Session 4: Statistical hypothesis testing

Aaron Ponti and Dimosthenis Gaidatzis

In this session we will only scratch the surface of the theory of hypothesis testing. Instead, we will play with a few selected examples to get familiar with the approach to statistical testing in MATLAB.

1 What is a p-value?

From Wikipedia:

> In statistical hypothesis testing, the p-value is the probability of obtaining a result at least as extreme as the one that was actually observed, given that the null hypothesis is true.

This requires one additional definition (again from Wikipedia):

> In statistics, a null hypothesis (H_0) is a plausible hypothesis (scenario) which may explain a given set of data. A null hypothesis is tested to determine whether the data provide sufficient reason to pursue some alternative hypothesis (H_1). When used, the null hypothesis is presumed sufficient to explain the data unless statistical evidence, in the form of a hypothesis test, indicates otherwise.

As an example, suppose someone tells you that currently the average price of a liter of regular unleaded gas in Canton Basel Stadt is CHF 1.94. How could you determine the truth of the statement?

You could try to find prices at every gas station in Basel, Riehen and Bettingen. That approach would be definitive, but it could be time-consuming, costly, or even impossible.

A simpler approach would be to find prices at a small number of randomly selected gas stations, and then compute the sample average. Sample averages differ from one another due to chance variability in the selection process. Suppose your sample average comes out to be CHF 1.97. Is the CHF 0.03 difference an artifact of random sampling or significant evidence that the average price of a liter of gas is in fact greater than CHF 1.94?

Hypothesis testing is a statistical method for making such decisions.

2 Hypothesis test terminology

We formulate the statistical test for the study in section 1 as follows:

- *Null hypothesis* H_0: the difference between the mean of the sample and the mean of the distribution[1] ($\mu_{measured} - \mu_{all}$) is 0.
- Alternative hypothesis H_1: the difference between the mean of the sample and the mean of the distribution ($\mu_{measured} - \mu_{all}$) is different from 0.

This is a so-called *two-tailed test*: either one mean can be larger than the other. One could also test for $\mu_{measured} > \mu_{all}$ (this is a *right-tail test*) or for $\mu_{measured} < \mu_{all}$ (this is a *left-tail test*).

- The p-value is the probability, assuming that the null hypothesis H_0 is true[2], to see a difference between the means as large as the one computed from the sample to the (hypothetical) distribution mean.
- The significance level of a test is a threshold of probability α agreed to before the test is conducted. A typical value of α is 0.05. If the p-value of a test is less than α, the test rejects the null hypothesis. (The significance level α can be interpreted as the probability of rejecting the null hypothesis when it is actually true: a *type I error*. A *type II error* is the probability β of not rejecting the null hypothesis when it is actually false).
- Results of hypothesis tests are often communicated with a *confidence interval*. A confidence interval is an estimated range of values with a specified probability of containing the true population value of a parameter.

Practical application of a statistical test boils down to essentially applying one or more mathematical operations (*tests*) to the data under study to obtain a p-value that will either reject or fail to reject the null hypothesis H_0. The choice of these mathematical operations is however dependent on several factors, and indeed statistics books will usually contain *decision tables* that will help in the selection of the appropriate test to perform. Essential factors to consider when selecting the appropriate statistical procedure are (i) the *type of the data,* i.e. whether one tries to compare *categorical to categorical*, *continuous to continuous*, or *categorical to continuous* variables; (ii) whether the test involves computing a correlation coefficient/measure of association between the variables; (iii) whether the study involves one or more samples; (iv) the size of the sample(s); (v) the probability distributions from which the samples are drawn (if known); (vi) if the samples are dependent or independent.

In chapters 3 and 4 we will discuss some selected examples. In chapter 5 we will look at a completely different approach to hypothesis testing.

3 Difference in location (Categorical vs. continuous variables)

3.1 Normal distribution: the *t*-test for two independent samples

If we could measure the sepal and petal widths and lengths of all Iris flowers in the world, we would have a definitive answer on whether *setosa* sepals are (at least on average!) smaller or larger than *versicolor* sepals. Let's imagine that we only have the

[1] All gas stations in Canton Basel Stadt.

[2] This is crucial for the correct interpretation of a p-value!

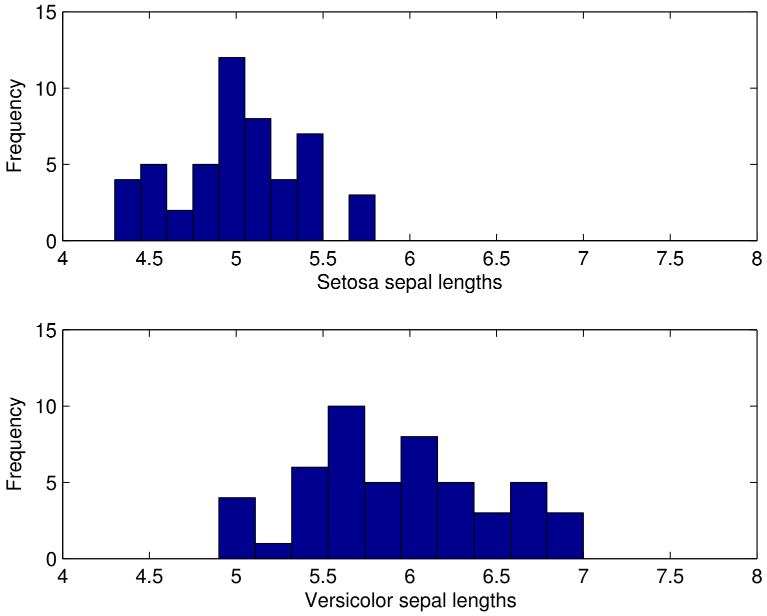

Figure 1: Distribution of the sepal lengths for *setosa* and *virginica*

sample of 50 measurements per species from out Fisher Iris dataset. How do we test for significant difference?

Let's start by creating our dataset (see also Session 3):

```
>> load fisheriris
>> NumObs = size( meas, 1 );
>> NameObs = strcat( { 'Obs' }, num2str( ( 1 : NumObs )', '%d' ) );
>> n = nominal( species );
>> iris = dataset( { n, 'species' }, ...
   { meas, 'SL', 'SW', 'PL', 'PW' }, 'ObsNames', NameObs );
```

We will compare the *sepal lengths* of *setosa* and *versicolor*.

First, we will plot the histograms of the sepal lengths of the two species for visual inspection.

```
>> figure;
>> subplot( 2, 1, 1 );
>> hist( iris.SL( iris.species == 'setosa' ) );
>> axis( [ 4 8 0 15 ] );
>> xlabel( 'Setosa sepal lengths' ); ylabel( 'Frequency' );
>> subplot( 2, 1, 2 );
>> hist( iris.SL( iris.species == 'versicolor' ) );
>> axis( [ 4 8 0 15 ] );
>> xlabel( 'Versicolor sepal lengths' ); ylabel( 'Frequency' );
```

Can we say from the histograms in Figure 1 that the sepal lengths of the two species are significantly different[3]? There seems to be quite some overlap: is the apparent shift just due to artifacts of the sampling? (After all, we only have 50 measurements per species.)

We formulate our statistical test as follows:

- *Null hypothesis* H_0: the difference between the means of the two samples ($\mu_{setosa} - \mu_{versicolor}$) is 0 (*two-tailed test*).
- Alternative hypothesis H_1: the difference between the means of the two samples ($\mu_{setosa} - \mu_{versicolor}$) is different from 0.

We will use the *t*-test for the comparison. The *t*-test is relatively robust with respect to departures from the normality assumption. In practice, the best way to assess the normality of the sample is *visually* by using the *normplot* function (as you should have done for the sepal lengths of setosa and versicolor as an exercise in Session 3), but one can also test for normality with the Lilliefors' composite goodness-of-fit test *(lillietest* function):

```
>> lillietest( iris.SL( iris.species == 'setosa' ) )

ans =
     0

>> lillietest( iris.SL( iris.species == 'versicolor' ) )

ans =
     0
```

If the *lillietest* returns 0, it indicates that the null hypothesis ("the data are normally distributed") cannot be rejected at the 5% significance level (meaning the sample *is* normally distributed[4]).

We can use the function *ttest2* to test if the two samples come from normal distributions with unknown but **equal** variances and the same mean, against the alternative that the means are unequal[5].

```
>> [ h, p, ci ] = ttest2( ...
   iris.SL( iris.species == 'setosa' ), ...
   iris.SL( iris.species == 'versicolor' ) )

h =
     1

p =
   8.9852e-18

ci =
   -1.1054    -0.7546
```

[3] We won't make any assumptions on which mean seems to be larger. We will just test for difference.

[4] At the same confidence level.

[5] *ttest2* assumes equal variance by default.

The null hypothesis is rejected ($h = 1$) at the default 5% significance level ($p < \alpha$): the samples do not come from the same underlying distribution. Indeed the confidence interval on the difference of the means does not include the hypothesized value of 0.

If the assumption of equal variance of the underlying distributions is not valid, one can instruct the *ttest2* function to perform the test assuming *unequal* variances (this is known as the Behrens-Fisher problem) as follows:

```
>> [ h, p, ci ] = ttest2( ...
   iris.SL( iris.species == 'setosa' ), ...
   iris.SL( iris.species == 'versicolor' ), ...
   0.05, 'both', 'unequal' )
h =
     1

p =
   3.7467e-17

ci =
   -1.1057    -0.7543
```

The additional parameters are the significance level (0.05, which was implicit in the previous call), the type of test ('both', that means perform a two-tailed test, again implicit) and the variance type 'unequal', which informs *ttest2* not to expect the same underlying distribution variance.

Even with unequal variances the null hypothesis is rejected ($h = 1$) at the default 5% significance level ($p < \alpha$). The result of the two tests is basically the same, suggesting that the difference between the means is so significant, that the simplification of equal variances does not play any role. **Exercise**: The *ttest2* function can also return the estimated variances. What are they?

Exercise

3.2 Unknown distribution: *the Wilcoxon test (rank-based)*

If it is obvious from the normal probability plot (*normplot*) or from the Lilliefors' composite goodness-of-fit test (*lillietest*) that your samples are not normally distributed, you can still test them for difference in location. One widely used test is the *Wilcoxon rank sum test*. The Wilcoxon rank sum test performs a two-sided rank sum test of the hypothesis that two independent samples come from distributions with equal medians. (The Wilcoxon rank sum test is equivalent to another known test, the Mann-Whitney U test.)

Let's create two samples from a Poisson distribution with $\lambda_1 = 3$ and $\lambda_2 = 6$:

```
>> sample1 = poissrnd( 3, 40, 1 ); % 40 random values
>> sample2 = poissrnd( 6, 60, 1 ); % 60 random values
```

Let's take a look at the histograms (Figure 2):

```
>> figure;
>> subplot( 2, 1, 1 );
>> hist( sample1, min( sample1 ):max( sample1 ) );
>> axis( [ 0 12 0 15 ] );
>> xlabel( 'lambda = 3' ); ylabel( 'Frequency' );
```

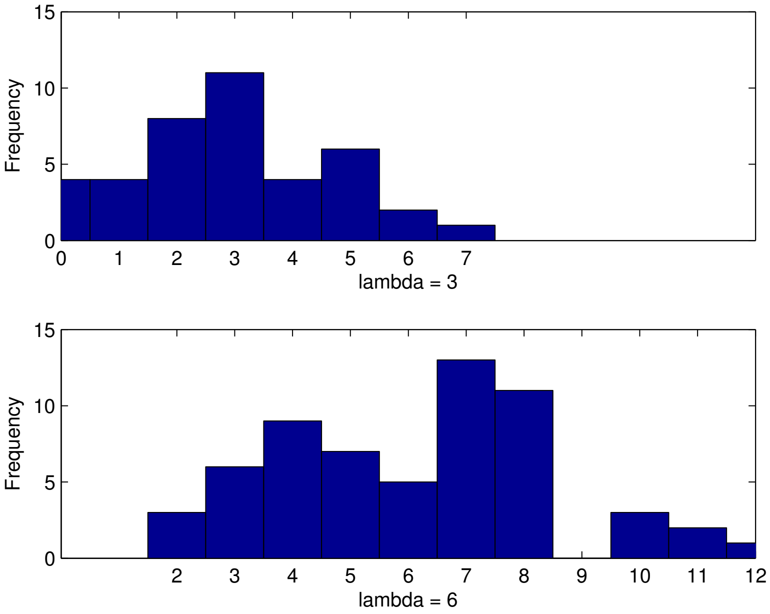

Figure 2: Distribution of Poisson random samples

```
>> subplot( 2, 1, 2 );
>> hist( sample2, min( sample2 ):max( sample2 ) );
>> axis( [ 0 12 0 15 ] );
>> xlabel( 'lambda = 6' ); ylabel( 'Frequency' );
```

We can perform the Wilcoxon rank sum test with the function *ranksum*:

```
>> [ p, h ] = ranksum( sample1, sample2, 'alpha', 0.05 )

p =
   1.9128e-09

h =
   1
```

The null hypothesis H_0 is rejected at the 0.05 significance level: the two samples are derived from different distributions.

4 Association between variables

One sometimes needs to know whether two variables of a given entity are associated. In other words, if we know the first variable (e.g. the height) of an entity (a human being), does this help us say something about the second variable (e.g. the weight) of the same entity?

Depending on whether the variables are categorical (e.g. 'methylated' vs. 'non methylated') or continuous (height in centimeters), the association can be summarized in a table (the *contingency table*) or measured by *correlation*, respectively.

4.1 Categorical vs. categorical variables

Assume that we know DNA methylation status and histone acetylation status for a number of promoters, summarized in a table as follows (this sample data is made-up and does not reflect biology):

```
>> DNAmet = [ 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 ];
>> H3acet = [ 1 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 ];
```

A value of 1 means that there is either a methylation or an acetylation, 0 means there is none. We can create a so-called *contingency table* like this:

```
>> table = crosstab( DNAmet, H3acet )[6]

table =
     9     1
     2     5
```

The table is to be interpreted like this:

	No H3acet	H3acet
No DNAMet	9	1
DNAMet	2	5

A contingency table can be used to express the relationship (or the *independence*) between two or more variables. Based on this table, we can reformulate our question of association between DNA methylation and histone acetylation more generally as: "Is there a correlation among DNA methylation and histone acetylation?" This is expressed by the comparison of the proportions of the various columns over the rows. So we can also reformulate the question as: Is there a difference between the rows (or columns) of the contingency table?" Intuitively, we would say yes: If DNA is unmethylated (first row in the contingency table), then it is also more likely for the histones not to be acetylated.

The statistical significance of the difference between the variables can be tested with a *Pearson's chi-square test*, a *G-test* or *Fisher's exact test*, provided the entries in the table represent a random sample from the population contemplated in the null hypothesis. The *chi-square test* should not be used if any of the entries in the contingency table are lower than 5 or the total number of measurement is lower than 20, since in this case the probabilities of the chi-square distribution may not provide an accurate estimate of the underlying sampling distribution. In this case, the *Fisher's exact test* should be used. We can perform the *Fisher's exact test* like this:

```
>> [ pr, pl, p ] = fisherextest( 9, 1, 2, 5 );
>> pr

p =
    0.9994
```

[6] The crosstab function also returns chi-square and p-value, but for such a small number of measurements the used test is far from optimal.

The four input parameters of the *fisherextest*[7] function are the elements of the contingency matrix, one row after the other. The *fisherextest* function can only be used for 2x2 contingency matrices. It returns up to three p-values for three different alternative hypotheses. Testing for independence is analogous to testing for zero correlation (two-tailed test). The other two alternative hypotheses would be testing for positive correlation (right tail) of the two variables (measurements mostly top-left and bottom-right), or for negative correlation (left tail, meaurements mostly top-right and bottom-left), respectively. An important note: in a statistic test one should **first** decide what hypothesis one wants to test and **then** decide which p-value is relevant! Let's test for positive correlation of DNA methylation and H3 acetylation since the measurement seem to concentrate on the top-left and bottom-right cells.

The null hypothesis H_0 that there is positive correlation between the two variables of interest **cannot** be rejected at the $\alpha = 0.05$ level.

4.2 Continuous vs. continuous variables

Correlation indicates the strength and direction of a linear relationship between two random variables and thus refers to their departure from independence. Depending on the underlying distribution, different approaches to calculate the *correlation coefficients* should be used.

4.2.1 Normal distribution: the Pearson correlation

The Pearson correlation coefficient makes use of the covariance of the variables of interest and their standard deviations and thus requires the samples to be drawn from a normal distribution. In MATLAB, the Pearson correlation can be calculated with the *corr* function.

```
% Create two independent, normally-distributed random samples
>> x  = normrnd( 10, 5, 100, 1 );
>> y1 = normrnd( 10, 5, 100, 1 );

% Create a new sample y2 that correlates with x
>> y2 = x + y1;

% Plot them side by side (see figure 3)
>> subplot( 1, 2, 1 ); plot( x, y1, '*' );
>> subplot( 1, 2, 2 ); plot( x, y2, '*' );

% Calculate the Pearson correlation coefficient
>> [r p] = corr( x, y1, 'type', 'Pearson' )

r =
   -0.0473

p =
    0.6405
```

[7] *This function is not part of the Statistics Toolbox: it can be downloaded from http://www.cis.hut.fi/Opinnot/T-61.5110/exercises/fisherextest.m.*

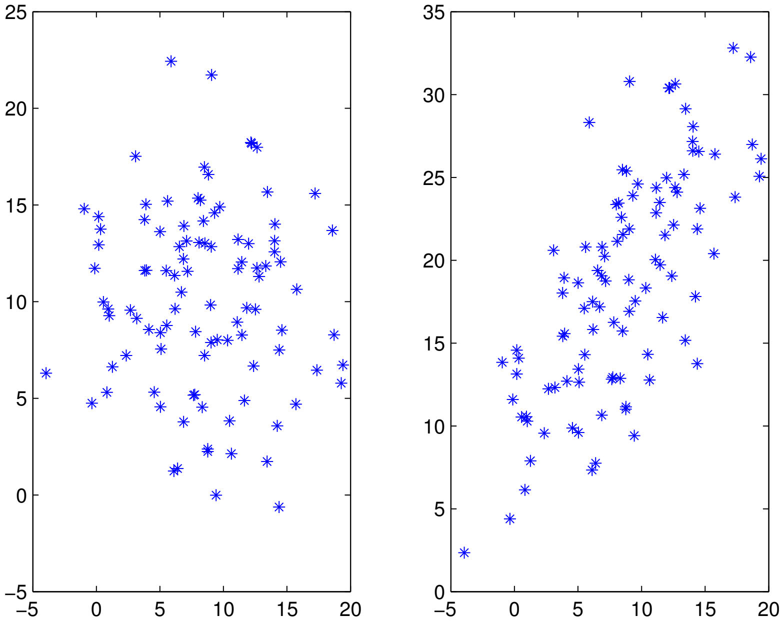

Figure 3: Non-correlating (left) vs. correlating (right) normally-distributed random samples.

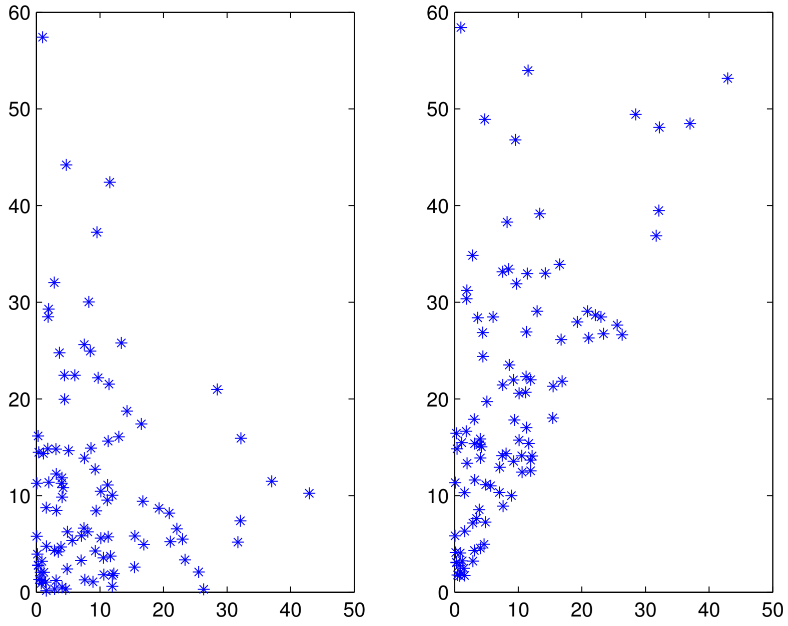

Figure 4: Non-correlating (left) vs. correlating (right) exponentially-distributed random samples.

```
>> [r p] = corr( x, y2, 'type', 'Pearson' )

r =
    0.7114

p =
   1.0841e-16
```

4.2.2 Unknown distribution: the Spearman correlation (rank-based)

The Spearman rank correlation coefficient (or Spearman's rho) often denoted by the Greek letter ρ or as *rs*, is another measure of correlation that does not make any assumptions about the distribution of the variables. The same function *corr* that we used to calculate the correlation coefficient in section 4.2.1 can be used to calculate the Spearman coefficient as follows:

```
% Create two independent, exponentially-distributed random samples
>> x  = exprnd( 10, 100, 1 );
>> y1 = exprnd( 10, 100, 1 );
```

```
% Create a new sample y2 that correlates with x
>> y2 = x + y1;

% Plot them side by side (see figure 4)
>> subplot( 1, 2, 1 ); plot( x, y1, ’*’ );
>> subplot( 1, 2, 2 ); plot( x, y2, ’*’ );

% Calculate the Spearman correlation coefficient
>> [r p] = corr( x, y1, ’type’, ’Spearman’ )

r =
    0.0718

p =
    0.4769

>> [r p] = corr( x, y2, ’type’, ’Spearman’ )

r =
    0.6304

p =
     0
```

5 Monte Carlo methods

Imagine we collected two measurement samples from our experiments and wanted to test for a difference in their medians. Unfortunately, we happen to have no idea of what is the underlying distribution from which we sampled. We have seen earlier how can we test for normality and how we could test even in case of unknown distributions. But we can also write our own test, using a Monte Carlo approach.

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results. Monte Carlo methods are often used when simulating physical and mathematical systems in particular when it is infeasible or impossible to compute an exact result with a deterministic algorithm. While their application in statistical testing has already been proposed around 1950, the high computational costs have long prevented their use. Monte Carlo methods have the potential to completely replace all classical statistical tests for hypothesis testing. The advantage of most classical tests, however, is that they are (much) faster to calculate.

Before we test our two samples for difference, we start with a little game to get familiar with the Monte Carlo methods.

Let’s consider a square of side a and a circle inscribed into it (with radius $a/2$). The ratio between the area of the circle and the area of the square is:

$$\frac{\left(\frac{a}{2}\right)^2\pi}{a^2} = \frac{\pi}{4}$$

If we now imagined to drop little stones uniformly over the area of the square, the fraction f of stones that would fall within the circle’s area vs. stones that fall within the square but outside of the circle should be approximately $f = \pi/4$ (like the ratio of the areas). This means that from this ratio we can estimate the value of π simply as $\pi = 4f$.

We can use the *rand* function to sample randomly from a standard uniform distribution on the open interval (0,1). Let's start generating (x,y) coordinates for $N = 10$ stones randomly between $-a/2$ and $a/2$ (the value of a is not relevant):

```
>> N = 10; a = 5;
>> pos = a .* rand( N, 2 ) - a / 2

pos =
     1.5736   -1.7119
     2.0290    2.3530
    -1.8651    2.2858
     2.0669   -0.0731
     0.6618    1.5014
    -2.0123   -1.7906
    -1.1075   -0.3912
     0.2344    2.0787
     2.2875    1.4610
     2.3244    2.2975
```

Let's calculate the fraction of stones that fell within the circle:

```
f = numel(find(sqrt(pos(:,1).^2+pos(:,2).^2)<(a/2)))/N

f =
    0.9000
```

We said that $\pi \approx 4f$:

```
Pi = 4 * f

Pi =
    3.6000
```

This is quite a lousy approximation. But let's try incrementally increasing the number of stones and see how our estimation of π converges (by the way, the real π is 3.141592653589793...):

N	Pi	$100\% \cdot (Pi - \pi)/\pi$
10	3.6000	+14.59%
10^2	2.9200	−7.05%
10^3	3.1800	+1.22%
10^4	3.1552	+0.43%
10^5	3.1452	+0.11%
10^6	3.1425	+0.03%
10^7	3.1414	−0.01%
10^8	3.1416	+0.00%

But let's now go back to our two samples and the difference between their medians. We will write a function that implements the following algorithm:

```
1:  For two samples x and y:
2:  Calculate the real difference in medians
3:  Repeat lines (4-6) N (many) times:
4:    Randomize x and y
5:    Calculate the difference of the medians of the randomized samples
6:    Count +1 if more extreme than real difference
7:  Calculate p value based on count
```

This pseudocode makes sense if you remember the meaning of a p-value, namely the probability of observing by chance a measure as extreme as the real measure. This is literally what the function does: it simulates many chance observations (starting at line 3), and counts the number of times such a chance observation is more extreme than the real observation (line 6). The p-value can easily be calculated from this simulation: it is the number of time we saw a difference larger than the real one divided by the total number of trials.

The crucial point in the function is the pooling and resampling of the input vectors. This is our hypothesis H_0: by doing this we hypothesize that the two samples come from the **same** underlying distribution and that the differences we see are entirely explained by the sampling variability. If we are right, the calculated p-value will exceed the significance level. If we are wrong, we will have to reject H_0.

The real code[8] could look like this:

```
function p = testDiffMedian( x, y, N )

% Calculate the real difference
realDiff = abs( median( x ) - median( y ) );

% Pool all measurements from both samples
allMeas = [ x y ];

% Initialize counter and lengths
count = 0;
nX    = numel( x );
n     = numel( allMeas );

% Now do the Monte Carlo stuff
for i = 1 : N

    % Randomize the measurements
    randData = allMeas( randperm( n ) );
    sampleX = randData( 1 : nX );
    sampleY = randData( nX + 1 : end );

    % Calculate the difference of the randomized samples
    randDiff = abs( median( sampleX ) - median( sampleY ) );

    % Is it larger than the real difference?
    if randDiff >= realDiff
        count = count + 1;
```

[8] The code assumes x and y to be row vectors.

```
        end

    end

    % Calculate the p-value (add one pseudo-count)
    p = ( count + 1 ) / ( N + 1 );
```

The pseudocount added to both *count* and *N* is to make sure we don't get an unreasonable p-value of 0 (the function cannot estimate a p-value smaller than $1/N$). Since *N* is usually much larger than 1, the influence of this single pseudocount can be neglected. Let's try our *testDiffMedian* function on some random samples:

```
    >> x = 1 + rand( 1, 40 );
    >> median( x )

    ans =
        1.5266

    >> y = 1.2 + rand( 1, 60 );
    >> median( y )

    ans =
        1.7475

    p = testDiffMedian( x, y, 1000 )

    p =
        0.0370
```

Our function rejected the null hypothesis that the two samples have the same mean at the 0.05 significance level.

Exercise

Exercise: how would you modify *testDiffMedian* to perform a single-sided instead of a two-sided (two-tailed) test?

Exercise

Exercise: Write a modified version of *testDiffMedian* (call it *testDiffVar*) that tests fro the difference in the variances using a similar Monte Carlo approach.

6 References

1. The official Statistics Toolbox documentation:
   http://www.mathworks.com/access/helpdesk/help/toolbox/stats/ (html).
   http://www.mathworks.com/access/helpdesk/help/pdf_doc/stats/stats.pdf (pdf)

2. David J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Third Edition. Chapman & Hall/CRC