Chapter 4
Analytical statistics

The most important questions of life are,
for the most part, really only questions of probability.
Pierre-Simon Laplace
(from <http://www-rohan.sdsu.edu/%7Emalouf/>)

In my description of the phases of an empirical study in Chapter 1, I skipped over one essential step: how to decide which significance test to use (Section 1.3.4). In this chapter, I will now discuss this step in some detail as well as then discuss how to conduct a variety of significance tests you may want to perform on your data. More specifically, in this chapter I will explain how descriptive statistics from Chapter 3 are used in the domain of hypothesis-testing. For example, in Section 3.1 I explained how you compute a measure of central tendency (such as a mean) or a measure of dispersion (such as a standard deviation) for a particular sample. In this chapter, you will see how you test whether such a mean or such a standard deviation differs significantly from a known mean or standard deviation or the mean or standard deviation of a second sample.

However, before we begin with actual tests: how do you decide which of the many tests out there is required for your hypotheses and data? One way to try to narrow down the truly bewildering array of tests is to ask yourself the six questions I will list in (24) to (29) and discuss presently, and the answers to these questions usually point you to only one or two tests that you can apply to your data. (A bit later, I will also provide a visual aid for this process.).

Ok, here goes. The first question is shown in (24).

(24) What kind of study are you conducting?

Typically, there are only two possible answers to that question: "hypothesis-generating" and "hypothesis-testing." The former means that you are approaching a (typically large) data set with the intentions of detecting structure(s) and developing hypotheses for future studies; your approach to the data is therefore data-driven, or bottom-up; an example for this will be discussed in Section 5.6. The latter is what most of the examples in this

book are about and means your approach to the data involves specific hypotheses you want to test and requires the types of tests in this chapter and most of the following one.

(25) What kinds of variables are involved in your hypotheses, and how many?

There are essentially two types of answers. One pertains to the information value of the variables and we have discussed this in detail in Section 1.3.2.2 above. The other allows for four different possible answers. First, you may only have one dependent variable, in which case, you normally want to compute a so-called goodness-of-fit test to test whether the results from your data correspond to other results (from a previous study) or correspond to a known distribution (such as a normal distribution). Examples include

- is the ratio of *no*-negations (e.g., *He is no stranger*) and *not*-negations (e.g., *He is not a stranger*) in your data 1 (i.e., the two negation types are equally likely)?
- does the average acceptability judgment you receive for a sentence correspond to that of a previous study?

Second, you may have one dependent and one independent variable or you may just have two sets of measurements (i.e. two dependent variables). In both cases you typically want to compute a monofactorial test for independence to determine whether the values of one/the independent variable are correlated with those of the other/dependent variable. For example,

- does the animacy of the referent of the direct object (a categorical independent variable) correlate with the choice of one of two postverbal constituent orders (a categorical dependent variable)?
- does the average acceptability judgment (a mean of a ratio/interval dependent variable) vary as a function of whether the subjects doing the rating are native speakers or not (a categorical independent variable)?

Third, you may have one dependent and two or more independent variables, in which case you want to compute a multifactorial analysis (such as a multiple regression) to determine whether the individual independent variables and their interactions correlate with, or predict, the dependent variable. For example,

- does the frequency of a negation type (a categorical dependent variable with the levels *NO* vs. *NOT*; cf. above) depend on the mode of communication (a binary independent variable with the levels *SPOKEN* vs. *WRITTEN*), the type of verb that is negated (a categorical independent variable with the levels *COPULA*, *HAVE*, or *LEXICAL*), and/or the interaction of these independent variables?
- does the reaction time to a word *w* in a lexical decision task (a ratio-scaled dependent variable) depend on the word class of *w* (a categorical independent variable), the frequency of *w* in a reference corpus (a ratio/interval independent variable), whether the subject has seen a word semantically related to *w* on the previous trial or not (a binary independent variable), whether the subject has seen a word phonologically similar to *w* on the previous trial or not (a binary independent variable), and/or the interactions of these independent variables?

Fourth, you have two or more dependent variables, in which case you may want to perform a multivariate analysis, which can be exploratory (such as hierarchical cluster analysis, principal components analysis, factor analysis, multi-dimensional scaling, etc.) or hypothesis-testing in nature (MANOVA). For example, if you retrieved from corpus data ten words and the frequencies of all content words occurring close to them, you can perform a cluster analysis to see which of the words behave more (or less) similarly to each other, which often is correlated with semantic similarity.

(26) Are data points in your data related such that you can associate them to each other meaningfully and in a principled way?

This question is concerned with whether you have what are called independent or dependent samples (and brings us back to the notion of independence discussed in Section 1.3.4.1). For example, your two samples – e.g., the numbers of mistakes made by ten male and ten female non-native speakers in a grammar test – are independent of each other if you cannot connect each male subject's value to that of one female subject on a meaningful and principled basis. You would not be able to do so if you randomly sampled ten men and ten women and let them take the same test.

There are two ways in which samples can be dependent. One is if you test subjects more than once, e.g., before and after a treatment. In that case, you could meaningfully connect each value in the before-treatment sample to a value in the after-treatment sample, namely connect each subject's two values. The samples are dependent because, for instance, if subject #1 is

very intelligent and good at the language tested, then these characteristics will make his results better than average in both tests, esp. compared to a subject who is less intelligent and proficient in the language and who will perform worse in both tests. Recognizing that the samples are dependent this way will make the test of before-vs.-after treatments more precise.

The second way in which samples may be dependent can be explained using the above example of ten men and ten women. If the ten men were the husbands of the ten women, then one would want to consider the samples dependent. Why? Because spouses are on average more similar to each other than randomly chosen people: they often have similar IQs, similar professions, they spend more time with each other than with randomly-selected people, etc. Thus, one should associate each husband with his wife, making this two dependent samples.

Independence of data points is often a very important criterion: many tests assume that data points are independent, and for many tests you must choose your test depending on what kind of samples you have.

(27) What is the statistic of the dependent variable in the statistical hypotheses?

There are essentially five different answers to this question, which were already mentioned in Section 1.3.2.3 above, too. Your dependent variable may involve frequencies/counts, central tendencies, dispersions, correlations, or distributions.

(28) What does the distribution of the data or your test statistic look like? Normal, some other way that can ultimately be described by a probability function (or a way that can be transformed to look like a probability function), or some other way?
(29) How big are the samples you collected? $n < 30$ or $n \geq 30$?

These questions relate back to Section 1.3.4, where I explained two things: First, if your data / test statistics follow a particular probability distribution, you can often use a computationally simpler parametric test, and if your data / test statistics don't, you must often use a non-parametric test. Second, given sufficient sample sizes, even data from a decidedly non-normal distribution can begin to look normal and, thus, allow you to apply parametric tests. It is safer, however, to be very careful and, maybe be conservative and run both types of tests.

Let us now use a graph (<sflwr_navigator.png>) that visualizes this pro-

cess, which you should have downloaded as part of all the files from the companion website. Let's exemplify the use of this graph using the above example scenario: you hypothesize that the average acceptability judgment (a mean of an ordinal dependent variable) varies as a function of whether the subjects providing the ratings are native or non-native speakers (a binary/categorical independent variable).

You start at the rounded red box with *approach* in it. Then, the above scenario is a hypothesis-testing scenario so you go down to *statistic*. Then, the above scenario involves averages so you go down to the rounded blue box with *mean* in it. Then, the hypothesis involves both a dependent and an independent variable so you go down to the right, via *1 DV 1 IV* to the transparent box with (tests for) *independence/difference* in it. You got to that box via the blue box with *mean* so you continue to the next blue box containing *information value*. Now you make two decisions: first, the dependent variable is ordinal in nature. Second, the samples are independent. Thus, you take the arrow down to the bottom left, which leads to a blue box with *U*-test in it. Thus, the typical test for the above question would be the *U*-test (to be discussed below), and the R function for that test is already provided there, too: `wilcox.test`.

Now, what does the dashed arrow mean that leads towards that box? It means that you would also do a *U*-test if your dependent variable was interval/ratio-scaled but violated other assumptions of the *t*-test. That is, dashed arrows provide alternative tests for the first-choice test from which they originate.

Obviously, this graph is a simplification and does not contain everything one would want to know, but I think it can help beginners to make first choices for tests so I recommend that, as you continue with the book, you always determine for each section which test to use and how to identify this on the basis of the graph.

Before we get started, let me remind you once again that in your own data your nominal/categorical variables should ideally always be coded with meaningful character strings so that R recognizes them as factors when reading in the data from a file. Also, I will assume that you have downloaded the data files from the companion website.

Recommendation(s) for further study
Good and Hardin (2012: Ch. 6) on choosing test statistics

1. Distributions and frequencies

In this section, I will illustrate how to test whether distributions and frequencies from one sample differ significantly from a known distribution (cf. Section 4.1.1) or from another sample (cf. Section 4.1.2). In both sections, we begin with variables from the interval/ratio level of measurement and then proceed to lower levels of measurement.

1.1. Distribution fitting

1.1.1. One dep. variable (ratio-scaled)

In this section, I will discuss how you compare whether the distribution of one dependent interval-/ratio-scaled variable is significantly different from a known distribution. I will restrict my attention to one of the most frequent cases, the situation where you test whether a variable is normally distributed (because as mentioned above in Section 1.3.4, many statistical techniques require a normal distribution so you must some know test like this).

We will deal with an example from the first language acquisition of tense and aspect in Russian. Simplifying a bit here, one can often observe a relatively robust correlation between past tense and perfective aspect as well as non-past tenses and imperfective aspect. Such a correlation can be quantified with Cramer's *V* values (cf. Stoll and Gries, 2009, and Section 4.2.1 below). Let us assume you studied how this association – the Cramer's *V* values – changes for one child over time. Let us further assume you had 117 recordings for this child, computed a Cramer's *V* value for each one, and now you want to see whether these are normally distributed. This scenario involves

- a dependent interval/ratio-scaled variable called TENSEASPECT, consisting of the Cramer's *V* values;
- no independent variable because you are not testing whether the distribution of the variable TENSEASPECT is influenced by, or correlated with, something else.

You can test for normality in several ways. The test we will use is the Shapiro-Wilk test (remember: check <sflwr_navigator.png> to see how we get to this test!), which does not really have any assumptions other than ratio-scaled data and involves the following procedure:

Procedure
- Formulating the hypotheses
- Visualizing the data
- Computing the test statistic W and p

As always, we begin with the hypotheses:

H_0: The data points do not differ from a normal distribution; $W = 1$.
H_1: The data points differ from a normal distribution; $W \neq 1$.

First, you load the data from <_inputfiles/04-1-1-1_tense-aspect.csv> and create a graph; the code for the left panel is shown below but you can also generate the right panel using the code from the code file.

```
> RussianTensAsp<-read.delim(file.choose())¶
> attach(RussianTensAsp)¶
> hist(TENSE_ASPECT, xlim=c(0, 1), main="", xlab="Tense-Apect
   correlation", ylab="Frequency") # left panel¶
```

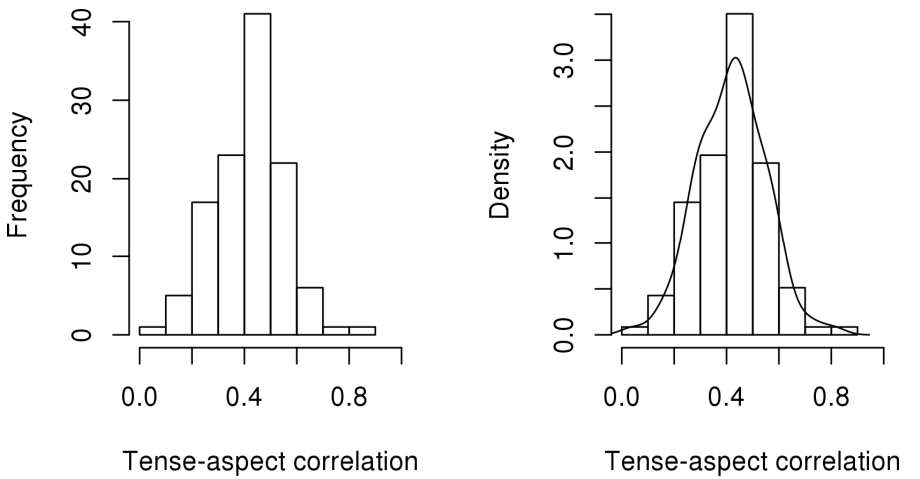

Figure 39. Histogram of the Cramer's V values reflecting the strengths of the tense-aspect correlations

At first glance, this looks very much like a normal distribution, but of course you must do a real test. The Shapiro-Wilk test is rather cumbersome to compute semi-manually, which is why its manual computation will not be discussed here (unlike nearly all other monofactorial tests). In R, how-

ever, the computation could not be easier. The relevant function is called `shapiro.test` and it only requires one argument, the vector to be tested:

```
> shapiro.test(TENSE_ASPECT)¶
        Shapiro-Wilk normality test
data:  TENSE_ASPECT
W = 0.9942, p-value = 0.9132
```

What does this mean? This simple output teaches an important lesson: Usually, you want to obtain a significant result, i.e., a *p*-value that is smaller than 0.05 because this allows you to accept H_1. Here, however, you may actually welcome an insignificant result because normally-distributed variables are often easier to handle. The reason for this is again the logic underlying the falsification paradigm. When $p < 0.05$, you reject H_0 and accept H_1. But here you 'want' H_0 to be true because H_0 states that the data are normally distributed. You obtained a *p*-value of 0.9132, which means you cannot reject H_0 and, thus, consider the data to be normally distributed. You would therefore summarize this result in the results section of your paper as follows: "According to a Shapiro-Wilk test, the distribution of this child's Cramer's *V* values measuring the tense-aspect correlation does not deviate significantly from normality: $W = 0.9942$; $p = 0.9132$." (In parentheses or after a colon you usually mention all statistics that helped you decide whether or not to accept H_1.)

As an alternative to the Shapiro-Wilk test, you can also use a Kolmogorov-Smirnov test for goodness of fit. This test requires the function `ks.test` and is more flexible than the Shapiro-Wilk-Test, since it can test for more than just normality and can also be applied to vectors with more than 5000 data points. To test the Cramer's *V* value for normality, you provide them as the first argument, then you name the distribution you want to test against (for normality, `"pnorm"`), and then, to define the parameters of the normal distribution, you provide the mean and the standard deviation of the Cramer's *V* values:

```
> ks.test(TENSE_ASPECT, "pnorm", mean=mean(TENSE_ASPECT),
   sd=sd(TENSE_ASPECT))¶
 One-sample Kolmogorov-Smirnov test
data:  TENSE_ASPECT
D = 0.078, p-value = 0.4752
alternative hypothesis: two-sided
```

The result is the same as above: the data do not differ significantly from normality. You also get a warning because `ks.test` assumes that no two

values in the input are the same, but here some values (e.g., 0.27, 0.41, and others) are attested more than once; below you will see a quick and dirty fix for this problem.

Recommendation(s) for further study
- as alternatives to the above functions, the functions `jarqueberaTest` and `dagoTest` (both from the library `fBasics`)
- the function `mshapiro.test` (from the library `mvnormtest`) to test for multivariate normality
- the function `qqnorm` and its documentation (for quantile-quantile plots)
- Crawley (2005: 100f.), Crawley (2007: 316f.), Sheskin (2011: Test 7)

1.1.2. One dep. variable (nominal/categorical)

In this section, we are going to return to an example from Section 1.3, the constructional alternation of particle placement in English, which is again represented in (30).

(30) a. He picked up the book. (verb - particle - direct object)
b. He picked the book up. (verb - direct object - particle)

As you already know, often both constructions are acceptable and native speakers can often not explain their preference for one of the two. One may therefore expect that both constructions are equally frequent, and this is what you are going to test. This scenario involves

- a dependent nominal/categorical variable CONSTRUCTION: *VERB-PARTICLE-OBJECT* vs. CONSTRUCTION: *VERB-OBJECT-PARTICLE*;
- no independent variable, because you do not investigate whether the distribution of CONSTRUCTION is dependent on anything else.

Such questions are generally investigated with tests from the family of chi-squared tests, which is one of the most important and widespread tests. Since there is no independent variable, you test the degree of fit between your observed and an expected distribution, which should remind you of Section 3.1.5.2. This test is referred to as the chi-squared goodness-of-fit test and involves the following steps:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Computing the frequencies you would expect given H_0
- Testing the assumption(s) of the test:
  - all observations are independent of each other
  - 80% of the expected frequencies are ≥ 5[17]
  - all expected frequencies are > 1
- Computing the contributions to chi-squared for all observed frequencies
- Computing the test statistic χ^2, *df*, and *p*

The first step is very easy here. As you know, H_0 typically postulates that the data are distributed randomly/evenly, and that means that both constructions occur equally often, i.e., 50% of the time (just as tossing a fair coin many times will result in a largely equal distribution). Thus:

H_0: The frequencies of the two variable levels of CONSTRUCTION are identical – if you find a difference in your sample, this difference is just random variation; $n_{\text{V Part DO}} = n_{\text{V DO Part}}$.

H_1: The frequencies of the two variable levels of CONSTRUCTION are not identical; $n_{\text{V Part DO}} \neq n_{\text{V DO Part}}$.

Note that this is a two-tailed H_1; no direction of the difference is provided. Next, you would collect some data and count the occurrences of both constructions, but we will abbreviate this step and use frequencies reported in Peters (2001). She conducted an experiment in which subjects described pictures and obtained the construction frequencies represented in Table 19.

Table 19. Observed construction frequencies of Peters (2001)

Verb - Particle - Direct Object	Verb - Direct Object - Particle
247	150

17. This threshold value of 5 is the one most commonly mentioned. There are a few studies that show that the chi-squared test is fairly robust even if this assumption is violated – especially when, as is here the case, H_0 postulates that the expected frequencies are equally high (cf. Zar 1999: 470). However, to keep things simple, I stick to the most common conservative threshold value of 5 and refer you to the literature quoted in Zar. If your data violate this assumption, then you must compute a binomial test (if, as here, you have two groups) or a multinomial test (for three or more groups); cf. the recommendations for further study.

Obviously, there is a strong preference for the construction in which the particle follows the verb directly. At first glance, it seems very unlikely that H_0 could be correct, given these data.

One very important side remark here: beginners often look at something like Table 19 and say, oh, ok, we have interval/ratio data: 247 and 150. Why is this wrong?

THINK
BREAK

It's wrong because Table 19 does not show you the raw data – what it shows you is already a numerical summary. You don't have interval/ratio data – you have an interval/ratio summary of categorical data, because the numbers 247 and 150 summarize the frequencies of the two levels of the categorical variable CONSTRUCTION (which you probably obtained from applying `table` to a vector/factor). One strategy to not mix this up is to always conceptually envisage what the raw data table would look like in the case-by-variable format discussed in Section 1.3.3. In this case, it would look like this:

Table 20. The case-by-variable version of the data in Table 19

CASE	CONSTRUCTION
1	vpo
2	vpo
247	vpo
248	vop
	vop
397	vop

From this format, it is quite obvious that the variable CONSTRUCTION is categorical. So, don't mix up interval/ratio summaries of categorical data with interval/ratio data.

As the first step of our evaluation, you should now have a look at a graphical representation of the data. A first possibility would be to generate, say, a dot chart. Thus, you first enter the two frequencies – first the frequency data, then the names of the frequency data (for the plotting) – and then you create a dot chart or a bar plot as follows:

```
> VPCs<-c(247, 150) # VPCs="verb-particle constructions"¶
> names(VPCs)<-c("V-Part-DO", "V-DO-Part")¶
> dotchart(VPCs, xlim=c(0, 250))¶
> barplot(VPCs)¶
```

The question now of course is whether this preference is statistically significant or whether it could just as well have arisen by chance. According to the above procedure, you must now compute the frequencies that follow from H_0. In this case, this is easy: since there are altogether 247+150 = 397 constructions, which should be made up of two equally large groups, you divide 397 by 2:

```
> VPCs.exp<-rep(sum(VPCs)/length(VPCs), length(VPCs))¶
> VPCs.exp¶
[1] 198.5 198.5
```

You must now check whether you can actually do a chi-squared test here, but the observed frequencies are obviously larger than 5 and we assume that Peters's data points are in fact independent (because we will assume that each construction has been provided by a different speaker). We can therefore proceed with the chi-squared test, the computation of which is fairly straightforward and summarized in (31).

(31) $$\text{Pearson chi-squared} = \chi^2 = \sum_{i=1}^{n} \frac{\left(observed - expected\right)^2}{expected}$$

That is to say, for every value of your frequency table you compute a so-called contribution to chi-squared by (i) computing the difference between the observed and the expected frequency, (ii) squaring this difference, and (iii) dividing that by the expected frequency again. The sum of these contributions to chi-squared is the test statistic chi-squared. Here, it is approximately 23.7.

(32) $$\text{Pearson } \chi^2 = \frac{\left(247 - 198.5\right)^2}{198.5} + \frac{\left(150 - 198.5\right)^2}{198.5} \approx 23.7$$

```
> sum(((VPCs-VPCs.exp)^2)/VPCs.exp)¶
[1] 23.70025
```

Obviously, this value increases as the differences between observed and

expected frequencies increase (because then the numerators become larger). That also means that chi-squared becomes 0 when all observed frequencies correspond to all expected frequencies: then the numerators become 0. Thus, we can simplify our statistical hypotheses to the following:

H_0: $\chi^2 = 0$.
H_1: $\chi^2 > 0$.

But the chi-squared value alone does not show you whether the differences are large enough to be statistically significant. So, what do you do with this value? Before computers became more widespread, a chi-squared value was used to look up whether the result is significant or not in a chi-squared table. Such tables typically have the three standard significance levels in the columns and different numbers of degrees of freedom (*df*) in the rows. *Df* here is the number of categories minus 1, i.e., *df* = 2-1 = 1, because when we have two categories, then one category frequency can vary freely but the other is fixed (so that we can get the observed number of elements, here 397). Table 21 is one such chi-squared table for the three significance levels and *df* = 1 to 3.

Table 21. Critical χ^2-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $1 \leq df \leq 3$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 1$	3.841	6.635	10.828
$df = 2$	5.991	9.21	13.816
$df = 3$	7.815	11.345	16.266

You can actually generate those values yourself with the function `qchisq`. That function requires three arguments:

- `p`: the *p*-value(s) for which you need the critical chi-squared values (for some *df*);
- `df`: the *df*-value(s) for the *p*-value for which you need the critical chi-squared value;
- `lower.tail=FALSE`: the argument to instruct R to only use the area under the chi-squared distribution curve that is to the right of / larger than the observed chi-squared value.

```
> qchisq(c(0.05, 0.01, 0.001), 1, lower.tail=FALSE)¶
[1]  3.841459  6.634897 10.827566
```

More advanced users find code to generate all of Table 21 in the code file. Once you have such a table, you can test your observed chi-squared value for significance by determining whether it is larger than the chi-squared value(s) tabulated at the observed number of degrees of freedom. You begin with the smallest tabulated chi-squared value and compare your observed chi-squared value with it and continue to do so as long as your observed value is larger than the tabulated ones. Here, you first check whether the observed chi-squared is significant at the level of 5%, which is obviously the case: 23.7 > 3.841. Thus, you can check whether it is also significant at the level of 1%, which again is the case: 23.7 > 6.635. Thus, you can finally even check if the observed chi-squared value is maybe even highly significant, and again this is so: 23.7 > 10.827. You can therefore reject H_0 and the usual way this is reported in your results section is this: "According to a chi-squared goodness-of-fit test, the frequency distribution of the two verb-particle constructions deviates highly significantly from the expected one (χ^2 = 23.7; df = 1; $p_{\text{two-tailed}} < 0.001$): the construction where the particle follows the verb directly was observed 247 times although it was only expected 199 times, and the construction where the particle follows the direct objet was observed only 150 times although it was expected 199 times."

With larger and more complex amounts of data, this semi-manual way of computation becomes more cumbersome (and error-prone), which is why we will simplify all this a bit. First, you can of course compute the *p*-value directly from the chi-squared value using the mirror function of `qchisq`, viz. `pchisq`, which requires the above three arguments:

```
> pchisq(23.7, 1, lower.tail=FALSE)¶
[1] 1.125825e-06
```

As you can see, the level of significance we obtained from our stepwise comparison using Table 21 is confirmed: *p* is indeed much smaller than 0.001, namely 0.00000125825. However, there is another even easier way: why not just do the whole test with one function? The function is called `chisq.test`, and in the present case it requires maximally three arguments:

- `x`: a vector with the observed frequencies;
- `p`: a vector with the expected percentages (not the frequencies!);
- `correct=TRUE` or `correct=FALSE`: when the sample size *n* is small ($15 \leq n \leq 60$), it is sometimes recommended to apply a so-called continuity

correction (after Yates); `correct=TRUE` is the default setting.[18]

In this case, this is easy: you already have a vector with the observed frequencies, the sample size n is much larger than 60, and the expected probabilities result from H_0. Since H_0 says the constructions are equally frequent and since there are just two constructions, the vector of the expected probabilities contains two times $^1/_2 = 0.5$. Thus:

```
> chisq.test(VPCs, p=c(0.5, 0.5))¶
 Chi-squared test for given probabilities
data:  VPCs
X-squared = 23.7003, df = 1, p-value = 1.126e-06
```

You get the same result as from the manual computation but this time you immediately also get a p-value. What you do not also get are the expected frequencies, but these can be obtained very easily, too. The function `chisq.test` computes more than it returns. It returns a data structure (a so-called list) so you can assign a name to this list and then inspect it for its contents (output not shown):

```
> test<-chisq.test(VPCs, p=c(0.5, 0.5))¶
> str(test)¶
```

Thus, if you require the expected frequencies, you just retrieve them with a $ and the name of the list component you want, and of course you get the result you already know.

```
> test$expected¶
[1] 198.5 198.5
```

Let me finally mention that the above method computes a p-value for a two-tailed test. There are many tests in R where you can define whether you want a one-tailed or a two-tailed test. However, this does not work with the chi-squared test. If you require the critical chi-squared value for $p_{\text{one-tailed}} = 0.05$ for $df = 1$, then you must compute the critical chi-squared value for $p_{\text{two-tailed}} = 0.1$ for $df = 1$ (with `qchisq(0.1, 1, lower.tail=FALSE)¶`), since your prior knowledge is rewarded such that a less extreme result in the predicted direction will be sufficient (cf. Section 1.3.4). Also, this means that when you need the $p_{\text{one-tailed}}$-value for a chi-square value, just take half of the $p_{\text{two-tailed}}$-value of the same chi-square value. In this

18. For further options, cf. `?chisq.test¶`, `formals(chisq.test)¶` or `args(chisq.test)¶`.

case, if your H_1 had been directional, this would have been your *p*-value. But again: this works only with $df = 1$.

```
> pchisq(23.7, 1, lower.tail=FALSE)/2¶
```

Warning/advice

Above I warned you to never change your hypotheses *after* you have obtained your results and then sell your study as successful support of the 'new' H_1. The same logic does not allow you to change your hypothesis from a two-tailed one to a one-tailed one because your $p_{\text{two-tailed}} = 0.08$ (i.e., non-significant) so that the corresponding $p_{\text{one-tailed}} = 0.04$ (i.e., significant). Your choice of a one-tailed hypothesis must be motivated *conceptually*.

Another hugely important warning: never ever compute a chi-square test like the above on percentages – always on 'real' observed frequencies!

Recommendation(s) for further study

- the functions `binom.test` or `dbinom` to compute binomial tests
- the function `prop.test` (cf. Section 3.1.5.2) to test relative frequencies / percentages for deviations from expected frequencies / percentages
- the function `dmultinom` to help compute multinomial tests
- Baayen (2008: Section 4.1.1), Sheskin (2011: Test 8, 9)

1.2. Tests for differences/independence

In Section 4.1.1, we looked at goodness-of-fit tests for distributions and frequencies – now we turn to tests for differences/independence.

1.2.1. One dep. variable (ordinal/interval/ratio scaled) and one indep. variable (nominal) (indep. samples)

Let us now look at an example in which two independent samples are compared with regard to their overall distributions. You will test whether men and women differ with regard to the frequencies of hedges they use in discourse (i.e., expressions such as *kind of* or *sort of*). Again, note that we are here only concerned with the overall distributions – not just means or just variances. We could of course do that, too, but it is of course possible that the means are very similar while the variances are not and a test for differ-

ent means might not uncover the overall distributional difference.

Let us assume you have recorded 60 two-minute conversations between a confederate of an experimenter, each with one of 30 men and 30 women, and then counted the numbers of hedges that the male and female subjects produced. You now want to test whether the distributions of hedge frequencies differs between men and women. This question involves

- an independent nominal/categorical variable, SEX: *MALE* and SEX: *FEMALE*;
- a dependent interval/ratio-scaled: the number of hedges produced: HEDGES.

The question of whether the two sexes differ in terms of the distributions of hedge frequencies is investigated with the two-sample Kolmogorov-Smirnov test (again, check <sflwr_navigator.png>):

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test: the data are continuous
- Computing the cumulative frequency distributions for both samples, the maximal absolute difference D of both distributions, and p

First the hypotheses: the text form is straightforward and the statistical version is based on a test statistic called D to be explained below

H_0: The distribution of the dependent variable HEDGES does not differ depending on the levels of the independent variable SEX; $D = 0$.

H_1: The distribution of the dependent variable HEDGES differs depending on the levels of the independent variable SEX; $D > 0$.

Before we do the actual test, let us again inspect the data graphically. You first load the data from <_inputfiles/04-1-2-1_hedges.csv>, check the data structure (I will usually not show that output here in the book), and make the variable names available.

```
> Hedges<-read.delim(file.choose())¶
> str(Hedges)¶
> attach(Hedges)¶
```

You are interested in the general distribution, so one plot you can create is a stripchart. In this kind of plot, the frequencies of hedges are plotted separately for each sex, but to avoid that identical frequencies are plotted directly onto each other (and can therefore not be distinguished anymore), you also use the argument `method="jitter"` to add a tiny value to each data point, which decreases the chance of overplotted data points (also try `method="stack"`). Then, you include the meaningful point of $x = 0$ on the x-axis. Finally, with the function `rug` you add little bars to the x-axis (`side=1`) which also get jittered. The result is shown in Figure 40.

```
> stripchart(HEDGES~SEX, method="jitter", xlim=c(0, 25),
   xlab="Number of hedges", ylab="Sex")¶
> rug(jitter(HEDGES), side=1)¶
```

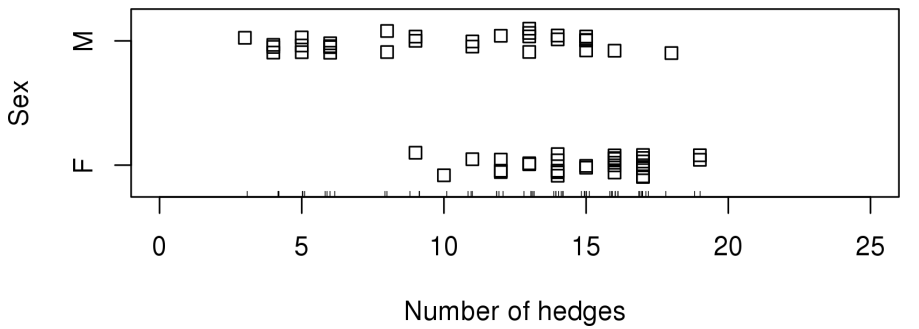

Figure 40. Stripchart for HEDGES~SEX

It is immediately obvious that the data are distributed quite differently: the values for women appear to be a little higher on average and more homogeneous than those of the men. The data for the men also appear to fall into two groups, a suspicion that also receives some *prima facie* support from the following two histograms in Figure 41. (Note that all axis limits are again defined identically to make the graphs easier to compare.)

```
> par(mfrow=c(1, 2))¶
> hist(HEDGES[SEX=="M"], xlim=c(0, 25), ylim=c(0, 10), ylab=
   "Frequency", main="")¶
> hist(HEDGES[SEX=="F"], xlim=c(0, 25), ylim=c(0, 10), ylab=
   "Frequency", main="")¶
> par(mfrow=c(1, 1))¶
```

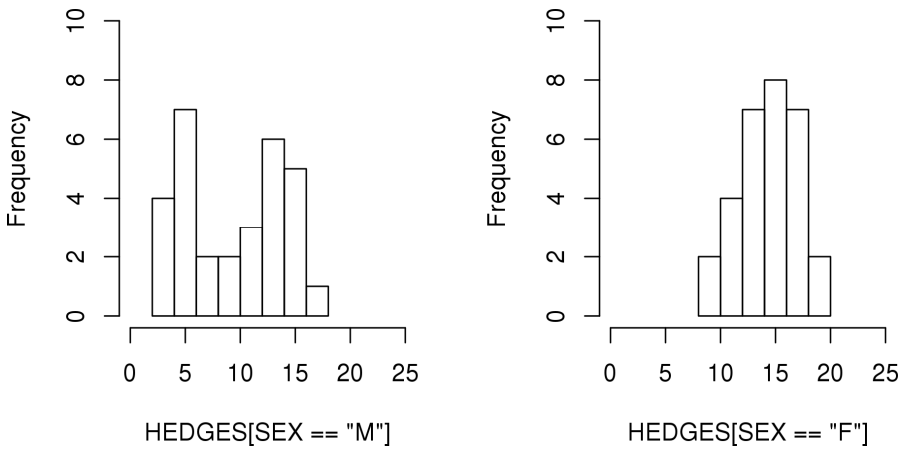

Figure 41. Histograms of the number of hedges by men and women

The assumption of continuous data points is not exactly met because frequencies are discrete – there are no frequencies 3.3, 3.4, etc. – but HEDGES spans quite a range of values and we could in fact jitter the values to avoid ties. To test these distributional differences with the Kolmogorov-Smirnov test, which involves the empirical cumulative distribution of the data, you first rank-order the data: You sort the values of SEX in the order in which you need to sort HEDGES, and then do the same to HEDGES itself:

```
> SEX<-SEX[order(HEDGES)]¶
> HEDGES<-HEDGES[order(HEDGES)]¶
```

The next step is a little more complex. You must now compute the maximum of all differences of the two cumulative distributions of the hedges. You can do this in three steps: First, you generate a frequency table with the numbers of hedges in the rows and the sexes in the columns. This table in turn serves as input to `prop.table`, which generates a table of column percentages (hence `margin=2`; cf. Section 3.2.1, output not shown):

```
> dists<-prop.table(table(HEDGES, SEX), margin=2); dists¶
```

This table shows that, say, 10% of all numbers of hedges of men are 4, but these are of course not cumulative percentages yet. The second step is therefore to convert these percentages into cumulative percentages. You can use `cumsum` to generate the cumulative percentages for both columns and can even compute the differences in the same line:

```
> differences<-cumsum(dists[,1])-cumsum(dists[,2])¶
```

That is, you subtract from every cumulative percentage of the first column (the values of the women) the corresponding value of the second column (the values of the men). The third and final step is then to determine the maximal absolute difference, which is the test statistic *D*:

```
> max(abs(differences))¶
[1] 0.4666667
```

You can then look up this value in a table for Kolmogorov-Smirnov tests; for a significant result, the computed value must be larger than the tabulated one. For cases in which both samples are equally large, Table 22 shows the critical *D*-values for two-tailed Kolmogorov-Smirnov tests (computed from Sheskin 2011: Table A23).

Table 22. Critical *D*-values for two-sample Kolmogorov-Smirnov tests

	$p = 0.05$	$p = 0.01$
$n_1 = n_2 = 29$	0.3571535	0.428059
$n_1 = n_2 = 30$	0.3511505	0.4208642
$n_1 = n_2 = 31$	0.3454403	0.4140204

Our value of $D = 0.4667$ is not only significant ($D > 0.3511505$), but even very significant ($D > 0.4208642$). You can therefore reject H_0 and summarize the results: "According to a two-sample Kolmogorov-Smirnov test, there is a significant difference between the distributions of hedge frequencies of men and women: women seem to use more hedges and behave more homogeneously than the men, who use fewer hedges and whose data appear to fall into two groups ($D = 0.4667$, $p_{\text{two-tailed}} < 0.01$)."

The logic of this test is not always immediately clear but worth exploring. To that end, we look at a graphical representation. The following lines plot the two empirical cumulative distribution functions (`ecdf`) of men (in black) and women (in grey) as well as a vertical line at position $x = 9$, where the largest difference ($D = 0.4667$) was found. This graph in Figure 42 below shows what the Kolmogorov-Smirnov test reacts to: different empirical cumulative distributions.

```
> plot(ecdf(HEDGES[SEX=="M"]), do.points=TRUE, verticals=
   TRUE, main="Hedges: men (black) vs. women (grey)",
   xlab="Numbers of hedges")¶
> lines(ecdf(HEDGES[SEX=="F"]), do.points=TRUE, verticals=
```

```
    TRUE, col="darkgrey")¶
> abline(v=9, lty=2)¶
```

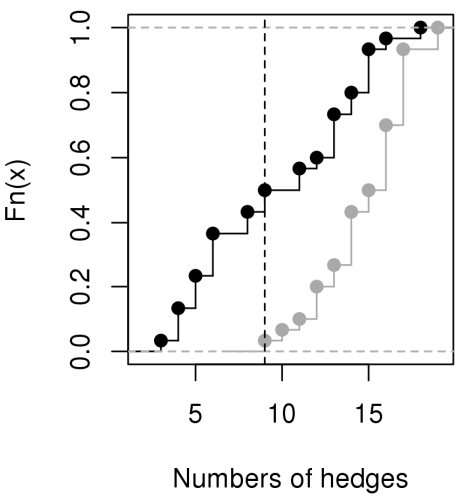

Figure 42. Empirical cumulative distribution functions of the numbers of hedges of men (black) and women (grey)

For example, the fact that the values of the women are higher and more homogeneous is indicated especially in the left part of the graph where the low hedge frequencies are located and where the values of the men already rise but those of the women do not. More than 40% of the values of the men are located in a range where no hedge frequencies for women were obtained at all. As a result, the largest difference at position $x = 9$ arises where the curve for the men has already risen considerably while the curve for the women has only just begun to take off. This graph also explains why H_0 postulates $D = 0$. If the curves are completely identical, there is no difference between them and D becomes 0.

The above explanation simplified things a bit. First, you do not always have two-tailed tests and identical sample sizes. Second, identical values – so-called *ties* – can complicate the computation of this test (and others). Fortunately, you do not really have to worry about any of this because the R function `ks.test` does everything for you in just one line. You just need the following arguments:[19]

– `x` and `y`: the two vectors whose distributions you want to compare;

19. Unfortunately, the function `ks.test` does not take a formula as input.

– `alternative="two-sided"` for two-tailed tests (the default) or `alternative="greater"` or `alternative="less"` for one-sided tests depending on which H_1 you want to test: the argument `alternative="…"` refers to the first-named vector so that `alternative="greater"` means that the cumulative distribution function of the first vector is above that of the second.

When you test a two-tailed H_1 as we do here, then the line to enter into R reduces to the following, and you get the same *D*-value and the *p*-value. (I omitted the warning about ties here but, again, you can use `jitter` to get rid of it; cf. the code file.)

```
> ks.test(HEDGES[SEX=="M"], HEDGES[SEX=="F"])¶
 Two-sample Kolmogorov-Smirnov test
data:  HEDGES[SEX == "M"] and HEDGES[SEX == "F"]
D = 0.4667, p-value = 0.002908
alternative hypothesis: two-sided
```

Recommendation(s) for further study

– apart from the function mentioned in the text (`plot(ecdf(…))`), you can create such graphs also with `plot.stepfun`
– Crawley (2005: 100f.), Crawley (2007: 316f.), Baayen (2008: Section 4.2.1), Sheskin (2011: Test 13)

1.2.2. One dep. variable (nominal/categorical) and one indep. variable (nominal/categorical) (indep. samples)

In Section 4.1.1.2 above, we discussed how you test whether the distribution of a dependent nominal/categorical variable is significantly different from another known distribution. A probably more frequent situation is that you test whether the distribution of one nominal/categorical variable is dependent on another nominal/categorical variable.

Above, we looked at the frequencies of the two verb-particle constructions. We found that their distribution was not compatible with H_0. However, we also saw earlier that there are many variables that are correlated with the constructional choice. One of these is whether the referent of the direct object is given information, i.e., known from the previous discourse, or not. Specifically, previous studies found that objects referring to given referents prefer the position before the particle whereas objects referring to new referents prefer the position after the particle. We will look at this hypothesis

(for the sake of simplicity as a two-tailed hypothesis). It involves

- a dependent nominal/categorical variable, namely CONSTRUCTION: *VERB-PARTICLE-OBJECT* vs. CONSTRUCTION: *VERB-OBJECT-PARTICLE*;
- an independent variable nominal/categorical variable, namely the givenness of the referent of the direct object: GIVENNESS: *GIVEN* vs. GIVENNESS: *NEW*;
- independent samples because we will assume that, in the data below, the fact any particular constructional choice is unrelated to any other one (this is often far from obvious, but too complex to be discussed here in more detail).

As before, such questions are investigated with chi-squared tests: you test whether the levels of the independent variable result in different frequencies of the levels of the dependent variable. The overall procedure for a chi-squared test for independence is very similar to that of a chi-squared test for goodness of fit, but you will see below that the computation of the expected frequencies is (only superficially) a bit different from above.

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Computing the frequencies you would expect given H_0
- Testing the assumption(s) of the test:
  - all observations are independent of each other
  - 80% of the expected frequencies are ≥ 5 (cf. n. 17)
  - all expected frequencies are > 1
- Computing the contributions to chi-squared for all observed frequencies
- Computing the test statistic χ^2, *df*, and *p*

The hypotheses are simple, especially since we apply what we learned from the chi-squared test for goodness of fit from above:

H_0: The frequencies of the levels of the dependent variable CONSTRUCTION do not vary as a function of the levels of the independent variable GIVENNESS; $\chi^2 = 0$.

H_1: The frequencies of the levels of the dependent variable CONSTRUCTION vary as a function of the levels of the independent variable GIVENNESS; $\chi^2 > 0$.

In order to discuss this version of the chi-squared test, we return to the data from Peters (2001). As a matter of fact, the above discussion did not utilize all of Peters's data because I omitted an independent variable, namely GIVENNESS. Peters (2001) did not just study the frequency of the two constructions – she studied what we are going to look at here, namely whether GIVENNESS is correlated with CONSTRUCTION. In the picture-description experiment described above, she manipulated the variable GIVENNESS and obtained the already familiar 397 verb-particle constructions, which patterned as represented in Table 23. (By the way, the cells of such 2-by-2 tables are often referred to with the letters *a* to *d*, *a* being the top left cell (85), *b* being the top right cell (65), etc.)

Table 23. Observed construction frequencies of Peters (2001)

	GIVENNESS: *GIVEN*	GIVENNESS: *NEW*	Row totals
CONSTRUCTION: *V DO PART*	85	65	150
CONSTRUCTION: *V PART DO*	100	147	247
Column totals	185	212	397

First, we explore the data graphically. You load the data from <_inputfiles/04-1-2-2_vpcs.csv>, create a table of the two factors, and get a first visual impression of the distribution of the data (cf. Figure 43).

```
> VPCs<-read.delim(file.choose())¶
> str(VPCs); attach(VPCs)¶
> Peters.2001<-table(CONSTRUCTION, GIVENNESS)¶
> plot(CONSTRUCTION~GIVENNESS)¶
```

Obviously, the differently-colored areas are differently big between rows/columns. To test these differences for significance, we need the frequencies expected from H_0. But how do we compute the frequencies predicted by H_0? Since this is a central question, we will discuss this in detail.

Let us assume Peters had obtained the totals in Table 24. What would the distribution following from H_0 look like? Above in Section 4.1.1.2, we said that H_0 typically postulates equal frequencies. Thus, you might assume – correctly – that the expected frequencies are those represented in Table 24. All marginal totals are 100 and every variable has two equally frequent levels so we have 50 in each cell.

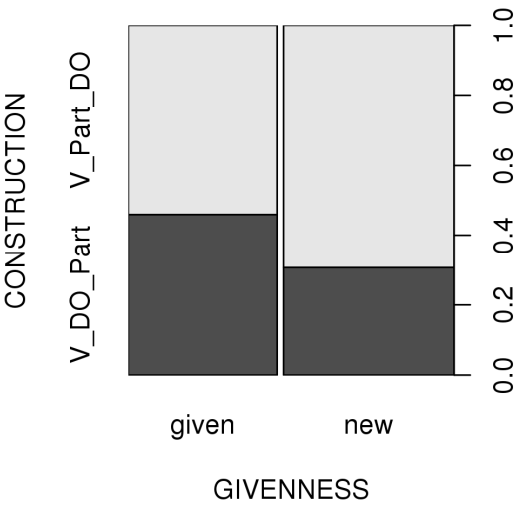

Figure 43. Mosaic plot for CONSTRUCTION~GIVENNESS

Table 24. Fictitious observed construction frequencies of Peters (2001)

	GIVENNESS: *GIVEN*	GIVENNESS: *NEW*	Row totals
CONSTRUCTION: *V DO PART*			100
CONSTRUCTION: *V PART DO*			100
Column totals	100	100	200

Table 25. Fictitious expected construction frequencies of Peters (2001)

	GIVENNESS: *GIVEN*	GIVENNESS: *NEW*	Row totals
CONSTRUCTION: *V DO PART*	50	50	100
CONSTRUCTION: *V PART DO*	50	50	100
Column totals	100	100	200

The statistical hypotheses that go beyond just stating whether or not $\chi^2 = 0$ would then be:

H_0: $n_{\text{V DO Part \& Ref DO = given}} = n_{\text{V DO Part \& Ref DO} \neq \text{given}} = n_{\text{V Part DO \& Ref DO = given}} = n_{\text{V Part DO \& Ref DO} \neq \text{given}}$

H_1: as H_0, but there is at least one "≠" instead of an "=".

However, life is usually not that simple, for example when (a) as in Peters (2001) not all subjects answer all questions or (b) naturally-observed data are counted that are not as nicely balanced. Thus, in Peters's real data, it does not make sense to simply assume equal frequencies. Put differently, H_0 cannot look like Table 24 because the row totals of Table 23 show that the different levels of GIVENNESS are not equally frequent. If GIVENNESS had no influence on CONSTRUCTION, you would expect that the frequencies of the two constructions for each level of GIVENNESS would exactly reflect the frequencies of the two constructions in the whole sample. That means (i) all marginal totals (row/column totals) must remain constant (as they reflect the numbers of the investigated elements), and (ii) the proportions of the marginal totals determine the cell frequencies in each row and column. From this, a rather complex set of hypotheses follows:

H_0: $n_{\text{V DO Part \& Ref DO = given}} : n_{\text{V DO Part \& Ref DO} \neq \text{given}} \propto$
$n_{\text{V Part DO \& Ref DO = given}} : n_{\text{V Part DO \& Ref DO} \neq \text{given}} \propto$
$n_{\text{Ref DO = given}} : n_{\text{Ref DO} \neq \text{given}}$ and
$n_{\text{V DO Part \& Ref DO = given}} : n_{\text{V Part DO \& Ref DO = given}} \propto$
$n_{\text{V DO Part \& Ref DO} \neq \text{given}} : n_{\text{V Part DO \& Ref DO} \neq \text{given}} \propto$
$n_{\text{V DO Part}} : n_{\text{V Part DO}}$

H_1: as H_0, but there is at least one "≠" instead of an "=".

In other words, you cannot simply say, "there are 2·2 = 4 cells and I assume each expected frequency is 397 divided by 4, i.e., approximately 100." If you did that, the upper row total would amount to nearly 200 – but that can't be right since there are only 150 cases of CONSTRUCTION: *VERB-OBJECT-PARTICLE*. Thus, you must include this information, that there are only 150 cases of CONSTRUCTION: *VERB-OBJECT-PARTICLE*, into the computation of the expected frequencies. The easiest way to do this is using percentages: there are $^{150}/_{397}$ cases of CONSTRUCTION: *VERB-OBJECT-PARTICLE* (i.e. 0.3778 = 37.78%). Then, there are $^{185}/_{397}$ cases of GIVENNESS: *GIVEN* (i.e., 0.466 = 46.6%). If the two variables are independent of each other, then the probability of their joint occurrence is 0.3778·0.466 = 0.1761. Since there are altogether 397 cases to which this probability applies, the expected frequency for this combination of variable levels is 397·0.1761 = 69.91. This logic can be reduced to (33).

(33) $$n_{\text{expected cell frequency}} = \frac{row\ sum \cdot column\ sum}{n}$$

If you apply this logic to every cell, you get Table 26.

Table 26. Expected construction frequencies of Peters (2001)

	GIVENNESS: *GIVEN*	GIVENNESS: *NEW*	Row totals
CONSTRUCTION: *V DO PART*	69.9	80.1	150
CONSTRUCTION: *V PART DO*	115.1	131.9	247
Column totals	185	212	397

You can immediately see that this table corresponds to the above H_0: the ratios of the values in each row and column are exactly those of the row totals and column totals respectively. For example, the ratio of 69.9 to 80.1 to 150 is the same as that of 115.1 to 131.9 to 247 and as that of 185 to 212 to 397, and the same is true in the other dimension. Thus, H_0 is not "all cell frequencies are identical" – it is "the ratios of the cell frequencies are equal (to each other and the respective marginal totals)."

This method to compute expected frequencies can be extended to arbitrarily complex frequency tables (see Gries 2009b: Section 5.1). But how do we test whether these deviate strongly enough from the observed frequencies? Thankfully, we do not need such complicated hypotheses but can use the simpler versions of $\chi^2 = 0$ and $\chi^2 > 0$ used above, and the chi-squared test for independence is identical to the chi-squared goodness-of-fit test you already know: for each cell, you compute a contribution to chi-squared and sum those up to get the chi-squared test statistic.

As before, the chi-squared test can only be used when its assumptions are met. The expected frequencies are large enough and for simplicity's sake we assume here that every subject only gave just one sentence so that the observations are independent of each other: for example, the fact that some subject produced a particular sentence on one occasion does then not affect any other subject's formulation. We can therefore proceed as above and compute (the sum of) the contributions to chi-squared on the basis of the same formula, here repeated as (34):

$$\text{(34)} \qquad \text{Pearson } \chi^2 = \sum_{i=1}^{n} \frac{\left(observed - expected\right)^2}{expected}$$

The results are shown in Table 27 and the sum of all contributions to chi-squared, chi-squared itself, is 9.82. However, we again need the num-

ber of degrees of freedom. For two-dimensional tables and when the expected frequencies are computed on the basis of the observed frequencies as here, the number of degrees of freedom is computed as shown in (35).[20]

Table 27. Contributions to chi-squared for the data of Peters (2001)

	GIVENNESS: *GIVEN*	GIVENNESS: *NEW*	Row totals
CONSTRUCTION: *V DO PART*	3.26	2.85	
CONSTRUCTION: *V PART DO*	1.98	1.73	
Column totals			9.82

(35) df = (no. of rows-1) · (no. of columns-1) = (2-1)·(2-1) = 1

With both the chi-squared and the *df-val*ue, you can look up the result in a chi-squared table (e.g., Table 28 below, which is the same as Table 21). As above, if the observed chi-squared value is larger than the one tabulated for $p = 0.05$ at the required *df-val*ue, then you can reject H_0. Here, chi-squared is not only larger than the critical value for $p = 0.05$ and $df = 1$, but also larger than the critical value for $p = 0.01$ and $df = 1$. But, since the chi-squared value is not also larger than 10.827, the actual p-value is somewhere between 0.01 and 0.001: the result is very, but not highly significant.

Table 28. Critical χ^2-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $1 \leq df \leq 3$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 1$	3.841	6.635	10.828
$df = 2$	5.991	9.21	13.816
$df = 3$	7.815	11.345	16.266

Fortunately, all this is much easier when you use R's built-in function. Either you compute just the p-value as before,

```
> pchisq(9.82, 1, lower.tail=FALSE)¶
[1] 0.001726243
```

20. In our example, the expected frequencies were computed from the observed frequencies in the marginal totals. If you compute the expected frequencies not from your observed data but from some other distribution, the computation of *df* changes to: *df* = (number of rows · number of columns)-1.

or you use the function `chisq.test` and do everything in a single step. The most important arguments for our purposes are:

- `x`: the two-dimensional table for which you do a chi-squared test;
- `correct=TRUE` or `correct=FALSE`; cf. above for the correction.[21]

```
> test.Peters<-chisq.test(Peters.2001, correct=FALSE)¶
> test.Peters¶
 Pearson's Chi-squared test
data:  Peters.2001
X-squared = 9.8191, df = 1, p-value = 0.001727
```

This is how you obtain expected frequencies or the chi-squared value:

```
> test.Peters$expected¶
            GIVENNESS
CONSTRUCTION     given       new
   V_DO_Part  69.89924  80.10076
   V_Part_DO 115.10076 131.89924
> test.Peters$statistic¶
X-squared
 9.819132
```

You now know that GIVENNESS is correlated with CONSTRUCTION, but you neither know yet how strong that effect is nor which variable level combinations are responsible for this result. As for the effect size, even though you might be tempted to use the size of the chi-squared value or the *p*-value to quantify the effect, you must not do that. This is because the chi-squared value is dependent on the sample size, as we can easily see:

```
> chisq.test(Peters.2001*10, correct=FALSE)¶
 Pearson's Chi-squared test
data:  Peters.2001 * 10
X-squared = 98.1913, df = 1, p-value < 2.2e-16
```

For effect sizes, this is of course a disadvantage since just because the sample size is larger, this does not mean that the relation of the values to each other has changed, too. You can easily verify this by noticing that the ratios of percentages, for example, have stayed the same. For that reason, the effect size is often quantified with a coefficient of correlation (called ϕ in the case of $k\times2/m\times2$ tables or Cramer's V for $k\times m$ tables with k or $m >$

21. For further options, cf. again `?chisq.test¶`. Note also what happens when you enter `summary(Peters.2001)¶`.

2), which falls into the range between 0 and 1 (0 = no correlation; 1 = perfect correlation) and is unaffected by the sample size. ϕ / Cramer's V is computed according to the formula in (36):

(36) ϕ / Cramer's V / Cramer's index I =

$$\sqrt{\frac{\chi^2}{n \cdot (\min[n_{rows}, n_{columns}] - 1)}}$$

In R, you can of course do this in one line of code:

```
> sqrt(test.Peters$statistic/
   sum(Peters.2001)*(min(dim(Peters.2001))-1))¶
X-squared
0.1572683
```

Given the theoretical range of values, this is a rather small effect size.[22] The correlation is probably not random, but also not strong.

Another measure of effect size, which can however only be applied to 2×2 tables, is the so-called odds ratio. An *odds ratio* tells you how the likelihood of one variable level changes in response to a change of the other variable's level. The *odds* of an event E correspond to the fraction in (37).

(37) $odds = \frac{p_E}{1 - p_E}$ (you get probabilities from odds with $\frac{odds}{1 + odds}$)

The odds ratio for a 2×2 table such as Table 23 is the ratio of the two odds (or 1 divided by that ratio, depending on whether you look at the event E or the event $\neg E$ (not E)), as in (38):

(38) *odds ratio* for Table 23 = $\frac{85}{100} \Big/ \frac{65}{147} = 1.9223$

In words, the odds of CONSTRUCTION: *V DO PART* are $(^{85}/_{185}) / (1-^{85}/_{185}) = {}^{85}/_{100} = 0.85$ when the referent of the direct object is given and $(^{65}/_{212}) / (1-{}^{65}/_{212}) = {}^{65}/_{147} = 0.4422$ when the referent of the direct object is new. This in

22. The theoretical range from 0 to 1 is really only possible in particular situations, but still a good heuristic to interpret this value.

turn means that CONSTRUCTION: *V DO PART* is $^{0.85}/_{0.4422} \approx 1.9223$ times more likely when the referent of the direct object is given than when it is not. From this, it also follows that the odds ratio in the absence of an interaction is ≈ 1.[23]

Table 27 also shows which variable level combinations contribute most to the significant correlation: the larger the contribution to chi-squared of a cell, the more that cell contributes to the overall chi-squared value; in our example, these values are all rather small – none exceeds the chi-squared value for $p = 0.05$ and $df = 1$, i.e., 3.841. In R, you can get the contributions to chi-squared as follows:

```
> test.Peters$residuals^2¶
             GIVENNESS
CONSTRUCTION    given      new
   V_DO_Part 3.262307 2.846825
   V_Part_DO 1.981158 1.728841
```

That is, you square the Pearson residuals. The Pearson residuals, which you obtain as follows, reveal the direction of effect for each cell: negative and positive values mean that observed values are smaller and larger than the expected values respectively.

```
> test.Peterst$residuals¶
             GIVENNESS
CONSTRUCTION     given       new
   V_DO_Part  1.806186 -1.687254
   V_Part_DO -1.407536  1.314854
```

Thus, if, given the small contributions to chi-square, one wanted to draw any further conclusions at all, then one could only say that the variable level combination contributing most to the significant result is the combination of CONSTRUCTION: *V DO PART* and GIVENNESS: *GIVEN*, which is more often observed than expected, but the individual cells' effects here are really rather small.

An interesting and revealing graphical representation is available with the function `assocplot`, whose most relevant argument is the two-

23. Often, you may find the logarithm of the odds ratio (see especially Section 5.3). When the two variables are not correlated, this log of the *odds ratio* is log 1 = 0, and positive/negative correlations result in positive/negative log odds ratios, which is often a little easier to interpret. For example, if you have two odds ratios such as $odds\ ratio_1 = 0.5$ and $odds\ ratio_2 = 1.5$, then you cannot immediately and intuitively see, which effect is larger. The logs of the odds ratios – $\log odds\ ratio_1 = -0.693$ and $\log odds\ ratio_2 = 0.405$ – tell you immediately the former is larger because it is further away from 0.

dimensional table under investigation: In this plot (Figure 44), "the area of the box is proportional to the difference in observed and expected frequencies." The black rectangles above the dashed lines indicate observed frequencies exceeding expected frequencies; grey rectangles below the dashed lines indicate observed frequencies smaller than expected frequencies; the heights of the boxes are proportional to the above Pearson residuals and the widths are proportional to the square roots of the expected frequencies. Note I do not just plot the table, but the transposed table – that's what the `t()` does. This is so that the row/column organization of the plot corresponds to that of the original table:

```
> assocplot(t(Peters.2001), col=c("black", "darkgrey"))¶
```

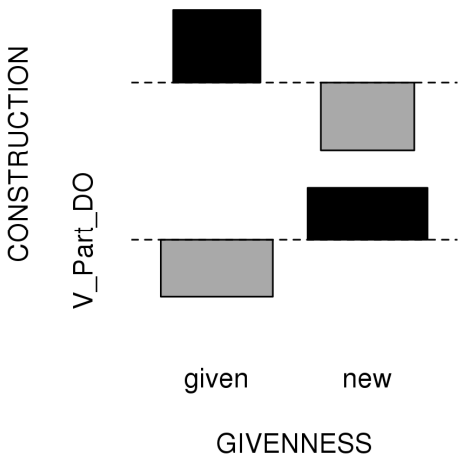

Figure 44. Association plot for CONSTRUCTION~GIVENNESS

Another interesting way to look at the data is a mixture between a plot and a table. The table/graph in Figure 45 has the same structure as Table 23, but (i) the sizes in which the numbers are plotted directly reflects the size of the residuals (i.e., bigger numbers deviate more from the expected frequencies than smaller numbers, where *bigger* and *smaller* are to be understood in terms of plotting size), and (ii) the coloring and the signs indicates how the observed frequencies deviate from the expected ones: black indicates positive residuals and grey indicates negative residuals. (For lack of a better term, I refer to this as a cross-tabulation plot.)

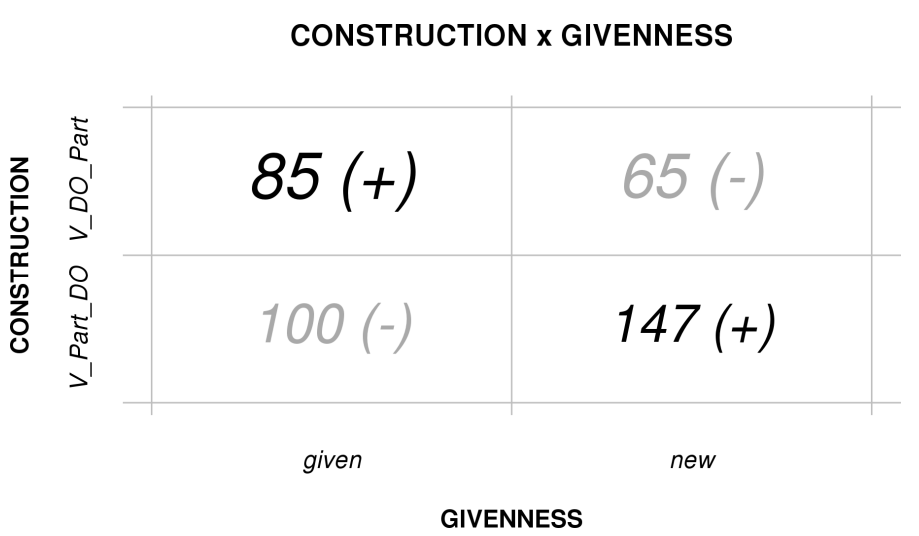

Figure 45. Cross-tabulation plot for CONSTRUCTION~GIVENNESS

This is how you would summarize all the results: "New objects are strongly preferred in the construction Verb-Particle-Direct Object and are dispreferred in Verb-Direct Object-Particle. The opposite kind of constructional preference is found for given objects. According to a chi-squared test for independence, this correlation is very significant ($\chi^2 = 9.82$; $df = 1$; $p_{\text{two-tailed}} < 0.002$), but the effect is not particularly strong ($\phi = 0.157$, odds ratio = 1.9223).

Let me finally emphasize that the above procedure is again the one providing you with a *p*-value for a two-tailed test. In the case of 2×2 tables, you can perform a one-tailed test as discussed in Section 4.1.1.2 above, but you cannot do one-tailed tests for tables with $df > 1$.

Recommendation(s) for further study

- the function `dotchart` as well as `mosaic` (from the library `vcd`) and `table.cont` (from the library `ade4`) for other kinds of plots
- the function `assocstats` (from the library `vcd`) for a different way to compute chi-square tests and effect sizes at the same time
- the function `CrossTable` (from the library `gmodels`) for more comprehensive tables
- the argument `simulate.p.value=TRUE` of the function `chisq.test` and the function `fisher.test`, which you can use when the expected frequencies are too small for a regular chi-squared test

- the Marascuilo procedure to test which observed row or column frequencies are different from each other in pairwise tests (cf. Gries to appear, who also discusses how to test a subtable out of a larger table)
- Crawley (2005: 85ff.), Crawley (2007: 301ff.), Sheskin (2011: Test 16)

Warning/advice
Again: never ever compute a chi-squared test on percentages – always on 'real' observed frequencies! (Trust me, there is a reason I repeat this …)

Let me mention one additional useful application of the chi-squared test (from Zar 1999: Section 23.4 and Sheskin 2011: 691ff.). Sometimes, you may have several isomorphic 2×2 tables on the same phenomenon, maybe because you found another source that discusses the same kind of data. You may then want to know whether or not the data are so similar that you can actually merge or amalgamate the data into one single data set. Here are the text hypotheses for that kind of question:

H_0: The trends in the different data sets do not differ from each other: $\chi^2_{\text{heterogeneity}} = 0$.
H_1: The trends in the different data sets differ from each other: $\chi^2_{\text{heterogeneity}} \neq 0$.

To explore this approach, let us compare Peters's data to those of Gries (2003a). You can enter the latter into R directly using the function `matrix`, which needs the vector of observed frequencies (columnwise), the number of columns, and the names of the dimensions (first rows, then columns):

```
> Gries.2003<-matrix(c(143, 53, 66, 141), ncol=2,
   dimnames=list(CONSTRUCTION=c("V_DO_Part",  "V_Part_DO"),
   GIVENNESS=c("given", "new")))¶
> Gries.2003¶
          given new
V_DO_Part   143  66
V_Part_DO    53 141
```

On the one hand, these data look very different from those of Peters (2001) because, here, when GIVENNESS is *GIVEN*, then CONSTRUCTION: *V_DO_PART* is nearly three times as frequent as CONSTRUCTION: *V_PART_DO* (and not in fact less frequent, as in Peters's data). On the other hand, the data are also similar because in both cases given direct objects increase the likelihood of CONSTRUCTION: *V_DO_PART*. A direct compari-

son of the association plots (not shown here, but you can use the following code to generate them) makes the data seem very much alike – how much more similar could two association plots be?

```
> par(mfrow=c(1, 2))¶
> assocplot(t(Peters.2001))¶
> assocplot(t(Gries.2003))¶
> par(mfrow=c(1, 1))¶
```

However, you should not really compare the sizes of the boxes in association plots – only the overall tendencies – so we turn to the heterogeneity chi-squared test. The heterogeneity chi-squared value is computed as the difference between the sum of chi-squared values of the original tables and the chi-squared value for the merged tables (that's why they have to be isomorphic), and it is evaluated with a number of degrees of freedom that is the difference between the sum of the degrees of freedom of all merged tables and the degrees of freedom of the merged table. Sounds pretty complex, but in fact it is not. The following code should make everything clear. First, you compute the chi-squared test for the data from Gries (2003a):

```
> test.Gries<-chisq.test(Gries.2003, correct=FALSE)¶
> test.Gries¶
  Pearson's Chi-squared test
data:  Gries.2003
X-squared = 68.0364, df = 1, p-value < 2.2e-16
```

Then you compute the sum of chi-squared values of the original tables:

```
> test.Peters$statistic+test.Gries$statistic¶
X-squared
[1] 77.85552
```

After that, you compute the chi-squared value of the combined table …

```
> chisq.test(Peters.2001+Gries.2003,
   correct=FALSE)$statistic¶
X-squared
[1] 65.87908
```

… and then the heterogeneity chi-squared and its degrees of freedom (you get the *df*-values with `$parameter`):

```
> het.chisq<-77.85552-65.87908 # 11.97644¶
> het.df<-1+1-1 # 1¶
```

How do you now get the *p*-value for these results?

THINK
BREAK

```
> pchisq(het.chisq, het.df, lower.tail=FALSE)¶
[1] 0.0005387742
```

The data from the two studies exhibit the same overall trend (given objects increase the likelihood of CONSTRUCTION: *V_DO_PART*) but they still differ highly significantly from each other ($\chi^2_{\text{heterogeneity}} = 11.98$; $df = 1$; $p_{\text{two-tailed}} < 0.001$). How can that be? Because of the different effect sizes: the odds ratio for Peters's data was 1.92, but in Gries's data it is nearly exactly three times as large, which is also what you would write in your results section; we will return to this example in Chapter 5.

```
> (143/66)/(53/141)¶
[1] 5.764151
```

1.2.3. One dep. variable (nominal/categorical) (dep. samples)

One central requirement of the chi-squared test for independence is that the tabulated data points are independent of each other. There are situations, however, where this is not the case, and in this section I discuss one method you can use on such occasions.

Let us assume you want to test whether metalinguistic knowledge can influence acceptability judgments. This is relevant because many acceptability judgments used in linguistic research were produced by the investigating linguists themselves, and one may well ask oneself whether it is really sensible to rely on judgments by linguists with all their metalinguistic knowledge instead of on judgments by linguistically naïve subjects. This is especially relevant since studies have shown that judgments by linguists, who after all think a lot about linguistic expressions, can deviate a lot from judgments by laymen, who usually don't (cf. Spencer 1973, Labov 1975, or Greenbaum 1976). In an admittedly oversimplistic case, you could ask 100 linguistically naïve native speakers to rate a sentence as 'acceptable' or 'unacceptable'. After the ratings have been made, you could tell the subjects which phenomenon the study investigated and which variable you

thought influenced the sentences' acceptability. Then, you would give the sentences back to the subjects to have them rate them once more. The question would be whether the subjects' newly acquired metalinguistic knowledge would make them change their ratings and, if so, how. This question involves

- a dependent nominal/categorical variable, namely BEFORE: *ACCEPTABLE* vs. BEFORE: *UNACCEPTABLE*;
- a dependent nominal/categorical variable, namely AFTER: *ACCEPTABLE* vs. AFTER: *UNACCEPTABLE*;
- dependent samples since every subject produced two judgments.

For such scenarios, you use the McNemar test (or Bowker test, cf. below). This test is related to the chi-squared tests discussed above in Sections 4.1.1.2 and 4.1.2.2 and involves the following procedure:

Procedure
- Formulating the hypotheses
- Computing the frequencies you would expect given H_0
- Testing the assumption(s) of the test:
  - the observed variable levels are related in a pairwise manner
  - the expected frequencies are ≥ 5
- Computing the test statistic χ^2, *df*, and *p*

First, the hypotheses:

H_0: The frequencies of the two possible ways in which subjects produce a judgment in the second rating task that differs from that in the first rating task are equal; $\chi^2 = 0$.

H_1: The frequencies of the two possible ways in which subjects produce a judgment in the second rating task that differs from that in the first rating task are not equal; $\chi^2 \neq 0$.

To get to know this test, we use the fictitious data summarized in Table 29, which you read in from the file <_inputfiles/04-1-2-3_accjudg.csv>. Table 29 suggests there has been a major change of judgments: Of the 100 rated sentences, only 31+17 = 48 sentences – not even half! – were judged identically in both ratings. But now you want to know whether the way in which the 52 judgments changed is significantly different from chance.

```
> AccBeforeAfter<-read.delim(file.choose())¶
> str(AccBeforeAfter); attach(AccBeforeAfter)¶
```

Table 29. Observed frequencies in a fictitious study on acceptability judgments

		AFTER		
		ACCEPTABLE	*INACCEPTABLE*	Row totals
BEFORE	*ACCEPTABLE*	31	39	70
	INACCEPTABLE	13	17	30
	Column totals	44	56	100

The McNemar test only involves those cases where the subjects changed their opinion, .i.e. cells *b* and *c* of the input table. If these are distributed equally, then the expected distribution of the 52 cases in which subjects change their opinion is that in Table 30.

Table 30. Expected frequencies in a fictitious study on acceptability judgments

		AFTER		
		ACCEPTABLE	*INACCEPTABLE*	Row totals
BEFORE	*ACCEPTABLE*		26	
	INACCEPTABLE	26		
	Column totals			

From this, you can see that both expected frequencies are larger than 5 so you can indeed do the McNemar test. As before, you compute a chi-squared value (using the by now familiar formula in (39)) and a *df-val*ue according to the formula in (40) (where *k* is the number of rows/columns):

$$(39) \qquad \chi^2 = \sum_{i=1}^{n} \frac{\left(observed - expected\right)^2}{expected} = 13$$

$$(40) \qquad df = \frac{k \cdot \left(k - 1\right)}{2} = 1$$

As before, you can look up this chi-squared value in the familiar kind of chi-square table and, again as before, if the computed chi-squared value is larger than the tabulated one for the relevant *df-val*ue for $p = 0.05$, you may reject H_0. As you can see, the chi-squared value is too large for H_0 and we accept H_1.

Table 31. Critical χ^2-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $1 \leq df \leq 3$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 1$	3.841	6.635	10.828
$df = 2$	5.991	9.21	13.816
$df = 3$	7.815	11.345	16.266

This is how you summarize this finding in the results section: "According to a McNemar test, the way 52 out of 100 subjects changed their judgments after they were informed of the purpose of the experiment is significantly different from chance: in the second rating task, the number of 'acceptable' judgments is much smaller ($\chi^2 = 13$; $df = 1$; $p_{\text{two-tailed}} < 0.001$)."

In R, this is again much easier. You need the function `mcnemar.test` and it typically requires two arguments:

- `x`: a two-dimensional table which you want to test;
- `correct=FALSE` or `correct=TRUE` (the default): when the number of changes < 30, then som recommend the continuity correction.

```
> mcnemar.test(table(BEFORE, AFTER), correct=FALSE)¶
 McNemar's Chi-squared test
data:  table(BEFORE, AFTER)
McNemar's chi-squared = 13, df = 1, p-value = 0.0003115
```

The summary and conclusions are of course the same. When you do this test for $k \times k$ tables (with $k > 2$), this test is sometimes called Bowker test.

Recommendation(s) for further study

- Sheskin (2011: Test 20) on the McNemar test, its exact alternative, which you can compute with `dbinom`
- Sheskin (2011: Test 26) for Cochran's extension of the McNemar test to test three or more measurements of a dichotomous variable, which takes only a few lines of code to compute in R – why don't you try to write such a function?
- the function `runs.test` (from the library `tseries`) to test the randomness of a binary sequence

2. Dispersions

Sometimes, it is necessary and/or interesting to not just look at the general

characteristics of a distribution but also at more narrowly defined distributional characteristics. The two most obvious characteristics are the dispersion and the central tendency of a distribution. This section is concerned with the dispersion – more specifically, the variance or standard deviation – of a variable; Section 4.3 discusses measures of central tendency.

For some research questions, it is useful to know, for example, whether two distributions have the same or a similar dispersion. Put differently, do two distributions spread around their means in a similar or in a different way? We touched upon this topic a little earlier in Section 3.1.3.6, but to illustrate the point once more, consider Figure 46.

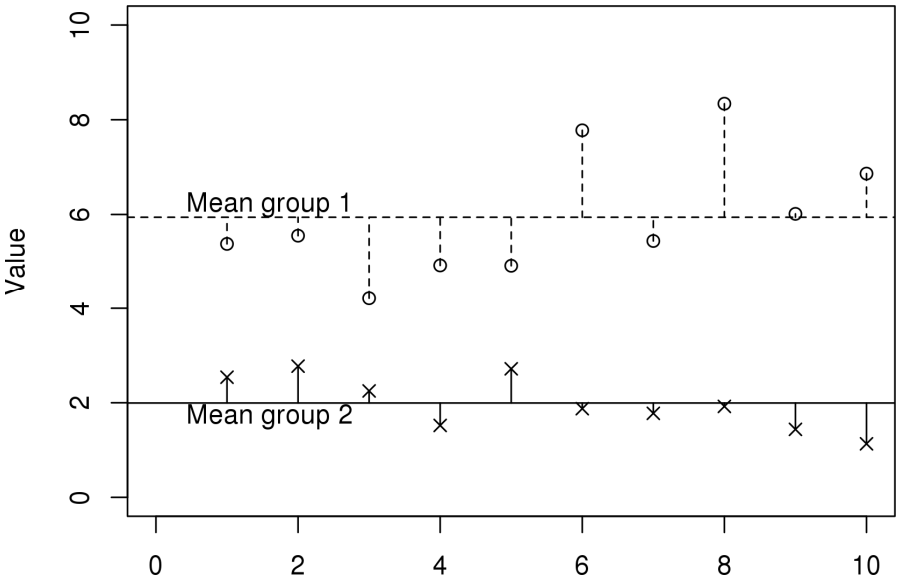

Figure 46. Two fictitious distributions

Figure 46 shows two distributions, one group of 10 values (represented by unfilled circles) and another group of 10 values (represented by crosses). The means of these groups are shown with the two horizontal lines (dashed for the first group), and the deviations of each point from its group mean are shown with the vertical lines. As you can easily see, the groups do not just differ in terms of their means ($mean_{\text{group 2}} = 1.99$; $mean_{\text{group 1}} = 5.94$), but also in terms of their dispersion: the deviations of the points of group 1 from their mean are much larger than their counterparts in group 2. While this difference is obvious in Figure 46, it can be much harder to discern in other cases, which is why we need a statistical test. In Section 4.2.1,

we discuss how you test whether the dispersion of one dependent interval/ratio-scaled variable is significantly different from a known dispersion value. In Section 4.2.2, we discuss how you test whether the dispersion of one dependent ratio-scaled variable differs significantly in two groups.

2.1. Goodness-of-fit test for one dep. variable (ratio-scaled)

As an example for this test, we return to the above data on first language acquisition of Russian tense-aspect patterning. In Section 4.1.1.1 above, we looked at how the correlation between the use of tense and aspect of one child developed over time. Let us assume, you now want to test whether the overall variability of the values for this child is significantly different from that of another child for whom you already have data. Let us also assume that for this other child you found a variance of 0.025.

This question involves the following variables and is investigated with a chi-squared test as described below:

- a dependent ratio-scaled variable, namely the variable TENSEASPECT, consisting of the Cramer's *V* values;
- no independent variable because you are not testing whether the distribution of the variable TENSEASPECT is influenced by, or correlated with, something else.

Procedure
- Formulating the hypotheses
- Computing descriptive statistics
- Testing the assumption(s) of the test: the population from which the sample whose variance is tested has been drawn or at least the sample itself from which the variance is computed is normally distributed
- Computing the test statistic χ^2, *df*, and *p*

As usual, you begin with the hypotheses:

H_0: The variance of the data for the newly investigated child does not differ from the variance of the child investigated earlier; sd^2 TENSEASPECT of the new child = sd^2 TENSEASPECT of the already investigated child, or sd^2 of the new child = 0.025, or the ratio of the two variances is 1.

H_1: The variance of the data for the newly investigated child differs

from the variance of the child investigated earlier; sd^2 TENSEASPECT of the new child ≠ sd^2 TENSEASPECT of the already investigated child, or sd^2 of the new child ≠ 0.025, or the ratio of the two variances is not 1.

You load the data from <_inputfiles/04-2-1_tense-aspect.csv>.

```
> RussianTensAsp<-read.delim(file.choose())¶
> str(RussianTensAsp); attach(RussianTensAsp)¶
```

As a next step, you must test whether the assumption of this chi-squared test is met and whether the data are in fact normally distributed. We have discussed this in detail above so we run the test here without further ado.

```
> shapiro.test(TENSE_ASPECT)¶
 Shapiro-Wilk normality test
data:   TENSE_ASPECT
W = 0.9942, p-value = 0.9132
```

Just like in Section 4.1.1.1 above, you get a *p*-value of 0.9132, which means you must not reject H_0, you can consider the data to be normally distributed, and you can compute this chi-squared test. You first compute the sample variance that you want to compare to the previous results:

```
> var(TENSE_ASPECT)¶
[1] 0.01687119
```

To test whether this value is significantly different from the known variance of 0.025, you compute a chi-squared statistic as in formula (41).

$$\chi^2 = \frac{(n-1)\cdot sample\ \ variance}{population\ \ variance} \tag{41}$$

This chi-squared value has *n*-1 = 116 degrees of freedom. In R:

```
> chi.squared<-((length(TENSE_ASPECT)-1)*var(TENSE_ASPECT))/
   0.025¶
> chi.squared¶
[1] 78.28232
```

As usual, you can create those critical values yourself or you look up this chi-squared value in the familiar kind of table.

```
> qchisq(c(0.05, 0.01, 0.001), 116, lower.tail=FALSE)¶
```

Table 32. Critical χ^2-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $115 \leq df \leq 117$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 115$	141.03	153.191	167.61
$df = 116$	142.138	154.344	168.813
$df = 117$	143.246	155.496	170.016

Since the obtained value of 78.28 is much smaller than the relevant critical value of 142.138, the difference between the two variances is not significant. You can compute the exact *p*-value as follows:

```
> pchisq(chi.squared, (length(TENSE_ASPECT)-1), lower.tail=
   FALSE)¶
[1] 0.9971612¶
```

This is how you would summarize the result: "According to a chi-squared test, the variance of the newly investigated child (0.017) does not differ significantly from the variance of the child investigated earlier (0.025): $\chi^2 = 78.28$; $df = 116$; $p_{\text{two-tailed}} > 0.05$."

2.2. One dep. variable (ratio-scaled) and one indep. variable (nominal)

The probably more frequent scenario in the domain 'testing dispersions' is the case where you test whether two samples or two variables exhibit the same dispersion (or at least two dispersions that do not differ significantly). Since the difference of dispersions or variances is probably not a concept you spent much time thinking about so far, let us look at one illustrative example from the domain of sociophonetics. Gaudio (1994) studied the pitch range of heterosexual and homosexual men. At issue was therefore not the average pitch, but its variability, a good example for how variability as such can be interesting. In that study, four heterosexual and four homosexual men were asked to read aloud two text passages and the resulting recordings were played to 14 subjects who were asked to guess which speakers were heterosexual and which were homosexual. Interestingly, the subjects were able to distinguish the sexual orientation nearly perfectly. The only (insignificant) correlation which suggested itself as a possible explanation was that the homosexual men exhibited a wider pitch range in

one of the text types, i.e., a result that has to do with variability/dispersion.

We will now look at an example from second language acquisition. Let us assume you want to study how native speakers of a language and very advanced learners of that language differed in a synonym-finding task in which both native speakers and learners are presented with words for which they are asked to name synonyms. You may now not be interested in the exact numbers of synonyms – maybe, the learners are so advanced that these are actually fairly similar in both groups – but in whether the learners exhibit more diversity in the amounts of time they needed to come up with all the synonyms they can name. This question involves

- a dependent ratio-scaled variable, namely SYNTIMES, the time subjects needed to name the synonyms;
- a nominal/categorical independent variable, namely SPEAKER: *LEARNER* and SPEAKER: *NATIVE*.

This kind of question is investigated with the so-called *F*-test for homogeneity of variances, which involves the following steps:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test:
  - the population from which the sample whose variance is tested has been drawn or at least the sample itself from which the variance is computed is normally distributed
  - the samples are independent of each other
- Computing the test statistic F, df_1 and df_2, and p

First, you formulate the hypotheses. Note that H_1 is non-directional / two-tailed.

H_0: The times the learners need to name the synonyms they can think of are not differently variable from the times the native speakers need to name the synonyms they can think of; the ratio of the variances $F = 1$.

H_1: The times the learners need to name the synonyms they can think of are differently variable from the times the native speakers need to name the synonyms they can think of; the ratio of the variances $F \neq 1$.

As an example, we use the (fictitious) data in <_inputfiles/04-2-2_synonymtimes.csv>:

```
> SynonymTimes<-read.delim(file.choose())¶
> str(SynonymTimes); attach(SynonymTimes)¶
```

You compute the variances for both subject groups and plot the data into Figure 47. The variability of the two groups seem very similar: the boxes have quite similar sizes, but the ranges of the whiskers differ a bit; cf. the code file for some additional exploration with more precise ecdf plots.

```
> tapply(SYNTIMES, SPEAKER, var)¶
  Learner   Native
10.31731 14.15385
> boxplot(SYNTIMES~SPEAKER, notch=TRUE)¶
> rug(jitter(SYNTIMES), side=2)¶
```

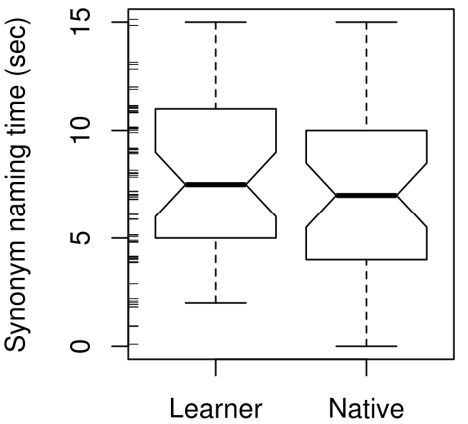

Figure 47. Boxplot for SYNTIMES~SPEAKER

The *F*-test requires a normal distribution of the population or at least the sample. We again use the Shapiro-Wilk test, this time with `tapply`. Nothing to worry about: both samples do not deviate significantly from normality and you can do an *F*-test. This test requires you to compute the quotient of the two variances (traditionally, but not necessarily – see below – the larger variance is used as the numerator). Now we compute the ratio of the two variances, which turns out to be not 1, but somewhat close to it.

```
>tapply(SYNTIMES, SPEAKER, shapiro.test)¶
$Learner
```

```
        Shapiro-Wilk normality test
data:  X[[1L]]
W = 0.9666, p-value = 0.2791
$Native
        Shapiro-Wilk normality test
data:  X[[2L]]
W = 0.9751, p-value = 0.5119

> F.value<-var(SYNTIMES[SPEAKER=="Native"])/
   var(SYNTIMES[SPEAKER=="Learner"]); F.value¶
[1] 1.371855
```

To see whether this value is significantly different from 1, you again need to consider degrees of freedom, this time even two: one for the numerator, one for the denominator. Both can be computed very easily by just subtracting 1 from the sample sizes (of the samples for the variances); cf. the formula in (42).

(42) $df_{\text{numerator}} = n_{\text{numerator sample}}\text{-}1$; $df_{\text{denominator}} = n_{\text{denominator sample}}\text{-}1$

You get 39 in both cases and can look up the result in an *F*-table.

Table 33. Critical *F*-values for $p_{\text{two-tailed}} = 0.05$ and $38 \leq df_{1,\,2} \leq 40$

	$df_2 = 38$	$df_2 = 39$	$df_2 = 40$
$df_1 = 38$	1.907	1.8963	1.8862
$df_1 = 39$	1.9014	1.8907	1.8806
$df_1 = 40$	1.8961	1.8854	1.8752

Obviously, the result is not significant: the computed *F*-value is smaller than the tabulated one for $p = 0.05$ (which is 1.8907). As usual, you can compute the critical *F*-values yourself, and you would have to use the function `qf` for that. We need four arguments:

- `p`: the *p*-value for which you want to determine the critical *F*-value (for some *df*-values);
- `df1` and `df2`: the two *df*-values for the *p*-value for which you want to determine the critical *F*-value;
- the argument `lower.tail=FALSE`, to instruct R to only consider the area under the curve above / to the right of the relevant *F*-value.

There is one last thing, though. When we discussed one- and two-tailed tests in Section 1.3.4 above, I mentioned that in the graphical representa-

tion of one-tailed tests (cf. Figure 6 and Figure 8) you add the probabilities of the events you see when you move away from the expectation of H_0 in *one* direction while in the graphical representation of two-tailed tests (cf. Figure 7 and Figure 9) you add the probabilities of the events you see when you move away from the expectation of H_0 in *both* directions. The consequence of that was that the prior knowledge that allowed you to formulate a directional H_1 was rewarded such that you needed a less extreme finding to get a significant result. This also means, however, that when you want to compute a two-tailed *p*-value using `lower.tail=FALSE`, then you need the *p*-value for $^{0.05}/_{2} = 0.025$. This value tells you which *F*-value cuts off 0.025 on only one side of the graph (say, the right one), but since a two-tailed test requires that you cut off the same area on the other/left side as well, this means that this is also the desired critical *F*-value for $p_{\text{two-tailed}} = 0.05$. Figure 48 illustrates this logic:

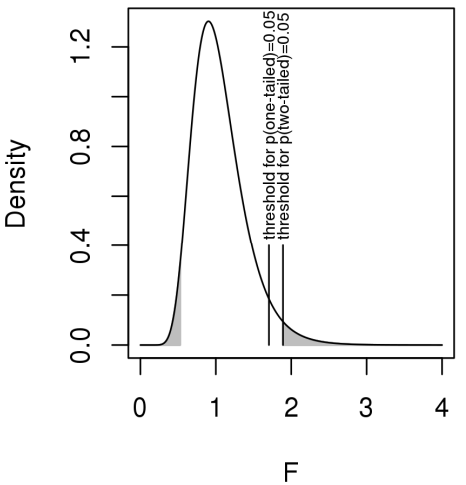

Figure 48. Density function for an *F*-distribution with $df_1 = df_2 = 39$, two-tailed test

As mentioned above, the expectation from H_0 is that $F = 1$. The right vertical line indicates the *F*-value you need to obtain for a significant two-tailed test with $df_{1,\,2} = 39$; this *F*-value is the one you already know from Table 33 – 1.8907 – which means you get a significant two-tailed result if either one of the variances is 1.8907 times larger than the other. The left vertical line indicates the *F*-value you need to obtain for a significant one-tailed test with $df_{1,\,2} = 39$; this *F*-value is 1.7045, which means you get a

significant one-tailed result if the variance you predict to be larger (!) is 1.7045 times larger than the one you predict to be smaller. To compute the *F*-values for the two-tailed tests yourself, as a beginner you may want to enter just these lines and proceed in a similar way for all other cells in Table 33, and the code file contains code to generate all of Table 33.

```
> qf(0.025, 39, 39, lower.tail=TRUE)¶
[1] 0.5288993
> qf(0.025, 39, 39, lower.tail=FALSE)¶
[1] 1.890719
```

The observed *F*-value is obviously too small for either a directional or a non-directional significant result: 1.53 < 1.89. It is more useful, however, to immediately compute the *p*-value for your *F*-value. Since you now use the reverse of `qf`, `pf`, you must now not divide but multiply by 2:

```
> 2*pf(F.value, 39, 39, lower.tail=FALSE)¶
[1] 0.3276319
```

As we've seen, with a *p*-value of $p = 0.3276$, the *F*-value of about 1.37 for $df_{1, 2} = 39$ is obviously not significant. The function for the *F*-test in R that easily takes care of all of the above is called `var.test` and it requires at least two arguments, the two samples. Just like many other functions, you can approach this in two ways: you can provide R with a formula,

```
> var.test(SYNTIMES~SPEAKER)¶
 F test to compare two variances
data:  SYNTIMES by SPEAKER
F = 0.7289, num df = 39, denom df = 39, p-value = 0.3276
alternative hypothesis: true ratio of variances is not
   equal to 1
95 percent confidence interval:
 0.385536 1.378221
sample estimates:
ratio of variances
         0.7289402
```

or you can use a vector-based alternative:

```
> var.test(SYNTIMES[SPEAKER=="Learner"],
   SYNTIMES[SPEAKER=="Native"])¶
```

Don't be confused if the *F*-value you get from R is not the same as the one you computed yourself. Barring mistakes, the value outputted by R is

then $^1/_F$-value – R does not automatically put the larger variance into the numerator, but the variance whose name comes first in the alphabet, which here is "Learner" (before "Native"). The *p*-value then shows you that R's result is the same as yours. You can now sum this up as follows: "The native's synonym-finding times exhibit a variance that is approximately 40% larger than that of the learners (14.15 vs. 10.32), but according to an *F*-test, this difference is not significant: $F = 0.73$; $df_{\text{learner}} = 39$; $df_{\text{native}} = 39$; $p_{\text{two-tailed}} = 0.3276$."

Recommendation(s) for further study

- Dalgaard (2002: 89), Crawley (2007: 289ff.), Baayen (2008: Section 4.2.3), Sheskin (2011: Tests 3, 11a)
- the function `fligner.test` to test the homogeneity of variance when the data violate the assumption of normality
- Good and Hardin (2012: 100ff.) for other (advanced!) possibilities to compare variances
- see the code file for a function `exact.f.test.indep` that I wrote to compute an exact version of this *F*-test, which you can use when your sample sizes are very small (maybe <15); careful, this test may take quite some time

3. Means

The probably most frequent use of simple significance tests apart from chi-squared tests are tests of differences between means. In Section 4.3.1, we will be concerned with goodness-of-fit tests, i.e., scenarios where you test whether an observed measure of central tendency is significantly different from another already known mean (recall this kind of question from Section 3.1.5.1); in Section 4.3.2, we then turn to tests where measures of central tendencies from two samples are compared to each other.

3.1. Goodness-of-fit tests

3.1.1. One dep. variable (ratio-scaled)

Let us assume you are again interested in the use of hedges. Early studies suggested that men and women exhibit different communicative styles with regard to the frequency of hedges (and otherwise). Let us also assume you

knew from the literature that female subjects in experiments used on average 12 hedges in a two-minute conversation with a female confederate of the experimenter. You also knew that the frequencies of hedges are normally distributed. You now did an experiment in which you recorded 30 two-minute conversations of female subjects with a male confederate and counted the same kinds of hedges as were counted in the previous studies (and of course we assume that with regard to all other parameters, your experiment was an exact replication of the earlier one). You now want to test whether the average number of hedges in your experiment is significantly different from the value of 12 reported in the literature. This question involves

- a dependent ratio-scaled variable, namely HEDGES, which will be compared to the value from the literature;
- no independent variable since you do not test whether HEDGES is influenced by something else.

For such cases, you use a one-sample *t*-test, which involves these steps:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics
- Testing the assumption(s) of the test: the population from which the sample whose mean is tested has been drawn or at least the sample itself from which the mean is computed is normally distributed
- Computing the test statistic *t*, *df*, and *p*

As always, you begin with the hypotheses:

H_0: The average of HEDGES in the conversations of the subjects with the male confederate does not differ significantly from the already known average; hedges in your experiment = 12, or hedges in your experiment-12 = 0, or $t = 0$;

H_1: The average ofHEDGES in the conversations of the subjects with the male confederate differs from the previously reported average; hedges in your experiment ≠ 12, or hedges in your experiment-12 ≠ 0, $t \neq 0$.

Then you load the data from <_inputfiles/04-3-1-1_hedges.csv>:

```
> Hedges<-read.delim(file.choose())¶
> str(Hedges); attach(Hedges)¶
```

Next, you compute the mean frequency of hedges you found in your experiment as well as a measure of dispersion (cf. the code file for a graph):

```
> mean(HEDGES); sd(HEDGES)¶
[1] 14.83333
[1] 2.506314
```

While the literature mentioned that the numbers of hedges are normally distributed, you test whether this holds for your data, too:

```
> shapiro.test(HEDGES)¶
 Shapiro-Wilk normality test
data:  HEDGES
W = 0.946, p-value = 0.1319
```

It does. You can therefore immediately proceed to the formula in (43).

$$t = \frac{\bar{x}_{sample} - \bar{x}_{population}}{sd_{sample} \Big/ \sqrt{n_{sample}}} \quad (43)$$

```
> (mean(HEDGES)-12) / (sd(HEDGES)/sqrt(length(HEDGES)))¶
[1] 6.191884
```

To see what this value means, we need degrees of freedom again. Again, this is easy here since $df = n\text{-}1$, i.e., $df = 29$. When you look up the t-value for $df = 29$ in the usual kind of table, the t-value you computed must again be larger than the one tabulated for your df at $p = 0.05$. To compute the critical p-value, you use `qt` with the p-value and the required df-value. Since you do a two-tailed test, you must cut off $^{0.05}/_{2} = 2.5\%$ on both sides of the distribution, which is illustrated in Figure 49.

Table 34. Critical t-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $28 \leq df \leq 30$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 28$	2.0484	2.7633	3.6739
$df = 29$	2.0452	2.7564	3.6594
$df = 30$	2.0423	2.75	3.646

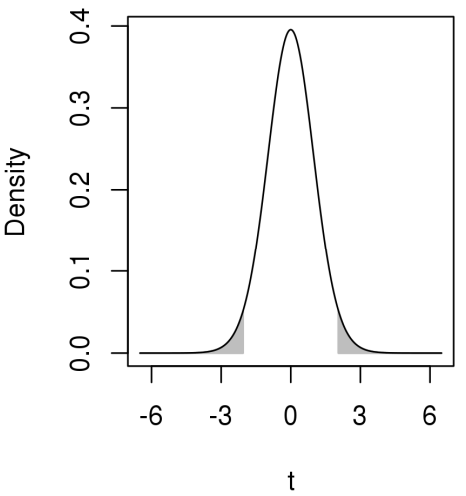

Figure 49. Density function for a *t*-distribution for *df* = 29, two-tailed test

The critical *t*-value for $p = 0.025$ and $df = 29$ is therefore:

```
> qt(c(0.025, 0.975), 29, lower.tail=FALSE)¶
[1]  2.045230 -2.045230
```

The exact *p*-value can be computed with `pt` and the obtained *t*-value is highly significant: 6.1919 is not just larger than 2.0452, but even larger than the *t*-value for $p = 0.001$ and $df = 29$. You could also have guessed that because the *t*-value of 6.19 is far in the right grey margin in Figure 49.

```
> 2*pt(6.191884, 29, lower.tail=FALSE)¶
[1] 9.42153e-07
```

To sum up: "On average, female subjects that spoke to a male confederate of the experimenter for two minutes used 14.83 hedges (standard deviation: 2.51). According to a one-sample *t*-test, this average is highly significantly larger than the value previously noted in the literature (for female subjects speaking to a female confederate of the experimenter): $t = 6.1919$; $df = 29$; $p_{\text{two-tailed}} < 0.001$."

With the right function in R, you need just one line. The relevant function is called `t.test` and requires the following arguments:

- `x`: a vector with the sample data;
- `mu=`..., the population mean to which the sample mean of `x` is compared;

– `alternative="two-sided"` for two-tailed tests (the default) or one of `alternative="greater"` or `alternative="less"`, depending on which H_1 you want to test: the value you assign to `alternative` states the relation of the sample mean to the population mean.

```
> t.test(HEDGES, mu=12)¶
 One Sample t-test
data:  HEDGES
t = 6.1919, df = 29, p-value = 9.422e-07
alternative hypothesis: true mean is not equal to 12
95 percent confidence interval:
 13.89746 15.76921
sample estimates:
mean of x
 14.83333
```

You get the already known mean of 14.83 as well as the *df*- and *t*-value we computed semi-manually. In addition, we get the exact *p*-value and the confidence interval of the mean which does not include the value of 12.

Recommendation(s) for further study
Baayen (2008: Section 4.1.2), Sheskin (2011: Test 2)

3.1.2. One dep. variable (ordinal)

In the previous section, we discussed a test that allows you to test whether the mean of a sample from a normally-distributed population is different from an already known population mean. This section deals with a test you can use when the data violate the assumption of normality or when they are not interval-/ratio-scaled to begin with. We will explore this test by looking at an interesting little morphological phenomenon, namely subtractive word-formation processes in which parts of usually two source words are merged into a new word. Two such processes are blends and complex clippings; some well-known examples of the former are shown in (44a), while (44b) provides a few examples of the latter; in all examples, the letters of the source words that enter into the new word are underlined.

(44) a. *brunch* (*br*eakfast × l*unch*), *motel* (*mot*or × h*otel*), *smog* (*smo*ke × *fog*), *foolosopher* (*fool* × *philosopher*)
b. *scifi* (*sci*ence × *fi*ction), *fedex* (*fede*ral × *ex*press), *sysadmin* (*system* × *admin*istrator)

One question that may arise upon looking at these coinages is to what degree the formation of such words is supported by some degree of similarity of the source words. There are many different ways to measure the similarity of words, and the one we are going to use here is the so-called Dice coefficient (cf. Brew and McKelvie 1996). You can compute a Dice coefficient for two words in two simple steps. First, you split the words up into letter (or phoneme or …) bigrams. For *motel* (*motor* × *hotel*) you get:

- *motor*: *mo*, *ot*, *to*, *or*;
- *hotel*: *ho*, *ot*, *te*, *el*.

Then you count how many of the bigrams of each word occur in the other word, too. In this case, these are two: the *ot* of *motor* also occurs in *hotel*, and thus the *ot* of *hotel* also occurs in *motor*.[24] This number, 2, is divided by the number of bigrams to yield the Dice coefficient:

(45) $Dice_{motor\ \&\ hotel} = {}^{2}/_{8} = 0.25$

In other words, the Dice coefficient is the percentage of shared bigrams out of all bigrams (and hence ratio-scaled). We will now investigate the question of whether source words that entered into subtractive word-formation processes are more similar to each other than words in general are similar to each other. Let us assume, you know that the average Dice coefficient of randomly chosen words is 0.225 (with a standard deviation of 0.0809; the median is 0.151 with an interquartile range of 0.125). These figures already suggest that the data may not be normally distributed.[25]

This study involves

- a dependent ratio-scaled variable, namely the SIMILARITY of the source words, which will be compared with the already known mean/median;
- no independent variable since you do not test whether SIMILARITY is influenced by something else.

The hypotheses should be straightforward:

24. In R, such computations can be easily automated and done for hundreds of thousands of words. For example, if the vector `a` contains a word, this line returns all its bigrams: `substr(rep(a, nchar(a)-1), 1:(nchar(a)-1), 2:(nchar(a)))¶`; for many such applications, cf. Gries (2009a).
25. For authentic data, cf. Gries (2006), where I computed Dice coefficients for all 499,500 possible pairs of 1,000 randomly chosen words.

H_0: The average of SIMILARITY for the source words that entered into subtractive word-formation processes is not significantly different from the known average of randomly chosen word pairs; Dice coefficients of source words = 0.225, or Dice coefficients of source words-0.225 = 0.

H_1: The average of SIMILARITY for the source words that entered into subtractive word-formation processes is different from the known average of randomly chosen word pairs; Dice coefficients of source words ≠ 0.225, or Dice coefficients of source words-0.225 ≠ 0.

The data to be investigated here are in <_inputfiles/04-3-1-2_dices.csv>; they are data of the kind studied in Gries (2006).

```
> Dices<-read.delim(file.choose())¶
> str(Dices); attach(Dices)¶
```

From the summary statistics, you could already infer that the similarities of randomly chosen words are not normally distributed. We can therefore assume that this is also true of the sample of source words, but of course you also test this assumption (cf. the code file for a plot):

```
> shapiro.test(DICE)¶
 Shapiro-Wilk normality test
data:  DICE
W = 0.9615, p-value = 0.005117
```

The Dice coefficients are not normally, but symmetrically distributed (as you can also clearly see in the ecdf plot). Thus, even though Dice coefficients are ratio-scaled and although the sample size is >30, you may want to be careful and not use the one-sample *t*-test but, for example, the so-called one-sample sign test for the median, which involves these steps:

Procedure
Formulating the hypotheses
Computing the frequencies of the signs of the differences between the observed values and the expected average
Computing the probability of error *p*

You first rephrase the hypotheses; I only provide new statistical ones:

H_0: $median_{\text{Dice coefficients of your source words}} = 0.151$.

H_1: $median_{\text{Dice coefficients of your source words}} \neq 0.151$.

Then, you compute descriptive statistics: the median and its interquartile range. Obviously, the observed median Dice coefficient is a bit higher than 0.151, the median Dice coefficient of the randomly chosen word pairs, but it is impossible to guess whether the difference is going to be significant.

```
> median(DICE); IQR(DICE)¶
[1] 0.1775
[1] 0.10875
```

For the one-sample sign test, you first determine how many observations are above and below the expected median, because if the expected median was a good characterization of the observed data, then 50% of the observed data should be above the expected median and 50% should be below it. (NB: you must realize that this means that the exact sizes of the deviations from the expected median are not considered here – you only look at whether the observed values are larger or smaller than the expected median, but not how much larger or smaller.)

```
> sum(DICE>0.151); sum(DICE<0.151)¶
[1] 63
[1] 37
```

63 of the 100 observed values are larger than the expected median (the rest is smaller than the expected median) – since you expected 50, it seems as if the Dice coefficients observed in your source words are significantly larger than those of randomly chosen words. As before, this issue can also be approached graphically, using the logic and the function `dbinom` from Section 1.3.4.1, Figure 7. Figure 50 shows the probabilities of all possible results you can get in 100 trials – because you look at the Dice coefficients of 100 subtractive formations. First, consider the left panel of Figure 50.

According to H_0, you would expect 50 Dice coefficients to be larger than the expected median, but you found 63. Thus, you add the probability of the observed result (the black bar for 63 out of 100) to the probabilities of all those that deviate from H_0 even more extremely, i.e., the chances to find 64, 65, …, 99, 100 Dice coefficients out of 100 that are larger than the expected median. These probabilities from the left panel sum up to approximately 0.006.

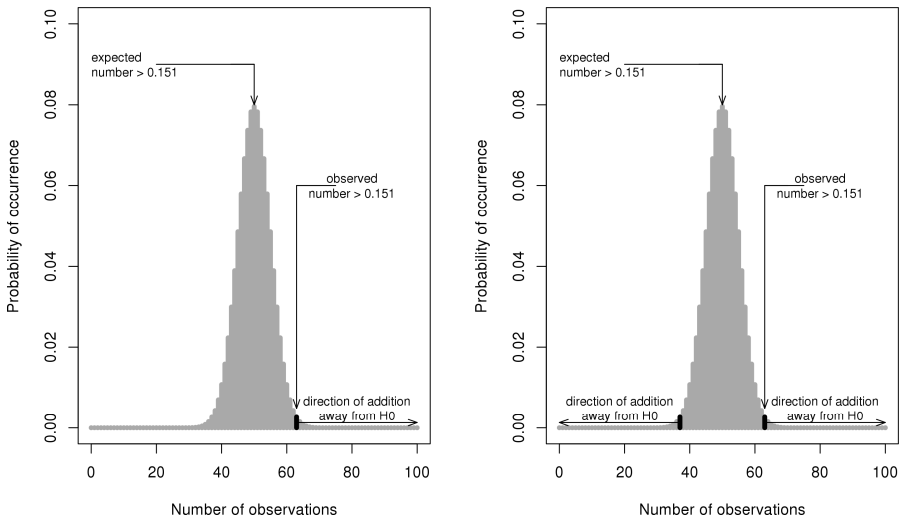

Figure 50. Probability distributions for 100 binomial trials test

```
> sum(dbinom(63:100, 100, 0.5))¶
[1] 0.006016488
```

But you are not finished yet … As you can see in the left panel of Figure 50, so far you only include the deviations from H_0 in one direction – but your H_1 is non-directional, i.e., two-tailed. You must therefore also include the probabilities of the events that deviate just as much and more from H_0 in the other direction: 37, 36, …, 1, 0 Dice coefficients out of 100 that are smaller than the expected median, as represented in the right panel of Figure 50. The probabilities sum up to the same value (because the distribution of binomial probabilities around $p = 0.5$ is symmetric).

```
> sum(dbinom(37:0, 100, 0.5))¶
[1] 0.006016488
```

Again: if you expect 50 out of 100, but observe 63 out of 100, and want to do a two-tailed test, you must add the summed probability of finding 63 to 100 larger Dice coefficients (the upper/right 38 probabilities) to the summed probability of finding 0 to 37 smaller Dice coefficients (the lower/left 38 probabilities). The $p_{\text{two-tailed}}$-value of 0.01203298 you then get is significant. You can sum up: "The investigation of 100 subtractive word formations resulted in an average source-word similarity of 0.1775 (median, *IQR* = 0.10875). 63 of the 100 source words were more similar to each

other than expected from random word pairs, which, according to a two-tailed sign test is a significant deviation from the average similarity of random word pairs (median =0.151, *IQR* range = 0.125): $p_{\text{binomial}} = 0.012$."

Recall that this one-sample sign test only uses nominal information, whether each data point is larger or smaller than the expected reference median. If the distribution of the data is rather symmetrical – as it is here – then there is an alternative test that also takes the sizes of the deviations into account, i.e. uses at least ordinal information. This so-called one-sample signed-rank test can be computed using the function `wilcox.test`. Apart from the vector to be tested, the following arguments are relevant:

- `alternative`: a character string saying which H_1 you want to test: the default is `"two.sided"`, other possible values for one-tailed tests are `"less"` or `"greater"`, which specify how the first-named vector relates to the specified reference median;
- `mu=...`: the reference median expected according to H_0;
- `exact=TRUE`, if you want to compute an exact test (only when your sample size is smaller than 50 and there are no ties) or `exact=FALSE`, if an asymptotic test is sufficient; the default amounts to the latter;
- `correct=TRUE` (the default) for a continuity correction or `correct=FALSE` for none;
- `conf.level`: a value between 0 and 1 specifying the size of the confidence interval; the default is 0.95.

Since you have a non-directional H_1, you do a two-tailed test by simply adopting the default setting for `alternative`:

```
> wilcox.test(DICE, mu=0.151, correct=FALSE)¶
 Wilcoxon signed rank test
data:  DICE
V = 3454.5, p-value = 0.001393
alternative hypothesis: true location is not equal to 0.151
```

The test confirms the previous result: both the one-sample sign test, which is only concerned with the directions of deviations, and the one-sample signed rank test, which also considers the sizes of these deviations, indicate that the source words of the subtractive word-formations are more similar to each other than expected from random source words. This should however, encourage you to make sure you formulate exactly the hypothesis you are interested in (and then use the required test).

Recommendation(s) for further study
- Baayen (2008: Section 4.1.2), Sheskin (2011: Test 9b, 6)

3.2. Tests for differences/independence

A particularly frequent scenario requires you to test two groups of elements with regard to whether they differ in their central tendency. As discussed above, there are several factors that determine which test to choose:

- the kind of samples: dependent or independent (cf. Section 1.3.4.1 and the beginning of Chapter 4);
- the level of measurement of the dependent variable: interval/ratio-scaled vs. ordinal;
- the distribution of (interval/ratio-scaled) dependent variable: normal vs. non-normal;
- the sample sizes.

To reiterate the discussion at the beginning of this chapter: is the dependent variable ratio-scaled as well as normally-distributed or both sample sizes are larger than 30 or are the differences between variables normally distributed, then you can usually do a *t*-test (for independent or dependent samples, as required) – otherwise you should do a *U*-test (for independent samples) or a Wilcoxon test (for dependent samples) (or, maybe, computationally intense exact tests). The reason for this decision procedure is that while the *t*-test for independent samples requires, among other things, normally distributed samples, we have seen that samples of 30+ elements can be normally distributed even if the underlying distribution is not. Therefore, it is sometimes sufficient, though not conservative, if the data meet one of the two conditions. Strictly speaking, the *t*-test for independent samples also requires homogenous variances, which we will also test for, but we will discuss a version of the *t*-test that can handle heterogeneous variances, the *t*-test after Welch.

3.2.1. One dep. variable (ratio-scaled) and one indep. variable (nominal) (indep. samples)

The *t*-test for independent samples is one of the most widely used tests. To

explore it, we use an example from the domain of phonetics. Let us assume you wanted to study the (rather trivial) non-directional H_1 that the first formants' frequencies of men and women differed. You plan an experiment in which you record men's and women's pronunciation of a relevant set of words and/or syllables, which you then analyze. This study involves

- one dependent ratio-scaled variable, namely F1-FREQUENCIES, whose averages you are interested in;
- one independent nominal variable, namely SEX: *MALE* vs. SEX: *FEMALE*;
- independent samples since, if every subject provides just one data point, the data points are not related to each other.

The test to be used for such scenarios is the *t*-test for independent samples and it involves the following steps:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test:
  - the population from which the samples whose means are tested have been drawn or at least the samples itself from which the means are computed are normally distributed (esp. with samples of $n < 30$)
  - the variances of the populations from which the samples have been drawn or at least the variances of the samples are homogeneous
  - the samples are independent of each other
- Computing the test statistic *t*, *df*, and *p*

You begin with the hypotheses.

H_0: The average F1 frequency of men is the same as the average F1 frequency of women: $mean_{\text{F1 frequency of men}} = mean_{\text{F1 frequency of women}}$, or $mean_{\text{F1 frequency of men}} - mean_{\text{F1 frequency of men}} = 0$, or $t = 0$;

H_1: The average F1 frequency of men is not the same as the average F1 frequency of women: $mean_{\text{F1 frequency of men}} \neq mean_{\text{F1 frequency of women}}$, or $mean_{\text{F1 frequency of men}} - mean_{\text{F1 frequency of men}} \neq 0$, or $t \neq 0$.

The data you will investigate here are part of the data borrowed from a similar experiment on vowels in Apache. First, you load the data from <_inputfiles/04-3-2-1_f1-freq.csv> into R:

```
> Vowels<-read.delim(file.choose()¶
> str(Vowels); attach(Vowels)¶
```

Then, you compute the relevant means and the standard deviations of the frequencies. As usual, we use the more elegant variant with `tapply`.

```
> tapply(HZ_F1, SEX, mean)¶
       F        M
528.8548  484.2740
> tapply(HZ_F1, SEX, sd)¶
        F         M
110.80099  87.90112
```

To get a better impression of the data, you also immediately generate a boxplot. You set the limits of the *y*-axis such that it ranges from 0 to 1,000 so that all values are nicely represented; in addition, you use `rug` to plot the values of the women and the men onto the left and right *y*-axis respectively; cf. Figure 51 and the code file for an alternative that includes a stripchart.

```
> boxplot(HZ_F1~SEX, notch=TRUE, ylim=(c(0, 1000)),
   xlab="Sex", ylab="F1 frequency"); grid()¶
> rug(HZ_F1[SEX=="F"], side=2); rug(HZ_F1[SEX=="M"], side=4)¶
```

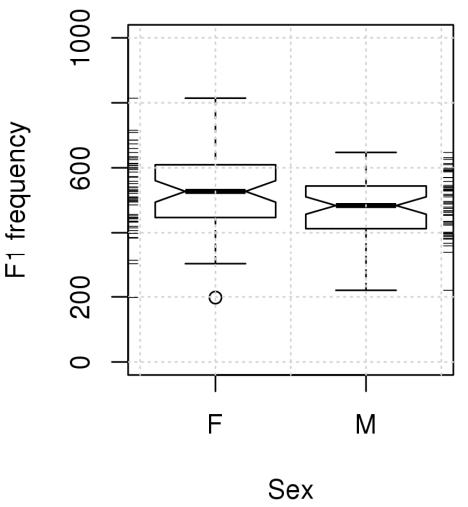

Figure 51. Boxplot for HZ_F1~SEX

The next step consists of testing the assumptions of the *t*-test. Figure 51 suggests that these data meet the assumptions. First, the boxplots for the

men and the women appear as if the data are normally distributed: the medians are in the middle of the boxes and the whiskers extend nearly equally long in both directions. Second, the variances seem to be very similar since the sizes of the boxes and notches are very similar. However, of course you need to test this and you use the familiar Shapiro-Wilk test:

```
> tapply(HZ_F1, SEX, shapiro.test)¶
$F
 Shapiro-Wilk normality test
data:  X[[1L]]
W = 0.987, p-value = 0.7723
$M
 Shapiro-Wilk normality test
data:  X[[2L]]
W = 0.9724, p-value = 0.1907
```

The data do not differ significantly from normality. Now you test for variance homogeneity with the *F*-test from Section 4.2.2 (whose assumption of normality we now already tested). This test's hypotheses are:

H_0: The variance of the first sample equals that of the second; $F = 1$.
H_1: The variance of one sample is larger than that of the second; $F \neq 1$.

The *F*-test with R yields the following result:

```
> var.test(HZ_F1~SEX) # with a formula¶
 F test to compare two variances
data:  HZ_F1 by SEX
F = 1.5889, num df = 59, denom df = 59, p-value = 0.07789
alternative hypothesis: true ratio of variances is not
   equal to 1
95 percent confidence interval:
 0.949093 2.660040
sample estimates:
ratio of variances
         1.588907
```

The second assumption is also met: since the confidence interval includes 1 and $p > 0.05$ so the variances are not significantly different from each other and you can compute the *t*-test for independent samples. This test involves three different statistics: the test statistic *t*, the number of degrees of freedom *df*, and of course the *p*-value. In the case of the *t*-test we discuss here, the *t*-test after Welch, the *t*-value is computed according to the formula in (46), where sd^2 is the variance, *n* is the sample size, and the subscripts 1 and 2 refer to the two samples of men and women.

$$(46) \qquad t = \left| \left(\bar{x}_1 - \bar{x}_2 \right) \div \sqrt{sd_1^2 / n_1 + sd_2^2 / n_2} \right|$$

```
> t.numerator<-mean(HZ_F1[SEX=="M"])-mean(HZ_F1[SEX=="F"])¶
> t.denominator<-sqrt((var(HZ_F1[SEX=="M"])/
   length((HZ_F1[SEX=="M"])))+(var(HZ_F1[SEX=="F"])/
   length((HZ_F1[SEX=="F"]))))¶
> t.value<-abs(t.numerator/t.denominator)¶
```

You get $t = 2.441581$. The formula for the degrees of freedom is somewhat more complex. First, you need to compute a value called c, and with c, you can then compute df. The formula to compute c is shown in (47), and the result of (47) gets inserted into (48).

$$(47) \qquad c = \frac{sd_1^2 / n_1}{sd_1^2 / n_1 + sd_2^2 / n_2}$$

$$(48) \qquad df = \left(\frac{c^2}{n_1 - 1} + \frac{(1-c)^2}{n_2 - 1} \right)^{-1}$$

```
> c.numerator<-var(HZ_F1[SEX=="M"])/length(HZ_F1[SEX=="M"])¶
> c.denominator<-t.denominator^2¶
> c.value<-c.numerator/c.denominator¶
> df.summand1<-c.value^2/(length(HZ_F1[SEX=="M"])-1)¶
> df.summand2<-((1-c.value)^2)/(length(HZ_F1[SEX=="F"])-1)¶
> df<-(df.summand1+df.summand2)^-1¶
```

You get $c = 0.3862634$ and $df \approx 112.195$. You then look up the t-value in the usual kind of t-table (cf. Table 34) or you compute the critical t-value (with `qt(c(0.025, 0.975), 112, lower.tail=FALSE)¶`; as before, for a two-tailed test you compute the t-value for $p = 0.025$).

Table 34. Critical t-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $111 \leq df \leq 113$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 111$	1.9816	2.6208	3.3803
$df = 112$	1.9814	2.6204	3.3795
$df = 113$	1.9812	2.62	3.3787

As you can see, the observed *t*-value is larger than the one tabulated for $p = 0.05$, but smaller than the one tabulated for $p = 0.01$: the difference between the means is significant. The exact *p*-value can be computed with `pt` and for the present two-tailed case you simply enter this:

```
> 2*pt(t.value, 112.195, lower.tail=FALSE)¶
[1] 0.01618534
```

In R, you can use the function `t.test`, which takes several arguments, the first two of which – the relevant samples – can be given by means of a formula or with two vectors. These are the other relevant arguments:

- `alternative`: a character string that specifies which H_1 is tested: the default value, which can therefore be omitted, is `"two.sided"`, other values for one-tailed hypotheses are again `"less"` or `"greater"`; as before, R considers the alphabetically first variable level (i.e., here "F") as the reference category so that the one-tailed hypothesis that the values of the men are smaller than those of the women would be tested with `alternative="greater"`;
- `paired=FALSE` for the *t*-test for independent samples (the default) or `paired=TRUE` for the *t*-test for dependent samples (cf. the next section);
- `var.equal=TRUE`, when the variances of the two samples are equal, or `var.equal=FALSE` if they are not; the latter is the useful default, which should hardly be changed;
- `conf.level`: a value between 0 and 1, which specifies the confidence interval of the difference between the means; the default is 0.95.

Thus, to do the *t*-test for independent samples, you can enter either variant listed below. You get the following result:

```
> t.test(HZ_F1~SEX, paired=FALSE)¶
 Welch Two Sample t-test
data:  HZ_F1 by SEX
t = 2.4416, df = 112.195, p-value = 0.01619
alternative hypothesis: true difference in means is
   not equal to 0
95 percent confidence interval:
 8.403651 80.758016
sample estimates:
mean in group F mean in group M
       528.8548        484.2740
> t.test(HZ_F1[SEX=="F"], HZ_F1[SEX=="M"], paired=FALSE)¶
```

The first two lines of the output provide the name of the test and the data to which the test was applied. Line 3 lists the test statistic *t* (the sign is irrelevant and depends on which mean is subtracted from which, but it must of course be considered for the manual computation), the *df*-value, and the *p*-value. Line 4 states the H_1 tested. Then, you get the confidence interval for the differences between means (and our test is significant because this confidence interval does not include 0). Finally, you get the means again.

You can sum up your results as follows: "In the experiment, the average F1 frequency of the vowels produced by men was 484.3 Hz (*sd* = 87.9), the average F1 frequency of the vowels produced by the women was 528.9 Hz (*sd* = 110.8). According to a *t*-test for independent samples, the difference of 44.6 Hz between the means is statistically significant, but not particularly strong: $t_{\text{Welch}} = 2.4416$; $df = 112.2$; $p_{\text{two-tailed}} = 0.0162$."

In Section 5.2.2, we will discuss the extension of this test to cases where you have more than one independent variable and/or where the independent variable has more than two levels.

Recommendation(s) for further study

- Crawley (2007: 289ff.), Baayen (2008: Section 4.2.2), Sheskin (2011: Test 11)
- see the code file for a function `exact.t.test.indep` that I wrote to compute an exact version of this *F*-test, which you can use when your sample sizes are very small (maybe <15); careful, this test may take quite some time (and it requires the library `combinat`)

3.2.2. One dep. variable (ratio-scaled) and one indep. variable (nominal) (dep. samples)

The previous section illustrated a test for means from two independent samples. The name of that test suggests that there is a similar test for dependent samples, which we will discuss in this section on the basis of an example from translation studies. Let us assume you want to compare the lengths of English and Portuguese texts and their respective translations into Portuguese and English. Let us also assume you suspect that the translations are on average longer than the originals. This question involves

- one dependent ratio-scaled variable, namely the LENGTH of the texts;
- one independent nominal/categorical variable, namely TEXTSOURCE: *ORIGINAL* vs. TEXTSOURCE: *TRANSLATION*;

- dependent samples since the LENGTH values for each translation are connected to those of each original text.

Performing a *t*-test for dependent samples requires the following steps:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test: the differences of the paired values of the dependent samples are normally distributed
- Computing the test statistic *t*, *df*, and *p*

As usual, you formulate the hypotheses, but note that this time the H_1 is directional: you suspect that the average length of the originals is *shorter* than those of their translations, not just different (i.e., shorter *or* longer). Therefore, the statistical form of H_1 does not just contain a "≠", but something more specific, "<":

H_0: The average of the pairwise differences between the lengths of the originals and the lengths of the translations is 0; $mean_{\text{pairwise differerences}} = 0$; $t = 0$.

H_1: The average of the pairwise differences between the lengths of the originals and the lengths of the translations is smaller than 0; $mean_{\text{pairwise differerences}} < 0$; $t < 0$.

Note in particular (i) that the hypotheses do not involve the values of the two samples but the pairwise differences between them and (ii) how these differences are computed: original minus translation, not the other way round (and hence we use "< 0"). To illustrate this test, we will look at data from Frankenberg-Garcia (2004). She compared the lengths of eight English and eight Portuguese texts, which were chosen and edited such that their lengths were approximately 1,500 words, and then she determined the lengths of their translations. You can load the data from <_inputfiles/04-3-2-2_textlengths.csv>:

```
> Texts<-read.delim(file.choose()¶
> str(Texts); attach(Texts)¶
```

Note that the data are organized so that the order of the texts and their translations is identical: case 1 is an English original (hence, TEXT is 1,

TEXTSOURCE is *ORIGINAL*, LANGUAGE is *ENGLISH*), and case 17 is its translation (hence, TEXT is again 1, but TEXTSOURCE is now *TRANSLATION*, and LANGUAGE is *PORTUGUESE*), etc. First, you compute the means and generate a plot.

```
> tapply(LENGTH, TEXTSOURCE, mean)¶
    Original  Translation
    1500.062     1579.938
> boxplot(LENGTH~TEXTSOURCE, notch=TRUE, ylim=c(0, 2000))¶
> rug(LENGTH, side=2)¶
```

The median translation length is a little higher than that of the originals and the two samples have *very* different dispersions (only because the lengths of the originals were 'set' to approximately 1,500 words and thus exhibit very little variation while the lengths of the translations were not controlled like that).

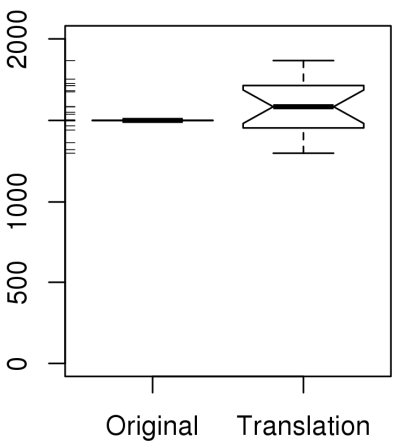

Figure 52. Boxplot for LENGTH~TEXTSOURCE

Now, this is actually a bad plot to represent the data – why?

THINK
BREAK

This plot does not portray the information that the data points from the left part – the lengths of the originals – are related to those from the right

part – the lengths of their translations! Thus, see the code file for three better plots (esp. the third). Given the controlled original lengths, the difference here is not that huge, but in other applications, a boxplot for dependent samples like the above can be very misleading.

Unlike the *t*-test for independent samples, the *t*-test for dependent samples does not presuppose a normal distribution or variance homogeneity of the sample values, but a normal distribution of the differences between the pairs of sample values. You can create a vector with these differences and then apply the Shapiro-Wilk test to it in one line with this shortcut.

```
> shapiro.test(differences<-LENGTH[1:16]-LENGTH[17:32])¶
 Shapiro-Wilk normality test
data:  differences
W = 0.9569, p-value = 0.6057
```

The differences do not differ significantly from normality so you can in fact do the *t*-test for dependent samples. First, you compute the *t*-value according to the formula in (49), where *n* is the number of value pairs.

$$t = \left|\bar{x}_{differences}\right| \cdot \sqrt{n} \Big/ sd_{differences} \tag{49}$$

```
> t.value<-(abs(mean(differences))*
   sqrt(length(differences)))/sd(differences)¶
> t,value
[1] 1.927869
```

Second, you compute the degrees of freedom *df*, which is the number of differences *n* minus 1:

```
> df<-length(differences)-1; df¶
[1] 15
```

First, you can now compute the critical values for $p = 0.05$ – this time *not* for $^{0.05}/_{2} = 0.025$ because you have a directional H_1 – at $df = 15$ or, in a more sophisticated way, create the whole *t*-table.

```
> qt(c(0.05, 0.95), 15, lower.tail=FALSE)¶
[1]  1.753050 -1.753050
```

Second, you can look up the *t*-value in such a *t*-table, repeated here as Table 35. Since such tables usually only list the positive values, you use the

absolute value of your *t*-value. As you can see, the differences between the originals and their translations is significant, but not very or highly significant: 1.927869 > 1.7531, but 1.927869 < 2.6025.

Table 35. Critical *t*-values for $p_{\text{one-tailed}} = 0.05$, 0.01, and 0.001 (for $14 \leq df \leq 16$)

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 14$	1.7613	2.6245	3.7874
$df = 15$	1.7531	2.6025	3.7328
$df = 16$	1.7459	2.5835	3.6862

Alternatively, you can compute the exact *p*-value. Since you have a directional H_1, you only need to cut off 5% of the area under the curve on one side of the distribution. The *t*-value following from H_0 is 0 and the *t*-value you computed is approximately 1.93 so you must compute the area under the curve from 1.93 to $+\infty$; cf. Figure 53. Since you are doing a one-tailed test, you need not multiply the *p*-value with 2.

```
> pt(t.value, 15, lower.tail=FALSE)¶
[1] 0.03651146
```

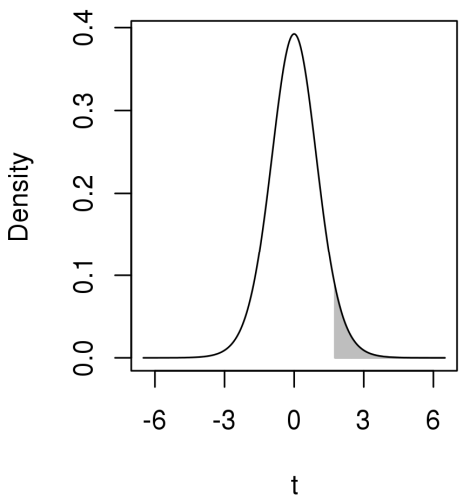

Figure 53. Density function for a *t*-distribution for $df = 15$, one-tailed test

Note that this also means that the difference is only significant because you did a one-tailed test –a two-tailed test with its multiplication with 2 would not have yielded a significant result but $p = 0.07302292$.

Now the same test with R. Since you already know the arguments of the function `t.test`, we can focus on the only major differences to before, the facts that you now have a directional H_1 and need to do a one-tailed test and that you now do a paired test. To do that properly, you must first understand how R computes the difference. As mentioned above, R proceeds alphabetically and computes the difference 'alphabetically first level minus alphabetically second level' (which is why H_1 was formulated this way above). Since "Original" comes before "Translation" and we hypothesized that the mean of the former would be smaller than that of the latter, the difference is smaller than 0. You therefore tell R that the difference is "less" than zero.

Of course you can use the formula or the vector-based notation. I show the output of the formula notation but both ways result in the same output. You get the *t*-value (ours was positive only because we used `abs`), the *df*-value, a *p*-value, and a confidence interval which, since it does not include 0, also reflects the significant result.

```
> t.test(LENGTH~TEXTSOURCE, paired=TRUE, alternative="less")¶
 Paired t-test
data:  LENGTH by TEXTSOURCE
t = -1.9279, df = 15, p-value = 0.03651
alternative hypothesis: true difference in means is less
   than 0
95 percent confidence interval:
      -Inf -7.243041
sample estimates:
mean of the differences
                -79.875
> t.test(LENGTH[TEXTSOURCE=="Original"], LENGTH[TEXTSOURCE==
   "Translation"], paired=TRUE, alternative="less")¶
```

To sum up: "On average, the originals are approximately 80 words shorter than their translations (the 95% confidence interval of this difference is -Inf, -7.24). According to a one-tailed *t*-test for dependent samples, this difference is significant: $t = -1.93$; $df = 15$; $p_{\text{one-tailed}} = 0.0365$. However, the effect is relatively small: the difference of 80 words corresponds to only about 5% of the length of the texts."

Recommendation(s) for further study

- Crawley (2007: 298ff.), Baayen (2008: Section 4.3.1), Sheskin (2011: Test 17)
- see the code file for a function `exact.t.test.dep` that I wrote to compute an exact version of this *F*-test, which you can use when your sam-

ple sizes are very small (maybe <15); careful, this test may take quite some time (for this example, it returns nearly the exact same *p*-value)

3.2.3. One dep. variable (ordinal) and one indep. variable (nominal) (indep. samples)

In this section, we discuss a non-parametric test for two independent samples of ordinal data, the *U*-test. Since I mentioned at the beginning of Section 4.3.2 that the *U*-test is not only used when the samples to be compared consist of ordinal data, but also when they violate distributional assumptions, this section will again involve an example where only a test of these distributional assumptions allows you to decide which test to use.

In Section 4.3.1.2 above, you looked at the similarities of source words entering into subtractive word formations and you tested whether these similarities were on average different from the known average similarity of random words to each other. The data you used were of the kind studied in Gries (2006) but in the above example no distinction was made between source words entering into different kinds of subtractive word formations. This is what we will do here by comparing similarities of source words entering into blends to similarities of source words entering into complex clippings. If both kinds of word-formation processes differed according to this parameter, this would provide empirical motivation for distinguishing them in the first place. This example, thus, involves

- one dependent ratio-scaled variable, namely the SIMILARITY of the source words whose averages you are interested in;
- one independent nominal variable, namely PROCESS: *BLEND* vs. PROCESS: *COMPLCLIP*;
- independent samples since the Dice coefficient of any one pair of source words has nothing to do with any one other pair of source words.

This kind of question would typically be investigated with the *t*-test for independent samples we discussed above. According to the above procedure, you first formulate the hypotheses (non-directionally, since we may have no a priori reason to assume a particular difference):

H_0: The mean of the Dice coefficients of the source words of blends is the same as the mean of the Dice coefficients of the source words of complex clippings; $mean_{\text{Dice coefficients of blends}} = mean_{\text{Dice coefficients of}}$

complex clippings, or $mean_{\text{Dice coefficients of blends}} - mean_{\text{Dice coefficients of complex clippings}} = 0$.

H_1: The mean of the Dice coefficients of the source words of blends is not the same as the mean of the Dice coefficients of the source words of complex clippings; $mean_{\text{Dice coefficients of blends}} \neq mean_{\text{Dice coefficients of complex clippings}}$, or $mean_{\text{Dice coefficients of blends}} - mean_{\text{Dice coefficients of complex clippings}} \neq 0$.

You can load the data from the file <_inputfiles/04-3-2-3_dices.csv>. As before, this file contains the Dice coefficients, but now also in an additional column the word formation process for each Dice coefficient.

```
> Dices<-read.delim(file.choose())¶
> str(Dices); attach(Dices)¶
```

As usual, you should begin by exploring the data graphically:

```
> boxplot(DICE~PROCESS, notch=TRUE, ylim=c(0, 1),
   ylab="Dice")¶
> rug(jitter(DICE[PROCESS=="Blend"]), side=2)¶
> rug(jitter(DICE[PROCESS=="ComplClip"]), side=4)¶
> text(1:2, tapply(DICE, PROCESS, mean), "x")¶
```

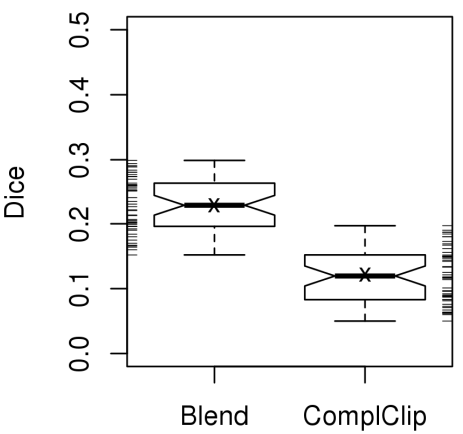

Figure 54. Boxplot for SIMILARITY~PROCESS

As usual, this graph already gives away enough information to nearly obviate the need for statistical analysis. The probably most obvious aspect is the difference between the two medians, but since the data are ratio-

scaled you also need to explore the means. These are already plotted into the graph and here is the usual line of code to compute them; note how large the difference is between the two.

```
> tapply(DICE, PROCESS, mean)¶
    Blend ComplClip
  0.22996   0.12152
> tapply(DICE, PROCESS, sd)¶
      Blend  ComplClip
  0.4274985 0.04296569
```

In order to test whether the *t*-test for independent samples can be used here, we need to test both of its assumptions, normality in the groups and variance homogeneity. Since the *F*-test for homogeneity of variances presupposes normality, you begin by testing whether the data are normally distributed. The rugs in Figure 54 suggest they are not, which is supported by the Shapiro-Wilk test.

```
> tapply(DICE, PROCESS, shapiro.test)¶
$Blend
 Shapiro-Wilk normality test
data:  X[[1L]]
W = 0.9455, p-value = 0.02231
$ComplClip
 Shapiro-Wilk normality test
data:  X[[2L]]
W = 0.943, p-value = 0.01771
```

Given these violations of normality, you can actually not do the regular *F*-test to test the second assumption of the *t*-test for independent samples. You therefore do the Fligner-Killeen test of homogeneity of variances, which does not require the data to be normally distributed and which I mentioned in Section 4.2.2 above.

```
> fligner.test(DICE~PROCESS)¶
 Fligner-Killeen test of homogeneity of variances
data:  DICE by PROCESS
Fligner-Killeen:med chi-squared=3e-04, df=1, p-value=0.9863
```

The variances are homogeneous, but normality is still violated. It follows that even though the data are ratio-scaled and even though the sample sizes are larger than 30, it may safer to compute a test that does not make these assumptions, the *U*-test.

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test:
  - the samples are independent of each other
  - the populations from which the samples whose central tendencies are tested have been drawn are identically distributed [26]
- Computing the test statistic U, z, and p

The two boxplots look relatively similar and the variances of the two groups are not significantly different, and the U-test is robust (see above) so we use it here. Since the U-test assumes only ordinal data, you now compute medians, not just means. You therefore adjust your hypotheses and compute medians and interquartile ranges:

H_0: The median of the Dice coefficients of the source words of blends is as large as the median of the Dice coefficients of the source words of complex clippings; $median_{\text{Dice coefficients of blends}} = median_{\text{Dice coefficients of complex clippings}}$, or $median_{\text{Dice coefficients of blends}} - median_{\text{Dice coefficients of complex clippings}} = 0$.

H_1: The median of the Dice coefficients of the source words of blends is not as large as the median of the Dice coefficients of the source words of complex clippings; $median_{\text{Dice coefficients of blends}} \neq median_{\text{Dice coefficients of complex clippings}}$, or $median_{\text{Dice coefficients of blends}} - median_{\text{Dice coefficients of complex clippings}} \neq 0$.

```
> tapply(DICE, PROCESS, median)¶
    Blend ComplClip
   0.2300    0.1195
> tapply(DICE, PROCESS, IQR)¶
    Blend ComplClip
   0.0675    0.0675
```

Here, the assumptions can be tested fairly unproblematically: The values are independent of each other since no word-formation influences another one, the distributions of the data in Figure 54 appear to be rather similar, and a Kolmogorov-Smirnov test of the z-standardized Dice values for both word-formation processes is completely insignificant ($p = 0.9972$).

Unfortunately, computing the U-test is more cumbersome than many

26. According to Bortz, Lienert, and Boehnke (1990:211), the U-test can discover differences of measures of central tendency well even if this assumption is violated.

other tests. First, you transform all Dice coefficients into ranks, and then you compute the sum of all ranks for each word-formation process. Then, both of these *T*-values and the two sample sizes are inserted into the formulae in (50) and (51) to compute two *U*-values, the smaller one of which is the required test statistic.

```
> Ts<-tapply(rank(DICE), PROCESS, sum)¶
```

$$(50) \qquad U_1 = n_1 \cdot n_2 + \frac{n_1 \cdot (n_1 + 1)}{2} - T_1$$

$$(51) \qquad U_2 = n_1 \cdot n_2 + \frac{n_2 \cdot (n_2 + 1)}{2} - T_2$$

```
> n1<-length(DICE[PROCESS=="Blend"])¶
> n2<-length(DICE[PROCESS=="ComplClip"])¶
> U1<-n1*n2+((n1*(n1+1))/2)-Ts[1]¶
> U2<-n1*n2+((n2*(n2+1))/2)-Ts[2]¶
> U.value<-min(U1, U2)¶
```

The *U*-value, 84, can be looked up in a *U*-table or, because there are few *U*-tables for large samples,[27] converted into a normally-distributed *z*-score. This *z*-score is computed as follows. First, you use the formulae in (52) and (53) to compute an expected *U*-value and its dispersion.

$$(52) \qquad U_{\text{expected}} = 0.5 \cdot n_1 \cdot n_2$$

$$(53) \qquad \textit{Dispersion } U_{\text{expected}} = \sqrt{\frac{n_1 \cdot n_2 \cdot (n_1 + n_2 + 1)}{12}}$$

Second, you insert these values together with the observed *U* into (54).

$$(54) \qquad z = \frac{U - U_{expected}}{\textit{Dispersion } U_{expected}}$$

```
> expU<-n1*n2/2¶
> dispersion.expU<-sqrt(n1*n2*(n1+n2+1)/12)¶
> z<-abs((U.value-expU)/dispersion.expU)¶
```

27. Bortz, Lienert and Boehnke (1990:202 and Table 6) provide critical *U*-values for $n \leq 20$ and mention references for tables with critical values for $n \leq 40$ – I at least know of no *U*-tables for larger samples.

To decide whether H_0 can be rejected, you look up this value, 8.038194, in a *z*-table such as Table 36 or you compute a critical *z*-score for $p_{\text{two-tailed}} = 0.05$ with `qnorm` (as mentioned in Section 1.3.4.2 above). Since you have a non-directional H_1, you apply the same logic as above and compute *z*-scores for half of the $p_{\text{two-tailed}}$-values you are interested in:

Table 36. Critical *z*-scores for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001

z-score	*p*-value
1.96	0.05
2.575	0.01
3.291	0.001

```
> qnorm(c(0.9995, 0.995, 0.975, 0.025, 0.005, 0.0005),
   lower.tail=FALSE)¶
[1] -3.290527 -2.575829 -1.959964  1.959964  2.575829
   3.290527
```

It is obvious that the observed *z*-score is not only much larger than the one tabulated for $p_{\text{two-tailed}} = 0.001$ but also very distinctly in the grey-shaded area in Figure 55: the difference between the medians is highly significant, as the non-overlapping notches already anticipated. Plus, you can compute the exact *p*-value with the usual 'mirror function' of `qnorm`.

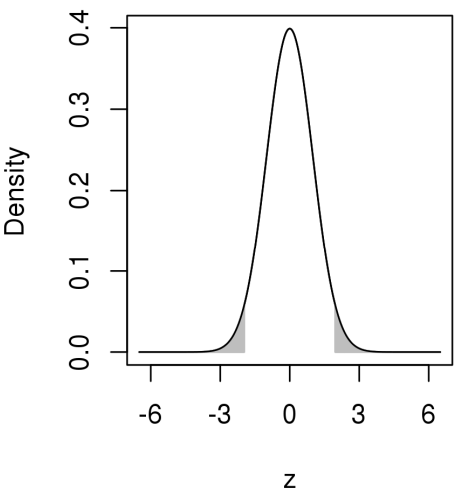

Figure 55. Density function of the standard normal distribution; two-tailed test

```
> 2*pnorm(z, lower.tail=FALSE)¶
[1] 9.117223e-16
```

In R, you compute the *U*-test with the same function as the Wilcoxon test, `wilcox.test`, and again you can either use a formula or two vectors. Apart from these arguments, the following ones are useful, too:

- `alternative`: a character string specifying which H_1 you want to test: the default is `"two.sided"`, other possible values for one-tailed tests are again `"less"` or `"greater"`, which specify how the first-named vector or factor level relates to the second;
- `paired=FALSE` for the *U*-Test for independent samples or `paired=TRUE` for the Wilcoxon test for dependent samples (cf. the following section);
- `exact=TRUE`, if you want to compute an exact test, or `exact=FALSE` if you don't (if you don't change `exact`'s default setting of `NULL` and your data set has fewer than 50 data points and no ties, an exact *p*-value is computed automatically);
- `correct=TRUE` for a continuity correction (the default) and `correct=FALSE` for none;
- `conf.level`: a value between 0 and 1 specifying the size of the confidence interval; the default is 0.95.

The standard version to be used here is this:

```
> wilcox.test(DICE~PROCESS, paired=FALSE, correct=FALSE)¶
 Wilcoxon rank sum test
data:  DICE by PROCESS
W = 2416, p-value = 9.072e-16
alternative hypothesis: true location shift is not equal to 0
```

You get a *U*-value (here referred to as *W*) and a *p*-value; *W* is not the minimum of U_1 and U_2, but the maximum here, which value you get depends on which vector or factor level comes first in the alphabet. The *p*-value here is a bit different from yours since R uses a slightly different algorithm. You can now sum up: "According to a *U*-test, the median Dice coefficient of the source words of blends (0.23, $IQR = 0.0675$) and the median of the Dice coefficients for complex clippings (0.12, $IQR = 0.0675$) are very significantly different: $U = 84$ (or $W = 2416$), $p_{\text{two-tailed}} < 0.0001$. The creators of blends appear to be more concerned with selecting source words that are similar to each other than the creators of complex clippings."

Recommendation(s) for further study:
Dalgaard (2002: 89f.), Crawley (2007: 297f.), Baayen (2008: Section 4.3.1), Sheskin (2011: Test 12)

3.2.4. One dep. variable (ordinal) and one indep. variable (nominal) (dep. samples)

Just like the *U*-test, the test in this section has two major applications. First, you really may have two dependent samples of ordinal data such as when you have a group of subjects perform two rating tasks to test whether each subject's first rating differs from the second. Second, the probably more frequent application arises when you have two dependent samples of ratio-scaled data but cannot do the *t*-test for dependent samples because its distributional assumptions are not met. We will discuss an example of the latter kind in this section.

In a replication of Bencini and Goldberg (2000), Gries and Wulff (2005) studied the question which verbs or sentence structures are more relevant for how German foreign language learners of English categorize sentences. They crossed four syntactic constructions and four verbs to get 16 sentences, each verb in each construction. Each sentence was printed onto a card and 20 advanced German learners of English were given the cards and asked to sort them into four piles of four cards each. The question was whether the subjects' sortings would be based on the verbs or the constructions. To determine the sorting preferences, each subject's four stacks were inspected with regard to how many cards one would minimally have to move to create either four completely verb-based or four completely construction-based sortings. The investigation of this question involves

- one dependent ratio-scaled variable, namely SHIFTS, the number of times a card had to be shifted from one stack to another to create the perfectly clean sortings, and we are interested in the average of these numbers;
- one independent nominal variable, namely CRITERION: *CONSTRUCTION* vs. CRITERION: *VERB*;
- dependent samples since each subject 'generated' two numbers of shifts, one to create the verb-based sorting, one to create the construction-based sorting.

To test some such result for significance, you should first consider a *t*-test for dependent samples since you have two samples of ratio-scaled values. As usual, you begin by formulating the relevant hypotheses:

H_0: The average of the pairwise differences between the numbers of rearrangements towards perfectly verb-based stacks and the numbers of rearrangements towards perfectly construction-based stacks is 0; $mean_{\text{pairwise differerences}} = 0$.

H_1: The average of the pairwise differences between the numbers of rearrangements towards perfectly verb-based stacks and the numbers of rearrangements towards perfectly construction-based stacks is not 0; $mean_{\text{pairwise differerences}} \neq 0$.

Then, you load the data that Gries and Wulff (2005) obtained in their experiment from <_inputfiles/04-3-2-4_sortingstyles.csv>:

```
> SortingStyles<-read.delim(file.choose())¶
> head(SortingStyles, 3); attach(SortingStyles)¶
```

As usual, you compute means and standard deviations and generate a graph of the results.

```
> tapply(SHIFTS, CRITERION, mean)¶
Construction          Verb
        3.45          8.85
> tapply(SHIFTS, CRITERION, sd)¶
Construction          Verb
    4.346505      4.107439
> differences<-SHIFTS[CRITERION=="Construction"]-
   SHIFTS[CRITERION!="Construction"]¶
> stripchart(differences, method="stack", xlim=c(-12, 12),
   xlab="Differences: ->construction minus ->verb");
   abline(v=0, lty=2, col="grey")¶
```

Note: since the two samples are dependent, we are plotting the differences, just as in Section 4.3.2.2 above. You then test the assumption of the *t*-test for dependent samples, the normality of the pairwise differences. Given Figure 56, those are obviously not normal:

```
> shapiro.test(differences)¶
 Shapiro-Wilk normality test
data:  differences
W = 0.7825, p-value = 0.0004797
```

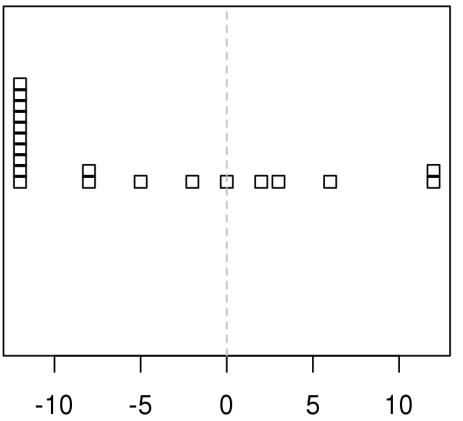

Figure 56. Strip chart of the differences of shifts

You cannot use the *t*-test. Instead, you compute a test for two dependent samples of ordinal variables, the Wilcoxon test.

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test:
  - the pairs of values are independent of each other
  - the populations from which the samples whose central tendencies are tested have been drawn are identically distributed
- Computing the test statistic *T* and *p*

As a first step, you adjust your hypotheses to the ordinal level of measurement, you then compute the medians and their interquartile ranges:

H_0: $median_{\text{pairwise differerences}} = 0$
H_1: $median_{\text{pairwise differerences}} \neq 0$

```
> tapply(SHIFTS, CRITERION, median)¶
Construction         Verb
           1           11
> tapply(SHIFTS, CRITERION, IQR)¶
Construction         Verb
        6.25         6.25
```

The assumptions appear to be met because the pairs of values are independent of each other (since the sorting of any one subject does not affect any other subject's sorting) and, somewhat informally, there is little reason to assume that the populations are distributed differently especially since most of the values to achieve a perfect verb-based sorting are the exact reverse of the values to get a perfect construction-based sorting. Thus, you compute the Wilcoxon test; for reasons of space we only consider the standard variant. First, you transform the vector of pairwise differences, which you already computed for the Shapiro-Wilk test, into ranks:

```
> ranks<-rank(abs(differences))¶
```

Second, all ranks whose difference was negative are summed to a value *T*-, and all ranks whose difference was positive are summed to *T*+; the smaller of the two values is the required test statistic *T*:[28]

```
> T.minus<-sum(ranks[differences<0])¶
> T.plus<-sum(ranks[differences>0])¶
> T.value<-min(T.minus, T.plus)¶
```

This *T*-value of 41.5 can be looked up in a *T*-table (Table 37), but note that here, for a significant result, the observed test statistic must be *smaller* than the tabulated one.

Table 37. Critical *T*-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $14 \leq df \leq 16$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$n = 19$	46	32	18
$n = 20$	52	37	21
$n = 21$	58	42	25

The observed *T*-value of 41.5 is smaller than the one tabulated for $n = 20$ and $p = 0.05$ (but larger than the one tabulated for $n = 20$ and $p = 0.01$): the result is significant.

Let us now do this test with R: You already know the function for the Wilcoxon test so we need not discuss it again in detail. The relevant difference is that you now instruct R to treat the samples as dependent/paired. As nearly always, you can use the formula or the vector-based function call.

28. The way of computation discussed here is the one described in Bortz (2005). It disregards ties and cases where the differences are zero; cf. also Sheskin (2011:812).

```
> wilcox.test(SHIFTS~CRITERION, paired=TRUE, exact=FALSE,
   correct=FALSE)¶
 Wilcoxon signed rank test
data:  SHIFTS by CRITERION
V = 36.5, p-value = 0.01527
alternative hypothesis: true location shift is not equal to 0
```

R computes the test statistic differently but arrives at the same kind of decision: the result is significant, but not very significant. To sum up: "On the whole, the 20 subjects exhibited a strong preference for a construction-based sorting style: the median number of card rearrangements to arrive at a perfectly construction-based sorting was 1 while the median number of card rearrangements to arrive at a perfectly verb-based sorting was 11 (both *IQR*s = 6.25). According to a Wilcoxon test, this difference is significant: $V = 36.5$, $p_{\text{two-tailed}} = 0.0153$. In this experiment, the syntactic patterns were a more salient characteristic than the verbs (when it comes to what triggered the sorting preferences)."

Recommendation(s) for further study:
- Dalgaard (2002:92), Sheskin (2011: Test 18)

4. Coefficients of correlation and linear regression

In this section, we discuss the significance tests for the coefficients of correlation discussed in Section 3.2.3.

4.1. The significance of the product-moment correlation

While the manual computation of the product-moment correlation above was a bit complex, its significance test is not. It involves these steps:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test: the population from which the sample was drawn is bivariately normally distributed. Since this criterion *can* be hard to test (cf. Bortz 2005: 213f.), we simply require both samples to be distributed normally
- Computing the test statistic t, df, and p

Let us return to the example in Section 3.2.3, where you computed a correlation coefficient of 0.9337 for the correlation of the lengths of 20 words and their reaction times. You formulate the hypotheses and we assume for now your H_1 is non-directional.

H_0: The length of a word in letters does not correlate with the word's reaction time in a lexical decision task; $r = 0$.
H_1: The length of a word in letters correlates with the word's reaction time in a lexical decision task; $r \neq 0$.

You load the data from <_inputfiles/04-4_reactiontimes.csv>:

```
> ReactTime<-read.delim (file.choose())¶
> str(ReactTime); attach(ReactTime)¶
```

Since we already generated a scatterplot above (cf. Figure 35 and Figure 36), we will skip plotting for now. We do, however, have to test the assumption of normality of both vectors. You can either proceed in a stepwise fashion and enter `shapiro.test(LENGTH)¶` and `shapiro.test(MS_LEARNER)¶` or use a shorter variant:

```
> apply(ReactTime[,2:3], 2, shapiro.test)¶
$LENGTH
 Shapiro-Wilk normality test
data:  newX[, i]
W = 0.9748, p-value = 0.8502
$MS_LEARNER
 Shapiro-Wilk normality test
data:  newX[, i]
W = 0.9577, p-value = 0.4991
```

This line of code means 'take the data mentioned in the first argument of `apply` (the second and third column of the data frame `ReactTime`), look at them column by column (the 2 in the second argument slot – a 1 would look at them row-wise; recall this notation from `prop.table` in Section 3.2.1), and apply the function `shapiro.test` to each of these columns. Clearly, both variables do not differ significantly from a normality.

To compute the test statistic *t*, you insert the correlation coefficient *r* and the number of correlated value pairs *n* into the formula in (55):

(55) $$t = \left| \frac{r \cdot \sqrt{n-2}}{\sqrt{1-r^2}} \right|$$

```
> r<-cor(LENGTH, MS_LEARNER, method="pearson")¶
> numerator<-r*sqrt(length(LENGTH)-2)¶
> denominator<-sqrt(1-r^2)¶
> t.value<-abs(numerator/denominator)¶
```

This *t*-value, 11.06507, has $df = n-2 = 18$ degrees of freedom.

```
> df<-length(LENGTH)-2¶
```

Just as with the *t*-tests before, you can now look this *t*-value up in a *t*-table, or you can compute a critical value: if the observed *t*-value is higher than the tabulated/critical one, then *r* is significantly different from 0. Since your *t*-value is much larger than even the one for $p = 0.001$, the correlation is highly significant.

```
> qt(c(0.025, 0.975), 18, lower.tail=FALSE)¶
[1]  2.100922 -2.100922
```

Table 38. Critical *t*-values for $p_{\text{two-tailed}} = 0.05$, 0.01, and 0.001 for $17 \leq df \leq 19$

	$p = 0.05$	$p = 0.01$	$p = 0.001$
$df = 17$	2.1098	2.8982	3.9561
$df = 18$	2.1009	2.8784	3.9216
$df = 19$	2.093	2.8609	3.8834

The exact *p*-value can be computed as follows, and do not forget to again double the *p*-value.

```
> 2*pt(t.value, 18, lower.tail=FALSE)¶
[1] 1.841060e-09
```

This *p*-value is obviously much smaller than 0.001. However, you will already suspect that there is an easier way to get all this done. Instead of the function `cor`, which we used in Section 3.2.3 above, you simply use `cor.test` with the two vectors whose correlation you are interested in (and, if you have a directional H_1, you specify whether you expect the correlation to be less than 0 (i.e., negative) or greater than 0 (i.e., positive)

using `alternative=…`):

```
> cor.test(LENGTH, MS_LEARNER, method="pearson")¶
 Pearson's product-moment correlation
data:  LENGTH and MS_LEARNER
t = 11.0651, df = 18, p-value = 1.841e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8370608 0.9738525
sample estimates:
      cor
0.9337171
```

Here are the (edited) results of the corresponding linear regression:

```
> model<-lm(MS_LEARNER~LENGTH)¶
> summary(model)¶
Call:
lm(formula = MS_LEARNER ~ LENGTH)

Residuals:
     Min       1Q   Median       3Q      Max
-22.1368  -7.8109   0.8413   7.9499  18.9501
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  93.6149     9.9169    9.44 2.15e-08 ***
LENGTH       10.3044     0.9313   11.06 1.84e-09 ***
---                                                […]
Multiple R-Squared: 0.8718,     Adjusted R-squared: 0.8647
F-statistic: 122.4 on 1 and 18 DF,  p-value: 1.841e-09
```

We begin at the bottom: the last row contains information we already know. The *F*-value is our *t*-value squared; we find the 18 degrees of freedom and the *p*-value we computed. In the line above that, you find the coefficient of determination you know plus an adjusted version we will only talk about later (cf. Section 5.2). We ignore the edited-out line about the residual standard error for now and the legend for the *p*-values. The table above that shows the intercept and the slope we computed in Section 3.2.3 (in the column labeled "Estimate"), their standard errors, *t*-values – do you recognize the *t*-value from above? – and *p*-values. The *p*-value for LENGTH says whether the slope of the regression line is significantly different from 0; the *p*-value for the intercept says whether the intercept of 93.6149 is significantly different from 0. We skip the info on the residuals because we discussed above how you can investigate those yourself (with `residuals(model)¶`).

There is one final but immensely useful thing to be discussed. Recall that above we used the function `predict` to get the predicted reaction times

for every observed word length, but also predicted reaction times for non-observed word lengths. The function `predict` can return more than this, however: it can also return confidence intervals for the predictions, which also allows to plot the regression line with its confidence interval. Since we will use this frequently in Chapter 5, we will go over one example here, which will involve three steps.

The first step repeats what we did above: we generate a data frame `preds.hyp` that contains a range of values covering the observed word lengths and that we will pass on to `predict`, and we do that as in Section 3.2.3 with `expand.grid()`. I call it `preds.hyp` to indicate that these are predictions from the model not for the actually observed lengths but for a range of hypothetical values. Note again that the column in `preds.hyp` has the same name as the independent variable in `model`.

```
> preds.hyp<-expand.grid(LENGTH=min(LENGTH):max(LENGTH))¶
```

The second step is also similar to Section 3.2.3 above, but with two small changes. We not only use `predict` to generate the predictions from `model` for this data frame, but (i) we also let R compute the confidence intervals for all predictions and (ii) we make the predictions and the confidence intervals columns 2 to 4 in `preds.hyp`:

```
> preds.hyp[c("PREDICTIONS", "LOWER", "UPPER")]<-predict(
   model, newdata=preds.hyp, interval="confidence")¶
```

If you look at the data frame `preds.hyp` now, you will see we now have a very nice result: the independent variable is in the column `preds.hyp$LENGTH`, the predicted dependent variable is in the column `preds.hyp$PREDICTIONS`, and the lower and upper confidence intervals for each prediction are in the columns `preds.hyp$LOWER` and `preds.hyp$UPPER` respectively.

The third step now involves generating a nice plot. The following code pulls many things together and introduces the function `matlines`:

```
> plot(MS_LEARNER~LENGTH, xlab="Word length in letters",
   ylab="Reaction time of learners in ms", pch=16,
   col=rgb(0, 0, 0, 70, maxColorValue=255)); grid()¶
> matlines(preds.hyp[,1], preds.hyp[,2:4], lwd=c(2, 1,
   1), lty=c(1, 2, 2), col=c("black", "blue", "blue"))¶
```

The first line just generates a regular scatterplot – the only new thing is the use of the function `rgb` to use a semi-transparent greyshade to avoid

information loss through overplotting. The second line uses `matlines`: the first argument is the first column of `preds.hyp` and provides the *x*-values for the lines to be plotted. The second argument is columns 2 to 4 of `preds.hyp` and provides three different sets of *y*-values to plot with separate lines: first the predicted values (= the regression line), second and third the lower and upper limits of the confidence intervals. The arguments `lwd` (line width), `lty` (line type), and `col` (color) describe what the lines should look like, in the order in which they appear in `preds.hyp`. The result you see when you run the code: a scatterplot with a regression line and its confidence band, and we can see again why the correlation is so high: not only is the regression line a good summary of the data, the confidence band is quite narrow around it and many points are right in it or very close to it.

This was a very detailed description, but since we will use this many times in Chapter 5, this is time well spent. To sum up: "The lengths of the words in letters and the reaction times in the experiment correlate highly positively with each other: $r = 0.9337$; adjusted $R^2 = 0.8647$. This correlation is highly significant: $t = 11.07$; $df = 18$; $p < 0.001$. The linear regression shows that every additional letter increases the reaction time by approximately 10.3 ms."

In Section 5.2, we deal with the extensions of linear regression to cases where we include more than one independent variable, and we will also discuss more comprehensive tests of the regression's assumptions (using `plot(model)¶`).

4.2. The significance of Kendall's Tau

If you need a *p*-value for Kendall's tau τ, you follow this procedure:

Procedure
- Formulating the hypotheses
- Computing descriptive statistics and visualizing the data
- Testing the assumption(s) of the test: the data from both samples are at least ordinal
- Computing the test statistic *z* and *p*

Again, we simply use the example from Section 3.2.3 above (even though we know we can actually use the product-moment correlation; we use this example again just for simplicity's sake). How to formulate the hypotheses should be obvious by now:

H_0: The length of a word in letters does not correlate with the word's reaction time in a lexical decision task; $\tau = 0$.

H_1: The length of a word in letters correlates with the word's reaction time in a lexical decision task; $\tau \neq 0$.

As for the assumption: we already know the data are ordinal – after all, we know they are even interval/ratio-scaled. You load the data again from <_inputfiles/03-2-3_reactiontimes.csv> and compute Kendall's τ:

```
> ReactTime<-read.delim (file.choose())¶
> str(ReactTime); attach(ReactTime)¶
> tau<-cor(LENGTH, MS_LEARNER, method="kendall")¶
```

To test Kendall's tau τ for significance, you compute a z-score of the kind that is by now familiar. You insert τ and the number of value pairs n into the formula in (56).

(56) $$z = |\tau| \div \sqrt{\frac{2 \cdot (2 \cdot n + 5)}{9 \cdot n \cdot (n - 1)}}$$

In R:

```
> numerator.root<-2*(2*length(LENGTH)+5)¶
> denominator.root<-9*length(LENGTH)*(length(LENGTH)-1)¶
> z.score<-abs(tau)/sqrt(numerator.root/denominator.root)¶
> z.score¶
[1] 5.048596
```

This value can be looked up in a z-table (cf. Table 36) or you generate these values yourself. The z-score for a significant two-tailed test must cut off at least 2.5% of the area under the standard normal distribution:

```
> qnorm(c(0.9995, 0.995, 0.975, 0.025, 0.005, 0.0005),
   lower.tail=FALSE)¶
[1] -3.290527 -2.575829 -1.959964  1.959964  2.575829
   3.290527
```

For a result to be significant, the z-score must be larger than 1.96. Since the observed z-score is even larger than 5, this result is highly significant:

```
> 2*pnorm(z.score, lower.tail=FALSE)¶
[1] 4.450685e-07
```

The function to get this result much faster is again `cor.test`. Since R uses a slightly different method of calculation, you get a slightly different *z*-score and *p*-value, but for all practical purposes the results are identical.

```
> cor.test(LENGTH, MS_LEARNER, method="kendall")¶
 Kendall's rank correlation tau
data:  LENGTH and MS_LEARNER
z = 4.8836, p-value = 1.042e-06
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.8189904
```

(The warning refers to ties such as that the length value 11 occurs more than once). To sum up: "The lengths of the words in letters and the reaction times in the experiment correlate highly positively with each other: $\tau = 0.819$, $z = 5.05$; $p < 0.001$."

4.3. Correlation and causality

Especially in the area of correlations, but also more generally, you need to bear in mind a few things even if H_0 is rejected: First, one can often hear a person A making a statement about a correlation (maybe even a significant one) by saying "The more X, the more Y" and then hear a person B objecting to that correlation on the grounds that B knows of an exception. This argument is flawed. The exception quoted by B would only invalidate A's statement if A considered the correlation to be perfect ($r = 1$ or $r = -1$) – but if A did not mean that (and A never does!), then there may be a strong and significant correlation although there is one exception (or more). The exception or exceptions are the reason why the correlation is not 1 or -1 but 'only', say, 0.9337. Second, a correlation as such does not necessarily imply causality. As is sometimes said, a correlation between X and Y is a *necessary* condition for a causal relation between X and Y, but not a *sufficient* one, as you can see from many examples:

– There is a positive correlation between the number of firefighters trying to extinguish a fire and the amount of damage that is caused at the site where the fire was fought. This does of course not mean that the firefighters arrive at the site and destroy as much as they can – the correlation results from a third, confounding variable, the size of the fire: the larger the fire, the more firefighters are called to help extinguish it *and*

the more damage the fire causes.
- There is a negative correlation between the amount of hair men have and their income which is unfortunately only due to the effect of a third variable: the men's age.
- There is a positive correlation such that the more likely a drug addict was to go to therapy to get off of his addiction, the more likely he was to die. This is not because the therapy leads to death – the confounding variable in the background correlated with both is the severity of the addiction: the more severely addicted addicts were, the more likely they were to go to therapy, but also the more likely they already were to die.

Thus, beware of jumping to conclusions …
Now you should do the exercise(s) for Chapter 4 …

Recommendation(s) for further study
- the functions `ckappa` and `lkappa` (from the library `psy`) to compute the kappa coefficient and test how well two or more raters conform in their judgments of stimuli
- the function `cronbach` (from the library `psy`) to compute Cronbach's alpha and test how consistently several variables measure a construct the variables are supposed to reflect
- Crawley (2007: Ch. 10), Baayen (2008: Section 4.3.2), Johnson (2008: Section 2.4), Sheskin (2011: Test 28, 30, 31, 32)
- the function `hints` (from the library `hints`) to get ideas about what to do next with a particular object