R tutorial, session 4

H. Seitz (IGH)
(herve.seitz@igh.cnrs.fr)

November 3, 2014

Contents

1 Comparing distributions: χ^2 test and Fisher's exact test **2**

2 Plotting boxplots **3**

3 Comparing more than two conditions: the ANOVA test **5**
3.1 One-way ANOVA 5
3.2 Multifactorial ANOVA 8

4 Control structures: loops and conditional tests **9**
4.1 Concepts 9
4.2 Usage example: plotting a volcano plot 11

1 Comparing distributions: χ^2 test and Fisher's exact test

These two tests can determine the significance of a difference in the distribution of category populations in two datasets. If we assume that the experiment compared the repartition of cells between the G1, S, G2 and M phase, in two experimental conditions ("untreated" and "treated"):

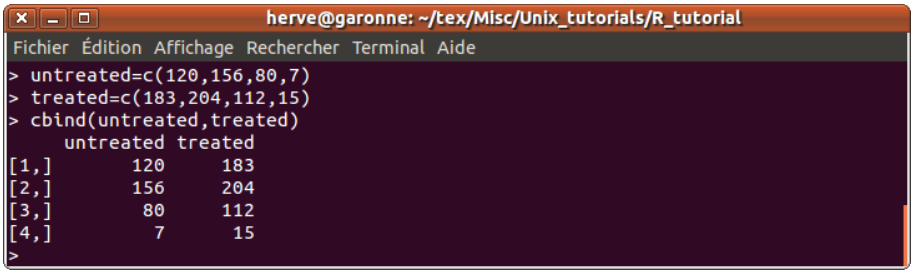

Figure 1: **Creating an array for distribution comparison.**

The χ^2 test is run by the command: `chisq.test` and Fisher's exact test, by the command: `fisher.test`. Each of these two commands takes a single mandatory argument: the array containing the distributions to be compared:

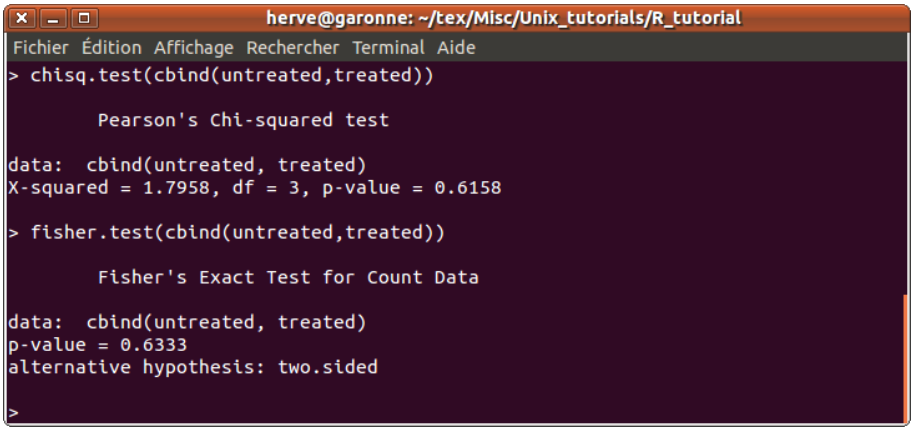

Figure 2: **Running the χ^2 test and Fisher's exact test.**

The χ^2 test is not very precise when populations are small (it is advised to avoid using that test when one of the populations is smaller than 5). In these cases, Fisher's exact test is preferable. When populations are large, the two tests tend to give very similar results:

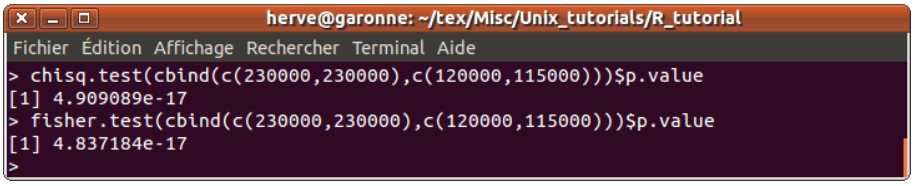

Figure 3: **Comparing the results of the χ^2 test and Fisher's exact test.**

So it is good practice to only use Fisher's exact test (in the worst case, it will give the same result than the χ^2 test). The only situation where Fisher's exact test is problematic is when populations are very large: the test can be very long to run, and sometimes (when numbers are really, very large) R is just unable to execute it:

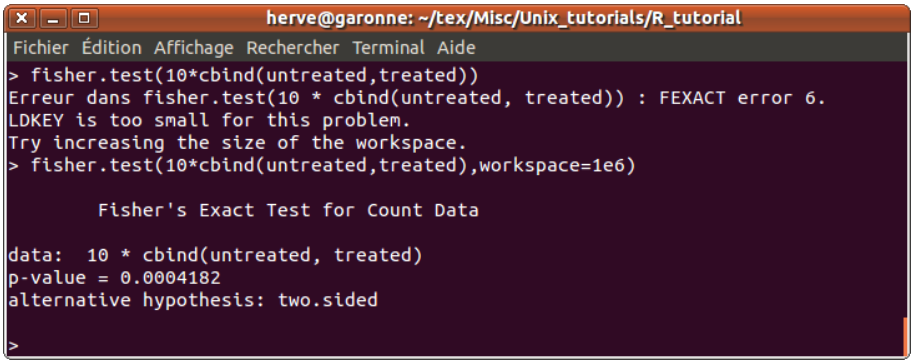

Figure 4: **Running Fisher's exact test on large datasets.** Here I had to change the amount of memory allocated to R (using the "workspace=" option) in order to perform the test. When the dataset is very large, this is not always sufficient: your computer may not have enough memory to perform the test.

2 Plotting boxplots

Boxplots can be helpful to display categorized datasets, when the number of points to be represented is too large to be easily visualized as individual points. Boxplots are traced using the `boxplot` command, with a vector of data from each category as arguments:

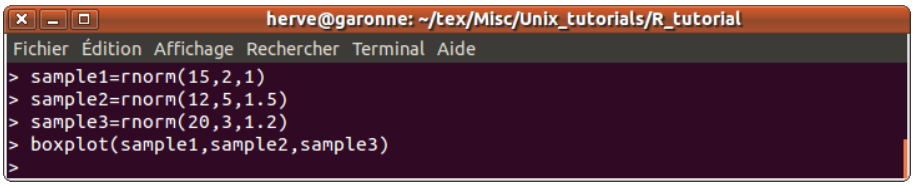

Figure 5: **Generating a boxplot.**

A pop-up window will appear, with the boxplot (*N.B.*: here the generated datasets are random, so the boxplots you will get with these commands should not be exactly identical to this one):

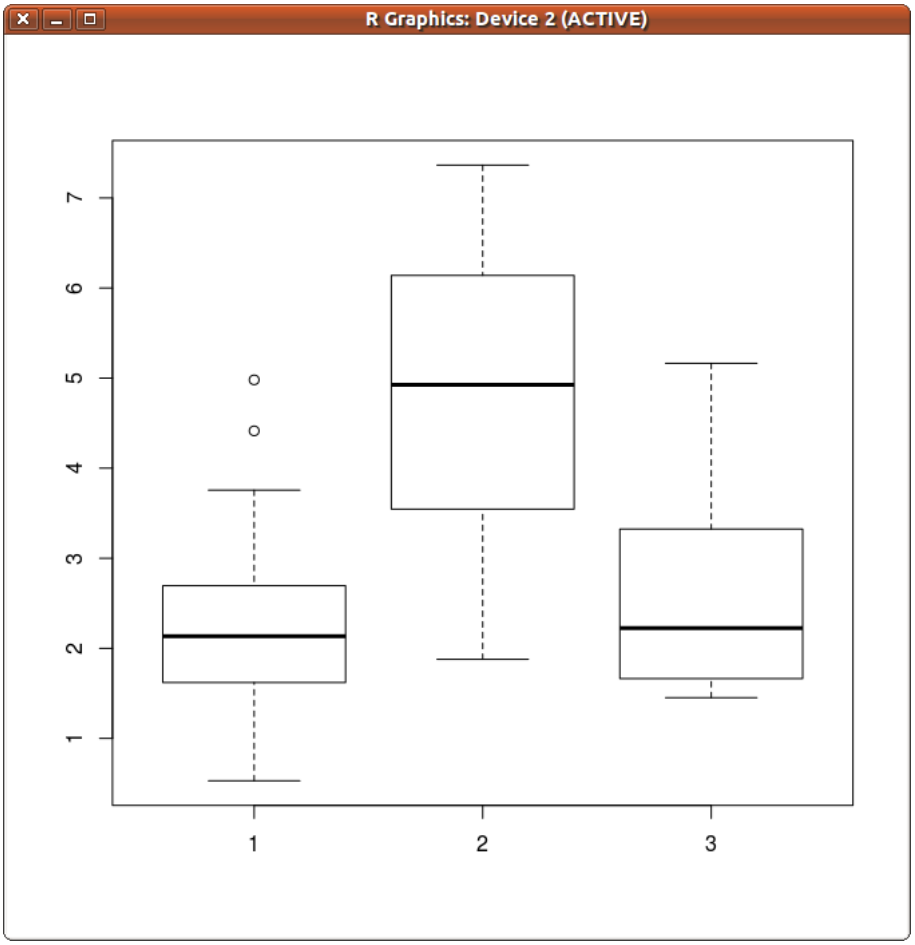

Figure 6: **A boxplot with default parameters.** Note that R follows the usual conventions for boxplots: the median is indicated as a thick line, the box delimits the 25th to the 75th percentiles, whiskers range to the most extreme non-outliers values, and outliers are indicated by circles. Usually (and it is the case with R's default settings) "outliers" are the values falling at a distance from the box that exceeds 1.5 times the height of the box.

Graphical parameters can be modified like with the `plot` command (see session 2), using the `xlab`, `xlim`, `col`, ... options. If you want to redefine x-axis tick labels, the easiest solution is to ask R not to trace axes, then tracing them afterwards:

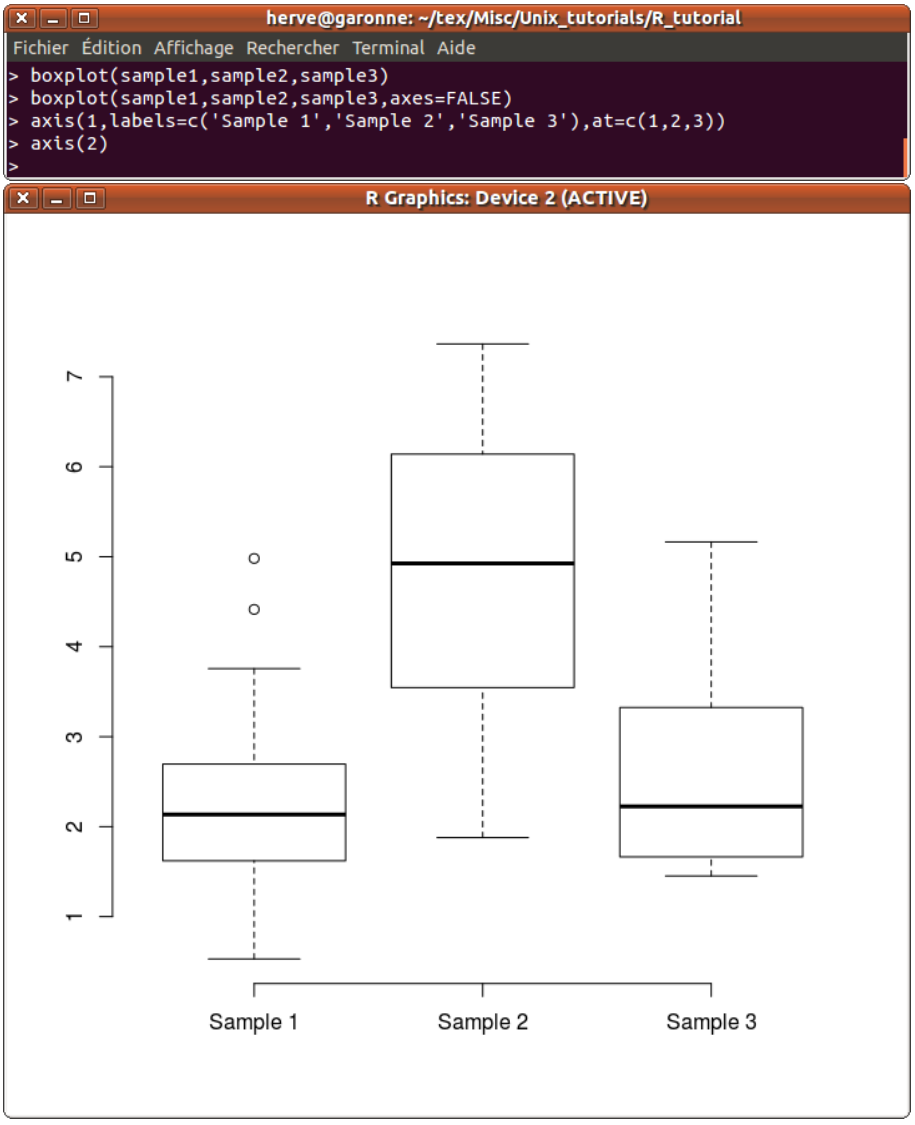

Figure 7: **Changing x-axis tick labels.**

3 Comparing more than two conditions: the ANOVA test

The ANOVA test (ANOVA stands for: *analysis of variance*) is a generalization of the t-test, for situations where more than two datasets are to be compared, or when several variables are to be assessed. Like Student's t-test, the ANOVA test needs data to follow a normal distribution, and to have homogeneous variances across samples.

3.1 One-way ANOVA

An ANOVA test is called "One-way ANOVA" when the effect of just one single variable is to be assessed. For example, when an experimenter treats cells with various drugs, and he wants to know whether some of these have an effect on a measurable output (*e.g.*, growth rate), he

can perform an ANOVA test to compare the growth rate after each treatment. It should be noted that the ANOVA test will only indicate whether some samples behave differently from others, without explicitly saying which ones are responsible for this heterogeneity: this can be addressed *a posteriori* by performing a pairwise t-test between the datasets (see below).

```
herve@sei-layon: ~/tex/Misc/Unix_tutorials/R_tutorials
File  Edit  View  Search  Terminal  Help
> rates=c(23,35,28,32,29,31,35,22,27,15,18,12,23,34,26,29)
> treatment=as.factor(c(rep('ethanol',times=4),rep('acetone',times=5),rep('DMSO'
,times=4),rep('benzene',times=3)))
>
```

Figure 8: **Defining data for one-way ANOVA analysis.** The first variable ("rate") is the concatenation of growth rates, measured in every replicate of every treatment. The second variable ("treatment") is a factor describing the treatment applied to each sample.

The test is run using the `anova` command:

```
herve@sei-layon: ~/tex/Misc/Unix_tutorials/R_tutorials
File  Edit  View  Search  Terminal  Help
> shapiro.test(rates[treatment=='ethanol'])$p.value
[1] 0.9058439
> shapiro.test(rates[treatment=='acetone'])$p.value
[1] 0.9923607
> shapiro.test(rates[treatment=='DMSO'])$p.value
[1] 0.9252345
> shapiro.test(rates[treatment=='benzene'])$p.value
[1] 0.7262263
> library(car)
> leveneTest(rates,treatment)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.1465   0.93
      12
> anova(lm(rates~treatment))
Analysis of Variance Table

Response: rates
          Df Sum Sq Mean Sq F value   Pr(>F)
treatment  3 451.97 150.657  6.6352 0.006829 **
Residuals 12 272.47  22.706
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Figure 9: **One-way ANOVA analysis.** First we have to verify that each dataset follows a normal distribution, using the Shapiro-Wilk test, and that their variances are homogeneous, using the Levene test.

Here, the low p-value (0.006829) indicates that there is a significant difference between some of the treatments and the others. Further tests (called "post-hoc tests") are needed to identify the origin of the difference. For example, pairwise t-test comparisons show that the difference is mainly due to the "DMSO" samples, which behave significantly differently from the other three treatments:

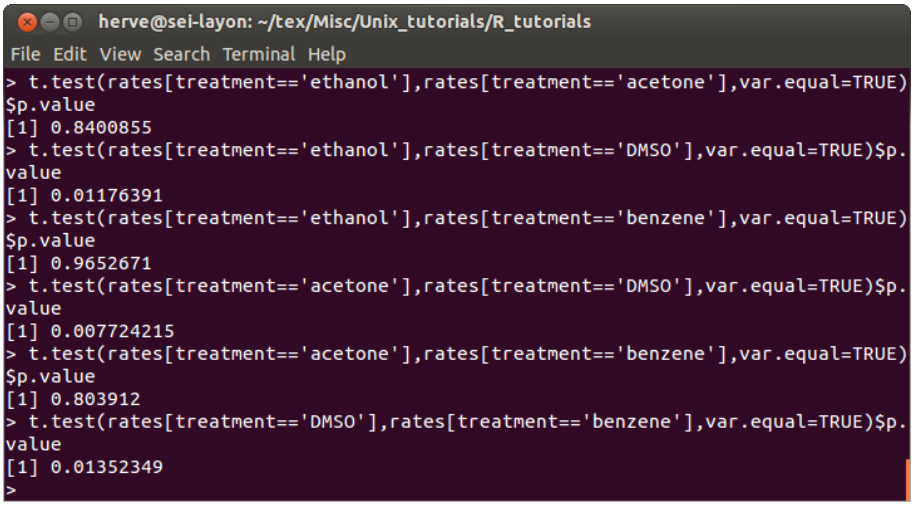

Figure 10: **"Post-hoc" analyses to identify the origin of the difference between treatments.** Here a series of pairwise t-tests is used to spot the samples that behave differently from others (alternative post-hoc methods exist).

It is possible to display graphically the results with a special usage of the `plot` command: instead of providing it a vector of x values and a vector of y values (*cf* session 2), we will provide it with the factor describing samples (factor "treatment") and the response values ("rates"). When the x value vector is replaced by a factor in the `plot` command, R plots a boxplot instead of a scatter plot:

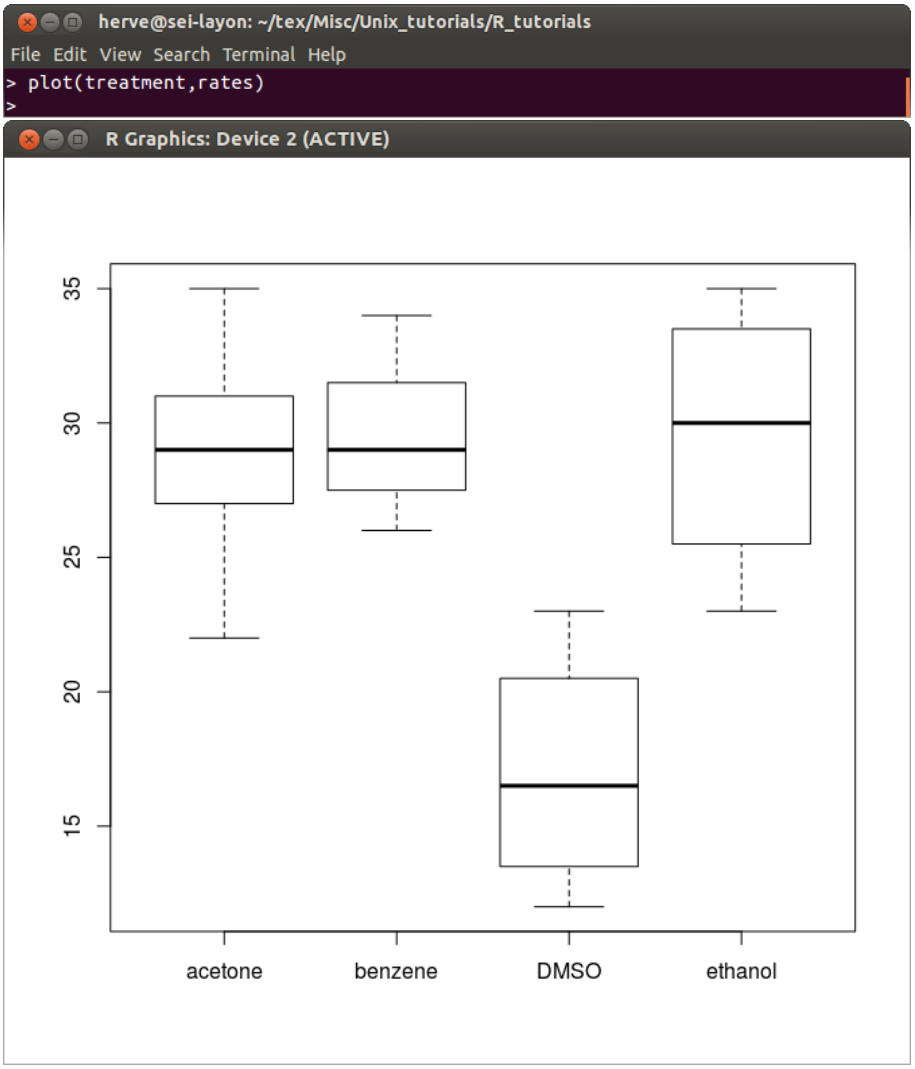

Figure 11: **Graphical representation of the data.**

3.2 Multifactorial ANOVA

Several factors may influence the system's response, and the experimenter may want to test the influence of each of them. For example, in addition to chemical treatment (ethanol, acetone, DMSO or benzene) the experimenter could test the effect of temperature on cell growth. The experiment could then measure the effect of chemical treatment, as well as the effect of temperature, and the cross effect of these two factors (*e.g.*, DMSO may slow down cell growth at 37°C but accelerate it at 30°C). The significance of such effects can be assessed using a "multifactorial ANOVA test":

```
herve@sei-layon: ~/tex/Misc/Unix_tutorials/R_tutorials
File  Edit  View  Search  Terminal  Help
> rates=c(23,35,28,32,29,31,35,22,27,15,18,12,23,34,26,29,18,20,15,21,19,25,28,2
3,24,27,21,20,18,24,20,27,21,22,24)
> treatment=as.factor(c(rep('ethanol',times=4),rep('acetone',times=5),rep('DMSO'
,times=4),rep('benzene',times=3),rep('ethanol',times=5),rep('acetone',times=5),r
ep('DMSO',times=5),rep('benzene',times=4)))
> temperature=as.factor(c(rep(37,times=16),rep(30,times=19)))
> shapiro.test(rates[treatment=='ethanol' & temperature==37])$p.value
[1] 0.9058439
> shapiro.test(rates[treatment=='ethanol' & temperature==30])$p.value
[1] 0.6852955
> shapiro.test(rates[treatment=='acetone' & temperature==37])$p.value
[1] 0.9923607
> shapiro.test(rates[treatment=='acetone' & temperature==30])$p.value
[1] 0.753973
> shapiro.test(rates[treatment=='DMSO' & temperature==37])$p.value
[1] 0.9252345
> shapiro.test(rates[treatment=='DMSO' & temperature==30])$p.value
[1] 0.607098
> shapiro.test(rates[treatment=='benzene' & temperature==37])$p.value
[1] 0.7262263
> shapiro.test(rates[treatment=='benzene' & temperature==30])$p.value
[1] 0.6889364
> leveneTest(rates~treatment*temperature)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  7  1.0763 0.4052
      27
> anova(lm(rates~treatment*temperature))
Analysis of Variance Table

Response: rates
                    Df Sum Sq Mean Sq F value    Pr(>F)
treatment            3 355.56 118.521  9.1153 0.0002472 ***
temperature          1 143.86 143.860 11.0640 0.0025457 **
treatment:temperature  3 243.05  81.018  6.2309 0.0023498 **
Residuals           27 351.07  13.002
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Figure 12: **Multifactorial ANOVA analysis.** In this example, two factors (chemical treatment and temperature) are assessed, as well as their cross effect. It is also possible to analyze the effect of more variables (*e.g.* if a factor "cell_line" was defined: treatment*temperature*cell_line).

Here, each of the two variables (chemical treatment and temperature) has a significant effect on cell growth, but also the cross effect of chemical treatment with temperature (each of the 3 p-values is low).

4 Control structures: loops and conditional tests

4.1 Concepts

It is possible to ask R perform similar operations a large number of times, using the `for` command:

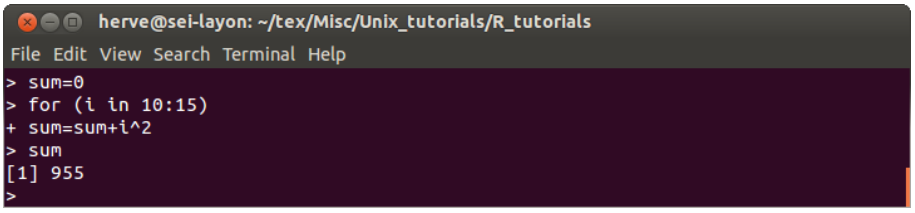

Figure 13: **A "for" loop.** For each integer value of "i" between 10 and 15, i^2 is added to "sum".

When more than one operation have to be performed in each cycle, the list of operations should be placed between curly brackets (otherwise, R would only perform the first one at each cycle):

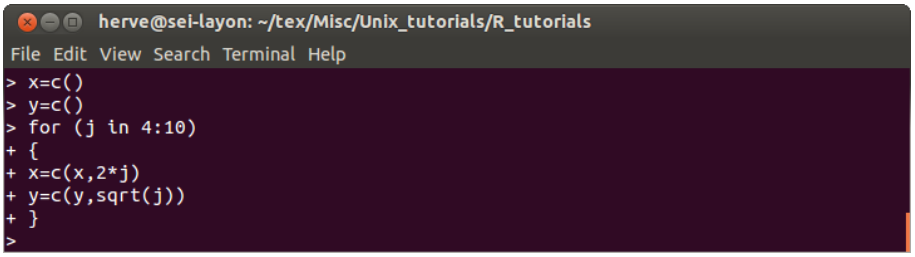

Figure 14: **A "for" loop containing several instructions.**

It is also possible to execute conditional operations (they will be executed only if some conditions are met), using the command `if`:

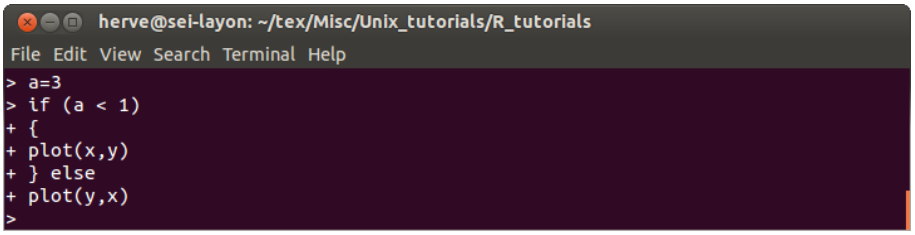

Figure 15: **An "if" test.** If several instructions are to be executed in the "else" case, they have to be embedded between curly brackets too. The "else" part is optional: it can be omitted if there is nothing to do when the tested condition is not met.

You may also want to repeat some operations, till some conditions are met. This can be done using the `while` command:

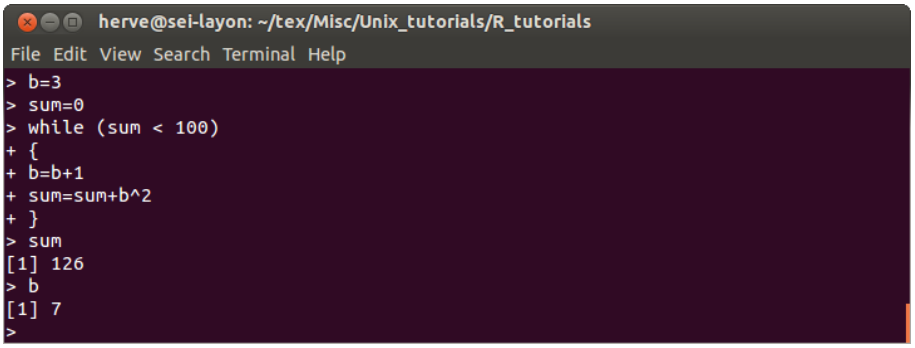

Figure 16: **A "while" loop.**

4.2 Usage example: plotting a volcano plot

Let's assume we have a data file containing gene expression values for a list of genes (three replicates of wild-type samples and three replicates of mutant samples for each gene): see data file 'for_volcano_plot.txt'.

We will first load the data file and compute the mean expression value across all three wt replicates, and across all three mutant replicates, for each gene:

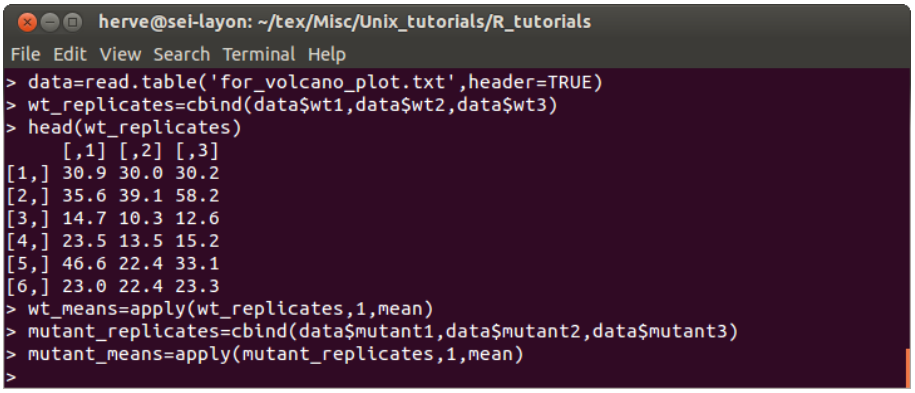

Figure 17: **Loading data and calculating means for wt and mutant samples.**

Then we will verify that the data follows a normal distribution and we will compare the variances of the wt and mutant samples:

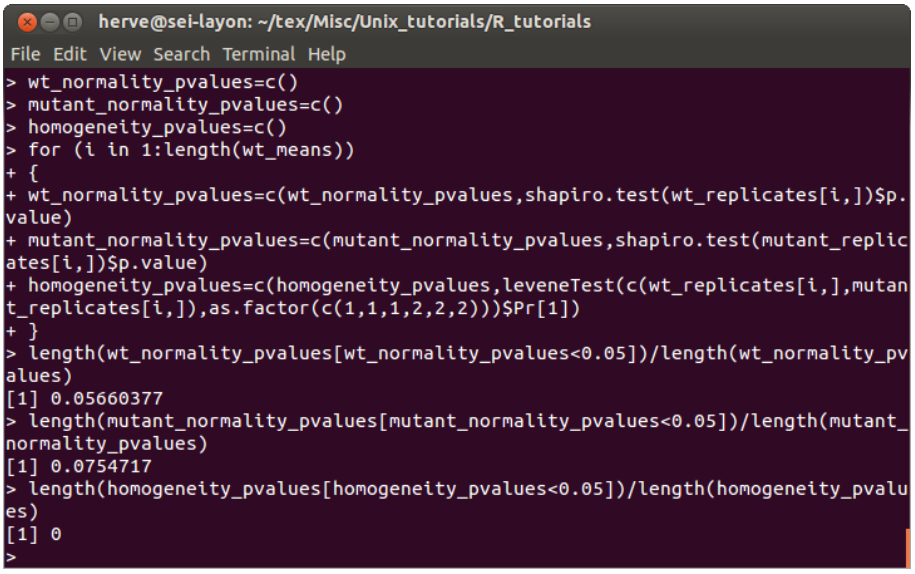

Figure 18: **Verifying t-test assumptions.** Here, only 6 to 7.5% of the genes display Shapiro-Wilk test p-values below 0.05, which is very close to the expected 5% under the assumption of normality: we can consider that the data is normally distributed. Variances appear to be homogeneous, because Levene test's p-value is larger than 0.05 for every gene.

We can then calculate the t-test p-value for each gene, assuming equal variances:

```
herve@sei-layon: ~/tex/Misc/Unix_tutorials/R_tutorials
File  Edit  View  Search  Terminal  Help
> pvalues=c()
> for (i in 1:length(wt_means))
+ pvalues=c(pvalues,t.test(wt_replicates[i,],mutant_replicates[i,],var.equal=TRU
E)$p.value)
>
```

Figure 19: **Calculating t-test p-values across the list of genes.**

The results can be conveniently displayed as a "volcano plot", where each gene is represented by its fold-change (on the x axis) and its p-value (on the y axis):

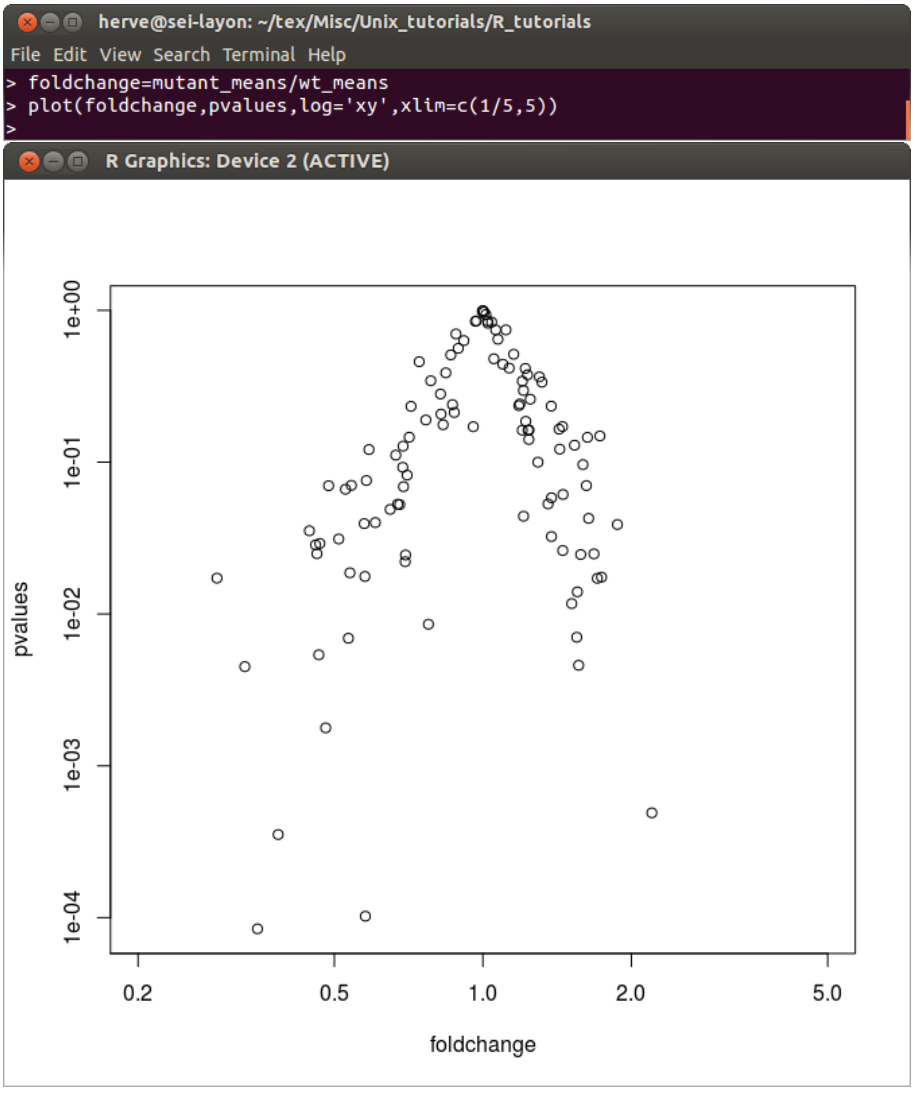

Figure 20: **Calculating fold-changes and plotting a volcano plot.**

Setting cutoffs on the fold-change and on the p-value, it is also possible to emphasize significantly misregulated genes:

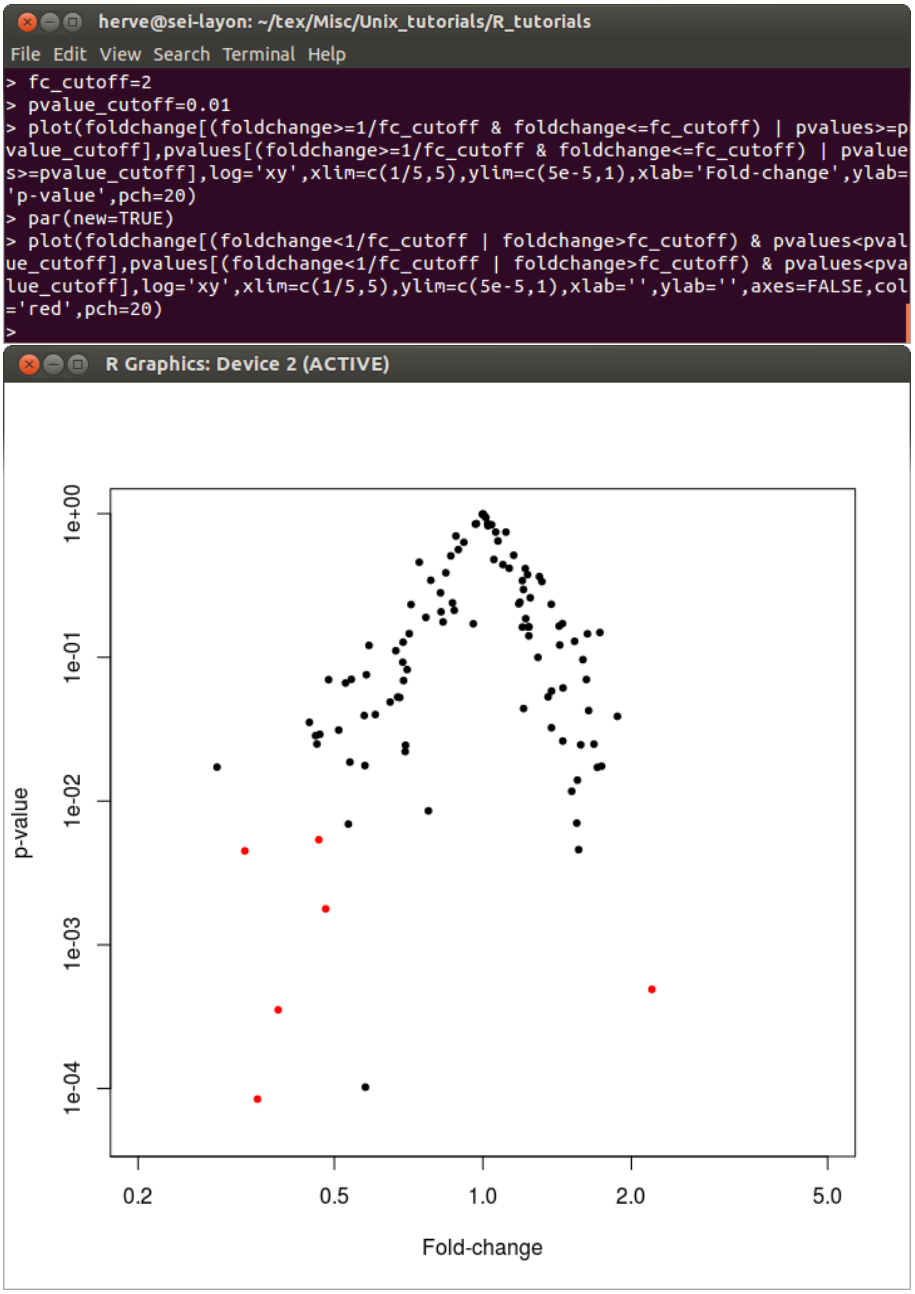

Figure 21: **Displaying significantly misregulated genes in red.**

Let's add an annotation of misregulated genes, using the `text` command:

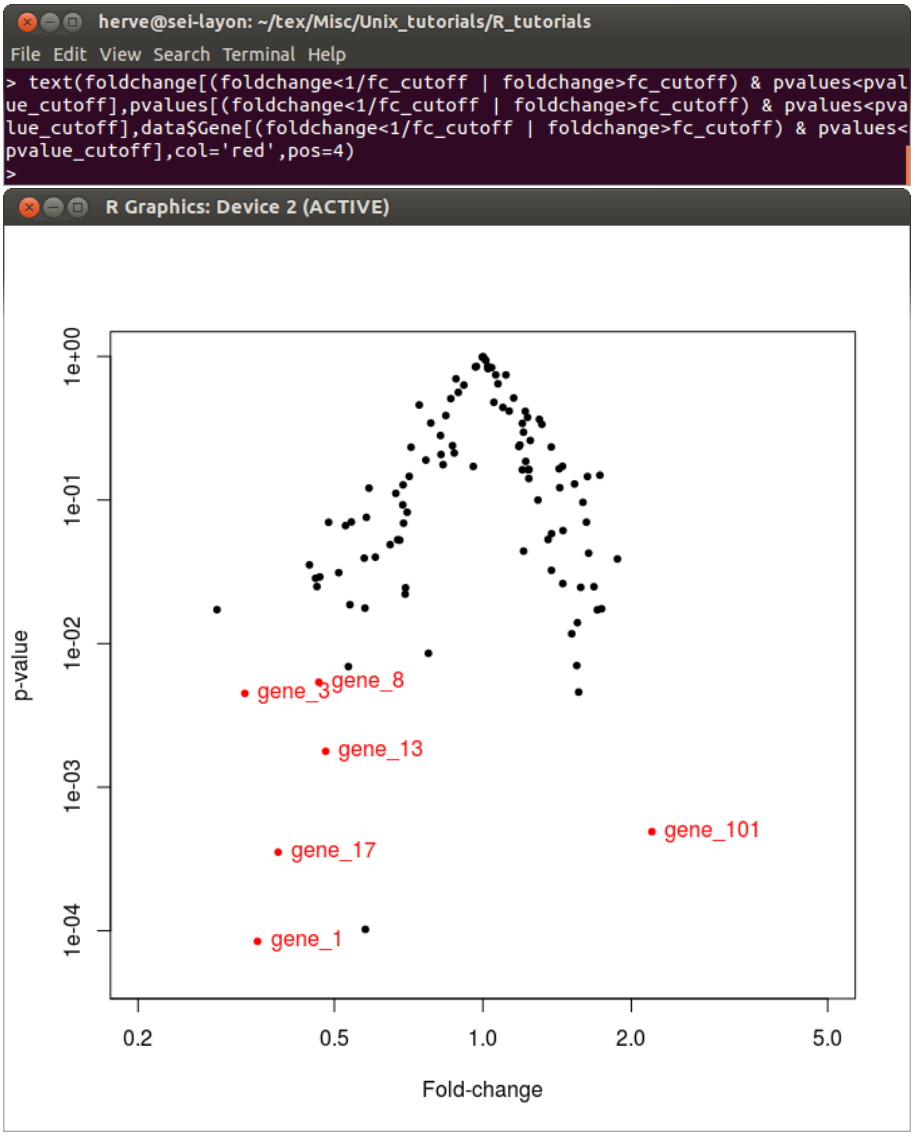

Figure 22: **Annotating selected points on the volcano plot.**

As there is a large number of tested hypotheses (106 in this example), it is better to correct p-values using the `p.adjust` command. In the following example, I corrected p-values using the Benjamini-Hochberg method, and applied a cutoff of 0.05 on corrected p-values:

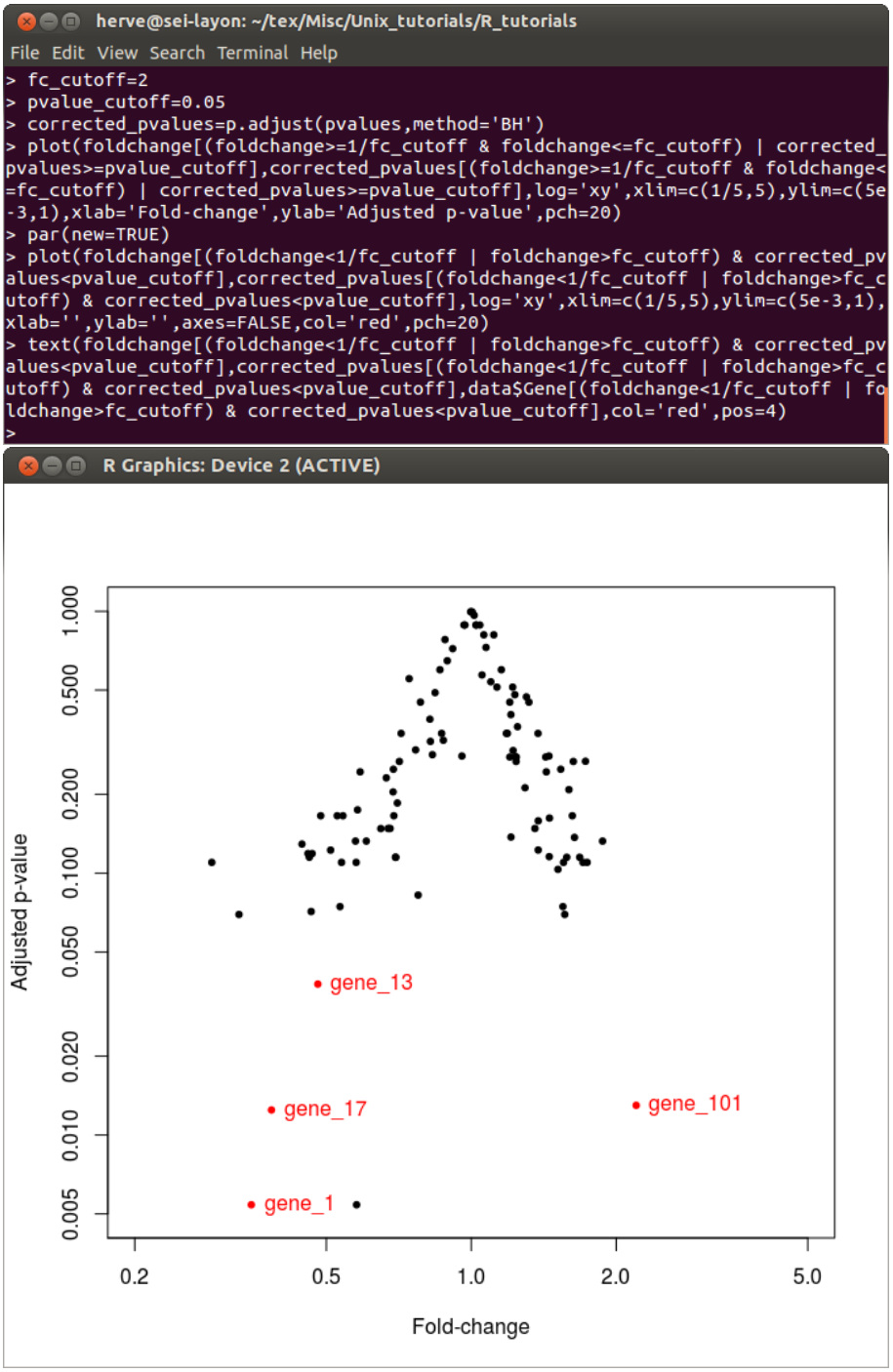

Figure 23: **After p-value correction.**