.

4.3 Reporting results of long simulations

Sometimes you want to report results of computations for various inputs. You can do it in several ways. Suppose for instance that you want to compute x^y for various values of x and y . If you know beforehand what these values are, you can define a matrix to hold the results and print that matrix at the very end. For instance, the following code prints a table of numbers along with their square and cubic roots:

```
> x      <- 1:4
> y      <- c(1, 0.5, 1/3)
> results <- matrix(0, length(x), length(y))
> for (i in 1:length(x)) {
  for (j in 1:length(y)) {
    results[i,j] <- x[i]^y[j]
  }
}
> print(results)
```

³Note that if equal variances are assumed, like in the standard t -test studied in class, we need to pass the argument `var.equal=TRUE`; otherwise, a slightly different kind of test is performed. Testing equality of means with unequal variances is a messy problem: you can google for “Behrens-Fisher” to get an idea of the solutions proposed.

	[,1]	[,2]	[,3]
[1,]	1	1.000000	1.000000
[2,]	2	1.414214	1.259921
[3,]	3	1.732051	1.442250
[4,]	4	2.000000	1.587401

5 Activity 2

You have seen that, barring extreme non-normality (such as in the case of the Cauchy), the power of the t -test is not much degraded if normality fails. This is the case also in the case of two-sample t -tests, which are not overly sensitive to lack of normality. They are, however, much more affected by unequal population variances.

Now it is your turn. You have to simulate samples from two different populations, one $N(0, 1)$ and the other $N(m = 0.5, \sigma^2)$ and check what happens as σ^2 is farther from 1. You will see that the power of the test of $H_0 : m_X = m_Y$ versus the (true) $H_a : m_X < m_Y$ degrades. You should provide estimates of the power for the given alternative and six different values of σ^2 : 1, 1.2, 1.4, 1.6, 1.8, 2.0.

You should observe decreasing power as σ^2 separates more from 1. Report the estimated powers for all six values of σ^2 .