Applied Statistics and Data Analysis

Lab 3b: A review of inference concepts - Statistical inference

Luca Grassetti and Paolo Vidoni
Department of Economics and Statistics, University of Udine

September, 2019

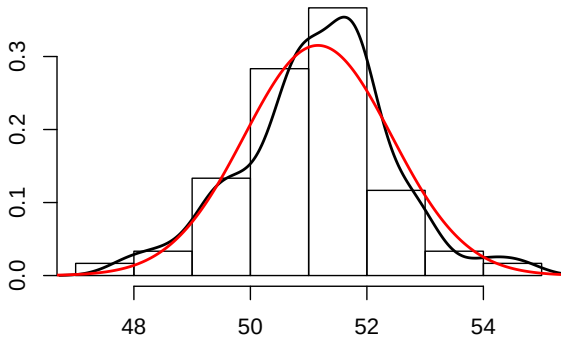
1 Introduction to statistical inference

1.1 Example: temperatures

The mean annual temperatures (°F) in New Haven, Connecticut, from 1912 to 1971 (data set `nhtemp`, available in the R system libraries) are used here to show how an empirical probability distribution can be associated with a convenient statistical model, specified by considering suitable values for the unknown parameters. As a matter of fact, the analysis of a interest phenomenon usually begins with the estimation of the model parameters.

Firstly, a graphical representation of the density function can be obtained by means of the histogram (with the option `freq=F`), the kernel density estimator and, in this particular case, the Gaussian density functions with parameters estimated by the sample mean and the sample standard deviation (that is, by `mean(nhtemp)` and `sqrt(var(nhtemp))`).

```
hist(nhtemp,freq=F,main=' ',xlab=' ',ylab=' ')
lines(density(nhtemp),lwd=2)
lines(seq(45,60,0.01),dnorm(seq(45,60,0.01),mean(nhtemp),
                            sqrt(var(nhtemp))),col='red',lwd=2)
```



In order to obtain parameter estimates, the following functions can be useful:

- `mean`, which computes the sample mean; this function presents two main arguments: `trim`, which allows to specify a proportion (from 0 to 0.5) of extreme observations to be trimmed, and `na.rm`, which allows to omit the `NA` values;
- `median`, which is used to compute the median, with the possible option `na.rm`;
- `var`, which gives the corrected sample variance; this function can be used also for computing the variance-covariance matrix, if the first argument is a data matrix; a further option, in addition to `na.rm`, is `use` which allows the specification of alternative methods for computing the covariances;
- `sd`, which computes the sample standard deviation (that is, the positive the square root of the variance), with the possible option `na.rm`.

```
mean(nhtemp)

[1] 51.16

median(nhtemp)

[1] 51.2

var(nhtemp)

[1] 1.601763

sd(nhtemp)

[1] 1.265608
```

These functions can be used to estimate the third and the fourth central moments, which may be considered for evaluating skewness and kurtosis of the probability distribution, respectively.

```
mean((nhtemp-mean(nhtemp))^3)/sqrt(var(nhtemp))^3
```

```
[1] -0.07178758
```

```
mean((nhtemp-mean(nhtemp))^4)/sqrt(var(nhtemp))^4
```

```
[1] 3.383275
```

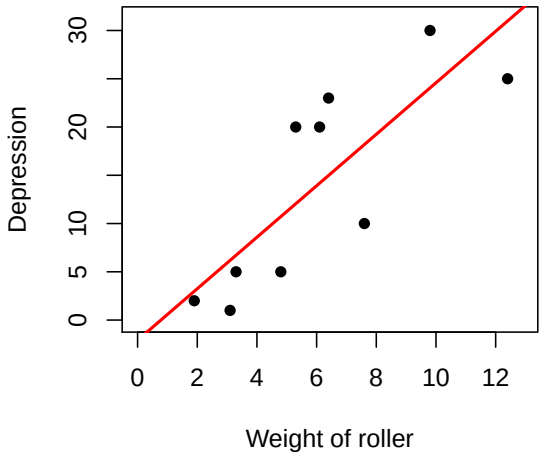
1.2 Example: roller data

The roller data (data set `roller` of the library `DAAG`) are used to present a first application of a simple linear regression model. The estimates of the model parameters can be obtained by considering the specific R function or, alternatively, by using some basic R commands.

The scatterplot given below illustrates the relationship between `weight` and `depression` and it is simply obtained by using the function `plot` with the argument `depression~weight`. The estimated regression line (in red) is drawn using function `abline`, with the first argument given by the result of the function `lm` applied to the available data.

Function `lm` specifies a linear regression model and it gives a list containing some useful elements, such as the least square estimates of the regression parameters. The first value corresponds to the intercept and the second value to the slope of the estimated regression line.

```
library(DAAG)
plot(depression ~ weight, data = roller,
     xlim=c(0,1.04*max(weight)),ylim=c(0,1.04*max(depression)),
     xlab = 'Weight of roller', ylab = 'Depression', pch = 16)
roller.lm <- lm(depression ~ weight,data=roller)
abline(roller.lm,col='red',lwd=2)
```




```
roller.lm$coef

(Intercept)      weight
  -2.087148    2.666746
```

The same results can be obtained by considering the following lines of code where function `t` is used to transpose a matrix and function `solve` computes the inverse of a square matrix. More precisely, we compute explicitly the parameter values which minimize the sum of squared errors, given when the observed data are described using a regression line.

```
X <- cbind(1, roller$weight)
y <- roller$depression

solve(t(X)%*%X)%*%(t(X)%*%y)

          [,1]
[1,] -2.087148
[2,]  2.666746
```

2 Basic concepts of point estimation

2.1 Example: elastic bands

The estimation of the difference of two population means can be obtained by considering, as point estimator, the difference between the two sample means or, in case of paired observations, the sample mean of the differences. We consider data from an experiment on the effect of heat on the amount of stretch (mm) of elastic bands; 21 bands were randomly divided into two groups: those

ones observed at the ambient temperature and those ones observed at heated temperature. The difference between the group means is computed.

```
ambient <- c(254, 252, 239, 240, 250, 256, 267, 249, 259, 269)

heated <- c(233, 252, 237, 246, 255, 244, 248, 242, 217, 257, 254)

mean(ambient)-mean(heated)

[1] 9.409091
```

The variance of the corresponding estimator is the pooled sample variance, namely the weighted mean of the two group specific sample variances.

```
s2p <- ((var(heated))*(length(heated)-1)
        +(var(ambient))*(length(ambient)-1))/
        (length(heated)+length(ambient)-2)
s2p

[1] 119.1268
```

Then, the pooled sample variance is used to estimate the standard error for the difference (SED).

```
sqrt(s2p)*sqrt((1/length(heated))+(1/length(ambient)))

[1] 4.768898
```

Indeed, the ratio between the mean difference and the SED can be used to evaluate the mean difference in terms of estimated standard error units.

```
(mean(ambient)-mean(heated))/(sqrt(s2p)*
      sqrt((1/length(heated))+(1/length(ambient))))

[1] 1.973012
```

2.2 Point estimators and their properties

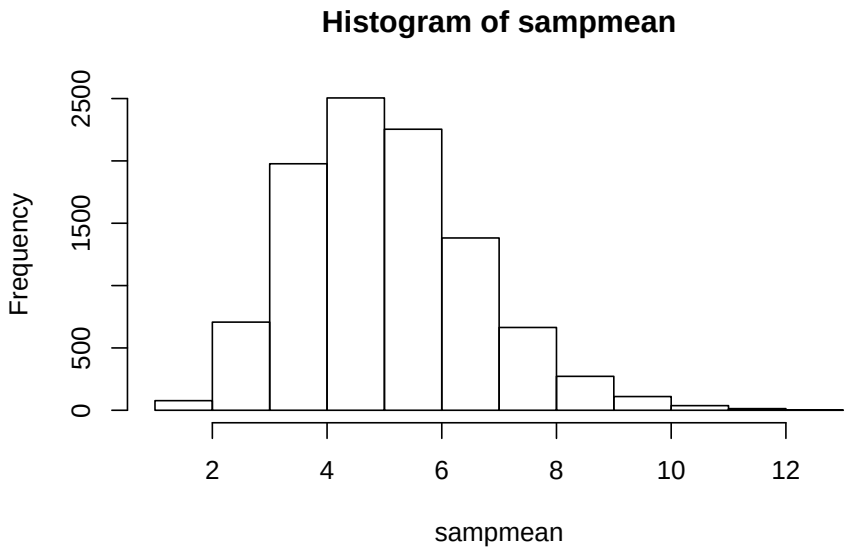
The sample mean is an unbiased, consistent estimator of the true population mean. The following simulation study confirms empirically this statement. We generate a sample of 10000 observations from an exponential distribution with mean 5 (hazard rate 1/5), and then variance 25. The sample mean and the corrected sample variance are very close to the true parameter values.

```
N <- 10000
set.seed(10)
samp <- rexp(N,1/5)
mean(samp)

[1] 5.029297

var(samp)

[1] 25.01032

sd(samp)

[1] 5.001032
```

Since an estimator is a sample statistic, which is in fact a random variable, we present a further simulation study with the aim of estimating its sampling distribution, as well as its mean and its standard error. We generate 10000 different samples of size 10 from an exponential distribution with mean 5 and, for each sample, we compute the sample mean. The histogram, based on these values, estimates the density function (namely, the sampling distribution) of the sample mean. Indeed, the mean of the sample means is very close to the true population mean and the standard deviation is close to the true standard error of the sample mean, given by $5/\sqrt{10} = 1.5811$.

```
set.seed(10)
repl<-10000
n <- 10
sampmean <- NULL
for (i in 1:repl)
{
  sam <- rexp(n,1/5)
  sampmean <- c(sampmean,mean(sam))

}
```

```
hist(sampmean)
```




```
mean(sampmean)

[1] 5.03592

sd(sampmean)

[1] 1.577941
```

Furthermore, it is well-known that the corrected sample variance is an unbiased estimator of the population variance, while the uncorrected one tends to underestimate the true value. We consider the same samples already generated from the exponential distribution and we compute both the corrected and the uncorrected sample variances for each sample. We find out that, as expected, the mean of the corrected sample variances is very close to the true population variance and the mean of the uncorrected ones presents an evident negative bias.

```
set.seed(10)
repl<-10000
n <- 10
sampvar <- NULL
variance <- NULL
for (i in 1:repl)
{
  sam <- rexp(n,1/5)
  sampvar <- c(sampvar,var(sam))
  variance <- c(variance,var(sam)*9/10)
}

mean(sampvar)
```

```
[1] 25.24354

mean(variance)

[1] 22.71919
```

3 Basic concepts of interval estimation

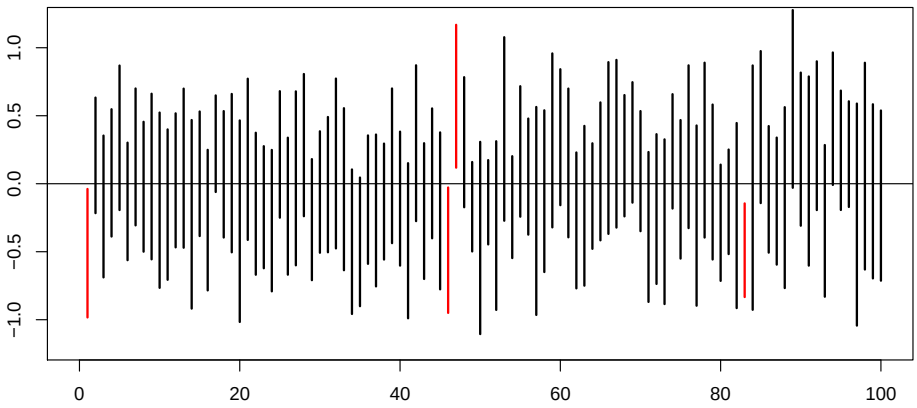
3.1 Confidence interval for the mean

In order to show how an interval estimation procedure works we develop a simulation study regarding the computation of 95% confidence intervals for the mean of a normal population. In particular we consider 100 replications of the same experiment by considering the following steps:

1. set the seed of the simulation procedure;
2. initialize the `flag` counter and replicate 100 times the subsequent steps 3., 4., 5.;
3. generate a sample of 15 observations from a standard Gaussian distribution;
4. compute the confidence interval limits using the sample mean, the estimated standard error and `qt(0.975, df=14)`, that is the 0.975-quantile of the Student's t distribution with 14 degrees of freedom;
5. if the interval includes the true value 0, it is represented as a black segment and the flag count is increased by 1; if the interval does not include 0, it is represented as a red segment and the flag is not updated;
6. an horizontal line is added to the plot at level 0;
7. the proportion of intervals including zero is computed.

```
# Step 1)
set.seed(12)
# Step 2)
flag <- 0
# Step 3)
y <- rnorm(15)
# Step 4)
ci <-c(mean(y)-qt(0.975,df=14)*sd(y)/sqrt(15),
       mean(y)+qt(0.975,df=14)*sd(y)/sqrt(15))
# Step 5)
if(ci[1]*ci[2]<0){
plot(c(1,1),ci,ylim=c(-1.1,1.2), xlim=c(0,100),
```

```
     type='l',xlab=' ',ylab=' ',lwd=2)
flag <- flag+1}
if(ci[1]*ci[2]>0){
plot(c(1,1),ci,ylim=c(-1.2,1.2), xlim=c(0,100),
     type='l',xlab=' ',ylab=' ',col='red',lwd=2)}
for (i in 2:100){
# Step 3)
y <- rnorm(15)
# Step 4)
ci <-c(mean(y)-qt(0.975,df=14)*sd(y)/sqrt(15),
       mean(y)+qt(0.975,df=14)*sd(y)/sqrt(15))
# Step 5)
if(ci[1]*ci[2]<0){
lines(c(i,i),ci,type='l',lwd=2)
flag <- flag+1
}
if(ci[1]*ci[2]>0){
lines(c(i,i),ci,type='l',col='red',lwd=2)
}
}
# Step 6)
abline(0,0)
```




```
# Step 7)
flag/100

[1] 0.96
```

The estimated confidence level is 0.96, close to the nominal value 0.95.

3.2 Confidence intervals with hypothesis testing commands

The confidence intervals, obtained in the previous simulation procedure, can be easily calculated using the `t.test` function, which is considered for testing statistical hypothesis on the mean of a Gaussian population. In particular, we set the command for a two-sided test with the null hypothesis $\mu = 0$, which correspond to the default options. The confidence level is specified by the option `conf.level`, with 0.95 as default value.

In particular, the first confidence interval of the above simulation study is

```
set.seed(12)
y <- rnorm(15)
ci <-c(mean(y)-qt(0.975,df=14)*sd(y)/sqrt(15),
       mean(y)+qt(0.975,df=14)*sd(y)/sqrt(15))
print(ci)

[1] -0.98335647 -0.03847978
```

ant it can be easily calculated using the `t.test` function

```
t.test(y)


One Sample t-test

data:  y
t = -2.3195, df = 14, p-value = 0.03599
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.98335647 -0.03847978
sample estimates:
 mean of x
-0.5109181
```

This result is an object with the following elements, where, in particular, the element `conf.int` gives the confidence interval for the mean with the specified confidence level.

```
attributes(t.test(y))

$names
[1] "statistic"   "parameter"   "p.value"     "conf.int"
[5] "estimate"    "null.value"  "alternative" "method"
[9] "data.name"

$class
[1] "htest"
```

```
t.test(y)$conf.int

[1] -0.98335647 -0.03847978
attr(,"conf.level")
[1] 0.95
```

4 Basic concepts of hypothesis testing

4.1 Example: maximum temperature

We present a simple hypothesis testing procedure on the mean of a Gaussian population, by considering a data set with the maximum temperature (ºC) registered in 1981 at $n = 25$ weather stations in Portugal. The mean, the median, the standard deviation, and the standard error of the mean are computed. The `qt` function is used in order to obtain quantiles or critical values for a Student's t distribution with 24 degrees of freedom. In particular, the 0.975-quantile corresponds to the 0.025-critical value and it is obtained with the option `lower.tail=FALSE`, which specifies the right tail of the distribution (the default value is `TRUE`, giving the left tail area).

```
t81 <- c(39,39,40,33,36,40,37,41,39,34,42,41,
         42,44,42,42,39,42,41,40,43,43,40,39,37)
mean(t81)

[1] 39.8

median(t81)

[1] 40

sd(t81)

[1] 2.738613

sem <- sd(t81)/sqrt(length(t81))
sem

[1] 0.5477226

qt(0.025,24,lower.tail = FALSE)

[1] 2.063899
```

With these data it is possible to compute the 95% confidence interval for the mean.

```
c(mean(t81)-qt(0.025,24,lower.tail = FALSE)*sem,
  mean(t81)+qt(0.025,24,lower.tail = FALSE)*sem)

[1] 38.66956 40.93044
```

Since a "typical" year has an average maximum temperature of 37.5 °C, we may perform a two-sided t test with null hypothesis $H_0 : \mu = 37.5$ and alternative hypothesis $H_1 : \mu \neq 37.5$. If we assume a significance level $\alpha = 0.05$, the rejection region is $R_{0.05} = \{y : |t| \geq 2.064\}$. Then, since the observed value for the test statistic is

```
(mean(t81)-37.5)/sem

[1] 4.199206
```

the null hypothesis is rejected at the level $\alpha = 0.05$ of significance. The same result can be obtained using the function `t.test`, where the argument `mu` is used to specify the null hypothesis.

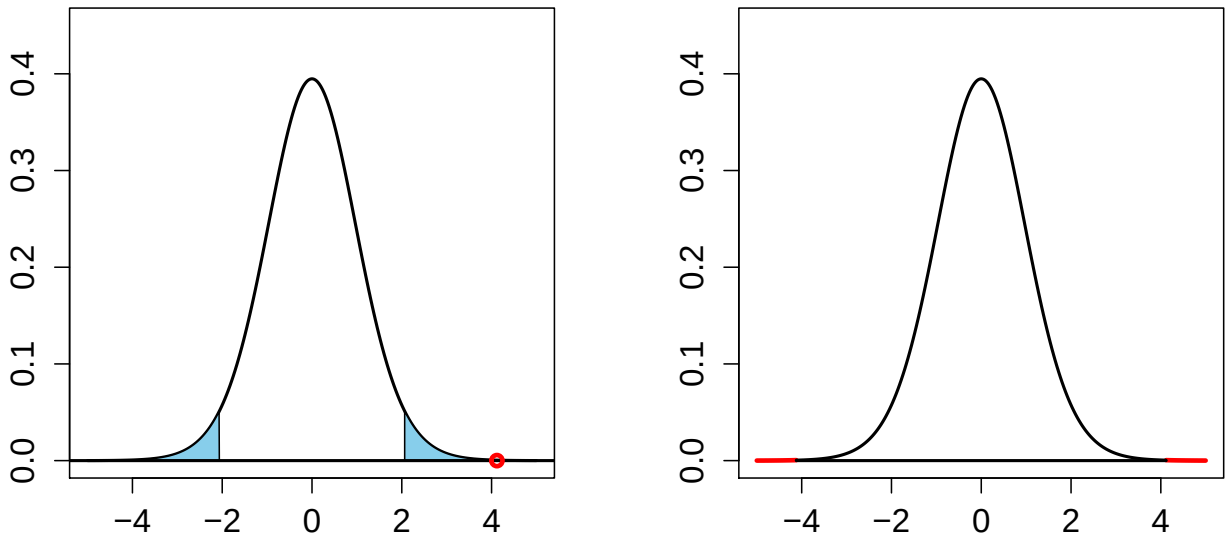
```
t.test(t81,mu=37.5)


One Sample t-test

data:  t81
t = 4.1992, df = 24, p-value = 0.0003181
alternative hypothesis: true mean is not equal to 37.5
95 percent confidence interval:
 38.66956 40.93044
sample estimates:
mean of x
     39.8
```

The above command gives a number of results and, in particular, the p-value turns out to be lower than 0.05, leading to the rejection of H_0. Notice that the confidence interval does not include 37.5, the parameter value specified under the null hypothesis, and this corresponds, once again, to the rejection of the null hypothesis. It is possible to perform one-sided tests using the option `alternative` with values `"less"` or `"greater"` (the default option is `alternative="two.sided"`).

The following lines of code give a graphical representations of the results. The first plot describes the rejection region in blue, and then the limits of the acceptance region. The red circle identifies the observed value of the test statistic. The second plot shows in red the tails area giving the value of the p-value. We apply the `polygon` function, which may be used to define an irregular polygon, as that one approximating the area underlying the density curve, which can also be filled using the available colors. The base arguments of the `polygon` function are very similar to those of the function `plot`. Moreover, the argument `border` can be used to specify the color of the border, the

argument `density` sets the density of the shading lines, the argument `angle` gives the angle of the shading lines and the argument `fillOddEven` corresponds to a logical value controlling the shading mode.

```
par(mfrow=c(1,2))
xx<-seq(-5,5,0.01)
plot(xx,dt(xx,24),type='l',lwd=2,cex.axis=1.3,
     ylim=c(0,0.45),xlab=" ",ylab=" ")
cord.x <- c(-5,seq(-5,-2.064,0.01),-2.064)
cord.y <- c(0,dt(seq(-5,-2.064,0.01),24),0)
polygon(cord.x,cord.y,col='skyblue')
cord.x <- c(2.064,seq(2.064,5,0.01),2.064)
cord.y <- c(0,dt(seq(2.064,5,0.01),24),0)
polygon(cord.x,cord.y,col='skyblue')
abline(0,0,lwd=2)
lines(4.119,0,type='p',lwd=3,col='red')
xx1<-seq(-5,-4.119,0.01)
xx2<-seq(4.119,5,0.01)
plot(xx1,dt(xx1,24),type='l',lwd=3,cex.axis=1.3,
     xlim=c(-5,5),ylim=c(0,0.45),xlab=" ",ylab=" ",col='red')
lines(xx2,dt(xx2,24),lwd=3,col='red')
yy <- seq(-4.119,4.119,0.01)
lines(yy,dt(yy,24),type='l',lwd=2)
lines(c(-4.119,4.119),c(0,0),lwd=2)
```



```
par(mfrow=c(1,1))
```

4.2 Example: physical activity

We consider an hypothesis testing procedure for a proportion (or probability), which can be interpreted as the mean of a Bernoulli distribution. The available data corresponds to a sample of 200 adults, such that 108 of them meet the US guidelines for aerobic physical activity. We want to know if this observed proportion is in accordance with the national target value of 49.2%. Thus the two hypotheses are $H_0 : p = 0.492$ and $H_1 : p \neq 0.492$.

The following commands give the observed proportion, the estimated standard error, the observed value of the z-test statistic, which is approximately Gaussian distributed, and its squared value, and the p-value, leading to the conclusion the the null hypothesis should not be rejected.

```
p <- 108/200
p

[1] 0.54

se <- sqrt(p*(1-p)/200)
se

[1] 0.03524202

z <- (p-0.492)/sqrt(0.492*(1-0.492)/200)
z

[1] 1.357819

z^2

[1] 1.843672

2*pnorm(-abs(z))

[1] 0.1745212
```

In order to develop a test for a proportion we can also use the specific function `prop.test`. This function is based on a test statistics which corresponds to the squared z-statistic and it follows, approximately, a $\chi(1)$ distribution. We may specify the following arguments:

- `x`, which is the vector of counts of successes (it can also be specified as the matrix with two columns including successes and failures, respectively);

- `n`, which is a vector of counts of trials (not necessary if `x` is a matrix);
- `p`, which gives a vector of probabilities of success, and specifies the null hypothesis;
- `alternative`, which specifies the alternative hypothesis (with the same options considered previously);
- `conf.level`, which sets the level of the confidence interval for the proportion (the default value is 0.95);
- `correct`, which is a logical argument indicating whether the Yates' continuity correction should be applied (the default is `TRUE`).

```
prop.test(108, 200, p = 0.492,correct = FALSE)


1-sample proportions test without continuity correction

data:  108 out of 200, null probability 0.492
X-squared = 1.8437, df = 1, p-value = 0.1745
alternative hypothesis: true p is not equal to 0.492
95 percent confidence interval:
 0.4708229 0.6076695
sample estimates:
   p
0.54
```

It is interesting to enumerate the elements given by the outcome of `prop.test`.

```
attributes(prop.test(108, 200, p = 0.492,
                     correct = FALSE))

$names
[1] "statistic"   "parameter"   "p.value"     "estimate"
[5] "null.value"  "conf.int"    "alternative" "method"
[9] "data.name"

$class
[1] "htest"
```

All these objects can be recalled singularly. For instance, we can recall the value of the test statistic. It is not necessary to specify the entire name of the element, since the abbreviation is unique. The value is equal to that one of the squared z-statistic considered before.

```
prop.test(108, 200, p = 0.492,correct = FALSE)$stat

X-squared
 1.843672
```

4.3 Example: white and red wines

We consider two data sets giving the aspartame content (mg/l) in two independent samples of white and red wines. Theses observations can be interpreted as independent realizations of two normal distributions. Some elementary statistical summaries can be calculated.

```
xw <- c(28.4,32.2,37.0,32.4,33.2,18.7,33.7,50.0,49.8,34.5,45.8,33.1,
        24.1,31.0,24.8,19.0,17.5,19.4,24.7,9.9,29.1,18.4,34.7,29.3,
        15.6,20.7,22.2,18.7,11.8,12.1)
xr <- c(7.3,27.9,20.4,18.5,6.6,9.1,1.5,13.9,11.1,34.7,57.0,1.3,17.6,
        6.1,22.9,27.3,30.0,19.6,21.8,18.2,8.6,12.8,18.6,29.4,28.5,
        16.6,30.1,27.2,19.6,16.3,29.9,26.3,26.5,24.3,19.1,28.3,36.8)
mean(xw)

[1] 27.06

median(xw)

[1] 26.6

mean(xr)

[1] 20.85676

median(xr)

[1] 19.6

var(xw)

[1] 110.4328

var(xr)

[1] 120.3359

sd(xw)

[1] 10.5087

sd(xr)

[1] 10.96977
```

The above calculations give evidence to the fact that the two sample means are different and that the two sample variances are quite similar. However, a formal test of hypothesis is required in order to confirm these empirical suggestions.

We first consider a test for the equality of variances (homoscedasticity). In R we can use the specific function `var.test` (or the function `bartlett.test`), but as usual we begin by computing explicitly the F test statistic for the equality of variances, under the null hypothesis, and the associated p-value for a two-sided alternative hypothesis.

```
var(xw)

[1] 110.4328

var(xr)

[1] 120.3359

F <- var(xw)/var(xr)
F

[1] 0.9177051

2*min(pf(F,length(xw)-1,length(xr)-1),
      pf(F,length(xw)-1,length(xr)-1,lower.tail = FALSE)) # p-value

[1] 0.8193624
```

The p-value is substantial, so that the null hypothesis of equal variances is not rejected. The same conclusion is confirmed by using function `var.test`, with the option `ratio = 1` which specifies the null hypothesis.

```
var.test(xw, xr, ratio = 1)


F test to compare two variances

data:  xw and xr
F = 0.91771, num df = 29, denom df = 36, p-value = 0.8194
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4599788 1.8808573
sample estimates:
ratio of variances
         0.9177051
```

In order to compare the two means, a two-sample t-test can be performed. The variances can be treated as equal (this option is supported by the previous analysis) or as unequal; in this last case, we in fact apply the Welch test. In order to calculate explicitly the two test statistics, we consider the following steps:

- the variances of the two sample means are estimated;

```
sem2x <-var(xw)/length(xw)
sem2x

[1] 3.681094

sem2y <-var(xr)/length(xr)
sem2y

[1] 3.25232
```

- the pooled variance estimation is obtained as the weighted mean of the two sample variances;

```
s2p <- (var(xw)*(length(xw)-1)+var(xr)*(length(xr)-1))/
  (length(xw)+length(xr)-2)
s2p

[1] 115.9176
```

- the standard error of the mean difference is computed by assuming equal variances (`sedt`) and unequal variances (`sedw`);

```
sedt <- sqrt(s2p)*sqrt(1/length(xw)+1/length(xr))
sedt

[1] 2.645152

sedw <- sqrt(sem2x+sem2y)
sedw

[1] 2.633138
```

- the two-sample t-test statistic, under the null hypothesis, is obtained under both the assumptions of equal and unequal variances.

```r
tt <- (mean(xw)-mean(xr))/sedt # equal variances
tt
```

```
[1] 2.345137
```

```r
tw <- (mean(xw)-mean(xr))/sedw # unequal variances
tw
```

```
[1] 2.355837
```

The same results can be obtained using the specific `t.test` function, with the options `var.equal=T` and `var.equal=F` (which is the default option). Notice that, in the case of two samples, both the data sets are considered as arguments of `t.test`. There is a moderate evidence against the assumption of equal means.

```r
t.test(xw,xr,var.equal = TRUE)  # equal variances
```

```
Two Sample t-test

data:  xw and xr
t = 2.3451, df = 65, p-value = 0.02208
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.9205107 11.4859758
sample estimates:
mean of x mean of y
 27.06000  20.85676
```

```r
t.test(xw,xr,var.equal = FALSE)  # unequal variances
```

```
Welch Two Sample t-test

data:  xw and xr
t = 2.3558, df = 63.163, p-value = 0.0216
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.9416034 11.4648831
sample estimates:
mean of x mean of y
 27.06000  20.85676
```

4.4 Example: temperatures

The present example is considered in order to describe the two-sample t-test for paired data. We have two data sets with the maximum temperature (°C) registered in 1980 and in 1981 at 25 weather stations in Portugal. In this case, the assumption of independence between the samples is obviously not valid. Then, the mean comparison has to be performed by specifying an alternative procedure based on the difference of the paired data.

The vectors with the two series of temperatures are created (they have the same length) and some summary statistics are applied.

```
t80 <- c(36,35,36,34,37,40,37,41,38,32,36,39,36,40,37,37,38,40,37,39,
         39,41,38,38,35)
t81 <- c(39,39,40,33,36,40,37,41,39,34,42,41,42,44,42,42,39,42,41,40,
         43,43,40,39,37)
summary(t80)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  32.00   36.00   37.00   37.44   39.00   41.00

summary(t81)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   33.0    39.0    40.0    39.8    42.0    44.0
```

In order to test the null hypothesis of equal means, in this particular situation, we consider the following steps:

- a new vector with the differences between the 1980 and the 1981 temperatures is defined and some summary statistics are computed;

```
diff <- t80-t81
summary(diff)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -6.00   -4.00   -2.00   -2.36   -1.00    1.00

sd(diff)/sqrt(length(diff)) # the standard error of the difference

[1] 0.4118252
```

- the paired t-test statistic, under the null hypothesis, is derived.

```
mean(diff)/(sd(diff)/sqrt(length(diff)))

[1] -5.730587
```

The same result is obtained by considering the `t.test` function with the option `pair=TRUE` and, in particular, the null hypothesis of equal means is rejected.

```
t.test(t80,t81,pair=TRUE)


Paired t-test

data:  t80 and t81
t = -5.7306, df = 24, p-value = 6.632e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.209965 -1.510035
sample estimates:
mean of the differences
                  -2.36
```

4.4.1 Non-parametric testing procedures

We shall discuss two examples with the aim of presenting some particular non-parametric testing procedures based on ranks. The focus is on the the two-sample Wilcoxon signed-rank test and on the corresponding R function `wilcox.test`, which is very similar to the `t.test` function.

We first consider the red and white wines example, but now we suppose that the normality assumption is not plausible. We are interest in testing whether the aspartame content is higher in the white wines than in the red wines. The Wilcoxon signed-rank test can be used to this aim, with the advice that the test is on the median values instead on the mean values.

```
wilcox.test(xw,xr)

Warning in wilcox.test.default(xw, xr):  cannot compute exact p-value with ties


Wilcoxon rank sum test with continuity correction

data:  xw and xr
W = 738.5, p-value = 0.02103
alternative hypothesis: true location shift is not equal to 0
```

There is a moderate evidence against the null hypothesis of equality of the median values.

As a second example, we consider the data sets on the maximum temperature (ºC) registered in 1980 and in 1981 in Portugal. If the normal distribution assumption is not valid (as for instance when very large outliers are observed) we can use the Wilcoxon signed-rank test in the paired samples version (the option `paired=T` has to be specified). In this case the null hypothesis of equal median values is rejected.

```
wilcox.test(t80,t81, paired=T)

Warning in wilcox.test.default(t80, t81, paired = T): cannot compute exact p-value
with ties
Warning in wilcox.test.default(t80, t81, paired = T): cannot compute exact p-value
with zeroes


Wilcoxon signed rank test with continuity correction

data:  t80 and t81
V = 7, p-value = 0.0001015
alternative hypothesis: true location shift is not equal to 0
```

4.5 Example: labor training program

Since a proportion may be viewed as the mean of a Bernoulli distribution, the test on the difference between proportions can be interpreted as a particular case of the test for the mean difference. In particular, in case of large samples, a z-test statistic is considered, taking into account that its null distribution is approximately standard Gaussian.

Two groups of individuals are randomly selected: 297 who had participated in labor training programs and 128 who had not. The aim of the analysis is to test if the proportion of high school dropouts is the same in the two reference populations. We observe 217 and 65 dropouts in the two samples, respectively. The null hypothesis is that the proportions of successes is equal in the two populations.

The testing procedure is based on the following steps:

- compute the success proportions in the two groups and the success proportion under the null hypothesis (that is, using all the data without distinguishing the two groups);

```
px <- 217/297
px

[1] 0.7306397

py <- 65/128
py

[1] 0.5078125

p <- (217+65)/(297+128)
p

[1] 0.6635294
```

- determine the standard error of the difference proportion, under the null hypothesis (namely, using the global success proportion);

```
sed <- sqrt(p*(1-p)*(1/297+1/128))
sed

[1] 0.04995913
```

- calculate the observed value of the z-test statistic, under the null hypothesis;

```
z <- (px-py)/sed
z

[1] 4.46019
```

- compute the (approximate) p-value, using the standard normal approximation.

```
2*pnorm(abs(z),lower.tail = FALSE)

[1] 8.188701e-06
```

A similar result can be obtained by means of a chi-square testing procedure, since in this case the observed value of the test statistic corresponds to the squared value of the z-test statistic. Then, we use the function `prop.test`, where the first two arguments correspond to the vector of counts of successes and to the vector of counts of trials, respectively. The p-values of the two testing procedures are equal.

```
z^2

[1] 19.8933

prop.test(c(217,65),c(297,128), correct = FALSE)


2-sample test for equality of proportions without continuity
correction

data:  c(217, 65) out of c(297, 128)
X-squared = 19.893, df = 1, p-value = 8.189e-06
alternative hypothesis: two.sided
95 percent confidence interval:
 0.1225948 0.3230596
sample estimates:
   prop 1    prop 2
0.7306397 0.5078125
```

Furthermore, the testing procedure can be developed by considering the `chisq.test` function. The data must be organized in matrix form (namely, as a contingency table with respect to the dichotomous variables related to the training program and the school dropout). In order to obtain the same result as before, we have to specify the option `correct=FALSE`.

```
X <- t(matrix(c(217,(297-217), 65,(128-65)),2,2))
colnames(X) <- c("Drop_yes","Drop_no")
row.names(X) <- c("Prog_yes","Prog_no")
X

         Drop_yes Drop_no
Prog_yes      217      80
Prog_no        65      63

chisq.test(X, correct=FALSE)


Pearson's Chi-squared test

data:  X
X-squared = 19.893, df = 1, p-value = 8.189e-06
```

4.6 Testing for correlation

The linear relationship between two numerical variables can be studied by means of the Pearson correlation index, under the assumption that the random sample derives from a bivariate normal

distribution or, at least, that marginally the two variables are normally distributed. As an alternative, the Spearman correlation index can be considered. The following application is based on a simulated data set.

In order to obtain values for the first variable, we simulate 100 observations from a standard normal distribution. The values for the second variable are simulated introducing the correlation `r`.

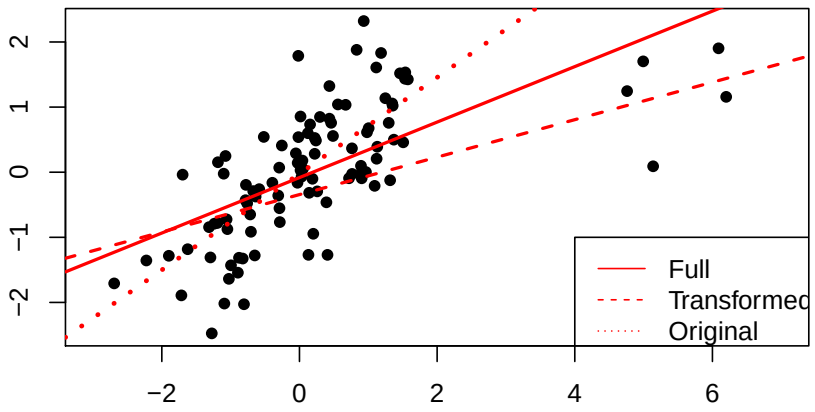
```
set.seed(442)
n <- 100
r <- 0.8
x <- rnorm(100)
y <- r*x + sqrt(1-r^2)*rnorm(n)
```

Then, the 5 largest x values are multiplied by 3, in order to create 5 outliers.

```
ii <- order(-x)
x[ii[1:5]] <- x[ii[1:5]]*3
```

The scatterplot is drawn and three different regression lines are computed: the first is based on the full sample, the second on the sample considering only the 5 transformed observations and the third on the untransformed observations (the subset values are specified using the `subset` option, where the indices of the elements are obtained with the `is.element` command).

```
plot(x,y,xlab=' ',ylab=' ',xlim=c(-3,7),pch=16)
abline(lm(y~x),col=2,lwd=2)
abline(lm(y~x,subset=is.element(1:100,ii[1:5])),col=2,lty=2,lwd=2)
abline(lm(y~x,subset=!is.element(1:100,ii[1:5])),col=2,lty=3,lwd=3)
legend(4,-1, legend=c("Full", "Transformed", "Original"),lty=1:3, col="red")
```



The Pearson and the Spearman correlation indices are computed on the full sample and on the reduced sample with only the original data, and without the outliers.

```
cor(x,y,method='pearson')

[1] 0.6435266

cor(x,y,method='spearman')

[1] 0.7462946

cor(x[-ii[1:5]],y[-ii[1:5]],method='pearson')

[1] 0.7262847

cor(x[-ii[1:5]],y[-ii[1:5]],method='spearman')

[1] 0.7302352
```

It is quite evident that the effect of the outliers is higher on the Pearson correlation index.

A correlation test can be considered for testing the null hypothesis that the correlation index is equal to zero. We use the `cor.test` function and the test based on the Pearson correlation coefficient can be obtained as follows, considering both the full data and the reduced data without the outliers.

```
cor.test(x,y,method='pearson')


Pearson's product-moment correlation

data:  x and y
t = 8.323, df = 98, p-value = 5.165e-13
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5118002 0.7456894
sample estimates:
      cor
0.6435266

cor.test(x[-ii[1:5]],y[-ii[1:5]],method='pearson')


Pearson's product-moment correlation

data:  x[-ii[1:5]] and y[-ii[1:5]]
```

```
t = 10.189, df = 93, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6147235 0.8093561
sample estimates:
      cor
0.7262847
```

With the option `method='spearman'`, one can obtain the same result by considering the non-parametric version of the test based on the Spearman correlation coefficient.

```
cor.test(x,y,method='spearman')


Spearman's rank correlation rho

data:  x and y
S = 42280, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7462946

cor.test(x[-ii[1:5]],y[-ii[1:5]],method='spearman')


Spearman's rank correlation rho

data:  x[-ii[1:5]] and y[-ii[1:5]]
S = 38544, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7302352
```

The `cor.test` function allows the specification of the following main options:

- `alternative`, which defines the alternative hypothesis (`"two-sided"`, `"less"` and `"greater"`);
- `method`, which indicates the Pearson (`'pearson'`) and the Spearman (`'spearman'`) indices and also the Kendall-Tau index (`'kendal'`);
- `exact`, which is a logical option indicating whether an exact p-value should be computed;
- `continuity`, which is a logical option for the continuity correction;
- `conf.level`, which gives the level for the confidence interval (only for `method='pearson'`).

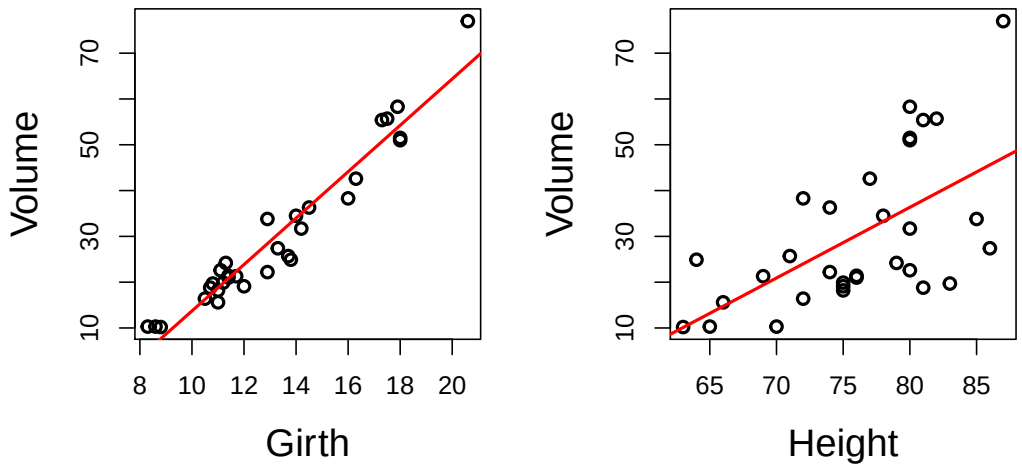
5 Basic concepts of model selection

5.1 Example: black cherry trees

The aim of model selection and model cheeking procedures is to verify whether a statistical model could be consistent with the available data and, eventually, select the best model among a number of possible models. In order to introduce a simple model selection problem, focusing on alternative linear regression models, the data set `trees` of the basic R distribution is considered. The data frame contains the measurements of the `Girth` (diameter of the tree, in inches, measured at a fixed distance above the ground), the `Height` (ft) and the `Volume` of timber (cubic ft) in $n = 31$ felled black cherry trees.

The first step in the statistical analysis is to represent graphically the relationships between `Volume` and `Girth` and between `Volume` and `Height`, using the function `lm` which, in this case, estimates the linear regression models `mod1` and `mod11` and returns the estimated regression lines.

```
par(mfrow=c(1,2))
mod1 <- lm(Volume ~ Girth, data = trees)
mod11 <- lm(Volume ~ Height, data = trees)
plot(Volume ~ Girth, data = trees,lwd=2,cex.lab=1.5)
abline(mod1,lwd=2,col='red')
plot(Volume ~ Height, data = trees,lwd=2,cex.lab=1.5)
abline(mod11,lwd=2,col='red')
```




```
par(mfrow=c(1,1))
```

A further, more general linear regression model (`mod2`) is specified, where both `Girth` and `Height` aim at explaining the response variable `Volume`.

```
mod2 <- lm(Volume ~ Girth + Height, data = trees)
```

The simple model, including only `Girth` as esplicative variable, and the full model are compared by considering the values of the associated log-likelihood and of two information criteria, namely the AIC and the BIC.

```
logLik(mod1)

'log Lik.' -87.82236 (df=3)

AIC(mod1)

[1] 181.6447

BIC(mod1)

[1] 185.9467

logLik(mod2)

'log Lik.' -84.45499 (df=4)

AIC(mod2)

[1] 176.91

BIC(mod2)

[1] 182.6459
```

The best model is clearly the second one, since it achieves the lowest value for the AIC and the BIC; moreover, also the log-likelihood function presents the highest value for the full model. In order to compute these summary statistics we used three specific R functions. Notice that the value for the BIC can also be obtained using the function `AIC`, with the option `k=log(length(trees$Volume))`, which specifies the penalty term (the default value is `k=2`, which gives the classical AIC).

```
AIC(mod2, k=log(length(trees$Volume)))

[1] 182.6459
```

The two models are also compared by considering the so-called (simple) cross-validation strategy. The idea is to estimate the models excluding one observation at a time (step 1) and then use the obtained estimates to compute a fitted value for the excluded observation. The mean value and the standard deviation of these estimated models are computed (step 2) and the associated estimated log-density, evaluated at the omitted observation, is obtained (step 3). Finally, the cross-validation statistic is update. The procedure is replicated for all the observations in the data set.

```
# Initialise the CV index
cv1 <- 0
cv2 <- 0
n <- length(trees$Volume)
i <-1
for (i in 1:n){
# step 1
mod1i <- lm(Volume ~ Girth, data = trees[-i,])
mod2i <- lm(Volume ~ Girth + Height, data = trees[-i,])
# step 2
mu1 <- mod1i$coefficients[1] + mod1i$coefficients[2]*trees$Girth[i]
mu2 <- mod2i$coefficients[1] + mod2i$coefficients[2]*trees$Girth[i] +
           mod2i$coefficients[3]*trees$Height[i]
sd1 <- sqrt(sum(mod1i$residuals^2)/(n-3))
sd2 <- sqrt(sum(mod2i$residuals^2)/(n-4))
# step 3
cv1 <- cv1 - log(dnorm(trees$Volume[i],mu1,sd1))
cv2 <- cv2 - log(dnorm(trees$Volume[i],mu2,sd2))
}
cv1

[1] 92.35513

cv2

[1] 90.62264
```

6 Contingency tables

Contingency tables (two-way tables) display the observed frequencies associated to bivariate sample data and they can be used to study the association between two qualitative variables (factors).

6.1 Example: steel rods

Four machines produce steel rods, whose diameter can be not defective, too short or too long. A sample of $n = 500$ steel rods is randomly selected and the two categorical variables, type of machine and rod diameter, are observed. The aim is to study the association (dependence) between the two variables and the statistical analysis can be developed by considering the following steps.

- The contingency table is saved in a 4×3 matrix.

```
rods <- matrix(c(10, 102, 8, 34, 161, 5,
                 12, 79, 9, 10, 60, 10),nrow=4,byrow=TRUE)
rods
```

```
     [,1] [,2] [,3]
[1,]   10  102    8
[2,]   34  161    5
[3,]   12   79    9
[4,]   10   60   10
```

- The rows and columns totals are saved into two numerical vectors. In order to obtain the row and column totals the `apply` function is used. The second argument of the function sets the margin of the table on which the function `sum` is applied.

```
xtot <- apply(rods,1,sum)
ytot <- apply(rods,2,sum)
```

- The vectors are transformed into matrices in order to obtain their cross-product.

```
xtot <- as.matrix(xtot)
ytot <- as.matrix(ytot)
```

- The cross-product of the row and column totals is computed to define the expected values of the joint absolute frequencies under the hypothesis of independence (no association).

```
rods_ind <- xtot%*%t(ytot)/sum(xtot)
rods_ind

      [,1]   [,2]  [,3]
[1,] 15.84  96.48  7.68
[2,] 26.40 160.80 12.80
[3,] 13.20  80.40  6.40
[4,] 10.56  64.32  5.12
```

- The observed and the expected values of the frequencies are combined in order to obtain the χ^2 test statistic, which measures the intensity of the potential association (an observed value closed to 0 corresponds to low or no association).

```
chisq_obsstat <- sum((rods-rods_ind)^2/rods_ind)
chisq_obsstat

[1] 15.58435
```

- The p-value is computed as 1 minus the probability function computed at the observed value of the test statistic, where 6 indicates the degrees of freedom.

```
1-pchisq(chisq_obsstat, 6)


[1] 0.0161676
```

The χ^2 test can be obtained also using the `chisq.test` function. The results are the same and the low value for the p-value gives a moderate evidence against the independence hypothesis. Using the function `residuals` we obtain the values for the observed standardizd residuals.

```
chisq.test(rods)


Pearson's Chi-squared test

data:  rods
X-squared = 15.584, df = 6, p-value = 0.01617

residuals(chisq.test(rods))

           [,1]        [,2]       [,3]
[1,] -1.4673552  0.56197944  0.1154701
[2,]  1.4791480  0.01577201 -2.1801663
[3,] -0.3302891 -0.15613491  1.0277402
[4,] -0.1723281 -0.53865504  2.1566757
```

6.2 Example: labor training program

We consider data on high school dropout, with regard to two independent samples of, respectively, 128 individuals who had participated in labor training programs and 297 individuals who had not. The observed absolute frequencies, related to the two outcomes (`yes` and `no`) of the variable `high school graduate`, are summarized in the columns of a 2×2 contingency table. The rows correspond to the two samples and, in this case, the totals are assumed to be fixed.

The present situation is different from the previous one, since we aim at comparing two distinct bernoulli populations (in general, multinomial populations) with respect to the interest variable `high school graduate`. In this case the χ^2 test can be considered as-well and it is equivalent to the test assessing whether the "success" probabilities are reasonably the same in the two samples. The statistical analysis is based on the following steps.

- As in the previous example, the contingency table is created.

```
dropout <- matrix(c(63, 65, 80, 217), nrow=2, byrow=TRUE)
colnames(dropout) <- c("yes","no")
rownames(dropout) <- c("yes","no")
names(dimnames(dropout)) <- c("program","high school graduate")
dropout
```

```
       high school graduate
program yes  no
    yes  63  65
    no   80 217
```

- The row and column totals are computed and saved into matrices.

```
xtot <- apply(dropout,1,sum)
ytot <- apply(dropout,2,sum)
xtot <- as.matrix(xtot)
ytot <- as.matrix(ytot)
```

- The expected values of the absolute frequencies under the hypothesis of independence (namely, the proportion of high school graduation is the same in the two populations) are computed and compared with the observed values.

```
xtot%*%t(ytot)/sum(xtot)
```

```
         yes        no
yes 43.06824  84.93176
no  99.93176 197.06824
```

- Finally, the `chisq.test` is considered and, since the p-value is extremely low, a substantial evidence is achieved against the null hypothesis of independence. This conclusion is in accordance with that obtained using the z-test for testing the difference between two proportions (in fact, the value of the χ^2 statistic is the square of that of the z statistic), thought it does not reflect the sign of the difference.

```
chisq.test(dropout,correct =FALSE)
```

```
Pearson's Chi-squared test

data:  dropout
X-squared = 19.893, df = 1, p-value = 8.189e-06
```

Some arguments of the function `chisq.test` are specific for 2×2 contingency tables. In particular, the logical argument `correct`, which specifies whether the continuity correction is considered, the logical argument `simulate.p.value`, which indicates whether to compute the p-value by Monte Carlo simulation, and the option `B`, which gives the number of replications to be used in Monte Carlo procedure.

Whenever the multinomial samples are not independent, we have to consider a different test of hypothesis. In case of two dependent bernoulli samples, the test for large samples is the McNemar's test.