Background in Statistics and R

Benjamin Yakir

November 2, 2008

1 General Introduction

The goal of this course is to explore a selected collection of statistical inferential tools. The common feature of all tools is their dependence on the availability of computers for implementation. Other than that common trait, the procedures frequently differ from each other in the context in which they may be implemented, in the goals of the the inference, in the interpretation of the outcomes, or in any other aspect.

We would initiate the discussion of each procedure by providing a brief motivation, followed by a description of of the actual implementation of the procedure, typical situations in which one may apply the procedure, and the interpretation of outcomes. We would attempt to investigate the statistical properties of the procedure and compare its performance, when possible, to those of more classical ones. For the comparison we may apply theoretical probabilistic computations or use computerized simulations.

We would follow the basic paradigm of mathematical statistics which separates between the implementation of statistical tool to a specific dataset and the discussion of the statistical properties of the tools which carried out in the context of all datasets that *could have emerged*, and not only the one observed. (For example, in statistical testing the procedure can be: "Compute the standardized mean and reject the null hypothesis if the outcome is larger than 1.645." When carried out for the given experiment the decision may be to reject or not. When considered over all possible outcomes one may talk about the the significance level, the power function, etc.)

The investigation of the probabilistic properties of a procedure may, and will be conducted via simulations. This, in principle, will be carried out by simulating independent copies of datasets, applying the procedure to each copy, and the investigation of the resulting distribution of the outcomes of the procedure. Hence, for example, if the sample size is 100, we may simulate 100 observations, compute the standardized mean of the simulated sample and store it. The simulation may be iterated a large number of times, say 100,000 times, and result in 100,000 numbers. If the simulation was carried out under the conditions of the null distribution then one may compute the relative frequency of the numbers that larger than 1.645 in order to obtain (an approximation) of the significance level. If the simulation was done under given values of the alternative parameters one may obtain the power for those given values.

It should be noted that some procedures, the bootstrap for example, may use simulation as part of their implementation on a given dataset. Still, this simulation is not related to the investigation of the properties of the statistic. Hence, if one is

interested in properties such as power or MSE one still needs to produce independent copies datasets as above and apply the bootstrap procedure to each, including the simulation step which is part of the procedure.

Occasionally, in order to save time, we may simulate directly sufficient statistics, using their actual or approximated distribution. Still, the basic paradigm holds.

Notice that when we simulate datasets we do some under given conditions and for given parameter values. This knowledge is not available for the *applied statistician* in the field, who gets a dataset and is required to make inference on it. Hence, the inference tool may not assume knowledge of parameter values. However, we as *theoretical statistician*, may ask the question: "What would have been the statistical properties of the procedure if the state of nature is such and such?". Thus, for example, if we want to investigate an estimator that is used by the applied statistician we may carry out the simulations for parameter values of our choice, "hand" the simulated dataset to the practical statistician and see, on the average, how closely the estimator she compute hits the target unknown to her.

2 Introduction to R

R is a freely distributed software for data analysis. In order to introduce R let us quote the first paragraphs from the manual *Introduction to R* by W. N. Venables, D. M. Smith and the R Development Core Team. (The full document, as well as access to the installation of the software itself, are available online at *http://cran.r-project.org/*):

> "R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has
>
> - an effective data handling and storage facility,
> - a suite of operators for calculations on arrays, in particular matrices,
> - a large, coherent, integrated collection of intermediate tools for data analysis,
> - graphical facilities for data analysis and display either directly at the computer or on hardcopy, and
> - a well developed, simple and effective programming language (called S) which includes conditionals, loops, user defined recursive functions and input and output facilities. (Indeed most of the system supplied functions are themselves written in the S language.)
>
> The term environment is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.
>
> R is very much a vehicle for newly developing methods of interactive data analysis. It has developed rapidly, and has been extended by a large collection of packages. However, most programs written in R are essentially ephemeral, written for a single piece of data analysis."

R may be obtained as a source code or installed using a pre-compiled code on the Linux, Mackintosh and the Windows operating systems. Programming in R for this book was carried out under Windows. You may find more detailed information regarding the installation of R on the Windows operating system at *http://www.biostat.jhsph.edu/ kbroman/Rintro/Rwin.html*.

After installing R under the `Windows` operating system an icon will be added to the desktop. Double clicking on that icon will open the window of the `R` system, which contains the `R Console` sub-window. We found it convenient to have a separate working directory for each project. It is convenient to copy the `R` icon into that directory and to set the working directory by coping its path (in double quotes) in the appropriate box (`"start in:"`) in the `Shortcuts` slip of the `Properties` of the icon (which can be selected by right-clicking the icon.)

The `R` language is an interactive expression-oriented programming language. The elementary commands may consist of expressions, which are immediately evaluated, printed to the standard output and lost. Alternatively, expressions can be assigned to object, which store the evaluation of the expression. In the later case the result is not printed out to the screen. These objects are accessible for the duration of the session, and are lost at the end of the session, unless they are actively stored. At the end of the session the user is prompted to store the entire workspace image, including all objects that were created during the session. If "`Yes`" is selected then the objects used in the current session will be available in the next. If "`No`" is selected then only objects from the last saved image will remain.

Commands are separated either by a semi-colon (`;`), or by a newline. Consider the following example, which we type into the `R Console` window:

```
> x <- c(1,2,3,4,5,6)
> x
[1] 1 2 3 4 5 6
```

Note that in the first line we created an object named `x` (a vector of length 6, which stores the value $1 \ldots, 6$). In the second line we evaluated the expression `x`, which printed out the actual values stored in `x`. In the formation of the object `x` we have applied the *c*oncatenation function `c`. This function takes inputs and combine them together to form a vector.

Once created, an object can be manipulated in order to create new objects. Different operations and functions can applied to the object. The resulting objects, in turn, can be stored with a new name or with the previous name. In the latter case, the content of the object is replaced by the new content. Continue the example:

```
> x*2
[1]  2  4  6  8 10 12
> x
[1] 1 2 3 4 5 6
> x <- x*2
> x
[1]  2  4  6  8 10 12
```

Observe that the original content of `x` was not changed due to the multiplication by two. The change took place only when we deliberately assigned new values to the object `x`.

Say we want to compute the average of the vector `x`. The function `mean` can be applied to produce:

```
> mean(x)
[1] 7
```

A more complex issue is to compute the average of a subset of `x`, say the values larger than 6. Selection of a sub-vector can be conducted by use of the vector index, which is accessible by the use of square brackets next to the object. Indexing can be implemented in several ways, including the standard indexing of a sequence using integers. An alternative method of indexing, which is natural in many applications, is via a vector with logical `TRUE/FALSE` components. Consider the following example:

```
> x > 6
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE
> x[x > 6]
[1]  8 10 12
> mean(x[x > 6])
[1] 10
```

Observe that the vector `x > 6` is a logical vector of the same length as the vector `x`. Only the components of `x` parallel to the components with a `TRUE` value in the logical indexing vector are selected. In the last line of the example the resulting object is used as the input to the function `mean`, which produces the expected value of 10.

For comparison consider a different example:

```
> x*(x > 6)
[1]  0  0  0  8 10 12
> mean(x*(x >6))
[1] 5
```

In this example we multiplied a vector of integers `x` with a vector of logical values `(x > 6)`. The result was a vector of length 6 with zero components where the logical vector takes the value `FALSE` and the original values of `x` where the logical value takes the value `TRUE`. Two points should be noted. First, observe that R can interpret a product of a vector with integer components and a vector with logical components in a reasonable way. Standard programming languages may have produced error messages in such a circumstance. In this case, R translates the logical vector into a vector with integer values — one for `TRUE` and zero for `FALSE`. The outcome, a product of two vectors with integer components, is a vector of the same type. The second point to make is that multiplication of two vectors is conducted term by term. It is not the inner product between vectors. A different operator is used in R in order to preform inner products.

3 Basic Statistical Models

The binomial model

Assume that the observations can be represented as a sequence of n binary outcomes. The possible outcomes may be denoted a "success" or a "failure". Such a sequence

is termed a sequence of *Bernoulli trials* if the probability of a "success" is the same for all elements in the sequence and if the elements are statistically independent of each other (i.e., the probability of "success" in one trial is not affected by the outcomes in the other trials).

Denote by p be the probability of "success"and let the random variable X denote the total number of successes among the n trials. Then X is said to have a binomial distribution. The probability density function of X is given by:

$$f(x) = \mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, 2, \ldots, n.$$

The short notation $X \sim B(n, p)$ will be used in order to refer to this distribution. It is well known that the expectation of X (i.e., the average value of X, denoted "$\mathbb{E}(X)$") is equal to np and its variance (denoted "var(X)") is equal to $np(1-p)$ (with $\sqrt{np(1-p)}$ the standard deviation of X).

The normal distribution

The normal distribution — also known as the *Gaussian* distribution — is the most popular statistical model. The formula for the density of the normal distribution is given by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}, \quad x \in \Re,$$

which forms the famous bell shape. The parameter μ is the mean, or the location, of the distribution and σ^2 is its variance (σ is the standard deviation or the scale). In particular, when $\mu = 0$ and $\sigma^2 = 1$ the distribution is called the *standard* normal distribution. The density of the standard normal distribution is symbolized by: $\phi(x)$ and the cumulative distribution function (cdf) is symbolized by

$$\Phi(x) = \int_{-\infty}^{x} \phi(z) dz.$$

We denote the fact that X has a normal distribution with mean μ and variance σ^2 by the notation $X \sim N(\mu, \sigma^2)$.

Statistics are quantities computed as functions of the observations. The distribution of a statistic can be quite complex. Surprisingly enough, in many occasions it is the case that the distribution of the statistics resembles the bell shaped distribution of the normal random variable, provided that the sample size is large enough and the statistic is computed as an average or a function of averages. This observation will be stated more formally later in this chapter when we discuss *the Central Limit Theorem* (CLT).

The Poisson distribution

The Poisson distribution is useful in the context of counting the occurrences of rare events. Like the binomial distribution, it takes integer values. Indeed, as we will see later in this chapter, it can arise as an approximation of the binomial distribution when p is small but n is large.

We say that a random variable X has a Poisson distribution with rate λ (denoted $X \sim \text{Poisson}(\lambda)$) if the probability function of X has the form

$$f(x) = e^{-\lambda}\frac{\lambda^x}{x!}, \quad x = 0, 1, 2, \ldots.$$

It follows that the expectation and the variance of X are both equal to λ.

4 Testing hypothesis

Statistics inference is used in order to detect and characterized meaningful signals which may be hidden in a environment contaminated by random noise. Statistical hypothesis testing is a typical step in the process of making inferences. In this step one tries to answer the fundamental question: "Is there a signal at all?". In other words, can the observed data can be reasonably explained by a model for which there is no signal but only noise?

Assuming the statistical model has been selected, we describe the process of testing statistical hypothesis by partitioning it into three formal steps: (i) Formulation of the hypotheses, (ii) specification of the test, and (iii) reaching the final conclusion. The first two steps are carried out based on the statistical model, and in principal can be carried out prior to the collection of the observations. Only the third step involves the analysis of the actual observations.

An example: testing genetic identity of affected siblings

Gene mapping involves an array of strategies for associating between expressed heritable traits and genes — the carrier of the genetic information for the formation of proteins. One of these strategies, denoted the *Affected Sib-Pairs* (ASP) approach, calls for the collection of a large number of nucleus families, each with a pair of affected siblings which share the medical condition under investigation. Here we will consider an artificial scenario where all the sibling pairs of are composed of half-siblings, which share only one parent in common, and concentrate on a given gene, which may or may not be associated with the disease. The aim is to test for the presence of association.

The gene may be embodied in any one of several variant forms, called *alleles*. Moreover, for autosome chromosomes an individual carries two homolog copies of the gene, one inherited from the mother and the other from the father. Therefore, each offspring carries two versions of the given gene, which may not be identical in form. Still, one of the copies is an identical copy of one of the two homolog genes in the common parent while the other is a copy of one of the homolog genes in the other parent. Concentrate on the copies in the siblings which originated from the common parent. Two possibilities emerge: Either, both siblings' copies emerge from a common source or else each was inherited from a different source. In the former case we say that the two copies are identical by decent (IBD) and in the latter case we say that they are not. It is natural to model the IBD status of a given pair as a Bernoulli trial, with and IBD event standing for "success". Counting the number of pairs for which an IBD occurred would produce a binomial random variable.

The standard laws, which govern segregation of genetic material form a parent to an offspring, will produce IBD or not in equal probabilities. This is the expected

probability of success when the gene is not associated with the trait. When it is associated, however, one may expect elevated level of sharing of genetic material within the pair and thus elevated levels of IBD. Denote by p this probability of IBD=1. A natural formulation of the statistical hypothesis is: $H_0 : p = 0.5$ versus $H_1 : p > 0.5$. As a test statistic, one may use the number of pairs with an IBD share, which we denote by X. Alternatively, one may standardize this statistic by subtracting out the expectation and dividing by the standard deviation, both computed under the null $B(n, 0.5)$ distribution, where n is the total number of pairs in the trial. The resulting statistic is:

$$Z_n = \frac{X - n/2}{\sqrt{n/4}}.$$

A standard recommendation is to use a threshold of 1.645. Values of the test statistic above that threshold lead to the rejection of the null hypothesis and to the conclusion that an association is present.

Let us investigate the significance level of the proposed test. Assume that a total of $n = 100$ pairs were collected. Then the results of the test may look like this:

```
> n <- 100
> X <- rbinom(1,n,0.5)
> X
[1] 44
> Z <- (X-n/2)/sqrt(n/4)
> Z
[1] -1.2
> Z > 1.645
[1] FALSE
```

Observe that the number of pairs which share an IBD copy of the gene was 44. This result was generated using the function `rbinom`, which simulates the binomial distribution. The first argument of the function is the number of independent copies to produce; a single copy in our case. The second argument is the number of Bernoulli trials. In this example, the number is `n`, which was assigned a value of 100. The third argument is the probability of success. The statistic `Z` was computed by standardizing the statistic `X`. It obtained in this case the negative value -1.2. Obviously, the null hypothesis is not rejected. Note that the function `rbinom` simulates random occurrences of a binomial random variable. Running the same code again may produce different outcomes.

In order to evaluate the significance level of the test it is not enough to simulate a single trial. Consider the following code:

```
> X <- rbinom(10^6,n,0.5)
> Z <- (X-n/2)/sqrt(n/4)
> mean(Z > 1.645)
[1] 0.044226
```

Observe that the function `rbinom` produces in this case one million independent copies of the binomial distribution, all stored in a vector `X` of that length. Each of the components of the vector `X` is then standardized in the same way as the single

number was in the previous example. The result is the vector `Z`, which contains the standardized values. The last line of code involves an application of the function `mean`, which computes, as we have previously seen, the average value of its input. Note that the input here is a vector with logical `TRUE/FALSE` components. A component takes the value `TRUE` if the null hypothesis is rejected and `FALSE` when it is not. When introduced to the function `mean`, the logical values are translated into numerical values: one for `TRUE` and zero for `FALSE`. As a result, the function `mean` produces the relative frequency of rejecting the null hypothesis, which is an approximation of the significance level. Observe, that the resulting number is 0.044226, which is close, but not identical, to the expected significance of 0.05.

5 Limit theorems

The rational behind the selection of 1.645 as a threshold in the above test is based on the similarity between the standardized binomial distribution and the standard normal distribution. The given threshold is the appropriate threshold in the normal case. This similarity is justified by the Central Limit Theorem (CTL). In this section we will formulate (without proof) the CLT in the context of sums of independent and identically distributed (i.i.d.) random variables. Actually, the scope of the central limit theorems is much wider. It includes multivariate distributions as well as sums of non-identical and weakly-dependent random variables. When rare events are considered, the Poisson distribution may provide a better approximation than the normal. A Poisson limit theorem will be presented here in the context of binomial random variables. Again, generalizations of the basic theorem in various directions do exist.

The central limit theorem states that the distribution of a standardized sum of independent and identically distributed random variables converges to the standard normal distribution. More precisely (recall that Φ is the cdf of the standard normal distribution):

> **Central Limit Theorem:** Let $X_1, X_2, \ldots,$ be a sequence of i.i.d. random variables. Denote the expectation of these random variables by μ and the variance by σ^2, which we assumed to be finite. Consider, for each n, the random variable:
>
> $$Z_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n\sigma^2}} = \frac{\sqrt{n}}{\sigma}[\bar{X}_n - \mu] = \frac{1}{\sqrt{n}} \sum_{i=1}^n \frac{X_i - \mu}{\sigma}.$$
>
> Then, for any $-\infty < x < \infty$,
>
> $$\lim_{n\to\infty} \mathbb{P}(Z_n \leq x) = \Phi(x).$$

As an example of the application of the central limit theorem consider the binomial distribution. Recall that if $X \sim B(n,p)$ then X can be represented as a sum of Bernoulli variables. Moreover, it is easy to see that the expectation of each of these Bernoulli variables is p and the variance is $p(1-p)$. It can be concluded that the distribution of $Z_n = (X - np)/\sqrt{np(1-p)}$ can be approximated by the standard normal distribution. In particular, when n is large,

$$\mathbb{P}(Z_n > 1.645) \approx 1 - \Phi(1.645) = 0.05.$$

The normal approximation works best when the distribution of independent components X_i is closer to being symmetric. This will be the case, for example, in Bernoulli trial when the probability of success is in the central part of the interval $[0, 1]$. The approximation will produce less satisfactory results when the probability of a success is closer to zero or to one. In such scenarios the Poisson approximation will tend to produce better results. We can state the theorem which establishes the Poisson approximation:

> **Poisson Approximation:** Let $X_n \sim B(n, p_n)$ be a sequence of binomial random variables. Assume that the sequence of p_n of success probabilities obeys the relation $np_n \to_{n\to\infty} \lambda$, where $0 < \lambda < \infty$. Then, for any $x = 0, 1, 2, \ldots$,
>
> $$\lim_{n\to\infty} \mathbb{P}(X_n = x) = e^{-\lambda}\frac{\lambda^x}{x!}.$$

Note that the requirement $np_n \to \lambda$ is equivalent to stating that the probability of of success, p_n, converges to zero in a rate which is proportional to the number of Bernoulli observations.

Let us demonstrate both the normal and the Poisson approximation in the binomial setting. The following lines of code will produce the plot in Figure **??**:

```
> n <- 100; p <- 0.5
> X <- rbinom(10^6,n,p)
> Z <- (X-n*p)/sqrt(n*p*(1-p))
> z <- seq(-4,4,by=0.01)
> plot(z,pnorm(z),type="l",col="red")
> lines(ecdf(Z))
> x <- z*sqrt(n*p*(1-p)) + n*p
> lines(z,ppois(x,n*p),type="s",col="blue")
```

The first three lines of code require no explanation. They are essentially identical to the code which was used in order to generate the distribution of the test statistic.

In the forth line we generate a sequence of numbers, ranging between -4 and 4, in jumps of size 0.01. This sequences is generated with the aid of the function `seq`. The first argument to the function is the starting point of the sequence and the second argument is the ending point. The third argument is the jump size, and it is introduced using the name of the argument: `by`. The rule in the introduction of arguments to functions is that arguments may be set either by placing them in the same order in which they appear in the definition of the function, or by using the argument assignment format `par_name = par_value`. If not all preceding arguments are assigned, then the argument must be assign using the argument assignment format.

The subsequent line produces a plot. The function `plot` is a generic function for making plots. In its simplest application it requires as input a sequence of x values and a sequence of y values, both of the same length. It produces the appropriate scatter plot of the points. This basic behavior may be modified by setting arguments. For example, the argument `type` determines the plotting style. Setting its value to `"l"` will result in sequential connecting the points by segments, which will produce a line. Likewise, setting the argument `col` to `"red"` will color the line in the given color. Observe that the y values are produced here by the function `pnorm`. This

function takes as input real values and produces as output the normal cdf at these values. Execution of the code will result in opening of a graphical window within R and the formation of the plot of the normal cdf over the range of `z`.

The function `plot` is considered to be a function of high-level plotting, since it produces a plot independently. Lower-level plotting functions, on the other hand, add features to existing plots. The function `lines` is a low-level function. In its generic usage it adds lines to a plot. In its first appearance in this example it takes as input the output of the function `ecdf`. This function calculates the empirical distribution function from a set of observation – the vector `Z` in this case. The empirical distribution is then added to the plot. Note that the empirical distribution is a step function with jumps, indicated as small circles, in the points where the density has a point-mass. As a matter of fact, the application of the function `lines` in this line is non-generic but is specific to output of the function `ecdf`.

For comparison, we would like to add the cdf function of the Poisson distribution to the plot. The vector `x` is the image of the vector `z` in the original scale of the binomial random variable (the integers between zero and n). The cdf function of the Poisson distribution in the original scale is computed with the aid of the function `ppois`. The second argument to the function is the mean parameter of the Poisson distribution, which we equate with the mean of the binomial distribution. The function `lines` adds this cumulative distribution function to the plot. Like for the function `plot`, the default behavior of the function `lines` can be modified. Here we used the option `type="s"` in order to produce a step function and the option `col="blue"` in order to paint it blue.

6 Correlation and regression

In most scientific experiments not one but several variables are measured. It is of interest to quantify and assess the relationships between variables. A popular summary statistic for the quantification of pairwise relationships is the covariance. An alternative is the correlation, which is its standardized form.

Imagine that we are given a sample of n unrelated individuals, which express some quantitative phenotype of interest. Concentrate on a target autosome gene and assume that the phenotype and the alleles of the gene are measured. This produces a phenotype measurement and a combination of two allele measurements for each subject in the sample. For simplicity, let us suppose that the gene has only two possible alleles, or versions of the gene, one denoted the *wild type* and the other the *mutated type*. Use y_i to symbolize the level of the phenotype for subject i and let x_i be the count of the number of mutated alleles for the same subject. Observe that y_i obtains numerical values and that x_i may obtain the values 0, 1, or 2. The empirical covariance between the phenotypic and genotypic measurement is defined by:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}),$$

where $\bar{y}$ and $\bar{x}$ are the average values of the y's and the x's, respectively. The correlation between the two measurements is obtained by dividing the above by the

product of the standard deviations of the two sequences. It takes the form:

$$\hat{\rho} = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}.$$

The correlation coefficient is a quantification of dependence between two measurements. It may obtain values in the range between -1 and 1. A value of 1 corresponds to an exact linear relation with a positive slope. Likewise, a value of -1 corresponds to an exact linear relation with a negative slope. When the value of the coefficient is equal to zero we say that the two measurements are uncorrelated. Independence between variables implied lack of correlation in the sense that the correlation coefficient will tend to take values closer to zero. (To be exact, the formulation of lack of correlation is typically given in the context of the population, or an infinite sample size. The covariance and the correlation coefficient are computed then by taking expectations, instead of averages. In this setting independence between measurements implies that the correlation coefficient is equal exactly to zero.)

The covariance and correlation parameters are closely related to the popular statistical model of linear regression, which may be used as a model for the relation between the two measurements. Regression models sets one of the variables to be the dependent variable and the other to be the independent, or explanatory, variables. According to the model, the conditional expectation of the dependent variable is a linear function of the explanatory variable. Typically, the model assumes that the residuals are independent of the explanatory variable.

In the genetic example, linear regression is denoted the *additive model*. This model proposes the relation:

$$y_i = \mu + \alpha x_i + e_i.$$

The parameter μ is the expected value of the phenotype among wild-type homozygote and the parameter α is the additive effect of the mutation on the phenotype. The residual e_i is a zero-mean random variable, which is independent of x_i. In this example we will take the distribution of the residuals to be normal, and denote its variance by σ^2.

In the case where $\alpha = 0$ we have that x and y are independent. This can be interpreted as the gene not having an effect on the trait. In order to test for such independence one may formulate the problem in terms of the hypotheses $H_0 : \alpha = 0$ versus $H_1 : \alpha \neq 0$, and look for an appropriate test statistic.

Let $Z = \sqrt{n}\hat{\rho}$ and consider $U = Z^2 = n\hat{\rho}^2$. Consideration, which will be given in the next section and further discussed in the last section, will suggest using U as a test statistic and will propose the chi-square distribution as an approximation of its null distribution. Large values of U, which correspond to values of $\hat{\rho}$ bounded away from zero, are an indication for a non-zero value for the slope parameter α. Consequently, the null hypothesis of independence between the phenotype and the genotype measurements will be rejected when U is larger than the 95%-quintile of the chi-square distribution on one degree of freedom.

Let us examine the statistical properties of this test. Consider a sample of size $n = 100$. We generate the test statistic under the null assumption of independence between y and x:

```
> n <- 100
> U <- numeric(length=10^4)
> for (i in 1:length(U))
+
{
+     x <- rbinom(n,2,0.3)
+     y <- rnorm(n,15,3)
+     U[i] <- n*cor(x,y)^2
+ }
> mean(U > qchisq(0.95,1))
[1] 0.0527
```

Observe the use of the `for` loop. The indexing variable, to the left of the word "`in`" inside the brackets, sequentially obtains as a value the components of the vector to the right of that word. The expression following the brackets is evaluated for each value of the indexing variable. Several expressions may be grouped together by placing them between curly brackets. In this example we compute the test statistic 10,000 times. In each iteration `x` values are generated according to the binomial distribution. The phenotypes are generated according to the normal distribution with mean equals 15 and standard deviation of 3. The test statistic is formed by squaring the correlation coefficient and multiplying the result by the sample size. The result is stored in the vector `U`. The function `qchisq` computes the quintile of the chi-square distribution. The simulated significance level is approximately equal to the target value of 5%.

A more comprehensive description of the distribution of the test statistic under the null distribution may be given in the form of a plot. A plot may be created with the following code: using the code:

```
> u <- seq(0,30,length=100)
> plot(u,pchisq(u,1),type="l",ylab="probability")
> p <- (1:length(U))/length(U)
> U <- sort(U)
> lines(U,p,col=2)
```

The *black line* represents the theoretical cdf function of the chi-square distribution. The *red line* represents the empirical distribution function of the simulated `U` statistic. The *green line* represents the empirical distribution function of the statistic when it is simulated under the alternative hypothesis. Note that the simulated red line and the theoretical black line are practically identical. Observe that we have not used the built-in function `ecdf` for plotting the empirical distribution function. Instead, we wrote our own code, which applies the fact that the empirical distribution function increments in steps of size equal to the reciprocal of the number of observations. The steps occur at the observed values. In order to do the plot we sorted the values of the vector `U` with the function `sort`. The resulting vector serves as x values. The y values is a monotone vector in the range $(0, 1]$ of the same length as `U`, which is stored in an object named `p`.

Under the alternative hypothesis the value of α is different than zero. In the following example we take it to be equal to 1.5. The simulation of `U` under the alternative distribution is carried out exactly like the simulation under the null.

The only difference is in the expectations of the `y` values, which are taken to be a functions of the `x` values. This is implemented by using different `mean` argument in the function `rnorm`. The resulting power is about 0.89. The plot of the empirical distribution of the test statistic under the given alternative is added as a green line to the plot:

```
> for (i in 1:length(U))
+ {
+     x <- rbinom(n,2,0.3)
+     y <- rnorm(n,15+1.5*x,3)
+     U[i] <- n*cor(x,y)^2
+ }
> mean(U > qchisq(0.95,1))
[1] 0.8898
> U <- sort(U)
> lines(U,p,col=3)
```

7 Likelihood-based inference

Tools for making statistical inference can be constructed in a variety of ways. Yet, the practice in the statistical community, which is supported by solid theoretical foundations, is to favor likelihood-based approaches.

The likelihood function is the distribution density function of the observations, interpreted as a function of the parameters that determine the distribution. In many cases it is more convenient to consider the logarithm of the likelihood function, denoted the *log-likelihood* function. Either of these functions may be used in order to identify good estimates of unknown parameters or in order to construct efficient test statistics.

For example, if the observation has a binomial distribution then the log-likelihood function takes the form:

$$\ell(p) = \log \binom{n}{X} + X \log p + (n - X) \log(1 - p).$$

It is considered as a function of p, the probability of a success. The argument X is the observed number of successes in the trial. An estimate of the unknown parameter, as a function of the observation, may be obtained by maximizing the likelihood function with respect to the parameter. The result is the *maximum likelihood estimate* (MLE). Equivalently, the log-likelihood function may be used. The maximizer can be found by equating the derivative of the log-likelihood to zero:

$$\dot{\ell}(p) = \frac{X}{p} - \frac{n - X}{1 - p} = \frac{X - np}{p(1 - p)} = 0.$$

The MLE in this case turns out to be $\hat{p} = X/n$, the empirical frequency of successes in the sample. Note that we have used the 'dot' notation to represent derivatives.

The log-likelihood function and its derivatives are a good source for finding efficient test statistics. The motivation for this proposition is a well known lemma, attributed to Neyman and Pearson, that states that the likelihood ratio is the most powerful statistic for testing a given null distribution against a given alternative

distribution. The likelihood ratio statistic is computed as the ratio of the likelihood at the given alternative divided by the likelihood at the given null value. Alternatively, the difference of the log-likelihoods can be used. The null hypothesis is rejected when the statistic is above a threshold; a threshold which is set by the null distribution of the statistic.

For a composite hypotheses it is not clear which parameter values to use in the log-likelihoods that form the difference. One approach, termed the *generalized likelihood ratio test* (GLRT), is to use maximum likelihood estimates. One of the log-likelihood functions is maximized over the entire space of parameters. This corresponds to plugging the MLE into the function. The other log-likelihood is maximized over the subset of parameters that form the null hypothesis. This correspond to plugging into the function the MLE for a sub-model that corresponds to the null hypothesis. Considerations which are hinted for in the next section can be used in order to show that under suitable regularity conditions the null distribution of twice the log-likelihood difference is approximately chi-square when the sample size is large. The number of degrees of freedom for the chi-square distribution is the difference between the dimension of the space of parameters and the dimension of the sub-space formed by the null hypothesis. For example, the GLRT for the hypothesis $H_0 : p = 0.5$ versus $H_1 : p \neq 0.5$ in the binomial case is:

$$2(\ell(\hat{p}) - \ell(0.5)) = 2n\left[\hat{p}\log\frac{\hat{p}}{0.5} + (1-\hat{p})\log\frac{1-\hat{p}}{0.5}\right].$$

One degree of freedom is assigned to the asymptotic chi-square distribution. This follows from the fact a single parameter is used, which corresponds to dimension one. The null hypothesis contains only one point – a subspace of dimension zero. The difference in the dimensions equals one.

A second approach for testing in a setting of composite hypothesis is termed the *Wald test*. This test measures the distance between the unconstrained maximum likelihood estimate of the parameter and the estimate produced under the null hypothesis. Again, under suitable regularity conditions it can be shown that, for an appropriate definition of distance between estimates, the limiting distribution of the statistic is chi-square. The number of degrees of freedom is, like before, the difference in dimensions. In the binomial case the distance between the estimates becomes $4n(\hat{p} - 0.5)^2$, which is equal to the square of the test statistic which we have been using in the previous sections (for a one-sided, rather than a two-sided, alternative).

The reset of this section will be devoted to the third approach, which comes under the heading of the *Lagrange multiplier test* or *the score statistic*. This test statistic uses the first and second derivatives of the log-likelihood function. In the multi-parameter setting it uses the gradient vector and the hessian matrix of the function, i.e. the vector of partial derivatives and the matrix of mixed partial second derivatives. Give the likelihood function $\ell(\theta)$ we will denote the gradient by $\dot{\ell}(\theta)$ and the hessian by $\ddot{\ell}(\theta)$.

For the construction of the score statistic the maximum likelihood estimated of the parameters under the null assumption is computed. Denote this estimate by $\hat{\theta}_0$. One ingredient in the test statistic is the gradient of the log-likelihood, evaluated at the null MLE $\hat{\theta}_0$: $\dot{\ell}_0 = \dot{\ell}(\hat{\theta}_0)$. A second ingredient is a matrix, which is computed with the aid of the hessian matrix. One possibility is to consider

the *Fisher information* matrix, which is minus the expected value of the hessian. An alternative is to use the *empirical* Fisher information matrix, which is simply negative the computed hessian. Other approaches may use approximations of these two information matrices. The second component in the construction of the score statistic, denoted here by H, is the inverse of the information matrix, evaluated at $\hat{\theta}_0$. The score statistic itself is

$$U = (\dot{\ell}_0)' H (\dot{\ell}_0).$$

Like before, it can be shown that the asymptotic null distribution of this statistic is chi-square with number of degrees of freedom equal to the difference in dimensions.

Let us consider the binomial model for the last time. The second derivative of the log-likelihood function is given by:

$$\ddot{\ell}(p) = -\frac{X}{p^2} - \frac{n-X}{(1-p)^2}.$$

Using $\hat{p}_0 = 0.5$ we obtain:

$$\dot{\ell}_0 = \dot{\ell}(0.5) = 4(X - n/2), \quad H = -1/\mathbb{E}[\ddot{\ell}(0.5)] = 1/(4n).$$

As a result we get:

$$(\dot{\ell}_0)^2 \times H = \frac{16(X-n/2)^2}{4n} = \left(\frac{X-n/2}{\sqrt{n/4}}\right)^2$$

which produces again the square of the statistic that was proposed before.

For a more complex example, let us consider testing for a slope α in the regression model which was introduced in the previous section. Observe that the joint density function of the phenotype and the genotype of a subject, assuming the binomial distribution for the genotype, is given by:

$$f(y,x) = f(y \,|\, x) \cdot f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\mu-\alpha x)^2} \cdot \binom{2}{x} p^x (1-p)^{2-x}.$$

The parameters of this model are μ, α, σ^2, and p. It follows that the log-likelihood function of the entire sample is:

$$\begin{aligned} \ell(\mu, \alpha; \sigma^2, p) &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu - \alpha x_i)^2 \\ &\quad + \sum_{i=1}^{n}\log\binom{2}{x} + \Big(\sum_{i=1}^{n} x_i\Big)\log p + \Big(2n - \sum_{i=1}^{n} x_i\Big)\log(1-p). \end{aligned}$$

For the simplicity of the exposition, let us assume that the parameters σ^2 and p are known. Taking partial derivatives with respect to the unknown parameters produces the gradient:

$$\dot{\ell} = \begin{pmatrix} \dot{\ell}_\mu \\ \dot{\ell}_\alpha \end{pmatrix} = \frac{1}{\sigma^2} \cdot \begin{pmatrix} \sum_{i=1}^{n}(y_i - \mu - \alpha x_i) \\ \sum_{i=1}^{n}(y_i - \mu - \alpha x_i) x_i \end{pmatrix} = \frac{n}{\sigma^2} \cdot \begin{pmatrix} \bar{y} - \mu - \alpha\bar{x} \\ \overline{yx} - \mu\bar{x} - \alpha\overline{xx} \end{pmatrix}.$$

Notice the notation that was adopted. The term $\overline{yx}$ stands for the average of the product of the y and the x values and the term $\overline{xx}$ stands for the average of the square of the x values. The hessian is produced by taking second partial derivatives:

$$\ddot{\ell} = \begin{pmatrix} \ddot{\ell}_{\mu\mu} & \ddot{\ell}_{\mu\alpha} \\ \ddot{\ell}_{\mu\alpha} & \ddot{\ell}_{\alpha\alpha} \end{pmatrix} = -\frac{n}{\sigma^2} \begin{pmatrix} 1 & \bar{x} \\ \bar{x} & \overline{xx} \end{pmatrix}.$$

Under the null hypothesis the slope parameter equals zero. The maximum likelihood estimator of μ is then simply the average $\bar{y}$. Plugging these estimates into the gradient and using the empirical Fisher information produces:

$$\dot{\ell}_0 = \frac{n}{\sigma^2} \begin{pmatrix} 0 \\ \overline{yx} - \bar{y}\bar{x} \end{pmatrix}, \quad H = (-\ddot{\ell})^{-1} = \frac{\sigma^2}{n(\overline{xx} - \bar{x}^2)} \begin{pmatrix} \overline{xx} & -\bar{x} \\ -\bar{x} & 1 \end{pmatrix}.$$

Putting everything together we obtain the score statistic:

$$(\dot{\ell}_0)' H (\dot{\ell}_0) = \frac{n(\overline{yx} - \bar{y}\bar{x})^2}{\sigma^2(\overline{xx} - \bar{x}^2)} = n(\hat{\rho})^2 \times \left(\frac{\overline{yy} - \bar{y}^2}{\sigma^2} \right).$$

Note that the resulting statistic is almost identical to the test statistic that was proposed in the previous section. The difference involves an extra factor, which is the ratio between the estimate of the variance of the phenotype (under the null) and the the actual variance. As a matter of fact, if we would have treated the parameter σ^2 itself as unknown then the resulting test statistic would coincide exactly with the test statistic proposed in the previous section. The parameter space as considered here is two-dimensional, since it includes two unknown parameters. Under the null assumption the dimension is one, since only the mean μ is unknown. Consequently, 2-1=1 degrees of freedom should be used for the selection of a threshold based on the chi-square distribution. The same difference will emerge in the case were σ^2 is also treated as an unknown parameter. The dimension of the full space is three. The dimension of the null hypothesis is two and the difference is one.

8 * The distribution of the score test

In the previous section we defined the statistic of the score test to be $(\dot{\ell}_0)' H (\dot{\ell}_0)$, where $\dot{\ell}_0$ is the derivative of the log-likelihood function, evaluated at the constraint maximum likelihood estimate of the parameter, and H is the inverse of the Fisher information matrix or an approximation thereof. In this section we bring an outline of a proof that the approximate null distribution of this test statistic is chi-square .

Consider, as first exercise, the distribution of the gradient of the log-likelihood function, evaluated at the actual value of the parameter. In many settings, including the setting of independence sampling, the gradient is of the form of a sum of independent terms. As such, it may be concluded from the Central Limit Theorem that the distribution of the gradient is normal. If order of taking derivatives with respect to θ and integrals with respect to the observations can be changed with out changing the outcome then the expectation and variance of the gradient can be easily evaluated. It can be shown that the expectation is the zero vector and the variance is the matrix $I(\theta)$ of Fisher Information. Indeed, for the expectation:

$$\mathbb{E}[\dot{\ell}(\theta)] = \mathbb{E}\frac{\dot{f}(x)}{f(x)} = \int \frac{\dot{f}(x)}{f(x)} f(x)\, dx = \frac{\partial}{\partial\theta} \int f(x)\, dx = \frac{\partial}{\partial\theta} 1 = 0.$$

In a similar fashion the statement regarding the variance matrix can be concluded from the fact that

$$\ddot{\ell}(\theta) = \frac{\ddot{f}(x)}{f(x)} - \dot{\ell}(\theta)\dot{\ell}(\theta)'$$

and that

$$\mathbb{E}\frac{\ddot{f}(x)}{f(x)} = \int \frac{\ddot{f}(x)}{f(x)} f(x)\, dx = \frac{\partial^2}{\partial^2 \theta} \int f(x)\, dx = \frac{\partial^2}{\partial^2 \theta} 1 = 0,$$

which leads to:

$$\mathrm{var}(\dot{\ell}(\theta)) = \mathbb{E}\big[\dot{\ell}(\theta)\dot{\ell}(\theta)'\big] = \mathbb{E}\frac{\ddot{f}(x)}{f(x)} - \mathbb{E}[\ddot{\ell}(\theta)] = I(\theta).$$

Given $\theta_0 \in H_0$. Consider the setting where the null hypothesis can be defined locally by a linear constraint of the form $\{\theta : A\theta = A\theta_0\}$, for some matrix A of full rank. Add the assumption that the log-likelihood function is well approximated in the vicinity of θ_0 by its second order Taylor expansion:

$$\ell(\theta) \approx \ell(\theta_0) + \dot{\ell}(\theta_0)'(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)'\ddot{\ell}(\theta_0)(\theta - \theta_0),$$

that the gradient score function is well approximated by its first order Taylor expansion:

$$\dot{\ell}(\theta) \approx \dot{\ell}(\theta_0) + \ddot{\ell}(\theta_0)(\theta - \theta_0),$$

and that the inverse of the empirical information matrix is well approximated by H, the inverse of the Fisher information matrix.

When the assumptions hold, then the problem of finding a constraint maximizer of the log-likelihood function is asymptotically equivalent to a solution of the quadratic maximization problem. The target function in the quadratic problem is the Taylor expansion of the log-likelihood function and constraint of belonging to the null hypothesis is the linear constrain: $A(\theta - \theta_0) = 0$. Application of the Lagrange multiplier technique will produce the necessary condition:

$$\dot{\ell}(\theta_0) + \ddot{\ell}(\theta_0)(\theta - \theta_0) - A'\lambda = 0,$$

for an appropriate vector of Lagrange multipliers λ. As a result, it may be concluded that $\dot{\ell}_0 \approx A'\lambda$. In the case when the matrix that produces the quadratic form, the empirical information matrix in our case, is invertible then the Lagrange multiplier has an explicit form:

$$\lambda = [A(\ddot{\ell}(\theta_0)^{-1})A']^{-1}A(\ell(\theta_0)^{-1})\dot{\ell}(\theta_0),$$

which proposes the approximation: $\dot{\ell}_0 \approx A'[AHA']^{-1}AH\dot{\ell}(\theta_0)$, which relates $\dot{\ell}_0$ to $\dot{\ell}(\theta_0)$. The matrix $B = H^{1/2}A'[AHA']^{-1}AH^{1/2}$ is a idempotent matrix with a rank that equals the rank of A. The distribution of $H^{1/2}\dot{\ell}(\theta_0)$ is approximately the distribution of independent standard normal random variables. It follows that the distribution of

$$(\dot{\ell}_0)'H(\dot{\ell}_0) \approx [H^{1/2}\dot{\ell}(\theta_0)]'B[H^{1/2}\dot{\ell}(\theta_0)]$$

is approximately chi-square with number of degrees of freedom equals to the rank of A, as required.