Part I

INTRODUCTION

The theory of statistical hypothesis testing was basically founded one hundred years ago by the Britons Ronald Aylmer Fisher, Egon Sharpe Pearson, and the Pole Jerzy Neyman. Nowadays it seems that we have a unique test theory for testing statistical hypothesis, but the opposite is true. On one hand Fisher developed the theory of significance testing and on the other hand Neyman and Pearson the theory of hypothesis testing.

Whereas with the Fisher theory the formulation of a null hypothesis is enough, Neyman's and Pearson's theory demands alternative hypotheses as well. They open the door to calculating error probabilities of two kinds, namely of a false rejection (type I error) and of a false acceptance (type II error) of the null hypothesis. This leads to the well known Neyman–Pearson lemma which helps us to find the best critical region for a hypothesis test with a simple alternative. The largest difference of both schools, however, are the Fisherian measure of evidence (p-value) and the Neyman–Pearson error rate (α).

With the Neyman–Pearson theory the error rate α is fixed and must be defined before performing the test. Within the Fisherian context the p-value is calculated from the value of the test statistic as a quantile of the test statistic distribution and serves as a measure of disproving the null hypothesis. Over the decades both theories have merged together. Today it is common practice – and described by most textbooks – to perform a Neyman–Pearson test and, instead of comparing the value of the test statistic with the critical region, to decide from the p-value. As this book is on testing statistical hypothesis with SAS and R we follow the common approach of mixing both theories. In SAS and R the critical regions are not reported, only p-values are given. We want to make the reader aware of this situation. In the next two chapters we shortly summarize the concept of statistical hypothesis testing and introduce the performance of statistical tests with SAS and R.

Statistical Hypothesis Testing with SAS and R, First Edition. Dirk Taeger and Sonja Kuhnt.

1

Statistical hypothesis testing

1.1 Theory of statistical hypothesis testing

Hypothesis testing is a key tool in statistical inference next to point estimation and confidence sets. All three concepts make an inference about a population based on a sample taken from it. Hypothesis testing aims at a decision on whether or not a hypothesis on the nature of the population is supported by the sample.

In the following we shortly run through the steps of a statistical test procedure and introduce the notation used throughout this book. For a detailed mathematical explanation please refer to the book by Lehmann (1997).

We denote a sample of size n by $x_1, \ldots, x_n$, where the x_i are observations of identically independently distributed random variables X_i, $i = 1, \ldots, n$. Usually some further assumptions are needed concerning the nature of the mechanism generating the sample. These can be rather general assumptions like a symmetric continuous distribution. Often a parametric distribution is assumed with only parameter values unknown, for example, the Gaussian distribution with both or either unknown mean and variance. In this case hypothesis tests deal with statements on the unknown population parameters. We exemplify our general discussion by this situation.

Each of the statistical tests presented in the following chapters is introduced by a verbal description of the type of conjecture to be decided upon together with the made assumptions. Next the **test problem** is formalized by the null hypothesis H_0 and the alternative hypothesis H_1. If a statement on population parameters is of interest, often the parameter space Θ, is partitioned into disjunct sets Θ_0 and Θ_1 with $\Theta_0 \cup \Theta_1 = \Theta$, corresponding to H_0 and H_1, respectively.

As the next building stone of a statistical test the **test statistic**, which is a function $T = f(X_1, \ldots, X_n)$ of the random sample, is stated. This function fulfills two criteria. First of all its value must provide insight on whether or not the null hypothesis might be true. Next the distribution of the test statistic must be known, given that the null hypothesis is true. Table 1.1 shows the four possible outcomes of a statistical test. In two of the cases the result of the test is a correct decision. Namely, a true null hypothesis is not rejected and a false null hypothesis is rejected. If the null hypothesis is true but is rejected as a result of

Statistical Hypothesis Testing with SAS and R, First Edition. Dirk Taeger and Sonja Kuhnt.

Table 1.1 Possible results in statistical testing.

		Test decision	
		Do not reject H_0	Reject H_0
Nature	H_0 true	Correct decision	Type I error
	H_0 false	Type II error	Correct decision

the test, a **type I error** occurs. In the opposite situation that H_1 is true in nature but the test does not reject the null hypothesis, a **type II error** occurs.

Generally, unless sample size or hypothesis are changed, a decrease in the probability of a type I error causes an increase in the probability for a type II error and vice versa. With the **significance level** α the maximal probability of the appearance of a type I error is fixed and the **critical region** of the test is chosen according to this condition. If the observed value of the test statistic lies in the critical region, the null hypothesis is rejected. Hence, the error probability is under control when a decision is made against H_0 but not when the decision is for H_0, which needs to be kept in mind while drawing conclusions from test results. If possible, the researcher's conjecture corresponds to the alternative hypothesis due to primarily controlling the type I error. However, in goodness-of-fit tests one is forced to formulate the researcher's hypothesis, that is, the specific distribution of interest, as null hypothesis as it is otherwise usually unfeasible to derive the distribution of the test statistic.

The power function measures the quality of a test. It yields the probability of rejecting the hypothesis for a given true parameter value θ. The test with the greatest power among all tests with a given significance level α is called the most powerful test.

Traditionally a pre-specified significance level of $\alpha = 0.5$ or $\alpha = 0.1$ is selected. However, there is no reason why a different value should not be chosen.

Up to here we are in the context of the Neyman–Pearson test theory. Most statistical computer programs are not returning whether the calculated test statistic lies within the critical region or not. Instead the **p-value** (probability-value) is given. This is the probability to obtain the observed value of the test statistic or a value that is more extreme in the direction of the alternative hypothesis calculated when H_0 is true. If the p-value is smaller than α it follows that H_0 is rejected, otherwise H_0 is not rejected.

As already mentioned in the introduction this is the common approach. For further reading on the differences please refer to Goodman (1994), Hubbard and Bayarri (2003), Johnstone (1987), and Lehmann (1993).

1.2 Testing statistical hypothesis with SAS and R

Testing statistical hypotheses with SAS and R is very convenient. A lot of tests are already integrated in these software packages. In SAS tests are invoked via procedures while R uses functions. Although many test problems are handled in this way situations may occur where a SAS procedure or a R function is not available. Reasons are manifold. The SAS Institute decides which statistical test to include in SAS. Even if a newly developed test is accepted for inclusion in SAS it takes some time to develop a new procedure or to incorporate it in an existing SAS procedure. If a test is not implemented in a SAS procedure or in the R standard packages the likelihood is high to find the test as a SAS macro or in R user packages which

are available through the World Wide Web. However, in this book we have refrained from presenting tests from SAS macros or R user packages for several reasons. We do not know how long macros, program code, or user packages are supported by the programmer and are therefore available for newer versions of SAS or R. In addition it is not possible to trace if the code is correct. If a statistical test is not implemented in the SAS software as procedure or in the R standard packages we will provide an algorithm with small SAS and R code to circumvent these problems. All presented statistical tests are accompanied by an example of their use in a given dataset. So it is easy to retrace the example and to translate the code to your own datasets. Sometimes more than one SAS procedure or R function is available to perform a statistical test. We only present one way to do so.

1.2.1 Programming philosophy of SAS and R

Testing statistical hypothesis in SAS or R is not the same, while R is a matrix language orientated software, SAS follows a different philosophy (except for SAS/IML). With a matrix orientated language some calculations are easier. For instance the average of a few observations, for example, the age 1, 4, 2 and 5 of four children in a family, can be calculated with one line of code in R by applying the function `mean()` to the vector containing the values, `c(1,4,2,5)`.

```
mean(c(1,4,2,5))
```

Here the numeric vector of data values to be analyzed is inserted directly in the R function. However, it is also possible to call data from a previously defined object, for example, a dataframe

```
children<-data.frame(age=c(1,4,2,5))
 mean(children$age)
```

In SAS a little more effort is necessary due to the required division into data and proc steps.

```
data children;
 input age;
 datalines;
1
4
2
5
;
run;

proc means;
 var age;
run;
```

The dataset `children` holds the variable `age` with observed values 1, 4, 2 and 5. The SAS procedure `proc means` calculates the mean value. This type of programming philosophy must not be a disadvantage. It can save a lot of time, because the SAS procedures are very powerful and incorporate many statistical calculations in one go.

We assume that the reader is familiar with the basic programming features of SAS or R, such as data input and output, and only remark on some important points related to conducting statistical tests. Concerning data format usually one entry per observation and a column for each variable are suitable. However, in some cases it may be required to reorganize the dataset for test procedures. We accompany our examples with small datasets (see Appendix A), such that it is easy to see how data need to be arranged for the specific test.

In SAS most statistical tests are performed with procedures, which usually follow the schema:

```
proc proc-name data=dataset-name options;
  var variable-names options;
  options;
run;
```

The `data=` statement identifies the dataset to be analyzed. If missing, the most recent dataset is taken. In some procedures it is necessary to fix some options to set up the statistical test, for example, to define the value to test against, or if the test is one or two sided. The `var` statement is followed by the variables on which the test shall be performed. Sometimes further options can be stated in separate command lines, for instance requesting an exact test. Note, some procedures differ from this general set-up. The procedure `proc freq` as an example has no `var` but a `table` statement. Occasionally the statement `class` *class-variable* is needed indicating a grouping variable which assigns each observation to a specific group. As options of procedures can be numerous and not all of them may be needed for the treated test, we restrict our exposure to the indispensable options. The same applies to the output we present for the examples.

Conducting a statistical test in the program R usually only requires one line of code. The common layout of R functions is:

```
function-name(x, options)
```

The *function-name* identifies the function to be applied to the data *x*. In two-sample tests data on a second variable are needed, such that the general layout is extended to:

```
function-name(x,y, options)
```

Options differ for each test, but the option `alternative=`*alternative-hypothesis* occurs often. As *alternative-hypothesis* of `"two.sided"`, `"less"`, or `"greater"` is chosen, depending on how the alternative hypothesis is to be specified. It suffices to state only the first letter, that is, `"t"`, `"l"`, or `"g"`. As in SAS we only present the options that are necessary to perform the test and restrict the presented output to the relevant parts.

1.2.2 Testing in SAS and R – An example

To demonstrate the testing of hypothesis in SAS and R let us look at the ordinary t-test which tests if a population mean μ differs from a given values μ_0. We employ the dataset in Table A.1 from Appendix A containing observations on three variables for 55 people: subject number (*no*), status of the subject (*status*), and systolic blood pressure in millimeters of

mercury (*mmhg*). Now, we want to test if the mean systolic blood pressure of the population differs statistically significantly from 140 mmHg at the 5% level . The null hypothesis is given by $H_0 : \mu = 140$ and the alternative hypothesis is $H_1 : \mu \neq 140$. We assume that the systolic blood pressure is normally distributed.[1]

SAS provides the procedure `ttest` to handle this test problem. The SAS code is:

```
proc ttest data=blood_pressure ho=140;
 var mmhg;
run;
```

The dataset option `data=` specifies the dataset and the option `ho=` the null value to test. With `var mmhg` you tell SAS that the variable `mmhg` is the variable which contains the observations to be used. In the output containing, for example, the mean, standard error and 95% confidence interval, the following refers to the statistical test:

```
DF      t Value    Pr > |t|
54      -3.87      0.0003
```

`DF` characterizes the degrees of freedom of the t-distribution, as the test statistic is t-distributed. The value of the test statistic (`t Value`) is −3.87 and the corresponding p-value (`Pr > |t|`) is 0.0003. So we can conclude that the mean value differs statistically significantly from 140 mmHg at a significance level of 5%.

As in SAS it is also simple in R to conduct a t-test:

```
t.test(blood_pressure$mmhg,mu=140)
```

The first argument calls the data on the variable `mmhg` from the dataset `blood_pressure`. The second argument `mu=` specifies the value of the null hypothesis. The most relevant part of the output for the testing problem is:

```
t = -3.8693, df = 54, p-value = 0.0002961
```

The values are the same as for the SAS procedure of course, except for the fact that they are rounded to more digits. A nice feature of R is that it returns the alternative hypothesis with the output:

```
alternative hypothesis: true mean is not equal to 140
```

If a ready to use SAS procedure or R function is not available, we have to calculate the test statistic and compare it to the corresponding test statistic distribution by hand. The formula for the test statistic of the t-test is given by:

$$T = \frac{\overline{X} - \mu_0}{s}\sqrt{n}, \text{ with } s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2},$$

[1] As the systolic blood pressure only takes positive values, the assumption of a normal distribution is strictly speaking not appropriate. However, blood pressure measurements usually lie in a region far away from zero, so that in this case the t-test can be expected to be reasonably robust against this violation.

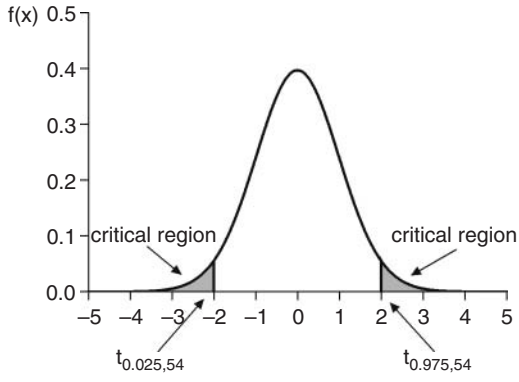

Figure 1.1 Critical regions (shaded areas) of a t-distribution with 54 degrees of freedom with significance level α=0.05.

and follows a t-distribution with $n-1$ degrees of freedom, where n is the sample size. The null hypothesis is rejected if for the observed value t of T either $t < t_{\alpha/2,n-1}$ or $t > t_{1-\alpha/2,n-1}$ holds. These quantiles are describing the critical regions of this test (see Figure 1.1 for $\alpha = 0.05$). The p-value is calculated as $p = 2\,P(T \leq (-|t|))$, where $P(.)$ denotes the probability function of the t-distribution with $n-1$ degrees of freedom.

Let us start with SAS to program this test by hand.

```
* Calculate sample mean and standard deviation;
proc means data=blood_pressure mean std;
 var mmhg;
 output out=ttest01 mean=meanvalue std=sigma;
run;

* Calculate test statistic;
data ttest02;
 set ttest01;
  mu0=140;    * Set mean value under the null hypothesis;
  t=sqrt(55)*(meanvalue-mu0)/sigma;
run;

* Output results;
proc print;
 var t;
run;
```

The output gives a t-value of -3.86927. The critical values $t_{0.025,54}$ and $t_{0.975,54}$ can be calculated with the SAS function `TINV`, which returns the quantiles of a t-distribution.

```
data temp;
x=tinv(0.025,54);
run;
```

Here `tinv(0.025,54)` gives -2.004879 and `tinv(0.975,54)` returns 2.004879. Because the t-value -3.86927 calculated for the dataset is less than -2.004879 we reject the null hypothesis at the 5% level.

The p-value is also not complicated to calculate. The probability distribution function of the t-distribution in SAS is `PROBT` and `2*probt(-3.86927,54)` gives a p-value of 0.0002961135.

To write a code for the same t-test in R is quite easy as well.

```
# Calculate sample mean and standard deviation
xbar<-mean(blood_pressure$mmhg)
sigma<-sd(blood_pressure$mmhg)

# Set mean value under the null hypothesis
mu0<-140

# Calculate test statistic
t<-sqrt(55)*(xbar-mu0)/sigma

# Output results
t
```

This R code returns the test statistic value of t = −3.869272. To calculate the boundaries of the critical regions the R function `qt` can be used, where `qt(0.025,54)` returns −2.004879 and `qt(0.975,54)` returns 2.004879. The p-value is calculated as `2*pt(-3.869272,54)` with the function `pt` of the probability function of the t-distribution and has a value of 0.0002961135.

The three typical hypotheses for a t-test are:

(A) $H_0 : \mu = \mu_0$ vs $H_1 : \mu \neq \mu_0$

(B) $H_0 : \mu \leq \mu_0$ vs $H_1 : \mu > \mu_0$

(C) $H_0 : \mu \geq \mu_0$ vs $H_1 : \mu < \mu_0$

with μ the sample mean and $\mu_0 = 140$ mmHg in our example. So far case (A) has been treated. Let us now look at the t-tests for hypotheses (B) and (C) at the 5% significance level.[2] The significance level α is no longer split between the lower and upper critical regions. For hypothesis (B) the decision rule is: reject H_0 if for the observed value t of T it holds that $t > t_{1-\alpha,n-1}$ and for hypothesis (C) reject H_0 if for the observed value t of T it holds that $t < t_{\alpha,n-1}$. In our example with significance level 0.05 the boundaries for the critical regions are 1.673565 for hypothesis (B) and −1.673565 for hypothesis (C). See Figure 1.2 and Figure 1.3 for a graphical representation.

In SAS these values are computed as `tinv(0.95,54)` for (B) and `tinv(0.05,54)` for (C). In R these values are computed as `qt(0.95,54)` for (B) and `qt(0.05,54)` for (C). Please note, both boundaries of the critical regions are the same except for the algebraic sign as the t-distribution is a symmetric distribution. SAS and R do not report the critical values, only p-values−as any statistical software we know. Some tables of critical values for several distributions can be found in Appendix B.

The option `sides=U` of the procedure `proc ttest` forces SAS to test the one-sided hypothesis were the alternative hypothesis is that the true mean is greater than μ_0. The output is:

```
DF     t Value     Pr > t
54       -3.87     0.9999
```

[2] For scientific correctness the significance level always needs to be decided upon before conducting the test.

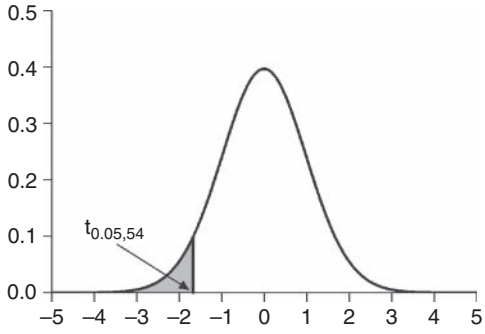

Figure 1.2 Lower critical region (shaded area) and critical value of a one-sided test with significance level of 5% (t-distribution with 54 degrees of freedom).

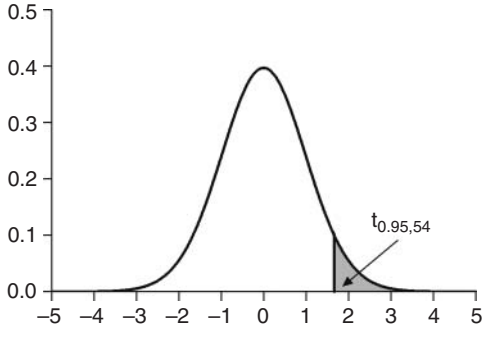

Figure 1.3 Upper critical region (shaded area) and critical value of a one-sided test with significance level of 5% (t-distribution with 54 degrees of freedom).

The R code for this hypothesis is:

```
t.test(blood_pressure$mmhg,mu=140,alternative="greater")
```

and gives a similar output:

```
t = -3.8693, df = 54, p-value = 0.9999
alternative hypothesis: true mean is greater than 140
```

To test the hypothesis (C) $H_0 : \mu \geq \mu_0$ vs $H_1 : \mu < \mu_0$ is not complicated at all. In SAS the following code

```
proc ttest data=blood_pressure ho=140 sides=L;
 var mmhg;
run;
```

yields the output.

```
DF     t Value     Pr < t
54       -3.87     0.0001
```

Here the option `sides=L` forces SAS to test the one-sided hypothesis where the alternative hypothesis is that the true mean is lower than μ_0. In R the tests is done with

```
t.test(blood_pressure$mmhg,mu=140,alternative="less")
```

and returns

```
t = -3.8693, df = 54, p-value = 0.0001481
alternative hypothesis: true mean is less than 140
```

1.2.3 Calculating p-values

Generally the calculation of a p-value is straightforward. In the case of the t-test the p-value is either the area under the probability curve lower or greater than the calculated t-value, that is, the p-value is $P(T \leq t)$ for hypothesis (C) and $P(T \geq t)$ for hypothesis (B), where $P(T \geq t) = 1 - P(T \leq t)$. The SAS function `probt` and the R function `pt` calculate the probability that $P(T \leq t)$ for the t-distribution.

The p-value of hypothesis (A) is twice the minimum of the lowest p-value of the one-sided tests, which is equal to $2 * P(T \leq -|t|)$. However, keep in mind that this is only valid for symmetric distributions like the t- or Gaussian distribution. If the test statistic is a non-symmetric distribution like the F-distribution it is necessary to work out if the observed value is at the lower or upper tail of the distribution. For a two-sided test use $2 * \min[1 - P(X \leq x), P(X \leq x)]$. This ensures the lowest two-sided p-value is obtained and not, on occasion, a p-value above 1 [see Gallagher (2006) for further discussion].

It is usual to format p-values such that values between 0 and 0.0001 are printed as `<0.0001` and values above are restricted to four digits. In SAS procedures this is the case. If calculating a p-value yourself you can us the SAS format `pvalue`.

Table 1.2 Some distribution functions in SAS.

SAS function	Parameters	Distribution	Examples
probnorm(x)	x: value of the test statistic	Gaussian	probnorm(1.96)
probt(x,df,nc)	x: value of the test statistic df: degrees of freedom nc: noncentrality parameter (optional)	t	probt(12.71,1)
probchi(x,df,nc)	x: value of the test statistic df: degrees of freedom nc: noncentrality parameter (optional)	χ^2	probchi(5.02,1)
probf(x,ndf,ddf,nc)	x: value of the test statistic ndf: numerator degrees of freedom ddf: denominator degrees of freedom nc: noncentrality parameter (optional)	F	probf(647.80,1,1)

All functions calculate the probability that an observation of the specific distribution is $\leq X$

```
data ttest;
 format p_value pvalue.;
 p_value=2*probt(-3.86927,54);
run;
```

This will result in a p-value of `0.0003`.

Usually the p-value in R is not formatted in this way. The function `format.pval` can be used to format it. But first with the R option `scipen=3` the scientific notation should be suppressed.

```
options(scipen=3)
format.pval(2*pt(-3.86927,54),1,eps=0.0001)
```

This R code will also give a p-value of `0.0003`.

If necessary it is possible to derive quantiles, and thereby p-values and critical values, by simulation. Let us stick to the assumed symmetric t-distribution. In SAS the code could look as follows:

```
data random;
 do i=1 to 100000;
 r=rand('T',54);
 output;
 end;
run;

proc univariate data=random noprint;
  var r;
  output out=critical pctlpts=2.5 97.5 pctlpre=crit_;
run;

proc print data=critical;
run;
```

The `rand('T',54)` function calculates a random number from a t-distribution with 54 degrees of freedom. This is repeated 100 000 times and the random numbers are stored. The `proc univariate` procedure calculates the desired quantiles using the option *pctlpts*=. The additional option *pctlpre*= is used to give a prefix to the variable names of the calculated quantiles. The output is then for the 0.025-quantile and 0.975-quantile calculated here

```
crit_2_5    crit_97_5
-2.01062    2.01350
```

In R we need only one line of code

```
quantile(rt(100000,54),c(0.025,0.975))
```

The `quantile(.)` function calculates quantiles and the `rt(.)` function calculates 100 000 random numbers of the desired t-distribution. The option `c(0.025,0.975)` then calculates the 0.025-quantile and 0.975-quantile of these random numbers. We get the output

```
     2.5%     97.5%
-2.020038  2.014382
```

Table 1.3 Some distribution functions in R.

R function	Parameters	Distribution	Examples
pnorm(x)	x: value of the test statistic	Gaussian	pnorm(1.96)
pt(x,df,ncp)	x: value of the test statistic df: degrees of freedom ncp: noncentrality parameter (optional)	t	pt(12.71,1)
pchisq(x,df,nc)	x: value of the test statistic df: degrees of freedom ncp: noncentrality parameter (optional)	χ^2	pchisq(5.02,1)
pf(x,ndf,ddf,nc)	x: value of the test statistic df1: numerator degrees of freedom df2: denominator degrees of freedom ncp: noncentrality parameter (optional)	F	pf(647.80,1,1)

All listed functions calculating the probability that an observation of the specific distribution is $\leq X$

These numbers are different to those SAS returned and they will always vary if you try it by yourself, because random numbers should differ from run to run. However, we see that these values are very close to the critical values ± 2.004879 that are given by the quantiles of the t-distribution.

For some tests, for example, the Jarque–Bera test (see Test 11.2.2), these kinds of Monte Carlo simulation are needed to get the critical values. For some tests implemented in SAS and R this Monte Carlo approach can be additionally requested.

Remember that in most cases the p-value is stated in the output of the applied procedure or function. If the statistical test you would like to apply is not implemented in SAS or R you have to write it yourself (or use the code we provide in such situations). Table 1.2 and Table 1.3 list some common distribution functions in SAS and R, respectively, which are of interest in calculating p-values from parametric distributions. For nonparametric tests or tests with distributions other than stated above or implemented in SAS or R, p-value calculation is sometimes cumbersome. If necessary we provide code for such problems.

1.3 Presentation of the statistical tests

In this section we use the single sample t-test again to describe the presentation of statistical tests in this book. The layout follows a structured table.

First the name of the test is given
2.1.2 t-Test

A brief description of the test follows	
Description:	Tests if a population mean μ differs from a specific value μ_0.

Assumptions of the test are listed

Assumptions: • Data are randomly sampled from a Gaussian distribution.

Different hypotheses are listed. In this case the non-directional hypothesis and the two directional hypotheses can be tested with this test

Hypotheses:
(A) $H_0 : \mu = \mu_0$ vs $H_1 : \mu \neq \mu_0$
(B) $H_0 : \mu \leq \mu_0$ vs $H_1 : \mu > \mu_0$
(C) $H_0 : \mu \geq \mu_0$ vs $H_1 : \mu < \mu_0$

The test statistic is displayed

Test statistic: $T = \frac{\overline{X}-\mu_0}{s}\sqrt{n}$ with $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2}$

Decision rules for each hypothesis are listed

Test decision: Reject H_0 if for the observed value t of T
(A) $t < t_{\alpha/2,n-1}$ or $t > t_{1-\alpha/2,n-1}$
(B) $t > t_{1-\alpha,n-1}$
(C) $t < t_{\alpha,n-1}$

Formulae of p-values for each hypothesis are given

p-value:
(A) $p = 2\ P(T \leq (-|t|))$
(B) $p = 1 - P(T \leq t))$
(C) $p = P(T \leq t))$

Annotations of the test, such as the distribution of the test statistic, are pointed out

Annotations: • Test statistic T is t-distributed with $n - 1$degrees of freedom.

An explaining example on a dataset is introduced

Example: To test the hypothesis that the mean systolic blood pressure of a certain population equals 140 mmHg. The dataset at hand has measurements on 55 patients (dataset in Table A.1).

The SAS code of the example is given

SAS code

```
proc ttest data=blood_pressure ho=140 sides=2;
 var mmhg;
run;
```

The SAS output of the example code is given

SAS output

```
   DF      t Value    Pr < t
   54       -3.87     0.0003
```

Remarks concerning the SAS code and/or output are given

Remarks:

- `ho=`*value* is optional and defines the value μ_0 to test against. Default is 0.

The R code of the example is given

R code

```
t.test(blood_pressure$mmhg,mu=140,alternative="two.sided")
```

The R output of the example code is given

R output

```
t = -3.8693, df = 54, p-value = 0.0002961
```

Remarks concerning the R code and/or output are given

Remarks:

- `mu=`*value* is optional and defines the value μ_0 to test against. Default is 0.

References

Gallagher J. 2006 The F test for comparing two normal variances: correct and incorrect calculation of the two-sided p-value? *Teaching Statistics* **28**, 58–60.

Goodman S.N. 1994 P values, hypothesis tests, and likelihood: implications for epidemiology of a neglected historical debate. *American Journal of Epidemiology* **139**, 116–118.

Hubbard R.H. and Bayarri M.J. 2003 Confusion over measures of evidence (p's) versus errors (α's) in classical statistical testing (with discussions). *The American Statistician* **57**, 171–182.

Johnstone D.J. 1987 Tests on significance following R. A. Fisher. *The British Society for the Philosophy of Science* **38**, 481–499.

Lehmann E.L. 1993 The Fisher, Neyman–Pearson theory of testing hypothesis: one theory or two? *Journal of the American Statistical Association* **88**, 1242–1249.

Lehmann E.L. 1997 *Testing Statistical Hypotheses*, 2nd edn. Springer.